COMPLEMENTING THE GSP ROUTING PROTOCOL IN WIRELESS SENSOR NETWORKS

by

María Gabriela Calle Torres

BS Electronics Engineering, Universidad Pontificia Bolivariana, 1995

MS Telecommunications, University of Pittsburgh, 2006

Submitted to the Graduate Faculty of

School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

María Gabriela Calle Torres

It was defended on

April 7th, 2009

and approved by

Dr. Richard Thompson, Professor, Telecommunications, SIS

Dr. Prashant Krishnamurthy, Associate Professor, Telecommunications, SIS

Dr. Vladimir Zadorozhny, Associate Professor, Information Science, SIS

Dr. Tommaso Melodia, Assistant Professor, EE, University at Buffalo

Dissertation Advisor: Dr. Joseph Kabara, Assistant Professor, SIS

COMPLEMENTING THE GSP ROUTING PROTOCOL IN WIRELESS SENSOR NETWORKS

María Gabriela Calle Torres, MST

University of Pittsburgh, 2009

Gossip-Based Sleep Protocol (GSP) is a routing protocol in the flooding family with overhead generated by duplicate packets. GSP does not have other sources of overhead or additional information requirements common in routing protocols, such as routing packets, geographical information, addressing or explicit route computation. Because of its simple functionality, GSP is a candidate routing protocol for Wireless Sensor Networks. However, previous research showed that GSP uses the majority of energy in the network by keeping the nodes with their radios on ready to receive, even when there are no transmissions, situation known as Idle Listening. Complementing GSP implies creating additional protocols that make use of GSP particular characteristics in order to improve performance without additional overhead. The research analyzes the performance of GSP with different topologies, number of hops from source to destination and node densities, and presents one alternative protocol to complement GSP decreasing idle listening, number of duplicate packets in the network and overall energy consumption. The study compared the results of this alternative protocol, MACGSP6, to a protocol stack proposed for Wireless Sensor Networks: Sensor MAC (S-MAC) with Dynamic Source Routing (DSR), showing the advantages and disadvantages of the different approaches.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I want to thank firstly to God, my strength and guidance through all stages in my life.

I would like to express my deepest gratitude to my advisor, Dr. Joseph Kabara, who taught me how to think critically and how to explore wireless communications research in innovative ways. His guidance and constant help were determinant in achieving my academic and research goals. I also want to thank the members of my PhD Dissertation Committee, Dr. Richard Thompson, Dr. Prashant Krishnamurthy, Dr. Vladimir Zadorozhny and Dr. Tommaso Melodia, for their guidance and valuable suggestions to improve this research.

I want to thank Universidad del Norte in Barranquilla, Colombia, and The Graduate Program in Telecommunications and Networking at University of Pittsburgh for sponsoring me during the time I pursued my graduate studies. I also want to thank my fellow students at SiNE: Dr. Natthapol Pongthaipat, Dr. Debdhanit Yupho, Yuttasart Nitipaichit and Aylin Aksu for giving me very valuable ideas and feedback in my research.

I am most grateful to my parents, Carmen and Mario, my sister Carmen and my brothers Jesus and Miguel. They, along with all family and friends in Colombia, Pittsburgh and elsewhere, showed me their love and support all the time, making this process more bearable.

1.0 INTRODUCTION

Wireless Sensor Networks (WSNs) have hundreds or potentially thousands of nodes built as small computers capable of measuring some physical characteristics from the environment and transmitting information using a radio link. WSNs can be used in monitoring applications such as weather, crops, surveillance, human health care and structural health [1], [2]. Nodes have constraints in memory, processing power and energy usage [3], since they are supposed to use small hardware platforms and they are very likely to be battery powered. Once the battery is depleted, it is often very difficult to recharge or replace it, so the node is considered dead [1]. Figure 1 shows one example sensor node: Crossbow's Mica2.

Figure 1. MPR400 (Mica2).

One possible application of WSNs is structural health monitoring of a bridge such as the Golden Gate. There may be hundreds of nodes measuring vibrations in the bridge and transmitting this information to a sink (main receiver) not located on the bridge. Engineers can use this information to schedule maintenance or repair jobs. If the batteries in the nodes last a week, somebody must climb the bridge every week to replace hundreds of batteries, making the

application very costly. Another example application is leakage in an industrial plant with hazardous chemicals. People must evacuate, but a WSN can employ sensor nodes dropped from a plane. In this case, there is no control over the network topology since the nodes may land in random locations, and no way to change batteries either, because people cannot access the site.

Energy use is not the only concern in WSNs. Diverse applications have different requirements in throughput, delay, network topology, etc. Regarding physical topology, the bridge monitoring and the chemical leak monitoring are applications using nodes possibly located in random positions. If the situation is patient monitoring in a medical facility, the network may need a specific layout because of certain medical equipment, such as MRI machines, may obstruct signals from the nodes. Regarding delay, human health monitoring may have a tighter delay requirement than the other two mentioned applications since vital signs of the patient might indicate the need of immediate treatment within seconds. As different applications have different requirements, a very plausible scenario includes not one communication standard for all possible applications but several standards, every one designed to optimize the critical parameter(s). There are no standard protocols widely accepted in WSNs in any of the communication layers in the OSI sense.

1.1 MOTIVATION

WSNs continue to be a major field of research because each of their many applications has unique technical challenges and this technology cannot utilize traditional communication protocols due to the aforementioned constraints. The literature describes many examples of Media Access Control (MAC) and routing protocols proposed to operate within the restrictions

present in WSNs. However, several problems exist in the research so far. The first problem is employing variations of schemes currently working in networks where the main constraint is bandwidth (e.g. Wireless Local Area Networks, WLANs) and not energy use. Examples of this approach are Sensor MAC (S-MAC) [4] and Timeout-MAC (T-MAC) [5]. Other problem is complexity: protocols such as Low-Energy Adaptive Clustering Hierarchy (LEACH) [6] and Power Efficient and Delay Aware Medium Access Protocol (PEDAMACS) [7] are highly sophisticated and there is not an easy way of implementing them in actual hardware devices within the limitations of WSNs. Another problem found in the current research in WSNs is overhead. Protocols such as basic Carrier Sense Multiple Access (CSMA) [8], Sift with small data packets [9] and Gossip-based Sleep Protocol (GSP) [10] operate without imposing additional overhead. However, other protocols include overhead in the form of control packets (e.g S-MAC [4] and Sensor Protocol for Information via Negotiation SPIN [11]), additional bytes in the headers (e.g. Berkeley MAC (B-MAC) [12] and Minimum Cost Forwarding [13]), geographic information requirements (e.g. Directed Diffusion [14]), table creation and maintenance in the nodes (e.g. Directed Diffusion [14]), etc. Sections 2.2 and 2.3 describe all these protocols in more detail.

GSP is a routing protocol in the Flooding family. Since flooding protocols retransmit every packet received, there is no control information overhead. GSP needs no addressing, no route calculation and no state information. A routing protocol with these characteristics may be interesting for applications where the network has several sinks or high reliability requirements. However, because GSP belongs to the family of flooding protocols, nodes retransmit multiple copies of a packet across the network, generating overhead [15]. The GSP routing protocol was

shown to increase network lifetime in 802.11 networks with mobile nodes [10] and to improve on the Average Remaining Energy (ARE) in networks with static nodes [3].

1.2 PROBLEM STATEMENT

Early research in WSNs followed the OSI model paradigm assuming all communication layers are independent of each other. However, WSNs and energy-constrained networks in general may benefit from including several layers when designing the communication system. Every protocol in a particular layer should be matched to protocols in layers below and above in a way to complement the functionality and fulfill the goals of the network. The study in [16] tried to improve GSP performance without increasing overhead, by using two MAC protocols designed explicitly to complement GSP, MACGSP1 and MACGSP2. The latter protocol showed the best improvement by decreasing energy use and number of duplicates in the network without increasing overhead or delay and with a small reduction in packet reception probability. However, packet reception probability is highly dependent on p_{gsp}, the Gossip Probability, which is the only parameter in GSP. Additionally, GSP generates overhead in duplicates, which may continue to propagate in the network indefinitely. A clear conclusion from [16] is MACGSP2 spends the majority of energy during Idle Listening, that is turning on the radio to verify the medium when neighboring nodes are not transmitting, thus wasting energy. Several applications in WSNs may benefit from using GSP and some other protocols designed specifically with GSP functionality as design criteria may improve the characteristics mentioned.

The study in [16] only considered square grid network topologies. However, the performance of MACGSP1 and 2 may vary in different topologies, with different node densities and different number of hops required to traverse the network from source to destination.

The objectives of this research are:

Decrease energy use of GSP during Idle Listening.

Decrease the number of duplicates generated by GSP without affecting metrics such as delay and packet reception probability.

Verify the behavior of the GSP family of protocols in different topologies.

Compare the GSP family to other protocols previously proposed for WSN, specifically, protocols designed under the independent layer paradigm according to the OSI model.

The limitations of this work are the following:

Research includes protocol characteristics in three lower layers in OSI model and low traffic applications such as crop monitoring, were nodes may send data every several minutes.

The most important constraint is energy use. Other constraints are secondary.

Nodes in the networks are stationary and synchronized. They send data with a small number of bytes, thus information fits in one data packet.

The hypotheses of this research are:

H1: GSP has higher energy use, packet reception probability and number of duplicates than MACGSP1 and 2, whether these metrics are evaluated until the first packet is received by the sink or until the last duplicate is eliminated in the network.

H2: Use of overhearing as a source of implicit ACKs improves packet reception probability for higher values of p_{gsp}, without an overall increase in energy use.

H3: MACGSP1 and MACGSP2 decrease energy use of GSP if a perfect capture effect exists, irrespective of physical topology.

H4: Increasing the node density in the network increases the packet reception probability when using the GSP family of protocols or some other protocols present in the literature such as S-MAC and Dynamic Source Routing (DSR).

H5: For the GSP family of protocols, during listening operations, a duty cycle other than 100% can decrease energy use without affecting packet reception probability or delay and can be implemented in a decentralized manner, with no additional overhead bits.

H6: GSP, when complemented by a MACGSPn protocol, has lower energy use, lower delay and packet reception probability compared to protocols such as S-MAC and DSR.

These hypotheses were tested using both simulations and closed form analytical solutions. Simulations employed 40 runs, transmitting 200 packets from source to destination on each run, with statistical validation of 90% confidence levels. Unless noted, all figures include this validation without showing confidence intervals to make figures clearer. Whenever results overlap, the lines in the figures are close enough that they do not exhibit observable differences. When lines have observable differences in the figures, the results do not overlap at 90% confidence levels. Analytical results are shown with the simulation results for comparison. In all cases, the analytical results matched the simulation results. Both the simulations and the analytical results used a radio channel model which allows nodes within a distance r to receive packets without error, and all nodes at a greater distance to not receive packets. The radio channel model also accounted for the Capture Effect, thus packet dropping occurs according to the capture probability if several packets arrive to the destination simultaneously. One packet is

also lost if all receivers in range have their radios off. The investigation employed one source and one sink.

The organization of the document is as follows:

Section 2 presents Background in the three lower layers of the communication model in the OSI sense in the context of WSNs, including example protocols proposed in the literature for every one of these layers, the Cross-Layer approach and Layering as Optimization Decomposition. The section also reviews energy use models and presents some examples of energy harvesting that may be useful for WSNs. The last subsection deals with Cost-based models and the approach followed in the study of databases for query optimization. The method for generating these models may be extended to generate Cost-based models for WSNs.

Section 3 shows the results obtained so far in the study of GSP, including three published works authored by Maria Calle and Joseph Kabara: "Measuring Energy Consumption in Wireless Sensor Networks Using GSP", PIMRC 2006 [8], "MAC Protocols for GSP in Wireless Sensor Networks," Journal of Networks, 2008 [16] and “Influence of Capture Effect on the Effectiveness of GSP on Wireless Sensor Networks”, DSN 2008 [17]. The section shows one energy use model developed using GSP, two MAC protocols designed for enhancing GSP characteristics and the impact of Capture Effect in GSP performance.

Section 4 presents the results of this Dissertation in complementing GSP, improvements achieved and results using different topologies. The section also presents analytical models for the GSP family of protocols, which allow validation of simulation results when evaluating the models with different values of p_{gsp}, the Gossip Probability. Additionally, the section shows comparison of the GSP family of protocols with protocols previously presented in the literature

and the influence of different factors in forwarding one packet from source to destination. Section 5 concludes the Dissertation.

2.0 BACKGROUND

2.1 PHYSICAL LAYER

The Physical Layer of any communication system has three components: Transmitter, Channel and Receiver [18]. Wireless communications have particular requirements since channel conditions are harsher than those in wired communications are and WSNs are no exceptions to this rule. Sections 2.1.1 to 2.1.3 explain the roles of the three components and their effects in WSNs hardware and protocol design. Note transmitter and receiver are implemented using hardware, which performs signal transformations required, and software, used for protocol implementation. Hardware ultimately expends the energy in a communication system, but protocols define the rules of communication and thus modify the rate of energy use.

2.1.1 Channel models

WSNs must transmit information in such a way that data arrives to the destination not damaged by channel or interference issues, but using the minimum resources possible in processing power, memory and energy. The resources required are a function of the channel behavior and many models exist to represent it. One example of an optimistic model is the study of S-MAC where the channel does not affect the information and collisions are the only source of packet loss [4]. However, one LEACH study in [19] shows an overly pessimistic model assuming a path

loss index of 4 (definition of this concept comes later in this chapter), even though there are no obstacles or movement in the settings. Sift protocol [9] also uses an overly pessimistic model because all cases of simultaneous packet receptions generate collisions, which is not always true (see section c about capture effect). The Section presents different channel models and their impact in protocol design for wireless communications.

a. Additive White Gaussian Noise Channel (AWGN)

AWGN is the basic channel model where electromagnetic noise is the only phenomenon altering the signal. The noise has Gaussian distribution with zero mean and it affects the signal in an additive manner [18]. AWGN model allows performance comparisons of different modulation schemes particularly in Bit Error Rate (BER, ratio of the bits received with errors out of all bits transmitted). Sections b to d present more conditions affecting the signal in wireless communications.

b. Path Loss

Other phenomena affect the signal in wireless channels and not just noise. The following methods may model the propagation of a signal [18]:

Free Space or Line of Sight: transmitter and receiver have a direct link without obstructions.

Reflection: electromagnetic waves bounce on different obstacles when traveling from source to destination. Every reflection attenuates the signal.

Diffraction: transmitted waves bend around obstacles when going from source to destination. Diffraction generates more losses than Reflection alone [18].

Ideally, the receiver detects only one signal, as in Free Space propagation. However, most probably the receiver perceives combined radio waves from different directions at slightly different times. The situation is known as Multipath propagation [18].

The three propagation methods consider a signal transmitted with one Power Level (P_t) which in turn arrives at the receiver with a different Power Level (P_r). The difference between these two levels is known as Path Loss, usually expressed a function of an exponent α. P_r is proportional to α, to the distance d traveled by the signal, to a reference distance d_0 (e.g. 1 meter) and the constant L_0 computed at d_0. Equation (1) shows a model to compute P_r.

$$P_r(d) = (P_t / (L_0 (d/d_0)^{\alpha}) \quad (1)$$

Note α = 2 when using Free Space propagation, but it changes for other conditions such as flat rural environments (α=3) and dense urban environments with skyscrapers (α=4.5) [18]. Path loss affects the transmitted signal in a multiplicative way. One alternative to compensate for path loss is increasing P_t, which in turn increases energy use in the transmitter.

c. Fading Channels

When either the transmitter or receiver is moving, the propagation phenomena vary with time, generating fading. It also may occur if both transmitter and receiver are stationary but the environment is changing [18]. The following are two categories of fading channels.

Large Scale Fading occurs when signal exhibits a "large" scale variation of signal strength with the distance. Since changes in the signal are "large", it is possible to analyze the situation by computing the average signal strength over short epochs or over short distances. Path loss phenomenon is a form of large scale fading. Other form is shadow fading (also called slow or Lognormal fading), which happens when both transmitter and receiver are stationary.

Large scale fading affects the coverage area of the transmitter, so in presence of this type of fading the effective coverage area reduces unless transmission power is increased [18].

Small Scale Fading happens when the amplitude of the signal changes rapidly with time, influenced by Multipath propagation, movement of transmitter, receiver or objects around them. The rate of change in channel conditions determines how the signals are affected. When the receiver node is moving, Doppler Shift determines how frequently the signal changes, which depends upon the speed of the mobile terminal v, the carrier frequency f_c, and the speed of light c [18]. Equation (2) shows the maximum Doppler shift f_m:

$$f_m = f_c \, v / c \tag{2}$$

Using Doppler shift, equation (3) estimates the time during which the channel can be assumed to be constant (Coherence Time T_c) [18]:

$$T_c = 9 / (16\pi f_m) \tag{3}$$

Slow fading (shadowing or Lognormal fading) occurs when the Coherence Time is relatively larger than the symbol duration, which is the case when both terminals are stationary and the symbol is affected in a constant manner by the channel. Fast fading arises in the opposite case: when Coherence Time is smaller than symbol duration and fading may destroy the symbol. Two examples of fast fading are Rayleigh and Rice. Rayleigh fading occurs when two portable devices (which can move but the communication takes place when they are stationary) try to communicate but there is no direct path between them, so the signal propagates through different paths generated by reflections. Rice fading occurs in portable devices when there is one dominant path, such as direct line of sight [18].

Coherence Time affects the receiver performance, specifically Bit Error Rate. Diversity schemes help to compensate for this situation and examples include Time diversity (repeat the

information at different times), Frequency diversity (repeat the information in different frequency channels), Space diversity (repeat the signal using different antennae, such as Multiple Input Multiple Output (MIMO) systems), Angle diversity (using sectored antennae) and Polarization diversity (different polarization in the same antenna) [18]. Error Correcting codes are special cases of Time diversity. Section 2.2.1 explains Error Control in more depth.

The channel may also have different frequency response for different frequencies in the signal, so, similar to Coherence Time, there is a Coherence Bandwidth, which is the range of frequencies over which the channel can be considered to be flat. Coherence Bandwidth limits the maximum data rate and influences the design of transmitter and receiver as well as T_c.

d. Channel Classification

The following is a classification of Wireless Channels according to the concepts presented in this section:

Time-flat channels: do not change their conditions with time. One example is stationary nodes within a non-changing environment.

Frequency-flat channels: their frequency response is constant over a bandwidth greater than or equal to the signal bandwidth.

Time-selective channels: change their conditions with time, such as terminals moving with Rayleigh fading present.

Frequency-selective channels: do not have a flat response over the bandwidth of the signal [18].

The proposed research focuses in networks with large node density thus line of sight exists between adjacent nodes, and the nodes do not move thus shadowing applies. Applications

considered for this research need few bytes to send information (such as monitoring) so required data rate is small and the channel model is frequency-flat.

2.1.2 Transmitter and Receiver

The transmitter adapts information signals to send them through the channel and the receiver should be able to reproduce the information at the destination [18]. Figure 2 shows a general block diagram of a transmitter.

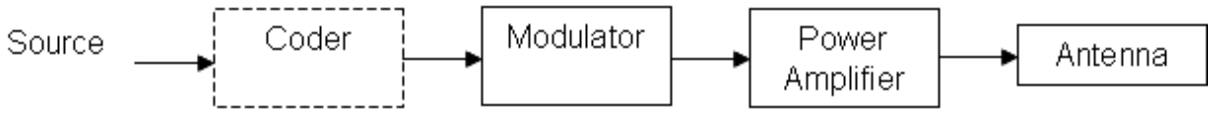

Figure 2. Radio Transmitter general functions. Coder is optional.

The source generates information in digital or analog form. Digital communication systems use coders to transform the information to a digital format appropriate for transmission. Analog communication systems do not usually employ coders, but both systems utilize modulators to embed the information in signal waveforms suitable for wireless transmission. Power amplifiers give the signals enough amplitude to overcome the attenuation caused by the medium and to reach the receiver. The antenna is the impedance matching element and it radiates signal to the medium. Coder and modulator generate energy use but the power amplifier expends the most energy in a wireless communication system [18]. The receiver performs opposite operations to the ones performed by the transmitter. Many different modulation schemes for wireless communication exist and a complete description of all of them is outside the scope of this work. Since WSNs require simple devices to transmit the information with minimum energy use and minimum cost, the following are examples of simple modulation schemes.

a. Amplitude Modulation

Amplitude Modulation is a linear function inscribing the information into the amplitude of the carrier signal. Analog (Amplitude Modulation, AM) or digital (Amplitude Shift Keying, ASK) amplitude modulations can be implemented using simple circuits. However, noise affects both of them and they waste energy during transmission. Figure 3 shows example circuits for amplitude modulator and demodulator [20]. Note Figure 3 a) is one possible implementation of the Modulator block in Figure 2.

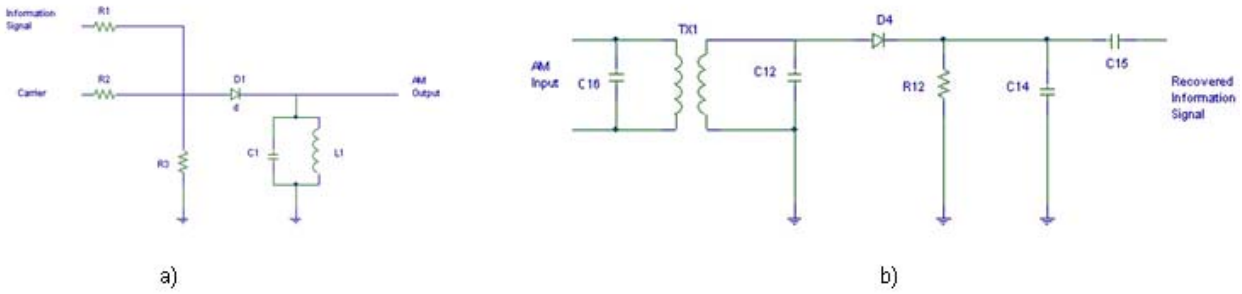

Figure 3. AM circuits: a) modulator, b) demodulator.

b. Angle Modulations

Angle modulations are not Linear Functions and they record the information in the angle of the carrier wave. Some examples are Frequency Modulation (FM), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK). Angle modulations require more complex circuits than amplitude modulations, but they present higher immunity to noise. Figure 4 displays examples of FM Modulator and Demodulator [20]. Note circuit complexity increases compared to AM circuits. The purpose of showing Figure 3 and Figure 4 is merely illustrative, since modern modulators and demodulators use integrated circuits. However, in general as the complexity in the modulation scheme increases, so does the hardware complexity.

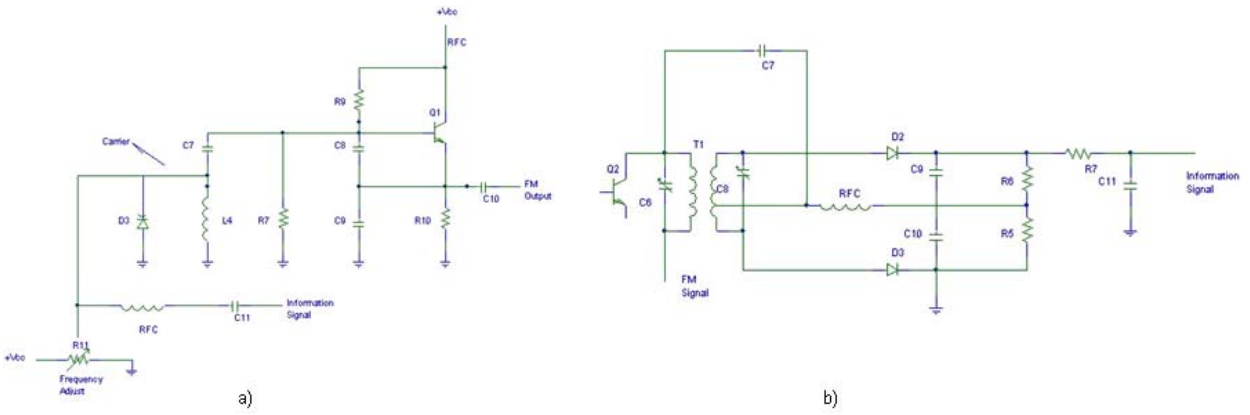

Figure 4. FM Circuits. a) modulator, b) demodulator: Foster-Seeley discriminator [20].

One phenomenon present in angle modulations as FM and FSK is Capture Effect, explained in the next section.

c. Capture Effect

When two signals arrive at a receiver simultaneously, the signal with the highest power level is received and the other one is attenuated and rejected without interfering with the received signal [21]. No collisions occur because the receiver detects all the information in the strongest signal. Therefore, a packet dropping process may serve as a model for the capture effect and it can constructively deliver data in wireless systems using FSK modulation schemes. Additionally, FSK is often employed in WSNs [8] because this modulation scheme is easily implemented with energy efficient hardware.

Previous work in the field of WSNs employed capture effect to detect and recover messages from packet collisions using a CSMA MAC protocol [22], but impact of the phenomenon in the effectiveness of a routing protocol was not characterized. Studies of routing protocols for ad hoc networks included capture effect as a characteristic in the experiments (e.g. [23], [24] and [25]), however the influence of capture effect on protocol performance has not been characterized either.

802.11 wireless networks have been studied extensively to characterize the capture effect in wireless packet data systems. Studies include an analytical model for evaluating the performance of CSMA/CA protocols used in channels with multipath fading, shadowing and capture effect [26], a model for the throughput of one 802.11b network considering capture effect [27] and the throughput of one 802.11 Access Point (AP) in the presence of hidden terminals and capture effect [28]. However, these studies considered one AP and transmitting nodes all within its radio range. There is no multi-hop environment so routing protocols are not required nor included in the studies.

Capture effect influence in linear modulations such as AM and ASK is unlikely because in case of simultaneous receptions of signals, the reception hardware adds the different amplitudes (linear function behavior), thus the receiver does not decode any of the original signals. The situation may result in amplitudes not suitable to the receiver circuits or in signals different from the ones transmitted. One example presented in [29] uses On-Off Keying, a version of ASK using two amplitude values of signal strength to discriminate "1" from "0". The work demonstrates when a node listens to two synchronized signals simultaneously, the radio receiver performs a logical "OR" operation over the two original signals.

2.1.3 Energy Usage Models

The OSI model for communications did not consider energy use as a physical layer characteristic [15] under the implicit assumption that communication devices had an unlimited energy supply. The situation with WSNs is different since main assumptions are nodes must work unattended and they depend upon a limited source of energy, such as batteries, which are not easy to recharge or to change [8]. As a result, energy use is a major concern since dead nodes limit the

useful lifetime of the network, thus the communication process should spend the minimum energy possible. Research in communication protocols for WSNs usually estimates energy use as a comparison metric between different protocols. The estimation typically includes simulation or analytical expressions based upon an energy use model. Two classes of models for energy use by sensor circuits are Analytical models and Measurement-based models. Different studies in the literature (e.g. S-MAC[4] and LEACH [6]) consider the radio circuits as the largest energy expender of the system, thus neglecting other sources of energy use.

a. Analytical models

Table 1 lists the parameters of one often cited energy use model, which reflects a low energy use radio, slightly better than Bluetooth [6].

Table 1. **Radio characteristics.**

Operation	Energy Dissipated
Transmitter Electronics ($E_{Tx\text{-}elec}$) Receiver Electronics ($E_{Rx\text{-}elec}$) ($E_{Tx\text{-}elec} = E_{Rx\text{-}elec} = E_{elec}$)	50 nJ/bit
Transmit Amplifier (ϵ_{amp})	100 pJ/bit/m^2
Idle (E_{idle})	40 nJ/bit
Sleep	0

However, it is unknown if the parameters for this model are realizable in a physical circuit. Additionally, the model implicitly uses a d^2 path loss model. Depending on the frequency a d^{α} ($\alpha > 2$) path loss model may be required [30]. Also, the communication system design should include path loss either explicitly as a random variable or through a shadow fading margin [18]. When computing energy use of a sensor node, the CPU and the sensing circuits also consume energy at a rate the design may or may not neglect, depending on the nature of the application. Therefore, a designer must use the model of energy use by the radio in conjunction with a model of energy use for the other elements.

One reported energy use model of the μAMPS sensor includes parameters for the StrongARM SA 1110 microprocessor, which can work with frequencies from 50 to 206 Mhz [31]. The model includes energy used by CPU, energy lost due to leakage and average radio use. The radio uses a Bluetooth compatible transceiver at 2.4 Ghz, with maximum data rate of 1 Mbps. Table 2 summarizes the model, which specifies the energy use rate in units of time, rather than in units of bits [13].

Table 2. μAMPS energy use model.

State	SA-1110	Sensor,A/D	Radio	Pk (mW)
Active	Active	Sense	tx/rx	1040
Ready	Idle	Sense	Rx	400
Monitor	Sleep	Sense	Rx	270
Observe	Sleep	Sense	Off	200
Deep Sleep	Sleep	Off	Off	10

Another platform specific model exists for the Crossbow Mica2 motes, using an ATMEL Atmega 128L microcontroller and Binary Frequency Shift Keying (BFSK) modulation with up to 38.4 kbps. In this case, energy use can be estimated based on hardware datasheets and time spent by the mote completing primitive operations [12]. Table 3 presents the main characteristics for this model, which specifies the current use rate for selected operations [12].

Table 3. Mica2 current use model.

Operation	Time (s)	I (mA)
Initialize radio	350E-06	6
Turn on radio	1.50E-03	1
Switch to RX/TX	250E-06	15
Time to sample radio	350E-06	15
Evaluate radio sample	100E-06	6
Receive 1 byte	416E-06	15
Transmit 1 byte	416E-06	20
Sample sensors	1.1	20

The Mica2 platform consumes energy at a slower rate than the μAMPs model. However, note both platforms have different hardware with different CPUs and different radios, thus the comparison may not be straightforward. Table 4 shows current use in two devices with same CPU but different radios; the MICAz radio is 802.15.4 compliant, working at 250 kbps, 2.4 Ghz and MICA2 uses BFSK with Manchester Encoding at 38.4 kbps and 915 Mhz [32].

Table 4. Current requirements for two Crossbow platforms.

Operating current (mA)	MICAz (Zigbee)	MICA2
Microcontroller ATMega128L, full operation	12 (@ 7.37 Mhz)	12 (@ 7.37 Mhz)
Microcontroller ATMega128L, sleep	0.01	0.01
Radio, receiving	19.7	7
Radio, transmitting (1mW power tx)	17	10
Radio, sleep	0.001	0.001
Serial flash memory write	15	15
Serial flash memory read	4	4
Serial flash memory sleep	0.002	0.002

Table 4 gives a better example of influence of radio design on current use. The previous three models use expected energy use from hardware datasheets, thus the models do not consider variances in real electronic circuits, which may affect these values.

b. Measurement models

One advantage of this approach is it includes actual electronic circuit performance, which may differ than the one expected in manufacturers data sheets. However, measurement models also include errors generated by the available test instruments and by the human operator performing the experiments. The following studies performed measurements to obtain energy use models that reflect the behavior of physically realizable circuits. One study measured the average current use rate with a multimeter [33]. Another study refined these measurements by employing an

oscilloscope to determine energy use in each of several states [34]. However, the researchers performed these measurements for only short time intervals. Results from [34] are shown in Table 5 and they give an idea of which activities generate more energy use for Mica2 platform during actual work.

Table 5. Mica2 measurement model.

Mode	Current	Mode	Current
CPU		**Radio**	
Active	8.0 mA	Rx	7.0 mA
Idle	3.2 mA	Tx (-20 dBm)	3.7 mA
ADC Noise Reduce	1.0 mA	Tx (-19 dBm)	5.2 mA
Power-down	103 µA	Tx (-15 dBm)	5.4 mA
Power-save	110 µA	Tx (-8 dBm)	6.5 mA
Standby	216 µA	Tx (-5 dBm)	7.1 mA
Extended Standby	223 µA	Tx (0 dBm)	8.5 mA
Internal Oscillator	0.93 mA	Tx (+4 dBm)	11.6 mA
LEDs	2.2 mA	Tx (+6 dBm)	13.8 mA
Sensor board	0.7 mA	Tx (+8 dBm)	17.4 mA
EEPROM access		Tx (+10 dBm)	21.5 mA
Read	6.2 mA		
Read Time	565 µs		
Write	18.4 mA		
Write Time	12.9 ms		

2.1.4 Energy Harvesting

Given the tight constraints in energy use for WSNs, researchers have proposed different alternatives to harvest energy from the environment in order to power sensor nodes or to help recharge the batteries and extend lifetime. Studies have addressed different energy sources and this section presents examples of energy harvesting systems.

One example of solar energy harvesting is using a module to obtain the most energy possible out of a solar panel under variable light conditions. The module improves energy efficiency from the solar panel and consumes 1mW [35]. As an example comparison, consider

information from Table 5: Crossbow's Mica 2 platform consumes 11.6 mA to transmit at 4 dBm, or equivalently, 0.13 mW; thus, employed during same time, the module to improve energy harvesting consumes 10 times more energy than the radio.

The study in [36] designed a system to obtain energy from vibrations in the body of the sensor node. The system employs a micro power generator built with Microelectromechanical systems (MEMS) which produces electrical energy when vibrating around its resonating frequency (204 Hertz in this study) via a piezoelectric transducer. The node can be located close to an engine and is expected to constantly generate around 1 micro Watt to recharge the battery.

Another alternative is energy harvesting from RF signals. Researchers have studied methods to utilize Gigahertz frequencies, or even Megahertz frequencies as in Radio-frequency identification (RFID) technologies, which mostly implement their own RF transmitter to provide the signals [37]. However, a different approach considers harvesting energy from already existent RF sources, such as radio or television stations. One example of this approach is a method to harvest energy from one AM local station working at 1584 kilo Hertz. Authors claim they obtained energy from the radio station transmitter located several kilometers away, showing a maximum current of 8 μA [37]. Nonetheless, the study used an L antenna of 8 m whose size may be a problem for small sensor nodes.

Energy harvesting could extend lifetime of WSNs. However, no matter what source of energy the nodes use, all processes in the network must achieve the minimum energy use possible. Data communication is a vital part of WSNs and there is still work to be done in order to decrease energy use while fulfilling all application requirements.

2.1.5 Summary

Section 2.1 explained physical layer concepts influencing design of communication protocols working in harsh conditions created by the wireless medium. Path loss and fading affect WSNs, thus hardware design of transmitters and receivers should use modulation schemes complex enough to overcome these problems but simple enough to allow implementation with simple circuits and minimal energy use.

2.2 DATA LINK LAYER

The Open Systems Interconnection (OSI) reference model defines the Data Link layer to allow transmission from one source to the destination through one hop. Error control, frame delimiting, sequence and flow control are among the functions of the Data Link layer [38]. IEEE standard 802 defined the MAC sublayer as a part of the Data Link Layer. Among the main functions of MAC sublayer are error control, arbitrate access to a shared channel and addressing [39].

2.2.1 Error Control

Data Link layer divides data in units called Frames. Frames frequently present errors due to channel conditions which reflect in changing the bits from "1" to "0" and vice versa [15]. Error control coding transforms the data to compensate for these changes. There exist four ways of implementing Error Control Coding [18]:

Waveform Coding maps k bits to 2^k waveforms (also called "chips"). E.g. for transmitting 2 bits, the source sends 4 (2^2) waveforms, one for each possible combination of the 2 bits. The approach generates a bandwidth expansion of 2^k / k. The total overhead is (2^k – k) bits.

Block Coding treats data as blocks of k bits, performs an operation on them and generates an n bit codeword. Block codes add n-k redundant bits to the original data, thus the overhead is (n-k)/n. The coding rate indicates the efficiency of the block code and is given by k/n. A higher coding rate indicates a more efficient block code. One example is Cyclic Redundancy Code-16 (or CRC-16) where n-k=16. If one frame has 8 bytes of data, the coding rate is 64/80=4/5.

Convolutional Coding transforms a continuous data stream into a continuous encoded stream, using a constraint length C. A convolutional coder generates an n-bit codeword utilizing the current k data bits and the previous C-1 blocks of k input bits. As an example, the convolutional coder used in 802.11a has coding rate 1/2 and constraint length 7. Convolutional codes are more powerful than block codes but they need bigger redundancy.

Turbo Codes use convolutional codes with fixed length input, which makes them linear block codes. Turbo codes add redundancy to the original message in two ways simultaneously: the first way is encoding the original bits; the second way is taking the same original bits, changing their positions in the block (process called Interleaving) and then encoding them.

Sections a and b describe two main approaches to compensate for errors created by the channel.

a. Error Detection

Error detection techniques utilize Block codes. The transmitter operates on the k data bits, appends redundant bits and sends the frame. The receiver performs the same operation in the received frame and the result should match the redundant bits. If the result does not match the redundant bits received, there are one or more bits received with errors. The number of errors that can be detected depends upon the Hamming distance, which is the minimum number of different bits between two valid bit sequences. To detect e errors, a code must have a Hamming Distance d = e + 1. Examples of Error Detection methods are Parity bits (which check if the number of "1"s transmitted is even or odd) and Cyclic Redundant Codes (CRC) which allow detection of errors in more than one bit. Error detection codes increase overhead in the frame because of the additional bits transmitted which also increase energy use. Some examples include CRC-32 (adding 32 bits per frame, as in 802.11) [15] and the previously mentioned CRC-16, used in Crossbow Motes [40]. Error detection schemes only allow the receiver to know there is a problem with the frame. The receiver may choose to discard the frame and do nothing or ask for frame retransmission, depending on the Link Layer protocol used [15]. One example is Basic CSMA, which discards frames with errors. Forward error correction shows a different approach.

b. Forward Error Correction

The second alternative is adding bits to the data frame such that in case of bit errors, the receiver can rebuild the information sent eliminating the need for retransmissions. The name for this strategy is Forward Error Correction (FEC) and the implementation may use Block, Convolutional or Turbo coding. To correct e errors the scheme needs a Hamming distance d = 2e + 1 [15], which gives an idea of the differences in complexity and overhead generated by error

correction compared to error detection. Error detection in general is simple but asking for retransmission generates more energy use. Error correction schemes are usually more complex but they effectively improve the Signal to Noise Ratio helping to overcome fading channel conditions. Authors in [41] show analytically that using no error control is more energy efficient than using CRC and retransmission, which is expected since there are less transmissions involved. The same study shows error correction schemes implemented with block codes such as BCH (Bose-Chaudhuri-Hocquenghem) codes are more energy efficient than using no error control for packet sizes larger than 500 bits, which may be high for WSNs. Authors also tested BCH codes vs. Convolutional codes. Results showed BCH codes are more energy efficient with packet sizes of 500 bits or more. The study assumed independent errors, Binary Orthogonal Non Coherent FSK and Rayleigh fading, with distances between nodes of 20-30 meters and BER in the order of 10^{-4}. Adding the effect of carrier frequency and using other FSK modulation schemes may improve the model.

2.2.2 Access to a Shared Medium

MAC protocols manage medium access when different nodes have to use the same physical link. Protocols designed for WSNs must be energy efficient to maximize lifetime, scalable according to the network size, and robust to changes in the network such as addition of new nodes and death of existing nodes [42]. MAC protocols developed for WSNs may be grouped into two main approaches: Scheduled-Based and Contention-Based [43]. Since the research in MAC protocols for WSNs is vast, this chapter does not present a comprehensive list of protocols but rather examples of protocols that summarize the major concepts present in the literature.

a. Schedule-Based Protocols

Schedule-Based protocols regulate medium access by defining an order or Schedule for nodes to transmit, receive or be inactive and they can be useful for deterministic topologies and for applications with strict delay requirements. The following are examples of this approach.

Low-Energy Adaptive Clustering Hierarchy (LEACH) [19] includes application, routing, MAC and physical characteristics for communication in WSNs. The specific application considered is remote monitoring where data gathered by neighboring nodes may be redundant. The scheme assumes all nodes are synchronized, they can control their transmission power and they can reach one Base Station (BS, the final destination node) if they must. The nodes also have sufficient processing capabilities to implement different MAC protocols and perform signal processing functions, thus they can aggregate all information in only one message. LEACH works in rounds [19] presented in Figure 5: during Setup phase, all nodes organize in clusters and elect a Cluster Head (CH) using CSMA. During Frames (labeled with F in Figure 5), nodes send information according to the schedule and they can sleep the rest of the time. Ni are time slots assigned to node i.

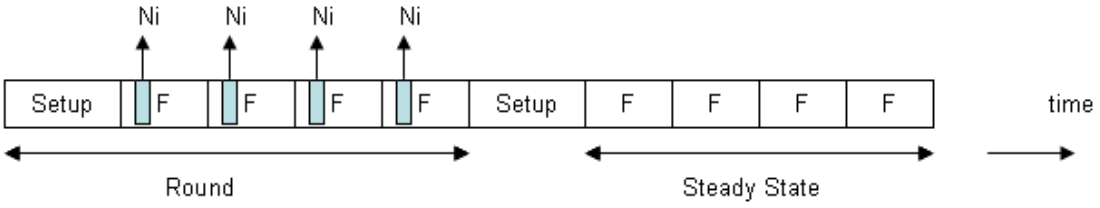

Figure 5. LEACH operation rounds.

The CH receives information from all the nodes, aggregates it and sends it to the BS. The protocol elects a new CH every round. Some advantages of LEACH are sleeping saves energy; CH rotation extends the lifetime of the network because energy use spreads to all nodes, so it

should take longer to drain their energy resources; including several layers in protocol design benefits the whole communication scheme. However, when a CH uses all its energy, the whole cluster becomes inactive during the whole steady-state phase, even if several nodes inside the cluster have enough energy to function. LEACH assumes one hop communication between the nodes and the CH and among the cluster heads and the BS, which may not be easy to achieve when random deployment is required. There are no documented hardware implementations of LEACH, so the scheme may require adjustments for actual operation. LEACH requires tight synchronization (for the TDMA schedule and for using DSSS) which may not be easy to accomplish. The PEDAMACS protocol improves on synchronization methods and number of hops, at the cost of increased complexity.

Power Efficient and Delay Aware Medium Access Protocol (PEDAMACS) [7] assumes one Access Point (AP) with the ability to reach all nodes in one hop. The nodes may need one or more hops to reach the AP. There are three transmission power levels defined to reach three distances: Pl the maximum, Pm the medium and Ps the minimum. The protocol has four phases, summarized in Figure 6, where solid arrows are packets sent by nodes and dashed arrows show packets sent by the AP. During topology learning phase, all nodes use a MAC scheme similar to 802.11. The AP broadcasts a packet with Pl to synchronize the nodes. Then, the AP sends another packet with Pm so all nodes receive the topology currently held by the AP and can update it. Each node identifies its local neighbors (nodes able to decode one packet transmitted with Ps), its interferers (nodes unable to decode one packet transmitted with Ps, but receiving the signal with enough strength to interfere with other signal) and its parent node in the route to the AP. The topology collection phase allows each node to send topology information to the AP

using Ps, so data may possibly go through several hops. The protocol also uses a CSMA scheme in this phase.

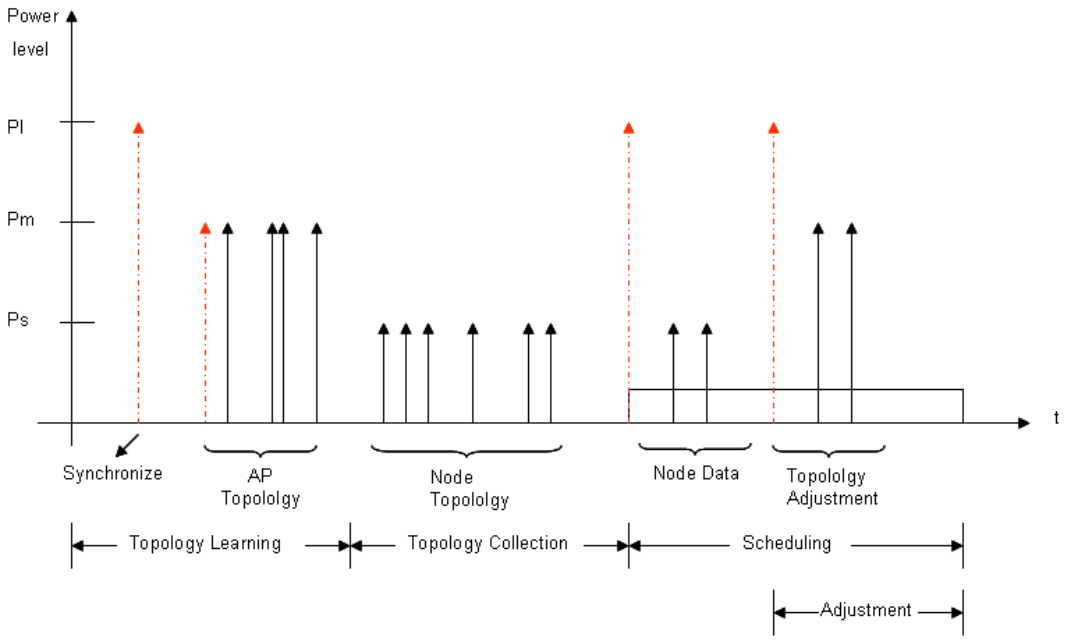

Figure 6. PEDAMACS phases.

Scheduling is the next phase and in it the AP broadcasts the schedule so every node adjusts its clock and knows the time slots allowed for it to transmit and receive. The rest of the time, the nodes should be sleeping. Nodes transmit data with Ps. The last phase is Adjustment, located at the end of the Scheduling phase. Here, the AP requests and the nodes send adjustment topology packets with changes in the neighbors or interferers. Nodes can also send this information during the scheduling phase inside data packets [7].

Advantages of PEDAMACS include it considers characteristics from physical and network layers; the protocol can be used for sending periodic data or for event-driven sensing, using the assigned time slots only when the event happens; otherwise, the nodes keep on sleeping. The protocol can be extended to use more than one AP and to handle nodes outside the range of the AP. Delay results are bounded for different network sizes [7]. However, there are no hardware implementations yet, thus there may be problems not encountered in the simulation.

PEDAMACS has additional overhead beside RTS, CTS and ACK packets. The protocol assumes an AP with infinite energy, which may not be possible in WSNs, especially with random deployment. Additionally, low transmission power levels save energy, but radio ranges decrease significantly. One example with Mica2 motes shows 25 cm radio range for -20 dBm which is the minimum transmission power [40], so nodes must be very close to each other to maintain connectivity in the network. The TRAMA protocol uses another approach where no direct communication with a central entity is required.

Traffic-Adaptive Medium Access Protocol (TRAMA) [43] scheme assumes the time is slotted and all nodes are synchronized. There are Transmission slots for sending data or schedule information with no collisions, and Signaling slots, which are contention-based and are typically smaller than Transmission slots. The nodes can be in any of three states: Transmission, Reception and Sleep. TRAMA has three components:

Neighbor Protocol (NP): sends one-hop neighbor information to closest nodes using Signaling slots. Every node stores this data conveying, in fact, information from two-hops away. New nodes can join the network in this period.

Schedule Exchange Protocol (SEP): all nodes send the list of their receivers using Schedule Packets in Transmission slots, before starting their data transmissions. A bitmap is used instead of explicitly provide all the receivers addresses.

Adaptive Election Algorithm (AEA): using information from SEP and NP, AEA selects transmitters and receivers and assigns priorities so nodes will communicate without collisions [43]. Figure 7 shows one example communication using TRAMA. Note Sw is a guarding time between Transmission and Signalling.

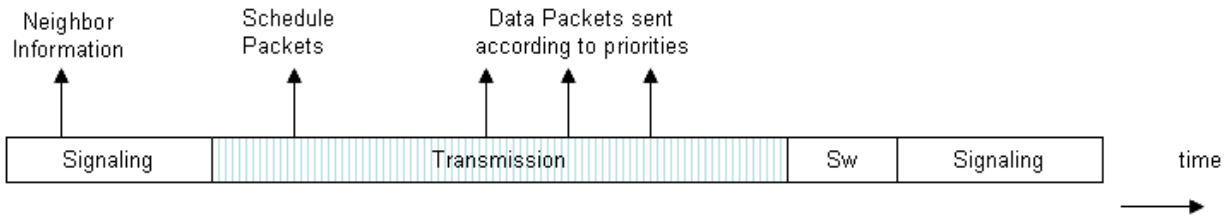

Figure 7. Packets sent in different time slots in TRAMA.

Advantages include TRAMA can adapt the schedules to the application (monitoring or event driven) and time slot managing is distributed (no central entity is required). Disadvantages of TRAMA include the protocol assumes synchronization is achieved and this is not an easy task. Control packets and control fields sent in data packets create overhead. The protocol generates processing overhead to assign and manage the time slots. Energy use comparison does not consider all the overhead packets in TRAMA. The protocol has not yet been implemented in hardware so there can be more issues not found during the simulations. The NATP protocol considers a different approach.

Sensors using Neighbor Aware Probabilistic Transmission Protocol (NATP) [44] monitor periodic events and transmit their information towards one Base Station (BS), possibly through several hops. NATP assumes equal time slots for the whole network with synchronization provided by the BS through beacons. Every node has a unique ID; all sensors can transmit or receive within a limited range using a common carrier frequency; sensors can be in transmit, receive or idle (radio off) mode; there is a predetermined route to reach the BS. Collisions may occur when different transmitters send data simultaneously to one receiver. However, the study considers capture effect so it is possible to receive one of the colliding packets. Figure 8 illustrates a communication session using NATP where solid lines show node packets, numbers indicate node ID and dashed lines are periodic events. During network deployment, NATP uses flooding to exchange data and to create a path to the BS. Collisions may happen since nodes do not have a schedule yet. Each node calculates the number of hops to the BS and they receive

information about their two-hop neighbors (through overhearing), which allows them to create a table including neighbors with their same hop count to the BS also plus or minus one and two hop counts. If this table is large, the node is in a dense area, so the collision probability may be high.

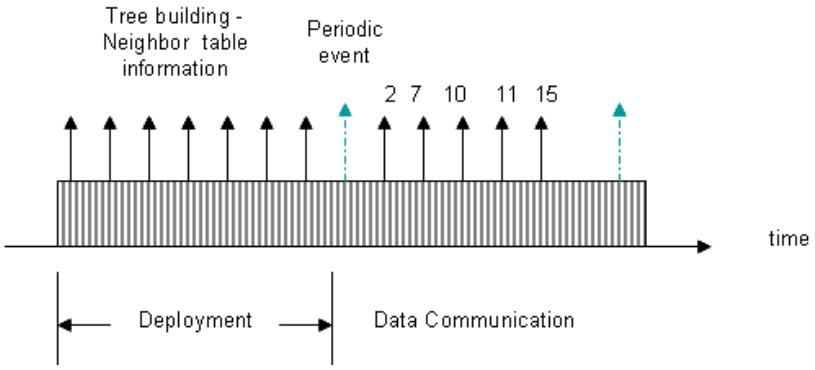

Figure 8. Communication example using NATP.

To decrease the chance of collisions, nodes assign transmission time slots in decentralized fashion using the neighboring node IDs. The node with the lowest ID transmits in the first time slot after the monitored event occurred, and the rest of the nodes will wait for following time slots, in ascending order [44].

Advantages of NATP are time slot assignment does not require a central entity and the scheme considers characteristics from several communication layers (routing, MAC and physical), which may benefit communication process. However, synchronization is assumed by periodically generating beacons, thus all nodes should listen to them in order to maintain the time slots, consuming energy and thus affecting protocol performance. In addition, there is not an explicit way for updating the routes whenever necessary and there are no documented hardware implementations yet.

b. Contention-Based Protocols

Contention-Based protocols do not require central coordination, so they may be more appropriate for random deployment applications with no stringent delay requirements. However, they use

energy during periods of "Idle listening", which occurs when nodes are listening to the medium and there are no transmissions, wasting energy [4]. The following are examples of this category of MAC protocols.

IEEE 802.11 is a family of standards for Wireless data communications with definitions for characteristics in the Physical and MAC layers. Many of the concepts and mechanisms introduced in the standard form the basis of contention-based protocols for wireless networks in general and WSNs in particular. This section reviews the main concepts adopted by researchers in the field of WSN. The MAC protocol defined in the standard allows two modes [45]: The first mode is DCF: Distributed Coordination Function: mode with no central device controlling the communication. DCF uses CSMA/CA in any of these ways:

Carrier Sensing: The node simply senses the medium. If it is idle, the node transmits the whole frame. If the medium is busy, the node waits until it becomes idle again, waits for a random time and transmits. Upon frame reception, the receiver node answers with an ACK (Acknowledgement) control frame. If a collision occurs, transmitting nodes wait a random time and try again later.

Virtual Carrier Sensing: the node with a frame to transmit senses the medium. If it is idle, the node sends a control frame called RTS (Request To Send), which contains the intended receiver address and the time required to send the information (transmission delay). If the destination node agrees to communicate, it will answer with a CTS (Clear To Send) control frame which also contains the delay. All nodes hearing RTS or CTS should refrain from transmission until two conditions are fulfilled: transmission delay has elapsed and the medium is idle again. The receiver must answer with an ACK to each data frame received.

The second mode allowed in the standard is PCF: Point Coordination Function: A special node, The Access Point (AP), polls every node and controls all communication process. AP periodically broadcasts a beacon control frame with systems parameters and invitations to join the network

Another contention-based protocol is Sensor MAC (S-MAC) [4], which operates in a similar way to 802.11 in DCF: Transmitter and receiver exchange RTS, CTS and ACK frames in order to send data. Additionally, nodes in S-MAC go to sleep and wake up following a schedule given by a SYNC frame. Control frames generate overhead. Figure 9 shows an example communication between four nodes using S-MAC. Nodes A, B and C are within range of each other. D is within range of C and A transmits to B. Note D cannot listen to the conversation between A and B, but its radio is on anyway, showing an example of Idle Listening.

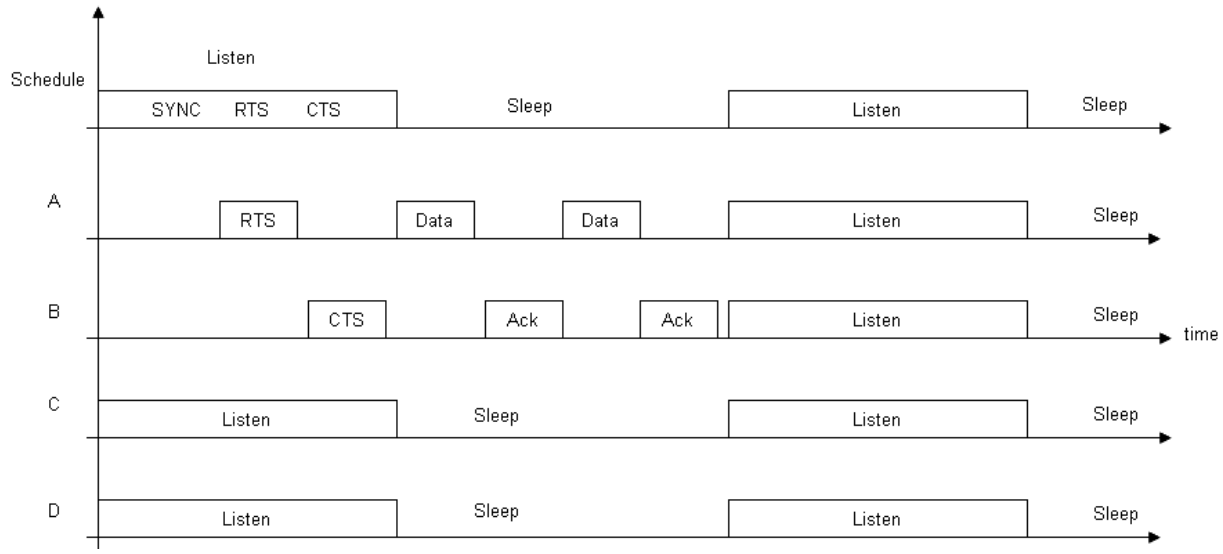

Figure 9. S-MAC example.

Timeout MAC (T-MAC) T-MAC [5] improves on S-MAC energy use following the same basic idea: using a schedule for sleeping and waking up. However, T-MAC makes nodes sleep earlier during the schedule if there are no activation events, such as the node needing to

send information or hearing activity in the channel. As in S-MAC case, RTS, CTS, ACK and SYNC frames generate overhead. Figure 10 shows one example communication with T-MAC. Note the time labeled as “Contend” in the figure is the random time node A waited before transmitting inside the Contention Interval. Node B is within range of A, C is within range of B and D is in range of C. A sends to B and D can sleep earlier since it does not listen to A or B.

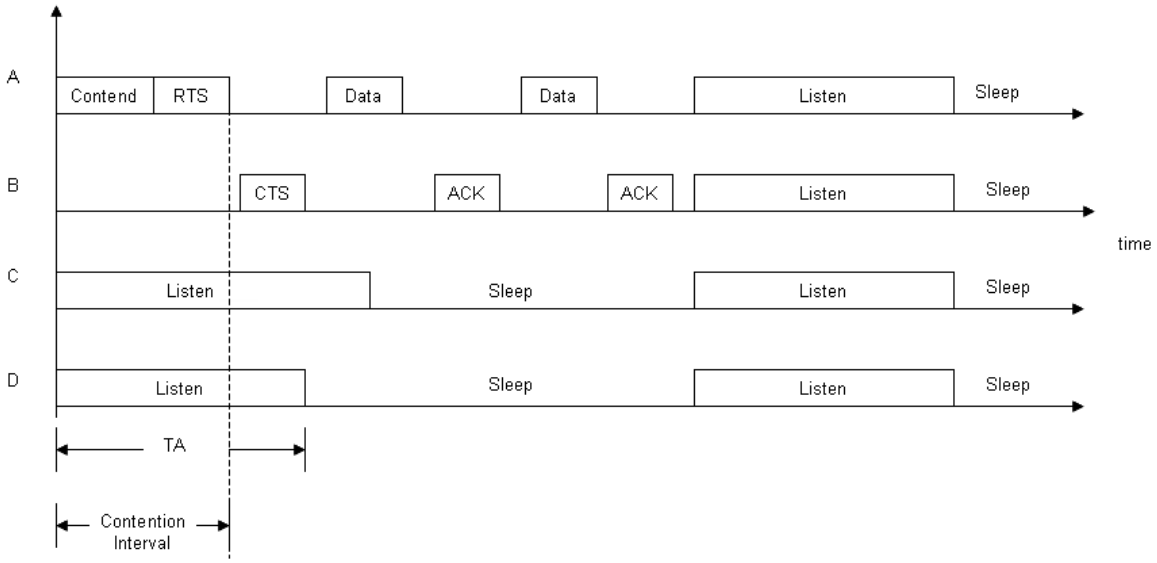

Figure 10. T-MAC example.

Berkeley Media Access Control for Low Power Sensor Networks (B-MAC) is a protocol with no control frame overhead [12]. B-MAC uses a long preamble in data frames and nodes verify the medium periodically, with a period equal to the preamble size. When they are not verifying the medium, nodes go to sleep. However, the preamble itself creates overhead to ensure nodes will check the medium in the proper time. One example is using 271 bytes for sending 36 bytes of data [12]. Figure 11 presents one example transmission using B-MAC where all nodes are within range of each other.

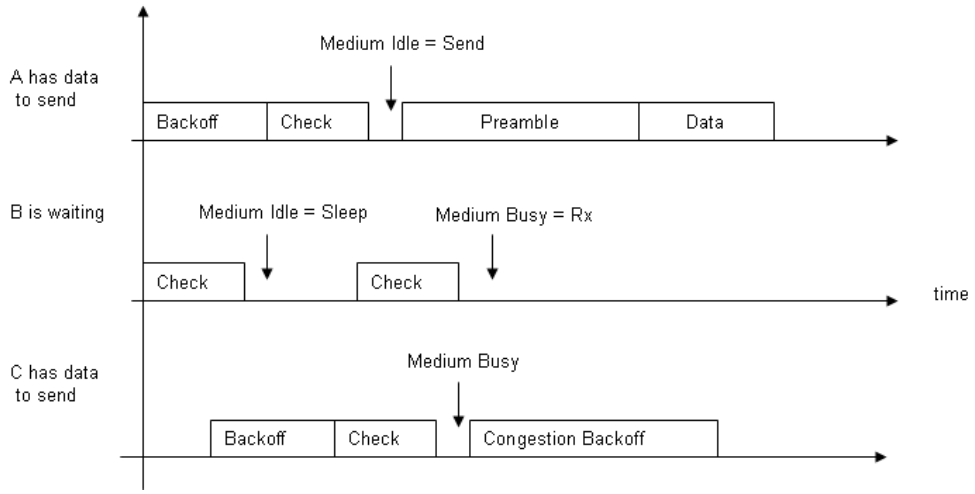

Figure 11. B-MAC communication example.

Uncertainty Driven MAC protocol (UBMAC) [46] reduces preamble size from B-MAC by estimating clock uncertainty using Rate Adaptive Time Synchronization (RATS). RATS exchanges frames with timestamps between neighbors and computes clock uncertainty within an error boundary, allowing smaller preambles when used over B-MAC. On the other hand, the protocol requires a learning phase generating additional overhead besides timestamp frames.

Sift [9] is a CSMA-type of protocol which uses a non-uniform probability distribution for selecting the backoff waiting time. S-MAC, B-MAC and 802.11 use Binary Exponential Backoff with uniform probability for selecting the backoff time. Sift has significant smaller delay than 802.11 when several sources are sending data in the same zone of the network. The protocol uses RTS, CTS and ACK frames when the packet size is big, but there is no control frame overhead for small packets. However, there is no prevision for turning off the radio, so Idle Listening problem occurs.

2.2.3 Addressing

Addresses differentiate nodes in the network. IEEE 802 standard defines 6-byte individual and group addresses, so every device connected to a network should have a unique identifier in the world (at the MAC level), and different devices may belong to one set identified with a group address [39]. Addressing in WSNs is different, because of all hardware and energy constraints. As an example, consider temperature monitoring, which may code information with as low as one byte; using 6-byte addresses creates considerable overhead. 2-byte addresses allow identification of 65535 nodes, so they may be adequate for WSNs. Even 1-byte addresses may be enough to identify nodes in one zone of the network and there may be no need to have unique addresses all over the network, if nodes do not communicate with all other nodes (as happens in cluster-based approaches, such as LEACH [19]). Network layer addresses for WSNs may use attribute-based naming or location based naming. Using the same example of temperature monitoring application, it may be more interesting to find zones where the temperature goes above a threshold and not the temperature measured by a particular node [1].

2.2.4 Summary

Section 2.2 presented concepts from Data Link layer and MAC sublayer, such as error control schemes, addressing and main categorization of MAC protocols for WSNs, showing their advantages and disadvantages especially in energy use.

Table 6 shows a list of features generating energy use for the protocols presented in this section. Note the table does not show units or quantitative values. The studies tested different protocols using diverse network sizes, topologies, energy use models and different metrics, thus

it is difficult to compare all protocols to define which one and under what circumstances has better performance. That is the reason of presenting just features in the table and not numeric data. Retx+ means nodes should retransmit the information increasing energy use. Synch Req means Synchronization requirements for the protocol. Sift keeps the radio on all the time wasting energy if no information is sent.

Table 6. Energy use comparison for MAC protocols.

Protocol	Control Frame Overhead	Header bytes	Additional Overhead	Synch Req	Idle Listening?	Collision impact	Node states
LEACH	ADV, Join-Req, Schedule	No	Synch.	Tight	No	Retx+	Tx, Rx, Off
PEDAMACS	RTS, CTS, ACK, Synch, Topology learning	No	No	Tight	Yes (guard band)	Retx+	Tx, Rx, Off
TRAMA	Neighbor info, Schedule	No	Synch.	Tight	Yes (signaling slots)	Retx+	Tx, Rx, Off
NAPT	Beacon, Neighbor info	No	No	Tight	Yes	Retx+	Tx, Rx, Off(Idle)
S-MAC	RTS, CTS, ACK, SYNC	No	No	Loose	Yes	Retx +	Tx, Rx, Off
T-MAC	SYNC, RTS, CTS, ACK, FRTS, DS	No	No	Loose	Yes	Retx +	Tx, Rx, Off
B-MAC	None	Preamble	No	None	Yes	None	Tx, Rx, Off
UBMAC	Synch packets	Preamble	Learning phase	None	Yes	None	Tx, Rx, Off
Sift	RTS, CTS, ACK, SYNC	No	No	None	Yes	Retx+	Tx, Rx

In order to develop a quantitative analysis, all protocols must use the same scenario and communication conditions. Table 7 details the communication process of a network with four nodes, all within range of each other, where one node (other than a CH or BS) sends one data

packet. The last column shows the number of overhead packets generated. The analysis assumes the best conditions for each protocol (e.g. no collisions or interference). Note N/A underestimates the number of packets required for executing a particular operation, because the protocol does not define the exact method or conditions for it (as it is the case for Synchronization and learning phase). B-MAC case is different from the others since the preamble may have the length of one or more full additional packets; thus, as a best case for the protocol, the table considers just one overhead packet. The best case for Sift is when data packets are small and it has zero packet overhead.

Table 7. Quantitative overhead comparison for MAC protocols.

Protocol	Description	From	To	OH (number of Packets)
LEACH	Adv for CH election	One node	Network	1
	Join request	all other nodes	CH	3
	Schedule	CH	Network	1
	Synchronization			N/A
	Data packet	Node	CH	0
LEACH Total				5
PEDAMACS	Synchronization	AP	Network	1
	Topology from AP	AP	neighbor nodes	1
	Node Topology	neighbor nodes	AP	3
	Schedule	AP	Network	1
	adjustment request	AP	Network	1
	adjustment topology	neighbor nodes	AP	0
	Data packet	Node	AP	0
PEDAMACS Total				7
TRAMA	Neighbor information (NP)	each node	each node	4
	List of Receivers (SEP)	each node	each node	1
	Synchronization	Assumed	Assumed	N/A
	Data packet	Node	other node	0
TRAMA Total				5
NATP	Beacons	BS	Network	1
	Flooding	Network	Network	4
	Data packet	Node	Node	0
NATP Total				5
S-MAC	SYNC	Node	Network	1
	RTS, CTS, ACK	Node	Node	3
	Data packet	Node	Node	0
S-MAC Total				4
T-MAC	SYNC	Node	Network	1
	RTS, CTS, ACK	Node	Node	3
	Data packet	Node	Node	0
T-MAC Total				4
B-MAC	Preamble in Data packet	Node	Node	1
B-MAC Total				1
UBMAC	Timestamps	Node	Node	4
	Learning phase	Network	Network	N/A
	Preamble in Data packet	Node	Node	1
UBMAC Total				5
Sift	None	Node	Node	0
	Data packet	Node	Node	0
Sift Total				0

2.3 ROUTING PROTOCOLS

Different classes of WSNs applications have diverse routing requirements, therefore a wide variety of routing protocols and protocol classifications exist. The following is one classification of routing protocols according to how the source finds a route to the destination [47]: Proactive: protocols compute all routes in the network beforehand. Proactive protocols may be useful when the designer defines the network topology in advance and it does not change frequently. Reactive: routes are calculated only when they are required. Reactive protocols are more relevant for WSNs since the topology is changing constantly due to node movement, environmental changes and death or disconnection of the nodes from the network. Hybrid: use a combination of both previous ideas.

Routing protocols optimize routes according to one metric as delay, number of hops or cost. A common approach for routing protocols in WSNs is selecting the routes that would generate minimum energy use when sending a packet through them. These routes are referred to as Energy Efficient Routes, not to be confused with Energy Efficient Routing Protocols, whose main goal is to minimize energy use in the whole network, not just in individual routes [47]. Just as research in MAC protocols, studies in routing protocols designed for WSNs is wide, and a comprehensive list of them is outside the scope of this document. Section 2.3.1 describes protocols that exemplify main concepts in the literature.

2.3.1 Network structure

Routing protocols may be also categorized according to the Network structure, as either Hierarchical Network Routing or Flat Network Routing Protocols [47]. Hierarchical Routing

schemes define nodes with special functions during the communication process, such as Cluster Heads. These nodes define when and which nodes communicate. Examples of hierarchical routing are LEACH [6] which generates overhead for cluster creation and for schedule propagation, PEGASIS [48] which creates overhead to know the energy status of nodes in order to adjust network topology and TEEN [49], which has overhead for cluster formation and additional packets sent to establish thresholds to transmit data. In each case, the overhead increases the rate of energy use. Hierarchical Routing protocols may be useful when network topology is deterministic, with few changes in the network and stringent delay requirements.

Flat Network Routing protocols assume all nodes play identical roles in the network and they are useful when the network has random topology, moving or stationary nodes and delay tolerant applications. The following protocols are examples in this category.

Flooding is a fundamental protocol where every packet received is retransmitted [15], Flooding exhibits undesirable behaviors for WSNs, such as duplicate packet generation, information overlapping when several nodes transmit the same information from one region, and resource blindness since Flooding uses bandwidth and energy regardless the constraints found in Wireless Sensor Networks [47].

Sensor Protocol for Information via Negotiation (SPIN) [11] solves some of the problems generated by Flooding, because nodes advertise the type of data they have through an ADV packet. If neighbor nodes are interested, they send REQ packets and the source sends data to them. These advertising packets constitute overhead.

Directed Diffusion implements queries of interests (sensing tasks) from the sink to all nodes in the network or nodes in a particular region [14]. A node receiving a query creates a gradient towards the neighbor transmitting the query and, in case one event happens, the node

sends the information to the proper neighbor, creating routes for data propagation. The protocol reinforces best paths to prevent further flooding, but there is still overhead to send and refresh the queries and to maintain the routes.

Minimum Cost Forwarding (MCFA) [13] computes the least cost from each node to the Base Station (BS), this way: the BS advertises its cost as zero (0) using an ADV packet; neighbors receive the advertisement and add the advertised cost to the cost of the link between them and the BS. Every neighbor retransmits its updated cost and the neighbors in the next hop follow the same procedure until all nodes in the network have their own cost to reach the BS. Every packet transmitted carries a required cost and the cost already spent since the packet left the source. When a node receives a packet, it checks if the cost so far added to the cost from itself to the BS is less than the required cost. If it is, the node is in the shortest path to the BS and the node retransmits the data; the same procedure repeats until the packet reaches the BS. Note computing and updating the minimum cost and the cost between neighbors generates overhead.

Dynamic Source Routing (DSR) [50] has two phases: Route Discovery and Route Maintenance. When a node needs to send data to a destination, it looks for it on a routing table. If the destination is not in the table, the source executes the first phase by sending a Route Request packet, which propagates through the network. The protocol adds the address of every node participating in the route to the packet header. When the destination receives the packet, it answers with a Route Reply packet containing the full route from source to destination. The Route Maintenance phase identifies unavailable links when no acknowledge is received for a packet just sent, so the protocol needs to use another route or start the Route Discovery phase again.

A Gossiping protocol requires that a node receiving a packet must retransmit it with a probability less than 1.0 [51], which improves upon Flooding performance because if the packet is not retransmitted, there is one less duplicate in the network. However, sensor nodes using Gossip waste energy receiving a packet if they do not retransmit it.

2.3.2 Summary

Section 2.3 introduced categorizations of routing protocols according to the procedure to select the routes (beforehand or on demand) and according to network structure. Table 8 presents a summary of the routing protocols presented in this section, emphasizing energy use tasks. Note Retx+ means nodes should retransmit the information increasing energy use. Dup. Elim- means the situation eliminates duplicated packets, decreasing energy use. Routing protocols presented in this section do not have synchronization requirements and they keep the radios on all the time wasting energy when there are no transmissions, which is a disadvantage for WSNs. As in MAC protocol case, all tests performed on these routing protocols use different network sizes, topologies, energy use models, evaluation metrics and units, making it difficult to decide which one is suitable for one particular application.

Table 8. Routing protocol energy use comparison.

Protocol	Control Frame Overhead	Header bytes	Additional Overhead	Idle Listening?	Collision impact	Node states
Flooding	None	None	Duplicate packets	Yes	Dup. Elim -	On all the time
SPIN	ADV, REQ	meta-data	No	Yes	Retx +	On all the time
Directed Diffusion	Periodical Interests broadcasts	Gradients	Geographical Info, Interest table creation & maintenance	Yes	Retx +	On all the time
Minimum Cost Forwarding	ADV	Required and spent costs	Packet rx not tx, Computing all costs	Yes	Retx +	On all the time
Dynamic Source Routing	RouteREQ, RouteREP, RouteError	Addresses of all nodes in the route	Routing table creation and maintenance	Yes	Retx +	On all time
Gossip	None	None	Duplicate packets, packet rx not tx	Yes	Dup. Elim -	On all the time

In spite of the differences presented in the studies, it is possible to develop a quantitative analysis to determine the overhead packets generated by each protocol. Table 9 shows the result of a four-node linear network where the range of each node is just enough to reach its next neighbor. A source located in one extreme of the network sends one data packet to the sink, which is located in the other extreme, so information must go through three hops. The analysis considers overhead packets only until the data packet reaches the sink and the best conditions for each protocol (e.g. no errors and no route updates).

Table 9. Quantitative overhead comparison for routing protocols.

Protocol	Description	From	To	OH (Number of packets)
Flooding	Data packet	Source	Neighbor1	0
	Data packet	Neighbor 1	Neighbor2	0
	Duplicate	Neighbor 1	Source	1
	Data packet	Neighbor 2	Sink	0
	Duplicate	Neighbor 2	Neighbor 1	1
	Duplicate	Source	Neighbor 1	1
Flooding Total				3
SPIN	ADV	source	all network	3
	REQ	sink	Source	3
	Data packet	Source	Sink	0
SPIN Total				6
Directed Diff	Interests	Sink	Source	3
	Data packet	Source	Sink	0
Directed Diffusion Total				3
MCFA	ADV	Sink	Network	3
	Data packet	Source	Sink	0
Minimum Cost Forwarding Total				3
Gossip	Data packet	Source	Neighbor1	0
	Data packet	Neighbor 1	Neighbor2	0
	Duplicate	Neighbor 1	Source	1
	Data packet	Neighbor 2	Sink	0
	Duplicate	Neighbor 2	Neighbor 1	1
Gossip Total				2
DSR	RouteRQ	Source	Network	3
	RouteREP	Sink	Source	3
	Data packet	Source	Sink	0
Dynamic Source Routing Total				6

Communication protocols presented in the literature in both MAC and routing layers generate overhead increasing energy use. Additionally, Contention-Based MAC and Flat Network Routing protocols still suffer from Idle Listening, a major source of energy wasting. Table 6 and Table 8 show examples of protocols affected by this situation. Schedule-Based protocols may improve on Idle Listening but they usually require node synchronization, something not easy to achieve in WSNs. Note complex protocols in both layers have limited

applicability in WSNs, since they are hard to implement in sensor nodes because of their memory and processing-power constraints.

2.4 CROSS-LAYER ANALYSIS AND DESIGN

Researchers in WSNs seldom design MAC and routing protocols to take advantage of characteristics of other communication layers, because of the traditional assumption behind layering: With proper interface design, one protocol in communication layer n can work with any combination of protocol in layers below and above. However, communication system design should consider several (if not all) communication layers in the OSI model sense, since not every protocol in a particular layer may correctly match the characteristics of protocols in other layers. The term Cross-Layer has different meanings in the literature. Sections 2.4.1 and 2.4.2 explain two approaches to Cross-Layer research in WSNs.

2.4.1 Inter-layer communication

The approach assumes Cross-Layer functions include information exchange from MAC layer activities to the routing protocol (or vice versa) to improve efficiency. The following are three examples of this approach. The work in [52] shows a routing protocol employing statistics generated by the MAC protocol to achieve energy conservation and fairness. On the other hand, the study does not include physical characteristics and it evaluates the method using six nodes. It is not clear if it will scale to larger network sizes such as the ones expected in WSNs. The study in [53] applies a fully distributed and self-organizing time-division multiple access (TDMA)

scheme, in which each active node periodically listens to the channel and broadcasts a short control message. The routing algorithm uses information obtained by the MAC protocol. Nonetheless, control messages create additional overhead. Routing-Enhanced Duty-Cycle MAC Protocol, RMAC [54], sends one packet through different hops in one S-MAC sleeping period, using routing information. However, the work assumes the routing protocol exists (without selecting a particular protocol), none of its overhead is considered and the method requires node synchronization.

2.4.2 Multiple Layer analysis

The approach studies one particular layer-n protocol in combination with certain protocols in n+1 or n-1 layers, as a MAC protocol in combination with other protocol either in Physical or in Network layer. There is no explicit communication across the layers. Some examples in this approach are:

The study in [55] employs a conventional routing algorithm for creating routes and then a MAC protocol creates a schedule across the routes. However, synchronization between the nodes must exist and the study does not specify a particular routing protocol. The work in [56] suggests employing Time Division Multiple Access (TDMA) over Ultra Wide Band (UWB) to keep track of costumers in a ski resort. The paper does not present quantitative results and it does not consider any particular routing protocol. Another example employs Binary Phase Shift Keying (BPSK) Code Division Multiple Access (CDMA) with different codes for WSN. Nevertheless, the study does not specify the routing protocol and the study assumes synchronization exists, thus underestimating the overhead required for generating strict synchronization needed for using different CDMA codes [57].

Research in [58] presents analysis of energy use using physical characteristics from the lower ISM band and protocol operations from three particular MAC protocols: S-MAC, non persistent-CSMA and nanoMAC. In addition, they compare energy use in a multihop case versus a single hop case. However, no specific application or routing protocol is considered and the energy cost of finding the route is not included in the analysis. Additionally, nanoMAC generates overhead by using RTS, CTS and ACK frames. Other example uses Linear Programming techniques to model tradeoffs between different performance metrics such as energy use, rate and lifetime. The research includes specific MAC and transport constraints [59]. However, there is not an apparent way of implementing practical algorithms using the analytical models presented. In addition, the authors mention their analysis uses proactive routing, but they do not include the overhead required to find the routes and proactive routing is not useful in applications with random deployment or when topology changes due to nodes ending their lifetime.

Cross-layer analysis and design is a step forward in communication protocols for WSNs. The studies show the communication process improves by including two or more layers, and different protocols may benefit by sharing information generated by other layers. However, none of the approaches (Inter-layer communication and Multiple Layer analysis) selects a MAC protocol with the criteria of complementing or taking advantage of the characteristics in the other layers. In other words, it is not clear if protocols selected for the studies are the best choice to improve system performance. Sections a and b present examples of different protocol combinations, to highlight the importance of proper protocol selection in a Communication System.

a. Examples of Inappropriate Protocol Combinations

Even if it is not related to WSNs, a typical non-appropriate combination is the case of data transmission using Transmission Control Protocol (TCP), protocol belonging to the transport layer, over a satellite link: delay can limit TCP's full utilization of the link, generating slow data transfer [60]. WSNs are also prone to protocol mismatches. Consider the fact that many MAC protocols turn off the radio to conserve energy (e.g. S-MAC [4], T-MAC [5], B-MAC [12]). A routing protocol working with any of these MAC protocols must find new routes very frequently, since topology changes not just when nodes use all their energy, but also when they are temporarily out of the network due to MAC functionality. One example is using Directed Diffusion [14] and B-MAC [12]: Directed Diffusion reinforces routes for sending the information, but nodes in that route may be sleeping due to B-MAC functionality, so the route to the destination must change, adding overhead to the routing protocol. Another example is the use of SPIN [11] and S-MAC [4]. One node may have data to transmit, so it sends an ADV message. If the interested node is off because of S-MAC functionality, it does not receive the data.

b. Examples of Appropriate Protocol Combinations

The field of WSNs may present adequate protocol combinations in different layers. For example routing protocols with no explicit route calculations used together with MAC layer protocols turning nodes radios On and Off, as may be the case of Gossip [51] used over S-MAC [4] or T-MAC [5]. GSP may also make a good combination with those MAC protocols. However, GSP decides when to turn the radio On or Off, and to optimize performance, those decisions must agree with the characteristics of the MAC layer. As an example, consider using S-MAC or T-MAC with GSP: the Gossip Period must harmonize with the SYNC schedule and the designer must decide all transmission and sleeping periods in advance, after considering the Gossip

Probability. A similar situation occurs with B-MAC. Sift does not generate conflicts in turning the radio On or Off, but it requires to interact with GSP to know when the node is able to send a packet (because the radio is on and also the appropriate backoff time has elapsed). Observe arbitrating the interaction between different protocols adds complexity, further places to introduce errors and the opportunity for hackers to find protocol exploits.

2.4.3 Summary

Cross-layering research includes at least two protocols in the analysis of a communication system, either with or without communication between different layers. Protocol selection affects the performance of a Communication System. Current research does not consider interactions of the three lower layers in the OSI model sense nor tries to use characteristics from one layer to benefit another one.

2.5 LAYERING AS OPTIMIZATION DECOMPOSITION

One step forward in cross-layer analysis is the framework known as Layering as Optimization Decomposition. The framework builds upon the idea of obtaining mathematical understanding of network architectures, considering modeling protocols in all the layers and their interaction in order to fulfill a particular goal in the network. That goal is the optimization problem, expressed as a generalized Network Utility Maximization (NUM), whose metric can be defined according to user requirements (such as delay, jitter or bit rate) or to network operator requirements (such as energy use, lifetime or congestion) [61]. The framework employs the network as the

optimizer, the user (or operator) requirements as the optimization objective, provides a global performance benchmark and helps design the solutions to obtain the benchmark [61]. The main idea is to divide (or decompose) every NUM in different subproblems. The decomposition is not unique, since a problem can be divided in different ways. However, each subproblem in a decomposition corresponds to a layer in a communication network, and all the subproblems form a "Vertical Decomposition". Every vertical decomposition represents a protocol stack designed considering the network as a whole. "Horizontal Decompositions" allow execution of one function in different network elements [61]. Since different decompositions can solve the same problem, the framework should also offer criteria in how to choose the best decomposition for a particular situation. One idea is to create a methodology for listing, compare and determine the best of all alternatives, but this is an open field for future research.

Different studies show mathematical modeling of protocols as solution to optimization problems, such as equilibrium for heterogeneous congestion control protocols for TCP [62], the Border Gateway Protocol (BGP) [63] and Scheduling Based MAC protocols [64]. Physical layer protocols try to maximize data rate considering all constraints present in the medium, so this is another example using the framework [61]. The work in [65] presents two network problems (constraints for the whole network such as dynamic channel allocation, and Distributed Transmission Scheduling), three alternative decomposition schemes and a way to compare their performance in solving the problem.

2.6 COST-BASED MODELS FOR QUERY OPTIMIZATION

Databases store information accessible to a user through queries. Different methods exist to execute the query and, although all of them should provide the same information, the cost of executing them may be different [66]. One metric to evaluate this cost is the time it takes to execute the query, which can vary significantly from one method to the other. Other metrics are total work (as cpu and disk usage) or delay, when the query should be performed in remote sites only accessible through a wide area network [67]. All Database Management Systems have a Query Optimizer function to evaluate the different methods to execute the query and to choose the least costly. The query optimizer uses a Cost Model based in mathematical formulas to estimate the time (or other cost) required to execute the query using each method [66]. To create the cost model, the designer tests a large amount of queries and statistically derives the coefficients for the formulas. A model does not require absolute precision in estimating the cost. As a matter of fact, a cost model providing an estimation with an error less than 20%, can be useful and cost-effective in terms of resources employed [68]. Some examples of cost model developing are: a cost model factoring parallelism and scheduling in query execution, highlighting the fact that the objective of query optimization is not to find the best method, but rather to avoid the bad ones [69]. Research in [68] shows a cost model for a federated database system with different type of data located in multiple remote sites, which increases the complexity in building an adequate cost model. Another example is a framework to help federated systems to obtain the information they require to get the cost of executing queries through diverse data sources [70].

3.0 PRELIMINARY WORK

The section presents a summary of the results obtained from previous studies of GSP regarding two possible MAC protocols, Capture Effect and one Energy use Model.

3.1 GOSSIP-BASED SLEEP PROTOCOL (GSP)

Gossip-based Sleep Protocol (GSP) improves on Gossiping because it drops a packet by not receiving it. A node receiving a packet must retransmit it, thus energy spent for receiving was not wasted. GSP divides time in Gossip Periods with fixed duration [10]. At the beginning of each gossip period, every node decides with probability p, the Gossip Probability, to turn off its radio, and with probability (1-p) to turn it on, ready to receive. A node receiving one packet must retransmit it in the following gossip period. All sleeping nodes must wake up in the next gossip period. Figure 12 shows one example of GSP where each gossip period has duration T. One node can be in one of three possible states: On Receiving, On Transmitting and Off.

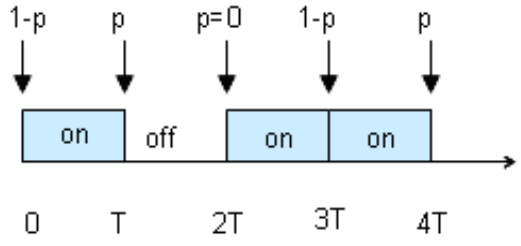

Figure 12. Behavior in one node using GSP.

Yupho, Calle and Kabara used the energy use model presented in [8] to analyze Average Remaining Energy (ARE) using GSP [71]. ARE shows the average energy available to the nodes in the network after the first node uses all its energy, which literature often considers as a reference of network lifetime. GSP increased network lifetime both with Known Path (finding one best route using non-sleeping nodes) and Unknown Path (just broadcasting the packet) schemes on several network sizes, obtaining better improvement with smaller networks.

3.2 MAC PROTOCOL STUDY

Calle and Kabara presented two MAC protocols belonging to the contention-based category to take advantage of GSP characteristics, decreasing energy use and not increasing overhead [16]. The work assumes nodes generate data in small increments and therefore fits in one data packet. In addition, because these protocols do not need a central entity controlling the medium access, no control bits are needed in the data frame and no control frames such as Request to Send (RTS), Clear To Send (CTS) or Acknowledge (ACK).

3.2.1 Protocol Description

MACGSP1 is the first MAC protocol investigated and it operates according to the following algorithm: when a node has information of its own to transmit, the node is a source and it will send the packet in the next gossip period T. After sending this packet, the source will sleep for the next two gossip periods. When a node receives one packet from another node, the receiver node acts as a relay, retransmitting the packet in the next gossip period, after which the node will

sleep for one gossip period. A sleeping node decides (after 2T if it is a source or T if it is a relay) to continue sleeping with the gossip probability p, referred to as p_{gsp}, or to wake up with probability $(1-p_{gsp})$.

MACGSP2 operates in a similar fashion to MACGSP1. However, MACGSP2 requires that all sleeping nodes wake up after their corresponding gossip period. That is, the source wakes up after sleeping 2T and relay nodes wake up after 1T. Nodes that did not transmit but were sleeping in the previous gossip period also wake up.

The routing protocol GSP turns off the radio with probability p_{gsp}, but not necessarily after each transmission, so a node that sent a packet may receive the same packet in the next gossip period. Both MACGSP1 and 2 eliminate duplicate frame reception in the gossip period following a transmission but they partition the network because the nodes are "disconnected" from the network when their radios are off.

The study analyzed performance of GSP with no MAC protocol, referred to as REALGSP, GSP over MACGSP1 (termed MACGSP1) and GSP over MACGSP2 (called MACGSP2), via simulations on square grid networks of 100, 400 and 900 nodes. Simulations synchronize all nodes and they can reach their one-hop neighbors with no bit errors due to channel conditions. Figure 13 depicts one example square grid with 36 nodes and the radio range defined for the simulations. Node A is the source, node S is the sink and all nodes have a radio range as the circle showed around node S. Nodes B, C, D and E are out of range of S, so the packets cannot travel directly through the grid diagonals.

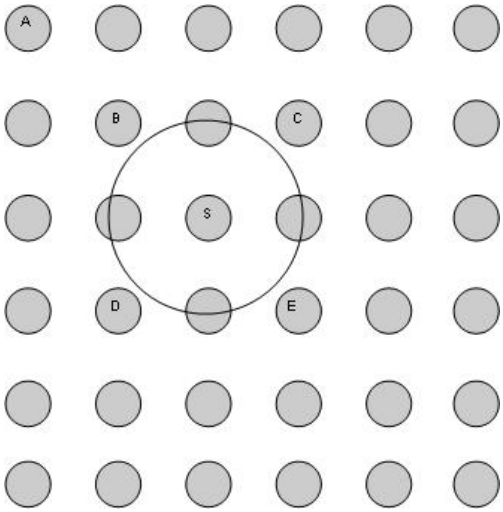

Figure 13. Square grid network.

A physical layer protocol employing FSK modulation [8] is used and therefore capture probability β (the probability of a node correctly detecting one out of two simultaneous received packets) was considered with values ranging from 0.5 to 1.0. β1 is the probability of capturing one packet when receiving three packets; β1 is assumed constant and equal to 0.35. β2 is the capture probability when four packets arrive simultaneously, also assumed constant and equal to 0.1. These experiments employed a single source and sink for each grid. Simulations employed source and sink in fixed positions in three particular cases: opposite corners of the network, source in the upper corner and sink in the center of the grid (as in Figure 13), source and sink in the opposite ends of the middle row. Table 10 shows the energy use model for a system employing 5 dBm transmission power, 21 bytes frame size and 19.2 kbps data rate [40].

Table 10. Energy Model for Crossbow Mica2

Node state	Energy per bit (μJoules)	Energy per frame (μJoules)
Transmit	4.28	720
Receive	2.36	397
Sleep (RadioOff)	0.87	150

The gossip period duration is the Time required to send a Packet (TP). Graphs show results averaged over 40 runs, 200 packets each run with statistical validation of 90% confidence levels, unless otherwise noted. The following are the metric definitions:

Packet Delivery Probability = Number of Packets that reached the sink / Total Packets sent by the source

Delay = Average number of gossip periods from the time a packet is transmitted by the source to the time it arrives at the sink.

Mean Duplicated Packets in Source = Total number of repeated packets received by the source / total new packets sent

Network Energy use = total energy used by all the nodes in the network to send 200 packets. The study further differentiates this metric in Total Energy for Transmitting, Total Energy for Receiving and Total Energy with Radio Off. The simulation stops when the first packet arrives to the sink or when the packet is lost in the network.

3.2.2 Results

Results for each case of β were consistent across protocols, e.g. β=1 obtained the smallest delay and decreasing values of β increased delay in all protocols under test. Additionally, most metrics did not show consistent differences for various values of β. Therefore results are shown for a single value of β=0.7, as it portrays typical performance of the protocols in all experiments. Also to show typical values, figures present results when the source is located in the upper corner and sink is in the center of the grid (see Figure 13). Results with both sink and center in the middle row show the minimum values for the metrics and results with sink in the upper corner and sink in the lower corner show the extreme values.

Figure 14 presents the probability of the sink receiving a packet. Each line type represents a different network size. Each protocol has the same marker. REALGSP has the highest probability of the three protocols for all network sizes. MACGSP2 shows between 5% and 10% less packet reception probability for $p_{gsp} \geq 0.2$. Additionally, Figure 14 shows lower probabilities for each protocol as size increases; however, values for network sizes 400 and 900 overlap at 90% confidence level, so there is no appreciable difference. Note when a relay retransmits a packet and all receiving neighbors are in the Off state, the packet is lost. Since GSP retransmits the same packet, several duplicates should traverse the network at the same time, so the event of all neighbors turned off may benefit communication by eliminating duplicates. On the other hand, if the eliminated packet is the original (the one just sent by the source), the information in this packet is lost. If this happens to all duplicates before any of them reaches the sink, the information is also lost. However, this situation depends on the value of p_{gsp} (the Gossip Probability) and the total number of possible receiving neighbors. As an example, consider $p_{gsp} = 0.3$ in Figure 14: smaller values of p_{gsp} show high probability of receiving the packet. When p_{gsp} increases, the probability of discarding the packet increases substantially, diminishing the packet reception probability. One possible mechanism for improving this situation without increasing overhead may be using retransmissions as implicit ACKs: a transmitter node can listen to the medium after transmitting, in order to check if other node relayed its packet.

Observe curves in Figure 14 seem to follow a square-law function but MACGSP1 abruptly falls for $p_{gsp} \geq 0.3$, which results in either a very high packet loss rate or low energy savings. Nonetheless, analyzing the other metrics it is possible to determine if the reduction in packet reception is worth the implementation of MACGSP1 or 2 instead of GSP.

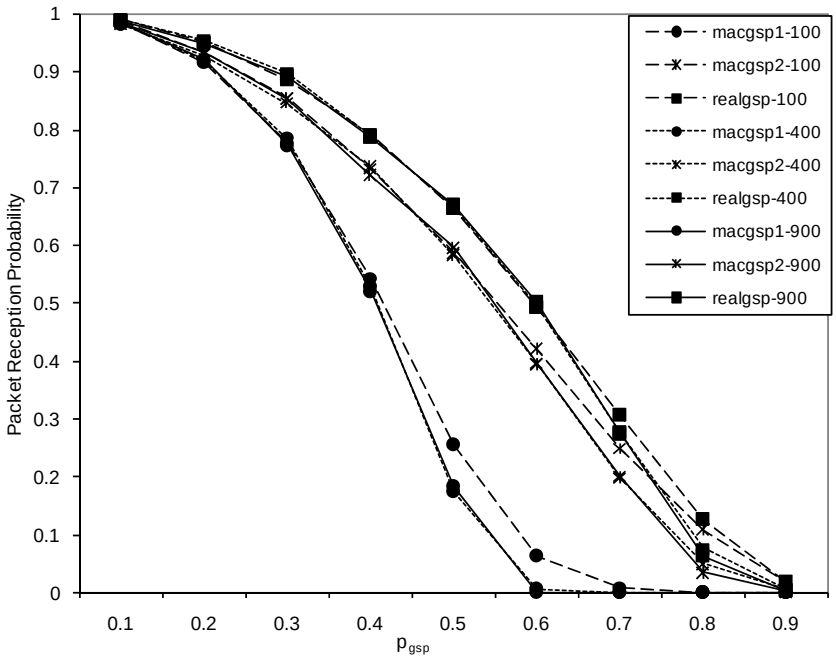

Figure 14. Probability of packet reception by the sink.

Figure 15 shows the mean delay for each packet from source to destination. At 90% confidence level, there is no noticeable difference between MACGSP2 and REALGSP when p_{gsp} ≤0.4. MACGSP1 grows exponentially due to the sharp fall the protocol has in Packet Reception Probability, so the packets are not likely to arrive to the destination and delay grows accordingly. Note delay can only be calculated when the packet arrives at the sink, so for values of p_{gsp} ≥0.6, delay in MACGSP1 is not shown in the graph because the reception probability is close to zero. MACGSP2 and REALGSP seem to follow an opposite behavior to Packet Reception Probability, looking like exponential functions for 400 and 900 nodes where packets take longer paths due to the network partition given by the gossip probability. However, the reception probability is noticeable up to p_{gsp} =0.8, so the delay can be computed up to this value.

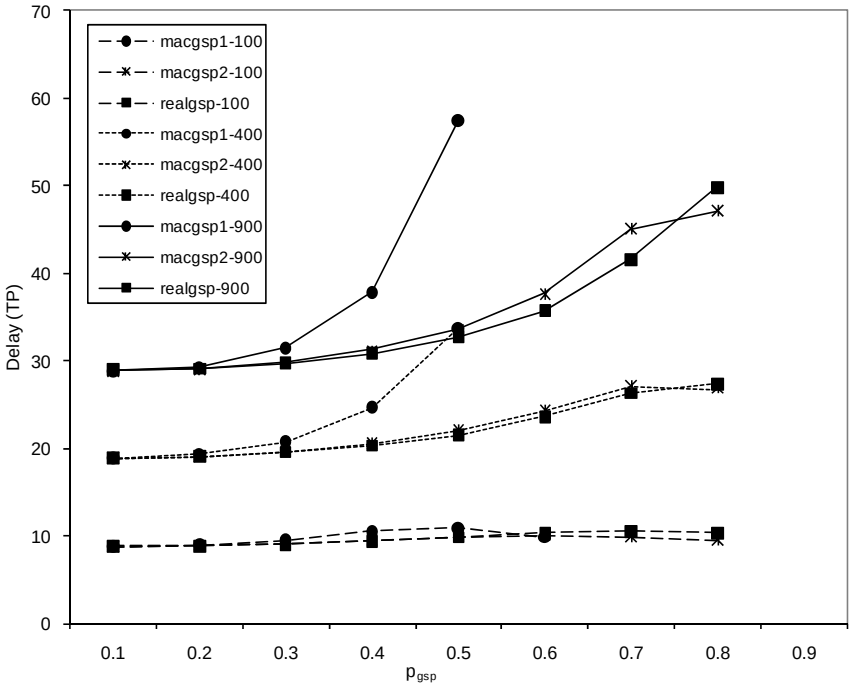

Figure 15. Delay from source to sink vs. gossip probability.

Delay obviously increases in larger networks. However, since flooding finds all paths from source to destination, including the shortest [15], the analysis can compare these results against an optimal solution. Let

N=number of nodes in one row

Then the shortest path from the sink to the destination should be N-2, since the packets cannot travel through the diagonals of the network (see Figure 13). Note that for MACGSP2 and REALGSP most values for $p_{gsp} \leq 0.5$ are close to N-1, so the protocols add only 1 hop to the shortest path, even though nodes in the network are disconnected and *the best route may be different in each gossip period*, since different nodes may be sleeping.

Although there are no routing tables or control packets generating overhead, the three GSP protocols have overhead in the form of duplicated packets. Figure 16 illustrates the number of duplicates received at one node, in this case the source. Curves decrease exponentially and MACGSP1 again falls sharper, meaning it reduces the number of duplicates faster than the other

protocols. Nonetheless, duplicate reduction means fewer packets are traveling the network trying to find routes to arrive to the sink, explaining the abrupt fall in reception probability and the corresponding delay behavior. Note MACGSP1 and 2 show no visible difference when p_{gsp} =0.1 at 90% Confidence level. Both protocols reduce the number of duplicates generated by REALGSP at least by 55% for all network sizes when $p_{gsp} \leq 0.5$.

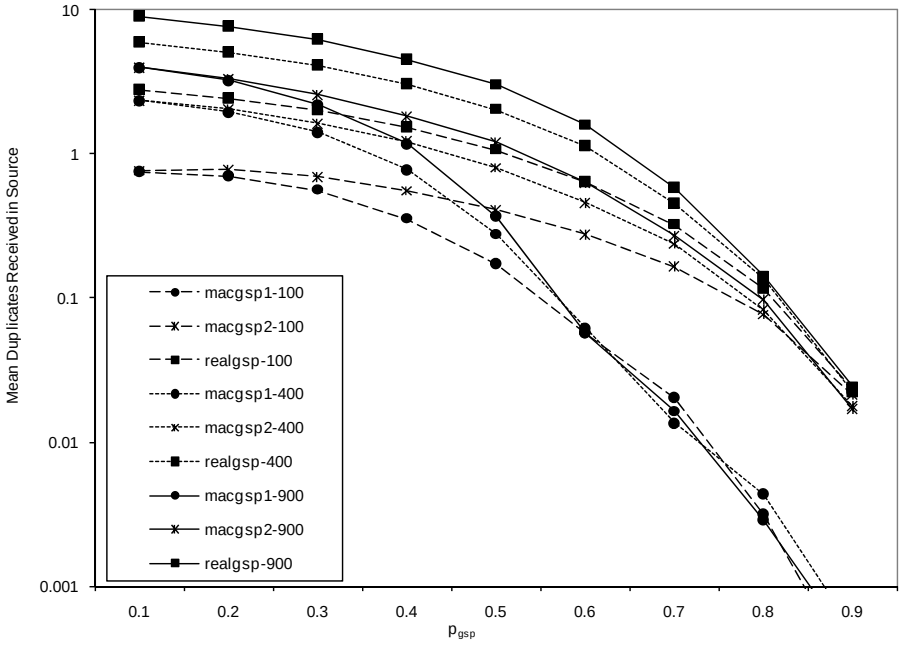

Figure 16. Average number of duplicates received by the source.

Reducing the number of duplicate packets saves energy by reducing both the number of packet transmissions and receptions. Figure 17 plots the Total Energy use as a function of gossip probability. MACGSP1 has the lowest Total Energy use but its Packet Reception Probability is also the lowest, so even if the protocol seems to be consuming less energy, it is not delivering packets. Curves decrease exponentially, showing the effects of duplicate reduction and the gossip probability, turning off the radios in more nodes. The trend for each protocol is consistent regardless of network size (i.e. MACGSP1 always has a smaller energy use than the other two

protocols for each network size). REALGSP and MACGSP2 have similar behavior, but MACGSP2 reduces total energy use between 4% and 12%.

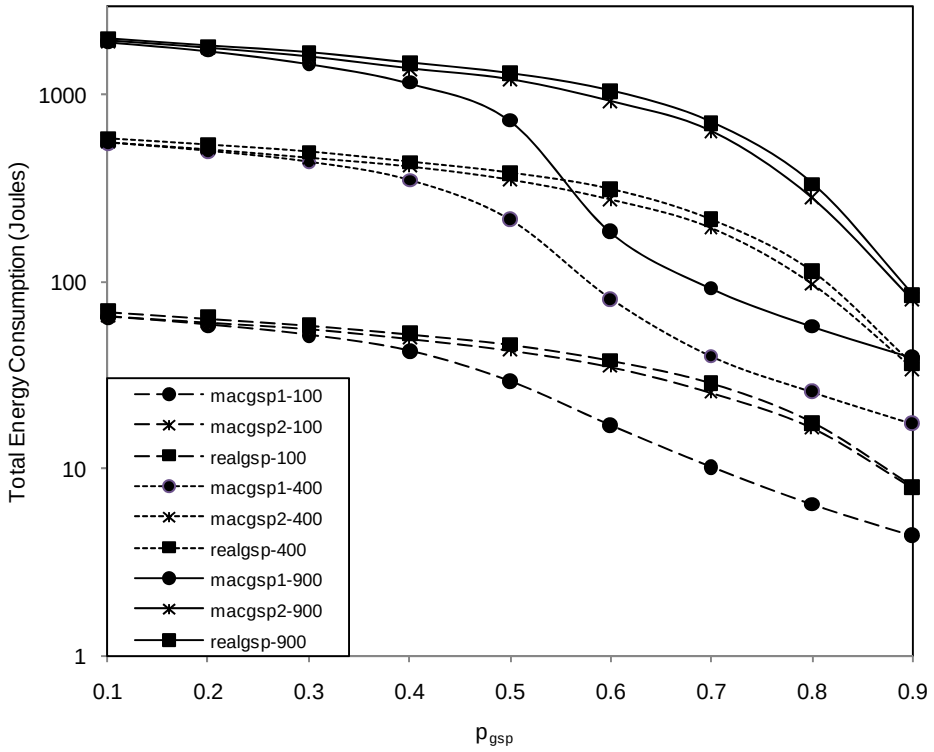

Figure 17. Energy use for all nodes.

Figure 18 plots the energy required for transmitting the 200 packets simulated in each run as a function of the step probability. Again, there is no noticeable difference at 90% confidence level for MACGSP1 and 2 with p_{gsp} =0.1. MACGSP2 reduces the energy use by 27% or more when compared to REALGSP. The effect of reducing duplicates reflects more directly in this metric, showing also exponentially decreasing behavior. Notice that energy used for transmitting is one order of magnitude smaller than Total Energy, so when added to the total, the 27% is not apparent.

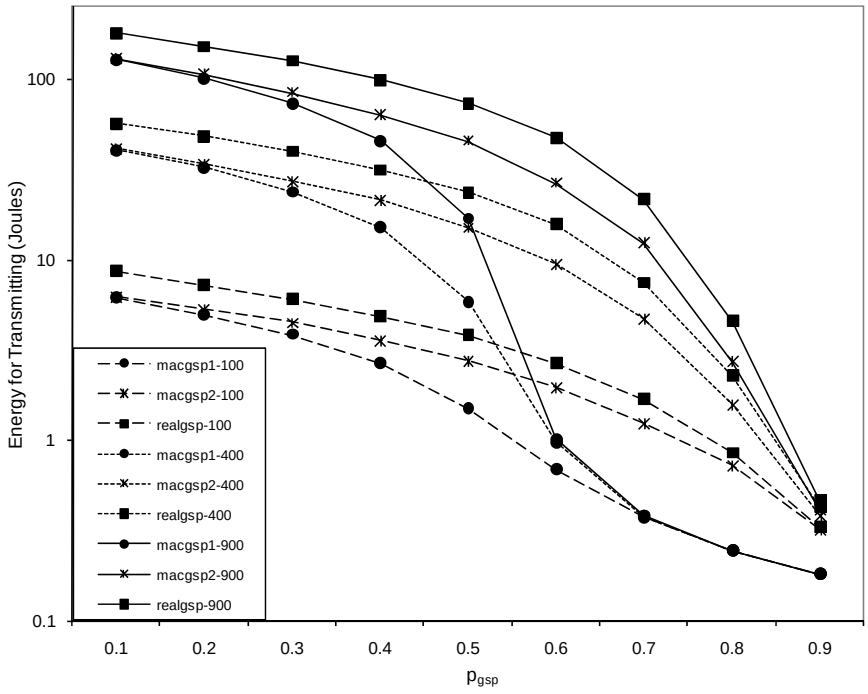

Figure 18. Energy for transmitting 200 packets in the network.

Figure 19 shows the energy used by nodes during the Radio Off (sleep) state. The 400 and 900 node networks show that MACGSP2 spends a slightly more energy than REALGSP for p_{gsp} <0.3. The reason for that is MACGSP2 places the nodes into sleep mode more than REALGSP. However, for p_{gsp} =0.5 there are no evident differences among the three protocols at 90% confidence level. As p increases, more nodes sleep more often, resulting in REALGSP spending more energy in this state. These increases appear as increasing trends in all curves when p_{gsp} ≤0.5. The curves in Figure 19 also display a trend downward for p_{gsp} >0.5. The behavior occurs because the simulation stops if the packet is lost in the network. As p_{gsp} increases, the probability that a packet is lost also increases. Simulation counts energy use during the life of the packet and its duplicates, so the energy used during sleep reflects the shortened packet lifetimes.

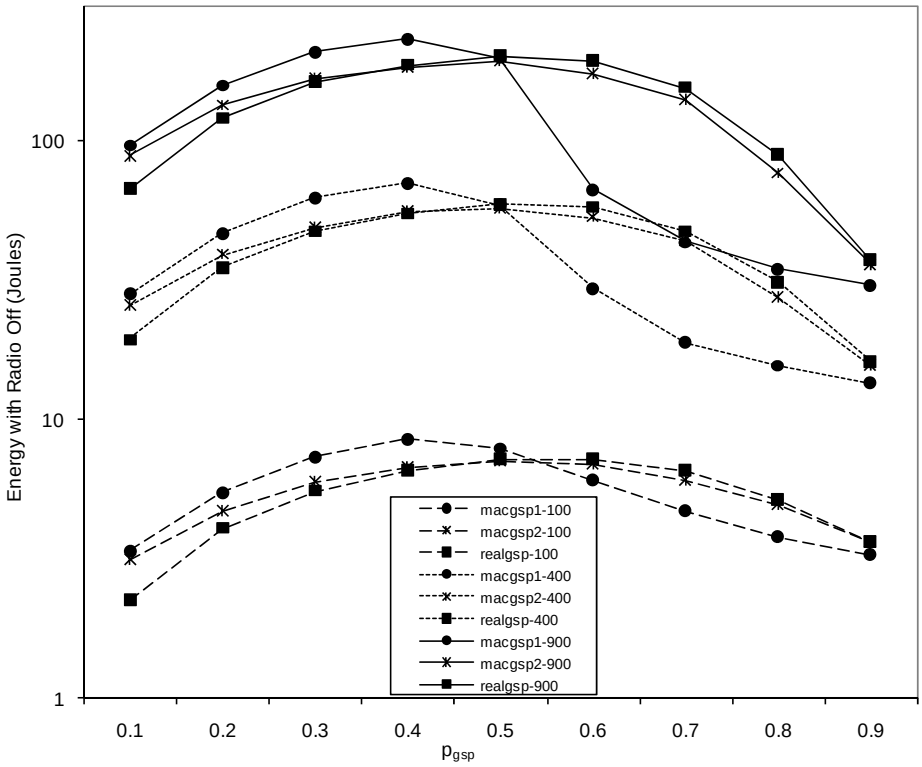

Figure 19. Energy used with radio off.

In the radio off state all other subsystems in the node still draw power. Shutting down additional subsystems may save additional energy, but may increase the time it takes for packet transmission, since all systems must restart every time nodes wakeup.

Figure 20 plots energy spent in the entire network, as a function of p_{gsp}, for receiving 200 packets. Only when p_{gsp} =0.9 do MACGSP2 and REALGSP exhibit no obvious differences at 90% confidence level. Values are of the same order of magnitude as the Total Energy (plotted in Figure 17) therefore the communication process spends most of the energy for receiving and idle listening. As the sleep probability increases, the energy spent in receiving decreases exponentially showing the influence of the Gossip Probability for turning off the radios, and the packet lost situation previously described. Results show approximately 85% of all energy spent in the receiving state.

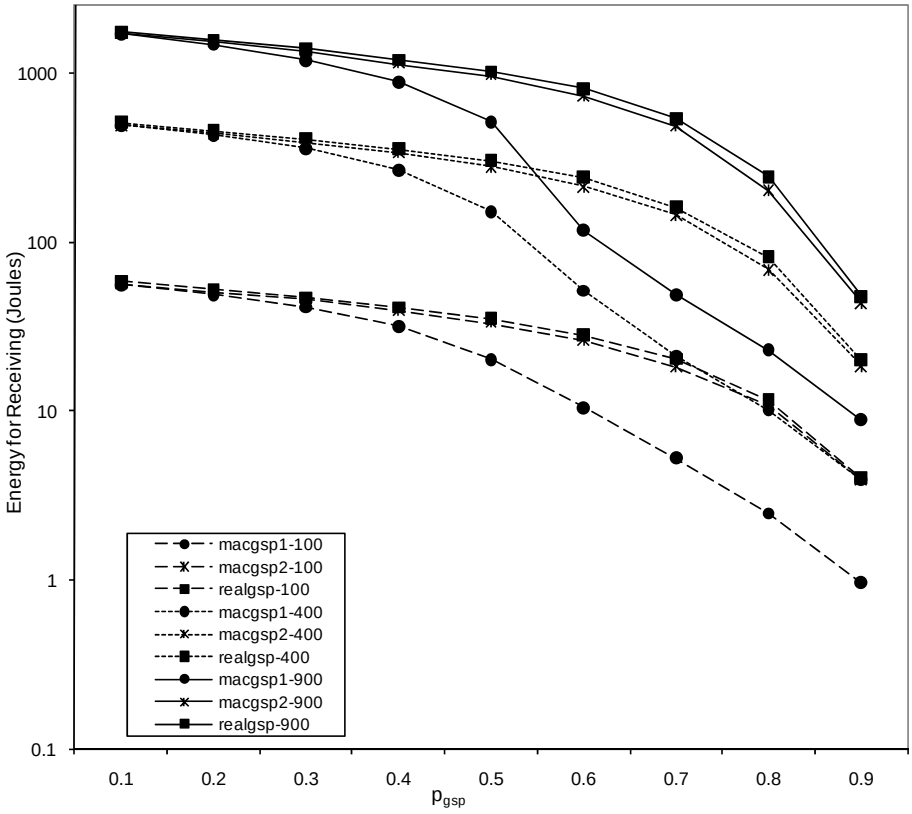

Figure 20. Energy for receiving.

All figures show similar trends and behavior when going from one network size to the next one. Compared to MACGSP2, for p_{gsp} <0.3 MACGSP1 results in less energy use and fewer duplicates but increased delay and packet loss. However, for larger values of p_{gsp}, MACGSP2 reduces the rate of energy use without reducing packet delivery.

3.2.3 Conclusions from this Study

Accounting for the Capture Effect in systems using angle modulation schemes shows the system will deliver more packets compared to channel models that assume a collision will occur. However, as the probability of two signals reaching a receiver decreases the impact of the capture effect also decreases. As the sleep probability for the Gossip-based Sleep Protocol (GSP)

increases, fewer nodes are awake to receive and retransmit packets and therefore the Capture Effect has the greatest impact for low sleep probabilities. Protocols other than GSP, which may experience collisions, will also see improved packet delivery. Additionally, in a fading channel environment, where the transmission range is smaller, there should be fewer collisions.

The trend for MACGSP2 packet reception probability, as a function of gossip probability, p_{gsp}, is the same for network sizes 400 and larger. The results show that each of the protocols in the GSP family scales. MACGSP2 results are independent of network size and therefore scales particularly well to large networks. Delay results show increases with network size because packets must travel a greater distance through more relay nodes. MACGSP2 and GSP do not use the shortest path, but the delay is very close to the minimum, on average only one hop longer and there is no additional overhead. Duplicates are still necessary for moving the information from source to destination because some paths will fail. MACGSP2 reduces the number of duplicates generated by GSP and saves energy with no change in delay and a small reduction in the packet reception probability (10%) compared to GSP with no MAC protocol.

MACGSP2 improves upon the energy use rate by reducing the number of duplicated packets generated, thus reducing the amount of energy spent receiving and retransmitting these duplicates. However, MACGSP2 does not reduce the energy used in the idle listening state. Applications requiring high Packet Reception Probability and low Delay should use Gossip probabilities of 0.3 or lower. However, these values of p_{gsp} generate the maximum energy use. If the application tolerates higher delay and lower packet reception probability as tradeoff for lower energy use, higher values of p_{gsp} may be used. MACGSP2 and GSP are extremely simple to implement because there are no control packets, no distributed algorithms, no addressing and no

location information requirements. MACGSP2 can improve by further reducing idle listening without adding overhead.

3.3 CAPTURE EFFECT STUDY

Calle and Kabara studied the influence of Capture Effect on GSP [17] using simulations without additional protocols, no entity controlling the medium access, no additional bits in the data frame and no control frames such as Request to Send (RTS), Clear To Send (CTS) or Acknowledge (ACK). Transmitted data is small enough to fit in one data packet. A physical layer protocol employing FSK modulation [8] is used making Capture Effect feasible. Capture Probability is the chance of receiving a packet when one node simultaneously receives signals from multiple transmitters. Figure 21 illustrates the physical 25-node network topology used in the experiments. Horizontal and vertical distances between two nodes have the same value, r. The study employs a radio model in which there are no bit errors if the receiver node is located at a distance r from the transmitter. The communication is symmetric, so if node 3 can listen to node 4, node 4 can also listen to node 3. The experiments employed a single source located in the upper corner of the grid and one sink, located in the opposite lower corner, in order to test the largest number of hops through the network.

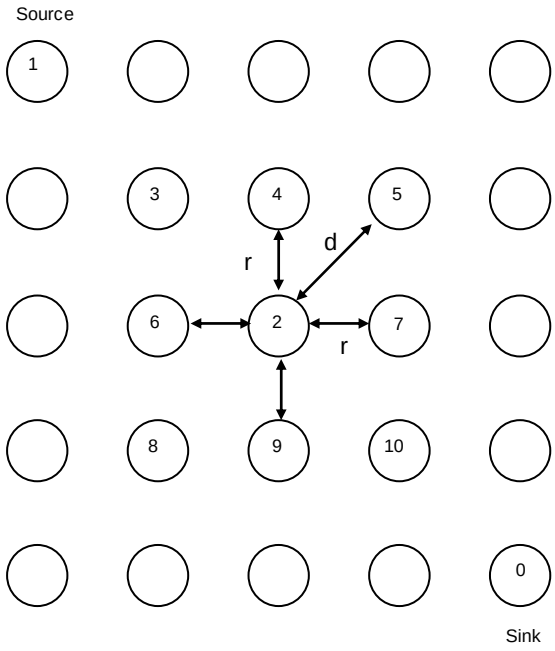

Figure 21. Network configuration.

Energy conservation is employed at the transmit power level, thus the transmit power Pt is the same for all nodes and it is only enough to reach the next neighbor within distance r. Consider the signals received at node 2 in Figure 21 using a squared path loss channel model such as Free Space Loss, LFREE. The received power at node 2, Pr, is given by:

$$P_r\ (dBm) = P_t\ (dBm) - L_{FREE} \tag{4}$$

$$L_{FREE}\ (dB) = 20\ log\ ((4\pi x) / \lambda) \tag{5}$$

With

λ = operating wavelength

x = distance from source to destination.

Therefore, a transmission from node 7 reaches node 2 with

$$P_{r7}\ (dBm) = P_t\ (dBm) - 20\ log\ ((4\pi r) / \lambda)$$

While a transmission from node 5 is received as:

$$P_{r5}\ (dBm) = P_t\ (dBm) - 20 \log\ (\ (4\pi\ r) / \lambda)$$

$$P_{r5}\ (dBm) = P_t\ (dBm) - 20 \log\ ((4\pi r) / \lambda) - 3dB$$

$$P_{r5}\ (dBm) = P_{r7}\ (dBm) - 3\ dB \quad (6)$$

Equation (6) shows signals from node 5 are received by node 2 with half the power as signals from node 7. If the channel is better modeled by an n^4 path loss model then the signal transmitted from node 5 will reach node 2 with 25% of the power of a signal coming from node 7. Both channel models result in a signal arriving from a diagonal neighbor that has negligible impact on the receiver's ability to capture a stronger signal (coming from neighbors located within distance r), so the experiments consider no communication across the diagonals.

Although many models exist for the physical nature of the capture effect, e.g. [23], [26] and [27], in order to systematically study the effect, the Capture Probability was defined as the chance of receiving a packet when one node simultaneously receives signals from multiple transmitters. When one node receives information from two transmitters simultaneously, the node correctly receives one of the packets with capture probability β and discards both packets with probability (1-β). For each simulation run, all nodes receiving two packets at the same time have identical probability β of detecting one of the packets, which is consistent with the radio model used. To obtain statistically significant results, 40 runs were performed with each value of β. After finishing each set of runs, experiments changed β in steps of 0.1. The study gradually measures the influence of capture effect by changing β from 0.5 to 1.0. β=1 implies there is a perfect capture effect, so in the event two transmissions are received simultaneously, one of the packets is always received and the other is dropped, similar to[3]. The energy use model is presented in Table 10 [40]. The simulation stops when the first packet arrives to the sink or when the packet is lost in the network. Each simulation run employed 200 original data packets. The

study obtained results for 40 runs of each simulation and averages and 90% confidence intervals were calculated for each of the following metrics.

Packet Delivery Probability = number of packets that reached the sink / total packets sent by the source

Delay = average number of gossip periods from the time a packet is transmitted by the source to the time it arrives at the sink.

Duplicated Packets in Source = total number of repeated packets received by the source / total new packets sent

Network Energy use = total energy used by all the nodes in the network to send 200 packets.

Energy per successful packet received = Total Energy spent by the network to send 200 packets / number of packets successfully received at the sink.

3.3.1 Results

Figures display all metrics as a function of the gossip probability p, referred as p_{gsp}. Only the Confidence Intervals for the extreme cases β=0.5 and β=1.0 are shown to clarify the behavior. Figure 22 presents the Probability of the sink receiving a packet. The case presented in [3] corresponds to β=1.0, perfect Capture Effect, which in Figure 22 exhibits the highest Packet Delivery Probability.

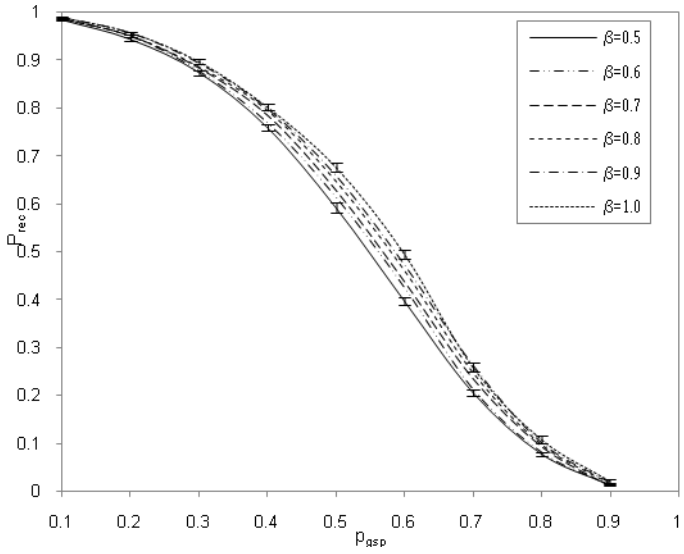

Figure 22. Probability of packet reception by the sink.

Decreasing values of β shift the curves downwards, generating smaller reception probabilities than a perfect Capture Effect. The result is intuitive in that a higher probability of receiving one out of two simultaneous packets should improve information forwarding toward the sink. However, there is no visible difference among closer values of β, since confidence intervals overlap at 90% level. Capture Effect may improve the reception probability by the sink, but for this particular configuration, statistically speaking the difference only can be seen when the difference in β value is ≥ 0.3 (e.g. β=0.5 and β=0.8) and for values of gossip probability between 0.4 and 0.7.

Figure 23 depicts the delay form the source to the destination. The metric shows differences for different β values when gossip probability < 0.6 at 90% confidence level. The Perfect Capture Effect generates the lowest delay. When decreasing β, Delay increases because nodes capture fewer packets and there are fewer retransmissions towards the sink; however, duplicates are still traveling through the network and some of them arrive to the sink at a later stage.

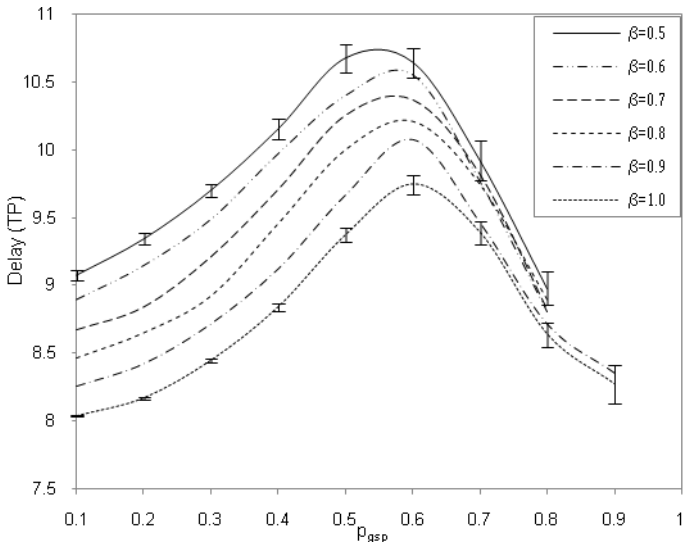

Figure 23. Delay from source to sink.

Delay also increases when gossip probability increases because more radios are in the off state, so the packet must travel through a longer route for arriving at the sink. When p_{gsp}=0.6 the difference achieved by Capture Effect is not consistent for all consecutive values of β. Also, for p_{gsp}>0.6, the curves descend, which occurs because energy metrics are cumulative (total energy in each of the states) and the conditions established for the simulation. Delay is calculated only when a packet arrives at the sink. When p_{gsp}>0.6 the network is often partitioned and few packets can reach the sink, however those that reach the sink arrive via short routes. Figure 23 shows that for p_{gsp}>0.6 the few packets left in the network arrive to the destination faster than they would with lower values of p_{gsp}.

Figure 24 plots duplicates received at source node as a function of p_{gsp}, to estimate the number of duplicates traversing the network and generating overhead. For $p_{gsp} \leq 0.4$ the Capture Effect significantly increases the number of duplicate packets. A Perfect Capture Effect allows nodes to receive more packets and to retransmit more duplicates inside the network. The opposite occurs for smaller values of β where the number of packets received at intermediate

nodes decrease and duplicates diminish accordingly. In addition, the number of duplicate packets reduces as p increases, which is a consequence of having better chances of turning off the radios in the network. The Gossip Probability is then a tool to limit the overhead generated by the protocol, which also helps to save energy.

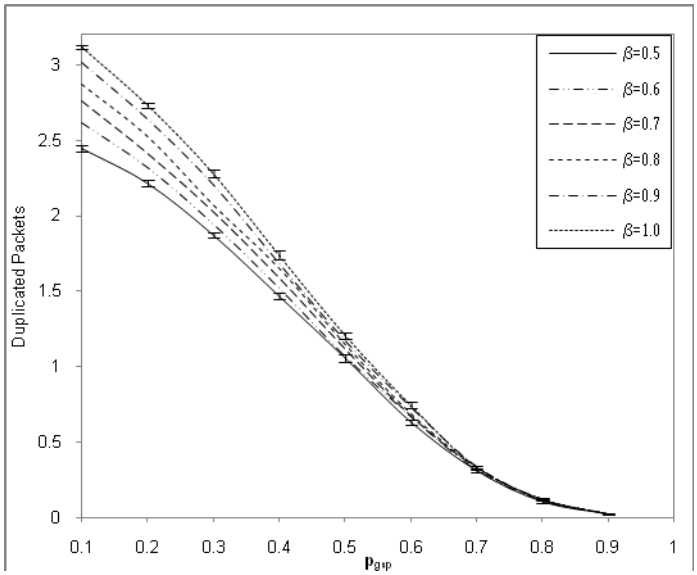

Figure 24. Duplicates received in source.

Figure 25 illustrates the Total Energy use, by all nodes in the network, necessary to send 200 packets from source to destination. Larger values of p_{gsp} create energy savings, as expected. A perfect Capture Effect decreases energy use compared to smaller values of β, so savings given by reducing duplicates are not noticeable in the total energy used. Confidence Intervals of consecutive β values overlap, so there is no visible difference between two consecutive values (such as β=0.5 and 0.6). However, β values at least 0.3 apart from each other show differences when p_{gsp}<0.6. However, this particular p value presents no observable difference in some of the previous metrics too, so Capture Probability has little or no effect for Gossip Probability values of 0.6 or higher.

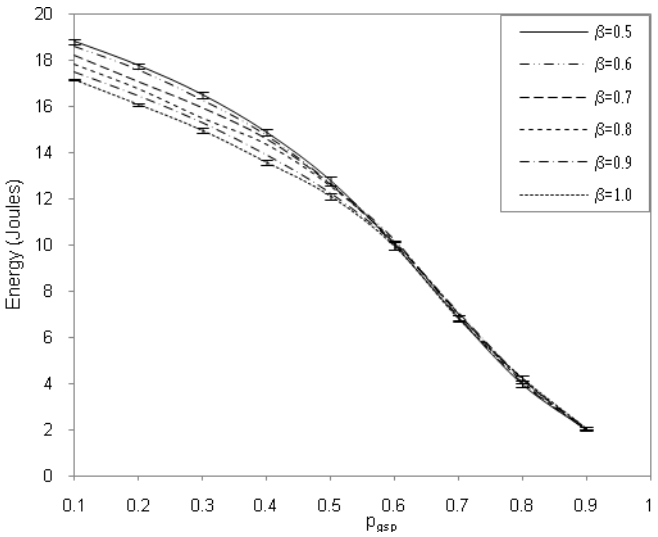

Figure 25. Total energy use.

Note a perfect capture effect generates the higher packet reception probability with the lowest energy use. To illustrate the relation between this two metrics, Figure 26 shows the energy spent to receive a packet by the sink.

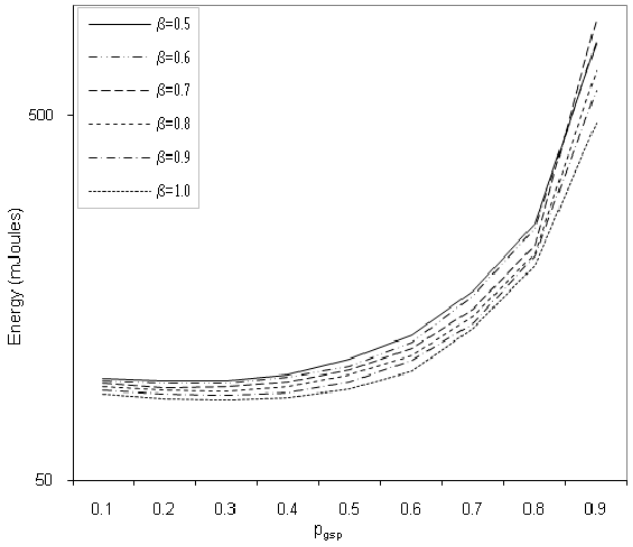

Figure 26. Energy per successfully received packet.

Figure 26 show exponential trends for all values of β. When increasing p_{gsp} the Total energy use decreases but the sink is receiving a smaller amount of packets (as seen in Figure 22).

Total energy use does not decrease at the same rate as the number of received packets, thus the energy for succesfully receiving one packet increases. Following the behavior in Figure 22, a perfect capture effect generates the smallest values of energy use.

3.3.2 Conclusions from this Study

Exploiting positive interactions between multiple layers of the OSI model may reduce the rate of energy use in Wireless Sensor Networks. The Capture Effect is an emergent property of physical characteristics present in some modulation schemes such as FSK. Many models exist for calculating the probability of capture effect occurring, therefore, results in this paper correspond to experiments in which the capture probability was systematically varied, independent of any underlying specific capture model. Results from these simulation studies show that the capture effect can increase packet reception probability, which in turn decreases delay, energy use and the number of duplicate packets generated in the network.

GSP can take advantage of capture effect to improve performance with no additional overhead. Results show that Gossip probabilities p_{gsp} <0.6 are more sensitive to the capture effect, for all metrics. Applications requiring low delay, high probability of packet delivery and low energy use are best served by p_{gsp} values between 0.3 and 0.4, because these values improve performance by allowing greater than 70% of the packets to be delivered, saving energy, eliminating duplicates while increasing delay by only one TP when compared to p_{gsp}=0.1.

Because of the low probability of collisions occurring, metrics studied were little influenced by the capture effect when p_{gsp} >0.6. Nonetheless, applications with lower requirements in packet reception probability, such as some monitoring applications that generate redundant data, may use Gossip probabilities p_{gsp} >0.6, because even with a packet reception

probability lower than 0.4, some packets arrive at the sink with the required information, after a moderate delay but with substantial savings in the general energy use of the network, which is the main goal. However, these values of p_{gsp} increase the energy use per successful packet received by the sink, because the savings do not increase at the same rate as the number of packets received by the sink decrease.

Future work may explore the influence of capture effect on GSP in topologies other than square grids. Future work may also explore the behavior of GSP with additional physical phenomena such as fading. In general, if networking protocols permit collisions to occur, the number of collisions and their effect on network performance will be greatly reduced after accounting for capture effect.

3.4 ENERGY USE MODEL

Calle and Kabara presented a Measurement-Based Energy use Model for Crossbow's Mica2 motes [8], implementing a network running GSP.

3.4.1 Experimental design

The study instrumented Mica2 motes for measuring energy use rate for various states employed by the GSP protocol. Figure 1 shows the MPR400 device. The Mica2 sensors receive energy from two AA alkaline batteries and use an Atmel Atmega 128L microcontroller circuit and the CC1000 integrated radio circuit [4]. The Chipcon CC1000, operates in the 902-928 MHz band. The 128L is an 8-bit RISC microcontroller with 128 Kbytes of programmable Flash memory, 4

Kbytes EEPROM, 4Kbytes internal SRAM and can manage up to 64 Kbytes of external memory, as an optional feature [72]. The microcontroller has an 8-channel 10bit ADC, an 8-Mhz crystal and programming options via UART or via JTAG interfaces [72]. The software for these experiments was loaded onto the Mica2 through the UART port, of the MIB510 base module [6]. The implementation of GSP kept the microcontroller in an active state at all times. The radio was set to use a 903 MHz channel with Binary FSK, transmission power of 5 dBm and Manchester encoding. The data rate was 19.2 kbps [73].

The Mica2 mote is provided with a default MAC protocol, which implements a CSMA/CA variant. The protocol specifies that the mote listen to the medium, if it is idle, the protocol waits an additional backoff time. At the end of the backoff time, if the medium is still free, the mote switches to transmit mode and sends the frame. The mote then switches to receiving mode [74]. There is no collision detection, no acknowledgements and no sequence numbers. The protocol will not attempt a retransmission after a failure. The frame size is 21 bytes.

The routing protocol is an implementation of GSP reflecting an extreme case. Figure 27 illustrates the timing of the implementation. The radio is on during 2.5 seconds and with probability p = 1, the radio will be off during the next 2.5 seconds. When the mote receives a packet, it will retransmit it, as long as the radio is on.

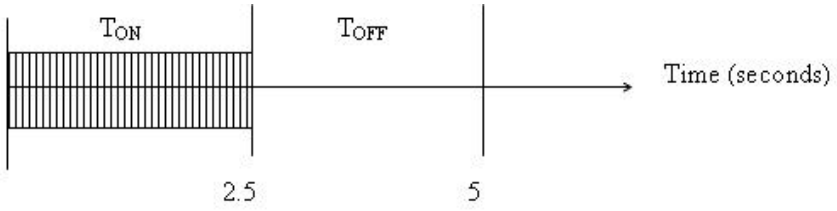

Figure 27. 50% Duty cycle.

The sensors implement a rudimentary application. Each node has one counter that is incremented and sent to the network every 80 milliseconds. If there is a transmission error reported by the MAC layer, the application will resend the same counter. It will only try to send the next value of the counter when it receives a success signal from the MAC layer.

The network was implemented using one sensor and one sink. Energy use measurements were done with an oscilloscope over one resistor of 10.03 ohms. Current was calculated using values given by the oscilloscope. Figure 28 illustrates the measurement circuit. The power supply was 2.971 V DC and the resistor value (10.03 ohms) was chosen to minimize voltage drop in the measurement circuit.

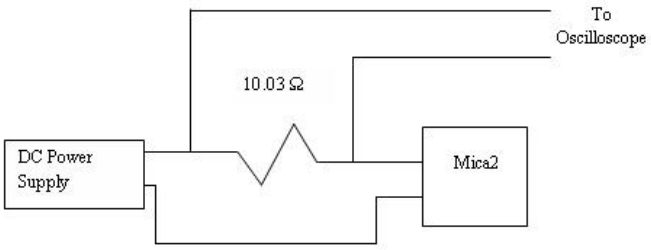

Figure 28. Equipment connection.

The instantaneous voltage drop across the resistor was measured using an HP54600 oscilloscope. Figure 29 shows a waveform displaying the transmission of a frame, the time when the radio was on and time when the radio was off. For measuring current when the sensor was receiving, another sensor running GSP was used during some part of the tests. The second sensor started transmitting so the original sensor received the frames and retransmitted them accordingly. Tests were performed on six different motes independently.

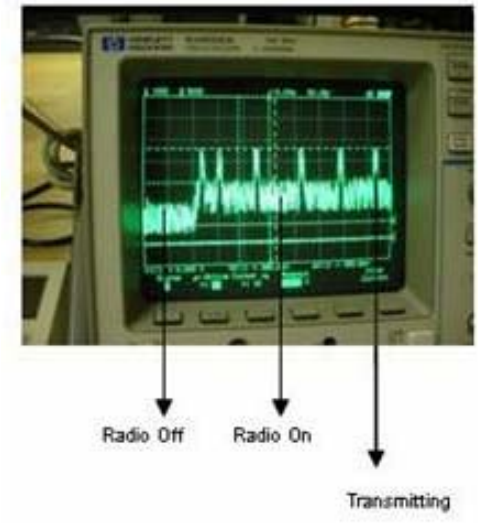

Figure 29. Waveform across the resistor.

3.4.2 Results

Table 11,Table 12 and Table 13 summarize the measurements taken from the six motes in the test. The values were identical when the radio was on but not receiving (listening) and when the radio was on and receiving a frame. Table 11 lists the voltage drop across the resistor. Observe in Figure 29, the waveforms had noise. The measurements obtained the maximum stable voltage for each state.

Table 11. Resistor voltage.

	Radio Off (mV)	Radio On (mV)	Transmitting (mV)
Average	59.22	161.35	309.54
CI 95%	1.82	1.80	1.99

Table 12 lists the current through the 10.03 ohm resistor and hence through the Mica2 mote. The values are calculated from the measured values in Table 11.

Table 12. Current through the circuit.

	Radio Off (mA)	Radio On (mA)	Transmitting (mA)
Average	5.92	16.14	30.95
CI 95%	0.18	0.18	0.20

Table 13 lists the voltage in the transmitting node. The value results from subtracting the resistor voltage from the source voltage.

Table 13. Voltage in transmitting node.

	Radio off (V)	Radio on (V)	Transmitting (V)
Average	2.91	2.81	2.66

Average results helped to find energy used by one sensor, using equation (7):

$$Energy = v(t) * i(t) * t \quad (7)$$

Where v(t) is voltage in the element in Volts, i(t) is current through the circuit in Amperes and t is time in seconds. Equation (8) determines energy use for the Radio Off state.

$$EnergyRadio\ Off = 17.23 * t\ (mjoules) \quad (8)$$

Where t is the time in seconds when the radio was off. The radio was off half of the time, so in one hour this part of the system used 31 joules or 2.96 mAh. Using a similar procedure and equation (7), equation (9) shows the Energy use for Radio On state without transmitting.

$$EnergyRadio\ On = 45.35 * t\ (mjoules) \quad (9)$$

In this case, t is the time during which the radio was on but not transmitting. The sensor was sending one message every 80 msec and the duration of the message is 9.0 msec, so the radio is on without transmitting during 71 msec. During 2.5 seconds for the radio being on, the mote sends a frame and listens for 31.25 times.

$$31.25\ times * 71\ msec = 2.2\ sec. \quad (10)$$

Therefore, the proportion of the Radio On but just listening is

$$2.22 \text{ sec} / 2.5 \text{ sec} = 88.75\ \% \quad (11)$$

There are 1597.5 seconds in one hour (88.75% of half an hour, because of the 50% duty cycle) where the radio is on but not transmitting. Therefore, energy spent listening in one hour is 72.45 joules using equation (9) or 7.16 mAh. Energy use for receiving a bit can be calculated, following the approach presented in [75]. The data rate is 19.2 kbps, so the time for transmitting one bit is 52 μsec. Using this value in equation (9):

$$EnergyRx \text{ per bit} = 2.36\ (\mu joules/bit) \quad (12)$$

Calculating the energy use for the transmission state follows a similar procedure resulting in:

$$Energy\ Tx = 82.33 * t\ (mjoules) \quad (13)$$

Where t is the time in seconds when the radio was transmitting a frame. In one hour, this implementation of GSP consumes 16.67 joules (because of the 50% duty cycle) or 1.74 mAh. Energy per bit can be calculated using equation number (13):

$$Energy\ Tx \text{ per bit} = 4.28(\ \mu joules/\ bit) \quad (14)$$

Table 14 shows the model found in this work for Mica2 platform transmitting with 5dBm, at 903 MHz and 19.2 kbps.

Table 14. Energy use summary.

State	Radio Off (mjoules)	Radio On (mjoules)	Transmitting (mjoules)
Equation	17.23 * t	45.35 * t	82.33 * t

Total Energy use of one mote can be calculated using equation (15).

$$E(t) = \sum_{i=1}^{3} S_i(t) \quad (15)$$

Where $S_i(t)$ is the equation for energy use in each state of the mote. S_1 is for Radio Off, S_2 for Radio On (listening or receiving) and S_3 for Transmitting. Total energy use in one hour in this work is 120.12 joules. It is also possible to compute energy spent by the microcontroller executing one instruction, considering the completion of roughly one instruction every clock cycle. With an 8 MHz clock, one instruction takes 125 ns. Using the value in S_1 (Radio Off, just the microcontroller working), 17.23 * 125 ns = 2.15 njoules.

3.4.3 Conclusions from this study

The study measured energy use of a Mica2 sensor while employing the GSP protocol. Results for specific experiment parameters can be a basis for obtaining the total use of one sensor in a wireless sensor network. The radio model used in [10] indicates reception and listening having almost the same energy use which is around 73% of energy needed for transmission. The model found here shows that listening and receiving for Mica2 have the same energy use, which is 55% of the energy required for transmission. Future work will evaluate the impact of these results for GSP. Additionally, extending the results will allow comparing GSP with other protocols, providing all times the radio spends in different states.

The measurements show that for the Mica2 platform, CPU energy use in the active state is three orders of magnitude smaller than energy spent in transmitting or receiving a bit. Therefore, in this case the microcontroller can execute approximately one thousand instructions to spend the same energy required for one bit communication. Compared to the analytical model presented in [6], energy use is two orders of magnitude greater after accounting for the energy used by the microcontroller. Measurements also show that the energy used in transmitting one bit is almost twice the energy used in receiving one bit.

Experiment settings in this work allowed for the creation of a precise set of energy use measurements. The use of the oscilloscope enabled the measurement of short duration signals in different frequencies, as the ones expected when the message frequency and message duration are less than 16 msec. The measurements described will enable the creation of more accurate simulations for GSP. However, additional measurements at various frequencies and transmission power levels and varying processing loads in the microcontroller will further improve the simulations.

4.0 MAC GSP

GSP is a candidate routing protocol for Wireless Sensor Networks because of its simplicity and low overhead. GSP may find use in applications with low traffic and flexible delay requirements, such as crop and temperature monitoring. Other application that may benefit from the use of GSP is monitoring lights and the number of people in classrooms, in order to turn off the systems when a room is empty, helping to conserve energy. Applications may rarely transmit a packet, thus the traffic generated is low. The section presents results from research based on low conditions. Figures show results with 90% confidence levels, generated by averaging 40 runs, 200 packets in each run. Capture Effect was included in the analysis, but there were no consistent differences for consecutive values of β, the capture probability, at 90% confidence level. Thus, figures consider β=0.7 as example of all data obtained, unless otherwise noted.

4.1 RESULTS UNTIL ALL PACKETS ARE TERMINATED IN THE NETWORK

Previous work with MAC protocols [16] and capture effect [17] analyzed the results of different metrics with data collected until the sink received the first data packet. However, a problem with MACGSP1 and MACGSP2 is they do not eliminate all duplicates traversing the network. The following results were collected from simulations run until all packets are terminated in the network. The results include only p_{gsp} as a mechanism for limiting the amount of duplicates (thus

no additional overhead is required). All the metrics have the same definition as in the previous section, except Delay, which now reflects the duration of the process until all packets exit the network.

Delay = Average number of gossip periods from the time a packet is transmitted by the source to the time all duplicates are terminated in the network.

Figure 30 illustrates packet reception probability for MACGSP1, MACGSP2 and REALGSP in 100 nodes square grids, with source in the upper corner and sink in the center. Note low values of p_{gsp} do not eliminate most duplicates in the network and the packet traverses the network continuously; thus Figure 30 shows results just for p_{gsp} >0.3. The situation prevents the use of the protocols in WSNs applications unless the implementation includes additional mechanisms to limit the number of duplicate packets. The trend and relative relationship of the curves in Figure 30 are the same as the relationship shown for packet reception probability found until the sink receives the first packet (Figure 14). Results have observable differences at 90% confidence levels when $p_{gsp} < 0.7$.

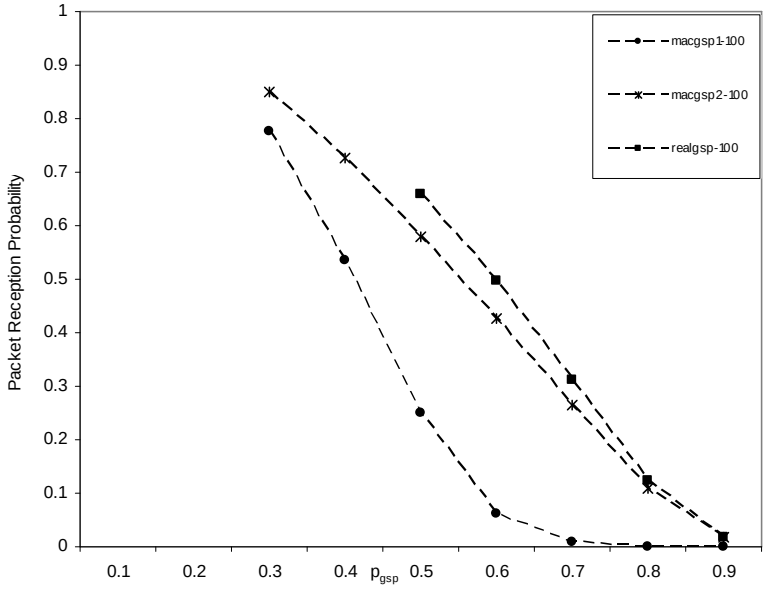

Figure 30. Packet reception probability, 100 nodes.

Figure 31 shows the number of duplicates received by the source as a measure of the number of packets traversing the network. The number of duplicates increases exponentially as p_{gsp} decreases, but the relative behavior of the three protocols remains the same as in [16], MACGSP1 has the fewest duplicates, followed by MACGSP2 and REALGSP.

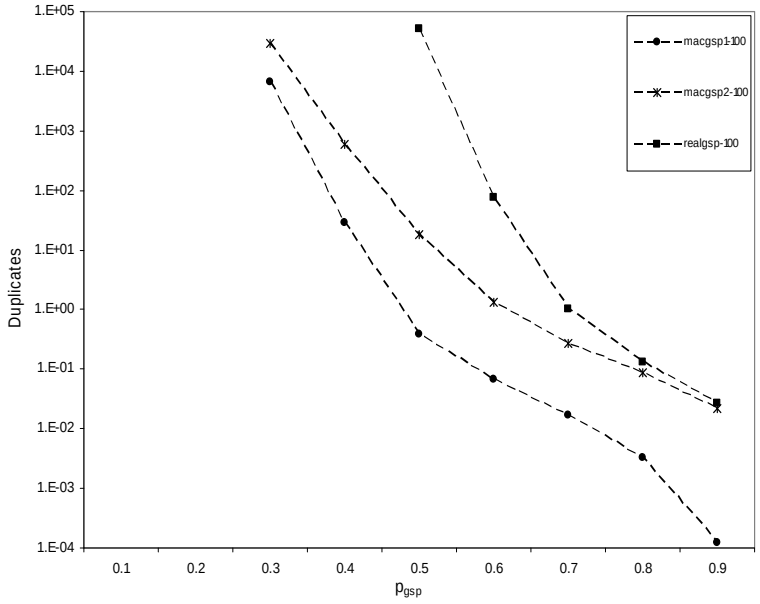

Figure 31. Duplicates, 100 nodes.

Figure 32 shows the time required to eliminate all duplicates in the network, which also increases exponentially as p_{gsp} decreases. Note REALGSP presents the highest delay (close to one million TPs for $p_{gsp} = 0.5$), followed by the two other protocols which have smaller numbers of duplicates. Additionally, MACGSP2 and REALGSP have no observable differences at 90% confidence levels for $p_{gsp} \geq 0.8$.

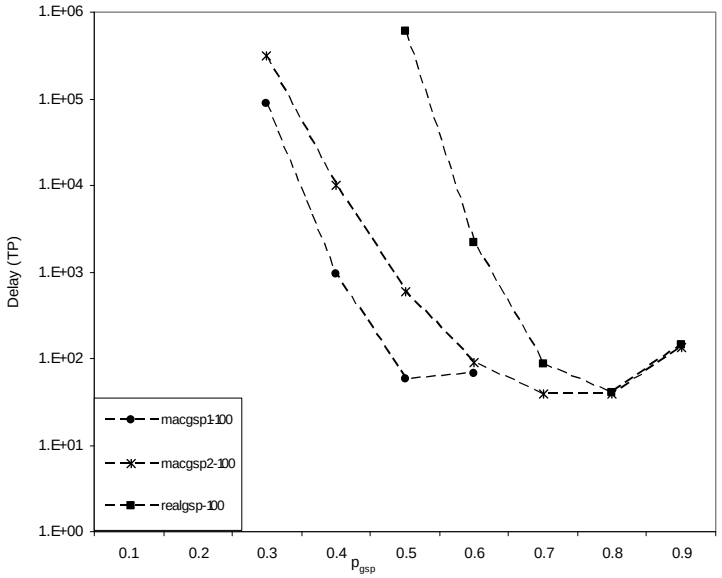

Figure 32. Delay, 100 nodes.

Figure 33 presents total energy use for sending 200 packets from source to destination. The exponential trend is clear again and REALGSP shows the highest energy use, followed by MACGSP2 and MACGSP1. Maximum value is in the order of millions of Joules.

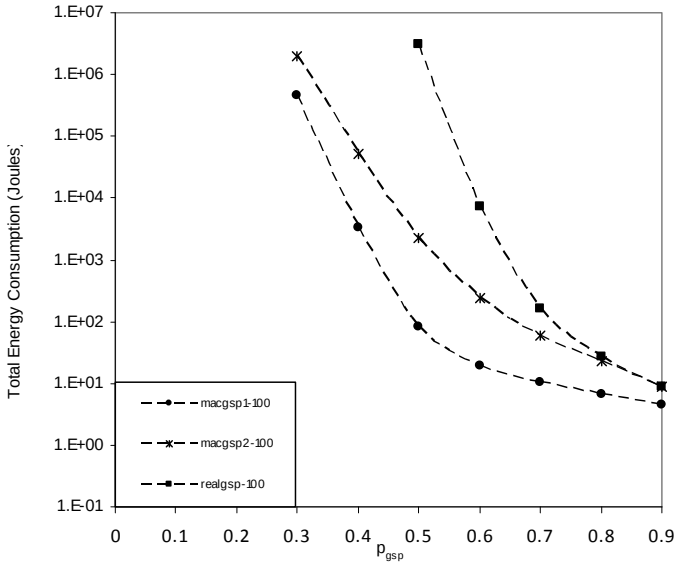

Figure 33. Total energy use, 100 nodes.

Figures 34, 35 and 36 display the different components of all energy use. Figure 34 presents energy spent in the transmission state by all nodes in the network. Note values are in the same order of magnitude as total energy use and REALGSP presents higher values than the other two protocols.

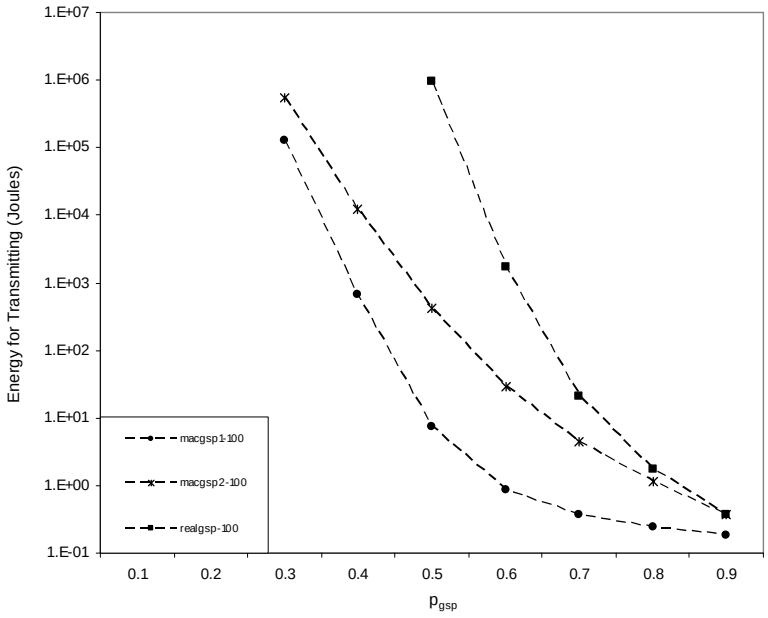

Figure 34. Energy spent in transmission, 100 nodes.

Figure 35 displays energy used by all nodes in reception state. REALGSP exhibits the highest values, like in Figure 34. Note values are in the same order of magnitude as energy for transmitting, opposite to results in [16]. Consider the ideal situation of forwarding a packet in a network without losses: in that case, the number of transmissions is the same as the number of receptions. According to the energy model presented in Table 10, the ideal situation of obtaining the same number of transmissions and receptions means energy employed for transmission is roughly twice as energy employed in the reception state. However, results show energy used for transmitting is at most 50% of energy used in reception, thus idle listening still occurs, but the

number of transmissions is higher than [16] which is due to the higher number of duplicates relayed in the network.

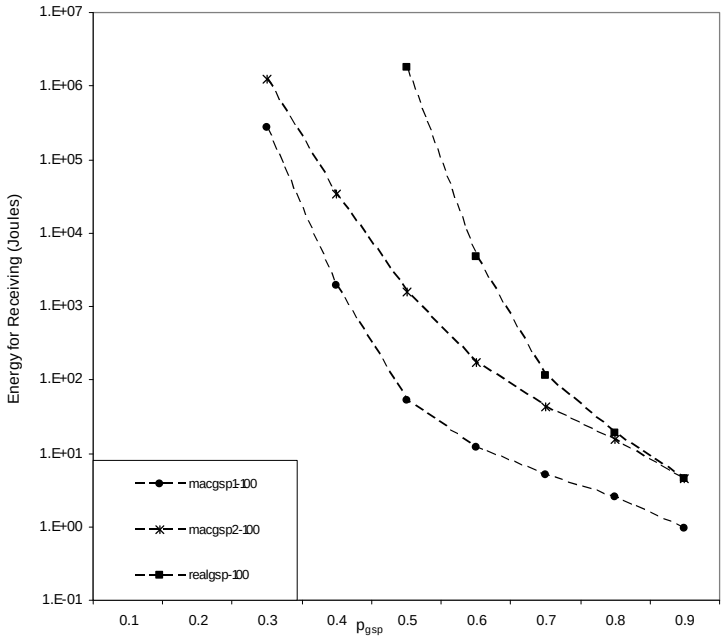

Figure 35. Energy spent in reception, 100 nodes.

The trend between curves is similar to Figure 33, exhibiting the same relative behavior of each protocol compared to the other two: REALGSP has the highest energy use, followed by MACGSP2 and MACGSP1. As before, REALGSP and MACGSP2 show no observable differences at 90% confidence levels for values of $p_{gsp} > 0.8$.

Figure 36 shows energy used in the radio off state, which is one order of magnitude smaller than energy employed in any of the other two states. The behavior is the opposite of findings in previous work (Figure 19) where MACGSP1 and 2 put more nodes to sleep than REALGSP for lower values of p_{gsp}. Figure 26 shows REALGSP makes more nodes sleep in all the range of p_{gsp}, followed by MACGSP2 and 1. Nonetheless, values measured until the last packet exits the network are up to five orders of magnitude higher than those in [16].

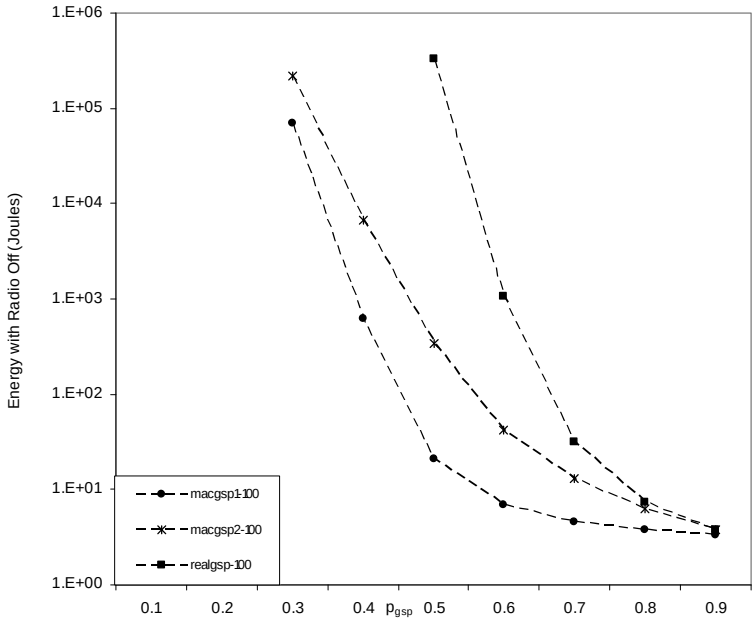

Figure 36. Energy spent with the radio off, 100 nodes.

The number of duplicates and the time that they remained in the network motivated an investigation into the protocols in order to eliminate duplicates earlier in the network without increasing overhead.

4.2 QUIESCENT PERIODS

Nodes employing MACGSP1 and 2 use Quiescent Periods immediately after transmission to avoid receiving a duplicate packet by turning off the radio. Increasing the duration of these quiescent periods can eliminate the duplicates traversing through the network. In MACGSP6, the duration of the Gossip Period (GP) increases from 1 TP (the time it takes to transmit one packet) to 10 TP, and source and relays sleep during the same amount of time after transmission. Figure 37 shows delay until all duplicates exit the network, using quiescent periods of 10 GP and making every GP=10*TP. Experiments with capture probability β from 0.5 to 1.0, showed no

consistent differences at 90% confidence level for consecutive values of β, thus all figures use β=0.7, similar to the results in [16].

Figure 37 illustrates delay for all network sizes and the maximum delay is close to 200 TP for 100 nodes. Note delay presented in Figure 32 is three orders of magnitude higher for the same network size. Additionally, because increasing p_{gsp} turns off more nodes in the network, which in turn reduces duplicates, higher values of p_{gsp} decrease the time it takes until the last packet exits the network.

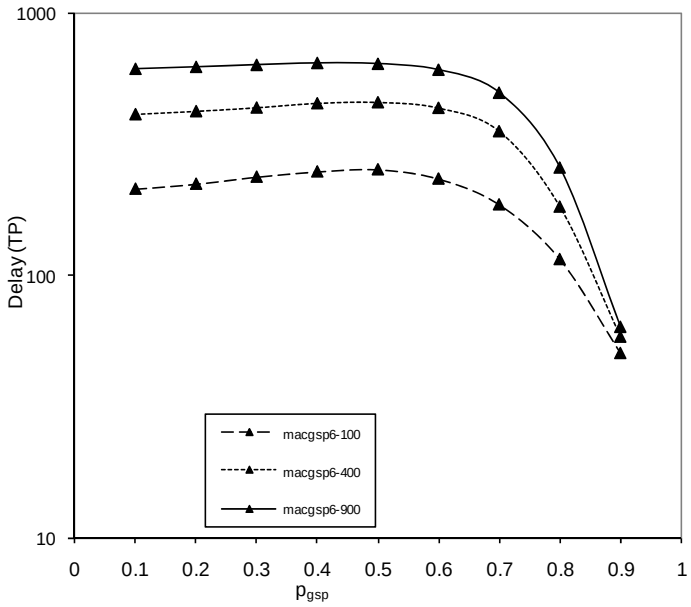

Figure 37. Delay for MACGSP6 until all packets terminate in the network.

However, all previously reported results measured delay until the sink received the first packet. In order to compare these results with the previous work [16], Figure 38 shows Delay until the sink receives the first packet as a function of p_{gsp} for MACGSP6 and GSP implemented without any MAC protocol (referred in the figure as REALGSP) using a gossip period of 10 TP, same as MACGSP6. Note that using a gossip period one order of magnitude bigger than the original increases the delay in delivering a packet by the same order of magnitude. The result is expected, since nodes are actively transmitting only at the beginning of each GP. However, delay

values are close to the minimum number of TPs required for the particular configuration, which is 71 TP for 100 nodes, 171 TP for 400 nodes and 271 TP for 900 nodes. Again, there are no observable differences at 90% confidence level for delay in MACGSP6 and REALGSP.

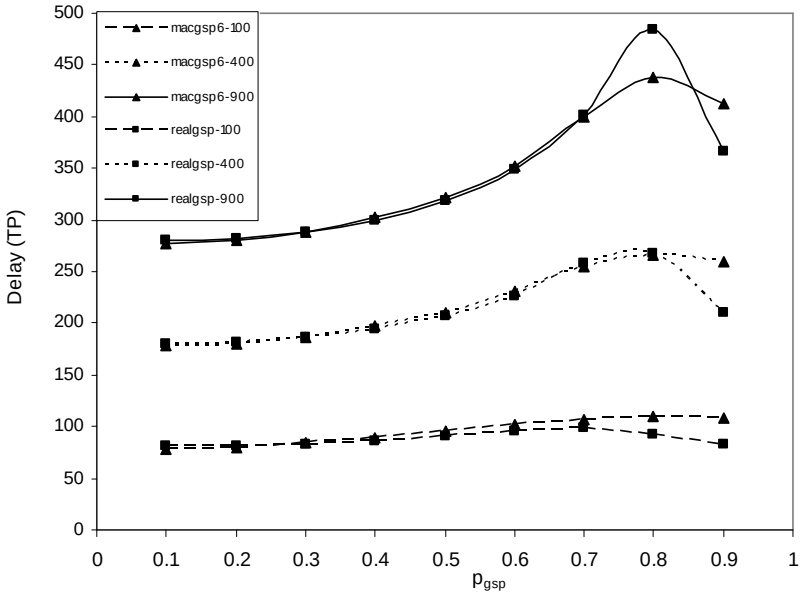

Figure 38. Delay until first packet received by sink, MACGSP6 and REALGSP.

Quiescent periods, in addition to eliminating duplicate packets, also avoid collisions. Figure 39 show the number of collisions in the network as a function of p_{gsp} for MACGSP6 and REALGSP. Results count a collision after considering the capture effect; that is, one collision exists whenever two or more packets arrive simultaneously to a node and all packets are lost according to the capture probability. If the probability allows for reception of one packet, there is no collision. Figure 39 shows MACGSP6 decreases GSP collisions up to 50% for lower values of p_{gsp}. Nonetheless, GSP with values of $p_{gsp} > 0.5$ create a smaller number of collisions than MACGSP6. Analysis presented in sections 4.3 and 4.4 help to clarify this behavior.

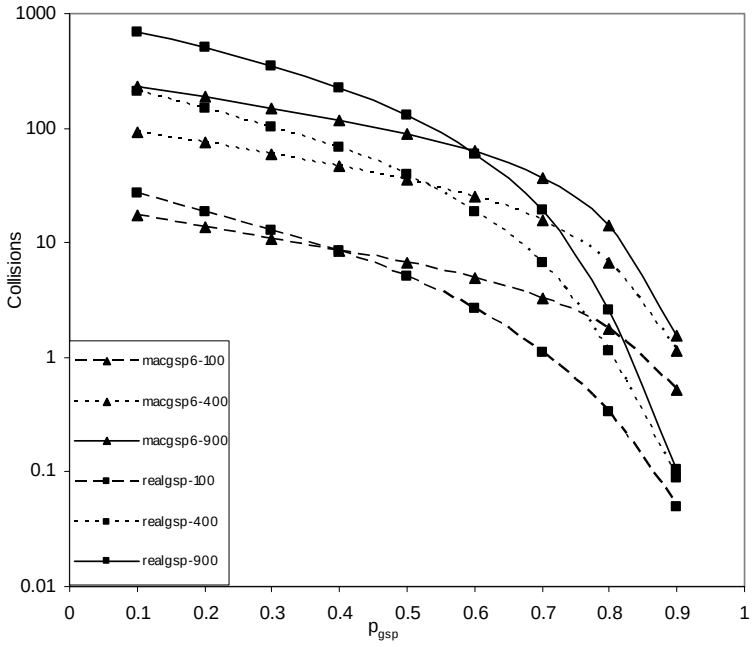

Figure 39. Collisions in MACGSP6 and GSP.

MACGSP6 uses quiescent periods of 10 GP. Values of quiescent periods of 3 and 5 GPs eliminated all duplicates after long delay. A value of 10 GP eliminated all duplicates in the network with shorter delay. Higher values of quiescent periods do not generate improvements in the metrics.

4.3 EFFECT OF DUTY CYCLE

Quiescent periods improved energy usage, decrease the number of duplicates in the network and allow for implementation of Duty cycles to save energy. The following figures show the difference in energy use with GSP as an example of 100% duty cycle and MACGSP6, whose duty cycle is 10%, for square grid topologies of 100, 400 and 900 nodes. Again, the experiments

employ one source located in the corner of the grid and the sink is located in the center. Figures present all metrics until the sink receives the first packet.

Figure 40 shows MACGSP6 requires 40% of the energy needed by GSP for p_{gsp} values smaller than 0.6. Note total energy use for MACGSP6 remains almost constant for p_{gsp} values smaller than 0.7, while REALGSP shows a decreasing trend depending on p_{gsp}. The decreasing trend finally overcomes MACGSP6 and REALGSP consumes less energy at the end of the curve.

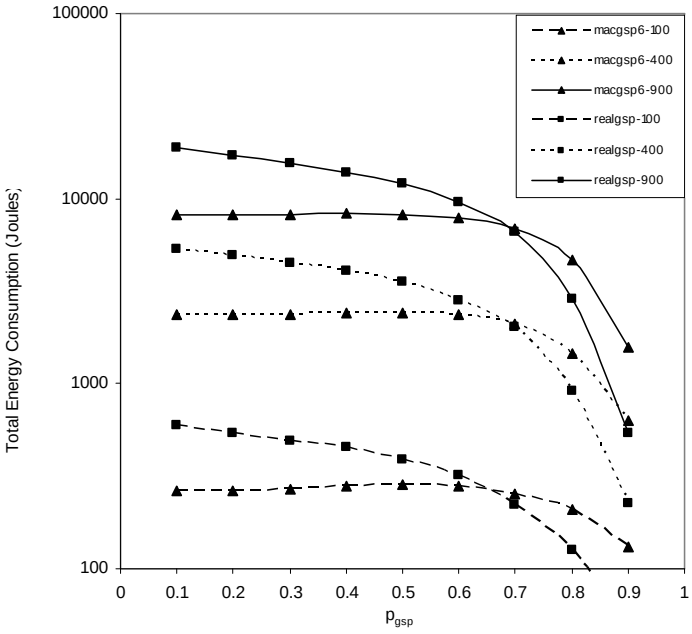

Figure 40. Total energy use, duty cycle.

The reasons for protocol behavior become clearer by analyzing separate components of energy use. Figure 41 displays energy used by all nodes in the transmitting state for sending 200 packets. The transmitting state for both REALGSP and MACGSP6 protocols use approximately 1% of the total energy. However, energy used by MACGSP6 is as low as 50% of energy used by GSP in the transmitting state. Note a node using GSP may receive a duplicate packet it just

transmitted and, according to protocol rules, the node must forward the packet again. However, MACGSP6 does not relay duplicates received immediately after transmission, thus saving energy in the transmitting state.

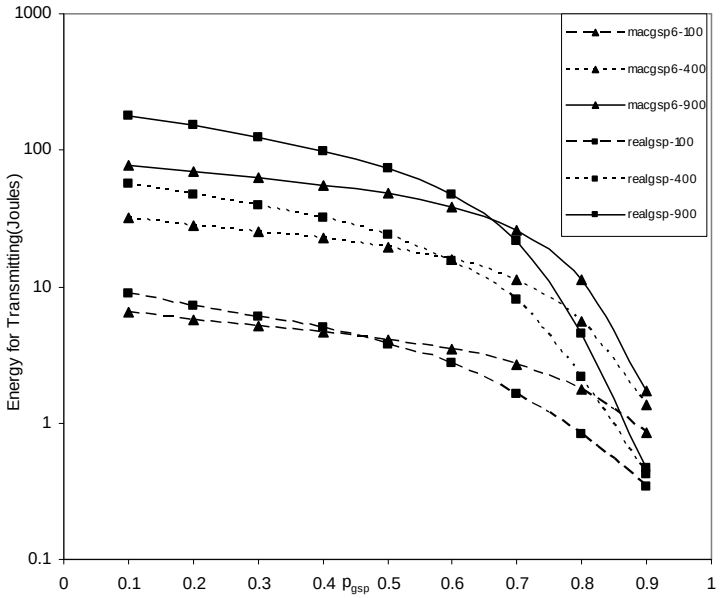

Figure 41. Energy for transmitting, duty cycle.

Figure 39 showed a decreasing number of collisions for REALGSP when $p_{gsp} > 0.5$. Figure 41 shows the reason for this behavior: REALGSP employs less transmitting states for those values of p_{gsp}, hence, there are less transmissions and the number of collisions decreases.

Figure 42 presents energy used while in the reception state. MACGSP6 uses slightly more energy in this state than in transmission (receiving is around 8% of the total). However, transmission and receiving states are of the same order of magnitude. REALGSP has the same behavior as in [16], 95% of total energy is used in the reception state. Because energy used for receiving by MACGSP6 is one order of magnitude smaller than REALGSP, the amount of energy wasted in idle listening by MACGSP6 is considerable smaller, showing an improvement from previous work [16]. Note both protocols have a slowly decreasing trend as p_{gsp} increases,

showing there are less nodes in the receiving state. Nonetheless, the trend is more noticeable for REALGSP, since energy spent in the receiving state is the main component of the total.

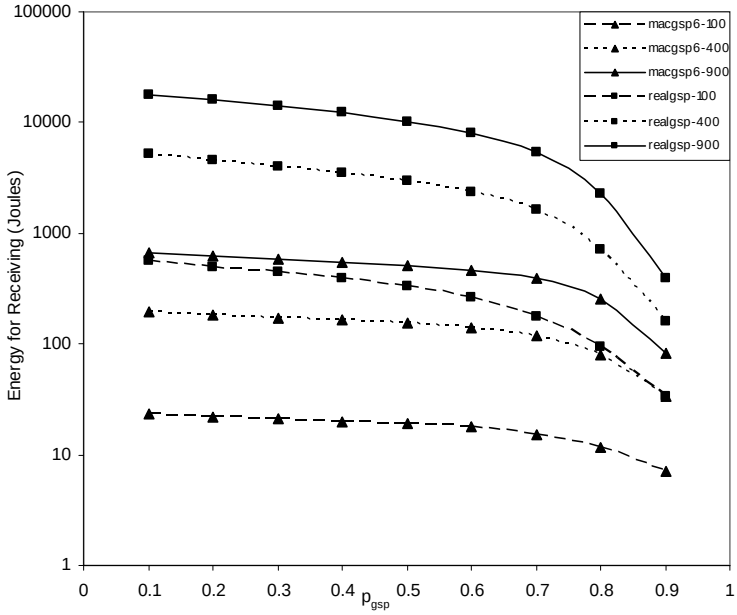

Figure 42. Energy for receiving, duty cycle.

Figure 43 shows energy used in the radio off state by all nodes in the network. REALGSP energy use in the radio off state is one order of magnitude smaller than MACGSP6, which in turn corresponds to roughly 4% of the total. Thus, REALGSP makes less nodes sleep during the whole communication process. Similar to the results in [16], energy in the radio off state for REALGSP increases and then decreases (see Figure 19); the situation is due to the metric definition: sum of all energy used by all nodes in off state until the sink receives the packet or until all packets exit the network. Higher values of p_{gsp} make simulation stop early because of packets dropped. Note MACGSP6 uses up to 92% of energy putting nodes to sleep in the network, which is valuable for WSNs with low traffic requirements since nodes can collect data during this sleeping time and the network spends less than 10% of its total energy in the

communication process. Energy use is an indirect measure of network lifetime, thus MACGSP6 can increase lifetime.

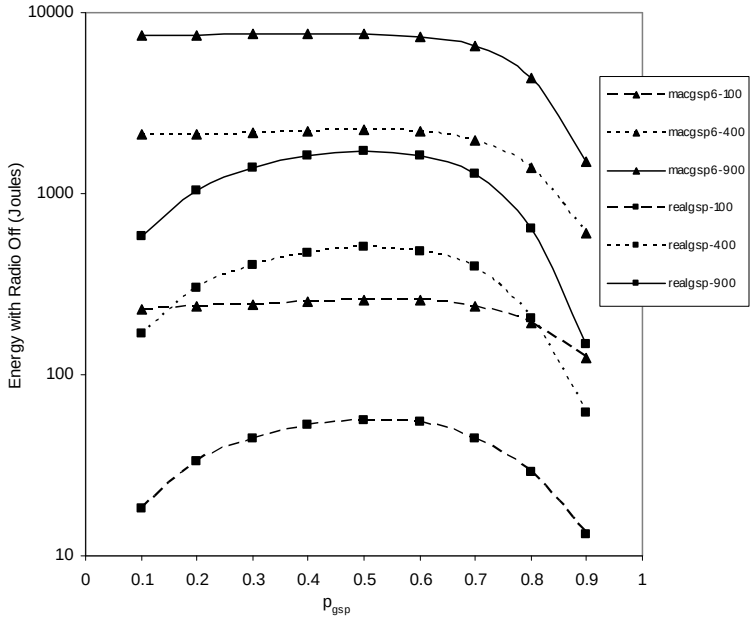

Figure 43. Energy with radio off, duty cycle.

4.4 OVERHEARING FOR ACKNOWLEDGMENTS

Overhearing a retransmission of the packet just transmitted is a way of creating implicit ACKs without adding protocol overhead. MACGSP6 implements this mechanism and this section shows the comparison with REALGSP, the protocol with highest packet reception probability from the preliminary work [16]. The rules for MACGSP6 are the following:

After transmission, the node stays on for the rest of the active part of the duty cycle and goes to sleep for the rest of the GP.

At the beginning of next GP, a node that transmitted in the previous GP verifies the medium during the time it takes to receive the preamble of a packet. If the node hears the

preamble, it goes to sleep immediately during the rest of the current GP and during all time required by the quiescent period. If the node hears nothing, it will retransmit the packet in the next GP.

After relaying the packet for a second time, the node goes to sleep according to the quiescent period.

A node that was awake and did not receive during the time it takes to receive the preamble of a packet goes to sleep for the rest of the current GP and decides its state in the beginning of the next GP according to p_{gsp}.

A node that was sleeping at the beginning of previous GP because of GSP rules must wake up in the next GP.

Every packet received (other than the one just sent) is relayed in next TP.

Figure 44 illustrates a time sequence for a node executing the rules of MACGSP6. For brevity, the example illustration uses a quiescent period of 3 Gossip Periods. The node wakes up at the beginning of the corresponding GP to transmit and listens for a retransmission (implicit ACK); if the node does not hear a preamble, it retransmits and sleeps after the second retransmission.

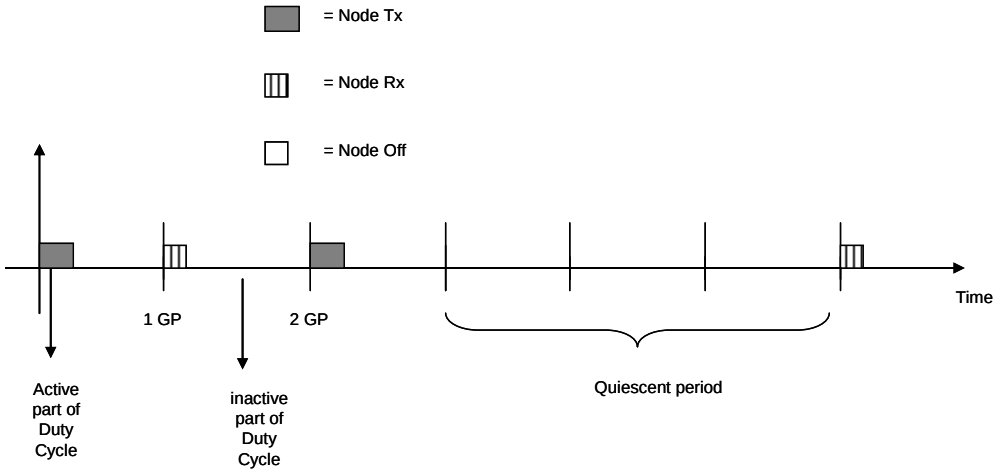

Figure 44. One node running MACGSP6.

Figures 45 to 49 show results evaluated until the first packet is received by the sink for square grid topologies of 100, 400 and 900 nodes with the source located in one corner and the sink in the center of the network, allowing comparison with protocols from the preliminary work [16].

Figure 45 illustrates packet reception probability for MACGSP6 and REALGSP. Note MACGSP6 improves packet reception probability of REALGSP up to 20% without adding overhead, displacing the useful values of p_{gsp} to the right of the curve. Values of p_{gsp} between 0.3 and 0.4 provided a good trade-off between energy use and packet reception probability [16]. Figure 45, together with energy results from the previous section (Figure 40) show MACGSP6 allows packet reception probability of up to 75% with $p_{gsp} = 0.6$ with essentially constant energy use for the protocol.

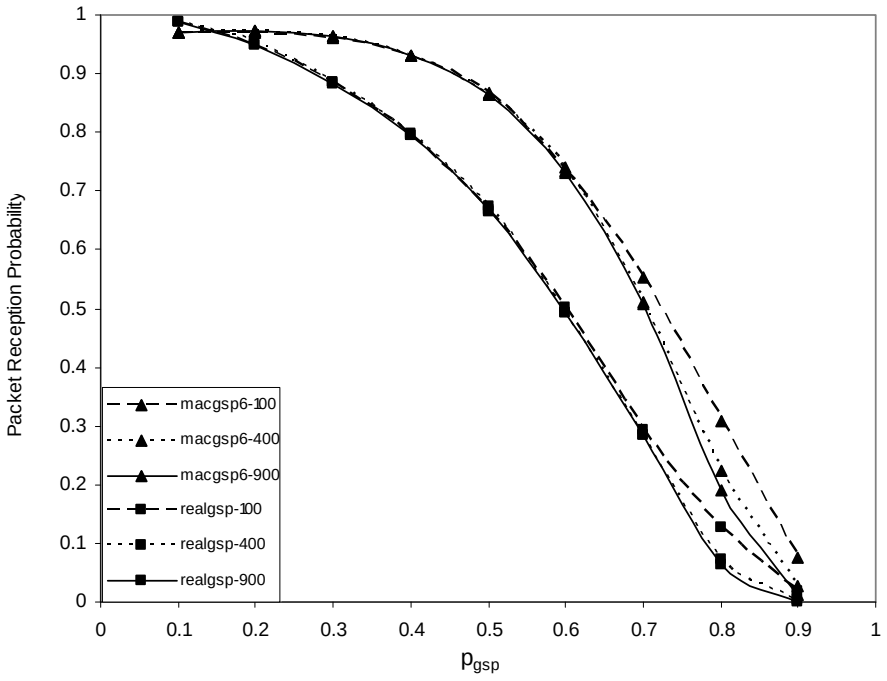

Figure 45. Packet reception probability, MACGSP6 and REALGSP.

REALGSP in Figure 40 appears to have smaller energy use for higher values of p_{gsp}, however it is delivering fewer packets to the sink. Figures 46 to 49 help in the analysis by showing the energy used per packet successfully received by the sink.

Figure 46 presents total energy use for successfully forwarding one packet from source to sink. Energy used by MACGSP6 is 45% of energy employed by REALGSP. Results have increasing trends at the end of the curve since the sink receives very few packets, so energy per packet increases. All values are different at 90% confidence levels.

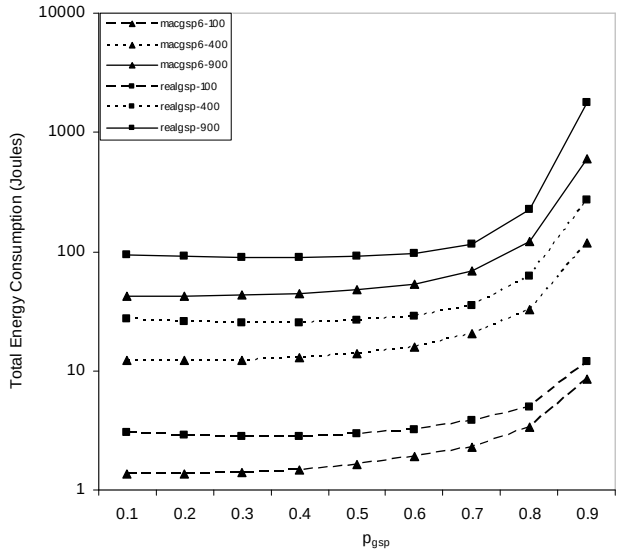

Figure 46. Total energy use per successfully received packet.

Figure 47 presents energy used in the transmitting state for successfully receiving a packet. MACGSP6 employs 20% less energy than REALGSP in the smallest network and around 50% less in the bigger networks.

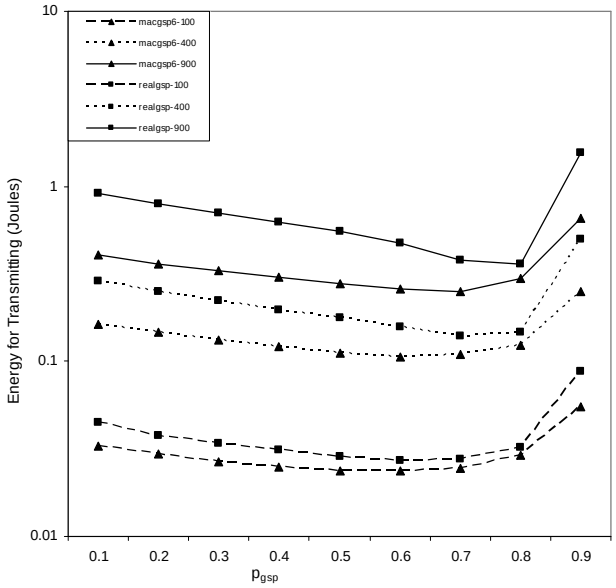

Figure 47. Energy used in transmission state per successfully received packet.

Figure 48 shows energy used in the receiving state to successfully forward a packet in the network. MACGSP6 uses one order of magnitude less energy than REALGSP for all values of p_{gsp}. Compared to the results in Figure 42, the curves show an opposite deflection; the trend of energy use is slowly increasing, with steeper increase at the end given by the smaller amount of packets delivered.

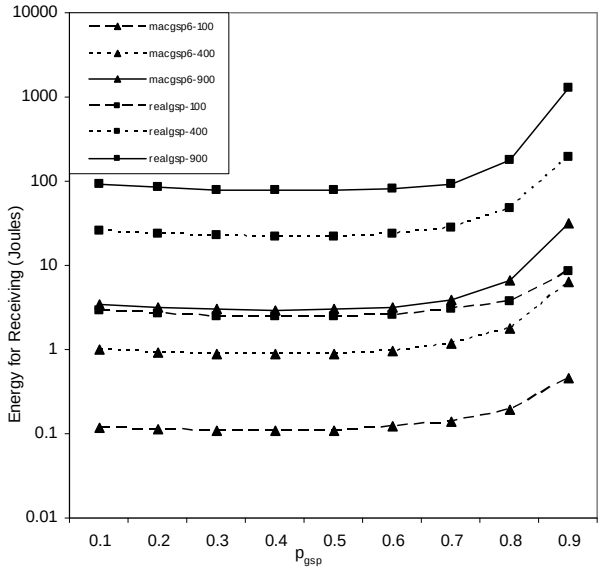

Figure 48. Energy used in receiving state per successfully received packet.

Figure 49 presents energy used in the off state by all nodes in the network. REALGSP decreases this component of energy use up to one order of magnitude compared to MACGSP6, therefore the number of nodes sleeping in a network using REALGSP is very small compared to MACGSP6. When p_{gsp} increases, the difference between both protocols decreases, but the differences are observable at 90% confidence levels for all values except p_{gsp}=0.9 with 900 nodes. Notice use of energy with the radio off in MACGSP6 is in the same order of magnitude as total energy use per packet, following the behavior given by the total energy use with all the packets considered.

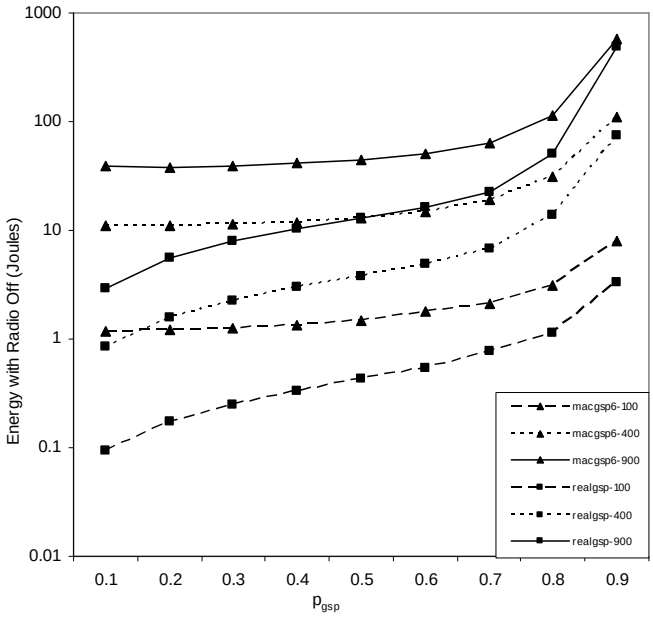

Figure 49. Energy in off state per successfully received packet.

4.5 BEHAVIOR WITH DIFFERENT TOPOLOGIES

To compare the advantages and disadvantages of MACGSP6, this section analyzes the behavior of the protocol over the various physical topologies proposed in [3]. As before, the source is located in one corner and the sink located at the center of the network. Figures show results with 90% confidence levels computed from the average of 40 runs with 200 packets in each run. The study considered capture probabilities of β=0.5 to β=1.0 (perfect capture effect). In order to test Hypothesis 3, results show all metrics until the sink receives the first data packet and with a perfect capture effect.

4.5.1 Linear

Linear topologies present an extreme case for testing the GSP family of protocols, since there is only one possible route from source to destination. Because nodes sleep according to the protocol rules, the network can be disconnected at any time. Packet reception probability in a linear network can be modeled assuming only one source located at the beginning of the network. Figure 50 presents one example linear topology. The sink is always on, listening to the medium, since it has unlimited energy supply.

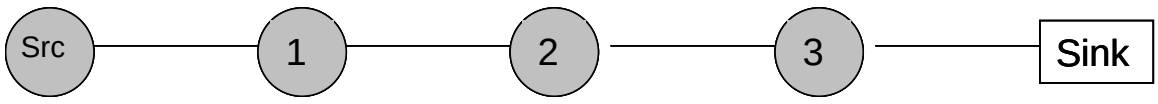

Figure 50. Example of linear topology.

The behavior of a node using REALGSP can be modeled using state transition diagrams. Figure 51 shows the diagram for a node when no packet is received. State 0 corresponds to radio off and state 1 means radio on, ready to receive. Recall p_{gsp} is the probability of turning off the radio and $(1\text{-}p_{gsp})$ is the probability of each node to have its radio on.

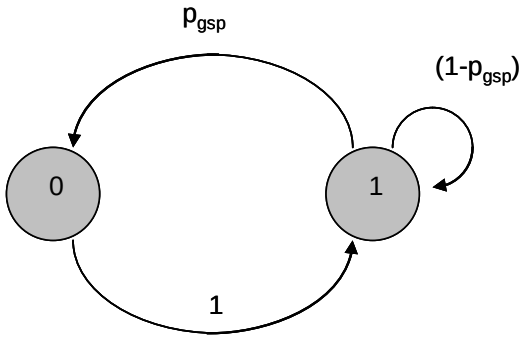

Figure 51. State diagram for a node working with GSP.

Figure 52 shows vector a, the initial probability distribution of the states, and P, the transition matrix for this state diagram. The first component of vector a, referred to as $a_{[0]}$, is the initial probability of having the radio off and the second component, $a_{[1]}$ is the initial probability of radio on.

$$a = [p_{gsp} \quad (1\text{-}p_{gsp})] \qquad P = \begin{bmatrix} 0 & 1 \\ p_{gsp} & (1\text{-}p_{gsp}) \end{bmatrix}$$

Figure 52. Initial probabilities and transition matrix for GSP.

The transition matrix reaches a steady state and it is possible to find the proportion of time that a node stays on and off, according to the values of p_{gsp}. Figure 53 illustrates the situation.

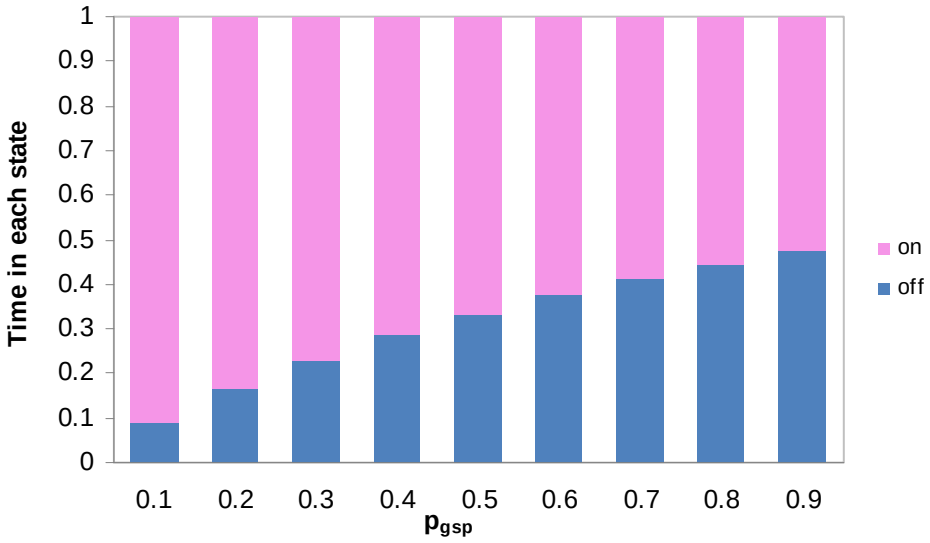

Figure 53. Fraction of time each node spends in each state.

Now, the behavior of the linear network in Figure 50 is as follows. At t=0, every node has an initial state given by vector a, except for the sink (which is always in state 1) and the source which transmits always at t=0. The probability of node 1 receiving the packet is just the probability of the node having the radio on at t=0. In other words:

$$p_{rx1,0} = p \text{ (node 1 on at t=0)} = (1-p_{gsp}) \quad (16)$$

Since every node receiving the packet must forward it, the probability of node 2 receiving the packet is just the probability of node 1 being on at t=0 and the probability of node 2 being on at t=1. This latter probability can be found by multiplying a by P and taking the second component, corresponding to radio on. Therefore:

$$p_{rx2,1} = P \text{ (node 1 on at t=0)} * p \text{ (node 2 on at t=1)}$$

$$p_{rx2,1} = (1-p_{gsp}) * [p_{gsp} + (1-p_{gsp})^2]$$

Similarly, the probability of receiving the packet at node 3 is given by the probability of node 3 staying on at t=2, which can be found using $aP^2_{[1]}$. Thus:

$$p_{rx3,2} = P \text{ (node 1 on at t=0)} * P \text{ (node 2 on at t=1)} * P\text{(node 3 on at t=2)}$$

$$p_{rx3,2} = (1- p_{gsp}) * [p_{gsp} + (1-p_{gsp})^2] * \{2*[p_{gsp} * (1-p_{gsp})] + (1-p_{gsp})^3\}$$

When node 3 receives the packet, it forwards it to the sink with probability 1.0, thus $p_{rx3,2}$ is the probability of receiving the first packet in the destination. Equation 17 generalizes the situation of a linear network with n intermediate nodes. That is the product of the second components of the vectors given by aP^n.

$$p_{rx\,n,n-1} = \prod_{i=0}^{n-1} aP^{i}{}_{[1]} \tag{17}$$

Note equation 17 shows the first packet is following the shortest path possible, hence, this expression is also the *probability of receiving the packet through the best route*. Note n corresponds to the number of nodes and more specifically to the minimum number of hops in the network:

$$\text{Minimum number of hops} = n + 1$$

Equations to find $p_{rx1,0}$, $p_{rx2,1}$ and $p_{rx3,2}$ show the exponents increase with the number of hops considered, which decreases packet reception probability.

Additionally, every intermediate node may have created duplicates when they forwarded the first data packet. Figure 54 introduces state number 2 (node transmitting), and illustrates the influence of p_{rx}, the reception probability, in the behavior of each node.

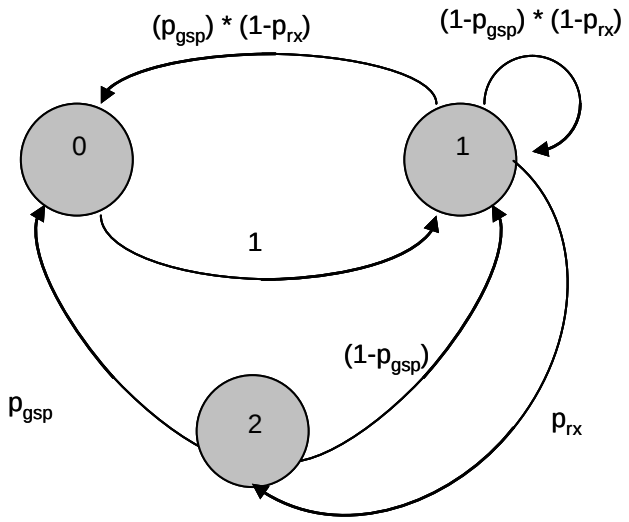

Figure 54. State diagram for a node including transmission state.

The sink is the first node leaving state number 2. According to protocol rules, it can turn its radio off, with p_{gsp}, or stay on with (1- p_{gsp}). Therefore, the probability of the source receiving one duplicate immediately after transmission is:

$$p_{rxsource,1} = p_{rx1,0} * p_{sourceon,1} = \text{p(node 1 on at t=0) * p (source on at t=1)}$$

or, equivalently, using equation (16)

$$p_{rxsource,1} = (1- p_{gsp}) * (1- p_{gsp})$$

Next, the source forwards this duplicate in the same way as the first packet. However, all other nodes in the network can create duplicates and the probability of receiving one packet correctly in the sink, assuming m duplicates, is then:

p_{rxsink} = p(rx1st packet at node 3) + p(rx duplicate 1 at node 3) + ... + p(rx duplicate m at node 3)

Equation 18 presents a general expression, for n nodes and m duplicates in the network, arriving at node n at different times t_i, including equation 17:

$$p_{rx\,\mathrm{sin}\,k} = p_{rx\,n,n-1} + \sum_{i=1}^{m} p(rx\ duplicate_{\ i}\ n, t_i\ ) \qquad (18)$$

Equation 18 is also the probability of receiving the packet through the shortest route or through any of the other routes. The procedure can be extended to analyze the situation when nodes have several neighbors. Assuming the source has h neighbors, the probability of dropping the packet from the source in the first hop is

$$p_{drop,0} = p(\text{all neighbors sleeping at } t=0)$$

$$p_{drop,0} = (p_{gsp})^h, \qquad \text{thus}$$

$$p_{success,0} = 1 - (p_{gsp})^h$$

Consequently, having a higher number of neighbors increases the probability of receiving the packet in the next hop. The chances of success also improve by decreasing p_{gsp}, as shown in all results starting from the previous work [16].

Now, consider what happens at the sink. Equation 19 presents the probability of dropping the packet at node n through all the routes, using equation 18.

$$p_{dropping\ n} = 1 - p_{rx\ \sin k} = 1 - \left[p_{rx\ n,n-1} + \sum_{i=1}^{m} p(rx\ duplicate_{\ i}\ n, t_i\ ) \right] \tag{19}$$

However, if there are several neighbors, in order for the packet not to arrive to the sink all neighbors have to drop the packet and all its duplicates. Hence, equation 20 presents the probability of dropping the packet when the sink has h neighbors.

$$p_{dropping\ } = (1 - p_{rx\ \sin k})^h = \left\{ 1 - \left[p_{rx\ n,n-1} + \sum_{i=1}^{m} p(rx\ duplicate_{\ i}\ n, t_i\ ) \right] \right\}^h \tag{20}$$

Equation 20 shows it is beneficial for REALGSP to have as many neighbors as possible, in order to decrease the probability of dropping packets. Note the equations do not consider interference of one neighbor to each other, so they are valid as long as a perfect capture effect exists.

The model can be used to cross-validate the simulations performed in this work. Consider the case of a linear network with only three nodes: Source, sink and intermediate node. The

probability of reception in the sink is the same as the probability of receiving the packet in the first hop. Table 15 compares analytical results and simulation results for this network, showing the values agree with equation 16. Note the particular topology does not include duplicates, since there are only two possibilities: first node receives the packet and relays it immediately so the sink receives it, or first node does not receive anything.

Table 15. Packet reception probability in the first hop, REALGSP.

p_{gsp}	Analytical	Experimental	
	Value	Value	Std Dev
0.1	0.9	0.898	0.023
0.2	0.8	0.806	0.032
0.3	0.7	0.703	0.031
0.4	0.6	0.597	0.039
0.5	0.5	0.494	0.032
0.6	0.4	0.399	0.029
0.7	0.3	0.293	0.034
0.8	0.2	0.201	0.034
0.9	0.1	0.094	0.017

The next case is analyzing a four node linear network with one source, one sink and two intermediate nodes. The case starts including duplicates, as in equation 18, thus total packet reception probability is:

$$p_{rx2} = p \text{ (receive a packet at node 2)} = p_{rx2,1} + p \text{ (rx duplicate)}$$

If node 2 receives a duplicate, it means node 2 was off at t=1 and the source received the packet forwarded by node 1 at t=1 and node 1 was on at t=2 (to receive the forwarding from the source) and node was on at t=3. The following equation shows the situation:

$$p \text{ (rx duplicate)} = p_{gsp} * (1-p_{gsp}) * (1-p_{gsp}) * (1-p_{gsp}) * (1-p_{gsp}) * 1 * (1-p_{gsp})$$

In other words:

$$p_{rx2} = (1-p_{gsp}) * [p_{gsp} + (1-p_{gsp})^2] + p_{gsp} * (1-p_{gsp})^5$$

Table 16 shows the values obtained through simulations agree with the values obtained with analytical expressions, thus cross-validating the results obtained with the simulation software developed during this research.

Table 16. Packet reception probability in the second hop, REALGSP.

p_{gsp}	Analytical	Experimental	
	Value	Value	Std Dev
0.1	0.878	0.885	0.030
0.2	0.738	0.749	0.033
0.3	0.603	0.602	0.039
0.4	0.487	0.499	0.036
0.5	0.391	0.390	0.029
0.6	0.310	0.311	0.036
0.7	0.239	0.228	0.028
0.8	0.168	0.174	0.030
0.9	0.091	0.096	0.019

Now consider the behavior of MACGSP6, which can also be modeled using state diagrams. In fact, the states and transition probabilities of nodes using MACGSP6 when there are no packets in the network are the same as shown in Figure 51 and Figure 52. However, MACGSP6 was designed to decrease the number of duplicates generated by REALGSP, thus the total state diagram is different. Figure 55 explains MACGSP6 functionality by including state 2 (transmitting), state 3 (retransmit, if no ACK was heard) and state 4 (sleep according to the number of quiescent periods).

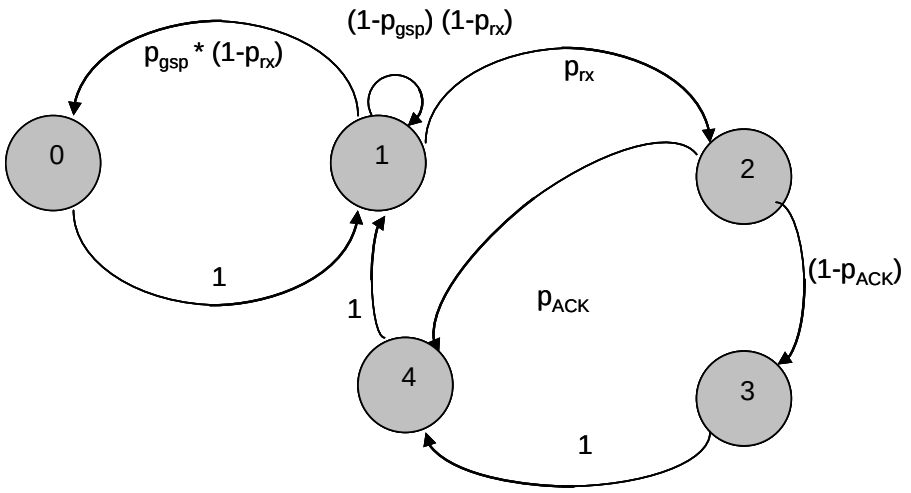

Figure 55. State diagram for MACGSP6.

The source is the first node in transmit, so it will leave state 2. Then, according to protocol rules, if the source hears a preamble, it will go to sleep. Otherwise, it will send the packet again. Therefore:

p_{rx1} = p(node 1 rx one packet) = p(node 1 on at t = 0) + p(node 1 rx at t=2),

$p_{rx1,2}$ is the probability of node 1 receiving at t=2. The situation arises when the first node did not receive the packet in the first occasion (at time =0, when the source transmits for the first time), but the node is on to receive the retransmission form the source at t=2. Hence:

$p_{rx1,2}$ = p(node 1 off at t=0)* p(node 1 on at t=1) * p(node 1 on at t=2), thus

$$p_{rx1} = (1-p_{gsp}) + (p_{gsp}) * 1 * (1-_{pgsp}) \quad (21)$$

$$p_{rx1} = p_{rx1,0} + (p_{gsp}) (1-_{pgsp}) = 1 - p_{gsp}^{2}$$

Note equation 21 yields a higher probability of receiving a packet in node 1 than equation 16. The situation in node 2 is as follows:

p_{rx2} = p (node 2 rx one packet) = $p_{rx2,1} + p_{rx2,3} + p_{rx2,5}$

since the node can receive the packet at time = 1 (the best case of packet forwarded from the source with minimum delay), time =3 or time =5 (if the originator node did not hear the proper ACK and retransmitted the packet).

$p_{rx2,1}$ is the same as in REALGSP:

$p_{rx2,1} = (1 - p_{gsp}) * [\ p_{gsp} + (1 - p_{gsp})^2\ ] = 2*p_{gsp}^2 - p_{gsp}^3 - 2*p_{gsp} + 1$

$p_{rx2,3}$ is the probability of receiving the packet at time = 3. The situation happens if:

Node 1 received the original packet at time =0 and node 2 was off , but it is on again at time = 3 to hear the retransmission from node 1. Or

Node 2 received the original packet at time = 2 and node 2 was on at t=3 to receive the packet. Hence

$p_{rx2,3} = p_{rx1,0}$ * p (node 2 off at t=1)* p (node 2 on at t=2) * p (node2 on at t=3) +

$prx_{1,2}$ * p(node 2 on at t=3)

p(node 2 off at t=1) = $(1- p_{gsp}) * p_{gsp}$

p(node 2 on at t=2 given it was off at t=1) = 1

p(node on at t=3 given it was on at t=2) = $(1-p_{gsp})$

p(node 2 on at t=3) = $aP^3_{[1]}$

Hence:

$p_{rx2,3} = p_{rx1,0} * (1-p_{gsp}) * p_{gsp} * 1 * (1-p_{gsp}) +$

$p_{rx1,2} * \{p_{gsp}*[p_{gsp} + (1-p_{gsp})^2\ ]+(1-p_{gsp})*[p_{gsp}*(1-p_{gsp}) + (1-p_{gsp})*[p_{gsp} + (1-p_{gsp})^2]]\}$

$p_{rx2,3} = 2*p_{gsp}^5 - p_{gsp}^6 - 3* p_{gsp}^4 + 5*p_{gsp}^3 - 5*p_{gsp}^2 + 2* p_{gsp}$

$p_{rx\ 2,5}$ is the probability of node 2 receiving the packet at time =5. The situation can only occur if node 1 received the packet at time =2, node 2 was off at t=3 to receive the relay packet and node 2 is on at t=5 to receive the retransmission. Thus

$p_{rx2,5} = p_{rx1,2} * p$ (node 2 off at t=3) * p(node 2 on at t=4) * p(node 2 on at t=5)

p(node 2 off at t=3) = $aP^3_{[0]}$

p(node on at t=4 given it was off at t=3) =1

p(node 2 on at t=5 given it was on at t=4) = $(1- p_{gsp})$

$p_{rx2,5} = p_{rx1,2} * aP^3_{[0]} * 1 * (1- p_{gsp})$

$p_{rx2,5} = p_{gsp} * (1 - p_{gsp})^2 * \{ p_{gsp}^2 * (1-p_{gsp}) + (1-p_{gsp})*[p_{gsp}^2 + p_{gsp} * (1- p_{gsp})^2] \}$

$p_{rx2,5} = 3*p_{gsp}^6 - p_{gsp}^7 - 4* p_{gsp}^5 + 4*p_{gsp}^4 - 3*p_{gsp}^3 + p_{gsp}^2$

Thus, p_{rx2}, the probability of receiving one packet at node 2 is:

$$p_{rx2} = p_{rx2,1} + p_{rx2,3} + p_{rx2,5}$$

$$p_{rx2} = 1-2*p_{gsp}^2 + p_{gsp}^3 + p_{gsp}^4 - 2* p_{gsp}^5 + 2* p_{gsp}^6 - p_{gsp}^7 \quad (22)$$

Retransmitting following overhearing increases the probability of receiving a packet at each node. The procedure continues and you can imagine a "wave" of transmissions in Figure 50, where the first packet traverses the network going to the sink and nodes that transmitted go to sleep, either immediately (since they overheard the relayed packet) or after retransmitting just one packet, because the packet previously sent was dropped.

Table 17 illustrates how analytical values and simulated values for MACGSP6 match, therefore the analytical model cross-validates the results obtained with the simulation tool developed for MACGSP6 as well.

Table 17. Packet reception probability for MACGSP6 in first and second hops.

p_{gsp}	First hop			Second Hop		
	Analytical	Experimental		Analytical	Experimental	
	Value	Value	Std Dev	Value	Value	Std Dev
0.1	0.990	0.990	0.006	0.981	0.981	0.011
0.2	0.960	0.963	0.011	0.929	0.928	0.019
0.3	0.910	0.912	0.023	0.851	0.856	0.025
0.4	0.840	0.840	0.026	0.756	0.758	0.032
0.5	0.750	0.745	0.028	0.648	0.643	0.026
0.6	0.640	0.638	0.031	0.535	0.529	0.040
0.7	0.510	0.506	0.041	0.420	0.423	0.035
0.8	0.360	0.358	0.038	0.301	0.295	0.033
0.9	0.190	0.189	0.029	0.169	0.169	0.021

Comparing equations from MACGSP6 and REALGSP, the first one has higher packet reception probability than the second one and both protocols decrease this metric when increasing the number of hops. However, REALGSP generates duplicates, which compensate for a smaller reception rate and MACGSP6 does not. Additionally, exponents in MACGSP6 increase at every hop with higher rate than REALGSP (as shown in equations 21 and 22), thus the effect of the number of hops in MACGSP6 is also higher. The analytical models suggest packet reception probability in REALGSP should be higher than MACGSP6 when the number of hops is large.

Analysis of neighbors is similar for MACGSP6 and REALGSP, thus increasing the number of neighbors increases the probability of reception in MACGSP6.

Figures 56 to 62 show results for 100, 400 and 900 nodes, with sink in the middle of the line. The routes for these topologies include 49, 199 and 449 hops respectively. Figure 56 exhibits results for packet reception probability. There is no observable difference for REALGSP at 90% confidence levels with different network sizes. The protocol delivers packets to the destination with $p_{gsp} < 0.3$. MACGSP6 shows smaller values of packet reception probabilities, which decrease when network sizes (and number of hops) increase.

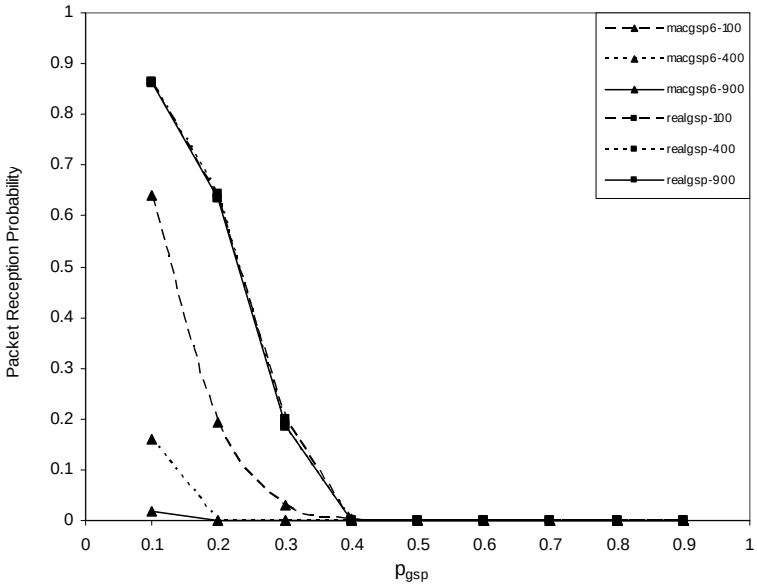

Figure 56. Packet reception probability, linear topologies.

Figure 57 shows the average number of duplicates received by the source for both protocols. MACGSP6 has less than one duplicate received, regardless network size. REALGSP presents different number of duplicates increasing with network size. Larger networks have more duplicates but also more hops to traverse, resulting in the same packet reception probability as the network size changes.

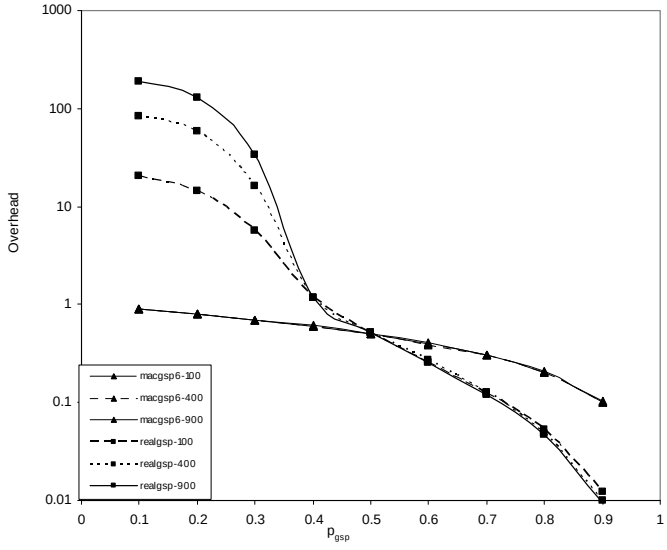

Figure 57. Duplicates, linear topologies.

Duplicating packets is the only mechanism implemented by the GSP family of protocols to forward the packet to the destination. Note figure 57 shows REALGSP exhibits an abrupt fall in the number of duplicates for the same values of p_{gsp} where packet reception probability is close to zero, consequence of REALGSP functionality. MACGSP6 eliminates most duplicate packets, thus decreasing probability of delivering the data packets.

Figure 58 curves illustrate delay as a function of p_{gsp}. MACGSP6 shows a constant trend, close to the ideal value (491 TP for 100 nodes). REALGSP exhibits values greater than ideal for all network sizes, showing the packet does not follow the shortest route. The packet actually goes back and forth, possibly several times, in the linear topology before reaching its destination.

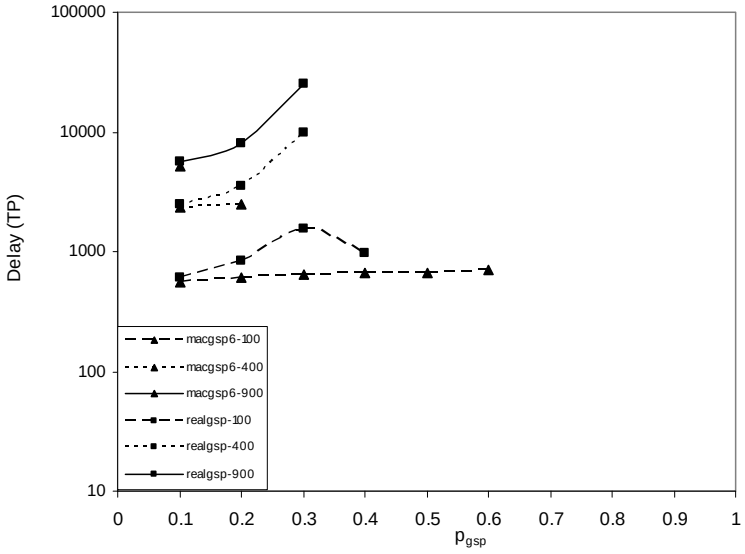

Figure 58. Delay, linear topologies.

Figure 59 presents total energy use, showing MACGSP6 uses one order of magnitude less energy than REALGSP for values of $p_{gsp} < 0.4$. Note again the abrupt fall in energy use around the same values of p_{gsp} where packet reception probability is very low. Therefore, energy values in the right side of the curve correspond to energy used by the network while there are no packets transmitted.

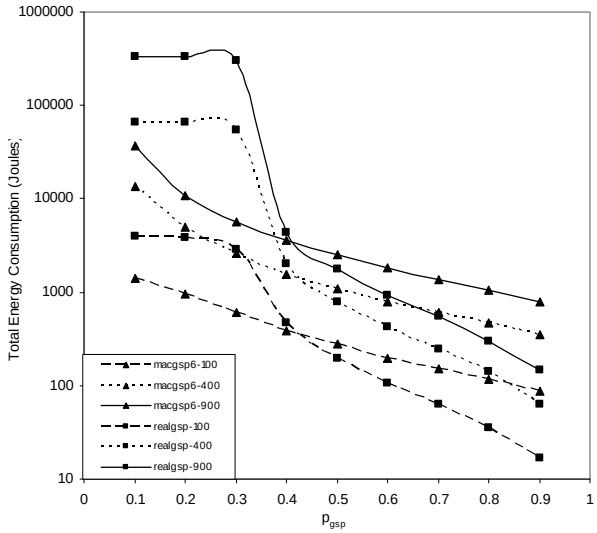

Figure 59. Total energy use, linear topologies.

Figure 60 illustrates energy used in transmission state. REALGSP uses up to three orders of magnitude more energy than MACGSP6. The behavior is consequence of the number of duplicates traversing the network. Energy used in the transmission state is two orders of magnitude smaller than the total, similar to results showed in section 4.4.

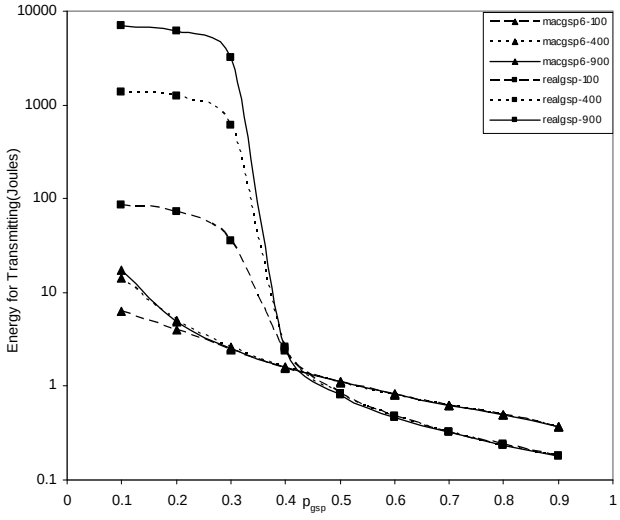

Figure 60. Energy for transmission, linear topologies.

Figure 61 presents energy for receiving. REALGSP is at least one order of magnitude higher than MACGSP6. The proportion of the total energy use is similar to trends previously observed.

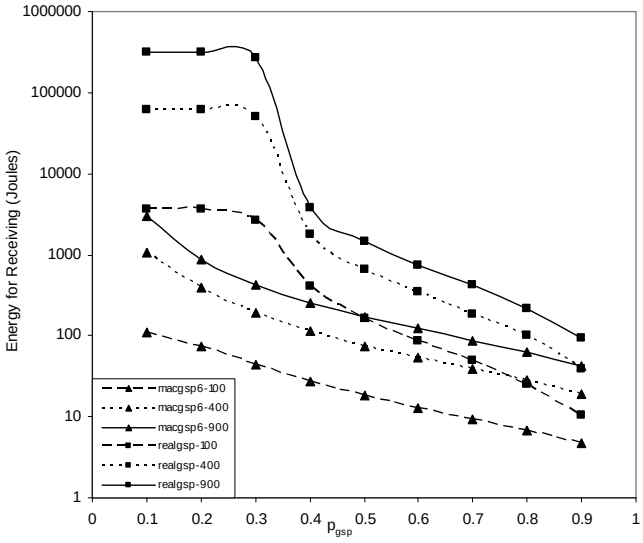

Figure 61. Energy for reception, linear topologies.

Figure 62 shows energy used by nodes in the off state. The trend is similar to results in section 4.4. REALGSP puts to sleep a smaller number of nodes than MACGSP6, thus energy spent in this state may be as low as one order of magnitude less for REALGSP.

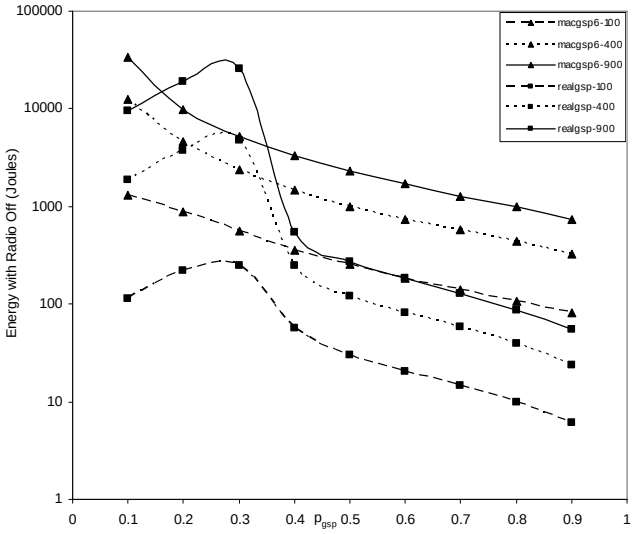

Figure 62. Energy with radio off, linear topologies.

Note both protocols deliver few packets to the destination and REALGSP has higher packet reception probability. Nonetheless, energy use has the same trends as in previous cases, with MACGSP6 using one order of magnitude less total energy than REALGSP. Observe the protocols do not deliver packets for values of $p_{gsp} > 0.4$, therefore data in the figures reflect the behavior of the network when nodes do not forward packets.

4.5.2 Rectangular

The topologies employ a configuration similar to square grids: horizontal and vertical distances between nodes are equal to r, the radio range. Table 18 shows the network sizes used in the study, which are the same as in [3].

Table 18. Rectangular topologies network sizes.

Number of Rows	Number of Columns	Total Number of Nodes	Minimum Number of Hops
5	20	100	11
5	100	500	51
5	200	1000	101

Note rectangular topologies may be represented as 5 linear topologies. Figure 63 presents packet reception probability. MACGSP6 has the highest value for the smallest network, since the number of hops is small. GSP shows no observable difference at 90% confidence levels, regardless network size or number of hops, for values of $p_{gsp} < 0.6$. Network sizes of 500 and 1000 nodes decrease packet reception probability in MACGSP6. Since the topology has 5 rows, the network has less chances of becoming disconnected than a linear topology. However, MACGSP6 has better probabilities than REALGSP for $p_{gsp} < 0.3$. Higher values of p_{gsp} show

MACGSP6 has 30% less probability of packet reception than REALGSP. The result suggests there is a minimum number of hops from source to destination for which MACGSP6 has better packet reception probability by the sink than REALGSP. According to Figure 63, this number is less than 51 but more than 11. In the results from square grids MACGSP6 performed always better than REALGSP and the numbers of hops employed in those networks were 8, 18 and 28.

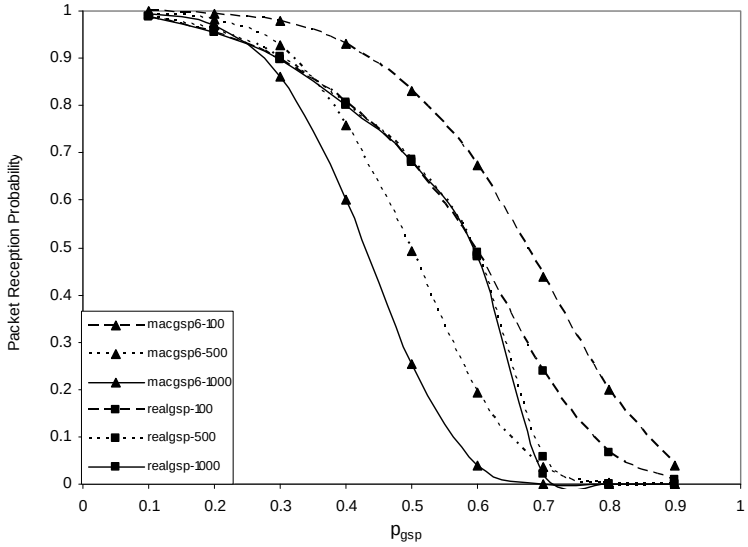

Figure 63. Packet reception probability, rectangular networks.

Figure 64 displays the average number of duplicate packets received by the source. As with linear topologies, MACGSP6 receives in average less than 1 duplicate, regardless network size. REALGSP receives a smaller amount of duplicates in the 100 network, and the number increases for bigger network sizes. Duplicates help in forwarding the packet; therefore, results in Figure 64 are consistent with packet reception probability.

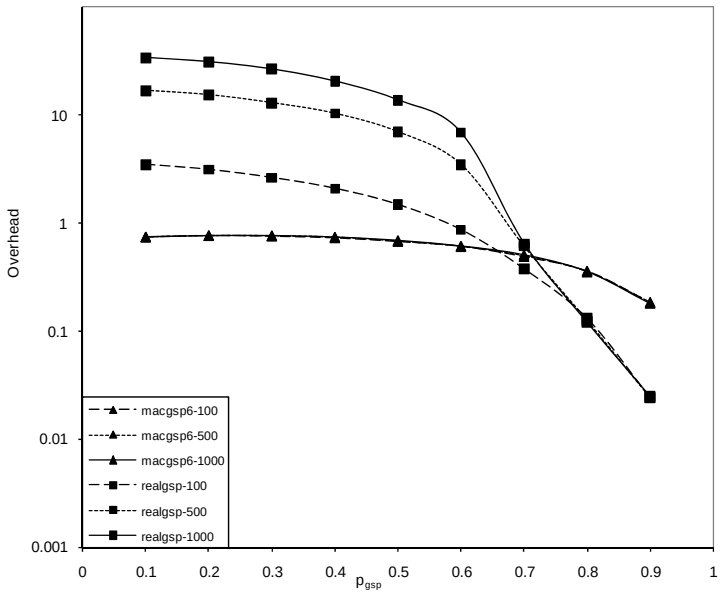

Figure 64. Duplicates, rectangular networks.

Figure 65 illustrates delay for receiving the first packet by the sink. There is no observable difference for delay in delivering a packet at 90% confidence levels for both protocols with $p_{gsp} < 0.5$. Values are closer to the ideal required, as expected.

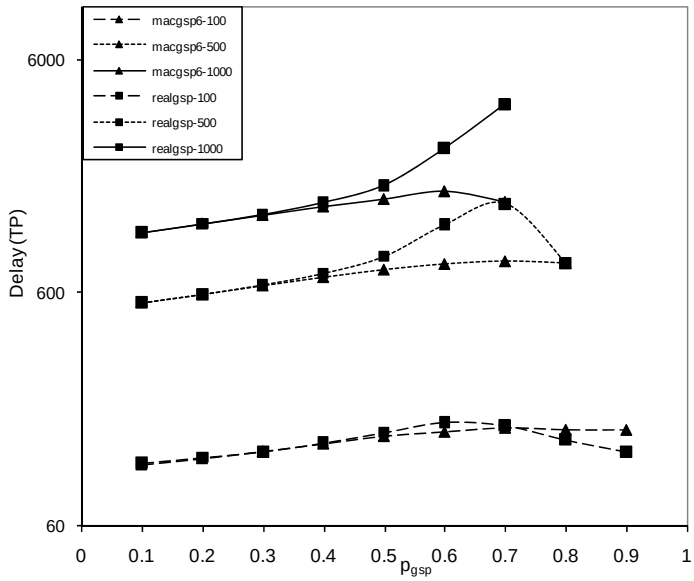

Figure 65. Delay, rectangular networks.

Figure 66 presents total energy used in the network. MACGSP6 employs as low as 50% of the total energy employed by REALGSP, with trends reversed for values of $p_{gsp} > 0.6$. However, MACGSP6 delivers 40% less packets than REALGSP in the worst case as shown in Figure 63. Energy delivered per successful packet received may help to determine which protocol has best energy use in rectangular networks.

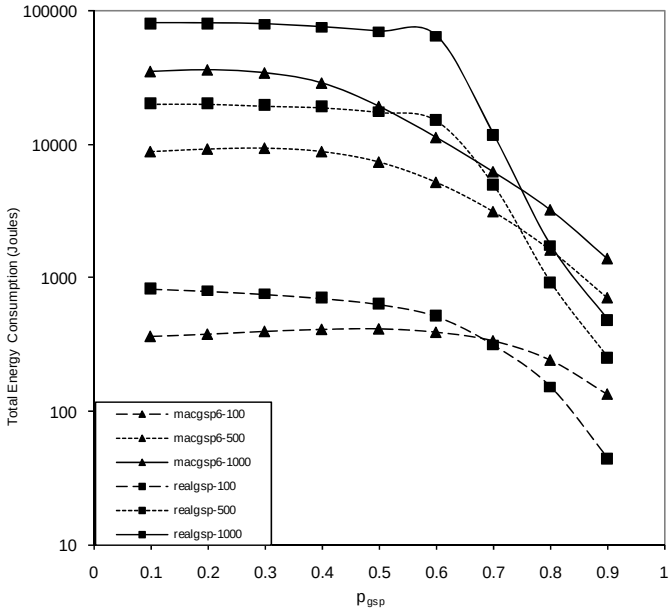

Figure 66. Total energy use, rectangular networks.

Figure 67 shows energy used per successful packet received is always smaller for MACGSP6 for 100 nodes. Bigger network sizes favor the use of MACGSP6 when $p_{gsp} < 0.4$.

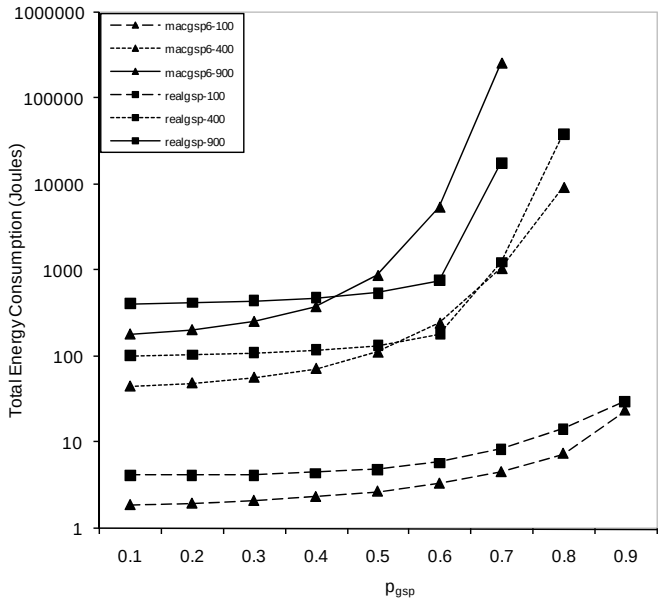

Figure 67. Total energy use per packet successfully delivered, rectangular networks.

The following figures illustrate the different components of total energy use presented in Figure 66. Figure 68 displays energy used by all nodes in the transmitting state to forward 200 packets. MACGSP6 has less energy in this state than REALGSP. The difference increases with network size and it can be up to one order of magnitude, showing REALGSP makes more nodes transmit due to the number of duplicates.

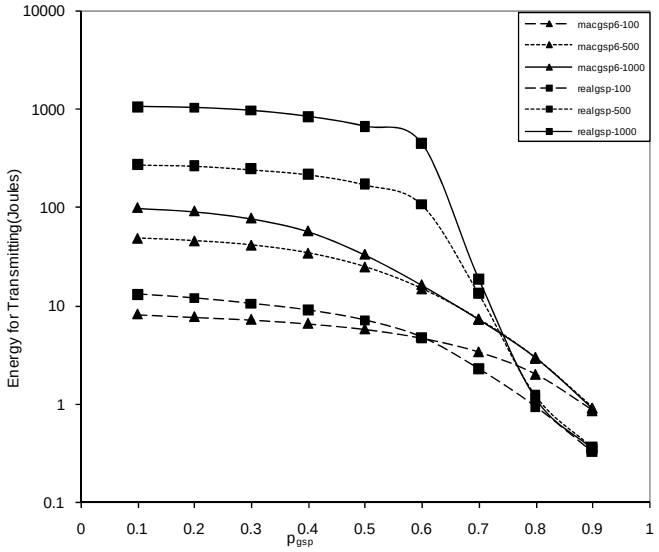

Figure 68. Energy for transmitting, rectangular networks.

Figure 69 shows energy used by all nodes in the receiving state. MACGSP6 uses one order of magnitude less energy than REALGSP in this state. Comparing the values used for transmitting and for receiving, the trend observed in previous results continues: REALGSP uses most of its energy in idle listening, while MACGSP6 decreases this situation. Both protocols show decreasing behavior in this metric when p_{gsp} increases, since nodes have more chance of entering the sleeping state.

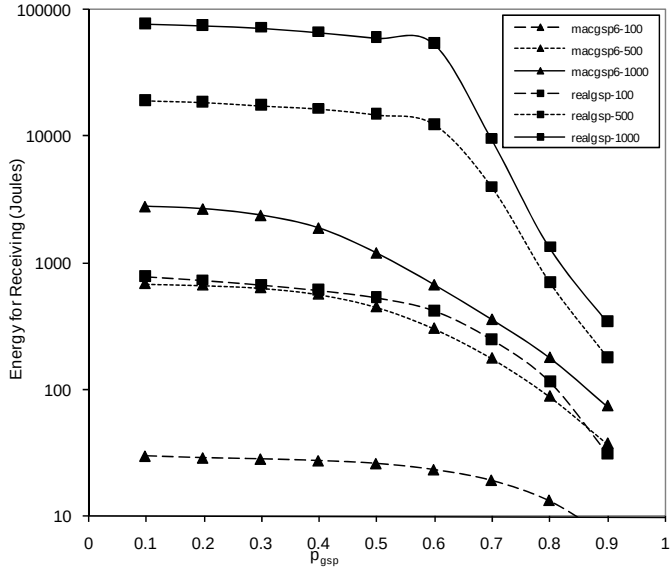

Figure 69. Energy for receiving, rectangular networks.

Figure 70 presents energy used in the radio off state for forwarding 200 packets. Trends are the same as observed in square grids. MACGSP6 uses around 90% of its total energy in putting nodes to sleep, while REALGSP uses one order of magnitude less than total energy with the radio off.

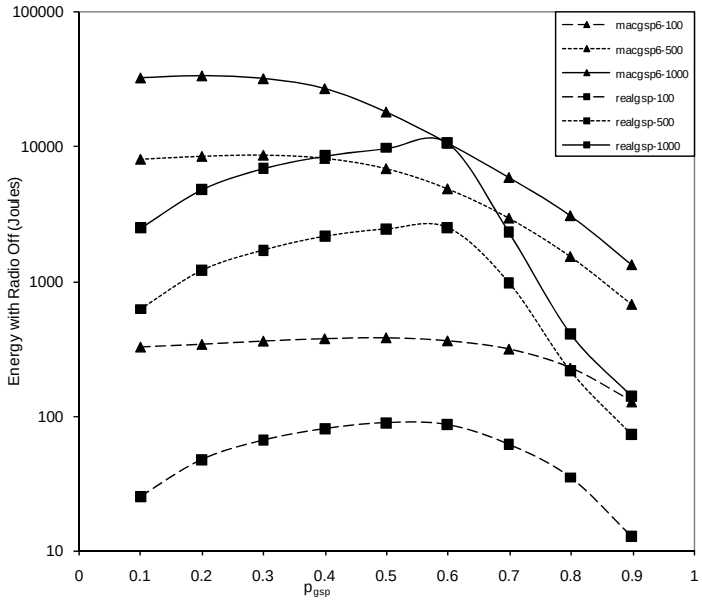

Figure 70. Energy with radio off, rectangular networks.

4.5.3 Random

Location of the nodes in these networks relays on selecting values on two coordinates (X and Y) from a uniform random variable inside a square area. Experiments used areas of 10 r x 10 r, 20 r x 20 r and 25 r x 25 r for 100, 800 and 1250 nodes respectively, similar to [3], where r is the maximum distance a node can communicate with another. Experiments used the same topologies for all the protocols in each network size. Figures 71 to 73 present the topologies employed in each case. The minimum number of hops for 100 nodes is 8, for 800 nodes is 21 and for 1250 nodes is 25.

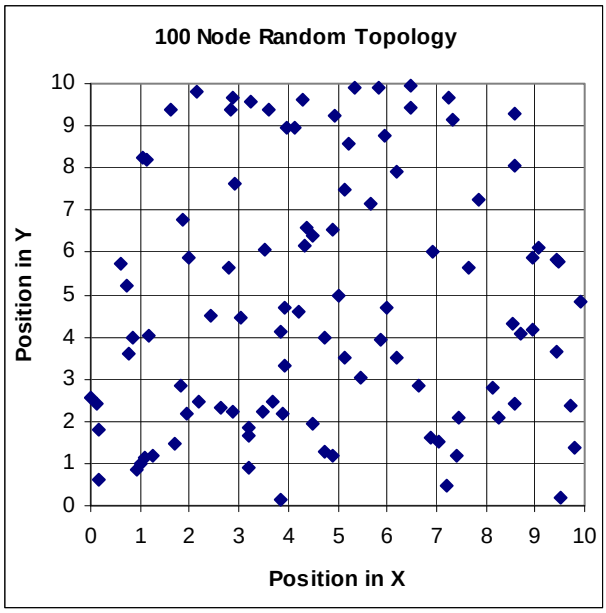

Figure 71. Random topology used for 100 nodes.

Note 800 and 1250 nodes topologies have more node density than 100 nodes; therefore, nodes have more neighbors increasing the number of possible routes from source to destination.

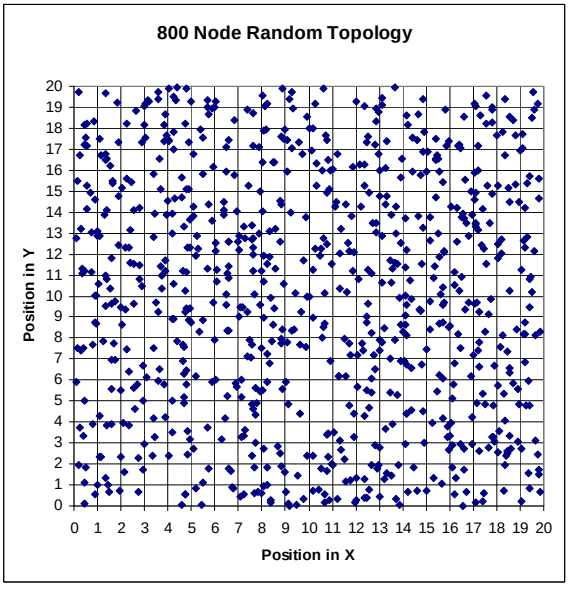

Figure 72. Random topology used for 800 nodes.

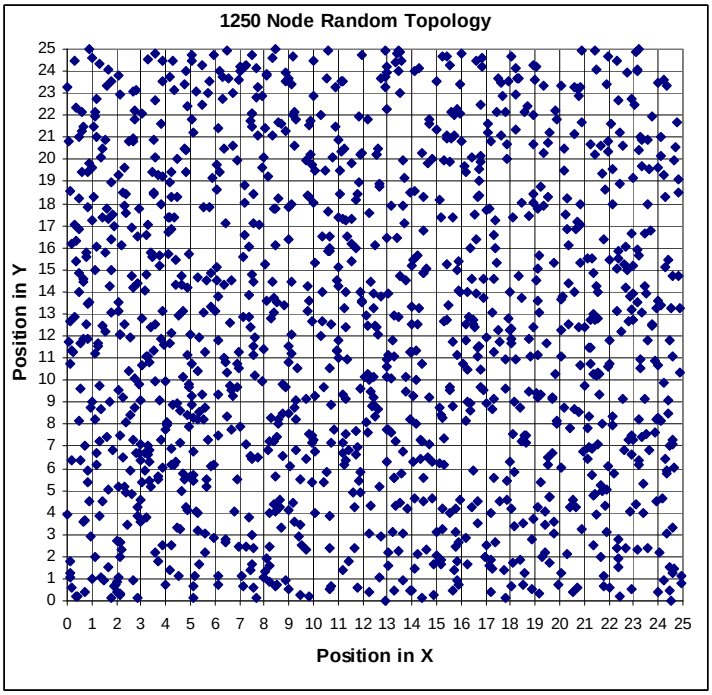

Figure 73. Random topology for 1250 nodes.

Figure 74 presents packet reception probability obtained for both protocols with the three random networks. REALGSP has better packet reception probability than MACGSP6. The difference is more noticeable for the smallest network, even if the minimum route from source to destination is just 8 hops. Turns out the sink has only one neighbor, thus reception probability decreases. Additional metrics help provide more information on this situation. Nonetheless, the trend is the opposite of the preliminary work in that bigger network sizes have higher packet reception probabilities for both protocols. The situation may be due to the network density, i.e the average number of neighbors, which is 2.84, 5.9 and 5.88 for 100, 800 and 1250 nodes respectively. Behavior according to network density will be addressed in section 4.7.

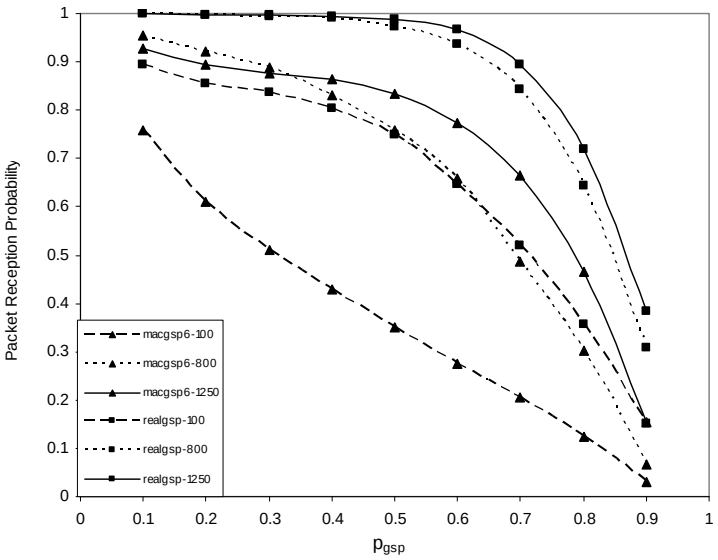

Figure 74. Packet reception probability, random topologies.

Figure 75 presents MACGSP6 decreases the number of duplicates generated by REALGSP by as much as two orders of magnitude. The situation should decrease energy use in the network; however, duplicates are the only mechanism employed by the GSP family of protocols for finding the route from source to destination. Decreasing this metric for this particular topology also decreased the number of packets successfully received, as shown in Figure 74.

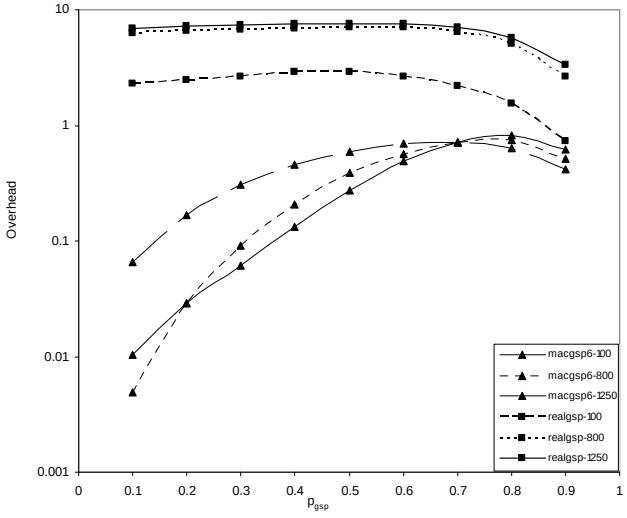

Figure 75. Duplicates received in source, random topologies.

Figure 76 illustrates delay values until the first packet is received by the sink. MACGSP6 and REALGSP have no observable differences at 90% confidence level for topologies of 1250 nodes. However, MACGSP6 presents higher delay with 800 nodes and considerable higher delay values for REALGSP with 100 nodes, thus the amount of duplicates in the network create longer routes but eventually more packets reach the sink. The fewer packets that finally arrived to the sink using MACGSP6 apparently employed shorter routes.

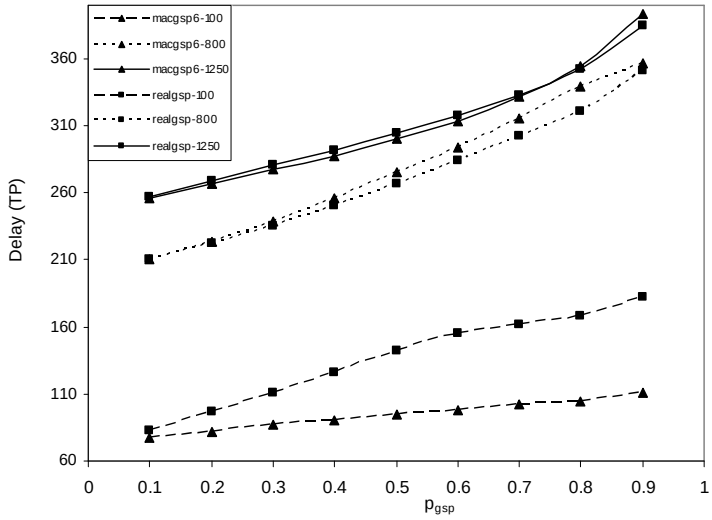

Figure 76. Delay, random topologies.

Figure 77 shows total energy use for MACGSP6 is around half of energy used by REALGSP, thus, even if MACGSP6 delivers a smaller amount of packets, energy savings may compensate for this situation. Energy per successful packet received provides more information in this matter. Note values for both protocols are constant regardless p_{gsp} with a small decreasing trend for $p_{gsp} > 0.8$, as opposite as results presented in Figure 40, where REALGSP has a clear decreasing trend during the whole range of p_{gsp}.

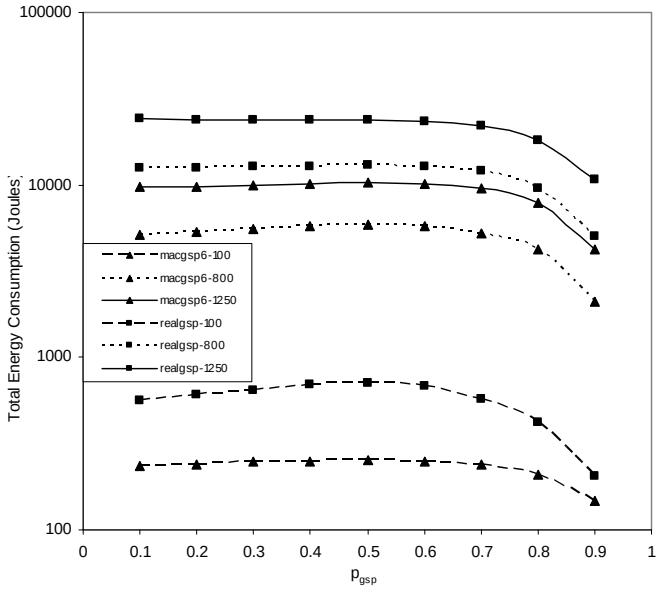

Figure 77. Total energy use, random topologies.

Figure 78 presents total energy use per successful packet received. MACGSP6 uses around 40% less energy than REALGP for all values of p_{gsp} in the biggest network size, and for $p_{gsp} < 0.7$ for the other two network sizes.

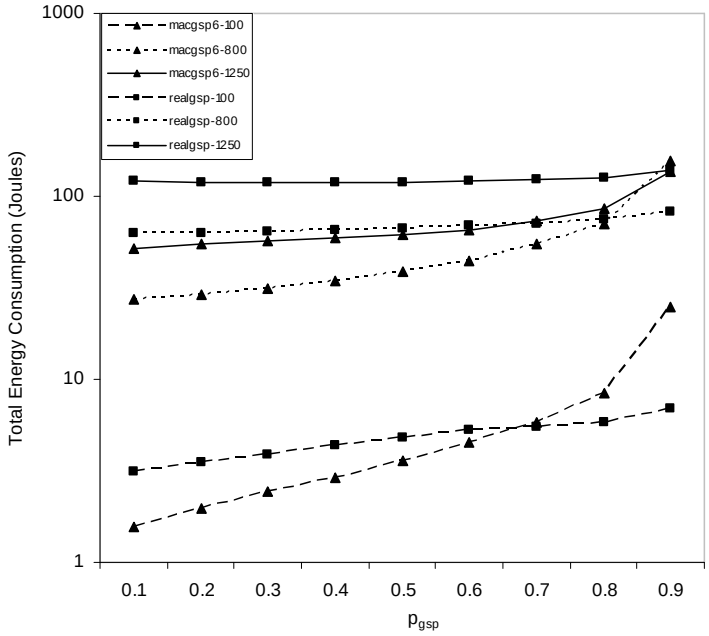

Figure 78. Total energy used per successful packet received, random topologies.

The following figures present the distribution of total energy use for sending 200 packets. Figure 79 presents energy used in the transmission state. REALGSP has twice as many nodes in this state as MACGSP6, due to the amount of duplicates traversing the network. As usual, energy spent in this state is two orders of magnitude smaller than total energy use.

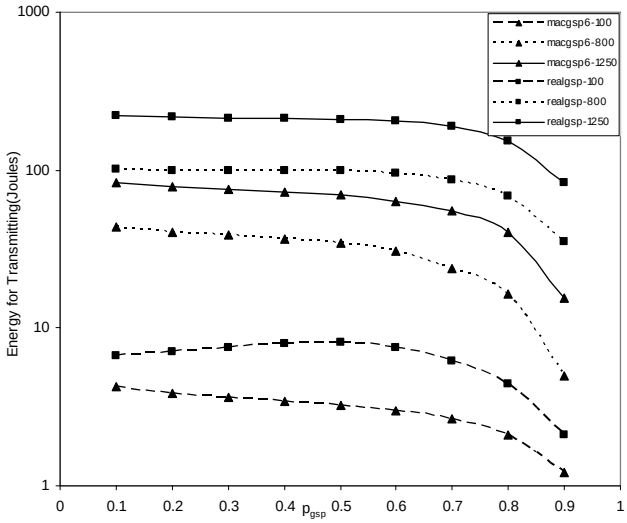

Figure 79. Energy spent in transmission, random topologies.

Figure 80 shows energy used by all nodes in receiving state for sending 200 packets.

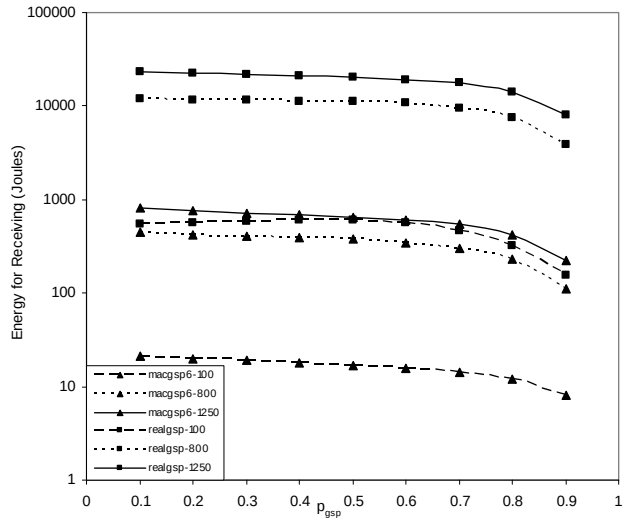

Figure 80. Energy for reception, random topologies.

Energy in receiving state follows trends presented before: REALGSP uses most of total energy for receiving as consequence of idle listening. MACGSP6 uses one order of magnitude less energy in receiving than the total energy use. As p_{gps} increases, both protocols present a slowly decreasing trend, created by having more possibility of nodes reaching the sleeping state.

Figure 81 illustrates energy spent in the radio off state. Again, REALGSP has lower energy use due to fewer nodes sleeping compared to MACGSP6. Quiescent periods in MACGSP6 make nodes sleep during longer times; therefore, networks using MACGSP6 consumes up to one order of magnitude more energy in this state than networks using REALGSP.

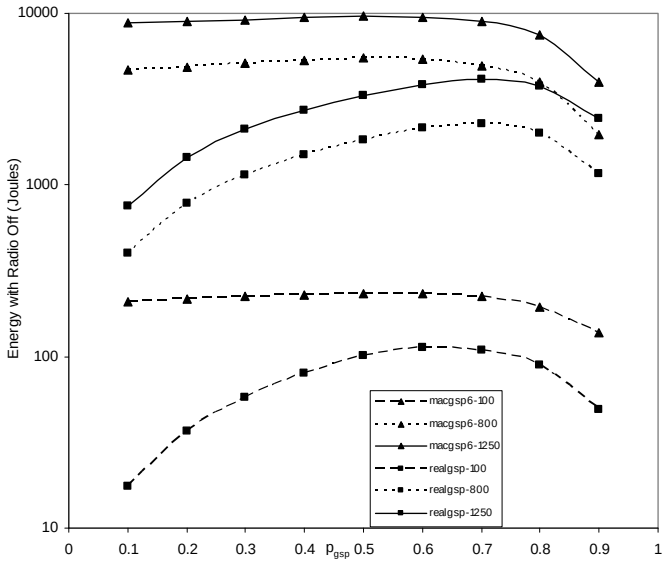

Figure 81. Energy spent with radio off, random topologies.

4.5.4 Lattice

Lattice are useful to monitor cars in streets in a city or airplanes in runways [3]. The experiments used lattices with 240, 656 and 1136 nodes, covering areas of 18 r x 18 r, 44 r x 44 r and 74 r x 74 r [76]. Horizontal and vertical distances between neighbor nodes are equal to r, the radio

range. The minimum number of hops form source to destination is 17, 39 and 62 for 240, 656 and 1136 nodes respectively. Figures 82 to 84 show the three topologies used in these experiments.

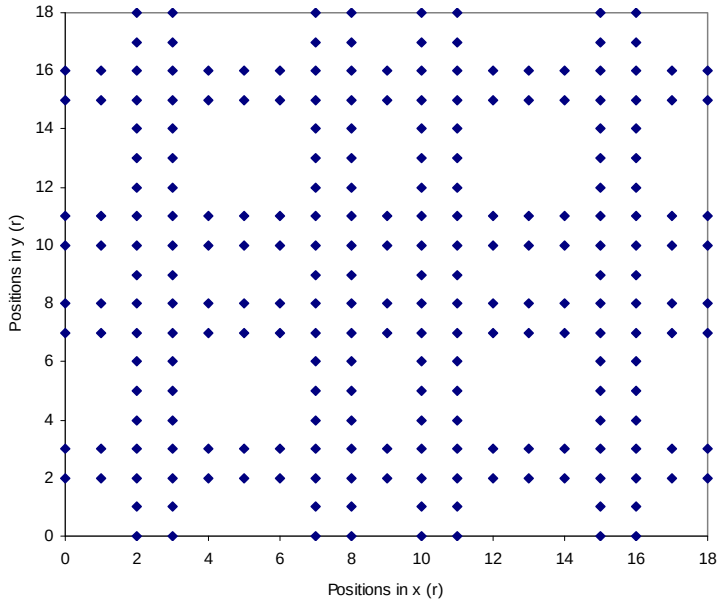

Figure 82. 240 node lattice topology.

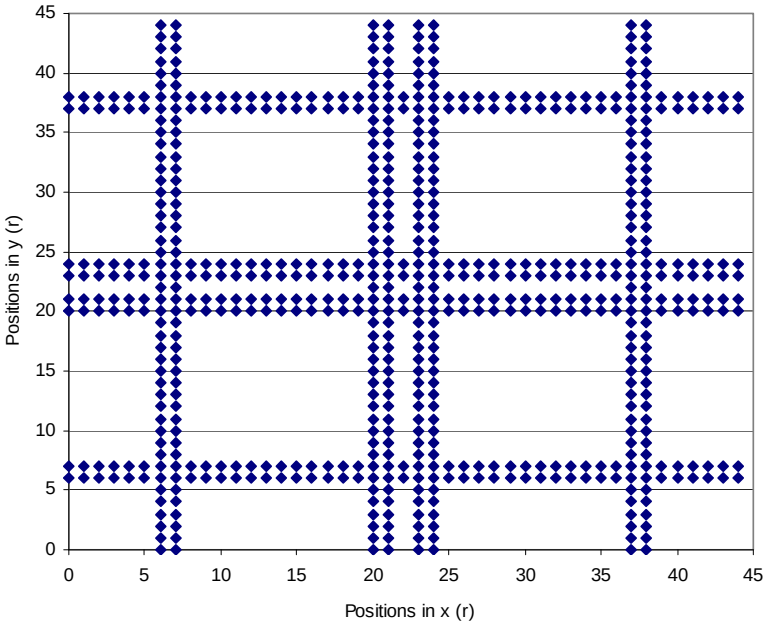

Figure 83. Lattice with 656 nodes.

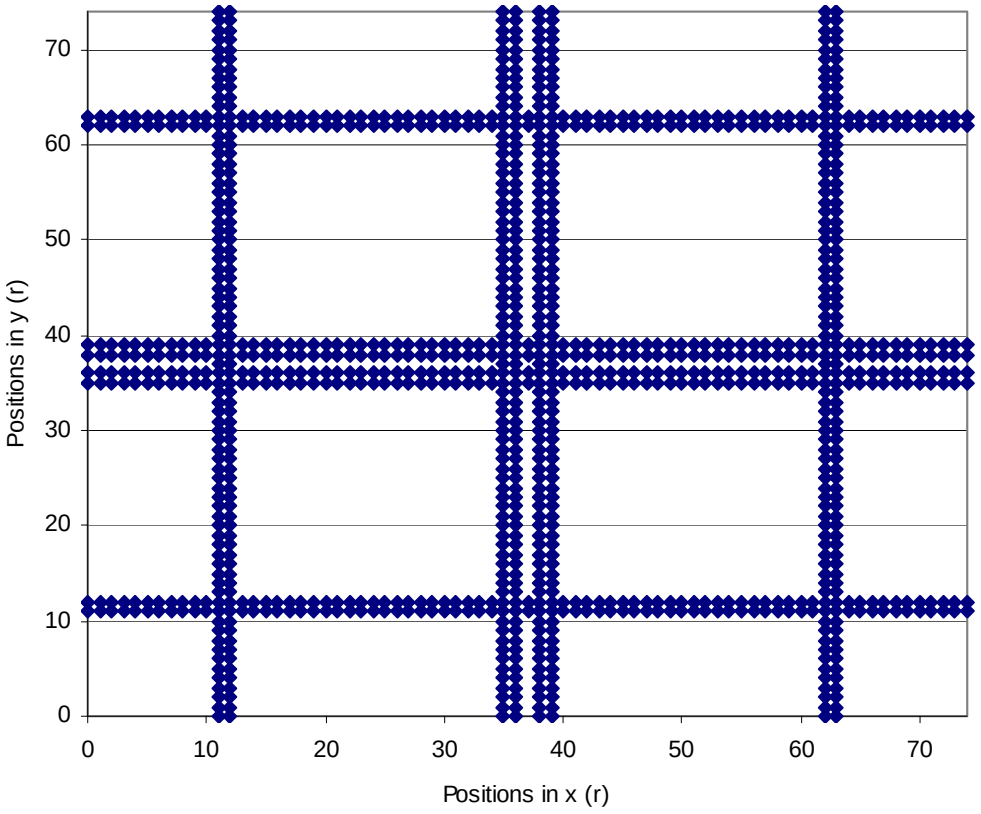

Figure 84. Lattice with 1136 nodes.

Figures 85 to 91 display results for the different topologies. All results presented have 90% confidence intervals unless otherwise noted. Figure 85 illustrates the trend for rectangular grids again: packet reception probability decreases when network size increases. MACGSP6 has higher packet reception probability than REALGSP for the smallest network (240 nodes) and the opposite situation occurs for bigger network sizes.

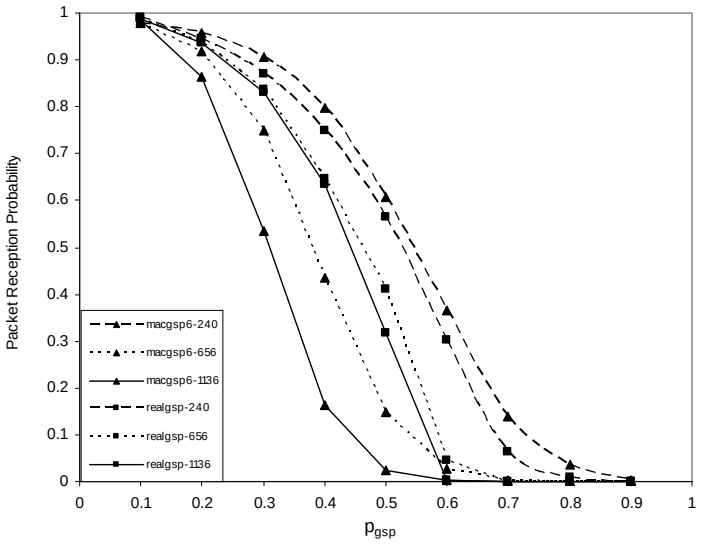

Figure 85. Packet reception probability, lattice topologies.

Figure 86 presents the average number of duplicates received by the source. As in previous cases, the number of duplicates for MACGSP6 is less than one in average, while REALGSP presents at least one order of magnitude more duplicates, increasing with the number of nodes. Note values of $p_{gsp} > 0.6$ have packet reception probabilities < 10%, thus there are very few packets forwarded through the network and the number of duplicates decreases accordingly.

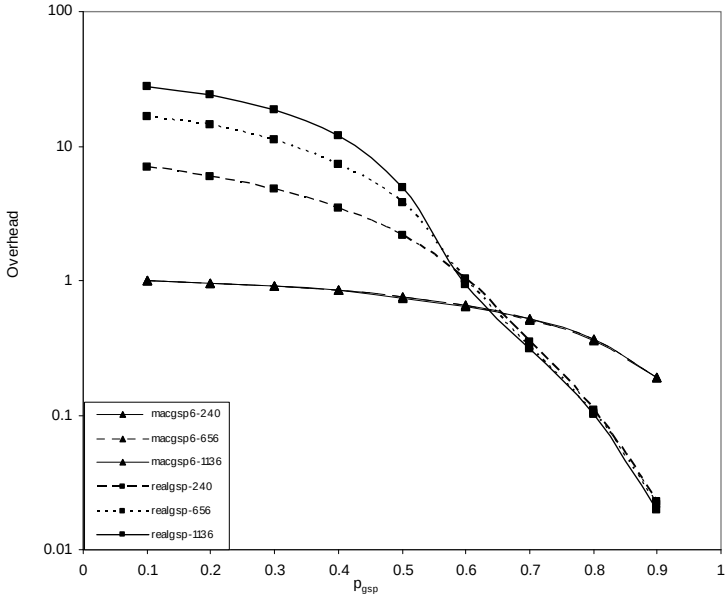

Figure 86. Duplicates, lattice topologies.

MACGSP6 has almost constant number of duplicates regardless network size. The value is one order of magnitude smaller than REALGSP for values of $p_{gsp} < 0.6$. Higher values of p_{gsp} have packet reception probabilities < 10%, thus there are very few packets forwarded through the network.

Figure 87 shows delay, with the same trend as previous cases: there are no observable differences for both protocols at 90% confidence level for small values of $p_{gsp,}$ and these values are close to the ideal. However, when p_{gsp} and network size increases, REALGSP takes longer to deliver the packets than MACGSP6.

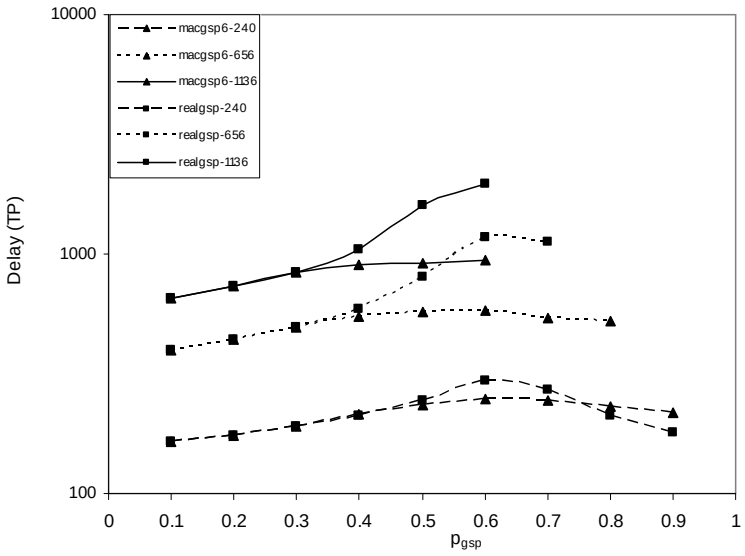

Figure 87. Delay, lattice topologies.

Figure 88 illustrates results for total energy use, which are consistent with previous cases: MACGSP6 uses 50% of the energy used by REALGSP.

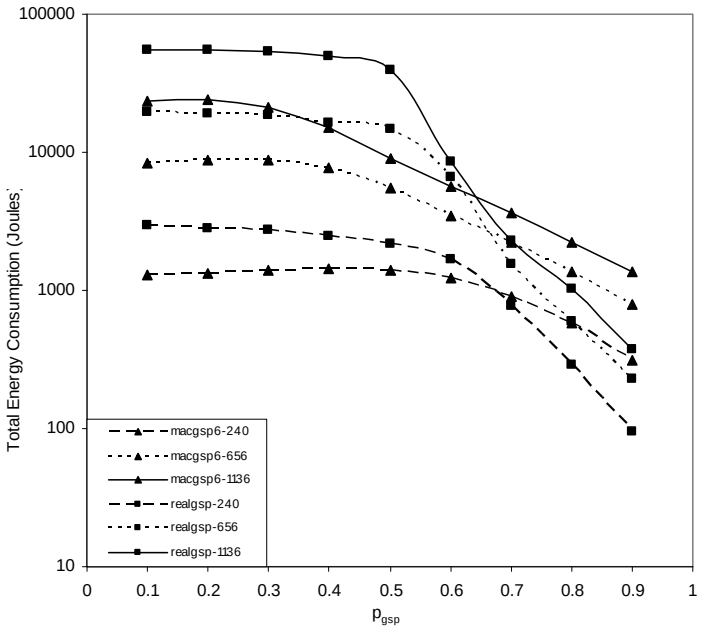

Figure 88. Total energy use, lattice topologies.

Figures 89 to 91 present the different components of energy use. Figure 89 exhibits energy used in the transmission state. As in previous cases, REALGSP has more nodes transmitting than MACGSP6, reflecting more duplicate packets in the network

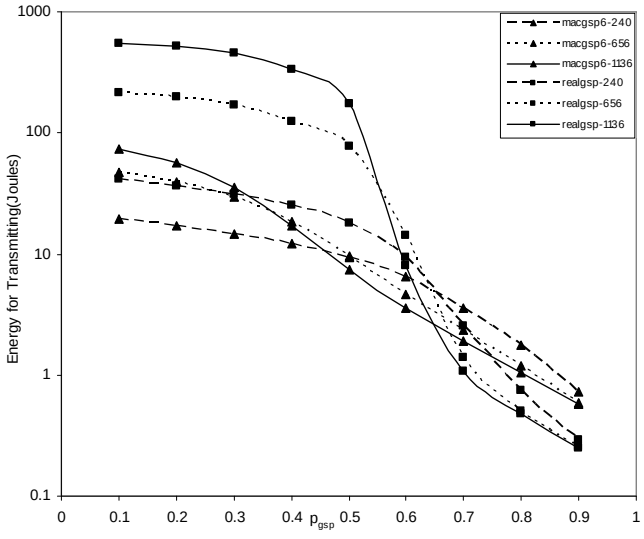

Figure 89. Energy for transmission, lattice topologies.

Figure 90 illustrates energy used in the receiving state, where again REALGSP uses the majority of energy in idle listening, while MACGSP6 uses a small percentage of energy for reception.

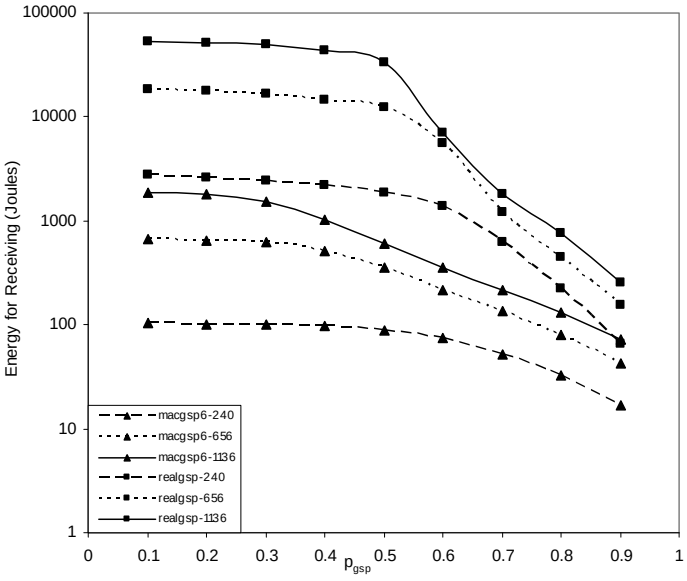

Figure 90. Energy for receiving, lattice topologies.

Figure 91 shows energy used in the radio off state, with same behavior as previous cases.

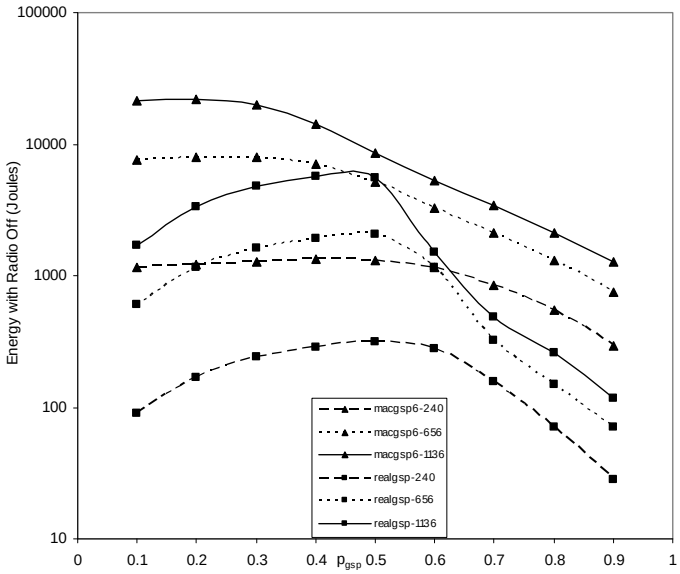

Figure 91. Energy with radio off, lattice topologies.

4.5.5 Star

The section presents results using star topologies implemented following the criteria presented in [76], that is one star is built with several lines of nodes which should send their information to the sink, located at the center of the star. The distance of two nodes located over the same diameter is r. Experiments employed stars of 320, 720 and 1280 nodes, covering circular areas with radius of 8 r, 12 r and 16 r and with minimum number of hops 8, 12 and 16 respectively. Figures 92 to 94 present the topologies used in this study.

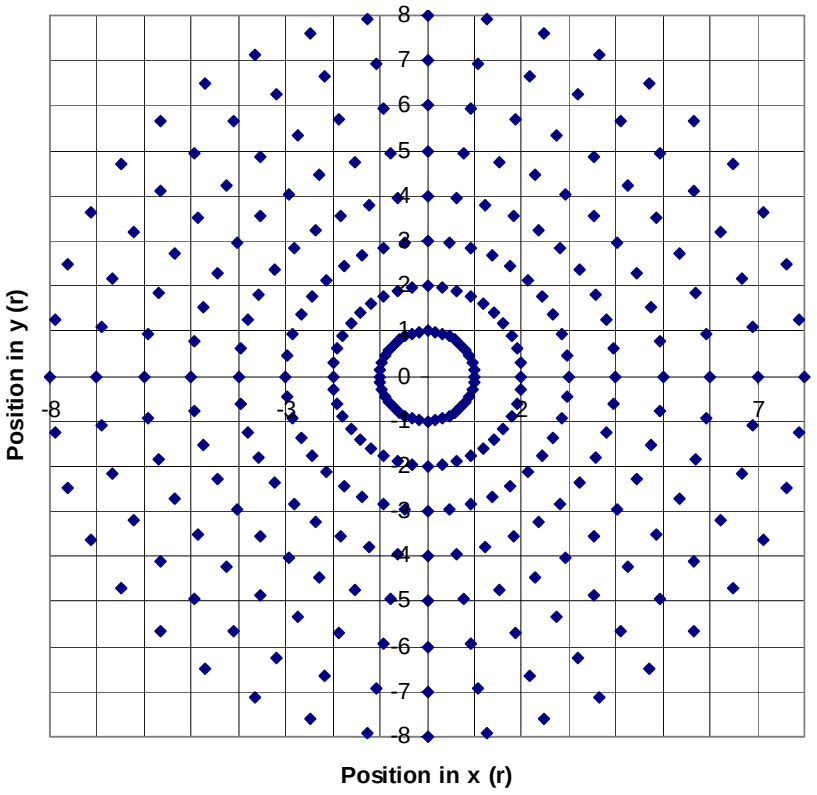

Figure 92. Star topology with 320 nodes.

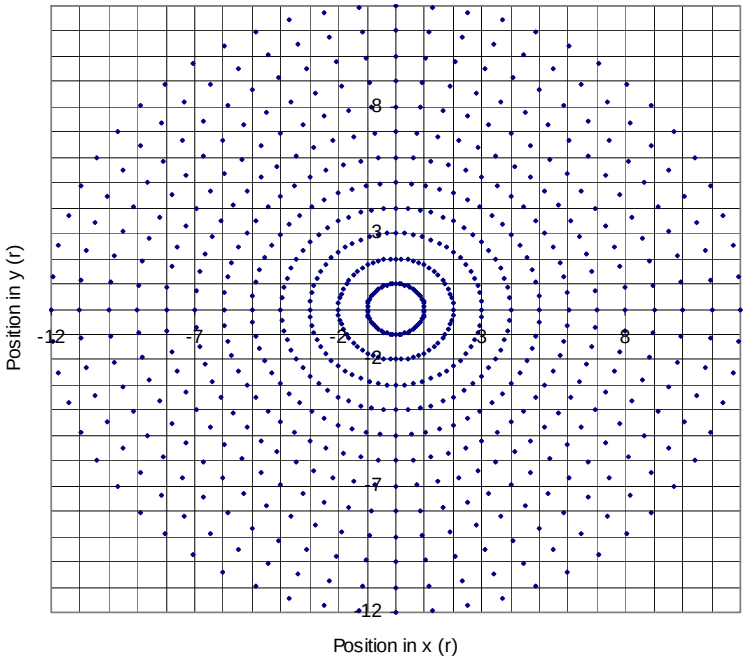

Figure 93. Star topology with 720 nodes.

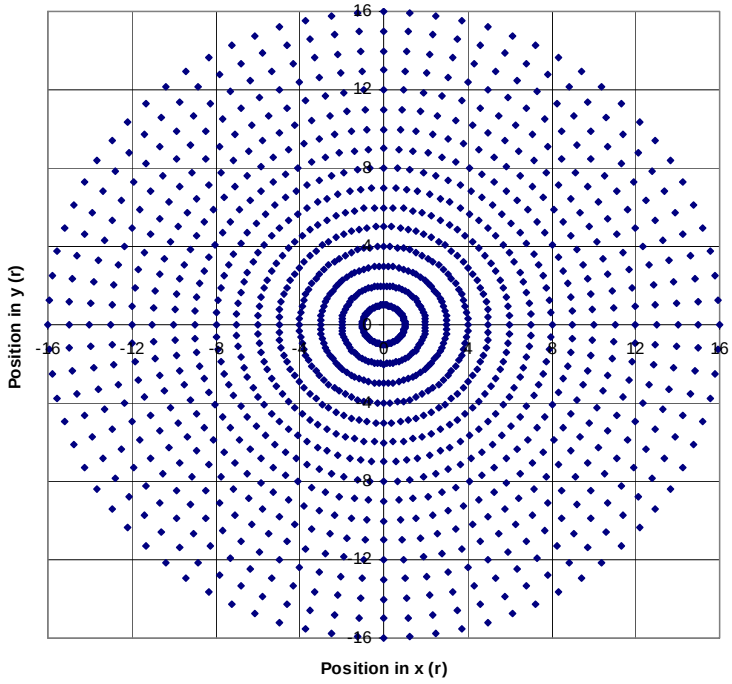

Figure 94. Star topology with 1280 nodes.

The following figures show results for the three different star topologies. Figure 95 presents packet reception probability. Star topologies present MACGSP6 having higher values in this metric than REALGSP for all values of p_{gsp}, regardless network size.

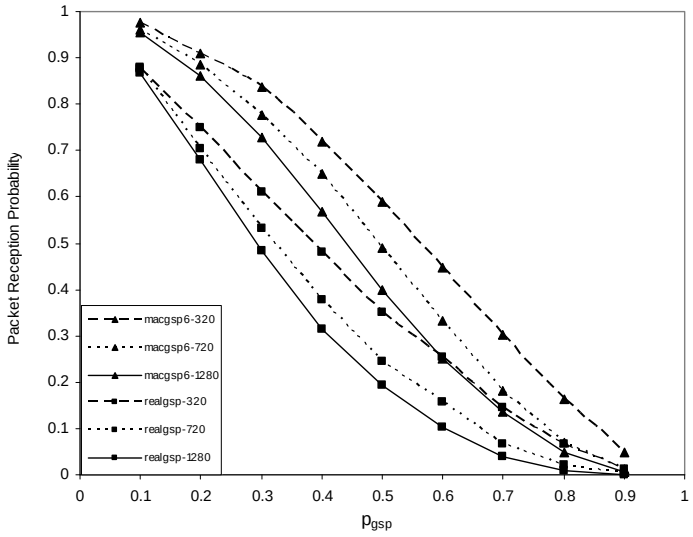

Figure 95. Packet reception probability, star topologies.

Figure 96 illustrates the average number of duplicates received by the source for both protocols. The metric follows the same trend presented before, where REALGSP has one order of magnitude more duplicates than MACGSP6 for lower values of p_{gsp}.

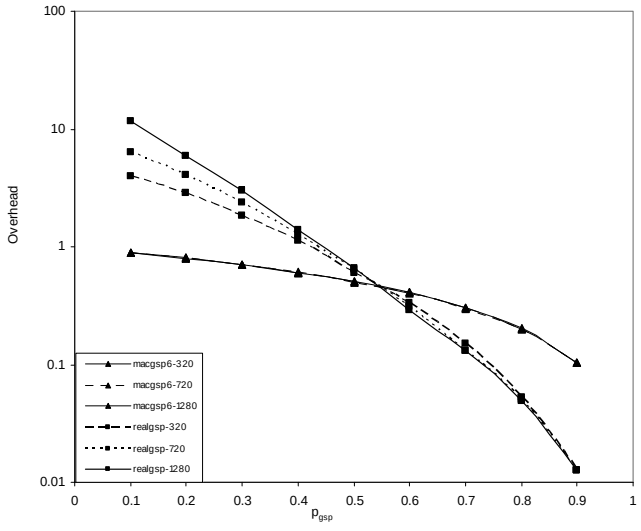

Figure 96. Duplicates, star topologies.

Figure 97 shows delay values for star topologies. MACGSP6 delivers the packets with slightly less delay than REALGSP when $p_{gsp} < 0.4$. However, both protocols are close to the ideal values for this configuration.

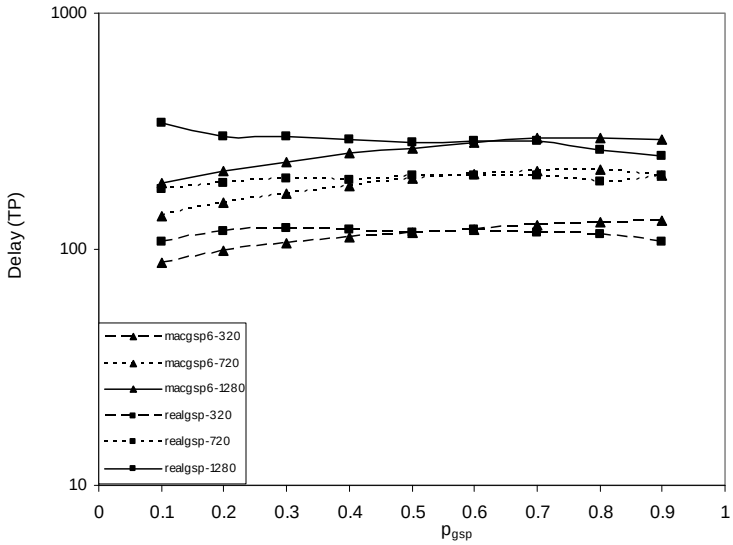

Figure 97. Delay, star topologies.

Figure 98 displays total energy use for star topologies following the same trend as in previous cases: MACGSP6 has smaller energy use than REALGSP for lower values of p_{gsp}.

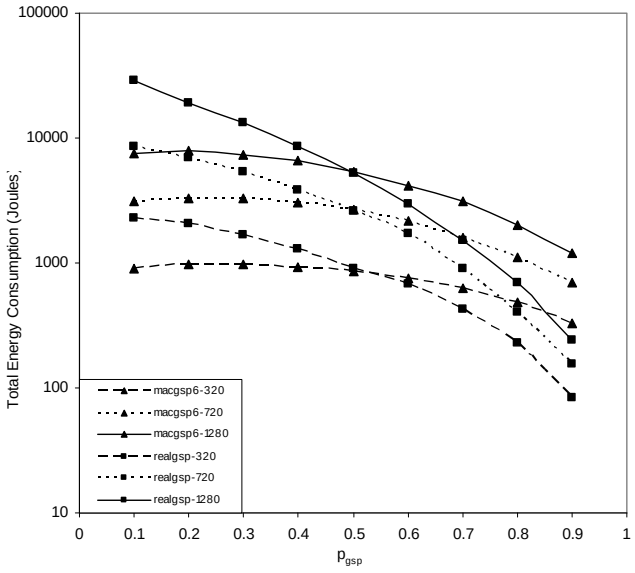

Figure 98. Energy use, star topologies.

Note packet reception probability is higher for MACGSP6 than REALGSP, so even if the total energy used is smaller with MACGSP6 only for smaller values of p_{gsp}, energy per successful packet received is smaller, as shown in Figure 99.

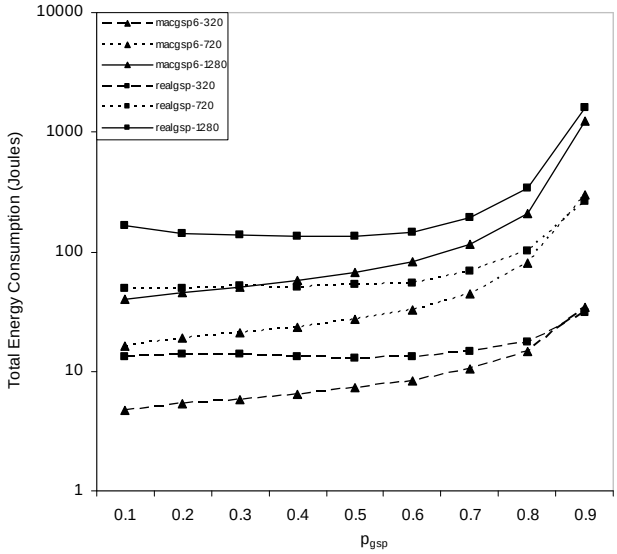

Figure 99. Total energy used per successfully received packet, star topologies.

Figure 100 presents results for the first component of total energy used for transmitting 200 packets. Energy employed in transmission states is higher for REALGSP, in agreement with all previous trends and reflecting additional duplicates traversing the network.

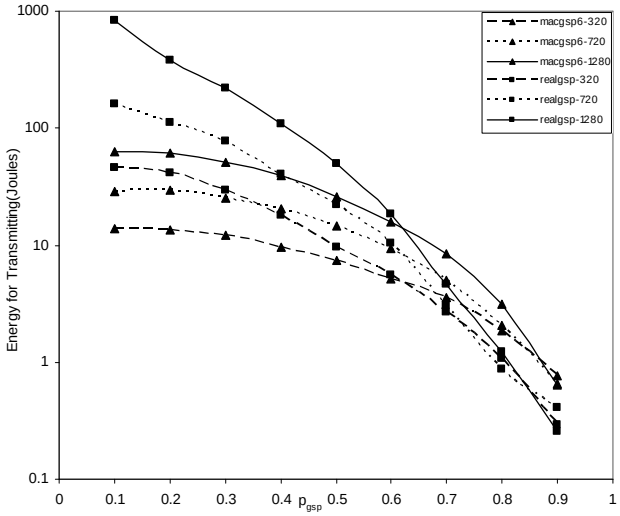

Figure 100. Energy for transmitting, star topologies.

Figure 101 presents energy used in the receiving state. Again, REALGSP uses the majority of energy in this state, corresponding to one order of magnitude more than MACGSP6.

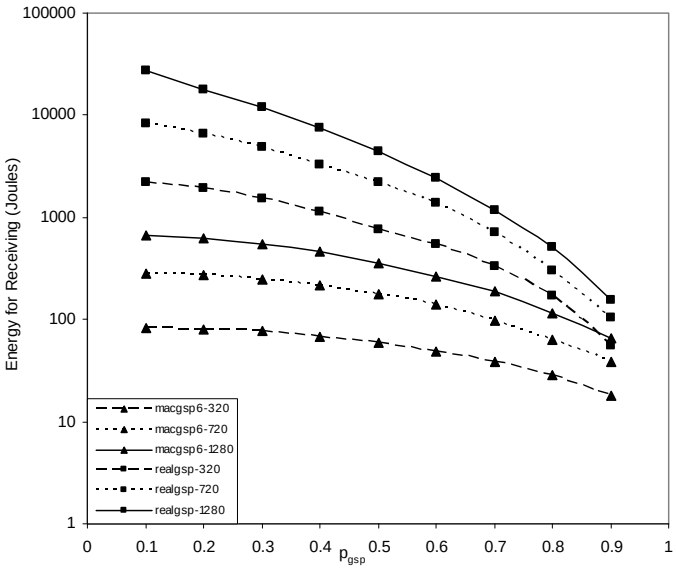

Figure 101. Energy for receiving, star topologies.

Figure 102 shows energy used in the radio off state, with same trends as all previous cases.

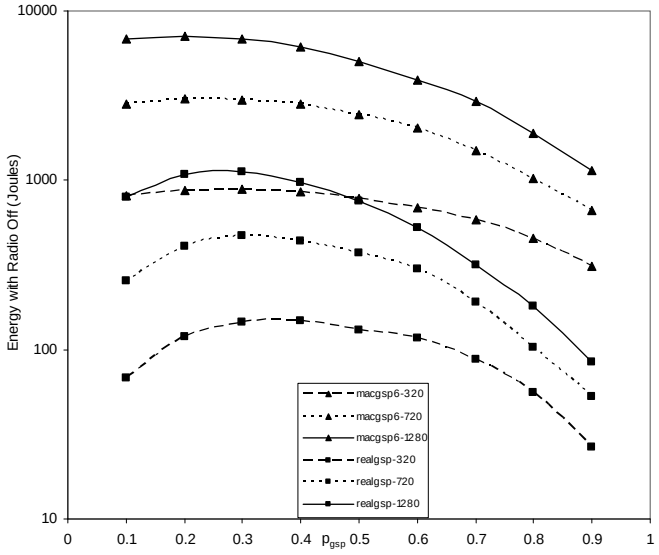

Figure 102. Energy with radio off, star topologies.

Summarizing the results with different topologies, MACGSP6 uses less total energy than REALGSP. The scale of energy savings is such that even in cases where REALGSP delivers more packets, energy per packet successfully received with MACGSP6 is smaller than REALGSP in the majority of cases.

4.6 COMPARISON WITH PREVIOUSLY PROPOSED PROTOCOLS

There have been many protocols proposed for WSNs in both MAC and routing layers. The section presents results obtained for experiments performed using S-MAC and Dynamic Source Routing (DSR). The experiments employed ns2, since both protocols work together in ns2 for small networks. All experiments used the same models for traffic, radio channel, data rate, packet size and energy use as presented in section 3.0 . Other alternatives considered for routing protocols were Directed Diffusion and Flooding, but both implementations in ns2 were not adequate for use with S-MAC under the conditions defined in the preliminary work.

To complement the comparison, this section uses closed form expressions for the best values achievable in the metrics presented throughout this document. The expressions can be evaluated for most networks, knowing parameters such as number of nodes in the network, energy use, time to forward a packet in one hop and minimum number of hops from source to destination. The section presents these expressions for square grids and figures show how close the results are to these best values.

4.6.1 Best values

The expressions assume processing time of the data packet is negligible. TP is the time it takes to transmit a packet in one hop. The following are the expressions for the best possible values.

Packet Reception Probability = 100%

Overhead = 0

The best case of energy use is where just the nodes in the shortest route are involved in the communication process and all other nodes are sleeping. The protocol is ideal and not physically realizable yet for randomly deployed networks, but it can serve as a benchmark to find how close other protocols are to it. Let:

N = total number of nodes in the network

TotEn = Total Energy Use in the network for sending one packet from source to sink (Joules)

EnTx = Total Energy spent in Transmitting State for sending one packet from source to sink (Joules)

EnRx = Total Energy spent in Receiving State for sending one packet from source to sink (Joules)

EnOff= Total Energy used in Radio Off State for sending one packet from source to sink (Joules)

EnTxPk = Energy required for transmitting one packet in one hop

EnRxPk = Energy required for receiving one packet

EnOffPk = Energy spent with radio off during the time it takes to transmit one packet in one hop.

MinHops= Number of Hops in shortest route

TP = Time to transmit one packet in one hop

TTP = Number of TPs in one Gossip Period.

Then:

$$TotEn = EnTx + En\ Rx + EnOff \quad (24)$$

Equations for ideal values are as follows:

$$EnTx = EnTxPk * MinHops \quad (25)$$

$$EnRx = EnRxPk * (MinHops - 1) \quad (26)$$

Equation (20) does not include energy used by the sink in receiving the packet since the sink does not have restrictions in energy use.

$$EnOff =$$

$$((EnOffPk * (N-3)) + (EnOffPk * (N-1)*(TTP-1)))* (MinHops - 1) + EnOffPk*(N - 2) \quad (27)$$

All nodes in the network should be sleeping except for the two nodes actively participating in the communication and the sink. That is the reason why during all hops minus the last one, three nodes do not contribute to energy off counting. During the last hop, only one node and the sink do not contribute to energy off states.

All equations for Energy concern the forwarding of one data packet from source to destination. The experiments sent 200 data packets, so the expressions used included a factor of 200 when computed for presentation in the figures.

The last metric to consider is Delay in TPs (Time it takes to send one Packet). The minimum delay achievable (in TPs) is:

$$Delay = TTP* (MinHops - 1) + 1 \quad (28)$$

Equation 22 considers the packet is delivered at the end of the first TP of the last hop, so all previous hops are counted as the total number of TPs, and for the last hop, only one TP is accounted.

The following figures show the results of the experiments. Note S-MAC labels in the figures mean S-MAC with DSR. S-MAC used 90% duty cycle since it allowed DSR to find routes in networks bigger than 10 nodes. Figures present comparison for square grid networks with 25, 36 and 100 nodes with source in one corner and sink in the center of the network, as all results reported in the previous sections. Figures show results for MACGSP6 running until last packet exits the network, in order to allow a fair comparison with S-MAC and DSR, therefore delay reported for MACGSP6 is the time required to eliminate the last duplicate in the network and delay for S-MAC is the time required to send the packet from source to destination. S-MAC values do not include capture probability, e.g. $\beta=0$, thus results for MACGSP6 also assume $\beta=0$. Experiments consider the same models for radio propagation and energy consumption for S-MAC with DSR and MACGSP6.

Figure 103 shows packet reception probability decreases when network size increases for MACGSP6 and follows the same trend presented before: probability decreases as p_{gsp} increases. However, DSR with S-MAC have 100% probability, regardless network size.

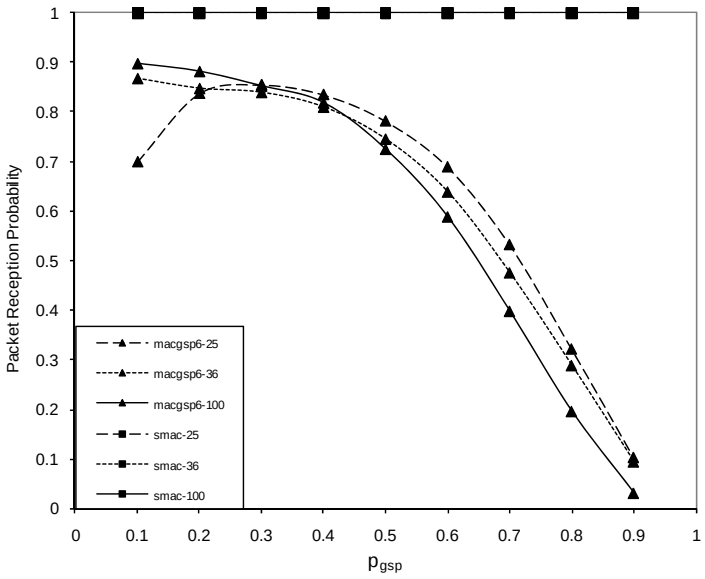

Figure 103. Packet reception probability including S-MAC-DSR.

Figure 104 illustrates the average number of packets received and transmitted at the source, which are not the original data packet. S-MAC and DSR generate different overhead packets such as SYNC, RTS, CTS, ACK and DSR packets used to find and update the route between source and destination. Overhead for MACGSP6 is the average number of duplicate packets received and forwarded by the source for each data packet sent. Figure 104 shows MACGSP6 has one duplicate packet in average, regardless network size or p_{gsp}. According to MACGSP6 functionality, the source transmits one data packet and waits for an implicit ACK. After receiving it, the ACK counts as overhead. When the source receives no ACK, it sends a duplicate, which also counts as overhead. The figure shows this situation at the beginning of the communication process and, since overhead values are a slightly over 1, the number of duplicate packets received until all packets exit the network is very small. S-MAC and DSR have at least four times as many overhead packets as MACGSP6.

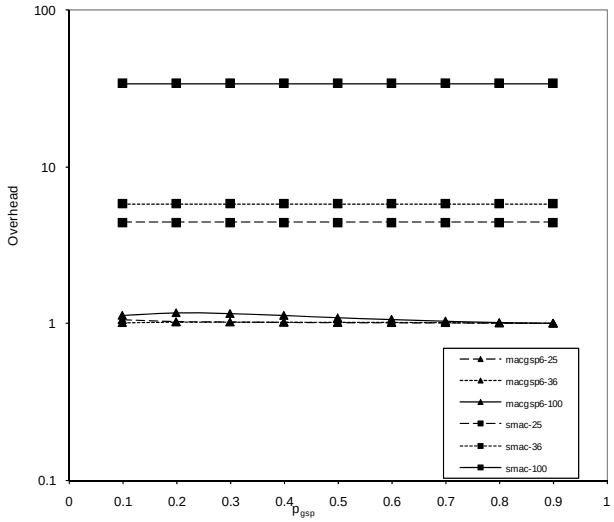

Figure 104. Overhead packets in source including S-MAC-DSR.

Figure 105 presents Delay results. Note delay from S-MAC is one order of magnitude bigger than the ideal protocol. The packets must follow the route found by DSR, which is not always the minimum since network appears disconnected to the routing protocol because of S-MAC functionality. Delay for MACGSP6 is around three times the ideal value. Figure 104 presented the time it takes to eliminate the last packet from the network, which decreases when p_{gsp} increases. Time required to deliver a packet to the destination is closer to the ideal, as previously shown in Figure 38.

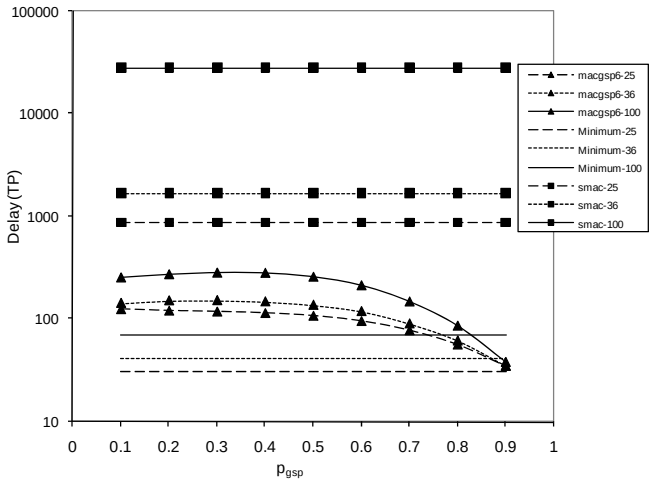

Figure 105. Delay including S-MAC-DSR.

Figure 106 shows total energy use for S-MAC is one order of magnitude higher than MACGSP6. The situation is due to longer times required to deliver the packet, additional overhead and differences in duty cycle: S-MAC is using 90% duty cycle for DSR to be able to find the route. MACGSP6 is using 10% duty cycle, which is adequate since GSP does not explicitly find routes from source to destination. However, S-MAC delivers all packets and MACGSP6 does not.

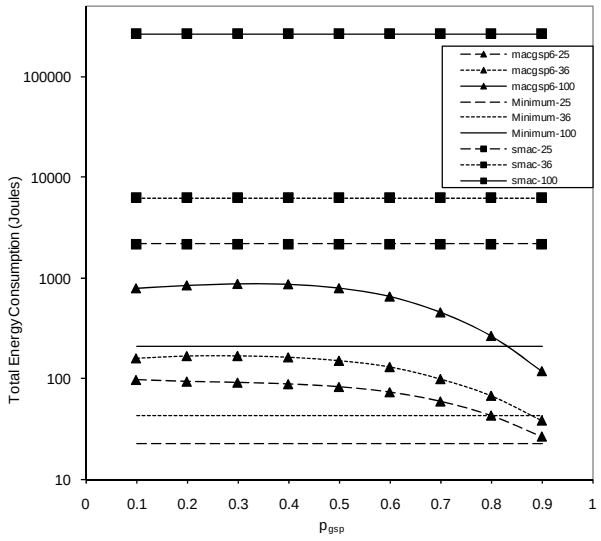

Figure 106. Total energy use including S-MAC-DSR.

Figure 107 illustrates total energy used to successfully send a packet. MACGSP6 in this case uses between three and four times the energy required by the ideal protocol. Nonetheless, MACGSP6 uses one order of magnitude less energy than S-MAC with DSR.

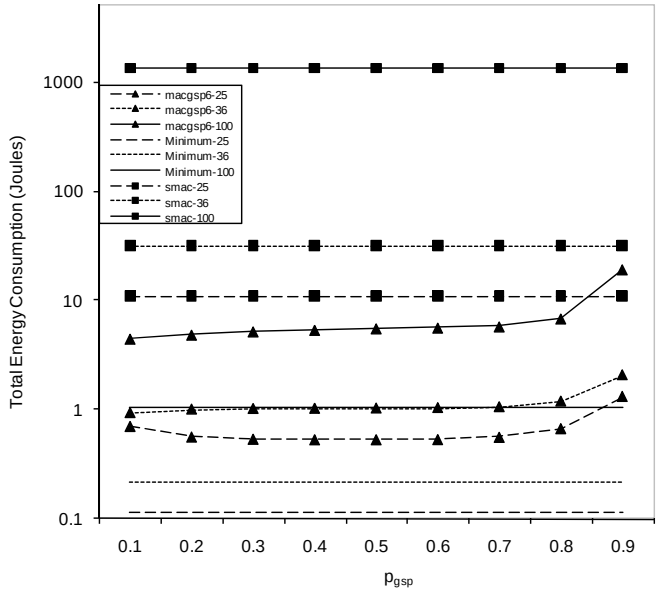

Figure 107. Energy per successful packet received including S-MAC-DSR.

Figures 108 to 110 present the different components of total energy use. Figure 108 displays energy used in the transmission state by all nodes in the network to forward 200 packets. MACGSP6 uses at least 50% of energy used by S-MAC and DSR in this state, thus nodes spend more time transmitting. Note both protocols employ a small percentage of the total energy in this state.

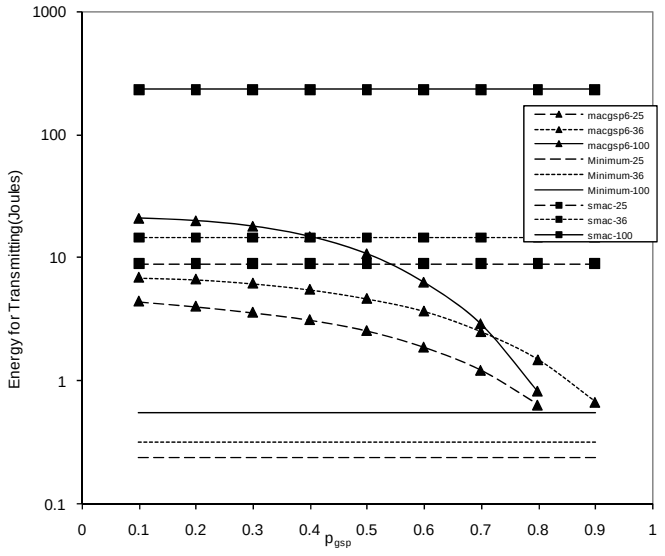

Figure 108. Energy for transmitting including S-MAC-DSR.

Figure 109 shows results for energy used in the reception state. S-MAC has the highest energy use in this state, with 98% of the total energy used even when there are no transmissions, reflecting idle listening, a known issue in S-MAC. However, MACGSP6 spends one order of magnitude less than the total energy in this state.

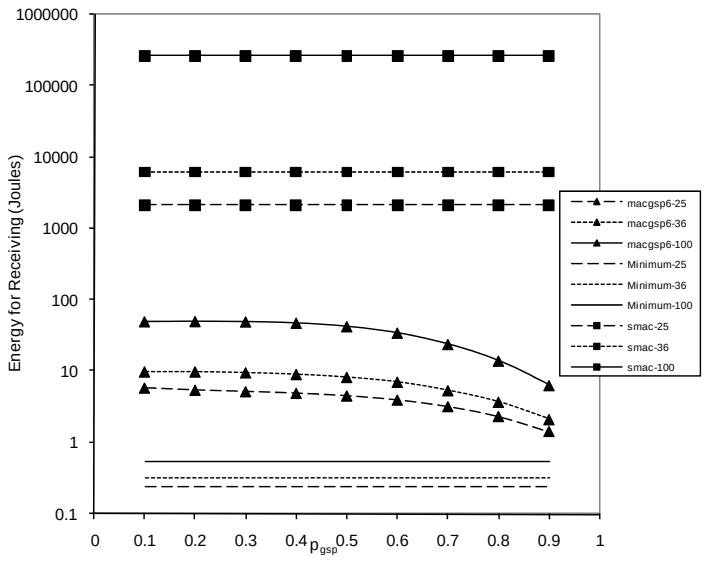

Figure 109. Energy for receiving including S-MAC-DSR.

Figure 110 presents energy used in the radio off state. S-MAC spends less than the ideal case for the 25-node network, showing nodes spend little time sleeping and most time in receiving state. However, MACGSP6 employs around 5% of total energy for transmitting, 5% for receiving and 90% in the sleeping state.

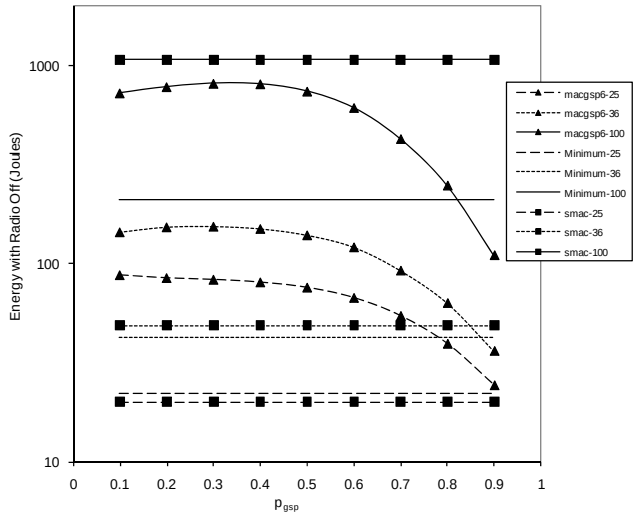

Figure 110. Energy with radio off including S-MAC-DSR.

There is one additional variable of interest for this study, which is the number of collisions. MACGSP6 avoids collisions by using quiescent periods as opposite to S-MAC. Figure 111 illustrates the number of collisions generated in the network for both protocols; MACGSP6 generates around 50% of the collisions created with S-MAC for small networks. 100-nodes networks present a difference of one order of magnitude between collisions using S-MAC and MACGSP6.

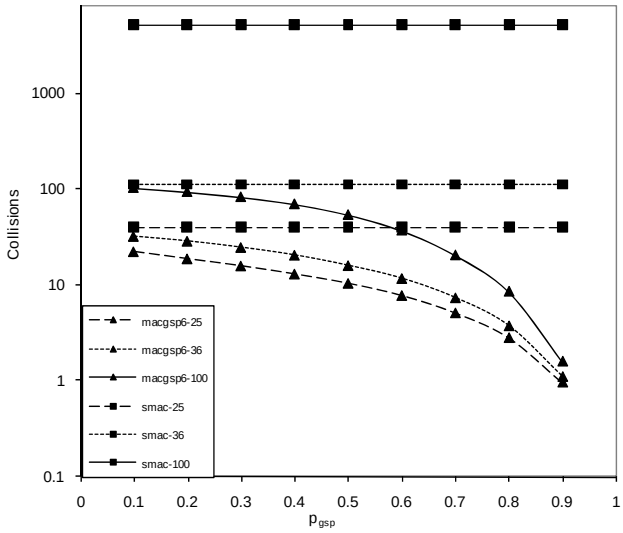

Figure 111. Collisions comparison between MACGSP6 and S-MAC with DSR.

4.7 INFLUENCE OF DIFFERENT FACTORS IN FORWARDING A PACKET

4.7.1 Node density

The definition of Node density in this work is the average number of neighbors in the network. The multiple topologies run for the GSP family of protocols allowed analysis of different node densities and number of hops from source to destination. Table 19 shows a summary of the different topologies, number of nodes, densities and minimum number of hops employed during experiments presented in section 4.5.

Table 19. Summary of densities and hops.

Topology	Total nodes	Density	Min. Numb of Hops
Lattice	240	3.13	17
Lattice	656	3.05	39
Lattice	1136	3.03	62
Line	10	1.13	9
Line	25	1.92	4
Line	100	1.98	49
Line	400	2.00	199
Line	900	2.00	449
Random	100	2.84	8
Random	800	5.90	21
Random	1250	5.88	25
Rectangular	100	3.50	11
Rectangular	500	3.58	51
Rectangular	1000	3.59	101
Square	100	3.60	8
Square	400	3.80	18
Square	900	3.87	28
Star	320	5.48	8
Star	720	5.99	12
Star	1280	6.49	16

Node densities varied from 1.13 to 6.49 and generally speaking, MACGSP6 has better packet reception probabilities when the node density is higher than 2.0 and the number of hops is smaller than 50. The situation allows more neighbors to receive and forward the packets, thus

providing different possible routes in the network. Remember turning off the radio when only one route exists (as is the case in linear topologies) disconnects the network making it difficult for the packet to arrive to the final destination.

Nonetheless, trying to analyze the effect that node density has in forwarding a packet varying the number of nodes and the number of hops at the same time did not provide more quantitative information. To improve the analysis, simulations were created for 25-node networks with a route with minimum number of hops = 4 varying only the node density. The following figures show the results of the same metrics employed during the whole study. To investigate the effect on forwarding a packet, the results presented for MACGSP6 consider metrics until the sink receives the first packet. The densities presented are 1.92, 2.56 and 3.2 for both MACGSP6 and S-MAC with DSR. Results include averages over 40 runs, 200 packets each run and 90% confidence levels.

Figure 112 shows S-MAC and DSR have the same packet reception probability, regardless node density. MACGSP6 has better packet reception probability with higher density.

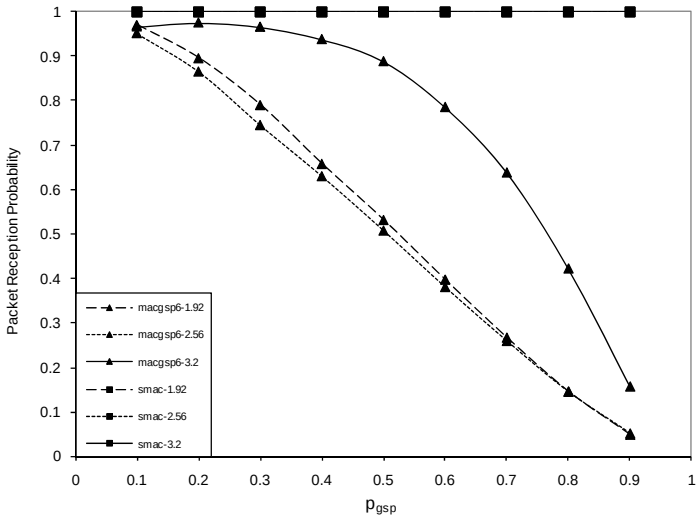

Figure 112. Packet reception probability for different node densities.

Figure 113 exhibits no observable differences for delay in MACGSP6 at 90% confidence levels, showing these cases found a route close to the shortest one. In equation 28:

$$Minimum\ Delay = TTP * (MinHops-1) + 1$$

The minimum number of hops for the experiments is 4 and every hop takes 10 TP to be executed, minus the last one, which takes only 1 TP, so the minimum delay for this configuration is 31 TP. However, S-MAC and DSR take longer times to deliver the packets and they have higher delay when density increases. The intermediate density has the lowest delay, even though the route found by DSR in this case was not four hops (the minimum) but five.

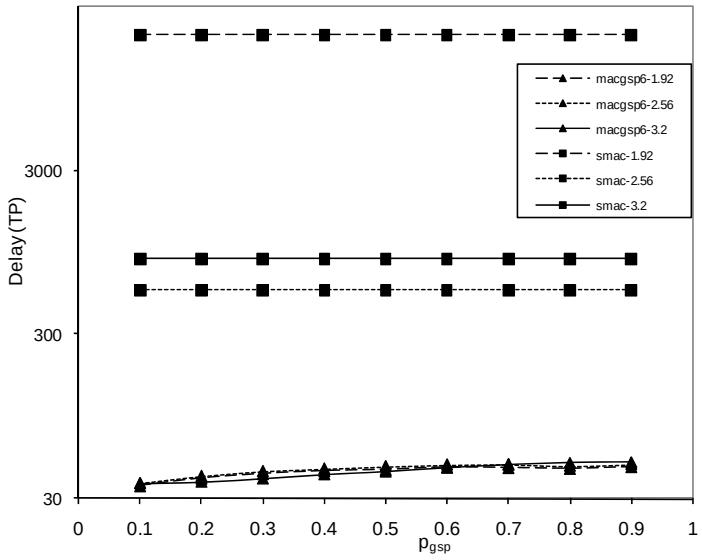

Figure 113. Delay for different node densities.

Figure 114 presents average node overhead. There are no observable differences at 90% confidence levels for MACGSP6 in all densities. S-MAC and DSR have the smallest amount of overhead for the lowest value of node density, however the intermediate density creates the highest amount of overhead.

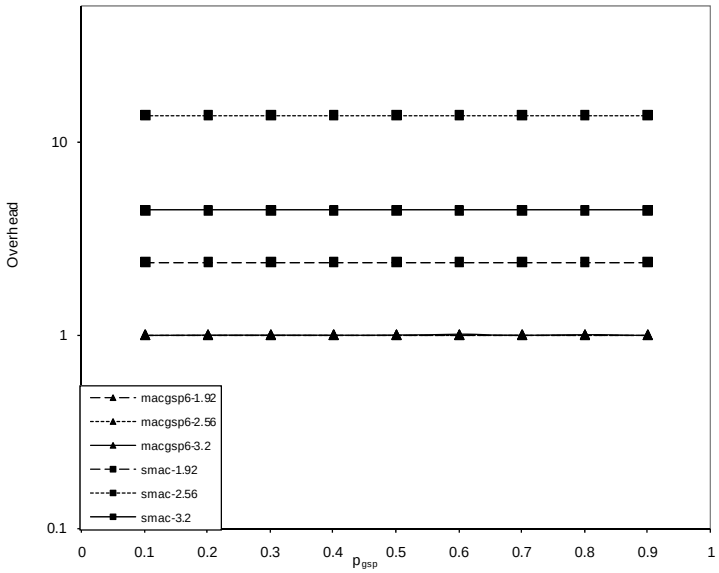

Figure 114. Average overhead for different densities.

Figure 115 illustrates total energy use for MACGSP6 increases with density. The opposite is true for S-MAC and DSR, which may be explained using delay, S-MAC and DSR have the highest delay for the smallest density and energy is accounted during the whole communication process, generating more energy use in more time.

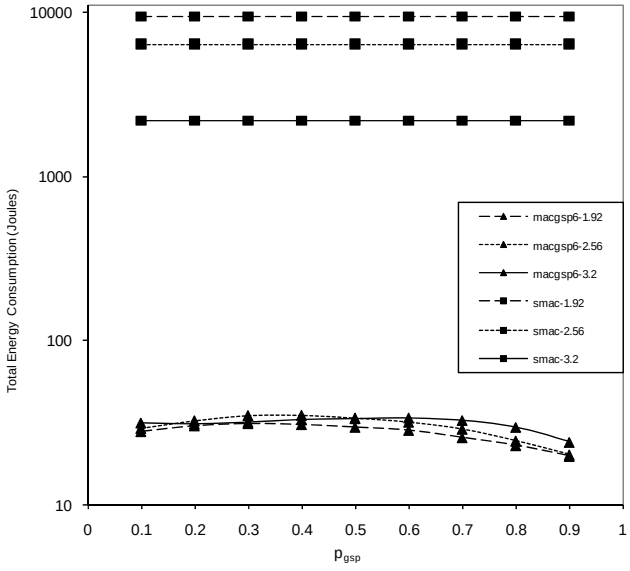

Figure 115. Total energy spent with different densities.

Higher node densities allow MACGSP6 to deliver more packets to the destination, since more neighbors can receive and relay the packet. The situation also produces more duplicates and higher energy use for higher densities. The trend is clearer in Figure 116, which shows energy used in transmission states.

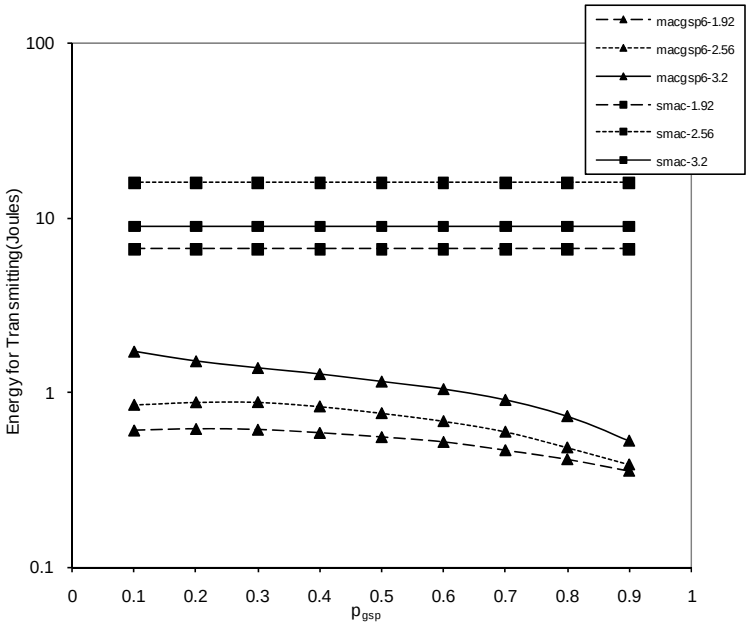

Figure 116. Energy for transmitting with different node densities.

Figure 117 shows energy for receiving with different node densities follow the same trend presented in section 4.6 for cases of S-MAC and DSR: Energy for receiving uses the majority of energy in the network, showing evidence of idle listening. S-MAC and DSR decrease energy used in the receiving state when increasing node density.

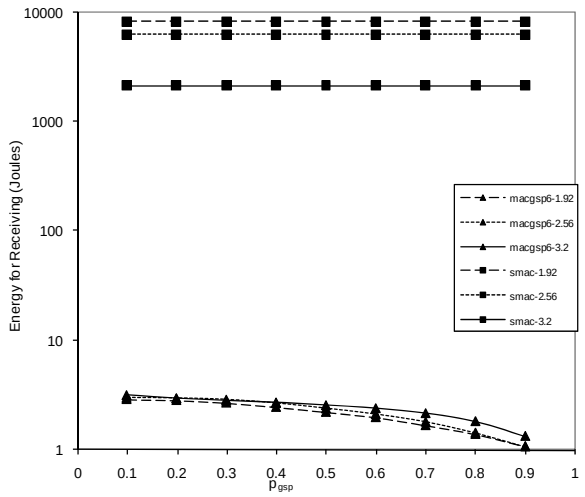

Figure 117. Energy for receiving with different node densities.

Figure 118 presents energy used in the radio off state. The metric along with total energy and energy with radio off present an increasing trend when densities decrease. S-MAC again spends the least amount of time in this state; additionally, energy used with radio off increases when density decreases. MACGSP6 does not show consistent differences at 90% confidence level for different densities.

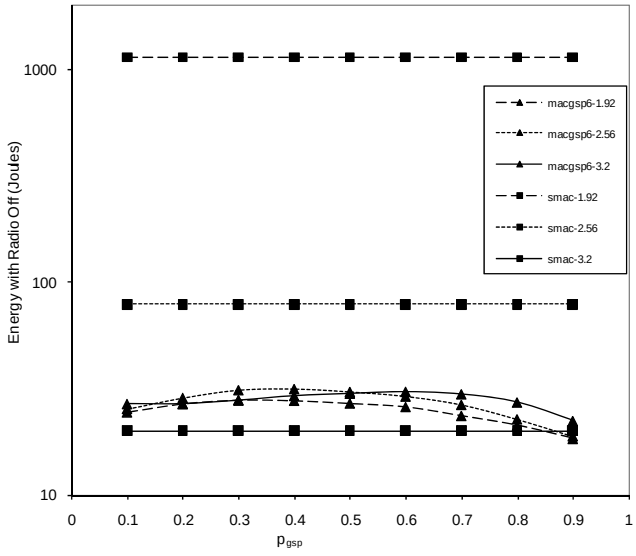

Figure 118. Energy with radio off with different node densities.

4.7.2 Number of hops from source to destination

This section employs the same methodology as section 4.7.1, keeping the same number of nodes and the same density but changing the number of hops. Experiments used 25-node grids for 4, 6 and 8 minimum number of hops from source to sink.

Figure 119 illustrates a clear influence of the number of hops in MACGSP6: a longer path decreases packet reception probability. S-MAC and DSR have the same packet reception probability regardless number of hops.

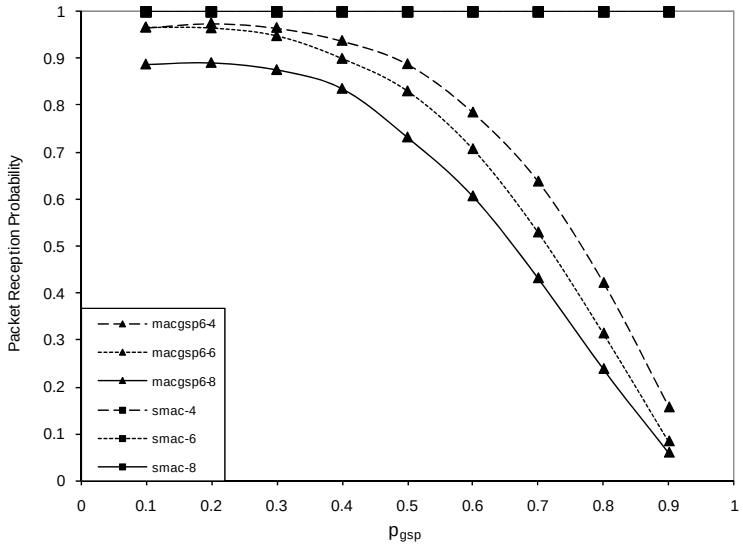

Figure 119. Packet reception probability with different number of hops.

Figure 120 shows delay for delivering one packet from source to the destination. The metric is completely dependent on the number of hops for both protocols.

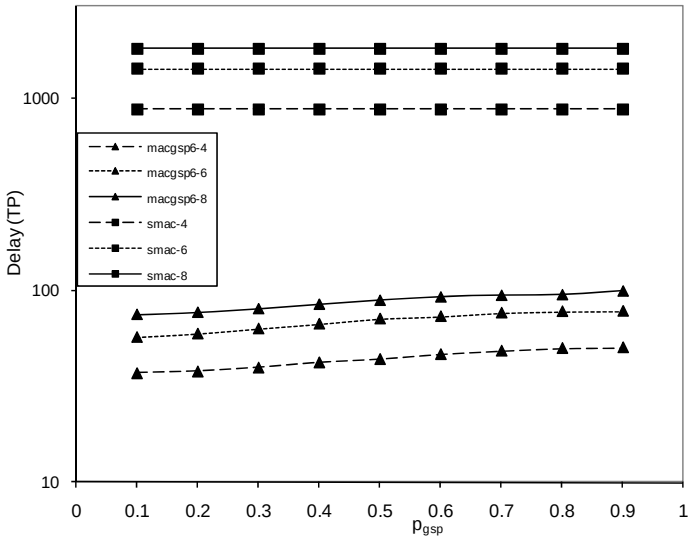

Figure 120. Delay with different number of hops.

Figure 121 presents overhead follows the same trend as shown in previous sections: overhead for MACGSP6 is constant regardless the conditions of the experiments. S-MAC with DSR show more overhead for higher number of hops; however, there are no observable differences for overhead with 6 hops and 8 hops at 90% confidence level.

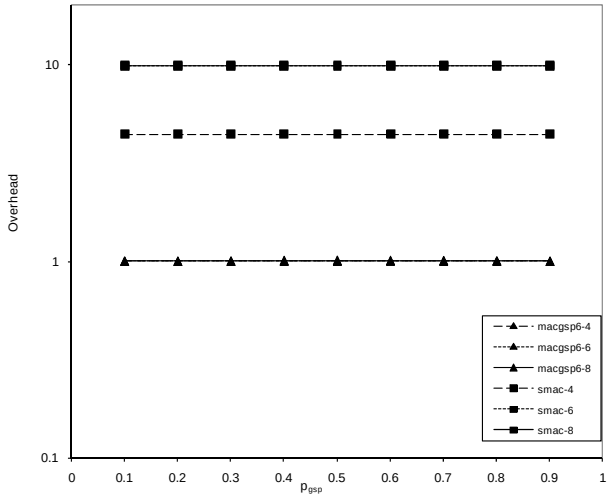

Figure 121. Overhead with different number of hops.

Figure 122 illustrates total energy use increases when the number of hops increases for both protocol. Note there are no observable differences at 90% confidence levels for S-MAC and DSR with 6 and 8 hops.

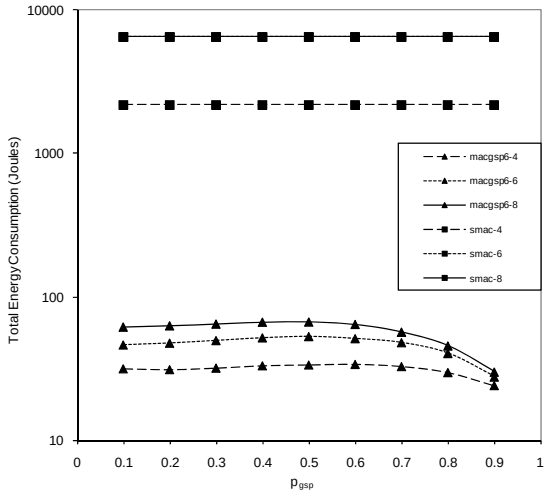

Figure 122. Total energy use with different number of hops.

Figure 123 shows energy for transmitting increases when number of hops increases for both protocols. The result is expected, since the nodes must forward packets through longer paths.

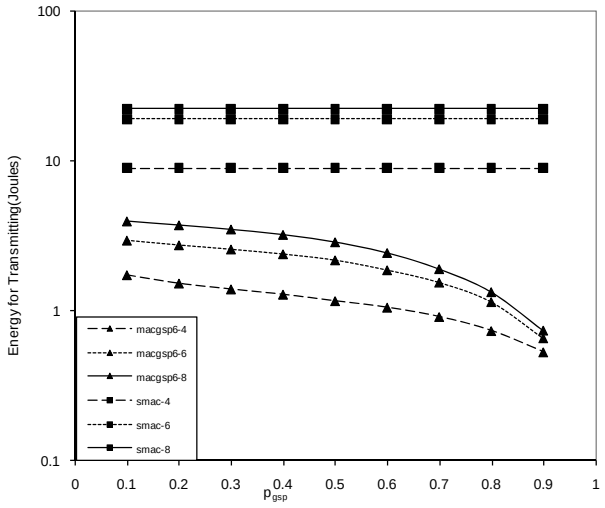

Figure 123. Energy for transmitting with different number of hops.

Figure 124 presents energy for receiving has the same proportion of the total presented in previous section. The trend is increasing energy use in this state with higher number of hops. Note S-MAC and DSR have no observable differences at 90% confidence levels for 6 and 8 hops.

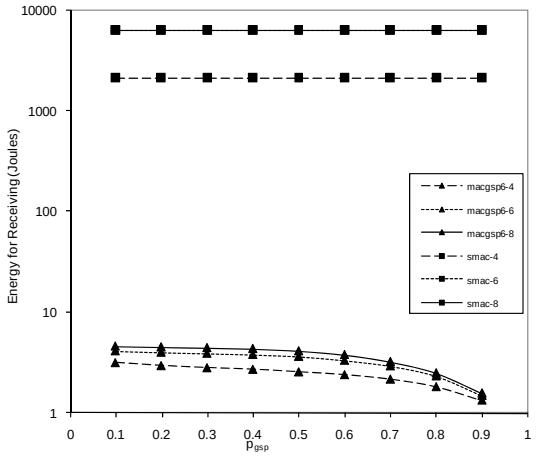

Figure 124. Energy for receiving with different number of hops.

Figure 125 illustrates energy used in the radio off state for MACGSP6 increases when number of hops increases. The metric can be related to delay, since more hops mean more quiescent periods in the network, therefore more energy used in the sleeping state. S-MAC and DSR present the same trend. However, there are no observable differences again for S-MAC with 6 and 8 hops at 90% confidence levels.

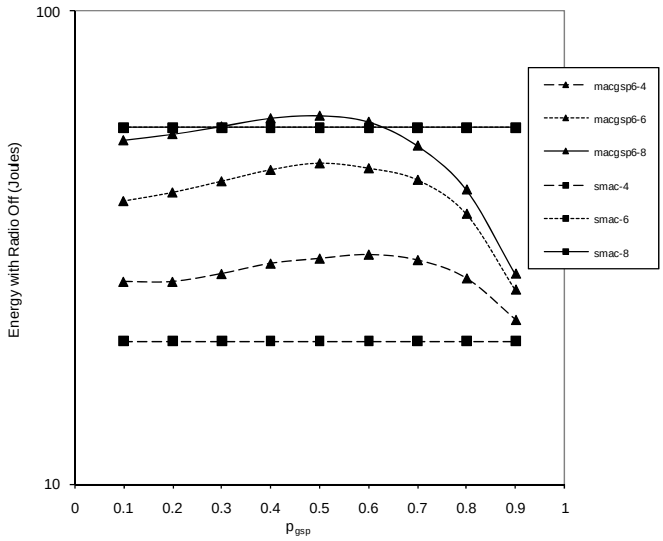

Figure 125. Energy with radio off for different number of hops.

Table 20 illustrates a summary of the influence of increasing node density and number of hops for both sets of protocols. Direction of the arrows show if the affected metric increases or decreases. Green color indicates the effect is desirable for the communication process, while red color shows the opposite idea. Note S-MAC with DSR do not show a consistent trend for overhead increasing the node density, thus the table shows two arrows in opposite directions.

Table 20. Summary of influence of node density and number of hops.

Increasing Parameter	Protocol	Effect on Prec	Delay	Overhead	Total Energy Use
Density	*MACGSP6*	↑	=	=	=
	S-MAC-DSR	=	↓	↑↓	↓
Hops	*MACGSP6*	↓	↑	=	↑
	SMAC-DSR	=	↑	↑	↑

4.7.3 Idle listening time

The values of idle listening time can be estimated by comparing the amount of energy spent in the receiving state against energy for transmitting and energy for receiving. All previous results show MACGSP6 decreases idle listening, while for S-MAC with DSR idle listening is the main source of energy use. The proportion is around the same regardless network size or number of hops from source to destination, or node density: S-MAC with DSR use around 98% of the total energy for receiving and only 2% is employed in transmitting or sleeping, so the majority of energy is used in idle listening. However, MACGSP6 uses 90% of energy in the sleeping state with transmission and reception using 5% each. However, if the number of states in reception where the same as the number of states in transmission, according to the energy model used in this research (see Table 10), energy used for transmission should be twice energy employed for reception. Thus, MACGSP6 still has idle listening time using roughly 2.5% of the total energy in the network.

From the point of view of packet reception probability, S-MAC and DSR forward all packets to the network, provided the routing algorithm can find the route. The ns-2 implementation can only find the route for small number of hops (the maximum found with statistically verified results is square grid of 100 nodes with 8 hops) and using a 90% Duty Cycle. During the research, the ns-2 implementation was tested for 9, 16, 25, 36, 49, 100, 121, 400 and 900 nodes. DSR could not find the route for square grids with more than 100 nodes in a reasonable time. Implementations required a setup time up to 1000 seconds in order to initialize S-MAC, which is another source of energy waste.

However, MACGSP6 is more scalable and worked properly in networks up to 1250 nodes, achieving better results with high network densities and less than 50 hops in the route.

However, MACGSP6 delivers 10% less packets than S-MAC when using p_{gsp} <0.4, as shown in Figure 103. The tradeoff is interesting, since MACGSP6 can use up to one order of magnitude less energy than S-MAC.

4.7.4 Overhead packets in MAC and routing protocols.

The overhead presented throughout this document considers all packets (different than the original data packet) required to transmit one packet from source to destination. Overhead for the GSP family of protocols has two components: first, the duplicate packets generated by forwarding every packet received (created by the routing protocol functionality). Second, the amount of packets sent twice due to the MAC protocol not overhearing an ACK immediately after transmission. The metric presented in sections 4.6 and 4.7 includes packets received by the source as implicit ACKs and packets received as duplicates. Observe the results are always close to one duplicate received in average, regardless number of hops, density or topology and less than 2% of that number corresponds to duplicate packets generated by the routing functionality, coming back to the source after traversing the network. Thus, MACGSP6 eliminates the risk of packet explosion while minimizing overhead required to transmit one packet from source to destination.

However, S-MAC and DSR have different overhead packets, as mentioned in section 4.6 and DSR generates roughly 3% of all overhead packets. Thus, for the maximum total overhead presented in Figure 103 (34 overhead packets received and transmitted by the source to forward 200 data packets), 1 packet corresponds to routing overhead and 33 belong to the MAC protocol. Note also 34 packets is the highest overhead reported for all cases considered with S-MAC and

DSR, corresponding to the maximum number of nodes studied. Networks with 36 and 25 nodes present 3.7 and 4.4 packets of overhead respectively.

Observe MACGSP6 generates less than half overhead packets as S-MAC with DSR in the worst case, despite the fact that MACGSP6 belongs to the flooding family of protocols, just as GSP does. Additionally, MACGSP6 packets only include data. There is no need for addressing (either in MAC or routing layers), routing table creation or maintenance, neighbor information and similar information required by S-MAC and DSR. The result is a protocol that is simple to implement and uses the same data packet as the sole source of overhead.

5.0 CONCLUSIONS AND FUTURE WORK

Section 4.1 shows GSP has higher energy use, packet reception probability and number of duplicates than MACGSP1 and 2, whether these metrics are evaluated until the first packet is received by the sink or until the last duplicate is eliminated in the network. Results in sections 4.2 and 4.3 demonstrate a duty cycle other than 100% can decrease energy use without affecting delay for the GSP family of protocols. The section showed alternatives for implementing a duty cycle in a decentralized manner with no additional overhead bits. Section 4.4 illustrates how the use of overhearing as a source of implicit ACKs improves packet reception probability for all values of p_{gsp}, without an overall increase in energy use. Energy used per packet successfully received with 10% duty cycle is smaller than energy employed with 100% duty cycle. Section 4.5 demonstrates MACGSP6 decreases energy use of GSP if a perfect capture effect exists, irrespective of physical topology. Changes in capture probability did not produce consistent differences at 90% confidence levels for most metrics, so even if a perfect capture effect presents the best possible situation, the differences with other values of β are small. Results in section 4.6 show MACGSP6, which complements GSP, has lower energy use and lower delay compared to protocols such as S-MAC and DSR. In all experiments presented, packet reception probability of S-MAC with DSR is close to 1.0 while MACGSP6 decreases when increasing p_{gsp}. However, energy use from S-MAC is two orders of magnitude larger than the ideal case, so although

MACGSP6 does not deliver all packets, energy spent per successful packet is still lower with MACGSP6 than with S-MAC with DSR.

Section 4.7.1 shows that increasing the node density in the network increases the packet reception probability when using the GSP family of protocols or S-MAC with DSR. Results in section 4.7.2 demonstrate MACGSP6 is more sensitive to the number of hops than it is to node density, as shown in equations 21 and 22. Results from rectangular networks with 500 and 1000 nodes showed one example of this situation: topologies presented node densities of 3.58 and 3.59 respectively, similar values to square grid networks. However, MACGSP6 has smaller packet delivery probability than REALGSP since the minimum number of hops is 51 and 101, respectively.

The metrics considered throughout the document for S-MAC with DSR increase with number of hops; the effect is clearer for delay and energy used in the transmitting state. S-MAC joined with DSR illustrates one example of how protocols can combine to result in inefficient network operation. DSR tries to find routes and S-MAC turns off nodes, increasing the overhead in route computation and requiring additional packets. S-MAC and DSR require time to setup linking and routing tables for the network. The time and energy required for setup depend on network size and number of hops from source to destination; the network cannot transmit user data before this time, thus adding overhead to the communication process.

The study illustrates the importance of considering the influence of different communication layers when designing a communication system. The protocol designed for complementing GSP improved performance regarding energy use and packet reception probability in the majority of cases studied. Results show that using MACGSP6 eliminates packet explosion, thus improving on preliminary work [16].

Previous research used one gossip period as the time it takes to transmit a packet [16]. The research presented here extended the size of the gossip period in order to analyze the effect of Duty Cycle, which reduced energy use. Listening for a packet retransmission during the preamble time decreased idle listening.

Protocols for WSNs must scale with network size and they should be as simple as possible to provide feasible solutions in real life applications. GSP protocols are simple to implement since no routing or geographic information or even addressing is required. However, MACGSP6 performance is best in network topologies having several possible routes from source to destination, higher network density and less than 50 hops from source to sink. However, REALGSP can find the route even in a linear network with hundreds of hops, which is a severe test for any routing protocol.

Future work should study the behavior of MACGSP6 without node synchronization and implement MACGSP6 in hardware devices to verify performance in actual applications. Additional studies should investigate tuning p_{gsp} according to the number of neighbors and number of hops, in order to increase packet reception probability without additional energy use. Research may study methods to dynamically adjust the values of p_{gsp}; for instance, nodes may start working with low values of p_{gsp} and if all packets are relayed (hence a transmitting node always hears ACKs), the node may increase p_{gsp} in order to save energy. Similarly, nodes may also increase or decrease the duty cycle, the number of times one node retransmits the packet (according to the number of ACKs received) and the duration of quiescent periods in proportion to the actual conditions in the network. Additionally, future work should address the behavior of MACGSP6 when relaxing the initial limitations of this research. Studies can employ harsher conditions in the radio channel by including different propagation models or moving nodes.

Also, studies can investigate the performance of MACGSP6 when more than one source transmits simultaneously. Since the protocol does not distinguish among different packets and nodes have to relay every packet received, MACGSP6 should deliver the packets regardless the source originating information. Future work can also include the behavior of MACGSP6 with applications requiring higher traffic levels.

GSP has features that can be useful in deployment of random sensor networks. Nonetheless, WSNs must minimize overhead, and this research shows that combining protocol functionality saves valuable bytes which otherwise must be included in the packet, which in turn requires spending additional energy with them.

BIBLIOGRAPHY

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE,* vol. 40, pp. 102-114, 2002.

[2] X. Ning, R. Sumit, C. Krishna Kant, G. Deepak, B. Alan, G. Ramesh, and E. Deborah, "A wireless sensor network For structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems* Baltimore, MD, USA: ACM Press, 2004.

[3] D. Yupho and J. Kabara, "The Effect of Physical Topology on Wireless Sensor Network Lifetime," *Journal of Networks,* vol. 2, September 2007.

[4] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 1567-1576 vol.3.

[5] T. v. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems* Los Angeles, California, USA: ACM Press, 2003.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10 pp. vol.2.

[7] S. C. Ergen and P. Varaiya, "PEDAMACS: power efficient and delay aware medium access protocol for sensor networks," *Mobile Computing, IEEE Transactions on,* vol. 5, pp. 920-930, 2006.

[8] M. Calle and J. Kabara, "Measuring Energy Consumption in Wireless Sensor Networks Using GSP," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, 2006, pp. 1-5.

[9] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: A MAC protocol for Event-Driven Wireless Sensor Networks," in *Third European Workshop on Wireless Sensor Networks, EWSN 2006*, Zurich, Switzerland, 2006.

[10] X. Hou, D. Tipper, D. Yupho, and J. Kabara, "GSP: Gossip-based Sleep Protocol for Energy Efficient Routing in Wireless Sensor Networks," in *The 16th International Conference on Wireless Communications, Calgary*, 2004.

[11] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* Seattle, Washington, United States: ACM, 1999, pp. 174-185

[12] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems* Baltimore, MD, USA: ACM Press, 2004.

[13] F. Ye, A. Chen, S. Lu, and L. Zhang, "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks," in *Proceedings of the 10th International Conference on Computer Communications and Networks*, 2001, pp. 304-309.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: a Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings ACM MobiCom 2000*, Boston, MA, 2000, pp. 56-67.

[15] A. Tanenbaum, *Computer Networks*, Fourth ed. Upper Saddle River, New Jersey: Prentice Hall PTR, 2003.

[16] M. Calle and J. Kabara, "MAC Protocols for GSP in Wireless Sensor Networks," *Journal of Networks, Academy Publisher* 2008.

[17] M. Calle and J. Kabara, "Influence of Capture Effect on the Effectiveness of GSP in Wireless Sensor Networks," in *The IASTED International Symposium on Distributed Sensor Networks ~DSN 2008~* Orlando, Florida: International Association of Science and Technology for Development 2008.

[18] S. Haykin and M. Moher, *Modern Wireless Communications*, 2005.

[19] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on,* vol. 1, pp. 660-670, 2002.

[20] L. E. Frenzel, *Principles of Electronic Communication Systems, Second Edition*: Mc Graw Hill Glencoe, 2003.

[21] K. Leentvaar and J. H. Flint, "The Capture Effect in FM Receivers," *Communications, IEEE Transactions on,* vol. COM-24, 1976.

[22] K. Whitehouse, A. Woo, F. Jiang, J. A. P. J. Polastre, and D. A. C. D. Culler, "Exploiting the Capture Effect for Collision Detection and Recovery," in *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, 2005, pp. 45-52.

[23] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking* Dallas, Texas, United States: ACM, 1998.

[24] Y.-C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking* Boston, Massachusetts, United States: ACM, 2000.

[25] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. A. J. D. B. Johnson, "The effects of on-demand behavior in routing protocols for multihop wireless ad hoc networks," *Selected Areas in Communications, IEEE Journal on,* vol. 17, pp. 1439-1453, 1999.

[26] J. H. Kim and J. K. Lee, "Capture effects of wireless CSMA/CA protocols in Rayleigh and shadow fading channels," *Vehicular Technology, IEEE Transactions on,* vol. 48, pp. 1277-1286, 1999.

[27] X. Ge, Y. Yang, H.-H. Chen, and Y. Zhu, "Cross-layer Throughput Analysis with Capture Effect in Wireless Local Area Networks," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, 2007, pp. 4161-4165.

[28] A. Zahedi and K. Pahlavan, "Throughput of a wireless LAN access point in presence of natural hidden terminals and capture effects," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, 1996, pp. 397-401 vol.2.

[29] M. Ringwald and K. Romer, "BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks," in *Second European Workshop on Wireless Sensor Networks.*, 2005, pp. pp. 57-69.

[30] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM,* vol. 43, pp. pp.51-58, November 5, 2000 2000.

[31] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *ACM SIGMOBILE 7/01* Rome, Italy, 2001, 2001.

[32] "MPR- Mote Processor Radio Board MIB- Mote Interface/Programming Board User's Manual," in *Crossbow Technology Inc.*, Revision B ed, 2006.

[33] G. Anastasi, M. Conti, A. Falchi, E. Gregori, and A. Passarella, "Performance Measurements of Mote Sensor Networks," in *MSWiM'2004* Venezia, Italy, , 2004.

[34] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the Power Consumption of LargeScale Sensor Network Applications," in *Proceedings of the 2nd international conference on Embedded networked sensor systems* Baltimore, Maryland, USA,: ACM Press, 2004.

[35] D. Brunelli, L. Benini, C. Moser, and L. Thiele, "An efficient solar energy harvester for wireless sensor nodes," in *Proceedings of the conference on Design, automation and test in Europe* Munich, Germany: ACM, 2008.

[36] Y. Ammar, A. Buhrig, M. Marzencki, B. Charlot, S. Basrour, K. Matou, and M. Renaudin, "Wireless sensor network node with asynchronous architecture and vibration harvesting micro power generator," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies* Grenoble, France: ACM, 2005.

[37] T. Sogorb, J. V. Llario, J. Pelegri, R. Lajara, and J. Alberola, "Studying the Feasibility of Energy Harvesting from Broadcast RF Station for WSN," in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, 2008, pp. 1360-1363.

[38] W. Stallings, *Handbook of Computer-Communications Standards* vol. 1. New York: Macmillan Publishing, 1987.

[39] IEEE, *IEEE Std 802-2001 (Revision of IEEE Std 802-1990)*, 2001.

[40] M. Calle, "Energy Consumption In Wireless Sensor Networks Using GSP," in *School of Information Scienes*. vol. Master of Science in Telecommunications Pittsburgh: University of Pittsburgh, 2006.

[41] Y. Sankarasubramaniam, I. F. Akyildiz, and S. W. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, 2003, pp. 1-8.

[42] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *Communications Magazine, IEEE,* vol. 44, pp. 115-121, 2006.

[43] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Networks* vol. 12, February 2006.

[44] C.-K. Lin, V. Zadorozhny, and P. Krishnamurthy, "Efficient Hybrid Channel Access for Data Intensive Sensor Networks," *Proceedings of Third IEEE International Workshop on Heterogeneous Wireless Networks, HWISE,* May 21-23 2007.

[45] "IEEE standard for information technology- telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements Part II: wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001),* pp. i-67, 2003.

[46] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and B. S. Mani, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, San Diego, California, USA, 2005.

[47] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE [see also IEEE Personal Communications],* vol. 11, pp. 6-28, 2004.

[48] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, 2002, pp. 3-1125-3-1130 vol.3.

[49] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Parallel and Distributed Processing Symposium., Proceedings 15th International*, 2001, pp. 2009-2015.

[50] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad hoc wireless networks," in *Mobile Computing*. vol. Kluwer Academic Publishers, T. Imielinski and H. Korth, Eds., 1996, pp. 153-181.

[51] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 1707-1716 vol.3.

[52] A. Safwati, H. Hassanein, and H. Mouftah, "Optimal cross-layer designs for energy-efficient wireless ad hoc and sensor networks," in *Performance, Computing, and Communications Conference, 2003. Conference Proceedings of the 2003 IEEE International*, 2003, pp. 123-128.

[53] L. van Hoesel, T. Nieberg, J. Wu, and P. J. M. Havinga, "Prolonging the lifetime of wireless sensor networks by cross-layer interaction," *Wireless Communications, IEEE [see also IEEE Personal Communications],* vol. 11, pp. 78-86, 2004.

[54] D. Shu, A. K. Saha, and D. B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 1478-1486.

[55] M. L. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 1740-1750 vol.3.

[56] I. Oppermann, L. Stoica, A. Rabbachin, Z. A. S. Z. Shelby, and J. A. H. J. Haapola, "UWB wireless sensor networks: UWEN - a practical example," *Communications Magazine, IEEE,* vol. 42, pp. S27-S32, 2004.

[57] S. De, Q. Chunming, D. A. Pados, M. A. C. M. Chatterjee, and S. J. A. P. S. J. Philip, "An integrated cross-layer study of wireless CDMA sensor networks," *Selected Areas in Communications, IEEE Journal on,* vol. 22, pp. 1271-1285, 2004.

[58] J. Haapola, Z. Shelby, C. Pomalaza-Raez, and P. Mahonen, "Cross-layer energy analysis of multihop wireless sensor networks," in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, 2005, pp. 33-44.

[59] J. Zhu, S. Chen, B. Bensaou, and K. L. A. H. K. L. Hung, "Tradeoff Between Lifetime and Rate Allocation in Wireless Sensor Networks: A Cross Layer Approach," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, pp. 267-275.

[60] C. Partridge and T. J. Shepard, "TCP/IP performance over satellite links," *Network, IEEE,* vol. 11, pp. 44-49, 1997.

[61] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as Optimization Decomposition: Questions and Answers," in *Military Communications Conference, 2006. MILCOM 2006*, 2006, pp. 1-10.

[62] A. Tang, J. Wang, S. H. Low, and M. Chiang, "Network equilibrium of heterogeneous congestion control protocols," in *INFOCOM 2005. 24th Annual Joint Conference of the*

IEEE Computer and Communications Societies. Proceedings IEEE, 2005, pp. 1338-1349 vol. 2.

[63] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *Networking, IEEE/ACM Transactions on,* vol. 10, pp. 232-243, 2002.

[64] J. W. Lee, M. Chiang, and A. R. Calderbank, "Utility-Optimal Medium Access Control: Reverse and Forward Engineering," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1-13.

[65] B. Johansson, P. Soldati, and M. Johansson, "Mathematical Decomposition Techniques for Distributed Cross-Layer Optimization of Data Networks," *Selected Areas in Communications, IEEE Journal on,* vol. 24, pp. 1535-1547, 2006.

[66] Y. E. Ioannidis, "Query optimization," *ACM Comput. Surv.,* vol. 28, pp. 121-123, 1996.

[67] T. Urhan, M. J. Franklin, and L. Amsaleg, "Cost-based query scrambling for initial delays," *SIGMOD Rec.,* vol. 27, pp. 130-141, 1998.

[68] G. Gardarin, F. Sha, and Z.-H. Tang, "Calibrating the Query Optimizer Cost Model of IRO-DB, an Object-Oriented Federated Database System," in *Proceedings of the 22th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc., 1996.

[69] R. Lanzelotte, P. Valduriez, and M. Zait, "On the Effectiveness of Optimization Search Strategies for Parallel Execution Spaces," in *19th Very Large Databases (VLDB) Conference* Dublin, Ireland: Very Large Data Base Endowment Inc., 1993, pp. 493-504.

[70] M. T. Roth, F. Ozcan, L. M. Haas, and Related, "Cost Models DO Matter: Providing Cost Information for Diverse Data Sources in a Federated System," in *Proceedings of the 25th International Conference on Very Large Data Bases*: Morgan Kaufmann Publishers Inc., 1999.

[71] D. Yupho, M. Calle, and J. Kabara, "Longer Network Lifetime when Using Energy Efficient GSP for Wireless Sensor Networks," in *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)* Sedona, AZ, 2007.

[72] ATMEL, *ATmega128 ATmega128L Summary. Rev:2467LA-AVR-05/04* ATMEL, 2004.

[73] Chipcon, *SmartRF© CC1000 Preliminary Datasheet (rev. 2.1), 2202-04-19*: Chipcon AS, 2004.

[74] *TinyOs v.1.1.3, source file CC1000RadioIntM.nc*.

[75] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "Architecture Directions for Networked Sensors," in *Nine International Symposium on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)* Cambridge, MA, USA, 2000.

[76] D. Yupho, "The Effect Of Interactions Between Protocols And Physical Topologies On The Lifetime Of Wireless Sensor Networks," in *School of Information Sciences*. vol. Doctor of Philosophy Pitttsburgh: University of Pittsburgh, 2007.