A General Model for MAC Protocol Selection in Wireless Sensor Networks

Abolfazl Asudeh[1], Gergely V. Zaruba[1], Sajal K. Das[2]

Abstract

Wireless Sensor Networks (WSNs) have become relatively common in recent years with application scenarios ranging from low-traffic soil condition sensing to high-traffic video surveillance networks. Each of these applications has its own specific structure, goals, and requirements. Medium access control (MAC) protocols play a significant role in WSNs and should be tuned to the particular application. However, there is no general model that can aid in the selection and tuning of MAC protocols for different applications, imposing a heavy burden on the design engineers of these networks. Having a precise analytical model for each MAC protocol, on the other hand, is almost impossible. Using the intuition that protocols in the same behavioral set perform similarly, our goal in this paper is to introduce a general model that can help select the protocol(s) that satisfy given requirements from a protocol set that performs best for a given context. We define the Combined Performance Function (CPF) to demonstrate the performance of different category protocols for different contexts. Having developed the general model, we then discuss the models scalability in terms of adding new protocols, categories, requirements, and performance criteria. Considering energy consumption and delay as the initial performance criteria of the model, we focus on deriving mathematical models for them. Previous rules of thumb for selecting MAC protocols support the results extracted from CPF, providing a practical verification for our model. We further validate our models with the help of simulation studies.

Keywords: Wireless Sensor Networks, MAC Protocol Selection, General Model, Mathematical Analysis.

Email addresses: ab.asudeh@mavs.uta.edu (Abolfazl Asudeh), zaruba@uta.edu (Gergely V. Zaruba), sdas@mst.edu (Sajal K. Das)

[1]the University of Texas at Arlington

[2]Missori University of Science and Technology

1. Introduction

Unique characteristics of wireless sensor networks (WSNs), in addition to being mostly application-specific, make traditional network algorithms and protocols unsuitable for them. Specifically: (i) wireless sensor nodes usually have limited resources such as available energy, storage, computation and communication capabilities; (ii) the amount of data transmitted is typically lower than in other networks (e.g., Wi-Fi); and (iii) wireless links are unreliable by nature, with an additional caveat that nodes usually spend a considerable amount of time in a sleep state, saving energy. We also note that the characteristics of sensor networks may be different in different contexts. For example, small sensor networks used in farming have fewer nodes with more resources [1]; traffic load may be significantly higher in multimedia sensor networks [2]; links are more unreliable in underwater sensor networks [3]; whereas at the other extreme, in some WSNs (e.g. the floating sensors project [4]), cell phones are used as sensor nodes and the cellular network provides a centralized infrastructure for communication.

In most WSNs, the medium access control (MAC) sub-layer provides mechanisms and policies for sharing the wireless medium. Clearly, not all MAC protocols are well suited for every situation. MAC protocols for WSNs can be classified in several ways. Some survey articles [5, 6, 7] have focused on traditional taxonomy, i.e., contention-based and reservation-based approaches. However, these classifications do not take the application context of individual sensor networks into account, and hence provide only limited insights. The authors in [8] classify MAC protocols based on their behavior and claim that each category is useful for a different traffic load. Similar behavioral categorization is depicted in [9] by showing the evolution of MAC protocols for wireless sensor networks over the years of 2002-2011.

During the design but before the deployment of a WSN, an important question needs to be answered: which MAC protocol is better for the given application scenario? Since there is a lack of unified analytical models addressing the behavior of MAC protocols under different conditions, it is hard to address this question satisfactorily. Thus, most decisions are made based on questionable ”rule of thumb” engineering principles. (For example, the most common rule of thumb is to employ preamble sampling protocols in low-traffic environments, common active period protocols for medium traffic situations, and scheduled protocols for high-traffic loads.) It could be claimed that using such rules of thumb is enough for making a decision but Example 1 presents two application scenarios that will help us show how difficult such a task may be.

Example 1. Suppose we are looking for a MAC protocol for an environment-monitoring application with the specifications and the network characteristics mentioned in Table 2 (except for the number of nodes, network radius, and packet generation rate). For security reasons, the MAC protocol should prevent overhearing; moreover as the network topology may be unknown we are looking for a distributed protocol. Based on the application, energy consumption is a main concern; however the delay should also be reasonable. Consider the following two scenarios. In the first scenario there are 90 nodes distributed uniformly in a field with the radius of 100 and the average network packet generation rate of 100 packets per second. The network in the second scenario contains 110 nodes and the network radius is 70.

We will show in Section 5 that even slight changes may greatly affect the performance of MAC protocols. For each scenario in Example 1, we will also select a MAC protocol based our current protocol pool and the model we propose in this paper.

The number of proposed MAC protocols for WSNs is large (and still rising); this, in addition to the complexity of some of these protocols, makes it almost impossible to obtain a precise analysis for each one of them. Intuitively, the protocols in the same behaviorally-similar set should have similar performance characteristics. Therefore, if we can decide which set is better for a given situation, we can use a qualitative comparison to find the best match. Using this intuitive assumption, we introduce a general model for selecting MAC protocols for wireless sensor networks. We attempt to make the model scalable as well, so that new sets, protocols, requirements, and performance criteria can be added to it gradually.

Our contributions are summarized as follows:

- The main contribution of the paper is the introduction of a general model for selecting MAC protocols for wireless sensor networks for different network specifications and protocol settings, requirements, and performance criteria importance/cost functions. Our model helps finding the protocol that satisfies the requirements, from the set that performs best for a given situation.

- We define the Combined Performance Function to compound performance analyses under different criteria.

- We show how new protocols, sets, and requirements can be added to the model, making our model future proof.
- Focusing on performance analysis, we consider energy consumption and end-to-end delay as the initial performance criteria, and derive the mathematical performance model for the three categories of MAC protocols mentioned in [8].

We will show in Section 5 that the rules of thumb strongly correlate with the findings based on our model. We also validate our models by performing detailed simulation studies. The initial version of our model with a web user interface is accessible online [10].

The rest of the paper is organized as follows. The general model is presented in Section 2, including the Combined Performance Function (CPF) and the description of model expendability. Section 3 develops energy consumption models used in our analyses. Approximate delay models are derived in Section 4. Section 5 presents the CPF incorporating the the previous two models. Simulation results are presented in Section 6 to validate our models. Related works are briefly discussed in Section 7. Finally, conclusions are drawn in Section 8.

2. General Model

In this section we present our general model for MAC protocol selection. The intuitive assumption behind our model is that if MAC protocols are behaviorally clustered, the protocols in the same category should have similar performance characteristics. Using the categorization presented in [8], Table 1 presents a qualitative comparison between the MAC protocols of different categories, listing major behavioral characteristics that affect their performance. Although Table 1 does not provide the numerical values, it indicates that protocols in the same category have similar characteristics. For example, DMAC [11] is an extension over TMAC [12] (which in turn is an extension on SMAC [13]) that defines a duty cycling chain in order to be tailored to data gathering trees in WSNs. Even though DMAC is not a direct extension of SMAC, looking at Table 1, some may notice that the basic properties of SMAC are still reflected in it. If a protocol is not similar to any of the protocols in any of the categories, it should be separately analyzed; this is further explained in Subsection 2.2.

[3] Cen: Centralized; Dis: Distributed.

[4] S: Synchronization; C: Control Messages; D: Duty Cycling; T: Timing Error; Sch: Scheduling, SP: Setup Phase; P: Preamble;B: Beacon; Pr: Probe.

Protocol	Category	Manner	Scalable	Delay	Energy Consumption Factors			
					Collision free	Idle listening	Over hearing	Overhead
TSMP [14]	ScP	Cen[3]	No	Long	No	Short	No	S, C, D, T[4]
Arisha [15]	ScP	Cen	No	Long	No	Short	No	S, C, D, T
GinMAC [16]	ScP	Cen	Yes	Long	No	Short	No	S, C, D, T
SMACs [17]	ScP	Dis	Yes	Long	Yes	Short	No	Sch, S, C, D, T
Pedamacs [18]	ScP	Cen	No	Long	Yes	Short	Yes	SP, S, C, D, T
AS-MAC [19]	ScP	Dis	Yes	Long	Yes	Yes	Yes	SP, S, C, D, T
SMAC [13]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
TMAC [12]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
NanoMAC [20]	CAP	Dis	Yes	Short	Yes	Yes	Yes	C, S, D
DMAC [11]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
UMAC [21]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
MSMAC [22]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
QMAC [23]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
CL-MAC [24]	CAP	Dis	Yes	Medium	Yes	Yes	Yes	C, S, D
PSA [25]	PSP	Dis	Yes	Short	Yes	Short	Short	P, D
BMAC [26]	PSP	Dis	Yes	Short	Yes	Short	Short	P, D
STEM [27]	PSP	Dis	Yes	Short	Yes	Short	No	C, P, D
MH-MAC [28]	PSP	Dis	Yes	Short	Yes	Short	Short	P, D
DSP-MAC [29]	PSP	Dis	Yes	Short	Yes	Short	Short	B, C, D
RICER [30]	PSP	Dis	Yes	Short	Yes	Short	Short	P, D
WiseMAC [31]	PSP	Dis	Yes	Short	Yes	Short	Short	S, P, D
RI-MAC [32]	PSP	Dis	Yes	Short	Yes	Long for sender	Yes for sender	B, D
X-MAC [33]	PSP	Dis	Yes	Short	Yes	Short	Short	P, D
Koala [34]	PSP	Dis	Yes	Short	Yes	Long for sender	Yes	P, Pr, C, D
CLOA[35]	PSP	Dis	Yes	Short	Yes	Short	Yes	B, D
A-MAC [36]	PSP	Dis	Yes	Short	Yes	Short	Short	Pr, C, D

Table 1: Qualitative comparison of some of existing MAC protocols for wireless sensor networks.

Algorithm 1 presents our MAC protocol selection framework for a given context (ξ represents the network specifications and protocols settings, R represents application requirements, and κ is used for importance/cost coefficients – cf. Subsection 2.1). The algorithm helps determine the categories that have at least one protocol that satisfies the requirements R. Note that a protocol-table describing which protocols satisfy which set of requirements (e.g. mobility, robustness, scalability, and security) is required (for example in Example 1, the requirements are security(over hearing prevention) and having a distributed manner). The algorithm then computes the performance of each category using the CPF (cf. Subsection 2.1) and finds the category C_{opt} that has maximum performance for the context and provided coefficients. Finally, it returns selecting the protocols in the optimal set that satisfy the requirements.

2.1. Combined Performance Function

We use the performance of the representative protocol (the protocol that generalizes the behavior of its set, i.e., the one that represents the high-level approach of the protocols in the same set) of a set of behaviorally similar protocols, as the performance estimate for all the protocols of that set. We define a Combined Performance Function (CPF) that relates the performance measurements into a single scalar measure, based on which the best category of MAC protocols for a given context is selected. The performance criteria positively effecting the

Algorithm 1: MAC protocol selection framework

Input:

- ξ: network specifications and protocols settings
- R: application requirements
- κ: importance/cost coefficients

Output: best matching protocol p_{opt}

$\Psi \leftarrow \{category\ C | \exists p \in C\ s.t. \forall r \in R:\ r[p] = true\}$;
foreach $C \in \Psi$ **do**
  $C.\eta \leftarrow CPF(C, \xi, \kappa)$;
$C_{opt} \leftarrow argmax(C.\eta)$;
$p_{opt} \leftarrow \{p \in C_{opt}\ s.t. \forall r \in R, r[p] = true\}$;
return p_{opt};

performance are placed in the numerator of formula, N, while the negatives are located in the denominator, D. To avoid comparing apples with oranges (i.e., combining the delay seconds; and energy – joules) we need a function (named cost function here) to scale each measurement. Moreover, different criteria may have a different importance in each application. For example, delay may be more important than energy consumption in a fire detection sensor network. Assuming that cost and importance functions are linear, we combine (multiply) them as the importance/cost coefficients (κ). We can now define the CPF as follows:

$$CPF = \frac{\sum\limits_{\forall N_i \in N} \kappa_{N_i} \times N_i}{\sum\limits_{\forall D_i \in D} \kappa_{D_i} \times D_i}$$

We leave specifying the importance/cost coefficients to the network engineers (finding an automated method for specifying the importance/cost coefficients is an interesting topic for future works), as they know how important each of the performance criteria is for the given application. We agree that specifying solid values for the coefficients is not an easy task, however we may start with specifying different values for the parameters, find the assignment ranges for which the results are the same, and select the proper range that the true assignment falls in. For example, suppose that delay and energy consumption are the set of performance criteria, and we are developing a WSN for fire detection. Based on the application, he knows that delay is more important here. Thus, assuming that the measures are normalized, we may start with the values $0.5+\epsilon$ and $0.5-\epsilon$ for the delay and energy coefficients, respectively. Suppose that any delay coefficient between $(0.5, 0.5+\delta]$ will result in selecting the category A, and any value higher than $0.5+\delta$ will results in selecting the category B. Even though, specifying the coefficient values is not easy, identifying the proper range in which the true assignment falls should not be hard.

Due to the nature and the application scenarios of wireless sensor networks, energy consumption and delay are two of the most important criteria. Thus we selected them as the current performance criteria for the model and we will show detailed analysis over them in Sections 3 and 4. Please note that other criteria can also be added to the model later, as we explain in Subsection 2.2, and the model is agnostic to the selected criteria. In the rest of paper we use α and β to represent the importance/cost coefficients of energy consumption and delay, respectively. As both the energy consumption and delay are inversely proportional to the performance, the CPF for delay and energy consumption is as following:

$$CPF = \frac{1}{\alpha E + \beta T_\delta}$$

2.2. Model Expansion

In this paper, we consider the behavioral categorization presented in [8]. Due to the large research interest in sensor networks, we cannot possibly mention and include all protocols, requirements, or criteria. There may also be current or future protocols that do not belong to the current categories and thus will need to be added into their own category.

Expandability is the important feature guaranteeing that new protocols, categories, requirements, and performance criteria can be added to the model progressively. In this section we focus on this aspect and explain how the model can be expanded; Figure 1 shows an outline of adding a new protocol or a new category to the model.

Adding a new category to the model requires the analysis of its representative protocol for every performance criterion in the model. (Considering that the current behavioral categorization of MAC protocols is relatively comprehensive, there is likely only a few categories that will surface and need to be added to the model.) Adding a new performance criterion requires precise analysis of the representative protocols of each category.

Adding a new requirement to the model compels a review of all included MAC protocols to check whether they satisfy the requirement; this means that all

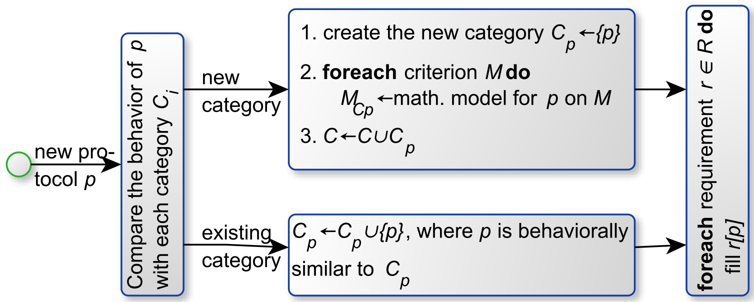


Figure 1: Model expansion with a new protocol.

protocols in the model repository need to be checked. We acknowledge that this can be a daunting task but we argue that new requirements surface with a significantly lower frequency than new MAC protocols. However, the following heuristic can be applied for such cases. Given that we are interested in the protocols that satisfy the application requirements, we can classify the protocols of each category based on the combination requirements they satisfy; then we can select a set of protocols that cover the maximum combinations, check if they satisfy the new requirements, and continue to update the set until we get a set of requirements that satisfy the new requirement and their collection covers the maximum combination of current requirements.

In sections 3 and 4, we will analyze the representative protocols of the scheduled protocols (Time Synchronized Mesh Protocol (TSMP) [14]), common active period protocols (Sensor MAC (SMAC) [13]), and preamble sampling protocols (Preamble Sampling Aloha (PSA) [25]) for the two current performance criteria – i.e. energy consumption and delay. We had two main reasons for selecting the above protocols: (i) since most of the more recent/advanced protocols can be seen as improvements on the basic protocol of their category, the basic protocol may present their common features better, (ii) rather than complicating the analysis, we wanted to make them simpler to show the benefit of the CPF. Table 2 summarizes the notations and the default values used in the analysis.

3. Energy Model

Given the bulk of the research and applications about wireless sensor networks, there are many important performance criteria that should be considered for

[5] The default energy values are computed based on the values and the formulas in [37] and [38].

	Notation	Meaning	Default value
General	L_m	Message length	4000 bits
	L_h	Control messages length	240 bits
	d	Transmission range	20 m
	R	Network radius	100 m
	N	Number of nodes	100
	G	Network packet generation rate	20 $^1/_{sec}$
	B	Bandwidth	256 $^{Kbit}/_{sec}$
	Δ	Node density	0.003 $^1/_{m^2}$
Energy[5]	P_{Idle}	Power consumption in idle state	3 mW
	E_{on}	Required energy to activate the node	3 μJ
	E_{off}	Required energy to deactivate the node	3 μJ
	$E_{send}(d)$	Required energy for transmitting 1 bit with range d	0.3 μJ
	E_{rcv}	Required energy for receiving 1 bit	0.03 μJ
TSMP	T_g	Timing error tolerance	2 ms
	T_{slot}	Length of a slot	0.0275 sec
	T_f	Length of a super frame	3.67 sec
SMAC	dc	Duty cycling active period	0.3 sec
	CW_{min}	Minimum size of collision window	0.01 ms
	CW_{max}	Maximum size of collision window	1 ms
	M	Number of increase to CW_{max}	6
PSA	$T_{Interval}$	Channel check period	0.01 sec
	L_p	Preamble length	4096 bits
	T_{check}	Channel checking duration	0.585 ms

Table 2: Notations explanation and the (default) values used for generating plots.

computing the CPF. However, in order to create the initial model, we selected energy consumption and delay, as two of the most important performance criteria. We note that other important performance criteria (e.g., throughput) could be added to the model, and the model is agnostic to the performance criteria selection.

Sensor nodes consume energy while acquiring, processing, transmitting, and receiving data. Although energy consumption due to computations is not negligible (e.g., when employing data fusion [39]), in general, it is not the task of MAC protocols to incur this computation overhead. On the other hand, since MAC protocols determine physical transmission policies, the largest share of energy consumption is due to transmission/reception of data. Therefore, in the model we focus on the amount of energy consumed for data transmission. The main factors leading to transmission-related energy consumption include:

- Collision: nodes use a shared wireless medium that is unreliable, asymmetric with spatio-temporal characteristics. A receiver within the interference range of a transmitting node, while trying to receive from another sender will experience a collision: as a result, the sender and all active nodes in its transmission range, waste energy for transmission and reception of a garbled-up message, respectively.

- Overhearing: When a sender sends a message to a receiver, all active nodes within its transmission range overhear (receive and decode) the message.

- Idle Listening: This results from nodes spending time actively listening to the channel while there are no transmissions on the channel.

- Overhead (Protocol Overhead): the actual payload is not the only component of a transmission instance. MAC protocols introduce additional fields in their protocol header or may even introduce additional control packets, which generally is referred as protocol overhead.

Relying on the categorization in [8], the representative protocols of the three current categories (TSMP: scheduled protocols, SMAC: common active period protocols, and PSA: preamble sampling protocols) will be analyzed in this section.

We use s to denote the scheduled protocols, c for common active period protocols, and p for preamble sampling protocols in the notations. We also use the indices 1, 2, 3, 4 to denote the energy consumption due to collision, overhearing, idle listening, and overhead, respectively. Each category's energy consumption model would therefore be the summation these four energy usage components:

$$E_k = \sum_{i=1}^{4} E_{k_i},\ k \in \{s, c, p\}$$

To have a general framework and to be independent from any specific energy consumption/battery model, we use the general terms $E_{send}(d)$ for the amount of energy required for transmitting 1 bit within range d and E_{rcv} for the required energy required for receiving 1 bit.

3.1. Scheduled Protocols (ScP)

We derive the energy consumption model for the representative of scheduled protocols, i.e., TSMP [14]. TSMP is a centralized protocol that uses prescheduled super frames containing cells assigned to pairs of nodes. Each super frame is a table of time division slots and frequency division channels (i.e., slot-frequency cells). More precisely, every cell of the table represents a given time slot and a given frequency which is dedicated to one link between a pair of nodes. None of the nodes can have an assigned cell on more than one frequency in the same time slot.

Since each cell is assigned to at most one link, collisions are impossible here (assuming no rapid node movement); and because each node knows its exact wake

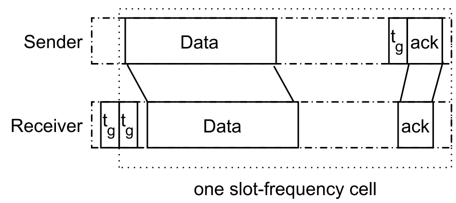


Figure 2: Packet transmission in one slot-frequency cell in TSMP.

up and sleep time, there are no costs associated with overhearing. However this protocol still suffers from idle listening (because the receiver does not know if there is a packet on the channel and has to stay active in its scheduled rounds) and also from overhead (Figure 2).

3.1.1. Idle Listening

Considering the probability of having a packet to transmit in each cell as Pr, the average energy consumed for idle listening in a cell is $E_{S3_{Cell}} = P_{Idle}(1 - Pr) \times T_{Idle}$; where P_{Idle} is idle listening power consumption, and T_{Idle} is the amount of time for which the receiver has to stay active to ensure that there is no packet on the channel.

Assuming that he network packet generation rate is G packets per second (even though we do not use the properties of any specific packet generation distributions here, in order to be consistent, we always assume that packet generation is based on a Poisson process), $G \times T_f$ packets are generated per super frame, where T_f is the length of the super frame in a second. Thus, $Pr = \min(1, \frac{G \times T_f}{N \times N\prime})$; where N is the number of nodes in the network and $N\prime$ is the number links (neighbors) of a node. Having the transmission range d and the node density Δ, $N\prime = \Delta \times \Pi d^2$.

Every receiver has to listen for $2T_g$ seconds to ensure there is no packet on the channel in this slot (see Figure 2). Thus, the total energy (E_{S3}) consumed per second for idle listening in network is derived as following equation:

$$E_{S3} = P_{Idle} \times N \times (\Delta \times \Pi d^2) \times [1 - \min(1, \frac{GT_f}{N \times (\Delta \times \Pi d^2)})] \times 2T_g \times \frac{1}{T_f}$$

3.1.2. Overhead

Receivers have to wake up T_g seconds before the beginning of their slot. As indicated by Figure 2 (because nodes may have T_g seconds of error in their synchronization), the average timing error overhead is $3\frac{T_g}{2}$. Thus the timing error overhead (E_{S4_1}) rate can be calculated as

$$E_{S4_1} = P_{Idle} \times G \times \frac{3T_g}{2}$$

To be synchronized with a maximum allowed ($T_g = 1$ msec) error, it is enough to send sync packets every 48 seconds [14] and two messages are enough for synchronization [40]. Therefore, the amount of energy (E_{S4_2}) used for synchronization overhead is

$$E_{S4_2} = \frac{1}{48} \times 2 \times N \times (\Delta \times \Pi d^2) \times (E_{rcv} + E_{send}(d))L_{Sync}$$

where L_{Sync} is the length of the sync message in bits. Sending and receiving the ACK packets also consume energy (E_{S4_3}):

$$E_{S4_3} = G \times L_{Ack}(E_{rcv} + E_{send}(d))$$

where L_{Ack} is the length of the ACK packet. The duty-cycling overhead (E_{S4_4}) can be computed

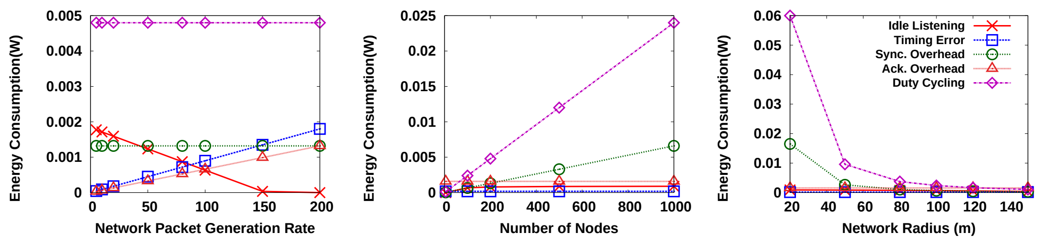
$$E_{S4_4} = 2 \times N \times (\Delta \times \Pi d^2) \times (E_{on} + E_{off})$$

Therefore the energy (E_{S4}) consumption due to the overhead is:

$$E_{S4} = (P_{Idle} \times G \times (\frac{3T_g}{2} + L_{Ack})) + (\frac{1}{48} \times 2 \times N \times (\Delta \times \Pi d^2) \times (E_{rcv} + E_{send}(d))$$

$$\times L_{Sync}) + (L_{Ack}(E_{rcv} + E_{send}(d))) + (2 \times N \times (\Delta \times \Pi d^2)(E_{on} + E_{off}))$$

Figure 3 shows the effects of energy consumption under different conditions for TSMP. (The values in the figures are dependent on the properties of sensor nodes and their antenna, and are produced based on the default values shown in Table 2.) When the population or the network density increases (the number of possible links increases), the energy consumption of duty cycling also increases. However, since nodes check the channel only for a short duration to ensure it is free, they do not spend a lot of energy for idle listening. Thus, the overall energy consumption is relatively low for TSMP.



(a) $D = 20, N = 100$ (b) $D = 20, G = 20, \#ofNeighbors = 4$ (c) $N = 100, G = 20, D = 20$

Figure 3: Energy consumption in TSMP for different a) packet generation rates (packet/sec), b) node populations, c) network radii (network density).

3.2. Common Active Period Protocols (CAP)

The main idea behind this category of protocols is to reduce the energy consumption due to idle listening (when compared to traditional random access MAC protocols). Nodes have a common schedule according to which they periodically sleep and wake up together. While idle listening is avoided, collisions become possible; these protocols are not flexible in duty cycling. The representative protocol in this category is SMAC [13] that uses CSMA/CA random access during active periods. It also uses relative time stamps (rather than absolute) for synchronization; with a recommended sync update message intervals of 10 seconds.

Every newly joining node listens to packets on the channel to see if there is a schedule being transmitted. If not, a node will produce its own schedule and broadcasts it to the network. Nodes with the same sync information form a cluster. Clusters connected by border nodes should work on different schedules to connect virtual clusters together.

Back-off and collision window techniques are used to reduce the collision probability and increase the network throughput. Since all the nodes in a cluster have a common schedule, they all are awake at the same time; and when a node sends a message, all other nodes in the transmission range hear it. Therefore, control messages and duty-cycling are the main overhead resulting in energy loss for this protocol.

3.2.1. Collision

We use the collision probability derived in [41] for CSMA/CA mechanism; this calculation can be adopted here with some adjustments. Transmissions are initiated with the minimum size collision window of CW_{min}; each node waits for a random uniformly distributed back-off time between 1 and CW before a message.

Every time a collision occurs (or it is avoided), nodes double the size of CW until it reaches CW_{max}. Therefore, as derived in [41], the collision probability is

$$p = 1 - (1 - \frac{\lambda}{\mu} \times \frac{1-2p}{1-p-p(2p)^m} \times \frac{2}{CW_{min}})^{N-1}$$

where λ is the packet generation rate, μ is the service rate in packet per second, CW_{min} is the minimum size of collision window, and m is the number of transmission fails that increases the size of the collision window to CW_{max}. In this equation $\frac{\lambda}{\mu}$ is the probability that the channel is not available, whereas $(\frac{1-2p}{1-p-p(2p)^m} \times \frac{2}{CW_{min}})^{-1}$ is the average window size in a saturated network.

By setting $\lambda = G \times dc$, $\mu = B$, we adopt the formula. dc denotes the duty cycle, i.e., the proportion of time that nodes are active together. All the transmissions take place during the active period (which increases the value of λ). The probability of a successful transmission after x trials is $P(x) = (1-p)p^{x-1}$. Thus the expected value of transmissions is $E(x) = \sum_{k=1}^{\infty}(1-p)p^{k-1} = {}^{1}/_{1-p}$ and the average number of collisions for each packet is $\frac{1}{1-p} - 1 = {}^{p}/_{1-p}$. Therefore, the average energy consumption due to the collision (E_{C_1}) is:

$$E_{C1} = G \times L_{RTS} \times ((\Delta\Pi d^2 - 1)E_{rcv} + E_{send}(d)) \times \frac{p}{1-p} \times dc$$

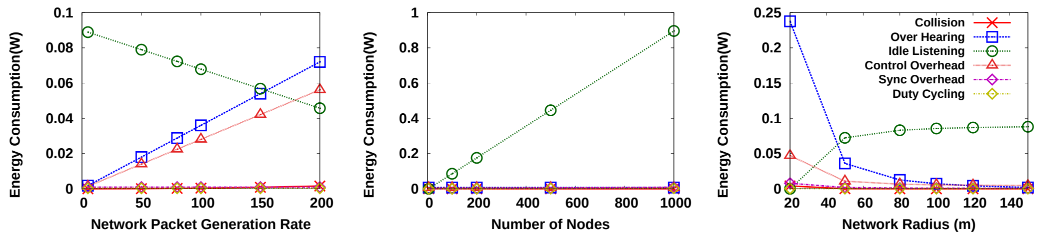
where L_{RTS} is the length of the RTS packet. Nodes that overhear a message (a population of $\Delta\Pi d^2$ nodes) and the sender waste energy during a collision. Since a transmission event can only take place during an active period, the above result contains the dc factor.

3.2.2. Overhearing, Idle Listening, and Overhead

All nodes in the transmission range of the sender overhear the message. The corresponding energy consumption is $E_{C2} = L_m \times E_{elec} \times (\Delta\Pi d^2 - 1) \times G$. Idle listening occurs when the channel is free of transmissions, however nodes are still listening to it. $G \times \frac{d^2}{R^2}$ is the rate of generated packets overheard by each node that can be used for determining the average idle listening time in each node. The energy consumption due to idle listening (E_{C_3}) is thus:

$$E_{C3} = N \times P_{Idle} \times \max(0, dc - (\frac{L_m + L_{rts} + L_{cts} + L_{ack}}{B} \times \frac{G \times d^2}{R^2}))$$

$\frac{L_m + L_{rts} + L_{cts} + L_{ack}}{B}$ is the amount of time required for transmitting a message (we supposed $L_{rts} = L_{cts} = L_{ack} = L_h$ for producing the graphs and for the experiments).



(a) $D = 20, N = 100$ (b) $D = 20, G = 20, \#ofNeighbors = 4$ (c) $N = 100, G = 20, D = 20$

Figure 4: Energy consumption in SMAC for different a) packet generation rates (packet/sec), b) node populations, c) network radii (network density).

RTS, *CTS*, and *ACK* packets are sent for every message and all the nodes in the transmission range of the sender overhear the message. So the overhead of these messages (E_{C4_1}) can be derived as $E_{C4_1} = G \times (L_{rts} + L_{cts} + L_{ack}) \times ((\Delta\Pi d^2 - 1)E_{rcv} + E_{send}(d))$.

Considering that the sync messages are sent every 10 seconds by every node, since all other nodes in the transmission range of the receiver hear it, the overhead of synchronization (E_{C4_2}) is computed as $E_{C4_2} = \frac{1}{10} \times ((\Delta\Pi d^2 - 1)E_{rcv} + E_{send}(d)) \times L_{sync} \times N$.

Nodes wake up at least once a second to decrease the delay. Each time the nodes sleep and wake up, they spend some energy during the transition. The overhead of duty cycling (E_{C4_3}) per second in the network, therefore, is $E_{C4_3} = N \times (E_{on} + E_{off})$. The total energy consumed for the overhead is:
$E_{C4} = (G \times (L_{rts} + L_{cts} + L_{ack}) \times ((\Delta\Pi d^2 - 1)E_{rcv} + E_{send}(d))) + N(\frac{L_{sync}}{10}(E_{rcv}(\Delta\Pi d^2 - 1) + E_{send}(d)) + E_{on} + E_{off})$

Figure 4 shows the energy consumption characteristics of SMAC. As shown, idle listening and overhearing are the main reasons for energy consumption. This is because nodes are awake for a long period of time and overhear all the messages that are in their transmission range.

3.3. Preamble Sampling Protocols (PSP)

In this class of protocols, nodes wake up periodically to check if there is a new message on the channel (Figure 5). Every node determines its schedule independently. Therefore, synchronization is not required here. When a node has a message to transmit, first it has to generate a preamble that is long enough to ensure that the intended destination node will receive it at least once ($Preamble \geq Check\_interval$). Since these protocols have a long preamble, collisions could be very energy consuming.

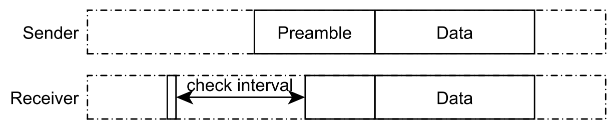


Figure 5: The mechanism of (sender initiated) Preamble Sampling protocols.

3.3.1. Collision

The assumption behind these protocols is that the traffic load (and consequently the collision probability) is low. Here we analyze Preamble Sampling Aloha (PSA) as the representative protocol of this category [25]. If the network packet generation rate is G, the packet generation range around each node during the time required for sending the packet is $G\prime = (G \times \frac{d^2}{R^2}) \times (\frac{L_p+L_m}{B})$. Note, that $\frac{L_p+L_m}{B}$ is the required time for sending a message. No other transmission can be happening in $2 \times (transmission\, time)$ in order to have the current transmission successfully completed. Thus, the probability of generating x messages during a message transmission is $Pr[x] = \frac{e^{-2G\prime}(2G\prime)^x}{x!}$. The probability of a successful transmission after x attempts is given by $P(x) = e^{-2G\prime} \times (1-e^{-2G\prime})^{x-1}$. Therefore, the expected value of transmission attempts is

$$E(x) = \sum_{k=1}^{\infty} k \times e^{-2G\prime} \times (1 - e^{-2G\prime})^{k-1} = e^{2G\prime}$$

and the expected value of collision per message is $e^{2G\prime} - 1$.

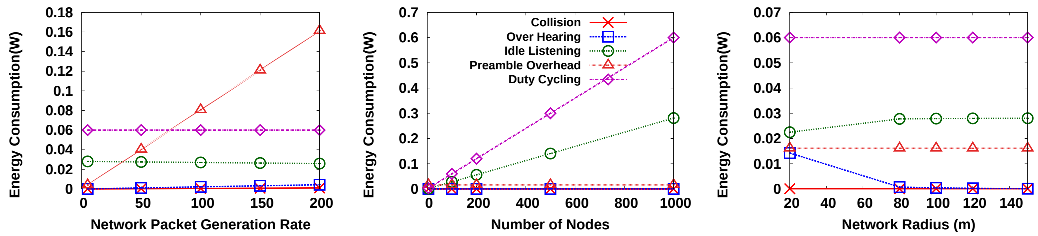
The receiver has to wait for ${}^{L_p}/_2$ seconds on average, before the preamble transmission is finished and data transmission is started. So, the sender has to send $L_p + L_m$ bits for every packet and receiver has to the receive ${}^{L_p}/_2 + L_m$ bits. Thus the energy consumption due to collision (E_{P1}) in PSA is:

$$E_{P1} = (e^{2G\prime} - 1) \times (E_{rcv}(\frac{L_p}{2} + L_m) + E_{send}(d)(L_p + L_m))$$

3.3.2. Overhearing, Idle Listening, and Overhead

For a given message, non-destination neighbors overhear $T_{Check} \times B$ bits of preamble during their check interval. Since T_{Check} is small, the energy consumption of overhearing is not significant. The overhearing energy consumption (E_{P2}) can be derived as $E_{P2} = T_{Check} \times B \times E_{rcv} \times (\Delta\Pi d^2 - 1) \times G$.

Idle listening occurs during check intervals, when the channel is unoccupied. The number of channel checks per second is ${}^{1}/_{T_{Interval}}$. The rate of packets generated in the transmission range of a given node is $\frac{G \times d^2}{R^2}$. Therefore, every node



(a) $D = 20, N = 100$ (b) $D = 20, G = 20, \#ofNeighbors = 4$ (c) $N = 100, G = 20, D = 20$

Figure 6: Energy consumption in PSA for different a) packet generation rates (packet/sec), b) node populations, c) network radii (network density).

is in the idle listening mode for $max(0, (\frac{1}{T_{Interval}} - \frac{G \times d^2}{R^2}))$ seconds. The energy consumption of idle listening (E_{P3}) is then

$$E_{P3} = N \times P_{Idle} \times T_{Check} \times max(0, (\frac{1}{T_{Interval}} - \frac{G \times d^2}{R^2}))$$

Although T_{Check} is short, since the amount of time that the preamble is in the channel has to be at least equal to $T_{Interval}$, the number of channel checks is significant.

Senders use a long preamble in PSA before sending the message. The receiver also has to listen to ${}^{L_p}/_2$ bits of preamble, in average. Thus, the overhead of preamble (E_{P4_1}) is calculated as $E_{P4_1} = G \times ((\frac{E_{rcv} \times L_p}{2}) + E_{send}(d) \times L_p)$. The number of check intervals in a second is ${}^{1}/_{T_{Interval}}$. Therefore, the energy consumption due to the duty cycling (E_{P4_2}) overhead is $E_{P4_2} = N \times \frac{1}{T_{Interval}} \times (E_{on} + E_{off})$. The energy consumption of overhead (E_{P4}) is $E_{P4_1} + E_{P4_2}$.

$$E_{P4} = (G \times ((\frac{E_{rcv} \times L_p}{2}) + E_{send}(d) \times L_p)) + (N \times \frac{1}{T_{Interval}} \times (E_{on} + E_{off}))$$

Figure 6 shows the energy consumption due to the above reasons under varying network conditions in PSA. Duty cycling, the overhead of preamble transmission, and idle listening are the dominant reasons of energy consumption in this protocol.

4. Approximate Delay Model

End to end delay is defined as the time between the instant a packet is passed to the network protocol stack until it gets delivered to the same level protocol (of the protocol stack) in the destination. Modeling such delay is problematic. The problem lies in modeling packet generation; as depending on the complexity of the network model, packets can be generated at the session, network, data link, or MAC layers. When investigating MAC protocols, we usually assume a random process that is generating packets to be transmitted by the MAC layer (without modeling the upper layers). Delay in general is the sum of the queuing delays at each layer and the transmission. In our model we can only consider the queuing delay at the MAC layer, i.e., from the time a packet is passed to the MAC layer for transmission to the time it is delivered (assuming only a single packet is stored at the MAC layer at any time). In addition we will only look at one-hop delays and thus will not consider the diameter of the network.

4.1. Scheduled Protocols

We consider that a packet can be generated any time during the super frame. In the best case the packet is generated exactly at the beginning of its corresponding cell while in the worst case the packet is generated right after the cell belonging to the node has started. Because scheduled protocols are not random access, the collision probability is zero; i.e., assuming that no transmission errors (other than self-interference) occur, it is guaranteed that the packet will be transmitted over the channel successfully in the first upcoming corresponding cell. So the average channel access delay is $\frac{T_f}{2}$ (considering TSMP as the representative protocol of this category). Since the packet is transmitted in one cell, the packet transmission delay is T_{slot}. Therefore, the delay can be modeled by

$$T_{\delta_s} = \frac{T_f}{2} + T_{slot}$$

One point to take into account is that in protocols with centralized scheduling, "finding a collision-free schedule is a two-hop coloring problem" [8]. The other issue in scheduled protocols is the size of the super frame. Adding a new node to the network adds several new links (depending on the network density and transmission range), each requiring a specific cell. The size of the super frame is the main reason for delay in this protocol. For example, as mentioned in [14], "with a 10 ms slot, a cell in a 1000-slot super frame repeats every 10 s".

4.2. Common Active Period Protocols

The activity of each node is divided into active and inactive periods in this protocol. The packets generated during the inactive period have to wait until the node is active. In average, the packets generated during the inactive period have to wait for ${}^{1-dc}/_{2}$ seconds. The portion of packets generated during the inactive period is $1 - dc$. As soon as the node becomes active, packets can be transmitted to the destinations. To simplify the analysis, in this paper we do not consider the number of back-offs into account and assume that the packet is successfully transmitted, if the channel is available. Based on Equation (1), the expected number of trials for transmitting a packet is ${}^{1}/_{1-p}$, where p is the collision probability. RTS-transmission time is spent for each collision; The packet transmission time has also to be added to the formula. Hence, the approximate delay model for the common active period protocols is given by:

$$T_{\delta_c} = (1 - dc) \times \frac{1 - dc}{2} + \frac{\frac{L_{RTS}}{1-p} + L_m}{B}$$

4.3. Preamble Sampling Protocols

These protocols do not feature carrier sensing and the packet is placed in the channel as soon as it is generated. In addition, even if collision occurs, the sender finishes transmitting the entire packet. We assume that there is a feedback informing the sender whether the data has been received. With such a feedback, the expected value of trials is calculated with the help of Equation (2). The approximate delay model can be derived as:

$$T_{\delta_p} = e^{2G\prime} \times \frac{L_p + L_m}{B}$$

5. Combined Performance Function

The next step after deriving performance models for each criterion and each category, is computing the CPF. Figure 7 shows the CPF of the current categories of protocols for $\alpha = {}^{10}/_{11}$ and $\beta = {}^{1}/_{11}$ under various network conditions and settings. With these CPF settings we can see the intuitive rules: preamble sampling protocols have a better behavior when the network packet generation rate is low. Scheduled protocols show a better performance when the number of nodes is low in the network. However, their CPF decrease rapidly when the network population increases. For medium packet generation rates, common active period protocols are considered to be the best choice.

Let us now revisit Example 1 (cf. Section 1). We use the protocol pool and information presented in Table 1. The requirements R are "over hearing avoidance" and having a "distributed" manner. Thus Algorithm 1 dictates: $\Psi = \{\text{ScP,PSP}\}$. Given that energy consumption is the main concern in this example, we have chosen the values $\alpha = {}^{10}/_{11}$ and $\beta = {}^{1}/_{11}$, i.e., each joule lost is as costly/important as

	ScP	CAP	PSP
Case 1	9.22	7.47	6.68
Case 2	3.22	5.16	5.92

Table 3: CPF comparison between aforementioned scenarios in Example 1.

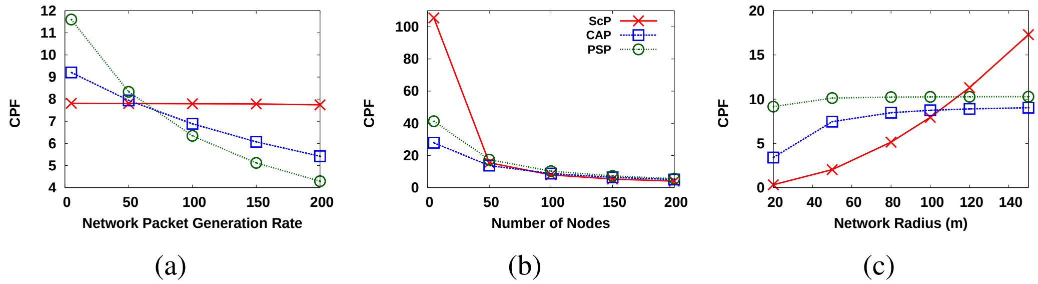


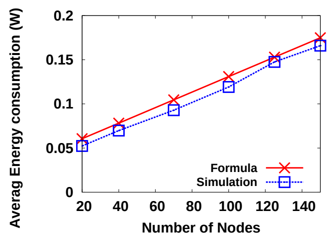
Figure 7: CPF ($\alpha = \frac{10}{11}$, $\beta = \frac{1}{11}$) for varying a) packet generation rates (packet/sec), b) node populations, c) network radii (network density).

10 seconds of delay. Table 3 presents the CPF of these categories for the two scenarios of the example (we also added a column for CAP). The table illustrates that ScP is better for Scenario 1 and PSP for Scenario 2. Finally, based on the requirements R, and considering Table 1, *SMACs* and *AS-MAC* are selected for Scenario 1 while *STEM* is selected for Scenario 2.

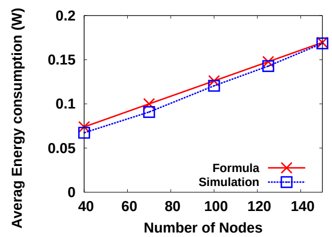
To make our CPF model more widely available, we have created an online calculator that can be used to determine performance characteristics of MAC protocols, thus enabling WSN designers a more scientifically grounded reasoning as to which MAC protocol(s) are most well suited for their particular application scenarios. This tool can be found at [10].

6. Simulation Study

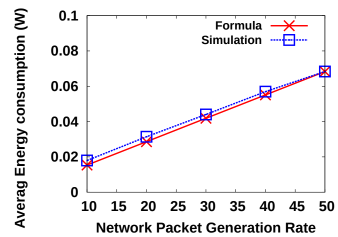
In order to verify the analytical energy and delay models derived in previous sections, we devised a simulation study using a discrete event simulator. Simulation will enable us to access and modify the underlying parameters of protocols as well as network scenarios. Thus we can compare the results obtained from simulation experiments to the values predicted by our analytical model. Each data point represents an average of at least 50 runs; more precisely, for each data point enough simulations are run to claim at least 95% confidence that the relative error is less than 5%. Nodes are randomly deployed in a $100m \times 100m$ area, each with a $20m$ transmission range. In order to reduce the simulation burden, we have



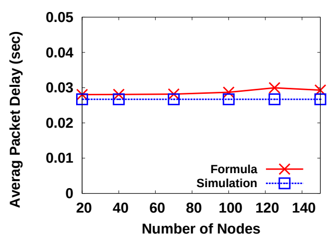
(a) PSA - Energy



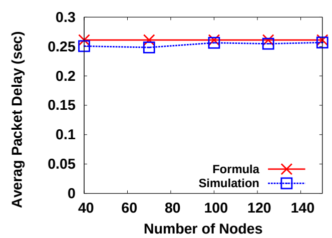
(b) SMAC - Energy



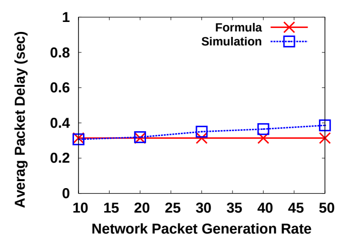
(c) TSMP - Energy



(d) PSA - Delay



(e) SMAC - Delay



(f) TSMP - Delay

Figure 8: Model prediction and simulation result comparison with regard to Energy Consumption (first row) and Delay (second row).

used a custom built C++ discrete event simulator. We acknowledge that there are simulation packages that model WSNs, however each of these simulation packages serve a general purpose and thus have their own idiosyncrasies to overcome. As our goal here was to validate our mathematical models, we elected to program our own simulations that way ensuring that only relevant parts and to the required detail are modeled.

The packet generation follows a Poisson point process with a rate of $\lambda = 20$ packets per second with an available channel bandwidth of $B = 256kbps$. We use the parameters (except the network density) that were presented in Table 2. Figure 8 shows the simulation results versus model prediction for the representative protocols for CAP, PSP, and ScP. The first row shows the plots for energy consumption and the second row presents the plots for delay.

The simulation results in Figure 8a diverge less than 7%, further validating model predictions for PSA energy consumption. Figure 8d compares the predicted average packet delivery delay of PSA as obtained by simulations and the delay model. Although delay due to queuing is not considered in the analytical model but is an integral part of the simulation study, the results (represented by the lines) are close to each other. This is due to the queues of the nodes being of insignificance

$1 \to 2$	$7 \to 8$	$2 \to 4$	$2 \to 3$	$1 \to 9$	$6 \to 8$	$3 \to 6$	$6 \to 9$	$2 \to 6$	$5 \to 9$	$2 \to 7$	$5 \to 10$	$3 \to 9$	$2 \to 9$	$2 \to 10$
$3 \to 4$	$9 \to 10$	$1 \to 10$	$4 \to 5$	$5 \to 7$	$1 \to 4$	$4 \to 7$	$7 \to 10$	$3 \to 7$	$6 \to 10$	$3 \to 8$	$1 \to 7$	$4 \to 10$	$3 \to 10$	$3 \to 5$
$5 \to 6$	$1 \to 3$	$6 \to 7$	$8 \to 9$	$8 \to 10$	$2 \to 5$	$5 \to 8$	$1 \to 5$	$4 \to 8$	$1 \to 6$	$4 \to 9$	$2 \to 8$	$1 \to 8$	$4 \to 6$	$7 \to 9$

Table 4: half of the TSMP super-frame created for the experiment, where $n = 10$; the other half is the same but the link directions are from right to left.

G	20	30	40	50	60	70	80
$^{T_e}/_{T_a}$	0.98	0.9	0.86	0.8	0.75	0.68	0.6

Table 5: Estimated delay divided by the actual simulation delay for TSMP.

at low loads.

Figures 8b and 8e show the results obtained by simulation versus those coming from the model prediction for SMAC (the representative common active period protocol). We focused on the steady state, assuming that nodes already have agreed on a schedule. Figure 8b shows the average energy consumption per second obtained by simulation and our model. The maximum difference is found to be below 6%. Figure 8e presents results for average packet delivery delay obtained from both simulations and the analytical model. Again, since the node queues are mostly empty, the simulation results validate the approximate delay model. Given that TSMP uses centralized prescheduling, we created a schedule for a network which contains 10 connected nodes. The super frame contains 3 rows (frequency division) and 30 columns as shown in Table 4 (the table shows only half of the slots as the other half is similar with reversed transmission directions). The length of the super frame is $0.588\,sec$, nodes are randomly deployed in a $14m \times 14m$ area, and the transmission range is $20m$ (other parameters are as listed in Table 2). Figure 8c show the simulation results for average energy consumption per second in scheduled protocols as well as the results obtained from our energy consumption model. The simulation results validate our derived energy consumption model. Figure 8f compares the average packet delivery delay between simulation results and our model derived estimate. There is a noticeable increase in the difference between the estimate and actual delay as the packet generation rate increases. That (as explained in Section 4) is an artifact of the analytical model not being aware of the queuing delay. In order to further investigate the effect of queuing delay, we kept increasing G, and divided the estimated value for the delay (T_e) by the actual simulation result for the delay (T_a), as is shown in Table 5. We note that when $G = 80$, most (8 out of 9 in average) of the links have data available for transmission, representing a high traffic load in the network (queuing delay is not negligible here).

	PSA	SMAC	TSMP
Delay	0.094	0.263	0.325
Energy Consumption	0.033	0.025	0.016

Table 6: The average values for the delay and energy consumption for TSMP where $n = 10$, $G = 5$.

To show the advantage of using our analytical model we have conducted an additional experiment. Being able to reuse the schedule we generated for TSMP, we considered the situation where there are 10 nodes in the network. We also assumed that every node, in average, generates one packet every two seconds – i.e. $G = 5$. Selecting the coefficient values to be $\alpha = {}^{10}/_{11}$ and $\beta = {}^{1}/_{11}$, the model suggests to choose a preamble-sampling protocol. Looking at the simulation results, provided in Table 6, we can notice the advantage of our model, as accepting its advice (i.e., using a preamble sampling instead of a Scheduled, or a Common Active Period, protocol) would result in significantly smaller delay, and a reasonable energy consumption.

7. Related Work

To the best of our knowledge, this is the first paper providing a generalizable model that can aid in the selection of the MAC protocol that best fits the application requirements of a particular WSN. In this section we will provide a short discussion of previous work that we feel is most related to our work and to work that can provide additional insight to the need of a common approach for the selection and tuning of MAC protocols. Thus, this section is not an exhaustive taxonomy on all WSN MAC protocols or models but is aimed at discussing a few works addressing the joint modeling of MAC protocols and model based tuning of MAC protocols.

Probably the closest work related to this paper was presented in [42]. In [42] the authors analyze the performance of low data-rate WSNs. While they focus on low data rate applications, in this work we aim to produce a more general model that is applicable to not only traditional low data-rate WSNs, but also to WSNs required to carry higher traffic (e.g., multimedia WSNs). As far as low data-rate WSNs are concerned, our results validate the results presented in [42].

The authors of [43] provide a quantitative analysis of IEEE 802.15.4 [43] with regards to delay, reliability, and throughput. The work in [44] in turn analyzes the energy consumption of a set of preamble sampling protocols. There seems to be general consensus that using rules of thumb when selecting and tuning (adapting parameters) a MAC protocol for a particular WSN application may not provide sufficiently good results. The authors of [45] argue along the same line when presenting a meta-approach: pTunes, where the base station selects the better protocol from among X-MAC [33] and Koala [34]) based on network feedback.

There seems to be an abundance of work on autonomic networking and learning-based MAC protocols, here we will briefly discuss only three of them. The work presented in [46] uses reservation based TDMA for high-traffic regions (i.e., nodes that are close to the sink), and contention based CSMA for other regions. Having a central role, the sink decides on the depth of the high-traffic region with the help of Beacon message broadcasts. [47] presents a meta MAC protocol that can switch between three modes (preamble sampling, common active period, and full-on) on-demand; it provides an API for switching the mode, without offering any methods or rules on when to change protocols. Adapting various parameters of MAC protocols is also important when selecting MAC protocols. For example, [48] focuses on common active period protocols, and provides a distributed method for controlling node duty cycles (sleep time) based on local information (queue length) obtained at nodes.

8. Conclusions and Future Work

Wireless sensor networks are generally used to sense the broadly defined environment and relay/store such sensed information for processing. This means that WSN application scenarios can be vastly different as the environments and data collection/relaying requirements can be extremely diverse. The designers of a WSN need to spend a considerable amount of time to decide which MAC protocol(s) to employ as MAC protocols play a crucial role in WSNs and can have crippling or strengthening effects on network performance and efficiency. In general, the MAC protocol to be used is selected by the engineers intuition and rules of thumbs, depending on the WSN requirements and scenarios. We argue that such rules of thumb are not sufficient to arrive at the best applicable MAC protocol and parameters to be used for this MAC protocol. We acknowledge that having precise models for all proposed MAC protocols for WSNs would be a daunting task. Our goal in this work was to provide a decision-making tool that can help designers in selecting the best MAC protocol and protocol parameters based on some categorization of MAC protocols. Thus, we derived a general model for selecting MAC protocols for WSNs. We defined the Combined Performance Function to determine the performance of the sets of behaviorally similar MAC categories under different application scenarios. The model helps select the set with the maximum CPF thus pinpointing the protocols that satisfy the application requirements. We also discussed the models expandability in terms of adding new protocols, categories, requirements, and performance criteria. Considering the energy consumption and delay as the initial performance criteria, we derived performance models for three protocol sets.

The main objective of this paper was to introduce the model itself, highlighting the notion of the CPF. Extending the model by a comprehensive set of protocols, requirements, protocols, categories, and performance criteria is among the future work. We did not consider hybrid MAC protocols such as [49, 46, 50] that combine different methods to take the advantage of their individual characteristics. We are currently studying these kinds of protocols and how they can be added to our framework. Section 4 provided an approximate delay analysis, that ignored the queuing delay at nodes, thus making that part of the analysis imprecise for high-traffic environments; our preliminary investigations into this sub-area suggest a straight forward addition that we hope to report on later. Additional future work includes i) real world experiments to further validate our results; ii) traffic models other than Poisson point process based ones, that represent real-life application scenarios more precisely.

References

[1] S. Li, J. Cui, Z. Li, Wireless sensor network for precise agriculture monitoring, in: Intelligent Computation Technology and Automation (ICICTA), IEEE, 2011.

[2] S. Poojary, M. M. Pai, Multipath data transfer in wireless multimedia sensor network, in: Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on, IEEE, 2010, pp. 379–383.

[3] C.-M. Chao, Y.-Z. Wang, M.-W. Lu, Multiple-rendezvous multichannel mac protocol design for underwater sensor networks, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 43 (1) (2013) 128–138.

[4] Q. Wu, X. Litrico, A. M. Bayen, Data reconciliation of an open channel flow network using modal decomposition, Advances in Water Resources 32 (2) (2009) 193–204.

[5] B. Krishnamachari, Networking wireless sensors, Cambridge University Press, 2005.

[6] I. Demirkol, C. Ersoy, F. Alagoz, Mac protocols for wireless sensor networks: a survey, Communications Magazine, IEEE 44 (4) (2006) 115–121.

[7] K. Kredo II, P. Mohapatra, Medium access control in wireless sensor networks, Computer Networks 51 (4) (2007) 961–994.

[8] A. Bachir, M. Dohler, T. Watteyne, K. K. Leung, Mac essentials for wireless sensor networks, Communications Surveys & Tutorials, IEEE 12 (2) (2010) 222–248.

[9] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, N. Xi, The evolution of mac protocols in wireless sensor networks: A survey, Communications Surveys & Tutorials, IEEE 15 (1) (2013) 101–120.

[10] A. Asudeh, An online tool for cpf calculation.
URL `http://crystal.uta.edu/~zaruba/MAC/GemMACs.htm`

[11] G. Lu, B. Krishnamachari, C. S. Raghavendra, An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks, in: Parallel and Distributed Processing Symposium, IEEE, 2004, p. 224.

[12] T. Van Dam, K. Langendoen, An adaptive energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, 2003, pp. 171–180.

[13] W. Ye, J. Heidemann, 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, Vol. 3, IEEE, 2002.

[14] K. Pister, L. Doherty, Tsmp: Time synchronized mesh protocol, IASTED Distributed Sensor Networks (2008) 391–398.

[15] K. Arisha, M. Youssef, M. Younis, Energy-aware tdma-based mac for sensor networks, in: System-level power optimization for wireless multimedia communication, Springer, 2002, pp. 21–40.

[16] P. Suriyachai, J. Brown, U. Roedig, Time-critical data delivery in wireless sensor networks, in: Distributed Computing in Sensor Systems, Springer, 2010, pp. 216–229.

[17] K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Pottie, Protocols for self-organization of a wireless sensor network, IEEE personal communications 7 (5) (2000) 16–27.

[18] S. C. Ergen, P. Varaiya, Pedamacs: Power efficient and delay aware medium access protocol for sensor networks, Mobile Computing, IEEE Transactions on 5.

[19] B. Jang, J. B. Lim, M. L. Sichitiu, An asynchronous scheduled mac protocol for wireless sensor networks, Computer Networks 57 (1) (2013) 85–98.

[20] J. Ansari, J. Riihijarvi, P. Mahonen, J. Haapola, Implementation and performance evaluation of nanomac: a low-power mac solution for high density wireless sensor networks, International Journal of Sensor Networks 2 (5) (2007) 341–349.

[21] S.-H. Yang, H.-W. Tseng, E.-K. Wu, G.-H. Chen, Utilization based duty cycle tuning mac protocol for wireless sensor networks, in: Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE, Vol. 6, IEEE, 2005, pp. 5–pp.

[22] H. Pham, S. Jha, An adaptive mobility-aware mac protocol for sensor networks (ms-mac), in: Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on, IEEE, 2004, pp. 558–560.

[23] N. Vasanthi, S. Annadurai, Energy efficient sleep schedule for achieving minimum latency in query based sensor networks, in: Sensor Networks, Ubiquitous, and Trustworthy Computing, Vol. 2, IEEE, 2006, pp. 214–219.

[24] M. S. Hefeida, T. Canli, A. Khokhar, Cl-mac: A cross-layer mac protocol for heterogeneous wireless sensor networks, Ad Hoc Networks 11 (1) (2013) 213–225.

[25] A. El-Hoiydi, Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks, in: Communications, 2002. ICC 2002. IEEE International Conference on, Vol. 5, IEEE, 2002, pp. 3418–3423.

[26] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM, 2004, pp. 95–107.

[27] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Optimizing sensor networks in the energy-latency-density design space, Mobile Computing, IEEE Transactions on 1.

[28] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, P. Pinto, A wireless sensor mac protocol for bursty data traffic, in: Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE, 2007.

[29] W.-H. Liao, H.-H. Wang, An asynchronous mac protocol for wireless sensor networks, Journal of Network and Computer Applications 31 (4) (2008) 807–820.

[30] E.-Y. Lin, J. M. Rabaey, A. Wolisz, Power-efficient rendez-vous schemes for dense wireless sensor networks, in: Communications, 2004 IEEE International Conference on, Vol. 7, IEEE, 2004, pp. 3769–3776.

[31] A. El-Hoiydi, J.-D. Decotignie, C. Enz, E. Le Roux, Wisemac, an ultra low power mac protocol for the wisenet wireless sensor network, in: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, 2003.

[32] Y. Sun, O. Gurewitz, D. B. Johnson, Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks, in: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM, 2008.

[33] M. Buettner, G. V. Yee, E. Anderson, R. Han, X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th international conference on Embedded networked sensor systems, ACM, 2006, pp. 307–320.

[34] C.-J. M. Liang, A. Terzis, et al., Koala: Ultra-low power data retrieval in wireless sensor networks, in: Proceedings of the 7th international conference on Information processing in sensor networks, IEEE Computer Society, 2008, pp. 421–432.

[35] H. Karvonen, C. Pomalaza-Ráez, M. Hämäläinen, A cross-layer optimization approach for lower layers of the protocol stack in sensor networks, ACM Trans. Sen. Netw. 11 (1) (2014) 16:1–16:30. doi:10.1145/2590810.

[36] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, A. Terzis, A-mac: A versatile and efficient receiver-initiated link layer for low-power wireless, ACM Transactions on Sensor Networks (TOSN) 8 (4) (2012) 30.

[37] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd annual Hawaii international conference on, IEEE, 2000.

[38] T. S. Rappaport, et al., Wireless communications: principles and practice, Vol. 2, prentice hall PTR New Jersey, 1996.

[39] H. Luo, H. Tao, H. Ma, S. K. Das, Data fusion with desired reliability in wireless sensor networks, Parallel and Distributed Systems, IEEE Transactions on 22.

[40] S. Ganeriwal, R. Kumar, M. B. Srivastava, Timing-sync protocol for sensor networks, in: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, 2003, pp. 138–149.

[41] O. Tickoo, B. Sikdar, Queueing analysis and delay mitigation in ieee 802.11 random access mac based wireless networks, in: INFOCOM 2004.

[42] K. Langendoen, A. Meier, Analyzing mac protocols for low data-rate applications, ACM Transactions on Sensor Networks (TOSN) 7 (2) (2010) 19.

[43] P. Park, Modeling, analysis and design of wireless sensor network protocols, Ph.D. thesis, KTH Royal Institute of Technology (2011).

[44] G. Corbellini, Adaptive medium access control for heterogeneous wireless sensor networks, Ph.D. thesis, Institut National Polytechnique de Grenoble-INPG (2012).

[45] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, L. Thiele, ptunes: Runtime parameter adaptation for low-power mac protocols, in: Proceedings of the 11th international conference on Information Processing in Sensor Networks, ACM, 2012, pp. 173–184.

[46] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, F. Cuomo, Funneling-mac: a localized, sink-oriented mac for boosting fidelity in sensor networks, in: Embedded networked sensor systems, ACM, 2006, pp. 293–306.

[47] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, P. Pinto, A wireless sensor mac protocol for bursty data traffic, in: Personal, Indoor and Mobile Radio Communications (PIMRC), IEEE, 2007.

[48] H. Byun, J. Yu, Adaptive duty cycle control with queue management in wireless sensor networks, Mobile Computing, IEEE Transactions on 12 (6) (2013) 1214–1224.

[49] Ieee std 802.15.4-2006: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans), IEEE Std.

[50] G. P. Halkes, K. Langendoen, Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks, in: Wireless Sensor Networks, Springer, 2007.