University of California
Merced

Energy Optimization Framwork for Wireless Sensor Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical Engineering and Computer Science

by

Niloufar P. Esfahani

2016

The thesis of Niloufar P. Esfahani is approved.

Shawn Newsam

Miguel Carreira-Perpinan

Alberto E. Cerpa, Committee Chair

University of California, Merced

2016

Dedicated to my father.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to thank my advisor Alberto Cerpa for his support and guidance throughout my graduate study. I am also thankful for the support of my dissertation committee, Miguel Carreira-Perpinan and Shawn Newsam whose advice and suggestions were very much appreciated. I would also like to express my gratitude to my parents, Maryam Mardy and Saeed P Esfahani, and my brother, Parsa P Esfahani for their loving support. I am also thankful for the delightful company and help of my colleagues, Alex Beltran and Daniel Winkler. Finally, a big thanks go to my friends, Shadi Zaheri and Farhad Shah for making this work possible and my time at UC Merced all the more enjoyable.

Vita

1991	Born, Tehran, Iran.
2013	Bachelor of Engineering, Sharif University of Technology, Iran.
2013–2016	Research Assistant, Electrical Engineering and Computer Science, University of California–Merced.

Publications

Niloufar P. Esfahani, Alberto E. Cerpa, "Poster Abstract: Energy Optimization Framework in Wireless Sensor Networks", Proceedings of the Thirteen ACM Conference on Embedded Network Sensor Systems (SenSys 2015), ACM, Province of Seoul, Korea, 2015.

Varick L. Erickson, Alex Beltran, Daniel A. Winkler, **Niloufar P. Esfahani**, John R. Lusby, Alberto E. Cerpa, "TOSS: Thermal Occupancy Sensing System" and ThermoSense: Thermal Array Sensor Networks in Building Management", Proceedings of the Fifth ACM Workshop on Embedded Sensing Systems for Energy-Efficient Buildings (BuildSys 2013), pp. 35:1–35:2, ACM,Rome, Italy,

2013.

Varick L. Erickson, Alex Beltran, Daniel A. Winkler, **Niloufar P. Esfahani**, John R. Lusby, Alberto E. Cerpa, "ThermoSense: Thermal Array Sensor Networks in Building Management", Proceedings of the Eleventh ACM Conference on Embedded Network Sensor Systems (SenSys 2013), pp. 87:1–87:2, ACM, Rome, Province of Rome, Italy, November, 2013.

Abstract of the Thesis

Energy Optimization Framwork for Wireless Sensor Networks

by

Niloufar P. Esfahani

Master of Science in Electrical Engineering and Computer Science

University of California, Merced, 2016

Professor Alberto E. Cerpa, Chair

We present a holistic framework for energy management in sensor networks. Our framework is based on a model-driven approach that attempts to (a) establish functional relationships across different modules of the software stack and the interrelated parameters based on empirical data, (b) use the maximum sensor value and time-synchronization errors acceptable by the users of the sensor network application as input to establish minimum quality of service requirements, and (c) do cross-layer optimization of the parameter values of all the software modules within the node's application stack to minimize total energy consumption for each sensor node. We explore the trade-offs of the design space by using a non-trivial application that includes sensing, time synchronization and routing modules, in multiple different testbeds, and with different routing software modules. We show that when using our framework, we can provide average energy savings from 38% to 62% when compared with the software modules default values, and from 11% to 33% when compared with the state-of-the-art AODC duty-cycle optimization scheme while still maintaining quality of service both in terms of the expected sensing and time-synchronization errors. We further show

that we can slightly decrease latency (1.5%-3.5%) and slightly improve reliability (1%-3%).

CHAPTER 1

Introduction

Energy management and efficiency has been a well studied problem in the wireless sensor network (WSN) literature, since in many applications, wireless nodes are untethered from the energy infrastructure and are battery operated [YHE02]. Since the cost of replacing batteries is expensive or very difficult to achieve, extending system lifetime has been paramount.

There have been many efforts geared towards improving energy efficiency of different WSN system software modules, including data sampling [DM06,KPS05, JC04,AAG07], data storage and query processing [LSK10,FZS08,MDc13,CPR03, MFH03], routing [GFJ13,LGD12,SZH04,SS01,DLV13,SR02,MA00,MA01,GGS01] and time synchronization [MKS04, GKS03a, ZCH11].

All these software modules are designed to do different works. They share the same hardware resources and as they are not coordinated, they will consume more energy. We can optimize the energy if we find a way to make it coordinated. Radio and CPU are some of the hardware resources that are shared with all the modules. As radio communication has high energy consumption, we need to optimize the radio consumption based on hardware and software constraints. Hardware constraints are maximum and minimum hardware rates that a system can provide. Software constraints are the quality of service that a user needs to provide. In our work we also consider the computational cost to be minimum while having a high accuracy.

However, these individual modules may not be flexible to adapt to changing network and application dynamics. Perhaps more importantly, each module is designed to optimize energy consumption from a *module-centric* point of view, without considering the inter-relationships among the different modules in the sensor stack. Moreover, there is a trade-off between energy use and the maximum sensing and timing errors admitted by the application's users. This trade-off has not been generally considered in past work.

We take a fresh look at this problem by developing an Energy Optimization Framework (EOF) scheme, which considers the energy trade-offs caused by the use of multiple software modules *simultaneously*, and optimizes the overall system energy consumption in a holistic way, based on the inter-relationships among software modules and the user constraints that bound the expected application quality of service.

In many applications, most of the system energy consumption comes from radio communication [PK00]. It is the use of the radio by different software modules' needs that dominates the energy consumption. Since the RF transceivers commonly used in WSN also consume energy while listening [DL03], we use the radio duty-cycling ratio as a proxy that approximates energy consumption. Our goal is to reduce this ratio while maintaining quality of service constraints based on user input. We do this by finding functional parametrized relationships based on empirical data among the different modules of the sensor software stack, and minimize the energy based on user constraints. We want to find the optimal parameters that allow minimization of the system energy consumption while still satisfying the user needs.

In our work, we consider an application stack consisting of data sampling, routing and time synchronization software components. These components are

representative of multiple data collection sensor network applications [WLR06, AAG07, JC04] and provide a representative sample of the design space to show the benefit of our EOF scheme. We evaluate our architecture using two different routing components, in two different testbed environments. We develop our EOF scheme in the following manner. First, we analyze how different parameters that control the software modules behavior, like data sampling frequency and error threshold for data sampling, beacon frequency and sleep interval for routing/MAC, and time synch frequency for time synchronization affect the duty-cycling ratio. Then, we establish functional relationships that approximate this behavior using data-driven approaches. Finally, using the functional relationships derived above, we optimize the parameter values in order to minimize energy consumption while satisfying the user constraints.

We would like to highlight the contributions of our work:

a) We designed and developed the Energy Optimization Framework (EOF) for wireless sensor network applications, which considers the energy trade-offs caused by the use of multiple software modules *simultaneously*, and optimizes the overall system energy consumption in a holistic way, based on the inter-relationships among software modules and the user constraints that bound the expected application quality of service.

b) We derived the functional relationships for sensing, routing and time-synchronization software modules parameters with respect to the radio duty-cycling in two different environments and with two different routing/MAC schemes. This includes the closed-form formulae, as well as the analysis of the amount of data required in order to derive a function with low error.

c) We extensively tested EOF and compared them with the default values used in the different software module as well as the state-of-the-art AODC framework.

Our results show average energy savings from 38% to 62% when compared to default parameter values, and from 11% to 33% when compared with AODC for a wide range of data and time synchronization errors accepted by the user. We also see slight (1%-3%) increases in data delivery reliability and lower latency (1.5%-3.5%) when compared to both default values and AODC.

CHAPTER 2

Background and Related Work

In WSNs, improving energy efficiency of different system software modules has been a relevant research topic for multiple different components, including data sampling [DM06,KPS05,JC04,AAG07], data storage and query processing [LSK10, FZS08, MDc13, CPR03, MFH03], routing [GFJ13, LGD12, SZH04, SS01, DLV13, SR02, MA00, MA01, GGS01] and time synchronization [MKS04, GKS03a, ZCH11]. The majority of these efforts have not considered a cross-layer optimization of multiple module parameters simultaneously, in order to holistically reduce the overall energy consumption. In our work, we leverage some of these works, like a MauveDB [DM06] data sampling sensing module, FTSP [MKS04] time-synchronization module, and CTP [GFJ13] and ORW [LGD12] routing modules, but we combine them together using EOF in order to optimize the different module parameters to minimize energy consumption while satisfying user data error and timing requirements.

In the remainder of this chapter, we will review some literature related to different components of our system: MAC-layer, data prediction, routing, and time synchronization techniques. There are some works for each component that are energy efficient and have cross-layer techniques similar to our work.

2.1 MAC-layer techniques

2.1.1 Non adaptive duty cycling ratio

Polastre et al. [PHC04] presented B-MAC, a MAC-layer protocol that outperforms LPL [SHL13] MAC protocols in terms of throughput, latency, and often energy consumption. B-MAC extends LPL techniques by having a user-controlled sleep interval. Each node periodically goes to sleep, with a predefined constant sleep interval. When a transmitter sends a data packet, it sends a preamble long enough to cover one sleep interval. X-MAC [MLT08] is another MAC-layer protocol which outperforms B-MAC. Instead of sending a long enough preamble, the transmitter sends a special control packet called a strobe to the receiver. When the receiver receives the right packet strobe, it sends an ack to the transmitter. Once the transmitter receives the ack message, it sends the data message to the receiver. X-MAC reduces energy consumption by using the ack mechanism instead of sending a long wake up preamble compared to B-MAC. X-MAC finds an appropriate sleep interval which reduces the energy consumption. However, it does not consider the reliability and delay constraints of the system.

2.1.2 Adaptive duty cycling ratio and no energy minimization

Jurdak et al. [JRO10] present an adaptive duty cycling technique of a preamble sampling protocol. The authors compute the energy consumption of each node. As a routing decision, each node sends the data packet to it preferred node. The preferred node is chosen based on the computed energy consumption. Energy consumptions are calculated based on the radio duty cycling ratio and the ratio is proportional to the number of transmitted packets. The authors do not guarantee to minimize the energy consumption of the whole system nor consider meeting

the delay and reliability constraints.

2.1.3 Energy efficiency with no reliability and delay constraints

Park et al. [PPL09] present an effective framework for optimizing the energy consumption of WSNs in which the nodes use an asynchronous wakeup schedule. The authors present two convex optimization problems to minimize the energy consumption of the network and maximize the network lifetime constraint to their design objectives. They use a broadcast algorithm which allows each node to have a different wake up interval. Similar to our work, they present an architecture to minimize the energy consumption of the system considering the MAC-layer parameter. In addition to the MAC-layer of the network, our objective function also contains the relationship between the components and the energy consumption of the system. The paper does not consider the reliability and delay of the system. However, we have declared different time and data error constraints and we have shown that we can meet the delay and reliability requirements. Unlike our work, some of the results of this paper are based on simulation and not real experiments.

Other papers have been written on reducing the number of sent packets and energy consumption of the system. They focused more on minimizing the energy consumption and packet delivering ratio of the system. The protocols find optimal MAC-layer parameters for energy consumption minimization. However, there is no adaptive duty cycled protocol that helps the system meet some constraints like reliability and delay constraints at the same time. In [LP08], the authors presented an architecture to select a routing path from every node to the sink for the duty-cycled WSNs. Similar to our work, the selection of the routing path is based on an optimization problem. In this case, the optimization problem is

to minimize an objective function which shows the energy consumption of the system based on latency constraints which should not exceed a threshold. The optimization tries to find some optimal sleeping schedule parameters. The results of this paper are gained based on different simulations and no experiments have been done. However, all of the results of our work are based on experiments on large enough testbeds. Similarly, in our work we optimize the energy consumption of the system. Our objective function formulation is not only based on sleeping schedule parameters (MAC-layer parameters), but also considers sensing and routing parameters.

In [MH10], the authors presented two techniques to reduce the number of delivered packets at one side and the energy consumption of the system at the other side. The first technique changes the sleep interval for each node according to the packet delivering ratio of the node. If a node successfully sends more than 5 consecutive number of packets, it's sleep interval will be decreased. Otherwise, it will increase the sleep time. In this paper authors use LPL MAC protocol. The second technique tries to reduce the energy consumption by considering the reliability constraints. Our work performs better than this paper in many ways. First of all, all of their results are based on simulations. In the paper, the authors have not focused on optimizing the system in an holistic manner considering different constraints. Their results do not guarantee to reduce the energy consumption and meet the delay and reliability constraints at the same time.

2.1.4 Energy efficiency considering reliability and delay constraints

Other works have investigated performing MAC-layer optimization to achieve higher energy efficiency subject to reliability and delay application constraints.

ZeroCal [MWZ10] presented the first scheme to perform MAC-layer optimization to reduced energy consumption based on application reliability constraints. The scheme used X-MAC [MLT08] and optimized the Sleep Interval (SI) based on the reliability constraints. In pTunes [ZFM12], the authors propose a framework to dynamically adapt the MAC level parameters based on routing topology, link dynamics and traffic load. They set up a specific objective function for their LPP [MLT08] MAC parameters (i.e. the number of successful/failed unicast TX (transmit) before success, the time for success/failed unicast transmissions, and the fraction of time in TX and RX (receive) mode), in order to reduce energy consumption subject to reliability and latency constraints. In [NC10], the authors presented two dynamic sleep time control approaches to reduce the control packet energy waste incurred in LPL [SHL13] MAC-layer technique. Unscheduled MACs using LPL include B-MAC [PHC04], WiseMAC [ED04] and X-MAC [BYA06a], and so on. Both approaches dynamically compute the sleep time based on some network constraints. The first approach finds the optimal sleep time based on the delay constraints and second approach finds the optimal sleep time in order to minimize the energy consumption. Similarly, a MAC-layer parameter optimization framework is proposed by [PEF13]. The authors presented the AODC algorithm to minimize the energy consumption while guaranteeing application reliability and delay requirements, by defining an objective function that finds the optimal SI for the BoX-MAC [ML08] medium access control. Our work differs from these work in multiple ways. First, our quality of service guarantees are *data-centric* and based on maximum data sensing and time synchronization errors accepted by the users, instead of the more *network-centric* reliability and delay used previously. We show that we can slightly improve network-centric quality of service guarantees. Second, and perhaps more importantly, our objective function is truly cross-layer, integrating software modules that deal with

data sampling sensing, time-synchronization and routing/MAC to holistically minimize energy consumption while satisfying applications constraints. Finally, we establish rule of thumb bounds for the practitioners as to how much data is necessary to collect to define the functional relationships among the different parameters of the system for multiple different software modules and environments. Our work is also quantitative compared to AODC. In a recent work in practical data prediction, Raza et al. [RCM15] propose extending the system lifetime by minimizing data communication from source nodes, by using a BDP model-predictive approach at the sensing module. While the authors do not perform cross-layer optimization, they show how the performance changes as a function of the Sleep Interval (SI) parameter in the MAC layer. Our work not only shows the performance change as a function of SI, but also approximates empirically the functional relationships among all the different software modules to perform a global parameter optimization to reduce energy consumption while satisfying quality of service constraints.

2.2 Data Prediction

There have been lots of research in the data prediction area. Data prediction techniques are used to reduce the transmitted data in the network. They use different models to predict the data; constant, linear, non-linear, and correlation models. Raza et al. [RCM12], Gaura et al. [GBA13], Palpanas et al. [PVK08], and tulone et al. [TM06b] (PAQ) present some linear models for data prediction. Raza et al. use BDP model and help the system reduce the energy consumption. Gaura et al. has presented edge mining, which transforms the sensed data to a sparse form in order to minimize the packet transmissions, energy use, and storage space. Palpanas et al. present a method which does the on-line approximation

of streaming time series. PAQ shows that autoregressive (AR) models which are linear models reduce the amount of data needed to be sent in an application. SAF [TM06a] proposes an energy-efficient framework using the non-linear models to reduce the amount of communication in the network. They detect data similarities by comparing the local model form of the data instead of the raw data. DKF [JCW04] proposes an idea for reducing data communication in a multiple data streams environment by using Dual Kalman Filter. Huang et al. [HJA13] compares the techniques from each model category. The authors have concluded that linear and constant models will have better results than non-linear models if the sensed data has a small variation.

The idea of MauveDB [DM06] is to change the real-world data of a network to a model-based view, which will be a usable data and can be stored in database, since the real-world data tends to be incomplete, imprecise, and erroneous. As our sensing module is inspired by this paper, we have named our first sensing application, MauveDB. In our sensing application, we have changed the temperature and time data to a time vs. temperature model. MauveDB has used linear regression methods to find the models like our sensing module.

Similar to our work, the goal of all these papers is reducing the amount of data reported, and therefore the energy consumption of the network. However, in addition to considering data reduction which is related to the sensing application, we have also considered different components like MAC-layer, routing, and time synchronization and we have minimized an objective function which consists of several components.

2.3 Routing Protocols

There are lots of applications which use the multi-hop network topology [PSM04] [XRC04] [WLJ06]. The routing protocol creates the network tree topology considering the wireless link quality such that the cost of the path of sending a packet from a sender node to the root is minimal.

CTP [GFJ13] presents a default tree based routing protocol implemented on TinyOS. The tree is created based on the ETX [CAB05] link quality metric. Each node updates its ETX value and estimates the ETX of neighbors with the link estimator, then the node finds the best path with the lowest ETX cost in order to send the packet to the root. CTP contains three main components: link estimator which estimates the link quality of a hop to its neighbors, routing engine which is used to find the routing path, and the forwarding engine that is for sending the data packet in the network. The link estimator method used in the CTP protocol is 4bit [FGJ07]. Other protocols [BKY10] and [BZV10] also can be used as a CTP link estimator.

2.3.1 Opportunistic Routing

GeRaF [ZR03] is the first publication proposing the any-cast routing. GeRaF presents an any-cast forwarding technique based on the geographical location of the nodes and a practical scheme for selecting the relaying node to ensure a unique forwarder. ExOR [BM05] pioneered the concept of opportunistic routing. ExOR is a uni-cast routing technique for multi-hop wireless networks. The algorithm sends a packet through a sequence of nodes and chooses the best forwarding node from all the nodes that successfully received the packet. CMAC [LFS07] combines the concept of GeRaF and ExOR. Its technique includes finding prioritized for-

warders, and choose a best forwarder based on geographical routing. However, it doesn't address the key challenges for opportunistic routing in duty-cycled WSNs.

More theoretical aspects in Opportunistic Routing is proposed in [DGV11] [KLS10] [KLS11] [BC08]. ORW [LGD12] presents a practical opportunistic routing scheme for wireless sensor networks. In a duty-cycled setting, packets are addressed to sets of potential receivers and are then forwarded by the neighbor that wakes up first and successfully receives the packet. All of these papers show that OR reduces the energy consumption of a network considering the delay and reliability constraints. They all show that OR reduces energy while maintaining the routing efficiency of a system.

2.4 Time Synchronization

There are two popular time synchronization methods used in WSN: TPSN [GKS03b] and RBS [EGE02]. TPSN is based on a simplistic approach for sender-receiver time synchronization, while RBS presents a receiver-receiver time synchronization method. TPSN has gained an additional performance over RBS, since TPSN adds time stamps to radio messages multiple times and averages them. TPSN is implemented on mica (a wireless platform), which makes it harder to be implemented on later platforms. FTSP [MKS04] protocol uses low communication bandwidth and is robust against node and link failure. It utilizes the network of mica2 motes. FTSP causes less network traffic and less resource allocation that TPSN and RBS. If a time synchronization period is T seconds, 1 message for FTSP, 2 messages for TPSN, and 1.5 messages for RBS will be sent in T seconds. As a conclusion, FTSP produces less traffic load compared to those two protocols. As a result, we use the FTSP time synchronization protocol in our work as one of our software modules.

CHAPTER 3

EOF System Description

Figure 3.1 shows the overall EOF system description. Our WSN software stack consists of sensing, time synchronization, routing and MAC software modules. We collect data measuring the duty cycling ratio as a function of the different software module parameter values. Once the data is collected, we proceed to calculate the functional (parametrized) relationships between the duty cycling ratio and each parameter (function variable) in the software stack.

Our objective function that represents the duty cycling ratio when all the software modules are run concurrently is derived by fitting multiple linear and non-linear models. The variables in this objective function are the software module parameters that affect the behavior of each module. In addition, the user's maximum data error and time synchronization error are input as constraints in the optimization process.

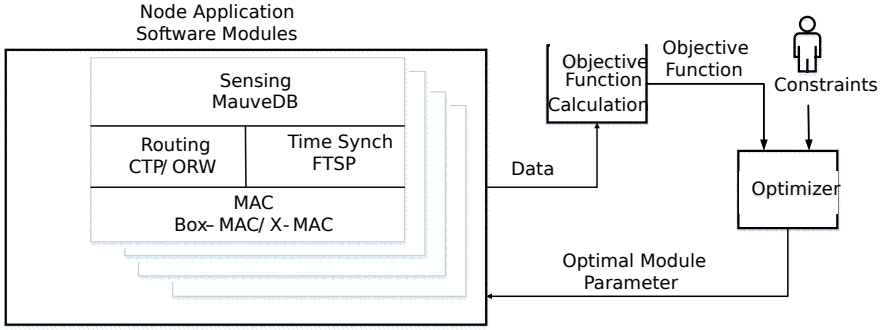


Figure 3.1: Energy Optimization Framework System Description

F_{sr}	E_t	F_b	F_d	S_i
Time synch Frequency	Error Threshold	Beacon Frequency	Data Sampling Frequency	Sleep Interval

Table 3.1: Parameter Description

We then proceed to solve the non-linear optimization problem and find the optimal software module parameter values (the variables of the optimization problem) that minimize the duty cycling ratio subject to the user constraints the bound the quality of service requirements.

In the following sections, we provide further details of each step in the process, describing the different software modules used to build a data collection application, the experimental setup to collect the data, the derivation of the different functional relationships between the duty cycling ratio and the module parameters, and the final optimization process.

3.1 Software Modules

We have used a non-trivial application consisting of three core components: time synchronization, sensing, routing and MAC modules. These components are representative of multiple data collection sensor network applications and provide a good sample of the design space. In order to evaluate our system more thoroughly, we use two different routing modules and use two different testbeds (see below). We have also used two different sensing applications: One of the sensing applications is similar to MAUVEDB [DM06]. Sensor nodes sense an environmental parameter (i.e. temperature in our application) over time, and build their own temperature model. Each node builds a simple linear regression model between temperature and time, and calculates the best fit. If the difference

between a value sensed by the node and the model prediction is larger than a specific threshold defined by the user, then the application forwards the out-lier data sample towards the sink, and the node recalculates the temperature model. While a sensor node could send every sample value for maximum accuracy, there is a trade-off between the number of transmissions required (and their energy usage), and the accuracy that can be achieved. By optimizing only for the required user's accuracy needed, we can minimize the overall energy consumption.

Another sensing application (POR(Polynomial Regression)) is same as the first one but the only difference is the time vs. temperature curve fitting model. We use second order polynomial curve fitting model which is a nonlinear regression model for the second sensing application.

For time synchronization we use FTSP [MKS04]. This is in general useful when the users of the application need to understand the spatio-temporal dynamics of the phenomena being sensed. For routing, we used two commonly used routing modules, CTP [GFJ13] and another one uses ORW [LGD12]. Finally, we used the BoX-MAC [ML08] module for medium access control when using CTP, and the X-MAC [BYA06b] when using ORW.

3.2 Experimental Setup

We run experiments in two different testbed. The first testbed (Local) is an indoor testbed in a building with 60 Tmote Sky motes placed along an corridor of a typical office building. The motes are divided into 20 groups. The distance between each node group ranges from 6 to 7 meters, except for the node group in the far left which sits around a corner at the end the corridor. The root node is denoted as sink.

F_{sr}	E_t	maxF	F_b (1024/maxF)	$F_d * 10^3$	S_i
2	2	40000	0.0256	0.4	0.05
1	3	100000	0.0102	0.8	0.5
0.5	4	150000	0.0068	2.4	1
0.3	5	512000	0.002	2.8	2
0.25	10	550000	0.0018	3.3	
0.2	20	600000	0.0017	4.2	
0.1	30	700000	0.0014	5.5	
0.05	40	900000	0.0011	8.3	
0.03	50			12	
0.025	60			16.7	
0.1	80				
	100				

Table 3.2: Parameter values tested

The second Indriya testbed is a three-dimensional wireless sensor network deployed across three floors of the School of Computing, at the National University of Singapore. The Testbed comprises of 139 TelosB sensor "motes", each of which built of TI-MSP430 microcontroller with 10KB of RAM, internal and external flash memories of size 48KB and 1 MB respectively, and a Chipcon CC2420 radio operating at 2.4GHz with an indoor range of approximately 20 to 30 meters.

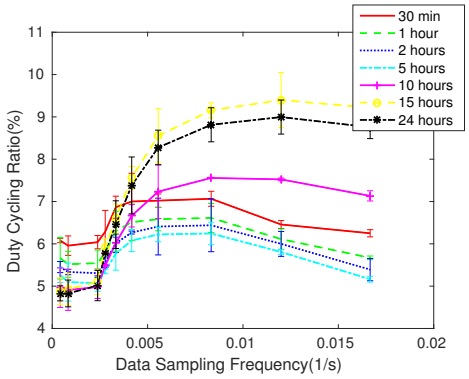
We run extensive experiments in each of the above testbeds and using different routing/MAC layer software modules in order to understand the functional relationships between the radio duty cycling ratio and the different parameters of the all the software modules in our application stack. Table 3.1 shows all the software module parameters that can affect energy consumption. Table 3.2 shows all the values for different parameters that are tested in the system in order to create the formulations. For each parameter value tested (45 for CTP and 38 for ORW), we run 24 hours experiments in the Local testbed for each CTP and ORW, and 24 hours experiments for CTP in the Indriya testbed. We experimented for a total of 215 days (many experiments were carried out in parallel on

both testbeds), and sent more than 130K packets.

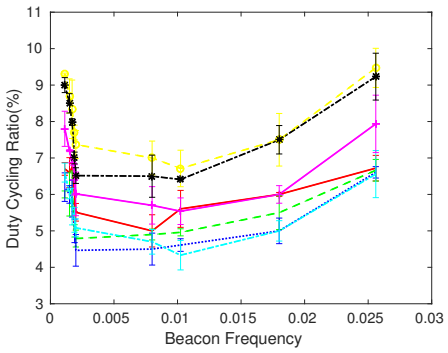
3.3 Model Learning

We want to be able to understand the relationship between the radio duty cycling ratio (a proxy for energy consumption), and the different software module parameter values. The goal is to derive from empirical data, parametric approximate functional relationships that are suitable to use in an optimization process. A secondary goal is to determine the minimum amount of data required in order to get good approximations that are useful in practice.

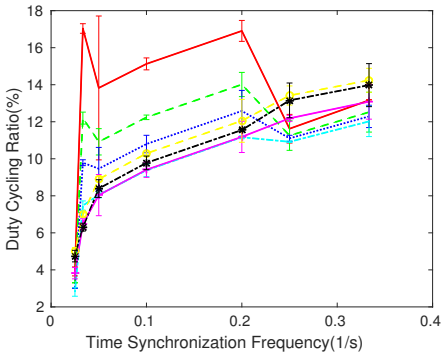
To achieve this goal, we collected extensive experimental data as explained above. Figures 3.2, 3.4, 3.5, 3.6, and 3.7 show the results of these experiments. These results are used as the training data to derive functional relationships between the radio duty cycling ratio and the software module parameters. We applied different linear and non-linear regression models with different polynomial degrees on the data in order to find an adequate fit. We compared all the results and, not surprisingly, we noticed that the non-linear models have smaller RMSE. The model finally selected had a smallest RMSE of 0.05. Among all the non-linear models, we tried to use lower degree polynomials. The reasons are three-fold: (a) Lower degree polynomials in the functional model make the optimization process less complex and faster, (b) while having higher degree polynomials may provide a better fit for the training data, they may not generalize well with testing data, and (c) even if in some cases higher polynomials perform better for also testing data, the optimal values will not change as we make the model more complex, as the goal is finding the accurate optimal values and having faster optimization model in our framework. Ultimately, this is a design decision between the model error and complexity due to the bias-variance trade-off.



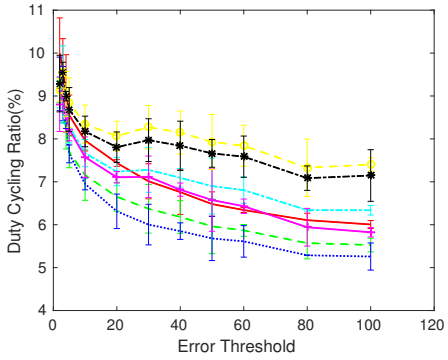
(a)



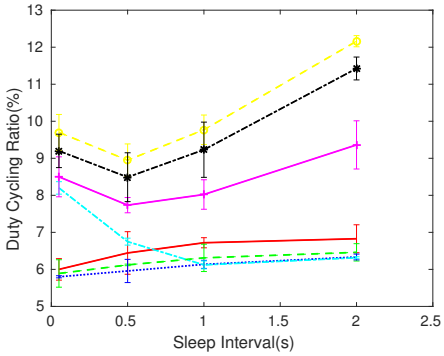
(b)



(c)



(d)



(e)

Figure 3.2: Radio duty cycling ratio vs. parameters (MauveDB – CTP/BoX-MAC – Local Testbed) repeated for three times.

Figure 3.3 compares the actual relationship of different parameters and r, with the predicted model for one case of experiments, while the experiment length is 15 hours. From the figure we can see the great performance of our predicted model.

We derive the generic parametrized functional relationship between the radio duty cycling ratio (energy consumption) and the different module parameters for CTP on both Local and Indriya testbeds:

$$r(E_t, F_d, F_b, F_{sr}, S_i) = r_1(E_t) + r_2(F_d) + r_3(F_b) + r_4(F_{sr}) + r_5(S_i)$$

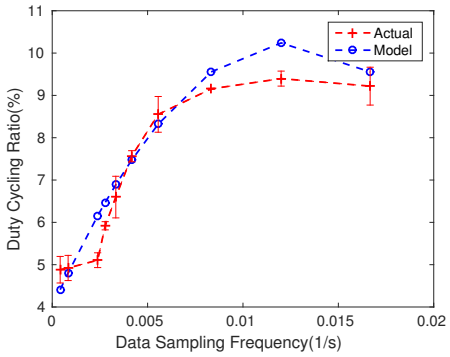
$$\begin{aligned} r = & \frac{a}{E_t} - bF_d^2 + cF_d + dF_b^2 \\ & - eF_b + fF_{sr} + gS_i^2 - hS_i + i \\ & a, b, c, d, e, f, g, h > 0 \end{aligned}$$

Similarly, we do the same for ORW for the Local testbed. Please note that we cannot control the beacon frequency in this case.

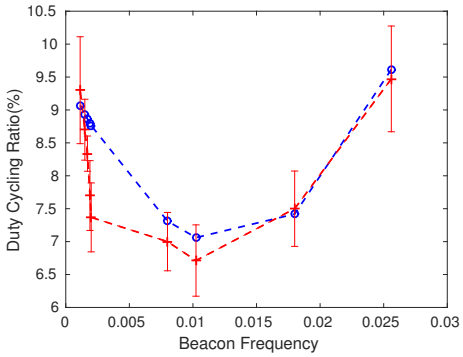
$$r(E_t, F_d, F_{sr}, S_i) = r_1(E_t) + r_2(F_d) + r_3(F_{sr}) + r_4(S_i)$$

$$\begin{aligned} r = & \frac{a}{E_t} - bF_d^2 + cF_d \\ & + dF_s r + eS_i^2 - fS_i + g \\ & a, b, c, d, e, f > 0 \end{aligned}$$

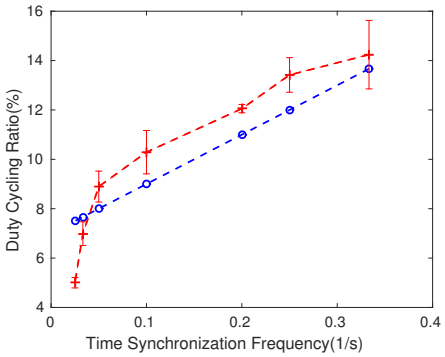
Since we would like to understand the model dynamics derived as a function of the amount of data collected (experiment time length), Figures 3.2, 3.4, 3.5, 3.6, and 3.7 show the different relationships established as a function of the



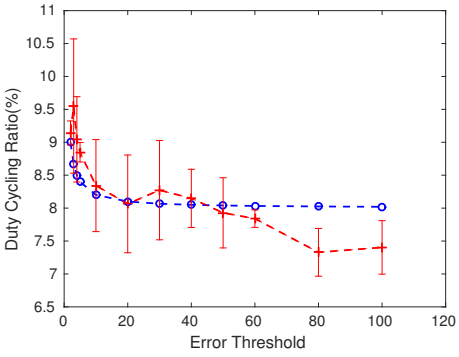
(a)



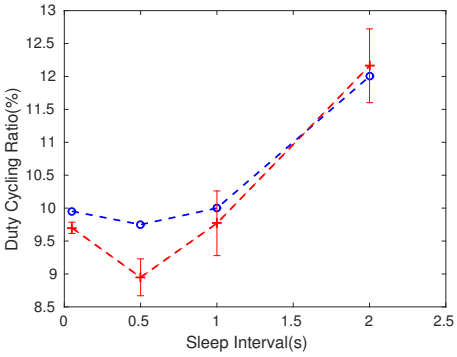
(b)



(c)



(d)



(e)

Figure 3.3: Comparing actual relationship and predicted model (MauveDB – CTP/BoX-MAC – Local Testbed) for 15 hours time experiments.

time (h)	a	b	c	d	e	f	g	h	i
0.5	0.015	490	9.2	152	3	0.18	0.006	0.008	-0.002
1	0.02	400	7.4	120	2.34	0.2	0.005	0.007	-0.0002
2	0.04	340	6.47	136	2.94	0.2	0.004	0.006	0.004
5	0.04	323	5.96	146	3.45	0.2	0.005	0.007	0.02
10	0.02	352	7.85	144	3.46	0.2	0.005	0.007	0.006
15	0.009	400	10	158	4	0.2	0.01	0.01	-0.0007
20	0.009	381	9.3	156	3.8	0.2	0.01	0.01	-0.0008

Table 3.3: CTP: Objective function coefficient results

time (h)	a	b	c	d	e	f	g
5	0.06	500	9.4	0.2	0.06	0.03	-0.03
10	0.017	6.7	1.14	0.3	0.008	0.01	0.0002
15	0.02	120	4.6	0.3	0.01	0.02	-0.006
20	0.04	308	8	0.3	0.01	0.02	-0.02

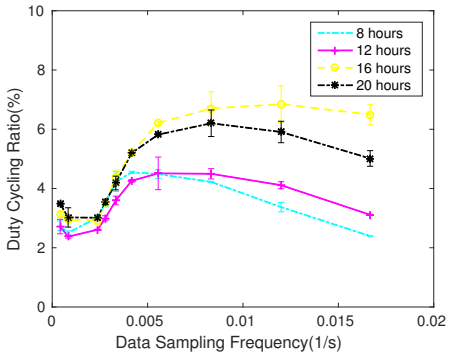
Table 3.4: ORW: Objective function coefficient results

experiment time length. We modify the time length from 30 minutes to 24 hours. Each of the figures above is obtained by modifying the module parameter being tested while leaving all other module parameters with a constant value. Tables 3.3 and 3.4 show the specific coefficients for formulas optimized for each specific data collection time length.

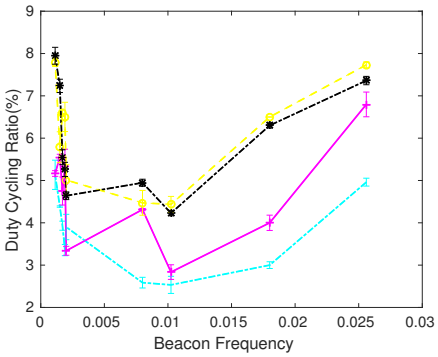
We now proceed to provide insights as to each parameter component that form the functional model derived above.

Data sampling frequency (F_d):

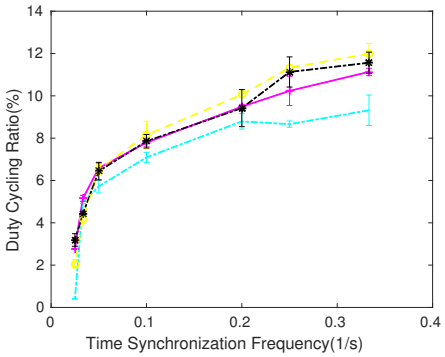
The sensing application can sense the environment at different data sampling frequencies. A sensor can improve the accuracy by sampling at higher frequencies, at the cost of more energy consumption. In our case, for the temperature sensing application used, since the temperature does not fluctuate too much in a few seconds, we defined the maximum data sampling frequency to be one minute. Figures 3.2a, 3.4a, 3.5a, and 3.6a show how the radio duty cycling ratio changes



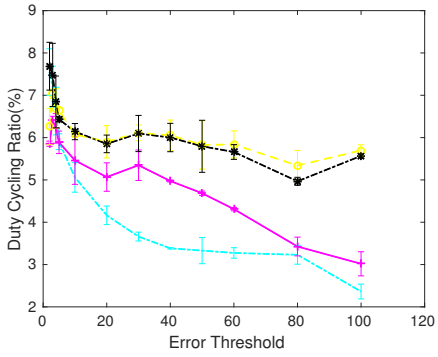
(a)



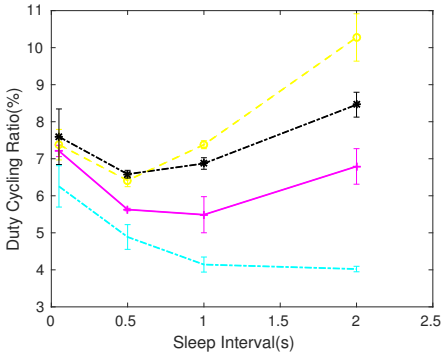
(b)



(c)



(d)



(e)

Figure 3.4: Radio duty cycling ratio vs. parameters (MauveDB – CTP/BoX-MAC – Indriya Testbed) repeated for three times.

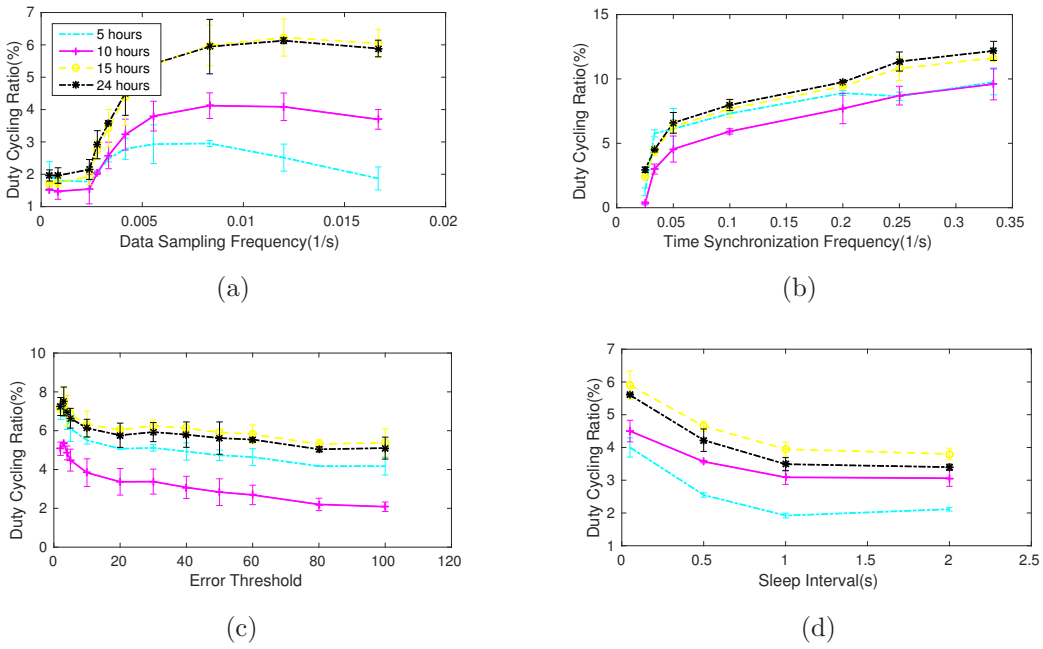


Figure 3.5: Radio duty cycling ratio vs. parameters (MauveDB – ORW/X-MAC – Local Testbed) repeated for three times.

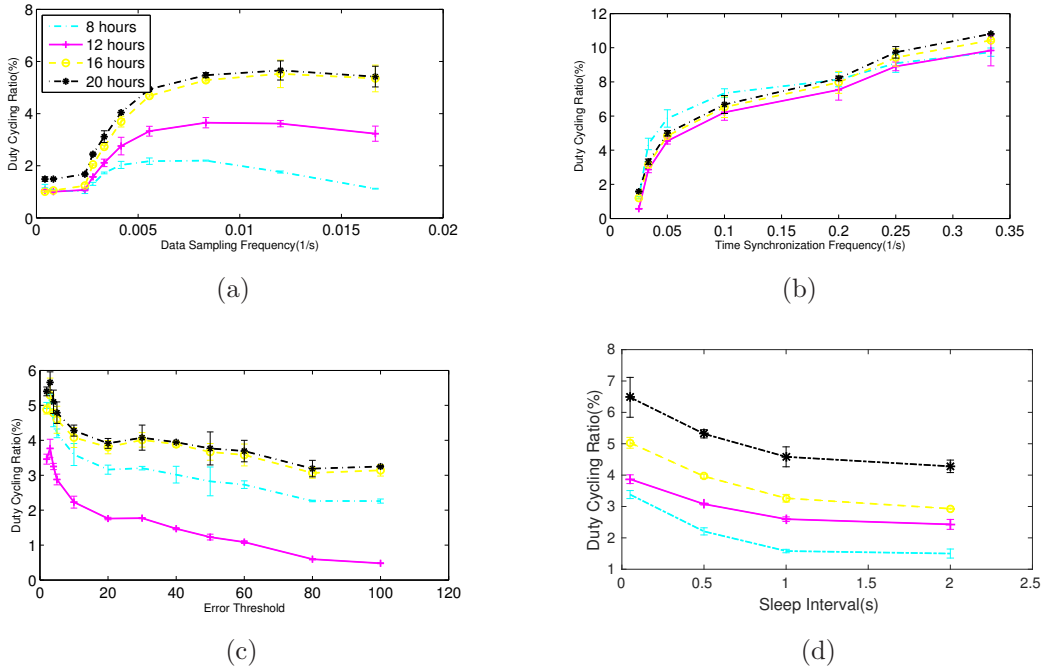
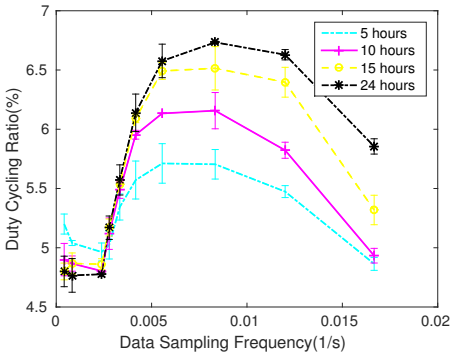
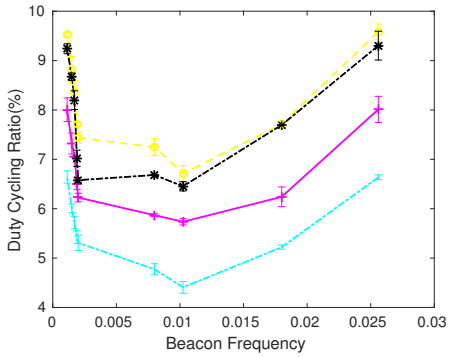


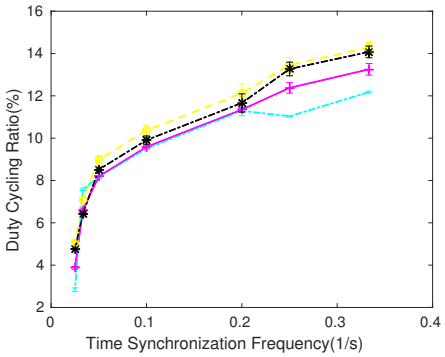
Figure 3.6: Radio duty cycling ratio vs. parameters (MauveDB – ORW/X-MAC – Indriya Testbed) repeated for three times.



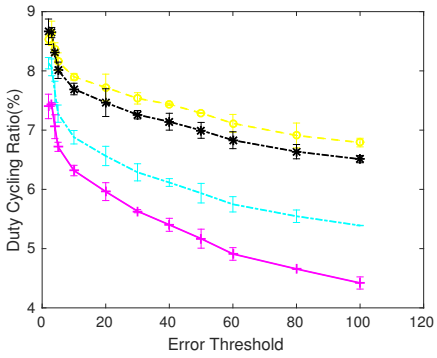
(a)



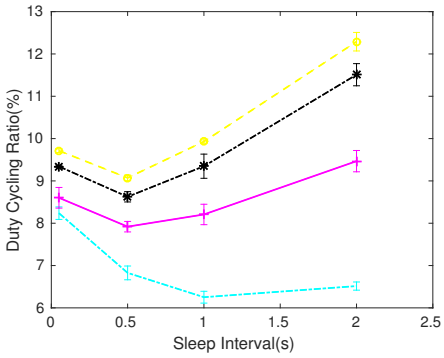
(b)



(c)



(d)



(e)

Figure 3.7: Radio duty cycling ratio vs. parameters (POR – CTP/BoX-MAC – Local Testbed) repeated for three times.

as F_d changes by using the first sensing application (MauveDB). From the figures we can see that by reducing the sampling frequency towards zero, the energy consumption for sensing also gets reduced, since there is less data to send, even when the model is not accurate and the sensing error is large. By increasing F_d from close to zero, as the sensing error is relatively large due to the small sampling rate, sending more data packets results in increase energy consumption. As F_d gets larger, the sensors sense the environment more often, and hence model accuracy is improved. At some point, increasing F_d results in an increase in accuracy. By increasing the accuracy, less data packets are sent in the network and as a result the radio duty cycling ratio decreases. Figure 3.7a shows the relationship between radio duty cycling ratio and F_d for the second sensing application (POR). By using nonlinear regression model in our second sensing application, we have a better time vs. temperature model fit than using linear regression model. By having a better model, the differences between predicted temperature value and actual temperature value decreases. As a result, we have less sent data packets. Furthermore, the radio duty cycling decreases. By comparing 3.7a with 3.2a, we see that we have a lower radio duty cycling ratio by using POR. Another difference is that when F_d gets large enough the radio ratio drops faster by using POR rather than using MauveDb.

From Figure 3.2a, we see that there is a similar trend for all curves except the 30 minutes one. This shows that a 30 minutes experiment is not enough for finding the relationship between F_d and duty cycling ratio (r).

Error threshold (E_t):

In the sensing component the nodes send data packets with their new sensing model (temperature in our case) if the difference between the predicted and actual data sensor value is greater than the error threshold. Figures 3.2d, 3.4d, 3.5c,

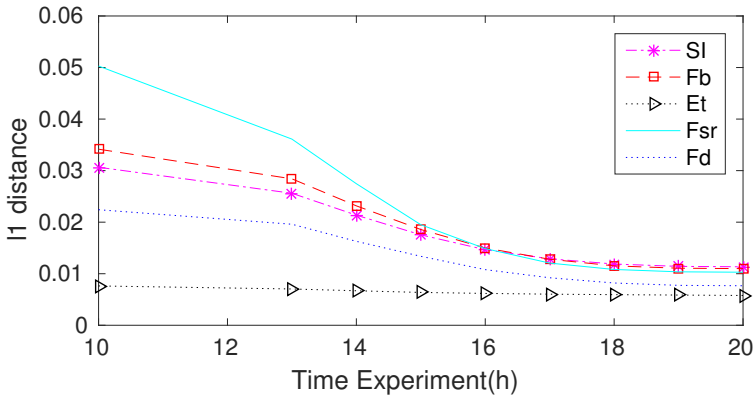


Figure 3.8: L_1 distance vs. time experiments (CTP – Local – MauveDB)

3.6c, and 3.7d show that by increasing Et, less data packets are likely to be sent to the sink so the radio will be off more often. As a result, there will be less energy consumption.

Time synchronization frequency (F_{sr}):

Figures 3.2c, 3.4c, 3.5b, 3.6b, and 3.7c show that as F_{sr} increases, more time synchronization packets are sent and the radio will turn the radio on more often, increasing energy consumption. From Figure 3.2c we see that the minimum time experiments of 10 hours is needed in order to find the relationship between F_{sr} and r.

Sleep Interval (S_i):

By putting nodes in sleep mode for longer periods, we obtain lower duty cycle. A larger sleep time reduces the cost of idle listening at the receiver. As the transmitter uses a longer preamble, the transmission cost increases. Therefore, there is a trade-off between the receiving cost and the transmission cost. Figure 3.2e, 3.4e, 3.5d, 3.6d, and 3.7e show how changing sleep intervals changes the energy consumption. Figure 3.2e shows a similar trend for time experiments of 10, 15, and 20 hours. Lower sleep intervals do not show the same trend.

When using CTP we also have another parameter:

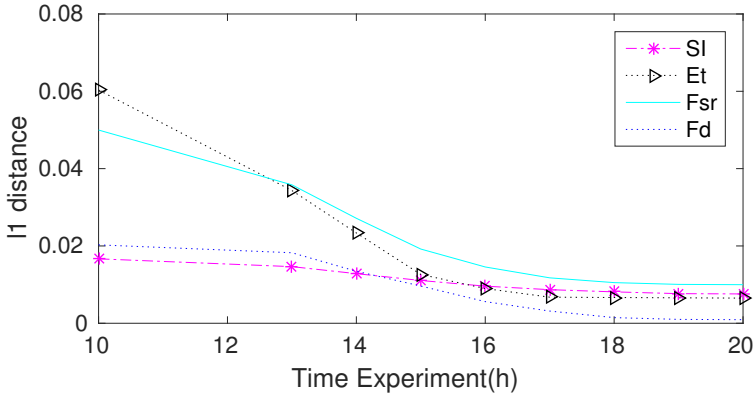


Figure 3.9: L_1 distance vs. time experiments (ORW – Local – MauveDB)

Beacon frequency (F_b):

This is frequency in which beacon packets are sent by the routing module in order to create efficient routing paths. This parameter is the maximum beacon frequency interval in the CTP routing engine implementation. In our work, we define *transmission cost* as the number of attempts for each data packet to be transmitted in the network and *routing cost* as the number of sent beacon packets in the network. Increasing F_b decreases the transmission cost and increases the routing cost. As a result, there is a trade-off in energy consumption by increasing F_b. From Figures 3.2b, 3.4b, and 3.7b we see that when F_b is close to 0, less efficient paths are being created, thus the radio will be on more in order to send the inefficient number of data packets in the network and hence the energy consumption increases. By increasing F_b, the energy consumption decreases as more efficient routing paths are created. In this case higher transmission cost is more effective than the lower routing cost. When F_b gets larger than a certain value (where more efficient paths can not be created), energy consumption increases. In this case higher routing costs dominate the energy consumption over the transmission cost. From Figure 3.2b we can see that the minimum experiment time needed for finding the functional relationship trend is 5 hours.

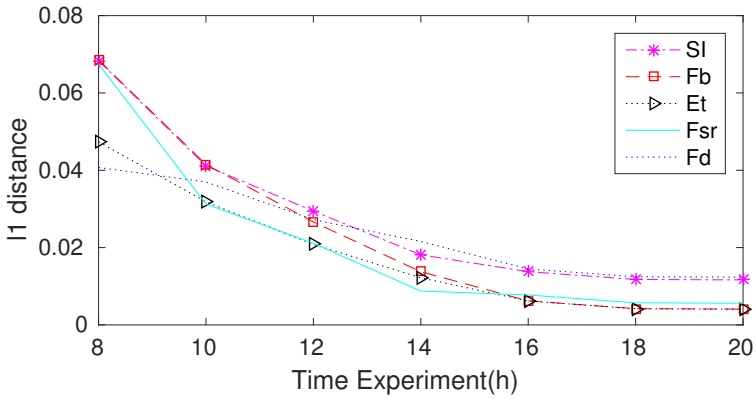


Figure 3.10: L_1 distance vs. time experiments (CTP – Indriya – MauveDB)

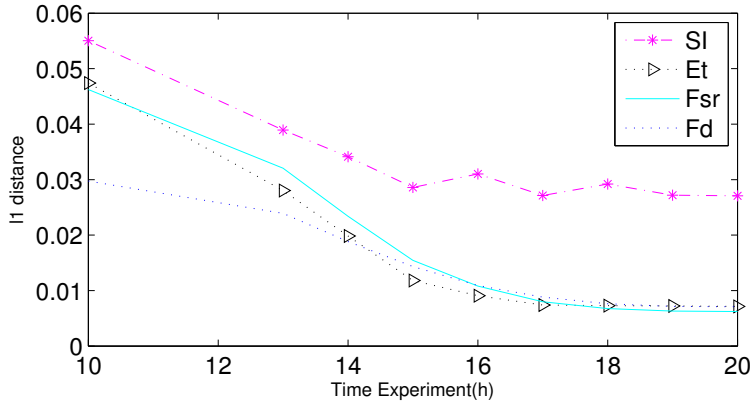


Figure 3.11: L_1 distance vs. time experiments (ORW – Indriya – MauveDB)

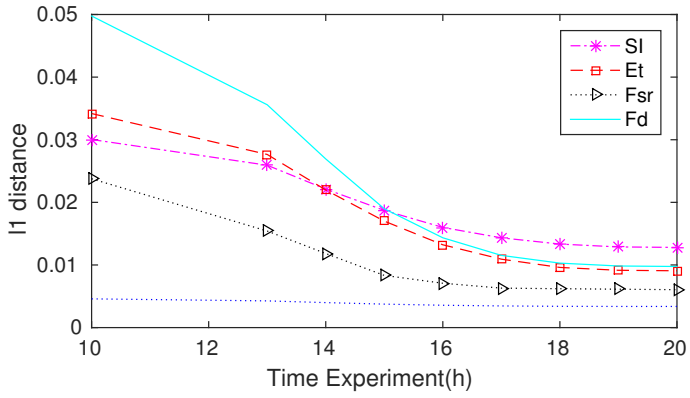


Figure 3.12: L_1 distance vs. time experiments (CTP – Local – POR)

We would like to gain a deeper understanding on the amount of data that is required to establish reasonable approximations for the functional relationships when deriving the overall model. Figures 3.8, 3.9, 3.10, 3.11, and 3.12 show the L_1 distance between the values of adjacent parameters as we increase the experimental time (one curve for each parameter). These figures provide an idea on how long it takes to converge to a small distance. When L_1 converges to a small value over time, there is no much to be gained by running longer experiments. From the figures, we can state that a minimum of 15 hours is required in order to get meaningful values that can lead to a good functional approximation without too much error due to a small sample size. This results are consistent for both CTP/BoX-MAC and ORW/X-MAC in both testbeds. In the remaining of the paper, we use 15 hours of data to determine all the functional relationships in the objective function and user constraints (see below).

3.4 Objective Function and User Constraints

In EOF, we have different objective functions by changing the time of experiments, routing, and sensing applications. Here we show the objective function as we have 15 hours time experiments, CTP and ORW routing applications, and MauveDB sensing application.

CTP:

$$r = -0.0007 + \frac{0.009}{Et} + 10Fd - 400Fd^2 \\ - 4Fb + 158Fb^2 + 0.2Fsr - 0.01S_i + 0.01S_i^2$$

ORW:

$$r = -0.006 + \frac{0.02}{E_t} + 4.6F_d - 120F_d^2$$
$$+ 0.3F_{sr} - 0.01S_i + 0.01S_i^2$$

The maximum time synchronization and data errors acceptable by the application's user can be provided to ensure that minimum application requirements are met while energy is minimized.

The data error is related to the sensing module. From the data collected, we have established the data error using 15 hrs of data and MauveDB sensing application as: The data error should be less than a threshold which is defined by the user (N_1).

$$Err_d = 0.3E_t + \frac{0.003}{F_d} - 0.5 \leq N_1$$

Similarly, the time synchronization error depends on the time synchronization module, and the formula is as follows: As another constraint, the time error should also be less than an user defined threshold (N_2).

$$Err_t = 0.2 + \frac{0.5}{F_{sr}} \leq N_2$$

All the other parameters are also changing in closed or open intervals:

$$2 \leq E_t \leq 100, 0.001 \leq F_b \leq 0.025, 4{\times}10^{-4} \leq F_d \leq 0.16, 0.05 \leq S_i \leq 2, 0.025 \leq F_{sr} \leq 0.4$$

3.5 Optimization

Finally, we have all the components to solve the following optimization problem:

$$E_t \rightarrow x_1, F_d \rightarrow x_2, F_b \rightarrow x_3, F_{sr} \rightarrow x_4, S_i \rightarrow x_5$$

$$\underset{x_1,x_2,x_3,x_4,x_5}{\text{minimize}}\ r(x_1,x_2,x_3,x_4,x_5) = \frac{a}{x_1} - bx_2^2 + cx_2 + dx_3^2$$
$$- ex_3 + fx_4 + gx_5^2 - hx_5 + i$$
$$a,b,c,d,e,f,g,h > 0$$

subject to

$$Err_d \leq N1 \rightarrow Err_d = jx_1 + \frac{k}{x_2} + l \leq N_1 \rightarrow jx_1 + \frac{k}{x_2} \leq N_1\prime$$
$$\rightarrow \frac{j}{y} + \frac{k}{x_2} \leq N_1\prime \rightarrow 0 \leq N_1\prime x_2 y - ky - jx_2$$
$$Err_t \leq N2 \rightarrow Err_t = \frac{m}{x_4} + n \leq N_2 \rightarrow \frac{m}{x_4} \leq N_2\prime \rightarrow \frac{m}{N2\prime} \leq x_4$$
$$0 \leq \alpha_1\prime \leq x_1 \leq \beta_1\prime \rightarrow \frac{1}{\beta_1\prime} \leq \frac{1}{x_1} \leq \frac{1}{\alpha_1\prime} \rightarrow 0 \leq \alpha_1 \leq y \leq \beta_1$$
$$0 \leq \alpha_2 \leq x_2 \leq \beta_2$$
$$0 \leq \alpha_3 \leq x_3 \leq \beta_3$$
$$0 \leq \alpha_4 \leq x_4 \leq \beta_4$$
$$0 \leq \alpha_5 \leq x_5 \leq \beta_5$$

with r() being the objective function and N_1 and N_2 being the quality of service constraints provided by the user.

There are different methods that can be used for optimizing the objective function:

- Quadratic penalty method and gradient projection, exact solution.
- SQP: Sequential quadratic programming is an iterative method for nonlinear optimization. SQP methods are used on mathematical problems for which the objective function and the constraints are twice continuously differentiable.

- Eliminating parameters that appear separately in both objective functions and constraints. We have used this method in our optimization:

We solved the above functions in a closed form. From the function we can see that the parameter x_5 is not related to other parameters in both function and the constraints. As a result, we can find the optimal value for x_5 separately. We found the optimal S_i value with a method similar to [PEF13].

Another parameter which can be solved separately, is x_3. x_3 is not related to the other parameters in the formulas, so it can be solved easily by optimizing the nonlinear part of the function. The x_4 parameter is not related to other parameters but it is part of one of the constraints. It can be solved individually from other parameters. x_1 and x_2 values are related to each other in the data error constraint formulation. We have eliminated the parameters which are seperate from other parameters and found the optimal values for them seperately.

$$F_3(x_3) = dx_3^2 - ex_3, 0 \leq d \rightarrow Convex$$
$$x_3* = median(e/2d, \alpha_3, \beta_3)$$
$$F_4(x_4) = fx_4, 0 \leq f \rightarrow Linear$$
$$x_4* = \frac{m}{N2\prime}$$
$$F_5(x_5) = gx_5^2 - hx_5, 0 \leq g \rightarrow Convex$$
$$x_5* = median(h/2g, \alpha_5, \beta_5)$$

Then our remaining objective function will be:

$$\underset{x_2,y}{\text{minimizer}}(y, x_2) = ay - b{x_2}^2 + cx_2$$

subject to

$$0 \leq N_1\prime x_2 y - ky - jx_2$$

$$0 \leq \alpha_2 \leq x_2 \leq \beta_2$$

$$0 \leq \alpha_1 \leq y \leq \beta_1$$

KKT method:

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in (I \cup \varepsilon)} \lambda_i c_i(x)$$

$$a) \nabla \underset{x}{\mathcal{L}}(x*, \lambda*) = 0$$

$$b) c_i(x*) = 0 \qquad \forall i \in \varepsilon$$

$$c) 0 \leq c_i(x*) \qquad \forall i \in I$$

$$d) 0 \leq \lambda_{i*} \qquad \forall i \in I$$

$$e) \lambda_i c_i(x*) = 0 \qquad \forall i \in (I \cup \varepsilon)$$

$$\lambda_1(\beta_2 - x_2) = 0$$

$$\lambda_2(x_2 - \alpha_2) = 0$$

$$\lambda_3(\beta_3 - y) = 0$$

$$\lambda_4(y - \alpha_3) = 0$$

$$\lambda_5(N_1 x_2 y - jx_2 - ky) = 0$$

Then we tried all the 32 different possible states and alot of them were elim-

inated. Here we show some of the important KKT steps:

$$First \quad active, others \quad non-active$$

$$\lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = 0$$

$$x_2 = \beta_2$$

$$\mathcal{L}(x, \lambda) = ay - bx_2^2 + cx_2 - \lambda_1(\beta_2 - x_2)$$

$$\nabla_x \mathcal{L}(x*, \lambda*) = 0$$

$$-2bx_2 + c + \lambda_1 = 0$$

$$\Rightarrow a = 0 \quad X$$

$$First \quad and \quad forth \quad active, \quad others \quad non-active$$

$$\lambda_2 = \lambda_3 = \lambda_5 = 0$$

$$x_2 = \beta_2$$

$$y = \alpha_3$$

$$\mathcal{L}(x, \lambda) = ay - bx_2^2 + cx_2 - \lambda_1(\beta_2 - x_2) - \lambda_4(y - \alpha_3) = 0$$

$$\nabla_x \mathcal{L}(x*, \lambda*) = 0$$

$$-2bx_2 + c + \lambda_1 = 0$$

$$a - \lambda_4 = 0 \Rightarrow a = \lambda_4$$

$$2bx_2 - c = \lambda_1 \Rightarrow \frac{c}{2b} \leq x_2$$

$$\Rightarrow y = \alpha_3, x_2 = \beta_2 \quad if \quad \frac{c}{2b} \leq x_2$$

$$Last \quad active, \quad others \quad non-active$$

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$$

$$N_1 x_2 y - j x_2 - k y = 0$$

$$\mathcal{L}(x, \lambda) = ay - bx_2^2 + cx_2 - \lambda_5(N_1 x_2 y - j x_2 - k y)$$

$$\nabla_x \mathcal{L}(x*, \lambda*) = 0$$

$$a + k\lambda_5 - \lambda_5 N_1 x_2 = 0$$

$$-2bx_2 + c - \lambda_5 N_1 y - j = 0$$

N_1 (%)	N_2 (μs)	E_t	S_i (s)	F_b (hz)	F_{sr} (hz)	$F_d * 10^3$ (hz)
0.1	3	1	0.5	0.01	0.3	16
4	3	5	0.5	0.01	0.3	1
8	3	8	0.5	0.01	0.3	0.5
12	3	21	0.5	0.01	0.3	0.5
16	3	35	0.5	0.01	0.3	0.5
20	3	48	0.5	0.01	0.3	0.5

Table 3.5: CTP: Optimization results when changing data error

N_1 (%)	N_2 (μs)	E_t	S_i (s)	F_b (hz)	F_{sr} (hz)	$F_d * 10^3$ (hz)
8	2	8	0.5	0.01	0.3	1.6
8	4	8	0.5	0.01	0.1	1.6
8	6	8	0.5	0.01	0.1	1.6
8	8	8	0.5	0.01	0.25	1.6

Table 3.6: CTP: Optimization results when time error changes

Tables 3.5, 3.6, 3.7, and 3.8 show the optimization results for CTP and ORW applications while having different user constraints (time and data errors). As the optimization results for CTP (for both sensing applications) and ORW on the Local and Indriya testbed were similar, we just show the results on Local testbed.

N_1 (%)	N_2 (μs)	E_t	S_i (s)	F_{sr} (hz)	$F_d * 10^3$ (hz)
0.1	3	1	1	0.3	16
4	3	5	1	0.3	1
8	3	8	1	0.3	0.5
12	3	21	1	0.3	0.5
16	3	35	1	0.3	0.5
20	3	48	1	0.3	0.5

Table 3.7: ORW: Optimization results when data error changes

N_1 (%)	N_2 (μs)	E_t	S_i (s)	F_{sr} (hz)	$F_d * 10^3$ (hz)
8	2	8	1	0.3	1.6
8	4	8	1	0.1	1.6
8	6	8	1	0.05	1.6
8	8	8	1	0.025	1.6

Table 3.8: ORW: Optimization results when time error changes

3.6 Summary

In this Chapter, we have focused on our overall architecture, which contains the software module, experimental setup, finding the relationships between different parameters of the system components as an objective function and adding some user constraints which are data and time errors to our objective function, and in the end, optimizing the function and finding the optimal values for each application tested in the system.

CHAPTER 4

Experimental Evaluation

In the experimental evaluation we want to determine how the inclusion of our optimized operational modules parameters affect the amount of radio transmissions that occur, as well of the merits of choosing a non-linear objective function. From the previous section we saw that the experiment length of 15 hours is enough in order to provide good approximations in the functional relationships. After finding the relationship formulations, we evaluate the overall application in experiments that run for 24 hours each. We evaluated the system in 2 different testbeds; Local testbed and Indriya testbed with two different routing/MAC modules (CTP/Box-MAC and ORW/X-MAC) and two different sensing applications (MauveDB and Pork). We compare three different results: results with default parameter values in each module, results obtained by EOF (optimization), and results of paper AODC [PEF13] while data and time error changes. For all the experiments that time error changes, data error is 8% and for the ones that data error changes, time error is 3 μsec. As mentioned in the Related Work, the scheme in [PEF13] finds the optimal value for SI by considering the reliability and latency constraints. By increasing sleep time, the reliability, throughput, and delay degrade because of increasing traffic. They propose a framework to optimize the system energy consumption by getting the optimal value of sleep time/interval while meeting the minimum requirements of reliability and delay constraints. In our work we find the optimal value of all the parameters including

Data Error (%)	Time Error (μs)	Et	SI (s)	Fb (hz)	Fs (hz)	Fd$*10^3$ (hz)
0.1	3	1	2	0.002	0.3	16
4	3	4	2	0.002	0.3	16
8	3	4	2	0.002	0.3	16
12	3	20	2	0.002	0.3	16
16	3	20	2	0.002	0.3	16
20	3	20	2	0.002	0.3	16

Table 4.1: CTP: Default results when data error changes

Data Error (%)	Time Error (μs)	Et	SI (s)	Fb (hz)	Fs (hz)	Fd$*10^3$ (hz)
0.1	3	1	0.5	0.002	0.3	16
4	3	4	0.5	0.002	0.3	16
8	3	4	0.5	0.002	0.3	16
12	3	20	0.5	0.002	0.3	16
16	3	20	0.5	0.002	0.3	16
20	3	20	0.5	0.002	0.3	16

Table 4.2: CTP: AODC when data error changes

SI.

In the following sections, we will use the parameters obtained from our EOF optimization framework, AODC framework, and default values in order to explore the results gained for different testbeds, software modules and sensing applications:

Data Error (%)	Time Error (μs)	Et	SI (s)	Fb (hz)	Fs (hz)	Fd$*10^3$ (hz)
8	2	4	2	0.002	0.3	16
8	4	4	2	0.002	0.3	16
8	6	4	2	0.002	0.1	16
8	8	4	2	0.002	0.05	16

Table 4.3: CTP: Default results when time error changes

Data Error (%)	Time Error (μs)	Et	SI (s)	Fb (hz)	Fs (hz)	Fd$*10^3$ (hz)
8	2	4	0.5	0.002	0.3	16
8	4	4	0.5	0.002	0.3	16
8	6	4	0.5	0.002	0.1	16
8	8	4	0.5	0.002	0.05	16

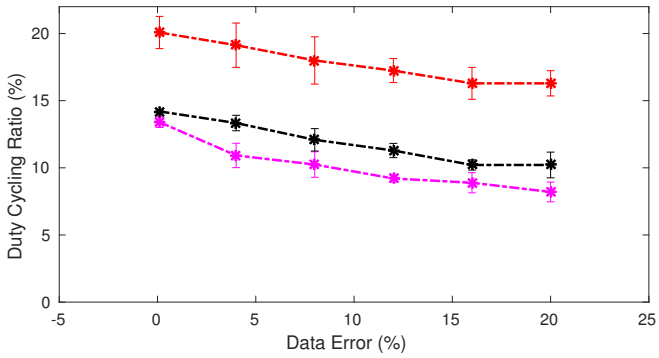
Table 4.4: CTP: AODC results when time error changes

4.1 CTP/BoX-MAC on Local Testbed using MauveDB

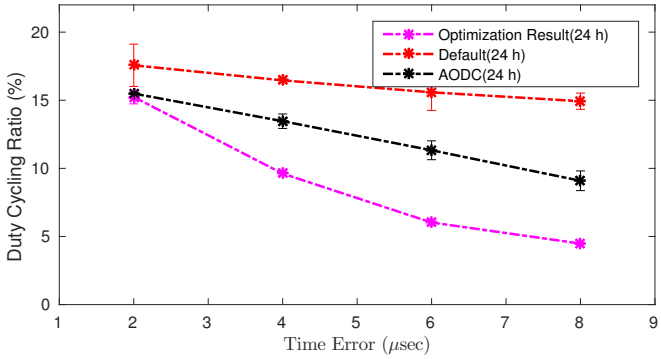
First we will show the parameter values which will be used as default values and values gained by the AODC framework for applications using CTP from the tables; 4.1, 4.2, 4.3, and 4.4. The results of ORW and CTP (Pork and MauveDB) on Indriya testbed is similar to the results on local testbed. As a result, we just show the table results for ORW and CTP.

Figure 4.1a shows the results as the data error user requirements change and Figure 4.1b shows the results as the time error user requirements change. According to [PEF13], we used AODC framework and optimized the functions by considering reliability and delay for finding the SI optimal value . This is because by increasing SI value, the network traffic increases and reliability of the system decreases.

AODC technique finds the optimal value for SI at which we have a good reliability and latency. From Figure 4.1 we see that the energy consumption decreases as the errors increase. As admissible data error increases, we have energy reduction in the average of 38% by using our EOF framework compared to just using the default values. We also get an energy reduction in average of 14% by using EOF compared to the AODC framework. As time error admissible increases, we have energy reduction in average of 55% by using EOF compared



(a)

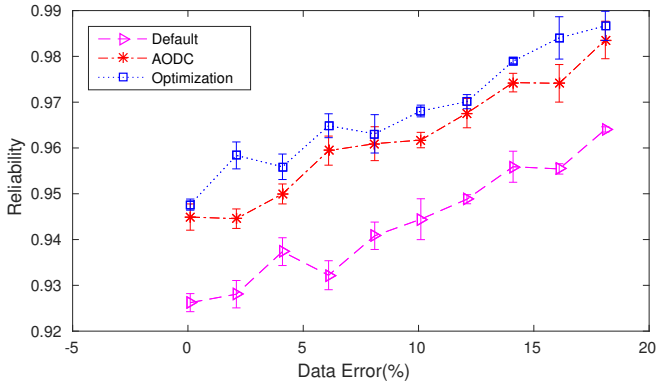


(b)

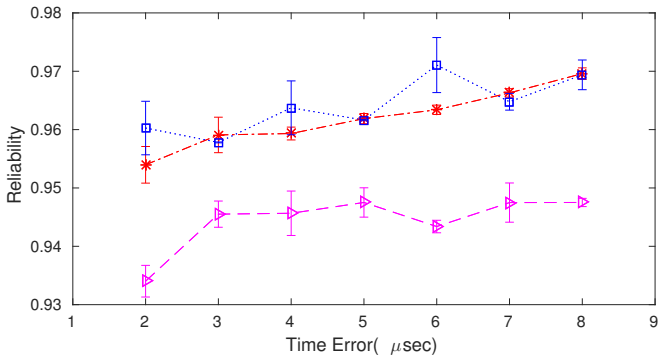
Figure 4.1: Radio duty cycling ratio vs. a) data error and b) time error (MauveDB – CTP – Local).

to just using the default values. We also get an energy reduction on average of 33% by using EOF compared to the AODC framework.

We also evaluate the reliability and latency of the system. Figures 4.2 and 4.3 show that by using EOF we even have a higher reliability and lower latency compared to the results of AODC and default values. This is because the optimization results cause the system to have lower traffic, hence we have higher reliability and lower delay. Figure 4.2 shows how the reliability of the system changes based on different time and data errors. In general, as data and time errors increase, the reliability of the system increases. The reason is that by increasing the error in this specific application, the number of data packets decreases as well, so we have



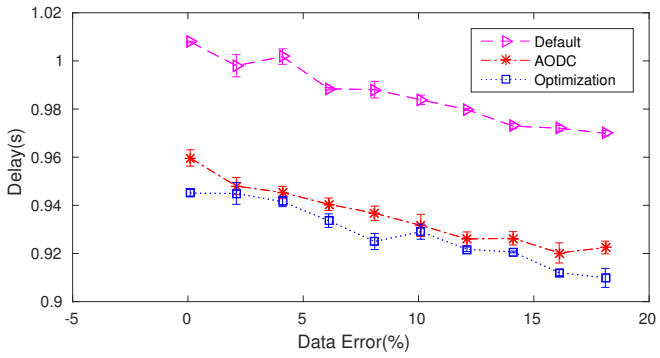
(a)



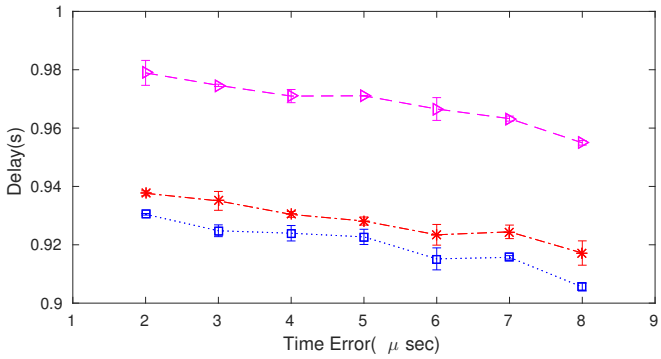
(b)

Figure 4.2: The reliability vs. data error and time error (CTP – Local)

less traffic in the network. By having less traffic, the reliability of the system increases. Figure 4.3 shows that as data and time errors increase, the network traffic increases and as a result, the latency of the system decreases. By comparing the 3 mentioned techniques, we obtain that reliability of the system by using EOF is about in the average of 4% larger than the default value and 2% larger than the values obtained by the AODC framework and we also saw that the average latency of the system is 0.06 seconds less than using default values by using our framework and 0.02 seconds less than the average latency of the system by using AODC framework.



(a)

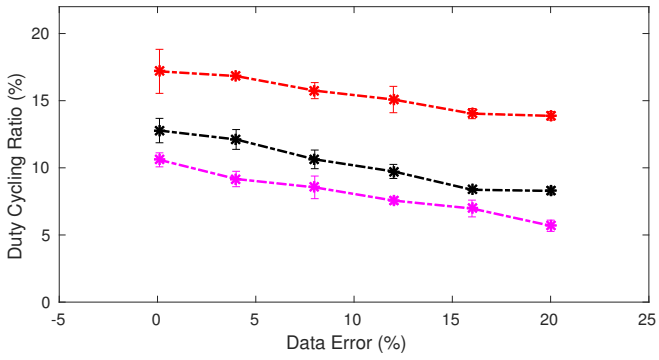


(b)

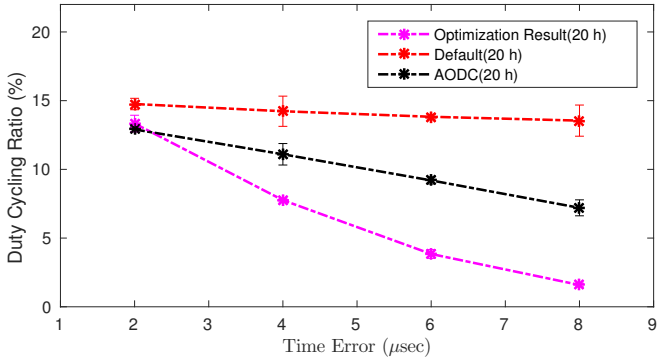
Figure 4.3: The delay vs. data error and time error (CTP – Local)

4.2 CTP/BoX-MAC on Indriya Testbed using MauveDB

After finding the relationship formulations for Indriya testbed, minimizing the objective function, and finding the optimal values for different parameters, we have evaluated the system for results gained for Indriya testbed. Figure 4.4 shows us how the radio duty cycling ratio of system changes as time and data errors of the system change. From the figure we can see that as time and data errors increase, the radio duty cycling ratio of the network decreases. Figure 4.4a displays that we have obtained in the average of 44% reduction in energy consumption by using EOF compared to using the default values and 16% reduction compared



(a)



(b)

Figure 4.4: Radio duty cycling ratio vs. a) data error and b) time error (MauveDB –CTP – Indriya).

to AODC framework for different data errors. Figure 4.4b shows that we have in the average of 62% reduction in energy by using EOF compared to using the default values for parameters and 32% reduction compared to using AODC.

4.3 ORW/X-MAC on Local Testbed using MauveDB

At this section we evaluate the framework by using results gained by optimizing ORW formulations. First we will show the parameter values which will be used as default values and values gained by the AODC framework for applications using

Data Error (%)	Time Error (μs)	Et	SI (s)	Fs (hz)	Fd$*10^3$ (hz)
0.1	3	1	2	0.3	16
4	3	4	2	0.3	16
8	3	4	2	0.3	16
12	3	20	2	0.3	16
16	3	20	2	0.3	16
20	3	20	2	0.3	16

Table 4.5: ORW: Default results when data error changes

Data Error (%)	Time Error (μs)	Et	SI (s)	Fs (hz)	Fd$*10^3$ (hz)
0.1	3	1	1	0.3	16
4	3	4	1	0.3	16
8	3	4	1	0.3	16
12	3	20	1	0.3	16
16	3	20	1	0.3	16
20	3	20	1	0.3	16

Table 4.6: ORW: AODC results when data error changes

CTP from the tables; 4.5, 4.6, 4.7, and 4.8. The results of ORW and CTP (Pork and MauveDB) on Indriya testbed is similar to the results on local testbed. As a result, we just show the table results for ORW and CTP.

By testing the application using ORW on our local testbed, we see that like the other application, the radio duty cycling ratio decreases as the time and data errors increase. figure 4.5a shows that running EOF on the other application (using ORW rather than CTP), we have gained reduction in energy consumption

Data Error (%)	Time Error (μs)	Et	SI (s)	Fs (hz)	Fd$*10^3$ (hz)
8	2	4	2	0.3	16
8	4	4	2	0.3	16
8	6	4	2	0.1	16
8	8	4	2	0.05	16

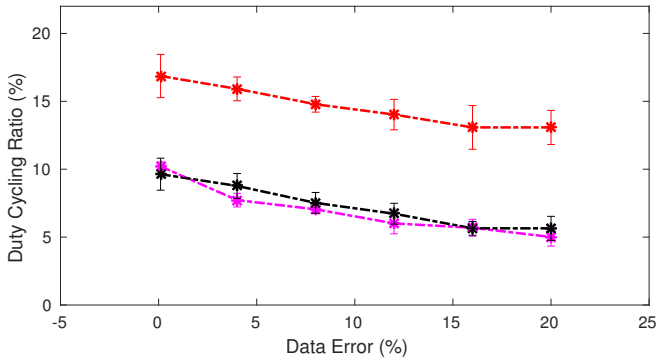
Table 4.7: ORW: Default results when time error changes

Data Error (%)	Time Error(μs)	Et	SI (s)	Fs (hz)	Fd$*10^3$ (hz)
8	2	4	1	0.3	16
8	4	4	1	0.3	16
8	6	4	1	0.1	16
8	8	4	1	0.05	16

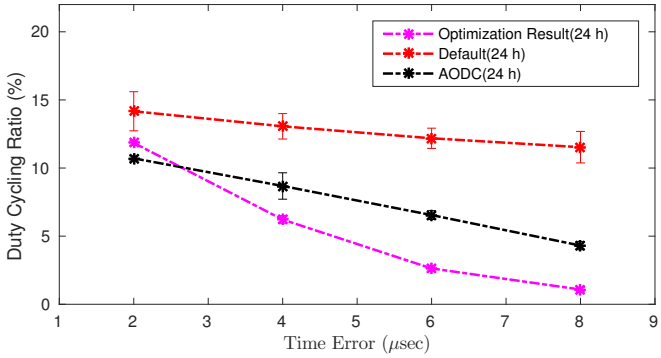
Table 4.8: ORW: AODC results when time error changes

in the average of 50% compared to using the default values and average of 11% reduction compared to using the results gained by running AODC framework as data error changes. We have also obtained reduction in energy consumption in the average of 60% compared to using default values and average of 13% compared to using the results of AODC framework as time error changes.

After evaluating the energy consumption of the application using ORW, we have evaluated the reliability and latency of the system. From the figures 4.6 and 4.7 we can see how the reliability and latency of the system change over data and time errors. Similar to the previous application (application with CTP), as the errors increase, the reliability of the system increases and the latency decreases. Figure 4.6a shows that the reliability of the system after using the architecture is 3% larger than the reliability of system by using default values and 1% larger than while we use the AODC framework as data error changes. We also have the average of 3.5% and 1.5% larger reliability while time error changes. Another metric that needs to be evaluated after experimenting this application is latency of the system. Figure 4.7 shows how the latency of the system is behaving as data error and time error change. By increasing the errors, the latency of the system decreases as we have less traffic in the network. We see that the latency of the system by using the optimized values for the parameters is 0.08 seconds smaller than the latency of the system using default values and 0.01 seconds smaller while using AODC framework over different data errors from



(a)



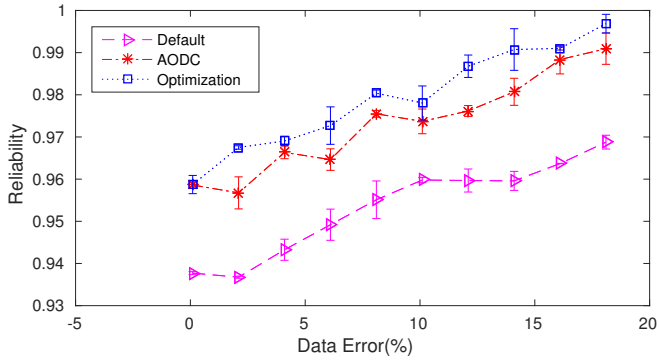
(b)

Figure 4.5: Radio duty cycling ratio vs. a) data error and b) time error (MauveDB – ORW – Local).

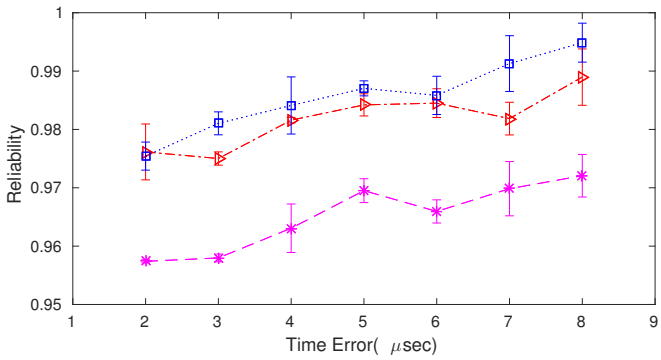
figure 4.6a and 0.05 and 0.02 seconds smaller over different time errors as shown in figure 4.6b.

4.4 ORW/X-MAC on Indriya Testbed using MauveDB

We obtained all the relationships between radio duty cycling ratio and components of the system and optimized the energy consumption from the section **??**. By using the optimal parameters, we have evaluated the energy consumption of the system. Similar to the other sections in this chapter, we see energy con-



(a)



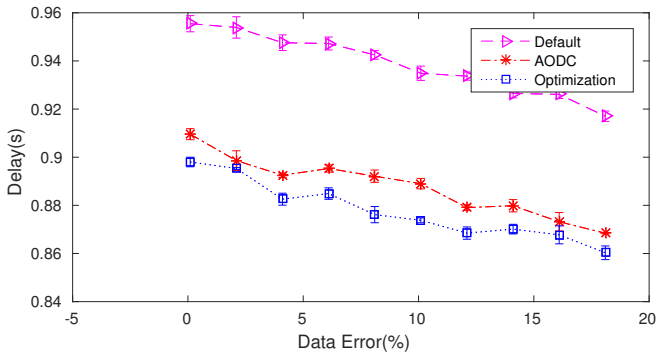
(b)

Figure 4.6: The reliability vs. data error and time error (ORW – Local)

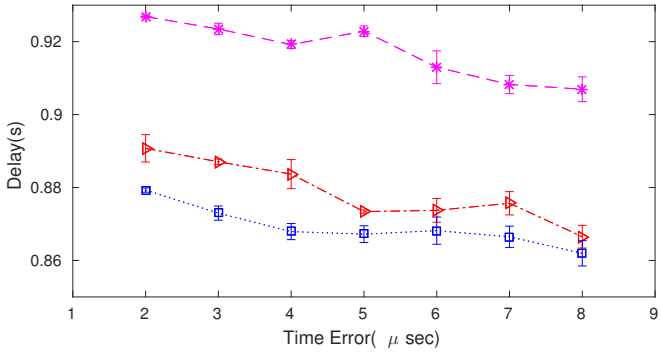
sumption reduction as time and data errors increase. Figure 4.8a shows that as data error increases we have energy reduction of about 13% by using the optimal values compared to using the AODC and we have in the order of 33% energy reduction compared to using the default values. Figure 4.8b shows that as time error increases, we have 60% energy reduction by using EOF compared to using default values and around 30% energy reduction compared to using AODC.

4.5 CTP/BOX-MAC on Local Testbed using Pork

After deploying the second sensing application (Pork) on the local testbed and finding the relationships between different parameters of each component, we use



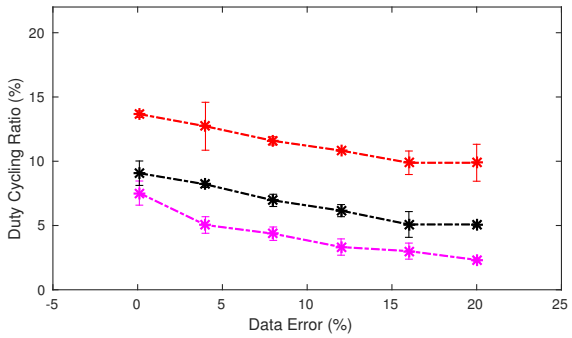
(a)



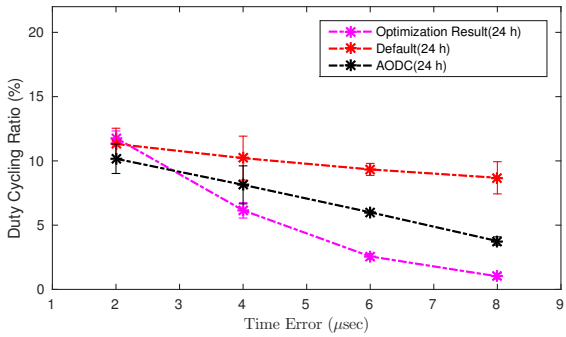
(b)

Figure 4.7: The delay vs. data error and time error (ORW – Local)

the optimal results in order to evaluate the energy consumption of the system. Similar to the other applications, as we increase the time and data errors, we have lower duty cycling ratio. By using Pork, we obtain a better fit for time vs. temperature model at each node. By having a better model, we have a lower difference between predicted and actual temperature. As a result, less packets will be sent in the network. Furthermore, The energy consumption of the network decreases. Figure 4.9 shows the radio duty cycling results as we increase time and data errors. By comparing the figure with figure 4.1, we have a lower energy consumption by using Pork. Also figure 4.9a shows that as the errors increase, the energy consumption decreases faster than the results of the figure 4.1a. This is also because of having a better model fit and having less data error by using



(a)



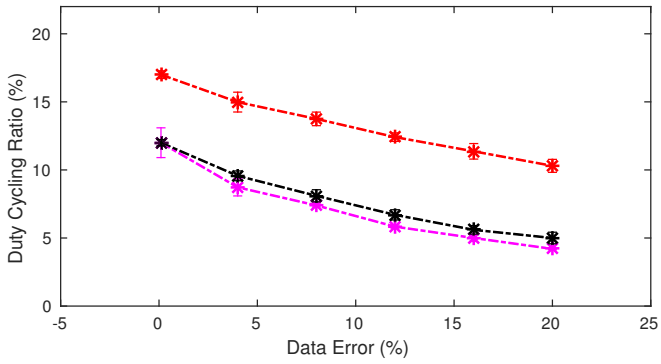
(b)

Figure 4.8: Radio duty cycling ratio vs. a) data error and b) time error (MauveDB – ORW – Indriya).

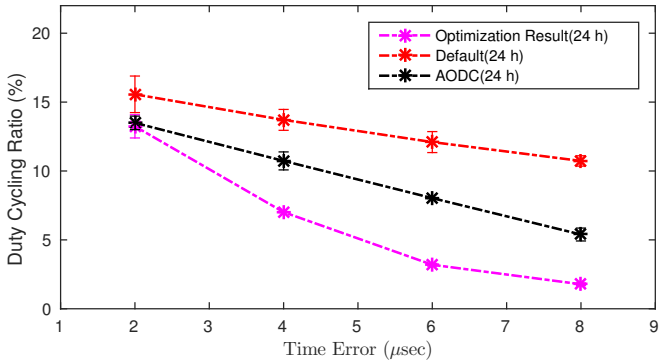
Pork.

4.6 Summary

In this chapter, we have used the optimal results gained by chapter 3. We have evaluated the five tested applicaitons (MauveDB – CTP – Local, MauveDB – CTP – Indriya, MauveDB – ORW – Local, MauveDB – ORW – Indriya, and Pork – CTP – Local). For each application, we have compared the radio duty cycling results of testing the application with EOF optimal values, default values, and AODC framework. From all the results we can conclude that using EOF results



(a)



(b)

Figure 4.9: Radio duty cycling ratio vs. a) data and b) time error (POR – CTP – Local).

has a better energy consumption than default and AODC results.

CHAPTER 5

Conclusion

We have presented an architecture for energy management of the wireless sensor nodes in a holistic manner. Our framework is based on a model-driven approach which has 3 steps: 1) Calculating the functional relationships among different software modules and their parameters. 2) Using the user constraints in the objective function. 3) Building an objective function based on the functional relationships among components and optimizing based on user constraints. We have used a non-trivial application consisting of three main components: sensing, routing and time synchronization. In order to build the EOF framework, we have tested our application with two different sensing modules (MauveDB and POR), two routing modules (ORW and CTP), and one time synchronization (FTSP) module on two different testbeds (Local and Indriya) with different time experiments. Radio communication is the most important source of energy consumption, as a result it can be a good metric for calculating the energy consumption of a system. We have created the functional relationship between the software modules and radio duty cycling ratio by considering the time and data errors as the user constraints. We have concluded that 15 hours experiments are enough for finding accurate relationships. We have evaluated the system after optimizing the objective function and finding the optimal values. We show that when using our framework, we can provide average energy savings from 38% to 62% when compared to the default values, and from 11% to 33% when com-

pared to the state-of-the-art AODC duty-cycle optimization scheme while still maintaining quality of service both in terms of the expected sensing and time-synchronization errors. We further show that we can slightly decrease latency (1.5%-3.5%) and slightly improve reliability (1%-3%).

By using EOF, we can minimize the energy consumption of a system in an holistic manner when we have more than two components working together. We used our architecture for different combinations of modules. We have shown that our framework worked for any combinations of applications. However, it needs plenty of experiments in order to create the functional relationships for each.

The most important limitation is that our framework needs huge amount of training data and so, more experiments for each set of modules in order to be created. In our work we have created the framework for 3 different sets of applications at 2 different environments. As we create the system for more conditions, we will need less hours of experiments for different applications that have similar modules to our application. We will need to do all the experiments again if we have completely new modules.

There are several possibilities for future improvements. At this time we have done around 130 hours of experiments in order to create the EOF framework. We need to minimize the number of experiments in order to obtain the accurate formulations. One thing that we can do in order to make the system less costly, is to use on-line training model instead of using an off-line trainer. We should train the model at run-time. As a result the formulations will be created on-line and will make the framework less costly with less numbers of experiments. Right now the framework is devoted to the applications with the three basic software modules (routing/MAC, sensing, and time synchronization). Although our framework worked for four different applications, we have done hours of

experiments for each setof components. We need to have few hours of experiments to create EOF, as a result the framework will work for any application with any software modules with less numbers of experiments than now.

References

[AAG07] Cesare Alippi, Giuseppe Anastasi, Cristian Galperti, Francesca Manici, and Manuel Roveri. "Adaptive Sampling for Energy Conservation in Wireless Sensor Networks for Snow Monitoring Applications." pp. 1–6, October 2007.

[BC08] Prithwish Basu and Chi-Kin Chau. "Opportunistic Forwarding in Wireless Networks with Duty Cycling." In *Proceedings of the Third ACM Workshop on Challenged Networks*, CHANTS '08, pp. 19–26, New York, NY, USA, 2008. ACM.

[BKY10] Nouha Baccour, Anis Koubâa, Habib Youssef, Maissa Ben Jamâa, Denis do Rosário, Mário Alves, and Leandro B. Becker. "F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks." In *Proceedings of the 7th European Conference on Wireless Sensor Networks*, EWSN'10, pp. 240–255, Berlin, Heidelberg, 2010. Springer-Verlag.

[BM05] Sanjit Biswas and Robert Morris. "ExOR: Opportunistic Multi-hop Routing for Wireless Networks." *SIGCOMM Comput. Commun. Rev.*, **35**(4):133–144, August 2005.

[BYA06a] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard 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*, SenSys '06, pp. 307–320, New York, NY, USA, 2006. ACM.

[BYA06b] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard 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*, SenSys '06, pp. 307–320, New York, NY, USA, 2006. ACM.

[BZV10] C. A. Boano, M. A. Zuniga, T. Voigt, A. Willig, and K. Romer. "The Triangle Metric: Fast Link Quality Estimation for Mobile Wireless Sensor Networks." In *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1–7, Aug 2010.

[CAB05] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. "a high-throughput path metric for multi-hop wireless routing." *Wireless Networks*, **11**(4):419–434, 2005.

[CPR03] Jim Chou, Dragan Petrovic, and Kannan Ramchandran. "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks." **2**:1054–1662, April 2003.

[DGV11] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. "Valuable Detours: Least-Cost Anypath Routing." *IEEE/ACM Transactions on Networking*, **19**(2):333–346, April 2011.

[DL03] Tijs van Dam and Koen Langendoen. "An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks." In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pp. 171–180, New York, NY, USA, 2003. ACM.

[DLV13] Simon Duquennoy, Olaf Landsiedel, and Thiemo Voigt. "Let the Tree Bloom: Scalable Opportunistic Routing with ORPL." *11th ACM Conference on Embedded Networked Sensor Systems*, (2), 2013.

[DM06] Amol Deshpande and Samuel Madden. "MauveDB: supporting model-based user views in database systems." *SIGMOD '06 Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pp. 53–84, 2006.

[ED04] Amre El-Hoiydi and Jean-Dominique Decotignie. *WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks*, pp. 18–31. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[EGE02] Jeremy Elson, Lewis Girod, and Deborah Estrin. "Fine-grained Network Time Synchronization Using Reference Broadcasts." *SIGOPS Oper. Syst. Rev.*, **36**(SI):147–163, December 2002.

[FGJ07] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. "Four-Bit Wireless Link Estimation." In *Proceedings of the Sixth ACM Workshop on Hot Topics in Networks (HotNets-VI)*, October 2007.

[FZS08] Kai-Wei Fan, Zizhan Zheng, and Prasuan Sinha. "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks." *SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems.*, pp. 239–252, 2008.

[GBA13] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic. "Edge Mining the Internet of Things." *IEEE Sensors Journal*, **13**(10):3816–3825, Oct 2013.

[GFJ13] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, Maria Kazandjieva, David Moss, and Philio Levis. "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks." *ACM Trans. on Sensor Networks (TOSN).*, **10**(16), November 2013.

[GGS01] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. "Highly-resilient, energy-efficient multipath routing in wireless sensor networks." *Newsletter ACM SIGMOBILE Mobile Computing and Communications Review*, **5**:11–25, October 2001.

[GKS03a] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. "Timing-sync protocol for sensor networks." *Proceeding SenSys '03 Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 138–149, 2003.

[GKS03b] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. "Timing-sync Protocol for Sensor Networks." In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pp. 138–149, New York, NY, USA, 2003. ACM.

[HJA13] N. Q. V. Hung, H. Jeung, and K. Aberer. "An Evaluation of Model-Based Approaches to Sensor Data Compression." *IEEE Transactions on Knowledge and Data Engineering*, **25**(11):2434–2447, Nov 2013.

[JC04] Ankur Jain and Edward Y. Chang. "Adaptive sampling for sensor networks." 2004.

[JCW04] Ankur Jain, Edward Y. Chang, and Yuan-Fang Wang. "Adaptive Stream Resource Management Using Kalman Filters." In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pp. 11–22, New York, NY, USA, 2004. ACM.

[JRO10] Raja Jurdak, Antonio G. Ruzzelli, and Gregory M. P. O'Hare. "Radio Sleep Mode Optimization in Wireless Sensor Networks." *IEEE Transactions on Mobile Computing*, **9**(7):955–968, July 2010.

[KLS10] J. Kim, X. Lin, N. B. Shroff, and P. Sinha. "Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks With Anycast." *IEEE/ACM Transactions on Networking*, **18**(2):515–528, April 2010.

[KLS11] J. Kim, X. Lin, and N. B. Shroff. "Optimal Anycast Technique for Delay-Sensitive Energy-Constrained Asynchronous Sensor Networks." *IEEE/ACM Transactions on Networking*, **19**(2):484–497, April 2011.

[KPS05] W. J. Kaiser, G. J. Pottie, M. Srivastava, G. S. Sukhatma, J. Villasenor, and D. Estrin. "Networked Infomechanical Systems (NIMS)." pp. 88–113, 2005.

[LFS07] S. Liu, K. W. Fan, and P. Sinha. "CMAC: An Energy Efficient MAC Layer Protocol Using Convergent Packet Forwarding for Wireless Sensor Networks." In *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 11–20, June 2007.

[LGD12] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. "Low power, low delay: opportunistic routing meets duty cycling." pp. 185–196, 2012.

[LP08] Wei Lai and Ioannis C. Paschalidis. "Optimally Balancing Energy Consumption Versus Latency in Sensor Network Routing." *ACM Trans. Sen. Netw.*, **4**(4):21:1–21:28, September 2008.

[LSK10] Ren-Shiou Liu, Prasuan Sinha, and Can Emre Koksal. "Joint Energy Management and Resource Allocation in Rechargeable Sensor Networks." pp. 1–9, March 2010.

[MA00] Arati Manjeshwar and Dharma P. Agrawal. "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks." *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pp. 2009–2015, April 2000.

[MA01] Arati Manjeshwar and Dharma P. Agrawal. "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Network." *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM*, April 2001.

[MDc13] Gaurav Mathur, Peter Desnoyers, Paul chukiu, Deepak Ganesan, and Prashant Shenoy. "Ultra-Low Power Data Storage for Sensor Networks." *ACM Trans. on Sensor Networks (TOSN).*, **10**(16), November 2013.

[MFH03] Samuel Madden, Michael J. Franklin, and Wei Hong Joseph M. Hellerstein. "The design of an acquisitional query processor for sensor networks." pp. 359–379, 2003.

[MH10] C. J. Merlin and W. B. Heinzelman. "Duty Cycle Control for Low-Power-Listening MAC Protocols." *IEEE Transactions on Mobile Computing*, **9**(11):1508–1521, Nov 2010.

[MKS04] Mikls Marti, Branislav Kusy, Gyula Simon, and kos Ldeczi. "The flooding time synchronization protocol." *Proceeding SenSys '04 Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 39–49, 2004.

[ML08] David Moss and Philip Levis. "BoX-MACs: Exploiting Physical and Link Layer Boundaries in LowPower Networking." Technical report, Computer Systems Laboratory Stanford University, 2008.

[MLT08] R. Musaloiu-E., C. J. M. Liang, and A. Terzis. "Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks." In *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, pp. 421–432, April 2008.

[MWZ10] Andreas Meier, Matthias Woehrle, Marco Zimmerling, and Lothar Thiele. "ZeroCal: Automatic MAC Protocol Calibration." In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'10, pp. 31–44, Berlin, Heidelberg, 2010. Springer-Verlag.

[NC10] Xu Ning and Christos G. Cassandras. "Dynamic Sleep Time Control in Wireless Sensor Networks." *ACM Trans. Sen. Netw.*, **6**(3):21:1–21:37, June 2010.

[PEF13] Pangun Park, Sinem Coleri Ergen, Carlo Fischione, and Alberto Sangiovanni-Vincentelli. "Duty-cycle Optimization for IEEE 802.15.4 Wireless Sensor Networks." *ACM Trans. Sen. Netw.*, **10**(1):1–32, December 2013.

[PHC04] Joseph Polastre, Jason Hill, and David Culler. "Versatile Low Power Media Access for Wireless Sensor Networks." In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pp. 95–107, New York, NY, USA, 2004. ACM.

[PK00] G. J. Pottie and W. J. Kaiser. "Wireless Integrated Network Sensors." *Commun. ACM*, **43**(5):51–58, May 2000.

[PPL09] T. R. Park, K. J. Park, and M. J. Lee. "Design and analysis of asynchronous wakeup for wireless sensor networks." *IEEE Transactions on Wireless Communications*, **8**(11):5530–5541, November 2009.

[PSM04] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson. "Wireless Sensor Networks." chapter Analysis of Wireless Sensor Networks for Habitat Monitoring, pp. 399–423. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[PVK08] T. Palpanas, M. Vlachos, E. Keogh, and D. Gunopulos. "Streaming Time Series Summarization Using User-Defined Amnesic Functions." *IEEE Transactions on Knowledge and Data Engineering*, **20**(7):992–1006, July 2008.

[RCM12] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. "What does model-driven data acquisition really achieve in wireless sensor networks?" In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pp. 85–94, March 2012.

[RCM15] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. "Practical Data Prediction for Real-World Wireless Sensor Networks." *Knowledge and Data Engineering, IEEE Transactions on*, **27**(8):2231–2244, August 2015.

[SHL13] Mo Sha, Gregory Hackmann, and Chenyang Lu. "Energy-efficient Low Power Listening for Wireless Sensor Networks in Noisy Environments." In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pp. 277–288, New York, NY, USA, 2013. ACM.

[SR02] Rahul C. Shah and Jan M. Rabaey. "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks." *Wireless Communications and Networking Conference*, **1**:350–355, March 2002.

[SS01] Curt Schurgers and Mani B. Srivastava. "Energy efficient routing in wireless sensor networks." *Military Communications Conference. Communications for Network-Centric Operations: Creating the Information Force*, **1**:357–361, 2001.

[SZH04] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks." *Proceeding SenSys '04 Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 108–121, 2004.

[TM06a] Daniela Tulone and Samuel Madden. "An Energy-efficient Querying Framework in Sensor Networks for Detecting Node Similarities." In *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '06, pp. 191–300, New York, NY, USA, 2006. ACM.

[TM06b] Daniela Tulone and Samuel Madden. "PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks." In *Proceedings of the Third European Conference on Wireless Sensor Networks*, EWSN'06, pp. 21–37, Berlin, Heidelberg, 2006. Springer-Verlag.

[WLJ06] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. "Fidelity and Yield in a Volcano Monitoring Sensor Network." In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pp. 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[WLR06] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. "Deploying a wireless sensor network on an active volcano." *IEEE Internet Computing*, **10**(2):18–25, March 2006.

[XRC04] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. "A Wireless Sensor Network For Structural Monitoring." In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, pp. 13–24, New York, NY, USA, 2004. ACM.

[YHE02] Wei 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*, volume 3, pp. 1567–1576 vol.3, 2002.

[ZCH11] Ziguo Zhong, Pengpeng Chen, and Tian He. "On-demand time synchronization with predictable accuracy." *INFOCOM, 2011 Proceedings IEEE*, pp. 2480–2488, April 2011.

[ZFM12] M. Zlmmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele. "pTUNES: Runtime parameter adaptation for low-power MAC protocols." In *Information Processing in Sensor Networks (IPSN), 2012 ACM/IEEE 11th International Conference on*, pp. 173–184, April 2012.

[ZR03] M. Zorzi and R. R. Rao. "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance." *IEEE Transactions on Mobile Computing*, **2**(4):337–348, Oct 2003.