UNIVERSITY OF CALIFORNIA, SAN DIEGO

Energy efficient data aggregation in wireless sensor networks

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical engineering (Communication Theory and Systems)

by

Jinseok Yang

Committee in charge:

Tajana Šimunić Rosing, Chair
Prof. Chung-Kuan Cheng
Prof. William G. Griswold
Prof. Tara Javidi
Prof. Ryan Kastner

2015

Copyright
Jinseok Yang, 2015
All rights reserved.

The dissertation of Jinseok Yang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2015

DEDICATION

To my parents, Jaeyul and Soonjoo & my loves Jiyoung, Ellie and Olivia

EPIGRAPH

We are drowning in

information and starving for knowledge.

—Rutherford D. Roger

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost I would like to thank my advisor Tajana Simunic Rosing. It has been an honor to be her Ph.D student. I appreciate her guidance and support not only in my research but also in my life. I appreciate all her contributions of time, ideas, and funding to make my Ph.D experience productive and stimulating. The joy and enthusiasm she has for her research was contagious and motivational for me, even during tough times in my Ph.D pursuit.

I would like to thanks the Dr. Sameer Tilak, without whom my research would have no doubt taken five times as long. It is his support that helped me in an immeasurable way. My research was possible by funding from National Science Foundation (NSF) Project SensorRocks, Terraswarm Research Center and UCSD Center for Networked Systems (CNS). I thank them for their support.

I cannot forget all of Seelab members. Their advice and comments helped keep me motivated in my research. I would like to thank my parents, Jaeyul Yang and Soonjoo Chung for all you have done for me. My brother Hongseok Yang, I am always proud of you and grateful for your care for me. I also want to thank my wife Jiyoung Kim, and our little special presents Ellie Yang and Olivia Yang. Your existence warms my heart. Lastly, I would like to thank my father and mother in law, Changhun Kim and Jungwon Son.

Chapter 2, in part, is a reprint of the material as it appears in ”Leaveraging application constext for efficient sensing”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE ISSNIP, 2014. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 2, in part, is a reprint of the material as it will appear in ”An Interactive Context-aware Power Management Technique for Optimizing Sensor Network Lifetime”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, SENSORNETS 2016. The

dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in ”Transmission manager in heterogeneous applications running WSNs”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE Globecom 2015. The dissertaion author was the primary investigator and author of this paper.

Chapter 3, in part, has been submitted for publication of the material as it may appear in ”Design of transmission manager in heterogeneous WSNs”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, which was submitted to IEEE Transactions on Emerging Topics in Computing. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in ” A novel protocol for adaptive broadcasting of sensor data in urban scenarios”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE Globecom, 2013. The dissertation/thesis author was the primary investigator and author of this paper.

VITA

2006	B. S. in Computer Science, Ajou University, South Korea
2009	M. S in Information technology, Ajou University, South Korea
2011	M. S in Electrical Engineering (Communication Theory and Systems), University of California, San Diego
2009-2011	Teaching assistance in Mechanical Engineering, University of California, San Diego
2011	Teaching assistance in Electrical Engineering, University of California, San Diego
2015	Ph. D. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

JS. Yang, A.S. Akyurek , S. Tilak, T.S. Rosing, Design of transmission manager in heterogeneous WSNs, submitted to IEEE Transactions on Emerging Topics in Computing, 2015

JS. Yang, S. Tilak, T.S. Rosing, An Interactive Context-aware Power Management Technique for Optimizing Sensor Network Lifetime, SENSORNETS, 2016

JS. Yang, S. Tilak, T.S. Rosing, Transmission manager in heterogeneous applications running WSNs, IEEE Globecom, 2015

P. Aghera, JS. Yang, P. Zappi, D. Krishnaswamy, A. Coskun, T.S. Rosing, Energy management in wireless mobile systems using dynamic task allocation, Journal of Low Power Electronics, Vol. 9, no. 2, Aug. 2013

JS. Yang, T.S. Rosing, S. Tilak, Leaveraging application constext for efficient sensing, IEEE ISSNIP, 2014

B. Milosevic, JS. Yang, N. Verma, P. Zapi, S. Tilak, E. Farela, T. S. Rosing, L. Benini, Eficient Energy Management and Data Recovery in Sensor Networks using Latent Variables Based Tensor Factorization, IEEE/ACM MSWiM, 2013

JS. Yang, T.S. Rosing, S. Tilak, A novel protocol for adaptive broadcasting of sensor data in urban scenarios, IEEE GlobeCom, 2013

N. Nikizad, JS. Yang, P. Zappi, T.S. Rosing, D. Krishnaswamy, 'Model-driven Adaptive Wireles Sensing for Environmental Healthcare Fedback Systems, IEEE ICC, 2012

ABSTRACT OF THE DISSERTATION

Energy efficient data aggregation in wireless sensor networks

by

Jinseok Yang

Doctor of Philosophy in Electrical engineering (Communication Theory and Systems)

University of California, San Diego, 2015

Tajana Šimunić Rosing, Chair

Wireless sensor networks (WSNs) form a critical interface between physical and digital worlds by converting physical qualities into measurements which can be used for a wide-ranging spectrum of applications. In the past, these WSNs have been application-specific and exposed only to a limited set of users. Going forward, WSNs will no longer be specialized networks. The new emerging IoT will run multiple applications that have diverse delay requirements for generated and received measurements. In this thesis, we propose a power management framework that operates on the sensing platforms which have multiple power managers. For example, sensors are controlled by sensor controller and battery is managed by battery manager in order to minimize the unnecessary energy

consumption. The proposed framework integrates these different power management approaches and optimizes their interactions to achieve optimality in terms of energy efficiency. Proposed approach saves 20% to 60% of energy consumption compared to the state of art approaches. In addition, we propose an optimal transmission manager that supports multiple applications in single-hop wireless sensor networks. We formulate the problem with Markov decision process model and dynamically adjust transmission instances based on random delay requirements of buffered measurements. Then, we propose a distributed transmission manager that leverages the optimal transmission manager to operate in multi-hop WSNs. We implement both transmission managers in ns3 simulator and compare with other state of the art designs. The results show that the proposed transmission manager consumes on average 148.3% less energy than the state of the art approaches while on average having 14.1% fewer measurements that expire. Lastly, we propose adaptive information dissemination protocol in order to provide information to users in vicinity. Sensors estimate users moving speed and adjust information provision interval in order to save energy. The results show that our proposed approach decrease in power consumption by a factor of 2x to 8x in a single sensor, and 2x to 16x in 10 node sensor network, when compared to the state of art approaches.

Chapter 1

Introduction

Wireless sensor networks (WSNs) are a cyber-physical interface to the physical world. They have been applied to various domains such as Smart Cities, Smart Buildings and environmental monitoring systems. Environmental monitoring systems [20][21] deploy either small scale nodes (ref. Fig. 1.1 (left)) or large scale systems (ref. Fig. 1.1 (right)) in order to observe natural phenomena variations in remote regions. Ecologists or scientists then access to the aggregated data and analyze them to monitor the biotic and abiotic factors in the area of interest [20]. Research in Smart Buildings [27][40] uses inputs from deployed sensors to control corresponding building components in order to decrease utility bills while guaranteeing user satisfaction. For example, light control system [40] monitors occupancy level of each room with motion detection sensors and controls the level of luminosity. Personal health care systems [44][36] adopt WSNs in order to collect health care related information from patients who live in remote regions. One of the key challenges in WSNs is their short battery lifetime. WSNs play an important role in data aggregation that collect raw data of the whole sensing field through data exchange and aggregation with their neighboring nodes [70]. Minimizing energy consumption indirectly decreases the amount of data forwarded to the sink. Methods that

Figure 1.1: Environmental monitoring sensing platforms, (left) Heliomote [33] (right) Buoy deployed in a lake located in northern Wisconsin, USA measuring several key limnological variables [68]

guarantee measurement quality while increasing energy efficiency are needed.

A number of power management approaches [13][26][32][33][58][68] have been proposed to address the challenges of WSNs. Energy efficient protocols [13][32][58][65] increase network lifetime with energy efficient MAC and routing protocols. They achieve energy efficiency by significantly reducing the amount of data to be transported to the sink. The amount of data reduction is constrained by required measurement quality. WSNs with energy harvesting [15][26][33][39][68] replenish energy from ambient sources in order to overcome battery capacity limitations. They treat the harvested energy as auxiliary energy source and smartly adjust system parameters such as wireless communication rate in order to keep a preset amount of remaining battery capacity. Previous approaches [26][33][68] decide how to allocate energy to various tasks (sensing, communication etc.) over time so that the deployed network continues to gather high-quality data. For example, solar panel equipped sensor nodes in Fig. 1.1 runs energy allocation algorithm in order to determine optimal sampling rate and duty cycle based on the current battery level & the amount of harvested energy. Again, these approaches also constrains the energy allocation based on data quality requirements.

However, even though there are many studies increase the lifetime of WSNs much

remains to be done. Over last years, more works are still needed in order to increase energy efficiency of WSNs.

Energy harvesting based power managers [16][33][47] minimize the waste of energy in low power embedded system by predicting the behavior of energy sources over short and medium time frames and adjusting task performance level of the system. Several prediction algorithms such as EWMA [33], WCMA [47] and Pro-Energy [16] have been proposed. The algorithms predict energy level at future time slots based on structured model which is constructed from find-grained training data set. Their prediction accuracy deteriorates in heavily shaded areas, such as forests and urban areas because these situations result in course grained training set of harvested energy data. The prediction error increases up to 30% without fine grained data set. Thus, a prediction algorithm that can accurately estimate future energy arrival with coarse-grained training set is needed.

Power managers [13][26][32][33][58][68] which run in each sensing platform control the amount of energy expenditure in order to increase the operation lifetime given data quality constraints. Even though they have the common goal, they control different hardware components in the sensing platform. For example, approaches in [26][32][33] adjust duty cycle in MAC layer to save idle energy consumption. Sensor controllers [10][34] adjust sampling rate and increase the lifetime. Sensor controller use linear regression models from observed variables and decrease sampling rate based on the level of temporal correlation. This decreases the communication energy cost by decreasing the data upload frequency. Application based approaches [12][70] handle buffering delay of generated and received packets in WSNs based on time constraints of them. They buffer all packets in order to decrease communication cost by decreasing the amount of transmitted packets. In a reality, all sensor nodes are equipped with both sensors and wireless radios, so approaches stated above can be implemented in a single

sensor node at the same time. Thus, a framework that integrates the operations of various power managers is needed.

Power manager with applications [12][70] adjust data forwarding interval based on delay tolerance of the application. For example, HVAC control application [25] use sensing data coming from WSNs, but it does not require real-time information. Then, nodes in WSNs buffer measurement and transmit with aggregated packets. This decreases the number of transmission activities, and then increases the energy efficiency. However, their work assumes that there is only a single application running, so they cannot operate in WSNs which run multiple applications that have diverse delay requirements. In fact, the new emerging Intenet Of Things (IoT) will have multiple application share in common WSNs infrastructure, so application based power manager that support heterogeneous types of measurements is needed. Traditionally, WSNs collect raw data from the sensors, upload to centralized server and users consume the collected data. However, with IoT, users can directly access to near sensor nodes and collect necessary information as needed. This requires a new information provision approach.

This thesis addresses all the challenges described above. We next outline the contributions of this thesis.

1.1 Thesis contributions

1.1.1 Adaptive power management framework

We propose a power management framework that includes sensor controller, battery manager and application based power manager. We identify the necessary interactions among different power managers and then integrate into a single power management framework. The framework is optimal in terms of energy efficiency with low computational complexity. The framework also includes an user interactive component,

so system administrator can adjust sampling rate based on their needs. We logically divide battery capacity into normal and emergency components in order to protect against to sudden sampling rate changes. When processes consume less energy than expected, the power manager allocates the remaining to emergency component. Power manager supports sudden high of requests with that extra energy. Our simulations use sensor data and system specifications (battery and solar panel specs, sensing and communication costs) from a real sensor network deployment. Our results show that the proposed approach saves significant amounts of energy by avoiding oversampling when application does not need it while using this saved energy to support sampling at high rates to capture events with necessary fidelity when needed. The more detail algorithms and experimental results are described in Chapter 2.

1.1.2 Heterogeneous application transmission manager

The goal of the transmission manager in WSNs with heterogeneous applications running is to adjust the delay at the data buffer in order to decrease the number of transmission activities by exploiting application specific delay requirements and the current topology of WSNs. The well-known approach, cascade time-out protocol (Cas) [55] determines the buffering delay based on a nodes distance to the sink. Their approach is completely distributed, but it does not consider the fact that heterogeneous applications require different timeout constraints. Application specific transmission managers are proposed by [12][11][70]. Their transmission managers determine optimal transmission time of buffered measurements based on the characteristics of a running application. However, they assume that all nodes in the wireless network run a single delay sensitive application. None of above approaches can support heterogeneous WSNs which support multiple applications with different delay requirements.We propose a transmission manager that determines the optimal transmission instance for measurements based on

applications' delay tolerance, using optimal stopping theory in Markov Decision Process. We implement our novel transmission manger in ns3 simulator [45] to compare it with other state of art transmission managers. We evaluate the energy consumption and the number of expired measurements under three different network topology: i) single hop, ii) linear topology and iii) grid topology. In all three cases, proposed transmission manager consumes average 148.3% less energy than state of art approaches while having on average 14.1% fewer expired measurements as compared to the state of art approaches.

1.1.3 User based energy efficient data aggregation in WSNs

In a number of sensing applications, users are only interested in the data relevant to their present location and current type. This implies that WSNs need a low-power protocol that allows mobile users, while moving around the deployed stationary sensor networks, to access the data from the sensors in their vicinity. This approach is different from mobile sink approaches [54][66] which typically assume that the system has control over the sink mobility to ensure that it collects data from all the sensors. Unlike mobile sink approaches, we assume that users carry their mobile phones while moving throughout the area where sensors are deployed. Recent work proposed a technique where a sensor node wakes up when it receives a RFID impulse from a user and then it unicasts data [32]. However, this mechanism requires users to carry an RFID reader that is expensive and cumbersome. We propose transmission manager that dynamically adjust transmission times based on user mobility. Transmission manager adjust its broadcasting rate as a function of the travel time of users in their transmission range (broadcast area). Proposed approach estimates the travel time based on handshaking process between sensing platform and users. The results show a decrease in power consumption by a factor of 2x to 8x in a single sensor, and 2x to 16x in 10 node sensor network, when compared to the existing protocols. More details on the algorithms and experimental results are described in Chapter 4.

Chapter 2

Adaptive power management framework

2.1 Introduction

Sensor networks are revolutionizing the scientific applications by gathering data about the environment at unprecedented spatio-temporal granularity [1][3][19][38][50]. A key problem in WSNs is deciding how to allocate energy to various tasks (sensing, communication etc.) over time so that the deployed network continues to gather high-quality data. There has been extensive research in the area of power management and resource allocation algorithms in sensor networks. A good example of a fair energy allocation algorithm is progressive filling [26] that takes into account battery levels and harvested energy to provide uniform sampling rates [68]. It has two implications: allocating as much energy as possible results in high frequency sampling and fair energy allocation over time results in sampling the environment at a fixed rate. Our experience with real-world sensor network deployments in collaboration with limnologists and coral reef ecologists shows that the high frequency fixed rate sampling technique does not

work well in practice because of the following reasons.

1) Real-world deployments depend on periodic interaction to maintain optimal sampling regime: Sensor networks need periodic interaction primarily for the following two reasons. (A) Early identification of system failures: Sensor networks embedded in inhospitable environment are prone to fail for a variety of reasons such as biofouling, exposure to extreme temperature or humidity etc. (B) Identification of interesting trends: Both anticipated (nightly temperature drops) and unanticipated episodic events (typhoon, hurricanes etc.). At present the interaction is manual, where the domain scientists periodically look at the incoming data to ensure that it is generating science-quality data [20]. Scientists also often explore the data to see if something interesting happened in last day or two and whether the current sampling rate is sufficiently capturing the events with necessary fidelity. At present, it is too complex to automate this process. This is both due to lack of a priori knowledge of the all possible events and system failures and specifying and capturing all the interesting events and system failures. In addition, even if the events are known, programming and detecting all possible events makes the system prohibitively complex. In future, as machine learning algorithms will be more sophisticated and sensor networks become equipped with more computing power, we believe that this manual approach will be replaced by an automated system that requires no human interaction. However, either a real end user or an automated system will interact with the deployed network on a periodic basis. In this thesis we use the term ”user request” to denote both the request generated by human beings as well as automated systems.

2) Periodic sampling at fixed rate is not sufficient: Sampling the environment at fixed predefined intervals is neither reliable (need to accommodate system failures) nor desirable (current sampling rate might not capture an important event with desired fidelity) [20]. We now explain this in detail. *Failure*: Suppose monitoring system monitors both temperature and humidity level every 1 min. In many applications if

either measurement is missing the other is useless. This indicates that the missing value should be compensated by either repeating a measurement within a few seconds or not sampling. *Interesting events*: Scientists often want systems that can adapt sampling rates to meet their science requirements. Consider an application that requires sampling a sensor at a high rate (i.e. 10 samples/seconds) when rain is detected and otherwise a much lower sample rate (i.e. 1 sample/minute). Periodic sampling can often result in either oversampling (thereby wasting energy) or under sampling (thereby not capturing an event with necessary fidelity).

3) Setting sampling regime is often an exploratory and iterative process: Scientists are often operating in unexplored territory and therefore setting up sampling rate is not a one-time process, but is an iterative and exploratory process. Scientists typically set the sampling rate to the best of their knowledge and then use the gathered data to adjust it. This process can take anywhere from a few days to a few months.

Even if the sampling were set optimally, there are still significant challenges with data transmission. The state of the art approaches for determining the optimal transmission policy include control-limit policy [70] and MDP based approaches [39]. The control-limit policy does not consider heterogeneous sensors. The MDP based approach [39] requires complete knowledge of transition probability of the harvesting level and channel condition variations, which is unrealistic. In addition, both offer limited control over the energy-delay tradeoff and are not suitable for real-time and delay-tolerant applications.

Based on all above observations, we propose i) power manager that adapts sampling rate as a function of both application-level context (e.g., user request) and system-level context (e.g harvesting energy availability and stored energy) and we leverage it to build a ii) power management framework (APMF) that integrates together adaptive sampling (AS) and transmission policy (TP). The interactive power manager is

described in Section 2.4 while the adaptive power management framework is described in Section 2.5

2.2 Related work

There has been considerable work in the area of sensor network reprogramming [31][37][42]. These approaches are mainly designed for rare network-wide software updates and are not suitable for more frequent sampling rate updates. The industrial automation systems or building management systems integrated with control system require guaranties for real-timeliness, functional safety, security, energy efficiency, etc [21]. In these sensor-actuator networks resource allocation decisions are typically done in a centralized manner (at the plan data center). In contrast, we propose a fully distributed approach for energy allocation.

Context has been used extensively for efficient sensor network protocol design in the area of routing [36][71], cluster formation [28], and power management [63]. Wood et al. [63] proposed a context aware power management protocol considers heterogeneous energy sources in which some nodes are powered by batteries and others are plugged into wall. However, they do not consider green energy sources in their research. Gorlatova et al. [26] proposed an algorithm to determine fair energy allocation along time dimension in systems with predictable as well as stochastic renewable energy inputs. Their energy allocation algorithm – Progressive Filling (PF) fairly allocates energy over time dimension and it has $O(n^2)$ computational complexity. PF algorithm starts from time slot 0 and increments its allocated energy by α until it reaches the target battery level. Since PF is the state-of-the-art energy algorithm, we use it to compare with our proposed interactive technique. In this work we use interchangebly PF algorithm and non-interactive technique. To the best of our knowledge, this is the first work that

proposes a novel interactive power management technique that adapts sampling rate as a function of both application-level context (e.g., user request) and the system-level context (e.g harvesting energy availability).

2.3 System model

We model a typical environmental monitoring system. Our system consists of two components (1) field deployed sensor network (2) data center. The sensor network consists of a network of platforms (e.g. buoys or towers), which are large enough to house large solar panels and bulky batteries and an embedded computer to which multiple sensors (order of 30) are connected either via serial or Bluetooth link. The computer runs a low-power operating system and is equipped with one or more network modalities (e.g, WiFi, cellular, and satellite). Figure 1.1 (right) shows the latest deployment of an instrumented buoy for a lake monitoring application. This buoy hosts a variety of sensor for monitoring lake processes, including temperature at twenty seven depths, dissolved oxygen, conductivity, pH/ORP, flourescence sensors (Chloraphyll a, Blue-green Algae, and Rhodamine WT) and voltage. These sensors are connected to an Android Cell phone via IOIO board. The phone runs the data acquisition program and sends data back to a data center over the cellular network. We use one Instapark 80W Mono-crystalline solar panel as our green energy source [4]. It has following power specifications: Maximum Power Voltage: 17.39V; Open Circuit Voltage: 21.97V; Maximum Power Current: 4.61A. Table 2.1 summarizes the sensing and communication and processing power consumption, which we use in our simulations.

The data center provides facilities to process store and visualize the gathered data. It also provides capabilities to remotely command and control the field deployed sensor network. A user monitors collected data and determines current optimal sampling rate

Table 2.1: Power specification for our deployment

Device	Power consumption(W)
3G cost+Processing	5
Vaisala Weather Station	0.168
Sonde	3.372
Templine	0.42
Sensing cost	3.96

that meets the science requirements. Sensor nodes receive the request and adjust their configuration based on onboard power management technique (described later).

2.4 Interactive power manager

Our approach uses application-context (e.g., feedback from domain scientists or an automated system running user-specified rules) to optimally set sensor sampling rates. Figure 2.1 describes the proposed power management framework that runs at each sensor node. It consists of two major subsystems, namely, power manager (PM) and and the Interactive Resource Allocator (IRA) subsystem. The PM subsystem makes resource allocation decisions based on the current battery level and predicted harvesting level (ref. Eq (2.1)). The IRA subsystem then adapts the aforementioned sampling rate in an interactive manner (ref. Algorithm 2). We now describe the details of PF and IRA algorithms.

Each sensor node divides a time into K slots [26]. We denote $S = [s_1, ..., s_K]$ as a set of allocated energy to K time slots and s_i is the allocated energy to slot i. The embedded power manager allocates energy to each time slot based on the current batter level and predicted battery level (ref. Eq. (2.1)). $B(i)$ is battery level and $H(i)$ is the predicted harvest level at time slot i. We use B_{max} to denote total Battery capacity and B_{min} to denote minimum battery capacity. $U(\cdot)$ is utility function that calculates sampling rate given allocated energy. It is a non-negative, increasing, strictly concave function [57].

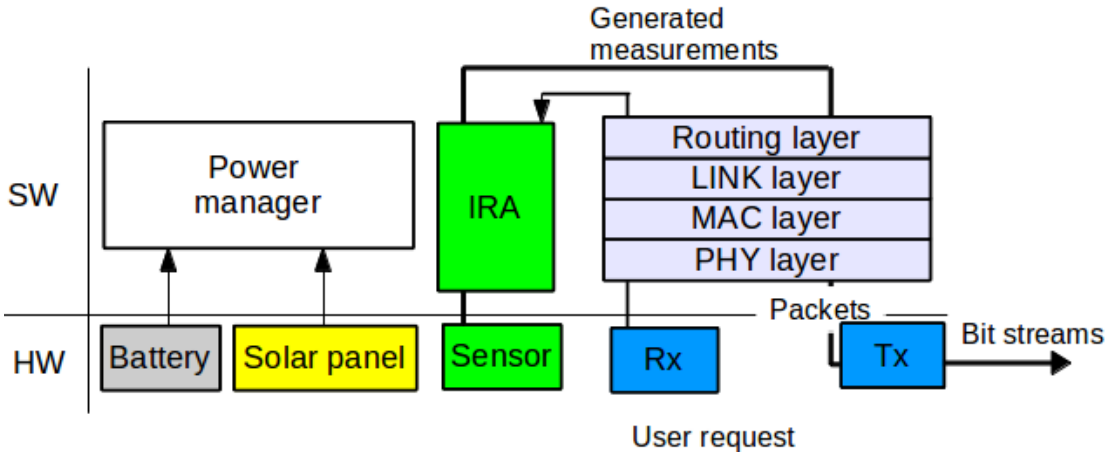

Figure 2.1: Interactive Context-Aware Power Management System Architecture

The first constraint of Eq. (2.1) obeys the energy neutral operation in energy harvested wireless sensor network[33].

$$\begin{aligned} &\max_{s_i} \sum_{i=1}^{K} a_{belief} \cdot (s_i) \\ &s.t. B(i) \leq B(i-1) + H(i-1) - s(i-1), \\ &s_i \leq B_{max}, s_i \geq 0 \;\; H(i) \geq 0, \quad B_{min} \leq B(i) \leq B_K \end{aligned} \tag{2.1}$$

In order to solve Eq. (2.1), PM allocates constant energy over K time slots as described in Algorithm 1. The validity of this approach has been proved by [26][57]. This reduces the computational complexity of PM to $O(n)$ compared to that of PF $O(n^2)$. However, constant energy allocation does not consider a situation where system needs to consume more energy than harvested. Interactive resource allocator (IRA) considers those scenario.

We now describe the IRA subsystem. The sensor node virtually divides its battery into two parts, $B_{current}$ and B_{saved}. The IRA subsystem (ref. Algorithm 2) interacts with the application (user or automated system) and then calculates the energy required to meet the requested sampling rate. Intuitively, when the energy needed to satisfy the user request

ALGORITHM 1: Advanced Progressive Filling(APF)

$avgHarvstEnergy = \sum_{i=1}^{K} H(i)/K$;
$s(1:K) := avgHarvestEnergy$;
for $i = K; i \geq 1; i = i-1$ **do**
    $[over, amount] \leftarrow check\_validity(s(i))$;
    if $over$ == $TRUE$ **then**
        $s(i) = s(i) - amount$;
    end
end

`Function` $[over, amount] = check\_validity(s)$
$B_{current} =$ `current battery level`
for $i = 1; i \leq K; i = i+1$ **do**
    $B_{current} \leftarrow \min\{B(i) + Q(i) - s(i), B_{max}\}$;
    if $B_{current} < s(i+1)$ **then**
        $return\ [TRUE, s(i+1) - B_{current}]$;
    end
end
$return\ [FALSE, 0]$;

(s^{req}) is less than the energy allocated ($s(i)$) by the APF algorithm (system is currently oversampling), IRA turns down the current sampling rate and saves this extra energy to B_{saved} while achieving the necessary fidelity. However, when user requires sampling at higher rate than the current sampling rate (the system is currently under sampling), the sensor node augments $B_{current}$ with B_{saved} to support it. When ($s(i) + B_{saved} < s^{req}$ given $s(i) < s^{req}$), the system is under sampling because it does not have enough energy to support the requested sampling rate. In this case, we consider three policies (1) Aggressive: The ongoing event is so critical that the user sees benefit in capturing that even at the cost of reduced network lifetime. In this case, the IRA algorithm increases the sampling rate for the given slot to the requested rate. (2) Conservative: IRA algorithm decides to continue sampling at the current sampling rate at the cost of reduced fidelity. (3) Hybrid policy: The system selects the best sampling rate it can support in a greedy manner. This happens in the case where although the requested rate is not feasible due to energy constraints, but there is still benefit in increasing the sampling rate to the level that can be supported.

ALGORITHM 2: Interactive Resource Allocator

```
1  s ← Algorithm 1
2  for i = 1; i ≤ K; i = i + 1 do
3      s^req ← sampling rate given s(i) ;
4      if s^req < s(i) then
5          // Support user request
6          s(i) ← s^req ;
7          B_saved ← B_saved + {s(i) − s^req} ;
8      else
9          if s^req < s(i) + B_saved then
10             s(i) ← s^req ;
11             B_saved = B_saved − {s^req − s(i)};
12         else
13             Aggressive :  s(i) ← s^req ;
14             Conservative :  s(i) ← s(i) ;
15             Hybrid :  s(i) ← s(i) + B_saved ;
16         end
17     end
18 end
```

2.4.1 Mathematical Analysis

In this section, we theoretically compare the performance of interactive and non-interactive power management approaches in terms of user satisfaction. The interactive approach makes its decision based on battery level, predicted harvesting energy, and user request, while the non-interactive approach typically considers only the first two as its inputs. At a given slot i, when the allocated energy ($s(i)$)is larger than energy required to meet the user request (s^{req}), a sensor node can satisfy user request at that slot. However, in this case, the node is oversampling and wasting its energy.

$$z_i = \begin{cases} 1 & if \;\; s(i) \geq s^{req} \\ 0 & if \;\; s(i) < s^{req} \end{cases} \tag{2.2}$$

As shown in equation (2.2), z_i defines the condition of i's time slot. When $s(i)$ is larger than s^{req} the z_i has 1 which indicates that the system is oversampling and spending extra energy. Otherwise z_i has 0 as shown in equation (2.2).

Let us define the probability $p = Pr(s(i) \geq s^{req})$. The average number of slots in which a node spends oversamples and wastes energy is given by equation (2.3). Thus, in this case, the average number of time slots in which the non-interactive approach overspends energy in $K \cdot Pr\ (s_i \geq x_i)$.

$$E[Z_i] = K \cdot p = K \cdot Pr(s(i) \geq s^{req}) \tag{2.3}$$

The proposed interactive power manager (ref. Algorithm 2) saves energy when the energy needed to satisfy user request (s^{req}) is less than allocated energy ($s(i)$), and uses the saved energy (B_{saved}) as a boost when the energy needed to satisfy a user request is more than allocated energy. The interactive approach with hybrid policy fails to satisfy user requests only when the sum of allocated and saved energy is lower than the amount of user request, $s(i) + B_{saved} < s^{req}$. We describe this in equation (2.4).

$$z_i = \begin{cases} 1 & if\ \ s(i) + B_{saved} \geq s^{req} \\ 0 & if\ \ s(i) + B_{saved} < s^{req} \end{cases} \tag{2.4}$$

The $s(i) + B_{saved} \geq s^{req}$ includes both $s_i \geq x_i$ and $s(i) + B_{saved} \geq s^{req}$ situations. Thus, the average number of time slots that satisfy user request with interactive approach is $K \cdot \{Pr(s(i) \geq s^{req}) + Pr(s(i) + B_{saved} \geq s^{req})\}$. This result means that user-interactive power management always satisfies more user requests than the non-interactive mechanism because $Pr(s(i) + B_{saved} \geq s^{req}) \geq 0$. The interactive approach with conservative policy will show same performance with non-interactive one and aggressive policy always satisfies user satisfaction.

2.4.2 Results

US climate Reference Network(USCRN), maintains a database of environmental data collected from various monitoring stations across the US. For our simulations, for solar energy prediction, we use data from USCRN database for Necedah, Wisconsin location since it is the closest location to our deployment. Our past research has shown that the state-of-the-art energy predictors such as Weather-Conditioned Moving Average, WCMA can be used to accurately predict the amount of harvesting energy [47]. Therefore, in this thesis we use WCMA algorithm for solar energy prediction. To calculate accuracy, we use one week worth of sensor data (Wind speed data) from our deployment. We use Matlab to conduct simulations.

Study of impact of time slot length variations on energy efficiency

In this study, we consider 24 hours duration and vary the time slot length from 1 (24 slots/day) hour to 24 hours (1 slot/day). We fix 1 sample per 10 min as default sampling rate. We consider user request pattern from 10% to 100%. In the case of 10% request pattern, among all time slots, 10% time slot support high request rate which requires 1 sample every 1 min. Remaining request pattern requires 1 sample every 5min. We use end point battery level as 11.1V. Thus, the capacity is (12-11.1)*55 = 49.5Wh. Table 2.2 shows percentage of energy consumed for each approach for different time slot lengths and request patterns. As expected (ref. Table 2.2), when we decrease request frequency, the overall energy consumption decreases. However, we observe an interesting patten when time slot length is varied. When time slot length is between 1 hour to 6 hours, the environmental conditions (for solar energy production) do not vary considerably and the overall energy consumption goes up as a function of slot length. However, for lengths greater than 6 hours the environmental conditions within a slot can vary significantly thereby changing the harvesting energy production (solar energy

Table 2.2: Impact of time slot length and request pattern variations on energy efficiency

Request pattern	1 hr	2 hr	3 hr	4 hr	6 hr	12 hr	24 hr
10%	0.956	0.834	0.831	1.293	0.9	0.469	0.376
20%	1.457	1.496	1.417	1.342	1.193	1.186	0.449
30%	1.677	1.69	1.821	1.929	2.074	1.251	0.889
40%	2.178	2.179	2.188	2.173	2.44	1.512	1.109
50%	2.508	2.791	2.554	2.663	2.514	1.512	1.403
60%	3.107	2.962	2.885	3.201	3.174	1.903	1.622
70%	3.302	3.279	2.995	3.543	3.541	2.36	2.063
80%	3.608	3.475	3.655	3.983	4.201	2.556	1.916
90%	3.95	3.695	3.801	4.081	4.055	2.686	2.43
100%	4.207	4.086	4.242	4.374	4.275	2.882	2.283

availability during day-night shifts). This results in lower energy consumption for 12 hours and 24 hours slot lengths as compared to slots of 1, 2, 3, and 4 hours duration.

Study of impact of End Point Voltage variations on energy-efficiency

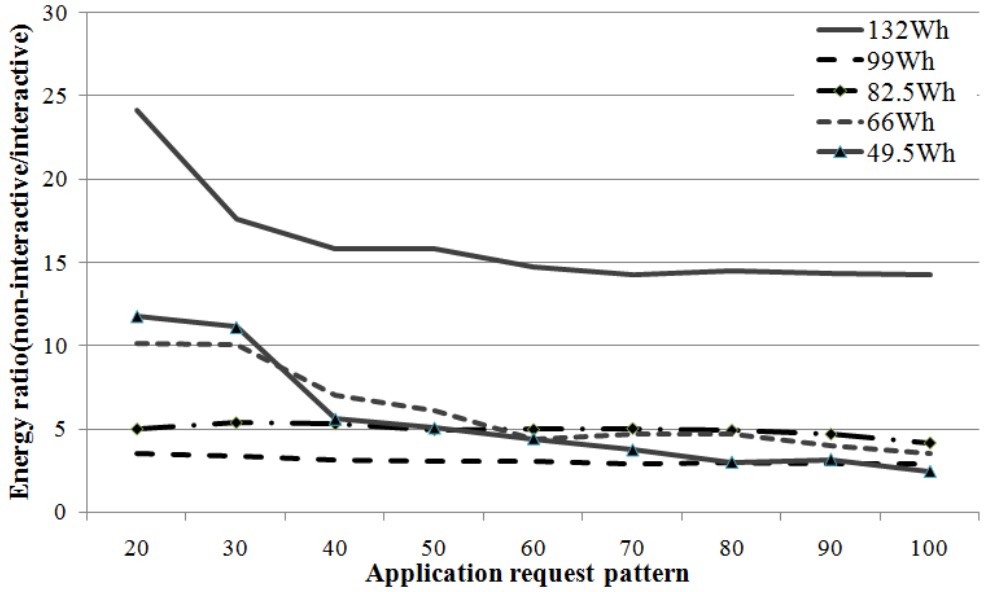

Figure 2.2: Impact of End Point Voltage variations on energy-efficiency

Our deployment uses Interstate DCM0055 Lead-Acid battery [2] with 55Ah capacity with Initial Battery Level (IBL) as 12V. The technical specification for this

battery mentions that there are five different End Point Voltage (EPV) levels : 9.6V, 10.2V, 10.5V, 10.8V, 11.1V for this battery. When the battery level reaches the EPV, it stops working until the recharge process starts. We then calculate the available/ target battery capacity for each of the discharge levels as: (IBL - EPV) * battery capacity. For example, for 9.5V EPL, the target battery capacity is: (12 - 9.6)* 55 = 132 Wh. It can be seen that the energy efficiency decreases as the application request ratio increases since in our case each request needs higher sampling rate (sampling every 1 minute). We can see that the proposed interactive approach is significantly more energy efficient than the non-interactive approach. This is because the later one allocates as much energy as it can in a fair manner, which leads to oversampling and wastage of energy. Figure. 2.2 also shows that higher discharging rate cannot use the total capacity, 55Ah because it draws high current. This situation is explained by Peukert's Equation [23].

Study of impact of harvesting energy variations on energy efficiency and accuracy

In this thesis we employ WCMA [47] algorithm for solar energy prediction for the interactive and non-interactive approaches. Solar energy availability varies significantly as a function of geographic location and season. To understand its impact on the performance of our approach in this study we consider solar energy variations during the winter (2012/01/4 - 2012/01/10) season at three different geographic locations in the United States, namely, 1) Necedah, Wisconsin (44.0262, -90.0737), 2) Austin, Texas (30.25, -97.75), and 3) Santa Barbara, CA (34.425833, -119.714167). We set default sampling interval to be 10 minutes and high request sampling interval to be 1 minute.

Our results indicate that the proposed interactive approaches are orders of magnitude more energy efficient than the non-interactive approach. In particular, Table 2.3 shows the the percentage of remaining battery level after one week of operation. In case of non-interactive approach it is just 0.8152% for the Wisconsin winter case. The

accuracy is expressed in terms of Root Mean Square Error (RMSE). Note that as shown Table 2.4 error of this approach is quite high (RMSE = 2.3057). We calculated that PF algorithm allocated energy to sample sensors every 17 seconds. This is counterintuitive because the approach samples data at very high frequency (default sampling rate for the interactive approach is every 10 minutes), but it still its error is higher than all the interactive approaches. A careful investigation shows that PF approach sets it Target Battery Level (TBL) to the battery end point voltage (9.6V). It will try to allocate maximum energy during each time slot in a fair manner. However, this includes the stored and harvesting energy. They assume an ideal solar prediction algorithm that always predicts the harvesting energy accurately. However, WCMA, the state-of-the-art solar energy prediction algorithm has a relative mean error of only 10%. When we plug-in this realistic solar energy prediction algorithm with the non-interactive algorithm, we see that during the one week of operation, the batter level goes below the end point level (target battery level) for approximately 11% of the slots. The system then stops operating thereby completely missing the sampling opportunities in those slots. In contrast, the interactive approaches avoid oversampling when not needed thereby saving the energy to allow higher sampling rates upon request. We also observe that the geographic locations did not have any major impact on the energy efficiency or the accuracy of the studied protocols.

Table 2.3: Percentage of remaining battery after 1 week

	WI	CA	TX
Non-interactive	0.815	0.981	1.795
Interactive-conservative	99.92	99.92	99.94
Interactive-aggressive	99.83	99.83	99.86
Interactive-hybrid	99.89	99.89	99.91

Table 2.4: Impact of harvesting energy variations on system accuracy

	WI	CA	TX
Non-interactive	2.305	2.498	2.132
Interactive-conservative	1.08	1.08	0.08
Interactive-aggressive	0.01	0.015	0.01
Interactive-hybrid	1.100	1.100	1.100

2.4.3 Discussion

The state-of-the-art energy allocation algorithm that takes into account current battery level and harvesting energy strives to fairly allocate as much energy as possible along the time dimension. This approach, by not considering application-context, leads to very high and uniform sampling rates. However, sampling the environment at fixed predefined intervals is neither possible (need to accommodate system failures) nor desirable (sampling rate might not capture an important event with desired fidelity). To that end, in this work we propose a novel interactive power management technique that adapts sampling rate as a function of both application-level context (e.g., user request) and system-level context (e.g harvesting energy availability). Our simulations use sensor data and system specifications (battery and solar panel specs, sensing and communication costs) for a real sensor network deployment. Existing interactive algorithm considers an ideal solar energy prediction algorithm that makes no prediction errors. However, by plugging-in a realistic solar energy prediction algorithm, we show that the existing approach often leads to draining the battery below the end point voltage thereby resulting in lower accuracy while spending high energy (due to high sampling rate). Our results show that the proposed approach saves significant amounts of energy compared by avoiding oversampling when application does not need it and uses this saved energy to support sampling at high rates to capture event with necessary fidelity when needed. The computational complexity of our approach is lower ($O(n)$) than the state-of-the-art non-interactive energy allocation algorithm ($O(n^2)$). This approach assume that each

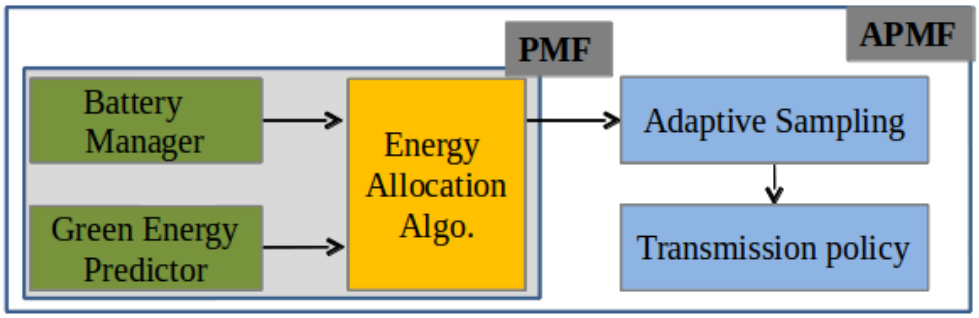

Figure 2.3: System architecture for PMF and APMF

sensing platform only run single power manager in order to achieve energy efficiency. However, in reality, different hardware components in sensor nodes run different power management strategies, so we still need to optimize their interactions. This is the topic of the next section.

2.5 Adaptive Power Management Framework

2.5.1 System architecture

Figure 2.3 describes the system architecture for traditional Power Management Framework (PMF) as well as the proposed Adaptive Power Management Framework (APMF). The traditional PMF includes three modules (ref. Figure 2.3) namely, battery manager, green energy predictor, and energy allocation algorithm.

The battery manager typically estimates the available battery capacity with one of the well-established capacity estimation algorithms [43][9][51], and provides this information to the energy allocation algorithm. Based on the information from battery manager and green energy predictor modules, the energy allocation algorithm calculates/allocates the optimal energy for each time slot (ref. Equation (4)). The state-of-art energy allocation algorithm, Progressive Filling (PF) [26], only considers battery level and harvested energy as optimization parameters, so the sampling rate is calculated only

based on energy information.

In contrast to the traditional PMF, the APMF (ref. Figure 2.3) consists of the following five modules: (1) battery manager, (2) green energy predictor, (3) Transmission Policy (TP), (4) Adaptive Sampling (AS) and (5) Energy allocation algorithm. The battery manager, green energy predictor, and the energy allocation modules are identical in PMF and APMF, while APMF also includes adaptive sampling (AS) and transmission policy (TP) modules to increase the energy efficiency by considering the application context. The AS module takes the energy budget calculated by the energy allocation module as its input and calculates the sampling rate at each decision epoch by capturing variations in phenomenon. AS and TP modules work in tandem. After AS determines sampling for the current decision epoch, the TP module uses this information to determine the optimal transmission policy so that both the transmission cost and the loss of data freshness (an application- specific metric) are minimized.

2.5.2 Adaptive Power Management Framework

Transmission policy module (TP)

TPs goal is to postpone data delivery in order to save energy while maximizing the data freshness. At each decision epoch, the TP module takes sampling rates, $R_{sampling}$, from the Adaptive Sampling (AS) module, and decides whether to transmit buffered packets or continue sampling them. We assume exponential distribution for inter arrival time of decision epoch as did work presented in [70]. The mean of inter- arrival time is μ. The monitoring system consists of m different sensors. Each measurement has a different lifetime, specified as the maximum delay of data arriving to the data center while still meeting application-specific data freshness requirements [14].

We leverage [70] to define the reward function (Equation (1)) as a product of

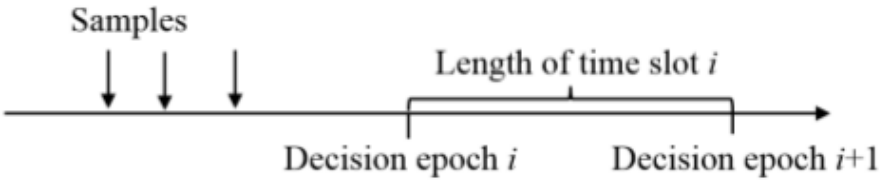

Figure 2.4: Illustration of decision process of TP

energy gain and data freshness loss. Note that energy gain is a monotonically increasing function, whereas the data freshness is a monotonically decreasing function. This reward function allows applications to easily tune the delay-data freshness tradeoff.

$$r_i(t_d) = e^{-\alpha_i \cdot t_d} \cdot g(t_d) = e^{-\frac{1}{Lifetime_i} \cdot t_d} \cdot g(t_d) \tag{2.5}$$

The $r_i(t_d)$ denotes the reward of sensor node i with delay t_d. The $g(t_d)$ is the energy gain achieved by decreasing transmission events by t_d, and $e^{-\alpha_i \cdot t_d}$ indicates the freshness loss with the delay t_d. Any function what has monotonicity and non-decreasing characteristic with delay can be used as $g(t_d)$. For example, in [70], author uses the number of buffered samples as energy gain.

We define α_i as a discount factor, which varies based different level of allowable latency of application. Note that the control-limit and MDP based approaches assume that all sensors have same discount factor, α, and therefore, they cannot support a wide-range of applications. In contrast, our approach allows an application to set the value of on per sensor basis. Figure 2.5 shows the effect of the varying the delay on reward for various discount factors. To show the effect of discount factor and delay, we set the initial value of reward (i.e. Equation (1)) when $t_d \in [0,1]$. The real time application should set α to have a large discount factor, so the TP transmits buffered packet with shorter delay to maximize rewards (e.g., when α is 1, t_d is 3 minutes the reward is 0.1). In other words, delay-tolerant applications can set α to have small delay, so TP can achieve maximum

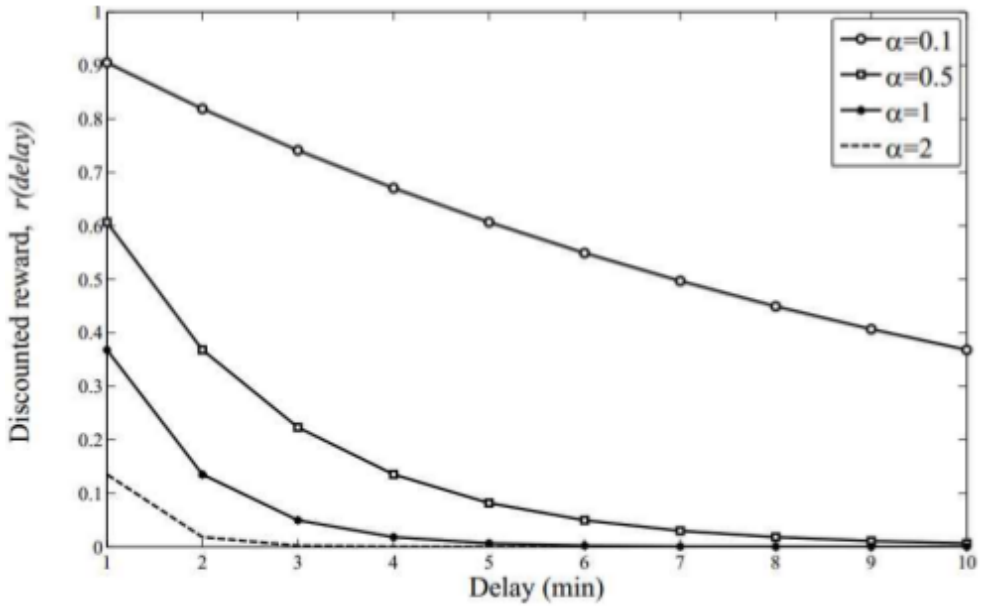

Figure 2.5: Different discount factors, α, result in different reward

reward with a certain delay (e.g., when is 0.1, t_d is 3 minutes the reward is 0.7). The proposed TP module extends the control-limit policy (ref. Algorithm 1) to account for different values of α. We now show that this policy is optimal (ref. Theorem 1).

Theorem 1. *We assume the system consists with m sensors, where each sensor has its own reward function (ref. Equation 2.5). This function has a finite value which monotonically decreases with delay (ref. Assumption 2.2 [70]). Each sensor i has* α_i *as its discount factor. The value of inter-arrival time of decision epoch is* δW*, so mean interval arrival time is* $\mu = 1/\delta W$*.* s_i^* *is the optimal number of samples that maximize reward for sensor i (ref. Equation 2.6). Then, the total number of optimal number of samples (*s^**) is the sum of* s_i^* *(ref. Equation 2.7). We set current sampling rate as* $R_{sampling}$*.*

$$s_i^* = \left\lceil \frac{E[Xe^{-\alpha_i \delta W}]}{1 - E[Xe^{-\alpha_i \delta W}]} + 1 \right\rceil = \left\lceil \frac{R_{sampling}\mu}{\alpha_i(\alpha_i + \mu)} + 1 \right\rceil \tag{2.6}$$

$$s^* = \sum_{i=1}^{m} s_i^* \tag{2.7}$$

Proof. We need to show that sum of optimal number of samples for each sensor is also optimal. This problem is same as weighted interval scheduling problem [35]. The scheduling problem can be formulated as follows. Assume a system with multiple requests, where each request has a start time and a finish time. The system can operate only one request at a time. A certain reward is returned when the system finishes a request. The goal of scheduling problem is to schedule requests such that the sum of rewards is maximized while ensuring that the scheduled requests do not overlap. It has been proved [35] that the scheduling problem has maximum value when all requests are mutually compatible which means they do not overlap. In our case, since all the sensors can be sampled independently, they are mutually compatible, so the maximum value of s^* in Equation 2.7 is indeed optimal. □

Algorithm 3 describes the operational details of TP. At every decision epoch, TP decides what to do with the buffered packets in the following manner. If number of buffered packets is larger than s^*, then TP decides to transmit all the buffered packets. If it is smaller than s^*, then it delays the transmission until the next decision epoch.

ALGORITHM 3: Transmission policy module

```
Input: R_sampling: calculated sampling rate at AS
Start at each decision epoch
BufLength = total number of buffered packets
for every sensors do
    s_i^* = Eq. 2.6 with R_sampling
end
 s^* = equation 2.7 with all s_i^*
if BufLength ≥ s^* then
    Transmit all buffered packets ;
end
Decide next decision epoch based on distributed of δW
```

The proposed Transmission Policy (TP) provides controls for tuning the energy-

delay and application data lifetime tradeoffs. For example, a delay-intolerant application might set the data lifetime to a large value to ensure that data freshness maximized at the expense of more frequent transmission. On the other hand, a delay-tolerant application might tolerate loss in data freshness by transmitting less frequently thereby saving more energy. The existing MDP [39] based approach does not allow applications to tune this tradeoff. The control-policy approach [70] considers tradeoff between energy gain and delay of buffered data because a system can save energy while decreasing the number of transmission events. To characterize the tradeoff they define reward as a function of energy gain and delay, and show that increasing delay increases reward until a certain point and after the point reward start decreasing.

Adaptive sampling module (AS)

The existing research on adaptive sampling assumes a discrete and bounded sampling rate and calculates optimal sampling rates over K time slot by applying task-allocation algorithm (i.e. Hungarian method [46]). The authors [34] use a linear programming approach with initial battery capacity before the first time slot as its constraint. They minimize distance between approximated sampled data and real data. Based on past measurements, sampling rates for K future time slots are predicted. However, this optimal adaptive sampling algorithm (OSA) requires high computational complexity $O(n^3)$. The AS module estimates the sampling rate using the allocated energy from the energy allocation module as its input. As we described in Section III.A, TP has a variable length decision epoch, thus, the AS calculates optimal sampling rate at every decision epoch. TP then uses this as its input. In Algorithm 4, The $E_{alloc}(i)$ is the allocated energy by the next decision epoch, and β denotes the factor reduction in energy by down sampling. An application sets β so that down sampling saves energy, while meeting its requirements in terms of data fidelity. The set of sampling rates of sensors are

$R = [r_1, ..., r_M]$, and r_1 and r_M as its lower and upper bounds respectively. AS calculates the quality of samples for all values of E and R and then chooses the sampling rate. Total Deviation (TD) [34] is a distance between predicted and real sample values. A larger value of TD implies lower quality of samples, so we calculate the largest TD (ref. calculateTD in Algorithm 2) for different sampling rates for a given energy limit. ITD in Algorithm 2 is a 2-dimensional matrix that saves the corresponding TD entry for each sampling rate and available energy combination. After calculating TDs for all elements in E and R, Algorithm 4 derives the sampling rate as the rate which has the smallest uncertainty & highest quality. Its computational complexity is $O(n^2)$, but with small set of E and R, it can be readily implemented on a resource-constrained smart phone.

ALGORITHM 4: Transmission policy module

```
1 Input:  E_alloc
2 R_sampling
3 for p = 1 to length(R) do
4 |   for q = 1 to length(E) do
5 |   |   I_TD(p,q) = calculateTD(R(p),E(q))
6 |   end
7 end
8 R_sampling = find the smallest value in I_TD
```

Green energy predictor

In case of periodic or partially periodic renewable energy such as solar energy, existing research has shown that the state of the art energy predictors such as Weather-Conditioned Moving Average, WCMA can be used to accurately predict the amount of harvesting energy [47]. Therefore, in this thesis we use WCMA algorithm for solar energy prediction.

2.5.3 Results

Our experimental results are based on two real-world data sets: (1) The deployment on a lake in Northern Wisconsin (ref. Figure 1) from Aug.8 to Oct. 9 2012. We use the measured temperature and humidity. (2) The second data set is from North temperate lake ecological study [5], with sampling from June 26 to Nov. 4 2008. For solar energy prediction we use data from USCRN [6] database for Necedah, Wisconsin location since it is the closest location to our deployment. Table 1 is the power specification of a subset of sensors and the smartphone from our deployment (i.e. dataset 1) [64]. The smartphone acquires data from all the sensors and then transmits it to a data center hosted on Amazon-EC2 cloud platform. The phone is equipped with Wi-Fi and cellular radios. In our deployment the phone transmitted data over cellular (3G) network. In Table 2.5, suspend means that the application processor is idle, while the communications processor performs a low level of activity, as it must remain connected to the network to be able to receive messages, etc. Idle state means that all components are in a low-power state without application operation.

Table 2.5: Power consumption specification for our deployment

	Current (mA)	Power (mW)	Time (s)
Wind speed (m/s)	2.5	30	5
Wind direction ($\circ$)	2.5	30	5
Barometric pressure (hPa)	0.8	9.6	5
Air temperature ($\circ C$)	0.8	9.6	5
Relative humidity (%)	0.8	9.6	5
Templine	31.7	380	5
Andriod phone - Suspend	2.4	7.2	-
Andriod phone - Idle	14.5	43.5	60
Andriod phone - Tx + Rx	53.9	161.9	1

We assume mean decision epoch interval of 13 minutes, same as the simulation setup in [70]. Thus, we restrict maximum sampling interval to 13 minutes. We model data lifetime in terms of a sensors sampling interval with three different values: 1, 5, and

10. In the remainder of this thesis, we use x1, x5, and x10 to represent these lifetime factors. We define data freshness loss as the percentage of data which stays longer than its lifetime in the local buffer. For example, when the sampling interval is 1 minute and lifetime factor is set to 10, the sample does not lose its data freshness for 10 minutes. Our proposed power management framework is implemented in Matlab.

Performance of AS with real world data sets

We compare AS with optimal adaptive sampling algorithm (OSA) [34] in terms of calculated average sampling rate over experimental period. The difference between OSA and AS is length of time slot: the OSA considers fixed length of time slot, and AS determines sampling rate at each decision epoch. In addition, AS discretize allowable energy (i.e. vector E in Algorithm 2) into a set of finite size. The Table 2.6 shows that the difference between OSA and AS is less than 1 minute. AS samples slightly at higher rates than OSA, while significantly reducing the computational overhead.

Table 2.6: Average sampling rate with real data sets

	Temperature	Humidity	Chloride
AS	4.5 ± 0.6	4.5 ± 0.6	6 ± 3.6
OSA	5.17 ± 1.6	5 ± 1.4	6.7 ± 6.4

Study of energy-delay tradeoff for various lifetime factors

In this study, we describe the impact of varying lifetime factors on relative power consumption (ref. Table 2.7) and data freshness loss (ref. Table 4). To check the performance of TP, we consider the scenario where each of these six sensors uniformly selects its sampling interval between 0 to a predefined upper bound during the initialization phase. We vary upper among 3, 5, 10, 13 minutes. In Table 3, the relative power consumption decreases with the increase in the lifetime factor. This is because the system can buffer

the samples for longer duration as the lifetime factor increases thereby reducing the number of transmissions (ref. Equation (2.5)). For example, TP with lifetime factors 1 (i.e. TP (x1)) consumes 1.25 times and 1.4 times more energy than TP (x5) and TP (x10) policies respectively. In Table 4, TP (x10) policy has between 1.3 to 3.3 times more loss data freshness than TP (x5) policy.

Table 2.7: Impact of variation of lifetime factor on relative power consumption (%) for different sampling upper bounds

	3min	5min	10min	13min
TP (x1)	97.7	96.3	88	78.7
TP (x5)	93.2	77.8	60.7	48.5
TP (x10)	89	68	51.8	44.8

Table 2.8: Impacts of different lifetime factor to freshness loss (%) for different sampling upper bound

	3min	5min	10min	13min
TP (x5)	1.2	2	2.6	6.4
TP (x10)	2.3	6.5	6.5	8.6

Study of energy-delay tradeoff

In this study, we describe the impact of various transmission policies on relative power consumption (ref. Table 2.9) and data freshness loss (ref. Table 2.10). The energy consumption of TP with 5x lifetime is consistently 20than the MDP based approach (ref. Table 2.9). However, because TP (x5) policy uses large lifetime factor, its data freshness (ref. Table 2.10) loss is higher (between 2% to 7%) as compared to the MDP based approach.

Study of PF, AS and TP with real data set

In this section, we compare APMF with PMF [26] (ref. In Table 2.11). APMF results in 27% to 72% energy savings in comparison with PMF. We also selectively

Table 2.9: Impacts of different transmission policies on power consumption (%) for different sampling intervals

	5min	10min	15min	20min
MDP [39]	98.6	79.6	6138	51.2
TP (x5)	77.8	60.7	48.5	41.3

Table 2.10: Impacts of different transmission policies to freshness loss (%) for different sampling interval

	5min	10min	15min	20min
MDP [39]	0	0	0	1
TP (x5)	2	2.6	6.4	7.9

disable TP and AS modules to evaluate their impact on overall performance of APMF. We can see the TP dominates energy savings because transmission cost is 25x higher than the sensing cost. Table 2.12 shows that APMF has a negligible data estimation error as compared to PMF. Therefore, APMF results in higher energy savings with no loss in accuracy. The low level of estimation error is because measured temperature and humidity are slowly varying (i.e. at most $4.2^{\circ}C$ and 4.5% variance respectively per day). We use another set of measurement from North template lake ecological study [5], which involves sampling from $6/26/2008$ to $11/4/2008$. This data includes a limnological variable that varies more quickly than temperature. From Table 2.13, we can observe that APMF consumes 62% less energy than PMF.

Table 2.11: Relative energy consumption in comparison with data set 1

vs. PMFTP (x1)	Humidity	Temperature
APMF	27%	32%
APMF w/o TP	62%	72%
APMF w/o AS (5min)	48%	53%
APMF w/o AS (5min)	16%	42%

Table 2.12: Estimation error in comparison with data set 1

vs. PMFTP (x1)	Humidity	Temperature
APMF	0.015%	0.005%
APMF w/o TP	0.012%	0.003%
APMF w/o AS (5min)	0.009%	0.003%
APMF w/o AS (5min)	0.013%	0.005%

Table 2.13: Relative energy consumption comparison with data set 2

vs. PMFTP (x1)	Energy consumption (%)
APMF	62 %
APMF w/o TP	88%
APMF w/o AS (5min)	74%
APMF w/o AS (5min)	69%

2.6 Discussion

In this thesis, we have Advanced Power Management Framework (APMF) that adapts sampling & transmission rates based on battery capacity level, harvesting energy amount and application-context (characteristics of gathered data). The adaptive sampling and transmission policy manager modules of APMF have low complexity and are suitable for resource-constrained devices. APMF provides applications a finer control over delay-energy tradeoff. We evaluated the performance of our proposed approach using dataset from two real-world deployments. Our results show that APMF saves 20% to 60% of energy consumption by avoiding oversampling. So far, we have studied power management designs that operate in different layers. In the next section, we study the user based data aggregation approaches for WSNs.

Chapter 2, in part, is a reprint of the material as it appears in ”Leaveraging application constext for efficient sensing”, by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE ISSNIP 2014. The dissertation/thesis author was the primary investigator and author of this thesis.

Chapter 2, in part, is a reprint of the material it appears in ”An Interactive

Context-aware Power Management Technique for Optimizing Sensor Network Lifetime", by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, SENSORNETS 2016. The dissertation/thesis author was the primary investigator and author of this thesis.

Chapter 3

Adaptive transmission manager

3.1 Introduction

A new generation of Wireless Sensor Networks (WSNs) envisions commodity sensing & actuation infrastructure to provide services. These WSNs form a critical and general interface between physical and digital worlds. They convert physical qualities into measurements which can be used for a wide-ranging spectrum of applications. The impact of such networks is significant: WSNs will no longer be specialized networks which are running a single application and supporting a limited set of users. Rather, multiple applications deployed by multiple organizations will share a deployed sensor network. The running applications will have diverse delay and accuracy requirements.

Fig. 3.1 describes the operation of a sensing platform which collects data from heterogeneous measurement sources in WSNs. Each sensing platform runs multiple applications that collect raw data from the corresponding sensors. Applications process the raw data to generate measurements, and send these measurements along with application-specific delay requirements to the data buffer in the transmission manager. In addition to generating its own data, the sensing platform also receives packets forwarded from its

neighboring nodes as WSNs may use multi-hop wireless links to forward the data to the sink node. Thus, the data buffer in the transmission manager has both the measurements generated by the node itself as well as the ones received from its neighbors. The job of the transmission manager is to determine the optimal single hop transmission instance of buffered measurements, based on the end-to-end delay requirements and distance to the sink along a routing path. Recent publications that look at delay guarantees in multi-hop WSNs [62][61] decompose the end-to-end delay problem into a set of single-hop delay subproblems. However, they don't determine a specific solution for how each node in WSNs will obtain a single-hop delay requirement. They show that end-to-end delay guarantee problem requires information about the buffer, the channel, and the system conditions of ancestor nodes in the routing paths such that the problem is NP-hard.

The key challenge of transmission manager is to minimize the number of messages that expire prior to reaching their destination while minimizing the energy consumption. We first propose an optimal transmission manager using the optimal stopping theorem based on Markov Decision Process model (MDP) in order to find out optimal transmission instance between two nodes. Then, we propose a distributed transmission manager that tunes the optimal transmission manager to operate in multi-hop WSNs. The optimal and distributed transmission managers work in tandem (ref. Fig. 3.4), and run on each node as shown in Fig. 3.1. In this figure, transmission managers in green box includes those approaches. They determine the transmission time for every buffered measurement. Measurements expire if they cannot arrive to a sink by application specific time constraint.

We implement both transmission mangers in ns3 simulator [45] to compare with the other state of the art transmission managers [55][12][70]. We evaluate the energy consumption and the number of expired measurements for three different network topologies: i) single hop, ii) linear and iii) grid. A single hop case is typical for small

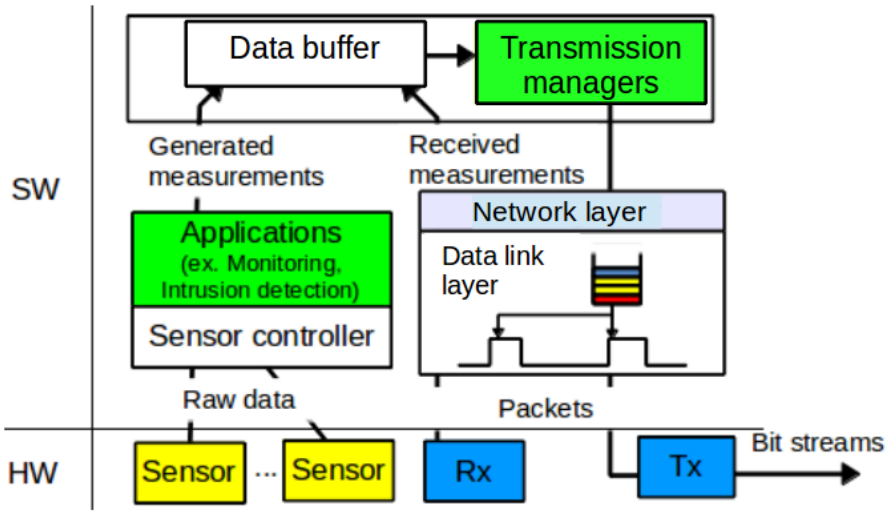

Figure 3.1: Applications generate measurements which have application-specific delay requirements and send them to the data buffer in the transmission manager. Transmission manager determines the transmission time of the buffered measurements.

scale wireless sensor networks such as Body Area Networks (BANs) [8]. Linear topology is commonly used in buoy deployments in order to monitor lake conditions such as temperature, dissolved oxygen, conductivity and pH/ORP. Grid topology has been shown to be effective for city-wide deployments such as air quality monitoring in downtown San Diego for projects such as CitiSense [44]. In all three cases, the optimal and distributed transmission managers work in tandem, and consume on average 148.3% less energy than the state of the art approaches while having on average 14.1% of measurements that expire.

In the remaining part of this work, we first summarize the related work on transmission managers in Section 3.2. Section 3.3.1 formulates our problem by using Markov Decision Process (MDP) model and proves the existence of the optimal transmission time. Details of proposed optimal transmission manager are described in Section 3.3.2. In Section 3.3.4, we describe how our proposed transmission manager operates in multi-hop WSNs. Section 3.4 discusses the experimental setup. Network simulation results with single hop, linear and grid topologies are summarized in Section 3.5

3.2 Related work

There are a few publications that focus on designing transmission manager that minimizes the energy consumption while ensuring a low number of expired measurements. They decrease energy consumption by decreasing the communications with an embedded buffer. They estimate transmission time based on different factors such as local application delay constraints, distance to the sink node and node's available energy.

The periodic per hop approach [55] waits for a pre-defined periodic time interval before forwarding received data to its parent node. The cascade time-out protocol, presented in the same paper, buffer all generated and received measurements in its buffer and transmit at calculated transmission time. The time is function of the hop distance to the sink node and sampling interval as described in the following: transmission time $= 2 *$ $\{$sampling interval $-$ (single hop delay $*$ hop distance to the sink)$\}$. In [55], authors set the single hop delay as 0.03sec. Transmission managers that consider application based delay constraints are proposed by [12][70][11]. Their transmission managers determine the optimal transmission time of buffered measurements based on the characteristics of a running application. The selective-forwarding approaches [12][11] maximize reward of all measurements that are generated in the wireless network to achieve the energy reduction while minimizing the number of expired measurements. They define the reward as a function of delay-sensitivity of measurements and a probability of arriving at the sink node. Selective-forwarding approaches assume that all nodes in the wireless network run a single delay sensitive application. In contrast, the delayed forwarding approach [70] uses a delay-sensitivity factor (i.e. α) to represent the trade-off between energy gain and expiration rate of measurements. For example, a large α value requires a node to schedule a transmission within a short amount of time. While this decreases the probability of measurements timing out, it also increases energy consumption. The choice of α is

arbitrarily specified by the author.

Prior works do not consider different delay requirements of heterogeneous applications in WSNs. This is the limitation that our proposed transmission managers seek to resolve. We first propose the optimal transmission manager that determines optimal transmission delay of buffered measurements on a single sensor node in Section 3.3. Then, we extend the transmission to operates in multi-hop WSNs and explain in Section 3.3.4. Both transmission managers work in tandem, and the results show that they consume on average 148.3% less energy than the state of the art approaches while increasing the number of measurements that expire by 14.1% on average.

3.3 Optimal transmission manager

The system model is shown in Fig. 3.1. The distributed and optimal transmission managers operate between running applications and the network layer on each sensing platform. All generated or received measurements are stored in the data buffer. They include two types of information: i) raw data and ii) end-to-end time constraints. Raw data is coming from embedded sensors. Applications assign end-to-end time constraints based on their own delay requirements for the generated measurements [62][61]. Distributed transmission manager determines upper bounds of transmission instance, N, from the end-to-end time constraints as described in Section 3.3.4. Then, optimal transmission manager (OptTM) decides the optimal transmission time of the measurements based on the time constraints of buffered measurements in order to minimize the number of measurements that expire and lower the energy consumption.

In order to satisfy these goals, OptTM evaluates actions at regular time intervals called *decision epochs*, as shown in Fig. 3.2. At each decision epoch, OptTM determines the optimal transmission time. The transmission manager's MDP model has the following

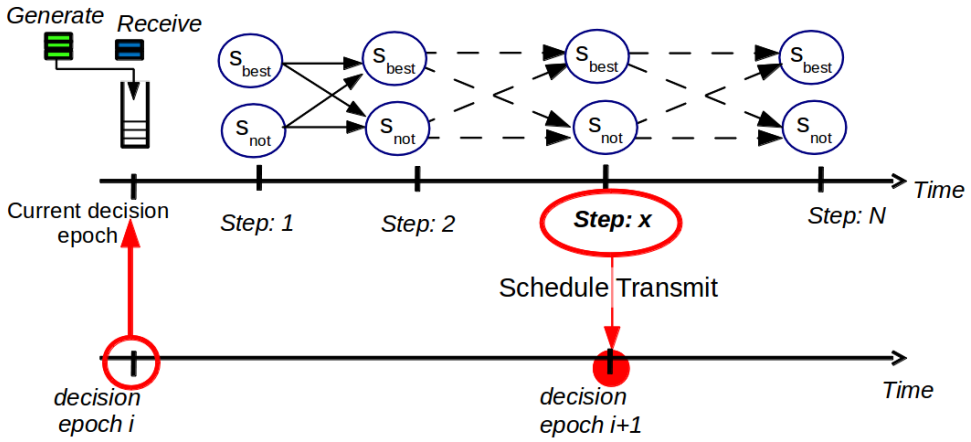

Figure 3.2: At every decision epoch, transmission manager uses Markov Decision Process model and find out optimal transmission instance of buffered measurements. Black arrows represent the transitions between the states

components:

- States, $S = \{s_1, ..., s_{S_{max}}\}$, represent buffer characteristics. For example, the state variables represent the current number of measurements in the buffer. OptTM buffer delays separated by Δ_{step} time interval and represented by $\{Step_1, ..., Step_{max}\}$. At each decision epoch, OptTM checks buffer characteristics and decides the optimal step (time) for transmission. Smaller Δ_{step} increases the granularity of buffer delays, but it also constantly increases computation complexity. In this work, we set Δ_{step} at 1 sec which is a reasonable value for the applications we use in this work. In the current decision epoch, as shown in Fig. 3.2, OptTM looks over all possible transmission times represented by steps 1 to N and chooses to transmit measurements at step k.

- Actions, $A = \{a_1, ..., a_{A_{max}}\}$, give the number of measurements that the transmission manager transmits at a particular step. A_{max} is the maximum number of actions.

- Transition probability defines the likelihood of the next state based on the current state and the current action. At step k, $P_k(s_{k+1}|s_k, a_m)$ is the probability of the next

state s_{k+1} when the current state is s_k and the action is a_m. For example, at step k, the chosen action, a_m maybe to transmit all measurements in the buffer, thus transitioning the system into state s_{k+1} with an empty buffer.

- Reward, $R_k(s_k, a_m)$, is calculated for all possible actions, $a_m \in A$, at each state $s_k \in S$. OptTM selects the action, a_k^*, that returns the largest reward, R_k^*, as described in Eq. (3.1) and Eq. (3.2).

$$a_k^* = arg \max_{a_m \in A} \left\{ R_k(s_k, a_m) + \sum_{s \in S} P_k(s_{k+1} = s | s_k, a_m) R_{k+1}^*(s_{k+1}) \right\} \tag{3.1}$$

$$R_k^*(s_k) = \max_{a_m \in A} \left\{ R_k(s_k, a_m) + \sum_{s \in S} P_k(s_{k+1} = s | s_k, a_m) R_{k+1}^*(s_{k+1}) \right\} \tag{3.2}$$

3.3.1 Criteria for optimality

Definition 1. *Let S and A be partially ordered sets and $f(s,a)$ a real-valued function on $S \times A$. The function $f(s,a)$ is super-additive when $f(s',a') - f(s,a') \geq f(s',a) - f(s,a)$ with $s' > s$ in S and $a' > a$ in A.*

Theorem 2. *There exists an optimal structured transmission rule if the formulated problem satisfies the following conditions [49]:*

1. *$R_k(s_k, a_m)$ is nondecreasing in s_k for all $a_m \in A$ and $k \in \{1, .., N-1\}$.*
2. *Cumulative transition probability, $q_k(j|s_k, a_m) = \sum_{s_{k+1}=j}^{s_{Smax}} P_k(s_{k+1}|s_k, a_m)$, is nondecreasing in s_k for all $k \in \{1, ..., N-1\}$ and $a_m \in A$, $k \geq 0$, $\forall j \in S$.*
3. *$R_k(s_k, a_m)$ is super-additive function on $S \times A$.*

4. *$q(j|s_k, a_m)$ is super-additive function on $S \times A$.*

5. *$R_N(s_k, a_m)$ is nondecreasing in s_k.*

Then, the optimal value is nondecreasing in state $s \leq s' \in S$: $R_k^*(s) \leq R_k^*(s')$.

Given Theorem 2, a reward function, $R_k(s,a)$, in OptTM is super-additive and nondecreasing in $s \in S$ for all $a \in A$ and $k \in \{1,..,N\}$. The nondecreasing condition of $R_k(s,a)$ in s becomes true when the reward function satisfies the following relationship with $s' \geq s \in S$ and $a \in A$: $R_k(s',a) - R_k(s,a) \geq 0$. The reward function becomes super additive function when it satisfies the following relationship with $s' \geq s \in S$ and $a' \geq a \in A$: $R_k(s',a') - R_k(s,a') \geq R_k(s',a) - R_k(s,a)$. We first derive a sufficient condition to satisfy the above two relationships with an assumption that $R_k(s,a) = f(s) \cdot r_k(a)$ (that is $f(s)$ is nondecreasing function with state s and $r_k(a)$ is a reward with an action a). With this assumption, we can derive a sufficient condition for super-additivity of $R_k(s,a)$ as described in Eq. (3.3). The derived sufficient condition that satisfies the relationship, $R_k(s',a') - R_k(s,a') \geq R_k(s',a) - R_k(s,a)$, is $r_k(a') \geq r_k(a)$ with nondecreasing $f(s)$. In addition, the condition also guarantees nondecreasing characteristic of $R_k(s,a)$ because $f(s') \geq f(s)$.

$$\begin{aligned} & R_k(s',a') - R_k(s,a') \geq R_k(s',a) - R_k(s,a) \\ & = r_k(a')\{f(s') - f(s)\} \geq c_k(a)\{f(s') - f(s)\} \\ & = r_k(a') \geq r_k(a) \end{aligned} \tag{3.3}$$

The cumulative transition probability is the nondecreasing and has super additive property when it satisfies the following two relationships with $s' \geq s \in S$ and $a' \geq a \in A$: i) $q_k(j|s',a) - q_k(j|s,a) \geq 0$, ii) $q_k(j|s',a') - q_k(j|s,a') \geq q_k(j|s',a) - q_k(j|s,a)$. We define X_s as the difference between the two states and prove the nondecreasing property

of cumulative transition probability as described in Eq. (3.4).

$$\begin{aligned} & q_k(j|s',a) - q_k(j|s,a) \\ &= \sum_{s_{k+1}=j}^{\infty} P_k(X_s = s_{k+1} - s') - P_k(X_s = s_{k+1} - s) \geq 0 \end{aligned} \tag{3.4}$$

When $j \geq s'$, $\sum_{s_{k+1}=j}^{\infty} P_k(X_s = s_{k+1} - s') - P_k(X_s = s_{k+1} - s) = \sum_{n=1}^{s'-s} P_k(n)$. When $s \leq j < s'$, $\sum_{s_{k+1}=j}^{\infty} P_k(X_s = s_{k+1} - s') - P_k(X_s = s_{k+1} - s) = \sum_{n=1}^{j-s} P_k(n)$ since $P_k(X_s < 0) = 0$. When $0 \leq j < s$, $\sum_{s_{k+1}=j}^{\infty} P_k(X_s = s_{k+1} - s') - P_k(X_s = s_{k+1} - s) = \sum_{n=1}^{j-s} P_k(n) = 0$. In all three cases, $q_k(j|s',a) - q_k(j|s,a)$ in Eq. (3.4) is always positive. This proves that the cumulative transition probability has a nondecreasing property. Without the loss of generality, we see that it is also super-additive.

With these properties, our model is guaranteed to give the optimal transmission time. In the next subsection we discuss how we define all of the parameters of our model in order to guarantee the optimality.

3.3.2 Optimal transmission manager implementation

At every decision epoch, OptTM checks the buffer state for each time step and determines when to transmit the currently buffered measurements. Thus, we define two actions corresponding to transmitting no messages if the optimality criteria are not met (a_0) and transmitting all messages if they are met (a_{all}).

Reward at step k, R_k, is the sum of all buffered measurements' remaining lifetime (the first term in Eq. (3.5)) with the expected rewards for future measurement arrivals (the second term in Eq. (3.5)). Previous works [12] and [70] use exponential function as their reward to emphasize the effect of decreasing message lifetime with delay. However, exponential function decreases the delay too quickly in realistic applications, so we

use a linear function of remaining lifetime as shown in Eq. (3.5) and Eq. (3.6). $T_{i,k}^{life}$ is the remaining lifetime of i^{th} measurement in the buffer at k^{th} step as shown in Eq. (3.6). It is the time difference between a time constraint of that measurement (T_i^{time}) and step k representing the time delay that all measurements incur before transmission. Future measurements are generated from different applications and arrive at different times. Thus, we calculate the expected future reward in terms of the number of running applications, data delay and the number of arrivals. The number of applications running in WSNs is defined by *nApp*. The expected delay by k^{th} step is β and γ is the number of measurements arrived. $P(\gamma,\alpha)$ is the probability of γ arrivals of application α. We model this probability using a Poisson distribution as described in Eq. (3.7). λ_α is arrival rate of measurements generated from an application α. $E\{T^{time}\}$ is the expected time constraint of future arrival measurements.

$$R_k(s_k) = \sum_i T_{i,k}^{life} + \sum_{\alpha=1}^{nApp} \sum_{\beta=1}^{k-1} \sum_{\gamma=0}^{\infty} \gamma \cdot \{E(T^{time}) - \beta\} \cdot P(\gamma,\alpha) \tag{3.5}$$

$$T_{i,k}^{life} = max\left\{T_i^{time} - k, 0\right\} \tag{3.6}$$

$$P(\gamma,\alpha) = \frac{e^{-\lambda_\alpha}(\lambda_\alpha)^\gamma}{\gamma!} \tag{3.7}$$

We use optimal stopping theorem [49] to implement OptTM. We define only two buffer states, $S = \{s_{best}, s_{not}\}$. When the reward at the current step is the largest seen so far, the system is in state s_{best}, otherwise, it is in state s_{not}. Transitions between the two states are denoted by the transition probabilities. Transition probabilities for the uncontrolled system are independent of the system state [49], so $P(s_{k+1}|s_k, a_m)$ becomes $P(s_{k+1}|s_k)$. More specifically, $P(s_{best}|s_k = s_{not}) = P(s_{best}|s_k = s_{best}) = P(s_{best})$ and $P(s_{not}|s_k = s_{not}) = P(s_{not}|s_k = s_{best}) = P(s_{not})$.

Reward at step k, $R(s_k)$, is the sum of all buffered measurements' remaining lifetime (the first term in Eq. (3.8)) with the expected rewards for future measurement arrivals (the second term in Eq. (3.8)). The arrival rate of measurements λ can be measuring by the number of arrivals between two consecutive decision epochs.

$$R_k(s_{best}, a_k) = \sum_i T_{i,k}^{life} + \sum_{\beta=1}^{k-1} \{E(T^{time}) - \beta\} \cdot \lambda \tag{3.8}$$

We can now derive the optimality criteria given our simpler model by using backward iteration approach [49]. This approach starts from final step N and checks the optimal reward at each step going back to the beginning (ref. Eq. (3.2)). Because transition probability only exists when OptTM continues buffering (action a_0), the optimal reward at step k, $k < N$, can be simplified as described in Eq. (3.9). We set $R_N^*(s_{best}) = 1$ and $R_N^*(s_{not}) = 0$ due to the nondecreasing property of the reward function.

$$R_k^*(s_k) = \max\left\{R_k(s_k, a_k = a_{all}), E[R_{k+1}^*(s_{k+1})]\right\} \tag{3.9}$$

The $R_k(s_k, a_m = a_{all})$, $k \leq N$ is the reward of buffered measurements at step k when OptTM transmits all buffered measurements. $R_{k+1}^*(s_{k+1})$ is the optimal reward at the next step, $k+1$.The expected reward for the next state, $E[R_{k+1}^*(s_{k+1})]$ can be calculated as described in Eq. (3.10).

$$E[R_{k+1}^*(s_{k+1})] = \sum_{s \in \{s_{best}, s_{not}\}} R_{k+1}^*(s) P(s_{k+1} = s) \tag{3.10}$$

We next calculate the optimal reward at each step. We denote $R_k^*(s_{best})$ as the optimal reward at step k when the current state is s_{best}. The $R_k^*(s_{not})$ is the optimal reward at step k when the current state is s_{not}. For $k < N$, $R_k^*(s_{best})$ and $R_k^*(s_{not})$ can be calculated as described below.

$$\begin{aligned} R_k^*(s_{best}) &= \max\left\{R_k(s_{best}), E\left\{R_{k+1}^*(s_{k+1})\right\}\right\} \\ &= \max\{R_k(s_{best}), P(s_{best})\cdot R_{k+1}^*(s_{best}) \\ &+ P(s_{not})\cdot R_{k+1}^*(s_{not})\} \end{aligned} \tag{3.11}$$

$$\begin{aligned} R_k^*(s_{not}) &= \max\{0, P(s_{best})\cdot R_{k+1}^*(s_{best}) \\ &+ P(s_{not})\cdot R_{k+1}^*(s_{not})\} \end{aligned} \tag{3.12}$$

Because the expected reward is always positive, we can simplify Eq. (3.11) and Eq. (3.12) to Eq. (3.13).

$$R_k^*(s_{best}) = \max\left\{R_k(s_{best}), E\left\{R_{k+1}^*(s_{k+1})\right\} = R_k^*(s_{not})\right\} \tag{3.13}$$

Solution of Eq. (3.13) yields the optimality criteria of the transmission manager. In state s_{best}, when $R_k(s_{best}) \geq E\left\{R_{k+1}^*(s_{k+1})\right\}$ the optimal action is to transmit all the currently buffered measurements. When $R_k(s_{best}) < E\left\{R_{k+1}^*(s_{k+1})\right\}$, the optimal action is to continue buffering without transmitting measurements, a_0. The optimal transmission instance that maximizes the reward while minimizing the number of measurements that expire is shown in Eq. (3.14).

$$N^* = \arg\min_k \{R_k(s_{best}) \geq E\left\{R_{k+1}^*(s_{k+1})\right\}\} \tag{3.14}$$

We can select any probability as transition probability that satisfy condition in Eq. (3.4). Thus, we select $P(s_{best}) = \frac{1}{k+1}$ which means the reward at next step $k+1$ is the largest seen so far. As s_{best} and s_{not} are mutually exclusive events, $P(s_{not}) = 1 - P(s_{best})$. Because reward function does not depend on the current state, Eq. (3.14) becomes Eq.

(3.15) with selected transition probability. This means that Eq. (3.15) is the optimal solution.

$$N^* = \arg\min_k \{R_k \geq R^*_{k+1}\} \tag{3.15}$$

ALGORITHM 5: Optimal single hop transmission manager (OptTM)

```
Input 1: time constraints set T
Input 2: D_sink: distance in number of hops from current node to sink
N = input from DistTM
A = zeros(1,N), this is action vector
R = zeros(1,N), this is reward vector
for k=1 to N do
    Calculate R(k) with Eq. (3.8)
end
// Determine optimal action
for k=N-1 to 1 do
    if max R(1:k) ≠ R(k) then
        //Current state is s_not
        A(k) ← a_0 // Continue ;
    else
        //Current state is s_best
        Calculate E[R*_{k+1}] with Eq. (3.10) and apply transmission
        probabilities
        if R(k) ≥ R*_{k+1} then
            A(k) ← a_all // transmit all currently buffered measurements;
        else
            A(k) ← a_0 ;
        end
    end
end
if ∃k, A(k) = a_all then
    Output: optimal transmission instance = min{k ≥ 1 : A(k) = a_all}
end
```

3.3.3 OptTM algorithm for a single hop WSNs

Algorithm 5 describes how OptTM determines the optimal transmission time for buffered measurements at each decision epoch in single hop WSNs. Inputs to the algorithm are buffer with time constraints of all measurements (T), upper bound of delay (N) which is calculated from distributed transmission manager, distance in number

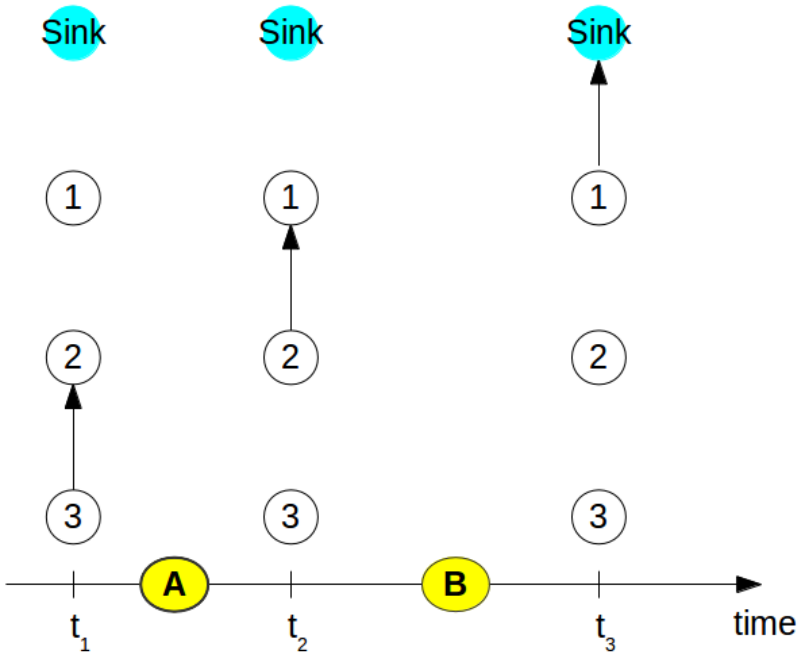

Figure 3.3: Additional delays at relay nodes (A+B)

of hops from current node to sink (D_{sink}), action (A) and reward (R) vectors. OptTM calculates the reward (ref. Eq. (3.8)) for each potential transmission step in lines 6:8. When the current reward is not the largest one calculated at previous steps (lines 11:13), OptTM continues buffering without transmitting measurements (ref. Eq. (3.12). Else OptTM compares the current reward with the expected reward (line 17). It transmits all the measurements if the current reward is greater than equal to the expected reward. Lastly, OptTM selects the soonest step for which Eq. (3.14) is satisfied. This is then defined as the optimal transmit time (line 25). The computation and space complexity of this algorithm is $O(N)$.

3.3.4 Distributed transmission manager

Applications assign end-to-end time constraints based on their own delay requirements for the generated measurements. Measurements coming from different applications may have different time constraints. Distributed transmission manager (DistTM) calcu-

ALGORITHM 6: Distributed multi-hop transmission manager (DistTM)

```
Input 1: buffer B_in
Input 2: D_sink: distance in number of hops from current node to sink
length(B_in): the number of measurements in B_in
T_i^end: end-to-end time constraint of i^th measurement in B_in
// Calculate upper bound of transmission delay
Calculate x with Eq. (3.17)
Calculate N with Eq. (3.16)
// Calculate optimal transmission instance
Optimal tx_instance ← Algorithm 5 with input N
// Check measurements that expire
for k=1 to length(B_in) do
    B_in(k) = B_in(k) − tx_instance ;
    if B_in(k) ≤ 0 then
        delete B_in(k) from B_in
    end
end
Output: Schedule transmission of measurements in B_in at tx_instance
```

lates N and provides the value to OptTM at each decision epoch. Then, as described in Algorithm 5, OptTM running on a relay node uses time constraints of buffered measurements (generated & received) to choose when to transmit between 1 and N. Unlike other state of the art approaches, OptTM is non-synchronized approach. This yields additional delays at relay nodes as described in Figure 3.3. Node 3 transmits a measurement to Node 2 at t_1. Node 2 relays the measurement to Node 1 at t_2. Node 1, then, relays the received measurements to the sink at t_3. A measurement generated from Node 3 can be expired if the additional delays at relay nodes (A+B) is larger than its time constraint. A measurement of Node 2 has additional delay at Node 1, but a measurement of Node 1 do not suffer additional delay because Node 1 can directly connect to the sink. This implies that even though all nodes have same measurements (which have same time constraints), Node 3 should have the shortest transmission interval in order to avoid measurement expirations at relay nodes. This means that DisTM should consider distance to the sink when it calculates the final step N.

We first check the worst case delay at relay nodes in multi-hop WSNs. Suppose all nodes have same measurements (with time constraint, T_{time}) in their buffer, and they

have same transmission interval x sec. Because each node transmits after x sec, the the worst case delay is a function of distance to the sink and the transmission interval, $D_{sink} \cdot x$. This means that when transmission interval x satisfies the following condition, $D_{sink} \cdot x \leq T^{time} = x \leq \frac{T^{time}}{D_{sink}}$, all measurements can arrive to the sink without expiring. The best case delay is x because of no additional delays at relay nodes. Thus, measurements do not expire if $x \leq T^{time}$. Average delay at relay nodes in multi-hop WSNs is a function of the worst and best case delays as described in Eq. (3.16). The right hand side of the Eq. (3.16) is bounded average delay at relay nodes with x sec transmission interval.

$$\frac{\frac{T^{time}}{D_{sink}} + x}{2} \geq \frac{\frac{x}{D_{sink}} + x}{2} = \frac{(D_{sink} + 1) \cdot x}{2 \cdot D_{sink}} \tag{3.16}$$

The goal of DistTM is to find upper bound of transmission instance (N) by estimating expected delay at relay nodes with D_{sink}. Thus, DistTM uses expected time constraints of measurements, $E(T^{time})$, and calculates x as described in Eq. (3.17). We use similar method as [55] because they show the best performance when all nodes are synchronized. Single hop delay is the sum of maximum value of propagation delay in wireless channel and staggering delay in MAC layer [55]. OptTM transmits measurements to MAC layer [48][24] to forward them to neighbor node. MAC layer takes random delay between 0 to maximum staggering delay in order to avoid collisions. The propagation delay depends on deterministic factors (ex. distance and packet size) and non-deterministic factors (ex. channel condition, weather and moving objects) [41]. Thus, it is impossible to accurately get the maximum propagation delay in general. In [55], authors set the maximum staggering delay to 0.03sec and estimate the maximum single hop delay, D_{maxHop}, of 0.3 sec for CSMA. Thus, we also set the D_{maxHop} to 0.3sec.

$$x = \left\{ E(T^{time}) - D_{maxHop} \cdot D_{sink} \right\} \tag{3.17}$$

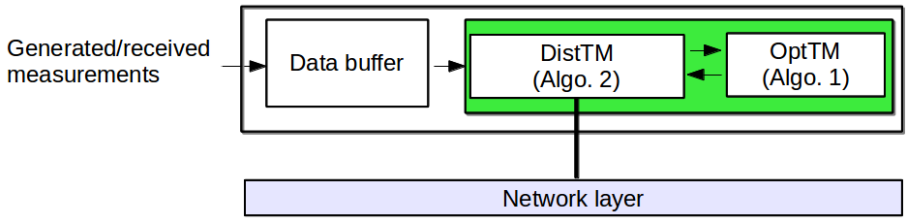

Figure 3.4: Three components of multi-hop TM

Algorithm 6 describes how distributed transmission manager determines final step N and connects to OptTM. Inputs to the algorithm are buffer with all the measurements (B_{in}) and distance to the sink (D_{sink}). All measurements in B_{in} have sensor readings and end-to-end time constraints. Each measurement may have different end-to-end time constraint. Network layer provides D_{sink} to the transmission manager. In lines 6:7, distributed transmission manager calculates N. Algorithm 5 uses N as an input and calculates optimal transmission instance between 1 and N in line 9. After distributed transmission manager gets the transmission instance, it updated end-to-end time constraints of all original measurements in B_{in} in lines 11:16. When updated end-to-end time constraints have zero or negative values, distributed transmission manager removes corresponding measurements from B_{in}. Lastly, distributed transmission manager schedules transmission of all measurements in B_{in} at the optimal transmission instance in line 17. The computation and space complexity of this algorithm is $O(length(B_{in})+N)$.

3.4 Experimental setup

We evaluate the performance of our proposed OptTM in terms of percentage of expired measurements and energy consumption (mJ) compared to the following state of the art approaches:

- *Fixed* [55]: All nodes have a fixed buffering time limit and periodically transmit all buffered measurements at every instance.

- *Cas* [55]: The **cascade time-out protocol** considers the distance to the sink node to evaluate the buffering time limit. Farther nodes have shorter buffering time limit. For single-hop analysis, this method converges to the shortest time interval between transmission instances.

- *CL* [70]: The **control-limit transmission manager** transmits all buffered measurements when the buffer size is over a predefined threshold. This threshold is a function of arrival rates and time constraints. The decision epoch can be either random or deterministic. We use the deterministic version in our simulations. Fig. 3.5 shows the buffer threshold for different sampling intervals and maximum time constraints. The threshold increases while we increase maximum time constraints and decrease the sampling interval. In this work, for a sampling interval of 2 sec, the buffer threshold varies from 8 to 45 measurements for maximum time constraints between 15 to 90 sec.

- *SF* [12]: The **selective forwarding transmission manager** calculates a threshold based on the consumed energy and importance of measurements. Importance is an inverse function of time constraints. It sends a measurement if the measurement's importance is larger than the calculated threshold. Otherwise, SF discards the measurements.

- DistTM: The **distributed transmission manager** employs OptTM which determines the optimal transmission instance based on Eq. (3.8) and Eq. (3.15) at every decision epoch. At each decision epoch, DistTM calculates upper bound of transmission instance, and OptTM uses it to get the optimal transmission instance. Then, DistTM schedules transmission of all buffered measurements at the optimal transmission instance. The details of OptTM and DistTM are described in Algorithm 5 and Algorithm 6.

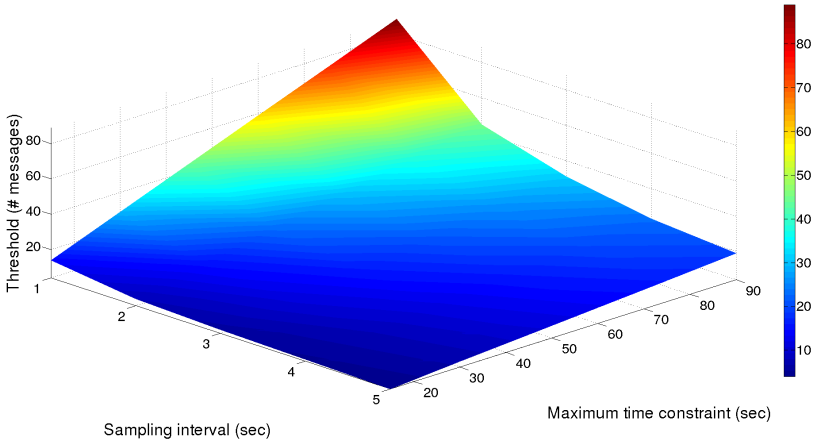

Figure 3.5: Buffer threshold of CL for different sampling interval and time constraints

We implement all the above approaches with ns3 network simulator [45]. Our approach operates on top of the network layer, so it is agnostic to the underlining routing protocol. Thus, we choose well established routing protocol, DSDV [29]. The distance between the two closest nodes is 100m.

We consider three different network topologies: i) single hop, ii) linear and iii) grid. The single hop considers a situation with a single source which is directly connected to a sink. Such direct communication between a sink and sensors is common in small scale WSNs such as body area networks (BANs) as shown in Fig. 3.6 [8]. In this case, we use CC2630 @2.4 GHz [17] as a transmission device. The radio module runs as 3V with transmit current at 6.1mA, receive current at 5.9mA and sleep current 100 nA. Linear topology establishes a routing path as depicted in Fig. 3.7. An good example is lake monitoring application that deploys multiple buoys in an single lake to monitor temperature, dissolved oxygen, conductivity, pH/ORP, flourescence (Chloraphyll a and Blue-green Algae) [69] to understand important issues such as spatial distribution of algal bloom and invasive species. Lastly, we imagine sensor nodes deployed in San Diego downtown form a grid as shown in Fig. 3.8. Such topology has been assumed for air quality monitoring in CitiSense project [44]. Blue node is the sink node while red

Figure 3.6: Wireless healthcare system architecture [8]

Figure 3.7: Example of linearly deployed buoys in an single lake [69]

nodes generate and relay measurements. Nodes in both linear and grid topologies have a non-QoS 802.11b radio which uses ad-hoc model with 1 Mb/s maximum data rate. The WiFi module runs at 3V with transmit current at 380mA, receive current at 313mA, idle listening current at 273 mA and sleep current at 33mA [45]. We only consider the energy consumption of transmission device with a linear battery model because our approach only changes the amount of data communicated. Constant speed propagation delay model characterizes the channel conditions.

We use synchronized and unsynchronized sampling methods. In synchronized scenario, all nodes periodically generate measurements at a predefined time interval. This is typical in monitoring systems which use WSNs to measure fine grained environmental

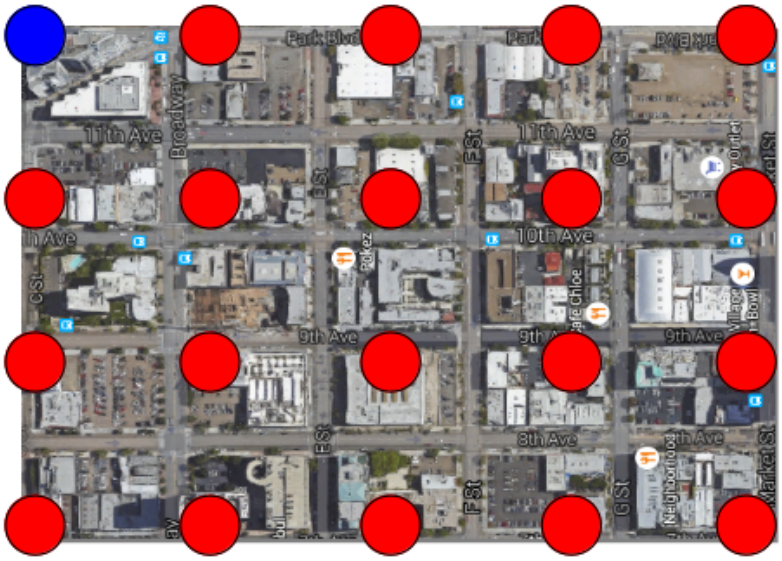

Figure 3.8: Considered topology for the realistic network simulation. Node 0 (Blue node) is the sink node. Other nodes generate and relay measurements to the sink node

factors [44]. We set the sampling time interval to 5 sec. For example, environmental monitoring application such as CIMIS [18] use temperature, humidity and pressure sensors use 5 sec granularity of sampling. In unsynchronized scenario, all nodes generate measurements with randomly selected sampling time which is uniformly distributed between 1 to 5 sec. This is typical when different applications require different sampling intervals. Time constraints of measurements are uniformly distributed between a minimum of 1 sec and a varying the maximum value ranging from 15 to 90 sec. The varying maximum value enables us to study the effect of heterogeneous time constraints. Similar values have been used in deployed WSNs. For example, in the air quality monitoring system [44] deployed sensors provide data to the patients every 30 sec. Vaisals weather stations [64] measure weather conditions such as temperature and humidity every 1 to 120 sec. The initial time interval between decision epochs is set to be 5sec. The total simulation time is set to 24 hours to represent a single day.

3.5 Experimental results

3.5.1 Network simulation of BAN

In this section, we compare the percentage of expired measurements and wireless energy consumption of transmission managers in a small scale, single hop WSN. This type of deployment is typical of Body-area Networks (BANs) which consist with one-hop distance devices and sink. We evaluate the performance of the state of the art transmission managers for both synchronized and unsynchronized sampling. *Fixed* transmits all the buffered measurements at every decision epoch (5sec). *CL* [70], *SF* [12] and DistTM check data buffer at every decision epoch (5sec) and decide when to transmit buffered measurements. *Cas* [55] calculates the time interval between transmissions based on the node distance to a sink. Since there is only one hop, it transmits measurements every 9.334sec.

Results with synchronized sampling: In this scenario, a source node generates measurements at a fixed time interval of 5 sec. Fig. 3.9 and Fig. 3.10 show the percentage of expired measurements and energy consumption with different maximum time constraints. We normalize the energy consumption by *Fixed* with 15 sec maximum time constraint. While we increase the maximum time constraints to 90 sec, *Fixed* and *Cas* consume same amount of energy because of fixed transmission intervals. *Cas* consumes less energy than *Fixed* because *Cas* has longer time interval (9.334 sec) between communications than *Fixed* (5 sec). As described in Fig. 3.5, *CL* adjusts buffer limit from 4 to 19 for different maximum time constraints, and transmits measurements if buffer length reaches to the limit. *CL* has between 80 to 99.7% of measurements that expire because the buffer length does not reach the transmission threshold for most decision epochs. On the other hand, *CL* also consumes the least energy. *SF* uses time constraints of a measurement, transmission and reception power to calculate the

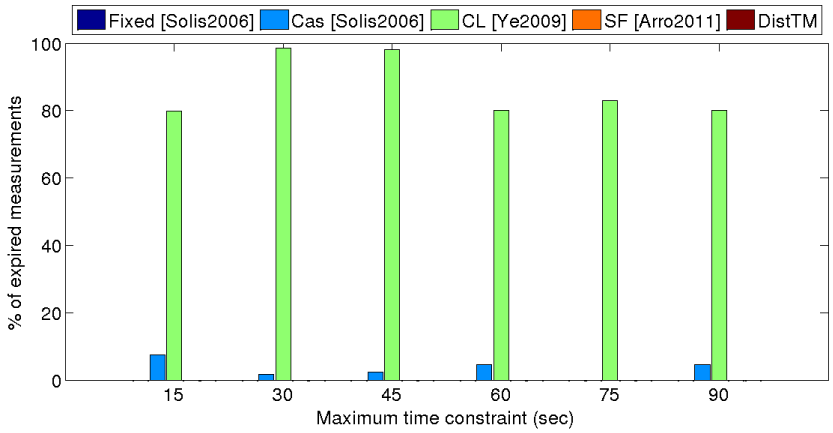

Figure 3.9: Percentage of expired measurements

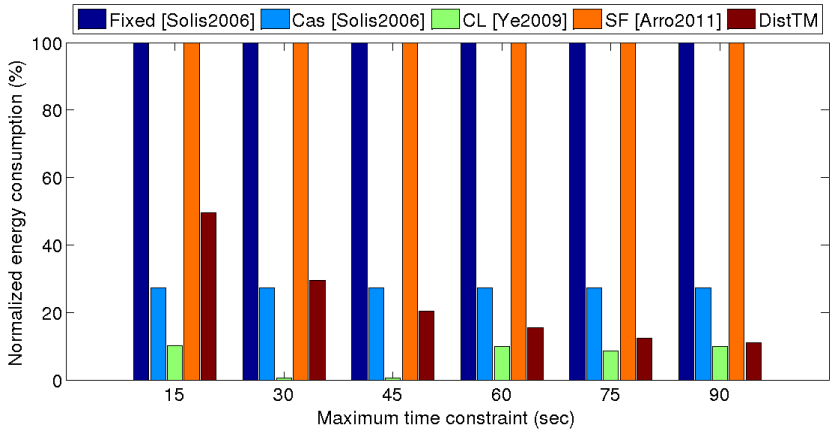

Figure 3.10: Energy consumption normalized by *Fixed*

Body Area Network (BAN)

threshold and the importance of the measurement. When the importance is larger than the threshold, SF transmits the measurements. Otherwise, it discards them. However, given varying maximum time constraint ranging from 15 to 90sec, all of measurements have higher importance than the thresholds. Thus, *SF* expires no measurements for different maximum time constraints. *SF* achieves energy saving by discarding some portion of measurements, but remaining messages have to be transmitted at every decision epoch, so it consumes a same amount of energy as *Fixed*. DistTM dynamically adjusts the transmission instance based on measurements' time constraints. While we increase maximum time constraints to 90sec, DistTM consumes on average 158.2% less energy

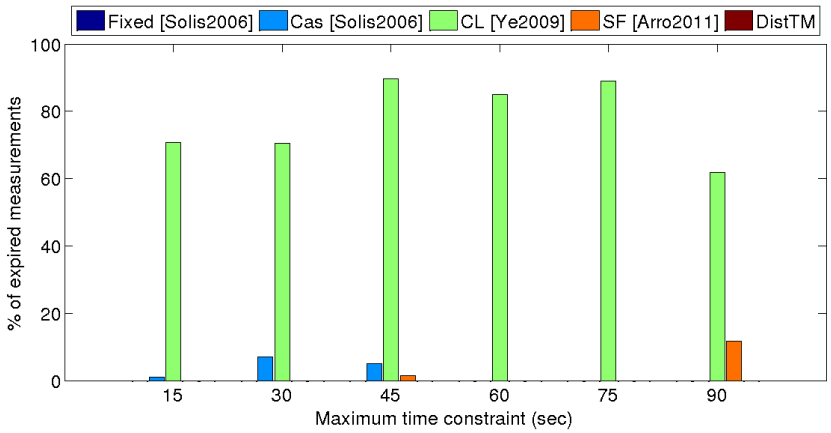

Figure 3.11: Percentage of expired measurements

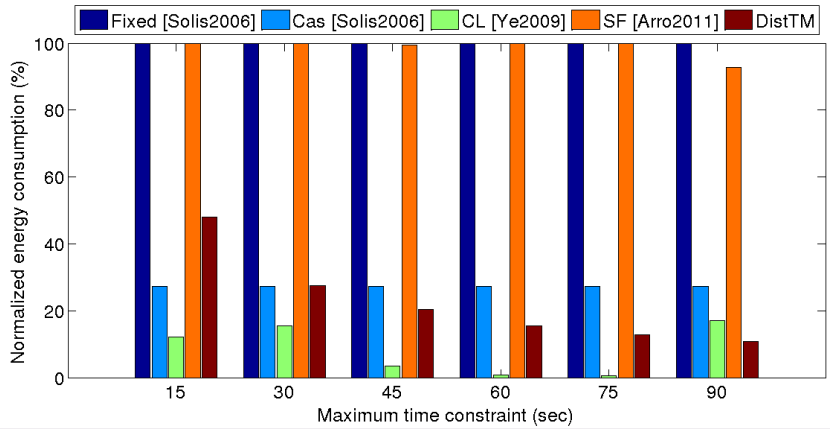

Figure 3.12: Energy consumption normalized by *Fixed*

BAN with random sampling interval

than other approaches (except for *CL* which does not deliver messages in timely fashion) with no measurements that expires.

Results with unsynchronized sampling: Unlike synchronized sampling, in this case each node generates measurements based on a varying sampling interval which is randomly selected between 1 to 5 sec. Fig. 3.11 and Fig. 3.12 describe the percentage of expired measurements and normalized energy consumption relative to *Fixed* with 15 sec maximum time constraint. *Fixed* has no expired measurements. *Cas* and *SF* expire at most 7.3% and 11.6% of measurements, but they consume on average 183.6% more energy than DistTM. While we increase the heterogeneity of time constraints, DistTM

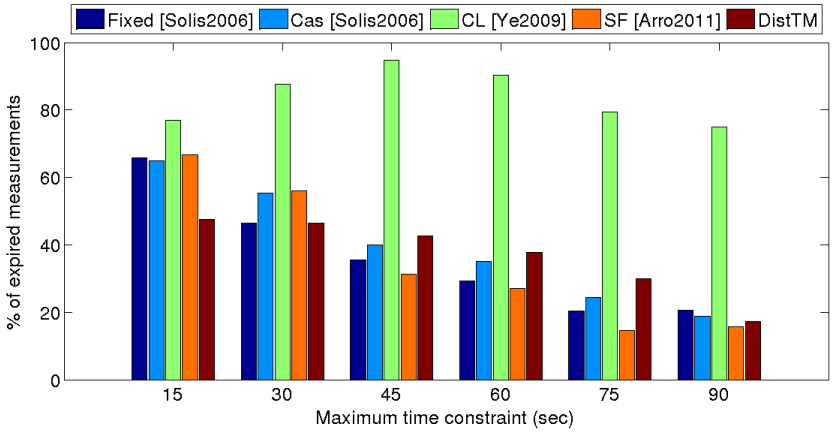

Figure 3.13: Percentage of expired measurements

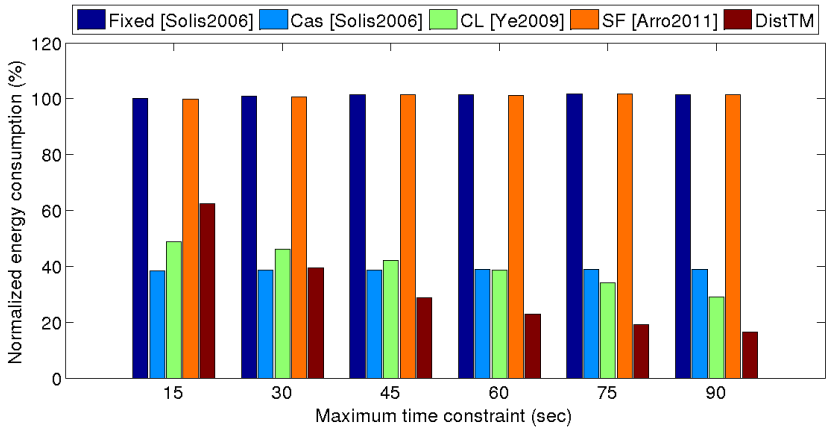

Figure 3.14: Energy consumption normalized by *Fixed*

Linear WSN

consumes on average 158% less energy than other approaches except for *CL* which does not meet the QoS requirements.

3.5.2 Network simulation of a linear WSN

This scenario is typical of buoys deployed on a lake. We compare the performance of transmission managers in terms of energy consumption normalized to *Fixed* with 15 sec maximum time constraint and the percentage of expired measurements when a set of 10 linearly connected nodes sample data and communicate findings to a single sink on the shore. Each node constructs routing path to the sink with table driven routing protocol,

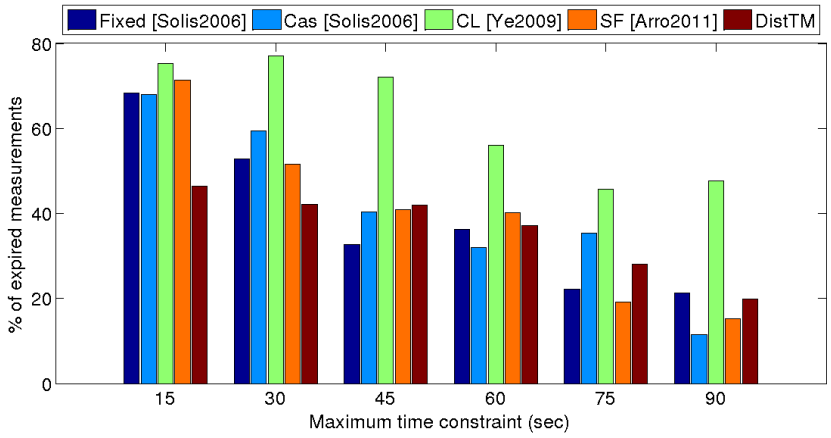

Figure 3.15: Percentage of expired measurements

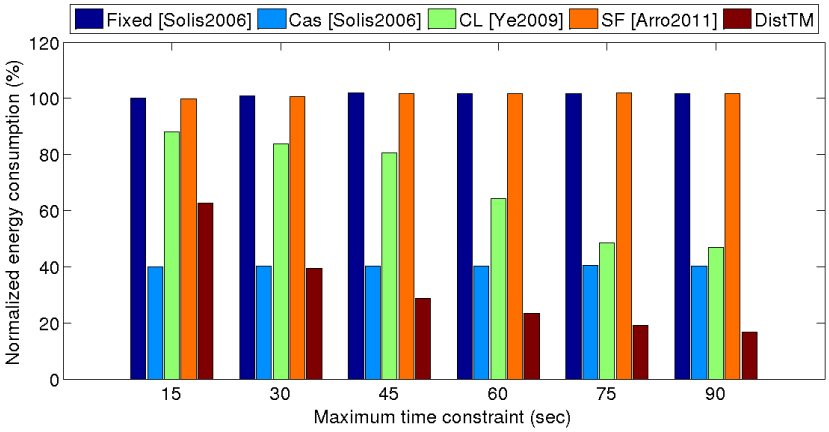

Figure 3.16: Energy consumption normalized by *Fixed*

Linear WSN with random sampling interval

DSDV. *Cas* determines transmission instance based on a node distance from the sink. More specifically, farther nodes have shorter time intervals between communications than nodes closer to the sink.

Results with synchronized sampling method: In this scenario, a source node generates measurements every 5 sec. Fig. 3.13 and Fig. 3.14 show the percentage of measurements that expire and the normalized energy consumption. DistTM adjusts transmission instance based on time constraints of buffered measurements. Thus, DistTM has on average 144.3% less energy consumption while it expires on average 19.3% less measurements than than all the other state of the art approaches. *Fixed*, *Cas*, *SF* periodically transmit buffered measurements without considering maximum time constraints. These strict approaches result in higher energy consumption as compared to DistTM. While we increase the heterogeneity of time constraints, *Fixed*, *Cas* and *SF* decreases the number of measurements that expire because they can deliver more measurements in timely fashion when we extend the maximum time constraints to 90 sec. However, they consume more energy than DistTM in order to transmit measurements. While we extend the the maximum time constraint to 90sec, *Cas* expires on average 2.8% more measurements while consuming on average 7% more energy than DistTM.

Results with unsynchronized sampling method: We further investigate the performance of transmission managers in a linearly constructed WSNs when all nodes use a uniformly distributed sampling interval between 1 and 5 sec. Fig. 3.15 and Fig. 3.16 show the percentage of measurements that expire and the normalized energy consumption for different maximum time constraints. In linear WSN, approaches using static transmission instance such as *Fixed* and *SF* do adjust the instance based on maximum time constraints, so they consume more energy and have lower percentage of measurements that expire as compared to DistTM. *Cas* consumes on average 8.3% more energy than DistTM while expiring 5.2% more measurements. However, with maximum time constraints

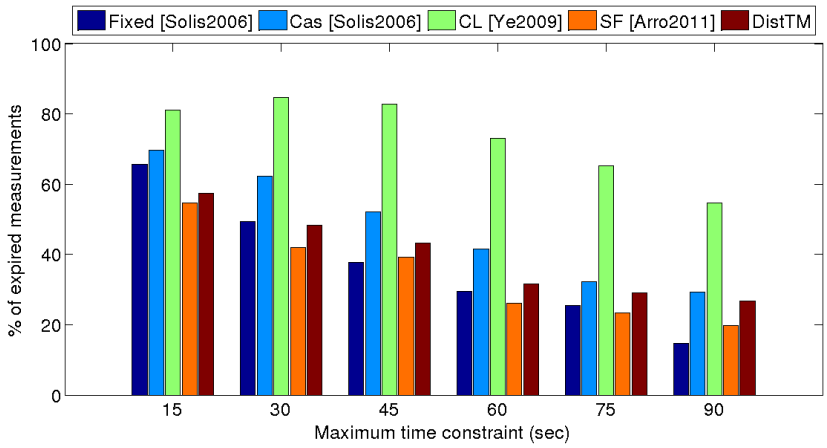

Figure 3.17: Percentage of expired measurements

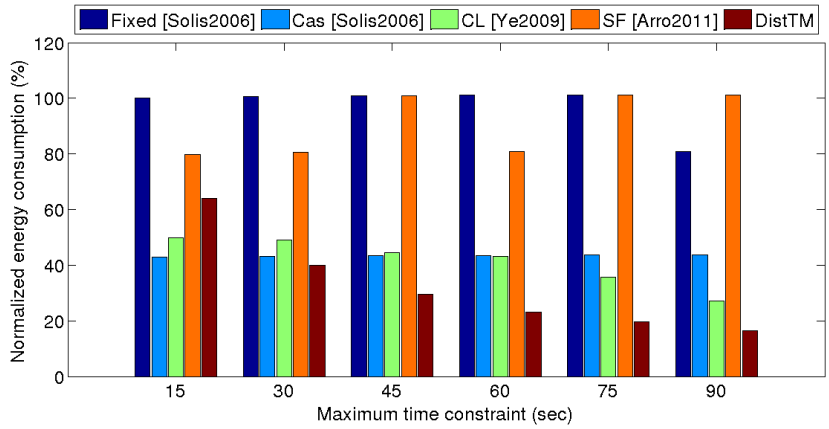

Figure 3.18: Energy consumption normalized by *Fixed*

Grid WSN (20 nodes)

larger than 30sec, *Cas* expires on average 2% less measurements than DistTM even though it consumes on average 18% more energy than DistTM. When we extend the maximum time constraints to 90 sec, DistTM dynamically adjusts transmission time between communications, so it decreases the energy consumption by decreasing the number of transmission events. DistTM consumes on average 146% less energy while having on average 4% less measurements that expire as compared to other approaches.

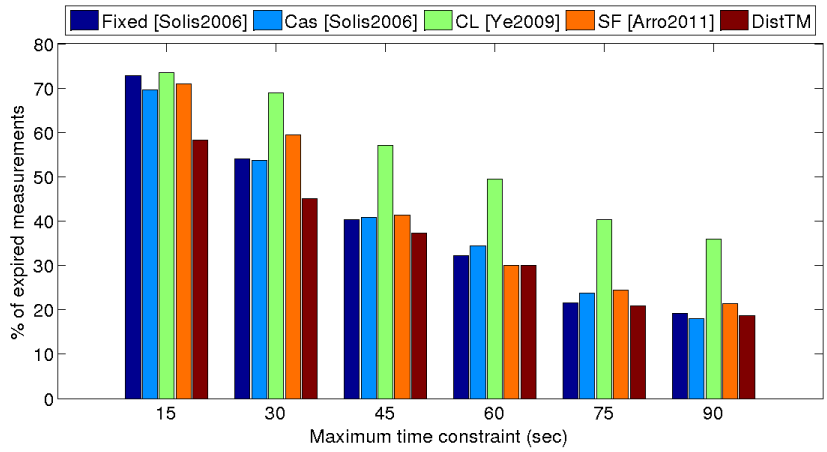

Figure 3.19: Percentage of expired measurements

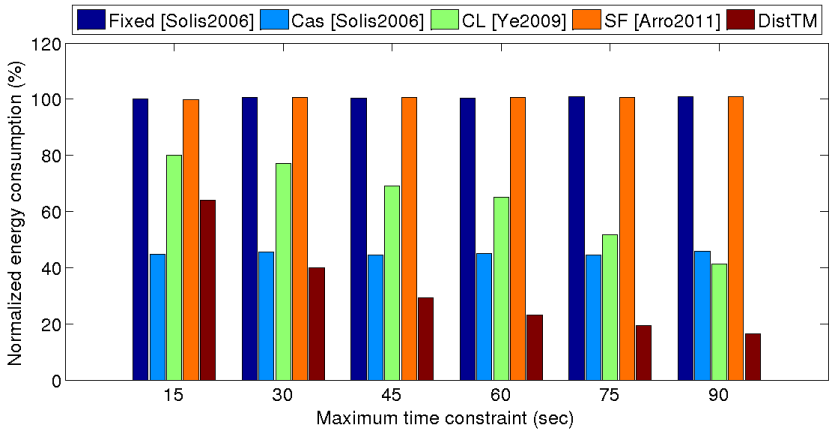

Figure 3.20: Energy consumption normalized by *Fixed*
Grid WSN with random sampling interval (20 nodes)

3.5.3 Network simulation of a grid WSN

In this scenario, we evaluate the performance of transmission managers in a grid topology based WSN. This is an example that has been developed to mimic an urban sensing project such as CitiSense [44]. We evaluate the performance of the state of the art transmission managers for both synchronized and unsynchronized samplings. As described in Fig. 3.8, WSN is constructed with 19 nodes and 1 sink. All nodes except for the sink construct routing paths with table-driven routing protocol, DSDV, and forward buffered measurements to the next stage nodes following the established routing path.

We study the effects of network size in Sec. 3.5.4.

Results with synchronized sampling method: All nodes in this WSN generate measurement every 5 sec. Fig. 3.17 and Fig. 3.18 show the percentage of expired measurements and the normalized energy consumption. *Fixed*, *Cas* and *SF* periodically transmit measurements without considering maximum time constraints at fixed time instances, so they have more measurements that expire while we increase the maximum time constraint to 90 sec. However, this results in higher energy consumption to transmit unexpired measurements. DistTM determines time interval between communications based on measurements' time constraint, so it consumes on average 141% less energy than other approaches with on average 0.3% less measurements that expire for different maximum time constraints as compared to *Fixed*, *Cas*, *CL* and *SF*.

Results with unsynchronized sampling method: In this section, we evaluate the performance of transmission managers when all nodes have sampling interval that is uniformly selected from 1 to 5 sec. Fig. 3.19 and Fig. 3.20 show the percentage of expired measurements and the normalized energy consumption for different maximum time constraints. *Fixed*, *Cas* and *SF* do not adjust time interval between communications based on time constraints of measurements. Unlike their approaches, DistTM dynamically finds the optimal transmission time based on measurements' time constraints as described in Algorithm 1 and Algorithm 2 at each decision epoch. DistTM decreases the energy consumption when we extend maximum time constraints to 90 sec. *Fixed*, *Cas* and *SF* decrease the percentage of expired measurements under same condition, but on average they consume 149% more energy than DistTM. The percentage of measurements that expire under these protocols is on average 4.3% more than DistTM.

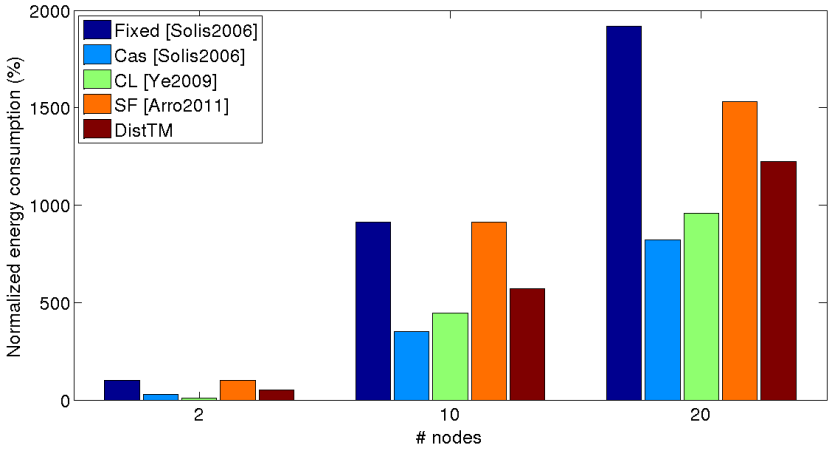

Figure 3.21: Maximum time constraint: 15 sec

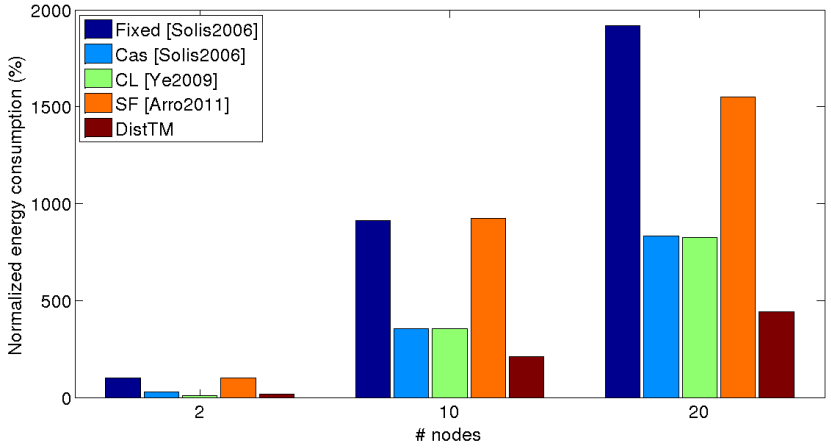

Figure 3.22: Maximum time constraint: 60 sec

Energy consumption normalized by *Fixed*

3.5.4 Results for different sizes of WSNs

In this section, we change network sizes ranging from 2 to 20 nodes with synchronized sampling method: single-hop WSN=2, linear WSN=10, and grid WSNs=20.

Fig. 3.21 and Fig. 3.22 show normalized energy consumption for different network sizes when maximum time constraints are 15 sec and 60 sec. Each result is normalized to *Fixed* in single-hop WSN with corresponding maximum time constraint. The number of nodes in WSN is shown on x-axis ranging from 2 to 20. Larger networks gather more measurements and generate more traffic. Thus, all approaches consume

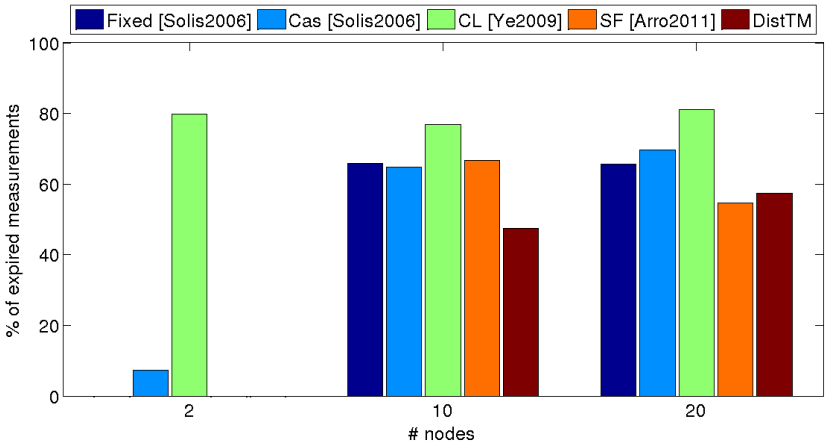

Figure 3.23: Maximum time constraint: 15 sec

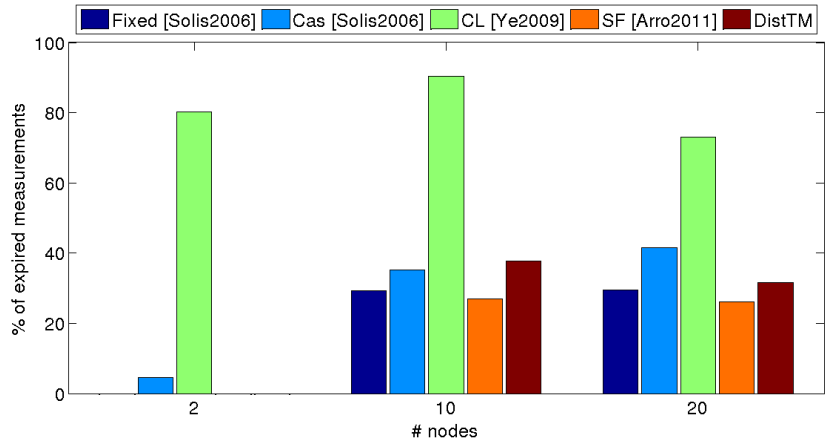

Figure 3.24: Maximum time constraint: 60 sec

Percentage of expired measurements

more energy with larger network sizes. DistTM dynamically adjusts the transmission time between communications based on measurements' time constraints, so it decreases energy consumption with larger maximum time constraints (ref. Fig. 3.22). When maximum time constraint is 15 sec, DistTM consumes on average 31.1% less energy and expires 17.8% less measurements. (ref. Fig. 3.21). When maximum time constraint is 60 sec, DistTM consumes on average 178% less energy with on average 13.2% less measurements that expire than all other approaches for different network sizes (ref. Fig. 3.22 and Fig. 3.24).

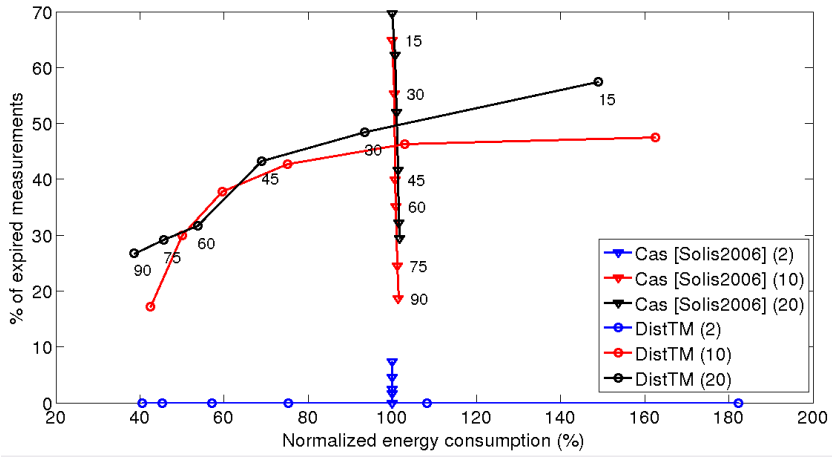

Figure 3.25: Synchronized.

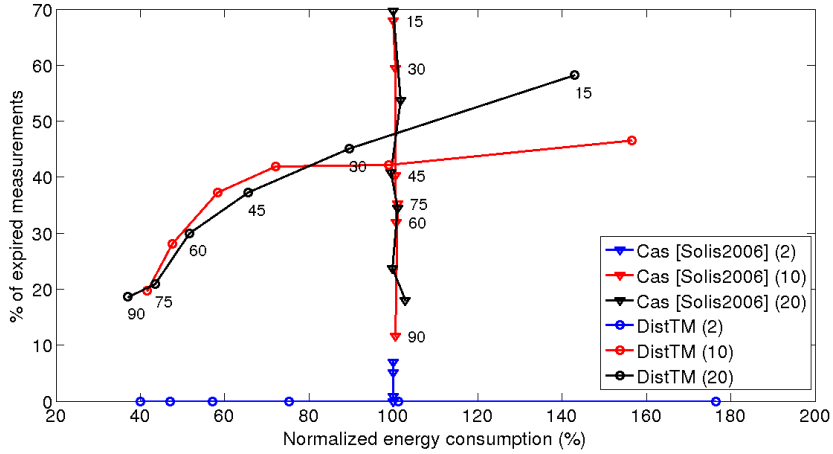

Figure 3.26: Unsynchronized.

Trade-off between normalized energy consumption and the percentage of expired measurements

3.5.5 Comparison between *Cas* and OptTM

As we can observe from previous results, DistTM always consumes less energy than the state of the art approaches except for *Cas* when maximum time constraints is less than or equal to 30sec in both synchronized and unsynchronized sampling methods. When maximum time constraint increases to 90sec, *Cas* consumes more energy than DistTM. Thus, in this section, we study their relationships for different maximum time constraints, network sizes and sampling methods. Fig. 3.25 and Fig. 3.26 describe the

trade-off between the percentage of expired measurements and energy consumption that has been normalized to *Cas*. Numbers in parenthesis denote network sizes: single-hop WSN=2, linear WSN=10, grid WSNs=20. Numbers on data points denote maximum time constraints. While we increase the maximum time constraints to 90 sec, *Cas* consumes the same amount of energy because *Cas* cannot adjust its transmission instance based on the time constraints of buffered measurements. However, DistTM decreases the energy consumption while we increase the maximum time constraints. In BAN, *Cas* expires on average 2.8% more measurements, and consumes on average 16.2% more energy than DistTM for different time constraints and sampling methods. In linear WSN, DistTM consumes on average 20% less energy than *Cas* while having on average 4% less measurements that expire. In Grid network, DistTM still outperform *Cas*. DistTM consumes on average 28% less energy and expires 6.7% less measurements. When we increase the network size, both approaches have high measurements' expiration rates with 15sec maximum time constraint. DistTM expires around 40% of measurements and *Cas* has at least 60% of measurements that expire. *Cas* only considers distance to sink, so its energy consumption is fixed for different maximum time constraints. DistTM considers both maximum time constraints and distance from the sink when it calculates transmission instance. When maximum time constraint is 15sec, DistTM consumes 50% more energy on average than *Cas* and has 13% less expired measurements on average. However, while we increase maximum time constraints, DistTM smartly adjusts transmission interval, so it consumes less energy then *Cas* with a lower number of expired measurements.

3.5.6 Overhead

In this section, we evaluate the computation overhead of transmission managers. We set 90 sec as the maximum time constraint of measurements because it causes the

worst case computation and space complexity of DistTM. We measure elapsed time of DistTM on a low-power and small-scale embedded device, Raspberry PI2 [7] (1GHz CPU and 1 GB main memory), and compare the elapsed time with the simplest approach, *Fixed*. 100 measurements are placed in data buffer. DistTM determines the optimal transmission instance based on measurements' time constraints. We repeat experiments 100 times. DistTM spends on average 0.4msec more time than *Fixed* resulting in 8% computation overhead. This shows that DistTM can operate on low-power embedded devices with minimum overhead while it significantly decreases both energy consumption and the number of measurements that expire.

3.6 Conclusion

In this paper, we propose optimal transmission manager (OptTM) in WSNs where multiple applications operate with different time constraints. OptTM determines the optimal transmission instance based on time constraints of both generated and received measurements in a single hop. We mathematically prove the existence of the optimal transmission instance. Distributed transmission manager (DistTM) evolves OptTM to operate in multi-hop WSNs. DistTM explicitly considers the relationship between end-to-end time constraints and the distance to sink unlike other approaches. We implement both DistTM and OptTM in ns3 simulator, and compare their performance with other state of the art approaches in terms of energy consumption and the number of measurements that expire. We consider three different network topologies, single hop, linear and grid, and vary maximum time constraints from 30 sec to 90 sec. For all different configurations, our proposed approach consumes on average 148% less energy than the other approaches with on average 14.1% less measurements that expire.

Chapter 3, in part, is a reprint of the material as it appears in "Transmission

manager in heterogeneous applications running WSNs", by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE Globecom 2015. The dissertaion author was the primary investigator and author of this paper.

Chapter 3, in part, has been submitted for publication of the material as it may appear in "Design of transmission manager in heterogeneous WSNs", by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, which was submitted to IEEE Transactions on Emerging Topics in Computing. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Adaptive information dissemination protocol

4.1 Introduction

The proliferation of smart portable devices (tablets and smart phones) is enabling mobile users to dynamically discover the sensors and interact with them in real-time. In this thesis, we focus on an important class of applications in which mobile users require only the data within a specific context (e.g., present location and current time), is of interest. Consider a sensor network deployed for air-quality monitoring. The deployed sensors can alert nearby users upon detecting high allergen levels. In this application, instead of continually streaming all the data to the backend servers, and having users access a small fraction of the data via long-haul networks, it is more efficient to enable users to access the sensed data within their context of interest. We propose a novel distributed technique that lets users, while moving around the deployed sensor network, collect information from the sensors in their vicinity. A key component of our technique is the sensor nodes ability to adjust their broadcasting rate as a function of the travel time

of users in their transmission range (broadcast area). To achieve this goal, each sensor node independently estimates users mobility at low cost. Most of the existing travel time estimation techniques rely on additional power hungry devices, such as cameras [22] or GPS. Such techniques are accurate at the expense of high power consumption and additional hardware, which makes them inappropriate for low-power, low-cost, ubiquitous sensor networks. In contrast, we use the number of acknowledgement (ACK) messages received from users as the statistical basis for estimating users mobility. As the users acknowledge the data, each sensor node selects the smallest sampling rate based on estimated users mobility thereby saving a significant amount of energy. Our technique is completely distributed and does not require any communication among sensor nodes. If the data is needed also at a later time, it can be forwarded by a users phone to a backend server. An application running on the server may provide improved interfaces, statistics and services based on the data collected. However, this has been addressed by other work [44] and is not the topic of this thesis. Our results show a decrease in power consumption of 2x to 8x (a single sensor) & 2x to 16x (network-wide) when compared to the existing protocols.

4.2 Related work

The mobile sink approaches [54][66] typically assume that the system has control over the sink mobility to ensure that it collects data from all the sensors. We instead assume that users carry their mobile phones while moving freely throughout the area where sensors are deployed. In the proposed approach we assume that the sensor nodes do not have any control over users mobility (e.g., user speed, direction, etc.). While Xu et al. [67] does not require any control over the sink mobility, the data is forwarded to mobile sink by acquiring trajectory of mobile sink and establishing energy-efficient

routing protocol, which does not scale well with number of mobile users. Recent work proposed a technique where a sensor node wakes up when it receives a RFID impulse from a user and then it unicasts the data [32]. However, this mechanism requires users to carry an RFID reader that is expensive and cumbersome. We next discuss the design of our adaptive broadcast mechanism.

4.3 Adaptive information dissemination protocol

Our system consists of a network of stationary wireless sensor nodes deployed in an area of interest. Each stationary sensor node has a low power microcontroller, one or more sensors, and a radio. We assume that each node uses a low power MAC protocol capable of sending broadcast packets and capable of minimizing channel contention and collisions. As shown in Figure 4.1, sensor nodes divide the time in a day in slots of variable length ($Slots_s, s \in [1,d]$). In addition, each time slot is divided into two components: (a) Traffic estimation slot and (b) Adjustment slot. During the traffic estimation slot, sensors broadcast messages at a fixed rate and gather data needed for estimating the users mobility. During the adjustment slot, sensors dynamically adjust their (1) broadcast rate and (2) the length of the adjustment slot to match the estimated users mobility pattern.

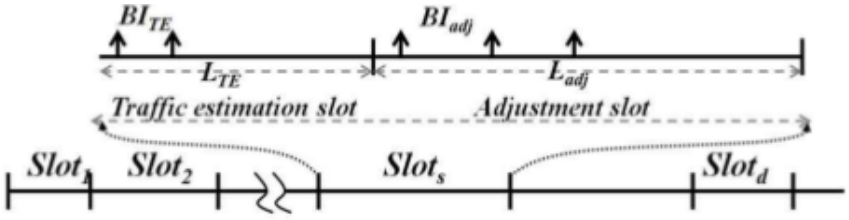

Figure 4.1: A day is divided in d time slot $Slot_s$

4.3.1 Traffic estimation slot

Ideally, the best way to estimate real user travel time is by using measured values of users speed and direction as shown in Equation (4.1). Here user i^s travel time, T_i, is a function of the users speed ($v_{real,i}$) and length of the users trajectory within the transmission range, where l_i denotes the distance from the sensor node, S with transmission radius r. Figure 4.2 shows that the distance between a user and a sensor node S, and l_i can be described with users angle of arrival, θ_i.

$$T_i = \frac{2\sqrt{r^2 - l_i^2}}{v_{real,i}} = \frac{2r\cos\theta_i}{v_{real,i}} \tag{4.1}$$

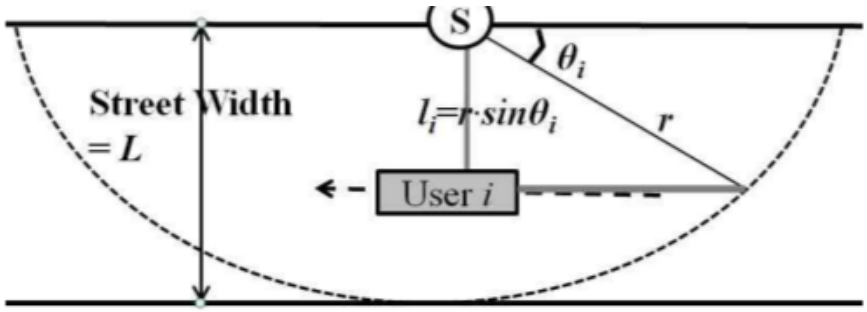

Figure 4.2: Example of distance and arrival angle of user

However, in reality, when a sensor node measures users travel time, users angle of arrival and speed are not measurable without special equipment. Thus, we propose a novel low- power ACK-based mechanism for estimating the users travel time which does not require above information. During the traffic estimation slot, sensors broadcast messages at a fixed rate with the goal of estimating the users mobility (i.e. travel time). Each time a mobile user receives a message from a sensor, it acknowledges it by unicasting an acknowledgement (ACK) message, which includes its unique ID. After every broadcast, a sensor node keeps its radio on for a fixed amount of time to ensure that it can receive the ACK messages from mobile users. A sensor node then uses the number of ACKs

received from the users as the statistical basis for estimating users mobility. Each sensor node maintains a two-column table. The first column stores the users IDs and the second column has the number of ACKs received from that user during the traffic estimation slot. This allows sensor nodes to detect if a new user has moved into the transmission range, and to estimate the user travel time using sequences of ACKs from the same user. The estimated users travel time (T_i) is a function of the amount of time between broadcasts (BI_{TE}) and the number of received ACKs (K_i) as shown in Equations (4.2) and (4.3). When a users actual travel time is larger than BI_{TE}, depending upon the users arrival time a sensor can receive either case 1: ceil(T_i/BI_{TE}), or case 2: floor(T_i/BI_{TE}) number of ACK messages. For example, when the users travel time is smaller than the BI_{TE}, in case 1, the user arrives within the sensors transmission range just before the broadcast and receives one message from the sensor node. In case 2, the user arrives and leaves the region between two successive broadcasts and cannot receive any messages from the sensor node.

$$T_i^{'} = K_i \cdot BI_{TE} \tag{4.2}$$

$$K_i = f(T_i, BI_{TE}) = \begin{cases} ceil(T_i/BI_{TE}) & case\ 1 \\ floor(T_i/BI_{TE}) & case\ 2 \end{cases} \tag{4.3}$$

The upper bound of estimation error between actual travel time (T_i) and ACK-based estimated travel time (T_i) are described in Equation (4.4).

$$est.Error_i = \left| \frac{T_i^{'} - T_i}{T_i} \right| = \left| \frac{K_i \cdot BI_{TE} - T_i}{T_i} \right| < \frac{BI_{TE}}{T_i} \tag{4.4}$$

We now describe how a sensor determines when to terminate its current traffic estimation slot. When δ percent of users (δ is 90% in our experiments) leave the

transmission range, a sensor node generates a set of users travel times, T_{set}. It next sets the upper bound on the length of the traffic estimation slot, L_{TE}, based on the largest estimated users travel time, $max_(T_{set})$, and BI_{TE}, as shown in Equation 4.5.

$$L_{TE} = ceil\left(\frac{\max T_{set}}{BI_{TE}}\right) \cdot BI_{TE} \tag{4.5}$$

The time interval between the two successive broadcasts during the traffic estimation slot, BI_{TE}, is updated for next traffic estimation slot using exponential moving average as described in Equation (6). This allows a sensor node to further save its energy during the traffic estimation slot.

$$BI_{TE} = (1-\alpha) \cdot BI_{TE} + \alpha \cdot mean(T_{set}), \;\; \alpha \in [0,1] \tag{4.6}$$

In Equation (4.6), α denotes an applications sensitivity to current traffic conditions. When α is 0, then BITE is not updated with current traffic condition, whereas, when α approaches 1, the current traffic condition is the dominant factor and the BI_{TE} is updated based sample mean of (T_{set}). The benefit of this can be seen as follows. When the current traffic condition is slower than past traffic conditions, mean(T_{set}) decreases, which in turn increases BI_{TE}. This optimization allows sensors to broadcast at a lower rate during the estimation slot and save energy. Parameter α is set in an application-specific manner. Existing research [15] shows that the average speed of all different races such as European, American, Austrian, and Asian is 1.34 m/s with a standard deviation of 0.26 m/s. We set the initial value of BI_{TE} as $(2 \cdot \text{Transmission range})/1.34$. Transmission range is determined from the transmission power specifications [13].

4.3.2 Adjustment slot

At the beginning of each adjustment slot, sensor nodes determine the length of time between broadcasts (BI_{adj}) and the length of adjustment slot (L_{adj}) based on T_{set}, obtained during the traffic estimation slot. During the adjustment slot, sensor nodes broadcast messages every BI_{adj} interval while successfully meeting application requirement. After every broadcast, sensor nodes turn off their radio to save power. To determine the value of optimal BI_{adj}, we first define Ni as the expected number of packets received by each user i during the adjustment slot over BI_{adj}, given the estimated travel time of user i, $K_i \cdot BI_{TE}$, is obtained during the traffic estimation slot.

$$N_i = floor\left(\frac{K_i \cdot BI_{TE}}{BI_{adj}}\right) \text{ where } K_i \cdot BI_{TE} \in T_{set} \tag{4.7}$$

Then, the packet reception reliability is calculated by counting the number of users who receive at least one packet during their travel time as shown in Equation (4.8). The $I(x)$ is an indicator function of x which returns 1 when x is positive, and returns 0 otherwise. Here $|T_{set}|$ is the size of T_{set}.

$$Pkt_{rel}(i) = E[I(N_i)] = \sum_{i=1}^{|T_{set}|} I(N_i)/|T_{set}| \tag{4.8}$$

The average number of packets received by every user when sensor nodes broadcast every BI_{adj} seconds, $Pkt_{cnt}(i)$, is described in Equation (4.9).

$$Pkt_{cnt}(i) = E[N_i] = \sum_{i=1}^{|T_{set}|} N_i/|T_{set}| \tag{4.9}$$

If we increase BI_{adj}, then both $Pkt_{rel}(i)$ and $Pkt_{cnt}(i)$ decrease as shown in Equations (4.8) and (4.9). Thus, the optimal BI_{adj}, BI^*_{adj}, is the largest BI_{adj} that guarantees $Pkt_{rel}(i)$ satisfies application defined quality of service (e.g. 90% of users receiving at

least one packet while travelling through a sensors transmission range), while minimizing $Pkt_{cnt}(i)$ as described in Equation 4.10. The selected BI^{*}_{adj} is derived at the beginning of the adjustment slot, and BI_{adj}, in equation (4.10), is any set of positive numbers.

$$BI^{*}_{adj} = arg\min_{BI_{adj}} Pkt_{cnt} \;\; s.t. \;\; Pkt_{rel} \geq QoS, \;\; QoS \geq 0 \tag{4.10}$$

The length of the adjustment slot determines the system reliability because a mismatch between the estimated and the ongoing traffic conditions degrades the system performance in terms of data collection reliability (when users speed increases) or energy efficiency (when the user slows down). We initially set the L_{adj} using T_{set} as shown in Equation (4.11). Then, the length is adjusted based on traffic condition similarity.

$$L_{adj} = mean(T_{set}) + 2 \cdot std(T_{set}) \tag{4.11}$$

To check for the similarity, we use the concept of Prediction Interval (PI), shown in Equation (12), where the μ_{prev} and σ_{prev} are mean and standard deviation from the previous traffic estimation slot. The t^{*} in Equation (12) follows students t-test and is determined by the application defined success rate. For example, if a sensor node wants 99% of the mean (T_{set}) to fall into prediction interval, the node can set t^{*} as 2.58 (most standard statistical textbooks list t* values [12]).

$$PI = [\mu_{prev} - t \cdot \sigma_{prev}, \mu_{prev} + t \cdot \sigma_{prev}] \tag{4.12}$$

Current users mobility metrics are considered to be similar to the previous ones if the current mean (T_{set}) lies within PI. However, the duration of the current traffic distribution has random nature, so when conditions are similar, the sensor node increments L_{adj} by using the binary exponential backoff algorithm mentioned in Equation (4.13).

$$L_{adj} = \begin{cases} L_{adj} & similar = FALSE \\ L_{adj} + rand([0, 2^{\#\,success}]) \cdot L_{adj} & similar = TRUE \end{cases} \tag{4.13}$$

Parameter (# success) is initialized to 1 and is incremented by one until it reaches the system defined parameter, S_{max}. If the current mobility estimates are not similar to the previous, the sensor node does not increment L_{adj} and set # success to 1. The above optimization allows sensor nodes to increase the length of the adjustment slot when the traffic conditions do not vary much and save more energy. When T_{set} is empty, a maximum value of travel time (T_{max}) is set to BI_{adj} to save energy. After the adjustment slot ends, the sensor nodes start a new traffic estimation slot. During the adjustment slot a sensor simply broadcasts data at BI_{adj} interval and does not require ACKs. Thus, when BI_{adj} is always larger than or equal to BI_{TE} (which we now prove), we prove that adjustment slot always spend less energy than traffic estimation slot.

Lemma 1. *If all $K_i = 1$ where $K_i \cdot BI_{TE} \in T_{set}$, the BI^*_{adj} that satisfies Equation (4.10) is BI_{TE} (i.e. $BI^*_{adj} = BI_{TE}$).*

Proof. a) If $BI_{adj} > BI_{TE}$, then all N_i are 0 (ref. Equation (4.7)) and $Pkt_{rel}(i)$ is also 0 for all i. This violates the constraints of Equation (4.10), so $BI_{adj} \leq BI_{TE}$. b) Pkt_{cnt} is a non-decreasing function of N_i (ref. Equation (4.9)). Since N_i is inversely proportional to BI_{adj}, Pkt_{cnt} is also inversely proportional to BI_{adj}. Thus, the min. Pkt_{cnt} is achieved when $BI_{adj} = BI_{TE}$. □

Theorem 3. *If all $K_i \geq 1$ where $K_i \cdot BI_{TE} \in T_{set}$, the optimal BI_{adj} that that satisfies Equation (4.10) is greater than equal to BI_{TE}.*

Proof. Proof: a) If $K_i \geq 1$ and $BI_{adj} < BI_{TE}$, then Ni 1 (ref. Equation (4.7)) and Pkt_{rel} becomes 1 (ref. Equation (4.9)). If $K_i \cdot BI_{TE} \geq BI_{adj}$ and Ni 1, then there exists some BI_{adj} such that $BI_{adj} \geq BI_{TE}$ (i.e. $K_i \geq BI_{adj}/BI_{TE} \geq 1$). b) Since both N_i and Pkt_{cnt}

are inversely proportional to BI_{adj}, the largest BI_{adj} that satisfies reliability constraint is always $\geq BI_{TE}$. □

4.4 Simulation setup

Radio specifications and accurate user mobility modeling are critical for evaluating our proposed approach.

Radio specification: We use Chipcorn CC2420 IEEE 802.15.4 radio specifications [13] in our simulations. We vary the transmission range as 10, 30, and 55 meters. These ranges are derived using Friis transmission equation. The corresponding power consumption values are shown in Table 4.1. With the maximum packet size of 123 bytes transmitted at 250kbps, the packet transmission takes 4ms, and average ACK reception time is 0.5ms [13]. In order to calculate the distance between receiver and sender, we use the Friis transmission equation, and set receiver and transmitter gain as 2. We also assume the frequency is 2.4MHz.

Table 4.1: Compare related works to proposed transmission manager

	CPU idle	Transmit (10m)	Transmit (30m)	Transmit (55m)	Receive
Power (mA)	0.712	8.41	9.71	10.9	18.8

As shown in Equation (4.14), in order to model transmission range in a realistic manner, we use log-normal shadowing radio propagation model [41]. In this equation X is the shadowing deviation, which determines the radio irregularity

$$P_{rx} = P_{tx} \cdot (C)^{\eta} \cdot 10^{X/10} \tag{4.14}$$

User mobility modeling: User mobility pattern has a significant impact on design,

Table 4.2: Simulation parameters for steady traffic flow conditions: congested and non-congested [60]

Traffic type	Mean (m/s)	Stdv (m/s)	Density (ped/m^2)	Flow level (m/s)
Non-congested	1.46	0.15	0.2	0.2
Congested	0.96	0.26	0.8	1.2

development, and performance of network protocols [41]. Existing research has shown that simple user mobility models do not generate realistic movement patterns [41]. Existing researches [59][30][53] show that the behavior of masses of people can be modeled similar to gases or fluids. Thus, we use the model that characterizes users mobility with three parameters: density (user/m2), flow level (user/sm), speed (m/s). The density is represented by the number of users in a confined space (e.g. street), and the flow level is the number of arrivals per second in a given area [52]. The relationships between those parameters are described in Equation (4.15).

$$Density(ped/m^2) = \frac{Flow\ level(ped/s)}{Mean\ of\ flow\ speed(m/s)} \tag{4.15}$$

The street width and the number of users in a confined area determine the mean and the variance of the flow speed [53][60]. When there are few users in confined space, the user density is proportional to the flow level. However, when there are a lot of users in a confined space, density and flow level have inversely proportional relationship and the mean of flow speed decreases because the users adjust their walking speed to avoid physical interaction with other users, which increases variation flow speed (ref. Table 4.2).

We evaluate the performance of the proposed protocol against the following three state of the art protocols: periodic information broadcast, non-uniform information dissemination, and RFID impulse protocol.

Periodic information broadcast protocol: In this protocol, the sensor nodes broadcast

data at fixed rate. This protocol is simple to implement, however since it cannot adapt to user mobility it results in either high overhead (when the users speed is low) or low reliability (when the user speed is high). The broadcast interval is calculated as function of the transmission range and mean users speed

Non-uniform information dissemination protocols: Tilak et al. [56] proposed a suite of non-uniform information dissemination protocols, where the packet forwarding probability is inversely proportional to the distance the packet has traveled. In other words, if a sensor receives a packet from a close neighbor, it is more likely to forward this than a packet received from a neighbor much farther away. Our approach is a special case of non-uniform protocol where only the one-hop information is relevant to a given user.

RFIDImpulseprotocol [32]: This protocol assumes that all sensor nodes turn off their radios as long as they have no packets to send or receive. When a sensor has a packet to send, it triggers RFID tag of a user. Then the user generates interrupt to wake up radios, and send ACK to the sensor when radios become fully active. The ACK represents the successfully wake-up of the user, so the sensor starts transmit packet. However, their approach is not suitable if the goal is provide information to all users who pass through the sensor nodes transmission region. Thus, we reversed the role between user and sensor in order to the protocol can serve multiple users request such that when a sensor node receives a RFID impulse from the user it wakes up and sends (unicasts) its data. Later, we use term RFIDImpulse to denote the revised RFIDImpulse protocol. The updated protocol is more suitable for the studied application than the original protocol.

4.5 Results

We evaluate the performance of the proposed protocol in terms of power consumption and data collection reliability under two different cases: (1) single sensor and

(2) multiple sensors nodes. We assume 100 users traverse the transmission range. The weighting factor (ref. Equation (6)) is set to 0.5. Since we consider pedestrian mobility, T_{max} is set as $\eta \cdot 2r$ where η is the reciprocal of the slowest users speed. We set η to 0.91m/swalking speed of an elderly person [60].

Energy-efficiency and reliability comparison study: In this study we evaluated the energy-efficiency of all the protocols. We vary the transmission range as 10, 30, and 55 meters. As described in Table 4.3 users arrive at a fixed flow level.

Table 4.3: Factor reduction in energy consumption for all the protocols (and number of messages transmitted by all the protocols in bracket) under steady state flow (non-congested traffic condition) for different transmission ranges

TxRange / Protocol	Adaptive	Non-uniform	Periodic	RFIDImpulse
10m	10.83 [21]	6.31 [41]	6.31 [41]	1.29 [100]
30m	23.97 [11]	12.06 [17]	12.06 [17]	1.12 [100]
55m	33.65 [6]	13.05 [12]	13.05 [12]	1 [100]

Table 4.3 shows the factor reduction in energy consumption and number of messages transmitted (in bracket) for various protocols. Since we do not consider forwarding overhead in case of the non-uniform protocol, its performance is identical to the periodic protocols performance. In the case of non-uniform and periodic broadcast protocols, once the broadcast interval selected at onset and then it does not change. However, in case of the adaptive protocol, a sensor node broadcasts at a fixed rate only during the traffic estimation slot whereas the broadcast interval during the adjustment slot is adapted based on user mobility estimated during the traffic estimation slot. Therefore the adaptive protocol transmits the least number of messages and significantly outperforms all other protocols (ref. Table 4.3). Since the adaptive protocol transmits significantly lower messages, and turn off radio after transmit packet, the factor reduction in energy consumptions varies from 10.83-33.65 (Adaptive), 0.06-0.05 (Non-uniform), and 1.29-1 (RFIDImpulse). The overall power consumption depends on both the transmission power/packet and the number of transmissions, but as shown in Table 4.3, as the transmission range increases

the overall energy consumption goes down. This is counter-intuitive; however, this happens for the following reason. During the traffic estimation slot, BI_{TE} is calculated as (2r / Average user speed), so BI_{TE} increases as a function of r. This implies that the number of messages transmitted is inversely proportional to the transmission range, r (ref. Table 4.3). For example, following tuples denote (transmission range in meters, average broadcast interval in seconds): (10, 15), (30, 45), and (55, 82). Therefore, the increase in power/packet due to higher transmission range is overcompensated by the decrease in number of transmissions.

We now compare the reliability of all the protocols. From, the reliability of the adaptive protocol is comparable to other protocols. In fact, at least 92% of users receive at least one packet while passing through the sensors transmission range, and the non-uniform/periodic protocols have 93% reliability. The RFIDImpulse protocol has the highest reliability since each sensor node wakes up and sends sensed data when it receives a RFID impulse from the user, but it requires the highest number of packet transmission as shown in Table 4.3 (in bracket).To sum up, reliability of adaptive protocol is slightly less than the non-uniform/periodic protocols by 1%, but it has significant energy savings at least 2x than others.

Effect of variation of δ: A sensor node generates a set of users travel times, T_{set} after the δ percent of users leaving the transmission range. Therefore, δ is a critical factor in evaluating the performance of the proposed protocol. To study its impact, we fixed BITE and traffic condition, and varied δ from 10% to 100%, which in turn varied L_{TE}. We observed the following "phase transition" phenomenon. When δ is less than 80%, the reliability is less than 20% (less than 20% of users receive at least one packet). However, when δ becomes larger than 80%, the system reliability becomes more than 95%. //

Effect of L_{adj} **adaptation**: We initially set the length of the adjustment slot using the average measured travel time of T_{set} as shown in Equation (4.11). However, as shown

in Equations (4.12) and (4.13), sensor nodes vary L_{adj}. The intuition behind this is that a sensor node can increase the length of the adjustment slot when the traffic conditions are do not vary much and save more energy. To quantify this benefit, we varied the transmission range as 10, 30, and 55 meters. We observed that when the traffic condition did not vary significantly over time, incrementing L_{adj} results in 20% decrease in sensors power consumption.

Impact of transmission range variation and shadowing on energy-efficiency: In this study we use log-normal shadowing propagation model [41] to explore the impact of radio irregularities on the energy efficiency and reliability of various protocols. As the shadowing deviation increases, the transmission range turns into a more irregular shape, which in turn increases the transmission range (users travel time). As shown in Figure 4.4 and Figure 4.5 this has two implications: (a) As the shadowing deviation increases, the ratio of energy spent in adjustment slot over estimation slot decreases. Therefore, a sensor node spends more time and energy in the estimation slot and less during the adjustment slot. (b) In case of congested traffic condition (ref. Figure 4.5), a sensor spends higher energy during estimation slot than in the case of non-congested traffic (ref. Figure 4.4). On the other hand, in case of congested traffic condition (ref. Figure 4.5), a sensor spends less energy during the adjustment slot as compared to the energy spend in non-congested traffic condition (ref. Figure 4.4). This happens because in the congested case since the user speed is lower than the non-congested traffic (ref. Table 4.2), both the $mean(T_{set})$ and $max(T_{set})$ values are higher, which in turn results in higher BI_{TE} and L_{TE} (ref. Equations (4.5) and (4.6)).

Sensor network-wide Power consumption: Till this point, we only considered power consumption for an individual sensor node. We now evaluate the power consumption at the network level. We assume that K sensors are placed in a field. Equation (4.16) denotes the energy consumption of non-uniform protocol for K (2) sensor nodes,

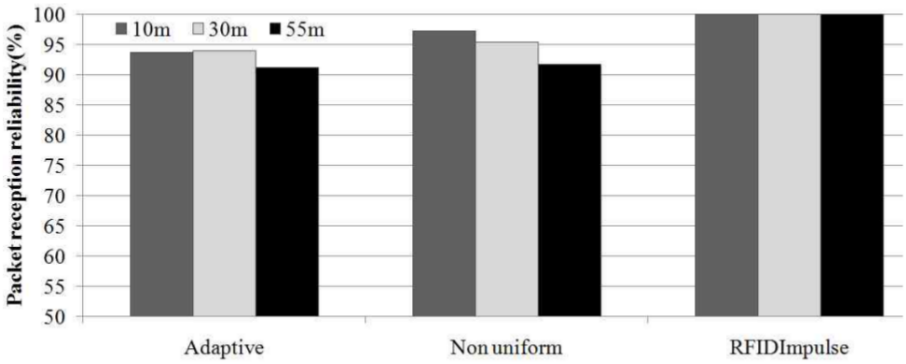

Figure 4.3: Comparison of reliability for all the protocols under steady state flow type (non-congested traffic condition) for different transmission ranges

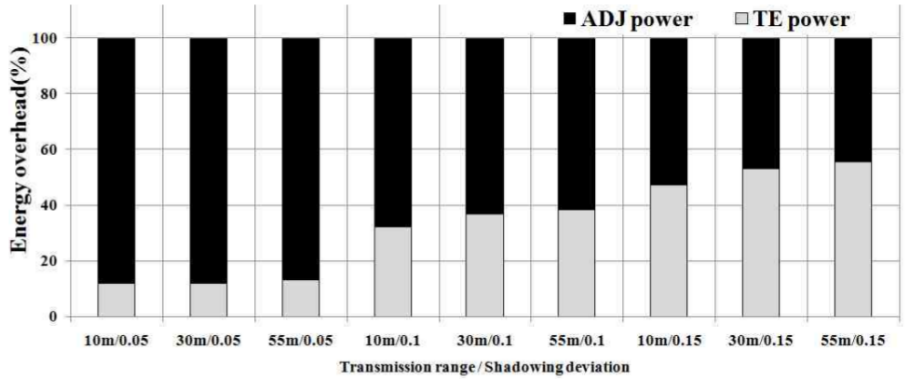

Figure 4.4: Impact of transmission range variation and shadowing on energy-efficiency of protocols non -congested traffic condition

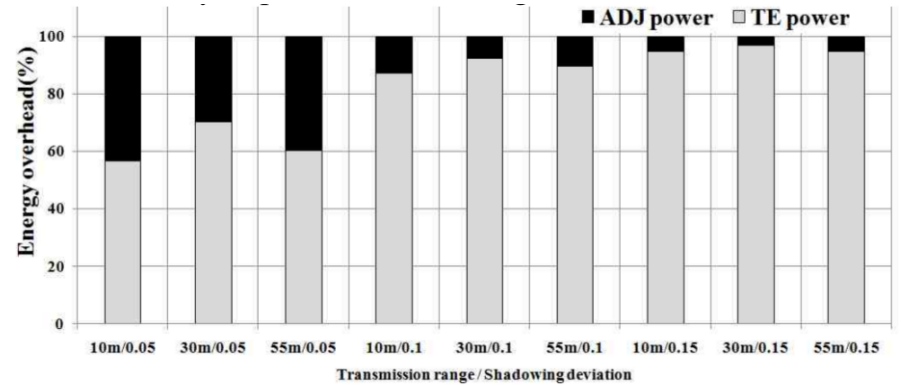

Figure 4.5: Impact of transmission range variation and shadowing on energy-efficiency of protocols congested traffic condition

where E_{tx}, E_{rx} are transmission and reception energies, N_{tx} is the number of packet transmissions. A sensor node placed I hops away from the sink has 1/i+1 as its forwarding probability [**?**].

$$E_i = \begin{cases} N_{tx} \cdot E_{tx} & ,i = k \\ N_{tx} \cdot E_{tx} + N_{tx} \cdot \left(E_{rx} + \frac{1}{i+1} E_{rx}\right) & ,otherwise \end{cases} \tag{4.16}$$

Equation (4.17) derives $K \cdot N_{tx} \cdot (E_{tx} + E_{rx}) + 0.5 \cdot E_{rx}$ is the lower bound of energy consumption of the non-uniform protocol. The energy consumption of adaptive protocol is affected by both the number of messages transmitted, N', and the number of sensor nodes, so its lower bound is $K \cdot N' \cdot (P_{tx} + P_{rx})$. Since N_{tx} is larger than N' as shown in Table 4.3 (in bracket) and the Non-uniform has an positive additional term, the adaptive protocol is more energy-efficient than the non-uniform protocol.

Table 4.4: Sensor network-wide Power consumption bound for all the protocols

Adaptive	Non-uniform	Periodic	RFIDImpulse
$K \cdot N' \cdot (E_{tx} + E_{rx})$	$K \cdot N \cdot (E_{tx} + E_{rx}) + 0.5 \cdot P_{rx}$	$K \cdot N \cdot (E_{tx} + E_{rx})$	$N_{user} \cdot E_{tx}$

$$\begin{aligned} \sum_{i=1}^{K} E_i &= \sum_{i=1}^{K-1} \left\{ N_{tx} \cdot E_{tx} + N_{i+1} \cdot \left(E_{rx} + \frac{1}{i+1} \cdot E_{tx} \right) \right\} + N_{tx} + \cdot E_{tx} \\ &= K \cdot N_{tx} \cdot E_{tx} + E_{rx} \cdot \sum_{i=1}^{K-1} N_{i+1} + E_{rx} \cdot \sum_{i=1}^{K-1} \frac{N_{i+1}}{i+1} \\ &> K \cdot N_{tx} \cdot E_{tx} + (K-1) \cdot \frac{3}{2} \cdot N_{rx} \cdot E_{rx} \\ &= K \cdot N_{tx} \cdot (E_{tx} + E_{rx}) + \left(K - \frac{3}{2} \right) \cdot E_{rx} \\ &\geq K \cdot N_{tx} \cdot (E_{tx} + E_{rx}) + 0.5 \cdot E_{rx}, \ \ when K \geq 2 \end{aligned} \tag{4.17}$$

Table 4.3 (in bracket) shows the number of messages transmitted by all the protocols by a single sensor. Using Table 4.3 and Equation (4.17), as shown in Table 4.5,

we can estimate the energy consumption (in mJ) of each protocol for a sensor network consisting of 10. Table 5 shows that the Adaptive protocol decreases energy consumption from 6x to 16x in comparison with the RFIDImpulse protocol, and 2.4x and 1.95x in comparison with Non-uniform and Periodic protocols respectively.

Table 4.5: Factor reduction in energy consumption for a 10-node sensor network

	Adaptive	Non-uniform	Periodic	RFIDImpulse
10m	6.16	2.54	3.15	1.29
30m	10.2	5.31	3.59	1.12
55m	16.31	6.71	8.33	1

Study of data collection resilience for all protocols: The percentage of users who receive at least one packet while traversing sensors transmission range is defined as data collection reliability. We now extend this to the network wide reliability that we call data collection resilience, as the percentage of users who receive at least one packet from every sensor node. RFIDImpulse protocol has the highest resilience because a sensor wakes up and unicasts data when it receives RFID signal from a user. Suppose K is the number of sensor nodes and r_a, r_u, and r_p denote average resilience of adaptive, non-uniform, and periodic protocols respectively and C is the sum of forwarding probabilities over K sensor nodes (ref.Equation (4.18)). K_1 and K_2 represent the distance between the sink node and the farthest node on the left and farthest node on the right respectively ($K_1 + K_2 = K$). Since C is always less than K, the data collection resilience of adaptive protocol is always larger than the non-uniform protocol. Then, we can estimate the data collection resilience of all the protocols as shown in Table 4.6. We denote d as the distance between the static sink, $K_1 + K_2 = K$.

Table 4.6: Data collection resilience comparison across all the protocols

Adaptive	Non-uniform	Periodic	RFIDImpulse
r_a	$(r_u \cdot C)/K$	r_p	100

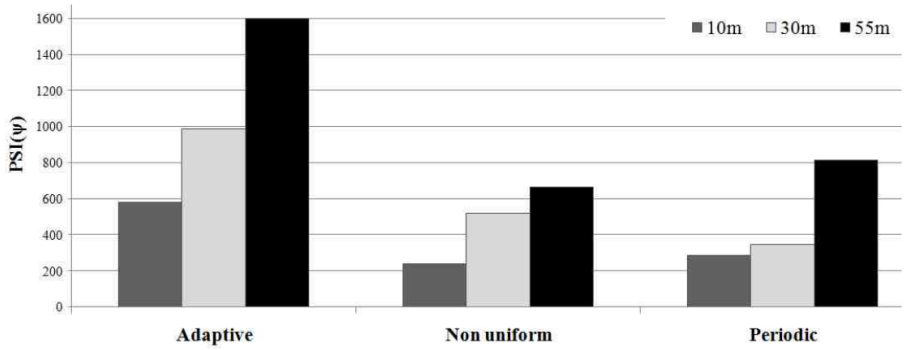

Figure 4.6: Ψ comparison across all the protocols

$$C = \sum_{i=1}^{K_1} 1/i + \sum_{j=1}^{K_1} 1/j \tag{4.18}$$

We define parameter, Ψ, as the product of factor reduction in energy consumption and data collection resilience. The higher value of Ψ, better the performance of the protocol. Figure 4.6 shows that Ψ of the proposed protocol is 2x and 5x in comparison with Periodic and Non-uniform protocols respectively.

4.6 Conclusion

In this thesis we present a novel distributed low-power protocol that lets users, while moving around the deployed sensor network, collect information from the sensors in their vicinity. As users mobile acknowledge the data, the sensors estimated user travel time and use it as a statistical basis for dynamically adjusting their data broadcast rate. We compared our approach with state-of-the-art protocols using realistic mobility models and radio propagation model. The results show a decrease of between in power consumption of 2x to 8x (single sensor) & 2x to 16x (network-wide) when compared to the existing protocols while reliability is only 1% lower than other protocols for three different transmission ranges.

Chapter 4, in full, is a reprint of the material as it appears in ” A novel protocol

for adaptive broadcasting of sensor data in urban scenarios", by Jinseok Yang, Sameer Tilak and Tajana Simunic Rosing, IEEE Globecom 2013. The dissertation/thesis author was the primary investigator and author of this thesis.

Chapter 5

Summary and Future work

5.1 Thesis summary

We designed three methods to decrease the energy consumption in WSNs while guaranteeing a level of QoS to win. We first proposed integrated power management framework that optimizes interactions among power management components. User interactive power manager optimally schedules battery usage patterns based on types of user request. Then, we proposed transmission managers both for a single sensing platform and multi-hop sensor networks. The optimal transmission manager (OptTM) calculates the optimal transmission time for measurements with different timing requirements. OptTM supports multiple applications in a single hop wireless sensor networks. Distributed transmission manager (DistTM) leverages OptTM to operate in multi-hop WSNs. Lastly, we design a novel broadcast mechanism that capable of seamlessly providing information to mobile users while minimizing communication energy costs. An overview of our contributions is next. After that we provide some ideas for future work.

5.1.1 Adaptive power management framework

We present a power management framework that adapts sampling & transmission rates based on battery capacity level, harvesting energy amount and application context. Our interactive power management adapts the sampling rate as a function of both application-level context (e.g., user request) and system-level context (e.g harvesting energy availability). Proposed approach saves 20% to 60% of energy consumption as compared to the state of the art approaches while having at most 8.6% of measurements that expire.

5.1.2 Adaptive transmission manager

We design optimal and distributed transmission managers that dynamically adjust transmission time based on the time constraint of measurements. The optimal transmission manager characterizes energy and delay trade-off and finds the optimal transmission time by using Markov Decision Process model for any two nodes. Then, we leverage structured monotone rule to decrease the complexity of the algorithm to $O(n)$. Distributed transmission manager calculates upper bound of the transmission time based on the distance to the sink. The two transmission managers work in tandem to determine the transmission time for all buffered measurements. Our proposed approach consumes on average 148.3% less energy than the other approaches while expiring on average 14.1% less measurements.

5.1.3 Adaptive information dissemination protocol

We present a novel distributed low-power protocol that lets users collect information from the sensors in their vicinity while moving around the deployed sensor network. As mobile users acknowledge the data, the sensors estimate user travel time and use it

as the statistical basis for dynamically adjusting their broadcast data rate. Each sensor node selects the smallest sampling rate based on estimated users mobility thereby saving a significant amount of energy. The results show that our proposed approach decreases power consumption by a factor of 2x to 8x in a single sensor, and 2x to 16x in 10 node sensor network as compared to the state of art approaches.

5.2 Future work directions

A sensor node power manager controls various system parameters to increase the energy efficiency. One of the main parameters controls the communication cost. Previous approaches decrease communication cost with synchronization, buffering and reliable routing path construction.

Recently, Internet Of Things (IoT) has emerged. It extends the scope of WSNs from information provider to both information provider & consumer. This rises a new challenges for WSNs. Sensor nodes need to extract context from collected measurements and adjust performance of power manager accordingly. This requires new machine learning algorithms that can operate in WSNs and interface between the learning algorithm and the power manager.

Power managers achieve energy savings by decreasing the amount of data to be transmitted and measured. More specifically, sensors on a sensing platform may decrease the sampling rate and transmit only representative values in order to save on communication energy. Previous approaches such as compressive sensing and reconstruction algorithms support this by reconstructing missing elements from the measurements.

Compressive sensing has been actively researched. It can reconstruct 90% of missing data from 10% of measurements. However, compressive sensing requires complex math, so it cannot operate on low-powered sensor nodes. In addition, compressive

sensing assumes that all datasets have a fixed rank parameter which is not true in real WSNs.

To address these issues, light weight data reconstruction algorithms need to be created that can perform as well as compressive sensing but can run on smaller computation infrastructure. Collaborative filtering algorithms can be leveraged to solve this problem because they are relatively light weight. Collaborative filtering is an essential tool to extract hidden interactions among measurements, so that reconstruction engine can interact with the power manager more efficiently.

Bibliography

[1] Creon, the coral reef environmental observatory network. http://www.coralreefeon.org.

[2] Dcm005 batter specification. https://www.interstatebatteries.com/content/product_info/specs/dcm0055.pdf.

[3] Gleon, the global lake ecological observatory network. http://www.gleon.org.

[4] Instapark solar panel. http://www.instapark.com/solar-power-panels.

[5] Noth template lake ecological study. http://lter.limnology.wisc.edu.

[6] Us climate reference network databaset. http://www.ncdc.noaa.gov/crn/qcdatasets.html.

[7] Raspberry pi2, 2015.

[8] P. Aghera, J. Yang, P. Zappi, D. Krishnaswamy, A. K. Coskun, and T. S. Rosing. Energy management in wireless mobile systems using dynamic task assignment. In *Journal of Low Power Electronics*, volume 9, pages 198–217, Aug 2013.

[9] B. Aksanli, T. Rosing, and E. Pettis. Distributed battery control for peak power shaving in datacenters. In *Green Computing Conference (IGCC), International*, pages 1–8, June 2013.

[10] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri. An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. volume 59, pages 335–344, Feb 2010.

[11] R. Arroyo-Valles, A. Marques, and J. Cid-Sueiro. Optimal selective transmission under energy constraints in sensor networks. volume 8, pages 1524–1538, Nov 2009.

[12] R. Arroyo-Valles, A. Marques, and J. Cid-Sueiro. Optimal selective forwarding for energy saving in wireless sensor networks. volume 10, pages 164–175, January 2011.

[13] B. Bougard, F. Catthoor, D. Daly, A. Chandrakasan, and W. Dehaene. Energy efficiency of the ieee 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives. In *Design, Automation and Test in Europe. Proceedings*, pages 196–201 Vol. 1, March 2005.

[14] M. Bouzeghoub. A framework for analysis of data freshness. In *Proceedings of the International Workshop on Information Quality in Information Systems*, IQIS '04, pages 59–67, New York, NY, USA, 2004. ACM.

[15] B. Buchli, F. Sutton, J. Beutel, and L. Thiele. Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys, pages 31–45, New York, NY, USA, 2014. ACM.

[16] A. Cammarano, C. Petrioli, and D. Spenza. Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), IEEE 9th International Conference on*, pages 75–83, Oct 2012.

[17] CC2630. Zigbee wireless mcu. `http://www.ti.com/product/CC2630`.

[18] CCIMIS-WSN. California irrigation management information system. `www.cimis.water.ca.gov`, 2013.

[19] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. volume 31, pages 20–41, New York, NY, USA, Apr. 2001. ACM.

[20] M. Chang and P. Bonnet. Meeting ecologists' requirements with adaptive data acquisition. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys, pages 141–154, New York, NY, USA, 2010. ACM.

[21] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun. Distributed collaborative control for industrial automation with wireless sensor and actuator networks. volume 57, pages 4219–4230, Dec 2010.

[22] T. E. Choe, M. W. Lee, and N. Haering. Traffic analysis with low frame rate camera networks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Computer Society Conference on*, pages 9–16, June 2010.

[23] D. Doerffel and S. A. Sharkh. A critical review of using the peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries. volume 155, pages 395 – 400, 2006.

[24] A. El-Hoiydi and J.-D. Decotignie. Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *Proceedings of Computers and Communications, ISCC*, volume 1, pages 244–251, June 2004.

[25] V. Erickson, M. Carreira-Perpin, and A. Cerpa. Observe: Occupancy-based system for efficient reduction of hvac energy. pages 258–269, 2011.

[26] M. Gorlatova, A. Wallwater, and G. Zussman. Networking low-power energy harvesting devices: Measurements and algorithms. In *INFOCOM, Proceedings IEEE*, pages 1602–1610, April 2011.

[27] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke. Smart grid technologies: Communication technologies and standards. volume 7, pages 529–539, Nov 2011.

[28] M. Haque, N. Matsumoto, and N. Yoshida. Context-aware multilayer hierarchical protocol for wireless sensor network. In *Sensor Technologies and Applications, SENSORCOMM. Third International Conference on*, pages 277–283, June 2009.

[29] G. He. Destination-sequenced distance vector (dsdv) protocol, May 06, 2002.

[30] D. Helbing, P. Molnar, I. Farkas, and K. Bolay. Self-organizing pedestrian movement. In *Environment and Planning B, Planning and Design*, 2001.

[31] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys, pages 81–94, New York, NY, USA, 2004. ACM.

[32] R. Jurdak, A. Ruzzelli, and G. O'Hare. Radio sleep mode optimization in wireless sensor networks. volume 9, pages 955–968, July 2010.

[33] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. volume 6, New York, NY, USA, Sept. 2007. ACM.

[34] J. Kho, A. Rogers, and N. R. Jennings. Decentralized control of adaptive sampling in wireless sensor networks. volume 5, pages 19:1–19:35, New York, NY, USA, June 2009. ACM.

[35] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[36] B. Koo, J. Won, S. Park, and H. Eom. Paar: A routing protocol for context-aware services in wireless sensor-actuator networks. In *Internet, AH-ICI. First Asian Himalayas International Conference on*, pages 1–7, Nov 2009.

[37] P. Levis and D. Culler. The firecracker protocol. In *Proceedings of the 11th Workshop on ACM SIGOPS European Workshop*, EW 11, New York, NY, USA, 2004. ACM.

[38] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA, pages 88–97, New York, NY, USA, 2002. ACM.

[39] S. Mao, M. H. Cheung, and V. Wong. An optimal energy allocation algorithm for energy harvesting wireless sensor networks. In *Communications (ICC), IEEE International Conference on*, pages 265–270, June 2012.

[40] L. Martirano. A smart lighting control to save energy. In *Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), IEEE 6th International Conference on*, volume 1, pages 132–138, Sept 2011.

[41] T. Muetze, P. Stuedi, F. Kuhn, and G. Alonso. Understanding radio irregularity in wireless networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, SECON, 5th Annual IEEE Communications Society Conference on*, pages 82–90, June 2008.

[42] V. Naik, A. Arora, P. Sinha, and H. Zhang. Sprinkler: A reliable and energy efficient data dissemination service for extreme scale wireless networks of embedded devices. volume 6, pages 777–789, July 2007.

[43] K.-S. Ng, Y.-F. Huang, C.-S. Moo, and Y.-C. Hsieh. An enhanced coulomb counting method for estimating state-of-charge and state-of-health of lead-acid batteries. In *Telecommunications Energy Conference, INTELEC. 31st International*, pages 1–5, Oct 2009.

[44] N. Nikzad, J. Yang, P. Zappi, T. Rosing, and D. Krishnaswamy. Model-driven adaptive wireless sensing for environmental healthcare feedback systems. In *ICC, IEEE International Conference on*, pages 3439–3444, June 2012.

[45] ns3. Discrete-event network simulator. `http://www.nsnam.org/`, 2015.

[46] Hungarian method. `http://en.wikipedia.org/wiki/Hungarian_algorithm/`, 2015.

[47] J. Piorno, C. Bergonzini, D. Atienza, and T. Rosing. Prediction and management in energy harvested wireless sensor nodes. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, Wireless VITA. 1st International Conference on*, pages 6–10, May 2009.

[48] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of International Conference on Embedded Networked Sensor Systems*, SenSys, pages 95–107, New York, NY, USA, 2004. ACM.

[49] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[50] K. S. Rachel Cardell-Oliver, Mark Kranz and K. Mayer. A reactive soil moisture sensor network: Design and field evaluation. In *International Journal of Distributed Sensor Networks*, volume 1, pages 149–162, 2005.

[51] D. Rakhmatov, S. Vrudhula, and D. Wallach. Battery lifetime prediction for energy-aware computing. In *Low Power Electronics and Design, ISLPED. Proceedings of the International Symposium on*, pages 154–159, 2002.

[52] M. N. Richard L. Knoblauch, Martin T. Pietrucha. Field studies of pedestrian walking speed and start-up time. In *Transportation Research Record: Journal of the Transportation Research Board*, pages 27–38 Vol. 1538, January 1996.

[53] A. S. Sahaleh, M. Bierlaire, B. Farooq, A. Danalet, F. S. Hnseler, A. S. Sahaleh, M. Bierlaire, B. Farooq, A. Danalet, and F. S. Hnseler, 2012.

[54] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 30–41, May 2003.

[55] I. Solis and K. Obraczka. In network aggregation trade offs for data collection in wireless sensor networks. In *International Journal of Sensor Networks*, volume 1, pages 200–212, Inderscience Publishers, Geneva, SWITZERLAND, Jan. 2006.

[56] S. Tilak, A. Murphy, and W. Heinzelman. Non-uniform information dissemination for sensor networks. In *Network Protocols, 2003. Proceedings. 11th IEEE International Conference on*, pages 295–304, Nov 2003.

[57] K. Tutuncuoglu and A. Yener. Optimum transmission policies for battery limited energy harvesting nodes. volume 11, pages 1180–1189, March 2012.

[58] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal. Energy-efficient packet transmission over a wireless link. volume 10, pages 487–499, Aug 2002.

[59] S. H. W. Daamen. Flow-density relations for user traffic, 2005.

[60] S. H. W. Daamen. Free speed distributions-based on empirical data in different traffic conditions, 2005.

[61] Q. Wang, P. Fan, D. Wu, and K. Ben Letaief. End-to-end delay constrained routing and scheduling for wireless sensor networks. In *IEEE Communications (ICC)*, pages 1–5, June 2011.

[62] X. Wang, X. Wang, L. Liu, and G. Xing. Dutycon: A dynamic duty-cycle control approach to end-to-end delay guarantees in wireless sensor networks. volume 9, pages 42:1–42:33, New York, NY, USA, July 2013. ACM.

[63] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. volume 22, pages 26–33, July 2008.

[64] WXT520. Weather station. `http://www.vaisala.com/en/products/multiweathersensors`.

[65] L. Xiang, J. Luo, and A. Vasilakos. Compressed data aggregation for energy efficient wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, pages 46–54, June 2011.

[66] X. Xu, J. Luo, and Q. Zhang. Delay tolerant event collection in sensor networks with mobile sink. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

[67] Z. Xu, W. Liang, and Y. Xu. Network lifetime maximization in delay-tolerant sensor networks with a mobile sink. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pages 9–16, May 2012.

[68] J. Yang, T. Rosing, and S. Tilak. An interactive context-aware power management technique for optimizing sensor network lifetime. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2013 IEEE Ninth International Conference on*, 2013.

[69] J. Yang, R. T.S, and S. Tilak. Leveraging application context for efficient sensing. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6, April 2014.

[70] Z. Ye, A. Abouzeid, and J. Ai. Optimal stochastic policies for distributed data aggregation in wireless sensor networks. volume 17, pages 1494–1507, Oct 2009.

[71] H.-Y. Zhou and K.-M. Hou. Civic: An power- and context-aware routing protocol for wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 2771–2774, Sept 2007.