ROLL-CALL: AN ENERGY EFFICIENT RADIO FREQUENCY IDENTIFICATION SYSTEM

BY SHWETA MEDHEKAR

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Wade Trappe

and approved by

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE THESIS

Roll-Call: An Energy Efficient Radio Frequency Identification System

by Shweta Medhekar

Thesis Director: Professor Wade Trappe

In this thesis, we investigate two of the major challenges in pervasive systems: energy efficiency and co-existence of uncoordinated wireless messages by exploring the design of a Radio Frequency Identification ($RFID$) system intended to support the simultaneous and real time monitoring of thousands of entities. These entities, which may be individuals or inventory items, each carry a low-power transmit-only tag and are monitored by a collection of networked base-stations reporting to a central database. We have built a customized transmit-only tag with a small form-factor, and have implemented a real-time monitoring application intended to verify the presence of each tag in order to detect potential disappearance of a tag (perhaps due to item theft). Throughout the construction of our system, we have carefully engineered it for extended tag lifetime and reliable monitoring capabilities in the presence of packet collisions, while keeping the tags small and inexpensive.

The major challenge in this architecture (called Roll-Call) is to supply the energy needed for long range continuous tracking for a year or more of reporting once a second while keeping the tags (called $PIPs$) small and inexpensive. We have used this as a model problem for optimizing cost, size and lifetime across the entire pervasive, persistent system from firmware to protocol.

Acknowledgements

It is a pleasure to thank everyone who made my thesis possible. It is difficult to overstate my gratitude for Dr. Rich Howard who, with his enthusiasm, inspiration, and great effort to explain things clearly and simply, made working fun and interesting for me. This thesis owes the most to his creativity.

I would like to sincerely thank Dr. Wade Trappe, for providing valuable guidance, constructive feedback and suggestions during the course of my Masters research. I also have sincere gratitude to Dr. Yanyong Zhang, Dr. Eitan Fenson and Peter Wolniansky for their inspiring guidance and expertise throughout the thesis work. Sincere thanks is due to Dr. Dipankar Raychaudhuri for his guidance and discussions related to this research. I am grateful my fellow WINLAB students Gautam Bhanage and Yu Zhang for all their help. Additionally, I want to express my gratitude to my friends (especially my roommates) for their camaraderie, entertainment, caring and patience.

Finally, I thank my family for all that they did for me.

Dedication

To Liza

Table of Contents

List of Tables

List of Figures

List of Abbreviations

RFID	Radio Frequency Identification
TDMA	Time Division Multiple Access
MAC	Multiple Access Control
PIP	Persistent Identification Packet
GAF	Geographic Adaptive Fidelity
STEM	Sparse Topology and Energy Management
DVS	Dynamic Voltage Scaling
LEACH	Low-Energy Adaptive Clustering Hierarchy
FDMA	Frequency Division Multiple Access
CDMA	Code Division Multiple Access
PAN	Personal Area Networks
PDA	Personal Digital Assistant
S-MAC	Sensor Multiple Access Control
PS	Power Saving
CSMA	Carrier Sense Multiple Access
ID	Identity
PAMAS	Power-Aware Multi-Access Protocol with Signaling
TSMA	Time-Spread Multiple-Access
IDE	Integrated Development Environment
RSSI	Received Signal Strength Indicator
CRC	Cyclic Redundancy Code
MSK	Minimum Shift Keying
SPP	Simple Packet Protocol
MCU	Microcontroller Unit
RF	Radio Frequency
SCLK	Serial Clock
SYSCLK	System Clock

Chapter 1

Introduction

1.1 Overview of the Problem

Recent advances in wireless technologies have made pervasive computing systems consisting thousands of radio devices, capable of collecting and reporting data (c.f. sensor networks) to support monitoring applications, or able to announce their presence in order to facilitate new types of applications [1][2] feasible. However, in spite of the advancements in the radio circuitry itself, there are two notable challenges that arise as one moves towards pervasive systems involving thousands of radio devices located in relatively confined spaces: energy efficiency and co-existence.

Both problems have been studied extensively in the past decade and a quick perusal of the literature reveals numerous papers on lifetime management and multiple access techniques. Generally, papers on lifetime management have focus on network-oriented issues whereby communicating entities adjust their duty cycling according to the presence of redundant transmitters [3][4] or, as in cluster based systems, the network is divided into clusters with each cluster having a long-haul communication gateway which may employ Time Division Multiple Access ($TDMA$) [5]. Similarly, there are many papers devoted towards multiple access issues [6][7][8] and, for the most-part, these papers assume that transmitters also have some form of receiver so as to facilitate separating the transmissions of different transmitters from each other. Quite notably, in the area of Radio Frequency Identification ($RFID$) systems, several well-known receiver-driven Multiple Access Control (MAC) schemes, such as tree-walking [9][10][11], have been proposed, while for networked sensor systems there are several schemes that employ carrier-sensing to avoid collisions in transmissions.

For many envisioned pervasive systems applications, however, it is not possible for the transmitters to interact with each other, or to receive communications so as to coordinate their transmissions for lifetime optimization or in order to better share the wireless medium. Rather, for many applications, such as inventory monitoring, the driving factor is ultimately the cost of the transmitter, and the solutions presented by receiver-oriented scheduling are unreasonable for these systems. For these pervasive systems, where thousands of transmit-only tags are actuating the environment and are observed by a separate monitoring infrastructure (e.g. to detect whether an item has been stolen), the challenges of lifetime maximization and co-existence must be addressed using different techniques. Notably, lifetime management must be carefully addressed at the individual transmitter by tightly optimizing the software running on each transmitter, while co-existence must instead be addressed by the monitoring infrastructure.

In this thesis, we investigate both of these performance issues by exploring the design of a $RFID$ system intended to support the monitoring of thousands of entities. These entities, which may be individuals or inventory items, each carry a low-power transmit-only tag and are monitored by a collection of base-stations. We have built a customized transmit-only tag with a small form-factor, which we affectionately call PIP (an acronym for Persistent Identification Packet), and have implemented a real-time monitoring application intended to verify the presence of each tag in order to detect potential disappearance of a tag (perhaps due to item theft). Throughout the construction of our system, we have carefully engineered it for extended tag lifetime while also having reliable monitoring capabilities, yet keeping the $PIPs$ small and inexpensive.

The major challenge in Roll-Call is to supply power needed for long range continuous tracking for a year or more while keeping the $PIPs$ small and inexpensive. We will use this as a model problem for optimizing cost, size and lifetime across the entire system from firmware to protocol.

1.2 Thesis Organization

We begin with Chapter 2 by presenting an overview of systems involving thousands of radio devices located in relatively confined spaces and aimed at being energy efficient while having reliable monitoring capabilities.

Chapter 3 is an overview of our system, as well as a specification for our customized transmitter. In Chapter 4, we focus on the challenge of extending the lifetime of the transmitter by identifying causes for needless energy expenditure, and carefully optimizing the hardware and code running on the transmitter. Then, in Chapter 5, we examine the issue of transmission collisions in our system to better understand packet collision dynamics and their effect on the information transfer from tag to base-station.

Finally, we conclude the thesis in Chapter 6 and give directions for future work.

Chapter 2

Related Work

Our *RFID* system involves deploying and monitoring thousands of relatively inexpensive radio devices in relatively confined spaces. This involves managing the lifetime and wireless medium contention of these devices. Both of these issues have been studied extensively in the past decade which are described in this section.

2.1 Lifetime Management

There are numerous papers addressing lifetime management for pervasive systems. In *SPAN* [12] a limited set of nodes forms a multi-hop forwarding backbone, which tries to preserve the original capacity of the underlying ad hoc network. Other nodes transition to sleep states more frequently, as they no longer carry the burden of forwarding data of other nodes. To balance out energy consumption, the backbone functionality is rotated between nodes, and as such there is a strong interaction with the routing layer.

Geographic Adaptive Fidelity (*GAF*) [13] exploits the fact that nearby nodes can perfectly and transparently replace each other in the routing topology. The sensor network is subdivided into small grids, so that nodes in the same grid are equivalent from a routing perspective. At each point in time, only one node in each grid is active, while the others are in an energy-saving sleep mode. Substantial energy gains are, however, only achieved in very dense networks.

STEM [3][4] assumes that the nodes are mostly in a 'monitoring' state and not in the 'transfer' state. Hence, the monitoring state must be an ultra-low power one. It is assumed that the network probably needs to transition to the 'transfer' state periodically to exchange network management and maintenance messages. When a possible event is detected, the main processor is woken up to analyze the data in more detail. The radio, which is normally turned off, is only woken up if the processor decides that the information needs to be forwarded to the data sink. Now, the problem is that the radio of the next hop in the path to the data sink is still turned off if it did not detect that same event. As a solution, each node periodically turns on its radio for a short time to listen if someone wants to communicate with it. The node that wants to communicate, the 'initiator node', sends out a beacon with the ID of the node it is trying to wake up, called the 'target node'. In fact, this can be viewed as the initiator node attempting to activate the link between itself and the target node. As soon as the target node receives this beacon, it responds to the initiator node and both keep their radio on at this point. If the packet needs to be relayed further, the target node will become the initiator node for the next hop and the process is repeated. Once the link between nodes is activated, data is transferred using any MAC protocol. This MAC protocol is only used in the transfer state as even the most efficient one would consume a lot more energy than the proposed low-power listen mode. The reason for this is MAC protocols are designed to organize access to the shared medium, in addition to contacting nodes. The strategy here is thus to decouple the transfer and wake-up functionalities but the price payed is an increase in the latency.

In [14], the focus is on strategically placing sensor nodes and finding an optimal path using Dijkstra's to optimize energy saving. The Cougar project [15] investigates a database approach to sensor networks. Energy-efficient data dissemination and query processing is achieved based on user queries. Passos et al. [16] propose an application driven energy saving technique.

For embedded systems [1] shutting off devices that are not operating has been suggested. Minimum hop communication is needed to reduce the number of transmissions and hence the power consumed. A power-efficient routing algorithm is used that evenly distributes node utilization and communication across the network. The assumption here is that, for every path, there exists a node that has the highest energy consumption rate. Hence, given a graph for a network and a specific traffic pattern, the objective is to route the packets so that after the completion of packet routing, the maximum traffic across all nodes is minimized to enhance system lifetime.

Hernandez et al. propose software profiling in a real-time environment to evaluate the minimum frequency needed to run a particular function [17]. The intent is to program the processor's clock generation subsystem to provide this frequency at the lowest voltage possible. The power sensitive parameters are tracked and controlled by the software. Once the system needs to make a function transition, the parameters are reevaluated and an idle power state is programmed by a destructor function. Destructor function is designed to return the supply voltage level to a safe maximum, so that the next function called will be able to begin its process with enough voltage for continued execution. The constructor for the next function will then create a new power control object for that function. The system then assumes a new power level.

Dynamic Voltage Scaling (DVS) [18][19] is used to achieve maximum energy savings. This analysis is based on the fundamental power dissipation equation for digital semiconductor products where, reducing the operating voltage or frequency, or both, can result in lowering overall system power consumption. Pedram [2] uses selective shutoff or slow-down of system components that are idle or under-utilized and DVS. Instruction rescheduling is a common compiler technique for improving the code performance. Without affecting the correctness, reordering instructions can eliminate pipeline stalls caused by load delay, branch delay, delay slot, etc which is very effective for saving power. Many bus-encoding schemes have been proposed. The encoding function can be optimized for specific access patterns such as sequential access. This minimizes power losses in bus access. Hill and Culler in [20] give a system architecture and design decisions for MICA sensor nodes to achieve system-level optimization.

2.2 Low Power Multiple Access Techniques

A wireless network consists of a large number of nodes equipped with embedded processors, sensors and radios. In a wireless communication packet collisions, overhearing and channel sensing or idle listening is bound to occur. There are some form of control packets being transmitted. A Medium Access Control (MAC) protocol achieves energy savings by controlling the radio to avoid or reduce energy waste from the above sources. Turning off the radio when it is not needed is an important strategy for energy conservation.

For cluster based systems [5][21] which may employ $TDMA$, the network is divided into clusters and each cluster has a gateway. Clusters are formed such that a node's gateway is located within the communication range of all of its cluster sensors. Gateways use long-haul communication to send reports fused from its cluster sensor data to other gateways, and eventually to the command node. The sensing circuits as well as radio transmitters and receivers can be turned on and off independently. Transmission power can be adjusted based on the required range. Sensors can act as store-and-forward relay nodes. Gateways are responsible of the dynamic configuration of the sensor network within its cluster. Routes for sensor data from the active sensor node to the gateway are set in order to optimize an objective function, i.e. a path optimization problem, which is known to be of polynomial time complexity based on Dijekstra's algorithm. The problem here is viewed as the transpose of a single-source routing algorithm, i.e. single destination routing. Routing setup can be dynamically adjusted to optimally respond to changes in the sensor organization. Re-routing decisions are based on three criteria: sensor reorganization such as an event that requires reselection of active sensors, node battery energy level if it drops to a certain level, and energy model adjustment after refresh updates. Low-Energy Adaptive Clustering Hierarchy ($LEACH$), proposed by Heinzelman et al. [22] is an example of utilizing $TDMA$ in wireless sensor networks.

Sohrabi and Pottie proposed a self-organization protocol for wireless sensor networks [23]. The protocol assumes that multiple channels are available (via $FDMA$ or $CDMA$), and any interfering links select and use different sub-channels. During the time that is not scheduled for transmission or reception, a node turns off its radio to conserve energy. Each node maintains its own time slot schedules with all its neighbors, which is called a $superframe$. Time slot assignment is only decided by the two nodes on a link, based on their available time. It is possible that nodes on interfering links will choose the same time slots. This protocol supports low-energy operation, but a disadvantage is the relatively low utilization of available bandwidth.

Bluetooth [24][25] is a communication protocol designed for Personal Area Networks

(PAN) where the target nodes are items such as battery-powered $PDAs$, cell phones and laptop computers. It is designed for low-energy operation and is inexpensive.

S-MAC [6][26] is an efficient protocol developed for sensor networks based on 802.11(PS) with energy efficiency being its primary aim. They have a complex sleep / wake cycle with low duty cycle. It overcomes high latency by using adaptive listening. Since it is a contention based protocol, it is flexible to topology changes. S-MAC allocates resources according to demand and hence is more flexible to traffic demands. Peer-to-peer communication is supported.

Woo and Culler proposed a MAC protocol [27] for wireless sensor networks, which combined $CSMA$ with an adaptive rate control mechanism. Piconet is a low-power ad hoc wireless network developed by Bennett et al. [28]. The basic MAC protocol used in Piconet is the 1-persistent $CSMA$ protocol. To reduce energy consumption, each node sleeps autonomously. Since nodes do not know when their neighbors are listening, they beacon their ID each time they wake up. Neighbors with data for a particular destination must listen until they hear the destinations beacon. They then coordinate using $CSMA$. $PAMAS$ [29] avoids overhearing by putting nodes into sleep state when their neighbors are in transmission. This is an extension to 802.11(PS).

$Sift$ [7] is a randomized $CSMA$ protocol developed for sensor networks. Sift uses a fixed-size contention window and a carefully-chosen, non-uniform probability distribution of transmitting in each slot within the window. Assumptions for the protocol are: sensing is spatially correlated, not all sensing nodes need to report an event and the size of the sensing field changes with time. $Sift$ makes sure that there is low latency and transmission is contention-free for the first few successful transmission slots. The key difference between $Sift$ and previous protocols like 802.11 is that the probability of picking a slot in this interval is not uniform. While S-MAC, $LEACH$, and $PAMAS$ govern medium-access to some degree they do not address the contention portion of a medium-access protocol. Since $Sift$ is a $CSMA$ protocol, it can be implemented concurrently with these protocols.

$TSMA$ [8] policies that have been proposed for ad hoc networks can also be used with sensor networks. This will remove the control overhead. Under the original $TSMA$

policy, nodes are allowed to transmit only at a (small) subset of the available time slots carefully selected so that at least one of them is collision free. The achieved throughput of this particular deterministic policy was shown that it could be further improved by allowing probabilistic transmission(P-policy) attempts during unallocated time slots that were not assigned under the deterministic assignment. For sensor nets the P-policy is further modified to capitalize on the characteristics of sensor nets and is known as the A-policy.

Chapter 3

System Overview

3.1 General Overview

The objective behind this thesis is to develop a system with the ability to simultaneously monitor a large number of assets. This monitoring should not be done one-by-one but all at once. In contrast, *RFID* today is partial asset tracking using hand-held readers or a portal. It tracks items one-by-one working as an enhanced form of barcode. The communication protocol used is query-response [30][31]. Hence, there is no overall visibility of the environment at any single instant. Ideally each object should say 'I am here' at all times for overall visibility. Thus our objective is full inventory tracking in real time, and for this reason we call our system 'Roll-Call'.

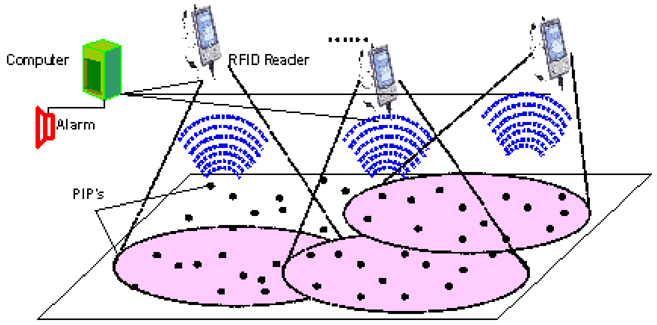


Figure 3.1: Roll-Call System.

A typical *RFID* system is composed of the *RFID* transmitter or transponder (tag) and the *RFID* reader or base-station. Existing *RFID* systems are of two types. The first one where the objects being tracked need cheap, small tags with many reads and a long life. Such tags tend to be passive and have a short range and are powered by high powered base-stations using backscatter coupling [32]. Because of reflected power passive tags use, reliability and range is compromised in the field. The second system is where the objects being tracked can afford large expensive tags. They have their own power supply on-board which makes them bulky. They incorporate many reads, long life and long range. These are systems with active tags. Our goal in this research is to construct our own tag by combining the best of both active and passive tags, i.e. small size, many reads, long range, long life.

The major challenge in Roll-Call is to supply power needed for long range continuous tracking for a year or more while keeping the $PIPs$ small and inexpensive. We will use this as a model problem for optimizing cost, size and lifetime across the entire system, from firmware to protocol. The protocol that we have devised aims at tags transmitting a repetitive beacon signal with a short predefined interval between two beacons. There is no feedback from the base-station to the tags.

It is necessary that the tags be very cheap. They should be much much cheaper than the item they are tagging. This will be achieved if they have low complexity. The base-stations can be complex as they are deployed in very small numbers. We achieve this goal in Roll-Call by designing an asymmetric system. Tags are transmit only and they are complemented with complex base-stations. This makes the overall system cheap. Hence, the task of detection, collision recovery and processing is placed on the base-station.

3.2 The Platform

Figure 3.1 shows a general Roll-Call system with $PIPs$ scattered in an enclosure. A number of networked readers are seen tracking the inventory. The networked base-stations are connected to a central computer which performs the necessary computations and generates an alarm signal whenever needed. Figure 3.2 shows a simplified system with one PIP and one base-station. The box on the left is the PIP with its major components. The box on the right is the base-station connected to the back-end computer with an RS-232 link. We are using commercially available components to construct our Roll-Call system. The radio is a $CC1100$ [33] radio chip. This versatile product has a powerful IDE available along with hardware abstraction that makes programming in C relatively easy. All radio parameters, like data-rate, modulation format, output power, filter bandwidth and others are programmable. The microcontroller used is $C8051F32x$ [34]. It has user control on all peripherals, which may be individually configured or shut-off for power saving options.

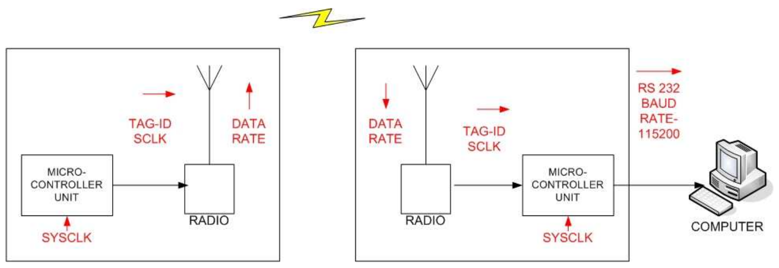


Figure 3.2: Dataflow Diagram.

Our PIP, shown in Figure 3.3, incorporates on-board a $CC1100$ radio, $C8051F32x$ microcontroller, antenna and programming interface. The microcontroller is used to monitor the working of the radio and to load bits into it. The radio is a high sensitivity transceiver. It is powered by a $3.3V$, 220 mA-hr coin battery[35], and has dimensions that are approximately 1 *inch* by 1 1/4 *inches*. The antenna is integrated on the board and is -10 *dbi* at 900 MHz. Flash programming is done through the on-board IDE. The base-station used is a Texas Instruments $CC1100$/$CC1150$-868/915 MHz Development Kit [36]. The base-station uses the same basic components as the tag and will be replaced by a tag in our next generation. There are many other peripherals on the development kit but we don't use them. The main difference is that the $CC1100$ in the base-station is used in receive only mode. The $C8051F32x$ on the base-station sends the data received over the wireless link through a RS-232 interface. The back-end sees the system as a network of base-stations. The back-end module uses java servlets to get data from the networked base-stations and generates log files. The log files include the time-stamp of reception, the Unique ID of the PIP, the sequence number, the received

$RSSI$ and the CRC check status.

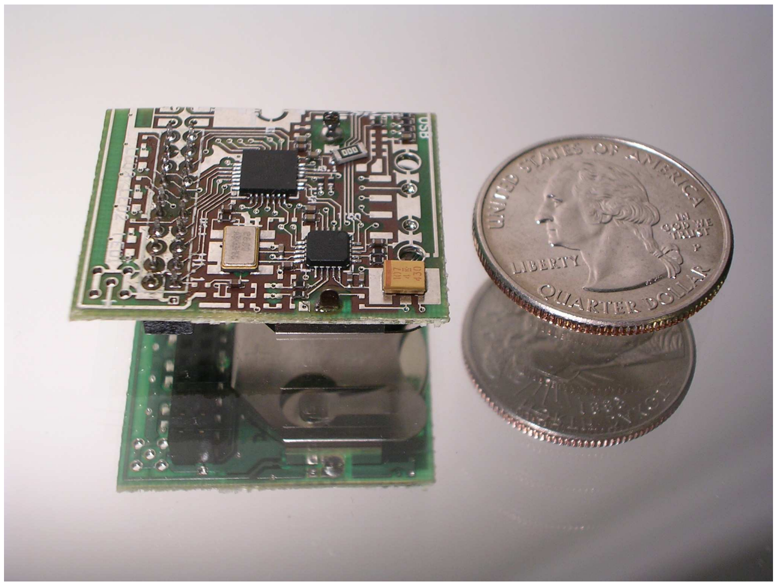


Figure 3.3: PIP.

Each PIP transmits a beacon signal using Minimum Shift Keying (MSK) modulation [37] at 902.1 MHz. MSK is preferred for our application as it is reliable and uses low power. There is a predefined interval between two beacons which depends on the application and we are using 1 sec in our prototype system. The beacon contains the serial ID of the transmitting PIP and the sequence number of the packet being transmitted. The transmission of data takes place using the Simple Packet Protocol (SPP) [33]. The packet structure for SPP is shown in Figure 3.4. A few important fields in the packet are the preamble, followed by the length field, followed by variable size data and finally the 16 bit CRC. The data field contains the unique ID of the transmitting PIP and the sequence number of the packet being transmitted. Sequence number is a 8-bit number which resets to 0 on overflow and resumes counting again. Packets are in error if they fail the CRC check or if certain sequence numbers are not received at the base-station.

The base-station is tuned to the same data-rate and modulation format as the $PIPs$ and the carrier frequency is tuned for minimum transmission error. The base-station demodulates the packets received from different $PIPs$ and stores the serial ID of the

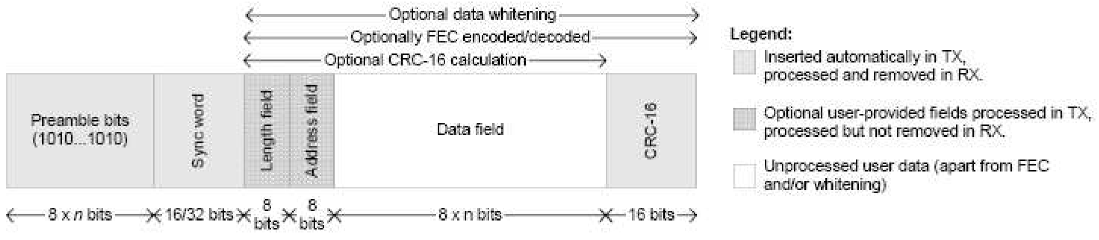

Optional data whitening
Optionally FEC encoded/decoded
Optional CRC-16 calculation
Preamble bits (1010...1010)
Sync word
Length field
Address field
Data field
CRC-16
8 x n bits
16/32 bits
8 bits
8 bits
8 x n bits
16 bits
Legend:
Inserted automatically in TX, processed and removed in RX.
Optional user-provided fields processed in TX, processed but not removed in RX.
Unprocessed user data (apart from FEC and/or whitening)

Figure 3.4: Packet Format used in Simple Packet Protocol.

PIP, the sequence number and the $RSSI$ of the received signal. It computes the CRC checksum and stores whether the packet is in error or not. There is a back-end code continuously running on the base-station which uses RS-232 to extract the raw data from the microcontroller flash and put it in a file on the work-station as well as provides a scrolling display on the console. After all the data collection is done, a java code is run on the stored file, which restructures the data in proper format for easy interpretation. The repetitive rate of the beacons is designed so as to have minimum collisions given the total number of $PIPs$ in the system and even if there are collisions, reception is possible in future periods. This is due to the jitter in the microcontroller clock, which works to our advantage. Conceptually, this is similar to random back-off schemes but because only one packet is sent, we do not need reception acknowledgements.

The system is such that all the $PIPs$ are known to the base-station. If a certain number of packets from a particular PIP are not received at the base-station due to disabling or removal of the item, then the item is said to be missing.

Chapter 4

System Energy Optimization

The goal for Roll-Call project is to have a working system monitoring approximately 10^2 - 10^4 *PIPs*, where each *PIP* should be such that their lifetime is a year and diameter of coverage is approximately 3 *m* when transmitting beacons with 1 *mW* power using a 220 *mA-hr* battery. The *PIPs* will transmit beacons with 1 *sec* between them. Due to continuous transmission of beacons in a high density system there is a probability of packet collisions. If the transmission takes place at 1 *Mbps* then approximately 10^3 - 10^4 can be detected with tolerable error rates [38] using signal processing at the base-station and knowledge of the expected signal. However, with the current version of hardware transmissions take place at 0.25 *Mbps* and the receiver does not allow access to the *RF* signals for data processing. This limits us initially to monitoring approximately 100 tags. The ultimate aim here is minimizing the data transmission time and the energy spent for each transmission while maintaining raw error rates of approximately 0.01% or less at the base-station.

4.1 State of the System

Figure 4.1 shows energy consumed by *PIP* during one beacon interval measured by recording the time-dependent current drain. Initially the tag is in a low power sleep state for 1 *sec*. The microcontroller wakes up after a 1 *sec* interval and loads data into the radio. The radio initializes itself and transmission takes place. After the transmission everything shuts down again for 1 *sec*.

In Figure 4.1, we present these states of the *PIP* operation labeled as regions A, B, C, D, E for a single epoch. The system stays in Region A for 1 *sec* which is the interval between the beacons. In this region the microcontroller is in the *idle-mode* and

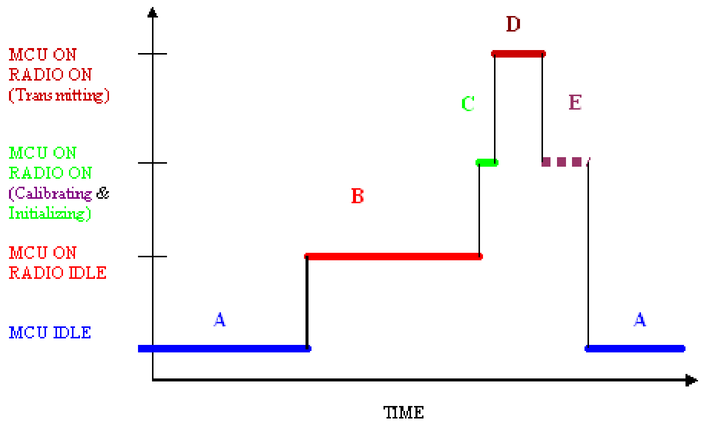


Figure 4.1: State Diagram. Note: Scale not uniform to show regions clearly.

draws minimum possible current of 24 μA from the battery. An external slow clock is makes the microcontroller count time and radio is off. In Region B, the microcontroller moves to the *on-mode* from the *idle-mode*. The radio moves into the *idle-mode*. The microcontroller gets initialized and starts transferring data to the radio using the 4 wire serial interface. The time duration of this region depends on the amount of data to be transmitted and the transfer-bus clock, $SCLK$. Timing is 372 μs for 40 bits of data and 600 μs for 168 bits of data when $SCLK$ is 3 MHz. The current drawn in this region is 4.25 mA. In Region C the microcontroller stays on and the radio moves into the *on-mode* from the *idle-mode*. The radio takes 80 μs to initialize. The current drain here is 10.8 mA. In Region D the actual transmission of data takes place. Both the microcontroller and the radio are on in this region. The time duration again depends on the amount of data to be transmitted. It is 485 μs for 40 bits of data plus the SPP overhead (refer to Figure 3.4) and 800 μs for 168 bits of data plus the SPP overhead bits when transmission data-rate is 250 $kbps$. The current draw is maximum here, at 36.5 mA. In Region E, the radio gets calibrated, which occurs once in 4 cycles to reduce power consumption and this is shown as a dotted line. Both the microcontroller and the radio are on whenever this region occurs. Region E has a duration of 720 μs and a current drain of 9.6 mA.

4.2 System Optimization

We have used a beacon with 168 bits of data plus SPP overhead. This implies 160 bits for the PIP ID, and 8 bits for specifying the buffer size. 160 bits for PIP ID is a very conservative approach as it implies the possibility of having 2^{160} unique $PIPs$ and is larger than the current $RFID$ standard of 96 bits. 160 bits also provides a means for error correction at the receiver. The modulation format used is MSK at a carrier frequency of 902.1 MHz. The battery used in the system is $CR2032$ [35] and is 220 mA-hr in capacity. The maximum data-rate for transmission is selected to be 250 *kbps* to facilitate error free reception. Empirical evidence has shown that transmitting above 300 *kbps* results in inconsistent reception. SPP is used for wireless data transmission at 10 dBm.

Using the system with its default settings results in the battery dying out in half a day. We analyzed the energy usage in each of the regions A, B, C, D, and E shown in Figure 4.1. Changes were incorporated to make the system power efficient and increase its lifetime.

4.2.1 Optimizing Region A

Region A from Figure 4.1 is the interval between beacons, i.e. 1 *sec*. Considering the durations for various regions explained in Section 4.1, it is evident that region A is present for approximately 99.9% of the time. To reduce the power consumption in this region, all the analog peripherals on the microcontroller are shut down. An additional capacitor on-board is enabled as a separate slow clock to run the system during the 1 *sec* interval between beacons. Clocks are switched on the fly for transmission, i.e. the PIP switches to the fast clock 6 MHz during transmission and at other times it works with the slow clock. For prototyping purposes we have used 32 KHz as the slow clock to ensure reliability. Since for the microcontroller system $Power \propto Frequency$, frequency switching reduces the energy consumed by a factor $6000/32 = 187.5$. Eventually several slower clocks can be used with this processor since no significant accuracy is needed for the interval between beacons. The wake-on-radio functionality of $CC1100$ is enabled, which periodically transfers the radio to idle state from the sleep state to check for the signal from the microcontroller timer. If the timer has overflowed it transmits data, else it goes back to sleep. These changes the energy of region A from 49800 $\mu J/pulse$ to 72 $\mu J/pulse$.

4.2.2 Optimizing Region B

Region B is the most complex region, where the tag initializes the microcontroller and transfers data to the radio using the on-board 4-wire serial interface. The energy consumed by this region will depend on the speed of the microcontroller clock ($SYSCLK$), the serial clock ($SCLK$) and the amount of data to be transmitted. For a fixed amount of data, $SYSCLK$ and $SCLK$ can be optimized to achieve a power efficient system. $SCLK$ is derived from $SYSCLK$. $SCLK$ transfers data and control bytes from the microcontroller to the radio. Increasing the clock speeds reduces the time of internal data transmission and initialization at the expense of higher current draw. Hence, clock speed and current have to be balanced to minimize the energy consumed.

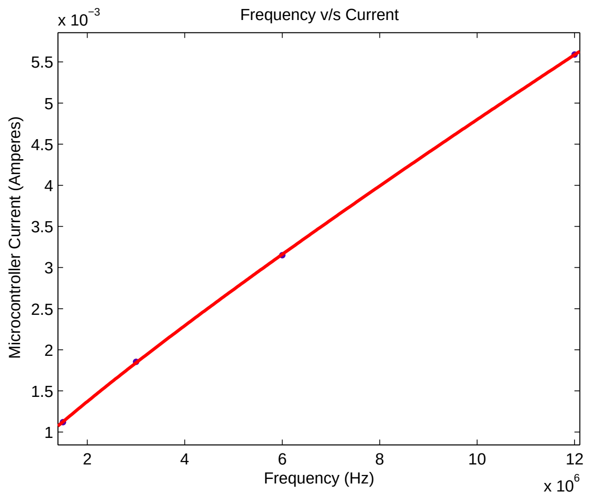


Figure 4.2: Plot of microcontroller current vs the main clock of microcontroller (SYSCLK).

The current drawn by the microcontroller as a function of $SYSCLK$ is shown in

SYSCLK (MHz)	SCLK (MHz)	TIME (ms)	CURRENT (mA)	ENERGY ($\mu J/pulse$)
1.5	0.75	1.43	3	12.87
3	1.5	0.865	3.5	9.083
6	3	0.575	4.25	7.32
12	6	0.44	7	9.24
24	6	0.385	16	18.48

Table 4.1: Energy in Region B with different SYSCLK's and the best SCLK.

Figure 4.2. This is consistent with the published characteristics of the microcontroller [34]. The relationship between microcontroller current and frequency is given by:

$$I = 3.132 \times 10^{-6} \times f^{0.88} + 0.2722mA \tag{4.1}$$

The maximum value allowed for $SCLK$ is $1/2 \times SYSCLK$ up to a maximum of 6 MHz [33][34]. From Table 4.1, we find that these are also the values for optimum energy usage. The energy consumed by region B at different $SYSCLK's$ is shown in Table 4.1. As seen from the table, the optimum configuration for the system is $SYSCLK = 6$ MHz and $SCLK = 3$ MHz.

With default settings for all regions, the energy used in region B was a small fraction of the total system energy usage. However, after optimizing the other regions from Figure 4.1, the initial region B was too large for the final system. In the following section, we will describe the operation of region B as an example of end-to-end energy optimization problem. We assume that the energy used by the bus, when data is being transferred, is negligible. As a result, the time taken in region B includes the time for data transfer to the radio, the time taken by the microcontroller for memory and control operations that are needed before data transfer can take place and the time to perform system initializations, i.e.

$$T = td_B + n_B/f + C \tag{4.2}$$

Here we use the following notation:

T = Length of region B.

td_B = Time taken to transfer data.

n_B = Number of microcontroller operations.

f = SYSCLK.

C = Constant time for initializations.

The constant C is introduced in the equation to get a better fit to the curve. Internal processes that lead to this are unknown. We have concluded that this may be the time taken for the system to initialize, i.e. for the microcontroller to go into the *on-state* from the *idle-state* and the radio to move into the *idle-state*. Also, from Table 4.1,

$$SCLK = f/2 \qquad \forall \qquad SCLK \leq 6MHz \tag{4.3}$$

By above definition,

$$\begin{aligned} T &= bits/SCLK + n_B/f + C \\ &= (2 \times bits + n_B)/f + C \end{aligned} \tag{4.4}$$

The current drawn by the microcontroller will be approximately proportional to its frequency of operation according to Figure 4.2 and Equation (4.1). We assume that the current drawn by the radio is negligible as it is in the idle state. Some $SYSCLK$ independent current is also observed to flow through the system. The reason for this current is not clear.

$$I_B = I_0 + (Q_{cycle} \times f^{0.88}) \tag{4.5}$$

where,

I_B = Current in region B.

I_0 = SYSCLK independent current.

Q_{cycle} = Charge/clock cycle.

f = SYSCLK.

Now substituting in the energy equation using a normal 3.0 V battery, we have

$$\begin{aligned} E &= Energy \\ &= Voltage \times Current \times Time \\ &= 3 \times I_B \times T \\ &= 3 \times (I_0 + (Q_{cycle} \times f^{0.88})) \times (bits/SCLK + n_B/f + C) \end{aligned} \quad (4.6)$$

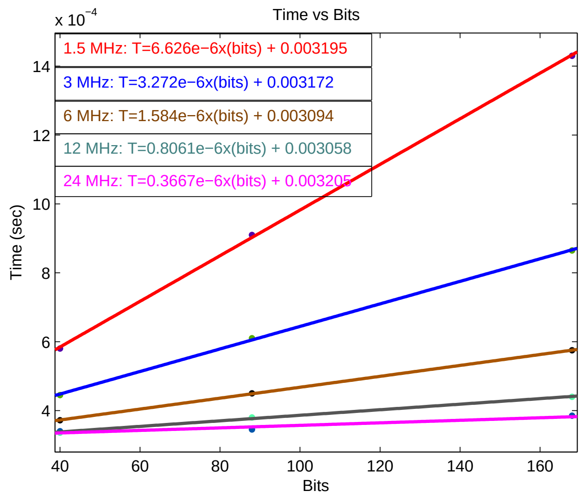


Figure 4.3: Plot of Time in Region B vs Bits of data transmitted at different SYSCLKs.

Figure 4.3 shows a plot of time in region B versus number of data bits transmitted. These readings are taken for different values of $SYSCLK$. For each set of readings the optimum value of $SCLK$ is chosen, which is half of the $SYSCLK$ provided $SCLK$ is no more than 6 MHz, as shown in Table 4.1. The relationship between the time taken and the number of bits transmitted is linear as expected. Also, as $SYSCLK$ increases, so does $SCLK$ and the time taken to transmit the same number of bits decreases. It is also evident from the graph that at 24 MHz the initialization constant $C = 300$ μsec from equation (4.4) dominates. Here the number of bits transmitted does not affect the time taken significantly.

Figure 4.4 is a plot of length of region B at different $SYSCLK$ values with 168 bits data. As described in Equation (4.4), the time taken decreases as frequency increases. The best fit to the graph is given by:

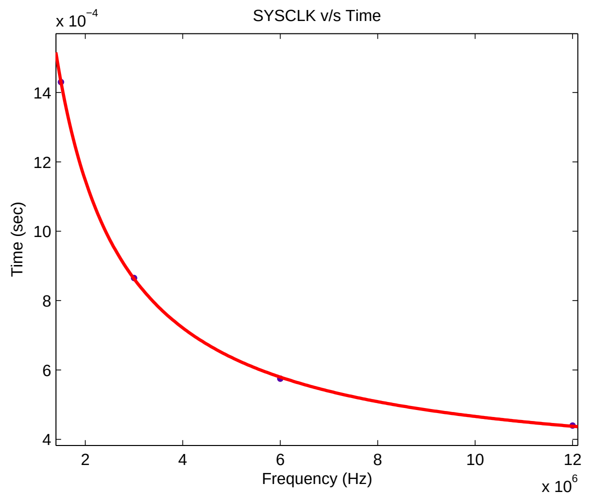


Figure 4.4: Plot of Time in Region B vs SYSCLK with 168 bits data.

$$T = 1701/f + 0.000296 \tag{4.7}$$

Comparing Equation (4.7) and Equation (4.4), $n_B = 1365$ and C=296 μsec.

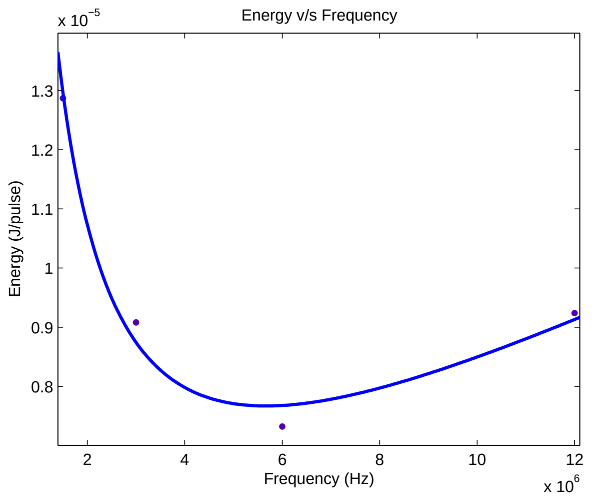


Figure 4.5: Plot of Energy dissipated in Region B vs SYSCLK with 168 bits of data.

Figure 4.5 is a plot of energy dissipated in region B at different $SYSCLKs$ with 168 bits data. The shape of the curve is as described in Equation (4.6). At lower $SYSCLKs$ the $1/f$ term is dominant and the energy used by the system decreases as

f increases. At higher $SYSCLKs$, the f term becomes more dominant and the energy dissipated increases with the increase in f. The best fit to this graph is:

$$E = 14.63 \times (1/f) + 3.643 \times 10^{-12} \times f^{0.88} + 1.224 \times 10^{-5} \times 1/f^{0.12} \tag{4.8}$$

The minimum energy point occurs at $SYSCLK = 6\ MHz$. Comparing Equation (4.8) and Equation (4.6), $I_0 = 2.867mA$ and $Q_{cycle} = 4.1 \times 10^{-9} coulomb/cycle$. Figure 4.4 and its fit suggests that by increasing the frequency of operation the time in Region B is reduced. However, we infer from Figure 4.5 and its fit that even though higher frequency will guarantee reduced time it does not imply reduced energy. From Equation (4.6) it is observed that after a certain point the energy spent will increase with the increase in frequency as $f^{0.88}$ becomes dominant.

4.2.3 Optimizing Region C

Region C is the time taken to initialize the radio. This is a constant 80 μs, irrespective of system clocks and the size of data. This region is not user controllable.

4.2.4 Optimizing Region D

The timing for region D depends on the amount of data to be transmitted, the data-rate and RF power of transmission. If the amount of data to be transmitted is reduced, the time in Region D will be reduced, implying lower energy consumption. If either data-rate or RF power or both are reduced the current drawn in this region will reduce, keeping the time constant, implying lower energy consumption. Experimentation is performed using larger than required RF transmission power of 10 dBm. Usually RF power of 0 dBm suffices. We are transmitting the maximum possible reliable modulation data-rate of 250 $kbps$. The time varies from 485 μs for 40 bits of data to 800 μs for 168 bits of data.

4.2.5 Optimizing Region E

The timing and energy in region E, i.e. time taken to calibrate the radio is constant irrespective of system clocks and size of data. This region occurs once every 4 cycles in our implementation. The radio has been found to be highly stable after estimating the radio drift and temperature dependence. Radio drift is a change in the carrier frequency from its nominal value. This is caused due to changes in temperature or by problems in the voltage regulator, which controls the bias voltage to the oscillator. Calibrating the radio once in 4 cycles is easy in firmware and hence is used in the prototype, but we could also calibrate it once in 10 cycles, once in 100 cycles or even less frequently. With such a configuration, the energy of region E can probably be ignored.

4.2.6 Summarizing

Table 4.2 gives a detailed description of the step-by-step improvement in the system. It indicates the hardware and software status of the PIP, the average total current drawn by the system and the percentage improvement in the system. Table 4.3 shows the detailed power consumption in each region (refer to Figure 4.1) with the default settings described in Table 4.2. Table 4.4 is the power consumption in each region with the unnecessary peripherals shut down. Table 4.5, Table 4.6, Table 4.7 and Table 4.8 give the detailed power consumption in each region with the final four configurations. A comparison of the energy consumption with different settings is shown in Figure 4.6. Figure 4.6 shows that energy consumption has reduced significantly compared to the original settings. Figure 4.7 shows the percentage of battery power consumed by each region in different configurations. It shows that with the improved system configuration, maximum power is consumed by the system during radio transmission in Region D. As a result, putting a major fraction of the energy into the radio and not the computation ensures better reliability.

It is evident that battery life can be increased further by reducing frequency of the slow clock, lowering the transmitted power and reducing the number of bits being transmitted over the air. Our measurements were performed with a transmitter power of $+10dBm$. This will probably be excessive for most applications and can be reduced as needed. The number of bits in the PIP ID depends on the number of unique $PIPs$ required to be deployed in the system. The number of unique $PIPs$ that can be deployed are $2^{Number-of-bits}$. All the bits will not necessarily represent the $PIP's$ ID. A few of them are put in for error correction. A 96 bit address for the $PIPs$ allows for 2^{96} distinct $PIPs$, which should be adequate as used in the current $RFID$ applications. It should also be noted that SPP adds a few bytes as the preamble (refer to Figure 3.4). The projected life of the battery with a few different settings is shown in Table 4.9.

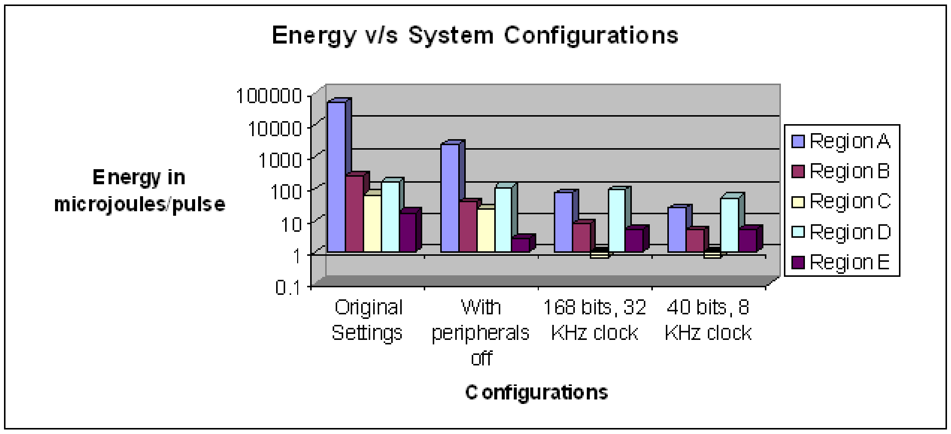


Figure 4.6: Bar graph of energy consumption with different system configurations.

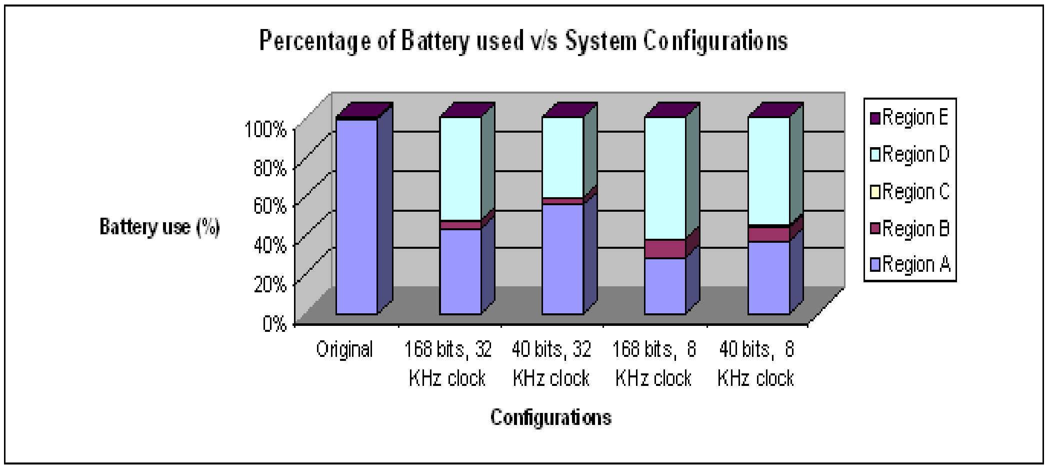


Figure 4.7: Bar graph showing the relative energy consumption of different regions with various system configurations. Note that final configuration is dominated by radio energy.

PIP STATUS	TOTAL CURRENT DRAW	IMPROVEMENT OVER DEFAULT
Radio-on always Microcontroller-on always Slow clock absent SYSCLK = 24 MHz SCLK = 6 MHz USB-on and ADC-on Comparator-on Voltage Regulator-on Voltage Reference-on VDD Monitor-on	16.6 mA	Default
Wake-on-Radio enabled Microcontroller on or idle Slow clock absent SYSCLK = 1.5 MHz SCLK = 41 KHz USB-on and ADC-on Comparator-on Voltage Regulator-on Voltage Reference-on VDD Monitor-on	1.5 mA	90.96%
Wake-on-Radio enabled Microcontroller on or idle Slow clock absent SYSCLK = 1.5 MHz SCLK = 41 KHz USB-off and ADC-off Comparator-off Voltage Regulator-off Voltage Reference-off VDD Monitor-on	800 μA	95.18%
Wake-on-Radio enabled Microcontroller on or idle Slow clock = 32 KHz SYSCLK = 6 MHz SCLK = 3 MHz USB-off and ADC-off Comparator-off Voltage Regulator-off Voltage Reference-off VDD Monitor-on	45 μA	99.73%
Wake-on-Radio enabled Microcontroller on or idle Slow clock = 32 KHz SYSCLK = 6 MHz SCLK = 3 MHz USB-off and ADC-off Comparator-off Voltage Regulator-off Voltage Reference-off VDD Monitor-off	24 μA	99.86%

REGION	TIME	CURRENT	ENERGY
A Microcontroller-on Radio-on	$1\ s$	$16.6\ mA$	49800 $\mu J/pulse$
B Microcontroller-on Radio-on	$4.21\ ms$	$19.6\ mA$	247.55 $\mu J/pulse$
C Microcontroller-on Radio-on	$800\ \mu s$	$26.1\ mA$	62.64 $\mu J/pulse$
D Microcontroller-on Radio-on	$1\ ms$	$52.6\ mA$	157.8 $\mu J/pulse$
E Microcontroller-on Radio-on	$300\ \mu s$	$19.6\ mA$	17.64 $\mu J/pulse$

Table 4.3: Default values with 168 bits data, slow clock not present, transmit power is 10 dBm, serial clock is 1.5 MHz and fast clock is 24 MHz.

REGION	TIME	CURRENT	ENERGY
A Microcontroller-on Radio-on	$1\ s$	$800\ \mu A$	2400 $\mu J/pulse$
B Microcontroller-on Radio-on	$4.21\ ms$	$3\ mA$	37.89 $\mu J/pulse$
C Microcontroller-on Radio-on	$800\ \mu s$	$9.5\ mA$	22.8 $\mu J/pulse$
D Microcontroller-on Radio-on	$1\ ms$	$36\ mA$	108 $\mu J/pulse$
E Microcontroller-on Radio-on	$300\ \mu s$	$3\ mA$	2.7 $\mu J/pulse$

Table 4.4: Values with peripherals shut down and with 168 bits data, slow clock not present, transmit power is 10 dBm, serial clock is 41 KHz and fast clock is 1.5 MHz.

REGION	TIME	CURRENT	ENERGY
A Microcontroller - idle (slow clock on) Radio-off	1 s	24 μA	72 $\mu J/pulse$
B Microcontroller - on (transferring data to radio) (fast clock on) Radio-idle	600 μs	4.25 mA	7.65 $\mu J/pulse$
C Microcontroller - on (fast clock on) Radio-on (transceiver initializing)	80 μs	10.8 mA	0.648 $\mu J/pulse$
D Microcontroller - on (fast clock on) Radio-on (transmitting)	800 μs	36.5 mA	87.6 $\mu J/pulse$
E (once in 4 cycles) Microcontroller - on (fast clock on) Radio-on (calibrating)	720 μs	9.6 mA	5.149 $\mu J/pulse$

Table 4.5: Optimized values with 168 bits data, 32 KHz slow clock, transmit power is 10 dBm, serial clock is 3 MHz and fast clock is 6 MHz.

REGION	TIME	CURRENT	ENERGY
A Microcontroller - idle (slow clock on) Radio-off	1 s	24 μA	72 $\mu J/pulse$
B Microcontroller - on (transferring data to radio) (fast clock on) Radio-idle	372 μs	4.25 mA	4.743 $\mu J/pulse$
C Microcontroller - on (fast clock on) Radio-on (transceiver initializing)	80 μs	10.8 mA	0.648 $\mu J/pulse$
D Microcontroller - on (fast clock on) Radio-on (transmitting)	485 μs	36.5 mA	53.1 $\mu J/pulse$
E (once in 4 cycles) Microcontroller - on (fast clock on) Radio-on (calibrating)	720 μs	9.6 mA	5.149 $\mu J/pulse$

Table 4.6: Optimized values with 40 bits data, 32 KHz slow clock, transmit power is 10 dBm, serial clock is 3 MHz and fast clock is 6 MHz.

REGION	TIME	CURRENT	ENERGY
A Microcontroller - idle (slow clock on) Radio-off	1 s	3 μA	24 $\mu J/pulse$
B Microcontroller - on (transferring data to radio) (fast clock on) Radio-idle	600 μs	4.25 mA	7.65 $\mu J/pulse$
C Microcontroller - on (fast clock on) Radio-on (transceiver initializing)	80 μs	10.8 mA	0.648 $\mu J/pulse$
D Microcontroller - on (fast clock on) Radio-on (transmitting)	800 μs	36.5 mA	87.6 $\mu J/pulse$
E (once in 4 cycles) Microcontroller - on (fast clock on) Radio-on (calibrating)	720 μs	9.6 mA	5.149 $\mu J/pulse$

Table 4.7: Optimized values with 168 bits data, 8 KHz slow clock, transmit power is 10 dBm, serial clock is 3 MHz and fast clock is 6 MHz.

REGION	TIME	CURRENT	ENERGY
A Microcontroller - idle (slow clock on) Radio-off	1 s	3 μA	24 $\mu J/pulse$
B Microcontroller - on (transferring data to radio) (fast clock on) Radio-idle	372 μs	4.25 mA	4.743 $\mu J/pulse$
C Microcontroller - on (fast clock on) Radio-on (transceiver initializing)	80 μs	10.8 mA	0.648 $\mu J/pulse$
D Microcontroller - on (fast clock on) Radio-on (transmitting)	485 μs	36.5 mA	53.1 $\mu J/pulse$
E (once in 4 cycles) Microcontroller - on (fast clock on) Radio-on (calibrating)	720 μs	9.6 mA	5.149 $\mu J/pulse$

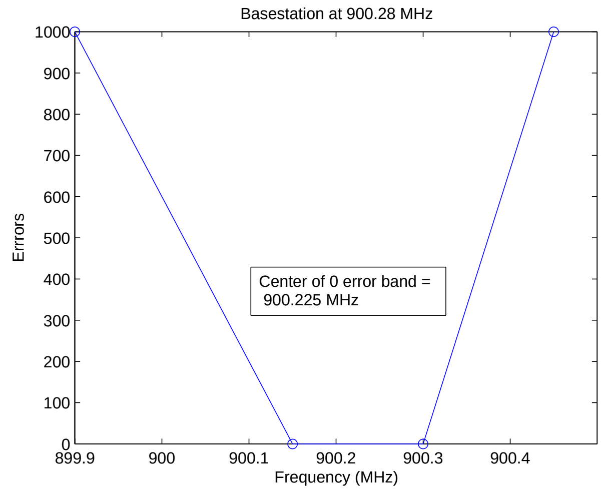
Table 4.8: Optimized values with 40 bits data, 8 KHz slow clock, transmit power is 10 dBm, serial clock is 3 MHz and fast clock is 6 MHz.

PARAMETERS	BATTERY LIFE
Data $\longrightarrow$ 168 bits Slow Clock $\longrightarrow$ Not Present	0.5 days
Data $\longrightarrow$ 168 bits Slow Clock $\longrightarrow$ 32 KHz	159 days
Data $\longrightarrow$ 40 bits Slow Clock $\longrightarrow$ 32 KHz	202 days
Data $\longrightarrow$ 168 bits Slow Clock $\longrightarrow$ 8 KHz	248 days
Data $\longrightarrow$ 40 bits Slow Clock $\longrightarrow$ 8 KHz	372 days

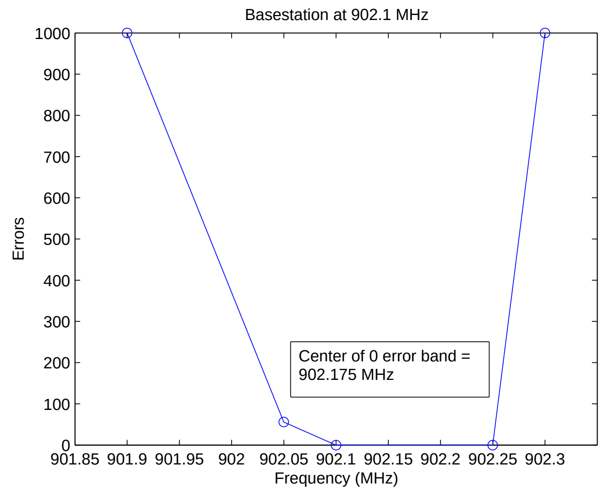
Table 4.9: Battery Life

4.3 Radio Link

In this section we look at several radio issues that affect the transmission and thus system energy usage. In Section 4.2 we achieved energy reduction in the system through optimizing firmware, clocking and packet protocol. The $PIPs$ and the base-station are transferring data at 250 *kbps* over a wireless medium using MSK modulation. MSK modulation was chosen for its high efficiency and low error in reception.



(a)



(b)

Figure 4.8: Plot of Errors (per 1000) at the base-station vs the carrier frequency at the transmitter. (a)With carrier frequency 900.28 MHz. (b)With carrier frequency 902.1 MHz.

From Figure 4.7, in the optimized system, Region D has the maximum energy use.

Region D is where the actual radio communication takes place. Accurate frequency tuning will give more effective use of the radio link and lower energy/bit transmission. This will further improve the system flexibility with regards to the intended application. It will provide more choice with regards to visibility range, interference, battery life trade-offs and also allow for multi-channel usage to further increase number of tags.

Because we are interested in end-to-end performance, we have used a tuning procedure based on reducing the overall error rate and not the received RF signal strength. For the radio section, the carrier frequency, channel number and channel width are programmable. The IF filter bandwidth is set to 500 KHz to maintain 250 *kbps* modulation rate. For error free reception at the base-station, it is necessary that the radio at the base-station is tuned with the radio on the $PIPs$. Packets are transmitted via the wireless medium using the Simple Packet Protocol described in Section 3.2. Packets are in error if they fail the CRC check or if certain sequence numbers are not received at the base-station. The carrier frequency is set[33] using:

$$\begin{aligned} f_{carrier} = & f_{XOSC}/2^{16} \times \\ & (FREQ + CHAN \times (256 + CHANSPC_M) \times \\ & 2^{CHANSPC_E - 2}) \end{aligned} \tag{4.9}$$

where,

f_{XOSC} = Frequency of the CC1100 oscillator.

$FREQ$ = Base frequency of the carrier.

$CHAN$ = Channel Number.

$CHANSPC_M$ = Channel spacing mantissa.

$CHANSPC_E$ = Channel spacing exponent.

Figure 4.8(a) and (b) are plots of errors at the base-station per 1000 packets transmitted by the PIP versus the carrier frequency at the transmitter. Both the graphs show a zero error band of 150 KHz width.

Observing Figures 4.8(a) and (b), it can be concluded that matching just the carrier frequency at the PIP and the base-station does not guarantee reliable radio operation, i.e. operation at the center of the minimum-error band. This difficulty is overcome by error rate as feedback. For tuning the system initially, the base-station is configured as a continuous transmitter. Its carrier frequency is set to the desired value using Equation (4.9). The actual frequency of transmission for the base-station is recorded using a spectrum analyzer. The $PIPs$ are set to continuous transmission mode and to the same carrier frequency using Equation (4.9). The actual frequency of transmission for the PIP was confirmed using the spectrum analyzer [39]. Let this frequency be f_{match}. The base-station is then configured to be a receiver with the frequency settings unchanged. The $PIPs$ are programmed to transmit beacons with a 1 sec interval between them. Errors per 1000 received packets at the receiver are noted down. With the f_{match} at the PIP there will be 0 errors at the base-station. Now the carrier frequency for the PIP is changed above and below f_{match} in steps of 50 KHz till there is no reception at the base-station. From the available readings the zero-error band was obtained. The operating frequency of the PIP is then set to the center of the minimum-error band.

Chapter 5

Collision Dynamics

The essence of our power optimization strategy is based on being able to manage packet collisions without having a receiver on the tag. In this section we look at packet collision dynamics to get a better understanding of their effect in this system. Theoretical justification for this approach has been published in [38] and we will report error rates for a multi-tag system in another paper. As indicated earlier, we propose an asset tracking system using transmit only active tags which has long range continual tracking with low power consumption. In our system, the tags transmit a beacon signal continuously with a predefined interval between two beacons. There is no feedback from the tags. There will be collisions between packets over the air in a multi-tag system even with the shortest possible packet. We have carefully studied a two tag system to understand the details of collision dynamics and develop strategies for reducing collisions and to recover from them.

5.1 A Two Tag System

In [40][38] we have simulated the behavior of a multi-tag system. In this section we present the results of an experiment to observe the working of a two-tag system.

Consider a system with a single base-station and two $PIPs$ (called $PIP2$ and $PIP5$). The base-station was approximately 1 m from the $PIPs$ and placed such that the received signal strength from both of them at the base-station is almost the same.

The PIP's transmit data at 250 $kbps$. Data includes the $PIP\ ID$ and the sequence number of the packet being transmitted and is 168 $bits$ long. After adding the SPP overhead the total packet length is approximately 205 $bits$. The transmission power

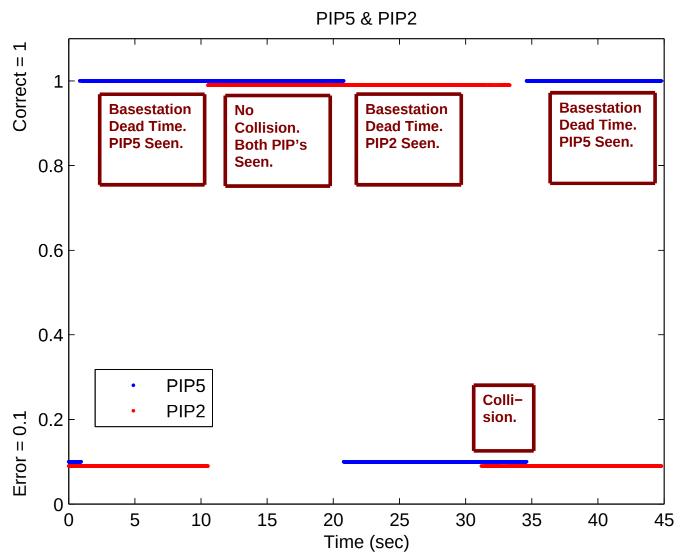
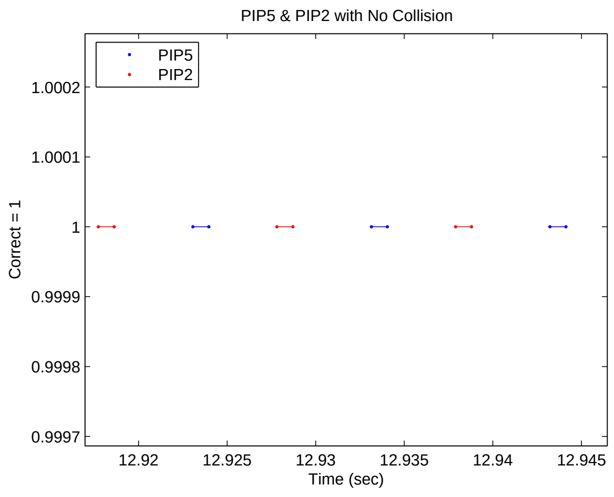


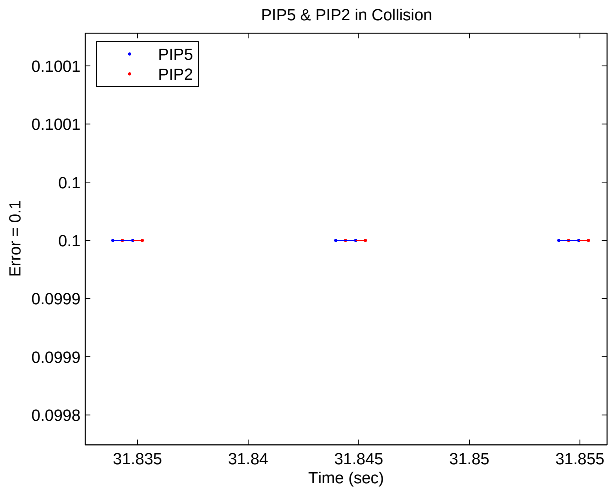
Figure 5.1: Plot of Correct Detection at the Receiver vs Time for PIP5 and PIP2.

was set to 0 dBm. The modulation was set to MSK with a carrier frequency of 902.1 MHz. The base-station was tuned to 902.1 MHz as described in Section 4.3. $PIP5$ continuously transmits a beacon with an interval of 10.080393 ms. $PIP2$ continuously transmits a beacon with an interval of 10.083471 ms. It should be noted that this is a special experiment with ultra-fast beacons. The $PIPs$ are transmitting approximately every 10 ms which is 100 times faster than the intended minimum beacon interval which is 1 sec. Ultra-fast transmission was done in order to collect the statistics in reasonable time. When only $PIP5$ (or $PIP2$) was transmitting there were no errors i.e. all packets are correctly received and processed at the base-station.

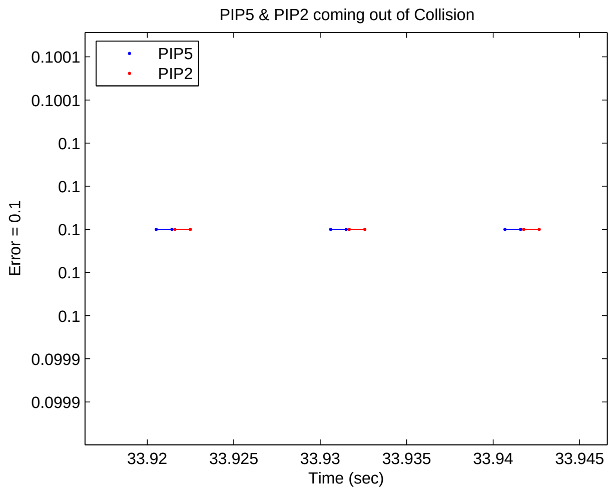
As said earlier, data bits were being transmitted at 250 $kbps$. Hence, each bit has a duration of 4 μs. Consider a situation in which both $PIP5$ and $PIP2$ are up and running. $PIP5$ transmits every 10.080393 ms and $PIP2$ every 10.083471 ms. Hence, there is an offset of approximately 3 μs between the 2 $PIPs$ and every 3000 periods the two tags should go in and out of collision. This experiment will demonstrate the behavior of the system during all phases of a collision run.



(a)



(b)



(c)

Figure 5.2: Collision dynamics for a 2 PIP system. (a)Plot of Correct detection at the receiver vs Time for PIP5 and PIP2 when there is no Collision. (b)Plot of Correct detection at the receiver vs Time for PIP5 and PIP2 in Collision. (c)Plot of Correct detection at the receiver vs Time for PIP5 and PIP2 moving out of Collision.

5.2 Experimental Results for A Two Tag System

For the experiment, $PIP5$ and $PIP2$ run simultaneously for 45 *seconds*. $PIP5$ and $PIP2$ were started at random time instances. Figure 5.1 shows correct detection at the receiver as a function of time at the base-station. On the y-axis, 0.1 indicates an error in the received packet. Error may be due to incorrect reception i.e. garbled data or loss packets. A 1 on the y-axis indicates correct reception at the base-station.

As seen in Figure 5.1 from 0 *sec* to about 33 *sec* $PIP2$ and $PIP5$ go into and come out of collision i.e. at around 33 *sec* time instance they are at the same position relative to each other as they were at the 0 *sec* time instance. The period of collision and no collision is shown in the figure. There is also a time period, which we call the 'Base-station Dead Time', during which only one of the tags is seen. This is due to the limitation of the back-end interface of the base-station which is the RS-232 link.

Figure 5.2 is the magnified view of Figure 5.1, showing the details of packets going into and coming out of collision. The time duration of each packet was 900 μsec with a transmission data-rate of 250 *kbps*. Figure 5.2(a) shows the $PIPs$ with their beacons separated from each other and for the case where there is perfect reception at the base-station with zero errors. This is the region of no collisions. Figure 5.2(b) shows clearly the $PIPs$ in collision with their beacons overlapping each other. All the packets received in this interval are in error. Figure 5.2(c) shows the $PIPs$ coming out of collision. About 400 packets received in this interval are in error.

Figure 5.3 shows a plot of the time difference between adjacent $PIPs$ transmission time. As mentioned earlier the 'Base-station Dead-time' is when packets of only one PIP are seen due to the networked hardware limitations. This occurs due to the buffer at the base-station being overwritten, before it can send data out through the serial port. Hence, when adjacent packets of $PIP2$ and $PIP5$ are less than 3 ms apart, data of only a single PIP is sent to the back-end, while data from the other PIP gets overwritten before it can be sent over the RS-232 link. This makes the effective packet length 3 ms. Note that the $PIPs$ have been transmitting approximately every 10 ms, which is 100 times *faster* than the intended minimum beacon interval which is 1

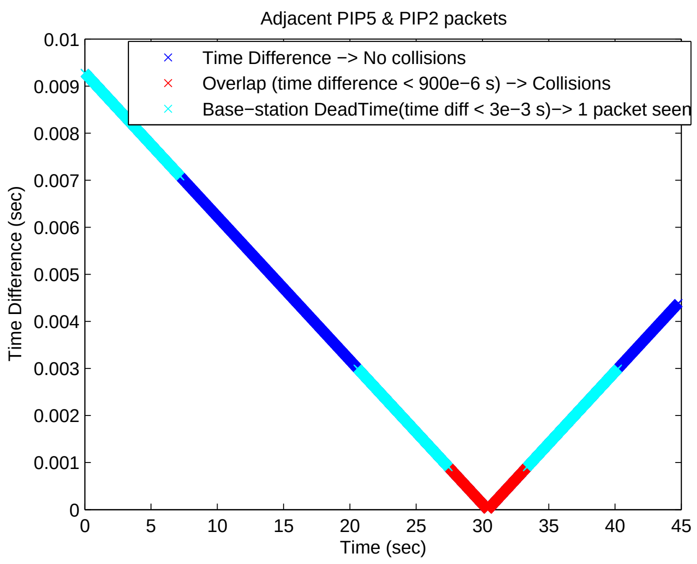


Figure 5.3: Plot of the Theoretical transmission time difference between PIP5 and PIP2 vs Time.

sec. This demonstrates the problem in pervasive systems that makes data aggregation important, implying that even though the radio is working perfectly fine in receiving adjacent packets of $PIP2$ and $PIP5$ less than 3 ms apart but out of overlap, due to buffer overflow at the base-station packets are being lost. In figure 5.3, the time interval 0 to 5 ms shows 'Base-station Dead-time' even when the interval between the intended packets of adjacent $PIPs$ is large. This is because there is a contention of the intended $PIP5$ packet with the previous $PIP2$ packet.

Figure 5.4 is a plot of the $RSSI$ for $PIP2$ and $PIP5$ at the base-station as a function of time. It is assumed that the received signal strength is minimum when packets are missing or the received data is garbled. Figure 5.4 shows both $PIP5$ and $PIP2$ being received at almost the same signal strength. It was observed that before the region of overlap, from 20 sec to 30 sec, $PIP2$ dominated and was being received at the base-station. However, most of the packets of $PIP2$ received in this interval fail the CRC checksum. Also, just after collision, its $PIP5$ that dominated as observed from 0 sec to 10 sec. 0 sec implies the $PIPs$ were just out of collision as, from 0 sec to about 33 sec $PIP2$ and $PIP5$ go into and come out of collision i.e. at around 33

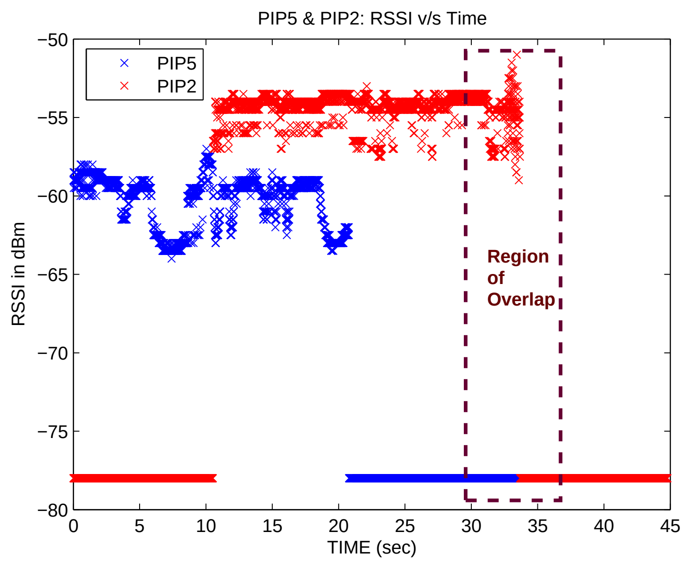


Figure 5.4: Plot of the received RSSI at the base-station for PIP2 and PIP5 vs Time.

sec time instance they are at the same position relative to each other as they were at the 0 sec time instance. During this interval too most of the packets of $PIP5$ failed the CRC checksum.

Figure 5.5 shows $PIP5$ coming out of collision. Y-axis indicates the number of bytes that can be deciphered by the base-station. It was mentioned earlier that $PIP5$ is faster than $PIP2$ by approximately 3 μs, as the $PIPs$ come out of collision, with each successive epoch an increasing number of bytes of $PIP5$ are seen at the base-station. Approximately 10 epochs are needed for 1 byte of $PIP5$ to come out of overlap. During this process and during the duration of complete overlap the received ID's of both $PIPs$ at the base-station are garbled. After all 10 bytes of PIP5 come out of overlap, both $PIPs$ can be recognized distinctly by the base-station.

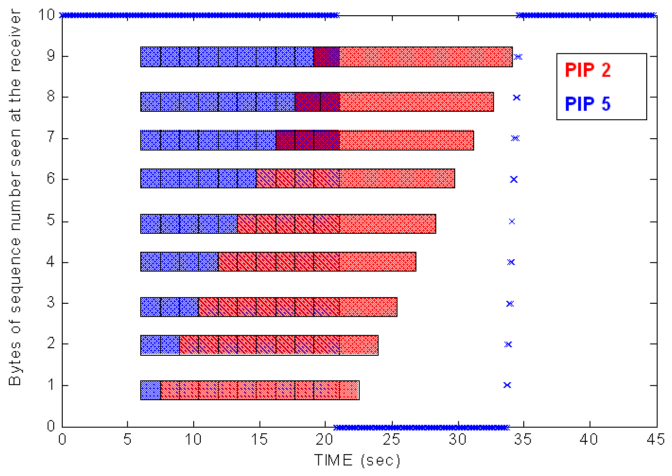


Figure 5.5: Plot of the sequence number bytes of PIP5 and PIP2 as PIP5 comes out of collision.

Chapter 6

Concluding Remarks

6.1 Conclusion

In order to provide perpetual monitoring of assets, it is necessary to have overall visibility of the environment at any single instant. In a system monitoring tagged objects, ideally each object should say ‘I am here’ at all times for overall visibility. Thus our objective for this thesis was full inventory tracking in real time.

In this thesis, we have presented a working energy efficient $RFID$ system monitoring entities tagged with our custom made $PIPs$, which combine the best features of both active and passive tags, i.e. small size, many reads, long range, long life.The major challenge in Roll-Call was to supply power needed for long range continuous tracking for a year or more while keeping the $PIPs$ small and inexpensive. We have used this as a model problem for optimizing cost, size and lifetime across the entire system, from firmware to protocol. In Section 4.2 we achieved energy optimization for the system, by careful analysis of each region in the $PIP's$ time-dependent current drain, by identifying causes for needless energy expenditure, and carefully optimizing the hardware and code running on the transmitter, which increased the battery life from half a day to 372 days making the system feasible for deployment. We also put a major fraction of the energy into the radio and not the computation to ensure optimum use of the system power. Design decisions regarding the system clock frequency and the amount of energy spent in computation and communication were made maintaining raw error rates of approximately 0.01% or less at the base-station.

Accurate frequency tuning was performed in Section 4.3 to facilitate effective use of the radio link keeping energy per bit low during data transmission. In Section 5 we performed an experiment to observe packet collision dynamics to get a better understanding and develop strategies to recover from them.

6.2 Future Work

There are many interesting directions for further improving the 'Roll-Call' system that we have built.

Base-station Dead Time was mentioned in Section 5.2 during which only one of the tags was seen. This was due to the limitation of the back-end RS-232 interface. This will be corrected in the next version. Once this limitation is removed it will be possible to run detailed experiments with hundreds of deployed $PIPs$. This will provide an opportunity to study collision dynamics for the system and to try out different ways to reduce collisions. Data-analysis software can also be incorporated in the back-end to further reduce the probability of false alarms. One way would be, for the back-end to predict the time instances of collision and to try and recover from them.

Another change in the near future would be to allow direct access to RF signals at the base-station for data processing. Theoretical methods to better demodulate the incoming RF for low error rates has been published in [38].

We are currently using Simple Packet Protocol for transmitting data over the wireless medium as it is easy to implement for our $CC$1100 radio. However, it is necessary change to some other protocol in future versions as the overhead for SPP is quite large. Also some form of error correction, maybe in the form of redundant bits, has to be incorporated.

The base-station for this version of PIP is a Texas Instruments $CC$1100/$CC$1150-868/915 MHz Development Kit. This will be replaced by $PIPs$ in the next version for both the base-station and the PIP have the same basic components. The radio currently transmits reliably at 0.25 $Mbps$. The hardware will be modified to have reliable transmission up to 1 $Mbps$. This will make it possible for the system to monitor approximately 10^3 - 10^4 tagged assets with tolerable error rates [38].

References

[1] M. Sarrafzadeh, F. Dabiri, R. Jafari, T. Massey, and A. Nahapetian. Low power lightweight embedded systems. In *International Symposium on Low Power Electronics and Design (ISLPED)*, October 2006.

[2] M. Pedram. Power optimization and management in embedded systems. In *Conference on Asia South Pacific design automation*, pages 239 – 244, 2001.

[3] C. Schurgers, V. Tsiatsis, and M.B. Srivastava. Stem: Topology management for energy efficient sensor networks. In *Aerospace Conference Proceedings, IEEE*, volume 3, pages 3–1099– 3–1108, 2002.

[4] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M.B. Srivastava. Optimizing sensor networks in the energy-latency-density design space. In *IEEE Transactions on Mobile Computing archive*, volume 1, pages 70 – 80, January 2002.

[5] K. Arisha, M. Youssef, and M. Younis. Energy-aware tdma-based mac for sensor networks. In *Proceedings of the IEEE Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT)*, 2002.

[6] W. Ye and J. Heidemann. Medium access control in wireless sensor networks. In *USC/ISI Technical Report ISI-TR-580*, October 2003.

[7] K. Jamieson, H. Balakrishnan, and Y.C. Tay. Sift: a mac protocol for event-driven wireless sensor networks. In *Third European Workshop on Wireless Sensor Networks (EWSN)*, February 2006.

[8] K. Oikonomou and I. Stavrakakis. An adaptive time-spread multiple-access policy for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking.*

[9] L. Bolotnyy and G. Robins. Multi-tag radio frequency identification systems. In *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AUTOID)*, pages 83 – 88, 2005.

[10] L. Bolotnyy and G. Robins. Randomized pseudo-random function tree walking algorithm for secure radio-frequency identification. In *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies (AUTOID)*, pages 43 – 48, 2005.

[11] A. Juels, R. L. Rivest, and M. Szydlo. The blocker tag: Selective blocking of rfid tags for consumer privacy. In *8th ACM Conference on Computer and Communications Security*, pages 103–111, 2003.

[12] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, July 2001.

[13] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom, Rome, Italy*, pages 70 – 84, July 2001.

[14] X. Wang, J. Ma, and S. Wang. Collaborative deployment optimization and dynamic power management in wireless sensor networks. In *Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC'06)*, 2006.

[15] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao. The cougar project: a work-in-progress report. In *ACM SIGMOD SPECIAL ISSUE: Special section on sensor network technology and sensor data managment*, pages 53 – 59, 2003.

[16] R. Passos, C. Coelho Jr, A. Loureiro, and R. Mini. Dynamic power management in wireless sensor networks: An application-driven approach. In *Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS05)*, 2005.

[17] O. Hernandez, G. Dande, and J. Ofri. C++ encapsulated dynamic runtime power control for embedded systems. In *SoutheastCon, 2005. Proceedings. IEEE*, pages 126 – 130, April 2005.

[18] S. Hua and G. Qu. Approaching the maximum energy saving on embedded systems with multiple voltages. In *Proceedings of the International Conference on Computer Aided Design (ICCAD03)*, 2003.

[19] Lattice Semiconductor Corporation. Dynamic power management in an embedded system. A Lattice Semiconductor White Paper.

[20] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization. In *Technical report, U.C. Berkeley*, 2001.

[21] S. Kulkarni and M. Arumugam. Tdma service for sensor networks. In *InternationalWorkshop on Assurance in Distributed Systems and Networks (ADSN), ICDCS04 Workshop*, 2004.

[22] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on Systems Sciences*, 2000.

[23] K. Sohrabi and G. Pottie. Performance of a novel self-organization protocol for wireless ad hoc sensor networks. In *Proceedings of the IEEE 50th Vehicular Technology Conference*, pages 1222 – 1226, 1999.

[24] Bluetooth SIG Inc. Specification of the bluetooth system: Core. http://www.bluetooth.org/, 2001.

[25] J. Haartsen. The bluetooth radio system. In *IEEE Personal Communications Magazine*, pages 28 – 36, Feb. 2000.

[26] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM*, 2002.

[27] A. Woo and D. Culler. A transmission control scheme for media access in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, Rome, Italy*, pages 221 – 235, July 2001.

[28] F. Bennett, D. Clarke, J. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded mobile networking. In *IEEE Personal Communications Magazine*, volume 4, pages 8 – 15, Oct. 1997.

[29] S. Singh and C.S. Raghavendra. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. In *ACM Computer Communication Review*, volume 28, page 5 26, July 1998.

[30] G. Roussos. Enabling rfid in retail. In *IEEE Computer Technology*, pages 27–33, March 2006.

[31] R. Weinstein. Rfid: A technical overview and its application to the enterprise. In *IEEE Computer Technology*, pages 27–33, May 2005.

[32] P. Sorrells. Passive rfid basics. In *MT Inc - Microchip Technology Inc*, 1998.

[33] Chipcon Products from Texas Instruments. Chipcon cc1100 radio's datasheet. http://focus.ti.com/docs/prod/folders/print/cc1100.html.

[34] Silicon Laboratories. Silicon laboratories c8051f32x microcontroller's datasheet. www.silabs.com/public/documents/tpub_doc/dsheet/ Microcontrollers/USB/en/C8051F32x.pdf.

[35] Panasonic. Panasonic's cr2032 coin battery's datasheet. www.panasonic.com/industrial/battery/oem/images/pdf/Panasonic_Lithium _CR2032_CR2320.pdf.

[36] Texas Instruments. Texas instrument's cc1100 development kit. http://focus.ti.com/docs/toolsw/folders/print/cc1100-1150dk-868.html.

[37] N. Mandayam and D. Wu. Wireless communication technologies: Minimum shift keying. www.winlab.rutgers.edu/ narayan/Course/WSID/Lectures02/lect11.pdf.

[38] Y. Zhang, G. Bhanage, W. Trappe, Y. Zhang, and R. Howard. Facilitating an active transmit-only rfid system through receiver-based processing. To appear in the Proceedings of the 4th Annual IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2007).

[39] Agilent Technologies. Agilent esa series spectrum analyzers. ftp://ftp.testequity.com/pdf/agt_esa.pdf.

[40] G. Bhanage, Y. Zhang, Y.Y. Zhang, W. Trappe, and R. Howard. Rollcall: The design for a low-power and highly-robust asset tracking system. In *Eurocon: International Conference on Computer as a tool.*, September 2007.