University of Massachusetts Amherst
ScholarWorks@UMass Amherst

Masters Theses 1911 - February 2014

2011

Leveraging Multi-radio Communication for Mobile Wireless Sensor Networks

Jeremy J. Gummeson
University of Massachusetts Amherst

LEVERAGING MULTI-RADIO COMMUNICATION FOR MOBILE WIRELESS SENSOR NETWORKS

A Thesis Presented

by

JEREMY J. GUMMESON

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

February 2010

Electrical and Computer Engineering

LEVERAGING MULTI-RADIO COMMUNICATION FOR MOBILE WIRELESS SENSOR NETWORKS

A Thesis Presented

by

JEREMY J. GUMMESON

Approved as to style and content by:

Tilman Wolf, Chair

Aura Ganz, Member

Prashant Shenoy, Member

C. V. Hollot, Department Chair
Electrical and Computer Engineering

To Mom, Dad, and all the other J's.

ACKNOWLEDGMENTS

Many people have played an important role in helping me through my career as a graduate student. I would like to thank those responsible for guiding me through the work presented in this thesis.

First to my advisor in the Computer Science Department, Deepak Ganesan, whose research experience, advice, and collaborative efforts helped mold my research ideas into a proper thesis. To Prashant Shenoy for being an invaluable mentor and advisor, whose research vision is always an inspiration. To Mark Corner, whose keen insight in embedded hardware systems helped shape this work. Also, to Tilman Wolf and Aura Ganz for serving on my thesis commitee and giving valuable feedback at my thesis proposal.

ABSTRACT

LEVERAGING MULTI-RADIO COMMUNICATION FOR MOBILE WIRELESS SENSOR NETWORKS

FEBRUARY 2010

JEREMY J. GUMMESON

B.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf

An important challenge in mobile sensor networks is to enable energy-efficient communication over a diversity of distances while being robust to wireless effects caused by node mobility. In this thesis, we argue that the pairing of two complementary radios with heterogeneous range characteristics enables greater range and interference diversity at lower energy cost than a single radio. We make three contributions towards the design of such multi-radio mobile sensor systems. First, we present the design of a novel reinforcement learning-based link layer algorithm that continually learns channel characteristics and dynamically decides when to switch between radios. Second, we describe a simple protocol that translates the benefits of the adaptive link layer into practice in an energy-efficient manner. Third, we present the design of *Arthropod*, a mote-class sensor platform that combines two such heterogneous radios (XE1205 and CC2420) and our implementation of the Q-learning based switching protocol in TinyOS 2.0. Using experiments conducted in a variety of urban and forested environments, we show that our system achieves up to

52% energy gains over a single radio system while handling node mobility. Our results also show that our system can handle short, medium and long-term wireless interference in such environments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Mobile sensor networks have received increased research attention recently with applications ranging from vehicular networks (e.g. DieselNet [3]) to animal tracking (e.g. ZebraNet [9]). The choice of the wireless radio is perhaps the single most crucial design parameter for designing a mobile sensor network. The wireless radio must enable node-to-node and node-to-basestation communication over distances dictated by application needs, while being energy-efficient and robust to wireless effects introduced by mobility patterns. With advances in communication technologies, a spectrum of wireless radios are available to meet the needs of a sensor network. Table 1.1 depicts four common wireless radios used by today's sensor network platforms. As shown in the table, wireless radios are generally designed with a communication range in mind. For example, the Xtend and the XE1205 radios are designed for low-bitrate long-range communication over distances of a mile or more. In contrast, 802.11 and CC2420 radios enable high and low bandwidth communication, respectively, over short ranges of hundreds of feet or less. Thus, the sensor network designer must make a critical design choice. She can either choose a long-range radio enabling nodes to communicate over long distances but at the expense of expending more power. Or she can choose a shorter range radio that is more power-efficient but forego communication over longer distances.

Note that traditional techniques for range adaptation via power control or range elongation via the use of directional antennas do not address this tradeoff for mobile sensor networks. As shown in Table 1.1, modern radios support range adaptation using power control — a higher power setting can be used to increase the communication range of the

Radio	Bandwidth	transmit power levels (min, max), steps	transmit energy/bit (min,max)	receive power	max outdoor range
CC2420	250 Kbps	(-25,0dBm),31	102,208nJ/bit	56.7mW	80m
XE1205	38.1 Kbps	(0,15dBm),4	1803,5276nJ/bit	42.0mW	80m - 800m
XE1205	76.8 kbps	(0,15dBm),4	894,2617nJ/bit	42.0mW	80m - 800m
802.11b	11 Mbps	(0,15dBm),4	-,120nJ/bit	900mW	100m
XTend	9.6kbps	(0,30dBm),4	57.3,380.2uJ/bit	240mW	2-3km

Table 1.1: A spectrum of radio hardware

radio. While it is possible to choose a long range radio and use lower power settings for short range communication, doing so is far less efficient than using a short range radio for communicating over shorter distances. As shown in Table 1.1, the lowest power setting on the XTend radio is still 561x more expensive than using the CC2420 radio. Using a radio at its maximum range is never desirable, as packet loss rates increase with distance; the radios mentioned in Table 1.1 typically have a packet loss rate of $\sim 30\%$ at the reported distances, further emphasizing the need for appropriate radio hardware. Similarly, it is not feasible to use a radio designed for short range communication and to "increase" its range by using directional antennas. Directional antennas have been used successfully to increase the communication range of such radios – for example, the Mobisteer project [13]. However, since directional antennas are bulky, it is not feasible to deploy them in many mobile sensor network settings; for instance, animal tracking deployments require compact packaging of the mobile sensors.

These observations about the hardware characteristics of low-power radios lead us to the following thesis statement: *A wireless sensor system that employs both short and long-range radios is capable of achieving the energy efficiency benefits of the short range radio as well as the range benefits of the long-range radio.*

We validate this statement by pairing two complementary radios with heterogeneous range characteristics. This hardware enables mobile sensor nodes with the ability to achieve a significantly greater range diversity at a lower total energy cost when compared to a single radio. The use of multiple radios has been extensively investigated in the cellular community [28], but the radios employed in cellular devices are used to either maximize bandwidth or achieve interoperability. The key idea of our work is to operate each radio over a range where it is more energy efficient and to switch to the other radio whenever a mobile node moves from one radio's effective range to another. Specifically, we choose a high bandwidth spread spectrum radio with poor range characteristics and a variable bitrate radio tuned to have low bandwidth but better range. The two radios also operate in two isolated frequency bands – an impossibility with a single radio. In this manner, we achieve the best-of-both-worlds and eliminate the drawbacks of a single radio platform. An additional benefit of pairing complementary radios is that it enables adaptation to channel interference—by dynamically choosing the radio with the least interferences from other wireless devices. When using the two radios to adapt to interference, the variable bitrate radio is tuned for higher bandwidth but reduced range to better complement the alternate radio interface. The isolated frequency bands used by each radio allows robust adaptation.

We present the design of a heterogeneous multi-radio platform and system for handling range dynamics, where the choice of which radio to use for communication is made dynamically based on current channel characteristics, specifically wireless channel variations caused by device mobility and range effects. To shield applications from the increased complexity of choosing between radios, we present the design of a unified link layer that *transparently* chooses which radio to employ for communication between a pair of nodes. At the core of such a link layer is an adaptive algorithm that can dynamically decide *when* to use each radio for a wide range of mobility patterns. Such an algorithm is non-trivial since it needs to continually monitor and "learn" channel characteristics for the two radios and determine which one provides the lowest energy communication channel. Additionally, the practical implementation of such an adaptive link layer on sensor platforms presents a significant challenge since the energy and resource overhead for monitoring, learning, and switching between radios needs to be kept as low as possible.

1.1 Contributions

In this thesis, we propose a multi-radio hardware and link layer solution for range-adaptive mobile wireless sensor networks. Our work has three major contributions:

Q-Learning based Unified Link Layer: Our first contribution is a reinforcement-learning based algorithm that enables adaptatation across radios with different power/range tradeoffs. This algorithm learns the characteristics of radio channels through exploration and continually adapts to use the more efficient one.

Multi-radio Switching Protocol: Our second contribution is a energy-efficient switching protocol that translates the benefits of the Q-learning based adaptation algorithm into practice. The protocol transparently switches between radios, thereby providing the abstraction of a unified link layer to applications executing on multi-radio platforms.

Heterogeneous Multi-Radio Sensor Platform: Our third contribution is the design of a new mote-class sensor platform, the *Arthropod*, that pairs two radios with complementary characteristics: the CC2420 and XE1205. These radios have very different maximum ranges (80 meters vs 800 meters), and also significantly differ in their maximum power output (0 dBm vs 15 dBm) Thus, the *Arthropod* offers good potential for range adaptation to handle mobility effects.

We conduct mobility experiments using our hardware and software prototype in a variety of settings—urban/indoor, urban/outdoor, foliage— and for a range of mobility patterns —continuous and nomadic— that are typical in mobile sensor network deployments. Our experiments show that we obtain up to 52% improvements in energy efficiency over using only one of the two radios on the platform, while achieving a loss rate only marginally higher than using just the high-power radio. Our experiments on interference dynamics show that our link layer can adapt to short, medium and long-term wireless interference, while yielding a significant reduction in energy usage over a single radio system.

CHAPTER 2

RELATED WORK

ince radio diversity presents clear benefits along a number of dimensions: energy, robustness to interference, increased bandwidth and ease of deployment, a number of multi-radio systems have been designed in recent years. This has primarily involved a separation of control tasks such as neighbor discovery or neighbor wakeup from data transmission. Such a separation has been achieved by pairing 802.11 with the CC2420 [12] or the CC1000 [10, 15, 23], 802.11 with a custom radio for Wake-On-Wireless [22], [18], and 802.11 with an XTend [26] radio [3] for the UMassDieselNet DTN [4]. While such static allocation of roles to radios offers useful benefits, it does not fully utilize the potential of multi-radio systems. In our system, either radio can be used for control or data communication and the choice of which radio to use for communication is made dynamically based on current channel characteristics.

Multiple radio interfaces have also been exploited for increasing bandwidth and tolerating disconnection on mobile wireless devices. The Mobile Access Router [20] exploits multiple types of radio interfaces (eg. 802.11, GPRS, etc), or interfaces tied to different service providers to aggregate bandwidth and avoid stalled transfers. A related technique is PTCP that uses link-layer striping [8] to achieve a similar goal. All these mechanisms are aggressive in using multiple interfaces and do not take energy into account when choosing an interface. An updated Wake-On-Wireless system [1] and Context-for-Wireless [17] use 802.11 with cellular radios for data transmissions, with a static preference given to 802.11 when available.

One dynamic, energy-aware system is Coolspots, which combines 802.11 with Bluetooth [14]. Coolspots chooses Bluetooth transmission when available, and 802.11 when Bluetooth is insufficient to meet the bandwidth requirements. However, the choice of when to use a radio is made using coarse-grained feedback from the network layer, and neglects the benefits of a fine-grained, link-layer approach; this type of approach is useful because it allows a system to react quickly to short term dynamics. Another approach to dynamically utilizing a multi-radio system to achieve is found in [21] and [11]. These systems pair 802.11 with 802.15.4 and chooses the appropriate interface based on data size; energy efficiency is achieved by batching packet transmission. Achieving efficiency by increasing latency is beneficial, but instead our work focuses on reducing energy consumption by reacting to variations in mobility and channel conditions. Other systems, such as Triage extend this paradigm from multiple radios to multiple platforms [2]; however in this work a single platform is sufficient to process data transmission from both radios.

Recent work on wireless mesh networks has explored designs with multiple radios per node. For instance, carefully planned mesh networks can exploit multiple radios to make channel assignment more effective [5]. However, these approaches have not addressed the problem of algorithms to dynamically react to changing channel characteristics, and do not consider energy efficiency.

CHAPTER 3

A Q-LEARNING BASED MULTI-RADIO LINK LAYER

Our work assumes the availability of a heterogeneous dual-radio hardware platform. As previously mentioned, a multiple radio system can achieve greater range diversity and better interference adaptation than a single radio alternative. A potential drawback of additional hardware is an increase in the complexity of software required to manage radio parameters. To overcome these challenges, we present a unified link layer that is driven by a learning algorithm that adapts radio state parameters based on packet statistics. To realize the decisions made by this algorithm, we also employ a radio-switching protocol that handles transitions between different hardware configurations. The result is a single-radio abstraction that is robust with respect to interference and mobility dynamics.

Mobile multi-radio systems regularly incur unpredictable and widely varying conditions due to channel effects such as shadowing, fading, and multi-path effects, as well as varying interference. While adapting to these dynamics, the channel has hidden state: conditions on the radio not being used. In order to avoid local minimum, the system must periodically attempt to explore other operating states.

In particular, we have chosen to use a reinforcement technique called Q-Learning that provides exactly the properties required: a simple reward for making correct decisions and an ability to explore other operating points periodically [24]. In this chapter, we introduce some concepts from reinforcement learning and outline the design of the adaptation algorithm that is at the core of our unified multi-radio link layer.

Algorithm 1 Q-Learning

```
Initialize $Q(s, a)$ aribitrarily
Repeat(for each step of episode):
Choose $a$ from $s$ using policy derived from $Q$ ($\epsilon$-greedy)
Take action $a$, observe $r, s'$
$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s', a') - Q(s, a)]$
$s \leftarrow s'$;
until $s$ is terminal
```

3.1 Introduction to Q-Learning

Q-Learning is a reinforcement-learning technique to enable decision-making for agents in an unknown environment [24]. An agent continually takes an action from a set of possible actions and observes some reward associated with the outcome of their decision. In Q-Learning, there is a ”Q-Matrix” that updates according to the reward received, and the state transitions. This Q-Matrix is used to determine which action is optimum while an agent is in a given state; A Q-Matrix has a corresponding Reward-Matrix that contains the reward to be received by the agent for arriving at a particular state. In Q-Learning, the agent will occasionally take a random action to explore states that have not been visited for some time.

Algorithm 1 shows the procedure for Q-Learning [24]. The learning algorithm uses three parameters — the learning rate α, discount factor γ, and the ϵ-greedy parameter. The learning rate places a limit on how quickly learning occurs. If this parameter is set too low, it will take a long time for the system to learn, while if set too high, will cause the Q-Values to never converge to optimal values. The discount factor is used to determine how much emphasis is placed on future rewards. Setting this parameter low will optimize for immediate rewards, while setting this parameter high will place more importance on future rewards. Parameter ϵ determines with what probability a random action is explored, rather than choosing an action with the highest Q-Value.

3.2 Designing a Unified Link Layer using Q-Learning

We now describe how Q-Learning can be used to adapt between different radios in the case of a dual-radio sensor node. As noted earlier, we assume the availability of a multi-radio platform with complementary radios (e.g. a short-range low-power and long-range high-power radio). In Chapter 5, we describe one such platform that we have designed that combines a CC2420 and XE1205 radio. We also assume a traffic model where mobile nodes periodically report sensor readings and transfer data between each other when in range.

We first consider the case where each radio is set to a single power level. In this case, Q-Learning uses a two state model (one for each radio) where the action taken by the agent is either to stay with the same radio or switch to the alternate radio. The agent will switch radios if conditions deteriorate on the current radio (or is disconnected), or if conditions improve on the alternate radio. These dynamics are captured in the Q-Matrix at a rate governed by α and γ through feedback from packets transmitted over the current radio, as well as from exploration packets transmitted over the other radio (with frequency determined by ϵ). The optimal rate exploration packets should be sent is determined largely by mobility rate and interference dynamics; ideally this rate would be determined adaptively. If the agent finds that the alternate radio has a higher Q-Value (*i.e.* lower energy consumption), the agent will choose to use this new radio interface.

This two state model may be expanded to an *n-state model*, where each state represents a radio at a particular transmit power level, each representing a particular range/power tradeoff. For example, four states would be required for two radios, each with two transmit power level options. However, increasing the number of states comes at the cost of either increased exploration overhead or decreased exploration frequency since exploration requires time and energy. We reduce this overhead in the n-state case by considering only three states at a time — the current state and two adjacent states, a lower-range/lower-power state, and a higher-power/higher-range state. Both these adjacent states could be on the same radio or a different radio. Thus, exploration is limited to only two states at any time.

Reward matrix: A key aspect of Q-Learning adaptation is defining the reward matrix R for each state. The unified link layer receives information about the number of retransmission attempts and number of congestion backoffs for each packet that it transmits through either radio; these metrics are used to determine the reward for the current choice of radio / power level. We model the reward as an estimate of the amount of energy associated with the channel metrics collected for a given packet. The amount of energy to transmit a given packet is a function of packet size, static radio parameters such as receive/transmit power and channel sense time, number of retransmission attempts, and the number of congestion backoffs. Energy is a cost, rather than a reward, so its value is negative. The following equation shows how rewards are calculated where i is the number of retransmissions:

$$\begin{aligned} r[i] \quad = \quad & -(i \cdot (PacketSize \cdot ByteTime \cdot TxPower + \\ & \ AckTimeOut \cdot RxPower) \\ & +RxPower \cdot (AckRTT) \\ & +PacketSize \cdot ByteTime \cdot TxPower) \end{aligned}$$

While the above equation determines the reward when a packet is *successfully* transmitted, we also need to consider the case when a packet is unsuccessful after a pre-defined maximum number of retries. In this case, we want the Q-Learning algorithm to progressively try higher power states until it reaches the highest power state. To obtain this behavior, when a packet transmission is unsuccessful on a low-power state, we assign a large negative reward to encourage the algorithm to switch to a higher power state sooner, thereby limiting the number of lost packets. We achieve this behavior by selecting a policy for choosing an action, a, from the set of actions, s, with the maximal Q value or the minimal expected energy consumption for a packet transmission. Once the highest power state is reached, if packet transmission is still unsuccessful, a zero reward is assigned since there is no point in switching back to other lower power states until the connection is re-established at the high-power state.

CHAPTER 4

A MULTI-RADIO SWITCHING PROTOCOL

Translating the Q-Learning based switching algorithm to a working protocol presents a non-trivial challenge. When a sender decides to switch to or explore another radio, it needs to notify the receiver of such an action. However, an explicit handoff may not always work, for example, the receiver may be unreachable by the current radio due to mobility. A trivial solution would be for the receiver to keep both radios always active, obviating the need for handoff. However, this option is clearly inefficient as it requires both radios to be in receive mode, consuming significant energy. Thus, a key challenge that we address is: *how can we design a practical protocol for switching between radios that is energy-efficient and reliable?*

In the rest of this chapter, we describe the sender and receiver side design for our adaptive multi-radio block transfer protocol. For simplicity, we consider a dual-radio system with a high-power radio (HIGH) and low-power radio (LOW) with only one power level per radio.

4.1 Sender State Machine

The state machine at the sender is shown in Figure 4.1. We first describe the normal operation of the state machine before discussing how we handle exceptional cases that arise due to losses and disconnections. When data transfer starts, the sender first needs to "wakeup" the receiver from its IDLE state. There are many approaches to duty-cycling and wake up (e.g. SMAC [27], BMAC [16]), and we assume that one of these approaches

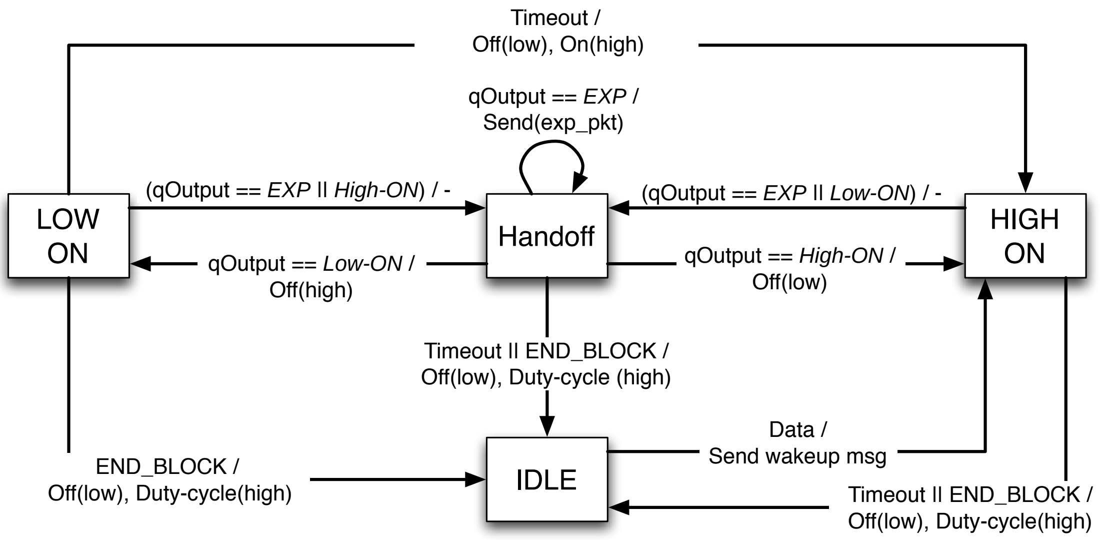


Figure 4.1: Sender state machine. qOutput denotes the output of the Q-Learning algorithm, which can be either explore, turn on low-power radio (low), or turn on the high-power radio (high). Transitioning from the IDLE state requires a wakeup message.

are available for the radio. Once the wakeup command is successful, the sender transitions from IDLE to HIGH-ON state.

Switching and exploration between the radios requires a handshake between the sender and receiver; first, the sender sends a packet indicating that a switch needs to be done, and if the packet is transmitted successfully, the sender and receiver can synchronously switch states to the second radio or explore on it. To perform such a handshake, the sender state machine includes a handoff state in which both radios are turned on. To illustrate, consider a switch from the HIGH-ON to LOW-ON state triggered by the Q-Learning algorithm. The state machine first sends a data packet while remaining in the current state with the *handoff flag* set. If the packet is successfully transmitted, the state machine transitions to the HANDOFF state. (Note that the receiver is in the BOTH-ON state at this point and can receive on both radios). From this state, the sender can send a packet on the LOW radio to transition to LOW-ON state. A similar process is done during exploration. The sender and receiver transition synchronously to the HANDOFF and BOTH-ON states respectively,

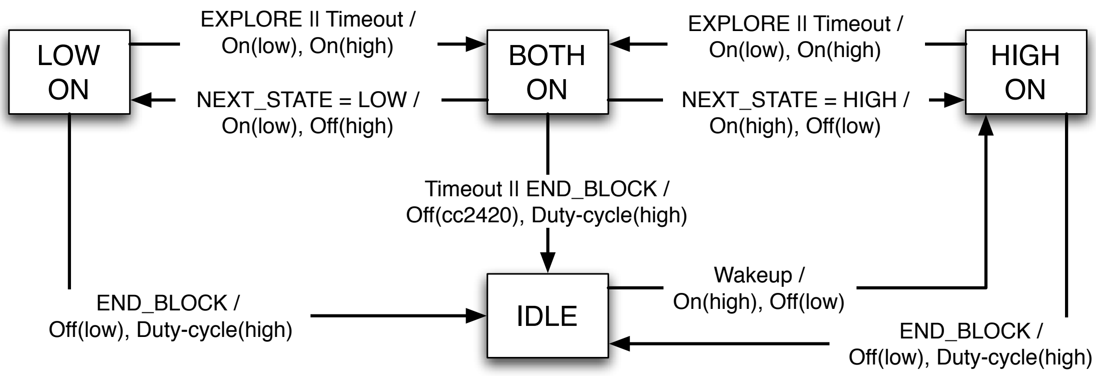

LOW ON
EXPLORE || Timeout / On(low), On(high)
NEXT_STATE = LOW / On(low), Off(high)
BOTH ON
EXPLORE || Timeout / On(low), On(high)
NEXT_STATE = HIGH / On(high), Off(low)
HIGH ON
Timeout || END_BLOCK / Off(cc2420), Duty-cycle(high)
END_BLOCK / Off(low), Duty-cycle(high)
IDLE
Wakeup / On(high), Off(low)
END_BLOCK / Off(low), Duty-cycle(high)

Figure 4.2: Receiver state machine

and stay in this state until exploration is complete, after which they switch back to whatever state they were in earlier.

Finally, we also deal with various cases where the state machines at the sender and receiver may become out-of-sync due to lost packets/acks, or complete loss of connectivity on one or both radios. If the LOW radio is currently in use and becomes disconnected, the sender times out, transitions to the HIGH-ON state and attempts to transmit using the long range radio. (Note that the receiver switches to the BOTH-ON state after a similar timeout, and is ready to receive on the HIGH radio.). If this fails as well, then after another timeout, the sender switches to IDLE mode since it means that the sender and receiver are out-of-range of both radios.

4.2 Receiver State Machine

The state machine at the receiver is shown in Figure 4.2. When data transfer starts, the receiver is in the *IDLE* state, where it operates with the HIGH radio in duty-cycled mode, and the LOW radio in off mode. This enables wakeup by the long-range radio to maximize contact time between the sender and receiver. The receiver is woken up out of this state by a long preamble on the HIGH radio, and switches to the *HIGH-ON* state. Switching between the two radios occurs through a handoff state where both radios are switched on and ready to receive. When the receiver gets a packet with the *handoff* flag set, it transitions to the BOTH-ON state. It stays in this state until the sender informs the receiver to switch to either the LOW-ON or the HIGH-ON state. The receiver transitions back to the IDLE mode when the END_BLOCK flag is set in a packet indicating that the sender has completed the current transfer of a block.

The receiver state machine also handles a number of exceptional cases that may arise. When the receiver is in the LOW-ON or HIGH-ON state and does not receive a packet for a short duration, it transitions to the BOTH-ON state. This enables the receiver to deal with two cases: (a) the sender is using one radio whereas the receiver is out-of-sync and listening on the other radio, (b) the sender is out of range of the current radio but in range of the other radio. If no packet is received in the BOTH-ON state, it implies that the sender has dropped out of contact of both radios, therefore the receiver switches back to the IDLE state.

Summary of benefits: Having described the sender and receiver state machines, we now briefly describe the main benefits of our switching protocol.

- *Active mode efficiency:* During a block transfer, we minimize the amount of time for which both radios are turned on at the sender and receiver. This ensures that our system almost always consumes only as much energy as a single radio system.

- *Low packet overhead:* All state transitions in our protocol are triggered by flags set in data packets. There are no additional control packets, hence our protocol has extremely low packet overhead.

- *Robustness:* Our protocol is robust to channel vagaries and different mobility patterns, and can recover from lost packets/acks, disconnections, and out-of-sync states.

CHAPTER 5

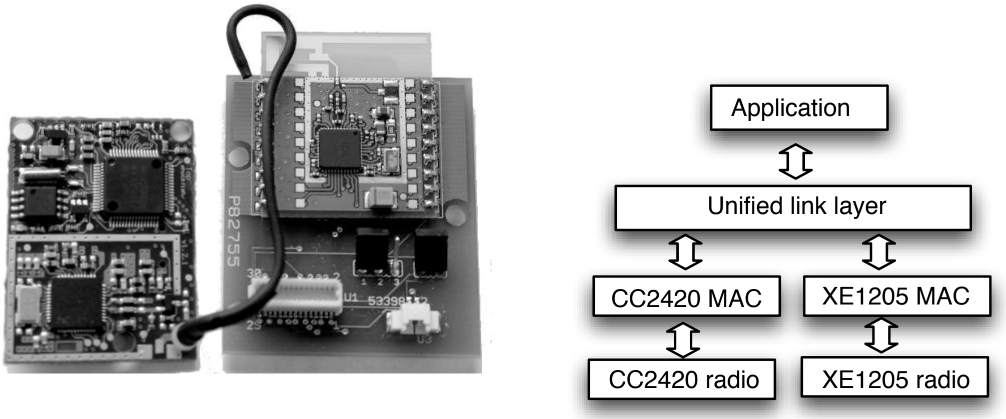
ARTHROPOD IMPLEMENTATION

We have built a prototype multi-radio platform called *Arthropod* and have implemented the Q-Learning based adaptive link-layer and switching protocol. This chapter describes the hardware and software implementation of our system.

5.1 Hardware Architecture

Our *Arthropod* sensor platform consists of a low-power microcontroller and a pair of heterogeneous low-power radios. The current prototype employs a MSP430 microcontroller, a CC2420 radio, and an XE1205 radio. Rather than constructing such a platform from scratch, we employed an existing *Tinynode* sensor platform [6], which contains a MSP430 processor and the XE1205 radio, and augmented it with a custom-built daughterboard comprising the CC2420 radio. We constructed the daughterboard by connecting an EasyBee CC2420 evaluation board [19] to several GPIO pins and an SPI bus available on the Tinynode. Figure 5.1(a) depicts the resulting prototype hardware of *Arthropod*.

The particular choice of the XE1205 and the CC2420 radios was governed by their complementary characteristics (see Table 1.1). The two radios operating in mutually exclusive frequency bands—900MHz and the 2.4GHz for the XE1205 and CC2420, respectively—enabling better interference adaptation. The table also shows that when operating at 0 dBm, both radios yield a range of 80m. However, the energy figures also indicate that if 5 or more retransmissions are needed on the CC2420, it is cheaper to use the XE1205 instead. In practice, retransmissions are more expensive since the sender needs to keep the radio active to receive the acknowledgment.



(a) Hardware Prototype (b) Software Architecture

Figure 5.1: System Components: (a) Hardware prototype comprising the Tinynode and a CC2420 expansion board and (b) Unified Link Layer for the radios

Arthropod also enables range diversity. While the peak range of the XE1205 is 2 kms for a bandwidth setting of 1.2 Kbps and +15dBm power level, we were unable to get reliable transmission on the XE1205 at this setting due to known calibration problems with the TinyNode's XE1205 radio. Therefore, we use a data rate of 38.1 Kbps @+15dBm, at which setting the maximum range is 800m. In contrast, the CC2420 cannot transmit beyond 0dBm and thus has a maximum range of 80m.

5.2 Software Architecture

The software implementation for *Arthropod* is an adaptive link-layer that unifies the individual MAC layers for the two radios. We have implemented a unified radio interface as part of the TinyOS-2.x operating system for motes [25]. The unified radio interface consists of two primary components: TinyOS-2.x drivers for the XE1205 and CC2420 radios and a unified link layer that manages the radio drivers. Figure 5.1(b) shows the

	XE1205 Radio	CC2420 Radio
Ack RTT	1.79ms	1.04ms
Ack Timeout	2.6ms	2.4ms
Avg. Congestion Backoff	10.4ms	10.4ms
Channel Sense Time	1.6ms	.756ms

Table 5.1: Timing for MAC layer operations

arrangement of these software components. More implementation details may be found in [7].

TinyOS-2.x Radio Drivers: The interfaces to the CC2420 and XE1205 radios allow for fine-grain control of many parameters including link layer acknowledgments, clear channel assessment, radio channel selection, data rate and transmit power. Table 6.6 shows some performance metrics collected from the corresponding drivers that are relevant to our multi-radio system. Each radio MAC layer supplies feedback required by our unified link layer: whether or not a packet was acknowledged, and the number of congestion backoffs experienced for the current transmission attempt. Table 6.6 shows performance metrics relevant to our multi-radio system.

Unified Link Layer: In order for our multi-radio platform to be usable at an application level, our system provides a unified link layer that determines the radio interface currently best suited for communication (shown in Figure 5.1(b)). To the programmer, there is a single interface to send and receive packets to and from the node, and the link-layer system handles addressing and transmission over the individual radios, hidden from the programmer. Our unified link layer takes care of monitoring channel conditions for each packet transmission, and determines which radio interface is currently most energy-optimum. At its core, this link layer is implemented using our Q-learning algorithm that uses feedback from the individual radio MAC layers to build a history and determine which radio is performing best. Our link layer also implements the multi-radio switching protocol described in the previous chapter. One existing limitation of our unified link-layer is that energy optimizations do not consider the energy being expended by the receiver. Several possible solutions to this limitation are discussed in Chapter 7.

Duty-cycling: Our implementation of the switching protocol uses the low power listen (LPL) protocol for duty-cycling radios [16]. In this approach, the sender can wakeup the receiver in a completely asynchronous manner by sending a long preamble that is at least as long as the sleep cycle of the receiver. The sender uses such a long preamble to "wakeup" the receiver and initiate the block transfer.

While LPL is available as part of both the CC2420 and XE1205 radio stacks in TinyOS 2.0, we experienced several problems with the XE1205 LPL implementation. For example, the XE1205 interface would occasionally silently drop a packet transmission while reporting successfully acknowledged delivery to the receiver. To circumvent this problem, the IDLE state switches off the long-range XE1205 radio and duty-cycles the CC2420 radio. This is not ideal for reasons outlined in Chapter 4 — wakeup using the long-range radio ensures greater contact duration between nodes.

CHAPTER 6

EXPERIMENTAL EVALUATION

In this chapter, we present a detailed evaluation of the Q-learning based unified link layer using a combination of experiments using data traces, results from Q-Learning running on an *Arthropod* mote, as well as implementation benchmarks. Our evaluation has four parts. First, we evaluate the performance of the Q-learning link layer in adapting to a diverse set of mobility patterns. Second, using traces we evaluate how well the learning algorithm handles power control across the two radio interfaces. Third, we evaluate the efficacy of Q-learning for handling interference dynamics. Finally, we present benchmarks from an implementation of the link layer for an *Arthropod* mote to demonstrate that the described Q-Learning algorithm is efficient and has low resource usage.

6.1 Datasets

To ensure repeatable experimentation of the link layer, we gathered datasets under different conditions using our hardware prototype. We obtained four types of datasets that are a good representation of mobility patterns found in mobile sensor network deployments. We then gathered four more datasets that represent different interference dynamics. Table 6.1 contains a brief summary of the four mobility datasets collected, while Table 6.2 summarizes the interference datasets.

The datasets were obtained from two *Arthropod* motes - one *Arthropod* mote sends 20 byte packets with increasing sequence numbers over both radio interfaces at a fixed rate (2 Packets / second). A second mote places both radios in receive mode and acknowledges all packets received on each radio interface. For each packet, the sending node records the

Environment	Mobility Pattern	example scenario
urban-indoor	continuous w/ obstructions	people in a building
urban-outdoor	continuous partial LOS	moving vehicle
urban-outdoor	nomadic	bus w/ stops
foliage	nomdic	bus w/ stops

Table 6.1: Brief summary of mobility traces.

number of congestion backoffs experienced while trying to send the packet, as well as the number of retransmissions before receiving an acknowledgement from the receiver. The maximum number of backoffs is set to 6, and retransmissions is set to a limit of 10, after which the link layer at the sender gives up on the transmission.

In addition to link layer statistics we also store the radio chosen by the Q-learning algorithm for each packet. This allows us to verify that the algorithm is functioning correctly by comparing the decisions to the losses seen on each interface.

For the mobility datasets, we configure the long-range XE1205 radio to a data-rate to 38.1 kbps and power level 15 dbm, whereas the short-range CC2420 radio transmits at the default 250 kbps at 0dbm. The traces are obtained by having both the radios transmit packets back to back at the rate of about 2 pkts/second. For each packet, the number of backoffs and retransmissions are logged in the local flash memory of the sender, and retrieved later.

6.1.1 Traces showing continuous mobility

We obtained datasets with continuous mobility to represent two practical sensor application scenarios: wearable sensors and vehicular sensor networks.

- **Indoors:** This trace was collected indoors within our department. The receiver is stationary while the sender moves up and down the length of the corridor of the building (120 meters) transmitting packets while moving at a normal walking speed.

The nodes are obstructed from each other by numerous walls as they move apart. The up-and-down movement pattern is repeated five times.

This trace was collected in our research building with the receiving node stationary in our lab. The sending node travels up and down the length of the corridor of the building (120 meters) transmitting packets while moving at a normal walking speed, transmitting at a rate of 2/second. The relative positions of the nodes cause an increasing number of walls to obstruct communication as the node moves away from the receiver. Analysis of the trace shows that the number of retransmissions required to successfully acknowledge a packet reaches its limit before reaching the end of the corridor; this indicates that the trace is suitable for showing transitions between the two radios using the Q-Learning algorithm. To verify the algorithm never becomes ”stuck” on a particular interface, we repeat the movement pattern five times.

- **Outdoors:** This trace was collected on a stretch of road outside our department. The receiving node was placed on a bus stop shelter; the sender approaches the receiver from 600 meters, initially obstructed by buildings and foliage. The node briefly pauses at the bus shelter and continues down the street another 300 meters disappearing behind foliage and buildings. The sender moves at a rate of roughly 9 meters/second.

6.1.2 Traces showing nomadic mobility

Nomadic mobility refers to the case where nodes move for some time, pause at a specific location for a while, and continue in the same pattern. This type of movement behavior is common amongst sensors monitoring people or a habitat sensor network. We obtained two such traces:

- **Urban Outdoor:** This trace was collected at our campus near some large HVAC buildings and parking lots. The sender starts in close proximity to the receiver and moves away at a normal walking speed. The sender pauses for a minute at locations

Dataset	Interference Type	Description
Long Interference	30 minute periods	indoors; 100ft range; XE1205@0dBm
Medium Interference	200-235 packet bursts	indoors; 100ft range; XE1205@0dBm
Short Interference	25-32 packet bursts	indoors; 100ft range; XE1205@0dBm
Low Interference	ambient	indoors; 100ft range; XE1205@0dBm

Table 6.2: Summary of interference traces.

50m, 60m, 80m, and 70m away from the receiver, and finally returns to the receiver location. Line of sight is limited during this trace resulting in poor performance for the CC2420 radio.

- **Habitat Outdoor:** This trace was collected outdoors in a wooded rural area with significant foliage. The sender starts 100 meters from the receiver and approaches the sender at a slow walking speed. The receiver pauses for 2 minutes near the receiver, and then moves away at a slow walking speed to a location 100 meters away. At all locations, significant number of trees introduce signal attenuation. The XE105 has good connectivity for the entire experiment, while the CC2420 moves in and out of range.

6.1.3 Traces showing interference dynamics

We obtained datasets with interference at four different time-scales, all representing an urban-indoor environment. To generate controlled interference on each radio channel, we arrange a set of jamming motes close to both the sending and receiving nodes. The Telos and Tinynode interferer motes run a program that sends packets at 0dBm as quickly as possible with CSMA/CA disabled on the CC2420 and XE1205 radios respectively.

An adjustment to the Tinynode radio stack was necessary for the interfering nodes: The packet preamble was modified to prevent the XE1205 on the sending *Arthropod*from dominating the SPI bus shared with the Flash.

- **Long interference trace:** This trace represents a scenario where one radio has a clear communications channel for a long period of time (30 minutes, 1800 packets) during which the second radio is being interfered with, followed by a period where the second radio has a clear communications channel for a long period (30 minutes) while the first radio is interfered with. For the final 30 minutes, all interfering nodes remain silent and any losses or congestion backoffs that occur are due to ambient interference.

- **Medium interference trace:** This trace demonstrates a scenario where channel conditions are changing more rapidly and favors different radios at different times. Interference is generated in shorter bursts such that each radio sees alternating busy and clear channels varying from 200 - 235seconds.

- **Short interference trace:** This trace represents a scenario where even shorter bursts of interference appear on each channel. This dataset was collected using an identical setup to the previous case, but with active and silent periods 25 - 32 seconds in length.

- **Low interference trace:** This trace represents a scenario where no motes are transmitting and the interfering nodes are completely turned off. The only backoffs and retransmissions observed, were due to cross-traffic from other wireless devices such as 802.11, which was very limited since the experiment was performed during the evening and the CC2420 was configured to use IEEE 802.15.4 channel 26, which does not overlap with 802.11 channels.

6.2 Evaluation of Q-Learning for Mobility Dynamics

These experiments evaluate the performance of the Q-learning algorithm in a MATLAB simulation environment and its performance for the various mobility traces described above. To get an accurate measure of the performance of the Q-learning based link layer, we emulate the behavior of the sender and receiver state machine (Chapter 3) given the

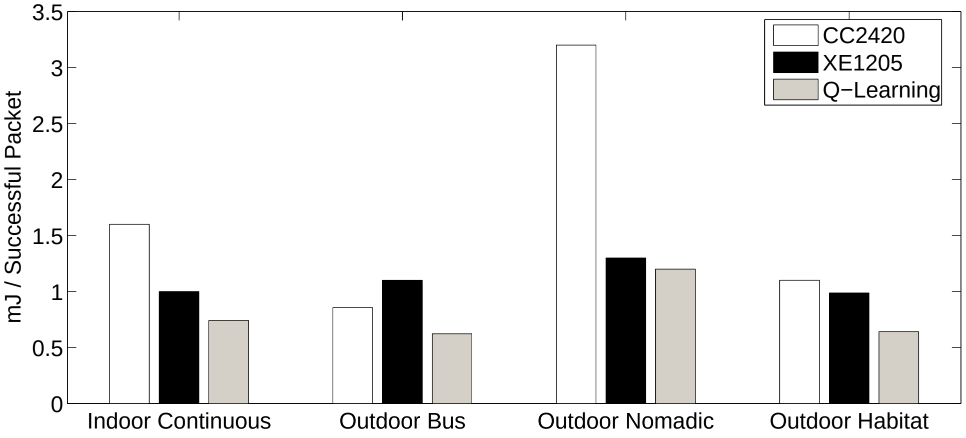


Figure 6.1: Energy consumed per successful packet for each dataset and strategy

sequence of packet losses observed in the traces. (Later, in Section 6.5 we show that this emulation accurately corresponds to the performance of the real protocol in practice.)

For all datasets, we used an identical set of Q-Learning parameters: $\alpha = 1.0, \gamma = 0.7$ and $\epsilon = 0.025$. These parameters were chosen since they seem to work well across a range of mobility datasets.

Q-Learning Performance: Figure 6.1 and Figure 6.2 summarize the energy per successful packet transmission and loss rates observed by our adaptive multi-radio link layer in comparison with using just one of the radios. In terms of energy consumption, the Q-Learning approach reduces energy consumption compared to the XE1205 radio by an average of 27% (the maximum reduction is 53.6% for the Outdoor Bus dataset), while incurring roughly 2-4% increased loss across the four cases. The slightly increased loss rate of Q-learning is caused by exploring an alternate interface periodically and transient losses caused while the algorithm is still learning. Similar energy gains of a maximum of 62.5% and an average of 44.6% are obtained over an approach that just uses the CC2420 radio but the improvements in loss rate are significantly higher (25%-60%). The results show that in all cases, the energy consumption of the adaptive multi-radio link layer is better than

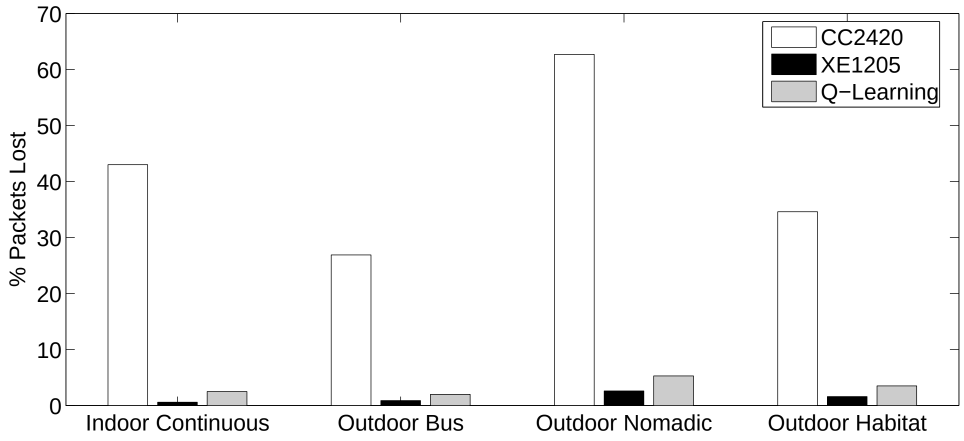


Figure 6.2: Percent Packets lost for the two radio interfaces and Q-Learning implementation

exclusively using either the CC2420 or XE1205 radio, while keeping the link loss rate to be close to that observed by the long-range XE1205 radio.

As can be seen, the worst case for the Q-Learning protocol is the outdoor nomadic trace where our benefits are only marginal in terms of energy. This is because connectivity using the CC2420 radio is highly sporadic and also very lossy (65% loss). Thus, our link layer is unable to take advantage of the CC2420 radio due to the high dynamics on it.

In summary, our results show that Q-learning can provide significant gains in terms of energy while only increasing packet loss marginally; when an opportunity arises for communication over the CC2420 radio interface, our unified link layer is capable of exploiting its increased energy efficiency.

6.3 Algorithm performance for power control across radios

A logical extension to the unified link layer is handling transmission power control in addition to radio selection. The CC2420 radio is capable of transmitting packets from -25dBm up to 0dBm, while the XE1205 can transmit from 0dBm to 15dBm. Increasing transmit power will provide longer range connectivity but uses additional energy; the op-

radio/power-level	% packets lost	energy consumed
XE1205@0dBm	4.24	.659mJ/Tx Success
XE1205@15dBm	0	.925mJ/Tx Success
CC2420@-25dBm	37.01	1.1mJ/Tx Success
CC2420@0dBm	35.45	1.2mJ/Tx Success
Q-Learning	3.53	.430mJ/Tx Success

Table 6.3: Statistics for Dual Radio / Power Control Trace

timum strategy will choose the minimum transmit power level on the most efficient radio without significantly increasing loss rate.

To evaluate power control across radio interfaces, we collected a packet trace similar to the *Indoor Continuous* described earlier. In addition to logging retransmissions and backoffs for the XE1205@15dBm and CC2420@0dBm, we log similar statistics for the XE1205 and CC2420@0dBm and -25dBm respectively. The number of states in the Q-Learning algorithm increases from 2 to 4; we maintain a Q-value for each radio/power level combination. To reduce exploration overhead, we only explore the radio/power combinations adjacent to the current setting. Logically, the next setting expected from a mobile node would be one higher if the distance between sender and receiver has increased and one lower if the distance has decreased. Such an approach would scale even if there were more power states being considered per radio.

Table 6.3 summarizes the results and compares the Q-learning approach to just using one of the two radios at one of the power levels. As can be seen, Q-learning is 54% better in terms of energy consumption per successful transmission than only using the XE1205 radio at 15dBm but has comparable loss rate. The energy benefits over using the CC2420 radio are 64%; the loss rate also reduces by an order of magnitude. Overall, the Q-Learning based adaptive algorithm sends roughly 40% and 10% of the packets using XE1205 at 0 dBm and 15 dBm; and 25% of the packets on the CC2520 at -25 dBm and 0 dBm. The results validate that Q-Learning is able to utilize each power state opportunistically.

These results show that the unified link layer is very effective at handling power control across multiple radio interfaces. Our scheme uses 2.5x less power than the highest power radio with only a negligable packet loss increase. Each setting is used effectively at the appropriate range.

6.4 Evaluation of Q-Learning for Interference Dynamics

Our next set of experiments evaluate the performance of the Q-learning algorithm and its adaptability to changing interference conditions. Like before, these experiments were performed in MATLAB, using the data traces described in the previous section as input.

We compare the Q-learning algorithm against two alternate techniques. As a baseline, we use an *omniscient* strategy that has knowledge of the complete dataset, and is always able to make an optimal decision regarding which radio should be used. It is important to note that it is impossible for this strategy to be realized in practice - it is only used as a baseline to gauge how closely Q-Learning performs relative to such a best case strategy. The second is a non-adaptive *naïve* approach that only looks at the first 100 packets sent across each radio interface. This approach takes the ratio of cumulative energy consumption expended by each radio for this set of packets, and determines a probability p, corresponding to the the ratio of energy expended by the CC2420 vs XE1205 radio. For all other packets, the link layer transmits the packet on the CC2420 radio with probability p, and on the XE1205 radio with probability $1 - p$.

Temporal Adaptability: First, we illustrate the adaptability of Q-Learning using a time-series plot of the medium interference trace, shown in Figure 6.3. The staircase shape of the CC2420 and XE1205 radios cumulative energy consumption is caused by bursts of interference introduced by the Telos motes and Tinynodes. The portions of the plot with a steep slope indicate that the radio is consuming an increased amount of energy per packet because of congestion and interference. Every time interference is encountered, this causes an entry in the Q-Matrix to grow increasingly negative at a rate determined by the

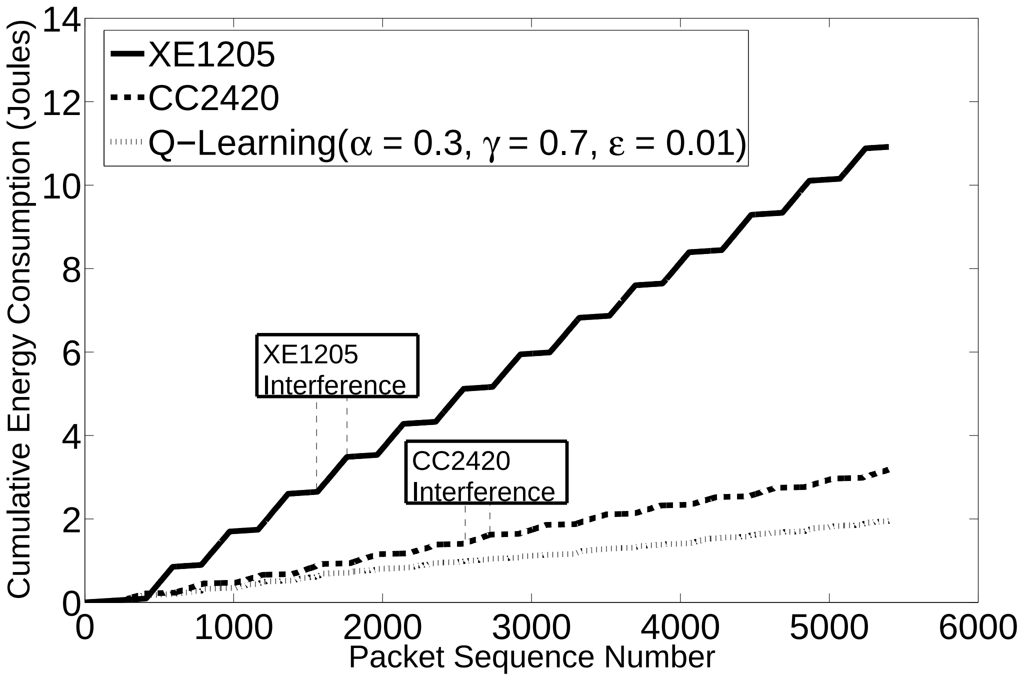


Figure 6.3: Cumulative energy consumption for Long Bursts of Interference

reward r (energy consumption) and Q-Learning parameters α and γ. Q-Learning settles on the radio with best channel characteristics, and periodically explores the other radio ($\epsilon = 0.01$). This gives Q-Learning the opportunity to adapt when communication patterns change in the network. As a result, the Q-Learning plot represents a hybrid of the CC2420 and XE1205 energy plots, where the radio with a minimum slope is chosen during each burst after a brief learning period, resulting in a smoothing of energy consumption over time with consistently better performance than each individual radio.

Aggregate Q-Learning Performance: Figure 6.4 summarizes how Q-learning performs relative to other schemes across all the datasets. For both long and medium term interference, Q-learning performs extremely well compared to choosing a single radio or the multi-radio naïve scheme. For long-term interference, the XE1205 radio consumes 4.2 times more energy than our system, the CC2420 consumes 4.8 times more energy, and the naïve algorithm consumes 4.5 times more energy. For medium time-scale interference, the

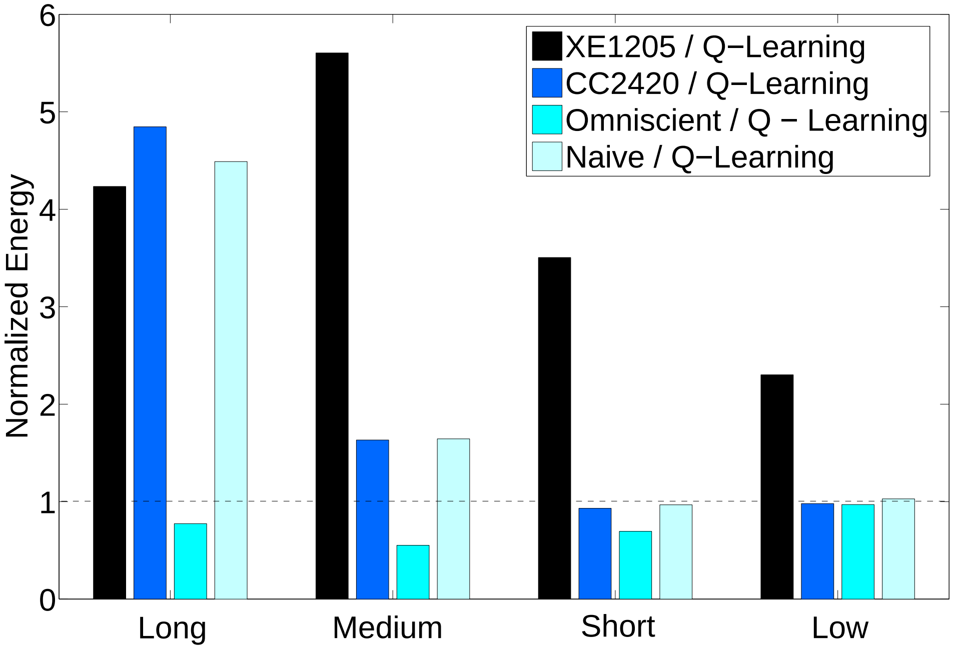


Figure 6.4: Relative performance of Q-Learning for different interference patterns: (1) Long, (2) medium, (3) short, (4) low external interference

XE1205 radio alone consumes 5.6 times more energy than is consumed by our algorithm, and the CC2420 radio consumes roughly 1.6 times more energy, while the naïve two-radio algorithm also uses around 1.6 times more energy. The gains are intuitive since the naïve algorithm is non-adaptive and assumes that the behavior across the first 100 packets will be representative of future channel conditions. The Q-learning algorithm, however, is not as efficient as the omniscient approach which consumes 77% of the energy used by Q-Learning in the case of long-term interference and 69% of the energy used by Q-Learning in the case of short-term interference. This is because the omniscient algorithm wastes no energy exploring the two channels and also is never impacted by channel dynamics.

The short interference trace represents a case where Q-learning can be expected to perform badly since the channel is switching behavior every 30 packets. When we choose the exploration factor, ϵ, to be small, it is difficult to learn with few samples that the current radio-channel has become poor and the other radio channel has improved from its previous

Switching Protocol State	% Time Spent
HIGH ON	10.7
LOW ON	78.1
BOTH ON	11.2

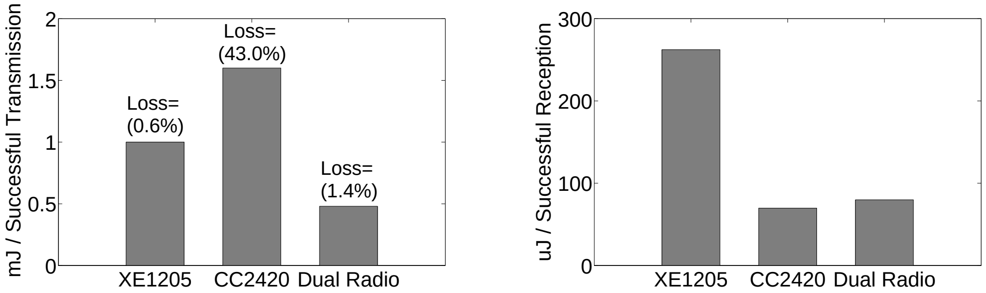
Table 6.4: Time Spent During Different Receive States

state. When undersampling the channel in this manner, the learning algorithm eventually converges on following the radio that uses less overall energy, but does not make improvements beyond the better of the two radios. The low interference trace represents a scenario where the environment has limited dynamics since there is very little external interference. For this scenario, our Q-Learning algorithm will choose the more energy-efficient radio (CC2420) and encounter a slight amount of overhead as a result of exploring the more energy-expensive channel periodically. In both these cases, Q-learning is almost as efficient as choosing the best radio.

In summary, our results show that Q-learning can provide significant performance gains when there is medium and long term interference that is greater than 200 packets in length. Even in hard to learn conditions such as short bursts of interference, and low interference conditions with limited dynamics, Q-Learning performs only marginally worse than the better of the two radios.

6.5 Implementation Results

To validate our implementation and show the performance of the radio switching protocol, we collect a new dataset with the same mobility pattern as the *indoor continuous* dataset. For this experiment, a pair of sender and receiver nodes run the switching protocol and Q-Learning algorithm online. This study aims to measure the actual per packet energy costs incurred by the sender and receiver. In particular, the receiver can become out-of-sync with the sender, resulting in the receiver turning both radios on, or timing out, all of which costs energy and results in more packet losses. For the mobility rates used in our



(a) Sender

(b) Receiver

Figure 6.5: Energy spent per packet by the sender and receiver. Labels on sender bars indicate packet loss rates.

experiments we chose a timeout value of 2 seconds. Finally, we also breakdown the % time spent by the receiver in different states of the receiver state machine.

The maximum data rate achievable by our software implementation for a pair of nodes transmitting continuously is 70kbps. However, while logging packet statistics to the external flash, the data rate reduces to 14kbps. This lower data rate is a result of the Tinynode platform multiplexing the SPI bus between the XE1205 radio and the external flash memory. As a result, the sender and receiver are not continuously sending data which causes idle gaps to appear between packets. This forces the receiver to expend additional energy while waiting for packets to arrive. Since the idle time is an artifact of our evaluation, we ignore these periods when presenting results.

To understand the energy efficiency of our protocol at the sender, Figure 6.5(a) compares the energy consumed by a single radio strategy to that of the dual radio implementation. The per packet energy consumption numbers presented for the CC2420 and XE1205 only cases are from Section 6.2. The results in Figure 6.5(a) show that our adaptive algorithm is 64% more efficient than a CC2420-only scheme, and 43% more efficient than an XE1205-only scheme verifying the gains found in simulation. These energy efficiency gains are achieved while maintaining a loss rate of 1.6% which is not substantially higher than the XE1205 loss rate of 0.6% and much lower than the 43.0% loss rate of the CC2420

Task	Latency	Energy
Radio Selection	19.3us	104nJ
Q-Matrix Update	43.6us	235nJ
Transmit 20-Byte packet (XE1205 @ 0dBm and 76.8kbps)	3.9ms	218uJ
Transmit 20-Byte packet (CC2420 @ 0dBm)	1.7ms	92uJ

Table 6.5: Latency and Energy consumption for link-layer components

radio. These results validate our simulation study and show that substantial sender-side energy gains are acheivable by opportunistically using the CC2420 radio, while providing a loss rate comparable to that of the XE1205 radio.

Figure 6.5(b) shows the amount of energy consumed at the receiver as a result of the decisions made by the sender. The energy efficiency of the receiver will always fall somewhere between the efficiency of the XE1205 and CC2420 radios, depending on how often each is used. Bringing up both radio interfaces is an unavoidable result of the radio switching protocol and represents overhead beyond that of a single radio strategy. Additionally, transition times from sleep to idle/receive mode represent overhead. Our evaluation shows that the dual-radio protocol used 70% less energy than the XE1205, but 13% more energy than the CC2420. It is important to note that the receiver uses an order of magnitude less power than the sender, which means the sender-side gains overshadow the receiver-side losses.

Finally, we provide a breakdown of the percentage of packets the receiver spends in each state of the switching protocol in Table 6.4. The receiver spends 10.7% of time in the HIGH-ON state, 78.1% of time in the LOW-ON state and 11.2% of time in the BOTH-ON state. Ideally the radio switching protocol will only force the receiver into the BOTH-ON state while exploring or handing off between radios. Exploration accounts for 4% of this time, while the other 7.2% is caused by explicit handoffs and timeouts.

	XE1205 Radio	CC2420 Radio
Ack RTT	1.79ms	1.04ms
Ack Timeout	2.6ms	2.4ms
Avg. Congestion Backoff	10.4ms	10.4ms
Channel Sense Time	1.6ms	.756ms
Sleep to Active Mode Transition Time	1.5ms	0.58ms

Table 6.6: Driver performance for MAC layer operations

6.6 Microbenchmarks

In this section, we briefly discuss measurement-based latency and energy consumption microbenchmarks based on our implementation of the unified link layer. As shown in Table 6.5, the energy/latency overhead imposed on the CPU by our multi-radio adaptation algorithm implementation on the *Arthropod* is highly efficient and consumes less than a hundredth of the energy/latency of the radios used. This shows that the overhead introduced by software can be compensated by larger performance gains achieved through intelligent radio selection. The amount of memory overhead of our implementation is 111 bytes, which is a very small portion of the available 10kB. A much larger portion of program memory is required, however, because two radio stacks need to be instantiated; supporting an additional radio stack requires an additional 12kB resulting in a total usage of 29kB out of the available 48kB of program memory, although we believe that this can be optimized considerably.

We also micro-benchmarked MAC layer operations in our TinyOS drivers. Table 6.6 shows the individual components based on which the reward matrix is populated as described in Chapter 3. These measurements were used to compute the total energy cost of a transmitted packet and account for overheads in the state transitions of a receiver.

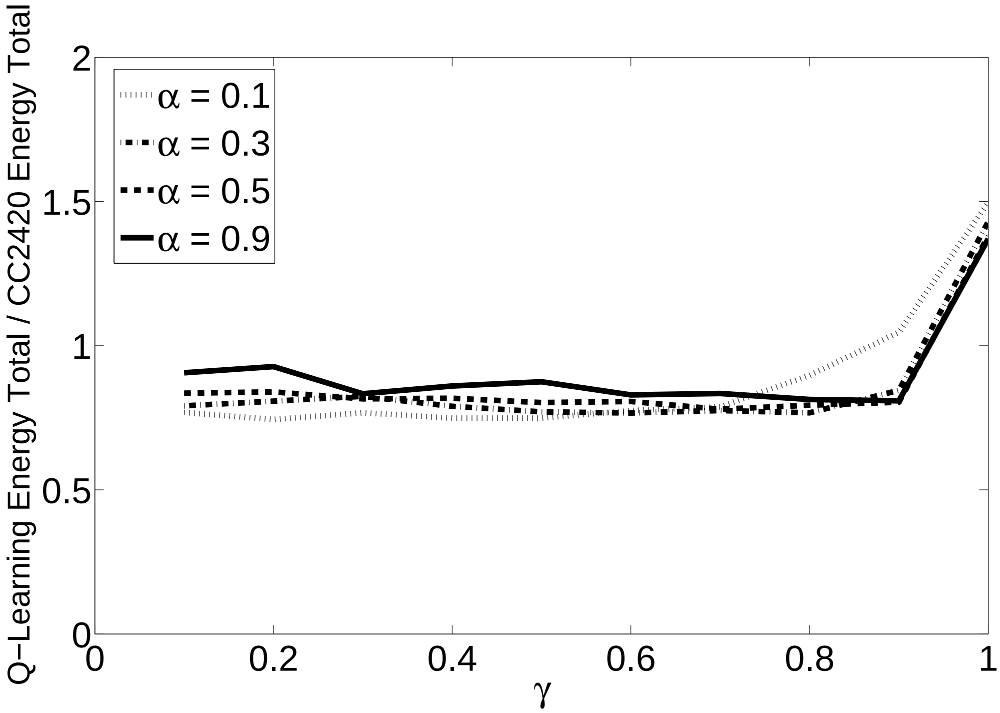


Figure 6.6: Cumulative energy performance increase for different parameter values

6.7 Parameter Sensitivity

In this section we study the sensitivity of Q-learning to its parameters: α, γ and ϵ. To demonstrate the algorithm's robustness to parameter variation, we use our mobility dataset to show how the total energy consumption of the Q-Learning algorithm changes with respect to changes in parameter values. Figure 6.6 shows a number of plots, where for each plot we fix α and vary γ. The plots are very similar across several different α values and demonstrate that as long as α and γ are chosen within a reasonable range, the performance of Q-Learning is stable. We repeated this procedure for the interference datasets, and found similar results.

Overall, we have found that a larger α value is generally helpful in mobility traces due to the need for fast switching. As the traces become more nomadic in nature (*i.e.* as they involve more waiting and less movement), the optimal choice of α reduces a little. However, $0.9 \leq \alpha \leq 1$ seems to be ideal in almost all settings. The choice of ϵ impacts how fast we can switch but it also impacts the energy consumption. A high ϵ can lead to more exploration overhead but is more reactive. We found that exploration roughly every 10 seconds or so provides a good balance but this can be tuned depending on expected dynamics. Finally, we found that the results were not very sensitive to γ, and works best in the range $0.5 \leq \gamma \leq 0.85$.

CHAPTER 7

CONCLUSIONS

In conclusion, we have made three major contributions in this thesis. First, we designed a new multi-radio sensor platform, the *Arthropod*, that pairs two radios - CC2420 and XE1205 - that offer diversity in frequency, power and range. Second, we presented the design of a Q-Learning-driven adaptive link layer that provides the abstraction of a single radio to the applications, and third, we presented a protocol that switches between radios depending on which radio offers the most energy-efficient communication channel. Experiments using a number of interference and distance datasets confirm that the system can provide effective adaptation to a range of dynamics. We also showed that the learning algorithm can be easily implemented with limited memory and computational overhead on a mote-class sensor platform.

BIBLIOGRAPHY

[1] Agarwal, Yuvraj, Chandra, Ranveer, Wolman, Alec, Bahl, Victor, Chin, Kevin, and Gupta, Rajesh. Wireless wakeups revisited: Energy management for voip over wi-fi smartphones. In *Proceedings of Mobisys* (Puerto Rico, USA, June 2007).

[2] Banerjee, N., Sorber, J., Corner, M. D., Rollins, S., and Ganesan, D. Triage: A Power-Aware Software Architecture for Tiered Microservers. In *Proc. Mobisys* (Puerto Rico, USA, June 2007).

[3] Banerjee, Nilanjan, Corner, Mark D., and Levine, Brian Neil. An Energy-Efficient Architecture for DTN Throwboxes. In *Proc. IEEE Infocom* (May 2007).

[4] Burgess, John, Gallagher, Brian, Jensen, David, and Levine, Brian Neil. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM* (April 2006).

[5] Draves, Richard, Padhye, Jitendra, and Zill, Brian. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom* (Philadelphia, PA, September 2004), pp. 114–128.

[6] Dubois-Ferrire, Henri, Meier, Roger, Fabre, Laurent, and Metrailler, Pierre. TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications. In *Information Processing in Sensor Networks (IPSN 2006)* (2006).

[7] Gummeson, J., Ganesan, D., Corner, M. D., and Shenoy, P. An adaptive link layer for heterogeneous multi-radio mobile sensor networks. Tech. Rep. UM-CS-2010-013, University of Massachusetts, 2010.

[8] Hsieh, Hung-Yun, and Sivakumar, Raghupathy. A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts. In *Proc. of ACM Mobicom* (Atlanta, GA, September 2002).

[9] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., and Rubenstein, D. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS* (San Jose, CA, October 2002).

[10] Jun, H., Ammar, M. H., Corner, M. D., and Zegura, E. Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization. In *Proc. ACM SIGCOMM Workshop on Challenged Networks (CHANTS)* (September 2006).

[11] Lymberopoulos, D., Priyantha, B., Goraczko, M., and Zhao, F. Towards energy efficient design of multi-radio platforms for wireless sensor networks. In *Proc. of IPSN, St. Louis, MO* (April 2008).

[12] Mishra, Nilesh, Chebrolu, Kameswari, Raman, Bhaskaran, and Pathak, Abhinav. Wake-on-WLAN. In *Proc. Intl Conf on the World Wide Web (WWW)* (2006), pp. 761–769.

[13] Navda, V., Subramanian, Anand P., Dhanasekaran, K., Timm-Giel, A., and Das, S. Mobisteer: using steerable beam directional antenna for vehicular network access. In *MobiSys* (New York, NY, USA, 2007).

[14] Pering, Trevor, Agarwal, Yuvraj, Gupta, Rajesh, and Want, Roy. CoolSpots: Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces. In *Proc. ACM MobiSys* (June 2006), pp. 220–232.

[15] Pering, Trevor, Raghunathan, Vijay, and Want, Roy. Exploiting Radio Hierarchies for Power-Efficient Wireless Device Discovery and Connection Setup. In *VLSI Design* (2005), pp. 774–779.

[16] Polastre, J., Hill, J., and Culler, D. Versatile low power media access for wireless sensor networks. In *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys)* (November 2004).

[17] Rahmati, Ahmad, and Zhong, Lin. Context-for-wireless: Context-sensitive energy-efficient wireless data transfer. In *Proceedings of Mobisys* (Puerto Rico, USA, June 2007).

[18] Redi, J., Solek, S., Manning, K., Partridge, C., Rosales-Hain, R., Ramanathan, R., and Castineyra, I. Javelen - an ultra-low energy ad hoc wireless network. In *Ad Hoc Networks, Vol. 6, No. 1* (January 2008).

[19] `http://www.rfsolutions.co.uk`. RF Solutions: EasyBee Zigbee Transceiver.

[20] Rodriguez, P., Chakravorty, R., Chesterfield, J., Pratt, I., and Banerjee, S. Mar: A commuter router infrastructure for the mobile internet. In *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services* (Boston, MA, June 2004).

[21] Sengul, C., Harris, A., Bakht, M., Abdelzaher, T., and Kravets, R. Improving energy conservation using bulk transmission over high-power radios in sensor networks. In *Proc. of ICDCS, Beijing, China* (June 2008).

[22] Shih, E., Bahl, P., and Sinclair, M. J. Wake on Wireless: An event driven energy saving strategy for battery operated devices. In *Proc. of ACM MobiCom* (Atlanta, GA, September 2002).

[23] Stathopoulos, Thanos, Lukac, Martin, McIntire, Dustin, Heidemann, John, Estrin, Deborah, and Kaiser, William. End-to-end routing for dual-radio sensor networks. In *Proceedings of the IEEE Infocom 2007* (Anchorage, Alaska, USA, May 2007).

[24] Sutton, Richard S., and Barto, Andrew G. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[25] TinyOS Website. http://www.tinyos.net/.

[26] `http://www.maxstream.com`. XTend OEM RF Module for Long-range Data Transfer.

[27] Ye, Wei, Heidemann, John, and Estrin, Deborah. An energy-efficient mac protocol for wireless sensor networks. In *In proc. of IEEE INFOCOM* (New York, NY, USA, June 2002), pp. 1567–1576.

[28] Zhu, J., Waltho, A., Yang, X., and Guo, X. Multi-radio coexistence: Challenges and opportunities. In *In proc. of ICCCN* (August 2007).