Report Number: wucse-2009-75

2009

Robust Sensor Networks in Homes via Reactive Channel Hopping

Mo Sha, Greg Hackmann, and Chenyang Lu

Home area networks (HANs) consisting of wireless sensors have emerged as the enabling technology for important applications such as smart energy and assisted living. A key challenge faced in deploying robust wireless sensor networks (WSNs) for home automation applications is the need to provide long-term, reliable operation in the face of the varied sources of interference found in typical residential settings. To better understand the channel dynamics in these environments, we performed an in-depth empirical study of the performance of HANs in ten real-life apartments. Our empirical study leads to several key insights into designing robust HANs for residential... Read complete abstract on page 2.

Recommended Citation

Sha, Mo; Hackmann, Greg; and Lu, Chenyang, "Robust Sensor Networks in Homes via Reactive Channel Hopping" Report Number: wucse-2009-75 (2009). *All Computer Science and Engineering Research.*
https://openscholarship.wustl.edu/cse_research/28

Robust Sensor Networks in Homes via Reactive Channel Hopping

Mo Sha, Greg Hackmann, and Chenyang Lu

Complete Abstract:

Home area networks (HANs) consisting of wireless sensors have emerged as the enabling technology for important applications such as smart energy and assisted living. A key challenge faced in deploying robust wireless sensor networks (WSNs) for home automation applications is the need to provide long-term, reliable operation in the face of the varied sources of interference found in typical residential settings. To better understand the channel dynamics in these environments, we performed an in-depth empirical study of the performance of HANs in ten real-life apartments. Our empirical study leads to several key insights into designing robust HANs for residential environments. For example, we discover that there is not always a persistently good channel over 24 hours in many apartments; that reliability is strongly correlated across adjacent channels; and that interference does not exhibit cyclic behavior at daily or weekly timescales. Nevertheless, reliability can be maintained through a small number of channel hops. Based on these insights, we propose Adaptive and Robust Channel Hopping (ARCH) protocol, a lightweight receiver-oriented protocol which handles the dynamics of residential environments by reactively channel hopping when channel conditions have degraded. We evaluate our approach through a series of simulations based on real data traces as well as a testbed deployment in real-world apartments. Our results demonstrate that ARCH can reduce the number of packet retransmissions by a median of 42.3% compared to using a single, fixed wireless channel, and can enable up to a 2.2 X improvement in delivery rate on the most unreliable links in our experiment. Due to ARCH's lightweight reactive design, this improvement in reliability is achieved with an average of 6 or fewer channel hops per link per day.

2009-75

Robust Sensor Networks in Homes via Reactive Channel Hopping

Authors: Mo Sha, Greg Hackmann, Chenyang Lu

Corresponding Author: {sham, gwh2, lu}@cse.wustl.edu

Abstract: Home area networks (HANs) consisting of wireless sensors have emerged as the enabling technology for important applications such as smart energy and assisted living. A key challenge faced in deploying robust wireless sensor networks (WSNs) for home automation applications is the need to provide long-term, reliable operation in the face of the varied sources of interference found in typical residential settings. To better understand the channel dynamics in these environments, we performed an in-depth empirical study of the performance of HANs in ten real-life apartments. Our empirical study leads to several key insights into designing robust HANs for residential environments. For example, we discover that there is not always a persistently good channel over 24 hours in many apartments; that reliability is strongly correlated across adjacent channels; and that interference does not exhibit cyclic behavior at daily or weekly timescales. Nevertheless, reliability can be maintained through a small number of channel hops. Based on these insights, we propose Adaptive and Robust Channel Hopping (ARCH) protocol, a lightweight receiver-oriented protocol which handles the dynamics of residential environments by reactively channel hopping when channel conditions have degraded. We evaluate our approach through a series of simulations based on real data traces as well as a testbed deployment in real-world apartments. Our results demonstrate that ARCH can reduce the number of packet retransmissions by

Type of Report: Other

Robust Sensor Networks in Homes via Reactive Channel Hopping

Mo Sha, Greg Hackmann, Chenyang Lu
Department of Computer Science and Engineering
Washington University in St. Louis, USA
{sham, gwh2, lu}@cse.wustl.edu

Abstract

Home area networks (HANs) consisting of wireless sensors have emerged as the enabling technology for important applications such as smart energy and assisted living. A key challenge faced in deploying robust wireless sensor networks (WSNs) for home automation applications is the need to provide long-term, reliable operation in the face of the varied sources of interference found in typical residential settings. To better understand the channel dynamics in these environments, we performed an in-depth empirical study of the performance of HANs in ten real-life apartments. Our empirical study leads to several key insights into designing robust HANs for residential environments. For example, we discover that there is not always a persistently good channel over 24 hours in many apartments; that reliability is strongly correlated across adjacent channels; and that interference does not exhibit cyclic behavior at daily or weekly timescales. Nevertheless, reliability can be maintained through a small number of channel hops. Based on these insights, we propose Adaptive and Robust Channel Hopping (ARCH) protocol, a lightweight receiver-oriented protocol which handles the dynamics of residential environments by reactively channel hopping when channel conditions have degraded. We evaluate our approach through a series of simulations based on real data traces as well as a testbed deployment in real-world apartments. Our results demonstrate that ARCH can reduce the number of packet retransmissions by a median of 42.3% compared to using a single, fixed wireless channel, and can enable up to a 2.2× improvement in delivery rate on the most unreliable links in our experiment. Due to ARCH's lightweight reactive design, this improvement in reliability is achieved with an average of 6 or fewer channel hops per link per day.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Measurement, Performance

Keywords

Channel Hopping, Wireless Sensor Networks

1 Introduction

In recent years, there has been growing interest in providing fine-grained metering and control of home appliances in residential settings. These *home automation* technologies enable important new applications, such as smart energy usage and assisted living. Wireless sensor networks offer a promising platform for home automation applications because they do not require a fixed wired infrastructure. Hence, home area networks (HANs) based on WSN technology can be used to easily and inexpensively retrofit existing apartments and households without the need to run dedicated cabling for communication and power.

The lack of fixed infrastructure also poses key challenges which do not exist under traditional systems of wired sensors and actuators. Home automation applications require a degree of reliability which can easily be met by wired communication, but are non-trivial when dealing with unreliable wireless communication channels. Likewise, the lack of fixed power wiring means that HAN devices must often operate from fixed battery power supplies. Energy-inefficient solutions would require the homeowner to frequently replace batteries, which could discourage their adoption.

Residential settings present a particularly challenging environment for low-power wireless networks due to the many and varied sources of interference. WSNs based on the IEEE 802.15.4 standard operate in the unlicensed 2.4 GHz band, which is increasingly becoming crowded with 802.11b/g routers, Bluetooth devices, 2.4 GHz cordless phones, and many other household devices. WSNs are particularly susceptible to interference from these devices due to their low transmission powers. Therefore, identifying (and mitigating) the causes of unreliable wireless communication in the crowded 2.4 GHz band is a fundamental and pressing issue for the deployment of reliable HANs.

This paper makes two major contributions. First, we perform an in-depth, real-world study of wireless channel properties in ten apartment buildings. Although numerous studies have explored wireless link properties before, to our knowledge ours is the first to study the behavior of HANs in real-world residential settings. Such networks offer unique challenges due the many sources of wireless interference and the unpredictable environmental dynamics. Home automation applications also often provide a different set of design goals from many WSN applications: energy efficiency, reliability, and ease-of-deployment are highly important, whereas throughput requirements are often low. Moreover, our study is geared toward enabling robust HANs through channel hopping. Accordingly, we look at the behavior of multiple wireless channels in order to understand correlations in performance across all 802.15.4 channels in residential settings. From our study, we identify several key insights into deploying reliable HANs:

1. The "best" channel for a network deployment can vary not just from apartment to apartment but from link to link.
2. In a typical apartment environment, there is usually no single channel which is persistently reliable for 24 hours at a time.
3. Even the "best" channels suffer from bursty packet loss which cannot be overcome with ARQ alone.
4. Switching channels a few times at runtime can effectively maintain reliable communication.
5. Channel conditions are not cyclic, so channel-hopping decisions must be made dynamically.
6. Reliability is strongly correlated across adjacent channels; channel-hopping should move as far away as possible from a failing channel.

From these findings, we conclude that channel diversity is a critical tool to achieving reliable HAN deployments.

Second, we propose an Adaptive and Robust Channel Hopping (ARCH) protocol based on the insights derived from our empirical study. ARCH distinguishes itself from existing channel diversity schemes, namely WirelessHART's TSMP [1] and Bluetooth's Adaptive Frequency Hopping (AFH) [2], by *reactively* switching channels according to dynamic link conditions. ARCH's reactive scheme enables existing 802.15.4 radio chips to effectively achieve consistent reliability with minimal overhead. An empirical evaluation demonstrates that ARCH can reduce the average ETX by up to 97.5% with no more than 22 channel hops per link per day.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 discusses the findings of our empirical study into the properties of wireless channels in residential environments. Based on the insights obtained in this study, Section 4 introduces the ARCH channel-hopping protocol. Section 5 presents a series of simulator- and testbed-based experiments which illustrate ARCH's efficiency in alleviating packet loss due to poor channel conditions. Finally, we conclude in Section 6.

2 Related Work

Several recent studies have analyzed the impact of interference on wireless networks through controlled experiments. [12] discusses a study on the impact of ZigBee and other interferers' impact on 802.11 links, proposing to alleviate interference with rapid channel-hopping in conjunction with 802.11b's existing support for Direct-Sequence Spread Spectrum (DSSS). [28] examines the packet delivery behavior of two 802.15.4-based mote platforms, including the impact of interference from 802.11 and Bluetooth. In contrast to these controlled studies, our own study examines the performance of HANs under normal residential activity. Our study reveals that the sources of interference in real-world residential environments are complex, varied, and unpredictable, revealing important new insights for the development of robust home automation applications.

More closely related to our study are [8, 14, 32], which perform empirical analyses of interference based on real-world measurements. [14] discusses a multi-channel measurement of Body Area Networks (BANs) and proposes a noise floor-triggered channel hopping scheme to detect and mitigate the effects of interference. [8] presents a study of UHF white space networking, while [32] presents a large-scale spectrum measurement study followed by a 2-dimensional frequent pattern mining algorithm for channel prediction. The three studies above feature different setups from our own study due to distinct focuses, goals and target applications. These studies are targeted at mobile WSNs, white space networking, and the GSM band, respectively, while our own study focuses on the impact of interference on static, indoor WSNs designed for home automation. Accordingly, our study provides new insights into the impact of interference on HANs, including burstiness, non-cyclic behavior, and the required frequency of channel hopping.

Recently, there is increasing interest in co-existence studies between different platforms [3, 15, 17, 25, 27]. [23, 26, 33] present theoretical analysis based on simulation study. In contrast to these studies, our focus is on the impact of interference and environmental dynamics in *real-life* apartments instead of controlled testbeds or simulations.

[16] discusses a real-world deployment of a WSN for high-fidelity energy metering in an office environment. In contrast to our own work, the study in [16] focuses primarily on hardware design and analyzing energy traces rather than network-related issues. Notably, the authors' specific application allows them to exploit the existing power infrastructure, so that energy efficiency is not a major concern.

Our empirical study illustrates the importance of alleviating interference through channel diversity. Existing approaches for introducing channel diversity include WirelessHART's TSMP [1] and Bluetooth's AFH [2]. While both TSMP and our approach are based on the 802.15.4 standard, ARCH employs a simpler reactive channel-hopping mechanism in contrast to TSMP's automatic pseudorandom channel-hopping scheme. Because WirelessHART is targeted to industrial applications with stringent reliability requirements (e.g., safety-critical monitoring and control systems), it uses sophisticated TDMA techniques and a complex centralized network controller to ensure channel reliability even in harsh environments. ARCH's relative simplicity makes it a more cost-effective and easier-to-deploy solution for home automation applications, where reliability requirements are less stringent. Bluetooth, particularly the emerging low-power Bluetooth standard, represents another potential approach to HANs; like TSMP, Bluetooth's AFH avoids persistent interference by constantly hopping pseudo-randomly across channels. ARCH serves as an alternative approach based on the 802.15.4 standard, where radio chips are typically not designed to accommodate AFH's aggressive channel-hopping schedules.

Other schemes have been proposed to use channel hopping as a means to enhance MAC layer performance. SSCH [9] aims to improve network capacity by using channel hopping to prevent interference among simultaneous transmissions. [21] proposes a rapid channel hopping scheme to protect from jamming attacks in the 802.11a band. Other multi-channel protocols [18–20, 30, 31, 34] have been proposed for WSNs with their limited resources in mind. Our work is distinguished from these protocols in two key ways. First, these protocols focus on enhancing throughput, while our own work aims for enhanced reliability. Second, these works deal primarily with in-network interference, while our study focuses on external sources of interference. Real-life HANs typically feature applications with low data rate requirements, but are subject to strong external interference and environmental impacts. Thus, our study considers the *integrated* behavior of interference and proposes a general-purpose solution for alleviating its effects.

3 Empirical Study

In this section, we present an empirical study which considers the reliability of HANs in typical residential environments. Specifically, this study addresses the following questions. (1) Can a HAN find a single persistently reliable channel for wireless communication? (2) If a good channel cannot be found, are packet retransmissions sufficient to deal with packet loss? (3) If the network must change channels for reliable operation, how often must this occur? (4) Do channel conditions exhibit cyclic behavior over time? (5) Is reliability strongly correlated among different channels? The results of our empirical study provide several key findings:

- The "best" channel can vary not just from apartment to apartment but from link to link.
- In a typical apartment environment, there is usually no single channel which is persistently reliable for 24 hours at a time.
- Even the "best" channels suffer from bursty packet loss which cannot be overcome with ARQ alone.
- Switching channels a few times at runtime can effectively maintain reliable communication.
- Channel conditions are not cyclic, so channel-hopping decisions must be made dynamically.
- Reliability is strongly-correlated across adjacent channels; channel-hopping should move as far away as possible from a failing channel.

These findings motivate the design of a low-overhead, reactive channel-hopping protocol for HANs. We will discuss our own such protocol in more detail in Section 4.

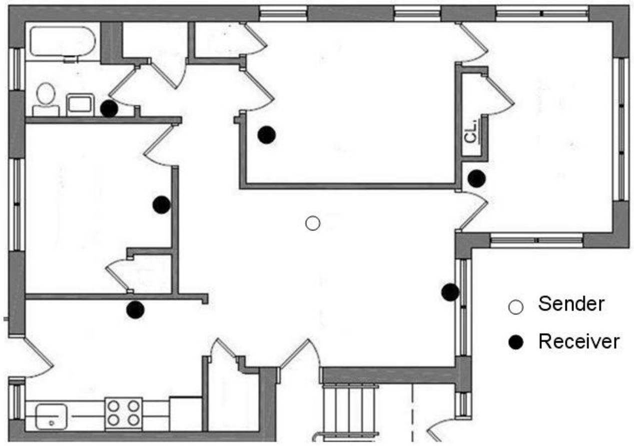


Figure 1. Floor plan of an apartment used in the study.

3.1 Experimental Methodology

To investigate the reliability of HANs, we carried out a series of experiments in ten real-world apartments constructed by different housing companies. Figure 1 shows an example floor plan of one of the apartments used in the study. Each experiment was carried out continuously for 24 hours with the residents' normal daily activities.

Our experiments were carried out using networks of Tmote Sky and TelosB [22] motes. Each mote is equipped with an IEEE 802.15.4 compliant Chipcon CC2420 radio [4]. IEEE 802.15.4 radios like the CC2420 can be programmed to operate on 16 channels (numbered 11 to 26) in 5 MHz steps. We leverage the CC2420's Received Signal Strength (RSS) indicator in our experiments to measure the signal power of environmental noise. Our experiments are written on top of the TinyOS 2.1 operating system [5] using the CC2420 driver's default CSMA/CA MAC layer.

To measure the Packet Reception Rate (PRR) of all channels at a fine granularity, we deployed a single transmitter node in each apartment which broadcasts 100 packets per channel to multiple recipient nodes, cycling over each of the 16 channels over 5 minutes. The recipient nodes record the PRR over each batch of packets into their onboard flash memory. The use of a single sender and multiple recipients allowed us to test multiple links simultaneously while avoiding interference between senders. (Inter-link interference is not a major concern in many HANs due to the low data rates that are typically employed; for example, 1 temperature reading every 5 minutes is sufficient for an HVAC system to control ambient temperature.)

3.2 Is There a Persistently Good Channel?

The first question we wished to address was whether it was possible to avoid the need for channel hopping altogether by simply locating a single consistently good channel. We considered several approaches to a fixed-channel network, ranging in complexity from using a single channel across all deployments to selecting the best channel on a per-link basis. However, we discovered that — even with per-link channel settings and perfect knowledge of wireless conditions — the complex interference patterns and dynamic

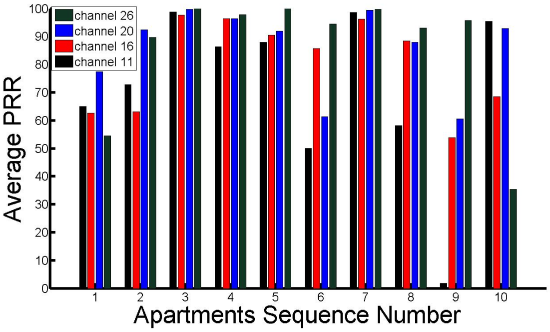


Figure 2. Average PRR in ten apartments.

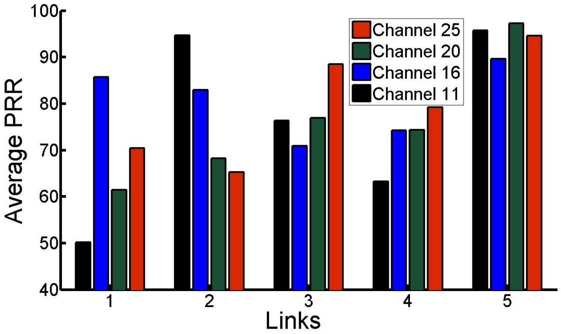


Figure 3. Average PRR of 5 links in a same apartment.

environments in typical real-life apartments often mean that *no* single channel is persistently reliable over the span of 24 hours.

We first analyzed our experimental data to determine if any one channel could have been used across all links in all apartments. Figure 2 presents the average PRR of 4 channels in ten apartments. (The remaining 12 channels are omitted for reasons of clarity.) From this figure, we see that a single channel may exhibit significant differences in reliability in different apartments. For example, channel 11 achieves a PRR $>$ 95% in apartments 3, 7, and 10; however, the same channel has a PRR $<$ 65% in apartments 1, 6, 8, and 9. Likewise, channel 26 achieves a PRR $>$ 95% in apartments 3, 4, 5, 7, and 9, but below 55% in apartments 1 and 10. The lack of a common good channel can be attributed to the residents' different daily activities and different devices that may interfere with 802.15.4 channels.

Looking at the entire dataset, Figure 4 plots a histogram of which 802.15.4 channel provided the highest PRR throughout the experiment for each of the 34 links in our study. While a large proportion of the links achieved the highest PRR using channels 25 or 26, the highest-quality channels for each link are distributed across 12 of the 16 channels in the spectrum. Indeed, numerous links had the best performance at the *opposite* end of the spectrum, in channels 11–15.

Insight 1*: There is no common good channel across different apartments.*

Moreover, further analysis showed that there is not likely to be a single good channel across multiple links in the *same* apartment. Figure 3 plots the average PRR from 5 links in an apartment. We observe that the average PRR on a given channel varies greatly across links. For example, links 1, 3, and 4 achieve to up 45% higher PRR than links 2 and 5 on channel 11.

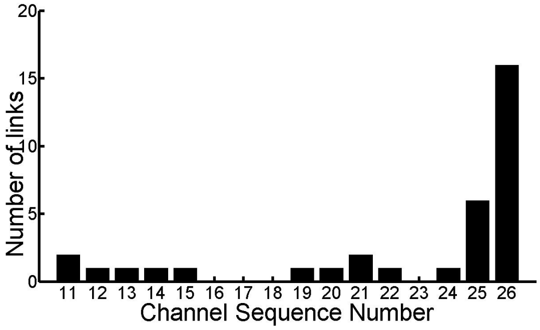


Figure 4. Histogram of the number of links for which each channel has the highest average PRR over 24 hours.

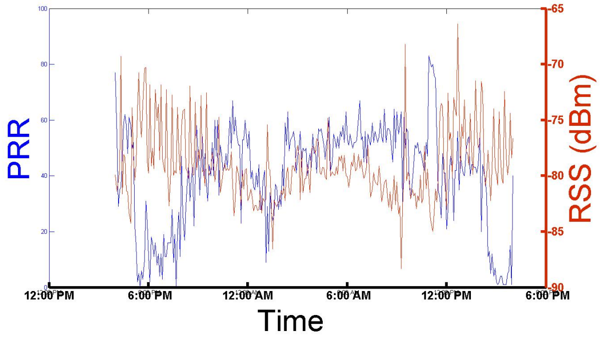


Figure 6. Changes in the PRR and RSS of a link over 24 hours.

Insight 2*: There is not always a common good channel even within a single apartment; channel selection must be done on a per-link basis.*

Finally, we observed that even for a single link, interference can cause channel quality to fluctuate greatly over time. Figure 5 illustrates the PRR over the entire 24-hour experiment on all 16 channels for a single link. We observe that PRR fluctuates greatly over the day and that none of the channels can maintain consistently high reliability for the entire day. For example, channel 20 achieves a PRR $>$ 90% at the beginning of the experiment but drops sharply to 0% in two hours. This disconnection lasts six hours, before the PRR increases to 50% at midnight and finally returns to 95% at 8:00 AM.

For this particular experiment, we deployed additional nodes close to the receivers which recorded RSS readings (i.e., environmental noise) collected every 10 ms. Figure 6 plots the average PRR and RSS across the 24-hour experiment for a single link. Although RSS alone has been shown to not always be a direct indicator of link quality [13], we can observe a general trend of PRR decreasing as environ-

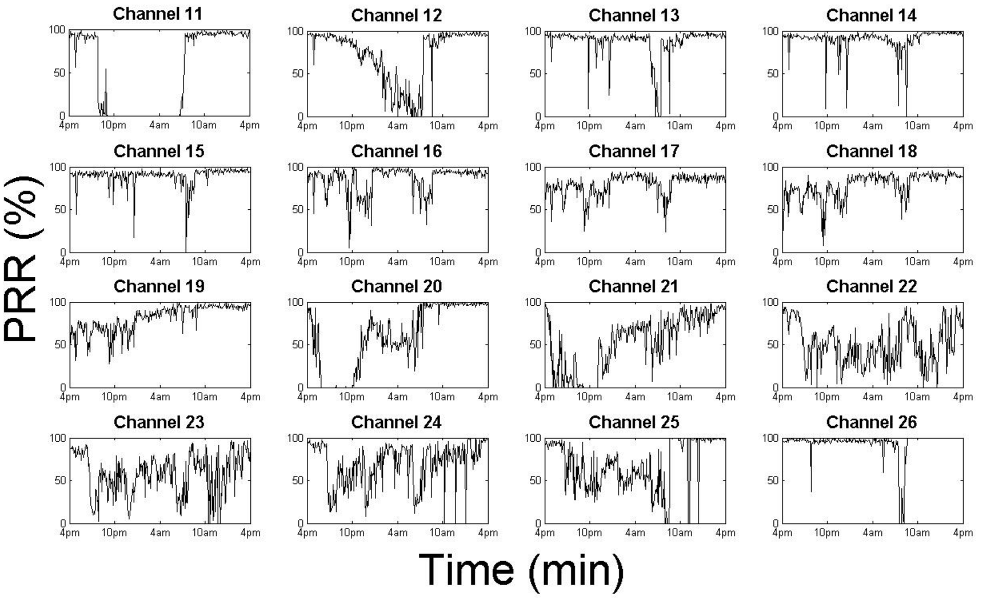


Figure 5. PRR changes over all channels for a single link during 24 hours.

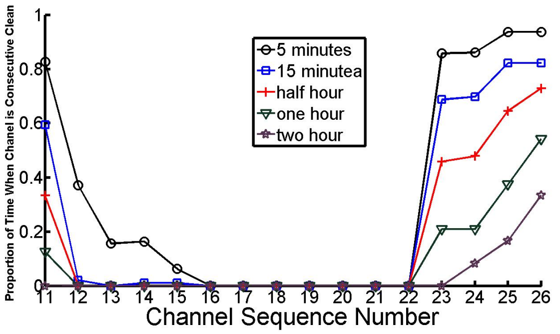


Figure 7. The proportion of time that a channel is "clean", i.e., no more than 30 dB above the noise floor.

mental noise increases. This relationship indicates that the variation in PRR comes mainly from ambient interference. Other known phenomenon which contribute to variations in PRR include the physical movement of residents [13] and variations in temperature.

To further explore this phenomenon, we used a Wi-Spy [6] spectrum analyzer to collect ambient RSS traces over 24 hours across all 802.15.4 channels in an apartment. We processed the data to find the proportion of time that a channel was persistently "clean" for windows ranging from 5 minutes to 2 hours. For the purposes of this analysis, we defined a channel to be "clean" if the RSS was 30 dB or less above the noise floor, since higher amounts of interference are known to disrupt mote transmissions [10, 24]. As shown in Figure 7, many channels (16 – 22) are not persistently clean even on the time-scale of 5 minutes. Even those which are clean a high proportion of the time in the short-term (11, 23, 24, 25, and 26) are not consistently clean on longer-term scales; no channel is persistently clean over two-hour windows for more than 40% of the experiment. This analysis indicates that every 802.15.4 channel may experience considerable interference in residential environments. IEEE 802.15.4 compliant radios share the unlicensed 2.4 GHz spectrum with IEEE 802.11 networks, Bluetooth devices, cordless phones, and many other sources of interference. Moreover, other home appliances such as microwave ovens may also interfere with the 2.4 GHz band. The diversity and dynamics of interference sources result in varying and fluctuating channel conditions across different links and different apartments.

Looking at the entire dataset across all apartments, we found that few links were able to achieve a consistently high PRR, even on their most reliable channels. Figure 8 plots the lowest PRR observed on each link's best channel: i.e., for the channel which achieves the highest average PRR over 24 hours, we plot the worst PRR out of all the 100-packet batches. Using various PRR thresholds to designate "good" channel quality, Figure 9 plots the same data as a proportion of links which have some consistently "good" channel. Even with a conservative PRR threshold of 70%, we observe that fewer than half of the links in our dataset have any consistently good channel.

***Insight 3**: Even when selecting channels on a per-link basis, there is not always a single channel with consistently high reliability.*

3.3 Is Automatic Repeat-reQuest Sufficient?

Automatic Repeat-reQuest (ARQ) is a widely used mechanism for achieving reliable data transmission over unreliable links by retransmitting a packet until the recipient acknowledges it (or a predefined number of retransmissions is exceeded). Because ARQ is effective in alleviating short temporal link failures, we next wished to analyze whether it would be effective in alleviating the link failures observed in

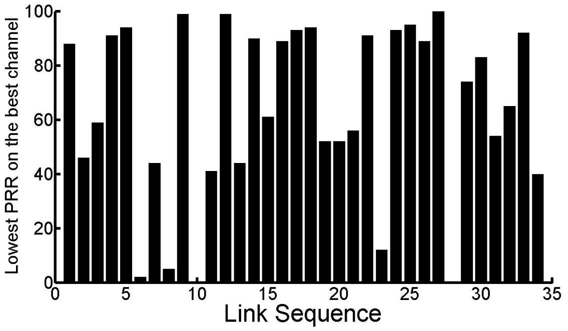


Figure 8. The lowest PRR observed on each link's highest-performing channel.

Figure 9. The proportion of links with any channel persistently above a specified PRR threshold.

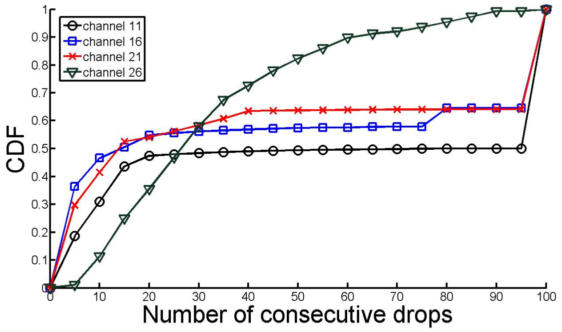


Figure 10. CDF of number of consecutive drops.

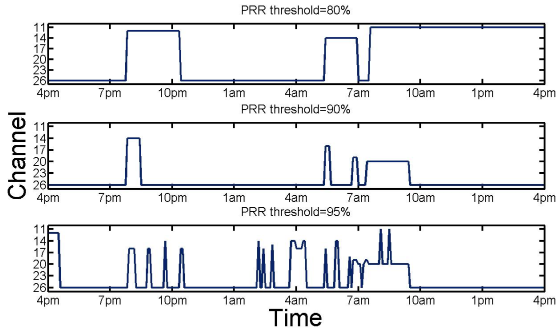


Figure 11. Optimal channel hopping schedule.

our experimental traces. However, we found that ARQ alone is insufficient in residential environments, due to the bursty nature of the packet losses.

Figure 10 illustrates this problem with the cumulative probability density (CDF) of consecutive packet drops on four channels. (Again, the other channels are omitted for clarity and space.) Even on the best channel (channel 26), up to 85 consecutive packet drops were observed, and 10% of link failures lasted for more than 60 consecutive packets. On the remaining three channels, bursts of more than 95 consecutive packet drops were observed.

Moreover, we observed that many individual links suffered long-lived disconnections on a particular channel. For example, in Figure 5, we note that channel 11 has a high PRR ($> 90\%$) for four hours, followed by 10 hours of almost zero connectivity before finally recovering at 7:00 AM. Under ARQ, the node would have had to retransmit almost continuously during this outage, which could have significantly drained the mote's limited energy supply. In contrast, had the link used a different channel during this long-lived outage, up to 80% of the packet loss would have been avoided.

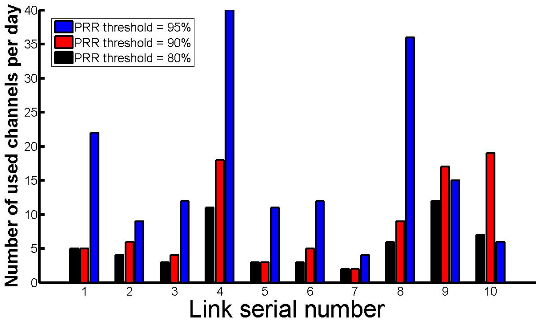
Insight 4*: ARQ alone is insufficient for HANs due to the burstiness of packet losses.*

3.4 How Frequently Should Links Switch Channels?

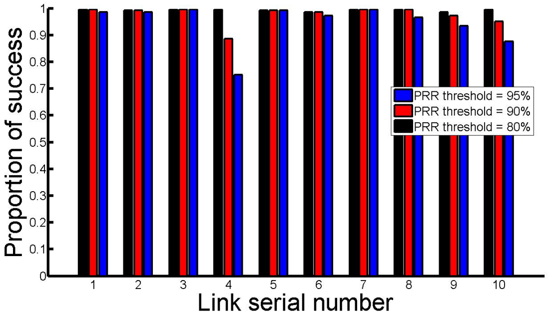
Our analysis above indicates that switching channels is often necessary to maintain long-term reliability. However, switching channels with 802.15.4 radios can incur non-negligible overhead, due to the need to coordinate senders and receivers. Hence, we wish to explore the number of channel switches necessary to maintain reliability over 24 hours.

We retrospectively processed our link quality data with an algorithm which finds the optimal channel schedule for each link. In this algorithm, we set a binary PRR threshold (e.g., PRR $> 80\%$ is considered "good"). For each link, we find the sequence of channels that exceeds this threshold as often as possible in our experimental data, while switching channels as few times as possible. (We note that this algorithm is for analysis only and cannot be implemented online, since it requires the whole data trace.)

Figure 11 plots the optimal channel hopping schedule for the link shown in Figure 5 under three different PRR thresh-

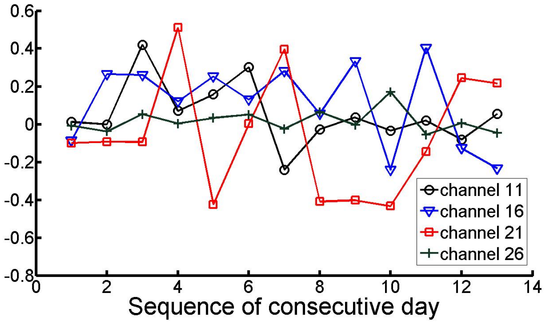


(a) Number of channel hops required under an optimal schedule; one link randomly selected per apartment.

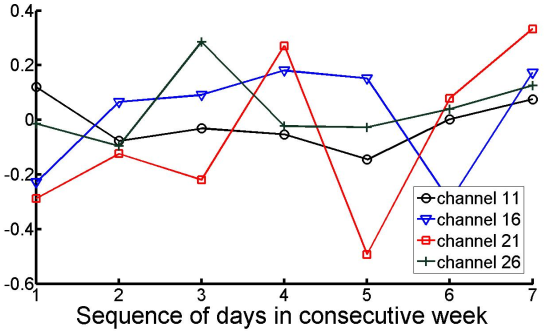


(b) The proportion of time the PRR threshold was met.

Figure 12. Optimal channel switching schedule in different apartments.



(a) PMCC of PRRs during the same time in consecutive days.



(b) PMCC of PRRs during the same time in consecutive weeks. (e.g., $x = 1$ means consecutive Mondays)

Figure 13. Pearson's product-moment coefficient.

olds. We see that relatively few channel hops are needed to maintain the target link quality: 5 hops for a PRR threshold of 80%, 8 hops for a threshold of 90%, and 36 hops for a threshold of 95%. Figure 12(a) shows similar performance across apartments, with 5 – 10 switches per day being enough to maintain a PRR threshold of 80% and at most 20 switches per day to meet at 90% threshold. We note that there are periods where none of the 16 channels meet the PRR threshold, and hence no channel hopping occurs during these times. Nevertheless, as shown in Figure 12(b), channel-hopping had a 99% success rate with a PRR threshold of 80%, and a success rate higher than 95% in most cases with a threshold of 90%. However, many more channel hops are needed to meet a threshold of 95%, with success rates as low as 75% in Apartment 4.

Insight 5*: Channel hopping is effective in alleviating packet loss due to channel degradation.*

Insight 6*: Only a small number of channel hops per day are needed to effectively maintain reliable communication.*

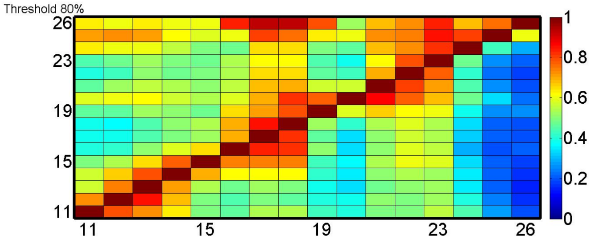
3.5 Can Hopping be Scheduled Statically?

Because channel quality varies over time, a natural question to ask is whether it exhibits cyclic properties. If so, then channel-hopping could be implemented in a lightweight fashion by generating a static channel schedule for each environment. However, our study found no obvious cyclic, predictable schedule of interference patterns. Figure 13 present the Pearson's product-Moment Coefficient (PMCC) [29] (the most common measure of dependence between two quantities) of an extended experiment which was repeated in an apartment over 14 days. Figure 13(a) shows the coefficient between PRRs during the same time in consecutive days, while Figure 13(b) compares the PRR during same time of consecutive weeks (e.g. 4pm on Monday and 4pm on next Monday). The coefficient is almost always smaller than 0.4, indicating that there is no obvious correlation between consecutive days or consecutive weeks. Therefore, channel-hopping decisions must be made *dynamically* based on channel conditions observed at runtime.

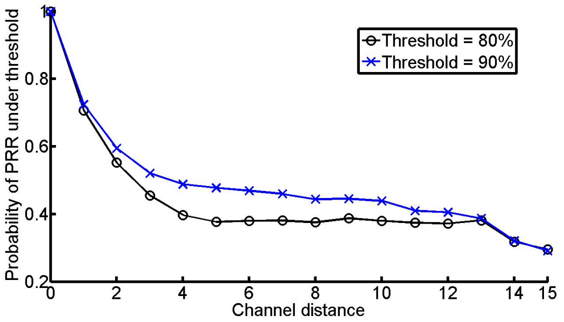
3.6 How Should New Channels be Selected?

Since channel-hopping must be performed dynamically, it is important to pick a good strategy for selecting new channels when the current channel has degraded beyond use. For the purposes of this analysis, we studied the effect of *channel distance* (the absolute difference between channel indices) on the *conditional probability* of channel failure (the probability that channel x is below a PRR threshold when channel y is below the same threshold).

We observed in our study that not all channels are equally good candidates: from Figure 14(a), we can see that performance is strongly correlated across adjacent channels. For instance, when channel 15 has poor PRR ($<$ 80%), there is a probability greater than 80% that channels 14, 16, 17, and 18 also suffer from poor PRR. In Figure 14(b), we plot the conditional probability of failure as a function of channel distance. We observe that this probability can be as high as 70%



(a) Correlation of channel performance (PRR threshold = 80%). The Y axis represents the chosen channel, and the X axis represents the probability of the second channel also being under the threshold.



(b) Overall conditional probability of channel condition.

Figure 14. Inter-channel correlation

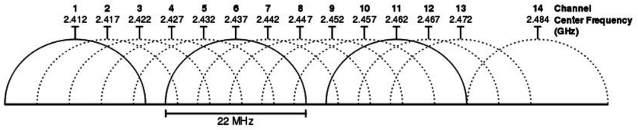


Figure 15. 802.11b/g channels in the 2.4GHz ISM band.

between neighboring channels and 60% between every other channel, but drops off as channel distance increases. From these results, we conclude that it is more beneficial to switch to a further-away channel when the current channel degrades beyond use.

The reason behind this phenomenon is that adjacent channels are likely to experience interference from the same device. For instance, the IEEE 802.11b/g standard [7] specifies 11 channels, each 22 MHz wide, with the channels' center frequencies placed 5 MHz apart (as shown in Figure 15) such that each channels overlap with the four channels supported by 802.15.4 (5 MHz channel width). Although Wi-Fi devices nominally occupy 22 MHz of spectrum, energy leakage in many off-the-shelf Wi-Fi devices can significantly interfere with 802.15.4 devices located further away in the spectrum.

4 Protocol Design

In this section, we present the design of our Adaptive and Robust Channel Hopping (ARCH) protocol. ARCH is designed based on the key observations in our empirical study and has the following salient features. First, ARCH is an *adaptive* protocol that channel-hops based on changes in channel quality (specifically, the Estimated Transmission Count, or ETX) observed in real time. We use ETX rather than RSSI/LQI to indicate link quality because RSSI/LQI are not sufficiently robust in complex indoor environments [13]. Second, ARCH is designed to be *robust and lightweight*. ARCH uses an efficient sliding-window scheme that does not involve expensive calculations or modeling and can be reasonably implemented on memory-constrained wireless sensor platforms. Third, ARCH introduces *minimal communication overhead* for applications where packet acknowledgements are already enabled.

We will begin by discussing the ARCH algorithm in outline. We will then describe several important subcomponents of ARCH — channel condition estimation, opportunistic channel selection, and coordination across nodes — in more detail. Finally, we will discuss mechanisms in ARCH for detecting and handling channel desynchronization errors.

4.1 ARCH Protocol Outline

ARCH is a receiver-oriented protocol; i.e., receivers select the communication channel for all incoming links, and senders switch to the recipient's channel when they wish to transmit a packet. Each link is initially set to use some predefined $Default\ Channel$ out of a provided $Channel\ Pool$. This pool specifies the channels which the application is allowed to use; this could be selected at design time to include all 16 channels or some subset (e.g., 4 orthogonal channels).

As a packet arrives, the channel's reliability (represented as ETX) is updated, as discussed in more detail in Section 4.2. When the ETX exceeds a specified $ETX\ Threshold$, the receiver node will select a new channel from the channel pool (see Section 4.3) and initiate a channel hop. The receiver then notifies all of its senders of this channel hop using the mechanism discussed later in Section 4.4.

To avoid the bursty packet loss observed in Section 3.3, ARCH blacklists bad channels so that they will not be used again for at least a short time period. ARCH ensures that enough candidate channels are available by un-blacklisting the entire channel pool when the number of candidate channels drops below a specified $Standby\ Channel\ Threshold$.

4.2 Channel Estimation

Estimating the reliability of a wireless link or channel is a challenging topic which has garnered significant interest in the research community. ETX represents link quality as the number of (re)transmissions required for a successful reception. ETX is particularly compelling for home automation applications because it can be estimated from sequence numbers embedded in existing packets. Thus, there is no need for expensive active probing.

We note that ARCH does *not* perform a moving average over multiple ETX values, as in e.g. TinyOS's four-bit link estimator [11]. Instead, ARCH maintains a sliding window of ETX values for the last m packets; a channel is predicted to be unreliable if all m ETX values exceed some threshold value. Our trace study in Section 5.2 demonstrates that this approach can predict channel reliability with sufficient accuracy using as little as 15 minutes' worth of history.

4.3 Opportunistic Channel Selection

As discussed in Section 3.6, we found a strong correlation among link quality on adjacent channels. Hence, using a fixed channel hopping sequence is therefore neither safe nor robust: we wish to avoid channels which are spatially close to the current, poor-quality channel. Likewise, we do not wish to continuously monitor all channels in order to support channel selection decisions; while effective, this would incur unreasonable overhead.

Instead, ARCH uses a probabilistic scheme to select new channels. When hopping channels, ARCH generates a random number $q \in [0,k]$ for each non-blacklisted channel in the *Channel Pool*, starting from the furthest-away channel to the closest. If q falls into the range $[0,c_i k]$ $(c_i < 1)$, then channel i will be selected. c_i is weighted according to the spectral distance away from the currently-used channel; the larger the distance, the more likely that a channel is selected.

4.4 Coordinated Channel Hopping

Because ARCH is a receiver-oriented protocol, nodes must notify neighbors on incoming links of any plans to switch channels. Two strategies exist to handle this situation. First, the node may notify its senders one-by-one. In the interest of minimizing overhead, this notification may be embedded in ACK packets the next time the sender transmits data to the node. Second, the node may broadcast an explicit channel-hopping message to all neighbors in range. The first approach introduces the lowest overhead, but may delay the channel hop for excessively long periods of time and cannot handle situations where the node has not yet discovered a neighbor. The second approach requires an additional control packet and may not work for asymmetric links (since broadcasts are unreliable), but allows a node to coordinate with undiscovered neighbors.

Based on these tradeoffs, ARCH implements a hybrid policy which combines the two forms of notification. Additional measures (described below) are employed to handle coordination failures.

We note that this coordination policy allows ARCH to transparently support multi-hop routing. Nodes stay on their own (receiving) channel as often as possible. When a node transmits data, it temporarily switches channels to match its recipient, then switches back after waiting long enough to receive an ACK. Thus, the node can continue to receive packets from other nodes further upstream; the only times a node leaves it own channel is when it transmits data downstream, when it could not have received data anyway.

4.5 Handling Channel Desynchronization

When channel conditions degrade, reliability may drop so far that the coordination messages described above are lost. Under this situation, a node and one or more of its senders may become desynchronized. ARCH uses two thresholds to detect these conditions: T_1 on the receiver side, which denotes the maximum waiting time between two packets; and N on the sender side, which denotes the maximum number of allowed packet retransmissions. T_1 and N are selected so that the receiver's timeout is longer than the sender's timeout, for reasons discussed below.

Based on these thresholds, ARCH uses the following procedure to detect and handle desynchronization. Let t denote the last time since the receiver received its last packet and n denote the number of times the sender has retransmitted the current packet. When either threshold is exceeded ($t > T_1$ or $n > N$), the node reverts to the default channel[1]. Because the receiver has the longer timeout, the sender will already have reverted to the default channel by the time the receiver arrives. The receiver may then initiate resynchronization with the sender.

A subtle complication is that desynchronization may be falsely detected. It is possible that the two nodes indeed switched to the same channel; however, this new channel was too noisy for communication, and hence the nodes falsely believed that they were desynchronized. Thus, ARCH has a policy that nodes exchange their previous channels when resynchronizing. If the channels do not match, then there was indeed a channel synchronization problem, and the nodes proceed to resynchronize on the receiver's previously-selected channel. However, if the channels match, then the nodes did successfully resynchronize on the new channel but were unable to communicate. In this case, the receiver selects an entirely new channel (since the previous channel was too unreliable) and repeats the channel-hopping procedure.

A salient feature of this scheme is that it provides an upper bound on disconnection time. This feature is important to home automation applications where, for example, extended disconnections in a thermal stack could cause a room to reach uncomfortable temperatures.

5 Evaluation

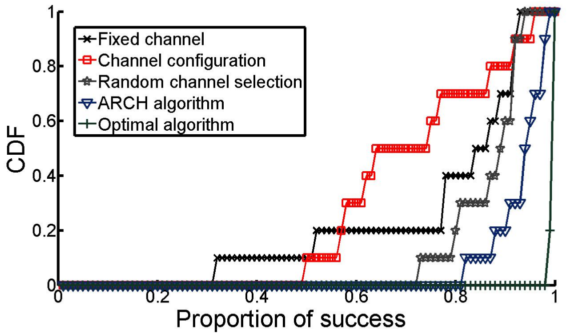
To validate the efficiency of ARCH in alleviating packet loss through channel-hopping, we performed a series of simulation- and testbed-driven experiments. We first performed a series of C++ simulations (based on the data traces obtained in Section 3) to evaluate the efficacy of our opportunistic channel selection scheme. We then performed a second set of simulations to verify that our ETX-based estimator can indeed adequately predict long-term channel quality. Finally, we measured ARCH's real-world performance by deploying implementation of ARCH on top of the TinyOS 2.1 operating system into real-life apartments.

5.1 Opportunistic Channel Selection

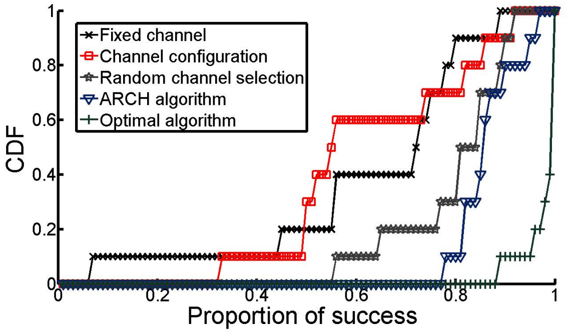
For the first group of simulations, we wished to isolate the performance of ARCH's opportunistic channel selection scheme by comparing ARCH against two widely-used channel diversity schemes. First, the *fixed channel* scheme used the default channel of 15 (which had the highest average PRR of all links in our data traces) for all links in all apartments. Second, the *channel configuration* scheme selected the channel with the best performance during the first 30 minutes of the empirical study (emulating a protocol which collects extensive link quality while bootstrapping). To further isolate the performance of ARCH's channel selection scheme from its channel estimation routines,

---

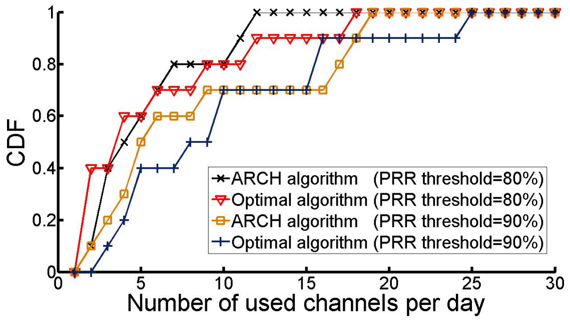
[1] If multiple default channels are specified, the node reverts to the channel spatially furthest from its last successful synchronization.



(a) CDF of proportion of meeting the PRR threshold 80% requirement.



(b) CDF of proportion of meeting the PRR threshold 90% requirement.



(c) CDF of number of used channels per day.

Figure 16. Performance comparison among ARCH, fixed-channel, channel configuration, random channel selection, and optimal algorithm under simulation.

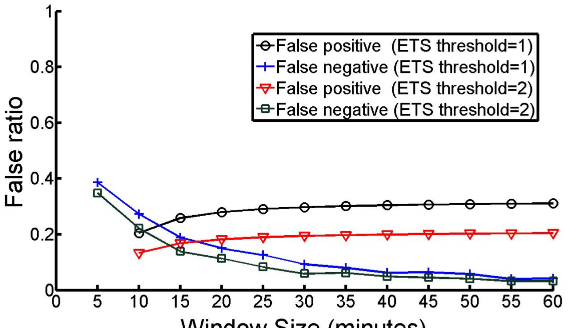


Figure 17. False-positive and false-negative rates of ETX-based estimation method.

we also performed a series of experiments using a *random channel-hopping* variant of ARCH. This variant detects channel degradation in the same way as the unmodified ARCH, but responds to degradation by hopping to random channels. Finally, we compare ARCH against the *optimal channel-hopping* algorithm used in Section 3.4 (which we remind the reader requires the whole data set and hence cannot be implemented in a real deployment).

The simulations were configured as follows. The *Channel Pool* was set to use all 16 channels, with the default channel set to 15. For the probabilistic channel selection, we set $k = 1$ and selected c_i to be the difference between the two channels' numbers divided by 100. We conducted two sets of experiments with different PRR thresholds: 80% and 90%. All simulations were carried out using the full dataset collected in Section 3, where retransmissions were disabled in order to better capture a link's reliability. To rule out the effects of the channel estimator, we replaced ARCH's ETX estimator with ground-truth PRR data over 5 minute windows.

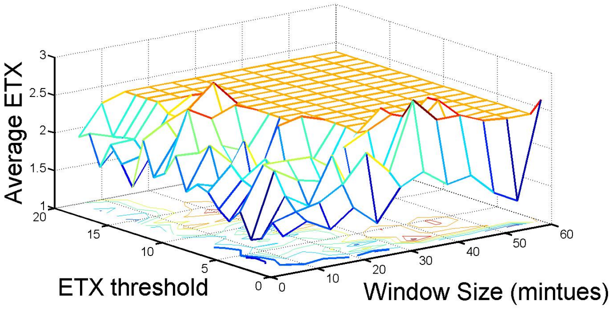
Figure 16(a) plots the CDF of the nodes' success, defined as the proportion of time that the node met the PRR threshold of 80%. On average, ARCH achieves 18% higher success than the fixed channel and channel configuration schemes. In addition, ARCH's channel selection and blacklisting schemes allow it to improve on the random channel selection scheme by 9%. Indeed, we note that ARCH comes within 6% of upper bound provided by the optimal scheme.

Increasing the PRR threshold to 90% provides similar results, as shown in Figure 16(b). Under ARCH, the links have a median success rate of 88%; in contrast, under the fixed channel and channel configuration schemes, the median success rates are 72% and 56%, respectively. Again, ARCH improves on the random channel selection scheme by 8%, coming within 12% of the optimal scheme's upper bound.

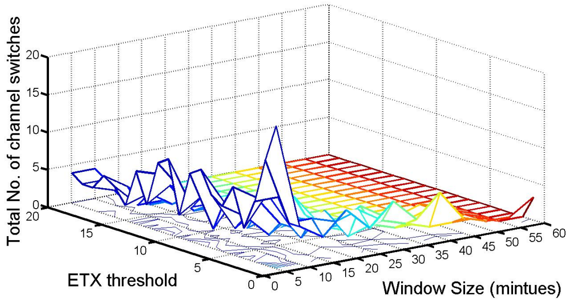
As shown in Figure 16(c), ARCH achieves this degree of reliability with relatively few channel hops. At most 25 channel switches are needed per link per day to meet the 90% PRR requirement, with a median of fewer than 10.

5.2 ETX-Based Estimator

Next, we wish to explore the ETX-based channel estimator's ability to accurately predict long-term channel condi-

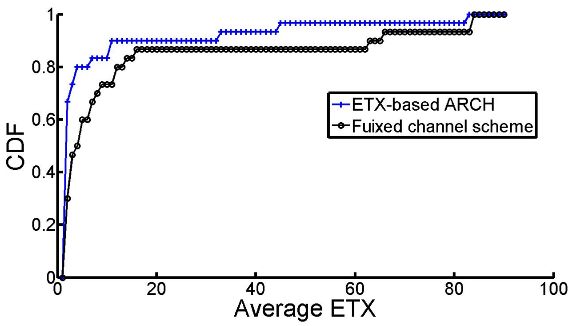


(a) Average ETX with different ETX thresholds and window sizes.

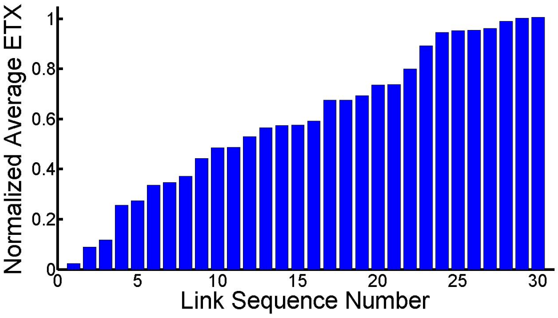


(b) Total # of channel switches with different ETX thresholds and window sizes.

Figure 18. The performance of various ETX parameters under simulation.



(a) CDF of average ETX of 30 links.



(b) Normalized average ETX (ARCH's divided by fixed-channel's ETX) of all links.

Figure 19. Comparison of ETX under ARCH and fixed-channel.

tions. For these experiments, we collected a new set of data traces with a reduced data rate of 1 packet/5 minutes and retransmissions enabled. We first compare ETX's predictions for channel reliability against ground-truth data, shown in Figure 17. This figure graphs the false positive (i.e., channel failure predicted when no failure actually occurred) and false negative (i.e., no channel failure predicted when the channel had actually failed) rates with various ETX thresholds and window sizes. We observe that an ETX threshold of 2 and window size of 15 minutes (i.e., 3 packets) achieves false positive and negative rates below 20%.

Figure 18 confirms that these parameters are ideal, even over a wider range of thresholds and window sizes. A threshold of 2 and window size of 15 minutes achieved the lowest ETX (an average of 1.66 transmissions) and total channel switches (5). For comparison, a fixed-channel scheme run over the same data trace produced an average ETX of 2.38 transmissions.

5.3 Experiments in Real-Life Apartments

To validate these simulator-driven experiments, we perform a new set of experiments in ten real-world apartments for 24 hours apiece. As with the second set of simulations, we use a data rate of 1 packet/5 minutes. We set ARCH's ETX threshold to 2 and window size to 3 packets per the previous simulator results.

Figure 19(a) plots the CDF of the average ETX for each of the experiment's 30 links. ARCH reduces the ETX by a median of 42.3% compared to a fixed-channel scheme. Figure 19(b) breaks down ARCH's improvements on a per-link basis. In many cases, the improvements are quite notable; ARCH reduced the transmissions by more than half for 11 of the 30 links, and in one extreme case reduced transmissions by 97.5%. Even in the worst case, ARCH performs comparably with the fixed-channel scheme, with a slight ETX increase of 0.7%.

Figure 20 compares the delivery rate (i.e., the proportion of packets successfully delivered after any number of retransmissions) of ARCH and the fixed-channel scheme. While ARCH does not achieve 100% delivery under all links, it does so for 26 of the 30 links. In comparison, the fixed-channel scheme achieves a 100% delivery rate for only 21 links. ARCH also achieves a much higher minimum delivery rate (54.2% vs. 17.0%) than the fixed-channel scheme. These results confirm our observation in Section 3.3 that ARQ alone is insufficient to ensure link reliability in residential environments.

Figure 21 illustrates the number of channel desynchronizations detected for each corresponding link in Figure 20. (The links are sorted in the same order as in Figure 20 so that a direct comparison may be made.) Although some of the links experience many desynchronization events, ARCH is still able to maintain a high delivery rate. For example, link 4 experienced over 100 desynchronizations during the 24-hour experimental run, but nevertheless achieved a delivery rate of close to 100%. This indicates that ARCH's desynchronization-handling mechanism, as de-

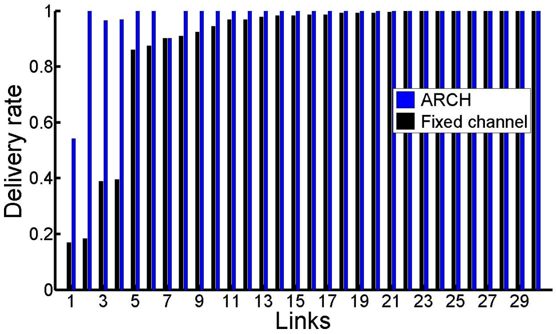


Figure 20. A comparison of each link's delivery rate.

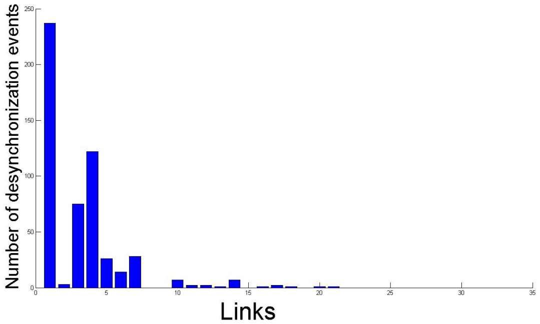


Figure 21. The number of channel desynchronization events for each link.

scribed in Section 4.5, is indeed effective at resolving these events. We note an outlier in link 1, which desynchronized more than 200 times throughout the experiment and achieved a delivery rate of only 54.2%. These statistics reflect the fact that the link was under such harsh, persistent interference that the recipient struggled to locate a single good channel. Nevertheless, as noted above, ARCH is still able to achieve a 2.2× improvement in delivery rate on this link over the fixed-channel scheme.

Figure 22 presents the overhead of ARCH in terms of channel switches. As with the simulator experiments, we observe that the number of channel switches is quite low. Half of the links in the experiment require 6 or fewer switches per day to maintain reliability; no link requires more than 22 switches per day.

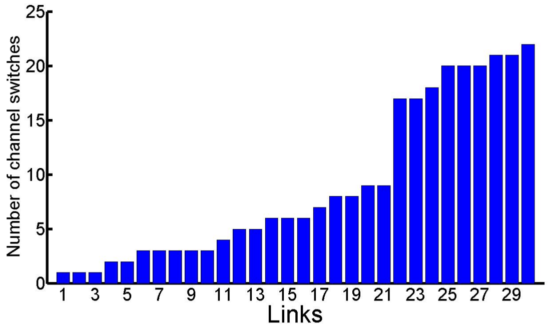


Figure 22. Number of total channel switches.

6 Conclusion

HANs based on wireless technology represent a promising platform for sophisticated home automation applications. However, the many and varied sources of interference in typical residential environments pose significant reliability and efficiency challenges. This paper first presents an empirical study on the performance of HANs in real-life apartments. Our study leads to several key insights for developing robust HANs based on WSNs. Notably, we found that there is usually no persistently reliable channel over 24 hours in typical apartments, that ARQ alone cannot compensate for poor channel conditions, and that interference in residential environments do not generally behave as cyclic phenomenon. Despite these challenges, we also found that only a small number of judicious channel hops are required to maintain link reliability; we also observed a correlation in performance across nearby channels that should be considered when selecting new channels. Based on these insights, we proposed the Adaptive and Robust Channel Hopping (ARCH) protocol, a lightweight yet effective channel hopping protocol that can handle the dynamics of channel conditions in apartments using a handful of channel hops per link per day. Trace-driven simulations and testbed-based experiments demonstrate the efficacy of ARCH's design, revealing a median decrease in packet retransmissions of 42.3% and increasing the proportion of links with perfect delivery rates by 17%. ARCH provides even greater benefit for the most challenging of links, increasing the minimum delivery rate in our experiments by a factor of 2.2×. ARCH's lightweight design enables these dramatic improvements in reliability with an average of 6 and a maximum of 22 channel hops per link per day.

7 References

[1] Technical Overview of Time Synchronized Mesh Protocol, White Paper, http://www.dustnetworks.com.

[2] Specification of the Bluetooth System, Version 4.0.

[3] ZigBee, White Paper, June 2007.

[4] 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, Texas Instruments.

[5] `http://www.tinyos.net/` .

[6] Wi-Spy, http://www.metageek.net/.

[7] IEEE 802.11, http://www.ieee802.org/11/.

[8] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with wi-fi like connectivity. In *Sigcomm*, 2009.

[9] V. Bahl, R. Chandra, and J. Dunagan. Ssch: Slotted seeded channel hopping for capacity improvement in ad hoc networks. In *MobiCom*, 2004.

[10] O. Chipara, G. Hackmann, C. Lu, W. D. Smart, and G.-C. Roman. Practical modeling and prediction of radio coverage of indoor sensor networks. In *IPSN*, 2010.

[11] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis. Four-bit wireless link estimation. In *HotNets VI*, 2007.

[12] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *Sigcomm*, 2007.

[13] G. Hackmann, O. Chipara, and C. Lu. Robust topology control for indoor wiress sensor networks. In *SenSys*, 2008.

[14] J.-H. Hauer, V. Handziski, and A. Wolisz. Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks. In *EWSN*, 2009.

[15] I. Howitt and J. A. Gutierrez. IEEE 802.15.4 low rate - wireless personal area network coexistence issues. In *WCNC*, 2003.

[16] X. Jiang, M. V. Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys*, 2009.

[17] X. Jing, S. S. Anandaraman, M. A. Ergin, I. Seskar, and D. Raychaudhuri. Distributed coordination schemes for multi-radio co-existence in dense spectrum environments: An experimental study on the orbit testbed. In *DySPAN*, 2008.

[18] Y. Kim, H. Shin, and H. Cha. Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *IPSN*, 2008.

[19] H. K. Le, D. Henriksson, and T. Abdelzaher. A control theory approach to throughput optimization in multi-channel collection s ensor networks. In *IPSN*, 2007.

[20] H. K. Le, D. Henriksson, and T. Abdelzaher. A practical multi-channel media access control protocol for wireless sensor networks. In *IPSN*, 2008.

[21] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein. Using channel hopping to increase 802.11 resilience to jamming attacks. In *Infocom Minisymposium*, 2007.

[22] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *IPSN*, 2005.

[23] S. Pollin, M. Ergen, M. Timmers, A. Dejonghe, L. van der Perre, F. Catthoor, I. Moerman, and A. Bahai. Distributed cognitive coexistence of 802.15.4 with 802.11. In *Cognitive Radio Oriented Wireless Networks and Communications*, 2006.

[24] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-MAC: Model-driven concurrent medium access control for wireless sensor networks. In *Infocom*, 2009.

[25] S. Y. Shin, H. S. Park, S. Choi, and W. H. Kwon. Packet error rate analysis of ZigBee under WLAN and Bluetooth interferences. In *IEEE trans. on wireless communications*, 2007.

[26] S. Y. Shin, H. S. Parky, S. Choi, and W. H. Kwon. Packet error rate analysis of IEEE 802.15.4 under IEEE 802.11b interference. In *WWIC*, 2005.

[27] A. Sikora and V. F. Groza. Coexistence of IEEE 802.15.4 with other systems in the 2.4 GHz ISM band. In *IMTC*, 2005.

[28] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low power wireless. In *ACM Transactions on Sensor Networks*, 2010.

[29] S. M. Stigler. Francis Galton's account of the invention of correlation. *Statistical Science*, 4(2), 1989.

[30] Y. Wu, J. A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Infocom*, 2008.

[31] G. Xing, M. Sha, J. Huang, G. Zhou, X. Wang, and S. Liu. Multi-channel interference measurement and modeling in low-power wireless networks. In *RTSS*, 2009.

[32] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: a large-scale spectrum measurement study. In *Mobicom*, 2009.

[33] D. G. Yoon, S. Y. Shin, W. H. Kwon, and H. S. Park. Packet error rate analysis of IEEE 802.11b under IEEE 802.15.4 interference. In *VTC Spring*, 2006.

[34] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher. MMSN: Multi-frequency media access control for wireless sensor networks. In *Infocom*, 2006.