Exploring Link Correlation for Performance Improvements in Wireless Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Shuai Wang

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Tian He

February, 2017

Acknowledgements

There are many people that have earned my gratitude for their contribution to my five-year PhD life in University of Minnesota.

First of all, I would like to thank my advisor Prof. Tian He, for his guidance, support, inspiration, as well as the great confidence and the tremendous patience he has shown me. I am also grateful to Prof. Abhishek Chandra, Prof. Gerald Sobelman, Prof. Kuang Rui, and Prof. David Du for serving as my dissertation committee and for their helpful suggestions and feedback.

I was fortunate to join MiNDs group. I sincerely appreciate the help and contributions of the former and current members including Dr. Yu Gu, Dr. Song Min Kim, Dr. Desheng Zhang, Shi Bai, Wenchao Jiang, Zhimeng Yin, Song Liu, Ruofeng Liu and Yi Ding. It has always been enjoyable and fruitful to work with them.

Furthermore, I would like to show my greatest thanks and love to my family – my parents, my sister, and my grandma – for their unwavering support. Without their love, care and encouragement, I would not have come this far.

Last but not the least, I gratefully acknowledge financial support from the Department of Computer Science and Engineering at the University of Minnesota, the National Science Foundation, ACM, and IEEE.

Dedication

To those who held me up over the years

Abstract

In wireless communication, many technologies, such as Wi-Fi, BlueTooth and ZigBee, operate in the same ISM band. With the exponential growth of wireless devices, the ISM band becomes more and more crowded. These wireless devices compete with each other to access spectrum resources, generating cross-technology interference (CTI). Since cross-technology interference may destroy wireless communication, the field is facing an urgent and challenging need to investigate the packet reception quality of wireless links under CTI.

In this dissertation, we propose an in-depth systematic study from empirical measurement, theoretical analysis, modeling, to design and implementation of protocols that exploit packet reception patterns of wireless links under cross-technology interference. Based on extensive measurements, we exploit link correlation phenomenon that packet receptions from a transmitter to multiple receivers are correlated. We then propose link correlation model which contradicts the widely made link independent assumption. The proposed model has a broad impact on network designs that utilize concurrent wireless links, which include (i) traditional network protocols such as broadcast, and (ii) diversity-based protocols such as network coding and opportunistic routing.

In the study of the impact of link correlation model on traditional network protocols, we present the design and implementation of CorLayer, a general supporting layer for energy efficient reliable broadcast that carefully blacklists certain poorly correlated wireless links. We integrate CorLayer transparently with sixteen state-of-the-art broadcast protocols specified in thirteen publications on three physical testbeds running TelosB, MICAz, and GreenOrbs nodes, respectively. The experimental results show that CorLayer remarkably improves energy efficiency across a wide spectrum of broadcast protocols and that the total number of packet transmissions can be reduced consistently by 47% on average.

In the study of the impact of link correlation model on diversity-based protocols, we propose link correlation aware network coding and link correlation aware opportunistic routing. In link correlation aware network coding, we introduce Correlated Coding which seeks to optimize the transmission efficiency by maximizing necessary

coding opportunities. In link correlation aware opportunistic routing, we propose a novel candidate forwarder selection algorithm to help opportunistic routing fully exploit the diversity benefit of the wireless broadcast medium. Testbed evaluation and extensive simulation show that the traditional network coding and opportunistic routing protocols' transmission efficiency is significantly improved with our link correlation model.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless technologies are widely utilized in people's daily life for personal communication, mobile internet surfing, global positioning, and smart home automation. To accommodate different application requirements on system performance (e.g., throughput, reliability, delay, and energy consumption), a wide range of wireless technologies, such as Wi-Fi, BlueTooth and ZigBee, have been proposed. Many of these wireless technologies share the same spectrum (e.g., 2.4G ISM bands). With the increasing number of wireless devices, these devices compete with each other to access spectrum resources, generating cross-technology interference. For example, in a residential building, Wi-Fi devices provide wireless internet connectivity for web surfing and video streaming, whereas ZigBee devices enable energy-efficient sensing and actuation for home automation. In close proximity, it has been shown that traffic generated by a Wi-Fi device can disrupt the communication of other Wi-Fi devices or ZigBee devices severely [1, 2, 3].

To understand how cross-technology interference impacts communication performance among wireless devices, extensive research [4, 5, 6, 7, 8, 9, 10, 11] has been done to measure the packet reception quality of individual links in realistic environments. The primary conclusion of these in-situ studies has been shown that idealistic models do not hold up well in practice. Clearly, protocols perform poorly in realistic environments if their designs are based on simplifying assumptions, such as presuming that (i) signal propagation is a fixed function of distance, (ii) RF signal strength attenuates identically in all directions, and (iii) link quality remains stable over time.

1.1 Thesis Statement

Although the foci of empirical studies on wireless network performance are highly diverse, they predominately examine statistics for an individual link. In contrast, little research has been done to investigate the reception correlation among neighboring wireless links, despite the fact that wireless communication essentially occurs in a broadcast medium with concurrent receptions.

This dissertation explores the link correlation phenomenon in which packet receptions from a transmitter to multiple receivers may be correlated. Link correlation challenges the widely made assumption that wireless transmissions are independent (i.e., the reception probability of a receiver is solely governed by that link's quality measure). The finding of link correlation brings in a more realistic radio model for wireless network design, opening up new opportunities for network performance optimization.

Link correlation has a broad impact on network protocols that utilize concurrent wireless links, which include but are not limited to (i) traditional network protocols such as broadcast [12, 13], multi-cast [14, 15], and multi-path routing [16], or (ii) diversity-based protocols such as network coding [17, 18, 19], collaborative forwarding [20] and opportunistic forwarding [21, 22, 23].

1.2 Outline and Contributions

This dissertation studies link correlation for performance improvements in broadcast, network coding, and opportunistic routing separately. The outline and the primary contributions of this dissertation are as follows:

• **Link Correlation Aware Broadcast (Chapter 3)**

This chapter shows how link correlation can significantly impact broadcast. We present the design and implementation of CorLayer, a general supporting layer for energy efficient reliable broadcast that carefully blacklists certain poorly correlated wireless links. This method uses only one-hop information, which makes it work in a fully distributed manner and introduces minimal communication overhead. The highlight of the work is CorLayer's broad applicability and effectiveness. the system effort is indeed significant. We integrate CorLayer transparently with sixteen state-of-the-art broadcast protocols specified in thirteen publications [12, 13, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]

on three physical testbeds running TelosB, MICAz, and GreenOrbs nodes, respectively. The experimental results show that CorLayer remarkably improves energy efficiency across a wide spectrum of broadcast protocols and that the total number of packet transmissions can be reduced consistently by 47% on average.

• **Link Correlation Aware Network Coding (Chapter 4)**

In this chapter, for the first time, we analyze the impact of link correlation on network coding and quantify the coding benefits. We propose Correlated Coding, which encodes packets only when necessary, to optimize the transmission efficiency. Correlated coding uses only one-hop information, which makes it work in a fully distributed manner and introduces minimal communication overhead. The highlight of the design is its broad applicability and effectiveness. We implement the design with four broadcast protocols and three unicast protocols, and evaluate them extensively on one 802.11 testbed and three 802.15.4 testbeds. The experiment results show that (i) more coding opportunities do not lead to more transmission benefits, and (ii) compared to coding aware protocols, the number of coding operations is reduced while the transmission efficiency is improved.

• **Link Correlation Aware Opportunistic Routing (Chapter 5)**

In this chapter, we propose a novel link-correlation-aware opportunistic routing scheme, which significantly improves the performance by exploiting the diverse low correlated forwarding links. We evaluate the design in a real-world setting with 24 MICAz nodes. Testbed evaluation and extensive simulation show that (i) higher link correlation leads to fewer diversity benefits, and (ii) with the link-correlation-aware design, the number of transmissions is reduced by 38%.

Chapter 2

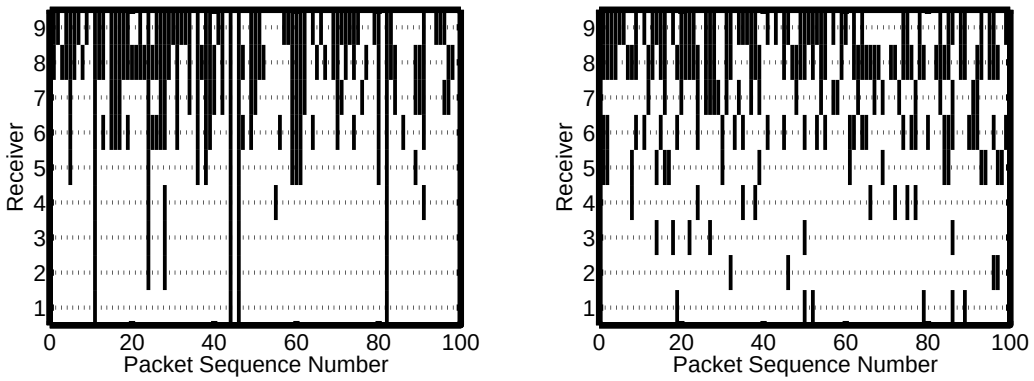
Motivation

Most existing studies in this area [4, 5, 6, 7, 8, 9, 10, 11, 34] focus on individual link quality. Little systematic study has investigated reception correlation among neighboring receivers, and ensuing research on link correlation is severely lacking. The unprecedent popularity of wireless technology has created an emerging need for the proposed research because of (i) the increasing cross-technology interference due to co-existence of heterogeneous wireless networks and (ii)the correlated shadow fading introduced by highly dynamic environments in which networks are located. In the following of this chapter, we first show the existence of link correlation from the empirical study. Then, we demonstrate the causes of link correlation.

2.1 Experimental Study

This section presents our empirical study on link correlation. This study confirms the existence of link correlation among low-power wireless devices and demonstrates the need for developing models to capture such correlation for the prediction and optimization of network performance in realistic settings.

We conducted a small experiment on an 802.15.4 testbed in UMN. In this testbed, ten MICAz nodes are deployed to form a single-hop network. A randomly selected node serves as the transmitter and broadcasts 100 messages to the others under channel 16, and the other 9 nodes act as receivers (i.e., a star topology). Each packet is identified by a sequence number. All the receivers report their reception results to a sink node.



(a) Empirical Trace (b) Synthetic Independent Trace

Figure 2.1: Packet receptions at nine receivers.

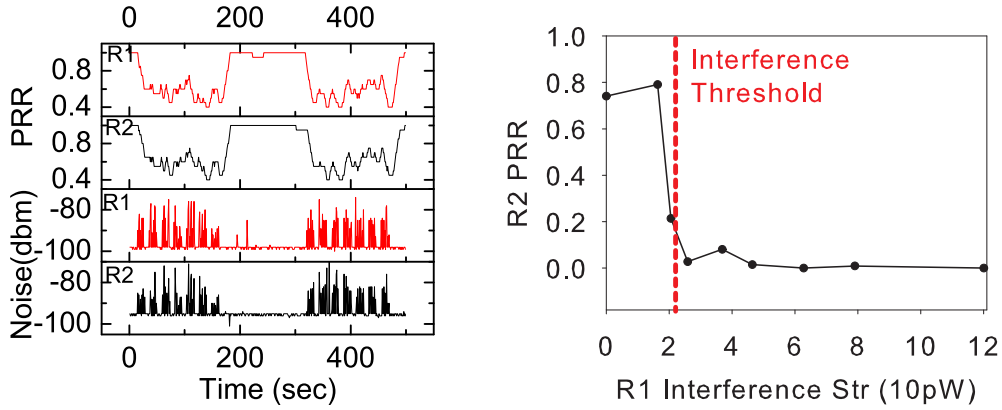
Figure 2.1(a) shows the packet receptions at different receivers from empirical measurements. Successful packet receptions are marked by white bands. Long vertical black bands indicate that packets are lost at multiple receivers, and vice versa for the white bands. As a comparison, Figure 2.1(b) shows the reception of packets in synthetically generated traces with independent wireless links. In the latter, we observe few multiple simultaneous receptions and losses. This comparison indicates that the packet receptions of different links in Figure 2.1(a) are indeed correlated.

2.2 Causes of Link Correlation

We now move further to show the underlying causes of link correlation: (i) cross-technology interference under shared medium; and (ii) correlated fading introduced by highly dynamic environments.

2.2.1 Cross-Technology Interference

With the increasing popularity of wireless network technologies, the wireless spectrum now often becomes crowded. For example, 802.11b, 802.11g, and 802.15.4 all use the

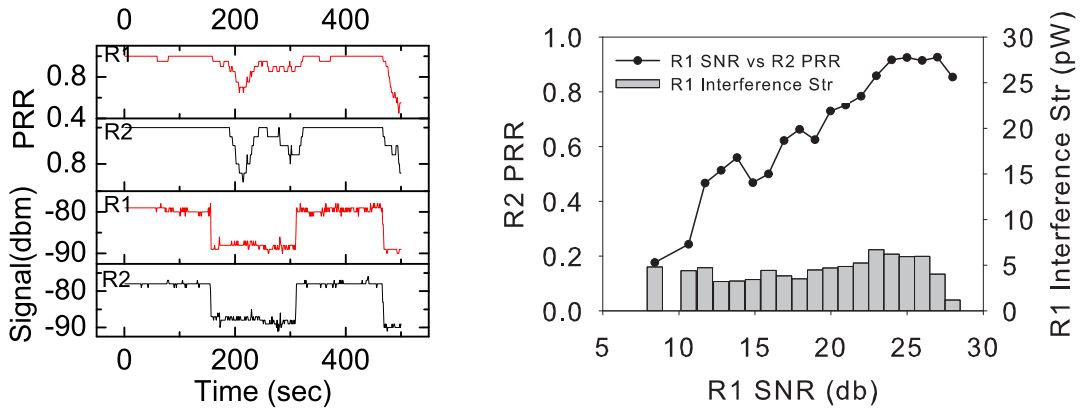


(a) PRR & Noise RSSI

(b) R_2's PRR vs. R_1's Interference

Figure 2.2: Cross-technology interference.

2.4 GHz ISM band, leading to possible cross-technology interference. Traffic from high-power wireless networks (e.g., Wi-Fi) could introduce destructive noise in other low-power networks (e.g., ZigBee), causing correlated packet loss in multiple links simultaneously. Non-network appliances such as microwave ovens can also introduce high-power interference in this unlicensed band and we succeed in reproducing the effect with a controlled experiment in which microwave is toggled on and off with a period of 2.5 minutes. This simple experiment consists of three MicaZ nodes. A sender node transmits packets at a fixed rate of one packet per second. In between transmissions, the two receivers record RSSI values, indicating noise level. Note that channel assessment (CCA) function of the sender is turned off so that transmission could take place regardless of the channel noise. The upper two figures in Figure 2.2(a) show the PRR of the two receivers, while the lower two show the noise levels. Similar peaks in the noise levels are observed in both receivers when microwave is left on. Such high-power noise corrupts packets at both receivers simultaneously, leading to highly correlated packet losses in the upper two figures. Figure 2.2(b) shows PRR of R_2 relative to interference power observed at R_1. From the figure, we see that R_2's PRR faces a steep decrease when interference at R_1 is above a certain level, which we call interference threshold (indicated by the dashed line), where almost no reception can be made on R_2 upon crossing it.



(a) PRR & Signal RSSI

(b) R_2's PRR vs. R_1's SNR

Figure 2.3: Correlated fading.

2.2.2 Correlated Shadow Fading

In reality, wireless signals suffer shadow fading caused by the presence of obstacles in the propagation path of the radio waves, leading to correlated reception performance among receivers that are closely located. Figure 2.3(a) plots two receivers' PRR and the corresponding signal RSSI with the introduction of shadow fading by object blocking with a period of 2.5 minutes in every 5 minutes. From Figure 2.3(a), we can find that both PRR and RSSI are in similar patterns at these two receivers, which means that the packet losses caused by object blocking (i.e., fading) are correlated. Figure 2.3(b) shows the case when interference is under the threshold. This is part where the fading correlation comes into play. Under threshold, there exists a near-linear relationship between R_1's SNR and R_2's PRR, as shown in Figure 2.3(b). In other words, fading correlation induces approximately linear relationship between the quality of R_1's link and reception probability of R_2. The bar graph (Figure 2.3(b)), which represents interference power, indicates that interference does not affect the relationship between R_1's SNR and R_2's PRR when it is under threshold.

Chapter 3

Link Correlation Aware Broadcast

In this chapter, we focus on how to exploit link correlation to improve the energy efficiency of reliable broadcast protocols that try to guarantee every node in the network receives packets. Reliable broadcast is a fundamental operation in wireless network design and plays a critical role in many other operations such as code dissemination (e.g., Deluge [35] and MNP [36]), content delivery (e.g., Cabernet [37]), and routing discoveries (e.g., ODRMP [38] and OLSR [39]).

In reliable broadcast designs, we essentially need to select a set of forwarding nodes that cover all the other nodes, called the dominating set. Once nodes in this set are connected, we call it a connected dominating set (CDS). For energy efficiency, the broadcast algorithms should seek a CDS with a minimal size. Based on different approaches of finding the CDS, existing reliable broadcast algorithms can be classified as (i) tree-based (e.g., ZigBee [40]); (ii) cluster-based (e.g., passive clustering [28]); (iii) multi-point relays (e.g., OLSR [39]); (iv) pruning-based (e.g., RNG relay subset [26] and partial dominating pruning [12]), (v) location-based (e.g., curved convex hall [31]). Prior efforts on reliable broadcast focus mainly on how to select the optimal set of forwarding nodes (e.g.,with the minimum size). Though the research along this line has been comprehensive, all of the designs implicitly or explicitly assume independent wireless links and do not yet take link correlation into consideration.

Instead of coming up with yet another broadcast protocol or modification to existing protocols, we attempt to improve a wide range of existing broadcast protocols by designing a general supporting layer, called *CorLayer*. Taking link correlation into account, we blacklist links that are poorly correlated with adjacent links. With the CorLayer, broadcast algorithms will naturally form clusters with higher link correlations, which means that a forwarder needs fewer transmissions to deliver a packet to all of its covered members. Specifically, the contributions are as follows:

• Although the phenomenon of link correlation has been mentioned in the literature [41, 42], we provide the first extensive study to exploit the root cause of link correlation. We reveal the impact of link correlation on broadcast efficiency and demonstrate experimentally and theoretically why link correlation matters in wireless broadcasting.

• we design a supporting layer by blacklisting certain links in the original wireless network. Sitting between the MAC and routing layers, CorLayer presents a reduced network topology that can be transparently utilized by broadcast protocols, requiring no modification of broadcast protocols.

• We evaluate the design on three real-world multi-hop testbeds: a network with 36 MICAz nodes in University of Minnesota, U.S., a network with 30 TelosB nodes in SIAT, Chinese Academy of Sciences, China, and a network with 20 GreenOrbs nodes in the Third Research Institute of Ministry of Public Security (TRIMPS), China. The results are very encouraging, as with CorLayer we are able to broadly improve the performance of sixteen state-of-the-art broadcast protocols specified in thirteen publications [12, 13, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33], ranging from 20% $\sim$ 64%.

The rest of this chapter is organized as follows. In Section 3.1, we review related work. Section 3.2 presents the motivation of the design. The main design is presented in Section 3.3. Sections 3.4 and 3.5 report testbed and simulation experiments. Finally, Section 3.6 concludes the chapter.

3.1 State of the Art

This section briefly reviews related works. As a primitive of wireless network communications, broadcasting has been extensively studied in the literature. A major problem in broadcast is that many intermediate nodes unnecessarily forward a message. Nodes

often hear the same packet multiple times. This is known as the *broadcast storm* problem [43]. The existing methods that address this problem can be divided into two categories: probabilistic and deterministic [44]. In probabilistic methods [45, 46], each node rebroadcasts the packet to its neighbors with a given forwarding probability. The deterministic approaches predetermine a set of forwarders to relay the broadcast packet. In this chapter, we design CorLayer for deterministic protocols. Generally, deterministic reliable broadcast algorithms can be classified into five types, namely tree-based, cluster-based, multi-point relays, pruning-based, or location-based.

• **Tree-based:** In [27], the authors present a fully distributed, online, and asynchronous method to maintain a spanning tree, along which the broadcast is performed. Ding et al. [40] present a tree-based reliable broadcast for IEEE 802.15.4 and ZigBee networks.

• **Cluster-based:** The authors in [25] construct a CDS using a cluster tree method. They first apply a distributed leader election algorithm to construct a rooted spanning tree. Then a maximum independent set (MIS) is calculated based on the tree level. The nodes in the MIS are then spanned by a *dominating tree*. Wu and Li [32] introduce the concept of intermediate node to calculate a CDS for reliable broadcast.

• **Multi-Point Relays:** In MPR [13], the set of relays is an optimal subset of a node's direct neighbors whose collective neighborhood entirely covers the node's two-hop neighbors. The authors in [24] propose two modified MPR algorithms, named Min-id MPR and MPR CDS, which compute multi-point relays without using the last-hop knowledge.

• **Pruning-based:** In [29], the authors propose two algorithms called self-pruning (SP) and dominant pruning (DP), using one-hop and two-hop neighbor information, respectively, to reduce redundant transmissions. Based on [29], Lou et al. further propose two extended algorithms: total dominating pruning (TDP) and partial dominant pruning (PDP) [12], which generate a smaller relay node set than the DP and SP algorithms do.

• **Location-based:** The authors in [31] propose a location based broadcast algorithm by calculating the curved convex hull.

• **Network coding:** On top of all these protocols, network coding schemes such as COPE [18] and CODEB [19] can deduce the broadcast transmission by retransmitting several lost packets with one coded packet.

Previous studies on wireless links focus on packet receptions of individual receivers [47,

Notation	Description
$e = \{u, v\}$	A link or transmission event from node u to v
$p(e)$	Link quality, measured by the transmission successful probability
$N(u)$	Node u's one-hop neighbor set, $N(u,v) = N(u) \cap N(v)$ is the common neighbor set of u and v
$K_i(u)$	A subset of i nodes among u's neighbors with the highest link quality
$p(K(u))$	Joint Packet Reception Probability (JPRP), a joint probability of links from u to $k(u)$ nodes
$Pr(v\|K(u))$	Set link probability, a conditional probability
$\epsilon(u)$	The expected transmission count for u to reliably broadcast one packet

Table 3.1: Notations used in this chapter

48]. The phenomenon of link correlation has recently been experimentally studied in [42, 49]. Zhu et al. [42] propose a probabilistic flooding algorithm to reduce energy consumption in transmission by using implicit ACKs inferred from link correlation. In [41], the authors explore a metric called κ that captures the degree of packet reception correlation on different links. The κ value of a network can be used to help network designers decide which protocol should be used for the network. As a generic correlation-aware middleware, the work is different from the aforementioned works.

3.2 Motivation

In this section, we present empirical studies to demonstrate the existence of link correlation. Then, we demonstrate theoretically that the performance of network protocols can be improved by exploiting link correlation. Finally, we show the impact of link blacklisting on broadcasting.

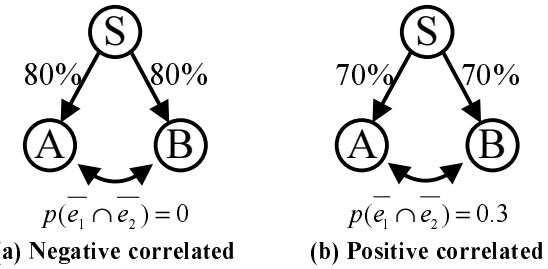


Figure 3.1: Impact of link correlation on broadcast.

3.2.1 How Link Correlation Affects Broadcast

This section theoretically analyzes the impact of link correlation upon the energy efficiency of broadcasting. Table 3.1 summarizes some notations that will be used in this chapter. We demonstrate that a node incurs *fewer* transmissions when the packet receptions of its one-hop neighbors have a *higher* correlation. Figure 3.1 shows two simple 3-node clusters where S is the source node and A and B are two receivers. If link quality is the only factor to be considered, intuitively we could deduce that, node S in Figure 3.1(a) should have fewer transmissions because the link quality (80%) in this cluster is higher than that in Figure 3.1(b) (70%). If link correlation is considered, however, this intuition no longer holds, as is evident from the following analysis: We use $p(e_i) \in (0, 1]$ to denote the probability that a source node can directly deliver a packet via link e_i. Let $p(e_1)$ and $p(e_2)$ denote the link qualities for the two children respectively. Corresponding packet loss probabilities are denoted as $p(\bar{e_1}) = 1 - p(e_1)$ and $p(\bar{e_2}) = 1 - p(e_2)$. Let $p(\bar{e_1} \cap \bar{e_2})$ denote the probability that a broadcasting packet from a parent is not received by either child. For an arbitrary positive integer threshold k, we denote ϵ as the number of transmissions a parent node needs to deliver the packet to its two children. The probability that ϵ exceeds the threshold k satisfies the following equation:

$$Pr(\epsilon > k) = p(\bar{e_1})^k + p(\bar{e_2})^k - p(\bar{e_1} \cap \bar{e_2})^k$$

Taking the difference yields

$$Pr(\epsilon = k) = Pr(\epsilon > k - 1) - Pr(\epsilon > k)$$

Then the expected number of transmissions $\mathrm{E}[\epsilon]$ to cover two children nodes can be calculated as

$$\begin{aligned} \mathrm{E}[\epsilon] &= \sum_{k=1}^{+\infty} k \cdot Pr(\epsilon = k) \\ &= \sum_{i=1}^{2} \frac{1}{p(e_i)} - \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2})} \end{aligned} \tag{3.1}$$

We note that $p(\bar{e_1} \cap \bar{e_2})$ obtains its maximum value when links e_1 and e_2 are perfectly positive correlated while it gets a minimum value when links e_1 and e_2 are perfectly

negative correlated. Let us revisit the two clusters in Figure 3.1. For the cluster in Figure 3.1(a), the expected number of transmissions E[ϵ] is 1.5 based on Eq.(3.1), given that $p(\bar{e_1} \cap \bar{e_2}) = 0$ since SA and SB are perfectly negative correlated. In Figure 3.1(b), however, since SA and SB are perfectly positive correlated, it is not difficult to get that $p(\bar{e_1} \cap \bar{e_2})$ is 0.3. As a result, E[ϵ] is 1.43 which is less than the cluster in Figure 3.1(a) even though it has better links. This suggests that we need to take link correlation into consideration in energy efficient broadcast and if we can manage to increase the link correlation within each cluster, then the number of transmissions can be significantly reduced. This can be done without modifying the broadcast algorithms at all – we simply *blacklist* certain links in the original network topology when they are found to be poorly correlated with others. By doing so, the upper layer broadcast protocols automatically avoid using them, thereby forming better-correlated clusters.

3.2.2 Link Blacklisting for Better Correlation

For the sake of clarity, we explained the impact of link correlation using a hypothetical example in the previous section. In this section, we present statistical evidence obtained from a physical setting. In the experiment, a sender is placed in the center, and the other 10 nodes are randomly deployed as single-hop receivers to receive 100 packets (i.e., a star topology).

In a practical broadcast protocol, some of the forwarder's neighbors may be dominated by other forwarders. In this experiment, we let the sender cover only five out of the ten nodes and measured the average number of transmissions to guarantee the successful reception of a single packet by the selected five nodes. There are $\binom{10}{5} = 252$ different combinations of such five nodes (out of ten). Then, we manually blacklist one negatively correlated link from the ten links and conduct the experiments again.

Figure 3.2 shows the transmission CDF before and after blacklisting. From Figure 3.2, we find that the average number of transmissions before blacklisting varies from 1.76 to 7.13 and is mainly concentrated around 2.8 and 5.5. The results after blacklisting show that the average number of transmissions for covering arbitrary five nodes (out of the nine receivers) is around 2.4. In other words, in this particular experiment a simple blacklisting of one link improves the link correlation, leading to a significantly reduction in the number of transmissions.

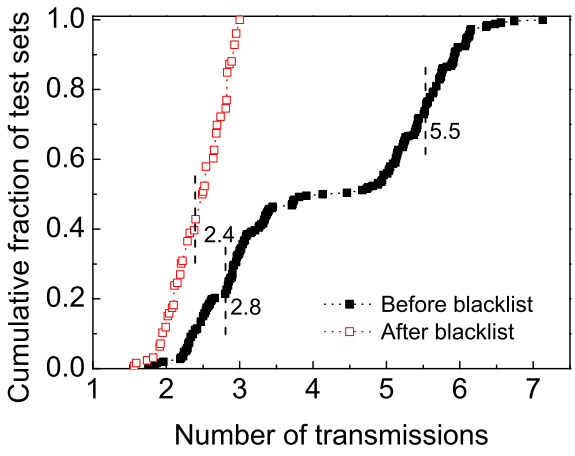


Figure 3.2: The effect of blacklisting on transmissions.

3.3 The Design of CorLayer

Since a wireless network has a much more complex structure than a star topology as we describe in Section 3.2.2, it is not practical to manually blacklist wireless links and an advanced design is needed. This section presents CorLayer, which handles link blacklisting automatically and transparently in an arbitrary topology. CorLayer is a supporting layer above the network topology, such as neighbor discovery and transmission power control, and beneath the network communication layer in which the broadcast protocol resides. In what follows, the design principles and core ideas of CorLayer are presented in Section 3.3.1. We then introduce the basis of CorLayer – the expected transmission count of broadcast – in Section 3.3.2. To support efficient calculation, we propose a heuristic to reduce computational cost in Section 3.3.3. The general design is presented in Section 3.3.4.

3.3.1 Design Insight and Principles

Generally, broadcast algorithms work as follows. Given the network topology, they select a subset of nodes as forwarders (in a centralized or distributed manner). These forwarders, called dominators, should be connected so that they alone can forward

packets. Though different broadcast algorithms differ in their way of selecting the forwarder nodes, they share the common feature that every dominator is responsible for covering its one-hop non-dominators (called dominatees), and every dominatee must be covered by at least one of its one-hop dominators. This feature provides the key insights we used to exploit link correlation when building CorLayer.

Recalling the example in Figure 3.1, roughly speaking, a dominator needs fewer transmissions when its covered dominatees have a higher link correlation. In other words, we can safely blacklist a wireless link and provide a better network topology to upper layer protocols if we can ensure that (i) the link correlation is increased by blacklisting the link and; (ii) the whole network is still connected.

The blacklisting procedure involves several key challenges that are addressed in this section. First, we need an efficient algorithm to assess the cost of covering one-hop neighbors, taking link correlation into consideration. Second, we need a low-cost method to deal with changes of link dynamics and maintain fresh values over time. Third, we need a localized light-weight algorithm for connectivity check. It is worth noting that CorLayer is constructed in a fully distributed manner where only one-hop neighbor information is needed. For the sake of description, we first assume that the link correlation between links within one-hop is known, and explain how to efficiently obtain the link correlation in Section 3.3.3.

3.3.2 Expected Transmission Count

In this section, we present the model by which a node assesses its cost to cover the one-hop neighbors in the presence of link correlation. With such assessments, we derive the relation between node cost and the link correlation, which provides an essential guideline for CorLayer design.

We assess the cost of a node' covering its neighbors using the expected transmission count $\epsilon(u)$, i.e., an expectation of how many transmissions u needs to successfully transmit a packet to all its neighbors. The total cost of a reliable broadcast is thus $\xi = \sum \epsilon(u)$ where u is a dominator.

From the theoretical analysis in Section 3.2.1, we know that to get $\epsilon(u)$ with two covered nodes, we need to compute $\binom{2}{1} + \binom{2}{2} = 3$ polynomial terms. Indeed, for more general cases of M covered nodes, the computational complexity of $\epsilon(u)$ is on the order of

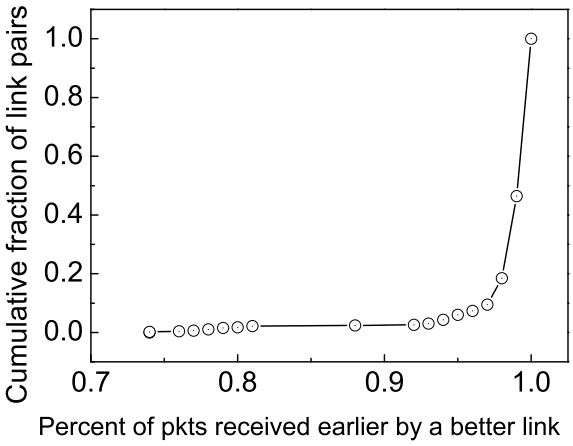


Figure 3.3: Statistics of receiving probability.

obtaining all possible logical combinations, i.e., $2^M - 1$. Although in wireless networks, the number of covered nodes M is relatively small, the exponential growth of complexity with M shall be avoided when possible. In the following section, we present a novel approach to exploiting conditional probability in concurrent receptions to simplify the calculation.

3.3.3 Transmission Count Approximation

Concerning the cost of computing $\epsilon(u)$, we seek a more efficient algorithm to approximate $\epsilon(u)$ with lower computational complexity. It is noted that in wireless broadcast, the nodes with a higher link quality usually receive the broadcast packet before those with a lower link quality [42]. To further confirm this observation, we deploy 31 nodes near a sender u, which broadcasts a packet every 0.2s. The total number of packet broadcasts is 1000. The receivers keep the packet sequence number and time stamp. After collecting the packet reception trace, for each packet, we compare the reception between each link pair (there are $\binom{31}{2} = 465$ such pairs). Figure 3.3 shows that the node with a better link from u receives about 98% of the packets earlier (or at the same time) than the node with a worse link from u.

Based on this observation, we propose a heuristic algorithm to approximate $\epsilon(u)$

denoted as $\hat{\epsilon}(u)$. For a given node u, we use $N(u)$ to denote u's one-hop neighbor set and $|N(u)|$ is u's out-degree. We define

Definition 1. *(**Joint Packet Reception Probability**) Suppose u is transmitting a packet to a nonempty neighbor set $K(u) \subset N(u)$. We define u's Joint Packet Reception Probability (JPRP) as the probability that all the nodes in $K(u)$ successfully receive a packet, denoted as $p(K(u))$*

When $|K(u)| = 1$, i.e., u has only one neighbor, JPRP equals to the quality of the link from u to this neighbor. Without loss of generality, assume the link qualities of the M covered nodes satisfy $p(e_1) \geq p(e_2) \geq \ldots \geq p(e_M)$, and the set of the first $m \leq M$ covered nodes is $K_m(u) = \{v_1, v_2, ..., v_m\}$.

Definition 2. *(**Set Link Correlation**) Given a source node u, a non-empty neighbor set $K(u) \subset N(u)$ and a receiver v that is not in set $K(u)$ (i.e., $v \in N(u) - K(u)$), the set link correlation $Pr(v|K(u))$ between $K(u)$ and v is the conditional probability that v successfully receives the packet under the condition that all the nodes in $K(u)$ receive the packet, i.e.,*

$$Pr(v|K(u)) = \frac{p(K(u) \cap v)}{p(K(u))}. \tag{3.2}$$

We note that the set link correlation more generically characterizes correlations among links. When $|K(u)| = 1$, the set link correlation reduces the traditional pair link correlation defined before [41, 42]. With these concepts, we are able to approximate the cost that a node delivers a message to its one-hop neighbors.

Lemma 3.3.1. *(**Approximation of** $\epsilon(u)$) Assuming nodes with higher link quality receive the broadcast packet earlier than those with lower link quality, u's expected transmission count with M covered nodes is approximated by*

$$\hat{\epsilon}(u) = \sum_{i=1}^{M} \frac{1}{p(e_i)} - \sum_{i=2}^{M} \frac{1}{p(e_i)} \cdot \frac{p(K_i(u))}{p(K_{i-1}(u))} \tag{3.3}$$

Proof.

$$\begin{aligned}\hat{\epsilon}(u) =& \frac{1}{p(e_1)} + \frac{\Pr(\bar{e_2}|p(e_1))}{p(e_2)} + ... + \frac{\Pr(\bar{e_M}|\bigcap_{i=1}^{M-1} p(e_i))}{p(e_M)} \\ =& \frac{1}{p(e_1)} + \frac{p(K_1(u)) - p(K_2(u))}{p(e_1) \cdot p(e_2)} + ... \\ & + \frac{p(K_{M-1}(u)) - p(K_M(u))}{p(K_{M-1}(u))p(e_M)} \\ =& \sum\nolimits_{i=1}^{M} \frac{1}{p(e_i)} - \sum_{i=2}^{M} \frac{1}{p(e_i)} \cdot \frac{p(K_i(u))}{p(K_{i-1}(u))}\end{aligned}$$

□

Special Case: Note that Eq.(3.3) includes the special case that the links are independent. When links are all independent, we have

$$p(K_i(u)) = p(e_i) \cdot p(K_{i-1}(u))$$

And $\hat{\epsilon}(u)$ reduces to

$$\hat{\epsilon}(u) = \sum\nolimits_{i=1}^{M} \frac{1}{p(e_i)} - M + 1. \tag{3.4}$$

Theorem 3.3.2. *The higher the set link correlation, the smaller the expected number of transmissions* $\hat{\epsilon}(u)$.

From Eq.(3.3), we find that the expected number of transmissions is composed of two parts: $\sum_{i=1}^{M} \frac{1}{p(e_i)}$ and $\sum_{i=2}^{M} \frac{1}{p(e_i)} \cdot \frac{p(K_i(u))}{p(K_{i-1}(u))}$. The first part is involved with the quality of each outgoing link and the second part is introduced by the link correlation. As the second part is always positive, the higher the set link correlation, the smaller the expected transmission count.

Implementation: To calculate $\hat{\epsilon}(u)$, we need to get every link's quality $p(e)$ and $p(K_i(u))$ for $\forall i = 2, ..., M$. Suppose each node maintains a packet reception bitmap (e.g., [1001]) recording the reception status of a fixed number (e.g., 4) of most recent packets. The link quality is given simply by the number of 1s in the bitmap divided by the bitmap length. The calculation of JPRP deserves a little more explanation.

In the design, every node broadcasts a probe packet to its neighbors at an adaptive time interval, the length of which is adjusted based on the link's stability. Every probe

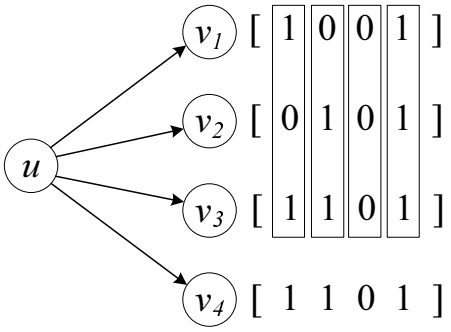


Figure 3.4: Example of Calculating u's JPRP for $\{v_1, v_2, v_3\}$

message is identified by the node ID and a packet sequence number. It is used not only for neighbor discovery, but also for updating the link quality and JPRP. Each node exchanges its reception bitmap with its neighbors. Assume the bitmap length is W, we have

$$p(K_i(u)) = \frac{1}{W}\sum_{k=1}^{W} B_{v_1}(k)\&B_{v_2}(k)\&\ldots\&B_{v_i}(k), \tag{3.5}$$

where $B_{v_i}(k)$ is a bit representing the neighbor v_i's reception status of the kth probe packet. $B_{v_i}(k) = 1$ if node $B_{v_i}(k)$ receives the packet, otherwise $B_{v_i}(k) = 0$. For example, in Figure 3.4, node u has 4 neighbors. we calculate u's JPRP for $\{v_1, v_2, v_3\}$. Suppose the bitmap of node v_1 is [1001], which indicates that v_1 receives the 1st and 4th packets and misses the 2nd and 3rd packets. When node u receives the bitmaps from all its neighbors, it can use Eq.(3.5) to calculate the corresponding JPRP $p(K_3(u)) = (1\&0\&1 + 0\&1\&1 + 0\&0\&0 + 1\&1\&1)/4 = 25\%$.

Overhead under Link Dynamics: In dynamic network environments, both link quality and set link correlation change over time. The nature question is that how much overhead is required to maintain these values fresh? To answer it, we conduct a set of experiments, in which the source node keeps broadcasting packets to ten receivers in every 3s while the probe packet is sent in every 32s. The main overhead of the design comes from two sources. First, we need to broadcast probe packets. The energy cost of this part is about 3.5% of the total energy in the setting. Note that periodically sending probe packets has been already required by other protocols [47, 50] to measure link quality or to improve the robustness of routing structure. We argue such cost is not introduced solely by the design, hence it should be amortized. Second, we need to

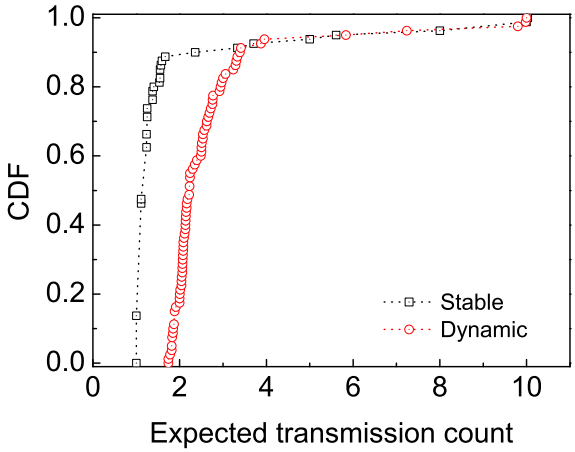


Figure 3.5: CDF of ϵ in stable and dynamic scenarios.

exchange packet reception bitmaps among one-hop neighbors in order to calculate the values of set link correlation, which is exclusively introduced by the blacklisting algorithm. Fortunately, binary bitmaps are small and are much less frequently exchanged, therefore the overhead occupies a tiny fraction (0.9%) of the total energy cost according to the measurements.

Estimation Accuracy under Link Dynamics: We run the experiments under both stable and dynamic scenarios in a period of eight hours, during which probe packets (3.5% overhead) and bitmap exchange (0.9% overhead) are used to fresh link quality and correlation values. As shown in Figure 3.5 and 3.6, we maintain the estimation of expected number of transmission ϵ accurately over time. Specifically, Figure 3.5 shows the CDF of ϵ, from which we can see that the metric ϵ is stable. Figure 3.6 shows the number of received packets (during a time period of 5 minutes) closely follows the number of sent packets (i.e., 100) divided by ϵ.

3.3.4 The Process of Link Blacklisting

As mentioned before, the objective of link blacklisting is to increase link correlation among neighbors of a dominating node (say u) by disabling a subset of its links, so

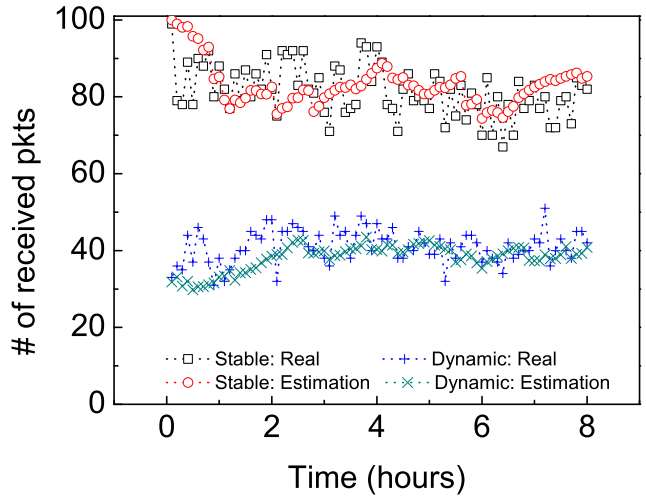


Figure 3.6: Received pkts vs. Estimation

that we can statistically reduce the number of transmissions $\epsilon(u)$. To achieve this goal, we implement the design of link blacklisting as a transparent layer – CorLayer. There are two key steps in the design. One is a connectivity check: a network should not be partitioned because of blacklisting. The other is a blacklisting efficiency check: blacklisting should not increase the number of transmissions to cover neighbors via alternative paths. We explain how we address these two issues in Section 3.3.5 and Section 3.3.6.

3.3.5 Connectivity Check

Clearly, when blacklisting a link, CorLayer must not disconnect a network. Otherwise, the broadcast protocols running above CorLayer will not function correctly. Furthermore, we need to guarantee connectivity in a distributed fashion, in which a node needs only one-hop information and blacklist links asynchronously. the design is simple yet effective. A link between node u and v can be blacklisted from the network only when at least a common neighbor (say w) of both u and v exists. In other words, eliminating a link requires the existence of an alternative path through at least a common neighbor. A similar idea has been used to guarantee connectivity when GPRS [51] constructs a reduced planar RNG graph.

Since blacklisting is performed asynchronously, we need to avoid a race condition. This is achieved by traditional two-phase locking. A node u that attempts to blacklist a link $u \rightarrow v$ first sends a lock messages to both v and their common neighbor w to "lock" them, requiring that v and w cannot perform link blacklisting at the same time as u. Only when u has received confirmation from both v and w can u start to blacklist its link to v. If u does not receive confirmation as expected or experiences a timeout (we assume a certain timeout period), u gives up the blacklisting. In either case, u will send a message to "unlock" v and w. v and w also start a timer when receiving a "lock" message, and release the lock when they receive an unlock message or the timer expires (for fault-tolerate purpose). This lock/unlock mechanism is very lightweight as it involves only one-hop neighbors and requires maintaining little state information (lock/unlock state and a timer), and so incurs negligible overhead.

3.3.6 Efficient Link Blacklisting Rule

In addition to ensure network connectively, more importantly, we need to make sure network efficiency improves using the CorLayer. Specifically, blacklisting needs to ensure that all neighboring nodes of a dominating node u can be covered with a *reduced* expected number of transmissions after a link is blacklisted. As shown in Figure 3.7, the key idea is to blacklist a link from $u \rightarrow v$ if the source node u could take fewer transmissions to broadcast a packet to v via intermediate nodes (two-hop broadcast) than broadcasting the packet directly to v.

Let a link from the transmitter to the receiver be $u \rightarrow v$, and the common neighbor set of nodes u and v be $N(u, v)$. Recall that in the process of connectivity checking, the algorithm guarantees that there exists at least one common neighbor (say w) of u and v, that is, $|N(u, v)| \geq 1$. Let $\epsilon_{N(u)-\{v\}}(u)$ be the expected number of transmissions for node u to broadcast a packet to all of its neighbors except node v. The efficient blacklist rule for the link $u \rightarrow v$ is:

$$\epsilon(u) - \epsilon_{N(u)-\{v\}}(u) > \frac{\epsilon_{N(u)-\{v\}}(u)}{|N(u)| - 1} + \frac{\epsilon(w)}{|N(w)|} \tag{3.6}$$

The left-hand part of Eq.(3.6), $\epsilon(u) - \epsilon_{N(u)-\{v\}}(u)$, is the additional number of transmissions for node u to cover node v directly through broadcast, compared with covering

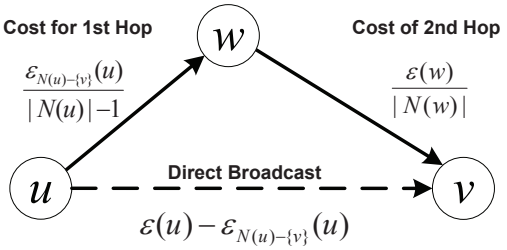


Figure 3.7: Triangular Blacklist Rule

only the neighbor set $N(u) - \{v\}$ (denoted as directed broadcast cost in Figure 3.7).

Intuitively, the link $u \to v$ is worth eliminating if node u could take fewer transmissions to broadcast the packet to v via the intermediate node w, a cost calculated by the right-hand part of the Eq.(3.6). Specifically, the right-hand part of the Eq.(3.6) is the sum of per-link two-hop broadcast: (i) $\frac{\epsilon_{N(u)-\{v\}}(u)}{|N(u)|-1}$ is the per-link cost for the first-hop broadcast between u and w, after the link $u \to v$ has been blacklisted (Note $|N(u)| - 1$ is the size of node u's neighbor set without considering node v). (ii) $\frac{\epsilon(w)}{|N(w)|}$ represents the per-link cost for the second hop broadcast between w and v. These two costs are shown in Figure 3.7 as "cost for 1st hop" and "cost for 2nd hop".

In summary, Eq.3.6 presents a *Triangular Blacklist Rule* shown in Figure 3.7: When it takes fewer transmissions to deliver a packet via alternative two-hop broadcast paths than broadcasting the packet to v directly, we blacklist the link. We note that more aggressive blacklist rules could be tailored for particular broadcast designs; the design conservatively eliminates links that are clearly detrimental to broadcast performance. Such a general design ensures that CorLayer, as a middleware, can improve a wide range of broadcast protocols.

When $|N(u, v)| > 1$, multiple alternative paths exist. In this case, we shall model the average cost of these alternative paths among $|N(u, v)|$ paths. Accordingly Eq.(3.6) shall be revised to a more generic one, which is used in the final design.

$$\epsilon(u) - \epsilon_{N(u)-\{v\}}(u) > \frac{\sum \frac{\epsilon_{N(u)-\{v\}}(u)}{|N(u)|-1} + \frac{\epsilon(w_i)}{|N(w_i)|}}{|N(u, v)|} \tag{3.7}$$

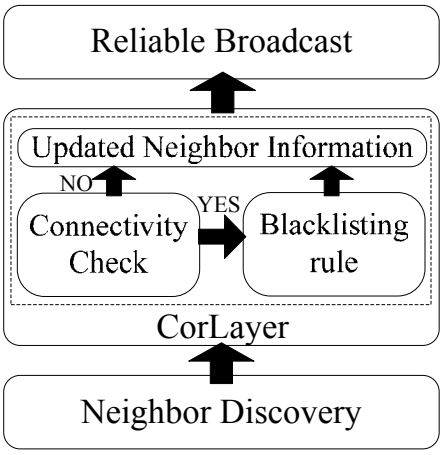


Figure 3.8: CorLayer in the protocol stack

3.3.7 CorLayer Embedding

CorLayer is designed as a generic middleware to assist a wide range of broadcast protocols and be compatible with other energy efficient MAC layers such as low power listening (LPL) [52]. To do that, we insert CorLayer beneath the broadcast protocol and above the MAC layer. As shown in Figure 3.8, CorLayer provides a reduced topology to the upper layer broadcast protocols in a transparent manner.

Protocol Name	Network Info.	Probe Msg	Category
S-Tree [27]	One-hop	ID	Tree-based
C-Tree [25]	Quazi-Global	Global	Tree and Cluster-based
Cluster [33]	Local	ID	Tree and Cluster-based
Intermediate [32]	Two-hop/One-hop	One-hop/Position	Cluster-based
Clustering [30]	Quazi-Local	Degree	Cluster-based
P-Clustering [28]	Two-hop	None	Cluster-based
MPR [13]	Two-hop	One-hop	Multi-point relay
Mini-id MPR [24]	Two-hop	One-hop	Multi-point relay
MPR-CDS [24]	Two-hop	One-hop	Multi-point relay
SP [29]	One-hop	One-hop	Pruning-based
DP [29]	Two-hop	One-hop	Pruning-based
Pruning [12]	Two-hop	One-hop	Pruning-based
TDP [12]	Two-hop	One-hop	Pruning-based
RNG [26]	Two-hop	One-hop	Pruning-based
CCH [31]	One-hop	One-hop + Position	Location-based
CODEB [19]	Two-hop	One-hop	Pruning-based+NC

Table 3.2: Sixteen state-of-the-art protocols supported by CorLayer

(a) Testbed in UMN (b) Testbed in SIAT (c) Testbed in TRIMPS

Figure 3.9: Testbed experiment

Platform	Location	Environment	Nodes/APs
MICAz	UMN	Lab	36/5
TelosB	SIAT	Office	30/8
GreenOrbs	TRIMPS	Outdoor	20/0
Physical Size	Degree	Channel	Power
$8m \times 2.5m$	$7 \sim 23$	Ch16	-25dBm
$18m\times$ 13m	$6 \sim 21$	Ch16, Ch26	-25dBm
$15m \times 5m$	$4 \sim 13$	Ch16	-25, -19.2dBm

Table 3.3: Testbed settings and topology properties

We classify the existing reliable broadcast algorithms into tree-based [27, 40], cluster-based [25, 28, 30, 32, 33], multi-point relays [13, 24, 39], pruning-based [12, 26, 29], and location based [31]. Recently, network coding (NC) has been adapted to support broadcast applications in wireless networks, e.g., COPE [18] and CODEB [19]. CorLayer also supports these network coding schemes and helps them save transmissions. Besides, in low-duty-cycle networks, a common wake-up time unit is assigned to nodes sharing common senders. CorLayer may improve the energy efficiency of low-duty-cycle protocols by helping them form clusters with higher correlation thus the nodes in the same cluster receive broadcasting packets simultaneously. Thus far, we have successfully implemented sixteen classical algorithms and embedded CorLayer with them. The basic information of these protocols is shown in Table 3.2.

3.4 Testbed Experimentation

Packet reception patterns vary significantly across network environments, as they are affected by environmental noise and external interference. We evaluate CorLayer on three testbeds, whose basic information is shown in Table 3.3. The first testbed is in a dedicated lab environment, in which a total of 36 MICAz nodes are randomly deployed on a $8m \times 2.5m$ wall, as shown in Figure 3.9(a). The second testbed consists of 30 TelosB nodes deployed in an $18m \times 13m$ open office environment following a grid pattern; see Figure 3.9(b). The third testbed is an outdoor environment (Figure 3.9(c)), in which 20 GreenOrbs nodes were deployed on the grass-covered curb along a river.

In all three testbeds, the default transmission power is set at -25dBm so that the nodes form multi-hop networks. The default channel is 16. After deployment all nodes are synchronized and start the neighbor discovery by sending out probe packets, based on which we get the link quality and set link correlation information about their one-hop neighbors. For broadcast without blacklisting, two nodes are considered as neighbors when the link quality between them is greater than 0.2. With CorLayer, it updates the one-hop neighbor information based on the Link Blacklist Rule (in Section 3.3.6). Based on the one-hop neighbor information, forwarders and their corresponding covered nodes are determined by using sixteen existing reliable broadcast protocols. In the experiment, each protocol sends out 20 data packets with a time interval of 2 seconds. For performance analysis purposes, each packet includes information – time stamp, hop count, and previous hop's node ID. Upon receiving the data packets, the intermediate node records the number of transmissions for each data packet.

3.4.1 Performance Metrics

We use the total number of transmissions needed to deliver one packet to all the nodes in the network as the metric for evaluating the energy efficiency of a broadcast protocol either with or without CorLayer. Furthermore, energy gain is defined as the percentage of saved transmissions.

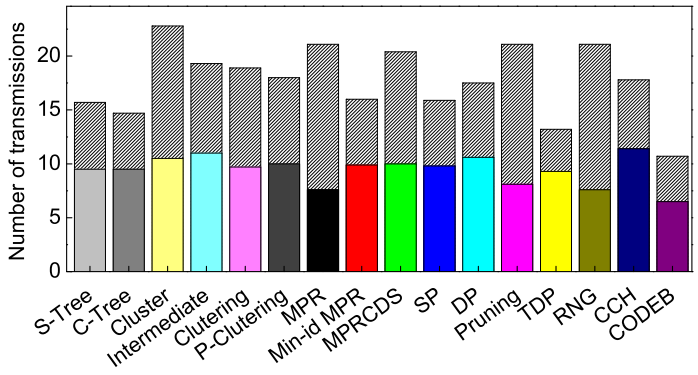


Figure 3.10: Improvements over 16 protocols

3.4.2 Main Performance Results

The experimental results of the sixteen classical reliable broadcast protocols are shown in Figure 3.10. The upper parts of the bars (in gray) represent the proportion of transmissions reduced by the blacklisting method. For example, for the MPR algorithm, the nodes need 21.1 transmissions on average to guarantee that every node in the network receives one packet, while the number is 7.6 after blacklisting, achieving a reduction of 64%. The average transmission of different protocols before and after blacklisting is 17.8 and 9.4, respectively. Compared with the schemes without using network coding, CODEB saves 41.2% transmissions. the design makes a further 39.3% improvement upon CODEB. On average, the blacklist design reduces transmissions by 47.2%. The reason why the design has better performance is as follows: Low link correlation may cause the nodes in a cluster losing different packets. The source node of a lower correlated cluster needs to retransmit more packets. By blacklisting those low correlated links, the upper layer broadcast protocols automatically avoid using them, thereby forming clusters with high correlation. Besides, in high correlated clusters, a transmission can recover the lost packet for multiple receivers.

Although we have collected results for all the sixteen protocols, space constraints do not allow presenting all of them here. Therefore, we choose four representative broadcast algorithms, namely Multi-Point Relay [13] (MPR for short), Forwarder Node

Method	Blacklist Rule
R_B	Each node blacklists $x\%$ of links ($x = 10$ in the experiment).
WL_B	Each node blacklists the worst $x\%$ of links ($x = 10$ in the experiment).
AVG_B	The design, please refer to Eq.(3.7).
MIN_B	A link is blacklisted when there exists one alternative path with lower cost.
MAX_B	A link is blacklisted when its broadcast cost is higher than all alternative paths.

Table 3.4: Different link blacklisting strategies.

Cluster [33] (Cluster for short), Partial Dominating Pruning [12] (Pruning for short), and CODE-B [19] for the rest of the experiments.

3.4.3 Impact of Blacklisting Rules

In this section, we consider alternative link blacklisting rules, including random blacklisting ("R_B" for short), worst link blacklisting ("WL_B"), AVG blacklisting ("AVG_B"), MIN blacklisting ("MIN_B"), and MAX blacklisting ("MAX_B"); see Table 3.4 for a brief description. We shall use "link blacklisting strategy_broadcast algorithm name" to represent a specified algorithm configuration. For example, WL_B_MPR means MPR with the worst link blacklisting strategy.

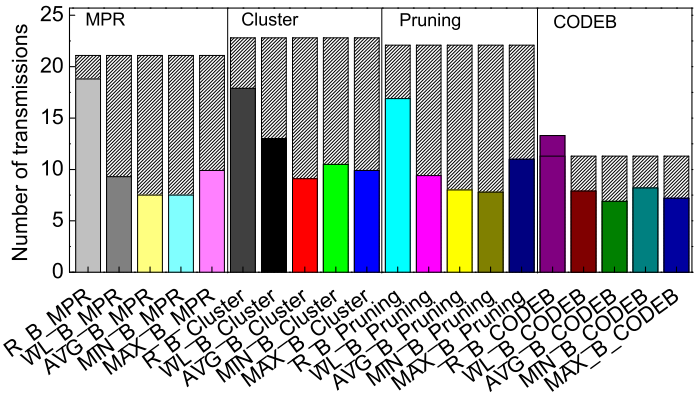
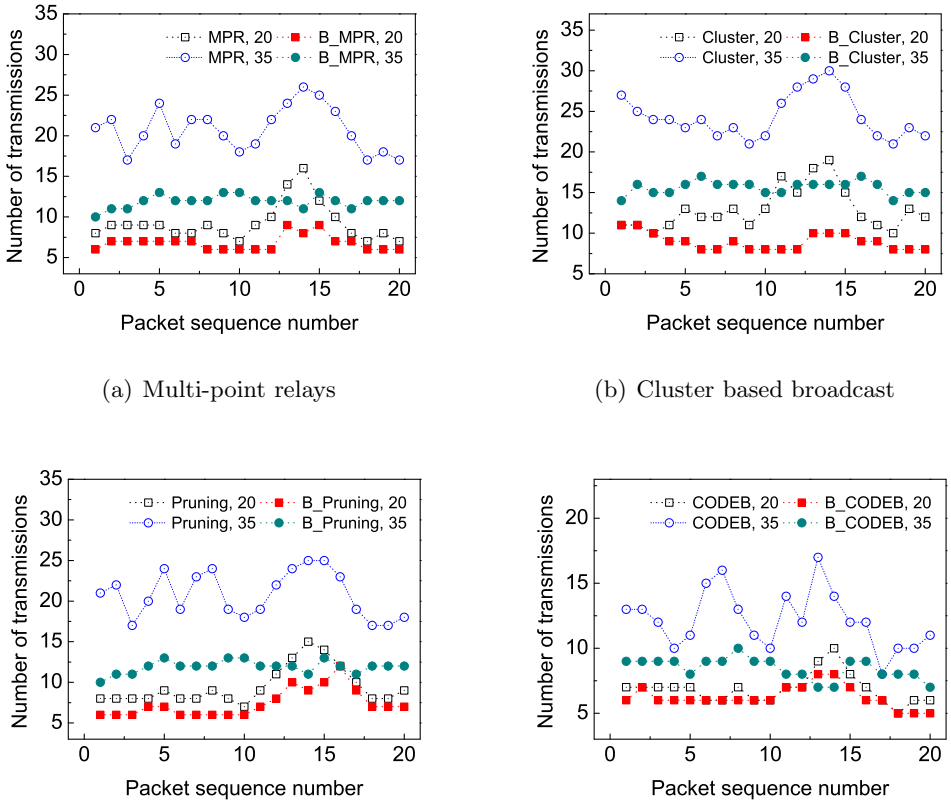


Figure 3.11: Impact of blacklist rules

Figure 3.11 shows the energy consumption of the four broadcast strategies with the five different blacklisting rules. On average, the design reduces the number of packet transmissions by 55.8%. The "MIN blacklisting" rule blacklists links when there exists an alternative path with a lower broadcast cost. This rule may blacklist too many links, and it is possible that the upper layer broadcast protocols do not select low-cost paths. For example, the energy consumption of "MIN blacklisting" with the pruning algorithm is 7.8, which is lower than that of the design. Yet, in the cluster algorithm, the energy consumption of "MIN blacklisting" is 10.5, while the design has an energy consumption of 9.1. The "MAX blacklisting" rule blacklists links when the broadcast cost of the link is greater than all the alternative paths. This rule guarantees that the upper layer protocols can always obtain energy gain from the blacklist strategy. It is too conservative, however, to obtain a further energy gain – many black sheep (the high cost broadcast links) failed to be blacklisted. From Figure 3.11, we see that the average energy gain of the "MAX blacklisting" rule is 48.9%. The worst link blacklisting produces an average energy gain of 45.7%. This strategy, however, faces two problems: (i) a threshold is required to blacklist x% of the worst links, but the threshold usually depends on the network environment, and (ii) it may remove some good quality links since it always blacklists x% worst links, which may well contain a good quality link. For the removed neighbors, there may exist no alternative paths with a lower message cost, so the removal can only reduce energy efficiency. The random blacklisting algorithm is a blind method, with an energy gain of only 18.6%. The negative effect of random blacklisting is understandable since some high-quality links may be removed.

3.4.4 Impact of Network Size

In this experiment, we use the data from the testbed in UMN. Figure 3.12 shows the total number of packets transmitted for each packet with 20 and 35 MICAz nodes. In the figure, the protocol x using blacklisting is labeled "B_x". It can be seen that without link blacklisting, the transmission count ranges from 5 to 30, while with link blacklisting, it ranges from 5 to 17. On average, the design obtains an energy gain of 31.3%. From Figure 3.12, we can also see that the trend of energy gain with increasing network sizes is quite stable, suggesting that the design scales well with large networks.



(a) Multi-point relays

(b) Cluster based broadcast

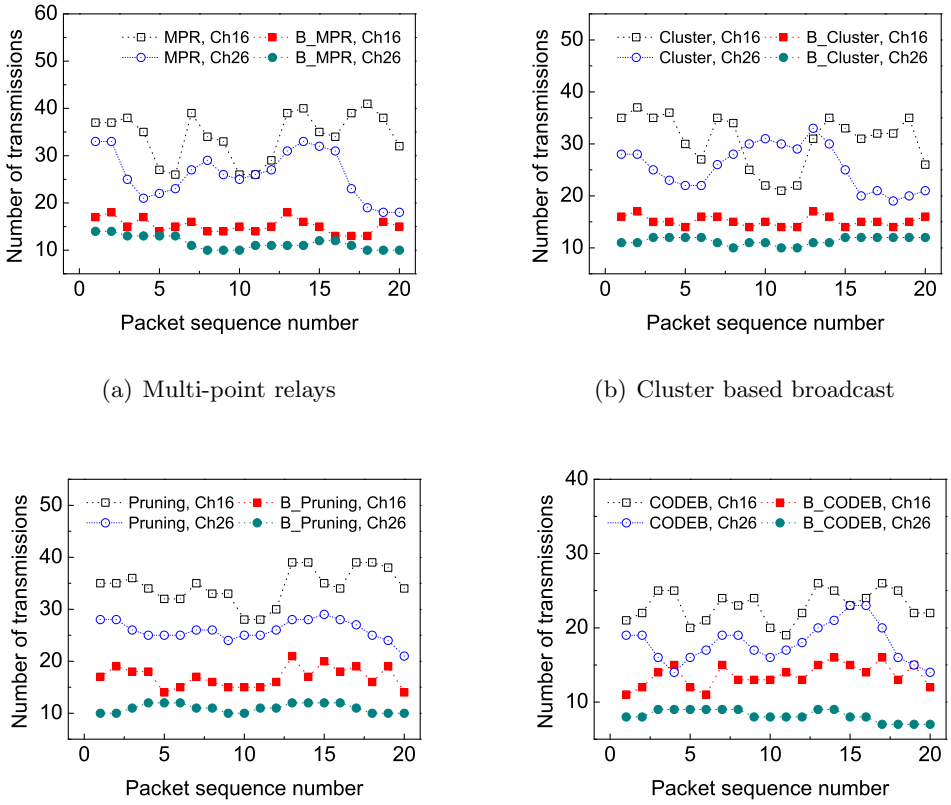
(c) Pruning based broadcast

(d) Network coding

Figure 3.12: Impact of network sizes

3.4.5 Impact of Different Channels

In this experiment, we explore the impact of channels on the design. We use the data from the testbed in SIAT that is in an open office environment that includes total of 8 access points. Note that channel 16 overlaps with a co-habiting access point's 802.11 channel and that channel 26 is free of Wi-Fi interference. We ran the experiments during normal office time. The power level for transmission is set to -25dBm. Figure 3.13 shows the energy consumption of the eight broadcast strategies for networks using two different channels – channels 16 and 26. From Figure 3.13, we can see that the broadcast protocols need more transmissions to finish the same task in channel 16. This is because the overlapped channel causes more packet losses. Besides, we find that the average gain



(a) Multi-point relays

(b) Cluster based broadcast

(c) Pruning based broadcast

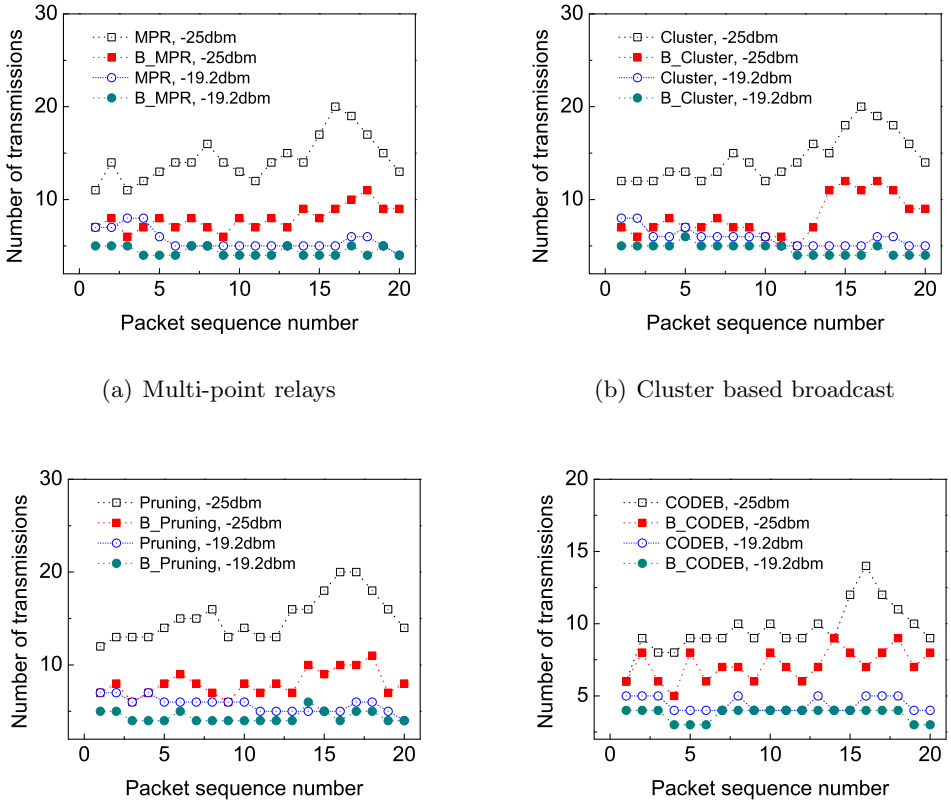
(d) Network coding

Figure 3.13: Impact of channels

of the design under the four broadcast algorithms is 56% using channel 26, and 50% using channel 16. This result shows that the design performs better using channel 26, the interference-free channel, because the interference of Wi-Fi signals makes the transmissions using channel 16 better correlated, a phenomenon consistent with the observation by Srinivasan et al. [41]. The better correlation thus leaves us less room to improve the energy efficiency since the gain relies on the improvement of link correlation.

3.4.6 Impact of Power Level

We conducted this experiment in an outdoor scenario where we can control the range among nodes freely; see Figure 3.9(c). The power level for transmission is set from



(a) Multi-point relays

(b) Cluster based broadcast

(c) Pruning based broadcast

(d) Network coding

Figure 3.14: Impact of power levels

-25dBm to -19.2dBm to form a multi-hop network. Figure 3.14 shows the transmissions of the four broadcast algorithms with and without link blacklisting for networks with different power levels. Again, we find that the link blacklisting greatly reduces transmissions for reliable broadcasting. Here we use the pruning algorithm as an example, although similar results have been observed with the other three algorithms. Under power level -25dBm, the transmission count for the pruning algorithm is 13.4 on average, while it is 7.8 after link blacklisting, providing a reduction of 42%. Under power level -19.2dBm, fewer transmissions are needed for broadcasting because a higher power level leads to better link quality. In this case, the link blacklisting layer still reduces transmissions by 20.1%.

3.5 Simulation Evaluation

In this section, we evaluate our CorLayer design in simulations with various network settings. To specify, we examine whether our design provides universal support for existing reliable broadcast algorithms in large-scale networks. Besides, adjusting the power level in the testbed experiment actually affects two things: link quality and network density (average node degree). Because both factors cannot be set directly on the testbed, we use simulations to study the performance trend. The following simulation results are the average values of 100 rounds over the same network settings.

3.5.1 Main Performance Results in Simulation

In the simulation, 200 nodes are randomly generated in a $1000m \times 1000m$ scenario. The communication range is 160m. Nodes' degrees range from 5 to 18 with an average of 12.1. for each node, we generate all its neighbors' packet reception information by modifying the sampling algorithm in [53]. For example, the packet reception bitmaps of a node's two neighbors may look like [111001] and [111001], where "1" means a packet is received and "0" means a packet is lost. In this example, the generated packet reception bitmaps have a perfect correlation. In this experiment, the worst link quality is 35% and the average link quality is around 70%.

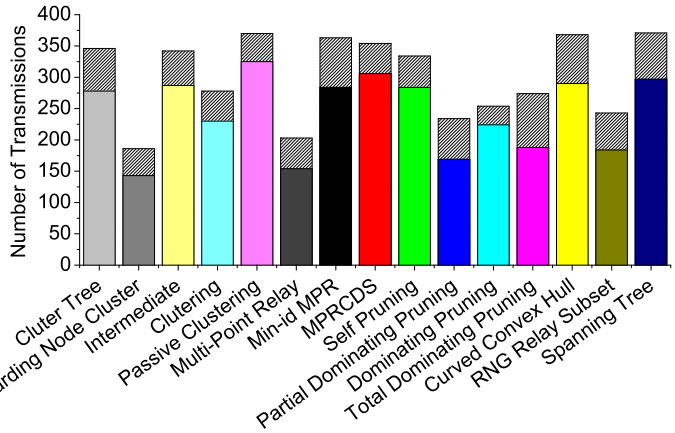


Figure 3.15: Improvements on 16 protocols (Simulations)

Figure 3.15 shows the energy consumption of the sixteen broadcast protocols in large-scale networks. The performance trends are similar to the testbed results, and we find all protocols can benefit from our design. In the 200-node network, on average, the protocols without link blacklisting need 325 transmissions to broadcast a packet. After blacklisting with CorLayer, the energy consumption is reduced to 238 transmissions, offering a 27% energy saving. From Figure 3.15, we can also find that the benefit of our design in simulation is lower than that in testbed. That's because in simulation, the link quality is set to be greater than 0.35 while in testbed, some links are probably in a very bad condition, i.e., less than 0.35 (but greater than 0.2).

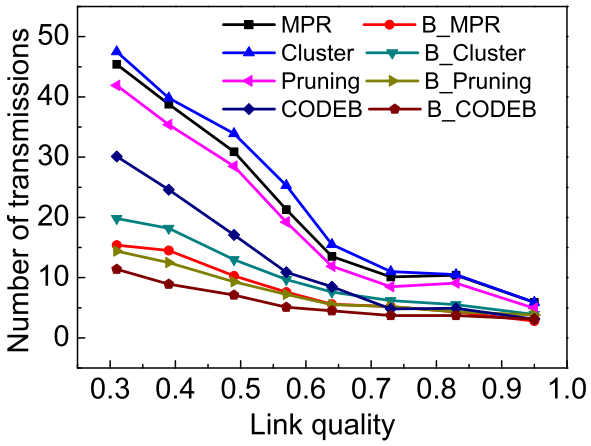
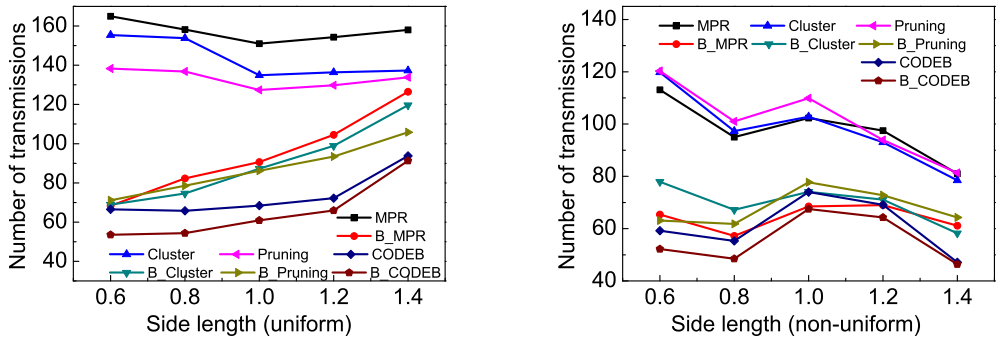


Figure 3.16: Impact of ink qualities

3.5.2 Impact of Link Quality

Let us consider the energy gain of CorLayer for networks with different link qualities. Since it is impossible to simply set a quality for each link on the testbed, it is necessary to vary link quality in a controlled way and look at each strategy's performance in different cases. In the experiment, we first collect the packet reception trace from the testbed (a 20-node scenario). Then, we introduce losses to each receiver using a link correlation packet loss model [53] that makes the one-hop receivers drop packets in a link correlated way with a controlled loss rate. The results are shown in Figure 3.16, where

we can see that the number of transmissions of our design varies from 19.8 to 2.8 when the average link quality varies from 0.31 to 0.95. The Cluster algorithm without link blacklisting, for example, needs 47.5 transmissions to finish the broadcast task when the average link quality is 0.31. Under the same condition, our design saves 58% of transmissions in this poor link quality scenario. For an increased average link quality of 0.95, our energy gain reduces to 25%, suggesting that the energy gain of our design decreases as link quality increases.



(a) Uniform distribution

(b) Non-uniform distribution

Figure 3.17: Impact of densities

3.5.3 Impact of Network Density

In this experiment, we consider both uniform and non-uniform node distribution. The network size is 64, and the field size is $800m \times 800m$ with a communication range of 160m. The average link quality is about 0.5. Figure 3.17(a) and 3.17(b) show the transmission counts of the eight broadcast strategies for networks with different densities. The average node degrees for side lengths (of the simulated square sensing field) 0.6, 0.8, 1, 1.2, 1.4 are, respectively, 20.2, 13.0, 8.4, 5.9, and 3.9. From Figure 3.17(a) and 3.17(b), we can see that with variation in density, the transmission count does *not* change monotonically. This is because with the increase of network density, a forwarder needs more transmissions to make sure all its covered nodes receive the packet, but the number of forwarders decreases in a fixed size (i.e., 64) network. In Figure 3.17(a) and 3.17(b), the energy gain of our design decreases as the side length increases (and

thus the density decreases). For example, in uniform case in Figures 3.17(a), the energy gain of our design under the MPR algorithm is 58.6% at an average node degree of 20.2, and it drops to 19.9% when the average node degree is only 3.9. As the network becomes denser, a forwarder tends to cover more nodes. This increases the possibility that links with poor link correlations are put into the same cluster, thus giving our algorithm more opportunities to improve the link correlation within the cluster.

3.6 Summary

In this chapter, we have presented CorLayer, a link correlation-based layer that enhances the energy efficiency of reliable broadcasting. This layer blacklists links with low correlation by following a triangular blacklisting rule, using only one-hop information. It is transparent to upper layer broadcast protocols, which can obtain significant gains without modifications. To test CorLayer's broad applicability and effectiveness, we integrated CorLayer transparently with sixteen state-of-the-art broadcast algorithms and evaluated the design on three real-world multi-hop testbeds: a network with 36 MICAz nodes, a network with 30 TelosB nodes, and a network with 20 GreenOrbs nodes. The results indicate that with CorLayer, reliable broadcast avoids unnecessary transmissions caused by wireless links that are less positively correlated.

Chapter 4

Link Correlation Aware Network Coding

4.1 Introduction

Wireless communication is essentially based on the broadcast medium with concurrent receptions. Network coding, which exploits the diversity benefit of wireless broadcast medium, has great potentials to improve the performance, e.g., the throughput and energy efficiency, in wireless communication [18, 22, 54]. For example, under the lossy broadcast channel, several nodes may lose different packets. With network coding, multiple missed packets are encoded together and then broadcast in a single transmission, thus improving the transmission efficiency.

Although some researchers are very optimistic about the decent performance of network coding, others express reservations about the benefits that network coding can obtain. They claim that network coding may only bring a negligible improvement while the coding cost may exceed the benefits. In real scenarios, this situation does happen. Take an extreme case for example, when the wireless links are perfectly positive correlated, network coding will not provide any improvement but coding overhead since all the receivers lose the same packets and there are no diversity benefits to exploit.

In this chapter, we introduce link correlation – a concept that captures the relationship of the packet receptions among the broadcast links, to network coding. In detail, network coding consists of two operations: coding and broadcast. On the one hand, the

coding operation prefers high spatial diversity (i.e., *low* link correlation). This is because when all receivers lose the same packets, network coding will not work better than the traditional routing protocols, as there is no coding opportunity [18] to exploit. On the other hand, the broadcast operation, in contrast, prefers low spatial diversity (i.e., *high* link correlation). This is because it takes fewer number of transmissions to deliver a coded packet to all receivers when the receptions of these nodes are correlated [55]. Clearly there exists a trade-off between the coding opportunity and broadcast effectiveness on the preference of link correlation. Ignoring this correlation in network coding protocol design may result in under-utilized diversity benefits [18, 41]. Even worse, because of the inaccurate link independent assumption, unnecessary coding operations exploited by coding aware routings [56, 57, 58] may lead to extra energy consumption and delay.

We thus propose correlated coding, a coding technology that (i) estimates the coding opportunity, (ii) measures the broadcast transmission efficiency, and (iii) quantifies diversity benefits that network coding exploits. Guided by correlated coding, a coding operation is executed only when necessary while diversity benefits are maximized. In summary, the contributions of this work are as follows:

• We experimentally show that the reception results of broadcasting packets at multiple receivers are not independent. This observation contradicts the widely made link independence assumption, which overestimates true diversity benefits that network coding can obtain in reality.

• A novel coding design named, correlated coding, is proposed to capture the expected number of transmissions with network coding under the effect of link correlation for both unicast and broadcast. As far as we know, this is the first work that explores link correlation both mathematically and experimentally in network coding.

• We experimentally verify the impact of correlated coding on three unicast and four broadcast protocols with one 802.11 testbed, and three 802.15.4 testbeds running TelosB, MICAz, and GreenOrbs nodes. The experiment results show that the design consistently enhances the performance of these protocols – the number of coding operations is reduced while the transmission efficiency is improved by $30\% \sim 50\%$.

The rest of this chapter is structured as follows. Section 4.2 reviews related works. Section 4.3 presents the motivation. Section 4.4 introduces the main design followed

by its applications in Section 4.5. Evaluation results from testbed experiments and simulations are shown in Sections 4.6 and 4.7. Finally, Section 4.8 concludes the chapter.

4.2 State of the Art

There exist two main bodies of studies, i.e., network coding and link correlation, which are closely related to this chapter. In the following, we first summary these existing works and then state the unique position of our work.

Network coding [17], which allows intermediate nodes to combine packets before forwarding, has the great potential to improve the transmission efficiency of both broadcast and unicast. Katti et al. propose the first practical network coding scheme named opportunistic coding (also known as COPE type network coding) [18]. In opportunistic coding, the coding strategy exploits the broadcast property of wireless channels and finds coding opportunities. Using this approach, multiple packets are encoded together and then broadcast in a single transmission, thus improving the transmission efficiency. Opportunistic coding does not fully exploit the benefits of network coding since the coding opportunity is dependent on the routing path and the coding un-aware routing strategy [18] misses many coding opportunities. Researchers thus propose coding-aware routings [56, 57, 58] to exploit more coding opportunities.

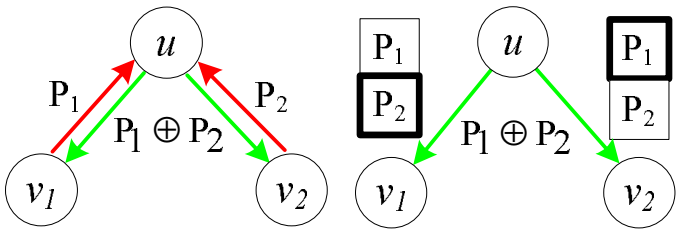
While some researchers are optimistic about the decent performance of network coding, others point out the improvement of network coding is marginal while the coding cost could be extremely high. The empirical study shows that the benefits of network coding change dynamically under different testbed measurements [41]. The coding opportunity highly depends on the packet reception information as well as the routing path construction, and its coding benefits could be marginal [56, 59].

In prior works, researchers explicitly or implicitly assume that the wireless links are independent [18, 19, 22, 56, 59, 60] when they exploit network coding benefits. This assumption, however, contradicts the empirical evidence that wireless links are correlated [41, 42, 61, 62]. For the first time, we introduce link correlation to model the packet reception information of the broadcast channel, which is further used to quantify the diversity benefits that network coding can exploit. Compared with the widely used link independence model, the model is more practical and accurate. With

the quantified coding benefits, the approach helps network designers decide whether to apply the network coding technology or not. Besides, the model can be further used to optimize the coding benefits when network coding is applied.

4.3 Motivation

In this section, we first demonstrate the existence of link correlation. Then, we introduce the basic idea of network coding, followed by the impact of link correlation on network coding.



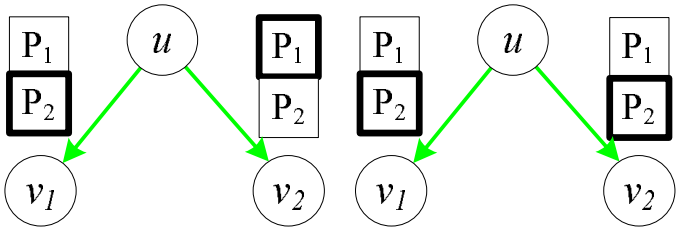
(a) Unicast scenario (b) Broadcast scenario

Figure 4.1: Examples of network coding occurs at unicast and broadcast scenarios.

4.3.1 The Idea of Network Coding

Network coding has great potential to improve the performance of both unicast and broadcast applications by allowing intermediate nodes to encode multiple packets together before forwarding. Figure 4.1 shows how network coding benefits both unicast and broadcast. In Figure 4.1(a), after node v_1 and v_2 send their packets to the relay node u, instead of sending packets p_1 and p_2 separately, node u broadcasts a coded packet $p_1 \oplus p_2$ with one transmission. In the broadcast scenario, a packet reception report is shown in Figure 4.1(b), in which a block with a thick borderline means a received packet, and a block with a thin borderline means a lost one. The receiver v_1 and v_2 lose packet p_1 and p_2 respectively. In traditional designs, to make sure that both receivers get the two broadcast packets, the source node u needs to send packets p_1 and p_2 using two transmissions. With the help of network coding, node u broadcasts an XORed packet $p_1 \oplus p_2$ using one transmission, thus saving one transmissions. From the

examples in Figure 4.1, we see that the nature behind network coding is that instead of sending each target packet one by one, the forwarder encodes all of them together and broadcasts them with a coded packet using one transmission.



(a) Coding scenario (b) Non-coding scenario

Figure 4.2: Impact of link correlation on network coding.

4.3.2 Impact of Link Correlation on Network Coding

From the above discussion, we learn that network coding obtains benefits following two steps. First, a node finds the coding opportunity based on the reception of its neighbors. Second, the node encodes several packets together and broadcasts with one transmission. Correspondingly, the impact of link correlation on network coding comes from two aspects. On the one hand, the coding opportunity highly depends on the diversity of each link's receptions, i.e., link correlation. For example, in Figure 4.2(b), both receivers lose the same packet p_1, i.e., the link uv_1 and uv_2 are positive correlated. There are no coding opportunities in this scenario. Compared with the coding scenario in Figure 4.2(a), we find that we have more coding opportunities when the links are *lower* correlated.

On the other hand, network coding could be effective if and only if the encoded packet is broadcast and received by all receivers that are involved in the coded packet. We theoretically analyze the expected number of transmissions to cover all potential receivers and demonstrate that the source node needs *fewer* transmissions when the links are *higher* correlated. We use $p(e_i) \in (0, 1]$ to denote the probability that a source node can directly deliver a packet via link e_i. Let $p(e_1)$ and $p(e_2)$ denote the link qualities for the two receivers respectively. Corresponding packet loss probabilities are denoted as $p(\bar{e_1}) = 1 - p(e_1)$ and $p(\bar{e_2}) = 1 - p(e_2)$. Let $p(\bar{e_1} \cap \bar{e_2})$ denote the probability

Notation	Description
$e = \{u, v\}$	A link or transmission event from node u to v
$p(e)$	Link quality, measured by the transmission successful probability
K	The total number of packets (or nodes) involved in a coding operation
l	The number of packets in the output queue
$S_i(u)$	A subset of i nodes among u's neighbors with the highest link quality
$p(S(u))$	A joint probability of links from u to $S(u)$ nodes
$Pr(v\|S(u))$	Set link probability, a conditional probability
$\varepsilon(u)$	The expected transmission count for u to reliably broadcast one packet

Table 4.1: Notations used in this chapter

that a coded packet from the source node is not received by either receiver. Then the expected number of transmissions E[ε] to deliver the coded packet to the two receivers can be calculated as

$$\mathrm{E}[\varepsilon] = \sum_{i=1}^{2} \frac{1}{p(e_i)} - \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2})} \tag{4.1}$$

We note that $p(\bar{e_1} \cap \bar{e_2})$ obtains its maximum value when links e_1 and e_2 are perfectly positive correlated while it gets a minimum value when links e_1 and e_2 are perfectly negative correlated. Let us revisit the two scenarios in Figure 4.2(a) and 4.2(b). For the scenario in Figure 4.2(a), the expected number of transmissions E[ε] is 3 based on Eq.(3.1), given that $p(e_1) = p(e_2) = 0.5$ and $p(\bar{e_1} \cap \bar{e_2}) = 0$ since uv_1 and uv_2 are perfectly negative correlated. In Figure 4.2(b), however, since uv_1 and uv_2 are perfectly positive correlated, it is not difficult to get that $p(\bar{e_1} \cap \bar{e_2})$ is 0.5. As a result, E[ε] is 2 which is less than the cluster in Figure 4.2(a). This suggests that in the broadcast procedure, positive correlation is preferred since all the receivers lose the same packets and few retransmissions are needed. Therefore, there exists a trade-off between the number of coding opportunity and the broadcast efficiency. Only considering coding opportunity without taking link correlation into account may not effectively utilize the broadcast diversity, or even worse – it may lead to a higher transmission cost when an undesired coding operation is executed.

4.4 Main Design

From the previous section, we know that there exists a trade-off between broadcast efficiency and coding opportunity. In this section, we theoretically analyze the broadcast efficiency (Section 4.4.1) and coding opportunity (Section 4.4.2). Then, we propose unified metrics dealing with the trade-off between broadcast efficiency and coding opportunity in Section 4.4.3. Some notations used in this chapter are listed in Table 4.1.

4.4.1 Broadcast Efficiency Analysis

We first examine the number of transmissions for node u to reliably broadcast the coded packet to all the potential receivers such that they can extract the original packet from the coded one. We denote ε as the number of transmissions needed by sender u to reliably broadcast a coded packet involved with K original packets to the K potential receivers – $V(u) = \{v_1, v_2, \ldots, v_K\}$.

We assume a widely used ARQ model for the reliable delivery. In ARQ, if a sender does not receive an ACK before the timeout, it retransmits the packet until it receives an ACK. With ARQ, for each link e with link quality $p(e)$, the expected number of transmissions needed to successfully send a packet over a single link e is $\frac{1}{p(e)}$. Although link quality of wireless links changes over time, it can be measured and refreshed through normal data traffic or periodic beacons. Let the link quality between u and the potential receiver v_j be $p(e_j)$, $j = 1, 2, \ldots, K$. The corresponding packet loss probability is denoted by $p(\overline{e_j}) = 1 - p(e_j)$. Without loss of generality, we assume $p(e_1) \geq p(e_2) \geq p(e_3) \geq \ldots \geq p(e_K)$. The expectation of ε can be calculated as

$$\begin{aligned} \mathrm{E}[\varepsilon] = & \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{1 \leq i < j \leq K} \frac{1}{1 - p(\bar{e}_i \cap \bar{e}_j)} + \sum_{1 \leq i < j < l \leq K} \frac{1}{1 - p(\bar{e}_i \cap \bar{e}_j \cap \bar{e}_l)} + \ldots \\ & + (-1)^{K-1} \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2} \cap \ldots \cap \bar{e_K})}. \end{aligned} \tag{4.2}$$

Proof. Let $Pr(\varepsilon > t)$ be the probability that u needs more than t transmissions to

deliver a coded packet to K potential receivers, we have

$$\begin{aligned} Pr(\varepsilon > t) = & \sum_{i=1}^{K} p(\bar{e_i})^t - \sum_{1 \le i < j \le K} p(\bar{e_i} \cap \bar{e_j})^t + \sum_{1 \le i < j < l \le K} p(\bar{e_i} \cap \bar{e_j} \cap \bar{e_l})^t + \ldots \\ & + (-1)^{K-1} p(\bar{e_1} \cap \bar{e_2} \cap \ldots \cap \bar{e_K})^t. \end{aligned} \tag{4.3}$$

Taking the difference yields $Pr(\varepsilon(u) = t) = Pr(\varepsilon(u) > t-1) - Pr(\varepsilon(u) > t)$. Then the expected transmission count for u to reliable broadcast one coded packet can be calculated as

$$E[\varepsilon] = \sum_{t=1}^{+\infty} t \cdot Pr(\varepsilon = t) = \sum_{t=1}^{+\infty} t \cdot (Pr(\varepsilon > t-1) - Pr(\varepsilon > t)) = \sum_{t=0}^{+\infty} Pr(\varepsilon > t) \tag{4.4}$$

Substitute the right part of the above equation with Eq.(4.3), we get

$$\begin{aligned} \mathrm{E}[\varepsilon] = & \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{1 \le i < j \le K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j})} + \sum_{1 \le i < j < l \le K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j} \cap \bar{e_l})} + \ldots \\ & + (-1)^{K-1} \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2} \cap \ldots \cap \bar{e_K})}. \end{aligned} \tag{4.5}$$

□

To get ε with K potential receivers, we need to compute $\binom{K}{1} + \binom{K}{2} + \ldots + \binom{K}{K} = 2^K - 1$ polynomial terms where $\binom{a}{b}$ is the number of b-element combination of an a-set. In network coding, although the number of packets that can be encoded together is relatively small (and thus the number of the potential receivers, i.e., K is small), the exponential growth of complexity with K shall be avoided when possible. In the following, we present a novel approach to simplify the calculation.

Transmission Count Approximation

Due to the high cost of computing ε, we seek a more efficient algorithm to approximate ε with less computational complexity. In wireless networks, the nodes with a higher link quality usually receive the broadcast packet before (or at the same time) those with a lower link quality. To confirm this observation, we deploy 30 MICAz near a sender u to form a star topology. The source node keeps broadcasting packets in every 0.2s until all

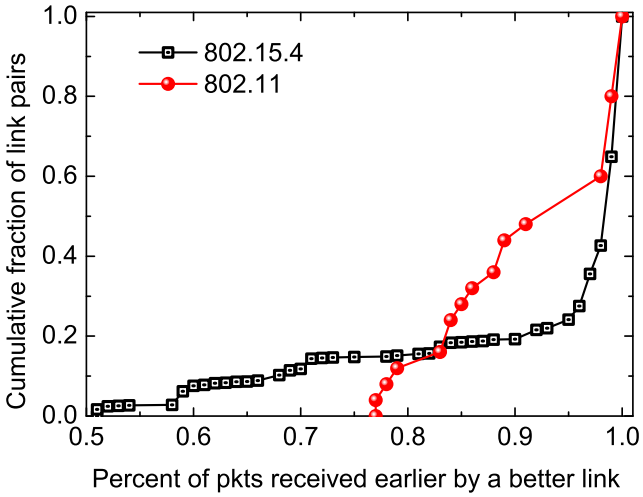


Figure 4.3: Statistics of receiving probability.

the receivers receive 100 packets. In each packet we include sequence number and time stamp. After collecting the packet reception trace, we compare the reception between each link pair (there are $\binom{30}{2} = 435$ such pairs). Figure 4.3 shows that the node with a better link from u receives more than 90% of the packets earlier (or at the same time) than the node with a worse link from u on both 802.15.4 and 802.11 testbed. In other words, statistically, the node with a better link needs fewer transmissions for a specified packet. Based on this observation, we propose an approximate algorithm to estimate ε, denoted as $\hat{\varepsilon}$.

Lemma 4.4.1. *(**Approximation of** ε) Assuming nodes with higher link quality receive the broadcast packet earlier than those with lower link quality, u's expected transmission count with K potential receivers is approximated by*

$$\hat{\varepsilon} = \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))} \tag{4.6}$$

where $S_i(u)$ is a subset of i nodes with the highest link quality among u's neighbors, and $p(S_i(u))$ is the probability that all the i nodes in $S_i(u)$ successfully receive a packet.

Proof. Based on the observation in Figure 4.3, we first estimate transmissions for the source node u to reliably send a packet to the node v_i with a better link. Then we

consider the transmissions of delivering a packet to the node v_j with a worse link under the situation that v_j fails to receive the packet when u sends it to v_i. Let $p(S_i(u))$ be the probability that all the i nodes in $S_i(u)$ successfully receive a packet. The approximation of ε is given by

$$\begin{aligned}\hat{\varepsilon} =& \frac{1}{p(e_1)} + \frac{\Pr(\bar{e_2}|e_1)}{p(e_2)} + ... + \frac{\Pr(\bar{e_K}|\bigcap_{i=1}^{K-1} e_i)}{p(e_K)} = \frac{1}{p(e_1)} + \frac{p(S_1(u)) - p(S_2(u))}{p(e_1)\cdot p(e_2)} + ... \\ &+ \frac{p(S_{K-1}(u)) - p(S_K(u))}{p(S_{K-1}(u))p(e_K)} = \sum\nolimits_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))}\end{aligned} \tag{4.7}$$

Thus we have

$$\hat{\varepsilon} = \sum\nolimits_{i=1}^{K} \frac{1}{p(e_i)} - \sum\nolimits_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))} \tag{4.8}$$

□

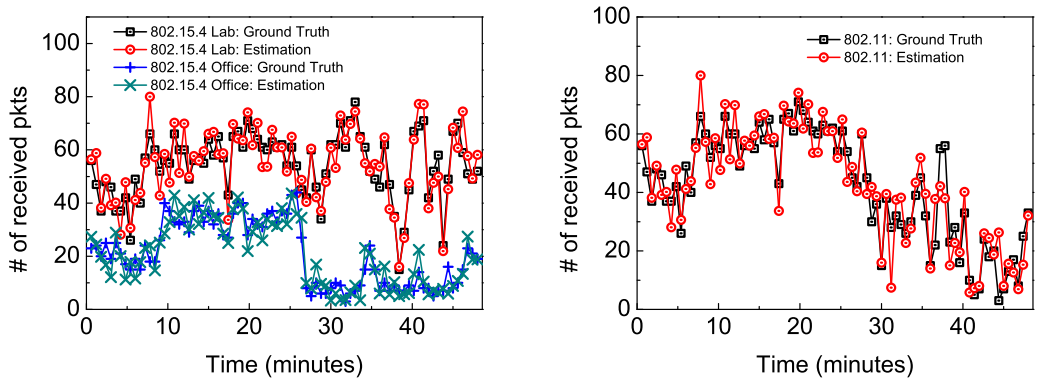
The computational complexity of $\hat{\varepsilon}$ is $O(K^2)$, where K is the number of receivers.

A Special Case: Note that Eq.(4.6) includes the special case that the links are independent. When links are all independent, we have $p(S_i(u)) = p(e_i) \cdot p(S_{i-1}(u))$. And $\hat{\varepsilon}(u)$ reduces to $\hat{\varepsilon} = \sum_{i=1}^{K} \frac{1}{p(e_i)} - K + 1$.

Estimation Overhead and Accuracy

In real-world environments, both link quality and link correlation change over time. A natural question is that how much overhead is required to maintain these values fresh under dynamic scenarios? To answer it, we conduct a set of experiments on both 802.11 and 802.15.4 testbeds in a period of 50 minutes.

The main overhead comes from two sources. The first comes from link status measurement, which is accomplished using reception report from normal traffic data. This part overhead is thus negligible. Second, we need to exchange packet reception reports (i.e., [1100]) among one-hop neighbors in order to calculate $\hat{\varepsilon}$. The exchange of reception report has already been required by network coding schemes, e.g., COPE [18]. The difference is that we use the reception report not only for the capture of coding opportunity but also for the calculation of link correlation. Besides, the binary report is small and much less frequently exchanged.



(a) 802.15.4 testbed (b) 802.11 testbed

Figure 4.4: Estimation Accuracy: Received pkts vs. Estimation

In the experiments, the source node keeps broadcasting packets to six receivers in every 0.3s. The reception report is sent in every 30s to fresh link quality and correlation values. We run the 802.15.4 experiments on a lab and an open office environment. The 802.11 experiments run on a university building.[1] The corresponding estimated number of received packets and the ground truth are shown in Figure 4.4. Here, the number of received packets (during a time period of 30s) is the number of sent packets (i.e., 100) over $\hat{\varepsilon}$. From the figure, we can find that the estimated values closely follows the ground truth. It indicates that $\hat{\varepsilon}$ is maintained accurately over time. In the above experiment settings (i.e., the reception report is sent in very 30s), the cost of exchanging reception reports occupies a tiny fraction (1%) of the total energy cost. Thus, the estimation of ε will not bring much overhead to the existing protocols.

4.4.2 Coding Opportunities Estimation

The coding opportunity in a sender u is crucially dependent on the packet reception patterns in its receivers. When node u broadcasts a coded packet to all its receivers, we need to make sure that all the receivers have already gathered enough packets to decode the original one. We specify the network coding rule as follows:

Definition 3. *(**Network Coding Rule**) Consider a sender u transmitting an encoded*

[1]The detail information of the testbed scenarios is described in Section 4.6.

packet $p' = \oplus(p_1, p_2, \ldots, p_K)$. *In order to decode* p', *each receiver should have already received* $K-1$ *packets among* p_i, $i = 1, 2, \ldots, K$.

Based on the definition of network coding rule, we estimate the benefit of network coding through the coding opportunity. The formal definition of coding opportunity is given by:

Definition 4. *(**Coding Opportunity**) For packets buffered in an output queue, if there exist a number of packets that satisfy the network coding rule and thus can be encoded together, we call this condition a coding opportunity.*

Let the set of nodes involved in node u's coding operation be $V(u) = \{v_1, v_2, \ldots v_K\}$, where $K = |V(u)|$. Assume the number of coding opportunities with i original packets involved in an encoded packet is $\phi(i)$, $2 \leq i \leq K$. Assume the number of packets in node u's output queue is l, with the help of network coding, the total number of packets that node u needs to transmit changes to α, which is given by

$$\alpha = l - \sum_{i=2}^{K} (i-1)\phi(i), \tag{4.9}$$

where $\sum_{i=2}^{K}(i-1)\phi(i)$ is the number of packets reduced by network coding. From Eq.(4.9), we find that the total number of packets can be greatly reduced when there are many coding opportunities.

4.4.3 Correlated Coding Metrics

In this section, we aim to optimize the transmission efficiency of network coding with the consideration of both broadcast efficiency and coding opportunity. We introduce broadcast correlated coding metric and unicast correlated coding metric, which can be used in broadcast and unicast protocols. First, we introduce the correlated coding metrics for broadcast which is defined as follows:

Definition 5. *(**Broadcast Correlated Coding Metric**) The broadcast correlated coding metric, denoted as BETX, is defined as the number of transmissions needed by sender u to reliably broadcast a packet (either original packet or coded packet) to all*

the packet's receivers – $V(u_i) = \{v_1, v_2, \ldots v_K\}$, divided by the number of the potential receivers, i.e.,

$$BETX = \frac{\alpha}{l} \cdot \frac{\hat{\varepsilon}}{K}. \tag{4.10}$$

The calculation of the correlated coding metric BETX involves two terms: (i) $\frac{\alpha}{l}$ is the percentage of packets left in the queue after network coding, and (ii) $\frac{\hat{\varepsilon}}{K}$ measures the broadcast efficiency. The first term $\frac{\alpha}{l}$ prefers low correlation which reduces the total number of packets need to send. The second term $\frac{\hat{\varepsilon}}{K}$ prefers high correlation which reduces the expected number of transmissions for each coded packet. To deal with the preference on link correlation, we use the product of these two terms which represents the expected transmission count for a successful packet delivery with network coding. The tradeoff between the broadcast efficiency and coding benefit is decided by the product value.

Definition 6. *(**Unicast Correlated Coding Metric**) Given a sender u_i and its potential receiver set $V(u_i) = \{v_1, v_2, \ldots v_K\}$. The unicast correlated coding metric UETX is the forwarding cost of the link $e(u_i, v_j)$, $j = 1, 2, \ldots K$, which is calculated as follows:*

$$UETX = K \times BETX_{V(u_i)} - (K-1) \times BETX_{V(u_i)-\{v_j\}}. \tag{4.11}$$

The items $K \times BETX_{V(u_i)}$ and $(K-1) \times BETX_{V(u_i)-\{v_j\}}$ in Eq.(4.11) represent the numbers of transmissions needed by sender u_i to reliably broadcast a packet to the receiver sets $V(u_i)$ and $V(u_i) - \{v_j\}$. And UETX, which is the difference of $K \times BETX_{V(u_i)}$ and $(K-1) \times BETX_{V(u_i)-\{v_j\}}$, thus represents the forwarding cost of the link $e(u_i, v_j)$. The calculation of UETX only involves $BETX$, which can be calculated using Eq.(4.10).

4.5 Applications

The correlated coding metric can help a wide range of routing protocols to efficiently exploit network coding benefits. Thus far, we have successfully implemented seven classic algorithms and integrated the correlated coding metric with them. The basic information of these algorithms is shown in Table 4.2.

Protocol Name	Category	Network Info.	Routing strategy
FMS [63]	Multicast	One-hop	Minimal cost
ST [27]	Broadcast	One-hop	Tree-based
FNC [33]	Broadcast	Local	Cluster-based
PDP [64]	Broadcast	Two-hop	Pruning-based
ZigBee [40]	Unicast	One-hop	Cluster tree
OLSR [39]	Unicast	Two-hop	Multi-point relay
ETX [65]	Unicast	One-hop	Minimal cost

Table 4.2: Seven state-of-the-art protocols supported by correlated coding

4.5.1 802.11 Networks

Flexible Multicast Service

In wireless LAN, flexible multicast service (FMS) is an efficient way to deliver the same contents to a large number of receivers. Notice that in FMS multiple APs may share the same upstream service provider. We thus propose a novel communication paradigm called *collaborative FMS*. As shown in Figure 4.5, multiple APs are connected via wires and form an infrastructure, while user devices, i.e., the laptops, are one (wireless) hop away from the infrastructure. The collaborative FMS design utilizes the infrastructure for sharing the packet through wires and *collaborates* to cover every user. With correlated coding, we can further improve the performance of collaborative FMS. In detail, we calculate UETX – the transmission cost for an AP to cover one receiver under the receiving status of the rest receivers. When one receiver can be covered by multiple APs, as shown in Figure 4.5, the receiver is assigned to the AP with minimal UETX.

4.5.2 802.15.4 Networks

Broadcast

We classify the existing deterministic broadcast algorithms into three categories: i.e., (i) tree-based [27], (ii) cluster-based [25], and (iii) pruning-based [64]. In the tree based broadcast [27], a minimal cost spanning tree is constructed in a distributed manner, and broadcast is performed based on the tree. The cluster-based approach [25] first finds a maximal independent set, and then connect the nodes in the set with connectors. The pruning-based broadcast path builds on the multi-point relays, where each relay's

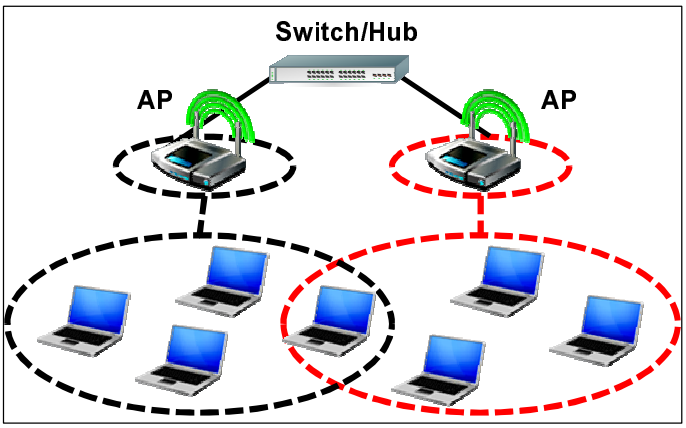


Figure 4.5: Collaborative FMS

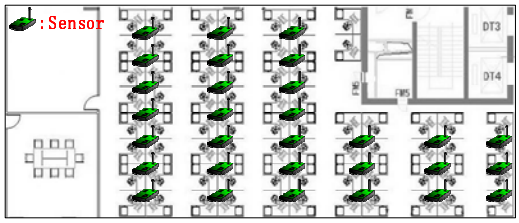
one-hop downstream forwarders can cover all its two-hop neighbors.

We briefly introduce how to integrate the correlated coding metric into these three types of algorithms, thus bring them transmission gain from link correlation and network coding. In a tree based algorithm [27], instead of finding the nodes with maximum leaves, we integrate correlated coding by choosing the nodes with min($BETX$) as the tree nodes. To combine the cluster based broadcast [25] with correlated coding, the algorithm first selects nodes with min($BETX$) to form a maximum independent set (MIS). Then, it finds connectors to link the nodes in MIS. In the pruning based scheme [64], each forwarder adds its one-hop neighbors with min($BETX$) to the forwarder set to cover its two-hop neighbors. In all three algorithms, if a covered node receives a message from different nodes in the tree, MIS or forwarder set, the node selects the node with min($UETX$) as its forwarder.

Unicast

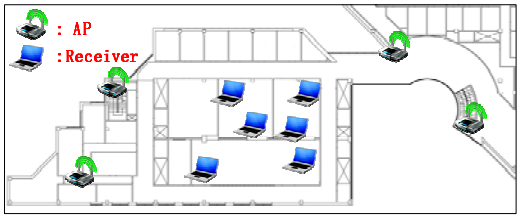
Unicast routing protocols can be divided into two categories – (i) backbone based, and (ii) flat protocols. In the first category, a backbone is built using the tree, cluster or pruning based method. For example, in ZigBee [40], a cluster tree is built. In OLSR [39], multi-point relays are selected as backbone nodes. In backbone based unicast routing protocols, if any node has a packet to transmit, it will first send the packet to its nearby backbone node. Then, the packet goes through the backbone to the destination. The

(a) 802.15.4 testbed: Lab



(b) 802.15.4 testbed: Open office

(c) 802.15.4 testbed: Outdoor



(d) 802.11 testbed: University building

Figure 4.6: Testbed environments

idea of integrating the correlated coding metric to this kind of routing protocols is similar to the deterministic broadcast protocols, i.e., we select the nodes with $\min(BETX)$ to form the cluster tree or multi-point relay set. For the flat protocols, a classic example is ETX [65], which selects a routing path with $\min(\sum \frac{1}{p(e)})$. With correlated coding, we select the path with $\min(\sum UETX)$.

4.6 Testbed Implementation

The performance of network coding changes dramatically since the link status as well as the packet reception pattern in different environments varies significantly. In this section, we report the experiment results of seven state-of-the-art protocols supported by correlated coding metrics on one 802.11 platform located in a university department building, and three 802.15.4 platforms in a lab, an open office, and an outdoor environment. The experiment scenarios are shown in Figure 4.6, and the testbed settings and topology properties are shown in Table 4.3.

Platform	Location	Environment	Physical Size
MICAz	UMN	Lab	$8m\times$ 2.5m
TelosB	SIAT	Open office	$18m\times$ 13m
GreenOrbs	TRIMPS	Outdoor	$15m \times 5m$
802.11g	UMN	University building	$73m\times$ 30m
No. of Nodes	Degree	Channel	Power
30	$7 \sim 23$	Ch16	-25dBm
30	$6 \sim 21$	Ch16, Ch26	-25dBm
20	$4 \sim 13$	Ch16	-25, -19.2dBm
6	6	Ch3, Ch6	15dBm, 20dBm

Table 4.3: Testbed settings and topology properties

4.6.1 Experiment Setup

802.15.4 testbed

We deploy three 802.15.4 testbeds. The first one is located in a lab environment where 30 MICAz nodes randomly on an $8m \times 2.5m$ wall, see Figure 4.6(a). The second testbed has 30 TelosB nodes which are deployed in an $18m \times 13m$ open office environment, as shown in Figure 4.6(b). On the third testbed, 20 GreenOrbs nodes are deployed on an open space along a river, as shown in Figure 4.6(c). On all the three testbeds, the default power is -25dBm and the default channel is 26.

In the beginning of the experiment, a control node is used to remotely configure radio parameters, i.e., transmission power and channel. Based on these radio settings, each node broadcasts 10^5 packets in turn. Each packet is identified by a sequence number. All the received packets are recorded in nodes' flash memory. When all the nodes finish broadcasting 10^5 packets, they send their packet reception information to a sink node which is connected to a PC. We thus obtain the information required by correlated coding, i.e., the packet receiving patterns, based on which we can calculate the broadcast or unicast correlated coding metric. The reception report length for the metric

calculation is 100. Then, the corresponding nodes on the testbed are selected as forwarders for unicast or broadcast according to the application description in Section 4.5. In the broadcast application, the forwarders keep on broadcasting packets until all their covered nodes receive the packets. In the unicast application, two pairs of data flows are picked up and the unicast sessions terminate when each source node reliably sends its packets to the destination.

802.11 testbed

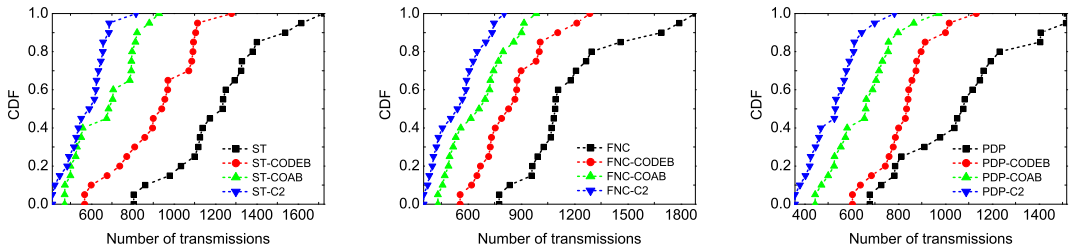
This testbed is located on the 4th floor of the Computer Science Department building in University of Minnesota, as shown in Figure 4.6(d). From the figure, we can see that four APs are deployed at the four corners of the floor, while six receivers are placed in three different rooms, separated by concrete walls. The AP in the testbed is a PC equipped with an Intel Core2 Duo T5470 processor, 2GB RAM, and an 802.11 wireless card with the Realtek RTL8101E chipset. The wireless card transmits at a default power of 20dBm and a default channel of 6. The standard we used is 802.11g. we use the `Lorcon2` packet injection library [66] to generate the traffic. During the experiment, four APs broadcast 10^5 packets in turn and the receivers record the packet reception information. Similar to the experiments on the 802.15.4 testbeds, we obtain the link correlation information for correlated coding.

4.6.2 Compared Schemes and Performance Metrics

We compare the correlated coding design with the seven state-of-the-art protocols as well as their enhanced versions which are integrated with network coding, e.g., CODEB [19], and COAB [58]. Among them, CODEB applies network coding over a wireless backbone built with a pruning method and COAB is a coding aware routing protocol. For those unnamed network coding protocols, we label them with "-NC". For example, ZigBee-NC means the enhanced version of ZigBee with network coding. the design is labelled with "-C2", the abbreviation of correlated coding. We use two metrics for the following performance evaluation.

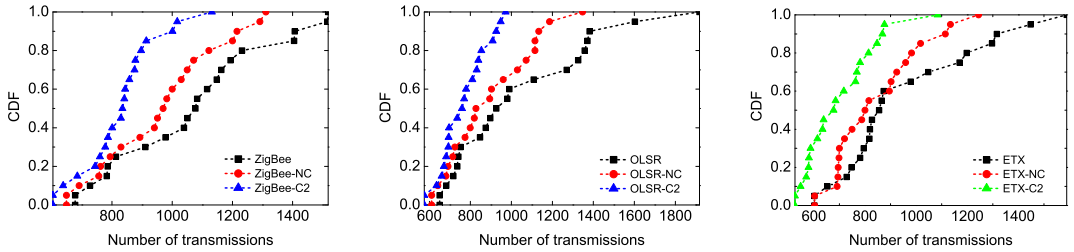
•**Number of transmissions** – the number of transmissions needed by a scheme to reliably send 100 packets to the receivers.

•**Number of coding operations** – the number of times that network coding occurs when 100 packets are reliably delivered.



(a) Spanning Tree (b) Forwarding Node Cluster (c) Partial Dominating Pruning

Figure 4.7: Main performance results: broadcast protocols in 802.15.4



(a) ZigBee (b) OLSR (c) ETX

Figure 4.8: Main performance results: unicast protocols in 802.15.4

4.6.3 Main Performance Results

The experimental results of the seven protocols are shown in Figures 4.7, 4.8, 4.9 and 4.10. Figure 4.7 plots the CDF of the transmissions with different broadcast protocols using correlated coding. As it shows, the correlated coding design significantly improves the transmission efficiency under the three different broadcast strategies. For example, for the tree based broadcast algorithm – Spanning Tree (ST), the nodes need 1238 transmissions, on average, to guarantee that all nodes in the network receive 100 packets, while the number is 528 when correlated coding is combined with ST (i.e., ST-C2), achieving a reduction of 57%. The average number of transmissions with ST-CODEB and ST-COAB is 940 and 675, respectively. On average, the design reduces

transmissions by 44% and 22%. In Figure 4.8, we find that the correlated coding design improves the performance of the unicast protocols (i.e., Zigbee, OLSR, and ETX), and their corresponding coding aware designs significantly. The average performance gain is about 35% and 16% separately. We note that the benefits of correlated coding in unicast applications are less than that in broadcast applications since the coding opportunity in unicast is less than that in broadcast. Similar results are also found on the 802.11 testbed. In Figure 4.9, compared with FMS, FMS-NC, and collaborative FMS, the correlated coding design, i.e, FMS-C2, saves 45%, 34% and 26% transmissions respectively.

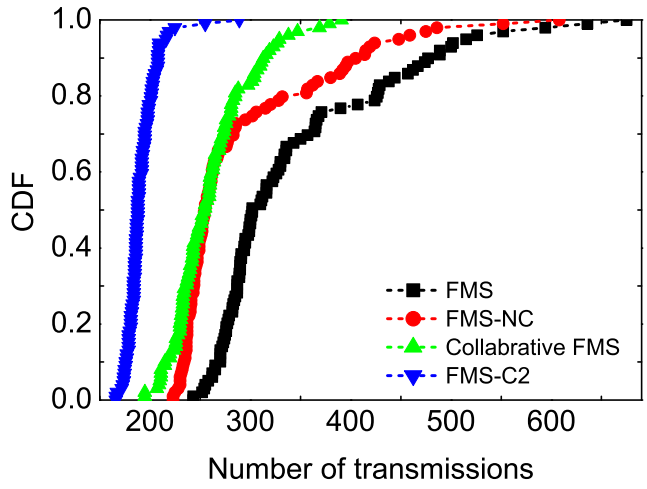


Figure 4.9: Main performance results: FMS in 802.11g.

From the experimental results in unicast, broadcast, and multicast on both 802.11 and 802.15.4 testbed, we can see that the correlated coding design outperforms traditional network coding protocols. the design achieves better performance than those protocols because we introduce the link correlation model to network coding. Compared with the traditional link independent model, the link correlation model has better performance on estimating link statuses. Furthermore, with the link correlation model, the correlated coding design quantifies the benefits of broadcast efficiency and coding opportunity and helps those protocols fully exploit network coding benefits while avoiding unnecessary coding operations.

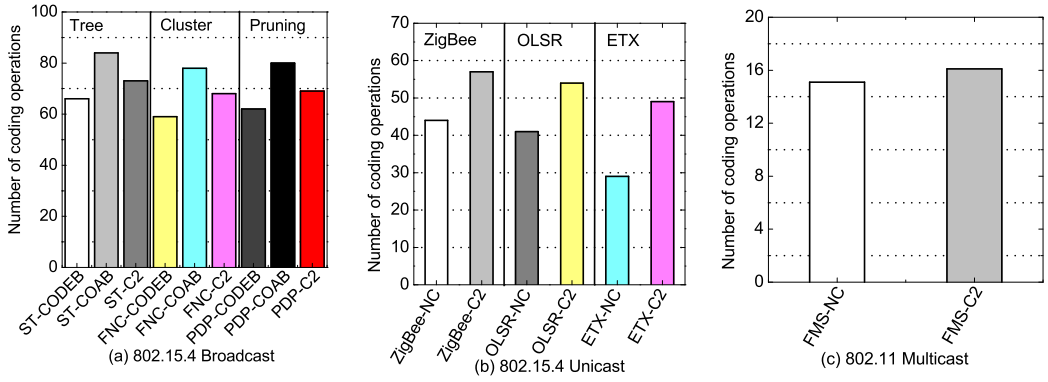


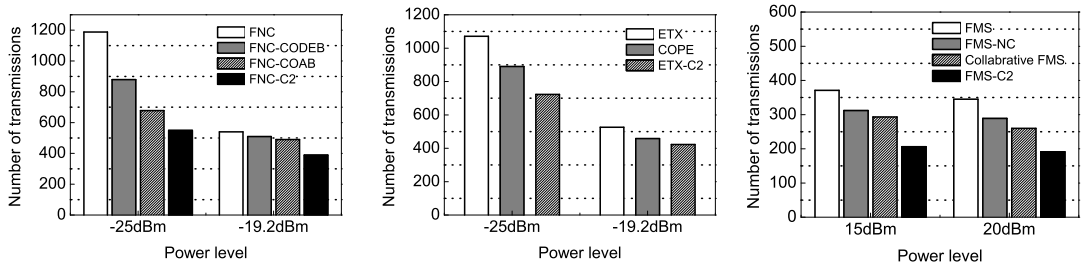
Figure 4.10: Main performance results in number of coding operations

Figure 4.10 plots the number of coding operations in broadcast, unicast and FMS applications. Take the broadcast application as an example (Figure 4.10(a)), we find that the coding aware routing COAB exploits the most coding opportunities while CODEB and COPE can not fully exploit the coding opportunities. Although the coding operation of correlated coding, i.e., C2, is less than COAB, we find that the performance of C2 is better than COAB. This is because correlated coding only encodes a packet when it can optimize the transmission gain, and thus avoid those unnecessary coding operations.

Although we collect results for all seven protocols, space constraints do not allow presenting all of them here. Therefore, we choose one representative algorithm for each application, i.e., Forwarder Node Cluster [33] (FNC for short) for broadcast application, ETX [13] for unicast application, and FMS for the Wi-Fi multicast application. For the rest of the 802.15.4 experiments, we assign correlated coding upon FNC and ETX, and compare them with COPE, CODEB, and COAB. For the 802.11 experiments, we compare correlated coding with FMS, FMS-NC, and collaborative FMS.

4.6.4 Impact of Power Level

• **802.15.4 testbed:** The power level for transmission is set from -25dBm to -19.2dBm to form a multi-hop network. Figure 4.11(a) shows the transmissions of FNC with CODEB, COAB and C2, and Figure 4.11(b) shows the transmissions of ETX with COPE and C2 under different power levels. we find that correlated coding greatly

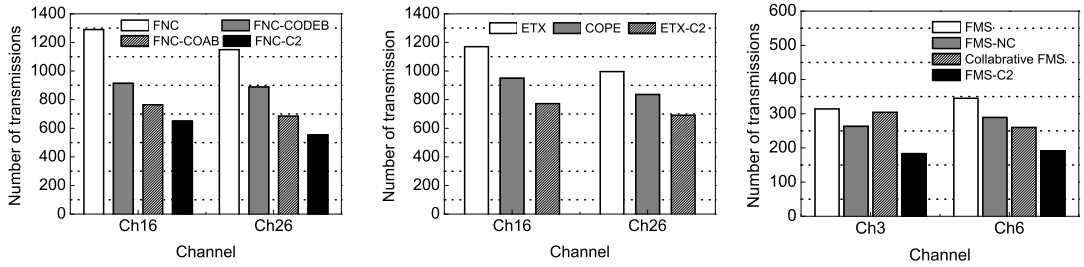


(a) 802.15.4 Broadcast (b) 802.15.4 Unicast (c) 802.11 Multicast

Figure 4.11: Impact of power levels

reduces transmissions for both broadcast and unicast applications. Under power level -25dBm, the transmission count for FNC is 1188, while it is 549 with correlated coding, providing a reduction of 54%. Under power level -19.2dBm, fewer transmissions are needed because a higher power level leads to better link quality. In this case, correlated coding still reduces transmissions by 28%.

• **802.11 testbed:** We examine the performance of correlated coding under power level 15dBm and 20dBm. From Figure 4.11(c), the number of transmission of C2 under 15dBm is 205 while it's slightly lower, i.e, 192 under 20dbm. In both cases, C2 saves collaborative FMS about 30% transmissions.



(a) 802.15.4 Broadcast (b) 802.15.4 Unicast (c) 802.11 Multicast

Figure 4.12: Impact of channels

4.6.5 Impact of Different Channels

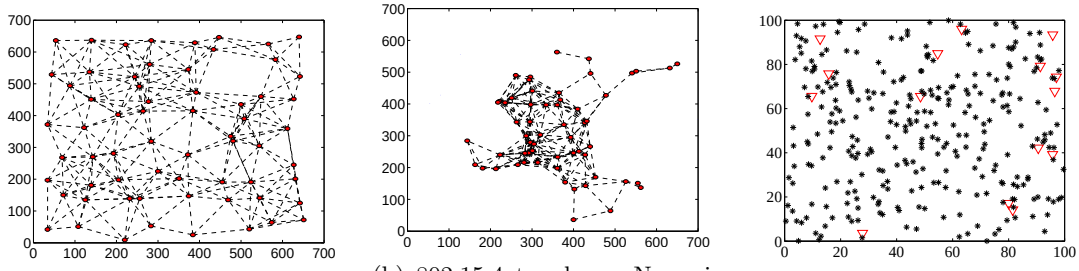
• **802.15.4 testbed:** In this experiment, we explore the impact of channels on correlated coding. We use two different channels – channels 16 and 26. Note that channel

16 overlaps with a co-habiting access point's 802.11 channel and that channel 26 is free of Wi-Fi interference. The power level for transmission is set to -25dBm. Figure 4.12 shows the energy consumption in broadcast and unicast protocols under different channels. The gains of correlated coding under the broadcast and unicast application are 52% and 31% under channel 26, and they are 50% and 34% using channel 16. In addition, we find that on both unicast and broadcast applications, all the algorithms need more transmissions to finish the same task in channel 16. This is because the interference introduced by the overlapped channel causes more packet losses.

• **802.11 testbed:** In this experiment, we examine the performance of C2 under channel 3 and 6. The transmission difference between these two channels is not as obvious as the observation on the 802.15.4 testbeds. That's because both channel 3 and 6 will be impacted by the nearby APs which usually use channel 1, 6 and 11.

4.7 Simulation

In this section, we provide extensive simulation results about the performance of correlated coding for large-scale networks under different system settings.



(a) 802.15.4 topology: Uniform

(b) 802.15.4 topology: Nonuniform

(c) 802.11 topology

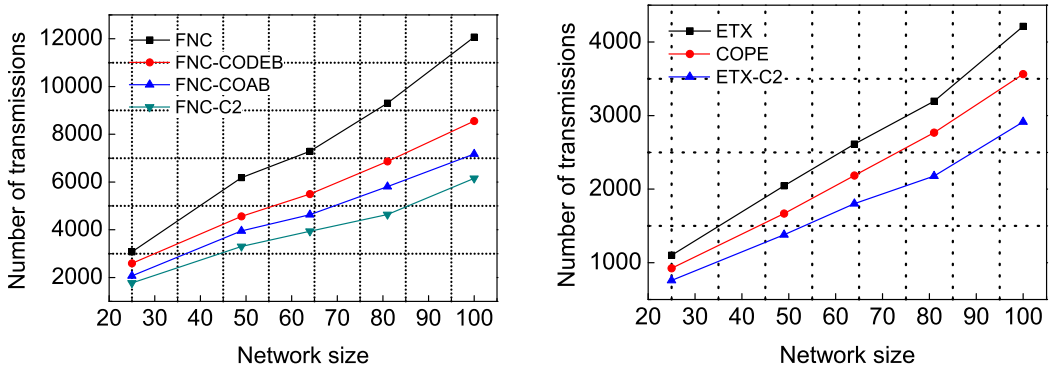
Figure 4.13: Simulation topologies for 802.15.4 and 802.11 networks

4.7.1 Simulation Setup

Given a scenario, we generate correlated reception reports for all the sender-receiver pairs by modifying the sampling algorithm for Bernoulli random variables in [53]. For a particular packet, the reception status at receivers could be either 0 or 1. we assume that the reception reports at different nodes are of the same length.

• **802.15.4 experiment:** we generate network topologies with different network sizes and densities. By default the network size is 64, and the field size is $700m \times 700m$ with a communication range of 160m. we conduct the experiments on both uniform and non-uniform scenarios, as shown in Figure 4.13(a) and 4.13(b). In the broadcast application, a random selected source node broadcasts 100 packets, and we record the number of transmissions required to finish broadcasting the 100 packets. In the unicast application, similar to the testbed experiment, we randomly pick up two pairs of data flows. The source nodes keep sending packets until the receivers successfully get 100 packets. The experimental results of each scenario are the average values of 100 rounds over different reception reports (i.e., different link correlations).

• **802.11 experiment:** we generate network topologies with varied number of APs and receivers. All the receiver can be connected to arbitrary AP with link quality varies from 0.2 to 1. And the average link quality is 0.6. The default number of AP and receiver is 15 and 300 separately, as shown in Figure 4.13(c).



(a) Broadcast scenario (b) Unicast scenario

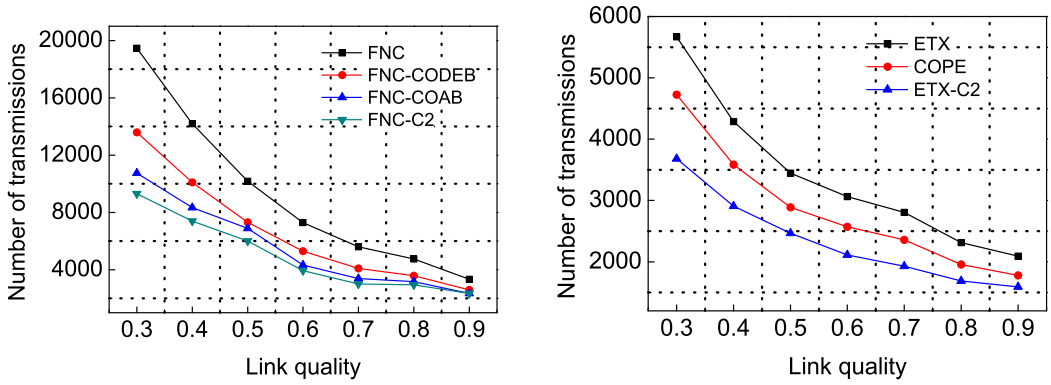
Figure 4.14: Impact of network sizes

4.7.2 Simulation Results on 802.15.4 networks

Impact of Network Size

Figure 4.14 shows the performance comparison of the correlated coding schemes (i.e., FNC-C2, and ETX-C2) and other coding schemes (i.e., FNC-CODEB, FNC-COAB, and

COPE) with network size ranging from 25 to 100. The left sub-figure in Figures 4.14 shows the results of FNC, FNC-CODEB, FNC-COAB and FNC-C2 respectively. From this figure, we can find that the average transmission count of the design is 3940, while those of FNC, FNC-CODEB, and FNC-COAB are 7585, 5612 and 4746 respectively. the design saves 47% of transmissions compared to FNC without using network coding. Compared with FNC-CODEB and FNC-COAB, correlated coding saves about 30% and 20% of transmissions because correlated coding better exploits the necessary coding opportunities. In the unicast application in the right sub-figure in Figures 4.14, compared with ETX and COPE, the transmission gain of correlated coding is 31% and 18%. From Figures 4.14, we can also see that the trends of transmission gain with increasing network size in both unicast and broadcast application are quite stable, suggesting that the design scales well with large networks.



(a) Broadcast scenario

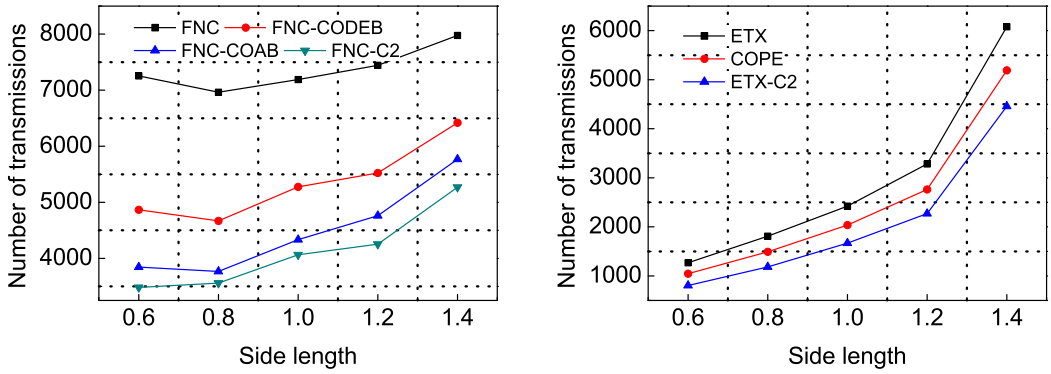
(b) Unicast scenario

Figure 4.15: Impact of link quality

Impact of Link Quality

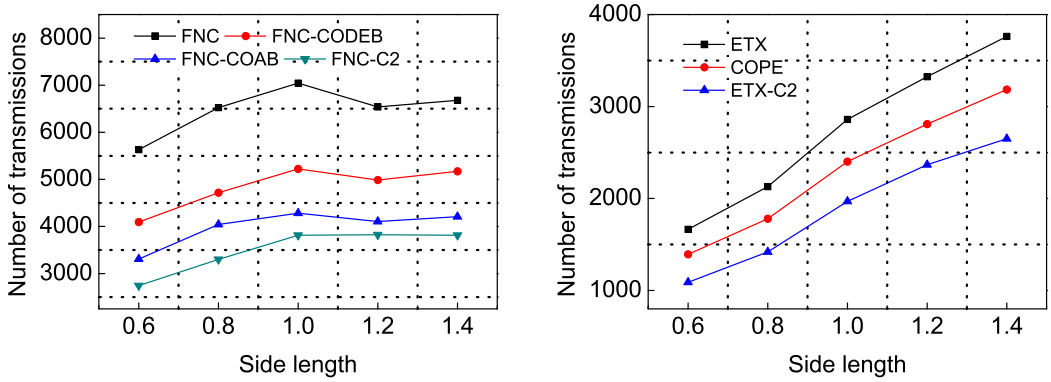
Let us consider the transmission gain of correlated coding for networks with different link qualities. The results are shown in Figure 4.15. From the left sub-figure in Figure 4.15, we can see that the broadcast transmission count of the design varies from 9302 to 2340 when the link quality varies from 0.3 to 0.9. Compared with FNC, the energy gain of FNC-C2 decreases from 52% to 29% when the link quality increases. A similar result is observed in the unicast application in the right sub-figure in Figure 4.15, where the

transmission gain of ETX-C2 upon ETX decreases from 35% to 23%. The reason is that with higher link quality, the transmission count of a forwarder to send a packet to its destinations is already small, leaving only marginal room for the algorithm to improve the energy gain.



(a) Broadcast scenario (b) Unicast scenario

Figure 4.16: Impact of network density (uniform)



(a) Broadcast scenario (b) Unicast scenario

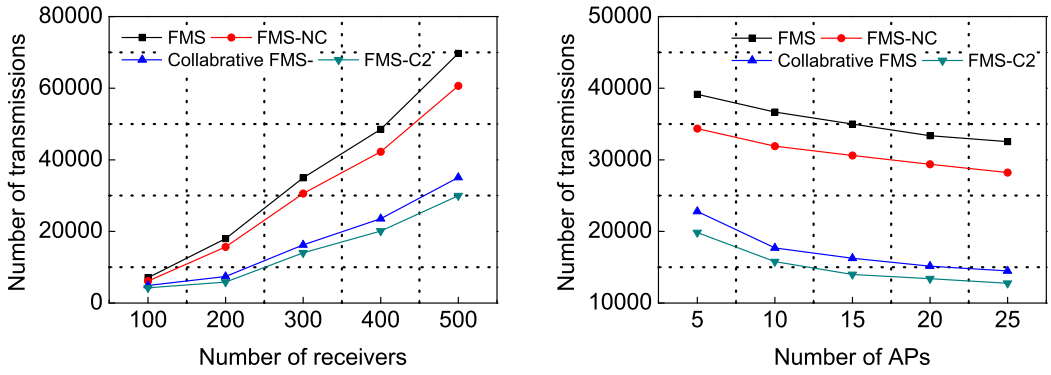
Figure 4.17: Impact of network density (non-uniform)

Impact of Network Density

we consider both uniform (Figure 4.16) and non-uniform (Figure 4.17) node distributions. Figures 4.16 shows the number of transmissions of the four broadcast protocols

and three unicast protocols for uniform networks, under different network densities. The average node degrees for side length (of the simulated square sensing field) 0.6, 0.8, 1, 1.2, 1.4 are 20.2, 13.0, 8.4, 5.9, and 3.9, respectively. From the left sub-figures in Figures 4.16 and 4.17, we can see that with variation in density, the number of broadcast transmissions does not change monotonically. With the increase of network density, on the one hand, a forwarder has more receivers and needs more transmissions to cover them. One the other hand, the number of forwarders decreases in a fixed size network.

In Figures 4.16 and 4.17, the transmission gain of C2 decreases as the side length increases (and thus the density decreases). For example, in the uniform network scenario in Figure 4.16, the broadcast transmission gain of FNC-C2 over FNC is 52% at node degree 20.2, and it drops to 34% when the average degree is only 3.9. Similarly, the unicast transmission gain of ETX-C2 upon ETX decreases from 37% to 26%. we also find a gain drop in the non-uniform network topology in Figures 4.17. This is because as the network becomes denser, a node tends to have more one-hop candidates and thus it overhears more packets, which increases the possibility of finding more coding opportunities. This explains the increasing energy gain when node density grows.



(a) Varied No. of Receivers

(b) Varied No. of APs

Figure 4.18: Experiments on large scale Wi-Fi networks

4.7.3 Simulation Results on 802.11 networks

The left sub-figure in Figure 4.18 shows the performance of C2 with the number of receivers increasing from 100 to 500. Correspondingly, the transmission of FMS (to

reliably broadcast 100 packets) increases from 7079 to 69691 while that of C2 increases from 4229 to 29988. In the right sub-figure of Figure 4.18, with the increase of number of APs from 5 to 25, the transmission of FMS decreases from 39152 to 32536 while that of C2 decreases from 19835 to 12760. With either increased number of receivers or increased number of APs, C2 has more chances to assign the "black sheep" who cause the APs massive retransmissions to the most suitable APs, and thus saving the number of transmissions.

4.8 Summary

In this chapter we study the impact of link correlation on network coding. we find that link correlation can help us decide whether a network coding operation is needed or not. we introduce correlated coding which optimizes the transmission efficiency of network coding. the design can be applied in both broadcast and unicast protocols. we integrate correlated coding with seven state-of-the-art routing protocols, and evaluate the design with testbed experiments and extensive simulations. The results confirm the effectiveness of the design compared with the existing network coding protocols under a wide range of system settings.

Chapter 5

Link Correlation Aware Opportunistic Routing

5.1 Introduction

Opportunistic Routing (OR), originally proposed by S. Biswas et al. in [67], has great potential to improve the network performance. The basic idea of opportunistic routing is fairly straightforward. Given a source and a destination in a multi-hop wireless network, instead of preselecting a single specific node to be the next-hop forwarder, a set of candidate forwarders are selected to deliver the packets. Taking advantage of the reception diversity in the candidate forwarder set, opportunistic routing defers the selection of the next hop for a packet until it acquires knowledge about the set of candidate forwarders that have received that packet.

Since the strength of opportunistic routing comes from the packet reception diversity of the candidate forwarder set, the candidate selection becomes one of the key issues in opportunistic routing. The selection of different candidates has a high effect on the performance of opportunistic routing. Extensive candidate selection algorithms based on the key metric, the expected number of transmissions, have been proposed in the literature [67, 68, 69, 70, 71, 72, 73]. In these studies, the researchers explicitly or implicitly assume that the wireless links are independent when they estimate the transmission cost from a upstream node to the next-hop candidate forwarder set.

Recent studies [74, 75, 76], however, provide clear evidence that wireless links are not

independent because of cross-technology interference and correlated shadowing. Cross-technology interference, which is caused by the external signal in the unlicensed shared spectrum, can lead to correlated packet losses since the high-power interferer's signal may corrupt nearby low-power links simultaneously. On the other hand, correlated shadowing, a channel propagation phenomenon that nearby links are affected by the same shadower, may also introduce correlated packet losses to wireless networks.

The finding of the link correlation phenomenon has significant impacts on network protocols that utilize concurrent wireless links, which include but are not limited to (i) traditional network protocols such as broadcast [12], multi-cast [15], and multi-path routing [16], or (ii) diversity-based protocols such as opportunistic routing [67, 68, 69], network coding [77], and hybrid routing [78]. Ignoring this phenomenon may cause serious estimation errors in modeling, which further leads to underutilized benefits or extra costs.

For example, when the packet loss patterns of the candidate forwarders are highly positive correlated (which means that they lose the same packets), the performance of opportunistic routing is the same as the traditional shortest path protocol (which uses the node with the best link among the candidates), since there are no diversity benefits to exploit from the candidate forwarder set. In this example, ignoring link correlation brings opportunistic routing no benefits but extra candidate set schedule costs.

Little research has been conducted to exploit link correlation to improve the performance of network protocols [74, 76]. In this chapter, we introduce link correlation to improve opportunistic routing's performance by optimizing the forwarder set selection and avoiding duplicate forwarding. Under link correlation, the forwarder set selection algorithm prioritizes low correlated nodes to increase the level of diversity while ensuring that neighboring nodes are close enough to each other such that the forwarded packets would be heard and duplicates are avoided.

In summary, the contributions are as follows:

• we reveal the impact of link correlation upon opportunistic routing. A novel link correlation aware metric is proposed to capture the expected number of any-path transmissions.

• With the link correlation aware metric, we propose a new candidate forwarder selection algorithm to help opportunistic routing fully exploit the diversity benefit of the wireless

broadcast medium.

• we evaluate the work extensively with testbed implementations and simulations. The experiment results identify the limitation of the traditional opportunistic routing under the appearance of link correlation. With the link correlation aware design, the number of transmissions of opportunistic routing is significantly reduced.

The remainder of this chapter is structured as follows. Section 5.2 reviews the related work. Section 5.3 presents the motivation. Section 5.4 introduces the metric, followed by its implementation in Section 5.5. Experiment results from the testbed and simulation are shown in Sections 5.6 and Sections 5.7. Finally, Section 5.8 concludes the chapter.

5.2 State of the Art

we begin with a brief survey of prior work on opportunistic routing and link correlation.

• **Opportunistic Routing:** The majority of previous studies in opportunistic routing are devoted to candidate selection [67, 70, 71, 72, 73], reception acknowledgement [79], forwarder coordination [80, 81], and rate control [82]. In this work we focus on the fundamental issue – the candidate selection in opportunistic routing. ExOR [67], the primary opportunistic routing protocol, uses the single path ETX to select candidates, where ETX is the average number of transmissions required to send a packet through a link. The ETX value of a single path is the sum of the ETX for each link in that path. Using the single path ETX as a metric for candidate selection is an approximation since it cannot capture the opportunistic paths. To account for the multiple paths that could be used by the candidates, expected any-path transmission (EAX) [83] is used in [70, 71, 72, 73] to capture a more accurate expected number of transmissions.

For example, H. Dubois et.al. propose least-cost opportunistic routing (LCOR) [72] which takes EAX as the metric to select the candidate sets. Similar to the well-known Bellman-Ford algorithm, LCOR exhaustively searches all possible candidates sets to find the paths with minimum transmissions. Its computational cost increases dramatically in dense networks because of the exponentially explosion in the exhaustive search. In minimum transmission selection (MTS) [71], the authors compute the transmission cost with EAX from the destination back to the source, using a dynamic programming

formulation analogous to the Dijkstra's algorithm. The authors in [73] investigate the candidate selection with identical maximum candidate set sizes. The result shows that if the maximum number of candidates is not limited, different opportunistic routing algorithms have almost the same performance. They prove that this assumption is not realistic since a large number of candidates may introduce large schedule overheads and duplicate transmissions.

Our work is different from the previous opportunistic routing schemes which implicitly or explicitly assume that packet receptions cross multiple receivers are independent when they exploit the diversity benefit of the wireless broadcast medium. we propose a link correlation aware opportunistic routing scheme to fully exploit the potential diversity benefit.

5.3 Motivation

In this section we first report the existence of link correlation. we then introduce the opportunistic routing framework and explain how wireless diversity serves opportunistic routing. Finally, we illustrate the impact of link correlation on the diversity benefit of opportunistic routing.

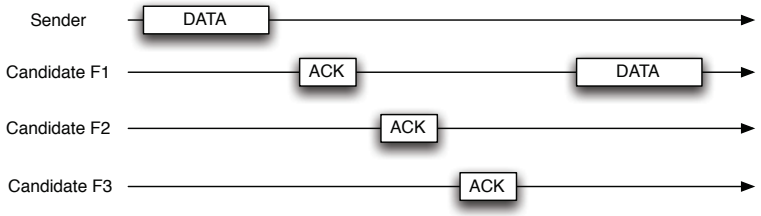


Figure 5.1: An example showing the opportunistic routing operation.

5.3.1 Opportunistic Forwarding Framework

The opportunistic packet forwarding process is shown in Figure 5.1. The sender selects a subset of nodes as candidate next-hops and assigns a priority to each of them. When the sender transmits a packet, it includes the ordered candidate forwarder set in its headers. Each candidate that receives the packet responds with an ACK. To avoid

the feedback implosion, candidates defer their ACKs according to their priorities in a TDMA-like approach. Since the candidates are likely to hear each other's ACK, they should include in their ACKs a list of higher priority candidates. In opportunistic routing, only the candidate with the highest priority forwards the packet to the next hop. Other candidates refrain from forwarding the packet as long as they overhear a higher priority ACK. Duplicate forwarding by more than one candidate could happen if a lower priority candidate cannot hear an ACK from a higher priority candidate. The whole forwarding process is initiated again by the sender as long as it does not receive an ACK from any candidates.

5.3.2 Diversity Benefits in opportunistic routing

Without considering link correlation, the previously opportunistic routing schemes do not fully capture the diversity benefit of the wireless broadcast medium. The following section will demonstrate the impact of diversity on opportunistic routing.

• **Diversity with Link Independence Model:** The strength of opportunistic routing comes from the diversity of the candidates' packet receptions. In opportunistic routing, when one candidate fails in receiving a packet, the other candidates may receive the packet. In the example in Figure 5.2, if node f_1 fails to receive a packet, candidate f_2 may receive the packet. Similarly if f_2 loses the packet as well, f_3 may probably receive it. In other words, the probability that all candidates lose the packet becomes quite low because of the packet reception diversity among multiple candidates.

The optimistic view of diversity comes from the assumption that the packet receptions cross multiple wireless links are independent. Under such an assumption, the packet loss in a receiver has no relationship with the packet losses in other receivers.

Let s be the source and d be the destination. Suppose $F_{s,d}$ is the set of candidate next-hop forwarders from s to d, and f_i is the candidate with priority i (with 1 being the highest priority). Assume that the packet delivery probability from s to f_i is p_{s,f_i}, and the ACK delivery probability from f_i to s is $p_{f_i,s}$.

we now mathematically demonstrate the impact of diversity upon opportunistic routing with the example in Figure 5.2. In the example, the size of the candidate set is set to be two, i.e., no more than two nodes are allowed to be candidate forwarders. Let us start by selecting the first forwarder. The expected number of transmissions

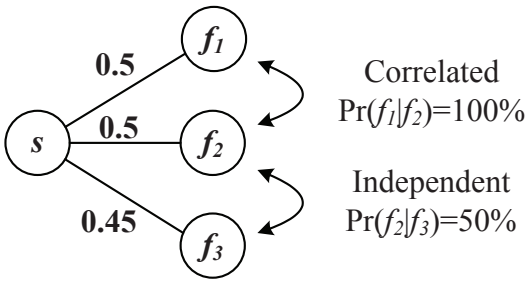


Figure 5.2: An opportunistic routing example with two candidates.

required for candidate f_1 or f_2 to successfully receive a packet from s is $1/p_{s,f_1} = 1/p_{s,f_2} = 1/0.5 = 2$. Similarly, the expected number of transmissions required by node f_3 is $1/p_{s,f_3} = 1/0.45 = 2.22$. Obviously selecting node f_1 or f_2 would be the optimal choice. Now consider the case that a second node is added to the candidate set. The expected number of transmissions for receiving one packet from s to at least one of the two candidates is given by

$$E(s, F_{s,d}) = \frac{1}{1 - \prod_i^2 (1 - p_{s,f_i})}. \tag{5.1}$$

In the example, $E(s, F_{s,d})$ with candidate set $\{f_1, f_2\}$ equals 1.33. Similarly, $E(s, F_{s,d})$ with candidate set $\{f_1, f_3\}$ or $\{f_2, f_3\}$ is 1.38. As a result, under link independence model, selecting nodes f_1 and f_2 as the forwarders will be the best choice in reducing the transmission cost.

• **Diversity with Link Correlation Model:** If links are not independent, the expected number of transmissions for delivering one packet to at least one of the two candidates is given by

$$E(s, F_{s,d}) = \frac{1}{1 - Pr(\overline{E_{s,f_1}}, \overline{E_{s,f_2}})}, \tag{5.2}$$

where $\overline{E_{s,f_i}}$ is the event that a transmission from source s is lost at forwarder f_i. In the example in Figure 5.2, the links from s to f_1 and f_2 are 100% correlated. $Pr(\overline{E_{s,f_1}}, \overline{E_{s,f_2}})$ equals 0.5. Thus, $E(s, F_{s,d})$ with candidate set $\{f_1, f_2\}$ turns out to be 2, which is greater than $E(s, F_{s,d})$ with candidate set $\{f_2, f_3\}$ (i.e., 1.38). As a result, selecting nodes f_2 and f_3 as candidates is the best choice under link correlation model. From this example, we find that the link independence assumption overestimates the real diversity of wireless

links. In the following section, we further analyze the opportunistic routing framework under the impact of link correlation.

5.4 Link Correlation Metric

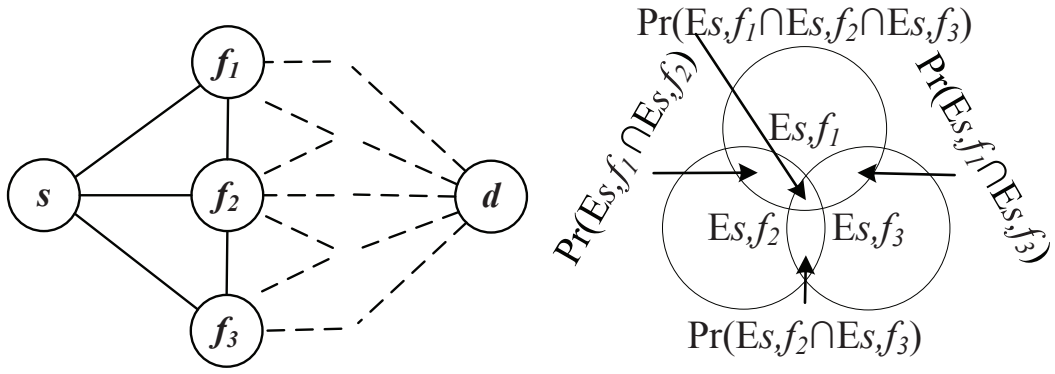
In this section we analyze opportunistic routing under the existence of link correlation. we explore the impact of link correlation on the candidate set selection process and reveal how link correlation awareness improves the performance of opportunistic routing. Finally we propose a novel link correlation aware opportunistic routing.

we define the expected number of any-path transmissions needed for reliably delivering a packet from the source s to the destination d, given the candidate set $F = \{f_1, f_2, \ldots, f_n\}$ with the link correlation awareness, as $E(s, F, d)$. In the design, the computation of $E(s, F, d)$ is recursive and is executed at individual nodes independently. At the receiver d, obviously, $E(s, F, d)$ is zero. The key idea is to radially calculate $E(s, F, d)$ starting from the destination d outward to the rest of the network. Specifically, we calculate $E(s, F, d)$ as follows:

$$E(s, F, d) = \alpha + \beta, \tag{5.3}$$

where α captures the expected number of transmissions for successfully transmitting a packet from s to at least one of the candidates and getting at least one acknowledgment. β captures the expected number of transmissions for delivering the packet in turn from those candidates to the destination.

• **Three-candidate Case:** we start by considering the 3-candidate case in Figure 5.3, where the receptions of candidate f_1, f_2 and f_3 are partially correlated. Figure 5.3(b) shows the Venn diagram representing the events that candidates successfully receive a transmission from the sender. E_{s,f_i} is the event that a packet is successfully received by forwarder f_i with one transmission. The intersection areas represent the correlated events. For example, $E_{s,f_1} \cap E_{s,f_2}$ represents the correlation between event E_{s,f_1} and E_{s,f_2}. Let α_0 be the expected number of transmissions for s to successfully deliver a packet to the candidate set. α_0 is the inverse of the total area in Venn diagram, which



(a) Topology (b) Joint Reception

Figure 5.3: Three-candidate case example.

is given by

$$\begin{aligned}\alpha_0^{-1} =& p_{s,f_1} + p_{s,f_2} + p_{s,f_3} - Pr(E_{s,f_1}, E_{s,f_2}) \\ & - Pr(E_{s,f_1}, E_{s,f_3}) - Pr(E_{s,f_2}, E_{s,f_3}) \\ & + Pr(E_{s,f_1}, E_{s,f_2}, E_{s,f_3}).\end{aligned} \tag{5.4}$$

To calculate α, we need to count in the lost ACKs from each candidate to s by multiplying each term in Eq.(5.4) with the probability of successfully receiving the ACK. we thus have

$$\begin{aligned}\alpha^{-1} =& p_{s,f_1}p_{f_1,s} + p_{s,f_2}p_{f_2,s} + p_{s,f_3}p_{f_3,s} \\ & - Pr(E_{s,f_1}, E_{s,f_2})p_{s,f_1}p_{s,f_2} \\ & - Pr(E_{s,f_1}, E_{s,f_3})p_{s,f_1}p_{s,f_3} \\ & - Pr(E_{s,f_2}, E_{s,f_3})p_{s,f_2}p_{s,f_3} \\ & + Pr(E_{s,f_1}, E_{s,f_2}, E_{s,f_3})p_{s,f_1}p_{s,f_2}p_{s,f_3}.\end{aligned} \tag{5.5}$$

When the candidate f_1, f_2 or f_3 succeeds in receiving a packet, it will take over the forwarding process as long as it does not receive an ACK from a higher priority candidate. For candidate f_1, it forwards a packet as long as it receives it since f_1 has the highest priority. we thus have

$$\beta_{f_1} = \alpha_0 \cdot \big(E(f_1, F, d) \cdot (1 - \gamma_{f_1})\big), \tag{5.6}$$

where α_0 is multiplied because of the implicit condition that at least one of the candidates has received the packet, and γ_{f_1} is the probability that candidate f_1 will not take the forwarding process, which equals $1 - p_{s,f_1}$. For the lowest priority candidate f_3, the forwarding process will not happen when f_3 fails to receive a packet, or when an ACK is received from f_1 or f_2. The probability that candidate f_3 will not take the forwarding process γ_{f_3} can be calculated as follows:

$$\begin{aligned}\gamma_{f_3} =& 1 - p_{s,f_3} + p_{f_1,f_3} Pr(E_{s,f_3}, E_{s,f_1}) \\ &+ p_{f_2,f_3} Pr(E_{s,f_2}, E_{s,f_3}).\end{aligned} \tag{5.7}$$

For candidate f_2, an ACK could be received explicitly from candidate f_1 or implicitly from candidate f_3 when f_3 receives the packet from s and the ACK from f_1. γ_{f_2} is thus given by

$$\begin{aligned}\gamma_{f_2} &= 1 - p_{s,f_2} + p_{f_1,f_2} Pr(E_{s,f_1}, E_{s,f_2}) \\ &+ (1 - p_{f_1,f_2}) \cdot p_{f_1,f_3} p_{f_3,f_2} Pr(E_{s,f_1}, E_{s,f_2}, E_{s,f_3}).\end{aligned} \tag{5.8}$$

Similar to (5.6), we now obtain β_{f_2} and β_{f_3} where γ_{f_2} and γ_{f_3} can be calculated with Eq.(5.7) and Eq.(5.8). Finally, β is the sum of β_{f_1}, β_{f_2} and β_{f_3}.

• **n-candidate Case:** we now extend the problem to n candidates. It turns out to be an inclusion-exclusion where we should sum the probabilities of individual links but remove the overlapped intersection:

$$\alpha = \frac{1}{\sum_{k=1}^{n} (-1)^{k-1} Pr(f^k)}, \tag{5.9}$$

where $f^k \subset F = \{f_1, \ldots, f_n\}$ is any candidate forwarder set with size k, and $Pr(f^k)$ is the probability that the k candidate forwarders successfully receive a packet and send back the ACK. $Pr(f^k)$ is calculated as follows:

$$Pr(f^k) = \sum_{f^k \subset F} Pr(E_{s,f_1^k}, \ldots, E_{s,f_k^k}) \prod_{i}^{k} p_{f_i^k,s}. \tag{5.10}$$

• **Special Case:** Eq.(5.9) includes the special link independence case. When wireless links are independent, we have

$$Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k}) = \prod_{i}^{k} p_{s,f_i^k}.$$

And $Pr(f^k)$ in Eq.(5.10) turns out to be

$$Pr(f^k) = \sum_{f^k \subset F} \prod_{i}^{k} p_{s,f_i^k} p_{f_i^k,s}.$$

For the candidates which have received the packet from the transmitter, the expected number of transmissions to forward the packet to the next hop is calculated as follows:

$$\beta = \alpha_0 \cdot \sum_{i=1}^{n} \big(E(f_i, F, d) \cdot (1 - \gamma_{f_i}) \big). \tag{5.11}$$

where γ_{f_i} is the probability that candidate f_i does not forward the packet. To calculate γ_{f_i}, we need to consider three cases: (i) candidate f_i loses the packet, (ii) candidate f_i receives an direct ACK from a higher priority candidate, and (iii) the candidate receives an indirect ACK through a low priority candidate. Compared with the first two cases, the third case can be ignored since it rarely happens. γ_{f_i} thus can be calculated using the following equation:

$$\gamma = (1 - p_{s,f_i}) + \sum_{k=1}^{j<i} (-1)^{k-1} Pr(f^k). \tag{5.12}$$

Eq.(5.12) consists two parts, i.e., $(1 - p_{s,f_i})$ and $\sum_{k=1}^{j<i}(-1)^{k-1}Pr(f^k)$. The first part represents the the first case. $\sum_{k=1}^{j<i}(-1)^{k-1}Pr(f^k)$ describes the second case, which represents the union of the probability of each higher priority candidate in receiving the packet and sending back the ACK.

5.5 Implementation

In this section, we introduce the implementation detail of the link correlation aware metric $E(s, F, d)$. The calculation of $E(s, F, d)$ finally goes to find the link quality (i.e., p_{s,f_i} and $p_{f_i,s}$) and the link correlation (i.e., $Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k})$). Suppose each

receiver maintains a packet reception report (e.g., [1101]) recording the reception status of a fixed number (e.g., 4) of most recent packets. With the reception report, the link quality is given simply by the number of 1s in the reception report divided by the length of reception report. The calculation of $Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k})$ deserves a little more explanation. Here we use an example to show how to calculate $Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k})$. Assume a reception report of length L, we have

$$Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k}) = \frac{1}{L}\sum_{j=1}^{L} B_{f_1}(j)\&\ldots\&B_{f_k}(j), \tag{5.13}$$

where $B_{f_i}(j)$ is a bit representing the candidate f_i's reception status of the jth packet. $B_{f_i}(j)=1$ represents candidate f_i receives the packet, otherwise $B_{f_i}(j)=0$. For example, in Figure 5.4, the sender s has three candidates. we calculate s's $Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k})$ when k equals two, i.e., $Pr(E_{s,f_1^2},\ldots,E_{s,f_2^2})$. Suppose the reception report of candidate f_1 is [1101], which indicates that f_1 receives the 1st, 2nd, and 4th packets and misses the 3rd packet. When the sender s receives the reception reports from the candidates, it uses Eq.(5.13) to calculate $Pr(E_{s,f_1^2},\ldots,E_{s,f_2^2})$, i.e.,

$$\begin{aligned} Pr(E_{s,f_1^2},E_{s,f_2^2}) &= \frac{1}{4}(1\&0+1\&1+0\&0+1\&1) \\ &= 50\%. \end{aligned}$$

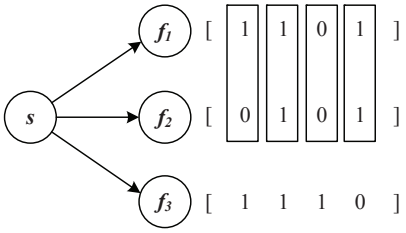


Figure 5.4: An example of calculating s's $Pr(E_{s,f_1^k},\ldots,E_{s,f_k^k})$ for $\{f_1,f_2\}$.

5.5.1 Metric Overhead

• **Computational Cost:** In opportunistic routing, having a large set of candidates may reduce the number of transmissions from the source to the destination. On the

other hand, it may bring serious problems of increasing the schedule overhead among candidates as well as the chance of duplicated transmissions, which may reduce the efficiency of opportunistic routing. Moreover, the computational cost of searching the optimal candidate set increases dramatically due to the exponentially increased combination [72, 73]. In practice, the number of candidates that can be used is set to a small number, e.g., 4. Therefore, the computation cost of the metric is low due to the small size of the candidate set.

• **Communication Cost:** the metric needs to calculate link quality and link correlation which may change over time. we now discuss the overhead for maintaining the metric accurate. we conduct an experiment in a dynamic scenario. In the experiment, the sender transmits packets every 0.2 seconds while the packet reception report is sent in every 20s. The total number of packets sent is 8000. The candidates keep track of the received packets through the packet sequence numbers.

The main communication overhead of the metric comes from the reception reports which are used to calculate link quality and $Pr(E_{s,f_1^k}, \ldots, E_{s,f_k^k})$. The required information for link quality and $Pr(E_{s,f_1^k}, \ldots, E_{s,f_k^k})$ is exactly the same, i.e., the reception report. It not only helps to calculate link quality, which only uses the bit information in rows in Figure 5.4, but also provides information of links' relationship, i.e., the bit information in columns in Figure 5.4.

To collect reception reports, we adopt the free piggyback mechanism with normal traffic data, which has been already applied by the existing protocols [47] to measure link quality or to improve the robustness of the routing structure. The binary reception report is small and is much less frequently transmitted, therefore the overhead occupies a tiny fraction (0.9%) of the total energy cost according to the measurements.

5.5.2 Metric Accuracy

we run the experiments in a period of 30 minutes, during which reception reports (0.9% overhead) are used to fresh link quality and $Pr(E_{s,f_1^k}, \ldots, E_{s,f_k^k})$ values. Figure 5.5 compares the real values with the estimates using the link correlation metric α_0 and the link independent metric ETX. From Figure 5.5, we find that α_0 is accurate over time and the number of received packets by the forwarder set (during 20s) closely follows the number of sent packets (i.e., 100) divided by α_0.

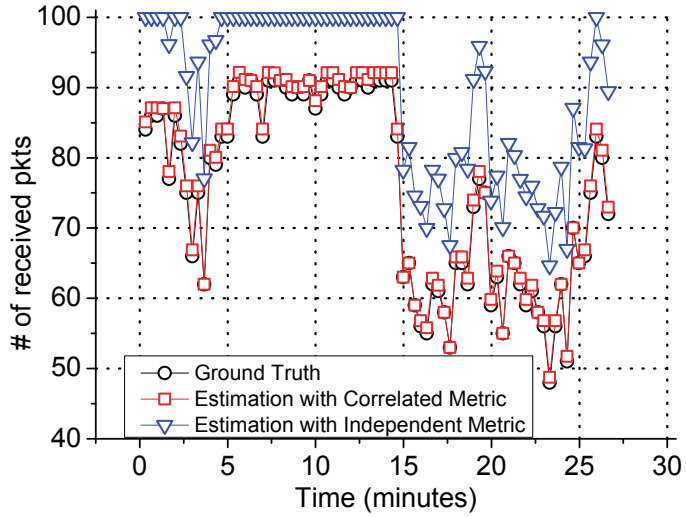


Figure 5.5: Received pkts vs. Estimation with α_0 and ETX.

From Figure 5.5, we also find that the link independence metric always overestimates the number of received packets. This is because in the testbed environment, the links are positive correlated because of cross-technology interference and correlated shadowing [74, 75, 76]. Under such an environment, the forwarders in the candidate set lose similar packets. With the link independence assumption, when one forwarder fails to receive a packet, others may have a better chance of receiving it. This optimistic view thus leads to an overestimate of diversity benefit and transmission efficiency.

Algorithm 1 CANDIDATES SELECT(s,d,setsize)

$F \Leftarrow \emptyset; \hat{F} \Leftarrow \emptyset; m_p \Leftarrow \infty; m_c \Leftarrow \infty$
for all $v \in N(s)$ **do**
  if $\mathrm{ETX}(v,d) < \mathrm{ETX}(s,d)$ **then**
    $\hat{F} \Leftarrow \hat{F} \cup v$
  end if
end for
while $|F| < setsize$ **do**
  $cand \Leftarrow \underset{c \in \hat{F}}{\mathrm{argmin}} E(s, F \cup c, d)$
  $m_c \Leftarrow E(s, F \cup cand, d)$
  if $m_c < m_p$ **then**
    $F \Leftarrow F \cup cand; \hat{F} \Leftarrow \hat{F} \backslash cand$
    $m_p \Leftarrow m_c$
  else
    $E(s,F,d) \Leftarrow m_p$; break
  end if
end while

5.5.3 Metric Embedding

This section describes how we integrate the link correlation aware metric $E(s, F, d)$ into opportunistic routing to select the candidate forwarder set. The key idea is to select candidates with good link quality and prioritize them according to the $E(s, F, d)$ value. The design is specified by the pseudo code in Algorithm 1. we initialize the candidate set $\hat{F}$ by adding nodes with smaller ETX values. At this step, we create a directed acyclic graph from the source s to the destination d and eliminate candidates which have higher ETX values (Lines 1-6). In lines 7-16, we loop through the initial candidate set and find the candidate with the minimum $E(s, F, d)$ value. we add the node to the candidate set F and remove it from the initial set $\hat{F}$. we loop the above procedures until we find enough candidates to meet the predesigned set size.

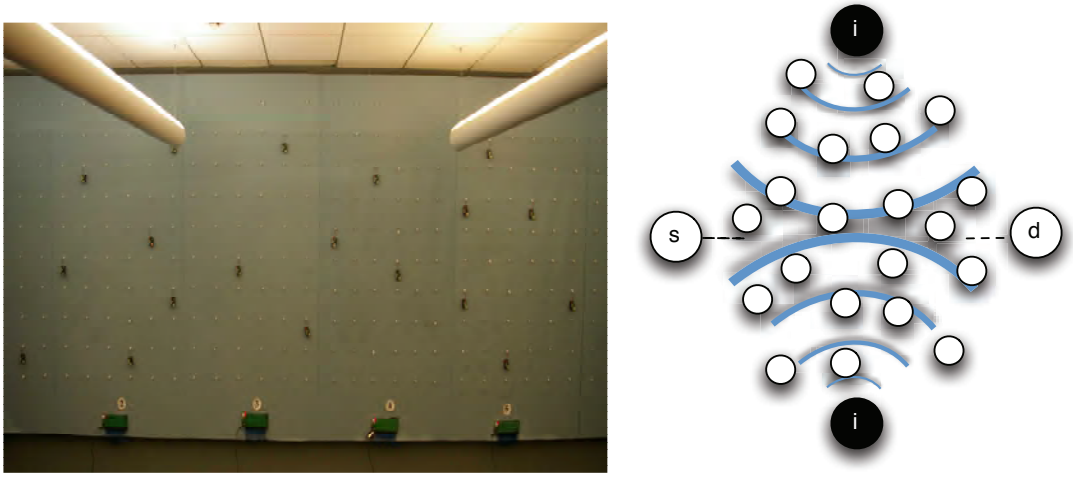
5.6 Testbed Experimentation

In the real world environment, interference exits everywhere because of the massive number of uncontrollably deployed wireless devices sharing the same unlicensed spectrum. In the testbed experiment, we adopt interferers to introduce negatively correlated, positively correlated and uncorrelated links. we investigate the effect of various correlation degrees on the performance of the design.

5.6.1 Generating Correlated Links

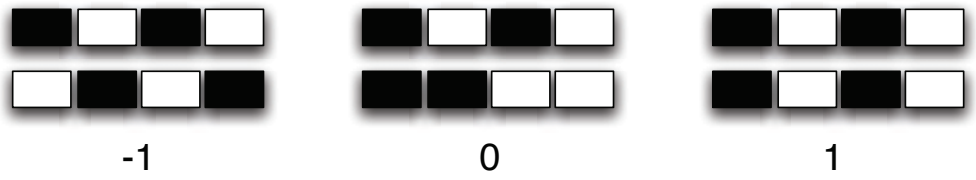
we deploy 24 MICAz nodes in the indoor 24 feet by 8 feet testbed, as shown in Figure 5.6(a). It forms a simple two-hop network as shown in Figure 5.6(b), where we have one source s, one destination d, two interferers, and a set of 20 candidate forwarders. The two interferers are placed randomly within a pool of forwarders. They are used to create interference of various patterns. Figure 5.7 shows selected examples of interference patterns which may further lead to different degrees of link correlation.

we use the maximum transmission power, i.e., 0 dBm, to make sure that the links from the forwarders to the destination d are perfect. For the links from the source s to the candidate forwarders, the transmission power is controlled carefully to introduce the packet loss in the presence of the interference signal. The source s sends packets



(a) Physical deployment (b) Logical topology

Figure 5.6: The testbed with 24 MICAz nodes.



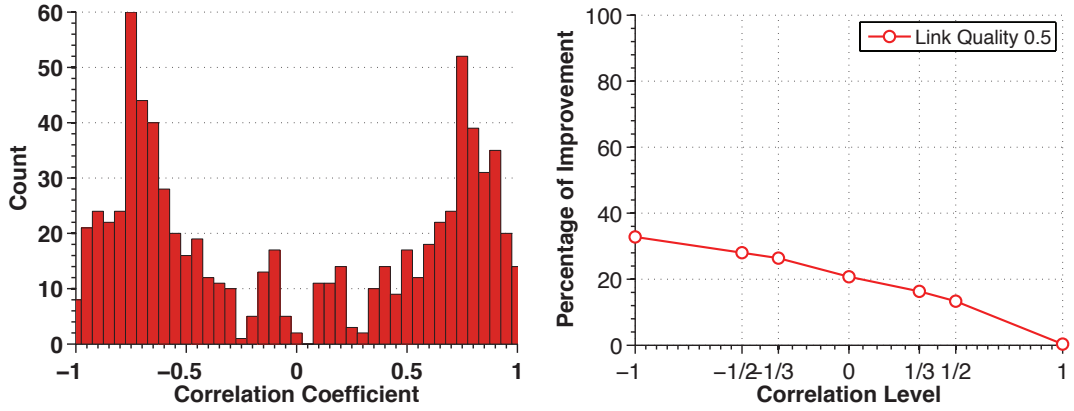
(a) Negative correlated (b) Uncorrelated (c) Positive correlated

Figure 5.7: The interference pattern which is used to generate link correlation.

with 2-byte data payload in every 0.2s. In the default setting, the source node keeps on sending packets until the destination returns an ACK. we use 802.15.4 channel 26 to avoid the effect of Wi-Fi interference.

5.6.2 Performance Evaluation

In this section, we compare the performance of opportunistic routing with and without considering link correlation. we use "CA" to represent the link correlation aware opportunistic routing. "CU" means the traditional correlation unaware opportunistic routing. The size of forwarder set is two. In the experiment, we maintain the link quality from the source to candidate forwarders to be almost the same. In the correlation unaware opportunistic routing, two forwarders which are the most positively correlated are selected. In correlation aware opportunistic routing, we select forwarders with different degrees of link correlation. The experiment results are the average values taken from 1000 samples. In the following, we show the performance results of the correlation aware and unaware design on the number of transmissions, energy consumption, and delivery ratio.

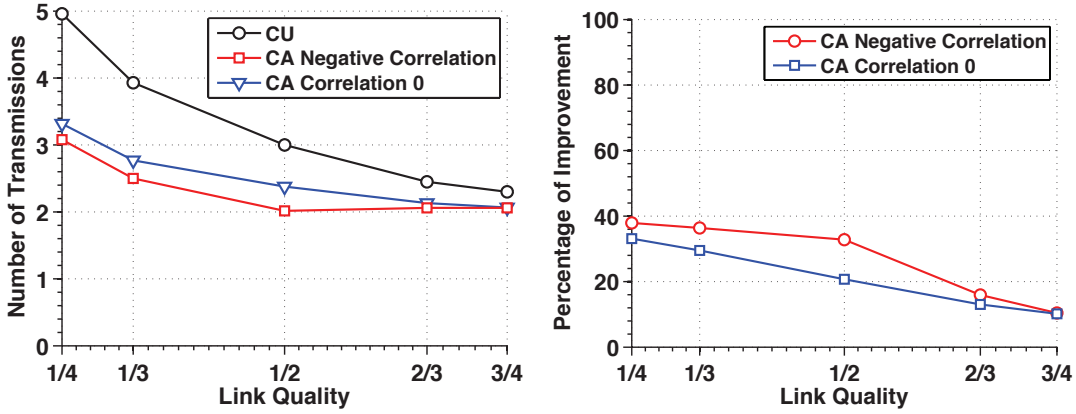


(a) Correlation distribution (b) Performance improvement

Figure 5.8: The impact of correlation level.

5.6.3 Results on Transmissions

• **Varied Correlation Levels:** we start off by showing the impact of link correlation degree on opportunistic routing. To focus on the effect of link correlation, link quality is kept constant at 0.5. Figure 5.8(a) plots the distribution of pairwise link correlations in the presence of the interference pattern of Figure 5.7(a). The achievable performance gain of the correlation aware opportunistic routing is shown in Figure 5.8(b). From the figure, we find that opportunistic routing obtains the maximum 33% improvement, when negatively correlated forwarders are selected. That's because when one forwarder fails to receive a packet, the other is very likely to receive it under negative correlated link correlation. we also find that the improvement decreases when the correlation level increases and there are no improvement at all when the links are perfectly correlated. That's because when the packet receptions of the candidate are highly correlated, link quality becomes the only factor affecting the selection process, and no diversity benefits can be exploited.



(a) Number of transmissions

(b) Percentage of improvement

Figure 5.9: The impact of link quality.

• **Varied Link qualities:** In this experiment, we investigate the impact of link quality on the design. The results are shown in Figure 5.9 where Figure 5.9(a) shows the number of transmissions needed by correlation aware and correlation unaware designs and Figure 5.9(b) shows the corresponding improvement percentage introduced by the correlation aware design. From the figures, we find that the correlation aware scheme

obtains significant improvement when link quality is low. For example, the link correlation aware opportunistic routing obtains the best performance – an improvement of 38% under 1/4 link quality. In the environment where we introduce independent interferences, we observe that the performance gain of the design is still significant, and its performance gain is lower than the negative link correlation scenario.

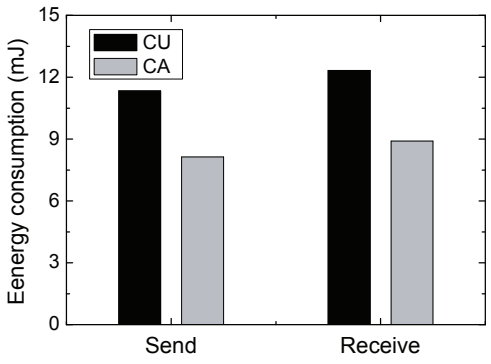


Figure 5.10: Energy Consumption.

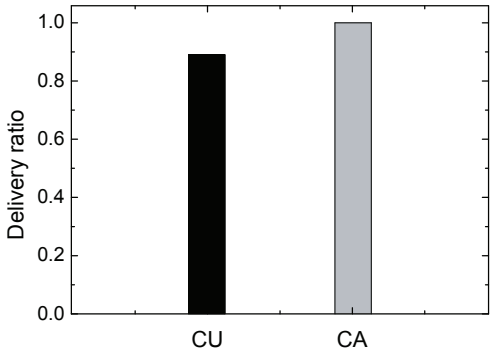


Figure 5.11: Delivery Ratio.

5.6.4 Results on Energy Consumption

Figure 5.10 plots the experiment results of the correlation aware and unaware designs on energy consumption from both sending and receiving aspects. From the figure, we can see that on average the sender takes 11.34 mJ, including Clear Channel Assessment (CCA) and transmitting operations, to deliver one packet to the destination with correlation unaware opportunistic routing. With the correlation aware design, this part energy consumption deduces to 8.13 mJ, which thus saves 28.3% energy consumption. At the receiver side, the energy consumption for CU is 12.33 mJ while it's 8.90 mJ for CA. This part energy consumption includes listening and receiving. the design improves 27.8% energy efficiency.

5.6.5 Results on Delivery Ratio

In the default setting, we retransmit as many packets as possible until the source node receives an ACK from the destination. In this experiment, we limit the maximum

number of retransmissions to three. The experiment result is shown in Figure 5.11. From the figure, the reliability of CA, and CU is 100%, and 89%, respectively. With the packet reception correlation information, the design helps opportunistic routing find the suitable candidate forwarders and save the number of retransmissions.

5.7 Simulation Evaluation

In this section, we evaluate the link correlation aware design in simulations with various network settings.

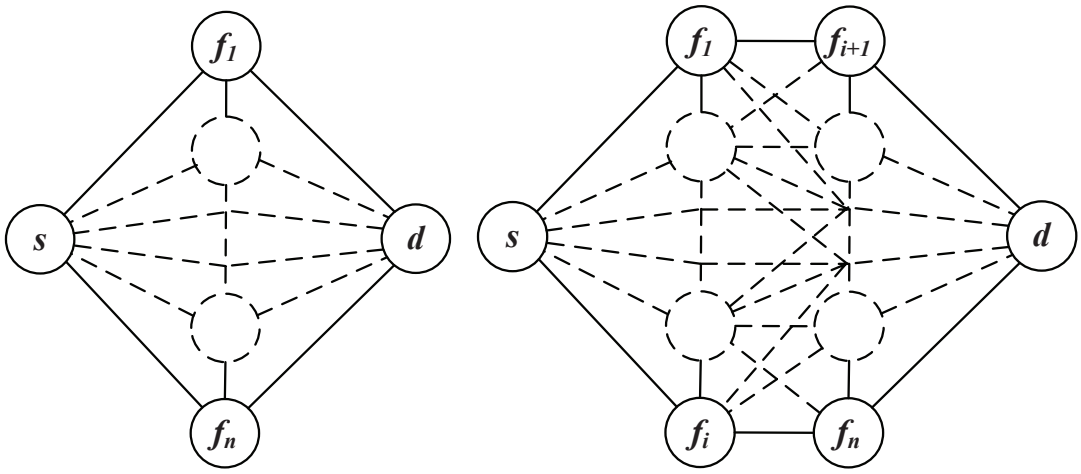
5.7.1 Link Correlation Generation Model

we generate correlated links using the sampling algorithm for correlated Bernoulli random variables, described in [84]. The inputs of the algorithm are the mean and the covariance matrix of the joint Bernoulli distribution. It then uses a dichotomized multivariate Gaussian distribution to sample the multivariate Bernoulli distribution. we need to choose the covariance matrix carefully since it cannot be always associated with a valid Bernoulli distribution. we thus use the algorithm in [85] to obtain the closest admissible matrix. The algorithm converts the inadmissible matrix using an iterative projection algorithm into the closest unique admissible matrix in the Euclidean norm.

Link correlation can be either positive or negative depending on the possible cause that affects the inter-link reception. To accommodate such correlations, we create a correlation matrix with the covariance matrix randomly selected from $[-1, 1]$. we generate a string of 10,000 sequences for each sender to capture the link correlation. we then run simulations on OMNeT++ and the Castalia Framework using the generated correlated traces to sample transmission success or failure events.

5.7.2 Simulation Setup

we randomly generate different network topologies using the Waxman model [86], where the nodes are uniformly distributed in the plane and edges are added according to the probability that depends on the distance between the nodes. The network size is 50. we consider both the single-hop and multi-hop scenarios. The logical topologies are shown



(a) Single-hop (b) Multi-hop

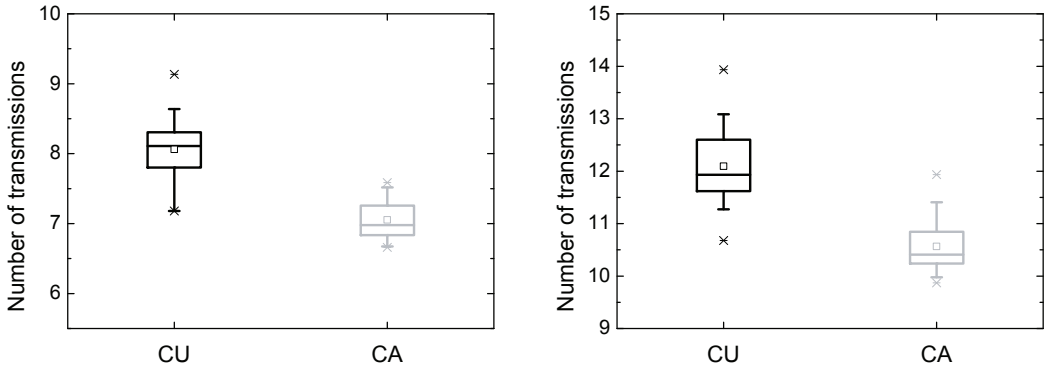
Figure 5.12: The topologies used to investigate the effect of link correlation awareness.

in Figure 5.12, where a sender s has n initial candidate forwarders to the destination d. The sender s transmits 10,000 packets to the destination d.

Opportunistic routing obtains significant benefit when link qualities are low. That's because when link quality is high, the forwarder set is not so necessary and its function is almost the same as the best link. The performance of opportunistic routing is quite close to the traditional shortest path routing under such scenarios. In the following simulations, we mainly focus on low link quality scenarios. Both the links from the source to the forwarder set and the links from the forwarder set to the destination are set to be lossy. In the experiments, the default size of the candidate forwarder set is two.

5.7.3 Main Performance Results

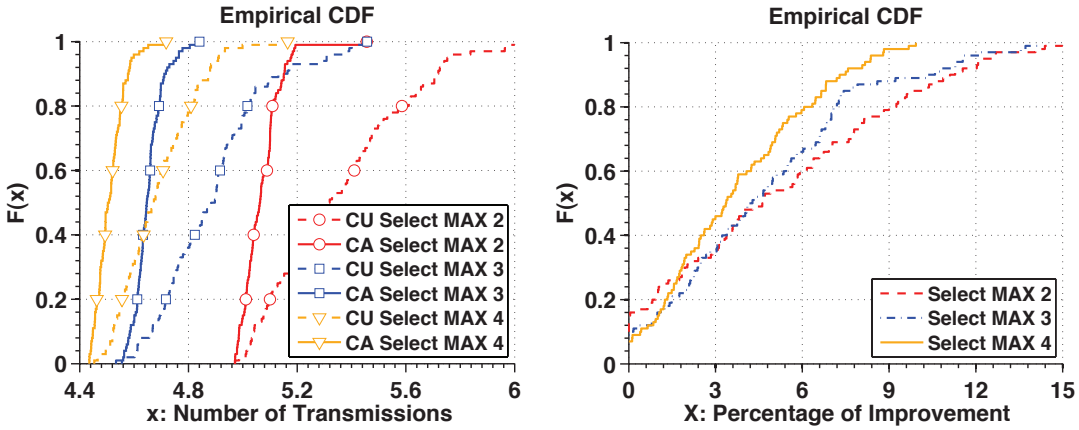
The main experimental results in both single-hop and multi-hop scenarios are shown in Figure 5.13. The box plot in Figure 5.13(a) shows the experiment result in the single-hop scenario where the average number of transmissions with CA is 7.05, which is much less than CU's 8.06. The CA design obtains the improvement because it replaces the candidates selected by previous link independent metric when it finds more diversity benefits can be exploited. The performance of the two algorithms in the multi-hop



(a) Single-hop scenario (b) Multi-hop scenario

Figure 5.13: Main simulation results in single-hop and multi-hop scenarios.

scenario is shown in Figure 5.13(b), where a similar result, i.e., a 15% improvement, is observed.



(a) The CDF of the number of transmissions (b) The CDF of the percentage of improvement

Figure 5.14: The impact of forwarder set sizes.

5.7.4 Impact of Candidate Set Sizes

In this experiment, we examine the performance of the design with different candidate set sizes. we investigate the cases with 2, 3, and 4 candidates. The results are shown in Figure 5.14. From Figure 5.14(a) we can see that increasing the size of the candidate

set would improve the performance of opportunistic routing with or without correlation awareness. This happens because the more links we have, the more diverse we can exploit. From Figure 5.14(b), we find that the percentage of improvement of the design is reduced when we add more nodes since little improvement room is left when we have enough candidates.

5.8 Summary

This chapter extensively studies the impact of link correlation on the performance of opportunistic routing. we provide a detailed analysis of the opportunistic routing framework under the influence of link correlation. we find that diversity benefit is overestimated when we assume that packet receptions of wireless links are independent. A link correlation aware metric is thus proposed to improve the performance of opportunistic routing by selecting the nodes with diverse low correlated links as forwarder candidates. we evaluate the work with testbed implementation and extensive simulations. The experiment result affirms the efficiency of the design in capturing the full advantage of opportunistic routing.

Chapter 6

Conclusion and Future Work

This dissertation studies the packet reception patterns under cross-technology interference and reveals link correlation phenomenon that packet receptions from a transmitter to multiple receivers are correlated. This finding contradicts the widely made link independent assumption. A novel link correlation model is thus proposed. The proposed model has a broad impact on network designs that utilize concurrent wireless links. We have studied the impact of link correlation on (i) traditional network protocols including sixteen broadcast protocols, and (ii) diversity-based protocols including seven network coding protocols and one opportunistic routing protocol. Our system effort is significant. We integrate our link correlation model with sixteen broadcast protocols, seven network coding protocols, and one opportunistic routing protocol. Experiment results show that link correlation model significantly improves the transmission efficiency of broadcast, network coding, and opportunistic routing.

This dissertation focuses on (i) building link correlation model under cross-technology interference and (ii) proposing its corresponding applications in broadcast, network coding and opportunistic routing. In the future, I plan to propose advanced designs to resolve the communication failure under cross-technology interference. In detail, I plan to design robust coding/decoding schemes in physical layers which are able to decode packets under cross technology interference with high success rate.

Furthermore, building cross technology communication is a novel way to resolve the cross technology interference problem. Cross technology communication provides a communication channel for multiple wireless devices with different technologies to

coordinate their communication, and thus resolving the communication failure problem under cross technology interference.

References

[1] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *Network Protocols (ICNP), 2010 18th IEEE International Conference on*, pages 305–314, Oct 2010.

[2] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, 2010.

[3] Xinyu Zhang and Kang G. Shin. Cooperative carrier signaling: Harmonizing coexisting wpan and wlan devices. *IEEE/ACM Trans. Netw.*, 21(2), April 2013.

[4] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *MobiCom'05*, 2005.

[5] E. Miluzzo, X. Zheng, K. Fodor, and A. T. Campbell. Radio characterization of 802.15.4 and its impact on the design of mobile sensor networks. In *EWSN'08*, 2008.

[6] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5:11–25, October 2001.

[7] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: Measuring wireless link burstiness. In *SenSys'08*, 2008.

[8] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *Technical Report SING-06-00*, 2006.

[9] Shan Lin, Jiangbin Zhang, Gang Zhou, Lin Gu, Tian He, and John A. Stankovic. Atpc: Adaptive transmission power control for wireless sensor networks. In *Fourth ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, November 2006.

[10] M. Zuniga, and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. In *ACM Transactions on Sensor Networks*, 2007.

[11] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys 2003)*, pages 1–13, 2003.

[12] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 2002.

[13] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *HICSS*, 2002.

[14] A. Forster and A.L. Murphy. Froms: A failure tolerant and mobility enabled multicast routing paradigm with reinforcement learning for wsns. *Elsevier Ad Hoc Networks*, 2011.

[15] Q. Huang, C. Lu, and G. Catalin Roman. Spatiotemporal multicast in sensor networks. In *SENSYS*, 2003.

[16] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE*, 5, 2001.

[17] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204 –1216, jul 2000.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 2008.

[19] E. L. Li, R. Ramjee, M. M. Buddhikot, and S. C. Miller. Network coding-based broadcast in mobile ad-hoc networks. In *Proceedings of IEEE INFOCOM*, 2007.

[20] Q. Cao, T. Abdelzaher, T. He, and R. Kravets. Cluster-based forwarding for reliable end-to-end delivery in wireless sensor networks. In *INFOCOM*, 2007.

[21] S. Biswas and R. Morris. Exor: Opportunistic multi-hop routing for wireless networks. In *SIGCOMM*, 2005.

[22] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *SIGCOMM*, 2007.

[23] J. Du, H. Liu, and P. Chen. Omr: An opportunistic multi-path reliable routing protocol in wireless sensor networks. In *ICPPW*, 2007.

[24] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. In *Technical Report 4597, INRIA-Rapport de recherche*, October 2002.

[25] K.M. Alzoubi, P.J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Hawaii Int. Conf. System Sciences*, 2002.

[26] J.Cartigny, F. Ingelrest, and D. Simplot. Rng relay subset flooding protocols in mobile ad hoc networks. *International Journal of Foundations of Computer Science*, 2003.

[27] A. Juttner and A. Magi. Tree based broadcast in ad hoc networks. *Mobile Networks and Applications*, 10(5), 2005.

[28] T. J. Kwon and M. Gerla. Efficient flooding with passive clustering (pc) in ad hoc networks. In *SIGCOMM Computer Communication Review*, January 2002.

[29] H. Lim and C. Kim. Flooding in wireless ad hoc networks. In *Computer Communications Journal*, 2001.

[30] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks. *IEEE TPDS*, 2002.

[31] M. T. Sun, X. Ma, C. Y. Yi, C. K. Yang, and T. H. Lai. Computing optimal local cover set for broadcasting in ad hoc networks. *Lecture Notes in Computer Science - Networking and Mobile Computing*, 2005.

[32] J. Wu and H. Li. A dominating set based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 2001.

[33] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. In *Wireless Communication and Mobile Computing*, 2003.

[34] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th International Conference on Mobile Computing and Networking (MobiCom 2003)*, 2003.

[35] J. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *SENSYS*, 2004.

[36] S. Kulkarni and L. Wang. Mnp: Multihop network reprogramming service for sensor networks. In *ICDCS*, 2005.

[37] Jakob Eriksson, Hari Balakrishnan, and Samuel Madden. Cabernet: vehicular content delivery using wifi. In *MOBICOM*, 2008.

[38] S. J. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. In *ACM/Kluwer MONET*, 2002.

[39] T. Clausen, C. Dearlove, and P. Jacquet. The optimized link state routing protocol version 2. In *MANET Working Group*, 2008.

[40] G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava. Tree-based data broadcast in ieee 802.15.4 and zigbee networks. *IEEE Transactions on Mobile Computing*, 2006.

[41] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. The κ-factor: Inferring protocol performance using inter-link reception correlation. In *MOBICOM*, 2010.

[42] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *NSDI*, 2010.

[43] S. Ni, Y. Tseng Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MOBICOM*, 1999.

[44] I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. *Ad Hoc Networking*, 2004.

[45] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. In *INFOCOM*, 2002.

[46] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza. Rbp: Robust broadcast propagation in wireless networks. In *SENSYS*, 2006.

[47] Nouha Baccour, Anis Koubaa, Luca Mottola, Marco Zuniga, Habib Youssef, Carlo Alberto Boano, and Mario Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks*, 8, November 2012.

[48] Emiliano Miluzzo, Xiao Zheng, Kristof Fodor, and Andrew T. Campbell. Radio characterization of 802.15.4 and its impact on the design of mobile sensor networks. In *EWSN*, 2008.

[49] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low power wireless. In *SING Technical Report*, 2008.

[50] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection Tree Protocol. In *SenSys '09*, 2009.

[51] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MOBICOM*, 2000.

[52] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SENSYS*, 2004.

[53] J. Macke, P. Berens, A. Ecker, A. Tolias, and M. Bethge. Generating spike trains with specified correlation coefficients. *Neural Computation*, 2009.

[54] A. Keshavarz-Haddad and R. Riedi. Bounds on the benefit of network coding: Throughput and energy saving in wireless networks. In *Proc. of IEEE INFOCOM*, 2008.

[55] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He. Correlated flooding in low-duty-cycle wireless sensor networks. In *ICNP*, 2011.

[56] S. Sengupta, S. Rayanchu, and S. Banerjee. Network coding-aware routing in wireless networks. *IEEE Transactions on Networking*, 18:1158–1170, 2010.

[57] J. Le, C. S. Lui, and D. M Chiu. Dcar: Distributed coding-aware routing in wireless networks. *IEEE Transactions on Mobile Computing*, 9:596–608, 2010.

[58] S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He. Coding opportunity aware backbone metrics for broadcast in wireless networks. In *Proc. of IEEE INFOCOM*, 2013.

[59] A. Atya, I. Broustis, S. Singh, D. Syrivelis, S. Krishnamurthy, and T. Porta. Wireless network coding: Deciding when to flip the switch. In *Proc. of IEEE INFOCOM*, 2013.

[60] P. Chaporkar and A. Proutiere. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *Proc. of ACM MOBICOM*, 2007.

[61] Shuai Wang, Song Min Kim, Yunhuai Liu, Guang Tan, and Tian He. Corlayer: A transparent link correlation layer for energy efficient broadcast. In *MobiCom 2013*.

[62] S. Wang, S. M. Kim, Y. Liu, G. Tan, and T. He. Corlayer: A transparent link correlation layer for energy-efficient broadcast. *IEEE Transactions on Networking*, 23:1970–1983, 2014.

[63] *IEEE Standard for Information technology*. IEEE STANDARDS ASSOCIATION, 2012.

[64] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1:111–122, 2002.

[65] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MOBICOM*, 2003.

[66] *Lorcon wireless packet injection library*. https://code.google.com/p/lorcon/.

[67] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[68] K. Zeng, Z. Yang, and W. Lou. Opportunistic routing in multi-radio multi-channel multi-hop wireless networks. In *INFOCOM*, pages 3512–3521. IEEE, 2010.

[69] C.P. Luk, W.C. Lau, and O.C. Yue. Opportunistic routing with directional antennas in wireless mesh networks. In *INFOCOM*, pages 2886–2890, 2009.

[70] Z. Zhong and S. Nelakuditi. On the efficacy of opportunistic routing. In *SECON'07*, pages 441–450. IEEE, 2007.

[71] Y. Li, W. Chen, and Z.L. Zhang. Optimal forwarder list selection in opportunistic routing. In *MASS'09.*, pages 670–675. IEEE, 2009.

[72] H. Dubois-Ferrière, M. Grossglauser, and M. Vetterli. Valuable detours: Least-cost anypath routing. *Networking, IEEE/ACM Transactions on*, 19(2):333–346, 2011.

[73] L. Darehshoorzadeh, A.and Cerda-Alabern and V. Pla. Modeling and comparison of candidate selection algorithms in opportunistic routing. *Computer Networks*, 55(13):2886–2898, 2011.

[74] T. Zhu, Z. Zhong, T. He, and Z.L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *USENIX NSDI*, pages 4–4. USENIX Association, 2010.

[75] Kannan Srinivasan, Mayank Jain, Jung Il Choi, Tahir Azim, Edward S. Kim, Philip Levis, and Bhaskar Krishnamachari. The kappa factor: Inferring protocol performance using inter-link reception correlation. In *MobiCom 2010*, 2010.

[76] A. Zubow, M. Kurth, and J.P. Redlich. Considerations on forwarder selection for opportunistic protocols in wireless networks. *EW*, 2008.

[77] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. *ACM SIGCOMM*, 37(4):169–180, 2007.

[78] A. Khreishah, I. Khalil, and J. Wu. Distributed network coding-based opportunistic routing for multicast. In *MobiHoc*, 2012.

[79] D. Koutsonikolas, C.C. Wang, and Y.C. Hu. Ccack: Efficient network coding based opportunistic routing through cumulative coded acknowledgments. In *INFOCOM*, pages 1–9. IEEE, 2010.

[80] E. Rozner, J. Seshadri, Y.A. Mehta, and L. Qiu. Soar: Simple opportunistic adaptive routing protocol for wireless mesh networks. *IEEE Transactions on Mobile Computing*, pages 1622–1635, 2009.

[81] D. Liu, Z. Cao, J. Wang, Y. He, M. Hou, and Y. Liu. Dof: Duplicate detectable opportunistic forwarding in duty-cycled wireless sensor networks. In *ICNP*, pages 1–10, 2013.

[82] X. Zhang and B. Li. Optimized multipath network coding in lossy wireless networks. *Selected Areas in Communications, IEEE Journal on*, 27(5):622–634, 2009.

[83] Z. Zhong, J. Wang, S. Nelakuditi, and G.H. Lu. On selection of candidates for opportunistic anypath forwarding. In *SIGMOBILE Mob. Comput. Commun. Rev.*, pages 1–2. ACM, 2006.

[84] J.H. Macke, P. Berens, A.S. Ecker, A.S. Tolias, and M. Bethge. Generating spike trains with specified correlation coefficients. *Neural Computation*, 21(2):397–423, 2009.

[85] N.J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329, 2002.

[86] B.M. Waxman. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, 6(9):1617–1622, 1988.