CorLayer: A Transparent Link Correlation Layer for Energy Efficient Broadcast

Shuai Wang
University of Minnesota
Minneapolis, MN, USA
shuaiw@cs.umn.edu

Song Min Kim
University of Minnesota
Minneapolis, MN, USA
ksong@cs.umn.edu

Yunhuai Liu
Third Research Institute of
the Ministry of Public Security
Shanghai, China
yunhuai.liu@gmail.com

Guang Tan
SIAT
Chinese Academy of Sciences
Shenzhen, China
guang.tan@siat.ac.cn

Tian He
University of Minnesota
Minneapolis, MN, USA
tianhe@cs.umn.edu

ABSTRACT

Wireless communication essentially occurs in a broadcast medium with concurrent receptions. Recent works [34,41] have shown clear evidence that wireless links are not independent and that transmissions from a transmitter to multiple receivers are correlated, a phenomenon that has profound implications for the performance of network protocols such as broadcast, multi-cast, opportunistic forwarding and network coding. In this paper, we show 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 our work is CorLayer's broad applicability and effectiveness. Our system effort is indeed significant. We integrate CorLayer transparently with sixteen state-of-the-art broadcast protocols specified in thirteen publications [1, 3, 18, 19, 23, 25–27, 32, 36, 38–40] 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.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communications

General Terms

Design, Experimentation, Algorithms, Performance

MobiCom'13, September 30–October 4, Miami, FL, USA.

http://dx.doi.org/10.1145/2500423.2500425.

Keywords

Energy Efficiency, IEEE 802.15.4, Wireless, Protocol

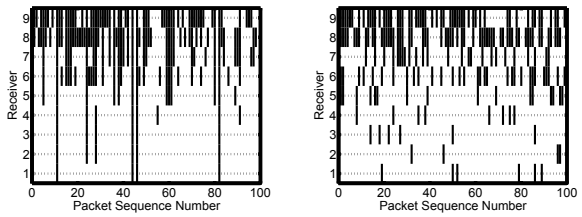
1. INTRODUCTION

Link correlation refers to the phenomenon in which packet receptions from a transmitter to multiple receivers may be correlated. As recently highlighted by Srinivasan et al. [34] and Zhu et al. [41], 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.

For example, suppose two receivers have a high correlation, and thus a reception acknowledgement from the first receiver strongly implies that the second one receives the packet as well. Statistically, the second acknowledgement is not quite necessary. This insight has been successfully used by Zhu et al. [41]. They proposed a concept called collective ACKs which solves the ACK storm problem [30] by inferring the success of a transmission to a receiver based on the ACKs from other neighboring receivers. More generally, 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 [27, 32], multi-cast [12, 16], and multi-path routing [13], or (ii) diversity-based protocols such as opportunistic forwarding [5, 7, 10], collaborative forwarding [6], and network coding [2, 21, 25].

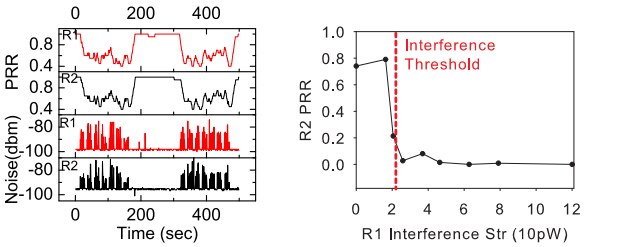
In this paper, we focus on how to exploit link correlation to improve the energy efficiency of *reliable broadcast protocols* that try to guarantee every node in the network receives packets. Reliable broadcast is a fundamental operation in wireless network design and plays a critical role in many other operations such as code dissemination (e.g., Deluge [17] and MNP [22]), content delivery (e.g., Cabernet [11]), and routing discoveries (e.g., ODRMP [24] and OLSR [8]).

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 [9]);



(a) Empirical Trace (b) Synthetic Independent Trace

Figure 1: Packet receptions at 9 receivers when a single transmitter broadcasts 100 packets; (a) empirical trace from the testbed in UMN on channel 16. (b) Synthetic trace with the independent links under the same packet reception ratio. Packet losses are marked by black bends. Compared with synthetic independent trace, empirical trace has many more long vertical black bands, indicating that the empirical links have more correlated losses.



(a) PRR & Noise RSSI (b) R2's PRR vs. R1's Interference

Figure 2: Cross-network interference; (a) two receivers' PRR and the corresponding noise level. Both receivers' PRR decreases when the noise is introduced by the microwave oven. (b) R2's PRR under R1's interference strength. We can tell R2's PRR from R1's interference strength – R2's PRR decreases when R1's interference is above a certain level.

(ii) cluster-based (e.g., passive clustering [23]); (iii) multi-point relays (e.g., OLSR [8]); (iv) pruning-based (e.g., RNG relay subset [18] and partial dominating pruning [27]), (v) location-based (e.g., curved convex hall [38]). 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, our contributions are as follows:

• Although the phenomenon of link correlation has been mentioned in the literature [34, 41], 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 our 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 [1, 3, 18, 19, 23, 25–27, 32, 36, 38–40], ranging from 20% $\sim$ 64%.

The rest of our paper is organized as follows. Section 2 presents the motivation of our design. The main design is presented in Section 3. Sections 4 and 5 report testbed and simulation experiments. In Section 6, we review related work. Finally, Section 7 concludes the paper.

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.

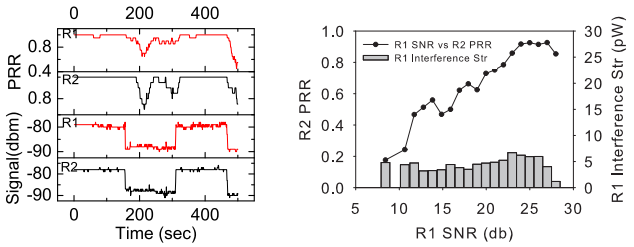
2.1 Existence of Link Correlation

To verify the existence of link correlation, 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.

Figure 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 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 1(a) are indeed correlated. We note that this experiment reaffirms the empirical study reported in recent works [34, 41].

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

• Cross-Network 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 2.4 GHz ISM band, leading to possible cross-network 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) PRR & Signal RSSI (b) R2's PRR vs. R1's SNR

Figure 3: Correlated fading; (a) two receivers' PRR and signal RSSI. Both receivers' PRR decreases when signal RSSI decreases because of shadow fading. (b) R2's PRR under R1's SNR. There exists a near-linear relationship between R2's PRR and R1's SNR when R1' interference is under threshold.

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(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(b) shows PRR of R2 relative to interference power observed at R1. From the figure, we see that R2's PRR faces a steep decrease when interference at R1 is above a certain level, which we call interference threshold (indicated by the dashed line), where almost no reception can be made on R2 upon crossing it.

• 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 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 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 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 R1's SNR and R2's PRR, as shown in Figure 3(b). In other words, fading correlation induces approximately linear relationship between the quality of R1's link and reception probability of R2. The bar graph (Figure 3(b)), which represents interference power, indicates that interference does not affect the relationship between R1's SNR and R2's PRR when it is under threshold.

2.2 How Link Correlation Affects Broadcast

This section theoretically analyzes the impact of link correlation upon the energy efficiency of broadcasting. Table 1 summarizes some notations that will be used in this paper. We demonstrate that a node incurs *fewer* transmissions when the packet receptions of its one-hop neighbors have a *higher* correlation. Figure 4 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 4(a) should have fewer transmissions because the link quality (80%) in this cluster is higher than that in Figure 4(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

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

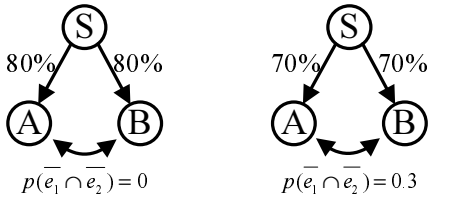
Then the expected number of transmissions $\mathrm{E}[\epsilon]$ to cover two children nodes can be calculated as

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

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 4. For the cluster in Figure 4(a), the expected number of transmissions $\mathrm{E}[\epsilon]$ is 1.5 based on Eq.(1),

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 1: Notations used in this paper



(a) Negative correlated (b) Positive correlated

Figure 4: Impact of link correlation on broadcast. The expected number of transmissions to cover two receivers is (a) 1.5, and (b) 1.43. Because of the high link correlation, case (b) needs less transmissions although its link quality is worse than case (a).

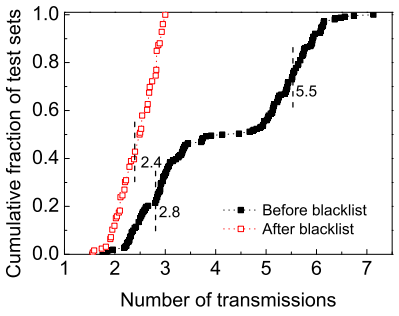


Figure 5: The effect of blacklisting on transmissions. The average number of transmissions before blacklisting is mainly concentrated around 2.8 and 5.5 while it's 2.4 after blacklisting. Blacklisting leads to a significant reduction in transmission.

given that $p(\bar{e_1} \cap \bar{e_2}) = 0$ since SA and SB are perfectly negative correlated. In Figure 4(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, $\mathrm{E}[\epsilon]$ is 1.43 which is less than the cluster in Figure 4(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.

2.3 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 5 shows the transmission CDF before and after blacklisting. From Figure 5, 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.

3. THE DESIGN OF CORLAYER

Since a wireless network has a much more complex structure than a star topology as we describe in Section 2.3, 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 3.1. We then introduce the basis of CorLayer – the expected transmission count of broadcast – in Section 3.2. To support efficient calculation, we propose a heuristic to reduce computational cost in Section 3.3. The general design is presented in Section 3.4.

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 4, 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.

3.2 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 2.2, 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 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.

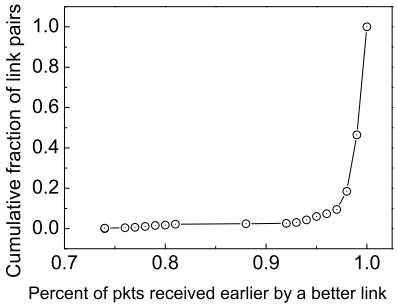


Figure 6: Statistics of receiving probability. The node with a better link receives about 98% of the packets earlier or at the same time than the node with a worse link.

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 [41]. 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 6 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.,*

$$Pr(v|K(u)) = \frac{p(K(u) \cap v)}{p(K(u))}. \quad (2)$$

We note that our 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 [34, 41]. With these concepts, we are able to approximate the cost that a node delivers a message to its one-hop neighbors.

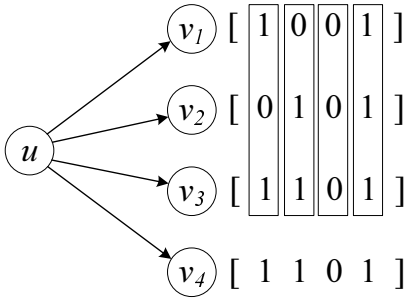


Figure 7: Example of Calculating u's JPRP for $\{v_1, v_2, v_3\}$

LEMMA 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*

$$\hat{\epsilon}(u) = \sum_{i=1}^{M} \frac{1}{p(e_i)} - \sum_{i=2}^{M} \frac{1}{p(e_i)} \cdot \frac{p(K_i(u))}{p(K_{i-1}(u))} \quad (3)$$

PROOF.

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

□

Special Case: Note that Eq.(3) 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

$$\hat{\epsilon}(u) = \sum_{i=1}^{M} \frac{1}{p(e_i)} - M + 1. \quad (4)$$

THEOREM 3.2. *The higher the set link correlation, the smaller the expected number of transmissions $\hat{\epsilon}(u)$.*

PROOF. From Eq.(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 our 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 message is identified by

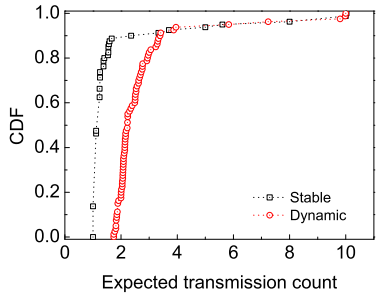


Figure 8: CDF of the expected transmission count – ϵ in stable and dynamic scenarios.

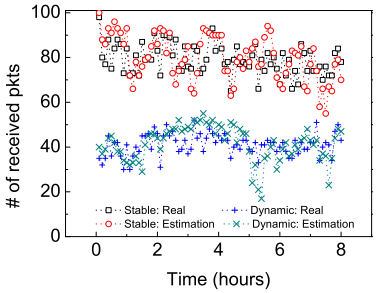


Figure 9: Received pkts vs. Estimation

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

$$p(K_i(u)) = \frac{1}{W}\sum_{k=1}^{W} B_{v_1}(k)\&B_{v_2}(k)\&\ldots\&B_{v_i}(k), \quad (5)$$

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 7, 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.(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 our 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 our setting. Note that periodically sending probe packets has been already required by other protocols [4, 14] to measure link quality or to improve the robustness of routing structure. We argue such cost is not introduced solely by our design, hence it should be amortized. Second, we need to exchange packet reception bitmaps among one-hop neighbors in order to calculate the values of set link correlation, which is exclusively introduced by our 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 our measurements.

Estimation Accuracy under Link Dynamics: We run our 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 8 and 9, we maintain the estimation of expected number of transmission ϵ accurately over time. Specifically, Figure 8 shows the CDF of ϵ, from which we can see that the metric ϵ is stable. Figure 9 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.

3.4 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 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 our 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 3.5 and Section 3.6.

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. Our 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 [20] 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.

3.6 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

Protocol Name	Reference	Network Info.	Probe Msg	Broadcast Msg	Category
Spanning Tree (S-Tree)	[19]	One-hop	ID	Msg only	Tree-based
Cluster Tree (C-Tree)	[3]	Quazi-Global	Global	Msg only	Tree and Cluster-based
Forwarding Node Cluster (Cluster)	[40]	Local	ID	Covered set	Tree and Cluster-based
Intermediate	[39]	Two-hop/One-hop	One-hop/Position	Msg only	Cluster-based
Clustering	[36]	Quazi-Local	Degree	Msg only	Cluster-based
Passive Clustering (P-Clustering)	[23]	Two-hop	None	Msg + 2 bits	Cluster-based
Multi-Point Relay (MPR)	[32]	Two-hop	One-hop	Msg + Covered set	Multi-point relay
Mini-id MPR	[1]	Two-hop	One-hop	Msg only	Multi-point relay
MPR-CDS	[1]	Two-hop	One-hop	Msg only	Multi-point relay
Self Pruning (SP)	[26]	One-hop	One-hop	Msg + Covered set	Pruning-based
Dominating Pruning (DP)	[26]	Two-hop	One-hop	Msg + Covered set	Pruning-based
Pruning	[27]	Two-hop	One-hop	Msg + Covered set	Pruning-based
TDP	[27]	Two-hop	One-hop	Msg + Covered set	Pruning-based
RNG Relay Subset (RNG)	[18]	Two-hop	One-hop	Msg only	Pruning-based
Curved Convex Hull (CCH)	[38]	One-hop	One-hop + Position	Msg only	Location-based
CODEB	[25]	Two-hop	One-hop	Msg + Covered set	Pruning-based+NC

Table 2: Sixteen state-of-the-art protocols supported through CorLayer embedding

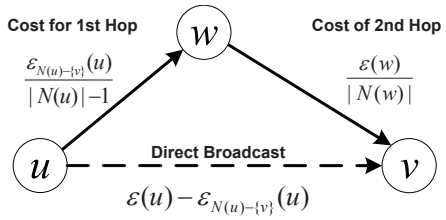


Figure 10: Triangular Blacklist Rule

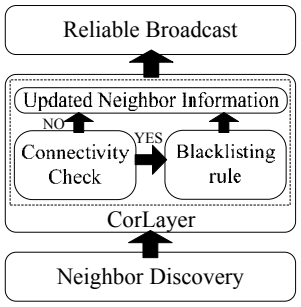


Figure 11: CorLayer in the protocol stack

number of transmissions after a link is blacklisted. As shown in Figure 10, 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, our 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:

$$\epsilon(u) - \epsilon_{N(u)-\{v\}}(u) > \frac{\epsilon_{N(u)-\{v\}}(u)}{|N(u)|-1} + \frac{\epsilon(w)}{|N(w)|} \qquad (6)$$

The left-hand part of Eq.(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 only the neighbor set $N(u) - \{v\}$ (denoted as directed broadcast cost in Figure 10).

Intuitively, the link $u \rightarrow v$ 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.(6). Specifically, the right-hand part of the Eq.(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 $u \rightarrow v$ 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 10 as "cost for 1st hop" and "cost for 2nd hop".

In summary, Eq.6 presents a *Triangular Blacklist Rule* shown in Figure 10: 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; our 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.

We note 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.6 shall be revised to a more generic one, which is used in our final design.

$$\epsilon(u) - \epsilon_{N(u)-\{v\}}(u) > \frac{\sum \frac{\epsilon_{N(u)-\{v\}}(u)}{|N(u)|-1} + \frac{\epsilon(w_i)}{|N(w_i)|}}{|N(u,v)|} \qquad (7)$$

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) [31]. To do that, we insert CorLayer beneath the broadcast protocol and above the MAC layer. As shown in Figure 11, CorLayer provides a reduced topology to the upper layer broadcast protocols in a transparent manner.

We classify the existing reliable broadcast algorithms into tree-based [9, 19], cluster-based [3, 23, 36, 39, 40], multi-point relays [1, 8, 32], pruning-based [18, 26, 27], and location based [38]. Recently, network coding (NC) has been adapted to support broadcast appli-

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

Figure 12: 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: Testbed settings and topology properties

cations in wireless networks, e.g., COPE [21] and CODEB [25]. 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 2.

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. 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 12(a). The second testbed consists of 30 TelosB nodes deployed in an $18m \times 13m$ open office environment following a grid pattern; see Figure 12(b). The third testbed is an outdoor environment (Figure 12(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 3.6). 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.

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.

4.2 Main Performance Results

The experimental results of the sixteen classical reliable broadcast protocols are shown in Figure 13. The upper parts of the bars (in gray) represent the proportion of transmissions reduced by our 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. Our design makes a further 39.3% improvement upon CODEB. On average, our blacklist design reduces transmissions by 47.2%. The reason why our 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 [32] (MPR for short), Forwarder Node Cluster [40] (Cluster for short), Partial Dominating Pruning [27] (Pruning for short), and CODE-B [25] for the rest of the experiments.

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 4 for a

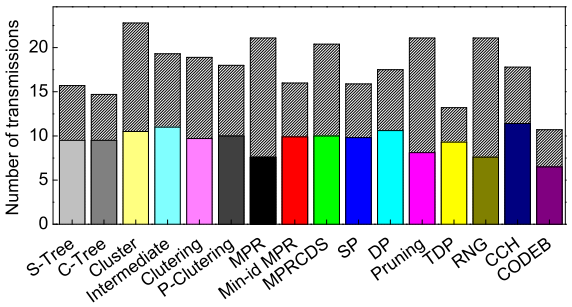


Figure 13: Improvements over 16 protocols

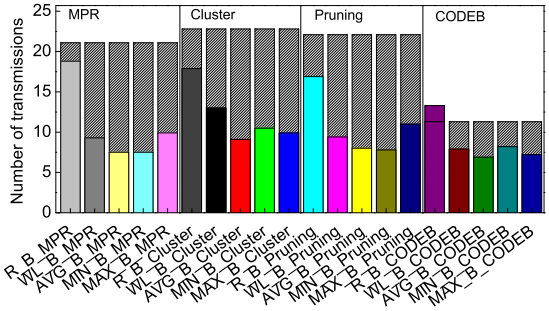
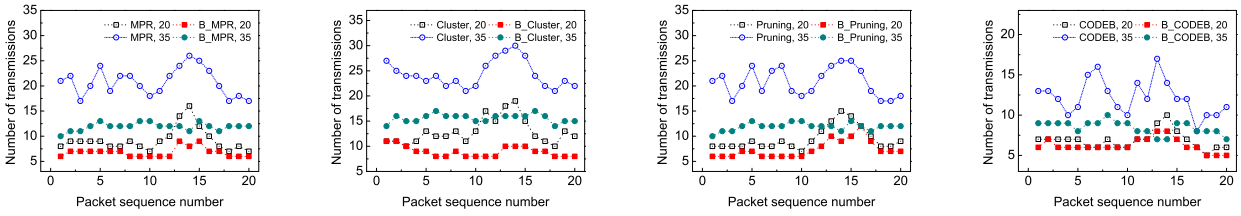


Figure 14: Impact of blacklist rules

Method	Blacklist Rule
Random Blacklist (R_B)	Each node blacklists $x\%$ of links ($x = 10$ in our experiment).
Worst Link Blacklist (WL_B)	Each node blacklists the worst $x\%$ of links ($x = 10$ in our experiment).
AVG Blacklist (AVG_B)	Our design, please refer to Eq.(7).
MIN Blacklist (MIN_B)	A link is blacklisted when there exists one alternative path with lower cost.
MAX Blacklist (MAX_B)	A link is blacklisted when its broadcast cost is higher than all alternative paths.

Table 4: Different link blacklisting strategies.



(a) Multi-point relays (b) Cluster based broadcast (c) Pruning based broadcast (d) Network coding

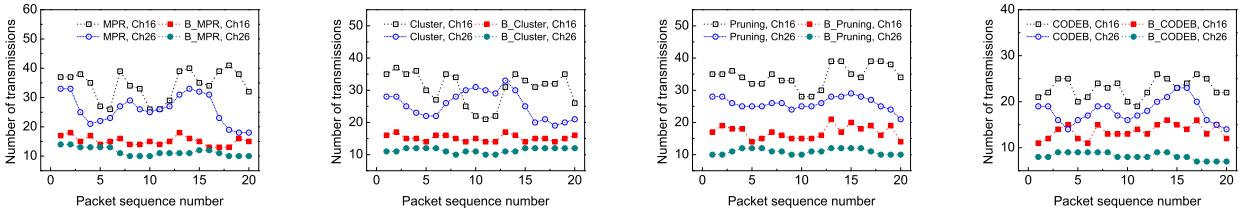
Figure 15: Impact of network sizes

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 14 shows the energy consumption of the four broadcast strategies with the five different blacklisting rules. On average, our 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 our design. Yet, in the cluster algorithm, the energy consumption of "MIN blacklisting" is 10.5, while our 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 14, 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.

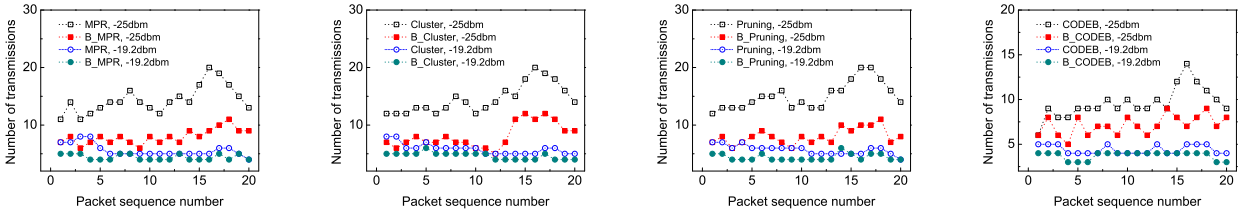
4.4 Impact of Network Size

In this experiment, we use the data from the testbed in UMN. Figure 15 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, our design obtains an energy gain of 31.3%. From Figure 15, we can also see that the trend of energy gain with increasing network sizes is quite stable, suggesting that our design scales well with large networks.



(a) Multi-point relays (b) Cluster based broadcast (c) Pruning based broadcast (d) Network coding

Figure 16: Impact of channels



(a) Multi-point relays (b) Cluster based broadcast (c) Pruning based broadcast (d) Network coding

Figure 17: Impact of power levels

4.5 Impact of Different Channels

In this experiment, we explore the impact of channels on our 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 16 shows the energy consumption of the eight broadcast strategies for networks using two different channels – channels 16 and 26. From Figure 16, 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 of our design under the four broadcast algorithms is 56% using channel 26, and 50% using channel 16. This result shows that our 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. [34]. The better correlation thus leaves us less room to improve the energy efficiency since the gain relies on the improvement of link correlation.

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 12(c). The power level for transmission is set from -25dBm to -19.2dBm to form a multi-hop network. Figure 17 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, our link blacklisting layer still reduces transmissions by 20.1%.

5. SIMULATION EVALUATION

Adjusting the power level 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.

5.1 Impact of Link Quality

First, let us consider the energy gain of the four link blacklisting strategies 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 [28] that makes the one-hop receivers drop packets in a link correlated way with a controlled loss rate. The results are shown in Figure 18, where we can see that the transmission 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.

5.2 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. Figures 19 shows the transmission counts of the eight broadcast strategies for networks with different

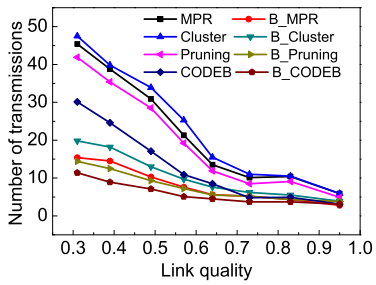


Figure 18: Impact of ink qualities

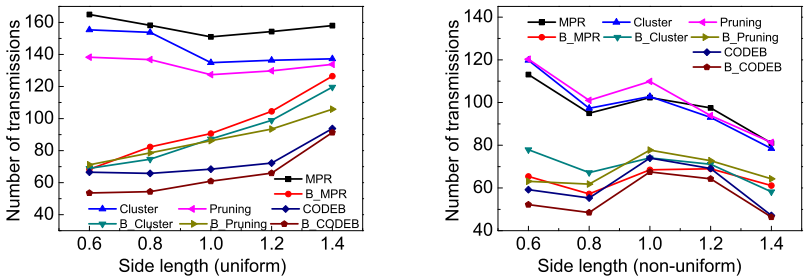


Figure 19: Impact of densities

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 Figures 19, 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 Figures 19, the energy gain of our design decreases as the side length increases (and thus the density decreases). For example, in uniform case, 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. This explains the increasing energy gain when the node density grows.

6. STATE OF THE ART

Our design leverages a recently discovered phenomenon, link correlation, to improve the performance of reliable broadcasting. In the following, we briefly review related work.

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 [30]. The existing methods that address this problem can be divided into two categories: probabilistic and deterministic [37]. In probabilistic methods [15, 35], 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 paper, 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 [19], the authors present a fully distributed, online, and asynchronous method to maintain a spanning tree, along which the broadcast is performed. Ding et al. [9] present a tree-based reliable broadcast for IEEE 802.15.4 and ZigBee networks.

• **Cluster-based:** The authors in [3] 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 [39] introduce the concept of intermediate node to calculate a CDS for reliable broadcast.

• **Multi-Point Relays:** In MPR [32], 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 [1] 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 [26], 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 [26], Lou et al. further propose two extended algorithms: total dominating pruning (TDP) and partial dominant pruning (PDP) [27], which generate a smaller relay node set than the DP and SP algorithms do.

• **Location-based:** The authors in [38] 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 [21] and CODEB [25] 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 [4, 29]. The phenomenon of link correlation has recently been experimentally studied in [33, 41]. Zhu et al. [41] propose a probabilistic flooding algorithm to reduce energy consumption in transmission by using implicit ACKs inferred from link correlation. In [34], 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, our work is different from the aforementioned works.

7. CONCLUSION

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.

8. ACKNOWLEDGMENTS

This work was supported in part by National Science Foundation grants CNS-0917097. We also received partial support from McKnight Land-Grant Professorship. We thank our shepherd Dr. Prabal Dutta and the reviewers for their valuable comments.

9. REFERENCES

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

[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 2000.

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

[4] N. Baccour, A. Koubaa, L. Mottola, M. Zuniga, H. Youssef, C. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM TOSN*, 2012.

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

[6] 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.

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

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

[9] 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.

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

[11] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: vehicular content delivery using wifi. In *MOBICOM*, 2008.

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

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

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *SENSYS*, 2009.

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

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

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

[18] 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.

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

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

[21] 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.

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

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

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

[25] 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.

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

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

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

[29] 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*, 2008.

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

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

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

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

[34] 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.

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

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

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

[38] 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. *Computer Science - Networking and Mobile Computing*, 2005.

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

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

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