# Encode When Necessary: Correlated Network Coding Under Unreliable Wireless Links

SHUAI WANG, ZHIMENG YIN, and TIAN HE, University of Minnesota
SONG MIN KIM, George Mason University

Recent research has shown that network coding has great potential to improve network performance in wireless communication. The performance of network coding in real-world scenarios, however, varies dramatically. It is reported that network coding brings negligible improvements but extra coding overhead in some scenarios. In this article, 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 performance improvement is achieved. 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 we evaluate them extensively on one 802.11 testbed and three 802.15.4 testbeds. The experimental results show that (i) more coding operations do not lead to fewer transmissions, and (ii) compared to existing network coding protocols, the number of transmissions is reduced with lower coding overhead.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, wireless networks, network coding, link correlation

**ACM Reference Format:**
Shuai Wang, Zhimeng Yin, Tian He, and Song Min Kim. 2017. Encode when necessary: Correlated network coding under unreliable wireless links. ACM Trans. Sen. Netw. 13, 1, Article 7 (January 2017), 22 pages.
DOI: http://dx.doi.org/10.1145/3023953

## 1. INTRODUCTION

Network coding, originally proposed by Ahlswede et al. [2000], has great potential to improve network performance (e.g., the throughput and energy efficiency) in wireless communication [Katti et al. 2006; Chachulski et al. 2007; Keshavarz-Haddad and Riedi 2008]. For example, under the lossy wireless 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 transmission efficiency.

Although some researchers are 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 and that

---

This work was supported by the National Science Foundation under grant CNS-1444021.

Authors' addresses: S. Wang, Z. Yin, and T. He (corresponding author), Department of Computer Science, University of Minnesota, 200 Union St SE, Minneapolis, MN 55455 USA; emails: {shuaiw, zhiyin, tianhe}@cs.umn.edu; S. Kim, Department of Computer Science, George Mason University, 4400 University Drive MSN 4A5, Fairfax, VA 22030 USA; email: song@gmu.edu.

DOI: http://dx.doi.org/10.1145/3023953

the coding costs may exceed the benefits. In real-world scenarios, this situation occurs. For example, consider the extreme case of when wireless links are perfectly positively correlated, network coding will not provide any improvement but coding overhead, as all receivers lose the same packets and there are no diversity benefits to exploit.

In this article, we present link correlation—a concept that captures the relationship among packet receptions. We investigate the impact of link correlation upon network coding. In further 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 [Katti et al. 2006] 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 transmissions to deliver a coded packet to all receivers when the receptions of these nodes are correlated [Guo et al. 2011]. 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 underutilized diversity benefits [Katti et al. 2006; Srinivasan et al. 2010]. Even worse, because of the inaccurate link independent assumption, unnecessary coding operations exploited by coding-aware routings [Sengupta et al. 2010; Le et al. 2010; Wang et al. 2013b] may lead to extra energy consumption and delay.

We thus propose correlated coding, a coding technique 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 called *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 experimental results show that our design consistently enhances the performance of these protocols—the number of coding operations is reduced, whereas the transmission efficiency is improved by 30% to 50%.

The rest of the article is structured as follows. Section 2 reviews related work. Section 3 presents our motivation for this work. Section 4 introduces the main design, followed by its applications in Section 5. Evaluation results from testbed experiments and simulations are shown in Sections 6 and 7. Finally, Section 8 concludes the article.

## 2. RELATED WORK

Two main bodies of study—network coding and link correlation—are closely related to our work. In the following, we first summarize existing works and then state the unique position of our own.

### 2.1. Network Coding

Network coding [Ahlswede et al. 2000], which allows intermediate nodes to combine packets before forwarding, has great potential to improve transmission efficiency of both broadcast and unicast. Katti et al. proposed the first practical network coding scheme, called *opportunistic coding* (also known as COPE type network coding) [Katti et al. 2006]. 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 transmission efficiency. Opportunistic coding does not fully exploit the benefits of network coding, as the coding opportunity is dependent on the routing path, and the coding-unaware routing strategy [Katti et al. 2006] misses many coding opportunities. Researchers thus propose coding-aware routings [Sengupta et al. 2010; Le et al. 2010; Wang et al. 2013b] to exploit more coding opportunities.

Whereas some researchers are optimistic about the decent performance of network coding, others point out that the improvement of network coding is marginal and that the coding costs could be extremely high. The empirical study shows that the benefits of network coding change dynamically under different testbed measurements [Srinivasan et al. 2010]. The coding opportunity highly depends on the packet reception information as well as the routing path construction, and its coding benefits could be marginal [Sengupta et al. 2010; Atya et al. 2013].

### 2.2. Link Correlation

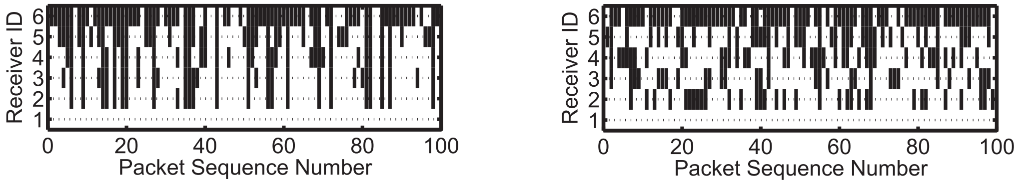
Most existing studies in wireless networks focus on individual links or path qualities; however, Srinivasan et al. and Zhu et al. reveal that the receptions of broadcasting packets at multiple receivers are correlated [Srinivasan et al. 2010; Zhu et al. 2010]. Srinivasan et al. [2010] explore a metric called $\kappa$ that captures the degree of packet reception correlation on different links. Zhu et al. [2010] solve the ACK storm problem by using implicit ACKs inferred from link correlation. Guo et al. [2011] propose a flooding design for low-duty-cycle sensor networks, in which nodes with high correlation wake up at the same time slots and are assigned to a common sender to save transmissions. Wang et al. [2013a] propose an independent layer called *CorLayer*, which improves a wide range of broadcast protocols by eliminating certain poorly correlated wireless links and forming better-correlated clusters. Whereas most previous works measure link correlation using packet-level transmissions and receptions, Zhao et al. [2015] proposed a model based on the underlying causes of link correlation. They explore four easily measurable parameters—received signal strengths, noise and interference, the packet length, and the packet transmission interval—for link correlation modeling.

### 2.3. This Work

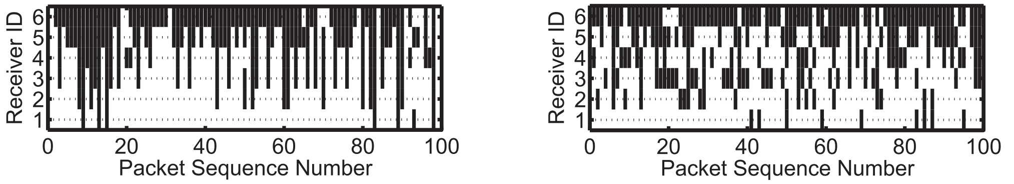
In prior works, researchers explicitly or implicitly assume that wireless links are independent [Katti et al. 2006; Chachulski et al. 2007; Li et al. 2007; Chaporkar and Proutiere 2007; Sengupta et al. 2010; Atya et al. 2013] when exploiting network coding benefits. This assumption, however, contradicts the empirical evidence that wireless links are correlated [Srinivasan et al. 2010; Zhu et al. 2010; Wang et al. 2013a, 2014]. 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 to the widely used link independence model, our model is more practical and accurate. With the quantified coding benefits, our approach helps network designers decide whether to apply the network coding technique or not. In addition, our model can be further used to optimize the coding benefits when network coding is applied.

## 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) 802.15.4: Empirical trace (b) 802.15.4: Synthetic independent trace



(c) 802.11: Empirical trace (d) 802.11: Synthetic independent trace

Fig. 1. Packet receptions at six receivers when a single transmitter broadcasts 100 packets. (a) and (c) show the empirical trace from the 802.15.4 and 802.11 testbeds. (b) and (d) show the corresponding synthetic trace with the independent links under the same packet reception ratio. Packet losses are marked by black bends. Compared to synthetic independent trace, empirical trace has more correlated losses.

### 3.1. The Existence of Link Correlation

Today, more and more devices with the same/different wireless technologies operate in unlicensed/open spectrums (e.g., ISM bands), generating massive intratechnology/cross-technology interference. These interference sources cause simultaneous packet losses (i.e., correlated losses) at those nearby devices. To verify the existence of link correlation, we conduct experiments on both 802.15.4 and 802.11 testbeds. In both experiments, seven nodes are deployed to form a star topology. The central node is selected as the sender, and the other six nodes are receivers. The sender broadcasts 100 packets to the receivers. Each packet is identified by a sequence number. On the 802.15.4 testbed, we use channel 16, which overlaps with the Wi-Fi signals. On the 802.11 testbed, channel 6 is utilized. After the broadcast task, a sink node collects the reception results of the receivers.

Figure 1(a) and (c) show the packet receptions at the six receivers from empirical measurements. The lost packet is marked by black bands, and long vertical black bands indicate that packets are lost at multiple receivers. As a comparison, Figure 1(b) and (d) plot the independent synthetically generated traces with the same PRR where few multiple simultaneous receptions and losses are observed. The comparison indicates that the packet receptions at multiple receivers in Figure 1(a) are correlated.

The statistical results affirm the existence of link correlation as well. Figure 2 shows the statistics of the conditional probability $Pr(e_H|\{e_L\})$, where $\{e_L\}$ denotes an arbitrary combination of links whose qualities are lower than that of $e_H$, on the 802.11 and 802.15.4 testbeds. In the figure, the black square represents the $Pr(e_H|\{e_L\})$ value calculated with the synthetic independent trace, whereas the red cross represents the $Pr(e_H|\{e_L\})$ value calculated based on the packet reception information of the empirical trace. From Figure 2(a) and (b), the $Pr(e_H|\{e_L\})$ value based on the empirical trace is higher than $Pr(e_H|\{e_L\})$ calculated with the synthetic independent trace, indicating the existence of a positive link correlation. The observation contradicts the widely made link independence assumption in existing network coding designs [Chachulski et al. 2007; Katti et al. 2006; Li et al. 2007; Sengupta et al. 2010].

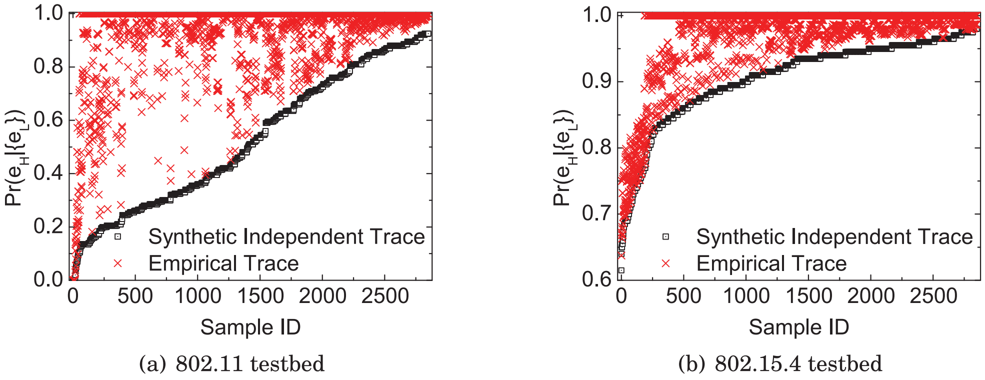


Fig. 2. Statistics of receiving probability on 802.11 and 802.15.4 testbeds.

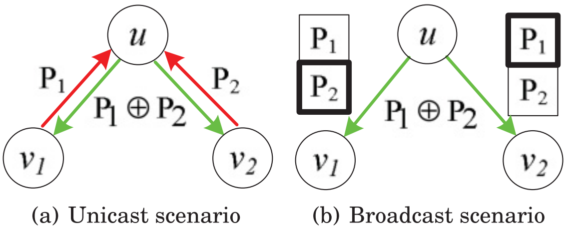


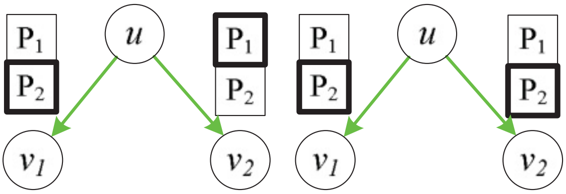
Fig. 3. Examples of network coding occurring in the unicast scenario (a) and the broadcast scenario (b). From both scenarios, network coding obtains a benefit because it broadcasts a coded packet that is involved with multiple original packets.

### 3.2. 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 3 shows how network coding benefits both unicast and broadcast. In Figure 3(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 3(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 receivers $v_1$ and $v_2$ lose packets $p_1$ and $p_2$, respectively. In traditional designs, to ensure 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 a XORed packet $p_1 \oplus p_2$ using one transmission, thus saving one transmission. From the examples in Figure 3, 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.

### 3.3. Impact of Link Correlation on Network Coding

From the preceding discussion, we learn that network coding obtains benefits by 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(b),



(a) Coding scenario (b) Noncoding scenario

Fig. 4. Impact of link correlation on network coding. In scenario (a), the links are negative correlated and coding opportunity occurs. In scenario (b), the links are positive correlated and there are no coding opportunities. The expected number of transmissions to deliver the coded packet to the two receivers is 3(a) and 2(b).

both receivers lose the same packet $p_1$ (i.e., links $uv_1$ and $uv_2$ are positive correlated). There are no coding opportunities in this scenario. Compared to the coding scenario in Figure 4(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 that a coded packet from the source node is not received by either receiver. Then the expected number of transmissions $\mathrm{E}[\varepsilon]$ to deliver the coded packet to the two receivers can be calculated as

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

The proof of Equation (1) is presented in the Appendix. We note that $p(\bar{e_1} \cap \bar{e_2})$ obtains its maximum value when links $e_1$ and $e_2$ are perfectly positively correlated, whereas it gets a minimum value when links $e_1$ and $e_2$ are perfectly negatively correlated. Let us revisit the two scenarios in Figure 4(a) and (b). For the scenario in Figure 4(a), the expected number of transmissions $\mathrm{E}[\varepsilon]$ is 3 based on Equation (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 negatively correlated. In Figure 4(b), however, since $uv_1$ and $uv_2$ are perfectly positively correlated, it is not difficult to get that $p(\bar{e_1} \cap \bar{e_2})$ is 0.5. As a result, $\mathrm{E}[\varepsilon]$ is 2, which is less than the cluster in Figure 4(a). This suggests that in the broadcast procedure, positive correlation is preferred since all receivers lose the same packets and few retransmissions are needed. Therefore, there exists a trade-off between the number of coding opportunities and 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. 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 4.1) and coding opportunity (Section 4.2). We then propose unified

Table I. Notations Used in This Article

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

metrics dealing with the trade-off between broadcast efficiency and coding opportunity (Section 4.3). Some notations used in this article are listed in Table I.

### 4.1. Broadcast Efficiency Analysis

We first examine the number of transmissions for node $u$ to reliably broadcast the coded packet to all potential receivers such that they can extract the original packet from the coded one. We denote $\varepsilon$ 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 that $p(e_1) \geq p(e_2) \geq p(e_3) \geq \ldots \geq p(e_K)$. The expectation of $\varepsilon$ can be calculated as

$$
\begin{aligned}
\mathrm{E}[\varepsilon] = & \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{1 \leq i < j \leq K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j})} + \sum_{1 \leq i < j < l \leq K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j} \cap \bar{e_l})} + \ldots \\
& + (-1)^{K-1} \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2} \cap \ldots \cap \bar{e_K})}.
\end{aligned}
\tag{2}
$$

The proof of Equation (2) is presented in the Appendix. To get $\varepsilon$ 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.

*4.1.1. Transmission Count Approximation.* Due to the high cost of computing $\varepsilon$, we seek a more efficient algorithm to approximate $\varepsilon$ 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 every 0.2 seconds until all receivers receive 100 packets. In each packet, we include a sequence number and timestamp. After collecting the packet reception trace, we compare the reception between each link pair (there are $\binom{30}{2} = 435$

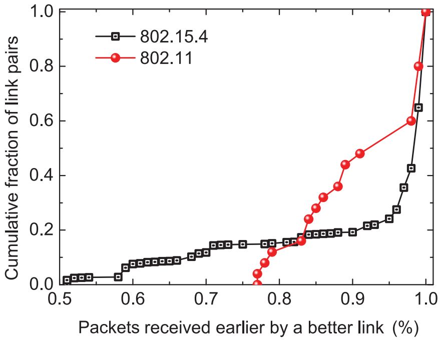


Fig. 5. Statistics of receiving probability. The node with a better link receives about 92.7% (802.15.4 testbed) and 92.1% (802.11 testbed) of the packets earlier or at the same time than the node with a worse link.

such pairs). Figure 5 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 the 802.15.4 and 802.11 testbeds. 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 $\varepsilon$, denoted as $\hat{\varepsilon}$.

*Approximation of* $\varepsilon$. Assuming that 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

$$\hat{\varepsilon} = \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))}, \tag{3}$$

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 $i$ nodes in $S_i(u)$ successfully receive a packet. The proof of Equation (3) is presented in the Appendix. The computational complexity of $\hat{\varepsilon}$ is $O(K^2)$, where $K$ is the number of receivers.

*A special case*. Note that Equation (3) includes the special case that the links are independent. When links are all independent, we have $p(S_i(u)) = p(e_i) \cdot p(S_{i-1}(u))$. Additionally, $\hat{\varepsilon}(u)$ reduces to $\hat{\varepsilon} = \sum_{i=1}^{K} \frac{1}{p(e_i)} - K + 1$.

*Implementation*. To calculate $\hat{\varepsilon}$, we need to get each receiver's link quality $p(e)$ and $p(S_i(u))$ for $\forall i = 2, \ldots, K$. Suppose that each receiver maintains a packet reception report (e.g., [1100]) recording the reception status of a fixed number (e.g., 4) of most recent packets. In our design, we use normal data traffic to update the reception report. With the reception report, the link quality $p(e)$ is given simply by the number of 1's in the reception report divided by the reception report length. The calculation of $p(S_i(u))$ needs each receiver's packet reception report, whose input is the same as the input of calculating $p(e)$. Let the reception report length be $W$, and we have

$$p(S_i(u)) = \frac{1}{W} \sum_{j=1}^{W} B_{v_1}(j) \& B_{v_2}(j) \& \ldots \& B_{v_i}(j), \tag{4}$$

where $B_{v_i}(j)$ is a bit representing receiver $v_i$'s reception status of the $j$th packet. $B_{v_i}(j) = 1$ represents that node $B_{v_i}(j)$ receives the packet, and otherwise $B_{v_i}(j) = 0$. For example, in Figure 6, node $u$ has three neighbors. We calculate $u$'s $p(S_i(u))$ for $\{v_1, v_2\}$. Suppose that the report of node $v_1$ is [1100], which indicates that $v_1$ receives the first and

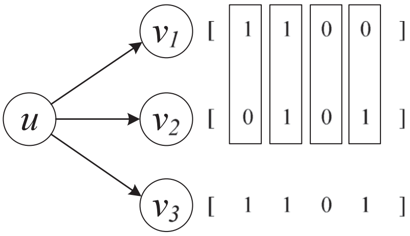
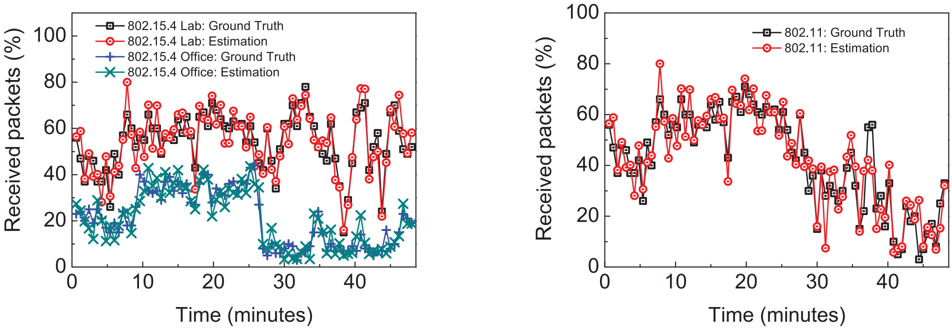


Fig. 6. Example of calculating $u$'s $p(S_i(u))$ for $\{v_1, v_2\}$.



(a) 802.15.4 testbed (b) 802.11 testbed

Fig. 7. Estimation accuracy: received packets versus estimation.

second packets and loses the third and fourth packets. When node $u$ receives the reception reports from the receivers, it uses Equation (4) to calculate $p(S_i(u))$ (i.e., $p(S_2(u)) = (1\&0 + 1\&1 + 0\&0 + 0\&1)/4 = 25\%$).

*4.1.2. 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 the 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 a 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 to calculate $\hat{\varepsilon}$. The exchange of reception reports has already been required by network coding schemes (e.g., COPE) [Katti et al. 2006]. 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. In addition, the binary report is small and much less frequently exchanged.

In the experiments, the source node keeps broadcasting packets to six receivers every 0.3 seconds. The reception report is sent every 30 seconds to freshen link quality and correlation values. We run the 802.15.4 experiments in 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 7. Here, the number of received packets (during a time period of 30 seconds) 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.

---

[1] Detailed information on the testbed scenarios is described in Section 6.

In the preceding experiment settings (i.e., the reception report is sent every 30 seconds), the cost of exchanging reception reports occupies a tiny fraction (1%) of the total energy cost. Thus, the estimation of $\varepsilon$ will not bring much overhead to the existing protocols.

### 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 of its receivers, we need to ensure that all receivers have already gathered enough packets to decode the original one. We specify the network coding rule as follows.

*Definition* 4.1 (*Network Coding Rule*). Consider a sender $u$ transmitting an encoded packet $p' = \oplus(p_1, p_2, \ldots, p_K)$. To decode $p'$, each receiver should have already received $K - 1$ packets among $p_i$, $i = 1, 2, \ldots, K$.

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

*Definition* 4.2 (*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 that the number of coding opportunities with $i$ original packets involved in an encoded packet is $\phi(i)$, $2 \le i \le K$. Assume that 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 $\alpha$, which is given by

$$\alpha = l - \sum_{i=2}^{K} (i-1)\phi(i), \tag{5}$$

where $\sum_{i=2}^{K}(i-1)\phi(i)$ is the number of packets reduced by network coding. From Equation (5), we find that the total number of packets can be greatly reduced when there are many coding opportunities. The computational complexity of $\alpha$ is $O(K)$.

### 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 the broadcast correlated coding metric and the unicast correlated coding metric, which can be used in broadcast and unicast protocols. First, we introduce the correlated coding metric for broadcast, which is defined as follows.

*Definition* 4.3 (*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 the original packet or the coded packet) to all packets' receivers—$V(u_i) = \{v_1, v_2, \ldots v_K\}$, divided by the number of the potential receivers—that is,

$$BETX = \frac{\alpha}{l} \cdot \frac{\hat{\varepsilon}}{K}. \tag{6}$$

The calculation of our 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 needed to send. The second term $\frac{\hat{\varepsilon}}{K}$ prefers high correlation, which

Table II. Seven State-of-the-Art Protocols Supported by Correlated Coding

| Protocol Name | Category | Network Info. | Hello Msg. | Broadcast Msg. | Routing Strategy |
|---|---|---|---|---|---|
| FMS [IEEE 802.11v 2012] | Multicast | One hop | One hop | Msg. only | Minimal cost |
| ST [Juttner and Magi 2005] | Broadcast | One hop | ID | Msg. only | Tree based |
| FNC [Wu and Lou 2003] | Broadcast | Local | ID | Covered set | Cluster based |
| PDP [Lou and Wu 2002] | Broadcast | Two hop | One hop | Msg. + covered set | Pruning based |
| ZigBee [Ding et al. 2006] | Unicast | One hop | ID | Msg. only | Cluster tree |
| OLSR [Clausen et al. 2008] | Unicast | Two hop | One hop | Msg + covered set | Multipoint relay |
| ETX [Couto et al. 2003] | Unicast | One hop | One hop | Msg. only | Minimal cost |

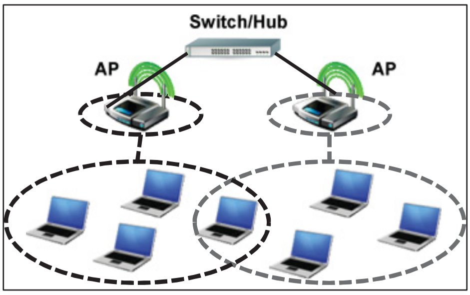


Fig. 8. Collaborative FMS.

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 trade-off between the broadcast efficiency and coding benefit is decided by the product value.

*Definition* 4.4 (*Unicast Correlated Coding Metric*). Given a sender $u_i$ and its potential receiver set $V(u_i) = \{v_1, v_2, \dots v_K\}$. The unicast correlated coding metric $UETX$ is the forwarding cost of the link $e(u_i, v_j)$, $j = 1, 2, \dots K$, which is calculated as follows:

$$UETX = K \times BETX_{V(u_i)} - (K-1) \times BETX_{V(u_i)-\{v_j\}}. \tag{7}$$

The items $K \times BETX_{V(u_i)}$ and $(K-1) \times BETX_{V(u_i)-\{v_j\}}$ in Equation (7) represent the number of transmissions needed by sender $u_i$ to reliably broadcast a packet to the receiver sets $V(u_i)$ and $V(u_i)-\{v_j\}$. Thus, $UETX$, which is the difference of $K \times BETX_{V(u_i)}$ and $(K-1) \times BETX_{V(u_i)-\{v_j\}}$, represents the forwarding cost of the link $e(u_i, v_j)$. The calculation of $UETX$ only involves $BETX$, which can be calculated using Equation (6).

## 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 II.

### 5.1. 802.11 Networks

*5.1.1. Flexible Multicast Service.* In wireless LAN, the flexible multicast service (FMS) is an efficient way to deliver the same contents to a large number of receivers. Notice that in FMS, multiple access points (APs) may share the same upstream service provider. We thus propose a novel communication paradigm called *collaborative FMS*. As shown in Figure 8, multiple APs are connected via wires and form an infrastructure, whereas 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 of the receivers. When one receiver can be covered by multiple APs, as shown in Figure 8, the receiver is assigned to the AP with minimal *UETX*.

## 5.2. 802.15.4 Networks

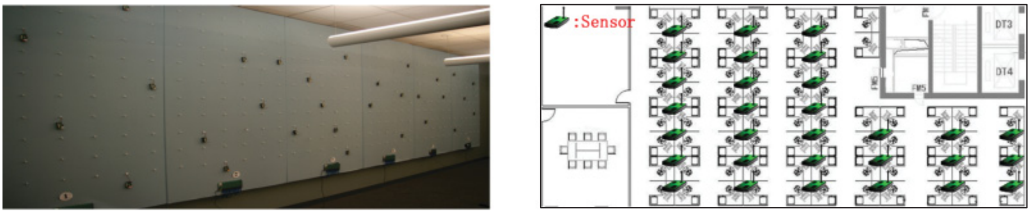
*5.2.1. Broadcast.* We classify the existing deterministic broadcast algorithms into three categories: (i) tree based [Juttner and Magi 2005], (ii) cluster based [Alzoubi et al. 2002], and (iii) pruning based [Lou and Wu 2002]. In the tree-based broadcast [Juttner and Magi 2005], a minimal cost spanning tree (ST) is constructed in a distributed manner, and broadcast is performed based on the tree. The cluster-based approach [Alzoubi et al. 2002] first finds a maximal independent set and then connects the nodes in the set with connectors. The pruning-based broadcast path builds on the multipoint relays, where each relay's one-hop downstream forwarders can cover all of its two-hop neighbors.

We briefly introduce how to integrate the correlated coding metric into these three types of algorithms, thus bringing them transmission gain from link correlation and network coding. In a tree-based algorithm [Juttner and Magi 2005], 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 [Alzoubi et al. 2002] 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 the MIS. In the pruning-based scheme [Lou and Wu 2002], 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.

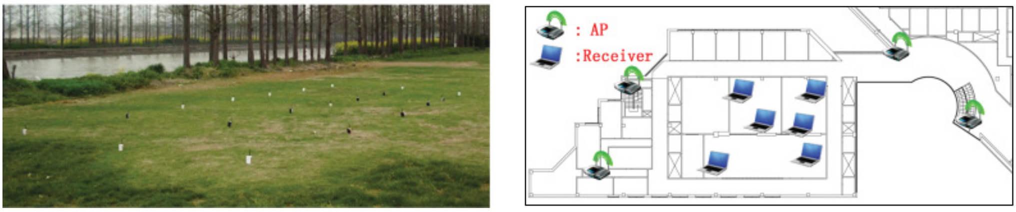
*5.2.2. Unicast.* Unicast routing protocols can be divided into two categories: (i) backbone based and (ii) flat protocols. In the first category, a backbone is built using the tree-, cluster-, or pruning-based method. For example, in ZigBee [Ding et al. 2006], a cluster tree is built. In OLSR [Clausen et al. 2008], multipoint 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 idea of integrating the correlated coding metric to this kind of routing protocol is similar to the deterministic broadcast protocols—that is, we select the nodes with min(*BETX*) to form the cluster tree or multipoint relay set. For the flat protocols, a classic example is ETX [Couto et al. 2003], which selects a routing path with $\min(\sum \frac{1}{p(e)})$. With correlated coding, we select the path with $\min(\sum \mathit{UETX})$.

## 6. TESTBED IMPLEMENTATION

The performance of network coding changes dramatically since the link status and the packet reception pattern in different environments vary significantly. In this section, we report the experimental 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, located in a lab, an open office, and an outdoor environment. The experiment scenarios are shown in Figure 9, and the testbed settings and topology properties are shown in Table III.



(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

Fig. 9. Testbed environments.

Table III. Testbed Settings and Topology Properties

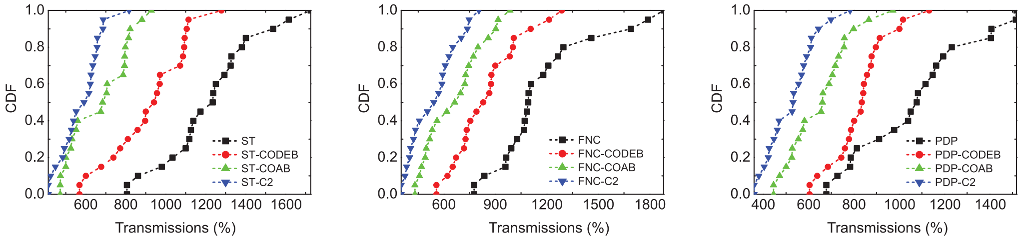
| Platform | Location | Environment | Physical Size | Nodes (#) | Degree | Channel | Power |
|---|---|---|---|---|---|---|---|
| MICAz | UMN | Lab | 8m× 2.5m | 30 | 7 ∼ 23 | 16 | –25dBm |
| TelosB | SIAT | Open office | 18m× 13m | 30 | 6 ∼ 21 | 16, 26 | –25dBm |
| GreenOrbs | TRIMPS | Outdoor | 15m × 5m | 20 | 4 ∼ 13 | 16 | –25dBm, –19.2dBm |
| 802.11g | UMN | University building | 73m× 30m | 6 | 6 | 3, 6 | 15dBm, 20dBm |

### 6.1. Experiment Setup

*6.1.1. 802.15.4 Testbed.* We deploy three 802.15.4 testbeds. The first one is located in a lab environment where 30 MICAz nodes are randomly on an 8m × 2.5m wall (see Figure 9(a)). The second testbed has 30 TelosB nodes that are deployed in an 18m × 13m open office environment, as shown in Figure 9(b). On the third testbed, 20 GreenOrbs nodes are deployed on an open space along a river, as shown in Figure 9(c). On all three testbeds, the default power is –25dBm and the default channel is 26.

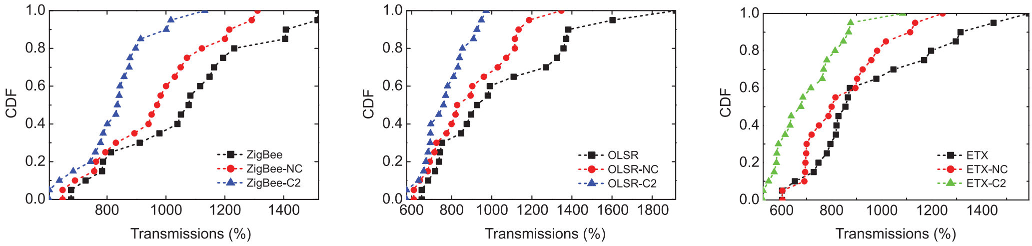
At the beginning of the experiment, a control node is used to remotely configure radio parameters (i.e., transmission power and channel). Based on these radio settings, each node broadcasts $10^5$ packets in turn. Each packet is identified by a sequence number. All received packets are recorded in the nodes' flash memory. When all nodes finish broadcasting $10^5$ packets, they send their packet reception information to a sink node that 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 (i.e., $W$ in Equation (4)) is 100. Then the corresponding nodes on the testbed are selected as forwarders for unicast or broadcast according to the application description in Section 5. In the broadcast application, the forwarders keep on broadcasting packets until all of their covered nodes receive the packets. In the unicast application, two pairs of dataflows are picked up, and the unicast sessions terminate when each source node reliably sends its packets to the destination.

*6.1.2. 802.11 Testbed.* This testbed is located on the fourth floor of the computer science department building at the University of Minnesota, shown in Figure 9(d). From the figure, we can see that four APs are deployed at the four corners of the floor, whereas six



(a) Spanning tree (b) Forwarding node cluster (c) Partial dominating pruning

Fig. 10. Main performance results: broadcast protocols in 802.15.4.



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

Fig. 11. Main performance results: unicast protocols in 802.15.4.

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 that we used is 802.11g. We use the `Lorcon2` packet injection library [Google Code 2012] 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.

### 6.2. Compared Schemes and Performance Metrics

We compare our correlated coding design to the seven state-of-the-art protocols, as well as their enhanced versions, which are integrated with network coding (e.g., CODEB [Li et al. 2007], and COAB [Wang et al. 2013b]). 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. Our design is labeled 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.

### 6.3. Main Performance Results

The experimental results of the seven protocols are shown in Figures 10, 11, 12, and 13. Figure 10 plots the CDF of the transmissions with different broadcast protocols using correlated coding. As the figure shows, our correlated coding design significantly

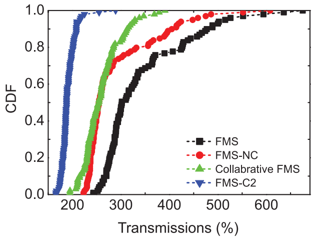


Fig. 12. Main performance results: FMS in 802.11g.

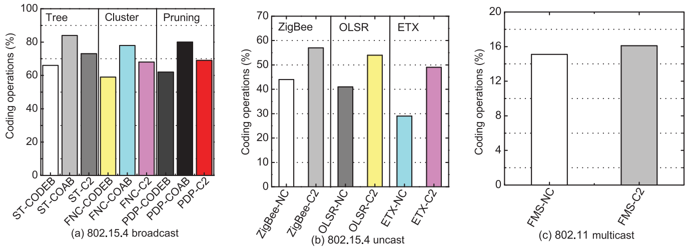


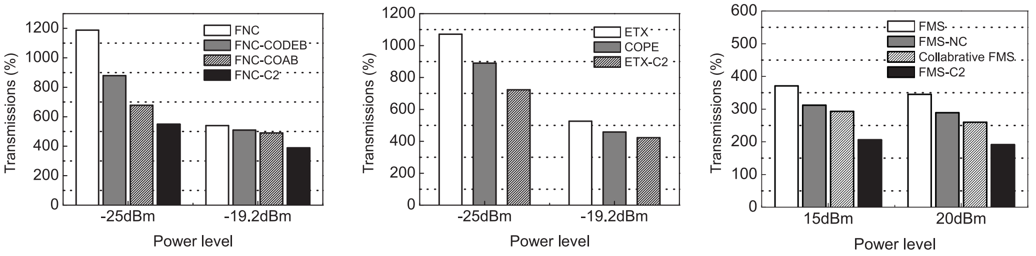
Fig. 13. Main performance results in several coding operations.

improves the transmission efficiency under the three different broadcast strategies. For example, for the tree-based broadcast algorithm (ST), the nodes need 1238 transmissions, on average, to guarantee that all nodes in the network receive 100 packets, whereas 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, our design reduces transmissions by 44% and 22%. In Figure 11, we find that our 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 12, compared to FMS, FMS-NC, and collaborative FMS, our correlated coding design (i.e., FMS-C2) saves 45%, 34%, and 26% transmissions, respectively.

From the experimental results in unicast, broadcast, and multicast on both the 802.11 and 802.15.4 testbeds, we can see that our correlated coding design outperforms traditional network coding protocols. Our design achieves better performance than those protocols because we introduce the link correlation model to network coding. Compared to the traditional link independent model, our link correlation model has better performance on estimating link statuses. Furthermore, with the link correlation model, our 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.

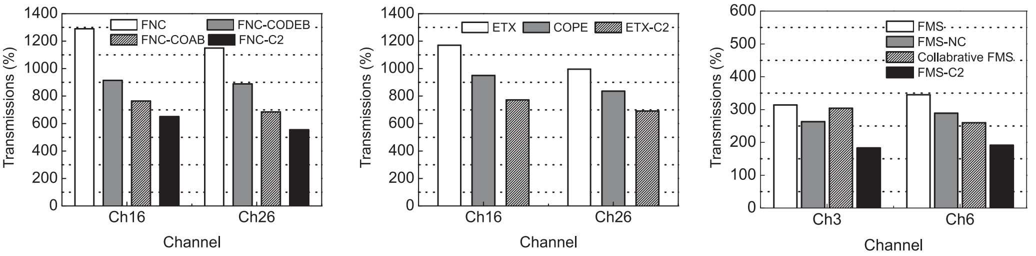
Figure 13 plots the number of coding operations in broadcast, unicast, and FMS applications. Using the broadcast application as an example (Figure 12(a)), we find that the coding-aware routing COAB exploits the most coding opportunities, whereas CODEB and COPE cannot fully exploit the coding opportunities. 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 avoids 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 (FNC) [Wu and Lou 2003]) for the broadcast application, ETX [Qayyum et al. 2002] for the 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 to COPE, CODEB, and COAB. For the



(a) 802.15.4 broadcast (b) 802.15.4 uncast (c) 802.11 multicast

Fig. 14. Impact of power levels.



(a) 802.15.4 broadcast (b) 802.15.4 uncast (c) 802.11 multicast

Fig. 15. Impact of channels.

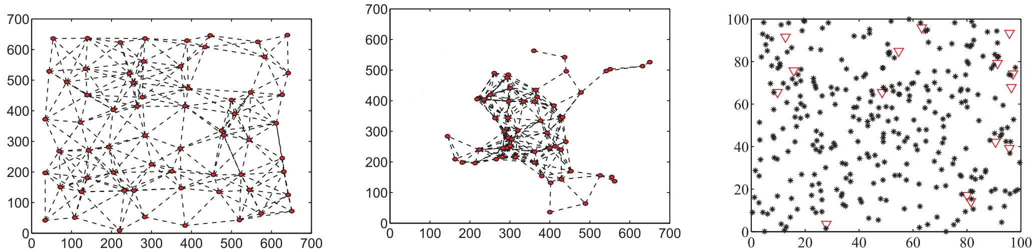
802.11 experiments, we compare correlated coding to FMS, FMS-NC, and collaborative FMS.

### 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 multihop network. Figure 14(a) shows the transmissions of FNC with CODEB, COAB, and C2, and Figure 14(b) shows the transmissions of ETX with COPE and C2 under different power levels. We find that correlated coding greatly reduces transmissions for both broadcast and unicast applications. Under power-level –25dBm, the transmission count for FNC is 1,188, whereas 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-levels 15dBm and 20dBm. From Figure 14(c), the number of transmission of C2 under 15dBm is 205, whereas it is slightly lower (i.e, 192) under 20dBm. In both cases, C2 saves collaborative FMS about 30% transmissions.

### 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: 16 and 26. Note that channel 16 overlaps with a cohabiting AP's 802.11 channel and that channel 26 is free of Wi-Fi interference. The power level for transmission is set to –25dBm. Figure 15 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 algorithms need more transmissions



(a) 802.15.4 topology: Uniform (b) 802.15.4 topology: Nonuniform (c) 802.11 topology

Fig. 16. Simulation topologies for the 802.15.4 and 802.11 networks.

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 channels 3 and 6. The transmission difference between these two channels is not as obvious as the observation on the 802.15.4 testbeds. This is because channels 3 and 6 will be impacted by the nearby APs, which usually use channels 1, 6, and 11.

## 7. SIMULATION

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

### 7.1. Simulation Setup

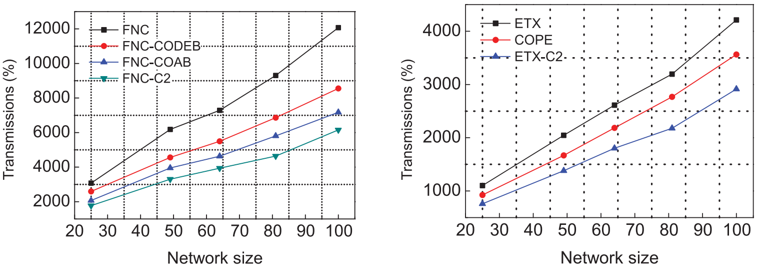
Given a scenario, we generate correlated reception reports for all sender-receiver pairs by modifying the sampling algorithm for Bernoulli random variables in Macke et al. [2009]. For a particular packet, the reception status at receivers could be either 0 or 1. We assume that the reception reports at 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 × 700m with a communication range of 160m. We conduct the experiments on both uniform and nonuniform scenarios, as shown in Figure 16(a) and (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 dataflows. The source nodes keep sending packets until the receivers successfully obtain 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 numbers of APs and receivers. All receivers can be connected to arbitrary APs with link quality varying from 0.2 to 1. The average link quality is 0.6. The default number of APs and receivers is 15 and 300 separately, as shown in Figure 16(c).

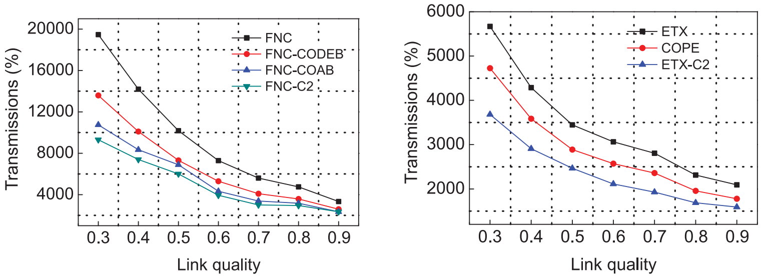
### 7.2. Simulation Results on 802.15.4 Networks

*7.2.1. Impact of Network Size.* Figure 17 shows the performance comparison of our 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. Figure 17(a) shows the results of FNC, FNC-CODEB, FNC-COAB, and FNC-C2. Here, we can find that the average transmission count of our design is 3,940, whereas those of FNC, FNC-CODEB, and FNC-COAB are 7,585, 5,612, and 4,746, respectively. Our design saves 47% of transmissions compared to FNC without using network coding.



(a) Broadcast scenario (b) Unicast scenario

Fig. 17. Impact of network sizes.



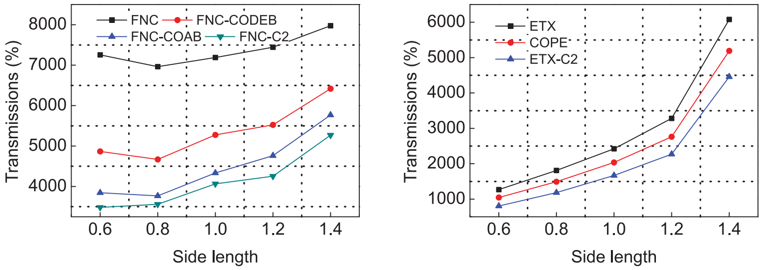
(a) Broadcast scenario (b) Unicast scenario

Fig. 18. Impact of link quality.

Compared to 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 Figure 17(b), compared to ETX and COPE, the transmission gain of correlated coding is 31% and 18%. We can also see that the trends in transmission gain with increasing network size in both unicast and broadcast application are quite stable, suggesting that our design scales well with large networks.

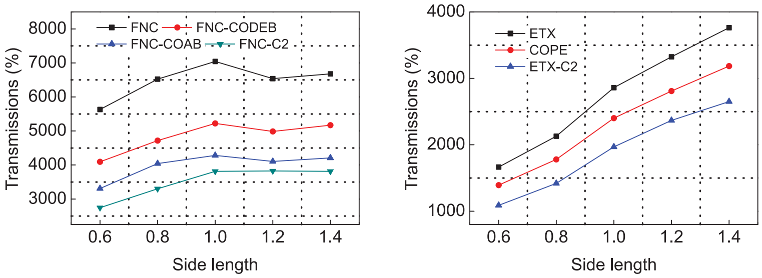
*7.2.2. Impact of Link Quality.* Let us consider the transmission gain of correlated coding for networks with different link qualities. The results are shown in Figure 18. From Figure 18(a), we can see that the broadcast transmission count of our design varies from 9,302 to 2,340 when the link quality varies from 0.3 to 0.9. Compared to 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 Figure 18(b), 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.

*7.2.3. Impact of Network Density.* We consider both uniform (Figure 19) and nonuniform (Figure 20) node distributions. Figure 19 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, and 1.4 are 20.2, 13.0, 8.4, 5.9, and 3.9, respectively. From Figures 19(a) and 20(a), 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



(a) Broadcast scenario (b) Unicast scenario

Fig. 19. Impact of network density (uniform).



(a) Broadcast scenario (b) Unicast scenario

Fig. 20. Impact of network density (non-uniform).

to cover them. One the other hand, the number of forwarders decreases in a fixed-size network.

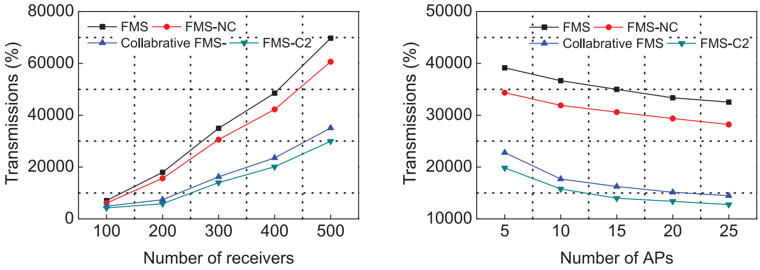
In Figures 19 and 20, the transmission gain of C2 decreases as the side length increases (and thus the density decreases). For example, in the uniform network scenario in Figure 19, the broadcast transmission gain of FNC-C2 over FNC is 52% at node degree 20.2, and it drops to 34% when the average degree 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 nonuniform network topology in Figure 19. 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.

### 7.3. Simulation Results on 802.11 Networks

Figure 21(a) 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 7,079 to 69,691, whereas that of C2 increases from 4,229 to 29,988. In Figure 21(b), with the increase in the number of APs from 5 to 25, the transmission of FMS decreases from 39,152 to 32,536, whereas that of C2 decreases from 19,835 to 12,760. With either increased number of receivers or increased number of APs, C2 has more chances to assign the "black sheep" causing the APs' massive retransmissions to the most suitable APs, thus saving the number of transmissions.

## 8. CONCLUSION

In this article, we study the impact of link correlation on network coding. We find that link correlation can help us decide whether or not a network coding operation is needed. We introduce correlated coding, which optimizes the transmission efficiency of



(a) Varied No. of receivers (b) Varied No. of APs

Fig. 21. Experiments on large-scale Wi-Fi networks.

network coding. Our design can be applied in both broadcast and unicast protocols. We integrate correlated coding with seven state-of-the-art routing protocols and evaluate our design with testbed experiments and extensive simulations. The results confirm the effectiveness of our design compared to the existing network coding protocols under a wide range of system settings.

## APPENDIX

*Proof of Equations (1) and (2).* Equation (1) is a special case of Equation (22) (when $K = 2$). We now demonstrate why Equation (2) holds. 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

$$\begin{aligned} Pr(\varepsilon > t) = & \sum_{i=1}^{K} p(\bar{e_i})^t - \sum_{1\le i<j\le K} p(\bar{e_i} \cap \bar{e_j})^t + \sum_{1\le i<j<l\le K} p(\bar{e_i} \cap \bar{e_j} \cap \bar{e_l})^t + \dots \\ & + (-1)^{K-1} p(\bar{e_1} \cap \bar{e_2} \cap \dots \cap \bar{e_K})^t. \end{aligned} \tag{8}$$

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 reliably broadcast one coded packet can be calculated as

$$E[\varepsilon] = \sum_{t=1}^{+\infty} t \cdot Pr(\varepsilon = t) = \sum_{t=1}^{+\infty} t \cdot (Pr(\varepsilon > t-1) - Pr(\varepsilon > t)) = \sum_{t=0}^{+\infty} Pr(\varepsilon > t). \tag{9}$$

Substituting the right part of the preceding equation with Equation (8), we get

$$\begin{aligned} \mathrm{E}[\varepsilon] = & \sum_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{1\le i<j\le K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j})} + \sum_{1\le i<j<l\le K} \frac{1}{1 - p(\bar{e_i} \cap \bar{e_j} \cap \bar{e_l})} + \dots \\ & + (-1)^{K-1} \frac{1}{1 - p(\bar{e_1} \cap \bar{e_2} \cap \dots \cap \bar{e_K})}. \end{aligned} \tag{10}$$

*Proof of Equation (3).* Based on the observation in Figure 5, we first estimate transmissions for the source node $u$ to reliably send a packet to node $v_i$ with a better link. Then we consider the transmissions of delivering a packet to 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 $i$ nodes in $S_i(u)$ successfully receive a packet. The approximation of $\varepsilon$ is given by

$$\begin{aligned}\hat{\varepsilon} &= \frac{1}{p(e_1)} + \frac{\Pr(\bar{e_2}|e_1)}{p(e_2)} + \cdots + \frac{\Pr(\bar{e_K}|\bigcap_{i=1}^{K-1} e_i)}{p(e_K)} = \frac{1}{p(e_1)} + \frac{p(S_1(u)) - p(S_2(u))}{p(e_1)\cdot p(e_2)} + \cdots \\ &\quad + \frac{p(S_{K-1}(u)) - p(S_K(u))}{p(S_{K-1}(u))p(e_K)} = \sum\nolimits_{i=1}^{K} \frac{1}{p(e_i)} - \sum_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))}.\end{aligned} \tag{11}$$

Thus, we have

$$\hat{\varepsilon} = \sum\nolimits_{i=1}^{K} \frac{1}{p(e_i)} - \sum\nolimits_{i=2}^{K} \frac{1}{p(e_i)} \cdot \frac{p(S_i(u))}{p(S_{i-1}(u))}. \tag{12}$$

## REFERENCES

R. Ahlswede, N. Cai, S. Y. Li, and R. Yeung. 2000. Network information flow. *IEEE Transactions on Information Theory* 46, 1204–1216.

K. M. Alzoubi, P. J. Wan, and O. Frieder. 2002. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proceedings of the HICSS Conference (HICSS'02)*.

A. Atya, I. Broustis, S. Singh, D. Syrivelis, S. Krishnamurthy, and T. Porta. 2013. Wireless network coding: Deciding when to flip the switch. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'13)*.

S. Chachulski, M. Jennings, S. Katti, and D. Katabi. 2007. Trading structure for randomness in wireless opportunistic routing. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM'07)*.

P. Chaporkar and A. Proutiere. 2007. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *Proceedings of the ACM MOBICOM Conference (MOBICOM'07)*.

T. Clausen, C. Dearlove, and P. Jacquet. 2008. The Optimized Link State Routing Protocol Version 2. Retrieved December 21, 2016, from https://tools.ietf.org/html/rfc7181

D. Couto, D. Aguayo, J. Bicket, and R. Morris. 2003. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the ACM MOBICOM Conference (MOBICOM'03)*.

G. Ding, Z. Sahinoglu, P. Orlik, J. Zhang, and B. Bhargava. 2006. Tree-based data broadcast in IEEE 802.15.4 and ZigBee networks. *IEEE Transactions on Mobile Computing* 5, 1561–1574.

Google Code. 2012. Lorcon Wireless Packet Injection Gallery. Retrieved December 21, 2016, from https://code.google.com/p/lorcon/

S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He. 2011. Correlated flooding in low-duty-cycle wireless sensor networks. In *Proceedings of the IEEE ICNP Conference (ICNP'11)*.

IEEE 802.11v. 2012. IEEE standard for information technology. *IEEE Standards Association*.

A. Juttner and A. Magi. 2005. Tree based broadcast in ad hoc networks. *Mobile Networks and Applications* 10, 753–762.

S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. 2006. XORs in the air: Practical wireless network coding. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM'06)*.

A. Keshavarz-Haddad and R. Riedi. 2008. Bounds on the benefit of network coding: Throughput and energy saving in wireless networks. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'08)*.

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

L. E. Li, R. Ramjee, M. Buddhikot, and S. Miller. 2007. Network coding-based broadcast in mobile ad-hoc networks. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'07)*.

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

J. Macke, P. Berens, A. Ecker, A. Tolias, and M. Bethge. 2009. Generating spike trains with specified correlation coefficients. *Neural Computation* 21, 397–423.

A. Qayyum, L. Viennot, and A. Laouiti. 2002. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the HICSS Conference (HICSS'02)*.

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

K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari. 2010. The $\kappa$-factor: Inferring protocol performance using inter-link reception correlation. In *Proceedings of the ACM MOBICOM Conference (MOBICOM'10)*.

S. Wang, S. Kim, Y. Liu, G. Tan, and T. He. 2013a. CorLayer: A transparent link correlation layer for energy efficient broadcast. In *Proceedings of the ACM MOBICOM Conference (MOBICOM'13)*.

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

S. Wang, G. Tan, Y. Liu, H. Jiang, and T. He. 2013b. Coding opportunity aware backbone metrics for broadcast in wireless networks. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'13)*.

J. Wu and W. Lou. 2003. Forward-node-set-based broadcast in clustered mobile ad hoc networks. *Wireless Communication and Mobile Computing* 3, 2, 155–173.

Z. Zhao, W. Dong, G. Guan, J. Bu, T. Gu, and C. Chen. 2015. Modeling link correlation in low-power wireless networks. In *Proceedings of the IEEE INFOCOM Conference (INFOCOM'15)*.

T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. 2010. Exploring link correlation for efficient flooding in wireless sensor networks. In *Proceedings of the USENIX NSDI Conference (NSDI'10)*.

Received December 2015; revised November 2016; accepted November 2016