Exploiting Causes and Effects of Wireless Link Correlation for Better Performance

Song Min Kim, Shuai Wang, and Tian He
Department of Computer Science and Engineering, University of Minnesota, USA
Email: {ksong, shuaiw, tianhe}@cs.umn.edu

***Abstract*—Contradicting the widely believed assumption of link independence, recently the phenomenon of reception correlation among nearby receivers has been revealed and exploited for varieties of protocols [3], [8], [17], [21], [23], [24]. However, despite the diversified correlation-aware designs proposed up to date, they commonly suffer from a shortcoming where link correlation is inaccurately measured, which leads them to sub-optimal performance. In this work we propose a general framework for accurate capturing of link correlation, enabling better utilization of the phenomenon for protocols lying on top of it. Our framework uses SINR (Signal to Interference plus Noise Ratio) to detect correlations, followed by modeling the correlations for in-network use. We show that our design is light-weight, both computation and storage-wise. We apply our model to opportunistic routing and network coding on a physical 802.15.4 test-bed for energy savings of 25% and 15%.**

I. Introduction

Currently, most existing radio link status studies [1], [5], [6], [7], [9], [14], [18], [22] focusing on the individual link or path quality are based on the assumption that receptions among links are independent. However, the unprecedented popularity of wireless technology has created an emerging need for taking link correlation into consideration because of (i) the increasing cross-technology interference due to co-existence of different wireless technologies and (ii) the correlated shadowing introduced by dynamic environments.

Correlation of concurrent receptions at nearby receivers was experimentally revealed in recent works [17], [24], contradicting the popular assumption that wireless links are independent. Then, the significant impact of link correlation was demonstrated, where noticeable performance gain were achieved by incorporating correlation-aware designs into a wide range of applications including: flooding [8], [24], code dissemination [23], reliable broadcast [21], opportunistic routing [3], protocol performance analysis [17], and more.

Despite various representations of link correlation introduced in the works up to date, they all share the common idea of measuring link correlation at the link layer; correlation is solely determined by the similarity of reception success and failure patterns between links. For example, by representing reception success and failure as 0 and 1, measuring link correlation simply turns to be a comparison between two Bernoulli processes. Although the link level correlation measurement is simple, it provides limited information on the phenomenon due to the ignorance on the root cause.

In this work, we propose a framework that addresses the issues in the previous studies, to accurately capture and model link correlation. We utilize SINR, which embeds fine-grained PHY layer information, to replace binary reception status to obtain high granularity measurement. We note that SINR is a well-known metric. Therefore, instead of reinventing the wheel, the focus of this work is in exploring the causes and intensities of link correlation using SINR, and showing how they are modeled. Our contributions are three-fold:

- Although the phenomenon of link correlation has been mentioned in previous literatures, we provide the first extensive study to explore the root causes of link correlation.
- Through test-bed experiments consisting of 41 MICAz nodes, we show SINR's superiority in capturing correlations in comparison with RSSI (Received Signal Strength Indicator) and LQI (Link Quality Indicator). We then demonstrate how correlations are modeled via SINR for in-network use.
- We apply our model to two popular routing protocols, opportunistic routing and network coding, on a test-bed with 15 MICAz nodes. Experiment results show that our model helps opportunistic routing and network coding to save about 25% and 15% transmissions on average.

The rest of the paper is organized as follows: Section II presents what motivated this work. Means to capturing the link correlation and distinguishing the root causes follows in sections III and IV. Section V illustrates the link correlation model, where its performance is evaluated in Section VI. Section VII demonstrates two applications. Related works are discussed in Section VIII. Finally, Section IX concludes the paper.

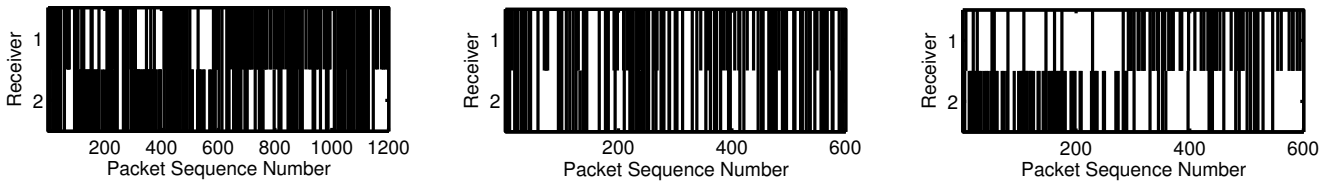
II. Motivation

In this section we first introduce link correlation, where the root causes of the effect is experimentally revealed. Then, the importance of identifying the causes and their intensities, which the existing studies fall short on, is illustrated.

A. Existence and Causes of Link Correlation

Link correlation is a phenomenon where broadcast packet receptions among closely-positioned receivers are not independent. That is, successful packet reception at one receiver strongly indicates success (i.e., positive correlation) or failure (i.e., negative correlation) in nearby receivers. The existence of link correlation was reported and its effect were exploited in recent works [17], [21], [24]. However they do so without

978-1-4799-8381-0/15/$31.00 ©2015 IEEE



(a) Original (b) CTI (c) Correlated Shadowing

Fig. 1. Independent effect of two correlations. While correlation between the two receivers is vague in the original trace, it becomes clear when we separate the cases of correlated shadowing and CTI.

considering the underlying causes, hindering accurate measurement of correlation and thus limited utilization of the phenomenon. Here we introduce the two causes, which are empirically revealed in the following subsection.

• **CTI (Cross-Technology Interference)**: Multiple wireless technologies used today share unlicensed band. For instance 802.11b/g/n and 802.15.4, all reside on the 2.4 GHz ISM band. This leads to CTI, as high-power wireless networks (e.g., Wi-Fi) introduce destructive interference in low-power networks (e.g., ZigBee) [11] throughout a large region, causing correlated packet loss in multiple links simultaneously.

• **Correlated Shadowing**: Wireless signals suffer shadow fading caused by the presence of obstacles in the propagation path of the radio waves. This affects closely positioned receivers altogether, leading to correlation among them [16].

B. Impact of the Two Causes

We now demonstrate the two distinct causes of link correlation, i.e., CTI and correlated shadowing. Based on the experimental result we show that they ought to be treated separately in order to accurately capture correlation. We run an experiment with three MICAz nodes, equipped with IEEE 802.15.4 compliant TI CC2420 radio. Two receivers are placed 1.6m apart while the sender sits in between them. While the sender broadcasts 1,200 packets, the two receivers listen and record the sequence numbers of the received packets, which is downloaded to a PC at the end of the experiment for analysis.

This experiment has two states to isolate CTI from correlated shadowing, and vice versa. In state one, nodes tune to channel 26 to avoid CTI from Wi-Fi, while people walk freely between sender and receivers, causing shadow fading. On the other hand, state two uses channel 12 to take the effect of CTI into account, while no obstacles are allowed. The nodes periodically switch between the two states every 100 packets. The reception traces obtained from the two receivers is shown in Figure 1(a), where the correlation is hardly observed. However, the trend becomes much more clear when CTI and correlated shadowing are distinguished from each other, as shown in figures 1(b) and 1(c). In Figure 1(b) the two receivers are positively correlated via CTI, due to powerful Wi-Fi signal propagating throughout a long distance to corrupt packets at multiple 802.15.4 receivers. Negative shadowing correlation in Figure 1(c) is a result of obstacles that block either one of the two. We note that, This result not only shows the existence of correlation induced by CTI and correlated shadowing, but also offers insight on the importance of differentiating one cause from another for an accurate correlation measurement.

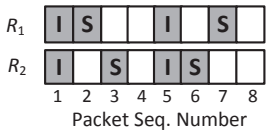


Fig. 2. Example packet reception history. Black boxes with I and S stand for packet losses due to CTI and correlated shadowing.

The walk-through example in Figure 2 clarifies that the lack of the capability of differentiating the two causes leads to incorrect correlation measurements. In the example we have two receivers, R_1 and R_2, receiving broadcast packets from a sender. The white boxes indicate successful receptions. The black boxes imply failures, where the letters indicate the cause; i.e., *I* and *S* stand for CTI and correlated shadowing. It is clear that receivers are perfectly positively correlated in CTI, and negatively correlated by correlated shadowing. However, treating CTI and correlated shadowing altogether falsely diagnose the receivers as independent. Let H_1 and H_2 denote R_1 and R_2's histories, where 1 indicates a success (white box) and 0 indicates a failure (black box). Then, we have $H_1 = \{0, 0, 1, 1, 0, 1, 0, 1\}$ and $H_2 = \{0, 1, 0, 1, 1, 1, 0, 0\}$. In this case, the correlation, computed by Pearson's rho, becomes $\rho = 0$, indicating independence.

C. The Need for High-granularity Measurement

Existing works on link correlation measures the phenomenon at the link layer, where they simply consider the relationship between the reception success or failure on different links, without taking the *quality* of the reception into account. Let us assume link correlation is to be measured between R_1 and R_2. Under existing approach, observing link correlation is simplified to measuring the chance of reception failure on R_2 given failure on R_1. Although simple and intuitive, this method lies on a rather strong assumption: all reception failures on R_1 implies the same chance of failure on R_2. However, this is often not the case in practice. The probability of failure on R_2 when R_1's reception has failed, is highly dependent on the intensity of the cause for the failure. For example, if failure on R_1 has occurred due to high-power CTI, the chance of R_2 facing a failure is high. Low-power CTI imposes lower chance of failure on R_2. The same applies to correlated shadowing. The degree of the cause can be obtained from the reception quality at R_1, from the PHY layer. In other words, link correlation should account for information from both link and PHY layer for accuracy.

III. Obtaining SINR

This section shows how SINR is obtained. Computed for each packet based on the RSSI, we adopt an improved RSSI sampling to overcome the limitation present in the default sampling method.

A. Partial-sampling Problem

In 802.15.4-compliant radios (e.g., CC2420), RSSI is measured for the first 8 symbols, following the very beginning of a packet indicated by SFD (Start of the Frame Delimiter) [20]. Therefore RSSI only reflects a small initial portion of the packet, and is unable to detect any event that occurs in the following parts; an issue we call *the partial-sampling problem*. This imposes a significant limitation on accurately presenting the quality of a packet reception, especially when Wi-Fi is identified as the main source of CTI in low power networks [15]. Specifically, there exists a high possibility of partially corrupted packet in 802.15.4 networks, as Wi-Fi packets usually have much shorter in-air durations than those of 802.15.4. For instance, the duration of the maximum-sized packet in 802.11b and 802.11g networks are 1,906μs and 542μs, whereas 802.15.4 packets can span up to 4,256μs [11]. To address the partial-sampling problem, we directly access the RSSI register on the radio chip to acquire series of RSSI samples throughout the whole duration of a packet. We refer to this approach as *full-sampling*, which is discussed in detail in the following subsection.

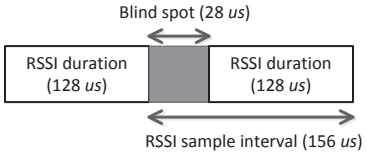


Fig. 3. 6.4kHz RSSI Sampling

B. Full-sampling

Packet duration can be detected via SFD pin, where it rises upon beginning of a packet reception and falls at the end. RSSI series that cover the entire packet can be obtained by sampling RSSI values between SFD pin rise and fall at a high frequency. Our 6.4kHz RSSI sampling rate is achieved using 32.768kHz watch crystal on MICAz's Atmega128 [2]. The rate is chosen to appropriately capture CTI induced by the minimal 802.11g packets. Referring to Figure 3, an RSSI sample is computed as an average during 8 symbols, which corresponds to 128μs. The 6.4kHz sampling rate indicates a sample interval of 156μs. Therefore, the duration of the *blind spot* where we cannot sense the interference comes to be only 28μs, which is sufficiently small to ensure capturing of a minimum-sized 802.11g packet (194μs) or even ACK (112μs).

The effectiveness of full sampling is validated via an experiment. A pair of closely-positioned MICAz receivers, R_1 and R_2, listen to a broadcast from a sender in the presence of a Wi-Fi interferer that generates CTI. Figure 4 depicts RSSI series obtained from the receivers. The figure demonstrates

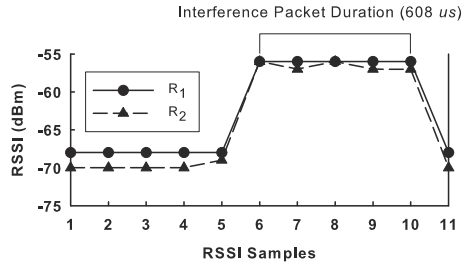


Fig. 4. Full-sampling of RSSI to capturing CTI.

two properties of full sampling: (i) It is capable of capturing CTI in partially corrupted packets. It exhibits a steep increase for the duration of CTI, which is 608μs (456-byte PHY Protocol Data Unit at 6Mbps data rate) in the figure. (ii) Link correlation caused by CTI is clearly observable via similar trend on the two series. In fact, the packet did not pass CRC checks on both receivers, indicating a correlated packet loss.

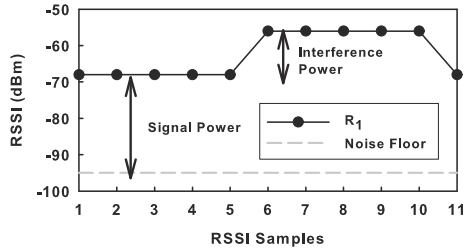


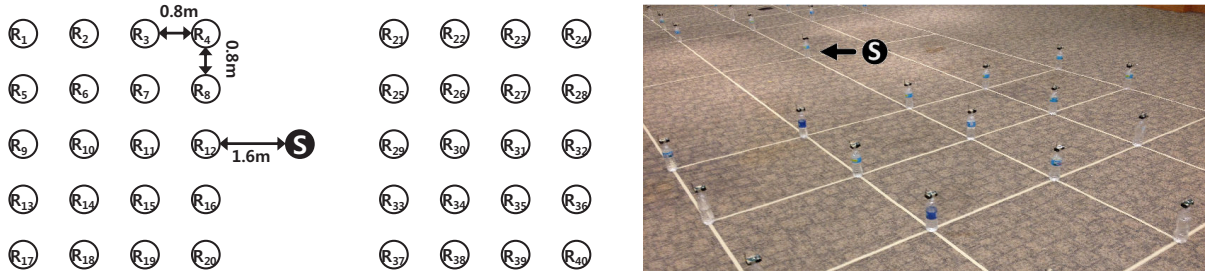
Fig. 5. Acquiring parameters for SINR computation

C. Computing SINR

Now we describe how SINR can be computed from RSSI series obtained via full-sampling, with the help of the noise floor. Figure 5 shows R_1's RSSI series borrowed from Figure 4. The signal corresponds to the packet being received. Note that when CTI occurs, its power simply adds up on top of the signal. The noise floor is acquired by the RSSI value as soon as the link turns idle after packet reception, in order to best reflect the noise floor at the time of reception. Let $\mathbf{R} = \{r_1, r_2, ..., r_m\}$ indicate the RSSI series obtained during receiving a packet when r_i represents the i-th sample in the series. Finally, let r_n denote the noise floor level. Then, we can compute the power of signal p_s, noise p_n, and CTI p_i as follows:

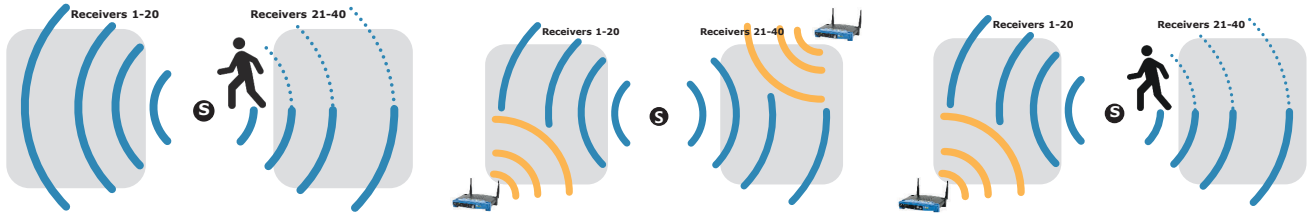
$$\begin{aligned} p_n &= 10^{r_n/10} \\ p_s &= 10^{min(\mathbf{R})/10} - p_n \\ p_i &= 10^{max(\mathbf{R})/10} - p_s - p_n \end{aligned} \quad (1)$$

where $min(\mathbf{R})$ and $max(\mathbf{R})$ indicate the minimum and maximum elements in $\mathbf{R}$. The minimum value is used for the signal power simply to exclude the effect of CTI. The rationale behind taking the maximum value for The CTI power is to consider a part of the packet that is the most vulnerable to corruption. This is because 802.15.4 does not support correction codes, and thus an error in any part of the packet directly leads to packet loss. Then SINR in dB can be computed as:



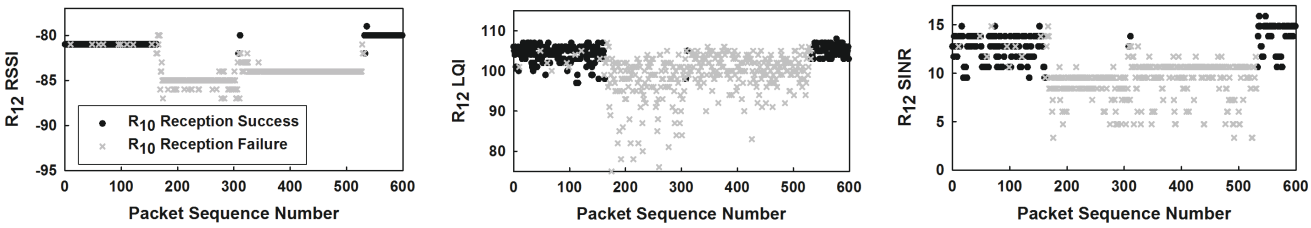
(a) Experiment Setting (b) On-site view

Fig. 6. Test-bed Deployment



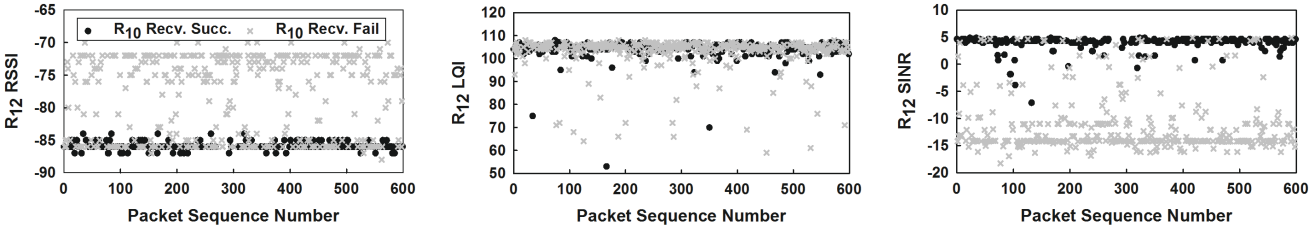
(a) Correlated Shadowing (b) CTI (c) Real-life

Fig. 7. Experimental Scenarios. In (a) and (b) we isolate the two causes to verify metrics' performances. (c) is when both simultaneously occur; the case where most of the real-life situations fall into.



(a) RSSI (b) LQI (c) SINR

Fig. 8. Capturing correlated shadowing. All three metrics are able to capture correlated shadowing.



(a) RSSI (b) LQI (c) SINR

Fig. 9. Capturing CTI correlation. Both RSSI and LQI suffer from partial-sampling problem, leading to failure in capturing CTI correlation.

$$SINR_{dB} = 10log_{10}\frac{p_s}{p_n + p_i} = 10log_{10}\frac{10^{min(\mathbf{R})/10} - 10^{r_n/10}}{10^{max(\mathbf{R})/10} - 10^{min(\mathbf{R})/10} + 10^{r_n/10}}$$

where the subscript $_{dB}$ is omitted from now on for simplicity. The strength of SINR lies in its capability to capture both correlated shadowing and CTI. Moreover, we note that the computation for SINR offers CTI power (i.e., p_i) as a byproduct. This allows us to distinguish the causes of correlation, which is another key point for accurate correlation measurement. Correlation capturing as well as differentiating the causes are further discussed in the next section.

IV. Performance of SINR

In this section we experimentally demonstrate that SINR successfully captures both correlated shadowing and CTI. In comparison, we validate the performance of two well-known metrics, RSSI and LQI. We also show how SINR can express the distinct correlation trends with the help of CTI power, p_i.

A. Experimental Setup

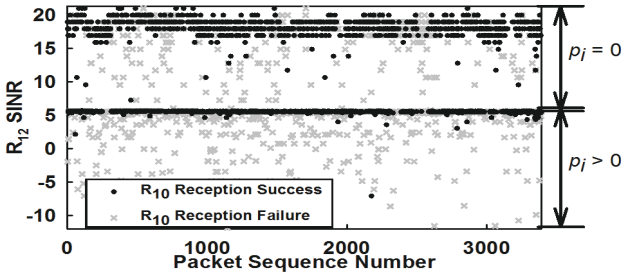
SINR, RSSI, and LQI are tested in three distinct experiments under correlated shadowing, CTI, and both. Each experiment consists of 41 MICAz nodes, where one is the sender and the rest are receivers. Deployment scenario depicted in Figure 6(a) is installed on an indoor test-bed as in Figure 6(b).

Specifically, the test-bed takes place in a large conference room in a university building where a high volume of Wi-Fi traffics (i.e., CTI) with more than 20 access points are present.

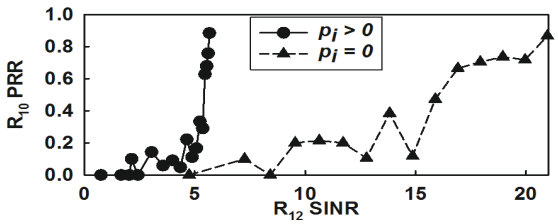
In all three experiments, the transmission power of the sender is tuned to 0dBm. The sender periodically broadcasts 54-byte payload packet at a rate of a packet per second for an hour for the scenarios in figures 7(a) and 7(b), and for 4 hours for the case in Figure 7(c) (a total of 21,600 packets). In the correlated shadowing scenario depicted in Figure 7(a) we allow people to walk or stand around the sender to induce shadowing. We avoid the CTI by running the test-bed on channel 26, which does not overlap with Wi-Fi. We do the opposite in the second experiment in Figure 7(b) where we tune the test-bed to channel 12, which overlaps with Wi-Fi channel 1 to introduce CTI. Meanwhile, this time we avoid obstacles. Finally, in the last experiment we test the case where both CTI and correlated shadowing are present; the scenario where most of the real-life situations fall into.

B. Capturing Correlation

Figure 8 shows the correlation between receivers 10 and 12 (i.e., R_{10} and R_{12}), which share the same LOS path from the sender. The three results for RSSI, LQI, and SINR are from the same series of broadcast packets for a 10-minute duration, to enable direct comparison of the performances. RSSI and SINR in figures 8(a) and 8(c) show a clear drop under correlated shadowing, between sequence numbers of 200 and 500 during which R_{10} experiences continuous packet reception failures. For example, when the RSSI of R_{12} is below to -80dBm, PRR (Packet Reception Ratio) on R_{10} drops to 28.6%. Similarly, for R_{12} SINR smaller than 13dB, R_{10} PRR is only 22.6%. This trend is also observable on LQI in Figure 8(b) but is not as clear compared to RSSI and SINR.



(a) R_{12} SINR vs R_{10} Reception



(b) R_{12} SINR vs R_{10} PRR

Fig. 10. Capturing correlated shadowing and CTI simultaneously in real-life scenario.

The limitation of RSSI and LQI is clearly revealed in Figure 9. In this result obtained from the CTI scenario in Figure 7(b), both metrics fail to show a clear relationship between R_{10} and R_{12} (figures 9(a) and 9(b)). This is because both RSSI and LQI suffer from partial-sampling problem, where only the first 8 symbols of the packet are considered. Partial-sampling brings fundamental limitation in detecting CTI, which often corrupt packets partially. Meanwhile, Figure 9(c) demonstrates that SINR successfully addresses this problem and effectively detects correlation under CTI: At R_{12} SINR above 2dB, R_{10} PRR is 90.5%, where otherwise is only 4.8%.

C. Distinguishing the Causes

Now that we know SINR is capable of detecting both correlated shadowing and CTI separately, we look into the real-life scenario in Figure 7(c) where both phenomena occur simultaneously. The result for 1 hour duration is depicted in Figure 10(a) in which two trends are observable, which can be distinguished via the value of p_i. Recall that p_i is a parameter in computing for SINR, representing CTI power; therefore $p_i = 0$ indicate CTI-free case (i.e, under shadowing) while $p_i > 0$ implies the existence of CTI. This essentially enables SINR to offer clear trends from the two independent phenomena as shown in Figure 10(b). Therefore, SINR is capable of capturing correlations in full under mixture of the two phenomena.

V. Link Correlation Model

This section discusses how the effects of CTI and correlated shadowing are modeled with SINR.

A. Applying Logistic Regression

As demonstrated in Figure 10(b), the relationship between SINR and PRR of two receivers induces two separate curves due to correlated shadowing and CTI. We adopt logistic regression to model each curve, a technique widely adopted especially in the field of machine learning. To offer a better idea on the technique, we first show the results in Figure 11, where the left-most plot in the figure shows the result when it is applied between R_{10} and R_{12} (cf. Figure 10(b)). Black curve indicate CTI correlation ($p_i > 0$), whereas correlated shadowing ($p_i = 0$) is presented in gray. The figure clearly shows that two logistic regression curves are very different, again justifying the need to separately model the two correlations. Note that R_{19} and R_{29} are negatively correlated by shadowing. This is because the two receivers are on the other side of the sender, where obstacle are only big enough to block one side of the sender.

Among various modeling techniques, logistic regression was selected due to its exclusive advantages for our purpose: (i) The y-axis of the model ranges from 0 to 1, which can be directly interpreted as probability. (i.e., PRR) (ii) Computational simplicity makes logistic regression feasible even under strict resource constraints, such as in sensor nodes [13]. Now we move on to how logistic regression is applied to express the relationship between R_A's PRR and R_B's SINR. Let Y be

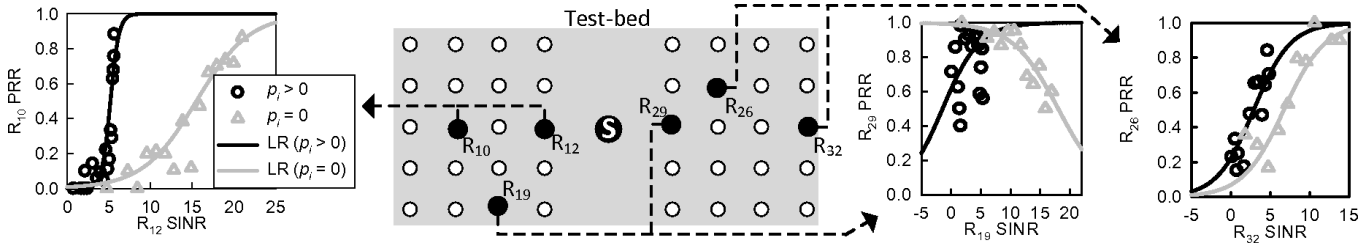


Fig. 11. Logistic regression results for representative receiver pairs in our test-bed. CTI and correlated shadowing are plotted in black and gray, respectively.

a Bernoulli random variable denoting reception success and failure on R_A, where 1 and 0 indicate reception success and failure, respectively. Next, let X be a 2×1 vector, $[X_0\ X_1]^T$, where X_0 is fixed to 1 (intercept term) and X_1 is SINR measured at R_B. Finally, according to logistic function, the success and failure probabilities of R_A given R_B's SINR is denoted as:

$$\begin{aligned} Pr(Y=1|X) &= \frac{1}{1+e^{-(\theta^T X)}} = h_\theta(X) \\ Pr(Y=0|X) &= \frac{e^{-(\theta^T X)}}{1+e^{-(\theta^T X)}} = 1 - h_\theta(X) \end{aligned} \tag{2}$$

where $\theta = [\theta_0\ \theta_1]^T$ indicates the parameters to be adjusted; that is, θ_1 is the slope of the function and the function shifts along x-axis according to $\frac{\theta_0}{\theta_1}$. Now, let $(X^i,\ Y^i)\ 1 \leq i \leq m$ indicate m training samples, where each sample is obtained from a packet. Then, training, or adjustment of θ, is done such that it maximizes the log likelihood below:

$$J(\theta) = \sum_{i=1}^{m} Y^i log(h_\theta(X^i)) + (1-Y^i) log(1-h_\theta(X^i)) \tag{3}$$

We apply Newton's method to solve for the maximum, where for each iteration θ is updated to

$$\theta \Leftarrow \theta - H^{-1} \nabla_\theta J \tag{4}$$

where H is Hessian matrix and $\nabla_\theta J$ is the gradient vector with respect to θ. They are computed as

$$\begin{aligned} \nabla_\theta J &= \sum_{i=1}^{m} (Y^i - h_\theta(X^i)) X^i \\ H &= \sum_{i=1}^{m} h_\theta(X^i)(h_\theta(X^i)-1)(X^i)(X^i)^T \end{aligned} \tag{5}$$

B. Optimization

As shown in Eq. 3 and 4, logistic regression adjusts its parameter, θ, such that it closely matches the measured relationship between SINR of one node and PRR of the other (by maximizing log-likelihood). There are different optimization techniques that can achieve this, including batch gradient descent, SGD(stochastic gradient descent), and Newton's method. Among these we chose Newtons's method for several reasons, described in the following.

The main advantage of Newton's method is its converge speed. In our data, the method took less than 3 iterations to converge, sufficiently light to be adopted even on platforms with low computational power including wireless sensor nodes. SGD was also tested on the same dataset; it returned worse curve fit even after taking more than 30 iterations under step size of 0.1. Moreover, Newton's method does not require step size parameter, which is a considerable benefit in terms of adaptability and simplicity. Meanwhile, the main disadvantage of the technique is the heavy computational complexity in obtaining the inverse of H in Eq. 4. However, this is not a severe issue in our case, since H is only a square matrix of size 2×2.

We note that, at the first glance, techniques that allow on-line learning (e.g., SGD) might appeal as a better candidate, due to their advantage in accommodating newly obtained samples without the need to reconstruct the entire model. However, we argue that such on-line update techniques suffer from keeping the old samples intact, which degrades their performance in the face of dynamically changing correlation over time. This issue will be further discussed in Section VI-D when we examine the effect of temporal dynamics.

VI. Model Evaluation

A. Baseline and Terminologies

Our model is evaluated against *conditional reception*, or CR in short. Recent studies have shown that CR offers considerable amount of benefit in reception estimation [3], [8], [24]. For a brief idea on CR, let there be two receivers R_A and R_B that hear a broadcast packet from a sender. When the event of reception success at each receiver is indicated as S_A and S_B, $CR = Pr(S_A|S_B)$. Note that CR differs from $PRR = Pr(S_A)$ due to correlation between S_A and S_B. CR is a link-level metric, as it only uses the information on reception success or failure. On the other hand, our model, which we refer to as CM(correlation model), takes advantage of information from the PHY layer.

Denoting the dataset from which both CM and CR is computed as the *training set*, whereas the dataset on which estimation is performed as the *test set*, our model is evaluated in terms of accuracy, estimation performance, and the cost of maintaining estimation quality in the face of temporal link dynamics. The results in this section is based on the data obtained from the real-life experiment shown in Section IV, Figure 7(c). Specifically, since the estimation is very obvious for links with very high and very low PRRs, evaluation is performed among 18 links (i.e., $\binom{18}{2} = 153$ link pairs) that

exhibit intermediate PRRs from 19.2% to 80.8%, with an average of 63.9%.

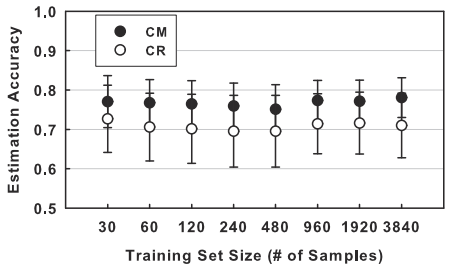


Fig. 12. Model Accuracy

B. Accuracy Verification

We first show how accurately our model reflects the training set. That is, we verify the estimation performance of CM when the test set is equal to the training set. After constructing CM, estimation is done per-packet basis, by taking SINR of one packet at a receiver and probabilistically guessing if reception on the other receiver is a success or failure. Specifically, this is done in two steps: (i) Given a SINR sample, a receiver first finds the corresponding PRR of the correlated receiver via CM. (ii) Then, we estimate the reception to be a success when PRR$\geq$0.5 and failure otherwise. In other words, the logistic regression (i.e., CM) is used as a linear classifier, based on a simple rationale where a reception is more likely to be a success when PRR is above 0.5. For a fair comparison, CR is evaluated similarly to CM, where the reception is estimated to be a success when CR$\geq$0.5.

Figure. 12 shows the result with respect to different training set sizes. Error bars indicate 95% confidence intervals. The figure offers two observations: (i) The accuracy of CM is stable across various set sizes. This demonstrates the fast convergence of CM which only requires a small training set, cutting down the cost for model construction. Moreover, performance enhancement with large training set size indicate the model's robustness to link dynamics. (ii) There exist consistent gaps between the accuracies of CM and CR. This is especially notable knowing that CM does not require additional transmission overhead compared to CR.

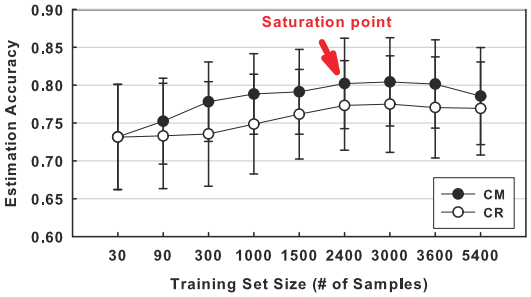


Fig. 13. Estimation Performance

C. Estimation Performance

We now move on to the estimation performance when CM and CR are examined on the test set which is different from the training set. Specifically, we set test set to be a record of 3,600 samples (*i.e.*, 1 hr) *immediately following* the training set. Simply put, this evaluation shows whether the model built on the past history (training set) can be applied to well estimate the future (test set) for an hour. Note that this test is directly related to the performance of CM and CR when applied to real networks, where they most likely face temporal variance in links. In other words, the relationship between correlated receivers dynamically changes over time.

Figure 13 presents the result. It shows that, with a very small training set of 30 samples, the accuracy of CM and CR is approximately equivalent. Performance is enhanced on both techniques as the size grows, where CM improves at a faster rate than CR. The figure exposes two interesting observations. First, the performance of CM reaches the peak at training set size of 2,400 and is kept stable until 3,600. We call 2,400 samples training set as the *saturation point*, as obtaining additional samples does not help to improve performance beyond this point. In practice, the saturation point can be regarded as the maximum number of samples to be obtained. Moreover, this point also indicates the size of the buffer required to fully utilize our model to its maximum. We note that 2,400 samples only take 9.6KB of memory. This indicates that our model can easily fit into sensor networks; Off-the shelf sensor platforms, for instance MICAz and TelosB, are equipped with 512KB and 1MB flash storage. Another interesting observation is that CM's performance starts to degrade at the size of 5,400. This is an effect of time-varying property of links. As 5,400 samples indicate 90 minutes, samples obtained obtained earlier start to be outdated and force CM to make false decisions. The effect of time-varying link dynamics is further evaluated in the following subsection.

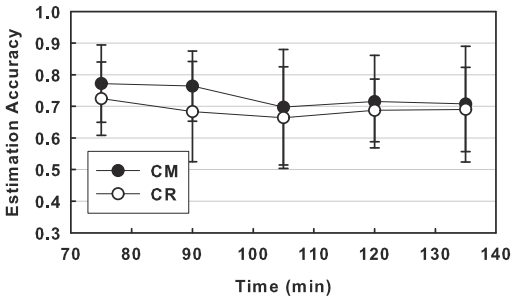


Fig. 14. Effect of Temporal Link Dynamics

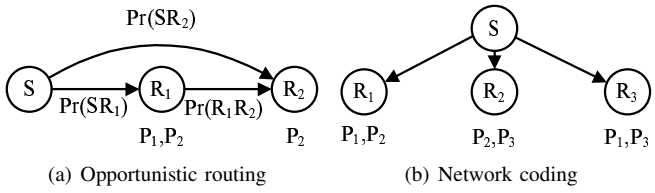
D. Impact of Temporal Link Dynamics

Knowing that the saturation point indicates the most effective training set size, we explore the effect of temporal link dynamics on the performance of CM with the corresponding training set size. Figure 14 shows the trend of the accuracy for every 15 minute duration, beginning an hour after the model is initially constructed, until two hours and 15 minutes. The figure clearly shows that the performance slowly degrades as the model gets outdated, agreeing with our previous observation. From practical standpoint, this figure suggests an optimal sampling frequency to maintain the performance to

the maximum; if 2,400 samples can be collected during 90 minutes, CM with the maximum performance can be continuously rebuilt, without allowing performance degradation. This yields a moderate sampling rate of a sample (i.e., a packet reception) every 2.25 seconds.

VII. Applications

Our model can be used to efficiently predict the nodes' all outgoing links' status, which provides us great opportunities to help a wide range of upper layer protocols obtain better performance. To illustrate its versatility, we show two applications on (i) opportunistic routing, and (ii) network coding.



(a) Opportunistic routing (b) Network coding

Fig. 15. Illustration examples

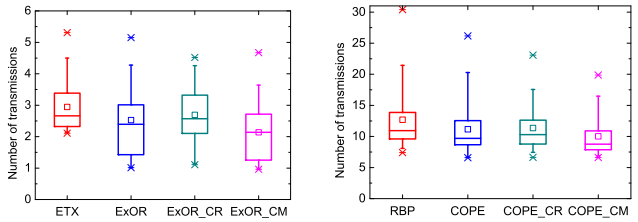
A. Application Examples

1) Opportunistic Routing: Opportunistic routing protocols, such as ExOR [4], defer the selection of the next hop for a packet until these protocols have knowledge about the set of nodes that have actually received that packet. By doing so, opportunistic routing could utilize links that become temporarily available. We use the example in Figure 15(a) to illustrate how opportunistic routing works. In this figure, three nodes form a two-hop network and the best shortest ETX path is "$S \rightarrow R_1 \rightarrow R_2$". With opportunistic routing, we can obtain a better results than ETX if node R_2 overhears the packets (e.g., P_2) from S.

An issue here is how does the source node, S, finds out if the link "SR_2" is available or not. In opportunistic routing, node R_2 explicitly or implicitly sends back an ACK when it overhears packets from S. In reality, when opportunistic routing applies, the link quality of "SR_2", i.e., $Pr(SR_2)$ is not high (otherwise ETX will choose path "$S \rightarrow R_2$" instead of "$S \rightarrow R_1 \rightarrow R_2$"). Note that the reversed link is not symmetric especially when the link quality of the outgoing link is low [25], opportunistic routing loses many opportunities even if R_2 does overhear the packets from S. With our model, source node S can tell the reception status of the long distance link, e.g., SR_2 from a close node, e.g., R_1's reception information.

2) Network Coding: Network coding allows the nodes of a network to take several packets and combine them into one transmission, thus improving the throughput. For example, in COPE [10], as shown in Figure 15(b), the source node S broadcasts three packets, i.e., P_1, P_2, and P_3, to the three receivers. The first receiver R_1 loses packet P_3, the second receiver R_2 loses packet P_1, and the third receiver R_3 loses packet P_2. Instead of transmitting all the three lost packets, COPE sends an XORed packet $P_1 \oplus P_2 \oplus P_3$. Despite the fact that they lost different packets, all three receivers can retrieve the three original packets using the XORed packet.

Now the question is: how does the source node know what packets its neighbors have? In COPE, it estimates the probability that a particular neighbor has a packet as the link quality. Here, our model can help COPE obtain more precise information on neighbor's possessed packets, thus offering more coding opportunities. In detail, one receiver will send back an ACK with its SINR value. Then, the source node computes all its neighbors' packet reception information with our model.



(a) Opportunistic routing (b) Network coding

Fig. 16. Evaluation results

B. Performance Evaluation

To further confirm the analysis, we conduct two experiments with fifteen MICAz nodes which form a two-hop network. The testbed size is 8m by 2.5m. In the experiments, a control node is used to remotely configure the channel and transmission power. The default channel is 16 and the default power is -25dBm. Based on these radio settings, each node sends 10^4 packets in turn. Each packet is sent with a time interval of 0.3 second. The received packets's information, e.g., RSSI, is recorded in MICAz nodes flash memory. When the nodes finish sending packets, they report their packet reception information to a sink node which is connected to a PC. We thus obtain the information required by our model. Then, the corresponding nodes in the testbed are selected as forwarders for opportunistic routing or network coding.

In the opportunistic routing experiment, we compare the performance of ETX [5], ExOR [4], ExOR_CR, and ExOR_CM. In ExOR_CR, we use conditional probability to estimate whether node R_2 received a packet from S or not, given R_1's reception status. In our method, i.e., ExOR_CM, node R_1 will send back an ACK including its SINR information to estimate the nearby links' receiving status. In the network coding experiment, we compare our design COPE_CM, with a traditional broadcast protocol, namely, RBP [19], and two network coding protocols, COPE [10] and COPE_CR.

1) Results on Opportunistic Routing: Figure 16(a) plots the number of transmissions for the source node to send a single packet to the destined receiver using different algorithms. On the average, the source node needs 3.0, 2.5, 2.7, and 2.1 transmissions for ETX, ExOR, ExOR_CR, and ExOR_CM, respectively, to deliver one packet to the destination. The three opportunistic routings perform better than ETX because the

source node learns that the destination overhears the packets and sends the packets without passing through the relay node. ExOR_CM obtains a 30% improvement compared to ETX, which is better than ExOR and ExOR_CR. That is because our model provides a more precise information compared to ExOR and ExOR_CR in predicting the availability of the temporarily unstable links.

2) Results on Network Coding: Figure 16(b) shows the experiment results with RBP, COPE, COPE_CR, and COPE_CM. From the figure, the average number of transmissions for RBP to reliably broadcast one packet to the 14 receivers are 12.7, while COPE, COPE_CR, and COPE_CM is 11.2, 11.4, and 10.0 respectively. Our model helps network coding obtain more benefits. Besides, we find that the gain of COPE is not very high in this scenario. The trace records show that this is because the broadcast packet loss is highly correlated; that is, the receivers lose the same packet and COPE has few opportunity to save transmissions.

VIII. Related Work

This paper is related to link correlation and link quality estimation. Link correlation has been recently enlightened in the networking community. Previous work experimentally revealed the phenomenon and presented protocols that can effectively utilize it to achieve high efficiency in flooding [8], [24], opportunistic routing [3], reliable broadcast [21], and code dissemination [23]. Another pioneering work on link correlation proposed a quantitative index to indicate the degree of the correlation [17]. Our work is fundamentally different from the existing studies for the following three reasons: It is the first to (i) observe the phenomenon using PHY layer information and (ii) explore the two causes of link correlation. Lastly, (iii) instead of proposing a protocol, we focus on accurate capturing and modeling that can benefit a wide range of protocols.

Among a large body of works on link quality estimation, our work is most related to the recent studies that adopt logistic regression for their modeling. In [12], authors model link estimator taking PRR and PHY information including RSSI, LQI, and SNR. This work was improved in [13], where instantaneous prediction was made possible to avoid the need for off-line training. However, our work is inherently different from these studies as our paper makes estimation across links, rather than within the same link.

IX. Conclusion

To further improve network performance, this paper aims to provide a framework that precisely captures the phenomenon of link correlation in reality, under both CTI and correlated shadowing. The framework utilizes SINR that is capable of capturing link correlation in full, while conventional metrics, RSSI and LQI, fail to do so. Using SINR we successfully model link correlation, where its estimation performance is consistently better than the conditional reception, which is the metric introduced in the in the state-of-the-art works [21], [24] as a means to measure correlation. Moreover, we show that our model can be built and maintained with a moderate cost in the face of temporal dynamics of links. Lastly, our model was applied to opportunistic routing and network coding protocols for an average energy savings of 25% and 15%.

Acknowledgement

This work was supported in part by the National Science Foundation under grants CNS-0845994 and CNS-1444021. We thank the anonymous reviewers for their valuable comments.

References

[1] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *MOBICOM '05*, 2005.
[2] Atmel. Atmega128/l datasheet. *http://www.atmel.com/Images/doc2467.pdf*, 2011.
[3] A. Basalamah, S. M. Kim, S. Guo, T. He, and Y. Tobe. Link correlation aware opportunistic routing. In *INFOCOM*, 2012.
[4] S. Biswas and R. Morris. Exor: Opportunistic multi-hop routing for wireless networks. In *SIGCOMM*, 2005.
[5] D. Couto, D. Aguayo, J. Bicket, and R. Morris. A High Throughput Path Metric for Multi-Hop Wireless Routing. In *MOBICOM*, 2003.
[6] 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.
[7] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2001.
[8] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He. Correlated flooding in low-duty-cycle wireless sensor networks. In *ICNP*, 2011.
[9] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks. In *Technical Report SING-06-00*, 2006.
[10] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *TON*, 2008.
[11] C.-J. M. Liang, B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *SENSYS*, 2010.
[12] T. Liu and A. Cerpa. Foresee (4c): Wireless link prediction using link features. In *IPSN*, 2011.
[13] T. Liu and A. Cerpa. Talent: temporal adaptive link estimator with no training. In *SENSYS*, 2012.
[14] M. Zuniga, and B. Krishnamachari. An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links. In *TOSN*, 2007.
[15] M. Petrova, L. Wu, P. Mähönen, and J. Riihijärvi. Interference measurements on performance degradation between colocated ieee 802.11g/n and ieee 802.15.4 networks. In *ICN*, 2007.
[16] M. Quwaider, J. Rao, and S. Biswas. Transmission power assignment with postural position inference for on-body wireless communication links. *TECS*, 2010.
[17] 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.
[18] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The beta-factor: measuring wireless link burstiness. In *SENSYS*, 2008.
[19] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza. Rbp: Robust broadcast propagation in wireless networks. In *SENSYS*, 2006.
[20] Texas Instruments. 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver datasheet. *http://www.ti.com/lit/ds/symlink/cc2420.pdf*, 2013.
[21] S. Wang, S. M. Kim, Y. Liu, G. Tan, and T. He. Corlayer: A transparent link correlation layer for energy efficient broadcast. In *MOBICOM*, 2013.
[22] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SENSYS*, 2003.
[23] Z. Zhao, W. Dong, J. Bu, T. Gu, C. Chen, X. Xu, and S. Pu. Exploiting link correlation for core-based dissemination in wireless sensor networks. In *SECON*, 2014.
[24] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *NSDI*, 2010.
[25] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *SECON*, 2004.