Modeling Link Correlation in Low-Power Wireless Networks

Zhiwei Zhao[†], Wei Dong[†*], Gaoyang Guan[†], Jiajun Bu[†], Tao Gu[‡], and Chun Chen[†]
[†]Zhejiang Key Laboratory of Service Robot,
College of Computer Science, Zhejiang University
[‡]School of Computer Science and IT, RMIT University
{zhaozw, dongw, bjj, chenc}@zju.edu.cn, guangaoyang@gmail.com, tao.gu@rmit.edu.au

***Abstract*—Wireless link correlation can greatly affect the performance of wireless protocols such as flooding, and opportunistic routing. Researchers have proposed a variety of approaches to optimize existing protocols exploiting link correlation. Most existing works directly measure link correlation using packet-level transmissions and receptions. Measurement alone is insufficient because it lacks predictive power and scalability. In this paper, we present CorModel, a model for predicting link correlation in low-power wireless networks. Based on the underlying causes of link correlation, we explore four easily measurable parameters for our modeling. Besides PHY-layer parameters that previous studies have explored, we find that network-layer parameters can also have significant impact on link correlation. We validate our model and illustrate its usefulness by integrating it into existing protocols for more accurate correlation estimation. Experimental results show that our model can significantly increase the accuracy of wireless link estimation, resulting in better protocol performance.**

I. Introduction

Wireless link correlation has attracted much research attention in recent years [1]–[3], since it can greatly affect the performance of wireless protocols such as flooding [4], and opportunistic routing [5]. Link correlation characterizes how packet receptions at multiple receivers correlate for a common sender. In flooding-based protocols, a node with high outbound link correlation might be selected as the next forwarder since it can potentially cover many receivers using one transmission, thus showing large flooding progress. In opportunistic routing protocols, however, a node might select a forwarder set with much reception diversity so that the packet reception probability at any forwarder becomes large.

Based on these observations, researchers have proposed a variety of approaches to optimize existing protocols exploiting link correlation. For example, CF [6] exploits link correlation for reliable flooding in wireless sensor networks. In CF, the mechanism of collective ACKs allows the sender to infer the success of a transmission to a receiver based on the ACKs from other neighboring receivers by utilizing the link correlation among them. Correlated Flooding [7] uses link correlation to establish the routing tree structure in low-duty-cycle sensor networks. By grouping strongly correlated links, the broadcast efficiency can be greatly improved. Link correlation aware opportunistic routing [8] improves the performance of opportunistic routing by optimizing the forwarder set selection algorithm. Under link correlation, the forwarder selection set algorithm prioritizes selecting uncorrelated nodes to increase the level of diversity while ensuring that neighboring nodes are close enough to each other.

All the above works directly measure link correlation using packet-level transmissions and receptions. Measurement alone is insufficient because it lacks predictive power and scalability. One often needs to predict the performance of many possible configurations in order to optimize the network performance [9]. However, measuring all possible configurations is not feasible. Hence, a model for predicting link correlation is necessary.

In this paper, we propose CorModel, a model for predicting link correlation in low-power wireless networks. The input parameters are the key elements for a model. Based on the underlying causes of link correlation, we explore four easily measurable parameters for our modeling, i.e., received signal strengths at multiple receivers, noise and interference, the packet length, and the packet transmission interval. Based on these parameters, we develop a model for predicting wireless link correlation.

Such a model allows us to better understand link correlation in low-power wireless networks. More importantly, it introduces many opportunities for network optimization. We illustrate the usefulness of our model by integrating it into existing protocols for more accurate correlation estimation. Experimental results show that our model can significantly increase the accuracy of wireless link estimation, resulting in better protocol performance.

The contributions of this paper are summarized as follows:

- Based on the underlying causes of link correlation, we explore four easily measurable parameters for our modeling. Besides PHY-layer parameters that previous studies have explored, we find that network-layer parameters, e.g., the packet length, and the packet transmission interval, can also have significant impact on link correlation.
- To the best of knowledge, we are the first to propose an analytical model for predicting wireless link correlation, exploiting both PHY-layer parameters and network-layer parameters.

*This work is supported by the National Science Foundation of China under Grant No. 61472360, 61202402 and 61370087, the Fundamental Research Funds for the Central Universities, and the Research Fund for the Doctoral Program of Higher Education of China (20120101120179). Wei Dong is the corresponding author.

- We validate our model and illustrate its usefulness by integrating it into existing protocols for correlation estimation. Experimental results show that our model can significantly increase the accuracy of wireless link estimation, resulting in better protocol performance.

The rest of this paper is organized as follows: Section II introduces the background and related works. Section III analyzes the impacting factors of link correlation. Section IV presents the model for link correlation. Section V describes the implementation details of the link correlation estimation component built upon our proposed model. Section VI shows the evaluation results, and finally, Section VII concludes this paper and gives future research directions.

II. Background and Related Work

In this section, we first introduce link correlation and its metrics. We then describe related works exploiting link correlation.

A. Link Correlation and Its Metrics

The existence of link correlation. Link correlation characterizes how packet receptions at multiple receivers correlate for a common sender. To confirm the existence of link correlation, we conduct an experiment using three TelosB nodes with one sender S and two receivers R1 and R2. S broadcasts data packets every 10 seconds. R1 and R2 are placed near to each other so that they have similar noise conditions. Figure 1 shows packet receptions as well as the noise readings at the receivers. We can see that the two receivers experience similar noises (including external interference), leading to correlated packet receptions at the two receivers.

The CPRP metric. Throughout this paper, we will use the conditional packet reception probability (CPRP) as the metric for link correlation. CPRP is widely adopted in many existing works [6], [8] for calculating upper-layer performance metrics. The link correlation between links $S \rightarrow R1$ and $S \rightarrow R2$ is defined as the probability that $R1$ receives a packet provided $R2$ receives the same packet (i.e., the percentage that packet receptions on $S \rightarrow R1$ are covered by packet receptions on $S \rightarrow R2$). To simplify the notations, we reuse R1 and R2 to indicate the reception bitmaps of the two receivers, where a 1 indicates a packet reception and a 0 indicates a packet loss. From the context, it should be clear whether R1 and R2 refer to the receivers or the reception bitmaps. CPRP of the two links $S \rightarrow R1$ and $S \rightarrow R2$ can be calculated as follows.

$$c(R1,R2) = \frac{\sum_{i=1}^{|R2|} R1[i] \& R2[i]}{\sum_{i=1}^{|R1|} R2[i]} \quad (1)$$

where $R[i]$ denotes the i-th bit in R_i and & denotes the bitwise and operation.

Consider the example shown in Figure 1. R2 receives 25 packets. 23 out of those packets are also received by R1. Hence, $c(R1,R2) = \frac{23}{25} = 92\%$, implying that R1 can successfully receive a packet with probability of 92%, given that R2 receives the same packet.

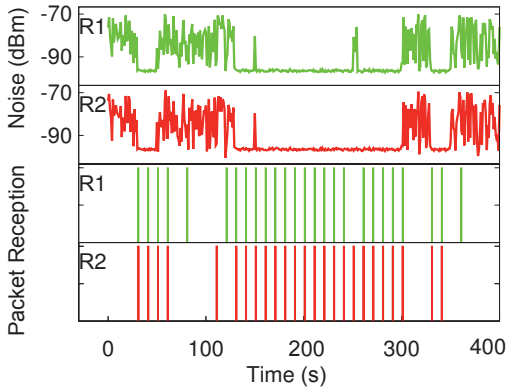


Fig. 1: Noise readings and packet receptions at receivers. Packet receptions are marked by vertical lines.

Other metrics. Similarly, we can also use the conditional packet loss probability (CPLP) as the metric for link correlation. If we use c' to denote CPLP, $c'(R1,R2)$ can simply be calculated as follows: $c'(R1,R2) = c(\neg R1, \neg R2)$ where $\neg R_i$ indicates the negation of R_i and $c()$ indicates the CPRP metric described above. There are other metrics such as χ [9], [10] and κ [2]. Our modeling approach can also be extended to predict these metrics.

B. Related Work

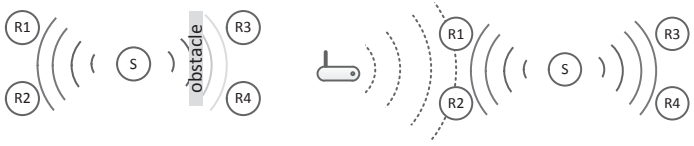
Link correlation has significant impacts on diversity based algorithms in wireless networks such as flooding and opportunistic routing. We briefly discuss related work exploiting link correlation.

Flooding. Zhu *et al*. [6] exploit link correlation for efficient flooding. The proposed protocol, CF (Collective Flooding) utilizes explicit and/or implicit ACKs on one link to infer the packet receptions on adjacent correlated links, so that ACKs on these links can be eliminated. Besides, link correlation is also used for estimating the impacts of potential forwarders. CF employs the dynamic forwarding algorithm to select the most effective forwarders for efficient flooding. Iftekharul *et al*. [11] extends CF [6] to support multi-packet forwarding by employing rateless codes.

Wang *et al*. [12] abstract a general supporting layer (Corlayer) for efficient reliable broadcast considering link correlation. Corlayer blacklists links that are weakly correlated with adjacent links. After blacklisting, broadcast protocols will naturally form clusters with strong link correlations, which means that a forwarder needs fewer transmissions to deliver a packet to all of its covered receivers.

Our previous work [13] employs link correlation for efficient bulk data dissemination. The proposed protocol uses both link correlation and link quality to estimate the ETX of potential forwarders. By constructing an efficient CDS with small transmission overhead, the dissemination efficiency over the CDS can be greatly improved.

Opportunistic routing. The key benefit of opportunistic routing (OR) is to exploit spatial diversity to improve the routing efficiency, i.e., when a node transmits a packet, any



(a) Impact of received signal strength.

(b) Impact of Wi-Fi interference.

Fig. 2: Impact of PHY layer parameters.

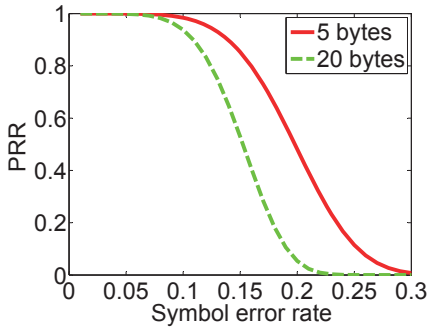


Fig. 3: The SER-PRR curves with different packet lengths.

relay node that overhears the packet can become the next forwarder.

Basalamah *et al*. [8] propose a forwarder set selection algorithm based on a metric that considers link correlation. Under link correlation, the forwarder selection set algorithm prioritizes selecting uncorrelated nodes to increase the level of diversity while ensuring that neighboring nodes are close enough to each other such that the forwarded packet would be heard and duplicates are avoided.

Wang *et al*. [14] extend the work [8] by formally designing a routing metric considering link correlation on multiple links. The metric favors links with weak link correlations. After that, they design an efficient prioritization approach to prioritize the transmissions from forwarders with high metrics. By exploiting the weak correlated links, the routing efficiency is further improved.

Difference of our work. All the above works directly measure link correlation using packet-level transmissions and receptions, suffering from a shortcoming where link correlation is inaccurately measured. Our current work aims to propose a model for predicting link correlation in low-power wireless networks. Such a model allows us to better understand link correlation in low-power wireless networks. More importantly, it introduces many opportunities for network optimization.

Similar to our work, Kim *et al*. propose a general framework for accurate capturing of link correlation [15]. The framework uses SINR (Signal to Interference plus Noise Ratio) to detect correlations, followed by modeling the correlations for in-network use. Our work differs from the above work in two ways. First, we explore network-layer parameters in our modeling approach. Second, we explicitly consider DSSS mechanism incorporated in 802.15.4 radios. It would be interesting to perform a quantitative comparison in the future.

III. Impacting Factors of Link Correlation

In this section, we investigate both PHY-layer parameters as well as network-layer parameters, showing that they can have significant impacts on wireless link correlation.

A. PHY-layer Parameters

Received signal strength. For a common sender, the received signal strengths at multiple receivers can be different due to different propagation paths. For example, wireless signals suffer shadow fading caused by the presence of obstacles, leading to correlated receptions/losses among receivers that are blocked by the same obstacle.

Figure 2(a) shows one sender S and four receivers R1–R4. All receivers have the same distance to node S. R1 and R2 are closely placed. R3 and R4 are closely placed. There is an obstacle between S and R3/R4. The received signal strengths at R1 and R2 will not be affected while the received signal strengths at R3 and R4 will simultaneously degrade due to the blocking. In this case, links $S \rightarrow R1$ and $S \rightarrow R2$ show strong correlation. Links $S \rightarrow R3$ and $S \rightarrow R4$ also show strong correlation. On the other hand, links $S \rightarrow R1$ and $S \rightarrow R3$ tend to have weak correlation.

Noise and interference. Packet receptions at multiple receivers can also be affected by noise and interference, leading to different levels of link correlation.

Figure 2(b) shows one sender S and four receivers R1–R4. There is a WiFi Access Point (AP), interfering packet receptions at R1 and R2. Packet receptions at R3 and R4 are not affected. In this case, links $S \rightarrow R1$ and $S \rightarrow R2$ may have strong correlation. Links $S \rightarrow R3$ and $S \rightarrow R4$ may also have strong correlation. Links $S \rightarrow R1$ and $S \rightarrow R3$ tend to have weak correlation.

B. Network-layer Parameters

Packet Length. Previous studies have shown that packet length has a great impact on packet receptions: with the same SINR (signal to noise and interference ratio), the packet reception ratio of short packets is higher than that of long packets. Figure 3 shows how PRR (packet reception ratio) relates to SER (symbol error rate). Note that we use symbol error rate instead of bit error rate since 802.15.4 radios transmit at the unit of symbols. We can see that different packet lengths lead to drastically different PRRs with the same SER. For example, when SER changes from 0 to 0.0175, PRR of 5-byte packets changes from 1 to $\approx$0.7 while PRR of 20-byte packets changes from 1 to $\approx$0.2. Under the same noise and interference, we can expect that links with larger packet lengths are more likely to experience common packet losses.

We proceed to investigate whether different packet lengths lead to different link correlations observed by upper layer protocols. We conduct two experiments in the same environment with one sender and two receivers. The sender S broadcasts data packets as fast as possible, i.e., immediately send the next packet when the previous packet is sent out. The receivers are closely placed so that they experience similar noise and

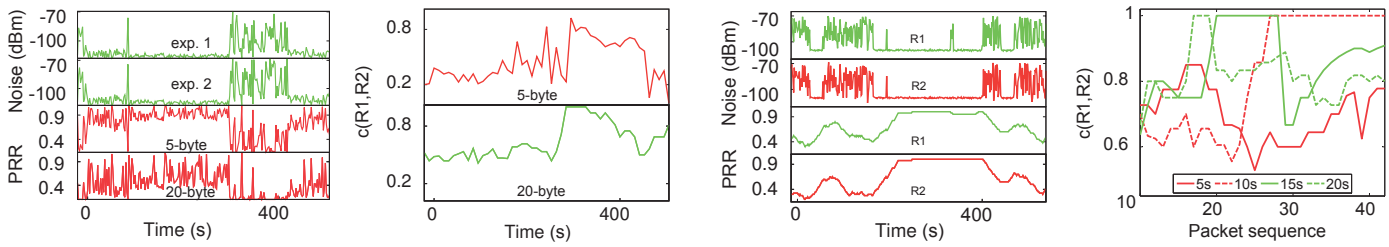


Fig. 4: Impact of network-layer parameters. (a) and (b) show the impact of packet length. (c) and (d) show the impact of data transmission interval.

interference. The sender uses 5-byte packet length in one experiment, and 20-byte packet length in another experiment. Each experiment lasts for 500 seconds. From 300s to 400s, we introduce an external interferer. We measure the RSSI readings at the receivers, and find that noises are very similar as illustrated in the top part of Figure 4(a). Figure 4(a) also shows the PRR of one receiver in the first experiment as well as in the second experiment. Note that in each experiment, the two receivers have similar PRRs. We see that PRR of long packets is lower than that of short packets. In addition, receptions of long packets show larger variations. Figure 4(b) shows the calculated link correlation for the two experiments. We can see that the trends of link correlation in two experiments are roughly similar. However, the absolute value of link correlation in the second experiment is stronger than that in the first experiment. This is because long packets are more likely to experience high packet loss rate when the noise and interference fluctuate, thus showing stronger correlated packet losses.

Packet transmission interval. Packet transmission interval (or transmission rate) can also have an impact on link correlation. We conduct experiments with one sender and two receivers where the receivers are closely placed. Figure 4(c) shows the ambient noise and the PRRs at the two receivers. The sender S broadcasts packets every 5 seconds. We can see that PRRs of the two receivers are very similar.

We then calculate link correlations with different transmission intervals. For example, with an interval of 5 seconds, we only consider packet receptions and losses with 5s transmission interval. Figure 4(d) shows the calculated link correlations with different intervals. We can see that link correlations are different with different transmission intervals. The reason is that transmissions with different intervals capture different noise patterns, thus resulting in different link correlation values.

C. Summary

Prior study has identified that correlated channel fading and cross-technology interference are two main causes for link correlation [1]. The PHY-layer parameters quantify how these causes affect link correlation at the signal level. The network-layer parameters, on the other hand, capture packet-level correlation under the physical conditions. We consider that these parameters are the most important in modeling link correlation. We leave investigation of other possible factors to our future work.

IV. Modeling Link Correlation

In this section, we will present a detailed description on CorModel, a link correlation model for low-power wireless networks.

A. Overview

We consider a local structure in a network with a sender S and two receivers R1 and R2. The sender broadcasts packets to the receivers with a fixed data rate. We divide time into discrete slots with each equal to the time to transmit one symbol, which corresponds to 17.5 μs for 802.15.4 radios. The data transmission period is d time slots.

We would like to estimate the link correlation between $S \rightarrow R1$ and $S \rightarrow R2$ at time slot t. We restrict t to be the time slot in which the sender starts to transmit. We use the m next packets after t for link correlation estimation. With known packet receptions for the m packets, the link correlation can be simply calculated according to Eq. (1). In reality, packet receptions for the m next packets cannot be known at time t. Hence, we need to develop a model for predicting link correlation in the future.

In order to determine link correlation, we require the following specifications.

- Two links with a common sender.
- Time slot t for the estimation.
- RSS at the two receivers from the sender.
- Noise traces (including both ambient noise and external interference) at the two receivers.
- Packet length of L bytes.
- Packet transmission period of d time slots.

We make the following assumptions in our model.

- We assume periodic packet transmissions with a known fixed packet length.

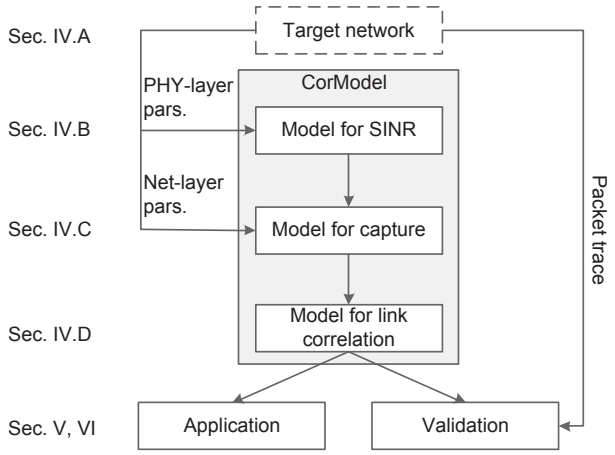


Fig. 5: CorModel Overview.

- We assume SINRs keep stable during a symbol transmission time, which means the chip error rates are identical for a symbol.
- We assume there is no in-network interference.

Figure 5 shows an overview of our modeling approach. Our approach consists of three key models. The SINR model models the future SINR based on the perceived history. The packet capture model models whether the SINR is high enough such that packet is received correctly. The link correlation model models the link correlation based on the packet capture rates. These models are seeded by PHY-layer parameters as well as network-layer parameters described in Section III. We validate the entire approach by comparing the estimated link correlation with direct measurement on the target network. We also apply our approach in 802.15.4-based wireless network for more accurate link correlation estimation. We summarize the notations used in this section in Table 1.

B. Model for SINR

The SINR model takes both the RSS and the noise trace as inputs, and outputs the SINR at arbitrary time slot t.

$$SINR_R^S(t) = \frac{RSS_R^S(t)}{noise_R(t)} \tag{2}$$

The signal strength from S to R at time t is predicted by historical RSS while the noise at receiver R at time t is generated by the CPM (Closest Pattern Matching) model seeded by the noise trace.

C. Model for Packet Capture

We would like to determine the packet capture rate for a given packet among the m next packets. We denote this capture rate as $p_R^S(i)$ where $1 \le i \le m$. In order to determine this quantity, we need to determine the symbol error rates for all the symbols in the i-th packet. We denote the symbol error rate as $SER_R^S(i,j)$ where j denotes the j-th symbol in the i-th packet (j ranges from $1 \sim 2L$ since a 802.15.4 symbol is 4 bits).

Table 1: Notations

Notations	Descriptions
$\bar{c}(R1,R2)$	Expected link correlation between $S \to R1$ and $S \to R2$
d	The data packet transmission interval (slots)
L	The data packet length (bytes)
m	The number of packets used for modeling link correlation
$RSS_R^S(t)$	The received signal strength value at time t
$noise_R(t)$	The noise value at t
$SINR_R^S(t)$	The signal to noise-and-interference ratio (SINR) value at time t
$SINR_R^S(i,j)$	The SINR value corresponding to the jth chip in ith packet
$CER_R^S(i,j)$	The chip error rate corresponding to the jth chip in ith packet
$SER_R^S(i,j)$	The symbol error rate corresponding to the jth symbol in ith packet
$p_R^S(i)$	The packet capture probability for ith packet
$R[i]$	The ith bit in bitmap R

Table 2: The probability of symbol error (rounded to 4 places of decimal) in IEEE 802.15.4 PHY for a given number of chip errors in the received sequence [16].

The number of chip errors, n	The probability of symbol error, $P_{symerr}(n)$
5 or less	0
6	0.0020
7	0.0134
8	0.0523
9	0.1498
10	0.3479
11	0.6496
12	0.9156
13	0.9968
14 or more	1

$$p_R^S(i) = \prod_{j=1}^{2L}(1 - SER_R^S(i,j)) \tag{3}$$

Note that we use symbol error rate instead of bit error rate since 802.15.4 radios transmit at the unit of symbols. The actual transmission takes place one symbol (or 4 bits) at a time. 802.15.4 uses Direct Sequence Spread Spectrum (DSSS) technology to increase packet reception reliability [16]. A 4-bit long symbol is translated to one of the 32-bit chip sequences. The chip sequences are then modulated onto the carrier using O-QPSK.

Determine $SER_R^S(i,j)$. The symbol error rate can be calculated if we know the chip error rate $CER_R^S(i,j)$ for each chip in the symbol identified by (i,j). According to [16],

$$SER_R^S(i,j) = \sum_{n=1}^{32} \binom{32}{n} (CER_R^S(i,j))^n (1 - CER_R^S(i,j))^{32-n} \times P_{symerr}(n) \tag{4}$$

where $P_{symerr}(n)$ denotes the probability of symbol error when n chips are received in error. $P_{symerr}(n)$ can be determined according to Table 2.

Determine $CER_R^S(i,j)$. For an O-QPSK modulated signal under Rayleigh fading model, the chip error rate can be calculated as [17]:

$$CER_R^S(i,j) = \frac{1}{2}\left(1 - \sqrt{\frac{SINR_R^S(i,j)}{1 + SINR_R^S(i,j)}}\right) \tag{5}$$

where $SINR_R^S(i,j)$ denotes the SINR for the symbol (i,j).

Determine $SINR_R^S(i,j)$. We first need to determine the transmission time slot for the symbol (i,j), denoted as t_{ij}.

$$t_{ij} = t + (i-1) \times d + j \tag{6}$$

where t is time for estimation. Then, we can determine the SINR for the symbol as

$$SINR_R^S(i,j) = SINR_R^S(t_{ij}) \tag{7}$$

D. Model for Link Correlation

When we know the packet capture rates for the m next packets at the receivers, we can now calculate the *expected* link correlation for links $S \to R1$ and $S \to R2$.

$$\bar{c}(R1,R2) = \sum_{\forall R1,R2} \frac{\sum_{i=1}^{|R2|} R1[i]\&R2[i]}{\sum_{i=1}^{|R1|} R2[i]} \mathbf{Pr}(R1)\mathbf{Pr}(R2) \tag{8}$$

where R1 and R2 denote the packet reception bitmaps for the two receivers, and $\mathbf{Pr}(R)$ (R=R1, R2) denotes the probability of the particular bitmap given the packet capture rates.

$$\mathbf{Pr}(R) = \prod_{\forall i,R[i]=1} p_R^S(i) \prod_{\forall i,R[i]=0} (1 - p_R^S(i)) \tag{9}$$

Computation according to Eq. (8), however, has exponential complexity. In practice, we can generate a given number of bitmap samples and calculate the *approximated* link correlation.

It is worth emphasizing some features of our link correlation model. First, it predicts the *future* link correlation, instead of using packet reception history. Second, it exploits both PHY-layer parameters as well as network-layer parameters, and thus provides more accurate prediction when these parameters vary.

One direct use of our model is for more accurate link correlation estimation. We will describe how our model can be integrated into existing protocols in Section V. Other possible uses include deployment optimizations, protocol performance prediction, and etc. For example, we can use one mobile sniffer node to sample the environmental conditions at multiple places before deployment. With the knowledge of network-layer parameters, the network performance for a given deployment may be predicted, providing opportunities for deployment optimizations.

E. Extensions

Other correlation metrics. Our model can easily be extended to calculate other correlation metrics. For example, the CPLP metric c' can be simply calculated as $c'(R1,R2) = c(R1',R2')$ where $R_i' = \neg R_i$.

The cross conditional probability χ [10] can be calculated as:

$$\chi(R1,R2) = \frac{\sum_{i=1}^{|R2'|} R1'[i]\&R2'[i]}{\sum_{i=1}^{|R1|} R2'[i]} - \frac{\sum_{i=1}^{|R1|} R1'[i]}{|R1|} \tag{10}$$

Similarly, the κ factor can also be predicted by our model. The κ factor is calculated as the normalized Pearson's coefficient of $R1$ and $R2$ [2].

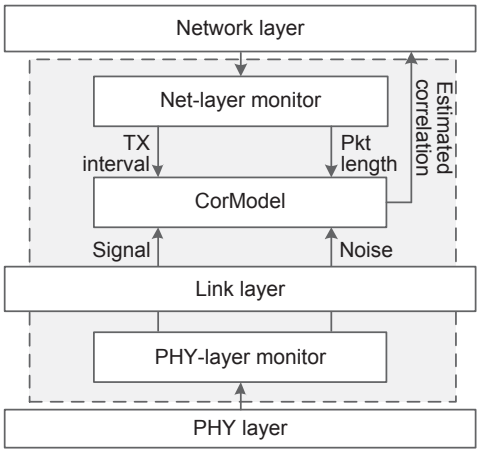


Fig. 6: Overview of CorModel estimation.

Link correlation for more than two links. It is useful to extend our model to a set of links originating from a common sender. For example, the set link correlation is introduced in [12] to model the ETX for reliable broadcast. We define the set link correlation as the link correlation between one link and a set of other links originating from a common sender. When using the metric CPRP, the set link correlation $c(R1,\mathbb{K})$ is the conditional probability that R1 receives the packet under the condition that all the nodes in $\mathbb{K}$ receive the packet. In this case, we can treat $\mathbb{K}$ as a virtual receiver with reception bitmap $K = \bigwedge_{R_i \in \mathbb{K}} R_i$. Therefore, $c(R1,\mathbb{K}) = c(R1,K)$.

V. Link Correlation Estimation

We design and implement a generic correlation estimation component based on our proposed model. Such a component allows the sender to accurately estimate the link correlation for its outbound links. Figure 6 shows the design overview. Each node initially samples the environmental RSSI values (i.e., noise) through the RSSI register. Each node also records the recent per-packet RSSI value from its neighbors. The above RSSI values are exchanged in the neighborhood. When the sender receives the environmental noise time series, it can calculate the noise time series for the future using the CPM model. The sender also uses the recent per-packet RSSI to predict the signal strength in the future. Combining the noise and signal, the sender can calculate the SINR for future packets destined to a given neighbor. The sender also passively monitors the network-layer parameters, e.g., packet length and packet transmission intervals. These parameters are fed into the proposed CorModel for accurate correlation estimation. In this section, we present our implementation in detail.

A. Monitoring PHY-layer Parameters

Our correlation model requires fine-grained SINR for accurate estimation of packet capture. However, the current TelosB/TinyOS platform samples the RSSI values for every 10 bytes. We modify the operating frequency of the SPI bus in the radio driver from 1MHz to 4MHz, significantly improving

the reading efficiency from the CC2420 RSSI register. Our modified driver starts sampling RSSI whenever a special beacon packet is received from a sender, and it keeps sampling at a rate of one sample/byte for a fixed period.

It is important to compress the noise traces before exchange. We employ a simple run-length encoding approach for compression. When a sender receives the noise traces from multiple receivers, it uses the CPM model to generate noises for the future.

B. Monitoring Network-layer Parameters

Our model also requires the network-layer parameters, i.e., packet length and packet transmission intervals. This is important since an application can employ multiple protocols, and different protocols may use different packet lengths and transmission intervals, e.g., Deluge [18] uses a default payload length of 23 bytes while CTP [19] uses user-defined packet length. Our monitoring component sits below the network layer and above the link layer. To differentiate parameters for different protocols, it should keep information for different protocols. Since TinyOS usually uses different AM types in different protocols, we use AM type field to differentiate parameters for different network protocols.

C. Computational Complexity

When both PHY-layer parameters and network-layer parameters are available, we can employ the CorModel for predicting the future link correlation. Since the calculation is performed on resource-constrained sensor nodes, it is important to adopt optimization techniques to reduce the computation complexity.

Pre-computed tables. We use pre-computed tables to speedup the calculation. The first table correlates SINR with SER, which simplifies the calculation of the symbol error rate. The second table correlates reception bitmap pairs to the correlation metric. For $m = 10$, the table costs about 4kB*. We store the table in the external flash with a total size of 1MB. For larger m, we can split the actual reception bitmaps into small segments and apply the pre-computed values on these segments. The final predicted correlation is the average value of the segments.

Generation of bitmap samples. As described in Section VI, we need to generate a sufficient number of bitmap samples in order to calculate the approximated link correlation. There is a tradeoff in deciding the number of generated samples. If we generate more bitmap samples for estimation, the result will be more accurate but the computational overhead will be larger.

We conduct an experiment to study the effects of the number of samples k. The bitmap length is 10 bits. We calculate the accuracy of estimation using k samples as $1-|(\bar{c}_k-\bar{c})/\bar{c}|$, where $\bar{c}$ is the expected link correlation and $\bar{c}_k$ is the expected link correlation using k samples. Table 3 shows the computational delay (on the TelosB platform) and accuracy for different k. In practice, we choose k to be 20 which has a high accuracy of 92% while introducing an acceptable computation delay.

*We divide the 1024 bitmaps to 64 sections, and save a 64x64 table, with each correlation value consuming one byte.

Table 3: Computation delay (on the TelosB platform) and accuracy for different number of samples

# of samples	Computation delay (ms)	Accuracy
5	1.506	0.754
10	3.024	0.832
15	4.534	0.887
20	6.042	0.921
25	7.553	0.946
30	9.063	0.957

Noise generation using CPM. The use of CPM model introduces an exploding state space: if noise takes a value from $M = 60$ possible ones in the range of [-110,-50) dBm with a step of 1dBm, the CPM model with a window size of $k = 20$ has a state space of $M^k = 60^{20}$.

To reduce the computational complexity, we apply a similar optimization approach adopted in [20]: we discretize the sample space into eight intervals, which reduces the number of available values. As a result, the state space can be reduced to 8^k. We need to reduce the window size k in order to maintain a reasonable size of the state space. In practice, we use a window size of 5, i.e., $k = 5$, which translates to a storage size of 32kB for storing the samples.

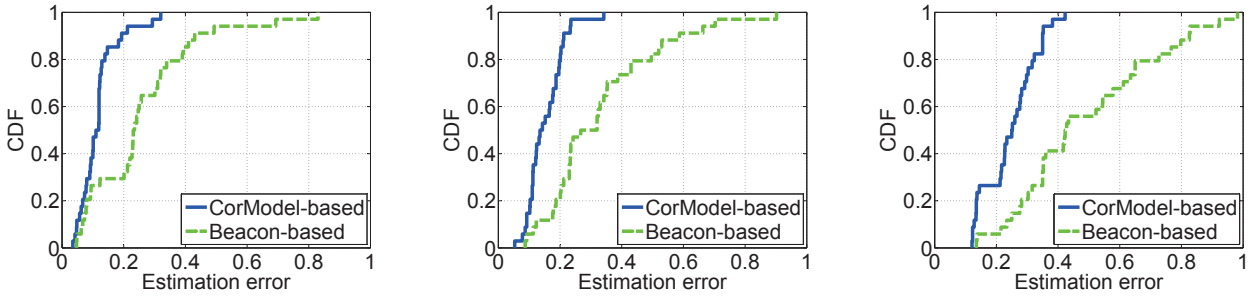
VI. Evaluation

In this section, we validate CorModel by testbed experiments. We also describe how CorModel can be used to improve the performance of two existing protocols, CAR [8] and CoCo [13], by two case studies.

A. Methodology

We conduct our experiments on the TelosB/TinyOS platform in an indoor laboratory. In order to evaluate the prediction accuracy of our model, we use a single hop network consisting of one sender and a set of receivers. The receivers are placed with different distances to the sender so that there will be different link conditions (e.g., poor links with PRR<30% and good links with PRR>90%). The communication channel is set to 20 which is overlapped with WiFi communication channel. In order to evaluate the accuracy of our model, we collect the packet reception traces after the experiments. The ground truth of link correlation at time t is calculated from the next m packets. For comparison, we also implement the commonly used link correlation measurement approach employing periodically exchanging beacon packets [6]–[8], [13].

We also illustrate the usefulness of CorModel by two case studies. We integrate CorModel into existing protocols to evaluate how it benefits other protocols, including link correlation aware opportunistic routing protocol (CAR) and correlation-aware and core-based dissemination protocol (CoCo). By default, both protocols employ beacon-based correlation estimation. We replace the default correlation estimation method by our model-based estimation method. The experiments are conducted in an indoor testbed consisting of 4x6 TelosB nodes. The internode spacing is 0.5m and the communication power is set to -32.5 dBm to enable multihop communication. In CAR, all nodes periodically send data packets to a sink node in a multihop manner. In CoCo, the



(a) A pair of good links. (b) A pair of poor links. (c) A pair of intermediate links.

Fig. 7: Estimation accuracy comparison.

sink node disseminates a code image of 6.73 kB (300 packets) to all other nodes.

B. Model Validation

We conduct 50 experiments for model validation. We use the relative estimation error to measure the estimation accuracy: $e = |c - \hat{c}|/c$ where c is the real correlation and $\hat{c}$ is the estimated correlation derived by our model.

Figure 7 shows the cumulative distribution function (CDF) of estimation errors for the beacon-based approach and our CorModel-based approach for different link conditions. Figure 7(a) shows the estimation errors with a pair of good links and Figure 7(b) shows the estimation errors with a pair of poor links. We can see that CorModel-based approach is more accurate than the beacon-based approach. This is because beacon packets have different length and transmission intervals compared with data packets. The resulting link correlations are thus different.

Figure 7(c) shows the estimation errors with a pair of intermediate links. We can make two observations. (1) The estimation errors of CorModel increase. This is because the accuracy of CPM model decreases due to randomly varied noises with the intermediate links. (2) The difference of estimation errors between the two approaches becomes larger.

C. Case Study 1: Link Correlation Aware Opportunistic Routing (CAR)

A key problem in opportunistic routing is how to select the forwarder set for a given node. A common strategy is to select a forwarder set so that the number of transmissions to any of the nodes in the set can be minimized. Link correlation helps in quantifying the number of transmissions to any of the nodes. Let's denote this quantity as aETX.

CAR is a recent work which exploits link correlation for selecting the forwarder set consisting of two nodes, say, R1 and R2. The aETX metric can be expressed as:

$$aETX = \frac{1}{PRR_{R1} + PRR_{R2} - c(R1,R2)PRR_{R2}} \qquad (11)$$

Consider two perfectly correlated receivers with both PRRs equal to 0.5. In this case, $c(R1,R2) = 1$ and aETX=2. On the other hand, consider two negatively correlated receivers with both PRRs equal to 0.5 (R1 receives a packet when R2 loses the packet and vice versa). In this case, $c(R1,R2) = 0$ and aETX=1. We can see that weak negatively correlated links are preferred in CAR.

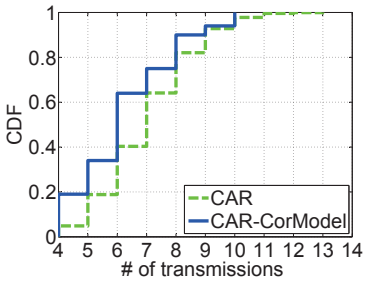


Fig. 8: Link correlation aware opportunistic routing

Link correlation plays a critical role in quantifying the transmission cost. A more accurate estimation of correlation will lead to a more accurate estimation of the transmission cost, resulting in better routing performance. We implement CAR and conduct experiments in a 24-node testbed. We compare two protocols: the original CAR protocol (CAR) and the CAR protocol using CorModel-based correlation estimation (CAR-CorModel). We let a source node to deliver one data packet to the sink node, and record the total number of transmissions of all forwarding nodes. The experiment is repeated 100 times. Figure 8 shows the CDF of the number of transmissions for delivering one data packet. We can see that CAR-CorModel reduces the transmission cost by 9.7% on average, compared with CAR. For CAR-CorModel, more than 60% data packets are delivered to the sink with less than 6 transmissions. On the other hand, only about 40% data packets are delivered to the sink with less than 6 transmissions for the original CAR.

D. Case Study 2: Link Correlation Aware Core-Based Dissemination (CoCo)

Many dissemination protocols rely on an established structure for data dissemination. For example, the connected dominating set (CDS) is a common structure for this purpose [21]. Data dissemination based on CDS is performed in two phases. First, data is disseminated among the dominating nodes (i.e., core nodes). Second, data is disseminated from dominating nodes to remaining nodes with single hop transmissions. The selection of core nodes is an important issue. A

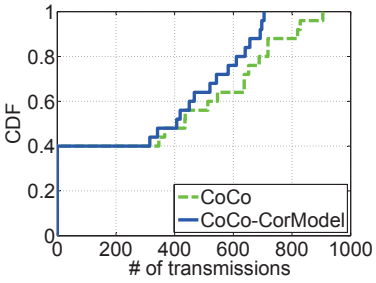


Fig. 9: Link correlation aware core-based dissemination

common strategy is to select a node such that the number of transmissions to cover all its neighbors (i.e., all the neighbors receive the packet) is minimized. CoCo is a recent work which exploits link correlation for core node selection. CoCo favors nodes with high benefit/cost ratio. Specifically, the benefit/cost metric is defined as N/bETX where N is the number of potential receivers and bTEX is the number transmissions to cover all the potential receivers. To illustrate how link correlation helps in quantifying bETX, we consider a sender with two receivers R1 and R2.

$$bETX = \frac{1}{PRR_{R1}} + \frac{1}{PRR_{R2}} - \frac{1}{1-(1-PRR_{R2})c'(R1,R2)} \quad (12)$$

where c' denotes the CPLP metric introduced in Section II-A. Consider two perfectly correlated receivers with both PRRs equal to 0.5 (R1 loses a packet when R2 loses the same packet), $c'(R1,R2) = 1$ and bETX=2. On the other hand, consider two negatively correlated receivers with both PRRs equal to 0.5. In this case, $c'(R1,R2) = 0$ and bETX=3. We can see that perfectly correlated links are preferred in CoCo.

It is worth noting that CoCo considers more than two receivers. In general, CoCo uses the set link correlation metric introduced in Section IV-E to quantify the bETX of a node.

We incorporate CorModel into CoCo (called CoCo-CorModel), and conduct experiments to study the performance of the original CoCo and CoCo-CorModel. In the experiment, the sink node disseminates a data object of 6.73kB (300 packets) to all the network nodes. We record the total number of transmissions on each node. Figure 9 shows the CDF of the number of transmissions for each node. We can see that CoCo-CorModel reduces the transmission cost by 13.7% in average. Note that there are 40% nodes with no transmission cost. This is because these nodes are leaf nodes, and thus only passively receives data packets.

VII. Conclusion

In this paper, we present CorModel, a model for predicting link correlation in low-power wireless networks. Based on the underlying causes of link correlation, we explore four easily measurable parameters for our modeling. Besides PHY-layer parameters that previous studies have explored, we find that network-layer parameters can also have significant impact on link correlation. We validate our model and illustrate its usefulness by integrating it into existing protocols for more accurate correlation estimation. Experimental results show that our model can significantly increase the accuracy of wireless link estimation, resulting in better protocol performance.

Future work leads to two directions. First, we would like to further reduce CorModel's computation complexity. Second, we would like to apply CorModel in more scenarios for network optimizations.

References

[1] N. Patwari and P. Agrawal, "Effects of Correlated Shadowing: Connectivity, Localization, and RF Tomography," in *Proc. of ACM/IEEE IPSN*, 2008.

[2] K. Srinivasan, M. Jain, J. Choi, T. Azim, E. Kim, P. Levis, and B. Krishnamachari, "The κ Factor: Inferring Protocol Performance Using Inter-link Reception Correlation," in *Proc. of ACM MobiCom*, 2010.

[3] S. Paris and A. Capone, "Correlation of Wireless Link Quality: A Distributed Approach for Computing the Reception Correlation," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1341–1343, 2011.

[4] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," in *Proc. of ACM MobiCom*, 1999.

[5] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," in *Proc. of ACM SIGCOMM*, 2004.

[6] T. Zhu, Z. Zhong, T. He, and Z. Zhang, "Exploring Link Correlation for Efficient Flooding in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–1, 2014.

[7] S. Guo, L. He, Y. Gu, B. Jiang, and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2787–2802, Nov 2014.

[8] A. Basalamah, S. Kim, S. Guo, T. He, and Y. Tobe, "Link Correlation Aware Opportunistic Routing," in *Proc. of IEEE INFOCOM*, 2012.

[9] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks," in *Proc. of SIGCOMM*, 2006.

[10] A. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos, "Divert: Fine-grained Path Selection for Wireless LANs," in *Proc. of ACM MobiSys*, 2004.

[11] S. I. Alam, S. Sultana, Y. C. Hu, and S. Fahmy, "SYREN: Synergistic Link Correlation-Aware and Network Coding-Based Dissemination in Wireless Sensor Networks," in *Proc. of MASCOTS*, 2013.

[12] S. Wang, S. M. Kim, Y. Liu, G. Tan, and T. He, "CorLayer: A Transparent Link Correlation Layer for Energy Efficient Broadcast," in *Proc. of ACM MobiCom*, 2013.

[13] Z. Zhao, W. Dong, J. Bu, T. Gu, and C. Chen, "Exploiting Link Correlation for Core-based Dissemination in Wireless Sensor Networks," in *Proc. of IEEE SECON*, 2014.

[14] S. Wang, A. Basalamah, S. Kim, S. Guo, Y. Tode, and T. He, "Link Correlation Aware Opportunistic Routing in Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2014.

[15] S. W. Song Min Kim and T. He, "Exploiting Causes and Effects of Wireless LinkCorrelation for Better Performance," in *Proc. of IEEE INFOCOM*, 2015.

[16] M. Goyal, S. Prakash, W. Xie, Y. Bashir, H. Hosseini, and A. Durresi, "Evaluating The Impact of Signal to Noise Ratio on IEEE 802.15.4 PHY-level Packet Loss Rate," in *Proc. of NBiS*, 2013.

[17] S. Haykin, *Communication systems*. John Wiley & Sons, 2008.

[18] J. W. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," in *Proc. of ACM SenSys*, 2004.

[19] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," *ACM Transactions on Sensor Networks*, vol. 10, no. 4, pp. 1–34, 2014.

[20] H. Lee, A. Cerpa, and P. Levis, "Improving Wireless Simulation Through Noise Modeling," in *Proc. of ACM/IEEE IPSN*, 2007.

[21] V. Naik, A. Arora, P. Sinha, and H. Zhang, "Sprinkler: A Reliable and Energy Efficient Data Dissemination Service for Wireless Embedded Devices," in *Proc. of IEEE RTSS*, 2005.