RxLayer: Adaptive Retransmission Layer for Low Power Wireless

Daibo Liu
School of Computer Science
and Engineering of UESTC
Chengdu, Sichuan, China
dbliu@uestc.edu.cn

Zhichao Cao
School of Software, TNLIST,
Tsinghua University
Beijing, China
caozc@greenorbs.com

Jiliang Wang
School of Software, TNLIST,
Tsinghua University
Beijing, China
jiliang@greenorbs.com

Mengshu Hou
School of Computer Science
and Engineering of UESTC
Chengdu, Sichuan, China
mshou@uestc.edu.cn

Yujun Li
School of Computer Science
and Engineering of UESTC
Chengdu, Sichuan, China
liyujun@uestc.edu.cn

ABSTRACT

In large scale wireless sensor networks, retransmission strategies are widely adopted to guarantee the reliability of multi-hop forwarding. However, keeping retransmission over a bursty link may fail consecutively. Moreover, the retransmission will also be useless over those back-up links which are spatial correlated with the failed link. Thus, it is necessary to design an unified retransmission strategy, which considers both temporal and spacial link properties, to further improve network reliability and efficiency. In this paper, we propose RxLayer, a practical and general supporting layer of data retransmission. Without inducing noticeable overhead, RxLayer captures the temporal and spatial link properties by conditional probability models. A sender will retransmit data over the candidate link with the highest delivery probability while failures occur. RxLayer can be transparently integrated with most of the existing forwarding protocols. We implement RxLayer and evaluate it on both indoor and outdoor testbeds. The results show that RxLayer improves networks reliability and energy efficiency in various scenarios. The network reliability is improved by up to 7.82%, and the total number of transmissions is reduced by up to 36.3%.

Categories and Subject Descriptors

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

Keywords

Retransmission Strategy; Wireless Sensor Networks; Bursty Link; Link Correlation

MobiHoc '14, August 11-14 2014, Philadelphia, PA, USA
Copyright 2014 ACM 978-1-4503-2620-9/14/08$15.00.
http://dx.doi.org/10.1145/2632951.2632962 .

1. INTRODUCTION

In recent years, more and more wireless sensor networks (WSNs) [15] [25] [24] are deployed to collect environment data for various applications. In most of the deployments, multi-hop data relay is utilized to adapt the large scale and low-power transmission. During data collection, however, packet loss is ubiquitous and network reliability is thus degraded. According to measurement study [14] [7], instability of wireless links has so far been regarded as an important cause. The link dynamic is inevitable so that how to efficiently predict the changes and enhance network reliability, namely *retransmission strategy*, is a critical issue.

Several kinds of retransmission strategies are proposed with the development of WSNs. One main kind of retransmission strategies utilize end-to-end acknowledgement (ACK) [13] [17]. Due to the constraint of bandwidth resources, instead of acknowledging every packet, sink node sends negative-acknowledgement (NACK) back to a source node. Then the source node retransmits the packets named in NACK. However, the transmission of NACK still consumes considerable bandwidth resources when the data rate is high. The loss of NACK will further degrade energy efficiency and delay.

Another widely adopted kind of retransmission strategies are hop-by-hop retransmission [9] [4] [23], namely fail-retry strategy. Each packet will be retransmitted over a preselected link until the packet is acknowledged or the number of retransmissions reaches the preconfigured threshold. According to recent link study [19], the short-term transmission over intermediate links are dependent. For example, it will probably fail to transmit during a short period after one transmission failure occurs. This link property is called *link burstiness*. Persistent retransmission over bursty link is obviously not a wise choice.

Further, with link estimation models [9] [8], nodes periodically update link quality. When continuous retransmission failures over individual link appear, the link quality will be getting low so that the sender will choose a back-up link to retransmit. To avoid the longer waiting time in low duty-cycle WSNs, dynamic forwarding [4] [10] chooses multiple next-hop candidates, who wake up in different time, to do quick retransmission. However, *link spatial correlation* [18] indicates that data transmission over the links in a local area

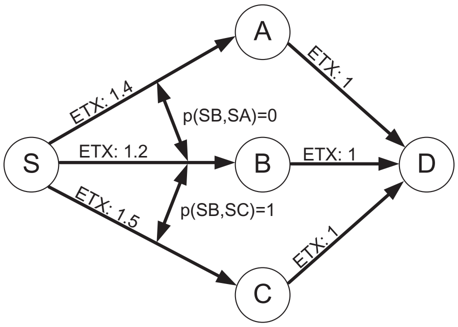


Figure 1: Model of retransmission, where S can forward data to D relayed by A, B, or C.

might be dependent. Hence choosing a link, which is high correlated with the degraded link, to retransmit is useless and energy inefficient. All these strategies that ignore the incorporation of both temporal and spatial link properties will result in unsatisfactory effect of data retransmission. Thus the design of an unified retransmission strategy to target network reliability and energy efficiency is a challenge problem for existing protocols.

In this paper, we propose RxLayer, a general and practical supporting layer of adaptive hop-by-hop retransmission. Without inducing noticeable overhead, RxLayer captures the temporal and spatial link properties by conditional probability models. Then a sender will retransmit data over a candidate link with the highest delivery probability when failures occur.

To our best knowledge, RxLayer is the first unified retransmission layer that incorporates both spatial and temporal link properties to improve data delivery reliability and efficiency. To achieve this goal, there are several challenges. First, the probabilistic models of RxLayer should accurately perceive the temporal and spatial link properties. Second, training the models should be lightweight on both communication and computation. Third, RxLayer should be compatible with most of the existing forwarding protocols.

To address these challenges, we first construct conditional probabilistic models to capture link burstiness and link spatial correlation. For example, a node maintains the probability that the i^{th} retransmission will success after $i-1$ consecutive failures to measure link burstiness. Then the models are trained by the records of data transmission and routing beacon. The models are also periodically updated to reduce the memory and computation consumption. Finally, RxLayer sits between network layer and MAC layer. It utilizes cross-layer information to further optimize forwarding decision and keeps its transparency.

In summary, the contributions of this work are as follows:

- We first propose an unified retransmission layer, RxLayer, which takes conditional probabilistic models to depict both link burstiniss and link spatial correlation so that RxLayer improves both delivery reliability and efficiency.
- RxLayer is practical for various applications since its maintain overhead is low and it is compatible with the most of existing forwarding protocols.
- We implement RxLayer with CTP [9] and LPL [16] and evaluate it on both indoor and outdoor testbeds. The results show that the network reliability is averagely improved by 7.82% and the average transmission count is decreased by up to 36.3% in outdoor experiments.

The rest of the paper is organized as follows. The related work is discussed in the following section. Empirical study is presented in Section 3. Section 4 introduces the detailed design of RxLayer. Followed by its implementation and evaluation in Sections 5. Section 6 concludes this paper.

2. RELATED WORK

Fast retransmission [11], partial packet recovery-based retransmission [12], and network coding [18] are some existing retransmission strategies in wireless networks, but they do not consider the temporal and spacial link properties. Next, we will discuss the existing link estimation methods and retransmission strategies that consider temporal or spacial link properties.

2.1 Link Burstiness and Correlations

The majority of existing link estimation techniques [22] assume that the different packet loss events on a link are statistically independent and follow Bernoulli distribution. Recent study [5] marks this assumption as inappropirate when wireless links are estimated in short time scale. β factor [19] introduces a mechanism to quantify link burstiness. κ factor [18] presents an algorithm to quantify link correlation between selected link and alternative link. However, these factors are computed offline with large set of history data. RxLayer proposes a lightweight way to capture link burstiness and correlation in real-time.

2.2 Exploiting Link Properties

Some forwarding techniques deal with link burstiness and correlation to improve network performance. CPRP[26] and CorLayer [21] exploit link correlation for data dissemination/broadcast. DSF [10] dynamically choose the parent from a set of neighbors to avoid the useless retransmission over a temporally degraded link. With dynamic forwarding, L^2 [4] further proposes the temporal link model to improve both the delivery reliability and efficiency. In contrast, RxLayer comprehensively combines both link burstiness and correlation.

STLE [2] takes the bursty available long link to reduce the number of transmission. It utilizes an overhearing scheme to estimate the short-term link behavior. The current parent is temporally replaced by a node which is much closer to sink and overhears consecutive three packets. However, STLE will be inefficient when the overhearing becomes rare with the diverse wake-up schedule of nodes in low duty cycle WSNs. RxLayer online captures link burstiness and correlations with cross-layer information. It can be transparently integrated with most of existing protocols.

3. EMPIRICAL STUDY

In this section, we conduct empirical studies to show that:

- Link burstiness and link spatial correlation have a significant impact on data delivery performance.
- The methods that solely exploit link burstiness or spatial link correlation in existing works may not always lead to an optimal performance.
- The combination of these link properties provides a great opportunity for efficient transmission.

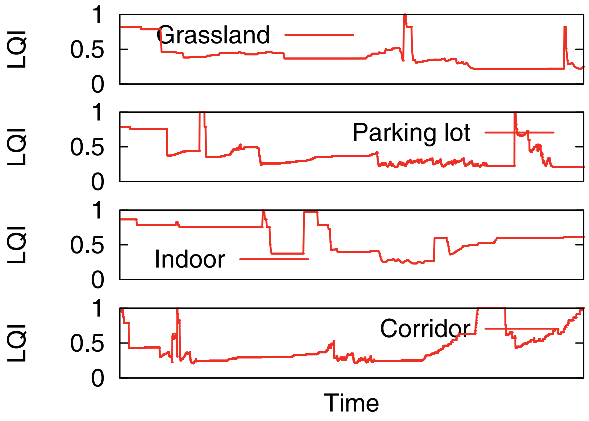


Figure 2: Link burstiness over four links in different scenarios, where LQI=1/ETX is measured by link estimator. Ascendent trend indicates acknowledged transmission, and conversely indicates transmission failures.

3.1 Theoretical Model of Retransmission

Before the empirical study, with a simple model depicted in Figure 1, we first analyze the retransmission efficiency by using consecutive retransmission (CR), exploiting link burstiness (LB), and exploiting both link burstiness and correlation (LBC), respectively. S can forward data to D relayed by A, B, or C. B is S's default parent node.We also assume that all links are bursty and the expected transmission count (ETX) [6] over a link denotes long-term link quality. After encountering bursty loss, for simplicity, we assume the following n consecutive retransmissions will be failed. $P(i, j)$ is the correlation probability between link i and j that a transmission can success on link j when it fails on link i. In the case as Figure 1 shows, $P(SB, SA)$ is 0 indicating the transmission on link $S{\rightarrow}B$ will be failed if the transmission on link $S{\rightarrow}A$ is failed. Inversely, $P(SB, SC)$ is 1 indicating a transmission will be successful on link $S{\rightarrow}C$ when a transmission fails on link $S{\rightarrow}B$.

By using CR to deliver data to B, the expected retransmission count of S is at least n+1. With LB, nodes forward data through sub-optimal link after m (m is far less than n) consecutive failures, first replacing $S{\rightarrow}B$ with $S{\rightarrow}A$ because the ETX of $S{\rightarrow}A$ is lower than that of $S{\rightarrow}C$, the expected retransmission count is $2m$+1 (m failures on link $S{\rightarrow}B$, m failures on link $S{\rightarrow}A$, and a successful transmission on link $S{\rightarrow}C$). By using LBC, after m transmission failures on link $S{\rightarrow}B$, S knows the link $S{\rightarrow}C$ is with temporally high quality according to $P(i, j)$, and will retransmit the data through $S{\rightarrow}C$. The expected retransmission count is only m+1, which is close half of LB and far less than CR. This model indicates the combination of link burstiness and link spatial correlations can provide much efficient data retransmission.

In practical networks, the burstiness of each link and the correlations between link pairs are diverse. It is impossible to capture the burstiness and correlations with a preconfigured model. In the following sections, we will conduct empirical studies to show the link properties (link burstiness and link spatial correlation), and the benefit and feasibility of exploiting both properties to improve performance.

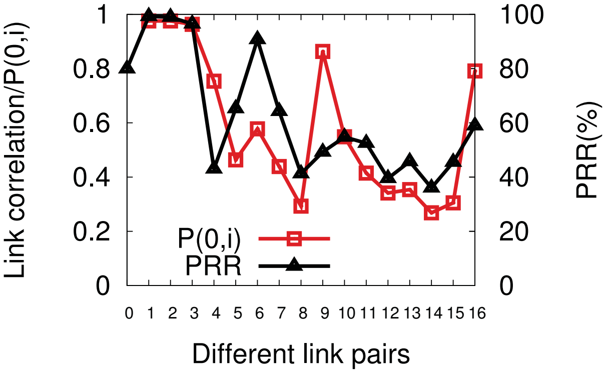


Figure 3: Correlation between link 0 and other links. *P(0,i)* denotes the probability that a packet lost in link *0* but received in link i (ranging from 1 to 16).

3.2 Impact of Link Properties

3.2.1 Link Burstiness and Link Correlation

Link burstiness is studied in a large collection of existing works [4] [19] [1]. In this experiment, we study the link burstiness in different environments, grassland, parking lot, indoor office and corridor, to depict the change of link quality indication (LQI). In the experiments, we let a sender generate packets every 100ms and forward them to a fixed relay. By logging the link's ETX according to link estimation every 10 seconds, we plot the LQIs according to $LQI{=}1/ETX$ in Figure 2. As we can see from this figure, the sharp decline of LQIs and the maintenance of continues low LQIs indicate that link burstiness exists in different environments, which is also widely studied in [4] [19] [1].

Additionally, although link correlation has been widely studied, e.g. [18] [26] [21], to further fine-grainedly test the correlation between link pairs that transmissions failed on one link while succeed on the other link, we conduct an one day experiment by setting that one node S broadcasts packets to the other 16 nodes in indoor scenario. We plot the correlation between a fixed link (0) and other links (from 1 to 16), i.e., $P(0, i)$ which is defined in Section 3.1, in Figure 3. We can see that there indeed exist several links with high packet receipt ratio (PRR), while they have low correlation with the fixed link. For example, the PRR of link 6 is 0.9, while $P(0, 6)$ is only 0.57. The inverse phenomenon exists on link 9, having low link PRR (0.49) but high $P(0, 9)$ (0.87). The analysis results indicate that once transmission failure on a link (e.g. link 0), it is feasible to deliver data packets through another link (e.g. link 9) with high correlation with the failed link, to improve data delivery reliability. This correlation also provides an evidence for the theoretical model mentioned above.

3.2.2 Impact to Retransmission

As has been inferred by [19] [18], inconsistent channel state is the main cause of link burstiness and diversity, and interference can cause inconsistent channel state at different locations. To exhibit how link burstiness and link spatial correlation impact on network performance, we conduct experiments in a 60ft×45ft office with 22 Telosb nodes, where the sink is located in a corner. Experiments are conducted using 19*th* channel overlaying the working frequencies of

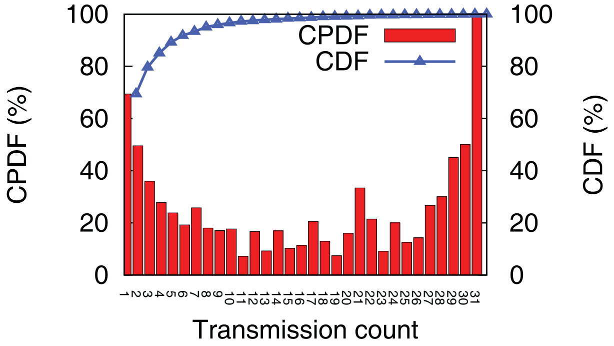
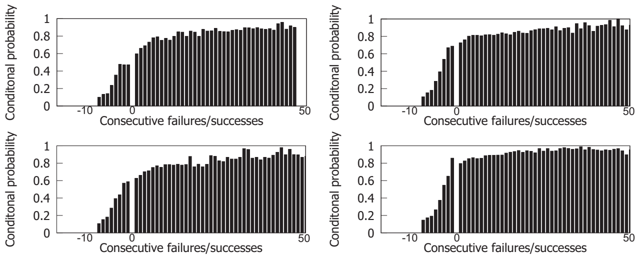


Figure 4: Conditional packet delivery functions (CPDF) that the i^{th} retransmission will success after $i-1$ consecutive failures, and the cumulative distribution function (CDF) of single-hop transmission count.



(a) Indoor link (b) Outdoor link

Figure 5: Distribution of conditional packet delivery probability changing along with the time. For each minipage, the trace data of the bottom one is collected two hours later than the top one.

WiFi. Each node generates a packet every 60 seconds and sets the RF power level to -25dBm. The maximum retransmission count threshold (RXT) is set to system default value (31) for each hop.

To show the efficiency of existing retransmission strategy, we use conditional packet delivery functions (CPDF), referred by [19], to denote the packet delivery properties that the i^{th} retransmission successes after $i-1$ consecutive failures. We denote the conditional packet delivery function as $P(i)$, which can be calculated by

$$P(i) = \frac{N_i - N_{i+1}}{N_i}. \qquad (1)$$

where N_i is the cumulative count that packets are retransmitted no less than i times. Note that $P(0)$ is also the probability that packets are transmitted only once. We plot the CPDFs in Figure 4. As the figure shows, after consecutive failures, the conditional probability sharply decreases to as little as 20%, indicating the immediate retransmission (CR in the theoretical model) is inefficient after consecutive failures.

3.3 Challenges and Problems

3.3.1 Capturing link Properties

To that extent, we further show the dynamics of the captured burstiness pattern. We construct a single-hop network with 10 nodes in indoor office and outdoor playground. To ensure the time continuity of data transmission, we set a marked node's inter-packet interval (IPI) to equal to nodes' wake-up interval, which is 512ms. The other nodes' IPI is set to 60 seconds. The retransmission count threshold is set to 100.

By logging the transmission count of each packet (more than 100,000 packets) generated by the marked node, we compute the conditional probability a packet will be successfully delivered after i consecutive failures according to Equation 1. And the conditional probability that a packet will be successfully delivered after i consecutive successes is computed by

$$P(i) = \frac{N_{i+1}}{N_i}. \qquad (2)$$

where N_i is the cumulative count that i consecutive transmissions are successful. The conditional probabilities of different scenarios are shown in Figure 5, where each bar in the histogram has a minimum of 50 data points. For each minipage, the trace data of the bottom figure is collected two hours later than the top one.

It is obviously that previous successes indicates the following transmission is likely to be successful and previous consecutive failures means the immediate retransmission will still fail to a great degree, which coincides with [19]. Furthermore, the burstiness pattern changes with time and scenarios. This phenomenon suggests the inefficiency of arbitrary consecutive retransmission, and indicates the availability of the foreseeable successful probability of the next transmission if each node embeds an online model to capture burstiness pattern.

3.3.2 Urgency of Leveraging link Properties

As discussed above, link properties have great influences on network performance. Figure 6 shows a simple example to illustrate how the widely applied protocols cope with link burstiness and correlations and to explain why the protocols do it in that way.

Node S forwards packets via relay nodes, A, J, and M, to the destination which is denoted by $Sink$. The shadow region denotes the area that can suddenly suffer from the controllable interference resulting in link burstiness and correlations. The path from D to $Sink$ is beyond the disturbed area. S generates packets each 10 seconds and forwards them to A. In our experiments, we record the retransmission count, the change of link quality, and the selection of relay nodes.

In Figure 7, during the sample sequences ranging from 200 to 900 (about 10 minutes), the controlled interference coexists with the low-power WSN. As can be seen from the figure, with the degradation of $S{\rightarrow}A$, selecting links $S{\rightarrow}B$ as the replacer according to LQI is unsatisfactory, because B is also suffering with the same interference. In this period, S repeatedly retransmits and degrades link quality wasting energy and impacting on network reliability until it changes its parent to D. However, with the end of the interference, $S{\rightarrow}D$ is indeed not the optimal link for data forwarding, but routing protocol fails to instantly know the situation because it needs long time to update the link quality of $S{\rightarrow}A$ (about 35 minutes) only exploiting routing beacons.

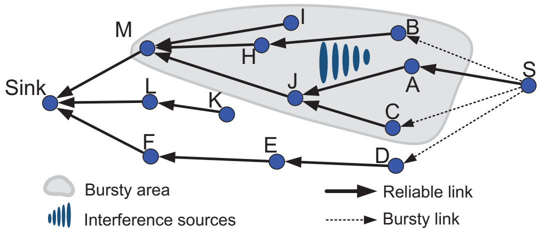


Figure 6: The network topology. Shadow region suffers from interference. A is S's parent. B, C and D are S's candidate relay nodes.

To avoid this case in retransmission-based protocols, it is necessary to consider link burstiness to pause retransmission on a severely degraded link instantly and exploit link correlations to switch to another link with temporally high quality.

3.4 Retransmission Layer

To address the problems in existing retransmission strategies, we propose a retransmission layer to exploit different link properties in order to improve the performance. Basically, to improve the performance with bursty link, existing works will adjust the wake-up interval [19]. Considering link correlations, many other works will switch between different links based on the correlation [26] [21]. We show that we can further improve the performance with a retransmission layer combining both link burstiness and spatial correlation. We build a similar retransmission layer as follows.

We allow neighbors which can provide routing progress to assist a sender to relay data packet without acknowledging the sender after 3 consecutive failures (by setting a bit in the packet header to ask for assistance). In such a case, the sender retransmits the packet to its parent node until being acknowledged or exceeding the RXT (20). To reduce the influence of duplicate packets on performance, we set nodes' IPI to 60 seconds. For simplicity, we mark it as RXTA:20. We also evaluate the consecutive retransmission-based protocol (RXT:20) and dynamic switch-based forwarding protocol (DSF-RXT:20) for comparison. Dynamic switch-based forwarding protocol exploits link burstiness to forward data packet to a candidate relay node according to link quality and wake-up schedule once a transmission fails.

We plot the network reliability and average single-hop transmission count in Figure 8. With such a retransmission layer, RXTA:20 reduces almost 17.1% transmissions compared with RXT:20 and reduces 38.6% compared with DSF-RXT:20 as Figure 8(b) shows, while increases the average PRR from 96.53% and 94.89% to 99.05%, respectively, shown as Figure 8(a).

This experiment sheds new light on the benefit of exploiting both link burstiness and link correlation. It is obvious that when the current used link is severely degraded, relaying packets through another temporally high quality link can provide much more efficient transmission and higher network reliability. The experimental results also motivate us to capture both link burstiness and link spatial correlation between link pairs in retransmission layer.

The empirical study also raises a problem how to efficiently capture link burstiness pattern and link correlation online without introducing extra computational and control overhead. In the next section, we will discuss online models to capture these network characteristics transparent to existing protocols.

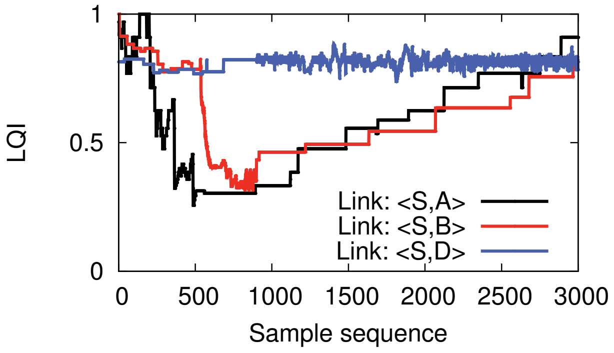
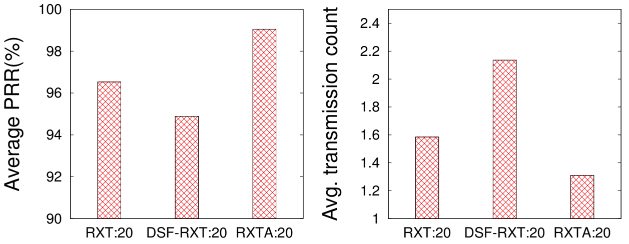


Figure 7: Routing behavior without considering link burstiness and correlation. During the smaple sequences ranging from 200 to 900, node A and B suffer from the controlled interference, and D is out of the disturbed area.



(a) Network reliability

(b) Transmission count

Figure 8: Network reliability and average single-hop transmission count of protocols: repeatedly retransmitting (RXT:20), exploiting link burstiness (DSF-RXT:20), and exploiting both link burstiness and correlation (RXTA:20).

4. DESIGN OF RXLAYER

From the empirical studies, we know the exist of link burstiness and correlation, and the benefit of exploiting them in retransmission layer. This section presents the design of RxLayer. In what follows, the design principles and core ideas of RXLayer are presented in Section 4.1. We then introduce the basis of RxLayer – the online models – in Section 4.2 and Section 4.3. The strategy how RxLayer works is discussed in Section 4.4. And then we discuss the embedding of RxLayer into forwarding protocols in Section 4.5.

4.1 Design Insign and Principles

Generally, retransmission strategy works as follows. Given a network, each node forwards data packets to the destination through a deterministic relay (deterministic forwarding) or a forwarding set (dynamic switch-based forwarding) determined by network communication layer. Once a transmission is not acknowledged, the sender will retransmit the packet until it receives an ACK. With data transmission, it also updates link quality. When the successive retransmission count exceeds the maximum threshold, the packet will

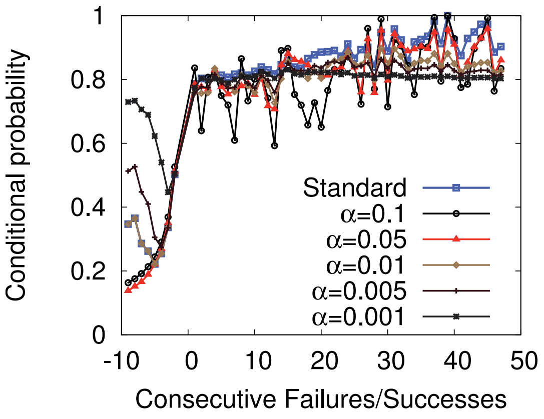


Figure 9: The computed conditional probability of P_i by setting different values to α, the *standard* one denotes the computed result by Equation 1.

be dropped. To provide reliable transmission, network protocols should endeavor to maintain and update link quality.

However, from the empirical studies, we know retransmission-based routing protocols are inefficient encountering link burstiness and correlation. The feature provides the key insights we used to exploit link burstiness and correlation when building RxLayer. Roughly speaking, a sender should instantly perceive the severely degraded link quality, and then, according to link properties captured online, RxLayer decides when to pause after consecutive failures and retransmits through a candidate link with temporally high quality.

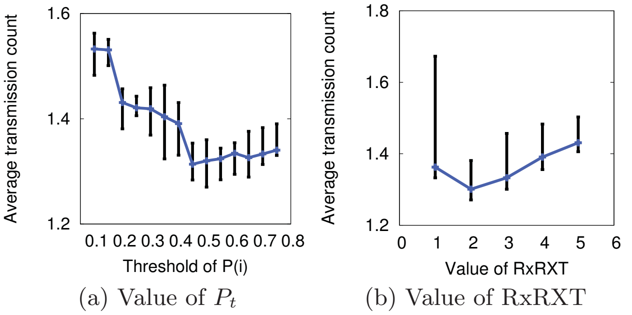
4.2 Link Burstiness Model

In this section, we present a conditional probability model by which a sender perceives the severe degradation of the current used link and gets a proper stoppint point of consecutive retransmission over a bursty link. To construct this model, each node maintains a probability table, marked as *PSRT*, to record the probability of successful retransmissions of each target link. *PSRT* contains M entries. Each entry marked as E_i, $0 \leq i < M$, denotes a conditional probability, P_i, that a packet will be successfully delivered after i consecutive failures. Initially, we consider links to be independent, and the probability of each individual trial is initialized to the inverse of link quality. Then, it can be formalized as: $P_i = \frac{1}{ETX}$.

When a sender transmits a data packet to its next-hop node, RxLayer will update the value of corresponding *PSRT* entry after each trial. Here, we use moving average to update P_i shown as follow.

$$P_i^{new} = \begin{cases} (1-\alpha) * P_i^{old} + \alpha & if\ success \\ (1-\alpha) * P_i^{old} & if\ fail \end{cases} \qquad (3)$$

P_i^{old} is the accumulated conditional probability, and P_i^{new} is the updated value. α is a parameter for making a tradeoff between the adaptability to network dynamics and accuracy. To find the optimal value of α, we use 5 hours trace data to compute conditional probability by setting different values to α. By Comparing these results with the standard case which is computed by Equation 1, as Figure 9 shows, it is rational to set α to 0.05 with the best fitting. Note that the optimal value of α changes with scenarios, a robust and efficient method should dynamically adjust the parameter. Some machine learning methods [3] could be used to adjust α. However, for simplicity, we use a fixed experimental value in our implementation and won't discuss it in detail in this paper.



(a) Value of P_t (b) Value of RxRXT

Figure 10: The average transmission count changing with the value of P_t or RxRXT.

By online updating *PSRT* entries, RxLayer can build a prediction model fastly like Figure 5 showing. To transmit a packet, RxLayer checks the conditional probability (P_i) by inputting the count (i) of consecutive failures. If P_i is less than the predefined threshold (P_t, which will be discussed in Section 4.4 in detail), the sender should pause and choose a candidate link with temporally high quality to deliver the packet.

When choosing a new link, maybe, it is the first time to be used, which means the link is considered to be independent by ignoring the burstiness property. In RxLayer, we define the retransmission count threshold over a new link no more than a constant RxRXT, which is discussed in the following section. Moreover, note that choosing a new relay node won't trigger parent change event. Parent change is controlled by network communication layer only according to link quality. Hence, it can be avoided that parent change is triggered by sharply decreased link quality when a burst of interference results in consecutive failures even though the link is long-term reliable.

4.3 Link Correlation Model

As mentioned before, by getting an stoppint point, RxLayer should select another link to deliver packets. Here, we discuss the model for capturing link correlation. A metric is presented in this section to represent the degree of link correlation. It's desirable for the metric to measure the probability that transmissions failed on one link while succeed on the other one. For simplicity, it is marked as ω, which is a 3-tuple quantity, defined on a transmitter, S, and two receivers, i and j, that can overhear packets from S, as:

$$\omega(S,i,j) = \begin{cases} P_{i/j}^{(S)}(0/1) & P_i^{(S)}(0) > 0 \\ PRR(j) & P_i^{(S)}(0) = 0 \end{cases} \qquad (4)$$

where, $P_{i/j}^{(S)}(0/1)$ is the probability, when S transmits, that a packet succeed on link $S{\rightarrow}j$ given that it failed on link $S{\rightarrow}i$, and $P_i^{(S)}(0)$ is the proportion of packets failed on link $S{\rightarrow}i$. If the two links, $S{\rightarrow}i$ and $S{\rightarrow}j$ are independent, then $\omega(S,i,j)$ is the packet delivery ratio of link $S{\rightarrow}j$.

To capture ω between link pairs, RxLayer adopts uniform broadcast sequence number (BSN) to differentiate broadcast

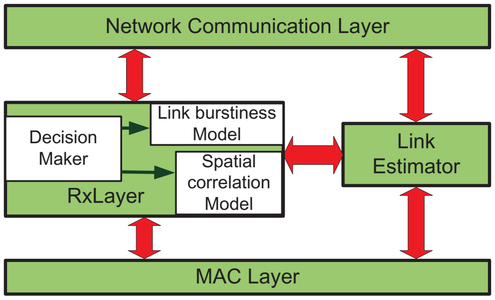


Figure 11: RxLayer in the protocol stack.

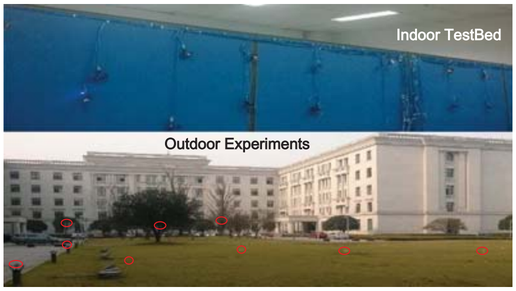


Figure 12: Indoor testbed with 22 TelosB nodes and outdoor scenario with 30 TelosB nodes.

Table 1: average gap of link correlation between the result of moving average and statistical result.

θ	gap	θ	gap
0.01	0.051	0.02	0.053
0.03	0.048	0.04	0.039
0.05	0.032	0.06	0.024
0.07	0.029	0.08	0.031
0.09	0.049	0.1	0.062

Table 2: parameters used in RxLayer.

M	10	α	0.05
θ	0.06	P_t	0.45
RxRXT	2		

packets from data packets. To transmit a broadcast packet, low-power WSN guarantees each neighbor has a chance to hear the packet. By recording and exchanging the bitmap of neighbors' BSNs, each node can capture the ω between different link pairs distributedly. Note that the BSN information is attached to routing beacon, so there is no additional control overhead.

For efficiency, we arrange a circular queue of the size of 2 bytes for each neighbor to record the BSNs of all heard broadcast packets. Each bit corresponds to a broadcast packet. When a bit was set, it indicates the corresponding packet has been heard. To denote all available broadcast packets (BSN ranging from 0 to 255), we use an index to record the BSN of the earliest received broadcast packet, and a point to indicate the index of the corresponding bit.

Through routing beacons, a node can get to know the correlation between link pairs like Figure 3 showing. Based on the information, a node can calculate ω between any link pairs according to Equation 4. Note that a 2 bytes-long circular queue only records limited BSNs. Hence, the result of each calculation of ω associated with two links is a partial view of the overall correlation. To fully show the correlation between link pairs, we use moving average to update ω as follow,

$$\omega(S,i,j) = (1-\theta) * \omega^{old}(S,i,j) + \theta * \omega^{new}(S,i,j) \tag{5}$$

where $\omega^{old}(S,i,j)$ is the accumulated correlation between link $S{\to}i$ and $S{\to}j$, and $\omega^{new}(S,i,j)$ is the calculation result of the latest BSN sets. We guarantee each sequence is exploited only once. The using of moving average not only reduces memory overhead, but also provides a tradeoff between the adaptability to network dynamics and the accuracy. To get an appropriate value to θ, we conduct an experiment by collect the feedback BSNs from two links for an entire day. We analyze the correlation between two links exploiting the collected BSNs according to the statistical approach like Equation 4 and moving average like Equation 5, respectively. The statistical approach exploits all history data to compute the correlation, while the moving average only exploits the most recent BSNs to update the correlation. We assume the statistical correlation is very precise although it may not be exactly true. By setting different values to θ, we calculate the gap of link correlation between the statistical result and the moving average result according to

$$gap = \sqrt{\frac{\sum(\omega_i - M_i)^2}{n}},$$

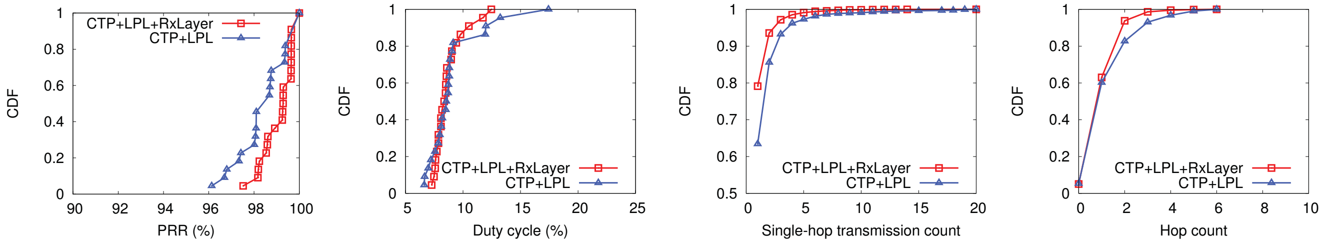
where M_i is the statistical correlation and ω_i is the related correlation calculated by moving average after sender broadcasting the i^{th} routing beacon. The values of θ and the corresponding gaps are listed in Table 1, which indicates 0.06 is the optimal value of θ with the minimal gap, then we set θ to 0.06 in RxLayer with the minimum gap.

4.4 Decision-Maker Rule

In this section, we discuss the rules when should RxLayer pause after encountering transmission failure and how to select a candidate relay.

As mentioned above, when the conditional probability, P_i, of successful transmission is lower than the predefined threshold P_t, RxLayer triggers the change of relay. In the indoor experiments, by integrating the state-of-the-art protocols with RxLayer and setting different values to P_t, we compute the average transmission count of each trial as Figure 10(a) shows. Overall, setting P_t to 0.45 obtains the most benefit. As mentioned above, the maximum retransmission count threshold via a link to a replacer is no more than RxRXT. By setting different values to RxRXT in indoor experiments under interference, the average transmission count is plotted in Figure 10(b). According to the figure, we set RxRXT to 2 in RxLayer with the minimum expected transmission count. In our additional experiments which are not plotted in this paper, by setting P_t to 0.45 and RxRXT to 2, the network performance is relatively better than setting to other values on the whole.

In the context of hop-by-hop retransmission-based protocols, deterministic forwarding [9] and dynamic switch-based forwarding [10] are two basic types protocols. For the former, network layer supports a determined relay node, but for the latter, network layer supports a forwarding set which



(a) Network reliability (b) Energy consumption (c) Single hop transmission count (d) Number of hops travelled

Figure 13: Performance of multi-hop indoor testbed experiment with 22 TelosB nodes. By embedding RxLayer into CTP+LPL, network performance is slightly improved.

Algorithm 1 Decision-Maker Algorithm

```
Input:
    Given $P_t \in [0,1]$; $Threshold \leftarrow$RXT; Retransmission
    count $i$=0; RxRXT; $Sum \leftarrow$0; Forwarding set $F$;
Iteration:
 1: while $P_i > P_t$ & $i < Threshold$ & $Sum < RXT$ do
 2:    Retransmission;
 3:    if Acknowledged then
 4:       Update link quality; Upgrade $P_i$;
 5:       Return.
 6:    else
 7:       Update link quality; Degrade $P_i$
 8:       $i \leftarrow i$+1; $Sum \leftarrow Sum$+1;
 9:    end if
10: end while
11: if $Sum = RXT$ then
12:    Drop the packet;
13:    Return.
14: else
15:    if $Card(F)$=1 then
16:       Selecting the replacer with largest $\omega$;
17:    else
18:       Considering both relay nodes' wake-up time schedule and $\omega$ between each of them and the previous unsuccessful link;
19:    end if
20:    $Threshold \leftarrow RxRXT$; $i \leftarrow$0;
21:    Goto: Iteration;
22: end if
```

contains a sequence of relay nodes scheduled by wake-up time. To support these protocols without distinction, we consider the relay node or a sequence of relay nodes provided by network layer as forwarding set uniformly.

The work process of RxLayer is presented in Algorithm *Decision-Maker*. The inputs to the algorithm are the measured threshold (P_t) of conditional probability, the maximum retransmission count threshold (RXT) provided as a system parameter, the retransmission count (i) over the current used link, the predefined maximum retransmission count to any candidate replacer (RxRXT), and the forwarding set provided by network layer.

When a packet is delivered down from network communication layer, where the relay node is set to be the first element of the forwarding set, the retransmission count (i) is set to 0, and the maximum retransmission count threshold (*Threshold*) is set to RXT, RxLayer checks whether the conditional probability P_i is larger than P_t and i is less than *Threshold*. If so, RxLayer delivers the packet to the next hop node. If the packet is acknowledged, RxLayer updates the link quality, upgrades P_i according to Equation 3, and returns the success to the upper layer; otherwise, updates link quality, degrades P_i, and retries to deliver the data packet. If the total retransmission count exceeds RXT, RxLayer drops the packet and returns related information to upper layer. Beyond that, no matter getting a stoppint point ($P_i \leq P_t$) or the retransmission count over the link to a replacer exceeding RxRXT ($i \leq Threshold$), RxLayer selects another replacer of the current relay node and starts a new iteration until the packet is successfully delivered or dropped. Note that *Card(F)* (line 15 in Algorithm *Decision-Maker*) denotes the selection of the optimal replacer from forwarding set F. *Card(F)* returns 1 if there exists a candidate replacer in the set, otherwise, returns 0. For the deterministic forwarding protocols, *Card(F)* always returns 0. There can have a variety of methods to schedule the forwarding set and select a replacer, however, we won't go into detail on this in the paper.

4.5 RxLayer Embedding

Rxlayer is designed as a generic middleware to assist a wide range of existing routing protocols (e.g, CTP [9], DSF [10]) and be compatible with other energy efficient MAC layers such as low power listening (LPL) [16] and low power probing (LPP) [20]. To do that, we insert RxLayer beneath network communication layer and above MAC layer as shown in Figure 11. Before discussing our evaluation, we give the parameters used by RxLayer in Table 2.

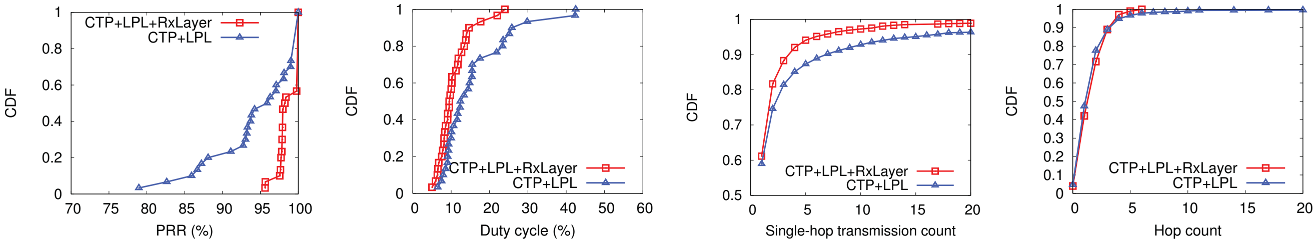
5. EVALUATION

In this section, we evaluate RxLayer through indoor and outdoor experiments. We compare the network performance of the stat-of-the-art protocols (CTP [9] with LPL [16]) embedding RxLayer with that without embedding RxLayer.

We use PRR as the indicator of the network reliability. The energy consumption is measured by the ratio of radio on time to total time (duty cycle). Moreover, we use the average transmission count to measure the cost of delivering a packet. Additionally, we use parent change count to measure the network dynamics.

5.1 Implementation

We implemented RxLayer on TelosB nodes in TinyOS



(a) Network reliability (b) Energy consumption (c) Single hop transmission count (d) Number of hops travelled

Figure 14: The performance of multi-hop outdoor experiment with 30 TelosB nodes. By embedding RxLayer into CTP+LPL, network performance is greatly improved.

2.1.1. The prototype implementation of RxLayer combines with CTP, a standard collection protocol for sensor networks shipped with TinyOS. CTP uses Four Bit Link Estimator [8] as its link estimation component. The RAM and ROM consumption of RxLayer are 833 bytes and 6656 bytes, respectively. RxLayer is not bound to any specific routing protocol. It can be easily integrated with other routing protocols, such as L2 [4] and DSF [10]. In the following experiments, we embed RxLayer with CTP above LPL.

5.2 Evaluation Setup

We evaluate the performance of protocols on indoor testbed with 22 Telosb nodes and outdoor networks with 30 Telosb nodes as Figure 12 shows. In the indoor experiments, we set the transmission power of CC2420 as level 1 to ensure multi-hop communication (maximum hop is 6). The outdoor network is deployed around our office build where there are 10 to 18 WiFi APs at different locations and the work frequency of some APs overlays with the 19*th* channel of ZigBee. Uniformly, we set our networks' work channel to 19, the transmission power of CC2420 is set to level 10, the maximum retransmission count threshold is set to the TinyOS default setting 31, and each node generates packets with a fixed IPI (60 seconds). The set of LPL's wake-up interval is 512ms. We conduct experiments using the protocol stack (CTP with LPL) and that by embedding RxLayer in the same scenarios. Each experiment lasts for at least 5 hours and is repeated no less than three times.

5.3 Network Reliability

We compare the network reliability of CTP+LPL with that by embedding RxLayer in indoor testbed experiments (as Figure 13 shows) and outdoor experiments (as Figure 14 shows). Figure 13(a) and Figure 14(a) show the distribution of each node's average PRR. By embedding RxLayer into CTP+LPL, network reliability (PRR) is averagely improved by 1.53% in the indoor experiments, and by 7.82% in the outdoor experiments.

As the figures show, by embedding with RxLayer, nodes' PRRs are relatively high and distributed in a smaller range. It is because channel state at different nodes is different and RxLayer exploits the correlation to forward packets via the temporally stable links when the current used link is severely degraded. No matter a sender or a receiver is distributed in interference area, by perceiving the shapely degraded link quality according to online conditional probability models, RxLayer utilizes the best available link if there exists one, so as to decrease the effect of local interference on network performance and improve channel utilization.

5.4 Energy Consumption

We also plot the distribution of energy consumption (duty cycle) in Figure 13(b) and Figure 14(b). In the indoor experiments, there is no significant difference between the energy consumption of CTP+LPL and that with RxLayer, but in the outdoor experiments, by embedding with RxLayer, the average duty cycle is significantly decreased from 19.3% to 10.4%. For the former, the little improvement of energy efficiency is due to the small-sized testbed reducing the correlation between link pairs, which is also indicated from the similar distributions of single hop transmission count plotted in Figure 13(c). RxLayer can benefit from the correlation. By embedding with RxLayer in the outdoor experiments, the average single hop transmission count is decreased by 1.35 times (decreased from 3.72 times to 2.37 times) as Figure 14(c) shows.

5.5 Transmission Cost

By embedding RxLayer into CTP with LPL, the single-hop transmission cost does have been reduced no matter in our indoor experiments (Figure 13(c)) or outdoor experiments (Figure 14(c)). The average single hop transmission count of indoor experiments is reduced from 1.74 times to 1.31 times (reduced by about 24.7%) by embedding RxLayer, and reduced from 3.72 times to 2.37 times (reduced by about 36.3%) in outdoor experiments mentioned above. The decreased transmission count makes nodes stay in sleep state, so as to further save energy consumption. Moreover, the reduced transmissions can further decrease packet collisions caused by hidden terminal and reduce suppressed transmission caused by exposed terminal.

The strategy of RxLayer is switching from data forwarding over a severely degraded link to retransmission using another temporally stable link. So, the increased transmission hop count associated with the change of next hop must be considered. We plot the distribution of hop count of indoor and outdoor experiments in Figure 13(d) and Figure 14(d), respectively. The average hop count of outdoor experiments by embedding RxLayer is slightly higher than that without embedding RxLayer, but the latter is with a long tail which means some packets are delivered through many hops to sink. We believe that the misadvice from network communication layer to link estimator mentioned in Section 3.3.2 gives rise to mistakenly degrading link quality and the using suboptimal links, inducing network dynamics and bringing about the long tail. RxLayer can effectively avoid this case by exploiting both link burstiness and correlation between link pairs.

6. CONCLUSIONS

Energy efficiency and network reliability are important issues in low-power wireless sensor networks. To address this problem, nodes work in duty cycle and retransmission strategy is widely applied in routing protocols. However, consecutive retransmission is energy inefficient by ignoring the change of channel state at receiving end.

This paper attempts to improve a wide range of existing deterministic routing protocols and dynamic forwarding protocols by designing a general supporting layer, called RxLayer, for adaptive retransmission by exploiting link burstiness and link correlation. By integrating RxLayer with the state-of-the-art protocols, it significant improves network performance.

Acknowledgment

This work is supported in part by the National Key Technology R&D Program under grant No. 2013BAH33F02, the NSFC program under Grant 61202359, and the NSFC program under Grant 61373146.

7. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, and et al. Link-level measurements from an 802.11b mesh network. In *Proceedings of SigComm*, pages 121–132. ACM, 2004.

[2] M. Alizai, O. Landsiedel, J. Link, and et al. Bursty traffic over bursty links. In *Proceedings of SenSys*, pages 71–84. ACM, 2009.

[3] J. Braams. Fitting curves to data using nonlinear regression: a practical and nonmathematical review. *The FASEB Journal*, 1(5):365–374, 1987.

[4] Z. Cao, Y. He, and Y. Liu. L2: lazy forwarding in low duty cycle wireless sensor networks. In *Proceedings of InfoCom*, pages 1323–1331. IEEE, 2012.

[5] A. Cerpa, J. Wong, M. Potkonjak, and et al. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *Proceedings of MobiHoc*, pages 414–425. ACM, 2005.

[6] D. Couto, D. Aguayo, J. Bicket, and et al. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of MobiCom*, pages 134–146. ACM, 2003.

[7] W. Dong, Y. Liu, Y. He, and et al. Measurement and analysis on the packet delivery performance in a large scale sensor network. In *Proceedings of InfoCom*, pages 2679–2687. IEEE, 2013.

[8] R. Fonseca, O. Gnawali, K. Jamieson, and et al. Four-bit wireless link estimation. In *Proceedings of HotNets*. ACM, 2007.

[9] O. Gnawali, R. Fonseca, K. Jamieson, and et al. Collection tree protocol. In *Proceedings of SenSys*, pages 1–14. ACM, 2009.

[10] Y. Gu and T. He. Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *Proceedings of SenSys*, pages 321–334. ACM, 2007.

[11] G. Hwang and D. Cho. Fast retransmission mechanism for voip in ieee 802.11e wireless lans. In *Proceedings of VTC*, pages 1090–3038. IEEE, 2004.

[12] K. Jamieson and H. Balakrishnan. Ppr: partial packet recovery for wireless networks. In *Proceedings of SigComm*, pages 409–420. ACM, 2007.

[13] S. Kim, R. Fonseca, P. Dutta, and et al. Flush: a reliable bulk transport protocol for multihop wireless networks. In *Proceedings of SenSys*, pages 351–365. ACM, 2007.

[14] Y. Liu, Y. He, M. Li, and et al. Does wireless sensor network scale? a measurement study on greenorbs. In *Proceedings of InfoCom*, pages 873–881. IEEE, 2011.

[15] L. Mo, Y. He, Y. Liu, and et al. Canopy closure estimates with greenorbs: sustainable sensing in the forest. In *Proceedings of SenSys*, pages 99–112. ACM, 2009.

[16] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of SenSys*, pages 95–107. ACM, 2004.

[17] B. Raman, K. Chebrolu, S. Bijwe, and et al. Pip: a connection-oriented, multi-hop, multi-channel tdma-based mac for high throughput bulk transfer. In *Proceedings of SenSys*, pages 15–28. ACM, 2010.

[18] K. Srinivasan, M. Jain, J. Choi, and et al. The κ-factor: inferring protocol performance using inter-link reception correlation. In *Proceedings of MobiCom*, pages 317–328. ACM, 2010.

[19] K. Srinivasan, M. Kazandjieva, S. Agarwal, and et al. The β-factor: measuring wireless link burstiness. In *Proceedings of SenSys*, pages 29–42. ACM, 2008.

[20] Y. Sun, O. Gurewitz, and D. Johnson. Ri-mac: a receiver initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of SenSys*, pages 1–14. ACM, 2008.

[21] S. Wang, S. Kim, Y. Liu, and et al. Corlayer: a rransparent link correlation layer for energy efficient broadcast. In *Proceedings of MobiCom*, pages 51–62. ACM, 2013.

[22] A. Woo and D. Culler. Evaluation of efficient link reliability estimators for low-power wireless networks. Technical Report UCB/CSD-03-1270, EECS Department, University of California, Berkeley, 2003.

[23] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of SenSys*, pages 14–27. ACM, 2003.

[24] X. Wu, M. Liu, and Y. Wu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *Proceedings of IPSN*, pages 1–12. ACM/IEEE, 2012.

[25] T. Xiang, Z. Chi, F. Li, and et al. Powering indoor sensing with airflows: a trinity of energy harvesting, synchronous duty-cycling, and sensing. In *Proceedings of SenSys*, pages 73:1–73:2. ACM, 2013.

[26] T. Zhu, Z. Zhong, T. He, and et al. Exploring link correlation for efficient flooding in wireless sensor networks. In *Proceedings of NSDI*, pages 49–64. USENIX, 2010.