Exploiting Sender-based Link Correlation in Wireless Sensor Networks

Junghyun Jun[1,2], Long Cheng[2,3], Liang He[2], Yu Gu[2] and Ting Zhu[4]

[1]Indian Institute of Technology Ropar, India
[2]Singapore University of Technology and Design, Singapore
[3]State Key Lab of Software Development Environment, Beihang University, Beijing 100191, China
[4]University of Maryland, Baltimore County, USA

***Abstract*—Link correlation in wireless sensor networks has recently attracted a considerable amount of attention in the research community. Various pioneer works have empirically demonstrated the existence of link correlations and designed novel network protocols to exploit such link correlations. While all existing works focus on the correlated receptions at multiple receivers from a single sender, in this work we empirically demonstrate another type of link correlation, called *sender-based link correlation*. For sender-based link correlation, we observe wireless links from multiple senders to a single receiver are also correlated. Based on this observation, we design a two-tiered data forwarding scheme for improving the energy efficiency of unicast in the network. At the micro-level, individual nodes reduce their transmission energy consumption by temporarily switching to a new forwarder or suppressing the current transmission with the knowledge of link correlations. At the macro-level, we schedule the ordering of transmission times among neighboring nodes so that the gains from all link correlation information in the network is maximized. Through trace-driven emulations and large scale simulations, we demonstrate that our design reduces data retransmissions by an average of 12% when compared with already highly energy efficient ETX-based protocols.**

I. Introduction

In wireless sensor networks, hundreds of protocol designs have assumed that packet receptions among different links are independent. Recently, researchers have discovered that correlated shadowing and interference can lead to correlated packet losses between neighboring receivers and empirically validated the existence of link correlation [1], [2]. For example, multiple wireless links being positively correlated is due to strong interference source affecting all its nearby wireless links. Different models have been proposed to capture the link correlation. In [1], Conditional Packet Reception Probability (CPRP) has been proposed to capture the link correlation. It states that if a packet is received by one receiver with a lower packet reception ratio, most of the time this packet is also received by the receiver with higher packet reception ratio. Here, packet reception ratio (PRR) is a common metric used to measure the quality of links. In [2], the receiver-based link correlation has been further divided into two categories (i.e., positive and negative link correlations) by introducing the κ factor. They also suggest that human or obstacles blocking a set of links together could result in the positive link correlation. For the negative link correlation, authors in [2] hypothesize that both human mobility and the CSMA/CA mechanism of nearby high power interferers are the potential causes. An experiment was also conducted to demonstrate that link correlation indeed changes under these scenarios.

All of these previous studies on wireless link correlation focus on receiver-based link correlation, i.e., for packets transmitted from the same sender, packets received by different receivers are correlated. Little systematic study has investigated the wireless link correlation from multiple senders to the same receiver. To fill this gap, we conducted extensive empirical study and discovered that packet receptions at a common receiver are correlated when multiple senders transmit packets to the receiver within small inter-packet time intervals. Based on the empirical data, we build a mathematical model to capture this sender-based link correlation and design a data forwarding scheme for improving performance of data collection under the sender-based link correlation. In summary, our contributions are as follows:

- Through empirical studies in our own testbed and the Indriya testbed [3], we first show the existence of *sender-based link correlation*. Then, we demonstrate how to utilize the sender-based link correlation for improving energy efficiency of data collection protocol by reducing number of retransmissions.
- We design a novel two-tier (micro and macro) correlation-aware forwarding scheme. At the micro-level, individual nodes dynamically switch to a new forwarder or suppress the current transmission with the knowledge of sender-based link correlations. At the macro-level, we schedule the sequence of transmission orderings among neighboring nodes so that it maximizes the gains from utilizing all correlated links within the network.
- We show that the problem of maximizing the gains from utilizing all correlated links is NP-hard, and we propose both the centralized and distributed heuristic solutions.
- We conduct both simulation and trace-based emulation to verify the performance gain of utilizing the sender-based link correlation. Evaluation results show that our proposed correlation-aware forwarding scheme can effectively reduce the number of total transmissions with small additional packet delay and extra probing packets.

978-1-4799-6204-4/14 $31.00 © 2014 IEEE
DOI 10.1109/ICNP.2014.67

The rest of this paper is organized as follows. Section II presents related work. Section III introduces the sender-based link correlation. Section IV describes the network model and the main correlation-aware forwarding protocol design. Evaluation results are shown in Section VII. Section VIII concludes this paper.

II. Related Work

Srinivasan et al. [2] first experimentally demonstrate the existence of link correlation in wireless networks and authors in [4] present an on-line algorithm to compute the link correlation in a fully distributed fashion. This existence of link correlation has attracted increased research interests in recent years. In general, they can be divided into two major applications, link correlation based broadcast and link correlation based unicast.

In link correlation based broadcast area, Zhu et al. [1] present the first extensive study to exploit spatial link correlation phenomena for communication improvement. They propose the concept of collective ACK and design a dynamic forwarding technique to improve the efficiency of reliable flooding operations. Guo et al. [5] solved the ACK implosion problem of traditional flooding and reduced energy consumption by proposing the Correlated Flooding, in which the packet receptions at multiple correlated nodes are acknowledged by only one ACK. In [6], Alam et al. extend the link correlation aware data dissemination to support multi-packet flooding by exploiting network coding. Finally, Shuai et al. [7] designed CorLayer, a supporting layer which improves wide range of existing broadcast algorithm. The CorLayer is a connected dominating set where each pair of links in the dominating set is strongly correlated. Due to this strong correlation, a single broadcast in CorLayer can reach many neighboring nodes. Therefore, it reduces total number of retransmissions. In link correlation based unicast area, authors in [8] analyze the opportunistic routing gain in the presence of link correlation and introduce a link correlation aware opportunistic routing scheme, which improves the performance by exploiting the diverse uncorrelated forwarding links.

While previous works focus on link correlation between receivers, we focus on the link correlation between senders to a common receiver. Principles of link correlation are same for both sender-based and receiver-based link correlation. However, the key difference between these two lies in the utilization of spatial-temporal correlation. The receiver-based link correlation shows correlation between two links at a same time instance but at two different receivers. In other words, it is only considering the spatial correlation between two links. Whereas, sender-based link correlation is considering both spatial and temporal correlation between two links. It shows correlation between two links at a common receiver but at two different time instances. We cannot apply existing receiver-based correlation protocols in case of sender-based link correlation because they make independent forwarding decisions between any two consecutive transmissions while our protocol schedules the sequence of transmission orderings among neighboring senders in an attempt to maximize the utilization of temporal link correlation.

Srinivasan [9] has revealed that wireless links are bursty at the timescale of about hundreds of milliseconds. That is, data delivery losses and successes over a link are highly correlated for small inter-packet intervals, which we call the temporal link correlation. Based on the observation from Srinivasan's work, [10] [11] show that energy efficiency and latency can be improved by utilizing the temporal link correlation in existing routing protocols. Packets can either be forwarded over bursty links if they offer better routing progress than other long-term stable links [10], or can be suppressed to minimize the worst case over bursty links [11]. In another field of domain, Liu et al. [12] propose to exploit link correlation to predict instantaneous link quality for energy efficient data transmission in mobile cellular networks, i.e., on the railway platform for cargo transportation. Since in cellular networks, for a mobile terminal, the received signal strength has a significant correlation with its locations, the transmission of an approaching mobile terminal can be scheduled by utilizing the channel status measured by current mobile terminal with the cellular tower. Different from [10], [11], [12], we experimentally reveal for the first time in this work the phenomenon of *spatial-temporal link correlation* between senders to a common receiver, and demonstrate how to utilize this new phenomenon for wireless sensor network performance improvement.

III. Sender-based Link Correlation

Different from *receiver-based link correlation*, our new observation of the *sender-based link correlation* essentially captures the fact when multiple senders are transmitting packets consecutively to a common receiver, the packet receptions at the receiver for different senders are also correlated. Specifically, the sender-based link correlation is defined as follows:

Definition Sender-based link correlation is defined as $P(s_1, r|s_2)$, a conditional probability of a high PRR sender s_1 successfully transmits a packet to a receiver r at time $t(s_1)$, given the condition that the packet is successfully transmitted by a lower PRR sender s_2 to the receiver r at time $t(s_2)$, where $0 < t(s_1) - t(s_2) < \epsilon$ and ϵ is the inter-packet interval.

Sender-based link correlation between two senders $\langle s_1, s_2 \rangle$ can be easily calculated at the common receiver r since it has the reception results of both sender s_1 and s_2. The receiver r calculates sender-based link correlation as follows:

$$P(s_1, r|s_2) = \frac{\sum_i^w B_{s_1,r}(i) \& B_{s_2,r}(i)}{B_{s_2,r}(i)} \qquad (1)$$

$B_{s_1,r}(i)$ is a bit representing node s_1's ith transmission result to receiver r. For example, $[1110]_{s_1}$ and $[0111]_{s_2}$ represents a reception results of four consecutive transmissions from s_1 and s_2, respectively. Here bit value of 1 represents successful transmission and value of 0 means failure. Then, the sender-based link correlation is calculated by $P(s_1, r|s_2) = \frac{1\&0+1\&1+1\&1+0\&1}{0+1+1+1} = 66.6\%$. A similar method is introduced in [1] for calculating receiver-based link correlations.

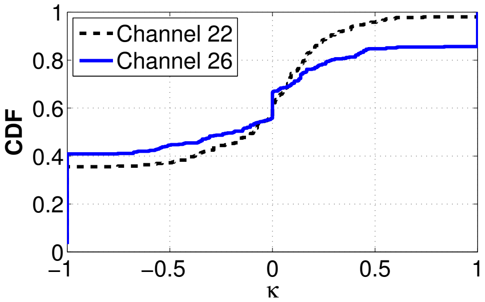


Fig. 1. Sender-based link correlation (κ) of neighboring pairs at receiver n64 in Indriya testbed at different channels

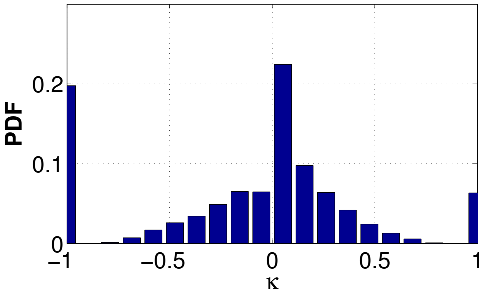


Fig. 2. Sender-based link correlation (κ) of 100 nodes in different locations of Indriya testbed

A. Existence of Sender-based Link Correlation

In order to validate the existence of sender-based link correlation, we perform a set of experiments at the Indriya Tesbed [3] and collect data transmission traces for a time period of 9 hours. During the experiments, 100 nodes are randomly selected from the Indriya Testbed (with 140 nodes) and each node requests each of its neighbors to transmit a total number of $10,000$ packets in a round robin fashion, with a time interval of 10 ms between two consecutive transmissions. The 10 ms interval is introduced to avoid the packet collisions at the receiver. We use the κ-factor metric introduced in [2] to measure the sender-based link correlation between sender pairs having common receivers.

Fig. 1 shows neighbors of node n64 in Indriya who are experiencing high link correlations. This affirms that the packet transmissions from a pair of senders to a common receiver is indeed correlated. In fact, from the measurement results, we found that about 30% of the sender pairs in Indriya testbed result in a strong link correlation where $\kappa > 0.5$ or $\kappa < -0.5$. Their link correlation distribution is shown in Fig. 2. One interesting observation from Fig. 2 is that about 20% of the sender pairs were negatively correlated with $\kappa = -1$, which indicates that for these pairs $\langle s_1, s_2 \rangle$, $P(s_1, r|s_2) = 0$ if s_2 successfully transmits to r. From the experiment results, the duration of strong link correlation where $\kappa > 0.5$ or $\kappa < -0.5$, is also measured and its distribution is shown in Fig. 3. We found that 25% of link correlation last more than 20 sec and some last more than 100 sec. However, since the duration of link correlation changes dynamically, the link correlation must be updated periodically. In order to obtain link correlation among neighboring nodes, several beacon messages need to be broadcasted and then quality of individual links can be estimated and the correlation between every pair of links can be measured. To achieve such efficient network wide flooding and data sharing, we adopt the recently introduced Glossy [13], which can very quickly flood several beacon messages and share the link correlation results to every node by multiple Glossy flooding (e.g. it takes less than 10ms for a network of 100 nodes).

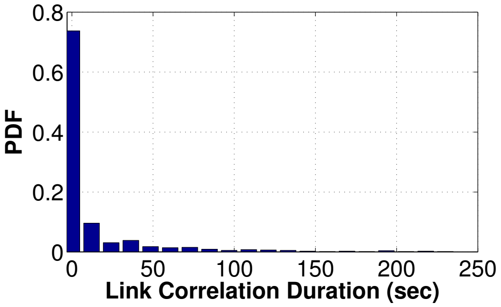


Fig. 3. Distribution of sender-based link correlation duration

B. Sender-based Link Correlation in Depth

From the experiment, we identify three types of link correlations, namely the *positive link correlation*, the *negative link correlation*, and the *packet-loss link correlation*.

1) Positive Link Correlation: For a given sender pair, if one sender's successful transmission indicates the successful transmission of the other sender, we refer to it as the *positive link correlation*. Assume the link quality from sender s_1 and s_2 to a common receiver r is $P(s_1, r)$ and $P(s_2, r)$, respectively, and $P(s_1, r) > P(s_2, r)$. Then $\langle s_1, s_2 \rangle$ are positively correlated at node r if $P(s_1, r|s_2) > P(s_1, r)$. In other words, the expected number of transmissions needed for s_1 to forward a packet to node r is smaller if node s_1 transmits after node s_2 sends its packet successfully to receiver r first.

2) Negative Link Correlation: In contrast to positive link correlation, we say the sender pair $\langle s_1, s_2 \rangle$ is negatively correlated at their common receiver r if $P(s_1, r|s_2) < P(s_1, r)$. In other words, when sender s_2 successfully transmit to receiver r transmission from sender s_1 to receiver r will have a higher probability to fail. This implies the expected number of transmissions needed for sender s_1 to forward a packet to r is larger if node s_1 transmits shortly after s_2 sends its packet successfully to node r first.

3) Packet-Loss Link Correlation: Another interesting observation is that the transmission failures of two senders to one common receiver are also correlated, denoted as packet-loss link correlation. Again, we quantify this link failure correlation by conditional probability of a packet reception failure. Assume $P(s_1, r) > P(s_2, r)$, the conditional packet reception failure probability $P(\overline{s_2}, r|\overline{s_1})$ is the probability that s_2 fails to transmit a packet to the receiving node r, given the condition that the transmission from s_1 to r fails. Opposite to the positive/negative link correlations, the packet-loss link correlation is conditioned on the transmission results of the sender with a higher link quality.

Unlike the link correlation for successful packet transmissions, packet-loss link correlation is much harder to directly measure in practice because packet failures can only be detected by missing packet sequences at a receiver side and a link with poor link quality could have many consecutive packet losses.

Fortunately, the conditional packet reception probability for packet-loss link correlation can be derived from positive link correlation. For example, if $P(s_1, r|s_2) = 95\%$ and link qualities of sender s_1 and sender s_2 to a common receiver r is 85% and 70%, respectively, then,

$$P(\overline{s_2}, r|\overline{s_1}) = \frac{P(\overline{s_2}) - P(s_1) + P(s_1, r|s_2)P(s_2)}{P(\overline{s_1})} = 76.7\% \tag{2}$$

where $\overline{s_2}$ means r fails to receive a packet from s_2. In this case, $1 - P(\overline{s_2}, r|\overline{s_1}) = 23.3\%$ indicates the probability for node r to successfully receive a packet from s_2, given that r fails to receive a packet from s_1. The detailed derivation of Eqn.(2) for $P(\overline{s_2}, r|\overline{s_1})$ is relegated to Appendix.

IV. Correlation-Aware Forwarding Protocol

The observation of sender-based link correlation motivates us to design a *correlation-aware forwarding protocol* to reduce number of total transmissions with small overheads.

A. Overview

Basically, our design consists of two tiers of packet forwarding control: the *micro-level control at individual nodes* and the *macro-level control for a group of nodes*.

At the micro-level, based on the link correlation and the results of real-time data transmissions at neighboring nodes, individual nodes either (i) forward its packet to the original forwarder, (ii) temporarily switch its forwarder, or (iii) suppress its current transmission temporarily.

As the forwarding decisions of individual nodes are conditional on the transmission results of other senders, different sending orders for a group of nodes could lead to fundamentally different performances. Therefore at the macro-level control, we aim to schedule the sending sequence of nodes in the network such that the link correlations are fully utilized and the performance gain can be further improved.

B. Network Model

The network is represented as a graph $G = (V, d, E)$, where V is the set of sensor nodes, node d represents a sink, and $|V| = n$. A directed link $s \rightarrow r$ belongs to E if and only if r can receive or overhear the packet transmitted from s. $P(s, r)$ represents the link quality of $s \rightarrow r$ and it can be estimated at node r based on the reception results of both hello messages and real traffics from its neighbors [14], [15], [16]. For example, $[0110]_s$ represents four consecutive transmission results from sender s to r. Then, $P(s, r)$ is estimated to be 50%. The reception results are also used by the receiver r to calculate the sender-based link correlation of their neighbors based on Eqn.(1) and receiver r periodically broadcasts or piggybacked its sender-based link correlation on ACK packets. Sender s recognizes the availability of correlated links by either overhearing this ACK packet or receiving beacon messages from the receiver. As long as most transmissions do not result in the collisions our method can be applied to improve the performance of any wireless sensor network. Therefore, our proposing method is not tie to any specific MAC layer protocols. However, to simplify our illustration of macro-level design, we assume schedule based TDMA as our MAC layer protocol.

We consider a periodic data collection in wireless sensor network as a motivating application for our protocol design. Here, all sensors periodically wakeup and forward data to sink node. During the collection period every node stays awake to receive and to forward data toward the sink. For reliable data collection, we assume an ETX-based energy efficient tree rooted at the sink is constructed based on G [14]. However, different from the traditional ETX-based data collection, we allow *multiple potential forwarders* at each node. For a given sink d, each node s maintains a possible forwarder set F_s. The sender s forwards the packet to $r \in F_s$ if it can minimize $ETX(s, d)$. $ETX(s, d)$ is the expected number of transmissions from s to sink d.

C. Correlation-Aware ETX

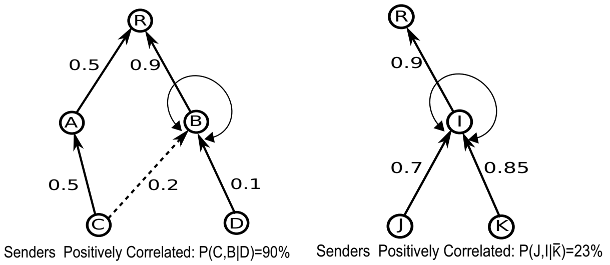
For energy efficient data collection, a routing protocol selects forwarding nodes based on the expected number of transmissions (ETX) to sink d. Specifically, each node $s \in V$ maintains a possible forwarder set F_s and forwards the packet to $r \in F_s$ if it can minimize $ETX(s, d)$. However, different from traditional ETX metric proposed in [14], [17], we propose the concept of correlation-aware ETX (*cETX*) by incorporating the link correlation. The *cETX* is adopted as the key metric in our design of the correlation-aware data forwarding protocol.

Consider two correlated senders s_1 and s_2 at a common receiver $r \in F_{s_1} \cap F_{s_2}$, where F_x represents the set of all possible receivers of x, and further assume the link quality from s_1 to r is better than s_2 to r (i.e., $P(s_1, r) > P(s_2, r)$), the *cETX* of an ordered sender pair $\langle s_1, s_2 \rangle$ to receiver r is defined as follows.

Definition $cETX(s_1, r|s_2)$ or $cETX(s_1, r|\bar{s_2})$ is the expected number of transmissions of a packet from s_1 to receiver r given that a transmission from s_2 to r just before s_1's transmission has succeeded or failed, respectively.

V. Micro-Level: Dynamic Forwarder Switching

At the micro-level, every forwarding node updates its cETX by exploiting three types of sender-based link correlations introduced in Section III and notifies the changes in the cETX to its predecessors. Then, every sender makes forwarding decisions based on the cETX metric from itself to the sink. In


R
0.5
0.9
A
B
0.5
0.2
0.1
C
D
Senders Positively Correlated: P(C,B|D)=90%
R
0.9
I
0.7
0.85
J
K
Senders Positively Correlated: P(J,I|K̄)=23%

(a) Positive link correlation (b) Packet-loss link correlation

Fig. 4. Examples of applying sender-based link correlation

what follows, forwarding nodes updating cETX and senders making forwarding decision for each type of sender-based link correlations are presented in Section V-A, V-B, and V-C, respectively. We then discuss the overheads in Section V-D and V-E.

A. Utilizing Positive Link Correlation

For two positively correlated senders $\langle s_1, s_2\rangle$ at a common receiver r with $P(s_1, r) > P(s_2, r)$, a successful packet transmission from s_2 to r indicates the temporary link quality improvement of link $s_1 \to r$. More specifically, the link quality of $s_1 \to r$ improves from $P(s_1, r)$ to $P(s_1, r|s_2)$. Thus the updated temporal *cETX* from s_1 to the destination d through receiver r is:

$$\begin{aligned} cETX^r(s_1, d|s_2) &= cETX(s_1, r|s_2) + ETX(r, d) \\ &= \frac{1}{P(s_1, r|s_2)} + ETX(r, d) \end{aligned} \quad (3)$$

The superscript r in Eqn.(3) is adopted to show the *cETX* is obtained when s_1 selects r as the forwarder. If the link quality improvement is sufficiently high such that $cETX^r(s_1, d|s_2) < ETX(s_1, d)$, node s_1 selects r as its forwarder.

Fig. 4(a) shows an example of utilizing the positive link correlation to reduce the number of transmissions, where the solid lines represent the ETX-based routing structure and the dashed lines represent those links that are physically existed but are not selected in the ETX-tree. The link quality is labelled for each link. A is the original forwarder of C and the transmissions from C and D to B are positively correlated with $P(C, B|D) = 90\%$. Without considering the link correlation, the shortest ETX path from C to the destination R is $C \to A \to R$ with an ETX value of $\frac{1}{0.5} + \frac{1}{0.5} = 4$. After considering the positive link correlation between C and D, and when D has just successfully transmitted a packet to B, the probability for C to successfully transmit a packet to B is temporarily increased from $P(C, B) = 0.2$ to $P(C, B|D) = 0.9$, and thus the minimal ETX value from C to R is temporarily reduced from 4 to $cETX^B(C, R|D) = \frac{1}{0.9} + \frac{1}{0.9} \approx 2.22$. Thus C switches its forwarder to B for this transmission.

B. Utilizing Negative Link Correlation

Let us consider the case that the successful transmissions from two senders $\langle s_1, s_2\rangle$ are negatively correlated at receiver r ($P(s_1, r) > P(s_2, r)$) and r is s_1's forwarder in the ETX tree. The successful packet transmission from s_2 to r indicates the temporal degradation of the link quality from s_1 to r ($P(s_1, r|s_2) < P(s_1, r)$). The receiver r sends an ACK packet in response to s_2's successful transmission. Sender s_1 overhears this ACK packet and recognizes that node s_2 has successfully transmitted a packet to receiver r. Sender s_1 also learns negative link correlation between itself and s_2 by extracting piggybacked link correlation information from the ACK packet. When still choosing r as the forwarder, the *cETX* of s_1 can be calculated in the same way as in Eqn.(3).

However, different from applying positive link correlation by switching the forwarder, if the link quality from s_1 to r degrades substantially, s_1 temporarily delays its packet transmission until its negatively correlated s_2 fails its transmission to receiver r, which indicates the vanishing of the temporary negative link correlation.

The reason for not adopting dynamic forwarder switching in the negative correlation case is that the ETX tree is already the optimal routing structure if no link correlation is considered. Thus, although the quality of link $s_1 \to r$ in the ETX tree may temporarily degrade, which can be inferred based on the identified negative link correlations, the performance of the path from r to sink d in the ETX tree still outperforms other paths in most of the time. In other words, the dynamic forwarder switching in applied the negative correlation, it may adopt a path that is much worse than the original solution. For this reason, temporarily delaying s_1's transmission would be a better choice, especially when considering the temporary feature of the negative link correlation.

C. Utilizing Packet-Loss Correlation

Exploiting the packet-loss correlation is another way of utilizing the positive link correlations. Let us again assume the successful packet transmissions of two senders $\langle s_1, s_2\rangle$ are positively correlated at a common receiver r and $P(s_1, r) > P(s_2, r)$, then the failure of packet transmission at s_1 implies the temporal link quality degradation of link $s_2 \to r$.

Specifically,

$$\begin{aligned} cETX^r(s_2, d|\bar{s_1}) &= cETX(s_2, r|\bar{s_1}) + ETX(r, d) \\ &= \frac{1}{P(s_2, r|\bar{s_1})} + ETX(r, d) \end{aligned} \quad (4)$$

Similar to utilizing negative link correlation, if the transmission from s_1 to r fails we delay the packet transmission at s_2 until r can successfully receive the transmission from s_1.

Empirically, we observe that the packet-loss correlation has a much larger impact on reducing the data collection energy consumption than directly utilizing the positive link correlation. This is because the ETX tree offers the most reliable path for individual nodes according to the link qualities, and thus the further reduction of energy consumption by exploiting even better links via positive link correlations is limited.

For example, the increase of temporal link quality from 90% to 95% when a node utilizes its positive link correlation would not significantly reduce the number of transmissions in the network. However, the identification of packet-loss correlation reveals significant temporal link quality degradation and its proper utilization can reduce the number of unsuccessful packet transmissions. For example, as shown in Fig. 4(b), if $\langle K, J\rangle$ are positively correlated at I with $P(K, I|J) = 95\%$, $P(K, I) = 85\%$, and $P(J, I) = 70\%$, then $P(J, I|\bar{K}) = 1 - P(\bar{J}, I|\bar{K}) = 1 - \frac{0.3-0.85+0.95\times0.7}{0.15} \approx 23.3\%$ (Eqn.(2)), which means a $\frac{0.7-0.23}{0.7} = 67.14\%$ link quality degradation. Consequently, the ETX value from J to R is increased from $\frac{1}{0.7} + \frac{1}{0.9} \approx 2.54$ to $\frac{1}{0.23} + \frac{1}{0.9} \approx 5.46$. This indicates almost 3 additional transmissions if the packet-loss correlation is not utilized.

D. Delay Overhead by Suppressing Transmission

When utilizing negative link correlation or packet-loss correlation, s_1 or s_2 delays its transmission until its correlated node s_2 fails or s_1 successes. The expected packet delay would be same as the time for the effect of negative or packet-loss correlation to vanish from two senders $\langle s_1, s_2\rangle$.

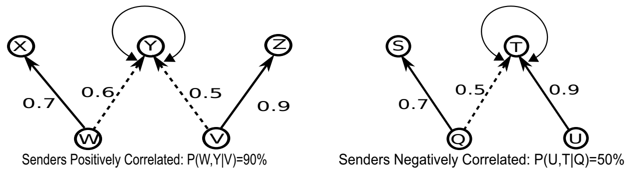
Intuitively, suppressing the packet transmission should increase the packet delay. However, our emulation and simulation results on the packet delay in Fig. 12 of Section VII reveal that the additional end-to-end packet delay due to suppressing the packet transmission is negligible compared to the end-to-end packet delay when the packets are transmitted without utilizing link correlation. This is due to our design at macro level which helps to improve the links with poor link quality rather than links with good link quality by ordering the sending sequence of nodes based on the link correlation.

E. Control Overhead Analysis

Nodes recognize the availability of correlated links by either overhearing ACK packets or receiving beacon messages from other nodes, and the latter introduces additional communication overhead. We do not count hello messages into the overhead since periodically sending probe packets has been already required by other protocols [14], [18] to measure link quality or to improve the robustness of routing structure. Next, we analyze the overhead of our design in detail.

Obtaining the correlation information by overhearing or receiving ACK packets is feasible for nodes in most case. For the example shown in Fig. 4(a) where D is the child of B, whenever B successfully receives the packet from D, B piggybacks $cETX^B(C, R|D)$ to the ACK package to D, and C can learn this correlation by overhearing this ACK.

The two cases requiring the beacon messages are demonstrated in Fig. 5. In Fig. 5(a), W and V are correlated at a common receiver Y, which is not the parent of either of them in the tree. Therefore, in order to inform the existence of the positively correlated links $W \to Y$ and $V \to Y$, Y sends beacons to W whenever it overhears the transmissions from V to Z. Similarly, in Fig. 5(b), T can overhear the transmission from Q to S but cannot send ACKs to Q such that U can overhear. Therefore, T sends beacons to inform U about Q's negative correlation with U at T.


X
Y
Z
W
V
0.7
0.6
0.5
0.9
Senders Positively Correlated: P(W,Y|V)=90%
S
T
Q
U
0.7
0.5
0.9
Senders Negatively Correlated: P(U,T|Q)=50%

(a) Y is not the parent of W or V (b) T is not the parent of Q

Fig. 5. Scenarios in which beacon messages are needed

However, we argue that the communication overhead is still small for the following two reasons. First, beacon messages are much smaller in size compared to the actual data packets. Second, we minimize the number of beacon messages by these means: (i) parents send out beacon messages if certain nodes in their neighbor table can surely benefit from the successful transmission of its correlated sender; (ii) the parents combine the link correlation information and broadcast them to every neighbor instead of unicast to each node; (iii) adaptive beaconing is adopted, which adjusts the frequency of beacon messages according to how fast the link correlation changes.

VI. Macro-Level: Effective Sending Sequence

While at the micro-level design we reduce energy consumption at individual nodes by utilizing their link correlations, at the macro-level design we focus on how to improve the utilization of link correlations at the network level.

The macro-level control is inspired by the key fact that all link correlations are conditional events. For example, if $\langle s_1, s_2\rangle$ are 95% positively correlated at their common receiver r with $P(s_1, r) = 0.6$ and $P(s_2, r) = 0.5$, then the transmission from s_1 to r would be successful with 95% probability if s_1 knows that r has just received a packet from s_2. In this case, if we ensure s_2 always transmits before s_1, then s_1 can increase the utilization of its correlation with s_2. In contrast, the correlation of $\langle s_1, s_2\rangle$ can never be utilized if the transmission from s_1 to r always happens before s_2.

Formally, for a network with n nodes, we define a sending sequence X_n as $X_n = \{x_1, x_2, \ldots, x_n\}$ $(x_i \in V)$, and thus for any X_n, node x_i can compute its *cETX* to the destination according to its correlation with the nodes transmitting before it, namely x_j $(1 \leq j < i)$.

Definition Given a sending sequence X_n and assume two nodes $x_i, x_j \in X_n$ are correlated at their common receiver r, *cETX* from x_i to destination d depends on the sending sequence of x_i and x_j. Denote $cETX^r_{X_n}(\langle x_i, x_j\rangle, d)$ as the cETX from x_i to d through successor r with sending sequence X_n:

$$cETX^r_{X_n}(\langle x_i, x_j\rangle, d) =$$

$$\begin{cases} cETX(\langle x_i, x_j\rangle, r) + ETX(r, d) & j < i\ ,\ x_i, x_j \in X_n \\ ETX(x_i, d) & \text{otherwise} \end{cases}$$

where

- $cETX(\langle x_i, x_j\rangle, r) = cETX(x_i, r|x_j) = \frac{1}{P(x_i, r|x_j)}$ if $\langle x_i, x_j\rangle$ are positively/negatively correlated at receiver r;

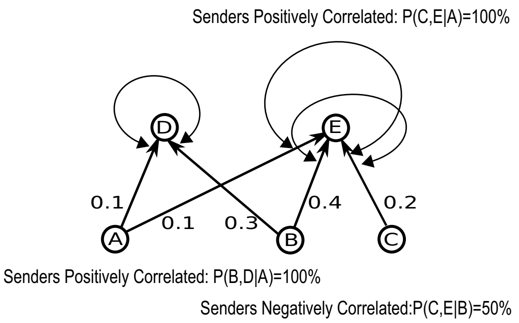

Senders Positively Correlated: P(C,E|A)=100%
D
E
0.1
0.1
0.3
0.4
0.2
A
B
C
Senders Positively Correlated: P(B,D|A)=100%
Senders Negatively Correlated:P(C,E|B)=50%

Fig. 6. An example demonstrating the execution of the centralized heuristic

- $cETX(\langle x_i, x_j\rangle, r) = cETX(x_i, r|\bar{x_j}) = \frac{1}{P(x_i, r|\bar{x_j})}$ if $\langle x_j, x_i\rangle$ are packet-loss correlated at receiver r.

Thus the *correlation gain* of a given sending sequence X_n is defined as following.

Definition The correlation gain $g_i(X_n)$ of node x_i in sending sequence X_n is the expected ETX reduction of x_i to the destination d if the link correlations of x_i with other nodes are utilized according to X_n:

$$g_i(X_n) = \sum_{j=1}^{i}\sum_{r=1}^{n} P(x_j, r)(ETX(x_i, d) - ETX^{r}_{X_n}(\langle x_i, x_j\rangle, d)). \tag{5}$$

where the outer summation is to consider all the correlations between x_i and the nodes before it in X_n, and the inner summation is to consider all the possible common receivers for each pair of correlated nodes x_i and x_j.

The total gain when adopting X_n as the sending sequence is the summation of $g_i(X_n)$ for all $x_i \in X_n$:

$$G(X_n) = \sum_{i=1}^{n} g_i(X_n).$$

Thus the objective of the macro-level optimization is to identify an optimal sending sequence $X^* = \{x_1^*, x_2^*, \dots, x_n^*\}$:

$$X^* = \arg\max_{X_n \in \mathbf{M}} G(X_n), \tag{6}$$

where $\mathbf{M}$ is the set of all possible sending sequences and $|\mathbf{M}| = n!$.

Since the cardinal of $\mathbf{M}$ in Eqn (6) increases factorially with n, directly solving this problem is not feasible. In fact, if we simplify the problem by only utilizing the link correlation between node x_i and x_{i-1} in X_n, the problem can be formulated as the well-known NP-hard *Travelling Salesmen Problem*, where the cities are the senders and the weights of directed edges are the link correlation gains. Thus instead of pursuing its optimal solution, we propose a heuristic approach to obtain both centrally and distributedly the good sending sequence $\bar{X} = \{\bar{x}_1, \bar{x}_2, \dots \bar{x}_n\}$ in polynomial time.

A. Centralized Heuristic

Essentially, the centralized heuristic is a bottom-up greedy algorithm that assigns nodes to $\bar{X}_n$ one by one. Starting with an empty $\bar{X}_n$, we continuously insert the node that can provide the maximum link correlation gain to other unassigned nodes to $\bar{X}_n$. Initially, set $Y \leftarrow V$ where V is the set of all nodes. For $\forall y_i \in Y$, the expected gain from y_i sending its packet before all the other nodes in Y is:

$$g_i(\bar{X}_n) = \sum_{y_j, y_k \in Y\setminus\{y_i\}} (ETX(y_j, d) - cETX^{y_k}_{\bar{X}_n}(\langle y_i, y_j\rangle, d)).$$

The centralized heuristic calculates the $g_i(\bar{X}_n)$ for all $y_i \in Y$, and selects the y^* that has the maximum gain. Then y^* is inserted at the end of the sending sequence $\bar{X}_n \leftarrow \{\bar{X}_n, y^*\}$, and Y is updated by $Y \leftarrow Y\setminus\{y^*\}$. The above process repeats until all nodes are inserted to $\bar{X}_n$. Fig. 6 presents an example of 3 nodes $V = \{A, B, C\}$ to illustrate the centralized heuristic, where $\langle B, A\rangle$, $\langle C, A\rangle$, and $\langle C, B\rangle$ are correlated. During the first iteration, each node sets its sending index to be the first one in the sequence and computes its link correlation gain that can be provided to others. $g_A(\bar{X}_3) = 0.63$ due to link correlation gain provided to B and C. $g_B(\bar{X}_3) = 0$ because the PRR of B to parent D and E is higher than A and C to their parent. $g_C(\bar{X}_3) = 0.1$ due to the gain provided to B. Therefore, we assign $\bar{x}_1 = A$ because $g_A(\bar{X}_3)$ is the maximal. During the second iteration, $Y = \{A, B, C\}\setminus\{A\} = \{B, C\}$, and we assign $\bar{x}_2 = C$ because $g_C(\bar{X}_3) = 0.1 > g_B(\bar{X}_3) = 0$. Finally, during the third iteration, B is added to the sending sequence and the centralized heuristic returns the sending sequence of $\bar{X}_3 = \{A, C, B\}$.

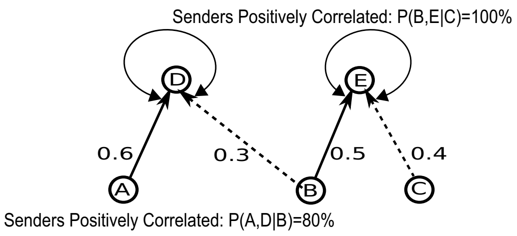

Senders Positively Correlated: P(B,E|C)=100%
D
E
0.6
0.3
0.5
0.4
A
B
C
Senders Positively Correlated: P(A,D|B)=80%

Fig. 7. An example demonstrating the Distributed Forward Assignment algorithm

The complexity of the centralized heuristic is $\mathcal{O}(n^4)$ because each recursive calculation of $g_i(\bar{X}_n)$ takes $\mathcal{O}(n^2)$ to calculate the link correlations of $\mathcal{O}(n)$ node pairs, which is repeated for $\mathcal{O}(n)$ times.

B. Distributed Heuristic

For wireless sensor networks, due to their distributed feature, directly applying the centralized heuristic may either be infeasible or introduce significant overhead. To facilitate the practical implementation, we further develop a distributed heuristic algorithm based on the centralized solution.

In the distributed heuristic, every node records and calculates the link correlations between its physical neighbors and periodically piggybacks the link correlation information such that its neighbors can update the most current link correlation from overhearing the piggyback messages. For a specific node s, upon receiving the link correlation information from its potential parents, s makes a union of all the neighbors of all its potential parents: $U_s = \bigcup_{f \in F_s} N_f$, and thus U_s is the set of nodes to which s can possibly provide link

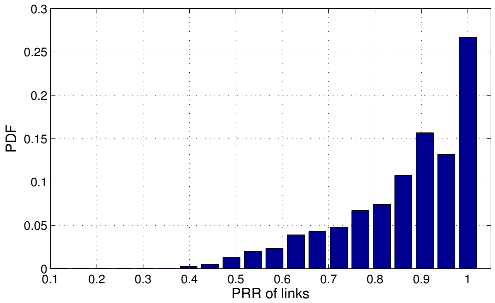


Fig. 8. Link quality distribution of CTP tree.

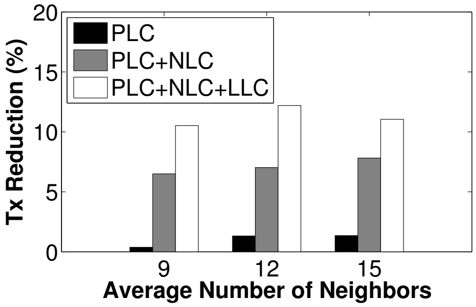


Fig. 9. Emulation result showing Transmission efficiency

correlation gain. s updates its local sending order sequence $\hat{X}^s = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{|N_f|}\}$ based on the received correlation information by following the same assignment process as in the centralized heuristic. Then s schedules its transmission according to this local sending order sequence.

An example of this distributed heuristic is shown in Fig. 7. According to the physical connectivity of nodes, $U_A = \{A, B\}$, $U_B = \{A, B, C\}$, and $U_C = \{B, C\}$.

Each node runs the distributed heuristic based on the link qualities and their correlations, and the locally computed sending order sequences are $\hat{X}^A = \{B, A\}$, $\hat{X}^B = \{C, B, A\}$, and $\hat{X}^C = \{C, B\}$. In this way, whenever they have packets to send, C will take the first time slot while A and B will compete for the second slot [19].

VII. Evaluation

In this section, we first evaluate our link correlation-aware forwarding design on trace-driven emulation using the collected trace at the Indriya testbed. The emulation is used, instead of testbed implementation, to obtain meaningful data because occurrences and duration of link correlation is random but it is not possible to occupy continuous long hours in the Indriya testbed. Then to investigate the protocol performance under diverse network settings, we also perform large-scale simulations with varying degrees of link correlations and varying link qualities in the network. The PRR of link in the tree is shown in Fig. 8.

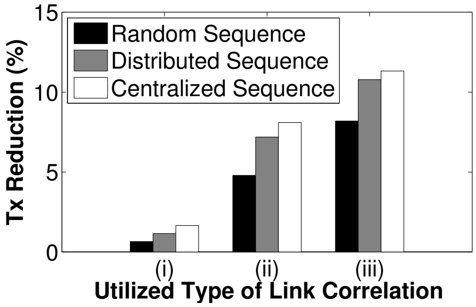


Fig. 10. Emulation result showing effect of optimized sending order sequence. (i), (ii), and (iii) indicate PLC, PLC+NLC, and PLC+NLC+LLC, respectively

A. Trace-driven Emulation

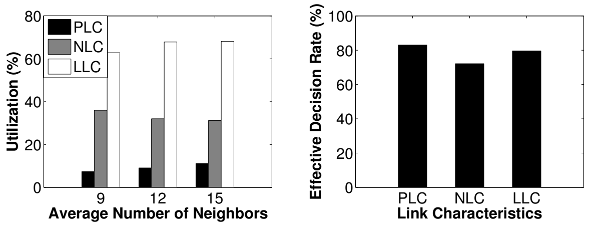
1) Trace-driven Emulation Setup: We use the trace-driven emulation to evaluate our design at micro and macro level. The results are averaged over 30 random trials. The topology of connected network with different densities is randomly generated for each trial. The packet traces from 100 Telosb sensors in Indriya Testbed are assigned randomly to each link in the network. Every node in the network generates 80 packets such that emulation finishes before the trace runs out and relay packets from other nodes to the sink. The successful packet receptions over a link follow the success packet transmissions from the trace assigned to that link.

After the network is generated, we build an energy efficient data forwarding tree to a sink (root) based on the ETXs. The tree is basically the shortest path tree to the root using accumulated ETX.

2) Performance Evaluation at the Micro-level: Fig. 9 represents ETX-based collection protocol (CTP) [14] against our design if positive link correlation (PLC), negative link correlation (NLC), and packet loss link correlation (LLC) are utilized together, controlling the packet forwarding at micro-level, which provides an overall transmission saving of 10-12 % from CTP protocol. The results show that increasing node density provides more opportunities for reducing the retransmission by utilizing link correlations because there exist more correlated links. When the node density is as high as 15 the network favors the CTP protocol to generate highly energy efficient trees with low average ETX. In Fig. 8, the traces we collected from Indriya testbed show 70% of links in the testbed have PRR of above 85%. Therefore, we can expect only minor improvement in the link quality due to utilizing PLC and LLC. However, our design still effectively utilizes NLC and provides transmission saving of 11%.

Fig. 9 also shows that almost 90% of energy saving is from dynamically delaying the packet forwarding based on utilizing NLC and LLC. Transmitting packets by temporarily switching the forwarding node based on utilizing PLC provides comparably lower energy saving because the initial forwarding tree is an energy efficient ETX-tree.

3) Performance Evaluation at the Macro-level: Fig. 10, represents percentage of transmission reduction by optimizing



(a) Ratio of Link Correlation being Utilized

(b) Making Effective Decision

Fig. 11. Trace-based emulation results showing the system behavior with different node density

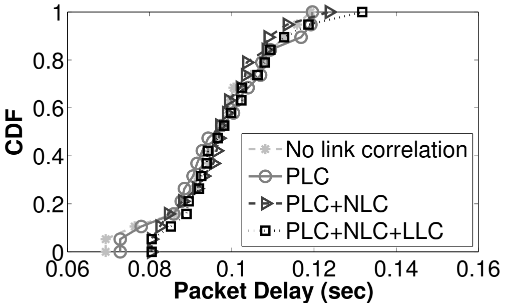


Fig. 12. Emulation results showing end-to-end packet delay for utilizing different link correlations

the sending order sequence of nodes in the network. When the packets are transmitted in random order, the link correlation is opportunistically utilized if its correlated neighbors transmit their packets first. Therefore, we use this random sending order sequence as a baseline to compare the performance gain of our centralized and distributed heuristic approaches which generate the sending order sequence to increase the utilization of sender-based link correlation.

The results in Fig. 10, show that we can further reduce transmissions when the nodes follow the sending order sequence obtained from our heuristic, compared to the random sending order sequence. Obviously, the gain from the sending order sequence is maximized if we utilize all three link correlation characteristics (PLC+NLC+LLC) together in the heuristic.

Also, our distributed heuristic solution, which uses local sending order sequence, performs closer to centralized heuristic in the case of utilizing all three link correlation characteristics (PLC+NLC+LLC).

4) System Insight: Fig. 11(a) represents how often each type of link correlation is utilized by our design to make forwarding decisions. Fig. 11(b) shows that, when the link correlation is utilized our design makes effective decisions about 80% of the time by switching forwarding node or delaying the current transmission by utilizing sender-based link correlation. The switching decision is considered effective if the packet is not lost due to switching to the forwarding node with smaller expected number of transmission. The delaying decision is considered effective if the packet was forwarded successfully after the delay but would otherwise have been lost. These two results show that, on average, 60% of the packet forwarding decisions are made based on the LLC and around 80% of the time it is an effective decision.

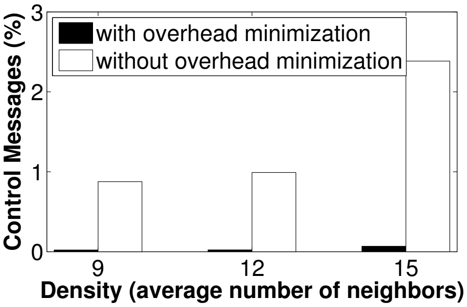


Fig. 13. Emulation results showing percentage of control messages used for utilizing link correlation compared to number of total message transmitted

Fig. 12 shows that additional end-to-end packet delay due to utilizing link correlation. The result of utilizing link correlation increases the end-to-end packet delay by a average of 0.005s. This is less than one consecutive time slot in our emulation, which is negligible. In fact, the largest addition packet delay by utilizing link correlation (PLC+NLC+LLC) is 0.01s, which is added to the packet with minimum packet delay. For many delay sensitive applications, increasing minimum packet delay is not as significant as increasing worst case packet delay. One possible reason behind this result is that our design at macro level helps to improve the links with poor link quality rather than links with good link quality by ordering the sending sequence of nodes based on the link correlation. Also, our design of utilizing sender-based link correlation, can be made more efficient if we apply our three overhead minimization methods, as discussed in Section IV to reduce the number of control messages transmitted directly to the nodes by biggybacking the control messages on the data packets and neighboring nodes can obtain link correlation information by overhearing these packets. Fig. 13 shows the percentage of control messages used to directly inform availability of link correlation is around 2.5% of total number of packets transmitted when density is 9. When the overhead minimization is applied in our design, we can see that the number of control messages actually sent out is below 0.2% of total message transmissions in the network.

B. Simulation

In the simulation, we evaluate the performance of our design under different expected link correlations and different distributions of PRR of links.

1) Simulation Setup: In the simulation, 100 nodes are randomly generated ($150m \times 150m$) and communication links are assigned to node pairs. The PRR of each link is assigned randomly. Given a random network topology, we generate correlated links as correlated multi-variate bernoulli random numbers as described in [20], [8]. We generate a bitmap of 1000 bits sequence for each sender to capture the correlated events. The results are averaged over 30 random trials.

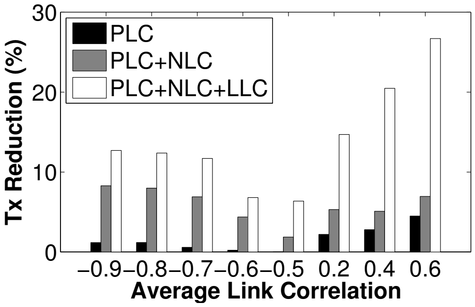


Fig. 14. Simulation results showing the effect of different link correlations at micro-level

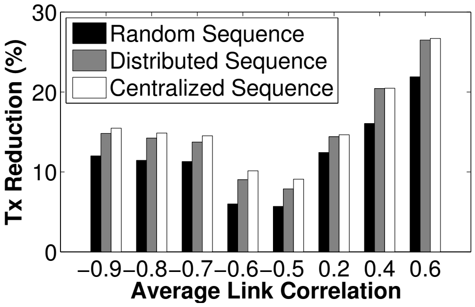


Fig. 15. Simulation results showing the effect of different link correlations at macro-level

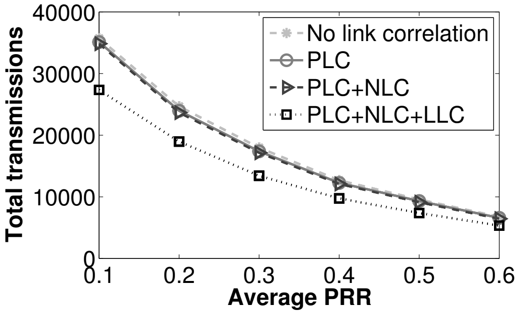


Fig. 16. Simulation results showing the effect of average PRR at micro-level

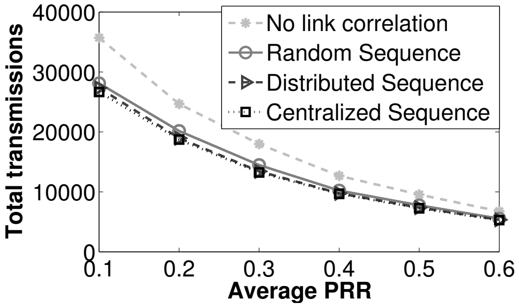


Fig. 17. Simulation results showing the effect of average PRR at macro-level

2) Simulation Results: Fig. 14 shows the percentage of transmission reduction by utilizing the PLC, PLC+NLC, and PLC+NLC+LLC at different link correlations. When the average sender pairs in the network experience relatively high positive link correlation of 0.6, utilizing PLC, PLC+NLC, and PLC+NLC+LLC at the micro-level can reduce the overall transmissions by 4.5%, 6.9%, and 26.7%, respectively, as compared to CTP protocol.

On the other hand, when the average sender pairs in the network experience the high negative link correlation of -0.9, effectively utilizing the sender pairs that are negatively correlated in the network alone provides overall transmission saving of 7%. We have also noticed that when more sender pairs are experiencing positive link correlation utilizing LLC alone can provide significant benefit by reducing overall transmission as compared to CTP protocol.

Fig. 15 shows the percentage of transmission reduction from improving the sending order sequence at macro-level by either using our distributed heuristic or centralized heuristic under the network experiencing different levels of link correlations. We can see from Fig. 15 that more than 25% of transmission reduction can be achieved by scheduling the transmission order of the correlated senders. This shows that the sending order sequence is an essential step for increasing the utilization of the sender-based link correlation in the network.

Next, we vary the PRR of links in the network while link correlation between 0.8 and 0.9. Fig. 16 shows total number of transmissions reduces when PRR of links in the network increases. This is expected because CTP protocol forwards packets over the most efficient links. Fig. 16 reveals that when the average PRR is low, the benefit of utilizing link correlation increases. For example, when average PRR is 0.1, utilizing all three types of link correlations (PLC+NLC+LLC) can reduce total transmissions by 23% as compared to not utilizing link correlation.

Fig. 17 reveals that the percentage of transmission reduction is more than 25% for the centralized heuristic. The performance of distributed heuristic is close to the performance of centralized heuristic. This indicates that the local sending order sequence computed by distributed heuristic can still achieve the performance of global sending order sequence. Also, notice that the percentage of saved transmissions for both distributed and centralized heuristic slowly reduces from CTP protocol as average PRR increases. This behavior is expected because the number of correlated links in the network reduces as the average PRR of links in the network increases.

VIII. Conclusion

In this paper, we introduce a link correlation-aware data forwarding protocol by exploiting the sender-based link correlations. We first design a micro-level control scheme at individual nodes by either switching to a new forwarding node or suppressing the current packet transmission according to the transmission results of correlated neighbors.

Furthermore, on observing that all link correlations are conditional events, we further improve the network performance at the macro-level by scheduling the ordering of packet transmissions among neighbors so as to increase the utilization of link correlations in the network. The results from emulation show that our design improves transmission efficiency close to 12% more than the energy efficient ETX-based protocols (CTP) with 0.2% additional control message overhead.

The proposed link correlation-aware data forwarding protocol serves as a reference design for utilizing various link correlations and we expect such sender-based link correlations can also be applied to other network services such as network coding and cooperative forwarding.

Acknowledgments

This work was supported by iTrust IGDSi1305013, Singapore-MIT International Design Center IDG31000101, NSFC 61300174, CPSF 2013M530511, 2014T70026, SKLNST-2013-1-02, and in part by NSF grant CNS-1217791.

References

[1] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *NSDI'10*, 2010.

[2] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari, "The κ factor: inferring protocol performance using inter-link reception correlation," in *MobiCom '10*, 2010.

[3] M. Doddavenkatappa, M. Chan, and A. Ananda, "Indriya: A low-cost, 3d wireless sensor network testbed," *Testbeds and Research Infrastructure. Development of Networks and Communities*, 2012.

[4] 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.

[5] G. Shuo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *Proc. ICNP '11*, 2011, pp. 383–392.

[6] S. Alam, S. Sultana, Y. Hu, and S. Fahmy, "Link correlation and network coding in broadcast protocols for wireless sensor networks," in *SECON' 12*, 2012.

[7] S. Wang, S. M. Kim, Y. Liu, G. Tan, and T. He, "Corlayer: A transparent link correlation layer for energy efficient broadcast," in *Mobicom '13*, 2013.

[8] A. Basalamah, S. M. Kim, S. Guo, T. He, and Y. Tobe, "Link correlation aware opportunistic routing," in *INFOCOM' 12*, 2012.

[9] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *SenSys '08*, 2008.

[10] M. H. Alizai, O. Landsiedel, J. A. B. Link, S. Götz, and K. Wehrle, "Bursty traffic over bursty links," in *SenSys '09*, 2009.

[11] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse, "Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks," in *IPSN '10*, 2010.

[12] W. Liu, Y. Liu, Z. Zhong, L. Sun, H. Zhu, and T. He, "Exploiting ephemeral link correlation for mobile wireless networks," in *SenSys '12*, 2012.

[13] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proc. IPSN'11*, 2011, pp. 73 –84.

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *SenSys' 09*, 2009.

[15] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," in *SenSys '04*, 2004.

[16] S. Lee, B. Bhattacharjee, and S. Banerjee, "Efficient geographic routing in multihop wireless networks," in *MobiHoc '05*, 2005.

[17] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-Throughput Path Metric for Multi-Hop Wireless Routing," in *Proc. of the Intl. Conference on Mobile Computing and Networking (MOBICOM)*, September 2003.

[18] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sen. Netw.*, vol. 8, no. 4, pp. 34:1–34:33, 2012.

[19] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: a hybrid mac for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, Jun. 2008.

[20] J. Macke, P. Berens, A. Ecker, A. Tolias, and M. Bethge, "Generating spike trains with specified correlation coefficients," *Neural Computation*, vol. 21, no. 2, pp. 397–423, 2009.

Appendix

Suppose two links $s_1 \rightarrow r$ and $s_2 \rightarrow r$, $r \in F_{s_1} \cap F_{s_2}$, are not independent.

$$P(\overline{s_2}, r|\overline{s_1}) = \frac{P(\overline{s_2}, r) - P(\overline{s_2}, r|s_1)P(s_1, r)}{P(\overline{s_1}, r)} \tag{7}$$

$$= \frac{P(\overline{s_2}, r)}{P(\overline{s_1}, r)} - \frac{P(s_1, r|\overline{s_2})P(\overline{s_2}, r)}{P(\overline{s_1}, r)} \tag{8}$$

$$= \frac{P(\overline{s_2}, r)}{P(\overline{s_1}, r)} - \frac{(P(s_1, r) - P(s_1, r|s_2)P(s_2, r))}{P(\overline{s_1}, r)} \tag{9}$$

where we have used law of total probability at line (7) and line (9), formula for Bayes's theorem at line (8), and the fact that $P(\overline{s_1}, r) = 1 - P(s_1, r)$.