Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 826702, 17 pages
doi:10.1155/2012/826702

Research Article

Exploiting the Burstiness of Intermediate-Quality Wireless Links

Muhammad Hamad Alizai,[1] Olaf Landsiedel,[1, 2] and Klaus Wehrle[1]

[1] *Communication and Distributed Systems (ComSys), Informatik 4, RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany*
[2] *School of Electrical Engineering, KTH, Stockholm, Sweden*

Correspondence should be addressed to Muhammad Hamad Alizai, hamad.alizai@rwth-aachen.de

Received 8 September 2011; Revised 16 December 2011; Accepted 19 December 2011

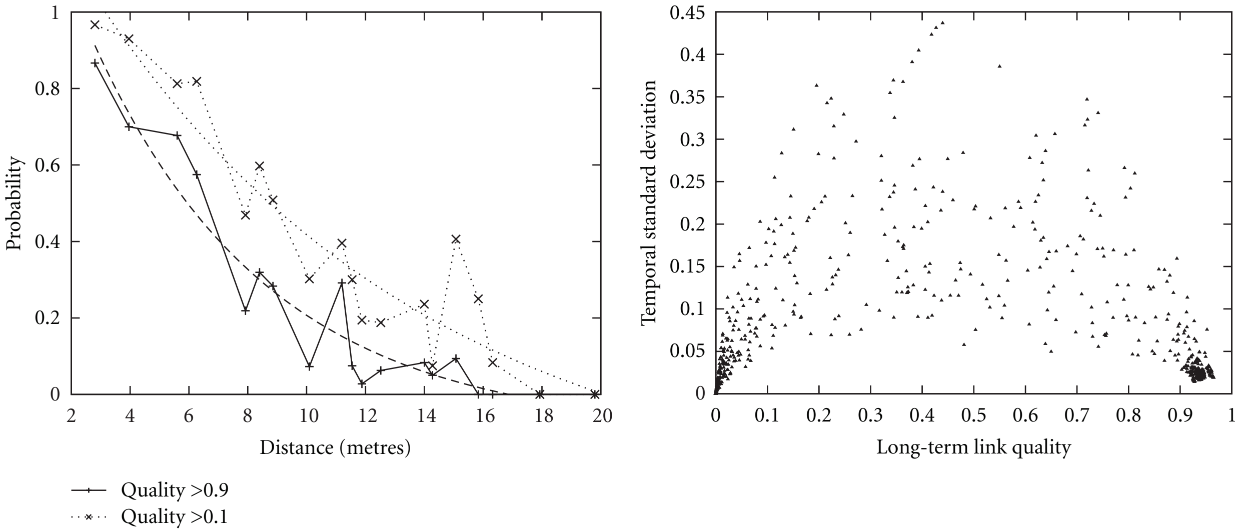
Academic Editor: George P. Efthymoglou

We address the challenge of link estimation and routing over highly dynamic links, thats is, bursty links that rapidly shift between reliable and unreliable periods of transmissions. Based on significant empirical evidence of over 100,000 transmissions over each link in 802.15.4 and 802.11 testbeds, we propose two metrics, *expected future transmissions* (EFT) and MAC_3, for runtime estimation of bursty wireless links. We introduce a bursty link estimator (BLE) that based on these two metrics, accurately estimates bursty links in the network rendering them available for data transmissions. Finally, we present bursty routing extensions (BRE): an adaptive routing strategy that uses BLE for forwarding packets over bursty links if they offer better routing progress than long-term stable links. Our evaluation, comprising experimental data from widely used IEEE 802.15.4-based testbeds, reveals an average of 19% and a maximum of 42% reduction in the number of transmissions when routing over long-range bursty links typically ignored by routing protocols. Additionally, we show that both BLE and BRE are not tied to any specific routing protocol and integrate seamlessly with existing routing protocols and link estimators.

1. Introduction

Instability of links and connectivity in low-power sensornets have so far been regarded as a difficult problem that existing routing algorithms try their utmost to avoid. Therefore, since the emergence of sensornets, research has mainly focused on link estimation and routing techniques [1–4] which identify and utilize consistently high-quality links for packet forwarding. Links of intermediate quality (we use the term intermediate quality to represent wireless links with a PRR between 10% and 90%) are ignored to ensure routing stability and to attain high end-to-end reliability. Protocol studies [5–8] have shown that these intermediate quality links are bursty, that is, they frequently switch between stable and unstable periods of transmission for a limited number of consecutive packets. In this paper, we argue that: (1) bursty links (we define an intermediate wireless link with a PRR between 10% and 90% as a *bursty link* if packet delivery on this link is correlated. This means that shifts between phases of reliable and poor packet delivery occur at short-time scales, but future packet delivery is correlated to the very recent success rate. Reliable links, that is, with a PRR >90%, can also be bursty but such links are not in the focus of this paper) can be used for packet forwarding during their stable periods without affecting the reliability and stability of existing routing protocols, (2) these links often achieve significantly better routing progress and routing throughput than the long-term stable links chosen by existing routing protocols. For example, Figure 1(a) shows that the probability of finding an intermediate link is higher than the probability of finding a high-quality link particularly at longer distances.

Despite establishing a very strong knowledge base regarding the causes of link burstiness over the past few years, we still lack metrics that define the quality and usability of bursty links. Similarly, we need a link estimator that can assess link usability online (i.e., during runtime) to enable the inclusion of these links in the routing process. However, the requirements and challenges of estimating intermediate links are substantially different from conventional link estimation. For example, Figure 1(b) shows that this class of links is subject to large and frequent temporal variations. Their dynamic connectivity poses a special challenge to any



(a) Probability of finding a high-quality or medium-to-high-quality neighbor depending on physical distance in our testbed

(b) Temporal variation of link quality. Each point represents a (directional) node pair

FIGURE 1: Low-power radio links in sensornets exhibit inevitable fluctuations in their quality.

link estimator. Therefore, long-term packet reception rates, otherwise the key link quality metric, of intermediate links do not suffice as a metric. Rather, (1) we are interested in whether or not packet delivery on an intermediate link is correlated to the recent delivery history, that is, if the link is bursty or not. (2) We want to know how long a bursty link remains reliable for transmission, that is, what is the length of successful transmission bursts. (3) We need to pinpoint exactly when a bursty link has a reliable or unreliable transmission period. None of these three pieces of information, which we consider keys to profitably using intermediate links for routing, are provided by existing link estimators. The definition of appropriate metrics, the design of a link estimator based on these metrics, and the development of an appropriate algorithm for routing over bursty links are our main contribution and departure from the existing work.

Because we are interested in online assessment of links, our definition of link burstiness is much lenient than how it is typically defined in the literature. For example, β [7] is a metric that primarily focuses on accurately measuring the burstiness of links based on their resemblance to an ideal bursty or independent link. However, as an input for its calculations, β requires traces of 100,000 packets over a link to achieve its desired level of accuracy in defining link burstiness. While this level of input size is neither desirable nor possible during runtime, we believe that such a strong metric is not necessary for the goals envisioned in this paper. We are only interested in predicting the fate of future transmissions over a link using minimal transmission history regardless if that link strongly resembles an ideal bursty link or not. For this purpose, based on significant empirical evidence of over 100,000 transmissions over each link in 802.15.4 and 802.11 testbeds, we propose two such metrics: *expected future transmissions* (EFT) and *MAC*$_3$, for runtime estimation of bursty wireless links. We define these metrics in Sections 3.2.4 and 3.2.3, respectively.

Overall, this paper (this paper is an extended version of ACM Sensys 2009 paper “Bursty Traffic over Bursty Links”) makes four key contributions.

Estimating Link Burstiness. We introduce MAC$_3$ as a lightweight metric to estimate the burstiness of links based on the recent delivery traces during runtime. MAC$_3$ extends the established *conditional packet delivery function* (CPDF) [9] by calculating a moving average over a sliding windows of CPDFs (*Moving Average CPDF*).

Estimating Burst Lengths. We define *expected future transmissions* (EFT) as a metric to estimate the duration for which a bursty link remains reliable for transmission. We also show that EFT is strongly correlated to MAC$_3$.

Bursty Link Estimator. Based on these two metrics, we introduce a bursty link estimator, derive requisite parameters for its usage and evaluate its efficacy in estimating intermediate links. Our results indicate that BLE identifies bursty links in the network with high accuracy, hence paving the way for such links to be included into the routing infrastructure.

Bursty Routing Extensions. To effectively utilize BLE, we present an adaptive routing algorithm that dynamically selects bursty links during the course of transmission. Our evaluation on widely used 802.15.14 testbeds indicates that BRE achieves an average of 19% and a maximum of 42% reduction in the number of transmissions when compared to a traditional collection protocol. Moreover, we show that both BLE and BRE are not tied to any specific routing protocol and integrate seamlessly with existing routing protocols and link estimators.

The remainder of this paper is structured as follows. Section 2 discusses related work. Section 3 introduces metrics and presents the complete design and evaluation of BLE. The design of BRE and the associated challenges is discussed in Section 4. We present our evaluation results in Section 3.4. Section 6 concludes our discussion and outlines future work.

2. Related Work

Capturing and utilizing link dynamics at different time scales and characterizing link burstiness have been the focus of many recent studies. We can divide prominent related efforts into three main categories.

2.1. Long-Term Link Estimation. Long-term link estimation is the traditional link estimation mechanism employed by the majority of current multihop wireless routing protocols [4, 10]. It is based on window mean exponential weighted moving averages (WMEWMA) of link PRRs or ETX [11]. Although WMEWMA is highly accurate and has a small settling time for good and bad links, that is, close to 0% and 100% PRR, it does not perform well for links of intermediate quality [3], as also indicated by our results in Section 3.3.1. Hence, this approach has two major pitfalls: First, it prohibits the use of neighbors with intermittent connectivity that might reach farther into the network. Second, in a sparse network with a low density of nodes, a node might have no high-quality neighbor in its communication range, requiring a mechanism to deal with unstable connectivity.

2.2. Link Burstiness. Srinivasan et al. [7] present a comprehensive study to quantify the extent and characteristics of bursty links. They define a factor β, which measures the burstiness of a wireless link. β is calculated by using conditional packet delivery functions (CPDFs), which determine the probability that the next packet will be received after n consecutive successes or failures. β is used to identify bursty links with long bursts of successes or failures and statistically independent links, with ideal bursty ($\beta = 1$) and independent ($\beta = 0$) links marking the two ends of spectrum. β is a very useful and elegant metric to measure link burstiness based on the existing traces rather than online assessment of a link.

Our evaluation in Section 3.2.3 also reveals that calculating β over short history sizes, a fundamental requirement of online assessment, results in fluctuating and error-prone results. Moreover, we present an entirely different concept of utilizing long-range intermediate links that are currently ignored by routing protocols.

2.3. Opportunistic Routing. Opportunistic routing [12] in 802.11-based wireless networks reports a throughput increase of 35% by utilizing long-range wireless links. It uses the broadcast primitive and an agreement protocol among the intermediate nodes that receive a batch of packets for prioritizing the intermediate node closest to the destination for forwarding packets. However, it has a relatively high overhead with regard to computational cost, storage, and communication which is not feasible in resource-constrained sensornets. Opportunistic routing operates on a batch of packets and strives to reach a delivery threshold of 90% before turning back to traditional routing for delivering the remaining 10% packets. We share the same spirit as opportunistic routing but differ significantly in detail. Our primary goal is to reduce the number of transmissions in the network. We apply unicast forwarding and hence the next forwarder of the packet is predetermined. Similarly, our approach rapidly falls back to traditional routing to avoid overshooting links with high-loss rates. Our aim is to utilize long-range bursty links to increase routing progress and throughput without introducing significant overhead in terms of computation, storage, and communication.

Overall, the metrics and algorithms presented in this paper are designed according to lessons learned from experimental studies on bursty wireless links, such as [5, 7]. However, our work does not aim at modeling and developing the analytical or experimental understanding of wireless links. Instead, we take a step further and use these experimental models for packet forwarding over bursty links, and hence, enabling better utilization of wireless links.

3. Link Estimation

Our major design goal is to reduce the number of transmissions in the network and increase routing throughput by utilizing long-range bursty links for packet forwarding. To achieve this goal, we need to provide relevant support at both link estimation and routing level. This section addresses the link estimation, while the corresponding routing extensions are discussed in Section 4.

3.1. Problem Analysis. To provide a clear motivation for our work as well as a separation from the previously mentioned related work, we now define the problem space and the requirements for a solution. First, we introduce the prevalent network scenario to illustrate use cases and benefits of our solution. Second, we detail the motivation for employing a bursty link estimator (BLE) in networks where established link estimation mechanisms are firmly in place. Based on this, we highlight the key requirements of a link estimator for incorporating bursty links in the routing process.

3.1.1. Network Scenario. Sensornets and meshnets provide flexible and robust ways of establishing network structures without the need for exhaustive infrastructure. Routing structures in these networks are self-established and maintained and depend on the presence of wireless links between nodes in the network. A resource-efficient utilization of these structures greatly increases throughput and network lifetime and reduces transmission latencies and failures. Our work targets sensornets and meshnets due to their equivalent routing mechanisms. However, our implementation and evaluation in this paper are focused on sensornets, a notoriously difficult class of multihop wireless networks.

3.1.2. The Need to Utilize Bursty Links. With regard to the characterization of links in [7], link estimation mechanisms typically utilize only good to perfect links with a PRR ≥90%. However, most links in wireless networks exhibit worse PRRs and are thus excluded from routing decisions. This results in (i) a stable and clear-cut routing topology, (ii) a very narrow set of routes in the network, and (iii) heavy utilization of these selected links. As the majority of links is not made available for routing, conservative routing schemes cannot benefit from the multitude of routes in the network which may offer better routing progress than conservatively chosen routes. In addition, unused links may relieve main links and thus improve the overall network performance.

3.1.3. Requirements of a Bursty Link Estimator. The design of a link estimator that reliably reflects the state of a given link has to fulfill multiple requirements. First, appropriate metrics need to be derived as key building blocks of the estimator. With regard to intermediate links, these metrics shall be able to (i) estimate the burstiness of a link and (ii) shed light on how long are the usual transmission bursts on an intermediate link. Moreover, such metrics must timely estimate the current link quality based on a very short transmission history in order to adapt to the rapidly changing conditions of bursty links. Additionally, the predicted link quality must accurately and reliably reflect the actual link quality, that is, the estimation error needs to be small and stable. Second, building upon such metrics, a link estimator must efficiently utilize the given information to select beneficial links for routing. This requires appropriate neighbor table management policies that select those links for routing, among all available links, which allow for the best routing progress.

3.2. Deriving Metrics for Bursty Links. Based on the properties specified in the previous section, this section defines and evaluates two metrics, MAC_3 and EFT that (i) identify bursty links in the network and (ii) estimate the length of successful transmission bursts. These metrics subsequently lay the foundation for BLE. In the following, we first provide detailed information on the particular data set used in the remainder of this study.

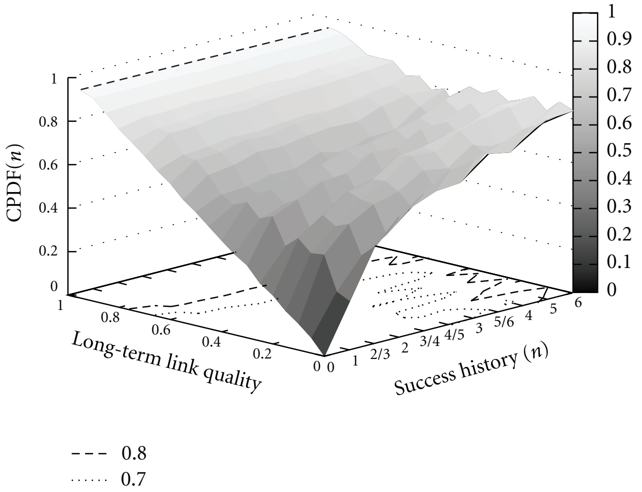
3.2.1. Data Set. The evaluation results presented in this section are based on the SING mesh data-set [13] compiled at Stanford University and used in many recent state-of-the-art studies [7, 14, 15] on wireless link dynamics. It is a comprehensive data set collected from multiple IEEE 802.11 and IEEE 802.15.4 testbeds (Motelab, Mirage, and SWAN testbeds. Please visit http://sing.stanford.edu/srikank/datasets.html and the websites of each testbed for further information) including both packet and byte level radios such as the cc2420 and cc1000. Specifically, the data comprises traces of transmissions on channel 26 at a transmission power level of 15 dBm. Each node broadcasts a burst of 100,000 packets with a packet inter arrival time of 10 ms. Out of all available links, we only include intermediate links in our evaluation and comparison. This is because good links would measure highly in our metrics and would thus improve our results. However, these links are not the focus of our work.

3.2.2. Case Study: Predicting Transmission Success from a Short History. Before introducing our metrics MAC_3 and EFT, we motivate our goals and approach with a case study. We address the question, whether a short history of successful transmissions is sufficient to predict with a high probability that the next transmission on this link will be successful, too. Figure 2(a) depicts the conditional probability of a successful packet transmission based on the average long-term link quality (i.e., PRR) and a short-term history of consecutively *successfully* transmitted packets. It shows that for a link with long-term quality greater than 60%, even a single or two successful transmission over that link raise the probability of next successful transmission to 90%. Similarly, it shows that any link regardless of its long-term link quality, if the last three packets over that link were successful, the probability of a future successful transmission is greater than 90%. Figure 2(b) depicts the probability of a successful packet transmission based on the average long-term link quality and a short-term history of consecutively *failed* packet transmissions. It indicates that after one or two consecutive losses, any link should be temporarily considered unreliable.

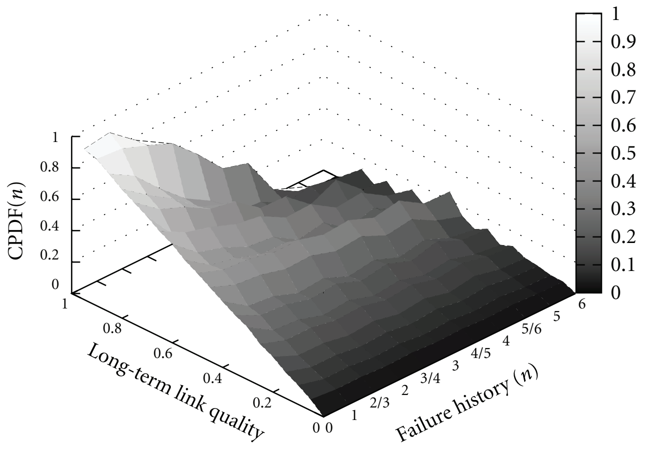
Overall, these results indicate that a short-term history of three packets over a link is sufficient to determine with a high probability whether the next transmission will be successful or not. In the following, we introduce two metrics MAC_3 and EFT that determine the success probability of future transmissions on a per link granularity, allowing us to reflect spatial properties of link dynamics.

3.2.3. Online Estimation of Link Burstiness. Estimating the burstiness of a link is mandatory to determine whether or not an intermediate link is beneficial to the overall routing performance. The key challenge is to clearly distinguish between intermediate links with correlated packet losses from those with independent losses. However, unlike offline measurement mechanisms like β, we are not interested in how close a link is to an ideal bursty link with one long burst of either successes or failures. Our goal to predict link burstiness at runtime strongly influences the definition of burtiness and the timescale of our prediction. In this context, we define a link as bursty as long as we can *recurrently* predict the fate of only the next transmission over a link with high probability. This is why we introduce a new metric that monitors a link for a limited transmission history and expresses if the occurrence of a successful transmission burst over a particular link is a mere coincidence or if it is a reoccurring trend. This information is important to determine if a link is beneficial for routing purposes.

Our online metric Moving Average CPDF (MAC_3) is based on a CPDF(n) (Conditional Packet Delivery Function) [9] which calculates the probability of one successful transmission following n previously successful transmissions.



(a) Influence of recent transmission success rate on the success probability of next transmission. A label of k/n stands for k successes during the last n transmissions, and n is a shorthand for n/n. CPDF(n) is the probability of the next transmission being successful



(b) Influence of recent transmission failure rate on the success probability of next transmission. A label of k/n stands for k failures during the last n transmissions, and n is a shorthand for n/n. CPDF(n) is the probability of the next transmission being successful

Figure 2: Influence of success and failure of recent transmissions events on short-term link quality.

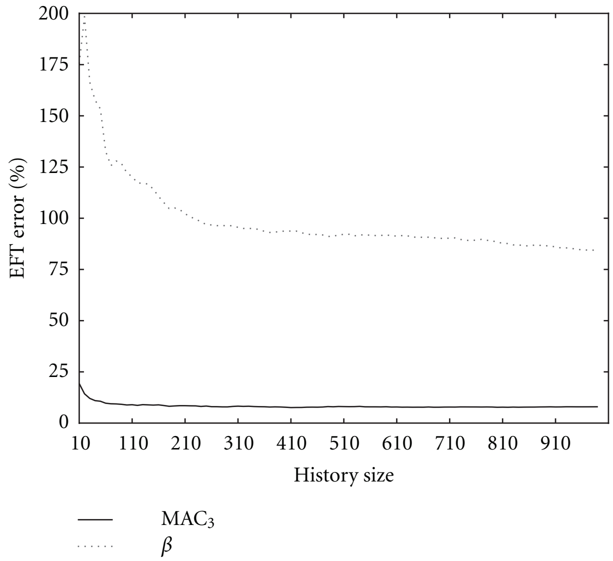
Based on the results in the initial case study in Section 3.2.2, we compute an average CPDF(3) over the recent history h of length m of a link and denote it AC_3 (see Definition 1). |CPDF(3)| defines the number of valid CPDF(3) in the history, that is, the instances of three consecutive transmissions. We define MAC_3 as the moving average of AC_3 that is computed by adding new values and removing old ones from the history at runtime.

$$AC_3 = \frac{\sum_{i=1}^{m} \mathrm{CPDF}_{h_i,h_{i+1},h_{i+2}}(3)}{|\mathrm{CPDF}(3)|}. \quad (1)$$

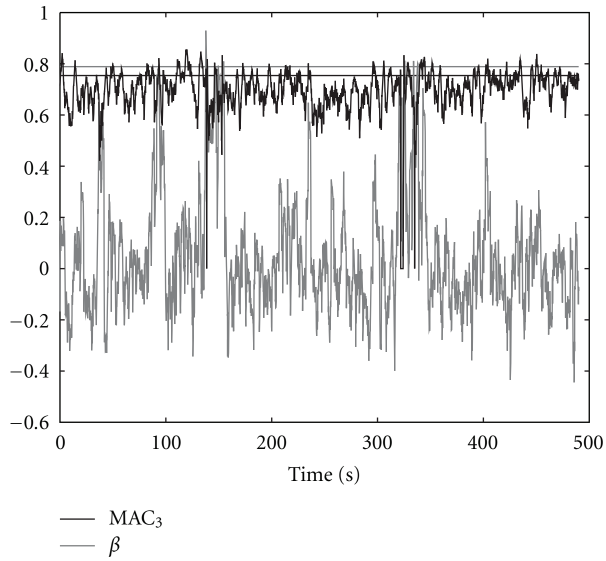
To evaluate MAC_3 we compare it with the β factor [7] because, (i) it is the only metric available that measures link burstiness and (ii) to enable better understanding and the effectiveness of MAC_3 as a runtime metric. However, this comparison, by any means, does not attempt to undermine the usefulness of β as β was never developed for runtime measurements.

Figure 3(a) illustrates the estimation error of MAC_3 and β over history sizes ranging up to 1000 packets. The estimation error is the difference between the estimated value of either metric when applied to a certain history size (plotted on the x-axis) and the value when applied to the whole transmission trace (i.e., base value). The base value of β is calculated according to the procedure prescribed in [7]: A CPDF(n) for a certain n is only considered in β calculations if it has at least 100 data points to achieve a 95% confidence interval of $[P - 0.1, P + 0.1]$. Whereas for calculating β over a shorter transmission history, we do not enforce the condition of 100 data points. This is because, (i) it is simply not possible to get 100 data points in a shorter transmission trace and (ii) we want to see if this smaller version of β provides accurate estimates and can be used for runtime estimation of link burstiness. The figure indicates that our online metric MAC_3 rapidly converges to a minimal error of 7% with a history size of less than 100 packets. In contrast, β shows a significantly slower initial convergence phase and is not able to achieve an error smaller than 83% even with a history size of 1000 packets. Moreover, β is not able to provide stable results for small history sizes as shown in Figure 3(b). Given a concrete history size, β generates severe fluctuations in its output over time when applied to an entire transmission trace of a particular link. The estimated values of MAC_3 on the other hand expose considerably smaller differences. In addition, the results of β again strongly deviate from the base value calculated over the whole trace (straight gray line) while the estimates of MAC_3 oscillate around its actual base value (straight black line). Overall, these results show the efficiency of MAC_3 as online metric: it is stable for short history sizes.

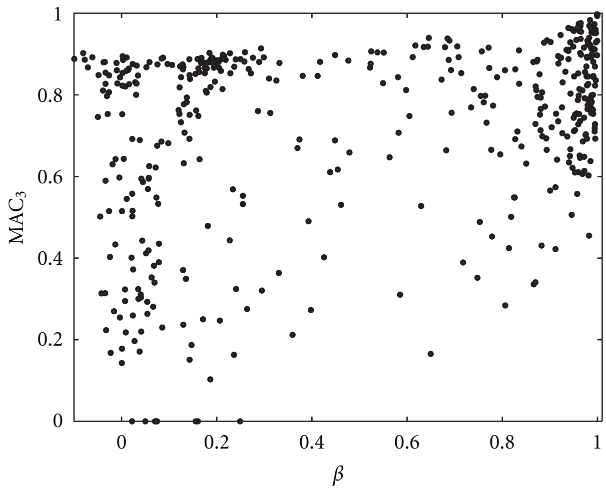
Next we show that MAC_3, in contrast to β, captures the short-term properties of a link. Figure 3(c) shows that many links with MAC_3 >80% have low β values. It means that on such links, the probability of a successful transmission after three consecutive deliveries is greater than 80%, but the use of β as a link metric will not let a routing protocol select this link. After evaluating the effectiveness of MAC_3, we need to analyze what proportion of the available intermediate links are actually useful for routing. Figure 3(d) shows the cumulative distribution function of MAC_3 and β for all intermediate links in the Mirage testbed. We can clearly observe that the majority of these links have a very high MAC_3. As a result, MAC_3 unlocks the formerly wasted potential of those links and enriches the routing process with a multitude of new routing opportunities.



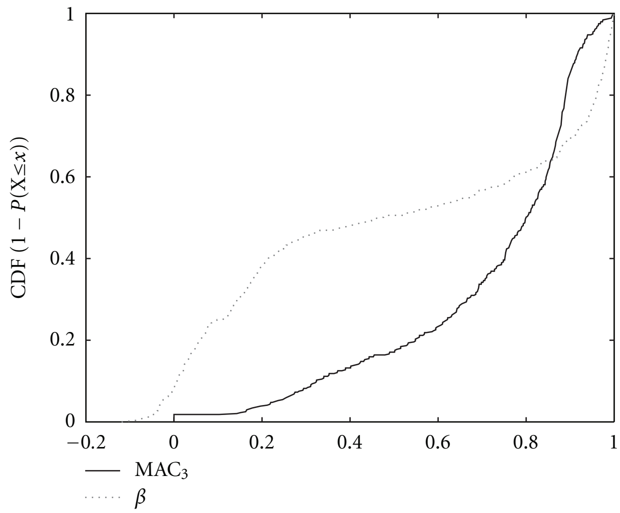
(a) Settling Time: MAC$_3$ shows a faster convergence towards its base value over the history size and achieves a smaller estimation error (7%) than β. Based on data of the Mirage testbed



(b) MAC$_3$ generates more accurate and more stable results over time than β. The straight lines show the base values of both metrics over the entire transmission trace



(c) MAC$_3$ reveals that many links have a high probability for a further successful transmission after three consecutive deliveries even though their β is very low



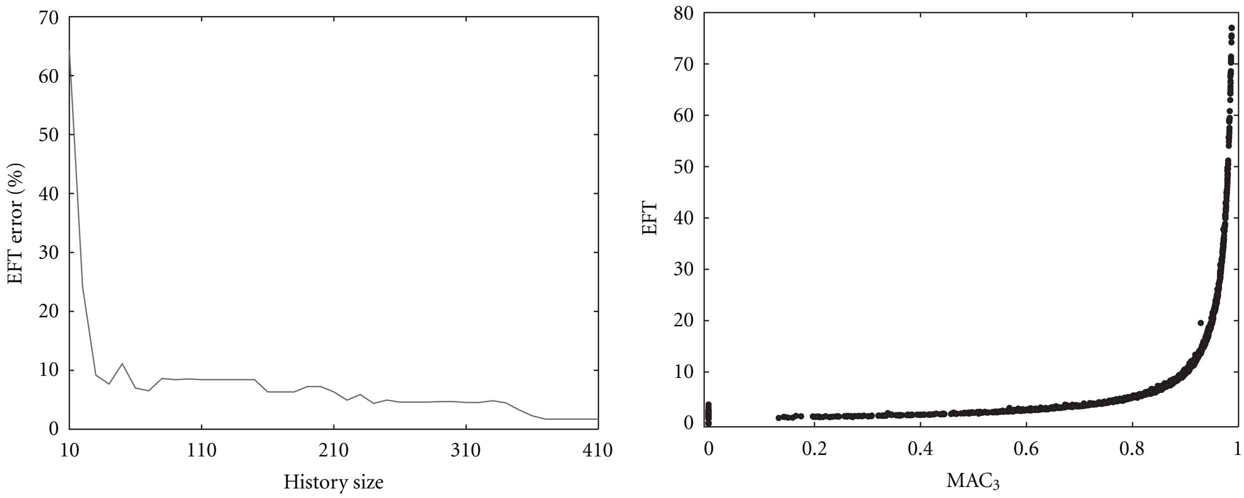
(d) Cumulative distribution of intermediate links: the majority of intermediate links is bursty (MAC$_3$ >0.7), offering useful transmission opportunities

Figure 3: Comparing MAC$_3$ and β as a link burstiness metric for runtime link estimation. We use a smaller version of β for online link assessment. Our version of β does not enforce a confidence interval of 95% for its data points.

Concluding, MAC$_3$ is a lightweight metric for estimating link burstiness during runtime. Our results in Section 5 demonstrate that, when used as metric to estimate link burstiness, MAC$_3$ accurately identifies bursty links in the network.

3.2.4. Estimating Burst Lengths. In addition to identifying whether or not a link is bursty, a second metric for estimating the length of bursts is required. To illustrate why, assume a bursty link with a steady rate of bursts covering four successful transmissions each before becoming unreliable again. Such a link exhibits a very high CPDF(3) value, causing MAC$_3$ to correctly identify it as bursty. However, if selected for transmission, this links allow for only one more successful transmission per burst, hence rendering it barely suitable for routing.

To solve this problem, we introduce a new metric named EFT(n) it estimates the number of r successful future transmissions after n successful packet deliveries. This metric thus predicts the length of bursts and allows the link estimator to identify bursts of relevant size. Just like MAC$_3$, EFT uses an averaging moving window to traverse a transmission history. For each occurrence of three successful consecutive transmissions in the history, EFT determines



(a) Influence of the history size on convergence of EFT

(b) Correlation between MAC_3 and EFT

FIGURE 4: EFT and MAC_3 as link quality metrics.

the number of subsequently following successful transmissions and incorporates it in the total average.

Similar to MAC_3, EFT has a very small settling time (cf. Figure 4(a)): it converges to within 10% error at a history size of approximately 100 packets. Figure 4(b) indicates a strong correlation between EFT and MAC_3. For values of MAC_3 in the range of 0.1 to 0.7, EFT predicts burst lengths not longer than five packets. However, when MAC_3 exceeds 0.7, the estimated burst lengths increase significantly. As a result, we derive a threshold of 0.7 of MAC_3 for a link to be considered for future transmissions.

3.3. The Bursty Link Estimator. We propose a packet snooping-based link estimation [3, 10] for BLE. BLE is not supposed to work independently: it is an additional component of the routing infrastructure that enables a fine grained estimation of intermediate links and allows for such links to be included in the routing process. In this section, we first discuss why PRR is not a suitable metric for intermediate links and propose a combination of MAC_3 and EFT to be used as link quality metrics for BLE. We then provide further details about the information maintained in BLE's table. Finally, we conclude this section by evaluating BLE.

3.3.1. Link Quality Metric. PRR (or ETX: the reciprocal of PRR) is commonly used as a link metric in current link estimators. The basic technique is to calculate weighted moving averages of PRR over a very long time period. Similar to β, PRR does not fulfill the desired properties of a metric for our envisioned link estimator. For example, it is unable to capture short-term dynamics exposed by bursty links of intermediate quality. Figure 5(a) highlights this fact: many links with a very high MAC_3 have very low PRRs. This means that over a long time scale, these links have bad reception rates. However, when observing a history size as small as 3, it is possible to predict the success of future transmissions with high probability. Similarly, Figure 5(b) supports this argument by comparing PRR and MAC_3 over time. It shows that though MAC_3 indicates a stable and high probability of successful delivery, PRR is unable to capture this reliable transmission period of a link. Hence, the use of PRR will prohibit the use of bursty links that offer useful transmission opportunities at shorter time scales.

In BLE, we use a hybrid metric that combines the information provided by MAC_3 and EFT. MAC_3 and EFT are calculated by applying a sliding window over the packet delivery history of size h for each link in the table. Since maintaining link history is an expensive memory operation and impacts the scalability, it is important to choose the threshold h appropriately as discussed in Section 3.4.1.

3.3.2. Table Management. BLE follows the basic table management algorithm outlined by Woo et al. [3] and used by the majority of current link estimators [2, 10]. We deviate from the established concept in terms of (1) link selection as BLE only estimates intermediate links (i.e., PRR <90%), specifically, the ones not present in the table of the associated long-term link estimator and (2) different ingredients for the link insertion, eviction, and reinforcement policies. The estimator maintains a small table (e.g., of size 10) of candidate links which holds the following information per link:

MAC_{3in}: the reception MAC_3 of the link (1 byte).

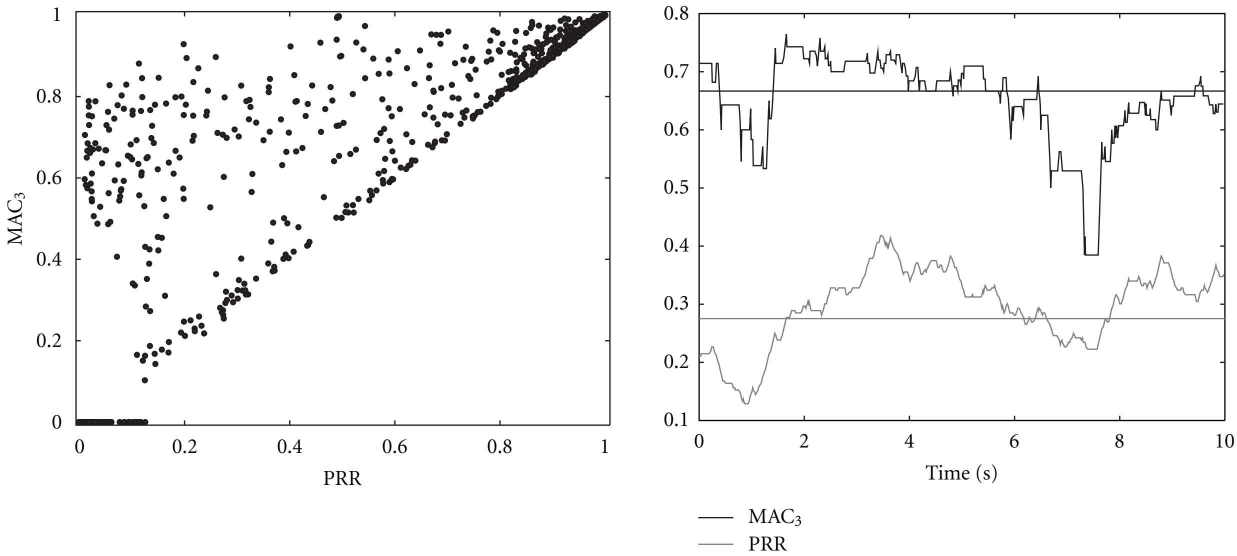
EFT_{in}: the reception EFT of the link (1 byte).

MAC_{3out}: the sending MAC_3 of the link (1 byte).

EFT_{out}: the sending EFT of the link (1 byte).

Link History: the packet delivery history of size h (16 bytes, see Section 3.4.1). Bit arrays are used with 1 representing a successful packet delivery and 0 representing a failed transmission.

Available: a flag to determine if the link, with MAC_3 and EFT above certain threshold, is currently available for transmission. Set to 1 if the last three transmissions were successful over the link, and 0 otherwise.



(a) Many links with low PRR exhibit a high success probability after 3 successful deliveries. All the links in Mirage testbed are plotted in this graph

(b) A 28% link shows a high MAC_3 and stable progress over time. PRR will never include such a link in neighbor tables

FIGURE 5: Comparison of MAC_3 with PRR.

Valid: a flag to determine if the link has a large enough delivery history and whether all other table entries are up-to-date.

The table management is concerned with three tasks: adding links, deleting links, and maintaining links in the table. A new link is added to the table upon reception of a packet on a nonresident link and (i) a vacant entry in the table exists, (ii) the product of MAC_3 and EFT of a resident link drops below a user-specified threshold, or (iii) an entry expired due to a broken link or an insufficient packet reception rate. Additionally, link maintenance is performed after i received packets. At this point, all entries in the table are recalculated. The value i is a tradeoff between the computational overhead and actuality of BLE.

3.4. Evaluation. We have implemented a prototype of BLE in TinyOS for a notoriously difficult class of wireless mesh networks sensornets. Our evaluation of BLE focuses on two factors. (i) *Link History Size*: we empirically derive a requisite history size h that shall be maintained by BLE to compute its link metrics. (ii) *Link Estimation*: we validate that BLE indeed includes bursty links of high quality in the neighbor table. The latter constitutes the key factor in assessing the performance of any link estimation mechanism, as the quality of the link selection process has a significant impact on the overall routing efficiency. For our evaluation, we only use intermediate links from the network.

3.4.1. Link History Size. Although determining link history is a user-desired accuracy threshold, we derive their values here for completeness and for evaluation purposes. Our goal is to find a requisite history size that balances estimation error and memory consumption. A too small history does not provide enough information to enable BLE to accurately predict the link quality. Conversely, a too large history blocks valuable system resources and potentially does not even improve prediction accuracy. We assume that an estimation error of 10% yields user-acceptable results. Figure 3(a) and Figure 4(a) show our results derived from the Mirage testbed data. We clearly observe that MAC_3 and EFT converge below a 10% error at a history size of approximately 100 packets. Hence, for our evaluation, we use 16 bytes of memory to store a single link history which corresponds to a (large-enough) history size of 128 packets.

3.4.2. Link Estimation. This evaluation aims to confirm that BLE correctly identifies bursty links in the network to provide these links with a high value of MAC_3 for inclusion in the routing process. Figure 6 illustrates the total number of links with a certain estimated quality and the fraction of links that were included in the neighbor table by BLE after 1000 transmissions over each link in the network. We observe that the fraction of selected links increases in conjunction with the estimated link quality. The fact that not all links with a high value of MAC_3 are included for routing stems from the criteria of link addition (see Section 3.3.2) and the requirements of a fixed and small table size. There may be more links than can be added in the table. Although Figure 6 presents an instantaneous snapshot of the BLE tables in the network, we observed a similar trend throughout our evaluation.

4. Bursty Routing Extensions

After discussing the design and evaluation of BLE, we now detail our corresponding bursty routing extensions (BRE),

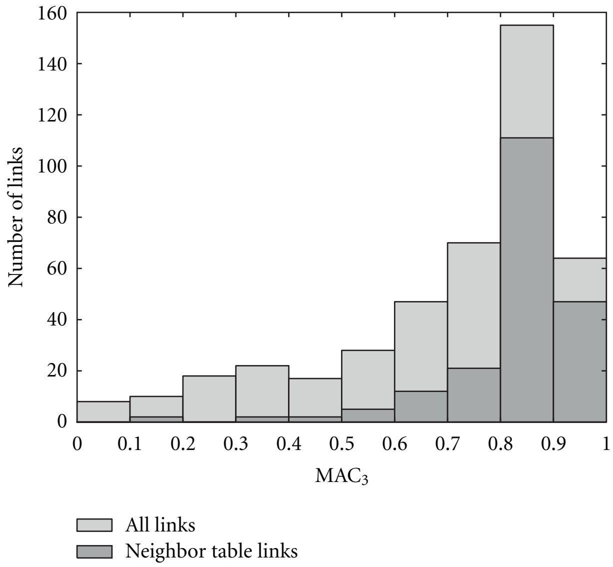


Figure 6: Number of links with a given estimated quality (light gray): subset of these links that are included in the neighbor table after 1000 transmissions by each node in the network (dark gray).

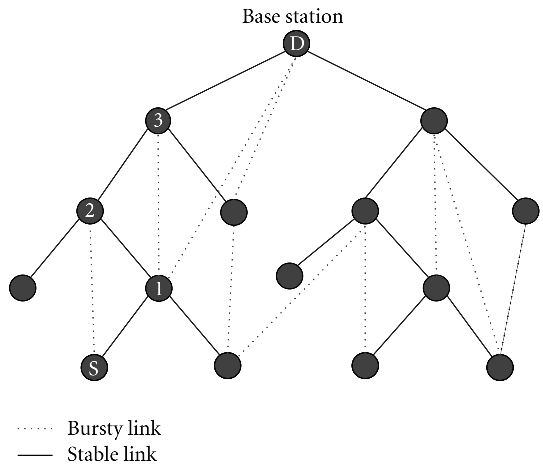


Figure 7: Bursty links provide routing shortcuts that can significantly reduce the hop count and the number of transmissions from source to destination.

a very cautious approach of forwarding packets over intermediate links. We first present the basic concept of our routing approach and then present the complete algorithm and its integration with existing routing protocols and link estimators.

4.1. Basic Concept. Typically, routing protocols in WSNs aim to establish a routing tree: some number of nodes in the network would advertise themselves as base stations, that is, as tree roots. All other nodes join the tree with ETX as the routing metric. Figure 7 shows an example of such a routing tree rooted at the base station D. A path from source S to the destination D consists of a subsequence of immediate parents of each node, for example S → 1 → 2 → 3 → D. The minimum number of transmissions required by a packet to travel from the source to the destination is four. Now consider a situation in which an intermediate link S → 2 or 1 → D has become temporarily reliable. Routing over these links could result in a path sequence S → 2 → 3 → D or S → 1 → D, respectively. Hence, using these links for routing could reduce the total number of transmissions to three in the former and two in the latter case. However, a traditional routing protocol does not make use of such an opportunity because it uses a long-term link estimate. Hence, this design is intentionally unable to realize short-term changes in the link quality. Similarly, even if these short-term changes are captured, traditional routing schemes adapt slowly to ensure routing stability.

In contrast, with a link estimation mechanism in place to estimate rapidly changing links in the short-term, our proposed technique takes advantage of the availability of such links. In this particular case for example, node 2 overhears the packets addressed to node 1 by source S. After 2 successfully overheard a certain number of consecutive packets from S, it informs S about the short-term availability of this link. Thereafter, S inquires the burstiness of this link and starts forwarding its packets to 2 to reduce the number of overall transmissions for a packet to reach its ultimate destination.

The packet overhearing technique employed in BLE benefits from the fact that WSNs typically reveal bursty traffic patterns. Common applications [16–20] operate as monitoring environment to detect and often track events. Typically, their occurrence results in long bursts of packets. Hence, they represent a major fraction of the overall network traffic although they occur rarely. In such situations, BLE, after overhearing the first few packets over a bursty link, identifies it as short-term available for transmission.

4.1.1. Algorithm. We define three roles for nodes in the network: (a) *sender node*: the node which is actively sending or forwarding packets (b) *parent*: the parent of any sender node in traditional routing and (c) *overhearing-node*: node(s) which can overhear the communication between the sender node and its parent. A node in the network can assume any or all of these three roles at a time. The BRE algorithm has the following four phases.

Link Discovery. When an overhearing node declares a link reliable for transmission, successfully overhearing three consecutive packets sent by a sender node to its parent, compared for. Section 3.2.2: it queries its routing table for the path-ETX of the packet's destination, that is, the parent of the sender node. If the path-ETX of the parent node is greater than that of the overhearing node, the overhearing-node declares the link between itself and the sender node active. Consequently, the active bursty link can offer a better routing progress than the traditional path used by the sender node. However, if the path-ETX of the parent node is not known or less than the path-ETX of overhearing node, the overhearing node temporarily ignores the sender node. In our example in Figure 7, it would be node 2 overhearing the communication between node S and its parent 1.

Link Announcement. If the path-ETX of the parent node is greater than that of the overhearing node, the overhearing node informs the sender node about the active bursty link (cf. Figure 7, and 2 informs S about the active bursty link between them). It volunteers itself to become the temporary parent of the sender node as long as this bursty links remains active. The path-ETX information used by BLE at the overhearing node can easily be obtained by using the neighborhood information maintained by any traditional routing protocol.

We assume that there is a high probability that the original parent of the sender node is also a neighbor of the overhearing node. This is because the overhearing node can listen to the ongoing communication between the sender node and its parent. An alternative approach to remove this neighbor-table dependency is to include path-ETX of the parent in each packet. However, this approach introduces 1 byte overhead in each data packet.

Additionally, the link announcement message, sent by the overhearing-node to the sender node, establishes a simple check to test for link asymmetry.

Routing Mode. The sender node, after receiving the announcement from the overhearing node, queries the BLE about the burstiness of this link. Consequently, if the burstiness of link exceeds certain threshold, the sender node makes the overhearing node its temporary parent and starts forwarding packets to it (cf. Figure 7, S forwards its packets to 2). However, this information is not propagated by the routing protocol to its descendant nodes, because these short term changes would trigger further parent changes down the tree. Eventually, it might destabilize the routing protocol and result in loops. This is one of the primary reasons why stability prevails over adaptability in today's routing protocols and link estimators. Hence, our routing strategy supplements their design considerations.

The main disadvantage of this approach is that BRE operates greedily. Although this approach is still effective for enhancing routing progress when compared to traditional routing, it does not promise the use of the optimal path currently available in the network. For example, a node may change its parent based on the recommendation of BRE. However, it is possible that along the traditional path, the sender node remains unaware of the availability of an even better bursty link currently reliable for transmission. Nonetheless, we believe that our approach strikes an efficient tradeoff between routing stability and performance adaptability.

Link Unavailability. The sender node declares a link unavailable for transmission after it fails to receive a number of acknowledgments (see Section 3.2.2) for the data packets sent over the bursty link. The sender node will then regress to traditional routing until it receives another link announcement.

4.2. Integration with Routing Protocols. Our goal is to enhance routing performance without affecting the stability and reliability of traditional routing protocols. Therefore, we neither replace the existing link estimators nor alter the stable routing topology maintained by traditional routing protocols. Rather, our approach adds an additional component to the system architecture that assists routing protocols and link estimators in identifying the previously ignored class of bursty links which can enhance routing performance. BRE seamlessly integrates with existing routing protocols and links estimators. Relying on BLE's to discover bursty links. (i) BRE does not introduce new routing tables and (ii) it does not require any modifications to the packet headers.

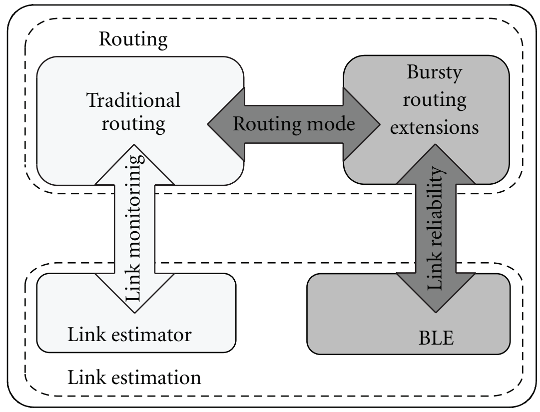


Figure 8: Design of bursty routing extensions.

We define two routing modes, a *bursty mod,* and a *traditional mode*. In *bursty mode,* packets are forwarded over the active bursty links identified by BLE. Conversely, in *traditional mode,* packets are forwarded along the path chosen by the regular routing algorithm.

We need only two interfaces for the integration of BLE and BRE into traditional routing protocols. The first interface is between BLE and the BRE. Using this interface, BLE informs the BRE about the availability of a potentially beneficial bursty link. The second interface is between the BRE and routing protocols. This interface is used to switch between different routing modes and to access the neighbor table maintained by the routing protocols to inquire the path-ETX of neighboring nodes. Figure 8 shows the major design components of BRE and their integration with traditional routing protocols.

5. Evaluation

In this section, we evaluate the performance of BRE when compared to the collection tree protocol (CTP) [4], a standard collection protocol for sensornets shipped with TinyOS. After describing the implementation details and experimental setup, we evaluate the impact of BRE on the routing cost and throughput. We give a detailed account of the timing properties of the bursty links used by BRE to enhance routing performance. We conclude our discussion by giving a detailed account of the overhead introduced by BLE.

Our data analysis mainly focuses on routing issues such as transmission costs, delivery reliability, and throughput.

Experimental studies, such as [1, 7], give further insight into the properties of intermediate and bursty links.

5.1. Implementation. We implemented BRE in nesC [21] for TinyOS 2.x. The prototype implementation of BRE is integrated with the collection tree protocol (CTP) [4]. CTP uses the four bit link estimator (4BLE) as its link estimation component. Although CTP is explicitly designed for relatively low data rates, we observed that it is capable of handling high traffic rates as well (i.e., it can deliver a packet every 25 to 30 milliseconds in a multihop network). Moreover, CTP has a very robust retransmission mechanism that ensures high delivery reliability. This property of CTP allows us to thoroughly evaluate the impact of BRE on the reliability of traditional routing protocols. However, BLE and its routing strategy are not bound to any specific routing protocol. It can easily be integrated with BVR [1] or other routing strategies that support higher data rates for bandwidth limited systems. Such strategies could, for example, merge multiple data frames into a single link layer packet.

5.2. Experimental Setup. The majority of our experiments were executed on MoteLab, a widely used sensor testbed at Harvard University. MoteLab is an indoor deployment of 190 TMoteSky [22] sensor motes on three different floors. However, due to the difficulty of maintaining such a large test-bed, only 142 motes were available to us at maximum. All our experiments had the following common characteristics unless stated otherwise: (1) motes transmit at full transmission power that is, OdBm, (2) we use an interpacket interval of 250 milliseconds (results are presented for different interpacket intervals as well), (3) we use the default $\alpha = 9$ for WMEWMA-based estimation [3] in 4BLE and 802.15.4 channel 26. (4) Each experimental run lasted for 30 minutes.

To ensure the validity of our MoteLab results, we reran our experiments on TWIST, a 100 node TMoteSky-based testbed at TU Berlin. TWIST is a high-density grid-like deployment with an intermote spacing of 3 meters. Therefore, to create a reasonably large multihop network, we reduced the transmission power to −15 dBm for our experiments on TWIST. The other characteristics are identical to our experiments on MoteLab.

5.3. Performance. In this section, we thoroughly evaluate the performance of BRE in terms of transmission cost, throughput, and reliability. Our major performance benchmark is to reduce the number of transmissions in the network by enhancing routing progress. Figure 9 shows a schema of the MoteLab topology (this is an abstract representation of the MoteLab topology. A detailed topology and connectivity graphs can be found at http://motelab.eecs.harvard.edu/) and highlights the motes that were used as senders and receivers in all our experiments. We define four different experimental classes, namely horizontal, vertical, diagonal, and nearby, to comprehend different network sizes and topological and physical scenarios (see Table 2). Our mote selection as a source and destination is also based on the these experimental classes.

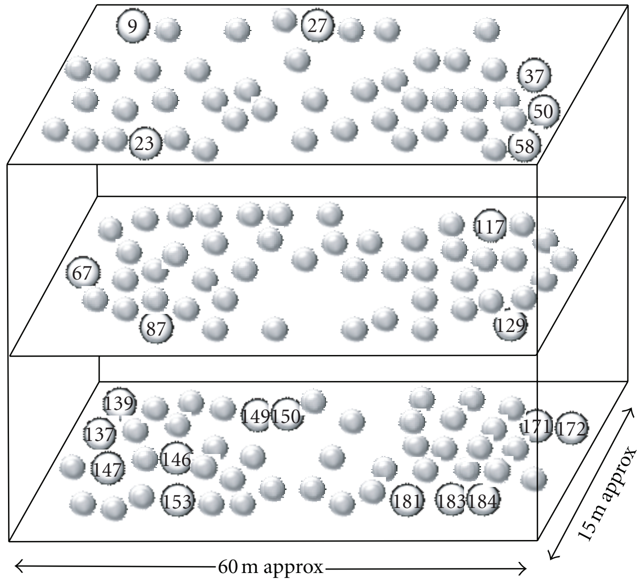


Figure 9: An abstract representation of the MoteLab topology on three different floors. The figure does not show the walls between rooms. The node IDs are only shown for the nodes that were used either as senders or collection roots during our experiments.

Before presenting our performance evaluation results, we demonstrate important topology characteristics that describe our analysis and allow for a deep understanding of the results that follow. These parameters are presented in Table 1 and their descriptions are presented in Table 2.

The high percentage of *Forwarders* and *Candidates* in Table 1 shows that a large number of nodes in the network can be utilized in our bursty forwarding approach. Table 1 testifies to the fact that more than 60% (i.e., 11.2 potential neighbors out of 15 neighbors in class *horizontal*) of a node's immediate neighbors had a better path-ETX than the original parent. Correspondingly, out of these *potential neighbors*, more than 70% (i.e., 8.6 candidate neighbors out of 11.2 potential neighbors in class *horizontal*) could even overhear three consecutive data packets. It means that these neighbors were not selected as a parent only because of a poor long-term quality estimate of their links with the sender. Algorithms that assess links based on average PRR, like most current approaches, do not use such a link, not even while it is in its good state. The high average of the measured packet loss rate based on broadcast beacons prevents the recognition of good transmission periods in such links.

Another observation is that, with the decrease in the number of intermediate links in the network, the number of *potential neighbors* and *candidate neighbors* also decreases (see Table 1 for class *nearby*). Although the *node density* of experimental class *nearby* is higher than class *horizontal*, the class *nearby* has a smaller number of *candidate neighbors*. It means that CTP (based on the link estimates of 4BLE estimator) indeed selected the best neighbor as a parent from the neighbors with high-quality links. This information

Table 1: MoteLab statistics for experimental parameters defined in Table 2. The statistics for *intermediate links*, *node density*, *potential neighbors*, and *candidate neighbors* were collected by randomly selecting 10 motes from different locations (i.e., corner, center) in the testbed. The statistics for *forwarders* and *candidates* were collected by running BRE on all the motes (sending a packet every 5 seconds) with a collection root (i.e., mote 183), located at one corner of the network.

Experimental class	Intermediate links %	Forwarders %	Candidates %	Node density	Potential neighbors	Candidate neighbors
Horizontal	33.3	94.8	90.2	15.0	11.2	8.6
Vertical and Diagonal	36.5	93.4	88.4	23.2	14.8	8.5
Nearby	14.2	86.2	79.3	16.3	9.6	4.0

Table 2: Description of experimental classes and parameters presented in Table 1.

Name	Description
Horizontal	Source and destination at the opposite ends on the same floor. Only the motes on the same floor were used for this class of experiments (e.g., node-pair 9 → 50).
Diagonal	Source and destination on different floors and on the opposite ends. All the motes in MoteLab were used (e.g., 137 → 50).
Vertical	Source and destination on different floors but on the same end. All the motes in MoteLab were used (e.g., 183 → 50).
Nearby	Source and destination are nearby to each other but surrounded by a high density of nodes. Only 30 to 50 neighboring motes were used (e.g., 153 → 183).
Intermediate links	The percentage of links in the network with average PRR less than 90%.
Forwarders	The percentage of the overhearing-nodes in the network that can overhear a data packet and have a lower path-ETX than the path-ETX of the parent of the sender.
Candidates	The percentage of the overhearing-nodes in the network that can overhear three consecutive data packets and have a lower path-ETX than the path-ETX of the parent of the sender.
Node density	Number of neighbors that can overhear a node's data packet.
Potential neighbors	Number of neighbors that can overhear a node's data packet and have a lower path-ETX than the path-ETX of its parent.
Candidate neighbors	Number of neighbors that can overhear three consecutive data packets from a node and have a lower path-ETX than the path-ETX of its parent.

supports the hypothesis in [3] that WMEWMA-based link estimators performs well on good-quality links. However, they perform poorly on the links of intermediate quality.

5.3.1. Transmission Cost. We compare the transmission cost of BRE with CTP. Figure 10 shows our results for 16 randomly selected node pairs as senders and collection roots. To observe the stability of results over time, we repeated our experiments for BRE and CTP three to five times for each of the 16 node pairs depending upon the difference in number of transmissions. For example, if the average number of transmissions in the case of CTP for a particular node pair differed by more than one transmission after three experiments, we executed the experiments five times. We calculate the average number of transmissions by counting the total data transmissions in the network and dividing it by the total number of unique packets received at the root node. By reprogramming all the motes involved in an experiment for each experimental run, we enforce CTP to reestablish its routing tree. As a result, we intensively validate our results for a particular node pair. In most of the cases, BRE performs better than CTP, averaging to approximately 19% overall reduction in the transmission costs, that is, the total number of transmissions from source to destination for single-node pairs.

Although BRE decreases the total number of transmissions in the network by reducing the number of hops, it increases the number of retransmissions when compared to CTP. This is because it risks transmission over links with high loss rates and retransmits all the lost packets via traditional routing. The percentage of retransmissions is 8.05% for BRE and 3.5% for CTP in the experimental results presented in Figure 10.

To see if these results carried over to other networks, we repeated our experiments on TWIST (the privacy rules of TWIST did not allow us to show the exact locations and IDs of the node pairs used in our experiments. We used the motes placed on the opposite corners (e.g., south-east and north-west corners) and different floors as senders and collection roots in the grid-like TWIST deployment) using a lower transmission power of −15 dBm (see Figure 11). These results are similar and sometimes even better than the results for our experiments on MoteLab. The presented results for an overall of 23 different node pairs from two different testbeds demonstrate the feasibility of our approach.

There are only a few cases (e.g., node pair 140 → 137) in which CTP is marginally better than BRE. This is due to a simple design tradeoff in our prototype implementation of BRE: for analyzing the precise impact of transmission over intermediate links, currently, we always select an

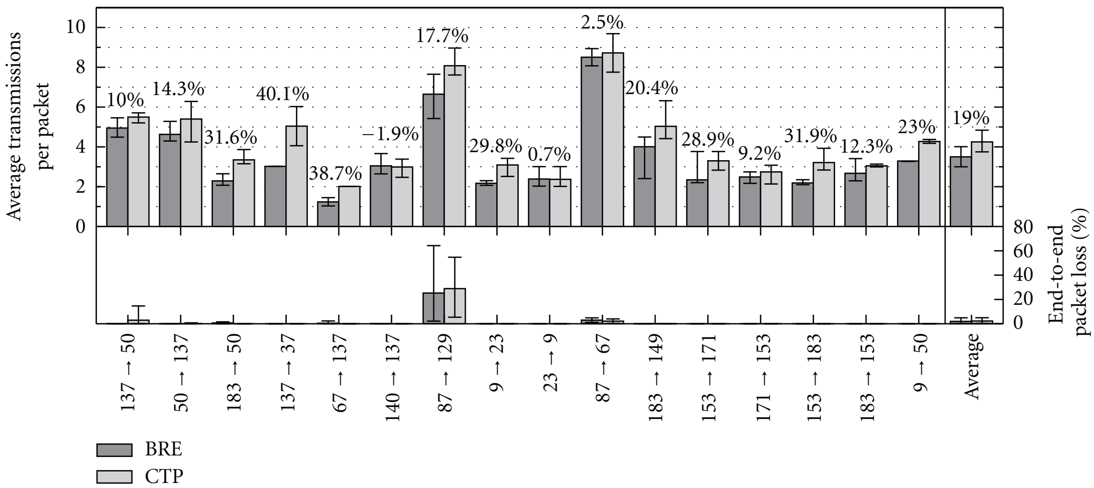


Figure 10: Transmission cost reduction and reliability comparison of BRE and CTP. The graph above shows average number of transmissions per packet using BRE and traditional CTP for our experiments on MoteLab. The graph below shows end-to-end packet loss for the same experiments. The bar represents a node pair's average of five experiments. The error bars represent the highest and the lowest average of the five experiments. The interpacket interval is 250 ms. For these experiment, the average retransmissions is 8.05% for BRE and 3.5% for CTP. The reduction in the number of transmissions in the case of BRE is mostly due to the reduction in the number of hops.

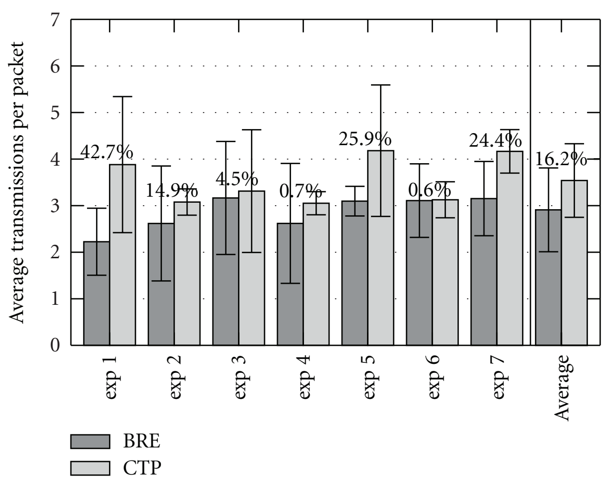


Figure 11: Average number of transmissions per packet for single experimental runs on TWIST. The error bars in this case represent the standard deviation. The results are similar to the MoteLab experiments.

intermediate link without assessing the risk of transmission over such a link. For example, always selecting an intermediate link which has a higher loss rate and only offers a mere 0.1% reduction in transmissions is not always feasible. Frequent failures of transmission over such a link can increase the overall number of transmissions in the network, as depicted in Figure 10 for node pairs 140 → 137 and 23 → 9.

5.3.2. Reliability. Figure 10 also presents the end-to-end packet loss for our experiments. In most cases, packet loss is negligible. From these results, it is fair to conclude that BRE does not affect the reliability of the underlying routing protocol and at the same time reduces the number of transmissions in the network. Using the adaptive routing strategy, BRE makes an attempt to forward packets over long-range bursty links. However, when it fails to transmit a packet over a bursty link, it backs off and allows CTP to retransmit the packet over the traditional path.

The only measurable end-to-end packet loss observed in our experiments is for the node pair 87 → 129 and 87 → 67. We regard these two node pairs as a sanity check for BRE, as they possess a very lossy path. The average number of hops traversed by each packet for these node pairs is 3 and 4, respectively. However, the average number of transmissions per packet is approximately 8. Therefore, the average link quality is less than 50% in both cases. The upper graph in Figure 10 shows that BRE performs better than CTP even in such lossy scenarios. Similarly, the average end-to-end packet loss for BRE in the case of 87 → 129 is less than in traditional routing. However, as discussed in Section 5.4.1, these two node pairs incur a higher transmission overhead.

Although these node pairs are surrounded by a large number of motes, as shown in Figure 9 as well as in official MoteLab connectivity maps on the web, we calculated the node density and link qualities for mote 129 to find the exact reasons of this high packet loss. The average PRR for node 129 was less than 40% for all the neighbors and node density was 4.

5.3.3. Throughput. The two key factors that impact routing throughput in a multihop sensornet are the number of retransmissions and the number of hops. Routing throughput can be increased by minimizing the number of retransmissions for a packet to travel from source to destination. Similarly, each hop traversed by a packet also negatively impacts the throughput. The modest computational capability of a sensor node and protocol-specific considerations result in additional delays, such as packet

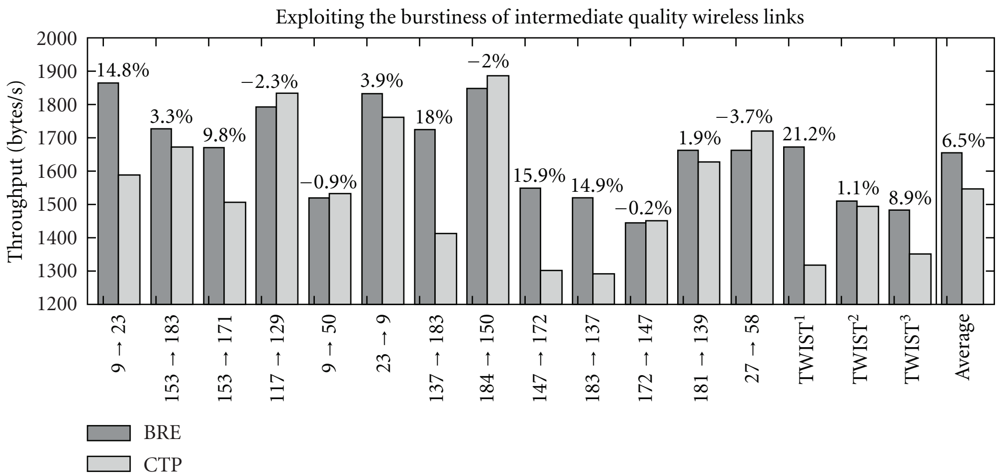


Figure 12: Evaluation results for measured throughput on MoteLab and TWIST. BRE increases routing throughput of traditional routing in most of the cases. The last three bar pairs show the results for our experiments on TWIST.

processing requirements and CSMA-backoff waiting time. BRE adapts to both these key factors. Although it slightly increases the number of retransmissions in the network, the significant reduction in the number of hops contributes to increasing routing throughput (see Figure 12).

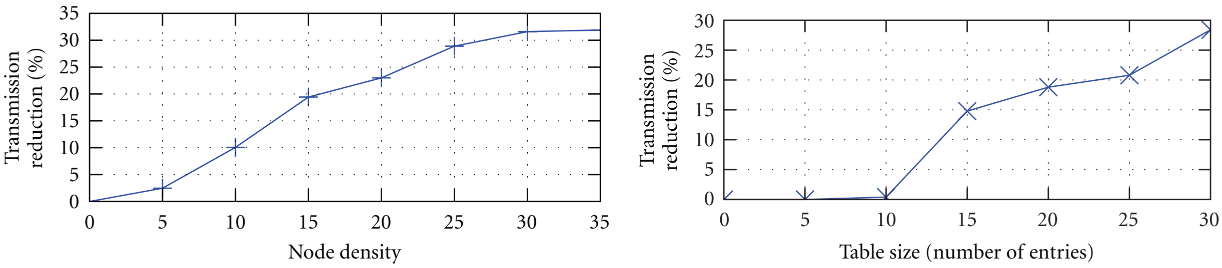
CTP is not an ideal candidate for throughput measurements, as it is a reliable routing protocol originally developed for relatively low traffic rates [4]. We still believe that it can provide us with useful hints about the significance of our approach in terms of routing throughput. Furthermore, we use CTP because we wanted to evaluate the maximum throughput without affecting the delivery reliability, a key property of sensor network routing. Our technique for evaluating throughput is to send a packet by calling the *send* interface of CTP immediately after CTP signals a *SendDone* event for the previous packet. Figure 12 presents our throughput evaluation results (these results are not comparable for corresponding node pairs in our performance measurement results in Figure 10. The reason is that all our experiments were carried out in a span of 3 months. The MoteLab topology changed significantly during that period. This is also the reason that we had to use different node pairs for throughput evaluations) for MoteLab and TWIST. It shows that in most of the cases, due to the reduction in the number of hops, BRE improves the routing throughput, with a maximum improvement of 21%. We expect our approach to be more beneficial if integrated with routing protocols supporting high traffic rates. Moreover, the room for throughput improvement in a bandwidth limited system, like a sensornets, is very limited: Langendoen [23] reports a maximum link throughput of 3 KB/s for CC2420 without routing in TinyOS. Therefore, in addition to our primary goal of reducing the number of transmission, the throughput increase revealed in Figure 12 is a welcome improvement in a multihop sensornet.

Concluding our performance evaluation results, BRE reduces the number of data transmission in the network without affecting delivery reliability. Additionally, by reducing the number of hops for a packet to reach from source to destination, it also enhances routing throughput.

5.3.4. Comparison with a Strawman. In this section, we compare BRE with a simple strawman approach, where if a node with lower path-ETX overhears a packet, it simply forwards it immediately, without updating any tables. The duplicate packets that arrive along the standard path are later dropped by the overhearing nodes. Comparison with a strawman allows to understand the limits and tradeoffs between the transmission cost and throughput of BRE (see Table 3). We performed our experiments by selecting a single node (node 183) as a root at one corner of Motelab, while other nodes (numbered in Figure 9) sending one at a time a total of 500 packets each. The results clearly show that, while strawman increases the routing throughput by 6%, it worsens the number of transmissions by a factor of 1.6 when compared to BRE.

5.3.5. Node Density and State Maintenance. In this section, we analyze how node density and the state of neighboring nodes maintained by BRE impact its performance. Node density positively impacts the performance of BRE as it has more neighboring nodes to choose from. Similarly, higher density increases the probability of finding neighboring nodes with lower path-ETX. This trend is shown in Figure 13(a). A similar trend can also be seen when comparing different experimental classes presented in Table 1 and the corresponding node pairs in Figure 10. The node pairs that belong to high-density experimental class *vertical and diagonal*, such as 137 → 37 and 67 → 137, achieve higher reduction in transmissions.

Finally, we evaluate the impact of table size on the performance of BRE. As discussed in Section 4.1.1, the presence of the original destination of the packet in the routing table of overhearing node (see Section 4.1.1) is necessary for path-ETX comparisons: greater the size of table, higher



(a) Impact of node density on the performance of BRE

(b) Impact of Table size on the performance of BRE for Motelab

Figure 13: Factors limiting the performance of BRE.

Table 3: Summary of the results for BRE and CTP when compared to a strawman. Strawman increases the throughput and the number of transmissions by a factor of 1.06 and 1.6, respectively, when compared to BRE.

Protocol	Transmissions per packet	Throughput bytes/sec
BRE	4.34	1677
CTP	5.25	1583
Strawman	6.88	1793

the probability of finding the original destination of the packet (i.e., parent of the sender node). Figure 13(b) shows that BRE achieves a very small performance gain for neighbor-table sizes of less than 10 entries on Motelab. However, BRE heavily relies on the existence of neighbors with intermediate links in the table. However, higher table sizes (e.g., 20 entries) benefit BRE by increasing the probability of finding the original destination of the packet in the table. As we discussed in Section 4.1.1, this neighbor table dependency can be removed completely by including path-ETX of the parent in each data packet. However, this approach and other routing strategies over BLE are beyond the scope of discussion in this paper, and therefore, we regard it as a future work.

5.4. Intermediate Link Characteristics. After evaluating the performance of BRE, we now analyze the properties of bursty links in more detail. First, we examine the level of correlation between transmission reduction and the number of bursty links used for transmission. Next we present empirical traces from our experiments to investigate the nature and timeliness of intermediate links used for packet forwarding. Finally, we evaluate the impact of different transmission speeds on the performance of BRE.

5.4.1. Timeliness. Another property of bursty links that we investigate is timeliness: how often do they occur and for how long are they active. Figure 14 presents empirical traces from our performance evaluation experiments. It shows that bursty links are regularly available over time and are reliable for variable durations. Figure 15 shows the average consecutive packet transmissions over bursty links in each of our experiment. Some of these links are active for only a

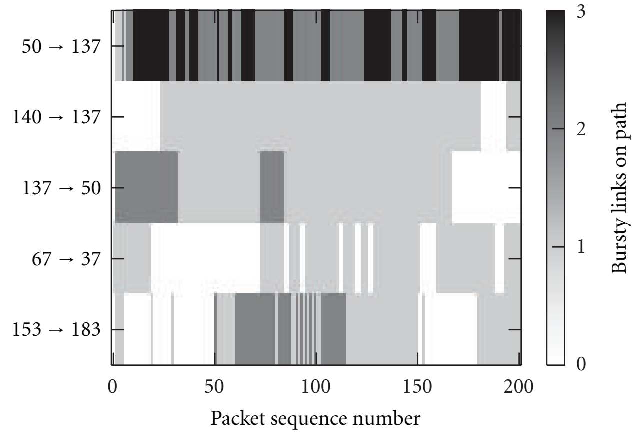


Figure 14: Timeliness of bursty links for 50 second empirical traces for selected node pairs. The graph shows the variability in the duration for which intermediate links are reliable. Most of the successful packets took one or more bursty links on the path from source to destination. Only the white segments in the graph represent complete packet transmissions on traditional path.

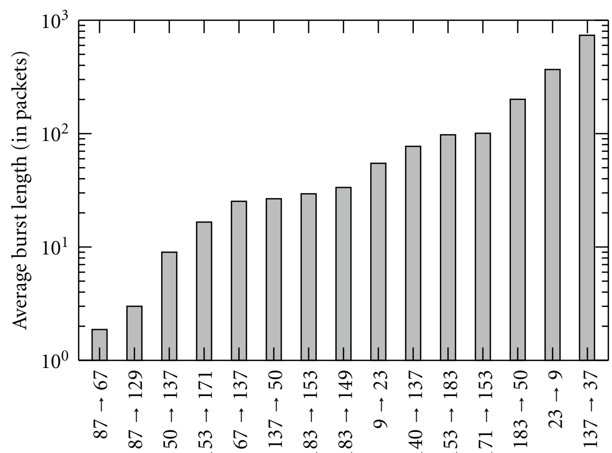


Figure 15: Availability of bursty Links in packet durations. This figure depicts that even relatively long-term (i.e., 750 packet durations) reliable links were also not utilized by CTP. It also shows the limited transmission overhead incurred by BRE.

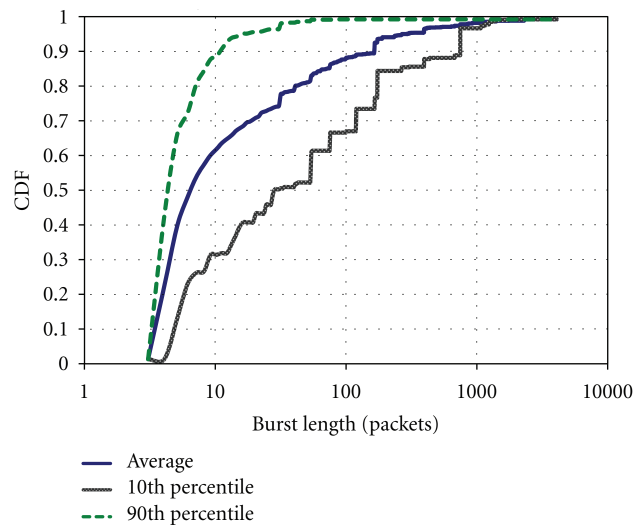


FIGURE 16: Cumulative distribution of packet bursts for all the experimental results presented in Section 5.3.1.

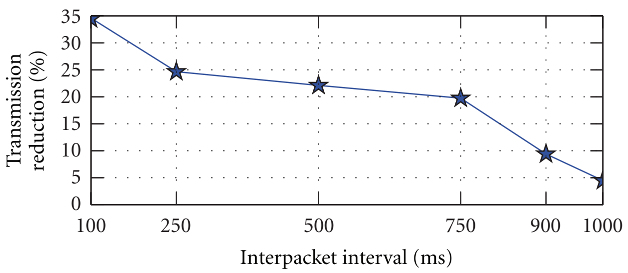


FIGURE 17: Impact of transmission speed on the performance of BRE for node pair 9 → 50. With the increase in the interpacket interval, the performance of BRE drops gradually. For the same node pair, the reduction in the number of transmissions drops from 34% at 100 ms to 4.9% at 1 s.

few milliseconds (e.g., 153 → 183), while others for seconds and even minutes (e.g., 140 → 37 in Figure 14). However, due to the slow adaptivity of traditional routing, that is, CTP, even these relatively long-term reliable links with higher routing progress would not be utilized. Figure 16 shows the cumulative distribution of the burst lengths for all the experimental results presented in Section 5.3.1.

5.4.2. Interpacket Intervals. We investigated the impact of different interpacket intervals on the performance of BRE. Figure 17 shows that the reduction in the number of transmissions decreases with an increase of the interpacket interval. This is because sending packets at higher rates over bursty links maintains a strong correlation between their success or failure providence. While by sending packets further apart, the packet loss during a certain measurement period becomes independent [7]. Thus, the correlation between $n = 3$ and CPDF(3) $= 0.9$ does not strongly hold for very low transmission rates. It means that at lower transmission speeds, it is less probable that BLE declares a bursty link as active in time. Nonetheless, with interpacket intervals as high as one second, BRE still offers a 5% improvement of the transmission efficiency compared to traditional routing.

However, as discussed in Section 4.1, we target sensornet applications with bursty traffic patterns. Such bursty traffic patterns are typically observed in tracking, monitoring, and surveillance applications. In these applications, the interpacket interval is expected to be much lower than 1 second during peek traffic times, that is, the times when the sensor nodes are triggered to track or monitor an activity and report it to the base station.

5.5. Overhead. We divide the overhead introduced by BRE into four different categories, namely, overhearing, processing, storage, and transmission. The passive overhearing technique that we employ in BLE comes at a cost because a node has to listen to the packets that are not addressed to it. However, due to the broadcast nature of wireless transmission, these packets are always received if the node's transceiver is in the receive state. State-of-the-art radio chips, such as the CC2420, can be configured to discard all the received packets that are not addressed to a node. Therefore, the overhead associated with overhearing amounts to packet reception and the processing required to deliver a packet from MAC to the link estimator.

Our current implementation of BRE requires 902 bytes of additional code memory and 270 bytes of additional data memory.

Finally, the only transmission overhead introduced by BRE is the announcement message sent by the overhearing node to the sender node informing about the temporary availability of a bursty link. There is no retransmission of this message because it also serves the purpose of testing the symmetry of that link. Moreover, as mentioned earlier, even a single successful transmission over a bursty link that reduces one hop would cancel out the overhead introduced by this additional message. However, Figure 16 shows that the burst lengths are much longer for most of our experiments. Considering the fact that BRE can reduce transmission costs by up to 40% and increase routing throughput by up to 20%, we believe that the processing, storage, and transmission overhead, presented in this section, is reasonable.

6. Conclusions

We present a simple greedy approach to utilize bursty links of intermediate quality for packet forwarding. Our evaluation results show that, by transmitting over long range intermediate links, the number of transmissions in the network can be reduced.

After evaluating the effectiveness of transmissions over intermediate links, we identify the following aspects as future work: (1) classifying overhearing nodes based on their success history to avoid repeated selection of a node that did not offer significant improvement over the traditional path, (2) implementing and comparing new routing strategies over BLE, such as including path-ETX of parent in the packet headers (cf. Section 4.1.1), (3) integrating BRE with low-power listening techniques, (4) extending this work towards

802.11 networks to show that our approach has a broader relevance in the wireless domain.

Acknowledgments

Many thanks to Kannan Srinivasan and Philip Levis for providing access to their large collection of data sets comprising measurements from both 802.15.4 and 802.11 testbeds. We also wish to thank the developers of MoteLab and TWIST testbeds for allowing us to use their testbeds for evaluation purposes. This research was funded in part by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation Grant DFG EXC 89 and the German Academic Exchange program (DAAD).

References

[1] A. Becher, O. Landsiedel, G. Kunz, and K. Wehrle, "Towards short-term link quality estimation," in *Proceedings of the 5th Workshop on Embedded Networked Sensors (HotEmNets '08)*, Association for Computing Machinery, Chalottesville, USA, 2008.

[2] R. Fonseca, O. Gnawal, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in *Proceedings of the 6th Workshop on Hot Topics in Networks (HotNets '07)*, 2007.

[3] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, 2003.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, 2009.

[5] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of the Computer Communication Review (SIGCOMM '04)*, vol. 34, no. 4, pp. 121–132, 2004.

[6] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Some implications of low power wireless to ip networking," in *Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets '06)*, 2006.

[7] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor System (SenSys '08)*, 2008.

[8] C. Jiao, L. Schwiebert, and B. Xu, "On modeling the packet error statistics in bursty channels," in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN '02)*, 2002.

[9] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 21–30, Association for Computing Machinery, 2007.

[10] R. Fonseca, S. Ratnasamy, J. Zhao et al., "Beacon vector routing: scalable point-to-point routing in wireless sensornets," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design and Implementation (NSDI '05)*, vol. 2, pp. 329–342, USENIX, Berkeley, Calif, USA, 2005.

[11] D. S. J. de Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[12] S. Biswas and R. Morris, "Exor: opportunistic multi-hop routing for wireless networks," in *Proceedings of the Computer Communication Review (SIGCOMM '05)*, vol. 35, no. 4, pp. 133–144, Association for Computing Machinery, 2005.

[13] SING DATASETS. Stanford Information Networking Group.

[14] K. Srinivasan, M. Jain, J. I. Choi et al., "The κ-factor: inferring protocol performance using inter-link reception correlation," in *Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 317–328, 2010.

[15] M. Wachs, J. I. Choi, J. W. Lee et al., "Visibility: a new metric for protocol design," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 73–86, Association for Computing Machinery, 2007.

[16] G. Wittenburg, K. Terfloth, F. L. Villafuerte, T. Naumowicz, H. Ritter, and J. H. Schiller, "Fence monitoring—experimental evaluation of a use case for wireless sensor networks," *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN '07)*, 2007.

[17] M. Welsh, G. W. Allen, K. Lorincz et al., "Sensor networks for high-resolution monitoring of volcanic activity," in *Proceedings of the 20th ACM Symposium on Operating Systems Principles (SOSP '05)*, 2005.

[18] T. He, S. Krishnamurthy, J. A. Stankovic et al., "Energy-efficient surveillance system using wireless sensor networks," *Proceedings of the 2nd international Conference on Mobile Systems, Applications, and Services(MobiSys '04)*, pp. 270–283, 2004.

[19] K. Lorincz and M. Welsh, "Motetrack: a robust, decentralized approach to RF-based location tracking," in *Springer Personal and Ubiquitous Computing*, Special Issue on Location and Context-Awareness, 2006.

[20] C. Sharp, S. Schaffert, A. Woo et al., "Design and implementation of a sensor network system for vehicle tracking and autonomous interception," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN '05)*, vol. 2005, pp. 93–107, 2005.

[21] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesc language: a holistic approach to networked embedded systems," in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '03)*, pp. 1–11, 2003.

[22] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, 2005.

[23] K. Langendoen, "Apples, oranges, and testbeds," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 387–396, 2006.