

# Link Quality Ranking: Getting the Best out of Unreliable Links

Marco Zuniga\*, Izabela Irzynska<sup>†</sup>, Jan-Hinrich Hauer<sup>‡</sup>, Thiemo Voigt<sup>§</sup>, Carlo A. Boano<sup>¶</sup>, and Kay Roemer<sup>¶</sup>

\*Networked Embedded Systems Group, University of Duisburg-Essen, Germany

<sup>†</sup>Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland

<sup>‡</sup>Telecommunication Networks Group, Technical University Berlin, Germany

<sup>§</sup>Swedish Institute of Computer Science, Kista, Sweden

<sup>¶</sup>Institute of Computer Engineering, University of Lübeck, Germany

**Abstract**—Link quality estimation has been an active area of research within the wireless sensor network community. It is now well known that the estimation of reliable links requires few sample packets – less than 10, while the estimation of unreliable links require many more – above 50. In scenarios where unreliable links are ubiquitous, and a rapid transfer of data is needed, traditional estimation techniques are not a viable option. In such scenarios, it is instead sufficient to identify the best link available at any given time. Within this context, we propose Link Quality Ranking (LQR), a mechanism that identifies the best link available when only unreliable links are present. Our testbed results indicate that with one sample packet, the delivery rate of LQR –with respect to the best link available– is above 93%. With 10 sample packets, the performance is above 96%.

## I. INTRODUCTION

Identifying good links for communication is a central problem in wireless networks due to its significant impact on the throughput and overall performance of the network. This task is especially challenging in wireless sensor networks, because the number of probe packets that can be sent for estimation is restricted by the limited energy resources of sensor nodes.

An even more complex problem is identifying the best link available among a set of unreliable links. Several studies have shown that unreliable links are commonplace in sensor networks [14], [19]. These studies also show that the quality of these links varies widely in time and their average capacity is below 90%. Unfortunately, traditional link quality estimation (LQE) techniques require many sample packets (above 50) to provide an accurate estimation of unreliable links [15], [3].

In scenarios where unreliable links are ubiquitous, and a rapid transfer of data is needed, traditional estimation techniques are not a viable option. For example, in Zebranet [12], animals wearing sensors take advantage of sporadic connectivity to transfer data among them. In FleaNet [16], opportunistic file sharing is performed among cars, and the BikeNet project [10] aims at exchanging route information among bikers. All these applications have some common characteristics: (i) a node needs to quickly identify a neighbor to transfer its data, (ii) the networks are sparse, and hence, there is a higher chance of encountering unreliable links, (iii) the nodes have limited (storage, energy) resources which rules out sending the data to all neighbors by means of broadcast. These scenarios require leveraging unreliable links as much

as possible. In order to do this, it is necessary to identify the best link available for communication at any given time.

*The main contribution of our study is Link Quality Ranking (LQR), an alternative to link quality estimation to identify the best link available, even when only unreliable links are present. Contrary to traditional LQE techniques, LQR makes no attempt to estimate the capacity of links. Instead, LQR compares physical-layer metrics and provides a relative ranking among the available links. That is, the best link could be a good or average link in absolute terms, but the node will only know that it is the best link available. Our results show that the performance of LQR with respect to the best available link is above 93% when using the information of a single probe packet, and above 96% when using the information of 10 probe packets. Compared to typical LQE techniques, the number of probe packets is significantly reduced.*

*The second contribution of our study is a framework to quantify the information provided by physical metrics, namely, link quality indicator, signal to noise ratio and packet reception rate. This framework shows that after 10 probe packets, these metrics provide about the same amount of information on link quality. This finding may explain why, when several probe packets are used, there is no estimator in the literature that clearly outperforms the others.*

## II. PRELIMINARIES

### A. Main Idea

In scenarios where nodes observe a mixture of reliable and unreliable links, several link quality estimation methods have been successfully used to identify the best forwarding links [7], [15], [22]. However, in the event that a node observes only unreliable links, these methods may face a sort of catch-22 dilemma: a large number of sample packets are required to accurately estimate a highly variable link, but only a small number of sample packets can be used due to the limited energy resources of sensor networks.

In order to overcome this dilemma, LQR utilizes the following tradeoff: instead of estimating a link-layer metric for each link, LQR performs a pairwise comparison of the physical-layer metrics and selects the best link.

Figure 1 depicts the steps followed by LQR. Step 1, during a time window  $[t_0, t_1]$  a node broadcasts  $n$  probes. Step 2, nodes

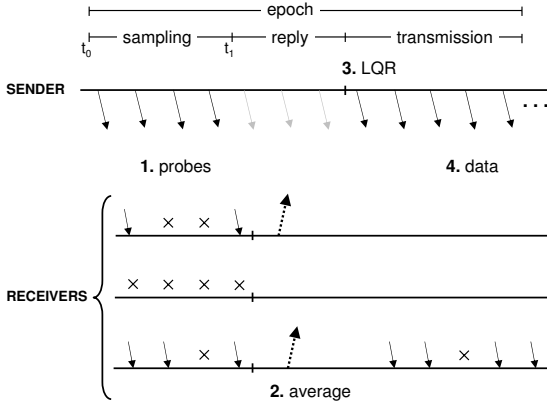


Fig. 1. Steps of LQR

receiving at least one probe are defined as *active receivers*. Active receivers set a timer that fires at the (estimated) end of the probe packet sequence and send back physical-layer metrics. Step 3, after receiving the information from the neighbors, the sender ranks the quality of its outgoing links according to the LQR framework (Section III) and determines the best link. Step 4, the sender transmits the data packets.

LQR creates a *priority* register for each newly observed link, and compares the physical metrics among all active receivers in a pairwise manner<sup>1</sup>. At each comparison, LQR increases the *priority* of the link with the best metrics. After all comparisons have been done, LQR selects the link with the highest *priority*.

**Link Asymmetry.** Unreliable links are prone to be asymmetric [19]. Given two nodes  $(i, j)$ , with  $i$  being the sender, LQR aims at taking advantage of the forward link  $(i \rightarrow j)$ . However,  $j$  may not be able to send the required *reply* packet to  $i$  due to link asymmetry. In order to cope with this, reply packets are sent with a higher output power.

### B. Key Challenge

While the main idea behind LQR is simple, there is an important challenge to overcome: *How much can we trust the result of each comparison?* The fact that the physical-layer metrics of one link are higher than the metrics of another link, might not imply that the capacity of the link is higher. This assessment becomes further complicated if radios are miscalibrated<sup>2</sup> or if the comparison of metrics contradict each other. For instance, considering two links  $(i, j)$ , one metric could state that link  $i$  is better, while the other two metrics could state that link  $j$  is better. In our study, we present a framework to assess the certainty of each comparison.

### C. Physical Metrics

Our study is based on the CC2420 radio [26], a widely used radio in the community. In order to have a better understanding

<sup>1</sup>If an epoch has  $m$  active receivers, there are  $\frac{m(m-1)}{2}$  comparisons.

<sup>2</sup>Some studies [29], [15], [9] have pointed that the calibration of *rss* can be significantly different among radios.

of LQR, first let us present the advantages and disadvantages of the different physical-layer metrics provided by this radio.

**Packet reception rate (*pr*)** has two advantages. First, it is simple and can be utilized on any radio. Second, contrary to *snr* and *lqi*, it is not prone to calibration errors. The packet is either received or not, there are no false positives or false negatives. The main disadvantage of *pr* is the insufficient information provided – the granularity is determined by the number of probes, which in our case are few.

**Signal to noise ratio<sup>3</sup> (*snr*)** has two advantages. First, it provides more information than *pr* – because it has higher granularity. Second, contrary to *pr* and *lqi* it does not have a maximum value, which permits a wider range for classification (*pr* can be at most 1.0 and *lqi* 110). The main disadvantage of *snr* is that it is a noisy metric due to radio miscalibration.

**Link quality indicator (*lqi*)** has two advantages. First, similar to *snr*, it provides more information than *pr*. Second, contrary to *snr*, it does not require to measure the noise floor, which could introduce significant noise. *lqi*'s disadvantage is similar to *snr*'s, it depends on radio calibration (i.e. noisy).

### D. Evaluation Methodology

We used two methods to evaluate LQR, one based on traces gathered on the TWIST testbed [28] (offline analysis), and the other based on an actual implementation on motes.

The traces permitted us to record a large number of instances where unreliable links were present. These traces are used in all subsequent sections except Section IV-A. The setup for obtaining the traces was as follows: a single node broadcasts a sequence of 60,000 packets at a rate of 50 pkts/s with an MPDU size of 20 bytes. All remaining nodes (TWIST has 102 motes) listen for these packets. Upon reception of a packet, active receivers record the sequence number, *rss*, *lqi*, and sample the noise floor three times (to compute the *snr*). We collected traces for seven different senders, which we had identified to provide several unreliable links (reliable links were filtered out). The traces were collected at different times of day and night spanning over several weeks. For each link, the 60,000-packet trace is divided in continuous non-overlapping epochs as depicted in Figure 1. The reply window on each epoch is 400 ms, and the transmission window consists of 100 pkts. The LQR algorithm takes as inputs the sampling window of each epoch, and provides as output the best link. The evaluation compares the delivery rate of the link selected by LQR and the actual best link.

In the analysis based on traces, receivers do not transmit reply packets back to the sender, we assume that the reply packets are delivered successfully (in practice some of these packets may get lost, but the probability is low considering that we use a slightly higher output power for the reply packets [19]). Our full implementation on motes (Section IV-A) eliminates this assumption and confirms that most of the reply packets are actually delivered.

<sup>3</sup>The *snr* was calculated using the received signal strength of the packet ( $S$ ) and sampling the noise floor after reception ( $N$ ):  $snr_{dB} = S_{dB} - N_{dB}$ .

event	comparison $\langle prr, snr, lqi \rangle$	description of event
$\mathbf{e}_1$	$\langle 1, 1, 1 \rangle$	3 metrics agree
$\mathbf{e}_2$	$\langle 1, 1, 0 \rangle$	2 metrics agree
$\mathbf{e}_3$	$\langle 1, 0, 1 \rangle$	1 metric has same value
$\mathbf{e}_4$	$\langle 0, 1, 1 \rangle$	
$\mathbf{e}_5$	$\langle 1, 1, -1 \rangle$	2 metrics agree
$\mathbf{e}_6$	$\langle 1, -1, 1 \rangle$	1 metric disagree
$\mathbf{e}_7$	$\langle -1, 1, 1 \rangle$	
$\mathbf{e}_8$	$\langle 1, 0, 0 \rangle$	2 metrics have same value
$\mathbf{e}_9$	$\langle 0, 1, 0 \rangle$	
$\mathbf{e}_{10}$	$\langle 0, 0, 1 \rangle$	
$\mathbf{e}_{11}$	$\langle 1, -1, 0 \rangle$	2 metrics disagree
$\mathbf{e}_{12}$	$\langle 1, 0, -1 \rangle$	1 metric has same value
$\mathbf{e}_{13}$	$\langle 0, 1, -1 \rangle$	
$\mathbf{e}_{14}$	$\langle 0, 0, 0 \rangle$	3 metrics have same value

TABLE I  
EVENTS WHEN TWO LINKS ARE COMPARED

### III. LINK QUALITY RANKING

Let  $X_i$ ,  $Y_i$  and  $Z_i$  be random variables representing the  $prr$ ,  $snr$ , and  $lqi$  of link  $i$ , and let the triplet  $\mathbf{e}$  represent the comparison of these metrics for two links  $(i, j)$  as follows:

$$\mathbf{e} = \langle \text{sgn}(x_i - x_j), \text{sgn}(y_i - y_j), \text{sgn}(z_i - z_j) \rangle \quad (1)$$

Where  $\text{sgn}(x)$  is the sign function<sup>4</sup>. Table I presents all the possible events  $\mathbf{e}$  that can occur when the metrics of two links are compared<sup>5</sup>. For example,  $\mathbf{e}_4$  represents the event where the  $prr$  of two links is the same, but the  $snr$  and  $lqi$  of one link are higher than the other. Our goal is to identify what events are the most relevant to perform link quality ranking. In order to do so, we need to identify the events that have (i) the highest likelihood of appearance and (ii) the highest accuracy in assessing the relative quality of two links. In the next subsections we investigate these questions.

#### A. Frequency of Events

Not all the events in Table I will appear with the same frequency. For example, intuitively we would predict that the likelihood that two links will have exactly the same metrics is low (i.e.  $\mathbf{e}_{14}$ ). Figure 2 (a) shows the frequency of events for three different sampling windows: 1, 5, and 10 packets. The results are based on 10 traces gathered at different nodes with the parameters described in Section II-D. The number of events (pairwise comparisons) obtained from these traces was more than 150,000.

When one sample packet is sent, all active receivers report a  $prr$  value of 1. Hence, the only valid events are those where  $\text{sgn}(x_i - x_j) = 0$ , such as event  $\mathbf{e}_4$ . When the number of sample packets is increased, most of the events become valid. Figure 2 (a) shows two important trends. First, the distributions for 5 and 10 packets are similar, which may indicate that

<sup>4</sup> $\text{sgn}(x) = \{1, 0, -1\}$  depending if  $x = \{\text{positive, zero, negative}\}$ , respectively.

<sup>5</sup>There are actually 27 events, but events 1 to 13 have a negative equivalent. For example, for  $\mathbf{e}_1$ ,  $i < j$  leads to  $\langle -1, -1, -1 \rangle$  which is equivalent to state that  $j > i$  leads to  $\langle 1, 1, 1 \rangle$

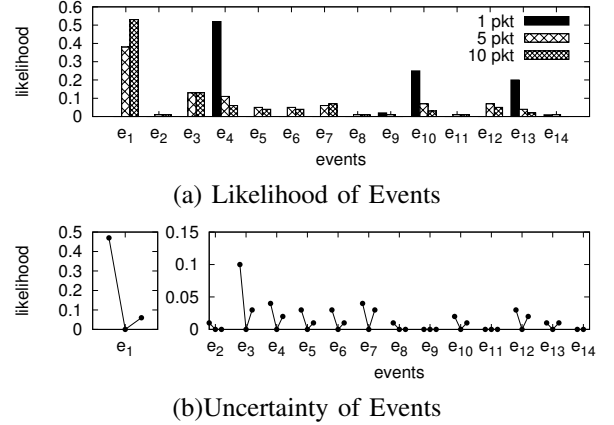


Fig. 2. (a) Likelihood of events in Table I. (b) Uncertainty of events.

the frequency of events reaches a steady state after a few sample packets are used. Second, some events have negligible likelihood of appearance ( $\mathbf{e}_2$ ,  $\mathbf{e}_8$ ,  $\mathbf{e}_9$ ,  $\mathbf{e}_{11}$ , and  $\mathbf{e}_{14}$ ), and hence, LQR can not rely on those events.

#### B. Uncertainty of Events

The previous section indicates the events LQR is most likely to encounter. Now we evaluate how accurate these events are in assessing the relative quality of links.

After comparing a pair of links  $(i, j)$ , each event can have three possible outcomes  $i > j$ ,  $i = j$ , and  $i < j$ . An ideal event would be one that leads to only one of these outcomes, because upon reception of this event, LQR would have complete certainty about the result. The worst event is one where the three outcomes have the same probability, because predicting the ranking would be the same as rolling a three-face fair dice (complete uncertainty). We want to identify the events close to the ideal event.

Figure 2 (b) depicts the uncertainty of events when 10 sample packets are used. Each event is represented by three points linked together in a v-shape style. The left point identifies the times when  $i > j$ , the middle point when  $i = j$ , and right point when  $j > i$ . There are two important things to highlight. First, the probability of obtaining links with the same capacity is zero, or almost zero, for all events. Second, the best events are the ones with highly asymmetric v-shapes such as  $\mathbf{e}_1$ , because they indicate that one of the outcomes is more probable. Following the same line of thought, the worst events are the ones with symmetric v-shapes, such as  $\mathbf{e}_{13}$ .

While the accuracy of each event could be described in terms of conditional probabilities, a better metric to capture uncertainty is entropy [6]. Given a random variable  $R$ , the entropy is given by:

$$H(R) = \sum_r -p(r) \log p(r) \quad (2)$$

Where  $r$  represents the outcomes of  $R$ . Considering that the probability of obtaining links with the same capacity is zero,

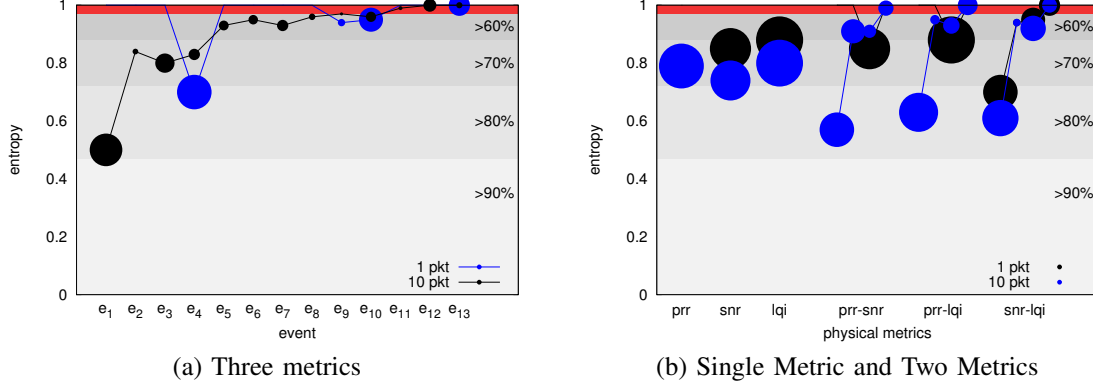


Fig. 3. Entropy of Events for Different Number of Metrics. The size of the bubble indicates the likelihood of the event. The bigger and the lower a bubble is, the better the event. The red zone indicates an accuracy between 50% and 60%. An entropy of 1 is as bad as tossing a fair coin (50% accuracy).

or almost zero (Figure 2b), each event can be seen as a random variable with two possible outcomes ( $i > j, i < j$ ). When two outcomes are available, the entropy varies from zero to one. An entropy of zero indicates a deterministic event (ideal case) and an entropy of one indicates zero information (same as tossing a fair coin). Figure 3 (a) shows the entropy of events for two sampling windows: 1 and 10 packets. The size of each bubble represents the frequency of the event (related to Figure 2a), hence, an event whose bubble is low and big is desirable, because it allows to make a reliable decision and occurs often. The figure is divided in five horizontal zones relating entropy to probability. The red zone indicates the highest uncertainty. The most important contribution of this figure is that it provides LQR with the weights required to increase the rank of a link after each comparison.

**Weights:** Figure 3 (a) quantifies an intuitive phenomena: the better the agreement among the physical metrics, the better the event. Event  $e_1$  has an accuracy close to 90%. Events  $e_2$  to  $e_4$  have an accuracy around 75% for 10 packets and 80% for 1 packet. Events  $e_5$  to  $e_{10}$  have an accuracy between 60% and 65%, and the accuracy of the last three events is similar to that of a fair coin. When faced with these last three events, LQR deems the links to be equal – same as event  $e_{14}$ , and hence, it does not increase the rank of any link.

LQR increases the ranking of a link according to Algorithm 1. First, we set to zero the priority of each active link (line 2). Then, we perform pairwise comparisons according to equation 1 and obtain the corresponding event (line 7). The weights are normalized with respect to the best event ( $e_1$ ) and assigned according to the type of event (lines 9). The priority of the link is updated (lines 15), and the link with the highest priority is selected. In case of a tie, select at random.

### C. Radios With Less Physical Metrics

Not all radios provide the same physical metrics. The simplest radios [21] provide only *pr* capabilities. Narrow-band radios [25] can provide *rssi*, and spread spectrum radios [26] can provide *rssi* and chip correlation metrics (*lqi*). LQR is a general framework that can be applied to these cases as well.

---

#### Algorithm 1 [bestLink] = LQR(activeLinks)

---

```

1: [numLinks, numMetrics] = size (activeLinks)
2: priority(1 × numLinks) ← 0
3: for i=1:numLinks do
4:   linki = activeLinks(i,:)
5:   for j=(i+1):numLinks do
6:     linkj = activeLinks(j,:)
7:     event = sgn(linki − linkj)
8:     sum = sum(event)
9:     switch(|sum|)
10:      case 3: weight = 1.0 // e1
11:      case 2: weight = 0.8 // e2 to e4
12:      case 1: weight = 0.7 // e5 to e10
13:      default weight = 0.0 // e11 to e14
14:    end switch
15:    if sum is positive then
16:      priority(i) += weight
17:    else if sum is negative then
18:      priority(j) += weight
19:    end if
20:  end for
21: end for
22: bestLink = max(priority)

```

---

The steps provided in the previous two subsections can be used to obtain the frequency and certainty of the events involved when less metrics are used (Table II).

In Figure 3 (b) we show the entropy of the individual metrics and their pairwise combinations. The figure depicts the entropy of the events in Table II for one and ten sample packets.

**Single Metric.** When single metrics are used, LQR is simply basic sorting, the best link is the one with the highest metric. However, the framework provides some further insights. First, when 10 sample packets are used, all metrics have similar accuracy (three blue bubbles on the left). This may indicate why, when several packets are used, there is no estimator in the literature that *clearly* outperforms the others. Second, the accuracy of *pr* improves rapidly from 1 packet (entropy=1,

event	comparison $\langle m_1, m_2 \rangle$	description of event
$e_1$	$\langle 1, 1 \rangle$	2 metrics agree
$e_2$	$\langle 1, 0 \rangle$	1 metric has same value
$e_3$	$\langle 0, 1 \rangle$	
$e_4$	$\langle 1, -1 \rangle$	2 metrics disagree
$e_5$	$\langle 0, 0 \rangle$	2 metrics have same value

event	comparison $\langle m_1 \rangle$	description of event
$e_1$	$\langle 1 \rangle$	metric has different value
$e_2$	$\langle 0 \rangle$	metric has same value

TABLE II  
EVENTS WHEN LESS METRICS ARE USED

i.e. random) to 10 packets (entropy $\approx 0.8$ ).

**Dual Metrics.** When 1 packet is used, *prr* has no effect on LQR; *prr-snr* and *prr-lqi* are the same as *snr-only* or *lqi-only*. The main insight of this figure is that the combination *snr-lqi* does not have a major difference in accuracy between 1 and 10 packets. As we will observe in the next section, this explains why 1 single packet is not that bad for ranking.

#### IV. EVALUATION

Our goal is to capture the performance of LQR on scenarios containing unreliable links. Given that the surrounding environment can affect significantly the dynamics of links, we divided the traces in two groups based on the variability of the links. It is important to remark that the traces collected for the evaluation are different from the ones used on the analysis.

Figure 5 depicts the two categories used in our evaluation. The figure shows the mean and standard deviation of the *prr* for two sample traces. Each mark represents a link and the vertical line indicates the links that were filtered out (reliable links). The top figure depicts scenarios where unreliable links have a much higher variance. These traces were gathered during the day, when the surrounding environment changes significantly through time. The more stable scenarios were collected at early hours of the morning when the activity on the building was minimal<sup>6</sup>.

The performance of LQR is evaluated relative to the best link available at each epoch. We use two parameters to capture

<sup>6</sup>If the surrounding environment and ambient temperature remain relatively constant, the quality of unreliable links does not change much over time [24].

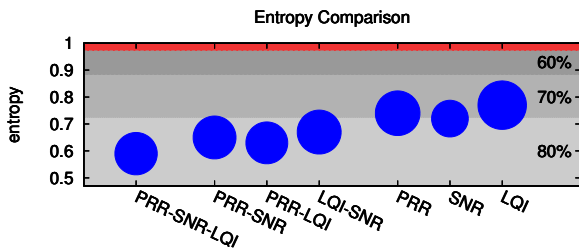


Fig. 4. joint Entropy for Different Combinations

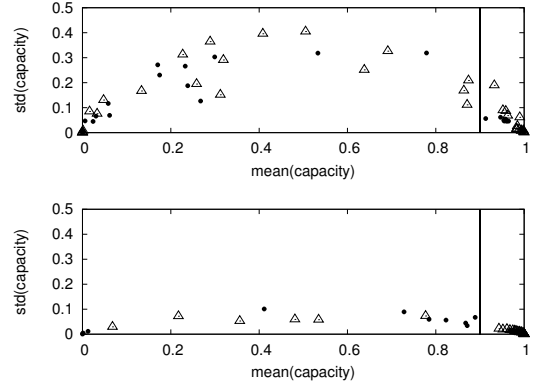


Fig. 5. Different scenarios with unreliable links. Each figure shows two traces (dots and triangles). Top figure: variable links. Bottom figure: stable links.

Scenario	Number of Traces	Delivery Rate	
		(mean)	(std)
high variance	10	0.91	0.10
low variance	5	0.79	0.06

TABLE III  
SCENARIOS WITH UNRELIABLE LINKS

this performance: normalized capacity and standard deviation of top-rank (std-*rank*). At each epoch, the capacity of the link selected by LQR was normalized with respect to the capacity of the best link. If several links were tied at the top rank, we calculated their standard deviation (std-*rank*). An std-*rank* of zero denotes that a single link occupied the top rank, while a high std-*rank* indicates that LQR was not able to identify a best link. Based on these metrics, the performance of LQR can be described as follows: the higher the normalized capacity and the lower the std-*rank*, the better the performance of LQR.

Figure 6 presents the performance of LQR for different number of sample packets. The figure depicts the performance of each individual metric and the combination of the three metrics. Table III shows the number of traces evaluated for each scenario (high and low variance). The table also shows the mean delivery rate and standard deviation of the best link. This information helps to position the performance of LQR in absolute terms. For example, in Figure 6 (a) the normalized capacity 1.0 represents an average delivery rate of 0.91. The results of Figure 6 permit us to obtain some important insights.

*One probe is not enough to estimate a link's capacity, but it may be good enough to identify the best link available.* Several studies have indicated that estimating unreliable links require several tens of packets [15], [2], [3]. However, if the aim is to simply identify the best link, then one probe does a decent job (except for *prr* which the best it can do is to select a node at random). In Figures 6 (a) and 6 (c), we observe that ranking based on *prr-snr-lqi* always provides the best performance – because they combine information from three metrics –, but in the case where unreliable links have high variance, even *snr* or *lqi* alone have a good performance (Figure 6a). The reason for

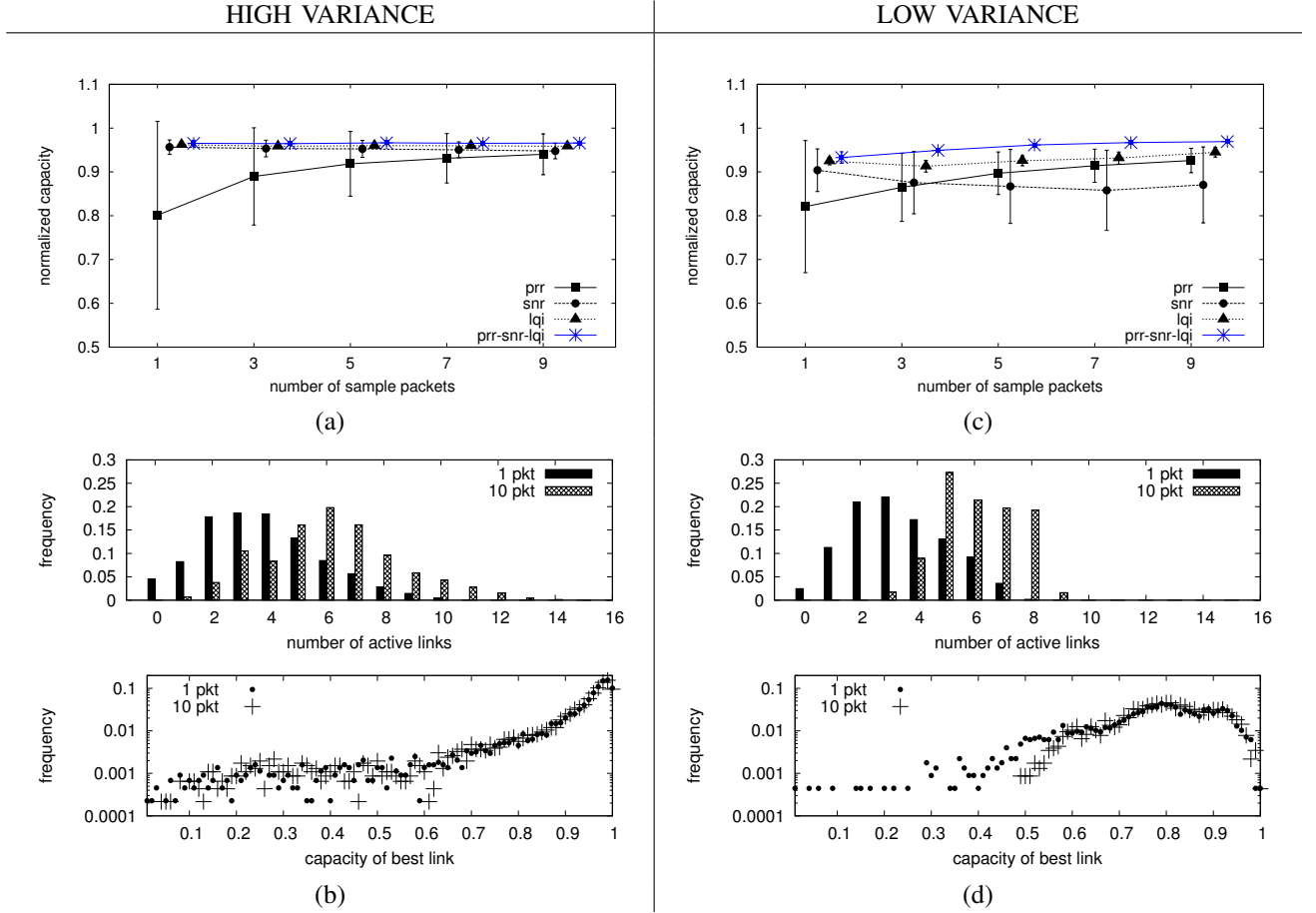


Fig. 6. Performance Of LQR Based on Traces.

this behavior is explained next.

*When faced only with unreliable links, high link variance helps.* When the surrounding environment is dynamic, links have a wider variability in link quality. This means that unreliable links have a higher chance of becoming temporarily good or bad. When links are good, simple ranking has an easier time identifying good links for all metrics (except *pr*). Figures 6 (b) and (d) capture this phenomena. The top histograms in these figures indicate that high variance leads to more active links, and the bottom histogram indicates that high variance leads to better links.

*All metrics seem to reach a steady state after 10 probes.* Some studies have reported that the estimation of unreliable links require 120 samples of *lqi* to reach a stable value [15], [5]. Figures 6 (a) and 6 (c) seem to indicate that for the simpler problem of ranking, significantly less packets are required to reach a steady state (for all metrics). Notice however that LQR with *pr-snr-lqi* does not benefit much from sampling beyond 1 packet, i.e. it is the most efficient method.

*Limitations on floating point operations have a limited effect.* Our implementation of LQR rounds the average value of *snr* and *lqi* before sending the reply packets. This rounding effect is the main reason why the *snr* curve in Figure 6c decreases its performance as the number of sample packets

Parameter	Effect
- transmission window	- smaller windows benefit LQR
- reply window	- smaller windows benefit LQR
- transmission rate	- higher rates benefit LQR
- use of floating point on reply packets	- minor benefit on LQR with <i>pr-snr-lqi</i> major benefit on LQR with <i>snr</i>

TABLE IV  
IMPACT OF DIFFERENT PARAMETERS ON LQR

increases. Low-power MCUs (like the MSP430) typically do not have a dedicated hardware floating point unit. Without this unit, floating point operations consume significant memory resources and processing time (besides the extra resources required to transmit longer packets).

*Other Parameters.* LQR consists of several parameters. Due to space constraints we can not present all the results. In Table IV we describe succinctly the effect of these different parameters. In general we found the performance of LQR to be robust independent of the parameters used.

#### A. LQR Implementation in TinyOS 2

In Section II-D, our empirical evaluation of LQR was performed post-facto (offline) based on a large set of traces collected in the TWIST testbed. While these traces permitted



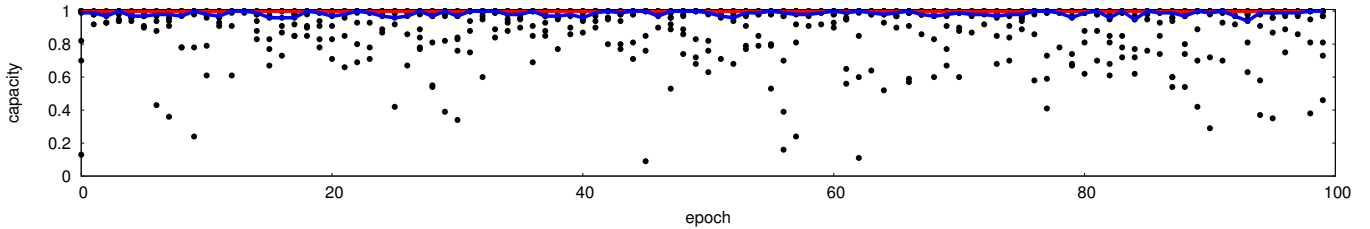


Fig. 7. LQR Running on Motes. 100 epochs were run. The black dots indicate the links available at each epoch. The red line indicates the best link available. The blue line represents the links selected by LQR with one sample packet.

an easy collection of scenarios containing unreliable links, they do not test important features such as the reply sent by receivers or the LQR algorithm running on a mote-class device. Our implementation in TinyOS removes these limitations and realize the complete set of steps depicted in Figure 1. Reply packets avoid collision through a simple TDMA-MAC, where time slots were assigned uniquely based on node IDs. The nodes that had reliable links with the sender were not activated.

We evaluated LQR as follows: in each epoch the sender sent a *single* probe packet; receivers responded with one reply packet each; the TDMA-slot was of 20 ms and the TDMA-frame was of 400 ms; then the sender transmitted 100 data packets at a rate of 50 pkts/s. Figure 7 shows the capacity of all links (black dots) during a representative experiment that spanned 100 epochs. The red line identifies the best link (based on the “god’s view perspective” obtained from the detailed statistics continuously output by all nodes over the serial control channel), while the blue line marks the estimate of the LQR algorithm executed on the sender node. It can be seen that LQR selected links that were either the best or very close to the best link available at each epoch.

#### B. Generalization of LQR to Other Scenarios and Platforms

The steps required to run LQR are not trivial (Sections III-A and III-B). In order to assess the applicability of LQR, it is necessary to validate if the results hold (or not) for other scenarios. We hypothesize that our results would hold for other scenarios using the CC2420 transceiver. Even though our evaluation of LQR was limited to a single indoor testbed, we took care of including different dynamics, i.e. various times of day which affect multipath and interference levels. These effects had a minor impact on the performance of LQR. On the other hand, if the radio transceiver changes (for instance, CC1000), the analysis would need to be adjusted because LQR relies on the probabilistic relationships of *snr* and *lqi*, and these metrics are affected by radio calibration.

#### C. Comparison with Link Quality Estimators

LQR solves a simpler problem than link quality estimation (LQE): LQE estimates the capacity of a link, while LQR identifies the link with the best capacity. Hence, a direct comparison would not be appropriate because estimators require more information (probe packets) to determine the capacity, for example, ETX [7] and [3] rely on periodic data.

Other estimators such as RNP [1], would not provide any meaningful result unless a large sequence of packets are used. In the related work section, we describe how LQR relates to these and other well known estimators in the literature.

### V. RELATED WORK

Link quality estimation has been an active research topic in sensor networks. In this section, we position our work within the related literature.

**Estimating reliable links with few samples.** In [15], Srinivasan and Levis report that for well-calibrated radios *rssr* is a good indicator of link quality for reliable links. They hypothesize that a *single packet* could be a good estimate of the average *rssr* over many packets.

*lqi* has also been used as a fast estimator of reliable links. The 4-bit estimator [22] uses one high-*lqi* sample to identify reliable links, and MultiHop-*lqi* [27] establishes routes selecting links with the best *lqi*. Similarly, Boano et al. [5] find that the variance of highly-reliable links is minimal, and hence, few packets can be used to identify good links.

The previous studies show that the *lqi* of a single packet is sufficient to identify reliable links – links with capacity 1 (or close to 1). Our study complements these findings with a new insight: a single packet is also good to identify (not estimate) the best unreliable link. Furthermore, by combining the information of all metrics, as in Table I, the ranking is guaranteed to perform better than individual metrics (Figure 6).

**Estimating unreliable links with several samples.** Several studies have pointed out that estimating unreliable links require a large number of packets or periodic probing. In [15], the authors report that *lqi* requires about 120 packets for an accurate estimate of link quality. Similarly, Meier et al. [2] and Boano et al. [5] report a wide variance of *lqi* in unreliable links. Other studies have reported similar difficulties with *rssr* measurements [8], [11], in particular if the radio is miscalibrated [9], [15].

*prr* has also been used actively to estimate link quality. In one of the earliest studies, Woo et al. [3] combine *prr* with weighted moving average techniques to estimate unreliable links. With periodic sampling, the estimator can reach 10% accuracy within 40 probes. ETX [7] combines the inverse of the *prr* in both directions of the link, and it performs well on protocols using periodic sampling (approximately 1 packet per second). Cerpa et. al. propose RNP [1] to complement

ETX. For a series of success and drop packets, RNP considered not only the  $prr$ , but also the “holes” in between successful transmissions. The bigger the holes, the higher the penalty. A more recent algorithm EAR [13] utilizes a weighted function of  $prr$  to exploit under-utilized asymmetric links.

The estimators above are all based on one metric. Other researchers combine several metrics to achieve higher accuracy. Rondinone [18] et al. multiply  $prr$  with a normalized value of  $rssi$  in order to differentiate good links from excellent links. Similarly, Boano et al. [4] combine  $prr$ ,  $snr$  and  $lqi$  into a *triangle* metric, and Baccour et al. [20] combine  $prr$ ,  $snr$ , link asymmetry and link stability with fuzzy-rules. The aim of these two estimators is to differentiate among bad, average, good, and excellent links. DUCHY [9] combines  $lqi$  and  $rssi$  to make a better classification of good and bad links.

The studies described above rely on several tens or hundreds of packets to estimate unreliable links. LQR proposes a new paradigm: instead of using several packets for estimation, let's use fewer packets for ranking. It is important remark that ETX [7] and the EWMA-based estimator in [3] are based on packet success rates, hence, their ranking performance would be similar to the  $prr$  metric shown in this study.

**Exploiting Temporal Correlation.** LQR does not aim for applications requiring periodic transmissions, but one-hop bursts. From that perspective, LQR is related to studies taking advantage of the good quality periods of unreliable links..

In ExOR [23], Biswas and Morris propose to broadcast a packet without explicitly stating the receiver. The node that receives the packet, and that is closest to the destination, is in charge of forwarding the packet. LQR shares the same spirit as ExOR but differs in two important ways. First, ExOR requires a distributed algorithm so receivers avoid sending duplicated packets (LQR does not require such mechanism). Second, LQR ranks links to be used on a future window, while ExOR performs broadcast transmissions that do not require prior estimation. In [17], Alizai et al. report that if three consecutive packets are received on an unreliable link, there is a high chance that the quality of the link will be good over a short period of time. LQR builds on top of these studies by ranking unreliable links with as little as one packet sample.

## VI. CONCLUSIONS

Our study proposes link quality ranking (LQR), a new way to tackle a problem that has been mainly approached based on estimation techniques. LQR utilizes few sample packets to identify the best link available when only unreliable links are present. Our results indicate that with a single probe packet, LQR has a delivery rate above 93% compared to the best link available. The characteristics of LQR are ideal for sparse deployments where a rapid transfer of data is needed. In these scenarios, reliable links may not be always available, and nodes would need to quickly identify the best unreliable link.

The efficiency of LQR, in terms of number of sample packets, comes at a cost: LQR does not assign a link-quality metric. This metric-less approach limits the application of LQR to multi-hop protocols that require shortest-path calculations.

**Acknowledgement** This work has been funded by an IRC-SET (PD200857), SFI (SFI08-CE-I1380), CONET (FP7-2007-2-224053), SSF and VINNOVA .

## REFERENCES

- [1] A. Cerpa, J. L. Wong, M. Potkonjak, D. Estrin. Temporal Properties of Low-Power Wireless Links: Modeling, Implications on Multi-Hop Routing. In *MobiHoc*, 2005.
- [2] A. Meier, T. Rein, J. Beutel, L. Thiele. Coping with Unreliable Channels: Efficient Link Estimation for Low-Power Wireless Sensor Networks. In *INSS*, 2008.
- [3] A. Woo, T. Tong, D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *SenSys*, 2003.
- [4] C. Boano, M. A. Zúniga, T. Voigt, A. Willig, K. Römer. The Triangle Metric: Fast Link Quality Estimation for Mobile Wireless Sensor Networks. In *ICCCN*, 2010.
- [5] C. Boano, T. Voigt, A. Dunkels, F. Österlind, N. Tsiftes, L. Mottola, P. Suárez. Exploiting the LQI Variance for Rapid Channel Quality Assessment. In *IPSN*, 2009.
- [6] T.M. Cover, J.A. Thomas, and J. Wiley. *Elements of information theory*. Wiley Online Library, 1991.
- [7] D. Couto, D. Aguayo, J. Bicket, R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *MobiCom*, 2003.
- [8] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian. Measurement, Characterization of Link Quality Metrics. In *GLOBECOM*, 2003.
- [9] D. Puccinelli, M. Haenggi. Duchy: Double Cost Field Hybrid Link Estimation for Low-Power Wireless Sensor Networks. In *HotEmNets*, 2008.
- [10] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G.S. Ahn, and A.T. Campbell. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)*, 2009.
- [11] J. Zhao, R. Govindan. Understanding Packet Delivery Performance in Dense Wireless Sensor Networks. In *SenSys*, 2003.
- [12] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs, early experiences with zebranet. In *ASPLOS-X*, 2002.
- [13] K. Kim, K. G. Shin. On Accurate Measurement of Link Quality in Multi-hop Wireless Mesh Networks. In *MobiCom*, 2006.
- [14] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, P. Levis. The Beta Factor: Measuring Wireless Link Burstiness. In *SenSys*, 2008.
- [15] K. Srinivasan, P. Levis. RSSI is Under Appreciated. In *EmNets*, 2006.
- [16] U. Lee, J. Lee, J. Park, and M. Gerla. Fleanet: A virtual market place on vehicular networks. *Vehicular Technology, IEEE Transactions on*, 2010.
- [17] M. Alizai, O. Landsiedel, J. Link, S. Götz, K. Wehrle. Bursty Traffic over Bursty Links. In *SenSys*, 2009.
- [18] M. Rondinone, J. Ansari, J. Riihijärvi, P. Mähönen. Designing a Reliable, Stable Link Quality Metric for Wireless Sensor Networks. In *RealWSN*, 2008.
- [19] M. Zúniga, B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. In *ACM TOSN*, 2007.
- [20] N. Baccour, A. Koubâa, H. Youssef, M. Jamâa, D. Rosário, M. Alves, L. B. Becker. F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks. In *EWSN*, 2010.
- [21] Nordic Semiconductor. nRF24L01-Single Chip 2.4GHz Transceiver.
- [22] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis. Four-Bit Wireless Link Estimation. In *HotNets*, 2007.
- [23] S. Biswas, R. Morris. ExOR: Opportunistic Multi-Hop Routing for Wireless Networks. In *SIGCOMM*, 2005.
- [24] L. Tang, K.C. Wang, Y. Huang, and F. Gu. Channel characterization, link quality assessment of IEEE 802.15. 4-compliant radio for factory environments. *Industrial Informatics, IEEE Transactions on*, 2007.
- [25] Texas Instruments. *CC1000 datasheet*.
- [26] Texas Instruments. *CC2420 datasheet*.
- [27] TinyOS MultiHopLQI routing algorithm. <http://www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI/>, 2004.
- [28] V. Handziski, A. Köpke, A. Willig, A. Wolisz. TWIST: A Scalable, Reconfigurable Testbed for Wireless Indoor Experiments with Sensor Networks. In *RealMAN*, 2006.
- [29] Y. Chen, A. Terzis. On the Mechanisms, Effects of Calibrating RSSI Measurements for 802.15.4 Radios. In *EWSN*, 2010.