A Multilayer Link Quality Estimator for Reliable Machine-to-Machine Communication

Wendley S. da Silva*, **Daniel F. Macedo***, **Michele Nogueira**¶, **Thi Mai Trang Nguyen**†, **José Marcos S. Nogueira***
*Department of Computer Science, Universidade Federal de Minas Gerais, Brazil
†Laboratoire d'Informatique de Paris 6, Sorbonne Universités, UPMC, CNRS, UMR 7606, France
¶Department of Informatics, Federal University of Paraná, Brazil

E-mails: wendley@ufmg.br; Thi-Mai-Trang.Nguyen@lip6.fr; {damacedo,jmarcos}@dcc.ufmg.br; michele@inf.ufpr.br

***Abstract*—An ever-growing number of embedded devices supports different kinds of applications, such as healthcare, surveillance, gas monitoring, and others, that require an elevated level of communication reliability. However, the expected high density of those embedded devices increases the competition for frequency spectrum, making it difficult to achieve a reliable machine-to-machine (M2M) communication. To overcome these difficulties, the use of link quality estimators (LQE) is crucial to provide a solid communication. In order to provide robust and faster communication under harsh conditions, this paper proposes a new LQE, called PRR^2, which uses two metrics and two levels of PRR (Packet Received Ratio). The use of two PRR sliding windows captures link quality variations in the short term and also considers the long-term. PRR^2 is compared against the state of the art on a prototype using USRPs, and the results show that the proposal reduces the number of retransmissions and increases the delivery rate, which are two important metrics for link layer reliability.**

I. Introduction

The popularization of embedded devices has attracted the attention of researchers and network designers, in part, due to the challenges produced by the high density and huge amount of traffic. It is expected that the number of embedded and connected devices will grow 9-fold from 2015 to 2020, as people and industries boost the deployment of sensors and smart devices [1]. Hence, the ever-growing amount of co-located devices and traffic flowing from one device to another require new approaches to maintain an effective communication. This type of traffic is usually referred to as Machine-to-Machine (M2M) communication, and is employed in industrial control and smart-space actuation and takes place with minimum or no human interaction [2].

Simultaneously, wireless communication supports more and more critical applications, requiring high reliability and performance [3]. For instance, the telemetry of cars on the move, blood pressure and heartbeat monitoring of the elderly demand high availability even over electromagnetically noisy areas. Further, with the growth in the number of devices, the competition in the wireless medium has increased, decreasing the network throughput year over year [4].

Communication protocols employ algorithms to dynamically adapt the data transmissions under the harsh conditions of the medium, in order to maximize its reliability as well as the link usage. Examples of such adaptations are transmission power control, bit rate adaptation or even the construction of routes taking link quality into account. These strategies require accurate *Link Quality Estimators (LQE)*. Effective estimators improve network throughput, reliability and energy-efficiency by increasing the percentage of successfully delivered messages and avoid retransmissions over low quality links.

One of the main challenges to obtain an accurate LQE is to identify an optimal trade-off between the stability of LQE and the ability to adapt according to channel variations. The estimation must be computed fast and the LQE must be able to detect link quality changes resulting from a dynamic physical layer. Further, reflection, diffraction, scattering, shadowing and multipath can affect the LQE accuracy.

The proposed estimator PRR^2 stands out for its ability to provide accurate estimates using only transmitter side RSSI and received acknowledgment information, without requiring changes to the receiving device. In order to address the limited reliability and performance on M2M communications, this work proposes a multilayer LQE to provide an accurate and power-safe estimation, using a merge of two Packet Reception Rate (PRR) and Received Signal Strength Indicator (RSSI) as input. This approach enables to identify long-term variations on link quality, without disregarding the variations in the short term. PRR^2 is implemented using USRPs (Universal Software Radio Peripheral), and it is evaluated in conjunction with a power control algorithm. Results show that PRR^2 provides a better packet delivery rate with fewer retransmissions than the state of the art.

The paper proceeds as follows. The related works are discussed in Section II. The proposal and experimental scenario are detailed in Sections III and IV. Section V presents the results. Finally, Section VI concludes the paper.

II. Related Work

Recent research investigated how to achieve a better prediction, in runtime, of the behavior of wireless links.

The following estimators are based in packet statistics, or software-based: Window Mean with Exponentially Weighted Moving Average (WMEWMA) [5] uses an Exponential Weighted Moving Average (EWMA) filter to combine recently and previously computed PRR estimates. Expected Transmission Count (ETX) [6] is an estimator located on the receiver side and has a similar strategy to Required Number of Packet retransmissions (RNP) metric. It considers the asymmetric link to estimate the quality of transmission in the uplink and downlink directions. The combination of both estimated values provides an estimate of the quality of the bidirectional link.

The ensuing three estimators are called multilayer estimators, as they treat information from different network layers. Kalman-filter-based LQE (KLE) [7] uses a Kalman filter scheme to estimate PRR using RSSI. The authors evaluated their proposal using sensor motes with CC2420 transceivers running ZigBee. Four-bit (4B) [8] uses four bits of information: one from the physical layer, representing the channel quality during a packet; one from the link layer, that measures if packets are delivered and acknowledged; and two from the network layer employed to identify which links are the most valuable for higher-layer performance. The prototype has shown significant improvements on cost and delivery rate when compared to a multi-hop LQI (Link Quality Indicator). The authors used TinyOS 2 and a testbed using TelosB motes. Triangle Metric LQE [9] provides a metric that combines geometrically the information of PRR, LQI, and SNR. The receiver calculates the mean of the LQI and SNR values over an estimation window of 10 packets and a second function include the PRR information.

The following three estimators are based on packet statistics and utilize training and learning processes. Bio-inspired LQE [10] is a self-improving estimator based on neural networks. It has been evaluated by simulations considering a 802.11b mesh network [11]. In [12], Baccour *et al.* presented Fuzzy-LQE (F-LQE), a fuzzy logic link quality estimator. It utilizes linear membership functions to compute the quality estimation based on PRR, link asymmetry, stability and SNR. The estimator reduces the number of packet retransmissions by up to 32% [13]. Authors employed TelosB motes to evaluate F-LQE [14]. Liu and Cerpa introduced 4C (Foresee), a novel link estimator that is data-driven [15]. It applies link quality prediction along with link estimation and uses a machine learning approach to predict the short temporal quality of the link with both physical layer information and PRR. The output is the success probability of delivering the next packet.

Unlike the above-mentioned LQE methods, the proposal in this paper considers simultaneously two PRRs and the RSSI. It does not requires changes on the receiving side, enabling the adoption in several M2M communication systems.

III. A MULTILAYER LQE FOR RELIABLE M2M COMMUNICATION

This section presents PRR^2, the novel link quality estimator approach designed to machine-to-machine communication. The PRR^2 estimator runs in the sender and considers two levels of PRR: one is computed over a short moving window, and another is calculated using a long moving window, as shown in Fig. 1. The use of two PRR sliding windows captures link quality variations in the short term, while at the same taking into account the long-term. This is useful in many situations: (i) to prevent erroneous overestimation due to a quick oscillation in the quality link; (ii) to reduce the effects of the anisotropic radiation pattern of the antennas [16], since the radio irregularities can be mitigate using two levels of confirmation, a short and long terms.

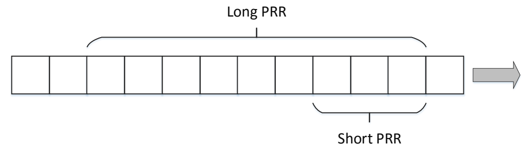

Fig. 1. PRR^2 scheme: short and long PRR windows

The approach also considers the RSSI estimation so it is a multilayer estimator, which provides greater robustness. The multiple layers add quality link information that could not be estimated separately, *e.g.*, an estimator that only considers the local RSSI cannot capture the quality of the whole link, while an estimator using only the PRR may not be able to capture the momentary link quality fluctuations close to the transmitter.

RSSI is used to identify unstable links, and to fill the gap when PRR cannot be reliably computed (i.e. when there are not enough packets to estimate the PRR, or when the interval among new packets is too long). While the PRR is less reactive to signal strength variations, which can indicate high competitiveness in the channel, the RSSI instantly captures these variations.

PRR^2 has as main inputs the value of the RSSI on the transmitter side, as well as the indication of whether a packet has been received or not. The PRR^2 output is a real number between $[0, 1]$, which can be used by the protocols in the network (e.g. a power control algorithm, a routing protocol) to improve the quality of the links used on the network or to select the most appropriate links. Fig. 2 shows the inputs and output of PRR^2. The operation of PRR^2 has the following steps: smoothing, PRR calculation, normalization and computation.

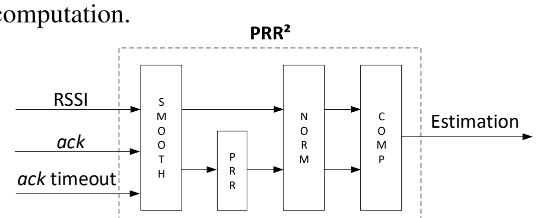

Fig. 2. Diagram of PRR^2

Smoothing: Since the wireless medium varies frequently, to smooth out short-term fluctuations, both PRR and RSSI are computed using Exponential Moving Average (EMA), except when there are not enough packets to reliably compute and EMA, and the Simple Moving Average (SMA) is used instead:

$$smooth(W) = \begin{cases} \frac{\sum_{i=1}^{|W|} w_i}{|W|}, & \text{if } |W| \leq k \\ \sum_{i=1}^{|W|}(1-\alpha)\alpha^{i-1}w_i, & \text{if } |W| > k \end{cases} \quad (1)$$

where W is the observation window vector, $|W|$ represents the window length, w_1 is the most recent measurement, $w_{|W|}$ is the least recent one, k represents the threshold to switch to/from EWMA and SMA, and α is the smoothing factor, computed as $\alpha = 2/(|W| + 1)$, according to [17]. This approach minimizes recent random fluctuations, providing a more stable estimate. The use of SMA for smaller windows occurs because the EMA algorithm requires a minimum number of values to settle correctly.

PRR computation: Our algorithm calculates the two packet reception rates from the received ack packet and the ack timeout indications. One PRR is computed over a small window of data, and another is computed over a long window of data. The PRR is obtained using Eq. 2, where i and j represent, respectively, the indexes of the start and end of the observation window. To determine the *longPRR*, the j index must be greater than the j index of *shortPRR*.

$$PRR_{i,j} = \frac{received\ ack_{(i,j)}}{received\ ack_{(i,j)} + timeout\ flag_{(i,j)}} \quad (2)$$

Normalization: By convention, the values estimated by PRR^2 are normalized between [0, 1], in which higher values indicate better link qualities. The normalization process of PRR values is straightforward, since the PRR already lies within that range. The normalization of RSSI values requires further consideration: The range should represent the highest and lowest RSSI values on a real deployment. Further, they should be calibrated based on the capacities of each transceiver.

Computation: The choice for the metric to use on PRR^2 is based on Eq. 3. $longPRR$ represents the PRR computed over the long window. $shortPRR$ represents the PRR calculated using the short window, and $estRSSI$ is the RSSI noted on sender location. At this stage all three values are already normalized between 0 and 1:

$$PRR^2 = min(longPRR, shortPRR, estRSSI) \quad (3)$$

where estAIFD acts according to the following strategy: initially estAIFD has a value of 0.5. For each ack successfully received consecutively, there is an increment of 0.05 in its value, and for each ack lost consecutively, there is a reduction of 0.1 for the first loss, reduction of 0.2 for the second loss, reduction of 0.3 for the third loss, and reduction of 0.4 for the fourth consecutive loss of ack.

The rationale of use a minimum function in Equation 3 is to act conservatively. Thus, it selects the minimum of $longPRR$, $shortPRR$, the estimated RSSI and $estAIFD$.

IV. Experimental setup

The proposed LQE is evaluated in a prototype, using software-defined radios. We devised an experiment that mimics the transmission of data among two low-power devices, e.g. a heart rate monitor sending packets to a sink. In this scenario, the main goal is to reliably transmit data, consuming the least amount of energy. Hence, we used the link estimations to dynamically adjust the transmission power of the transmissions.

The experiments are performed using the Ettus USRP model B210 [18], which provides a dual channel transceiver operating in the 70 MHz - 6 GHz band, and supporting MIMO. Was used two Ettus USRPs running the IEEE 802.15.4 protocol act as the communicating nodes: one transmitter and one receiver. A third USRP is used to generate Gaussian noise in the same operational frequency of the other two USRPs.

The USRPs run GNU Radio modules. We implemented modules to obtain the input metrics (RSSI and SNR) and calculate the LQE, as well as modifications on the protocol stack as follows. We employ an enhanced version of the IEEE 802.15.4 protocol developed by [19]. The original implementation does not provide a functional MAC layer, i.e. there is no carrier sensing nor retransmissions. We implemented carrier sensing as well as the Binary Exponential Back-off algorithm for retransmissions. The transmission power can be changed dynamically. This value is calculated in a separate module, which implements the LQE.

Since the evaluation scenario considers a M2M communication that requires reliable and low power communication, the following metrics are evaluated:

Packet delivery rate: measures the quality of the communication perceived by the user. Packet delivery rate also can be defined as ratio of the number of successfully received packets to the number of packets sent.

Transmission gain index: is the amount of energy spent per transmission (a higher gain will require a higher transmission power, increasing the power usage). The transmission gain index (TGI) is the value in dBm used by the attenuator in the USRP. In USRP B210, the gain index varies of 0 to 89 dB, and in model B100, the gain index ranges from 0 to 35 dB.

In the experiments, PRR^2 employs a $shortPRR$ window of 40 observations, while the $longPRR$ is computed over 80 observations. Those values are defined empirically. PRR^2 is compared against KLE, an LQE that employs Kalman filters over the RSSI [7]. In addition, we implemented a simple RSSI-based LQE, which averages a number of past RSSI readings. The RSSI estimator's parameters were empirically calibrated to provide their best results. The results shown in this section are an average of five independent runs of each configuration, with the transmission of 80 packets of 18 bytes in each run, in addition to retransmissions. The packets are generated periodically (every 50 ms). Results are plotted with a confidence interval of 95%. All the implemented code, including PRR^2 and the modifications on the physical and MAC layers of GNU radio are publicly available[1].

V. Results

A. First scenario

To identify how each part of PRR^2 contributes to the final performance, we broke down PRR^2 into 5 smaller estimators: (1) RSSI with Exponential Moving Average (RSSI EMA); (2) PRR with an observation window of size 40 (PRR 40); (3) PRR with an observation window of size 80 (PRR 80); (4) PRR combined with RSSI estimation; (5) and PRR with both long and short windows, but without RSSI estimation.

Fig. 3 presents the delivery rates for the above mentioned comparison. The smallest contribution is from the RSSI,

[1]Address of the GIT repository: https://bitbucket.org/wendley/gr-lqe.

while the highest contribution comes from the PRR alone. This is due to the fact that PRR is a richer metric, since it measures the link quality in both directions due to its reliance on ACK packets. Next, merging the estimation of two layers (PRR+RSSI) slightly increases the performance. Finally, PRR^2, which adds two PRR windows, increases the delivery rate by 3% when compared to an estimator that combines the two layers (RSSI and PRR). Thus, the use of two PRR windows smooths out variations due to momentary link quality variations. We emphasize that the confidence intervals (CI) of PRR^2 and PRR + RSSI do not overlap, as described in Table I.

TABLE I
AVERAGE AND CONFIDENCE INTERVALS OF DELIVERY RATE

	RSSI	**PRR w 40**	**PRR w 80**	**PRR + RSSI**	**2 PRR**	**PRR^2**
CI+	18,30%	52,35%	52,20%	53,99%	54,23%	57,98%
Average	13,50%	51,75%	51,00%	53,50%	53,25%	56,25%
CI-	8,70%	51,15%	49,80%	53,01%	52,27%	54,52%

There is also a reduction in the amount of retransmission per received packet when compared to RSSI estimators, as shown in Fig. 4. A smaller number of retransmissions is also important for a reliable network, since it reduces the packet delivery delays. Further, more precise estimators also reduced the confidence intervals, reducing the jitter, which is important for real-time control applications.

The performance improvements of PRR^2 are achieved via an increase of the mean TGI, as shown in Fig. 5. The reduced confidence intervals of PRR^2 when compared to the other configurations also show that the use of two PRR windows reduces the variability in the transmission power, which is also important for a stable link behavior.

B. Second scenario

This scenario compares PRR^2 against one of the LQEs in the literature, called KLE. It is broken down into two configurations. The **first configuration** varies the amount of traffic on the network. The interval between message transmissions is ranged from 50 ms up to 150 ms, and the artificial noise is kept constant with gain index = 66 dB.

As observed in Fig. 6, PRR^2 improves the delivery rate by 5.2% in the worst case (generation interval of 50 ms), and 10.3% in the best case (150 ms). Further, the lower confidence intervals in the PRR^2 indicate and increased stability in the link when compared to KLE. Further, on the best case scenario PRR^2 reduces the amount of transmissions per successfully received packet by 63.1%, on average, as shown in Fig. 7. Despite the improvements in both the delivery rate and in the number of retransmissions, PRR^2 presented a slight increase in the transmission power (0.07% on average) as in Fig. 8.

The **second configuration** evaluates the effect of the amount of background noise, in order to evaluate how each LQE performs with different levels of disturbances. In this configuration, the interval between messages is fixed in 150 ms, while the artificial noise power gain varies between 74 and 80 dB. Fig. 9 shows PRR^2 providing, in the worst case (noise gain index = 77 dB), an improved packet delivery rate of 23.7% on average, while in the best case the improvement achieves 62.1% when compared to KLE. The performance gap and the variance increase for KLE with the level of noise, indicating that KLE does not handle noise as well as PRR^2. The proposed LQE achieves a reduction of 54.7% in the average amount of retransmissions per received packet when compared to KLE, as shown in Fig. 10. Further, the reduced confidence interval with regards to KLE indicates that PRR^2 performs in a more predictable way.

Depending on the noise level, the TGI of PRR^2 in comparison to KLE is similar or slightly higher, while providing significant improvements in packet delivery rate. With gain index 74 and 80 dB, the transmitter gains are statistically equal. The situation of noise source using transmission power gain of 80 dB shows the PRR^2 employing 13.7% more power transmission than KLE to increase in 62.1% the packet delivery rate (Fig. 9). Although the transmission index is 62.1% larger for PRR^2 for a noise gain of 80, it is worth noticing that the increase of energy consumption for the entire experiment will be smaller than 62.1%. This difference should be smaller because less packets will be transmitted by PRR^2, as seen in Fig. 10, reducing the amount of retransmissions and hence the energy consumed.

C. Discussion

PRR^2 has been evaluated in two scenarios. In both, PRR^2 outperforms the delivery rate of KLE, with less retransmissions and with a minimum increase in the transmission power. Some situations, as the one presented in Fig. 11 (G=80), highlight the importance of a metric based in packet counts, such as PRR^2: since the noise source is situated closer to receiver USRP than the sender, the RSSI obtained by the sender does not reflect well the noise in the receiver.

VI. CONCLUSION

Machine-to-Machine (M2M) communication is gaining momentum due to the increased deployment of smart objects. This paper presented PRR^2, a multilayer, fast and simple Link Quality Estimator designed to overcome the limitations of M2M communication in terms of reliability. PRR^2 combines RSSI and two PRR to improve the reliability of the existing link quality estimators. PRR^2 handles quick oscillations in the background noise by using two observation windows for PRR. The use of two windows allow PRR^2 to capture variations in the short term, while maintaining long-term link quality information. Results from an experimental evaluation showed that PRR^2 provides better delivery rate with a lower number of retransmissions than the state of the art. Also, the proposed LQE is more stable than the other evaluated estimators, while consuming increasing slightly the output power of the transmitted packets. As future work, we will propose other LQEs that incorporate more input metrics and we intend to investigate other approaches, such as machine learning and hidden Markov chains.

REFERENCES

[1] Cisco, "Global mobile data traffic forecast update, 2015-2020." http://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html, 2016.

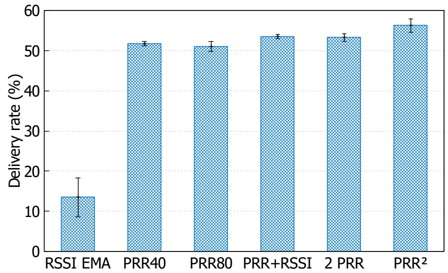

Fig. 3. Delivery rate

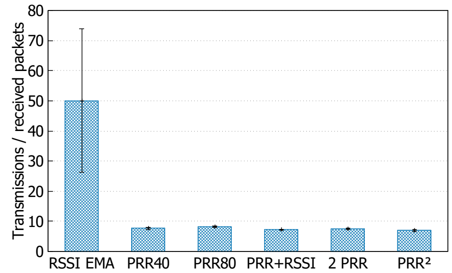

Fig. 4. Transmissions per received packets

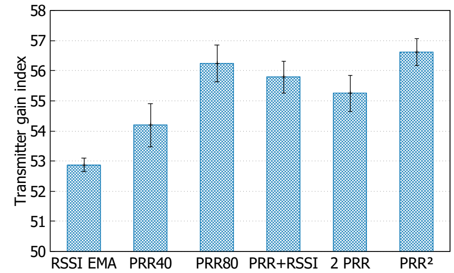

Fig. 5. Transmitter gain index

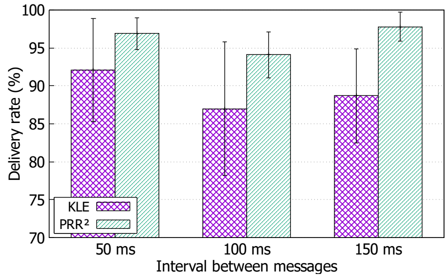

Fig. 6. Delivery rate on 2nd scenario, 1st configuration

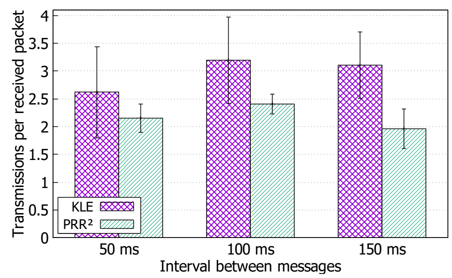

Fig. 7. Amount of retransmissions per received packets on 2nd scenario, 1st configuration

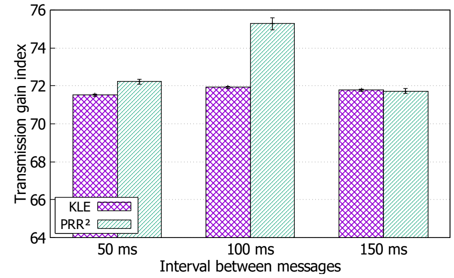

Fig. 8. Gain index Tx on 2nd scenario, 1st configuration

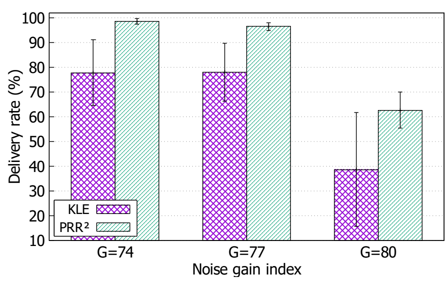

Fig. 9. Delivery rate on second scenario on 2nd scenario, 2nd configuration

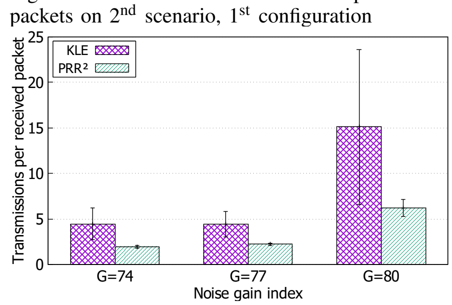

Fig. 10. Amount of retransmissions per received packets on 2nd scenario, 2nd configuration

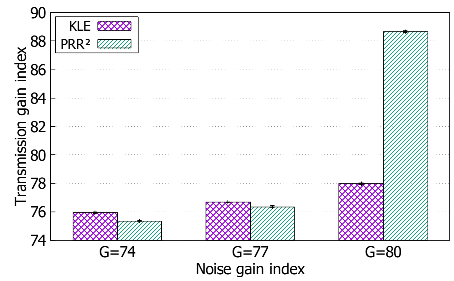

Fig. 11. Gain index Tx on 2nd scenario, 2nd configuration

[2] C. Anton-Haro and M. Dohler, *Machine-to-Machine (M2M) Communications: Architecture, Performance and Applications*. Elsevier, 2014.

[3] A. Alexiou and A. Gotsis, "10 - packet scheduling strategies for machine-to-machine (m2m) communications over long-term evolution (lte) cellular networks," in *Machine-to-machine (M2M) Communications* (C. Antón-Haro and M. Dohler, eds.), pp. 173 – 186, Oxford: Woodhead Publishing, 2015.

[4] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo, "Large-scale measurements of wireless network behavior," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, (New York, NY, USA), pp. 153–165, ACM, 2015.

[5] A. Woo and D. E. Culler, *Evaluation of efficient link reliability estimators for low-power wireless networks*. Computer Science Division, University of California, 2003.

[6] D. S. Couto, D. Aguayo, R. Morris, and J. Bicket, "A high-throughput path metric for multi-hop wireless routing," 2003.

[7] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, "A kalman filter based link quality estimation scheme for wireless sensor networks," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pp. 875–880, IEEE, 2007.

[8] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation.," in *HotNets*, 2007.

[9] C. A. Boano, M. A. Zúñiga, T. Voigt, A. Willig, and K. Romer, "The triangle metric: Fast link quality estimation for mobile wireless sensor networks," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1–7, IEEE, 2010.

[10] M. Caleffi and L. Paura, "Bio-inspired link quality estimation for wireless mesh networks," in *World of Wireless, Mobile and Multimedia Networks & Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, pp. 1–6, IEEE, 2009.

[11] A. S. Cacciapuoti, M. Caleffi, L. Paura, and M. A. Rahman, "Link quality estimators for multi-hop mesh network," in *Euro Med Telco Conference (EMTC), 2014*, pp. 1–6, IEEE, 2014.

[12] N. Baccour, A. Koubâa, H. Youssef, M. B. Jamâa, D. Do Rosario, M. Alves, and L. B. Becker, "F-lqe: A fuzzy link quality estimator for wireless sensor networks," in *Wireless Sensor Networks*, pp. 240–255, Springer, 2010.

[13] N. Baccour, A. Koubâa, H. Youssef, and M. Alves, "Reliable link quality estimation in low-power wireless networks and its impact on tree-routing," *Ad Hoc Networks*, vol. 27, pp. 1–25, 2015.

[14] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 364–369, IEEE, 2005.

[15] T. Liu and A. E. Cerpa, "Data-driven link quality prediction using link features," *ACM Transactions on Sensor Networks*, vol. 10, no. 2, pp. 1–35, 2014.

[16] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 221–262, 2006.

[17] S. Nahmias and Y. Cheng, *Production and operations analysis*, vol. 6. McGraw-Hill New York, 2009.

[18] E. Research, "Ettus research product detail - b210." http://www.ettus.com/product/details/UB210-KIT, 2016. Viewed in Apr 2016.

[19] B. Bloessl, C. Leitner, F. Dressler, and C. Sommer, "A gnu radio-based ieee 802.15. 4 testbed," *12. GI/ITG FACHGESPRÄCH SENSORNETZE*, p. 37, 2013.