Efficient Online Burst Transmission Scheme for Wireless Sensor Networks

Zeeshan Ansar, Jianjun Wen, and Waltenegus Dargie

Faculty of Computer Science
Technical University of Dresden
01062 Dresden, Germany
Email:{ zeeshan.ansar, jianjun.wen, waltenegus.dargie}@tu-dresden.de

***Abstract*—Wireless sensor networks supporting aggressive sampling in harsh environments (to deal with high packet loss or to provide reliable data from sensors such as 3D accelerometers and 3D gyroscopes) require transmission schemes which can achieve a relatively high throughput. Existing contention-based, low-power listening MAC protocols are not apt for these types of networks because their channel utilisation is considerably low. In this paper we propose a hybrid burst transmission scheme to achieve high throughput between static relay nodes, such as nodes deployed on a civil infrastructure (bridge or building). Our transmission scheme deals with link quality fluctuations and adaptively adjust the number of packets that can be transmitted in burst. Its essential features are relying on statistics that are obtained (1) offline and reflect the long-term characteristic of a link and (2) online and reflect the short-term link quality fluctuation. We experimentally compared our transmission scheme with two proposed state-of-the-art schemes in terms of throughput, transmission delay, packet loss, and energy consumption. We implemented all transmission schemes and integrated them into the TinyOS environment and the TelosB platform.**

***Index Terms*—Burst transmission; link quality estimation; link quality fluctuations; wireless link; intermediate links**

I. Introduction

Wireless sensor networks supporting aggressive sampling to deal with noise, interference, or harsh environmental conditions are crucial for many applications, such as monitoring the integrity of buildings and bridges. In the case of bridges, for example, the movement of structural elements (suspension cables, deck, towers) as well as cars can interfere with radio transmission and produces high packet loss [1]. Similarly, sensor nodes deployed on buildings may experience harsh interference and noise coming from femtocell and WiFi transmitters as well as from other sources. At the same time such networks are highly constrained by exhaustible batteries which may not easily be recharged or replaced. Consequently, providing robust and energy-efficient packet transmission schemes is crucial for deploying reliable wireless sensor networks.

One of the most formidable challenges in harsh environments is dealing with link quality fluctuations. Often statistics pertaining to link quality fluctuation is required to determine when and for how long nodes should transmit or refrain from transmission. As a result, link quality fluctuation is an active research area in the context of IEEE 802.15.4 standard [2], [3], [4] and [5]. Moreover, several link quality assessment metrics have been proposed to predict the success probability of transmitting packets in bursty links, among which are Packet Reception Rate (PRR) [6], [7], Acknowledgement Reception Rate (ARR) [8], Expected Transmission Count (ETX) [9], [10], [11] and Conditional Packet Delivery Function (CPDF) [12], [13].

One of the limitations of these metrics is that, regardless of the cost and complexity of obtaining them, they have been so far employed to determine the success rate of transmitting a single future packet. The justification is that since link quality fluctuates considerably, the correlation between the past n packets and the future $\{m : m >> 1\}$ packets is typically weak and, hence, relying on past statistics to transmit a large amount of packets would result in poor performance. This assertion, however, does not take long-term correlation into consideration and is valid only to short-term fluctuations. In certain deployments (where we have, for example, an oscillating bridge or a predictable traffic pattern) the long-term link quality fluctuation can be regarded as stationary in a statistical sense, in which case, it can be useful for predicting state transitions, where each state may represent a unique short-term fluctuation.

In this paper we propose a burst transmission scheme for wireless sensor networks by taking both long-term and sort-term link quality fluctuations into consideration. The statistics corresponding to these fluctuations are obtained in two phases: offline and online phases. During the offline phase, we model link quality fluctuation with a two-stage Markov process. The model (a) classifies link quality into different states and determines the transition probabilities between the states; (b) estimates the average number of packets that can be transmitted in each states and (c) the expected duration of a link staying in a particular state. During the online phase, the transmission scheme takes the short-term statistics of received acknowledgement packets to predict the most probable future state and the associated burst size. As a summary, the contributions of the paper are the following:

1) Using data collected from our sensor network, we studied the temporal characteristic of channel state variations.
2) Using discrete Markov models, we established a relationship between a state and the expected number of

packets that can be transmitted in burst.

3) We implemented and integrated our transmission scheme into the TinyOS operating system for the TelosB platform.
4) We quantitatively compared the throughput, latency, packet loss, and energy consumption of our model with two proposed approaches, namely, (a) the β-factor [13], which is a burst transmission scheme developed at Stanford University; and (c) an online adaptive burst-size estimation technique [14], which is developed at RWTH Aachen University (Germany).

The rest of the paper is organized as follows: In Section II, we review work on bursty links and transmission schemes for bursty links. In Section III, we introduce our approach and the three steps required to develop our transmission scheme. In Section IV, we define the system architecture of our online transmission scheme and in Section V introduce the online algorithm. In Section VI, we present the implementation and quantitative evaluation of our transmission scheme and compare its performance with existing state-of-the-art approaches. Finally, in Section VII, we provide concluding remarks and future work.

II. Related Work

Link quality estimation is a critical aspect of efficient and reliable packet delivery in wireless networks in general and in wireless sensor networks in particular. Experimental results suggest that the quality of wireless links fluctuates and this fluctuation can be broadly categorised into three regions: good, intermediate, and poor. The burstiness of wireless links is a well established fact and has been closely investigated by recent studies [13], [15], [16]. The effort of dealing or copping with link burstiness can be divided into three main categories; offline, online, or hybrid approaches.

A. Offline approach: Long-term characteristics

Srinivasan et al. [13] propose the β metric to measure the burstiness of a wireless links. β is calculated by using a conditional probability packet delivery function (CPDF), which determines the probability of successfully delivering the next packet after n previous packets have been successfully delivered or failed. A value of $\beta = 1$ and $\beta = 0$ represents a perfectly correlated link and an uncorrelated link, respectively. The key finding of the work is that the percentage of intermediate links for inter packet intervals greater than 500 ms is almost the same. This suggests that halting a transmission for 500 ms after a packet transmission failure can improve the packet reception ratio. However, β cannot determine the length of reliable and unreliable transmission periods, which are important to efficiently schedule packet transmission. Furthermore, β does not handle short-term link quality fluctuations because it considers all types of failures as similar. Furthermore, as Alizai et al. [14] observe, β is not suitable for online estimation as it requires a large amount of data to achieve a 95% confidence interval.

Munir et al. [17] define link burstiness as a period of continuous packet loss and propose a scheduling algorithm which produces latency bound for real-time periodic streaming of a large amount of packets. The authors introduce B_{max} and B_{min} to characterise the maximum number of consecutive packets loss and the minimum number of consecutive success, respectively. To calculate these metrics, they performed an empirical study for 21 days and collected traces of packet successes and failures for different links. An offline packet transmission schedule then computes transmission and intermission periods based on B_{max} and B_{min}. The authors observe that the most frequently observed consecutive success is $B_{min} = 1$ which means the transmission scheme should transmit a single packet followed by an intermission period which corresponds to B_{max}. This, however, does not correctly reflect the condition of most real links where good and stable links can be observed.

Wen et al. [15] propose an offline transmission scheme that uses the conditional probability distribution function of SNR fluctuations to estimate the expected reliable and unreliable transmission periods. Their approach employs an SNR threshold above which a link is considered to be good and stable. However, empirical studies reveal that the SNR varies between 3dB to 21dB in most intermediate links in which case the approach of Wen et al. potentially results in under- and over-estimated burst periods. Furthermore, the expected burst-size is fixed, as the proposed scheme regards only the long-term link quality fluctuation.

Recently Ansar et al. [16] propose a two stage Markov model to characterise link quality fluctuation. Hence, link quality is categorised into k states and the transition probability between the states is computed by transmitting a large amount of packets in burst. The proposed scheme attempts to address three questions pertaining to link quality fluctuation: 1) Given that a link is in a known state (for example, in a good, intermediate, or bad), how long will it stay in that state? 2) Given the link is in a known state, what will be the most probable next state? 3) Given the link quality can be in one of the k states, what is the optimal number of packets that can be transmitted in burst in that state? The limitation of the approach is that it poorly reacts to short-term link quality fluctuation, as the proposed scheme is optimised for long term link quality fluctuation. Moreover, there is no real-time feedback mechanism to correct wrong state transitions.

B. Online approach: Short-term link quality

Alizai et al. [14] combine two different metrics to characterise link quality fluctuation and to determine the number of packets that can be transmitted in burst. These are MAC_3 and the expected future transmission (EFT). MAC_3 is calculated by taking the moving average of incoming acknowledgement packets (the window for the moving average was set to 100). The authors experimentally determined that the probability of successfully transmitting a future packet increased to 80% given that the past 3 consecutive packets were successfully transmitted. The probability does not increase appreciably if the number of successfully transmitted past packets increases. So, MAC_3 calculates the probability of successfully transmitting n number of packets in future given that 3 consecutive packets were successfully transmitted in the past. Then EFT averages the number of packets that can be successfully transmitted in future given 3 past consecutive packets were transmitted successfully. This is set as the burst size. While the approach of Alizai et al. is interesting, it has two drawbacks. Firstly, the moving average is slow to "perceive" short-term fluctuations and react to them (as we shall show this experimentally). Secondly, whereas the number of packets that can be transmitted in burst is determined, it is not clear how long a transmitting node should pause before the next burst transmission begins.

Brown et al. [18] introduce BrustProbe, a mechanism to measure link burstiness (B_{max} and B_{min}) online to address the drawback of the approach proposed by Munir et al. [17]. Hence, the authors embed probing slots in the transmission schedule to probe and estimate link burstiness online and to share this knowledge among neighbour nodes. The probe mechanism is more reactive for capturing burst periods. The limitation of the approach is that it requires extra time slots to measure link burstiness. The proposed technique relies on TDMA scheduling which require tight time synchronization.

C. Hybrid approach

Liu et al.[19] employ a Hidden Markov Model (HMM) to estimate periods of poor channel quality (pushback period k). If a transmission is successful, the next packet will be transmitted immediately; if, however, a transmission fails, then the next transmission is pushed back by k-slots. The limitation of the approach is its difficulty to deal with independent losses as transmission is halted on a single failure (similar to [13]). Secondly, the scheme is computationally expensive as extra time is required to calculate the pushback period for each failure. Thirdly, it is not clear from the paper for how long the channel should be observed or how much statistics should be gathered from the link to compute the probability of success or failure.

In this paper, we propose a hybrid approach that takes advantage of both offline and online models. Our aim is to characterise the long-term link quality fluctuation with statistics that are obtained offline and to employ the statistics of received acknowledgement packets to deal with short-term link quality fluctuations. Our approach uses burst transmission to gather sufficient statistics and to use this statistics for determining when and for how long nodes should transmit packets in burst. It also determines the expected duration nodes should abstain from transmitting packets when link quality is bad. The online statistics are used to fine-tune and calibrate the offline model.

III. Offline Link Quality Fluctuation Model

In Section II, we mentioned that the research community classifies the quality of a link as good ($ARR \approx 1$), intermediate ($0.9 \leq ARR \leq 0.1$), or bad ($ARR < 0.1$) [20], [13] (we shall introduce ARR shortly). Of these, the success of packet transmission is the least certain in the intermediate state (in the good state packet delivery rate is high whilst in the bad state it is low). Thus, an efficient packet transmission scheme should (1) recognise in which of these states a link is likely to be found in the immediate future and (2) how many packets should on average be transmitted in burst in each of the states.

Commercially available transceivers make link quality indicator metrics available to higher layer services including received signal strength indicator (RSSI), link quality indicator (LQI), and background noise level. Nevertheless, it is not possible to establish a deterministic relationship between these metrics and the success of packet delivery. Packets can be successfully transmitted with a certain probability even when the metrics indicate that the link is not good and can be lost even when they indicate that it is good. Srinivasan et al. [8] introduce a metric called Acknowledgement Reception Ratio (ARR) to summarise the relationship between successful packet delivery and signal-to-noise ratio (SNR).

ARR is computed as follows: First, a sequence of packets (for example 1000 packets) are divided into a set of subsequence (10 packets per subsequence). Each subsequence is transmitted in succession and each packet in a subsequence is acknowledged when it is successfully received (say 7 packets are acknowledged). Then for that subsequence, the ARR is the ratio of the number of successfully received acknowledgement packets to the total number of transmitted packets ($7/10 = 0.7$). The SNR of that subsequence is the average SNR of the successfully received ACK packets. Likewise, all the set of subsequence (for our example which equals ($1000/10 = 100$) is transmitted, the ARR is produced for each subsequence, and the corresponding SNR is computed. Then a 2-dimensional graph of ARR vs. SNR is plotted to summarise the relationship between the two quantities. The merit of this approach is that the quality of a link can be evaluated independent of the distance of separation between the transmitter and the receiver and physical layer parameters such as the transmission power and the specific channel allocated. The weakness of the approach is that for a short duration, the channel's characteristic is assumed to be both symmetrical and correlated to account for the SNR of lost packets. We adopt this approach to evaluate the effect of link quality fluctuation on successful packet delivery.

Figure 1 displays the relationship between ARR and SNR for one of the links we established for our experiment. As can be seen in the figure, packets are successfully delivered with higher probabilities when the SNR is high (for SNR greater than 7 dB, the probability approaches unity), however, one can also observe the existence of successful packet delivery even when the SNR is below 1 dB.

A. State Transition Probabilities

The second step in characterising link quality fluctuation is determining the probability of transitions between link states (bad, intermediate, and good) which signifies link quality fluctuations during a continuous transmission of packets. The

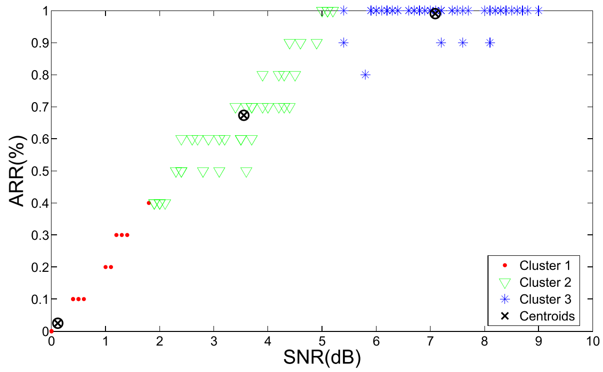

Fig. 1: A summary of relationship between the SNR and ARR of a wireless link. SNR is computed as the difference between RSSI and background noise power. A k-mean clustering is used to cluster the link into one of the three link states.

subsequence of 10 packets we used above to compute ARR can also be used to characterise a single state. Depending on the number of acknowledged packets after the transmission of the 10 packets and the average SNR, the corresponding state can be classified as bad, intermediate, or good. After the transmission of 1000 packets, we have a sequence of 100 states. We describe this sequence of states as a first order Markov chain [21]. Hence, the fluctuation in link quality is described by a state transition probability, which is computed as follows:

$$a_{ij} = P\left(S_j|S_i\right) = \frac{N_{i\rightarrow j}}{\sum_{m=1}^{M} N_{i\rightarrow m}} \tag{1}$$

where a_{ij} is the transition probability from state i (S_i) to state j (S_j), M is the total number of states and $N_{i\rightarrow j}$ is the number of transitions from state i to state j. An interesting aspect of Equation 1 is the possibility of asking (and answering) the following question: Given the channel is in a known state in the beginning of slot τ, what is the probability that it stays in the same state for the next d slots (as expressed by Equation 2). This is an important question because it directly addresses the question of link stability and state duration.

$$o = \left\{ \begin{matrix} S_n, & S_n, & S_n & \ldots\ldots & S_n, & S_m \\ 1 & 2 & 3 & & d & d+1 \end{matrix} \neq S_n \right\} \tag{2}$$

The question can be answered using the following expression:

$$P_n\left(d\right) = \left(a_{nn}\right)^d \left(1 - a_{nn}\right) \tag{3}$$

Where a_{nn} is the probability that the link quality is in state n and remains in the same state in the next round of transmission. Note that the plot of $P_n(d)$ for all d gives the probability mass function for state n, from which it is possible to determine the expected state duration (ESD) in which the link quality stays in state n:

$$\bar{d_n} = \sum_{d=1}^{\infty} dP_n\left(d\right) = \frac{1}{1 - a_{nn}} \tag{4}$$

where, d is the duration required to transmit 10 packets in burst and $\bar{d_n}$ is ESD expressed in d.

B. *Burst Size in a State*

After the ESD is determined, the next step is determining the expected number of packets that can be transmitted in burst in each state. The goal is to minimize the number of lost packets. Once again we employ a first order Markov chain for this step but this time we fix the number of states to two, success (1) and failure (0). The sequence of received acknowledgement packets during a test phase is used to determine the state transition probabilities. Consider Figure 2 in which after 10 packets are transmitted in burst, the sequence of acknowledgement packets is given. From the sequence of acknowledgement packets, it can be seen that there are altogether 9 transitions: once from 0 to 0, twice from 0 to 1, twice from 1 to 0 and four times from 1 to 1. Hence the state transitions probabilities, in respective order, are: $b_{00} = \frac{1}{9}, b_{01} = \frac{2}{9}, b_{10} = \frac{2}{9}, b_{11} = \frac{4}{9}$. Once the state transition probabilities are determined, the expected number of burst size can be calculated by applying equation 4. It must be noted, however, that the sequence obtained in figure 2 is not sufficient to produce reliable statistics. In reality, repeated experiments are conducted to obtain the state transition probabilities.

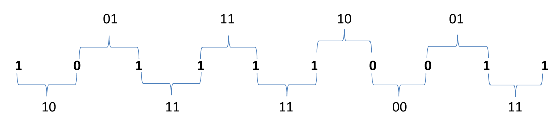

Fig. 2: Sequence of acknowledgement packets signifying successful or failed packets transmission and the determination of state transition probabilities.

IV. Online Link Fluctuation Model

The offline model reflects the long term characteristic of a link. Since state transition is a probabilistic phenomenon, the model may make wrong transitions or may fail to "perceive" short term proper transitions. The cumulative effect of both cases may lead to a link fluctuation perceived by the model which does not reflect the reality. To highlight this point, consider Figure 3. Suppose at time τ, the offline model accurately estimates that the channel is in state 1 (S_1) and according to Equation 4, the link quality should remain in the same state for 8 time slots. However, Equation 4 estimates the expected duration and the actual state duration may be different from the one estimated by the model. In the figure, for instance, the link quality changes to state 2 (S_2) at time $\tau + 2$ and to state 3 (S_3) at $\tau + 5$. In order to deal with these types of short-term transitions, we propose an online model the main purpose of which is to fine-tune or calibrate the off-line model.

Figure 4 shows the components of the online model, which are, a look-up table containing the state transition matrix generated by the offline model, a channel state estimator,

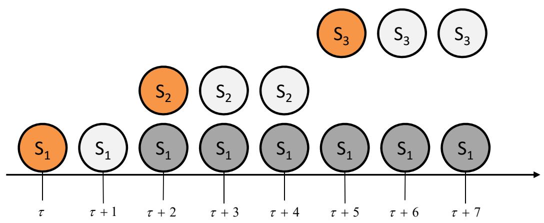

Fig. 3: A short term state transition that may not be "perceived" by the offline model. The orange states indicate proper transitions, the light-grey states indicate actual link quality states, and the dark-grey states indicate the estimated link quality state by the off-line model.

which evaluates the link quality metric of incoming acknowledgement packets and determines to which state the current link quality belongs, and a predictor, which estimates the next state of a link and the number of packets that should be transmitted in burst.

One of the challenges of relying on an online link quality estimation mechanism is the difficulty of gathering sufficient statistics. This is particularly the case for bad and intermediate states, in which the number of successfully delivered packets is few. Consequently, we defined a metric called *conditional probability of expected state duration* (CPESD)[1] to simplify our online estimation. The idea is as follows: suppose the expected state duration for a bad state is 8. In other words, once the link quality transits to a bad state, it stays there for the next 8 state durations (recall that a single state duration equals the time required to transmit 10 packets). The CPESD expresses the probability that the link quality stays as predicted by the model given that the past n states were as estimated by the model. Figure 5 displays the CPESD of a bad state for different links we established outdoors – 70% of the time, the link stays as predicted by the offline model if it stays in the same state at least for three consecutive state durations.

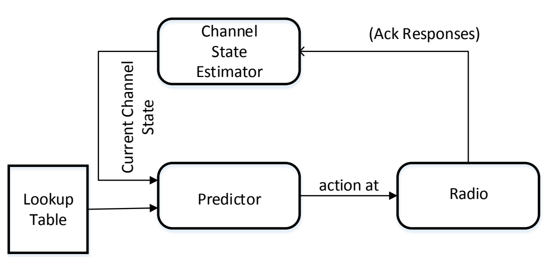

Fig. 4: The system architecture of the online link quality estimator.

V. Algorithm

In order to illustrate how the online transmission scheme functions, we fill the look-up table (Table I) with data obtained

[1]The essential notion is first proposed in [22] and adopted in [13] and [14].

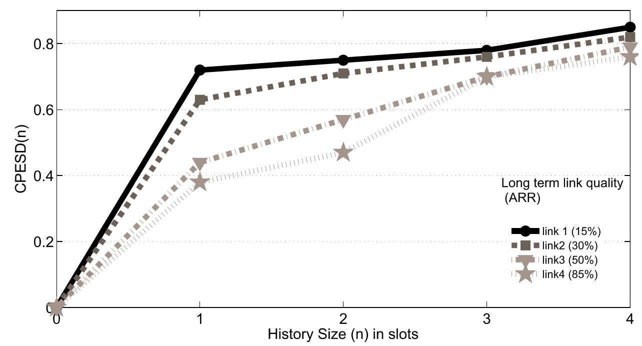

Fig. 5: Measuring the prediction accuracy of a bad state from a short history.

Link state	Burst size	ESD
Good	10	6
Intermediate	6	4
Bad	2	8

TABLE I: Expected state duration and burst size of different states.

from the offline model for one of our links. Initially (in the time slot τ), the burst transmission strategy transmits 10 packets in succession (the maximum number of packets that can be transmitted in a single state). Based on the number of packets which are successfully received and the average SNR of the received acknowledgement packets, the channel state estimator determines the current link quality state. If the current link quality state is *good*, then the link quality remains in the same state for the next 6 consecutive time slots and the online transmission scheme transmits 10 packets in burst in each subsequent time slot.

Suppose, however, after the initial burst transmission, only 2 packets are successfully delivered. Apparently, the link quality is in the *bad* state. The offline strategy would have transmitted for the next 8 state durations only 2 packets per state duration (because the expected state duration for the bad link is 8, according to Table I). However, we would not have sufficient statistics to determine the short-term link quality fluctuation during this time. Therefore, the online transmission scheme considers the link quality of slot $\tau+1$ as an intermediate state and sends 6 packets in burst (this is indicated in Fig. 6). If all the 6 packets are delivered successfully, the transmission

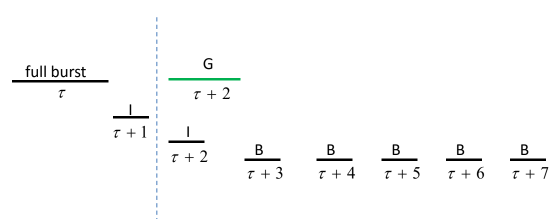

Fig. 6: Online burst transmission strategy. G: good state, I: intermediate state, B: bad state.

scheme considers the link quality as *good* and sends 10 packets in burst in the next time slot ($\tau + 2$) (indicated by the green line in Fig. 6). If, however, 2 or less packets are successfully delivered like in slot $\tau + 1$, the link quality is still in a *bad* state, but the transmission scheme transmits 6 packets in burst to gather enough statistics. If the SNR of the received acknowledgement packets still indicates the state is in a *bad* link in the time slot $\tau + 2$, then according to the CPESD, the link has been in the *bad* state for the past 3 state durations and will remain *bad* for the average duration determined by the offline model (the next 5 states for our example). Hence, it transmits only 2 successive packets in the next 5 time slots. After $\tau + 7$, the online scheme determines the next state based on the acknowledged packets for state $\tau + 7$ and repeat the same procedure.

A summary of our online algorithm is given in Algorithm 1 where O_τ refers to the link quality state for the time slot τ; $CPESD_{th}$ refers to the minimum number of state durations which result in $CPESD \geq 0.7$; (for our case, $CPESD_{th} = 3$); $P_{\tau+1}$ is the predicted state for the time slot $\tau + 1$, and S_i is the link quality states defined as *good*, *intermediate*, and *bad*.

Algorithm 1 Transmission Technique

```
Input:O_τ, CPESD_th
Output:P_τ+1
initialization:S_0 = Good, S_1 = Average, S_2 = Bad
if O_τ ≠ S_0 then
    if O_τ == S_i  where,  i≠0 then
        if O_τ == O_τ-1 then
            Count++
        else
            Count=0
        end
        if Count < CPESD_th then
            P_τ+1 ← S_i-1
        else
            Stay for the duration of (ESD − CPESD_th) in
            state S_i
        end
    end
else
    P_τ+1 ← O_τ
end
```

VI. Implementation and Evaluation

We experimentally compared the performance of our burst transmission scheme (we label it as O-DMB) with two proposed approaches. One of them is the β-factor (labelled simply as β), developed at Stanford university by Srinivasan et al. [13] and the other is the Bursty Link Estimator (labelled as MAC_3), developed at RWTH Aachen University by Alizai et al. [14]. We implemented all algorithms and integrated them into the TinyOS environment and the TelosB platform. Then we deployed a wireless sensor network consisting of 14 nodes in an outdoor environment. The 14 nodes were placed randomly; thus, the minimum and the maximum distances between the nodes were 10 and 45 m. The wireless channel we used for communication was channel 26 with different RF transmission power levels ranging from the minimum to the maximum (i.e., levels 1 to 31). We established 5 different links for communication.

TABLE II: A summary of physical parameters used to establish the links.

	link1	link2	link3	link4	link5
d (m)	35	15	15	10	22
P (dBm)	0	-10	0	-10	-3
IPI (ms)	25	100	20	25	50

For our case (to establish the offline statistics), we transmitted 2,000 packets in each link with an Inter Packet Interval (IPI) of 20 ms and gathered from the received acknowledgement packets RSSI, LQI, background noise, and timestamps and determined (1) the number of distinct link quality states for each link using a k-means clustering algorithm; (2) the link quality state transition probabilities, (3) expected stable duration for each state and (4) expected number of packets that can be transmitted in burst for each state. These link quality metrics are then entered in a lookup table and flashed to the nodes to be referred to during online adaptation.

The β factor divides time into 500 ms slots and transmits packets in burst within these slots. The number of packets that can be transmitted in a single slot depends on the IPI. So with IPI = 20, 25, 50, and 100 (ms), a maximum number of 50, 40, 20 and 10 packets could be transmitted in burst, in respective order. When packet transmission fails (i.e, no acknowledgement packet is received), β halts transmission for 500 ms and then resumes with the burst transmission until the next failure occurs. BLE first transmits 100 packets in burst and from the history of the acknowledgement packets, it determines the size of the next burst transmission. After each transmission period, a new acknowledgement sequence is added to the link history.

We evaluated the performance of all the three transmission schemes for a single hop link using the following metrics: (1) **Throughput:** Number of packets successfully acknowledged per second. (2) **Transmission time:** How much time is required to successfully transmit 'n' packets. (3) **Packet loss**: The number of lost packets after transmitting 'n' number of packets, and (4) **Energy consumption:** The energy consumed by transmitting nodes to deliver 'n' packets successfully. Table II summarises the transmission parameters of each link.

A. Throughput

Throughput is an important evaluation metric in wireless sensor network, particularly for aggregating nodes which are closer to a base station. It refers to the speed with which a node successfully delivers packets to its neighbour. In other words, throughput refers to the number of successfully acknowledged packets per unit time. We transmitted 20,000 packets with each link by considering different IPI, as described in Table II. Figure 7 compares the throughput of our scheme with the

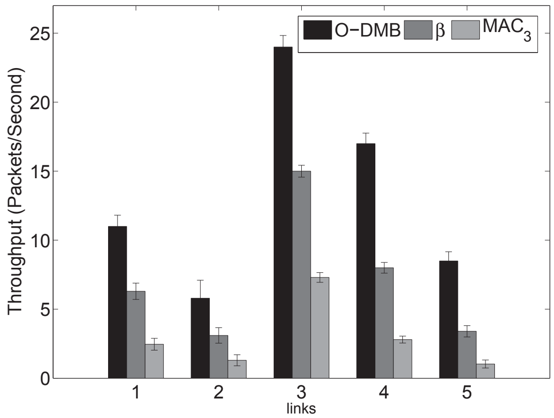

Fig. 7: Comparison of the throughput of the three burst transmission schemes for different wireless links.

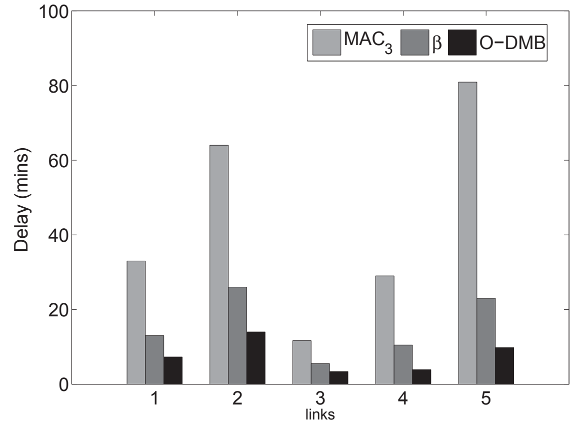

Fig. 8: Comparison of the time required to transmit 500 packets successfully. Lost packets were retransmitted until all the 5000 packets were successfully transmitted.

two online transmission schemes. Each bar graph represents the average throughput of 10 repeated experiments. In all the cases O-DMB has the highest throughput. The reason is that O-DMB deals with short-term link quality fluctuations by reducing the burst size when the link quality deteriorates and resuming transmission with the maximum burst size as soon as the quality of the links improves. On the other hand, β differs transmission for 500 ms as soon as it encounters failure but regards all failures as similar even though the underlying conditions are different. MAC_3 has the longest reactive time for short-term fluctuations since its history size is fixed. Moreover, even for a longer observation period, the expected burst size for MAC_3 is comparatively small (between 6 to 10 packets per burst). Due to this small burst size the history array holds outdated history and, as a result, the future burst size is often wrongly calculated. The throughput of O-DMB is twice higher than β and three times higher than MAC_3 on average.

B. Transmission Time

The transmission time (or delay) is another way of looking at throughput. It refers to the time required to successfully transmit a fixed number of packets. The term "successfully" indicates that lost packets were retransmitted. Figure 8 displays the time required to transmit 5000 packets successfully in different links. In accordance with the results we observed for the throughput, O-DMB performs better than the others in all the links. The transmission time of O-DMB reduced on average by half and three times in comparison to the transmission time of β and MAC_3, respectively.

C. Packet losses

Figure 9 compares the percentage of packet lost during the transmission of 2000 packets in different links. As can be seen, MAC_3 has the highest percentage of packet losses, apparently, due to its slow reaction to link quality fluctuation and difficulty with determining the suitable burst size. β has the lowest packet loss because it halts packet transition as soon as it perceives that a packet has been lost. O-DMB exhibits 4 to 10 percent higher packet losses in comparison to β; the reason is that O-DMB does not halt packet transmission on a single failure; instead, it reduces the burst size for the next slot.

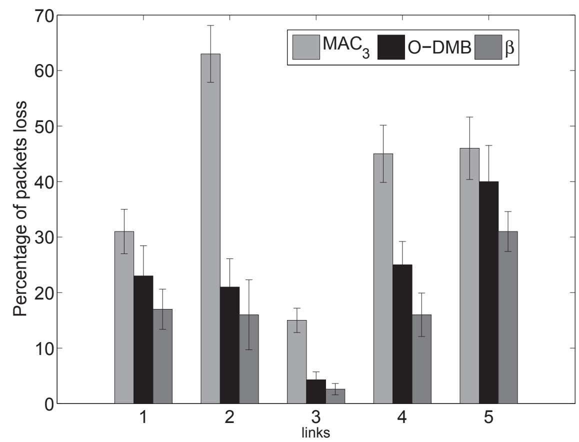

Fig. 9: Comparison of the percentage of packets lost during the transmission of 2000 packets with the three burst transmission schemes.

D. Energy Consumption

In order to measure the amount of energy consumed by transmitting nodes we moved the wireless sensor network indoors and employed Yokogawa digital power analysers (WT210). All the transmission schemes use the same configuration and should deliver 2000 packets successfully (i.e., lost packets were retransmitted). The maximum sampling rate the power analysers could support is 10 samples per second; i.e., a minimum of 100 ms interval existed between samples.

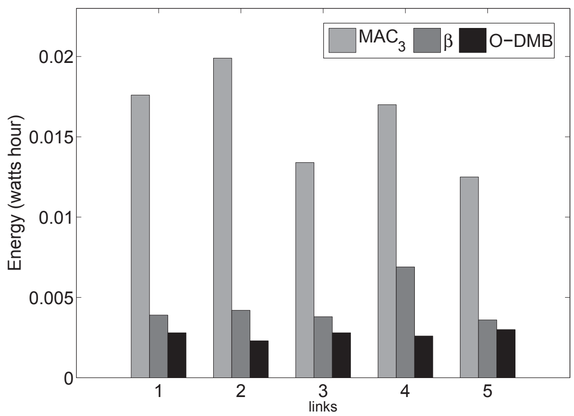

Fig. 10: Comparison of the energy consumption of the three transmission schemes for successfully delivering 2000 packets in an indoor environment (lab setting).

Therefore, in order to match the power sampling frequency with the power consumption of the transmitting nodes, we fixed the inter packet interval to 100 ms. Figure 10 shows the actual energy consumption in watts-hour. The transmission scheme which resulted in the highest amount of energy consumption was MAC_3. The next was β, because it has the longer transmission delay compared to our approach. Ours resulted in the least amount of energy consumption in all the links.

VII. Conclusion

In this paper we motivated burst transmission over an IEEE 802.15.4 wireless link to enable a relatively high throughput. In order to deal with link quality fluctuation and reduce packet loss, we proposed a hybrid approach that combines offline and online models. The offline model is a two-stage Markov model which classifies link quality fluctuations into different link states and associates transition probabilities to these links. Furthermore, using the transition probabilities it estimates the expected duration for each link state and the expected number of packets that can be transmitted in burst in each state. The offline model is optimal to characterise long-term link quality fluctuation. The drawback of the offline model is that it may be unable to deal with short-term fluctuations. The online model, on the other hand, "perceives" short-term link quality fluctuations and attempt to make appropriate transitions. Consequently, the integration of both models into a unified transmission scheme enables to deal with both short- and long-term link quality fluctuations.

We implemented our transmission scheme and two additional state-of-the-art burst transmission schemes (β and MAC_3) and integrated them into the TinyOS environment for the TelosB platform. We deployed a wireless sensor network consisting of 14 TelosB nodes in indoor and outdoor environments and experimentally compared the performance of the three transmission schemes in terms of throughput, bulk transmission delay, packet loss, and energy consumption. Our scheme produced the highest throughput, the shortest transmission delay, and the least amount of energy consumption but β produced the least packet loss while the performance of MAC_3 was the worst in terms of all the metrics we defined. This is because MAC_3 has slow reaction time and difficulty with obtaining the optimal burst size suitable for the current link quality.

References

[1] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Information processing in sensor networks, 2007. IPSN 2007. 6th international symposium on*, pp. 254–263, IEEE, 2007.

[2] T. Liu and A. E. Cerpa, "Temporal adaptive link quality prediction with online learning," *ACM Trans. Sen. Netw.*, vol. 10, pp. 46:1–46:41, May 2014.

[3] N. Baccour, A. Koubâa, H. Youssef, M. Ben Jamâa, D. Rosário, M. Alves, and L. B. Becker, *Wireless Sensor Networks: 7th European Conference, EWSN 2010, Coimbra, Portugal, February 17-19, 2010. Proceedings*, ch. F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks, pp. 240–255. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[4] Q. Dong and W. Dargie, "Analysis of the cost of handover in a mobile wireless sensor network," in *Wired/Wireless Internet Communication*, pp. 315–322, Springer Berlin Heidelberg, 2012.

[5] K. Srinivasan, M. A. Kazandjieva, M. Jain, and P. Levis, "Prr is not enough," 2008.

[6] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, IEEE, 2008.

[7] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 10, pp. 3557–3564, 2010.

[8] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sen. Netw.*, vol. 6, pp. 16:1–16:49, Mar. 2010.

[9] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, (New York, NY, USA), pp. 134–146, ACM, 2003.

[10] X. Mao, S. Tang, X. Xu, X.-Y. Li, and H. Ma, "Energy-efficient opportunistic routing in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 11, pp. 1934–1942, 2011.

[11] S. Guo, L. He, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," *Computers, IEEE Transactions on*, vol. 63, no. 11, pp. 2787–2802, 2014.

[12] L. Mottola, G. P. Picco, M. Ceriotti, Ş. Gun, and A. L. Murphy, "Not all wireless sensor networks are created equal: A comparative study on tunnels," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 2, p. 15, 2010.

[13] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The beta-factor: measuring wireless link burstiness," in *Proceedings of the 6th International Conference on Embedded Networked Sensor Systems, SenSys 2008, Raleigh, NC, USA, November 5-7, 2008*, pp. 29–42, 2008.

[14] M. Alizai, H. Wirtz, G. Kunz, B. Grap, and K. Wehrle, "Efficient online estimation of bursty wireless links," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pp. 191–198, June 2011.

[15] J. Wen, Z. Ansar, and W. Dargie, "A link quality estimation model for energy-efficient wireless sensor networks," in *IEEE,International Conference on Communications, ICC 2015, London, England, June 8-12*, 2015.

[16] Z. Ansar, J. Wen, E. D. Ayele, and W. Dargie, "An efficient burst transmission scheme for wireless sensor networks," in *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '15, (New York, NY, USA), pp. 151–155, ACM, 2015.

[17] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse, “Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks,” in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN ’10, (New York, NY, USA), pp. 303–314, ACM, 2010.

[18] J. Brown, B. McCarthy, U. Roedig, T. Voigt, and C. J. Sreenan, “Burstprobe: Debugging time-critical data delivery in wireless sensor networks,” in *Wireless Sensor Networks - 8th European Conference, EWSN 2011, Bonn, Germany, February 23-25, 2011. Proceedings*, pp. 195–210, 2011.

[19] S. Liu, R. Srivastava, C. E. Koksal, and P. Sinha, “Pushback: A hidden markov model based scheme for energy efficient data transmission in sensor networks,” *Ad Hoc Networks*, vol. 7, no. 5, pp. 973 – 986, 2009.

[20] M. H. Alizai, O. Landsiedel, J. Á. B. Link, S. Götz, and K. Wehrle, “Bursty traffic over bursty links,” in *Proceedings of the 7th International Conference on Embedded Networked Sensor Systems, SenSys 2009, Berkeley, California, USA, November 4-6, 2009*, pp. 71–84, 2009.

[21] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” in *PROCEEDINGS OF THE IEEE*, pp. 257–286, 1989.

[22] H. Lee, A. Cerpa, and P. Levis, “Improving wireless simulation through noise modeling,” in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pp. 21–30, April 2007.