Adaptive Burst Transmission Scheme for WSNs

Zeeshan Ansar and Waltenegus Dargie

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

***Abstract*—Recently, bulk-data or burst transmission has been the focus of research in wireless sensor networks. The aim is to benefit applications which require high throughput but small scale deployment. Healthcare applications and applications which are intended to support independent living for the elderly are typical examples. The main idea is to exclusively provide the channel for one transmitter only until it has transferred all of the packets it has accumulated in the buffer. This exclusive use avoids aimless contention and significantly reduces packet transmission latency by dispensing with repeated clear channel assessment, random back-off, and the transmission of RTS and CTS packets for every single packet. Existing or proposed MAC protocols supporting bulk-data transmission, however, do not react well to link quality fluctuation, since nodes make repeated attempt to retransmit lost packets even when the statistics of received packets suggests that the channel is still bad or packet transmission will be deferred arbitrarily even though packet loss is an isolated and uncorrelated occurrence. In this paper, we address these issues and estimate the duration of good and bad links from the statistics of received ACK packets. Moreover, we provide a MAC layer solution to enable the coexistence of multiple transmitters during bulk-data transfer.**

***Index Terms*—Burst transmission; link quality estimation; Bursty links; link quality fluctuations; wireless link; intermediate links; mobility; sleep-time; energy-efficiency**

I. Introduction

Existing physical and medium access layer protocols in wireless sensor networks suffer from two types of drawbacks. Firstly, they are designed to provide low data rate communication by employing low-power operations. The underlying assumption for this is that events of interest are rare and, therefore, it suffices to sample sensors occasionally or at long intervals. There are, however, applications, such as healthcare applications employing wireless electrocardiograms, 3D accelerometers, and 3D gyroscopes, which require high throughput. Secondly, the protocols are unresponsive to link quality fluctuation. As a result, in case of communication in harsh environments, the packet loss rate becomes considerably high to have an adverse impact on the quality of sensing as well as the energy efficiency of the entire network. Admittedly, there are some physical layer solutions to deal with this challenge, such as dynamic channel selection, synchronous communication, and dynamic adjustment of transmission power. However, these solutions are best suited for reacting to instant fluctuations and exhibit their own shortcomings.

Emerging contention based protocols attempt to address these drawbacks as follows: Instead of a packet-by-packet contention to win the medium (which requires separate clear channel assessment, random back-off, and the transmission of RTS and CTS packets for every data packet), they advocate bulk transfer. The strategy enables nodes to transfer all the packets they have in succession as soon as they win a medium [1], [2], [3]. If, during this time the transmitting node experiences packet loss, it suspends transmission for a brief amount of time or backs off randomly [4]. These approaches, however, have their own drawbacks. Firstly, they do not set a limit to the number of packets which should be transmitted in burst. Without a limit, a node may occupy the medium indefinitely, particularly, in case of frequent packet transmission failures and repeated attempts to retransmit lost packets. Secondly, temporal suspension is made under the assumption that link quality deterioration is a transient characteristic, which may hold for static deployments, but does certainly not hold for mobile deployments.

In order to illustrate the significance of this statement, we refer to Figure. 1 where we have three different scenarios. In the first scenario, a static transmitter sent packets in burst to a static receiver whereas in the second and third, a transmitter carried by a moving robot transmitted packets in burst. In the second scenario the robot took a random walk whilst it moved in a straight line in the third [5]. We used three different metrics to evaluate the fluctuation of link quality, namely, psr, rssi, and LQI. psr, packet reception ratio, is the ratio of the number of received ACK packets to the number of transmitted data packets in a given period of time. It can be expressed as a function of time by taking a sliding window. For the mobile node scenarios, as shown in Figure 1 (b) and (c), the link quality deteriorates considerably which signifies the need for an online link quality estimator to efficiently manage packet transmission.

In this paper we propose a link-quality-aware burst transmission protocol which takes the peculiar aspects of mobile scenarios into consideration. The protocol attempts to address some important issues: (1) How long should nodes transmit in burst and how should this duration be determined? (2) How does burst transmission accommodate the co-existence of multiple communicating nodes? and (3) How could the efficiency (in terms of channel utilisation, throughput, reliability, and energy, for example) of burst transmission be guaranteed?

The rest of the paper is organized as follows: In Section II, we review work on bulk data transfer, burst forwarding and bursty links. In Section III, we introduce our approach to determine the durations of good and bad links. In Section IV, we present our MAC protocol which enables burst transmission. In Section V, we discuss the implementation of our protocol

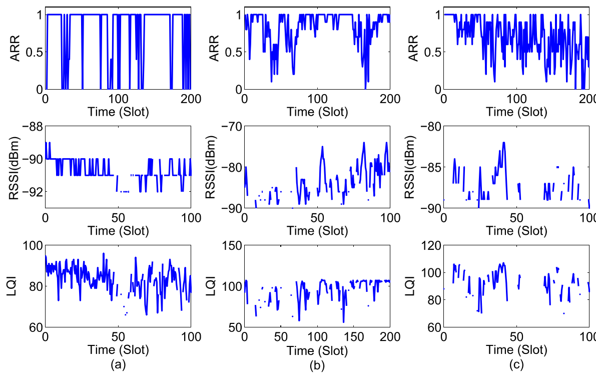


Fig. 1: Link quality metrics describing the fluctuation of links between (a) two static nodes (b) a static node and a mobile node (with a random walk) and (c) a static node and a mobile node (moving in a straight line).

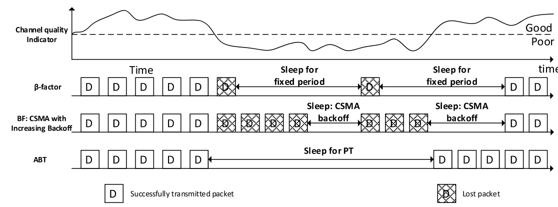


Fig. 2: A comparison of different burst transmission strategies (β-factor, BF and ABT) to deal with frequent link quality fluctuations.

and in Section VI, we provide the quantitative evaluation of our protocol and compare its performance with the state-of-the-art. Finally, in Section VII, we provide concluding remarks and future work.

II. Related Work

The demand for links supporting high throughput in wireless sensor networks has given rise to a new batch of MAC protocols, the essential concepts behind being bulk data transfer and the avoidance of aimless contention.

Kim et al. [6] propose FLUSH, a multi-hop bulk data transport protocol for wireless sensor networks. It is a result of a cross-layer design with the assumption that a single active flow should be supported in a network at a time, which connects a source node with the base station (the sink). Flush uses a dynamic rate-control algorithm in each hop along the path towards the sink in order to avoid intra-path interferences. The dynamic rate control algorithm adopts a snooping strategy and requires no extra control packets.

Raman et al. propose PIP (packets in pipe) [1], a TDMA based approach to transfer bulk data in a multi-hop network. PIP utilises TDMA, multi-channel operation and conditional immediate transmission technique propose by Österlind et al. [7] to achieve a high throughput. PIP assumes that the underlying link quality is stable and all the nodes participating in the transfer are always active (on) during the data transfer, which can be a potential cause of resource wastage.

The focus of the above protocols is to achieve high throughput without taking link quality fluctuation into consideration. But link quality fluctuation is one of the most significant challenges for wireless links, particularly, for those which should support mobile nodes. Duquennoy et al. [2]address this problem and propose a generic burst-forwarding (BF) technique which combines duty cycle with high throughput in bulk data transfer. BF employs a two-level retransmission scheme to overcome isolated and consecutive losses. At the first level of retransmission, a lost packet is retransmitted immediately. If retransmitted packets are lost repeatedly within a fixed period of time, the MAC layer stops burst transmission and backs-off using CSMA/EB. In [8], the authors employ a Hidden Markov Model (HMM) to estimate the durations of poor channel quality (what they call, a push-back period). Accordingly, 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.

One other way of dealing with link quality fluctuation and aimless contention is to employ diversity schemes where one and the same packet is simultaneously transmitted by many senders to multiple receivers. Recently, several researchers [3], [9], [10], [11] have proposed synchronous transmission in order to exploit constructive interference (CI) and Capture Effect (CE) phenomenon in order to improve the throughput of the system and mitigate link losses. However, this approach requires strict synchronization. However, even though these approaches may provide high throughput, they can do so at the cost of high power consumption.

We conclude this section by observing that most of the above protocols focus only on bulk transmission and do not address issues related to the time-varying characteristics of the wireless channel as a result of which they suffer from high packet loss and high energy wastage. Our work, on the other hand, not only focuses on providing high throughput but also on duty-cycle (by enabling nodes to sleep during poor channel conditions). Our approach enables the MAC protocol to learn the channel characteristics quickly and adaptively, and schedules packet transmission accordingly. Moreover, as our estimation scheme provides a more accurate information as regards the number of packets which can be transmitted or deferred, this information can be utilised by neighbour nodes to adapt their duty-cycle or transmission schedule.

III. Adaptive Burst Transmission

In order to motivate our proposal, we refer to Fig. 2. Suppose the quality of a given wireless link is described as shown in the figure (top). The threshold line is drawn to suggest that packets with a link quality metric below this threshold will not be delivered successfully. If a transmitter has this information a priori, it will stop transmitting exactly before the link quality falls below the threshold line and resumes transmission when it rises above the threshold line. But this information is impossible to obtain a priori. One of the proposed approaches, the β-factor, suspends transmission upon a single packet failure (shown in the middle) for a specific amount of time, then it resumes burst transmission. If, upon resumption, the packet is still lost, the scheme suspends transmission once again for the same duration. During suspension, the radio is turned off to save energy. The suspension period is determined empirically. The second approach, *burst forwarding*, does not suspend transmission upon a single packet failure; instead, it attempts to retransmit the lost packets, but if the attempt is not successful for the n-th time, it performs a random back-off before it attempts to resume transmission. If it still experiences failure, it increases the back-off window exponentially and performs a random back-off once again.

A more efficient approach would be to rely on the statistics of incoming ACK packets in order to determine the expected durations of good and bad states. In a static deployment, the link quality fluctuation statistics can be regarded as stationary in a wide sense, in which case, it is sufficient to transmit a large number of packets once, establish the statistics offline, determine the expected durations of good and bad states, and use this knowledge to schedule packet transmission and sleep times [12], [13]. The statistics can be refreshed at runtime by evaluating the link quality metrics of received ACK packets [14].

When a transmitting node is mobile (assuming that the predominant traffic flow is from the mobile node to a static relay node), the link quality fluctuation cannot be regarded as stationary. Furthermore, the statistics established offline may not accurately represent the current link quality fluctuation, since it is impossible to accurately emulate or reproduce movement patterns. Hence, the duration of good and bad states should be estimated online. Fortunately, compared to the speed of the mobile node (if the mobile node is carried by a human being), the packet transmission rate is higher, that it is possible to gather sufficient statistics and to predict with it the short-term link quality fluctuation. For example, with a data rate of 250 kbps, packet size of 28 B, and Inter-Packet-Interval (IPI) of 10 ms, approximately 92 packets can be transmitted in a second. If we assume that the person moves at 3 km/h and 1 m corresponds to one step, then the mobile transmitter can transmit the 92 packets before the person makes a single step (or travels 0.8m).

Suppose the mobile node transmits 100 packets in burst, some of which may fail to get delivered successfully due to a bad link, and collects ACK packets. From the sequence of the ACK packets, it is possible to determine the probability of successfully transmitting n number of packets in succession. Similarly, it is possible to determine the probability of losing m number of packets in succession. The expected number of packets which can be transmitted in succession successfully describes the expected duration of a good state and can be expressed as:

$$g = \sum_{n=1}^{100} n p_n \tag{1}$$

where p_n is the probability of successfully transmitting n number of packets in succession. Likewise, the expected number of packets which can be lost in succession describes the expected duration of a bad state and can be expressed as:

$$b = \sum_{m=1}^{100} m p_m \tag{2}$$

where p_m is the probability of losing m number of packets in succession. Since we have claimed that the link quality of a mobile link is not stationary, statistically speaking, the quality of the prediction we make with Equations 1 and 2 depends on the history data and how well they represent the current state of the link. A fixed history size with a sliding window can be used to update the link quality statistics, but this approach is inflexible. For example, suppose that after the transmission of 100 packets, we determined that $g = 10$ and $b = 6$. In other words, the mobile node transmits 10 packets in burst and sleeps for the duration it requires to transmit 6 packets. If after transmitting the 10 packets 9 of them are acknowledged, we can accept that our prediction was accurate and leave the history size intact. If, on the other hand, the transmitter receives only 4 ACK packets instead of 10, we can conclude that our prediction was inaccurate. Hence, when we make the prediction for the next round, we should give more emphasis to the latest statistics than the old statistics. One way to do this is by shrinking the history size and by including all the latest statistics. Similarly, if there is consistency between the statistics of the latest data and the data residing in the history buffer, we can gradually increase the history size in order to enrich our statistics.

In our approach, a mobile node initially transmits 100 packets in burst and determines the expected durations of good and bad states. Then it transmits g number of packets in burst and sleeps for the duration which corresponds with the transmission of b packets. If the number of ACK packets it receives confirms that the link quality is as expected, then the history size stays the same for the next estimation round. If, however, the number of ACK packets is lower than expected, then the transmitter should re-estimate the durations of good and bad links. To do so, it reduces the history size by half, so that the percentage of the latest statistics is high. If, once again, the number of ACK packets is lower than expected in the next round of transmission, the window size will be halved again, until it has reached 25, otherwise, it remains unchanged. We set $psr = 0.9$ to determine whether the history size should be halved. After the history size has reached the minimum size (25 packets) and the number of received ACK packets indicates that the channel is good, the transmitter waits until all ACK packets are received and when this happens, doubles the history size. Otherwise, the channel history remains unchanged for the next round of estimation.

IV. Protocol

The burst transmission scheme discussed in the previous section enables a single transmitter to efficiently utilise a link with a fluctuating quality. It does not, however, address (1) how the transmitter should coexist with other contending nodes which also wish to occupy the medium and transfer data and (2) how the medium can be shared in an efficient manner.

To address these issues, we embed the link quality information we obtained using Equation 1 (g) inside the MAC header of the data and ACK packets, and divide time into spans. The duration of a span is determined by the buffer size and the communication bandwidth of the relay node. In a single span, a single mobile node becomes a primary transmitter and all other contending nodes become secondary transmitters. If there is no mobile node in the two-hop neighbourhood of a relay node, then the relay node itself becomes the primary transmitter and transfers the accumulated data in its buffer to the base station. The reason we give priority to mobile transmitters is due to the unreliability of the link they establish with a static relay node. In a single span a mobile node first transmits N number of packets in burst to determine the channel statistics and the expected durations of good and bad links (by applying Equations 1 and 2), and based on this knowledge, defines its duty cycle. The active time of the mobile node corresponds with the short term good state in which it transmits packets in burst. Likewise, its inactive state corresponds with its short term bad state during which it sleeps to save energy.

In order to utilise the medium during the sleep period of a primary transmitter, nearby nodes can intercept data packets and utilise the information embedded in it to determine how long a good state lasts. Unlike the primary transmitter, however, these transmitters seize the medium for transmitting a single packet at a time. We limit the number of packets a sender can transmit as primary sender to 250 packets due to the limitation of available RAM storage (10 KB for TelosB) and to ensure the existence of fairness between different mobile senders. In our implementation, a single packet requires 28 bytes of RAM. Hence, 6.8 KB of memory is required to store the N packets.

An example scenario of our proposed protocol is illustrated in Fig. 3. The transmission time (TT) is a function of the total number of packets, the size of a packet, the transmission rate of the radio, and the IPI. After TT, the transmitter should vacate the medium and become either a secondary transmitter or contend for the medium all over again. TT is further divided into 'n' T_{BS} and 'n' T_{PT}, where T_{BS} is the time required to transmit g packets and T_{PT} is the pause-time corresponding to the time required to transmit b packets. Before the commencement of each T_{BS}, the history array is updated with fresh values in order to recalculate g and b.

To illustrate this with an example, consider a mobile sender MS_1 which wakes up and checks for the availability of a free medium by performing C&B. If the medium is free, it will send RTS to the receiver. Upon receiving RTS, the receiver checks the existence of a primary sender. If not, it will reply with a CTS and labels the sender as primary sender and initiates a timer to count down the duration in which the mobile transmitter stays as a primary transmitter. Once the mobile sender receives the CTS, it will start transmitting g packets in burst (Fig. 3 Case 1). When MS_1 completes transmitting the g packets in T_{BS} seconds, it will switch off its radio and go to sleep for T_{PT} seconds. Taking advantage of the intermission, the relay node (R) will start forwarding packets to the BS, as the base station is in 'always-On' receiving mode. However, the relay node transmits packets not in burst but on a packet-by-packet basis following the CSMA/CA procedure (because it is now acting as a secondary transmitter). Since the relay node does not know when MS_1 will next wake up and resume with its burst transmission, there may be a chance of collision with the first packet of MS_1, even though the probability of collision is small, as the relay node performs C&B before transmitting each packet. In case of collision, however, MS_1 will retransmit the first packet as illustrated in (Figure 3 Case 2), but this cost is tolerable.

In case of the existence of multiple mobile nodes, the protocol works in more or less the same fashion. Suppose MS_2 wakes-up during the burst transmission of MS_1 and overhears the data packets which contain information about the current burst size g and the packet sequence number (PSN) of MS_1. This overhearing notifies the contending node about the on-going burst transmission and enables it to sleep for the remaining period of T_{BS} (Figure 3 Case 3). After T_{BS}, MS_2 can wake-up and seize the medium by performing C&B. The medium can be free due to the following reasons: (i) because the transmission time of MS_1 as a primary sender is over or (ii) MS_1 is in transient-sleep. Since MS_2 does not have the history of MS_1's communication, it sends RTS to the relay node. But the relay node is aware of the actual state of MS_1, and, therefore, replies MS_2 with Data Send (DS) message which contains the total remaining time of MS_1 after which the medium will be free for contention. Upon receiving 'DS', MS_2 'knows' about the time for the next contention. Meanwhile, MS_2 can take advantage of the intermission of MS_1 and send packet to the relay node, one at a time (Figure 3 Case 4). Once the transmission time of MS_1 is over, MS_1 and MS_2 can contend for the medium again but this time MS_1 has to pick a larger random back-off thereby giving priority to MS_2 (Figure 3 Case 5).

V. IMPLEMENTATION

We implemented our protocol (adaptive burst transmission protocol, or, in short, ABT) in TinyOS and deployed it on TelosB nodes. We compare our protocol with three state-of-the-art techniques. The first is Burst Forwarding (labelled as BF), developed at Swedish Institute of computer science by Duquennoy et al. [2]. We utilize simplified version of BF as it uses a single channel rather than multiple channels and the maximum hop is set to two with no other network operating nearby. The retransmission attempt for this protocol is set to 4, as suggested in their paper. The second protocol is the β-factor (labelled simply as β), developed at Stanford university by Srinivasan et al. [4]. 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. Thus, with IPI = 20, 25, 50, and 100 ms, a maximum number of 50, 40, 20 and 10 packets, respec-

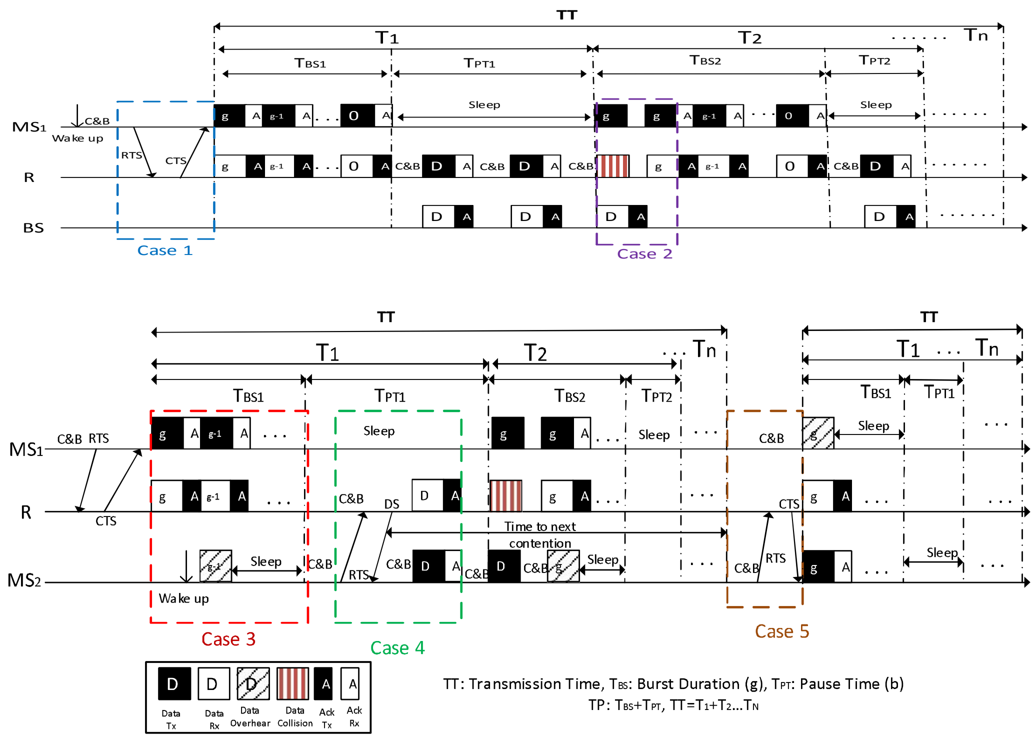


Fig. 3: An Illustration of the accommodation of primary and secondary transmitters. Top: A relay node exploits the sleep period of a mobile transmitter to transfer packets it accumulates in its buffer to the next hop in the direction of the base station. Bottom: A mobile, secondary transmitter exploits the sleep period of a primary transmitter and transfers packets to a relay node on packet-by-packet basis.

TABLE I: The summary of the network parameters for our experiment set up.

Environment	lobby, lab
Mobility Pattern	Random Walk, Straight line
Speed	1.3 - 2 m/s
Overall transmitted packets	100,000
Inter-packet transmission interval	20 ms, 100 ms (lab)
Transmission power	-25 dBm, -15 dBm
Packet payload	28 Byte

tively, can be transmitted in burst. When packet transmission fails (i.e, no acknowledgement packet is received), β halts transmission for the remaining period of time in that slot and then resumes with burst transmission at the beginning of next slot. The last protocol is Bursty Link Estimator (labelled as MAC_3), developed at RWTH (Aachen) by Alizai et al. [15]. BLE first transmits 100 packets in burst and from the history of the acknowledgement packets, determines the size of the next burst transmission. After each transmission period, a new acknowledgement sequence is added to the link history and the burst size for the next transmission is recomputed. By contrast, ABT utilize adaptive history array according to the current link quality as explained in section III.

VI. Evaluation

Our network consisted of two mobile robots carrying two transmitter nodes, a relay node, and a base station (Fig. 4).

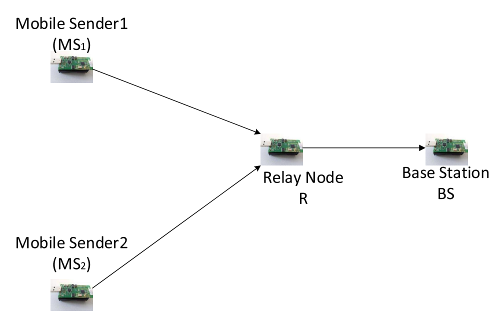


Fig. 4: The network topology of our experiments.

The size of the network is small as the focus of this paper is to show the gain which can be achieved by ABT by letting the secondary transmitter and relay nodes to transmit packets during the intermission period of the primary transmitter. Our protocol is an integral part of the MAC protocol. The mobile nodes followed two mobility patterns: a straight line walk and a random walk. We deployed the network both in the lobby of our faculty and in our lab (for the lab deployment we used Yokogawa digital power analysers to measure the energy consumption). Table I summarises the experiment parameters.

A. *Efficiency of burst transmission scheme*

We evaluated the performance of the four transmission schemes for a single-hop link using the following metrics: (1) throughput, (2) transmission time, and (3) energy consumption. In this experiment the transmitter node is mobile and the base station is static.

Throughput is an important evaluation metric in wireless sensor networks, particularly for aggregating nodes which are closer to a base station. We transmitted 5,000 packets with each mobility pattern. Figure 5(a) compares the throughput of our scheme with the three online transmission schemes. Each bar graph represents the average throughput of 10 independent experiments. For both movement patterns ABT has the highest throughput. The reason is that ABT deals with short-term link quality fluctuations by adapting the history size which results in adapting the burst size according to the link quality dynamics. On the other hand, BF stops burst transmission on 4 consecutive losses and calls random back-off resulting in poor channel utilization. The assumption that after a loss of 4 consecutive packets the channel should be regarded as unreliable is apparently less reactive to a highly dynamic link, which is typically the case for mobile, wireless links. In contrast to BF, β differs transmission for remaining period of the slot as soon as it encounters a failure and regards all types of failures as similar even though the underlying conditions can be 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 contains outdated data and as a result, the future burst size is often wrongly estimated.

The packet transmission time (or delay) is another way of looking at the throughput. It refers to the time required to successfully transfer a fixed number of packets. The term successfully" indicates that lost packets were retransmitted. Figure 5(b) displays the time required to transmit 5000 packets successfully with each mobility pattern. In accord with the results we observed during the evaluation of throughput, here, too, ABT performs better than the others.

To measure the energy consumption of the sensor nodes, we made the transmitting nodes stationary and instrumented them but made the relay node mobile. All the transmission schemes used the same transmission configurations and delivered 5000 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 between samples was required. 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 5(c) 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 BF. ABT resulted in the least amount of energy consumption in both cases.

The reason ABT performs better than other schemes is that it deals with short-term link quality fluctuation by estimating the link quality in real-time and adapting the history size accordingly. As in case of mobility the link quality is more dynamic in comparison to static deployment. All the others techniques offers fixed solution as they meant for static scenario and unable to deal with short-term fluctuation as introduce by the mobility.

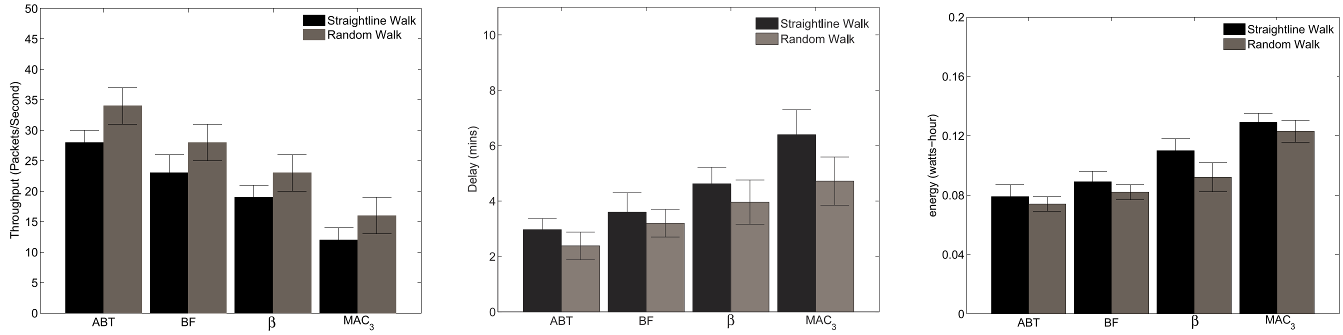
B. *Co-existence of multiple transmitters*

When multiple mobile nodes exist in a wireless sensor network and the network should serve them impartially, a single node should not monopolise the available channel indefinitely. Therefore, the idea of bulk-data transfer should not contradict with the idea of fairness. We compared the performance of our protocol with BF, which is the only protocol closest to ours as it supports cross-transfer of data during burst transmission. However, it utilizes different channel to support cross-transfer. We chose the random-walk mobility pattern for both mobile nodes and transmitted 5000 packets with each protocol.

Figure 6(a) shows the interaction between the primary transmitter (burst transmission) and the secondary transmitter (packet-by-packet). The experiments run for 160 seconds. MS_2 was awake 30 seconds after MS_1 began transmitting packets in burst. Before MS_2 joined, ABT achieved twice the throughput of BF for the same duration due to the efficient transmission technique. When MS_2 woke up and started contending for the medium, the throughput in BF was reduced by half as MS_2 has no knowledge of transmission duration of MS_1 and was contending for the medium contentiously. For our case, however, the throughput of MS_1 was not significantly affected as MS_2 utilised only those time periods when MS_1 stopped transmission. Figure 6(b) compares the real time energy consumption pattern of MS_2. In case of BF, MS_2 stayed "always on" to contend for the medium as it had no knowledge about the duration for which MS_1 would stay active. This resulted in energy wastage because the relay node consumed approximately 0.05 W when it was on a receiving mode as depicted in Figure 6(b). In case of ABT, MS_2 could switch to sleep mode directly after overhearing the data or ACK packet.

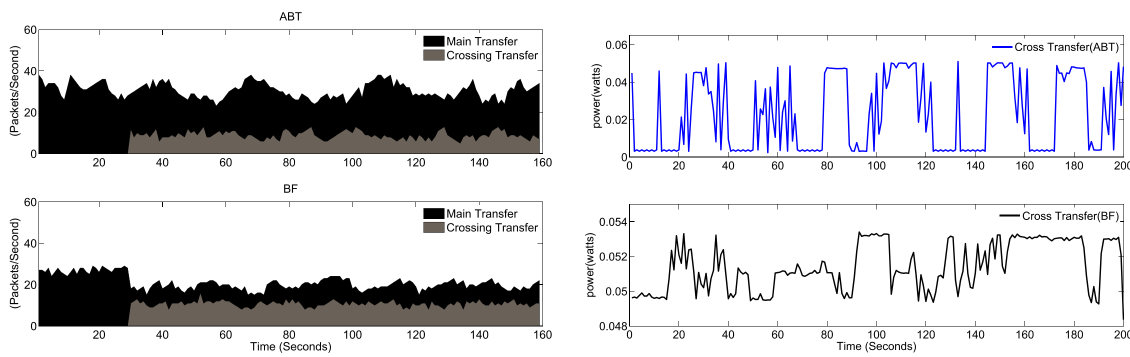
VII. Conclusion and Future Work

In this paper we proposed an adaptive burst transmission (ABT) protocol for providing high throughput and adaptive duty-cycles in wireless sensor networks which accommodate mobile nodes. Our approach deals with link fluctuation by estimating the durations of good and bad states from the statistics of incoming ACK packets. Furthermore, ABT enables neighbor nodes to share information pertaining to link states and, thereby, to achieve better channel utilisation. We compared the performance of our approach with the state-of-the-art *Burst Forwarding* approach. We demonstrated how our approach was able to achieve higher throughput, minimum packet transmission time, and less energy consumption. The



(a) Throughput of the mobile sender.
(b) Overall packet transmission time.
(c) The energy consumption of the mobile sender.

Fig. 5: Comparison of different burst transmission schemes.



(a) The throughput of the mobile senders.
(b) The energy consumption of a contending node.

Fig. 6: ABT allow multiple transmitters to co-exist by sharing the link quality metrics. In case of ABT the throughput of Main Transfer (primary transmitter) is not affected by Cross Transfer (secondary transmitter).

reason for this being its adaptive and efficient link estimation strategy. Our approach is easily integrable with existing or proposed MAC protocols and complies with the existing IEEE 802.15.4 specification. One of our future tasks is to extend the size of the network and to test the scalability of our approach.

Acknowledgment

This work has been partially funded by the German Research Foundation (DFG) under project agreement: DA 1211/5-2.

References

[1] B. Raman, K. Chebrolu, S. Bijwe, and V. Gabale, "Pip: A connection-oriented, multi-hop, multi-channel tdma-based mac for high throughput bulk transfer," in *SenSys '10*, (New York, NY, USA), pp. 15–28, ACM, 2010.

[2] S. Duquennoy, F. Österlind, and A. Dunkels, "Lossy links, low power, high throughput," in *SenSys '11*, (New York, NY, USA), pp. 12–25, ACM, 2011.

[3] M. Doddavenkatappa and M. Choon, "P3: A practical packet pipeline using synchronous transmissions for wireless sensor networks," in *IPSN 14*, pp. 203–214, April 2014.

[4] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The beta-factor: measuring wireless link burstiness," in *SenSys 2008*, pp. 29–42, 2008.

[5] J. Wen, Z. Ansar, and W. Dargie, *MobiLab: A Testbed for Evaluating Mobility Management Protocols in WSN*, pp. 49–58. Cham: Springer International Publishing, 2017.

[6] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica, "Flush: A reliable bulk transport protocol for multihop wireless networks," in *SenSys '07*, (New York, NY, USA), pp. 351–365, ACM, 2007.

[7] F. Österlind and A. Dunkels, "Approaching the maximum 802.15.4 multi-hop throughput," 2008.

[8] 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.

[9] H. Rahul, H. Hassanieh, and D. Katabi, "Sourcesync: A distributed wireless architecture for exploiting sender diversity," in *SIGCOMM '10*, (New York, NY, USA), pp. 171–182, ACM, 2010.

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

[11] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, (Lombard, IL), pp. 269–282, USENIX, 2013.

[12] J. Wen, Z. Ansar, and W. Dargie, "A link quality estimation model for energy-efficient wireless sensor networks," in *ICC 2015*, 2015.

[13] Z. Ansar, J. Wen, E. D. Ayele, and W. Dargie, "An efficient burst transmission scheme for wireless sensor networks," in *MSWiM '15*, (New York, NY, USA), pp. 151–155, ACM, 2015.

[14] Z. Ansar, J. Wen, and W. Dargie, "Efficient online burst transmission scheme for wireless sensor networks," in *The 25th International Conference on Computer Communication and Networks (ICCCN 2016), August 1-4, 2016, Waikoloa, Hawaii, USA*, 2016.

[15] 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.