

Green Partial Packet Recovery in Wireless Sensor Networks

Anas Daghistani^{*§}, Abderrahman Ben Khalifa⁺, Ahmad Showail^{*}, and Basem Shihada^{*}

^{*}CEMSE Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

[§]Department of Computer Engineering, Umm Al-Qura University (UQU), Makkah, Saudi Arabia

⁺Telecommunication Department, INSA Lyon, Lyon, France

ahdaghistani@uqu.edu.sa, abderrahman.ben-khalifa@insa-lyon.fr, {ahmad.showail, basem.shihada}@kaust.edu.sa

Abstract—Partial packet recovery is well known for increasing network throughput and reducing frame retransmissions. However, partial packet recovery methods in the literature are not energy-aware and hence they are not suitable for the battery powered wireless sensor nodes. We propose Green-Frag, a novel adaptive partial packet recovery mechanism that is energy friendly. It can help prolonging the battery life of wireless sensor nodes that are usually resource constrained. It dynamically partitions the frame into smaller blocks to avoid dropping the whole frame due to a single bit error. Also, Green-Frag is able to tolerate high interference and save energy by varying the transmit power based on channel quality and interference pattern. We experimentally evaluate the energy efficiency as well as goodput and delay of Green-Frag using our TelosB sensor node testbed. We find that Green-Frag reduces energy consumption by 33% on average compared to the state of the art partial packet recovery scheme in the literature in the presence of Wi-Fi interference. In the worst case, this reduction in energy consumption comes at the cost of 10% reduction in goodput. Finally, Green-Frag reduces the latency by 22% on average compared to other static frame fragmentation schemes.

Index Terms—Wireless Sensor Networks, Partial Packet Recovery, Frame Fragmentation, Adaptive Power, Interference-Resiliency, Energy-Efficiency

I. INTRODUCTION

Power management is an active area of research in Wireless Sensor Networks (WSNs). Efficient power management in WSN is necessary because wireless nodes are usually battery-powered and are often deployed in mission-critical applications. For WSN nodes to be energy efficient, they should be able to smartly choose both frame size and transmit power based on channel interference level. In fact, finding the optimal frame size is challenging in wireless networks. Large frames can provide good channel bandwidth utilization due to its low overhead in low interference environments. On the other hand, when the channel quality is bad, small frames provide better network utilization because of less overhead in the error recovery process. Large frames are often used in wired communication because the wired channel has low bit-error rate (BER), typically 10^{-15} to 10^{-12} . However, the BER in wireless networks is orders of magnitude higher [2], typically 10^{-5} to 10^{-3} . Additionally, BER in wireless networks changes

dramatically over short time intervals [3], [4], [5], [6], [7]. In fact, partial packet recovery techniques may be used to solve this issue retransmitting only the corrupted portion of the previously transmitted frame. However, current partial packet recovery mechanisms have only focused on improving the throughput by limiting retransmissions [8]. The energy efficiency aspect of these schemes has never been studied in the literature.

In general, WSN protocols use small data frames to avoid the overhead associated with retransmitting lost or corrupted frames. However, these small frames incur high overhead as each of these frames need additional PHY and MAC layer headers. These headers include sender and receiver IDs, CRC for error detection, and additional bytes for synchronization. The data link layer is responsible for partitioning the original payload into frames. Thus, when the wireless channel BER is low, using large frames help amortizing the PHY and MAC header overhead over large data payloads. Once BER becomes high, large frames could effectively lower the overall throughput due to more frequent retransmissions of large frames. Thus, the optimal frame size depends on the wireless channel quality, which varies over time and environmental conditions [9], [10]. Previously, we proposed two dynamic frame fragmentation schemes for WSNs, iFrag [11] and Hi-Frag [12]. These aim to achieve high goodput by dynamically changing frame partitioning according to the channel conditions.

In this paper, we analyze recent frame fragmentation schemes from both energy efficiency and throughput perspective. We experimentally compare two main types of frame fragmentation techniques, static and dynamic, in order to find the most energy-efficient scheme. We show that dynamic frame fragmentation techniques outperform other static approaches in terms of energy efficiency. Hence, using the most energy-efficient dynamic frame fragmentation scheme we develop a green protocol that can adapt to interference patterns. This novel scheme is called Green-Frag and it considers the environmental interference levels and patterns when deciding about the optimal frame structure and transmit power. Green-Frag aims to achieve high level of energy efficiency in all channel situations while maintaining a good level of throughput and delay performance.

Our experimental analysis show that Green-Frag outperforms all other schemes in terms of energy-efficiency while

A preliminary versions of this work [1] was presented at IEEE WiMob2013 and received the best student paper award.

maintaining comparable throughput and delay.

The main contributions of this paper are as follows:

- 1) Compare existing frame fragmentation schemes proposed in the literature for WSNs in terms of energy consumption.
- 2) Design, implement, and evaluate a green partial packet recovery scheme for WSN nodes that adapts the nodes transmit power based on channel quality.
- 3) Propose a novel method for experimentally calculating the energy consumption in WSN nodes.

This paper is organized as follows. Section II provides an executive summary of existing partial packet recovery protocols. Section III provides an energy evaluation of various static and dynamic frame fragmentation schemes, which also serves as a motivation for Green-Frag. In Section IV, we introduce Green-Frag and various considerations governing its design. Section V presents our experimental results. Finally, the conclusion and future work is discussed in Section VI.

II. RELATED WORK

There is a lack of literature on energy efficient partial packet recovery techniques. In fact, the main focus of all of the proposed partial packet recovery techniques is on increasing the throughput without studying the effects of these techniques on energy consumption. To the best of our knowledge, this work is the first experimental analysis that studies various partial packet recovery techniques from an energy consumption perspective. In this section, we discuss previous partial packet recovery techniques that were proposed in the literature.

Frame fragmentation is one of the partial packet recovery approaches. Frame fragmentation techniques can be classified into two categories: static or dynamic, based on whether they use fixed or dynamic frame sizes. One of the main static frame fragmentation techniques is Seda [3]. Seda main target is to enhance WSN throughput by reducing the number of retransmissions. Its design includes a number of enhancements that can improve the network throughput, such as reduced retransmissions and the use of compact acknowledgement (ACK) frame. Seda divides each frame into identical-sized blocks. It then adds a block number and a Cyclic Redundancy Check (CRC) to each block. This allows the receiver to identify corrupted blocks and only request for their retransmission. The authors of Seda claim that a block size of 20-25 bytes provides near-optimal throughput. However, this is not always correct because it highly depends on both the channel condition and the Bit Error Ratio (BER) as we show in [11]. A similar static partial packet recovery technique, but for wireless local area networks, was proposed in [13]. This scheme is called Fragment-Based Retransmission (FBR) and it works as follows. Within the same channel access, the sender tries to retransmit all the corrupted blocks. In fact, the authors mentioned that either 2 or 4 blocks could be used per packet. However, they did not discuss on what basis FBR is going to choose the number of blocks per frame. Moreover, it is not clear how the receiver is going to figure out the number of sent blocks per frame given that it changes over time. Finally, network fairness could be significantly degraded due to the extension of the sender transmission chance.

There are other partial packet recovery techniques in the literature that are dynamically changing the size of packet blocks. The authors of [14] try to maximize the throughput by proposing an adaptive subpacket scheme that optimizes the block size. The adaptive algorithm depends on SNR of the channel to change blocks' sizes. However, the authors never mentioned how does the receiver know once the sender decides to change the size of the blocks. Moreover, it is not possible to preserve data integrity without having a block number assigned to each block. In [15], the author proposed to use Luby-type erasure code for symbol recovery. The protocol depends on the channel BER to select the segment size. However, the author assumed that the sender has a precise knowledge of the channel BER, which is not realistic. Moreover, the assumption that the feedback channel is error-free and has no delay is not always true. In a similar work, the authors of [16] proposed an adaptive frame fragmentation scheme for wireless local area networks (WLANs) called Gathering Error-free Blocks (GEB). The main idea behind GEB is simple, the sender divides the frame into several blocks and the receiver gathers the error-free ones in order to assemble the original frame. GEB differentiates between dropped frames due to collision or due to high interference and adjust the contention window accordingly. This scheme suffers from unnecessary overhead because the error detection code is duplicated in the frame level as well as in the block level.

iFrag [11] and Hi-Frag [12] are two recent dynamic frame fragmentation schemes that were specifically designed to suite WSNs. iFrag [11] is a dynamic block size allocation protocol that adapts the block size based on current channel conditions, leading to lower block loss rates and a significant reduction in block retransmissions. This improves data transmission reliability, resulting in high network throughput. iFrag changes the partitioning size of frames dynamically depending on transmission history and some predefined thresholds. It has four predefined data frame modes, each of them partitioned differently. The four data frame mode structures are named iFrag 1, iFrag 2, iFrag 4, and iFrag 8 where the numbers represent the number of data blocks in the frame structure. As the number of blocks in the frame increases, the frame size increases since every block needs to have its own block number and error detection code. Hence, modes with smaller block sizes have higher overhead. The other scheme is called hybrid interference-resilient frame fragmentation (Hi-Frag) [12]. Hi-Frag is designed to reduce unnecessary retransmissions and lower the loss rates, leading to higher throughput. It does that by adaptively changing block sizes and arrangements within data frames according to the interference level and patterns. Unlike iFrag [11], it dynamically divides frames to blocks according to the observed error patterns. Also, Hi-Frag frames can contain heterogeneous blocks *i.e.* blocks with different sizes within a single data frame. Moreover, Hi-Frag reduces the per block overhead by introducing a new way of identifying block sequence numbers without necessarily having a specific field for that. This reduces the fragmentation process overhead by 50% compared to iFrag and Seda.

Another approach for recovering the part of the packet that is corrupted is packet combining. SPaC [5] recovers bit

errors by combining packets on the frame level. However, it is different from other frame fragmentation schemes in the fact that it does not divide frames into blocks. MRD [4] is another packet combining method that utilizes the presence of multiple access points in recovering process. Basically, MRD combines multiple copies of the same packet to correct bit errors. MRD is similar to other frame fragmentation techniques in the fact that it divides every packets into smaller blocks each of which has its own error detection code. However, it needs to retransmit the whole frame if the combining efforts fail, which wastes a lot of energy.

Several other techniques [17], [18], [19] used the concept of Forward Error Correction (FEC) for error recovery. In fact, FEC requires a knowledge of the channel BER. It has been shown in several papers [7], [6], [4], [5] that channels of any asynchronous wireless network suffer from unpredictable and frequent changes in channel condition. In addition, FEC requires heavy computation which makes it impractical for WSNs that are usually limited in resources.

There are other efforts that target frame size optimization in wireless networks [9] [20] [21] [22]. The basic idea behind all of them is to vary the frame size according to the quality of the channel. Recently, Dong *et al.* proposed Plena [23], which is doing exactly the same thing but in an energy efficient manner. Since these schemes do not partition the frames into smaller blocks, we believe that they could be complementing our proposed scheme.

III. MOTIVATION

Energy efficiency is extremely important in wireless sensor networks that have scarce resources. Prior dynamic frame fragmentation schemes, such as iFrag and Hi-Frag, target maximizing the overall throughput by reducing retransmissions and resisting interference. Although energy efficiency was not their main focus, such techniques can significantly lower the motes power consumption by simply reducing retransmissions. In this section, we compare the energy performance of various static and dynamic frame fragmentation schemes. We then show the potential benefits of implementing power adaptation on top of such schemes.

A. Energy Performance Comparison

We study the energy-efficiency of various frame fragmentation techniques under various channel conditions and radio transmit power levels. Our experiments compare Frame-based ARQ (FARQ), Seda, iFrag, and Hi-Frag schemes' energy performance. To make a fair comparison, the implementation of FARQ is similar to Seda except that it uses a single block instead of 4 blocks per frame. We evaluate the energy consumed for delivering one useful bit. The experiments were performed under two situations: with and without imposing external Wi-Fi interference. Experimental setup details, equations, and calculations used in this study are described in Sect. V.

Fig. 1, shows the energy performance of Hi-Frag, iFrag, Seda, and FARQ with various transmit powers. Hi-Frag outperforms all other schemes. It consumes the least energy per useful bit. Clearly, it also consumes unnecessary energy

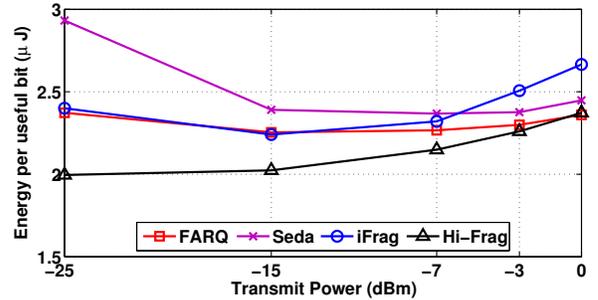


Fig. 1: Energy per useful bit without imposing Wi-Fi interference

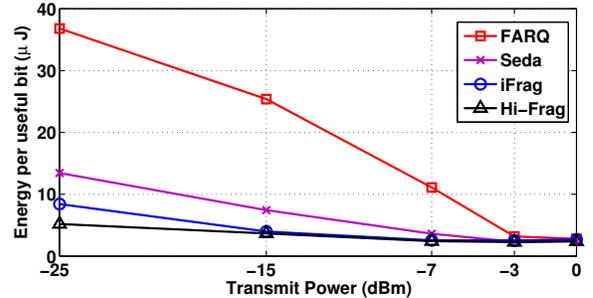


Fig. 2: Energy per useful bit while imposing Wi-Fi interference

when transmitting at high power. Overall, Seda has the worst performance due to the fixed per block overhead that is unnecessary in high quality channels. On the other hand, Fig. 2 shows the energy consumed per useful bit while imposing high Wi-Fi interference on the channel. FARQ shows the worst results because of the high cost of retransmitting the whole frame again. In fact, Hi-Frag maintains the best energy performance at all transmit powers. However, its energy per useful bit results vary between 2.3 and 5.2 μJ . This shows that Hi-Frag is not always transmitting using the optimal power.

To conclude this comparison, Hi-Frag is found to be the most energy-efficient frame fragmentation scheme under all channel conditions and transmit powers. On average, it saves 16% of energy compared to Seda in normal channel conditions and 11% compared to iFrag. With high interference, Hi-Frag shows an average of 49% less power consumption than Seda and 23% compared to iFrag. However, Hi-Frag has the potential to save more energy if it has a mechanism to choose the optimal transmit power, especially when the channel quality is high. This paper proposes a new scheme called Green-Frag that combines Hi-Frag frame fragmentation principles with transmit power adaptation.

B. Power Adaptation Benefits

In this section, we try to understand the level of energy saving that a transmit power adaptation mechanism can achieve. Lets assume that we have a network of N sensor nodes. For each sensor node S_i , where $i \in \{1, \dots, N\}$, the location is uniformly distributed over a distance range $[d_{min}, d_{max}]$. d_{max} represents the maximum distance for acceptable packet error rate (PER), i.e., $0.01 > PER > 0.001$. The corresponding

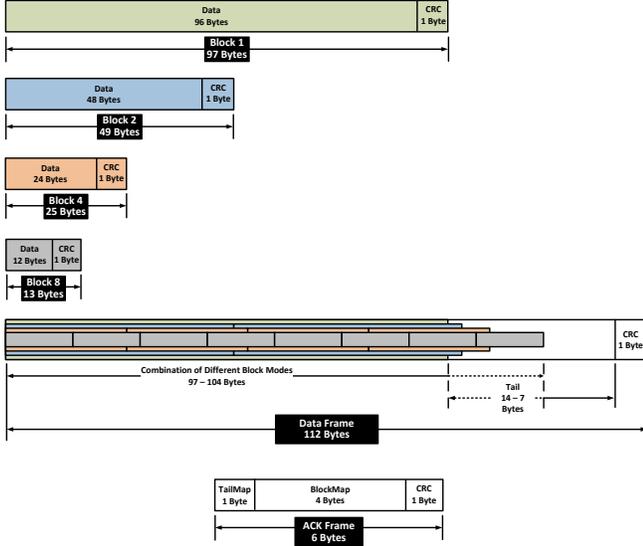


Fig. 3: Green-Frag various frame structures

PDF of each node's location at a certain distance d_i as illustrated by:

$$Pr\{D_{S_i} = d_i\} = \frac{1}{d_{max} - d_{min}} \quad (1)$$

Therefore, the average power consumption is:

$$\begin{aligned} P_{average} &= \sum_{i=1}^N (P(d_i) Pr\{D_{S_i} = d_i\}) \\ &= \sum_{i=1}^N P(d_i) \frac{1}{d_{max} - d_{min}} \end{aligned} \quad (2)$$

where $P(d_i)$ is the optimal power that could be used for transmitting to another sensor node at distance d_i . Subsequently, the power saved using the proposed power adaptive mechanism is:

$$P_{save} = Pr\{D_{S_i} = d_i\} [NP(d_{max}) - \sum_{i=1}^N P(d_i)] \quad (3)$$

This illustrates that significant energy savings can be obtained by implementing an adaptive transmit power mechanism.

IV. GREEN-FRAG DESIGN

Green-Frag reduces the energy consumed in delivering useful data under all channel conditions by combining a dynamic frame fragmentation technique with an adaptive power mechanism. In fact, Green-Frag is the first partial packet recovery technique that uses adaptive transmit power control. It implements a dynamic frame fragmentation technique that use hybrid block sizes within data frames, which minimizes packet loss rates and reduces retransmissions. Further, Green-Frag uses an adaptive transmit power mechanism to minimize energy usage without reducing link reliability. Most of the prior partial packet recovery techniques aim to increase the throughput without caring about energy consumption. On the other hand, Green-Frag targets lowering energy consumption as well as maximizing the throughput.

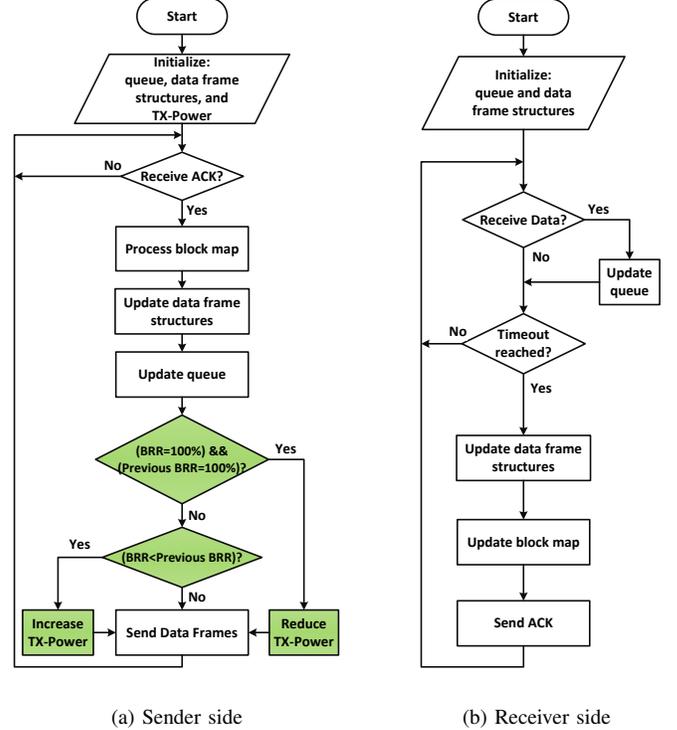


Fig. 4: Green-Frag operation flow chart

A. Frame Structure

Green-Frag data frame and ACK frame structures are shown in Fig. 3. Data frames are fixed to 112 bytes of payload; this is the maximum payload supported by the TelosB [24] hardware. An additional 16 bytes are used by the MAC and PHY layers. Each frame can contain combination of blocks with different sizes and a tail that fill the remaining frame size with data.

Green-Frag implements four main block modes, each with a different block size. These are Block 1, Block 2, Block 4, and Block 8, where the numbers represent the maximum number of blocks from the same mode that can fit in a single frame. For example, eight blocks of mode Block 8 can form a complete data frame. Each of those block modes contains a data field and a 1 byte CRC field. The data field varies between 12 and 96 bytes, depending on the block mode. The sender includes the frame number in the CRC calculations with the data field. This allows the receiver to identify frames during a frame loss, thus reducing data frames overhead.

In our Green-Frag implementation, a session is composed of four data frames. Each of them could have different data frame structure. This structure can also change after each session. A data frame has two main fields: blocks field and tail field. Blocks field could have a combination of different block modes, which make it vary from 97 to 104 bytes. The tail field is completing the frame with data bytes. Therefore, it varying from 7 to 14 bytes. Also, it has 1 byte CRC field for checking the correctness of the tail.

ACK frame is sent by the receiver after each session to inform the sender about the corrupted and missed blocks. It consists of 1 byte TailMap field, 4 bytes BlockMap field, and 1 byte CRC. Out of the 1 byte TailMap field 5

Algorithm 1: GREEN-FRAG SENDER

```

1 Initiate connection and inform the receiver of supported block modes
2 Initialize TX-Power to median of the supported transmit powers
3 Initialize Data Frame Structures to the smallest block mode
4 Arrange data into blocks to form frames according to Data Frame Structures
5 Add CRC to blocks and tails, reframe and handoff packet to MAC-layer
6 if ACK received then
7   for update all Data Frame Structures do
8     for all blocks check status from BlockMap field do
9       if (Block correct) && (aligned) && (next block correct) then
10        Merge two consecutive blocks into one larger block
11      else if (Block corrupted) && (size not the smallest) then
12        Divide into two smaller size blocks
13      else
14        Use the same block size
15    calculate BRR
16    if (BRR = 100%) && (Previous BRR = 100%) then
17      Reduce TX-Power
18    else if (BRR < Previous BRR) then
19      Increase TX-Power
20    else
21      Use the same TX-Power
22    Form new frames according to the Data Frame Structures and the
23    BlockMap field of the ACK containing missed bytes and new data bytes
24    Construct Tail to reach 112 bytes of payload
25    Add CRC to blocks and tails, reframe and handoff packet to MAC-layer

```

bits are used from as follows: 4 bits represent the four data frame tails' status, while 1 bit represents a `Color` bit. This `Color` bit is flipped each time the receiver sends a new `ACK`. Sender uses this to differentiate between successive `ACKs`. The `BlockMap` field contains the status of all blocks sent in the session. `BlockMap` which is 4 bytes in size is enough for all Green-Frag block combinations. This is because, in our implementation, the maximum number of blocks that could be sent in a session is 32 (4 frames each have 8 blocks). Both the sender and the receiver update their data frame structures of block modes depending on this field. Also, the sender adapts the transmit power based on this field.

B. Operation

Green-Frag sender and receiver operations are illustrated in Fig. 4. First, the sender and receiver agree on the supported block modes during the neighborhood discovery phase. The sender transmit a specific number of consecutive data frames with transmit power (`TX-Power`) equal to the median of the supported transmit powers (-7 dBm in our implementation). It then waits for an `ACK`. At the end of the session, the receiver sends an `ACK` containing the status of all blocks. Either the block is received correctly, or is corrupted and need to be retransmitted. The `ACK` frame maybe sent multiple times until the sender receives it and the next session is started. This design choice was made because losing the small `ACK` frame indicates a noisy channel, and thus it is more energy efficient to resend a small `ACK` instead of retransmitting the data frame that is large in size. Also, if the sender retransmits the same data of the previous session, there is a high chance that some of the blocks that are received correctly will be sent again. Therefore, sending data frames in such situations have high probability of wasting huge amount of energy.

Block arrangement in the frame varies with time. Both the sender and the receiver update these structures at the end of

Algorithm 2: GREEN-FRAG RECEIVER

```

1 Connection establishment (know sender supported blocks modes)
2 Initialize Data Frame Structures to the smallest block mode
3 Send ACK including Color and BlockMap to request data frames
4 if Data Frame received then
5   Check frame number: try from expected until last
6   for update all Data Frame Structures do
7     for all blocks check CRC do
8       if (Block correct) && (aligned) && (next block correct) then
9         Merge two blocks into one larger block for next session
10      else if (Block corrupted) && (size not the smallest) then
11        Divide into two smaller size blocks for next session
12      else
13        Use the same block size for next session
14    Check tail correctness
15  if all blocks are correctly received then
16    Re-assemble blocks into a network layer packet and hand-off to
17    network layer
18  else
19    Buffer correctly received blocks
20 construct ACK contains updated BlockMap and Color
21 if (session received) || (timeout is fired) then
22   Send ACK
23 Stop sending ACK when End Message received or timeoutend reached

```

each session. Blocks can be divided into smaller blocks or merged together forming larger blocks. The receiver updates the frame structure based on blocks' CRCs. It will also reflect the blocks status in the `BlockMap` *i.e.* a correctly received block will be represented with 1 and a missing or corrupted block will be represented by 0. In fact, the sender updates the frame structure based on this `BlockMap` after the `ACK` frame has been received. As mentioned earlier, the sender waits for an `ACK` with the correct `Color` bit to distinguish between retransmitted `ACKs`. Thus, the sender and the receiver are always synchronized with the same data frame structures. Additionally, Green-Frag sender also adapts the transmit power depending on the blocks status. The transmit power is increased or decreased depending on the ratio of correctly received blocks of two consecutive sessions.

After sending all its data, the sender sends an end message. If this message is lost, the receiver ends the connection after a predefined time `timeoutend`. Green-Frag sender and receiver operations are described in Algorithm 1 and 2 respectively. Green-Frag reduces the energy consumed in delivering useful data by dynamically choosing the block size, frame structure, and transmit power. This raises multiple challenges and design considerations. One of these challenges is to know the best way of switching between different block modes that guarantees delivering useful data with the minimal power. Green-Frag uses simple rules to switch between frame structures. The data frame structure is adapted based on blocks' status. A block is divided into two smaller blocks if it is corrupted or lost. On the other hand, two correctly received blocks could be merged into one larger block if they are aligned and next to each other. Besides these two situations, the same block mode is used. Fig. 5 shows all the possible combinations of block modes in a single data frame. Also, it shows an example of how the structure adapts, assuming that the checked and the crossed blocks represent correctly received and corrupted blocks, respectively. Green-Frag is slow in merging blocks, which is better for energy consumption because transmitting

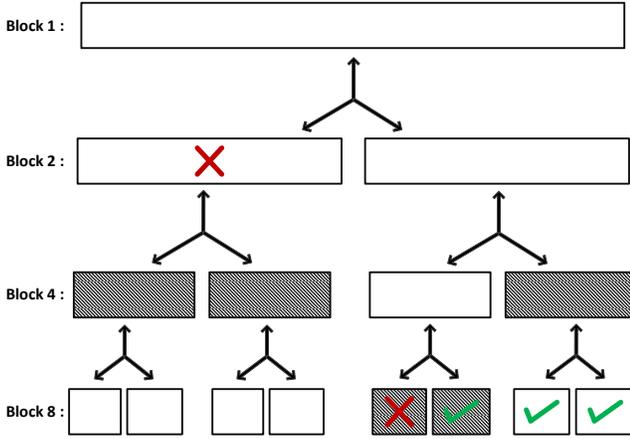


Fig. 5: The possible combinations of block modes in a frame

with additional overhead is better than being optimistic and losing bigger blocks of data.

C. Power Adaptation

Green-Frag implements an adaptive power technique that uses the minimum transmit power without losing data. One main challenge is deciding when to adjust the power. We considered two choices. The first choice is to change the power only when the channel condition is good and the system is using the biggest block mode. The other choice is to integrate adaptive power control into the frame fragmentation process. From experimental data, we observed that maximal power saving is obtained using the latter approach.

Another challenge is how to achieve transmit power control with minimal overhead. Green-Frag uses the same information for power control that it had already acquired for frame fragmentation. Green-Frag sender updates its transmit power (TX-Power) each time an ACK is received. It calculates Blocks Reception Ratio (BRR) depending on the BlockMap field in the ACK. It considers all blocks as they are of Block 8 mode. Therefore, correctly received blocks of mode Block 1, Block 2, Block 4, or Block 8 are counted as receiving 8, 4, 2, or 1 block, respectively. The maximum number of correctly received blocks in a session is 32, since a session is composed of 4 frames. Then BRR is calculated as follows:

$$BRR = \frac{\text{Count of Correctly Received Blocks}}{32} * 100 \quad (4)$$

After calculating BRR, the sender decides on increasing, decreasing, or using the same transmit power. In our implementation, the transmit power levels used are: 0 dBm, -3 dBm, -7 dBm, -15 dBm, and -25 dBm. These values includes the maximum and the minimum supported RF output power levels by our hardware [25]. The sender increases the transmit power if BRR is less than previous session's BRR. This indicates that the channel conditions have become more noisy. At the same time, the frame structure will be also adjusted according to the new situation. Sender reduces the transmit power only when both the current BRR as well as the previous session BRR are 100%. Otherwise, it keeps using the same transmit

power. This allows Green-Frag to stabilize before changing the transmit power. This adaptive power mechanism minimizes retransmission overhead by making the sender aiming always to transmit using the least power that guarantees the use of Block 1 mode.

Green-Frag receiver uses the maximum power (0 dBm) to send ACKs. This is because of the importance of receiving ACKs correctly. This does not impose any significant increase in energy consumption since ACKs are small in size and are transmitted less frequently compared to data frames.

D. Multi-hop Considerations

In the multi-hop scenarios, the sender transmits the data to the receiver over several intermediate relay nodes. These relay nodes need to have specific design features to do the relaying job. Any relay node is going to send only when it has enough data to make a session. This data is mainly composed of the correctly received data from the immediate neighbour. After sending a session to the other neighbour, the relay must keep a copy of the sent data to resend the corrupted part of the message in case it has not been received correctly. Thus, the data that a relay node sends is a combination of both newly received data and the previous data that has been received correctly. In our implementation, the relay node gives more priority to resending the missing parts from the data that has been sent over the newly received data. In fact, the relay node communicates with its neighbours under different channel conditions. In the case of static frame fragmentation schemes (Seda, FARQ) this will not cause any problem because the frame structure is fixed and does not change with the channel quality. However, in the case of dynamic frame fragmentation schemes (iFrag, Hi-Frag and Green-Frag) the frame structure is dependent on the channel condition. Hence, using the same frame structure for every neighbour will cause a lot of data loss since the sender and the receiver may have a different channel condition. To address this challenge, we design the relay node to be able to use different frame structure for each neighbour. When the relay node receives new data, it computes the correctly and corrupted received blocks and also uses this information to know the frame structure of this neighbour. For the channel on which it sends data, the relay node updates the frame structure once it receives the ACK packet. Using this technique the relay node is aware of the frame structure of each and every one of its neighbours without imposing extra overhead.

V. PERFORMANCE EVALUATION

In this section, we present an experimental evaluation of Green-Frag using our TelosB [24] testbed. We will start by showing the equations used to calculate energy consumption as well as goodput for various schemes. After that we describe our experimental setup. Finally, we compare the performance of Green-Frag to other schemes in the literature in terms of energy efficiency, goodput and delay over various scenarios.

Term	Meaning
M	The maximum supported number of blocks in a frame
H	Frame header size (bits)
F_L	Data frame length (bits)
F_{RX}	Number of received data frames
F_{TX}	Number of sent data frames
B_L	Block length (bits)
B_H	Block overhead length (bits)
B_{RX}	Number of received blocks
$Tail_{RX}$	Received tail length (bits)
ACK_L	Acknowledgment frame length (bits)
ACK_{TX}	Number of sent acknowledgment frames
P_{TX}	Power consumption of transmitting (Watts)
P_{RX}	Power consumption of receiving (Watts)
T_F	Average time to send a data frame (Sec.)
T_{ACK}	Average time to send an ACK frame (Sec.)
UB	Useful bits in correctly received frames without headers
E	Energy consumed

TABLE I: Terms used in the equations

A. Energy Efficiency and Goodput Equations

In this section we show the energy efficiency and goodput equations for FARQ, Seda, iFrag, Hi-Frag, and Green-Frag. Table I summarizes the terms used in these equations.

Radio transmission and reception is the major source of energy consumption in sensor nodes. Thus, all our energy consumption equations ignore the energy spent by CPU, memory operations, *etc.* Our energy efficiency comparison is based on how much energy is consumed for delivering a useful bit. To compute this, we divide the energy consumed in sending and receiving all frames by the number of useful received bits as Eq. 5 shows.

$$\text{Energy Per Useful Bit} = \frac{\text{Consume Energy } (E)}{\text{Useful Bits } (UB)} \quad (5)$$

The following equations show how to compute useful bits in FARQ, Seda, iFrag, Hi-Frag, and Green-Frag. Please note that useful bits are the correctly received data frame bits without counting frame headers and other scheme specific overhead such as frame sequence number. In fact, Green-Frag and Hi-Frag have similar equations due to the fact that they are using similar frame structures.

$$UB_{(FARQ)} = F_{RX} * (F_L - H) \quad (6)$$

$$UB_{(Seda)} = B_{RX} * (B_L - B_H) \quad (7)$$

$$UB_{(iFrag)} = \sum_{m=1}^M (B(m)_{RX} * (B(m)_L - B_H)) \quad (8)$$

$$UB_{(Hi-Frag, Green-Frag)} = \sum_{m=1}^M (B(m)_{RX} * (B(m)_L - B_H)) + \sum_{i=1}^{F_{RX}} Tail(i)_{RX} \quad (9)$$

Energy consumed for transmission/reception is equal to the transmit/receive power consumption multiplied by time spent in transmitting/receiving frames. The time for transmission and reception is the same because even corrupted frames will be

received. Eq. 10 shows the method used to calculate energy consumed by FARQ, Seda, and Hi-Frag, while Eq. 11 and 12 are the ones for iFrag and Green-Frag respectively. The way to calculate energy consumption for iFrag is different from others to account for the fact that it supports data frames with various sizes. Similarly, Green-Frag's energy consumption equation is different from others due to its adaptive transmit power nature. It is worth noting that the term $F(p)_{TX}$ in Eq. 12 represents the number of transmitted frames with transmit power p .

$$E_{(FARQ, Seda, Hi-Frag)} = (P_{TX} + P_{RX})[(F_{TX} * T_F) + (ACK_{TX} * T_{ACK})] \quad (10)$$

$$E_{(iFrag)} = (P_{TX} + P_{RX})\left[\sum_{m=1}^M (F(m)_{TX} * T(m)_F) + (ACK_{TX} * T_{ACK})\right] \quad (11)$$

$$E_{(GreenFrag)} = \sum_{p=P(\min)_{TX}}^{P(\max)_{TX}} [(P(p)_{TX} + P_{RX})(F(p)_{TX} * T_F)] + (P(\max)_{TX} + P_{RX})(ACK_{TX} * T_{ACK}) \quad (12)$$

In order to calculate the goodput of Green-Frag, Hi-Frag, iFrag, and FARQ experimentally, we divide the number of useful received bits by all the bits that had been sent. This should include all sent frames as well as all sent acknowledgments. The following Equation represents the method to compute the goodput as follows:

$$\text{Goodput} = \frac{UB}{(F_{TX} * F_L) + (ACK_{TX} * ACK_L)} \quad (13)$$

where each scheme has its own way of computing UB (Useful Bits) as mentioned in Eq.6, 7, 8, and 9.

B. Experimentation Setup

Our testbed is composed of 20 TelosB [24] motes with TinyOS 2.1.1 platform in an office environment. These motes use Chipcon-CC2420 radio [25] (2.4 GHz band) compatible with IEEE 802.15.4 (ZigBee) standard. Our experiments are performed at night to limit uncontrolled interference with campus Wi-Fi production network. The distance between motes is $1m$ unless otherwise stated. The motes are static and powered via USB cables to avoid low battery power issues. Also, MAC-layer automatic CRC is disabled to allow the reception of partially corrupted packets. All experiments are repeated with and without imposing external Wi-Fi interference, which is nothing but a large file transfer between two Linux machines. These machines are using IEEE 802.11g cards with 18 dBm transmit power and are $15m$ apart from each other. We select Wi-Fi since it is the main interference source in real WSN deployments such as smart buildings and traffic control applications.

In order to be able to evaluate the energy consumption of various schemes, we compute TelosB transmit and receive power consumption by multiplying CC2420 radio transceiver current consumption of transmitting and receiving [25] by

Transmit Power (dBm)	Current Consumption (mA)	Power Consumption (mW)
0	17.4	49.938
-3	15.2	43.624
-7	12.5	35.875
-15	9.9	28.413
-25	8.5	24.395
Radio Receiving:		
	19.7	56.539

TABLE II: TelosB power consumption while transmitting with various power levels

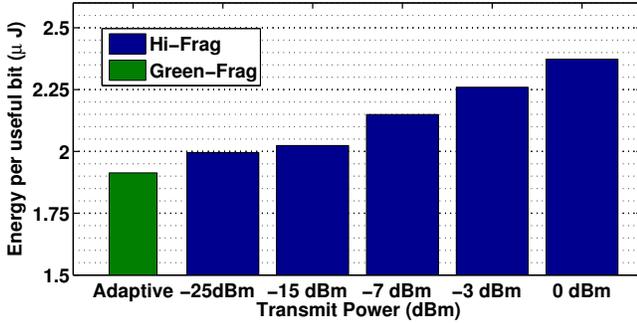


Fig. 6: Energy per useful bit of Green-Frag vs. Hi-Frag in normal channel conditions, *i.e.* without external Wi-Fi interference

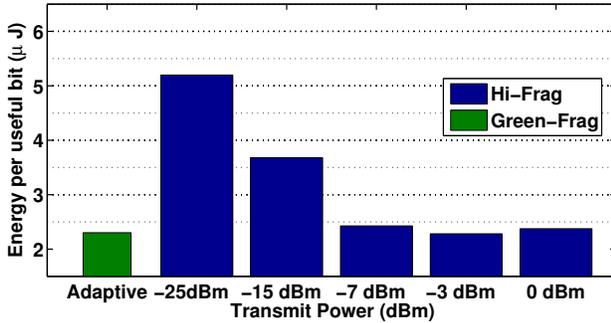


Fig. 7: Energy per useful bit of Green-Frag vs. Hi-Frag in bad channel conditions, *i.e.* with external Wi-Fi interference

TelsoB operation voltage. The operation voltage are measured and found to be constant at 2.87 V because motes are powered by USB. Table II shows the power consumption of reception and transmission at various transmit power levels.

The next step is to find the time spent in sending a data or ACK frame using any of the schemes. TinyOS does not provide the transmission time. To go around this, we implement a mechanism to measure the time from the occurrence of the send command until the end of the transmission. We then experimentally find the average of the time spent sending 3000 data frames and around 750 ACK frames over three different runs. A summary of the results are listed in Table III. From this table it is clear that Green-Frag and Hi-Frag have almost the same transmission time because they use similar frame structures. Same goes for FARQ and Seda except that the former uses one data block per frame. Since iFrag supports

Scheme	Average time to send one	
	Data frame (ms)	ACK frame (ms)
FARQ	15.755	7.427
Seda	16.419	7.348
iFrag 1	17.136	7.858
iFrag 2	17.340	
iFrag 4	17.773	
iFrag 8	18.367	
Hi-Frag	17.267	9.315
Green-Frag	17.270	9.316

TABLE III: Average time spends to send a data frame or an ACK frame in various schemes

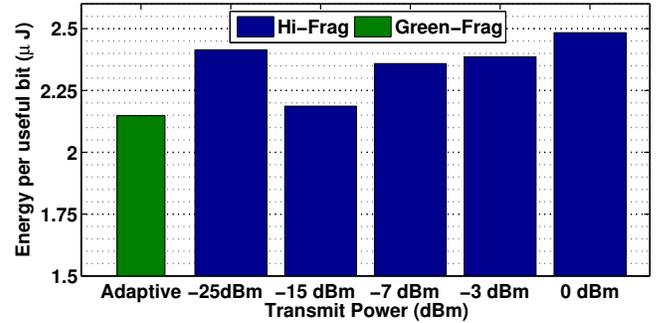


Fig. 8: Energy per useful bit of Green-Frag vs. Hi-Frag in normal channel conditions, while the distance is 2.5m

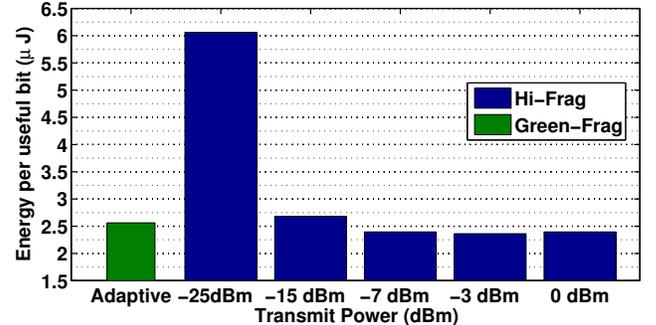


Fig. 9: Energy per useful bit of Green-Frag vs. Hi-Frag in bad channel conditions, while the distance is 2.5m

four different modes, we report the time for each one of these modes. The number associated with the word iFrag represents the number of blocks per frame. It is worth noting that the values in Table III also include time spent doing other activities in the radio, including buffering, encoding, and switching between RX and TX modes. However, these additional values are very small, and are the same across all the compared schemes.

The only remaining part to be able to calculate the consumed energy per useful bit as shown in Eq. 5 is to find the number of successfully received bits for each scheme. Therefore, several experiments are carried out with various interference and power setups. In each run, the sender mote transmits 1000 frames and results are averaged over five runs. The results are detailed in the next section.

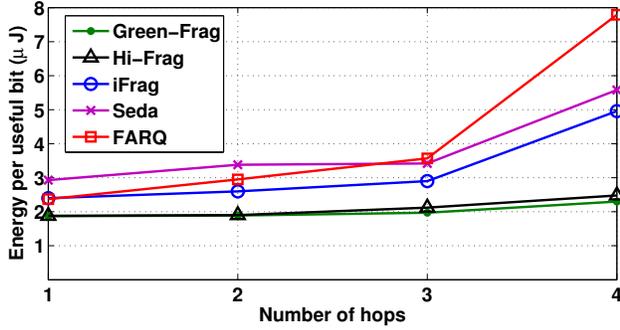


Fig. 10: Energy per useful bit of Green-Frag vs. all other schemes while varying the number of hops from 1 to 4

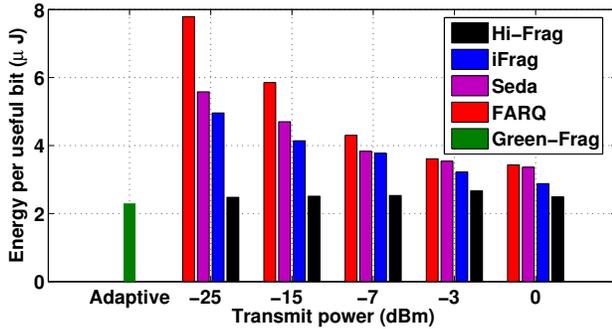


Fig. 11: Comparing the energy consumed by all the schemes over 4 hops while varying motes transmit power

C. Experimental Results

We divide our results into three parts. In the first part, we compare the performance of Green-Frag to other schemes in the literature in terms of energy efficiency. We start by showing single-hop experiments results and then show the multi-hop ones. After that, we discuss other performance metrics such as goodput and delay. Finally, we analyze in details the behaviour of Green-Frag over time compared to its competitors.

1) *Energy Consumption Results:* We start by comparing the energy consumption of various schemes in the single-hop topology *i.e.* using only one sender and one receiver that communicates directly. The comparison between FARQ, Seda, iFrag and Hi-Frag has already been shown in Sect. III. Since Hi-Frag shows the least energy consumption among all the compared schemes, we are going to use it as a base line to evaluate the performance of Green-Frag. In Fig. 6, Green-Frag is compared to Hi-Frag in terms of energy consumption of delivering a useful bit under normal channel. Hi-Frag experiments are performed multiple times using various transmit power levels, while Green-Frag controls mote transmit power adaptively. Green-Frag outperforms Hi-Frag because it changes the transmit power according to the channel quality. These results show that there is no single transmit power that is optimal at all times. Green-Frag transmits most of the time using -25 dBm because the channel quality is good. In fact, it also uses other power values as we are going to show later. When compared to Hi-Frag with maximum transmit power, Green-Frag saves about 20% of energy. On average, Green-

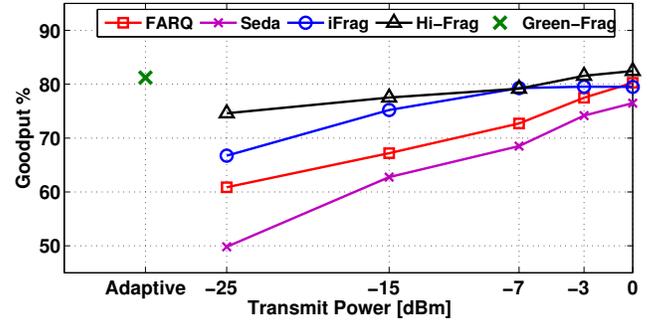


Fig. 12: Goodput of Green-Frag, Hi-Frag, iFrag, Seda, and FARQ in normal channel conditions

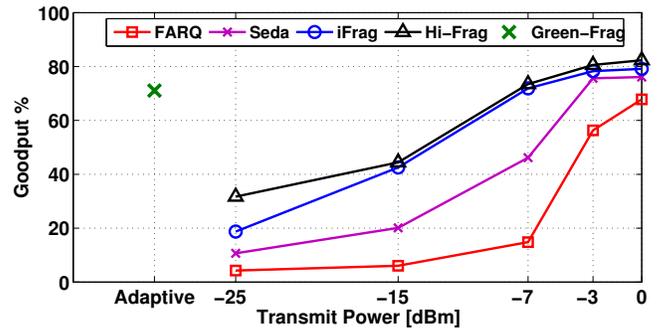


Fig. 13: Goodput of Green-Frag, Hi-Frag, iFrag, Seda, and FARQ in bad channel conditions

Frag is better than Hi-Frag by 10% across all transmit powers.

The same experiment is repeated under active Wi-Fi interference and results are shown in Fig. 7. Green-Frag achieves similar performance to Hi-Frag when transmitting at -3 dBm. However, Hi-Frag does not have a mechanism to figure out that this is the best transmit power under current channel conditions. In fact, Green-Frag consumes 56% less energy than Hi-Frag at the worst transmit power. Also, Green-Frag saves on average 33% of energy compared to Hi-Frag across all transmit powers. We note that the energy per useful bit becomes higher than that in Figure 6. This is because of data loss due to higher interference level. Also, it is interesting to note that Hi-Frag's best performance with interference is achieved when transmitting at -3 dBm whereas under normal channels the best transmit power is -25 dBm.

To fully evaluate Green-Frag in the single-hop scenario, we repeated the same experiment after changing distances between the motes. We also changed the Wi-Fi interference pattern. In this set of experiments, the motes are placed 2.5m apart from each other and the receiver is moved further away from the Wi-Fi source of interference. As a result, the transmitter/receiver separation increased while the interference level decreased. Fig. 8 shows the energy consumption per useful bit for Green-Frag compared to Hi-Frag under normal channel. As expected, Green-Frag shows the best energy performance compared to Hi-Frag in all cases. When compared to Hi-Frag transmitting at maximum power, Green-Frag saves about 14% of the energy. On average, Green-Frag is better than Hi-

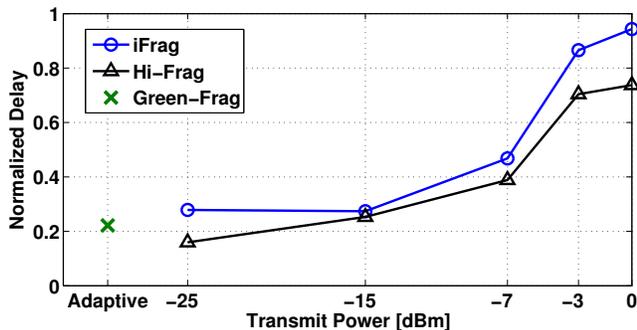


Fig. 14: Average delay per frame of Green-Frag, Hi-Frag, and iFrag vs. Seda in bad channel conditions. The y-axis shows the normalized delay results with respect to Seda

Frag by 9% across all transmit power levels. Fig. 9 presents the results of the same experiment while imposing external Wi-Fi interference. From this figure it could be noticed that Green-Frag gets stuck in a sub-optimal transmit power. This is because Green-Frag design slowly increase the transmit power when the channel quality is bad. Nevertheless, Green-Frag achieves slightly higher power consumption in comparison to the best Hi-Frag results. In fact, Green-Frag reduces energy consumption by 58% when compared to the worst transmit power used by Hi-Frag.

The energy performance of Green-Frag has been evaluated in a multi-hop scenario as well. The first experiment evaluates the energy efficiency of the proposed scheme while varying the number of hops between the transmitter and the receiver from 1 to 4. We compare the energy consumption of Green-Frag to Hi-Frag, iFrag, Seda and FARQ. For all the schemes, except Green-Frag since it uses adaptive power, we fix the transmit power to be the minimum supported power (-25 dBm) in order to minimize inter-node interference. Results are shown in Fig. 10. This figure shows that dynamic frame fragmentation protocols outperform the static ones in terms of energy efficiency. This is in agreement with the single-hop results discussed earlier in Sect. III. Green-Frag has almost the same energy consumption as Hi-Frag in 1 hop and 2 hops cases. This is due to the low inter-node interference. As the number of hops increases, Green-Frag start consuming less energy than Hi-Frag even though the latter always uses the minimum transmit power. This shows the advantage of using adaptive power technique which help choosing the optimal power that reduces retransmissions to get better energy efficiency.

In the last experiment, we fix the number of hops to 4 and vary the transmit power for all the schemes except Green-Frag. Energy consumption comparison is shown in Fig. 11. This figure shows that dynamic frame fragmentation protocols are outperforming the static ones in all cases. However, Green-Frag with the adaptive power saves more energy than both iFrag and Hi-Frag with any of the used transmit powers.

To conclude, there is no a priori-known optimal transmit power that can be used by motes. It varies based on the motes' separation, interference level and number of hops between the sender and the receiver. Hi-Frag, same goes with all other

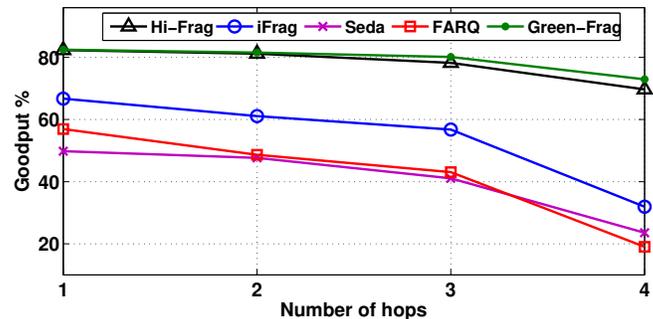


Fig. 15: Goodput of Green-Frag, Hi-Frag, iFrag, Seda, and FARQ while varying the number of hops from 1 to 4

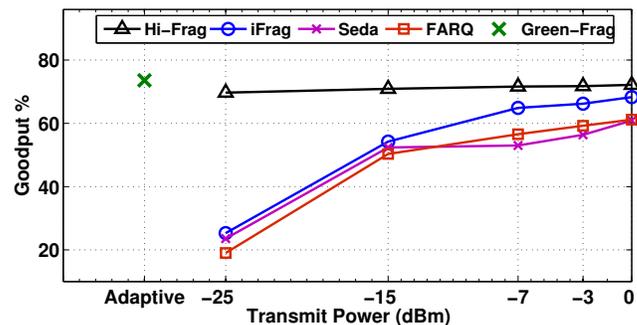


Fig. 16: Goodput of Green-Frag, Hi-Frag, iFrag, Seda, and FARQ while varying the transmit power and fixing the number of hops to 4

schemes, exhibits poor power usage if configured with sub-optimal power settings. Green-Frag saves energy by allowing the sender to choose the optimal transmit power.

2) *Goodput and Delay Evaluation:* In this section, we evaluate the goodput of FARQ, Seda, iFrag, Hi-Frag, and Green-Frag. Eq. 13 has been used to calculate the goodput for all the schemes. We will start by showing the single-hop results and then show the multi-hop ones.

Using only one single sender and receiver, we compare the goodput of Green-Frag to Hi-Frag, iFrag, Seda, and FARQ while varying the transmit power. Results are shown in Fig. 12. Hi-Frag achieves the highest goodput compared to iFrag, Seda, and FARQ. Besides being the most energy efficient among all other schemes, Green-Frag also achieves achieves 81% of goodput which is almost the same goodput as Hi-Frag transmitting with the maximum power. Fig. 13 shows the results of the same experiment while imposing Wi-Fi interference on the channel. Again, Hi-Frag maintains higher goodput compared to other schemes. This is due to the way that Hi-Frag dynamically selects the best blocks sizes and arrangements according to the channel condition. Green-Frag, on the other hand, achieves 10% less goodput compared to Hi-Frag transmitting with the maximum power. In fact, this drop in goodput is considered acceptable with more than 50% savings in energy. Moreover, Hi-Frag does not have a way to determine that using the maximum power is the best for this channel condition.

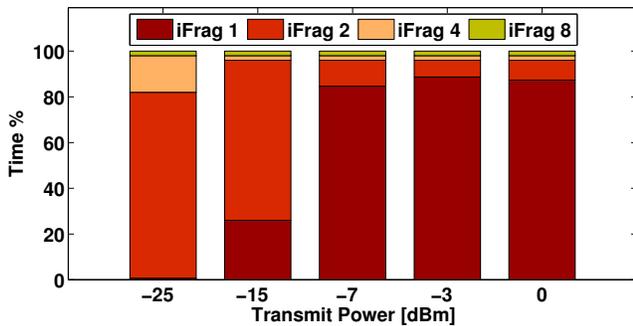


Fig. 17: Percentage of time iFrag spends in each mode in normal channel conditions

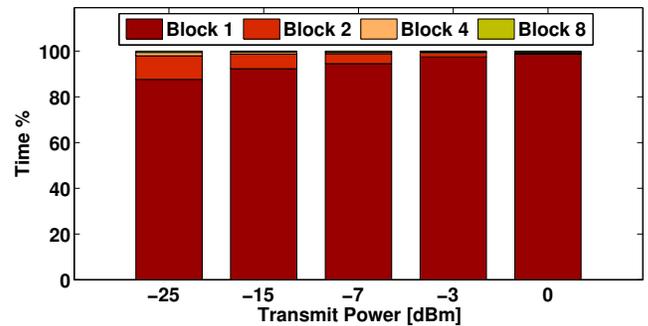


Fig. 19: Percentage of time Hi-Frag spends in each mode in normal channel conditions

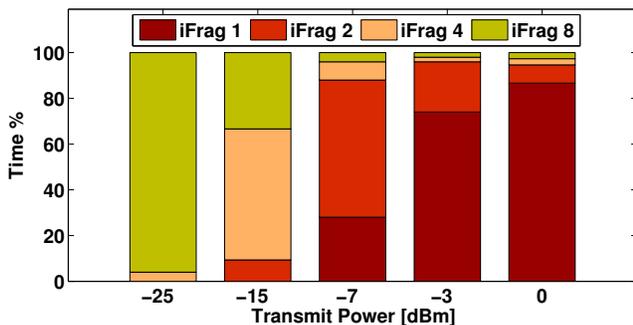


Fig. 18: Percentage of time iFrag spends in each mode in bad channel conditions

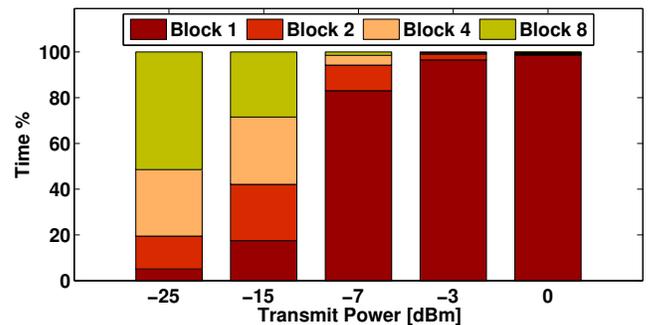


Fig. 20: Percentage of time Hi-Frag spends in each mode in bad channel conditions

Next, we compare the network end-to-end delay required to send 1000 frames under high level of Wi-Fi interference and various transmit powers. Fig. 14 shows the normalized end-to-end delay of Green-Frag, Hi-Frag, and iFrag with respect to Seda. Both Hi-Frag and iFrag achieve their peak performance at -25 dBm in which they managed to reduce the delay to only 16% and 28% respectively, compared to Seda. Hi-Frag achieves better delay than iFrag across all transmit powers. On average, Green-Frag achieves 22% reduction in delay compared to Seda. This is slightly higher than Hi-Frag best result. Although those schemes are not targeting network delay reduction, it is interesting to see that they perform better than Seda when the channel is suffering from interference. This improvement is attributed to the fact that the receiver in these schemes retransmits the ACK in case it has been lost or corrupted. As discussed earlier, this ACK is smaller in size than the data frame, and hence easier to be delivered correctly. As a result, the sender will trigger a new session of data sending as soon as it receives a recovery frame. Alternatively, the sender in Seda will wait for a predefined amount of time before sending the already sent data in case of ACK loss or corruption.

Finally, we evaluate all the schemes in terms of goodput using multi-hop topology. In the first experiment we vary the number of hops from 1 to 4 and fix the transmit power to the minimum supported value (-25 dBm) in order to minimize inter-node interference. Fig. 15 shows that both Hi-Frag and Green-Frag achieves the highest goodput over all hops. Green-

Frag has almost the same goodput as Hi-Frag in 1 hop and 2 hops and this due to the low interference. As the number of hops increases, Green-Frag outperforms Hi-Frag in terms of goodput. After that, we fix the number of hops of 4 and vary the transmit power. Fig. 16 shows that dynamic frame fragmentation protocols are outperforming the static ones in all cases. Green-Frag with the adaptive power mechanism provides more goodput than Hi-Frag across all transmit powers. This shows that even in multi-hop scenarios Green-Frag can still provide better goodput while saving energy.

3) *Behavioral Analysis:* In this section, we analyze how Green-Frag behaves over time and compare its behaviour to other dynamic partial packet recovery schemes such as iFrag and Hi-Frag. Lets start by showing the behaviour of iFrag. Fig. 17 represents the percentage of time iFrag spends in each mode. Noticeably, iFrag spends around 10% of its time in iFrag 2 when transmitting at high power. The main reason for this is the selection mechanism of threshold for switching between iFrag 1 and iFrag 2. In fact, iFrag needs the block reception ratio to be 100% in order to switch to iFrag 1, forcing iFrag to switch to iFrag 2 after each session with one or more lost packets. Also, it could be noticed that as the transmit power gets lower, more time is spent in smaller block sizes. This is due to higher bit error rates. In Fig. 18, we repeat the same thing while imposing W-Fi interference. iFrag spends more than 80% of the time in iFrag 1 when the channel quality is high. Alternatively, iFrag never switches back to iFrag 1 when the transmission power is -25 dBm or -15 dBm since

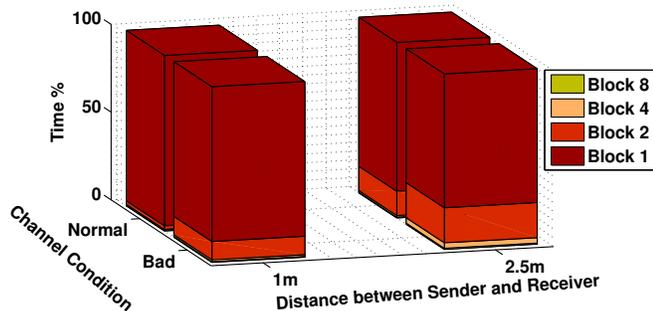


Fig. 21: Percentage of time Green-Frag spends in each block mode

the channel is considered very noisy.

Now we are going to repeat the exercise with Hi-Frag. Fig. 19 shows the percentage of time Hi-Frag spends in each mode. Unlike iFrag, Hi-Frag does not depend on predefined thresholds and can change block sizes faster. Therefore, Hi-Frag spends more time using blocks with bigger sizes which lowers the overhead. This explains why Hi-Frag always achieves more goodput compared to iFrag. Also, it could be noticed that there is a small number of 8 blocks in all transmit powers. This is because Hi-Frag starts with an 8-block frame structure then quickly adjusts itself to suite the channel requirements. The percentage of time Hi-Frag spends in each mode while imposing Wi-Fi interference is shown in Fig. 20. It is clear that Hi-Frag is now spending significantly longer time using smaller block sizes. When compared to iFrag, it could be noticed that Hi-Frag changes more frequently. Therefore, a wrong decision could be corrected fairly quickly.

We then presents the percentage of time Green-Frag spends in each block mode. This help evaluating the behaviour of the Green-Frag under various channel conditions. As shown in Fig. 21, Green-Frag spends most of the time using Block 1 mode. This was a design consideration, since bigger blocks have less overhead, leading to lower energy consumption. Every time the frame fragmentation technique fails to use larger sized blocks, the adaptive power mechanism increases the transmit power. Taking the previous four figures into consideration, it is clear that Green-Frag behaviour of partitioning is radically different from both iFrag and Hi-Frag. This is in spite the fact that Green-Frag is using the same Hi-Frag frame structures. In fact, Green-Frag can rise the transmit power each time the frame fragmentation technique fails to reduce the overhead.

Since Green-Frag is the only scheme that sets the transmit power adaptively, it would be interesting to evaluate this part of the protocol under various channel conditions. Fig. 22 represent the percentage of time Green-Frag spends in each transmit power. Noticeably, Green-Frag spends most of the time transmitting at low power. In these experiments, Green-Frag never used the maximum transmit power (0 dBm). Also, it did not use -3 dBm transmit power except under very high interference. Also, it is clear from this figure that any increase in the interference level or sender/receiver separation makes Green-Frag spend more time at higher transmit powers. For example, Green-Frag spends most of the time transmitting at

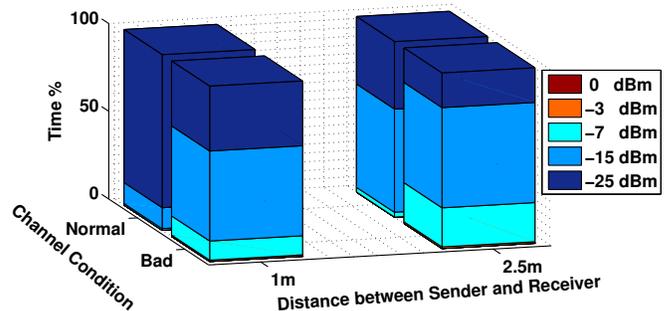


Fig. 22: Percentage of time Green-Frag spends in each transmit power

-25 dBm when there is no interference and when the motes are placed 1m apart from each other. Alternatively, when the distance is changed to 2.5m or the interference increases, Green-Frag spends most of the time transmitting at -15 dBm.

Finally, we study the adaptive power behaviour of Green-Frag over time. Fig. 23 shows Green-Frag transitions between various power levels according to channel quality. In this experiment, we placed both the sender mote as well as the receiver mote very close to the antenna of the Wi-Fi source. We start with very high interference, then we gradually reduce it until the interference source is turned off. The results show that Green-Frag transitions between various power level occur gradually. Green-Frag always starts transmitting using -7 dBm. Because of the high interference level at the beginning of the experiment, Green-Frag uses the maximum transmit power. However, as the channel quality starts improving, the protocol lowers the transmit power until it reaches the minimum transmit power when the interference source is completely removed.

VI. CONCLUSION AND FUTURE WORK

The main goal of partial packet recovery techniques is to achieve high throughput by harnessing retransmissions. Using our wireless sensor network testbed, we compare several partial packet recovery techniques proposed the literature from an energy perspective and find that these techniques are not very energy efficient and there is still room for improvement. In reality, the use of such methods might be infeasible in wireless sensor networks where the nodes are mostly battery powered and the energy is a scarce resource.

In this paper, we propose Green-Frag, a novel energy-efficient scheme that combines partial packet recovery with adaptive RF power selection. We present various design considerations behind Green-Frag that lowers energy consumption by minimizing the overhead. Green-Frag gives sensor nodes the ability to transmit data with optimal transmitting power and optimal frame structure based on the channel quality. This design results in lowering motes energy consumption significantly.

Green-Frag's energy performance is compared to four different partial packet recovery schemes in the literature, namely Hi-Frag, iFrag, Seda, and FARQ. Experimental evaluation on our TelosB motes testbed shows that Green-Frag outperforms these other schemes in term of energy efficiency over various

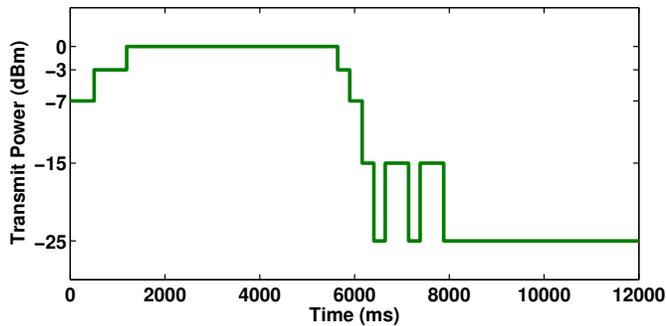


Fig. 23: Green-Frag transitions between TX-Powers over time

channel conditions. When the channel quality is low, Green-Frag lowers the energy consumption by an average of 33% compared to Hi-Frag, which is the most energy efficient partial packet recovery scheme in the literature. This saving in energy comes at the cost of 10% drop in goddput in the worst case. Moreover, Green-Frag reduces the end-to-end delay by 22% compared to Seda average delay.

There are interesting avenues for further work in this area. Green-Frag design can be improved by using machine learning techniques to estimate frame errors to determine the best transmit power and frame structure. Moreover, the concepts behind Green-Frag may be extended to IEEE 802.11/b/g/n networks for enhancing the overall energy performance in this kind of networks.

REFERENCES

- [1] A. Daghistani and B. Shihada, "Green-Frag: Energy-efficient frame fragmentation scheme for wireless sensor networks," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013.
- [2] K. Jamshaid, "Centralized Rate Allocation and Control in 802.11-based Wireless Mesh Networks," Ph.D. dissertation, University of Waterloo, Jan. 2010.
- [3] R. K. Ganti, P. Jayachandran, H. Luo, and T. F. Abdelzaher, "Datalink streaming in wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys, 2006, pp. 209–222.
- [4] A. Miu, H. Balakrishnan, and C. E. Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, ser. MobiCom, 2005, pp. 16–30.
- [5] H. Dubois-Ferrière, D. Estrin, and M. Vetterli, "Packet combining in sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, ser. SenSys, 2005, pp. 102–115.
- [6] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM, 2004, pp. 121–132.
- [7] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks," in *Proceedings of the ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, ser. E-WIND, 2005, pp. 11–16.
- [8] Z. Luo, Z. Han, A.-s. Wong, and S. Qiu, "Idma-based cooperative partial packet recovery: principles and applications," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–14, 2012.
- [9] E. Modiano, "An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols," *Wirel. Netw.*, vol. 5, no. 4, pp. 279–286, Jul. 1999.
- [10] W. Dong, X. Liu, C. Chen, Y. He, G. Chen, Y. Liu, and J. Bu, "DPLC: Dynamic packet length control in wireless sensor networks," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1–9.
- [11] A. Showail, A. Elrasad, A. Meer, A. Daghistani, K. Jamshaid, and B. Shihada, "iFrag: interference-aware frame fragmentation scheme for wireless sensor networks," *Wireless Networks*, pp. 1–18, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s11276-014-0722-1>
- [12] A. Meer, A. Daghistani, and B. Shihada, "An energy efficient hybrid interference-resilient frame fragmentation for wireless sensor networks," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2015.
- [13] C.-F. Kuo, H.-W. Tseng, and A.-C. Pang, "A fragment-based retransmission scheme with QoS considerations for wireless networks," in *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, ser. IWCMC '07, 2007, pp. 225–230.
- [14] Y. Zhou and J. Wang, "Optimum subpacket transmission for hybrid ARQ systems," *Communications, IEEE Transactions on*, vol. 54, no. 5, pp. 934 – 942, May 2006.
- [15] A. Willig, "Memory-efficient segment-based packet-combining schemes in face of deadlines," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 338 –350, 2009.
- [16] Y. hua Zhu, H. Xu, K. kai Chi, and H. Hu, "Accumulating error-free frame blocks to improve throughput for IEEE 802.11-based WLAN," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 743 – 752, 2012.
- [17] S. Lin and P. Yu, "A hybrid arq scheme with parity retransmission for error control of satellite channels," *IEEE Transactions on Communications*, vol. 30, no. 7, pp. 1701 – 1719, 1982.
- [18] K. C.-J. Lin, N. Kushman, and D. Katabi, "ZipTx: Harnessing partial packets in 802.11 networks," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom, 2008, pp. 351–362.
- [19] J. Xie, W. Hu, and Z. Zhang, "Revisiting partial packet recovery in 802.11 wireless lans," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys, 2011, pp. 281–292.
- [20] Y. Sankarasubramaniam, I. Akyildiz, and S. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE, 2003 IEEE International Workshop on*, 2003, pp. 1 – 8.
- [21] W. Dong, X. Liu, C. Chen, Y. He, G. Chen, Y. Liu, and J. Bu, "DPLC: Dynamic packet length control in wireless sensor networks," in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.
- [22] M. Vuran and I. Akyildiz, "Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, pp. 226 –230.
- [23] W. Dong, J. Yu, and P. Zhang, "Exploiting error estimating codes for packet length adaptation in low-power wireless networks," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [24] 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*, 2005, pp. 364–369.
- [25] Texas Instruments, "Chipcon CC2420 Datasheet." <http://www.ti.com/lit/ds/symlink/cc2420.pdf/>.