

Green-Frag: Energy-Efficient Frame Fragmentation Scheme for Wireless Sensor Networks

Anas Daghistani, Basem Shihada

CEMSE Division, King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{anas.daghistani, basem.shihada}@kaust.edu.sa

Abstract—Frame fragmentation techniques aim to achieve higher throughput by reducing retransmissions. Using experiments on a WSN testbed, we show that frame fragmentation also helps to reduce energy consumption. In this paper we propose Green-Frag, a new energy-efficient protocol based on efficient frame fragmentation technique. Green-Frag allows sensor nodes to transmit data with optimal transmit power and frame structure based on environmental conditions. Green-Frag takes into consideration the channel conditions, interference patterns and level, as well as the distance between sender and receiver. The paper discusses various design and implementation considerations for Green-Frag. Using experimental evaluation on a sensor mote testbed, we show that Green-Frag achieves the least energy consumption by choosing the best transmit power according to the channel conditions.

Index Terms—Wireless Sensor Networks, Energy-Efficiency, Frame Fragmentation, Adaptive Power, Interference-Resistance

I. INTRODUCTION

Power management is an active area of research in wireless sensor networks (WSNs). Efficient power management is necessary because WSNs are battery-operated devices that can be deployed in mission-critical applications. We believe that an energy efficient protocol should have the ability to smartly tackle retransmissions, low throughput, optimal frame size, optimal transmit power, and interference issues. Based on prior literature, we found that frame fragmentation techniques can be the foundation of such a protocol. However, current frame fragmentation techniques have only focused on improving the throughput by decreasing retransmissions [1]. They achieve this by allowing the sender to retransmit only the corrupted portion of the previously transmitted frame.

In this paper, we choose recent frame fragmentation schemes and analyse them from the energy efficiency perspective. We experimentally compare two main types of frame fragmentation techniques, static and dynamic, in order to find the most energy-efficient scheme. We then propose a new energy-efficient scheme called Green-Frag that uses the most energy-efficient frame fragmentation technique as a foundation for its design. Green-Frag is an adaptive power and frame fragmentation scheme that takes advantage of environmental interference levels and patterns to decide the optimal frame structure and transmit power. This new protocol aims to achieve high level of energy efficiency in all channel situations.

This paper is organized as follows. Section II provides a brief description of the related work that we compare. Section III shows the motivation of the work. In Section IV, we

introduce Green-Frag and various considerations governing its design. Section V presents our experimental results. Finally, the paper is concluded in Section VI.

II. RELATED WORK

Green-Frag is the first partial packet recovery techniques that uses adaptive transmit power control. Most of the prior partial packet recovery techniques aim for increasing the throughput without studying their impact on energy consumption. To the best of our knowledge, our paper is the first experimental study of frame fragmentation techniques from the energy consumption perspective. In this section we briefly describe the other schemes that we use in Sect. III to find an energy-efficient frame fragmentation technique and in Sect. V for performance benchmarks.

Frame fragmentation is one of the partial packet recovery approaches. Frame fragmentation techniques can be classified as either static [2], [3] or dynamic [4], [5], [6], [7], [8], based on whether they use fixed or dynamic frame sizes. One of the main static frame fragmentation technique is called Seda [2]. Its goal is to enhance throughput by reducing the number of retransmissions. Its design criterion includes a number of enhancements that can reduce energy consumption, such as reduced retransmissions, small number of acknowledgment (ACK) frames, and improved system throughput. 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. They claim that a block size of 20-25 bytes provides near-optimal throughput. However, this is not always correct because it is highly depend on the channel conditions and bit error ratio (BER).

iFrag [4] and Hi-Frag [5] are two recent dynamic frame fragmentation schemes. iFrag [4] is an interference-aware frame fragmentation scheme. It is dynamically changing the partitioning size of frames depending on transmission history and some predefined thresholds. It has a predefined data frame modes, each of them partitioned differently. The modes with smaller block sizes have higher overhead. Hi-Frag [5] is a hybrid interference-resilient frame fragmentation scheme. It takes the interference level and patterns into consideration while considering frames structure. Unlike iFrag [4], it dynamically divides frames to blocks according to observed error patterns. Also, its frames can contains blocks with different sizes. It have a mechanism that reduces blocks overhead.

III. MOTIVATION

1) *Energy Performance Comparison:* We study the energy efficiency of frame fragmentation techniques using several experiments under various channel conditions and radio transmit power. Our experiments compare FARQ, Seda, iFrag, and Hi-Frag schemes' energy performance. We evaluate the energy consumed for delivering one useful bit. The experiments were performed under two situations: normal and bad channel conditions. The bad channel conditions are achieved by imposing high Wi-Fi interference. Experimental setup details, equations, and calculations used in this study are described in Sect. V.

In Fig. 1, we show the energy performance of Hi-Frag, iFrag, Seda, and FARQ schemes with various transmit powers under normal channel conditions. Hi-Frag outperforms all other schemes. It consumes the least energy per useful bit. Clearly, it also consumes unnecessary energy when transmitting at high power. Overall, Seda has the worst performance.

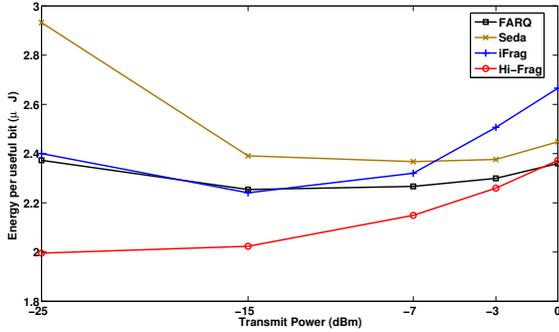


Fig. 1: Energy per useful bit in normal channel conditions

Figure 2 shows the energy per useful bit results in bad channel conditions with high Wi-Fi interference. Hi-Frag maintains the best energy performance at all transmit powers. Its energy per useful bit results varies between 2.3 and 5.2 μJ . FARQ shows the worst results in these channel conditions.

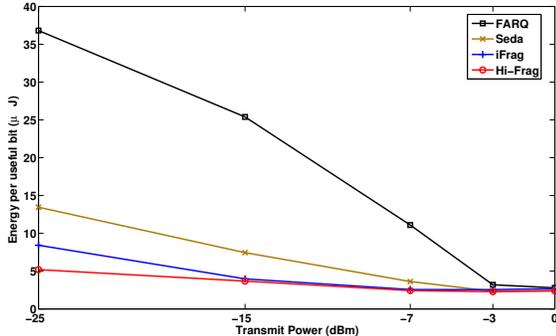


Fig. 2: Energy per useful bit in bad channel conditions

To conclude this comparison, Hi-Frag is the most energy-efficient frame fragmentation scheme under all channel conditions and transmit powers. It achieved an average of 16%

improvement than static frame fragmentation scheme (Seda) in normal channel conditions and an average of 11% compared to other dynamic scheme (iFrag). In bad channel conditions, Hi-Frag shows an average of 49% less power consumption than Seda and 23% compared to iFrag. However, Hi-Frag can achieve even better performance if it has a mechanism to choose the optimal transmit power, especially under normal channel conditions. This paper proposes a new scheme called Green-Frag that uses Hi-Frag principles for frame fragmentation and combines them with transmit power adaptation.

2) *Power Adaptation:* It is important to have a mechanism to select a transmit power that optimizes the energy usage. We prove that a power adaptive mechanism can dramatically improve network overall energy consumption. 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 PDF of each node's location at a certain distance d_i is illustrated by:

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

Therefore, the average power consumption without implementing the power adaptive mechanism is equal to,

$$\begin{aligned} P_{ave_{without}} &= \sum_{i=1}^N (P(d_{max}) Pr\{D_{S_i} = d_i\}) \\ &= P(d_{max}) \frac{N}{d_{max} - d_{min}} \end{aligned} \quad (2)$$

On the other hand, the average power consumption with the power adaptive mechanism is equal to the following,

$$\begin{aligned} P_{ave_{with}} &= \sum_{i=1}^N (P(d_i) Pr\{D_{S_i} = d_i\}) \\ &= \sum_{i=1}^N P(d_i) \frac{1}{d_{max} - d_{min}} \end{aligned} \quad (3)$$

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

$$\begin{aligned} P_{save} &= P_{ave_{without}} - P_{ave_{with}} \\ &= Pr\{D_{S_i} = d_i\} [NP(d_{max}) - \sum_{i=1}^N P(d_i)] \end{aligned} \quad (4)$$

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

IV. GREEN-FRAG SCHEME

In this section we provide an overview of Green-Frag. We discuss its design considerations and describe how main design challenges were addressed.

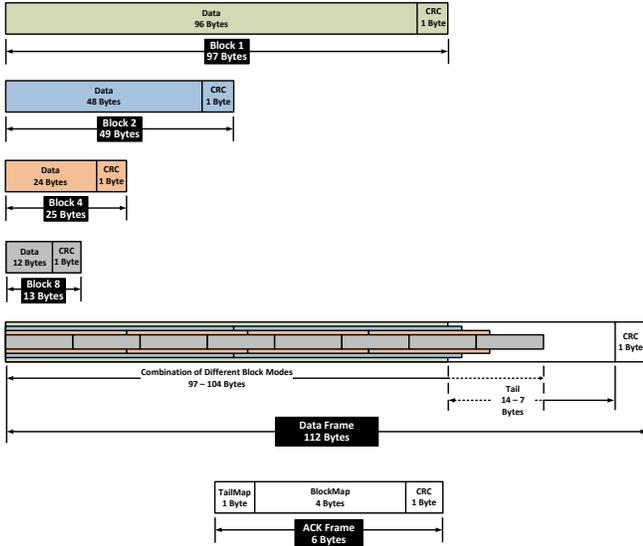


Fig. 3: Green-Frag frame structures

A. Overview

Green-Frag reduces energy consumed in delivering useful data under all channel conditions by combining a frame fragmentation technique and an adaptive power mechanism. 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.

B. 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 TelosB hardware. An additional 16 bytes are used by the MAC and PHY layers. Each frame could 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. 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

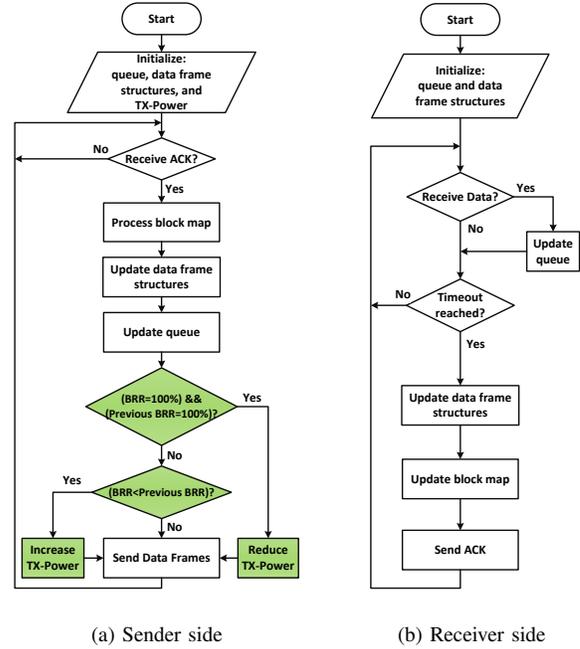


Fig. 4: Green-Frag operation flow chart

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. 5 bits are used from the 1 byte TailMap field: 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 with 4 bytes size is enough for all Green-Frag block combinations 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.

C. 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 needs a retransmission. 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 better to resend a

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
BlockMap field of the ACK containing missed bytes and new data bytes
23   Construct Tail to reach 112 bytes of payload
24   Add CRC to blocks and tails, reframe and handoff packet to MAC-layer
```

small ACK than retransmitting the large data frame. Also, if the sender retransmits the previous session data, there is a high chance that some of the blocks received correctly and will be sent again. Therefore, sending data frames in such situations have high probability of wasting huge amount of energy.

Each frame can have a different structure of how blocks are arranged. Both the sender and the receiver update these structures at the end of each session. Blocks can be divided into smaller blocks or merged into larger blocks. The receiver updates the frame structure depending on blocks' CRCs and specify their status in the **BlockMap**. On the other hand, the sender updates the frame structure after receiving an ACK depending on its **BlockMap** field. As mentioned before, the sender waits for an ACK with the correct Color bit to distinguish between retransmitted ACKs. Thus, the sender and the receiver work using 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 $timeout_{end}$. Green-Frag sender and receiver operations are described in Algorithm 1 and 2 respectively.

D. Discussion

Green-Frag reduces the energy consumed in delivering useful data by dynamically choosing blocks sizes, frame structure, and transmit power. This raises multiple challenges and design considerations. One challenge is to know the best way of switching between different block modes that guarantee delivering useful data with the minimal power. Based on our analysis in Sect. III, Green-Frag uses Hi-Frag as a basis for its

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
network layer
17  else
18    Buffer correctly received blocks
19  construct ACK contains updated BlockMap and Color
20  if (session received) || (timeout is fired) then
21    Send ACK
22  Stop sending ACK when End Message received or  $timeout_{end}$  reached
```

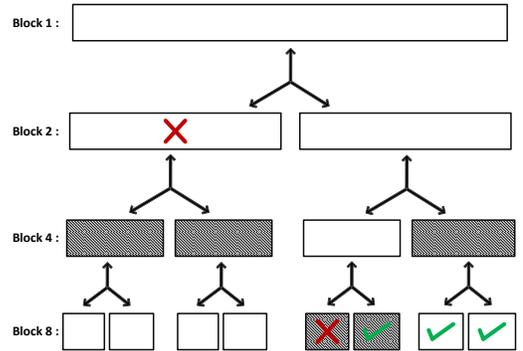


Fig. 5: The possible combinations of block modes in a frame

frame fragmentation technique and adapts it to make it work with dynamic transmit power control.

Green-Frag uses simple rules to change frame structures. The data frame structure is adapted based on the 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 sending with additional overhead is better than being optimistic and losing bigger blocks of data.

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 savings are 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 Received Ratio (BRR) depending on the BlockMap field in the ACK. It consider 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 blocks, 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:

$$BRR = \frac{\text{Count of Correctly Received Blocks}}{32} * 100 \quad (5)$$

After calculating BRR, the sender decides on increasing, decreasing, or using the same transmit power. In our implementation, the transmit power used are: 0 dBm, -3 dBm, -7 dBm, -15 dBm, and -25 dBm. 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 fragmentation techniques also adjust the frame structure according to the new situation. Sender reduces the transmit power only when both the BRR and previous session BRR equal 100%. Otherwise, it uses the same transmit power. This allows frame fragmentation techniques to stabilize before changing the transmit power. This adaptive power adaptive mechanism minimizes overhead by making the sender transmit with the power that makes frame fragmentation technique use 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 infrequently compared to data frames.

V. PERFORMANCE EVALUATION

This section introduces the equations used to analyze the performance of the compared schemes. We also describe our experimental setup and performance measurement results.

A. Energy Efficiency Equations

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

Radio transmission and reception is a major source of energy consumption in sensor nodes. Thus, the equations below ignore energy spent by CPU, memory operations, etc.

Our energy efficiency comparison based on how much energy is consumed for delivering a useful bit. This need

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
$T_{tail_{RX}}$	Received tail 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

dividing consumed energy in sending and receiving all frames by number of useful received bits as Eq. 6 below shows.

$$\text{Energy Per Useful Bit} = \frac{\text{Consume Energy } (E)}{\text{Useful Bits } (UB)} \quad (6)$$

Below equations represent methods used to compute useful bits in FARQ, Seda, iFrag, Hi-Frag, and Green-Frag schemes respectively. Useful bits are bits in correctly received frames without counting frames headers and scheme overheads. Noticeably, Green-Frag and Hi-Frag have the same equation due to the fact that they are using similar frame structures.

$$UB_{(FARQ)} = F_{RX} * (F_L - H) \quad (7)$$

$$UB_{(Seda)} = B_{RX} * (B_L - B_H) \quad (8)$$

$$UB_{(iFrag)} = \sum_{m=1}^M (B(m)_{RX} * (B(m)_L - B_H)) \quad (9)$$

$$UB_{(Hi-Frag, Green-Frag)} = \sum_{m=1}^M (B(m)_{RX} * (B(m)_L - B_H)) + \sum_{i=1}^{F_{RX}} Tail(i)_{RX} \quad (10)$$

Energy consumed for transmission/reception is equal to the transmit/receive power consumption multiplied by time spend in sending/receiving frames. Sending time equals the receiving time because even receiving corrupted frames consumes power. Eq. 11 shows the method used to calculate the energy consumed by FARQ, Seda, and Hi-Frag, while Eq. 12 and 13 represent iFrag and Green-Frag, respectively. iFrag has a separate equation because it uses data frames with several sizes, while Green-Frag's Eq. is different due to its adaptive transmit power. In Green-Frag's Eq. 13, $F(p)_{TX}$ represents the number of transmitted frames with transmit power p .

$$E_{(FARQ, Seda, Hi-Frag)} = (P_{TX} + P_{RX})[(F_{TX} * T_F) + (ACK_{TX} * T_{ACK})] \quad (11)$$

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
<hr/>		
Radio Receiving:	19.7	56.539

TABLE II: TelosB power consumption of transmitting with various powers and receiving

$$E_{(iFrag)} = (P_{TX} + P_{RX}) \left[\sum_{m=1}^M (F(m)_{TX} * T(m)_F) + (ACK_{TX} * T_{ACK}) \right] \quad (12)$$

$$E_{(GreenFrag)} = \sum_{p=P(min)_{TX}}^{P(max)_{TX}} [(P(p)_{TX} + P_{RX})(F(p)_{TX} * T_F)] + (P(max)_{TX} + P_{RX})(ACK_{TX} * T_{ACK}) \quad (13)$$

B. Experimentation Setup

We use TelosB [9] motes with TinyOS 2.1.1 platform in an office environment. These motes use Chipcon-CC2420 radio [10] (2.4 GHz band) compatible with IEEE 802.15.4 (ZigBee) standard. Our experiments were performed at night to limit uncontrolled interference with campus Wi-Fi network.

The experimental setup consists of a sender and a receiver mote separated by a distance of 1m and powered via USB to avoid low battery power. MAC-layer automatic CRC is disabled to allow the reception of partially corrupted packets. All experiments were repeated under two channel conditions: (1) normal interference conditions, and (2) interference from two Linux machines 15m apart, transferring a large file using IEEE 802.11g cards with 18 dBm transmit power. We used Wi-Fi for interference because it is the main interference source in many WSN deployments such as smart buildings and traffic control applications.

C. Experimental Results

Several experiments were carried out in order to evaluate the energy consumption of FARQ, Seda, iFrag, Hi-Frag, and Green-Frag schemes. First, we computed TelosB transmit and receive power consumption by multiplying CC2420 radio transceiver current consumption of transmitting and receiving [10] with TelosB operation voltage. The operation voltage were measured and found to be constant at 2.87 V because motes were powered by USB. Table II shows the power consumption of reception and transmission at various powers.

We then executed multiple experiments to find average time spent to send a data or ACK frame in the compared schemes. FARQ implementation is similar to Seda, except that it uses one data block per frame. We averaged the time of sending 3000 data frames and around 750 ACK frames in three different settings. A summary of our results are shown

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 the compared schemes

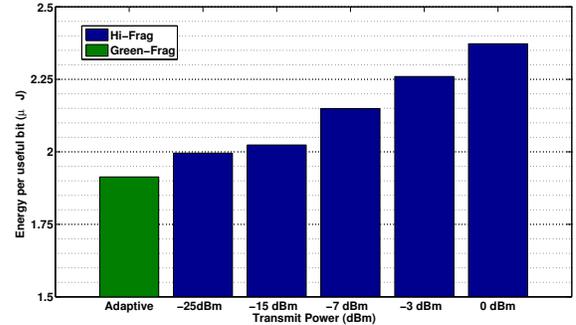


Fig. 6: Energy per useful bit of Green-Frag vs. Hi-Frag in normal channel conditions

in Table III. Green-Frag and Hi-Frag have almost the same times because they are using similar frame structures.

TinyOS does not provide the time spent in transmission. We implemented a mechanism to measure the time when the send command occurs until the transmission finishes, which is shown in Table III. Other than the transmit time, these values also include time spent 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.

To substitute in Equation 6, we also need to find received useful bits for each scheme. The experimental results that compare FARQ, Seda, iFrag and Hi-Frag were shown in Sect. III. These results motivated us to use Hi-Frag frame fragmentation principles as a foundation for Green-Frag.

In Fig. 6, Green-Frag is compared with Hi-Frag in terms of energy consumption of delivering a useful bit under normal channel conditions. Hi-Frag experiments were performed multiple times with different transmit power, while Green-Frag controls transmit power adaptively. Green-Frag outperforms Hi-Frag because it changes the transmit power depending on the channel changes. These results show that none of the transmit powers is optimal at all times. The optimal transmit power changes with channel conditions. Green-Frag spends most of the time using -25 dBm transmit power; however, it also uses other power values as shown in subsequent figures. Green-Frag saves about 20% of energy compared to Hi-Frag transmitting at maximum power. Additionally, Green-Frag is

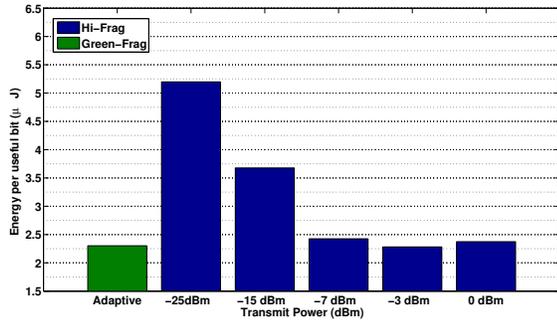


Fig. 7: Energy per useful bit of Green-Frag vs. Hi-Frag in bad channel conditions

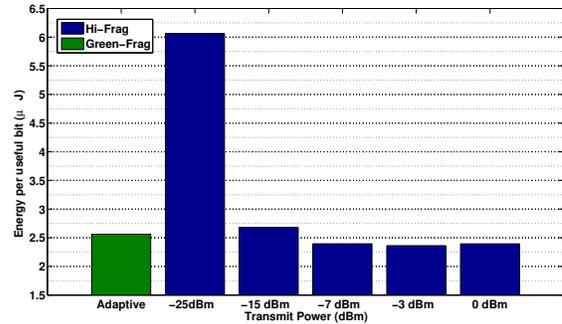


Fig. 9: Energy per useful bit of Green-Frag vs. Hi-Frag in bad channel conditions, while the distance is 2.5m

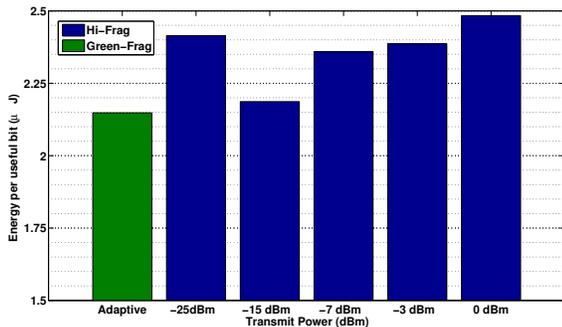


Fig. 8: Energy per useful bit of Green-Frag vs. Hi-Frag in normal channel conditions, while the distance is 2.5m

better than Hi-Frag by an average of 10% compared to the results across all transmit powers.

In Fig. 7, the experiment was repeated under bad channel conditions (with active Wi-Fi interference). Green-Frag achieves a performance that is almost the same as the best results of Hi-Frag while 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. 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 results 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, Hi-Frag's best performance is achieved when transmitting at -3 dBm, unlike under normal channel conditions when the best transmit power is -25 dBm. This value (-25dBm) is the worst choice for transmit power in noisy channel conditions.

We repeated the same experiments with different distances and interference patterns to fully evaluate Green-Frag. In this set of experiments, the motes were placed 2.5m apart from each other and the receiver moved farther from the interference source. Therefore, the transmitter/receiver separation increased while the interference level decreased. Fig. 8 shows the energy consumption per useful bit for Green-Frag in comparison to Hi-Frag under normal channel conditions. Similarly, Green-

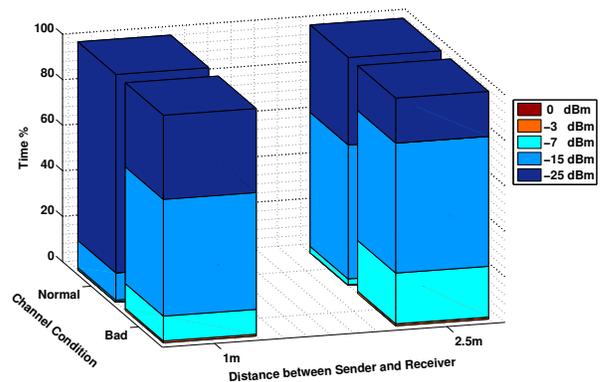


Fig. 10: Percentage of time Green-Frag is spending in each transmit power

Frag shows the best energy performance compared to all Hi-Frag results at constant transmit power. Green-Frag saves about 14% of the energy compared to Hi-Frag transmitting at maximum power. Green-Frag is better than Hi-Frag by an average of 9% across all transmit power levels.

Fig. 9 presents energy per useful bit results of Green-Frag compared to Hi-Frag in bad channel conditions with motes placed 2.5m apart. In this experiment, Green-Frag gets stuck in a sub-optimal transmit power. This is because Green-Frag design is slow in increasing the transmit power under bad channel situations. Green-Frag achieves slightly higher power consumption in comparison to the best Hi-Frag results. Green-Frag reduces energy consumption by 58% when compared to the worst transmit power used by Hi-Frag.

To conclude, there is no a priori-known optimal transmit that can be used by motes. It varies based on the motes' separation as well as interference levels. Hi-Frag exhibits poor power usage if configured with sub-optimal power settings. Green-Frag changes the transmit power and frames structure at runtime based on channel conditions.

Fig. 10 represent the percentage of time Green-Frag spends in each transmit power. It shows the behavior changes under varying channel conditions. Noticeably, Green-Frag spends most of the time transmitting at low power. In these exper-

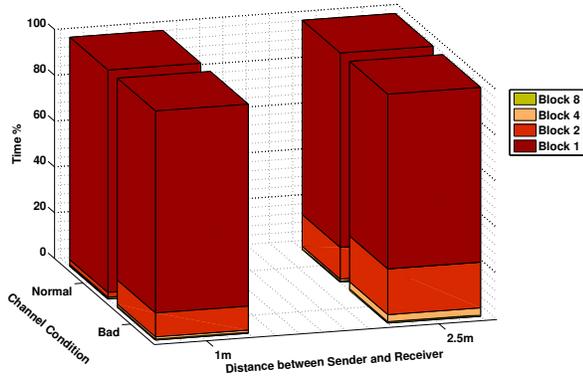


Fig. 11: Percentage of time Green-Frag is spending in each block mode

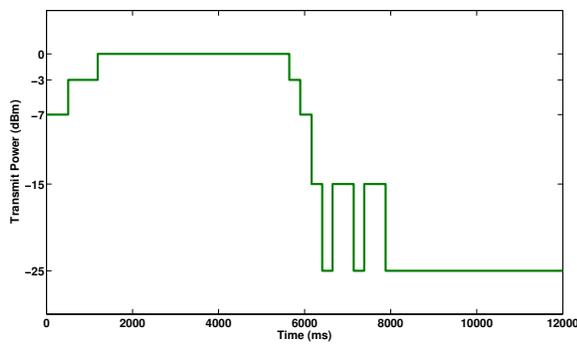


Fig. 12: Green-Frag transitions between TX-Powers over time

iments, Green-Frag never used the maximum transmit power (0 dBm). Also, it did not use -3 dBm transmit power except under bad channel conditions. Increase in interference level or increasing sender/receiver separation makes Green-Frag spend more time at higher transmit powers. Green-Frag spends most of the time transmitting at -25 dBm under normal channel conditions and when the nodes placed 1m apart from each other. However, when the distance changed to 2.5m or the interference increases, Green-Frag spends most of the time transmitting at -15 dBm.

Fig. 11 shows the percentage of time Green-Frag spends in each block mode. It evaluates the behavior of the Green-Frag frame fragmentation technique. 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 large sized blocks, the adaptive power mechanism increases the transmit power.

We studied the adaptive power behavior of Green-Frag over time. Fig. 12 shows Green-Frag transitions between transmit power levels according to channel conditions. In this experiment, we placed both nodes next to the antenna of the interference source. At the beginning, we imposed very high interference. We then gradually reduced this interfer-

ence, until the interference source is completely removed. The result shows that Green-Frag transitions occur gradually. Green-Frag always starts transmitting using -7 dBm output power. Because of the high interference level at the beginning, Green-Frag uses the maximum transmit power. However, when the channel quality starts improving, the protocol lowers the transmit power. It used the minimum transmit power when the interference source is removed.

VI. CONCLUSIONS

In this paper, we proposed Green-Frag, a new energy-efficient scheme that combines frame fragmentation with adaptive transmit power mechanism. Green-Frag design is a result of studying and comparing previously proposed frame fragmentation schemes from the energy perspective. We presented the various design considerations behind Green-Frag that improve energy consumption and lower the overhead. Green-Frag gives sensor nodes the ability to transmit data with optimal transmitting power and optimal frame structure based on the channel conditions.

Green-Frag's implementation and energy performance was compared with Hi-Frag, iFrag, Seda, and FARQ schemes. Experimental results show that Green-Frag outperforms these other schemes. It chooses the best transmit power according to the channel conditions. Subsequently, Green-Frag achieves the least energy consumption in all environment situations.

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 structures. Finally, the concepts behind Green-Frag may be extended to IEEE 802.11/b/g/n networks for enhancing their energy performance.

REFERENCES

- [1] 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*, no. 1, 2012.
- [2] 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.
- [3] 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.
- [4] A. Showail, A. Elrasad, A. Meer, A. Daghistani, K. Jamshaid, and B. Shihada, "iFrag: Interference-aware frame fragmentation scheme for wireless sensor networks," *Submitted to ACM Wireless Networks Journal*, 2013.
- [5] A. Meer, A. Daghistani, and B. Shihada, "Hi-Frag: Hybrid interference-resilient frame fragmentation for wireless sensor networks," *Submitted to IEEE International Conference on Global Communications (GLOBECOM)*, 2013.
- [6] Y. Zhou and J. Wang, "Optimum subpacket transmission for hybrid ARQ systems," *IEEE Transactions on Communications*, 2006.
- [7] 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.
- [8] 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, 2012.
- [9] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Information Processing in Sensor Networks*, 2005.
- [10] Texas Instruments, "Chipcon CC2420 Datasheet." <http://www.ti.com/>.