Bursting: Increasing Energy Efficiency of Erasure-Coded Data in Animal-Borne Sensor Networks

Björn Cassens
TU Braunschweig
cassens@ibr.cs.tu-bs.de

Markus Hartmann
FAU Erlangen
markus.hartmann@fau.de

Thorsten Nowak
FAU Erlangen
thorsten.nowak@fau.de

Niklas Duda
FAU Erlangen
niklas.duda@fau.de

Jörn Thielecke
FAU Erlangen
joern.thielecke@fau.de

Alexander Kölpin
BTU Cottbus
alexander.koelpin@b-tu.de

Rüdiger Kapitza
TU Braunschweig
kapitza@ibr.cs.tu-bs.de

Abstract

Observing the (social) behavior of animals and especially bats contributes to improved epidemic forecasts and improved protection of rare or endangered species. For this task, animal-borne sensor nodes are powerful tools since they allow collecting more information at less influence to the species compared to traditional radio telemetry. However, as bats are inherently small vertebrates, a strict weight limit of 2 g is dictated for attached sensor nodes. To acquire enough data, the sensor nodes must achieve a runtime of at least two weeks, which demands a rigorous energy management. As the data transmission is the most energy hungry task, minimizing the energy demand in this segment is a key to achieve the desired runtimes.

In this paper we present a novel technique that enables bursts of erasure coded data without sacrificing reliability (in a wildlife setting). To keep data losses at a minimum, our approach exploits the characteristics of the erasure channel and combines encoded packets in smart way. We evaluated this approach extensively with experiments, field tests and a theoretical analysis. The results show a decreased energy demand by 30.177 % to 36.951 % while the data rate is increased up to a factor of 4 at the same time.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; J.4 [**Computer Applications**]: Social and behavioral science

General Terms

Experimentation, Measurement, Design, Energy

Keywords

Energy-aware systems, Communication, Firmware, Animal tracking, Wireless sensor networks

1 Introduction

Observing the behavior of animals is key to understand their social relations and enables epidemic forecasts [11]. While acquiring the behavioral data, the investigated animals should be exposed to disturbances as little as possible, to ensure an unbiased behavior. In our interdisciplinary research unit BATS [8], we focus on bats. Bats are in general a species that is hard to observe since they are highly mobile, active during night time and can often be found in difficult areas such as rain forests or caves. To minimize the impact on the individual animal and enable continuous observation, Wireless Sensor Networks (WSNs) have been utilized [22, 25, 12, 5]. By using miniaturized nodes, later called Mobile Nodes (MNs), encounters between individuals can be logged and automatically downloaded to a Base Station (BS) enabling an easy access to the data for the biologist.

Attaching a MN to small vertebrates such as bats comes along with tight and strict weight limits of only 2 g for a MN (including battery) [8]. Furthermore, a runtime of at least 1-2 weeks is a desired goal to collect enough data for in-depth insights of the behavior of an observed individual. In order to achieve the desired runtime replacing batteries (high impact on the individual) or installing larger batteries (not possible due to weight limit) is not an option, instead a rigorous power management is necessary. A fair share of the energy in such scenarios is required for communication between the MNs but communication becomes even more critical when the data is downloaded to the BS [23, 1]. We found out, that communication to a BS can consume a significant part of the overall energy budget, depending on the behavior of the bat and group dynamics. In our project, new hardware is developed which acquires more sensor data like acceleration, magnet field and air-pressure at a high rate. Additionally, 60

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

bats should be supported to study big groups of bats, which also increases the amount of data. Together, this puts a high pressure on reliable, fast and energy aware data transmission since an increase of data of up to 150 times is expected. With this expected increase of data, more than 20 % of stored energy is spend for data transmission to a BS, which in turn decreases the runtime by two days. Therefore, optimizing the data transmission is attractive to prolong the runtime.

Data download to a BS is particular challenging, as a bat can reach high median velocities of up to 14.3 m/s [13] whereas the BS is stationary. Relying on an acknowledgment-based protocol would incur a high number of retransmissions due to missed acknowledgment packets. Therefore, a unidirectional protocol has been proposed by Mutschlechner et al. [7], which saves energy and sends data in an opportunistic way. As packet loss rates are high, an Erasure Code (EC) is used, to keep data reception rates high at moderate overhead in terms of energy. This is achieved by sending original data and redundant packets, where only a subset of send data must be received to recover all data. However, an EC is only effective if packet loss does not occur in bursts and failures are distributed uniformly.

In order to increase energy efficiency batching of data is a common technique [10, 6, 4] and can decrease the energy demand significantly. However, increasing payload size of a (wirelessly) transmitted packet also increases the likelihood of a corrupted packet. Also sending multiple packets at once renders the EC useless as errors inside the header also cause the loss of other packets inside the containing frame and as a consequence higher data losses. This behavior is dictated by the manufacturer of the transceiver. Using software defined radios are able to handle bit errors inside frame headers which incurs a dramatically increased energy demand resulting in short run times.

In this paper, we combine both techniques, batching and EC to gain higher energy efficiency compared to the approach presented by Mutschlechner et al. while not sacrifice data reception rates. We apply a novel technique, we call *bursting* in which multiple packets are scrambled in a smart way to achieve independence of multiple erasure-coded data. Besides an in-depth explanation on how our approach works, we answer the following questions in this paper:

1. How to model packet and data losses in an efficient and realistic way for our scenario?
2. How does batching affect reliability and data reception rate when used in combination with EC?
3. How much energy can be saved with both approaches and compared to the method, proposed by Mutschlechner et al.?

To answer the first question, we modeled packet losses and data losses with the aid of Markov models. We used a simulator which bases on those and compared the results against real world deployments. In order to answer the second question, we conducted multiple simulations which shows that batching in a naive way receives only 20 % of send packets. The state of the art and the improved approach, however, receives 55 % of send packets at the same bit error rate of 0.45 %. This is amplified due to the ECs as we discovered that the data reception rate is only 20 % for the naive and 68 % for our approach. We modeled the energy demand energy demand for sending data to a BS, thus answering the third question. Our energy model is, again, compared to measurements with a high precision source meter and shows only negligible errors of up to 4.57 %. With this model, we analyzed potential energy savings up to 37 % for our approach compared to the state of the art.

The paper is organized as following, in Section 2 the background and normal operation of our system is presented. Furthermore, the energy demands for our application is discussed in Section 3. Section 4 related systems and protocols are discussed. As we focus on data transmissions to a BS, the communication scheme, frame layout and erasure coding is explained in detail in Section 5. For batching data, the naive and our approach is presented and compared in Section 6. As packet losses are not necessarily translate directly to data losses due to the erasure code, used models and the results of our simulation is discussed in Section 7. In Section 8, the energy demand and energy savings are analyzed in detail. A realistic estimate of bit errors is performed in Section 9. Based on the bit error rate, the naive batching and bursting are compared in Section 10. A practical evaluation of our approach is presented in Section 11. In order to get a realistic estimate on energy savings, a real deployment has been performed and is explained in Section 12 and Section 13 concludes our paper.

2 Background on Bats Tracking

The goal of our work is to track bats as they form the second largest group of mammals with more than 1000 species. Additionally, the majority is living in groups whereas the group sizes and social systems vary depending on the species. Even though detailed knowledge on foraging strategies is scarce and evidence is increasing that social interactions contribute to foraging success. This alone makes (social) behavior studies attractive but also last incidents showed bats, which were involved in spillovers to humans and livestock [9]. Developing conservation strategies requires knowledge on the interdependencies and dynamics in bat populations. Also deriving models for infectious diseases from this data is key to prevent future outbreaks. Therefore, bats are one of the best animals for testing next generation wildlife tracking systems.

Tracking bats rises new challenges, as the MN is not allowed to exceed 10 % of the body weight of the observed animal [2]. In Germany, the largest bat species exhibit an average body weight of 20 g, resulting in a weight limit of 2 g for our MN. The used hardware, weights in sum 1.3 g including battery, housing and circuitry. 0.3 g is spend for the battery which provides an energy budget of 22 mAh which translates to 0.08 Wh. The minimum runtime of our nodes is 10-14 days, which leaves us approximately 0.01 Wh per day at maximum and requires sophisticated energy saving mechanisms.

The system or in particular the software needs to be able to log (social) behavioral information automatically. As the MN is attached to the bat itself, logging encounters between individuals becomes feasible. The required data which

Table 1. Configurations and settings for the field test in Berlin. Additionally, average number of parallel encounter and data transmissions per hour and node.

Description	Formular	Value
Period beaconing	T_{MNB}	2 / 10 s
Day night ratio	R_{DAY}	0.33
Period base station lookup	T_{BSB}	2 s
Average parallel encounter	$N_{MEETING}$	2.05
Data transmission per hour	$N_{PACKETS}$	23.4
Standby	E_{STDBY}	13.173 μ J
Beaconing (TX)	E_{MNB_TX}	271.663 μ J
Beaconing (RX)	E_{MNB_RX}	51.282 μ J
Base station lookup	E_{BSB_RX}	51.282 μ J
Data transmission	$E_{MEETING\_TX}$	47.624 μ J

is necessary for interpretations includes, besides who met whom also the time and duration of the encounter. This data is derived automatically by our software and stored locally on the system memory. Afterwards, data is forwarded to a BS automatically. This keeps disturbances to the observed animals as low as possible, as data can be collected on day time when the bats are resting in their roosts. Doing this way also ensures an eased use for the biologist, as finding and collecting fallen off nodes becomes unnecessary. Therefore, data is available early after deployment and fast reactions to changes or anomalies during a field test is possible.

In previous deployments with small group sizes of up to 15 individuals a runtime of two weeks have been achieved [5]. These field tests showed that available memory was already exhausted by meetings, leading to data losses. Due to an increased number of bats, also more data must be stored and downloaded to a BS. In addition, the next generation of nodes collects more accessory data contributing to a higher amount of data to be downloaded. Therefore, inside roosts a high communication effort must be considered for downloading data, as more encounters between individuals will occur. This is in particular interesting, as decreasing communication effort also decreases likelihood of collisions and might lead to increased successful data reception. This makes an increased data rate even more interesting, to keep the memory usage as low as possible. However, an increased data rate should not lower communication reliability or increase energy demand significantly.

3 Application Energy Demand

Determining the runtime of our application is done by using a model, which is based on the results of previous field tests. Our model uses data like the average number of parallel encounter or average data transmission to a BS to calculate an average current draw. As the battery capacity is given in mAh, the current draw directly translates to the expected runtime. The core parameters of our model are depicted in Table 1 and are explained next. In order to save energy, we acquire data on day time with lower granularity as we expect low frequent changes when the bats are resting. Therefore, our model differentiates between day and night times. In order to initiate a data download, a BS look up is performed to detect the presence of a BS. For the encounter detection, the average number of parallel encounters mainly drives the energy demand for this task. This is because, the receiver must be turned on more frequently, if more encounters are happening. As the calculation of the energy demand is done the same way for all values, we only show the computation of the energy demand for sending a beacon in detail. In our equation the operating voltage of 2.1 V is used and computes the average current draw of sending a beacon over the whole day:

Table 2. Average current consumption per task/action. With these values, the average runtime can be estimated.

Description	Average current draw	Percentage
Standby	1.528 μA	1.804 %
Beaconing (TX)	31.513 μA	37.203 %
Beaconing (RX)	38.687 μA	45.674 %
Base station lookup	12.820 μA	15.136 %
Data transmission	0.155 μA	0.183 %
Sum	84.703 μA	100 %

$$\begin{aligned} I_{Beaconing} &= \frac{E_{MNB\_TX}}{U} \cdot (\frac{R_{DAY}}{T_{MNB}} + \frac{1-R_{DAY}}{T_{MNB}}) \\ &= \frac{271.663\mu J}{2.1V} \cdot (\frac{0.33}{2s} + \frac{0.67}{10s}) = 31.513\mu A \end{aligned}$$

The results of our model are depicted in Table 2 and is explained in the following. With the average current draw of 84.703 μA and a battery capacity of 22 mAh, a runtime of 259.73 h can be expected. The most energy is spent for transmitting and receiving beacons as the results show. Data transmission consumes only 0.185 % of the overall energy but becomes significant once more sensory data is collected about the bats. In order to optimize the energy demand, our applied optimizations of the most energy demanding actions should be explained in the following. The beaconing uses a wake-up receiver to operate the on-board receiver only for short periods and on-demand. Therefore, the on-board receiver, is active only for short periods. As the pattern for the wake-up receiver cannot be changed due to a dictated frame format, no further optimizations are possible. The same applies to the base station look up, which is already nearby physical limits. This is, because the base station beacons are sent in a high rate (each millisecond) and transmitting a beacon requires 900 μs. Sending at such high rates allows the mobile node to activate the on-board receiver only for a short time (1.2 ms), which in turn saves energy.

The next generation nodes, however, are producing data in a high rate. For instance the magnet field sensor and the acceleration sensor are producing 16 Bit values for each dimension. Additionally, an air-pressure sensor is used to determine the altitude of the bat, which is encoded in a 32 Bit value. This, in sum will generate a packet per second, which increases the packet rate per hour to 3623.4. In this scenario an average current draw of 108.515 μA results and shows a significantly increased energy demand for data transmission. The increased data transmission decreases the runtime to 202.736 hours. Our model shows that sending data to a BS consumes 22.076 % of available energy on the next generation nodes. As the bat to bat communication and sending base station beacons are already optimized to the maximum, only optimizing the data transmission remains.

4 Related Work

In the field of wildlife tracking various implementation and systems are available, which are presented in Subsection 4.1. Next, energy efficient protocols, especially in combination with erasure-coded data are discussed (Subsection 4.2).

4.1 Wildlife Tracking Systems

Ossi et al. [18] gives an overview of proximity sensor and Global Positioning System (GPS)-based telemetry systems. As stated out before, weight is besides runtime one of the most important properties for our system.

The Camazotz platform [22] contains several sensors to collect accessory data on-board. It utilizes GPS in order to get localization and movement data of the animal. The more of hardware also comes along with increased weight and energy demand. For instance, a Camazotz node weights 30 g which exceeds our requirement of 2 g. In order to prolong the runtime of the node, as GPS is an high energy demanding device, solar panels and duty cycle adaptions are used. Using solar panels, however, is not an option for our system, as our aim is to track night active animals which are hiding in a roost on day time.

The closest work to our system is EncounterNet [12], which belongs to the class of proximity sensors and also collects encounters between individuals and does not rely on GPS or solar panels. The nodes weights the same (1.3 g) as our nodes and come with dramatically decreased runtime of less than one day and conflicts with our runtime requirement. This is important as longer observation times gives deeper insights into the plasticity of individual behavior and creates more representative data sets.

4.2 Protocols and Optimizations

In order to ensure a reliable communication while keeping energy demand low, several works have been published and are introduced in the following. Mutschlechner et al. [7, 14, 15] worked inside the BATS project on a communication scheme for transmitting data to BSs in an energy efficient way. As a basis, simulations were used, which also contain a movement model of bats to test their approach. They found out, that an acknowledgement-based protocol wastes energy as unnecessary retransmissions were performed. Therefore, they proposed a protocol, which sends data in an opportunistic way without back-channel. In order to keep data reception rates high, they propose the use of EC. Our presented approach builds on-top this approach and is compared to the work of Mutschlechner et al. Therefore, the protocol is discussed in detail in Subsection 5.1.

ECs causing high computational effort as we focus on optimal ECs based on a cauchy matrix. Blomer et al. [3] introduces an approach to transform such cauchy matrix into a Galois field which allows encoding and decoding with XOR's. Therefore, such algorithms becomes more efficient even for less powerful devices like microcontrollers. We use the ZFec algorithm, which shows good overall performance, presented in the work by Plank et al. [19].

Minimizing overheads by using batching in WSNs have been discussed before in the work by Xu Ning et al. [16]. They discuss the problem of adding delays to the data transmission and also how batching can improve energy efficiency. Albeit, no estimates were given on how much energy could be saved or how to use the approach in combination with EC. As our system inherently has latency due to the behavior of the animals, a delayed message does not affect our system at all. Due to the communication scheme, energy awareness and reliability is more important to us and no or less information about these two parameters are given in the related work.

5 Basics on Data Transmission

Keeping disturbances to animals as low as possible is one of our goals. Therefore, BS are placed in promising areas to download data from MNs automatically. Thus, data can be retrieved on day time while the bats are resting.

In order to indicate presence of a BS to a MN each BS sends out a so-called Base Station Beacon (BSB). The BSBs are sent in a high rate (each ms) and allow to activate the radio on the MN for short times. A BSB contains information of the current configuration, cycle-timer for synchronization purposes and an area information. Latter is used to identify where a BS is placed and to alter the communication scheme to the requirements of the area. In sum, two different kind of communication schemes have been identified and are described in the following.

The first one is the roost, in which we have to assume a high communication effort. Therefore, collisions might occur, resulting in data losses. In order to prevent these collisions, we implemented a Time Division Multiplex Access (TDMA) alike communication scheme. Therefore, a synchronization is necessary, which synchronizes the MN to the BS. The BS sends out a BSB which indicates when a cycle of 2 s ends in a granularity of 1 ms. This data is used to synchronize the MN to a BS and to determine when the time slot for sending starts. Afterwards, when the time slot is reached, the data transmission of one packet is initiated.

Additionally, a second communication scheme is required if a bat is flying by a BS. This is the case if a BS is placed in promising foraging sites. Using a TDMA would incur a delay to transmit data which leads to a higher packet loss as a bat might have left the communication range already. Therefore, using a TDMA alike communication scheme becomes infeasible. In such areas, we can expect only small group sizes which allows us to send data directly after a BSB is received. Therefore, we achieve the smallest possible delay and an increased likelihood of a bat inside the communication range.

Both schemes are implemented in our application and a switch to the appropriate scheme is performed automatically. However, both schemes have in common that they are time critical. Therefore, low delays for sending data are desired to minimize possible conflicts with both communication schemes.

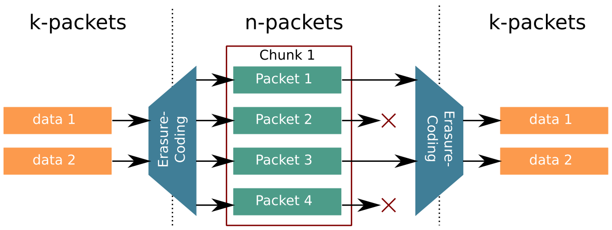


Figure 1. Schematic principle of the erasure codes with data losses, which can be sustained due to the error correction capability of an erasure-code

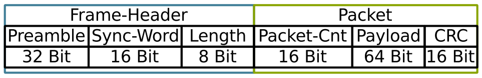
5.1 Erasure Coding

Our approach builds on top of the communication scheme proposed by Mutschlechner et al. [7, 14, 15] as the scenarios and assumptions also apply to our system. The work from Mutschlechner et al. shows that using acknowledgments would cause a high rate message retransmissions [15]. They found out, that most of time data is received successfully at the BS, but acknowledgements in return are not received by the MN. Therefore, energy is wasted due to unnecessary retransmission of data. To circumvent this problem, data is sent without any acknowledgments as kind of a data shower and losses of data cannot be detected on the MN. Therefore, an EC is used to increase the reliability of the data transmission.

The EC adds redundancy to the transmitted data to increase the reliability. In our scenario, we use $k = 2$ original data which are encoded to $n = 4$ packets (later on called chunks), as depicted in Figure 1. Due to data losses, some packets may not be received by a BS. However, if at least k packets from a chunk are received, all data can be restored. Therefore, EC gains less data losses compared to sending data twice, as it does no matter which packet has been received. Furthermore, the first k packets contain the original data without any encoding. Thus, using those packets can be done if only one packet is received and becomes important in our protocol, as we make use of this property later on.

5.2 Frame Layout

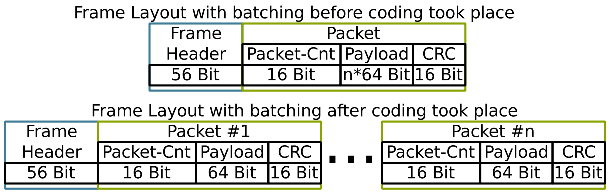
In the concept, an erasure-channel is assumed which means a packet is received without any errors or no packet at all. As a consequence, each packet is sent in its own frame, which is depicted in Figure 2. In our implementation of the communication scheme, each frame consists of in sum six different data fields. The first are the preamble and the sync word, which are used to synchronize the receiver to the transmitter. Additionally, a packet length field is used to support different frame / packet lengths. These three data fields are later on called frame header and are evaluated by the radio hardware itself. Despite the frame header, a packet follows and consists of three data fields. In order to reassemble packets in the right order, as data might be transmitted among different BSs, a packet counter is used. This data field is followed by the payload itself, which contains the EC encoded data to be sent. Besides the payload, a CRC is computed over the packet counter and the payload to ensure an erasure channel.



Frame-Header			Packet		
Preamble	Sync-Word	Length	Packet-Cnt	Payload	CRC
32 Bit	16 Bit	8 Bit	16 Bit	64 Bit	16 Bit

Figure 2. Packet scheme for data transmission according to Mutschlechner et al. Due to the Cyclic Redundancy Check (CRC) an erasure channel is ensured if a bit error occur.

6 Theoretical Evaluation on Packet Losses



Frame Layout with batching before coding took place

Frame Header	Packet		
	Packet-Cnt	Payload	CRC
56 Bit	16 Bit	n*64 Bit	16 Bit

Frame Layout with batching after coding took place

Frame Header	Packet #1			. . .	Packet #n		
	Packet-Cnt	Payload	CRC		Packet-Cnt	Payload	CRC
56 Bit	16 Bit	64 Bit	16 Bit		16 Bit	64 Bit	16 Bit

Figure 3. Packet layout, on naive batching Vs. improved batching.

As stated out before, decreasing energy demand while also keeping data losses low is the main goal of our protocol. In order to rate our solution, we assume a constant, independent Bit Error Rate (BER) p_{error_bit}. With this BER, we can compute the conditional probability to all bits transferred successfully. This results in the following equation:

$$P_{good}(n) = (1 - p_{error\_bit})^n \tag{1}$$

where n refers to the number of bits to be transmitted. Furthermore, we define the variable r for the burst rate, which indicates how many payloads or packets are batched inside a frame. The burst rate $r = 1$ generates in both approaches (naive and improved) the same packet layout as proposed by Mutschlechner et. al. and is, therefore, used as ground truth. At the beginning, we can consider to batch data before and after the EC took place. Both variants will alter the error probability and also energy efficiency as a different amount of data must be transferred and is discussed in the following.

6.1 Naive Approach

The naive way batches all data before the EC took place. Therefore, the payload is increased and a CRC is computed over the whole payload and packet counter. Doing this way, keeps overheads caused by CRC and packet counter at a minimum, which is depicted in Figure 3. To compute the packet error rate, we need to calculate the conditional probability for sending the frame header successfully and also for the payload including packet counter and CRC. In our implementation, we use a 16 bit CRC and a 16 bit packet counter which must be added to the payload of 64 bit.

$$P_{good\_packet\_naive}(p) = P_{good}(56) \cdot P_{good}(32 + r \cdot 64) \tag{2}$$

6.2 Bursting: Improved Approach

Our bursting approach, performs batching after the EC took place. This opens the opportunity, to compute multiple CRCs over each payload and packet counter. Therefore, bit errors of one packet does not affect other packet inside the frame. As the length for each computed CRC is independent

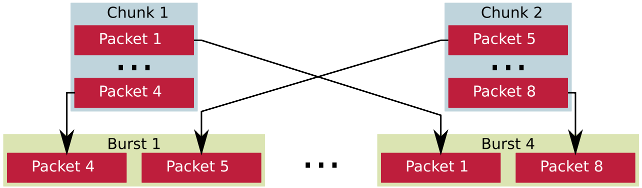


Figure 4. Packet layout, for batching after the EC and requires for each packet its own packet counter and CRC

of the bust length, the error probability for one undamaged payload stays the same, as the following equation shows:

$$P_{good\_packet\_improved}(p) = P_{good}(56) \cdot P_{good}(96) \qquad (3)$$

This would result into a higher reliability which comes at a price of more data to be sent, which decreases energy efficiency.

Since the EC produces four packets, sending those packets in one frame would render the EC useless, refer to Figure 1. This is because the frame header is a single point of failure to lose all packets, resulting in data loss. Distributing packets over multiple frames would gain higher probability to receive a packet of a chunk. Therefore, a scrambling must be done to ensure that each packet is independent from each other in one frame, which is later on called *burst*. In our approach, a packet from multiple chunks are used to be transmitted in one burst. Depending on the available chunks c, we build bursts with a rate of $r = c$ packets. As our approach is applicable on other erasure code rate we assume a configuration described in Subsection 5.1 later on. The scheme for $r = 2$, is depicted in Figure 4 and takes into account that the first two packets contain original data which does not require any decoding phase. Therefore, packet 1, 2, 5 and 6 are distributed among all available bursts and placed at the beginning of a frame header as synchronization errors are less likely directly after the frame header. The remaining packets are also distributed in a way, that each burst contains packets from different chunks. However, due to the fixed number of bursts, not all original data packets can be sent in an individual burst. Therefore, at higher burst rates ($r >= 3$) a burst can contain more than one original packet. Besides this scheme, we also adapt the burst rates accordingly to the amount of available data when a BS comes into receiving range. This causes minimal latency, as the data is (naturally) accumulated while the animals are foraging or absent from the roosts. Additionally, computing chunks in advance is possible. Therefore, delays are kept minimal, which lowers influences to our TDMA alike communication scheme and effectively increases data rate up to a factor of four.

6.3 Comparison of Packet Losses

We solved both equations for different BERs and is depicted in Figure 5. For a burst rate $r = 1$ the naive and the bursting approach shows the same packet error rate, as the same frame layout is send to a BS. However, at higher burst rates $r > 1$ the packet losses are almost similar for low and high BER values. In the range of 0.2% up to 2% BER, the naive approach shows significantly higher packet losses. At 0.45% BER the naive approach at a burst length of $r = 4$ receives only 21.2% of all packets, whereas the improved approach receives 50.38% of all packets. This is caused by the higher amount of data per CRC which increases likelihood of a damaged frame. Therefore almost the double amount of packets can be received with our approach and stays comparable to the state of the art $r = 1$. However, due to the EC, the packet losses might be compensated and should be discussed in the next section to rate both approaches on data losses.

7 Theoretical Evaluation on Data Losses

Due to the use of batching, we expect higher data losses as the frame header poses a single point of failure for multiple packets. We analyzed the packet losses so far, however, due to the EC a lower data loss might occur, which makes modeling of the error correcting capabilities necessary. In our application we used $k = 2, n = 4$ to keep the overhead in terms of memory and communication effort low. Therefore, out of four packets (chunk) at least two have to be received in order decode data successfully, i.e., no data loss occurs. However, our approach is not limited to those values in general and can be used with other code rates which may require an adapted distribution of data among bursts.

We chose Hidden Markov Models (HMMs) as they have been proven to be very accurate to model packet losses in networks [24]. With these models, we model the packet losses depending on a BER which is discussed in Subsection 7.1. In Subsection 7.2 we compare both, the naive and our approach in terms of data losses.

7.1 Packet Error Model and Simulation

In the following, we will showcase the error model by using Markov models and derive a formula to estimate the error rate for each packet. We model a data transmission to be received successfully with conditional probabilities without any erroneous bits as defined in Equation 1. Furthermore, we define the probability for a non successful transmission to:

$$P_{fail}(n) = 1 - P_{good}(n) \qquad (4)$$

As the number of bits for the frame header and payload differ, we calculate the probability of a wrong header to $P_h = P_{fail}(l_{header})$ and of a wrong payload to $P_p = P_{fail}(l_{payload})$. By using this values, we can model the probability of an erroneous and non-erroneous packet with a HMM. Figure 6 shows the HMM to receive a packet consisting of a header and a payload. The circles are indicating the state, whereas boxes indicates the emission. $\textcircled{T_H}$ is the state to send a header which can fail with the probability P_h which in turn is indicated by the emission $\boxed{+0}$ as no packet is received. Furthermore, with the probability $1 - P_h$ we can reach the state for sending payload $\textcircled{T_P}$. Also in this state, a corrupted bit with the probability P_p would cause a loss of the whole packet and the emission $\boxed{+0}$ is triggered. However, if the payload contains no bit failures the emission $\boxed{+P1}$ is triggered. From this model, we can derive the equations for a failed packet P_{fail} and a successfully received packet P_{good}.

$$P_{fail\_packet} = P_h + (1 - P_h) \cdot P_p \qquad (5)$$
$$P_{good\_packet} = (1 - P_h) \cdot (1 - P_p) \qquad (6)$$

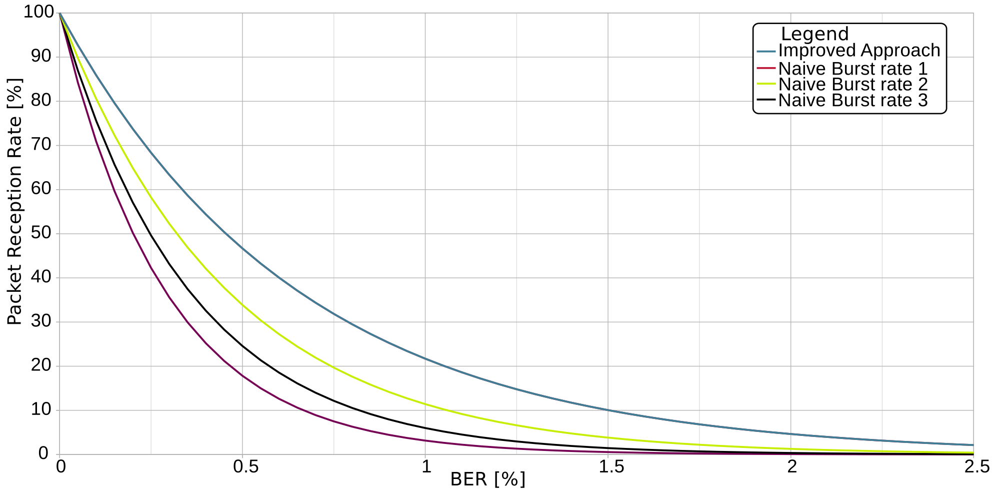


Figure 5. Packet reception rate depending on the bit error rate. As the improved and naive approach for burst rate $r = 1$ results into the same formula, both graphs are the same and cannot be distinguished. However, for higher burst rates the packet reception rate drops significantly compared to the improved burst layout.

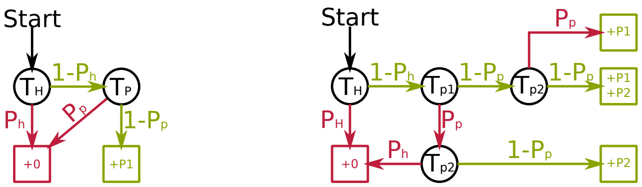


Figure 6. HMM to model a packet-loss with a conditional probability for sending payloads

With our given lengths, of 56 and 96 Bits respectively and a BER P_{error_bit} we can compute the values for $P_p = P_{fail}(96)$ and $P_h = P_{fail}(56)$.

For our naive approach, the length of the payload must be simply increased. Thus the equations for $P_p = P_{fail}(32 + 64 \cdot r)$ for $r = 1...4$ must be used. However, in case of the bursting (improved) approach, the models are getting more complicated and should be discussed exemplarily for $r = 2$. The corresponding HMM is depicted in Figure 6 on the right side. As multiple packets can be received or lost in sum three states are possible. $\boxed{+0}$ occurs if no payloads have been received successfully or if the frame header was corrupted. Furthermore, if the frame header has been received successfully, $\boxed{+P1}$ is raised, if no bit failure occurs during the transmission of the first payload. Independently to the first payload, $\boxed{+P2}$ is reached if no failure inside the second payload occurs. The models for higher burst lengths, are modeled the same way but are not depicted as they require up to 17 states.

Our simulation uses those models, which are walked through similar to a Monte-Carlo simulation. Therefore, in each state of the HMM we generate a random number which is mapped to the the probability to determine which state should be entered next. If an emission is available, the emission is triggered, in our case for each chunk in a burst a counter is increased by one. With this counter values, we can determine how many packets per chunk have been received, which is necessary to determine whether a data loss occurred or not. If at least two packets have per chunk has been received, no packet loss occurs and a global data counter is increased by 2 (as two data points could be restored). As in sum four bursts are send, we re-execute the model four times until the chunk counters are evicted. In order to get statistically significant data, this procedure is repeated 100,000 times to keep variations low.

7.2 Comparison on Data Losses

In order to ensure that our modeling and assumptions are correct, we compared our simulation against the results from Mutschlechner et al. [15]. In the paper we found a packet loss rate of 57 % which can be translated to a BER of 0.38 % and a data reception rate of 79 %. However, our models show a slightly lower packet loss rate of 56.26 % (error: 1.31 %) and a slightly lower data reception rate of 77 % (error: 2.53 %). The differences between both models are most likely due to the frame header, as no information can be obtained in the work by Mutschlechner et al. Additionally, simulations inherently have noises in form of numerical noise which also explain the differences. However, as the error is less than 3 % we ignore the error as our simulation produces comparable results.

In order to rate the naive and improved approach, we swept through multiple BERs. For the improved approach, the data losses are almost the same for every burst rate, as the diagram in Figure 5 already supposes. The reason is, that one bit failure does not affect the other packet inside a burst. Compared to the state of the art (burst rate $r = 1$) an increased data loss of up to 2% for the burst rate $r = 4$ at 0.67 % BER can be observed. The naive approach, however, shows a very different behavior. With increasing burst rate, the data losses are also increasing. This results into much higher data losses at a burst rate $r = 4$ with data receptions of only 20 %. Compared to the state of the art, which receives 68 % at a BER of

0.45 % a 3.4 times higher data loss occurs.

Comparing the improved and the naive approach with the highest burst rate $r = 4$ the improved one outperforms the naive approach by a factor of 3.3 in maximum. For very low and very high BERs values, both approaches show almost the same data losses. Therefore, it is necessary to determine realistic BER for our application, which is discussed in Section 9. As more data must be transmitted for the bursting approach, also the energy demand must be evaluated which is presented in Section 8.

8 Energy Savings

In order to determine the energy savings capability of both solutions, we firstly calculate the energy demand for sending packets including overheads for starting and initializing the radio. We used values given from the data sheet [21]. In order to determine the energy demand for sending a packet, we have to express the energy demand for starting the microcontroller E_{start}, copying data to the radio E_{copy} and sending the data itself. The start itself requires a startup time of $T_{EM2\rightarrow active} = 10.7\,\mu s$. As no current consumption is given for starting up the microcontroller ($I_{EM2\rightarrow active}$), we used the average from the current consumption in EM2 $I_{EM2} = 3.3\mu A$ and $I_{active} = 6.288\,mA$. However, measuring the energy demand is almost impossible due to too small currents and no possibility to toggle a pin during this phase. As our design runs at a voltage of 2.1 V, the energy demand for starting the microcontroller can be computed to:

$$E_{start} = I_{EM2\rightarrow active} \cdot T_{EM2\rightarrow active} \cdot V = 70.68nJ \tag{7}$$

Copying data and starting the radio is assumed to last $T_{copy}(r) = T_{radio\_start} + T_{radio\_copy}(r)$. We measured the times for starting the radio $T_{radio\_start} = 0.8\,ms$ and also for copying data depending on the burst rate r to the radio $T_{radio\_copy}(r) = 0.4\,ms \cdot r$ via toggling a bit on the microcontroller. With these values, the energy demand can be expressed to:

$$E_{copy}(r) = I_{active} \cdot T_{copy}(r) \cdot V \tag{8}$$

Sending r packets, the time for sending is computed with a given length of $N_{payload}$ for the payload length and N_{FH} for the length of the frame header. Furthermore, our radio transmits data with a fixed data rate of f_s. In the case of the naive approach, the time for sending a packet can be expressed to:

$$\begin{aligned} E_{send\_naive}(r) &= \frac{N_{payload} \cdot r + 32 + N_{FH}}{f_s} \cdot V \cdot I_{send} \\ &= \frac{64\,Bit \cdot r + 32 + 56\,Bit}{300,000\,Bit/s} \cdot V \cdot I_{send} \end{aligned} \tag{9}$$

Please note, that the naive approach only copies the payload plus additional 32 Bits for CRC and packet counter. The equation for the improved approach, is slightly different as only the size of the packet $N_{packet} = N_{payload} + 32$ changes and can be expressed to:

$$\begin{aligned} E_{send\_improved}(r) &= \frac{N_{packet} \cdot r + N_{FH}}{f_s} \cdot V \cdot I_{send} \\ &= \frac{96\,Bit \cdot r + 56\,Bit}{300,000\,Bit/s} \cdot V \cdot I_{send} \end{aligned} \tag{10}$$

Table 3. Energy demand for sending data with the naive and improved approach. From this table, it can be observed that the naive approach can save up to 20 % more energy.

r	Energy $E_{naive}(r)$	Energy $E_{improved}(N)$	Difference
1	47.62 μJ	47.62 μJ	0 %
2	66.25 μJ	72.93 μJ	10.075 %
3	84.89 μJ	98.24 μJ	15.73 %
4	103.52 μJ	123.55 μJ	19.34 %

Additionally, the current draw for sending a packet cannot be derived from the data sheet. Therefore, we measured the current draw to $I_{send} = 29.5\,mA$ with our parameters and the overall energy demand for both approaches can be computed to:

$$E_{naive}(r) = E_{copy}(r) + E_{start} + E_{send\_naive}(r) \tag{11}$$

$$E_{improved}(r) = E_{copy}(r) + E_{start} + E_{send\_improved}(r) \tag{12}$$

Computing the energy demand for the state of the art is easy, as both approaches are falling back to the state of the art at $r = 1$. Therefore, for sending one packet the following equation can be used to compute the energy demand:

$$E_{reference} = E_{naive}(1) = E_{improved}(1) = E_{state\_of\_the\_art} \tag{13}$$

8.1 Comparison of Energy Demands

For both approaches, the energy demand for different burst length are computed and depicted in Table 3. As stated out before, the naive approach consumes less energy compared to the improved one. The surprising result, however, at a burst rate of $r = 4$ the bursting approach sends 384 bits whereas the naive approach only 288 bits as payload. Even with an overhead of 96 Bits or 33 % the higher energy demand is at most 19.34 %. The smaller increase of energy can be explained due to the overhead for sending frame header, copying data and starting the microcontroller. In order to rate the energy saving capabilities compared the state of the art, we compute the energy per packet and is depicted in Table 4. For higher burst rates, more energy can be saved as overheads like frame header or starting the microcontroller are constant and distributed among multiple payloads/packets. With the naive approach up to 45.65 % of energy can be saved compared to the state of the art. In contrary the bursting approach saves up to 35.14 % per packet. This shows, that both approaches can save a significant amount of energy compared to the state of the art.

8.2 Theory Versus Measurements

In order to keep uncertainties as low as possible, the overall energy demand have been measured with a source meter from Agilent (DC Power Analyzer N6705A). The results are depicted in Table 5 and shows the measurement of the improved approach. As explained before, measuring correct energy demand is time consuming, as multiple measurements must be made to keep noise caused by the internal analog-digital converters as low as possible.

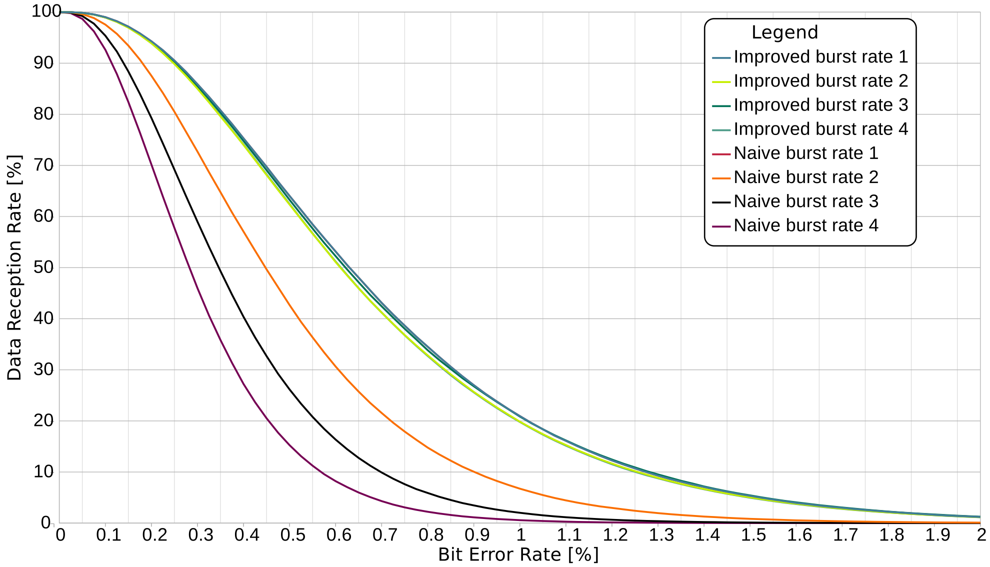


Figure 7. Data reception rates depending on bit error rates.

Table 4. Energy demand for sending one packet/payload and the relative energy demand compared to the base line (r=1).

r	Energy $E_{naive}(N)/N$	Percentage	Energy $E_{improved}(N)/N$	Percentage
1	47.62 μJ	100 %	47.62 μJ	100 %
2	33.125 μJ	69.57 %	36.5 μJ	76.65 %
3	28.3 μJ	59.43 %	32.75 μJ	68.77 %
4	25.88 μJ	54.35 %	30.89 μJ	64.86 %

Table 5. Overview of the measured energy demand for sending packets and the relative energy demand compared to the theoretical estimate and to the base line (r=1).

r	Energy $E_{send}(N)$	Energy $E_{send}(N)/N$	Percentage	Difference to Estimate
1	49.799 μJ	49.799 μJ	100 %	4.57 %
2	75.172 μJ	37.586 μJ	75.476 %	3.07 %
3	100.231 μJ	33.410 μJ	67.091 %	2.029 %
4	125.590 μJ	31.398 μJ	63.049 %	1.66 %

Furthermore, the sourcemeter also might add an absolute errors due to component variations.

Table 5 shows that our estimates are slightly too small compared to the measurements. The error ranges up to 4.57 % and is most likely caused by measurement errors like determining start/end of sending, component variations and round-off errors. Therefore, for burst rates up to $r = 4$ our equations approximates the energy demand correctly and the previous discussion can be used for further analysis.

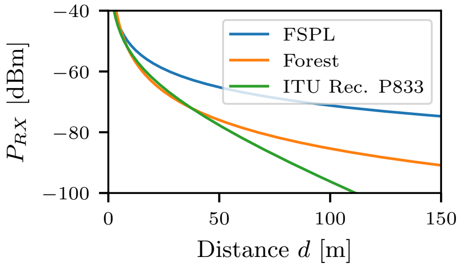


Figure 8. Path loss model for a mid dense forest compared to the free space path loss and a ITU channel model.

9 Estimation of Realistic Bit Error Rates

The naive and improved approach generate similar data losses at small BERs. Therefore, understanding errors in the wireless data transmission from the MN to the BS, basic effects of the wireless radio propagation channel are mandatory and introduced in this section. The most basic property of the wireless channel is the bulk path loss, cf, Figure 8. The path loss attenuates the signal power received at the BS. This limits the Signal Noise Ratio (SNR) at the receiver. Thus, path loss has an impact on the recovery of the emitted data bits. Pass-loss L between transmitter and receiver is depending on the distance d and is given by:

$$L = E \cdot 10 \cdot \lg(d) + 20 \cdot \lg\left(\frac{4\pi f_c}{c}\right) + C,$$

where E denotes the path loss exponent, d the distance in meters, f_c denotes the carrier frequency in GHz, c denotes the speed of light and C denotes a constant offset. The parameter E and C depend on the environment. In Nowak at al. [17] these parameters are derived for a mid dense forest in Germany with $E = 3.2$ and $C = 11$. Furthermore, own investigations and measurements have shown similar results. In a multipath-free environment an Additive White Gaussian Noise (awgn), e.g. thermal noise at the receiving node, is added to the captured signal. The BER for the considered signal is expressed by [20]:

$$P_e = \frac{1}{2}\mathrm{erfc}\sqrt{\frac{E_b}{N_0}}$$

where E_b is the energy per bit, N_0 is the thermal noise power spectral density and erfc is the complementary error function.

Unfortunately, this theoretical BER will represent a far to low BER value in realistic scenarios. Besides thermal noise arising in the receiver, other more dominant effects have to be considered. A significant increase of the BER at the BS is caused by in-band interference. The utilized frequency bands, 868/915 MHz, are free to be used by anybody. Hence, systems are occupying the band and causing in-band interference. Furthermore the TX and RX at the base station are not synchronized and the BSB are sent with a high rate to keep the receiving energy consumption at the MN at a decent level. This full duplex transmission with a frequency separation of 5.5 MHz leads to further in-band interference at the RX path caused by the out-of-band radiation of the TX path. To keep the out-of-band emission at a minimum, GMSK modulation is used with BT=1.0. However, still the TX is transmitting in the vicinity of the receive antenna with a power $P_{TX} = 13\,\mathrm{dBm}$. TX and RX antenna decoupling has been measured to be 22 dB. In conclusion, there is strong interference from the BS TX path. This results in an BER that is by orders of magnitude larger than for a AWGN channel. The adjacent channel power, i.e. out-of-band power of the interfering transmitter, have been measured in the laboratory. Average noise density in the adjacent band is $-126.4\,\mathrm{dBm/Hz}$. With decoupling of TX and RX this results in a $N_0 = -148.4\,\mathrm{dBm/Hz}$, cf. $N_0 = -174.0\,\mathrm{dBm/Hz}$ for thermal noise.

Another significant increase of the BER at the BS is caused by the arbitrary orientation of the MN linear polarized antenna, due to the unknown flight orientation of the bat. TX and RX antennas at the BS are cross-polarized for better decoupling. However, both of them are linear polarized. Hence, the wireless channel is not reciprocal. Therefore, fading of the signal due to shadowing and multipath will vary on the channel from the MN to the BS and in the opposite direction. This effect will additionally lead to fading of the channel even when the transmission from the MN to the BS are in the coherence time of the channel after the BSB is received. Simulations have shown that this increases the BER significantly. However, due to the unknown flight behavior and orientation there is hardly no quantitative evaluation feasible. For the propagation channel simulations we assumed that that all antennas are linear polarized. TX and RX antenna are rotated by 90° towards each other. For the polarization losses we assume that possible rotation angles are uniformly distributed. Forward and backward link exhibit independent log-normal fading with $\sigma = 7\,\mathrm{dB}$ [17]. The bat node is triggered at a sensitivity of $= P_{RX,sens} - 75\,\mathrm{dBm}$. We compute the distance-equivalent path loss for the position at which the bat is triggered to send data. The path loss for the forward link is given by

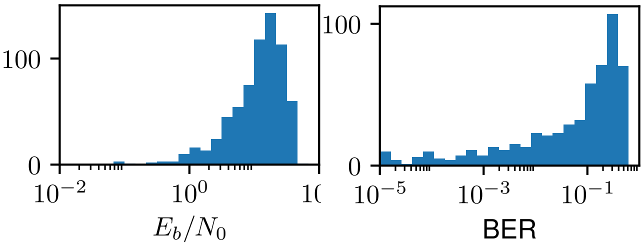


Figure 9. Statistical distribution of E_b/N_0 and bit-error rate (BER) for the simulated radio propagation channel.

$$L^{fw} = P_{TX} - P_{RX,sens} - L_P^{fw} - F^{fw}$$

where L_P^{fw} is the forward link fading and L_P^{fw} denotes the forward link polarization loss. As the mobile node is immediately transmitting its data, i.e. there is no change in the distance for the backward path, the receiver E_b/N_0 is expressed by

$$\begin{aligned} E_b/N_0 = {} & P_{TX} - L^{fw} - L_P^{bw} - F^{bw} \\ & - N_0 - NF + T_{s,\mathrm{dB}}, \end{aligned}$$

where L_P^{bw} denotes the backward link polarization loss. F^{bw} describes the uncorrelated fading of the backward channel. NF is the receiver noise figure and $T_{s,\mathrm{dB}}$ is the symbol duration in dB. In Figure 9 probability distributions for E_b/N_0 and BER are depicted for the simulated propagation channel.

For the our scenario this results in a mean packet received rate of 56.14 % which corresponds to a mean BER of 0.38 %, which is in line with the results from Mutschlechner et al. The simulation parameters and models have been verified by a previous field test in Essehof. In this scenario a received packet rate and BER of 93.03 % and 0.05 %, respectively, have been computed. These values perfectly match the results of the Essehof field test.

10 Summary

For the field tests, it is necessary to pick one promising approach, as a field test consumes a lot of time for the preparations and the test itself. In our evaluation, we ran two field tests in sum, the first took place in Essehof nearby Brunswik and in Königsheide in Berlin. The first is used to check whether our estimates are correct as the second one is a real deployment of our system on the animals. Thus, the results should be summarized and a trade off of both approaches should be made in this section.

As stated out before, the naive and improved approach behave the same, if the BER is low. However, as our values from channel model indicates, the BER ranges from 0.05 % up to 0.38 %. In this range, both approaches are performing very differently in terms of data losses. The naive approach, shows at minimum a data reception rate of 31 % whereas the improved receives 76 % of all packets. In terms of energy, the improved approach can save up to 35.14 % and the naive approach up to 45.65 % of energy per data point. Therefore, the improved approach is chosen, as we spend 10 % more energy to receive 245 % more data. Thus, we expect better overall performance and more data compared to the naive approach, which is not discussed further.

11 Practical Evaluation on Data Losses

Fitting our system to a bat results in uncertainties as bats are unpredictable. For instance, the movement of a bat will alter the communication channel as the antenna follows the movement of the bat. Furthermore, inside the roost, the antennas might get bent while a bat is resting which also affects the communication parameters. Therefore, a static field test is used to evaluate data losses. In order to get realistic values, we chose a setup inside a forest nearby Brunswick (Essehof) instead of testing our system in an office building. Each deployed node had a height of 1.3 m from the ground and altered the burst lengths in a high rate to avoid biased data due to weather conditions. The MNs were placed in different distances to a BS and we checked the values against our theoretical model at a distance at 45 m which showed a packet loss of 7.5 % which is in line with our model (6.97 %). However, in order to get statistically significant data, the data and packet losses over all mobile nodes is used. Therefore, a lower packet loss rate is the result compared to a distance of 45 m. Additionally, we changed the antenna orientation during the field test in order to prevent any biased data due to antenna orientation. In sum, we transmitted 5,043,199 packets to a BS within 2 weeks and evaluated the packet and data losses for each burst length individually.

In the first week of the field test we chose an antenna orientation parallel to the receiving antenna of the BS. On the following week we rotated the antennas by 90° to each other to get the worst antenna orientation possible. The values are shown in Table 6 are discussed individually in the following.

A parallel antenna orientation is the most reliable receiving orientation, thus, low packet losses are expected. In our field test, the state-of-the-art, presented by Mutschlechner et al., showed the highest packet losses with 1.44 % whereas the highest burst length showed the smallest packet loss with 1.29 %. A systematic decreased packet loss with increasing burst length is not observable as burst length 2 shows less packet losses compared to burst length 3. In this case, we assume that the different packet losses are caused by noise as only a small subset of packets were lost. Therefore, we conclude that each burst length shows comparable packet losses independently from its length and is in line with our first theoretical estimation. The data losses also shows this behavior and lies in a range of 0.022‰up to 0.115‰and is negligible. In our second half of the field test we changed the antenna orientation to be perpendicular to each other. Therefore, we expected higher packet losses due to a non optimal antenna orientation and checked whether higher burst rates have any negative impact compared to the state-of-the-art. The first expectation is fulfilled as the packet losses are increased, ranging from 3.02 % up to 3.11 %. Also in this experiment, the state of art showed higher packet losses compared to higher burst lengths. However, with respect to data losses all higher burst lengths show a slightly increased data loss ranging from 0.443 ‰ to 0.650 ‰. This leads to an increased packet loss of 0.226 ‰ and is also negligible.

Table 6. Packet and data losses for different burst lengths.

Burst	Packets		Data	
Rate	Sent	Lost	Sent	Lost
Parallel RX and TX antenna orientation				
1	225,840	3,246 (1.44 %)	112,923	13 (0.115‰)
2	451,708	5,879 (1.30 %)	225,859	24 (0.106‰)
3	677,595	9,434 (1.39 %)	338,803	37 (0.109‰)
4	903,345	11,696(1.29 %)	451,672	10 (0.022‰)
Perpendicular RX and TX antenna orientation				
1	278,462	8,664 (3.11 %)	139,231	59 (0.424‰)
2	556,958	16,797 (3.02 %)	278,495	181 (0.650‰)
3	835,435	25,641 (3.07 %)	417,722	185 (0.443‰)
4	1,113,856	34,075 (3.06 %)	556,935	265 (0.476‰)

Table 7. Send packets per burst rate. With our used improved approach 1.181 J (19.995 %) can be saved in a real deployment for sending data to a BS

Burst Rate	Packets N	Energy $r = 1$ [J]	Energy Improved[J]
1	44,876	2.235	2.235
2	6,799	0.338	0.255
3	4,743	0.236	0.158
4	62,192	3.097	2.077
Overall	118,610	5.906	4.725

12 Decreased Energy Demand Effectiveness

After analyzing our approach, theoretically and in a static field test, we also tested our approach in the wild. Therefore, we set up a field test in Königsheide, Berlin in Germany. We outfitted 34 individuals with our system of the same species (*Nyctalus noctula*). *Nyctalus noctula* are known as a species, which often changes the roosts. Therefore, multiple BSs are placed among different roosts, albeit not every roost has its own BS due to a limited number of BSs. Furthermore, a biologist evaluated promising locations with a mobile BS to place a BS in areas of high population to collect as much data as possible and was repeated daily.

We used the observation times of the BSs to determine the runtime of nodes as well as the received packets for further analysis. After deployment, eight nodes were absent from any BS within 48 hours. This can be caused by bats leaving the observation area or failed nodes. As we cannot track this events down in more detail, these nodes are not considered for further analysis. The remaining nodes sent 118,610 packets in total and 4,942 packets in average per node. In order to determine how many energy is spend for sending data, we use the values determined in Section 8.

$$E_{sending} = E_{packet} \cdot N_{packets}$$

With this equation, we can compute the energy for each burst rate in order to determine the energy demand which is depicted in Table 7.

Most packets are sent with a burst rate of $r = 1,4$. The reason is, that, when the memory is empty (which is the case in the roost) only low burst rates are possible. Furthermore, if a bat was absent from the roost, data is stored until a BS comes into receiving range. Due to the accumulated number of data inside the node, data is transmitted with the highest burst rate. In order to compare the energy savings, we firstly calculated the energy demand for all packets with the burst rate 1. With this rate, 5.906 J is used. On the contrary with bursting only 4.725 J is used. Thus 19.996 % of energy can be saved in a real deployment.

13 Conclusion

In this paper, we presented our approach bursting. With this technique, we achieved a decreased energy demand of 30.18 % for sending data to a base station. Our approach has been extensively analyzed and tested in a theoretical and practical manner. The results are in line and comparable to each other and show a theoretical energy saving capability of 35.14 % compared to the state-of-the-art.

Due to concatenation of packets to a burst a higher packet loss is possible. As data reception rates are important for our application, we also analyzed packet and data losses extensively. In our theoretical evaluation we identified the critical single point of failure, which may lead to an increased packet loss of up to 2 % at maximum. We modeled packet losses via hidden Markov models and also computed the theoretical bit error rates. In a two-week lasting field test, which is also matched by the outcomes of our theoretical models, we observed a negligible data loss rate as the bit error rates were lower. Therefore, we conclude, that our approach can significantly decrease the energy demand for transmitting data to base stations with negligible drawbacks.

14 Acknowledgments

This work was partly supported by the German Research Foundation (DFG) under grant no. FOR 1508 (research unit BATS), grant no. KA 3171/3-2 and grant no. KO 4340/2-2.

We want to thank Sebastian Hofmann for his work on the measurements during the field test in Essehof.

15 References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422, 2002.

[2] S. Amelon, D. Dalton, J. Millspaugh, and S. Wolf. Radiotelemetry: Techniques and analysis. In *Ecological and behavioral methods for the study of bats*. 2009.

[3] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman. An xor-based erasure-resilient coding scheme, 1995.

[4] Q. Cao, D. Wang, T. Abdelzaher, B. Priyantha, J. Liu, and F. Zhao. Energy-optimal batching periods for asynchronous multistage data processing on sensor nodes: Foundations and an mplatform case study. In *2010 16th IEEE Real-Time and Embedded Technology and Applications Symposium(RTAS)*, volume 00, pages 101–110, April 2010.

[5] B. Cassens, S. Ripperger, M. Hierold, F. Mayer, and R. Kapitza. Automated encounter detection for animal-borne sensor nodes. International Conference on Embedded Wireless Systems and Networks (EWSN), 2017.

[6] F. R. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 107–122, New York, NY, USA, 2010. ACM.

[7] F. Dressler, M. Mutschlechner, B. Li, R. Kapitza, S. Ripperger, C. Eibel, B. Herzog, T. Hönig, and W. Schröder-Preikschat. Monitoring Bats in the Wild: On Using Erasure Codes for Energy-Efficient Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 2016.

[8] F. Dressler, S. Ripperger, M. Hierold, T. Nowak, C. Eibel, B. Cassens, F. Mayer, K. Meyer-Wegener, and A. Koelpin. From Radio Telemetry to Ultra-Low Power Sensor Networks - Tracking Bats in the Wild. *IEEE Communications Magazine*, 2015.

[9] J. Felix Drexler, V. Corman, M. Müller, G. Maganga, P. Vallo, T. Binger, F. Gloza-Rausch, A. Rasche, S. Yordanov, A. Seebens, S. Oppong, Y. Adu-Sarkodie, C. Pongombo, A. N Lukashev, J. Schmidt-Chanasit, A. Stoecker, A. José Borges Carneiro, S. Erbar, A. Maisner, and C. Drosten. Bats host major mammalian paramyxoviruses. 3:796, 04 2012.

[10] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi. Energy consumption anatomy of 802.11 devices and its implication on modeling and design. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pages 169–180, New York, NY, USA, 2012. ACM.

[11] A. Groseth, H. Feldmann, and J. E. Strong. The ecology of ebola virus. *Trends in Microbiology*, 15(9):408 – 416, 2007.

[12] I. I. Levin, D. M. Zonana, J. M. Burt, and R. J. Safran. Performance of encounternet tags: field tests of miniaturized proximity loggers for use on small birds. *PloS one*, 2015.

[13] G. F. McCracken, K. Safi, T. H. Kunz, D. K. N. Dechmann, S. M. Swartz, and M. Wikelski. Airplane tracking documents the fastest flight speeds recorded for bats. *Royal Society Open Science*, 3(11), 2016.

[14] M. Mutschlechner, F. Klingler, F. Erlacher, F. Hagenauer, M. Kiessling, and F. Dressler. Reliable communication using erasure codes for monitoring bats in the wild. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014.

[15] M. Mutschlechner, B. Li, R. Kapitza, and F. Dressler. Using erasure codes to overcome reliability issues in energy-constrained sensor networks. In *11th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2014.

[16] X. Ning and C. G. Cassandras. Message batching in wireless sensor networks—a perturbation analysis approach. *Discrete Event Dynamic Systems*, 2010.

[17] T. Nowak, M. Hartmann, T. Zech, and J. Thielecke. A path loss and fading model for rssi-based localization in forested areas. In *IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (IEEE-APWC 2016)*, 2016.

[18] F. Ossi, S. Focardi, G. P. Picco, A. Murphy, D. Molteni, B. Tolhurst, N. Giannini, J.-M. Gaillard, and F. Cagnacci. Understanding and georeferencing animal contacts: proximity sensor networks integrated with gps-based telemetry. *Animal Biotelemetry*, 2016.

[19] J. S. Plank, J. Luo, C. D. Schuman, L. Xu, and Z. Wilcox-O'Hearn. A performance evaluation and examination of open-source erasure coding libraries for storage. In *7th USENIX Conference on File and Storage Technologies (FAST 09)*, San Francisco, CA, 2009. USENIX Association.

[20] J. Proakis. *Digital communications*. McGraw-Hill, Boston, 2008.

[21] Silabs. *EFR32FG1 Flex Gecko Proprietary Protocol SoC Family Data Sheet*, 2017. EFR32FG1P133F256GM48, Revision 1.2 (2017).

[22] P. Sommer, J. Liu, K. Zhao, B. Kusy, R. Jurdak, A. McKeown, and D. Westcott. Information bang for the energy buck: Towards energy- and mobility-aware tracking. In *International Conference on Embedded Wireless Systems and Networks*, EWSN '16, 2016.

[23] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, pages 171–180, New York, NY, USA, 2003. ACM.

[24] K. Vieira Cardoso and J. de Rezende. Accurate hidden markov modeling of packet losses in indoor 802.11 networks. IEEE Communications Letters, 2009.

[25] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys, 2004.