

# Using Erasure Codes to Overcome Reliability Issues in Energy-Constrained Sensor Networks

Margit Mutschlechner\*, Bijun Li<sup>†</sup>, Rüdiger Kapitza<sup>†</sup>, and Falko Dressler\*

\*Computer and Communication Systems, Institute of Computer Science, University of Innsbruck, Austria

<sup>†</sup>Distributed Systems, Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, Germany

{mutschlechner, dressler}@ccs-labs.org

{bli, kapitza}@ibr.cs.tu-bs.de

**Abstract**—We study the use of Erasure Codes (ECs) for transmitting information from mobile sensor nodes to stationary base stations. In particular, we are interested in improving the overall communication reliability of the wireless communication. Our scenario is wildlife monitoring in which bats are equipped with tiny sensor nodes, just being capable to store a few kB of data and to exchange information over a wireless communication link. This link is used, on the one hand, for determining contact times between individuals. On the other hand, these contacts are communicated in aggregated form to stationary base stations. Since the channel quality may vary quickly due to the continuous movements of bats and the heterogeneous environment, the communication is in general assumed to be highly unreliable. Conventional reliability improving approaches such as full data replication or on-demand retransmission are too expensive or even not possible due to very strict energy constraints and asymmetric channels. ECs allow to enhance the reliability of data transmissions by transmitting redundant data. In this work, we investigate the trade-off between reliability achieved and the cost in form of additional transmissions, i.e., the additional energy costs. Our results clearly show that ECs improve the communication reliability considerably with almost no impact on the resulting delay.

## I. INTRODUCTION

The use of sensor networking technology for wildlife monitoring already has quite some tradition. This application provides more sophisticated methods for biologists to study of a specific species, in terms of gathering a huge amount of data by long-term observations. The first projects relied on typical sensor platform as used on labs, e.g., the Great Duck Island project [1], or on special hardware that is even robust enough to be carried by larger animals, e.g., the ZebraNet project [2]. Besides the manpower and other resources saved by employing sensor nodes instead of human beings, they succeeded in maintaining a reliable system with a very high data collection rate.

In more recent activities, heterogeneous sensor nodes have been used for tracking generic animals and endangered species such as Iberian lynx in the surrounding area of wildlife passages, which was built to establish safe ways for animals to cross transportation infrastructures [3]. This system allows target identification through the use of video sensors connected to strategically deployed nodes.

From these successful approaches to wildlife monitoring using sensor network, we learned about hardware design issues, network management, and data collection techniques. In the new BATS project<sup>1</sup> on monitoring the group dynamics of bats in their natural habitat, we go one step further and investigate potentials of ultra-low power sensor systems carried by the bats to monitor contacts between individuals and to track their routes. The aim of the project is to support biologists with their study on bats, one of the most protected species in European Union, to track their living habitats and social behaviors. *Mouse-eared bats* (*Myotis myotis*) are the main study target [4], [5]. The key challenge is that the animals weighing about 20 g can carry sensors of at most 2 g, which strongly limits the available energy budget as well as the computational power and storage capabilities.

The scenario employs mobile nodes which are situated on bats and base nodes on the ground, as shown in Figure 1. All bats continuously exchange contact information but only appear in the communication range of a base station on an irregular basis. If in communication range, they are supposed to upload all contacts. Unfortunately, the channel quality may vary quickly due to the continuous movements of bats and the heterogeneous environment, thus, the communication is in general assumed to be highly unreliable. Conventional reliability improving approaches such as full data replication or on-demand retransmission are too expensive or even not possible due to very strict energy constraints and asymmetric channels. This is of course in conflict with the objective to improve the overall sensor network lifetime [6].

In this paper, we investigate the use of Erasure Codes (ECs) to improve the communication reliability between the mobile nodes and the ground network. Compared to the simplistic approach to send packet replica together with the original packet, ECs offer a better performance with reduced costs. We carefully investigate the performance of three types of ECs in terms of reliability improvement via simulations, since the target hardware is still under development at this moment. One of the most critical characteristics of the used simulation model is the mobility pattern of the bats. We based this model on empirical data provided by biologists.

<sup>1</sup>Dynamically adaptive applications for bat localization using embedded communicating sensor systems, <http://www.for-bats.org/>



Figure 1: The BATS deployment scenario.

According to our findings, we can report several advantages of using ECs, which have not yet been considered in other sensor data upload applications. It is clear that the redundancy introduced by using ECs or data replication will increase energy consumption with the number of additional packets. Simulation results show that with the same or even less overhead compared to classical data replication, ECs can provide a higher degree of reliability in our specific application scenario.

The rest of this paper is organized as follows. Section II outlines related work. An introduction to ECs is given in Section III as well as details about the selected ECs. The simulation is discussed in Section IV, whereas Section V introduces the mobility and contact pattern, which was specifically created for the simulation scenario. Simulation results are shown and analyzed in Section VI. Finally, Section VII concludes this paper and presents future work.

## II. RELATED WORK

Erasure Codes are widely employed to improve the reliability in wireless transmissions. The usage of EC techniques for wireless transmissions without a feedback channel has been investigated in [7]. Here, the optimal trade-off between error-correction coding within packets and erasure-correction coding across packets has been determined. The authors show that the trade-off depends on both the fading statistics and the average Signal to Noise Ratio (SNR) of the wireless channel, where for severe fading channels the trade-off leans towards more redundancy across packets and less redundancy within each packet. Hence, since we are facing a highly unreliable channel, the application of EC adding redundancy across packets is appropriate.

In [8], Random Linear Network Coding (RLNC) has been proposed as a packet-level EC in combination with intra-packet error-correction at the physical layer for low data rate indoor Wireless Sensor Networks (WSNs). The results indicate that RLNC at a code rate of  $r = \frac{4}{8}$  provides an SNR improvement of 3.4 dB and a gain of 5.6 dB when combined with intra-packet error-correction.

Also several studies to compare ECs with traditional reliability enhancing approaches such as data replication and Automatic Repeat Request (ARQ) have been conducted. A

cross-layer methodology for analyzing error control schemes in WSNs has been proposed in [9]. The analysis includes a comprehensive comparison of ARQ and several Forward Error Correction (FEC) codes. The results presented outline that FECs codes are well suited as reliability improvement technique in delay sensitive WSNs since energy consumption and the end-to-end latency is reduced. Furthermore, it has been shown that this improvement can be exploited by employing transmit power control and hop length extension.

However, to the best of our knowledge, there exists no study on the feasibility of ECs for scenarios with spontaneous connectivity such as the scenario we are investigating with its specific channel properties.

## III. ERASURE CODES

An Erasure Code (EC) is a FEC code for the erasure channel that enhances data transmission reliability by introducing redundancy, however, without the overhead of strict replication. In the presented usage scenario, erasures take place on a per-packet basis, hence, ECs are used to introduce inter-packet redundancy.

ECs consist of an encoding and a decoding algorithm. The former one extends a group of  $k$  packets to  $n$  packets by generating  $m = n - k$  redundant packets, where  $k < n$ . Each subset of the  $n$  packets containing at least  $k'$  packets is sufficient to successfully decode the original data, where  $k \leq k'$ . The code rate  $r = \frac{k}{n}$  describes the overhead in terms of redundant packets.

There exist various kinds of ECs. To identify the most suitable EC, we accomplished a study to determine feasible candidates. We then evaluated the most promising ECs with the help of simulations.

### A. EC Selection

In general, ECs can be divided into optimal and nearly-optimal EC. Optimal ECs, such as Reed-Solomon (RS) codes [10], have the property that any  $k$  out of  $n$  packets are sufficient to successfully decode the original data, i.e.,  $k' = k$ . Nearly-optimal ECs, for example Tornado codes [11], introduce a slight overhead such that  $k' = (1 + \epsilon) * k$  packets are required to decode the data successfully, where  $\epsilon > 0$ , hence,  $k' > k$ . However, the encoding and decoding algorithms are less expensive. They have a linear complexity with respect to  $n$ , whereas optimal ECs can have up to quadratic coding complexity for large  $n$ .

In recent years rateless ECs, such as Luby Transform (LT) codes [12] and Rapid Tornado (RAPTOR) codes [13], evolved. These are a special kind of nearly-optimal ECs where the encoding algorithm generates a potentially infinite amount of redundant data without having a fixed code rate. The main advantage emerges in a scenario with multiple receivers, where a feedback channel is present. The encoding entity generates and transmits redundant data up until obtaining a notification about the successful decoding from all receivers. If a receiver holds an insufficient amount of packets, i.e., the amount of received packet is smaller than  $k'$ , it must obtain not yet

received packets in order to be able to decode successfully. Since these might be distinct packets for each receiver, the encoding entity might have to retransmit multiple packets individually for each receiver when using fixed-rate ECs. The encoding algorithm of rateless ECs, however, produces an infinite amount of redundant data, hence, transmitting a newly generated redundant packets is suitable for each receiver. Thus, the amount of transmissions is reduced.

According to our study, which has been confirmed also in [8], optimal ECs are most suitable for the presented scenario. In order to use rateless nearly-optimal ECs effectively, a feedback channel is needed. This, however, is not given in our scenario due to the high mobility of the nodes (cf. Section V-A). Moreover, the encoding entity is highly energy-constrained, hence sending an unlimited amount of redundant data is clearly not feasible. Therefore, rateless ECs have been excluded from the simulation.

Regarding nearly-optimal codes in general, the overhead introduced by  $\epsilon$  is a major drawback in our scenario. The necessary value for  $\epsilon$  increases as the amount of original data  $k$  decreases. Hence, the amount of data required for decoding  $k'$  is growing if  $k$  decreases. To obtain a low overhead,  $k$  is supposed to have a large value, however, this is not achievable by the highly energy-constrained mobile nodes responsible for the encoding. Our mobile nodes have a restricted amount of storage (a few 100 kB) due to the highly limited node weight and size. Furthermore, mobile nodes may have only infrequent contact to the base network, hence, waiting up until enough data is gathered to get a large value for  $k$  might result in forfeit the rare communication possibilities.

In contrast, we have the drawback of optimal ECs exhibiting a higher coding complexity for large  $n$ . However, since the amount of original and redundant data is supposed to be very small this is negligibly low for our scenario. Therefore, we mainly focus on optimal ECs, using a selected nearly-optimal EC for comparison purposes. In particular, we rely on the following existing open source implementations:

- Cauchy: a RS code based on a Cauchy matrix, developed by Michael Luby [14]
- Vandermonde: a RS code based on a Vandermonde matrix, developed by Luigi Rizzo [15]
- Tornado: a nearly-optimal EC, developed by Michael Noisternig [16]

Each one of the three implementations has been evaluated with four different code rates:  $r = \{\frac{4}{5}, \frac{4}{6}, \frac{4}{7}, \frac{4}{8}\}$ . As a baseline experiment, we simulated the scenario with no reliability improvement, i.e., data is sent without encoding, as well as the full replication idea, i.e., data is sent together with an exact replica to increase reliability.

## B. Coding Algorithms

The significant difference between the various ECs is the mathematical background of the encoding and decoding algorithms. The two RS codes Cauchy and Vandermonde share the same algorithms, however, they work on different kinds of

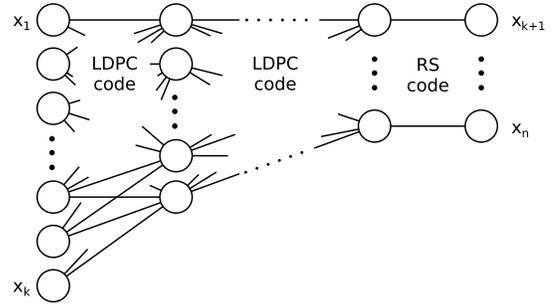


Figure 2: Encoding process of Tornado.

matrices, whereas Tornado varies significantly in the algorithm itself.

RS codes are cyclic block codes that split the original data  $x$  into  $k$  equally sized blocks  $x_1 \dots x_k$ . These blocks are considered as the coefficients of a polynomial over a finite field  $F$ :

$$P_x(c) = \sum_{i=1}^k x_i c^{i-1} \quad (1)$$

The encoding algorithm extrapolates it at  $n$  distinct sampling points  $P_x(c_1) \dots P_x(c_n)$ , where the first  $k$  points  $P_x(c_1) \dots P_x(c_k)$  correspond to the original blocks (thus, successfully receiving the first  $k$  blocks corresponds to an error free transmission of the original messages). This encoding function is a linear mapping and can be realized as  $x \rightarrow x \times A$ , where  $A$  is a  $k \times n$  generator matrix with elements from  $F$ . Therein lies the difference between the two chosen RS codes since they use a Cauchy and a Vandermonde matrix, respectively. The decoding algorithm inverts the encoding by interpolating over some of the values of  $P_x(c_1) \dots P_x(c_n)$ , where at least  $k$  out of the  $n$  sampled points are needed to recover the original blocks. The encoding and decoding algorithms have a complexity of  $\mathcal{O}(n \cdot \log n)$  and  $\mathcal{O}(n^2)$ , respectively, where the computational effort of the decoding algorithm is not crucial since only the encoding is performed by the energy-constrained nodes.

In contrast, the encoding and decoding algorithms of Tornado are based on a bipartite irregular graph. The original data is split again into  $k$  equally sized blocks  $x_1 \dots x_k$ , each one represented by one node in the graph. The algorithm is realized in multiple levels visualized in Figure 2.

Each level except the last one performs a Low-Density Parity-Check (LDPC) code by combining multiple nodes of the graph with an inexpensive XOR operation to generate redundant data. For each level the nodes itself can be chosen randomly without repetition, however, the total amount is given by a specific distribution for each level. This operation is performed recursively until the final level encodes the nodes with a RS code. Since each level reduces the number of nodes, the complex operations during the final level have to be performed on a much smaller subset.

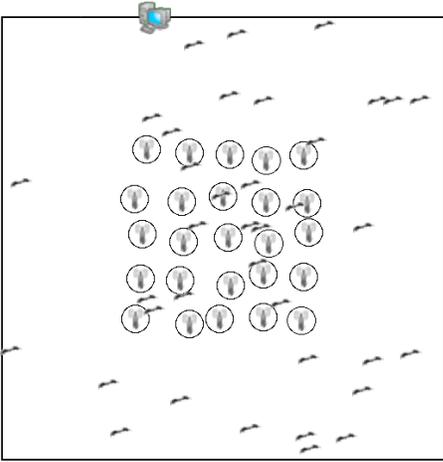


Figure 3: A two-dimensional overview of the simulated scenario.

Decoding reverses the encoding algorithm by executing the RS decoding algorithm before applying the XOR operations starting from the second to last level. Both the encoded and decoded algorithms have a complexity of  $\mathcal{O}(n \cdot \ln \frac{1}{\epsilon})$ , where  $\epsilon$  is a positive constant representing the overhead needed for decoding.

#### IV. SIMULATION MODEL

All the simulations are based on the OMNeT++ simulation core [17], a discrete and event-based simulation framework. For simulating the wireless channel, we used the MiXiM framework [18], which provides all the means for accurate wireless simulation and the integration of mobility models.

##### A. Scenario Setup

The simulation scenario is depicted in Figure 3, which reflects the envisioned deployment scenario. The scenario consists of three node types: mobile nodes situated on bats, base nodes laid out in the habitat form a stationary backbone network, and a single central station is used as an information sink. In total 40 mobile nodes and a base network of 25 nodes are deployed. The base nodes are arranged in form of a grid-shaped but not fully regular manner in an area of  $320 \text{ m} \times 320 \text{ m}$  with a distance of 30 m between each other.

The movements of mobile nodes are based on a predefined mobility model that resembles the flying behavior of bats, whereas base nodes are stationary with a fixed position across all simulations. The mobility model of the mobile nodes is described and discussed in Section V-A.

##### B. Communication Protocols

Mobile nodes periodically broadcast beacons of one byte containing their unique identifiers to inform nearby mobile nodes about their existence. The interval between two beacons was set to be 0.5 s. Upon reception of a beacon, the receiving mobile node stores this information to build up a rendezvous table. Each row of this table consists of the identifier contained

in the beacon, the time when the first beacon from this node was received, and the time interval for which beacons were continuously received. Two beacons are considered to be received continuously if the difference of their arrival time is smaller than a pre-defined threshold, which is set to 2 s.

Each row of the rendezvous table is transmitted in an individual data packet to the base network. The transmission of data packets has to be reliable, whereas the successful reception of beacons is not as critical since the quantity of useful information within a beacon is comparably less than within a data packet. Therefore, ECs are used only for data packet transmissions.

When a mobile node is within the transmission range of at least one base node, it initializes the data transmission. This simulates the envisioned use of multi-stage wake-up receivers that power-up the radio of the mobile nodes only if located within the communication range of a base node where the wake-up signal can be received.

When a mobile node is in transmission range of a base node and the number of table entries reaches a pre-defined threshold the encoding process is triggered. Afterwards, the encoded data is sent in multiple data packets. This threshold is set according to the used ECs. A single data packet has a size of 32 B, where 30 B are reserved for the payload, which is either one encoded chunk or one entry of the rendezvous table in plain text without using any EC. The overhead of two bytes is formed by the mobile node's identifier of one byte and the needed metadata of the EC. This metadata consists of the chunk index (3 bit) and an identifier of which encoding process this chunk belongs to (5 bit).

Due to the mobile node's high speed and the rapidly changing environment the contact times between mobile nodes and base nodes can be very small as shown in Section V-B. Therefore, no carrier sensing techniques are performed prior to transmission since this could prevent the mobile node from sending data before exiting the transmission range. Moreover, the mobile nodes are highly constrained in terms of computational power and energy, hence, very complex protocols are not applicable. Instead, a wake-up receiver powered-up by a signal from the base nodes initializes the data transmission.

Upon receiving a data packet from a mobile node, base nodes store this information for decoding. For each received data packet, the central station records its reception and, if the threshold for a successful decoding is reached, it tries to recover the original data.

The communication channel resulting from this specific mobility model represents the channel in the real deployment scenario we are facing. To simulate realistic data transmissions over the wireless channel we chose a pathloss propagation model attenuating signals with the distance  $d$ :  $\frac{1}{d^3} \left(\frac{c}{4\pi f}\right)^2$ , where  $c$  denotes the speed of light and  $f$  the frequency of the signal. Moreover, we integrated a log-normal shadowing on top of it to simulate objects obstructing the signal. This can be motivated with the fact that we are facing multiple shadowing objects in the envisioned deployment scenario, i.e. trees. Each such object contributes a random multiplicative factor to the shadowing,

which leads to a normal distribution when converted to dB [19].

### C. Evaluation

The analysis of the simulation results is twofold. On the one hand, we analyze the mobility model of mobile nodes and the thereby resulting contact times in Sections V-A and V-B, respectively. On the other hand, we investigate the usability of ECs based on three metrics: reliability, energy consumption and delay. To evaluate the impact on the reliability we compare the amount of received data with the amount of recovered data in Section VI-B. The influence on the energy consumption is evaluated in Section VI-C by comparing the number of sent packets with the amount of recovered data. In terms of delay the question arises as to whether ECs slow down data transmission. We study these effects in Section VI-D, considering the time interval from the creation of an entry in the rendezvous table at the mobile node up until it is recovered by the base network.

A further impact of ECs is the increased memory usage on the mobile nodes due to the storage of the redundant data as memory space is highly limited. In case of insufficient memory we discard redundant packets since the original data is more crucial. However, during simulations memory was never exhausted due to the specific selection and configuration of the ECs explained in Section III-A. Therefore, we do not investigate this issue in more details.

For each configuration 20 repetitions were performed. Each repetition of one configuration is initialized with a unique random seed, however, the  $n^{\text{th}}$  repetition of each configuration has the same  $n^{\text{th}}$  random seed. In the following we are going to present our evaluation results. The variance of the recorded metrics was so small that we only plot the average values in the respective figures.

## V. MOBILITY AND CONTACT PATTERN

The mobility model of mobile nodes has a high impact on the results of the simulation. It influences both the quality of the communication channel as well as the duration of the communication between mobile and base nodes. In the real deployment scenario each mobile node corresponds to a *greater mouse-eared bats* (*Myotis myotis*), and the simulated area resembles the foraging patch. Therefore, in order to have realistic results, the mobility model of mobile nodes must resemble the flying behavior of this species during foraging in the most realistic way.

### A. Mobility Model

Although the specific characteristics of bat movements are not completely known yet, various figures in terms of flight speed, flight height and flight routes are known nowadays [4], [5]. Furthermore, differences to the flying behavior of birds can be found [20]. To the best of our knowledge there exists no mobility model for the specific flight of bats during foraging, however, there are various mobility models suitable as basis.

We decided to adapt an existing mobility model to fit the special characteristics of bat flights. The Lévi flight model is well-known for describing the movements of free-living

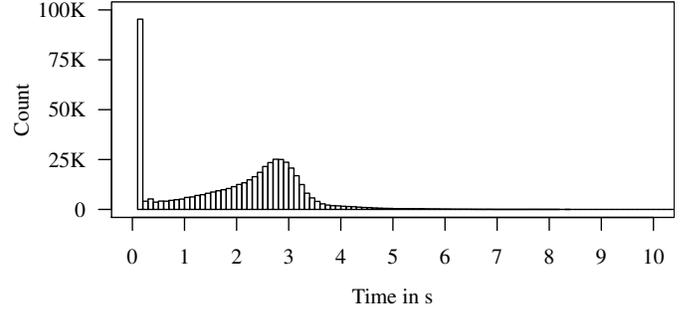


Figure 4: Histogram of the contact times.

animals, and it is shown to resemble the flight of birds [21]. It is based on the random waypoint model with a heavy-tailed probability distribution for the step length. Therefore, we decided to rely on the random waypoint model, which was extended by the foraging movement patterns of *Mouse-eared bats* (*Myotis myotis*) described in [4], [5].

Each mobile node, i.e., the bat, starts in the foraging state where it moves according to the random waypoint model with  $15 \text{ km h}^{-1}$  to  $35 \text{ km h}^{-1}$  at a height of 0.3 m to 0.7 m. A bat accelerates with  $2 \text{ m/s}^2$  and decelerates with  $4 \text{ m/s}^2$  towards a target speed, which, in our model, is randomly chosen from the aforementioned range. Within 0.1 s to 100 s, it starts the capturing phase. First, it moves to a position at the ground within a radius of 0.5 m to catch the prey. Then, it resembles the eating behavior by moving in circles with a radius of 2 m to 4 m at a speed of 15 km on a height of 5 m to 15 m for 10 s to 20 s. Afterwards, the bat switches back to the foraging state.

In order to obtain reproducible results but still to exploit all the variations in the distribution of the needed random variables, we decided to run different mobility pattern using different seeds for the mobility related random variables. Therefore, mobile nodes move distinctly in each repetition of the simulation, whereas their movements are the same for the  $n^{\text{th}}$  repetition of a certain configuration.

### B. Contact Possibilities

One crucial property for the communication protocol from an overall perspective are the contact possibilities, i.e., the time intervals in which communication between mobile nodes and the base network is possible. We investigated these contact times before finally assessing the performance of the EC based communication.

A histogram of the contact times is given in Figure 4. It summarizes the contact time for all 20 repetitions as the movement pattern of mobile nodes differ for each. The minimum possible contact time is 0.1 s, which is a simulation artifact since the transmission range of a mobile node towards the base network is checked in discrete time intervals of 0.1 s.

As the figure indicates, the contact time is mainly short-termed. Most contacts have a duration of less than 3 s. Since mobile nodes tend to leave the communication range of a base node quickly after entering, the amount of data that can be

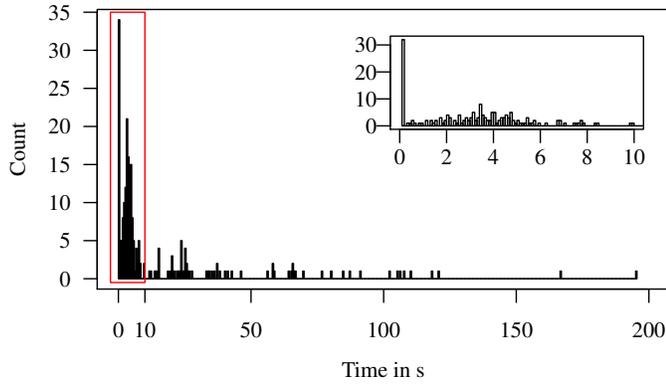


Figure 5: Histogram of the inter-contact times.

transmitted is limited.

A second measure of interest is the inter-contact time, i.e., the time intervals in which no communication between a mobile node and the base network has been possible. Figure 5 visualizes a complete histogram of the inter-contact times as well as a histogram enlarging the distribution for the first 10 s, again summarizing all 20 repetitions. Most inter-contact time has a duration of less than 10 s. We already discussed the high peak at 0.1 s as being a simulation artifact. The distribution is highly heavy-tailed with measures of up to 200 s.

## VI. COMMUNICATION PERFORMANCE

In the following, we investigate the performance of the communication protocols in more detail. We concentrate on the reliability vs. overhead issue but also look into energy consumption and delay questions.

### A. Transmitted Data

To investigate the quality of the channel, we consider the relationship between sent and received packets. All packets received erroneously or not received at all were considered to be lost. Figure 6 shows this relationship by visualizing both the amount of sent packets containing original data as well as the amount of sent packets containing redundant information. Furthermore, the amount of received packets of both types is indicated by shaded areas. All numbers are relative to the amount of sent data, i.e., 100 % corresponds to the amount of sent data. The figure summarizes the results of all 20 repetitions, however, the variance is negligibly low with at most 5.503 %.

Apparently, when using reliability improvement techniques the amount of sent redundant packets increases according to the data rate, whereas the amount of sent original packets stays constant. For both packet types the data reception rate stays constant at about 57 %, independent of the total amount of transmitted packets. Therefore, we argue that the number of packet transmissions does not influence the reception of packets, i.e., transmissions do not notably interfere with each other. Hence, we conclude that the channel is hardly saturated, which is a result of the very limited transmission range of mobile nodes and their sparse distribution.

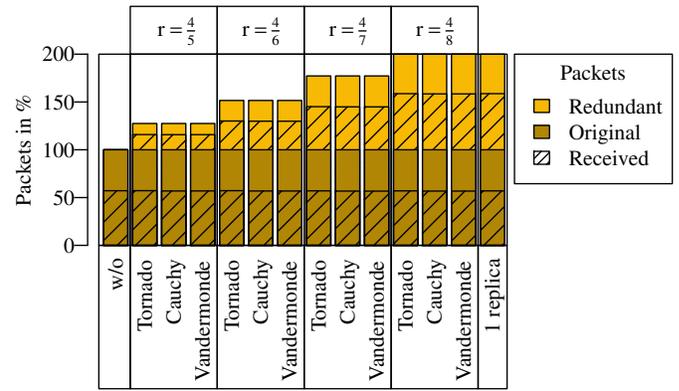


Figure 6: Sent and received packets relative to the total number of transmitted data.

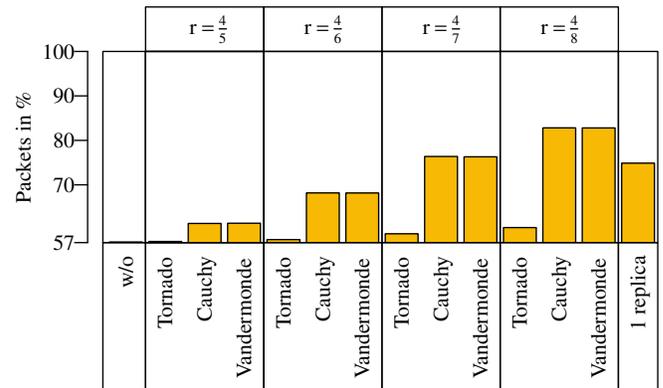


Figure 7: Recovered data relative to the total number of transmitted data.

The high packet loss is mainly caused by the highly unreliable channel. These characteristics correspond to the channel we are facing in the real deployment scenario where the fast moving bats and the very heterogeneous and rapidly changing environment leads to a highly varying channel quality.

### B. Reliability

The improvement of ECs on the transmission reliability is analyzed based on the ratio of received and recovered data compared to the amount of sent data. This is visualized in Figure 7, where all numbers are relative to the amount of sent data, i.e., 100 % corresponds to the amount of sent data or a 100 % reception rate.

A first observation is the low but stable data reception rate, which we already discussed before. More importantly, the data reception rate greatly increases when reliability improvement techniques are involved.

Apparently, without using ECs or replication techniques we are not able to recover any data not received by the base network. Using simple duplicates (full replication approach), the base network is able to recover 18 % of the data from the redundant packets, so in total 75 % of the data can be recovered. However, the drawback is a highly increased

power consumption since the number of transmitted packets is increased by 100% compared to the gain of only 18%.

When using ECs, we can observe a steadily growing amount of recovered data with an increasing code rate. Furthermore, we see a huge difference between the performance of Tornado and the two RS codes Cauchy and Vandermonde.

Tornado slightly increases the amount of recovered data, which means redundant data does not noticeably improve the reliability. Even with a code rate of  $r = \frac{4}{8}$ , Tornado performs worse than replicated sending, although the same amount of packets are transmitted. The poor behavior is due to the different encoding algorithm of Tornado in contrast to RS codes. On the one hand, the threshold for successful decoding is slightly higher for Tornado than it is for RS codes. Furthermore, Tornado is designed to work on a binary erasure channel, which means losses are assumed to be equally distributed. However, the simulations show that base nodes are facing bursty losses.

Cauchy and Vandermonde perform exactly the same regarding reliability, which is due to the similar encoding and decoding algorithms. At a code rate of  $r = \frac{4}{8}$  both can recover 83% of the sent data. It increases the data recovery rate by 8% compared to the replicated sending, although the same amount of packets is transmitted. Of course, as the data rate decreases also the amount of recovered data is reduced.

However, even with a code rate of  $r = \frac{4}{7}$ , Cauchy and Vandermonde are able to recover 2% more than the replicated sending approach, although the overhead in terms of redundant packets is reduced by approximately 25%. This shows that RS codes clearly outperform replicated sending in terms of reliability.

With an even smaller data rate of  $r = \frac{4}{6}$  and  $r = \frac{4}{5}$  the RS codes are able to recover 68% and 61%, respectively. Compared to replicated sending the impact on reliability is reduced by 7% and 14%, respectively, but at the same time the overhead of transmitted packets is decreased by 50% and 75%, respectively.

### C. Energy Efficiency

The usage of ECs and replicated sending inevitably increases energy consumption. Primarily the sending of redundant packets drains energy, however, in the former case the execution of the encoding algorithm has to be taken into consideration as well. We have already discussed the coding efficiency of different ECs in Section III.

As the main energy draining task is the sending of redundant packets, we focus mainly thereon to decide which performance improvement technique is the most feasible. The trade-off between the improved reliability and the overhead caused by redundant packets is outlined in Figure 8. The graph summarizes the results presented in the previous section and gives an overview of the suitability of the presented reliability improvement techniques. As we move from left to right in the graph, reliability measured against the amount of recovered data increases, whereas moving from bottom to top the energy efficiency increases with a decreasing overhead. The theoretical optimum would be on the top-right corner, meaning that no

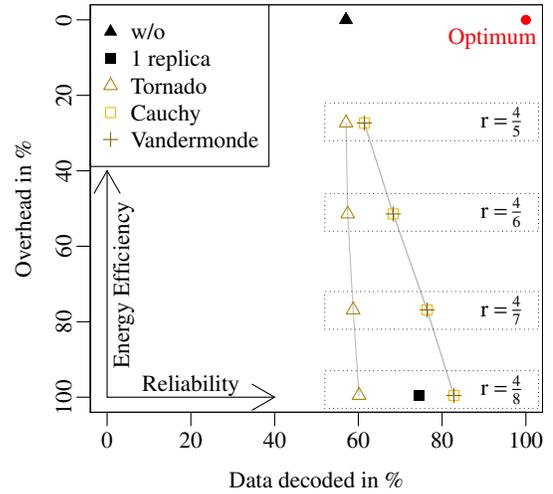


Figure 8: Reliability of the five transmission strategies versus their energy efficiency in terms of packet transmissions.

energy is spent by means of overhead in the packet transmission but still all packets can be successfully received.

As was already observed, Tornado as well as replicated sending perform poorly compared to the two RS codes. The remaining question as to which data rate for Cauchy and Vandermonde is most feasible strongly depends on the final application. If a high level of reliability must be achieved, this comes to the cost of a reduced network lifetime. Apparently, absolute values for the energy consumed during sending and encoding cannot be provided yet since it depends on the underlying hardware, which is currently under construction.

### D. Communication Delay

The final measure to compare the various performance improvement techniques is the duration from the creation of a rendezvous table entry until a base node receives or recovers the data, i.e., the overall communication delay. Only the delay of data received by at least one base node is taken into consideration. We assumed that, given the rather short contact times and the substantial overhead of ECs for higher code rates, the resulting delay will clearly increase.

An Empirical Cumulative Density Function (eCDF) of the distribution of the data delay up to 200s is visualized in Figure 9. Furthermore, the figure contains an excerpt of the cumulative density up to 5s. The median of the experienced delay is in the range of 8.18s to 9.26s, depending on the used algorithm. However, the distribution is highly heavy-tailed, which can be explained by considering the inter-contact time. Mobile nodes mostly have short inter-contact time, hence, even if they are not in transmission range of the base network they reenter after short time periods and are able to transmit data with small delays. However, the high upwards outliers increase the data delay by orders of magnitude.

No notable difference between the various transmission strategies can be seen for higher delays, i.e., the data delay is almost independent of the reliability improvement technique

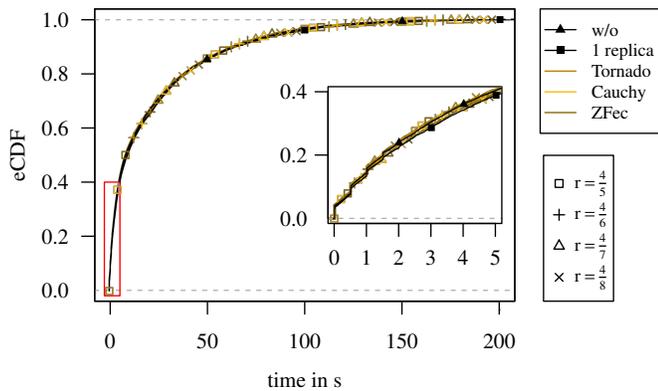


Figure 9: eCDF of the data delay up to 5 s and 200 s.

and the data rate. When focusing on smaller delay values, the values are slightly smaller when no reliability improvement technique is used, whereas the replicated sending approach gives the upper bound. The communication delay for all ECs at the various code rates is within these two bounds. This marginal increase is acceptable given the improved reliability.

## VII. CONCLUSION

In this paper, we studied the usage of Erasure Codes (ECs) to improve the reliability of the data transmission in a sensor network scenario featuring high mobility of nodes that are used to observe the group dynamics of bats in their natural habits. We developed a novel mobility model for the mobile nodes, which resembles the flying behavior of the target species, and analyzed the resulting communication possibilities. This scenario features rather short contact times between the mobile nodes and stationary base nodes that are used to upload contact information a bat collected for further processing. The radio channel varies quickly due to the speed of the animals. The use of reliable communication using acknowledgments is therefore not feasible.

We identified three feasible ECs and studied their performance in detail. As a baseline, we simply send the collected data with no additional reliability improvement techniques in place as well as full replication, i.e., sending all data items twice. We showed that the nearly-optimal EC Tornado does not show a noticeable improvement on the data transmission, whereas the selected Reed-Solomon (RS) codes Cauchy and Vandermonde increase the reception rate up to 26%. Moreover, according to our results, EC do not have a critical impact on the overall transmission delay. This improvement comes with the cost of an overhead due to additional data messages that need to be send. Yet, the RS-based ECs clearly outperform simple data replication. In summary, the results show that ECs provide a significant reliability improvement with an acceptable overhead for our extremely energy-constrained hardware platform.

## VIII. ACKNOWLEDGMENTS

This work has been supported by the German Research Foundation (DFG) under grants no. FOR 1508.

## REFERENCES

- [1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *1st ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, September 2002.
- [2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," *ACM SIGOPS Operating Systems Review*, vol. 36, no. 5, pp. 96–107, December 2002.
- [3] A.-J. Garcia-Sanchez, F. Garcia-Sanchez, F. Losilla, P. Kulakowski, J. Garcia-Haro, A. Rodr'iguez, J.-V. L'opez-Bao, and F. Palomares, "Wireless Sensor Network Deployment for Monitoring Wildlife Passages," *Sensors*, vol. 10, no. 8, pp. 7236–7262, 2010.
- [4] R. Arlettaz, "Feeding behaviour and foraging strategy of free-living mouse-eared bats, *Myotis myotis* and *Myotis blythii*," *Animal Behaviour*, vol. 51, no. 1, pp. 1–11, 1996.
- [5] B.-U. Rudolph, A. Liegl, and O. Von Helversen, "Habitat Selection and Activity Patterns in the Greater Mouse-Eared Bat *Myotis myotis*," *Acta Chiropterologica*, vol. 11, no. 2, pp. 351–361, 2009.
- [6] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–39, February 2009.
- [7] C. R. Berger, S. Zhou, Y. Wen, P. Willett, and K. Pattipati, "Optimizing Joint Erasure- and Error-Correction Coding for Wireless Packet Transmissions," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4586–4595, 2008.
- [8] G. Angelopoulos, A. Paidimarri, A. P. Chandrakasan, and M. M'edard, "Experimental Study of the Interplay of Channel and Network Coding in Low Power Sensor Applications," in *IEEE International Conference on Communications (ICC 2013)*. IEEE, June 2013, pp. 5126–5130.
- [9] M. C. Vuran and I. F. Akyildiz, "Error Control in Wireless Sensor Networks: A Cross Layer Analysis," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1186–1199, 2009.
- [10] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial & Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
- [11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient Erasure Correcting Codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.
- [12] M. Luby, "LT Codes," in *43rd Symposium on Foundations of Computer Science (FOCS 2002)*, 2002, pp. 271–280.
- [13] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.
- [14] J. Blömer, M. Kalfane, R. Karp, M. Karpinski, M. Luby, and D. Zuckerman, "An XOR-Based Erasure-Resilient Coding Scheme," *International Computer Science Institute, Berkeley, Tech. Rep. TR-95.048*, August 1995.
- [15] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 27, no. 2, pp. 24–36, April 1997.
- [16] M. Noisternig, "Tornado-Codes," <http://notion.muelln-kommune.net/cgi/tornado-paper.ps.cgi>, 2004.
- [17] A. Varga, "The OMNeT++ Discrete Event Simulation System," in *European Simulation Multiconference (ESM 2001)*, Prague, Czech Republic, June 2001.
- [18] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Hanefeld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ – The MiXiM Vision," in *1st ACM/ICST International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008): 1st ACM/ICST International Workshop on OMNeT++ (OMNeT++ 2008)*. Marseille, France: ACM, March 2008.
- [19] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, January 1995.
- [20] A. Hedenström, L. C. Johansson, and G. R. Spedding, "Bird or bat: comparing airframe design and flight performance," *Bioinspiration & Biomimetics*, vol. 4, no. 1, p. 015001, 2009.
- [21] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, May 1996.