Coding Schemes for Heterogeneous Communication Links Using Channel Bundling

Vanessa Eichhorn(✉), Maciej Mühleisen, and Andreas Timm-Giel

Institute of Communication Networks (ComNets),
Hamburg University of Technology, Hamburg, Germany
{vanessa.eichhorn,maciej.muehleisen,timm-giel}@tuhh.de

Abstract. Communication over geostationary satellite links is improved by introducing end-to-end Forward Error Correction (FEC) and simultaneous transmissions over two links (channel bundling). The main objective of this work is to investigate to which degree the goodput and the reliability can be enhanced using the mentioned techniques. The performances of the two FEC schemes Reed-Solomon Codes and Random Linear Network Coding are compared. Uncorrelated and correlated packet errors are considered, the latter with a Gilbert-Elliot channel model. Experiments are conducted in a testbed consisting of a single PC with virtual network interfaces to determine the influence of various parameter settings on performance. Results are compared against a scenario with one link offering the same capacity as the two links together. It is concluded that using two heterogeneous links is beneficial for the goodput and losses for generation sizes larger than 20 for three correlated lost packets on average.

Keywords: Forward Error Correction · Random Linear Network Coding · Reed-Solomon codes · Gilbert-Elliot channel model · Satellite communication

1 Introduction

Reliable communication is a major goal for many applications in today's networks. A common approach to achieve reliable per hop and end-to-end transmission is the use of Automatic Repeat reQuest (ARQ), which is a Backward Error Correction (BEC) scheme. Here, retransmissions generate extra delays and therefore significantly decrease the performance in high delay scenarios such as satellite communications. ARQ also requires resources on the backchannel and is not feasible for multicast. Forward Error Correction (FEC) can be an alternative or complement to ARQ. It is a common approach to gain reliability in lower layers through introduction of channel coding on point-to-point links. Redundancy is added at the sender allowing reconstruction of the original packet even with some bit errors. Coding schemes can also be introduced in higher layers across several hops. There redundancy is added by packet-level FEC schemes generating extra packets to recover lost ones. A trade-off must be found between added

R. Agüero et al. (Eds.): MONAMI 2016, LNICST 191, pp. 160–173, 2017.
DOI: 10.1007/978-3-319-52712-3_12

additional packets and the gain to be expected in reduced data loss and increased goodput. Coding results in en- and decoding delays as well as additional bandwidth usage. Furthermore, channel bundling can increase the throughput and also adds diversity. This is in particular beneficial, if the channels show a burst error behavior. If the different channels are uncorrelated to each other, substantial improvement can be expected. During bad channel conditions in one link, the other link still is expected to show good performance.

This paper is organized as follows: In Sect. 2 an overview of related work is given, with an introduction to two types of linear codes used in this work. Afterwards, the evaluated scenario is presented in Sect. 3 and the results are given in Sect. 4. Finally, conclusions are drawn and future work is described in Sect. 5.

2 Overview of Coding Schemes and Related Work

In the following, the basic concept of linear codes and their applications is presented. The use of these codes in previous work is also discussed. Finally, the Gilbert-Elliot model is explained to address how to model a transmission channel with correlated errors.

2.1 Coding Schemes

Two coding schemes will be evaluated in this paper: Packet-level Reed-Solomon (RS) codes and Random Linear Network Coding (RLNC). Both schemes are Linear Codes. This codes use the properties of linear algebra [1]. A (n, k)-linear block code is specified most of all by the choice of the so-called generator matrix G. The code can be written as $Y = X \times G$. X denotes a (m, k)-matrix with packet length m and number of original packets k. G is the (k, n)-generator matrix, where n is the number of coded packets. Y represents the (m, n)-matrix of n coded packets with length m.

The decoder solves a linear equation system $\hat{X} = \hat{Y} \times \hat{G}^{-1}$ for any k received linearly independent packets. In this case, $\hat{G}$ describes a (k, k)-invertible matrix. The received packets are placed in the matrix $\hat{Y}$. The generator matrix coefficients must be known to the decoder.

Computations are performed in a finite (or Galois) field GF. A field is closed under addition and multiplication, so the result still is part of the field. A finite field has a finite number of elements meaning that the results only needs as much bits for representation as the original data [1]. Usually a finite prime field $GF(q = p^r)$ is used with p prime and r being a positive integer. Using packet erasure channels, p equals 2.

For encoding, every packet is split into smaller chunks of length r. These are separately multiplied with the coding coefficients in the generator matrix which results in encoding delay.

There are some sources for additional decoding delay. First, there is the processing delay at the receiver, as Gaussian elimination algorithm for matrix inversion and costly multiplications have to be done. A second source is the fact that a generation can only be decoded after receiving enough packets.

Reed-Solomon Codes. Using a (n, k)-Reed-Solomon code, k source packets defined over a finite field $GF(q = 2^r)$ are encoded to n coded packets. The number of different coded packets n is upper bounded by $q - 1$. The generator matrix is built from a Vandermonde matrix $V_{k,n}$ [2]. The matrix consists of $v_{i,j} = \alpha^{i \cdot j}$, where $0 \leq i \leq k - 1$ and $0 \leq j \leq n - 1$. α is a fixed root of the primitive polynomial of degree r [2]. This matrix is transformed to a systematic matrix. The code rate is fixed and defined as k/n. The decoder does not need to receive the coding coefficients from the sender, as the generator matrix is fixed for different r. Nevertheless, the decoder needs the index of the generator matrix used for encoding. Protocols can either use a single matrix or define a set of matrices and point at the used one on session establishment.

RS codes are Maximum Distance Separable (MDS) codes. Therefore, there exists no other FEC coding scheme that is able to recover lost packets from fewer received coded packets [3]. The computational complexity increases with the use of larger finite fields, so e.g. [4] focuses on $GF(2^8)$ only. The field size of $GF(2)$ cannot be chosen in RS as the number of different coded packets per generation would be upper bounded by 1. The minimum field size is $GF(2^2)$.

The minimum number of redundant packets h should be chosen depending on the measured or estimated loss rate of the channel. Thus, $h = (k \cdot p)/(1 - p) = p \cdot n$ with $0 \leq p \leq 1$ and $n = k + h$. The generator matrix can be computed with complexity $\mathcal{O}((n - k) \cdot k \cdot (log(k))^2)$ [2]. Then for encoding, k additional operations per vector-matrix multiplication are needed. For decoding using Gaussian elimination algorithm, the matrix inversion takes $\mathcal{O}(k^3)$ operations and the vector-matrix multiplication requires $O(k^2)$ operations.

Random Linear Network Codes. In RLNC k original packets are encoded into n coded packets. The parameter k is denoted as generation size. The field size q describes the size of the Galois field from which the coding coefficients are chosen. The case $q = 2$ is possible in RLNC and is called binary coding. In binary coding, either a packet is chosen to be mixed or not, thus a coded packet formed of original packets A, B and C is, e.g., $1A \oplus 0B \oplus 1C$. In the case of $q = 2^8$ the coefficients are chosen between 0 and 255. Thus, a packet like $255A \oplus 3B \oplus 145C$ is possible.

As soon as the decoder receives k linearly independent packets, decoding can be performed. For decoding to be possible, the receiver must know the coding coefficients used to create the coded packet. Therefore, a coding vector in the header is necessary. In [5], it was shown that choosing the coding coefficients independently and uniformly at random from elements of a finite field is sufficient. The size of the coefficients in the header is $k\ log_2(q)$ bits in total. It does not matter exactly which coded packets are received, as long as there are enough linearly independent packets for decoding. So the system is stateless.

There is a trade-off between the computational complexity of encoding and decoding, introduced overheads and the residual error probability. On the one hand, coding information (the random generator matrix coefficients) must be known to the receiver and are therefore added to the header of each packet. On the other hand, overhead is introduced due to linearly dependent packets resulting from unfortunate random number constellations and not adding any information for decoding. Thus, packets can be linearly dependent because of the randomness of the code.

In network coding each node in a network can generate new coded packets and forward them, instead of only storing and forwarding packets. *Recoding* at intermediate nodes means mixing different coded packets without decoding them first [6]. If no recoding option at nodes in the network is needed, there is the possibility to exchange once a seed at the beginning of the transmission. Then the random coefficients do not have to be sent over the network in every packet. Thus, the overhead is reduced. Using this method, the number of unique coding vectors is reduced to the size of the seed [6].

Comparison. The field size q defines the number of unique field elements. In RLNC, a large field size has the advantage that the packets are linearly independent with a high probability. Therefore, the number of additional redundant packets can be reduced compared to using small field sizes. Apart from that, a high field size in RLNC results in a large coding vector added to the header of each packet with a size of $k\ log_2(q)$ bits in total. In RS the field size upper bounds the number of different packets per generation. Furthermore, the computational complexity grows with the field size. Addition in $GF(q = 2^r)$ can be implemented with complexity $\mathcal{O}(r)$, whereas multiplication requires $\mathcal{O}(r^2)$ operations [7]. Multiplication runtime can be improved by using look-up tables. The generation size k defines the maximal number of packets that can be mixed to generate a coded packet. In case a high generation size is chosen, it is possible to mix many packets into the coded packet. This leads to a large decoding delay, because at least k coded packets must be received before decoding can be performed. From this perspective, high generation sizes can be chosen for a file download, whereas for streaming live events a small generation size is preferable to reduce delay [8]. Decoding is performed by Gaussian elimination algorithm. The computational complexity increases with $\mathcal{O}(k^3)$ [9].

In RLNC and RS, the number of additional packets h should be chosen at least according to the packet loss probability of the link. In RLNC it can be adapted flexible to the link conditions if feedback is available. Then it is possible to increase the number of additional packets h according to the new estimated error probability of the link. Systematic coding means that the original packets are part of the coded packets. This option is possible in both RLNC and RS. In case systematic coding is used, the encoding process can be speeded up because for the first k packets no encoding is needed. This is especially beneficial if there are only a few losses and the number of required coded packets is small.

For RLNC, there are multiple variants which have to be evaluated depending on the scenario, e.g. also sparse codes. Using sparse codes a lot of coding coefficients are set to zero and therefore reduces computational complexity. This means that in a coded packet only a few packets of the generation are mixed.

2.2 Related Work

In [9], a channel bundling scenario was analyzed over multiple wireless interfaces with half-duplex constraints. In a testbed a file transfer was implemented between two Android smartphones using Bluetooth, WiFi and cellular networks. Instead of sending data through a single interface at a time, splitting the data across interfaces or repeating the same data over multiple links, the data was coded at packet-level using RLNC. The throughput was increased by channel bundling and by making transmissions more robust.

In [10], an example was discussed with a RS code considered at packet-level within the DVB-H standard. Gaussian elimination algorithm was compared to Berlekamp-Massey algorithm for decoding. It was shown that the complexity of both algorithms is similar for small packets considered in the standard.

In [3], RS codes were used for reliable multisource video streaming. This work uses an extended Gilbert-Elliot model. The system dynamically choose one of four different FEC schemes to adapt to the network conditions. Besides pure RS, an unequally interleaved FEC for correlated packet losses was considered. Here, RS codes were extended by a specific uneven FEC interleaving scheme, which needs feedback through the backchannel. In [11], Reed-Solomon codes were analyzed analytically using a Gilbert-Elliot channel model and interleaving.

In [4], the behavior of packet-level RS code was analyzed in a real-time video streaming application. It was found that using RS results in high CPU loads. The real-time performance constraints were not met in scenarios with high losses.

Compared to the previous work, the contribution of this paper is to investigate the use of pure FEC schemes without feedback and using channel bundling for links with correlated and uncorrelated losses.

2.3 Transmission Channel Models

Two kinds of packet erasure channels are investigated in this paper. First, a channel with uncorrelated errors is considered. Second, a channel with correlated errors is evaluated. Either the whole packet without error is received or the packet is lost in packet erasure channels.

The Gilbert-Elliot channel model is used to describe correlated burst errors in the wireless channel. In [12], it is validated that the two-state Gilbert-Elliot model is suitable for satellite channels. This model is also used in, e.g., [13,14]. Here, a two-state continuous time Markov model consisting of states $\Omega = \{Good, Bad\}$ is used, where ϵ_{Good} defines the error rate in the *Good* state. In the *Bad* state, the error rate ϵ_{Bad} is much higher than in the *Good* state. In Fig. 1, λ_{Good} and λ_{Bad} are the transition probabilities between the states. The state sojourn time can be

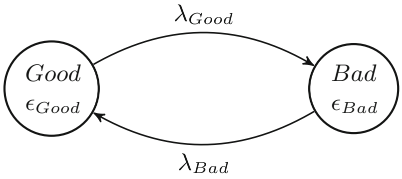


Fig. 1. The continuous time Gilbert-Elliot model

approximated by an exponential distribution. Thus, the sojourn time in the *Good* state is exponentially distributed with rate $\lambda_{Good} = 1/\mu_{Good}$, where μ_{Good} denotes the mean of the exponential distribution [15]. The sojourn time in the *Bad* state is exponentially distributed with rate $\lambda_{Bad} = 1/\mu_{Bad}$.

In Fig. 2, the received packets over a time of 10 s are shown in the emulator. The Gilbert-Elliot model is implemented with the parameters $\epsilon_{Bad} = 1$, $\epsilon_{Good} = 0$ and μ_{Bad} being at least three times the packet sending interval. This results in approximately three consecutive packet losses on average. μ_{Good} is chosen in such a proportion to μ_{Bad} so the overall error probability of the model equals the intended loss rate, e.g.: $P(Bad) = 0.1 \stackrel{!}{=} \frac{\mu_{Bad}}{\mu_{Good}+\mu_{Bad}}$.

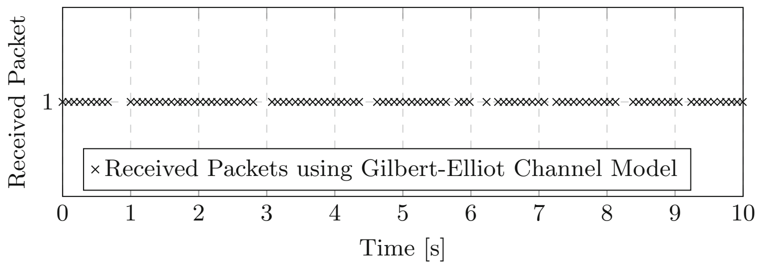


Fig. 2. Gilbert-Elliot channel with 10% loss, $\mu_{Bad} = 0.125$ s and $\mu_{Good} = 1.125$ s

3 Problem Description

A single link scenario will be compared to a two link channel bundling scenario with a lossy and a lossless link. While using uncorrelated error models, this should not make a difference. These errors can be compensated relatively well with FEC. In a correlated error model, the lossless link might help to reduce the losses in a row depending on the burst duration. This is due to still receiving parts of the data through the lossless link. Gains are expected although this is not a proper interleaving behavior. Losing only a few packets in a generation is important for coding, because as soon as more packets per generation are lost than redundancy is added, no decoding is possible and the entire generation is lost. Similar benefits should be possible in a channel bundling scenario with two independent correlated error channels, as the channels are lossy with a low probability at the same time.

Different FEC schemes are compared, namely RS codes and RLNC. It is expected that RS performs slightly better than RLNC in terms of goodput, because in RS the coding vector overhead is not needed.

3.1 Scenario

Presenting the channel bundling scenario with two links, the network can be represented as a directed graph with four nodes. The source, e.g. a ship or an aircraft, located at a vertex of the graph sends the information to a single receiver. The edges from the sender to the two different ground stations correspond to packet erasure channels, the different satellite links. The ground stations are connected to the receiver via wired links. The losses and delays on these links are neglected, because they are small compared to the ones of the satellite links. The throughput of the wired links is sufficient.

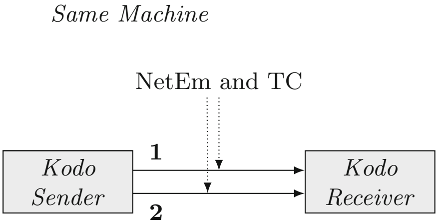
A satellite link is characterized by the following parameters. An available and guaranteed throughput of 96 kbit/s per link on layer 2 is assumed in the two link scenario. Using one link, a throughput of 192 kbit/s is assumed. Those values correspond to typical ones offered by the Inmarsat BGAN service [16]. A latency of 0.25 s is chosen for all modeled links corresponding to the propagation delay from the sender to a geostationary satellite to a ground station. Correlated losses are typical for satellite links [13]. Correlated losses happen in case the antenna of the sender is not oriented correctly, e.g. due to heavy waves in maritime communication. For a short time there is no connection possible until the antenna is again aligned. In Ku-Band high losses also occur during heavy rain.

The communication on the lossless link is done, e.g., in the L-Band, the lossy one in the Ku-Band. In general, Ku-Band communication is cheaper but less reliable than L-Band communication.

The goodput and packet loss ratio of the link, being important performance metrics, are evaluated.

3.2 Testbed Set-Up

The multipath scenario is emulated using one PC (see Fig. 3). This brings the advantage of easy and exact time measurements. Two virtual loopback interfaces (see Fig. 4) are created. This is done by assigning two IPv4 address to the local interface. The Linux based Network Emulator NetEm [17] and Traffic Control are used to modify the static loss rate and the delay of the virtual links. The functionalities of NetEm were studied in [18]. The Gilbert-Elliot channel is implemented by adjusting the packet loss through NetEm on state transition. The Steinwurf Kodo C++ library [19] is used for implementation of Reed-Solomon and Random Linear Network Coding. The coding in done above the transport layer. The coded packets are sent via UDP.



1 Satellite Link 1: Delay = 0.25 s, Throughput = 96 kbit/s, Loss = 0 %

2 Satellite Link 2: Delay = 0.25 s, Throughput = 96 kbit/s, Loss variable

Fig. 3. Testbed scenario with two links

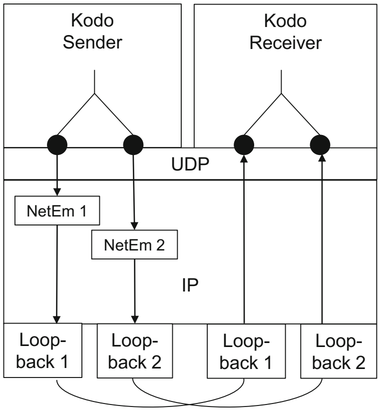


Fig. 4. Loopback interfaces OSI-Layer

In the following, the set-up for the emulator is described based on the criteria defined in Sect. 2.1. In the presented scenario two links with 96 kbit/s of throughput and a delay of 0.25 s are assumed. The links are used simultaneously with a 50%/50% scheduling. The total packet size is chosen to be 960 bytes in all tests, so the maximum possible throughput of the link is achieved at 40 ms fixed interarrival time. A data packet is sent every 40 ms on one of the two links. Thus, every 80 ms a packet is put on the lossy and every 80 ms a packet is put on the lossless link. This means that 25 packets per second are sent in the ideal case. For comparison, an identical scenario with regard to packet loss rate and total throughput with just one link is evaluated. Here, the throughput is set to 192 kbit/s and 960 bytes packets are sent every 40 ms. UDP, IP and MAC headers as well as network coding headers are subtracted from the total packet size to determine the payload size. For RLNC, the size of the network coding header changes with the field size and generation size.

For all tests a field size of $GF(q = 2^8)$ is chosen as in [9]. For RLNC, the packets should be linearly independent with a high probability. For RS, the number of different packets per generation is upper bounded by 255.

Depending on the test, the generation size is either fixed or varied. A specific amount of redundancy h is added. The redundancy must be added in the form of whole additional packets. Therefore, generation sizes of, e.g., $k = 2$ are not that useful. In case generation size $k = 2$ is chosen, 3 packets (50% redundancy) have to be sent also in a case where only 10% redundancy should be added. Therefore, comparison would be unfair and this is excluded. Very large generation sizes are not investigated. For example, a coding vector of 128 bytes has to be added using generation size $k = 128$ and field size $GF(256)$ in RLNC. This results in around 19% of total overhead for the 960 bytes long packets and is therefore not further analyzed. Decoding delay is not important in the scenario, as a file transfer scenario is emulated. That is why also larger generations of up to 80 are

Table 1. Content of a single network coded packet with a generation size of 16 packets and the field size $GF(q = 2^8)$ on layer 2

Size in bytes	Content
893	Payload
16	Network coding vector
1	Network coding meta data
2	Network coding sequence number
2	Network coding generation number
8	UDP header
20	IP header
18	MAC header
960	Total packet size

taken into account. Choosing large generation sizes and a low field size might be a considerable option for RLNC, but is not considered in this work.

It is expected that depending on the redundancy still high losses are possible. This is because of generations being lost entirely when they cannot be decoded. Therefore, even more data might be lost as without using any coding scheme. In this case, a backward error correction scheme might improve the performance.

For comparison, the parameters for RS are the same as for RLNC. This means, that the code rate is chosen at least according to the loss rate of the channel. The field size is set to $GF(q = 2^8)$, which is the default setting in [2] and is used, e.g., in [4]. A RLNC packet with generation size 16 has the structure displayed in Table 1. RS packets require no coding vector but instead an additional symbol index increasing their payload size to 908 bytes.

4 Results

At first, it will be investigated how much redundancy should be added to a packet erasure link given that the goodput should be maximized while the losses are preferably low. For the objective to nearly eliminate losses, the redundancy should be as high as possible. This contradicts with the aim to have an appropriate goodput. Therefore, a test with one packet erasure link with a throughput of 192 kbit/s is performed. The behavior of the system with correlated errors is compared to the system with uncorrelated errors in terms of measured average goodput and the loss rate. An uncorrelated loss rate of 5% is configured. For correlated errors, a mean sojourn time of 0.125 s in the *Bad* state and 2.375 s in the *Good* state of the Gilbert-Elliot model is chosen. The generation size is fixed to $k = 32$, as previous tests showed that this is a reasonable size. It is expected that RS will perform slightly better in terms of goodput due to its lower header overhead. Nevertheless, the loss rates should be the same.

The behavior of RLNC is displayed in Fig. 5 and the one of RS in Fig. 6. The average goodput is shown with a 95% confidence interval of multiple runs. As only full-packets can be added, sending an amount of 34 to 42 coded packets is observed, corresponding to around 6% to 31% redundancy. In each run data with a size of 91 MB is transmitted, so every run lasts around 10 min.

The theoretical maximum goodput for no losses is displayed as well. This threshold decreases as more redundancy is added and therefore the transmission takes longer. Assuming that k is the generation size, r is the redundancy to be added in percentage terms, g_p represents the goodput per packet in bit, and the number of send packets per second is p_s, then the maximum goodput g_s in bit/s is calculated as:

$$g_s = \frac{k}{(k + \lceil r \cdot k \rceil)} \cdot g_p \cdot p_s \tag{1}$$

Observing uncorrelated losses, an amount of extra four to five packets, thus a redundancy of 12.5% or 15.625%, should be chosen in terms of the observed goodput. Choosing a redundancy of 18.75% nearly no losses can be observed. This is noticeable, since the loss on the link is 5%. Therefore, a lot more redundancy must be added than there are losses. This shows a major disadvantage of coding with pure FEC: In case more losses occur than redundancy was added, the entire generation is lost as it cannot be decoded.

Using a correlated burst error model, even with an added redundancy of over 30% still errors of around 5% occur. This is not satisfactory. In terms of goodput, an added redundancy of 18.75% gives the best results. All in all, correlated losses affect the goodput far more than uncorrelated ones. Observing uncorrelated losses, nearly no losses can be achieved with less redundancy.

Comparing RLNC to RS, using RS results in a higher goodput in both cases of correlated and uncorrelated losses. The losses show nearly identical behaviors. To achieve almost zero losses using RS and an uncorrelated error model, a redundancy of 18.75% should be chosen as well.

For the next tests, a fixed redundancy is chosen and the generations sizes are varied. The channel bundling scenario with two heterogeneous links (see Fig. 3) will be compared to the scenario with one link, whereas the single link has 5% loss and the lossy channel in the two link scenario has a loss of 10%. The tests are restricted to RLNC, as it is assumed that RS has a similar behavior with the difference that RS has more goodput available due to the lower coding header size. The redundancy is set to 18.75% in all tests. This is not feasible in the correlated case for a real scenario, since then high losses are accepted. However, a comparison of the goodput between correlated and uncorrelated errors is possible this way.

The usage of two heterogeneous links should perform equal or better compared to one link in a scenario with correlated errors. In a scenario with two links with equal end-to-end delay and 50%/50% scheduling, packets are received alternating from one link and the other one. The observed phenomenon of losing an entire generation is reduced, because packets still arrive on the lossless link. This is especially useful in case the correlated errors occur close to the end of

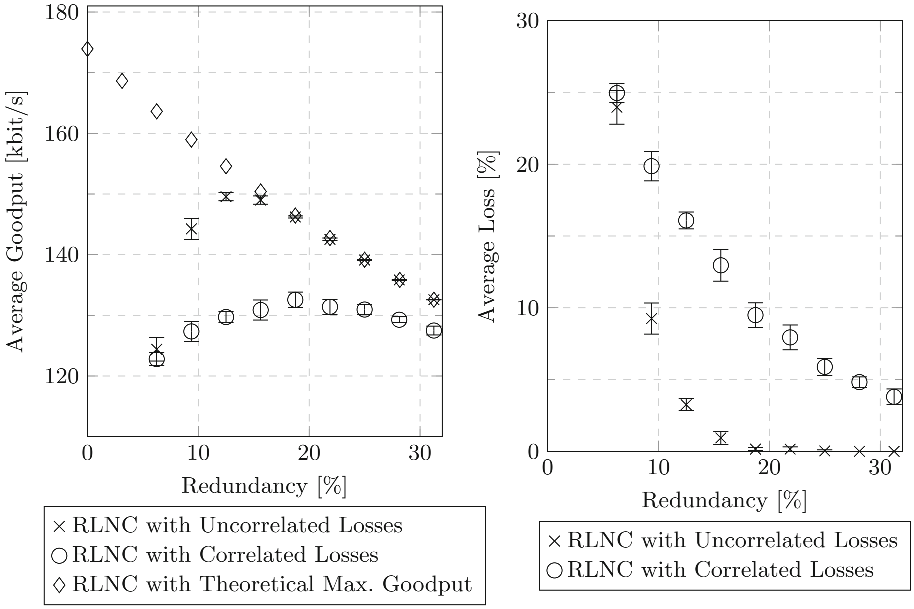


Fig. 5. Results for RLNC using one link with 192 kbit/s and 5% loss

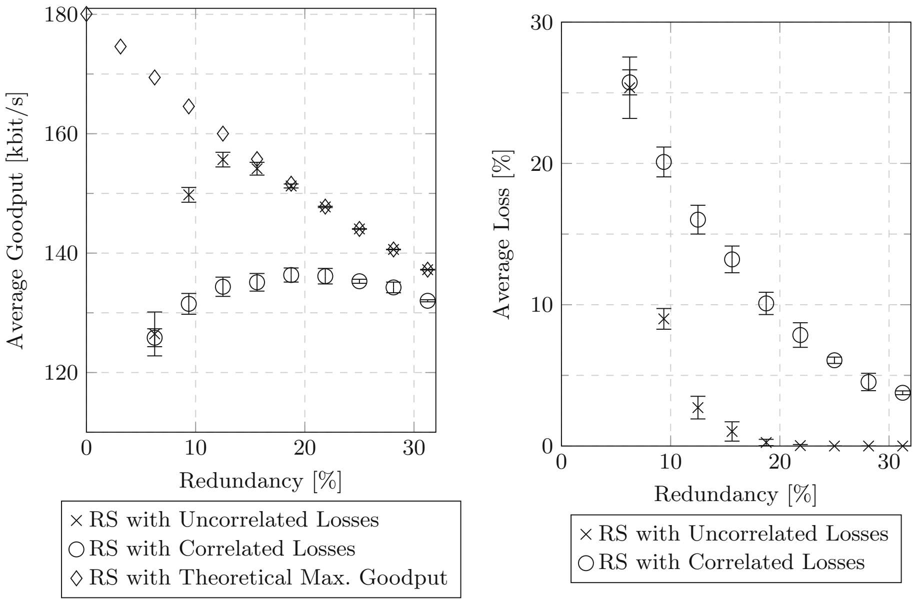


Fig. 6. Results for RS using one link with 192 kbit/s and 5% loss

a generation. Imagine a generation on the single link is lost. In the two link scenario however, the generation is not lost necessarily with conditions being equal. It is not lost in case two generations are affected by the errors and the first generation loses data at the end, the second one at the beginning. Both receive enough packets to be decoded. As an example, it is assumed that eight packets and two redundant packets are transmitted. In case three packets are lost, also the remaining seven packets are useless. If the seventh to ninth packet out of ten are lost on a single link due to correlated errors, the generation is useless. The following generation is not affected. Having two links in the same constellation, due to the scheduling the seventh and ninth packet of the first generation and the first packet of following generation are lost by a burst duration of three packets. Therefore, both generations can be decoded. This advantage is linked to the ratio of burst duration to the number of send packets or rather the generation size.

The impacts on losses and goodput is investigated for different generation sizes and a fixed burst duration of three packets on average. A performance improvement in terms of goodput should be visible, as less generations are expected to be lost. To achieve confident results, the transmitted data is increased and more runs are considered. For correlated errors of two links, a mean sojourn time of 0.250 s in the *Bad* state and 2.25 s in the *Good* state in

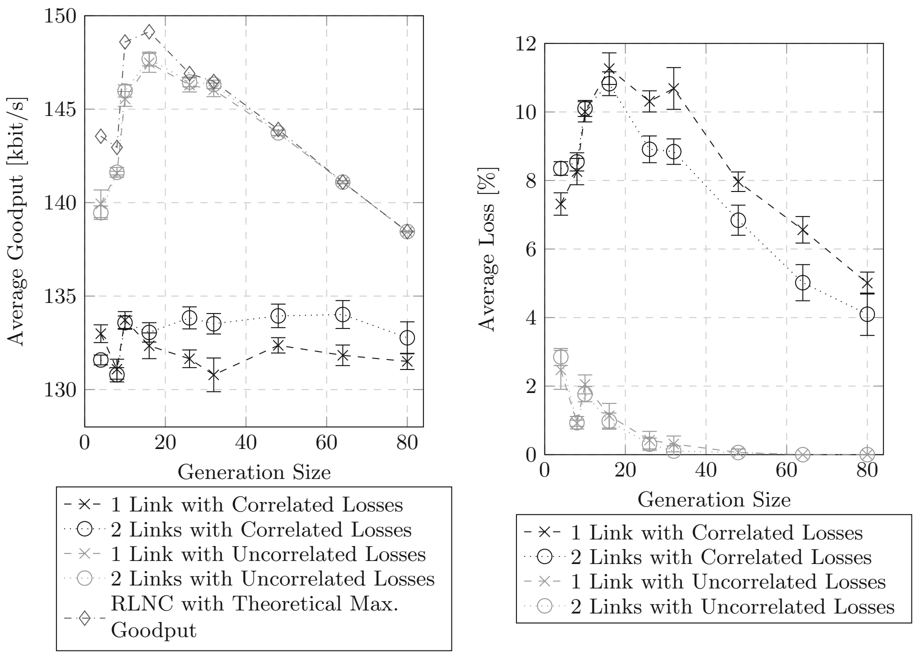


Fig. 7. Comparison of 1 and 2 links for different generation sizes using RLNC and 18.75% redundancy. 5% loss on 1 link equals 10% loss/0% loss on 2 links.

the Gilbert-Elliot model is chosen, which corresponds to three lost packets on average and a 10% loss rate. In Fig. 7 the goodput using at least 18.75% redundancy is displayed for different generation sizes. For generation sizes $k = 4$ and $k = 8$, 25% redundancy is assumed as only whole packets can be added. Nevertheless, it is possible to compare the one link scenario to the two link scenario.

As a conclusion, it is estimated that using two links is beneficial for higher generation sizes. Here, the number of sent packets is large compared to the burst duration. For generation size $k = 4$ using one link performs better. This is related to the fact that the average burst duration nearly equals the generation size. For uncorrelated losses it makes no difference whether one link or two links are considered, as expected.

5 Conclusion and Outlook

The tests showed that the amount of redundancy determines the performance of FEC. Choosing the redundancy too low or too high results in low average goodput. For redundancies of at least 18.75% on a channel with 5% uncorrelated errors, packet losses can be nearly eliminated. Pure FEC coding in the presented way suffers from losing entire generations. Therefore, the coding parameters have to be chosen carefully. RLNC and RS show a similar behavior of losses when varying the redundancy. The RS payload is larger due to lower header sizes, which leads to a better performance in terms of goodput in the given scenario.

A channel bundling scenario with two links was compared to a single link with the determined redundancy for different generation sizes and a fixed average burst duration. Here, a benefit while using two links was observed for generation sizes much larger than the burst duration. An analytic model is to be developed to predict the losses and goodput for different redundancies, generation sizes and burst durations. The model is to be extended to adaptively react to link changes. It is planned to enhance the scenario with, e.g., highly heterogeneous links with different delays and bandwidths as well as additional links. Furthermore, instead of emulating a file download the data should be coded on-the-fly.

Acknowledgment. This work was partially funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) as part of the *RekoTrans* project with reference number 20Y1504B.

References

1. Rizzo, L.: Effective Erasure Codes for Reliable Computer Communication Protocols. SIGCOMM Comput. Commun. Rev. **27**(2), 24–36 (1997)
2. Lacan, J., Roca, V., Peltotalo, J., Peltotalo, S.: Reed-Solomon Forward Error Correction (FEC) Schemes, April 2009. https://tools.ietf.org/html/rfc5510
3. Lamoriniere, C., Nafaa, A., Murphy, L.: Dynamic switching between adaptive FEC protocols for reliable multi-source streaming. In: Global Telecommunications Conference, GLOBECOM 2009, pp. 1–6. IEEE, November 2009

4. Matsuzono, K., Detchart, J., Cunche, M., Roca, V., Asaeda, H.: Performance analysis of a high-performance real-time application with several AL-FEC schemes. In: 2010 IEEE 35th Conference on Local Computer Networks (LCN), pp. 1–7, October 2010
5. Ho, T., Medard, M., Shi, J., Effros, M., Karger, D.R.: On randomized network coding. In: Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing (2003)
6. Heide, J., Pedersen, M.V., Fitzek, F.H.P., Medard, M.: On code parameters and coding vector representation for practical RLNC. In 2011 IEEE International Conference on Communications (ICC), pp. 1–5, June 2011
7. Langberg, M., Sprintson, A.: Recent results on the algorithmic complexity of network coding. In: Tutorial appearing in Proceedings of Workshop on Network Coding, Theory, and Applications (NetCod) (2009)
8. Pedersen, M.V., Lucani, D.E., Fitzek, F.H.P., Sorensen, C.W., Badr, A.S.: Network coding designs suited for the real world: what works, what doesn't, what's promising. In: 2013 IEEE Information Theory Workshop (ITW), pp. 1–5. IEEE, September 2013
9. Moreira, A., Lucani, D.E.: Coded schemes for asymmetric wireless interfaces: theory and practice. IEEE J. Sel. Areas Commun. **33**(2), 171–184 (2015)
10. Garrammone, G.: On decoding complexity of Reed-Solomon codes on the packet erasure channel. IEEE Commun. Lett. **17**(4), 773–776 (2013)
11. Celandroni, N., Gotta, A.: Performance analysis of systematic upper layer FEC codes and interleaving in land mobile satellite channels. IEEE Trans. Veh. Technol. **60**(4), 1887–1894 (2011)
12. Lutz, E., Cygan, D., Dippold, M., Dolainsky, F., Papke, W.: The land mobile satellite communication channel-recording, statistics, and channel model. IEEE Trans. Veh. Technol. **40**(2), 375–386 (1991)
13. Shang, Y., Hadjitheodosiou, M.: TCP splitting protocol for broadband aeronautical satellite network. In: The 23rd Digital Avionics Systems Conference, DASC 2004, vol. 2, pp. 11.C.3–11.1-9, October 2004
14. Casone, L., Ciccarese, G., De Blasi, M., Patrono, L., Tomasicchio, G.: An efficient ARQ protocol for a mobile geo-stationary satellite channel. In: Global Telecommunications Conference, GLOBECOM 2001, vol. 4, pp. 2692–2697. IEEE (2001)
15. Zhang, Y., Hu, H., Fujise, M.: Resource, Mobility, and Security Management in Wireless Networks and Mobile Communications. Wireless Networks and Mobile Communications. CRC Press, Boca Raton (2006)
16. Colledge, G., Hiesler, R., Febvre, P., Platt, P., Sharma, A., Maiolla, V., Schuster Bruce, A., Huggins, G.: SwiftBroadband capabilities to support aeronautical safety services WP1: technical description and application to ATS, June 2006
17. Linux Foundation. NetEm - Network Emulator (2016). http://www.linuxfoundation.org/collaborate/workgroups/networking/netem. Accessed 10 May 2016
18. Jurgelionis, A., Laulajainen, J.P., Hirvonen, M., Wang, A.I.: An empirical study of NetEm network emulation functionalities. In: 2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pp. 1–6, July 2011
19. Pedersen, M.V., Heide, J., Fitzek, F.H.P.: Kodo: an open and research oriented network coding library. In: Casares-Giner, V., Manzoni, P., Pont, A. (eds.) NETWORKING 2011. LNCS, vol. 6827, pp. 145–152. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23041-7_15