Binary Codes for Packet Error and Packet Loss Correction in Store and Forward

Maximilien Gadouleau and Alban Goupil
CReSTIC, Université de Reims Champagne-Ardenne
Reims, France
E-mails: {maximilien.gadouleau, alban.goupil}@univ-reims.fr

***Abstract*—In this paper, we introduce a novel approach to correct packet errors and packet losses in store and forward by using binary error-correcting codes. Using a framework similar to what has been proposed for error control in random linear network coding, we investigate error control under two scenarios. First, we show that the Hamming metric is suitable for error control in the case of errors intrinsic to the network. Second, we investigate the case of an adversary on the network who erases and injects packets in order to corrupt the communication. Under this setting, we show that error correction is performed using a new metric, referred to as the modified Hamming metric. We then investigate using constant-weight codes and linear codes for error correction in store and forward. We thus show that the traditional approach of indexing the packets in order to recover their original order is a suboptimal restriction of our approach.**

I. Introduction

Store and forward is the traditional means to transmit data through a network. In this scheme, the intermediate nodes simply retransmit the packets they receive towards their destinations without combining them. The traditional techniques to correct packet loss in store and forward include ARQ and erasure codes based on Reed-Solomon proposed in [1], [2], [3], [4]. The approach based on Reed-Solomon codes, referred to as the traditional approach henceforth, distributes a codeword amongst the packets, hence protecting against packet loss. In order to recover the order of packets, a header indicating the index of the packet in the message is added in front of every packet.

In this paper, we propose a novel approach to correct errors and packet losses in store and forward by using binary error-correcting codes. We assume that the collection of packets, referred to as the message, undergoes four possible modifications. We first assume that the packets are randomly permuted, and hence do not arrive at the receiver's end in the order they were originally sent. This assumption is motivated by networks whose topologies change over time, or where several routes with unequal delays are available to transmit the data. Also, the message can suffer from three types of packet modifications: some packets can be lost due to fading or if some links fail, some packets can be injected by an adversary, and others can be corrupted by errors. We hence model the communication of a message using store and forward as the transmission of the set of packets, where the modifications of packets correspond to set alterations. Using a correspondence between subsets of a set and binary vectors, we then model communications using store and forward as a binary channel with additive error. Thus, error control for store and forward can be performed using binary error-correcting codes. Although the Hamming metric is appropriate in the case of modifications due to the network, we also introduce an adversarial scenario where error correction is done via a new metric, referred to as the modified Hamming metric.

We then investigate using constant-weight codes, which is equivalent to transmitting messages with the same number of packets. Constant-weight codes have been widely studied due to their numerous applications and their theoretical significance (see [5] and [6] for comprehensive surveys). We show that the traditional approach of packet loss protection based on Reed-Solomon codes can be viewed in our model as using a subclass of constant-weight codes, referred to as liftings of Hamming metric codes. Therefore, the traditional approach is a suboptimal restriction of our approach to the set of all liftings.

Since the length of the binary codes proposed here increases exponentially with the number of symbols in a packet, encoding and decoding complexities could become an issue. In order to tackle this, we finally investigate using linear codes for error control in store and forward, as they have a low encoding complexity and many classes of linear codes also have a low decoding complexity. However, other practical issues arise when using these codes. For instance, transmitting the all-zero vector, which is a codeword in any linear code, would result in sending an empty message. Other issues are also investigated, such as the standard deviation and the maximum value of the number of packets in a message.

Our approach offers many advantages, listed below. First, it is based on a simple network protocol, store and forward, where the intermediate nodes do not operate on packets. Our approach is well suited for store and forward, as the error control operations are done at the receiver's end only. Second, unlike ARQ schemes, our approach does not require any feedback, and unlike fountain codes, we only transmit the message once without combining packets. Third, our codes have a higher rate than the erasure codes previously proposed. Although the rate gain increases with the number of packets, it is always positive even for a small number of packets, and increases rapidly for a number of packets that is small compared to the total number of possible packets. Fourth,

ISBN 978-3-8007-3211-1

unlike traditional methods which only protect against packet losses, our approach also corrects packet injections and packet errors. Hence the packets need not be coded against link errors, and the data rate can be further increased. Fifth, our scheme is universal, as it is always based on binary codes, regardless of the alphabet or the length of the packets.

We remark that the framework introduced here is similar to the one introduced in [7], [8] for error correction in random linear network coding. However, the model for network coding is defined only in terms of subspaces, while our model is defined in terms of both subsets and binary vectors. This allows us to take advantage of the structure of binary vector spaces and of the wealth of results on binary error-correcting codes. Also, the modified Hamming metric defined here has similar properties to the injection metric defined in [8] for network coding and both arise from adversarial scenarios. However, the assumptions on the adversary in these two scenarios are completely different.

The rest of the paper is organized as follows. Section II introduces the operator channel model for store and forward, the metrics used for error correction, and demonstrates how binary codes can be used for that purpose. Section III then investigates using constant-weight codes and compares it to the traditional approach. Section IV discusses the implementation issues of using linear codes. Finally, Section V sumarizes our results and provides a list of possible extensions of our work.

Due to length restrictions, we have omitted all the proofs in this paper.

II. Channel model and metrics

A. Channel model

For any positive integer q, we denote $[q] = \{0, 1, \ldots, q-1\}$ henceforth. Note that we do not assume any underlying structure for $[q]$. Suppose a source wants to transmit a collection of packets in $[q]^n$, referred to as a *message*, to one or several receivers across a network. The protocol used is store and forward, where the intermediate nodes do not operate on the packets they receive. We first assume that all network links and nodes are error-free and that no packets are lost or injected, hence all the packets sent by the source will be correctly received by all the receivers. However, we also assume that the packets may not be received in the order they were originally sent. We remark that the information carried by the message can only be a property of this message which remains invariant after the modifications operated by the channel. Since the order of packets in the original message is lost through the communication, store and forward can thus be viewed as the error-free transmission of only the *set* of packets, rather than the ordered sequence of packets.

Let us now consider the case where the network is not error-free, hence the original message may be modified due to link errors, packet losses, malfunctioning nodes, an adversary injecting packets maliciously, etc. All these modifications can be viewed as alterations of the set of packets $P \in \mathcal{P}([N])$ sent by the source, where $N = q^n$ represents the total number of possible packets and $\mathcal{P}(E)$ represents the power set of a set E. These alterations can always be expressed as products of erasures and disclosures, where an *erasure* is the deletion of an element in the subset, and a *disclosure* is the addition of a new element into the subset[1].

We now illustrate how message modifications lead to alterations of the corresponding set of packets. We consider three basic alterations of the message. First, a packet loss leads to either an erasure if no copy of the packet is being transmitted on the network, or no alteration otherwise. Second, the injection of a new packet leads to either a disclosure if the newly added packet is distinct to the other packets in the message, or no alteration otherwise. Third, a packet in error can be viewed as a packet loss immediately followed by a packet injection. Therefore, a packet error can lead to an erasure, to a disclosure, or to both an erasure and a disclosure.

After ϵ erasures and δ disclosures, the received subset Q can be expressed as

$$Q = (P \backslash E) \cup D, \tag{1}$$

where $E = Q \cap P$ and $D = Q \backslash P$ have cardinalities ϵ and δ, respectively. Note that E and D thus represent the packets lost in the channel, and the error packets injected by the channel, respectively. Equivalently, denoting $A = P \Delta Q = E \cup D$, where Δ denotes the symmetric difference between two sets, (1) can be expressed as

$$Q = P \Delta A, \tag{2}$$

where $|A| = \epsilon + \delta$ is the cardinality of A. We thus model the communication of a message using store and forward as an *operator channel*, where the input P is a subset of $[N]$ and the output Q is a random subset of $[N]$ related to P by (2), where ϵ and δ are upper bounded. We remark that the counterpart of (1) for subspaces was given in [7, (2)], while (2) has no counterpart for subspaces. This is due to the underlying group structure of the power set which has no counterpart in the projective space.

B. Metrics for correction of erasures and disclosures

The Hamming distance between two subsets $P, Q \in \mathcal{P}([N])$, where Q is obtained from P after ϵ erasures and δ disclosures, is defined as $d_{\mathrm{H}}(P, Q) = \epsilon + \delta = |P \Delta Q|$. The Hamming distance clearly is a metric, and is the appropriate metric for error correction when the sum of the number of erasures and disclosures is upper bounded.

We now introduce a scenario where a different metric than the Hamming metric is more appropriate for error correction in store and forward. Suppose a source transmits packets to a destination across a network, where an adversary wishes to corrupt the communication is present. The adversary is capable

[1]For random linear network coding, Kötter and Kschischang introduced the terms "erasure" and "error," respectively, in order to describe the counterparts of the alterations to subspaces [7]. However, the terms "erasure" and "disclosure" seem more appropriate, as a disclosure represents the gain of some amount of information. Also, the terms of "errors and erasures" are commonly referred to describe significantly different alterations to vectors or matrices. Therefore, in order to avoid confusion, we shall use the terms "erasure" and "disclosure" henceforth.

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach

of removing some packets transiting in the network and injecting new packets into the network, however it cannot modify the packets going through the network. In order to search for malicious adversaries, a network monitor counts the injections and the erasures of packets separately. If either counter crosses a given upper bound, the communication will be invalidated. Therefore, the adversary will make sure that both the number of packets erased and lost are below the alarm bound. The modified Hamming distance between P and Q is thus defined as $d_{\mathrm{M}}(P,Q) = \max\{\epsilon,\delta\} = \max\{|P\backslash Q|, |Q\backslash P|\}$.

Lemma 1: The modified Hamming distance is a metric.

We have for all $P, Q \in \mathcal{P}([N])$,

$$d_{\mathrm{M}}(P,Q) \leq d_{\mathrm{H}}(P,Q) \leq 2d_{\mathrm{M}}(P,Q), \tag{3}$$

which implies that these metrics are equivalent. By (3), the problems of finding optimal codes in the Hamming metric and in the modified Hamming metric are closely related, but not necessarily equivalent. We remark that both metrics introduced above are symmetrical in terms of erasures and disclosures. However, there may be other scenarios for which asymmetrical metrics are more appropriate for error control.

C. Encoding and decoding using binary codes

We now demonstrate how the channel model and the metrics introduced for subsets in Sections II-A and II-B, respectively, can be equivalently defined in terms of binary vectors. Any subset $P \in \mathcal{P}([N])$ is uniquely represented by a binary vector $\mathbf{p} = (p_0, p_1, \ldots, p_{N-1}) \in \mathrm{GF}(2)^N$, where $p_i = 1$ if and only if $i \in P$, or equivalently, $\mathrm{supp}(\mathbf{p}) = P$. Therefore, the operator channel can be defined as a channel on $\mathrm{GF}(2)^N$, where the input $\mathbf{p}$ and the output $\mathbf{q}$ are related by

$$\mathbf{q} = \mathbf{p} + \mathbf{a}, \tag{4}$$

where $|\mathrm{supp}(\mathbf{a}) \cap \mathrm{supp}(\mathbf{p})| = \epsilon$ and $|\mathrm{supp}(\mathbf{a})\backslash\mathrm{supp}(\mathbf{p})| = \delta$. Note that $|\mathrm{supp}(\mathbf{a})| = w_{\mathrm{H}}(\mathbf{a}) = \epsilon+\delta$. Moreover, for all $\mathbf{p}, \mathbf{q} \in \mathrm{GF}(2)^N$, we define the modified Hamming distance between $\mathbf{p}$ and $\mathbf{q}$ as $d_{\mathrm{M}}(\mathbf{p},\mathbf{q}) = \frac{1}{2}d_{\mathrm{H}}(\mathbf{p},\mathbf{q}) + \frac{1}{2}|w_{\mathrm{H}}(\mathbf{p}) - w_{\mathrm{H}}(\mathbf{q})|$. We thus have

$$d_{\mathrm{H}}(\mathbf{p},\mathbf{q}) = d_{\mathrm{H}}(\mathrm{supp}(\mathbf{p}), \mathrm{supp}(\mathbf{q})), \tag{5}$$

$$d_{\mathrm{M}}(\mathbf{p},\mathbf{q}) = d_{\mathrm{M}}(\mathrm{supp}(\mathbf{p}), \mathrm{supp}(\mathbf{q})) \tag{6}$$

for all $\mathbf{p}, \mathbf{q} \in \mathrm{GF}(2)^N$. Therefore, error control in store and forward can be treated a coding theory problem on binary vectors with the Hamming metric or the modified Hamming metric, depending on the considered scenario. We describe below the encoding and decoding processes for the transmission of a message using a binary code.

The encoding of a message when using a binary error correcting code $\mathcal{C} \subseteq \mathrm{GF}(2)^N$ with cardinality M and minimum Hamming distance d proceeds as follows. Suppose the source wishes to send a word $\mathbf{x} \in [M]$. It first uses the encoding mapping f from $[M]$ to $\mathcal{C}$ to determine $f(\mathbf{x}) = \mathbf{c} \in \mathcal{C}$. Let $i_0 < i_1 < \ldots < i_{L-1}$ be the nonzero coordinates of $\mathbf{c}$, where $L = w_{\mathrm{H}}(\mathbf{c})$. Then the message will consist of exactly L packets $\mathbf{x}_j \in [q]^n$, where $\mathbf{x}_j$ is the representation of the number i_j in basis q. We remark that this encoding process guarantees that the packets are in lexicographic order.

Suppose the channel has applied ϵ erasures and δ disclosures to the original set of packets, where $\epsilon + \delta < \frac{d}{2}$. The decoding thus proceeds. The receiver obtains a different set of L' packets, which after sorting into lexicographic order, are denoted as $\mathbf{y}_l \in [q]^n$, $0 \leq l \leq L'-1$. Viewing these words as integers $k_l \in [N]$, the receiver then determines $\mathbf{y} \in \mathrm{GF}(2)^N$ where $\mathrm{supp}(\mathbf{y}) = \{k_l : 0 \leq l \leq L'-1\}$. Finally, the receiver applies the decoding algorithm for the code $\mathcal{C}$ on $\mathbf{y}$ to determine $\mathbf{c} \in \mathcal{C}$, and retrieves $\mathbf{x} = f^{-1}(\mathbf{c})$.

In order to illustrate the encoding and decoding of packets using binary codes, we consider the following example: $q = 2$, $n = 3$, and $\mathcal{C}$ is the $(8,4,4)$ extended Hamming code. Suppose $\mathbf{x} = (1000) \in \mathrm{GF}(2)^4$, then its corresponding codeword is $\mathbf{c} = (100001101)$ and hence the message sent by the source consists of the following 4 packets: $\mathbf{x}_0 = (000)$, $\mathbf{x}_1 = (100)$, $\mathbf{x}_2 = (101)$, $\mathbf{x}_3 = (111)$. If the packet $\mathbf{x}_0$ is lost during transmission, then the receiver obtains $\mathbf{y} = (00001101)$. Using the decoding algorithm for $\mathcal{C}$, the receiver then determines the orignial word $\mathbf{x} = (1000)$.

III. Constant-weight codes and liftings

A. Constant-weight codes

In order to simplify the transmission protocol and to facilitate the decoding process at the receiver end, the source may choose to send messages with a constant number of packets. In the model introduced in Section II-C, this corresponds to using a binary constant-weight code [5]. Constant-weight codes have other advantages listed below.

First, the set of binary vectors with the same Hamming weight is highly structured: when endowed with the Hamming metric, it forms an association scheme, referred to as the Johnson scheme [9], [10]. Constant-weight codes also have many connections to other classes of codes, such as spherical codes [6], binary codes [11], and constant-dimension codes [12]. Because of these features, constant-weight codes have attracted a lot of interest in the literature.

Second, by (3), (5), and (6), we have $d_{\mathrm{H}}(\mathbf{p},\mathbf{q}) \leq 2d_{\mathrm{M}}(\mathbf{p},\mathbf{q})$ for all $\mathbf{p}, \mathbf{q} \in \mathrm{GF}(2)^N$, and equality holds if and only if they have equal Hamming weights. Hence the minimum Hamming distance of a constant-weight code is equal to twice its minimum modified Hamming distance, which is the largest minimum Hamming distance once the minimum modified Hamming distance is fixed. Constant-weight codes can thus be used interchangeably for both scenarios previously considered. The property above also implies that the combinatorial and geometric properties of the Johnson scheme when endowed with the Hamming metric are preserved when it is endowed with the modified Hamming metric instead.

Third, let us denote the maximum cardinality of a binary code of length N with minimum Hamming distance d as $A_{\mathrm{H}}(N,d)$, and the maximum cardinality of a constant-weight code of length N and weight L with minimum Hamming distance d as $A_{\mathrm{H}}(N,L,d)$. Clearly, $A_{\mathrm{H}}(N,d) \geq A_{\mathrm{H}}(N,L,d)$

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach

for all $0 \leq L \leq N$. Also, the Bassalygo-Elias bound [13], [11, Ch. 17, Theorem 33] indicates that

$$A_{\mathrm{H}}(N, L, d) \geq \frac{\binom{N}{L}}{2^N} A_{\mathrm{H}}(N, d) \tag{7}$$

for all $0 \leq L \leq N$. Following Stirling's formula, the binomial coefficient $\binom{N}{L}$ satisfies for all N and $0 \leq L \leq N$ [11, Chapter 10, Lemma 7],

$$\frac{2^{NH(\lambda)}}{\sqrt{8L(1-\lambda)}} \leq \binom{N}{L} \leq \frac{2^{NH(\lambda)}}{\sqrt{2\pi L(1-\lambda)}}, \tag{8}$$

where $\lambda = \frac{L}{N}$ and $H(\lambda)$ is the binary entropy function. In particular, for N even and $L = \frac{N}{2}$, (8) and (7) lead to

$$\frac{A_{\mathrm{H}}(N, d)}{\sqrt{2N}} \leq A_{\mathrm{H}}\left(N, \frac{N}{2}, d\right) \leq A_{\mathrm{H}}(N, d), \tag{9}$$

and hence constant-weight codes with weight around half their length form asymptotically optimal binary codes in the Hamming metric. We now derive a similar result for the modified Hamming metric.

Proposition 1: The maximum cardinality $A_{\mathrm{M}}(N, d)$ of a binary code of length N with minimum modified Hamming distance d and the maximum cardinality $A_{\mathrm{M}}(N, L, d)$ of a constant-weight code of length N and weight L with minimum modified Hamming distance d satisfy

$$\frac{A_{\mathrm{M}}(N, d)}{(N+1)\sqrt{2N}} \leq A_{\mathrm{M}}\left(N, \frac{N}{2}, d\right) \leq A_{\mathrm{M}}(N, d). \tag{10}$$

Thus constant-weight codes form asymptotically optimal codes for both metrics. As a corollary, there exist codes which are nearly optimal for both metrics. However, codes that are simultaneously optimal for both metrics do not necessarily exist.

B. Liftings of Hamming metric codes

Fourth, constant-weight codes are related to Hamming metric codes over larger alphabets through the *lifting* operation, described below.

Definition 1: Let $L, M \geq 1$ and $\mathbf{X} = (X_0, X_1, \ldots, X_{L-1}) \in [M]^L$. Representing each coordinate X_i into a binary vector $\mathbf{x}_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,M-1})$ where $\mathrm{supp}(\mathbf{x}_i) = X_i$, the lifting of $\mathbf{X}$, denoted as $I(\mathbf{X})$, is the vector in $\mathrm{GF}(2)^{LM}$ obtained by concatenating all the L vectors $\mathbf{x}_i$.

Note that since the vector $\mathbf{x}_i$ has Hamming weight 1 for all $0 \leq i \leq L-1$, $I(\mathbf{X})$ has Hamming weight L. We remark that although the original word $\mathbf{X}$ can have coordinates over any alphabet, its lifting $I(\mathbf{X})$ is always a binary vector. The distance between two liftings is related to the distance between the original words below.

Lemma 2: For all $\mathbf{X}, \mathbf{Y} \in [M]^L$, $d_{\mathrm{H}}(I(\mathbf{X}), I(\mathbf{Y})) = 2d_{\mathrm{M}}(I(\mathbf{X}), I(\mathbf{Y})) = 2d_{\mathrm{H}}(\mathbf{X}, \mathbf{Y})$.

The definition of lifting is naturally extended to codes in $[M]^L$ as follows: $I(\mathcal{C}) = \{I(\mathbf{X}) : \mathbf{X} \in \mathcal{C}\}$. Lemma 2 establishes a strong relation between an M-ary Hamming metric code and its lifting, which is a constant-weight code. This relation is summarized in Proposition 2 below.

Proposition 2: Let $\mathcal{C}$ be a code in $[M]^L$ with minimum Hamming distance d, then its lifting $I(\mathcal{C})$ is a constant-weight code in $\mathrm{GF}(2)^{ML}$ with weight L and minimum Hamming distance $2d$.

We now compare our subset approach to the traditional way of transmitting data across a channel which permutes the packets. Let the source send a message of L packets in $[q]^n$, which is viewed as a binary vector $\mathbf{p} \in \mathrm{GF}(2)^N$ with Hamming weight L. By definition, our approach chooses amongst all $\binom{N}{L}$ such vectors. However, in the traditional approach, the source adds a header to the packets indicating their index in the original order for the receivers to determine the original order of packets. Proposition 3 indicates that such a message corresponds to the lifting of some word.

Proposition 3: A message of L packets in $[q]^n$ transmitted using the traditional method corresponds to the lifting of a word in $[q^{n-\lceil l \rceil}]^L$, where $l = \log_q L$.

Conversely, it is easily shown that any lifting corresponds to a message where the header gives the index of the packet. Therefore, the traditional index method restricts itself to the set of all liftings. Since there are L packets in every message, the header takes $\lceil l \rceil$ symbols, leading to a total overhead of $L\lceil l \rceil$ symbols. Therefore, only $q^{L(n-\lceil l \rceil)} \leq \left(\frac{N}{L}\right)^L < \binom{N}{L}$ vectors can be obtained through the traditional index method. Thus, the traditional approach only considers a proper subset of all vectors with weight L and is hence suboptimal.

We want to emphasize the gain in terms of data rate of our binary approach over the traditional approach by comparing their respective asymptotic rates. We remark that the asymptotic rates of constant-weight codes and Hamming metric codes are still unknown for all minimum distances [11]. Therefore, the gain cannot be derived for all possible minimum distances. Instead, we only consider the alphabets on which these codes are based, which is equivalent to setting the minimum distance to 1. Although this is only a special case, we believe it provides an interesting insight on the rate improvement obtained by using the binary vector approach over the traditional approach. Thus consider the combinatorial rates $R_b(L) = \frac{\log_q \binom{N}{L}}{Ln}$, $R_t(L) = \frac{L(n-\lceil l \rceil)}{Ln}$ of the binary approach and of the traditional approach, respectively. We also introduce the gain

$$G(L) = \frac{R_b(L)}{R_t(L)} = \frac{\log_q \binom{N}{L}}{L(n - \lceil l \rceil)} \tag{11}$$

of the binary approach over the traditional approach. Proposition 4 determines the asymptotic form of the gain, denoted as $g(\lambda) = \lim_{N \to \infty} G(\lambda N)$, where $\lambda = \frac{L}{N}$.

Proposition 4: The asymptotic gain of the binary approach over the traditional approach, where all messages consist of L packets in $[q]^n$ is given by

$$g(\lambda) = -\frac{H(\lambda)}{\lambda \log_2 \lambda} = 1 + \frac{(1-\lambda)\log_2(1-\lambda)}{\lambda \log_2 \lambda}, \tag{12}$$

where $\lambda = \frac{L}{N}$ and $H(\lambda)$ is the binary entropy function.

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach

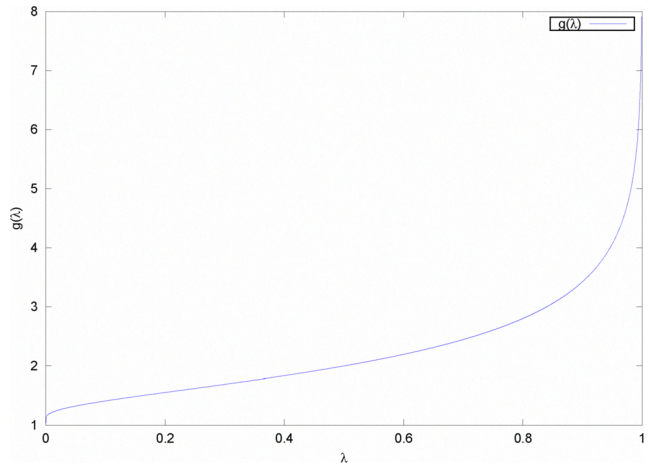

Fig. 1. Asymptotic gain of the binary approach over the traditional approach.

The gain $g(\lambda)$ is plotted in Figure 1 for λ ranging from 0 to 1. Recall that the parameter λ represents the ratio of the number of packets in a message over the number of possible packets in $[q]^n$. We remark that the range of λ can be split into two regions. The first region, where λ is low, represents the case where the messages consist of a small number of large packets. The other region, where λ is high, represents the case where many small packets are transmitted in each message. It is clear from Figure 1 that our approach, although it always improves upon the traditional approach, is more suitable for the second region. When λ approaches 1, the index in the traditional approach covers nearly the whole packet, and hence the rate $R_t(L)$ tends to 0 and the gain of the binary approach tends to infinity.

The binary approach is more efficient for large values of λ; however, small values of λ more typically occur in applications. Nonetheless, we remark that because $g(\lambda)$ has an infinite derivative at 0, the gain increases rapidly for small values of λ, hence our approach offers non-negligible gain, even for very small values of λ. In order to illustrate this gain for small values of λ, suppose the source wishes to send $L = 64$ packets of 16 bytes, i.e. $N = 2^{128}$ and $\lambda = 2^{-122}$. Then the total number of bits sent is $L \log_2 N = 8192$, and the number of useful bits sent using the the traditional approach is given by $L(\log_2 N - \log_2 L) = 7808$, while the number of useful bits sent using the subset approach is 7892. Therefore, our approach saves 84 bits, hence nearly a whole packet.

IV. Linear codes

As shown in Section II-C, any binary code can be used for error control in store and forward. In Section III, we described the advantages and issues of using constant-weight codes. In this section, we focus on linear codes instead. The main advantages of linear codes are the ease of encoding and the fact that many classes of linear codes have efficient decoding algorithms as well. This is particularly desirable, as the code length increases exponentially with the length n of the packets. However, a practical issue arises when using linear codes, as described below.

Suppose the source wants to transmit a vector $\mathbf{x} \in \mathrm{GF}(2)^K$ using an (N, K) linear code $\mathcal{C}$ with minimum Hamming distance d. The encoding function of $\mathcal{C}$ is a matrix multiplication: $\mathbf{c} = \mathbf{x}\mathbf{G}$, where $\mathbf{G}$ is the generator matrix of the code. According to Section II-C, the corresponding message consists of $w_{\mathrm{H}}(\mathbf{c})$ packets. In particular, if $\mathbf{x} = \mathbf{0}$, then $\mathbf{c} = \mathbf{0}$ and the transmitted message consists of $w_{\mathrm{H}}(\mathbf{0}) = 0$ packet. In other words, the source does not transmit anything, and hence the receiver is unaware of the transmission.

If $\mathcal{C}$ is not a perfect code, then this issue can be solved by transmitting $\mathbf{x}\mathbf{G} + \mathbf{b}$ instead, where $\mathbf{b}$ is not a codeword (or, without loss of generality, $\mathbf{b}$ is a coset leader) at distance greater than $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ from the code. We then have $w_{\mathrm{H}}(\mathbf{x}\mathbf{G} + \mathbf{b} + \mathbf{a}) > 0$ for any $\mathbf{x} \in \mathrm{GF}(2)^K$ and any $\mathbf{a}$ with weight no more than t, and hence the messages always have a positive number of packets. Note that this new encoding is equivalent to using the code $\mathcal{C}+\mathbf{b}$, which has the same minimum distance and the same distance distribution as $\mathcal{C}$. On the other hand, if $\mathcal{C}$ is a perfect code, then no such vector $\mathbf{b}$ exists. However, we remark that perfect linear binary codes exist for limited sets of parameters only [11, Chapter 6, Theorem 33], and hence most of the codes that would be used in our setting are not perfect. Also, if such a code should be used, then selecting a translate vector $\mathbf{b}$ with Hamming weight t may reduce the probability of obtaining an empty message at the receiver's end. Finally, if a large perfect code is used, then removing one codeword from the codebook would make it non-perfect with only a slight decrease in rate.

We now investigate some of the desirable properties of the translate vector $\mathbf{b}$ and how to determine it once the property is fixed. Note that for any linear code $\mathcal{C}$ and any coordinate $0 \le i \le N-1$, then either all codewords in $\mathcal{C}$ have a 0 in coordinate i or one half have a 0 and the other half have a 1. In the first case, the coordinate i is not coded and hence need not be transmitted. Therefore, without loss of generality, we assume the code $\mathcal{C}$ satisfies the following property: for all $0 \le i \le N-1$, half the codewords in $\mathcal{C}$ have a 0 in coordinate i and half have a 1. Also, we assume that $\mathcal{C}$ is not a perfect code, and we denote the set of coset leaders of $\mathcal{C}$ with weight greater than t as $B(\mathcal{C})$. Since the distribution of the number of packets received at the receiver's end depends on the distribution of the error patterns and hence on the channel, we only consider the distribution of the number of packets in a message sent by the source. This assumption is consistent with our choice of studying constant-weight codes in Section III, which is equivalent to considering a constant number of packets in the messages sent by the source.

First, we remark that the average number of packets in a message, given by $\mu(\mathbf{b}) = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c} \in \mathcal{C}} w_{\mathrm{H}}(\mathbf{c} + \mathbf{b})$, is independent of $\mathbf{b}$ and is hence denoted by μ. Denoting the weight distribution of $\mathcal{C}$ as A_i for $0 \le i \le N$, we have $\mu = \sum_{i=0}^{N} i A_i$.

Second, one of the main advantages of using constant-weight codes is the constant number of packets in all messages. In other words, the standard deviation of the number

ISBN 978-3-8007-3211-1

of packets in a message is equal to zero. Accordingly, the translate vector $\mathbf{b}$ may be chosen as to minimize the standard deviation of the number of packets in a message, so that the receiver usually expects a number of packets that is near the average. The minimum standard deviation of the number of packets over all possible choices for $\mathbf{b}$ is given by $\sigma = \min_{\mathbf{b}\in B(\mathcal{C})} \sigma(\mathbf{b})$, where

$$\sigma(\mathbf{b}) = \sqrt{\frac{1}{|\mathcal{C}|}\sum_{\mathbf{c}\in\mathcal{C}} \left(w_{\mathrm{H}}(\mathbf{c}+\mathbf{b}) - \mu\right)^2}. \tag{13}$$

Proposition 5 below gives bounds on σ.

Proposition 5: The minimum standard deviation σ of a linear code $\mathcal{C} \subseteq \mathrm{GF}(2)^N$ with error correction capability t satisfies

$$\sqrt{(t+1)^2 - \frac{\mu^2}{|\mathcal{C}|}} \leq \sigma \leq \max\{\mu - t + 1, N - \mu\}. \tag{14}$$

We have $\sigma = \sqrt{\frac{\Sigma-\mu^2}{|\mathcal{C}|}}$, where $\Sigma = \min_{\mathbf{b}\in B(\mathcal{C})} \sum_{\mathbf{c}\in\mathcal{C}} w_{\mathrm{H}}(\mathbf{c}+\mathbf{b})^2$. Therefore, the value of σ can be determined by solving the following binary linear program. Denote the elements of $B(\mathcal{C})$ as $\mathbf{b}_i$ for $0 \leq i \leq |B(\mathcal{C})|-1$ and consider the vector $\mathbf{s} = (\sigma(\mathbf{b}_0), \sigma(\mathbf{b}_1), \ldots, \sigma(\mathbf{b}_{|B(\mathcal{C})|-1}))$. Then σ is the optimal solution of

$$\begin{aligned} \min \quad & \mathbf{s}\cdot\mathbf{x} && (15)\\ \text{subject to} \quad & \mathbf{1}\cdot\mathbf{x} = 1 && (16)\\ & \mathbf{x} \in \{0,1\}^{|B(\mathcal{C})|}. && (17) \end{aligned}$$

Finally, according to Chebyshev's inequality [14], minimizing the standard deviation of the number of packets minimizes the probability that the messages have a large number of packets. Such restriction on the size of large messages can be tightened by choosing $\mathbf{b}$ to minimize the maximum number of packets in a message. This choice is equivalent to determining $m = \min_{\mathbf{b}\in B(\mathcal{C})} \max_{\mathbf{c}\in\mathcal{C}} w_{\mathrm{H}}(\mathbf{c}+\mathbf{b})$. By definitions of μ and m, we easily obtain that $m \geq \mu$. We remark that m can also be viewed as the optimal solution of a linear program similar to (17)-(19).

V. Conclusion

In this paper, we introduce a novel model for data transmission using store and forward, where we demonstrate that store and forward can be viewed as the transmission of binary vectors. Therefore, the correction of packet errors, losses, and injections can be performed using binary error-correcting codes. We then investigate using two classes of binary codes: constant-weight codes and linear codes. Constant-weight codes have many desirable properties and allow us to demonstrate that the traditional erasure codes, based on coding across packets and indexing the packets in order to recover their original order, form a suboptimal subclass of constant-weight codes. Therefore, our approach offers a higher rate than the traditional erasure codes. However, the high encoding and decoding complexity of constant-weight codes make them difficult to implement in real-life applications. Linear codes, on the other hand, have low encoding and decoding complexity. However, they suffer from different issues when used for error correction in store and forward. These issues are introduced and partially addressed in this paper.

The work in this paper can be extended in multiple ways. First, our investigation of the implementation of binary codes can be extended. Our study of constant-weight codes and their relations to liftings of Hamming metric codes indicates that a compromise between rate gain and low complexity is desirable. Also, the issues arising when using linear codes can be further investigated, hopefully leading to possible implementation of these codes. Furthermore, specific classes of linear codes, such as Hamming codes or Reed-Muller codes, can be studied individually. Second, our transmission model can be generalized in several fashions. This model focuses on one-shot error correction, viewing each message as a codeword of a binary code. Multi-shot error-correcting codes, where a codeword consists of several packets, can offer a higher rate gain. However, using multi-shot codes would lead to another increase of complexity, hence another compromise between rate gain and low complexity needs to be determined. Our model is also based on block codes, while convolutional techniques may be more advantageous in terms of complexity.

References

[1] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, August 1998.

[2] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, April 1997.

[3] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, April 1997.

[4] Y. Xu and T. Zhang, "Variable shortened-and-punctured Reed-Solomon codes for packet loss protection," *IEEE Trans. Broadcast.*, vol. 48, no. 3, pp. 237–245, September 2002.

[5] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith, "A new table of constant weight codes," *IEEE Trans. Info. Theory*, vol. 36, no. 6, pp. 1334–1380, November 1990.

[6] E. Agrell, A. Vardy, and K. Zeger, "Upper bounds for constant-weight codes," *IEEE Trans. Info. Theory*, vol. 46, no. 7, pp. 2373–2395, November 2000.

[7] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Info. Theory*, vol. 54, no. 8, pp. 3579–3591, August 2008.

[8] D. Silva and F. R. Kschischang. (2008) On metrics for error correction in network coding. [Online]. Available: http://arxiv.org/abs/0805.3824

[9] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance-Regular Graphs*, ser. A Series of Modern Surveys in Mathematics. Springer-Verlag, 1989, vol. 18, no. 3.

[10] P. Delsarte and V. I. Levenshtein, "Association schemes and coding theory," *IEEE Trans. Info. Theory*, vol. 44, no. 6, pp. 2477–2504, October 1998.

[11] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[12] S.-T. Xia and F.-W. Fu, "Johnson type bounds on constant dimension codes," *Designs, Codes and Cryptography*, vol. 50, no. 2, pp. 163–172, February 2009.

[13] L. A. Bassalygo, "New upper bounds for error correcting codes," *Problems of Information Transmission*, vol. 1, no. 4, pp. 32–35, October-December 1965.

[14] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed., 2002.

ISBN 978-3-8007-3211-1 © VDE VERLAG -GMBH · Berlin · Offenbach