Low Delay Random Linear Coding Over a Stream

Mohammad Karzand, Douglas J. Leith

***Abstract*—We introduce a random linear code construction for erasure packet channel. We then analyze its in-order delivery delay behavior. We show that for rates below the capacity, the mean in-order-delivery delay of our scheme is better than the mean delay introduced by the scheme which implements the random linear block coding. We also compute the decoding failure probability and encoding and decoding complexity of our scheme.**

I. Introduction

It is known that TCP performs poorly in the presence of interference [1], or when the round trip time is large or there exist high packet loss rates over the networks [2]. To overcome the challenges posed by these networks, a transport layer solution referred to as Network Coded TCP is proposed in [3]. The adaptation of transport layer approach ensures backward compatibility with legacy equipment and easy implementation.

This motivates us to investigate different error correcting approaches tailored to packet streams. The challenges with real-time constraints on information transmission are very different from the classical information theory problem. The fundamental theorems of information theory are asymptotic in nature and deal with transmission of sequences of long blocks that introduces long delays in communication. Recent information theoretic results in [4] investigate the trade-off between block code length and reliability for short sequences. However, it is not instantly obvious that these block coding schemes minimize the in order delivery delay for communication rates below capacity. As an example, for the BEC with erasure probability $\epsilon = 0.01$ and a block coding scheme with maximal block error rate 10^{-3} and coding rate 0.95 the minimum blocklength of the code should be at least 60 [4]. This results in a large in order delivery delay in case of an erasure.

In this note, we are interested in the tradeoff between the throughput and the delay. As mentioned, the study is motivated by transport layer coding, but it is also of more general interest where in order delivery delay is important. We are investagating alternative methods to transmit data other than block coding schemes to reduce in order delivery delay for rates well below capacity.

In section II, we introduce a random linear code construction. In section III, the in-order delivery delay and the throughput of a system based on this code is analyzed. In section IV, the decoding failure probability is computed. In section V, the encoding and decoding complexity of the coding scheme is investigated and in section VI, we compare our scheme with the scheme based on random linear block codes through simulations. The section VII talks about the path ahead.

II. Low Delay Coding Over a Stream

A. Problem Setup

We consider a real time streaming system where time is slotted, with each slot corresponding to one packet. In each slot the sender transmits either an information packet or a coded packet over an erasure channel to one receiver. Packets are erased with i.i.d probability ϵ. At the receiver arriving packets are stored in a reassembly buffer until a decoder is able to reconstruct the information packet stream and deliver the information packets in-order to an application. We seek to analyze the in order delivery delay of such a scheme.

B. Low Delay Code Construction

We consider the following code construction. A coded packet c_i is inserted after every $l-1$ information packets u_i, and the distance between coded packets is constant l. Each coded packet is constructed as a random linear combination of the preceding information packets. We assume that information packets and the coded packets are drawn from an alphabet with cardinality of Q and

$$c_i = f_i(u_1, u_2, \dots, u_{(l-1)*i}) := \sum_{j=1}^{(l-1).i} w_{ij} \cdot u_j \qquad (1)$$

with coefficients w_{ij}, selected identically and independently, uniformly at random from field with size Q. At the receiver, at slot i generator matrix G_i is constructed with rows formed from the coefficients of the received packets (information packets adding a row with a 1 in the corresponding diagonal, all other entries being 0). Decoding can then be carried out on-the-fly using *e.g.* Gaussian elimination.

In our analysis, unless stated otherwise, we will make the standing assumption that the field size Q is sufficiently large that with probability one each coded packet helps the receiver recover from one information packet erasure. That is, each coded packet row added to generator matrix G_i increases the rank of G_i by one. Our experimental measurements do not, of course, make this assumption. We also compute the decoding failure probability in section IV, taking into account the probability that the matrix is not full rank.

This method is different from block code constructions, as the coded packets are constructed causally and streamed in between information packets. This construction will help the receiver to mitigate the effect of an

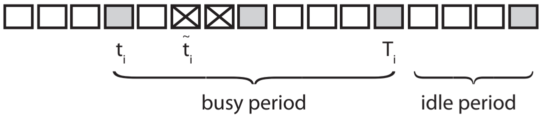


Fig. 1. Illustrating notation used. Clear rectangles indicate information packets, shaded rectangles coded packets, crosses indicate erasures, coded packets are inserted every $l = 4$ slots. t_i is the coded packet slot immediately preceding the information packet erasure at slot $\tilde{t}_i$ which pauses in-order delivery at the receiver, T_1 the coded packet slot at which in-order delivery resumes. The information packet at slot $t_i + 1$ is delivered without delay, but any information packets in slots $\{\tilde{t}_i, \cdots, T_i\}$ are delayed.

erasure sooner since the receiver does not need to wait till the end of the block to start the decoding procedure, however the causal construction will limit the power of the code. The goal is to analyze the behavior of the decoding delay and compare it to concatenated block codes.

In this construction, we assume that the distance between two coded packets is constant and deterministic. The delay analysis of randomized placement of coded packets will be dealt in a later work.

III. In-Order Delivery Delay and Throughput

A. In-order delivery delay

Information packets are delivered in-order at the receiver until an erasure of an information packet occurs. Upon erasure, in-order delivery is paused (arriving packets are buffered) until the decoder receives as many coded packets as the number of erasures, at which point in-order delivery resumes.

Let $\{\tilde{t}_i\}$ denote the sequence of slot times at which erasure of an information packet pauses in-order delivery and $\{T_i\}$ the corresponding sequence of times at which in-order delivery resumes. Note that the T_i must be a slot at which a coded packet is transmitted. Letting $t_i = \lfloor \tilde{t}_i / l \rfloor l$ be the coded packet slot immediately preceding slot $\tilde{t}_i$, we can then define the sequence of coded packet slots $\{t_1, T_1, t_2, T_2, \cdots\}$. See Figure 1 for a schematic illustration. Slots $\{t_i + 1, t_i + 2, \cdots, T_i\}$ contain information packets delayed by the i'th pause, plus perhaps non-delayed packets $\{t_i + 1, \tilde{t}_i\}$ and this set of slots is referred to as the i'th "busy" period. Slots $\{T_i + 1, \cdots, t_{i+1}\}$ can be partitioned into intervals $\{T_i + 1, T_i + l\}$, $\{T_i + l + 1, T_i + 2l\}$ *etc* each of size l slots and ending with a coded packet slot (since T_i and t_i are both coded packet slots). Each of these intervals of l slots is referred to as an "idle" period.

The busy/idle period terminology is by analogy with a queueing system operating in embedded time corresponding to the coded packet slots. Information packet erasures can be thought of as queue arrivals and reception of coded packets as queue service. Pauses in in-order delivery then correspond to periods when the queue size is non-zero.

Index the busy/idle periods by $j = 1, 2, \cdots$ and let $i(j)$ be the index of the pause corresponding to the j'th busy period i.e. the j'th busy period consists of slots $\{t_{i(j)}, \cdots, T_{i(j)}\}$. With the j'th period we associate a random variable S_j, with $S_j = 0$ for an idle period and $S_j = (T_{i(j)} - t_{i(j)})/l$ for a busy period i.e. S_j equals the number of coded packets transmitted before delivery resumes. Since packet erasures are i.i.d., the busy/idle periods form a renewal process and the $\{S_j\}$ are i.i.d. Letting $S \sim S_j$ the following theorem characteries the probability distribution of S.

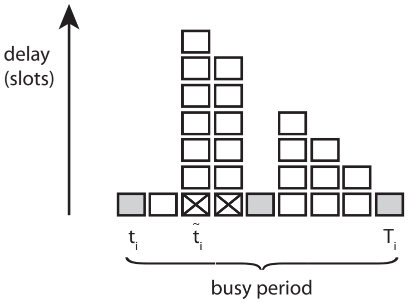


Fig. 2. Illustrating delay introduced by erasures. In this example the information packet at slot $\tilde{t}_i$ is delayed by 6 slots, the information packet at slot $\tilde{t}_i + 1$ by 5 slots, the information packet at $\tilde{t}_i + 3$ by 3 slots and so on. It can be seen that the sum-delay is the area under a triangle of base $T_i - t_i$ slots and height $T_i - t_i$ slots, less the area associated with any coded packets.

Theorem 1 (Busy Time). *In an erasure channel with erasure probability ϵ, suppose we insert a coded packet in between every $l - 1$ information packets. Assume that each coded packet can help us to recover from one erasure. We have:*

I. For all values of ϵ and l such that $l\epsilon < 1$, the mean of the probability distribution of S exists and is finite.

II.
$$P(S = 0) = (1 - \epsilon)^{l-1} \tag{2}$$
$$P(S = 1) = (l - 1)\epsilon(1 - \epsilon)^{l-1} \tag{3}$$
$$P(S = k) = \frac{l-1}{k}\epsilon^k (1 - \epsilon)^{k(l-1)} \binom{(k-1)l}{k-1}, \ \forall k > 1 \tag{4}$$

III.
$$E(S) = \frac{(l-1)\epsilon(1-\epsilon)^{l-1}}{1 - l\epsilon} \tag{5}$$
$$E(S^2) = E(S) + \frac{l(l-1)\epsilon^2(1-\epsilon)^l}{(1 - l\epsilon)^3} \tag{6}$$

Observe that the requirement that $l\epsilon < 1$ for S to have finite mean is a natural one. The rate of the coding scheme is $R = \frac{l-1}{l} = 1 - \frac{1}{l}$. Since this rate of transmission should be less than the channel capacity, we require $R < 1 - \epsilon$, and so $l\epsilon < 1$. Combining Theorem 1 with the following result allows us to obtain a simple closed-form bound on the mean in-order delivery delay:

Theorem 2 (In-Order Delivery Delay). *At the receiver, the mean in-order delivery delay for information packets is upper bounded by $\frac{E[S^2]}{\max\{1, E[S]\}} \frac{l}{2}$ slots.*

Using simulations in Fig. 3, we show that the upper bound on in-order delivery delay is reasonably tight.

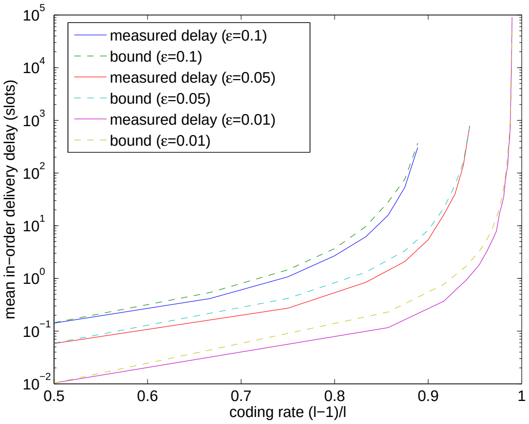


Fig. 3. Measured in-order delivery delay and upper bound vs coding rate $(l-1)/l$ and packet erasure rate ϵ.

B. Throughput and rate

The rate of transmission of this scheme is $\frac{l-1}{l} = 1 - \frac{1}{l}$.

The Good Throughput (GT) of this scheme is a random variable and its behavior is characterized in the following theorem:

Theorem 3 (Throughput)**.** *Consider the transmission of a coded stream of length N over an erasure channel. For any $R_0 < 1 - \frac{1}{l}$ and $\delta > 0$, there exist an N large enough such that*

$$\Pr\left(GT > R_0\right) > 1 - \delta.$$

The length of the possible failure at the end of transmission is independent of length of the transmission and as the length of the transmission grows, the sacrifice in the good throughput of the system becomes negligible.

IV. Computation of The Decoding Failure Probability

Till here, we assume that the decoding procedure is always successful upon the reception of enough coded packets. In this section we are going to find upper and lower bounds to the probability of decoding failure.

We assumed that information packets and the coded packets are drawn from an alphabet with the cardinality of Q and

$$c_i = f_i(u_1, u_2, \ldots, u_{(l-1)*i}) := \sum_{j=1}^{(l-1).i} w_{ij} \cdot u_j \qquad (7)$$

with coefficients w_{ij}, drawn identically and independently, uniformly at random from field with size Q. At the receiver and at the time kl, assuming a certain pattern of erasures and that enough coded packets have been received which in our case means $S = k$, the decoding matrix is formed based on the the coefficients of w_{ij} and the decoding can then be carried out on-the-fly using *e.g.* Gaussian elimination. The decoding will be successful if the decoding matrix is of full rank. In what follows, we provide upper and lower bounds to the probability of the decoding matrix to be full rank.

Assume that k erasures happen and the erasure pattern is admissible as defined in the lemma **??**[1], so the decoding is performed exactly after receiving k coded packets in k l-interval. Assume $E_1, E_2, \ldots, E_k$ are the number of erasures in each l-interval. Constructing the decoding matrix, the number of non-zero w_{ij}s in each row i is equal to $\sum_{z=1}^{i} E_z$. We also know that since the decoding does not stop before k, the number of erasures in any k' l-interval for any k' less than k is strictly greater than k'. We denote a decoding matrix admissible, if its elements are formed based on an admissible erasure pattern.

Looking at the admissible decoding matrices from a column perspective, the number of zeros in each column i is equal to the number of rows j in which $\sum_j E_j$ is strictly less than i.

To illustrate this further, assume that $S = 3$ which means there have been three erasures in the first three intervals. There exists two admissible erasure patterns that can lead to this case:

- All the erasures happen in the first interval. The decoding operation is carried out as follows:

$$\begin{pmatrix} w_{1e_1} & w_{1e_2} & w_{1e_3} \\ w_{2e_1} & w_{2e_2} & w_{2e_3} \\ w_{3e_1} & w_{3e_2} & w_{3e_3} \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}.$$

  In this case $E_1 = 3, E_2 = 0, E_3 = 0$ and as we explained the number of w_{ij}s in each row i is equal to $\sum_{z=1}^{i} E_z$. Furthermore, none of columns contains any zero because for all columns i , all $\sum_j E_j$ is greater than or equal to i.

- Two erasures happen in the first interval and the last one happens in the second interval. The decoding operation is carried out as follows:

$$\begin{pmatrix} w_{1e_1} & w_{1e_2} & 0 \\ w_{2e_1} & w_{2e_2} & w_{2e_3} \\ w_{3e_1} & w_{3e_2} & w_{3e_3} \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

  In this case $E_1 = 2, E_2 = 1, E_3 = 0$ and as we explained the number of w_{ij}s in each row i is equal to $\sum_{z=1}^{i} E_z$. Furthermore, the third column contains one zero because for the first row $\sum_j E_j = 2$ is less than 3.

Theorem 4 (Decoding failure for $S = k$)**.** *Consider an admissible decoding matrix $G_{k\times k}$, and assume that the w_{ij}s are drawn identically and independently, uniformly at random from field with size Q. The probability of the matrix G to be of full rank is bounded as follows:*

$$\Pr(rank(G) = k) \leq \prod_{j=0}^{k-1} (1 - \frac{1}{Q^{k-j}}) \qquad (8)$$

$$\Pr(rank(G) = k) \geq \frac{Q}{Q+1} (1 - \frac{1}{Q^2})^k. \qquad (9)$$

[1] Reminder: The term *admissible* is used here to mean that at least two erasures should happen during a time $t = l$, at least three within $2l$, and so on, so that the decoding process will remain activated for the whole of the time $t = kl$.

Note that these bounds coincide for the cases where $k = 1$ and $k = 2$, as there exists only one admissible decoding matrix in each case.

Theorem 5 (Decoding failure for a stream of length N)**.** *The decoding failure(DF) probability in a stream of length N satisfies*

$$\lim_{N_t \to \infty} \frac{1}{N_t} Pr(DF) \leq \frac{(1-l\epsilon)}{l(1-\epsilon)^l}. \sum_{i=1}^{\infty} (1 - \frac{Q}{Q+1}(1-\frac{1}{Q^2})^i) \Pr(S=i).$$

Corollary 1 (Decoding Failure Probability)**.** *The decoding failure probability satisfies*

$$\lim_{N_t \to \infty} \frac{1}{N_t} Pr(DF) \leq \frac{(1-l\epsilon)}{l(1-\epsilon)^l}. (\frac{Q}{Q+1}(1-\epsilon_0)^{l-1} - (1-\epsilon)^{l-1} + \frac{1}{Q+1}), \quad (10)$$

where ϵ_0 is the solution to the equation

$$\epsilon_0(1-\epsilon_0)^{l-1} = (1 - \frac{1}{Q^2})\epsilon(1-\epsilon)^{l-1}.$$

Since the function $f(\epsilon) = \epsilon(1-\epsilon)^{l-1}$ is increasing for $\epsilon < \frac{1}{l}$, the solution always exists and $\epsilon_0 < \epsilon$.

Note that as Q becomes larger the decoding failure probability goes to zero. As illustrated in the following tables, we argue that in all practical purposes, Q is large enough for this probability to be very small.

TABLE I
The upper bound on decoding failure probability per packet in the base 10 logarithmic scale, $\epsilon = 0.02$.

		$Q = 2^{8x}$				
	x	1	2	4	10	20
l	25	-1.50	-3.91	-8.73	-22.76	-46.84
	30	-1.41	-3.82	-8.64	-22.74	-46.82
	35	-1.38	-3.79	-8.60	-22.75	-46.83
	40	-1.41	-3.82	-8.64	-22.82	-46.91
	45	-1.58	-3.99	-8.81	-23.03	-47.11

TABLE II
The upper bound on decoding failure probability per packet in the base 10 logarithmic scale, $\epsilon = 0.1$

		$Q = 2^{8x}$				
	x	1	2	4	10	20
l	5	-2.24	-4.65	-9.46	-23.45	-47.53
	6	-2.13	-4.54	-9.36	-23.42	-47.51
	7	-2.09	-4.50	-9.31	-23.43	-47.52
	8	-2.11	-4.52	-9.34	-23.51	-47.59
	9	-2.27	-4.69	-9.51	-23.71	-47.79

V. Encoding and Decoding Complexity

Concerning the encoding complexity, note that we do not need to encode over all preceding information packets, but only those not yet received or decoded. To limit the encoding complexity one can propose two schemes. The first scheme uses feedback to keep an approximate track of the decoding process. This will result in an encoding complexity which is polynomial in $E(S)$. The second scheme assumes that the encoding is constrained to kl preceding information packets and hence, there is a small probability of decoding failure $\Pr(S > k)$ (which goes to zero exponentially as k grows as given by the theorem 1).

The decoding complexity in our coding scheme also depends on the process S. Assuming that $S = k$, a simple Gaussian elimination is performed at each step with a total of approximately $\frac{2k^3}{3}$ arithmetic operations[10]. Using the elementary renewal theorem, very similar to the theorem 5 we have:

Theorem 6. *The total number of arithmetic operation C_d in a stream of length N_t which consists of $N_i = \frac{(l-1)N_t}{l}$ information packets, satisfies the following:*

$$\lim_{N_i \to \infty} \frac{1}{N_i} C_d = \frac{3}{2} \frac{1-l\epsilon}{(l-1)(1-\epsilon)^l} E(S^3),$$

where

$$E(S^3) = E(S^2) + \frac{l(l-1)\epsilon^2(1-\epsilon)^l(2-2\epsilon-2l\epsilon^2+l\epsilon+l^2\epsilon^3)}{(1-l\epsilon)^5}.$$

Proof. The proof is very similar to the proof of the theorem 5. The computation of $E(S^3)$ is similar to the computation of $E(S^2)$ performed in the theorem 1. □

Note that the number of arithmetic operations per information packet becomes very large as the coding rate gets closer to the capacity. However, as illustrated in our examples beloew, we propose that one uses such a communication scheme in the low-delay regime, in which the the complexity of decoding is not very high.

TABLE III
The average number of arithmetic operations performed in the decoder per information packet, $\epsilon = 0.02$

l	25	30	35	40	45
# of operations	0.67	1.93	7.11	41.74	769.58

TABLE IV
The average number of arithmetic operations performed in the decoder per information packet, $\epsilon = 0.1$

l	5	6	7	8	9
# of operations	3.13	8.87	32.56	190.96	3525

VI. Experimental Results

In this section, we compare our coding scheme with random linear block codes with feedback. The linear block code with feedback is implemented so that if the decoding of the block code fails at any step, the receiver asks for a retransmission. In our figures, RTT is round trip delay time, cwnd is the congestion windows size, and BDP is the bandwith delay product in the network.

A. Delay/Throughput Performance

In Fig. 4, we compare the mean in-order packet delivery delay for our scheme with random linear block code with different blocklength sizes with the same coding rate. As

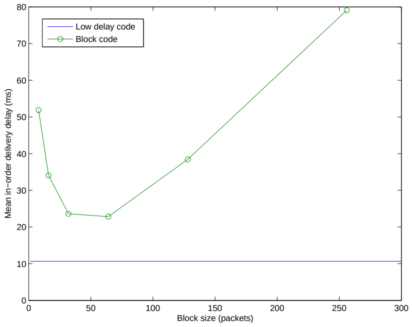
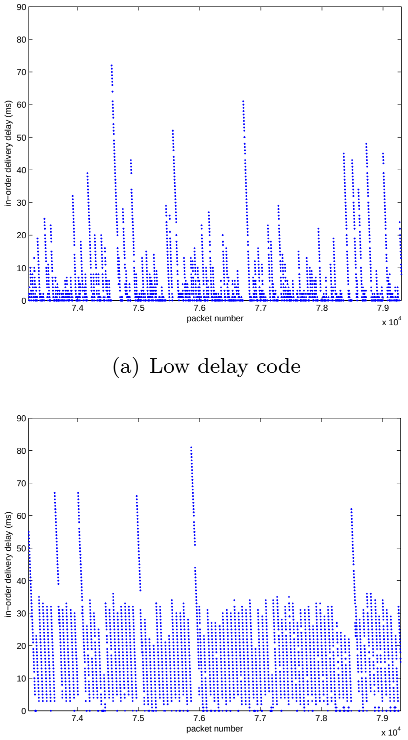


Fig. 4. Mean in-order packet delivery delay vs block size for random linear block code, and for low-delay coding scheme. Link rate 25Mbps, RTT 60ms, loss rate 10%, redundancy 15%, cwnd fixed at BDP (125 packets).

shown in the figure the delay is larger for linear block codes.

In Fig. 5, the time history snapshots of the delay in two schemes are shown. As can be seen, our construction has a better delay profile in the sense that more packets are experiencing lower in-order delivery delay.



(a) Low delay code

(b) Block size 64

Fig. 5. Time history snapshots of in-order packet delivery delay for random linear code with block size 64, and for low-delay coding scheme. Link rate 25Mbps, RTT 60ms, loss rate 10%, redundancy 15%, cwnd fixed at BDP (125 packets).

In Fig. 6, the number of packets transmitted per information packets in both schemes are compared.

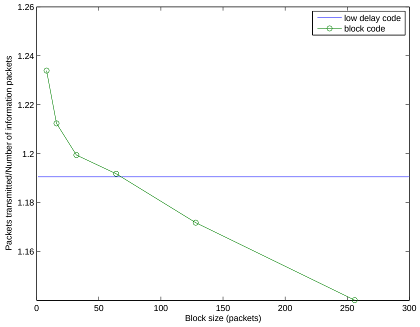


Fig. 6. Packets transmitted vs block size for random linear block code, and for low-delay coding scheme. Link rate 25Mbps, RTT 60ms, loss rate 10%, redundancy 15%, cwnd fixed at BDP.

B. Delay/Throughput vs Coding Rate

In Fig. 7, the mean delivery delay of different coding schemes is compared for different values of coding rate(More precisely, the inverse of the coding rate). As shown in the figure, our scheme performs the best in the sense that for any given redundancy it has the lowest mean in order delivery delay.

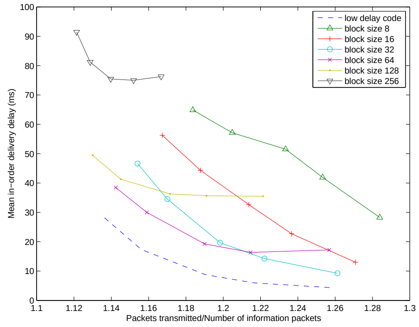


Fig. 7. Mean in-order packet delivery delay vs packets transmitted for random linear block code, and for low-delay coding scheme. Link rate 25Mbps, RTT 60ms, loss rate 10%, cwnd fixed at BDP.

C. Low Delay Block Codes

We can place an upper limit on the number of packets encoded over by partitioning the sequence of information packets into blocks (or perhaps "chunks" is a better terminology) and applying the low delay coding scheme separately within each block. As the block size is increased, the performance can be expected to approach that of the streaming approach.

D. Example: Satellite link

As shown in Fig 9, at a link rate of 10Mbps, the mean delay is 101ms with a block size of 128 packets and 33ms with low delay code – factor of three reduction with low delay approach. At a link rate of 2.5Mbps, the mean delay is 162ms with a block size of 64 packets and 62ms with low delay code – a factor of 2.5 reduction.

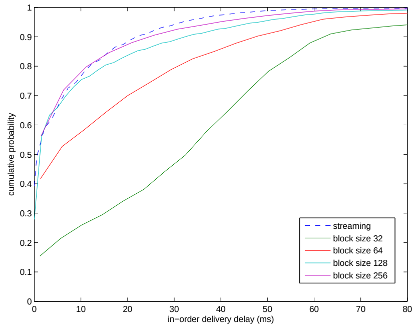
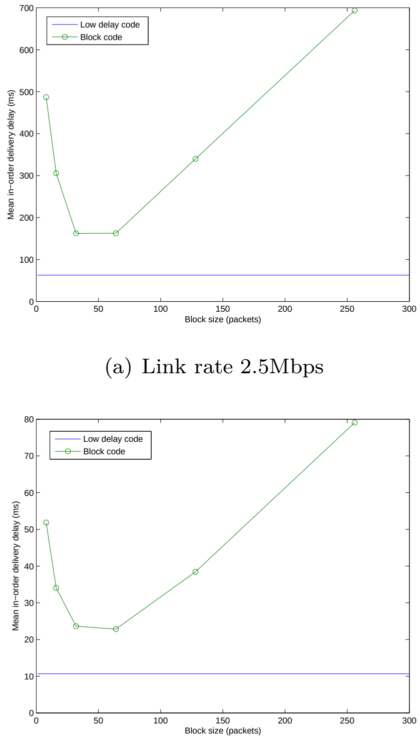


Fig. 8. Distribution of in-order packet delivery delay for low-delay coding scheme when using streaming and when partitioning the information packets into blocks of various sizes. Link rate 25Mbps, RTT 60ms, redundancy 1.5×loss rate, cwnd fixed at BDP.



(a) Link rate 2.5Mbps

(b) Link rate 10Mbps

Fig. 9. Mean in-order packet delivery delay vs packet loss rate for random linear block code, and for low-delay coding scheme. Link RTT 600ms, loss rate 10%, redundancy 15%, cwnd fixed at BDP (500 packets).

VII. Conclusion and Future Work

We introduce a random linear code construction for erasure packet channels and analyze its in-order delivery delay behavior and show that for rates below the capacity, the average in-order-delivery delay is better than the mean delay introduced by the scheme which implements the random linear block coding.

For the future work, we are going to investigate the effect of putting more than one coded packet in each interval which will help us to compare this method to random linear block codes better. We will also investigate the case where the placement of the coded packets is not deterministic.

References

[1] B. Bakshi, P. Krishna, N. Vaidya, and D. Pradhan, *Improving performance of tcp over wireless networks*, in Proceedings of ICDCS, pp. 365373, 1997

[2] A. Pirovano, F. Garcia, *A New Survey on Improving TCP Performances over Geostationary Satellite Link*, Network and Communication Technologies, 2013

[3] M. Kim, J. Cloud, A. ParandehGheibi, L. Urbina, K. Fouli, D. Leith, M. Medard, *Network Coded TCP*, arxive: http://arxiv.org/abs/1212.2291

[4] Y. Polyanskiy, H. V. Poor and S. Verdu, *Channel coding rate in the finite blocklength regime*, IEEE Trans. Inf. Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.

[5] J. C. Tanner *A Derivation of Borel distribution* Biometrika , Vol. 48, No. 1/2, Jun., 1961.

[6] R. L. Graham, D. E. Knuth, O. Patashnik. *Concrete Mathematics (Second ed.)* (1994). Addison-Wesley.

[7] G. Landsberg. *Ueber eine Anzahlbestimmung und eine damit zusammenhangende Reine.*, J. Reine Angew. Math. 111 (1893), 87-88.

[8] E.R. Berlekamp, *The Technology Of Error-Correcting Codes*, Proc.of the IEEE, 68: 564-593 , 1980.

[9] R.G. Gallager, *Stochastic Processes, Theory for Applications*, Cambridge University Press, 2013.

[10] R.W. Farebrother, *Linear Least Squares Computations, statistics: Textbooks and Monographs*, Marcel Dekker, 1988.

[11] U. Tamm, *Majorization in Lattice Path Enumeration and Creating Order*, Preprint 00-108, SFB 343, Discrete Structures in Mathematics, University of Bielefeld, 2000.

[12] G. Kreweras, *Sur une classe de problemes de denombrement lies au treillis des partitions des entiers*, Thesis, University of Paris, 1965.

[13] T. V. Narayana, *Lattice Path Combinatorics*, University of Toronto Press, 1979.

[14] M. Karzand, D. Leith *Low Delay Random Linear Coding over Streams* submitted to IEEE Trans. on Networking.