A Coded Generalization of Selective Repeat ARQ

Jason Cloud
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA USA
email: jcloud@mit.edu

Douglas Leith
School of Computer Science and Statistics
Trinity College
Dublin, Ireland
email: doug.leith@tcd.ie

Muriel Médard
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA USA
email: medard@mit.edu

***Abstract*—Reducing the in-order delivery, or playback, delay of reliable transport layer protocols over error prone networks can significantly improve application layer performance. This is especially true for applications that have time sensitive constraints such as streaming services. We explore the benefits of a coded generalization of selective repeat ARQ for minimizing the in-order delivery delay. An analysis of the delay's first two moments is provided so that we can determine when and how much redundancy should be added to meet a user's requirements. Numerical results help show the gains over selective repeat ARQ, as well as the trade-offs between meeting the user's delay constraints and the costs inflicted on the achievable rate. Finally, the analysis is compared with experimental results to help illustrate how our work can be used to help inform system decisions.**

I. Introduction

Reliable transport protocols are used in a variety of settings to provide data transport for time sensitive applications. In fact, video streaming services such as Netflix and YouTube, which both use TCP, account for the majority of fixed and mobile traffic in both North America and Europe [1]. In fixed, wireline networks where the packet erasure rate is low, the quality of user experience (QoE) for these services is usually satisfactory. However, the growing trend towards wireless networks, especially at the network edge, is increasing non-congestion related packet erasures within the network. This can result in degraded TCP performance and unacceptable QoE for time sensitive applications. While TCP congestion control throttling is a major factor in the degraded performance, head-of-line blocking when recovering from packet losses is another. This paper will focus on the latter by applying coding techniques to overcome lost packets and reduce head-of-line blocking issues so that overall in-order delivery delay is minimized.

Head-of-line blocking issues result from using techniques like selective repeat automatic-repeat-request (SR-ARQ), which is used in most reliable transport protocols (e.g., TCP). While it helps to ensure high efficiency, one problem with SR-ARQ is that packet recovery due to a loss can take on the order of a round-trip time (RTT) or more [2]. When the RTT (or more precisely the bandwidth-delay product (BDP)) is very small and feedback is close to being instantaneous, SR-ARQ provides near optimal in-order delivery delay. Unfortunately, feedback is often delayed due to queuing and large physical distances between a client and server. This can have major implications for applications that require reliable delivery with constraints on the time between the transmission and in-order delivery of a packet. As a result, we are forced to look at alternatives to SR-ARQ.

This paper explores the use of a systematic random linear network code (RLNC), in conjunction with a coded generalization of SR-ARQ, to help reduce the time needed to recover from losses. Redundancy is added to the original data stream by injecting coded packets at key locations. This helps reduce delay by overcoming packet losses and limiting the number of required retransmissions. However, correlated losses or incorrect information about the network can result in the receiver's inability to decode. Feedback and coded retransmissions are used to ensure that this does not happen.

The following paper will provide the answers to two questions: when should redundant packets be inserted into the original packet stream to minimize in-order delivery delay; and how much redundancy should be added to meet a user's requested QoE. These answers will be provided through an analysis of the in-order delivery delay as a function of the coding window size and redundancy. We will then use numerical results to help determine the cost (in terms of rate) of reducing the delay and as a tool to help determine the appropriate coding window size for a given network path/link. While an in-depth comparison of our scheme with others is not within the scope of this paper, we will use SR-ARQ as a baseline to help show the benefits of coding at the transport layer.

The remainder of the paper is organized as follows. Section II provides an overview of the related work in the area of transport layer coding and coding for reducing delay. Section III describes the coding algorithm and system model. Section IV provides the tools needed to analyze the proposed scheme; and an analysis of the first two moments of the in-order delay are provided in Sections V and VI. Furthermore, the throughput efficiency is derived in Section VII to help determine the cost of coding. Numerical results are finally presented in Section VIII and we conclude in Section IX.

II. Related Work

A resurgence of interest in transport layer coding has taken place to help overcome TCP's poor performance in wireless networks. Sundararajan et. al. [3] first proposed TCP with Network Coding (TCP/NC). A coding shim is inserted between the TCP and IP layers that introduces redundancy into the network. This spoofs TCP into believing the network is error-free. Loss-Tolerant TCP (LT-TCP) [4], [5], [6] is another approach using Reed-Solomon (RS) codes and explicit congestion notification (ECN) to overcome random packet erasures and improve performance. In addition, Coded TCP (CTCP) [7] uses RLNC [8] and a modified additive-increase, multiplicative decrease (AIMD) algorithm for maintaining high throughput in high packet erasure networks. These proposals have shown coding can help increase throughput, but only anecdotal evidence has been provided showing the benefits for time sensitive applications.

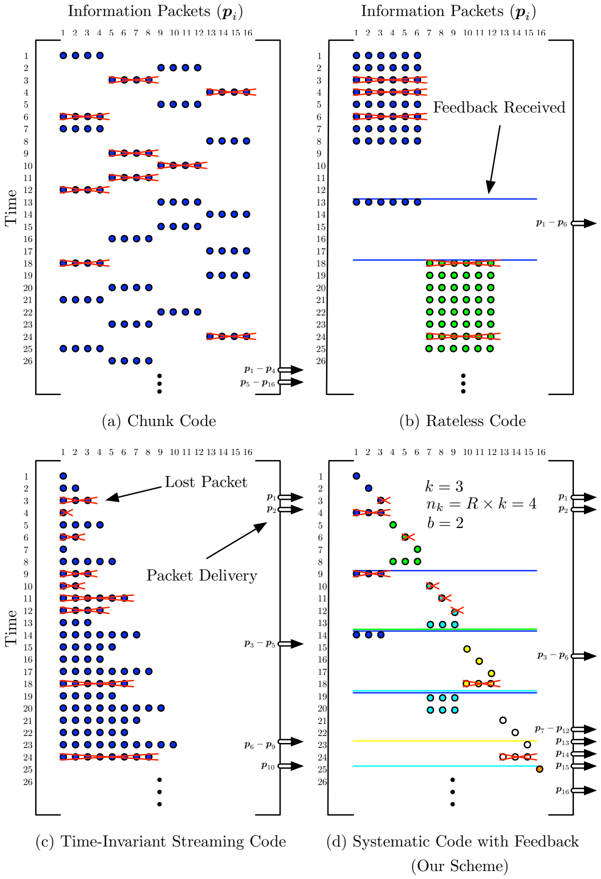
On the other hand, a large body of research investigating coding delay in different settings has taken place. Most of these works can be summarized by Figure 1. The rows and columns of each matrix in the figure indicate the time and the specific information/uncoded packets, $\boldsymbol{p}_i$, that need to be transmitted respectively. The composition of the transmitted packet is shown by the dots in each row, different color dots indicate specific generations, horizontal lines show the time when feedback about a specific generation is obtained, and the red crosses show lost packets. Furthermore, the time packets are delivered, in-order, to the client application is shown by the double arrows on the right-hand side of each matrix.

The coding delay of chunked and overlapping chunked codes [9] (shown in Figure 1(a)), network coding in time-division duplexing (TDD) channels [10], [11], [12], and network coding in line networks where coding also occurs at intermediate nodes [13] is well understood. In addition, a non-asymptotic analysis of the delay distributions of RLNC [14] and various multicast scenarios [15], [16], [17] using a variant of the scheme in Figure 1(b) have also been investigated. The research that looks at the in-order delivery delay is provided in [2] and [18] for uncoded systems, while [19], [20], and [21] considers the in-order delivery delay for non-systematic coding schemes similar to the one shown in Figure 1(b). However, these non-systematic schemes may not be the optimum strategy in networks or communication channels with a long RTT.

Possibly the closest work to ours is that done by [22], [23] and [24]. Bounds on the expected in-order delay and a study of the rate/delay trade-offs using a time-invariant coding scheme is provided in [22] and [23] where they assume feedback is instantaneous, provided in a block-wise manner, or not available at all. A generalized example of their coding scheme is shown in Figure 1(c). While their analysis provides insight into the benefits of coding for streaming applications, their model is similar to a half-duplex communication channel where the sender transmits a finite block of information and then waits for the receiver to send feedback. Unfortunately, it is unclear if their analysis can be extended to full-duplex channels or models where feedback does not provide complete information about the receiver's state-space. Finally, the work in [24] considers the in-order delivery delay of online network coding where feedback determines the source packets used to generate coded packets. However, they only provide experimental results and do not attempt an analysis.

III. Coding Algorithm and System Model

We consider a time-slotted model to determine the coding window size k and added redundancy $R \geq 1$ that minimizes the in-order delivery, or playback, delay D, which is defined as:


Information Packets ($\boldsymbol{p}_i$)
Information Packets ($\boldsymbol{p}_i$)
Time
Feedback Received
$\boldsymbol{p}_1 - \boldsymbol{p}_6$
$\boldsymbol{p}_1 - \boldsymbol{p}_4$
$\boldsymbol{p}_5 - \boldsymbol{p}_{16}$
Lost Packet
Packet Delivery
$\boldsymbol{p}_1$
$\boldsymbol{p}_2$
$\boldsymbol{p}_3 - \boldsymbol{p}_5$
$\boldsymbol{p}_6 - \boldsymbol{p}_9$
$\boldsymbol{p}_{10}$
$k = 3$
$n_k = R \times k = 4$
$b = 2$
$\boldsymbol{p}_3 - \boldsymbol{p}_6$
$\boldsymbol{p}_7 - \boldsymbol{p}_{12}$
$\boldsymbol{p}_{13}$
$\boldsymbol{p}_{14}$
$\boldsymbol{p}_{15}$
$\boldsymbol{p}_{16}$

(a) Chunk Code

(b) Rateless Code

(c) Time-Invariant Streaming Code

(d) Systematic Code with Feedback (Our Scheme)

Figure 1: Coding matrices for various schemes assuming an identical loss pattern and a feedback delay of 4 time-slots.

Definition 1. The in-order delivery delay D is the difference between the time a packet is first transmitted and the time that the same packet is delivered, in-order.

The duration of each time-slot is $t_s = {}^{s_{pkt}}/_{Rate}$ where s_{pkt} is the packet size and $Rate$ is the transmission rate of the network. The propagation delay between the sender and the receiver is t_p (i.e., $RTT = t_s + 2t_p$ assuming that the acknowledgement size is sufficiently small). Packet erasures are assumed to be independently and identically distributed (i.i.d.) with ϵ being the packet erasure probability.

Information packets $\boldsymbol{p}_i$, $i = \{1, \ldots, N\}$, are first partitioned into coding generations $\boldsymbol{G}_j = \left\{\boldsymbol{p}_{(j-1)k+1}, \ldots, \boldsymbol{p}_{\min(jk,N)}\right\}$, $j \in [1, \lceil {}^{N}/_{k} \rceil]$, of size k packets as they are made available to the transport layer. Once enough information packets are available to the encoder to fill a generation, random linear combinations of these packets produce coded packets $\boldsymbol{c}_{j,m}$, $m \in [1, n_k - k]$ where $n_k = Rk$. Both the information and coded packets are then transmitted. This systematic network scheme is summarized in Algorithm 1 where the coding coefficients $\alpha_{i,j,m} \in \mathbb{F}_q$ are chosen at random and each $\boldsymbol{p}_i$ is treated as a vector in $\mathbb{F}_q$. Once this process completes for one $\boldsymbol{G}_j$, the coding window slides to the next generation $\boldsymbol{G}_{j+1}$ and the process repeats without waiting for feedback.

We assume that feedback is delayed (i.e., $t_p > Rkt_s$ and multiple generations can be in-flight at any time), and

Algorithm 1: Code Generation Algorithm

```
for each $j \in [1, \lceil N/k \rceil]$ do
    for each $i \in [(j-1)k+1, \min(jk, N)]$ do
        Transmit $\boldsymbol{p}_i$
    for each $m \in [1, n_k - k]$ do
        Transmit $\boldsymbol{c}_{j,m} = \sum_{i=(j-1)k+1}^{\min(jk,N)} \alpha_{i,j,m}\boldsymbol{p}_i$
```

Algorithm 2: DOF Retransmission Algorithm

```
$ACK$ From $G_j$ Received
if No packets from $G_j$ in-flight and $l > 0$ then
    for each $m \in [1, Rl]$ do
        Transmit $\boldsymbol{c}_{j,m} = \sum_{i=(j-1)k+1}^{\min(jk,N)} \alpha_{i,j,m}\boldsymbol{p}_i$
```

it contains the number of degrees of freedom ($dofs$) l still required to decode the generation. If $l > 0$, an additional $n_l = Rl \geq l$ coded packets (or $dofs$) are retransmitted. This process is shown in Algorithm 2. When at least k $dofs$ have been received, the generation is decoded and delivered. This naturally leads to the concept of rounds.

Definition 2. The ith round for a single generation begins with the transmission of n_j $dofs$ where j is the number of $dofs$ required by the receiver at the conclusion of round $i-1$; and it ends when feedback is obtained from the receiver regarding the number of $dofs$ still needed to successfully decode.

Figure 1(d) provides an example of this concept as well as the proposed scheme. Here we see that information packets are partitioned into coding generations of size $k = 3$ packets, and one coded packet is transmitted for redundancy (i.e., $R = 1.33$). Round 1 begins for the first (blue) generation at time $t = 1$ with the transmission of $n_3 = 4$ $dofs$. In this example, the first two packets are delivered, but the remaining two are lost resulting in a decoding error. The client sends feedback, which is received by the server at $t = 8$ indicating that $l = 1$ dof is still needed. Round 2 begins at $t = 9$ with the transmission of $n_1 = 1$ dof, and round 3 begins at $t = 14$. Since the final dof required to decode the generation is obtained during round 3, no subsequent rounds are required. Likewise, the second (green) generation begins its round 1 at $t = 5$. Three $dofs$ are received and the generation can be immediately decoded. However, the second generation cannot be delivered until the first generation is decoded and delivered at $t = 16$.

Before proceeding, several assumptions are needed due to the complexity of the process. First, retransmissions occur immediately after feedback is obtained indicating additional $dofs$ are needed without waiting for the coding window to shift to a new generation. Second, the time to transmit packets after the first round does not increase the delay. For example, the packet transmission time is t_s seconds. Assuming l dof retransmissions are needed, the additional $n_l t_s$ seconds needed to transmit these packets are not taken into account. Third, the number of previously transmitted generations that can cause head-of-line blocking is limited to $b-1$ where $b = \lceil BDP/n_k \rceil$.

It is important to note that these assumptions lower bound the delay. The first two assumptions ensure feedback is acted upon immediately and does not impact the delay experienced by other generations. The third assumption limits the possibility of a previously transmitted generation preventing delivery, thereby decreasing the overall delay. Finally, we make two additional assumptions that do not explicitly affect the delay. All packets within a generation are available to the transport layer without delay (i.e., we assume an infinite packet source); and the coding window/generation size with the added redundancy is smaller than the BDP (i.e., $n_k < BDP$). Without the final assumption, feedback will be received prior to the transmission of the coded packets allowing for the use of SR-ARQ without a large impact to the performance.

IV. Preliminaries

We first define several probability distributions and random variables that will be used extensively in later sections. Define $[P] \in \mathbb{R}^{(k+1)\times(k+1)}$ to be the transition matrix of a Markov chain. Each transition within the chain represents the number of $dofs$, or packets, successfully received after a round of transmissions, and each state represents the number of $dofs$ still needed by the client to decode. As a result, the elements of $[P]$ can be defined as follows:

$$[P_{ij}] = \begin{cases} B\left(n_i, i-j, 1-\epsilon\right) & \text{for } i \in [1,k], 0 < j \leq i \\ \sum_{m=i}^{n_i} B\left(n_i, m, 1-\epsilon\right) & \text{for } i \in [1,k], j = 0 \\ 1 & \text{for } i = 0, j = 0, \end{cases} \tag{1}$$

where $B(n,k,p) = \binom{n}{k}p^k(1-p)^{n-k}$. Let X_r be the state of the chain at time r. It follows that $\Pr\{X_r = j | X_0 = i\} = \left[P_{ij}^r\right]$ for $r \geq 1$ and $\left[P_{ij}^0\right] = 0$. In our model, $X_0 = k$ with probability equal to 1 and a generation is successfully decoded when state 0 is entered at time $r \geq 1$. Furthermore, the probability $\left[P_{i0}^r\right]$ is the probability that all packets within a single generation have been successfully received in or before r transmission rounds.

Using this Markov chain, define Y to be the number of transmission rounds required to transfer a single generation. The distribution on Y is:

$$p_Y(y) = \begin{cases} \left[P_{k0}^y\right] - \left[P_{k0}^{y-1}\right] & \text{for } y \geq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Next, define Z_i to be the number of transmission rounds required to transfer i generations. The distribution on Z_i is provided by the following lemma (see appendix for proof):

Lemma 3. *Let N independent processes defined by the transition matrix $[P]$ start at the same time. The probability that all processes complete in less than or equal to z rounds, or transitions, with at least one process completing in round z is Pr$\{Z = z\} = \left[P_{k0}^z\right]^N - \left[P_{k0}^{z-1}\right]^N$.*

Specifically, $p_{Z_i}(z_i)$ is:

$$p_{Z_i}(z_i) = \begin{cases} \left[P_{k0}^{z_i}\right]^i - \left[P_{k0}^{z_i-1}\right]^i & \text{for } z_i \geq 1, i \leq b-1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Also define S to be the number of uncoded packets that are successfully transferred within a generation prior to the first packet loss. The distribution on S is:

$$p_{S|Y}(s|y) = \begin{cases} \epsilon\left(1-\epsilon\right)^{s} & \text{for } s \in [0, k-1], y = 1 \\ \left(1-\epsilon\right)^{s} & \text{for } s = k, y = 1 \\ \frac{\epsilon(1-\epsilon)^{s}}{1-(1-\epsilon)^{k}} & \text{for } s \in [0, k-1], y \neq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

and its first three moments are given by the following lemma (see appendix for proof).

Lemma 4. *Define* $\overline{s_1^i} = \mathbb{E}\left[S^i | Y = 1\right]$ *and* $\overline{s_2^i} = \mathbb{E}\left[S^i | Y \neq 1\right]$. *Then given* $Y = y$, *the first three moments of* S *are*

$$\overline{s_1^1} = \frac{1-\epsilon}{\epsilon}\left(1 - \left(1-\epsilon\right)^{k}\right), \quad (5)$$

$$\overline{s_1^2} = \frac{2\left(1-\epsilon\right)}{\epsilon^2}\left(1 - \left(k\epsilon + 1\right)\left(1-\epsilon\right)^{k}\right) - \overline{s_1^1}, \quad (6)$$

$$\overline{s_1^3} = \frac{6\left(1-\epsilon\right)^3}{\epsilon^3}\left(1 - \left(k\epsilon+1\right)\left(1-\epsilon\right)^{k}\right) + 3\left(1-\epsilon\right)\overline{s_1^2} - \frac{3k}{\epsilon}\left(k+1\right)\left(1-\epsilon\right)^{k+1} + \left(4 - 3\epsilon\right)\overline{s_1^1}, \quad (7)$$

and

$$\overline{s_2^i} = \frac{\overline{s_1^i} - k^i\left(1-\epsilon\right)^{k}}{1 - \left(1-\epsilon\right)^{k}}, \quad (8)$$

for $i = 1, 2, 3$.

Finally, let V_i, $i \leq b - 1$, describe the position of the last received generation preventing delivery in round z_i. The following lemma helps to define the distribution on V_i (see appendix for proof).

Lemma 5. *Let* N *independent processes defined by the transition matrix* $[P]$ *start at the same time, and all processes complete in or before round* z_N *with at least one process completing in round* z_N. *The probability that the* j*th process is the last to complete is defined by the distribution*

$$p_{V_N|Z_N}\left(v_N|z_N\right) = \frac{\left[P_{k0}^{z_N}\right]^{N-v_N-1}\left[P_{k0}^{z_N-1}\right]^{v_N} p_Y\left(z_N\right)}{p_{Z_N}\left(z_N\right)} \quad (9)$$

for $v_N = 0, \ldots N-1$, $j = N - v_N$, $p_Y(y)$ *defined by (2), and* $p_{Z_i}\left(z_i\right)$ *defined by (3). Furthermore, define* $\overline{v_N^i} = \mathbb{E}\left[V_N^i | Z_N\right]$. *Then*

$$\overline{v_N^1} = \frac{\left[P_{k0}^{z_N-1}\right]}{p_Y\left(z_N\right)} - \frac{N\left[P_{k0}^{z_N-1}\right]^{N}}{p_{Z_N}\left(z_N\right)}, \quad (10)$$

and

$$\overline{v_N^2} = \frac{\left[P_{k0}^{z_N-1}\right]}{p_Y\left(z_N\right)} + \frac{2\left[P_{k0}^{z_N-1}\right]^2}{p_Y^2\left(z_N\right)} - \frac{N^2\left[P_{k0}^{z_N-1}\right]^{N}}{p_{Z_N}\left(z_N\right)} - \frac{2N\left[P_{k0}^{z_N-1}\right]^{N+1}}{p_Y\left(z_N\right)p_{Z_N}\left(z_N\right)}. \quad (11)$$

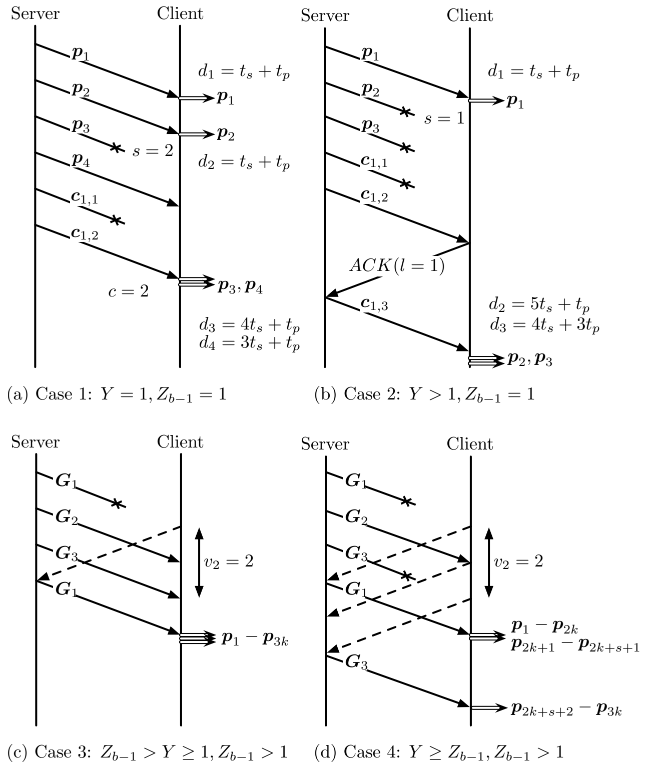
Now that we have the distributions and relevant moments for the random variables Y, Z_i, S, and V_i, we have the tools needed to derive $\mathbb{E}\left[D\right]$ and $\mathbb{E}\left[D^2\right]$.

V. Expected In-Order Delivery Delay

A lower bound on the expected delay, $\mathbb{E}[D]$, can be derived using the law of total expectation:

$$\mathbb{E}[D] \;=\; \mathbb{E}_Y\left[\mathbb{E}_{Z_{b-1}}\left[\mathbb{E}_D\left[D|Y, Z_{b-1}\right]\right]\right]. \quad (12)$$

From (12), there are four distinct cases that must be considered. These are shown in Figure 2. For each case, define $\bar{d}_{Y=y,Z=z} = \mathbb{E}\left[D|Y = y, Z_{b-1} = z\right]$.



(a) Case 1: $Y = 1, Z_{b-1} = 1$

(b) Case 2: $Y > 1, Z_{b-1} = 1$

(c) Case 3: $Z_{b-1} > Y \geq 1, Z_{b-1} > 1$

(d) Case 4: $Y \geq Z_{b-1}, Z_{b-1} > 1$

Figure 2: Example of cases 1-4. The delay d_i of each packet is listed next to the time when it is delivered to the application layer in cases 1 and 2. The number of in-flight generations is $b = 3$ for cases 3 and 4.

A. Case 1: $Y = 1, Z_{b-1} = 1$

The latest generation in transit completes within the first round of transmission and no previously transmitted generations prevent delivery. As a result, all packets received prior to the first loss (i.e., packets $\boldsymbol{p}_1, \ldots \boldsymbol{p}_s$) are immediately delivered. Once a packet loss is observed, packets received after the loss (i.e., packets $\boldsymbol{p}_{s+1}, \ldots, \boldsymbol{p}_k$) are buffered until the entire generation is decoded. An example is given in Figure 2(a) where $n_k = 6$, $k = 4$, the number of packets received prior to the first loss is $s = 2$, and the number of coded packets needed to recover from the two packet losses is $c = 2$. Taking the expectation over all S and all packets within the generation, the mean delay is provided by (16) where $\overline{s_1^1}$ and $\overline{s_1^2}$ are given by Lemma 4; and $\mathbb{E}\left[C|S\right]$ is the expected number of coded packets needed to recover from all packet erasures occurring in the first k packets. When $s < k$, the number of coded packets required is at least one (i.e., $\mathbb{E}\left[C|S\right] \geq 1$) leading to the bound in (15).

B. Case 2: $Y > 1, Z_{b-1} = 1$

All packets $\{\boldsymbol{p}_1, \ldots \boldsymbol{p}_s\}$ are delivered immediately until the first packet loss is observed. Since $Y > 1$, at least one retransmission event is needed to properly decode. Once all k $dofs$ have been received and the generation can be decoded, the remaining packets $\left\{\boldsymbol{p}_{s+1}, \ldots, \boldsymbol{p}_k\right\}$ are delivered in-order. An example is provided in Figure 2(b). The generation cannot be decoded because there are too many packet losses during

$$\bar{d}_{Y=1,Z=1} = \sum_{s=0}^{k-1}\left(\left(t_s+t_p\right)\frac{s}{k}+\frac{1}{k}\sum_{i=0}^{k-s-1}\left(t_p+\big(k-s-i+\mathbb{E}\left[C|S\right]\big)t_s\right)\right)p_{S|Y}\left(s|1\right)+\left(t_s+t_p\right)p_{S|Y}\left(k|1\right) \tag{13}$$

$$= t_p+\frac{t_s}{2k}\left(\overline{s_1^2}-\left(2k-1\right)\overline{s_1^1}+k\left(k+1\right)+2\sum_{s=0}^{k-1}\left(k-s\right)\mathbb{E}\left[C|S\right]p_{S|Y}\left(s|1\right)\right) \tag{14}$$

$$\geq t_p+\frac{t_s}{2k}\left(\overline{s_1^2}-\left(2k-1\right)\overline{s_1^1}+k\left(k+1\right)+2\sum_{s=0}^{k-1}\left(k-s\right)p_{S|Y}\left(s|1\right)\right) \tag{15}$$

$$= \frac{t_s}{2k}\left(\overline{s_1^2}-\left(2k+1\right)\overline{s_1^1}+k\left(k+3\right)\right)+t_p, \tag{16}$$

$$\bar{d}_{Y>1,Z=1} = \frac{1}{k}\sum_{s=0}^{k-1}\left(\left(t_p+t_s\right)s+\sum_{i=0}^{k-s-1}\left(t_p+2\left(y-1\right)t_p+\left(k-s-i+\left(n_k-k\right)\right)t_s\right)\right)p_{S|Y}(s|y) \tag{17}$$

$$= \left(\frac{1}{2k}\overline{s_2^2}-\frac{1}{2k}\left(2n_k-1\right)\overline{s_2^1}+n_k-\frac{1}{2}k+\frac{1}{2}\right)t_s-\left(\frac{2}{k}\left(y-1\right)\overline{s_2^1}-2y+1\right)t_p, \tag{18}$$

$$\bar{d}_{Z>Y\geq 1,Z>1} = \frac{1}{k}\sum_{i=1}^{k}\left(\left(n_k-k+i\right)t_s+t_p+2t_p\left(z-1\right)-\left(\overline{v_{b-1}^1}+1\right)n_kt_s\right) \tag{19}$$

$$= \left(2z-1\right)t_p-\left(\overline{v_{b-1}^1}n_k+\frac{k-1}{2}\right)t_s, \tag{20}$$

$$\bar{d}_{Y\geq Z,Z>1} = \frac{1}{k}\sum_{s=0}^{k-1}\left(\sum_{i=1}^{k}\left(n_k-i+1\right)t_s+\sum_{i=1}^{s}\left(2t_p\left(z-\frac{1}{2}\right)-\left(\overline{v_{b-1}^1}+1\right)n_kt_s\right)+\sum_{j=s+1}^{k}\left(2t_p\Big(y-\frac{1}{2}\Big)\right)\right)p_{S|Y}(s|y) \tag{21}$$

$$= \left(\frac{2\left(z-y\right)}{k}\overline{s_2^1}+2y-1\right)t_p-\left(\frac{n_k}{k}\left(\overline{v_{b-1}^1}+1\right)\overline{s_2^1}-n_k+\frac{1}{2}k-\frac{1}{2}\right)t_s. \tag{22}$$

the first transmission attempt. As a result, one additional dof is retransmitted allowing the client to decode in round two (i.e., $Y = 2$). Taking the expectation over all S and all packets within the generation, the expected delay is provided by (18) where $\overline{s_2^1}$ and $\overline{s_2^2}$ are given by Lemma 4. It is important to note that we do not take into account the time to transmit packets after the first round (see the assumptions in Section III).

C. Case 3: $Z_{b-1} > Y \geq 1, Z_{b-1} > 1$

In this case, generation $\boldsymbol{G}_j$ completes prior to a previously sent generation. As a result, all packets $\left\{\boldsymbol{p}_{(j-1)k+1},\ldots,\boldsymbol{p}_{jk}\right\} \in \boldsymbol{G}_j$ are buffered until all previous generations have been delivered. Once there are no earlier generations preventing in-order delivery, all packets in $\boldsymbol{G}_j$ are immediately delivered. Figure 2(c) provides an example. Consider the delay experienced by packets in $\boldsymbol{G}_3$. While $\boldsymbol{G}_3$ is successfully decoded after the first transmission attempt, generation $\boldsymbol{G}_1$ cannot be decoded forcing all packets in $\boldsymbol{G}_3$ to be buffered until $\boldsymbol{G}_1$ is delivered. Taking the expectation over all packets within the generation and all possible locations of the last unsuccessfully decoded generation, the expected delay is provided by (20) where $\overline{v_{b-1}^1}$ is given by Lemma 5.

D. Case 4: $Y \geq Z_{b-1}, Z_{b-1} > 1$

Finally, this case is a mixture of the last two. The generation $\boldsymbol{G}_j$ completes after all previously transmitted generations, but it requires more than one transmission round to decode. Packets received before the first packet loss are buffered until all previous generations are delivered, and packets received after the first packet loss are buffered until $\boldsymbol{G}_j$ can be decoded. An example is provided in Figure 2(d). Consider the delay of packets in $\boldsymbol{G}_3$. Both $\boldsymbol{G}_1$ and $\boldsymbol{G}_3$ cannot be decoded after the first transmission attempt. After the second transmission attempt, $\boldsymbol{G}_1$ can be decoded allowing packets $\left\{\boldsymbol{p}_{2k+1},\ldots,\boldsymbol{p}_{2k+s+1}\right\} \in \boldsymbol{G}_3$ to be delivered; although packets $\left\{\boldsymbol{p}_{2k+s+2},\ldots,\boldsymbol{p}_{3k}\right\} \in \boldsymbol{G}_3$ must wait to be delivered until after $\boldsymbol{G}_3$ is decoded. Taking the expectation over all S, all packets within the generation, and all possible locations of the last unsuccessfully decoded generation, the expected delay is provided by (22). The expectations $\overline{s_2^1}$ and $\overline{v_{b-1}^1}$ are given by Lemmas 4 and 5 respectively.

Combining the cases above, we obtain the following:

Theorem 6. *The expected in-order delivery delay for the proposed coding scheme is lower bounded by*

$$\mathbb{E}[D] \geq \sum_{z_{b-1}\geq 1}\sum_{y\geq 1}\bar{d}_{Y=y,Z=z_{b-1}}p_Y(y)p_{Z_{b-1}}(z_{b-1}). \tag{23}$$

where $\bar{d}_{Y=y,Z=z_{b-1}}$ is given in equations (16), (18), (20), and (22); and the distributions $p_Y(y)$ and $p_{Z_{b-1}}(z_{b-1})$ are given in equations (2) and (3) respectively.

VI. In-Order Delivery Delay Variance

The second moment of the in-order delivery delay can be determined in a similar manner as the first. Again, we can use

$$\begin{aligned}\overline{d}^2_{Y=1,Z=1} \;&\geq\; t_p^2 + (k+3)\,t_p t_s + \frac{1}{6}\left(2k^2+9k+13\right)t_s^2 - \left(\left(k+3+\frac{7}{6k}\right)t_s^2 + \left(\frac{2k+1}{k}\right)t_p t_s\right)\overline{s_1^1}\\ &\quad+\left(\left(\frac{2k+3}{2k}\right)t_s^2 + \frac{1}{k}t_p t_s\right)\overline{s_1^2} - \frac{1}{3k}t_s^2\overline{s_1^3}, \qquad (25)\\ \overline{d}^2_{Y>1,Z=1} \;&=\; \left(n_k\left(n_k-k+1\right) + \frac{1}{6k}\left(2k^3-3k^2+k+6\right)\right)t_s^2 + \left(2n_k\left(2y-1\right) - 2y\left(k-1\right) + k - 1 + \frac{2}{k}\right)t_p t_s\\ &\quad+\left(\left(2y-1\right)^2 + \frac{1}{k}\right)t_p^2 - \frac{1}{3k}t_s^2\overline{s_2^3} + \frac{1}{2k}\left(\left(2n_k+1\right)t_s^2 + 2\left(2y-1\right)t_p t_s\right)\overline{s_2^2}\\ &\quad-\frac{1}{k}\left(\left(n_k^2+n_k+\frac{1}{6}\right)t_s^2 + \left(2y-1\right)\left(2n_k+1\right)t_p t_s + \left(2y-1\right)^2 t_p^2\right)\overline{s_2^1}, \qquad (26)\\ \overline{d}^2_{Z>Y\geq1,Z>1} \;&=\; \left(n_k^2\overline{v_{b-1}^2} + (k-1)\left(n_k\overline{v_{b-1}^1} + \frac{1}{3}k - \frac{1}{6}\right)\right)t_s^2 - \left(2z-1\right)\left(2n_k\overline{v_{b-1}^1} + k - 1\right)t_p t_s + \left(2z-1\right)^2 t_p^2, \quad (27)\\ \overline{d}^2_{Y\geq Z,Z>1} \;&=\; \left(2y-1\right)\left(\left(2n_k-k+1\right)t_s t_p + \left(2y-1\right)t_p^2\right) + \left(n_k\left(n_k-k+1\right) + \frac{1}{6}\left(2k^2-3k+1\right)\right)t_s^2\\ &\quad+\frac{1}{k}\left(n_k\left(\overline{v_{b-1}^1}+1\right)t_s^2 + 2\left(y-z\right)t_p t_s\right)\overline{s_2^2} + \frac{1}{k}\left(n_k\left(n_k\left(\overline{v_{b-1}^2}-1\right) - \overline{v_{b-1}^1} - 1\right)t_s^2\right.\\ &\quad\left.-2\left(n_k\left(\overline{v_{b-1}^1}\left(2z-1\right) + \left(2y-1\right)\right) + y - z\right)t_p t_s - 4\left(y-z\right)\left(y+z-1\right)t_p^2\right)\overline{s_2^1}. \qquad (28)\end{aligned}$$

the law of total expectation to find the moment:

$$\mathbb{E}[D^2] \;=\; \mathbb{E}_Y\left[\mathbb{E}_{Z_{b-1}}\left[\mathbb{E}_D\left[D^2|Y,Z_{b-1}\right]\right]\right]. \qquad (24)$$

As with the first moment, four distinct cases exist that must be dealt with separately. For each of the cases in Section V, define $\overline{d}^2_{Y=y,Z=z} = \mathbb{E}\left[D^2|Y=y,Z_{b-1}=z\right]$. While we omit the initial step in the derivation of each case, $\overline{d}^2_{Y,Z}$ can be determined using the same assumptions as above and is shown in equations (25)-(28). The expectations $\overline{s_i^j}$ and $\overline{v_{b-1}^i}$ for $i = \{1,2\}$ and $j = \{1,2,3\}$ are provided in Lemmas 4 and 5 respectively; and the bound in (25) follows from assuming that the number of coded packets needed to decode a generation is limited to one packet (i.e., $c = 1$ for all s).

Combining equations (24) through (28):

Theorem 7. *The second moment of the in-order delivery delay for the proposed coding scheme is lower bounded by*

$$\mathbb{E}[D^2] \geq \sum_{z_{b-1}\geq1}\sum_{y\geq1}\overline{d}^2_{Y=y,Z=z_{b-1}}p_Y(y)p_{Z_{b-1}}(z_{b-1}), \qquad (29)$$

where $\overline{d}^2_{Y=y,Z=z_{b-1}}$ is given in equations (25) through (28); and the distributions $p_Y(y)$ and $p_{Z_{b-1}}(z_{b-1})$ are given in (2) and (3) respectively. Furthermore, the in-order delay variance is $\sigma_D^2 = \mathbb{E}[D^2] - \mathbb{E}[D]^2$ where $\mathbb{E}[D]$ is given in (23).

VII. Efficiency

The above results show adding redundancy into a packet stream decreases the in-order delivery delay. However, doing so comes with a cost. We characterize this cost in terms of efficiency. Before defining the efficiency, let M_i, $i \in [0,k]$, be the number of packets received at the sink as a result of transmitting a generation of size i. Alternatively, M_i is the total number of packets received by the sink for any path starting in state i and ending in state 0 of the Markov chain defined in Section III. Furthermore, define M_{ij} to be the number of packets received by the sink as a result of a single transition from state i to state j (i.e., $i \to j$). M_{ij} is deterministic (e.g., $m_{ij} = i - j$) when $i,j \geq 1$ and $i \geq j$. For any transition $i \to 0$, $i \geq 1$, $m_{i0} \in [i, n_i]$ has probability

$$\begin{aligned}p_{M_{i0}}(m_{i0}) \;&=\; \frac{B\left(n_i,m_{i0},1-\epsilon\right)}{\sum_{j=i}^{n_i}B\left(n_i,j,1-\epsilon\right)} \qquad (30)\\ &=\; {}^{1}\!/_{a_{i0}}B\left(n_i,m_{i0},1-\epsilon\right). \qquad (31)\end{aligned}$$

Therefore, the expected number of packets received by the sink is

$$\mathbb{E}[M_{ij}] = \begin{cases} i-j & \text{for } i,j\geq1, i\geq j\\ \frac{1}{a_{i0}}\sum_{x=i}^{n_i}x\cdot B\left(n_i,x,1-\epsilon\right) & \text{for } i\geq1, j=0.\end{cases} \qquad (32)$$

Given $\mathbb{E}[M_{ij}]\ \forall i,j$, the total number of packets received by the sink when transmitting a generation of size i is

$$\mathbb{E}\left[M_i\right] = \frac{1}{1-a_{ii}}\left(\sum_{j=0}^{i-1}\left(\mathbb{E}\left[M_{ij}\right] + \mathbb{E}\left[M_j\right]\right)a_{ij}\right) \qquad (33)$$

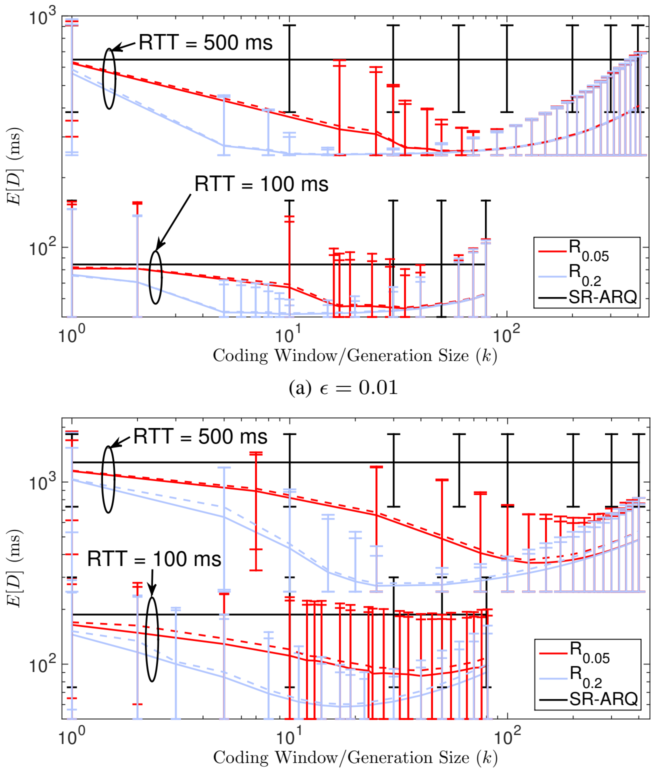
where $\mathbb{E}[M_0] = 0$. This leads us to the following theorem.

Theorem 8. *The efficiency η_k, defined as the ratio between the number of information packets or $dofs$ within each generation of size k and the expected number of packets received by the sink, is*

$$\eta_k \triangleq \frac{k}{\mathbb{E}\left[M_k\right]}. \qquad (34)$$

VIII. Numerical Results

Unfortunately, the complexity of the process prevents us from determining a closed form expression for the in-order delay. However, we now provide numerical results that show



(a) $\epsilon = 0.01$

(b) $\epsilon = 0.1$

Figure 3: The in-order delay for two erasure rates ϵ as a function of the generation size k on a 10 Mbps link. The error bars show $2\sigma_D$ above and below the mean and $R_x = {}^{(1+x)}/_{(1-\epsilon)}$. The analytical and simulated results are represented using solid and dotted lines respectively. Note the log scale of both the x-axis and y-axis.

(23) and (29) are fairly tight for most cases, which allows us to use it as a fairly accurate estimate of $\mathbb{E}[D]$ and $\mathbb{E}\left[D^2\right]$.

Before proceeding, several items need to be noted. First, the terms where $p_Y(y)\, p_{Z_{b-1}}(z_{b-1}) < 10^{-6}$ when calculating $\mathbb{E}[D]$ and $\mathbb{E}\left[D^2\right]$ are not considered since they have little effect on the overall calculation. Second, the analytical curves are sampled at local maxima. As the code generation size increases, the number of in-flight generations, $b = \lceil {}^{BDP}/_{k} \rceil$, incrementally decreases. Upon each decrease in b, a discontinuity occurs that causes an artificial decrease in $\mathbb{E}[D]$ that becomes less noticeable as k increases towards the next decrease in b. This transient behavior in the analysis is more prominent when $R \approx {}^{1}/_{1-\epsilon}$ and less so when $R \gg {}^{1}/_{1-\epsilon}$. Regardless, the figures show an approximation with this behavior removed. Third, we note that Rk may not be an integer. To overcome this issue, $\lceil Rk \rceil - k$ and $\lfloor Rk \rfloor - k$ coded packets are generated and transmitted with probability $Rk - \lfloor Rk \rfloor$ and $\lceil Rk \rceil - Rk$ respectively. Finally, comparisons with the schemes shown in Figure 1 are not provided due to considerably worse performance or lack of specific implementation details. Instead, we only compare our results with those of SR-ARQ, which we assume is a valid baseline.

A. Coding Window Size and Redundancy Selection

Results for four different networks/links are shown in Figure 3. The simulation was developed in Matlab using a model similar to that presented in Section III, although several of the assumptions are relaxed. The time it takes to retransmit coded packets after feedback is received is taken into account. Furthermore, the number of generations preventing delivery is not limited to a single BDP of packets, which increases the probability of head-of-line blocking. Both of these relaxations effectively increases the delay experienced by a packet. Finally, the figure shows the delay of an idealized version of SR-ARQ where we assume infinite buffer sizes. This is intended to provide a baseline with other proposed schemes such as those shown in Figure 1.

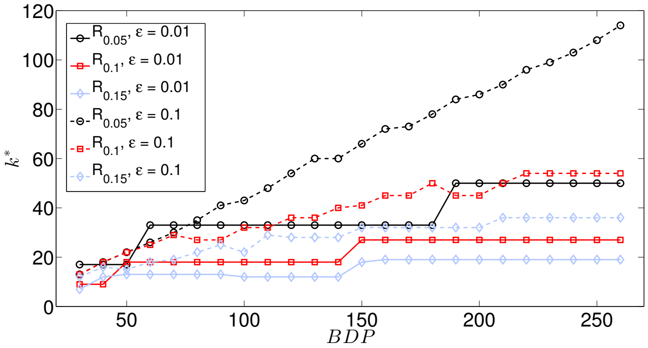


Figure 4: k^* as a function of the BDP.

Figure 3 illustrates that adding redundancy and/or choosing the correct coding window/generation size can have major implications on the in-order delay. Not only does choosing correctly reduce the delay, but doing so can also reduce the jitter. However, it is apparent that the proper selection of k for a given R is critical for minimizing $\mathbb{E}[D]$ and $\mathbb{E}\left[D^2\right]$. In fact, Figure 3 indicates that adding redundancy and choosing a moderately sized generation is needed in most cases to ensure both are minimized.

The shape of the curves in the figure also indicate that there are two major contributors to the in-order delay that need to be balanced. Let k^* be the generation size where $\mathbb{E}[D]$ is minimized for a given ϵ and R, i.e.,

$$k^* = \arg\min_k \mathbb{E}[D]. \tag{35}$$

To the left of k^*, the delay is dominated by head-of-line blocking and resequencing delay created by previous generations. To the right of k^*, the delay is dominated by the time it takes to receive enough $dofs$ to decode the generation. While there are gains in efficiency for $k > k^*$, the benefits are negligible for most time-sensitive applications. As a result, we show k^* for a given ϵ and R as a function of the BDP in Figure 4 and make three observations. First, the coding window size k^* increases with ϵ, which is opposite of what we would expect from a typical erasure code [25]. In the case of small ϵ, it is better to try and quickly correct only some of the packet losses occurring within a generation using the initially transmitted coded packets while relying heavily on feedback to overcome any decoding errors. In the case of large ϵ, a large generation size is better where the majority of packet losses occurring within a generation are corrected using the initially transmitted coded packets and feedback is

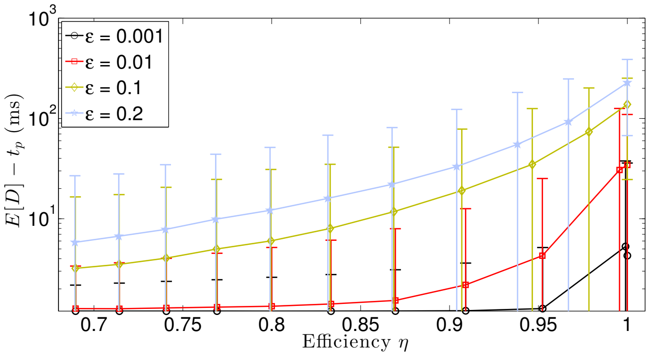


Figure 5: Rate-delay trade-off for a 10 Mbps link with a RTT of 100 ms. The error bars represent $2\sigma_D$ above and below the mean, and the delay for ARQ is shown for $\eta = 1$. Note the log scale of the y-axis.

relied upon to help overcome the rare decoding error. Second, increasing R decreases k^*. This due to the receiver's increased ability to decode a generation without having to wait for retransmissions. Third, k^* is not very sensitive to the BDP (in most cases) enabling increased flexibility during system design and implementation.

Before proceeding, it is important to note that a certain level of redundancy is needed to see benefits. Each curve shows results for $R > {}^{1}/_{1-\epsilon}$. For $R \leq {}^{1}/_{1-\epsilon}$, it is possible to see in-order delays and jitter worse than the idealized ARQ scheme. Consider an example where a packet loss is observed near the beginning of a generation that cannot be decoded after the first transmission attempt. Since feedback is not sent/acted upon until the end of the generation, the extra time waiting for feedback can induce larger delays than what would have occurred under a simple ARQ scheme. We can reduce this time by reacting to feedback before the end of a generation; but it is still extremely important to ensure that the choice of k and R will decrease the probability of a decoding failure and provide improved delay performance.

B. Rate-Delay Trade-Off

While transport layer coding can help meet strict delay constraints, the decreased delay comes at the cost of throughput, or efficiency. Let $\mathbb{E}[D^*]$, σ_D^*, and η^* be the expected in-order delay, the standard deviation, and the expected efficiency respectively that corresponds to k^* defined in eq. (35). The rate-delay trade-off is shown by plotting $\mathbb{E}[D^*]$ as a function of η^* in Figure 5. The expected SR-ARQ delay (i.e., the data point for $\eta = 1$) is also plotted for each packet erasure rate as a reference.

The figure shows that an initial increase in R (or a decrease in η) has the biggest effect on $\mathbb{E}[D]$. In fact, the majority of the decrease is observed at the cost of just a few percent (2-5%) of the available network capacity when ϵ is small. As R is increased further, the primary benefit presents itself as a reduction in the jitter (or $\mathbb{E}\left[D^2\right]$). Furthermore, the figure shows that even for high packet erasure rates (e.g., 20%), strict delay constraints can be met as long as the user is willing to sacrifice throughput.

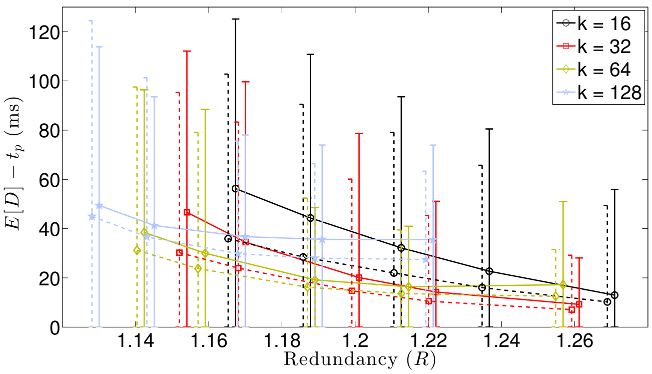


Figure 6: Experimental (solid lines) and analytical (dotted lines) results for various k over a 25 Mbps link with $RTT = 60$ ms and $\epsilon = 0.1$.

C. Real-World Comparison

We finally compare the analysis with experimentally obtained results in Figure 6 and show that our analysis provides a reasonable approximation to real-world protocols. The experiments were conducted using Coded TCP (CTCP) over an emulated network similar to the one used in [7]. The only difference between our setup and theirs was that we fixed CTCP's congestion control window size ($cwnd$) to be equal to the BDP of the network in order to eliminate the affects of fluctuating $cwnd$ sizes.

There are several contributing factors for the differences between the experimental and analytical results shown in the figure. First, the analytical model approximates the algorithm used in CTCP. Where we assume feedback is only acted upon at the end of a generation, CTCP proactively acts upon feedback and does not wait until the end of a generation to determine if retransmissions are required. Second, the experiments include additional processing time needed to accomplish tasks such as coding and decoding, while the analysis does not. Finally, the assumptions made in Sections III and V effectively lower bounds $\mathbb{E}[D]$ and $\mathbb{E}\left[D^2\right]$. The bounds are fairly tight for large k and R, but they can be very loose for either small k or small R. An example of this is evident in Figure 6 for $k = 16$ and $R = 1.65$ where there is a significant difference between the experimental and analytical results. However, simulation results suggest neither small k nor small R result in k^* therefore making a tight bound in these regimes less important. For all other choices of k and R, the analysis can provide a fairly good estimate of the in-order delay and can be used to help inform system decisions.

IX. Conclusion

In this paper, we addressed the use of transport layer coding to improve application layer performance. A coding algorithm and an analysis of the in-order delivery delay's first two moments were presented, in addition to numerical results addressing when and how much redundancy should be added to a packet stream to meet a user's delay constraints. These results showed that the coding window size that minimizes the expected in-order delay is fairly insensitive to the network's BDP for some cases. Finally, we compared our analysis with the measured delay of an implemented transport protocol, CTCP. While our analysis and the behavior of CTCP do not provide a one-to-one comparison, we illustrated how our work can be used to help inform system decisions when attempting to minimize delay.

Appendix

Proof: (Lemma 3) Let $f(z) = 1 - \sum_{j=1}^{z-1} p_Y(j)$. The probability of N independent processes completing in less than or equal to z rounds with at least one process completing in round z is:

$$\begin{aligned}
\Pr\{Z = z\} &= \sum_{i=1}^{N} B(N, i, f(z)) \left(\frac{p_Y(z)}{f(z)}\right)^i && (36)\\
&= \sum_{i=1}^{N} \binom{N}{i} (p_Y(z))^i \left[P_{k0}^{z-1}\right]^{N-i} && (37)\\
&= \frac{\left[P_{k0}^{z-1}\right]^{N+1} \left(\left[P_{k0}^{z}\right]^{N+1} - \left[P_{k0}^{z-1}\right]^{N+1}\right)}{\left[P_{k0}^{z-1}\right]^{N+1} \left[P_{k0}^{z}\right]} \\
&\quad - \frac{\left[P_{k0}^{z-1}\right]^{2N+1} \left(\left[P_{k0}^{z}\right] - \left[P_{k0}^{z-1}\right]\right)}{\left[P_{k0}^{z-1}\right]^{N+1} \left[P_{k0}^{z}\right]} && (38)\\
&= \left[P_{k0}^{z}\right]^N - \left[P_{k0}^{z-1}\right]^N . && (39)
\end{aligned}$$

■

Proof: (Lemma 4) Define the moment generating function of S when $Y = 1$ to be

$$\begin{aligned}
M_{S|Y}(t) &= \mathbb{E}[e^{tS} | Y = 1] && (40)\\
&= \frac{\epsilon \left(1 - e^{kt}(1-\epsilon)^k\right)}{1 - e^t + \epsilon e^t} + e^{kt}(1-\epsilon)^k . && (41)
\end{aligned}$$

The first, second, and third moments of S when $Y = 1$ are then ${}^{\delta}/_{\delta t} M_{S|Y}(0)$, ${}^{\delta^2}/_{\delta t^2} M_{S|Y}(0)$, and ${}^{\delta^3}/_{\delta t^3} M_{S|Y}(0)$ respectively.

For $Y \neq 1$, scale the above expectations by subtracting the term $k^i (1-\epsilon)^k$ from each of the moments above and dividing by $1 - (1-\epsilon)^k$. ■

Proof: (Lemma 5) Let $\beta_{z_N} = {}^{p_Y(z_N)}/_{[P_{k0}^{z_N}]}$, be the probability of a generation finishing in round z_N given all of the N generations have completed transmission in or before round z_N. The distribution on $V_N \in [0, N-1]$ is

$$\begin{aligned}
p_{V_N|Z_N}(v_N|z_N) &= \frac{\beta_{z_N}(1-\beta_{z_N})^{v_N}}{\sum_{j=0}^{N-1} \beta_{z_N}(1-\beta_{z_N})^j} && (42)\\
&= \frac{\beta_{z_N}(1-\beta_{z_N})^{v_N+1}}{1-(1-\beta_{z_N})^N} && (43)\\
&= \frac{\left[P_{k0}^{z_N}\right]^{N-v_N-1} \left[P_{k0}^{z_N-1}\right]^{v_N} p_Y(z_N)}{p_{Z_N}(z_N)}. && (44)
\end{aligned}$$

Define the moment generating function of V_N given Z_N to be

$$\begin{aligned}
M_{V_N|Z_N}(t) &= \mathbb{E}[e^{tV_N} | Z_N = z_N] \\
&= \frac{\left(\left[P_{k0}^{z_N}\right]^N - \left[P_{k0}^{z_N-1}\right]^N e^{Nt}\right) p_Y(z_N)}{\left(\left[P_{k0}^{z_N}\right] - \left[P_{k0}^{z_N-1}\right] e^t\right) p_{Z_N}(z_N)}. && (45)
\end{aligned}$$

The first and second moments of V_N given Z_N are ${}^{\delta}/_{\delta t} M_{V_N|Z_N}(0)$ and ${}^{\delta^2}/_{\delta t^2} M_{V_N|Z_N}(0)$ respectively. ■

References

[1] Sandvine, "Global Internet Phenomena." Online, May 2014.

[2] Y. Xia and D. Tse, "Analysis on Packet Resequencing for Reliable Network Protocols," in *INFOCOM*, vol. 2, pp. 990–1000, Mar. 2003.

[3] J. K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proc. of the IEEE*, vol. 99, pp. 490–512, Mar. 2011.

[4] V. Subramanian, S. Kalyanaraman, and K. K. Ramakrishnan, "Hybrid Packet FEC and Retransmission-Based Erasure Recovery Mechanisms for Lossy Networks: Analysis and Design," in *COMSWARE*, 2007.

[5] O. Tickoo, V. Subraman, S. Kalyanaraman, and K. K. Ramakrishnan, "LT-TCP: End-to-End Framework to Improve TCP Performance Over Networks with Lossy Channels," in *IWQoS*, pp. 81–93, 2005.

[6] B. Ganguly, B. Holzbauer, K. Kar, and K. Battle, "Loss-Tolerant TCP (LT-TCP): Implementation and Experimental Evaluation," in *MILCOM*, 2012.

[7] M. Kim, J. Cloud, A. ParandehGheibi, L. Urbina, K. Fouli, D. J. Leith, and M. Médard, "Congestion Control for Coded Transport Layers," in *ICC*, June 2014.

[8] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. on Info. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[9] A. Heidarzadeh, *Design and Analysis of Random Linear Network Coding Schemes: Dense Codes, Chunked Codes and Overlapped Chunked Codes*. Ph.D. Thesis, Carleton University, Ottawa, Canada, Dec. 2012.

[10] D. Lucani, M. Médard, and M. Stojanovic, "Broadcasting in Time-Division Duplexing: A Random Linear Network Coding Approach," in *NetCod*, pp. 62–67, June 2009.

[11] D. Lucani, M. Médard, and M. Stojanovic, "Online Network Coding for Time-Division Duplexing," in *GLOBECOM*, Dec. 2010.

[12] D. Lucani, M. Stojanovic, and M. Médard, "Random Linear Network Coding For Time Division Duplexing: When To Stop Talking And Start Listening," in *INFOCOM*, pp. 1800–1808, Apr. 2009.

[13] T. Dikaliotis, A. Dimakis, T. Ho, and M. Effros, "On the Delay of Network Coding Over Line Networks," in *ISIT*, June 2009.

[14] M. Nistor, R. Costa, T. Vinhoza, and J. Barros, "Non-Asymptotic Analysis of Network Coding Delay," in *NetCod*, June 2010.

[15] E. Drinea, C. Fragouli, and L. Keller, "Delay with Network Coding and Feedback," in *ISIT*, pp. 844–848, June 2009.

[16] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On Delay Performance Gains From Network Coding," in *CISS*, pp. 864–870, Mar. 2006.

[17] B. Swapna, A. Eryilmaz, and N. Shroff, "Throughput-Delay Analysis of Random Linear Network Coding for Wireless Broadcasting," *IEEE Trans. on Information Theory*, vol. 59, pp. 6328–6341, Oct. 2013.

[18] H. Yao, Y. Kochman, and G. W. Wornell, "A Multi-Burst Transmission Strategy for Streaming Over Blockage Channels with Long Feedback Delay," *IEEE JSAC*, vol. 29, pp. 2033–2043, Dec. 2011.

[19] M. Nistor, J. Barros, F. Vieira, T. Vinhoza, and J. Widmer, "Network Coding Delay: A Brute-Force Analysis," in *ITA*, Jan. 2010.

[20] J. Sundararajan, P. Sadeghi, and M. Médard, "A Feedback-Based Adaptive Broadcast Coding Scheme for Reducing In-Order Delivery Delay," in *NetCod*, June 2009.

[21] W. Zeng, C. Ng, and M. Médard, "Joint Coding and Scheduling Optimization in Wireless Systems with Varying Delay Sensitivities," in *SECON*, pp. 416–424, June 2012.

[22] G. Joshi, Y. Kochman, and G. W. Wornell, "On Playback Delay in Streaming Communication," in *ISIT*, pp. 2856–2860, July 2012.

[23] G. Joshi, Y. Kochman, and G. Wornell, "Effect of Block-Wise Feedback on the Throughput-Delay Trade-Off in Streaming," in *INFOCOM Workshop on Contemporary Video*, Apr. 2014.

[24] M. Tömösközi, F. H. Fitzek, F. H. Fitzek, D. E. Lucani, M. V. Pedersen, and P. Seeling, "On the Delay Characteristics for Point-to-Point Links using Random Linear Network Coding with On-the-Fly Coding Capabilities," in *European Wireless 2014*, May 2014.

[25] R. Koetter and F. Kschischang, "Coding for Errors and Erasures in Random Network Coding," *IEEE Trans. on Information Theory*, vol. 54, pp. 3579–3591, Aug. 2008.