The Effect of Block-wise Feedback on the Throughput-Delay Trade-off in Streaming

Gauri Joshi
EECS Dept., MIT
Cambridge, MA 02139, USA
Email: gauri@mit.edu

Yuval Kochman
School of CSE, HUJI
Jerusalem, Israel
Email: yuvalko@cs.huji.ac.il

Gregory W. Wornell
EECS Dept., MIT
Cambridge, MA 02139, USA
Email: gww@mit.edu

***Abstract*—Unlike traditional file transfer where only total delay matters, streaming applications impose delay constraints on each packet and require them to be *in order*. To achieve fast in-order packet decoding, we have to compromise on the throughput. We study this trade-off between throughput and in-order decoding delay, and in particular how it is affected by the frequency of block-wise feedback, whereby the source receives full channel state feedback at periodic intervals. Our analysis shows that for the same throughput, having more frequent feedback significantly reduces the in-order decoding delay. For any given block-wise feedback delay, we present a spectrum of coding schemes that span different throughput-delay tradeoffs. One can choose an appropriate coding scheme from these, depending upon the delay-sensitivity and bandwidth limitations of the application.**

I. Introduction

A. Motivation

A recent report [1] shows that 62% of the Internet traffic in North America comes from real-time streaming applications. Unlike traditional file transfer where only total delay matters, streaming imposes delay constraints on each individual packet. Further, many applications require in-order playback of packets at the receiver. Packets received out of order are buffered until the missing packets in the sequence are successfully decoded. In audio and video applications some packets can be dropped without affecting the streaming quality. However, other applications such as remote desktop, and collaborative tools such as Dropbox and Google Docs have strict order constraints on packets, where packets represent instructions that need to be executed in order at the receiver.

Thus, there is a need to develop transmission schemes that can ensure in-order packet delivery to the user, with efficient use of available bandwidth. To ensure that packets are decoded in order, the transmission scheme must give higher priority to older packets that were delayed, or received in error. However, repeating old packets instead of transmitting new packets results in a loss in the overall rate of packet delivery to the user, i.e., the throughput. Thus there is a fundamental trade-off between throughput and in-order decoding delay.

The throughput loss incurred to achieve low in-order decoding delay can be significantly reduced if the source receives feedback about packet losses, and thus can adapt its future transmission strategy to strike the right balance between old and new packets. We study this interplay between feedback and the throughput-delay trade-off.

B. Previous Work

When there is immediate and error-free feedback, it is well understood that a simple Automatic-repeat-request (ARQ) scheme is both throughput and delay optimal. But only a few papers in literature have analyzed streaming codes with delayed or no feedback. Fountain codes [2] are capacity-achieving erasure codes, but they are not suitable for streaming because the decoding delay is proportional to the size of the data. Streaming codes without feedback for constrained channels such as adversarial and cyclic burst erasure channels were first proposed in [3], and also extensively explored in [4], [5]. The thesis [3] also proposed codes for more general erasure models and analyzed their decoding delay. Decoding delay was also studied in [6], [7] in a multicast scenario with immediate feedback to the source.

However, decoding delay does not capture *in order* packet delivery, which is required for streaming applications. This aspect is captured in the delay metrics in [8] and [9], which consider that packets are played in-order at the receiver. The authors in [8] analyze the throughput-delay trade-off for uncoded packet transmission over a channel with long feedback delay. In [9] we propose coding schemes that minimize playback delay in point-to-point streaming for the no feedback and immediate feedback cases. However, the case of block-wise feedback to the source remains to be explored.

C. Our Contributions

In this paper we consider this unexplored problem of how to effectively utilize feedback received by the source to ensure in-order packet delivery to the user. We consider block-wise feedback, where the source receives information about past channel states at periodic intervals. In contrast to playback delay considered in [8] and [9], we propose a more versatile delay metric called the in-order decoding exponent. This metric captures the burstiness in the in-order decoding of packet for applications that require packets in-order, but do not necessarily play them at a constant rate.

In the limiting case of immediate feedback, we can use ARQ and achieve the optimal throughput and delay simultaneously. But as the feedback delay increases, we have to compromise on at least one of these metrics. Our analysis shows that for the same throughput, having more frequent block-wise feedback significantly improves the in-order decoding exponent. This

978-1-4799-3088-3/14/$31.00 ©2014 IEEE

conclusion is reminiscent of [10] which studied the effect of feedback on error exponents. We present a spectrum of coding schemes spanning the throughput-delay trade-off, and prove that they give the best trade-off within a broad class of schemes for the no feedback, and small feedback delay cases.

II. Problem Setup

A. System Model

We consider a point-to-point packet streaming scenario where the source produces a infinitely large stream of packets s_i, for $i \in \mathbb{N}$. In each slot n, the encoder creates a coded packet $y_n = f(s_1, s_2 ..s_n)$, a causal function of source packets, and transmits it over the channel. The source packets s_n and coded packets y_n have the same alphabet size. Assume that the encoding function f is known to the receiver. For example, if y_n is a linear combination of the source packets with respect to some field, the coefficients are included in the transmitted packet so that the receiver can use them to decode the source packets from the coded combination. Without loss of generality, we can assume that y_n is a linear combination of the source packets.

We consider an i.i.d. packet erasure channel where every transmitted packet is correctly received with probability p, and otherwise received in error and discarded. An erasure channel is a good model when encoded packets have a set of checksum bits that can be used to verify with high probability whether the received packet is error-free.

The receiver application requires the stream of source packets to be *in order*. Packets received out of order are buffered until the missing packets in the sequence are decoded. Due to this in-order property, the transmitter can stop including s_k in coded packets when it knows that the receiver can decode s_k once all s_i for $i < k$ are decoded. We refer to such packets as "seen" packets. The notion of "seen" is defined formally as follows.

Definition 1 (Seen Packets)**.** *The transmitter marks packet s_k as "seen" when it knows that a coded combination that only includes s_k, and packets s_i for $1 \leq i < k$, is received successfully.*

We consider that the source receives block-wise feedback about channel erasures after every d slots. Thus, before transmitting in slot $kd + 1$, for all integers $k \geq 1$, the source knows about the erasures in slots $(k-1)d + 1$ to kd. It can use this information to adapt its transmission strategy in slot $kd + 1$. Block-wise feedback can be used to model a half-duplex communication channel where after every d slots of packet transmission, the channel is reserved for the receiver to send feedback about the status of decoding.[1]

B. Throughput and Delay Metrics

We consider two metrics to measure the quality of streaming: the throughput τ and in-order decoding exponent λ. The throughput is the rate at which "innovative" coded packets are received. A coded packet is said to be "innovative" if it is linear independent with respect to the coded packets received until then. The bandwidth required is proportional to $1/\tau$. The throughput captures the overall rate at which packets go through the channel, irrespective of the order. The *in-order* decoding aspect is captured by a metric called the in-order decoding exponent λ which is defined as follows.

Definition 2 (In-order Decoding Exponent)**.** *Let T be the time between two successive instants of decoding one or more packets in-order. Then the in-order decoding exponent λ is*

$$\lambda \triangleq -\lim_{n\to\infty} \frac{\log \Pr(T > n)}{n} \text{ when the limit exists.} \quad (1)$$

The relation (1) can equivalently be expressed as $\Pr(T > n) \doteq e^{-n\lambda}$ using the notation in [11, Page 63] where the $\doteq$ stands for asymptotic equality. The in-order decoding exponent captures the burstiness in packet decoding. For example, if the application plays one in-order packet in every slot, and there are b packets in the receiver buffer, then the probability of an interruption in playback is proportional to $e^{-\lambda b}$. In [9], the expected playback delay with constant rate in-order playback is shown to be asymptotically equal to $(1/\lambda) \cdot \log n$.

We analyze how the trade-off between τ and λ is affected by the block-wise feedback delay d. We first consider the extreme cases of immediate feedback $(d = 1)$ and no feedback $(d = \infty)$ in Section III and Section IV respectively. This gives us insights into analyzing the trade-off for general d in Section V.

III. Immediate Feedback

In the immediate feedback $(d = 1)$ case, the source has complete knowledge of past erasures before transmitting each packet. We can show that a simple automatic-repeat-request (ARQ) scheme is optimal in both τ and λ. In this scheme, the source transmits the lowest index unseen packet, and repeats it until the packet successfully goes through the channel.

Since a new packet is received in every successful slot, the throughput $\tau = p$, the success probability of the erasure channel. The ARQ scheme is throughput-optimal because the throughput $\tau = p$ is equal to the information-theoretic capacity of the erasure channel [11]. Moreover, it also gives the optimal the in-order decoding exponent λ because one in-order packet is decoded in every successful slot. To find λ, first observe that the tail distribution of the time T, the interval between successive in-order decoding instants is,

$$\Pr(T > n) = (1-p)^n \quad (2)$$

Substituting this in Definition 2 we get the exponent $\lambda = -\log(1-p)$. Thus, the trade-off for the immediate feedback case is $(\tau, \lambda) = (p, -\log(1-p))$.

From this analysis of the immediate feedback case we can find limits on the range of achievable (τ, λ) for any feedback delay d. Since a scheme with immediate feedback can always simulate one with delayed feedback, the throughput and delay metrics (τ, λ) achievable for any feedback delay d must lie in the region $0 \leq \tau \leq p$, and $0 \leq \lambda \leq -\log(1-p)$.

[1] In addition to its role developed in this paper, such feedback can also be used to estimate p, the probability of success of the erasure channel. For our analysis, we assume that p has been reliably estimated already.

IV. No Feedback

Now we consider the other extreme case ($d = \infty$), corresponding to when there is no feedback to the source. We propose a coding scheme that gives the best (τ, λ) trade-off among the class of full-rank codes, defined as follows.

Definition 3 (Full-rank Codes)**.** *In slot n we transmit a linear combination of all packets s_1 to $s_{V[n]}$, where the coefficients are chosen from a large enough field such that the coded combinations are independent with high probability. We refer to $V[n]$ as the transmit index in slot n.*

Conjecture 1. *Given transmit index $V[n]$, there is no loss of generality in including all packets s_1 to $s_{V[n]}$.*

We believe this conjecture is true because the packets are required in-order at the receiver. Thus, every packet s_j, $j < V[n]$ is required before packet $s_{V[n]}$ and there is no advantage in excluding s_j from the combination. Hence we believe that there is no loss of generality in restricting our attention to full-rank codes. A direct approach to verifying this conjecture would involve checking all possible channel erasure patterns.

Theorem 1. *The optimal throughput-delay trade-off among full-rank codes is $(\tau, \lambda) = (r, D(r\|p))$ for all $0 \le r \le p$. It is achieved by the coding scheme with $V[n] = \lceil rn \rceil$ for all n.*

The term $D(r\|p)$ is the binary information divergence function, which is defined for $0 < p, r < 1$ as

$$D(r\|p) = r \log \frac{r}{p} + (1-r) \log \frac{1-r}{1-p}, \tag{3}$$

where $0 \log 0$ is assumed to be 0. As $r \to 0$, $D(r\|p)$ converges to $-\log(1-p)$, which is the best possible λ as given in Section III.

Proof: We first show that the scheme with transmit index $V[n] = \lceil rn \rceil$ in time slot n achieves the trade-off $(\tau, \lambda) = (r, D(r\|p))$. Then we prove the converse by showing that no other full-rank scheme gives a better trade-off.

Achievability Proof: Consider the scheme with transmit index $V[n] = \lceil rn \rceil$, where r represents the rate of adding new packets to the transmitted stream. The rate of adding packets is below the capacity of the erasure channel. Thus it is easy to see that the throughput $\tau = r$. Let $E[n]$ be the number of combinations, or equations received until time n. It follows the binomial distribution with parameter p. All packets $s_1 \cdots s_{V[n]}$ are decoded when $E[n] \ge V[n]$. Define the event $G_n = \{E[j] < V[j] \text{ for all } 1 \le j \le n\}$, that there is no packet decoding until slot n. The tail distribution of time T between successive in-order decoding instants is,

$$\begin{aligned}\Pr(T > n) &= \sum_{k=0}^{\lceil nr \rceil - 1} \Pr(E[n] = k) \Pr(G_n | E[n] = k), \\ &= \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k} \Pr(G_n | E[n] = k),\end{aligned}$$

where $\Pr(G_n|E[n] = k) = 1 - k/n$ as given by the Generalized Ballot theorem in [12, Chapter 4]. Hence it is sub-exponential and does not affect the exponent of $\Pr(T > n)$ and we have

$$\begin{aligned}\Pr(T > n) &\doteq \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k}, &(4)\\ &\doteq \binom{n}{\lceil nr \rceil - 1} p^{\lceil nr \rceil - 1} (1-p)^{n - \lceil nr \rceil + 1}, &(5)\\ &\doteq e^{-nD(r\|p)}, &(6)\end{aligned}$$

where in (4) we take the asymptotic equality $\doteq$ to find the exponent of $\Pr(T > n)$, and remove the $\Pr(G_n|E[n] = k)$ term because it is sub-exponential. In (5), we only retain the $k = \lceil nr \rceil - 1$ term from the summation because for $r \le p$, that term asymptotically dominates other terms. Finally, we use the Stirlings approximation $\binom{n}{k} \approx e^{nH(k/n)}$ to obtain (6).

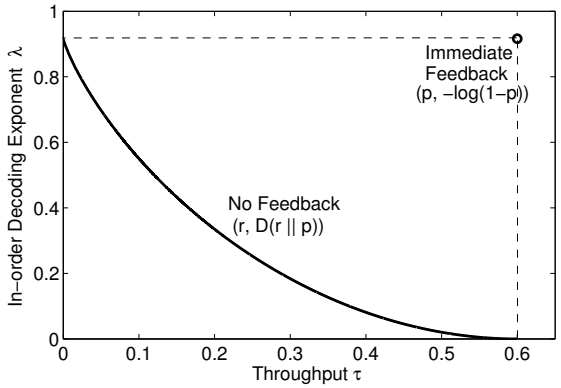

Fig. 1. The trade-off between in-order decoding exponent λ and throughput τ with success probability $p = 0.6$ for the immediate feedback ($d = 1$) and no feedback ($d = \infty$) cases.

Converse Proof: First we show that the transmit index $V[n]$ of the optimal full-rank scheme should be non-decreasing in n. Given any scheme, we can permute the order of transmitting the coded packets such that $V[n]$ is non-decreasing in n. This does not affect the throughput τ, but it can improve the in-order decoding exponent λ because decoding can occur sooner when the initial coded packets include fewer source packets.

We now show that it is optimal to have $V[n] = \lceil rn \rceil$, where we add new packets to the transmitted stream at a constant rate r. Suppose a full-rank scheme uses rate r_i for n_i slots for all $1 \le i \le L$, such that $\sum_{i=0}^{L} n_i = n$ and $\sum_{i=1}^{L} n_i r_i = nr$. Then, the tail distribution of time T between successive in-order decoding instants is,

$$\Pr(T > n) = \sum_{k=0}^{\lceil \sum_{i=1}^{L} n_i r_i \rceil - 1} \Pr(E[n] = k) \Pr(G_n | E[n] = k), \tag{7}$$

$$\doteq \sum_{k=0}^{\lceil nr \rceil - 1} \binom{n}{k} p^k (1-p)^{n-k}, \tag{8}$$

$$\doteq e^{-nD(r\|p)}. \tag{9}$$

Varying the rate of adding packets affects the term

$\Pr(G_n|E[n] = k)$ in (7), but it is still $\omega(1/n)$ and we can eliminate it when we take the asymptotic equality in (8). As a result, the in-order delay exponent is same as that if we had a constant rate r of adding new packets to the transmitted stream. Hence we have proved that no other full-rank scheme can achieve a better (τ, λ) trade-off than $V[n] = \lceil nr \rceil$ for all n. ∎

Fig. 1 shows the (τ, λ) trade-off for the immediate feedback and no feedback cases, with success probability $p = 0.6$. The optimal trade-off with any feedback delay d lies in between these two extreme cases.

V. General Block-wise Feedback

In Section III and Section IV we considered the extreme cases of immediate feedback $(d = 1)$ and no feedback $(d = \infty)$ respectively. We now analyze the (τ, λ) trade-off with general block-wise feedback delay of d slots. We restrict our attention to a class of coding schemes called time-invariant schemes, which are defined as follows.

Definition 4 (Time-invariant schemes)**.** *A time-invariant scheme is represented by a vector $\mathbf{x} = [x_1, \cdots x_d]$ where x_i, for $1 \leq i \leq d$, are non-negative integers such that $\sum_i x_i = d$. In each block we transmit x_i independent linear combinations of the i lowest-index unseen packets in the stream.*

The above class of schemes is referred to as time-invariant because the vector $\mathbf{x}$ is fixed across all blocks. Note that there is also no loss of generality in restricting the length of the vector $\mathbf{x}$ to d. This is because each block can provide only up to d innovative coded packets, and hence there is no advantage in adding more than d unseen packets to the stream in a given block. Observe that as $d \to \infty$, the class of time-invariant schemes are equivalent to full-rank codes defined in Definition 3.

Conjecture 2. *To find the best (τ, λ) trade-off, there is no loss of generality in focusing on time-invariant schemes.*

We believe this conjecture is true because, it can be shown that any full-rank code can be expressed as a randomized combination of time-invariant schemes. Thus, if Conjecture 1 is true, it follows that there is no loss of generality in focusing on time-invariant schemes.

A. Analyzing the (τ, λ) of time-invariant schemes

. Given a vector $\mathbf{x}$, define p_d, as the probability of decoding the first unseen packet during the block, and S_d as the number of innovative coded packets that are received during that block. We can express $\tau_{\mathbf{x}}$ and $\lambda_{\mathbf{x}}$ in terms of p_d and S_d as,

$$(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}}) = \left(\frac{\mathbb{E}[S_d]}{d}, -\frac{1}{d} \log(1 - p_d) \right), \tag{10}$$

where we get throughput $\tau_{\mathbf{x}}$ by normalizing the $\mathbb{E}[S_d]$ by the number of slots in the slots. We can show that the probability $\Pr(T > kd)$ of no in-order packet being decoded in k blocks is equal $(1 - p_d)^k$. Substituting this in (1) we get $\lambda_{\mathbf{x}}$.

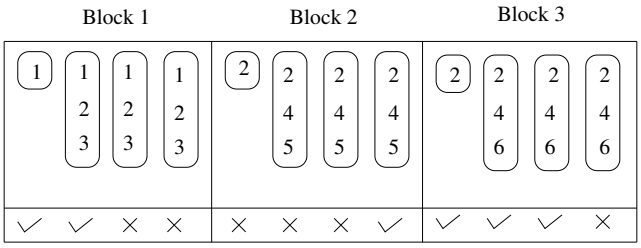

Fig. 2. Illustration of the time-invariant scheme $\mathbf{x} = [1, 0, 3, 0]$ with block size $d = 4$. Each bubble represents a coded combination, and the numbers inside it are the indices of the source packets included in that combination. The check and cross marks denote successful and erased slots respectively. The packets that are "seen" in each block are not included in the coded packets in future blocks.

Example 1. *Consider the time-invariant scheme $\mathbf{x} = [1, 0, 3, 0]$ where block size $d = 4$. That is, we transmit 1 combination of the first unseen packet, and 3 combinations of the first 3 unseen packets. Fig. 2 illustrates this scheme for one channel realization. The probability p_d and $\mathbb{E}[S_d]$ are,*

$$p_d = p + (1 - p)\binom{3}{3} p^3 (1 - p)^0 = p + (1 - p)p^3, \tag{11}$$

$$\mathbb{E}[S_d] = \sum_{i=1}^{3} i \cdot \binom{4}{i} p^i (1 - p)^{4-i} + 3p^4 = 4p - p^4, \tag{12}$$

where in (12), we get i innovative packets if there are i successful slots for $1 \leq i \leq 3$. But if all 4 slots are successful we get only 3 innovative packets. We can substitute (11) and (12) in (10) to get the (τ, λ) trade-off.

Remark 1. *Time-invariant schemes with different $\mathbf{x}$ can be equivalent in terms of the (τ, λ). In particular, given $x_1 \geq 1$, if any $x_i = 0$, and $x_{i+1} = w \geq 1$, then the scheme is equivalent to setting $x_i = 1$ and $x_{i+1} = w - 1$, keeping all other elements of $\mathbf{x}$ the same. For example, $\mathbf{x} = [1, 1, 2, 0]$ gives the same (τ, λ) as $\mathbf{x} = [1, 0, 3, 0]$.*

B. Cost of Achieving Optimal τ or λ

In Section III we saw that with immediate feedback, we can achieve $(\tau, \lambda) = (p, -\log(1 - p))$. However, with block-wise feedback we can achieve optimal τ (or λ) only at the cost of sacrificing the optimality of the other metric. We now find the best achievable τ (or λ) with optimal λ (or τ).

Claim 1 (Cost of Optimal Exponent λ)**.** *With block-wise feedback after every d slots, and in-order decoding exponent $\lambda = -\log(1 - p)$, the best achievable throughput $\tau = (1 - (1 - p)^d)/d$.*

Proof: If we want to achieve $\lambda = -\log(1-p)$, we require p_d in (10) to be equal to $1-(1-p)^d$. The only scheme that can achieve this is $\mathbf{x} = [d, 0, \cdots, 0]$, where we transmit d copies of the first unseen packet. The number of innovative packets S_d received in every block is 1 with probability $1 - (1 - p)^d$, and zero otherwise. Hence, the best achievable throughput is $\tau = (1 - (1 - p)^d)/d$ with optimal $\lambda = -\log(1 - p)$. ∎

This result gives us insight on how much bandwidth (which is proportional to $1/\tau$) is needed for a highly delay-sensitive application that needs λ to be as large as possible.

Claim 2 (Cost of Optimal Throughput τ)**.** *With block-wise feedback after every d slots, and throughput $\tau = p$, the best achievable in-order decoding exponent is $\lambda = -\log(1-p)/d$.*

Proof: If we want to achieve $\tau = p$, we need to guarantee an innovation packet in every successful slot. The only time invariant scheme that ensures this is $\mathbf{x} = [1, 0, \cdots, 0, d-1]$, or its equivalent vectors $\mathbf{x}$ as given by Remark 1. With $\mathbf{x} = [1, 0, \cdots, 0, d-1]$, the probability of decoding the first unseen packet is $p_d = p$. Substituting this in (10) we get $\lambda = -\log(1-p)/d$, the best achievable λ when $\tau = p$. ∎

Tying back to Fig. 1, Claim 1 and Claim 2 correspond to moving leftwards and downwards along the dashed lines from the optimal trade-off $(p, -\log(1-p))$. From Claim 1 and Claim 2 we see that both τ and λ are $\Theta(1/d)$, keeping the other metric optimal.

C. Finding the Best (τ, λ) Trade-off

For any given throughput τ, our aim is to find the coding scheme that maximizes λ. We first prove that any convex combination of achievable points (τ, λ) can be achieved.

Lemma 1 (Combining of Time-invariant Schemes)**.** *By randomizing between time-invariant schemes $\mathbf{x}^{(i)}$ for $1 \le i \le B$, we can achieve the throughput-delay trade-off given by any convex combination of the points $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$.*

Proof: Here we prove the result for $B = 2$, that is randomizing between two schemes. It can be extended to general B using induction. Given two time-invariant schemes $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ that achieve the throughput-delay trade-offs $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(2)}}, \lambda_{\mathbf{x}^{(2)}})$ respectively, consider a randomized strategy where, in each block we use the scheme $\mathbf{x}^{(1)}$ with probability μ and scheme $\mathbf{x}^{(2)}$ otherwise. Then, it is easy to see that the throughput on the new scheme is $\tau = \mu\tau_{\mathbf{x}^{(1)}} + (1-\mu)\tau_{\mathbf{x}^{(2)}}$.

Now we prove the in-order decoding exponent λ is also a convex combinations of $\lambda_{\mathbf{x}^{(1)}}$ and $\lambda_{\mathbf{x}^{(2)}}$. Let p_{d_1} and p_{d_2} be the probabilities of decoding the first unseen packet in a block using scheme $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ respectively. Suppose in an interval with k blocks, we use scheme $\mathbf{x}^{(1)}$ for h blocks, and scheme $\mathbf{x}^{(2)}$ in the remaining blocks, we have

$$\Pr(T > kd) = (1-p_{d_1})^h (1-p_{d_2})^{k-h}. \tag{13}$$

Using this we can evaluate λ as,

$$\lambda = \lambda_{\mathbf{x}^{(1)}} \lim_{k\to\infty} \frac{h}{k} + \lambda_{\mathbf{x}^{(2)}} \lim_{k\to\infty} \frac{k-h}{k} \tag{14}$$

$$= \mu\lambda_{\mathbf{x}^{(1)}} + (1-\mu)\lambda_{\mathbf{x}^{(2)}} \tag{15}$$

where we get (14) using (10). As $k \to \infty$, by the weak law of large numbers, the fraction h/k converges to μ. ∎

The main implication of Lemma 1 is that, to find the best (τ, λ) trade-off, we only have to find the points $(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}})$ that lie on the convex envelope of the achievable region spanned by all possible $\mathbf{x}$.

For general d, it is hard to search for the $(\tau_{\mathbf{x}}, \lambda_{\mathbf{x}})$ that lie on the optimal trade-off. We propose a set of time-invariant schemes that are easy to analyze and give a good (τ, λ) trade-off. In Theorem 2 we give the (τ, λ) trade-off for the proposed codes and show that for $d = 2$ and $d = 3$, it is the best trade-off among all time-invariant schemes.

Definition 5 (Proposed Codes for general d)**.** *For general d, we propose using the time-invariant schemes with $x_1 = a$ and $x_{d-a+1} = d-a$, for $a = 1, \cdots d$.*

In other words, in every block of d slots, we transmit the first unseen packet a times, followed by $d-a$ combinations of the first $d-a+1$ unseen packets. These schemes span the (τ, λ) trade-off as a varies from 1 to d, with a higher value of a corresponding to higher λ and lower τ. In particular, observe that the $a = d$ and $a = 1$ codes correspond to codes given in the proofs of Claim 1 and Claim 2.

Theorem 2 (Throughput-Delay Trade-off for General d)**.** *The codes proposed in Definition 5 give the trade-off points*

$$(\tau, \lambda) = \left(\frac{1-(1-p)^a + (d-a)p}{d}, -\frac{a}{d}\log(1-p) \right). \tag{16}$$

for $a = 1, \cdots d$. For $d = 2$ and $d = 3$, the piecewise linear curve joining these points is the best trade-off among all time-invariant schemes.

Proof: To find the (τ, λ) trade-off points, we first evaluate $\mathbb{E}[S_d]$ and p_d, as given in Section V-A. With probability $1-(1-p)^a$ we get 1 innovative packet from the first a slots in a block. The number of innovative packets received in the remaining $d-a$ slots is equal to the number of successful slots. Thus, the expected number of innovative coded packets received in the block is

$$\mathbb{E}[S_d] = 1 - (1-p)^a + (d-a)p \tag{17}$$

If the first a slots in the block are erased, the first unseen packet cannot be decoded, even if all the other slots are successful. Hence, we have $p_d = 1-(1-p)^a$. Substituting $\mathbb{E}[S_d]$ and p_d in (10), we get the trade-off in (16). By Lemma 1, we can achieve any convex combination of the (τ, λ) points in (16). In Lemma 2 and Lemma 3 below, we prove that for $d = 2$ and $d = 3$, the codes proposed in Definition 5 give the best trade-off among all time-invariant schemes. ∎

Lemma 2. *For $d = 2$, the codes proposed in Definition 5 give the best (τ, λ) trade-off among all time-invariant schemes.*

Proof: When $d = 2$ there are only two possible time-invariant schemes $\mathbf{x} = [2, 0]$ and $[1, 1]$ that give unique (τ, λ). By Remark 1, all other $\mathbf{x}$ are equivalent to one of these vectors in terms (τ, λ). The vectors $\mathbf{x} = [2, 0]$ and $[1, 1]$ correspond to the $a = 1$ and $a = 2$ codes proposed in Definition 5. Hence, the line joining their corresponding (τ, λ) points, as shown in Fig. 3, is the best trade-off for $d = 2$. ∎

Lemma 3. *For $d = 3$, the codes proposed in Definition 5 give the best (τ, λ) trade-off among all time-invariant schemes.*

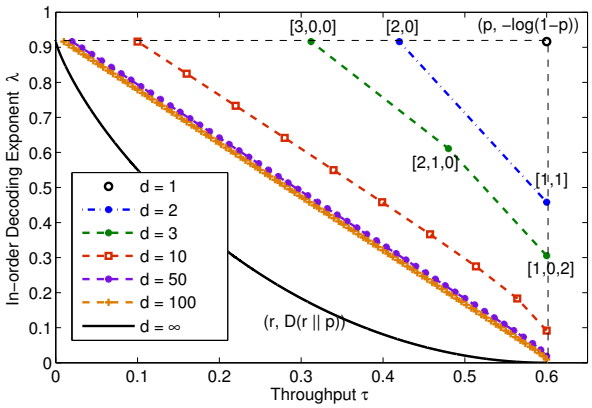

Fig. 3. The throughput-delay trade-off of the suggested coding schemes in Definition 5 for $p = 0.6$ and various values of block-wise feedback delay d. The trade-off becomes significantly worse as d increases. The point labels on the $d = 2$ and $d = 3$ trade-offs are $\mathbf{x}$ vectors of the corresponding codes.

Proof: When $d = 3$ there are four time-invariant schemes $\mathbf{x}^{(1)} = [1, 0, 2]$, $\mathbf{x}^{(2)} = [2, 1, 0]$, $\mathbf{x}^{(3)} = [1, 2, 0]$ and $\mathbf{x}^{(4)} = [3, 0, 0]$ that give unique (τ, λ), according to Definition 4 and Remark 1. The vectors $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(4)}$ correspond to the codes with $a = 1, 2, 3$ in Definition 5. The throughput-delay trade-offs $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, 2, 4$ achieved by these schemes are given by (16). From Claim 1 and Claim 2 we know that $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(4)}}, \lambda_{\mathbf{x}^{(4)}})$ have to be on the optimal trade-off. By comparing the slopes of the lines joining these points we can show that the point $(\tau_{\mathbf{x}^{(2)}}, \lambda_{\mathbf{x}^{(2)}})$ lies above the line joining $(\tau_{\mathbf{x}^{(1)}}, \lambda_{\mathbf{x}^{(1)}})$ and $(\tau_{\mathbf{x}^{(4)}}, \lambda_{\mathbf{x}^{(4)}})$ for all p. Fig. 3 illustrates this for $p = 0.6$. For the scheme with $\mathbf{x}^{(3)} = [1, 2, 0]$, we have

$$(\tau_{\mathbf{x}^{(3)}}, \lambda_{\mathbf{x}^{(3)}}) = \left((3p - p^3)/3, -(\log(1-p)^2(1+p))/3\right).$$

Again, by comparing the slopes of the lines joining $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, \cdots 4$ we can show that for all p, $(\tau_{\mathbf{x}^{(3)}}, \lambda_{\mathbf{x}^{(3)}})$ lies below the piecewise linear curve joining $(\tau_{\mathbf{x}^{(i)}}, \lambda_{\mathbf{x}^{(i)}})$ for $i = 1, 2, 4$. ■

Fig. 3 shows the trade-off given by (16) for different values of d. We observe that the trade-off becomes significantly worse as d increases. Thus we can imply that frequent feedback to the source is important in delay-sensitive applications to ensure fast in-order decoding of packets. As $d \to \infty$, and $a = \alpha d$, the trade-off converges to $((1-\alpha)p, -\alpha \log(1-p))$ for $0 \leq \alpha \leq 1$, which is the line joining $(0, -\log(1-p))$ and $(p, 0)$.

In Lemma 2 and Lemma 3 we showed that the codes proposed in Definition 5 give the best trade-off among all time-invariant schemes. Numerical results suggest that even for general d these schemes give a trade-off that is close to the best trade-off among all time-invariant schemes.

VI. Concluding Remarks

In this paper we analyze how block-wise feedback affects the trade-off between throughput τ and in-order decoding exponent λ, which measures the burstiness in-order packet decoding in streaming communication. When there is immediate feedback, we can simultaneously achieve the optimal τ and λ. But as the block size increases, and the frequency of feedback reduces, we have to compromise on at least one of these metrics. Our analysis gives us the insight that frequent feedback is crucial for fast in-order packet delivery.

Given that feedback comes in blocks of d slots, we present a spectrum of coding schemes that span different points on the (τ, λ) trade-off. Depending upon the delay-sensitivity and bandwidth limitations of the applications, these codes provide the flexibility to choose a suitable operating point on trade-off. Future directions include exploring the multicast scenario where there is a trade-off between throughput and delay, as well as between the different users that are sharing the channel.

References

[1] Sandvine Intelligent Networks, "Global Internet Phenomena Report." http://www.sandvine.com, Mar. 2013.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *ACM symposium on Theory of computing*, (New York, NY, USA), pp. 150–159, ACM, 1997.

[3] E. Martinian, *Dynamic Information and Constraints in Source and Channel Coding*. PhD thesis, MIT, Cambridge , USA, Sept. 2004.

[4] A. Badr, A. Khisti, W. Tan and J. Apostoupoulos, "Robust Streaming Erasure Codes based on Deterministic Channel Approximations," *International Symposium on Information Theory*, July 2013.

[5] P. Patil, A. Badr, A. Khisti and W. Tan, "Delay-Optimal Streaming Codes under Source-Channel Rate Mismatch," *Asilomar*, Nov. 2013.

[6] J. Sundararajan, D. Shah and M. Médard, "ARQ for Network Coding," in *International Symp. on Information Theory*, pp. 1651–1655, July 2008.

[7] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective Delay Control in Online Network Coding," in *International Conference on Computer Communications*, pp. 208–216, Apr. 2009.

[8] H. Yao, Y. Kochman and G. Wornell, "A Multi-Burst Transmission Strategy for Streaming over Blockage Channels with Long Feedback Delay," *IEEE Journal on Selected Areas in Communications*, Dec. 2011.

[9] G. Joshi, Y. Kochman, G. Wornell, "On Playback Delay in Streaming Communication," *International Symp. on Information Theory*, July 2012.

[10] A. Sahai, "Why Do Block Length and Delay Behave Differently if Feedback Is Present?," *IEEE Transactions on Information Theory*, vol. 54, pp. 1860–1886, May 2008.

[11] T. Cover and J. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 2nd ed., 1991.

[12] R. Durrett, *Probability: Theory and Examples*. Cambridge University Press, 4th ed., 2010.