Delay-Throughput Tradeoff for Streaming over Blockage Channels with Delayed Feedback

Huan Yao
MIT Lincoln Laboratory
Lexington, MA, USA
yaohuan@ll.mit.edu

Yuval Kochman and Gregory W. Wornell
Research Laboratory of Electronics, MIT
Cambridge, MA, USA
{yuvalko,gww}@mit.edu

***Abstract*—We focus on the problem of real-time streaming over a blockage channel with long feedback delay, as arises in real-time satellite communication from a comm-on-the-move (COTM) terminal. For this problem, we introduce a definition of delay that captures the real-time nature of the problem, which we show grows at least as fast as $O(\log(k))$ for memoryless channels, where k corresponds to the number of packets in the transmission. Moreover, we show that a tradeoff exists between this delay and a natural notion of throughput that captures the bandwidth requirements of the communication. We develop and analyze an efficient "multi-burst" transmission protocol for achieving good delay-throughput tradeoffs within this framework, which we show can be augmented with coding for additional performance gains. Simulations validate the new protocols on channels with and without memory.**

***Index Terms*—real-time communication; communications-on-the-move (COTM); ARQ; scheduling; packet-loss channel;**

I. Introduction

In recent years, military satellite communication capabilities are being extended to communications-on-the-move (COTM) terminals at the tactical edge. One challenge for on-the-move communication is channel blockage caused by foliage or buildings as terminals traverse rural or urban environments. To first order, these channels can be modeled as erasure channels with certain channel statistics. Techniques for dealing with erasures include error-control coding, retransmission protocols, and various hybrid schemes; see, e.g., [1]–[12] and the references therein. However, the long round-trip time (RTT) associated with satellite communication further complicates the problem. After a packet is transmitted, the transmitter has to wait a significant fraction of a second before an acknowledgment (ACK) is received indicating that the transmission was successful.

For non-real-time traffic such as bulk data transfers, rateless schemes provide natural solutions. Indeed, examples such as digital fountain codes [1], [2] and Raptor codes [3] are able to achieve low excess bandwidth and delay for different channel behaviors simultaneously. In particular, to communicate k bits, an arbitrarily long sequence of coded bits is sent through the erasure channel until sufficiently many bits are received to enable decoding. The number of bits required for decoding is only slightly greater than k, so that the code operates at near capacity. With such protocols, a single bit of feedback suffices to terminate the transmission. Moreover, provided the message size is large, the inefficiency due to feedback delay is small.

For real-time traffic consisting of a stream of (ordered) packets, such as voice, good solutions require more work. Indeed, rateless schemes are ill-suited as they would require waiting at the encoder until the end of the stream to encode and transmit, and waiting at the decoder until the entire stream can be decoded. Real-time applications require encoding and decoding to be progressive, with decoding of earlier parts of the stream happening before later parts are even encoded, so that playback can begin as soon as possible.

The design of such encoding and decoding is nontrivial when the RTT in the system is substantial. Assuming we simply transmit packets as they become available to the encoder, it is unclear how to handle packet losses. Retransmitting packets after a full RTT may incur a very large delay, but preemptive retransmission may be wasteful of bandwidth.

In [13], [14], an efficient approach is described for achieving the minimum possible delay at maximum possible throughput in such streaming scenarios, when no feedback is available. Examples of additional work beyond individual links include [15], which studies methods for achieving low delay via network coding and feedback. Additionally, field experiment work is described in [16].

In this work, we develop a framework for analyzing the problem of real-time streaming over a packet-erasure channel with long feedback delay, and show that a fundamental tradeoff exists between the bandwidth efficiency (throughput) and delay efficiency that can be achieved. Moreover, we develop an efficient multi-burst transmission (MBT) strategy for achieving good delay-throughput characteristics.

II. System Model and Performance Metrics

The model of interest is depicted in Fig. 1. A source generates a stream of mutually independent packets $p_1, p_2, \ldots$ of fixed size R at the rate of 1 packet per time unit (TU) starting at time 1, such that the kth packet p_k is generated at time k. A transmitter sends channel messages c_k at time k, which are a causal function of the source packets, i.e., c_k is only a function of $p_1, \ldots, p_k$. The channel has dynamic bandwidth, allowing messages c_k to have variable size $R_k = N_k R$, where $N_k > 0$

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

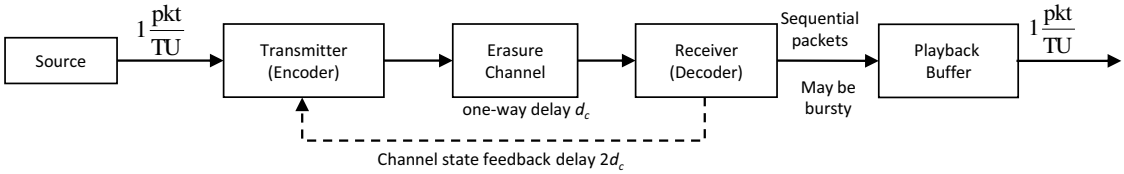

Fig. 1. Streaming system block diagram.

is arbitrary, though in this work we will generally restrict N_k to integer values. The dynamic bandwidth allows for the possibility of adapting to channel variations in real time, as periods with fewer blockages require fewer retransmissions.

The blockage channel is governed by a state sequence s_k, where the message c_k is either received after a fixed delay d_c if $s_k = 1$, or blocked if $s_k = 0$. The transmitter learns the channel state through an error-free packet acknowledgment fed back after a further delay of d_c, so the transmitter function is given by $c_k = g(p_1, \ldots, p_k, s_1, \ldots, s_{k-2d_c-1})$.

When the receiver is able to determine a source packet, it forwards it to a playback buffer. Due to the real-time nature of the transmission, the receiver is required to reproduce packets at the output in sequential order—if packet p_k is not received, all later packets p_j, $j > k$, must wait. We assume an infinite buffer for simplicity. The playback buffer can only playback one packet per time unit, all later packets are buffered.

For such real-time streaming systems, a natural definition of the delay experienced by packet p_k is

$$D_k \triangleq M_k - k, \tag{1}$$

where M_k denotes the time p_k is played back and k is the time p_k is generated. D_k is nondecreasing as there is no mechanism to "catch up."

Consider an example with $d_c = 3$. Packet p_1 is generated at time 1 and transmitted right away. Assume $s_1 = 1$ (channel open), then p_1 is received at time $d_c + 1 = 4$ and played back at time $M_1 = d_c + 2 = 5$, so $D_1 = M_1 - 1 = d_c + 1 = 4$. If $s_1 = 0$ (blocked), then the transmitter finds out about it at time $2d_c + 1 = 7$. If it chooses c_8 to be the concatenation of p_1 and p_8 (thus $N_8 = 2$), and $s_8 = 1$, then p_1 is received at time 11 and played back at time 12, thus $D_1 = 11$.

We would like to keep the delay low, while also keeping the number of channel uses (utilized bandwidth) small. However, there is tension between these two goals: while the transmitter still does not know if a message was blocked or not, for the sake of minimizing delay it should assume that it was blocked, thus repeat whatever information it contained until it is acknowledged. For low throughput the opposite holds: the transmitter should assume that the message was received, thus avoid the danger of unnecessarily repeating information.

Equipped with a statistical model for the state sequence s_k, we can quantify this tradeoff achieved by a specific scheme using particular delay and throughput figures of merit. Specifically, we consider an independent identically-distributed (i.i.d.) sequence with $\Pr\{s_k = 1\} = \rho$ in our analysis.

The *throughput metric* TM, which reflects the inefficiencies in channel utilization, is the ratio between the expected volume of data that was admitted by the channel, and the amount of data that the source emitted, averaged over time, i.e., $\mathrm{TM} = \lim_{k\to\infty} \mathrm{TM}_k$ where [1]

$$\mathrm{TM}_k \triangleq \frac{\sum_{j=1}^{\infty} E\{s_j R_j\}}{\sum_{j=1}^{k} R} = \frac{1}{k}\sum_{j=1}^{\infty} E\{s_j N_j\}. \tag{2}$$

$1/\mathrm{TM}$ corresponds to the bandwidth efficiency of the scheme.

We define the *delay metric* DM as the delay in excess of the delay $D_k^{\min}$ achievable by any scheme (such as the genie-assisted scheme in Section III-A), i.e., $\mathrm{DM} = \lim_{k\to\infty} \mathrm{DM}_k$ where

$$\mathrm{DM}_k \triangleq E\left[D_k - D_k^{\min}\right]. \tag{3}$$

III. Retransmission Protocols

This section analyzes some relatively simple retransmission (pure repetition) protocols. In such protocols, the transmitted packet c_k is the concatenation of the source packet p_k with $N_k - 1$ previously transmitted source packets. With such protocols, the throughput metric TM is the expected number of times that each source packet is received.

A. Ideal Genie-Assisted System Performance

As a performance bound on delay and throughput, we first examine a genie-assisted system in which the channel state is revealed in advance to the transmitter.[2] We later use the same framework to derive an achievable delay-throughput tradeoff in the presence of feedback delay.

Given channel knowledge, the optimal strategy (even without the restriction to simple repetition protocols) is to transmit each source packet p_k exactly once at the first instant $t \geq k$ the channel is open. This achieves the minimum delay possible, so $\mathrm{DM} = 0$. Each packet is received exactly once, so $\mathrm{TM} = 1$. No scheme can achieve lower values for these metrics.

Fig. 2 gives an example. Packet p_1 is sent and played back immediately. Packet p_2 is not sent at times 2 and 3 due to channel blockages. However, p_2, p_3, p_4 are successfully transmitted together at time 4. Packet p_2 is played back immediately, but p_3 and p_4 are buffered.

Examining the delay experienced by each packet, note that D_k depends on the longest burst of zeros experienced so far. Each time the longest burst of zeros lengthens, playback is interrupted and the delay for all subsequent packets is

[1] Note that after the transmitter is sure that the receiver has decoded all the information, transmission stops and from that moment on $R_j = N_j = 0$, so that the sum in the numerator of (2) is typically finite.

[2] Equivalently, one may assume that the state become known to the transmitter immediately after each message is sent, corresponding to instantaneous feedback.

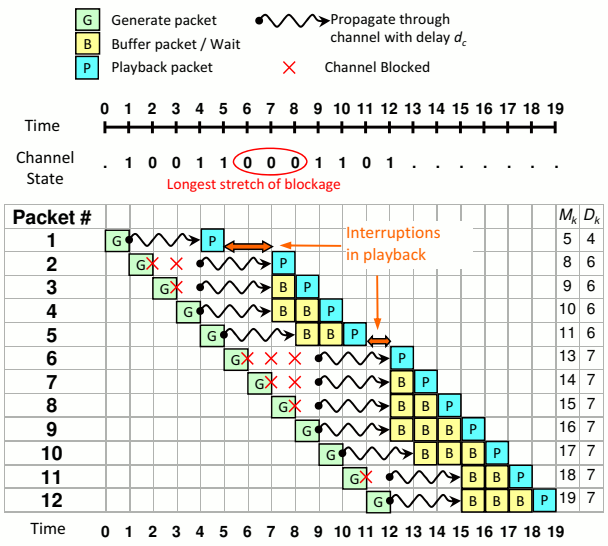

Fig. 2. Example of genie-assisted system operation with $d_c = 3$.

increased, such as at times 6 and 7, and again at time 12. These interruptions will generally happen with decreasing frequency. This is because, as we show next, the longest stretch of blockage, as well as the delay experienced by the genie-assisted scheme, grows like $O(\log(k))$.

Denoting by B_k the longest stretch of continuous blockage that *starts* at a time up to k, we have $D_k^{\min} = d_c + 1 + B_k$. In the example of Fig. 2, $B_k = 3$ for $6 \le k \le 12$. Defining

$$\bar{P}_{B_k}(b) \triangleq \Pr\{B_k \ge b\}, \tag{4}$$

we have

$$E[D_k^{\min}] = \sum_{t=1}^{\infty} \Pr\{D_k^{\min} \ge t\} = d_c + 1 + \sum_{b=1}^{\infty} \bar{P}_{B_k}(b), \tag{5}$$

where the sum is due to computing expected values from cumulative density functions for nonnegative integer random variables. We now develop an upper bound on this sum for the i.i.d. channel model, which is tight in the logarithmic sense.

Proposition 1: For an i.i.d. blockage channel sequence with $\Pr\{s_k = 1\} = \rho$, we have $\bar{P}_{B_k}(b) \le \min(1, (1-\rho)^b k)$.

Proof: The first term in the minimization, 1, is trivial. For the second term, in order to have b consecutive zeros, the channel sequence must take the form

$$\underbrace{*\cdots*}_{i}\underbrace{0\cdots0}_{b}\underbrace{*\cdots*}_{k-i-1}, \quad \text{for } i = 0, 1, \cdots, k-1.$$

For each i, the probability of having a particular pattern of this form is $(1-\rho)^b$. The probability of the union of $i = 0, 1, \cdots, k-1$ is at most $(1-\rho)^b k$. ■

Although this method double counts some channel patterns with multiple bursts of zeros, together with the upper bound of 1, it provides an upper bound sufficient for our analysis. It shows that $\bar{P}_{B_k}(b)$ decays exponentially in b for a fixed k. [3]

[3]For other channel types, $\bar{P}_{B_k}(b)$ mostly likely also decays exponentially in b for sufficiently large b.

Using Proposition 1 in (5), we obtain the upper bound

$$E[D_k^{\min}] \le d_c + 1 + \sum_{b=1}^{\infty} \min(1, (1-\rho)^b k) \tag{6}$$

$$\le d_c + 1 + \log_{\frac{1}{1-\rho}} k + \frac{1-\rho}{\rho}. \tag{7}$$

Eq. (7) is obtained by breaking the summation to two parts, the portion corresponding to $b \le \log_{\frac{1}{1-\rho}} k$ in which the summand is 1, and the remaining values of b for which the summand decays as $(1-\rho)$. To improve the bound, care is taken in handling the fractional part of $\log_{\frac{1}{1-\rho}} k$. It is easy to check that (7) holds with equality when $\log_{\frac{1}{1-\rho}} k$ is integer.

The bound (7) is tight up to a constant with respect to k. Specifically, first using numerical evaluation, and then propagating the inequality between (6) and (7), we obtain

$$E[D_k^{\min}] \ge d_c + 1 + \log_{\frac{1}{1-\rho}} k + \frac{1-\rho}{\rho} - \Delta(\rho) \tag{8}$$

$$\ge d_c + 1 + \sum_{b=1}^{\infty} \min(1, (1-\rho)^b k) - \Delta(\rho), \tag{9}$$

where the constant $\Delta(\rho)$ has the value of $\Delta(0.5) = 1.67$ and $\Delta(0.9) = 0.43$, and generally decreases with ρ.

As we will see, this logarithmic behavior in D_k holds not only in this genie-assisted case, but in our general case of interest involving delayed feedback of state information.

B. Optimizing Throughput: ARQ

In a traditional ARQ protocol, after the transmitter sends a packet, it waits for a full RTT, and only retransmits the packet when it is certain that the previous transmission was lost, so no packet can be received twice. Such a scheme achieves the minimal TM of 1, but suffers a large delay due to the long wait time between the retransmissions. In fact, $\mathrm{DM} = \infty$.

C. Optimizing Delay: Send-Until-ACK (SUA)

As an alternative to ARQ, the send-until-ACK (SUA) protocol minimizes delay without regard to the cost in throughput by repeatedly transmitting all packets generated so far at every time unit until each such packet is acknowledged. SUA achieves the lowest possible delay, as each source packet p_k is successfully transmitted at the first instant channel t opens up on or after time k, which is the same as in the genie-assisted case, so $\mathrm{DM} = 0$. However, SUA is very wasteful of bandwidth. The average number of times each packet is received is $\mathrm{TM} = 1 + 2d_c\rho$. There are always $2d_c$ additional transmissions after the first successful one due to the feedback delay; among those, on average $2d_c\rho$ are received.

D. Efficient Tradeoffs: Multi-Burst Transmission (MBT)

We now propose a multi-burst transmission (MBT) protocol as a balance between the extremes of ARQ and SUA. It differs from ARQ as follows. First, instead of transmitting a packet only once and waiting a full RTT, a packet is repeatedly transmitted one or more times in a burst of consecutive time units. After each burst, the transmitter waits a full RTT to see whether any of the transmissions made it through. Each

burst should be no longer than a full RTT. If not successful, additional bursts are attempted, up to a total of $N_{\rm TB}$, where $N_{\rm TB}$ is a design parameter. If all the bursts fail, the transmitter goes into SUA mode, i.e., repeatedly transmitting that packet until it is acknowledged. The motivation is to prevent overly long delays. Since the overall delay is determined by the fate of the most unfortunate packet, once a packet suffers repeated blockages, extra resources are spent to expedite its delivery. This incurs minimal degradation in TM, however, since relatively few packets enter SUA mode.

Each MBT scheme is fully characterized by the vector of burst lengths $\mathbf{v} = [v_1, v_2, \cdots, v_{N_{\rm TB}}]$. For example, $\mathbf{v} = [2, 4]$ and $d_c = 10$ means packet p_k is transmitted at times k, $k+1$ in the first burst. If both are lost, then a full RTT later, p_k is transmitted at $k+22$, $k+23$, $k+24$, $k+25$. If all four are lost, then p_k is transmitted continuously starting at time $k+46$ until an ACK is received. This retransmission schedule is carried out for all packets independently and simultaneously. For example, at time 26, p_1, p_2, p_3, p_4, p_{25}, p_{26} may all be (re)transmitted.

To evaluate the delay, see that in the above example, after the first transmission is lost, it delays the reception of p_k by 1. After the second transmission is lost, however, it incurs a delay of a full RTT, i.e., 21. We define w_b as the time between the bth and $(b+1)$st transmission. In the above example, with $\mathbf{w}$ denoting the vector of such inter-transmission times, we have $\mathbf{w} = [1, 21, 1, 1, 1, 21, 1, 1, \cdots]$. (For SUA, $w_b \equiv 1$; for ARQ, $w_b \equiv 2d_c + 1$.) Using techniques similar to those used to obtain (6), the average delay can be bounded by

$$E[D_k] \le d_c + 1 + \sum_{b=1}^{\infty} \min(1, (1-\rho)^b k) \cdot w_b, \qquad (10)$$

where $\min(1, (1-\rho)^b k)$ is the union bound on the probability that *any* one of the k packets has *all* of its first b transmissions blocked. Subtracting (9) from (10) and noting that only $N_{\rm TB}$ elements of $\mathbf{w}$ are larger than 1 (and equal $2d_c + 1$), we have

$$\begin{aligned} {\rm DM}_k &\le \sum_{b=1}^{\infty} \min(1, (1-\rho)^b k) \cdot (w_b - 1) + \Delta(\rho) \\ &\le 2d_c N_{\rm TB} + \Delta(\rho). \end{aligned} \qquad (11)$$

We now turn to the problem of designing the burst length vector $\mathbf{v}$ to minimize TM given a target DM. From the analysis above, given a target DM, a total of $N_{\rm TB} \approx {\rm DM}/(2d_c)$ bursts are required. We optimize the $N_{\rm TB}$ elements of $\mathbf{v}$ by induction. To this end, we define $\tau(\mathbf{v})$ to be the TM associated with an MBT scheme with burst length vector $\mathbf{v}$. It can be shown that

$$\tau(\mathbf{v}) = \rho v_1 + (1-\rho)^{v_1} \cdot \tau(\mathbf{v}_2^{\rm end}), \qquad (12)$$

where v_1 is the first burst length, and $\mathbf{v}_2^{\rm end}$ denotes the vector of remaining entries in $\mathbf{v}$. The intuition is that the first burst of v_1 transmissions always happen; among those, ρv_1 many will be received on average. The probability of all v_1 transmissions being blocked is $(1-\rho)^{v_1}$; conditioned on this, $\tau(v_2^{\rm end})$ transmissions are expected to be received. This induction takes advantage of the full RTT wait time between bursts, so after a burst completely fails, there is a "clean restart." The initial condition of the induction is $\tau(\varnothing) = 1 + 2d_c\rho$, corresponding to doing SUA alone. The elements of $\mathbf{v}$ can be optimized one at a time starting from $v_{N_{\rm TB}}$ working backwards:

$$\begin{aligned} v_n &= \arg\min_{v_n} \tau(\mathbf{v}_n^{\rm end}) \\ &= \arg\min_{v_n} \rho v_n + (1-\rho)^{v_n} \cdot \tau(\mathbf{v}_{n+1}^{\rm end}) \\ &= \lfloor \log_{\frac{1}{1-\rho}} \tau(\mathbf{v}_{n+1}^{\rm end}) \rfloor + 1. \end{aligned} \qquad (13)$$

In the case of $d_c = 10$ and $0.44 \le \rho \le 0.56$, the optimal burst lengths are $v_{N_{\rm TB}} = 4$, $v_{N_{\rm TB}-1} = 2$, and $v_{N_{\rm TB}-2} = \cdots = v_1 = 1$. This solution suggests that when the allowable DM is sufficiently large, we should first do ARQ. After $N_{\rm TB} - 2$ transmissions, we get close to the allowable delay, we should then do a longer burst of 2 and then an even longer burst of 4. Should all those fail, we send continuously until ACK is received. When channel blockage is less frequent, i.e., larger ρ, the optimal MBT burst lengths are shorter, i.e., more "ARQ-like." For example, when $0.62 \le \rho \le 0.75$, the optimal burst lengths are $v_{N_{\rm TB}} = 3$, and $v_b = 1$ for $b < N_{\rm TB}$.

The above scheme achieves ${\rm DM} \approx 2d_c N_{\rm TB}$, spaced $2d_c$ apart as $N_{\rm TB}$ takes on various integer values. To achieve intermediate DM values, we allow the time β between the last burst and the SUA region to be in $1 \le \beta \le 2d_c$ rather than fixing $\beta = 2d_c + 1$. With this generalization, the optimal $\mathbf{v}$ has $v_{N_{\rm TB}} = \lfloor \log_{\frac{1}{1-\rho}} \left((1-\rho)^{\beta - 2d_c - 1} + \rho\beta - \rho \right) \rfloor + 1$, with $v_{N_{\rm TB}-1}, \ldots, v_1$ obtained via the induction using (13). We omit the proof due to space limitations.

The delay-throughput performance of the optimized MBT protocol is shown by the lower curve in Fig.3. It shows a steep initial decline in TM: allowing a little excess delay (small DM) leads to dramatic bandwidth savings. But additional delays provide diminishing returns. The upper curve in Fig. 3 corresponds to the $({\rm DM}, {\rm TM})$ pairs achievable using a truncated ARQ scheme that simply switches from ARQ to SUA after a prescribed time, which is essentially a special case of the MBT scheme with all burst lengths equal to 1. Earlier switching leads to lower DM and higher TM. The gap between the two curves shows that there is a significant benefit to doing bursts rather than doing single retransmissions. The genie-assisted and SUA cases are also shown for comparison.

IV. Coded Protocols

In this section we suggest simple coded enhancements to the preceding retransmission protocols. The schemes we consider utilize the idea of encoding by computing random linear combinations of packets commonly seem in, e.g., network coding [17]. In particular, at its simplest, any basic repetition protocol can be augmented with coding by replacing any transmission of packet p_k by a random linear combination y_k of all packets p_j, $j \le k$,[4] over a large enough alphabet of coefficients. The size of each y_k is also R, same as p_k.

[4]Packets that the transmitter knows that were already decoded may be excluded without affecting performance.

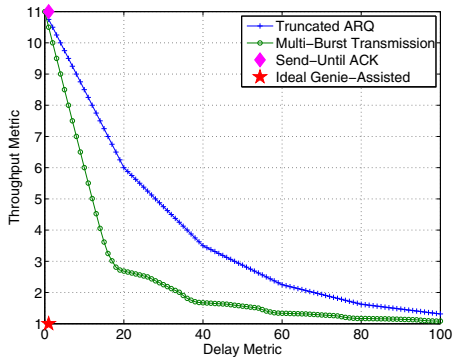

Fig. 3. The delay-throughput tradeoff for optimized MBT and truncated ARQ, compared to the genie-assisted bound and to the SUA scheme. ARQ performance is not shown, since it yields $\mathrm{DM} = \infty$. Channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.

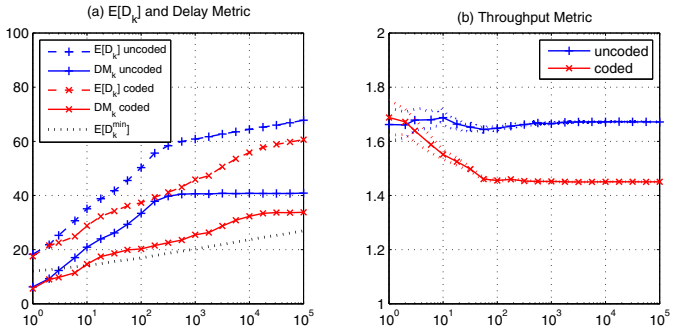

Fig. 4. The delay and throughput behavior achieved by a two-burst MBT protocol with $\mathbf{v} = [2, 4]$, with and without coding. Channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.

The performance of any such coded scheme is always at least as good as that of its base retransmission protocol, since by our definition of delay a packet must wait for all previous packets, thus the linear combination can always be undone. The gain provided by coding is that if p_k has been decoded already, y_k may be used to contribute towards decoding one of the earlier packets. While the analysis of this coding gain is difficult, the gains can be easily quantified by simulation, as we develop next.

V. Simulation Results

We now present simulation results on the delay and throughput performance of MBT schemes both when augmented by coding, and for channels with memory. For each scenario, simulations were performed using k up to 10^5 packets and 500 Monte Carlo trials to obtain suitable statistical averaging. For each run, D_k was recorded. Using the same channel sequence, the genie-assisted delay $D_k^{\min}$ was also evaluated. The difference, $D_k - D_k^{\min}$, was then computed. Averaging over all runs leads to an estimate of $E[D_k]$, $E[D_k^{\min}]$, and DM_k. TM is obtained by counting packets received.

A. Coding Gain

To illustrate the performance gain due to coding, we compare a particular scheme and its coded version under an i.i.d. channel with $\rho = 0.5$ and $d_c = 10$. We choose the two-burst MBT protocol described in Section III-D with $\mathbf{v} = [2, 4]$.

Fig. 4(a) shows $E[D_k]$ and DM_k. The dotted line at the bottom is the delay $E[D_k^{\min}]$ achieved by the genie-assisted system. The dashed lines are the $E[D_k]$ achieved by the uncoded and coded schemes. They both have the same limiting slope as the dotted line. The solid lines are delay metrics, which are the differences between $E[D_k]$ and $E[D_k^{\min}]$. They both become flat and reach a final value (as $k \to \infty$). The red $(\times)$ lines are noticeably lower than the blue $(+)$ lines, indicating the advantage of coding at all values of k.

Fig. 4(a) also shows how delay behaves at finite k. Though both $E[D_k]$ and $E[D_k^{\min}]$ grow to infinity as $k \to \infty$, their finite difference, DM_k, is significant for finite k values of interest. For example, for $1\mathrm{TU} = 20\mathrm{ms}$, $k = 10^5$ corresponds to 33 minutes. At this point, $E[D_k^{\min}]$ is 0.54 sec (best possible), and $E[D_k]$ achieved by the uncoded and coded two-burst MBT are 1.36 sec and 1.21 sec, respectively. The differences are noticable to end users.

Fig. 4(b) plots TM_k as a function of k together with the one standard deviation spread (dotted lines). It shows TM is indeed constant in k in the uncoded case (top curve). With coding, for large k, TM improves significantly, from 1.67 to 1.45, about 33% closer to the minimum TM of 1. When k is smaller, there is less gain. When $k = 1$, there is no coding gain, as there is no coding to perform.

TABLE I
DM and TM achieved for memoryless channel with $\rho = 0.5$

	Retransmission Analytical		Retransmission Simulation		Coded Simulation	
v	DM	TM	DM	TM	DM	TM
[4]	20	2.6875	20.51	2.6868	13.93	2.1563
[2,4]	40	1.6719	40.88	1.6718	33.80	1.4505
[1,2,4]	60	1.3359	60.88	1.3358	55.89	1.2422
[1,1,2,4]	80	1.1680	80.87	1.1680	77.68	1.1309
[1,1,1,2,4]	100	1.0840	100.95	1.0841	98.76	1.0709

Table I shows the DM and TM achieved by a range of MBT retransmission protocols and their coded counterparts. The worst-case one-standard deviation of the DM and TM values are 0.13 and 0.0005, respectively. The simulation values are evaluated at $k = 10^5$ as an approximation to $k \to \infty$. The simulation results for the uncoded retransmission schemes closely match the analytical values. In particular, TM matches to three decimal places; the DM simulation values are within a factor of $\Delta(0.5) = 1.67$ above the analytical values.

These results are also shown in Fig. 5 in blue with circular markers. Comparing the coded schemes to their uncoded counterparts, coding improves both DM and TM. Also, there is a greater improvement for schemes with fewer bursts (smaller DM). This is because when N_{TB} is large, MBT generally starts with an ARQ phase of single transmissions. Since the form of coding we consider does not help with ARQ, typically MBT schemes do not benefit from coding when DM is large.

B. Channels with Memory

While the retransmission schemes described in this paper have only been analyzed in the context of memoryless channels, the framework (and resulting protocols) can also be applied to channels with memory. To study the impact

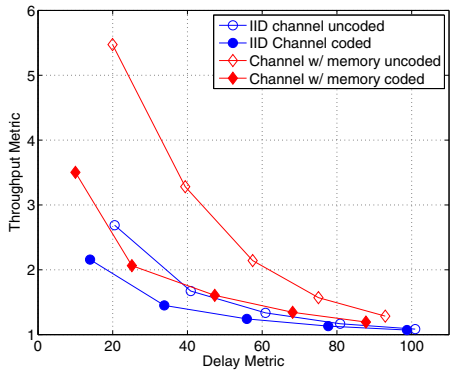

Fig. 5. The delay-throughput tradeoff for channels with and without memory, using MBT protocols with and without coding, $d_c = 10$ and $\rho = 0.5$.

of channel memory, additional simulations were run with a two-state Markov blockage channel with average open and blockage durations both being 6 TU. All other aspects of the simulation were kept the same including the MBT scheme parameters. The DM and TM achieved are plotted in Fig. 5 together with the i.i.d. channel results for comparison.

For both uncoded and coded schemes, and across all five MBT schemes in Table I, TM increased while DM decreased when channel memory is introduced. TM becomes larger because after the initial successful reception, due to channel memory, there is a higher conditional probability of successful reception of the retransmissions that follow closely after. For the $\mathbf{v} = [4]$ scheme, 50% of the time, the first transmission is successfully received, and the next three retransmissions, which are likely to also be received, contribute significantly to the TM, which overall increased by nearly 3 in the uncoded case. For the MBT protocols with more bursts, fewer packets require the last burst of 4, so there is less impact.

The slight decrease in the delay metric is actually due to the way it is defined. When the channel has memory, it is more likely to have long bursts of blockages. This increases $E[D_k^{\min}]$. The packet delays $E[D_k]$ are also increased, but to a lesser degree. Consider the $\mathbf{v} = [1, 1, 2, 4]$ case, after the first transmission is blocked, we wait for 21 TU before retransmission. This leads to the same 21 TU delay as in the i.i.d. channel case. However, in the genie-assisted case, after the initial blockage, the packet is likely to be blocked a bit longer, so the delay is relatively longer than the i.i.d. channel case. Accordingly, DM decreases.

Coding offers more significant improvement when channel has memory. This is because while a short burst of transmission often leads to duplicate (useless) receptions in the uncoded case, in the coded case, the multiple receptions can be used to decode earlier packets.

The MBT schemes used in this section are optimized for the i.i.d channel case, as shown in Section III-D, but not for channels with memory. Optimization for additional classes of channels is part of ongoing research.

VI. Summary

We studied the problem of real-time streaming over blockage channel with long feedback delay. We showed that a practical MBT scheme, blending ARQ and SUA, achieves an $O(\log(k))$ delay that is only an additive factor worse than a genie-assisted system. The MBT scheme achieves a particular delay-throughput tradeoff by varying its design parameters, from which we see that relaxing delay requirements even slightly can significantly reduce bandwidth requirements. The framework was also used to evaluate the benefit of coding and the impact of channel memory via simulations. Coding improves both delay and throughput, especially in the low delay regime, and for channels with memory. Similar analysis can be used for other protocols and blockage models.

References

[1] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, (Vancouver, Canada), pp. 56-67, Jan. 1998.

[2] M. Luby, "LT codes," in *Proc. IEEE Symp. Found. Comp. Sci. (FOCS)*, (Vancouver, Canada), pp. 271-282, Nov. 2002,

[3] Shokrollahi, "Raptor codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, (Chicago, IL), July 2004.

[4] S. Lin and D. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[5] S. Lin, D. Costello, and M. Miller, "Automatic-repeat-request error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-17, Dec. 1984.

[6] D. Costello, J. Hagenauer, H. Imai, and S. Wicker, "Applications of error-control coding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2531-2560, Oct. 1998.

[7] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Trans. Inform. Theory*, vol. 20, no. 3, pp. 388-389, May 1974.

[8] R. Mantha and F. R. Kschischang, "A capacity approaching hybrid ARQ scheme using Turbo codes," in *Proc. Global Commun. Conf. (GLOBECOM)*, (Rio de Janeiro, Brazil), pp. 2341-2345, Dec. 1999.

[9] D. N. Rowitch and L. B. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948-959, June 2000.

[10] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, July 2001.

[11] S. Sesia, G. Caire, and G. Vivier, "Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1311-1321, Aug. 2004.

[12] Soljanin, N. Varnica, and P. Whiting, "Incremental redundancy hybrid ARQ with LDPC and raptor codes," submitted to *IEEE Trans. Inform. Theory*, (preprint, 2005).

[13] E. Martinian, *Dynamic information and constraints in source and channel coding*, PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, Sep. 2004.

[14] E. Martinian and G. .W. Wornell, "Universal Codes for Minimizing Per-User Delay on Streaming Broadcast Channels," in *Proc. Allerton Conf. Commun., Contr., Computing*, (Monticello, IL), Oct. 2003.

[15] E. Drinea, C. Fragouli, and L Keller, "Delay with Network Coding and Feedback," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, (Seoul, Korea), pp. 844-848, June 2009.

[16] J. Schodorf, "EHF Satellite Communications on the Move: Experimental Results," Tech. Rep. 1087, MIT Lincoln Laboratory, Lexington, MA, Aug. 2003.

[17] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. Allerton Conf. Commun., Contr., Computing*, (Monticello, IL), Sep. 2004.