A Multi-Burst Transmission Strategy for Streaming Over Blockage Channels with Long Feedback Delay

Huan Yao, *Member, IEEE*, Yuval Kochman *Member, IEEE*, Gregory W. Wornell *Fellow, IEEE*

Abstract—We consider streaming over a blockage channel with long feedback delay, as arises in, e.g., 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, a tradeoff exists between this delay and a natural notion of throughput we introduce to capture the bandwidth requirements of the communication. We develop and analyze an efficient "multi-burst" transmission (MBT) protocol for achieving good delay-throughput tradeoffs within this framework, which we show to be robust and near-optimal within the class of retransmission protocols with fixed schedules. The MBT protocol can be augmented with coding for additional performance gains. Simulations validate the new protocols, including when peak bandwidth and delay constraints are imposed.

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

I. INTRODUCTION

IN RECENT years, system designers have been working to extend military satellite communication capabilities 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. These channels can be modeled as packet erasure channels with certain channel statistics. In addition to blockage, satellite communication suffers from a long round-trip time (RTT): 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, forward error correction (FEC) provides a natural solution. Moreover, rateless schemes allow optimality even when there is no statistical model for the channel. Indeed, examples such as

digital fountain codes [1], [2] and Raptor codes [3] are able to achieve near-optimal delay and throughput simultaneously. In particular, 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 the number of data bits, so that the code operates close to 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 that are both generated and consumed in real time—such as arises when watching a real-time surveillance video or having a voice conversation—ARQ methods can be used. The retransmission protocol ensures reliable delivery of the ordered packets. However, existing ARQ methods typically do not work well for severely-blocked, long-delay channels, nor when very low delay is required.

Basic types of ARQ protocols include stop-and-wait ARQ, go-back-N ARQ, and selective repeat (SR) ARQ, with SR-ARQ generally considered the most efficient of these. In all three variants, a packet is retransmitted only if the transmitter knows that the previous transmission is lost. This avoids any unnecessary retransmissions. However, when the channel loss rate is high, there is a non-negligible probability that it would take several retransmissions for a packet to get through. For example, for a channel that is blocked half of the time, with probability $1/4$ a packet would suffer two failed retransmissions and would thus take at least two RTT to be received. Furthermore, when ordered packet delivery is required, one packet that is not received delays all packets that come after. For a long stream, the probability of at least one packet requiring many retransmissions, which implies a large delay for the whole stream, is high.

Hybrid ARQ (HARQ) schemes that utilize FEC reduce the number of retransmissions by lowering the effective channel loss rate [4]–[6]. However, HARQ schemes perform well only when the blockage probability is low; see, e.g., [7], where a HARQ technique was applied to interactive voice over a moderately blocked satellite channel as a COTM terminal was driven in a city environment, and the resulting user experience was poor. In order to be effective in more severe conditions, the kinds of FEC traditionally used would have to span many independent channel realizations lasting many RTT, and inducing large delays. For some real-time applications, it may be desirable to achieve shorter delays (on the order of at most a few RTT). In the extreme, a user may require the minimum delay that the channel allows, corresponding to

every packet being delivered as soon as the physical channel becomes unblocked after the packet is generated—despite whatever cost in throughput this might incur. This kind of tradeoff is not offered by traditional ARQ.

In this paper, we explicitly consider the above tradeoff and propose a class of multi-burst transmission (MBT) schemes that achieve lower delay at the expense of throughput by allowing preemptive retransmission, the amount of which can be adjusted between two extremes. One extreme has no preemptive retransmission, corresponding to ARQ. The other extreme uses as much preemptive retransmission as is necessary to achieve the minimum delay allowed by the physical channel. Between the two extremes our MBT schemes achieve an efficient throughput-delay tradeoff. Given a target delay, MBT starts with relatively little preemptive retransmission, then increases the amount as a packet's delivery deadline approaches, corresponding to changing the strategy from one extreme to the other in time. As we will see, such a dynamic approach can be quite efficient in its use of bandwidth.

This paper builds on our initial results in [8], [9], further developing, analyzing and optimizing the scheme. Among other results, we show that such schemes can be designed to be robust to mismatch in the channel model, and that the MBT delay-throughput tradeoff achieved is nearly optimal within a rather natural class of retransmission strategies, even when practical considerations are taken into account. To develop the key insights, our analysis in this paper largely focuses on memoryless channels; a more detailed development of the case of channels with memory is contained in [10].

There have been a variety of other delay-focused investigations, under different feedback assumptions. For example, in [11], [12], an efficient approach is described for achieving the minimum possible delay at maximum possible throughput in such streaming scenarios, when no feedback is available. Additional examples of work beyond individual links include [13], which studies methods for achieving low delay via network coding and immediate feedback with zero delay.

An outline of the paper is as follows. Section II introduces the system model of interest and defines the throughput and delay performance metrics. Section III discusses some key baseline retransmission protocols: two extreme strategies and a conceptual genie-assisted system for lower bounding delay in our analysis. Section IV develops the MBT schemes, analyzing and optimizing their delay and throughput performance, and Section V introduces coded extensions of MBT, the performance benefits of which are developed via simulations. Section VI investigates practical considerations, including the impact of peak bandwidth constraints, maximum delay constraints, and having both simultaneously. Finally, Section VII contains some concluding remarks.

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 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* data bandwidth, allowing messages c_k to have variable size $R_k = N_k R$, where $N_k > 0$ is arbitrary, though in this work we will generally restrict N_k to integer values.

Traditionally, a streaming session would be allocated a *fixed* amount of data bandwidth to support the flow and the expected retransmissions. However, this fixed allocation has inefficiencies when the channel is dynamic. During periods when the channel happens to be blocked less than average, some data bandwidth are wasted; when the channel happens to be blocked more often, there may not be enough data bandwidth to perform all the retransmissions needed, thus incurring additional delay. Incorporating dynamic data bandwidth in the model allows for the possibility of adapting to channel variations in real time. This is possible when, e.g., the flow of interest shares a physical channel (fixed bandwidth) with many other traffic flows from the same transmitter, and it has relatively lower rate but higher priority, so that it can transmit at higher instantaneous rates as needed. While our analytical study puts no limit on N_k for simplicity, in practical systems, there would be a limit. This is addressed in Section VI-A.

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 $c_k = g(p_1, \ldots, p_k, s_1, \ldots, s_{k-\text{RTT}})$, where RTT $= 2d_c + 1$.

When the receiver is able to determine a source packet, it forwards it to a playback buffer. The playback buffer can only playback one packet per time unit; all later packets are buffered. 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. For analysis, we assume an infinite buffer. In practice, users would stop waiting for a packet that is delayed excessively; such modifications are considered in Section VI-B.

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

$$D_k \triangleq M_k - k, \qquad (1)$$

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

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 packet was blocked or not, for the sake of minimizing delay it should assume that it was blocked, and thus continually resend the information until it
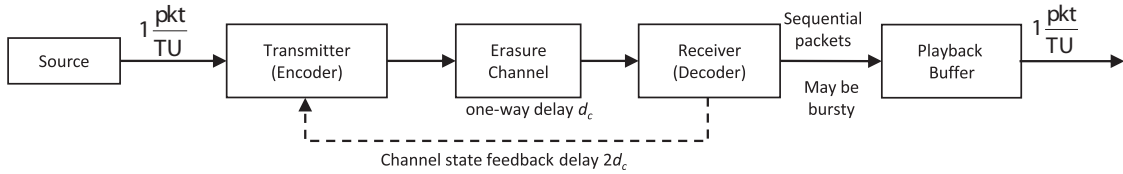
Fig. 1. Streaming system block diagram.

is acknowledged. For low bandwidth utilization the opposite holds: the transmitter should assume that the packet was received, thus avoiding the danger of unnecessarily repeating information.

Equipped with a statistical model for the state sequence s_k, we can quantify the tradeoff achieved by a specific scheme using particular delay and throughput figures of merit. For our analysis, we choose an independent identically-distributed (i.i.d.) sequence model with $\Pr\{s_k = 1\} = \rho$, with $0 < \rho \le 1$.

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., $\text{TM} = \lim_{k \to \infty} \text{TM}_k$, where[1]

$$\text{TM}_k \triangleq \frac{\sum_{j=1}^{\infty} E\{s_j R_j\}}{\sum_{j=1}^{k} R} = \frac{1}{k} \sum_{j=1}^{\infty} E\{s_j N_j\}. \qquad (2)$$

Note that TM reflects the number of *received* retransmissions, rather than the sent ones, thus it does not reflect the full bandwidth utilization. However, for an i.i.d. channel model, the expected number of retransmissions performed is simply TM/ρ, thus minimizing TM indeed optimizes the usage of transmit bandwidth. Our choice of definition is such that, regardless of the channel parameter ρ, $1/\text{TM}$ is the excess bandwidth factor of the scheme, relative to that of a genie-assisted system that knows the entire state sequence s_k in advance (see Section III-A).

We define the *delay metric* DM as the delay in excess of the best delay $D_k^{\min}$ achievable when the channel realization is known in advance (which is a fundamental limit of the channel), i.e., $\text{DM} = \lim_{k \to \infty} \text{DM}_k$ where

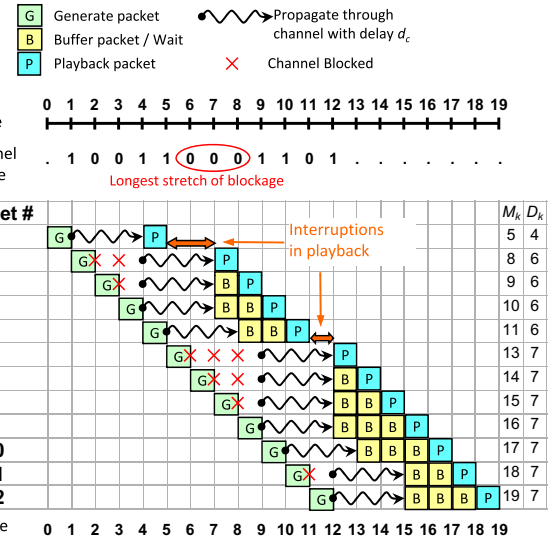$$\text{DM}_k \triangleq E\left[D_k - D_k^{\min}\right]. \qquad (3)$$

III. SIMPLE RETRANSMISSION PROTOCOLS

This section analyzes three 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

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

revealed in advance to the transmitter.[2] We later use similar analysis 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 $t \ge k$ the channel is open. This achieves the minimum delay possible, so DM = 0. Each packet is received exactly once, so TM = 1. No scheme can achieve lower values for these metrics.

Fig. 2 depicts 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 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

$$D_k^{\min} = d_c + 1 + B_k.$$

---

[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 becomes known to the transmitter immediately after each message is sent, corresponding to instantaneous feedback.

In the example of Fig. 2, $B_k = 3$ for $6 \leq k \leq 12$. Defining

$$\bar{P}_{B_k}(b) \triangleq \Pr\{B_k \geq b\}, \qquad (4)$$

we have

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

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 (5) for the i.i.d. channel model, which is tight in the logarithmic sense.

We begin by showing that $\bar{P}_{B_k}(b)$ decays exponentially in b for a fixed k.[3]

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

The following proof double-counts some channel patterns with multiple bursts of zeros, but when combined with the upper bound of 1, is sufficient for our purposes.

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

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

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 $(1-\rho)^b k$.   ∎

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

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

$$\leq d_c + 1 + \log_{\frac{1}{1-\rho}} k + \frac{1-\rho}{\rho}. \qquad (7)$$

Eq. (7) is obtained by breaking the summation into two parts: that corresponding to $b \leq \log_{1/(1-\rho)} k$ 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 $\log_{1/(1-\rho)} k$.[4] As is easy to verify, (7) holds with equality when $\log_{1/(1-\rho)} k$ is an integer.

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

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

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

where the constant $\Delta(\rho)$ represents the slack in the bound. It generally decreases with ρ, taking values shown in Table I.

As we will see, the 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.

---

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

[4]Let $0 \leq q < 1$ be the fractional part of $\log_{1/(1-\rho)} k$. The second part of the summation in (6) sums to $(1-\rho)^{1-q}\rho^{-1}$. But $(1-\rho)^{-1} > 1$ since $\rho \in (0,1]$, so $(1-\rho)^{-q}$ is convex, whence $(1-\rho)^{-q} \leq 1 + \rho(1-\rho)^{-1}q$.

TABLE I
SLACK IN DELAY BOUND $\Delta(\rho)$, EVALUATED NUMERICALLY

ρ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\Delta(\rho)$	25.88	9.126	4.591	2.664	1.668	1.095	0.746	0.534	0.425

B. Optimizing Throughput

In a traditional ARQ protocol, after the transmitter sends a packet, it waits for a full RTT, i.e., $2d_c+1$, 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 minimum TM of 1, but suffers a large delay due to the long wait time between the retransmissions, as we now quantify.

We consider an ideal SR-ARQ scheme with unlimited transmit and receive window size. A packet p_k that is blocked b times, i.e., at times $k, k+\text{RTT}, \cdots, k+(b-1)\cdot\text{RTT}$, suffers a delay of at least $d_c + 1 + b \cdot \text{RTT}$ instead of $d_c + 1 + b$ as in the genie-assisted case. The factor of RTT leads to $E[D_K]$ growing as $\text{RTT} \cdot \log_{1/(1-\rho)} k$, [cf. (7)], whence 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 DM $= 0$. However, SUA is very wasteful of bandwidth. The average number of times each packet is received is 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.

IV. EFFICIENT TRADEOFFS: MULTI-BURST TRANSMISSION (MBT)

We now develop 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_{TB}, where N_{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 $\mathbf{v} = [v_1, v_2, \cdots, v_{N_{\text{TB}}}]$. 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 $= 2d_c+1 = 21$ 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, note that in the above example, after the first transmission is lost, the reception of p_k is delayed by 1. After the second transmission is lost, however, a delay of a full RTT is incurred, i.e., 21. We define w_b as the time between the bth and $(b+1)$st transmission. In the above example, with **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, $w_b \equiv \text{RTT}$.) Using techniques similar to those used to obtain (6), the average delay can be bounded by

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

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), we have

$$\text{DM}_k \leq \sum_{b=1}^{\infty} \min(1, (1-\rho)^b k) \cdot (w_b - 1) + \Delta(\rho)$$
$$\leq \sum_{b=1}^{\infty} (w_b - 1) + \Delta(\rho). \qquad (11)$$

Similar to the genie-aided case, numerical evidence suggests that the inaccuracy in the union bound approximation is small and (11) is close to holding with equality. This insight provides a key rule-of-thumb: DM *is approximately the total accumulated non-transmission time.*

For MBT, only N_{TB} elements of **w** are larger than 1, and they all equal $\text{RTT} = 2d_c + 1$. Therefore, we have, using (11) but neglecting the $\Delta(\rho)$ term,

$$\text{DM} \approx 2d_c N_{\text{TB}}. \qquad (12)$$

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

$$\tau(\mathbf{v}) = \rho v_1 + (1-\rho)^{v_1} \cdot \tau(\mathbf{v}_2^{\text{end}}), \qquad (13)$$

where v_1 is the first burst length, and $\mathbf{v}_2^{\text{end}}$ denotes the vector of remaining entries in **v**. The intuition is that the first burst of v_1 transmissions always happens; 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, $\tau(v_2^{\text{end}})$ 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 **v** can be optimized one

at a time starting from $v_{N_{\text{TB}}}$ working backwards:

$$v_n = \arg \min_{v_n} \tau(\mathbf{v}_n^{\text{end}})$$
$$= \arg \min_{v_n} \rho v_n + (1-\rho)^{v_n} \cdot \tau(\mathbf{v}_{n+1}^{\text{end}})$$
$$= \lfloor \log_{1/(1-\rho)} \tau(\mathbf{v}_{n+1}^{\text{end}}) \rfloor + 1. \qquad (14)$$

In the case of $d_c = 10$ and $0.44 \leq \rho \leq 0.56$, the optimal burst lengths are $v_{N_{\text{TB}}} = 4$, $v_{N_{\text{TB}}-1} = 2$, and $v_{N_{\text{TB}}-2} = \cdots = v_1 = 1$. This solution suggests that when the allowable DM is sufficiently large, we should first do ARQ. After $N_{\text{TB}} - 2$ 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 $0.62 \leq \rho \leq 0.75$, the optimal burst lengths are $v_{N_{\text{TB}}} = 3$, and $v_n = 1$ for $n < N_{\text{TB}}$.

From (12) we see that it is straightforward to achieve DM values that are integer multiples of $2d_c$. To achieve intermediate DM values, we shrink the wait time between the last burst and the SUA region by α time units, $0 \leq \alpha \leq \text{RTT}$, down to $\text{RTT} - \alpha$. All earlier wait times between bursts are still fixed at RTT. With this generalization, $\text{DM} \approx 2d_c N_{\text{TB}} - \alpha$, which can take on any non-negative integer value. For TM, the induction in (13) remains the same except at the end when there is only one burst left. With $\tau(v_{N_{\text{TB}}}; \alpha)$ denoting the TM value associated with doing a single burst of length $v_{N_{\text{TB}}}$, it can be shown that

$$\tau(v_{N_{\text{TB}}}; \alpha) = \rho \cdot \max(v_{N_{\text{TB}}}, \alpha) + (1-\rho)^{\max(v_{N_{\text{TB}}}, \alpha) - \alpha}$$
$$+ (1-\rho)^{v_{N_{\text{TB}}}} (2d_c - \alpha)\rho. \qquad (15)$$

Eq. (15) suggests that the optimal $v_{N_{\text{TB}}}$ should be at least α so that the third term is minimized without affecting the first two terms. With $v_{N_{\text{TB}}} \geq \alpha$,

$$\tau(v_{N_{\text{TB}}}; \alpha) = \rho v_{N_{\text{TB}}} + (1-\rho)^{v_{N_{\text{TB}}}} \left((1-\rho)^{-\alpha} - \alpha\rho + 2d_c\rho \right). \qquad (16)$$

Similar to (14), the optimal $v_{N_{\text{TB}}}$ can be computed using $\lfloor \log_{1/(1-\rho)} \left((1-\rho)^{-\alpha} - \alpha\rho + 2d_c\rho \right) \rfloor + 1$. After computing the resulting minimum $\tau(v_{N_{\text{TB}}}; \alpha)$, and using it in place of $\tau(v_{N_{\text{TB}}})$, we obtain $v_{N_{\text{TB}}-1}, \ldots, v_1$ via the induction (14).

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 (DM, 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.

A fundamental property of the MBT protocol is that when a packet delivery gets close to its "deadline," more resources are spent to improve the chance of packet delivery—in particular, channel bandwidth resources. More generally, it is possible to
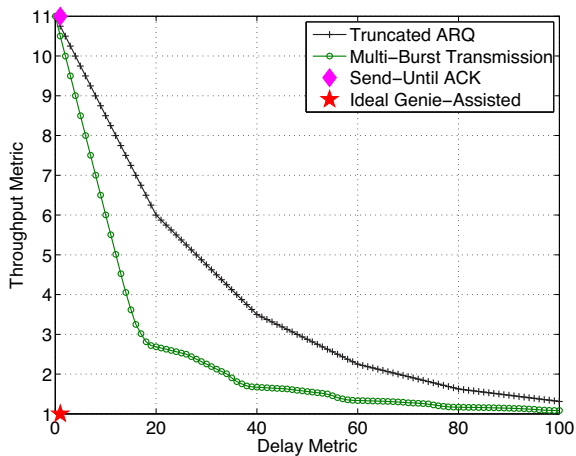
Fig. 3. The delay-throughput tradeoff for optimized MBT and truncated ARQ, compared to the genie-assisted bound and to the SUA scheme. ARQ performance is not shown, since it yields TM = 0 but DM = ∞. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.

apply this principle to other resources, such as transmission power or rate, which may be useful in other contexts, though we do not consider such extensions in this work.

A. Robustness of the MBT Solution

Optimization of the MBT burst-length vector $\mathbf{v}$ as developed above requires knowledge of the channel parameters ρ and d_c. In practice, these parameters may not be known accurately. Therefore, it is desirable to identify solutions that would achieve near-optimal performance for wide ranges of channel parameter values.

We first show that simply knowing whether the channel is lightly, moderately, or severely blocked allows us to choose a burst-length vector that is near-optimal for that operating environment. Fig.4(a) shows the TM achieved as a function of ρ for three 2-burst MBT schemes when $d_c = 10$. All 2-burst MBT schemes achieve about the same DM of $4d_c$, so the lower the TM the better. The black dashed curve at the bottom is the lowest TM achievable when $\mathbf{v}$ is optimized for each ρ assuming it is known exactly. The three burst-length choices, $\mathbf{v} = [1, 2]$ (blue +), $\mathbf{v} = [2, 4]$ (green ○), and $\mathbf{v} = [4, 8]$ (red ×), are optimal for $\rho = 0.8, 0.5,$ and 0.2, respectively, representing lightly, moderately, and severely blocked conditions. Fig.4(a) shows that each of the solutions is near-optimal over the broader (colored) range of ρ values for which it is intended. The worst case gap to the optimal TM is only 0.13.

Alternatively, if the operator expects ρ to be within a particular range, $\mathbf{v}$ may be optimized in a minimax sense, guaranteeing a worst-case performance. For example, if ρ is between 0.5 and 0.8, $\mathbf{v} = [1, 4]$ yields a sub-optimality in TM of no more than 0.19 in this range.

If a single solution were desired for all ρ from 0 to 1, for the case of $d_c = 10$, $\mathbf{v} = [2, 7]$ achieves TM ≤ 2 for all ρ, with TM = 2 at $\rho = 1$. While this solution has a wide range of applicability, it is highly suboptimal when ρ is close to 1. It is possible that when no prior information on ρ is available, a system may start operating with such a solution

and then dynamically switch to a better choice of $\mathbf{v}$ based on the measured erasure probability.

The optimal $\mathbf{v}$ generally depends on the channel delay d_c. Fig. 4(b) shows the performance of the three solutions when the channel delay changes from 10 to either 5 or 20. When the channel delay is smaller, lower TM can be achieved, as the $\tau(\varnothing) = 1 + 2d_c\rho$ term associated with the SUA region is smaller. When $d_c = 5$, the same three solutions still work rather well, achieving a maximum gap to the optimal TM of 0.07; although a slightly smaller gap of 0.05 is possible with shorter bursts of $\mathbf{v} = [1, 2], [2, 3],$ and $[3, 5]$. When $d_c = 20$, the maximum gap increased to nearly 0.5 at $\rho = 0.1$. A much smaller gap of 0.33 is achievable with longer bursts of $\mathbf{v} = [1, 3], [2, 5],$ and $[5, 10]$. Evidently, the solutions are more robust to realized values of d_c that are lower rather than higher than expected. Therefore, when optimizing $\mathbf{v}$, the designer may want to assume a value for d_c that is on the large side.

When there is uncertainty in both ρ and d_c, $\mathbf{v}$ may be optimized in a minimax sense with respect to both variables.

B. Optimality of the MBT Solution

Our family of MBT schemes is one of a number of approaches for scheduling retransmissions. In this section, we show that the MBT solution is near-optimal in terms of delay-throughput tradeoff in the class of all retransmission schemes with a *fixed* schedule, i.e., the same retransmission schedule is used for all packets.

Every retransmission scheme with a fixed retransmission schedule is characterized by a retransmission time vector $\mathbf{x} = [x_1, x_2, x_3, \cdots, x_{N_x}]$, with N_x non-negative strictly-increasing integer entries. Each packet p_k is first transmitted at time $k + x_1$, then retransmitted at times $k + x_2, k + x_3, \cdots, k + x_{N_x-1}$, and finally enters the SUA mode starting at time $k + x_{N_x}$. Retransmission stops when the acknowledgment is received. For the MBT example with $\mathbf{v} = [2, 4]$ and $d_c = 10$, we have $\mathbf{x} = [0, 1, 22, 23, 24, 25, 46]$.

To lower bound the TM achieved by any schedule, we consider a hypothetical *batch-feedback* model. We divide all transmissions into consecutive batches, each being RTT $= 2d_c + 1$ long, until the SUA region. For packet p_k, the nth batch is all its transmissions from time $k + (n-1) \cdot \text{RTT}$ to time $k + n \cdot \text{RTT} - 1$, for $n \leq \lceil x_{N_x}/\text{RTT} \rceil$. Let u_n denote the size of the nth batch and let $\mathbf{u}$ denote the batch length vector $\mathbf{u} = [u_1, u_2, u_3, \cdots, u_{N_u}]$ with length $N_u = \lceil x_{N_x}/\text{RTT} \rceil$. In the above example, $\mathbf{u} = [2, 4, 17]$.

In the batch-feedback model, the feedback corresponding to all u_n transmission in the nth batch become available at the transmitter at the end of the nth RTT, $k + n \cdot \text{RTT} - 1$. If *any* of the u_n transmissions is successful, the $(n + 1)$th batch does not need to take place; if *all* u_n transmissions are blocked, the entire $(n+1)$th batch must be transmitted. Since the batch-feedback model advances some of the feedback, allowing retransmissions to terminate earlier, it provides a lower bound to the throughput metric.

The TM achieved with batch-feedback can be evaluated in a manner similar to that used to obtain (13). Let $\gamma(\mathbf{u})$ be the TM associated with $\mathbf{u}$ using the batch-feedback model, we have

$$\gamma(\mathbf{u}) = \rho u_1 + (1 - \rho)^{u_1} \cdot \gamma(\mathbf{u}_2^{\text{end}}), \qquad (17)$$
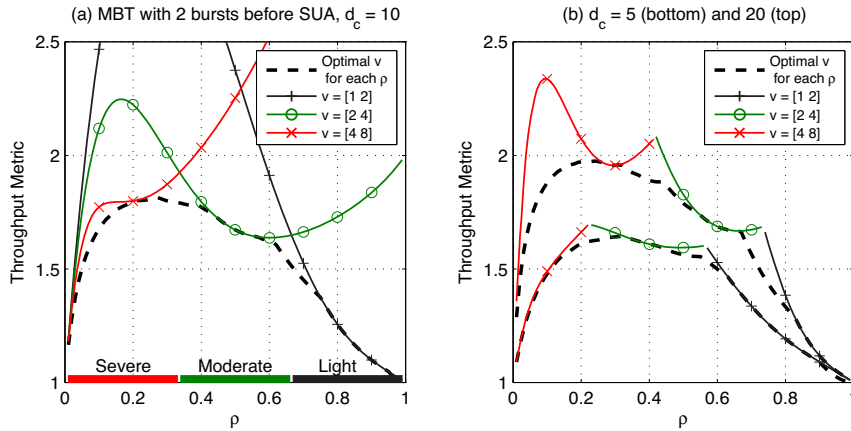
Fig. 4. Throughput metric (TM) achieved using three 2-burst MBT schemes intended for severe (red), moderate (green), and light (blue) blockage conditions for ρ from 0.01 to 0.99 and d_c being (a) 10 (b) 5 and 20.

with $\gamma(\varnothing) = 1 + 2d_c\rho$ corresponding to the SUA region after the last batch.

The DM achieved can be approximated using the conclusion from (11), i.e., that DM is approximately the total accumulated non-transmission time:

$$\text{DM} \approx \sum_{n=1}^{N_u}(\text{RTT} - u_n) = N_u\text{RTT} - \sum_{n=1}^{N_u} u_n. \quad (18)$$

Note that with the batch-feedback model, for both TM and DM, the exact locations of the u_n transmissions in the nth batch do not matter, only the number of transmissions matters.

To obtain the lower bound on the TM achievable at a particular DM by any retransmission scheme with a fixed schedule, we need to minimize $\gamma(\mathbf{u})$ subject to $N_u\text{RTT} - \sum_{n=1}^{N_u} u_n = \text{DM}$. The minimization is over N_u, and all elements of $\mathbf{u} = [u_1, \ldots, u_{N_u}]$, each of which is an integer between 0 and RTT, inclusive.

This search space can be reduced. First, the optimal $\mathbf{u}$ should not have any zero elements: removal of any zero element in $\mathbf{u}$ reduces DM by RTT while TM remains the same. Consequently, each element of an optimal $\mathbf{u}$ must be at least 1, and $N_u \leq \text{DM}$.

To further limit the search space, we note that to reduce TM the elements of an optimal $\mathbf{u}$ must be nondecreasing, i.e., the batches should not shrink. More specifically, if there are two neighboring elements where $u_i > u_{i+1} > 0$, then swapping them strictly reduces TM while maintaining DM. This can be shown by combining two levels of the induction in (17) and then showing $(1 - (1 - \rho)^u)/u$ decreases for $u > 0$

We perform the minimization of $\gamma(\mathbf{u})$ via exhaustive search for each value of DM. In Fig. 5(a), the resulting inner-bound in delay-throughput tradeoff (red ∗) is compared with the tradeoff achieved by the MBT scheme (green ○), for the case of $\rho = 0.5$ and $d_c = 10$. As the figure shows, the MBT scheme is at least near-optimal for the given channel parameters.[5]

In general, the gap between the MBT scheme and the batch-feedback inner bound increases with decreasing ρ and
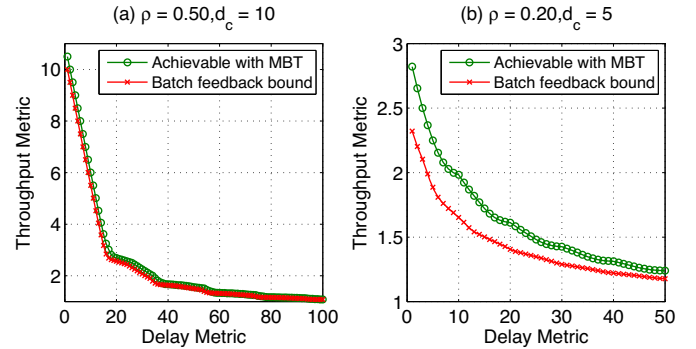

Fig. 5. The delay-throughput tradeoff achieved by the MBT scheme compared to the batch-feedback inner bound achievable by any retransmission scheme with a fixed schedule. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.

decreasing d_c. When the channel blockage is more severe, longer bursts are needed to improve the probability of a successful burst. When the burst length becomes comparable to the round trip time, the batch-feedback model deviates substantially from the true model. Fig. 5(b) shows the results for the $\rho = 0.2$ and $d_c = 5$ case. The gap is significantly wider. Furthermore, we identified specific instances where the MBT solution is indeed not optimal, although the suboptimality (in TM) is very small.[6]

V. CODED PROTOCOLS

In this section we suggest simple coded enhancements to the preceding retransmission protocols. In the schemes we consider, encoding is accomplished by computing random linear combinations of packets. In particular, at its simplest, any basic repetition protocol (see Sections III and IV) can be augmented with coding, as follows.

At each time where the scheme calls for transmission of packet p_k, it is replaced by a linear combination y_k of all packets p_j, $j \leq k$ (which has size R, same as p_k). We draw the

---

[5]In this example, we observe that the optimal MBT solution $\mathbf{v}$ often matches the optimal batch-feedback solution $\mathbf{u}$; it appears that for most values of DM, the MBT solution is indeed optimal, the gap is due to the batch-feedback model being overly optimistic in computing TM.

[6]For the case of $\rho = 0.2$, $d_c = 5$, target DM = 15, the MBT solution, burst 4 - wait 10 - burst 7 - wait 5 - SUA, achieves TM = 1.790; an alternative solution, burst 4 - wait 9 - burst 7 - wait 6 - SUA, achieves TM = 1.788.

combining weights in an i.i.d. manner for each transmission (thus also independent between replacements of the same p_k). We take the coefficient alphabet to be large with respect to the number of participating packets. This number, in turn, does not have to be as large as the stream duration k: packets that the transmitter knows that were already decoded may be excluded without affecting performance, thus the number of packets that should be considered grows at most as D_k, which we know that can be made to grow logarithmically. Under these assumptions, a packet k can be decoded as soon as for some $k' \geq k$, k' independent linear combinations that are function of packets up to k' arrive. The random combinations can be replaced by a structured code; see [11].

The performance of any such coded scheme is always at least as good as that of its base retransmission protocol. Indeed, since by our definition of delay a packet must wait for all previous packets, the linear combining can always be undone as part of the processing. 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 a detailed analysis of this coding gain is difficult, the gains can be quantified by simulation. To that end, 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 IV with $\mathbf{v} = [2, 4]$. For each scenario, simulations were performed using up to $k = 10^5$ 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. Finally, TM is obtained by counting packets received.

Fig. 6(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 curves represent $E[D_k]$ achieved by the uncoded and coded schemes. They both have the same limiting slope as the dotted curve. The solid curves 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 ($\times$) curves are noticeably lower than the blue ($+$) ones, indicating the advantage of coding at all values of k.

Fig. 6(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, when $1TU = 20ms$, $k = 10^5$ 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 noticeable to end users.

Fig. 6(b) plots TM_k as a function of k together with the one standard deviation spread (dotted curves). 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, closing about a third of the gap 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 II shows the DM and TM achieved by a range of MBT retransmission protocols and their coded counterparts.

TABLE II
DM AND TM ACHIEVED FOR MEMORYLESS CHANNEL WITH $\rho = 0.5$

	Retransmission Analytical		Retransmission Simulation		Coded Simulation	
$\mathbf{v}$	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

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 the range of $\Delta(0.5) = 1.67$ above the analytical values.

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.

VI. PRACTICAL CONSIDERATIONS

We now consider some variations of the idealized model and metrics defined in Section II, in order to account for some issues encountered in practice. Specifically, we concentrate on cases where the instantaneous bandwidth is constrained, the delay is constrained (thus packet dropping is allowed), or both. The MBT approach has good performance under these conditions as well, as we verify empirically.

A. MBT with Limited Peak Bandwidth

As mentioned earlier, MBT consumes data bandwidth in a dynamic way, sending more retransmissions when the channel is blocked more frequently and less when the channel is better. In the baseline version of the scheme discussed so far, while we aim to reduce the average bandwidth consumption, there is no peak bandwidth constraint. However, in reality, a transmitter can only transmit a certain maximum number of packets each time unit. In this section, we first show the distribution of the number of packets being transmitted at each time unit in the baseline MBT, then we modify the protocol to work with the peak bandwidth constraint and evaluate its performance.

Fig. 7(a) shows the histogram of the number of packets sent each time unit for the baseline MBT with $\mathbf{v} = [2, 4]$, $d_c = 10$ and $\rho = 0.5$, using about 10^6 samples. The mean of the empirical distribution is 3.34, in agreement with the theoretical value $TM/\rho = 1.67/0.5$, the expected number of times each packet is transmitted. Since every packet is always first transmitted twice, the minimum number of packets transmitted each time unit is at least 2. We see that 2 and 3 are common, and beyond that the histogram decays quickly. The probability for sending over 4 packets at a time is less than $1/6$, and that for sending over 8 is only 0.005. The low
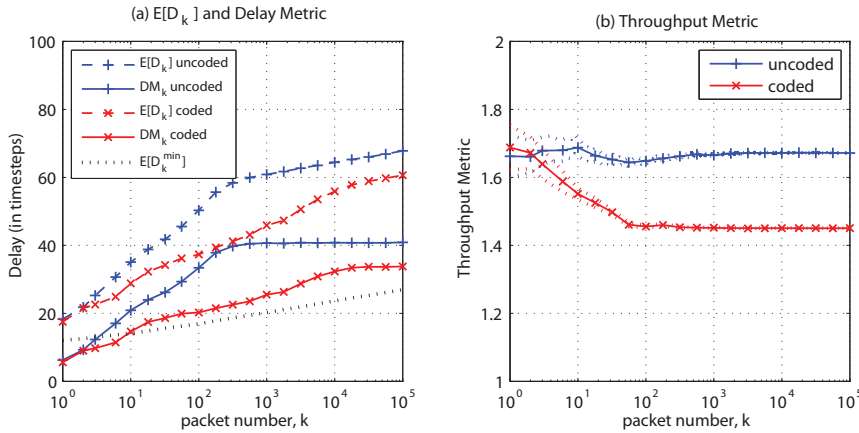
Fig. 6. The delay and throughput performance achieved by a two-burst MBT protocol with $\mathbf{v} = [2, 4]$, as a function of the packet index k, with and without coding. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.
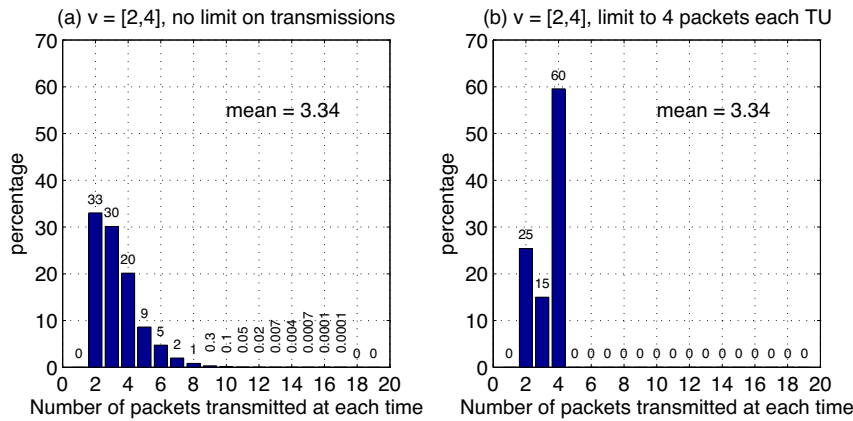

Fig. 7. Histogram of the number of packets sent each time unit for MBT with $\mathbf{v} = [2, 4]$. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$.

probability at high instantaneous load motivates the following simple limited-bandwidth modification to the baseline MBT.

First, all packet transmissions are scheduled according to the baseline MBT. Suppose only $N_{\max}$ packets can be transmitted each time unit. When the baseline MBT calls for more than $N_{\max}$ packets to be transmitted at a certain time, only the $N_{\max}$ oldest packets (with the smallest packet IDs) are transmitted. The remaining relatively newer packets are postponed to the next time unit. A packet can be postponed as many times as needed until transmission can be performed or the packet is acknowledged. When postponing a particular packet, we must postpone all the future scheduled transmissions of that packet. This ensures that the wait time between transmissions is long enough. For example, for $\mathbf{v} = [2, 4]$, $d_c = 10$, p_{100} is scheduled to be transmitted at times 100, 101, 122, 123, 124, etc.. Suppose after the initial transmission, the retransmission at 101 cannot take place, then the new future retransmission schedule for p_{100} becomes 102, 123, 124, 125, etc.

In order to evaluate the performance of this modified MBT protocol for limited peak bandwidth, we perform simulation for MBT with $\mathbf{v} = [2, 4]$, $d_c = 10$ and $\rho = 0.5$, with $N_{\max} = 8$ and 4. The resulting histogram for the limit-8 case is nearly identical to the unconstrained case except without the tail beyond 8 and a slightly increased value at 8. The limit-4 histogram is shown in Fig 7(b), the probability of
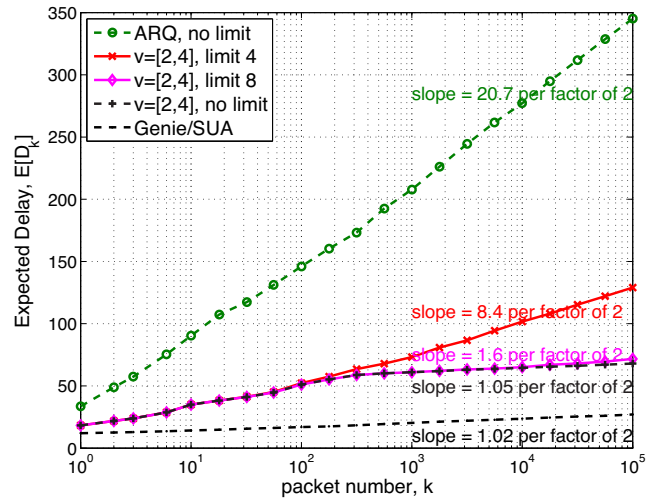

Fig. 8. Expected delay as a function of packet number, k, for ARQ and MBT with various bandwidth limits, as well as the optimal genie/SUA performance. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$. Slopes are measured from $k = 10^3$ to $k = 10^5$.

using the maximum load is 0.6, i.e., the bandwidth constraint is significant. In both cases, the mean of the distribution remain as in the unconstrained case, as almost all postponed transmissions are eventually performed.

Fig. 8 compares the delay performance of various scenarios. The three dashed curves correspond to scenarios with no bandwidth limit, which confirm our earlier results. In particular, the genie/SUA and MBT cases (bottom two) have slopes of approximately 1 per factor of $1/(1-\rho) = 2$ as shown in (7), while the slope in the ARQ case (uppermost curve) is around RTT as discussed in Section III-B. The two solid curves show the impact of bandwidth limit. While the limit-8 case is only slightly worse than the unconstrained case, the limit-4 delay starts to deviate around $k = 200$ and shows an empirical slope around 8. Although the degradation is significant, the delay is still much less than the ARQ case. We observe that one significant contribution to the increased delay is when, for example, 5 packets enter the SUA stage, but only 4 can be transmitted simultaneously, so the fifth packet has to be postponed nearly a RTT.

B. MBT with Maximum Delay

The MBT scheme discussed so far attempts to deliver all packets, where a single packet that is not received may delay the playback indefinitely. In practice, for real-time applications, after a packet has been delayed beyond a certain threshold, it is no longer useful and should be considered lost.

In this section, we propose a variation on the MBT protocol that operates within a prescribed maximum delay constraint μ while maintaining a packet loss rate ϵ. This maximum delay requirement differs from DM in that it is the maximum rather than the expected delay, so users have guaranteed performance. Also, the delay requirement does not grow with the message length, in contrast to the expected delay growth experienced by the genie-assisted scheme.

The variation we suggest is "truncated MBT", i.e., retransmission of a packet stops after a pre-determined number of bursts. Accordingly, similar to the original MBT scheme, truncated MBT is characterized by the burst length vector $\mathbf{v} = [v_1, v_2, \cdots, v_{N_{\mathrm{TB}}}]$, except that the bursts are *not* followed by an SUA period. Since each packet may be transmitted up to $S_v \triangleq \sum_{i=1}^{i=N_{\mathrm{TB}}} v_i$ times, the packet loss rate achieved is $\epsilon_a = (1-\rho)^{S_v}$. Since the last retransmission of p_k occurs at $k + S_v + 2d_c(N_{\mathrm{TB}} - 1) - 1$, the maximum packet delay achieved is $\mu_a = S_v + 2d_c(N_{\mathrm{TB}} - 1) + d_c$. The receiver would wait for packet p_k up to time $k + \mu_a$ and declare it lost.

Therefore, given a pair of required μ and ϵ, we can set the sum and length of $\mathbf{v}$ as follows:

$$S_v = \lceil \log_{1-\rho} \epsilon \rceil, \qquad (19)$$

$$N_{\mathrm{TB}} = \left\lfloor \frac{\mu + d_c - S_v}{2d_c} \right\rfloor. \qquad (20)$$

Of all such vectors $\mathbf{v}$, the one that leads to minimum TM is desired. The computation of TM for a given $\mathbf{v}$ can be performed using the recursion (13), with initialization $\tau(\varnothing) = 0$ (reflecting no SUA region). However, due to the additional constraint on the sum of $\mathbf{v}$, the optimization is more complicated than that in Section IV. Nevertheless, a numerical search may be performed when N_{TB} is sufficiently small.

Consider the channel with $d_c = 10$ and $\rho = 0.5$, used in earlier examples. Suppose the requirements are $\mu = 40 \approx 2\,\mathrm{RTT}$ and $\epsilon = 0.02$. Using (19) and (20), we obtain $S_v =$

6 and $N_{\mathrm{TB}} = 2$. Numerical optimization yields $\mathbf{v} = [2, 4]$, and the resulting TM is 1.5. This truncated MBT can achieve $\mu_a = 36 < 40$ and $\epsilon_a = 0.5^6 = 0.016$. We thus see that in this channel, which is too severe for ARQ schemes to handle, the MBT scheme is able to achieve a reasonable set of delay, packet loss rate, and average bandwidth consumption.

C. MBT with Limited Peak Bandwidth and Maximum Delay

The case where there are both limited peak bandwidth and maximum delay constraints can be treated by implementing both techniques presented in the previous two sections, i.e., transmit-older postpone-newer and truncated-MBT. Due to the combination it is possible that some retransmissions for packet p_k may be postponed beyond $k + \mu_a$ so that the receiver would not wait for it. To save bandwidth, the transmitter does not repeat p_k after time $k + \mu_a - d_c - 1$. As some packets may have fewer than S_v chances to go through the channel, the resulting packet loss rate ϵ_a may increase. If μ_a no longer meets the requirement, the designer may choose to perform a combination of: 1) increase the maximum delay μ_a to allow some transmissions to be postponed a few times before declared lost; 2) increase the peak bandwidth constraint $N_{\max}$ so fewer transmissions need to be postponed; or 3) increase S_v (and TM) so each packet may be transmitted more times. Essentially, the designer's task is to jointly minimize μ_a, ϵ_a, $N_{\max}$ and TM.

Continuing with the example of the last section, Fig. 9 shows the expected delay and packet loss rate as functions of the message length k. The solid curve shows that without a peak bandwidth constraint, $\mu_a = 36$ and $\epsilon_a = 0.016$ are achieved as calculated in Section VI-B. Since the average bandwidth requirement is $\mathrm{TM}/\rho = 1.5/0.5 = 3$, the designer may choose a peak bandwidth constraint of $N_{\max} = 4$. The dotted curve shows that with $N_{\max} = 4$ and $\mu_a = 36$, the packet loss rate rose to $\epsilon_a = 0.028$, which is higher than the requirement of 0.02. Noticing that the μ_a is still less than the required 40, the designer chooses option 1) and increases μ_a to 40 to allow some packets to be postponed up to four times and get S_v chances. The dashed curve shows that with $N_{\max} = 4$ and $\mu_a = 40$, ϵ_a is reduced to 0.018, which meets the requirement.

Obviously, the strategy presented here is not the only way for the receiver to decide to drop a packet (and for the transmitter to stop retransmitting). Nevertheless, it is one of the simplest.

VII. CONCLUDING REMARKS

We studied the problem of real-time streaming over blockage channel with long feedback delay. We showed that a practical multi-burst transmission 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. We showed that the MBT solution is robust to channel parameter knowledge inaccuracy and that it is also near optimal in terms of delay-throughput tradeoff in the class
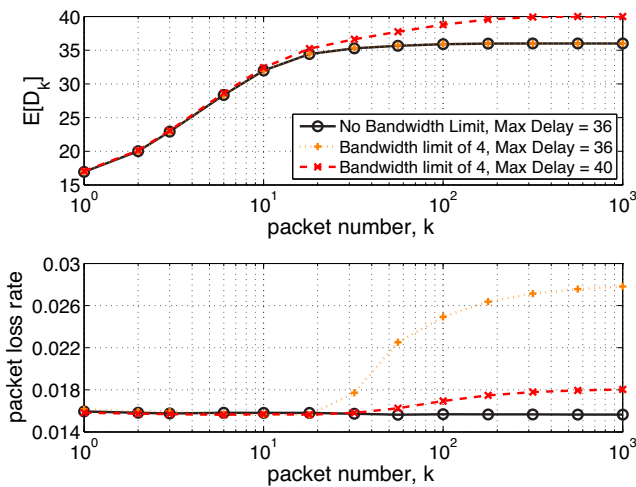
Fig. 9. Expected delay and packet loss rate as a function of the message length k with various peak bandwidth limit and maximum delay constraint. The channel is i.i.d. with $\rho = 0.5$ and $d_c = 10$. The MBT parameter are $\mathbf{v} = [2, 4]$.

of fixed-schedule retransmission schemes. The framework was also used to evaluate the benefit of coding via simulations. Coding improves both delay and throughput, especially in the low delay regime. We also show that the proposed MBT schemes may be modified to work with limited bandwidth and maximum delay constraints. While this study focused on the i.i.d. channel case, our initial results on the channel with memory case are reported in [10].

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, D. Costello, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-17, Dec. 1984.

[5] L. Shu, P. Yu, "A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels," *IEEE Trans. Commun.* vol. 30, pp. 1701-1719, Jul. 1982.

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

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

[8] H. Yao, Y. Kochman, G. W. Wornell, "Delay-Throughput Tradeoff for Streaming Over Blockage Channels with Delayed Feedback," in *MILCOM*, (San Jose, CA), Nov. 2010.

[9] H. Yao, Y. Kochman, G. W. Wornell, "On Delay in Real-Time Streaming Communication Systems, " in *Proc. Allerton Conf. Commun., Contr., Computing*, (Monticello, IL), Sep. 2010.

[10] H. Yao, Y. Kochman, G. W. Wornell, "A Retransmission Strategy for Real-Time Streaming Over Satellite in Blockage with Long Memory," submitted to *MILCOM*, (Baltimore, MD), Nov. 2011.

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

[12] 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.

[13] 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.
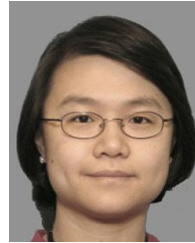
Huan Yao received the S.B. degrees in physics and electrical engineering, and the M.Eng. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1997, 1998, and 2003, respectively. Since 2003 she has been a member of the Technical Staff at the MIT Lincoln Laboratory, Lexington, MA, in the Advanced Satcom Systems and Operations Group in the Communication Systems and Cyber Security Division. She has worked on topics including dynamic resource allocation for future Satcom systems, waveform design and development, and blockage channel modeling. Her current research interests include wireless communication, ad hoc networks, and enabling communication and networking in blockage environments.

Yuval Kochman received his B.Sc., M.Sc., and Ph.D. degrees from Tel Aviv University, Israel, in 1993, 2003 and 2010, respectively, all in electrical engineering. Since 2009 he has been a postdoctoral associate in the Signals, Information and Algorithms Laboratory of the Research Laboratory of Electronics at the Massachusetts Institute of Technology (MIT). Outside academia, he has worked in the areas of radar and digital communications. His research interests include information theory, communications and signal processing.

Gregory W. Wornell (S'83-M'91-SM'00-F'04) received the B.A.Sc. degree from the University of British Columbia, Vancouver, BC, Canada, and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, all in electrical engineering and computer science, in 1985, 1987, and 1991, respectively.

Since 1991, he has been on the faculty at MIT, where he is Professor of Electrical Engineering and Computer Science, leads the Signals, Information, and Algorithms Laboratory in the Research Laboratory of Electronics, and is Co-Director of the Center for Wireless Networking. He has held visiting appointments at the former AT&T Bell Laboratories, Murray Hill, NJ, the University of California, Berkeley, CA, and Hewlett-Packard Laboratories, Palo Alto, CA. His research interests and publications span the areas of signal processing, digital communication, and information theory, and include algorithms and architectures for wireless and sensor networks, broadband systems, and multimedia environments.

Dr. Wornell has been involved in the Information Theory and Signal Processing Societies of the IEEE in a variety of capacities, and maintains a number of close industrial relationships and activities. He has won a number of awards for both his research and teaching.