RETRANSMISSION SCHEMES FOR METEOR-BURST COMMUNICATIONS

S. D. Sandberg
MIT Lincoln Laboratory
Lexington, MA 02173

M. B. Pursley
University of Illinois
Coordinated Science Laboratory
1101 W. Springfield, Urbana, IL 61801

ABSTRACT We examine the performance of various hybrid ARQ retransmission schemes for meteor-burst communications, including fixed-rate type-I hybrid ARQ and type-II hybrid ARQ. Another scheme, which we introduce here and refer to as variable-rate type-I hybrid ARQ, is designed to compensate for the variations in received power that occur on the meteor-burst channel. For variable-rate type-I hybrid ARQ, the code rate is allowed to decrease as the power received from the meteor trail decreases. For type-II hybrid ARQ, the code rate variation is inherent in the coding scheme: on the first transmission for a block, the code is effectively of high rate, but if an additional transmission is required, additional redundancy is sent, and the code rate is effectively lower for each subsequent transmission. The performance measure is the throughput per trial, which is defined to be the expected number of successfully received information bits for a given meteor trail. It is shown that throughput is larger for type-II hybrid ARQ and variable-rate type-I hybrid than for fixed-rate type-I hybrid ARQ.

I. INTRODUCTION

We examine the performance of several different automatic-repeat-request (ARQ) protocols for meteor-burst communications. The general model we use for ARQ protocols is as follows. One of the terminals, the source, has a message to send to another terminal, the destination. Upon detection of the probe, the source sends a sequence of code-words to the destination, and the destination attempts a decoding for each received word. In the basic ARQ scheme, the code is used for error detection only; no attempt is made to correct errors in the received word. If the destination detects no errors in a received word, it makes the source aware of this fact by sending an acknowledgement (ACK) on the return channel. If the destination determines that there are one or more errors in the received word, it sends a negative acknowledgement (NACK) to the source, and the source retransmits the identical codeword.

Basic ARQ has been considered for meteor-burst communications in [7] and [1]. We consider a form of ARQ known as hybrid ARQ, in which the code corrects some errors in the received word and detects most other commonly occurring errors. For hybrid ARQ, the destination first attempts to correct any errors in the received word. If it determines the error sequence is one that can not be corrected (this event is referred to as a decoding failure), it sends a NACK to the source. There are two types of hybrid ARQ schemes. For type-I hybrid ARQ, the source, upon receipt of a NACK, retransmits the same codeword as on the original transmission, and the destination attempts an independent decoding of the corresponding received word without making use of the original transmission (i.e., this attempt is made independently of any previous decoding attempts). For type-II hybrid ARQ [5], the source, upon receipt of a NACK, sends additional redundancy and the destination attempts another decoding using the additional redundancy along with the stored words from one or more previous transmissions for that codeword.

We introduce a variation of type-I hybrid ARQ that is tailored for the meteor-burst channel, and refer to this scheme as variable-rate type-I hybrid ARQ. The power received from a meteor trail generally decreases with time. For most trails, the power decreases throughout the lifetime of the trail, and for others, the power decreases after the trail has diffused sufficiently. Since the signal power is relatively large early in the trail, symbol errors occur infrequently there, and it is not necessary to include much redundancy in the codeword for transmissions during this part of the trail. The use of high-rate codes during the early part of the trail will provide a high information rate and lead to increased throughput. As the signal power decreases, symbol errors occur more frequently, and more redundancy is needed to correct the errors that will occur. The use of a high-rate code results in a high information rate during the early stages of the trail, but the use of a lower-rate code increases the time the trail is useful. Maximum throughput is achieved by use of a variable-rate coding scheme in which the code rate is decreased as the trail decays.

For one implementation of variable-rate type-I hybrid ARQ, referred to as protocol A, the source periodically obtains estimates of the instantaneous signal power at the destination, and it uses these estimates to select the rate of the code. This implementation also requires real-time coordination between the source and destination for the selection of the code rate and the determination of the times that changes in the code rate become effective. The throughput provided by protocol A when a particular heuristic is employed in selecting the code rate is derived in [10]. For an alternative approach, referred to as protocol B, the code rate is determined completely by the successes and failures of previous transmissions. As a result, the information for controlling the code rate passes from the destination to the source through the sequence of ACK's and NACK's, and no additional coor-
of the trail parameters, which the source uses to select a throughput. A derivation of the throughput for optimal throughput for protocol variable-rate type-I hybrid ARQ. For this scheme, it is assumed that the source and destination have perfect knowledge of the parameters, which the source uses to select a sequence of codes that gives the maximum possible throughput. A derivation of the throughput for optimal variable-rate type-I hybrid ARQ is given in [10].

The hybrid ARQ schemes we consider are of the selective-repeat variety [2,4]. For selective-repeat hybrid ARQ, the source transmits a continuous stream of data blocks. A data block is a codeword in type-I hybrid ARQ, but it may be a set of additional parity check symbols for a codeword in type-II hybrid ARQ. An additional transmission is made for a data block only if the previous transmission for that data block resulted in a decoding failure. The destination must have a buffer in which it can store received blocks until they are correctly decoded. The analysis presented here, we assume the buffer has infinite capacity. Variations of selective-repeat hybrid ARQ have been developed for buffers with finite capacity, and these perform nearly as well as selective-repeat hybrid ARQ with an infinite buffer [4]. We believe that the schemes developed in this paper can be adapted to work with finite buffer capacity without much degradation in performance. Other common forms of hybrid ARQ, such as stop-and-wait hybrid ARQ and go-back-N hybrid ARQ, are not considered here, because they are not as efficient as selective-repeat for applications with relatively high data rates and long acknowledgement delays, and these conditions are prevalent in meteor-burst communications.

II. PRELIMINARIES

A meteor trail is categorized according to its electron line density as either overdense or underdense. The electron density for overdense trails is large enough that the radio wave does not enter the interior of the trail, but is reflected from its surface. Underdense trails have an electron density small enough that the radio wave penetrates the surface and is reflected from each individual electron within the trail. As the underdense trail expands, the resulting multipath causes an exponential decay of the amplitude of the received radio wave. The majority of meteor trails are underdense, and the quantitative results of this paper are for underdense trails only. However, the results for underdense trails allow us to comment on the performance of the various schemes for overdense trails as well.

An \((n,k)\) singly extended Reed-Solomon code is employed with bounded-distance decoding. Such a code has alphabet size \(n\), where \(n\) is an integral power of 2. The code can correct any set of \(e\) symbol errors and \(k\) erasures, provided that \(2e + k \leq n - k\). An \((n,k)\) Reed-Solomon code can be shortened to give an \((n-b,k-b)\) code which corrects the same number of errors and erasures as the original code. For a Reed-Solomon code with sufficient redundancy, the probability of decoding into an incorrect codeword is much lower than the probability that the codeword simply fails to decode [6]. This fact, coupled with the possible use of a high-rate error detecting code for extra protection, makes it reasonable to ignore the possibility of an undetected error in a decoded block. For our analysis, it is assumed that if \(2e + k > n - k\), the destination determines this, declares a decoding failure, and sends a NACK to the source. It is also assumed that each acknowledgement is received without error at the source.

The signaling scheme is binary phase-shift-keying (PSK), and the \(n\)-ary symbols are sequences of \(m = \log_2 n\) binary digits. The channel noise is white Gaussian noise with two-sided power spectral density \(N_0/2\). Suppose an \(n\)-ary symbol is represented by the binary symbols \(1,2,\ldots,m\). An error occurs for the \(n\)-ary symbol only if one or more errors occur for the binary symbols \(1,2,\ldots,m\). Let \(\sigma_i\) be the probability that binary symbol \(i\) is transmitted in error. Since the noise is white and Gaussian, binary symbol errors are independent, and the probability of error for the \(n\)-ary symbol is \(1 - \prod_{i=1}^{m}(1-\sigma_i)\).

Define the time origin to be the instant reception of the first block begins. Let \(R_0\) be the rate at which binary symbols are transmitted, in binary symbols per second. Define \(\Gamma(t)\) to be the ratio of the energy received from the trail during the time interval \((t,t+1/R_0)\), divided by \(N_0\). Let \(\Gamma_0 = \Gamma(0)\), and let \(t\) be the exponential amplitude decay constant for the underdense trail. It can be shown that

\[
\Gamma(t) = \Gamma_0 \exp\left(-2\pi tR_0\right). \tag{1}
\]

The quantity \(\Gamma(t)\) is the binary symbol energy to noise power spectral density ratio, for a binary symbol reception beginning at time \(t\). We refer to \(\Gamma(t)\) as the instantaneous signal-to-noise ratio, and to \(\Gamma_0\) as the initial signal-to-noise ratio. The probability of error for a \(n\)-ary symbol received at time \(t\), can be calculated by using (1) in conjunction with well established expressions for the \(\sigma_i\). For each of the protocols we discuss, it is assumed transmission is halted for a trail following the transmission of a block during which \(\Gamma(t) = \eta\). Let \(T\) represent the length of time for which \(\Gamma(t) > \eta\). Notice from (1) that \(T = (\pi/2) \ln(\Gamma_0/\eta)\).

III. DESCRIPTION OF THE HYBRID ARQ SCHEMES

The particular implementation that is considered here for type-II hybrid ARQ with Reed-Solomon coding is similar to that described in [9]. In the first transmission for a given codeword from the \((n,k)\) code, the source sends \(\eta\) symbols, \(\eta \geq k\), of the codeword. The destination demodulates these \(\eta\) symbols, and it simply inserts erasures in the remaining \(n-\eta\) positions to fill out the block. The full sequence of \(n\) symbols is then decoded as if it were a sequence received from
the transmission of the entire codeword. The destination can decode this codeword correctly if the number of symbol errors does not exceed \(e_1 = \lfloor (\eta - k)2 \rfloor \). If the transmission decodes correctly, the destination sends an ACK to the source, but if the transmission results in a decoding failure, the destination sends a NACK. The acknowledgement arrives at the source \(T_d\) seconds following the end of the corresponding transmission. The time \(T_d\) accounts for the receiver processing time and for the round-trip propagation time from the source to the destination via the meteor trail. For most applications, the height of a trail is a small fraction of the direct path distance between the source and the destination, so the variations in trail height are expected to result in negligible variations in \(T_d\).

Upon receiving a NACK, the source first finishes its current transmission, then, for the following transmission, it sends \(\eta\) different symbols from the original codeword. Note that the source makes \(D = [R_q T_q/m \eta] \) transmissions for other codewords before it receives the NACK and sends the second transmission for the original codeword. The destination attempts a decoding for the second transmission, using those symbols received in the first transmission along with those received in the second, and inserting erasures in the remaining \(n - 2\eta\) symbols. The decoder is able to decode successfully if the total number of errors for the two transmissions does not exceed \(e_2 = \lfloor (2\eta - k)2 \rfloor \). The source continues sending increments of redundancy in this manner until either a successful decoding results, or a total of \(\omega\) transmissions, \(\omega \leq n/\eta\), have been made for the codeword without a successful decoding. After receiving the \(i\)-th transmission for a codeword, the decoder is able to decode successfully if the total number of errors for the \(i\) transmissions does not exceed \(e_i = \lfloor (i \eta - k)2 \rfloor \), \(i = 1, \ldots, \omega\).

To simplify the analysis, we assume that if each of \(\omega\) transmissions for a codeword results in a decoding failure, the destination discards the symbols received for the first \(\omega\) transmissions, and the transmission process for this codeword begins anew. That is, after it receives the NACK for the \(\omega\)-th transmission, the source again sends the first \(\eta\) symbols from the codeword, and the decoder attempts another decoding, without incorporating any of the previous transmissions. For the purpose of analysis, the repetition of the initial transmission is considered an initial transmission for a new codeword.

It is not necessary to discard the received symbols when the first \(\omega\) transmissions do not produce a successful decoding. Some improvement would result by combining [3] transmissions following the \(\omega\)-th with the first \(\omega\) transmissions, but it is felt that this would contribute little to the throughput of a meteor-burst system. Greater improvement could be obtained by using incremental redundancy with a larger codeword length, and a larger value for \(\omega\), so that more transmissions are made before discarding the received symbols. Even without increasing the codeword length, it is possible to improve the throughput by limiting the second and all subsequent transmissions to \(\alpha \eta\) symbols. The decoder attempts a decoding for the second transmission, using the \(\alpha\) symbols along with the \(\eta\) symbols received for the first transmission. If the second transmission also results in a decoding failure, additional redundancy is sent and another decoding is attempted. The process continues until a successful decoding occurs or all \(n\) symbols from the codeword have been transmitted. The goal is to send enough redundancy that a decoding success occurs, but not so much redundancy that the overall information rate is too low.

For fixed-rate type-I hybrid ARQ, the code is an \((\eta_1, k)\) code. Consequently, the code can correct as many as \(e_1\) errors on the original transmission. If a transmission results in a decoding failure, the destination sends a NACK to the source, and the source retransmits the identical codeword. The decoder then attempts another decoding, and this decoding is independent of the previous reception (i.e., it does not use any of the previously received symbols). As with type-II hybrid ARQ, the source makes transmissions for \(D\) other codewords, between transmissions for this codeword. In order to make valid comparisons between the type-I and type-II hybrids, it is necessary that the symbol alphabet size be the same. Thus, the \((\eta_1, k)\) code used in the type-I hybrid scheme has an alphabet of size \(n\), and the code itself is actually a shortened \((n, k+n-\eta)\) code.

For variable-rate type-I hybrid ARQ, a sequence of codes is used. The source transmits a codeword from the code \((u_1, k)\) for its \(i\)-th transmission. If the \(i\)-th transmission results in a decoding failure, the destination sends a NACK to the source, and the NACK is received during the transmission of another codeword. Suppose this transmission is the \((i+1)\)-th. The source encodes the \(k\) information symbols from the \(i\)-th transmission using the code \((u_{i+1}, k)\), and transmits this codeword for transmission \(i+1\). We use the notation \(u = (u_1, u_2, \ldots)\) to represent the sequence of codes used for a particular trail. A code can be used more than once, so the \(u_i\) are not necessarily distinct. The sequence \(u\) of codes is actually a sequence of shortened blocklength \(n\) codes, so the alphabet size is the same as for fixed-rate type-I hybrid ARQ and type-II hybrid ARQ. For protocol B, the code \((u_i, k)\) is restricted to be one of the \(L\) codes \((v_1, k), (v_2, k), \ldots, (v_L, k)\). The codes \((v_i, k)\) are used sequentially. That is, at the beginning of the trail, the source uses the code \((v_1, k)\), and the code \((v_i, k)\) is not used until all transmissions from the codes \((v_1, k), (v_2, k), \ldots, (v_{i-1}, k)\) that will be made, have been made. It is not necessary that the entire sequence of \(L\) codes be used; in fact, the \((v_{L-1}, k)\) code may be the only code used before transmission is halted for the trail.

Protocol B proceeds, from the onset of the trail, the same as fixed-rate type-I hybrid ARQ, beginning with the \((v_1, k)\) code and continuing until the second NACK is received. When the source receives the second NACK, it completes the current transmission, then transmits a codeword from the \((v_{L-1}, k)\) code. This codeword is the one formed by encoding the \(k\) information symbols from the codeword associated with the NACK. The source continues to employ the code \((v_{L-1}, k)\), until the second NACK for a \((v_{L-1}, k)\) code word is received. The source then begins using the \((v_{L-1}, k)\) code. The procedure continues in this manner until either transmission for the trail is halted, or until the \((v_{L-1}, k)\) code is
used. Once the \( (v, k) \) code is used, transmission continues with this code until the trail ends.

We also propose a variation of protocol B. The variation is more difficult to analyze, and the results of Section V are for the protocol described above only. For the variation, the code is changed each time two NACK’s occur in a length of time smaller than some prearranged length. The prearranged length may correspond to the time required for some number of transmissions, and can be chosen to give an indication of how well the current code is performing. For protocol B, as described before, the code is changed after the second decoding failure, without regard to the number of successful transmissions that occur between the failures. As a result, a code may be changed even though it is performing well.

IV. THROUGHPUT FOR FIXED-RATE TYPE-I HYBRID ARQ AND TYPE-II HYBRID ARQ

For either fixed-rate type-I hybrid ARQ or type-II hybrid ARQ, the length of a block transmission is \( m \eta R_b \) seconds. As a result, there are a total of

\[
M = \left[ \frac{T_R}{m \eta} \right]
\]

transmissions for the trail. Let the random variable \( X_i \) denote the number of correctly decoded information bits for transmission \( i \). Of course, the distribution for \( X_i \) is different for fixed-rate type-I hybrid ARQ than for type-II hybrid ARQ. Let \( G(\tau, \Gamma_0) \) represent the throughput for a trail with parameters \( \tau \) and \( \Gamma_0 \). Recall that throughput is defined as the expected number of correctly decoded information bits for a trail, so, for either of the schemes,

\[
G(\tau, \Gamma_0) = \frac{1}{m} \sum_{i=1}^{M} E X_i.
\]

Note that \( X_i \) takes the value \( mk \) when transmission \( i \) results in a successful decoding, and it takes the value 0 when transmission \( i \) results in a decoding failure. Let \( P_i \) be the probability that transmission \( i \) results in a successful decoding, then \( E X_i = mkP_i \) and

\[
G(\tau, \Gamma_0) = \frac{1}{m} \sum_{i=1}^{M} P_i.
\]

Let \( Z_i \) denote the number of symbol errors that occur for the \( i \)-th transmission, and define the function \( f_i \) by

\[
f_i(\ell) = P(Z_i = \ell), \quad \ell = 0, 1, \ldots \end{equation}

Note that the \( j \)-th symbol in block transmission \( i \) is received at time \( m(i-1)\eta + j-1)/R_b \). The functions \( f_i \) can be generated from the symbol error probabilities using methods developed in [8] and [10]. In what follows, let \( P_i^{(1)} \) and \( P_i^{(\infty)} \) be the probability that transmission \( i \) results in a successful decoding, for fixed-rate type-I hybrid ARQ and type-II hybrid ARQ, respectively. Clearly,

\[
P_i^{(1)} = \sum_{\ell=0}^{\infty} f_i(\ell), \quad 1 \leq i \leq M.
\]

The calculation of \( P_i^{(\infty)} \) is more difficult than the calculation of \( P_i^{(1)} \). In particular, for type-II hybrid ARQ, if the \( i \)-th transmission for the trail is the \( j \)-th transmission for some codeword, then the probability that transmission \( i \) results in a successful decoding is a function of the number of symbol errors in the previous \( j-1 \) transmissions for that codeword. The previous \( j-1 \) transmissions for the codeword are transmissions \( i-(j-1)(D+1), i-(j-2)(D+1), \ldots, i-(D+1) \) for the trail.

For the derivation of an expression for \( P_i^{(\infty)} \), it is convenient to define the following events. For \( 1 \leq i \leq M \), let \( F_i^{(j)} \) be the event that transmission \( i \) is the \( j \)-th transmission for some codeword. A second transmission for a codeword can not be made until an amount of time equal that required for \( D \) transmissions passes, following the first transmission for that codeword. Consequently, the first \( D+1 \) transmissions for the trail must be initial transmissions for some codeword; that is,

\[
F_i^{(i)} = \begin{cases} \Omega, & i=1, \quad 1 \leq i \leq D+1, \\ \emptyset, & j \geq 1, \quad 1 \leq i \leq D+1, \end{cases}
\]

where \( \Omega \) is the set of all possible events, \( \emptyset \) is the empty set, and \( D+1 \leq M \). Extending this line of reasoning we can show that the \( i \)-th transmission for the trail can be the \( j \)-th transmission for some codeword only if \( j \leq \lfloor i/(D+1) \rfloor \). If, contrary to assumption, \( M < D+1 \), then each transmission for the trail is an initial transmission for some codeword, and

\[
P_i^{(\infty)} = P_i^{(1)}, \quad 1 \leq i \leq M.
\]

Define \( \sigma(i) \) as

\[
\sigma(i) = \min \{ \alpha, \lfloor i/(D+1) \rfloor \}.
\]

Notice that

\[
F_i^{(j)} = \emptyset, \quad j > \sigma(i),
\]

that is, it is not possible that the \( i \)-th transmission for the trail is the \( j \)-th transmission for some codeword if \( j > \sigma(i) \). Let \( E_i \) be the event that transmission \( i \) results in a successful decoding. It is easily shown that

\[
P_i^{(\infty)} = P(E_i) = \sum_{j=1}^{\sigma(i)} p_j^{(i)} \beta_j^{(i)},
\]

where \( p_j^{(i)} \) and \( \beta_j^{(i)} \) are defined as

\[
p_j^{(i)} = P(E_i | F_i^{(j)}), \quad j < \sigma(i),
\]

and

\[
\beta_j^{(i)} = P(F_i^{(j)}).
\]

We show in [10] that

\[
\beta_j^{(i)} = \begin{cases} 0, & \sigma(i) < j, \quad i \geq 1, \\ 1, & \sigma(i) \geq j, \quad i \geq 1, \end{cases}
\]

Define \( Y_j^{(i)} \) by

\[
Y_j^{(i)} = \sum_{i=1}^{j-1} Z_{i-(D+1)}, \quad j \leq \sigma(i).
\]
Notice that $Y^{(j)}_i$ is the total number of symbol errors for transmissions $i-(j-1)(D+1)$, $i-(j-2)(D+1), \ldots$, and $i$. The conditional probability of successful decoding for the $i$-th transmission, conditioned on the fact it is the $j$-th transmission for some codeword is the conditional probability that the total number of errors for the $j$ transmissions does not exceed $e_j$, i.e.,

$$p^{(j)}_i = \sum_{l=0}^{e_j} g^{(j)}_l(i), \quad j \leq \sigma(i),$$

(11)

where $g^{(j)}_l(i)$ is the conditional probability mass function for $Y^{(j)}_i$, defined as

$$g^{(j)}_l(i) = P(Y^{(j)}_i = l \mid F^{(j)}_i), \quad j \leq \sigma(i).$$

It is shown in [10] that

$$g^{(j)}_l(i) = f^{(j)}_i(l), \quad \forall \, l, \quad (12)$$

and

$$g^{(j)}_l(i) = (f^{(j)}_i h^{(j)}_{l-1})_i(l), \quad 1 < j \leq \sigma(i),$$

(13)

where "*" represents convolution. The function $h^{(j)}_l(i)$ takes the form

$$h^{(j)}_l(i) = \begin{cases} 
0, & 1 \leq e_j, \quad 1 \leq j \leq \sigma(i), \\
\sum_{l=0}^{e_j} g^{(j)}_l(i) (1-p^{(j)}_i), & l > e_j, \quad 1 \leq j \leq \sigma(i). 
\end{cases}$$

(14)

The analysis given in (3)-(14) can be used to develop a recursion for the values of $p^{(j)}_i$, $i=1,\ldots,M$, and these values can be used in (3) to evaluate throughput for type-II hybrid ARQ. Figure 4.1 in [10] illustrates a procedure that can be used to generate the values for $p^{(j)}_i$ recursively.

V. NUMERICAL RESULTS

Figures 1-6 display throughput as a function of $\Gamma_0$ for several selective-repeat hybrid ARQ schemes. The odd numbered figures are for $\tau=0.5$ s and the even numbered figures are for $\tau=0.1$ s. The nominal blocklength is $n=64$, so the code symbol alphabet size is 64. The bit rate is $R_b=64,000$ binary symbols per second, and the values assumed for $T_d$, $\gamma$, $\eta$, and $\omega$ are $10$ ms, $-10$ dB, 16, and 4, respectively. The direct path distance between the source and the destination is $1000$ km for the meteor-burst system described in [1]. For this source-destination pair, the round-trip propagation time from the source to the destination, via the meteor trail, is roughly $6.7$ ms. Hence, for this source-destination pair, the value $T_d=10$ ms is consistent with a receiver processing time of $3.3$ ms.

Figures 1 and 2 display the throughput for fixed-rate type-I hybrid ARQ and type-II hybrid ARQ for various values of $k$. For type-II hybrid ARQ, notice that when $k=\eta$ the code can not correct any errors in the first transmission for a codeword. For fixed-rate type-I hybrid ARQ, no errors can be corrected in any transmission, when $k=\eta$. Recall that we have assumed that a high-rate error-detecting code is used in conjunction with the Reed-Solomon code to detect the errors at the Reed-Solomon decoder output. Consequently, for $k=\eta$, fixed-rate type-I hybrid ARQ is equivalent to basic ARQ. We begin the discussion of Figs. 1 and 2 by considering the results for fixed-rate type-I hybrid ARQ. Note that for fixed-rate type-I hybrid ARQ, the (16,12) code gives largest throughput, among the codes considered, for values of $\Gamma_0$ between 6 dB and 9 dB, but the (16,14) code gives largest throughput for values of $\Gamma_0$ between 9 dB and 12 dB. This is true for both values of $\tau$.

Decoding successes are rare for codes of high rate during the later part of the trail, because the signal-to-noise ratio is relatively small. Hence, during the later part of the trail, the contribution to throughput is larger for codes of low rate than for codes of high rate; in addition, the trail is useful for a longer time for codes of low rate. During the portion of the trail when the signal-to-noise ratio is large, however, most transmissions are successful for any code rate, so a code of high rate makes a larger contribution to throughput than a code of low rate. Hence, when $\Gamma_0$ is small, the
throughput is dominated by the contributions from transmissions that are successful at low signal-to-noise ratios, and codes of low rate perform better than codes of high rate. However, when $\Gamma_0$ is large, the throughput is dominated by the contribution from the portion of the trail during which nearly all transmissions are successful, and the codes of high rate give larger throughput than the codes of low rate. We conjecture there is a $\gamma'$ such that basic ARQ provides larger throughput than fixed-rate type-I hybrid ARQ for all $\Gamma_0 \geq \gamma'$ and for any value of $k < 16$. However, it seems that such a $\gamma'$ is so large that $\Gamma_0$ will exceed $\gamma'$ for only a small fraction of the usable meteor trails.

For an arbitrary value for $k$, consider the portion of the trail for which $\Gamma(t)$ has decreased to such a small value that the probability of successful decoding for the $(16, k)$ code is small. Little additional throughput can be obtained from this part of the trail for fixed-rate type-I hybrid ARQ, since transmissions are rarely successful. Initial transmissions for type-II hybrid ARQ are also rarely successful for this part of the trail since, on the first transmission, the code is effectively a $(16, k)$ code. However, on the $i$-th transmission, $i \leq \tilde{a}$, the code is an $(i, \tilde{a}, k)$ code. As a result, additional throughput can be obtained from this part of the trail for type-II hybrid ARQ, and so the life of the trail is extended by using type-II hybrid ARQ.

Now consider that part of the trail for which $\Gamma(t)$ is large enough that the $(16, k)$ code rarely results in a decoding failure. The contribution to throughput from this part of the trail is the same for type-II hybrid ARQ as for fixed-rate type-I hybrid ARQ, because nearly all initial transmissions are successful for both schemes. This is reflected in Figs. 1 and 2, where it is seen that the difference between the throughput for type-II hybrid ARQ and that for fixed-rate type-I hybrid ARQ is constant for all values of $\Gamma_0$ exceeding roughly 7 dB. This is true for $k = 14$ and $k = 16$ and for both values of $r_\pi$. For signal-to-noise ratios exceeding 7 dB, the probability of successful decoding for the $(16, k)$ code is nearly equal to one. The increased throughput offered by type-II hybrid ARQ over fixed-rate type-I hybrid ARQ is due entirely to some portion of the trail for which $\Gamma(t) < 7$ dB. For $r_\pi = 0.5$, the increase in throughput is 6,500 bits per trail for $k = 16$, and 4,000 bits per trail for $k = 14$. The increase for $r_\pi = 0.1$ is one-fifth that for $r_\pi = 0.5$.

Figures 3 and 4 display throughput for optimal variable-rate type-I hybrid ARQ, fixed-rate type-I hybrid ARQ (for $k = 14$), and variable-rate type-I hybrid ARQ with protocol B. Recall that for optimal variable-rate type-I hybrid ARQ, it is assumed that $\tau$ and $\Gamma_0$ are known and the sequence of codes $u$ is selected so that the maximum possible throughput results. The code selected for a given transmission is constrained to be one of the codes $(u, 16)$, where $16 \leq u \leq 64$. Recall that for protocol B, the source uses sequentially the $L$ codes $(v_1, k), (v_2, k), \ldots , (v_L, k)$. The source employs the code $(v_1, k)$ at the onset of the trail, and the code $(v_1, k)$ is used only after the second NACK is received for the code $(v_{i-1}, k)$. The results of Figs. 3 and 4 are for $L = 4$, $k = 16$, $v_1 = 16$, $v_2 = 24$, $v_3 = 36$, and $v_4 = 64$. For protocol B, it is important that the rate of the first code is high. When $\Gamma_0$ is relatively large, very few decoding failures occur for the first part of the trail, no matter what the code rate is, so the contribution to the throughput for this part of the trail is largest for codes of high rate. For fixed-rate type-I hybrid ARQ, a penalty is incurred in the later stages of the trail if a code of high rate is used, because the contribution to throughput during the later stages of the trail is smaller for codes of high rate. For protocol B, however, the rate of the code is decreased as the signal-to-noise ratio decreases.

Protocol B seems to be fairly robust with respect to variations in the decay constant $\tau$. The percentage difference between the throughput for protocol B and the throughput for
optimal variable-rate type-I hybrid ARQ is nearly the same for $\tau = 0.1$ as for $\tau = 0.5$. This is a consequence of the fact that there is a substantial difference between the rates of any two consecutive codes. The rate of the code decreases by a fairly large value each time the code is changed. If the rate were decreased in small steps, the throughput would be larger, relative to the optimal throughput, for large values of $\tau$ than for small. A time $T_d$ passes between the time the source makes a transmission that results in the second decoding failure for the code $(v_i, k)$ and the time the source begins transmitting from the $(v_i+1, k)$ code. The trail decays to a current signal-to-noise ratio. Hence, if the code rate is larger, relative to the optimal throughput, for large values of $z = 0.1$ as for $z$, decoding failures occur for the code not substantially smaller than $(v_i, k)$. The process continues if $k/\nu_{i+1}$ is not substantially smaller than $k/\nu_i$, it is likely that two decoding failures occur for the code $(v_i, k)$ in rapid succession, and the code is changed to the $(v_i+1, k)$ code without obtaining much throughput from the $(v_i, k)$ code. The process continues if $k/\nu_{i+2}$ is not significantly smaller than $k/\nu_i$. As a result, a large amount of time is wasted in making transmissions for codes that are not well suited for the current signal-to-noise ratio. Hence, if the code rate is changed in small increments and $T_d$ is too large relative to $\tau$, it may not be possible for the protocol to change rates quickly enough to adapt properly to the decreasing signal strength.

Figures 5 and 6 display throughput for type-II hybrid ARQ (with $k = 16$), basic ARQ (equivalently, fixed-rate type-I hybrid ARQ with $k = 16$), and protocol B. Both type-II hybrid ARQ and protocol B benefit from the ability to change the code rate and adapt to the changing channel conditions, and these schemes give significantly larger throughput than basic ARQ. Type-II hybrid ARQ gives larger throughput than protocol B. For type-II hybrid ARQ with $k = 16$, the source begins transmission for a block of information symbols by using a code that has an effective rate equal to $1$, and if this transmission fails to decode, no penalty is paid for wasted time. This is because additional redundancy is sent so the code has an effective rate equal to $1/2$ for the second transmission, if the first transmission does not succeed. For type-I hybrid ARQ in general (and specifically for protocol B), the time for a transmission that fails to decode is wasted, but, for type-II hybrid ARQ, previous transmissions are incorporated in the decoding for a codeword, and no penalty is paid for first trying higher rates.

Note from Figs. 1-6 that, for each protocol, the throughput is a log-linear function of $\Gamma_0$, that is,

$$G(t, \Gamma_0) = a_1 + a_2 \ln(\Gamma_0).$$

for $\Gamma_0$ sufficiently large. The constants $a_1$ and $a_2$ in (15) depend on the particular protocol and the parameter values. The approximation (15) is exploited in [10] to obtain an approximation for average throughput for underdense trails. Average throughput is defined as the expected number of successfully received information bits for an underdense trail, and the expectation is taken by averaging over all possible values for $\Gamma_0$ and $\tau$, using probability density functions for these random variables that are adapted from [11]. Suppose the expected value of $\tau$ is 0.5 seconds and that the minimum value of $\Gamma_0$ required for synchronization is 8 dB. It is shown in [10] that, for the parameter values considered in this section, the average throughput for protocol B exceeds the average throughput for basic ARQ by 5,000 bits per trail. The average throughput for type-II hybrid ARQ exceeds average throughput for basic ARQ by 6,600 bits per trail.

VI. CONCLUSIONS

We have seen that the adaptive schemes (type-II hybrid ARQ and variable-rate type-I hybrid ARQ) perform at least as well as basic ARQ and fixed-rate type-I hybrid ARQ during the portion of a trail that the signal power is large. However, the contribution to throughput from the later part of the
underdense trail is always greater for the adaptive schemes than for basic ARQ and fixed-rate type-I hybrid ARQ, because for the adaptive schemes, the code rate is decreased as the signal strength decays. The use of an adaptive scheme extends the lifetime of the trail. The power received from an overdense trail increases rapidly during the early part of the trail, remains at a high level for a large portion of the trail, and then exhibits the same exponential decay as an underdense trail for the later part of the trail. Consequently, the adaptive schemes seem well suited for overdense trails as well, and it is expected that the additional throughput offered by the adaptive schemes (in bits per trail) is at least as large for overdense trails as for underdense trails. Both type-II hybrid ARQ and variable-rate type-I hybrid ARQ with protocol B are attractive from an implementation standpoint. Neither scheme requires measurements of the channel parameters to adapt the code rate. For either scheme, the information for controlling the code rate passes through the sequence of acknowledgements, and no additional coordination is required.

ACKNOWLEDGEMENT

This research was sponsored in part by the Joint Services Electronics Program under Contract N00014-84-C-0149 and in part by a subcontract from ITT Aerospace/Optical Division under Rome Air Development Center Contract F30602-89-C-0024.

REFERENCES