Coding Schemes for Multislot Messages in Multichannel ALOHA With Deadlines

Dror Baron, *Student Member, IEEE* and Yitzhak Birk, *Senior Member, IEEE*

***Abstract*—Slotted multichannel ALOHA is the access scheme of choice for short messages and for reserving channels for longer ones in many satellite-based networks. This paper proposes schemes for increasing the capacity (maximum attainable throughput) of multichannel slotted ALOHA subject to meeting a user-specified deadline with a (high) required probability, thereby jointly capturing the users' requirements and the system owner's desires. The focus is on short yet multislot messages. A key idea is to achieve a low probability of missing the deadline by permitting a large maximum resource expenditure per message, while holding the mean expenditure low in order to minimize "pollution." For a K-slot message, redundant single-slot fragments are constructed using block erasure-correcting codes, such that any K fragments suffice for message reception. With multiround coding, an optimized number of fragments are transmitted in each round until K are received or the deadline is reached. Even with very strict constraints, capacities that approach the $1/e$ limit are attained. The coding–reservation scheme raises capacity above $1/e$ by allowing the hub, upon receipt of any message fragment(s), to grant contention-free slots for the remaining required fragments. Both schemes are also adapted for use with single-transmitter stations at a small performance penalty in most cases. Finally, because capacity is maximized by minimizing the mean per-message transmission resources, the transmission scheme is also energy-efficient.**

***Index Terms*—Coding, deadline, delay, energy-efficient design, multichannel ALOHA, reservation ALOHA, satellite.**

I. Introduction

ALOHA [1] is the simplest access scheme because it does not require channel sensing or collision detection, but performs worse than more elaborate schemes when those are practical. An important use of ALOHA at present is for the transmissions of satellite ground stations, because the long propagation delay precludes timely channel sensing. It is used as the primary access scheme for short messages, and in order to reserve channels for long ones [2]. ALOHA is also used in some cellular networks, wherein the control up-link channels from the cellular phones to base stations are multiple access. A future application for ALOHA may be transmission of short messages over the upstream channel of high speed point-to-multipoint terrestrial wireless networks.

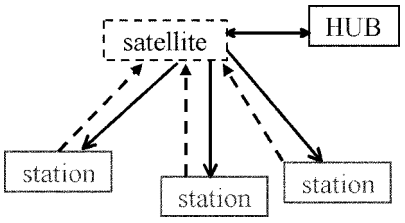

Fig. 1. A typical hub-based satellite network.

Fig. 1 depicts a typical satellite-based ALOHA network. The *stations* transmit data in globally synchronized time slots over contention up-link channels (dashed lines). Successful reception by the *hub* is acknowledged by it immediately over contention-free down-links (solid lines). The hub can be terrestrial or in space. If several simultaneous transmissions occur on the same channel, they all fail. Stations can only learn about a collision through the absence of an acknowledgment (ACK). Once a station learns that its transmission was not received, it retransmits after some delay. These transmission *rounds* are repeated until an ACK is received or the deadline is reached. While the results of this paper are also applicable, with little modification if any, to unslotted ALOHA networks, we restrict the discussion to slotted systems. We omit "slotted" for brevity, and use "classical ALOHA" to refer to slotted ALOHA with no particular optimizations.

In a single-channel ALOHA network, retransmission delay (upon collision) must be randomized to prevent definite repeated collisions [3]. To improve stability, a station must moreover increase the mean back-off time in later rounds. Current ALOHA satellite networks employ as many as hundreds of channels [4]. A station picks a channel at random for each transmission. The hub can receive concurrently over all channels, and the randomized retransmission delay is replaced with immediate retransmission over a randomly chosen channel.

Over the years, the bulk of the research on ALOHA and related reservation schemes, e.g., [5], concerned maximizing capacity. Some attention was given to delay-throughput trade-offs and other performance measures. The advent of multichannel ALOHA networks has given rise to the use of redundant transmissions for performance improvement. For example, [6] studies multicopy ALOHA, whereby a station transmits several *copies* of a packet in each round, as a way of improving delay–throughput performance. We refer to the transmission of multiple copies per round as "redundancy"

Manuscript received March 17, 2000; accepted July 2, 2001. The editor coordinating the review of this paper and approving it for publication is Bo Li. This work was supported in part by the Information Superhighway in Space Consortium, administered by the office of the Chief Scientist of the Israeli Ministry of Industry and Trade.

D. Baron was with the Electrical Engineering Department, Technion, Haifa 32000, Israel. He is now with the ECE Department, University of Illinois at Urbana-Champaign, Urbana, IL 61801-3028 USA (e-mail: dbaron@uiuc.edu).

Y. Birk is with the Electrical Engineering Department, Technion, Haifa 32000, Israel (e-mail: birk@ee.technion.ac.il).

Publisher Item Identifier S 1536-1276(02)02096-2.

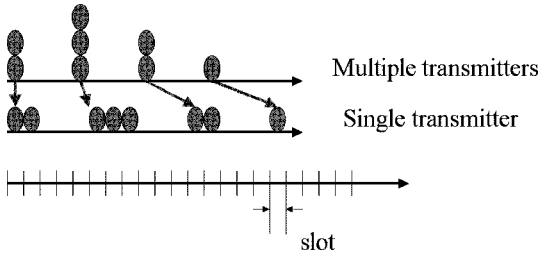

Fig. 2. Round stretching.

because, unlike retransmission upon failure, some of the transmissions may not be required.

Virtually all current applications of ALOHA entail the transmission of single-packet messages, be it for short transactions or in order to reserve channel resources for the transmission of large amounts of data. Also, the user is typically charged per actual traffic, while the system owner pays for bandwidth (channel) resources. From a user's perspective, the key performance criterion is delay, and it is most naturally expressed as a constraint (e.g., deadline). From the system owner's perspective, capacity maximization is the main design goal.

Recently, Birk and Keren [7] proposed an optimization problem that reflects both intuitive user requirements and the desires of network designers: maximization of capacity (the maximum attainable throughput) subject to a deadline and a permissible probability of exceeding it. We also use this performance measure. They proposed a *nonstationary* multicopy transmission policy, whereby a station transmits a monotonically nondecreasing number of copies in successive rounds until successful reception or deadline. Dynamic programming [8] was used to optimize the transmission sequence, resulting in a substantial increase in capacity relative to that of classical ALOHA or even that of (fixed) multicopy ALOHA [6]. The advantage is more pronounced for stricter constraints. They moreover adapted the optimized scheme to the practical situation wherein a station only has a single transmitter. This was done by transmitting a burst of copies in successive slots over randomly chosen channels, then waiting to learn the fates of all of them, and proceeding to the next round only if all copies failed. This technique, dubbed *round stretching*, was shown to achieve similar capacities to the multitransmitter scheme in most situations. Fig. 2 illustrates the idea. Note that, for any given deadline, round stretching may reduce the permissible number of rounds.

The main idea in the replication-based scheme of [7], which is employed in this paper as well, is to permit a large *maximum* per-message resource expenditure without substantially increasing the *average* expenditure. The large maximum expenditure attains a low probability of missing the deadline, while the low mean minimizes the resulting "pollution" that would act to reduce the attainable throughput. Noting that late rounds occur far less frequently than early ones (because transmission ceases upon successful reception), the foregoing goal is attained by spending more resources on a late-round transmission, thereby increasing the probability of success in such a round, than on an early-round transmission. In [7] and in this paper, the resource expenditure manifests itself as (speculative) redundant transmissions. Another (inferior) approach [9] is to partition the channels into groups, one per round, with lower offered loads ("working points") in the channels used for later rounds.

One can use "*pure*" multicopy policies, whereby the number of copies transmitted in any given round is deterministic (albeit not the same for all rounds), or "*impure*" policies whereby it is randomized. This idea is studied in [10] in the context of optimizing the throughput–delay trade-off with multicopy ALOHA. An impure variant of the replication-based scheme of [7] produces an insignificant increase in capacity [11].

The case of single-round transmissions, be it due to short deadlines or one-way communications, was also studied in [7]. The proposed solution was to chop a message into several fragments, use fragment-size slots, and combine header replication with erasure correcting codes [12] for the payload.

In this paper, we explore the use of erasure correcting codes for multislot messages. Our focus is on message lengths of a small number of slots, as very long messages should best be handled by reserving slots for their transmission. (The scheme of [7] can be used for making the reservations.) The design goal is to determine the optimal number of message fragments that should be transmitted in each round. (This number may well exceed the number of fragments that must still be received for the successful reception of the message.) The optimization is more difficult than for replication-based schemes, because the decision must also take into account the number of fragments that have already been received. We refer to the resulting scheme as *multiround coding.*

Upon reception of at least one fragment, the hub may allocate contention free slots for the transmission of the remaining fragments. Based on this, we propose a *coding–reservation* (C–R) scheme, whereby a first coding phase carries useful payload and also serves for making reservations, and a second phase handles the remaining fragments without contention. This is different from traditional reservation schemes, whereby the reservation-making phase does not carry payload. The performance of the new schemes is studied, and they are compared with replication-based schemes as well as with two traditional reservation schemes that use dedicated channels for making reservations. Another approach would include the first payload fragment with the reservation request. This approach, however, is always inferior to C–R, as will be explained later, and is not studied here.

The remainder of the paper is organized as follows. In Section II, we present the network model that is subsequently used for performance analysis, and derive some preliminary mathematical relations for use in later sections. Sections III and IV are devoted to the multiround coding and C–R schemes, respectively. The effect of overhead is discussed in Section V, Section VI compares C–R with traditional reservation schemes, and Section VII offers concluding remarks.

II. Network Model and Preliminaries

A. Model and Definitions

The network comprises ground stations that transmit single-slot message fragments over randomly chosen channels. A hub monitors all channels and ACKs all successful receptions. The lack of an ACK when it is expected indicates

a collision. A station continues transmitting until success or expiration of the deadline.

The time from the beginning of a transmission (of one or more single-slot fragments) until the time by which an ACK for every transmitted fragment must be received (or else it is considered to have collided) is referred to as a *round.* Unlike slots, which must be synchronized among the stations, a round is "private" and requires no coordination. The typical duration of a round is up to several tens of slots.

A station transmits in rounds, waiting for the results of one round before continuing to the next, until the deadline; then, an as-yet unreceived message is declared lost. (We will consider very small permissible loss probabilities, so "lost" messages may be reissued with negligible effect on performance.)

Multiround Coding: A message is partitioned into K single-slot fragments, and a block erasure-correcting code is used to construct additional fragments from those, such that any K fragments suffice for correct decoding. A transmission scheme is mostly an algorithm for deciding how many fragments to transmit in each round as a function of the history of the message, the remaining time until the deadline and the permissible probability of missing the deadline.

User-Specified Constraints: A user-specified deadline is expressed in time units. For facility of exposition, we define this to be the time from the first transmission until the time of the latest transmission that would still arrive by the deadline. With fixed size slots, we use D_s to express the deadline in slots. For rounds of fixed duration, we use D_r to denote the maximum permissible number of rounds. P_e denotes the permissible probability of missing the deadline. The user-specified constraints are thus (P_e, D_s) or (P_e, D_r).

When round stretching [7] is used, let T_A denote the number of slots from single-slot transmission until ACK or until the next retransmission (round) may take place. Then

$$D_s = (D_r - 1)T_A + N_{\max} \tag{1}$$

where $N_{\max}$ is the maximum total number of transmitted fragments of any given message. When $T_A \gg 1$, D_r is not affected much by $N_{\max}$, and round stretching hardly changes performance. For small T_A the effect varies.

Channel Utilization: Because messages may be dropped, albeit with a low probability, a distinction was made in [7] between the generation rate of messages S_g and the throughput S. Specifically, $S = (1 - P_e)S_g$.

Remark: By a slight abuse of notation, we use P_e both as the failure-probability constraint and as the actual failure probability at any given working point. The intent should be obvious to the reader in each instance.

A successful K-slot message conveys K useful fragments. We define S, *channel utilization*, as the effective rate of successful fragments (per channel per time slot). For this purpose, only fragments of successful messages are counted, exactly K fragments per such message. The fragment generation rate S_g is K times the message generation rate (regardless of success). Consequently, we can again write $S = (1 - P_e)S_g$.

Stability: Multichannel ALOHA with message discarding upon deadline expiration can be bistable in certain load regions. (It is never unstable because of the limited message lifetime.) However, the hub can detect such situations and "push" the network into the "good" stable point, namely one in which increasing G increases S. The analysis in this paper applies to "good" stable operation. For additional details, see [13].

We assume an infinite number of stations and a large number of channels. The number of transmissions over any given contention channel in any given time slot is modeled as a Poisson random variable, independent from slot to slot and from channel to channel. With these assumptions, the probability of collision of a packet is only a function of the offered load on the channel over which it is transmitted. While this model is approximate, the approximation is close, normally within less than 10 percent of the true values. Moreover, because of the randomization in the choice of a channel, the quality of the approximation degrades gracefully when finite networks are considered. Finally, the independence assumption biases the performance of all schemes in the same direction, thereby reducing the inaccuracy of a comparison among them to a few percents at the most. For further discussion of the approximation and simulation results for finite networks, see [13].

B. Useful Relations

Let us briefly review some relations for pure single-working-point policies for single-slot messages [7]. Since $K = 1$, coding reduces to replication and we speak of *copies* of a message rather than fragments.

The offered load G is directly proportional to the generation rate of messages S_g and to the expected number of transmitted copies per message until success or deadline $E(N)$. Therefore

$$S_g = \frac{G}{E(N)}. \tag{2}$$

Channel utilization is thus

$$S = S_g(1 - P_e) = \frac{G(1 - P_e)}{E(N)}. \tag{3}$$

The total number of copies transmitted per message, N, is

$$N = \sum_i n_i \le \sum_{i=1}^{D_r} n_i = N_{\max} \tag{4}$$

where n_i denotes the number of copies transmitted in round i. The probability of collision is

$$P_c = 1 - e^{-G}. \tag{5}$$

Since $P(\text{reach round } i) = (P_c)^{\sum_{j=1}^{i-1} n_j}$, the expected total number of copies per message is

$$E(N) = n_1 + \sum_{i=2}^{D_r} n_i (P_c)^{\sum_{j=1}^{i-1} n_j}. \tag{6}$$

In Sections III and IV, we present and analyze multiround coding and C–R for multislot messages, respectively.

III. Multiround Coding

This scheme entails the partitioning of each message into K single-slot fragments; an erasure-correcting code is then used to derive additional fragments such that any K suffice for the reconstruction of the original message. As long as the deadline is not reached and fewer than K fragments have been received, a station may transmit one or more fragments per round. The challenge is to minimize the expected number of transmitted fragments per message while ensuring that, with probability $(1 - P_e)$, at least K fragments are received before the deadline.

Here, unlike the use of multicopy schemes for the individual message fragments, any redundant fragment can compensate for the loss of any message fragment, which is an advantage. Also, there is a useful notion of partial reception of a message.

A. Classes of Multiround Coding Schemes

The information pertaining to the progress of the transmission of the message comprises: the total number of fragments transmitted in previous rounds t; the number of fragments that must still succeed k; and the number of rounds remaining until the deadline d. We next introduce and analyze two classes of multiround coding schemes, which differ in the information used for deciding how many fragments to transmit in any given round.

The *fixed-$N_{\max}$* class bases its decisions on t, k, and d. If following the next-to-last round, successful message reception has not been achieved, the remaining budget of $N_{\max} - t$ fragments is transmitted in the final round. The main advantage of fixed-$N_{\max}$ policies is that the error probability P_e can be derived easily. A detailed study of this class appears in Section III-B.

The *budget-independent* class is motivated by the observation that, given k and d, the performance of the policy in the future is independent of t, the "budget" consumed in previous rounds, and simply ignores t. Since the optimization of this scheme is less constrained, it outperforms the previous one, but its optimization is much more computationally intensive. A detailed study of this class appears in Section III-C.

B. Fixed-$N_{\max}$ Class

1) Analysis: The probability that i of the n fragments transmitted in a given round succeed is

$$P(i \text{ good}|n) = \binom{n}{i} (1 - P_c)^i (P_c)^{n-i}. \tag{7}$$

A message is only abandoned after making the maximum effort, namely transmitting a total of $N_{\max} > K$ of its fragments. Therefore, and because the probability of collision of a fragment is the same in all rounds

$$P_e = \sum_{i=0}^{K-1} \binom{N_{\max}}{i} (1 - P_c)^i (P_c)^{N_{\max}-i}. \tag{8}$$

Given $N_{\max}$ and the required P_e, P_c and G can be calculated using (5) and (8). In order to maximize channel utilization, it is necessary to maximize S_g. Extending (2) to multislot messages

$$S_g = \frac{KG}{E(N)}. \tag{9}$$

The optimization goal is, thus, the minimization of $E(N)$, the expected number of fragments transmitted per message, for a given maximum number $N_{\max}$.

Remark: To understand (9), consider a C-channel network with K-slot messages generated at a rate of S_g/K messages per channel per slot, and a mean of $E(N)$ fragments transmitted per message until success or deadline. Then, $C \cdot G = C \cdot (S_g/K) \cdot E(N)$. This form of "normalized" (per channel per slot) expressions will be used extensively.

Optimization by Dynamic Programming: We must now determine $n(t, k, d)$, the number of fragments that should be transmitted in the current round. We begin by determining $n(t, k, 1)$, the number of fragments that should be transmitted in the last round, and continue by increasing d in each iteration and determining $n(t, k, d)$. If we are in the final round, $n(t, k, 1) = N_{\max} - t$. In earlier rounds, we choose $n(t, k, d)$ so as to minimize $f(t, k, d)$, the expected number of fragments that will be transmitted in the remaining rounds. When $d = 1$

$$f(t, k, 1) = n(t, k, 1) = N_{\max} - t. \tag{10}$$

When $d > 1$, suppose we transmit n fragments. If at least k of the fragments are successful, we have completed processing the entire message. Else, if $i < k$ of the fragments are successful, we need to transmit an expected $f(t+n, k-i, d-1)$ additional fragments in future rounds. Accordingly

$$f(t, k, d) = \min_{0 \le n \le N_{\max} - t} \left\{ n + \sum_{i=0}^{\min(k-1, n)} P(i \text{ good}|n) \cdot f(t+n, k-i, d-1) \right\} \tag{11}$$

where $1 < d \le D_r$, and $P(i \text{ good}|n)$ is obtained from (7). The fragment generation rate is

$$S_g = \frac{KG}{f(0, K, D_r)} \tag{12}$$

and the channel utilization is $S_g(1 - P_e)$.

The optimization is carried out using dynamic programming. This requires $N_{\max}$ as input, so the optimization iterates over $N_{\max}$, performing the dynamic programming in each iteration and picking the best result.

2) Results: When different values of K are used, one can interpret them either as reflecting different message sizes or as different degrees of fragmentation of fixed-size messages. In the latter case, the issue of overhead arises. The results presented here ignore the possible dependence of overhead on K. The effect of overhead, which applies equally to multiround coding and to the C–R scheme, will be discussed in Section V.

TABLE I
CHANNEL UTILIZATION OF FIXED-$N_{\max}$ MULTIROUND CODING

D_r	K	$P_e = 10^{-2}$			$P_e = 10^{-3}$			$P_e = 10^{-4}$		
		S	N_{max}	$\frac{N_{max}}{K}$	S	N_{max}	$\frac{N_{max}}{K}$	S	N_{max}	$\frac{N_{max}}{K}$
3	Classical	0.190	3	3	0.095	3	3	0.045	3	3
	1	0.279	7	7	0.247	12	12	0.233	15	15
	2	0.284	13	6.5	0.267	16	8	0.256	21	10.5
	3	0.294	17	5.67	0.279	21	7	0.270	27	9
	4	0.300	20	5	0.288	27	6.75	0.281	31	7.75
5	Classical	0.306	5	5	0.217	5	5	0.145	5	5
	1	0.340	8	8	0.321	10	10	0.312	15	15
	2	0.330	12	6	0.321	18	9	0.315	22	11
	3	0.333	17	5.67	0.327	21	7	0.322	26	8.67
	4	0.336	21	5.25	0.331	26	6.5	0.327	30	7.5

When $N_{\max}$ is small, it follows from (8) that a small P_c is required, in turn lowering G and the utilization (12). On the other hand, when $N_{\max}$ is too large, $E(N)$ is large, which also lowers the utilization (12). Thus, there is an intermediate value of $N_{\max}$ that is optimal. According to numerical results, the tradeoff between these two factors provides an optimal $N_{\max}$ using $P_c \approx 0.5$ across the range of P_e and D_r values that seem reasonable.

Table I presents results for networks with messages comprising up to four slots, several reasonable error probabilities, and deadlines that permit three or five rounds. Results for classical (slotted) ALOHA (but still the same performance measure) and single-slot replication [7] are shown for reference, both for single-slot messages. (This is an upper bound on the performance of single-slot schemes with multislot messages, because using them unaltered for a K-slot message would result in $P_e' = 1-(1-P_e)^K \approx K \cdot P_e$.) The table shows channel utilization and the $N_{\max}$ used by the optimal policy. An interesting measure is $\tilde{N}_{\max} = N_{\max}/K$, the maximum total number of transmitted fragments per message fragment.

The conclusions from Table I are as follows:

- Increasing K increases channel utilization. This happens because, according to (8) and the Chernoff bound, when $\lim_{K\to\infty} \tilde{N}_{\max}(K) > 1/(1-P_c)$, at least K fragments succeed with high probability, so $\lim_{K\to\infty} P_e(K) \to 0$. When $(D_r, P_e) = (5, 10^{-2})$ this is not the case, because the probabilities of reaching the last round are significant, and transmitting the remainder of the budget of $N_{\max}$ fragments is often wasteful, causing an unnecessary increase in $E(N)$ and reducing the channel utilization.
- Increasing K increases the optimal value of $N_{\max}$, because more fragments must be transmitted in order for more to succeed. On the other hand, $\tilde{N}_{\max}$ decreases.
- Decreasing P_e or D_r decreases channel utilization and requires larger $N_{\max}$ in order to satisfy the stricter constraints.

Fig. 3 depicts channel utilization with round stretching for the fixed-$N_{\max}$ class, using the (P_e, D_s) constraint. Results for classical ALOHA and single-slot replication [7] are shown for reference. Neglecting overhead, messages can be partitioned into K parts by dividing slot lengths by K. The length of a round is a physical parameter and is unchanged. Thus, T_A, the number of slots per round, is linear in K. The figure uses a normalized time scale. For $K = 4$, the figure also depicts multiround coding with an unlimited number of transmitters per station. The "bumps" in each curve represent the employment of an additional round. The conclusions are as follows:

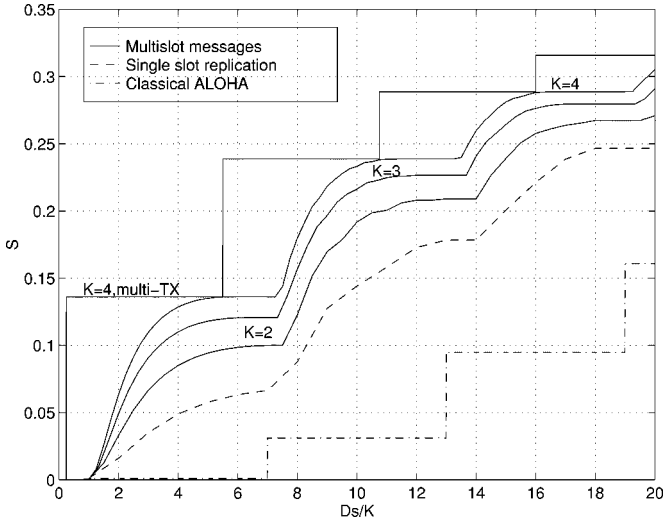

Fig. 3. Channel utilization with fixed-$N_{\max}$ multiround coding and round stretching. $P_e = 10^{-3}$; $T_A = 5 \cdot K$.

- For large D_s, channel utilization approaches $1/e$, the upper bound on utilization with slotted ALOHA in the absence of delay constraints.
- For any given scheme, utilization increases with an increase in D_r. With round stretching, however, especially for values of D_s that barely permit another round, one must decide whether to increase D_r at the cost of significantly reducing $N_{\max}$ or stay with one fewer round and slightly increase $N_{\max}$. The result of optimization is that, as D_s is increased and permits an additional round, the channel utilization with multiple transmitters rises immediately, whereas that with round stretching stays flat until such value of D_s for which the use of an additional round is warranted. Then, utilization rises sharply and eventually comes close to that with multiple transmitters per station. When assessing the performance penalty of round stretching in practical situations, it is useful to remember

that the addition of a few slots to the permissible delay is usually not critical, so the "critical" values of D_s at which the capacity difference is significant do not really exist, and the noticeable penalty of round stretching is thus small.

C. Budget-Independent Class

1) Analysis: Consider a situation in which k of a given message's fragments have yet to succeed, with d rounds remaining until the deadline. We denote the state of such a message by $Z_{(k,d)}$. A new message is in state $Z_{(K,D_r)}$, and its subsequent state trajectory is determined by the number of fragments that succeed in each round. If at least K fragments succeed before the deadline, or if the deadline is exceeded, the message enters some dummy state. Define $n(k, d)$ as the number of fragments transmitted for a message in $Z_{(k,d)}$, and $P(k, d)$ as the probability that a message goes through $Z_{(k,d)}$. Then

$$P(k, d) = \sum_{j=k}^{K} P(j, d+1)P(j-k \text{ good}|n(j, d+1)), \quad 1 \le d \le D_r - 1 \quad (13)$$

where $P(j-k \text{ good}|n(j, d+1))$ can be derived using (7).

Message failure occurs if, with a single round remaining, k fragments have yet to be received, and fewer than k succeed in the final round. Thus

$$P_e = \sum_{k=1}^{K} P(k, 1) \sum_{j=0}^{k-1} P(j \text{ good}|n(k, 1)) \quad (14)$$

where $P(j \text{ good}|n(k, 1))$ is calculated using (7).

According to (2), $S_g = KG/E(N)$. The mean number of fragments transmitted per message can be calculated given state probabilities and the number of fragments transmitted in each state.

$$E(N) = \sum_{k=1}^{K} \sum_{d=1}^{D_r} P(k, d)n(k, d). \quad (15)$$

Finally, the channel utilization is $S_g(1 - P_e)$.

2) Results: Given the constraint (P_e, D_r) and a transmission policy expressed as $n(k, d)$, we can find the P_c that fulfills the constraint. By iterating over $n(k, d)$, optimal budget-independent policies are found.

Table II presents channel utilization and $n(k, d)$ matrices for optimal budget-independent policies in the 3 round case. Increasing K improves the channel utilization, which is approximately 1% better than for fixed-$N_{\max}$ policies. (Results for $K = 1$ are not included because they are identical to those of [7].) Since the optimization requires an exhaustive search over $n(k, d)$ values, generating results for more rounds would be exceedingly time-consuming. However, the budget-independent class is better than the fixed-$N_{\max}$ class largely because the number of fragments transmitted in the last round is independent of the budget previously consumed. When more rounds are used, the probability of reaching the last round diminishes, so performance gains should become smaller.

TABLE II
CHANNEL UTILIZATION OF BUDGET-INDEPENDENT MULTIROUND CODING ($D_r = 3$)

P_e	$K=2$		$K=3$	
	S	$n(k, 4-d)$	S	$n(k, 4-d)$
10^{-2}	0.289	$\begin{pmatrix} - & 2 & 5 \\ 3 & 4 & 7 \end{pmatrix}$	0.300	$\begin{pmatrix} - & 3 & 5 \\ - & 5 & 9 \\ 6 & 7 & 12 \end{pmatrix}$
10^{-3}	0.271	$\begin{pmatrix} - & 3 & 8 \\ 4 & 6 & 12 \end{pmatrix}$	0.285	$\begin{pmatrix} - & 3 & 8 \\ - & 6 & 12 \\ 6 & 8 & 16 \end{pmatrix}$
10^{-4}	0.259	$\begin{pmatrix} - & 3 & 11 \\ 4 & 6 & 15 \end{pmatrix}$	0.275	$\begin{pmatrix} - & 4 & 11 \\ - & 9 & 16 \\ 6 & 9 & 20 \end{pmatrix}$

IV. C–R

This scheme begins with multiround coding (first phase). However, as soon as at least one fragment is received successfully prior to the last round, the hub immediately allocates channels for the contention-free transmission of the remaining k fragments within the remaining time (second phase). It is assumed that there are sufficient slots for allocation. C–R differs from traditional reservation schemes that use contention channels to make the reservation in several important ways: 1) the reservation-making packets of traditional schemes do not carry any payload and do not contribute to the throughput, 2) they must succeed within the first $D_r - 1$ rounds (whereas C–R can operate in contention mode in all D_r rounds), and 3) they may use shorter slots on special channels (in contention mode) for making reservations, thereby consuming less channel resources per transmission. (Note, however, that the actual duration of a round (and thus D_r) remains nearly unchanged because it is determined mostly by propagation delay and processing time.)

A reservation scheme can also carry the first message fragment along with the reservation request. By so doing, however, it cannot use shorter time slots (because of the payload), yet the reservation must succeed in $D_r - 1$ rounds. Such schemes can be viewed as a sub-optimal special case of C–R, and will not be discussed further.

We next analyze C–R, and derive a tight upper bound on channel utilization; the bound also offers some insight. Then, C–R is optimized using dynamic programming, and performance results are presented. A quantitative comparison between C–R and traditional reservation schemes cannot be divorced from the effect of header overhead, so we bring it in Section V.

A. Analysis

A C–R transmission policy enters the second phase as soon as a fragment is received. Therefore, and because all policies are deterministic, there is only a single possible path through the (t, k, d) trellis while it is in the first phase. Consequently, $n(t, k, d) = n(d)$.

We denote the number of fragments transmitted in round i while in the first phase by n_i. For a message to fail, all the

fragments transmitted in the first $D_r - 1$ rounds must fail, and at most $K - 1$ may succeed in the last round.

Let N_{first} denote the total number of fragments transmitted in the first $D_r - 1$ rounds, i.e., $N_{\text{first}} = \sum_{i=1}^{D_r-1} n_i$. Given that all fragments transmitted in those rounds failed, we must either transmit $n_{D_r} \geq K$ fragments in the last round or abandon the message. If $n_{D_r} \geq K$, we use (8) to arrive at

$$P_e = (P_c)^{N_{\text{first}}} \sum_{i=0}^{K-1} \binom{n_{D_r}}{i} (1-P_c)^i (P_c)^{n_{D_r}-i}. \quad (16)$$

Otherwise, $n_{D_r} = 0$ and

$$P_e = (P_c)^{N_{\text{first}}}. \quad (17)$$

The maximum total number of fragments is $N_{\max} = N_{\text{first}} + n_{D_r}$.

Derivation of channel utilization for C–R is complicated by the fact that two "types" of channels are used: contention channels with an offered load G in the first phase, and reserved contention-free channels in the second one. If the mean traffic on a set of contention channels with an offered load G is n fragments per slot, then the required number of channel slots is n/G. (Here, "channel slots" is a measure of channel resources, not delay.) We, therefore, derive the expected number of channel slots consumed by a message instead of the mean number of fragments transmitted.

Let us begin by deriving the fragment generation rate. Let $E(N_1)$ denote the mean number of fragments per message transmitted in the first phase, and $E(N_2)$—the expected number of fragments transmitted over contention-free channels. Then, the mean total number of channel slots required per message is $E(N_1)/G + E(N_2)$. Thus, the generation rate of fragments [counting K per message and using the same argument as in (9)] is

$$S_g = \frac{K}{\frac{E(N_1)}{G} + E(N_2)}. \quad (18)$$

Like (6), the expected number of fragments transmitted in the first phase is

$$E(N_1) = n_1 + \sum_{i=2}^{D_r} n_i (P_c)^{\sum_{j=1}^{i-1} n_j}. \quad (19)$$

For the second phase to take place in round i, no fragments may have been received prior to round $i-1$, and $1 \leq j \leq K-1$ must have been received in that round. Setting $n_k = 0$ for $k < 1$

$$E(N_2) = \sum_{i=2}^{D_r} \left((P_c)^{\sum_{k=\min(1,\,i-2)}^{i-2} n_k} \sum_{j=1}^{\min(n_{i-1},\,K-1)} (K-j) \cdot \binom{n_{i-1}}{j} (1-P_c)^j (P_c)^{n_{i-1}-j} \right). \quad (20)$$

An Upper Bound on Channel Utilization: Channel utilization is maximized by transmitting as many fragments as possible over reserved channels. However, in order to enter the second phase, at least one fragment must succeed over a contention channel. Therefore, and recalling that the unconstrained capacity of slotted ALOHA is $1/e$, the number of channel slots required for a K-slot message is at least $K - 1 + e$. In the best case, it follows from (18) that $S_g \leq K/(K-1+e)$. Since $S \leq S_g$, it follows that channel utilization with C–R is bounded from above by

$$S \leq \frac{K}{K-1+e}. \quad (21)$$

This bound is tight. With a long delay threshold and a large number of rounds, we can use a policy that transmits one fragment in every round and, once it succeeds, the remainder of the message is transmitted over reserved channels. The utilization in this case approaches the bound. Note that with $K = 1$, we never enter the second phase and the bound equals $1/e$, as expected; with $K \to \infty$, virtually all fragments are transmitted over reserved channels and the bound is indeed 1.0. An equivalent bound was derived in [5] for a traditional reservation scheme.

Optimization by Dynamic Programming: Given that N_{first} copies are transmitted during the first $D_r - 1$ rounds, and n_{D_r} during the last round, we want to find the sequence (n_i) that minimizes the expected number of required channel slots (taking into account the offered load, as was done in the analysis). Let $f(t, d)$ denote the expected remaining number of channel slots needed by C–R, given that t fragments were transmitted in previous rounds and d rounds remain until the deadline. Although t is necessary for the dynamic programming, it will not be used by the policy itself. A policy that reaches the last round while in the first phase must have transmitted N_{first} fragments in the past, and must transmit n_{D_r} in the last round. Therefore

$$f(t, 1) = \begin{cases} \dfrac{n_{D_r}}{G}, & t = N_{\text{first}} \\ \infty, & \text{otherwise.} \end{cases} \quad (22)$$

(The second term is merely an artifact of the method.)

When $d > 1$ and the policy is still in the first phase, the expected number of channel slots needed by the message is made up of contention slots used in the current round as well as either reserved slots used in the next round or $f(t + n_{D_r-d+1},\, d-1)$. The former are required when some fragment(s) succeed in the current round, causing the policy to move to the second phase and transmit any remaining required fragments in the next round. The latter are required if all fragments transmitted in the current round collide and the policy remains in the first phase. Accordingly

$$f(t, d) = \min_{0 \leq n \leq N_{\text{first}} - t} \left\{ \frac{n}{G} + (P_c)^n f(t+n, d-1) + \sum_{i=1}^{\min(n,\,K-1)} (K-i) \binom{n}{i} (1-P_c)^i (P_c)^{n-i} \right\}, \quad 1 < d \leq D_r. \quad (23)$$

Once the dynamic programming is performed up to $d = D_r$

$$S_g = \frac{K}{f(0, D_r)} \quad (24)$$

and the channel utilization is $S = S_g(1 - P_e)$.

The dynamic programming requires N_{first} and n_{D_r} as input. From those, P_c and G are calculated. Then, $f(t, d)$ and (n_i)

TABLE III
CHANNEL UTILIZATION WITH CODING–RESERVATION (C–R)

D_r	K	$P_e = 10^{-2}$		$P_e = 10^{-3}$		$P_e = 10^{-4}$	
		S	(n_i)	S	(n_i)	S	(n_i)
3	Classical	0.190	(1,1,1)	0.095	(1,1,1)	0.045	(1,1,1)
	1	0.279	(1,2,4)	0.247	(2,3,7)	0.233	(2,3,10)
	2	0.430	(2,3,6)	0.408	(2,3,9)	0.394	(2,4,13)
	3	0.521	(2,3,8)	0.505	(2,4,12)	0.493	(2,4,16)
	4	0.585	(2,4,10)	0.571	(2,4,15)	0.561	(2,5,19)
5	Classical	0.306	(1,1,1,1,1)	0.217	(1,1,1,1,1)	0.145	(1,1,1,1,1)
	1	0.340	(1,1,1,2,3)	0.321	(1,1,1,2,5)	0.313	(1,1,2,3,8)
	2	0.500	(1,1,2,3,4)	0.490	(1,1,2,3,8)	0.482	(1,1,2,3,11)
	3	0.597	(1,1,2,4,0)	0.587	(1,1,2,4,10)	0.582	(1,1,2,4,14)
	4	0.664	(1,1,2,4,0)	0.653	(1,1,2,4,12)	0.648	(1,1,2,4,16)

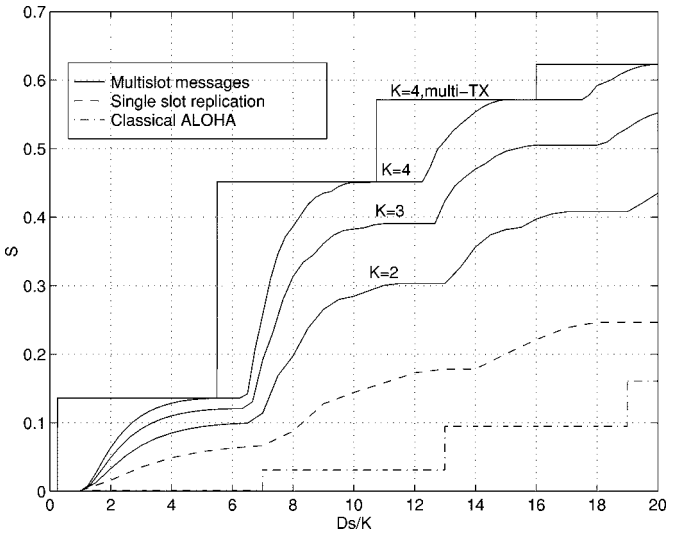

Fig. 4. Channel utilization of C–R with round stretching. $P_e = 10^{-3}$ and $T_A = 5 \cdot K$.

are found with dynamic programming. Lastly, the channel utilization is calculated. In order to find optimal C–R policies, a software package performing all of these was written. External nested loops iterate over N_{first} and n_{D_r}; the dynamic programming is carried out in each iteration, and the best solution among all iterations is the optimum.

B. Results

Table III shows results for networks with messages comprising up to four slots, several reasonable error probabilities, and delay thresholds of three and five rounds. Results for classical (slotted) ALOHA and for optimized single-slot replication [7] are shown for reference. The table shows channel utilization and (n_i).

The conclusions from Table III are similar to those from Table I. Points of interest are as follows.

- When K and D_r are increased, the channel utilization approaches the bound (21) instead of $1/e$.
- The optimal value of n_{D_r} is indeed sometimes zero.

Fig. 4 depicts channel utilization subject to the (P_e, D_s) constraint for C–R with round stretching. Classical ALOHA and single-slot replication [7] are shown for reference. The graph uses a normalized time scale, like Fig. 3. A curve for C–R with $K = 4$ and an unlimited number of transmitters per station is included in order to illustrate the effect of round stretching on performance. As before, the increase in overhead due to the partitioning of a message into K fragments is neglected. The conclusions are also similar, except that the channel utilization approaches the bound of (21) rather than $1/e$ when D_s is sufficiently large.

V. THE EFFECT OF OVERHEAD

Multislot messages can be viewed as resulting from a chosen slot size and independently chosen (possibly by an application) message sizes. In this case, the value of K is given and the analysis presented can be taken at face value. In other cases, however, message sizes are given, and selection of slot size is a design parameter, with a trade-off between internal fragmentation and header overhead.

Partitioning a message into K single-slot fragments would reduce utilization by a factor $(1+\Delta)/(1+K\Delta)$, where Δ is the ratio of header (overhead) to payload in single-slot messages. With our schemes, however, this is offset (at least in part) by the increase in channel utilization that is brought about by larger K. While the optimal choice can only be determined by examining the actual numbers, we can conclude that the use of the new schemes would lead to smaller optimal slot sizes.

In order to gain some quantitative insight, let us consider a "typical" internet message payload length distribution [14]

$$p = \begin{cases} 0.5 & 50 \text{ Bytes} \\ 0.3 & 500 \text{ Bytes.} \\ 0.2 & 1500 \text{ Bytes} \end{cases}$$

Additionally, we assume that a 50-byte header must be included in each slot, and a delay constraint of $(P_e, D_r) = (10^{-3}, 3)$. Next, consider two schemes: the optimal multicopy scheme of [7], applied individually to the single-slot fragments comprising a message, and C–R. In [13], each scheme was optimized (including slot length used by each scheme), and the performance was compared. The results, which are biased in favor of the multicopy scheme due to the use of bounds, were expressed in terms of the mean number of channel bytes consumed per message. This measure is used instead of channel slots because the optimal slot lengths are different for the two schemes. It was found that C–R requires approximately half as many channel bytes per

message than does the optimized multicopy scheme. The optimal slot lengths were 150 and 300 bytes, respectively. One could further optimize C–R by jointly optimizing slot length and coding for different packet lengths while adhering to the (P_e, D_r) constraint.

VI. C–R Versus Traditional Reservation Schemes

In traditional reservation schemes, the policy can be divided into two phases. The first phase uses contention channels (only) to request contention-free slots from the hub. If it succeeds prior to the deadline, the hub allocates slots that are used without contention in the following round. For the first phase, the basic traditional reservation (BTR) scheme transmits one copy per round to make the reservation; the optimized traditional reservation (OTR) scheme employs the single-slot scheme of [7] in making the reservations. Both must succeed in reserving slots within $D_r - 1$ rounds with probability $(1 - P_e)$. The second phase entails the transmission of the entire message without contention. Given the message size distribution and header size, optimal per-slot payload size can be derived through the trade-off between header overhead and internal fragmentation. (Payload size is unaffected by the constraints because no payload is transmitted in the reservation-making phase.) Throughout the comparison, our schemes will use the same working point for all channels and we will only consider pure policies.

Let us define the slot length L, the length of the header H, and the length of the payload P, then $L = H + P$. For convenience, we let the size of the slot used in reservation-making channels equal the message header H. (We note in passing that both C–R and the traditional reservation schemes can be made more efficient by allocating contiguous slots on the same channel for the transmission of all required fragments in the second phase, thereby requiring only a single header for all of them. Analysis of this optimization is left for future research.)

Both the basic and optimized traditional reservation schemes (BTR and OTR) use a mean of $(E(N)/G) \cdot H + K(H + P)$ channel bytes per message, where $E(N)$ is the mean number of copies transmitted in the first phase and depends on the scheme as well as on the constraints.

For BTR, $E(N)$ is the mean number of transmission attempts until success or deadline for the given (P_e, D_r). In this case, $P_e = P_c^{D_r - 1}$, so [7]

$$E(N) = \sum_{i=1}^{D_r - 1} P_e^{(i-1)/(D_r - 1)} = \frac{1 - P_e}{1 - P_e^{1/(D_r - 1)}}. \qquad (25)$$

For OTR, $E(N)$ is the minimum mean total number of copies per message transmitted by the optimized single-slot scheme of [7] for $(P_e, D_r - 1)$. H is assumed given, as is the message-size distribution; P, G and $E(N)$ are assumed to have been jointly optimized for each scheme based on the inputs.

Based on (18), C–R uses a mean of $((E(N_1)/G) + E(N_2))(H + P)$ channel bytes per message. Again, parameter values are optimized for each scheme based on the inputs.

Despite the difference in the optimal choice of parameters for the different schemes for any given situation, the foregoing expressions suggest the existence of a crossover point: given (P_e, D_r) and the packet-length distribution, there are header sizes above which C–R outperforms BTR and possibly OTR.

TABLE IV
C–R Versus OTR (Numerical Example)

D_r	P_e	$H_{crossover}$	L_{OTR}^{opt}	L_{C-R}^{opt}
2	10^{-2}	204	704	341
	10^{-3}	71	321	155
	10^{-4}	50	300	96
	10^{-5}	38	205	80
3	10^{-2}	694	1194	944
	10^{-3}	894	1394	1394
	10^{-4}	351	851	601
	10^{-5}	329	829	579

TABLE V
C–R Versus BTR (Numerical Example)

D_r	P_e	$H_{crossover}$	L_{BTR}^{opt}	L_{C-R}^{opt}
3	10^{-2}	38	205	110
	10^{-3}	8	108	32
	10^{-4}	2	52	17
	10^{-5}	1	51	16
4	10^{-2}	228	728	395
	10^{-3}	98	348	348
	10^{-4}	17	142	67
	10^{-5}	4	54	29

The delay constraint also affects the comparison, because C–R can always use all D_r rounds, whereas the traditional schemes must succeed in making a reservation within $D_r - 1$ rounds. Consequently, we expect the the header-size crossover point to become smaller when D_r is smaller. We next present numerical results for the message length distributions of the numerical example in Section V. The parameter optimizations for the different schemes are omitted for brevity.

Table IV presents the header-size crossover point between optimality of OTR and that of C–R, as well as the optimal slot lengths for the two schemes for various (P_e, D_r) values. The conclusions are as follows.

- In general, when harsher constraints are imposed, C–R is superior across a broader range of packet lengths and header sizes.
- For the message length distributions of the numerical example, OTR outperforms C–R for reasonable levels of overhead when three or more rounds are permitted.
- C–R uses smaller optimal slot lengths than do the traditional schemes.

Table V presents the header-size crossover point between optimality of BTR and that of C–R, as well as the optimal slot lengths for the two schemes with various (P_e, D_r) values. Message length distributions were taken from the numerical example in Section V. The conclusions are similar to those from the previous table, except that for the packet length distributions of the numerical example, C–R outperforms BTR across a broader range of parameters, especially for small P_e.

Remark: In both comparisons, we used the lower bound on channel utilization of C–R. The situation is thus actually more favorable to C–R.

VII. Conclusion

This paper addressed multislot messages in multichannel ALOHA networks, focusing on capacity maximization subject to a user-specified deadline and a permissible probability of failing to meet it.

The multislot approaches introduced here provide substantial performance improvements relative to even the best single-slot approaches [7], even if reservation is not allowed. The channel utilization of multiround coding approaches $1/e$ when K or D_r are increased. C–R is even better, providing utilization well in excess of $1/e$ even when $K = 2$, because it uses contention-free channels for part of the fragments. Multiround methods are practical because they work well when harsh constraints are imposed, even in systems with a single transmitter per station.

With extremely harsh delay constraints, C–R outperforms optimized traditional reservation schemes. This is due in part to the fact that C–R can use all D_r rounds, whereas BTR and OTR must succeed in making a reservation within $D_r - 1$ rounds. The optimization of slot lengths along with coding for different message lengths, while maintaining some delay constraint, is beyond the scope of this paper.

Capacity was maximized in this paper by minimizing the mean amount of transmission resources per message. As a result, the proposed schemes are energy efficient, a very important feature for battery operated devices. An interesting related problem is the minimization of mean per-message transmission energy given P_e, D_r, and S, where S is below capacity.

The discussion in this paper was limited to time-slotted multichannel systems. Nonetheless, at a cost of a reduction in capacity, the schemes are also applicable to unslotted systems. Also, multiple channels can be emulated by a single, high-speed channel, but this would require higher transmission power because of the shorter time per bit.

Finally, we note that the results of this paper serve as yet another example of the benefits gained from the judicious use of redundancy for performance enhancement. By deferring the expenditure of redundancy to the late rounds, we were able to attain a low probability of missing the deadline with very little "pollution." This enabled the attainment of maximum throughput that is not much smaller than the throughput attainable in the absence of a delay constraint.

Acknowledgment

The authors wish to thank R. Rom, T. Kol, and the two anonymous reviewers for insightful comments.

References

[1] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Trans. Commun.*, vol. COM-25, pp. 117–128, Jan. 1977.

[2] R. Rom and M. Sidi, *Multiple Access Protocols*. New York: Springer-Verlag, 1990.

[3] L. Kleinrock and S. S. Lam, "Packet switching in a multiaccess broadcast channel: Performance evaluation," *IEEE Trans. Commun.*, vol. COM-23, pp. 410–423, Apr. 1975.

[4] W. Yung, "Analysis of Multichannel ALOHA Systems," Ph.D. dissertation, Univ. California, Berkeley, Nov. 1978.

[5] S. S. Lam, "Packet broadcast networks—A performance analysis of the R-ALOHA protocol," *IEEE Trans. Comput.*, vol. C-29, pp. 596–603, July 1980.

[6] E. W. M. Wong and T. S. P. Yum, "The optimal multicopy Aloha," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1233–1236, June 1994.

[7] Y. Birk and Y. Keren, "Judicious use of redundant transmissions in multichannel ALOHA networks with deadlines," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 257–269, Feb. 1999.

[8] L. Cooper and M. W. Cooper, *Introduction to Dynamic Programming*. New York: Pergamon, pp. 31–44.

[9] D. Baron and Y. Birk, "Multiple working points in multichannel ALOHA with deadlines," *Wireless Networks*, vol. 8, pp. 5–11, Jan. 2002.

[10] Y. W. Leung, "Generalized multicopy ALOHA," *Electron Lett.*, vol. 31, pp. 82–83, Jan. 1995.

[11] D. Baron and Y. Birk, "On the merits of impure multicopy schemes for multichannel slotted ALOHA with deadlines," Technion, Haifa, Israel, EE Tech. Rep. no. 1249 (also CC-pub 315), June 2000.

[12] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1978.

[13] D. Baron and Y. Birk, "Multiround coding and Coding–Reservation for multislot messages in multichannel ALOHA with deadlines," Technion, Haifa, Israel, EE Tech. Rep. no. 1293 (also CC-pub 359, Oct. 2001.

[14] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, Nov./Dec. 1997.

Dror Baron (S'97) was born in Haifa, Israel, in 1973. He received the B.Sc. (*summa cum laude*) and M.Sc. degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1997 and 1999, both in electrical engineering.

From 1992 to 1994, he developed software at the Israel Defense Forces. In 1997, he was a Teaching Assistant in the Electrical Engineering Department at the Technion. From 1997 to 1999, he worked at Witcom Ltd., Yokneam, Israel, in modem design. Since 1999, he has been a Research Assistant at the University of Illinois at Urbana-Champaign, Urbana, IL, where he is pursuing the Ph.D. degree. His research interests include information theory, data compression, communications systems, signal processing, and hardware design.

Yitzhak Birk (SM'00) received the B.Sc. (*cum laude*) and M.Sc. degrees from the Technion, Haifa, Israel, in 1975 and 1982, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, in 1987, all in electrical engineering.

From 1976 to 1981, he was project engineer in the Israel Defense Forces. From 1986 to 1991, he was a Research Staff Member at the IBM Almaden Research Center, San Jose, CA, where he worked on parallel architectures, computer subsystems, and passive fiber-optic interconnection networks. From 1993 to 1997, he also served as a consultant to Hewlett Packard Labs, Palo Alto, CA, in the areas of storage systems and video servers. He has been on the faculty of the Electrical Engineering Department at the Technion, Haifa, Israel, since October 1991, and heads its Parallel Systems Laboratory. His research interests include computer systems and subsystems, as well as communication networks. He is particularly interested in architectures for information systems, including communication-intensive storage systems (e.g., multimedia servers), and satellite-based systems with special attention to the true application requirements in each case. The judicious exploitation of redundancy for performance enhancement in these contexts has been the subject of much of his recent work.