Judicious Use of Redundant Transmissions in Multi-Channel ALOHA Networks with Deadlines

Yitzhak Birk and Yaron Keren
Electrical Engr. Dept, Technion
Haifa 32000, ISRAEL
birk@ee, yak@tx .technion.ac.il

Abstract—**This paper shows how to improve the classic multi-channel slotted ALOHA protocols by judiciously using redundant transmissions. The focus is on user-oriented requirements: deadlines and a permissible probability of failing to meet them. Subject to those, maximization of throughput is the optimization goal. When there is with no success/failure feedback prior to the deadline, the use of information dispersal with some redundancy provided by error-correcting codes for the data in conjunction with a replicated, separately-transmitted synchronization preamble sharply reduces the overhead resulting from the use of shorter packets and significantly increases capacity. When the permissible delay is several times greater than the round-trip propagation delay, we propose a novel retransmission policy: all attempts except the final one entail the transmission of a single or very few copies, and the remaining copies are transmitted in the final attempt. This sharply increases channel capacity.**

Key words and phrases: *ALOHA, multi-channel, redundancy, VSAT, deadline scheduling, information dispersity, dispersity routing.*

I. INTRODUCTION

The slotted ALOHA access scheme, first suggested in [1], gained popularity due to the simple implementation and random-access nature. Essentially, each station transmits a packet of data when it becomes available (aligned to a common time clock) on a multiple-access channel. Should another station transmit in the same time slot, both transmissions would not be received correctly. After such failure, some randomization takes place according to a retransmission policy [2] in order to avoid a definite repeated collision, and another transmission is attempted. Attempts repeat until the transmission is received successfully at its destination. With a single channel, temporal randomization is the only choice. With a multi-channel, in contrast, the choice of channel is the primary avenue for randomization, as this permits immediate retransmission following a collision.

Most of the work on ALOHA has been carried out for single-channel systems, with a focus on capacity and sometimes on the interplay between throughput and mean delay. In practice, however, a communication system is often viewed as the provider of a service, whose quality is specified by the users. This may, for example, include a maximum permissible delay (deadline) along with a maximum permissible probability of exceeding it. Subject to meeting these requirements, a secondary goal may be the minimization of the mean delay. The main optimization goal of a system designer may be to maximize communication capacity while meeting the quality-of-service requirements. Our focus is on this situation in multi-channel systems.

The main contribution of this paper is new and optimized schemes for judicious use of redundancy in order to improve performance as just defined. This entails the choice of the proper degree of redundancy, the timing of redundant transmissions and, optionally, the use of several power levels as a priority mechanism. (The latter is not considered in this paper.) One form of redundancy entails dividing a packet into several sub-packets, computing from those a larger set of sub-packets, and transmitting them; any subset of sufficiently large cardinality, typically equal to the original packet, suffices for reconstruction of the original packet. In parts of the paper, we will use a simpler form, namely packet replication, whereby several copies of the packet are transmitted. (Replication does not permit fine control over the degree of redundancy.) This form of redundancy has been analyzed in [3] and [4] with the primary objective of maximizing throughput without deadlines.

In using redundancy, two cases must be considered: 1) all decisions must be made up front, without waiting for success/failure notification; 2) there are several transmission–feedback rounds prior to the deadline. We refer to those as *single-round* and *multi-round*, respectively. (Without redundancy, retransmissions take place only post-feedback, and are not redundant since they are known to be necessary.) A "round" is composed of a transmission attempt and the delay until feedback arrives. The terms "attempt" and "round" will often be used interchangeably in this work. The single-round case is of primary interest for systems that do not provide feedback, as well as for cases wherein waiting for feedback would cause the deadline to be missed. The multi-round case is of primary interest when the permissible delay is several fold larger than the time until feedback is received.

The judicious use of redundancy in slotted ALOHA networks with various goals has recently been addressed in [5] [6]. In [5], a single two-copy transmission attempt is considered on a single channel; given a deadline and the permissible probability of missing it, the goal is to minimize the expected delay of successful packets. The optimal probability function of the inter-copy delay is derived, and is shown to be a linear monotonically decreasing function. In [6], multi-copy transmissions are considered on a multi-channel with no deadlines. The throughput–mean-delay characteristics are improved substantially by a proper choice of the (fixed) number of copies transmitted in each attempt.

The remainder of this paper is organized as follows. Section II describes the channel models and performance measures. Section III presents and analyzes the use of error-correcting codes for the single-round case on a multi-channel, exploiting unique characteristics of geostationary satellite systems and their ground stations. In section IV, we consider the multi-round case for a multi-channel system with a deadline, and present retransmission policies that dramatically increase capacity for any given prob-

This research was supported in part by the Consortium for Satellite Groundstations.

ability of meeting the deadline. In section V we scrutinize our assumptions, and section VI offers concluding remarks.

II. Traffic model and performance measures

A. Traffic model

Our model is similar to [7], as follows. There are M multiple-access channels, over which an infinite number of user stations transmit data packets at slotted starting times. The transmission of a data packet takes a single time slot, unless a packet is partitioned into sub-packets, in which case the transmission of a sub-packet takes a single time slot. We assume an erasure channel and that collisions are the sole source of erasure. Feedback, if available, arrives after transmission and serves as a collision-notification mechanism. Whenever there is a deadline, which arrives after D_s time slots (or D_r rounds), a station ceases to retransmit a packet if it will not meet the deadline. Such a packet is considered lost. Accordingly, we distinguish between the *generated throughput* S_g and the actual *throughput* S, though the difference between them is very small in most practical situations. The number of new data packets per channel in each slot is generated according to a Poisson distribution with mean S_g. This, together with the retransmitted data, is a random variable distributed according to a Poisson distribution with mean G; the Poisson assumption for the offered load G is justified by randomizing retransmissions [2]. We emphasize that that the packet-generation *process* is not assumed to be a Poisson process. A station is assumed to know the current value of G, possibly based on hub estimation. Note that S_g itself is not a random variable; rather, it is a given (or maximized) number. In the case of multi-channel systems, even when multi-copy transmissions occur, we will continue to use the per-channel measures. This is correct since the transmission of multiple copies takes place on randomly selected channels. Stations are assumed to continue generating new data even while attempting to retransmit a previous packet, in contrast with the commonly used station state model (idle/backlogged). This reflects a situation whereby the generation of messages by applications is unaffected by the details of the state in lower-level protocols.

With temporal randomization on a single channel, the limited permissible delay often severely restricts the number of time slots from which the retransmission time can be chosen, and this brings about a dependence among the fates of different copies of a packet, which reduces performance. In contrast, with multichannel systems featuring more than 100 channels and a restricted number of copies transmitted in each attempt, there is effectively no such dependence among the fates of transmissions in different time slots and among those of multiple copies transmitted in the same time slot. The analysis in this paper is carried out under such an independence assumption, which is confirmed by simulations.

A station is likely to have a limited number of transmitters, which limits the number of concurrent transmissions by a given station. In parts of the paper, this constraint will initially be ignored but will subsequently be addressed.

The techniques presented in this paper may decrease the network stability. We will assume that a higher level protocol level will stabilize the network. The analysis in this work applies only to the stable periods.

B. Performance Measures

The *success probability*, P_s, is the probability of decoding a data packet correctly prior to the specified deadline. In the case of a packet that is broken into sub-packets, this refers to the decoding of the entire packet from received sub-packets. The *error probability* P_e is $1-P_s$. *Throughput* (S) is the mean number of data packets per channel that are decoded correctly in each time slot. It is related to the generation rate S_g through $S = S_g \cdot P_s$. *Mean delay* ($\overline{d}$) is the expected time from the first transmission of a (new) data packet until the transmission of its first copy that is received correctly. (Whenever retransmission ceases upon expiration of a deadline, mean delay applies only to successfully-received data packets; the remaining ones are accounted for by the probability of failure.) $\overline{d_s}$ and $\overline{d_r}$ will be used to denote expected delay in slots and in rounds, respectively. (In classic ALOHA, $\overline{d_r} = e^G$.) Note that the performance measures relate to entire data packets, as opposed to sub-packets.

III. A single-round transmission technique

In this section, we explore the case of a multi-channel system, in which the deadline is such that all transmissions related to a given packet must take place prior to the receipt of any success/failure feedback. Given the permissible number of time slots (limited by the deadline) and a large number of channels, we may spread our transmissions in time and/or frequency. Our goal in this case is to maximize the attainable throughput subject to a permissible probability of failure in the first and only transmission round.

One proposal for using redundancy in order to enhance performance in similar situations is *redundant dispersity routing* [8]. This entails breaking a packet down into several sub-packets, constructing several redundant sub-packets and transmitting all sub-packets. If a number of sub-packets which equals or exceeds the number of non-redundant sub-packets is received prior to the deadline, this is considered a success. A similar idea was presented in [9]. In [10], the idea was extended to prioritized dispersal, whereby the redundant sub-packets receive a lower priority than the "original" ones.

The discussion here will be carried out in the context of very small aperture terminals (VSATs) and geostationary satellites. Such systems are characterized by a very small variability in propagation delay and a single point of synchronization (satellite or hub). Accordingly, the temporal guard bands required between time slots may be very small, thereby making it practical to break a packet down into many sub-packets without incurring a large overhead due to guard bands. In this case, two important issues must be addressed:

- Helping the transmitter and receiver coordinate a "code", i.e., a sequence of (time, frequency) slots in which the sub-packets of a given packet are transmitted. Previously-analyzed options for code selection are transmitter-based codes, receiver-based codes and hybrid methods [11], [12]. With finite networks, one can indeed assume that a specific code and hopping pattern is assigned to each transmitter [13]. With a very large population, however, as in our case, this is problematic. Our solution is to include the code in the header of a message.

- Overcoming the large header-overhead that results from partitioning a data packet into many small sub-packets. Despite the small guard bands, overhead increases as one reduces packet size due to the required header. (This header contains the usual source and destination information; in our case, it must also contain code-related information.)

Our solution to both problems is as follows. To make the initial connection, a transmitter selects a seed for a previously agreed upon random number generator. The seed, together with synchronization, code and other overhead information is transmitted as a sub-packet several times, so the probability of not receiving any copies of this initial sub-packet is negligible. After this phase, the transmitter proceeds to transmit the data sub-packets on channels selected according to the random number generator. Since the receiver knows exactly when and on which channel the next transmission will occur, no overhead is needed in each sub-packet except for the small guard bands. Before transmission, the sub-packets are coded redundantly so as to ensure correct decoding for the expected number of collisions. We note that sub-packet collisions are independent since channels are selected at random. With this technique, throughput is maximized for a given probability of success. This method can be considered as a hybrid technique combining ALOHA and frequency-hopping spread-spectrum.

It should be noted that the permissible delay and the number of channels define a boundary within which the copies may be placed. The only constraint is that the synchronization sub-packets must be transmitted prior to the data sub-packets. To minimize delay, one would transmit the synchronization sub-packets in one slot, followed immediately by all copies of the data sub-packets. However, hardware constraints such as a limited number of transmitters may mandate the spreading of transmissions over several time slots.

Analysis

Each original packet comprises d bits of data and an h-bit header. Each data sub-packet comprises $\frac{d}{k}$ bits of data and no header. The "synchronization" sub-packets contain only an h-bit header. For facility of analysis, we set k such that $\frac{d}{k} = h$. Also, we ignore the fact that the preamble packet must be longer than the header of the original packet due, for example, to the need to include the seed for the code generator. This effect is secondary, and in any case our results in this section should be taken as an indication rather than precise numbers.

Subpacket size will be derived from the number of overhead bits h (including the seed), and the d bits of data will be split among k subpackets, each consisting of h bits. The k data subpackets will be redundantly coded to $n > k$ subpackets and transmitted on "randomly-selected" channels. Transmission of the preamble subpacket will be repeated R times.

The receiver can decode the original packet from any k-subset of the n transmitted subpackets (erasure channel), provided that it has successfully received at least one copy of the preamble subpacket. The probability of receiving at least one of the copies of the preamble is

$$1-(1-e^{-G})^R. \qquad (1)$$

The probability of losing fewer than $n-k$ data sub-packets is

$$\sum_{i=0}^{n-k}\binom{n}{i}\cdot(1-e^{-G})^i\cdot e^{-G(n-i)}. \qquad (2)$$

Since those are independent events, the probability of success P_s is

$$P_s = (1-(1-e^{-G})^R)\cdot\sum_{i=0}^{n-k}\binom{n}{i}\cdot(1-e^{-G})^i\cdot e^{-G(n-i)}. \qquad (3)$$

The overhead for R sub-packets is $R\cdot h$ bits, compared with h bits of ordinary ALOHA. The useful data enclosed is $d = k\cdot h$ bits. The throughput is therefore

$$S = G\cdot P_s\cdot\frac{h+k\cdot h}{R\cdot h+n\cdot h} = G\cdot P_s\cdot\frac{k+1}{n+R}. \qquad (4)$$

(If the difference between the size of the preamble packet and the size of the original packet header were taken into account, "$k+1$" in the above equation would be replaced with "$k+h'/h$", where h' is the original header size and h is the size of the preamble sub-packet.)

Numerical results

In obtaining the results, we used $d = 1000, h = 100$, $k = 1000/100 = 10$, and a $(32,10)$ error correcting code ($n = 32$). Several values of P_s were used, and R was selected to maximize throughput in each case. (Note that k,n,G,R and P_s are related through (3) so G is determined once the others are assigned values.)

Table 1 presents comparative results for our scheme and single-channel ALOHA (our throughput is per-channel). $S(ALOHA)$ is the throughput achievable by classic ALOHA for a probability of success P_s, namely

$$S = G\cdot e^{-G} = (-ln(P_s))\cdot P_s. \qquad (5)$$

P_s	R	S	$S(ALOHA)$
0.9	6	0.2219	0.0948
0.99	8	0.1820	0.0099
0.999	5	0.1458	0.0001

Table 1
Attainable throughput S for $P_s = 0.9, 0.99, 0.999$.

The results show that our method provides a dramatic increase in the throughput that can be attained while still providing a very high probability of success in the first attempt. Moreover, it permits the use of simple narrow-bandwidth transmitters together with cheap processing power. [12] and [14] provide further analysis of ECC usage.

The non-monotonic behavior of the optimal value of R can be explained by the trade-off between the negative effects of increasing it on the probability of success of data sub-packets through increasing the load on one hand, and increasing the probability of successful synchronization on the other hand. Finally, we note that even better results can be obtained by jointly optimizing n and R, so the above serves only as an example of achievable improvement.

IV. Multi-round retransmission policies

In this section, we consider the situation wherein the deadline permits up to D_r transmission attempts (rounds), with a new attempt being made only after success/failure feedback has been received for the previous one. Immediately following the receipt of feedback which indicates failure of a transmission attempt, a station transmits one or more of copies of the lost packet over randomly-chosen channels. Since there is no benefit from delays in a multi-channel system, where collisions are independent of each other, at least one packet will be transmitted in each attempt. Following success, retransmission ceases; after D_r attempts, a packet is declared lost and discarded. (In practice, a higher-level protocol may resubmit the packet, but it would be considered a new one. Moreover, assuming a low permissible probability of failure, the high-level retransmission traffic is negligible.)

Given the D_r and the permissible probability of failure, our goal is again to maximize the attainable throughput. This is done through a judicious choice of the number of copies that should be transmitted in each attempt (not necessarily the same number in all attempts).

Conventional back-off policies aimed at preventing instability, when applied to a multi-channel, would call for a monotonically non-increasing number of copies in successive retransmission attempts. However, while the stability argument behind this approach is valid asymptotically, we claim that this monotonicity may be violated for any bounded number of retransmissions without hurting stability, and will in fact show that so doing can dramatically increase performance.

Our approach typically entails the transmission of a single or very few copies in all but the final attempt, in which the remaining copies are transmitted if necessary. This implies that, given the maximum number of copies that may be transmitted (jointly in all attempts), our goal is to minimize the expected aggregate number of copies transmitted within the D_r permissible attempts for any achievable probability of success P_s. This, in turn, minimizes the amount of load generated per successful message, thereby maximizing the system's communication capacity.

There is a tradeoff in selecting how to use an allotted "budget" of copies. On one hand, we would like to postpone the transmission of all but one copy per attempt in the hope that an early attempt will be successful and later ones thus avoided. On the other hand, it might be beneficial to transmit more than one copy per attempt prior to the last attempt, since this increases the probability of avoiding the last, "costly" attempt.

Analysis

We begin by presenting the relations between the various variables. Next, we analyze two schemes: 1) multi-copy ALOHA [4] (a constant number of copies in each attempt), and 2) a new scheme, whereby a single copy per attempt transmitted in all but the last attempt. Finally, we employ dynamic programming [15] to derive the optimal retransmission strategy.

The number of copies transmitted by a single station, even in its last attempt, is assumed to be much smaller than the number of channels. As explained in the channel model, this, combined with the large population, makes the probability of success of any given copy effectively independent of the number of copies of the same packet that were transmitted at the same time.

Let E denote the expected number of copies for each data packet (until success or deadline). Then, the throughput S is

$$S = \frac{G \cdot (1 - P_e)}{E}. \tag{6}$$

Therefore, if G and P_e are held constant, minimizing E will maximize S. The channel capacity will be the maximized throughput value.

Since failures are independent, the probability of all transmissions failing, P_e, is

$$P_e = (1 - e^{-G})^N, \tag{7}$$

where N is the maximum aggregate number of transmitted copies of a given packet. Alternatively, we can derive N from P_e and G:

$$N = \frac{ln(P_e)}{ln(1 - e^{-G})}. \tag{8}$$

The number of copies transmitted in the ith attempt will be denoted n_i. Therefore,

$$\sum_{i=1}^{D_r} n_i = N. \tag{9}$$

The ith transmission attempt will occur iff all preceding attempts fail. The probability of this event is

$$Pr(ith\ attempt) = (1 - e^{-G})^{\sum_{j=1}^{i-1} n_j},\ \ i = 2, ..., D_r \tag{10}$$

and

$$Pr(1st\ attempt) = 1. \tag{11}$$

Transmission proceeds until one of the attempts is successful or until the deadline. Therefore, the expected number of copies transmitted is

$$E = \sum_{i=1}^{D_r} n_i \cdot (1 - e^{-G})^{\sum_{j=1}^{i-1} n_i}. \tag{12}$$

For the first method (c copies per attempt), the results are:

$$P_e = \left(1 - e^{-G}\right)^{c \cdot D_r}, \tag{13}$$

$$E = c \cdot \left(1 + (1 - e^{-G})^c + ... + (1 - e^{-G})^{c(D_r - 1)}\right) = \\ = c \cdot \frac{1-(1-e^{-G})^{cD_r}}{1-(1-e^{-G})^c}. \tag{14}$$

The second method (single copy in all but the last attempt and c copies in the last one) results in:

$$P_e = (1 - e^{-G})^{D_r + c - 1}, \tag{15}$$

$$E = 1 + (1 - e^{-G}) + ... + (1 - e^{-G})^{D_r - 2} + \\ + c(1 - e^{-G})^{D_r - 1} = \\ = e^G \cdot (1 - (1 - e^{-G})^{D_r}) + (c - 1)(1 - e^{-G})^{D_r - 1}. \tag{16}$$

Given D_r and P_e, we now proceed to minimize E using a dynamic programming approach. Since N is not given yet is required for the dynamic programming method, we will iterate over its increasing integer values, using dynamic programming as a "solver" to find minimum E (and maximum S) when N, D_r and P_e are held constant. It will be assumed, without proof, that a single local maximum (over N) exists for $S(N)$, in which case the iteration will proceed until the maximum throughput (which is also the channel capacity) starts to decrease. Even if this is not the case, N cannot exceed the product of D_r and the number of transmitters per station, which is normally very small (more on this later). Consequently, the range of the search will be very limited in practical settings. (Also, a more efficient search order can be used.)

In dynamic programming terms, the number of attempts used so far will be the system **stage**. The vector of number of copies per attempt n_i is the **state** of the system. The **decision** made at each attempt is how many copies should be transmitted out of the remaining budget. After the decision, the state variables undergo a **transformation** whereby the selected number of copies is appended to the vector. Our **return function** is the expected number of transmissions so far which will be denoted as $E_t(n)$. t and n will denote the iteration variables, $1 \leq t \leq D_r$ and $1 \leq n \leq N$.

To use the Optimality Principle, which states that the dynamic programming technique will find a global optimum [15], two conditions must be met: 1) the objective function must be separable in the sense that the effect of the final stage on the objective function depends only on the previous state and the last decision. 2) state separation property: after a decision is made, the next state depends upon the previous state and the decision.

Condition 1 is met since the expected contribution of a possible transmission of x copies after $\sum n_i$ previous transmissions is $x \cdot (1-e^{-G})^{\sum n_i}$, which depends only on the state variables n_i and the decision variable x. From the transformation definition, our next state vector depends only on the previous state vector and on the decision made, so the second condition holds true.

The recurrence equation is

$$E_t(n) = min_{i=1..n-t+1}\left(i \cdot (1-e^{-G})^{n-i} + E_{t-1}(n-i)\right), \qquad (17)$$

with the boundary condition

$$E_1(n) = n. \qquad (18)$$

The optimal policy for transmitting n packets in t attempts is composed by abutting an optimal sub-policy for transmitting fewer than n packets in $t-1$ attempts, with the transmission of the remaining copies in the last attempt.

Thus, we have found a recursive equation for the minimum expected number of transmissions of N packets in D_r attempts, $E_{D_r}(N)$.

Numerical results

The probability of failure was held equal for all schemes, and in each of the two parameterized schemes c was chosen to maximize throughput. Numerical results have been obtained for several values of P_e, for $D_r = 3$ and for $D_r = 5$. Sample results are provided in Table 2, and more are plotted in Figures 1 and 2.

For $P_e = 0.001$, the optimal sequence $(1,1,1,2,5)$ achieves throughput of 0.3213. This is 50% higher than a $(1,1,...,1)$ sequence, 23% higher than a $(2,2,...,2)$ sequence and 3% higher than the $(1,1,...5)$ sequence. The advantage of the optimal solution is even more pronounced for lower values of P_e.

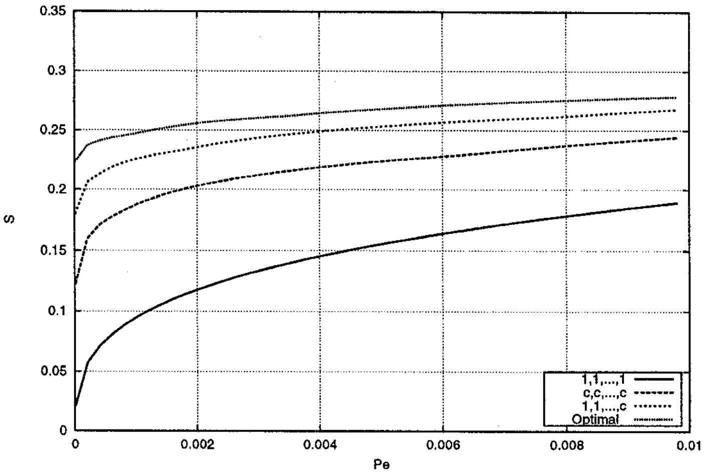

Fig. 1. Attainable throughput versus P_e for multi-round transmissions on a multichannel with a deadline. 3 rounds are permitted.

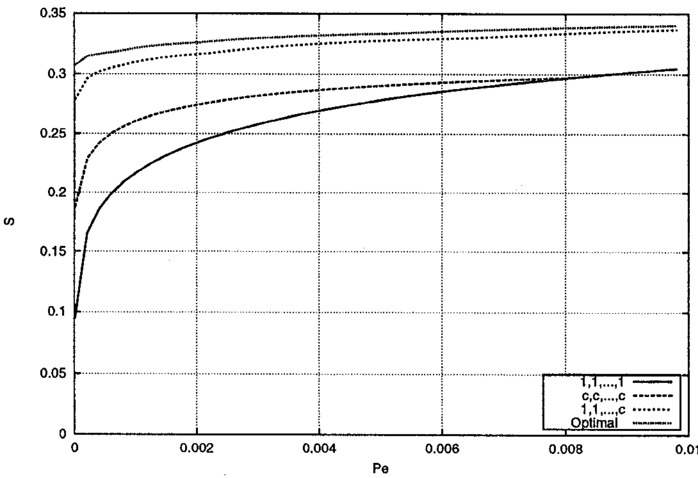

Fig. 2. Attainable throughput versus P_e for multi-round transmissions on a multichannel with a deadline. 5 rounds are permitted.

Remarks.

1) The maximum capacities of the different schemes (even for the same probability of failure and deadline) are obtained with different values of G (different probability of each copy colliding), so the maximum total number of copies need not be equal.
2) We see that our optimal method achieves the highest throughput for a given P_e. The advantage becomes more pronounced as the permitted error probability is reduced.

A limited number of transmitters

In practice, the number of transmitters (and hence concurrent transmissions by a single station) is severely limited. We denote this limit by K. The foregoing analysis is next modified to reflect this limitation.

$$E_t(n) = min_i\left(i \cdot (1-e^{-G})^{n-i} + E_{t-1}(n-i)\right), \qquad (19)$$

with i constrained to the range

$$i = max(1, n-K\cdot(t-1))..min(K, n-t+1), \qquad (20)$$

Pe	D_r	$S(1,1,...,1)$	$S(c,c,...,c)$	c	$S(1,1,...,c)$	c	S(opt)	Sequence(opt)
0.01	3	0.1904	0.2447	2	0.2683	3	0.2787	[1 2 4]
0.01	5	0.3056	0.3056	1	0.3374	3	0.3404	[1 1 1 2 3]
0.001	3	0.0948	0.1872	3	0.2254	5	0.2470	[2 3 7]
0.001	5	0.2166	0.2604	2	0.3098	5	0.3213	[1 1 1 2 5]
0.0001	3	0.0453	0.1488	4	0.1984	6	0.2330	[2 3 10]
0.0001	5	0.1452	0.2186	3	0.2913	7	0.3125	[1 1 2 3 8]

Table 2

Attainable-throughput comparison among various retransmission sequences on a multichannel with up to D_r transmission attempts (rounds) and an error probability P_e.

and the boundary condition

$$E_1(n) = n. \tag{21}$$

Comparable results to the unconstrained case are presented in Tables 3 and 4. The number of concurrent transmissions is limited to $K = 2,3$. For larger values of P_e, the limit on the number of concurrent transmissions does not hurt performance.

ECC-based schemes

In this section, we have established the very substantial benefits offered by using a higher degree of redundancy in the last retransmission attempts rather than following the common wisdom that is suitable for an unconstrained delay. In order to avoid confusion with other issues and to demonstrate the approach in as practical a setting as possible, we used the simplest form of redundancy, namely replication. Nonetheless, it is possible to use more general redundancy techniques, such as the the codes discussed in Section III.

V. Comments on Assumptions

In the last two sections, we have presented some promising schemes that can very substantially increase the effective capacity of multichannel ALOHA networks in deadline-constrained operation. In this section, we briefly review the main simplifying assumptions that were made and comment on them in order to try and assess the practical value of our findings.

Offered load

The results in this paper were obtained under an assumption that the load on the network is known to the stations, which can judiciously tailor the redundancy to the load. In practice, the offered load can be estimated by the stations or by a central hub.

Stability and control policy

The ALOHA protocol was previously shown to be unstable [16] and [17]. To operate the channel at a stable operating point, a higher-level control policy is needed. This policy will influence the performance obtained from multi-copy techniques. In spite of this, we conjecture that much of the performance gain obtained by using the techniques in this work will be retained when the control policy is considered.

Independent collisions

We have assumed that given enough channels, collisions can be considered independent. This assumption is not accurate for the case wherein several copies are transmitted over tens of channels, as in practical systems today. The dependency among multiple collisions in such cases will lower the performance gains seen in this work, and an exact analysis is warranted. Still, we expect to achieve much of the performance gains of the "ideal", independent case even in practical system with tens of channels.

VI. Conclusions

We have shown how to judiciously exploit redundancy in order to substantially increase the capacity of multi-channel ALOHA networks with deadlines and permissible probabilities of not meeting them. These are realistic, user-oriented requirements.

For a single round of transmissions on a multi-channel, we focused on a scenario that is typical of geostationary satellites and their VSAT ground stations. We showed that the combination of replicated preambles and lower-overhead error-correction for the data portion results in a dramatic increase of the attainable throughput subject to a required probability of success. This is an adaptation of redundant dispersity routing to this situation. As an example, throughput of 0.18 (per channel) with 99% probability of successful attempt was shown, as compared with throughput of 0.01 for conventional ALOHA. Further improvement of this result is possible by optimizing the code selection.

Optimal replication-based multi-round retransmission policies were devised for the multi-channel. Most important, it was shown that the best policy in the case of a deadline is to transmit one or very few copies of the packet at a time until the last transmission attempt. Then, a burst of packets is transmitted. This method sharply increases the attainable throughput for a given probability of failing to meet the deadline, and the increase is greater when the permissible P_e is smaller. We addressed the practical constraint of a limited number of transmitters per station and showed one approach for accommodating it. Another approach, whereby a round is "stretched" in order to permit transmission in more than one time slot per round (a time-hardware trade-off) is presently being investigated with very promising preliminary results. Yet another approach, whereby the rounds are "pipelined", may also hold some promise.

One interesting direction for continued research on this topic entails the application of more general error-correction techniques to the multi-round case. A first step would entail the application

Pe	$S_{opt}(K=2)$	Seq.	$S_{opt}(K=3)$	Seq.	$S_{opt}(K=\infty)$	Seq.
0.01	0.2614	[1 2 2]	0.2772	[1 2 3]	0.2787	[1 2 4]
0.001	0.1884	[1 2 2]	0.2198	[1 2 3]	0.2470	[2 3 7]
0.0001	0.1302	[1 2 2]	0.1716	[1 3 3]	0.2330	[2 3 10]
0.00001	0.0877	[1 2 2]	0.1355	[1 3 3]	0.2232	[2 4 13]

Table 3

Attainable-throughput comparison among the optimal retransmission sequences on a multichannel with up to 3 transmission attempts (rounds), error probability P_e and up to K concurrent transmissions.

Pe	$S_{opt}(K=2)$	Seq.	$S_{opt}(K=3)$	Seq.	$S_{opt}(K=\infty)$	Seq.
0.01	0.3378	[1 1 1 2 2]	0.3404	[1 1 1 2 3]	0.3404	[1 1 1 2 3]
0.001	0.2954	[1 1 2 2 2]	0.3135	[1 1 2 3 3]	0.3213	[1 1 1 2 5]
0.0001	0.2471	[1 1 2 2 2]	0.2828	[1 1 2 3 3]	0.3125	[1 1 2 3 8]
0.00001	0.2034	[1 2 2 2 2]	0.2489	[1 1 3 3 3]	0.3071	[1 1 2 3 11]

Table 4

Attainable-throughput comparison among the optimal retransmission sequences on a multichannel with up to 5 transmission attempts (rounds), error probability P_e and up to K concurrent transmissions.

of those to individual transmission attempts (rounds); however, they could also be applied across rounds, allowing the receiver to accumulate sub-packets of the same original data packet.

Finally, it is important to stress that the results obtained in this paper are for user-oriented performance measures. In view of this, the fact that the performance improvements are very substantial, and since the simplifying assumptions appear to have a minor impact, the schemes suggested in this paper may be of practical value in addition to their academic merit.

REFERENCES

[1] L. G. Roberts, "ALOHA packets, with and without Slots and Capture", *Comp. Commun. Review* 5 (1975), pp. 28-42.

[2] L. Kleinrock and S. S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation", *IEEE Trans. Commun.*, vol. COM-23, no. 4, pp. 410-423, Apr 1975.

[3] Y. W. Leung, "Generalized multicopy ALOHA", *Electronics Letters*, vol. 31, no. 2, pp. 82-83, 19 Jan 1995.

[4] E. W. M. Wong and T-S P. Yum, "The Optimal Multicopy Aloha", *IEEE Trans. Auto. Control*, vol. 39, no. 6, pp. 1233-1236, June 1994.

[5] Y. Birk and Y. Keren, "Optimal inter-copy delay for dual-copy transmissions in ALOHA with no feedback", CC-PUB 234 (EE-PUB 1134), Technion, Jan. 1998.

[6] Y. Birk and Y. Keren, "Redundant transmissions and retransmission scheduling for improved throughput–delay in ALOHA networks", CC-PUB 209 (EE-PUB 1109), Technion, Dec. 1997.

[7] N. Abramson, "THE ALOHA SYSTEM - Another alternative for computer communications", *AFIPS Conference Proc., 1970 Fall Joint Comp. Conf.*, vol 37, pp. 281-285.

[8] N.F. Maxemchuk, "Dispersity Routing", *proc. Int. Commun. Conf.*, pp. 41.10-41.13, 1975.

[9] M.O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault tolerance", *J. ACM*, vol. 36, pp. 335-348, Apr. 1989.

[10] Y. Birk and N. Bloch, "Prioritized dispersal: a scheme for selective exploitation of redundancy in distributed systems", *Proc. 8th Israeli Conf. on Computer Sys. And Software Engr.* (ISySE'97), June 1997, pp.77–85.

[11] E. S. Sousa and J. A. Silvester, "Spreading Code Protocols for Distributed Spread-Spectrum Packet Radio Networks", *IEEE Trans. Commun.*, vol. 36, no. 3, pp. 272-281, March 1988.

[12] S. W. Kim and W. Stark, "Optimum Rate Reed-Solomon Codes for Frequency-Hopped Spread-Spectrum Multiple-Access Communications Systems", *IEEE Trans. Commun.*, vol. 37, no. 2, pp. 138-144, Feb 1989.

[13] M. B. Pursley, "Frequency-Hop Transmission for Satellite Packet Switching and Terrestrial Packet Radio Networks", *IEEE Trans. IT*, vol. IT-32, no. 5, pp. 652-667, Sep. 1986.

[14] E. Lutz, "Slotted ALOHA Multiple Access and Error Control Coding for Land Mobile Satellite Networks", *Int. J. of Satellite Commun.*, vol. COM-10, pp. 275-281, 1992.

[15] L. Cooper and M.W. Cooper, "Introduction to Dynamic Programming", Pergamon Press, pp. 31-44.

[16] A. B. Carlelial and M. E. Hellman, "Bistable Behavior of Aloha-Type Systems", *IEEE Trans. Commun.*, vol. COM-23, no. 4, pp. 401-409, Apr 1975.

[17] W. A. Rosenkrantz and D. Towsley, "On The Instability of the Slotted ALOHA Multiaccess Algorithm", *IEEE Trans. Auto. Control*, vol. AC-28, no. 10, pp. 994-996, Oct 1983.