SOME NEW APPROACHES TO RANDOM-ACCESS COMMUNICATIONS

James L. MASSEY

Institute for Signal and Information Processing
Swiss Federal Institute of Technology
8092 Zurich, Switzerland

Random-accessing is defined as any technique to accomplish unscheduled seizure of a many-user communications channel; its purpose is to reduce transmission delay below what can be achieved by scheduled-accessing or by channel division. Some general principles regarding channel division, channel seizure, and the effect of feedback are formulated. The "classical" approach to random-accessing, i.e., ALOHA-like techniques, is seen to be subject to instability. A newer approach, collision-resolution algorithms (CRA's), is shown to avoid this problem. The analysis of CRA's has led to bounds on the performance of any random-access system that are briefly discussed. Two new approaches to random-accessing without feedback information are described, viz., protocol sequences for the M-user collision channel and coding for the M-active-out-of-T-user collision channel. Examples are generously used throughout the paper, and some speculations on the practicality of the new approaches are offered.

1. INTRODUCTION

Before describing "new approaches to random-access communications", we should make clear what we mean by "random-accessing" and what we see as its main purpose. To do this, we must first say a few words about "multiple-accessing" in general.

A multiple-access technique is any technique that permits two or more senders to operate on a single communications channel. Time-division multiple-accessing (TDMA), frequency-division multiple-accessing (FDMA) and code-division multiple-accessing(CDMA) are well-known multiple-access schemes of the channel-division type: i.e., they divide the single channel into many "smaller" channels, one for each sender. This division may be fixed, or it may be adjusted from time to time to correspond to the changing needs of the senders as in so-called "demand assignment" schemes. A second class of multiple-access schemes is that of what we shall call the channel-seizure type. In this type of multiple-accessing, a single sender can use the full (time and frequency) resources of the channel for himself alone on some sort of temporary basis. An example of a channel-seizure scheme is a token-ring in which, when the "token" arrives at a sender's station on the ring, that sender can remove the token, send his own message as if he were the only sender on the ring, and then reinsert the token.
A random-access technique can be defined as a multiple-accessing scheme of the channel-seizure type (i) in which it can happen that two or more senders may simultaneously attempt to seize the channel, and (ii) which provides in some way for the recovery from such "access conflicts". In a random-access system, a sender generally "takes a chance" when he attempts to seize the channel, and he relies on the access protocol to repair the damage when he encounters "bad luck".

In some communication scenarios (as we shall see later), access conflicts cannot be avoided. More often, however, it is a matter of choice whether or not to allow access conflicts and hence whether or not to use random-accessing. The obvious question is: why should anyone choose to allow such an obviously bad thing as access conflicts? The answer can be put as a second question: why should anyone demand that a sender always wait for a guarantee of exclusive access before he attempts to seize the channel? When traffic on the channel is

Reprinted with permission from *Performance '87*, pp. 551–569, 1988.

Pet., Exh. 1017, p. 1

light, the bold sender will be almost sure to succeed in his gamble for access and can thus avoid the delay that a timid sender would incur. The primary purpose of random-accessing is to reduce the delay between the time that a sender obtains an information input and the time that he transmits this information successfully over the channel. Random-accessing is a gamble, but one in which the odds can be on the side of the player rather than on the side of the "house".

In Section 2 of this paper, we show why channel seizure is generally preferable to channel division for multiple-accessing, and we examine the role of channel feedback information. Section 3 describes the ALOHA approach to random-accessing and points out its virtues and defects. In Section 4, we describe one new approach to random-accessing, viz. collision resolution, and we contrast it with the ALOHA approach. Section 5 considers certain general bounds on the throughput of random-access schemes. Section 6 describes two new approaches to random-accessing without feedback. Some concluding remarks are given in Section 7.

2. SOME GENERAL MULTIPLE-ACCESS PRINCIPLES

The simplest multiple-access channel is surely the two-sender binary adder channel (2SBAC) shown in Fig. 1. Each time instant, each sender sends a binary digit (0 or 1) and the received digit is the sum (0, 1 or 2) of these two numbers, i.e.,

$$Y_n = X_{1n} + X_{2n}$$

where X_{1n} and X_{2n} are the binary digits sent by senders 1 and 2, respectively, at time n and Y_n is the received digit. The "wall" shown between the two senders in Fig. 1 signifies that the user on one side is not privy to the information to be sent on the other side, although the two users are allowed in advance to have formulated a common strategy for sending this information.

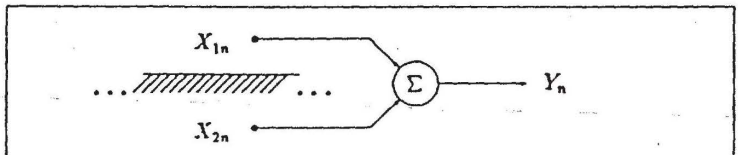

Fig.1 : The two-sender binary adder channel (2SBAC)
$X_{1n} \in \{0,1\}, X_{2n} \in \{0,1\}, Y_n \in \{0,1,2\}$

Fig. 2 shows the pentagonal "capacity region" of the 2SBAC, i.e., the region of rate pairs (R_1, R_2) such that Sender 1 can send data at the rate R_1 (bits per channel use) and Sender 2 can send at the rate R_2, both with arbitrarily small error probability.

It is easy to see how the point $(R_1,R_2) = (1,0)$ on the capacity-region boundary can be achieved. Sender 2 simply always sends 0's (and thus $R_2 = 0$) so that $Y_n = X_{1n}$, and hence Sender 1 can directly send his "raw" information bits over the channel with no need for coding ($R_1 = 1$). The point $(R_1,R_2) = (0,1)$ can be similary achieved. By agreeing to alternate between these two schemes for appropriate periods, Senders 1 and 2 can achieve any point (R_1,R_2) such that $R_1 + R_2 = 1$, i.e., at any point on the "time-sharing line" shown in Fig. 2.

Pet., Exh. 1017, p. 2

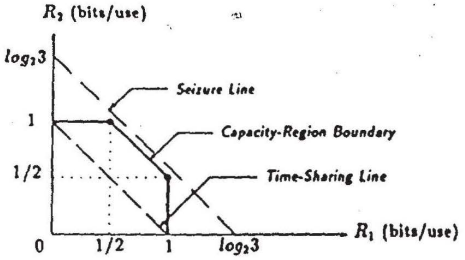

Fig.2 : Capacity Region of the 2SBAC of Fig.1

It is almost as easy to see how the point $(R_1, R_2) = (1, 1/2)$ on the capacity-region boundary can be approached. Sender 1 transmits his raw information bits ($R_1 = 1$). This causes the channel seen by Sender 2 to be that shown in Fig. 3, because, for instance, if Sender 2 should send a 1 then with probability 1/2 Sender 1 will also send a 1 and 2 will be received, while with probability 1/2 Sender 1 will send a 0 and 1 will be received. But the channel of Fig. 3 is the familiar binary erasure channel (in which a received 1 is the "erasure symbol") with erasure probability $\delta = 1/2$ and capacity $C = 1-\delta = 1/2$. Thus, Shannon's noisy coding theorem ensures the existence of a coding scheme that will allow Sender 2 to send information at a rate R_2 arbitrarily close to 1/2 with arbitrarily small error probability. After the receiver has decoded Sender 2's codeword, he can subtract it from the received sequence to obtain the uncoded sequence that was transmitted by Sender 1. The price of making R_2 closer to the capacity 1/2 is an increasingly longer codeword length or, equivalently, a longer delay in recovering the information at the receiver. The point $(R_1, R_2) = (1/2, 1)$ can, of course, be similarly approached. By appropriately alternating between coding schemes, any point (R_1, R_2) on the capacity-region boundary line $R_1 + R_2 = 3/2$ between the points (1, 1/2) and (1/2, 1) can be approached.

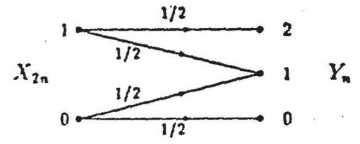

Fig.3 : The binary erasure channel seen by Sender 2 when Sender 1 sends random binary digits over the 2SBAC of Fig.1.

Perhaps the best interpretation of the "wall" shown in Fig. 1 is as a prohibition against seizure of the channel by a single sender. If a single sender is allowed to control both X_{1n} and X_{2n}, then he can by choosing (X_{1n}, X_{2n}) to be (0,0), (0,1) or (1,1) cause Y_n to be 0, 1 or 2, respectively, i.e., he can create a noiseless ternary channel with capacity $\log_2 3$ (bits per use). By alternating appropriately between such seizures, two senders could achieve any point on the "seizure line" shown in Fig. 2 that lies strictly outside the (seizure-prohibited) capacity region.

Pet., Exh. 1017, p. 3

Suppose now that there is a feedback channel from the receiver to the two senders in Fig. 1 so that each sender learns the value of Y_n immediately after X_{1n} and X_{2n} have been sent. The point $(R_1,R_2) = (1, 1/2)$ can now be achieved with the greatest of ease. Sender 1 still sends his raw information bits ($R_1 = 1$) so that Sender 2 still sees the binary erasure channel of Fig. 3. Sender 2, however, can now (because of the feedback of Y_n) simply send each of his information bits repeatedly until it is received "unerased", i.e., until $Y_n = 0$ when this information bit is a 0 or until $Y_n = 2$ when this information bit is a 1. Because the erasure probability δ is 1/2, Sender 2 will be sending information at the rate $R_2 = 1-\delta = 1/2$ bits/use. Moreover, the average delay between first transmission and successful transmission is only 2-1 = 1 time instant. Something even more remarkable, however, results from the availability of feedback (as was first shown by Gaarder and Wolf [1]): points outside the capacity region of Fig. 2 can be achieved! This was quite surprising when first discovered because it had long been known that feedback could not increase the capacity of a single-sender memoryless channel. The actual capacity region of the 2SBAC with feedback was only recently determined by Willems [2]; it differs from the capacity region without feedback, shown in Fig. 2, in that the boundary line between the points (1, 1/2) and (1/2, 1) is bowed slightly outward (but still well away from the "seizure line").

The simple 2SBAC of Fig. 2 is a rich source of lessons about multiple-accessing. With its help, we have been able to illustrate all of the following general principles of multiple-access communications:

(1) Channel seizure, when possible, is the most effective way to utilize a multiple-access channel.

(2) When channel seizure is prohibited, time-sharing (or other types of channel division) generally is still sub-optimum in the sense that it cannot be used to achieve all points in the capacity region.

(3) Feedback, when available, can be exploited to reduce the coding delay and complexity required to achieve a given transmission rate.

(4) When channel seizure is prohibited, feedback can also enlarge the capacity region.

The first of these principles supports the way that computer communications is carried out today. Virtually all newer local area networks (LAN's) operate on a channel seizure basis, sometimes with deterministic access (as in a token ring) and sometimes with random access (as in Ethernet). The third principle suggests that feedback will play an especially crucial role in random-accessing, because some kind of "coding" is absolutely necessary to overcome the losses due to access conflicts.

3. THE ALOHA APPROACH TO RANDOM-ACCESSING

The ALOHA system, devised by Abramson [3] and his colleagues at the University of Hawaii, was the first random-access system; its approach underlies most present-day random-access systems, e.g. Ethernet.To illustrate the ALOHA approach, we now describe the ALOHA system, including the modification of "time slotting" that was introduced by Roberts [4].

Suppose that all data to be sent is in the form of "packets", all of which have the same length (measured in transmission time on the seized channel) that we take to be the unit of time. We define the time interval $(n-1) \leq t < n$ to be the n-th channel "slot". "Time-slotting" means that senders can transmit packets

Pet., Exh. 1017, p. 4

only by beginning transmission at a slot boundary. Thus, transmitted packets from two senders will either overlap completely at the receiver or not at all.

The channel model postulated by Abramson was that, when 2 or more transmitted packets overlap at the receiver, then they mutually destroy one another, but otherwise packets are received error-free. Moreover, there is feedback from the receiver at the end of each slot so that all users learn whether or not a collision occurred (collision/no-collision binary feedback).

The information-generation model postulated by Abramson was that of a very large number (essentially infinite) of identical sources, each with an associated sender, such that the number of new information packets generated during any slot is a Poisson random variable with mean λ (packets/slot), independent of previously generated packets. The essentially infinite number of senders means that access conflicts cannot be entirely avoided, i.e., random-accessing becomes a necessity. [In fact, the original operational ALOHA system had a very small number of transmitters so that random-accessing was a matter of choice, made by Abramson and his colleagues for the express purpose of reducing access delay.]

The random-access protocol devised by Abramson was ingeniously simple. A new packet must be transmitted in the slot immediately following that in which it was generated. When a collision occurs, each "colliding" sender must retransmit in a randomly-selected later slot. Each such sender, of course, independently makes this random selection of retransmission delay.

Abramson's analysis of the ALOHA system was equally ingenious, if not rigorous. He postulated that the retransmission policy could be shaped in such a way that the number of retransmitted packets in any slot would also be a Poisson random variable, independent from slot to slot and independent of the new-packet generation process, with a mean of λ_r (packets/slot). Because the sum of independent Poisson random variables is again Poisson, this implies that the total number of packets transmitted in any slot is also a Poisson random variable with mean $\lambda_t = \lambda + \lambda_r$. Because the throughput τ of successful packets at the receiver is the fraction of slots in which exactly one packet is transmitted, it follows that τ is just the probability that a Poisson random variable with mean λ_t takes on the value 1, i.e.,

$$\tau = \lambda_t e^{-\lambda_t}. \qquad (1)$$

Equation (1), which is the so-called throughput equation for slotted-ALOHA, is shown graphically in Fig. 4. It is easy to check from (1) that τ is maximized when $\lambda_t = 1$ (packet/slot), which seems quite natural, and that this maximum is

$$\tau_{max} = e^{-1} \approx .368 \text{ (packets/slot)},$$

which seems quite fundamental. It is common to say that e^{-1} is the "capacity of the slotted-ALOHA channel", but, as we shall see, this description is misleading.

The reader may (and should) be disturbed by the fact that the new-packet arrival rate λ appears nowhere in the throughput equation (1). To bring λ into the picture, one must invoke the equilibrium hypothesis which states that packets are entering and leaving the system at the same rate, i.e.,

$$\tau = \lambda.$$

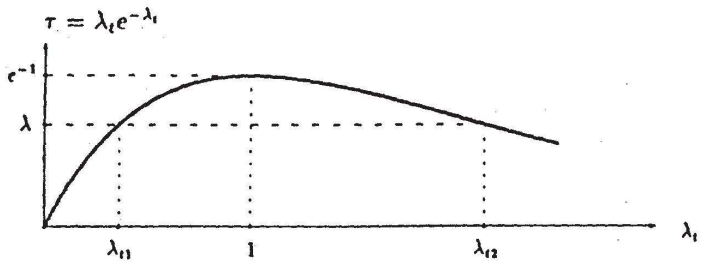

Fig.4 : Graphical depiction of the throughput equation for slotted-ALOHA.

This seems similar to the constancy assumption for the retransmission rate, but in fact neither assumption implies the other. The equilibrium hypothesis is really an expression of the hope that the ALOHA system is stable, i.e., that the queue of packets awaiting retransmission is not steadily growing at a positive rate $\lambda-\tau$ (packets/slot). Such a positive growth rate is not inconsistent with a constant retransmission rate if the retransmission delay is chosen randomly in a way to depend on how many times the given packet has been previously transmitted unsuccessfully. The equilibrium hypothesis should in fact be called the stability hypothesis for ALOHA.

It is easy to argue from Fig. 4 that the ALOHA system cannot be stable for a retransmission policy that does not take into account the number of previous unsuccessful transmissions. Suppose that the arrival rate λ satisfies $\lambda < e^{-1}$ as shown in Fig. 4. If equilibrium prevails, then the traffic rate λ_t will be λ_{t1} as shown in Fig. 4. This is of course an "average" rate and, over any fixed length interval, the actual rate will fluctuate about this mean. If the actual traffic rate moves a little above λ_{t1}, the actual throughput increases a little above λ. Thus, packets leave the system faster than they arrive, which causes the actual traffic rate to move back down to λ_{t1}. Hence, the point $(\tau,\lambda_t) = (\lambda,\lambda_{t1})$ is a conditionally stable point, i.e., it is stable under small fluctuations. But if a large fluctuation causes the actual traffic rate to move to the right of λ_{t2} in Fig. 4, then the actual throughput decreases below λ. Thus, packets leave at a slower rate than they enter, which causes a further increase in the actual traffic rate, a further decrease in actual throughput, etc. The system never returns to the point (λ,λ_{t1}), but rather drifts relentlessly toward the catastrophic "unconditionally stable" point $(\tau,\lambda_t) = (0,\infty)$. The maximum stable throughput of a fixed-policy ALOHA system is 0.

The virtue of the ALOHA approach is its simplicity, its Achilles' heel is its instability. In fact, it is possible to devise retransmission policies that stabilize an ALOHA system, cf. [5], but the protocol then loses its simplicity. Most practical ALOHA-type random-access systems appear in fact to be unstable. These systems incorporate some kind of time-out feature that switches the system to a non-random type of accessing to clear away the backlog of traffic when the channel is jammed with collisions, then switches back again to the basic ALOHA protocol. Again this means some loss of simplicity in the access protocol, as well as some performance anomalies. Published simulations of ALOHA-type systems

invariably appear to have been purged of any anomalies, if indeed any occurred. In fact, there is usually very little information provided with such simulations about what total time period was simulated, whether time-out provisions were included, etc., so that it is generally very difficult to determine the real meaning of the simulated performance results that are presented.

4. COLLISION-RESOLUTION ALGORITHMS

Concern over the instability of most ALOHA-like protocols led some researchers to search for random-access schemes that were provably stable. The breakthrough in these efforts was made in 1977 by J. Capetanakis [6], then an M.I.T. doctoral student working with Prof. R. Gallager, and independently achieved shortly thereafter by two Soviet researchers, B. Tsybakov and V. Mikhailov [7]. The essence of their contributions was the "collision-resolution approach" to random-accessing, which we now consider.

The channel model assumed for collision resolution is the same as that for slotted-ALOHA, namely a time-slotted collision-type channel with some form of feedback to the senders at the end of each slot. We will assume the same binary (collision/no-collision) feedback as for slotted-Aloha [as Capetanakis also assumed; Tsybakov and Mikhailov considered ternary (collision/success/idle) feedback]. The information-generation model is the same as for slotted-ALOHA, i.e., essentially-infinitely many identical sources, each with an associated sender, such that the number of new packets generated in each slot is an independent Poisson random variable with mean λ.

A collision-resolution algorithm can be defined as a random-access protocol such that, whenever a collision occurs, then at some later time (provided λ is not too large) all senders will simultaneously learn from the feedback information that all packets involved in that collision have now been successfully transmitted. The crux of collision resolution is the exploitation of the feedback information to control the "random" retransmission process in such a way that chaotic retransmission can never occur. Because there is no upper bound on the number of packets that initially collide, it was not at all obvious that collision-resolution algorithms existed before the first such algorithms were presented by Capetanakis and by Tsybakov and Mikhailov.

As an example of a collision-resolution algorithm, we now describe the binary stack algorithm, which is essentially Capetanakis' binary "tree algorithm", but we prefer the terminology "stack algorithm" introduced by Tsybakov and Mikhailov. The terminology "binary" stems from the fact that every sender is assumed to have a fair "binary coin"(with "0" on one side and "1" on the other) which he flips whenever his packet is involved in a collision. The term "stack" comes from the fact that one can conveniently visualize the operation of the algorithm in terms of the conceptual stack shown in Fig. 5.

Suppose at the outset that the stack is empty, i.e., that $S = 0$. Suppose also that the access gate is opened and some number X of senders then enter the "current sending box", which is just the conceptual location of all senders who will send packets in the current slot. Perhaps $X = 2$ or $X = 0$ or even $X = 6 \times 10^{23}$; for the moment, we assume only that $X \geq 2$ so that a collision ("C") occurs in this slot. Then these colliding senders flip their binary coins, those who flip "0" remain in the "current sending box" (i.e., they again transmit in the next slot) while those who flip "1" are pushed down (conceptually)

Pet., Exh. 1017, p. 7

into level 1 of the stack. The stack size is now S = 1. The general rule is: if "C", then S ← S + 1. One sees that now about X/2 of the original X colliding senders will remain in the "current sending box" while about X/2 will be pushed down into the stack. For the moment, we assume the blocked-access protocol, which states that the "access gate" is closed at the initial collision and remains closed until all senders learn that the original X colliding packets have all been successfully transmitted. The same process of stack growth and concomitant "thinning out" of the "current sending box" continues until the feedback "no-collision" ("NC") occurs, which means that either 1 sender or none had been in the "current sending box". This is the signal for the stack to be pushed upward one level so that senders who were in level 1 of the conceptual stack are now again in the "current sending box" and the stack size is reduced by 1. The general rule is: if "NC" and S > 0, then S ← S − 1. After some time, because the above process will "thin out" crowded levels in the stack, the stack size will again reach S = 0. If now no collision occurs, this means that all of the original X senders must have successfully sent their packets. (If a collision occurs, the previous process continues.) The general rule is: if "NC" and S = 0, then the collision is resolved.

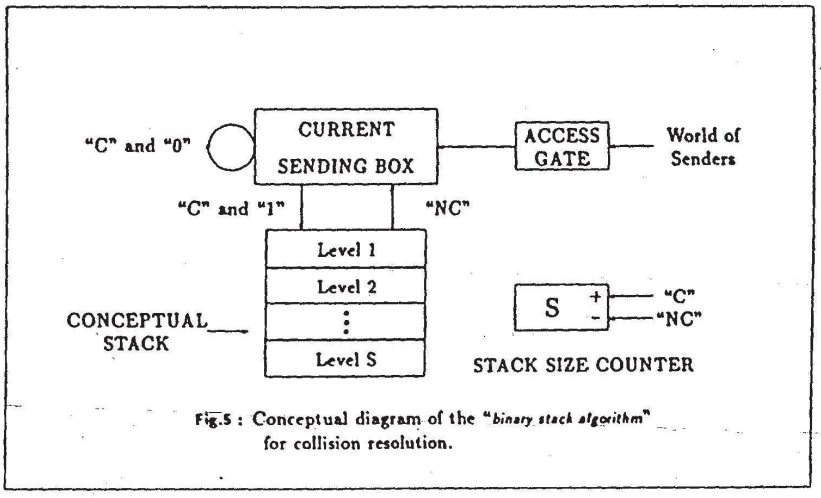

Fig.5 : Conceptual diagram of the "*binary stack algorithm*" for collision resolution.

The binary stack algorithm is clearly simple to implement. When involved in a collision, a sender need only generate a binary random variable ("0" or "1") so a minimum of retransmission "randomization" is required. His only other requirements are to maintain two counters, one of which gives his own position in the stack (if he were a party to the collision) while the other keeps track of the stack size S so that he knows when the collision has been resolved. The big question of course is: how effective is this random-access protocol?

Perhaps the main theoretical advantage of the collision-resolution approach over the ALOHA approach to random-accessing is that the former lends itself to precise (and reasonably simple) analysis as we now demonstrate for the binary stack algorithm with blocked-access. Letting Y be the number of slots needed to resolve the original collision of X senders, then the quantity of principal interest is $L_N = E[Y|X = N]$, the average number of slots needed to resolve a collision of N transmitters. We see that $L_0 = L_1 = 1$ as then there is no initial

Pet., Exh. 1017, p. 8

collision. We see further that

$$L_2 = 1 + \frac{1}{2}(L_0 + L_2) + \frac{1}{2}(L_1 + L_1)$$

as, after the initial collision, with probability 1/2 the two senders will flip the same binary number leaving no one in the "current sending box" and both in level 1 (or vice versa), while with probability 1/2 they will flip different binary numbers leaving 1 sender in the "current sending box" and 1 sender in level 1 of the stack. Solving for L_2 gives

$$L_2 = 5$$

slots required on the average to resolve a collision of 2 packets. It is easy to write the general recursion for L_N and to show that the solution satisfies

$$2.8810 < \frac{L_N}{N} + \frac{1}{N} < 2.8867, \quad N \geq 4; \qquad (2)$$

the interested reader is referred to [8] for details of this argument. The conclusion to be drawn from (2) is quite remarkable: whenever the initial collision is moderately large, then just about 2.89 slots will be required to "service" each of the packets involved in this initial collision. This means that the algorithm will be stable (the "server" will not drop hopelessly behind in serving customers) provided only that

$$\lambda \leq \frac{1}{2.8867} \approx .346 \text{ (packets/slot)}$$

whereas it will be unstable if

$$\lambda \geq \frac{1}{2.8810} \approx .347 \text{ (packets/slot).}$$

Thus, the maximum stable throughput of the binary stack algorithm with blocked-access is just about .346 packets/slot. Moreover, this stability holds not only for the assumed Poisson arrival process, but for virtually any arrival process that can be characterized by an average arrival rate λ (packets/slot). This robustness (or, equivalently, insensitivity to the statistics of the arrival process) is, or should be, an attractive practical feature of many collision-resolution algorithms.

To examine the role placed by the "blocked-access protocol" in the above analysis, we first observe that the binary stack algorithm never makes any assumption in advance about the occupancy of the "current sending box" or any of the S stack levels. There could just as well be none or 6×10^{23} senders in any of these locations as far as the binary stack algorithm is concerned. This means that there is no reason to block new senders from entering the "current sending box" at any time. The free-access protocol leaves the "access gate" of Fig. 5 open at all times. Free-access has the practical advantage that senders need to monitor the channel feedback only after they become "active" in the transmitting process. Intuition suggests that free-access should also give a better throughput than blocked-access. Unfortunately, free-access also complicates the analysis, but Mathys and Flajolet have shown that the binary stack algorithm with free-access has a maximum stable throughput of $\lambda = .360$ (packets/slot). More surprisingly, they showed also that the ternary stack algorithm (in which senders flip a fair three-sided coin after a collision, moving into levels 1 or 2 if they flip "1" or "2" while S is increased by 2) with free-access has an even larger maximum stable throughput of

$$\lambda = .401 \text{ (packets/slot).}$$

Pet., Exh. 1017, p. 9

(Because this maximum stable throughput exceeds $e^{-1} \approx .368$, we see that it is indefensible to call e^{-1} the "capacity of the slotted-ALOHA channel".) Mathys and Flajolet also showed that this ternary stack algorithm with free-access has a better delay vs. throughput characteristic than does either the binary stack algorithm or the usual ALOHA algorithm (when analyzed assuming optimistically that the equilibrium hypothesis holds. [9]

Many other collision-resolution algorithms have been proposed that have maximum stable throughputs exceeding .401 (packets/slot) -- the current record for binary (collision/no-collision) feedback is .4493 packets/slot [10]. However, the ternary stack algorithm with free-access appears to us to be the best practical choice by virtue of its simplicity, its robustness and its relatively high maximum stable throughput. It also seems to us to be a better practical choice than any ALOHA-like algorithm for random-accessing on the collision channel with feedback, and we wonder why algorithms of the latter type are still being proposed for new random-access systems.

The reader interested in delving more deeply into the mathematics needed for a precise analysis of collision-resolution algorithms will find the recent book [11] by Hofri to be a useful source of information.

5. UPPER BOUNDS ON MAXIMUM STABLE THROUGHPUT

Our discussion of collision-resolution algorithms may have raised the question in the reader's mind: what is the capacity of the collision channel with feedback or, equivalently, what is the largest possible maximum stable throughput that can be achieved for Poisson arrivals? It is known that the answer depends in general on the kind of feedback available so we continue to consider only binary (collision/no-collision) feedback. Some ingenious and complex arguments have been used to obtain upper bounds on the maximum stable throughput. Rather than describing the best bound, we illustrate the general idea by describing a very simple bound due to Kelly [12].

Kelly's bound actually applies only to algorithms (such as the ALOHA algorithm or any collision-resolution algorithm with free-access) with the immediate-first-transmission property that a newly-generated packet must be transmitted in the slot immediately following that in which it was generated. Suppose such an algorithm is operating stably for Poisson new arrivals with a mean of λ (packets/slot). Let p_r be the fraction of slots in which at least one packet is retransmitted. Then, because the number of new senders in any slot is independent of the number of retransmitted packets in that slot, it follows that the fraction of slots with exactly one packet (i.e., the throughput τ) satisfies

$$\tau \leq \lambda e^{-\lambda}(1-p_r) + p_r e^{-\lambda}$$

with equality when at most 1 packet is retransmitted in any slot (i.e., perfect scheduling of retransmissions). But stability implies $\tau = \lambda$ and thus

$$\lambda \leq \lambda e^{-\lambda}(1-p_r) + p_r e^{-\lambda} \qquad (3)$$

if the system is stable. It is readily checked that the choice $p_r = 1$ maximizes λ for which (3) can be satisfied. Thus

$$\lambda \leq e^{-\lambda} \qquad (4)$$

is required for stability. The largest λ satisfying (4) is Kelly's bound on

Pet., Exh. 1017, p. 10

maximum stable throughput, namely

$$\lambda_{max} \leq .5671 \text{ packets/slot.} \quad (5)$$

that holds for any random-access algorithm with immediate-first-transmission. In fact, our argument has never used the assumption of binary (collision/no-collision) feedback so that Kelly's bound (5) applies to any random-access algorithm with immediate-first-transmission on the collision channel with any kind of feedback.

By using similar but much more intricate arguments, Tsybakov and Likhanov [13] have recently proved that, for any random-access algorithm, the maximum stable throughput on the collision channel with binary (collision/no-collision) feedback satisfies

$$\lambda \leq .5683 \text{ (packets/slots)} \quad (6)$$

for a Poisson new arrival process. This is significantly greater than the largest stable throughput yet achieved, .4493 (packets/slot). The "capacity" of the ALOHA channel with binary (collision/no-collision) feedback lies somewhere between .4493 and .5683 (packets/slot), which is the most that can be said today.

The thoughtful reader may well ask: why are we giving so much attention to maximum stable throughput when the real purpose of random accessing is small (average) delay? The incomplete answer is this. The maximum stable throughput of a random-access algorithm is the smallest throughput where the (average) delay becomes infinite. Thus if one algorithm has a larger maximum stable throughput than another, then it will also have a better delay-throughput characterstic for all sufficiently large throughputs. The complete answer is that one hopes that if the first algorithm is reasonably simple (so that the large maximum stable throughput was not achieved by "trickery" that used high arrival rates to special advantage) then the first algorithm will have a better delay-throughput characteristic for all throughputs. The previous discussion of the ternary stack algorithm shows that there is some justification for this hope.

6. RANDOM-ACCESS WITHOUT FEEDBACK

We now consider some quite recent developments in random-accessing that deal with the situation where there is no feedback to notify senders whether or not their packets have suffered collisions. At first glance, it might seem that random-accessing would be impossible in this situation. The "trick" that makes it possible is for the senders to send redundant packets so that the information packets can still be recovered at the receiver when some of the packets are lost through collisions.

6.1 The M-Sender Collision Channel without Feedback

Again we assume a time-slotted collision channel, but now with no feedback to the senders. Rather than essentially infinite, we suppose there is a given number M ($M \geq 2$) of senders, each with its own information source. We further assume that, although the senders are slot-synchronized, they are unsynchronized at any higher level, i.e., they may all have a different idea of which slot if slot 1. The senders can be thought of as having clocks that "tick" together at

Pet., Exh. 1017, p. 11

clock boundaries but are otherwise unsynchronized. Because there is no feedback from the channel, the senders can never attain any further synchronization of their clocks. The senders each have a "protocol sequence generator" that emits a binary digit at each clock tick; the sender sends a packet if this bit is a 1 and keeps silent in that slot if this bit is a 0. The task is to choose the protocol sequences for the M users in such a way that, by proper coding of the information packets, the receiver can reconstruct the information packets from each user regardless of relative time shifts of the protocol sequences (corresponding to the senders' different understanding of which slot is the first slot). This scenario describes the M-sender collision channel without feedback that was introduced by this writer in 1982 [14]. The capacity C_M is defined as the maximum rate R in packets/slot such that each user can send information packets at a rate at least R/M and the receiver can reconstruct these packets at the receiver with arbitrarily small error probability. In [14], it was shown that

$$C_M = \left(1 - \frac{1}{M}\right)^{M-1} \quad \text{(packets/slot)} \tag{7}$$

and moreover that zero-error probability could be achieved at this rate. Note that

$$\lim_{M\to\infty} C_M = e^{-1} \approx .368$$

so we now see that e^{-1} is indeed a capacity, but not that of the slotted-ALOHA channel where feedback is present, but rather that of the M-user collision channel without feedback when M is essentially infinite. Rather than to describe the general argument that leads to (7), we illustrate the main ideas by considering the special case M = 2.

For M = 2 users, capacity is achieved by choosing periodic protocol sequences with period 4 whose first periods are

[1, 1, 0, 0] Sender 1
[1, 0, 1, 0] Sender 2.

(Note that Sender 2's protocol sequence actually has least period 2.) The coding scheme is very simple, the Sender simply transmits each packet twice, namely at the two positions where there are 1's in the next period of his protocol sequence. Because Sender 1 always sends these two packets in adjacent slots, it follows that, whatever is the time offset between the two protocol sequences, exactly 1 of these packets will be lost in a collision with a packet from Sender 2. The same conclusion holds for the two packets sent by Sender 2; exactly one is lost in a collision. Moreover, the receiver knows to whom a successfully received packet belongs; if it is adjacent to a collision it came from Sender 1, otherwise it came from Sender 2. Each sender thus sends information error-free at a rate of 1 packet every 4 slots so that the senders achieve R = 1/4 + 1/4 = C_2 = 1/2 packet per slot.

For larger M, the scheme to achieve C_M becomes much more complex. In fact, the protocol sequences have period M^M and rather sophisticated coding schemes are required. More interesting and somewhat surprisingly, the capacity is still given by (7) even when the assumption of slot synchronization is removed [14], i.e., when the M senders are completely unsynchronized. The paper by Massey and Mathys [15], which also treats the generalization to the case when the M users wish to send information at different rates, gives full details of the necessary arguments.

Pet., Exh. 1017, p. 12

It is illuminating to think upon the M = 2 user scheme described above as a way of creating a deterministic server for each sender that services information packets at the rate of 1 packet every 4 slots. Thus, if the sender is receiving new packets from his information source at any rate $\lambda/2 < 1/4$ (packet/slot), then the queue at that sender will not grow without bound. Thus, the system will be stable if the total arrival rate λ is equally distributed between the two users and satisfies $\lambda < 1/2 = C_2$ (packets/slot). Conversely, if $\lambda > 1/2 = C_2$, then the system will be unstable. Thus, C_2 is the maximum stable throughput for traffic equally divided between the two users. The similar conclusion holds for all $M \geq 2$.

6.2 The M-out-of-T-Sender Collision Channel without Feedback

A very interesting generalization of the previously described model for random-accessing without feedback was made by Bassalygo and Pinsker [16]. Their model differs from that of the M-Sender collision channel without feedback only in that there is assumed to be a total of T senders, but only at most M of these senders (in advance it is unknown which M) happen to have active information sources. We write C_M^T to denote the capacity of this "M-out-of-T-sender collision channel without feedback".

By using random coding arguments, Bassalygo and Pinsker proved the quite surprising fact that, when M is fairly large, then $C_M^T \approx e^{-1}$. In other words, it costs almost nothing in the achievable maximum stable throughput to have designed the protocol sequences for many more senders than will actually be using the channel actively. The price of increasing T for fixed M is rather an increase in the "complexity" of the necessary protocol sequences (i.e., an increase in the period of these sequences) and a concomitant increase in the complexity of the scheme for coding the information packets. As is typical for random coding arguments, the work of Bassalygo and Pinsker does not provide any specific schemes for achieving any given throughput less than C_M^T, but rather provides a proof of their existence. Thus, it seems especially appropriate here to illustrate the ideas involved in random-accessing on the M-out-of-T-sender channel, by describing a very specific scheme.

For our example, we take T = 3 and M = 2. We begin by choosing protocol sequences of period N = 19 for the 3 possible senders. We number the N = 19 locations in the first period from 0 to N-1 = 18 and represent the first period by those locations where this binary sequence of length N contains 1's. We choose the first periods of the protocol sequences for senders 1, 2 and 3 to be (0, 1, 8), (0, 2, 16) and (0, 4, 13), respectively. Note that (0, 1, 8), by our convention, denotes the binary sequence of length N = 19 with a 1 in positions 0, 1 and 8 only.

We consider next the set of distances between two 1's in the protocol sequence with first period (0, 1, 8) when those two 1's are less than N positions from each other. This set of distances is seen to be {1, 7, 8, 18, 12, 11} as follows from the fact that the 1 in position 0 in one period is distance 8 from the 1 in position 8 of the same period but is distance N-8 = 19-8 = 11 from the 1 in position 8 of the previous period, etc. Similarly, the sets of distances between 1's in the protocol sequences with first periods (0, 2, 16) and (0, 4, 13) are {2, 14, 16, 17, 5, 3} and {4, 9, 13, 15, 10, 6}, respectively. The point to be noticed is that these three sets of distances are pairwise disjoint — no distance appears in more than one set. This means that, regardless of the

Pet., Exh. 1017, p. 13

synchronization among the 3 protocol sequences, no two users can collide in more than one slot in any span of N consecutive slots; because if two protocol sequences both have a 1 corresponding to some slot, then they cannot both have a 1 in some other slot less than N slots distant from the former.

It follows that if any M = 2 senders are active, then each is guaranteed that at least two of the three packets that he sends within one period of his protocol sequence will be successfully received. Moreover, each sender can send two information packets error-free (assuming that the packet itself is a binary sequence of some fixed length n) in each period simply by sending the information packets in the first two slots where his protocol sequence has 1's and sending the bit-by-bit modulo-two sum of these packets as the "redundant packet" in the remaining slot. If either information packet is lost in a collision, it can be recovered at the receiver by subtracting the other information packet from the redundant packet. It follows that this T = 3 sender protocol-sequence/coding scheme allows any two senders to be active and each to send information packets error-free at a rate of 2/N = 2/19 (packets/slot). The total rate of 4/19 ≈.211 (packets/slot) is not unreasonably smaller than C_2 = 1/2, the best that can be done with T = M = 2.

6.3 Delay Considerations

It is time to recall once again that the usual purpose of random-accessing is small (average) delay rather than high throughput. Space does not permit us to say much about how the "high-throughput" schemes in sections 6.1 and 6.2 can be modified to reduce delay, but we will give the flavor by considering again the two sequences, [1, 1, 0, 0] and [1, 0, 1, 0], that were used as first periods of periodic protocol sequences in section 6.1. The trick to reducing delay is not to use these sequences periodically but rather to use them only when the corresponding sender actually has a new packet to send, filling 0's into the protocol sequence during idle periods. [In general, one also needs to add some (at most N-1) 0's to the end of each first period of length N to ensure that the scheme still functions correctly, but no such additional 0's are needed for this M = 2 scheme.] The coding scheme is unchanged: each sender still sends each information packet twice. One sees that, if the traffic is light, there will usually be zero transmission delay as the first information packet will get through correctly. When this packet experiences a collision, the transmission delay will be 1 slot if it was from sender 1 and 2 slots if it was from sender 2.

Similar delay considerations apply to the M-out-of-T channel coding schemes. One sees, moreover, from these arguments that one can allow the set of active senders to change with time, as long as no more than M are ever active and provided that there is an idle period of at least N - 1 slots between the time that one out of exactly M active senders ends his activity and the next one begins.

6. CONCLUDING REMARKS

We have described some new approaches to random-accessing, with emphasis on the collision-resolution approach for the reason that this approach appears to us to be an eminently practical one but not yet familiar to many practicioners of random-accessing. The reader may wonder why we have said nothing about "carrier-sensing", "collision detection", and many other techniques that are often used in practice to improve the efficiency of random-accessing with feedback. Our

Pet., Exh. 1017, p. 14

reason for avoiding a discussion of these "frills" is that (1) they can be used just as effectively with the collision-resolution approach as with the ALOHA approach to random-accessing, cf. [8], and (2) their introduction tends to obscure the real issue of random-accessing, viz., how effectively does one handle access conflicts. The recent book by Bertsekas and Gallager [17] is "must reading" for anyone who wants to deepen his understanding of the real issues.

We have also described two approaches to random-accessing without feedback. This area is still very much in the research stage, but our guess is that it will also find practical applications. Indeed, when lecturing recently on the material in section 6.1, we were pleasantly surprised by the enthusiastic response of a listener who had wanted to build a random-access system for the remote collection of data from a few sensors in his laboratory, but had previously thought it would be necessary to build a two-way channel so that the sensors could be provided with feedback after their access attempts. There may be other random-access applications where we are now providing feedback out of assumed necessity rather than as a practical choice. If so, the approaches described in section 6 merit some scrutiny by practicioners of random-accessing.

REFERENCES

[1] N.T. Gaarder and J.K. Wolf, "The capacity region of a multiple-access discrete memoryless channel can increase with feedback", IEEE Trans. Info.Th., vol. IT-21, pp. 100-102, Jan. 1975.

[2] F.M.J. Willems, "On multiple access channels with feedback", IEEE Trans. Info. Th., vol. IT-30, pp. 842-845, Nov. 1984.

[3] N. Abramson, "The ALOHA system - Another alternative for computer communications", Proc. Fall Joint Computer Conf., AFIPS Press, vol. 37, pp. 281-285, 1970.

[4] L.G. Roberts, "ALOHA packet system with and without slots and capture", Comp. Comm. Rev., vol. 5, pp. 28-42, April 1975.

[5] B. Hajek, "Stochastic approximation methods for decentralized control of multiaccess communications", IEEE Trans. Info. Th., vol. IT-31, pp. 176-184, March 1985.

[6] J. Capetanakis, "A protocol for resolving conflicts on ALOHA channels", Abstracts of Papers, IEEE Int. Symp. Info. Th., Ithaca, New York, pp. 122-123, Oct. 1977.

[7] B.S. Tsybakov and V.A. Mikhailov, "Slotted multiaccess packet broadcasting feedback channel", Probl. Peredachi Inform., vol. 14, pp. 32-59, Oct. - Dec. 1978.

[8] J.L. Massey, "Collision resolution algorithms and random access communications", pp. 73-137 in Multi-User Communications (Ed. G. Longo),CISM Courses and Lectures, No. 265. Wien and New York: Springer 1981.

[9] P. Mathys and Ph. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free or blocked channel access", IEEE Trans. Info. Th., vol. IT-31, pp. 217-243, March 1985.

[10] P. Studer and H. Pletscher, "Part-and-try algorithms for random-access with the Poisson arrival model", Report of semester project, Inst. Signal & Info. Proc., Swiss Federal Inst. Tech., Zurich, Feb. 1984.

[11] M. Hofri, Probabilistic Analysis of Algorithms. Heidelberg and New York: Springer, 1987.

[12] F.P. Kelly, "Stochastic models of computer communication systems", J.Royal Stat. Soc., vol. 47, 1985.

[13] B.S. Tsybakov and N.B. Likhanov, "An upper bound for capacity of a random multiple-access system", Probl. Peredachi Inform., vol. 23, pp. 64-78, Jul-Sept., 1987.

[14] J.L. Massey, "The capacity of the collision channel without feedback", Abstracts of Papers, IEEE Int. Symp. Info. Th., Les Arcs, France, p. 101, June 1982.

[15] J.L. Massey and P. Mathys, "The collision channel without feedback", IEEE Trans. Info. Th., vol. IT-31, pp. 192-204, March 1985.

[16] L.A. Bassalygo and M.S. Pinsker, "Restricted asynchronous multiple access", Probl. Peredachi Inform., vol. 19, pp. 92-96, Oct.-Dec. 1983.

[17] D. Bertsekas and R.G. Gallager, Data Networks. Englewood Cliffs, NJ: Prentice-Hall, 1987.

Pet., Exh. 1017, p. 15