On Queueing Problems in Random-Access Communications

Invited Paper

LEONARD KLEINROCK, FELLOW, IEEE

Abstract—The general problem of allocating the capacity of a communication channel to a population of geographically distributed terminals is considered. The main focus is on the queueing problems that arise in the analysis of random access resolution algorithms. The performance measures of interest are the channel efficiency and the mean response time. The nature of known solutions for various random access schemes is discussed and a lower bound for the mean response time is conjectured.

I. Introduction

WAITING IN LINE is "a pain in the neck," but too often we find we cannot resist a good queue. Queueing theory is concerned with evaluating how badly customers get treated when they compete for access to a server. In fact, Mother Nature is especially unkind in this case since, even when the server is able to keep up with the average demand, it is well-known that performance will suffer due to random arrival patterns and random service requirements. For most of the twentieth century, queueing theory has devoted itself to evaluating these waiting times, queue lengths, busy period durations, server utilization, throughput, etc. in terms of the arrival and service time distributions.

Rather than complain about queues, we should instead by thrilled whenever allowed to form them! For, indeed, the technological advances which have provided *remote access* to computing facilities have also given rise to the technological problems of *multiaccess* computer communications. In these multiaccess systems, not only are we faced with the usual queueing problems which arise from unpredictable message generation times and lengths, but we are also faced with the nasty issue of allocating a communications resource (the server) to a *geographically distributed* set of message sources (the customers). Were we not in a distributed environment, then queueing theory would provide us with the ultimate delay-throughput performance profiles. Now, however, we find that there is a performance cost to form an organized queue in this distributed environment. (Classical queueing theory has always assumed that customers could trivially form themselves into a cooperating queue as, for example, first-come-first-serve, last-come-first-serve, etc.). In our new environment, we must now account for the additional loss in throughput or increase in delay required to create a cooperative queue which, somehow, permits intelligent shared access to the available channel bandwidth. Basically, the problem is that the geographically distributed demands for access to the server (the shared communication channel) are unaware of other demands also requiring access, and so it is clear that contention will exist, due not only to the random elements but also the inability of users to observe each other as these demands arise. One should recognize the beauty of a queue —it is an example of an *ideal* resource-sharing mechanism. In a typical queue, the service capacity is not preassigned to certain customers, but is rather made available *on demand* to whichever customers happen to have already arrived. This is a perfect, dynamically allocated resource in which no service capacity is idle when any work is to be done and in which no service capacity is wasted due to collisions.

Let us examine this situation more carefully. We have a contention system in which two factors contribute to a degradation of performance: first, there are the usual queueing effects due to the random nature of the generation process; second, there is the cost due to the fact that our message sources are geographically distributed. If all our sources were co-located (i.e., if communications of control information among them were free and instantaneous), then we could form a common queue of the generated message packets and achieve the optimum delay-throughput profile that queueing theory [1] would predict (e.g., $G/G/1$ theory). In such a classical queueing situation, contention (simultaneous demands for the server) is usually handled by one or more of the following procedures.

- Queueing: i.e., one customer gets served while the others wait for service.
- Splitting: all customers get served simultaneously, each with a fraction of the service capacity.
- Blocking: one customer gets served and all the others are asked to leave.

When we place ourselves in a distributed environment, a fourth mechanism for resolving contention is possible,

Manuscript received August 30, 1984; revised January 15, 1985. This work was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-82-C-0064.

The author is with the Computer Science Department, 3732 Boelter Hall, University of California, Los Angeles, Los Angeles, CA 90024, USA.

0018-9448/85/0300-0166$01.00 ©1985 IEEE

namely

Smashing: when more than one user attempt to seize the server simultaneously, there is destructive interference, and none receive any service.

To use a familiar example, let us consider a meeting which is interrupted for a coffee break. Suddenly, all the attendees of that meeting request service from a common server, namely, the coffee urn, which dispenses its contents by means of a manually operated spigot. As this mass of attendees converges on the coffee urn, the casual observer might liken the situation to a mob; however, it is really a queue, in the sense that one individual will get served at a time, while all the others wait. (It may be that the most aggressive or the strongest or the prettiest or the richest individual gets served first but, nonetheless, it is a queue). On the other hand, if we were to blindfold the attendees, place coffee cups in their hands, point them in the direction of the coffee urn and ask that they attempt to receive coffee, then it is clear that, occasionally, more than one cup will be jammed under the open spigot at the same time, resulting in more coffee on the floor than in anyone's cup; this corresponds to smashing.

Unfortunately, as we have seen, our distributed user environment has created a situation into which we must invest some effort towards allocating the capacity of the communication channel. Thus, we are faced with controlling access to a common server in which the control information for organizing this access must pass over the same communication channel which is being controlled. We have quite a collection of choices for introducing the control information (or lack of it) which contributes to the formation of a cooperative queue. This control ranges, essentially, from no control at all to either an extremely tight, static control on the one hand or to some form of dynamic control on the other.

At one end of this spectrum, where *no control* is enforced, more than one terminal may transmit at the same time, and collisions may occur, as described above (the spilled coffee); such uncontrolled schemes are extremely simple to implement and involve little or no control function or hardware, but extract a price from the system in the form of wasted channel capacity due to collisions. Indeed, let us denote, by f_{col}, the fraction of the channel capacity which is wasted due to colliding transmissions. At a second extreme, we might introduce an extremely rigid system of *fixed controls* under which each terminal is permanently assigned a portion of the overall channel capacity for its exclusive use. Whereas such a scheme avoids collisions, it is inefficient for two reasons: first, because the terminals tend to be bursty and, therefore, much of their permanently assigned capacity may well be wasted due to their high peak to average ratio; second, as noted in [2], the response time will be far worse in this channelized case due to the "scaling effect." Such schemes lead to the creation of idle slots, namely slots which could have been used by other (busy) terminals but were wasted since they were permanently assigned to a given terminal which had no data to transmit. Let us denote the fraction of the channel capacity which is wasted in idle slots as f_i. At a third extreme, we find the class of *dynamic control* schemes in which a portion of the channel is set aside for control and this control is used to determine the identity of those users (known as "busy" users or terminals) who currently are on the queue (have packets ready to transmit). Using this information, some form of either perfect or (due to a failure to obtain complete information) imperfect dynamic allocation of the channel capacity will be made according to a terminal's demand; these schemes, however, extract a price in the form of the overhead due to the control channel. Let us denote the fraction of the channel capacity used in this control function to be f_{con}. In one form or another, nature will extract a price, either in the form of collisions due to poor or no control, or in idle (and therefore wasted) time due to rigid fixed control or overhead due to dynamic control. This TRANSACTIONS is devoted to random-access methods, which can be defined to be those for which $f_{col} > 0$. Note that this eliminates from considerations such schemes as time-division multiple-access (TDMA), frequency-division multiple-access (FDMA), token passing and many others; these are eliminated since they are collision-free algorithms, i.e., $f_{col} = 0$. (We shall say a few words about collision-free algorithms in Section VI.)

It is important to understand that much of the activity in this area of multiaccess communications has been generated by the recent technological advances in computer communications. One of the earliest applications was in the area of satellite packet switching, followed soon thereafter by ground radio packet switching [14]. More recently, the rapidly growing field of local area networks has given renewed importance to the study of multiaccess techniques, and some of these techniques have even been popularized in the public news media.

II. The Model

Our model consists, basically, of a $G/G/1$ queue with some additional features. The interarrival time distribution $A(t) = P_r[\textit{interarrival time} \leq t]$ and the service time distribution $B(x) = P[\textit{service time} \leq x]$ characterize the situation; for purposes of this paper, we will mostly consider only single-server systems. We shall consider both infinite and finite population models. We further assume that there is a common broadcast channel that is to be shared by all users and can be heard by all users whenever any transmission takes place (corresponding to what is commonly known as a "one-hop" system). To characterize the geographical distribution of the customers (or, if you will, traffic sources), we introduce the distance metric d_{ij} which represents the distance between sources i and j. We let v be the basic system velocity (such as the speed of light in free space, or the speed of information propagation along a copper bus system, etc.). As a result, we see that $\tau_{ij} = d_{ij}/v$ represents the propagation time to transmit energy between sources i and j. As usual, we denote the average service time (i.e., the average time to pump the entire packet into the channel) by $\bar{x}$. We may then define

$$a_{ij} = \tau_{ij}/\bar{x}, \tag{2.1}$$

where a_{ij} represents the ratio of the propagation delay to the average service time for two sources, and we denote by A the matrix of such values. We may also interpret a_{ij} as the number of packets which can fit into the medium over the distance spanned between users i and j. If we have a finite population of M distributed sources attempting to share the common channel, we will assume that the mth such source generates demands at a rate of $\lambda m/\mathrm{s}$. The load placed on the channel by this source is, therefore, $\rho_m = \lambda_m \bar{x}$, and the total load placed on the server is simply

$$\rho = \sum_{m=1}^{M} \rho_m. \tag{2.2}$$

(For an infinite population, we let λ denote the overall arrival rate, in which case we have $\rho = \lambda \bar{x}$.)

We have now characterized this distributed access problem in terms of the following parameters: M, $\{\rho_m\}$, ρ, and A, in addition to the interarrival time distribution $A(t)$ and the service time distribution $B(x)$. We are interested in calculating the loss and performance capabilities of a distributed system with this set of parameters. There are some special cases where, in fact, this problem loses its distributed character. One such case is when $a_{ij} = 0$ (for all i, j); however, one should note the caution described in the third paragraph of Section III. Another case is when $M = 1$. A third case is when all but one of the ρ_m goes to 0. Except for these limiting cases, we are faced with some loss of service capacity which must be devoted to organizing these geographically separated sources into a cooperative queueing structure.

Queueing theory is capable of solving for a number of system performance variables. Ordinarily, however, we usually ask simply for the average response time as a function of load. Let us define $T(\rho)$ to be the *normalized* average response time from when a packet is generated until it is successfully received. Our main concern is the way in which the normalized average response time, $T(\rho)$ varies with the overall system load ρ. $T(\rho)$ is expressed in units of packet transmission times for a data channel whose capacity is C bits per second, i.e., $T(\rho)$ is normalized with respect to b/C seconds, where b is the average number of bits in a packet. (Note that $\bar{x} = b/C$.) Further, since the analysis of most random access systems assumes $\tau_{ij} = \tau$ (for all i, j)—which, by the way, is clearly an impossibility for $M > 4$—we subtract the constant τ from all of our delay expressions. Thus

$$T(\rho) \triangleq \frac{T_u(\rho) - \tau}{b/C}, \tag{2.3}$$

where $T_u(\rho)$ is the unnormalized average response time. Note that $\rho = \lambda b/C$.

III. A Conjectured Lower Bound on the Mean Response Time

In this distributed environment, the efficiency of the channel usage is a function of the total load in the system; that is, the wasted capacity in the form of idle slots, collisions, and overhead depends very much upon the number of terminals that have packets ready to send. In the case when we have a finite number of terminals and when we allow queues of packets to form at each terminal, the unfortunate part as far as queueing analysis is concerned is that these queues are *coupled*, i.e., the behavior of one queue depends upon the state of the other queues in the system, and this renders the analysis problem quite difficult. In fact, a rather extensive analysis is required in order to study a particular case of *only two* interfering queues [3], and no exact analysis is yet available for the problem of $M > 2$ interfering queues. One possible approach to allocating the channel optimally in this distributed environment is to use a decentralized optimization procedure as discussed in [4], that is, one in which the individual users attempt to optimize their performance in a way where further improvement in their own behavior is impossible without negatively affecting the other users; this is known as a pareto-optimal solution [5]. Sometimes this approach leads to a globally optimal solution [6]. Since the exact analysis is so difficult in general, it is useful to seek bounds. In this section, we conjecture a lower bound on the mean response time.

In 1979, we conjectured that there is a useful way to obtain a lower bound to the optimal behavior, i.e., a lower bound to the mean response time [7]. We examine that approach here. First we observe that, if we have a Poisson arrival process and if all packets have the same number of bits, then the problem is reduced to an $M/D/1$ problem in a distributed environment. Clearly then, the response time of our broadcast channel can be no better than that of a simple $M/D/1$ queue, since we are not charging ourselves for the "cost of distribution." One wonders how closely one can approach this overidealized behavior. Our conjectured lower bound below would be an improvement over the $M/D/1$ bound.

First, we must point out that any scheme which takes advantage of the ability of a terminal to sense the state of the channel requires a further refinement in our model of its performance. In particular, it is clear that the detection of silence, upon which such schemes often depend, is really the detection of another symbol in the alphabet of channel symbols. As a result, one must not allow the normalized parameter a_{ij} to shrink below that of the time required to transmit a symbol, and so one must not accept the published performance evaluation equations of "carrier-sensing" schemes when a_{ij} approaches 0; rather, one must then introduce an additional time for detecting the silence symbol.

In order to obtain a lower bound on performance, we observe that it is sufficient for a terminal to be aware of the exact number of busy terminals (say, N) at the time when that terminal itself becomes busy. In such a case, the terminal will know its exact position on queue, namely that it is in position $N+1$ (assuming first-come first-served). Let us first imagine that an all-knowing gremlin is available to provide this information to a terminal as soon as it becomes busy. We define P to be the row vector describing the equilibrium probability for the number of busy terminals in equilibrium, that is,

$$P = [P_0, P_1, P_2 \cdots] \tag{3.1}$$

where P_m is the equilibrium probability

$$P_m = P[m \text{ terminals are busy}]. \tag{3.2}$$

It is clear that, on the average, the minimum amount of information which must be transmitted to a terminal by the gremlin is simply the entropy (say, in bits) of the distribution given above. This entropy, $H(P)$, is

$$H(P) = -\sum_{m=0}^{\infty} P_m \log_2 P_m. \tag{3.3}$$

We are here assuming a straightforward $M/D/1$ model with an infinite population of terminals. (Were we considering instead a finite population of M terminals, then the appropriate distribution to use in this entropy calculation would be that for the other $M-1$ terminals.) For the case $M/D/1$, it is well-known [8] that the generating function for the equilibrium probabilities is given by

$$P(z) = \sum_{m=0}^{\infty} P_m z^m = \frac{(1-\rho)(1-z)}{1 - ze^{\rho(1-z)}} \tag{3.4}$$

and the expression for P_m is given by

$$P_0 = (1-\rho) \tag{3.5a}$$

$$P_1 = (1-\rho)(e^{\rho} - 1) \tag{3.5b}$$

$$P_m = (1-\rho)\left[\sum_{k=1}^{m} e^{k\rho}(-1)^{m-k}\frac{(k\rho)^{m-k}}{(m-k)!} + \sum_{k=1}^{m-1} e^{k\rho}(-1)^{m-k}\frac{(k\rho)^{m-k-1}}{(m-k-1)!}\right], \quad m \geq 2. \tag{3.5c}$$

Since the arrival rate (in packets/s) of terminals to the busy population is simply λ, then the rate at which the gremlin must provide information to the (infinite) population of terminals is simply $\lambda H(P)$, assuming that the gremlin takes care to code the information he must transmit in the most efficient form according to Shannon's noiseless coding theorem [9]. This operation on the part of the gremlin involves three activities. First, the gremlin must observe the state of the system; we assume this is done at no cost to the system. Second, in encoding the information to be transmitted, some delay will be incurred due to the coding procedure; this, too, we assume, costs the system nothing in terms of delay. Third, the gremlin must use some of the system channel capacity in transmitting this information, and it is this price which we include in order to calculate the lower bound on performance.

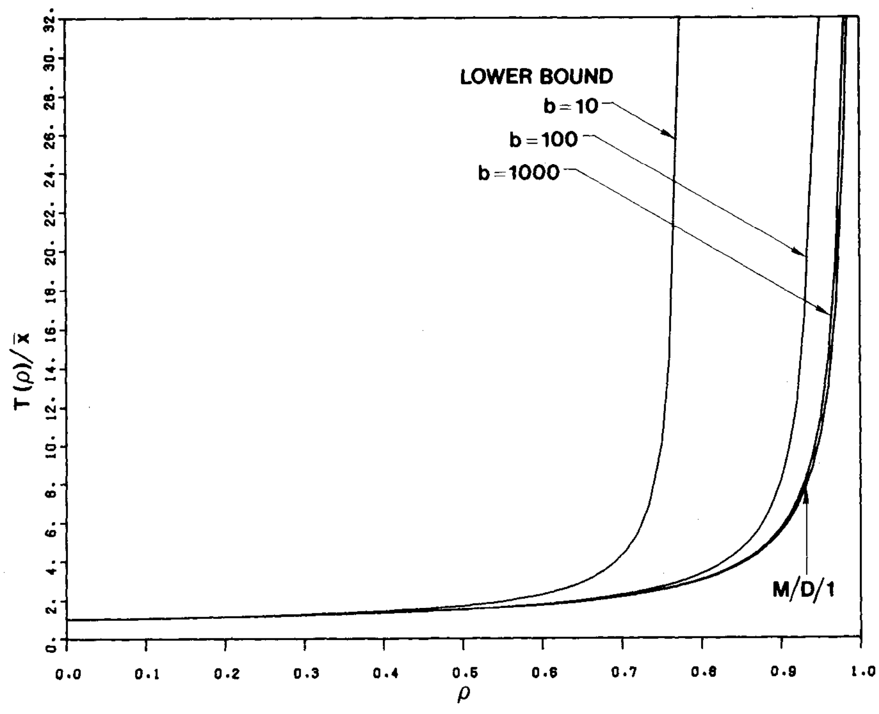

Fig. 1.

Now we know that the $M/D/1$ system incurs a normalized delay $T(\rho)$ when the (total) system load is at the value ρ. However, of this load, we now assume that some of the capacity is used for the control information transmitted by the gremlin; as a result, only a portion of the load is useful data, and this portion we define as ρ', where

$$\rho' = \frac{b}{b + H(P)}\rho. \tag{3.6}$$

This last is true since each newly activated terminal will transmit b useful bits, and the gremlin will be required to transmit $H(P)$ control bits per busy terminal on the average. Note that $H(P)$ is a function of ρ. If we now charge our system for this reduction in useful throughput, we find that the lower bound for the delay-throughput profile of any access scheme is simply given as

$$T_{LB}(\rho') = T_{M/D/1}(\rho) \tag{3.7a}$$

where

$$T_{LB}(\rho) = \frac{T_u(\rho) - \tau}{b/C}. \tag{3.7b}$$

Thus, the behavior of the delay-throughput profile for *any* access scheme is lower bounded by

$$T(\rho') \geq T_{M/D/1}(\rho) = \frac{1 - \frac{\rho}{2}}{1 - \rho}. \tag{3.8}$$

We note, for $\rho \to 0$, that the lower bound approaches the $M/D/1$ curve; this is true since in this limit the entropy of the distribution approaches zero, and no capacity is lost in the transmissions due to the gremlin. Furthermore, for a finite population of terminals, it is clear that, as $\rho \to 1$, the entropy will once again approach zero, and the lower bound will approach the pure queueing curve.

The behavior of three infinite population cases is given in Fig. 1 (for $b = 10$, 100, and 1000), and they are compared to the classical $M/D/1$ curve. Note the minimal loss when $b = 100$ and the significant loss when $b = 10$.

IV. Infinite Population Random Access Models

Here we consider an infinite population of terminals, each one of which generates traffic at an infinitesimal rate, the aggregate rate being λ packets/s. In this case, no queue will form at any individual terminal but, nevertheless, the performance of a given terminal will certainly depend upon the total system state.

In this section (and the next), we shall describe various random access algorithms and indicate the nature of the solution which currently exists for the mean response time; in those cases where interesting queueing problems have been generated, we shall make the appropriate comments. We have no intention of giving a total coverage of *all* known schemes—far from it. It is our intention, rather, to focus on some of the common properties and techniques which have emerged from these studies.

A. Pure (Unslotted) ALOHA

In this algorithm, at the instant of its generation, a packet will be transmitted by its terminal. If, at any instant, more than one packet is in the process of being transmitted, then all the packets involved in that overlap will be destroyed, and each will have to be retransmitted. Abramson [10] was the first to analyze the behavior of this system. He made the "bold Poisson assumption" in which he assumed that any packets which must be retransmitted (due to an earlier collision) increased the total incoming traffic in such a fashion that the total transmission process was still a Poisson arrival process. This allowed him to solve for the throughput S (expected number of successful packets per packet transmission time) as a function of G, the total applied traffic (same units as S). He found the now classic result

$$S = Ge^{-2G}. \tag{4.1}$$

Note that $\max S(G) = 1/2e$.

One way to achieve a system in which the bold Poisson assumption is quite accurate is to assume that any collided packet gets retransmitted at a randomly selected time with an infinite mean delay until retransmission; this, of course, completely corrupts any attempt to find the mean response time of a packet in a real system. However, it is easy to show that the ratio G/S is exactly equal to the total expected number of transmissions that a packet must be subjected to until it is successfully received. Thus, the ratio G/S is a *measure* of the mean response time which one would incur in a pure ALOHA system.

Ferguson [11] studied the effect on performance when packet lengths are selected from a random distribution. His main result was that, if the mean of the packet length is the same as the constant packet length considered above, then the optimum performance is obtained for constant packet lengths. He also found bounds and approximations for the distribution of delay [12], [13].

ALOHA channels are fundamentally unstable [14], but there exist a number of simple control procedures which stabilize these channels. Most of these control schemes estimate the number of busy terminals in the system (e.g., see [15]). It is shown in [16] that a control scheme based on perfect knowledge of the number of busy terminals leads to a mean response time which is directly proportional to this number. The same result has also been shown for tree access algorithms (see Section IV-E) [17].

B. Slotted ALOHA

This scheme operates exactly the same as pure ALOHA except that a new packet transmission must begin at the next slot boundary, as time is divided into slots with lengths equal to the packet transmission time; if a packet is generated in the middle of a slot, then it must wait until the next slot boundary before it is transmitted. Collided packets, once again, must be transmitted at some random future time. The S, G relationship is simply

$$S = Ge^{-G}. \tag{4.2}$$

Note that $\max S(G) = 1/e$, yielding twice the capacity (0.368) as that of pure ALOHA (0.184). The analysis of this system in given in [18] and turns out to provide a solution to the mean response time which is not given in "closed form" but is, rather, given as the lower envelope of an infinite set of curves.

C. CSMA (Carrier-Sense Multiple Access)

This scheme operates the same as does pure ALOHA except that a terminal first senses (listens to) the channel and can hear the carrier of any other terminal's transmission. If such a carrier is detected, then the terminal refrains from transmitting and follows one of many well known protocols for deferred transmission. The key analyses here may be found in the [19]. The S, G relationship can be obtained for most of these protocols and demonstrates that CSMA, typically, has a far greater capacity than either of the ALOHA schemes. However, the capacity of carrier sense depends very strongly upon the propagation delay parameter $a = \tau/(b/C)$ (assuming a constant propagation time between all pairs of terminals); indeed, the system capacity degrades badly as a increases. However, synchronization effects can come into play to bring back improved performance in certain cases (e.g., see [20]). Since we have analytic expressions for the S, G relationship, we can easily plot $G/S = E$ [number of total transmissions until success]. Mean response time as a function of load is once again more difficult to obtain. However, CSMA systems have been simulated to expose their mean response time [14]; attempts have been described for fitting simple analytic expressions to these curves (e.g., the ZAP approximation in [21]). Focusing on stability issues, one can determine when a CSMA system is stable. Two different approaches may be found in [22] and [23].

D. CSMA / CD (Carrier-Sense Multiple Access with Collision Detection)

This system behaves exactly as CSMA except that, if a collision does occur, then after a small detection time, all terminals involved in that collision will have detected that they are in a collision and will then cease transmission. (In actuality, at this point, a terminal in the ETHERNET system, which uses CSMA/CD, instead of ceasing its transmission, continues to blast a transmission over the common communication channel for a short period of time in order to guarantee that all terminals are aware that a collision has occurred). In this fashion, collisions will not continue for the entire packet duration, and the efficiency of CSMA/CD exceeds that of pure CSMA. Two delay analyses have been published for slightly different versions of this algorithm, and they may be found in [24], [25]. In [23] the issue of stability is addressed.

E. Tree Access Algorithms

There are a variety of decision tree algorithms in which a sequential decision process is followed in order to isolate a single busy terminal, which is then allowed to transmit its packet. A large number of the other papers in this issue are devoted to the analysis and evaluation of these "polling" algorithms, and so we choose not to pursue them much further in this paper. Nevertheless, it is important to understand why those algorithms are interesting. The basic approach in tree access algorithms is to allow a certain set of terminals to transmit. If none of these terminals have any data to send, then the channel will remain silent, and then it is known that these terminals are idle. If exactly one of these terminals transmits a packet, that packet transmission will be successful, and all terminals will become aware of that fact. If, however, more than one terminal transmits, a collision will occur, and a sequential resolution procedure will then be initiated. Various forms of this resolution procedure exist, but the basic idea is to split the population into a subset of the original colliding population and then repeat the resolution procedure, hoping for a successful transmission. The sequential decision procedure continues to operate until it is known that all members of the originally selected group of terminals have transmitted any data they had to send, and when that is known, then the process begins once again with another set of terminals. An interesting application of this idea has been applied to ETHERNET whereby the basic CSMA/CD algorithm is modified to limit the number of terminals involved in resolving a collision; this policy increases the success probability of a transmission and reduces the time until success [26].

As one can see from the table of contents of this TRANSACTIONS, the evaluation of tree algorithms has captured the fancy of many in the information theory community, and they have been intensely pursuing an exact calculation of the capacity of various versions of this algorithm.

F. Virtual Time CSMA

Before describing the operation of this scheme, an important observation must be made. In many of the random access schemes mentioned above, the channel access procedure is recursively confronted with the task of trying to isolate a single busy terminal for transmission. The assumption, ordinarily, is that the terminals are indistinguishable from each other at the time the resolution algorithm begins. However, this is not the case; there is a clear, distinguishing feature among these terminals, and that is simply that each one of them generated a packet (if any) for transmission at a *unique* instant on the time axis. If one could identify those packet generation times, then one would have the means of distinguishing these terminals perfectly, thereby giving them permission to transmit, one at a time, in a way which would avoid all collisions. Virtual time CSMA (VTCSMA) exploits this observation [27]. The way in which it operates is to provide each terminal with two clocks, one a real-time clock and the other a virtual-time clock. The virtual-time clock begins ticking at the same time the real-time clock does, and whenever the virtual-time clock's reading is the same as real time, then the virtual-time clock ticks at the same rate as real time, i.e., it is never allowed to race ahead of real time. However, whenever a transmission is detected on the channel, then the virtual-time clock stops ticking until the channel goes silent again, at which time the virtual-time clock begins ticking, but this time at a rate, say v, which is faster than real time. Each terminal that has generated a packet awaiting transmission constantly compares the virtual-time clock with the real time at which its packet was generated. When these two values are equal, then that terminal will initiate transmission of its packet and will continue transmission until the packet is fully transmitted. Note that, in this scheme, two packets will collide only if their packet generation times were the same within the propagation delay time of the system. A transform approach to the queueing analysis for the response time has been carried out for two models of this system. The mean response time performance of this system is excellent.

G. Multi-Channel ALOHA and CSMA

It is possible to divide the data channel into a number of data subchannels, each with a fraction of the total channel capacity. One way in which a user may then operate in this environment is to transmit his packet over more than one of the subchannels at the same time. Marsan [28] has reported upon the analysis of a few variations of this scheme and has been able to establish real improvements in response time due to a variety of effects (including a reduction in the parameter a).

V. Finite Population Random-Access Models

A. ALOHA (Pure and Slotted)

Abramson [10] studied the throughput in pure and slotted ALOHA systems. Defining S_m to be the probability that the mth terminal ($m = 1, 2, \cdots, M$) generates a packet during a given slot and denoting by G_m the probability that the mth terminal transmits a packet in a given slot, he was able to show that the S, G relationship was simply

$$S_m = G_m \prod_{i \neq m} (1 - G_i). \tag{5.1}$$

He further established the very important result that, in order to give the pareto-optimal throughput of this system, the collection of terminals should adjust their transmission probabilities such that

$$\sum_{m=1}^{M} G_m = 1. \tag{5.2}$$

Using a Markov chain approach for a slotted ALOHA channel with M unbuffered terminals, Tobagi has found the distribution of response time and interdeparture time [29]. For the case of an exhaustive slotted ALOHA system with M fully buffered terminals, Levy has given an approximation to the mean response time using results from a random polling queueing system [20].

B. CSMA and CSMA / CD

Using a semi-Markov approach to CSMA, Tobagi has found the z transform of the distribution of response time and interdeparture time with recursive procedures to compute the moments of these variables [30]. A simpler approximate solution for the mean response time (and, with somewhat more computation, the distribution) may be found in [31].

A direct Markov chain approach for CSMA/CD lends itself to a recursive solution for throughput and the mean response time [32]; this approach also emphasizes the use of the "instantaneous expected drift" to discuss stability. An interesting approach to the estimation of the mean response time in CSMA/CD is given in [33].

C. The URN Scheme

This was, perhaps, the first scheme that recognized that asymmetry in the resolution procedure was a clear benefit in isolating busy terminals. The way in which it operates is as follows: We assume that every terminal is aware of the exact number of terminals which have packets awaiting transmission; however, the terminals are unaware of the exact identity of these busy terminals. Let us assume (the same assumption made in the last paragraph of Section IV-A) that it is known that N terminals out of the M total terminals are busy. Then, in the URN scheme [34], a randomly selected group of k terminals is selected from an urn containing the total of M, and these k are given permission to transmit. If none of them have any data to send, then the channel will remain idle. If exactly one of them has a packet to transmit, then that packet will be successfully transmitted. If more than one have packets to transmit then we will once again have a collision. The idea is to optimally select k based on a knowledge of N and M in order to maximize the success probability. It turns out that the optimum value for k is simply the ratio $[M/N]$ where the bracket notation here refers to the greatest integer not exceeding the ratio M/N (i.e., the floor function). When this optimum value of k is used, it is easy to show that the expected number of busy terminals selected will be exactly one, and this will maximize the probability of successful transmission. This scheme was one of the earliest of the dynamically varying schemes. Its behavior is very much like the ALOHA system under very light load (where ALOHA is optimal) and very much like TDMA under heavy load (again, approaching an optimal access scheme at this load).

A dynamic control for the input to the URN scheme is analyzed in [35] using a Markov decision model whose numerical evaluation shows an improvement in performance to the uncontrolled URN.

D. CSMA / CD / DP (CSMA / CD with Dynamic Priorities)

A hybrid system using CSMA/CD until a collision is detected and, following that, using a deterministic resolution algorithm to resolve the conflict among those terminals which contributed to the collision is studied in [36]. The way in which this resolution can occur is due to a deterministic numbering scheme among the terminals which may be changed dynamically according to a desired priority among the various terminals. Here an approximate performance evaluation for the mean response time is compared to simulation. The key characteristic of this system is that, like the URN system, it performs optimally under light and under heavy loads, thereby giving it a dynamic quality.

VI. Collision-Free Access Schemes

As mentioned above, collision-free access schemes do not fall in the class of random-access schemes. Nevertheless, we would be remiss if we neglected to mention some of the very well-known algorithms which fall into this class, since there are situations (e.g., in heavy traffic) in which collision avoidance is preferable to collision resolution. Frequency division multiple access (FDMA) [14], [37] and time division multiple access (TDMA) [14], [37] are examples of schemes where no collisions occur and no control overhead is incurred, but idle slots appear due to the fixed assignment of capacity. Both of these schemes tend to perform badly at light loads but perform fairly efficiently at heavy loads; furthermore, TDMA is uniformly superior to FDMA over the entire range of loads.

The second most famous access algorithm for local area networks (the first being ETHERNET using CSMA/CD) is that which IBM has been proposing for its local area network, namely, the token ring.[1] In the past, this has been called minislotted alternating priority (MSAP) as described in [38] and also as hub go-ahead polling. In fact, it is one of many possible polling schemes (see, for example, [39]). Another collision-free scheme is known as the broadcast recognizing access method (BRAM) [40]. In all of these schemes, a token is passed around a closed ring (and in some cases sequenced along a bus in some order[2]); a terminal is not permitted to transmit until the token reaches it on the medium, at which time it removes the token, inserts its data and then, under some protocol, replaces the token on the medium. Bux [41] has done an excellent job of comparing the IBM token ring and the ETHERNET access schemes and shows that, in many cases, the performance of the token ring is superior to that of ETHERNET.

Another rather interesting conflict-free resolution algorithm has recently been proposed for use with AT & T's DATAKIT product. In this scheme, each terminal has a unique ID. Expressing this ID in binary notation (of fixed length), the access method uses an initial contention resolution period whereby terminals compete with each other, following which the winner of that resolution transmits his data packet. Specifically, all busy terminals will engage in this contention resolution period by first transmitting a one on the channel if the most significant bit in the binary representation of their address is a one but will refrain from transmitting if their leading binary digit is a zero. All terminals listen to the medium, as well. If a one is heard at the time of this transmission, then only those terminals which transmitted a one may continue to compete for channel access. On the other hand, if no one is heard (e.g., all competing terminals had a leading zero), then all those who were in the competition continue. In either case, those allowed to continue will transmit their next most significant bit (one or zero). Only those who transmitted a one may continue if anyone transmitted a one, etc. By the time $\log M$ bits are transmitted (M = finite number of terminals involved in this access scheme), then exactly one winner will survive, and it is that winner who will successfully transmit his data packet. This is a rather interesting scheme and has been analyzed in [42].

VII. Multihop Access Environments

If a given terminal cannot hear the transmission of all other terminals, then we have what is known as a *multihop* environment. A number of interesting features of this environment can be studied, but for our purposes the fact that carrier sense may fail to resolve conflicts is the most important observation. This is known as the "hidden terminal" problem [43]. A number of studies have been launched in this area, but perhaps the most interesting is that by Boorstyn and Kershenbaum [44], who have developed an analytic procedure which allows one to evaluate throughput (and even response time) for a large class of multihop packet radio networks, using a rather interesting aggregation technique in which a Markov chain is defined whose states correspond to sets of terminals which, if they simultaneously transmit, will not interfere with each other (due to the fact that certain terminals do not hear and, therefore, do not interfere with each other). Recently, Tobagi and Brazio [45] have extended the Boorstyn–Kershenbaum results to identify exactly when a multihop broadcast packet network will enjoy the very important product form solution; in a word, they find that the product form solution will hold if and only if there is a reciprocity in the hearing matrix (i.e., if terminal i can be heard by terminal j, then j can also be heard by terminal i).

[1] This is not IBM's only local area network, as witnessed in its announcement on August 14, 1984 of a broadband CSMA/CD net called PC Network for their Personal Computer (PC) line.

[2] Indeed, it is clear that an effective way to "poll" a bus topology is to traverse the bus by starting at one end and traveling down the bus, polling all terminals along the way, reaching the end and then traveling back in the opposite direction. For example, if the terminals are numbered from left to right on the bus, i.e., $1, 2, \cdots, M$, then the polling order should be $1, 2, \cdots, M, M, M-1, \cdots, 2, 1$, and this pattern should be repeated cyclically. This procedure minimizes the effect of the propagation delay τ_{ij}. For a tree-structured bus, the same idea may be used, but at each 'T' intersection, one should always turn right (say) and traverse up and down each branch, making right turns at each T and turning around at each end. In this way, each terminal is polled twice during the cycle.

Other interesting work in this multihop environment is that of Silvester [46], Nelson [47], and Takagi [48], in which the issue of what transmission radius (i.e., power) should be used in order to maximize the throughput and to minimize delay in this environment is studied.

VIII. Conclusion

We have discussed the problems inherent in random-access communication channels. A conjectured lower bound on the mean response time was given. Many of the common random access schemes were described, and the status of their analytic solution was discussed. The field is very dynamic and new access schemes are constantly being proposed. The analytic techniques are fairly sophisticated but, unfortunately, still lack the power needed to solve many problems. In the midst of this activity, pressure from the local area network user community is growing daily for standards, solutions and end-user products.

References

[1] L. Kleinrock, *Queueing Systems, Volume I: Theory*. New York: Wiley, 1975.

[2] ——, "Resource allocation in computer systems and computer communication networks," in *Information Processing '74, Proc. IFIP Cong.*, Stockholm, Sweden, Aug. 1974. Amsterdam: North Holland, 1974, pp. 11–18.

[3] M. Sidi and A. Segall, "Two interfering queues in packet-radio networks," *IEEE Trans. Commun.*, vol. COM-31, no. 1, pp. 123–129, Jan. 1983.

[4] Y. Yemini, "Research directions in distributed computing and communications," in *Proc. IEEE Joint Conf. Information Theory*, Jerusalem, Israel, May 1984.

[5] M. Shubik, *Game Theory in the Social Sciences*. Cambridge, MA: Massachusetts Institute of Technology, 1983.

[6] J. Kurose, "Real-time communications in computer networks," Ph.D. dissertation, Computer Science Dept., Columbia Univ., New York, NY, 1984.

[7] L. Kleinrock, "On a new class of queueing models for distributed environments," in *Proc. Ninth International Teletraffic Congress*, Torremolinos, Spain, Oct. 1979.

[8] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*. New York: Wiley, 1974.

[9] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois, 1949.

[10] N. Abramson, "The ALOHA system—Another alternative for computer communications," in *Proc. 1970 Fall Joint Comput. Conf. AFIPS Conf.*, Vol. 37. Montvale, NJ: AFIPS, 1970, pp. 281–285.

[11] M. Ferguson, "The study of unslotted ALOHA with arbitrary message lengths," Tech. Rep. B75-13, Univ. of Hawaii, Honolulu, Feb. 1975.

[12] ——, "A bound and approximation of delay distribution for fixed-length packets in an unslotted ALOHA channel and a comparison with time division multiplexing (TDM)," *IEEE Trans. Commun.*, vol. COM-25, no. 1, pp. 136–139, Jan. 1977.

[13] ——, "An approximate analysis of delay for fixed and variable length packets in an unslotted ALOHA channel," *IEEE Trans. Commun.*, vol. COM-25, no. 7, pp. 644–654, July 1977.

[14] L. Kleinrock, *Queueing Systems, Vol. II: Computer Applications*. New York: Wiley, 1976.

[15] S. Lam and L. Kleinrock, "Packet switching in a multiaccess broadcast channel: Dynamic control procedures," *IEEE Trans. Commun.*, vol. COM-23, pp. 891–904, Sept. 1975.

[16] M. Ferguson, "On the control, stability and waiting time in a slotted ALOHA random access system," Tech. Rep. B74-9, Univ. of Hawaii, Honolulu, Dec. 1974.

[17] G. Fayolle, P. Flajolet, and M. Hofri, "On a functional equation arising in the analysis of a protocol for a multi-access broadcast channel," No. 131 Rapports de Recherche, Institut National de Recherche en Informatique et en Automatique, Rocquencourt, France, Apr. 1982.

[18] L. Kleinrock and S. Lam, "Packet-switching in a slotted satellite channel," in *Proc. 1973 National Computer Conf.*, vol. 42. Montvale, NJ: AFIPS, 1973, pp. 703–710.

[19] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, Dec. 1975.

[20] H. Levy, "Non-uniform structures and synchronization patterns in shared channel communication networks," Ph.D. dissertation, Computer Science Dept., Univ. of California, Los Angeles, 1984.

[21] L. Kleinrock, "Performance of distributed multi-access computer-communication systems," in *Information Processing 77*, B. Gilchrist, Ed. Amsterdam: North-Holland, 1977.

[22] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part IV—Stability considerations and dynamic control in carrier sense multiple access," *IEEE Trans. Commun.*, vol. COM-25, pp. 1103–1120, Oct. 1977.

[23] J. Meditch and C. T. A. Lea, "Stability and optimization of the CSMA and CSMA/CD channels," *IEEE Trans. Commun.*, vol. COM-31, no. 6, pp. 763–774, June 1983.

[24] F. Tobagi and V. B. Hunt, "Performance analysis of carrier sense multiple access with collision detection," Tech. Rep. 173, Computer Systems Lab., Stanford Univ., Stanford, CA, June 1979.

[25] S. Lam, "A carrier sense multiple access protocol for local networks," Tech. Rep. 113, Dept. of Computer Sciences, Univ. of Texas, Austin, Oct. 1979.

[26] E. Arthurs and B. W. Stuck, "A modified access policy for ETHERNET version 1.0 data link layer," *IEEE Trans. Commun.*, vol. COM-32, no. 8, pp. 977–979, Aug. 1984.

[27] M. Molle and L. Kleinrock, "Virtual time CSMA: Why two clocks are better than one," *IEEE Trans. Commun.*, Feb. 1985, to be published.

[28] M. A. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE J. Select. Areas Commun.*, vol. SAC-1, no. 5, pp. 885–897, Nov. 1983.

[29] F. Tobagi, "Distributions of packet delay and interdeparture time in slotted ALOHA channels," *JACM*, vol. 29, no. 3, Oct. 1982.

[30] ——, "Distribution of packet delay and interdeparture time in carrier sense multiple access," *JACM*, vol. 29, no. 3, Oct. 1982.

[31] J. Silvester and I. Lee, "Performance modeling of buffered CSMA—An iterative approach," in *Proc. GLOBECOM '82, IEEE Global Telecommunications Conf.*, Miami, Nov. 1982, pp. F1.6.1–F1.6.5.

[32] E. J. Coyle and B. Liu, "Finite population CSMA/CD networks," *IEEE Trans. Commun.*, vol. COM-31, pp. 1247–1251, Nov. 1983.

[33] E. Arthurs and B. W. Stuck, "A theoretical performance analysis of polling and carrier sense collision detection communication systems," in *Proc. Int. Conf. Performance Data Communication Systems and Applications*, Paris, Sept. 1981.

[34] Y. Yemini and L. Kleinrock, "Broadcasting in an URN: An optimal adaptive multiaccess scheme," Working paper ATS 80011, Computer Science Dept., Univ. of California, Los Angeles, 1980; submitted to *IEEE Comput. Commun.*, 1980.

[35] K. K. Mittal and A. Venetsanopoulos, "On the dynamic control of the URN scheme for multiple access broadcast communication systems," *IEEE Trans. Commun.*, vol. COM-29, no. 7, pp. 962–970, July 1981.

[36] W. Kiesel and P. Kuehn, "A new CSMA-CD protocol for local area networks with dynamic priorities and low collision probability," *IEEE J. Select. Areas Commun.*, vol. SAC-1, no. 5, pp. 869–876, Nov. 1983.

[37] I. Rubin, "Message delays in FDMA and TDMA communication channels," *IEEE Trans. Commun.*, vol. COM-27, pp. 769–777, May 1979.

[38] L. Kleinrock and M. Scholl, "Packet switching in radio channels: New conflict-free multiple access schemes," *IEEE Trans. Commun.*, vol. COM-28, pp. 1015–1029, July 1980.

[39] H. Takagi and L. Kleinrock, "Analysis of polling systems," UCLA Computer Science Dept. Rep., Univ. of California, Los Angeles, 1983.

[40] I. Chlamtac, W. Franta, and K. D. Levin, "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.*, vol. COM-27, pp. 1183–1190, Aug. 1979.

[41] W. Bux, "Analysis of a local area bus system with controlled access," in *Proc. Int. Conf. Performance Data Communication Sys-*

tems and Applications, Paris, Sept. 1981, pp. 11–12.
[42] A. Fraser, "Towards a universal data transport system," *IEEE J. Select. Areas Commun.*, vol. SAC-1, no. 5, pp. 803–816, Nov. 1983.
[43] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II—The hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. COM-23, pp. 1417–1433, Dec. 1975.
[44] R. R. Boorstyn and A. Kershenbaum, "Throughput analysis of multihop packet radio," in *Proc. ICC '80*, Seattle, WA, 1980, pp. 13.6.1–13.6.6.
[45] F. Tobagi and J. Brazio, "Throughput analysis of multihop packet radio networks under various channel access schemes," in *Proc. IEEE INFOCOM '83*, San Diego, 1983, pp. 381–389.
[46] J. Silvester and L. Kleinrock, "Optimum transmission radii for packet radio networks, or Why six is a magic number," in *Proc. NTC*, Birmingham, AL, Dec. 1978.
[47] R. Nelson, "Channel access protocols for multi-hop broadcast packet radio networks," Ph.D. dissertation, Computer Science Dept., Univ. of California, Los Angeles, 1982.
[48] H. Takagi, "Analysis of throughput and delay for single- and multi-hop packet radio networks," Ph.D. dissertation, Computer Science Dept., Univ. of California, Los Angeles, 1983.