On the Delay in a Multiple-Access System with Large Propagation Delay

Bruce Hajek, *Fellow, IEEE,* N. B. Likhanov, and B. S. Tsybakov

***Abstract*—The effect that large propagation delay has on the problem of network access is explored for the infinite population model with success–idle–collision feedback information, where the feedback information suffers a large propagation delay N. A simple lower bound is given on the probability that a packet is not successfully transmitted within $N/2$ time units (not including the forward propagation delay), where N is the station-to-station propagation delay. The bound implies a lower bound on the mean access delay. We also display an algorithm for which the transmission delay is within a factor of three of the lower bound, for moderate traffic loads and asymptotically large propagation delay.**

***Index Terms*—Multiple access, large propagation delay, random access, collision resolution.**

I. Introduction

RECENT advances in optical technology have made it possible to transmit at very high data rates. Consequently, the propagation delay for a packet of information is long compared to the length of a packet. For example, consider a wide-area network with a single star topology, such that the stations are located 50 km from the hub, packets are 1000 bits long, and the transmission rate is 1 Gb/s. The propagation delay from one station to another is dictated by the speed of light in glass. The delay is about 500 μs, roughly 500 times as long as the transmission time of one packet. In contrast, classic protocols such as the ALOHA protocol were investigated with a propagation delay roughly 12 times the packet length.

Architectures for switches and networks which are designed with large propagation delays in mind are currently of much interest. For example, an optical star network might be based on wavelength-division multiplexing, in which case several stations may compete to communicate with several other stations. In wavelength-division packet switching with a large number of stations, contention still occurs. Two transmitters typically should not send to the same receiver at the same time, and/or a given transmitter should not be required to send different packets to two different receivers at the same time. One might use time-division multiplexing to avoid collisions, but the delay can be large for a large number of stations. Use of a control channel to dynamically schedule transmissions does not eliminate the effect of propagation delay, which is also suffered by the control channel. Random access with large propagation delay is thus at the heart of many problems now facing communications engineers.

The particular model discussed in this paper is now described. New packets are generated according to a Poisson process on $[0,\infty)$ with rate λ. Time is divided into slots of unit length, where time is normalized so that one packet can be transmitted in one slot. We denote by slot i the time interval $[i, i+1)$. Those packets with generation times in the set B_i are transmitted during slot i, where $(B_1, B_2, \cdots)$ is specified by a random access algorithm. We require that $B_i \subset [0, i)$ for a packet cannot be transmitted until the first full slot after it is generated. The outcome of slot i, denoted by $\theta_i = \theta(B_i)$, satisfies $\theta_i \in \{0,1,2\}$. If no packets are transmitted in slot i, then $\theta_i = 0$. If one packet is transmitted in slot i, then the packet transmission is successful and $\theta_i = 1$. If two or more packets are transmitted in slot i, then the packets collide and the transmission is not successful.

There are two, often the same, propagation delays associated with the model—the propagation delay of feedback and the propagation delay in the forward channel. The propagation delay of feedback is denoted by the positive integer N. The outcome θ_i is assumed to be announced to all stations by time $i+N$. Thus, we require B_{i+N} to be a function of $(\theta_1, \theta_2, \cdots, \theta_i)$. The usual model, in which the outcome of slot i is known by the beginning of slot $i+1$, corresponds to $N = 1$. We define the *access delay* of a packet to be the number of whole slots that elapse between the time the packet is generated until the beginning of the slot in which the packet is first successfully transmitted. With this definition, the access delay is a nonnegative integer value, and it does not include the transmission time or the forward propagation delay.

We allow that a packet can be transmitted more than once within N slots. It is therefore possible for a packet to be successfully transmitted more than once, but we assume that the receiver can discard extra copies.

A. Lower Bound

Suppose that T is a fixed positive number. The mean access delay of a typical packet generated in the interval

Manuscript received January 18, 1993; revised August 11, 1993. This paper was presented in part at the IEEE International Symposium on Information Theory, San Antonio, TX, January 1993.
B. Hajek is with the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801.
N. B. Likhanov and B. S. Tsybakov are with the Institute for Problems in Information Transmission, Moscow, Russia.
IEEE Log Number 9403845.

$[0, T]$ is D_T, defined by

$$D_T = \frac{E[\text{sum of access delays of packets generated in } [0,T]]}{\lambda T}. \tag{1.1}$$

Similarly, the complementary distribution function of the access delay of a typical packet generated in the interval $[0, T]$ is G_T, defined by

$$G_T(u) = \frac{E[\text{number of packets generated in } [0,T] \text{ suffering access delay at least } u]}{\lambda T}. \tag{1.2}$$

By finding a lower bound on $G_T(N/2)$, the probability that a typical packet is not successfully transmitted within $N/2$ slots of its generation time, we obtain a lower bound on the mean access delay suffered by a typical packet.

Proposition 1.1: Under any random access algorithm and $T > 0$,

$$G_T\left(\frac{N}{2}\right) \geq 2^{-(\ln 2)(1+\lceil N/2 \rceil/T)/\lambda}. \tag{1.3}$$

In particular,

$$\liminf_{T\to\infty} G_T\left(\frac{N}{2}\right) \geq 2^{-(\ln 2)/\lambda} > (0.618)^{1/\lambda} \tag{1.4}$$

and

$$\liminf_{T\to\infty} D_T/N \geq 0.5(2^{-(\ln 2)/\lambda}) > 0.5(0.618)^{1/\lambda}. \tag{1.5}$$

The inequality (1.5) gives a sense in which the mean access delay suffered by a typical packet is at least $0.5N(0.618)^{1/\lambda}$ for any random access algorithm. Proposition 1.1 is proved in Section II.

B. Upper Bound for Asymptotically Large N

An upper bound on the achievable mean access delay of a typical packet is obtained by considering a specific random access algorithm. Given $k \geq 1$, let $\lambda_{\max}(k) = \max\{G[1 - (1 - \exp(-kG))^k] : G \geq 0\}$. Given λ with $0 < \lambda < \lambda_{\max}(k)$, let G_0 be the minimum positive solution to the equation

$$\lambda = G_0\left[1 - (1 - \exp(-kG_0))^k\right]. \tag{1.6}$$

Finally, let $\gamma_0 = (1 - \exp(-kG_0))^k$ and $d_0(k, \lambda) = \gamma_0/(1 - \gamma_0)$.

Proposition 1.2: There exists a family of random access algorithms so that if $D(k, \lambda, N)$ is the average access delay of a typical packet, then

$$\lim_{N\to\infty} D(k, \lambda, N)/N = d_0(k, \lambda).$$

The proof of Proposition 1.2 appears in Section III. The rough idea of the proof is as follows. Each new packet is transmitted in k slots. These slots are chosen at random, and occur soon after the generation time of the packet, but they are spaced out enough so that the k outcomes are approximately independent. The packet's station then waits N slots, where N is very large, so the outcomes of the first k transmissions are learned. If at least one of the transmissions is successful, then no further transmissions of the packet are necessary. Otherwise, the process of transmission begins over, with the packet being transmitted in k more slots chosen at random, there is a wait of N more slots, and so on, until the packet is successfully transmitted. If we suppose the number of packets transmitted in each slot has the Poisson distribution with parameter kG_0, then the probability a packet collides in every one of k independent attempts is γ_0. Consequently, the mean number of times that a packet needs to be retransmitted k times is $d_0(k, \lambda)$. Finally, new packets are generated at rate λ and are transmitted $k/(1 - \gamma_0)$ times on average, so G_0 should satisfy $kG_0 = \lambda k/(1 - \gamma_0)$, which is equivalent to (1.6).

The algorithm just described is similar to the basic ALOHA algorithm, and like the basic ALOHA algorithm, it is unstable for the assumed Poisson process model of packet generation times. The algorithm actually used in the proof of Proposition 1.2 is more complicated and has three phases. The first is similar to what we just described, and the second and third phases use tree algorithms for conflict resolution.

C. Comparison of Bounds

Propositions 1.1 and 1.2 combined show that for fixed λ, the mean access delay suffered by a typical packet must grow linearly with the propagation delay N. Table I gives a comparison of the upper and lower bounds on the coefficient of N given by our two propositions for some values of λ. The upper bound was obtained by computing $d_0(k, \lambda)$ for fixed λ and a range of integers k, thereby identifying the minimizing value k^* of k. For λ near zero, $d_0(k, \lambda) \approx \gamma_0$, and the optimizing value of k is approximately given by $k^* = 1/\lambda$. As λ tends to zero, the ratio of the bounds tends to 2, while the ratio is approximately 3 for $\lambda = 0.20$.

One approach for dealing with large propagation delay is to use N versions of a traditional random access algorithm (i.e., one designed and run for $N = 1$), running in parallel in an interleaved fashion. The resulting average access delay is N times larger than the access delay of the underlying traditional random access algorithm. It is thus interesting to compare the numbers in Table I with a lower bound on the access delay for a system with $N = 1$. The greatest known lower bound is provided in [1], and for $\lambda = 0.10$, 0.20, or 0.30, the values of the lower bound are 0.162, 0.366, or 0.645, respectively. These values exceed even the upper bounds given in Table I. We conclude that for $\lambda \leq 0.30$, the interleaving approach cannot yield the minimum average access delay for systems with small arrival rate and large propagation delay. In particular, this is true for $\lambda = 0.30$, in which case the optimal value of k for the algorithm used to prove Proposition 1.2 is $k = 1$, making the algorithm rather similar to the basic ALOHA algorithm. For values of λ somewhat larger than

TABLE I
COMPARISON OF LOWER AND UPPER BOUND ON THE COEFFICIENT OF N IN THE MEAN ACCESS DELAY FOR SOME VALUES OF Λ AND N TENDING TO INFINITY

λ	lower bound $0.5(2^{-\frac{\ln 2}{\lambda}})$	k^*	upper bound $d_o(k^*, \lambda)$
0.05	0.0000336	16	0.0000716
0.10	0.00410	7	0.00861
0.15	0.0203	5	0.0506
0.20	0.0452	3	0.138
0.25	0.0731	2	0.293
0.30	0.101	1	0.631
0.35	0.127	1	1.05

0.3, our upper bound to mean delay is greater than the mean delay for the part-and-try algorithm with interleaving.

II. PROOF OF LOWER BOUND ON PACKET DELAY

A. Introduction of Test Packet

Proposition 1.1 is proved in this section. Since (1.4) and (1.5) are immediate consequences of (1.3), only (1.3) needs to be established in this section. The concept of a *test packet* is used in the proof. By assumption, new packets are generated according to a Poisson process on the set $[0, +\infty)$. Consider a modified system on the same probability space in which one more packet is generated, called a test packet, at a fixed time $t > 0$. Let d_t be the random variable giving the access delay of the test packet, assuming that the test packet participates with the other packets in the random access algorithm. The connection between the access delay of a test packet and the access delay of a typical packet is given by the next lemma.

Lemma 2.1: For $T > 0$,

$$G_T(u) = \frac{1}{T}\int_0^T P[d_t \geq u]\, dt. \tag{2.1}$$

Proof: The right-hand side of (2.1) can be rewritten as $P[d_\tau \geq u]$ where d_τ is the access delay suffered by a test packet which is generated at a random time τ, where τ is uniformly distributed over the interval $[0, T]$ and is independent of the original generation process. (An enlargement of the underlying probability space may be required to construct τ.) Let P^* denote the probability distribution for the system with such a test packet with random time of generation. The generation times of packets in the interval $[0, T]$ are conditionally independent and uniform, given the number of packets generated in the interval, so that

$$E[\text{number of packets generated in } [0,T] \text{ with access delay } \geq u | k \text{ packets generated in } [0,T]] =$$
$$kP^*[d_\tau \geq u | k-1 \text{ other packets are generated in } [0,T]].$$

Therefore,

$$\begin{aligned} G_T(u) &= \frac{1}{\lambda T}\sum_{k=1}^{\infty} kP^*[d_\tau \geq u | k-1 \text{ other packets are generated in } [0,T]]\frac{\exp(-\lambda T)(\lambda T)^k}{k!} \\ &= \sum_{k=1}^{\infty} P^*[d_\tau \geq u | k-1 \text{ other packets are generated in } [0,T]]\frac{\exp(-\lambda T)(\lambda T)^{k-1}}{(k-1)!} \\ &= P[d_\tau \geq u]. \end{aligned}$$

and the lemma is proved. □

B. Completion of Proof

In order to prove Proposition 1.1, it remains to establish that

$$\frac{1}{T}\int_0^T P\left[d_t \geq \frac{N}{2}\right] dt \geq 2^{-(\ln 2)(1+\lceil N/2\rceil/T)/\lambda}. \tag{2.2}$$

It suffices to prove (2.2) for N odd because if N is even, the corresponding statement for $N-1$ implies that for N. So let N be odd and set $l = \lceil N/2 \rceil$. Consider an arbitrary but fixed algorithm. The sets $B_1, \cdots, B_N$ are deterministic, B_{N+1} is a function of θ_1, B_{N+2} is a function of (θ_1, θ_2), and so on. In the course of the proof, we will define a deterministic sequence $(\tilde{\theta}_1, \tilde{\theta}_2, \cdots)$ and a deterministic sequence of sets $(\tilde{B}_1, \tilde{B}_2, \cdots)$. For ease of notation, we define $\theta_k = \tilde{\theta}_k = 0$ for $k \leq 0$, and we set $\Theta(k) = (\cdots, \theta_0, \theta_1, \cdots, \theta_k)$ and $\tilde{\Theta}(k) = (\cdots, \tilde{\theta}_0, \tilde{\theta}_1, \cdots, \tilde{\theta}_k)$.

Consider the conditional probability

$$P\left[d_t \geq \frac{N}{2}\middle|\Theta(n-l+1) = \tilde{\Theta}(n-l+1)\right]$$

where n is a fixed integer with $0 \leq n \leq T$ and $t \in [n, n+1)$. By abuse of notation, we denote this conditional probability by $P[d_t \geq N/2 | \tilde{\Theta}(n-l+1)]$.

Note that given $\Theta(n-l+1) = \tilde{\Theta}(n-l+1)$, the sets $B_{n+1}, \cdots, B_{n+l}$ are determined. Let $B_{i_1}, \cdots, B_{i_k}$ denote the subsequence of $B_{n+1}, \cdots, B_{n+l}$ such that $t \in B_{i_j}$ for each j. Also define

$$\tilde{B}_k = B_k \cap [k-l, k) \cap [0, T] \qquad \text{for } k \leq n+l.$$

The sets $\tilde{B}_1, \cdots, \tilde{B}_{n+l}$ are deterministic since they are determined by the deterministic sequence $\tilde{\Theta}(n-l+1)$. Obviously, $\tilde{B}_k \subset B_k$. The sets $\tilde{B}_{n+1}, \cdots, \tilde{B}_{n+l}$ are subsets of the interval $[n-l+1, n+l) \cap [0, T]$, and the arrival process on this interval is independent of $\Theta(n-l+1)$.

Thus, using $\omega(B)$ to denote the number of packets with generation times in a set B,

$$\begin{aligned} P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right] &= P\left[\omega(B_{i_1}) > 0, \cdots, \omega(B_{i_k}) > 0 | \tilde{\Theta}(n-l+1)\right] \\ &\geq P\left[\omega(\tilde{B}_{i_1}) > 0, \cdots, \omega(\tilde{B}_{i_k}) > 0 | \tilde{\Theta}(n-l+1)\right] \\ &= P\left[\omega(\tilde{B}_{i_1}) > 0, \cdots, \omega(\tilde{B}_{i_k}) > 0\right]. \end{aligned}$$

By a positive correlation property of Poisson processes (see [1]), we have

$$P\left[\omega\left(\tilde{B}_{i_1}\right) > 0,\cdots,\omega\left(\tilde{B}_{i_k}\right) > 0\right] \geq \prod_j P\left[\omega\left(\tilde{B}_{i_j}\right) > 0\right] = \prod_j \left\{1 - \exp\left(-\lambda|\tilde{B}_{i_j}|\right)\right\}.$$

Therefore, for $t \in [n, n+1)$,

$$P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right] \geq \prod_{k=n+1}^{n+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}\exp\left(-\lambda|\tilde{B}_k|\right)\right\} = \prod_{k=1}^{T+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}\exp\left(-\lambda|\tilde{B}_k|\right)\right\}. \tag{2.3}$$

We claim for $0 \leq n \leq T$ and a proper choice of $\tilde{\Theta}(n-l+1)$ that

$$\int_0^T P\left[d_t \geq \frac{N}{2}\right]dt \geq \int_0^n \prod_{k=1}^{T+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}dt + \int_n^T P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right]dt. \tag{2.4}$$

If $n = 0$, the event $\Theta(n-l+1) = \tilde{\Theta}(n-l+1)$ trivially has probability one (use the fact $l \geq 1$), so that the claim is trivially true for $n = 0$. Suppose, for the sake of induction, that the claim is true for given n with $0 \leq n \leq T-1$. Applying (2.3), we obtain

$$\begin{aligned}\int_n^T P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right]dt &= \int_n^{n+1} P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right]dt + E\left[\int_{n+1}^T I_{\{d_t\geq N/2\}}\,dt\middle|\tilde{\Theta}(n-l+1)\right]\\ &\geq \int_n^{n+1}\prod_{k=1}^{T+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}dt + \min_x E\left[\int_{n+1}^T I_{\{d_t\geq N/2\}}\,dt\middle|\tilde{\Theta}(n-l+1),\ \theta(n-l+2) = x\right]\end{aligned} \tag{2.5}$$

where x ranges over values in $\{0,1,2\}$ such that $P[\theta(n-l+2) = x|\tilde{\Theta}(n-l+1)] > 0$. We define $\tilde{\theta}(n-l+2)$ to be the minimizing value of x in (2.5), so the inequality (2.5) becomes

$$\int_n^T P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+1)\right]dt \geq \int_n^{n+1}\prod_{k=1}^{T+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}dt + \int_{n+1}^T P\left[d_t \geq \frac{N}{2}\middle|\tilde{\Theta}(n-l+2)\right]dt. \tag{2.6}$$

Combining (2.4) and (2.6) establishes (2.4) for n replaced by $n+1$. Hence, (2.4) is true for $0 \leq n \leq T$ by induction.

Starting with the proved claim (2.4) with $n = T$, we obtain (using Jensen's inequality for the second inequality)

$$\begin{aligned}\frac{1}{T}\int_0^T P\left[d_t \geq \frac{N}{2}\right]dt &\geq \frac{1}{T}\int_0^T \prod_{k=1}^{T+l}\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}dt\\ &= \frac{1}{T}\int_0^T \exp\left(\sum_{k=1}^{T+l}\ln\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}\right)dt\\ &\geq \exp\left(\frac{1}{T}\int_0^T \sum_{k=1}^{T+l}\ln\left\{1 - I_{\{t\in\tilde{B}_k\}}e^{-\lambda|\tilde{B}_k|}\right\}dt\right)\\ &= \exp\left(\frac{1}{T}\sum_{k=1}^{T+l}|\tilde{B}_k|\ln\left(1 - e^{-\lambda|\tilde{B}_k|}\right)\right)\\ &\geq \exp\left(\frac{1}{T}\sum_{k=1}^{T+l}\min_{s>0} s\ln\left(1 - e^{-\lambda s}\right)\right)\\ &= \exp\left(\frac{T+l}{T}\left[\min_{s>0} s\ln\left(1 - e^{-\lambda s}\right)\right]\right)\\ &= \left(\min_{u>0}\left(1 - e^{-u}\right)^u\right)^{(T+l/T)/\lambda}\\ &= 2^{-(\ln 2)(1+l/T)/\lambda}.\end{aligned}$$

This establishes (2.2) as required, so the proof of Proposition 1.1 is complete.

III. Algorithm Providing Asymptotic Upper Bound

A. The Algorithm

Proposition 1.2 is established in this section by consideration of the random access algorithm described next. The algorithm has parameters $N, k, l, R, B, l_1, l_2, \cdots, l_{R+B}$ which have the interpretation shown in Fig. 1. It is assumed that N/l and l_i/k are integers for $1 \leq i \leq R$ and that $l = l_1 + \cdots + l_{R+B} + 1$. The behavior of the parameters as N tends to infinity is described in the next subsection. The time axis is divided into *l-intervals* (of length l), and each l-interval is subdivided into $R + B + 1$ subintervals: one *l_i-interval* for each i with $1 \leq i \leq R + B$, and a single slot at the end of each l-interval called a *Q-slot*. In turn, for $1 \leq i \leq B$, each l_i-interval is partitioned into k intervals of length l_i/k each, called *l_i/k-intervals*.

To specify the algorithm we describe the fate of a batch of packets that are generated during a particular l-interval. These packets are first transmitted k times each during the l_1-interval in the next l-interval as follows. Each packet is transmitted once during each l_1/k-interval

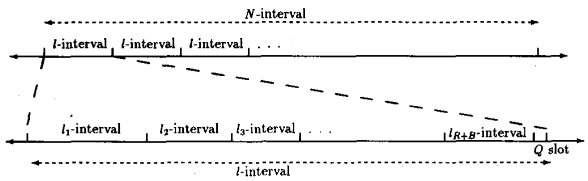

Fig. 1. Partition of time interval used by the algorithm.

of the l_1-interval, in a slot chosen at random with all slots of an l_1/k-interval having equal probability. The k transmission times of a packet within the k different l_1/k-intervals are mutually independent, and are independent of the transmission times of the other packets. The outcomes of the transmissions made during the l_1-interval are learned by all stations within N time units after the end of the l_1-interval. A packet that collides in all k attempts during an l_1-interval is called a *1-survivor*.

All 1-survivors (from the original batch) are transmitted k times in the l_2-interval that begins N time units after the end of the l_1-interval. As before, the k slots in which a 1-survivor is transmitted are chosen independently, one slot from each l_2/k-interval, using the uniform distribution over the slots of each l_2/k-interval. The outcomes of the l_2-interval are learned within N time units after the end of the l_2-interval. A 1-survivor that collides in all k attempts during the l_2-interval is called a 2-survivor. All 2-survivors are transmitted k times each during the l_3-interval starting N time units after the end of the l_2-interval. The process continues for up to R rounds. Thus, for $1 \leq i \leq R-1$, the i-survivors that are transmitted in a particular l_i-interval are all again transmitted k times in the l_{i+1}-interval which begins N time units after the l_i-interval ends.

The R-survivors continue to participate in the algorithm as follows. Each chooses a *type*, uniformly and independently, from the set $\{1,\cdots,l_{R+1}\}$. The packets of a given type contend among themselves for successful transmission using the basic binary tree algorithm. Thus, l_{R+1} distinct executions of the basic binary tree algorithm occur in parallel. Initially, all type t packets (if any) are transmitted in the tth slot of the l_{R+1}-interval that begins N time units after the end of the l_R-interval. If there is exactly one type t packet, it is known to be successfully received by the beginning of the l_{R+2}-interval which begins N time units after the end of the l_{R+1}-interval. If, however, there are two or more type t packets, they collide in slot t of the l_{R+1}-interval, and hence become $R+1$-survivors.

The l_{R+2}-interval which begins N time units after the end of the l_{R+1}-interval is dedicated to the second rounds of the executions of the binary tree algorithm, as needed. Each of the collisions in the l_{R+1}-interval is assigned a pair of slots in the l_{R+2}-interval, with the pairs assigned to distinct collisions being disjoint. Each packet involved in a collision in the l_{R+1}-interval is transmitted in one of the two slots of the pair assigned to the collision, each of the two slots being chosen with probability one half. The process is repeated. In general, for $R+2 \leq i \leq R+B$, a collision in a slot of an l_{i-1}-interval is assigned a pair of slots in the l_i-interval that begins N time units after the end of the l_{i-1}-interval. The packets involved in the collision are retransmitted in a randomly chosen slot in the pair, each slot having probability 0.5.

However, if there are more than $l_i/2$ collisions during the l_{i-1}-interval (which should happen with small probability), "overflow" is said to occur. In that case, only $l_i/2$ of the collisions in the l_{i-1}-interval are assigned pairs of slots in the l_i-interval. The packets involved in the remaining collisions in the l_{i-1}-interval are called "overflow" packets. We also say that packets involved in a collision in the l_{R+B}-interval are overflow packets. The collection of all overflow packets emerging from the original l-interval is called a batch of overflow packets. Such a batch is either empty or contains at least two packets. Packets in a batch of overflow packets participate in transmissions in Q-slots, as described in the next paragraph. It is only at this point that there is some interaction between packets that are generated in different l-intervals.

A subsequence of Q-slots with uniform spacing N is called a *track* of Q-slots. There are N/l disjoint, interleaved tracks. A batch of overflow packets is first potentially eligible for transmission in the Q-slot that begins N time units after the end of the l_{R+B}-interval associated with the batch. Furthermore, packets from that batch are only transmitted in Q-slots in the same track as that original Q-slot. Each track of Q-slots corresponds to a $D/G/1$ queue in which the customers are batches of overflow packets. One batch arrives just before each Q-slot. The batches are served in first-come, first-served order. A particular batch is served using the improved binary symmetrical tree algorithm (IBSA) for conflict resolution [2]. Thus, the service time of a batch (measured in number of Q-slots) is equal to the time needed for the IBSA to process the batch. In particular, if the batch is empty, the service time is zero.

Given the parameters $N, k, l, R, B, l_1, l_2, \cdots, l_{R+B}$, the algorithm is completely specified. It is not difficult to check that it can be implemented in a distributed fashion by the stations, each using the delayed feedback information.

B. Choice of Parameters in the Algorithm

In order to establish Proposition 1.2, we need to specify how the parameters of the algorithm just described should be chosen as N tends to infinity. We take k fixed and R large but fixed, and let l tend to infinity. Implicitly, the other variables $N, B, l_1, l_2, \cdots, l_{R+B}$ we discuss are indexed by l. We assume that $N/l \to \infty$. The following notation is used. If X and Y are numbers implicitly indexed by l, we write $X \asymp Y$ if $|X - Y|/l \to 0$ as $l \to \infty$. More generally, if X, Y, or both X and Y are random variables implicitly indexed by l, we write $X \asymp Y$ if, for any $\epsilon > 0$, there exists $\delta > 0$ such that $P[|X - Y| \geq \epsilon l] \leq \exp(-\delta l)$ for sufficiently large l.

Let λ, G_0, and γ_0 be as in Section I-B. Choose G with $G > G_0$ so that $\lambda < G(1-(1-\exp(-kG))^k)$, and let $\gamma = (1-\exp(-kG))^k$. Let h be a fixed number with $1 < h \le 1.20$. Note that $G - G_0$ and $\gamma - \gamma_0$ can be made arbitrarily small. Also, $\lambda < G(1-\gamma)$ or $1 > (\lambda/G)(1 + \gamma + \gamma^2 + \cdots)$, so we can take R so large that

$$1 > \frac{\lambda}{G}\left(1 + \gamma + \gamma^2 + \cdots + \gamma^{R-1} + G\gamma^R\left(1 + \frac{6}{1-h^{-1}}\right)\right). \tag{3.1}$$

Then take

$$l_i \asymp \frac{l\lambda\gamma^{i-1}}{G} \qquad 1 \le i \le R \tag{3.2}$$

$$l_{R+1} \asymp l\lambda\gamma^R \tag{3.3}$$

$$l_{R+i} \ge 2\left\lfloor \frac{3l_{R+1}}{h^{i-2}} \right\rfloor \qquad 2 \le i \le B \tag{3.4}$$

where $B = 2 + \lfloor \log_h(3l_{R+1}) \rfloor$. The inequality (3.1) ensures that conditions (3.2)–(3.4) can be met, while also meeting the requirement that $l = l_1 + \cdots + l_{R+b} + 1$.

The following claim is established in the remainder of this section, after three lemmas are presented.

Claim: For l and N/l large enough, the mean access delay is less than $N(\gamma + \gamma^2 + \cdots + \gamma^{R-1} + 3\gamma^R)$. Since $\gamma - \gamma_0$ can be taken arbitrarily small and R can be taken arbitrarily large, the term $\gamma + \gamma^2 + \cdots + \gamma^{R-1} + 3\gamma^R$ can be made arbitrarily close to $\gamma_0/(1-\gamma_0)$. Thus, the claim implies Proposition 1.2.

C. Lemma: k-ALOHA Success Probabilities

Let X, W, k be given, where X is a nonnegative integer-valued random variable, and W and k are positive integers. Suppose that there are k disjoint intervals of W slots each, and that there are X packets. Suppose each packet is transmitted k times: once during each interval. The slot chosen by a packet in a given interval is uniformly distributed over the W slots in the interval, and all choices are made independently. A packet is said to collide in a given interval if at least one other packet is transmitted in the same slot as the given packet in the given interval. A packet is said to be a survivor if it collides in all k intervals. Let Y denote the number of survivors.

Lemma 3.1: Let a, w, k be fixed. If $X \asymp al$ and $W \asymp wl$, then $Y \asymp a(1-\exp(-a/w))^k l$.

Before proving Lemma 3.1, we prove Lemma 3.2 under the conditions of Lemma 3.1.

Lemma 3.2: Let X_i denote the number of packets that collide in the ith interval, for $1 \le i \le k$. Then $X_i \asymp a(1-\exp(-a/w))l$.

Proof: Note that both $X \asymp al$ and $\hat{X} \asymp al$ where $\hat{X}$ is a Poisson random variable with mean al. Note also that if the number of packets is changed by one (by the addition of a new packet or deletion of an existing packet), the number X_i changes by at most two. Therefore, if the total number of packets is changed from X to $\hat{X}$, the asymptotic behavior of X_i (in the sense of $\asymp$) does not change. We thus can and do assume in the proof of this lemma, without loss of generality, that X has a Poisson distribution with mean al. Then X_i is the sum of W independent random variables with the same distribution as $U - I_{\{U=1\}}$, where U is a Poisson variable with mean a/w. Since $U - I_{\{U=1\}}$ has mean $(a/w)(1-\exp(-a/w))$ and an exponentially bounded tail distribution, the lemma follows from the Chernoff inequality. □

Proof of Lemma 3.1: Note that if the number X of packets is changed by one, then the number of survivors changes by at most $k+1$. Therefore, if the total number of packets is changed from X to $\lfloor al \rfloor$, the asymptotic behavior of Y (in the sense of $\asymp$) does not change. We thus can and do assume in the proof of this lemma, without loss of generality, that there are exactly $X = \lfloor al \rfloor$ packets with probability one.

Since Lemma 3.2 characterizes the asymptotic behavior of X_i for each i, it is useful to focus on the conditional distribution of Y given $(X_1, X_2, \cdots, X_k)$. First, another representation for this distribution is given. Imagine that there are k independent permutations of the $\lfloor al \rfloor$ packets, with all permutations being equally likely, independently of $(X_1, X_2, \cdots, X_k)$. Let $\hat{Y}$ denote the number of packets which are among the first X_i packets in the ith permutation for each i. Then $\hat{Y}$ and Y have the same conditional probability distribution given $(X_1, X_2, \cdots, X_k)$. Notice that if one of the variables X_i is changed by one, then $\hat{Y}$ changes by at most one. Therefore, $\tilde{Y}$ and Y have the same asymptotic behavior, where $\tilde{Y}$ is defined in the same way as $\hat{Y}$, but with $(X_1, X_2, \cdots, X_k)$ replaced by any random vector $(\tilde{X}_1, \tilde{X}_2, \cdots, \tilde{X}_k)$ with coordinates in $\{0, 1, \cdots, \lfloor al \rfloor\}$ such that $\tilde{X}_i \asymp a(1-\exp(-a/w))l$ for each i.

To complete the proof of Lemma 3.1, we choose the variables $\tilde{X}_1, \tilde{X}_2, \cdots, \tilde{X}_k$ to be independent binomial random variables with parameters $\lfloor al \rfloor$ and $1-\exp(-(a/w))$. Then, by the Chernoff inequality applied to the binomial distribution, $\tilde{X}_i \asymp a(1-\exp(-(a/w)))l$ as desired. Furthermore, the subset of packets that are among the first X_i in the ith permutation has the same distribution as if each packet, independently of all other packets, is chosen with probability $1-\exp(-(a/w))$. Therefore, $\tilde{Y}$ has the binomial distribution with parameters $\lfloor al \rfloor$ and $a(1-\exp(-a/w))^k$. Thus, $\tilde{Y} \asymp a(1-\exp(-a/w))^k l$ by Chernoff's inequality, and so also $Y \asymp a(1-\exp(-a/w))^k l$, as was to be proved. □

D. Lemma: Tree Algorithm in Parallel

Let ν and n be positive integers such that $n \le (1+\epsilon)\nu$ for some ϵ. Suppose n packets each choose a type, independently from the set $\{1, \cdots, \nu\}$, with each type being selected with probability $1/\nu$. Suppose that a separate execution of the basic binary tree algorithm is applied to transmit the packets of each type. Let Z denote the sum, over all n packets, of the numbers of collisions suffered by the packets.

Lemma 3.3: If $\epsilon < 0.05$ and $\nu \geq 4$, then $E[Z] \leq (1.9)\nu$.

Proof: We have $E[Z] = n\sum_{i=1}^{\infty} p_i$, where p_i is the probability that a given packet suffers at least i collisions. Calculate p_i by considering a given packet P_0. Packet P_0 suffers at least i collisions if and only if at least one of the other $n-1$ packets chooses the same type as P_0, and also makes the same binary decision (for executing the tree algorithm) after each of the first $i-1$ collisions it is involved in. Since a given one of the other $n-1$ packets so interferes with P_0 with probability $2^{1-i}/\nu$, it follows that $p_i = 1-(1-(2^{1-i}/\nu))^{n-1}$. Since $p_i \leq 2^{1-i}n/\nu$, we have for $\nu \geq 4$ and $\epsilon < 0.05$ that

$$\begin{aligned} E[Z] &\leq n\left\{1-\left(1-\frac{1}{\nu}\right)^{(1+\epsilon)\nu} + \frac{n}{\nu}\sum_{i=2}^{\infty} 2^{1-i}\right\} \\ &\leq \nu(1+\epsilon)\{1-(0.3164)^{(1+\epsilon)} + 1 + \epsilon\} \\ &\leq (1.9)\nu. \end{aligned}$$ ■

Let ξ_i denote the number of collisions, summed over all ν executions of the basic binary tree algorithm, which occur in round $i-1$. Thus, a total of $2\xi_i$ slots are needed for round i of the executions of the algorithm. Let h be a constant with $1 < h < 1.20$, and let $l_i = 2\lfloor 3\nu/h^{i-2}\rfloor$ for $i \geq 2$.

Lemma 3.4: If $\epsilon < 0.05$ and ν is sufficiently large,

$$P[2\xi_i > l_i \text{ for some } i \geq 2] \leq [3\log_h(3\nu)]\nu^{(1/h)\log_h(h/2)}. \tag{3.5}$$

Note that $(1/h)\log_h(h/2)$ is negative, and its magnitude can be made as large as desired by taking h near 1. Since $1 < h < 1.20$, $(1/h)\log_h(h/2) < -2.3$.

Proof: Let Y be a Poisson random variable with mean $(1+\epsilon)\nu$. By the central limit theorem, for ν sufficiently large

$$\frac{1}{3} \leq P[Y \geq (1+\epsilon)\nu] \leq P[Y \geq n].$$

Consider a different experiment, the same as the original, but in which there are Y packets rather than n packets. Let P_ϵ denote the probability measure induced on $(\xi_i: i \geq 2)$ under the different experiment. Since $P_\epsilon[2\xi_i > l_i$ for some $i|Y=y]$ is increasing in y, for ν sufficiently large that $P[Y \geq n] \geq 1/3$,

$$P[2\xi_i > l_i \text{ for some } i] \leq 3P_\epsilon[2\xi_i > l_i \text{ for some } i]. \tag{3.6}$$

Under probability measure P_ϵ, ξ_i has the binomial distribution with parameters $\nu 2^{i-2}$ and p_i, where

$$\begin{aligned} p_i &= 1-(1+\Delta_i)e^{-\Delta_i} \\ \Delta_i &= 2^{2-i}(1+\epsilon). \end{aligned}$$

Since $\xi_2 \leq \nu$ and $\xi_3 \leq 2\nu$ with probability one, $P[2\xi_i \geq l_i] = 0$ for $i=2$ or $i=3$. Since $l_i = 0$ for $i \geq 3 + \lfloor\log_h(3\nu)\rfloor$, the events $\{2\xi_i \leq l_i$ for $i = 3+\lfloor\log_h(3\nu)\rfloor\}$ and $\{2\xi_i \leq l_i$ for $i \geq 3+\lfloor\log_h(3\nu)\rfloor\}$ are identical. Therefore,

$$\begin{aligned} &P_\epsilon[2\xi_i > l_i \text{ for some } i] \\ &\quad = P_\epsilon[2\xi_i > l_i \text{ for some } i \text{ with } 4 \\ &\qquad \leq i \leq 3+\lfloor\log_h(3\nu)\rfloor]. \end{aligned} \tag{3.7}$$

Next, the facts: 1) if $\mathscr{B}$ is a binomial random variable with parameters (m,p), then $P[\mathscr{B} > \lfloor x\rfloor] \leq (mpe/x)^x$, 2) $p_i \leq \Delta_i^2$, and 3) $e(1+\epsilon)^2 \leq 3$ yield

$$\begin{aligned} P_\epsilon[2\xi_i > l_i] &= P_\epsilon\left[\xi_i > \left\lfloor\frac{3\nu}{h^{i-2}}\right\rfloor\right] \\ &\leq \left(\frac{\nu 2^{i-2}(2^{2-i}(1+\epsilon))^2 e h^{i-2}}{3\nu}\right)^{3\nu/h^{i-2}} \\ &= \left\{\left(\frac{h}{2}\right)^{i-2}\frac{e(1+\epsilon)^2}{3}\right\}^{3\nu/h^{i-2}} \\ &\leq \left(\frac{h}{2}\right)^{3\nu(i-2)/h^{i-2}}. \end{aligned}$$

The expression $(i-2)/h^{i-2}$ as a function of $i-2$ in $(1/\ln(h), +\infty)$ decreases to zero, so for ν sufficiently large,

$$\min_{1 \leq i-2 \leq 1+\log_h(3\nu)} \frac{i-2}{h^{i-2}} = \frac{1+\log_h(3\nu)}{3h\nu} \geq \frac{\log_h(3\nu)}{3h\nu}. \tag{3.8}$$

Therefore, if ν is sufficiently large,

$$P_\epsilon[2\xi_i > l_i] \leq \left(\frac{h}{2}\right)^{(1/h)\log_h(3\nu)} \qquad 4 \leq i \leq 3+\log_h(3\nu). \tag{3.9}$$

Combining (3.6), (3.7), (3.9), and using a union bound yields

$$\begin{aligned} P_\epsilon[2\xi_i > l_i \text{ for some } i] &\leq 3\sum_{i=4}^{\lfloor 3+\log_h(3\nu)\rfloor} P_\epsilon[2\xi_i > l_i] \quad (3.10) \\ &\leq 3\log_h(3\nu)\left(\frac{h}{2}\right)^{(1/h)\log_h(3\nu)}. \quad (3.11) \end{aligned}$$

Finally, applying the bound

$$\left(\frac{h}{2}\right)^{(1/h)\log_h(3\nu)} \leq \left(\frac{h}{2}\right)^{(1/h)\log_h(\nu)} = \nu^{(1/h)\log_h(h/2)}$$

to (3.11) yields the lemma. □

E. Lemma: Poisson Variables on Events of Small Probability

The following lemma bounds the *conditional* moments of a Poisson random variable, given an event of small probability.

Lemma 3.5: Let X be a Poisson random variable with mean μ, and let A be an event with $P[A] \leq \epsilon$. Then

$$E\left[X^j I_{\{X \in A\}}\right] \leq (3\mu)^j(\epsilon + 11(.3)^\mu). \tag{3.12}$$

Proof: Given μ and ϵ, the left-hand side of (3.12) is clearly maximized when there is an integer J such that $\{X > J\} \subset A \subset \{X \geq J\}$. We assume that such J exists. Then, for some γ with $0 \leq \gamma \leq e^{-\mu}\mu^J/J!$,

$$P[A] = \gamma + \sum_{k=J+1}^{\infty} \frac{e^{-\mu}\mu^k}{k!}$$

and

$$E\left[X^j I_{\{X \in A\}}\right] = \gamma J^j + \sum_{k=J+1}^{\infty} \frac{e^{-\mu}\mu^k k^j}{k!}. \quad (3.13)$$

Note that if $J \geq 3\mu$, then

$$E\left[X^j I_{\{X \in A\}}\right] \leq \sum_{k=\lceil 3\mu \rceil}^{\infty} \frac{e^{-\mu}\mu^k k^j}{k!}. \quad (3.14)$$

On the other hand, if $J < 3\mu$, then $k^j \leq (3\mu)^j$ for $J \leq k < 3\mu$, so that

$$E\left[X^j I_{\{X \in A\}}\right] \leq (3\mu)^j \left[\gamma + \sum_{k=J+1}^{\lceil 3\mu \rceil - 1} \frac{e^{-\mu}\mu^k}{k!}\right] + \sum_{k=\lceil 3\mu \rceil}^{\infty} \frac{e^{-\mu}\mu^k k^j}{k!}. \quad (3.15)$$

Hence, in general,

$$E\left[X^j I_{\{X \in A\}}\right] \leq (3\mu)^j \left[\gamma + \sum_{k=J+1}^{\infty} \frac{e^{-\mu}\mu^k}{k!}\right] + \sum_{k=\lceil 3\mu \rceil}^{\infty} \frac{e^{-\mu}\mu^k k^j}{k!}. \quad (3.16)$$

Recognizing $P[A]$ in square brackets in (3.16) and using $k! \geq (k/e)^k$, we obtain

$$E\left[X^j I_{\{X \in A\}}\right] \leq (3\mu)^j \epsilon + (\mu e)^j \sum_{k=\lceil 3\mu \rceil}^{\infty} e^{-\mu}\left(\frac{e}{3}\right)^{k-j}$$

$$= \epsilon(3\mu)^j + \frac{(\mu e)^j e^{-\mu}\left(\frac{e}{3}\right)^{\lceil 3\mu \rceil - j}}{1 - \frac{e}{3}}$$

$$= (3\mu)^j \left(\epsilon + \frac{\left(\frac{e}{3}\right)^{\lceil 3\mu \rceil} e^{-\mu}}{1 - \frac{e}{3}}\right)$$

$$\leq (3\mu)^j \left(\epsilon + \frac{\left(\frac{e^2}{27}\right)^{\mu}}{1 - \frac{e}{3}}\right)$$

which yields the lemma. □

F. Bounding the Delay of Algorithm

In this section, we apply the lemmas of the previous three subsections in order to bound the average access delay for the algorithm described in Section III-A. In particular, the claim in Section III-B is verified, establishing Proposition 1.2.

Let Y_0 denote the number of packets that are generated in an l-interval, and for $1 \leq i \leq R + B$, let Y_i denote the number of i-survivors from among the Y_0 originally generated packets. Let $\overline{W}$ denote the mean number of Q-slots in a track of Q-slots that a typical batch of overflow packets must wait before beginning transmission. Note that $\overline{W}$ is greater than or equal to the mean number of packets waiting to be transmitted in the track, just before a Q-slot, not including the packets in the new batch. It can be seen that the access delay of a typical packet D satisfies

$$\frac{D}{N} \leq \frac{2l}{N} + \frac{E[Y_1 + \cdots + Y_{R+B}]}{\lambda l} + \frac{\overline{W}}{\lambda l}. \quad (3.17)$$

The first term on the right-hand side of (3.17) accounts for the delay packets can suffer before their first transmission, and the delays packets can suffer between N slots after their first transmission in an l_i-interval until their first transmission in an l_{i+1}-interval (or in a Q-slot if $i = R + B$). This term can be made small as N tends to infinity by simply letting l tend to infinity more slowly. The second term on the right-hand side of (3.17) accoaunts for the rest of the access delay spent by packets from the time that it is learned that they are 1-survivors until it is learned that they are successfully transmitted or until the first Q-slot after it is learned that they are $R + B$ survivors. The final term on the right-hand side accounts for the delay packets spend after they join the queue associated with a track of Q-slots. In what follows, we first consider Y_i for $1 \leq i \leq R$, then $Y_{R+1} + \cdots + Y_{R+B}$, and then $\overline{W}$.

By the assumption that new packets are generated according to a Poisson process, $Y_0 \asymp \lambda l$. Taking (X, W, a, w) in Lemma 3.1 equal to $(Y_i,\ l_{i+1}/k,\ \lambda\gamma^i,\ \lambda\gamma^i/kG)$ yields, by induction on i, that $Y_i \asymp \lambda\gamma^i l$ for $0 \leq i \leq R$. For $1 \leq i \leq R$ and $\epsilon > 0$,

$$(\lambda\gamma^i - \epsilon) l I_{A^c} \leq Y_i \leq (\lambda\gamma^i + \epsilon) l + I_A Y_0 \quad (3.18)$$

where I_E denotes the indicator function and E^c the complement of an event E, and $A = \{|Y_i - \lambda\gamma^i l| > \epsilon l\}$. There exists $\delta > 0$ so that $P[A] \leq \exp(-\delta l)$ for all sufficiently large l. For such l, we apply Lemma 3.5 to obtain

$$E[I_A Y_0] \leq (3\lambda l)\left(\exp(-\delta l) + 11(0.3)^{\lambda l}\right) \overset{l \to \infty}{\to} 0. \quad (3.19)$$

Combining (3.18) and (3.19) implies that

$$E[Y_i] \asymp \lambda\gamma^i l. \quad (3.20)$$

Next, we consider the sum $Y_{R+1} + \cdots + Y_{R+B}$.

Note that $Y_R \asymp l_{R+1}$. Let H_0 denote the event that $Y_R > l_{R+1}(1 + \epsilon)$, and let H_1 denote the event that overflow occurs in any of the intervals associated with the original set of Y_0 packets (which contains Y_R R-survivors).

Finally, let Z denote the sum, over all the Y_R R-survivors, of the number of collisions they would suffer if the l_{R+1} basic binary tree algorithms were to serve all the packets to completion (without constraint of overflow). If H_1 is not true, then $Y_{R+1} + \cdots + Y_{R+B} = Z$, and in general, $Y_{R+1} + \cdots + Y_{R+B} \leq BY_0$. Therefore,

$$Y_{R+1} + \cdots + Y_{R+B} \leq ZI_{H_0^c \cap H_1^c} + BY_0 I_{H_0 \cup H_1}. \quad (3.21)$$

Lemma 3.3 yields

$$E[ZI_{H_0^c \cap H_1^c}] \leq E[ZI_{H_0^c}] \leq E[Z|H_0^c] \leq (1.9)l\lambda\gamma^R. \quad (3.22)$$

Since $Y_R \asymp l_{R+1}$, we have $P[H_0] \leq \exp(-\delta l)$ for some $\delta > 0$ and all sufficiently large l. Also, as long as $0 < \epsilon < 0.05$, Lemma 3.4 implies that

$$P[H_1] \leq l^{-2.3} \quad (3.23)$$

for sufficiently large l. Lemma 3.5 thus yields

$$E[BY_0 I_{H_0 \cup H_1}]$$
$$\leq 3B\lambda l\left(\exp(-\delta l) + l^{-2.3} + 11(.3)^{\lambda l}\right) \overset{l\to\infty}{\to} 0. \quad (3.24)$$

Combining (3.21), (3.22), and (3.24), we obtain

$$E[Y_{R+1} + \cdots + Y_{R+B}] \leq (1.95)\lambda\gamma^R l \quad (3.25)$$

for sufficiently large l. Let Y_{R+B+1} denote the size of the batch of overflow packets emerging from the Y_0 packets generated in an l-interval. We have $Y_{R+B+1} \leq I_{H_1}Y_0$, and (3.23) holds for sufficiently large l. Thus, for sufficiently large l,

$$E\left[Y_{R+B+1}^j\right] \leq (3\lambda l)^j (l^{-2.3} + 11(.3)^{\lambda l}) \overset{l\to\infty}{\to} 0 \quad (3.26)$$

for $j = 1, 2$.

Let X denote the number of slots needed by the IBSA to process the Y_{R+B+1} packets. By [2, eq. (3.4), (3.8)], $E[X^j|Y_{R+B+1}] \leq (8Y_{R+B+1}/3)^j$ for $j = 1, 2$, so that

$$E[X^j] \leq E\left[\left(\frac{8Y_{R+B+1}}{3}\right)^j\right] \overset{l\to\infty}{\to} 0. \quad (3.27)$$

Since X has the distribution of the service time of the $D/G/1$ queue associated with a track of Q-slots, the formula for the waiting time in a $D/G/1$ queue yields that

$$\overline{W} = \frac{E[X^2] - E[X]}{2(1 - E[X])} \overset{l\to\infty}{\to} 0. \quad (3.28)$$

Combining (3.17), (3.20), (3.25), and (3.28) implies the claim in Section III-B, and hence Proposition 1.2 is proved.

IV. Summary

We considered the access delay in a multiple-access sytem with large propagation delay. The model given is a traditional one for collision access communications, and does not permit the use of forward error correction. It is shown that the mean access delay grows at least linearly with the propagation delay because there is a lower bound on the probability that a packet is not successfully transmitted within $N/2$ slots of its generation time, where N is the propagation delay. As discussed in Section I, for large propagation delays, the access algorithm given in Section III provides the following performance. For $\lambda \leq 0.2$, the average access delay is within a factor of three of the lower bound, and for $\lambda \leq 0.3$, the average access delay is smaller than that possible by the approach of interleaving N versions of a random access algorithm designed for $N = 1$.

Acknowledgment

This work was supported in part by a travel grant from the US National Academy of Sciences, and the work of B. Hajek was supported by NSF Contract NCR 90-04355.

References

[1] B. S. Tsybakov and N. B. Likhanov, "Lower bound for the delay in multiple access system," *Probl. Peredachi Inform.*, vol. 27, pp. 73–88, July–Sept. 1991. English transl. in *Problems Inform. Transmission.*

[2] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Probl. Peredachi Inform.*, vol. 14, pp. 32–59, Oct.–Dec. 1978. English transl. in *Problems Inform. Transmission.*