Network Control by Bayesian Broadcast

RONALD L. RIVEST

Abstract—A transmission control strategy is described for slotted-ALOHA-type broadcast channels with ternary feedback. At each time slot, each station estimates the probability that n stations are ready to transmit a packet for each n, using Bayes' rule and the observed history of collisions, successful transmissions, and holes (empty slots). A station transmits a packet in a probabilistic manner based on these estimates. This strategy is called Bayesian broadcast. An elegant and very practical strategy—pseudo-Bayesian broadcast—is then derived by approximating the probability estimates with a Poisson distribution with mean ν and further simplifying. Each station keeps a copy of ν, transmits a packet with probability $1/\nu$, and then updates ν in two steps:

- For collisions, increment ν by $(e-2)^{-1} = 1.39221\cdots$. For successes and holes, decrement ν by 1.
- Set ν to $\max(\nu + \hat{\lambda}, 1)$, where $\hat{\lambda}$ is an estimate of the arrival rate λ of new packets into the system.

Simulation results are presented showing that pseudo-Bayesian broadcast performs well in practice, and methods that can be used to prove that certain versions of pseudo-Bayesian broadcast are stable for $\lambda < e^{-1}$ are discussed.

I. Introduction

We propose a new strategy for the problem of controlling traffic on a local-area or satellite broadcast communications network. We begin by first presenting a strategy (called Bayesian broadcast) which is powerful but unlikely to be cost effective in practice. The name Bayesian broadcast was chosen because each station uses Bayes' rule to estimate dynamically the probability that n stations are active, for each n. The stations calculate a broadcast (or transmission) probability that is optimum given the available global information. This strategy is essentially equivalent to a proposal of An and Gelenbe [1, pp. 305–306], based on earlier work by Segall [2].

Our new strategy (which we call pseudo-Bayesian broadcast) is an extremely simple and elegant approximation to Bayesian broadcast. We give simulation results showing that pseudo-Bayesian broadcast is exceptionally effective and stable in practice.

Consider a network with some number (possibly infinite) of stations. Each station is given packets to transmit by an associated processor. In practice a station may have a queue of packets ready to send if its processor is generating packets more quickly than the station can transmit them. However, we assume that each station has at most one packet to transmit at any time. We say a station is *active* if it has a packet to transmit; otherwise, it is *inactive*.

We assume time is divided into *slots*, each long enough to transmit one packet (the "slotted ALOHA" or "S-ALOHA" model). Our procedures generalize for other models; however, we do not treat these issues here.

When a slot begins each active station must decide, either deterministically or stochastically, whether or not to transmit its packet. There are three possible outcomes:

- a *hole* if no stations transmit;
- a *success* if one station transmits; or
- a *collision* if more than one station transmits.

We assume that each station in the network can tell which of the three possible outcomes has occurred—this is the *ternary feedback* model.

We assume that the network objective is to minimize the average delay experienced by a packet between the time it is given to a station and the time it is successfully transmitted; by Little's result [3] this is equivalent to minimizing the average backlog in the network. Each station will have a common *control strategy* specifying how often it will transmit packets, including how often it will retransmit a packet which was involved in a collision.

Our approach has the following general form. Just before slot t begins, each station k in the network computes a value for its *broadcast probability* $b_{k,t}$. Then station k will transmit a packet (if it has one) with probability $b_{k,t}$, independent of whether previous attempts had been made to transmit that packet.

We assume that each station k computes $b_{k,t}$ from the globally available network history, indicating whether each slot was a hole, a success, or a collision. Since the stations only use *global* information to compute the broadcast probabilities $b_{k,t}$, each station will compute the same value b_t for $b_{k,t}$, and our Bayesian updating procedure will be relatively straightforward.

In Section II we develop the "theory" of Bayesian broadcast, showing how each station can choose a broadcast probability for each slot which is optimum given the available global information. However, the full Bayesian broadcast is a bit demanding to implement, so in Section III we provide a very simple practical implementation based on these ideas, which we call pseudo-Bayesian broadcast. In Section IV we present some very encouraging experimental results on the average backlog when using pseudo-Bayesian broadcast. In Section V of this paper we review related work on this problem, and relate our results to this work.

Manuscript received September 30, 1986. This work was supported by the National Science Foundation under Grant MCS-80-06938.
The author is with the MIT Laboratory for Computer Science, Room NE43-324, 545 Technology Square, Cambridge, MA 02139, USA.
IEEE Log Number 8611430.

0018-9448/87/0500-0323$01.00

II. Bayesian Broadcast

Let N_t denote the number of active stations at time t (i.e., the number of stations which are ready to transmit a packet). This value will decrease with successes and increase when a processor gives an inactive station packet.

To motivate our development, we begin by considering the "complete knowledge" case where each station knows the value of N_t before slot t begins. This is unrealistic, since N_t cannot be determined from the available information, but it is of interest to determine how the stations should act in this case.

How likely is it to have a hole, success, or collision for a given broadcast probability b_t (and *waiting probability* $w_t = 1 - b_t$) and given value $N_t = n$? The probabilities are

$$P(\text{hole} \mid N_t = n) = H_{b_t}(n) = w_t^n \tag{1}$$

$$P(\text{success} \mid N_t = n) = S_{b_t}(n) = n \cdot b_t \cdot w_t^{n-1} \tag{2}$$

$$P(\text{collision} \mid N_t = n) = C_{b_t}(n) = 1 - H_{b_t}(n) - S_{b_t}(n). \tag{3}$$

The optimum value for b_t is

$$b_t = 1/N_t; \tag{4}$$

this maximizes $S_{b_t}(N_t)$. Note that b_t depends only on N_t.

If b_t is chosen optimally as $1/N_t$, the expected number of stations attempting to transmit will be one, and the probabilities of holes, successes, and collisions will be

$$H_{1/N_t}(N_t) = \left(1 - \frac{1}{N_t}\right)^{N_t} \approx \frac{1}{e} \tag{5}$$

$$S_{1/N_t}(N_t) = \left(1 - \frac{1}{N_t}\right)^{N_t - 1} \approx \frac{1}{e} \tag{6}$$

$$C_{1/N_t}(N_t) = 1 - H_{1/N_t}(N_t) - S_{1/N_t}(N_t) \approx 1 - \frac{2}{e}. \tag{7}$$

(The approximations hold for large N_t.)

However, the stations will typically not know the correct value for N_t. For example, some inactive stations may have received newly generated packets during slot $t-1$ which they will be ready to transmit during slot t.

In the first procedure we describe, which we call the Bayesian broadcast algorithm, each station will use the evidence available up to time t to estimate the likelihood $p_{n,t}$ that $N_t = n$ for each $n \geq 0$. That is,

$$p_{n,t} = \Pr(N_t = n), \qquad \text{for } n = 0, \cdots \tag{8}$$

given the available evidence. We call this procedure Bayesian broadcast, since it relies on Bayesian reasoning to estimate $\bar{p}_t = (p_{0,t}, p_{1,t}, \cdots)$.

The Bayesian broadcast procedure described here is not new; it is essentially the same as the proposal of An and Gelenbe [1, pp. 305–306] based on the technique proposed by Segall [2] to estimate recursively the number of stations waiting to transmit a packet. The only difference between our formulation and theirs is that in our version new packets are not transmitted immediately with probability one, but rather with the same transmission probability that is used for the backlogged packets.

In the Bayesian broadcast procedure, each station begins with the initial distribution $\bar{p}_0 = (1, 0, 0, \cdots)$—it assumes that all stations are inactive. Each station will compute the *same* vector $\bar{p}_t$ using the available *global* feedback information. The vector $\bar{p}_t = (p_{0,t}, \cdots)$ summarizes the global information available about N_t.

With the Bayesian broadcast procedure, each station performs the following four steps during each time slot.

- Compute the optimal broadcast probability b_t from the initial probability vector $\bar{p}_t$.
- If the station is active, transmit its packet with probability b_t.
- Perform a Bayesian update of $\bar{p}_t$ (the initial probability distribution for N_t) to obtain $\bar{p}_t'$ (the final probability distribution for N_t), using the evidence (hole, success, or collision) observed in time slot t.
- Convert the final probabilities $\bar{p}_t'$ for N_t into initial probabilities $\bar{p}_{t+1}$ for N_{t+1} by considering the generation of new packets and the fact that a packet may have been successfully transmitted during time slot t (i.e., modeling the flow of packets into and out of the system).

In Sections II-A–C below we consider the details involved in the preceding steps.

A. Computing the Broadcast Probability

One can choose b_t to maximize the expected chance of a success, even though there is uncertainty about N_t as summarized in $\bar{p}_t$, since

$$E(P(\text{success at time } t)) = \sum_n p_{n,t} \cdot S_{b_t}(n). \tag{9}$$

Given $\bar{p}_t$, this is a polynomial in the unknown variable b_t. In practice there would be at most a finite number of nonzero coefficients at any time, so that we can compute the value $\hat{b}_t$ which maximizes (9) by differentiating and root finding.

(In practice the computation required to compute b_t would probably be excessive. One could use the approximation $(E(N_t))^{-1}$. However, we believe that in practice the pseudo-Bayesian broadcast algorithm to be described later will be the best choice.)

B. Bayesian Updating of the Probability Vector

We now describe how each station computes its *final* probability distribution for N_t, given that slot t was a hole, a success, or a collision. This problem is well suited for an application of Bayes' rule:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}. \tag{10}$$

(The final probability $P(H|E)$ of a hypothesis H, after evidence E is received, is equal to the initial probability $P(H)$ of H times the probability $P(E|H)$ that E will occur given H, divided by the overall probability $P(E)$ of evidence E.)

We have a hypothesis $N_t = n$, for each $n \geq 0$. The values $p_{n,t}$ are the initial probabilities of these hypotheses before the evidence from time slot t is considered.

Let $p'_{n,t}$ denote the final probability $P(N_t = n | E_t)$ where E_t is the slot t evidence (hole, success, or collision), and let

$$\bar{p}'_t = (p'_{0,t}, p'_{1,t}, \cdots) \tag{11}$$

denote the corresponding final probability vector. The $p'_{n,t}$ are easily obtained using Bayes' rule by multiplying each initial probability $p_{n,t}$ by the appropriate likelihood $H_{b_t}(n)$, $S_{b_t}(n)$, or $C_{b_t}(n)$ according to whether a hole, success, or collision was observed, and then normalizing so that the $p'_{n,t}$ add up to one.

This completes our description of how each station incorporates the slot t evidence into its probability distribution for N_t. The resulting distribution makes the best possible use of the globally available information; one cannot improve over this application of Bayes' rule. (See [4, ch. 39].) However, it may be possible to achieve better results by using information local to the transmitting processors, or by somehow using a strategy to minimize the long-run delay which is not a "one-step look-ahead" strategy as is Bayesian broadcast. We do not pursue these possibilities here.

C. Converting the Final Probabilities $\bar{p}'_t$ into the Initial Probabilities $\bar{p}_{t+1}$

Finally, we convert $\bar{p}'_t$, the final probability distribution for N_t, into an initial distribution for N_{t+1}. Why might $p_{i,t+1}$ be different than $p'_{i,t}$? First, if slot t was a success, the expected number of active stations will decrease by one. Second, we expect some inactive stations to receive new packets from their processors during slot t, so the expected number of active stations will increase for this reason.

1) Modeling Successful Packet Transmission: We model the effect of successes as follows. We let $p''_{n,t}$ denote a station's estimate of the probability that the number of active stations is n, taking into account the evidence from the channel, including the effect of a success on the number of active stations.

If time slot t contained a success, then we set

$$p''_{n,t} = p'_{n+1,t}, \qquad \text{for } n = 1, \cdots \tag{12}$$

(note that $p'_{0,t} = 0$ if $E_t =$ success); otherwise, we set

$$p''_{n,t} = p'_{n,t}, \qquad \text{for } n = 0, \cdots. \tag{13}$$

The vector $\bar{p}''_t = (p''_{0,t}, p''_{1,t}, \cdots)$ is used as input into the next step, where the generation of new packets is taken into account.

2) Modeling the Generation of New Packets: There are many ways to model the generation of new packets. We are actually concerned with the rate at which stations "become active," i.e., convert from having no packet to send during time slot t to having a packet to send during time slot $t+1$. We take the usual approach and assume that new packets arrive according to a Poisson distribution with parameter λ, and that λ is estimated reasonably accurately (let us call the estimate $\hat{\lambda}$). We can compute the initial probabilities for N_{t+1}:

$$p_{n,t+1} = \sum_{j=0}^{n} p''_{j,t} \cdot P_{\hat{\lambda}}(n-j). \tag{14}$$

Here $P_{\hat{\lambda}}(n-j)$ denotes the value of the Poisson density function at point $n-j$; i.e., the estimated probability that $n-j$ new packets will arrive during a time slot. This completes our description of the Bayesian broadcast procedure since we now have our initial estimates for the distribution of N_{t+1} for the next time slot.

III. The Pseudo-Bayesian Broadcast Algorithm

We now present a practical implementation of the above ideas, which we call the pseudo-Bayesian broadcast algorithm. We derive this algorithm by assuming that $\bar{p}_t$ can be reasonably approximated by a Poisson distribution with mean ν; the station's value of ν at time t represents the station's estimate of N_t. We use the notation ν rather than the subscripted form ν_t for convenience in this section: ν now denotes a changeable control parameter for the stations. Let

$$P_\nu(n) = \frac{e^{-\nu} \cdot \nu^n}{n!} \tag{15}$$

denote the Poisson density at n for Poisson parameter ν. Each station will keep only ν, rather than the vector $\bar{p}_t$, and will approximate the initial probability $p_{n,t}$ by $P_\nu(n)$.

To develop the pseudo-Bayesian broadcast and probability updating procedure, we first consider the equations that would be used for a true Bayesian update of the Poisson approximation for $\bar{p}_t$ if b_t is the actual broadcast probability (and $w_t = 1 - b_t$). These equations represent the unnormalized final probability values:

$$P_\nu(n) \cdot H_{b_t}(n) = e^{-\nu b_t} \cdot P_{\nu w_t}(n) \tag{16}$$

$$P_\nu(n) \cdot S_{b_t}(n) = \nu b_t \cdot e^{\nu b_t} \cdot P_{\nu w_t}(n-1) \tag{17}$$

$$P_\nu(n) \cdot C_{b_t}(n) = P_\nu(n) \cdot \left(1 - H_{b_t}(n) - S_{b_t}(n)\right). \tag{18}$$

From (9) and (17) it is easy to compute the best broadcast probability:

$$b_t = \min\left(\frac{1}{\nu}, 1\right); \tag{19}$$

no complicated root-finding is needed. Thus we have derived our first practical benefit from the Poisson approximation: it becomes trival to compute the desired broadcast probability b_t.

We next consider the problem of updating ν in as Bayesian a manner as possible, while preserving our Poisson approximation. We shall see that for holes and successes we can use Bayes' rule exactly, while for collisions we must introduce an approximation error to preserve the Poisson approximation.

From (16) we see that for holes the Bayesian updating takes a simple form since the resulting distribution will also be Poisson with mean $\nu w_t = \max(\nu - 1, 0)$. In other words, when a hole occurs, the stations reduce their estimate of the expected number of active stations by one, unless ν is already less than one, in which case they set ν to zero.

From (17) we see that for successes the Bayesian updating and success modeling also takes a simple form. Here (17) will yield a Poisson distribution with mean $\nu - 1$ shifted one place to the right. However, the effect of modeling a successful transmission shifts the distribution one place to the left. The net result is that the Poisson assumption remains valid, and each station should decrement its state variable ν by 1.

If there is a collision, Bayes' rule will not yield a Poisson distribution for the final probabilities. However, we approximate the result by a Poisson distribution by setting ν to be the mean of the resulting distribution, which is (using x to denote $\nu \cdot b_t$):

$$\nu + \frac{x^2}{e^x - x - 1} \tag{20}$$

which simplifies in the case $\nu \geq 1$, $b_t = 1/\nu$ to

$$\nu + \frac{1}{e - 2}. \tag{21}$$

(It is somewhat surprising that we get a constant increment to ν in this case.) For $\nu \leq 1$, (20) is reasonably well approximated by

$$2.39221 \tag{22}$$

which is the value (20) yields for $\nu = 1$. Using (22) is equivalent to requiring that $\nu \geq 1$ at all times. The following algorithm makes this simplification. We now summarize the above analysis, assumptions, and approximations in the following presentation of the pseudo-Bayesian broadcast algorithm.

The Pseudo-Bayesian Broadcast Procedure: Each station maintains a copy of ν and, during each slot,

- broadcasts with probability $1/\nu$ if it has a packet;
- decrements ν by 1 if the current slot is a hole or a success, and increments ν by $(e - 2)^{-1} = 1.392211 \cdots$ if the current slot is a collision;
- sets ν to $\max(\nu + \hat{\lambda}, 1)$, where $\hat{\lambda}$ is an estimate of the arrival rate λ of new packets into the system. (For example, one might estimate the arrival rate by the observed average success rate, or use the constant value $\hat{\lambda} = e^{-1}$. See Section V for some discussion of this issue.)

We note that the pseudo-Bayesian broadcast procedure actually only needs *binary* feedback since it only needs to distinguish collisions from noncollisions.

We note that since each station now only maintains a single parameter ν, it would be simple to broadcast ν with every packet. In this way stations which have just powered-up can "synchronize" easily.

IV. Experimental Results

The pseudo-Bayesian broadcast procedure was simulated for 10^6 time slots (40 trials of 25 000 steps each) for a number of different Poisson arrival rates λ. For each trial the average backlog (i.e., the average of N_t over the 25 000 steps) was computed as a sample data point. The mean N_λ^{av} and standard deviation σ_λ^{av} of these 40 sample points are given in Table I.

TABLE I
Mean and Standard Deviation for Backlog

λ	N_λ^{av}	σ_λ^{av}
0.10	0.144	0.0069
0.15	0.28	0.012
0.20	0.555	0.85
0.25	1.00	0.097
0.30	2.31	0.32
0.32	3.73	0.54
0.34	7.03	1.58
0.35	12.35	3.82
0.36	28.38	20.86
0.37	63.11	39.7

For these experiments the arrival rate λ was estimated by setting $\hat{\lambda}$ to 0.500 initially and then using the recursion

$$\hat{\lambda} = 0.995\hat{\lambda} + 0.005s \tag{23}$$

where $s = 1$ if the current slot contained a success, and $s = 0$ otherwise. This estimates the average success rate, which will only be a good estimate of the average arrival rate when the scheme is acting in a stable manner. Tsitsiklis [5] discusses this issue further.

In Table II we give F_λ^{av}, the average (over the 40 trials) frequency of having no backlog during a trial of 25 000 steps, and L_λ^{av}, the average last step number when there was no backlog. The last statistic supports the view that the method is stable for $\lambda = 0.36$ and unstable for $\lambda = 0.37$.

TABLE II
Frequency of No Backlog and Last Instance of No Backlog

λ	F_λ^{av}	L_λ^{av}
0.10	2320	24991
0.15	3274	24995
0.20	3154	24964
0.25	4204	24993
0.30	3714	24966
0.32	3235	24967
0.34	2312	24867
0.35	1701	24433
0.36	1026	22361
0.37	424	13605

It is clear from (6) that we should not expect to be able to handle $\lambda > e^{-1} = 0.3678 \cdots$. We see that the algorithm becomes unstable for $\lambda > e^{-1}$, as expected.

V. Discussion

In this section we discuss the pseudo-Bayesian broadcast strategy in relation to previous work. Gallager [6] provides an excellent overview of the state of the art in multiaccess channels, and the special issue of this Transactions [7] contains many excellent papers on this topic. Tanenbaum [8] surveys a number of possible approaches to this problem, in a more general framework. Note that the S-ALOHA model used here differs from the more common "Ethernet" model [9] since here we have fixed length packets and fixed length time slots.

In the original ALOHA scheme [10] a station broadcasts a new packet in the next slot. (A policy of this sort is called immediate first transmission (IFT) instead of delayed first transmission (DFT).) If a collision occurred, then the packet becomes *backlogged* and is broadcast in succeeding slots with a fixed probability f until it is successfully transmitted. We note that in our scheme new packets and backlogged packets are treated identically—the control policy does not distinguish them.

The idea of decreasing the broadcast probabilities in the event of a collision, and increasing it in the event of a success and/or a collision, is not new. For example, Gerla and Kleinrock [11] discuss a number of adaptive strategies for the S-ALOHA network, some of which do not distinguish new from backlogged packets, and which may be sensitive to the observed congestion on the channel. An and Gelenbe [1] discuss other variations, including a version of the Bayesian broadcast procedure as noted previously.

Hajek and Van Loon [12] present another adaptive scheme. Their scheme was IFT, and multiplies f (the transmission probability for backlogged packets) by 1.518 for holes, by 1.000 for successes, and by 0.559 for collisions (keeping f within some pre-established bounds as well). For the algorithm of Hajek and Van Loon [12] it is reported that the average number of backlogged packets for $\lambda = 0.32$ is approximately 5.0 (To compare our results with theirs, subtract λ from our values of N_λ^{av} since they do not count newly arrived packets in the backlog.)

We see that the pseudo-Bayesian broadcast algorithm appears to offer significantly improved performance over the Hajek and Van Loon algorithm. (To be fair, we note that their main objective was to prove that their algorithm was stable for $\lambda < e^{-1}$.)

Merakos and Kazakos [13] have made a careful study of Hajek and Van Loon's scheme and have rederived in a different way the parameters (1.518, 1.000, 0.559) mentioned earlier. They also study the effect of errors in the feedback process on the performance of the scheme. Kumar and Merakos [14] present a related scheme for updating broadcast probabilities, but which is not provably stable for all $\lambda < e^{-1}$.

The closest previous work is that of Kelly [15]. In his scheme each station maintains a variable ν, which—as in our scheme—is intended to track the value of N_t, the number of active stations. As in our scheme, an active station transmits with probability $1/\nu$. He considers the class of schemes parameterized by (a, b, c), where ν is increased by a in the case of a hole, by b in the case of a successful transmission, and by c in the case of a collision. The pseudo-Bayesian broadcast scheme is a member of this class, with parameters $(-1 + \hat{\lambda}, -1 + \hat{\lambda}, (e-2)^{-1} + \hat{\lambda})$. This particular choice of parameters was not mentioned by Kelly.

The most important question is, of course, "Is the pseudo-Bayesian broadcast scheme stable for all $\lambda < e^{-1}$?" The answer turns out to be a conditional "yes" (conditional on the manner in which the stations compute $\hat{\lambda}$).

- Kelly [15] has shown that the ratio $\kappa = N_t/\nu$ will drift towards unity whenever (a, b, c) are chosen so that

$$[(a-c)e^{-\kappa} + (b-c)\kappa e^{-\kappa} + c] \tag{24}$$

(the expected "drift" of ν) is negative for $\kappa < 1$ and positive for $\kappa > 1$, and furthermore that

$$\lambda - \kappa e^{-\kappa} < \kappa[(a-c)e^{-\kappa} + (b-c)\kappa e^{-\kappa} + c] \tag{25}$$

for $\kappa > 1$. In our case the above inequalities hold for $\lambda < e^{-1}$, *assuming that stations use* $\hat{\lambda} = e^{-1}$ *uniformly*. The drift (24) will have the correct sign when the stations *overestimate* the arrival rate λ by using the upper bound e^{-1}. This analysis was pointed out in Tsitsiklis [5], who suggested as well that using $\hat{\lambda} = e^{-1}$ may be the most robust approach to estimating λ. (The reader may also want to consult Hajek [16] for a more fully described version of a drift analysis of this sort.)

- Mikhailov [17] has presented techniques that are capable of proving the stability of this type, according to Kelly [15] and Tsitsiklis [5]. (The author has not seen the details of these methods.)

- Tsitsiklis [5] has presented a nice direct proof that the pseudo-Bayesian broadcast algorithm is indeed stable whenever $\lambda < e^{-1}$, assuming that $\lambda \leq \hat{\lambda}$, i.e., that the estimate of λ is always an *overestimate*. Tsitsiklis also shows that if $\lambda > \hat{\lambda}$, then instabilities may arise.

VI. Conclusion

We believe the proposed pseudo-Bayesian broadcast procedure will be found to be exceptionally effective in practice since it makes nearly the "best possible use" of the information available on the network in determining the broadcast probabilities to use.

Acknowledgment

I would like to thank Bard Bloom, Robert Gallager, Bruce Hajek, Pierre Humblet, Tom Leighton, Charles Leiserson, and John Tsitsiklis for helpful suggestions and discussions. I would also like to thank Michelle Lee and Brian Rogoff for helping to obtain the simulation results. Finally, the referees provided invaluable advice and pointers to the relevant literature.

REFERENCES

[1] B. T. An and E. Gelenbe, "Etude et simulation de reseaux de diffusion sous controle," *RAIRO Informatique/Comput. Sci.*, vol. 11, pp. 301–321, 1977.

[2] A Segall, "Recursive estimation from discrete-time point processes," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 422–431, July 1976.

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

[4] P. Whittle, *Optimization over Time, vol. II.* New York: Wiley, 1983, ch. 39.

[5] J. N. Tsitsiklis, "Analysis of a multiaccess control scheme," MIT LIDS Rep. LIDS-P-1534.

[6] R. G. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 124–142, Mar. 1985.

[7] J. L. Massey, Ed., *Special Issue on Random-Access Communications, IEEE Trans. Inform. Theory*, vol. IT-31, Mar. 1985.

[8] A. S. Tanenbaum, "Network protocols," *Computing Surveys*, vol. 13, pp. 453–489, Dec. 1981.

[9] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *Commun. Ass. Comput. Mach.*, vol. 19, pp. 395–404, July 1976.

[10] N. Abramson, "Packet switching with satellites," in *Proc. AFIPS Nat. Computer Conf.*, vol. 42, 1973, pp. 695–702.

[11] M. Gerla and L. Kleinrock, "Closed loop stability controls for S-ALOHA satellite communications," in *Proc. Fifth Data Communications Symp.*, Sept. 1977, pp. 2.10–2.19.

[12] B. Hajek and T. Van Loon, "Decentralized dynamic control of a multiaccess broadcast channel," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 559–569, June 1982.

[13] L. Merakos and D. Kazakos, "On retransmission control policies in multiple-access communication networks," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 109–117, Feb. 1985.

[14] P. R. S. Kumar and L. Merakos, in *Proc. 23rd IEEE Conf. Decision and Control*, Dec. 1984, pp. 1143–1148.

[15] F. P. Kelly, "Stochastic models of computer communication systems," *J. Roy. Statist. Soc.*, Series B, vol. 47, pp. 379–395, 1985.

[16] B. Hajek, "Hitting-time and occupation time bounds implied by drift analysis with applications," in *Advances Appl. Probab.*, vol. 14, pp. 502–525, 1982.

[17] V. A. Mikhailov, "Methods of random multiple access," Candidate Eng. thesis, Moscow Inst. of Physics and Technol., Moscow, USSR, 1979.