# Arrival Rate Estimation Algorithm for Single Group Slotted ALOHA Systems

Atheer Al-Amir, *Member, IEEE*, and Mustafa K. Gurcan, *Member, IEEE*

***Abstract*—In this letter, a new recursive tracking algorithm is presented that is capable of estimating the real arrival rate, $\lambda_{\text{real}}$, to the system. The estimated value of the arrival rate, $\lambda_{\text{estimated}}$, is used to dynamically adjust the control parameters of the system, hence ensuring that the operating point of the system is pushed toward the required settling point, whatever the real arrival rate to the system. This algorithm utilizes system information through the feedback channel in order to dynamically adjust the estimated value of the arrival rate and hence update the values of the control parameters.**

***Index Terms*—Multiple access, multiplexing, stabilization of Aloha, slotted Aloha.**

## I. Introduction

REFERENCE [3] identifies how a slotted Aloha system can be stabilized for a given arrival rate. Stability of the system is achieved by dynamically adjusting control parameters to determine the retransmission probability $\rho$. These retransmission probability calculations depend on the estimated number, $S_k$, of backlogged users which in turn is adjusted using control parameters. These control parameters ($u_0$, $u_1$ and $u_c$ relate, respectively, to the no-transmission, one successful transmission and the collision situation during any time slot) are chosen for a given value of the arrival rate [1]–[3]. If a system is operated with an estimated arrival rate value different to the real value, the system will not necessarily settle at the optimum offered load, $K_{\text{opt}}$ that maximizes the throughput of the system. The maximum throughput of a slotted Aloha system is equal to 0.368 packets/slot and is achieved when $K_{\text{opt}} = 1$.

In this letter, it is assumed that a large but finite number of users transmit their packets of identical length utilizing the ideal slotted Aloha system, operating in Deferred First Transition [2]. The channel propagation delay is assumed to be negligible. Each user starts packet transmission at the start of a time slot and ends before the start of the next time slot. Each user has at most one packet to transmit at any time. The number of new packets arriving to the backlog state over a single time slot is taken to be Poisson distributed with a mean rate of $\lambda_{\text{real}}$ packets/slot. A collision occurs on the channel if more than one packet is transmitted in the same time slot, by different users, otherwise the packet is assumed to have been received correctly in that slot. Users whose packets have suffered a collision will retransmit their packets at a later time with some probability known as the retransmission probability, $\rho$. The joint drift equation is denoted by $dj$ and its roots identify the offered loads at which the system settles [3].

This letter presents a new arrival rate tracking algorithm that is capable of estimating the arrival rate to the system continuously, using the latest updated estimate to select new values of the control parameters.

Manuscript received April 16, 2001. The associate editor coordinating the review of this paper and approving it for publication was Dr. J. Evans.

The authors are with the Communications and Signal Processing Research Group, Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London SW7 2BT, U.K. (e-mail: m.gurcan@ic.ac.uk).

Publisher Item Identifier S 1089-7798(01)09052-4.

## II. Regions of Tracking Algorithm

An assumed initial value of the arrival rate, $\lambda_{\text{initial}}$, is taken to be between the minimum, $\lambda_{\min} = 0$ packets/slot and maximum, $\lambda_{\max} = \exp(-1)$ packets/slot, values of $\lambda_{\text{real}}$. Using the initial arrival rate, $\lambda_{\text{initial}}$, in the control parameter selection algorithm [3], initial control parameters values are obtained using

$$u_1 = u_c \geq \lambda_{\max} = \exp(-1) \tag{1}$$
$$u_0 = \lambda_{\text{initial}} e^{K_{\text{opt}}} + u_c (1 - e^{K_{\text{opt}}}) - K_{\text{opt}}. \tag{2}$$

For these control parameters, the system will settle at a point between the lower, $G_{\text{lower}}$ and upper, $G_{\text{upper}}$, bounds of the offered load. These bounds correspond to the roots of the joint drift equation [3], taking into consideration $K_{\text{opt}} = 1$,

$$dj = dn - G\, ds \tag{3}$$

when the system is assumed to be operating at $\lambda_{\text{minimum}}$ and $\lambda_{\text{maximum}}$ respectively. In (3), $G$ is the offered load which can take values between zero and infinity and $dn$ is the drift of the real number of backlogged users with this being expressed as [4]

$$dn = \lambda_{\text{real}} - G \exp(-G). \tag{4}$$

The drift, $ds$, of the estimated number of users is [4]

$$ds = u_c + (u_0 - u_c) \exp(-G) + (u_1 - u_c)\, G \exp(-G). \tag{5}$$

These settling points are shown in Fig. 1. From this figure, two lemmas can be deduced which are of great significance to the development of the tracking algorithm.

*Lemma 1:* If the Aloha system operates at an offered load value, $G_{\text{operate}}$, using control parameters selected utilizing $\lambda_{\text{estimate}}$, then the real arrival rate, $\lambda_{\text{real}}$, to the system must satisfy

$$0 < \lambda_{\text{real}} < \lambda_{\text{estimate}} \text{ for } G_{\text{lower}} < G_{\text{operate}} < K_{\text{opt}}. \tag{6}$$

*Proof:* Let $dn_{\text{estimate}} = \lambda_{\text{estimate}} - G \exp(-G)$ and $dn_{\text{real}} = \lambda_{\text{real}} - G \exp(-G)$. Also let $dj_{\text{estimate}} = dn_{\text{estimate}} - G\, ds$ and $dj_{\text{real}} = dn_{\text{real}} - G\, ds$. For the control

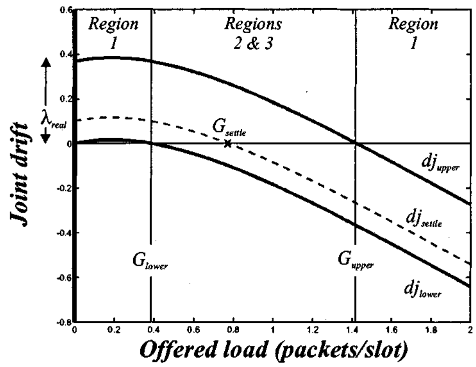


Fig. 1. Drift curves displaying the corresponding roots of the joint drift equation for minimum and maximum arrival rates to the system and the three general regions of the tracking algorithm.

parameters selected using $\lambda_{\text{estimate}}$, the joint drift equation satisfies

$$dj_{\text{estimate}} = 0 \quad \text{at} \quad G = K_{\text{opt}}. \tag{7}$$

When $\lambda_{\text{estimate}} > \lambda_{\text{real}}$, the value of the drift, $dn_{\text{real}}$ of the real number, $N_k$, of backlogged users will satisfy the relationship

$$dn_{\text{real}} < dn_{\text{estimate}} \quad \text{at} \quad G = K_{\text{opt}}. \tag{8}$$

From (3), (5), and (8), the following relationship for $dj_{\text{real}}$ can be established:

$$dj_{\text{real}} < 0 \quad \text{at} \quad G = K_{\text{opt}} \tag{9}$$

since $\lambda_{\text{real}} > 0$ and $dj_{\text{real}}(G = 0) = \lambda_{\text{real}}$

$$dj_{\text{real}} > 0 \quad \text{at} \quad G = 0 \tag{10}$$

Equations (9) and (10) imply that the root of $dj_{\text{real}}$ lies between $G_{\text{lower}}$ and $K_{\text{opt}}$ when $\lambda_{\text{estimate}} > \lambda_{\text{real}}$.

*Lemma 2:* If a system operates at an offered load value, $G_{\text{operate}}$, using control parameters selected utilising $\lambda_{\text{estimate}}$, then the real arrival rate, $\lambda_{\text{real}}$, to the system must satisfy

$$\lambda_{\text{estimate}} < \lambda_{\text{real}} < \lambda_{\text{maximum}} \quad \text{for} \quad K_{\text{opt}} < G_{\text{operate}} < G_{\text{upper}}. \tag{11}$$

*Proof:* Can be proven in a similar way to Lemma 1.

The proposed tracking algorithm will perform different functions in different regions depending on the load offered to the system, as well as the number of users in the backlog. In the proposed arrival rate tracking algorithm, three regions of operation exist, as shown in Fig. 1. Transition from regions 1 to 2 is determined by the value of $G$, whereas the transition from regions 2 to 3 is determined by an arbitrary value of $S_k$ that differentiates between the backlogged users being in a high or low population region and is denoted as $S_{\text{threshold}}$.

*Region 1:* Region 1 corresponds to $G > G_{\text{upper}}$ when $N_k > S_k$ and $S_k > S_{\text{threshold}}$ (when system is initiated in the heavy traffic region, which is assumed in this paper). Users in the system will measure the offered load of the system using feedback information broadcast from the base station. Once $G_{\text{upper}}$ is reached, region 1 is terminated and region 2 is initiated.

*Region 2:* Region 2 is defined by $N_k \approx S_k$ and $S_k > S_{\text{threshold}}$ when $G_{\text{lower}} < G_{\text{settle}} < G_{\text{upper}}$. In region 2, the algorithm will update the arrival rate recursively according to the offered load at which the system is operating. Region 2 will be terminated when either $G_{\text{settle}} > G_{\text{upper}}$ or $S_k$ reaches $S_{\text{threshold}}$.

*Region 3:* The system is in region 3 if $S_k < S_{\text{threshold}}$ and $G_{\text{settle}} < G_{\text{upper}}$. In this regions the joint drift equation will satisfy $dj = 0$. In region 3, the estimated number, $S_k$, of backlogged users will tend toward the minimum allowable value, $S_{\min}$. Therefore the retransmission probability will converge toward unity ($\rho = K_{\text{opt}}/S_k$). This implies that $ds$ will be equal to zero. Since $dj$ is also equal to zero in this region, $dn$ must be equal to zero in order to satisfy the joint drift equation $dj = dn - G\,ds = 0$. The fact that $dn = 0$ implies that the arrival rate is equal to the departure rate. Region 3 is terminated if $S_k$ increases beyond $S_{\text{threshold}}$.

## III. Steps of Proposed Algorithm

The steps of the proposed tracking algorithm are as:

Step 1: Initialize the system with initial values of the control parameters in accordance with (1) and (2) for $\lambda_{\text{initial}} = \exp(-1)/2$. Estimate the offered load of the system using the maximum likelihood estimation method [5].

Step 2: Once the system starts to operate in the region specified by $G_{\text{lower}} < G < G_{\text{upper}}$, the system moves to region 2 of operation (if $S_k > S_{\text{threshold}}$).

Step 3: In region 2, an initial estimate of the arrival rate to the system is carried out using the joint drift equation, $dj(G) = 0$ [3], from

$$\begin{aligned}\lambda_{\text{estimate}} =& G\left[u_c + (u_0 - u_c)\exp(-G)\right.\\ &\left.+ G\,(u_1 - u_c)\exp(-G)\right]\\ &+ G\exp(-G)\end{aligned} \tag{12}$$

Step 4: Find the error signal indicator, $\varepsilon$, based on Lemmas 1 and 2, from

$$\varepsilon = \begin{cases} \text{sign}|\text{G} - \text{K}_{\text{opt}}|, & \text{for } G \neq K_{\text{opt}} \\ 0, & \text{for } G = K_{\text{opt}}. \end{cases} \tag{13}$$

Step 5: Update $\lambda_{\text{estimate}}$ using

$$\lambda_{\text{estimate,current}} = \lambda_{\text{estimate,previous}} + \varepsilon\Delta\lambda \tag{14}$$

where $\Delta\lambda$ represents a constant scaling factor of the correction applied.

Step 6: Calculate a new value of $u_0$ (keeping $u_1$ and $u_c$ fixed at the initial values) from

$$u_0 = \lambda_{\text{estimate}}\, e^{K_{\text{opt}}} + u_c\,(1 - e^{K_{\text{opt}}}) - K_{\text{opt}}. \tag{15}$$

Step 7: Repeat Steps 4–6 until region 3 is reached.

Step 8: In region 3 the arrival rate to the system is estimated using (taking into consideration that the number of users is finite and small and assuming $(S_k/t{+}1) \to 0$ as $t \to \infty$ [1])

$$\lambda_{\text{estimate}}(t+1) = \frac{\sum_{k=0}^{t} \varepsilon(k)}{t+1} \tag{16}$$

where

$$\varepsilon(t) = \begin{cases} 1 & \text{if a success was observed during time slot } t \\ 0 & \text{otherwise} \end{cases}$$

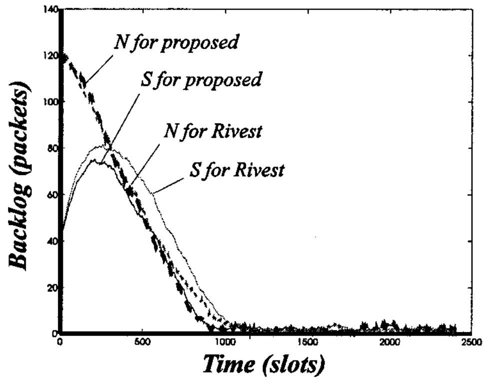


Fig. 2. Tracking performance of $N_k$ by $S_k$ using the following parameters $N_{\text{initial}} = 120$ users, $S_{\text{initial}} = 40$ users, $\lambda_{\text{real,initial}} = 0.2$ packets/slot$(u_0, u_1, u_c) = (-1.2, 0.4, 0.4)$ for proposed algorithm, with $\lambda_{\text{initial}} = 0.18$ $(u_0, u_1, u_c) = (-0.33, 0.4, 0.4)$ for Rivest's algorithm, with $\lambda_{\text{initial}} = 0.5$.

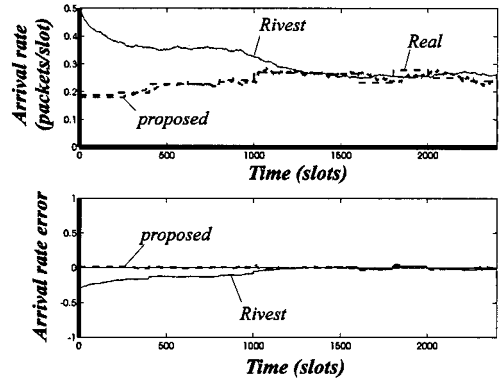


Fig. 3. Tracking performance of $\lambda_{\text{real}}$ by $\lambda_{\text{estimate}}$ using following parameters $(u_0, u_1, u_c) = (-1.2, 0.4, 0.4)$ for proposed algorithm, with $\lambda_{\text{initial}} = 0.18$ $(u_0, u_1, u_c) = (-0.33, 0.4, 0.4)$ for Rivest's algorithm, with $\lambda_{\text{initial}} = 0.5$.

Step 9: Calculate a new value of $u_0$ (keeping $u_1$ and $u_c$ fixed at the initial values) from

$$u_0 = u_c\,(1 - e^{K_{\text{opt}}}) \tag{17}$$

Step 10: Repeat Steps 8 and 9. If number of backlogged users increases, repeat from Step 3.

The size of the correction $\Delta\lambda$ determines the speed at which the estimated arrival rate converges toward the real arrival rate and can be chosen to be either of fixed or variable size.$\Delta\lambda$

## IV. Simulation Results

Simulation results are obtained by updating $N_k$ and $S_k$ according to the changes occurring in the system for a given set of initial input parameters ($\lambda_{\text{initial}}$, real number of backlogged users $N_0$, total number of users (real) in the system $N_{\text{total}}$, estimated number of backlogged users $S_0$ and control parameters $u_0$, $u_1$ and $u_c$). For each time iteration, the number of users that have new packets to transmit are computed and are added to the number of users in the backlog state that are waiting to retransmit their packets. Users in the backlog state transmit their packets according to their retransmission probability. $N_k$ $S_k$ At the end of each iteration, the values of the output parameters ($N$, $S$, $\lambda_{\text{estimate}}$) are processed to obtain the different plots presented.

The control parameters are adjusted using the arrival rate estimation algorithms presented by Rivest [2] and the tracking algorithm proposed in this letter, are shown in Figs. 2 and 3. Fig. 2 shows the tracking performance of $N_k$ by $S_k$ when the control parameters are updated using the two algorithms. From this figure it can be seen that $S_k$ tends to overestimate $N_k$ after some time when control parameters are updated using the algorithm proposed in [2]. This indicates that the load offered to the system is not at the optimum value, hence it cannot utilize the Aloha channel efficiently. For the proposed algorithm, $S_k$ closely follows $N_k$ once it approaches it, thus causing the offered load to settle at the optimum offered value. Fig. 3 shows the tracking performance of $\lambda_{\text{real}}$ based on its estimation using both algorithms. For the algorithm presented in [2] it can be seen that the large initial value of the arrival rate makes the time taken for the estimated arrival rate to reach the real arrival rate much longer. For the proposed algorithm, the precise tracking of the real arrival rate causes the system to settle at the optimum operating point, thereby achieving maximum throughput. On the other hand, the algorithm presented in [2] is not suited in estimating the arrival rate over the entire load range.

## V. Conclusions

This letter proposes a new recursive tracking algorithm for the real arrival rate for Slotted Aloha systems. Each estimate of the arrival rate is used to update the control parameters of the system in order to ensure that the operating point of the system is pushed toward the optimum settling point. Simulation results verify the effectiveness of the algorithm.

In real time systems, feedback information from the base station is made available to users in the system at the end of each time slot, assuming that the channel propagation delay is negligible. This is possible by keeping the packet slot duration smaller than the time slot to overcome these propagation delay problems occurring due to users being at different distance away from the base station.

## References

[1] S. C. A. Thomopoulos, "A simple and versatile decentralised control for slotted Aloha, reservation Aloha and local area networks," *IEEE Trans. Commun.*, vol. 36, pp. 662–674, June 1988.

[2] R. L. Rivest, "Network control by Bayesian broadcast," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 323–328, May 1987.

[3] M. K. Gurcan, A. Al-Amir, and M. Hijazi, "A sequential method for control parameters in stabilised ALOHA systems," *IEEE Commun. Lett.*, vol. 3, no. 5, pp. 148–150, May 1999.

[4] L. P. Clare, "Control procedures for slotted ALOHA systems that achieve stability," in *Proc. ACM SIGCOMM '86 Symp. Commun., Arch. & Protocols*, Stowe, VT, Aug. 1986, pp. 302–309.

[5] Y.-W. Leung, "Mean power consumption of artificial power capture in wireless networks," *IEEE Trans. Commun.*, vol. 45, pp. 957–964, Aug. 1997.