Maximum Stable Throughput of FS-ALOHA under Delay Constraints

B. Van Houdt* and C. Blondia

University of Antwerp (UA), Dept. Mathematics and Computer Science,
Middelheimlaan 1, B-2020 Antwerp, Belgium.

This paper studies the maximum stable throughput of FS-ALOHA, a random access algorithm for dynamic bandwidth reservation in access networks, subject to delay constraints. Requests that are not transmitted successfully before the maximum delay t_{max} expires are dropped. We state that the algorithm is stable for a certain input rate λ, if the dropping probability is below a predefined tolerance ϵ, e.g., $\epsilon = 10^{-9}$. The matrix analytic method (MAM) is used to determine the maximum input rate for which the system is stable. Numerical examples for different parameter setting provide a useful insight on how to optimize the maximum stable throughput as a function of t_{max}, the maximum delay, and ϵ, the drop tolerance.

1 INTRODUCTION

Future wireless LANs are expected to support a large increment of customer demands for mobile services and applications. Therefore, efficient network and service architectures must be devised to comply to these demands with adequate Quality of Service (QoS). One of the trends towards designing such LANs is to allocate the uplink bandwidth, that is, from the end users towards the network, in a dynamic way. This calls for an efficient mechanism allowing mobile stations (MSs) to declare their current bandwidth needs to the base station (BS). An often proposed solution, e.g., [8, 6, 9, 14], both in wired and wireless networks, is to combine the technique of piggybacking with a contention channel. The performance of the contention scheme used determines the reaction speed of the system on changing traffic conditions; therefore, it is an important factor in the QoS provisioning.

Although Slotted ALOHA is easy to implement in such an environment, it is unable to guarantee good delay bounds [3, 5]. Fifo-by-Sets ALOHA (FS-ALOHA), on the other hand, maintains the simplicity of Slotted ALOHA and was specifically designed to operate in a wireless LAN with QoS provisioning. Its superiority on Slotted ALOHA was demonstrated by means of simulation and analytical methods in [3, 5] using Poisson arrivals. In [13, 12] we studied the performance of FS-ALOHA using Markovian arrivals and investigated the influence of errors and capture events.

*B. Van Houdt is a postdoctoral fellow of the FWO Flanders

In this paper we determine the maximum stable throughput of FS-ALOHA subject to delay constraints. In the past, the maximum stable throughput λ^1_{max} was defined as the maximum input rate for which all packets have a finite delay with probability one. A substantial amount of work has gone into determining λ^1_{max} for a wide variety of random access algorithms (RAAs) [1, 11]. However, a communication network guaranteeing Quality-of-Service (QoS) wants (nearly) all packets to have a delay below a certain delay bound t_{max}, thus, not just a finite delay. Therefore, we redefine the maximum stable throughput λ_{max} as follows. Packets that are not transmitted successfully before the maximum delay t_{max} expires are dropped. We state that a RAA is stable for a certain input rate λ, if the dropping probability is below a predefined tolerance ϵ, e.g., $\epsilon = 10^{-9}$. Hence, λ_{max} is a function of t_{max} and ϵ. We are not aware of any stability results that take such a delay constraint into account. Meanwhile, two more advanced versions of FS-ALOHA have been defined: FS-ALOHA++ [4, 13] and E-FS-ALOHA [10]. These more advanced algorithms are beyond the scope of this paper.

2 FS-ALOHA: A REVIEW

In this section the operation of FS-ALOHA, and the environment in which it operates, are described in some detail, additional comments and discussions can be found in [3, 4, 5]. Consider a cellular network with a centralized architecture, i.e., the area covered by the wireless access network is subdivided into a set of geographically distinct cells each with a diameter of approximately 100m. Each cell contains a base station (BS) serving a finite set of mobile stations (MSs). This BS is connected to a router, which supports mobility, realizing seamless access to the wired network. Two logically distinct communication channels (uplink and downlink) are used to support the information exchange between the BS and the MSs. Packets arriving at the BS are broadcasted downlink, while upstream packets must share the radio medium using a MAC protocol. The BS controls the access to the shared radio channel (uplink). The access technique is Time Division Multiple Access (TDMA) combined with Frequency Division Duplex (FDD).

Traffic on both the uplink and downlink channel is grouped into fixed length frames, with a length of L slots, to reduce the battery consumption [14]. The uplink and downlink frames are synchronized in time, i.e., the header of a downlink frame is immediately followed by the start of an uplink frame (after a negligible round trip time that is captured within the guard times). Each uplink frame consists of a fixed length contentionless and a fixed length contention period. An MS is allowed to transmit in the contentionless period after receiving a permit from the BS. The BS distributes these permits among the MSs based on the requests it receives from the MSs and the existing QoS agreements between the end users and the network. Within these requests, MSs declare their current bandwidth needs to the BS, e.g., by indicating how many packets they have ready for transmission. Requests are transmitted using the contention channel, unless the MS can piggyback the request to a data packet for which a permit was already obtained, thereby reducing the load on the contention channel and avoiding the delay caused by the contention channel.

A request is generally much smaller than a data packet; therefore, slots part of the contention period can be subdivided into k minislots (realistic values for k in a wireless medium are 1 to 3, in a wired medium higher values for k are possible). Each downlink frame starts with a frame header in which, among other things, the required feedback on the contention period of the previous uplink frame is given. This feedback informs the MSs participating in the contention period whether there was a collision or whether the request was successfully received.

FS-ALOHA operates on the slots that are part of the fixed length contention period. Define T as the number of minislots part of the contention period of a frame. From hereon we refer to minislots as slots. In slotted ALOHA systems, an MS with a pending request will randomly choose one out of the T slots to send its request in the hope that no other MS with a pending request will choose the same slot. If an MS is unsuccessful it will retransmit in the next frame. It is important to note that with slotted ALOHA, new requests are allowed to transmit on the contention channel immediately after being generated; hence, they are not blocked. FS-ALOHA on the contrary, divides the T slots of the contention period into two disjoint sets of S and N slots such that $T = S + N$. The operation of FS-ALOHA is as follows:

(A) Newly arrived requests are transmitted, for the first time, by randomly choosing one out of the S slots; this is the first set of S slots after the request was generated. If some of the transmissions taking place in the S slots of a frame are unsuccessful, because multiple MSs transmitted in the same slot, the unsuccessful requests are grouped into a Transmission Set (TS), which joins the back of the queue of TSs waiting to be served.

(B) The other N slots are used to serve the queue of backlogged TSs on a FIFO basis. Backlogged TSs are served, one at a time, using slotted ALOHA, that is, all the requests part of the TS select one out of the N slots and are transmitted in this slot. The requests that were transmitted successfully leave the TS, the others retransmit in the N slots of the next frame using the same procedure. The service of a TS lasts until all the requests part of the TS have been successfully transmitted, in which case the service of the next TS, if there is another TS in the queue, starts service in the N slots of the next frame.

Hence, two parameters play an important role in FS-ALOHA: (i) The number of $S \geq 1$ slots in a frame. These slots are used by the MSs to transmit newly arrived requests; S determines the TS generation rate. (ii) The number of $N \geq 2$ slots in a frame. These slots are allocated to the service of the TSs in the distributed FIFO queue.

Notice, two requests that were generated in different frames can never be part of the same TS. Thus, it is said that the grouping of requests in Transmission Sets is based on a time period corresponding to the frame length. Therefore, FS-ALOHA can be regarded as a Group Random Access Algorithm that uses Slotted ALOHA as its conflict resolution algorithm (CRA). More details on the operation of FS-ALOHA can be found in [3, 4, 5].

Thus, according to the description above a request is retransmitted until successful (this is also the case in all prior studies of FS-ALOHA). In this paper, we add a delay constraint that demands a request to be successful within t_{max} frames. Otherwise, the request is dropped. Notice, all requests part of the same TS, have the same age. Thus, if the age of a TS exceeds the maximum delay, all requests part of the TS (that were not successful so far) are dropped. This adaptation is easy to implement, because the identifier of a TS can be used as a time stamp (see [5]). Also, the scheduler, located in the BS, always announces the identifier of the TS (in the feedback field of the downlink frame) that is served in the N slots of the next uplink frame. Hence, if the difference between the current time and the TS' identifier becomes larger than t_{max}, it simply announces the next TS.

3 ANALYTICAL MODEL

In this section an exact analytical model is developed, allowing the computation of the drop probability p_{drop} associated to the request packets under the following conditions: (i) We assume a discrete time batch Markovian request arrival process (D-BMAP, [2]) with a mean rate of λ arrivals per frame. The time unit of the D-BMAP arrival process is one frame time. (ii) If there are no Transmission Sets in the distributed FIFO queue nor in service, the total $T = S + N$ slot is available to the new arrivals. (iii) The Bit Error Rate (BER) is assumed to be zero. These assumptions are identical to [5], except that we assume D-BMAP arrivals instead of Poisson arrivals.

A D-BMAP is characterized by a set of $l \times l$ matrices D_i, for $i \geq 0$, for some integer $l > 0$. The $(j_1, j_2)^{th}$ entry of the matrix D_i represents the probability that i new requests are generated within a frame, while a transition from state j_1 to j_2 occurs. Let q_m be maximal index i for which $D_i \neq 0$. For D-BMAPs that do not posses such an index i or for D-BMAPs for which this index i is very large, we choose q_m such that the sum of the entries of the matrices D_i, for $i > q_m$, is negligible (i.e., $\leq 10^{-14}$). In this way, the impact on the accuracy of the results should be minimized. If the D-BMAP is in state j at the start of the n^{th} frame then i new request transmit in frame n with probability $(D_i e)_j$, where e is a $l \times 1$ vector with all its entries equal to one.

3.1 Defining the Markov Chain

Consider a Markov chain (MC) with a finite number of states labeled $1, 2, \ldots, l + t_{max} l(q_m - 1)$. The set of states $\{1, \ldots, l\}$ is referred to as level zero of the MC, whereas the set of states $\{(i-1)b_s + l + 1, \ldots, ib_s + l\}$ is referred to as level i of the MC for $0 < i \leq t_{max}$, and where $b_s = l(q_m - 1)$. The states of level i, with $0 < i \leq t_{max}$, are labeled as (q, j), where $2 \leq q \leq q_m$ and $1 \leq j \leq l$. The MC is in state j, $1 \leq j \leq l$, of level zero at time instant n, if there is no TS served in the n^{th} frame (i.e., all $T = S + N$ slots are used for new arrivals) and the state of the D-BMAP at time n is j. If there is a TS in service in the n^{th} frame, the MC is in some state (q, j) of level i, where i indicates how many frames ago the Transmission Set in service was generated, $q \geq 2$ denotes the number of requests left in the Transmission Set that is in service in frame n, and j denotes the state of the D-BMAP associated with the start of the frame that follows the frame in which the Transmission Set in service was generated (that is, frame $n - i + 1$).

3.2 The Transition Matrix P

In order to facilitate the description of the transition matrix P corresponding to this MC, we introduce some notations. Define $p_x(q, q')$, for $q \geq q'$, as the probability that in a set of q requests, $q - q'$ request are successful when a set of x slots is used to transmit the q request packets[†]. We are particularly interested in $p_S(q, q')$, $p_N(q, q')$ and $p_{S+N}(q, q')$.

[†]This corresponds to the following combinatorial problem: Provided that we, randomly, distribute q balls among a set of x urns, what is that probability that we have exactly $q - q'$ urns holding a single

Von Mises [15] has shown, in 1939, that

$$p_x(q,q') = \sum_{v=q-q'}^{\min(q,x)} (-1)^{v+q-q'} C^v_{q-q'} C^x_v \frac{q!}{(q-v)!} \frac{(x-v)^{q-v}}{x^q}, \tag{1}$$

where C^r_s denotes the number of different ways to choose s from r different items. Eq. (1) is numerically stable for the parameter ranges of interest (see Section 4). It is also possible to calculate the $p_x(q,q')$ values recursively using the $p_{x-1}(q,q')$ values, thus, higher parameter values do not cause any problems.

Next, denote P_N as an $(q_m-1)\times(q_m-1)$ matrix whose $(i,j)^{th}$ element equals $p_N(i+1,j+1)$[‡]. Let P_0 be a $(q_m-1)\times 1$ vector whose i^{th} component equals $p_N(i+1,0)$. The $l\times l$ matrices F_S, F_{S+N}, $E^k_S, 2\le k\le q_m$, and $E^k_{S+N}, 2\le k\le q_m$, are defined as

$$F_S = \textstyle\sum_{i\ge 0} D_i\ p_S(i,0), \qquad F_{S+N} = \textstyle\sum_{i\ge 0} D_i\ p_{S+N}(i,0),$$

$$E^k_S = \textstyle\sum_{i\ge k} D_i\ p_S(i,k), \qquad E^k_{S+N} = \textstyle\sum_{i\ge k} D_i\ p_{S+N}(i,k),$$

where the D-BMAP arrival process is characterized by the matrices D_i. Notice that $(E^k_x)_{j,j'}$, with $x=S$ or $S+N$, represents the probability that a new TS with k requests is generated in a frame where x slots are used for the new arrivals, and the D-BMAP governing the new arrivals makes a transition from state j to j'. F_x on the other hand holds the probabilities that no new TS is generated in a frame where x slots are used for new arrivals.

Suppose that the MC is in state (q,j) of level $i<t_{max}$ at time n. Thus, a TS that was generated in frame $n-i$, is served in frame n. q requests are still part of this TS and the state of the D-BMAP at time $n-i+1$ is j. We distinguish two scenarios:

1. With probability $(P_N)_{q-1,q'-1}$, $q'>1$ of the q request in the TS are unsuccessful in the N slots of frame n. Thus, the MC is in state (q',j) of level $i+1\le t_{max}$ at time $n+1$.

2. With probability $(P_0)_{q-1}$, all the requests part of the TS are successful. Thus, a new TS can start service in the N slots of frame $n+1$.

   2a. With probability $(F_S^{i-i'}E_S^{q'})_{j,j'}$, for $i\ge i'>0$, there is no TS generated in the S slots of frame $n-i+1,\ldots,n-i'$, while frame $n-i'+1$ generates a new TS holding q' requests and the state of the D-BMAP at time $n-i'+2$ equals j'. The MC would be in state (q',j') of level i', with $0<i'\le i$, at time $n+1$. Notice, this probability depends only upon the difference $i-i'$ and not on the value of i or i'.

   2b. With probability $(F^i_S)_{j,j'}$, there is no TS generated in frame $n-i+1,\ldots,n$ and the D-BMAP is in state j' at time $n+1$, therefore, all $T=S+N$ slots of frame $n+1$ are used for new arrivals. As a result, the MC would be in state j' of level 0 at time $n+1$.

ball.

[‡]Notice, for $j>i$, $p_N(i+1,j+1)$ equals zero.

As a result of the above discussion, we find that the transition matrix P can be written as

$$P = \begin{bmatrix} B_1 & B_0 & 0 & \dots & 0 \\ B_2 & A_1 & A_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{t_{max}} & A_{t_{max}-1} & A_{t_{max}-2} & \dots & A_0 \\ E & C_{t_{max}} & C_{t_{max}-1} & \dots & C_1 \end{bmatrix},$$

where A_k and C_k are $l(q_m - 1) \times l(q_m - 1)$ matrices, $B_k, k > 1$, and E are $l(q_m - 1) \times l$ matrices, B_1 is an $l \times l$ matrix and B_0 is an $l \times l(q_m - 1)$ matrix. Moreover, $A_0 = P_N \otimes I_l$, where $\otimes$ represents the Kronecker product between matrices and I_l is an $l \times l$ unity matrix (see scenario 1). A_k, for $k > 0$, represents the transitions from level i to $i-(k-1)$; hence, $A_k = P_0 \otimes F_S^{k-1}[E_S^2, E_S^3, \dots, E_S^{q_m}]$ (see scenario 2a). While, B_k, for $k > 1$, equals $P_0 \otimes F_S^{k-1}$ (see scenario 2b).

Let us now discuss E, C_k, B_0 and B_1. Suppose that the MC is in state (q, j) of level t_{max} at time n. Assume that q' of the q requests are unsuccessful in the N slots of frame n. The age of the TS at time n was t_{max}, therefore, the q' remaining requests part of the TS are dropped. Thus, even if we have unsuccessful transmissions in the N slots of frame n, a new TS can start service in frame $n+1$. Or stated differently, as far as the transition probabilities are concerned, it is as if P_0 equals e, a $(q_m - 1) \times 1$ vector filled with ones. Hence, $C_k = e \otimes F_S^{k-1}[E_S^2, E_S^3, \dots, E_S^{q_m}]$ and $E = e \otimes F_S^{t_{max}}$. Finally, one easily finds that $B_1 = F_{S+N}$ and $B_0 = [E_{S+N}^2, E_{S+N}^3, \dots, E_{S+N}^{q_m}]$.

3.3 The Drop Probability p_{drop}

Define $\pi_i^n(q, j), i > 0$, and $\pi_0^n(j)$, as the probability that at time n the system is in state (q, j) of level i and state j of level 0, respectively. Let $\pi_0(j) = \lim_{n\to\infty} \pi_0^n(j)$, and $\pi_i(q, j) = \lim_{n\to\infty} \pi_i^n(q, j)$. Define the $1 \times l$ vector $\pi_0 = (\pi_0(1), \dots, \pi_0(l))$ and the $1 \times l(q_m - 1)$ vectors $\pi_i = (\pi_i(2, 1), \dots, \pi_i(2, l), \pi_i(3, 1), \dots, \pi_i(3, l), \pi_i(4, 1), \dots, \pi_i(q_m, l))$, for $0 < i \leq t_{max}$. Let the matrix Q be $P - I$, where I is the unity matrix with an appropriate dimension. The matrix Q can be seen as an infinitesimal generator of a continuous time Markov chain and Q can be written in a lower block-Hessenberg form by relabeling the states appropriately. Therefore, we can calculate the unique stochastic vector π for which $\pi Q = 0$, i.e., $\pi P = \pi$, in an efficient manner using the Latouche-Jacobs-Gaver algorithm [7], which has a time complexity of $O(l^3 q_m^3 t_{max}^2)$ and a space complexity of $O(l^2 q_m^2 t_{max})$.

The probability that a request gets dropped p_{drop} is equal to the expected number of requests that are dropped in an arbitrary frame divided by the arrival rate λ. Hence,

$$p_{drop} = \frac{1}{\lambda} \sum_{q=2}^{q_m} q(1 - (1 - 1/N)^{q-1}) \sum_{j=1}^{l} \pi_{t_{max}}(q, j),$$

where $q(1 - 1/N)^{q-1}$ equals the expected number of successful transmissions in N slots provided that q requests attempt transmission.

4 NUMERICAL RESULTS

In this paper we restrict ourselves to Poisson and Markov Modulated Poisson Arrivals, other arrival processes can be analyzed using the framework developed in this paper. Currently, it is hard to state whether these arrival processes are adequate to model the actual input traffic on a bandwidth reservation channel. For instance, the load of such a channel depends on the amount of piggybacking that can be done, on the way the scheduler works, on the characteristics of the applications and much more.

4.1 Poisson Arrivals

Given the couple (t_{max}, ϵ) we determine the maximum stable throughput for the parameters (S, N) as follows. First, we set $\lambda_{min} = 0$ and $\lambda_{max} = S + N$. Then, we calculate the drop probability p_{drop} for Poisson arrivals with a mean rate equal to $\lambda_m = (\lambda_{min} + \lambda_{max})/2$, thus, the probability that k new requests are generated during a frame equals $\lambda_m^k e^{-\lambda_m}/k!$. If the drop probability p_{drop} is larger than ϵ we set $\lambda_{max} = \lambda_m$, otherwise we set $\lambda_{min} = \lambda_m$. This procedure is repeated until $\lambda_{max} - \lambda_{min} < 10^{-8}$. The maximum stable throughput is then found as $\lambda_m/(S + N)$.

Figure 1 presents the maximum stable throughput (MST) of FS-ALOHA for $t_{max} = 10, 20$ and 50 and $\epsilon = 10^{-9}$ and 10^{-6}. The number of S slots varies from 1 to 10, while the number of N slots varies from 2 to 10 (except for $t_{max} = 10$ where S and N have somewhat larger ranges). For each value of S we plotted a '*' in Figure 1 to indicate where the surface reaches a maximum for this value of S. A '$\diamond$' is plotted to indicate the maximum of the entire surface, along with the (S, N) value where this maximum is reached.

A first, obvious, result is that the smaller ϵ, i.e., the fewer packets that are allowed to be dropped, the lower the MST. Also, the larger the maximum tolerable delay t_{max} the higher the MST. A second observation is that one needs larger values for (S, N) when decreasing t_{max}. For instance, for $t_{max} = 10$ and $\epsilon = 10^{-9}$ one achieves the best MST with $(S, N) = (13, 15)$, whereas for $t_{max} = 50$ one finds that $(S, N) = (2, 4)$ is the best choice. Notice, if we have $T = 24$ slots available for the contention channel, than one could implement 4 instances of FS-ALOHA, each with $(S, N) = (2, 4)$. This would be a good choice if the maximum tolerable delay is large, e.g., 50, however, if t_{max} is small, it would be better to use a single instance with $(S, N) = (14, 16)$. A similar result is found with respect to ϵ: decreasing ϵ, i.e., demanding a lower loss, results in higher optimal (S, N) values. This observation was confirmed by setting $\epsilon = 10^{-4}$ (for brevity, these plots are not included in the paper).

In conclusion, the more sensitive the requests are with respect to delays and losses, the larger the optimal (S, N) values (as far as the MST is concerned). It is possible to get some idea on where the optimal N value is situated for $S = 1$ using the following rule. Suppose that we have a TS that consists of 2 requests. Then, with probability $N^{-t_{max}}$ they choose the same N slot in t_{max} consecutive frames. Thus, in order to get a system with a drop probability below ϵ, for $S = 1$, we need $(1 - e^{-\lambda} - \lambda e^{-\lambda})N^{-t_{max}} < \epsilon$ or roughly speaking $N > \epsilon^{-1/t_{max}}$, e.g., for $t_{max} = 10$ and $\epsilon = 10^{-9}$ we find $N > 7.9433$. If we study the optimal N as a function of the number of S slots, we find that the best N value can be roughly approximated as $S/3 + \epsilon^{-1/t_{max}}$.

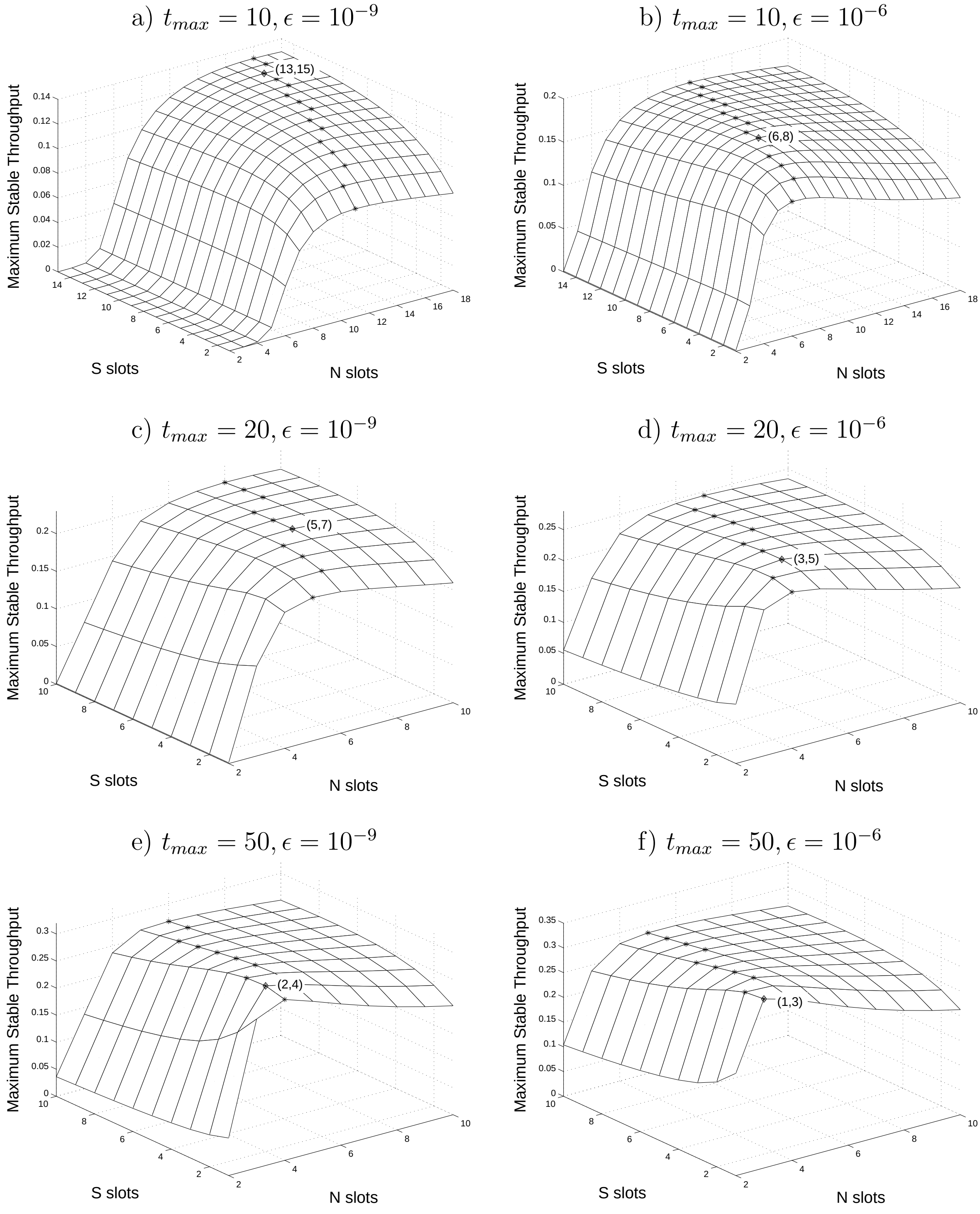

Figure 1: *Maximum stable throughput of FS-ALOHA with Poisson arrivals: a)* $t_{max} = 10, \epsilon = 10^{-9}$*, b)* $t_{max} = 10, \epsilon = 10^{-6}$*, c)* $t_{max} = 20, \epsilon = 10^{-9}$*, d)* $t_{max} = 20, \epsilon = 10^{-6}$*, e)* $t_{max} = 50, \epsilon = 10^{-9}$*, f)* $t_{max} = 50, \epsilon = 10^{-6}$

Looking at Figure 1, one tends to believe that as t_{max} goes to infinity, setting $(S, N) = (1, 2)$ is the optimal choice. This idea was confirmed by plotting the MST for $t_{max} = \infty$

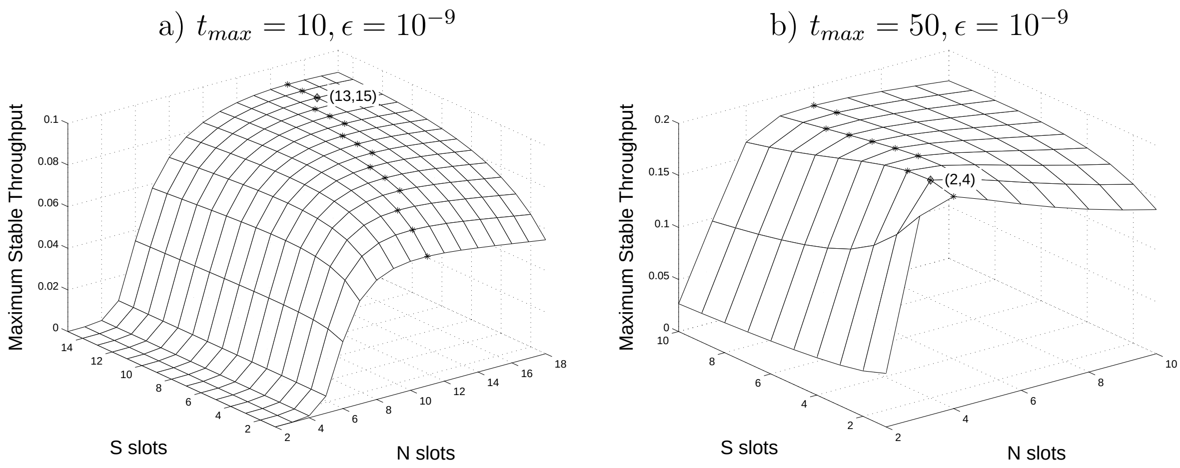

Figure 2: *Maximum stable throughput of FS-ALOHA with MMPP arrivals ($\alpha = 50, \lambda_i = i\lambda/2$): a) $t_{max} = 10, \epsilon = 10^{-9}$, b) $t_{max} = 50, \epsilon = 10^{-9}$*

(this figure is not included due to space lmimitations). The $t_{max} = \infty$ results were found by studying the ergodicity condition of the GI/M/1 type Markov chain developed in [12, Section 4].

4.2 Markov Modulated Poisson Arrivals

In this section we consider a 3-state Markov Modulated Poisson process (MMPP). Let λ_i be the arrival rate when the process is in state i, that is, the probability that k arrivals occur in a frame when the MMPP is in state i, for $i = 1, \ldots, 3$, at the start of the frame equals $\lambda_i^k e^{-\lambda_i}/k!$. Transitions (occurring at the end of each frame) between the 3 states occur according to the following transition matrix D:

$$D = \begin{bmatrix} 1 - 1/\alpha & 1/\alpha & 0 \\ 1/\alpha & 1 - 2/\alpha & 1/\alpha \\ 0 & 1/\alpha & 1 - 1/\alpha \end{bmatrix}. \tag{2}$$

Notice, the smaller α the stronger the correlation. The matrices D_i characterizing the D-BMAP are then found as

$$D_i = \begin{bmatrix} \lambda_1^i e^{-\lambda_1}/i! & 0 & 0 \\ 0 & \lambda_2^i e^{-\lambda_2}/i! & 0 \\ 0 & 0 & \lambda_3^i e^{-\lambda_3}/i! \end{bmatrix} D. \tag{3}$$

Figure 2 presents the MST for $\alpha = 50$, $\lambda_i = i\lambda/2$, for $i = 1, \ldots, 3$, $\epsilon = 10^{-9}$ and $t_{max} = 10$ and 50. The MST for this arrival process equals the maximum λ such that $p_{drop} < \epsilon$. Although the MST is, due to the more bursty nature of the MMPP, considerably lower compared to Poisson traffic (see Figures 1 and 2), the shape of the surface is, however, very similar. This result supports the idea that the optimal settings for (S, N) are not very sensitive to the characteristics of the arrival process. This is an important result when deploying FS-ALOHA in a real environment. One could argue that this is a very obvious result for the MMPP considered, because most of the requests are dropped when the MMPP is in state 3 and the mean sojourn time in state 3 is 50 frames, which is in both examples larger than or equal to t_{max}. However, we plotted the same surface for $\alpha = 5$ instead of 50, thus, the mean sojourn time in state 3 is only 10 frames, and found a very similar surface as before (this figure is not included due to space limitations).

References

1. D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall Int., Inc., 1992.
2. C. Blondia. A discrete-time batch markovian arrival process as B-ISDN traffic model. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32(3,4), 1993.
3. D. Vázquez Cortizo. *Design and Analysis of MAC protocols for Wireless LAN*. PhD thesis, University of Antwerp, May 2000.
4. D. Vázquez Cortizo, J. García, and C. Blondia. FS-ALOHA++, a collision resolution algorithm with QoS support for the contention channel in multiservice wireless LANs. In *Proc. of IEEE Globecom*, Dec 1999.
5. D. Vázquez Cortizo, J. García, C. Blondia, and B. Van Houdt. FIFO by sets ALOHA (FS-ALOHA): a collision resolution algorithm for the contention channel in wireless ATM systems. *Performance Evaluation*, 36-37:401–427, 1999.
6. N. Golmie, Y. Saintillan, and D.H. Su. A review of contention resolution algorithms for IEEE 802.14 networks. *IEEE Communication Surveys*, 2(1), 1999.
7. G. Latouche, P.A. Jacobs, and D.P. Gaver. Finite markov chain models skip-free in one direction. *Naval Research Logistics Quarterly*, 31:571–588, 1984.
8. Y-D Lin, W-M Yin, and C-Y Huang. An investigation into HFC MAC protocols: mechanisms, implementation, and research issues. *IEEE Communication Surveys*, 3(3), 2000.
9. L. Musumeci, Paolo Giacomazzi, and Luigi Fratta. Polling and contention-based schemes for TDMA-TDD access to wireless ATM networks. *IEEE JSAC*, 18(9):1597–1607, 2000.
10. Deepak Kumar Sood and Vishwanath Sinha. Enhanced FS-ALOHA (E-FS-ALOHA) algorithm for contention resolution in wireless atm systems. In *National Conference on Communications I.I.T.*, pages 335–339, Bombay, January 2002.
11. B. Van Houdt. *Performance Analysis of Contention Resolution Algorithms in Random Access Systems*. PhD thesis, University of Antwerp (UA), 2001.
12. B. Van Houdt and C. Blondia. Robustness of FS-ALOHA. In *Proc of the 4th Int. Conf. on Matrix Analytic Methods (MAM4)*, pages 381–402, Adelaide (Australia), July 2002.
13. B. Van Houdt and C. Blondia. Robustness properties of FS-ALOHA(++): a random access algorithm for dynamic bandwidth allocation. *Mobile Networks and Applications*, 8(3):237–253, 2003.
14. B. Van Houdt, C. Blondia, O. Casals, and J. García. Performance evaluation of a MAC protocol for broadband wireless ATM networks with QoS provisioning. *Journal of Interconnection Networks (JOIN)*, 2(1):103–130, 2001.
15. R. von Mises. *Mathematical Theory of Probability and Statistics*. Academic Press Inc., New York, 1964.