Robustness of FS-ALOHA

B. Van Houdt and C. Blondia[1]

Abstract: This paper evaluates the robustness of FS-ALOHA, a random access algorithm used to reserve uplink—that is, from the end user to the network—bandwidth in centralized wireless access networks. The performance of FS-ALOHA when subject to Poisson arrivals and operating on an error free channel was evaluated in [3] by means of a Quasi-Birth-Death (QBD) Markov chain. In this paper we relax these assumptions and study discrete time batch Markovian arrivals on a channel with memoryless errors by means of a Markov chain of the GI/M/1 type. It is concluded that FS-ALOHA performs well under correlated and bursty arrivals and memoryless errors. However, error rates above $1/5T$, where T is a protocol parameter, can seriously increase the delays suffered on the contention channel and might even make the system unstable. Finally, it is concluded that implementing multiple instances of FS-ALOHA can significantly improve the delays and the robustness of the algorithm.

1 Introduction

There are, roughly speaking, two ways to transmit information on a communication channel that is shared among multiple users. Either, the protocol followed by the users avoids that two or more users transmit information at the same time, or it allows for simultaneous transmissions to occur. In the first case we refer to the channel as a contention free channel, in the latter case, the channel is referred to as a contention channel. Simultaneous transmissions are commonly known as collisions (between information) and any information that collides is considered lost, that is, the receiver is unable to retrieve the original information. Although collisions always result in the loss of information, there are many situations in which it is beneficial to use

[1] University of Antwerp, Department of Mathematics and Computer Science, Performance Analysis of Telecommunication Systems Research Group, Universiteitsplein, 1, B-2610 Antwerp - Belgium, {*vanhoudt,blondia*}@*uia.ua.ac.be*

a protocol that allows for collisions to occur, e.g., when the number of users is large and each user uses the channel on a sporadic basis.

A protocol that operates on a contention channel is called a random access algorithm (RAA) or a random access protocol. The functionality of a RAA is often subdivided as follows:

- Controlling the transmission of new informantion. This task is referred to as the channel access protocol (CAP).

- Managing retransmission after a collision occured. A task that is referred to as the contention resolution algorithm (CRA).

Thus, a RAA is a combination of a CAP and a CRA. One way to classify RAAs is to subdivide them based on their CAP. In this case, there are two main categories: RAAs with free and RAAs with blocked access, meaning that either users that generate a new information packet transmit this information immediately, or they are blocked until a certain event occurs. A subclass of the RAAs with blocked access are the RAAs with grouped access. In this particular case, new arrivals are grouped based on their arrival time and packets belonging to a certain group are not allowed to make a first transmission attempt until all the packets belonging to the previous groups have been transmitted successfully. A packet is successfully transmitted if it did not collide with another packet.

FS-ALOHA is a RAA that can be regarded as a RAA with grouped access because the requests—we refer to information packets transmitted on the contention channel as requests—are grouped in Transmission Sets (TSs) so that just one TS attempts transmission at a time (i.e., a subset of all pending requests). Also, the requests belonging to a certain TS use a CRA, in the case of FS-ALOHA one uses slotted ALOHA[2], to gain access to the medium. Hence, FS-ALOHA combines the simplicity of slotted ALOHA with the efficiency obtained by grouping the requests that arrive at the mobile stations (MSs). Although FS-ALOHA was designed to reserve bandwidth in centralized wireless access networks, it can be used for the same purpose in hybrid fiber coaxial cable (HFC) networks as an alternative for the binary exponential backoff (BEB) algorithm.

A Quasi-Birth-Death (QBD) Markov chain that allowed the performance evaluation of FS-ALOHA on an error free channel subject to Poisson arrivals was developed in [3]. This study indicated that FS-ALOHA is capable of guaranteeing low delay bounds and high throughput rates. Moreover, FS-ALOHA was shown to outperform ALOHA both in terms of delay and

[2] The slotted ALOHA algorithm is described in the next Section.

throughput. In this paper we address the robustness of FS-ALOHA and develop a Markov chain of the GI/M/1 type that allows us to evaluate FS-ALOHA on a channel with memoryless errors and D-BMAP arrivals. Thus, we can see how bursty and correlated arrivals, as well as errors, influence the performance of the algorithm. We also investigate whether some of the engineering rules, obtained from the study in [3], still apply in such errorprone systems with bursty and correlated arrivals.

Although FS-ALOHA is not believed to be as powerful as a RAA with grouped access that uses a tree algorithm [1]—also known as a splitting algorithm or an algorithm of the CTM type—as its CRA, it presents an attractive tradeoff between simplicity, that is, the ease to implement the algorithm, and its performance. The fact that simplicity is indeed a major player in the standardization of any Medium Access Control (MAC) layer was demonstrated once more during the development of the DOCSIS standard for HFC networks. Finally, it should be noted that, from an information theoretical point of view, FS-ALOHA is a full-sensing algorithm[3]; hence, it belongs to the same class of algorithms as the RAA with grouped access that uses a tree algorithm as its CRA.

The remainder of this paper is structured as follows. FS-ALOHA is described in Section 2. An informal outline of the model is given in Section 3. Section 4 and 5 present the analytical models used to evaluate FS-ALOHA. Numerical results can be found in Section 6 and conclusions are drawn in Section 7.

2 FS-ALOHA Algorithm: 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]. 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

[3] This means that a user requires feedback, that is, an indication that a packet collided or not, from the channel for each packet transmission attempt made on the channel and not merely for its own packet transmission attempts.

medium using a MAC protocol. The BS controls the access to the shared radio channel (uplink). A different frequency band is used for the uplink and downlink traffic (that is, the access technique is Frequency Division Duplex (FDD)).

Traffic on both the uplink and downlink channel is grouped into fixed length frames, with a length of L time slots, to reduce the battery consumption[4] [11]. 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[5], see Figure 1). Each uplink frame consists of a fixed length contentionless and a fixed length contention period, where the length of the contentionless period, in general, dominates that of the 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. These requests are used by MSs to declare their current bandwidth needs to the BS, e.g., by indicating how

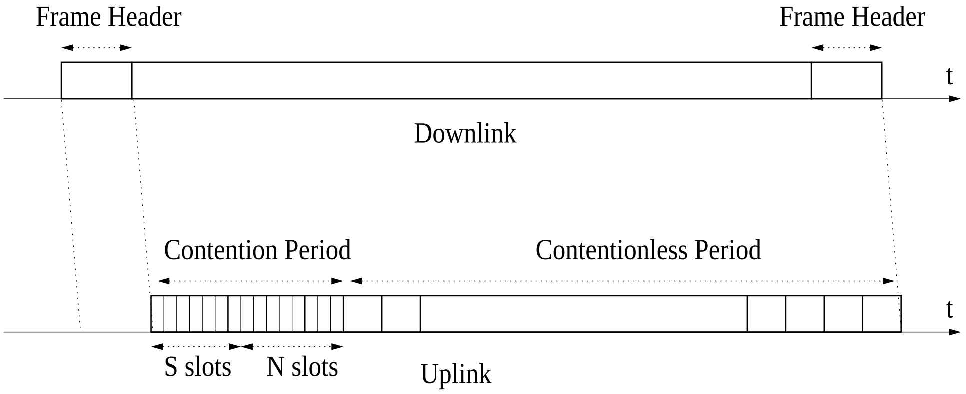

Figure 1: Frame Structure

[4] The frame structure enables the BS to inform the MSs, at the start of the frame, about the destination addresses of the downlink packets within the frame. As a result, an MS can switch to the sleep mode for the remaining frame time, unless there is a packet destined for this MS.

[5] A guard time is a small time interval at the end of each time slot during which the MSs and the BS do not transmit information. Guard times are necessary to avoid that a collision can occur between a packet that is transmitted in time slot t and $t+1$. Indeed, any information transmitted, i.e., broadcasted, by an MS (or the BS) needs a small fraction of time to reach the other MSs, therefore, the guard time has to be larger than the maximum time required by an electromagnetic wave to travel from an arbitrary MS to any other MS. Given the small size of the cells (approximately 100m), we get a small guard time.

many packets they have ready for transmission. Requests are transmitted using the contention channel, unless the MS can piggyback it to a data packet for which a permit was already obtained, thereby reducing the load on 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 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, i.e., another MS also decided to transmit in this particular slot, it will retransmit the request in one of the T slots in the next frame. It is important to note that with slotted ALOHA, new requests join the competition 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:

- 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 these transmissions were 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.

- The other N slots are used to serve the queue of backlogged TSs on a FIFO basis. A TS is served using slotted ALOHA, that is, all the requests part of the TS select one out of the N slots and transmit 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:

- The number of $S \geq 1$ slots in a frame. These slots are used to transmit newly arrived requests; S determines the TS generation rate.

- The number of $N \geq 2$ slots in a frame. These slots are allocated to the service of the backlogged TSs in the distributed 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 RAA with grouped access that uses Slotted ALOHA as its CRA, that is, the algorithm used to resolve the TSs is Slotted ALOHA. More details and extensions of FS-ALOHA can be found in [3, 2].

3 An Informal Outline of the Model

Before we proceed with a detailed description of the model, it might be useful to outline how to translate the operation of FS-ALOHA to a queueing system. We wish to evaluate the performance of FS-ALOHA under correlated and bursty arrivals, therefore, we assume that new requests generated by the MSs arrive according to a D-BMAP arrival process. The time unit of the D-BMAP is chosen to be one frame. Thus, provided that the D-BMAP is characterized by the matrices $D_0, D_1, \ldots$, there is a probability $(D_i)_{j_1,j_2}$ that i new request, each originating from a different MS, are generated in a frame, provided that the D-BMAP is in state j_1, resp. j_2, at the start, resp. end, of the frame. A first transmission attempt for each of these i new requests will take place in the S slots of this frame (that is, each of the i corresponding MSs selects one of the S slots and transmits its request in this particular slot). If all i are successful, meaning that each of the i requests was transmitted in a different slot, we state that there is no customer arrival (that is, no TS is being formed). Otherwise, the unsuccessful requests are grouped to form a TS and this transmission set is considered a customer of our queue. The number of requests part of a TS varies (each TS holds at least two requests); hence, we state that a customer is of type k if there are $k-1$ requests part of the TS. As a result, the input process of our queue can be regarded as a discrete time MMAP[K] arrival process that generates either zero or one customer during a time instance (the value of K is equal to maximum number of requests q_m that can be generated in a single frame minus one).

The service time of a customer of type k equals the number of frames required to successfully transmit each of the $k-1$ requests part of the TS. During each frame, each of the requests that remain in the TS will be transmitted in one of the N slots. Those requests that were successfully transmitted—recall that a request is successfully transmitted if it is the only request to select a particular slot—leave the TS. The unsuccessful requests remain in the TS. Thus, the progress of a service of a customer, i.e., TS, can be represented at the start of each frame by the number of requests that remain within the TS. Hence, the service time distribution of a type k customer can be represented as a discrete time phase type distribution, with matrix representation (m_k, T_k, α_k), where the phase represents the number of requests left in the TS. As a result, the queue holding the TSs is nothing but a MMAP[K]/PH[K]/1 queue. We could generalize the idea introduced in this paper to obtain a procedure that calculates the delay distribution of a type k customer in an arbitrary MMAP[K]/PH[K]/1 queue [9] and apply this general approach to this particular queue. However, in this case we are not interested in the delay distribution of a type k customer, but in the delay of a request. Moreover, the service time distributions of all the customers are very similar. Indeed, we could define a matrix T such that the service time distribution of a type k customer is identical to the phase type distribution represented by (m, T, α_k). The integer m and the matrix T are equal to m_{max} and T_{max}, where $(m_{max}, T_{max}, \alpha_{max})$ was the representation of the service time distribution related to a TS with q_m requests. The entries of the vector α_k are identical to zero, except for the k-th entry which equals one.

4 Performance Evaluation of FS-ALOHA on an Error Free Channel

4.1 Analytical Model

In this section an exact analytical model is developed, allowing the computation of the delay density function associated to the request packets under the following conditions:

- We assume a D-BMAP request arrival process with a mean rate of λ arrivals per frame.

- The number of slots T for contention is fixed and within these T slots, $S > 0$ are used by the new arrivals and $N > 1$ are used for the service of the Transmission Sets in the queue.

- If there are no Transmission Sets in the queue nor in service, the total $T = S + N$ slots is used by new arrivals.
- The Bit Error Rate (BER) is assumed to be zero, this assumption is relaxed further on.

These assumptions are identical to [3], except that we assume D-BMAP arrivals instead of Poisson arrivals. In the next section we will also relax the assumption on the BER. For Poisson arrivals one obtains a QBD Markov chain by observing the couple $(\hat{q}, \hat{Q})$ at the start of each frame, where $\hat{q}$ represents the number of requests in the TS that is currently in service (provided that a TS is in service) and $\hat{Q}$ is the number of TSs waiting in the distributed FIFO queue[6]. If we consider the same stochastic process for D-BMAP arrivals and add the current state of the D-BMAP, say $\hat{j}$, we no longer have a Markov chain. Therefore, a different approach is required; the basic idea is to remember the "age" of the TS currently in service instead of the number of TSs waiting in the TS queue[7]. The state of the system is modeled by the triple (q, j, Q), where

- $q \geq 2$ denotes the number of requests in the Transmission Set that is currently in service (if there is a Transmission Set in service).
- j denotes the state of the D-BMAP associated with the frame that follows the frame in which the Transmission Set currently in service was generated (if there is a Transmission Set in service, otherwise it is the state of the D-BMAP associated with the current frame).
- Q indicates how many frames ago the Transmission currently in service Set was created ($Q = 0$ if there is no Transmission Set in service).

For instance, $(q, j, Q) = (4, j, 3)$ indicates that 4 requests will attempt a transmission in the N slots of the current frame, say frame n. Each of these

[6] Level 0 consists of one state that corresponds to the case where there are no TSs waiting in the queue and there is no TS in service, level $i > 0$ consists of multiple states that correspond to the case where there are $i - 1$ TSs waiting in the queue and a TS is in service (the j-th state of level i indicates that there are $j + 1$ requests left in the TS).

[7] This trick can also be used to obtain the waiting time distribution for each class of customers in a discrete time FCFS MMAP[K]/PH[K]/1 queue provided that the MMAP[K] arrival process has $D_J = 0$ for all strings J with a length $|J| > 1$ (i.e., the customers arrive one at a time). In this case one remembers: the age of the customer currently in service, its class type, the state of its service and the state of the MMAP[K] input process. Because the age can only increase by one at a time we obtain a GI/M/1 Type Markov chain by observing the system at each time slot. Moreover, in [9], we have generalized this technique to MMAP[K] arrival processes with batch arrivals.

4 stations has had at least 1, in frame $n-3$, and at most 3 unsuccessful attempts in the previous 3 frames (depending on the service completion time of the previous TS) and the state j of the D-BMAP determines the number of requests that make use of the S slots in frame $n-2$. If, for example, 2 of the 4 request are transmitted successfully (within the N slots of frame n), the new state, associated with frame $n+1$, would be $(2, j, 4)$.

Notice that this model can be used for Poisson arrivals as well. Moreover, although the model in [3] uses a QBD Markov chain, the calculations required to obtain the delay distribution from the steady state probabilities are cumbersome. Whereas with this model, that uses a GI/M/1 type Markov chain, one obtains the delay distribution from the steady state probabilities by means of a simple formula (see Section 4.5).

4.2 Transition Matrix

The transitions in the system take place at the start of each frame. The maximum value of q, say q_m, corresponds to the highest possible i for which D_i contains entries that differ from zero, where D_i are the $l \times l$ matrices that characterize the input D-BMAP traffic. 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 $D_i, i > q_m$ is negligible. Therefore, the impact on the accuracy of the results is minimized. The range of j is equal to $\{j \mid 1 \leq j \leq l\}$. During a state transition, Q can never increase by more than one.

Therefore, the system can be described by a transition matrix P with the following structure:

$$P = \begin{bmatrix} B_1 & B_0 & 0 & 0 & 0 & \ldots \\ B_2 & A_1 & A_0 & 0 & 0 & \ldots \\ B_3 & A_2 & A_1 & A_0 & 0 & \ldots \\ B_4 & A_3 & A_2 & A_1 & A_0 & \ldots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{1}$$

where A_i are $l(q_m - 1) \times l(q_m - 1)$ matrices, $B_i, i > 1$, 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.

The matrices B_0 and B_1 describe the system when the current frame is not serving a Transmission Set ($Q = 0$). This implies that the total of $T = S + N$ slots is used for new arrivals. B_0 describes the transitions when a Transmission Set is generated within these T slots, whereas B_1 describes the situation in which no Transmission Set is generated.

The matrices A_i and $B_i, i > 1$, hold the transition probabilities provided that a Transmission Set t is in service in the current frame. A_0 covers the case in which the service of the current Transmission Set t is not completed within the current frame. The transition probabilities held by the matrices $A_i, i > 0$, correspond to the following situation: the service of the current Transmission Set t is completed within the current frame, say frame n, and the first $i-1$ frames following frame $n-Q$, i.e., the frame in which the Transmission Set t was generated, do not generate a new Transmission Set, whereas frame $n-Q+i$ $(\leq n)$ does generate a new Transmission Set. The matrices $B_i, i > 1$ on the other hand correspond to case where the service of the current Transmission set t is completed within the current frame, frame n, and the first $i-1$ $(= Q)$ frames following frame $n-Q$ do not generate a new Transmission Set (as a result the total of $T = S+N$ slots is used for new arrivals in frame $n+1$).

4.3 Calculating the Transition Probabilities

In this subsection we indicate how to calculate the matrices A_i and B_i described above. 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[8]. We are particularly interested in $p_S(q,q')$, $p_N(q,q')$ and $p_{S+N}(q,q')$. Von Mises [12] has shown, in 1939, that

$$p_x(q,q') = \sum_{v=q-q'}^{\min(q,x)} (-1)^{v+q-q'} C_{q-q'}^{v} C_v^x \frac{q!}{(q-v)!} \frac{(x-v)^{q-v}}{x^q}, \tag{2}$$

where C_s^r denotes the number of different ways to choose s from r different items. Equation 2 is numerically stable for the parameter ranges of interest ($x \leq 20$). It is also possible to calculate the $p_x(q,q')$ values recusively 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 $P_{N,0}$ be a $q_m - 1 \times 1$ vector whose i^{th} component equals $p_N(i+1, 0)$. In order to describe the matrices A_i and B_i we also define the matrices F_S, F_{S+N}, $E_S^k, 2 \leq k \leq q_m$, and $E_{S+N}^k, 2 \leq k \leq q_m$, as (these

[8] 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 ball.

matrices are $l \times l$ matrices)

$$\begin{aligned} F_S &= \sum_{i \geq 0} D_i \; p_S(i,0) & (3) \\ F_{S+N} &= \sum_{i \geq 0} D_i \; p_{S+N}(i,0) & (4) \\ E_S^k &= \sum_{i \geq k} D_i \; p_S(i,k), & (5) \\ E_{S+N}^k &= \sum_{i \geq k} D_i \; p_{S+N}(i,k), & (6) \end{aligned}$$

where the D-BMAP arrival process is characterized by the matrices D_i. Notice that $(E_S^k)_{j,j'}$ represents the probability that a new TS with k requests is generated in a frame where S slots are used for the new arrivals, thus, another TS is currently in service in the remaining N slots, and the D-BMAP governing the new arrivals makes a transition from state j to j'. F_S on the other hand holds the probabilities that no new TS is generated in a frame where S slots are used for new arrivals. Similar interpretations exist for the matrices F_{S+N} and E_{S+N}^k. The transition probability matrices A_i and B_i are then found as follows:

$$\begin{aligned} A_0 &= P_N \otimes I_l, & (7) \\ A_i &= P_{N,0} \otimes \left((F_S)^{i-1} \left[E_S^2 \; E_S^3 \; \ldots \; E_S^{q_m} \right] \right), & (8) \\ B_0 &= \left[E_{S+N}^2 \; E_{S+N}^3 \; \ldots \; E_{S+N}^{q_m} \right], & (9) \\ B_1 &= F_{S+N}, & (10) \\ B_i &= P_{N,0} \otimes (F_S)^{i-1}, & (11) \end{aligned}$$

where $\otimes$ denotes the Kronecker product between matrices and I_l the $l \times l$ unity matrix. Notice that the matrices A_i and B_i decrease to zero according to $(F_S)^i$. Looking at the probabilistic interpretation of F_S, it should be clear that, in general, the smaller the arrival rate λ the slower A_i and B_i decrease to zero. Therefore, the model is not suited for very small arrival rates λ (because this would imply that thousands of A_i and B_i matrices are needed to perform the calculations).

4.4 Calculating the Steady State Probabilities

Define $\pi_i^n(q,j), i > 0$, resp. $\pi_0^n(j)$, as the probability that the system is in state (q,j,i), resp. $(j,0)$, at time n, i.e., at the start of frame n. Let

$$\begin{aligned} \pi_0(j) &= \lim_{n \to \infty} \pi_0^n(j), & (12) \\ \pi_i(q,j) &= \lim_{n \to \infty} \pi_i^n(q,j). & (13) \end{aligned}$$

Define the $1 \times l$ vector $\pi_0 = (\pi_0(1), \ldots, \pi_0(l))$ and the $1 \times l(q_m - 1)$ vectors $\pi_i = (\pi_i(2,1), \ldots, \pi_i(2,l),\ \pi_i(3,1), \ldots, \pi_i(3,l), \pi_i(4,1), \ldots, \pi_i(q_m, l)),\ i > 0$. From the transition matrix P (Equation 1) we see that the Markov chain is a generalized Markov chain of the $GI/M/1$ Type [6]. From such a positive recurrent Markov chain, we have $\pi_i = \pi_{i-1}R, i > 1$, where R is an $l(q_m - 1) \times l(q_m - 1)$ matrix that is the smallest nonnegative solution to the following equation:

$$R = \sum_{i \geq 0} R^i A_i. \tag{14}$$

This equation is solved by means of an iterative scheme [6]. In order to obtain π_0 and π_1 we solve the following equation

$$(\pi_0, \pi_1) = (\pi_0, \pi_1) \begin{bmatrix} B_1 & B_0 \\ \sum_{i \geq 2} R^{i-2} B_i & \sum_{i \geq 1} R^{i-1} A_i \end{bmatrix}. \tag{15}$$

The vector (π_0, π_1) is normalized as $\pi_0 e_l + \pi_1 (I - R)^{-1} e_{l(q_m - 1)} = 1$, where I is the unity matrix of size $l(q_m - 1)$ and e_i is an $i \times 1$ vector filled with ones. Theorem 1.5.1 in [6] states that the Markov chain with transition matrix P is positive recurrent if and only if the spectral radius $sp(R)$ of the matrix R, where R is the minimal nonnegative solution to Equation 14, is smaller than one and there exists a positive solution to Equation 15. It is not difficult to see that $A = \sum_{i \geq 0} A_i$ is irreducible[9], provided that the input D-BMAP is irreducible, therefore a simple condition exists to check whether $sp(R) < 1$ [6, 7]. We could also check the positive recurrence by noticing that FS-ALOHA, when subject to D-BMAP arrivals, is equivalent to a discrete time MMAP[K]/PH[K]/1 queue with a generalized initial condition, where the MMAP[K] stands for a Markov chain with marked arrivals [5]. The stability of such queues has been studied by He [4, Theorem 7.1].

4.5 Calculating the Delay Density Function

Let D be the random variable that denotes the delay suffered by a request packet. We state that $D = 0$ if a request packet is successful during its first attempt. $D = i$ if a request packet is successful in frame $n + i$ provided that

[9] After removing the possible (obvious) transient states of level $Q > 0$. Indeed, the states (q, j, Q), for $Q > 0$, are transient if the j-th entry of the vector $\theta \sum_{i \geq q} D_i$ equals zero, where θ is the stochastic stationary vector of $\sum_{i \geq 0} D_i$. It is not necessary to remove their corresponding rows and columns when calculating the steady state probabilities, because the algorithm outlined in Section 4.4 will automatically assign a probability zero to these states.

the first attempt took place in frame n. Using the steady state probabilities we easily find

$$P[D = i] = \sum_{q=2}^{q_m} \frac{(1-1/N)^{q-1}q}{\lambda} \sum_{j=1}^{l} \pi_i(q,j), \tag{16}$$

for $i > 0$, with λ the arrival rate of the D-BMAP, i.e., the mean number of newly arriving request packets per frame. While $P[D = 0]$ is found as $1 - \sum_{i>0} P[D = i]$.

5 Performance Evaluation of FS-ALOHA on a Channel with Memoryless Errors

In this section we relax the assumption on the BER made in the previous section, and allow for memoryless errors to occur. From a practical point of view, Markovian errors would probably be more appropriate, but there seems to be no apparent way to incorporate such errors in the current model, even if we were to restrict ourselves to Poisson arrivals. Perhaps a short explanation is appropriate. If we assume Markovian errors, the number of requests in a TS depends, among other things, upon the error state related to the frame in which the TS is created. We define the error state as the state of the Markov chain governing the errors. This is similar to the model in the previous section where the number of requests in a TS depended, in a similar way, on the state of the arrival process. However, with Markovian errors the resolution of a TS with k requests is influenced by the error state, whereas this is not the case for the state associated to the arrival process. Thus, if we want to enrich the previous model with Markovian errors we need to keep track of the error state in the current frame, and of the error state related to the frame in which the TS currently in service was created; therefore, in order to obtain a Markov chain that observes the system at every frame time—a desirable property if we want to calculate the delay distribution with a simple formula from the steady state probabilities—we need to keep track of the entire history of the error state between these two time instances. This would clearly result in an explosion of the state space, unless the Markov chain has only one state, that is, if the errors are memoryless. Therefore, we restrict ourselves to memoryless errors and state that an error occurs in a slot with a probability $0 \leq e \leq 1$.

Errors occurring on the channel influence the transmissions as follows. If a slot holds a collision, that is, if two or more MSs transmit a request in the same slot, then the BS, correctly, interprets this slot as a collision, whether or not an error occurred in this slot. On the other hand, if a slot does not hold a collision and an error does occur in the slot, the BS will, incorrectly, interpret the slot as holding a collision. A slot that neither holds a collision or an error is correctly recognized by the BS. As a result, a single error in the slots dedicated to the new arrivals is sufficient to create a new TS; hence, TSs with zero or one request exist, as opposed to the model in the previous section. Also, the average number of frames required to resolve a TS with k requests increases due to the presence of errors. The service of a TS ends if the N slots, assigned to the service of TSs, do not hold an unsuccessful transmission nor an error.

It should be clear that the triple (q, j, Q) as defined in the previous section is still a Markov chain of the GI/M/1 type. However, the entries and the size, because TSs with zero or one request exist, of the matrices A_i and B_i have changed. These matrices will be denoted as $\tilde{A}_i$ and $\tilde{B}_i$ in order to avoid any confusion with the matrices of the previous section (this is also done for other matrices or vectors that appear in both sections).

First, define $p_x^E(q, q')$ as the probability that in a set of q requests, $q - q'$ are successful when a set of x slots is used to transmit the q request packets and this provided that at least one error occurs in these x slots. Because the errors are memoryless we have

$$p_x^E(q, q') = \sum_{k=1}^{x} C_k^x e^k (1-e)^{x-k} \sum_{v=\max(0,q'-k)}^{q'} p_x(q, v) \frac{C_{q'-v}^k C_{q-q'}^{x-k}}{C_{q-v}^x}, \qquad (17)$$

where $p_x(q, q')$ was defined in Section 4.3 and e represents the probability that an arbitrary slot holds an error. Obviously, we are interested in $p_S^E(q, q')$, $p_N^E(q, q')$ and $p_{S+N}^E(q, q')$.

Next, denote P_N^E as a $q_m + 1 \times q_m + 1$ matrix whose $(i, j)^{th}$ element equals $p_N^E(i-1, j-1)$. $\tilde{P}_N$ is defined as a $q_m + 1 \times q_m + 1$ matrix whose first two columns are equal to zero and whose $(i, j)^{th}$ element, for $j > 2$, equals $(1-e)^N p_N(i-1, j-1)$. The $q_m + 1 \times 1$ vector $\tilde{P}_{N,0}$ has its i^{th} entry equal to $(1-e)^N p_N(i-1, 0)$. Finally, the $l \times l$ matrices $\tilde{F}_S$, $\tilde{F}_{S+N}$, $\tilde{E}_S^k$, for $0 \leq k \leq q_m$, and $\tilde{E}_{S+N}^k$, for $0 \leq k \leq q_m$, are defined as

$$\tilde{F}_S = \sum_{i \geq 0} D_i \; p_S(i, 0) \; (1-e)^S \qquad (18)$$

$$\tilde{F}_{S+N} = \sum_{i \geq 0} D_i \; p_{S+N}(i, 0) \; (1-e)^{S+N}, \qquad (19)$$

$$\begin{aligned}
\tilde{E}_S^k &= \sum_{i\geq k} D_i \ \left[1_{\{k>0\}}\ p_S(i,k)\ (1-e)^S + p_S^E(i,k)\right], && (20)\\
\tilde{E}_{S+N}^k &= \sum_{i\geq k} D_i \ \left[1_{\{k>0\}}\ p_{S+N}(i,k)\ (1-e)^{S+N} + p_{S+N}^E(i,k)\right], && (21)
\end{aligned}$$

where $1_A = 1$ if A is true and 0 otherwise. Notice that $p_x(i,1) = 0$, therefore, it is sufficient to write $1_{\{k>0\}}$ instead of $1_{\{k>1\}}$. The matrices $\tilde{E}_x^k$ hold the probability that a new TS with $k \geq 0$ requests is generated in a frame where x slots are used for the new arrivals. $\tilde{F}_x$ on the other hand holds the probabilities that no new TS is generated. We are now in a position to specify the matrices $\tilde{A}_i$ and $\tilde{B}_i$:

$$\begin{aligned}
\tilde{A}_0 &= (\tilde{P}_N + P_N^E) \otimes I_l, && (22)\\
\tilde{A}_i &= \tilde{P}_{N,0} \otimes \left((\tilde{F}_S)^{i-1} \left[\tilde{E}_S^0\ \tilde{E}_S^1\ \ldots\ \tilde{E}_S^{q_m}\right]\right), && (23)\\
\tilde{B}_0 &= \left[\tilde{E}_{S+N}^0\ \tilde{E}_{S+N}^1\ \ldots\ \tilde{E}_{S+N}^{q_m}\right], && (24)\\
\tilde{B}_1 &= \tilde{F}_{S+N}, && (25)\\
\tilde{B}_i &= \tilde{P}_{N,0} \otimes (\tilde{F}_S)^{i-1}, && (26)
\end{aligned}$$

where I_l is the $l \times l$ unity matrix. The steady state probabilities, denoted as $\tilde{\pi}_i$, are calculated in a similar manner as before. Finally, the delay distribution $P[\tilde{D} = i]$, for $i > 0$, is found as

$$P[\tilde{D} = i] = \sum_{q=0}^{q_m} \frac{(1-e)(1-1/N)^{q-1}q}{\lambda} \sum_{j=1}^{l} \tilde{\pi}_i(q,j). \qquad (27)$$

$P[\tilde{D} = 0]$ is found as $1 - \sum_{i>0} P[\tilde{D} = i]$.

6 Numerical Results

In this section we explore the influence of correlation, burstiness, the number of $T = S + N$ slots and memoryless errors on the delay distribution of a request packet. A first, important, question that needs to be addressed is: What type of D-BMAP arrivals should be considered, that is, are of practical relevance ? Clearly, for any arrival rate λ and medium access protocol we can find a D-BMAP that causes delays as high as we like.

From Section 2 we know that if the traffic flow generated by an MS is very irregular, the MS is obliged to use the contention channel frequently. Therefore, depending on the characteristics of the traffic flow, we regard an MS as either being in a period where most the requests are piggybacked to the data packets transmitted in the contention free period, or in a period where the contention channel is used to transmit most of the requests. As a result, we will identify M different levels of activity, where a higher level indicates that more MSs are in a period where the contention channel is used frequently. We use M states to model these activity levels and state that the number of requests generated in a frame, by the arrival process, in state j is distributed binomially with parameters (jm, β), where m and β are parameters of the model. Hence, denoting $(D_i e)_j$, for $1 \leq j \leq M$, as the j-th component of D_i multiplied with e, an $M \times 1$ vector with all entries equal to one, results in

$$(D_i e)_j = C_i^{jm} \beta^n (1-\beta)^{jm-i}. \tag{28}$$

Transitions between these M states, occuring at the end of each frame, take place according to the following $M \times M$ transitions matrix P_M:

$$P_M = \begin{bmatrix} 1-\alpha^+ & \alpha^+ & 0 & \dots & 0 \\ \alpha^- & 1-\alpha^+-\alpha^- & \alpha^+ & \dots & 0 \\ 0 & \alpha^- & 1-\alpha^+-\alpha^- & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1-\alpha^- \end{bmatrix}. \tag{29}$$

Therefore, $(D_i)_{j_1,j_2}$ equals $(D_i e)_{j_1} (P_M)_{j_1,j_2}$. Thus, the arrival process is characterized by the following five parameters: M, m, β, α^- and α^+. In this section, the parameters M and m are fixed at 6 and 5, whereas the parameter β is set such that de arrival rate λ is $0.2T$ requests per frame; hence, the throughput on the contention channel is 20% (provided that the Markov chain is positive recurrent). An average input rate of 20%, on a contention channel, is considered as realistic because higher values would imply that the number of contention slots T is underestimated by the network designer and the network would have great difficulties in guaranteeing any QoS, whatever protocol is used on the contention channel. Notice, with $M = 6$ and $m = 5$, the mean arrival rate related to state j is $5j/\beta$. Finally, it should be clear that this arrival process is an M-state D-BMAP.

6.1 Poisson Arrivals vs. D-BMAP Arrivals

In this section we compare the delay distribution of a request packet for Poisson and D-BMAP arrivals. For now, the bit error rate (BER) is equal to zero; hence, we use the model presented in Section 4. For the D-BMAP arrivals we fix $\alpha^+ = \alpha^- = 1/5$, therefore, the mean sojourn time in a state

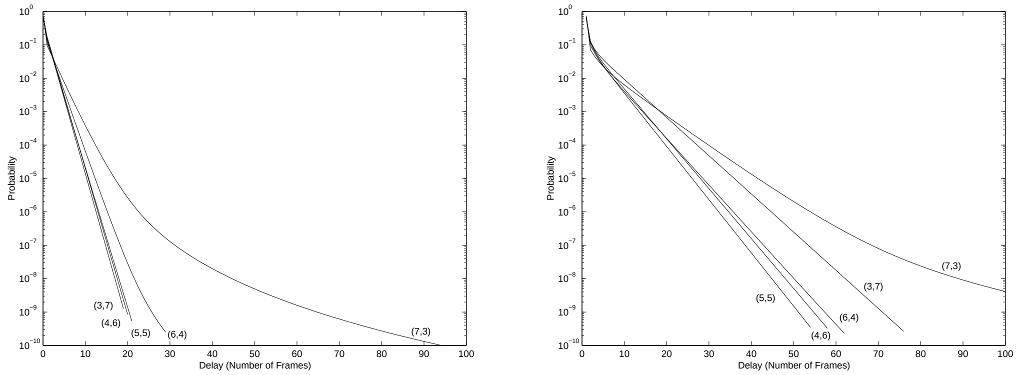

Figure 2: Delay distribution for $T = 10$, Left: Poisson arrivals ($\lambda = 2$), Right: D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^{+} = \alpha^{-} = 1/5$ and β such that the arrival rate $\lambda = 2$).

is 2.5 frames. The number of contention slots $T = S + N = 10$, whereas the number of S and N slots varies and is represented in the figures as (S, N). The results are presented in Figure 2.

A first, obvious, observation in Figure 2 is that the delays are larger for D-BMAP arrivals. This follows from the fact that for Poisson arrivals the mean arrival rate is always 2, whereas for the D-BMAP arrivals we have periods were the mean arrival rate is as low as $2/3.5 = 4/7$, being when the arrival process is in state 1, and periods were the mean arrival rate is as high as $24/7$, being when the arrival process is in state $M = 6$. A second observation is that the delay distribution decays exponentially[10], except for N small. To some extent, this can be explained by means of Equation 16, that is, if we forget about the q in Equation 16 and approximate $(1-1/N)^{q-1}$ by one, we get an exponential decay. Finally, in [3], it was shown that, for Poisson arrivals, the best delays are obtained with $S \approx N$. Figure 2 seems to confirm the usefulness of this engineering rule, which is also based on the intuitive idea that $S \approx N$ provides the best balance between the TSs generation rate, related to S, and the TSs service times, related to N.

6.2 The Influence of the Number of Contention Slots (T)

Apart from checking whether the engineering rule concerning the number of S and N slots still applies, this section addresses the issue whether it is worth implementing parallel instances of FS-ALOHA in the contention

[10] This is not exactly true, what we mean here is that this seems to be the case if we consider the 1 to 10^{-10} region only.

period. With parallel instances we mean the following. Suppose that we have $T = T_1 T_2$ contention slots, with $T_1 \geq 3$. Then, we could use T_2 instances of FS-ALOHA, that each use T_1 slots. New arrivals decide which instance they use based on their arrival time—that is, we partition the frame in T_2 subframes and any new arrival occurring in the i-th subframe, uses the i-th instance[11]. In this scenario we have T_2 distributed queues with TSs, instead of one. Clearly, implementing multiple instances increases the complexity of the algorithm, but perhaps the delay improvements outweigh the additional implementation effort.

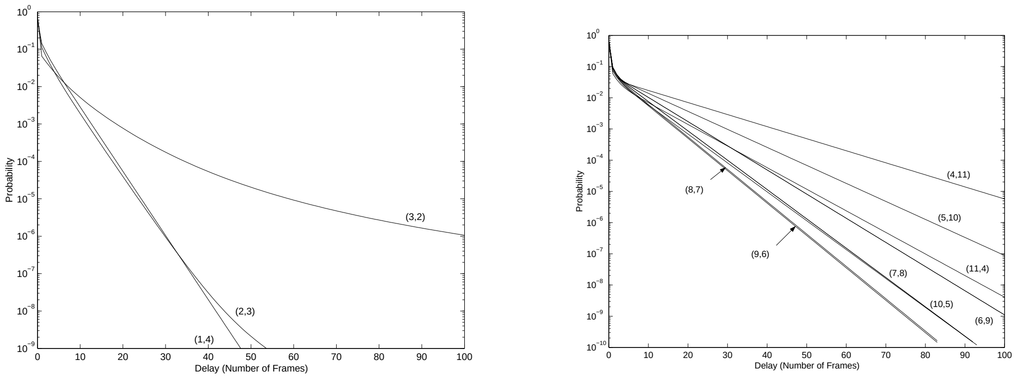

Figure 3: Delay distribution for D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^+ = \alpha^- = 1/5$), Left: $T = 5$ and β such that $\lambda = 1$, Right: $T = 15$ and β such that $\lambda = 3$.

Figure 3 presents the results for $T = S + N = 5$ and $T = 15$ contention slots[12]. The input process is the same as in the previous paragraph, except that β is chosen such that $\lambda = 0.2T$. For $T = 5$ the best results are found for N larger than S, whereas for $T = 15$ we get the best results for S slightly larger than T. In conclusion choosing $S \approx N$ seems like a useful rule of thumb. As far as the parallel instances are concerned, we can see by comparing the results for $T = 5$ and 15 that the delays can be reduced by a factor two using three instances with $T = 5$ instead of one with $T = 15$. Thus, if a network designer provisions a lot of contention slots, we suggest to implement more than one instance of FS-ALOHA.

[11] Instead of using their arrival time, a request could also select the instance randomly. Given that the arrivals occur uniformly in a frame, these two scenarios are the same.

[12] It should be noted that, provided that the arrivals occur uniformly in a frame, we can evaluate the performance of multiple instance by adapting the value of β appropriately. Indeed, it is easy to show that $\sum_{g \geq k} C_g^{mi} \beta^g (1-\beta)^{mi-g}\, T_2^{-k}(1-T_2^{-1})^{g-k} = C_k^{mi}(\beta/T_2)^k(1-\beta/T_2)^{mi-k}$, where T_2 denotes the number of instances used.

6.3 Correlation and Burstiness

In this section we study the influence of the mean sojourn time on the delay distribution. We start with $\alpha^+ = \alpha^- = 1/2$ and decrease both gradually until $1/50$, in which case the mean sojourn time in a state is 25 frames. The results are in presented in Figure 4, the other parameters are the same as in Section 6.1. From this figure we can conclude that the grouping strategy works well in limiting the delay increase due to the augmented correlation and burstiness.

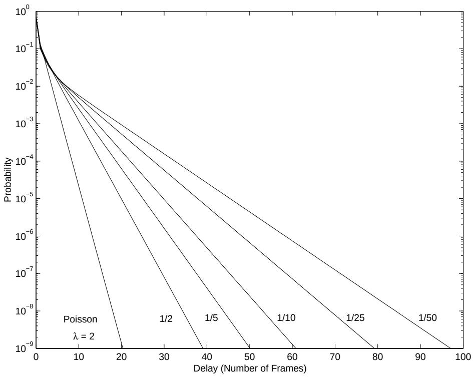

Figure 4: Delay distribution for D-BMAP arrivals ($M = 6$, $m = 5$, β such that $\lambda = 2$), $T = 10, S = N = 5$.

6.4 Errors on the Channel

In this section we investigate the influence of errors on the channel by means of the model presented in Section 5. We start by setting e, the probability that a slots holds an error, equal to $1/50, 1/100$ and $1/250$. It is hard to state whether such a value of e is an optimistic or pessimistic estimate as the probability of an error depends on the modulation scheme, the signal-to-noise ratio (SNR), the forward error control (FEC), length of a slot and much more [8]. For a wired channel it is safe to say that $e = 1/50$ is very pessimistic. We start by reproducing Figure 2 for $e = 0, 1/50, 1/100$ and $1/250$ and $S = N = 5$. Numerical experiments, omitted for brevity, show that errors have a similar impact on the delay for other choices of S and N, with $S + N = 10$ (actually, the impact of errors is slightly smaller for larger values of S).

The results are presented in Figure 5, where the curves for $e = 0$ where obtained with the model in Section 4. A first, obvious, observation is that

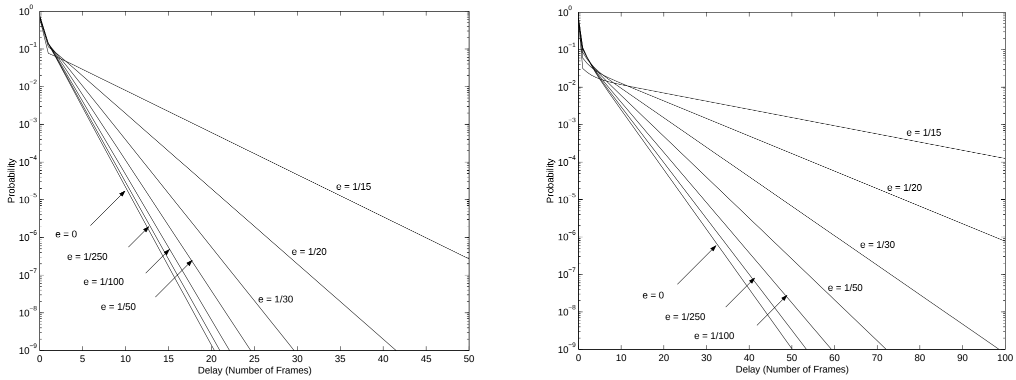

Figure 5: Delay distribution for $T = 10, S = N = 5$ and $e = 0, 1/250, 1/100, 1/50$, Left: Poisson arrivals ($\lambda = 2$), Right: D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^{+} = \alpha^{-} = 1/5$ and β such that the arrival rate $\lambda = 2$).

the delay increases with increasing e. Moreover, the results show that the increase for Poisson arrivals is less compared to D-BMAP arrivals. Thus, models that study the impact of errors using Poisson arrivals are, from a practical point of view, somewhat optimistic. Therefore, we use D-BMAP arrivals for our remaining experiments. Finally, although the impact on the delay distribution is small for $e = 1/100$ or smaller, errors can seriously increase the delay for higher error rates (for $e = 1/20$ the delays are more than three times as high compared to $e = 0$). Therefore, if the modulation scheme, error codes, signal-to-noise ratio, ... cannot guarantee an error rate e less than $1/5T$, the performance of FS-ALOHA might degrade drastically.

This rule is confirmed by Figure 6, where we study FS-ALOHA for $T = 5$ and 15. For $T = 15$ the Markov chain becomes transient for $e \geq 1/20$ (actually, the chain becomes unstable for e somewhere in between 1/20 and 1/21). For Poisson arrivals and $T = 15$ we get instability for $e \geq 1/19$, thus the instability is only slightly influenced by the arrival process and is mainly determined by the error rate.

These observations further indicate that the use of multiple instances of FS-ALOHA, each with a small value of T, is not only better in terms of the suffered delay, but also improves the sensitivity of the algorithm to errors. As a result, we strongly support the use of multiple instances for wireless networks, i.e., networks with high error rates.

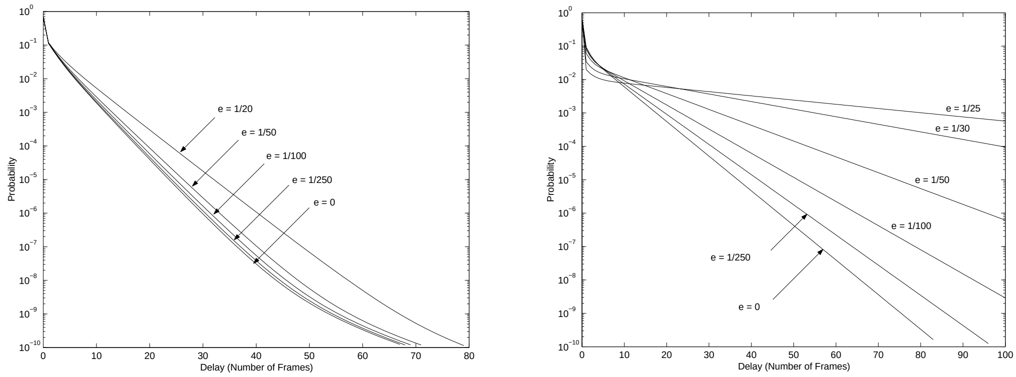

Figure 6: Delay distribution for D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^+ = \alpha^- = 1/5$ and β such that the arrival rate $\lambda = 0.2T$), $e = 0$ to $1/20$, Left: $T = 5, S = 2, N = 3$, Right: $T = 15, S = 8, N = 7$.

7 Conclusions

In this paper we have evaluated the robustness of FS-ALOHA, a random access algorithm, by means of an GI/M/1 Type Markov chain. The robustness was investigated by relaxing prior assumptions [3] made on the arrival process, that is, discrete time batch Markovian arrivals were considered as opposed to Poisson arrivals. Moreover, memoryless errors were also added to the channel. Using the analytical model, it is concluded that FS-ALOHA, in general, performs well under correlated and bursty arrivals and memoryless errors. However, error rates above $1/5T$, were T is a protocol parameter, can seriously increase the delays suffered on the contention channel and might even make the system unstable at moderate arrival rates. It should be mentioned that FS-ALOHA++ [2] might, to some extent, improve the stability of FS-ALOHA on a channel with errors, because FS-ALOHA++ services multiple TSs simultaneously, thereby reducing the penalty introduced by empty transmission sets. This and many other properties of FS-ALOHA++ are reported in [10], where we also use matrix analytic methods to obtain the performance measures of interest. Finally, it is concluded that implementing multiple instances of FS-ALOHA can significantly improve the delays and the robustness of the algorithm and is therefore advisible for wireless channels with high error rates.

Acknowledgements

B. Van Houdt is a postdoctoral Fellow of the FWO Flanders. We would like to thank the reviewers for their valuable comments and suggestions.

References

[1] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall Int., Inc., 1992.

[2] 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.

[3] 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.

[4] Q. He. Classification of Markov processes of M/G/1 type with a tree structure and its applications to queueing models. *O.R. Letters*, 26:67–80, 1999.

[5] Q. He. Classification of Markov processes of matrix M/G/1 type with a tree structure and its applications to the MMAP[K]/G[K]/1 queue. *Stochastic Models*, 16(5):407–434, 2000.

[6] M.F. Neuts. Markov chains with applications in queueing theory, which have a matrix geometric invariant probability vector. *Adv. Appl. Prob.*, 10:185–212, 1978.

[7] M.F. Neuts. *Matrix-Geometric Solutions in Stochastic Models, An Algorithmic Approach.* John Hopkins University Press, 1981.

[8] K. Pahlavan and A.H. Levesque. *Wireless Information Networks.* John Wiley and Sons, Inc., New York, 1995.

[9] B. Van Houdt and C. Blondia. The delay distribution of a type k customer in a first come first served MMAP[K]/PH[K]/1 queue. *Journal of Applied Probability (submitted)*, 2002.

[10] B. Van Houdt and C. Blondia. Robustness properties of FS-ALOHA(++): a random access algorithm for dynamic bandwidth allocation. *Journal on Special Topics in Mobile Networking and Applications (MONET) on Performance Evaluation of QoS Architectures in Mobile Networks (submitted)*, 2002.

[11] 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.

[12] R. von Mises. *Mathematical Theory of Probability and Statistics.* Academic Press Inc., New York, 1964.