Robustness Properties of FS-ALOHA(++): a contention resolution algorithm for dynamic bandwidth allocation

B. Van Houdt[1], C. Blondia

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

Abstract

This paper studies the robustness of FS-ALOHA(++), a contention resolution algorithm used to reserve uplink bandwidth in wireless centralized LANs. The model takes into account errors on the contention channel, the capture effect and allows packets to arrive according to a general Markovian arrival process. Where channel errors and capture are typical for a wireless channel. Several analytical models are developed, using matrix analytical methods, allowing us to calculate the delay distribution of a request packet under different circumstances. Using these analytical models, we demonstrate that both FS-ALOHA and FS-ALOHA++ perform well under bursty and correlated arrivals. FS-ALOHA++ is shown to be more robust towards errors and capture. Moreover, it is concluded that implementing multiple instances of FS-ALOHA(++) improves significantly the delays and sensitivity of the algorithm towards errors.

[1] Postdoctoral Fellow of the FWO-Flanders

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, 5, 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 [2, 4]. 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 [2, 3, 4]. However, all these studies assumed Poisson arrivals, an error free channel and no capture events. In order to obtain a more realistic view of the performance of FS-ALOHA(++) it is essential to allow for errors and capture to occur. Moreover, assuming Poisson arrivals may not be adequate for many current and future wireless applications. Therefore, we consider packets arriving according to a discrete time batch Markovian arrival process (D-BMAP). DBMAPs form a class of tractable Markovian arrival processes, which, in general, are non-renewal, and which include the discrete time variants of the Markov modulated Poisson process, the PH-renewal process and superpositions of such processes as particular cases. Because of its versatility, it lends itself very well to modeling bursty arrival processes commonly arising in computer and communications applications [1, 11, 12]. During the last ten years, D-BMAPs have been used by many researchers to develop more realistic arrival processes.

The paper is structured as follows. Section 2 introduces FS-ALOHA and FS-ALOHA++, as well as the environment in which they operate. Sections 3, 4 and 7 present three different analytical models that evaluate FS-ALOHA under different circumstances. Section 3 considers D-BMAP arrivals, but still not errors or capture. In Section 4 errors are included in the model, while Section 7 discusses the influence of capture. Sections 5, 6 and 7 present three analytical models for FS-ALOHA++, a slightly more advanced version of FS-ALOHA, under similar circumstances. Sections 8 and 9 present a variety of numerical examples. Conclusions are drawn in Section 10.

2 FS-ALOHA(++): a Review

In this section the operation of FS-ALOHA(++), and the environment in which they operate, are described in some detail, additional comments and discussions can be found in [2, 3, 4]. 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) [13].

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, 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. 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 $T = S + N$ (see Figure 1). 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 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.

- 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 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 by the MSs 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 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 Protocol that uses Slotted ALOHA as its collision resolution algorithm (CRA). More details on the operation of FS-ALOHA can be found in [2, 3, 4].

Numerical experiments [2, 4] have indicated that the maximum stable throughput of FS-ALOHA, under Poisson arrivals, tends to decrease as a function of T. The maximum stable throughput for larger values of T was augmented by introducing FS-ALOHA++ [2, 3]. It differentiates itself from FS-ALOHA by serving K TSs simultaneously in the N slots of each frame (provided that K TSs were waiting in the FIFO queue at the start of the service, otherwise it serves, simultaneously, all the TSs that were waiting). Each of the requests contained in the TSs in service, selects one of the N slots and transmits in this slot hoping that no other station does so. Thus, in multiple access terminology, one could state that FS-ALOHA++ alters the channel access protocol (CAP) of FS-ALOHA, while both use Slotted ALOHA as their collision resolution algorithm (CRA). The strategy used by FS-ALOHA and FS-ALOHA++ when errors or capture occurs will become apparent in the remainder of the paper.

3 FS-ALOHA with D-BMAP arrivals on an Error Free Channel

3.1 Analytical Model

FS-ALOHA with Poisson input has been studied in [4]. 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 [1] with a mean rate of λ arrivals per frame. The time unit of the D-BMAP arrival process is one frame time.
- 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.
- The Bit Error Rate (BER) is assumed to be zero, an assumption that is relaxed in the next section.

These assumptions are identical to [4], except that we assume D-BMAP arrivals instead of Poisson arrivals. For Poisson arrivals one obtains a Quasi-Birth-Death (QBD) Markov chain [11] by observing the couple $(\hat{q}, \hat{Q})$ at the start of each frame, where $\hat{q}$ represents the number of requests left in the TS that is currently being served (provided that a TS is being served) and $\hat{Q}$ is the number of TSs waiting in the distributed FIFO queue[2]. If we consider the same stochastic process for D-BMAP arrivals and add the current state of the D-BMAP, say $\hat{j}$, the resulting process is no longer a Markov chain, because the number of requests part of a TS depend on the history of the D-BMAP arrival process. Therefore, a different approach is required; the basic idea is to remember the "age" of the TS in service instead of the number of TSs waiting in the TS queue. The state of the system is described by the triple (q, j, Q), where

- $q \geq 2$ denotes the number of requests left in the Transmission Set in service (if there is a Transmission Set in service).
- 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 (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 Set in service was generated ($Q = 0$ 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 4 stations has had at least 1, in one of the $S(+N)$ slots of frame $n-3$, and at most 3 unsuccessful attempts in the previous 3 frames (depending on the service completion time of the previous TS[3]) 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 [4] leads to 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,

[2] More specifically, level zero of the QBD consists of one state that corresponds to the case where there are no TSs waiting in the queue nor being served. Level $i > 0$ consists of multiple states that correspond to the case where there are $i-1$ TSs waiting in the queue, while a TS is being served (the j-th state of level i indicates that there are $j+1$ requests left in the TS).

[3] If the previous TS completed service in frame $n-x$, for $x \leq 3$, then these 4 requests have had an attempt in the S slots of frame $n-3$ and one attempt in one of the N slots in each of the frames $n-x+1, \ldots, n-1$; hence, at most 3 (if $x = 3$). If the previous TS completed service before frame $n-3$, these 4 requests had an attempt in one of the $S+N$ slots of frame $n-3$ and one attempt in one of the N slots of frame $n-2$ and $n-1$.

one obtains the delay distribution from the steady state probabilities by means of a simple formula (see Section 3.5).

3.2 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, for $i \geq 0$, are the $l \times l$ matrices that characterize the input D-BMAP traffic. 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. 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. Hence, the range of q equals $\{q \mid 2 \leq q \leq q_m\}$, while $\{j \mid 1 \leq j \leq l\}$ is the range of j.

During a state transition, Q can never increase by more than one. Therefore, the system can be described by a transition matrix P with a GI/M/1 structure:

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

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 available to the 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 being served 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 are available to the new arrivals in frame $n + 1$).

3.3 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, $q - q'$ requests are successful when a set of x slots is available to transmit the q request packets. We are particularly interested in $p_S(q,q')$, $p_N(q,q')$ and $p_{S+N}(q,q')$. Von Mises [15] has shown, in 1939, that

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

where C^r_s denotes the number of different ways to choose s from r different items.

Next, denote P_N as a $q_m - 1 \times q_m - 1$ 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}, $E^k_S, 2 \leq k \leq q_m$, and $E^k_{S+N}, 2 \leq k \leq q_m$, as (these matrices are $l \times l$ matrices)

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

where the D-BMAP arrival process is characterized by the matrices D_i. Notice that $(E^k_S)_{j,j'}$ represents the probability that a new TS with k requests is generated in a frame where S slots are available to the new arrivals, thus, another TS is currently being served 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 available to the new arrivals. Similar interpretations exist for the matrices F_{S+N} and E^k_{S+N}. The transition probability matrices A_i and B_i are then found as follows[4]:

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

where $\otimes$ denotes the Kronecker product between matrices and I_l is the $l \times l$ unity matrix. Notice that the matrices A_i and B_i decrease to zero with a rate $(F_S)^i$. 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] The notation $[E^2_x \ldots E^{q_m}_x]$ is used to denote a block vector whose i-th block equals E^{i+1}_x, thus, this notation should not be confused with a matrix product

3.4 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

$$\pi_0(j) = \lim_{n\to\infty} \pi_0^n(j), \tag{12}$$

$$\pi_i(q,j) = \lim_{n\to\infty} \pi_i^n(q,j). \tag{13}$$

Define the $1 \times l$ vector $\pi_0 = (\pi_0(1), \ldots, \pi_0(l))$ and the $1 \times l(q_m - 1)$ vectors $\pi_i = (\pi_i(2,1), \ldots, \pi_i(2,l), \pi_i(3,1), \ldots, \pi_i(3,l), \pi_i(4,1), \ldots, \pi_i(q_m,l))$, for $i > 0$. From the transition matrix P (Equation 1) we see that the Markov chain is a generalized Markov chain of the $GI/M/1$ Type [10]. For 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 [10]. 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 [10] 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 an irreducible stochastic matrix, provided that the input D-BMAP is irreducible, and therefore, a simple condition exists to check whether $sp(R) < 1$ [10, 11]. We could also study the stability of FS-ALOHA by noticing that FS-ALOHA, when subject to D-BMAP arrivals, is equivalent to a discrete time MMAP[K]/G[K]/1 queue with a generalized initial condition, where the MMAP[K] stands for a Markov chain with marked arrivals [7]. The stability of such queues has been studied by He [6, Theorem 7.1].

3.5 Calculating the Delay Density Function

Let D be the random variable that denotes the delay experienced by a request packet. We state that $D = 0$ if a request packet is successful during its first transmission attempt. $D = i$ if a request packet is successful in frame $n + i$ provided that the first attempt took place in frame n. The probability that a request has a delay of i frames, can be calculated as the expected number of requests with an "age" of i frames that transmit successfully during an arbitrary frame, divided by the expected number of requests that transmit successfully during an arbitrary frame (that is, λ for a stable system). 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. The probability $P[D = 0]$ is found as $1 - \sum_{i>0} P[D = i]$.

4 FS-ALOHA with D-BMAP arrivals and Memoryless Errors

In this section we relax the assumption on the BER made in the previous section, and allow for memoryless errors to occur. The feedback from the BS is considered error free, a condition that can be realized by protecting the feedback field with a strong error correction code. From a practical point of view, Markovian errors would 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. This is due to the fact that error can occur both in the S and N slots of a frame. Our Markov chain observes the S and N slots of different frames during a state transitions, therefore, one needs to keep track of the entire history of the Markovian environment of the error. 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. We assume an error occurs in a slot with a probability $0 \leq \tilde{e} \leq 1$.

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 interprets, correctly, 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 interpret, incorrectly, 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 error free model of the previous section (recall that $q \geq 2$). 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 a collision 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 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 done for all the matrices or vectors defined in this section).

First, define $\tilde{p}_x^E(q, q')$ as the probability that in a set of q requests, $q - q'$ are successful when a set of x slots is available to transmit the q request packets and at least one error occurs in these x slots. Because the errors are memoryless we have

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

where $p_x(q, q')$ was defined in Section 2 and $\tilde{e}$ represents the probability that an arbitrary slot holds an error. Obviously, we are interested in $\tilde{p}_S^E(q, q')$, $\tilde{p}_N^E(q, q')$ and $\tilde{p}_{S+N}^E(q, q')$.

Next, denote $\tilde{P}_N^E$ as a $(q_m + 1) \times (q_m + 1)$ matrix whose $(i, j)^{th}$ element equals $\tilde{p}_N^E(i - 1, j - 1)$. $\tilde{P}_N$ 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 $(1 - \tilde{e})^N p_N(i - 1, j - 1)$. The $(q_m + 1) \times 1$ vector $\tilde{P}_{N,0}$ has its i^{th} entry equal to

$(1-\tilde{e})^N p_N(i-1,0)$. 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

$$\begin{aligned}
\tilde{F}_S &= \sum_{i\geq 0} D_i \; p_S(i,0) \; (1-\tilde{e})^S & (18)\\
\tilde{F}_{S+N} &= \sum_{i\geq 0} D_i \; p_{S+N}(i,0) \; (1-\tilde{e})^{S+N} & (19)\\
\tilde{E}_S^k &= \sum_{i\geq k} D_i \; \left[1_{\{k>0\}} \; p_S(i,k) \; (1-\tilde{e})^S + \tilde{p}_S^E(i,k)\right], & (20)\\
\tilde{E}_{S+N}^k &= \sum_{i\geq k} D_i \; \left[1_{\{k>0\}} \; p_{S+N}(i,k) \; (1-\tilde{e})^{S+N} + \tilde{p}_{S+N}^E(i,k)\right], & (21)
\end{aligned}$$

where $1_A = 1$ if A is true and 0 otherwise. Notice that $p_x(i,1) = 0$ and 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 available to 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 $\tilde{B}_i$:

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

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

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

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

5 FS-ALOHA++ with Poisson Arrivals and Memoryless Errors

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 Poisson request arrival process with a rate of λ arrivals per frame.
- If there are no Transmission Sets in the queue nor in service, the total $T = S + N$ slots is available to the new arrivals.
- FS-ALOHA++ serves $K \geq 2$ TSs at once provided that K or more Transmission Sets are waiting to be served (otherwise it serves those that are waiting).
- Memoryless bit errors occur on the channel, however, the feedback present in the frame headers is considered error free.

These assumptions are identical to [2, 3], except that we allow for memoryless errors to occur. Ideally we would like to consider D-BMAP arrivals as we did in Sections 3 an 4 for FS-ALOHA. However, so far, we have not managed to obtain an exact model. We have developed a model, presented in Section 6, for D-BMAP arrivals that allows to determine the maximum stable throughput of FS-ALOHA++ and that underestimates the delay suffered by a request packet. Evidence that the margin of underestimation produced by this model is small is presented in Section 9.1.

The approach followed in this section is a generalization of the model in [2, Chapter 6]. Therefore, we skip some of the details when presenting the model. Also, we add a bar to all the vectors and matrices defined in this section. The vector $\tilde{P}_{N,0}$ and the matrices $\tilde{P}_N^E$ and $\tilde{P}_N$, defined in the previous section, will reappear in this section, however, in this section they should be regarded as vectors and matrices of dimension $Kq_m + 1$ instead of $q_m + 1$. Their entries are still calculated using the same formulas.

A GI/M/1 Type Markov chain can be obtained as follows. Define level zero, a single state, as the case where there are no TSs being served nor waiting in the queue and level $i, i > 0$, with $Kq_m + 1$ states, as the case in which there are $i - 1$ TSs waiting in the queue and one to K TSs are being served simultaneously. The q-th state of level $i > 0$ indicates that there are q, $0 \leq q \leq Kq_m$, requests left in the TSs that are currently in service—due to the memoryless errors this number can be zero or one. Let us now consider the different transitions that might occur if we observe the system at the start of each frame. For level $i > 0$ there are two possibilities:

- First, the service does not finish in the current frame. As a result, a transition is made to level i or $i + 1$ depending on whether a new TS is generated in the S slots. The corresponding transition probability matrices are denoted by $\bar{A}_1$ and $\bar{A}_0$.

- Secondly, the service finishes in the current frame. As a result, for $i > K$, the new level is $i - K$ or $i - K + 1$ depending on whether a new TS is generated in the S slots (with corresponding transition probability matrices $\bar{A}_{K+1}$ and $\bar{A}_K$). For $1 < i \leq K$, we get a transition to level one—there are at most $K - 1$ TSs waiting; therefore, a possible new TS generated in the S slots is served simultaneously with all the other waiting TSs. The corresponding transition probability matrix is $\bar{B}_{i,1}$. For $i = 1$, there is a transition to $i = 0$ or $i = 1$ depending on whether a new TS is generated in the S slots (matrices $\bar{B}_{1,0}$ and $\bar{B}_{1,1}$).

For level zero, we remain at level zero if no TS is generated in the $S+N$ slots, otherwise a transition to level one is made (matrices $\bar{B}_{0,0}$ and $\bar{B}_{0,1}$). As a result we have the following transition matrix $\bar{P}$:

$$\bar{P} = \begin{bmatrix} \bar{B}_{0,0} & \bar{B}_{0,1} & 0 & \dots & & & & & & & \\ \bar{B}_{1,0} & \bar{B}_{1,1}+\bar{A}_1 & \bar{A}_0 & 0 & \dots & & & & & & \\ 0 & \bar{B}_{2,1} & \bar{A}_1 & \bar{A}_0 & 0 & \dots & & & & & \\ 0 & \bar{B}_{3,1} & 0 & \bar{A}_1 & \bar{A}_0 & 0 & \dots & & & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & & & \\ 0 & \bar{B}_{K,1} & 0 & \dots & 0 & \bar{A}_1 & \bar{A}_0 & 0 & \dots & & \\ 0 & \bar{A}_{K+1} & \bar{A}_K & 0 & \dots & 0 & \bar{A}_1 & \bar{A}_0 & 0 & \dots & \\ 0 & 0 & \bar{A}_{K+1} & \bar{A}_K & 0 & \dots & 0 & \bar{A}_1 & \bar{A}_0 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{28}$$

where all the matrices are $Kq_m + 1 \times Kq_m + 1$ matrices, except for the 1×1 matrix $\bar{B}_{0,0}$, the $1 \times Kq_m + 1$ matrix $\bar{B}_{0,1}$ and the $Kq_m + 1 \times 1$ matrix $\bar{B}_{1,0}$.

Let $\bar{p}_j$ be the probability that j new requests are generated in a frame, i.e., $\bar{p}_j = \lambda^j / j! e^{-\lambda}$. In order to describe the transition probability matrices, we define the following variables:

$$\begin{aligned} \bar{p}_{gt}(x) &= 1 - (1-\tilde{e})^x \sum_{j=0}^{q_m} \bar{p}_j p_x(j,0), \\ (\bar{p}_{nt}(x))_{1,q} &= \begin{cases} \sum_{j=q-1}^{q_m} \bar{p}_j \left[\tilde{p}_x^E(j,q-1) + 1_{\{q>1\}}(1-\tilde{e})^x p_x(j,q-1)\right] & q \leq q_m + 1 \\ 0 & q > q_m + 1 \end{cases}, \end{aligned}$$

where $p_x(i,j)$ and $\tilde{p}_x^E(i,j)$ were defined in Section 2 and 4, where $\tilde{e}$ is the probability that an error occurs in a slot. Notice, $\bar{p}_{gt}(x)$ is the probability that a TS is generated in a frame provided that x slots are available to the new arrivals (where $x = S$ and $S+N$ are of interest to us). The vector $\bar{p}_{nt}(x)$ is a $1 \times Kq_m + 1$ vector whose q-th component is the probability that a TS with $q-1$ request is generated provided that x slots are available to the new arrivals. We need to define one more vector operation before we can express the transition probability matrices, being

$$\begin{aligned} X^{(n*)} &= X^{(n-1*)}(*)X, \\ X^{(1*)} &= X, \end{aligned}$$

for $n > 1$, where X is a $1 \times Kq_m + 1$ vector and $(*)$ denotes the convolution operation followed by a truncation which keeps the first $Kq_m + 1$ elements. With these notations and definitions we can easily express the transition probability matrices $\bar{B}_{i,j}$ as follows:

$$\begin{aligned} \bar{B}_{0,0} &= 1 - \bar{p}_{gt}(S+N), \\ \bar{B}_{0,1} &= \bar{p}_{nt}(S+N), \\ \bar{B}_{1,0} &= (1-\bar{p}_{gt}(S))\tilde{P}_{N,0}, \\ \bar{B}_{1,1} &= \tilde{P}_{N,0}\bar{p}_{nt}(S), \\ \bar{B}_{i,1} &= \tilde{P}_{N,0}\left[\bar{p}_{gt}(S)\left(\frac{\bar{p}_{nt}(S)}{\bar{p}_{gt}(S)}\right)^{(i*)} + (1-\bar{p}_{gt}(S))\left(\frac{\bar{p}_{nt}(S)}{\bar{p}_{gt}(S)}\right)^{(i-1*)}\right], \end{aligned}$$

where $2 \leq i \leq K$. Similarly, for $\bar{A}_i$, we find

$$\begin{aligned}
\bar{A}_0 &= \bar{p}_{gt}(S)\left(\tilde{P}_N^E + \tilde{P}_N\right), \\
\bar{A}_1 &= (1-\bar{p}_{gt}(S))\left(\tilde{P}_N^E + \tilde{P}_N\right), \\
\bar{A}_K &= \bar{p}_{gt}(S)\tilde{P}_{N,0}\left(\frac{\bar{p}_{nt}(S)}{\bar{p}_{gt}(S)}\right)^{(K*)}, \\
\bar{A}_{K+1} &= (1-\bar{p}_{gt}(S))\tilde{P}_{N,0}\left(\frac{\bar{p}_{nt}(S)}{\bar{p}_{gt}(S)}\right)^{(K*)}.
\end{aligned}$$

We denote $\bar{\pi}_0$, resp. $\bar{\pi}_i(q)$ for $i > 0$, the steady state probability related to level zero, resp. state q of level i. These probabilities are found by means of an iterative scheme similar to the one described in Section 3.4. Next, we indicate how to obtain the delay distribution $\bar{D}$ of a request packet from the steady state probabilities. We separate the following two cases. First, a request could be successful during its first attempt, in which case the delay $\bar{D}$ equals zero, this happens with probability

$$P[\bar{D}=0] = (1-\tilde{e})\sum_{j=0}^{q_m-1}\bar{p}_j\left[\bar{\pi}_0(1-\frac{1}{S+N})^j + (1-\bar{\pi}_0)(1-\frac{1}{N})^j\right], \tag{29}$$

because Poisson arrivals see time averages (PASTA). Otherwise, if a request is unsuccessful during its first transmission, we subdivide its delay, as in [2], into three parts:

- First, we have the time required for the completion of the TSs that were (still) in service when the arbitrary request made its first transmission attempt, denoted as t_A.

- Second, assuming there are G TSs waiting in the FIFO queue at the arrival time of our arbitrary request, we have the delay associated to $K\lfloor G/K \rfloor$ TSs that are served by placing K TSs in the server at a time. This delay is denoted by t_B.

- Finally, t_C entails the time between the start of the service of the TS holding our arbitrary request and its successful transmission. Notice, this TS may be served simultaneously with 1 to $K-1$ other TSs.

If the arbitrary request, that was unsuccessful during its first transmission attempt, arrived when the Markov chain was at level zero, its delay is $1 + t_C$, otherwise it equals $t_A + t_B + t_C$. Next, we briefly describe how t_A, t_B and t_C are calculated.

The delay $t_A \geq 1$ depends upon q, the number of requests left in service at the arrival time of our arbitrary request and is found as

$$P[t_A(q) > r] = \left[\left(\tilde{P}_N + \tilde{P}_N^E\right)^r 1_e\right]_{q+1,1}, \tag{30}$$

where $r \geq 0$, 1_e is a column vector of ones and $X_{i,1}$ denotes the i-th element of a column vector. Let $P[t_A(q) = r]$ be the $(q+1, r+1)^{th}$ element of the matrix $\bar{F}_A$.

The delay $t_B \geq 0$ depends upon G, the number of TSs in the FIFO queue at the arrival time of our arbitrary request and is calculated as follows. For $G < K$, $t_B(G)$ equals zero, whereas for $K \leq G < 2K$

we find

$$P[t_B(G) = r] = \left[\left(\frac{\bar{p}_{nt}(S)}{\bar{p}_{gt}(S)} \right)^{(K*)} \bar{F}_A \right]_{1,r+1} . \tag{31}$$

Finally, $t_B(G)$ for $G \geq 2K$ is found as the convolution of $t_B(G-K)$ and $t_B(K)$.

Because t_A and t_B are independent, the delay $t_A(q) + t_B(G)$, denoted as $t_{A+B}(q,G)$, is found as the convolution of $t_A(q)$ and $t_B(G)$. Finally, the delay t_C depends upon G and the length l of $t_A + t_B$.

$$P[t_C(G,l) > r] = \sum_{s \geq 1} P[\bar{p}_{tag}(G,l) = s] \left[\left(\bar{P}_N^T \right)^{r+1} 1_e \right]_{s,1} , \tag{32}$$

where the $(i,j)^{th}$ element of the $Kq_m \times Kq_m$ matrix $\bar{P}_N^T$ equals $\frac{j}{i}$ times the $(i+1,j+1)^{th}$ element of $\tilde{P}_N + \tilde{P}_N^E$. $P[\bar{p}_{tag}(G,l) = s]$ represents the probability that $s-1$ other requests are served simultaneously with our arbitrary request, that was unsuccessful during its first transmission attempt, provided that $t_A + t_B = l$ and that G TSs were in the FIFO queue at the arrival time of our request. Also, when calculating the probability $P[p_{tag}(G,l) = s]$, one should notice that the distribution of number of requests, part of the TS of our arbitrary request, depends on whether our request arrived when the Markov chain was at level zero, in which case the new arrivals used $S+N$ slots, or not. The calculation of $\bar{p}_{tag}(G,l)$ is a generalization of [2]. Having found the distribution of $t_{A+B}(q,G)$ and $t_C(G,l)$, we can easily calculate $P[\bar{D} = i]$, for $i > 0$, by remarking that Poisson arrivals see time averages (PASTA).

6 FS-ALOHA++ with D-BMAP Arrivals and Memoryless Errors

In this section, we present an analytical model that allows us to derive a lower bound of the delay experienced by a request packet under the same conditions as in Section 5, with the generalization of D-BMAP arrivals (instead of Poisson arrivals). This model approximates the delay as follows:

- For request packets belonging to a TS that is served simultaneously with $K-1$ other TSs, the approximated delay is the sum of two components. The first component equals the time that elapses between the moment that the youngest of these K TSs was created and the moment that the service of these K TSs starts. The second component equals the time that elapses between the start of the service and the moment at which the request packet is transmitted successfully.

- For packets belonging to a TS that is served simultaneously with less than $K-1$ other TSs, the approximated delay is one frame plus the time that elapses between the start of the service and the moment at which the request packet is transmitted successfully.

Hence, the approximated delay provides a low bound. Next, a GI/M/1 Type Markov chain, similar to the one considered in Section 4, is defined and the approximated delay is easily found from its steady state probabilities. The vector $\tilde{P}_{N,0}$, resp. the matrices A_0 and B_i, defined in the previous sections, will reappear in this section, however, their dimension is now $Kq_m + 1$, resp. $(Kq_m + 1)l$.

As in Section 4, level zero of the Markov chain contains l states, where l is the number of states that the D-BMAP has, and state $1 \leq j \leq l$ corresponds to the situation in which there are no TSs in service (nor waiting) and the state of the D-BMAP at the start of the current frame is j. Thus, the full $T = S+N$ slots are available to the new arrivals. Level $i > 0$, with $(Kq_m + 1)l$ states, corresponds to either one of the following two situations:

- First, the Markov chain is at level $i > 0$, in state (q, j), if there are currently K TSs, the youngest of which was generated i frames ago, being served simultaneously, if $0 \leq q \leq Kq_m$ requests remain in service in the current frame, say frame n, and if $1 \leq j \leq l$ was the state of the D-BMAP corresponding to the frame following the one in which the youngest of the K TSs in service was generated, that is, frame $n - i + 1$. An example scenario for $i = 5$ is presented in Figure 2.
- Second, state (q, j) at level i also corresponds to the situation in which $1 \leq s < K$ TSs are served simultaneously, the service of these s TSs started $i - 1$ frames ago, there are q requests remaining in service in the current frame, say frame n, and the D-BMAP state corresponding to frame $n - i + 1$ is j. An example scenario for $i = 4$ is presented in Figure 3.

As far as the transition probabilities are concerned, it does not matter whether the first or the second situation applies.

Similar to Section 4, we find that the transition matrix has the following structure:

$$\breve{P} = \begin{bmatrix} \breve{B}_1 & \breve{B}_0 & 0 & 0 & 0 & \dots \\ \breve{B}_2 & \breve{C}_1 + \breve{A}_1 & \breve{A}_0 & 0 & 0 & \dots \\ \breve{B}_3 & \breve{C}_2 + \breve{A}_2 & \breve{A}_1 & \breve{A}_0 & 0 & \dots \\ \breve{B}_4 & \breve{C}_3 + \breve{A}_3 & \breve{A}_2 & \breve{A}_1 & \breve{A}_0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{33}$$

where $\breve{B}_1$ is an $l \times l$ matrix, $\breve{B}_0$ an $l \times (Kq_m + 1)l$ matrix, $\breve{B}_i$ for $i > 1$ a $(Kq_m + 1)l \times l$ matrix, while the others are $(Kq_m + 1)l \times (Kq_m + 1)l$ matrices. Moreover, from the probabilistic interpretations provided, it should be clear that the matrices $\breve{B}_i$ for $i > 0$ are equal to $\tilde{B}_i$, as defined in Section 4. Moreover, the matrix $\breve{A}_0$ equals $\tilde{A}_0$. The matrices $\breve{A}_i$, for $i > 0$, hold the probabilities that the current service finishes, that K TSs are generated in an interval of i frames and the last of these K TSs was generated in the i^{th} frame of the interval. The matrices $\breve{C}_i$ on the other hand, hold the probabilities that the current service finishes and that less that K TSs are generated in i frames.

In order to calculate the matrices $\breve{A}_i$ and $\breve{C}_i$ we define the following $l \times l$ matrices $\breve{E}_{S,i}^{k,n}$, where $i \geq 1, 0 \leq k \leq Kq_m$ and $1 \leq n \leq K$:

$$\begin{aligned} \breve{E}_{S,i}^{k,1} &= \left(\tilde{F}_S\right)^{i-1} \tilde{E}_S^k, \\ \breve{E}_{S,i}^{k,n} &= \sum_{l=0}^{k} \sum_{j=n-1}^{i-1} \breve{E}_{S,j}^{l,n-1} \breve{E}_{S,i-j}^{k-l,1}, \end{aligned}$$

where $1 < n \leq K$ and the matrices $\tilde{F}_S$ and $\tilde{E}_S^k$ as defined in Section 4. Looking at the probabilistic interpretations of these matrices, it is clear that

$$\breve{A}_i = \tilde{P}_{N,0} \otimes \left[\breve{E}_{S,i}^{0,K} \dots \breve{E}_{S,i}^{Kq_m,K}\right], \tag{34}$$

for $i > 0$. Next, let the $l \times l$ matrices $\breve{E}^k_{S,i}$ be defined as

$$\breve{E}^k_{S,i} = \sum_{1 \leq j \leq i} \left(\breve{E}^{k,1}_{S,j} + \ldots + \breve{E}^{k,K-1}_{S,j}\right) \left(\tilde{F}_S\right)^{i-j}. \tag{35}$$

Using these matrices and their probabilistic interpretation, we can express the matrices $\breve{C}_i$ as

$$\breve{C}_i = \tilde{P}_{N,0} \otimes \left[\breve{E}^0_{S,i} \ldots \breve{E}^{Kq_m}_{S,i}\right], \tag{36}$$

for $i > 0$. The steady state probabilities $\breve{\pi}_0$ and $\breve{\pi}_i(q,j)$, for $i > 0$, are calculated in a similar way as before and the approximated delay $\breve{D}$, for $i > 0$, is found as

$$P[\breve{D} = i] = \sum_{q=0}^{Kq_m} \frac{(1-\tilde{e})(1-1/N)^{q-1} q}{\lambda} \sum_{j=1}^{l} \breve{\pi}_i(q,j), \tag{37}$$

and $P[\breve{D} = 0]$ equals $1 - \sum_{i>0} P[\breve{D} = i]$. Notice, if the D-BMAP arrival process is a Poisson process, $P[\breve{D} = 0]$ is equal to $P[\bar{D} = 0]$. Evidence that the lower bound for the delay is close to the exact delay is presented in Section 9.1.

7 FS-ALOHA(++) and the Capture Effect

In radio channels, a collision of two (or more) requests does not necessarily destroy both requests. Because of signal fading, requests from different transmitting stations can arrive with very different power levels, and the strongest request may survive. This situation, in which one of the requests involved in a collision is nevertheless successfully received, is termed capture. The surviving request is said to have captured the other request(s) with which it collided. When capture occurs, clearly, the BS receives the successful request, but does not perceive the captured requests. However, the feedback from the BS allows the captured requests to realize that they have been captured, because the BS adds the address of the successful packet to the feedback field. As a result, captured requests will be retransmitted in the next frame. Notice, with FS-ALOHA(++), the service ends whenever the N slots dedicated to the service, as perceived by the BS, do not hold a collision (nor an error). Thus, because the BS considers a capture event as a success, the service could end before all requests belonging to the TSs in service have been successfully transmitted. As a result, requests part of one TS might be obliged to join the next TS (if there is a next one, otherwise they will be retransmitted as if they were new requests).

The theoretically unlimited shifting of requests between successive TSs, due to the capture effect, is an undesirable effect. Indeed, if only two requests are left in service, without capture both would be successful in the very near future. However, due to the capture, one of these requests might not be successful for quite some time. Thus, we can expect that capture increases the worst case delays. There is another argument that supports this idea. Capture causes requests to shift between TSs, thus the FCFS order of FS-ALOHA(++) is somewhat damaged. With respect to the delay variation of a queueing system, it is generally known that a FCFS service discipline guarantees the lowest delay variation (whereas LCFS the worst). Thus, any process that deranges the FCFS order, increases the delay variation.

To avoid these negative effects due to capture, we could slightly modify FS-ALOHA(++) in such a way that more capture tends to improve the delay characteristics. Although, we are slightly adapting FS-ALOHA(++), we do not assign a new name to the algorithm. The first modification goes as follows. Instead of ending the service of a TS when the N slots dedicated to its service do not hold a collision (nor an error), we end the service if these N slots are all error free and empty. As a result, a capture event cannot occur in the last frame of the service and all the requests belonging to the TSs that are being served remain part of their TS.

Obviously, capture can also occur in the S (or $S+N$) slots dedicated to the new arrivals. The approach used by such requests is referred to as the second modification. One way to deal with this is to generate a TS unless all the slots used by the new arrivals are error free and empty. This turns out to be a bad approach, because it causes FS-ALOHA to become unstable even for Poisson arrivals with a mean rate below 1.5 requests per frame, a moderate error rate ($e = 1/50$), $T = 10$ contention slots and no capture. The unstability is formally observed by adapting the GI/M/1 Type Markov chain of Section 4, in order to incorporate both modifications. Therefore, we suggest that a request that is captured when transmitted in a slot that is available to the new arrivals, uses the following strategy: If the feedback provided by the BS indicates that a TS is formed, the captured request simply joins the TS as if it was part of a capture free collision. Otherwise, the request will reattempt transmission in the slots dedicated to the new arrivals in the next frame.

Without capture occurring, the modified FS-ALOHA(++) performs worse than the original one. In this section, we present a model that allows us to assess the magnitude of this penalty. Only a few minor changes are required to the GI/M/1 Type Markov chain presented in Sections 4 and 5, to evaluate FS-ALOHA(++) under the following conditions:

- The modified version of FS-ALOHA under D-BMAP arrivals, memoryless errors and no capture.
- The modified version of FS-ALOHA++ under Poisson arrivals, memoryless errors and no capture.

Due to the modifications made to FS-ALOHA(++), we can expect capture to improve the delay characteristics. Indeed, a capture event in the N slots of the modified version of FS-ALOHA(++) will always reduce the time needed to complete service, as well as the request packet delays. A capture event in the S slots could slightly increase the delay of a request, however, this increase is believed to be minor. As a result, the "no capture" scenario can, to a certain extent, be considered a worst case.

First, define $\tilde{P}_N^*$ as a $(K)q_m + 1 \times (K)q_m + 1$ matrix whose $(i,j)^{th}$ element, for $(i,j) \neq (1,1)$, equals $p_N(i-1, j-1)$ and whose $(1,1)^{th}$ equals zero. The i^{th} component of the $(K)q_m + 1 \times 1$ vector $\tilde{P}_{N,0}^*$ represents the probability that a TS ends provided that $i-1$ requests are still competing. Thus, due to the first modification to FS-ALOHA(++), we find

$$\tilde{P}_{N,0}^* = \begin{bmatrix} (1-\tilde{e})^N \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{38}$$

Replacing the matrix $\tilde{P}_N$ and the vector $\tilde{P}_{N,0}$ by $\tilde{P}^*_N$ and $\tilde{P}^*_{N,0}$ in Section 4 and 5, allows us to calculate the delay distribution under both above-mentioned conditions. To further clarify this, we express the transition probabilities for FS-ALOHA:

$$\begin{aligned}
\tilde{A}^*_0 &= (\tilde{P}^*_N + \tilde{P}^E_N) \otimes I_l, && (39)\\
\tilde{A}^*_i &= \tilde{P}^*_{N,0} \otimes (\tilde{F}_S)^{i-1} \left(\tilde{E}^0_S \; \tilde{E}^1_S \; \ldots \; \tilde{E}^{q_m}_S \right), && (40)\\
\tilde{B}^*_0 &= \tilde{B}_0 = \left(\tilde{E}^0_{S+N} \; \tilde{E}^1_{S+N} \; \ldots \; \tilde{E}^{q_m}_{S+N} \right), && (41)\\
\tilde{B}^*_1 &= \tilde{B}_1 = \tilde{F}_{S+N}, && (42)\\
\tilde{B}^*_i &= \tilde{P}^*_{N,0} \otimes (\tilde{F}_S)^{i-1}. && (43)
\end{aligned}$$

Notice, only the first l rows of $\tilde{A}^*_i$, for $i > 0$, and $\tilde{B}^*_i$, for $i > 1$, differ from zero. This observation does not reduce the computation time of the R matrix significantly, because the iterations are done using Horner's scheme and the R matrix is dense.

8 Numerical Results for FS-ALOHA

In this section we explore the influence of correlation, burstiness, the number of contention slots T, memoryless errors and capture on the delay distribution of a request packet using FS-ALOHA by means of the analytical models presented in Section 3, 4 and 7. Thus, new requests are either generated by a Poisson arrival process, or by a D-BMAP arrival process, allowing us to incorporate correlation and burstiness into the traffic stream. The computation times for each of the delay curves presented varies between a few seconds (Poisson arrivals) and a few minutes (D-BMAP arrivals) on a Sun Ultra Enterprise 2170 with two 167 Mhz processors and 3x128 Mbyte RAM. Unless Poisson arrivals are considered, we use the following M state Markov chain to model the arrival process. The number of new requests generated in a frame, when the D-BMAP is in state j, is distributed binomially with parameters (jm, β), where m and β are parameters of the D-BMAP. Transitions between these M states can occur at the end of each frame according to the following $M \times M$ transitions matrix P_M:

$$P_M = \begin{bmatrix} 1-\alpha^+ & \alpha^+ & 0 & \ldots & 0 \\ \alpha^- & 1-\alpha^+-\alpha^- & \alpha^+ & \ldots & 0 \\ 0 & \alpha^- & 1-\alpha^+-\alpha^- & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1-\alpha^- \end{bmatrix}. \qquad (44)$$

Thus, the arrival process is characterized by the following five parameters: M, m, β, α^- and α^+. Higher values of β increase the arrival rate, while decreasing α^- and α^+ results in a stronger correlated arrival process. In this section, the parameters M and m are fixed at 6 and 5, unless stated otherwise, whereas the parameter β is set such that the 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). Finally, it should be clear that this arrival process is an M-state D-BMAP that is characterized by the $M \times M$ matrices $D_0, \ldots, D_{Mm}$. The $(j_1, j_2)^{th}$ entry of D_i equals $(P_M)_{j_1,j_2} \; C^{mj_1}_i \; \beta^i \; (1-\beta)^{mj_1-i}$.

8.1 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 3. For the D-BMAP arrivals we fix $\alpha^+ = \alpha^- = 1/5$, therefore, the mean sojourn time in a state is small, i.e., 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 by (S, N). The results are presented in Figure 4.

A first, obvious, observation in Figure 4 is that the delays are larger for D-BMAP arrivals. This is easily explained by noticing that the mean arrival rate for Poisson arrivals is fixed at 2, whereas for the D-BMAP arrivals we have time periods were the mean arrival rate is as low as $2/3.5 = 4/7$, i.e., when the arrival process is in state one, and time periods were the mean arrival rate is as high as $24/7$, i.e., when the arrival process is in state $M = 6$. A second observation is that the delay distribution decays exponentially[5], 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 [4], it was shown that, for Poisson arrivals, the best delays are obtained with $S \approx N$. Figure 4 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.

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

Apart from checking whether the engineering rule concerning the number of S and N slots as mentioned in the previous section, still applies, this section addresses the issue whether it is worth implementing parallel instances of FS-ALOHA in the contention 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, each one using 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[6]. The use of parallel instances implies that we have T_2 distributed FIFO 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 5 presents the results for $T = S + N = 5$ and $T = 15$ contention slots[7]. 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 N. 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

[5] 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.

[6] 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 coincide.

[7] It should be noted that, provided that the arrivals occur uniformly in a frame, we can evaluate the performance of multiple instances 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.

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.

8.3 Correlation and Burstiness

In this section we study the influence of the mean sojourn time related to the M D-BMAP states on the delay distribution. Notice, longer sojourn times imply a stronger correlated and more bursty arrival process. We use the D-BMAP defined in Section 8 and start by setting $\alpha^+ = \alpha^-$ large and decrease both gradually. Recall, the mean sojourn time in a state is $\alpha^+/2$ frames. The results presented in Figure 6 indicate that the grouping strategy works well in limiting the delay increase due to the augmented correlation and burstiness. The sojourn time has only a limited influence on the delay in the right plot, because the mean arrival rate associate to state $M = 2$ equals 2.666 requests per frame, which is well below the maximum stable throughput for Poisson arrivals [2, 3]. The mean arrival rate associated to state $M = 6$ in the left plot is 3.429, which is very close to the maximum stable throughput; hence, the large delays for $\alpha^+ = \alpha^-$ small.

8.4 Errors on the Channel

In this section we investigate the influence of errors on the channel by means of the model presented in Section 4. We start by setting $\tilde{e}$, the probability that a slot holds an error, equal to $1/250, \ldots, 1/15$. It is hard to state whether such a value of $\tilde{e}$ 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 [13]. For a wired channel it is safe to say that $\tilde{e} = 1/250$ is very pessimistic. We start by reproducing Figure 4 for $\tilde{e} = 0, 1/250, \ldots, 1/15$ and $S = N = 5$. Numerical experiments, not reported here, have shown 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 7, where the curves for $\tilde{e} = 0$ where obtained with the model in Section 3. A first, obvious observation is that the delay increases with increasing $\tilde{e}$. 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 $\tilde{e} \leq 1/50$, errors can seriously increase the delay for higher error rates $\tilde{e}$. Therefore, if the modulation scheme, error codes, signal-to-noise ratio, ... cannot guarantee an error rate $\tilde{e}$ less than $1/5T$, the performance of FS-ALOHA might degrade drastically.

This rule is confirmed by Figure 8, where we study FS-ALOHA for $T = 5$ and 15. For $T = 15$ the Markov chain becomes transient for $\tilde{e} \geq 1/20$ (actually, the chain becomes unstable for $\tilde{e}$ somewhere between 1/20 and 1/21). For Poisson arrivals and $T = 15$ we get instability for $\tilde{e} \geq 1/19$, 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 towards errors.

8.5 FS-ALOHA and Capture Events

In Section 7, we introduced a modified version of FS-ALOHA, designed to avoid requests to switch between TSs. In this section, a comparison between the original FS-ALOHA and this modified version is made. We consider both Poisson and D-BMAP arrivals, different error rates (these are included in the figures) and assume no capture. Notice, for the modified version, the "no capture" scenario is believed to be close to a worst case, that is, including capture is expected to improve the tail of the distribution.

Figure 9 shows that the delay increase of the modified version, compared to the original FS-ALOHA algorithm, is substantial. Thus, the modified version of FS-ALOHA should not be implemented. This is somewhat expected, because high delays are not caused by a few TSs with a high number of requests, but are a consequence of a high number of TSs with few competitors. In case of no capture, using the modified version increases the service time of each TS by one frame, and this additional frame causes, in its turn, a larger backlog in the FIFO queue. One can expect this to improve with the modified version of FS-ALOHA++, because the penalty of the modified FS-ALOHA++ algorithm is, approximately, one frame per K TSs.

9 Numerical Results for FS-ALOHA++

The main opbjective of this section is to demonstrate, by means of the models presented in Section 5, 6 and 7, that FS-ALOHA++, with $K = 2$ or 3, improves the robustness of FS-ALOHA. The best results for FS-ALOHA++ with Poisson arrivals and $\tilde{e} = 0$, are found when $SK \approx N$ [2, 3]. Numerical experiments, not included in this paper, have indicated that this remains true if $\tilde{e} > 0$. Figure 10 reproduces the left plot of Figure 7 for $K = 2$ and 3, by means of the model presented in Section 5. By comparing these figures, it should be clear that FS-ALOHA++ performes much better when subjet to high error rates $\tilde{e}$. Serving TSs simultaneously improves the robustness of the algorithm with respect to errors, because high error rates generate a substantial number of empty TSs, i.e., TSs holding zero requests, and each of these empty TS requires one or more frames to be served when FS-ALOHA is employed. Whenever such an empty TSs is served simultaneously with some other TS, FS-ALOHA++ gains at least one frame in comparison with FS-ALOHA.

9.1 FS-ALOHA++ and D-BMAP arrivals

In Section 6 we presented a model that allowed us to find a lower bound for the delay of a request packet when FS-ALOHA++ is subject to D-BMAP arrivals and memoryless errors. Poisson arrivals are a special case of D-BMAP arrivals, thus, we can compare the results for Poisson arrivals using the model of Section 5. Afterwards, we present some intuitive arguments that indicate why a similar lower bound can be expected for the D-BMAP arrival process defined in Section 8. The results are presented in Figure 11. They indicate that the lower bound is fairly close to the actual delay for different parameter settings. Moreover, Figure 11 indicates that we get a better approximation with higher error rates $\tilde{e}$ and arrivals rates λ. Finally, as expected, smaller K values result in a better approximation.

We can expect a similar, or perhaps even better, approximation for the D-BMAP arrival process defined in Section 8, because the worst delays for D-BMAP arrivals are caused by the time periods with a high mean arrival rate and the higher the arrival rate, the better the approximation. The following D-BMAP arrival process is used: $M = 2, m = 15$ and β is chosen such that $\lambda = 0.2T$. α^+ and α^- are set at $1/5$ and are gradually decreased to $1/1000$. The parameters M and m are changed, when compared to the previous sections, to reduce the memory requirements of the analytical model. The rather high memory requirements are caused by the fact that the $KM^2m+M \times KM^2m+M$ matrices $\breve{A}_i$ decrease rather slowly to zero. By keeping $Mm = 30$ and reducing M, we have significantly reduced the memory requirements. The results are presented in Figure 12. When compared with the right plot in Figure 6, it is fair to say that FS-ALOHA and FS-ALOHA++ perform similar as far as dealing with bursty and correlated arrivals is concerned.

9.2 FS-ALOHA++ and Capture Events

We introduced a modified version of FS-ALOHA(++) in Section 7. Section 8.5 demonstrated that the modified version of FS-ALOHA is unable to guarantee good delay bounds. Moreover, the algorithm was very sensitive to errors (see Figure 9). The modified version of FS-ALOHA++ on the other hand is capable of guaranteeing delay bounds and is much more robust with respect to errors. This is shown in Figure 13 for Poisson arrivals ($\lambda = 2$) and $K = 2$ and 3. Recall, the "no capture" scenario was considered a worst case scenario for the modified algorithm, whereas capture could easily increase the worst case delays of the original algorithm (see Section 7). Based on these and previous results, it is fair to say that the modified FS-ALOHA++ algorithm, where $K = 3$ outperforms $K = 2$, possesses good robustness properties with respect to correlated and bursty arrivals, errors and capture.

10 Conclusions

The robustness of the contention resolution algorithm FS-ALOHA(++) has been studied in this paper with respect to errors, capture and Markovian arrivals. Several analytical models were developed by means of matrix analytical methods, allowing us to calculate the delay distribution of a request packet under different circumstances. Prior work on FS-ALOHA has been limited to Poisson arrivals and error and capture free channels [2, 3, 4]. A variety of numerical examples has shown that both FS-ALOHA and FS-ALOHA++ are capable of dealing with correlated and bursty arrivals. However, the performance and stability of FS-ALOHA becomes troublesome for the high error rates and is unsure when subject to capture events. FS-ALOHA++ was shown to be more robust towards errors. FS-ALOHA(++) was also slightly modified to eliminate possible negative effects of capture. However, the modified version of FS-ALOHA performs very poor and should not be used in practice. The modified version of FS-ALOHA++ on the other hand performs much better and seems suitable for practical purposes. Finally, it was concluded that implementing multiple instances of FS-ALOHA(++) presents an attractive tradeoff between the suffered delay and the implementation costs.

References

[1] 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.

[2] D. Vázquez Cortizo. *Design and Analysis of MAC protocols for Wireless LAN*. PhD thesis, University of Antwerp, May 2000.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[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] 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.

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

[12] M.F. Neuts. *Structured Stochastic Matrices of M/G/1 type and their applications.* Marcel Dekker, Inc., New York and Basel, 1989.

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

[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.

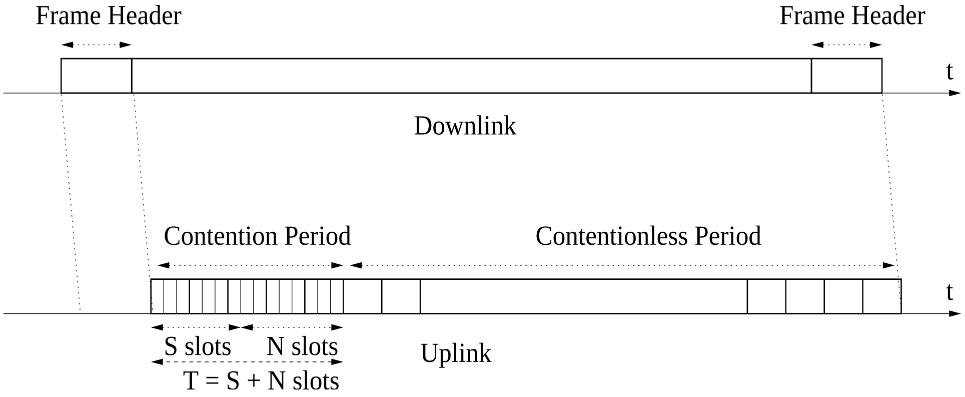

Figure 1: *Frame Structure*

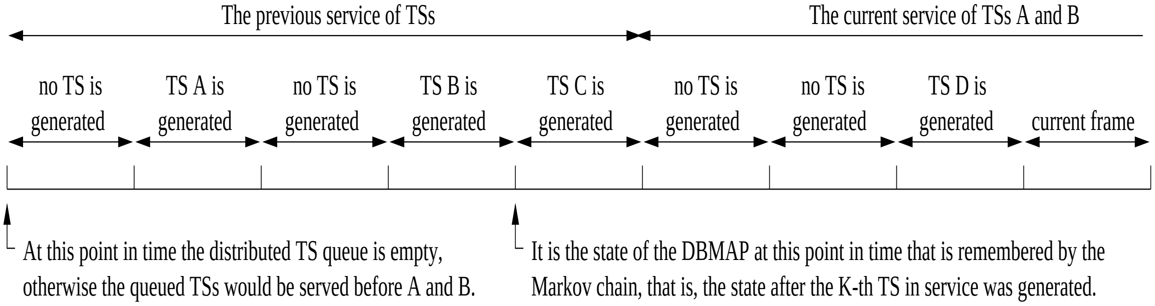
The previous service of TSs
The current service of TSs A and B
no TS is generated
TS A is generated
no TS is generated
TS B is generated
TS C is generated
no TS is generated
no TS is generated
TS D is generated
current frame
At this point in time the distributed TS queue is empty, otherwise the queued TSs would be served before A and B.
It is the state of the DBMAP at this point in time that is remembered by the Markov chain, that is, the state after the K-th TS in service was generated.

Figure 2: *Example scenario for $K = 2$ where the Markov chain is at level $i = 5$.*

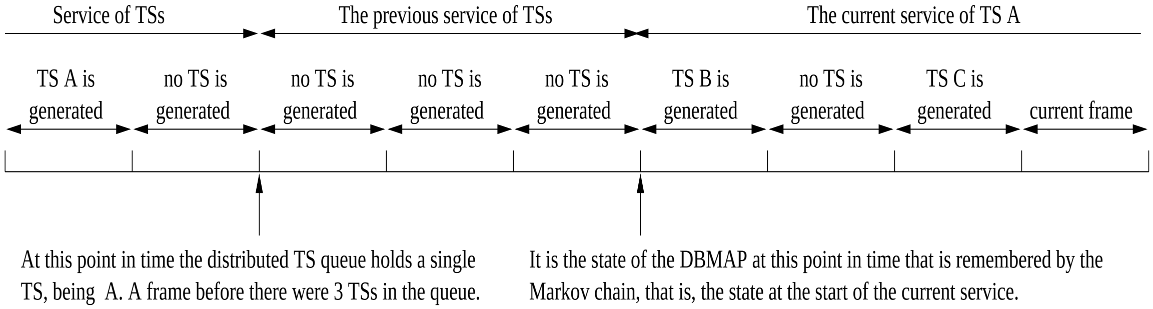
Service of TSs
The previous service of TSs
The current service of TS A
TS A is generated
no TS is generated
no TS is generated
no TS is generated
no TS is generated
TS B is generated
no TS is generated
TS C is generated
current frame
At this point in time the distributed TS queue holds a single TS, being A. A frame before there were 3 TSs in the queue.
It is the state of the DBMAP at this point in time that is remembered by the Markov chain, that is, the state at the start of the current service.

Figure 3: *Example scenario for $K = 2$ where the Markov chain is at level $i = 4$.*

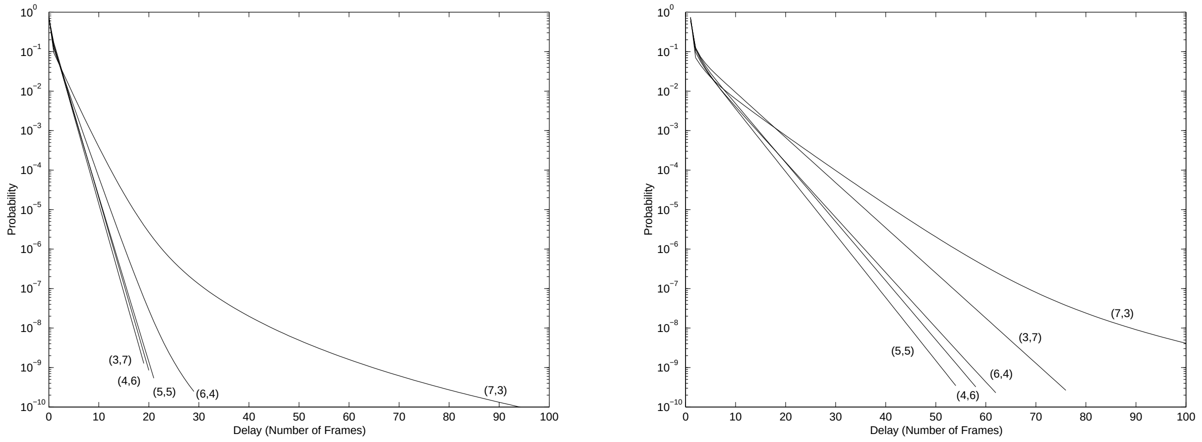

Figure 4: *For $T = 10$, Left: Poisson arrivals ($\lambda = 2$), Right: D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^+ = \alpha^- = 1/5$ and β such that the arrival rate $\lambda = 2$).*

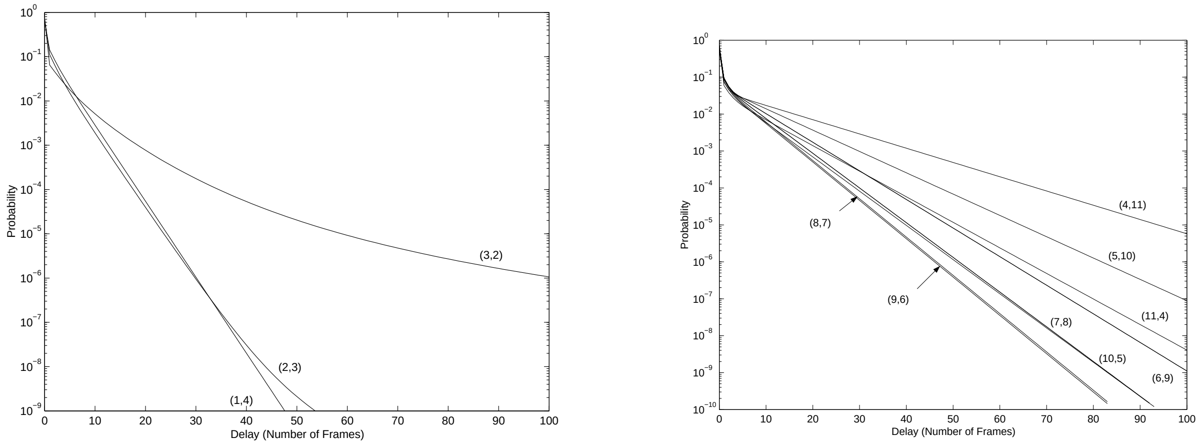

Figure 5: *D-BMAP arrivals ($M = 6$, $m = 5$, $\alpha^+ = \alpha^- = 1/5$), Left: $T = 5$ and β such that $\lambda = 1$, Right: $T = 15$ and β such that $\lambda = 3$.*

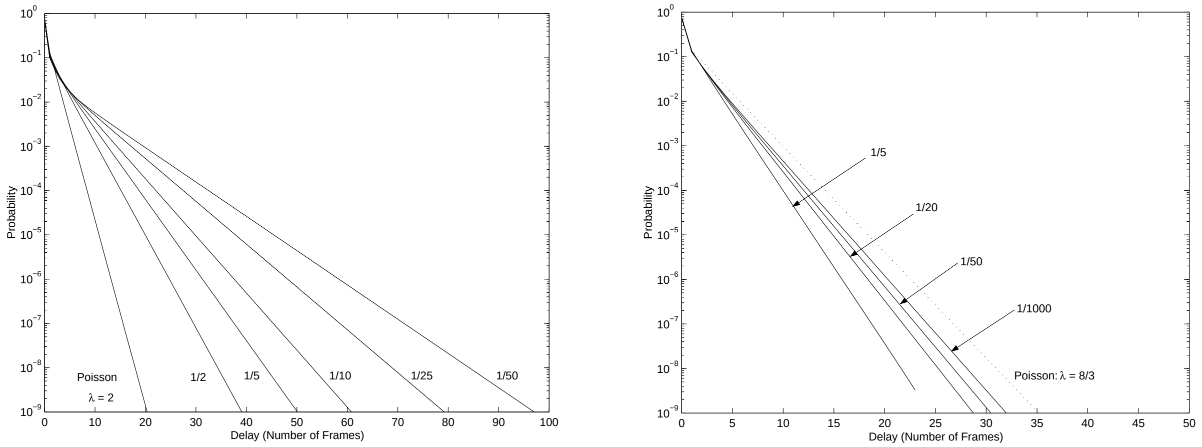

Figure 6: *D-BMAP arrivals, β such that $\lambda = 2$, $T = 10, S = N = 5$, Left: $m = 5, M = 6$, Right: $m = 15, M = 2$*

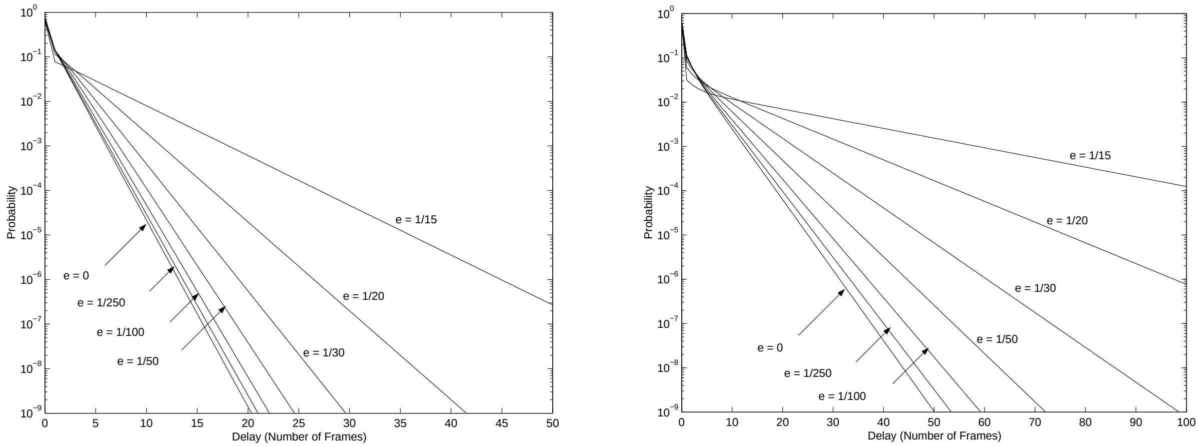

Figure 7: *For* $T = 10, S = N = 5$ *and* $\tilde{e} = 0, 1/250, \ldots, 1/15$ *, Left: Poisson arrivals* ($\lambda = 2$)*, Right: D-BMAP arrivals* ($M = 6$*,* $m = 5$*,* $\alpha^+ = \alpha^- = 1/5$ *and* β *such that the arrival rate* $\lambda = 2$)*.*

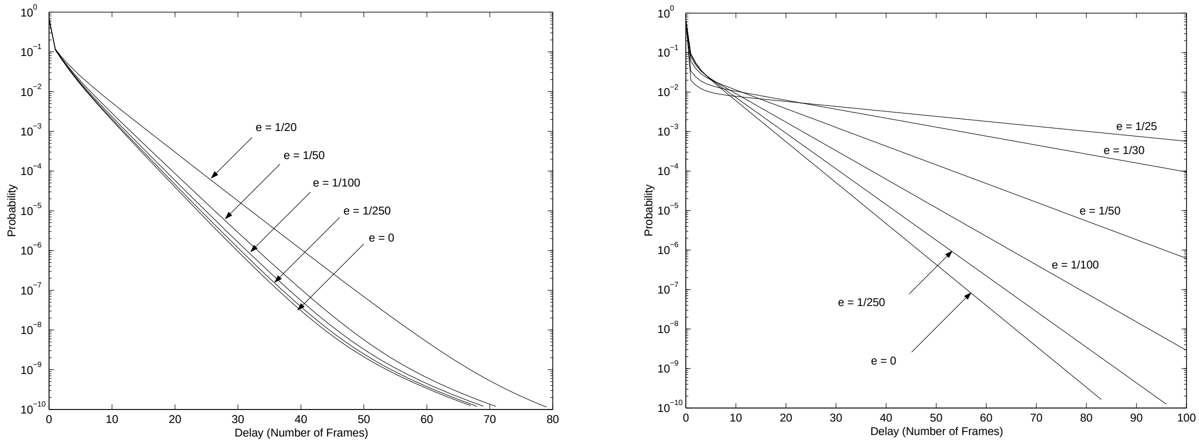

Figure 8: *For D-BMAP arrivals* ($M = 6$*,* $m = 5$*,* $\alpha^+ = \alpha^- = 1/5$ *and* β *such that the arrival rate* $\lambda = 0.2T$)*,* $\tilde{e} = 0$ *to* $1/20$ *, Left:* $T = 5, S = 2, N = 3$*, Right:* $T = 15, S = 8, N = 7$*.*

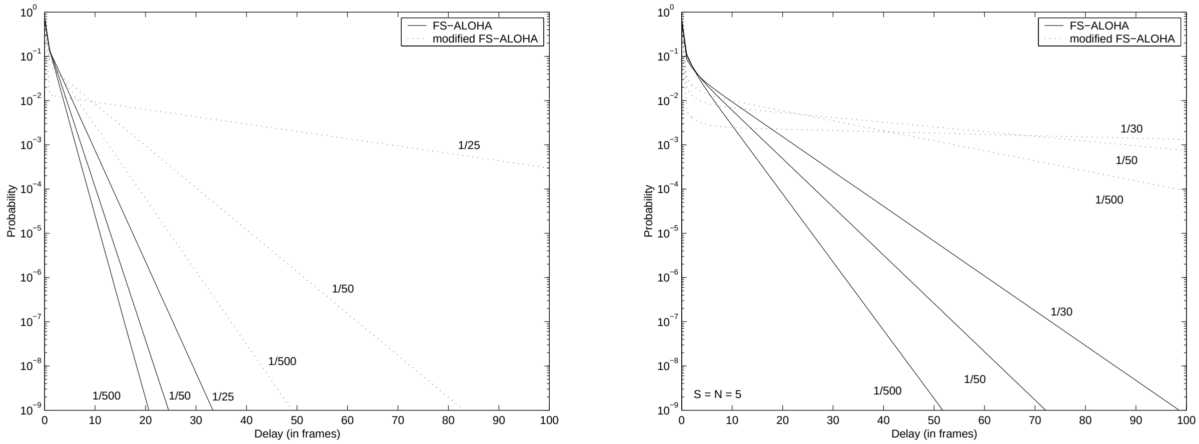

Figure 9: $S = N = 5$*, Left: Poisson arrivals* $\lambda = 2$*, Right: D-BMAP arrivals* ($M = 6$*,* $m = 5$*,* $\alpha^+ = \alpha^- = 1/5$ *and* β *such that the arrival rate* $\lambda = 2$)*.*

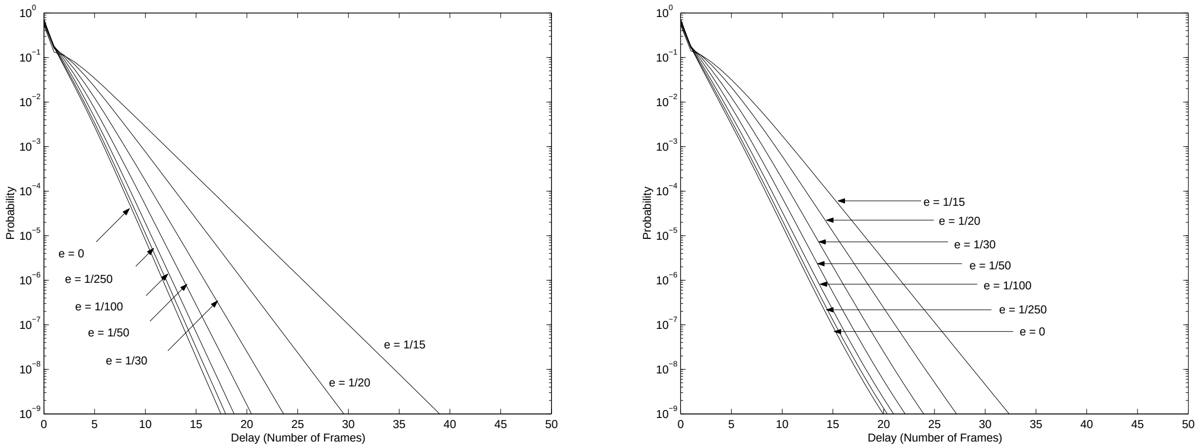

Figure 10: *For* $T = 10, S = 3, N = 7$ *and* $\tilde{e} = 0, 1/250, \ldots, 1/15$ *, Poisson arrivals (*$\lambda = 2$*), Left:* $K = 2$*, Right:* $K = 3$*.*

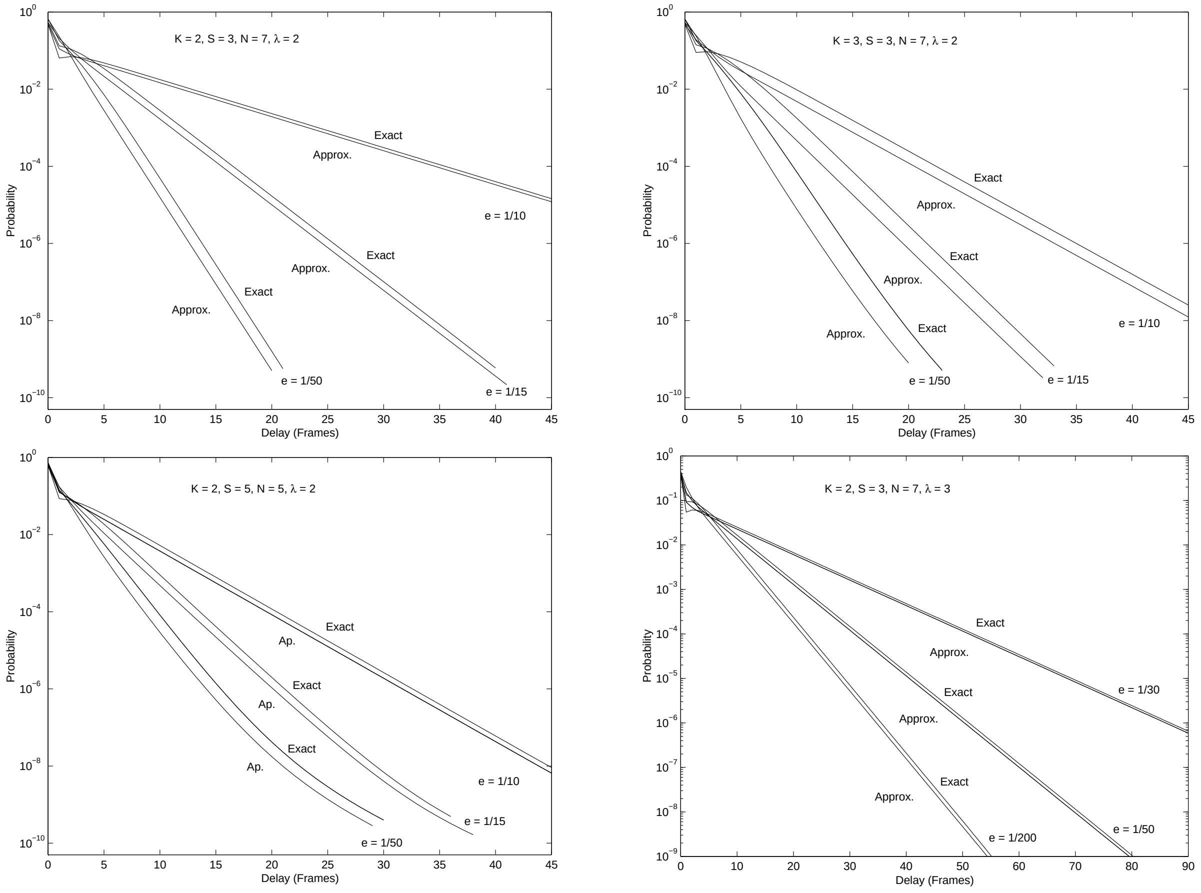

Figure 11: *For* $T = 10$*,* $\tilde{e} = 1/200, \ldots, 1/10$ *and Poisson arrivals, Top Left:* $K = 2, S = 3, N = 7, \lambda = 2$*, Top Right:* $K = 3, S = 3, N = 7, \lambda = 2$*, Bottum Left:* $K = 2, S = N = 5, \lambda = 2$*, Bottum Right:* $K = 2, S = 3, N = 7, \lambda = 3$*.*

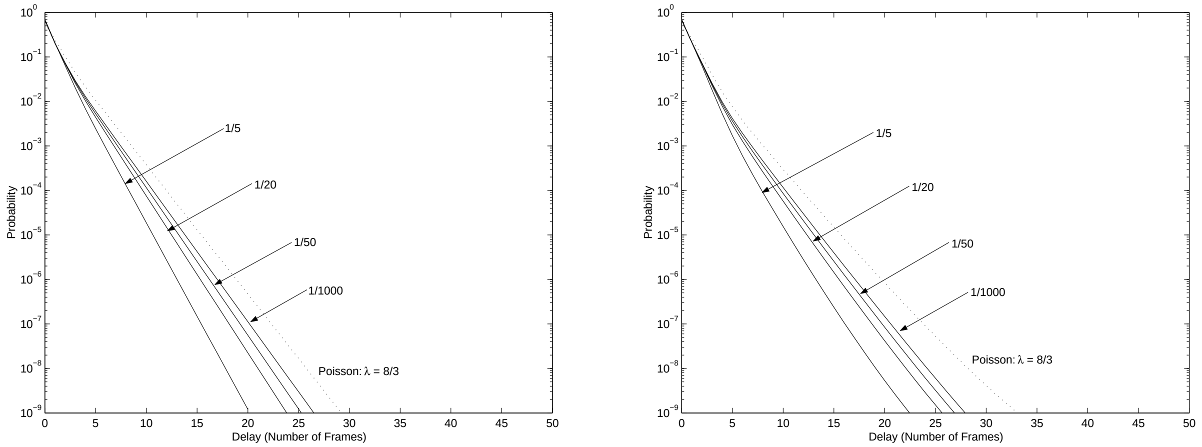

Figure 12: *D-BMAP arrivals, β such that $\lambda = 2$, $M = 2$, $m = 15$, $\tilde{e} = 0$, $T = 10, S = 3, N = 7$, Left: $K = 2$, Right: $K = 3$.*

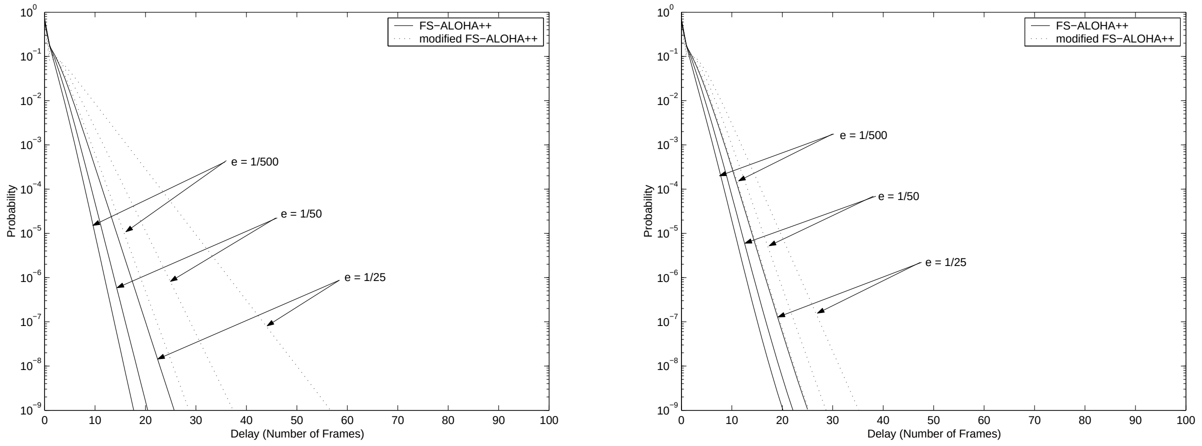

Figure 13: *For $T = 10, S = 3, N = 7$, $\tilde{e} = 1/25, 1/50$ and $1/500$, Poisson arrivals ($\lambda = 2$), Left: $K = 2$, Right: $K = 3$.*