A Queuing Model for the Non-continuous Frame Assembly Scheme in Finite Buffers

Boris Bellalta

NeTS Research Group - Dpt. of Information and Communication Technologies
Universitat Pompeu Fabra
Passeig de la Circumval-lacio 8, 08003 Barcelona, Spain
boris.bellalta@upf.edu

Abstract. A batch-service finite-buffer queuing model for the non continuous frame assembly scheme is presented. The steady-state probabilities are computed by simply solving the bi-dimensional Markov chain which models the queueing system. However, transitions between states require to know the conditional (as it depends on each state) probability distribution of the on-going batch size, which in turn is computed from the queue departure distribution. Finally, the model is used to evaluate the system performance when the minimum (a) and maximum (b) batch size thresholds are tuned.

1 Introduction

Packet aggregation, from simple packet concatenation to complex data fusion techniques, is used to enhance communication systems. It is able to improve the system performance in terms of throughput, delay and/or energy consumption, by reducing both unnecessary protocol headers and channel contention / transmission attempts. Examples of technologies using aggregation are: Optical Burst Switching [1] or IEEE 802.11n WLANs [2]. Additionally, packet aggregation is also applied on top of a broad set of technologies, such as in mesh networks [3] or in 3G cellular communications [4], in order to enhance the transmission of traffic types which suffer from the inclusion of multiple large headers from the different layers of the protocol stack together with the small portion of useful data (e.g. Voice over Internet Protocol), resulting in a significant waste of transmission resources.

Considering the instant when the batch size is completely determined, two aggregation schemes can be depicted: *i)* Continuous Packet Assembly (CPA), where new incoming packets can be aggregated to the already on-going batch transmission, until the maximum batch size is reached, and *ii)* Non-Continuous Packet Assembly (NCPA), where the batch length is completely determined at the instant it is scheduled, based on the number of packets stored in the queue.

In this paper, only the NCPA scheme is considered. In Figure 1 the transmission queue is shown, where the a and b thresholds are the minimum and maximum allowed batch size, respectively. The queue main features considered are:

K. Al-Begain, D. Fiems, and G. Horváth (Eds.): ASMTA 2009, LNCS 5513, pp. 219–233, 2009.

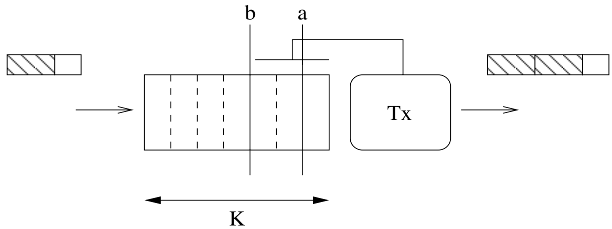


Fig. 1. The NCPA queue

– A packet concatenation strategy is used, where a single header is added to each batch (where a batch is the payload of a group of packets assembled together). The maximum batch size is b packets.
– The transmitter does not store the packets in transmission. They remain in the queue until the batch is completely transmitted.
– Next batch is scheduled as soon as previous transmission has finished and there are at least a packets in the queue. Otherwise, it remains idle.

The proposed model allows to obtain performance metrics such as the packet blocking probability, the average batch size or the average transmission delay. To solve it, a very intuitive relation between the departure $\boldsymbol{\pi^d}$ and the steady-state $\boldsymbol{\pi^s}$ probability distributions is used. This relation is based on the conditional batch length probability distribution $\boldsymbol{\alpha_q}$, which can be computed from the queue departure distribution and is used to determine the transitions inside the set of possible queuing states.

The NCPA queue is based on the queues with bulk-service times [5]. They have received a lot of attention in past years (for example the $GI/M^{[1,b]}/1/K$ [6], the $M/G^{[a,b]}/1/K$ [7] or the $GI/MSC^{[a,b]}/1/K$ [8]), although there are still few works applying these models to real scenarios or communication problems, probably due to its mathematical complexity. Examples are the works from S. Kuppa et al. [9] and Kejie Lu et al. [10] focusing on WLANs performance analysis. Thus, even though the model presented here only works under the assumption of Poisson arrivals and Exponential (batch-length dependant) batch service time distribution. If these assumptions are acceptable, it benefits from two main properties: i) a clear and elegant state-based description (Markov chain) and ii) a striking simplicity which makes it suitable for further enhancements to the packet assembly scheme (e.g. dynamically adapt the b and κ (aggregation factor) parameters to the queuing state) and for its consideration in a joint design with more complex schemes (MAC/PHY protocols).

Once the model is introduced, it is used to show the queue response when the a, b and κ (aggregation factor) parameters are tuned.

2 NCPA Model Description

The NCPA scheme is modeled using a $M/M^{[a,b]}/1_{bd}/K$ (bd:batch dependent) queue with space for K packets, included those in service (there is not a specific space for them). Packets arrive to the queue according to a Poisson process with rate λ and are served in batches of length l packets, with l taking values between a and b, the minimum and maximum batch size respectively. The batch-service times are exponentially distributed with rate μ_l and depend on the number of packets assembled in each batch. Let q_m be the number of packets in the queue after the departure of the $m-1$ batch. Then, next batch size satisfies the policy $l_m = \beta(q_m)$ where:

$$\beta(q_m) = \begin{cases} a & q_m < a \\ q_m & a \leq q_m < b \\ b & b \leq q_m \end{cases} \tag{1}$$

with the following queue state recursion at departure instants

$$q_m = q_{m-1} - \beta(q_{m-1}) + \min\left(v_{m-1}, K - q_{m-1}\right) \tag{2}$$

where v_{m-1} are the packet arrivals between the $m-2$ and $m-1$ batch departure instants (note that the m batch is scheduled as soon as the $m-1$ batch departs, which leaves q_m packets in the queue).

The average batch service time depends on the aggregation factor κ, the packet length (L, bits), the number of packets assembled together in a single batch (l) and the channel capacity (C, bits/second):

$$\frac{1}{\mu_l} = \frac{L + (l-1) \cdot \kappa \cdot L}{C} \tag{3}$$

The κ parameter must be understood as the proportional part of useful data (payload) in each packet. For example, a packet of length L bits has a payload length equal to $\kappa \cdot L$ bits and a header of length $(1-\kappa) \cdot L$ bits. Thus, the aggregation process consists on, given l individual packets, the extraction of the header from each one and assemble their payload together, adding a single header for the entire batch. This will result in a final batch-length equal to $(1-\kappa)L + l \cdot \kappa \cdot L$ bits.

Note how the κ parameter has a double effect: first, it results in batches of variable size and second, it impacts on the effective traffic load to the queue. Regarding nomenclature, throughout the paper, the traffic load (A) only refers to the load related to the relation between the packet arrival rate and the service time given that no aggregation is done (or equivalently, each batch comprises a single packet) and therefore, $A = \lambda \frac{1}{\mu_1}$.

2.1 Departure Distribution

The limiting departure probability distribution, $\boldsymbol{\pi^d}$, gives the probabilites of having q packets in the queue right after batch departures. It is obtained using the Embedded Markov chain approach, solving the linear system $\mathbf{P}\boldsymbol{\pi^d} = \boldsymbol{\pi^d}$, together with the normalization condition $\boldsymbol{\pi^d}\mathbf{1}^T = 1$. $\mathbf{P}$ is the probability

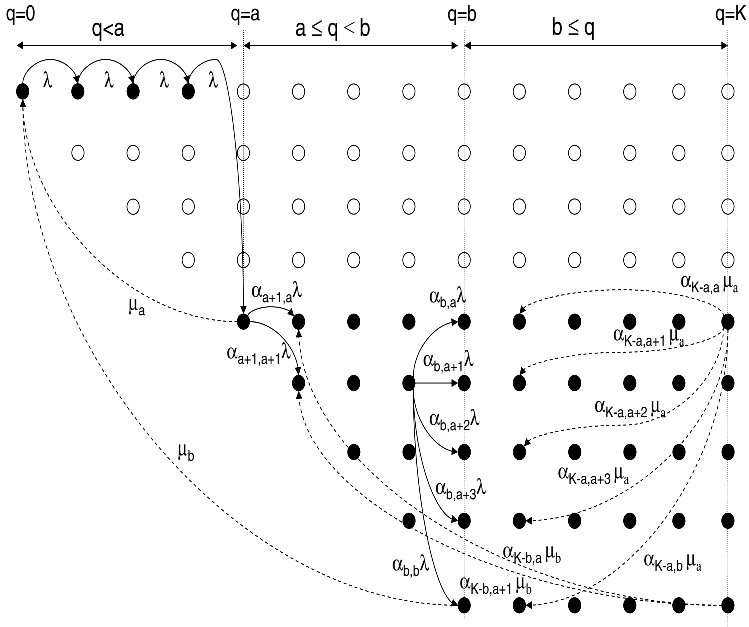


Fig. 2. 2-dimensional Markov chain modelling the NCPA scheme

Table 1. Transition Rates from state $S_{q,l}$, $l\epsilon[a, min(q, b)]$

Next State	Rate	Conditions
Packet arrivals: $q < K$		
$S_{q+1,0}$	λ	$q < a$
$S_{q+1,l'}$	$\alpha_{q+1,l'}\lambda$	$q \geq a$, $l'\epsilon[a, min(q+1, b)]$
Packet departures: $0 < a \leq q \leq K$		
$S_{q-l,0}$	μ_l	$q - l < a$
$S_{q-l,l'}$	$\alpha_{q-l,l'}\mu_l$	$q - l \geq a$, $l'\epsilon[a, min(q-l, b)]$

transition matrix, where the transition probabilities from state $i\epsilon[0, K]$ to state $j\epsilon[0, K]$ are given by:

$$p_{i,j} = \begin{cases} p_{a,j} & i < a, \quad j\epsilon[0, K-a] \\ d_{j+\beta(i)-i,\beta(i)} & i \geq a, \quad j\epsilon[i-\beta(i), K-\beta(i)-1] \\ 1 - \sum_{u=i-\beta(i)}^{K-\beta(i)-1} p_{i,u} & i \geq a, \quad j = K-\beta(i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $d_{n,l}$ is the probability of n arrivals during the transmission of a batch involving l packets, with

$$d_{n,l} = \int_0^\infty f_v(t) f_l(t) dt = \int_0^\infty e^{-\lambda t} \frac{(\lambda t)^n}{n!} \mu_l e^{-\mu_l t} dt \tag{5}$$

where $f_v(t)$ and $f_l(t)$ are the probability functions of the packet arrivals and the l-batch service time, respectively.

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the $\boldsymbol{P}$ matrix is given by:

$$\boldsymbol{P} = \begin{bmatrix} d_{0,1} & d_{1,1} & d_{2,1} & d_{3,1} & 1-d_{0,1}-d_{1,1}-d_{2,1}-d_{3,1} & 0 \\ d_{0,1} & d_{1,1} & d_{2,1} & d_{3,1} & 1-d_{0,1}-d_{1,1}-d_{2,1}-d_{3,1} & 0 \\ d_{0,2} & d_{1,2} & d_{2,2} & 1-d_{0,2}-d_{1,2}-d_{2,2} & 0 & 0 \\ 0 & d_{0,2} & d_{1,2} & 1-d_{0,2}-d_{1,2} & 0 & 0 \\ 0 & 0 & d_{0,2} & 1-d_{0,2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

2.2 Steady-State Distribution

Let $\{s(t) : t > 0\}$ be the stochastic process which represents the temporal evolution of the number of packets inside the queue. Its state-space is:

$$\mathcal{S} = \{S_q :\ 0 \le q \le K\} \quad (7)$$

and its extended counterpart, including the on-going batch size:

$$\mathcal{S}^e = \{S_{q,l} :\ 0 \le q \le K;\ 0 \le l \le b\} \quad (8)$$

A sketch of $\mathcal{S}^e$ is shown in Figure 2 (black states), including some representative transitions between states, where the x-axis corresponds to the number of packets in the queue and the y-axis to the size of the current batch in service. The transitions from state $S_{q,l}$ are summarized in Table 1. Let $\boldsymbol{\pi}^{\boldsymbol{s}^e}$ be the limiting steady-state probability vector. It can be obtained by solving the linear system $\boldsymbol{\pi}^{\boldsymbol{s}^e}\mathbf{Q} = \mathbf{0}$, where $\mathbf{Q}$ is the infinitesimal generator of $s^e(t)$, together with the normalization condition, $\boldsymbol{\pi}^{\boldsymbol{s}^e}\mathbf{1}^T = 1$. Note that

$$\pi_q^s = \sum_{j=a}^{min(q,b)} \pi_{q,j}^{s^e}, \qquad q \ge a \quad (9)$$

with $\pi_q^s = \pi_{q,0}^{s^e}$ otherwise. The $(q, 0)$ state means that there are q packets in the queue, although there is not any active transmission.

From Table 1, transition rates between states are partitioned by the conditional batch size distribution $\boldsymbol{\alpha}_q = \{\alpha_{q,l}, a \le q \le K, l\epsilon[a, \min(q, b)]\}$, where each $\alpha_{q,l}$ is the probability that given a queuing state q, the system is transmitting a batch of length l.

With respect to the $\boldsymbol{\alpha}_q$ probability distribution, computed in next subsection, it is important to remark that it is the steady-state distribution of the on-going batch length when the queue is in state q, regardless if the queue moves to that state after a new packet arrival or after a departure. This assumption allows the simple model construction, although it introduces some transitions which are physically impossible, such as the one with label $\alpha_{a+1,a+1}$ in Figure 2.

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the ininitesimal generator $\boldsymbol{Q}$ is given by:

$$\boldsymbol{Q}_\lambda = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_1 & \alpha_{2,1}\lambda & 0 & 0 & 0 & \alpha_{2,2}\lambda & 0 & 0 & 0 \\ 0 & 0 & -\lambda & \alpha_{3,1}\lambda & 0 & 0 & 0 & \alpha_{3,2}\lambda & 0 & 0 \\ 0 & 0 & 0 & -\lambda & \alpha_{4,1}\lambda & 0 & 0 & 0 & \alpha_{4,2}\lambda & 0 \\ 0 & 0 & 0 & 0 & -\lambda & \alpha_{5,1}\lambda & 0 & 0 & 0 & \alpha_{5,2}\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\lambda & 0 & 0 & -\lambda & \alpha_{3,2}\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{4,1}\lambda & 0 & 0 & -\lambda & \alpha_{4,2}\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_{5,1}\lambda & 0 & 0 & -\lambda & \alpha_{5,2}\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

$$\boldsymbol{Q}_\mu = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mu_1 & -\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu_1 & -\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_{2,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{2,2}\mu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{3,2}\mu_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_{4,1}\mu_1 & -\mu_1 & 0 & 0 & \alpha_{4,2}\mu_1 & 0 \\ \mu_2 & 0 & 0 & 0 & 0 & 0 & -\mu_2 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & -\mu_2 & 0 & 0 \\ 0 & 0 & \alpha_{2,1}\mu_2 & 0 & 0 & 0 & \alpha_{2,2}\mu_2 & 0 & -\mu_2 & 0 \\ 0 & 0 & 0 & \alpha_{3,1}\mu_2 & 0 & 0 & 0 & \alpha_{3,2}\mu_2 & 0 & -\mu_2 \end{bmatrix} \tag{11}$$

where the bidimensional Markov chain, considering only the black states from Figure 2, has been transformed in a single dimensional chain. The infinitesimal generator is $\boldsymbol{Q} = \boldsymbol{Q}_\lambda + \boldsymbol{Q}_\mu$.

2.3 Conditional Batch Size Probability Distribution

Given that there are q packets in the queue, with $a \leq q \leq K$ (otherwise there is no batch transmission), the probability that the length of the on-going batch is l packets, with $a \leq l \leq b$, is:

$$\alpha_{q,l} = \frac{\sum_{i=l}^{q} \mathbf{I}_{\beta(i)=l} \left(p_i^d \cdot p_a(q-i|1/\mu_{\beta(i)}) \right)}{\sum_{j=a}^{q} p_j^d \cdot p_a(q-j|1/\mu_{\beta(j)})} \tag{12}$$

where $\mathbf{I}_{\beta(i)=l}$ is a boolean indicator function which returns 1 if the condition $\beta(i) = l$ is satisfied. Otherwise, it returns 0. Additionally,

$$p_i^d = \begin{cases} \sum_{j=0}^{a} \pi_j^d & i = a \\ \pi_i^d & a < i \leq K \\ 0 & otherwise \end{cases} \tag{13}$$

and $p_a(i|1/\mu_j)$ is the probability of at least i arrivals during the service time with rate $1/\mu_j$.

$$p_a(i|1/\mu_j) = \begin{cases} 1 & i = 0 \\ 1 - \sum_{m=0}^{i-1} d_{m,j} & otherwise \end{cases} \tag{14}$$

Example. Considering the queue $M/M^{[1,2]}/1_{bd}/5$, the $\boldsymbol{\alpha}_4$ probability distribution is given by:

$$\begin{aligned}
\alpha_{4,0} &= 0 \\
\alpha_{4,1} &= \frac{p_1^d \cdot p_a(3|1/\mu_1)}{p_1^d \cdot p_a(3|1/\mu_1) + p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)} \\
\alpha_{4,2} &= \frac{p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)}{p_1^d \cdot p_a(3|1/\mu_1) + p_2^d \cdot p_a(2|1/\mu_2) + p_3^d \cdot p_a(1|1/\mu_2) + p_4^d \cdot p_a(0|1/\mu_2)}
\end{aligned} \tag{15}$$

Note that $p_i^d = 0$ if $i > K - b$. Expressions from Equation 15 can be simplified accordingly.

2.4 Performance Metrics

Once the $\boldsymbol{\pi^d}$ and $\boldsymbol{\pi^{se}}$ distributions are obtained, several performance metrics can be computed, such as the packet blocking probability,

$$P_b = \pi_K^s \tag{16}$$

and the probability that the transmitter is empty or non-transmitting:

$$P_{nt} = \sum_{q=0}^{a-1} \pi_q^s \tag{17}$$

Additionally, the average size of the transmitted batches,

$$E[\gamma] = \sum_{i=a}^{K} p_i^d \beta(i) \tag{18}$$

the average number of packets in the queue,

$$E[N] = \sum_{q=0}^{K} q\pi_q^s \tag{19}$$

and the average packet transmission time, obtained from $E[N]$ by applying Little's Law:

$$E[R] = \frac{E[N]}{\lambda(1 - P_b)}. \tag{20}$$

3 Model Validation and NCPA Analysis: Numerical Example

Some numerical and simulation results are provided to validate the accuracy and applicability of the presented queuing model. The $M/M^{[a,b]}/1_{bd}/K$ queue

Table 2. Arrival/Steady-state and Departure distributions for $A = 0.5$ and 1.5 Erlangs

$A = 0.5$ Erlangs	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$
(0)	0.6243 (0.6239)	0.5385 (0.5381)	0.6465 (0.6468)	0.5443 (0.5445)
(1)	0.2359 (0.2363)	0.2473 (0.2477)	0.2279 (0.2276)	0.2454 (0.2452)
(2)	0.0889 (0.0886)	0.1153 (0.1150)	0.0863 (0.0862)	0.1129 (0.1129)
(3)	0.0401 (0.0403)	0.0564 (0.0565)	0.0283 (0.0284)	0.0525 (0.0525)
(4)	0.0106 (0.0106)	0.0254 (0.0255)	0.0107 (0.0108)	0.0253 (0.0253)
(5)	0.0 (0.0)	0.0168 (0.0170)	0.0 (0.0)	0.0193 (0.0194)
$A = 1.5$ Erlangs	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$
(0)	0.2484 (0.2483)	0.1164 (0.1163)	0.3294 (0.3299)	0.1424 (0.1427)
(1)	0.2388 (0.2386)	0.1370 (0.1369)	0.2274 (0.2271)	0.1445 (0.1444)
(2)	0.1739 (0.1739)	0.1387 (0.1386)	0.2423 (0.2422)	0.1592 (0.1592)
(3)	0.2756 (0.2758)	0.1779 (0.1779)	0.1285 (0.1285)	0.1439 (0.1439)
(4)	0.0631 (0.0631)	0.1391 (0.1392)	0.0721 (0.0721)	0.1206 (0.1206)
(5)	0.0 (0.0)	0.2907 (0.2908)	0.0 (0.0)	0.2891 (0.2888)

Table 3. Arrival/Steady-state and Departure distributions for $A = 0.5$ and 1.5 Erlangs

$A = 0.5$ Erlangs	$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$
(0)	0.5662 (0.5662)	0.2638 (0.2639)	0.5000 (0.5000)	0.1538 (0.1541)
(1)	0.2452 (0.2457)	0.3781 (0.3784)	0.2500 (0.2503)	0.2307 (0.2313)
(2)	0.1155 (0.1151)	0.1851 (0.1851)	0.2500 (0.2495)	0.3076 (0.3057)
(3)	0.0729 (0.0729)	0.0963 (0.0961)	0.0 (0.0)	0.1538 (0.1561)
(4)	0.0 (0.0)	0.0425 (0.0424)	0.0 (0.0)	0.0769 (0.0754)
(5)	0.0 (0.0)	0.0340 (0.0338)	0.0 (0.0)	0.0769 (0.0772)
$A = 1.5$ Erlangs	$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$	$\boldsymbol{\pi}^{\boldsymbol{d}}$	$\boldsymbol{\pi}^{\boldsymbol{s}}$
(0)	0.2933 (0.2936)	0.0908 (0.0909)	0.2500 (0.2497)	0.0533 (0.0540)
(1)	0.2066 (0.2066)	0.1547 (0.1549)	0.1875 (0.1877)	0.0933 (0.0968)
(2)	0.2508 (0.2505)	0.1609 (0.1609)	0.5625 (0.5624)	0.2133 (0.2441)
(3)	0.2756 (0.2758)	0.1779 (0.1779)	0.0 (0.0)	0.1600 (0.1591)
(4)	0.0631 (0.0631)	0.1391 (0.1392)	0.0 (0.0)	0.1200 (0.1192)
(5)	0.0 (0.0)	0.2907 (0.2908)	0.0 (0.0)	0.3600 (0.3565)

is evaluated using different traffic loads and a and b parameters. A small queue, $K = 5$, is considered in order to be able to show the complete $\boldsymbol{\pi}^{\boldsymbol{s}}$, $\boldsymbol{\pi}^{\boldsymbol{d}}$ and $\boldsymbol{\alpha}_q$ values. The channel has a constant capacity equal to $C = 100$ Kbps, $L = 100$ bits (including the header) is the average packet length and $\kappa = 0.5$, resulting in these batch-service times are: $1/\mu_l = [100/C, 150/C, 200/C]$, for $l = 1, \ldots, 3$.

In Table 2 and 3, the departure and steady state distributions obtained by solving the queuing model are compared with the simulation ones (between

Table 4. $\boldsymbol{\alpha}_q$ and $\boldsymbol{\pi}^{se}$ distributions with $A = 1.5$ Erlangs

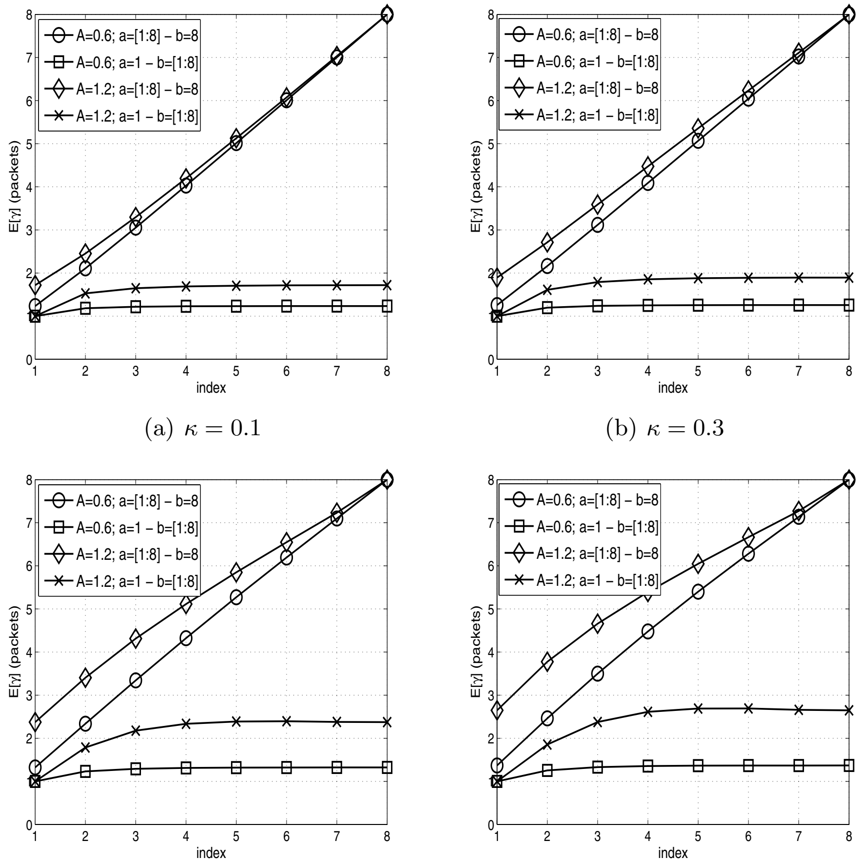
-	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$		$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
State	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$	$\alpha_{q,l}$	$\boldsymbol{\pi}^{se}$
(0,0)	1	0.1164	1	0.1424	1	0.0908	1	0.0533
(1,0)	-	-	-	-	1	0.1547	1	0.0933
(1,1)	1	0.1370	1	0.1445	-	-	-	-
(2,0)	-	-	-	-	-	-	1	0.2133
(2,1)	0.6269	0.0822	0.5795	0.0867	-	-	-	-
(2,2)	0.3730	0.0564	0.4204	0.0725	1	0.1609	-	-
(3,0)	-	-	-	-	-	-	-	-
(3,1)	0.3069	0.0493	0.4035	0.0520	-	-	-	-
(3,2)	0.6930	0.1285	0.3377	0.0502	0.676	0.1114	-	-
(3,3)	-	-	0.2587	0.0416	0.324	0.0578	1	0.1600
(4,0)	-	-	-	-	-	-	-	-
(4,1)	0.2377	0.0296	0.2969	0.0312	-	-	-	-
(4,2)	0.7622	0.1095	0.2860	0.0347	0.6582	0.0771	-	-
(4,3)	-	-	0.4162	0.0546	0.3417	0.0433	1	0.1200
(5,0)	-	-	-	-	-	-	-	-
(5,1)	0.2128	0.0444	0.2586	0.0468	-	-	-	-
(5,2)	0.7871	0.2463	0.2882	0.0782	0.64	0.1735	-	-
(5,3)	-	-	0.4531	0.1640	0.36	0.1301	1	0.3600

Table 5. $E[\gamma]$ (packets) and $E[R]$ (milliseconds) against A Erlangs

-	$M/M^{[1,2]}/1_{bd}/5$		$M/M^{[1,3]}/1_{bd}/5$		$M/M^{[2,3]}/1_{bd}/5$		$M/M^{[3,3]}/1_{bd}/5$	
A	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$	$E[\gamma]$	$E[R]$
0.5	1.139	1.695	1.164	1.687	2.073	2.852	3	4.333
1.0	1.351	2.367	1.429	2.293	2.177	2.756	3	3.592
1.5	1.512	2.780	1.643	2.647	2.249	2.858	3	3.416
2.0	1.621	3.027	1.799	2.855	2.296	2.952	3	3.353
2.5	1.694	3.182	1.911	2.986	2.329	3.003	3	3.325
3.0	1.746	3.286	1.995	3.075	2.353	3.085	3	3.312
3.5	1.783	3.360	2.058	3.139	2.372	3.131	3	3.306

brackets). The simulator has been developed in C programming language using the COST simulation libraries [11]. In Table 4, the steady-state distribution of the expanded state-space, along with the conditional batch distribution are introduced and, finally, in Table 5, the average batch size and the average transmission delay are shown.

At low traffic conditions, comparing the $M/M^{[1,2]}/1_{bd}/5$ and $M/M^{[1,3]}/1_{bd}/5$ queues, the latter shows a higher blocking probability, $P_b = \pi^s_5$. As there is not an specific space for the packets in service and they are only dequeued after the batch is completely transmitted, it results in long periods of time in which there are no packet departures, increasing the probability of losing several consecutive packets. However, this result changes when traffic increases as the



(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

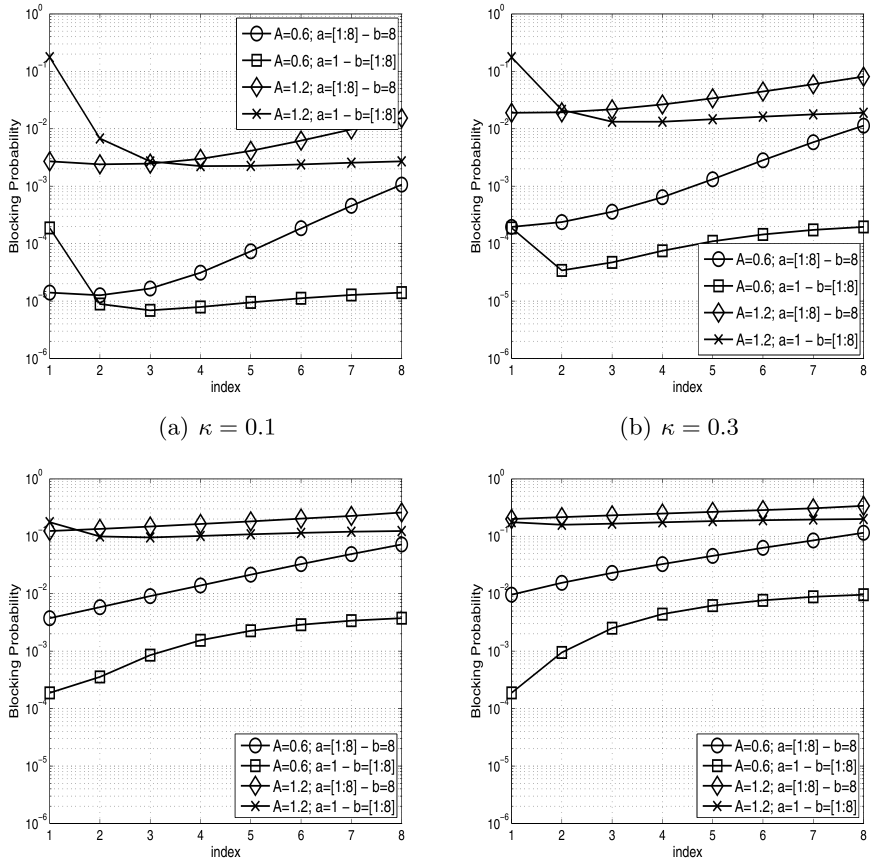
(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 3. Average Batch Size

queue $M/M^{[1,3]}/1_{bd}/5$ becomes more efficient. Thus, a higher b value allows to increase the $P_{nt} = \pi_0^s$ probability, as well as the $E[\gamma]$, resulting also in a lower transmission delay, $E[R]$ (Table 5).

On the other hand, the $M/M^{[2,3]}/1_{bd}/5$ and $M/M^{[3,3]}/1_{bd}/5$ increase the a value, which results in higher $P_b = \pi_5^s$ and $P_{nt} = \pi_0^s + \pi_1^s$ and $P_{nt} = \pi_0^s + \pi_1^s + \pi_2^s$ probabilities respectively. At low traffic conditions, the extra delay to schedule a batch is clearly shown, as at least two and three packets are required in each case. This inefficiency is proportionally reduced at high traffic loads, as single packet batches are avoided.

Regarding the values of $\boldsymbol{\alpha}_q$ (Table 4), notice that they also provide a very useful information to understand how the queue performs. For example, notice how increasing b from 2 to 3 in the $M/M^{[1,b]}/1_{bd}/5$ queue results in a higher probability to transmit single packet batches (in Table 4, this is to be in a given state and with a single packet batch active). This, in turn, is caused by



(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

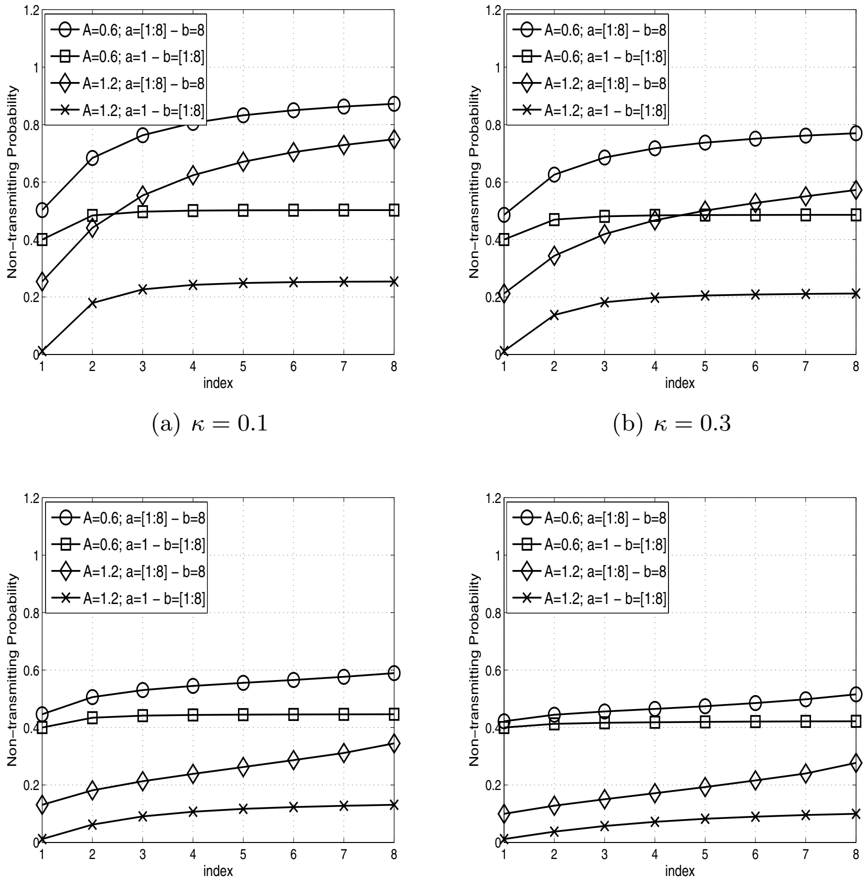
(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

Fig. 4. Blocking Probability

the transmission of these longer batches (3 packets) that now are possible, which left the system at lower states when they depart. Furthermore, increasing a from 1 to 2, it is even worst in terms of increasing the presence of 3 packet batches when there are 4 or 5 packets in the queue.

4 Results: Impact of *a* and *b*

This section focuses on the impact of different a and b values, traffic loads ($A = 0.6$ and $A = 1.2$ Erlangs) and κ values (0.1, 0.3, 0.7 and 0.9) on the queue response. The same NCPA parameters used in previous example, with the exception of $K = 15$ and a and b that range from 1 to 8, are considered. Notice that in the Figures, the x-axis values are the index of the set of possible a and b values, which are between brackets in the legend.



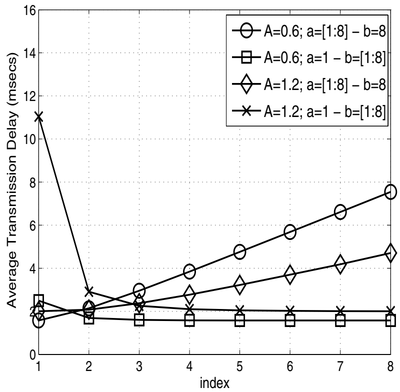
(a) $\kappa = 0.1$ (b) $\kappa = 0.3$

(c) $\kappa = 0.7$ (d) $\kappa = 0.9$

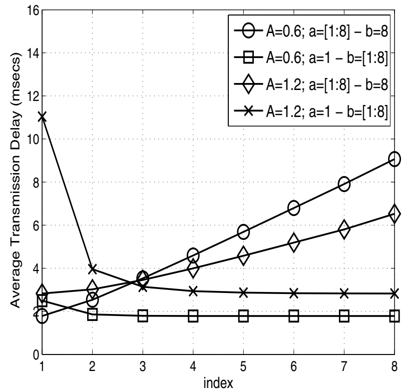
Fig. 5. Non-transmitting probability

In Figure 3, the average batch size ($E[\gamma]$) is shown. As expected, higher traffic loads and κ values result in longer batches (as the batch size is proportional to the number of packets stored in the queue when it is scheduled, that in turn depends on the duration of previous batch). Increasing a, the batch size is constrained to this minimum value. Plus, notice that for a values greater than $K/2$ all scheduled batches will have a size equal to a packets. On the other hand, increasing b results in higher $E[\gamma]$ values until a saturation point is reached, from where $E[\gamma]$ remains approximately constant. Clearly, this saturation point is also proportional to the traffic load and κ and it is also related to the number of packets arriving at the queue during previous batch transmission.

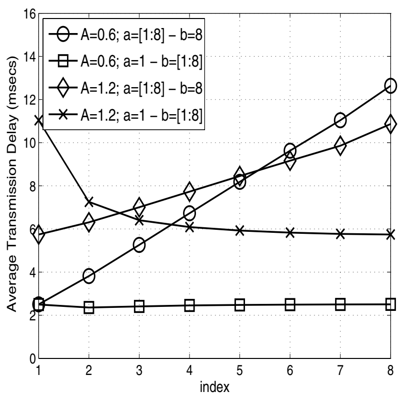
In Figure 4 the blocking probability is plotted. Increasing a results in higher blocking probabilities, except the case with $\kappa = 0.1$. In this case, it is slightly better to wait until a second packet arrives to the queue than to start the batch



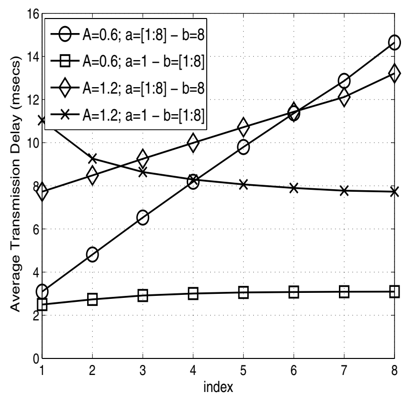
(a) $\kappa = 0.1$



(b) $\kappa = 0.3$



(c) $\kappa = 0.7$



(d) $\kappa = 0.9$

Fig. 6. Average Transmission Delay

transmission as soon as possible (note that the low κ makes the aggregation process very efficient). Conversely, when the b parameter is increased, a minimum in the blocking probability appears for some traffic loads and κ values. For higher traffic loads ($A = 1.2$ Erlangs) there is always a minimum (for all κ values). On the contrary, for low traffic loads, it requires low κ values to show that minimum. For example, in the cases with $A = 0.6$ and κ equal to 0.7 and 0.9, always a longer batch results in higher blocking probabilities. These results, in general, are caused by the fact that the packets of a batch are not deallocated until its completion and thus longer batches increase the probability that new arrivals fill the queue.

In terms of the non-transmitting probability (Figure 5), higher b values result always in higher P_{nt}, but not significantly. Thus, although increasing b results in a more efficient use of the channel, the channel utilization still remains high because of the next batch is scheduled as soon as a single packet ($a = 1$) is ready in the queue (single packet batches are more frequent). Conversely, increasing a this probability boosts substantially as the transmitter remains idle until there are at least a packets. Furthermore, it also causes more efficient aggregations (compared when increasing b), as longer batches are mandatory. Increasing κ, batches also become longer, which increases the channel utilization, thus reducing P_{nt}. A high P_{nt} value means a lower link occupation which in single-user point to point links can result, for example, in lower energy consumption (less transmitted bits or longer sleep periods). However, in random access solutions, it becomes a fundamental parameter as their performance is influenced by the number and persistence of the transmission attempts, which depends on P_{nt} [10]. Then, low P_{nt} values will reduce the collision probability, which can result in higher throughput or lower delay, as well as lower energy consumption.

Finally, the a and b values also show a notably impact in terms of transmission (including queuing) delay (Figure 6). There is a relation (quasi linear) between a, the response delay and the time between packet arrivals. The transmitter remains idle until there are at least a packets in the queue, which is proportional to a and inversely proportional to the traffic load, as $1/\lambda = 1/(A\mu_1)$. On the contrary, increasing b always result in lower queuing delays, except for high κ values (and low traffic loads, Figure 6.d), where the extra delay waiting for the completion of previous batch (which is longer) becomes more relevant than the aggregation gain achieved.

5 Conclusions

A queuing model for the non-continuous frame aggregation scheme in finite buffers has been presented. It is used to provide some directions about the impact of tuning the batch size parameters.

Acknowledgement

This work was partially supported Spanish Government under project TEC2008-06055/TEC. The author would like to specially acknowledge the contribution of the reviewers to improve the final quality of this paper.

References

1. Yao, S., Xue, F., Mukherjee, B., Ben Yoo, S.J., Dixit, S.: Electrical Ingress Buffering and Traffic Aggregation for Optical Packet Switching and Their Effect on TCP-Level Performance in Optical Mesh Networks. IEEE Communications Magazine (September 2002)
2. Xiao, Y.: IEEE 802.11n: enhancements for higher throughput in wireless LANs. IEEE Wireless Communications 12(6), 82–91 (2005)
3. Ganguly, S., Navda, V., Kim, K., Kashyap, A., Niculescu, D., Izmailov, R., Hong, S., Das, S.R.: Performance Optimizations for Deploying VoIP Services in Mesh Networks. IEEE Journal On Selected Areas In Communications 24(11), 2147 (2006)

4. Rittenhouse, G., Zheng, H.: Providing VOIP service in UMTS-HSDPA with frame aggregation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings(ICASSP 2005), vol. 2 (2005)
5. Medhi, J.: Stochastic models in queueing theory. Academic Press, Inc., London (1991)
6. Vijaya Laxmi, P., Gupta, U.C.: On the finite-buffer bulk-service queue with general independent arrivals: $GI/M^{[b]}/1/N$. In: The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2006), Waterloo, Ontario, Canada (2006); Operations Research Letters, vol. 25, pp. 241–245 (1999)
7. Chaudhry, M.L., Gupta, U.C.: Modelling and analysis of $M/G^{[a,b]}/1/N$ queue - A simple alternative approach. Queueing Systems 31(1-2), 95–100 (1999)
8. Banik, A.D., Gupta, U.C., Chaudhry, M.L.: Finite-buffer bulk service queue under Markovian service process. In: Proceedings of the 2nd International Conference on Performance evaluation methodologies and tools, Nantes, France (October 2007)
9. Kuppa, S., Dattatreya, G.R.: Modeling and analysis of frame aggregation in unsaturated WLANs with finite buffer stations. In: IEEE International Communications Conference (ICC 2006), Istanbul, Turkey (June 2006)
10. Lu, K., Wang, J., Wu, D., Fang, Y.: Performance of a burst-frame-based CSMA/CA protocol: Analysis and enhancement. Wireless Netw. 15, 87–98 (2007)
11. Chen, G. (G.).: Component Oriented Simulation Toolkit (2004), `http://www.cs.rpi.edu/~cheng3/`