Performance Evaluation of a Worst Case Model of the MetaRing MAC Protocol with Global Fairness

G. Anastasi, L. Lenzini, B. Meini

University of Pisa, Department of Information Engineering
Via Diotisalvi, 2 - 56126 Pisa, Italy
anastasi@iet.unipi.it, lenzini@iet.unipi.it

University of Pisa, Department of Mathematics
Via Buonarroti, 2 - 56127 Pisa, Italy
meini@dm.unipi.it

ABSTRACT
The MetaRing is a Medium Access Control (MAC) protocol for high-speed LANs and MANs. The MetaRing MAC protocol offers its users synchronous, and asynchronous types of services and can operate under two basic access control modes: *buffer insertion* for variable size packets, and *slotted* for fixed length packets (i.e., cells). The latter mode of operation is considered in this paper, which only reports performance results of an analysis related to the asynchronous type of service. In this paper we propose and solve a specific worst-case model that enables us to calculate quantiles of the queue length distribution at cell departure time as a function of the offered load, and for three different arrival processes: Poisson, Batch Poisson (B-Poisson), and Batch Markov Modulated Poisson Process (BMMPP). The model proposed is a discrete time discrete state Markov chain of *M/G/1-Type,* and hence we used a matrix analytic methodology to solve it. Exploitation of the structure of the blocks belonging to the transition probability matrix, considerably reduces the computational costs. Our results show that the more realistic the arrival process is, the longer the tail of the queue length distribution.

1. Introduction
The MetaRing MAC protocol has been analysed in ([CID93], [WU92], [ANA95a]). In the first two papers the cell destination addresses are assumed to be uniformly distributed among half of the ring, while the third paper focuses on a scenario which consists of a group of N+1 equally spaced stations located in one half of the ring, as shown in Figure 1.

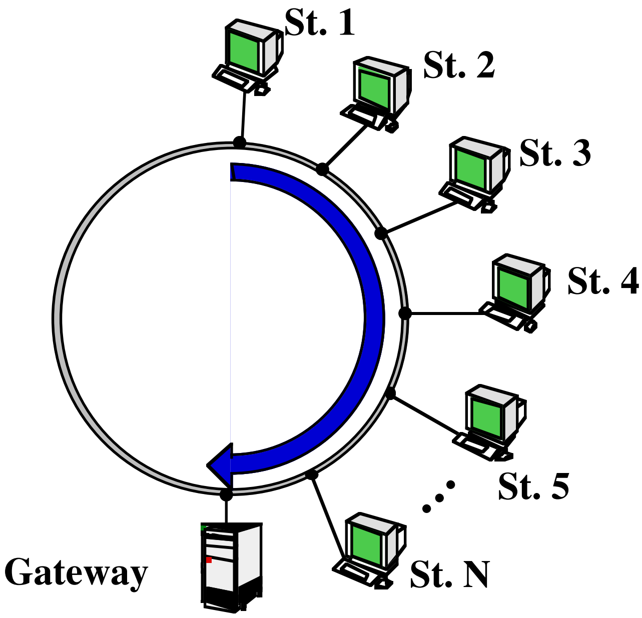

Figure 1: Network scenario

The $(N+1)-th$ station operates as a gateway interconnecting the MetaRing with other subnetworks. Furthermore, it is assumed that stations from $\{1\}$ up to $\{N\}$ transmit asynchronous traffic to the gateway. Station $\{1\}$ always observes empty slots, while station $\{N\}$ is covered by traffic transmitted by its upstream stations, and therefore it can only send a cell if a slot has not been used by stations $\{1,2,...,N-1\}$.

In [ANA95a], the asynchronous part of the MetaRing MAC protocol is analyzed in depth. Its behavior is first analyzed in *asymptotic conditions* (i.e., when each station is trying to seize all the medium capacity), then in *underload conditions* (i.e., when the offered load is lower than the maximum achievable aggregate throughput), and finally in *overload conditions* (i.e., when the offered load is slightly higher than the maximum achievable aggregate throughput). Specifically, under asymptotic traffic conditions a closed formula for the throughput achieved by any station was derived, while in underload and overload conditions a simulative analysis was performed. The rationale behind the simulative choice was that the modelling and performance analysis of the MetaRing MAC protocol is known to be a very difficult problem. The reason for this is the high degree of interactions among a plethora of processes, which makes an exact analysis of the network almost impossible. To overcome these difficulties we introduce a simplified model which can be analytically solved and yet still provides useful information on the quality of service (QoS). This model represents a worst case scenario in which network congestion is stressed; i.e., all the stations apart from station $\{N\}$ (which will be referred to throughout as the *tagged station*) are in *never empty queue* conditions.

Throughout the paper this model will be referred to as a *worst-case model*. We solve this model by using the matrix analytic technique [NEU89].

In LAN/MAN network performance evaluation studies, the data traffic arrival process is often modelled as a Poisson process or Batch Poisson process, though a number of traffic studies have shown that packet interarrivals are not exponentially distributed ([CIN94], [GUS90], [GUS91], [LEL94], [PAX94a], [PAX94b]), that packets are structured into batch of cells, and that successive batches are correlated.

We will show below that there is experimental evidence that the Batch Markov Modulated Poisson Process (BMMPP) is a good approximation for the aggregate cell arrival process in a LAN. Since BMMPP is a correlated, nonrenewal process, and modulated in a Markovian manner, the distribution of the number of cell arrivals in a given (fixed) time interval has a tail which is heavier than in the Poissonian and Batch Poissonian cases. Hence, although for data the QoS is generally expressed in terms of average delay and throughput, to compare the effect of this tail on the cell buffer occupancy, we decided to compute the cell buffer occupancy distributions for all three arrival processes.

This paper is organized as follows. Section 2 outlines the MetaRing MAC protocol. Section 3 describes the worst case model, while the BMMPP arrival process is characterized in Section 4. Section 5 describes the Markov chain with which the worst-case model is represented. Sections from 6 to 10 describe the *M/G/1-Type* Markov chain and its solution. Our results are discussed in Section 11, and conclusions are drawn in Section 12.

2. MetaRing MAC Protocol Description

The MetaRing is a Medium Access Control (MAC) protocol proposed for high speed LANs and MANs. It connects a set of stations by means of a bi-directional ring made up of full duplex point-to-point serial links. A MetaRing can operate under two basic access control modes: *buffer insertion* for variable size packets, and *slotted* for fixed length cells. In this paper only the slotted access mode is considered.

The MetaRing MAC protocol provides two types of services: asynchronous and synchronous. This section outlines the MAC protocol aspects which are relevant to the study reported in the paper (i.e., asynchronous traffic only). Details on the MAC protocol can be found in [CID93], [WU92], and [OFE94].

According to the slotted access mode, information is segmented into cells and each cell transmitted in one slot. Slots are structured into a header and an information field. The header includes a busy bit which indicates whether the slot is empty (busy bit=0) or busy. A cell can be accommodated in one empty slot. Since the ring is bi-directional the MetaRing MAC protocol uses the *shortest path* criterion to choose one of the possible directions. Cells are removed by the destination station, which frees the slot by resetting the busy bit to zero. After cell removal the slot can be reused by the same station or by its downstream stations. Slot reuse increases the aggregate throughput beyond the capacity of a single link but can cause starvation.

In order to prevent starvation, the MetaRing MAC protocol includes a fairness mechanism which is based on a control signal, called SAT (*SATisfied*), which circulates in the opposite direction with respect to the flow of information it regulates. Each station has a counter which increases by one every time the station transmits a cell. The station can transmit cells whenever it observes an empty slot, unless the counter is equal to the value of a MAC protocol parameter denoted by K. When the counter reaches the value of K, the station must refrain from sending new cells until the SAT signal arrives at the station. Upon SAT arrival, the station resets the counter to zero if it has been sent a number of cells greater than or equal to an additional protocol parameter denoted by L, with $K \geq L \geq 0$, and then releases the SAT to its upstream station. When the counter is less than L, the station is not satisfied, and the following two mutually exclusive events can occur:

- if the station has data to transmit, it holds the SAT until the counter reaches the value of L or the buffer becomes empty (whichever comes first), then the station resets the counter and releases the SAT;
- if the station has no data to transmit, it resets the counter and releases the SAT.

Hereafter, the ring latency (S) and the distance (d) between equally spaced stations are expressed in slots and they are assumed to be integers greater than or equal to one.

3. Worst-case Model Description

To reduce the complexity of the MetaRing MAC protocol modelling, we focus on station $\{N\}$ (tagged station) and we assume that

1. The remaining stations (i.e., stations $\{1,2,....,N-1\}$ operate in asymptotic conditions (i.e., stations $\{1,2,....,N-1\}$ always have packets ready for transmission).

2. Condition $K = L$ holds for each station. This condition guarantees that, in asymptotic conditions, station $\{N\}$ achieves the same throughput as the others, i.e., the network is fair [ANA95a].
3. $K \cdot N \geq S$. This condition implies that, in asymptotic conditions, the maximum aggregate throughput is achieved [ANA95a].
4. $K \cdot (N-1) + 2d \leq S$.

Assumption 4 makes the model analytically tractable. Clearly, it identifies a specific set of network parameters which nevertheless are realistic.

If we define the *transmitting state* of a MetaRing (for which assumptions 1-4 hold) at time *n* by the vector $\{k^{(SAT)}, Count(1), Count(2), ..., Count(N)\}_{(n)}$ where $1 \leq k^{(SAT)} \leq S$ is the slot position (measured, say, from station $\{N\}$) of the SAT within the ring, and $Count(j),\ j = 1,2,..,N$, is the value of the counter at station $\{j\}$, we will prove that the MetaRing converges to a steady state operation, such that active stations visit a finite number of transmission states on successive steady state cycles, hereafter called *SAT cycles*. More formally, it will be proved that the process

$$\Omega = \left\{ \left[k^{(SAT)}, Count(1), Count(2), ..., Count(N) \right]_{(n)}, n = 0,1,2,... \right\} \qquad [1]$$

is regenerative with respect to the time instants (i.e., regeneration epochs) at which the SAT leaves station $\{N\}$ while this station is observing a stream of empty slots. Hence, we concentrate on a SAT cycle (i.e., the time interval between two consecutive regeneration epochs). Without lose of generality, throughout the paper it is assumed that the SAT cycle is evaluated by observing the SAT rotation from station $\{N\}$. A trajectory of the Ω process is plotted below in Figure 2. The figure plots, as a function of time, the SAT position along the ring in addition to the evolution of $Count(j),\ j = 1,2,....,N$ starting from the time that station $\{j\}$ is hit by the SAT. The SAT position is represented by a straight line with a slope equal to $+1$. Each $Count(j),\ j = 1,2,....,N$ evolution is represented by a horizontal segment. There is one segment for each station. For ease of representation both the SAT position and the $Count(j)$ evolutions are represented by continuous lines. Stations are reported in the ordinate of each figure. Each horizontal segment is interrupted when either the station observes busy slots or when the station has transmitted K cells, whichever occurs first. In the former case the $Count(j),\ j = 1,2,....,N$, evolution is resumed when empty slots are again observed by the station, while in the latter situation the evolution restarts when the SAT is observed again during the next round around the ring. When a station has transmitted its maximum quota of cells (i.e., K cells) and before the SAT is observed again, each empty slot observed by a station is left unchanged by the station itself. With this type of representation SAT cycles can be identified straightforwardly, as it is easy to realize when the process Ω restarts visiting the same set of states already visited in the previous cycle.

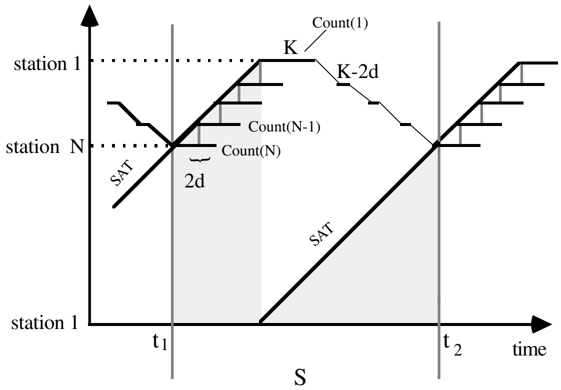

Figure 2: A trajectory of the Ωprocess

Without lose of generality, for our purposes it is convenient to start from the state where the SAT leaves station $\{N\}$ and $Count(i) = K$, for $i = N-1, N-2, ..., 2, 1$. For ease of representation, hereafter this state will be denoted by

$$\{\Phi\} = \left\{k^{(SAT)} = 1, Count(1) = K, Count(2) = K, ..., Count(N) = 0\right\} \qquad [2]$$

The THEOREM proved in [ANA95b], shows that $\{\mathbf{\Phi}\}$ is always visited by the Ω process whatever the initial state of Ω is, and furthermore, from the time $\{\mathbf{\Phi}\}$ is visited, station $\{N\}$ observes a continuous stream of $2d$ empty slots before being covered by the slots made busy by its upstream stations. Also, from $K \cdot N \geq S$ and $d \leq \lceil (S/2)/N \rceil$ (i.e., stations are equally spaced in half of the ring), the condition $K \geq 2d$ readily follows. This condition will be used extensively throughout this paper.

Starting from state $\{\mathbf{\Phi}\}$ (see epoch t_1 in Figure 2) and following the same approach developed in [ANA95a], it can be shown that station $\{N\}$ can transmit k $(0 \leq k \leq 2d \leq K)$ cells before being covered by the busy slots generated by the upstream stations. After time $t_1 + 2d$, the transmissions by station $\{N\}$ will be inhibited until the other stations have transmitted K cells (and are thus satisfied). The SAT circulate around the ring without being withheld by the other stations since they all have $Count(j) = K$, $(j = N-1, N-2, ..., 1)$. After a complete round around the ring the SAT comes back to station $\{N\}$. At this point in time, three mutually exclusive events may occur depending upon the status of the queue and the value of k. Specifically, if $k = K$ the station has already been satisfied and thus the SAT is released; if $k < K$ and the queue is empty the SAT is immediately released; if $k < K$ and the queue is not empty, station $\{N\}$ withholds the SAT until either it has transmitted $K - k$ cells or the queue becomes empty, whichever occurs first. In all cases, starting from the time the SAT is released, station $\{N\}$ observes empty slots.

Note that condition $K \cdot (N-1) + 2d \leq S$ guarantees that when the SAT comes back to station $\{N\}$, $Count(j) = K$, for $j = N-1, N-2, ..., 1$, i.e., all the other stations are satisfied.

The above considerations imply that, from the time the SAT leaves station $\{N\}$ until it comes back, station $\{N\}$ observes on the ring the slot occupancy pattern represented in Figure 3. This figure clearly shows that the SAT cycle length varies in the range $[S, S+K]$.

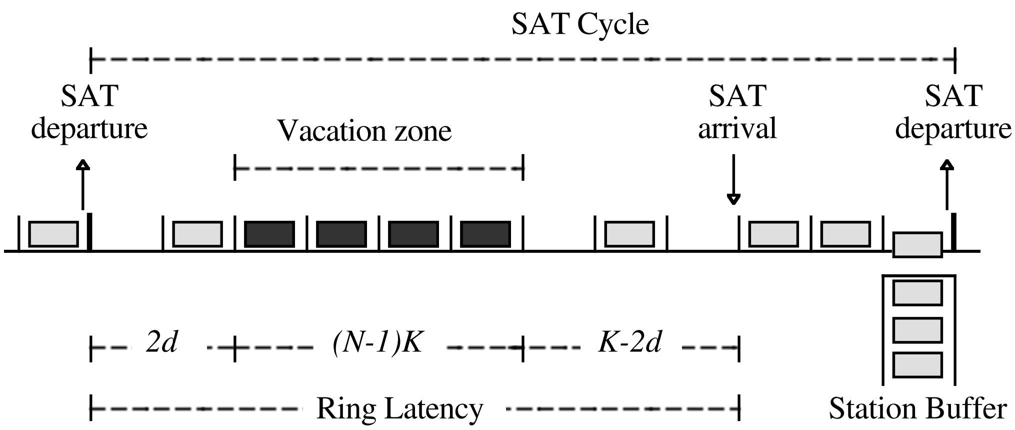

Figure 3: Slot occupancy pattern observed by station $\{N\}$

Hence, the worst-case model that will be analysed in the paper consists in one queue and one server. The service time is constant and equal to the slot duration. Furthermore, the server goes on vacation for a duration which is equal to $V = K \cdot (N-1)$ time slots. This time interval will be referred to throughout as *vacation zone* (see Figure 3).

4. Arrival Process Characterization

Generally, when modelling data traffic, packet arrivals are often assumed to be Poisson processes. However, a number of studies have shown that the distribution of packet interarrival times significantly differs from the exponential distribution [GUS90], [GUS91], [PAX94b], [LEL94], [CIN94]. Specifically, in our analysis we represent the packet interarrival times at station $\{N\}$ by means of a 2-state MMPP process[1]. Unlike renewal models, MMPP can represent correlation between interarrival times. When the Markov chain is in state $\{i\}$ $(i = 1,2)$, the arrival process is Poisson with rate λ_i. In the literature, the MMPP is often referred to as (Q,Λ) source, with Q and Λ given by

$$Q = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix}, \ \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad [3]$$

where σ_1 and σ_2 are, respectively, the transition rates from state 1 to state 2 and vice versa.

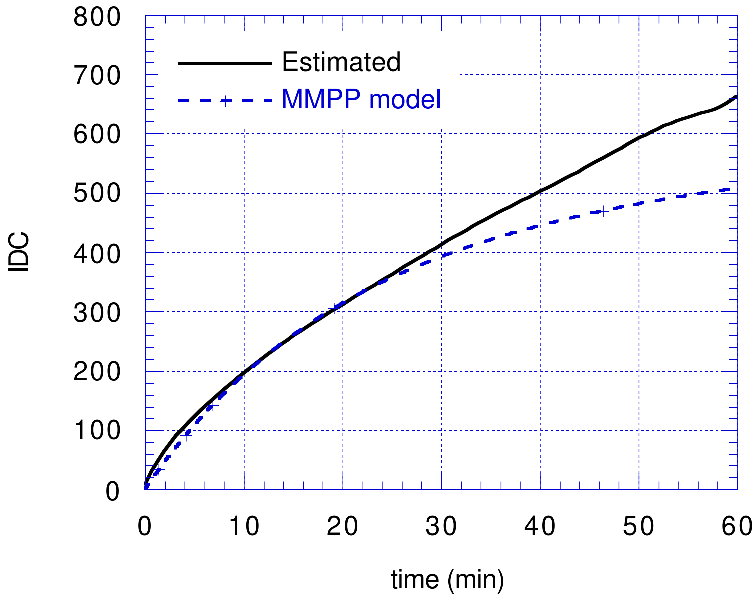

Figure 4: Indices of Dispersion for Counts for a real arrival process and for the 2-MMPP model.

For the purpose of this paper, we calculated λ_i and σ_i using the methodology reported in [GUS91] to fit an MMPP model to real packet arrival processes. According to this methodology we first estimated the Index of Dispersion for Counts (IDC)[2] by using a trace of packet arrival times obtained at Bellcore [BELL]. Then we chose the parameters of the 2-MMPP process in such way that its IDC matches the IDC of the real arrival

---

[1] MMPP is a doubly stochastic Poisson process whose arrival rate varies according to the state of a 2-state irreducible continuous time Markov chain.

[2] The IDC of an arrival process is defined as $IDC(t) = Var[N(t)]/E[N(t)]$ where $N(t)$ indicates the number of arrivals in an interval of duration t.

process at least in a wide range. With our choice of parameters the MMPP process captures the correlation between arrivals inside time intervals of about half an hour (see Figure 4).

The above MMPP is observed at the end of each time slot of duration Δ of our slotted MetaRing. Because $1/\sigma_1 >> \Delta$, and $1/\sigma_2 >> \Delta$ we model the behavior of the MMPP process as a discrete time discrete state Markov chain $C = \{C_n, n = 0,1,2,...\}$ (with transition at the end of each time slot), with state space $\{1,2\}$ and probability transition matrix

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \qquad [4]$$

where the relationship between r_{ij} and σ_1, σ_2 is

$$\begin{aligned} r_{12} &= 1 - e^{-\sigma_1 \Delta} \\ r_{21} &= 1 - e^{-\sigma_2 \Delta} \end{aligned} \qquad [5]$$

Clearly, the above Markov chain is irreducible and aperiodic.

Furthermore, in data applications of wide spread usage (file transfer, remote login, etc.), a packet is structured into more than one cell. Many studies have shown that the packet size distribution is bimodal ([GUS90]). We confirmed this by plotting the probability mass function (*pmf*) of the packet size in the Bellcore trace (Figure 5).

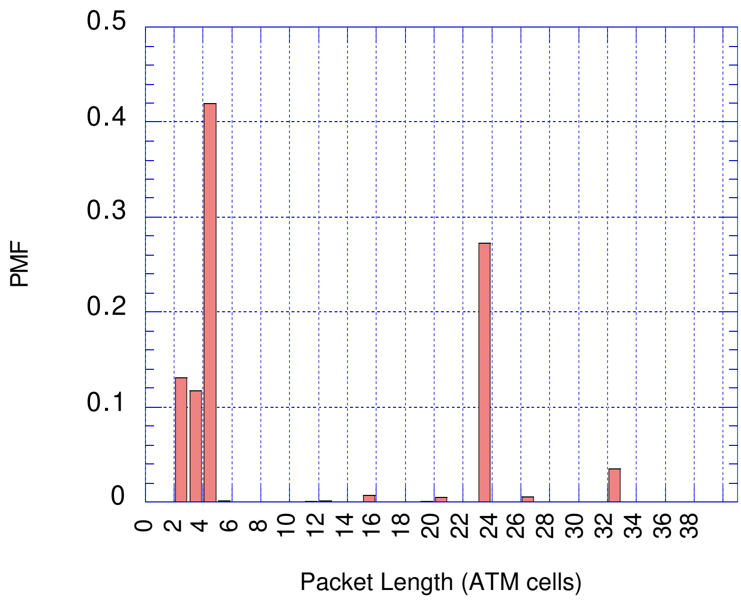

Figure 5: Packet size probability mass function.

Figure 5 clearly shows that packets can be partitioned into two classes: *short packets* and *long packets*. If we assume as short packets all those with size less than 10 ATM cells, we find that the fraction of short and long packets in the trace is 0.66 and 0.34, respectively.

To further simplify the arrival model used in our worst case model we assumed that the packet size distribution has only two possible values. The sizes of short packets (3 ATM cells) and long packets (24 ATM cells) was calculated so that the same average packet size as the real arrival process would be obtained.

The trace also shows that a packet of a given class (e.g., a short packet) is followed, with a high probability, by a packet belonging to the same class. In other words, packet trains of short and long packets alternate throughout the trace. The pmfs of the train length for short and long packets are shown in Figures 6 and 7, respectively. As a comparison, the geometric pmf is also reported in the same figures.

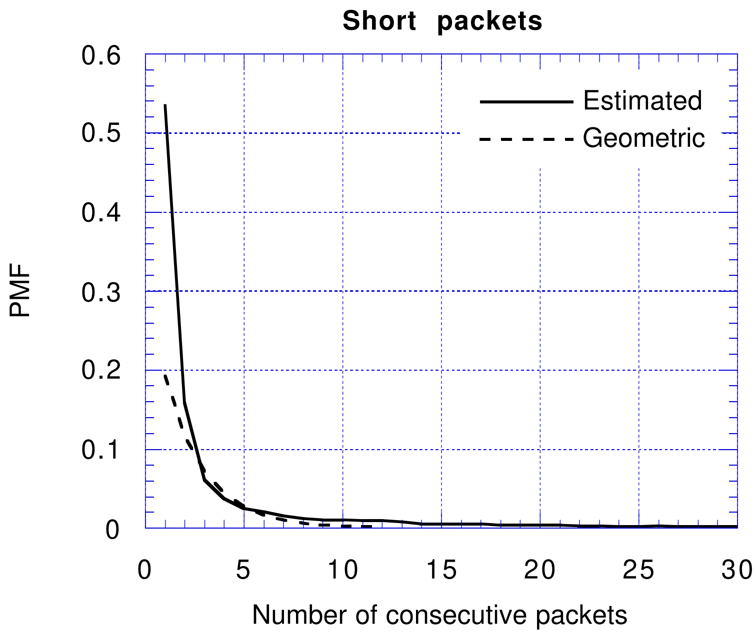

Figure 6: Probability mass function of the number of consecutive short packets.

Since there is a good fitting of the geometric to the estimated probability mass functions, the behavior of the packet size process can be modelled by a two-state irreducible and aperiodic Markov chain $J = \{J_n, n = 0,1,2,...\}$. When J is in state $\{j\}$ $(j = 1,2)$ the arrival packet (if any) has a length l_j $(l_1 = 3;\ l_2 = 24)$. The probability transition matrix of J is

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \quad [6]$$

where the elements of this matrix were estimated from the real trace.

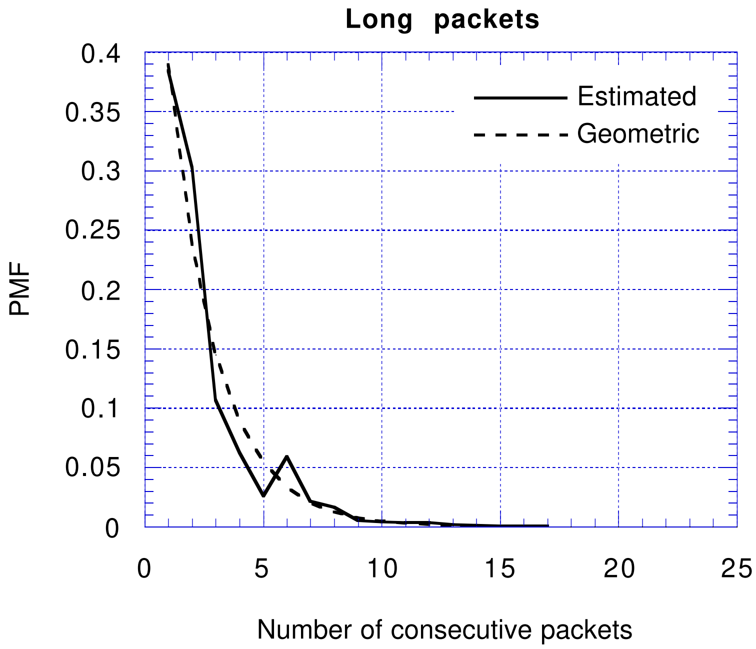

Figure 7: Probability mass function of the number of consecutive long packets.

From the above we can conclude that the arrival process can be characterized by two independent two-state Markov chains: the first modulates the packet arrival process and hence is used to derive the packets arrivals in a slot, while the second represents the number of cells contained in a packet. In order to express the arrival process in terms of the customary matrix-vector form, we introduce the tensor product $\otimes$, where $C \otimes J$ is the block matrix defined by the blocks $c_{n,m}J$. Therefore, the arrival process can be represented as $C \otimes J$, where C and J have been assumed to be independent. This results in a two dimensional irreducible and aperiodic Markov chain $T = \{(C_n, J_n), n = 0,1,2,...\}$ on the state space $\{(c, j) : c = 1,2; j = 1,2\}$ with transition probability matrix $R \otimes S$ of size 4. Hereafter we will make use of the invariant probability vector $\mathbf{x}$ (4 components) of $R \otimes S$

$$\begin{cases} \mathbf{x}(R \otimes S) & = & \mathbf{x} \\ \mathbf{x} \times \mathbf{e} & = & 1 \end{cases} \quad [7]$$

where $\mathbf{e}$ denotes the vector having all the components equal to 1, while $\times$ indicates the scalar product between vectors $\mathbf{x}$ and $\mathbf{e}$.

5. Markov Chain Description

Within any SAT cycle, we observe the system immediately after the slot boundary (*embedding points*) with the exception of the vacation zone (see Figure 8). In fact, the embedding point which follows the $2d-th$ embedding point is located after the slot which follows the vacation zone. Arrivals, if any, are assumed to occur just before the embedding points. Due to the MetaRing MAC protocol and the network parameter values (Section 3), the SAT signal comes back to the tagged station immediately before the $K-th$ embedding point. According to the MetaRing MAC protocol, on SAT arrival if the tagged station has transmitted K cells (i.e., the tagged station is satisfied), or if the tagged station is empty, the SAT is immediately forwarded to the next downstream station and a new SAT cycle begins. On the other hand, if the tagged station has transmitted less than K cells (i.e., the tagged station is not yet satisfied) and the tagged station is not empty, cell transmissions occur until either K cells are transmitted or the tagged station becomes empty, whichever occurs first. In both cases the SAT signal is forwarded to the next downstream station and a new SAT cycle begins. The longest SAT cycle is obviously when at least one cell arrival occurs in the $(K-1)th$ slot, no cells have been transmitted beforehand and the remaining $K-1$ cells arrive before the buffer becomes empty. In this case the SAT cycle includes $2K-1$ embedding points.

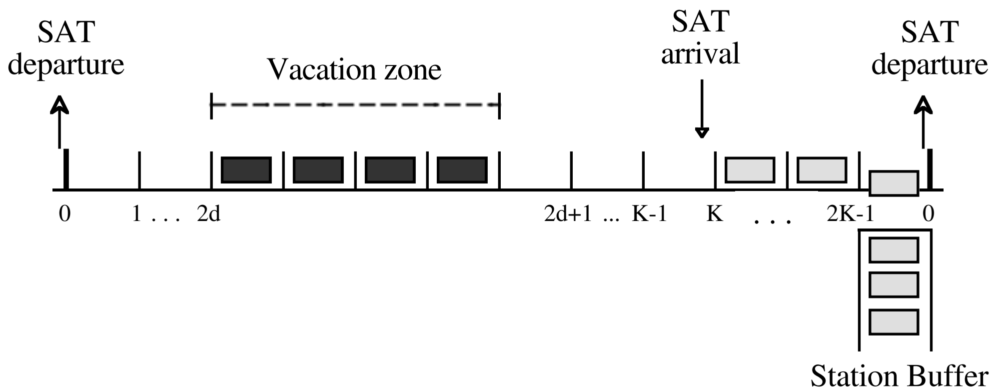

Figure 8: Maximum SAT cycle length.

To characterize the state of the system at the n-th embedding point, the following tuple of random variables is used $[Q,(S,U),(C,J)]_{(n)}$, where

- Q represents the size of the tagged station queue at the n-th embedding point.

- S is an r. v. which represents the current embedding point within a SAT cycle.
- U represents the number of cells transmitted while S is the current slot.
- C and J represent the arrival process (described in the previous section).

The MetaRing MAC protocol mechanism for transmitting cells induces constraints among the values that the above r. v. can assume. Specifically, in the portion of the SAT cycle which precedes the SAT arrival $(0 \le s \le K-1)$, the following inequality $0 \le u \le s$ must hold for u. Beyond the $(K-1)th$ embedding point, s can assume values up to $2K-1$ provided there are cells to be sent $(K \le s \le 2K-1)$. In this portion of the SAT cycle, u must increase by 1 at each embedding point up to the value of $K-1$. Therefore, u can vary in the following range: $s-K \le u \le K-1$ (see Figure 9).

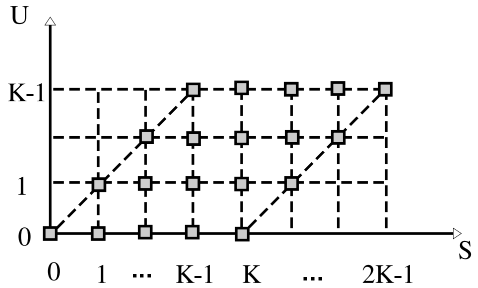

Figure 9: possible states

It thus follows that the discrete time, discrete state process $Y = \{[Q,(S,U),(C,J)]_{(n)}, n \in \boldsymbol{N}\}$ is a Markov chain with state space

$$E = [(q,s,u,c,j):\ q \in \boldsymbol{N},$$
$$\{(0 \le s \le K-1),(0 \le u \le s)\} \cup \{(K \le s \le 2K-1),(s-K \le u \le K-1)\}, c \in \{1,2\}, j \in \{1,2\}]$$

Since at each embedding point n the Q_n value decreases by at most one unit, it follows that Y is a Markov chain of $M/G/1-Type$ with a transition probabilities matrix P with the following structure

$$P = \begin{array}{c} Q_n = 0 \\ Q_n = 1 \\ Q_n = 2 \\ Q_n = 3 \\ \cdots\cdots \end{array} \begin{pmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & \cdots \\ A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \qquad [8]$$

where A_h and B_h are matrices of size $4K(K+1)$.

6. Structure of the A_h and B_h Matrices

The matrices $\left\{P^{(n)}(h), n \geq 1, h \geq 0\right\}$, which are introduced next, are basic for the study of the $Y = \left\{\left[Q,(S,U),(C,J)\right]_{(n)}, n \in \boldsymbol{N}\right\}$ process. Let us consider a group of consecutive n slots, and denote with m and $m+n$ the times (or slot numbers) at which the Markov chain (see Section 4) T is observed. $P^{(n)}_{(c,j),(c_1,j_1)}(h)$, $c,c_1 \in \{1,2\}$, $j,j_1 \in \{1,2\}$, $h \geq 0$, $n \geq 1$ denotes the conditional probability that the Markov chain T is in phase (c_1, j_1) at time $m+n$ and that h cells arrived during the interval $[m, m+n]$, given that the Markov chain T started in phase (c, j) at time m. Because T is a homogeneous Markov chain, and since arrivals in a slot are Poissonian, the number of arrivals in n consecutive slots does not depend on m. With simple stochastic considerations it is very easy to derive an expression for $P^{(1)}(h)$.

$$P^{(1)}{}_{(c,j)(c_1,j_1)}(h) = \begin{cases} 0 & \text{if } \left(h \bmod l_j\right) \neq 0 \\ \dfrac{e^{-\lambda_i}\left(\lambda_i\right)^{h/l_j}}{\left(h/l_j\right)!} r_{cc_1} s_{jj_1} & \text{if } \left(h \bmod l_j\right) = 0 \end{cases} \qquad [9]$$

for $c,c_1 \in \{1,2\}$, $j,j_1 \in \{1,2\}$, $h \geq 0$.

For a generic $n > 1$, $P^{(n)}(h)$ can be expressed in terms of $P^{(1)}(h)$ by using the Chapmann-Kolmogorov equation. Furthermore, it can be readily shown (see Appendix A) that for $n \geq 1$ the following relation holds

$$P^{(n)}(h) = \sum_{m=0}^{h} P^{(1)}(m) P^{(n-1)}(h-m) \qquad [10]$$

Using [10] we can easily verify that A_h and B_h matrices have the following structure

$$A_h\left[(\alpha,\beta),(\alpha_1,\beta_1)\right] = \begin{cases} Q(h) & \text{if } \alpha = 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta+1 \\ P^{(1)}(h) & \text{if } \beta = K-1,\ \alpha_1 = 0,\ \beta_1 = 0 \\ P^{(1)}(h) & \text{if } \alpha \neq 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta+1 \\ O & \text{otherwise} \end{cases} \qquad [11]$$

with

$$Q(h) = \sum_{i=0}^{h} P^{(V)}(i) P^{(1)}(h-i) = P^{(V+1)}(h), \qquad [12]$$

where V indicates the duration (measured in slots) of the vacation zone.

These transition probabilities are derived by partitioning all the possible transitions within a SAT cycle into the following three classes:
- transitions between the embedding points $2d \to 2d+1$ which occur with probability $Q(h)$;
- the last transition within the SAT cycle which always occurs between the embedding points $\alpha \to 0,\ \alpha \in \{K-1, K, ..., 2K-1\}$ with probability $P^{(1)}(h)$;
- all the other transitions which occur with probability $P^{(1)}(h)$.

It can be verified that A_h is reducible and can be partitioned as follows

$$A_h = \begin{bmatrix} A_h(1) & O \\ T_h(1) & T_h(0) \end{bmatrix} \qquad [13]$$

where $A_h(1)$ is a $K \times K$ block matrix , $T_h(1)$ is a rectangular $K^2 \times K$ block matrix , and $T_h(0)$ is a $K^2 \times K^2$ block matrix;. Each block in the above matrices has size 4. For ease of further computations, we write the above submatrices in the following form

$$T_h(1) = \begin{bmatrix} S_h^{(1)} \\ O \\ O \\ \cdots \\ O \end{bmatrix}, \quad T_h(0) = \begin{bmatrix} O & O & \cdots & O & O \\ S_h^{(2)} & O & \cdots & O & O \\ O & S_h^{(3)} & \cdots & O & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & S_h^{(K)} & O \end{bmatrix} \qquad [14]$$

where $S_h^{(i)} \in \mathbb{R}^{(4K)\times(4K)},\ i = 1,2,..,K$ are $K \times K$ block matrices with blocks of size 4, and, in addition, the matrices with $i = 2,..,K$ are block diagonal.

It can be readily verified that the $A = \sum_{h=0}^{\infty} A_h$ matrix is stochastic $(A\mathbf{e} = \mathbf{e})$ and has the following structure

$$A = \begin{bmatrix} A(1) & 0 \\ T(1) & T(0) \end{bmatrix} \qquad [15]$$

where matrix $A(1)$ is irreducible and matrix $I - T(0)$ is nonsingular. Specifically, matrix $A(1)$ includes the transition probabilities associated with the class of $4K$ positive recurrent states with phases that have the following form $\{(S = m, U = m), (C = c, J = j) \mid 0 \le m \le K-1, 0 \le c, j \le 1\}$. The remaining states associated with $T(1)$ and $T(0)$ are transient and communicate with the class of positive recurrent states.

Following the same line of reasoning, the transition probabilities related to B_h $(h \geq 0)$ can be readily derived.

$$B_0\left[(\alpha,\beta),(\alpha_1,\beta_1)\right] = \begin{cases} P^{(1)}(0) & \text{if } \alpha \neq 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta \\ Q^{*}(0) & \text{if } \alpha = 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta+1 \\ Z(0) & \text{if } \alpha = 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta \\ P^{(1)}(0) & \text{if } \alpha = K-1,\ \alpha_1 = 0,\ \beta_1 = 0 \\ I & \text{if } \alpha \geq K,\ \alpha_1 = 0,\ \beta_1 = 0 \\ O & \text{otherwise} \end{cases} \qquad [16]$$

$$B_h\left[(\alpha,\beta),(\alpha_1,\beta_1)\right] = \begin{cases} P^{(1)}(h) & \text{if } \alpha \neq 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta \\ Q^{*}(h) & \text{if } \alpha = 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta+1 \\ Z(h) & \text{if } \alpha = 2d,\ \alpha_1 = \alpha+1,\ \beta_1 = \beta \\ O & \text{otherwise} \end{cases} \qquad [17]$$

where

$$Q^{*}(h) = \sum_{i=1}^{h+1} P^{(V)}(i) P^{(1)}(h+1-i) \qquad [18]$$

$$Z(h) = P^{(V)}(0) P^{(1)}(h) \qquad [19]$$

It can be readily verified that the $B = \sum_{h=0}^{\infty} B_h$ matrix is stochastic $(B\mathbf{e} = \mathbf{e})$.

7. Stability Study

From [9], [11] and [12] it follows that

$$A_h(1) = \begin{array}{c} (0,0) \\ (1,1) \\ \vdots \\ (2d,2d) \\ \vdots \\ (K-2,K-2) \\ (K-1,K-1) \end{array} \left[\begin{array}{ccccccc} O & P^{(1)}(h) & O & O & O & O & O \\ O & O & P^{(1)}(h) & O & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & O & P^{(V+1)}(h) & O & O \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & O & O & O & O & P^{(1)}(h) \\ P^{(1)}(h) & O & O & O & O & O & O \end{array} \right] \qquad [20]$$

from where it can be seen that $A(1) = \sum_{h=0}^{\infty} A_h(1)$ is an irreducible and stochastic matrix with the following block periodic structure

$$
A(1) = \begin{array}{c} (0,0) \\ (1,1) \\ \vdots \\ (2d,2d) \\ \vdots \\ (K-2,K-2) \\ (K-1,K-1) \end{array}
\begin{bmatrix}
O & R\otimes S & O & O & O & O & O \\
O & O & R\otimes S & O & O & O & O \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
O & O & O & O & (R\otimes S)^{V+1} & O & O \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
O & O & O & O & O & O & R\otimes S \\
R\otimes S & O & O & O & O & O & O
\end{bmatrix} \qquad [21]
$$

The invariant probability vector $\boldsymbol{\pi}(1)$ of $A(1)$ is therefore a vector of K block components. Each block component is of size 4. Clearly, $\pi(1)$ has the following structure

$$
\pi(1) = \frac{1}{K}[\underbrace{\mathbf{x},\mathbf{x},\ldots,\mathbf{x}}_{K\ Components}] \qquad [22]
$$

where $\mathbf{x}$ is the invariant probability vector of size 4 (see Section 4) of $R\otimes S$. Since $A(1)$ is irreducible (and $G(1)$ is also irreducible, see Section 8), from [NEU89] (see page 153) it follows that the system is stable and $G(1)$ is stochastic if and only if

$$
\pi(1)\times\beta(1) < 1 \qquad [23]
$$

where the vector $\beta(1)$ is defined as:

$$
\beta(1) = \sum_{h=1}^{\infty} hA_h(1)\mathbf{e} \qquad [24]
$$

From [20] it follows that

$$
\beta(1) = \sum_{h=1}^{\infty} hA_h(1)\mathbf{e} = [\alpha,\alpha,\ldots,\hat{\alpha},\ldots,\alpha,\alpha]^{\mathrm{T}} \qquad [25]
$$

where

$$
\alpha = \sum_{h=1}^{\infty} hP^{(1)}(h)\mathbf{e},\ \ \hat{\alpha} = \sum_{h=1}^{\infty} hP^{(V+1)}(h)\mathbf{e} \qquad [26]
$$

Clearly, α and $\hat{\alpha}$ are column vectors of size 4. Each component of both vectors is associated with a phase of the arrival process. Hence, each component of α ($\hat{\alpha}$) represents the average number of arrivals in one slot (V+1 slots) given that the arrival counting process, at the beginning of the slot (vacation zone), was in that phase. Hence, from [22] and [25] it follows that

$$\pi(1)\times\beta(1)=\frac{1}{K}\left[\mathbf{x}\times\alpha+\ldots+\mathbf{x}\times\hat{\alpha}+\ldots+\mathbf{x}\times\alpha\right]<1 \qquad [27]$$

The inequality [27] has a very intuitive interpretation as it states that the system is stable iff the mean number of arrivals in S slots (the intertime - measured in slots - between two consecutive SAT arrivals at station $\{N\}$) does not exceed K. This condition is assumed to hold throughout the paper.

8. Computation of the G Matrix

Before showing the computation of matrix G [NEU89], we need to introduce the following result.

LEMMA 1. The G matrix inherits the A structure, i.e.,

$$G=\begin{bmatrix} G(1) & 0 \\ \hat{G}(1) & \hat{G}(0) \end{bmatrix} \qquad [28]$$

where:
- $G(1)$ is an irreducible $K\times K$ block matrix;
- $\hat{G}(1)$ and $\hat{G}(0)$ are $K^2\times K$ and $K^2\times K^2$ block matrices respectively, having the structures

$$\hat{G}(1)=\begin{bmatrix} G_{1,1} \\ G_{2,1} \\ G_{3,1} \\ \ldots \\ G_{K,1} \end{bmatrix}, \quad \hat{G}(0)=\begin{bmatrix} O & O & \ldots & O & O \\ G_{2,2} & O & \ldots & O & O \\ G_{3,2} & G_{3,3} & \ldots & O & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ G_{K,2} & G_{K,3} & \ldots & G_{K,K} & O \end{bmatrix}$$

with $G_{i,j}\in\mathbb{R}^{(4K)\times(4K)}$, $i,j\in(1,2,,\ldots,K)$, $K\times K$ block matrices and $G_{i,j}\in\mathbb{R}^{(4K)\times(4K)}$, $i\in(2,3,,\ldots,K)$, $j\in(2,3,\ldots,i)$ block diagonal matrices.

Each block in the above matrices has size 4.

PROOF Observe that the matrix algebra generated by the A_h matrices [13] is contained in the class $\mathcal{A}$ made up by the $(K+1)\times(K+1)$ block lower triangular matrices defined as follows

$$
R = \begin{bmatrix}
R(1) & O & O & \cdots & O & O \\
R_{1,1} & O & O & \cdots & O & O \\
R_{2,1} & R_{2,2} & O & \cdots & O & O \\
R_{3,1} & R_{3,2} & R_{3,3} & \cdots & O & O \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
R_{K,1} & R_{K,2} & \cdots & R_{K,K-1} & R_{K,K} & O
\end{bmatrix}
$$

where $R(1)$ is a $K \times K$ block matrix, $R_{i,j} \in \mathbb{R}^{(4K)\times(4K)}$, $i, j \in (1,2,,...,K)$, are $K \times K$ block matrices with blocks of size 4. Furthermore, matrices $R_{i,j} \in \mathbb{R}^{(4K)\times(4K)}$, $i \in (2,3,,...,K)$, $j \in (2,3,...,i)$ are block diagonal. The R structure is kept by matrix G since it is the limit of the sequence generated by the recurrence

$$X_{j+1} = \sum_{h=0}^{\infty} A_h X_j^h, \;\; X_0 = O$$

where X_j belongs to the closed set $\mathcal{A}$ for any $j \geq 0$. Matrix $G(1)$ is irreducible since matrices $A_h(1)$, $h \geq 0$, are irreducible.

◊

Since matrices A_h and G are reducible with the block partitioned form shown in [13] and [28] respectively, the non-linear matrix equation

$$G = \sum_{v=0}^{\infty} A_v G^v \tag{29}$$

leads to the following matrix equations

$$G(1) = \sum_{v=0}^{\infty} A_v(1) G(1)^v \tag{30}$$

$$\hat{G}(0) = \sum_{v=0}^{\infty} T_v(0) \hat{G}(0)^v \tag{31}$$

$$\hat{G}(1) = \sum_{v=0}^{\infty} T_v(1) G(1)^v + \sum_{v=1}^{\infty} T_v(0) \sum_{k=0}^{v-1} \hat{G}(0)^k \hat{G}(1) G(1)^{v-k-1} \tag{32}$$

Hence, the solution of the matrix equation [29], where G has size $4K(K+1)$, can be reduced to the solution of three matrix equations [30], [31] and [32] with lower sizes than $4K(K+1)$, which substantially speeds up the whole computation. Below, we will show how to compute [30], [31] and [32]. Specifically, we will show how to exploit the special structure of matrix $\hat{G}(0)$ to obtain a substantial reduction in the computation time of $\hat{G}(0)$ itself.

$G(1)$ COMPUTATION

In the matrix equation [30], which has the same structure as [29], the unknown $G(1)$ has size $4K$. In order to solve [30] we used the algorithm described in [BIN95a], based on the recursive state reduction which provides a quadratic convergence, has a low computational cost and is numerically stable. In Appendix B we outline this algorithm, for more details see ([BIN95],[BIN95a]).

$\hat{G}(0)$ COMPUTATION

To compute $\hat{G}(0)$ from matrix equation [31] the following auxiliary matrices are worth defining

$$T_{h,j}(1) = \begin{bmatrix} S_h^{(K-j+1)} \\ O \\ O \\ \vdots \\ O \end{bmatrix}, \qquad [33]$$

$$T_{h,j}(0) = \begin{bmatrix} O & O & \cdots & O & O \\ S_h^{(K-j+2)} & O & \cdots & O & O \\ O & S_h^{(K-j+3)} & \cdots & O & O \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & S_h^{(K)} & O \end{bmatrix}, \quad j = 2,3,...,K \qquad [34]$$

Clearly, from [14] it follows that $T_{h,K}(1) = T_h(1)$, and $T_{h,K}(0) = T_h(0)$. With the above definitions it is quite straightforward to show the following lemma

LEMMA 2. Let $\hat{G}_j(0)$ $(j = 2,3,...,K)$ be the solution of the matrix equation

$$\hat{G}_j(0) = \sum_{h=0}^{+\infty} T_{h,j}(0)\hat{G}_j(0)^h \qquad [35]$$

Then, for $j = 2,3,...,K-1$, the following relation holds

$$\hat{G}_{j+1}(0) = \left[\begin{array}{c:c} O & O \\ \hdashline \hat{G}_j(1) & \hat{G}_j(0) \end{array}\right] \qquad [36]$$

where

$$\hat{G}_j(1) = \left(I - \sum_{h=1}^{j-1} T_{h,j}(0)\hat{G}_j(0)^{h-1} \right)^{-1} T_{0,j}(1) \qquad [37]$$

PROOF

By using [34] with the appropriate index *j*, the following matrix equation

$$\hat{G}_{j+1}(0)=\sum_{h=0}^{+\infty}T_{h,j+1}(0)\hat{G}_{j+1}(0)^h,\ \ j=2,3,...,K-1$$

leads to [36], where $\hat{G}_j(0)$ is given by [35] and

$$\hat{G}_j(1)=T_{0,j}(1)+\sum_{h=1}^{j-1}T_{h,j}(0)\hat{G}_j(0)^{h-1}\hat{G}_j(1)$$

which coincides with [37].

Please note that the series in the above equation is truncated at index j-1 since, due to the structure of the matrices $T_{h,j}(0)$, it holds $T_{h,j}(0)\hat{G}_j(0)^{h-1}=0,\ \ for\ \ h\geq j$.

◊

From [35] and [34] it follows that we can compute $\hat{G}_K(0)=\hat{G}(0)$ by applying recursively [36] and [37], starting from

$$\hat{G}_2(0)=\sum_{h=0}^{+\infty}\begin{bmatrix}O & O\\ S_h^{(K)} & O\end{bmatrix}\hat{G}_2(0)^h=\begin{bmatrix}O & O\\ S_0^{(K)} & O\end{bmatrix}$$

In fact, since $\hat{G}_2(0)$ is known, from [37] we can calculate $\hat{G}_2(1)$. Hence, $\hat{G}_3(0)$ is known from [36]. Therefore, we can recursively apply the same relations to $\hat{G}_3(0),\ \ \hat{G}_4(0),.....$ until $\hat{G}_K(0)$ is computed.

Note that the computation of $\hat{G}(0)$ is reduced to perform matrix multiplications and additions, where the matrices have nonnegative entries leading to numerically stable results.

Furthermore, other relevant simplifications are obtained by observing the special structure of $T_{h,j}(1)$ (see [33]) and that $\left(I-\sum_{h=1}^{j-1}T_{h,j}(0)\hat{G}_j(0)^{h-1}\right)^{-1}$ is block lower triangular with diagonal blocks equal to the identity matrices. Hence, its block entries can be calculated using block products with substantial simplifications in the overall computation.

$\hat{G}(1)$ COMPUTATION

Once the matrices $G(1)$ and $\hat{G}(0)$ are known, the computation of the matrix $\hat{G}(1)$ in [32] can be carried out by solving a linear system of equations. In fact, if we represent a matrix A by a vector $\mathbf{a}$ obtained by row-wise arranging the entries of A, we find that the relation AXB can be rewritten as $\left(A\otimes B^T\right)\mathbf{x}$, where A,X,B are matrices of compatible sizes. Hence the condition [53] can be rewritten as

$$\left(I_{16K^3}-\sum_{i=1}^{+\infty}\left(T_i(0)^T\otimes I_{4K}\right)\sum_{h=0}^{\min(i-1,K-2)}\left(\left(\hat{G}(0)^h\right)^T\otimes G(1)^{i-1-h}\right)\right)\hat{\mathbf{g}}(1) \\ =\sum_{i=0}^{+\infty}\left(I_{4K^2}\otimes\left(G(1)^i\right)^T\right)\mathbf{t}_i(1) \qquad [38]$$

where $\mathbf{t}_i(1)$ is the vector associated with the matrix $T_i(1)$, $\hat{\mathbf{g}}(1)$ is the vector associated with $\hat{G}(1)$, and I_n denotes the $n\times n$ identity matrix. The matrix of the above system is a $K\times K$ block upper triangular matrix with blocks of size $16K^2$ and with diagonal blocks equal to the identity matrix I_{16K^2}. Also, the blocks in the upper triangular part have nonpositive entries. Thus, the solution of [38] can be directly computed by means of back-substitution. In addition, since the right-hand side of [38] has nonnegative entries, the whole computation is simply performed by means of additions of nonnegative numbers. This makes the computation fast and numerically stable.

9. Computation of $\mathbf{x}_i,\ i\geq 0$

In this section we will instance the general treatment described in Section 3.4 of [NEU89] on our problem.

COMPUTATION OF $\mathbf{x}_0$

Once G is known we evaluate the matrix

$$K = \sum_{k=0}^{\infty} B_k G^k$$

The vector $\mathbf{x}_0$ of the steady-state probabilities corresponding to level 0 is given by $\mathbf{x}_0=\kappa/\kappa\times\tilde{\kappa}_1$, where κ is the invariant probability vector of the stochastic matrix K, and $\tilde{\kappa}_1$ is defined as follows

$$\tilde{\kappa}_1=\left[\frac{\partial K(z)}{\partial z}\right]_{z=1}\cdot\mathbf{e}$$

Without repeating the general developments in [NEU89] Chapters 2 and 3, we give a few intermediate results of independent interest. By differentiating $K(z)=z\sum_{v=0}^{+\infty}B_vK(z)^v$, and bearing in mind that $G(z)=z\sum_{v=0}^{+\infty}A_vG(z)^v$ we find that

$$\tilde{\kappa}_1 = \mathbf{e} + \sum_{\nu=1}^{+\infty} B_\nu \sum_{r=0}^{\nu-1} G^r \tilde{\mu}_1$$

where vector $\tilde{\mu}_1$, defined in Section 3 of [NEU89], is given by

$$\tilde{\mu}_1 = \left[\frac{\partial G(z)}{\partial z}\right]_{z=1} \cdot \mathbf{e}$$

The differentiation of $G(z)$ immediately yields

$$\left[I - \sum_{\nu=1}^{+\infty} A_\nu \sum_{r=0}^{\nu-1} G^r\right]\tilde{\mu}_1 \;=\; \mathbf{e}$$

Following the development of Section 3.5 in [NEU89] we find that vector $\tilde{\mu}_1$ can be partitioned as follows

$$\tilde{\mu}_1 = \begin{bmatrix} \tilde{\mu}_1(1) \\ \tilde{\mu}_1(3) \end{bmatrix}$$

$$\tilde{\mu}_1(1) = \left[I - G(1) + \mathbf{e}\times\mathbf{g}_1\right]\left[I - A(1) + \left(\mathbf{e} - \beta(1)\right)\times\mathbf{g}_1\right]^{-1}\mathbf{e}$$

$$\tilde{\mu}_1(3) = \left\{\left[I - \hat{G}(0)\right]\left[I - T(0)\right]^{-1}T(1) - \hat{G}(1)\right\}\left[I - A(1) + \left(\mathbf{e} - \beta(1)\right)\times\mathbf{g}_1\right]^{-1}\mathbf{e}$$

$$+\left[I - \hat{G}(0)\right]\left[I - T(0)\right]^{-1}\frac{1}{1-\rho}\left[\beta(3) - \rho\mathbf{e}\right] + \frac{\mathbf{e}}{1-\rho}$$

where

$$\beta(3) = \sum_{\nu=1}^{+\infty} \nu T_\nu(0)\mathbf{e} + \sum_{\nu=1}^{+\infty} \nu T_\nu(1)\mathbf{e}$$

and $\mathbf{g}_1$ is the invariant probability vector of $G(1)$.

Clearly, with that partition for $\tilde{\mu}_1$ we can simplify the computation for vector $\tilde{\kappa}_1$.

COMPUTATION OF $\mathbf{x}_i$, $i \geq 1$

To compute the remaining steady state probability vector components $\mathbf{x}_i$, $i \geq 1$, we apply Ramaswami's algorithm [NEU89],[RAM88] in which $\mathbf{x}_i$, $i \geq 1$ are computed by the recursion formula

$$\mathbf{x}_i \;=\; \left[\mathbf{x}_0 \overline{B}_i + \sum_{j=1}^{i-1} \mathbf{x}_j \overline{A}_{i+1-j}\right]\left(I - \overline{A}_1\right)^{-1}, \quad i \geq 1, \qquad [39]$$

where

$$\overline{A}_v = \sum_{i=v}^{\infty} A_i G^{i-v} \text{ and } \overline{B}_v = \sum_{i=v}^{\infty} B_i G^{i-v} \text{ for } v \geq 0.$$

10. Computation of $\boldsymbol{\pi}_i,\ i \geq 0$

In this section we calculate the probability $\pi_i,\ i = 0,1,2,...$, which differs from $\mathbf{x}_i,\ i = 0,1,2,....$, since the latter refers to an embedding point whereas the former refers to the embedding points at which packets leave the system. To compute $\pi_i,\ i = 0,1,2,...$, we can observe that

$$\mathbf{x}_i[s,u,c,j] = \pi_i[s,u,c,j] + \overline{\pi}_i[s,u,c,j]$$

where $\overline{\pi}_i[s,u,c,j]$ is the probability that i cells are in the buffer at an embedding point in steady state condition and that the embedding point is not a departure point. Hence,

$$\pi_i[s,u,c,j] = \mathbf{x}_i[s,u,c,j] - \overline{\pi}_i[s,u,c,j] \qquad [40]$$

It can be verified that

$$\overline{\pi}_i[s,u,c,j] =$$
$$\begin{cases} \sum_{(l,m)} x_0[s-1,u,l,m] \cdot B_i\left[(s-1,u,l,m)(s,u,c,j)\right] & \textit{if } s \neq 0 \textit{ and } u < s \\ 0 & \textit{if } s \neq 0 \textit{ and } u = s \\ \sum_{s_1=K}^{2K-1} \sum_{u_1=s_1-K}^{K-1} \overline{\pi}_i[s_1,u_1,c,j] \ + & \\ \quad \sum_{u_1=0}^{K-1} \sum_{(l,m)} x_0[K-1,u_1,l,m] \cdot B_i\left[(K-1,u_1,l,m)(s,u,c,j)\right] & \textit{if } s = 0,\ u = 0 \textit{ and } i = 0 \\ 0 & \textit{if } s = 0,\ u = 0 \textit{ and } i > 0 \end{cases} \qquad [41]$$

By substituting [41] into [40] the probability $\pi_i,\ i = 0,1,2,...$, and (using them) the queue length distribution can easily be calculated.

11. Results

Figures 10-13 refer to the following network parameter values: $K = L = 4,\ d = 1,\ N = 50,\ S = 200$, medium capacity=$150$ $Mbit/sec$, and slot size equal to the ATM cell size.

Figure 10 shows the Complementary Distribution Function (CDF) of the queue length for several offered loads (OL) when the arrival process is MMPP and packet sizes equal to 3 and 24 cells respectively are considered.

As expected when the offered load increases, the queue length dramatically increases as well.

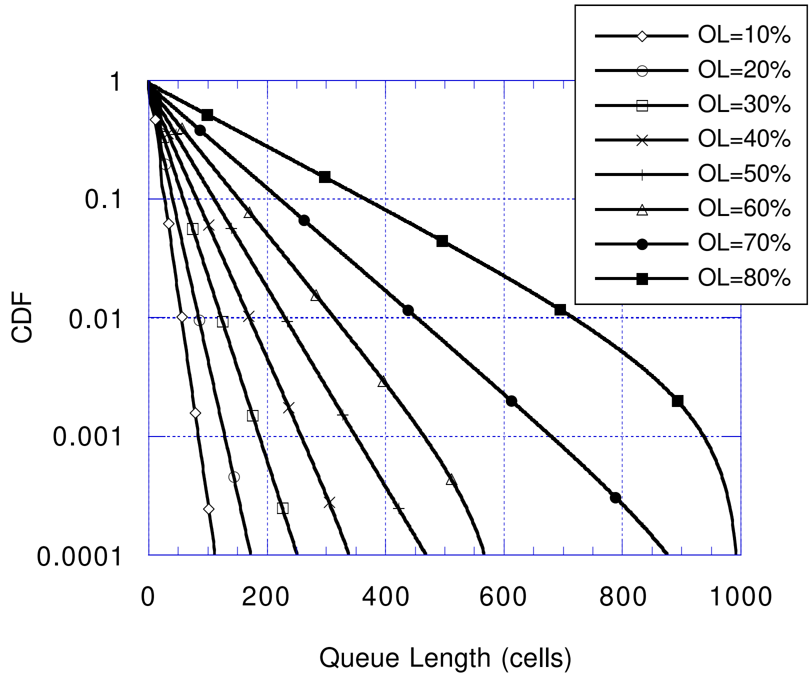

Figure 10: Complementary distribution function of the queue length. The arrival process is BMMPP with packet sizes equal to 3 and 24 cells, respectively.

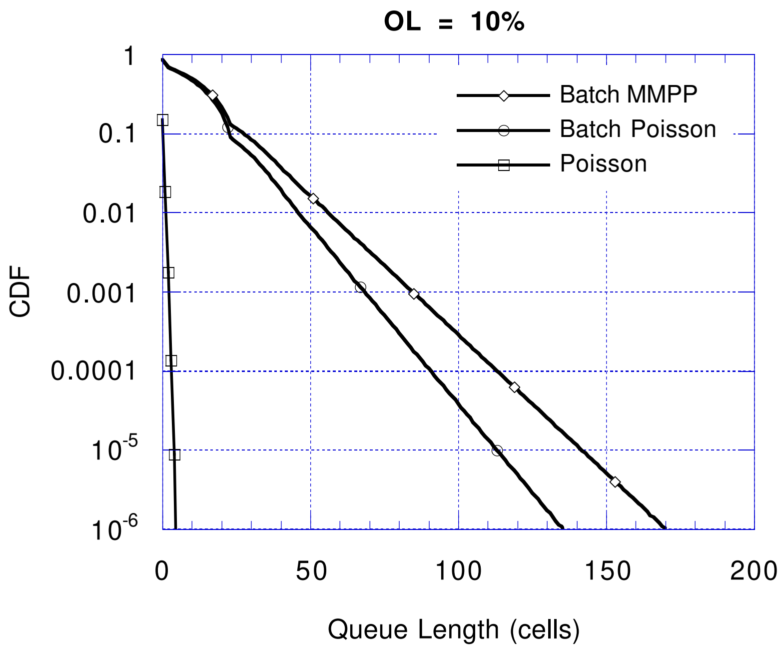

Figure 11: Complementary distribution functions of the queue length for different arrival processes.

Figures 11 to 13 compare the Complementary Distribution Functions of the queue length for BMMPP, B-Poisson, and Poisson arrival processes for several offered loads (OL=10,50,80%). These figures highlight the influence of correlation both at the batch level and at the interarrival time level. Specifically, when the arrival is simply Poisson the departure cells leave a queue length which is much smaller than when the arrival process is B-Poisson. The situation is even worse while passing from Batch Poisson to BMMPP where the correlation between successive arrivals is taken into consideration.

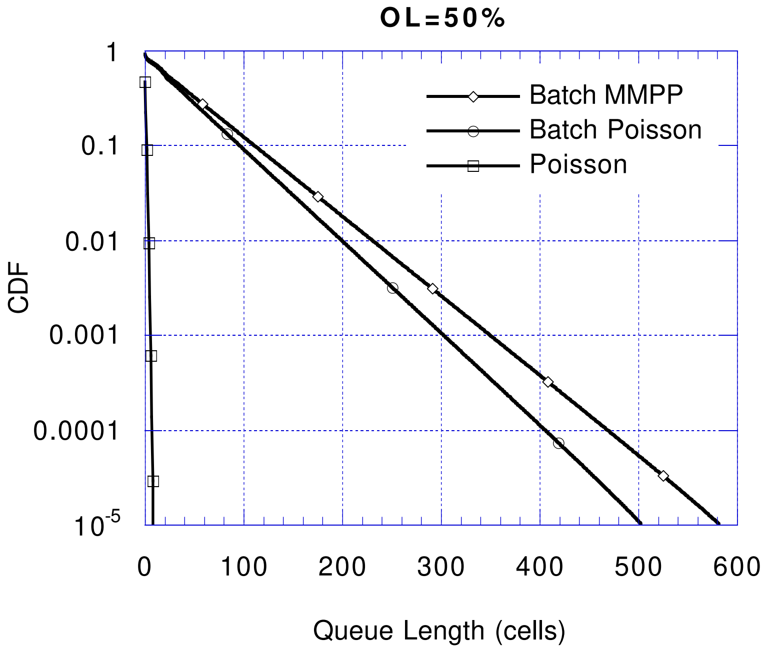

Figure 12: Complementary distribution functions of the queue length for different arrival processes.

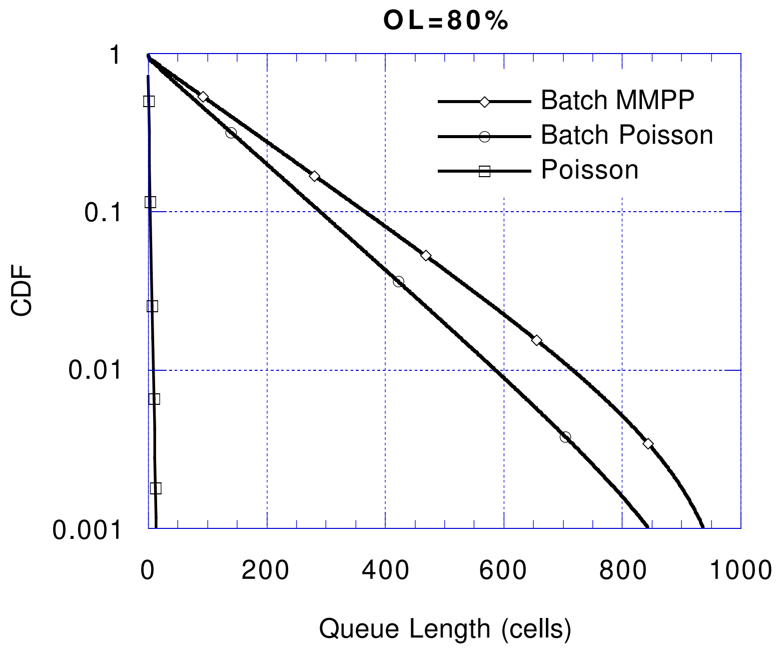

Figure 13: Complementary distribution functions of the queue length for different arrival processes.

12. Conclusions

The worst-case model proposed and solved in this paper enables us to calculate quantiles of the queue length distribution. The model proposed can be characterized by a discrete time discrete state Markov chain of *M/G/1-Type*; hence the matrix analytic methodology was used to solve it. The G matrix was computed both by taking advantage of its special structure and by adopting an innovative method which substantially reduces the computational costs. We focused on the queue length distribution of the tagged station at cell departure instants as a function of the offered load and for Poisson, Batch Poisson and BMMPP arrival processes. Our results show that the more realistic the arrival process is the longer the tail of the queue length distribution.

Appendix A

The probability of h arrivals in n slots, i.e., $P^{(n)}(h)$, is clearly given by the following expression

$$P^{(n)}(h) = \sum_{(k_1,k_2,\ldots,k_n)\in E(h,n)} \prod_{i=1}^{n} P^{(1)}(k_i) \qquad \text{[A1]}$$

where $E(h,n) = \left\{ (k_1,k_2,\ldots,k_n) \mid k_i \in \{0,1,\ldots,h\}, i \in \{1,\ldots,n\}, \sum_{i=1}^{n} k_i = h \right\}$.

If $n > 1$ [A1] can be developed as follows

$$P^{(n)}(h) = \sum_{k_n=0}^{h} P^{(1)}(k_n) \sum_{\substack{(k_1,k_2,\ldots,k_{n-1}) \\ 0\le k_i \le h-k_n \\ \sum_{i=1}^{n-1} k_i = h-k_n}} \prod_{i=1}^{n-1} P^{(1)}(k_i) = \sum_{k_n=0}^{h} P^{(1)}(k_n) P^{(n-1)}(h-k_n) \qquad \text{[A2]}$$

To implement [A2] let us refer to the two dimensional array containing $(h+1)\times n$ entries as shown in Figure A1. Each entry is a matrix of size 4. Interior elements of this array are filled in column by column. The first column in this table is initialized with matrices $P^{(1)}(j)$, $j = 0,1,\ldots,h$. From (A2) it clearly emerges that matrices related to the n-th column can be computed by using matrices belonging to column n-1. Furthermore, if the elements of the n-th column are calculated starting with the last element $P^{(n)}(h)$ and proceeding to the first element $P^{(n)}(1)$, then only $h+1$ locations (i.e., matrix of size 4) are required. Note that this algorithm is equivalent to Buzen's convolution algorithm [BUZE73].

h $n \rightarrow$ $\downarrow$	1	L	$n-1$	n	L
0	$P^{(1)}(0)$	L	$P^{(n-1)}(0)$ $\rightarrow$	$P^{(1)}(h)P^{(n-1)}(0)$ $\downarrow +$	L
1	$P^{(1)}(1)$	L	$P^{(n-1)}(1)$ $\rightarrow$	$P^{(1)}(h-1)P^{(n-1)}(1)$ $\downarrow +$	L
M					
$h-1$	$P^{(1)}(h-1)$	L	$P^{(n-1)}(h-1)$ $\rightarrow$	$P^{(1)}(1)P^{(n-1)}(h-1)$ $\downarrow +$	L
h	$P^{(1)}(h)$	L	$P^{(n-1)}(h)$ $\rightarrow$	$P^{(1)}(0)P^{(n-1)}(h)$	L

Figure A1: Convolution Algorithm

Appendix B: Computation of the G Matrix

In order to solve the problem

$$G = \sum_{k=0}^{+\infty} A_k G^k \qquad \text{[B1]}$$

we initially considered the algorithm reported in [NEU89], which is based on the following functional relation

$$X_{j+1} = \sum_{k=0}^{\infty} A_k X_j^k, \; X_0 = 0 \qquad \text{[B2]}$$

Since the sequences $\{X_j\}$ of matrices generated by [B2] *linearly* converge to the seeked solution G, for high values of K the computation slowed down significantly. We thus decided to use the algorithm described in [BIN95], based on the recursive state reduction, which provides a quadratic convergence, has a low computational cost, and is numerically stable. Below, we outline this algorithm; see ([BIN95],[BIN95a]) for more details.

Starting from $A_k^{(0)} = A_k, \; k = 0,1,2,...,$ the method generates a sequence of matrix equations

$$G_j = \sum_{k=0}^{+\infty} A_k^{(j)} G_j^k, \; j = 0,1,2,... \qquad \text{[B3]}$$

whose solution G_j is such that $G_j = G^{2^j}$, where G is the solution of [B1] and $A_k^{(j)}$ are suitable matrices which will be defined later. Matrices G_j quadratically converge to a matrix G'. This fact is used to compute the matrix G by means of the relation

$$G_0 = \left(I - \sum_{k=0}^{+\infty} \hat{A}_{k+1}^{(h)} G_h^k \right)^{-1} A_0, \qquad \text{[B4]}$$

which involves suitable matrices $\hat{A}_k^{(h)}$ related to $A_k^{(h)}$, where h is chosen so that G_h is a good approximation of G'. More precisely, it can be proved, under mild hypotheses, that the sequence of matrices $R^{(j)} = \left(I - \sum_{k=1}^{+\infty} A_k^{(j)} \right)^{-1} A_0$ converges quadratically to matrix G'. Therefore, integer h is such that the condition

$$\left| R^{(h)} - R^{(h-1)} \right| < \varepsilon E \qquad \text{[B5]}$$

is satisfied, where $\varepsilon > 0$ is fixed and E is the matrix that has all the entries equal to 1.

Matrices $A_k^{(j)}$ and $\hat{A}_k^{(j)}$ are defined in terms of the following matrix power series, which converge for $|z| \le 1$,

$$\varphi^{(j)}(z) = \sum_{k=0}^{+\infty} A_k^{(j)} z^k, \qquad \hat{\varphi}^{(j)}(z) = \sum_{k=0}^{+\infty} \hat{A}_k^{(j)} z^k, \tag{B6}$$

through the relations

$$\begin{aligned}\varphi^{(j+1)}(z) &= z\varphi_{odd}^{(j)}(z) + \varphi_{even}^{(j)}(z)\left(I - \varphi_{odd}^{(j)}(z)\right)^{-1} \varphi_{even}^{(j)}(z),\\ \hat{\varphi}^{(j+1)}(z) &= \hat{\varphi}_{odd}^{(j)}(z) + \hat{\varphi}_{even}^{(j)}(z)\left(I - \varphi_{odd}^{(j)}(z)\right)^{-1} \varphi_{even}^{(j)}(z),\end{aligned} \tag{B7}$$

where

$$\begin{aligned}\varphi_{even}^{(j)}(z) &= \sum_{k=0}^{+\infty} A_{2k}^{(j)} z^k, & \varphi_{odd}^{(j)}(z) &= \sum_{k=0}^{+\infty} A_{2k+1}^{(j)} z^k,\\ \hat{\varphi}_{even}^{(j)}(z) &= \sum_{k=0}^{+\infty} \hat{A}_{2(k+1)}^{(j)} z^k, & \hat{\varphi}_{odd}^{(j)}(z) &= \sum_{k=0}^{+\infty} \hat{A}_{2k+1}^{(j)} z^k,\end{aligned} \tag{B8}$$

and $A_k^{(0)} = A_k,\ k = 0,1,2,...,\ \hat{A}_k^{(0)} = A_k,\ k = 1,2,...$

It is possible to prove that the blocks $\hat{A}_k^{(j)} = A_k,\ k \ge 2$, converge to zero for $j \to +\infty$. This fact suggests to add to the stop condition [B5] the following one

$$\left(I - \left(I - \hat{A}_1^{(h)}\right)^{-1} A_0\right)\mathbf{e} < \varepsilon\mathbf{e}, \tag{B9}$$

where $\mathbf{e}^T = (1,1,....,1)$. If this condition is verified then the matrix G can be readily approximated by $\left(I - \hat{A}_1^{(h)}\right)^{-1} A_0$.

Matrices $A_k^{(j)}$ and $\hat{A}_k^{(j)}$ are computed very rapidly using FFT-based fast polynomial arithmetic (cf. [BIN94]) extended to the case of matrix power series ([BIN95],[BIN95a]). In fact, equation [B7] consists in computing the products and inverses of the matrix power series which, due to their convergence in the unit circle of the complex plane, are numerically reduced to polynomials since $\lim_k A_k^{(j)} = \lim_k \hat{A}_k^{(j)} = 0$.

The degree of such polynomials is bounded by the *cutting level* m_j such that matrices $\sum_{k=0}^{m_j} A_k^{(j)}$ and $A_0 + \sum_{k=0}^{m_j} \hat{A}_k^{(j)}$ are *numerically stochastic*, i.e., $\left(I - \sum_{k=0}^{m_j} A_k^{(j)}\right)\mathbf{e} < \delta\mathbf{e}$, $\left(I - A_0 - \sum_{k=0}^{m_j} \hat{A}_k^{(j)}\right)\mathbf{e} < \delta\mathbf{e}$ where δ is an upper bound to the machine precision. Matrix polynomial multiplication and inversion can be performed by means of the evaluation/interpolation technique at the roots of the unity, which is ultimately reduced to FFT's, in $O\left(p^3 m_j + p^2 m_j \log m_j\right)$ operations where p is the size of the blocks $A_k^{(j)}$. The overhead constant hidden in the $O(\cdot)$ is fairly small, in fact, the computational cost needed to carry out a single reduction step is about the same cost needed to compute seven block convolutions.

At each step j the diagonal entries of blocks $A_1^{(j)}$ and $\hat{A}_1^{(j)}$ are computed by following the diagonal adjustment technique proposed in [GRA85]. Therefore, the computation of blocks $A_k^{(j)}, \hat{A}_k^{(j)}$ is reduced to perform additions between nonnegative numbers alone, leading to the numerical stability of the algorithm.

Another nice feature of this algorithm is that, under mild conditions it holds that $\lim_k A_k^{(j)} = O,\ k \geq 2,$, that is $\lim_j m_j = 1$; this leads to a substantial speed up in the overall computation.

References

[ANA95a] G. Anastasi, L. Lenzini, P. Motta, "Performance Evaluations of a MetaRing MAC Protocol Integrating Video and Data Traffics in an Interconnected Environment", Proceedings of INFOCOM '95, Boston (USA), April 1995.

[ANA95b] G. Anastasi, L. Lenzini, "Performance Evaluation of a MetaRing MAC Protocol Carrying out Asynchronous Traffic in an Internetworking Environment", Submitted for publication.

[BELL] Trace available via anonymous ftp from `bellcore.flash.com`, file `/pub/lan_traffic/pAug.TL`.

[BIN94] D. Bini, V. Pan, Matrix and Polynomial Computations, Vol. 1: Fundamental Algorithms, Birkhauser, Boston 1994.

[BIN95] D. Bini, B. Meini, "On cyclic reduction applied to a class of Toeplitz-like matrices arising in queuing problems", in Proceedings of the Second International Workshop on Numerical Solution of Markov Chains, Raleigh, North Carolina, 1995, pp. 21-38.

[BIN95a] D. Bini, B. Meini, "On the solution of a nonlinear matrix equation arising in queuing problems", accepted for publication in SIAM J. Matrix Analysis and Applications.

[CID93] I. Cidon, Y. Ofek, "MetaRing, a Full Duplex Ring with Fairness and Spatial Reuse", IEEE Transaction on Communications, Vol. 41, No. 1, January 1993.

[CIN94] M. Cinotti, E. Dalle Mese, S. Giordano, F. Russo, "Long Range Dependence in Ethernet Traffic Offered to Interconnected DQDB MANs", Proceedings of IEEE ICCS '94, Singapore, November 1994.

[BUZE73] Buzen, J.P., "Computational Algorithms for Closed Queuing Systems with Exponential Servers", Comm. ACM 16, 9, September 1973, pp. 527-531.

[GRA85] W. K. Grassman, M. I. Taksar, D. P. Heyman, "Regenerative analysis and steady state distribution for Markov chains", Oper. Res., 33 (1985), pp. 1107-1116.

[OFE94] Y. Ofek, "Overview of the MetaRing Architecture", Computer Networks and ISDN Systems, Vol. 26, Nos. 6-8, March 1994, pp. 817-830.

[WU92] H. T. Wu, Y. Ofek, K. Sorhaby, "Integration of Synchronous and Asynchronous Traffic on the MetaRing Architecture and its Analysis", Proceedings of ICC '92.

[GUS90] R. Gusella, "Measurement Study of Diskless Workstation Traffic on an Ethernet", IEEE Transactions on Communications, Vol. COM-38, pp. 1557-1568, Sept. 1990.

[GUS91] R. Gusella, "Characterizing the Variability of Arrival Process with Indexes of Dispersion", IEEE JSAC, Vol. 9, No 2, pp. 203-211, February 1991.

[LEL94] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic", IEEE/ACM Transactions on Networking, Vol. 2, No 1, pp. 1-15, February 1994.

[NEU89] M. F. Neuts, "Structured Stochastic Matrices of M/G/1 Type and their Applications", Marcel Dekker, Inc., 1989.

[PAX94a] V. Paxson, "Empirically Derived Analytic Models of Wide Area TCP Connections", IEEE/ACM Transactions on Networking, Vol. 2, No 4, pp. 316-336, August 1994.

[PAX94b] V. Paxson, S. Floyd " Wide Area Traffic: the Failure of Poisson Modelling", Proceedings of SIGCOMM '94.

[RAM88] V. Ramaswami "A stable recursion for the steady state vector in Markov chains of M/G/l type", Stochastic Models, 4, pp. 183-188, 1988.