Aggregated bounding Markov processes applied to the analysis of tandem queues[*]

Hind Castel-Taleb
GET/INT/SAMOVAR
INT 9,rue Charles Fourier
91011 Evry Cedex, France
hind.castel@int-evry.fr

Lynda Mokdad
Lamsade, Univ. Paris Dauphine
Pl. du Maréchal de Lattre de Tassigny
75775 Cedex 16, France
mokdad@lamsade.dauphine.fr

Nihal Pekergin
LACL, Université Paris 12
61, av. du Général de Gaulle
94010 Créteil Cedex, France
nih@prism.uvsq.fr

ABSTRACT

Performance evaluation of telecommunication and computer systems is essential but a complex issue in general. Quantitative analysis of systems represented by multidimensional Markov processes models is very difficult and may be intractable if there is no specific solution form. In this study, we propose an algorithm in order to derive aggregated Markov processes providing upper and lower bounds on performance measures. We prove using stochastic comparisons that these aggregated Markov processes give bounds on performance measures defined as increasing reward functions on the transient and stationary distributions. The stochastic comparison has been largely applied in performance evaluation however the state space is generally assumed to be totally ordered which induces less accurate bounds for multidimensional Markov processes.

Our proposed algorithm assumes only a preorder on the state space, and is applied to the analysis of an open tandem queueing network with rejection in order to derive loss probability bounds. Numerical results are computed from two parametric aggregation schemes : a fine and a coarse in order to show the improvement of the accuracy of the bound with respect to the state space size. We propose an attractive solution to the performance study : given a performance measure threshold, we study if it is guaranteed or not by studying less complex aggregated bounding processes.

Keywords: Markov processes, Stochastic comparisons, Tandem queueing networks, loss probability bounds.

1. INTRODUCTION

Network architectures become very complex due to the va-

---

riety of technologies (such as ADSL, WIFI, WIMAX, satellite, ...) and the different traffic flows. It is crucial to evaluate the performance of the whole network from the source to the destination node in order to guarantee to every user an end to end QoS. These systems are usually represented by multidimensional processes with very large state spaces. As a result, quantitative analysis is difficult if there is no specific solution form (product form solutions, ...). Since exact performance measures can only be obtained using numerical methods [11] with small sizes, it is important to develop new powerful mathematical tools.

In this paper, we propose to use a mathematical method based on stochastic comparisons of Markov processes. The key idea of this method is that given a large size Markov process, we propose to bound it by a smaller new Markov process which provides bounds on performance measures.

A stochastic ordering is defined as a relation order between random variables, or stochastic processes [12]. The most known stochastic order is the strong stochastic ordering ($\preceq_{st}$). When the state space is multidimensional, weak stochastic orderings can also be defined using increasing sets families [12], [8]. The stochastic ordering theory can be also applied between processes represented on different state spaces by mapping functions on a common state space [4]. This formalism is interesting as it can be used in order to reduce the size of large state space Markov processes by defining bounding Markov processes on a smaller state space. These processes provide bounds on performance measures. We can indeed construct the aggregated process without generating the original one, thus there is no supplementary complexity and the bounding procedure can be included in the construction phase.

The advantage of this method is that it can be applied for different kinds of network architectures. We have already obtained some interesting results. In [1], we apply this method on mobile networks in order to obtain dropping handover bounds. In [3], we use it to compute loss rates packets in an optical switch, and in [2] for the loss rates packets in an IP switch. [10] presents in details this method with different applications.

These different studies lead to think the main steps of the generation of an aggregated bounding Markov process on multidimensional state spaces. In the case of totally or-

dered state spaces, the lumpability and the stochastic ordering have been combined to derive bounding Markov chains [5, 13].

The originality and the significance of the present paper is the definition of an algorithm which generates bounding Markov processes as the aggregated version of a large size multidimensional Markov process. Bounding Markov processes provide upper or lower bounds on performance measures. This algorithm can be applied for general multidimensional processes, endowed with only a preorder (so not necessarily a total order, see Definition 1 in the Appendix for more details) on the state space. We have proved using the stochastic comparison methods that the proposed aggregated Markov processes provide really bounds. These proofs are based on the comparisons of the infinitesimal generators, and the use of the monotonicity property which are complex notions when the state spaces are multidimensional and not totally ordered.

As an application of our algorithm, we evaluate the performances of a system represented by a series of network nodes (switches or routers), where only one flow of packets transits. This system represents a path in a network, from a source to a destination node. The performance study of this system is performed in order to verify that end to end Quality of Service (QoS) constraints are maintained. This system can be represented as an open tandem queueing network with rejection. We represent the tandem queues system by a Markov process which has not a product-form solution. One way of analyzing such queueing system is to solve numerically for the stationary probability vector of the Markov process. Meanwhile, memory complexity limits this approach to small queueing networks. Note that tandem queues with blocking have been analyzed in [9] using approximation algorithms for any number of queues. Because of the complexity of systems, most of the studies about tandem queues are approximations based on system decomposition.

This paper is organized as follows. In the next section, we briefly explain how to reduce the state space using stochastic comparisons. In section 3 we present the algorithm which generates aggregated bounding Markov processes. We prove in section 4 that this algorithm provides really upper and lower bounds for performance measures. In section 5, we apply the algorithm to a tandem queues system in order to evaluate the loss probabilities. Analytical results prove that the methodology is really interesting. Achieved results are discussed in section 6, and comments about further research items are given. Finally, we summarize in an appendix the stochastic ordering theory used in this paper.

2. STATE SPACE REDUCTION

The stochastic comparison is a mathematical tool which allows to compute bounds on transient distributions and the stationary distribution of a Markov process. In fact, if the underlying Markov process does not have a specific solution form like a product-form or matrix-geometric solutions, etc. the computation of stationary probability distributions becomes difficult or intractable for large state spaces. By means of the stochastic comparison method, it is possible to overcome this problem by reducing the size of the state space of the underlying Markov process.

We focus on performance measures computed as an increasing reward function on the stationary distribution. In
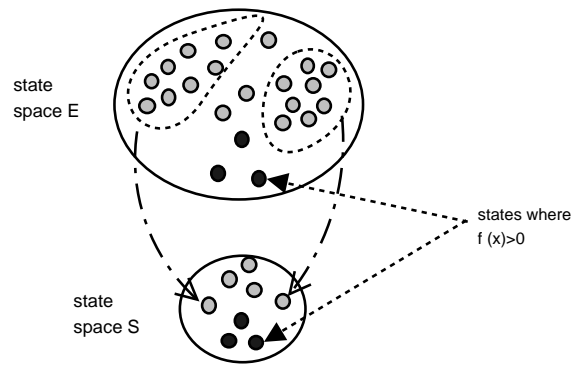

Figure 1: State space aggregation

fact we can consider transient reward functions with stochastic comparison approach in the same way, but we give here only the stationary rewards for the sake of brevity. In some cases (for example loss probabilities) this reward function depends only on few states. It equals 0 for a lot of states, and has a positive value for few states. So it is not necessary to represent all the states, and some of them can be put together (which are not used in the computation of the performance measure), in order to reduce the size of the state space. We apply the stochastic comparison approach to define a new Markov process which is an aggregation of the initial one and which provides bounds on the measure of interest.

Let $X^e(t)$ be a large state space Markov process defined on a multidimensional and preordered (not necessarily a totally ordered) state space E. We suppose that the process is irreducible so the stationary distribution Π^e exists. We denote by R^e a performance measure computed on Π^e as follows

$$R^e = \sum_{x \in E} \Pi^e(x) f(x) \qquad (1)$$

where f is an increasing reward function on Π^e according to the preorder defined on E. If there is no specific solution for Π^e, then it is very difficult to obtain it as the state space E is very large. So we reduce E by gathering some states and by mapping them into one state.

As we want to compute an upper bound (resp. a lower bound) for R^e, then we define a mapping function Ag^s (resp. Ag^l) which provides the state space S^s, where $S^s = Ag^s(E)$ (resp. S^l, where $S^l = Ag^l(E)$). We build an upper (resp. lower) "aggregated" Markov processes $X^s(t)$ (resp. $X^l(t)$) on S^s (resp. S^l), with the stationary distribution Π^s (resp. Π^l) in order to compute R^s (resp. R^l) :

$$R^s = \sum_{x \in S^s} \Pi^s(x) f(x) \text{ and } R^l = \sum_{x \in S^l} \Pi^l(x) f(x) \qquad (2)$$

with the following relation :

$$R^l \le R^e \le R^s$$

In order to have R^s and R^l close to R^e, the aggregation is performed only on some states $x \in E$ such that $f(x) = 0$. So states used for the computation of R^e are the same as in R^s and R^l , and we have supposed that $f(x)$ has the same values in R^e than in R^s and R^l. In Figure 1, we have represented the state space E as a set of grey dots (states

where $f(x) = 0$) and black dots (states where $f(x) > 0$). Some states where $f(x) = 0$ are aggregated and are mapped into one state in the state space S (where S represents S^s or S^l), and states where $f(x) > 0$ are not aggregated.

Next we will present the algorithm which generates the aggregated bounding Markov processes. We have the following assumptions. $\{X^e(t), t \geq 0\}$ is a Markov process defined on a large state space E with an infinitesimal generator matrix Q^e. It is irreducible, so the stationary distribution Π^e exists, and R^e is the performance measure to compute as given in equation (1).

We have defined an algorithm which generates aggregated Markov processes providing upper and lower bound on performance measures. This algorithm can be applied only if two conditions are verified:

1. First, we need to define on the state space E a preorder $\preceq$ compatible with R^e (R^e is written as an increasing reward function f on Π^e according to $\preceq$).

2. Secondly, $\{X^e(t), t \geq 0\}$ must be $\preceq_{st}$ monotone (see Definition 7).

We present now the algorithm which generates the aggregated Markov processes.

3. THE PROPOSED ALGORITHM

The main steps of the algorithm are the definition of the aggregated state spaces with the mapping functions, and the construction of the infinitesimal generator matrices of aggregated processes. Next, we explain how to aggregate the state space with a general mapping function Ag, and after give the particular cases for the upper bound and the lower bound.

3.1 State space definition

We explain how to derive the mapping function $Ag : E \rightarrow S$. Some states are not aggregated which means that they are exactly represented, and others are put together in order to be mapped into one state.

1. If a state x_i of E is not aggregated with other states, then it is mapped into the same state x_i. So $Ag(x_i) = x_i$ and x_i is called a "simple" state of S.

2. If the states of the set of states $\{x_1, \ldots, x_i, \ldots, x_n\}$ of E where $x_1 \preceq x_2 \ldots \preceq x_n$ are put together, then they are mapped into only one state x_i of the set. The choice of the state x_i depends if we define an upper or a lower bound. The mapping function Ag is such that $Ag(x_1) = \ldots = Ag(x_i) = \ldots = Ag(x_n) = x_i$ and x_i is called a "macro" state of S.

Note that the mapping function must be defined as an increasing function (see Definition 2) in order to have the comparison of the performance measures written as increasing reward functions on stationary distributions. The definition of an aggregated state space S can generate the problem of the irreducibility of the aggregated Markov chain although the original Markov chain is irreducible.

3.2 Transitions and irreducibility conditions

First, we explain how the transitions are defined on the aggregated state space without giving the transition rates (which are defined in the next section for the bounding aggregated Markov processes) in order to derive the irreducibility conditions.

When states of E are put together in order to be mapped into one state, some links between states are removed, so the aggregated Markov chains may be not irreducible. When the set of states $\{x_1, \ldots, x_i, \ldots, x_n\}$ are mapped into the state x_i, then states of the set of states $\{x_1, \ldots, x_n\} - \{x_i\}$ do not exist in the state space S, and they are replaced by x_i. All the transitions of the initial Markov chain to states $\{x_1, \ldots, x_n\} - \{x_i\}$ are removed and replaced by transitions to x_i in the aggregated Markov chain. And all the transitions from states $\{x_1, \ldots, x_n\} - \{x_i\}$ are removed and replaced by transitions from x_i.

Let see first the impact of the suppression of only one state, for example x_1. Let $x_p \in E$ a state which is a predecessor in a transition of x_1, and x_s a successor of x_1. As x_1 belongs to the set of states which are removed, then x_s could be not reachable from x_p, and so the aggregated Markov chain may be not irreducible. The condition to have an irreducible aggregated Markov chain is that there is an other path connecting x_p to x_s, and so x_s will be reachable from x_p. This condition must be verified for any predecessor x_p and successor x_s of x_1. It must be generalized to all predecessors x_p and successors x_s of $\{x_1 \ldots x_n\} - \{x_i\}$. If this condition is not verified for at least one state, then we come back to step 2. of the state space definition in order to define another set of states to aggregate. For the moment, in the numerical examples presented in this paper (see Section 5.) the irreducibility conditions have not been verified at this step, but only after the generation of the aggregated state space, by the Markov chain resolution algorithm.

3.3 Upper and lower bounds

State space reduction can generate a lower or an upper bound according to the choice of the state on which the set of states are mapped. For the set $\{x_1, \ldots, x_n\}$ of E where $x_1 \preceq x_2 \ldots \preceq x_n$, we have two cases :

1. if we define an upper bound, then it is mapped into the "upper" state x_n. We denote by Ag^s the mapping function which defines the upper bound. We have : $Ag^s(x_1) = \ldots = Ag^s(x_i) = \ldots = Ag^s(x_n) = x_n$ and x_n is called a "macro" state of S^s.

2. if we define a lower bound, then it is mapped into the "lower" state x_1. We denote by Ag^l the mapping : $Ag^l(x_1) = \ldots = Ag^l(x_i) = \ldots = Ag^l(x_n) = x_1$ and x_1 is called a "macro" state of S^l.

As we have defined the increasing mapping function Ag^s (resp. Ag^l), then we obtained the state space S^s, where $S^s = Ag^s(E)$ (resp. S^l where $S^l = Ag^l(E)$). We suppose that the irreducibility conditions described above are satisfied on the state space S^s (resp. S^l). Let us now explain how the transition rates of aggregated Markov processes are computed.

We introduce M_{Ag^s}, the matrix representation of the mapping function Ag^s, described in Theorem 4 of the Appendix. The infinitesimal generator Q^s is defined from Q^e and M_{Ag^s} as follows:

$$\forall x \in S^s, \ Q^s[x, *] = Q^e[x, *] M_{Ag^s}$$

where $Q^e[x, *]$ represents the row in matrix Q^e corresponding to state x. Similarly, the infinitesimal generator Q^l is

computed from Q^e and M_{Ag^l} as follows:

$$\forall x \in S^l, \ Q^l[x,*] = Q^e[x,*]M_{Ag^l}$$

The main advantage of this algorithm is to generate automatically an aggregated Markov process providing performance measure bounds. Obviously, the bounds are more accurate if Q^s and Q^l are defined as close as possible to Q^e, in order to obtain tight bounds values. However, the definition of the mapping functions Ag^s and Ag^l, which means the choice of the states to aggregate is not simple. It is performed according to different criteria:

1. States where f is not null in R^e expression (see equation 1) are not aggregated, and correspond to simple states. Oppositely to states where f is null, some states are aggregated, corresponding to macro-states.

2. Aggregation is performed so as to obtain an irreducible aggregated Markov process.

3. The choice of states to aggregate is fixed after trying different aggregation schemes in order to see their impact on the quality of the bounds, as it will be presented in section 5.

Next, we prove using the stochastic comparisons of Markov processes that the proposed algorithm really provides aggregated bounds (upper or lower).

4. PROOFS

Using the stochastic ordering theory presented in the Appendix, we prove that aggregated Markov processes generated by the algorithm represent bounds for the exact Markov process $\{X^e(t), t \geq 0\}$. So we have to verify that :

$$\{Ag^s(X^e(t)), t \geq 0\} \preceq_{st} \{X^s(t), t \geq 0\} \qquad (3)$$

and

$$\{X^l(t), t \geq 0\} \preceq_{st} \{Ag^l(X^e(t)), t \geq 0\} \qquad (4)$$

We give only the proof for Equation (3), since the second one can be similarly proved. We use Theorem 4 of the Appendix, where g represents Ag^s, h is the identity function, $\{X(t), \ t \geq 0\}$ and $\{Y(t), \ t \geq 0\}$ represent respectively $X^e(t)$ and $X^s(t)$. Let us remark that the theorems and definitions that are applied in this section are given in the Appendix.

In order to apply Theorem 4, we have to prove the condition 2) which means the monotonicity of one of the processes by mapping functions.

4.1 The monotonicity condition

We need to define the following proposition for a Markov process $\{X(t), \ t \geq 0\}$ defined on E.

PROPOSITION 1. *If the following conditions 1 and 2 are satisfied:*

1. *$\{X(t), \ t \geq 0\}$ is $\preceq_{st}$-monotone*

2. *$f : E \to S$ is an increasing function*

then $\{f(X(t)), \ t \geq\}$ is also $\preceq_{st}$-monotone

PROOF. We use Theorem 6, so we have to prove that there exist two processes $\{\widehat{X}(t), t \geq 0\}$, and $\{\widehat{X'}(t), t \geq 0\}$ governed by the same infinitesimal generator matrix than $\{X(t), t \geq 0\}$, representing different realizations of $\{X(t), t \geq 0\}$ such that :

$$f(\widehat{X}(0)) \preceq f(\widehat{X'}(0)) \Rightarrow f(\widehat{X}(t)) \preceq f(\widehat{X'}(t)), \ \forall t > 0 \quad (5)$$

As $\{X(t), t \geq 0\}$ is $\preceq_{st}$-monotone (from condition 1 of proposition 1), then according to Theorem 5, we can define two processes $\{\widehat{X}(t), t \geq 0\}$, and $\{\widehat{X'}(t), t \geq 0\}$ governed by the same infinitesimal generator matrix than $\{X(t), t \geq 0\}$, representing different realizations of $\{X(t), t \geq 0\}$ such that if:

$$\widehat{X}(0) \preceq \widehat{X'}(0)$$

then we have:

$$\widehat{X}(t) \preceq \widehat{X'}(t), \ \forall t > 0$$

Since f is an increasing function as given in Definition 2, we can deduce from the precedent inequalities that equation 5 is verified.

Thus it follows from Theorem 6, that $\{f(X(t)), t \geq 0\}$ is also $\preceq_{st}$-monotone, and Proposition 1 is proved. $\square$

We can apply Proposition 1 to our study. We have supposed as conditions of the algorithm that : $\{X^e(t), \ t \geq 0\}$ is $\preceq_{st}$-monotone, and the mapping function Ag^s is $\preceq$-increasing function. Thus with Proposition 1, we deduce that $\{Ag^s(X^e(t)), t \geq 0\}$ is also $\preceq_{st}$-monotone, and condition 2) of Theorem 4 is verified.

4.2 Infinitesimal generator comparisons

We will prove now that condition 3) of Theorem 4 is also satisfied :

$$\forall x \in E, \ y \in S^s \ | \ Ag^s(x) = y, \ Q^e M_{Ag^s}[x,*] \preceq_{st} Q^s[y,*]$$

As presented in the algorithm, we have two cases for a state $y \in S^s$:

- if y is a simple state, then y is the mapping of only the same state y of E such that $y = Ag^s(y)$, and in this case, we have according to the definition of Q^s, that:

$$Q^s[y,*] = Q^e[y,*]M_{Ag^s}$$

so for a simple state y we have that:

$$Q^e[y,*]M_{Ag^s} \preceq_{st} Q^s[y,*]$$

- if y is a macro state, then $\exists x_1, \ldots x_n \in E$ such that $Ag^s(x_1) = \ldots = Ag^s(x_n) = y$. As y represents the upper state, then if $x_n \succeq \ldots, \succeq x_1$, we have that $y = x_n$. In this case, Q^s is defined as:

$$Q^s[y,*] = Q^e[x_n,*]M_{Ag^s}$$

Since $\{Ag^s(X^e(t)), \ t \geq 0\}$ is $\preceq_{st}$-monotone, then using Theorem 7 (where f represents Ag^s), we have:

$$\begin{aligned} Q^e[x_1,*]M_{Ag^s} &\preceq_{st} Q^e[x_2,*]M_{Ag^s} \\ Q^e[x_2,*]M_{Ag^s} &\preceq_{st} Q^e[x_3,*]M_{Ag^s} \\ &\cdots \\ Q^e[x_{n-1},*]M_{Ag^s} &\preceq_{st} Q^e[x_n,*]M_{Ag^s} \end{aligned}$$

and as

$$Q^s[y, *] = Q^e[x_n, *]M_{Ag^s}$$

then:

$$\forall 1 \le i \le n, \ Q^e[x_i, *]M_{Ag^s} \preceq_{st} Q^s[y, *]$$

so for a macro state $y \in S$, we have:

$$Q^e[x, *]M_{Ag^s} \preceq_{st} Q^s[y, *], \forall x \in E \ | \ Ag^s(x) = y$$

So the inequality is established for all states $y \in S$, then from Theorem 4, if the condition 1) is satisfied, we deduce that:

$$\{Ag^s(X^e(t)), t \ge 0\} \preceq_{st} \{X^s(t), t \ge 0\}$$

The stochastic comparison of stochastic processes generates the stochastic comparison of transient and stationary (if it exists) probability distributions. We consider here only the stationary case. Thus we can deduce the stochastic comparison of the stationary probability distributions:

$$\Pi^e M_{Ag^s} \preceq_{st} \Pi^s$$

For all performance measures written as increasing reward functions f on the stationary distributions Π^e or Π^s (if we suppose that the aggregation is done on states $x \in E$ such that $f(x) = 0$), we have see(Def. 3 of Appendix)

$$R^e \le R^s$$

The proof is similar for the lower bound, so we can also verify that :

$$\forall x \in S^l \ y \in E \ | \ x = Ag^l(y) \ Q^l[x, *] \preceq_{st} Q^e[y, *]M_{Ag^l}$$

We obtain the stochastic comparison of the stationary distributions :

$$\Pi^l \preceq_{st} \Pi^e M_{Ag^l}$$

and also :

$$R^l \le R^e$$

5. EXAMPLE AND NUMERICAL RESULTS

We propose to apply the proposed algorithm to an open tandem queueing network.

5.1 Open tandem queueing network

The system understudy represents a path in a network defined as a series of network nodes (switches, routers) where transits only one flow of packets. We suppose that the leftmost node has the index 1, and indexes increase in the path until node n. This system can be represented by n finite capacity queues in tandem (see Figure 2).
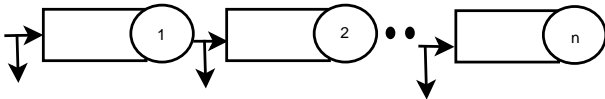

Figure 2: Tandem queueing network

External arrivals occur only in queue 1 , after the flow transits in queues $2, \dots, n$ if there is enough place in each queue. We suppose that arrivals are Poisson in queue 1 with rate λ_1. Each queue i has an Exponential service time with rate μ_i, and a finite capacity B_i. After a service in queue i, the customer transits to the next queue $i + 1$ if there is enough place, otherwise the customer is lost.

This system is represented by a Markov process $\{X^e(t), \ t \ge 0\}$ on $E = \{0, \dots, B_1\} \times \dots \times \{0, \dots, B_i\} \times \dots \times \{0, \dots, B_n\}$. Each state $x \in E$ is represented by a vector:

$$x = (x_1, \dots, x_i, \dots, x_n)$$

where x_i is the number of customers waiting in queue i. We suppose that the stationary distribution denoted Π^e exists. The goal of this performance study is to compute the loss probabilities R_i^e of any queue i written as follows:

$$R_i^e = \sum_{x \in E | x_i = B_i} \Pi^e(x) \tag{6}$$

The numerical resolution of $\{X^e(t), \ t \ge 0\}$ in order to compute Π^e is very difficult or intractable : there is no product-form, and the number of states increases exponentially with the number of components. We propose to apply the algorithm generating aggregated bounding Markov processes in order to derive loss probability bounds. Two conditions must be satisfied to apply the algorithm. These conditions are given just before the section 3 about the presentation of the algorithm.

5.2 Algorithm conditions

The first one is the definition on the state space E of an order compatible with R_i^e. We propose the component-wise partial order:

$$\forall x, \ y \in E \ x \preceq y \Leftrightarrow x_1 \le y_1, \dots, x_n \le y_n$$

We choose this preorder because it allows to establish comparisons on each queue, and it is compatible with the loss probabilities R_i^e which can be written as an increasing reward function f according to the order $\preceq$ defined on E. From expression of R_i^e (equation (6)), for a state $x \in E$, the reward function f is: $f(x) = 1$ if $x_i = B_i$, and $= 0$ otherwise, thus f is an increasing reward function according to the order $\preceq$ defined on E. Note that others performance measures as mean queues lengths, or delays are also written as increasing reward function.

The second condition is the monotonicity so we have to prove that $\{X^e(t), t \ge 0\}$ is $\preceq_{st}$-monotone.

PROPOSITION 2. *The considered tandem queue with rejection is $\preceq_{st}$-monotone.*

PROOF. We use Theorem 5 to prove that there exist two processes $\{\widehat{X}(t), \ t \ge 0\}$ and $\{\widehat{X'}(t), \ t \ge 0\}$ with the same infinitesimal generator matrix than $\{X^e(t), \ t \ge 0\}$ representing two different realizations with different initial conditions, and we prove that:

$$\widehat{X}(0) \preceq \widehat{X'}(0) \Rightarrow \widehat{X}(t) \preceq \widehat{X'}(t), \ t > 0$$

Remember that $\{X^e(t), t \ge 0\}$ is a multidimensional process on E, it is represented by the vector:

$$X^e(t) = (X_1^e(t), \dots, X_i^e(t), \dots, X_n^e(t))$$

also for $\{\widehat{X}(t), \ t \ge 0\}$ and $\{\widehat{X'}(t), \ t \ge 0\}$ which are represented by n components.

Let suppose that $\widehat{X}(t) \preceq \widehat{X'}(t)$, and show that $\widehat{X}(t+\Delta t) \preceq \widehat{X'}(t + \Delta t)$, by considering the evolution in all states even boundary states. We consider all events occurring during the time interval Δt:

• an arrival in queue 1: from $\widehat{X_1}(t)$, we obtain $\widehat{X_1}(t + \Delta t) = min\{B_1, \widehat{X_1}(t)+1\}$, and from $\widehat{X_1'}(t)$, we obtain $\widehat{X_1'}(t+\Delta t) = min\{B_1, \widehat{X_1'}(t)+1\}$, Since others components do not change, and $\widehat{X}(t) \preceq \widehat{X'}(t)$ then $\widehat{X}(t + \Delta t) \preceq \widehat{X'}(t + \Delta t)$.

• a termination of a service in queue i: obviously this occurs if $\widehat{X_i}(t) > 0$ and the customer is accepted in queue $i + 1$ if $\widehat{X_{i+1}}(t) < B_{i+1}$, otherwise it is lost. From $\widehat{X}(t)$, we obtain $\widehat{X_i}(t+\Delta t) = max\{0, \widehat{X_i}(t) - 1\}$, and $\widehat{X_{i+1}}(t+\Delta t) = min\{B_{i+1}, \widehat{X_{i+1}}(t)+1\}$. From $\widehat{X'}(t)$, similarly, $\widehat{X_i'}(t+\Delta t) = max\{0, \widehat{X_i'}(t)-1\}$, and $\widehat{X_{i+1}'}(t+\Delta t) = min\{B_{i+1}, \widehat{X_{i+1}'}(t)+1\}$. Since others components do not change, and $\widehat{X}(t) \preceq \widehat{X'}(t)$ then $\widehat{X}(t + \Delta t) \preceq \widehat{X'}(t + \Delta t)$. $\square$

As we have verified the two conditions of the algorithm, then we can apply it to compute performance measures as mean queue lengths, higher moments on queue lengths, delays, or loss probability bounds. We focus in this paper on the loss probabilities.

5.3 Bounding the loss probabilities

We use the proposed algorithm in order to derive two aggregated Markov processes:

1. The upper bound $\{X^s(t), t \geq 0\}$ on the state space $S^s \subset E$, with infinitesimal generator Q^s. We denote by Ag^s the mapping function from E to S^s.

2. The lower bound $\{X^l(t), t \geq 0\}$ on the state space $S^l \subset E$, with infinitesimal generator Q^l. We denote by Ag^l the mapping function from E to S^l.

As aggregated Markov processes are defined to be irreducible, then we can compute the stationary probability distributions Π^s and Π^l, and we obtain the following relation between the loss probabilities :

$$\forall 1 \leq i \leq n,\ R_i^l \leq R_i^e \leq R_i^s$$

where

$$R_i^l = \sum_{x \in S^l | x_i = B_i} \Pi^l(x)\ and\ R_i^s = \sum_{x \in S^s | x_i = B_i} \Pi^s(x)$$

We explain now how we define the mapping functions Ag^s and Ag^l of the upper bound and the lower bound. As it has been already stated, states where we have loss of customers are explicitly represented, but the other states can be aggregated according to different schemes. We choose two different aggregation schemes of the state space. Each one provides mapping functions Ag^s and Ag^l, in order to compute upper and lower bounds on the considered performance measure. In this analysis, we are interested in the loss probabilities of the last queue, since in terms of state space size it is the most costly one. It is possible to derive loss probabilities for other queues as we consider the component-wise ordering on the state space.

The first aggregation scheme called *Aggregation1* aggregates by considering more precisely values of the states. We can say that *Aggregation1* is a "fine aggregation" and *Aggregation2* a "coarse aggregation".

The first one is based on a parameter Δ (delta) which indicates the absolute difference between the number of packets in queues i and j. The states for which the difference between the number of packets in queues i and j is greater than Δ are aggregated to *upper* states for the upper bounds and to *lower* states for the lower bounds. The aggregation scheme for *upper* states is given in the procedure called Aggregation1 and it is presented below in this section. It is defined for three tandem queues and where we denote by $F[i]$ the number of packets in queue i, $1 \leq i \leq 3$. As we can see, several calls to the procedure Agg1 are made in order to test the difference between the components. For lower states, it is the same principle except that, if the difference between the number of packets in queues i and j is greater than Δ, then the number of packets in queue i becomes equal to the number of packets in queue $j + \Delta$. Obviously, the accuracy will be better for larger values of Δ and we have the exact process if $\Delta = B$. This aggregation scheme is interesting since it lets to find a tradeoff between the accuracy of bounds and the numerical complexity.

In this considered aggregation scheme, we suppose that all the buffers have the same size B. However, our aggregated scheme can be more general, in fact, we can have a model with different capacities at each buffer. We take parameters Δ_i as the difference of packets in queue i and $i + 1$ and we have the exact process for $\Delta_i = max(B_i, B_{i+1})$.

```
Procedure Aggregation1  (F[1], F[2],F[3])
/* F[i]: number of packets in queue i */
 Begin
    if ((F[1] <=F[2]) AND (F[2] <=F[3]))
                /*F1<=F2<=F3*/
        then Agg1(F[1], F[2], F[3])

    if ((F[1] <=F[3]) AND (F[3] <=F[2]))
                /*F1<=F3<=F2*/
        then Agg1(F[1], F[3], F[2])

    if ((F[2] <=F[1]) AND (F[1] <=F[3]))
                /*F2<=F1<=F3*/
        then Agg1(F[2], F[1], F[3])

    if ((F[2] <=F[3]) AND (F[3] <=F[1]))
                /*F2<=F3<=F1*/
        then Agg1(F[2], F[3], F[1])

    if ((F[3] <=F[2]) AND (F[2] <=F[1]))
                /*F3<=F2<=F1*/
        then Agg1(F[3], F[2], F[1])

    if ((F[3] <=F[1]) AND (F[1] <=F[2]))
                /*F3<=F1<=F2*/
        then Agg1(F[3], F[1], F[2])
 End
End Aggregation1

Procedure Agg1 ( X1, X2, X3)
/*  X1<=X2<=X3*/
   Begin
    if ((X3-X2) >delta)
                then X2=X3-delta;
                    if ((X2-X1)>delta)
                        then X1=X2-delta;
End Agg1
```

Aggregation1 procedures

The second scheme is based on two parameters max and Δ (delta), where for any state, the max is the maximum of the number of packets in all queues. The states for which the difference between the number of packets in queue i and max is greater than Δ are aggregated to *upper* states for the upper bounds. The aggregation scheme for the upper bound is given in the following procedure called Aggregation2.

```
Procedure Aggregation2 (In Out F[0], F[1],F[2])
 Begin
    max=Max(F[0], F[1], F[2]);
    if (F[0]< max-delta)
        F[0]=max-delta;
    if (F[1]< max-delta)
        F[1]=max-delta;
    if (F[2]< max-delta)
        F[2]=max-delta;
 End Aggregation2
```

Aggregation2 procedure

For lower states, it is the same principle except that, we do not consider the maximum but the minimum of the packet number of all queues. To illustrate these aggregation schemes, we give an example: if we have the state $(5, 10, 13)$ which means that in queue 1, we have 5 packets, in queue 2, we have 10 packets and 13 packets in queue 3 and $\Delta = 2$. With the *Aggregation1*, the *upper* state is $(9, 11, 13)$ and with *Aggregation2*, the *upper* state is upper than the precedent: it is $(11, 11, 13)$, which shows that the first aggregation is finer than the second one.

We give numerical results for the model with four buffers in tandem. We suppose that the service rate μ_i in each queue is $100Mb$, and the packet size is 512 bytes. We plot the Packet Loss Probabilities (PLP) of the last queue for input bit rate λ varying from 50 Mb/s to 90 Mb/s. We construct directly by means of evolution equations the discrete time bounding Markov chains which are the uniformized versions for the proposed aggregation schemes.

First we consider a system with capacities $B_i = B = 20$ for $1 \leq i \leq 4$ in order to compare the exact values with the bounds. The size of the exact Markov process is 194481 and we take different values of $\Delta = \{10, 15\}$. Let us give the sizes of the bounding chains for the considered aggregation schemes. For *Aggregation 1*, with $\Delta = 10$, the size of the aggregated Markov process is 158071 while with $\Delta = 15$, the size is 191751. For *Aggregation2*, with $\Delta = 15$, the size of the aggregated Markov process is 140091 while with $\Delta = 10$, the size is 61051.

In **Figures 3, 4**, we present upper bounds on loss probabilities obtained by *Aggregation 1* and *Aggregation 2* with $\Delta = 10$ and $\Delta = 15$ under different arrival throughputs. Clearly, the loss probabilities obtained with $\Delta = 10$ provide upper bounds on the loss probabilities obtained with $\Delta = 15$ for both aggregation schemes. Thus the bounds obtained with $\Delta = 15$ are more accurate. Moreover we can see that for *Aggregation 2*, the difference between the bounds obtained with $\Delta = 10$ and $\Delta = 15$ decreases when the arrival throughput increases. *Aggregation 1* provides tighter bounds than *Aggregation 2* as expected, in the expense of larger bounding chains.

In **Figure 5**, we give the lower and upper bounds of loss probabilities by the aggregated model with $\Delta = 15$, and

the exact values to show the quality of the bounds. Clearly, *Aggregation1* scheme provides interesting results.
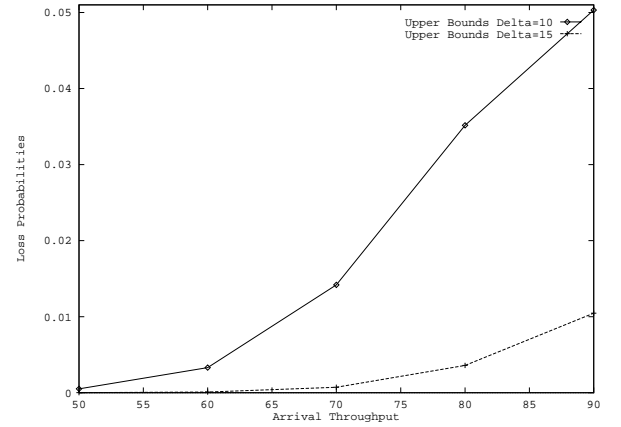

Figure 3: Upper bounds for Aggregation1
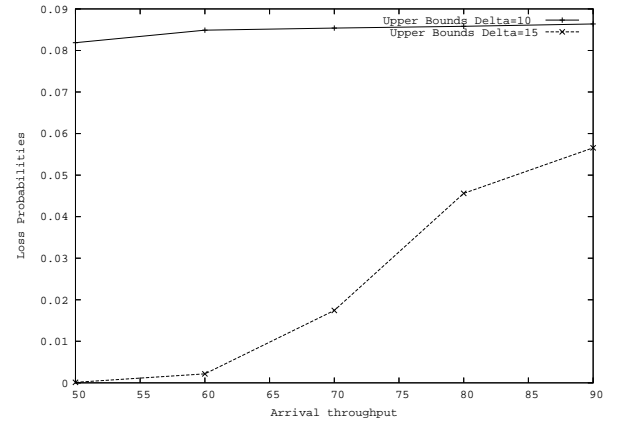

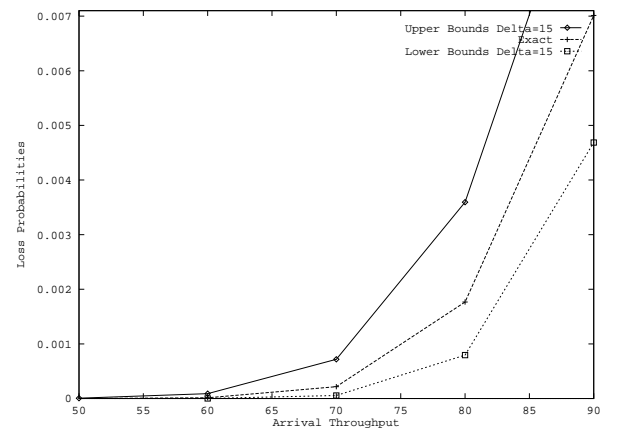**Figure 4: Upper bounds for Aggregation2**

Figure 5: Upper and Lower bounds and Exact values for Aggregation1

In **Table 1**, we present the Relative Errors (ER) defined as the ratio of the absolute error to the exact value (

$|exact - UpperBound|/exact$) computed both with *Aggregation1* and *Aggregation2* for the packet loss probabilities. We can see that for both schemes the accuracy increases with the arrival throughput and this is more important for *Aggregation 2*. *Aggregation 1* provides generally very tight bounds.

B	Δ	λ	RE Aggregation1	RE Aggregation2
20	15	50	13.48	274.36
20	15	60	5.31	151.38
20	15	70	2.30	79.18
20	15	80	1.03	24.81
20	15	90	0.49	7.06

Table 1: Relative Errors (RE)

In the following, we present only the upper bounds for the model with four buffers in tandem. We consider both aggregations with buffer sizes equal to 40 and two values of Δ, $\Delta = 15$ and $\Delta = 20$.

In **Table 2** , we give upper packet loss probabilities for buffer sizes equal to 40 and $\Delta = 20$ where the size of the exact Markov chain is 2825761. The size of aggregated chain with *Aggregation1* is equal to 2296141 and with *Aggregation2* it is equal to 884101.

B	Δ	λ	PLP Aggregation1	PLP Aggregation2
40	20	50	$33.92\ 10^{-8}$	$11.54\ 10^{-3}$
40	20	60	$12.87\ 10^{-6}$	$24.56\ 10^{-3}$
40	20	70	$28.24\ 10^{-5}$	$42.91\ 10^{-3}$
40	20	80	$38.62\ 10^{-4}$	$44.24\ 10^{-3}$
40	20	90	$20.09\ 10^{-3}$	$44.58\ 10^{-3}$

Table 2: Packet Loss Probabilities (PLP)

In **Table 3**, we give upper packet loss probabilities for buffer sizes equal to 40 and $\Delta = 15$ where the size of the exact Markov chain is 2825761. The size of aggregated chain with *Aggregation1* is equal to 1587311 and with *Aggregation2* it is equal to 438311.

B	Δ	λ	PLP Aggregation1	PLP Aggregation2
40	15	50	$55.92\ 10^{-5}$	$52.81\ 10^{-3}$
40	15	60	$63.44\ 10^{-4}$	$58.19\ 10^{-3}$
40	15	70	$32.04\ 10^{-3}$	$58.25\ 10^{-3}$
40	15	80	$43.59\ 10^{-3}$	$58.43\ 10^{-3}$
40	15	90	$62.70\ 10^{-3}$	$58.86\ 10^{-3}$

Table 3: Packet Loss Probabilities (PLP)

The choice of the aggregation scheme and the value of Δ lets to find a tradeoff between the accuracy of the bounds and the complexity of numerical resolution. We can start with the less expensive scheme and try to see if the constraint on the loss probability is satisfied or not : if the upper bounding value is less than the imposed constraint, then the system would satisfy this constraint. Otherwise we can increase the value of Δ or apply *Aggregation 1* to derive tighter bounds. Thus, this approach seems promising to study if the constraints on QoS are guaranteed or not for large Markovian models. We have only considered loss probabilities, but

other performance measures defined as increasing rewards on queue length distributions can be considered. As mentioned before we obtain transient distribution comparisons, thus transient performance measures may be studied by this approach. Note that we have proposed in this paper particular aggregation schemes, others must be tried in order to see the impact on the quality of the bounds.

6. CONCLUSION

We apply stochastic comparisons methods in order to reduce the size of large state Markov processes. We define an algorithm which generates aggregated Markov processes in order to compute upper and lower performance measures bounds. Our algorithm has been applied to the analysis of an open tandem queueing network with rejection. Loss probability bounds have been computed to illustrate the feasibility and the efficiency of the proposed methodology. Different aggregation schemes can be performed by this approach since the constraints on the aggregated states are not very restrictive. Thus we can have a tradeoff between the accuracy of bounds and the computation complexity. As a next work, we are applying the proposed algorithm to the computation of others performance measures as end to end mean delays. We are also applying it to other communication systems in order to see if it is sufficiently general for the quantitative analysis of multidimensional Markov processes.

APPENDIX

We present in this appendix some theorems and definitions about stochastic orderings used in proofs presented in this paper.

Two formalisms can be used for the definitions: increasing functions [12], [4] or increasing sets [8] .

The $\preceq_{st}$ ordering is the most known stochastic ordering, it is equivalent to the sample path ordering (see Strassen's theorem [12]). Stochastic orderings are defined only on discrete and countable state space E, where a binary relation $\preceq$ is defined at least as a preorder [12]:

DEFINITION 1. $\preceq$ *is called a preorder if and only if it is a reflexive, and transitive relation.*

If $\preceq$ has also the anti-symmetric property, then it is called a partial order, and it is a total order if it verifies also

$$\forall x, y \in E,\ x \preceq y\ or\ y \preceq x$$

As an example, on the state space $E = \mathbb{R}^n$, componentwise order is a partial order, and on $E = \mathbb{R}$, $\leq$ is a total order.

In the sequel, $\preceq$ denotes at least a preorder on E. We consider two random variables X and Y defined respectively on E, and their probability measures given respectively by the probability vectors p and q where $p[i] = Prob(X = i)$, $\forall i \in E$ (resp. $q[i] = Prob(Y = i)$, $\forall i \in E$). The $\preceq_{st}$ ordering can be defined using real increasing functions. Let us give first the definition of an increasing function taking values in any state space S.

DEFINITION 2. *We say that $f : E \mapsto S$ is $\preceq$-increasing if and only if: $\forall x, y \in E,\ x \preceq y \Rightarrow f(x) \preceq f(y)$*

The definition of the $\preceq_{st}$ is [12]:

DEFINITION 3. $X \preceq_{st} Y \Leftrightarrow E[(f(X))] \preceq E[(f(Y))] \forall f : E \rightarrow R$, $\preceq$-increasing whenever the expectations exist.

Different methods are associated to the $\preceq_{st}$ ordering: the coupling [12], [6], or the increasing set theory [8].

First, we present the coupling theorem (from Strassen's theorem) [12], [6]:

THEOREM 1. We say that $X \preceq_{st} Y$, if and only if there exists a random variable $\widehat{X}$ (resp. $\widehat{Y}$) with the same probability distribution as X (resp. Y), such that

$$\widehat{X} \preceq \widehat{Y}$$

almost surely.

Increasing set method is a more general formalism and and lets to define different stochastic orderings in the case of multidimensional and partially ordered state spaces. Indeed the $\preceq_{st}$ ordering can be defined, but also other orderings called weak orderings: $\preceq_{wk}$, $\preceq_{wk*}$ [8]. We focus in this paper only on the $\preceq_{st}$ ordering. The key idea is to define a stochastic ordering from a family of increasing sets [8]. Let $\Gamma \subseteq E$, we denote by $\Gamma \uparrow = \{y \in E \mid y \succeq x, x \in \Gamma\}$.

DEFINITION 4. Γ is called an increasing set if and only if $\Gamma = \Gamma \uparrow$

Let $\Phi_{st}(E)$ the family of increasing sets which induces the $\preceq_{st}$ ordering:

$$\Phi_{st}(E) = \{\text{all increasing sets on } E\}$$

The $\preceq_{st}$ ordering theorem using the increasing set theory [8] states as follows:

THEOREM 2.
$$X \preceq_{st} Y \Leftrightarrow p \preceq_{st} q \Leftrightarrow p(\Gamma) \leq q(\Gamma), \forall \Gamma \epsilon \Phi_{st}(E)$$

where

$$p(\Gamma) = \sum_{x \in \Gamma} p(x)$$

We present now the comparison of stochastic processes. Let $\{X(t), t \geq 0\}$ and $\{Y(t), t \geq 0\}$ be stochastic processes defined on E.

DEFINITION 5. We say that

$$\{X(t), t \geq 0\} \preceq_{st} \{Y(t), t \geq 0\}$$

if $X(t) \preceq_{st} Y(t), \forall t \geq 0$

Methods as increasing sets and coupling can also be used for Markov processes. Here we give the theorem of the coupling of the processes [6], [12]. Two processes are defined in this theorem:

- Let $\left\{\widehat{X}(t), t \geq 0\right\}$ have the same infinitesimal generator as $\{X(t), t \geq 0\}$

- and $\left\{\widehat{Y}(t), t \geq 0\right\}$) have the same infinitesimal generator as $\{Y(t), t \geq 0\}$

THEOREM 3. We say that

$$\{X(t), t \geq 0\} \preceq_{st} \{Y(t), t \geq 0\}$$

if and only if there exists the coupling $\{(\widehat{X}(t), \widehat{Y}(t)), t \geq 0\}$ such that:

$$\widehat{X}(t) \preceq \widehat{Y}(t), \forall t \geq 0$$

When the processes are defined on different states spaces we can compare them on a common state space using mapping functions. Let :

- $X(t)$ (resp. $Y(t)$) defined on E (resp. F),

- g (resp. h) be a many to one mapping from E (resp. F) to S

The stochastic comparisons of these processes by mapping functions is [4]:

DEFINITION 6. We say that

$$\{g(X(t)), t \geq 0\} \preceq_{st} \{h(Y(t)), t \geq 0\}$$

if $g(X(t)) \preceq_{st} h(Y(t)), \forall t \geq 0$

For processes defined on different states spaces, Theorem 3 can be reformulated [4]. We present in the sequel only the increasing set theory using infinitesimal generators matrices because it is the formalism developed in the algorithms presented in this paper. Some times, the comparison of the processes uses the monotonicity property in order to establish the comparison of the infinitesimal generators.

The monotonicity of a stochastic process is defined as an increasing in t [6] :

DEFINITION 7. We say that $\{X(t), t \geq 0\}$ is $\preceq_{st}$ -monotone if $X(s) \preceq_{st} X(t), \forall s, t \in \mathbb{R}^+, s \leq t$

If we suppose that $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$) is a Markov process with infinitesimal generator matrix Q_1 (resp. Q_2), then we present the theorem of the stochastic comparison of Markov processes defined on different state spaces using increasing set formalism [8], [12].

The mapping functions are represented by a matrix formalism as follows. Let M_g (resp. M_h) denote the matrix representing the mapping g (resp. h).

$$M_g[i, j]_{i \in E \text{ and } j \in S} = \begin{array}{ll} 1 & \text{if } g(i) = j \\ 0 & \text{else} \end{array}$$

and M_h is similar, using h instead of g, and F instead of E.

THEOREM 4. If the following conditions 1, 2 and 3 are satisfied:

1. $g(X(0)) \preceq_{st} h(Y(0))$

2. $\{g(X(t)), t \geq 0\}$ or $\{h(Y(t)), t \geq 0\}$ is $\preceq_{st}$-monotone

3. $Q_1[x, *]M_g \preceq_{st} Q_2[y, *]M_h, \forall x \in E, y \in F, g(x) = h(y)$

then we have:

$$\{g(X(t)), t \geq 0\} \preceq_{st} \{h(Y(t)), t \geq 0\}$$

where $Q_1[x, *]$ is the row in the matrix Q_1 corresponding to the state x. And,

$$Q_1[x, *]M_g \preceq_{st} Q_2[y, *]M_h$$

is equivalent to: $\forall x \in E, y \in F \mid g(x) = h(y)$

$$\sum_{g(z) \in \Gamma} Q_1(x, z) \leq \sum_{h(z) \in \Gamma} Q_2(y, z), \ \forall \Gamma \in \Phi_{st}(S)$$

Note that the stochastic comparison of Markov processes by mapping functions can be interesting for reducing the state space size of Markov processes, in order to define bounding aggregated Markov processes as we will see in this paper. The monotonicity of the Markov process is used in condition (2) of this theorem. If we suppose that $X(t)$ is a Markov process, then to establish the monotonicity of a process $\{X(t), t \geq 0\}$, we can use the coupling of the processes [6], [7]. As presented in [6], [7], it remains to define two processes:

$$\{\widehat{X}(t), \ t \geq 0\} \text{ and } \{\widehat{X'}(t), \ t \geq 0\}$$

governed by the same infinitesimal generator matrix as $\{X(t), t \geq 0\}$, and representing different realizations and initial conditions. The theorem of the monotonicity using the coupling states as follows:

THEOREM 5. *We say that $\{X(t), t \geq 0\}$ is $\preceq_{st}$ -monotone if and only if there exists the coupling $\{(\widehat{X}(t), \widehat{X'}(t)), \ t \geq 0\}$ such that:*

$$\widehat{X}(0) \preceq \widehat{X'}(0) \Rightarrow \widehat{X}(t) \preceq \widehat{X'}(t), \ \forall t > 0$$

This theorem may also hold when infinitesimal generators are different, for the comparison of Markov processes by coupling [7], [6], [12] as presented in Theorem 3. When we study the mapping of a process, we can also define the monotonicity, and it is formulated as follows [12], [6]:

DEFINITION 8. *We say that $\{f(X(t)), t \geq 0\}$ is $\preceq_{st}$ -monotone if $f(X(s)) \preceq_{st} f(X(t)), \forall s, \ t \in \mathbb{R}^+, \ s \leq t$*

The coupling also holds for the monotonicity of the mapping of a Markov process [12], [6]. As presented in [6], [7], it remains to define two processes:

$$\{\widehat{X}(t), \ t \geq 0\} \text{ and } \{\widehat{X'}(t), \ t \geq 0\}$$

governed by the same infinitesimal generator matrix than $\{X(t), t \geq 0\}$, representing different realizations of $\{X(t), t \geq 0\}$ with different initial conditions. And the coupling states as follows for the monotonicity of the mapping of a Markov process:

THEOREM 6. *We say that $\{f(X(t)), t \geq 0\}$ is $\preceq_{st}$ -monotone if and only if there exists the coupling $\{(\widehat{X}(t), \widehat{X'}(t)), \ t \geq 0\}$ such that:*

$$f(\widehat{X}(0)) \preceq f(\widehat{X'}(0)) \Rightarrow f(\widehat{X}(t)) \preceq f(\widehat{X'}(t)), \ \forall t > 0$$

The increasing set formalism can be used to formulate the monotonicity of the mapping of a Markov process [12], [6].

THEOREM 7. *We say that $\{f(X(t)), t \geq 0\}$ is $\preceq_{st}$ -monotone if and only if:*

$$Q_1[x, *]M_f \preceq_{st} Q_1[y, *]M_f, \ \forall x, y \in E \mid f(x) \preceq f(y)$$

A. REFERENCES

[1] H.Castel, L. Mokdad, "Performance measure bounds in wireless networks by state space reduction", 13th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2005), 27-29 september, Atlanta Georgia.

[2] H.Castel, L. Mokdad, N .Pekergin, "Loss rates bounds for IP switches in MPLS networks", ACS/IEEE International Conference on Computer Systems and Applications AICCSA-06, 8-11 March 2006, Dubai/Sharjah, UAE.

[3] H. Castel, J.M. Fourneau, N. Pekergin, "Stochastic bounds on partial ordering: application to memory overflows due to bursty arrivals", 20th International Symposium on Computer and Information Sciences (ISCIS 2005), October 26-28 2005, Istanbul, Turkey, published in LNCS by Springer-Verlag.

[4] M.Doisy, "Comparaison de processus Markoviens", PHD thesis, Univ. de Pau et des pays de l'Adour 92.

[5] J. M. Fourneau, N. Pekergin, "An algorithmic approach to stochastic bounds", in Performance Evaluation of Complex Systems: Techniques and Tools, LNCS 2459, 2002.

[6] T.Lindvall, "Lectures on the coupling method", Wiley series in Probability and Mathematical Statistics, 1992.

[7] T. Lindvall, "Stochastic monotonicities in Jackson queueing networks", Prob. in the Engineering and Informational Sciences 11, 1997, 1-9.

[8] W. Massey, "Stochastic orderings for Markov processes on partially ordered spaces" Mathematics of Operations Research, Vol.12, No. 2, May 1987.

[9] H.G. Perros, "Queueing networks with blocking, exact and approximate solutions", Oxford University Press, 1994.

[10] N. Pekergin, "Stochastic performance bounds by state space reduction", Performance evaluation, 36-37, (1-17), 1999.

[11] W.J. Stewart,"An Introduction to the Numerical Solution of Markov Chains", Princeton, 1993.

[12] D. Stoyan, "Comparison methods for queues and other stochastic models", J. Wiley and Sons, 1976.

[13] L. Truffet, "Reduction technique for discrete time Markov chains on totally ordered space using stochastic comparisons", J. App. Prob. 37 (3), 2000.

[14] L. Zheng, L. Zhang, "Modeling and performance analysis for IP traffic with multi-class QoS in VPN", Milcom2000, 21st Century Military Communications Conference Proceedings, Vol 1, 22-25 Oct. Page 330-334,???