View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Institutional Research Information System University of Turin

Noname manuscript No.
(will be inserted by the editor)

Exploiting Flow Equivalent Server in Transient Analysis

Received: date / Accepted: date

Abstract In this paper we deal with approximate transient analysis of complex queuing networks. In particular, we exploit the idea of flow equivalence to reduce the size of the model. It is well-known that flow equivalent servers lead to exact steady state solution in many cases.

Our goal is to investigate the applicability of flow equivalence to transient analysis. We show that exact results can be obtained even in the transient phase, but the definition of the equivalent server requires the analysis of the whole original network. We propose thus to use approximate aggregate servers whose characterization demands much less computation. Specifically, the characterization corresponds to the steady state equivalent server of the stations that we aim to aggregate and thus can be achieved by analyzing the involved stations in isolation. This way, approximations can be derived for any queuing network, but the precision of the results depends heavily on the topology and on the parameters of the model.

We propose a methodology in which we check the aggregate servers in order to decide whether the resulting approximate transient analysis is satisfactory. We motivate the approach by several numerical case studies.

Keywords Queuing networks, transient analysis, flow-equivalence.

1 Introduction

Performance oriented design of modern Computer, Communication, and Manufacturing systems relies on the construction and analysis of models that help in gaining insights in their behaviors and in ensuring the development of effective and efficient products [18,16]

Modelling formalisms exist for the representation of these Discrete Event Dynamic Systems [20] making easier the task of capturing in a precise manner

Address(es) of author(s) should be given their relevant characteristics. In most cases the mathematical representations of these models reduce to Continuous Time Markov Chains (CTMCs) for which numerous theoretical and numerical results are available [21].The complexities of real systems naturally translate in the complexities of their models, thus making their analysis computationally difficult, mostly due to the state space explosion of these representations. Consequently, techniques have been devised to control their growths and to simplify the models with divide-and-conquer approaches [6].

When the models are characterized by the interaction among sub-models, decomposition methods can be used and substantial advantages can arise when the stationary solution is in product form [13, 7, 14].

Simple Product Form Queueing Networks (PFQN) [3] which can be solved by efficient algorithms and that are easy to parameterize [18] are often used for these studies. Real systems with simultaneous resource possession, blocking, synchronization, large service time variability, and finite capacity constraints violates the assumptions underlying PFQNs, but can be conveniently and accurately approximated using the Flow-Equivalent technique [8, 2, 15] for computing performance figures related to their stationary behaviors.

When models of this type are used to guide reconfiguration decisions that must be taken within contexts characterized by high variability of the system workload, the steady state analysis becomes questionable and the transient solutions of the models become mandatory.

Product form is a property of the stationary distribution of PFQNs and holds only in very specific cases in the transient phase [4]. As a consequence, transient analysis of real systems gains little advantages when the system can be modeled with PFQNs. Moreover, little is known about the quality of the transient solution of a model computed when subsystems are replaced by their flow equivalent counterparts. Given the practical relevance of the transient phase (for example in case of frequent reconfiguration) in this paper we investigate the application of flow-equivalence in the context of transient analysis. In particular, given a queuing network, we provide a methodology for deciding when the transient behavior can be conveniently approximated with the results of the transient analysis of a reduced version of the model obtained using steady-state flow-equivalence concepts. We limit ourselves to an approximation because, excluding very special cases, devising an exact approach that reduces the complexity of the transient analysis as it happens for the steady state solution of PFQNs, is practically un-realistic. Our criteria are of two different types. The first concerns the structure of the sub-network (of stations) that we intend to aggregate and to replace with an approximate aggregate server. The second checks the adequacy of this representation by testing the response of the aggregate to load conditions that might reveal its inaccuracy in the transient phase.

The paper is organized as follows. In Section 2 we specify the model that we consider in this work and summarize the methods used to compute the stationary distribution of PFQNs. In Section 3 we describe the concept of *flow equivalent server* and show how to restructure the model to accommodate the aggregate representation. In Section 4 we discuss the difficulties that arise when using flow equivalence in a finite time domain. In Section 5, we discuss the structural criteria that must be satisfied by the sub-network that we intend to aggregate to ensure a good aggregation. In Section 6 we argue that the approximate aggregate server must also provide an adequate (dynamic) response to the load represented by an input flow of customers, similar to that expressed by the original sub-network analyzed in isolation and subject to the same load conditions. In this Section we also discuss the quality of the approximation by applying the method to a set of (relatively) large networks exhibiting different characteristics and draw some general indications concerning both the cases in which the method performs well as well as those where the results are not satisfactory. In Section 7 we discuss the application of the method to a case concerning the design and management of a web-service. Finally, in Section 8 we draw some conclusions and outline some future research directions.

2 Model

We consider a closed network of M queues (service stations), $\mathcal{N} = \{s_1, s_2, \ldots, s_M\}$, with a fixed number N of statistically identical jobs circulating through the network at all times (single class network). Jobs (or customers) get service at the stations and move from one station to another according to pre-defined routing probabilities denoted by $r_{i,j}$ with $i, j \in \{1, 2, ..., M\}$ and globally represented by the routing matrix $\boldsymbol{R}$. Service times have exponential distributions which can depend on the length of the corresponding queues; the service intensity of queue i in the presence of n jobs at the station is denoted by $\mu_i(n)$. The service disciplines of all the queues in the network are assumed to be FIFO.

A state of the model is a vector $\boldsymbol{x} = [x_1, ..., x_M]$ where x_i denotes the number of jobs at station i and $\sum_{i=1}^{M} x_i = N$. Accordingly, the state space is defined as

$$\mathcal{S}(N, M) = \left\{ \boldsymbol{x} = [x_1, ..., x_M] \;\middle|\; x_i \geq 0, i = 1, ..., M;\ \sum_{i=1}^{M} x_i = N \right\} \tag{1}$$

and its cardinality is $|\mathcal{S}(N, M)| = \binom{N+M-1}{M-1}$.

The number of jobs at station i at time t is denoted by $X_i(t)$ and the full system state by $\boldsymbol{X}(t) = [X_1(t), ..., X_M(t)]$. The probability that the system is in state $\boldsymbol{x}$ at time t is denoted by $\pi_{\boldsymbol{x}}(t)$, i.e.,

$$\pi_{\boldsymbol{x}}(t) = Pr\{\boldsymbol{X}(t) = \boldsymbol{x}\} = Pr\{X_1(t) = x_1, ..., X_M(t) = x_M\} \tag{2}$$

The probabilities of the whole state space are conveniently collected in a vector $\boldsymbol{\pi}(t) = [\pi_{\boldsymbol{x}}(t)]_{\boldsymbol{x} \in \mathcal{S}(N,M)}$. As the network of queues forms a CTMC, $\boldsymbol{\pi}(t)$ satisfies the well-known ordinary differential equation

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)\boldsymbol{Q} \tag{3}$$

where $\boldsymbol{Q}$, the (homogeneous) infinitesimal generator, is a square matrix of size $|\mathcal{S}(N,M)| \times |\mathcal{S}(N,M)|$ with the non-null entries defined as

$$q_{\boldsymbol{x},\boldsymbol{y}} = \begin{cases} \mu_i(x_i)\, r_{i,j} & \boldsymbol{y} \neq \boldsymbol{x}, \boldsymbol{y} = \{x_1, \ldots, x_i - 1, \ldots x_j + 1, \ldots, x_M\} \\ -\sum_{\forall \boldsymbol{z} \neq \boldsymbol{x}} q_{\boldsymbol{x},\boldsymbol{z}} & \boldsymbol{y} = \boldsymbol{x} \end{cases} \tag{4}$$

2.1 Steady state analysis

The steady state distribution, denoted by $\boldsymbol{\pi}$, satisfies the following system of linear equations

$$\begin{cases} \boldsymbol{\pi Q} = \boldsymbol{0} \\ \sum_{\forall \boldsymbol{x} \in \mathcal{S}(N,M)} \pi_{\boldsymbol{x}} = 1. \end{cases}$$

It has been proved by Gordon and Newell [12] that the equilibrium distribution of customers in a closed PFQN of this type is given by

$$\pi_{\boldsymbol{x}} = \frac{1}{G} \prod_{i=1}^{M} f_i(x_i) \tag{5}$$

where G is a normalizing constant defined so that we have $\sum_{\forall \boldsymbol{x}} \pi_{\boldsymbol{x}} = 1$, and the function f_i (often called *service function* of station i) is defined as

$$f_i(x) = \begin{cases} 1 & x = 0 \\ \dfrac{V_i}{\mu_i(x)} f_i(x-1) & x \geq 1 \end{cases} \tag{6}$$

being $\boldsymbol{V} = [V_1, ..., V_M]$ a real positive solution of the eigenvector-like equation

$$\boldsymbol{V} = \boldsymbol{V R} \tag{7}$$

The direct computation of the normalization constant G is of exponential complexity, but, due to the pioneering work of Buzen [7], computationally efficient algorithms have been devised [10] which obtain the desired quantity in polynomial time. One such method, called *Mean Value Analysis (MVA)* [19], which derives the average performance indices of all the stations of the network without the explicit computation of the normalization constant, is used to obtain the results presented in the last sections of this paper.

3 Flow-equivalent aggregation in steady state

On the basis of the results recalled in the previous section, we can now define an *equivalent server.* Consider a PFQN $\mathcal{N}$ defined as before and let $\mathcal{K} = \{s_1, s_2, \ldots, s_K\}$, i.e., $\mathcal{K}$ is the set of the first K stations, and let $\mathcal{K}'$ denote the set of the remaining stations.

The goal of the aggregation step is to characterize the behavior of the sub-model $\mathcal{K}$ in order to replace it with an equivalent server that will then interact with the other sub-model $\mathcal{K}'$ without affecting its behavior for what concerns the steady state.[1] This characterization of the equivalent server is performed with an *offline experiment* [8,11,9] which corresponds to solving the sub-model $\mathcal{K}$ in isolation under a fixed load $z = 1, ..., N$. For this purpose let us assume $\mathcal{N}'$ to be a queuing network having the same topology of $\mathcal{N}$, but such that the stations in $\mathcal{K}'$ have null service times (they are often referred to as "short-circuited"). Denoting by $\boldsymbol{\pi}'(z)$ the stationary distribution of the customers in the network $\mathcal{N}'$ when the number of customers is z, i.e.,

$$\begin{aligned}\pi'_{\boldsymbol{x}}(z) &= Pr\{\boldsymbol{X} = \boldsymbol{x} | z \text{ jobs in the network}\} \\ &= Pr\{X_1 = x_1, ..., X_K = x_K, X_{K+1} = 0, ..., X_M = 0 \;\Big|\; \sum_{i=1}^{K} x_i = z\}\end{aligned}$$

we can compute the marginal distribution of the customers in the ith station $(i = 1, ..., K)$

$$P'_i(k, z) \;=\; \sum_{\forall \mathbf{x}' \in \mathcal{S}(z,K): x'_i = k} \pi'_{\boldsymbol{x}'}(z) \tag{8}$$

The stationary throughput of station i of $\mathcal{N}'$ when there are z customers in the network can then be written as

$$\chi_i(z) \;=\; \sum_{k=1}^{z} P'_i(k, z)\, \mu_i(k) \tag{9}$$

and the aggregated throughput of the sub-network $\mathcal{K}$ in the presence of z jobs is

$$\chi_{agg}(z) \;=\; \sum_{i \in \mathcal{K}} \left[\chi_i(z) \sum_{j \in \mathcal{K}'} r_{i,j} \right] \tag{10}$$

Finally, denote with s_{eq} a load-dependent queuing station which behaves in such a way that, when z customers are present at the station, the service intensity equals $\chi_{agg}(z)$.

[1] In the rest of the paper, we will often refer to the stations in $\mathcal{K}$ as the *aggregated stations* meaning that these are the stations whose aggregated behavior will be captured by the flow-equivalent server.

Theorem 1 *If station s_{eq} is put in $\mathcal{N}$ in place of the stations that belong to $\mathcal{K}$ in such a way that*

- *the routing probabilities from a station $s_i \notin \mathcal{K}$ to s_{eq} are equal to the sum of the routing probabilities from s_i to each station in $\mathcal{K}$, i.e.*

$$r_{i,eq} \;=\; \sum_{l \in \mathcal{K}} r_{i,l} \tag{11}$$

- *the routing probabilities from s_{eq} to a station $s_j \notin \mathcal{K}$ are equal to the weighted sum of the routing probabilities from each station in $\mathcal{K}$ to s_j,*

$$r_{eq,j} \;=\; \frac{\sum_{i \in \mathcal{K}} V_i \, r_{i,j}}{\sum_{l \in \mathcal{K}'} \sum_{h \in \mathcal{K}} V_h \, r_{h,l}} \tag{12}$$

then the resulting new network (denoted by $\mathcal{N}_{eq}$) has measures of interest equal to those of $\mathcal{N}$ when time approaches infinite.

A different version of Theorem 1 that does not specify the new routing probabilities appeared many times in the literature [8,11,2,1]. The proof usually focuses on showing that the normalization constant of the original network $\mathcal{N}$ is identical to that of the reduced network $\mathcal{N}_{eq}$. Indeed, the recursive computation developed by Buzen [7] shows that the normalization constant G can be obtained via the repeated application of a *convolution* operator that is *commutative* and thus yields the same final result independently of the order in which the stations of the network are considered. This allows to first account for the stations that are aggregated, and then to show that the normalization constant of the reduced network $\mathcal{N}_{eq}$ in which s_{eq} is listed as first station of the network is identical to that of the original one (network $\mathcal{N}$).

Little is available instead in the literature for what concerns the modifications that need to be introduced in the routing matrix of network $\mathcal{N}$ in order to provide a detailed and precise specification of network $\mathcal{N}_{eq}$ as defined by (11) and (12) included in the statement of Theorem 1. This is due to the fact that these details are not relevant for the computation of the stationary distribution of the whole network; they are instead needed for the transient analysis based on flow equivalent servers that will be discussed starting in the next section of this paper. To verify the validity of (11) and (12), it is sufficient to show that the visit ratios computed for the stations that belong to $\mathcal{K}'$ remain identical when considered both within the original network $\mathcal{N}$ and within the reduced network $\mathcal{N}_{eq}$. Appendix A contains the details of this proof.

4 Use of flow-equivalence in transient analysis

In the previous section we showed that the key point for applying flow-equivalence is the computation of the service rates $\chi_{agg}(z)$, $1 \leq z \leq N$, that are required to define the equivalent station. Since the stationary probabilities of a PFQN depend only on the mean service times of the stations, these service rates (which are constants as time tends to infinity) fully characterize an equivalent server which can replace an arbitrary number of aggregated stations without affecting the stationary measures of the system [8,11,2,1].

Within the context of stationary analysis, flow-equivalence has been extensively used [9,16] to perform sensitivity analysis of queuing networks due to the fact that the evaluation of the constants $\chi_{agg}(x)$ is computationally simple and the equivalent server can be reused also for different parameterizations of the remaining stations of the network [8].

Unfortunately, the application of flow equivalence in transient contexts is more difficult and no general results are known for this purpose since the computation of the service rates of a "transient" equivalent server requires the knowledge of the transient distribution of the whole network. In order to clarify this concept in Section 4.1 we will derive general expressions for aggregation in a transient context. In Section 4.2 we introduce the approximate flow equivalence method that we propose in this paper.

4.1 Exact aggregation in transient analysis

Using the notation introduced in the previous section, let us assume that network $\mathcal{N}$ is split into two sub-networks, one comprising the first $M-1$ stations, that we call $\mathcal{K}$ (stations with indices from 1 up to $M-1$), and the other that is made of the last station only (station with index M). A general state of this network ($\boldsymbol{x}$) can be denoted by means of a pair $\boldsymbol{x} = (\boldsymbol{x}', x_M)$ so that the entire state space of the network (that we have previously denoted with $\mathcal{S}(N,M)$) can now be seen as the union of $N+1$ subsets denoted by $\mathcal{S}'(k,N,M), k = 0, ..., N$, and defined as

$$\mathcal{S}'(k,N,M) = \left\{ \boldsymbol{x} = (x_1, ..., x_M) \,\middle|\, x_i \geq 0; \sum_{i=1}^{M-1} x_i = k,\ x_M = N-k; \right\} \quad (13)$$

Let us also assume that the states $\boldsymbol{x}$ of $\mathcal{N}$ are organized in a lexicographical order, so that first we have the (only) state with $x_M = N$, followed by the states with $x_M = (N-1)$ up to the group of states characterized by $x_M = 0$. According to this organization of the state space, the system of differential equations given in (3) can now be divided into $N+1$ sub-systems whose left-hand-sides are characterized by the derivatives of the transient probabilities of the states of the corresponding groups. Let us denote by

$$\tilde{\pi}_k(t) \;=\; \sum_{\boldsymbol{x}' \in \mathcal{S}'(k,N,M)} \pi_{\boldsymbol{x}'}(t) \quad (14)$$

the probability that there are $k, 0 \leq k \leq N$, clients in $\mathcal{K}$ at time t. Then by proper summations of the equations in (3) we can write a system of ordinary differential equations for the quantities defined in (14) in the form of

$$\frac{d\tilde{\boldsymbol{\pi}}(t)}{dt} = \tilde{\boldsymbol{\pi}}(t)\,\tilde{\boldsymbol{Q}}(t) \quad (15)$$

where $\tilde{\boldsymbol{Q}}(t)$ is an $(N+1)\times(N+1)$ matrix, whose non-null entries are

$$\tilde{q}_{i,j}(t) = \begin{cases} \chi_{agg}(i,t) & j = i-1 \\ \mu_M(N-i)(1-r_{M,M}) & j = i+1 \\ -\sum\limits_{\forall k \neq i} \tilde{q}_{i,k} & i = j. \end{cases} \tag{16}$$

being $\chi_{agg}(i,t)$ the aggregated service rate of the stations in $\mathcal{K}$ at time t if there are i clients in $\mathcal{K}$. The term $\chi_{agg}(i,t)$ can be computed according to the following derivation.

Let $\nu_{\boldsymbol{x}'}(h,t)$ be the conditional probability of finding the sub-network $\mathcal{K}$ in a given state $\boldsymbol{x}' \in \mathcal{S}'(h,N,M)$, given that there are h customers in $\mathcal{K}$:

$$\nu_{\boldsymbol{x}'}(h,t) \;=\; \frac{\pi_{\boldsymbol{x}'}(t)}{\tilde{\pi}_h(t)}$$

Given a specific state $\boldsymbol{x}'$ of the sub-network $\mathcal{K}$, the rate at which customers flow from $\mathcal{K}$ to station M is expressed as

$$Y_{\boldsymbol{x}'} \;=\; \sum_{l=1}^{M-1} \mu_l(x'_l)\, r_{l,M}$$

so that

$$\chi_{agg}(h,t) \;=\; \sum_{\boldsymbol{x}' \in \mathcal{S}'(h,N,M)} Y_{\boldsymbol{x}'}\, \nu_{\boldsymbol{x}'}(h,t) \tag{17}$$

When started from identical initial conditions, i.e.,

$$\tilde{\pi}_k(0) \;=\; \sum_{\boldsymbol{x}' \in \mathcal{S}'(h,N,M)} \pi_{\boldsymbol{x}'}(h,0)$$

the aggregated and the original systems (given in (15) and (3), respectively) lead to the same transient behavior for what concerns qauantities related with the number of clients in $\mathcal{K}$ and with respect to station M.

The derivation of the service rate of the flow equivalent server, that we have proposed for the case of a single sub-network to avoid unneeded complexity, can be easily generalized to the case of any number of sub-networks without changing the essence of the result.

Appendix B contains a simple example which provides an illustration of the exact transient aggregation described above, showing its dependencies on the initial conditions and clarifying the relevance of this discussion.

It is clear from the preceding general discussion and from the example of Appendix B that the exact flow equivalence characterization is (computationally) difficult to obtain. In particular, we have shown that

– in order to capture the transient behavior of the original network, the aggregated CTMC must be time-inhomogeneous,

– the computation of the time-dependent rates of the aggregated CTMC requires the solution of the original model,
– the time-dependent rates depend on the initial state of the original model which precludes the possibility of using the characterization of the aggregated servers obtained on the basis of the initial conditions of a certain study, within a different study with different initial conditions.

These observations emphasize the limited practical applicability of the above results and the problems of using the flow equivalent approach in transient analysis. Still they provide motivations for investigating the possibility of computing approximate solutions using the heuristics that we introduce in the following sections.

4.2 Approximate aggregation for transient analysis

Here, we present the simplest strategy that maintains the advantages of the original technique, namely, that the aggregate server is characterized on the basis of the analysis of the aggregated stations in isolation, and that can be used to approximate the transient probabilities of the original model with reduced computational cost.

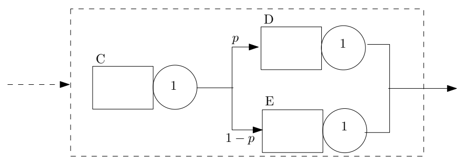
The idea is to impose that $\chi_{agg}(x,t)$ as defined in (17) is equal to the constant value (independent of t) $\chi_{agg}(x)$, $1 \leq x \leq N$, and thus to assume that the flow equivalent servers that appear in the aggregated network are characterized by their steady state behaviors computed with the "offline experiment" described at the beginning of Section 3. Since the rates $\chi_{agg}(x)$ can be evaluated using computationally efficient algorithms [5] such as MVA, their cost is negligible with respect to that of the transient analysis. Moreover, this substitution guarantees that the approximate transient analysis tends to the correct one as time progresses.

On the other hand, this strategy corresponds to assuming that the rate of service of the equivalent server is not affected by the transient probabilities of the remaining stations of the system and by the position of the customers within the aggregated stations. Indeed, this is a very strong assumption that, in general, does not hold. For this reason, in the following sections, we provide conditions under which this assumption is reasonable.

We briefly mention here that an approach based on information gained from the transient analysis of the "short-circuited" servers is also possible to investigate. This would lead however to a situation in which the behavior of the whole network is more difficult to define because subsequent periods with different number of customers in the aggregate would need to be handled. Hence the overall model would become non-Markovian.

5 Structural criteria

The characterization of the aggregate server that substitutes a set of servers $\mathcal{K}$ depends only on the parameters of the sub-net represented by $\mathcal{K}$. In steady



Example 1

Fig. 1 A block of three $M/M/1$ stations

state, for the class of models described in Section 2, this characterization leads to exact results, i.e., the original and the aggregated models show the same behavior. According to the results of Section 4.1 this characterization does not provide instead exact results in the transient phase except for very specific situations. We believe that a precise characterization of all the models for which the approximation is satisfactory is not possible. Consequently, we aim to identify only a subset of such models for which the approximation is foreseen to be good. In this section we propose structural criteria that, when met, indicate the possibility for the transient analysis of the aggregated network to provide accurate results too. These criteria relay on the topology (interconnection structure among the servers) of the network and do not depend on the actual (non zero) values of the transition probabilities from station to station and on the service rates of these stations.

Our first structural condition refers to the way the servers in $\mathcal{K}$ are connected to the rest of the model. We use an example to illustrate the criteria that we propose.

Consider the block (subnetwork) comprising the three stations[2] depicted in Figure 1. As shown in Figure 2, this block can be connected to the rest of the network in different ways. Moreover, note that we can define an aggregated station (that is equivalent in steady state and) which can take the place of stations C, D and E in every possible network containing the block, independently of the parameters of the other stations composing the system.

Let $\mu_A = \mu_B = 1.5$, $\mu_C = 0.1$, $\mu_D = 10$, $\mu_E = 5$, $p = q = 0.5$ and assume that initially there are 20 customers at queue A and 20 at queue B. In Figure 3 we report the mean number of customers at station A for both models as a function of time, denoting by OR the original model and by AG the aggregated one. One can observe that for case (a) the approximation is rather poor while it is accurate for case (b). The reason of the poor approximation is that in model (a) the time spent by a customer inside the aggregated stations is strongly dependent on the station from which it joins the subnetwork and it has high variance. On the contrary, in model (b) every customer enters the aggregate via the same station C, independently of the originating one.

Accordingly, as first structural condition we require is that the route a job follows in the aggregate is independent of where it comes from and toward

[2] Throughout this paper, the number reported in the service part of a queuing station refers to the number of servers working in parallel within that station.

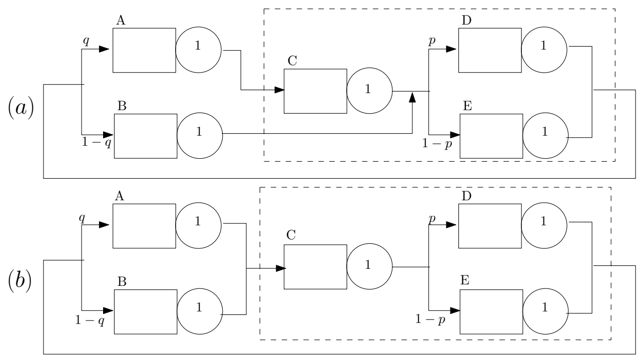


Fig. 2 Two models having the same block of stations

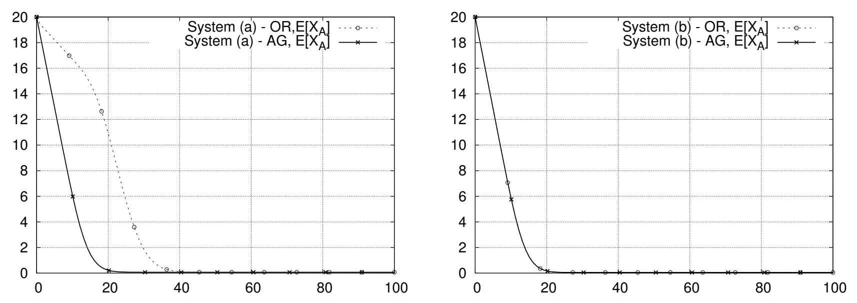


Fig. 3 Mean queue lengths in two models having the same block of stations

which station it leaves the aggregate. Formally, denoting by $\mathcal{A}$ ($\mathcal{D}$) the set of stations from (to) which clients can arrive directly to (from) $\mathcal{K}$, the above condition holds if and only if

$$r_{k,i} = r_{k,j} \ \forall k \in \mathcal{A}, \forall i, j \in \mathcal{K} \ \text{ and } \ r_{j,k} = r_{j,k} \ \forall i, j \in \mathcal{K}, \forall k \in \mathcal{D}. \tag{18}$$

The second structural condition that we require refers to the initial position of the customers of the network. In particular, we require that there are no costumers in the aggregate server initially. This is necessary because the way customers leave a set of stations in the transient setting depends on where they are positioned originally. This aspect is clearly not considered by the characterization we propose for the aggregate server. Consider again the subnet given in Figure 1. For the aggregate model it does not make any difference if the clients are put originally at stations C or D originally while the transient behavior can be substantially different for the two cases.

It is evident that the above two conditions are not sufficient to obtain a good transient approximation because they do not consider the quantitative aspects of the model. Conditions involving quantitative aspects will be provided in the next section where we propose a quantitative check to decide between acceptance and rejection of a candidate aggregate.

6 Behavioral criteria

To identify possible aggregations in a reliable manner, we must also take into account the actual (numerical) parameters of the network to decide if aggregating a set of stations $\mathcal{K}$ provides a reasonable approximation of the transient behavior of the whole model. In order to propose a feasible way of testing the validity of a possible aggregation, we must devise a methodology which considers the aggregate stations of the sub-network $\mathcal{K}$ in isolation, still retaining some overall information on the rest of the network which provides a context where to analyze its transient behavior. Obviously, the context must be simple as we cannot afford to devise a procedure which takes into account a detailed description of the rest of the network at a cost comparable to that of the analysis of the original model.

A first consideration is that the stations over which the equivalent server is defined must be such that, when perturbed by arrivals from the rest of the network, they reach their "quasi steady state" before the number of customers in the aggregate server changes. Identifying in a precise manner the rates at which customers arrive at the equivalent server starting from a given initial state has a cost which is comparable with that of the transient analysis of the whole model. We must thus accept approximations and heuristics.

A first criteria that would be in line with these last considerations is that the service rates of the stations in $\mathcal{K}$ must be faster than those not belonging to $\mathcal{K}$. A possible heuristic to establish whether or not an equivalent server defined over $\mathcal{K}$ is trustworthy, could be that of comparing the products $V_i \times max_x \mu_i(x), \forall i \in \mathcal{K}$, with those referring to the stations that directly feed the equivalent server and thus represent the *interface* between the aggregate and the rest of the network; the more the firsts are fast, the more we expect the approximation to provide accurate results. However, especially in presence of load-dependent stations in the interface, this strategy is excessively restrictive ruling out aggregations that lead instead to acceptable approximation.

We propose hence a quantitative criteria to compare the speed of the equivalent server and the speed of the rest of the network in each possible state of the equivalent server considering the initial state $\mathbf{y}$ as well. The comparison is carried out on a network composed of only two stations, namely, the equivalent station s_{eq} and another server (called s_{max}) that represents all the remaining stations of the original network. The service rate of s_{max} denoted by $\chi_{max}(x, \mathbf{y})$ approximates by a heuristic approach (described later) the maximal arrival rate toward the equivalent station given that x customers are present in s_{max} and the initial state is $\mathbf{y}$. Then we analyze the ratio

$$\rho(x) = \frac{\chi_{agg}(x)}{\chi_{max}(N - x, \mathbf{y})}, \quad 1 \le x \le N - 1 \tag{19}$$

where $\chi_{agg}(x)$ is the load-dependent service rate of the equivalent server.

For what concerns $\rho(x)$ we have the following cases.

- If ρ is always greater than one, i.e., the equivalent station serves with a rate that is constantly greater than that of the rest of the network. In this case, the number of clients at the equivalent station is always small with high probability. This is the situation in which we expect a very accurate approximation due to the fact that if there is only one customer within the aggregate station, there is no congestion and the average time that it spends there is very similar both in transient and steady state periods.

– There is a threshold k greater than 1 and smaller than $N-1$ such that ρ is smaller than one for $1 \leq x < k$ and greater or equal to one for $k \leq x \leq (N-1)$. In this case, since we imposed that the equivalent station is empty at the beginning of the transient period, we have that the number of customers at the equivalent server increases monotonically up to a certain level l and then starts to stabilize[3]. The more l is small, the faster the equivalent station stabilizes; roughly speaking, this means that if l is close to one, the approximation is accurate; whereas if l is close to N the error will be significant.
– ρ constantly below one. This is the worst case where the use of the approximation leads to inaccurate results.

Now we turn our attention to the definition of the function χ_{max}, which is needed for the evaluation of ρ. Note that the exact calculation of χ_{max} would require the transient analysis of the model and thus our parametrization is based on computationally negligible heuristics. In our experiments we found that these heuristics provide reliable information of the model.

In order to proceed we need the following notation. Let ω_i denote the set of all paths from station i to the aggregate station. Given a path $\sigma \in \omega_i$, let $s(\sigma)$ denote the set of stations along the path and $p(\sigma)$ the probability that a client follows σ. We assume that the sub-model that connects the initially loaded stations to the aggregate server is acyclic; the generalization to the cyclic case is straightforward.

Then χ_{max} is calculated as

$$\chi_{max}(x, \mathbf{y}) = \max_{\mathbf{y}' \leq \mathbf{y}, \sum_i y'_i = x} \sum_{\forall i \ s.t. \ y'_i > 0} \sum_{\forall \sigma \in \omega_i} p(\sigma) \min_{\forall j \in s(\sigma)} \max_{z=1,..,x} \mu_j(z) \tag{20}$$

where from left to right

– the first max operator chooses an initial condition that is congruent with $\mathbf{y}$ and provides maximal load toward the aggregate;
– the first $\sum$ considers all initially loaded stations;
– the second $\sum$ considers all paths toward the aggregate;
– the combination of the min and max operators results in a heuristic service rate that overestimates the speed with which the clients may arrive at the aggregate.

The formula in (20) treats the clients in some sense in average (the routes are weighted according to their probabilities) and in some sense in a pessimistic

[3] As a further condition, we assume also that χ_{max} and χ_{eq} are monotonic increasing functions in such a way that ρ is guaranteed to overtake 1 at most once. Note that this assumption is violated only if the load-dependent service rates are "ill-parametrised"

manner (maximal service rates are chosen at all involved stations). In general the computation given in (20) can be excessively complicated due to the combinatorial complexity introduced of the first max operator and due to the numerous path but in most practical cases (as for those presented later) it boils down to simple calculations.

Note that the test is applicable also when the transient analysis of interest starts from a set of states. In this case, the function χ_{max} is computed for every possible initial state and weighted according to the initial probabilities.

In the following we illustrate the use of the test.

Example 2 Consider the network depicted in Figure 4 and composed of four single server stations.

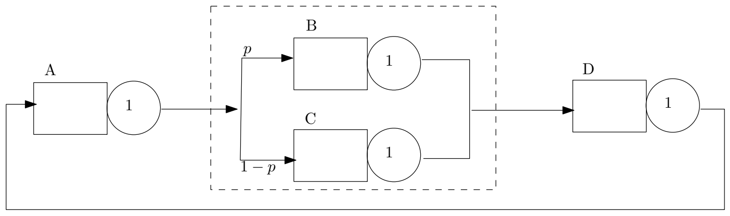


Fig. 4 Example 2 - Queuing network composed of four M/M/1 stations

Assume that 40 customers are present inside the system and that the parameters of the network are $\mu_B = 0.1$, $\mu_C = 1$, $\mu_D = 1$, and $p = 0.5$. Let us consider the case in which all the customers are in station A at the beginning and we are interested in assessing the quality of the aggregation of stations B and C for different values of μ_A. In this case, since station A is single server then χ_{max}, as it has been defined in (20), is constant, independent of the value of x. In Figure 5 we report the results of this test. It is possible to notice that the only "safe" approximations are those with μ_A equal to 0.1 and 0.15, whereas the others can lead to significant inaccuracies.

This is confirmed by Figure 6 which provides a comparison between the original and the aggregated model by showing the mean and the variance of the number of customers present at station D for the cases with μ_A equal to 1, 0.5 and 0.1[4]. Indeed, the time dependent behaviors generated by the aggregated model when it is subject to an initial arrival rate equal to 1 and 0.5 are completely different from those observed in the original model. Specifically, for the case $\mu_A = 1$ the aggregated model reaches steady state after 5 time units whereas the original network arrives near the steady state only at time 75 after a spike that the model with the aggregate station is unable to reproduce. A worse situation occurs with $\mu_A = 0.5$, where the aggregate stabilizes around 50 time units, whereas the original model requires more than 450 time units

[4] For sake of synthesis, we do not report the results obtained for many other values of μ_A.

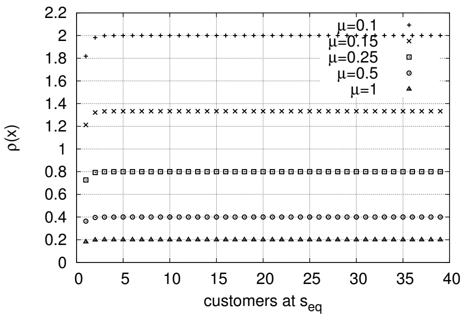


Fig. 5 Check of the aggregation of station B and station A by varying μ_A. (Example 2, $\mu_A = 1$).

to reach the equilibrium. On the contrary, we can observe that when $\mu_A = 0.1$ the approximation provided by the equivalent station is accurate.

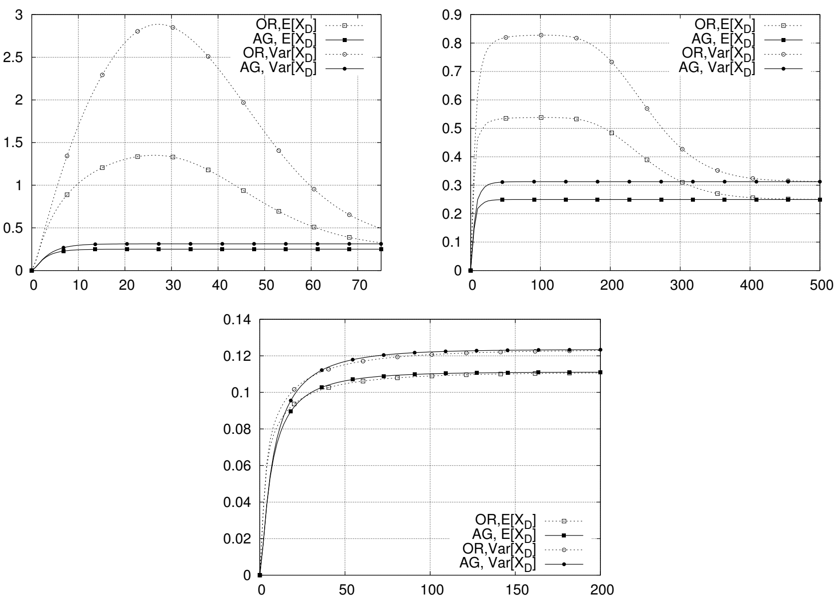


Fig. 6 Comparison of the original and the aggregated models: the mean and variance of the number of customers at station D as function of the time (Example 2, for the cases with μ_A equal to 1 (top left), 0.5 (top right), 0.1 (bottom)).

Another example of use of the test is provided in the following where, in order to put under stress our methodology, we considered a larger sub-network characterized by a more intricate interconnecting structure.

As for the previous examples, since the purpose is to show the potentials of the method, we perform the experiments on a relatively small network in which we aggregate all the stations except for the two that are used as *observation* points to evaluate the accuracy of the approximation.

Example 3 The model we consider is depicted in Figure 7 and corresponds to a network composed of five $M/M/1$ stations and an infinite server. The customers go through a cycle that starts at station A and continues with a choice between two parallel paths: the first composed of stations B and D, and the second of stations C and E. Upon their exit from these parallel paths, customers may either move forward to station F or return to the choice-point between the two parallel paths to follow one of them once more. Finally, after a service at station F customers return to station A.

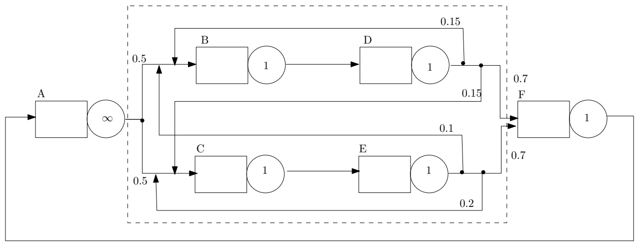


Fig. 7 Example 3 - Queuing System

We assume to be interested in measures computed for stations A and F. In this way, stations B, C, D, and E can become part of aggregate servers. Because of the structural conditions described in Section 5, the possible aggregate servers are

1. stations B, C, D and E (from now-on we will denote this aggregate server with $s_{B,C,D,E}$);
2. stations B and D ($s_{B,D}$);
3. stations C and E ($s_{C,E}$).

For our tests, we chose the first aggregation because it minimizes the number of stations of the network.

We use the following parameters: $\mu_B = 5$, $\mu_C = 0.5$, $\mu_D = 2$, $\mu_E = 0.2$ and $\mu_F = 1$, $r_{A,B} = r_{A,C} = 0.5$, $r_{D,F} = r_{E,F} = 0.7$, $r_{D,B} = r_{D,C} = 0.15$, $r_{E,B} = 0.1$ and $r_{E,C} = 0.2$. The parameters highlight that the two parallel paths of the network have very different passage times. Indeed, the path composed of stations B and D is much faster than that composed of C and E.

We test different values of the service rate of station A by assuming that 40 customers are present at station A whereas the other stations are initially empty. Since station A is an infinite server then $\chi_{max}(x) = x \cdot \mu_A$. Figure 8 depicts the results (in log scale for sake of readability). Observing the plot it is possible to notice that: when station A serves customers with rate 0.001 the function ρ is constantly larger than one; when the rate is 0.005, ρ is always greater than one except for the case in which the equivalent station contains a customer; when μ_A is equal to 0.01, ρ starts to be greater than one when 15 customers are present in the the equivalent station; when μ_A is 0.05, it is greater than one only when 34 customers are in the equivalent station. Always in Figure 8 it is possible to notice how the four trajectories diverge. In fact, the smaller μ_A is and the more $\rho(x)$ tends to infinity for large values of x.

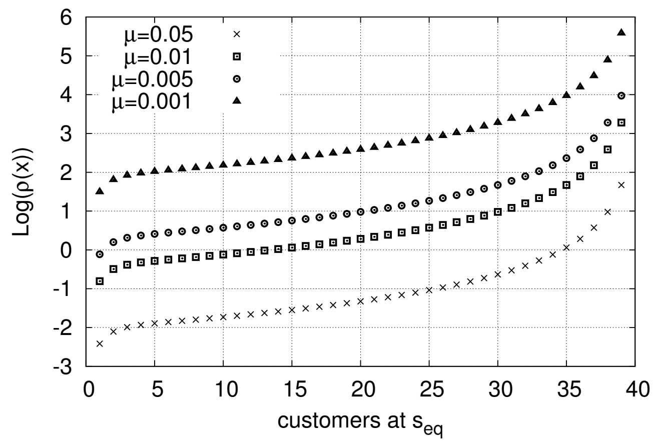


Fig. 8 Log of function ρ by using the aggregate station $s_{B,C,D,E}$ when all the customers are at station A at the beginning with μ_A equal to 0.001, 0.005, 0.01 and 0.005. (Example 3)

In this situation we expect that in the first two cases the aggregation will be able to provide accurate results; in the third case it will provide an approximation with a significant error; whereas in the last case the aggregation will not be able to represent the dynamics of the original network.

Figure 9, that depicts the mean and the variance of the number of customers present at station F as function of the time, confirms our prevision. In fact, it is possible to observe that for the cases with μ_A equal to 0.001 and 0.005, the use of the equivalent server provides time dependent behaviors that are almost indistinguishable from those generated by the original network. On the contrary, the other two cases present a substantial error until the system is close to be in steady state. As pointed out by the test in Figure 8, the approximation works better when μ_A is equal to 0.01 than when it is 0.05. In fact, in the third scenario, the use of the equivalent server provides a maximum relative error around 12 percent on the mean and about 20 per cent on the variance whereas in the fourth case they are around 78 and 88 percent respectively. The large difference is consequence of the fact that although both the mean and the variance are underestimated, in the third scenario the original and the approximated trajectories flatten out at the same time. This does not happen in the fourth scenario where the flow equivalent approximation misses completely the peaks present in both the mean and the variance around 25 time units.

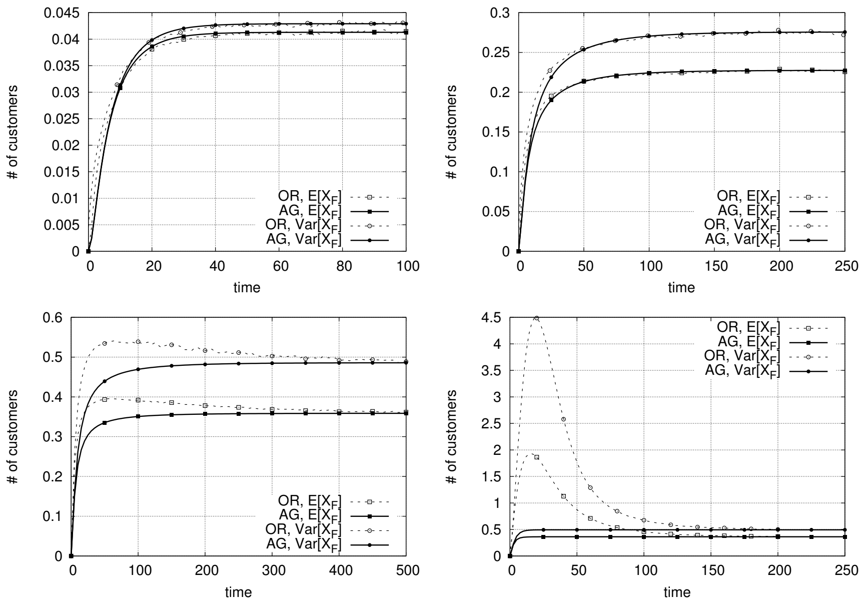


Fig. 9 Comparison of the original and the aggregated models: the mean and variance of the number of customers at station F as function of the time (Example 3, for the cases with $\mu_A = 0.001$ (top-left), $\mu_A = 0.005$ (top-right), $\mu_A = 0.01$ (bottom-left) and $\mu_A = 0.05$ (bottom-right)

As a last example, we propose a variation of the model depicted in Figure 7. The aim of the test is to show the impact of the initial state on the quality of the approximation.

Example 4 In order to illustrate the robustness of the test, we consider the net depicted in Figure 11 which is identical to that considered in Example 3, for except station A that has been replaced by three stations: namely, A_1, A_2, and A_3.

Stations A_1 and A_2 are single-server stations having service rate 0.1 whereas A_3 is an infinite server with service rate 0.05; they are connected with the rest of the net in such a way that when a customer finishes its service at F it arrives to A_1 or to A_2 with equal probability 0.5. After a service at A_1 a customer arrives to A_3; finally, when a customer departs from stations A_2 or A_3 it arrives to B or C with probability 0.5. The remaining parameters of the network are identical to those used in the previous experiment.

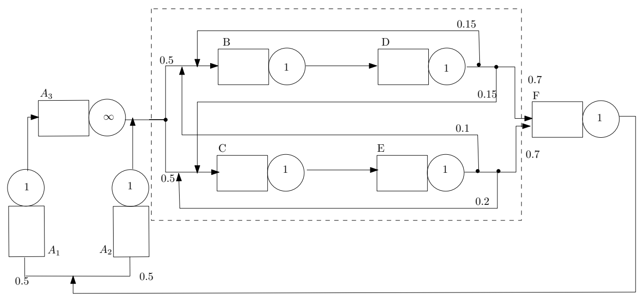


Fig. 10 Example 4 - Queuing System

Also in this new version of the network the set of stations that we aim to aggregate consists of stations B, C, D and E. In particular, we assume a scenario in which we want to perform transient analyses for three different initial states: $A_1 = A_2 = 20$, $A_2 = A_3 = 20$ an $A_1 = A_3 = 20$.

The result, reported in Figure 11, shows that: i) the initial state $A_2 = A_3 = 20$ provides the same $\rho(x)$ of the case $A_1 = A_3 = 20$; ii) since $\rho(x)$ starts to be greater than one only at 35, the initial states in which customers are present at station A_3 are candidates to be unfavourable situations to apply our approximation; iii) the initial state $A_1 = A_2 = 20$ is an optimal situation to use the transient flow-equivalence approximation because the corresponding ρ is smaller than one only for $x = 1$.

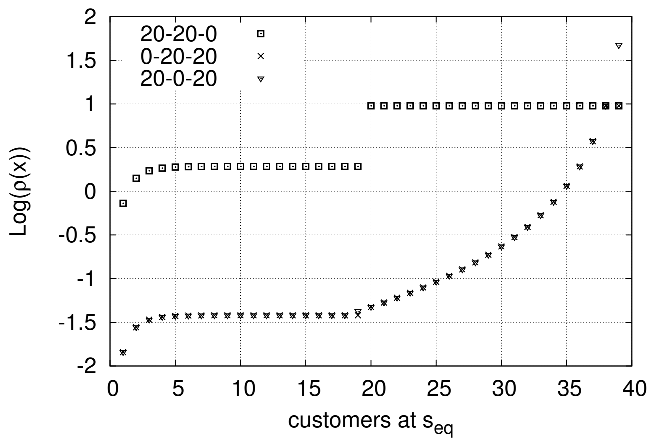


Fig. 11 Log of function ρ by using the aggregate station $s_{B,C,D,E}$ for three different initial states: $A_2 = A_3 = 20$, $A_1 = A_2 = 20$ and $A_1 = A_3 = 20$.

As for the previous cases, we performed both the exact and the approximated transient analysis and used the station placed right after the aggregation as observation point. The results are reported in Figure 12 where the average and the variance of the number of customers at station F is depicted. Observing the results, it is possible to notice that our conjectures are confirmed; in fact transient flow-equivalence fails to reproduce the peaks that characterize the two situations where customers are present in station A_3 whereas it provides an accurate approximation when the initial state is $A_1 = A_2 = 20$.

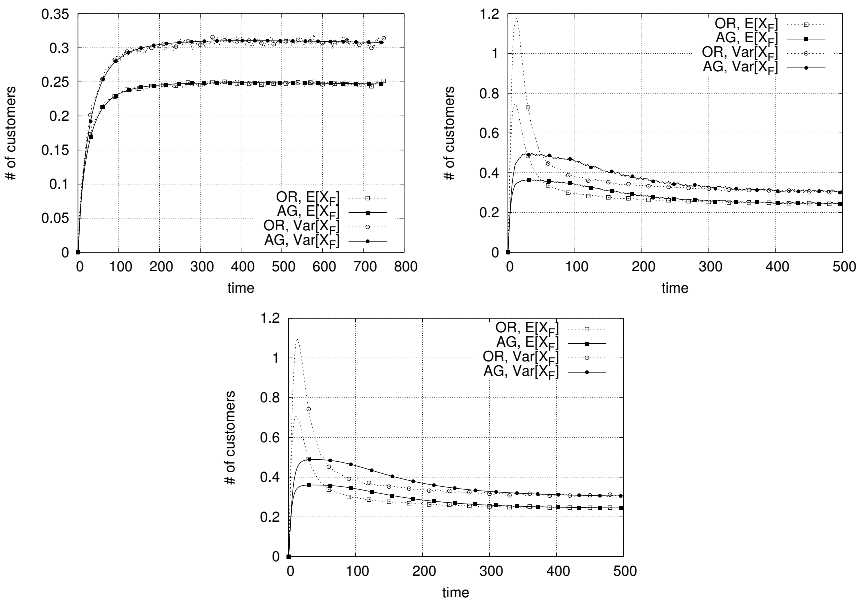


Fig. 12 Comparison of the original and the aggregated models: the mean and variance of the number of customers at station F as function of the time (Example 4, for the cases with initial state $A_1 = A_2 = 20$ (top left), $A_2 = A3 = 20$ (top right) and $A_1 = A3 = 20$ (bottom).

7 Numerical illustration - Analysis of a web-service network

In this section, we apply the methodology for the analysis of a realistic model of a web-service considered under several operational conditions. The model is the synthesis of several models presented in [17] where flow-equivalence has been massively used to perform stationary analyses.

Following the indications in [17] we construct a PFQN model consisting of three main parts: an infinite-server station that represents thinking clients; two single-server stations modeling the I/O channels of the router used by the web-service to communicate with the clients; a set of load-dependent stations corresponding to the web-servers that process clients requests. A graphical representation of the structure of our model is depicted in Figure 13 where the web-servers are not provided in detail. The detailed representation of the web-servers are depicted in Figure 14.

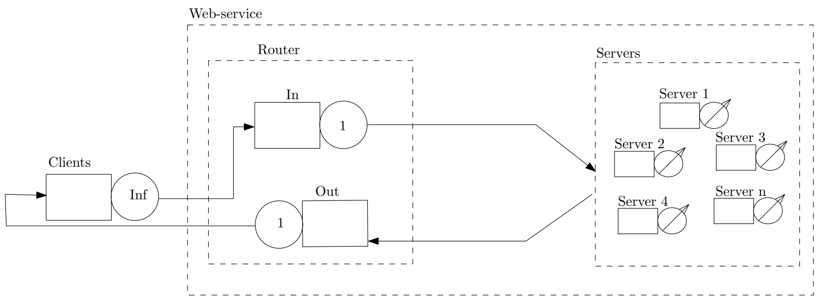


Fig. 13 Case 2 -Clients connected to a web-service

For sake of simplicity, the modeling of each web-server considers only the interaction among the CPU and two storage disks as well as the communication between the CPU and the router. Requests coming from the clients are managed by the router and delivered to the $CPUs$ of the different servers by the In station according to pre-defined routing probabilities; answers produced by the servers return to the interested clients via the Out station of the router.

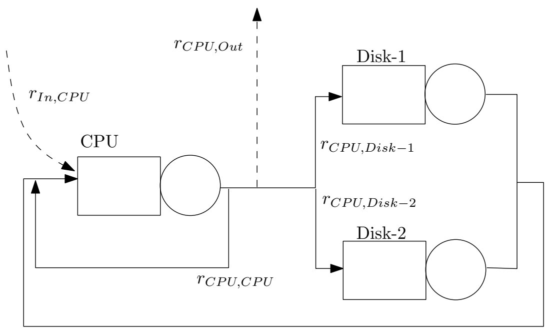


Fig. 14 Case 2 - Server structure

This basic model is used to study the behavior of the web-service with respect to three different types of questions:

– the effect of the change of the number of servers with respect to the transient dynamics of the web-service;

– the effect that different strategies adopted for the variation of the number of servers allocated to the web-service may have shortly after the variations;
– the effect that different server architectures may have on the transient dynamics of the web-service.

All these questions are studied with the aim of illustrating possible situations in which the use of equivalent stations provide good results with additional benefit of reducing consistently the computational cost of the analyses.

7.1 Analysis of the time that the system requires to stabilize

In this first context, we take into consideration three systems that differ only in the number of web-servers connected beyond the router, i.e. one, two, and three web servers, respectively. Requests arriving to systems with more than one web-server are directed with equal probability to the different web-servers.

Web-servers are networks of single server stations having rates $\mu_{CPU} = 300$, $\mu_{Disk-1} = \mu_{Disk-2} = 1$ and routing probabilities $r_{CPU,CPU} = 0.7$, $r_{CPU,Disk-1} = r_{CPU,Disk-2} = r_{CPU,Out} = 0.1$. The thinking rate of each client is 0.03 and the I/O channels of the router receive/send requests with rate equal to 1.1.

For all the three alternatives, we aim to estimate the time the system needs to reach a stationary condition, given that the router and the web-servers are idle at time 0 and that the number of clients is equal to 40. The detailed analysis of this case study is quite costly and we show that the results that we obtain by applying the flow-equivalence approach are robust and relatively inexpensive. In order to maximize the reduction of the network we collapsed all the stations beyond the router in a single aggregate.

Figure 15 shows function ρ for all the three cases, points-out how the more web-servers compose the aggregate and the more the aggregation is foreseen to be accurate. In particular, we can observe that the function referring to the case in which only one web-server is connected to the network goes beyond one at 28 whereas those with two and three serves are already greater than one at 10 and 6 respectively. Although we expect a poorer approximation in the first case, by considering that all the function diverge quickly, we can assume that all the aggregations will provide a satisfactory approximation of the dynamics of the original systems.

Figures 16 and 17 confirm our expectations by showing respectively the mean and the variance of the number of thinking clients and the time evolution of the utilizations of stations In and Out. In fact, the use of equivalent stations produces an accurate approximation of the transient behavior of the original model for all the cases. In particular, we can see that the accuracy of the approximation improves by using more equivalent servers. Although this phenomenon is heavily dependent on the parametrization of the model, we conjecture that it is a consequence of the fact that for small queue levels, the equivalent stations serve request with rates that are very close to those that they have in steady state.

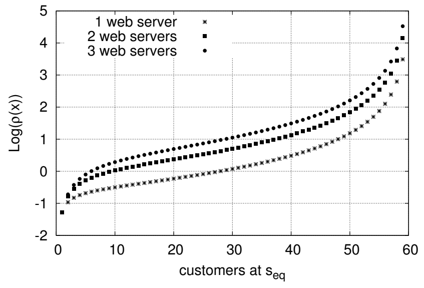


Fig. 15 Log of function ρ for the cases in which the webservice is composed of one, two an three web-serves by considering the case in which all the customers are in the infinite server at the beginning.

Observing these results, a point of interest is that the web-service with a single web-server requires a time to stabilize which is about twice longer than that of the other two cases.

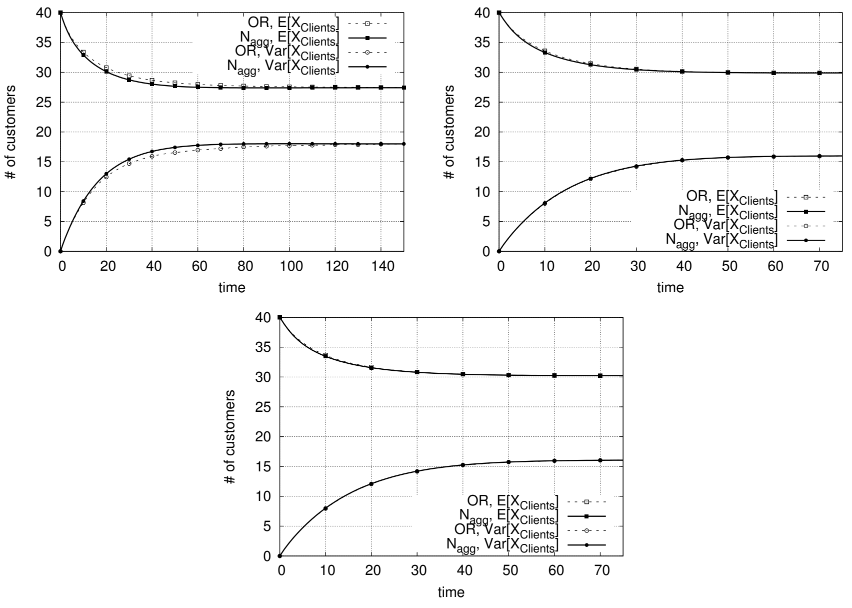


Fig. 16 Mean and variance of the number of thinking clients as function of time - single web-server (top left), two web-servers (top right), three web-servers (bottom).

The longer transient period of the system with only one web-server may be explained by observing that in this first case the bottleneck station is the

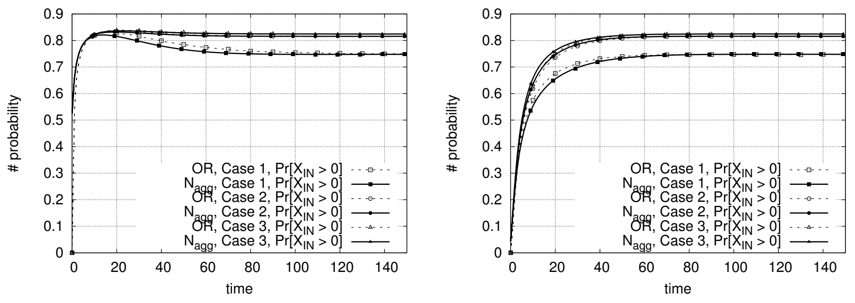


Fig. 17 Utilization of stations *In* and *Out* as function of the time.

web-server itself, whereas for the second and the third system's configurations the bottlenecks are always stations *In* and *Out*.

7.2 Upgrading the system - Analysis of different strategies

The second aspect of the web-service that we study with our flow-equivalence aggregation methodology concerns the possibility of addressing different design and planning strategies concerning the choice of changing the number of connected servers at run-time.

The first question that we consider concerns the possibility of connecting two additional servers to a web-service which originally works with a single web-server. We further assume that: i) the addition of the new servers is performed on a pre-existent system that is stable; ii) the system is constrained in such a way that the router utilization cannot exceed the 85 percent limit (as an example, we can imagine that part of the bandwidth is dedicated to protocols, such as SNMP, for the remote monitoring to the entire system).

A common way of addressing this problem is to consider the PFQN of the system with three web-servers and then compute the stationary utilization of the stations modeling the router in order to verify that they do not exceed the pre-fixed threshold. For this particular case, by considering stationary measures only, such property is satisfied; in fact, if the web-service has three web-servers then the stationary utilization of each channel is around 0.824. In spite of this, stationary analysis do not provide any information about the behaviour of the system during the interval that goes from the moment in which the system is perturbed because of the upgrading (connection of the two additional components) up to the time when stability is reached again.

Let us consider two possible strategies to upgrade the system: the first considers the scenario in which we connect the second web-server to the system, we wait until the system has reached again a stable state and then we connect the last web-server; the second consists of a brute force approach in which the two new web-servers are connected to the system simultaneously.

In order to decide which strategy is preferable, we perform three experiments: the first that considers the upgrade of the system from one web-server to two; the second considering the case in which the system with two servers is enhanced with a third web-server; the last that considers the simultaneous connection of two new web-servers to the system. Each analysis has been performed by using the stationary distribution of the system before the upgrading as initial condition. Denote with π^i_x the stationary probability to observe the system with i web-servers in state x, with $\pi^j_{x'}(0)$ the initial condition of the system having j web-servers in state x', and assume that all the entries of the vector x are ordered in such a way that they correspond to those in x' if $i < j$. We perform the analyses by starting from initial conditions that have $\pi^j_{x'}(0) = \pi^i_x$ if $x'_k = x_k$ for $1 \leq i \leq |x|$ and $x'_k = 0$ for $|x| + 1 \leq k \leq |x'|$. Indeed, this time we cannot collapse all the network beyond the router as a single aggregate since we must discriminate between the web-service that is already working and those that have been just connected to the system. Thus, we consider each web-server as a different equivalent station.

Moreover, since the initial state of these transient analyses are different from those considered in the study reported in the previous sub-section, we need to check again the reliability of the equivalent station representing the web-servers. Note that in this case it is more complex to define function χ_{max} because we must account for the fact that the original position of the clients/requests is not deterministic, but is specified with a probability distribution. As already mentioned, this can be done by averaging function χ_{max} according to the starting distribution. In this particular case, this operation is particularly simple since the average rates with which customers move inside the network at the beginning of the analysis corresponds to the stationary throughput of the network when a single web-server is connected (two for the upgrading from two to three) and $1, \ldots, N$ customers are present in the system. Figure 18 depicts the result of the three tests from which can observe that the three trajectories are indistinguishable. We can notice that they to be greater than zero at 20; thus, by using a sort of rule of thumb, we expect an approximation that is of the same quality of the first case of the experiment presented in the previous section.

We compute the time evolution of the utilization of stations In and Out from the moment immediately after an upgrading of the system to that in which the system can be considered stable again. The results are reported in Figures 19 and 20.

Specifically, Figure 19 shows how the utilization of station In increases monotonically over the whole transient period and then stabilizes around the stationary value for all the three upgrading policies that we consider. As a consequence, station In does not overtake the 85 percent threshold that we imposed as constraint.

On the contrary, the utilization of station Out, depicted in Figure 20, is characterized by peaks in all the three cases. These peaks occur because the new web-servers are idle at the moment of their connection to the system; thus, immediately after the connection of a new web-server, requests have

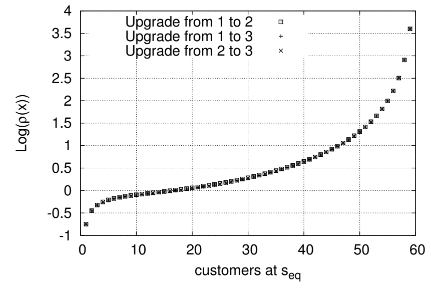


Fig. 18 Log of function ρ for the three upgrading cases.

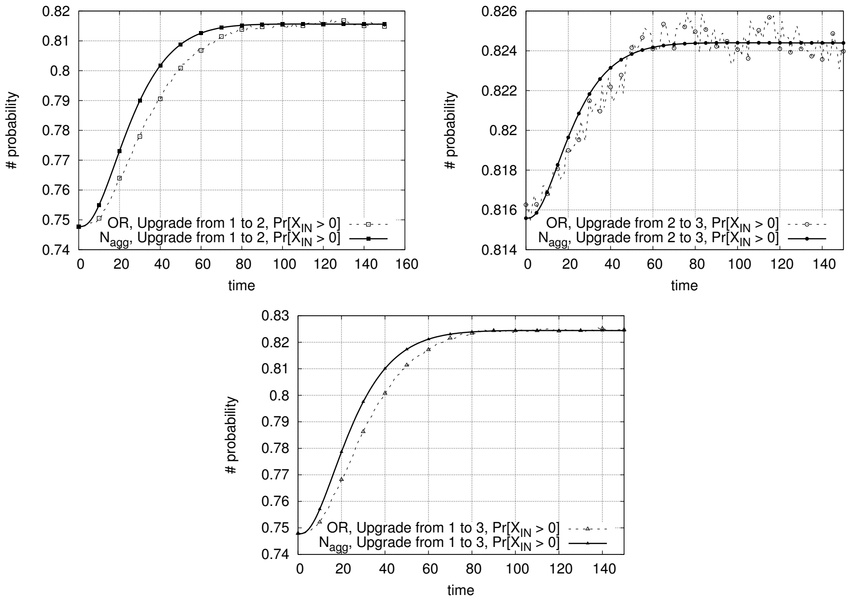


Fig. 19 Utilization of station *In* as function of the time during different upgrading of the system.

a not negligible probability to be served promptly, without queuing. It thus follows that at the beginning of each transient period, the response times of the web-servers are lower than their steady-state values. As a consequence, station *Out* becomes congested in these initial periods. From Figure 20, we can notice that the simultaneous connection of two new web-servers violates the constraint (on the router utilization) that we imposed at the beginning. Our approach is able to represent this facet as well, but it slightly overestimates the height of the peaks and underestimates their time extensions (note the scale of the figures). This inaccuracy is present also in the other two cases, leading to misleading results concerning the "from one to two" web server upgrading of the system. In fact, our approach predicts a violation of the threshold in the time interval $(10, 30)$.

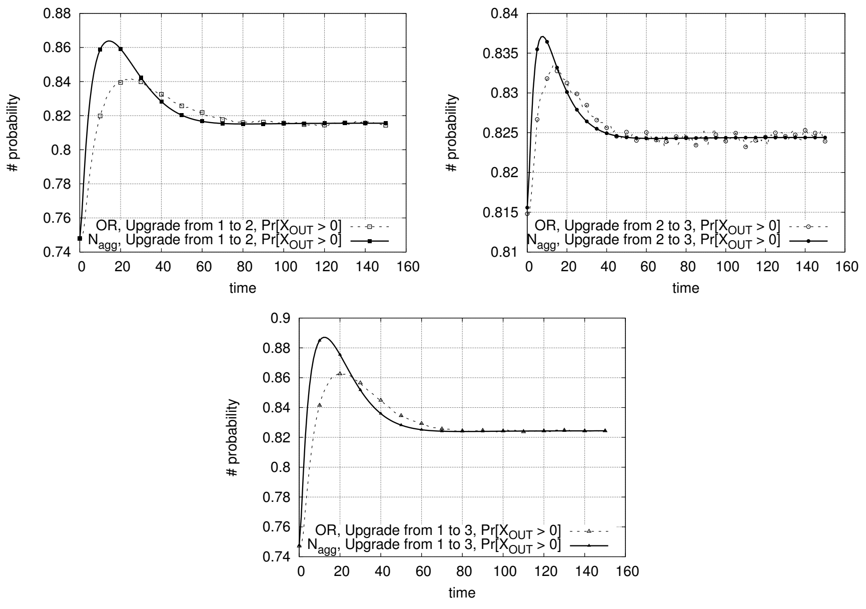


Fig. 20 Utilization of station *Out* as function of the time during different upgrading of the system.

Despite this, it is important to point out that the error induced by the use of the equivalent stations is limited, and that an important result is the capability of the method to highlight (with reasonable costs) a possible problem that would be completely neglected by any analysis based exclusively on steady-state behaviors. Indeed, we provide in Figure 21 the relative error between the exact and the approximated utilization of station *Out*. By observing the plot it is possible to notice that the relative error does not exceed the 0.7 percent in the worst case.

Summarizing the results of this analysis, we can observe that the trajectories generated using our approach show clearly that:

- the first strategy minimizes the risk of malfunctions;
- the second strategy minimizes the time required by the system to return to stability conditions.

7.3 Example of use of the approximation on a non-PFQN

The third aspect of the web-service that we study with our methodology refers to a case in which the web-servers connected to the router are not identical, but

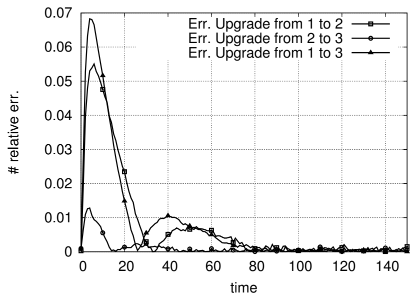


Fig. 21 Relative error on the utilization of station *Out* between exact and approximate solution as function of the time.

show some specific differences in the internal parallelism that they apply. The type of analysis that we perform during this last set of experiments is quite similar to that of the study conducted in Section 7.1, but this time we consider a web-service having one web-server as in the previous case and another one that mantains the structure depicted in Figure 14, but is composed of multi-server stations; specifically, the CPU has four servers, $Disk-1$ has two and $Disk-2$ has three with rates $\mu_{CPU} = 150$, $\mu_{Disk-1} = 2$ and $\mu_{Disk-2} = 1.5$ respectively. The routing probaiblities are $r_{CPU,CPU} = 0.5$, $r_{CPU,Disk-1} = r_{CPU,Disk-2} = 0.2$ and $r_{CPU,Out} = 0.1$.

As a further assumption, we consider the scenario in which the router has knowledge of the number of requests present in the two web-servers and decides the routing of incoming requests by using a shortest queue policy. This means that a new request is sent to the second web-server after the ending of a service at station In if the sum of the number of requests present at the stations composing the first web-server exceeds the number of those under process at the second web-server and viceversa. As last, we give priority to the second web-server whenever the two web-servers are dealing with an identical number of requests. Note that within this new setting, the queuing network representing the whole model does not have a product form solution and cannot be analyzed using the standard and efficient computational algorithms (e.g. MVA algorithm). In this case the cost of the computation of the original model is not negligible also for what concern stationary measures.

We perform the transient analysis of the whole system till it can be considered in steady state. Figure 22 depicts the average arrival rate to the two web-servers from which it is possible to observe that the router addresses the requests to the second second web-server more frequently. This is not surprising since it has higher priority and is almost twice faster than the first one under saturation condition. However, for smaller utilizations the second web-server is slower than the first due to the fact that it is composed of multi-server stations. Figure 22 points out this phenomenon in the time interval (0,5) when the curve describing the arrival rate to the second server flattens after the first arrival. This phenomenon occurs because the second-server receives almost immediately the first request from the router due to its higher priority. Since it is not able to process the request before the arrival of the next one, the router directs the new request to the first web-server whose queue is still empty.

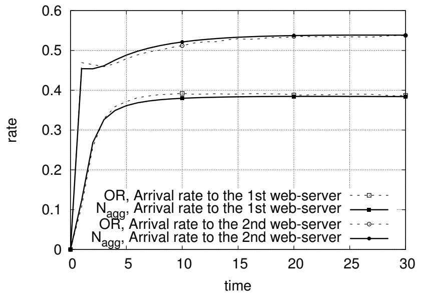


Fig. 22 Average arrival rate to the web-servers.

Another point of interest, shown by Figure 23, is the fact that even if the arrival rates to the web-servers require only 30 time units to stabilize (see Figure 22), the whole system can be considered completely stable only after 70 time units. Both the figures show that the use of equivalent stations provides

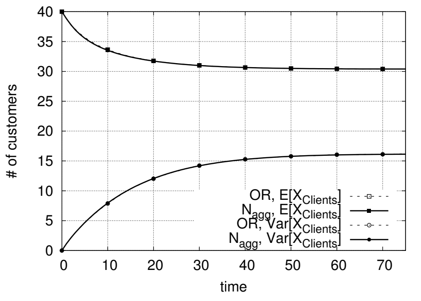


Fig. 23 Mean and variance of the number of thinking clients as function of time.

a quite accurate approximation of the original trajectories. In particular, for what concerns the number of thinking clients the original curves cannot be distinguished from their approximation.

The computation of the transient analyses required in the worst case (web-service having three web-servers) the uniformization of a $CTMC$ with 1221759 states whereas the original process was composed of 4.7×10^{11} states. In regards to this case the time required to carry on the computation was around 20 minutes. The computation of the other cases did not exceed two minutes per case. The original trajectories have been computed on the basis of 500.000

simulation runs. The most expensive cases were the upgradings that required 45 minutes against the 10 minutes of the other cases.

8 Conclusions

In this paper we considered the transient analysis of closed queuing networks. Specifically, we investigated the use of the concept of the equivalent server in transient analysis, using the characterization which provides exact steady state results for some classes of networks. Similarly to what is done for the steady state behavior, we showed that equivalent servers can be exact in the transient phase as well, but their characterizations require the knowledge of the solution of the whole original network and depend also on the initial location of the customers in the network. Consequently, the exact characterization does not lead to advantages from the computational point of view.

For this reason, we opted for an approximate approach. Specifically, we proposed to use the steady state characterization which can be efficiently computed considering in isolation only those stations that we aim to aggregate, thus leading to a significant computational gain with respect analyzing the original network. Naturally, in many cases this characterization leads to highly inaccurate results in the transient phase. For this reason, we provided structural conditions and devised quantitative tests to ensure that the approximation yields reasonable results.

As future work we plan to continue to study this difficult and important problem with the objective of defining a characterization that stays "between" the exact one and that used in this paper. This would increase the computational cost with respect to the approximation introduced in this work, but would provide more accurate results. We also plan to apply the method to a wider range of models, including those represented by open networks, and to devise new structural conditions that will allow to identify classes of models that are intrinsically well suited for this approach.

References

1. Balbo, G., Bruell, S.C.: Calculation of the Moments of the Waiting Time Distribution of FCFS Stations in Product Form Queueing Networks. Computer Performance **4**(2) (1983)
2. Balsamo, S., Iazeolla, G.: An Extension of Norton's Theorem for Queueing Networks. IEEE Trans. on Software Eng. **SE-8** (1982)
3. Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J. ACM **22**(2), 248–260 (1975)
4. Boucherie, R., Taylor, P.: Transient Product Form Distributions in Queueing Networks. Discrete Event Dynamics Systems: Theory and Applications **3**, 375–396 (1993)
5. Bruell, S.C., Balbo, G.: Computational Algorithms for Closed Queueing Networks. The Computer Science Library. Elsevier North Holland (1980)
6. Buchholz, P.: Hierarchical Markovian Models - Symmetries and Reduction, pp. 234–246. Elsevier Science Publishers B. V. (1992)

7. Buzen, J.P.: Computational algorithms for closed queueing networks with exponential servers. Commun. ACM **16**(9), 527–531 (1973). DOI 10.1145/362342.362345
8. Chandy, K.M., Herzog, U., Woo, L.: Parametric analysis of queueing networks. IBM Journal of Res. and Dev. **1**(1), 36–42 (1975)
9. Chandy, K.M., Sauer, C.H.: Approximate Methods for Analyzing Queueing Network Models of Computing Systems. ACM Comput. Surv. **10**(3), 281–317 (1978). DOI 10.1145/356733.356737. URL http://doi.acm.org/10.1145/356733.356737
10. Chandy, K.M., Sauer, C.H.: Computational algorithms for product form queueing networks. Commun. ACM **23**(10), 573–583 (1980)
11. Denning, P.J., Buzen, J.P.: The Operational Analysis of Queueing Network Models. ACM Comput. Surv. **10**(3), 225–261 (1978). DOI 10.1145/356733.356735. URL http://doi.acm.org/10.1145/356733.356735
12. Gordon, W.J., Newell, G.F.: Cyclic queueing networks with restricted length queues. Operations Research **15**(2), 266–277 (1967)
13. Jackson, J.R.: Jobshop-Like Queueing Systems. Management Science **10**, 131–142 (1963)
14. Kelly, F.: Reversibility and Stochastic Networks. Wiley (1979)
15. Kritzinger, P., van Wyk, S., Krezesinski, A.: A generalization of Norton's theorem for multiclass queueing networks. Perform. Eval., Elsevier **2**, 98–107 (1982)
16. Lavenberg, S.S.: Computer Performance Modeling Handbook. Academic Press, New York (1983)
17. Menasce, D.A., Almeida, V.: Capacity Planning for Web Services: Metrics, Models, and Methods, 1st edn. Prentice Hall PTR, Upper Saddle River, NJ, USA (2001)
18. Menasce', D.A., Almeida, V.A.F., Dowdy, L.W.: Performance by Design: Computer Capacity Planning by examples. Prentice Hall (2004)
19. Resiser, M., Lavenberg, S.S.: Mean Value Analysis of Closed Multichain Queueing Network. J. ACM **27**(2), 313–320 (1980)
20. Silva, M., Balbo, G.: Performance Models for Discrete Event Systems with Synchronisations: Formalisms and Analysis Techniques. Editorial KRONOS, Zaragoza, Spain (1998). URL http://webdiis.unizar.es/GISED/?q=news/matchbook
21. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press (1995)

A Derivation of routing probabilities in Theorem 1

If we detail the expression represented by Equation 7, we obtain

$$\begin{cases} V_i = \sum\limits_{j\in\mathcal{K}'} V_j\, r_{j,i} + \sum\limits_{h\in\mathcal{K}} V_h\, r_{h,i} \; i \in \mathcal{K}' \\ V_l = \sum\limits_{j\in\mathcal{K}'} V_j\, r_{j,l} + \sum\limits_{h\in\mathcal{K}} V_h\, r_{h,l} \; l \in \mathcal{K} \end{cases} \tag{21}$$

which can be rewritten in the following way

$$\begin{cases} V_i = \sum\limits_{j\in\mathcal{K}'} V_j\, r_{j,i} + \sum\limits_{h\in\mathcal{K}} V_h\, r_{h,i} & i \in \mathcal{K}' \\ \sum\limits_{l\in\mathcal{K}} V_l = \sum\limits_{j\in\mathcal{K}'} V_j \sum\limits_{l\in\mathcal{K}} r_{j,l} + \sum\limits_{h\in\mathcal{K}} V_h \left[1 - \sum\limits_{j\in\mathcal{K}'} r_{h,j}\right] \end{cases} \tag{22}$$

and subsequently re-organized to obtain

$$\begin{cases} V_i = \sum\limits_{j\in\mathcal{K}'} V_j\, r_{j,i} + \sum\limits_{h\in\mathcal{K}} V_h\, r_{h,i} \; i \in \mathcal{K}' \\ \sum\limits_{j\in\mathcal{K}'} \sum\limits_{h\in\mathcal{K}} V_h\, r_{h,j} = \sum\limits_{j\in\mathcal{K}'} V_j \sum\limits_{l\in\mathcal{K}} r_{j,l} \end{cases} \tag{23}$$

If we now define (see also [1])

$$V_{eq} \;=\; \sum_{j\in\mathcal{K}'}\sum_{h\in\mathcal{K}} V_h\, r_{h,j} \tag{24}$$

we can transform the previous system of equations as follows

$$\begin{cases} V_i \;=\; \displaystyle\sum_{j\in\mathcal{K}'} V_j\, r_{j,i} \;+\; V_eq\, \frac{\displaystyle\sum_{h\in\mathcal{K}} V_h\, r_{h,i}}{\displaystyle\sum_{j\in\mathcal{K}'}\sum_{h\in\mathcal{K}} V_h\, r_{h,j}} \quad i\in\mathcal{K}' \\ V_{eq} \;=\; \displaystyle\sum_{j\in\mathcal{K}'} V_j \sum_{l\in\mathcal{K}} r_{j,l} \end{cases} \tag{25}$$

which proves our theorem when we define

$$\begin{cases} r_{eq,eq} = 0 \\ r_{i,eq} \;= \displaystyle\sum_{l\in\mathcal{K}} r_{i,l} \qquad i\in\mathcal{K}' \\ r_{eq,j} \;= \frac{\displaystyle\sum_{h\in\mathcal{K}} V_h\, r_{h,j}}{\displaystyle\sum_{l\in\mathcal{K}'}\sum_{h\in\mathcal{K}} V_h\, r_{h,l}} \quad j\in\mathcal{K}' \end{cases} \tag{26}$$

B Derivation of the transient equivalent server of a small network

The following example gives an illustration of the exact transient aggregation described above and clarifies the relevance of this discussion.

Consider a network of three queues with routing probabilities $r_{1,3} = r_{3,2} = 1, r_{2,1} = r_{2,3} = 1/2$, service rates $\mu_1 = 4, \mu_2 = 3, \mu_3 = 2$ and with two clients. The corresponding infinitesimal generator is

$$\boldsymbol{Q} = \begin{vmatrix} -2 & 2 & 0 & 0 & 0 & 0 \\ \frac{3}{2} & -5 & \frac{3}{2} & 2 & 0 & 0 \\ 4 & 0 & -6 & 0 & 2 & 0 \\ 0 & \frac{3}{2} & 0 & -3 & \frac{3}{2} & 0 \\ 0 & 4 & \frac{3}{2} & 0 & -7 & \frac{3}{2} \\ 0 & 0 & 4 & 0 & 0 & -4 \end{vmatrix}$$

where the states are ordered as $|0,0,2|, |0,1,1|, |1,0,1|, |0,2,0|, |1,1,0|, |2,0,0|$. Let us assume that both clients are at the third queue initially, and denote the transient probabilities of the network by $\pi_i(t)$ where i is one of the states of the network. The system of ordinary differential equations for this model is

$$\begin{cases} \frac{\pi_{|0,0,2|}(t)}{dt} \;=\; -\pi_{|0,0,2|}(t)\mu_3 + \pi_{|0,1,1|}(t)\mu_2 r_{2,3} + \pi_{|1,0,1|}(t)\,\mu_1 \\ \\ \frac{d\pi_{|0,1,1|}(t)}{dt} \;=\; \pi_{|0,0,2|}(t)\,\mu_3 - \pi_{|0,1,1|}(t)\,(\mu_2{+}\mu_3) + \pi_{|0,2,0|}(t)\,\mu_2 r_{2,3} + \pi_{|1,1,0|}(t)\,\mu_1 \\ \frac{d\pi_{|1,0,1|}(t)}{dt} \;=\; \pi_{|0,1,1|}(t)\,\mu_2 r_{2,1} - \pi_{|1,0,1|}(t)\,(\mu_1{+}\mu_3) + \pi_{|1,1,0|}(t)\,\mu_2 r_{2,3} + \pi_{|2,0,0|}(t)\,\mu_1 \\ \\ \frac{d\pi_{|0,2,0|}(t)}{dt} \;=\; \pi_{|0,1,1|}(t)\,\mu_3 - \pi_{|0,2,0|}(t)\,\mu_2 \\ \frac{d\pi_{|1,1,0|}(t)}{dt} \;=\; \pi_{|1,0,1|}(t)\,\mu_3 + \pi_{|0,2,0|}(t)\,\mu_2 r_{2,1} - \pi_{|1,1,0|}(t)\,(\mu_1{+}\mu_2) \\ \frac{d\pi_{|2,0,0|}(t)}{dt} \;=\; \pi_{|1,1,0|}(t)\,\mu_2 r_{2,1} - \pi_{|2,0,0|}(t)\,\mu_1 \end{cases}$$

where we highlighted the groups of equations corresponding to specific values of the number of customers at the third station (the first equation refers to the state with all the customers on the third station; the second group of equations corresponds to 1 customer at the third station; and finally the last group to the case when there are no customers at the third station). Summing up the equations for each group we get

$$\begin{cases} \frac{d\pi_{|0,0,2|}(t)}{dt} = -\pi_{|0,0,2|}(t)\,\mu_3 + \pi_{|0,1,1|}(t)\,\mu_2 r_{2,3} + \pi_{|1,0,1|}(t)\,\mu_1 \\ \\ \frac{d[\pi_{|0,1,1|}(t)+\pi_{|1,0,1|}(t)]}{dt} = \pi_{|0,0,2|}(t)\,\mu_3 - [\pi_{|0,1,1|}(t) + \pi_{|1,0,1|}(t)]\,\mu_3 \\ \qquad -[\pi_{|0,1,1|}(t)\mu_2 r_{2,3} + \pi_{|1,0,1|}(t)\mu_1] \\ \qquad +[\pi_{|0,2,0|}(t)\,\mu_2 r_{2,3} + \pi_{|1,1,0|}(t)\,(\mu_1 + \mu_2 r_{2,3}) + \pi_{|2,0,0|}(t)\,\mu_1] \\ \\ \frac{d[\pi_{|0,2,0|}(t)+\pi_{|1,1,0|}(t)+\pi_{|2,0,0|}(t)]}{dt} = [\pi_{|0,1,1|}(t) + \pi_{|1,0,1|}(t)]\,\mu_3 \\ \qquad -[\pi_{|0,2,0|}(t)\,\mu_2 r_{2,3} + \pi_{|1,1,0|}(t)\,(\mu_1 + \mu_2 r_{2,3}) + \pi_{|2,0,0|}(t)\,\mu_1] \end{cases}$$

The left hand sides of these three equations express the derivatives of the probabilities of the aggregated states $|0,2|, |1,1|, |2,0|$ which correspond to lumping together stations 1 and 2. Looking at the right hand sides, we can identify the speed at which the lumped stations send clients to the third one. Indeed, when the state of the aggregated network is $|1,1|$ the lumped stations send client to the third queue with intensity

$$\pi_{|1,0,1|}(t)\mu_1 + \pi_{|0,1,1|}(t)\mu_2 r_{2,3}$$

and when the state of the aggregated network is $|2,0|$ the intensity is

$$(\pi_{|2,0,0|}(t) + \pi_{|1,1,0|}(t))\mu_1 + (\pi_{|1,1,0|}(t) + \pi_{|0,2,0|}(t))\mu_2 r_{2,3}$$

These intensities are expressed in terms of the transient probabilities $\pi_i(t)$of the original model state.

By defining the probability distribution of the aggregated network

$$\begin{aligned} \tilde{\pi}_{|0,2|}(t) &= \pi_{|0,0,2|}(t) \\ \tilde{\pi}_{|1,1|}(t) &= \pi_{|0,1,1|}(t) + \pi_{|1,0,1|}(t) \\ \tilde{\pi}_{|2,0|}(t) &= \pi_{|0,2,0|}(t) + \pi_{|1,1,0|}(t) + \pi_{|2,0,0|}(t) \end{aligned}$$

and the conditional probabilities of finding the aggregated stations in a specific state, given the total number of customers in the aggregation

$$\begin{aligned} &\nu_{|0,0|}(0,t) = \frac{\pi_{|0,0,2|}(t)}{\tilde{\pi}_{|0,2|}(t)} \\ &\nu_{|0,1|}(1,t) = \frac{\pi_{|0,1,1|}(t)}{\tilde{\pi}_{|1,1|}(t)} \qquad \nu_{|1,0|}(1,t) = \frac{\pi_{|1,0,1|}(t)}{\tilde{\pi}_{|1,1|}(t)} \\ &\nu_{|0,2|}(2,t) = \frac{\pi_{|0,2,0|}(t)}{\tilde{\pi}_{|2,0|}(t)} \qquad \nu_{|1,1|}(2,t) = \frac{\pi_{|1,1,0|}(t)}{\tilde{\pi}_{|2,0|}(t)} \qquad \nu_{|2,0|}(2,t) = \frac{\pi_{|2,0,0|}(t)}{\tilde{\pi}_{|2,0|}(t)} \end{aligned}$$

it is possible to re-write the reduced system of differential equations in the following manner

$$\begin{cases} \frac{d\tilde{\pi}_{|0,2|}(t)}{dt} = -\tilde{\pi}_{|0,2|}(t)\,\mu_3 + \tilde{\pi}_{|1,1|}(t)\left[\nu_{|0,1|}(1,t)\,\mu_2 r_{2,3} + \nu_{|1,0|}(1,t)\,\mu_1\right] \\ \frac{d\tilde{\pi}_{|1,1|}(t)}{dt} = \tilde{\pi}_{|0,2|}(t)\,\mu_3 - \tilde{\pi}_{|1,1|}(t)\,\mu_3 - \tilde{\pi}_{|1,1|}(t)\left[\nu_{|0,1|}(1,t)\,\mu_2 r_{2,3} + \nu_{|1,0|}(1,t)\,\mu_1\right] \\ \qquad +\tilde{\pi}_{|2,0|}(t)\left[\nu_{|0,2|}(2,t)\,\mu_2 r_{2,3} + \nu_{|1,1|}(2,t)\,(\mu_1 + \mu_2 r_{2,3}) + \nu_{|2,0|}(2,t)\,\mu_1\right] \\ \frac{d\tilde{\pi}_{|2,0|}(t)}{dt} = \tilde{\pi}_{|1,1|}(t)\,\mu_3 \\ \qquad -\tilde{\pi}_{|2,0|}(t)\left[\nu_{|0,2|}(2,t)\,\mu_2 r_{2,3} + \nu_{|1,1|}(2,t)\,(\mu_1 + \mu_2 r_{2,3}) + \nu_{|2,0|}(2,t)\,\mu_1\right] \end{cases}$$

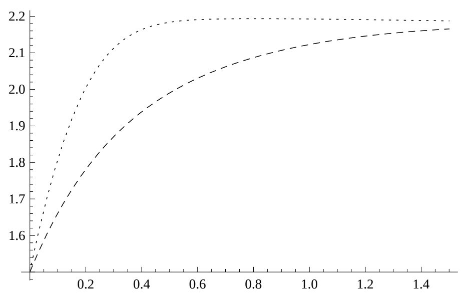


Fig. 24 Example for equivalent flow intensity for: starting from state $|0,0,2|$ (dashed) and from state $|0,1,1|$ (dotted)

	$t = 0.1$	$t = 1$	$t = \infty$
$\tilde{\pi}_0(t)$	0.831318	0.416356	0.386059
$\tilde{\pi}_1(t)$	0.152926	0.34847	0.353887
$\tilde{\pi}_2(t)$	0.015756	0.235174	0.260054

Table 1 Probability of having 0, 1 and 2 clients in the first two stations of the network starting from state $|0,0,2|$.

where, for example, the rate of the inhomogeneous Markov chain from state $|1,1|$ to state $|0,2|$ is

$$\tilde{q}_{|1,1|,|0,2|}(t) = \nu_{|0,1|}(1,t)\,\mu_2 r_{2,3} + \nu_{|1,0|}(1,t)\,\mu_1 \tag{27}$$

which corresponds also to a specific instance of (17).

The above 3 differential equations give the exact characterization of the behavior of the original model; the obvious downside of this solution is that we used the transient probabilities of the original network to describe the aggregated network. Moreover, the equivalent time-dependent service intensities depend on the initial conditions. This is illustrated in Figure 24 where we depicted the values of the aggregate rate expressed by (27) as a function of time for two different initial states.

For sake of reproducibility in Table 1 we provide the probability of having 0, 1 and 2 clients in the first two stations of the network starting from state $|0,0,2|$ for some time points.