Response time distributions and network perturbation into product-form

Peter G. Harrison[*] Maria G. Vigliotti[*]

ABSTRACT

Two new methodological results are obtained: first, a way to perturb a network into one with a product-form solution for its equilibrium state probabilities, and secondly, a new compositional approach to deriving corresponding response time distributions. The Reversed Compound Agent Theorem (RCAT) is used to construct suitable perturbations in near-product-form networks that render them separable by satisfying the conditions of this theorem. Response time calculations in stochastic networks are usually developed in terms of sample path analyses beginning in an equilibrium state. We consider the joint probability distribution of the sojourn times of a tagged task at each node of a path in a network and observe that this is the same in both the forward and reversed processes. Therefore if the reversed process is known, each node-sojourn time can be taken from either process. In particular, the reversed process can be used for the first node in a path and the forward process for the other nodes in a recursive analysis. This approach derives, quickly and systematically, existing results for response time probability densities in tandem, open and closed tree-like, and overtake-free Markovian networks of queues. An example shows that the technique is far more widely applicable, constructing a perturbed network with product-form, a new result in its own right, and then finding a very simple expression for its response time probability density function.

Categories and Subject Descriptors

B.8 [**PERFORMANCE AND RELIABILITY**];
G.3 [**PROBABILITY AND STATISTICS**];
C.4 [**PERFORMANCE OF SYSTEMS**]

[*]Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ {pgh,mgv98}@doc.ic.ac.uk.

The work presented was supported by the Engineering and Physical Sciences Research Council of the UK under research grant number EP/D047587/1.

VALUETOOLS 2009, October 20-22, Pisa, Italy
Copyright © 2009 ICST 978-963-9799-70-7
DOI 10.4108/ICST.VALUETOOLS2009.7776

Keywords

Queuing theory, Product-form solutions, Response time, Reversed process.

1. INTRODUCTION

Response times, or sojourn times, are an important quality of service (QoS) metric in many operational systems such as computer networks, logistical systems and emergency services. For example, ambulances in the United Kingdom are under contract to arrive at the scene of a life-threatening emergency within 8 minutes at least 75% of the time. For on-line transaction processing (OLTP) and other real-time systems, quantiles are often specified in Service Level Agreement contracts and industry benchmarks such as TPC-C, which specifies the 90^{th} percentile of response time [12].

The response time of a particular, 'tagged' task along a path in a network of nodes of some kind may be defined as the sum of the sojourn times of the task (i.e. its delays) at those nodes that constitute the path. More generally, the response time probability distribution follows directly from the joint probability distribution of the node-sojourn times. For a path comprising the sequence of nodes $(1, 2, \ldots, m)$, let the response time $R = T_1 + T_2 + \ldots + T_m$, where T_i is the sojourn time at node $i, (1 \leq i \leq m)$, with probability distribution function $T_i(t)$. Then the joint sojourn time distribution is $J(t_1, \ldots, t_m) = \mathbb{P}(T_1 \leq t_1, \ldots, T_m \leq t_m)$ and, denoting Laplace-Stieltjes transforms (LSTs) by asterisks, the m-dimensional LST of the joint sojourn time distribution is

$$J^*(\theta_1, \ldots, \theta_m) = \int_0^\infty \ldots \int_0^\infty e^{-(\theta_1 t_1 + \ldots + \theta_m t_m)} \mathbf{d}J(t_1, \ldots, t_m)$$

Clearly, the response time distribution then has LST $R^*(\theta) = J^*(\theta, \ldots, \theta)$. When the sojourn times T_i are independent, this simplifies to $R^*(\theta) = \Pi_{i=1}^m T_i^*(\theta)$.

If the sojourn time at each node i depends solely on the state, N_i say, existing at the node *immediately prior to the arrival of the tagged task*, the conditional joint sojourn time LST is $J^*(\theta_1, \ldots, \theta_m \mid \mathbf{n}) = \Pi_{i=1}^m T_i^*(\theta_i \mid n_i)$ where $T_i^*(\theta_i \mid n_i) = \int_0^\infty e^{-\theta_i t} \mathbf{d}\mathbb{P}(T_i \leq t \mid N_i = n_i)$[1]. In such networks, response time distributions can be computed iteratively through their LSTs using the result that:

$$J^*(\theta_1, \ldots, \theta_m \mid \mathbf{l}) = \Pi_{i=1}^m T_i^*(\theta_i \mid n_i) \mathbb{P}(\mathbf{N} = \mathbf{n} \mid \mathbf{L}(0) = \mathbf{l})$$

[1]For example, when $m = 2, J^*(\theta_1, \theta_2 \mid \mathbf{N} = \mathbf{n}) = \mathbb{E}[\mathbb{E}[e^{-(\theta_1 T_1 + \theta_2 T_2)} \mid T_1, \mathbf{N} = \mathbf{n}] \mid \mathbf{N} = \mathbf{n}] = \mathbb{E}[e^{-\theta_1 T_1} \mathbb{E}[e^{-\theta_2 T_2} \mid T_1, \mathbf{N} = \mathbf{n}] \mid \mathbf{N} = \mathbf{n}] = \mathbb{E}[e^{-\theta_1 T_1} \mathbb{E}[e^{-\theta_2 T_2} \mid N_2 = n_2] \mid N_1 = n_1]$.

where bold type indicates vectors and the random variable $L_i(t)$ is the state of node i at time t, so that the initial state is $\mathbf{L}(0)$ and $N_i = L_i(T_i^-)$ when the tagged task arrives at node i at time T_i. Of course, if the N_i are independent for $i = 1, \ldots, m$, this reduces to the above result that $J^*(\theta_1, \ldots, \theta_m) = \Pi_{i=1}^m T_i^*(\theta_i)$.

In queueing networks it is often the case that the node sojourn times depend only on the queue length at the arrival instant, for example in the overtake-free networks of [11], but the computation of the transient probabilities $\mathbb{P}(\mathbf{N} = \mathbf{n} \mid \mathbf{L}(0) = \mathbf{l})$ is problematic; see [7] for example. If these probabilities can be found (or avoided), the method applies in both open and closed networks; see the above citations, for example.

We apply a completely different approach to the computation of the LSTs of response times in Markovian networks at equilibrium, via joint sojourn time distributions and using reversed processes. Reversed processes are, loosely, stochastic processes on the same state space as the original, forward process, where the direction of time is reversed. The key idea of our method is based on the observation that sojourn time distributions are the same whether one considers the forward process or its reversed process. When sojourn times depend only on the state existing at a node at the arrival instant and the reversed process is separable, e.g. a pairwise synchronising network of m reversed nodes, we can use the forward sojourn time at the nodes $2, \ldots, m$ in the 'tail' of a path and the reversed sojourn time at the first node 1, the 'head' of the path. A recursive analysis allows us to consider only the case $m = 2$, the tail-nodes $2, \ldots, m$ constituting a single aggregate 'super-node' in the recursion.

In the next section, we consider a tandem pair of first-come-first-served (FCFS) queues, in which the first node has Erlang-2 service times (sum of two independent, identical, exponential random variables) and the second is an M/M/1 queue. Under FCFS queueing discipline, this network is known not to have a product-form solution. However, with the insight of the reversed process, we modify the first node of the network (Erlang-2 service) and obtain a perturbed queueing network in which there are additional external arrivals at the second queue – effectively an interrupted Poisson process – when and only when the first queue is in its second phase of service. In section 3, we explain our method to derive response time distributions using reversed processes, and in section 4 we show in detail how it works for standard queueing networks and for the new network constructed in section 2. The paper concludes in section 5, where future potential of the method is evaluated.

2. A NON-PRODUCT-FORM NETWORK

Before delving further into the objective-topic, we first introduce a simple formalism that allows us to describe Markov chains in terms of labelled transition diagrams, and so describe queueing networks as interactions amongst them. The formalism, similar to Stochastic Automata Networks (SAN) [2], allows us to simply express the RCAT theorem on which our product-forms depend [5], and is somewhat simpler than process algebra, which has been used for this purpose traditionally [1, 8]. Nevertheless, our work is very different from [2], where Markov chains are not described as labelled transition systems and reversed processes are not used at all.

We now introduce the notion of *Interactive Markov Automata*.

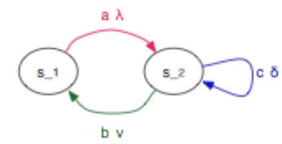

Figure 1: Graphical representation of the Markov automaton $\mathcal{M}_1$

DEFINITION 2.1. *A Markov automaton is a triple $\mathcal{M} = \langle S, Act, \rightarrow\rangle$ in which:*

1. *S is a denumerable state space, over which variables $s_1, s_2, \ldots$ range.*
2. *Act is the set of* actions*, ranged over by $a, b, \ldots$.*
3. *$\rightarrow$: $S \times Act \times \mathbb{R}^+ \times S$ is the transition relation between states where $\mathbb{R}^+$ is the set of positive real numbers.*

For readability we write $(s_1, a, \lambda, s_1) \in \rightarrow$ as $s_1 \xrightarrow{a,\lambda} s_1'$.

In essence, a Markov automaton is simply a time-homogeneous CTMC in which transitions from state to state are labelled. An example is the following:

$$\mathcal{M}_1 = \langle S_1, Act_1, \rightarrow\rangle$$

where $S_1 = \{s_1, s_2\}$, $Act_1 = \{a, b, c\}$ and $s_1 \xrightarrow{a,\lambda} s_2, s_2 \xrightarrow{b,\nu} s_1, s_2 \xrightarrow{c,\delta} s_2$.

Notice that we allow transitions from a state to itself, $s_2 \xrightarrow{c,\delta} s_2$, which does not affect the semantics of a CTMCand is usually not defined. Such transitions do have their uses, however, for example in uniformization and (as we will use them in this paper) to model 'invisible' synchronising transitions.

With each Markov automaton $\mathcal{M} = \langle S, Act, \rightarrow\rangle$, we associate a unique time-homogenous CTMC with state space S (from $\mathcal{M}$) and generator matrix $\mathbf{Q}$ in which

$$\mathbf{q}(s_i, s_j) = \sum_{(c,\lambda): s_i \xrightarrow{c,\lambda} s_j} \lambda$$

for all $s_i, s_j \in S$ and $i \neq j$, and $\mathbf{q}(s_i, s_i) = -\sum_{j \neq i} \mathbf{q}(s_i, s_j)$.

Conversely, for each time-homogenous CTMC with state space S and generator matrix $\mathbf{Q}$, it is possible to define at least one Markov automaton $\mathcal{M} = \langle S, Act, \rightarrow\rangle$ with which we would associate the given CTMC as above. For example, we could define $Act = \{a\}$ and $\rightarrow = \{(s_i, a, \lambda, s_j) : s_i, s_j \in S, q_{s_i,s_j} = \lambda > 0, i \neq j\}$. Note, however, that the relationship between Markov automata and CTMCs is not an isomorphism. The rôle of labels in Markov automata will become apparent in the definition of interactive Markov automata, where the labels will determine which actions interact and which do not. In what follows, we assume that our transition relation is a subset of $S \times Act \times (\mathbb{R}^+ \cup \mathbf{Var}) \times S$, where $\mathbf{Var}$ represents a set of variables. There are two kinds of actions in this setting: active and passive. *Active actions* are actions with an associated random *delay*, specified by a rate which is the real number parameter of a negative exponential probability distribution. *Passive actions* are actions whose delays are undefined, i.e. the rate parameter

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

is a variable. The meaning of passive action here is taken from the stochastic process algebra PEPA [8], but instead of using the symbol $\top$ to mean 'unspecified', we find it more convenient to use variables. Generally, we write x_a instead of simply x to specify that a variable is associated with an action labelled a.

Definition 2.2 (Interactive Markov automata). *Let $\mathcal{M}_1 = \langle S_1, Act_1, \rightarrow_1 \rangle$ and $\mathcal{M}_2 = \langle S_2, Act_2, \rightarrow_2 \rangle$ be two Markov automata. The* interactive Markov automaton $\mathcal{M}_1 \oplus_L \mathcal{M}_2 = \langle S, Act, \rightarrow \rangle$ *with $L \subseteq Act_1 \cap Act_2 \subset Act$ is a Markov automaton defined by:*

1. $S \stackrel{df}{=} S_1 \times S_2$.

2. $Act \stackrel{df}{=} Act_1 \cup Act_2$

3. *$\rightarrow$ is the smallest relation defined by the following rules:*

$$\frac{s_1 \xrightarrow{a,\lambda}_1 s_1' \qquad s_2 \xrightarrow{a,x_a}_2 s_2'}{(s_1,s_2) \xrightarrow{a,\lambda} (s_1',s_2')} (a \in L)$$

$$\frac{s_1 \xrightarrow{a,x_a}_1 s_1' \qquad s_2 \xrightarrow{a,\lambda}_2 s_2'}{(s_1,s_2) \xrightarrow{a,\lambda} (s_1',s_2')} (a \in L)$$

$$\frac{s_1 \xrightarrow{a,\lambda}_1 s_1'}{(s_1,s_2) \xrightarrow{a,\lambda} (s_1',s_2)} (a \notin L)$$

$$\frac{s_2 \xrightarrow{a,\lambda}_2 s_2'}{(s_1,s_2) \xrightarrow{a,\lambda} (s_1,s_2')} (a \notin L)$$

An action $a \in Act$ is said to be *enabled in state* $s_i \in S$ of a Markov automaton $\mathcal{M} = \langle S, Act, \rightarrow \rangle$ if $s_i \xrightarrow{a,\lambda} s_j$ for some $s_j \in S$ and $\lambda \in \mathbb{R}^+$ or $s_i \xrightarrow{a,x} s_j$ for some $s_j \in S$ and $x \in \mathbf{Var}$. Similarly, an action $a \in Act$ is said to be *incoming to state* $s_i \in S$ of a Markov automaton $\mathcal{M} = \langle S, Act, \rightarrow \rangle$ if $s_j \xrightarrow{a,\lambda} s_i$ for some $s_j \in S$ and $\lambda \in \mathbb{R}^+$ or $s_j \xrightarrow{a,x} s_i$ for some $s_j \in S$ and $x \in \mathbf{Var}$.

If $s_i \xrightarrow{a,\lambda} s_j$, the *instantaneous transition rate due to action* a is $\mathbf{q}(s_i, a, s_j) = \sum_{\lambda:\, s_i \xrightarrow{a,\lambda} s_j} \lambda$ and the *reversed instantaneous transition rate due to action* a is

$$\overline{\mathbf{q}}(s_j, a, s_i) = \frac{\pi_{s_i}}{\pi_{s_j}} \mathbf{q}(s_i, a, s_j)$$

In this formalism, RCAT can be expressed as follows.

Theorem 2.3 (RCAT). *Let $\mathcal{M}_1 = \langle S_1, Act_1, \rightarrow_1 \rangle$ and $\mathcal{M}_2 = \langle S_2, Act_2, \rightarrow_2 \rangle$ be Markov automata and G be an irreducible, stationary sub-chain of the Markov process associated with the interaction $\mathcal{M}_1 \oplus_L \mathcal{M}_2$, which has finite interaction set $L \subseteq Act_1 \cup Act_2$. Given that:*

1. *For $i = 1, 2$, for all active actions $a \in Act_i$ and states $s_1, s_1', s_2, s_2' \in S_i$, if $\mathbf{q}(s_1, a, s_1') \neq 0$ and $\mathbf{q}(s_2, a, s_2') \neq 0$, then $\overline{\mathbf{r}}(a) \stackrel{df}{=} \overline{\mathbf{q}}(s_1', a, s_1) = \overline{\mathbf{q}}(s_2', a, s_2)$;*

2. *for every action $a \in L$, $\{x_a\}$ satisfy the following* rate equations*:*

$$\{x_a = \overline{\mathbf{r}}(a) : a \in L\}$$

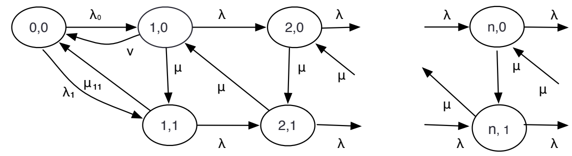

Figure 2: Transition diagram of the queue with Erlang-2 service

3. *Every passive action is enabled in all states of the automaton in which it is defined;*

4. *Every active action is incoming to all states of the automaton in which it is defined.*

then the interaction has product-form solution

$$\pi_{\mathcal{M}_1,\mathcal{M}_2}(\cdot,\cdot) \propto \pi_{\mathcal{M}_1^\dagger}(\cdot)\pi_{\mathcal{M}_2^\dagger}(\cdot)$$

where $\pi_{\mathcal{M}_\ell^\dagger}(\cdot)$ is proportional to the equilibrium state probability function of the automaton $\mathcal{M}_\ell^\dagger$ $(\ell = 1, 2)$, which is formed from $\mathcal{M}_\ell$ by replacing each occurrence of each variable x_a by its solution in the rate equations.

In what follows, we describe Markov chains by using labelled transition graphs, so that RCAT is easy to apply directly [5]. We omit the labels of all transitions that are not involved in synchronisation, i.e. not in the interaction set of actions, L.

2.1 Queue with Erlang-2 service

Consider a queue with Poisson arrivals, rate λ, and Erlang-2 (two-phase) service times with parameter μ. The state space of this queue is the set of ordered pairs $\{(0,0)\} \cup \{(n,b) \mid n \geq 1, b = 0,1\}$, where the first component is the number of tasks in the queue and the second denotes the phase of service of the task currently in service, if $n > 0$, or is 0 for the empty queue.

We require the steady state probabilities $\pi_{n,b}, n \geq 0, b = 0, 1$, for this queue, as well as its reversed instantaneous transition rates. From Figure 2 we can write down the balance equations as follows:

$$\begin{aligned} \pi_{0,0}(\lambda_0 + \lambda_1) &= \pi_{1,1}\mu_{11} + \pi_{1,0}\nu && (1) \\ \pi_{1,0}(\lambda + \mu + \nu) &= \pi_{2,1}\mu + \pi_{0,0}\lambda_0 && (2) \\ \pi_{1,1}(\mu_{11} + \lambda) &= \pi_{1,0}\mu + \pi_{0,0}\lambda_1 && (3) \end{aligned}$$

The three equations above represent the balance equations for the leftmost three states: $(0,0), (1,0), (1,1)$, which we treat as exceptional in the sense that it is the transitions amongst them that we vary in order to find suitable modifications of the standard Erlang-2 queue.

For any state $n \geq 2$ the balance equations are:

$$\begin{aligned} \pi_{n,0}(\lambda + \mu) &= \pi_{n-1,0}\lambda + \pi_{n+1,1}\mu && (4) \\ \pi_{n,1}(\lambda + \mu) &= \pi_{n-1,1}\lambda + \pi_{n,0}\mu && (5) \end{aligned}$$

For a product-form to exist in the tandem network (later), according to RCAT, the reversed rates of all instances of an actively synchronising transition must be the same. In a general, stationary, continuous time Markov chain, the reversed rate of a transition from state i to state j with

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

forwards rate q_{ij} is given by the equation

$$\pi_i q_{ij} = \bar{q}_{ji} \pi_j \tag{6}$$

where $\pi.$ is the equilibrium state probability function and the reversed rate of a transition is denoted by a bar over its forward rate [10]. Thus we require that the reversed rate of a transition resulting in a departure from the queue, i.e. from a state $(n+1,1)$ to $(n,0)$, be a constant, $\bar{\mu} = \frac{\pi_{n+1,1}\mu}{\pi_{n,0}} = \frac{\pi_{11}\mu_{11}}{\pi_{00}}$ for all $n \geq 0$.

In fact, direct substitution into the balance equations verifies that the steady state solution does indeed meet these requirements and so we can define the constant κ such that $\pi_{n+1,1} = \kappa\pi_{n,0}, \forall n \geq 2$, and $\pi_{11} = (\bar{\mu}/\mu_{11})\pi_{00} = \kappa(\mu/\mu_{11})\pi_{00}$.

Substituting into equations 4 and 5 we find that the ratio between successive states in the second phase of service is constant, $\frac{\pi_{n+1,1}}{\pi_{n,1}} = K$ say, where

$$\begin{aligned} K &= \frac{\lambda}{(\lambda+\mu) - \kappa\mu} && (7) \\ \text{and} \quad K &= \kappa^2 && (8) \end{aligned}$$

For equilibrium to exist, $K < 1$ which implies that $\kappa < 1$. Equating 7 and 8, we obtain a cubic equation for κ:

$$\mu\kappa^3 - (\lambda+\mu)\kappa^2 + \lambda = 0 \tag{9}$$

The three solutions to this equation are $\kappa = 1, \frac{\lambda \pm \sqrt{\lambda^2+4\lambda\mu}}{2\mu}$, but since $0 < \kappa < 1$, the only valid solution is

$$\kappa = \frac{\lambda + \sqrt{\lambda^2 + 4\lambda\mu}}{2\mu}$$

for which $\kappa < 1$ if and only if $\sqrt{\lambda^2 + 4\lambda\mu} < 2\mu - \lambda$, i.e. $\lambda < \mu/2$ as one would expect from traffic flow stability arguments.

Using the above recurrence equations, we now obtain for $n \geq 1$:

$$\begin{aligned} \pi_{n+1,0} &= K^n \pi_{10} \\ \pi_{n+1,1} &= \kappa K^{n-1} \pi_{10} \\ \pi_{00} &= \mu_{11}\pi_{11}/(\kappa\mu) \end{aligned}$$

The requirement on reversed rates has uniquely determined the steady state probabilities, regardless of the choice of parameters $\lambda_1, \lambda_2, \nu$ and μ_{11}. The remaining three balance equations, only two of them being independent of course, impose constraints on these four parameters. Henceforth we follow convention and assume that all service times are identically distributed Erlang-2 random variables, so that $\mu_{11} = \mu$, leaving three free parameters and two constraints. This implies that $\pi_{11} = \kappa\pi_{00}$ and hence that $\pi_{n,0} = K^n\pi_{00}, \pi_{n,1} = \kappa K^{n-1}\pi_{00}$ for all $n \geq 1$.

Normalizing, we now find

$$\pi_{00} + \sum_{i=1}^{\infty} K^i \pi_{00} + \sum_{i=1}^{\infty} \kappa K^{i-1} \pi_{00} = 1$$

so that

$$\pi_{00} = \frac{1-K}{(1+\kappa)} = 1 - \kappa$$

We have therefore established the following result.

PROPOSITION 2.4. *The continuous time Markov chain defined by the state transition diagram of Figure 2 is ergodic if* $\lambda < \mu/2$ *and its state probabilities at equilibrium are then:*

$$\begin{aligned} \pi_{00} &= 1 - \kappa \\ \pi_{n,1} &= (1-\kappa)\kappa^{2n-1} \\ \pi_{n,0} &= (1-\kappa)\kappa^{2n} \end{aligned}$$

for all $n > 1$*, where* $\kappa = \frac{\lambda + \sqrt{\lambda^2+4\lambda\mu}}{2\mu}$.

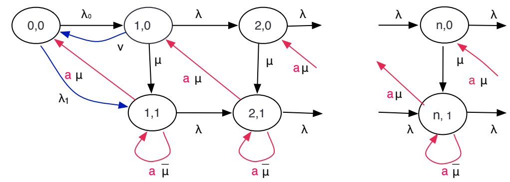

Figure 3: Transition diagram for the Erlang-2 queue in isolation

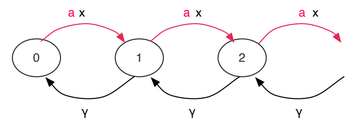

Figure 4: Transition diagram for the $M/M/1$ queue in isolation

2.2 Reversed rates

To find the product-from solution for the tandem network (in the next section) and the response time distribution's LST (section 4.2), we will need the reversed rates of the Erlang-2 queue. These follow from equation 6, already used above, for example $\bar{\mu} = \kappa\mu$. However, often they can be written down using the fact that the total instantaneous rate out of any state is the same in both the forwards and reversed processes [5]. In particular we utilize the property that $\lambda + \mu = \bar{\lambda} + \bar{\mu} = \bar{\lambda_0} + \bar{\mu} - \bar{\nu} = \bar{\lambda_1} + \bar{\mu}$, when $\mu_{11} = \mu$. Thus $\bar{\lambda} = \bar{\lambda_1} = \bar{\lambda_0}$ when $\mu_{11} = \mu$ and $\nu = 0$. In this case we have:

$$\begin{aligned} \bar{\mu} &= \kappa\mu \\ \bar{\lambda} = \bar{\lambda_0} = \bar{\lambda_1} &= \lambda/K \end{aligned}$$

Furthermore, considering the total outgoing rate from state $(0,0)$, we have $\lambda_0 + \lambda_1 = \bar{\mu}$.

2.3 Product-form solution

We now consider the tandem pair of queues where the first is as described in the previous sections and the second is an ordinary $M/M/1$ queue with arrival rate x and service rate γ. We seek a product-form solution for two-node synchronization using the original version of RCAT [5] – restated above as Theorem 2.3. There is only one synchronising transition, labelled a (coloured red) in the automaton in Figure 3, representing active departures from the Erlang-2 queue; similarly, there is only one synchronising action (passive arrivals) labelled a (coloured red) at the M/M/1 queue in the automaton of Figure 4.

Note that the Markov chain in Figure 3 describes the Erlang-2 queue with new transitions from each state $(i,1)$ to itself, with label a and rate $\bar{\mu}$. Since the new arcs do not change the Markov chain, the calculation of the steady state

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

probabilities and reversed rates carried out in the previous section are still valid. Furthermore, the reversed rate of an invisible transition is the same as its forwards rate, which ensures that the constant reversed transition rate property of the active transitions can be preserved.

Two synchronised arcs are therefore shown in Figure 3 on each state $(i, 1)$ for $i \geq 1$: the ones that have rate μ represent tasks leaving the first queue to join the second queue, while the self-loop generates the arrival of a task to the second queue without changing the state of the first queue. The specification is therefore equivalent to a normal queueing network, but with additional external arrivals at the second queue generated as an interrupted Poisson process with rate $\kappa\mu$ when and only when the first queue is in its second phase of service. For clarity, the resulting, joint state of the Markov chain, transition diagram of this quite complex tandem network of queues is shown in Figure 5.

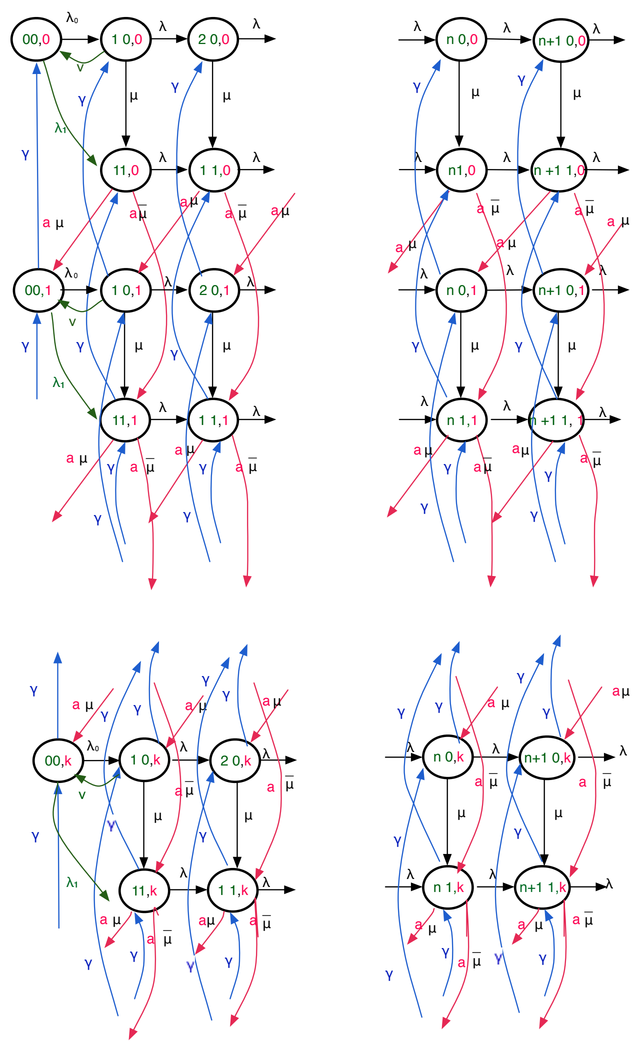

Figure 5: Transition diagram for the joint state of the tandem pair of queues

By our construction above, this network now satisfies the conditions of RCAT [5] and so has a product-form solution for its steady state probabilities when equilibrium exists.

THEOREM 2.5. *The tandem pair of nodes comprising the above modified Erlang-2 queue and an M/M/1 queue, with state transition diagram shown in Figure 5, has the following product-form for its equilibrium state probabilities* $\pi_{[(n,b),m]}$, *when* $\lambda < \min(\frac{\mu}{2}, \frac{\gamma^2}{2(2\mu+\gamma)})$:

$$\begin{aligned} \pi_{[(0,0),m]} &\propto \pi_{0,0}\rho^m \\ \pi_{[(n,0),m]} &\propto \pi_{n,0}\rho^m \\ \pi_{[(n,1),m]} &\propto \pi_{n,1}\rho^m \end{aligned}$$

for all $n \geq 1$ *and* $m \geq 0$, *where* $\rho = \frac{\kappa\mu}{\gamma}$.

PROOF. As we have already noted, the construction of the modification of the Erlang-2 queue guarantees that RCAT's conditions are met. It only remains to find the value of the passive rate x. This is given by the rate equation $x =$ the reversed rate of its active synchronising counterpart (here, two such actions), i.e. $x = 2\bar{\mu} = 2\kappa\mu$.

Finally, a steady state exists if (and only if) the given probabilities can be normalised, i.e. if $\kappa < 1$ (as we have already proved for the Erlang-2 queue) and $\kappa < \gamma/2\mu$. Together, these require that $\lambda < \mu/2$ and $\sqrt{\lambda^2 + 4\lambda\mu} < \gamma - \lambda$, i.e. $4\lambda\mu < \gamma^2 - 2\gamma\lambda$ $\square$

3. NODE-SOJOURN TIMES

Consider now the sojourn times spent by a task in a pair of nodes that are connected in the sense that the task first sojourns in node 1, for time T_1, after which it proceeds to node 2 and sojourns there, for time T_2, before departing from the system. We define the *middle state* $\mathbf{s}_0$ of the network to be that which excludes the tagged task itself at the instant when it passes from node 1 to node 2. The first component of the middle state is therefore the queue length at node 1 existing just after the instant of departure there, and the second component is the queue length existing just before the arrival instant at node 2. In many cases, e.g. a pair of tandem queues, the state $\mathbf{s}$ is a pair, $\mathbf{s} = (s_1, s_2)$, where s_i describes the state of node i only, $i = 1, 2$. We call such a state *separable*.

The sojourn time at node 1, T_1 say, can be calculated as the first passage time from the *initial state*, existing at the task's arrival instant, to exit from the state in which the task departs node 1. In general, this can involve arbitrary transitions in the whole system, i.e. be influenced by the evolution of node 2 as well as node 1. However, often, T_1 is determined solely by the initial state and the evolution of node 1, as in the case of constant rate queues, for example. In this case, the conventional approach to sojourn time analysis is to consider the state of the system at the instant of the task's departure from node 1 and use this as the initial state for the sojourn at node 2; this may also (or may not, of course) then depend solely on the evolution of node 2.

The properties we need to use this technique are therefore:

- The state of the system is separable, i.e. $\mathbf{s} = (s_1, s_2)$, where s_i describes the state of node i only, $i = 1, 2$;

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

- The sojourn time of the tagged task at each node depends *solely* on the node's state at its arrival instant – implying that the node has the 'overtake-free' property of [11] which requires that the passage of the tagged task through the node is not influenced by tasks at any other node;

- The sojourn time at each node can be characterised as a first passage time in a Markov chain describing the node's behaviour during that sojourn *insofar as it affects the tagged task*.

Notice that the last point does not necessarily require the Markov chain describing the whole system or even the node: for example a transient chain representing a queue with no arrivals is sufficient if the first property holds. This is a traditional approach that was used to obtain the Laplace transform of response time distributions in cyclic, tree-like and overtake-free networks in the 1980s [11, 7].

An alternative approach uses the observation that sojourn times are the same whether one considers the forward process or its reversed process. For example, given initial state $\mathbf{i}_0 = (i_{0;1}, i_{0;2})$ in a two-node network, we might take the sojourn time at the first node in the forward process (conditioned on $i_{0;1}$) and the *reversed sojourn time* at the second node in the reversed process, conditioned on the state existing at the end of the two sojourns. Notice that the reversed sojourn time is not necessarily dependent on only the initial state pertaining to the second node (final state in the forwards process). Indeed, the reversed process itself may depend on the joint state of the whole system, even if the forward node was overtake-free. In fact, this approach turns out to be no easier than the naive, purely 'forwards' one and a better method is as follows.

Theorem 3.1. *Suppose that a two-node Markovian network at equilibrium satisfies the following conditions:*

1. *The state of the system is separable, with middle states $(s_1, s_2) \in \mathcal{S}$ having probabilities $p_{s_1 s_2}$;*

2. *The reversed sojourn time at node 1 depends* solely *on the state existing at node 1 just after a particular, tagged task completes service at node 1 in the forwards process, i.e. on the first component of the middle state;*

3. *The forward sojourn time at node 2 depends* solely *on the state existing at node 2 just before the arrival of the tagged task there, i.e. on the second component of the middle state;*

Then the joint sojourn time probability distribution has LST

$$J^*(\theta_1, \theta_2) = \sum_{(s_1,s_2)\in\mathcal{S}} p_{s_1 s_2} \tilde{\mathcal{R}}_1^*(\theta_1)\mathcal{R}_2^*(\theta_2)$$

where $\tilde{\mathcal{R}}_1^*(\theta_1) = \tilde{T}_1^*(\theta_1 \mid S_1(T_1^+) = s_1)$, $\mathcal{R}_2^*(\theta_2) = T_2^*(\theta_2 \mid S_2(T_1^-) = s_2)$, $\tilde{T}_1^*(\theta_1 \mid T)$ denotes the conditional expectation $\mathbb{E}[e^{-\theta_1 \tilde{T}_1} \mid T]$ and similarly for $T_2^*(\theta_2 \mid T)$.

Proof. Let the reversed sojourn time at node i be denoted by $\tilde{T}_i$ and the (separable) middle state be written $\mathbf{S}(T_1)$, where the random variable $\mathbf{S}(t) = (S_1(t^+), S_2(t^-))$. Then the LST of the joint sojourn time distribution can be written

$$\begin{aligned} J^*(\theta_1, \theta_2) &= \mathbb{E}[\mathbb{E}[e^{-(\theta_1\tilde{T}_1+\theta_2 T_2)} \mid \mathbf{S}(T_1)]] \\ &= \mathbb{E}[\mathbb{E}[\mathbb{E}[e^{-(\theta_1\tilde{T}_1+\theta_2 T_2)} \mid T_2, \mathbf{S}(T_1)] \mid \mathbf{S}(T_1)]] \\ &= \mathbb{E}[\mathbb{E}[e^{-\theta_2 T_2}\mathbb{E}[e^{-\theta_1\tilde{T}_1} \mid T_2, \mathbf{S}(T_1)] \mid \mathbf{S}(T_1)]] \quad (10) \end{aligned}$$

If the reversed sojourn time at node 1 depends only on the state existing at the arrival instant of the tagged task there in the reversed process, then we have

$$J^*(\theta_1, \theta_2) \quad = \quad \mathbb{E}[\mathbb{E}[e^{-\theta_2 T_2}\mathbb{E}[e^{-\theta_1\tilde{T}_1} \mid S_1(T_1^+)] \mid \mathbf{S}(T_1)]]$$

If further, the (forward) sojourn time at node 2 depends only on its state just before the arrival, we find

$$J^*(\theta_1, \theta_2) = \mathbb{E}_{S_1,S_2}[T_2^*(\theta_2 \mid S_2(T_1^-))\tilde{T}_1^*(\theta_1 \mid S_1(T_1^+))] \quad (11)$$

The result now follows. □

The computation of $J^*(\theta_1, \theta_2)$ is usually facilitated as a first passage time calculation. We therefore add a fourth condition:

4 The sojourn, respectively reversed sojourn, time at nodes 2 and 1 can be characterised as first passage times in Markov chains describing the respective nodes' behaviour during that sojourn.

Conditions 2 and 4 are aided by a specification of the reversed process for node 1. Finding the reversed process might not be easy in general. If the reversed rates of the process are not given, then the direct calculation of such rates involves the computation of the steady state probabilities of the forward process. The forward process may be complicated but the task of finding the reversed process can sometimes be simplified by looking at certain sub-chains, which composed together yield the original process.Note that the reversed response time in a node is not, in general, a response time in the same sense and may be hard to determine even if the reversed process of the node is known. The above conditions can be relaxed, according to equation 10, but the ensuing analysis is very much more complex, involving the evolution of the joint state.

Paths of more than two nodes can be handled recursively, building a path by adding one node at a time – at each stage, a two-node path is considered comprising the current (partial) path as the second node and a new node added as the first. This method derives all the known results on response time distributions in overtake-free queueing networks, as we discuss in section 4. Furthermore, it opens the door to a variety of non-queueing applications, but it must be remembered that the above conditions can be quite tricky to apply, especially the third and fourth.

4. QUEUEING NETWORKS

Queueing networks are relatively tractable since the M/M/1 queue is *reversible*, i.e. its reversed process is the same M/M/1 queue[10, 7, 5]. Moreover, the queue left behind by any departing task comprises precisely the tasks that arrived during its sojourn. Therefore, we have the following result:

Proposition 4.1. *At equilibrium, the reversed sojourn time in an M/M/1 queue has the same probability distribution as the forward sojourn time.*

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

The result of the previous section is now easy to apply, in both open and closed queueing networks. We begin with a tandem pair and a cycle of two M/M/1 queues.

4.1 Tandem and cyclic pairs of queues

Consider first the tandem pair of queues depicted in Figure 6 – the cyclic counterpart is simply obtained by connecting the departures of the second queue to the arrivals of the first. The forward and reversed nodes are both shown;

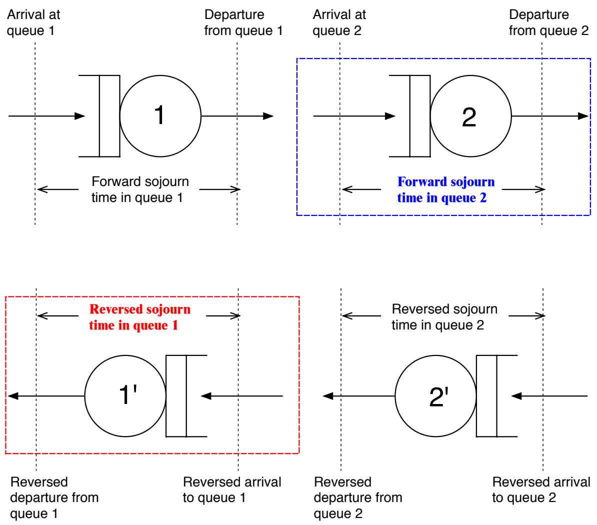

Figure 6: Two M/M/1 queues in tandem and the reversed process

correspondingly, the forward and reversed sojourn times are illustrated for both nodes, as per section 3. Possible sample paths for the node 1 and node 2 forward processes are shown in Figure 7, during the passage of the tagged task through the network. This task leaves behind a queue of length 4 (including the task in service) at node 1 on departure and finds a queue of length 3 just before its arrival at node 2, i.e. at the same instant, so that the middle state is (4,3). The traditional method of analysis investigates only forward sample paths and needs to consider the (transient) probability distribution of the node 2 queue length, starting with the middle state existing at the departure instant of the tagged task from node 1.

In our alternative approach, we consider the joint sample paths in the forward node 2 and reversed node 1 processes, beginning in a given middle state – (4,3) in the sample paths shown in Figure 8. For the forward response time at node 2, we look to the right of the vertical axis and for the reversed response time at node 1, we look to the left. However we only need to condition on the middle state. Since forward and reversed sojourn times are identically distributed by proposition 4.1, we have:

$$
\begin{aligned}
J^*(\theta_1, \theta_2) &= \mathbb{E}_{S_1,S_2}[\tilde{T}_1^*(\theta_1 \mid S_1) T_2^*(\theta_2 \mid S_2)] \\
&= \mathbb{E}_{S_1,S_2}[T_1^*(\theta_1 \mid S_1) T_2^*(\theta_2 \mid S_2)] \\
&= \sum_{n_1,n_2 \geq 0} \pi_{n_1 n_2} \left(\frac{\mu_1}{\mu_1+\theta_1}\right)^{n_1+1} \left(\frac{\mu_2}{\mu_2+\theta_2}\right)^{n_2+1}
\end{aligned}
$$

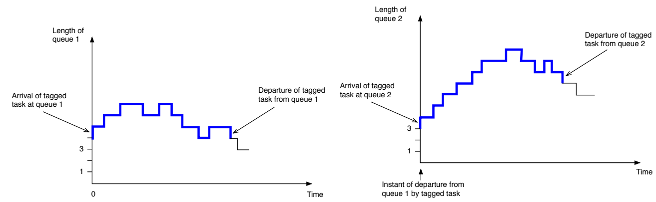

Figure 7: Possible sample paths for the queue lengths at each queue during the sojourn of the tagged task

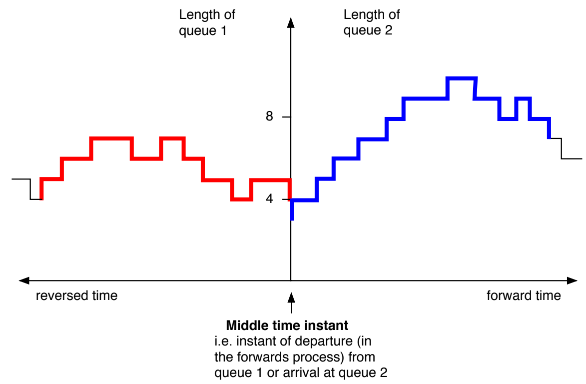

Figure 8: Forward and reversed sample paths given middle state (4,3)

The equilibrium probabilities π are the standard product-form solution [9, 4, 7]. The result therefore simplifies to:

$$
\sum_{n_1,n_2 \geq 0} \pi_{n_1} \left(\frac{\mu_1}{\mu_1+\theta_1}\right)^{n_1+1} \pi_{n_2} \left(\frac{\mu_2}{\mu_2+\theta_2}\right)^{n_2+1}
$$

To derive the reversed process and the product-from solution at the same time, one could use instead the Reversed Compound Agent Theorem [5]. This case is simple since we know what the reversed process is for node 1 – the same M/M/1 queue – but we do not know this for nodes in general.

The above result generalises inductively to overtake-free paths in both open and closed networks to give the following well known result:

PROPOSITION 4.2. *For overtake-free path* $\mathbf{z} = (z_1, \ldots, z_m)$ *in a queueing network of* M *nodes with state space* $\mathcal{S}$ *at equilibrium* ($1 \leq m \leq M$)*, the LST of the joint sojourn time probability distribution is*

$$
J^*(\theta_1, \ldots, \theta_m) = \sum_{(n_1,\ldots,n_M)\in\mathcal{S}} \pi_{n_1,\ldots,n_M} \prod_{j=1}^{m} \left(\frac{\mu_{z_j}}{\theta_j + \mu_{z_j}}\right)^{n_{z_j}+1}
$$

where $\pi_{n_1,\ldots,n_M}$ *is the equilibrium probability distribution of the network's state immediately prior to the instant of arrival of a task at any node.*

Notice that $\pi_{n_1,\ldots,n_M}$ is well defined by the arrival theorem [7], being the same as an open network's steady state probabilities (at a random time point) or the steady state probabilities of a closed network with population reduced by one, depending on whether the network in question is open or closed, respectively.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

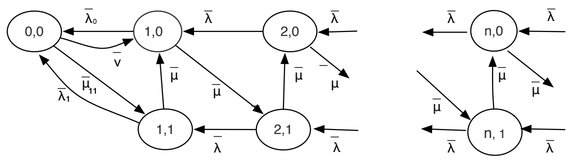

Figure 9: Transition diagram of the reversed queue with Erlang-2 service

In the case of open networks, $\pi_{n_1,\ldots,n_M}$ is a product of the form $\pi_1(n_1)\ldots\pi_M(n_M)$ where $\pi_i(n_i) = (1-x_i)x_i^{n_i}$ for some constants x_i, and so the result simplifies to

$$J^*(\theta_1,\ldots,\theta_m) = \prod_{j=1}^{m} \frac{\mu_{z_j}(1-x_{z_j})}{\theta_j + \mu_{z_j}(1-x_{z_j})}$$

This is consistent with the fact that in a tandem series of stationary M/M/1 queues with fixed-rate servers and FCFS discipline, the sojourn times of a given task in each queue are independent. Interestingly, the proof of this result uses properties of reversibility and so we include it as an appendix [10]. There is one obvious generalisation: the final queue in the series need not be M/M/1 since we are not concerned with its output.

In either approach, we observe that if service rates varied with queue length, we could not ignore tasks behind a given tagged task, even when they could not overtake, because they would influence the service rate received by the tagged task. Except in special cases, therefore, constant service rates are required.

4.2 Sojourn time and reversed sojourn time

Let us now consider the response time in the tandem pair of queues with Erlangian service times (section 2.3). To apply Theorem 3.1, we need to find the reversed sojourn time at the first node and the forward sojourn time at the second node, conditional on the middle state, which must be of the form $[(n,0),m]$ for $n,m>0$. For the second node, the conditional forward sojourn time probability density has Laplace transform $\left(\frac{\gamma}{\theta+\gamma}\right)^{m+1}$. For the first node, the required conditional reversed sojourn time probability density is given by the following proposition.

PROPOSITION 4.3. *When $\mu_{11}=\mu$, the reversed sojourn time probability density function in the Erlang-2 queue defined by Figure 2, given state $(N,0)$ just before arrival in the reversed process, has Laplace transform*

$$\tilde{S}^*(\theta \mid N) = \left(\frac{\bar{\lambda}}{\theta+\bar{\lambda}}\right)^{N+1}$$

PROOF. Looking at the reversed state transition diagram in Figure 9, it can be seen that a reversed sojourn time $\tilde{S}$ is the time elapsed from the (reversed) arrival instant, at which a transition from a state $(n,0)\to(n+1,1)$ occurs for some $n\ge 0$ until the $(n+1)$st subsequent departure. This follows from a consideration of sample paths and an argument analogous to that used for the M/G/1 queue with FCFS queueing discipline: that the tasks left behind at the node by a departing task are precisely those that arrived during that task's service time. Since the reversed arrival rate is $\bar{\lambda}$ for all arrival transitions, which originate in *all* states, the reversed arrival process is Poisson with rate $\bar{\lambda}$ and so the result follows. □

COROLLARY 4.4. *When $\mu_{11}=\mu$, the unconditional reversed sojourn time probability density function in the Erlang-2 queue is $(\bar{\lambda}-\lambda)e^{-(\bar{\lambda}-\lambda)t}$.*

PROOF. Since the reversed arrival rate is the same in all states, the probability $\pi'_{n,0}$ that the state just before a reversed arrival is $(n,0)$ is proportional to $\pi_{n,0}=\pi_{00}K^n$. Hence $\pi'_{n,0}=(1-K)K^n$. The unconditional reversed response time probability density is therefore

$$(1-K)\left(\frac{\bar{\lambda}}{\theta+\bar{\lambda}}\right)\sum_{n=0}^{\infty}\left(\frac{K\bar{\lambda}}{\theta+\bar{\lambda}}\right)^n$$

which simplifies to the result given. □

Remark:.
This result and its derivation is remarkably simple in the reversed process. In the corresponding forwards process the calculation is much more complex, involving sums of up to $2n+4$ service time random variables when the queue length on arrival is n. Tedious algebra does eventually give the same result for the unconditional forwards response time density, as indeed must be the case.

4.3 Network response time

To find the probability density function of the network's response time, we need to:

- Define the reversed sojourn time in this modified queue and find the Laplace transform of its probability density function, conditioned on the arrival-state in the reversed process;
- Find the forward response time LST for the second node, conditioned on the arrival state n – $\left(\frac{\mu_2}{\theta+\mu_2}\right)^n$ for an M/M/1 queue with service rate μ_2;
- Decondition the product of these LSTs with respect to the middle state probabilities.

Given the node sojourn time distributions, it is straightforward to check that the conditions of Theorem 3.1 are satisfied and to apply it. We need only to obtain the probability function for the middle state, using flux arguments.

PROPOSITION 4.5. *The middle state $[(n,0),m]$ of the tandem network has probability*

$$\pi'_{[(n,0),m]} = \frac{\mu_n^*(1-K)(1-\rho)K^n\rho^m}{(1-K)\mu_{11}+K\mu} \quad (m,n\ge 0)$$

at equilibrium, where $\mu_n^*=\mu$ for $n>0$ and $\mu_0^*=\mu_{11}$.

When $\mu_{11}=\mu$, we have the simpler result:

COROLLARY 4.6. *When $\mu_{11}=\mu$, the middle states have probabilities*

$$\pi'_{[(n,0),m]} = (1-K)(1-\rho)K^n\rho^m \quad (m,n\ge 0)$$

for $n,m\ge 0$.

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776

We can now state the expression for the joint sojourn time probability density function.

THEOREM 4.7. *The joint sojourn time probability density function in the tandem network, with $\mu_{11} = \mu$ and $\nu = 0$, is:*

$$\frac{\lambda(1-K)(\gamma - 2\kappa\mu)}{K(\gamma - 2\kappa\mu) - \lambda(1-K)} \left(e^{-(\lambda(1-K)/K)t_1} - e^{-(\gamma - 2\kappa\mu)t_2}\right)$$

PROOF. By Corollary 4.6, the sum of Theorem 3.1 separates and we obtain

$$J^*(\theta_1, \theta_2) = \left(\frac{\lambda(1-K)/K}{\theta_1 + \lambda(1-K)/K}\right)\left(\frac{\gamma - 2\kappa\mu}{\theta_2 + \gamma - 2\kappa\mu}\right)$$

The the result now follows by expansion into partial fractions. □

4.4 A network with feedback

Traditionally, closed networks have been considered more difficult in the analysis of sojourn times. However here, all we need do is find a value to replace the external rate λ and recalculate the middle state probabilities. RCAT is easily verified for the closed network and assigns to the passive arrival rate to node 1 (previously λ) the reversed rate of the service rate of node 2, $\bar{\gamma} = 2\kappa\mu$, where κ is a function of λ. We therefore solve, for λ, the equation $\lambda = \lambda + \sqrt{\lambda^2 + 4\lambda\mu}$, which has no non-zero solution! The closed network therefore cannot satisfy RCAT, but if we introduce an external Poisson arrival stream with rate ω at node 2, and departures with fixed probability d, the rate equation becomes:

$$\lambda = (1-d)\left(\omega + \lambda + \sqrt{\lambda^2 + 4\lambda\mu}\right)$$

We solve this equation for λ, noting that we are now dealing with a different network, which is no longer closed but does have feedback. This is a quadratic, requiring that $\omega < d\lambda/(1-d)$. Let the solution then be $\lambda = \alpha$. The joint sojourn time probability density can now be obtained from Theorem 4.7, as before.

5. CONCLUSION

Response time distributions – more generally, joint node-sojourn time distributions – can be derived much more simply and generally than previously using the reversed process of a separable network. In this way, most of the known separable solutions for the LSTs of response time distributions in queueing networks can be obtained. In [6] it was shown how to apply this approach to finding the response time probability density in G-networks, but other special cases can also be derived simply. The methodology is conducive to automation and, in fact, new product-forms for equilibrium state probabilities provide a basis for this, since they do at least provide the right, separable reversed node processes. The methodology has been demonstrated here for a non-trivial queueing network, so proving its potential. Moreover, the new product-form for this network, comprising an Erlang-2 and M/M/1 queues, is a novel result in its own right.

6. REFERENCES

[1] C. Baier, J.-P. Katoen, H. Hermanns, and V. Wolf. Comparative branching-time semantics for markov chains. *Information & Computation*, 200(2):149–214, 2005.

[2] J. M. Fourneau, B. Plateau, and W. Stewart. Product form for stochastic automata networks. In *Proceedings of ValueTools '07*, 2007.

[3] E. Gelenbe. Random neural networks with positive and negative signals and product form solution. *Neural Computation*, 1(4):502–510, 1989.

[4] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley-Sons, 1985.

[5] P.G. Harrison. Turning back time in markovian process algebra. *Theoretical Computer Science*, 290(3):1947–1986, January 2003.

[6] P.G. Harrison. Turning back time: what impact on performance? *The Computer Journal*, 2009.

[7] P.G. Harrison and Naresh M. Patel. *Performance Modelling of Communication Networks and Computer Architectures*. Addison-Wesley, 1992.

[8] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[9] J.R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, 1963.

[10] F.P. Kelly. *Reversibility and stochastic networks*. Wiley, 1979.

[11] F.P. Kelly and P.K. Pollett. Sojourn times in closed queueing networks. *Advances in Applied Probability*, 15:638–656, 1983.

[12] Transaction Processing Performance Council. *TPC benchmark C: Standard specificationrevision 5.2*. 2003.

APPENDIX

A. PROOF OF INDEPENDENCE IN TANDEM M/M/1 QUEUES

PROPOSITION A.1. *In a tandem series of stationary M/M/1 queues with fixed-rate servers and FCFS queueing discipline, the sojourn times in each queue of a tagged task are independent.*

PROOF. First we claim that the sojourn time of a tagged task, C say, in a stationary M/M/1 queue is independent of the departure process before the departure of C. This is a direct consequence of the reversibility of the M/M/1 queue. To complete the proof, let A_i and T_i denote C's time of arrival and sojourn time respectively at queue i in a series of m queues ($1 \le i \le m$). Certainly, by our claim, T_1 is independent of the arrival process at queue 2 before A_2 and so of the queue length faced by C on arrival at queue 2. Thus, T_2 is independent of T_1. Now, we can ignore tasks that leave queue 1 after C since they cannot arrive at (nor influence the rate of) any queue in the series before C, again because all queues have single servers and FCFS discipline. Thus, T_1 is independent of the arrival process at queue i before A_i and so of T_i for $2 \le i \le m$. Similarly, T_j is independent of T_k for $2 \le j < k \le m$. □

Digital Object Identifier: 10.4108/ICST.VALUETOOLS2009.7776
http://dx.doi.org/10.4108/ICST.VALUETOOLS2009.7776