Compositional reversed Markov processes, with applications to G-networks

P.G. Harrison

Dept. of Computing
Imperial College London
South Kensington Campus, London SW7 2AZ

Abstract

Stochastic networks defined by a collection of cooperating agents are solved for their equilibrium state probability distribution by a new compositional method. The agents are processes formalised in a Markovian Process Algebra, which enables the reversed stationary Markov process of a cooperation to be determined symbolically under appropriate conditions. From the reversed process, a separable (compositional) solution follows immediately for the equilibrium state probabilities. The well known solutions for networks of queues (Jackson's theorem) and G-networks (with both positive and negative customers) can be obtained simply by this method. Here, the reversed processes, and hence product-form solutions, are derived for more general cooperations, focussing on G-networks with chains of triggers and generalised resets, which have some quite distinct properties from those proposed recently. The methodology's principal advantage is its potential for mechanisation and symbolic implementation; many equilibrium solutions, both new and derived elsewhere by customized methods, have emerged directly from the compositional approach. As further examples, we consider a known type of fork-join network and a queueing network with batch arrivals.

1 Introduction

The existence of separable (compositional) stationary state probability distributions in certain queueing networks has greatly facilitated tractable models of performance over the past three decades; see [19, 1, 20, 15, 5, 7], to mention a few of many sources. However, more general stochastic networks than just networks of queues are separable at equilibrium, as in [6, 3, 2, 17, 18], for example. These have typically been derived in a rather ad-hoc way; guessing that such a solution exists, then verifying that the Kolmogorov equations of the defining Markov chain are satisfied and appealing to uniqueness. A more structured way uses properties of local balance or quasi-reversibility and verifies a set of *traffic equations*; see [3] for an excellent survey. We approach the problem in a different, hierarchical way, by seeking the *reversed process* of the Markov chain [20] in terms of the reversed processes of its sub-chains. From a reversed process, a separable solution for the equilibrium state probabilities follows immediately.

The formalism we use for this hierarchical analysis is PEPA [16], a Markovian Process Algebra (MPA), which has an appropriate recursive structure. The determination of the reversed process of a certain type of cooperation between two agents is based on the Reversed Compound Agent Theorem (RCAT) of [11]. This methodology is extended in this paper by considering multiple cooperations to facilitate the modelling of chains of immediate transitions in a set of two or more cooperating processes. It is also applied to approximating non-separable solutions, by finding perturbations to a cooperation's specifications that render it compositional.

In Section 2, the salient properties of reversed processes, our MPA-based formulation and RCAT are reviewed. The methodology is applied to G-networks with triggers and resets in section 3 and these results are extended with the introduction of negative, propagating triggers [3] and more general resets that can also propagate. These resets are quite distinct from those of [8]. Section 4 considers further applications, including a type of fork-join network (leading to the result on 'positive triggers' of [3]) and queueing networks with batches. The paper concludes in Section 5, where we assess the significance of this work and outline some directions for further research. The full statement of RCAT is given in Appendix A.

The methodology unifies many existing product-forms, derived elsewhere over many years in various, customized ways. Moreover, it generates new ones (to the author's best knowledge), such as the aforementioned G-networks with generalised resets. The RCAT-based proofs of product-forms given here are entirely novel to this paper, providing a serious alternative, or complement, to current teaching and understanding of separable Markov processes at equilibrium. The advantages for mechanisation and symbolic evaluation are clear, potentially leading to the automatic generation of steady state theorems by computer.

2 Background and previous results

2.1 Process algebraic formalism

We consider (continuous time) Markov chains that are composed of simpler chains in a particular way. In order to formulate these chains and the way they interact rigorously, we use process algebraic concepts. The *agents* of a Markovian process algebra, in which all time delays are exponential random variables, describe syntactically Markov chains with generic patterns in their generator matrices, given by the rates at which an agent's *actions* occur. For example, a single 'arrival' action can denote all transitions that increase the state of a queue-process by one unit. Moreover, synchronisation operators define how agents interact in a concise manner, using their generic actions' rates.

We use just two operators to define Markov processes, written in infix form in the syntax of the PEPA (Performance Evaluation Process Algebra) language of [16]: *prefix* and *cooperation*. New agents are named in the usual way by *assignment*, $A = P$, and we use *relabeling* $P\{y \leftarrow x\}$ to denote the process P in which all occurences of the symbol y are changed to x, which may be an expression. Thus, for example, $((a, \lambda).P)\{\lambda \leftarrow \mu\}$ denotes the agent $(a, \mu).P\{\lambda \leftarrow \mu\}$.

The prefix operator (called a 'combinator' in PEPA) is denoted by a full stop and defines a process, written in PEPA as the agent $(a, \lambda).P$, that carries out a state-transition, called an 'action', (a, λ) of *type* (or 'name') a at *rate* λ and subsequently behaves as agent P.

For example, a single server, M/M/1 queue with arrival rate λ and service rate μ can be defined by the agent Q_0:

$$\begin{aligned} Q_0 &= (a, \lambda).Q_1 \\ Q_n &= (a, \lambda).Q_{n+1} && (n > 0) \\ Q_n &= (d, \mu).Q_{n-1} && (n > 0) \end{aligned}$$

The type a is the generic name for arrivals occurring in any state of the queue and the type d is the generic name for departures occurring in any positive state. Thus a and d each denotes an infinite set of state transitions, with one syntactic *instance* of the type for each transition it denotes. The subscript 0 in this specification indicates that the queue is initially empty. If the queue is stable ($\lambda < \mu$) and we are interested in its steady state, the process is equally well described by Q_n for any $n \geq 0$.

The prefix combinator can describe every instantaneous transition between any two states of any continuous time Markov chain, and hence is sufficient alone to define *any* Markov chain. However, such a specification would usually be much more opaque than a conventional state transition graph. Hence the prefix operator is augmented with the cooperation combinator, written $\underset{L}{\bowtie}$, to facilitate hierarchical specifications.

In the cooperation $P \underset{L}{\bowtie} Q$ of two agents P and Q over a set L of action types occurring in P and Q, any action of type $a \in L$ can only take place in both processes P and Q *simultaneously*. We require that every action with type in L be active in *exactly one* of the agents P, Q and passive in (i.e. 'waits' in) the other.[1] A passive action is denoted by an unspecified rate, syntactically written

[1] This implies a restriction on the class of cooperating Markov processes we can express using

$\top$. The (joint) rate of the synchronised action is that of the active action of the pair.

For example, a tandem pair of M/M/1 queues with rates μ_1 and μ_2 and arrivals to the first queue at rate λ can be described by: $P_0 \underset{\{a\}}{\bowtie} Q_0$ where:

$$\begin{array}{rcll} P_n & = & (e,\lambda).P_{n+1} & (n \geq 0) \\ P_n & = & (a,\mu_1).P_{n-1} & (n > 0) \\ Q_n & = & (a,\top).Q_{n+1} & (n \geq 0) \\ Q_n & = & (d,\mu_2).Q_{n-1} & (n > 0) \end{array}$$

Here the external arrivals are named e, the external departures (from the second queue) are named d and the departures from the first queue that join the second queue, i.e. synchronise with its arrivals, are named a.

We denote reversed entities (agents, actions, action types, action rates) with an overbar. Thus, in the above example, $\overline{a}$ denotes the type of the reversed action with type a, indicated by a set of reversed arrows corresponding to instances of a in the Markov state transition graph. Similarly, when the context is unambiguous, $\overline{\lambda}$ denotes the rate of the reversed action of an action (a, λ). Notice that in general these rates need not be the same for all instances of $\overline{a}$.

In an agent with a bundle of multiple transitions between two states, the reversed rate of the aggregate (summed) transition is defined to be distributed amongst the reversed arcs in proportion to the forward transition rates. This definition is needed to handle components that can either proceed independently or cooperate. For example, a service completion at a queue may be able to cause either an external departure or the transfer of a customer to another queue.

2.2 Reversed Markov processes and RCAT

The reversed process of a stationary Markov process with state space S, generator matrix $Q = [q_{ij}|i,j \in S]$ and stationary probabilities $\boldsymbol{\pi}$ (such that $\boldsymbol{\pi}.\mathrm{Q} = \mathbf{0}$) is the stationary Markov process with generator matrix $Q' = [q'_{ij}|i,j \in S]$ defined by

$$q'_{ij} = \pi_j q_{ji}/\pi_i$$

and with the same stationary probabilities $\boldsymbol{\pi}$. This result is standard, see for example [20], and immediately yields a solution for $\boldsymbol{\pi}$ in the form of a product of ratios of rates in the forward and reversed processes. This is because, in an irreducible Markov process, we may choose a reference state 0 arbitrarily, find a sequence of connected states, in either the forward or reversed process, $0, \ldots, j$

this combinator and in PEPA the restriction is relaxed somewhat by allowing two active actions to synchronise. The rate of such a synchronisation is an arbitrary semantic choice. The most appropriate rate relates to the physical mechanism being described and is open to debate.

(i.e. with either $q_{i,i+1} > 0$ or $q'_{i,i+1} > 0$ for $0 \leq i \leq j-1$) for any state j and calculate

$$\pi_j = \pi_0 \prod_{i=0}^{j-1} \frac{q_{i,i+1}}{q'_{i+1,i}} = \pi_0 \prod_{i=0}^{j-1} \frac{q'_{i,i+1}}{q_{i+1,i}} \tag{1}$$

Under appropriate conditions, the reversed agent of a cooperation $P \underset{L}{\bowtie} Q$ between two agents P and Q is a cooperation between the reversed agents of P and Q, after some re-parameterisation. The precise statement of this result, the Reversed Compound Agent Theorem or RCAT, is given in Appendix A. Although expressed in process-algebraic terms, RCAT is easily applied without explicit knowledge of process algebra. All that is necessary is the generic grouping of actions (transitions) of the same type, e.g. arrivals. Its proof establishes *Kolmogorov's criteria* [11], originally established for reversible processes (see [20], for example). These state that X and Y are reversed processes of each other if and only if (a) the sum of the outgoing rates from every state (reciprocal of the mean state holding time) is the same in both X and Y; (b) for every cycle in X, the product of the rates around it is equal to the corresponding product of (reversed) rates in Y (in the opposite direction).

A synchronising action, with type $a \in L$ say, may or may not always be enabled. For example, if P and Q represent queues, a might represent a departure from queue-P (enabled in every derivative (state) of P with non-zero queue length) and an arrival to queue-Q (enabled in every state of Q). RCAT's conditions are that every passive action be enabled in every state of both the forward and reversed components (P, Q and $\overline{P}, \overline{Q}$) and that the reversed rate of every *active* action type a be constant, x_a say, over all of its instances. The enablement of action types is easily checked in each component separately (conditions 1 and 2). The reversed agents of these components are assumed known and so the reversed rate associated with each instance of an active action type (in its own participating agent) can be determined and checked if it is a constant (condition 3). The equations for these constant rates (x_a for action type a) can be posed, the existence of a solution established and the theorem applied.

2.3 Application of RCAT in practice

The practical application of RCAT, when possible, to any cooperation of two processes P and Q over a cooperation set L involves three simple steps related to each process P and Q separately. It is assumed that the symbolic reversed processes of each of these, $\overline{P}$ and $\overline{Q}$, are known. First, it is checked whether every state in P has an outgoing action of each type in L that is passive in P; similarly for Q. Then it is checked whether every state in P has an incoming action of each type in L that is active in P; similarly for Q. Suppose these conditions are satisfied. Then, from P and Q, the processes R and S respectively are constructed by symbolically setting the rates of each passive action $a \in L$

to the variable x_a. The reversed processes $\overline{R}$ and $\overline{S}$ are then computed – these are known by hypothesis. Denoting the (symbolic) reversed rates in $\overline{R}$ and $\overline{S}$ of all the active action types $a \in L$ by the expressions ν_a, constant if condition 3 of RCAT holds, the set of simultaneous equations $x_a = \nu_a$ are solved for the x_a. Assume a solution exists. Then, R^* and S^* (say) are constructed from $\overline{R}$ and $\overline{S}$ by making passive those actions $\overline{a} \in \overline{L}$ for which the corresponding actions $a \in L$ were originally active in P and Q respectively – i.e. by setting their rates to $\top$. The required reversed process of $P \underset{L}{\bowtie} Q$ is then $R^* \underset{\overline{L}}{\bowtie} S^*$.

To see the difference between the RCAT and established queueing theoretic approaches to finding product-forms, consider a two-node Jackson network [19] with external arrivals and departures at each node and transit of customers between the nodes in each direction. Essentially, RCAT looks at the synchronisations between specific *state transitions* in their respective Markov graphs. This involves solving certain linear equations (for the quantities x_a) that arise from condition 3. In contrast, applying a product-form theorem, such as Jackson's, involves solving an *explicit set* of 'traffic equations', checking the theorem's conditions and applying its result with suitable instantiations given by the solution of the traffic equations. The two solutions have to be the same by uniqueness of the invariant solution in a stationary Markov chain, and so, not surprisingly, the equations for the x_a are equivalent to the traffic equations.

To illustrate, let us follow through the steps in mechanically applying RCAT to this problem. The two queues are symmetrical, each having two arrival streams (internal, from the other node, and external) and, similarly, two departure streams. The first node, P_n say, is described as follows:

$$
\begin{aligned}
P_n &= (e_1, \lambda_1).P_{n+1} && (n \geq 0) \\
P_n &= (a_1, \top_1).P_{n+1} && (n \geq 0) \\
P_n &= (d_1, (1-p_{12})\mu_1).P_{n-1} && (n > 0) \\
P_n &= (a_2, p_{12}\mu_1).P_{n-1} && (n > 0)
\end{aligned}
$$

The external arrivals, named e_1 have rate λ_1 and the passive arrivals from the other node are named a_1. The service rate of the node is μ_1 and departures go to the other node with probability p_{12} (via a transition named a_2) or leave the network with probability $1-p_{12}$ (via a transition named d_1). The other node, Q_n say, is described similarly, interchanging the subscripts 1 and 2, and the network is defined by the cooperation $P_0 \underset{\{a_1,a_2\}}{\bowtie} Q_0$.

By symmetry, we only need to consider node P. Looking at its syntax, its passive transitions, type a_1 are enabled/outgoing in every state $n \geq 0$, so condition 1 of RCAT is satisfied. Its active synchronising transitions, type a_2, lead to all states $n-1$ with $n > 1$, i.e. again, all states $n \geq 0$. So condition 2 is also satisfied. Informally, the internal departures from each node synchronise, as *active* transitions, with the internal arrivals (*passive* transitions) at the other node. Thus, being arrivals, all passive transitions are enabled, or outgoing, in every state and there is also an active transition incoming to every state because in a single server queue, any state can be reached after a departure (the empty state included, in contrast to arrivals). Consequently, in the reversed process, there is a passive synchronising action going out of every state, the arrival that is the reverse of the corresponding forward departure.

For condition 3 of RCAT, we have to solve the equations $x_{a_i} = \nu_{a_i}$ for $i = 1, 2$, where a_i is the type of the internal arrivals to queue i (and so also the type of the internal departures from the other queue than i, i' say). ν_{a_2} is the reversed rate associated with active action type a_2 in node P. This rate is known by the hypothesis that the reversed process of P is known, but we can easily calculate it as follows using the M/M/1 basis of node P.

There are two actions that lead from P_n to P_{n-1}, $n > 1$, with types d_1, a_2 and rates in the ratio $(1 - p_{12}) : p_{12}$ respectively. Thus ν_{a_2} is the product of p_{12} and the total reversed departure rate, which is just the total forward arrival rate since the M/M/1 queue is reversible. Again, there are two 'arrival transitions', named e_1 and a_1, leading from P_{n-1} to P_n with combined rate $\lambda_1 + x_{a_1}$, after the substitution for passive action rates leading to the processes R and S in the application of RCAT. Thus $\nu_{a_2} = p_{12}(\lambda_1 + x_{a_1})$ and so we solve

$$x_{a_2} = p_{12}(\lambda_1 + x_{a_1})$$

and the symmetric equation for x_{a_1}. If these equations have a solution, condition 3 is satisfied since each ν_{a_i} is constant. We define

$$v_i = \lambda_i + x_i$$

and note that v_i satisfies precisely the traffic equations of Jackson's theorem – x_i is the *internal* traffic rate.

These entirely mechanical, transition-based steps yield the reversed processes $\overline{R}$ and $\overline{S}$ from which we obtain the reversed process of the cooperation $P_0 \underset{\{a_1,a_2\}}{\bowtie} Q_0$: $R_0^* \underset{\{\overline{a_1},\overline{a_2}\}}{\bowtie} S_0^*$ where:

$$\begin{array}{rcll} R_n^* & = & (\overline{e_1}, \frac{\lambda_1}{v_1}\mu_1).R_{n-1}^* & (n > 0) \\ R_n^* & = & (\overline{a_1}, (1 - \frac{\lambda_1}{v_1})\mu_1).R_{n-1}^* & (n > 0) \\ R_n^* & = & (\overline{d_1}, (1 - p_{21})v_1).R_{n+1}^* & (n \geq 0) \\ R_n^* & = & (\overline{a_2}, \top).R_{n+1}^* & (n \geq 0) \end{array}$$

and S_n^* is defined symmetrically.

3 Gelenbe networks and extensions

A simple G-queue is a single M/M/1 queue with negative customers, which may be represented by a normal M/M/1 queue with arrival rate λ^+ (that of the positive arrivals) and service rate $\mu + \lambda^-$, where μ is the usual rate of service and λ^- is the negative arrival rate. The reversed queue, with aggregated arrival streams, is then the same M/M/1 queue, with the same arrival and service rates, since it is reversible. In a G-network (network of G-queues), each G-queue may have multiple positive and negative arrival streams, together with a server that has multiple output channels. Each arrival stream has its own rate and each output channel is selected with a fixed probability. The reversed process is an M/M/1 queue with arrival rate equal to the total positive arrival rate of the G-queue (i.e. the sum of the rates of the positive arrival streams) and service rate equal to the sum of the total negative arrival rate of the G-queue and its service rate. The reversed process is then specified completely for this multi-arrival stream, multi-departure steam M/M/1 queue by:

- splitting its net service rate (given above) over its departure streams in proportion to the corresponding rates (of positive arrivals) in the G-queue;
- splitting its arrival rate (given above) over its arrival streams in proportion to the corresponding rates (of departures) in the G-queue, i.e. negative arrival rates and products of service rate and output channel selection probabilities.

Customers completing service normally at a node i may pass to a node j, as either a positive or a negative customer, with respective probabilities p_{ij}^+ or p_{ij}^-, or else leave the network. The generic single G-queue, for use in a network, and its reversed queue are as shown in Fig. 1. Departures from the queue shown go to another queue with probability p (p^+ if positive, p^- if negative, $p^+ + p^- = p$) or leave the network with probability $1 - p$. External (positive) arrivals have rate λ_e^+ and arrivals from other queues have rate λ_i^+. Depending on the number of links with other queues, the actions with rates λ_i^+, μp^+ and μp^- will have to be split further.

When the queue length is zero, we represent a passive negative arrival to a G-queue by an 'invisible' action from state 0 to itself, with arbitrary rate – being passive, this rate is unspecified, $\top$. In other words, negative arrivals to empty queues have no effect whilst still being enabled. The same invisible action will occur active in the reversed process, with a rate x_a, the reversed rate of the other active instances of the same type (a say), to establish condition 3 of RCAT.

Gelenbe introduced 'triggers' into G-networks in [6] as a generalisation of the concept of negative customers. A trigger is a negative customer that moves a (positive) customer to node j on arrival at (non-empty) node i with probability q_{ij}^+, with no effect on an empty queue. We call these *positive* triggers and extend

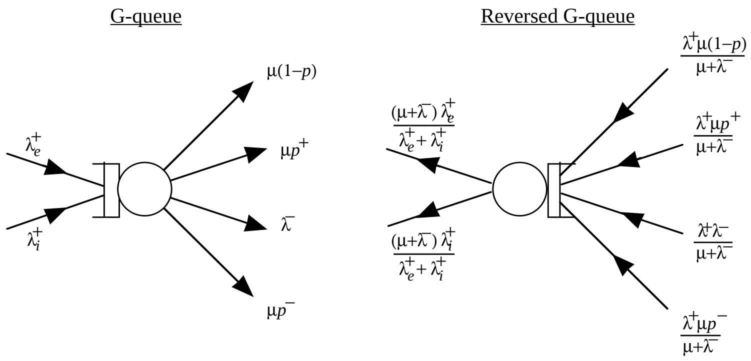

Figure 1: G-queue network node and its reversed queue

this idea to include *negative* triggers also. On arrival at node i, a trigger may be either positive or negative. A negative trigger removes a (positive) customer and sends a *trigger* to node j with probability q_{ij}^-. We write $q_{ij} = q_{ij}^+ + q_{ij}^-$, so that the probability of a trigger being positive is $\sum_j q_{ij}^+$, negative is $\sum_j q_{ij}^-$ and a conventional negative customer is $1 - \sum_j q_{ij}$. A generic negative arrival is sometimes also called a *signal*, which can either be a positive trigger, a negative trigger or only a local customer removal.

The study of product-form G-networks based on queueing theory is significantly different from conventional product-form analyses; for example the property of 'local balance' in Jackson networks ceases to hold and the traffic equations become non-linear. In contrast, the RCAT-based approach described in the previous section goes through unchanged – the only difference is that there are cooperations between one departure transition type and another as well as between departure transitions and arrival transitions.

3.1 Two G-queues with triggers

Consider now a two-node network of G-queues with general interconnections and triggers. Each node i has positive and negative external arrivals with rates λ_i^+, λ_i^- respectively, service rate μ_i, routing probabilities p_{ij}^+ for positive service completions and p_{ij}^- for negative service completions, with $p_{ij} = p_{ij}^+ + p_{ij}^-$, and trigger probabilities q_{ij}^+ for positive triggers and q_{ij}^- for negative triggers, with $q_{ij} = q_{ij}^+ + q_{ij}^-$, $i, j = 1, 2$. This network is depicted in Fig. 2.

We exclude $p_{ii} > 0$ and $q_{ii} > 0$ since these represent transitions local to node i, which must be handled by a modified cooperating component for that node before applying RCAT. This is not a difficult problem but requires more complicated notation. We defer our consideration of negative triggers to the next subsection since they require some additional, preliminary analysis. For now, therefore, we assume $q_{12}^- = q_{21}^- = 0$.

A PEPA specification of this network is straightforward to write – essentially

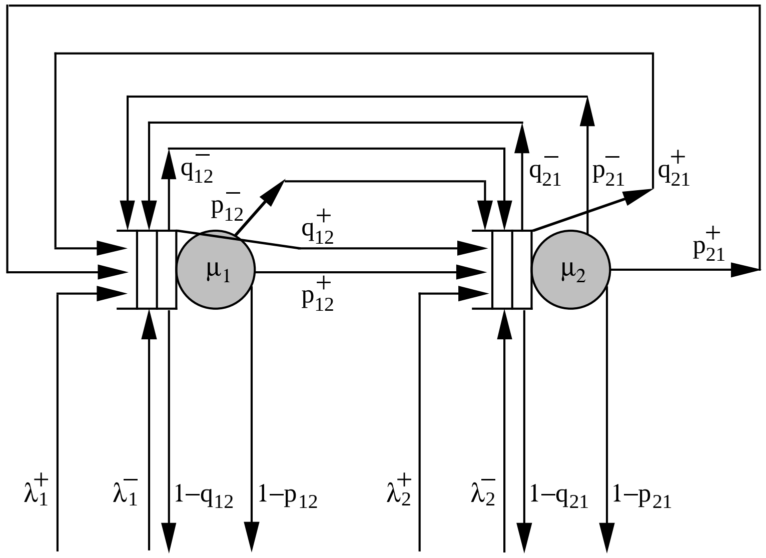

Figure 2: Network of two G-queues

isomorphic to Fig. 2. This would be used in a mechanised analysis but a diagram is probably the clearest description of a queueing network for human consumption and so we do not give the PEPA specification here.

We now apply RCAT to this G-network of two nodes. Its conditions are satisfied since:

1. Positive arrivals are always enabled in a single queue, as are negative arrivals if we use invisible actions on empty queue states (Condition 1);

2. The reversed actions of the active actions are arrivals and also always enabled (Condition 2);

3. The reversed rates of the active actions a are the rates x_a of RCAT and we will show that a solution for them exists, validating Condition 3.

We therefore solve equations for the reversed rates of the active actions in the cooperation set; bound to the passive rates in P and Q, the components describing queues 1 and 2 respectively, to give the agents R and S in RCAT. There are three such passive rates for each component: x_i^{p+} (arrivals corresponding to positive service completions at the other component), x_i^{p-} (triggers corresponding to negative service completions at the other component) and x_i^{q+} (arrivals corresponding to (positive) triggers from the other component), $i = 1, 2$ corresponding to P, Q respectively. We therefore have:

$$x_1^{p+} \quad = \quad \frac{\mu_2 p_{21}^+ (\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-}}$$

$$
\begin{aligned}
x_1^{q+} &= \frac{(\lambda_2^- + x_2^{p-})q_{21}^+(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-}} \\
x_1^{p-} &= \frac{\mu_2 p_{21}^-(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-}}
\end{aligned}
\tag{2}
$$

with similar equations for $x_2^{p+}, x_2^{q+}, x_2^{p-}$, obtained by interchanging the subscripts 1 and 2 throughout. Writing $v_1^+ = x_1^{p+} + x_1^{q+} + \lambda_1^+$, $v_1^- = x_1^{p-} + \lambda_1^-$ and v_2^+, v_2^- similarly, we find

$$
\begin{aligned}
v_1^+ &= \frac{\mu_2 p_{21}^+ v_2^+}{\mu_2 + v_2^-} + \frac{v_2^- q_{21}^+ v_2^+}{\mu_2 + v_2^-} + \lambda_1^+ \\
v_1^- &= \frac{\mu_2 p_{21}^- v_2^+}{\mu_2 + v_2^-} + \lambda_1^-
\end{aligned}
$$

with corresponding equations for v_2^+, v_2^-. These are exactly the traffic equations derived and solved by Gelenbe [6] and lead to the same product-form, as can be seen by the following argument. The instantaneous 'horizontal' rate $(i,j) \rightarrow (i+1,j)$ in the forward process is λ_1^+. Its reversed rate is therefore $(\lambda_1^+/v_1^+)(\mu_1 + v_1^-)$ and so the ratio (forward:reversed) of these rates is $y_1 = v_1^+/(\mu_1 + v_1^-)$. Similarly, the ratio of the forward:reversed 'vertical' rates between states (i,j) and $(i,j+1)$ is $y_2 = v_2^+/(\mu_2 + v_2^-)$. The product form for the stationary probability

$$
\pi_{ij} \propto y_1^i y_2^j \tag{3}
$$

then follows from equation 1 in section 2.2, choosing a path from reference state $(0,0)$ horizontally to state $(i,0)$ and then vertically to state (i,j). [2]

3.2 Multiple cooperations and negative triggers

A negative trigger instantaneously triggers a further customer removal from its destination node. If this removal is another negative trigger to a further node, a sequence of customer removals at several nodes ensues, terminating when a removed customer is *not* a negative trigger to a node with a non-empty queue. Hence, the network-state transition induced by an arriving (positive or negative) trigger at a given node is not uniquely defined locally: there could have been a sequence of queue length reductions, not just one at the adjacent source-node of the trigger. Networks with sequences of triggers are considered in depth in [3], where product forms are obtained using the notion of quasi-reversibility, itself defined in terms of reversed processes [20]. It is therefore not unexpected that such triggers can be accomodated in the present, more general approach. They

[2] As a check, note that if we chose transitions in the opposite direction, e.g. $(i+1,j) \rightarrow (i,j)$ with rate $\mu_1(1-p_{12})$, we would have reversed rate $(\mu_1(1-p_{12})/(\mu_1+v_1^-))v_1^+$ and hence reversed:forward ratio y_1, as required.

were considered in terms of inserted instantaneous states in [12], but such an extension of the process algebraic formalism is not necessary.

Multiple cooperations of this kind are easily described in MPA. Suppose agents $P_1, P_2, \ldots, P_n$ synchronise through an action with type a, which is active in P_1 and passive in $P_2, P_3, \ldots, P_n$ $(n \geq 2)$. This is described by the left-associative multiple cooperation:

$$(\ldots((P_1 \underset{\{a\}}{\bowtie} P_2) \underset{\{a\}}{\bowtie} P_3) \ldots) \underset{\{a\}}{\bowtie} P_n$$

The successive cooperations

$$P_1 \underset{\{a\}}{\bowtie} P_2,\ (P_1 \underset{\{a\}}{\bowtie} P_2) \underset{\{a\}}{\bowtie} P_3, \ldots\ldots, (\ldots((P_1 \underset{\{a\}}{\bowtie} P_2) \underset{\{a\}}{\bowtie} P_3) \ldots) \underset{\{a\}}{\bowtie} P_{n-1}$$

define new agents, in which a is active, that cooperate with the next agent in the sequence, $P_3, P_4, \ldots, P_n$ respectively. Henceforth, we assume left-associativity and omit the brackets. The following extension of RCAT applies to such multiple cooperations:

Proposition 1 *The reversed process of a multiple cooperation*

$$P_1 \underset{L_1}{\bowtie} P_2 \underset{L_2}{\bowtie} P_3 \ldots \underset{L_{n-1}}{\bowtie} P_n$$

with action types $a \in L \equiv \bigcap_{i=1}^{n-1} L_i$ that are active in P_1 and passive in $P_2, \ldots, P_n$, is given by the corresponding $n-1$ successive applications of RCAT provided that

1. *The conditions of RCAT apply for the cooperation $P_1 \underset{L_1}{\bowtie} P_2$;*

2. *An action with type $a \in L$ has a passive instance out of every state of P_i $(2 \leq i \leq n)$ and into every state of P_j $(2 \leq j \leq n-1)$;*

3. *The rate of reversed passive action type $a \in L$ in $P_i\{\top_b \leftarrow x_b | b \in \mathcal{P}_{P_i}\}$[3], for positive real numbers x_b, is constant over all instances of a $(2 \leq i \leq n-1)$;*

4. *The conditions of RCAT apply to each cooperation $P_1 \underset{L'_1}{\bowtie} P_2 \ldots \underset{L'_{i-1}}{\bowtie} P_i$ for $2 \leq i \leq n$, where $L'_j = L_j \backslash L$ $(1 \leq j \leq n-1)$.*

Comment

Notice that conditions 2 and 3 are equivalent to requiring that the conditions of RCAT apply for each cooperation $P_i\{\top_a \leftarrow x_a, \top_c \leftarrow x_c \mid a \in L, c \in \mathcal{P}_{P_i} \backslash L_i\} \underset{L_i}{\bowtie} P_{i+1}$, for positive real numbers x_a, x_c, where the relabeling of $\top_a$ makes the action types a active in P_i $(2 \leq i \leq n-1)$.

[3] $\mathcal{P}_P$ denotes the set of all passive actions in the process P.

Proof

By condition 4, we only need to consider action types $a \in L$. We claim that, for $2 \leq i \leq n-1$, each cooperation $(P_1 \underset{L_1}{\bowtie} \ldots \underset{L_{i-2}}{\bowtie} P_{i-1}) \underset{L_{i-1}}{\bowtie} P_i$ satisfies the conditions of RCAT *and* every (joint) state of the cooperation has an incoming instance of every action with type in L.

The proof of the claim is by induction on $i \geq 2$. Let $a \in L$. For $i = 2$, RCAT holds by condition 1. Hence, every state of P_1 has an incoming active instance of a; this is equivalent to passive action type $\overline{a}$ being enabled in every state of the reversed process of P_1. But by condition 2, every state of P_2 has an incoming passive instance of a. Thus, every state of $P_1 \underset{L_1}{\bowtie} P_2$ has an incoming instance of a which will be active in the cooperation with P_3. This proves the base case.

Now assume the claim is true for $n-2 \geq i \geq 2$ and consider the cooperation $Q_i \underset{L_i}{\bowtie} P_{i+1}$ where $Q_i = P_1 \underset{L_1}{\bowtie} \ldots \underset{L_{i-1}}{\bowtie} P_i$. By the induction hypothesis, incoming active action type a is enabled in every state of Q_i (giving condition 1 of RCAT). By condition 2, (outgoing) passive action type a is enabled in every state of P_{i+1} (giving condition 2 of RCAT). The reversed rate of a in Q_i is the reversed rate of the passive action type a in P_i in the cooperation Q_i, after binding of the passive rates according to RCAT, which is valid for Q_i by the induction hypothesis. By condition 3 of the proposition, this reversed rate is constant over all instances of $a \in P_i$, giving the third condition of RCAT. By condition 2, every state of P_{i+1} has an incoming passive instance of a. Hence, by the induction hypothesis, every state of $Q_i \underset{L_i}{\bowtie} P_{i+1}$ has an incoming instance of a. This proves the claim.

Finally, RCAT holds for the cooperation $Q_{n-1} \underset{L_{n-1}}{\bowtie} P_n$ by the claim and by conditions 2 and 3 of the proposition, as in the inductive step of the proof. ♠

With the invisible passive actions included at queue lengths zero, proposition 1 is satisfied for sequences of negative triggers. However, it is *not* satisfied for sequences of positive triggers since there is no incoming passive arrival to state zero. A modified network that allows sequences of positive triggers is considered in section 4.1.2.

RCAT is still valid even if there is a cycle in a sequence satisfying the conditions of proposition 1. This is because any sequence is considered two components at a time, the left of which always has the synchronising active action type a. The length of a cyclic sequence may be unbounded, e.g. $(((P \underset{\{a\}}{\bowtie} Q) \underset{\{a\}}{\bowtie} P) \underset{\{a\}}{\bowtie} Q) \underset{\{a\}}{\bowtie} \ldots$. Thus, in a cycle of negative triggers, it is possible to repeatedly reduce all the queue lengths until one has become zero.

3.2.1 Two-node G-network with negative triggers

Negative triggers in the two-node G-network of Fig. 2 produce two new passive actions with rates $\top_1^{q-}$ and $\top_2^{q-}$, increase the total 'service' rate of each node by these amounts and add two new equations to the set (2), which now becomes:

$$\begin{aligned}
x_1^{p+} &= \frac{\mu_2 p_{21}^+(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-} + x_2^{q-}} \\
x_1^{q+} &= \frac{(\lambda_2^- + x_2^{p-} + x_2^{q-}) q_{21}^+(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-} + x_2^{q-}} \\
x_1^{p-} &= \frac{\mu_2 p_{21}^-(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-} + x_2^{q-}} \\
x_1^{q-} &= \frac{(\lambda_2^- + x_2^{p-} + x_2^{q-}) q_{21}^-(\lambda_2^+ + x_2^{p+} + x_2^{q+})}{\mu_2 + \lambda_2^- + x_2^{p-} + x_2^{q-}}
\end{aligned}$$

with similar equations for $x_2^{p+}, x_2^{q+}, x_2^{p-}, x_2^{q-}$. Writing $v_1^+ = x_1^{p+} + x_1^{q+} + \lambda_1^+$ as before, but now $v_1^- = x_1^{p-} + x_1^{q-} + \lambda_1^-$, and v_2^+, v_2^- similarly, we obtain

$$\begin{aligned}
v_1^+ &= \frac{\mu_2 p_{21}^+ v_2^+}{\mu_2 + v_2^-} + \frac{v_2^- q_{21}^+ v_2^+}{\mu_2 + v_2^-} + \lambda_1^+ \\
v_1^- &= \frac{\mu_2 p_{21}^- v_2^+}{\mu_2 + v_2^-} + \frac{v_2^- q_{21}^- v_2^+}{\mu_2 + v_2^-} + \lambda_1^-
\end{aligned}$$

with corresponding equations for v_2^+, v_2^-. These equations describe a new problem and a new solution with exactly the same product-form as (3), but with different v_i^-, v_i^+ and hence y_i $(i = 1, 2)$.

Considering as an example the special case with $q_{12}^- = q_{21}^- = 1$, a negative arrival at either queue will repeatedly decrement the length of each queue alternately until one becomes zero. We explore this example further in section 4.1.

3.3 Split passive actions

Suppose state i in a cooperating component P has n outgoing instances of a passive action with type a. The cooperation is then non-deterministic and to remove the non-determinism we assign probabilities $p_1, \ldots, p_n$ to the correspondingly labeled instances $1, \ldots, n$. Such a passive action is said to be *split* (with probabilities) as opposed to *multiple*, which refers literally to multiple copies. Without loss of generality, the state entered by instance j is also j, $1 \leq j \leq n$. Then, in a cooperation with any active action (a, ν), the jth instance has rate νp_j. In the reversed cooperation, each instance, now active, cooperates at the rate given by the reversal of its component – the probabilities p_j are not used again explicitly.

The combined rate of the split passive action's instances is that of the synchronising active action a in a cooperation and so the first condition of RCAT holds. The second and third conditions apply without change, in the latter case reversing the process R in which the symbolic rate of instance j of a split passive action is bound to $x_a p_j$. The modifications to the proof are very minor and left to the reader. The proof of conservation of products of rates around cycles goes through almost verbatim, the probabilities p_j modifying the selection probabilities $\psi_{ii'}$ therein; see [11].

Split actions are actually just syntactic sugar to reduce the number of actions defined explicitly. The passive instances are equivalent to multiple actions of the same type together with corresponding multiple active actions in the other cooperating component. The jth instance of these multiple active actions has rate νp_j when the original active action had rate ν. RCAT therefore holds.

3.4 Extensions of G-networks

We can try to apply RCAT to any cooperation between two components, each defined on a flat state space (i.e. with no cooperations in these components); if its conditions are satisfied, there will be a product-form solution for an ergodic process. Consider a pair of generalised birth-death processes on the non-negative integers, where we define arrivals and departures to be subsets of the state-transitions that define a partition; i.e. every transition is either an arrival or a departure. Arrivals and departures need not be exclusively increments and decrements (respectively) in the state, nor necessarily unit. We split these arrival and departure streams by defining paired arcs in the state transition (derivation) graph, so that a departure substream cooperates, as the active actions, with a passive arrival sub-stream in the other component, causing (internal) arrivals there as well as departures locally. The other substream of a pair does not cooperate and causes only transitions local to its component. This is precisely the situation we have considered in a network of M/M/1 queues, where the arrival/departure transitions are all unit increments/decrements respectively.

Suppose now that an arrival to (generalised) queue $m = 1, 2$ in local state i causes a state change $i \to j$ with probability $r_{m;ij}$, where $\sum_j r_{m;ij} = 1$ for all states i. Thus, $r_{m;ij} = \delta_{i+1,j}$ (the Kronecker-delta) defines a conventional queueing network, and $r_{m;ij} = \alpha_m \delta_{i+1,j} + (1 - \alpha_m)\delta_{i-1,j}$ can define a conventional G-network for appropriate choices of the α_m, where we define $\delta_{-1,j} = \delta_{0j}$. Triggers can be incorporated similarly.

An arrival in state i of queue 1, say, causes a transition to state j with probability $r_{1;ij}$; these are the probabilities of section 3.3 associated with the split passive arrival-actions in state i. In a cooperation with active departures at queue 2, the first condition of RCAT is satisfied and Condition 2 is satisfied for cooperations in which all states have incoming (active) departures in queue 2.

It remains to check Condition 3, without loss of generality for queue 1. Suppose a departure is a unit state-decrement that occurs with total instantaneous rate $\mu_1 + \lambda_1^-$, arising from a pair of transitions between every pair of states $i + 1 \to i$, one transition of which cooperates with queue 2 at (active) rate λ_1^-, the other representing a local (external) departure at rate μ_1. Obviously we could also include conventional departures from queue 1 to queue 2 (that cause unit increments at queue 2) by further splitting the departure arc. The reversed rates of the departure-transitions are not at all obvious in this generalised queue, which is not reversible in general. Since μ_1 and λ_1^- are constants, Condition 3 holds if and only if the reversed rate $q'_{i,i+1}$ of the double departure transition $i+1 \to i$ with rate $q_{i+1,i} = \mu_1 + \lambda_1^-$ is the same for all $i \geq 0$. Now, referring to section 2.2, assuming the process is stationary, $q'_{i,i+1} = (\mu_1 + \lambda_1^-)\pi_{i+1}/\pi_i$. Hence Condition 3 holds if and only if π_{i+1}/π_i is constant, i.e. $\pi_i = \rho^i \pi_0$ for all $i \geq 0$ and some $\rho < 1$. In other words, we require a geometric equilibrium probability distribution for the local state space probabilities. Generalised (non-unit) departures are handled in exactly the same way, with no further complications, since the reversed rates are calculated directly.

This result applies to networks with an arbitrary number of nodes of this type, cf. section 3.5. Of course, the special cases of regular Jackson networks [19] and G-networks [5] are known to satisfy the above required property, but these networks are not unique in this.

3.4.1 Networks with reset queues

Consider a stationary generalised queue in which

$$\begin{aligned} r_{1;i+1,j} &= \delta_{i,j} \quad (i,j \geq 0) \\ r_{1;0j} &= \pi_j \quad (j \geq 0) \end{aligned}$$

where π_j is the equilibrium probability for state $j \geq 0$. Then it can be shown, for example by direct solution of the balance equations, that

$$\pi_j = (1-\rho_1)\rho_1^j$$

where $\rho_1 = \lambda_1^+/\mu_1$ and λ_1^+ is the external (positive) arrival rate. The 'arrival' transitions from states $i+1$ to i are regular negative arrivals, with rate ν, say, as considered above, and those from 0 to j have rate $\nu\pi_j$ (total reset rate also ν). Notice that the equilibrium probabilities are independent of the negative arrival rate ν. Queues similar to this were considered by Gelenbe and Fourneau in [8], where the authors called this particular type of negative arrival in state 0 a *reset* since it 'resets the queue to its steady state', rather than having no effect. Considering this queue as passive in a cooperation with triggers at another queue, the cooperation set has just one type, a say, for the active triggers and the passive negative arrivals and resets. It is clear that Conditions 1 and 2 of RCAT are satisfied, as is Condition 3 when the active queue-component is a G-queue of the kind already defined.

If the queue with resets can participate in a cooperation as the active component, Condition 3 does not hold in the above specification since the reversed rate on the reset-transitions are $q'_{j,0} = \nu\pi_j\pi_0/\pi_j = (1-\rho_1)\nu$, where ν denotes the negative arrival rate. However, the reversed rate of the negative arrivals at states

$i > 0$ is $q'_{i,i+1}\nu/(\nu+\mu_1) = (\nu+\mu_1)(\pi_{i+1}/\pi_i)\nu/(\nu+\mu_1) = \rho_1\nu$, which is only equal to the reversed reset rate when $\rho_1 = \frac{1}{2}$. Fortunately, when the queue is an active component, we can relabel the reset action types to b.[4] Condition 2 is preserved, since every state has incoming actions of both types, and Condition 3 is satisfied since the reversed rates of both action types are constant, as just shown. Resets therefore preserve a product-form solution in G-networks. Moreover, there are other Markov processes with geometric stationary state probability distributions that similarly yield product-form solutions in networks through RCAT. One example is G-networks with batch removal of customers by a negative arrival, introduced in [7] and considered further in section 4.3.

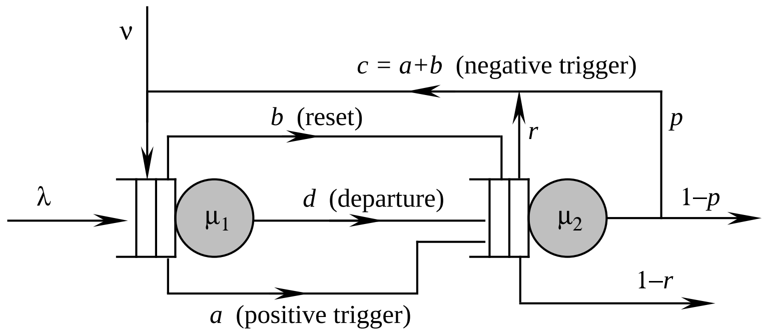

Figure 3: Active and passive cooperating resets

As an example of resets, consider the two-queue network shown in fig. 3. Queue 1 has external positive arrivals at rate λ, external negative arrivals at rate ν, service rate μ_1 and resets. Its service completions go to queue 2 as positive arrivals, with action-type d, its triggered departures (at non-zero queue lengths) go to queue 2 as positive arrivals, with action-type a, and its resets (at zero queue length) synchronise with negative arrivals at queue 2, with action-type b. Queue 2 does not have resets and its departures go to queue 1 as negative arrivals with probability p (action-type c) or leave the network with probability $1-p$. The negative arrivals synchronise with both triggers and resets at queue 1 and we indicate this by $c = a + b$ in the figure.

First consider the case $p = 0$, i.e. when there is no feedback. The synchronisation is over the cooperation set $\{a, b, d\}$, in which queue 1 is active and queue 2 is passive for all action types. Applying RCAT to the reset-queue defined above, we obtain:

$$x_d = \rho_1\mu_1$$

[4]We cannot do this when the queue is passive since each trigger with type a in the active queue would have to be replaced by a pair with types a and b. To satisfy Condition 1, in the passive queue, invisible actions of type b would have to be introduced at every state $i > 0$. But then, the triggers b in the active queue would proceed without causing a transition in the (non-empty) passive queue.

$$\begin{aligned} x_a &= \rho_1 \nu \\ x_b &= (1-\rho_1)\nu \end{aligned}$$

where $\rho_1 = \lambda/\mu_1$. The product-form solution may be obtained by taking a forward path from the reference state (0,0) following external arrivals in the first (queue 1) dimension and external arrivals in the reversed process in the second (queue 2) dimension.[5] We obtain $\pi_{ij} = (1-\rho_1)(1-y)\rho_1^i y^j$ where

$$y = \frac{x_a + x_d}{\mu_2 + x_b} = \frac{(\nu + \mu_1)\rho_1}{\mu_2 + (1-\rho_1)\nu}$$

.

When $p > 0$, there are two kinds of cooperations. As for the case $p = 0$, the first has cooperation subset $\{a, b, d\}$, with queue 1 active and queue 2 passive. The second has cooperation subset $\{c\}$, with queue 2 active and queue 1 passive. When resets occur at queue 1, chains of synchronisations between action types c and b may ensue. A departure (type c) from queue 2 that finds queue 1 empty causes a reset (type b), which causes a second departure from queue 2 (if not empty). With probability p, this will generate a trigger at queue 1 if the reset previously resulted in a non-empty queue (with probability ρ_1) and another reset otherwise. In the latter case, there may be a further departure from queue 2 and so on.

Notice that in the second type of cooperation (queue 2 active), action type a is considered equal to action type b in queue 1. This is not a problem mathematically and will cause us no trouble, but there are complications in the relabelling required by an implementation. This is because in the second cooperation, b would be relabelled to a, giving no actions with type b in the result. To then compute the first cooperation would require the relabelling to be reversed.

Applying RCAT to the reset-queue, this time we obtain:

$$\begin{aligned} x_d &= \rho_1 \mu_1 \\ x_a &= \rho_1(x_c + \nu) \\ x_b &= (1-\rho_1)(x_c + \nu) \\ x_c &= \frac{p\mu_2(x_a + x_d)}{\mu_2 + x_b} \end{aligned}$$

where $\rho_1 = \lambda/\mu_1$, with product-form solution $\pi_{ij} = (1-\rho_1)(1-y)\rho_1^i y^j$ where, considering external departure arcs in dimension 2,

$$y = \frac{(1-p)x_c}{p(1-p)\mu_2} = \frac{x_c}{p\mu_2} = \frac{x_a + x_d}{\mu_2 + x_b}$$

[5] In a cooperation of $m \geq 2$ processes, by the argument of section 2.2, we can choose the path from the reference state $\mathbf{0}$ in either the forward or reversed direction *independently in each dimension*, i.e. $\pi_{\mathbf{j}} = \pi_0 \prod_{k=1}^m \psi_k$ where $\psi_k = \prod_{i=0}^{j-1} \frac{q_{k;i,i+1}}{q'_{k;i+1,i}} = \prod_{i=0}^{j-1} \frac{q'_{k;i,i+1}}{q_{k;i+1,i}}$.

Solving for x_c, we obtain

$$x_c = \frac{p\mu_2\rho_1(\mu_1 + \nu + x_c)}{\mu_2 + (1-\rho_1)(\nu + x_c)}$$

This leads to the following quadratic equation for y:

$$(1-\rho_1)p\mu_2 y^2 + (\mu_2 + (1-\rho_1)\nu - \rho_1 p\mu_2)y - (\lambda + \rho_1\nu) = 0$$

This has one positive and one negative root and the product-form is given by the positive root.

Now suppose that the negative arrivals at queue 2, action type b, are negative triggers that create a negative arrival at queue 1 with probability r when queue 2 is non-empty. With probability $1-r$ a customer at queue 2 (if any) leaves the network. There is no effect if queue 2 is empty, but we could include resets at queue 2, handled in the same way as those at queue 1. Only the equation for x_c needs to be modified, giving:

$$x_c \quad = \quad \frac{(x_a + x_d)(p\mu_2 + r x_b)}{\mu_2 + x_b}$$

where x_a, x_b, x_d, ρ_1 are defined as above. The product-form solution is now: $\pi_{ij} = (1-\rho_1)(1-y)\rho_1^i y^j$ where, again considering external departure arcs in dimension 2,

$$y = \frac{\mu_2(1-p)(x_a + x_d)}{(\mu_2 + x_b)\mu_2(1-p)} = \frac{x_a + x_d}{\mu_2 + x_b}$$

Solving for x_c, we obtain the quadratic:

$$x_c = \frac{(\lambda + \rho_1(\nu + x_c))(\mu_2 p + r(1-\rho_1)(\nu + x_c))}{\mu_2 + (1-\rho_1)(\nu + x_c)}$$

which has exactly one positive root. The value for y then follows via

$$x_c = \frac{\mu_2 p + r\nu(1-\rho_1))y}{1-(1-\rho_1)ry}$$

This is a complicated network and the high relative complexity of the direct solution method can be seen by inspecting the balance equations listed in Appendix C.

Finally, it is interesting to observe that, if a generalised queue does not co-operate actively with another, it does not violate Condition 3. Hence, assuming the other queues satisfy it, there is still a product-form regardless of the choice of $r_{1;ij}$. This is not entirely surprising since such a queue would then be largely detached from the rest of the network, acting as a kind of sink.

3.4.2 Gelenbe and Fourneau's resets

The reset queue of Gelenbe and Fourneau [8] does not allow a reset to result in state 0, i.e. the invisible transition on state zero is excluded, in contrast to our analysis of the previous subsection[6]. Also, they have no negative triggers and their resets cannot cooperate actively. This is a much simpler system to deal with since no chains of cooperations occur with length greater than 3; this happens when a negative service completion generates a (positive) trigger to a third queue. The reset queue 1 is now defined by

$$\begin{aligned} r_{1;i+1,j} &= \delta_{i,j} \quad (i,j \geq 0) \\ r_{1;0j} &= \pi'_j/(1-\pi'_0) \quad\quad (j>0) \end{aligned}$$

where π'_j is the equilibrium probability for state $j \geq 0$. Then we have

$$\pi'_j = (1-\rho'_1)\rho_1'^j$$

where $\rho'_1 = (\lambda_1^+ + \nu')/(\mu_1 + \nu)$ and ν, ν' are the negative arrival rates at positive and zero queue lengths respectively.[7] RCAT's conditions are obviously still satisfied and we obtain

$$\begin{aligned} x_d &= \frac{\lambda + x_c + \nu}{\mu_1 + x_c + \nu}\mu_1 \\ x_a &= \frac{\lambda + x_c + \nu}{\mu_1 + x_c + \nu}(x_c + \nu) \\ x_c &= p(x_a + x_d) \end{aligned}$$

There is no x_b term since resets at queue 1 are not active and so there are no negative arrivals at queue 2. Obviously we could accomodate more general transitions whereby service completions at queue i could yield positive or negative arrivals at the other (or same) queue j $(i,j = 1,2)$. This will be considered further in the next section. We could also specify that, for example, external negative arrivals at queue 1 only have an effect on an empty queue, causing a reset, and that internal negative arrivals from queue 2 (type c) only act on a non-empty queue 1. At the same time, we could consider triggers from queue 1 to cause negative arrivals at queue 2. Then we would obtain:

$$x_d = \frac{\lambda + \nu}{\mu_1 + x_c}\mu_1$$

[6] In Gelenbe and Fourneau's model, a reset coming to node j from node i is 'enabled' with probability β_{ij}, and disabled, leaving the queue empty, with probability $1-\beta_{ij}$. However, an enabled reset always leads to a *positive* queue length. Thus, for example, if $\beta_{ij} = 1$, a reset *always* causes a change of state at node j, whereas in our model queue j remains in the empty state with probability π_0.

[7] In the previous, corresponding reset queue with transitions from state 0 to 0, we obtain $\rho_1 = \lambda_1^+/(\mu_1 + \nu - \nu')$.

$$
\begin{aligned}
x_a &= \frac{\lambda + \nu}{\mu_1 + x_c} x_c \\
x_c &= \frac{p\mu_2}{\mu_2 + x_a} x_d
\end{aligned}
$$

In the Gelenbe-Fourneau networks, there is a problem with introducing active resets. This is because condition 2 of RCAT is not satisfied since there is no incoming action of type b to state 0 in the active queue 1. To include an invisible one would be tantamount to reverting to the reset semantics considered above.

3.5 Generalised G-networks

The reversed process of an arbitrary G-network with positive and negative triggers and generalised resets follows directly from RCAT by induction on the number of nodes in the network. The product-form solution for the network's equilibrium state probabilities then follows directly from equation 1 in section 2.2.

Theorem 1 *An M-node, ergodic, open or closed G-network with service rate μ_i, external positive arrival rate λ_i^+ and external negative arrival rate λ_i^- at node i ($1 \leq i \leq M$), routing probability matrices P^+ and P^-, standard triggering probability matrices Q^+ and Q^- and reset-triggering probability matrices R^+ and R^- for positive and negative customers respectively, has equilibrium probability for state $\mathbf{k} = (k_1, \ldots, k_M)$:*

$$\pi_{\mathbf{k}} \propto y_i^{k_i}$$

where

$$y_i = \frac{v_i^+}{\mu_i + v_i^-} \ , \quad \frac{v_i^+}{\mu_i}$$

for non-reset, reset nodes respectively and the visitation rates at each node i, v_i^+ and v_i^-, are the unique solution of the equations

$$
\begin{aligned}
v_i^+ &= \lambda_i^+ + \sum_j y_j \mu_j p_{ji}^+ + \sum_j y_j v_j^- q_{ji}^+ + \sum_j (1 - y_j) \Delta_j v_j^- r_{ji}^+ \\
v_i^- &= \lambda_i^- + \sum_j y_j \mu_j p_{ji}^- + \sum_j y_j v_j^- q_{ji}^- + \sum_j (1 - y_j) \Delta_j v_j^- r_{ji}^-
\end{aligned}
$$

where $\Delta_j = 1$ if node j has resets and $\Delta_j = 0$ if not.

This theorem is the analogue of [8], but resets to state 0 are allowed and hence chains of resets. To handle precisely that case, with no resets to state 0, we would set the probabilities $r_{ij}^{\pm} = 0$ and define $y_i = \frac{v_i^+ + v_i^-}{\mu_i + v_i^-}$ for reset nodes i.

As written, theorem 1 does not allow for a node j to accept a trigger or reset from a node i with a specified probability α_{ij}, β_{ij} respectively, having no effect with probabilities $1 - \alpha_{ij}, 1 - \beta_{ij}$. However, these are easily accomodated by adding passive invisible actions in the node-j process. For each node i, a passive invisible action (with type the same as that of the passive triggers, a in section 3.4.1) would be added at states $n > 0$ of the node j process, selected with probability $1 - \alpha_{ij}$, and the existing passive trigger action (to state $n - 1$) would be selected with probability α_{ij}. Similarly, passive invisible actions would be added at state 0, selected with probability $1 - \beta_{ij}$, and the existing passive reset actions would be selected with probabilities scaled by a factor β_{ij}.

Notice that it is straightforward to generalise the theorem to certain locally state-dependent service rates (subject to condition 3 of RCAT)[8]. We could generalise G-networks further by allowing the destination of a trigger's target task (positive or negative) to depend on the trigger's sending node as well as the node it arrives at; indeed on its whole path up to the current node. The triggering matrix would then become a 3-dimensional array (in the former case). This would make explicit construction of the reversed process prohibitive but RCAT would still apply and yield separable equilibrium state probabilities.

4 Further product-forms

4.1 A fork-join network

Consider a fork-join network of $M > 1$ parallel servers, where tasks arrive and spawn one sub-task into each of the servers' queues; i.e. arrivals are synchronised. The sub-tasks are processed independently and stored in a buffer associated with their own server. Tasks are then reconstituted at certain instants by an asynchronous process that collects n sub-tasks from each of the M buffers, where n is equal to the minimum buffer occupancy, i.e. at least one buffer is left empty. Such a system is known not to be separable but we use RCAT to find product-form models that match this specification quite closely under certain conditions. We first consider the output buffers and collecting process, and then look at synchronised arrivals. Notice that arrivals are usually not exactly synchronised in practice and the more critical issue is often the synchronisation in sub-task collection.

4.1.1 A synchronised join-buffer

Consider an M-node network with positive arrivals and negative triggers circulating round-robin amongst the nodes. We consider the nodes to represent buffers, as opposed to conventional queues, which are serviced at intervals by the negative triggers only; i.e. there is no conventional service process. We therefore define the following parameters for a G-network model, where $i = 1, \dots, M$:

[8]This is harder than with Jackson's theorem, where departures are not split in a state-dependent way, there being no triggers

- $\mu_i = 0$ for all i;
- $P^+ = P^- = 0$, although this is arbitrary with our choice of μ_i;
- $Q^+ = 0$;
- $q^-_{M1} = q^-_{i,i+1} = u$ for $i = 1, \ldots, M-1$, $q^-_{ij} = 0$ otherwise;

The synchronised join-buffer is then approximated by taking $u = 1$. The only discrepancy is that those queues in the round-robin cycle up to, but excluding, the smallest will be reduced by one too many. Only if one of the queues with the least length is the one that received the external negative arrival to start the cycle will the same (correct) number be removed from every queue. To balance the removals across all the queues, external negative arrivals occur at every queue. For $u < 1$, the removal process is attenuated in the sense that the total number of removals is stochastically bound above by a geometric random variable with parameter u.

With this approximation, we have a product-form from Theorem 1, given by the following equations:

$$\begin{aligned} y_i &= \frac{v_i^+}{v_i^-} \\ v_i^+ &= \lambda_i^+ \\ v_i^- &= \lambda_i^- + y_{i-1} v_{i-1}^- u = \lambda_i^- + v_{i-1}^+ u = \lambda_i^- + \lambda_{i-1}^+ u \end{aligned}$$

where the subscript 0 is synonymous with M. Thus,

$$y_i = \frac{\lambda_i^+}{\lambda_i^- + \lambda_{i-1}^+ u}$$

giving a solution for the equilibrium state probability $\prod_{i=1}^{M}(1 - y_i) y_i^{k_i}$ for state $(k_1, \ldots, k_M)$. Interestingly, this solution does not exist when $\lambda_i^+ \geq \lambda_i^- + \lambda_{i-1}^+ u$ for any i, suggesting a stability condition. Obviously the system is stable if $\lambda_i^+ < \lambda_i^-$ for all i since λ_i^- is equivalent to a 'service rate'. The term $\lambda_{i-1}^+ u$ is the steady state rate at which *negative* arrivals come from the previous queue in the round-robin cycle. This is because, at equilibrium, every positive arrival will eventually be cleared from queue $i - 1$, causing a negative trigger to pass to queue i with probability u.

Even for two queues, the raw balance equations for the equilibrium probabilities are complex, involving unbounded transitions of geometric size. It is a non-trivial task even to verify that a product-form solution exists, with several special boundary cases to check. RCAT has provided a very simple, automated solution to what is a quite complicated problem. It would provide a similar solution if we added service rates at the queues.

4.1.2 Synchronised arrival processes

The fork component of a fork-join network comprises the spawning of sub-tasks into each of M parallel queues on each arrival. Let queue i have constant service rate $\mu_i, 1 \leq i \leq M$, and the external arrival rate be λ. There are no negative customers, triggers or resets, although it would be straightforward to include these. We model this system by M parallel queues with external arrivals at queue 1. The arrival process at queue 2 is passive, cooperating with the active arrivals at queue 1. Similarly, arrivals at queue $i+1$ are passive, cooperating with synchronised arrivals at queues $1, \ldots, i$ ($1 \leq i \leq M-1$).

The PEPA specification of this system, starting arbitrarily in the state with all queues empty, is the following, where $P_i(n)$ denotes queue $i = 1, 2, \ldots, M$ at length $n \geq 0$:

$$S = P_1(0) \underset{\{a\}}{\bowtie} P_2(0) \underset{\{a\}}{\bowtie} \ldots \underset{\{a\}}{\bowtie} P_M(0)$$

where

$$\begin{aligned}
P_1(n) &= (a, \lambda).P_1(n+1) & \\
P_{i+1}(n) &= (a, \top).P_{i+1}(n+1) & (1 \leq i \leq M-1) \\
P_i(n+1) &= (d_i, \mu_i).P_i(n) & (1 \leq i \leq M)
\end{aligned}$$

Notice that action type a is active in the left component of the multiple cooperation, passive in the others.

Conditions 1 and 2 of proposition 1 are not satisfied since state 0 has no incoming instance of action type a in queues $1, \ldots, M-1$. However, we can proceed analogously to our treatment of negative arrivals to an empty queue and introduce invisible actions with type a on the empty states of queues $1, \ldots, M-1$ (active in the case of queue 1, passive for the others). The rate x_a of $\overline{a}$ in queue i is μ_i and so, to secure Condition 3 of RCAT, we set the rate of the active invisible action a in the forward queue 1 to μ_1. This is because invisible actions are self-reversing (have equal forward and reversed rates, by Kolmogorov's criteria [11]). This gives the following new definition of $P_i(n)$:

$$\begin{aligned}
P_1(n) &= (a, \lambda).P_1(n+1) & \\
P_{i+1}(n) &= (a, \top).P_{i+1}(n+1) & (1 \leq i \leq M-1) \\
P_i(n+1) &= (d_i, \mu_i).P_i(n) & (1 \leq i \leq M) \\
P_1(0) &= (a, \mu_1).P_1(0) & \\
P_i(0) &= (a, \top).P_i(0) & (2 \leq i \leq M)
\end{aligned}$$

All the conditions of proposition 1 now hold and RCAT can be applied repeatedly. But the difference between this and our previous use of invisible actions to handle negative arrivals to an empty queue is that here the added forward, invisible actions are *active*. They therefore modify the forward process, which is now not quite what we set out to model! In it, the arrivals are indeed synchronised, but the arrival rate at queue 1 is λ, at queue 2 is λ when queue 1 is non-empty and $\lambda+\mu_1$ when queue 1 is empty, at queue 3 is λ when queues 1 and 2 are both non-empty, $\lambda+\mu_1$ when queue 1 is empty but queue 2 is not, etc. In general, the instantaneous arrival rate to queue i is λ plus the sum of the service rates μ_j of all servers $j < i$ with empty queues. The arrivals at rate λ are properly synchronised, but there are additional arrivals (also synchronised) to each server from lower numbered servers. Whilst not the original problem, it is still instructive to apply RCAT.

For $M = 2$, RCAT gives $x_a = \mu_1$, so the arrival rate in the reversed process of queue 2 is μ_1. The arrival rate in the reversed process of queue 1 is λ. The product-form solution is then obtained from equation 1 in section 2.2 by taking a path in the reversed process from the reference state $(0, 0)$, along the queue 1 axis (following arrivals in the reversed process) to $(i, 0)$ and similarly up the queue 2 axis to (i, j), retracing the path backwards to $(0, 0)$ in the forward process, following service completions. The forward service rates are μ_1 and μ_2; the corresponding reversed arrival rates are λ and μ_1, as per above. The product-form is therefore

$$\pi_{ij} \propto \left(\frac{\lambda}{\mu_1}\right)^i \left(\frac{\mu_1}{\mu_2}\right)^j$$

In the reversed M-queue process, the arrival rate at queue i (rate on the reverse of the departure arc in the forward process) is μ_{i-1} for $2 \le i \le M$ and λ for $i = 1$. We prove this by induction on M; for $M = 2$, we proved it above. By proposition 1, we can apply RCAT to the cooperation between the $(m-1)$-queue process, comprising the set of (cooperating) queues 1 to $m-1$, and queue m, $3 \le m \le M$. In the reversed $(m-1)$-queue process, the reverse arcs of the (forward) arrivals at queue $m-1$ have rate μ_{m-1}, because queue $m-1$, which cooperates with an $(m-2)$-queue process, is reversible. Thus, in the cooperation with queue m, we set $x_a = \mu_{m-1}$, which then implies that the arrival rate to reversed queue m is μ_{m-1}, again by reversibility. The arrival rates to queues $1, \ldots, m-1$ in the m-queue reversed process are unchanged from the $(m-1)$-queue reversed process and the claim is proved.

In the forward process, the service rate of each queue j is μ_j and hence we obtain the product-form

$$\pi_{\mathbf{k}} \propto \left(\frac{\lambda}{\mu_1}\right)^{k_1} \prod_{i=2}^{M} \left(\frac{\mu_{i-1}}{\mu_i}\right)^{k_i}$$

by following a path along each dimension $1, \ldots, M$ successively (as above for $M = 2$) from reference state $(0, \ldots, 0)$ to general state $(k_1, \ldots, k_M)$ in the reversed process, and back in the forward process.

For our modified process to have a steady state, the service rates of the queues must increase strictly monotonically with their index. The addition of invisible transitions, necessary to secure a product-form, essentially implies that each queue feeds the next to its full capacity. Suppose we choose $\mu_i = \mu_{i-1}/\rho_i$ for $2 \leq i \leq M$, where $\rho_i = \lambda/\mu_i < 1$. Then we obtain

$$\pi_{\mathbf{k}} \propto \prod_{i=1}^{M} \rho_i^{k_i}$$

This is the product-form solution we would obtain if the M queues received *independent* Poisson arrivals at rate λ. However, to achieve this, with the extra traffic generated by the invisible actions, we had to increase the service rates at all but one of the queues. Nevertheless, the result provides a useful bounding approximation. Supposing that $\mu_i = \mu$ for all i, at high loads, $\lambda \approx \mu$, queues will rarely be idle and so the invisible transitions will have little effect. The system then approaches a separable, quasi-independent set of queues. This result provides some quantitative explanation for this plausible property.

4.1.3 Chains of positive triggers

The synchronisation of arrivals developed above can equally well be applied to positive arrivals in general queueing networks, giving the positive trigger analogue to the multiple negative triggers considered in section 3.2. Positive triggers were considered in [3], where also an exact product-form could not be found. However, a product-form bound was given on the joint (complementary) cumulative queue length distribution by comparing the network with one that had extra transitions, which satisfied certain quasi-reversibility constraints. The proof of this bound is by a routine sample path analysis and was omitted.

Here, we have a corresponding situation. In order to obtain a product-form solution, we introduced (active) invisible transitions at empty queue states of the source nodes, i say, with rate equal to the reversed rate of the arrival action, i.e. μ_i. These create additional arrivals at the synchronising (passive) nodes, with rate $(1 - \rho_i)\mu_i = \mu_i - \lambda$ and give a product-form solution when equilibrium is preserved. Comparing sample paths in the original network, with state-vector random variable $\vec{E}$, and in the one modified with the invisible transitions, with state-vector random variable $\vec{M}$, we have, componentwise, $\vec{E} \leq \vec{M}$ so that $P(\vec{E} \geq \mathbf{x}) \leq P(\vec{M} \geq \mathbf{x})$. Hence, as in [3], the exact stationary distribution of $\vec{E}$ is stochastically dominated by the (product-form) distribution of $\vec{M}$, i.e.

$$P(\vec{E} \geq \mathbf{x}) \leq \prod_{i=1}^{M} \rho_j^{x_j}$$

4.2 A queueing network with batches

Separable queueing networks with batch arrivals are more problematic. To see this in the context of RCAT, consider the reversed process of an $M^B/M/1$ queue, where M^B denotes a Poisson point process of batch arrivals with batch-size probability mass function having generating function $B(z)$. The equilibrium queue length probability generating function (pgf) of this queue, with (batch) arrival rate λ and service rate μ, is, (see, e.g. [15]):

$$\Pi(z) = \frac{\pi_0(1-z)}{\rho z(B(z)-1)+1-z}$$

where $\rho = \lambda/\mu$. Suppose that batches are geometric, having size $n \geq 1$ with probability $b_n = (1-\alpha)\alpha^{n-1}$. Then

$$B(z) = \frac{(1-\alpha)z}{1-\alpha z}$$

and so, after some simplification,

$$\Pi(z) = \frac{(1-\alpha z)\pi_0}{1-(\alpha+\rho)z}$$

from which we find (the coefficient of z^n), for $n \geq 1$,

$$\pi_n = \pi_0\rho(\alpha+\rho)^{n-1} \tag{4}$$

In the reversed process, the reversed rates of the service completions, $q'_{i,i+1}$ say, are:

$$\begin{aligned} q'_{i,i+1} &= \frac{\pi_{i+1}}{\pi_i}\mu = (\alpha+\rho)\mu = \lambda+\alpha\mu \quad (i>0) \\ q'_{0,1} &= \frac{\pi_0\rho}{\pi_0}\mu \ = \lambda \end{aligned}$$

Thus, in a cooperation where this queue's departure actions, a say, synchronise with passive 'arrival' actions in another process (e.g. a queue), RCAT's condition 3 is not satisfied since the reversed rate of the active (departure) action a is not constant. Consequently, if a network of queues contains a node with batch arrivals such as this, RCAT cannot find a reversed process. Direct analysis of the balance equations confirms that a tandem network comprising this $M^B/M/1$ queue and an $M/M/1$ queue does not have a product-form.

However, here it is only in state 0 of the reversed process that $\overline{a}$ has a different rate from its rates in all other states. This suggests that we consider a modified $M^B/M/1$ queue with (point) arrival rate λ_0 in state 0 and all other parameters unchanged. At equilibrium, this queue has balance equations

$$\begin{aligned} \pi_0\lambda_0 &= \pi_1\mu \\ \pi_i(\lambda+\mu) &= \pi_{i+1}\mu + \sum_{j=1}^{i-1}\pi_j\lambda b_{i-j} + \pi_0\lambda_0 b_i \quad (i \geq 1) \end{aligned}$$

Multiplying the equation for π_i by z^i and summing from $i = 0$ to ∞ yields the following queue length pgf (after some algebra):

$$\Pi(z) = \frac{\pi_0(1 + (\rho - \rho_0)zB(z) - (1 + \rho - \rho_0)z)}{\rho z(B(z) - 1) + 1 - z}$$

where $\rho_0 = \lambda_0/\mu$, which simplifies (for the geometric batches) to:

$$\Pi(z) = \frac{(1 - (\rho - \rho_0 + \alpha)z)\pi_0}{1 - (\alpha + \rho)z}$$

The coefficient of z^n is now $\pi_n = \pi_0\rho_0(\alpha + \rho)^{n-1}$ for $n \geq 1$ and we obtain the reversed rates:

$$\begin{aligned} q'_{i,i+1} &= (\alpha + \rho)\mu = \lambda + \alpha\mu \quad (i > 0) \\ q'_{0,1} &= \lambda_0 \end{aligned}$$

(Notice that a simple argument using Kolmogorov's citeria shows that changing just λ_0 will have no effect on the reversed rates amongst states $i > 0$. This is essentially because the original reversed rates will still preserve the total outgoing rate from each state and no minimal cycle includes the changed rate λ_0.)

We therefore have the following result.

Proposition 2 *The $M^B/M/1$ queue with geometrically batched arrivals, defined above, preserves the product-form in a network if it receives an additional external arrival stream when empty with the same batch size distribution and rate equal to $\alpha\mu$, the product of its service rate and the batch size parameter.*

The proof is by direct appeal to RCAT.

In fact a more general result holds which we deal with in section 4.4: batch sizes do not have to be geometric, provided those arriving at empty and non-empty queues can have different probability mass functions, and service completions can also cause batch departures.

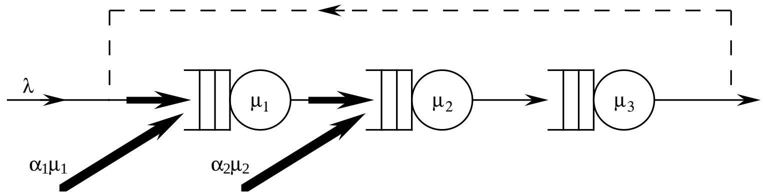

Figure 4: Network with batch arrivals

Consider, for example, the network with geometric batch arrivals shown in fig. 4, where the external arrivals with rates $\alpha_i\mu_i \quad (i = 1, 2)$ only exist when queue i is empty (is in state 0) and the arrivals with rate λ to queue 1 occur in all states. Let the synchronising departure action types from queues 1, 2, 3 be a, b, c respectively. First, omitting the feedback (type c) represented by the dotted lines, in RCAT terminology, $x_a = \lambda + \alpha_1\mu_1$ and $x_b = x_a + \alpha_2\mu_2$. This gives the product-form for state (i, j, k):

$$\pi_{ijk} = \left(\frac{\lambda + \alpha_1\mu_1}{\mu_1}\right)^i \left(\frac{\lambda + \alpha_1\mu_1 + \alpha_2\mu_2}{\mu_2}\right)^j \left(\frac{\lambda + \alpha_1\mu_1 + \alpha_2\mu_2}{\mu_3}\right)^k$$

If we include the feedback, selected by departures from node 3 with probabilty p, we obtain instead $x_a = \lambda + x_c + \alpha_1\mu_1$, $x_b = x_a + \alpha_2\mu_2$ and $x_c = px_b$, so that $x_a = (\lambda + p\alpha_2\mu_2 + \alpha_1\mu_1)/(1 - p)$, giving x_b, x_c and product form $\pi_{ijk} = (x_a/\mu_1)^i \, (x_b/\mu_2)^j \, (x_b/\mu_3)^k$.

Notice that a batched arrival queue without the additional external arrival stream does not violate RCAT (specifically Condition 3) if it does not synchronise *actively*, whereupon a product-form for the whole network still exists, equation 4 giving the factor associated with this particular queue. For example, the network shown in fig. 5 has (unnormalised) equilibrium probability for state (i, j):

$$\pi_{ij} = \left(\frac{x_a}{\mu_1}\right)^i \left(\frac{x_a + \alpha_2\mu_2}{\mu_2}\right)^j \quad (i \geq 0, j > 0)$$

$$\pi_{i0} = \left(\frac{x_a}{\mu_1}\right)^i \left(\frac{x_a + \alpha_2\mu_2}{x_a}\right) \quad (i \geq 0)$$

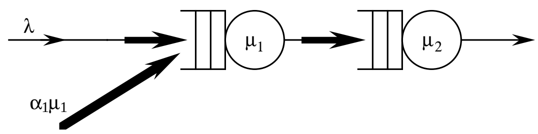

Figure 5: Tandem network with batch arrivals and unmodified queue

4.3 Batch removals

A single G-queue in which negative arrivals remove a *batch* of customers also has a geometric queue length probability distribution at equilibrium, for any batch size probability mass function. For a more complete treatment of this result, the reader is referred to [7]; below we just state it and apply it in networks with batch transfers. The solution is given by that of the $M/M^C/1$ queue, with (positive) arrival rate λ, service rate μ and batch size probability mass function c_i $(i \geq 1)$ with generating function $C(z)$. Service completions at queue length j remove the whole queue with probability $\sum_{i=j}^{\infty} c_i$, i.e. batch sizes greater than the queue length result in an empty queue. It can be shown that, when equilibrium exists, the queue has length n with probabililty $(1-\xi)\xi^n$ where ξ is the solution of the non-linear equation

$$\lambda = \xi[\lambda + \mu(1 - C(\xi))]$$

with $0 < \xi < 1$.

Such a G-queue with passive batch removals out of the network (either due to triggers or service completions) therefore preserves the separable solution in a G-network since the reversed rates of its active cooperating actions (*not* batch removals) will be constant, by virtue of the geometric equilibrium queue length distribution, validating condition 3. The other conditions of RCAT are the same as with single removals. This is the result of [7]. However, if the removals are transfers to another queue, as opposed to out of the network, the removal actions are *active* in the cooperation with the said other queue. The rate of the reversed action of an action type a_j that removes j customers is easily seen to be: $q'_{n,n+j} = \mu c_j \xi^j$ at queue length $n > 0$ and $q'_{0,j} = \mu C_j \xi^j$ where we define $C_j = \sum_{k=j}^{\infty} c_k$. These are not the same and so condition 3 of RCAT does not hold, although the other two do for all actions a_j.

We can secure condition 3 by allocating action type a_j out of each state $j > 0$ only to batch sizes *equal* to j. Batches of size greater than j will also lead to state 0 but not participate in the cooperation with another queue; they depart the network. This is the 'assembly-transfer network' of Chapter 8 of [3], which is given the interpretation that the cooperating actions a_j are 'full batches' and the others are 'partial batches', which are discarded in an assembly line. RCAT then holds in a cooperation with any node that has a geometric equilibrium queue length probability distribution when it receives batch arrivals with probability mass function $b_j = \mu c_j \xi^j$ $(j \geq 1)$ at all queue lengths. Such nodes are considered in the next subsection.

In chapter 7 of [3], it is shown that if all batch transfers cause a *unit* arrival in the cooperating queue, a product-form exists. In our analysis, this follows since, applying RCAT, the cooperating queue receives net arrivals at constant rate $\sum_{j=1}^{\infty} x_{a_j} = \sum_{j=1}^{\infty} \mu c_j \xi^j$ in all states and so has the usual $M/M/1$ factor in the product-form.

4.4 Arbitrary batch input and output

Consider now a more general queue with batches. Suppose one arrival stream has rate λ and batch size with pgf $B(z)$ at all queue lengths $n \geq 0$. In addition, suppose there is another arrival stream to an empty queue only, $n = 0$, with rate λ_0 and batch size pgf $B_0(z)$. Let the service rate be μ in all non-empty states, with departures having batch size pgf $C(z)$.

Proposition 3 *The above queue with batches has geometrically distributed equilibrium queue length probabilities with parameter $x < 1$, $\pi_n = (1-x)x^n$ for $n \geq 0$,* *if*

$$\lambda_0 = \frac{\lambda(1-\eta(x))}{\eta(x)(1-x)} \tag{5}$$

$$B_0(z) = \left[\frac{(1-x)z}{1-xz}\right]\left[\frac{1-\eta(x)B(z)/z}{1-\eta(x)}\right] \tag{6}$$

where

$$\eta(x) = \frac{\lambda}{[\lambda+\mu(1-C(x))]x} < 1$$

.

Proof

At equilibrium, the queue has balance equations

$$(\lambda+\mu)\pi_i = \lambda\sum_{j=1}^{i} b_j\pi_{i-j} + \lambda_0\pi_0 b_{0i} + \mu\sum_{j=1}^{\infty} c_j\pi_{i+j} \qquad (i \geq 1) \tag{7}$$

$$(\lambda+\lambda_0)\pi_0 = \mu\sum_{j=1}^{\infty} C_j\pi_j \tag{8}$$

Multiplying equation 7 by z^i and summing from $i = 1$ to ∞ leads to the following equation for the pgf of the queue length, $\Pi(z)$:

$$(\lambda+\mu)(\Pi(z)-\pi_0) = \lambda B(z)\Pi(z) + \lambda_0\pi_0 B_0(z) + \mu\sum_{i=1}^{\infty}\sum_{j=1}^{\infty} c_j\pi_{i+j}z^i$$

For the required geometric queue length distribution, $\Pi(z) = (1-x)/(1-xz)$, $\pi_0 = 1-x$. Then we obtain:

$$[\lambda+\mu(1-C(x))]xz = \lambda B(z) + \lambda_0 B_0(z)(1-xz) \tag{9}$$

At $z = 1$, this yields

$$\mu x(1-C(x)) = (\lambda+\lambda_0)(1-x)$$

and so

$$\lambda_0(1-x) = \lambda/\eta(x) - \lambda$$

proving equation 5. (In fact this also follows from the redundant equation 8.) Since $\lambda > 0$, $\eta(x) < 1$ for all $x < 1$.

Using equation 5 in equation 9 now gives (omitting the argument (x) from $\eta(x)$ for brevity):

$$\begin{aligned}\lambda(1-\eta)(1-xz)B_0(z) &= (1-x)\eta[z\lambda/\eta - \lambda B(z)] \\ &= (1-x)z\lambda[1-\eta B(z)/z] \qquad \spadesuit\end{aligned}$$

Given valid x, the arrival rate of the batch arrivals to an empty queue, λ_0, and the batch size pgf, $B_0(z)$, are determined uniquely in terms of the queue's parameters $\lambda, \mu, B(z), C(z)$. The value of x is not unique, there being a different λ_0, B_0 for each valid value. It is interesting to note that the resets of section 3.4.2 are a special case of this observation; take $B(z) = C(z) = z$ giving $\lambda_0 = x\mu - \lambda, B_0(z) = (1-x)z/(1-xz)$. Reset-jumps *must* be geometric to keep a product-form. We have the following constraint on the range of valid x.

Proposition 4 $\lambda(B(x^{-1}) - 1) \leq \mu(1 - C(x))$ *when* $B(x^{-1})$ *exists.*

Proof

For $i \geq 1$, the coefficient of z^i in the series expansion of $B_0(z)$ is

$$\begin{aligned} b_{0i} &= \frac{1-x}{1-\eta}\left[x^{i-1} - \eta\sum_{j=1}^{i} b_j x^{i-j}\right] \\ &= \frac{(1-x)x^i}{1-\eta}\left[x^{-1} - \eta\sum_{j=1}^{i} b_j x^{-j}\right] \end{aligned}$$

For b_{0i} to be non-negative for all $i \geq 1$, we require $x^{-1} \geq \eta\sum_{j=1}^{\infty} b_j x^{-j}$ and the summation to exist. Hence we require $B(x^{-1}) \leq 1/(\eta x) = 1 + \mu(1 - C(x))/\lambda$. ♠

Comments

1. $B(x^{-1})$ does not exist for all pgfs B and all $x < 1$. For example, if B is the pgf of a geometric random variable with parameter α, i.e. $B(z) = (1-\alpha)/(1-\alpha z)$, $B(x^{-1})$ does not exist for $x \leq \alpha$.
2. $B(x^{-1})$ does exist for all discrete random variables defined on a finite sample space and, for some values of $x < 1$, for those defined on an infinite sample space with continuous pgfs. In the above geometric case, for example, $B(x^{-1})$ exists for $x > \alpha$.

Proposition 5 $x_0 < x < 1$ *where* x_0 *is the unique solution of the equation* $\lambda(B(x^{-1}) - 1) = \mu(1 - C(x))$ *for continuous pgfs* B *and* C, *when* $B(x^{-1})$ *exists.*

Proof

Define the function $F(x) = \lambda(B(x^{-1}) - 1) - \mu(1 - C(x))$, so we are seeking solutions of $F(x) = 0$. One solution is $x = 1$ since $B(1) = C(1) = 1$.

As positive $x \to 0$, $F(x) \to +\infty$ since $B(y) \to \infty$ as $y \to \infty$. The derivative $F'(x) = -\lambda B'(x^{-1})x^{-2} + \mu C'(x)$, and so $F'(1) = -\lambda m_B + \mu m_C$, where m_B, m_C are the mean arrival and service batch sizes respectively. Thus, for equilibrium to exist, $F'(1) > 0$ and so a solution to $F(x) = 0$ exists in $(0, 1)$ by continuity.

But $F''(x) = \lambda B''(x^{-1})x^{-4} + 2\lambda B'(x^{-1})x^{-3} + \mu C''(x) > 0$ since the derivatives of pgfs are non-negative at all non-negative arguments. Hence the solution x_0 is unique.

Finally, since $F'(x_0)$ must be negative, $F(x) < 0$ for $x_0 < x < 1$, satisfying proposition 4. ♠

We now return to our treatment of geometric batch sizes in section 4.2 and show that if the extra arrival batch sizes in state 0 are to be distributed as the given arrival batch sizes at all queue lengths, both must be geometric.

Proposition 6 *$B_0 = B$ implies both B_0 and B are the pgfs of geometric random variables with parameter $x - \lambda(1-x)/(\mu(1-C(x))$.*

Proof

When $B_0 = B$, equation 6 gives

$$(1-\eta)(1-xz)B(z) = (1-x)z - (1-x)\eta B(z)$$

so that

$$B(z) = \frac{(1-x)z}{(1-\eta)(1-xz) + (1-x)\eta} = \frac{(1-x)z}{(1-\eta x) - (1-\eta)xz} = \frac{\alpha z}{1-\alpha z}$$

where $\alpha = \frac{(1-\eta)x}{1-\eta x} = x - \frac{1-x}{(\eta x)^{-1}-1}$ and the result follows. ♠

The equation for α determines x and hence λ_0, so the arrival rate of the additional batches is also determined. Notice that when departures are not batched, we have $C(x) = x$ and $\alpha = x - \lambda/\mu$, i.e. $x = \alpha + \rho$ in the product-form solution, as in section 4.2. Then $\eta(x) = \frac{\rho}{(\alpha+\rho)(1-\alpha)}$ so that $\lambda_0 = \alpha\mu$ by equation 5.

Having estalished these geometric equilibrium queue length probability mass functions for single queues with both input and output batches, RCAT can be applied and, for each network specification, will supply a product-form solution together with the required batch arrival processes to empty queues. These depend on the internal arrival rates, which are given by the reversed rates x_a of RCAT. We call a node in a network with internal or external batched arrivals a *batch-node.*

Theorem 2 *A G-network of the kind defined in theorem 1, with batch transfers, has product-form joint queue length probabilities at equilibrium given by RCAT, assuming a solution exists to the equations for the rates x_a, when additional external batched arrivals are introduced to empty queues of batch-nodes. The rate and batch size pgf of each additional arrival stream is given by proposition 3, such that each batch-node, considered in isolation, has geometrically distributed queue length at equilibrium, with correspondingly given parameter.*

Proof

Condition 3 of RCAT holds since each reversed rate x_a is constant over instances of a by construction. The other conditions are satisfied since all the passive actions are arrivals in both the forward and reversed processes, using invisible actions at empty queues to ensure negative arrivals have no effect on a non-reset queue. Finally, a solution exists for the quantities x_a by hypothesis; the reversed process can therefore be computed and hence the product-form. ♠

The existence of solutions for the x_a is not considered here, but typically this can be established by Brouwer's theorem, as in [10], for example. It is an open question to establish the existence of solutions in general for RCAT.

By proposition 1, chains of these queues can be linked by negative triggers so that we can model multiple batch transfers in a queueing network without further effort using this approach in a mechanised way – ultimately by computer. Chapter 8 of [3] gives explicit product-forms for many such networks and the required definitions of parameters. As in section 4.1.3, the product-form obtained gives a stochastic bound on the exact equilibrium state probabilities since the extra arrivals cannot decrease queue lengths on sample paths. Consequently it makes sense to choose x so as to give the least additional per-customer arrival rate to empty queues in order to get the tightest bound.

5 Conclusion

The use of Markovian Process Algebra and the Reversed Compound Agent Theorem of [11] introduces a new, compositional methodology for deriving the equilibrium state probabilities in separable Markov processes. This approach does not require balance equations to be solved but instead determines the reversed process whence a simple, separable solution ensues. The origins of RCAT and the methodology based on it lie in a combination of MPA and the theory of reversed stationary Markov processes. In this paper, the domain of application of RCAT has been extended to account for sequences of process-transitions caused by a single action.

The RCAT-based methodology derives many product-forms, known and possibly unknown, in a uniform way. This is exemplified here by the derivation, from the same theorem, of the product-form solutions for interacting systems that include G-networks, with triggers and generalised resets, as well as networks with particular kinds of batch transfers. Indeed, prior to the advent of G-networks around 1989 [5], many believed that partial balance was a necessary condition for a product-form. Significantly, there is no difference in the RCAT approach: negative customers satisfy the same conditions (with respect to different action types) as do positive ones.

In fact other well-known product-forms, e.g. in 'Boucherie' networks with competition for resources, also follow immediately from a generalisation of RCAT [13]. Specifically, this generalisation relaxes the first two conditions relating to the enablement of synchronising actions. Interestingly, there have been several occasions when the author has applied RCAT to a stochastic network and found a significant product-form he believed to be new, only to discover subsequently it had been derived already by a more direct method! Whether or not the one for generalised resets, for example, is new, the routine way in which they are all derived here demonstrates the power of the methodology.

Although Jackson's theorem [19] follows immediately from RCAT, as shown in [11], the BCMP theorem [1] is more complicated, involving both other queueing disciplines than FCFS and multiple classes. It is shown how to derive the single class version of the BCMP theorem [1] using the extended RCAT in [13]. The crux of this work is the representation of processor sharing (PS) and last-come-first-served (LCFS) queues with Coxian service times and finding their individual product-form solutions using RCAT. Multiple class networks are a straightforward extension since each class has its own synchronising action types, which separately satisfy the conditions of RCAT. Note that FCFS nodes do not thus admit multiple classes in a product-form solution, a fact long established. This is unsurprising in the RCAT approach since it would involve a complex model of a single FCFS multi-class queue, the class of any task in service being crucial to the analysis. We therefore expect to be able to use the methodology presented here to derive, analogously to the method of [13] for a single class, the BCMP result for multi-class queueing networks as well as corresponding results for multi-class G-networks [4, 9]. In fact for G-networks, the cited works make certain restrictions to make the analysis tractable and it may be possible to generalise using the simpler compositional approach.

The principal advantage of the RCAT-based methodology is its potential for mechanisation and symbolic implementation; this comes from the compositional approach. By incorporating it into a suitable support environment – possibly, but not necessarily, for process algebras – the derivation of many product-form theorems could be automated and new ones derived in a unified stochastic modelling framework.

Appendix A: Reversed Compound Agent Theorem (RCAT)

Let the subset of action types in a cooperation set L which are *passive* with respect to a process P be denoted by $\mathcal{P}_P(L)$ and the subset of corresponding active action types by $\mathcal{A}_P(L) = L \setminus \mathcal{P}_P(L)$. Assuming that the set of outgoing passive, and set of incoming active, synchronising actions in any state of each component contains at most one of each type in L, the Reversed Compound Agent Theorem of [11] states the following:

Theorem 3 *(Reversed Compound Agent)*
Suppose that the cooperation $P \underset{L}{\bowtie} Q$ *has a derivation graph with an irreducible subgraph* G*. Given that*

1. *every passive action type in* L *is always enabled (i.e. enabled in all states of the transition graph);*
2. *every reversed action of an active action type in* L *is always enabled;*
3. *every occurrence of a reversed action of an active action type in* $\mathcal{A}_P(L)$ *(respectively* $\mathcal{A}_Q(L)$*) has the same rate in* $\overline{P}$ *(respectively* $\overline{Q}$*).*

the reversed agent $\overline{P \underset{L}{\bowtie} Q}$*, with derivation graph containing the reversed subgraph* $\overline{G}$*, is*

$$\overline{R}\{(\overline{a}, \overline{p_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_P(L)\} \underset{\overline{L}}{\bowtie} \overline{S}\{(\overline{a}, \overline{q_a}) \leftarrow (\overline{a}, \top) \mid a \in \mathcal{A}_Q(L)\}$$

where

$$\begin{aligned} R &= P\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_P(L)\} \\ S &= Q\{\top_a \leftarrow x_a \mid a \in \mathcal{P}_Q(L)\} \end{aligned}$$

$\{x_a\}$ *are the solutions (for* $\{\top_a\}$*) of the equations*

$$\top_a = \overline{q_a} \qquad a \in \mathcal{P}_P(L)$$

$$\top_a = \overline{p_a} \qquad a \in \mathcal{P}_Q(L)$$

and $\overline{p_a}$ *(respectively* $\overline{q_a}$*) is the symbolic rate of action type* $\overline{a}$ *in* $\overline{P}$ *(respectively* $\overline{Q}$*).*

Appendix B: Balance equations in section 3.4.1

The balance equations for the equilibrium probabilities π_{ij} of the network of section 3.4.1 are as follows:

$$(\lambda + \nu + \mu_1 I_{i>0} + \mu_2 I_{j>0})\pi_{ij} =$$

$$\begin{aligned} &\lambda \pi_{i-1,j} I_{i>0} + (\mu_1 + \nu)\pi_{i+1,j-1} I_{j>0} + \mu_2(1-p)\pi_{i,j+1} + \mu_2 p \pi_{i+1,j} I_{j>0} \\ &+\nu \pi_{i\bullet}(1-r) \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+1+k} r^k + \nu \pi_{i+1,\bullet} r I_{j>0} \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+k} r^k \\ &+\mu_2 p \pi_{i\bullet}(1-r) \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+2+k} r^k + \mu_2 p \pi_{i+1,\bullet} r I_{j>0} \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+1+k} r^k \\ &+\nu \pi_{i\bullet}(1 - I_{j>0}) \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+k} r^k + \mu_2 p \pi_{i\bullet}(1 - I_{j>0}) \sum_{k=0}^{\infty} \pi_{0\bullet}^k \pi_{0,j+1+k} r^k \end{aligned}$$

where $\pi_{i\bullet}$ is the marginal equilibrium probability that the length of queue 1 is $i \geq 0$, $\sum_{j=0}^{\infty} \pi_{ij}$, and I is the indicator function.

References

[1] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal ACM*, 22(2):248-260, 1975.

[2] R.J. Boucherie. A Characterisation of independence for competing Markov chains with applications to stochastic Petri nets. *IEEE Transactions on Software Engineering*, 20(7):536–544, July 1994.

[3] X. Chao, M. Miyazawa and M. Pinedo. *Queueing networks: customers, signals and product form solutions*. Wiley, 1999

[4] J.-M. Fourneau, E. Gelenbe and R. Suros. G-networks with multiple classes of positive and negative customers. *Theoretical Computer Science*, 155:141–156, 1996.

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

[6] E. Gelenbe. G-networks with triggered customer movement. *Journal of Applied Probability*, 30:742–748, 1993.

[7] E. Gelenbe. G-networks with signals and batch removal. *Probability in the Engineering and Informational Sciences*, 7:335–342, 1993.

[8] E. Gelenbe and J.-M. Fourneau. G-networks with resets. In *Proceedings of PERFORMANCE '02*, Rome, 2002, *Performance Evaluation*, 49:179–191, 2002.

[9] E. Gelenbe and A. Labed. G-networks with multiple classes of signals and positive customers. *European Journal of Operations Research*, 108(2):293–305, 1998.

[10] E. Gelenbe and R. Schassberger. Stability of product form G-networks. *Probability in the Engineering and Informational Sciences*, 6:271–276, 1992.

[11] P.G. Harrison. Turning back time in Markovian process algebra. *Theoretical Computer Science*, 290:1947–1986, 2003.

[12] P.G. Harrison. Mechanical solution of G-networks via Markovian Process Algebra. In *Proceedings of the International Conference on Stochastic Modelling and the IV International Workshop on Retrial Queues*, Cochin, India, December 2002, *Notable Publications*, 2002.

[13] P.G. Harrison. Reversed processses, product forms, non-product forms and a new proof of the BCMP theorem. In *Proceedings of Numerical Solutions of Markov Chains*, University of Illinois at Urbana-Champaign, September 2003.

[14] P.G. Harrison. Compositional reversed Markov processes, with applications to G-networks. Submitted to *Performance Evaluation Journal*, 2003.

[15] P.G. Harrison and N.M. Patel. *Performance modelling of communication networks and computer architectures.* Addison-Wesley, 1992.

[16] J. Hillston. *A Compositional approach to performance modelling.* PhD thesis, University of Edinburgh, 1994.

[17] J. Hillston and N. Thomas. A syntactic analysis of reversible PEPA models. In *Proceedings of PAPM 1998*, Nice, July 1998.

[18] J. Hillston and N. Thomas. Product form solution for a class of PEPA models. *Performance Evaluation*, 35:171–192, 1999.

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

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