Multiple Class G-Networks with Restart

Jean-Michel Fourneau
PRiSM, CNRS / Université de Versailles St Quentin
45, avenue des Etats-Unis
F-78035 Versailles
jean-michel.fourneau@prism.uvsq.fr

Katinka Wolter
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin, Germany
katinka.wolter@fu-berlin.de

Philipp Reinecke
HP Labs Bristol
Long Down Avenue, Bristol
BS34 8QZ
philipp.reinecke@hp.com

Tilman Krauß
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin
tilman.krauss@fu-berlin.de

Alexandra Danilkina
Institute of Computer Science
Freie Universität Berlin
Takustr. 9, 14195 Berlin
alexandra.danilkina@fu-berlin.de

ABSTRACT

Restart is a common technique for improving response-times in complex systems where the causes of delays can either not be discerned, or not be addressed by the user. With restart, the user aborts a running job that exceeds a deadline, and resubmits it to the system immediately. In many common scenarios, this approach can reduce the response-times that the user experiences. Restart has been well-studied for scenarios where only one user applies restart, and typically in cases where queueing effects can be neglected. In this paper we approach the question of restart in a scenario where restart is applied by many users in a system that can be modelled as an open queueing network. We apply the G-Networks formalism to this problem. We use negative customers to model the abortion and retry of a request. The open G-network uses multiple classes with phase-type distributed service times. This allows the approximation of a preemptive repeat different behaviour as it is natural for multiple restarts of a request. We compute the response time of a request and show that an optimal restart interval can be found. The results are compared with simulation.

Categories and Subject Descriptors

Networks [**Network performance evaluation**]: Network performance modeling; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Keywords

Restart, G-Networks, Phase-type distributions

ICPE'13, April 21–24, 2013, Prague, Czech Republic.

1. INTRODUCTION

The restart mechanism is known to most people, since there exist many situations in daily life where the impatient customer after some waiting time refuses to wait any longer for completion of his job, cancels the job and restarts it. Internet downloads are the most widely known situation that can benefit from restarts, but many other instances do exist. While restart is often an elegant and simple solution, it may also have a negative effect on the system to which it is applied, as restart can effectively mean an increase of the load on the system, thereby exacerbating the problem it should address. Therefore, a careful choice of the restart interval is often necessary.

Several aspects of the restart problem have been explored in recent years. In [18, 19] a stochastic model for restart to minimise job completion times has been proposed. The probability of job completion under restart has been maximised in [20]. In these works the authors considered an individual user issuing independent jobs that are completed according to some completion-time distribution. It could be shown that with this restriction restart can be successful in reducing the completion-times experienced by the user.

In this paper we address the question whether restart is still beneficial if applied by several users on one or more shared resources. This naturally leads to the formulation of the problem as an open queueing-network model. We utilise the formalism of G-networks, that is, queueing networks with signals. Signals in our G-network model restarts as they remove a random job in the queue. The restarted job returns to the queueing network in a different class which allows us to model a different processing speed upon restart. Our model uses phase-type (PH) distributions for the service-time distributions [14], in order to be able to reflect characteristics of real systems. In particular, the distributions are more general than the exponential distribution.

The remainder of this paper is structured as follows: In Section 2 we provide the necessary background on G-networks. In Section 3 we formulate our model. Section 4 provides a product-form solution for this model. In Section 5 we discuss computation of measures of interest using our solution. We then illustrate the approach in Section 6 and discuss its

application in the evaluation of practical questions related to restart in Section 7.

2. G-NETWORKS

In this paper we define a class of G-networks that can model restart. The theory of queues with signals has received considerable attention since the seminal paper on positive and negative customers [6] published by Gelenbe 20 years ago. Traditional queueing networks model systems that are used to represent contention among customers for some resources. Customers move from server to server, they wait for service, but they do not interact among each other. Signals have been used to change these rules. In a network of queues with signals (also denoted as a G-network of queues) customers are allowed to change to signals at the completion of their service and signals interact at their arrival into a queue with customers already present in the queue. Moreover signals are never queued. They try to interact with customers and disappear immediately. Note that they may fail to interact with some probability or due to some conditions which are not satisfied. Despite this deep modification of the model, G-networks still preserve the product-form property for the steady-state distribution of some Markovian queueing networks.

The first type of signal [6] was introduced as a negative customer. A negative customer deletes a positive customer at its arrival at a backlogged queue. A negative customer is never queued. Positive customers are usual customers which are queued and receive service or are deleted by negative customers. Under typical assumptions for Markovian queueing networks (Poisson arrival for both types of customers, exponential service time for positive customers, Markovian routing of customers, open topology, independence) Gelenbe proved that such a network has a product-form solution for its steady-state distribution. It must be clear that the results are more complex than those for Jackson networks. The G-networks flow equations exhibit some uncommon properties: they are neither linear as in closed queueing networks nor contracting as in Jackson queueing networks. Therefore the existence of a solution had to be proved [10] by new techniques from the theory of fixed point equation. A numerical algorithm was developed in [5].

G-networks have also motivated many new important results in the theory of queues. The original proofs were based on the analysis of the global balance equations. Indeed, as negative customers lead to customer deletions, the original description of quasi-reversibility by arrivals and departures did not hold anymore and new versions have been proposed. At the time being, the description proposed by Chao and his co-authors in [2] looks sufficient to study queues with customers and signals. A completely different approach, based on Stochastic Process Algebra, was proposed by Harrison [12, 11]. The main results (CAT and RCAT theorems and their extensions [1, 12, 11]) give some sufficient conditions for product-form stationary distributions. This technique clearly has a different range of applications as it allows to represent component-based models which are much more general and more detailed than networks of queues.

These techniques have been used to study many new signals which all lead to product-form solutions: triggers which redirect other customers among the queues, catastrophes which flush all the customers out of a queue [8, 7], resets [9], synchronised arrivals in a set of queues [4], signals which change the class of the customer in service [16]. Here we present a new result for open G-networks where the effect of the signal is to restart a customer service. The service-times follow PH distributions, which are class dependent.

3. THE MODEL

We investigate generalized networks with an arbitrary number N of queues. We consider K classes of positive customers and only one class of negative customers (also denoted as signals). The external arrivals to the queues follow independent Poisson processes. The external arrival rate to queue i is denoted by $\lambda_i^{(k)}$ for positive customers of class k and Λ_i^- for signals.

The customers are served according to the processor sharing (PS) policy. The service times are assumed to be phase-type distributed. At phase p, the intensity of service for customers of class k in queue i is denoted as $\mu_i^{(k,p)}$. The transition probability matrix $H_i^{(k)}$ describes how, at queue i, the phase of a customer of class k evolves. Without loss of generality we assume that the PH distributions which describe the service times follow these rules:

- The initial state has index 1.
- The exit state has index 0.

Thus the service in queue i is an excursion from state 1 to state 0 following matrix $H_i^{(k)}$ for a customer of class k and we have:

$$\forall i,k,p \quad \sum_{q=0}^{P} H_i^{(k)}[p,q] = 1. \tag{1}$$

At its service completion time in queue i (i.e. transition from phase p to phase 0 in $H_i^{(k)}$), and according to a Markovian transition matrix, a customer of class k may join queue j as a positive customer of class l with probability $P_{i,j}^{+(k,l)}$. It may also leave the network with probability $d_i^{(k)}$. We assume that a customer cannot return to the queue it has just left: $P_{i,i}^{+(k,l)} = 0$ for all i, k and l. As usual, we have:

$$\forall i,k \quad \sum_{j=1}^{N}\sum_{l=1}^{K} P_{i,j}^{+(k,l)} + d_i^{(k)} = 1. \tag{2}$$

Signals do not stay in the network. At its arrival into a queue, a signal interacts with a selected customer and then vanishes instantaneously. If, at its arrival, the queue is already empty, it also vanishes instantaneously. The selected customer is chosen at random following a state-dependent distribution which mimics the PS scheduling. At state $\vec{x}_i$, the probability for a customer to be selected is $\frac{x_i^{(k,p)}}{|\vec{x}_i|}\mathbb{1}_{\{|\vec{x}_i|>0\}}$ and the signal has an effect with probability $\alpha_i^{(k,p)}$. The effect is the restarting of the customer: this customer (remember it has class k and phase p) is routed as a customer of class l at phase 1 with probability $R_i^{(k,l)}$. We assume for all k, $R_i^{(k,k)} = 0$. Of course we have:

$$\forall k \quad \sum_{l=1}^{K} R_i^{(k,l)} = 1. \tag{3}$$

The state of the queueing network is represented by the vector $\vec{x} = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N)$, where the component $\vec{x}_i$ denotes

the state of the queue i. As usual with multiple class PS queues with Markovian distribution of service, the state of queue i is given by the vector $(x_i^{(k,p)})$, for all class index k and phase index p. Furthermore, we note $|\vec{x}_i|$ the total number of customers in queue i.

Clearly $\vec{x}$ is a Markov chain.

Note that at the end of a service, we do not allow that a customer becomes a signal. This is not needed in our model because we do not want to represent the joint restart of a group of customers.

4. PRODUCT-FORM THEOREM

Let $p(\vec{x})$ be the stationary probability distribution of the network state if it exists. The following result establishes the existence of a product-form solution if a fixed point system has a solution which satisfies the stationarity constraints.

THEOREM 1. *Consider an arbitrary open G-network with p classes of positive customers and a single class of negative customers the effect of which is to restart one customer in the queue. If the system of linear equations:*

$$\rho_i^{(k,1)} = \frac{\lambda_i^{(k)} + \sum_{o=1}^{P} \mu_i^{(k,o)} \rho_i^{(k,o)} H_i^{(k)}[o,1] + \nabla_i^{k,1} + \Delta_i^{k,1}}{\mu_i^{(k,1)} + \Lambda_i^- \alpha_i^{(k,1)}}, \tag{4}$$

where

$$\Delta_i^{k,1} = \sum_{p=1}^{P} \sum_{l=1}^{K} \Lambda_i^- \alpha_i^{(l,p)} \rho_i^{(l,p)} R_i^{(l,k)}, \tag{5}$$

$$\nabla_i^{k,1} = \sum_{j=1}^{N} \sum_{l=1}^{K} \sum_{q=1}^{P} \mu_j^{(l,q)} \rho_j^{(l,q)} H_j^{(l)}[q,0] P_{j,i}^{+(l,k)}, \tag{6}$$

and,

$$\forall p > 1, \quad \rho_i^{(k,p)} = \frac{\sum_{o=1}^{P} \mu_i^{(k,o)} \rho_i^{(k,o)} H_i^{(k)}[o,p]}{\mu_i^{(k,p)} + \Lambda_i^- \alpha_i^{(k,p)}} \tag{7}$$

has a positive solution such that for all station i:

$$\sum_{k=1}^{K} \sum_{p=1}^{P} \rho_i^{(k,p)} < 1, \tag{8}$$

then the system stationary distribution exists and has product form:

$$p(\vec{x}) = \prod_{i=1}^{N} p_i(\vec{x}_i), \tag{9}$$

where

$$p_i(\vec{x}_i) = (1 - \sum_{k=1}^{K} \sum_{p=1}^{P} \rho_i^{(k,p)}) |\vec{x}_i|! \prod_{k=1}^{K} \prod_{p=1}^{P} \frac{(\rho_i^{(k,p)})^{x_i^{(k,p)}}}{x_i^{(k,p)}!}. \tag{10}$$

Before we can give the proof, we introduce some usual notations. We denote by $(\vec{x}_i + e_i^{(k,p)})$ (resp. $(\vec{x}_i - e_i^{(k,p)})$) the state of queue i obtained by adding (resp. suppressing) one customer of class k at phase p of service. We note $M_i^{(k,p)}(\vec{x}_i)$ the service rate of customers of class k in phase p at queue i. Since the service discipline considered is processor sharing, $M_i^{(k,p)}(\vec{x}_i)$ can be written as a function of $\mu_i^{(k,p)}$:

$$M_i^{(k,p)}(\vec{x}_i) = \mu_i^{(k,p)} \frac{x_i^{(k,p)}}{|\vec{x}_i|} \mathbb{1}_{\{|\vec{x}_i|>0\}}. \tag{11}$$

As the selection of customers mimics the PS discipline, the probability of restarting a customer of class k at step p when the queue is at state $\vec{x}_i$ is:

$$N_i^{(k,p)}(\vec{x}_i) = \alpha_i^{(k,p)} \frac{x_i^{(k,p)}}{|\vec{x}_i|} \mathbb{1}_{\{|\vec{x}_i|>0\}}. \tag{12}$$

The proof consists mainly of algebraic manipulations of the Chapmann-Kolmogorov equation for steady-state distribution:

$$\begin{aligned}
p(\vec{x}) \quad & \textstyle\sum_{i=1}^{N} \sum_{k=1}^{K} \Big[\lambda_i^{(k)} + \\
& \textstyle\sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i) \mathbb{1}_{\{x_i^{(k,p)}>0\}} + \\
& \textstyle\sum_{p=1}^{P} \Lambda_i^- N_i^{(k,p)}(\vec{x}_i) \mathbb{1}_{\{x_i^{(k,p)}>0\}}\Big] \\
= \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} \lambda_i^{(k)} p(\vec{x} - e_i^{(k,1)}) \mathbb{1}_{\{x_i^{(k,1)}>0\}} && [1] \\
+ \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i + e_i^{(k,p)}) d_i^{(k)} * \\
& \qquad H_i^{(k)}[p,0] p(\vec{x} + e_i^{(k,p)}) && [2] \\
+ \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{q=1}^{P} M_i^{(k,p)}(\vec{x}_i + e_i^{(k,p)} - e_i^{(k,q)}) * \\
& \qquad p(\vec{x} + e_i^{(k,p)} - e_i^{(k,q)}) H_i^{(k)}[p,q] \mathbb{1}_{\{x_i^{(k,q)}>0\}} && [3] \\
+ \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{j=1}^{N} \sum_{l=1}^{K} \sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i + e_i^{(k,p)}) P_{i,j}^{+(k,l)} * \\
& \qquad p(\vec{x} + e_i^{(k,p)} - e_j^{(l,1)}) H_i^{(k)}[p,0] \mathbb{1}_{\{x_j^{(l,1)}>0\}} && [4] \\
+ \quad & \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{p=1}^{P} \Lambda_i^- N_i^{(k,p)}(\vec{x}_i + e_i^{(k,p)}) \sum_{l=1}^{K} R_i^{(k,l)} * \\
& \qquad p(\vec{x} + e_i^{(k,p)} - e_i^{(l,1)}) \mathbb{1}_{\{x_i^{(l,1)}>0\}}. && [5]
\end{aligned} \tag{13}$$

Let us give some explanations about the right-hand side of this equation. The first term corresponds to external arrivals of customers. The second term is used to describe an end of service and departure to the outside while the third one is associated to the completion of phase p of service. With the fourth term we consider the case where a customer of class k leaves queue i to queue j as a customer of class l. The last term is associated with the restart: a signal arriving at queue i restart a customer of class k and phase p which joins queue i as a customer of class l at step 1 (term [5]). For the sake of readibility the proof of the theorem is transferred into an appendix. But it is also important to prove that the necessary conditions of the theorem do not exclude the existence of a feasible solution for Equations 4 to 7. Such a question is easy for Jackson's networks because the

necessary conditions are equivalent to a liear system which is proved to be non singular or contracting depending on the network topology. For G-networks, the problem is much more complex: see for instance [10].

4.1 Existence of a solution to the f ow equation

We now have to prove that there exists a non-empty region of the set of parameters where the flow equation has a solution. We reorganize the equations to simplify the presentation.

Let $\vec{q}$ be the vector of values $\rho_i^{(k,p)}\mu_i^{(k,p)}$ for all indices i, k, p. Thus $\vec{q}$ takes value in $\mathbb{R}^{NKP}$. The system of equation can easily be written as a linear system:

$$\vec{q} = \vec{q}G + \vec{a},$$

where $\vec{a}$ is the renormalized arrival rate (i.e. $\lambda_i^{(k)}\frac{\mu_i^{(k,p)}}{\mu_i^{(k,p)}+\Lambda_i^-\alpha_i^{(k,p)}}$) to take into account the signal. Proving that $(Id - G)$ is not singular is sufficient to guarantee the existence of $\vec{q}$. Thus we now consider matrix G and check if the system has a unique solution. Note that we do not check if the solution satifies Equation 8. We propose two solutions to prove that $(Id - G)$ is not singular.

The first idea consists in numerically compute matrix G from the parameters. Then we can check that the matrix is strictly sub-stochastic. Indeed, remember that if G is sub-stochastic and does not have any recurrent class, then $(Id - G)$ is not singular. The strict sub-stochastic property (every lines has a sum strictly smaller than one) implies that the matrix does not have any recurrent class and it is easier to check than the existence of recurrent classes. This is stated in the following two properties.

PROPERTY 1. *If G is strictly sub-stochastic, then Equations 4 to 7 have a solution in R^{NKP}.*

PROPERTY 2. *If G is sub-stochastic and does not contain any recurent class, then Equations 4 to 7 have a solution in R^{NKP}.*

Let us now consider a slightly more complex technique based on Brouwer's theorem to establish the existence of a fixed point system. We can easily decomposed matrix G into two matrices:

$$G = G1 + G2,$$

where Matrix $G1$ models the PS queues where all the entries have been multiplied by a damping factor equal to $\frac{\mu_i^{(k,p)}}{\mu_i^{(k,p)}+\Lambda_i^-\alpha_i^{(k,p)}}$ and $G2$ represents the terms associated to the arrival of restart signal. More precisely, the entries of $G1$ and $G2$ are described by:

- Matrix $G1$ (row (i,k,1), column (j,l,q))

$$\frac{\mu_i^{(k,1)}}{\mu_i^{(k,1)}+\Lambda_i^-\alpha_i^{(k,1)}}(H_i^{(k)}[q,1] + H_i^{(k)}[q,0]P_{j,i}^{+(l,k)}).$$

- Matrix $G1$ (row (i,k,p), column (i,k,o))

$$H_i^{(k)}[o,p]\frac{\mu_i^{(k,p)}}{\mu_i^{(k,p)}+\Lambda_i^-\alpha_i^{(k,p)}}.$$

- Matrix $G2$ (row (i,k,1), column (i,l,p))

$$\frac{\alpha_i^{(l,p)}}{\mu_i^{(l,p)}}R_i^{(l,k)}\frac{\mu_i^{(k,1)}}{\mu_i^{(k,1)}+\Lambda_i^-\alpha_i^{(k,1)}}\Lambda_i^-.$$

The system can be written as:

$$\vec{q}(Id - G1) = \vec{q}G2 + \vec{a},$$

where $\vec{a}$ has been previously defined. Let us now define $\mathcal{S}$ as the stability set (i.e. the set of non negative values of values of $\rho_i^{(k,p)}$ which satisfy the constraints in Equation 8). We have the following properties:

PROPERTY 3. *Assume that Matrix $(Id - G1)$ is not singular. One can find $\vec{b}$ such that $\vec{q}\,G2 \leq \vec{b}$ for all vectors $\vec{q}$ in $\mathcal{S}$.*

Furthermore let $\vec{q}_0$ be the vector such that:

$$\vec{q}_0 = (\vec{a} + \vec{b})\ (Id - G1)^{-1}.$$

$\vec{q}_0$ exists because matrix $(Id - G1)$ is not singular.

Proof: Indeed Matrix $G2$ is non negative and $\mathcal{S}$ is a closed and convex subset of R^{NKP}.

DEFINITION 1. *The network is superstable if $\vec{q}_0$ is in $\mathcal{S}$.*

Intuitively, superstability means that the network where signals are considered as arrival of fresh customers at the maximum rate allowed by the model is stable when considered as an ordinary network of PS queues with ordinary customers. Of course we expect that superstability implies stability. This is proved in Property 4. We begin by Brouwer's theorem.

THEOREM 2. *(Brouwer) Let F be a fixed point system, (i.e $x = F(x)$) in R^n. If*

1. *function F is continuous on a subset H of R^n*
2. *H is non empty*
3. *H is a compact and convex subset,*
4. *$F(H) \subseteq H$*

then $x = F(x)$ has at least one solution.

PROPERTY 4. *If the network is superstable, then there exist a solution to the equations 4 to 7 which satisfies the stability constraint of Equation 8.*

Proof: We will build a set $\mathcal{S}2$ such that the assumptions of Brouwer's theorem will be satisfied by the fixed point function on $\mathcal{S}2$. First note that $F(\vec{q}) = \vec{q}G + \vec{a}$. Therefore F is continuous on any subset of R^{NKP}. Now let $\mathcal{S}2$ be the set of non negative vectors $\vec{r}$ such that $\vec{r} \leq \vec{q}_0$ component-wise. $\mathcal{S}$ is compact, convex and non empty because $\vec{q}_0$ is non zero. Furthermore G is non negative, therefore $\vec{q} \leq \vec{q}_0$ implies that $\vec{q}G \leq \vec{q}_0 G$ and $F(\vec{q}) \leq F(\vec{q}_0) = \vec{q}_0$. And $F(\vec{q})$ is clearly non negative for a non negative vector $\vec{q}$. Therefore $F(\mathcal{S}2) \subseteq \mathcal{S}2$. All the assumptions of the theorem are satisfied. There exists a fixed point solution for F in $\mathcal{S}2$.

5. COMPUTATION OF MEASURES

Remember that the proof based on global balance has shown that

$$p(\vec{x}) = \prod_{i=1}^{N} p_i(\vec{x}_i), \tag{14}$$

where

$$p_i(\vec{x}_i) = C(1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \omega_i^{(k,p)})|\vec{x}_i|! \prod_{k=1}^{K}\prod_{p=1}^{P} \frac{(\omega_i^{(k,p)})^{x_i^{(k,p)}}}{x_i^{(k,p)}!}.$$

where C is a normalisation constant.

PROPERTY 5. *The normalization constant C is equal to 1.*

Proof: As the network is separable and all the states are reachable, we just have to verify that for each queue the sum of the probability is equal to one.

$$\sum_{\vec{x}_i} p_i(\vec{x}_i) = C \sum_{\vec{x}_i} (1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)})|\vec{x}_i|! \prod_{k=1}^{K}\prod_{p=1}^{P} \frac{(\rho_i^{(k,p)})^{x_i^{(k,p)}}}{x_i^{(k,p)}!}.$$

We partition the summation on $\vec{x}_i$ according to the norm of $\vec{x}_i$.

$$\begin{aligned}\sum_{\vec{x}_i} p_i(\vec{x}_i) &= C \sum_{m=0}^{\infty} (1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}) * \\ &\quad \sum_{\vec{x}_i/|\vec{x}_i|=m} |\vec{x}_i|! \prod_{k=1}^{K}\prod_{p=1}^{P} \frac{(\rho_i^{(k,p)})^{x_i^{(k,p)}}}{x_i^{(k,p)}!}.\end{aligned}$$

Substitute $|\vec{x}_i|$ by m in the previous equation. Remember the definition of the multinomial

$$\sum_{\vec{x}_i/|\vec{x}_i|=m} m! \prod_{k=1}^{K}\prod_{p=1}^{P} \frac{(\rho_i^{(k,p)})^{x_i^{(k,p)}}}{x_i^{(k,p)}!} = \left[\sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}\right]^m$$

After substitution we obtain:

$$\sum_{\vec{x}_i} p_i(\vec{x}_i) = C \sum_{m=0}^{\infty} (1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}) \left[\sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}\right]^m$$

As by assumption we have $\left[\sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}\right]^m < 1$, the sum converges and we get $\sum_{\vec{x}_i} p_i(\vec{x}_i) = C$. Thus the normalization constant in the product form is equal to one.

PROPERTY 6. *The former proof also establishes that the probability to have exactly m customers in the queue is equal to:*

$$Pr(m\ customers) = (1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}) \left[\sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}\right]^m.$$

This allows to get the expectation of the number of customers.

PROPERTY 7. *The expected number of customers in the queue is equal to:*

$$E[N] = \frac{\sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}}{1 - \sum_{k=1}^{K}\sum_{p=1}^{P} \rho_i^{(k,p)}}. \tag{15}$$

To get the average queuing delay, one usually applies Little's law. However, one must be very careful when describing a G-network with signals in this context. Indeed, signals may increase the queue size (see for instance the joint arrival described in [4] or the resets). In general one must take into account all these increases of the queue size when computing an "artificial" arrival rate into the queue. Here the situation is relatively simple. Signals do not increase the number of customers. Thus the arrival rate in Little's law is exactly the arrival of real customers.

6. EXAMPLES

We will now illustrate the approach using two examples. We start exploring the modelling power of the G-networks with restart using a relatively simple example. Throughout this section we use our Mathematica implementation of Eqn. 15.

6.1 Cycle of Erlangs

The first queueing-network we study consists of one queue with $K = 5$ classes. Jobs arrive only to the first class at rate $\lambda = 0.1$. In each class $k = 1, \ldots, 5$ service times follow an Erlang distribution of length 2. Subsequent classes have increasingly higher rates. For a job in $k = 1, \ldots, 4$, restart causes the job to move from class k to class $k + 1$, which implies that service gets faster upon restart. For jobs in class 5, restart means that the job is moved back to class 1. This type of model is appropriate to study the effect of restart in scenarios where the service becomes faster (or, equivalently, service-demand is reduced) on subsequent trials because the service was already partially completed on previous attempts, but where the user aborts and repeats the complete request, including all partial results, after several restarts. For instance, this is often the case with downloads of files or websites from the Internet. A similar model could also be used to study restart in a scenario with load-balancing, where incoming and restarted jobs are assigned to servers with different processing speeds.

The parameters are defined as follows: Jobs only arrive to class 1 at rate $\lambda_1 = 0.1$. The arrival rate to all other classes $k > 1$ equals zero, i.e. $\lambda_k = 0$, for $k > 1$. The rates of the Erlang service-time distributions are defined as $\mu_1^{(1,p)} = 1, \mu_1^{(2,p)} = 2, \mu_1^{(3,p)} = 3, \mu_1^{(4,p)} = 4, \mu_1^{(5,p)} = 5$. Signals arrive at rate Λ to the queue, which we vary across the experiments. Signals always lead to abort and restart, hence $\alpha_i^{(k,p)} = 1, \forall i, k, p$. The matrix H is for the Erlang distribution in all k classes $H_1^k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. The routing matrix R sends each job after restart to the next class, where it completes the remainder of its service. If a job does not meet the restart deadline in the final class it is reissued and starts all over from scratch. The matrix R describes the change of class from k_1 to k_2 as a job is restarted.

$$R_i[k_1, k_2] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Completed jobs leave the network with probability $d_i^k = 1$. In consequence, the probability Pr that a job remains in the network after completion is zero. The parameters are not

motivated by experimental data, they purely illustrate the dynamics of the model.

The scenario is illustrated in Figure 1.

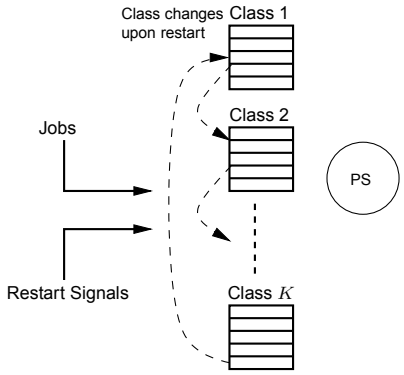

Figure 1: Multiple class G-network with restart

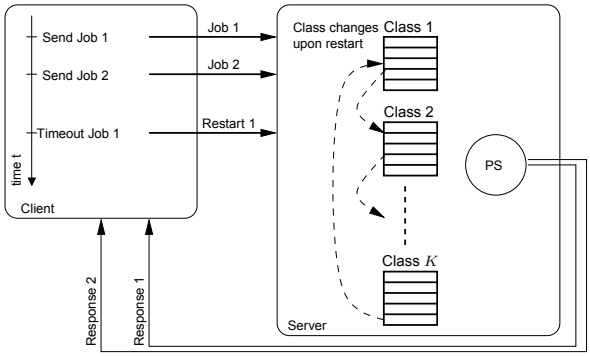

Figure 2: Simulation scenario

Figure 3 shows the results from this model. Several observations can be made. Even though restarts increase the load, the system remains stable for the range of parameters we investigate. Even stronger, for zero restart rate the system behaves as a system without restart, which is the well-known M/PH/1 queue with processor sharing discipline [**?**]. We can observe that for a suitably chosen restart rate the utilisation of the system can be reduced with restart. Further, we can see that the queue length and the waiting time are a function of the utilisation ρ and hence both have the same minimum as the utilisation with respect to the restart rate. The waiting time is computed using Little's law where the restart rate is not included in the arrival rate of jobs. Obviously, restart does not increase the number of jobs that enter the system, but it does manipulate the time spent in the system by each job.

In order to validate the results, we simulated the scenario using the SFERA framework [3]. SFERA is a framework for evaluation of restart algorithms using discrete-event simulation. We implemented the model shown in Figure 2: The client generates jobs as a Poisson process and sends them to the server, where they are processed according to the processor-sharing strategy. Upon completion of a job, the server sends a response back to the client. Before sending each job to the server, the client draws a restart interval for that job from an exponential distribution. When the interval elapses, the client sends a restart signal for this job to the server. The server aborts the processing of the job and restarts it in the next class, according to the configuration of the restart acceptance and routing probabilities. We configured the classes to form a circle, as for the G-network. Note that this model is much more realistic than the G-network model, in that it allows restart intervals specific to each job. In contrast, in the G-network model the restart signals arrive independently of jobs and affect random jobs.

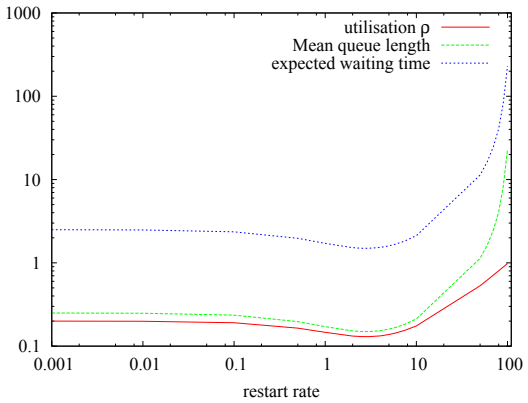

Figure 3: Expected queue length and expected waiting time for different values of the restart rate

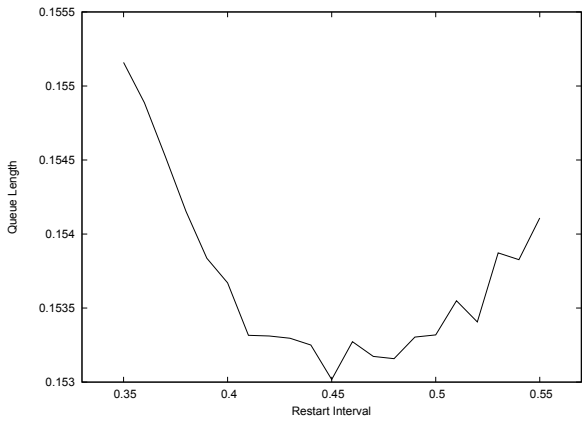

Figure 4: Average queue length and average response time for the simulation

With SFERA we can observe the average queue length in the server and the completion-times, as experienced by the client. We show the results of the simulation in Figure 4. It is obvious that the simulation results also indicate the existence of an optimal restart rate at which the queue length and completion-times are minimised. Please note that the optimal restart interval in Fig. 4 of approximately 0.45 corresponds to an optimal restart rate of 2.2, which is close to the optimal value shown in Figure 3. For restart intervals larger than this value, the measures approach the case without restart, while for intervals below it both measures increase because the server cannot complete jobs within the allotted restart intervals. We also note that the optimal value of the restart rate is slightly different from the simulation results. We attribute this to the dif-

ferences in the models, since the simulation model includes dependencies between the job and the signal arriving processes, which are not present in the G-network model. In the simulation model each arriving job is assigned its individual restart timeout that is sampled from an exponential distribution at that time. The timeout could also be drawn from an arbitrary distribution (including the deterministic). Each event in the simulation, i.e. the arrival or departure of a job, leads to a reschedule of all present deadlines but the timeout is not resampled. In contrast, the semantics of the analytical model imply that timeouts are not stored. Instead, the time between signals (restarts) is a randomly sampled sequence of non-overlapping time intervals.

While in the simulation a job at the head of the queue experiences expiry of its timeout with higher probability than a job at the end of the queue, in the G-network each job in the queue is equally likely to be hit by a signal and restarted. Through the decreasing timeout value the simulation includes some notion of age of a job, which is not present in the G-network.

Taking into account that in real-world scenarios restart is typically triggered by a job-specific timeout, rather than an independent stream of signals, our simulation results thus also illustrate that the predictions of the theoretical model hold reasonably well in realistic scenarios.

6.2 A more complex model

In our second example we formulate a more complex, and more realistic, model. The model again has only one queue, which is a limitation in the implementation, not in the formalism. The model has 5 classes of customers. Jobs arrive at rate $\lambda = 0.004$ to class 1 and at rate $\lambda = 0.01$ to class 2. We make the following assumptions about the restart:

- Classes 1 and 2 represent "fresh" customers while class 3 (resp. class 4 and 5) represent customers of class 1 (resp. class 2 and 3) which have been restarted once.
- It is not possible to restart a customer of class 4.
- When we restart a customer of class 5, it becomes a customer of class 1 again.

We assume the following distributions for the service (the first state of the PH has state number zero and it represents the end of the PH (see page 1).

- Class 1 $H_1^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ with $\mu_1^{1,1} = 5$, $\mu_1^{1,2} = 2$ and $\mu_1^{1,3} = 0.1$. Thus the PH describes an exponential of rate 5 followed by either an exponential of rate 2 with probability 0.9 or an exponential of rate 0.1 (which means a long average time compared to the other transitions).

  We also assume the following about the effect of the restart: if the customer selected at the queue by the restart signal has class k and is in phase p, the restart succeeds with probability $\alpha_i^{(k,p)}$. We model the following typical assumptions: restarts do only succeed with probability 1/2 for customer of class 1 when it is in phase 1 or 2. And it succeeds when it is in phase 3.

  $$\alpha_1^{(1,1)} = 0.5,\ \alpha_1^{(1,2)} = 0.5,\ \alpha_1^{(1,3)} = 1.$$

- Class 2 $H_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 2/3 & 1/3 \\ 3/4 & 1/4 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ with $\mu_1^{1,1} = 1$, $\mu_1^{1,2} = 1$ and $\mu_1^{1,3} = 0.1$. Thus the PH has a directed loop (this is possible; we are not restricted to acyclic PH). Again phase 3 has a long average time compared to the other transitions). Restarts do not succeed for customer of class 2 when it is in phase 1 or 2, $\alpha_1^{(2,1)} = \alpha_1^{(2,2)} = 0$. A restart succeeds with probability 1 when it is in phase 3.

  $$\alpha_1^{(1,3)} = 1.$$

- Class 3 represents restarted class 1 customers. We assume an Erlang 3 with rate equal to 0.5. This is modelled by matrix $H_1^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ with $\mu_1^{3,1} = \mu_1^{3,2} = \mu_1^{3,3} = 0.5$. We assume that it is possible to restart class 3 customers.

  $$\alpha_1^{(3,1)} = 1,\ \alpha_1^{(3,2)} = \alpha_1^{(3,3)} = 2/3$$

- Class 4 represents restarted class 2 customers. We assume an Erlang-3 with rate 0.5. This is modelled by matrix $H_1^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$ with $\mu_1^{4,1} = 0.5$, $\mu_1^{4,2} = 0.5$, $\mu_1^{4,3} = 0.5$. We assume that it is unlikely to restart class 4 customers in phase 1 but quite likely to restart them successfully in phase 2 and 3. Thus

  $$\alpha_1^{(4,1)} = 0.1,\ \alpha_1^{(4,2)} = 2/3,\ \alpha_1^{(4,3)} = 2/3$$

- Class 5 represent customers which have been restarted twice. The service is identical to that for class 4.

  The customer can be restarted: $\alpha_1^{(5,1)} = 0.5,\ \alpha_1^{(5,2)} = 0.5\ \alpha_1^{(5,3)} = 1$.

The routing matrix is as follows.

$$R_i[k_1, k_2] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We show the analytical results for this model in Figure 5. Again, we observe the existence of an optimal restart rate, which for this model is around 10, and the typical increase in the measures above and beyond this value.

7. APPLICATIONS

The G-Networks-based approach to the analysis of restart we presented here can be used to evaluate the effects of restart in systems that can be modelled as queueing networks, such as service compositions in service-oriented systems. The approach can be applied both by the designer or operator of such systems and, to a limited degree, by the service user.

In the following we sketch application of the approach in the analysis of a practical system. We use a service-oriented

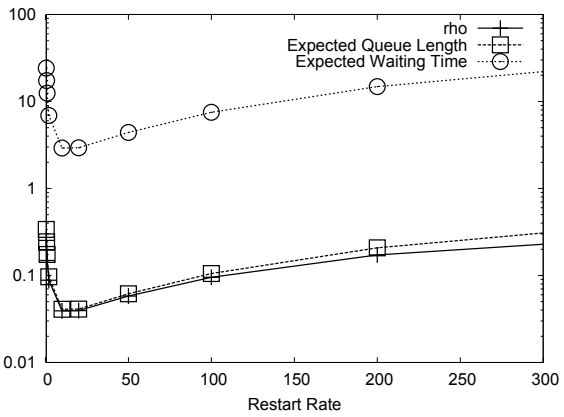

Figure 5: Load ρ, average queue length and average waiting time for the second model.

system as our example, where we assume that several services are combined to form a service-composition. Each service is implemented with a queue for incoming requests and processing threads to handle the requests. The first step in the evaluation then consists in obtaining a structural model of the system. This structural model represents the different services as queues with associated servers, and captures the request flows between services in the routing matrix. In the second step, the processing characteristics of the services must be captured and modelled as phase-type distributions, which will then be included as the service-time distributions in the model. Adequate service-time distributions would typically be obtained by measuring processing times and fitting phase-type distributions to the samples, using one of several well-known tools [13, 17, 15]. The impact of restart can then be evaluated in the third step using Equation 15. For instance, the designer or operator of the system may be interested in whether the system will be stable under restart, and for which range of restart intervals stability holds. This is of particular importance with systems whose users are likely to restart requests due to impatience, such as online shopping. If the evaluation indicates that restart can endanger the stability of the system, countermeasures may be applied, such as increasing the system's capacity or preventing restart. On the other hand, application of Equation 15 allows a user of the system to compute an optimal restart timeout. However, this application of the method is currently limited to cases where the service-time distributions and the system architecture are published by the services.

The approach may also be employed in the development and evaluation of algorithms to compute restart timeouts. Here, the fact that subsequent restarts may encounter different service-time distributions can be modelled explicitly, as in our first example. Furthermore, the implications of many users applying restart can be incorporated by increasing job and signal arrival rates.

It should be mentioned that modelling restarts with G-networks may come with some hardships. The possibility of infinitely many restarts cannot directly be incorporated into the model. We solve it with creating a cycle from the first to the last queue. This only leads to an exact representation if all service time distributions are equal. Then the model can be reduced to two classes and jobs alternate between them.

We did not present models for a finite number of restarts as we did not obtain optimal restart intervals for those and found them otherwise difficult to validate. It is unclear whether or not such an optimal restart interval exists with finite trials. The formulation and solution of the model is straight forward.

8. CONCLUSIONS

In this paper we considered multiple-class G-networks with restart as an approach for studying the effectiveness of the restart method in systems that can be modelled as queueing-networks with multiple classes of users with class-dependent service-times. The G-network formalism allowed us to show that such systems can remain stable, and indeed superstable, under restart. Furthermore, we showed that in this case there exists an optimal restart rate at which the system load, and consequently other measures of interest (such as the waiting time), is minimised. Our comparison of the analytical results to simulation results indicates that this is also the case in more realistic scenarios where the arrival process of restart signals is not independent of the job arrival process.

Acknowledgements

Katinka Wolter and Philipp Reinecke are partly supported by the German Science Foundation (DFG) under grant No. Wo 898/3-1 and by DAAD PROCOPE programme for cooperation with French research institutions.

9. REFERENCES

[1] Simonetta Balsamo, Peter G. Harrison, and Andrea Marin. A unifying approach to product-forms in networks with finite capacity constraints. In Vishal Misra, Paul Barford, and Mark S. Squillante, editors, *SIGMETRICS 2010, Proceedings of the 2010 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, New York*, pages 25–36. ACM, 2010.

[2] X. Chao, M. Miyazawa, and M. Pinedo. *Queueing Networks Customers, Signals and Product Form solutions*. John Wiley & Sons, 1999.

[3] Alexandra Danilkina, Philipp Reinecke, and Katinka Wolter. Sfera: A simulation framework for the performance evaluation of restart algorithms in service-oriented systems. *Electronic Notes in Theoretical Computer Science*, 291:3–14, 2013. <ce:title>Second Workshop on Quantitative Models for Performance and Dependability (QMPD 2012)</ce:title>.

[4] Thu-Ha Dao-Thi, J.-M. Fourneau, and Minh-Anh Tran. G-networks with synchronised arrivals. *Perform. Eval.*, 68(4):309–319, 2011.

[5] J.-M. Fourneau. Computing the steady-state distribution of networks with positive and negative customers. In *13^{t}h IMACS World Congress on Computation and Applied Mathematics, Dublin*, 1991.

[6] E. Gelenbe. Product-form queuing networks with negative and positive customers. *Journal of Applied Probability*, 28:656–663, 1991.

[7] E. Gelenbe. G-networks with instantaneous customer movement. *Journal of Applied Probability*, 30(3):742–748, 1993.

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

[9] E. Gelenbe and J.-M. Fourneau. G-networks with resets. *Perform. Eval.*, 49(1-4):179–191, 2002.

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

[11] Peter G. Harrison. Compositional reversed Markov processes, with applications to G-networks. *Perform. Eval.*, 57(3):379–408, 2004.

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

[13] András Horváth and Miklós Telek. PhFit: A General Phase-Type Fitting Tool. In *TOOLS '02: Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*, pages 82–91, London, UK, 2002. Springer-Verlag.

[14] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The Johns Hopkins University Press, 1981.

[15] Philipp Reinecke, Tilman Krauss, and Katinka Wolter. Hyperstar: Phase-type fitting made easy. In *Quantitative Evaluation of Systems (QEST), 2012 Ninth International Conference on*, pages 201 –202, sept. 2012.

[16] Thu-Ha Dao Thi, J.-M. Fourneau, and Minh-Anh Tran. Networks of symmetric multi-class queues with signals changing classes. In Khalid Al-Begain, Dieter Fiems, and W. J. Knottenbelt, editors, *Analytical and Stochastic Modeling Techniques and Applications, 17th International Conference, ASMTA 2010, Cardiff, UK*, volume 6148 of *Lecture Notes in Computer Science*, pages 72–86. Springer, 2010.

[17] Axel Thümmler, Peter Buchholz, and Miklos Telek. A novel approach for phase-type fitting with the em-algorithm. *IEEE Transactions on Dependable and Secure Computing*, 3:245–258, 2006.

[18] Aad van Moorsel and Katinka Wolter. Analysis and Algorithms for Restart. In *Proc. 1st International Conference on the Quantitative Evaluation of Systems (QEST)*, pages 195–204, Twente, The Netherlands, September 2004. Best paper award.

[19] Aad P. A. van Moorsel and Katinka Wolter. Analysis of restart mechanisms in software systems. *IEEE Transactions on Software Engineering*, 32(8):547–558, August 2006.

[20] Aad P.A. van Moorsel and Katinka Wolter. Optimal restart times for moments of completion time. *IEE Proceedings Software*, 151(5):219– 223, October 2004.

Appendix: Proof of Theorem 1:

First remember the Chapman-Kolmogorov equation for the steady-state distribution:

$$
\begin{aligned}
p(\vec{x}) \quad & \sum_{i=1}^{N}\sum_{k=1}^{K}\Big[\lambda_i^{(k)}+ \\
& \sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i)\mathbb{1}_{\{x_i^{(k,p)}>0\}}+ \\
& \sum_{p=1}^{P} \Lambda_i^- N_i^{(k,p)}(\vec{x}_i)\mathbb{1}_{\{x_i^{(k,p)}>0\}}\Big] \\
= & \sum_{i=1}^{N}\sum_{k=1}^{K}\lambda_i^{(k)}p(\vec{x}-e_i^{(k,1)})\mathbb{1}_{\{x_i^{(k,1)}>0\}} \\
+ & \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)})d_i^{(k)}* \\
& H_i^{(k)}[p,0]p(\vec{x}+e_i^{(k,p)}) \\
+ & \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{q=1}^{P} M_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)}-e_i^{(k,q)})* \\
& p(\vec{x}+e_i^{(k,p)}-e_i^{(k,q)})H_i^{(k)}[p,q]\mathbb{1}_{\{x_i^{(k,q)}>0\}} \\
+ & \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{p=1}^{P} M_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)})P_{i,j}^{+(k,l)}* \\
& p(\vec{x}+e_i^{(k,p)}-e_j^{(l,1)})H_i^{(k)}[p,0]\mathbb{1}_{\{x_j^{(l,1)}>0\}} \\
+ & \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\Lambda_i^- N_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)})\sum_{l=1}^{K}R_i^{(k,l)}* \\
& p(\vec{x}+e_i^{(k,p)}-e_i^{(l,1)})\mathbb{1}_{\{x_i^{(l,1)}>0\}}.
\end{aligned}
$$

We divide both sides by $p(\vec{x})$ and we take into account the following simplications:

$$\frac{p(\vec{x}-e_i^{(k,1)})}{p(\vec{x})} = \frac{x_i^{(k,1)}}{\rho_i^{(k,1)}|\vec{x}_i|} \tag{16}$$

$$\frac{p(\vec{x}+e_i^{(k,p)})}{p(\vec{x})} = \frac{\rho_i^{(k,p)}(|\vec{x}_i|+1)}{x_i^{(k,p)}+1} \tag{17}$$

$$\frac{p(\vec{x}+e_i^{(k,p)}-e_i^{(k,q)})}{p(\vec{x})} = \frac{\rho_i^{(k,p)}x_i^{(k,q)}}{\rho_i^{(k,q)}(x_i^{(k,p)}+1)} \tag{18}$$

$$M_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)}) = \mu_i^{(k,p)}\frac{x_i^{(k,p)}+1}{|\vec{x}_i|+1} \tag{19}$$

$$M_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)}-e_i^{(k,q)}) = \mu_i^{(k,p)}\frac{x_i^{(k,p)}+1}{|\vec{x}_i|}\mathbb{1}_{\{|\vec{x}_i|>0\}} \tag{20}$$

$$N_i^{(k,p)}(\vec{x}_i+e_i^{(k,p)}) = \alpha_i^{(k,p)}\frac{x_i^{(k,p)}+1}{|\vec{x}_i|+1}. \tag{21}$$

Combining these relations we get: $\frac{p(\vec{x}+e_i^{(k,p)}-e_j^{(l,q)}+e_j^{(l,o)})}{p(\vec{x})} = \frac{\rho_i^{(k,p)}(|\vec{x}_i|+1)}{x_i^{(k,p)}+1}\frac{\rho_j^{(l,o)}x_j^{(l,q)}}{\rho_j^{(l,q)}(x_j^{(l,o)}+1)}$

After substitution and simplification we get:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=1}^{K}\Big[\lambda_i^{(k)}+ \\
&\quad \textstyle\sum_{p=1}^{P}(\mu_i^{(k,p)}+\Lambda_i^-\alpha_i^{(k,p)})* \\
&\quad \tfrac{x_i^{(k,p)}}{|\vec{x}_i|}\mathbb{1}_{\{|\vec{x}_i|>0\}}\mathbb{1}_{\{x_i^{(k,p)}>0\}}\Big] \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\lambda_i^{(k)}\frac{x_i^{(k,1)}}{|\vec{x}_i|\rho_i^{(k,1)}}\mathbb{1}_{\{x_i^{(k,1)}>0\}} \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\mu_i^{(k,p)}\rho_i^{(k,p)}H_i^{(k)}[p,0]d_i^{(k)} \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{q=1}^{P}\mu_i^{(k,p)}\frac{\rho_i^{(k,p)}}{\rho_i^{(k,q)}}\frac{x_i^{(k,q)}}{|\vec{x}_i|}* \\
&\qquad H_i^{(k)}[p,q]\mathbb{1}_{\{x_i^{(k,q)}>0\}} \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{p=1}^{P}\mu_i^{(k,p)}\frac{\rho_i^{(k,p)}}{\rho_j^{(l,1)}}* \\
&\qquad H_i^{(k)}[p,0]P_{i,j}^{+(k,l)}\tfrac{x_j^{(l,1)}}{|\vec{x}_j|}\mathbb{1}_{\{x_j^{(l,1)}>0\}} \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{l=1}^{K}\Lambda_i^-\alpha_i^{(k,p)}\rho_i^{(k,p)}* \\
&\qquad R_i^{(k,l)}\tfrac{x_i^{(l,1)}}{\rho_i^{(l,1)}|\vec{x}_i|}\mathbb{1}_{\{x_i^{(l,1)}>0\}}
\end{aligned} \tag{22}$$

First remark that $\frac{x_i^{(k,p)}}{|\vec{x}_i|}\mathbb{1}_{\{x_i^{(k,p)}>0\}} = \frac{x_i^{(k,p)}}{|\vec{x}_i|}\mathbb{1}_{\{|\vec{x}_i|>0\}}$. Thus we can simplify the left hand side of the equation.

Now consider the right hand side. We exchange some indices to simplify the expressions:

- queues indices i and j in the fourth term
- class indices k and l in the fourth and the fifth terms
- and phases indices p and q in the third and fourth terms.

It is now easy to factorize the first term with the term and the fifth term. We obtain:

$$\begin{aligned}
&\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{x_i^{(k,1)}}{|\vec{x}_i|\rho_i^{(k,1)}}\mathbb{1}_{\{|\vec{x}_i|>0\}}\Big[\lambda_i^{(k)}+ \\
&\quad \sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P}\mu_j^{(l,q)}\rho_j^{(l,q)}H_j^{(l)}[q,0]P_{j,i}^{+(l,k)}+ \\
&\quad \sum_{p=1}^{P}\sum_{l=1}^{K}\Lambda_i^-\alpha_i^{(l,p)}\rho_i^{(l,p)}R_i^{(l,k)}\Big]
\end{aligned}$$

Now we remember that:

$$\Delta_i^{k,1} = \sum_{p=1}^{P}\sum_{l=1}^{K}\Lambda_i^-\alpha_i^{(l,p)}\rho_i^{(l,p)}R_i^{(l,k)}$$

and,

$$\nabla_i^{k,1} = \sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P}\mu_j^{(l,q)}\rho_j^{(l,q)}H_j^{(l)}[q,0]P_{j,i}^{+(l,k)}$$

We can now substitute all these relations in the balance equation.

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{k=1}^{K}\lambda_i^{(k)} &+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\Big[\mu_i^{(k,p)}+\Lambda_i^-\alpha_i^{(k,p)}\Big]* \\
&\qquad \frac{x_i^{(k,p)}}{|\vec{x}_i|}\mathbb{1}_{\{|\vec{x}_i|>0\}} \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\frac{x_i^{(k,1)}}{|\vec{x}_i|\rho_i^{(k,1)}}\mathbb{1}_{\{|\vec{x}_i|>0\}}* \\
&\qquad \Big[\lambda_i^{(k)}+\nabla_i^{k,1}+\Delta_i^{k,1}\Big] \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\mu_i^{(k,p)}\rho_i^{(k,p)}H_i^{(k)}[p,0]d_i^{(k)} \\
&+ \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\frac{x_i^{(k,p)}}{|\vec{x}_i|\rho_i^{(k,p)}}\mathbb{1}_{\{x_i^{(k,p)}>0\}}* \\
&\qquad \sum_{q=1}^{P}\mu_i^{(k,q)}\rho_i^{(k,q)}H_i^{(k)}[q,p]
\end{aligned} \tag{23}$$

Taking into account equations 4 to 7, all the state-dependent terms cancel and we obtain:

$$\sum_{i=1}^{N}\sum_{k=1}^{K}\lambda_i^{(k)} = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{p=1}^{P}\mu_i^{(k,p)}\rho_i^{(k,p)}H_i^{(k)}[p,0]d_i^{(k)} \tag{24}$$

This equation is a flow equation. Indeed the l.h.s. is the flow of positive customers entering the system while the r.h.s represents the customers leaving the network. This is formally proved in the next section.

9.1 Proof of the f ow equation

Let us now consider Equations 4 to 7 to prove formally that equation 24 is a flow equation. Multiply $\rho_i^{(k,p)}$ and $\rho_i^{(k,1)}$ by the denominator in equation 7 or 4 and make the summation for all p.

$$\begin{aligned}
\sum_{p=1}^{P}\rho_i^{(k,p)}(\mu_i^{(k,p)} &+ \Lambda_i^-\alpha_i^{(k,p)}) \\
&= \lambda_i^{(k)} \\
&+ \sum_{p=1}^{P}\sum_{q=1}^{P}\mu_i^{(k,q)}\rho_i^{(k,q)}H_i^{(k)}[q,p] \\
&+ \sum_{p=1}^{P}\sum_{l=1}^{K}\Lambda_i^-\alpha_i^{(l,p)}\rho_i^{(l,p)}R_i^{(l,k)} \\
&+ \sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P}\mu_j^{(l,q)}\rho_j^{(l,q)}* \\
&\qquad H_j^{(l)}[q,0]P_{j,i}^{+(l,k)}.
\end{aligned} \tag{25}$$

Note that for all i,k,q we have:

$$\sum_{p=1}^{P}H_i^{(k)}[q,p] = 1 - H_i^{(k)}[q,0].$$

Thus equation 25 becomes after substitution and cancellation :

$$\sum_{p=1}^{P} \rho_i^{(k,p)} \Lambda_i^- \alpha_i^{(k,p)} \quad + \quad \sum_{q=1}^{P} \rho_i^{(k,q)} \mu_i^{(k,q)} H_i^{(k)}[q,0] \quad = \quad \lambda_i^{(k)} \quad + \quad \sum_{p=1}^{P}\sum_{l=1}^{K} \Lambda_i^- \alpha_i^{(l,p)} \rho_i^{(l,p)} R_i^{(l,k)} \quad + \quad \sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P} \mu_j^{(l,q)} \rho_j^{(l,q)} H_j^{(l)}[q,0] P_{j,i}^{+(l,k)}. \tag{26}$$

We now sum over all values of k and we take into account that for all l we have $\sum_{k=1}^{K} R_i^{(l,k)} = 1$. We cancel some terms.

$$\sum_{k=1}^{K}\sum_{q=1}^{P} \rho_i^{(k,q)} \mu_i^{(k,q)} H_i^{(k)}[q,0] \quad = \quad \sum_{k=1}^{K} \lambda_i^{(k)} \quad + \quad \sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P} \mu_j^{(l,q)} \rho_j^{(l,q)} H_j^{(l)}[q,0] P_{j,i}^{+(l,k)}. \tag{27}$$

We now sum over all values of queue index i and we remember that for all j and l, we have:

$$\sum_{i=1}^{N}\sum_{k=1}^{K} P_{j,i}^{+(l,k)} = 1 - d_j^{(l)}.$$

After substitution in Equation 27 and cancellation, we get:

$$\sum_{i=1}^{N}\sum_{k=1}^{K} \lambda_i^{(k)} \quad = \quad \sum_{j=1}^{N}\sum_{l=1}^{K}\sum_{q=1}^{P} \rho_j^{(l,q)} \mu_j^{(l,q)} H_j^{(l)}[q,0] d_j^{(l)},$$

which is the same as Equation 24. Thus it is a flow equation which is consistent with Equations 4 to 7