Job migrations in queueing networks: some non-conventional product-forms

Andrea Marin

Università Ca' Foscari Venezia, Italy

Outline

Section 1

Motivation and goals

Networks of exponential queues

- We consider queueing networks (QNs) of Jackson/Gordon-Newell type
- N queueing stations numbered from 1 to N
- Single class, open or closed topology
- Independent, exponential, state independent service time distribution with rate μ_i
- State independent probabilistic routing $\mathbf{P} = [p_{ij}]$ with $j = 0$ stands for the outside for open networks
- Arrivals from the outside at station i follow a homogeneous independent Poisson process with rate λ_i (for open topologies)

Product-form in closed topologies

- Assuming the irreducibility of the routing process the CTMC underlying this class of QNs is irreducible
- $\mathbf{n} = (n_1, \ldots, n_N)$ is the system's state, n_i number of jobs at station i
- For closed topologies there always exists a stationary distribution in product-form, i.e.:

$$\pi(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^{N} g_i(n_i)\,,$$

where $g_i(n_i) = (e_i/\mu_i)^{n_i}$, $\mathbf{e} = (e_1, \ldots, e_N)$ is a non-trivial solution of the traffic equations, and G is a normalising constant

Product-form in open topologies

- Assumptions: irreducibility of the routing process, ergodicity of the CTMC underlying the QN
- There exists a stationary distribution in product-form, i.e.:

$$\pi(\mathbf{n}) = \prod_{i=1}^{N} g_i(n_i)\,,$$

where $g_i(n_i) = (1 - e_i/\mu_i)(e_i/\mu_i)^{n_i}$, $\mathbf{e} = (e_1, \ldots, e_N)$ is the solution of the traffic equations, and G is a normalising constant

- The CTMC is ergodic if and only if $e_i/\mu_i < 1$ for all $i = 1, \ldots, N$

The proposed mechanisms for load-balancing

- LB-networks[1]: Move customers from highly loaded queueing stations to lowly loaded ones with same capabilities
- Fetching queueing networks[2]: Customers can skip some phases of service to maximize the resource utilization
- Global stateless methods: the stations do not record statistics on the network behaviour and each station knows only its own state
- Product-forms: Under certain conditions the QNs maintain the product-form solution

[1] Joint work with J.M. Fourneau and S. Balsamo
[2] Joint work with D. Olliaro, S. Rossi and G. Casale

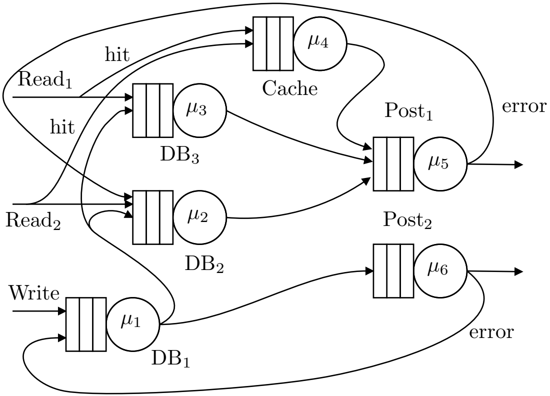
Two approaches from the literature

- The problem of dynamic load-balancing is widely studied in the literature
- Two major approaches
  - **Sender initiated**: a station with many jobs tries to send some of them to another station
  - **Receiver initiated**: a station with few jobs tries to serve some of the jobs of another station
- Receiver initiated strategies have been shown to work better in most practical cases
- Our approaches belong to the class of receiver initiated strategies

Section 2

LB-networks

Example: Stations with same functionalities



Job migration policy

- When station i is empty for a random amount of time α_{ij}, it polls station j
- Let t be the time epoch of polling and T_{ji} be the number of jobs moved from station j to i
- The distribution of $T_{ji}(t)$ is a truncated geometric with parameter b_{ij}:

$$Pr\{T_{ji}(t) = k\} = \begin{cases} (1-b_{ij})b_{ij}^{n_j(t)} & \text{if } k < n_j(t) \\ b_{jk}^{n_j(t)} & \text{if } k = n_j(t) \\ 0 & \text{otherwise} \end{cases}$$

- We assume the job transfer time to be negligible w.r.t. the other delays
- When $b_{ij} = 1$ all the jobs present at time t in j are moved to i

Some problems to solve

- Which is the best parameter for the exponential delay α_{ij}?
- Which is the best parameter b_{ij} for the geometric batch size distribution?
- Can we maintain the analytical tractability in product-form of QN with such job migration policy?
  - Notice that we have batches of jobs moving in the QN and their routing is conditioned on some station states

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{z} + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{w} + \lambda_i}{\mu_i + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{v} b_{ki}}$$

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1\\k\neq i}}^{N} x^z_{ki} + \sum_{\substack{k=1\\k\neq i}}^{N} x^w_{ki} + \lambda_i}{\mu_i + \sum_{\substack{k=1\\k\neq i}}^{N} x^v_{ki} b_{ki}}$$

- arrivals due to ordinary routing

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{z} + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{w} + \lambda_i}{\mu_i + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{v} b_{ki}}$$

- arrivals due to ordinary routing
- arrivals due to migrations into station i

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{z} + \sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{w} + \lambda_i}{\mu_i + \sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{v} b_{ki}}$$

- arrivals due to ordinary routing
- arrivals due to migrations into station i
- exogenous arrivals

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{z} + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{w} + \lambda_i}{\mu_i + \sum_{\substack{k=1\\k\neq i}}^{N} x_{ki}^{v} b_{ki}}$$

- arrivals due to ordinary routing
- arrivals due to migrations into station i
- exogenous arrivals
- departures due to internal service

The load factor of a queue

- Let ρ_i be the load factor of queue i

$$\rho_i = \frac{\sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{z} + \sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{w} + \lambda_i}{\mu_i + \sum_{\substack{k=1 \\ k\neq i}}^{N} x_{ki}^{v} b_{ki}}$$

- arrivals due to ordinary routing
- arrivals due to migrations into station i
- exogenous arrivals
- departures due to internal service
- departures due to job migration

The product-form theorem

Given a LB-network with ergodic underlying CTMC, the stationary distribution is in product-form if the following system of rate equations in the unknowns ρ_i, x_{ij}^v, x_{ij}^w and x_{ij}^z, admits a solution:

$$\begin{cases} x_{ij}^z = \rho_i \mu_i p_{ij} \\ x_{ij}^v = \frac{x_{ji}^w}{\rho_i} \\ \alpha_{ij} = x_{ij}^v \\ x_{ij}^w = \rho_i x_{ji}^v b_{ji} \end{cases}$$

and $\pi(\mathbf{n}) = \frac{1}{G} \prod_{i=1}^{N} \rho_i^{n_i}$ where G is a normalising constant.

Properties of the solutions

Let χ_{ij}^v, χ_{ij}^w, χ_{ij}^z, q_i be a feasible solution of the rate equation system for the unknowns x_{ij}^v, x_{ij}^w, x_{ij}^z , ρ_i, respectively. Then, if $x_{ji}^v > 0$ we have:

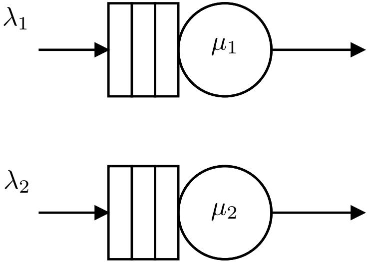
$$q_j/q_i = b_{ji}$$

In other words, if $b_{ji} = 1$ is a feasible solution in which a station i polls a station j with a positive rate (when empty) the load factor of the two stations are the same

Section 3

A simple example

Two parallel queues



- We analyse the solutions of the system of rate equations

The trivial solution

- $q_1^{(1)} = \lambda_1/\mu_1$
- $q_2^{(1)} = \lambda_2/\mu_2$
- and all the other unknowns set to 0.

Interpretation: No polling is performed, the systems remains unbalanced

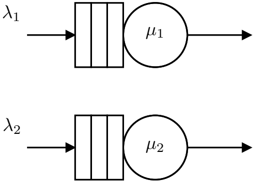
Solution for $b_{12} = b_{21} = 1$

- The system admits infinite non-trivial solutions that all give perfect load balance
- Without loss of generality, assume the unbalanced load factors $\overline{\rho}_2 > \overline{\rho}_1$
- We have that the perfect balance is obtained if the polling rates satisfy the following equation:

$$x_{12}^v = \frac{\lambda_2\mu_1 - \lambda_1\mu_2}{\lambda_1 + \lambda_2} + x_{21}^v$$

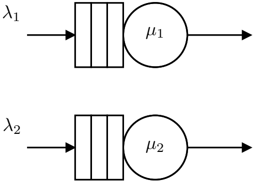
- As expected $\alpha_{12} = x_{12}^v > \alpha_{21} = x_{21}^v$
- Among the infinite solutions, in order to reduce the migration of jobs, one would choose the solution where $x_{21}^v = 0$

A numerical example



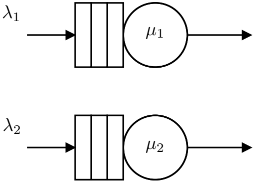
- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \Rightarrow S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

A numerical example



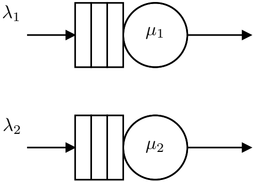
- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \Rightarrow S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

A numerical example



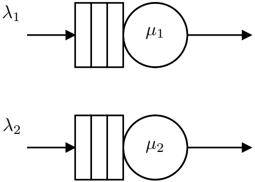
- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \Rightarrow S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

A numerical example



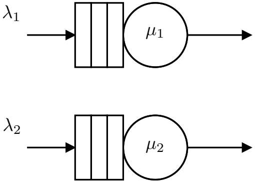
- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \Rightarrow S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

A numerical example



- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \ \Rightarrow \ S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

A numerical example

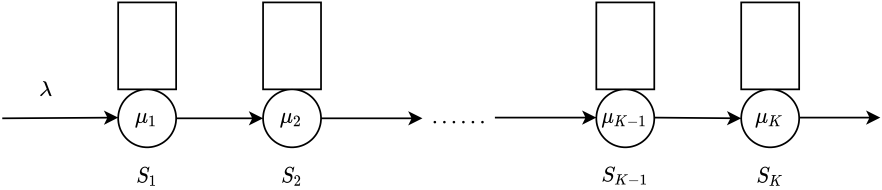


- $\lambda_1 = 0.8s^{-1}$, $\lambda_2 = 1.2s^{-1}$, $\mu_1 = 1s^{-1}$, $\mu_2 = 2s^{-1}$
- $\overline{\rho}_1 = 0.8$, $\overline{\rho}_2 = 0.6 \Rightarrow S_2$ polls S_1
- $\alpha_{21} = 0.2s^{-1}$ (polling rate)
- After load balancing $\rho_1 = \rho_2 = 2/3$
- No LB: Expected occupancy $\overline{N} = 5.5$, expected response time $\overline{R} = 2.75s$
- With LB: Expected occupancy $N = 4$, expected response time $R = 2s$

Section 4

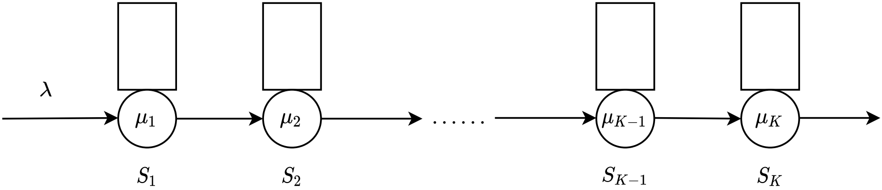
Fetching networks

Introduction



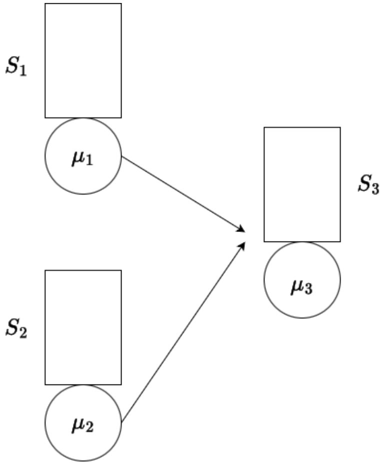
- If station S_i gets empty, it sends a signal to steal a job from S_{i-1}, $i > 1$
- This policy propagates backward until a non-empty station is found
- If the system is open, we can always fetch a job from the outside
- If the system is closed, thanks to the ergodicity of the routing process, we can always find a non-empty queue

Advantages



- In Jackson networks, the stability condition would be $\lambda < \min\{\mu_i\}$
- In this case, the stability condition is $\lambda < \mu_K$
- If we can order the stations $\mu_1 < \mu_2 < ... < \mu_K$, the stability condition is $\lambda < \max\{\mu_i\}$ and the expected number of stages experienced by a job is maximized

Who do I choose?



- The provider is chosen with a probability that is proportional to the arrival stream from that station
- If S_3 gets empty, it chooses S_1 as a provider with probability $\mu_1/(\mu_1 + \mu_2)$

The product-form result

Let $X(t)$ be the irreducible CTMC underlying a fetching queueing network, then its invariant measure has product-form:

$$\pi(\mathbf{n}) \propto \prod_{i=1}^{K} \rho_i^{n_i} ,$$

where:

$$\rho_i = \frac{\sum_{k=0}^{K} \mu_k p_{ki}}{\mu_i p_{i0} + \sum_{k=1}^{K} x_{ik}} ,$$

and x_{ik} is the solution the linear system of rate equations:

$$x_{ik} = \left(\sum_{j=1}^{K} x_{kj} + \mu_k p_{k0} \right) \frac{\mu_i p_{ik}}{\sum_{j=0}^{K} \mu_j p_{jk}} .$$

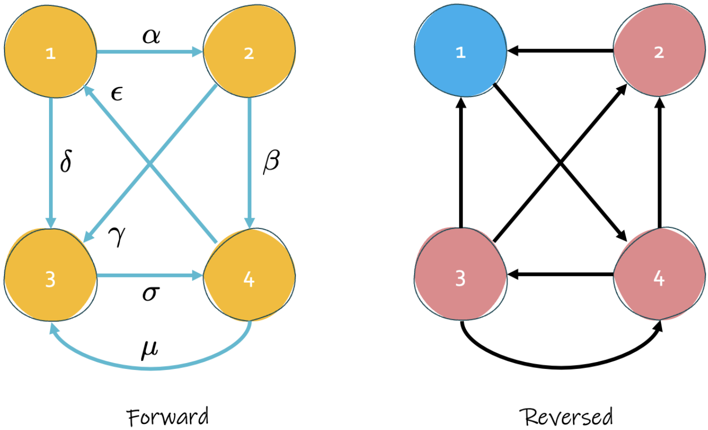
Section 5

On the proof technique

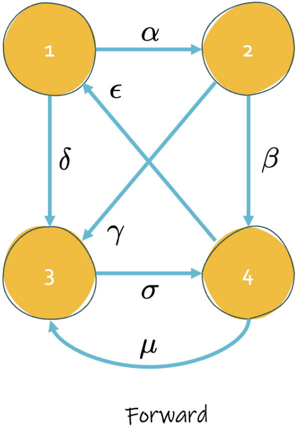
Reversing a process using Kolmogorov's criteria

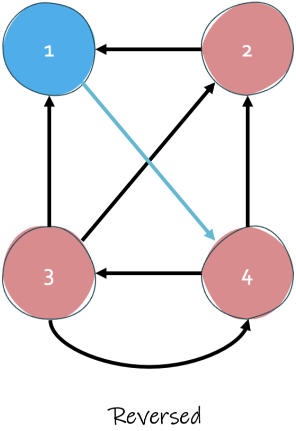
- Goal: Given a stationary, possibly non-reversible, CTMC $X(t)$ determine its reversed stationary chain $X^R(t)$
- The residence time in a state has the same distribution if the forward and reversed process
- For every cycle in the forward process, the product of the transition rates is the same of those present in the dual cycle of the reversed one
- In some cases, we can get the solution without solving the system of global balance equations

Reversing a process

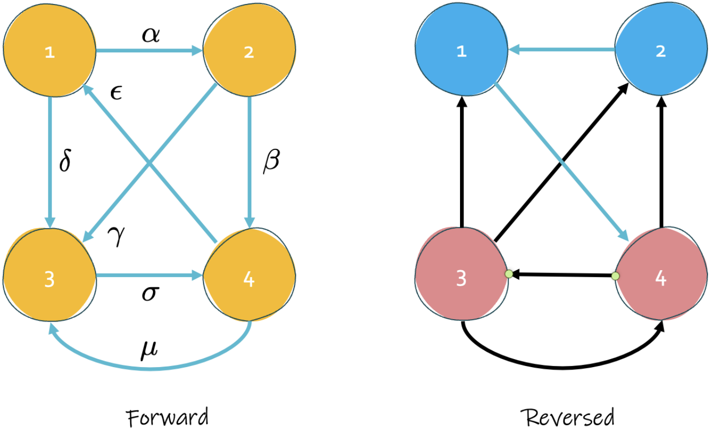


Reversing a process

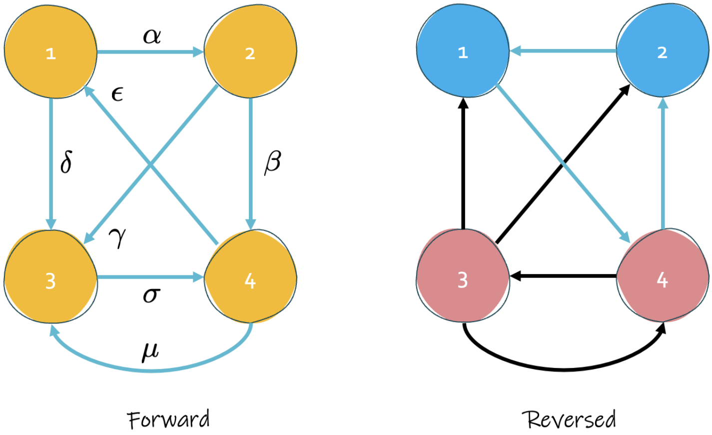




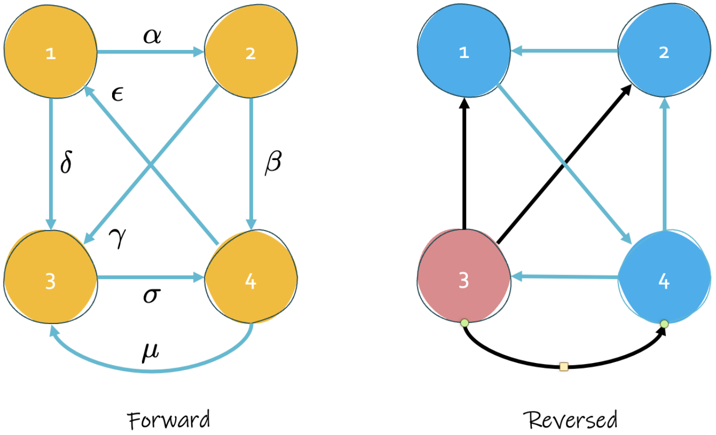
Reversing a process



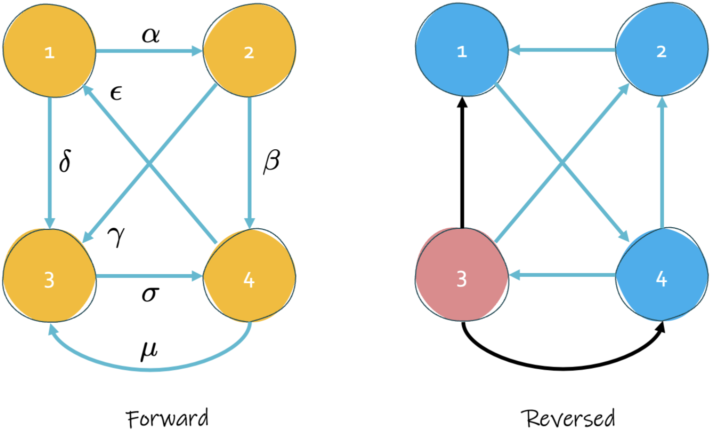
Reversing a process



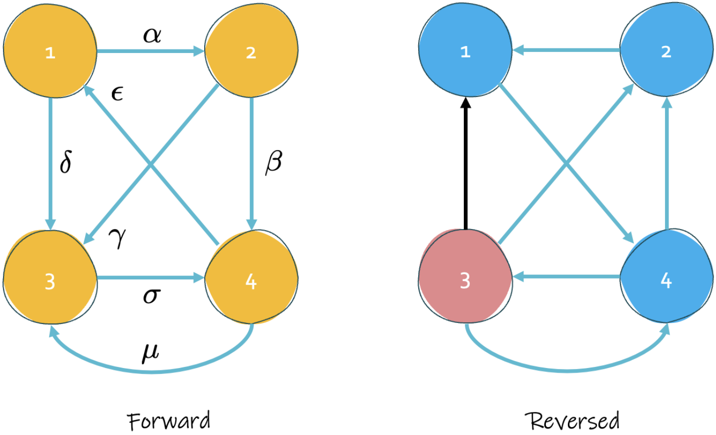
Reversing a process



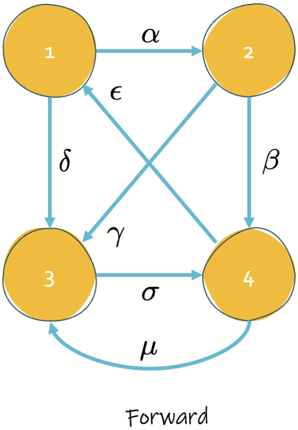
Reversing a process

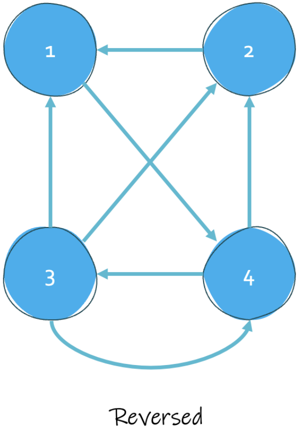


Reversing a process

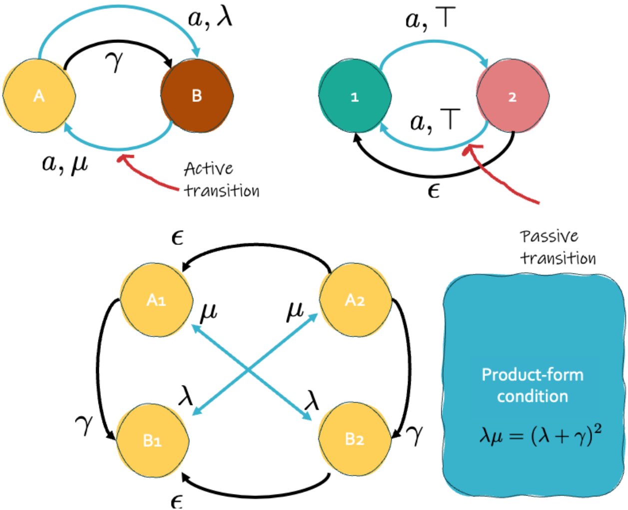


Reversing a process

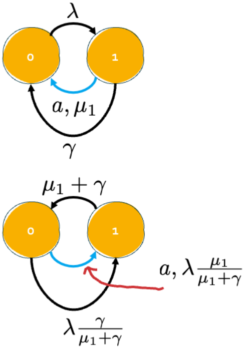




PEPA-like Synchronization (labels)



Reversing labelled agents



- This is a reversible process
- The transition rate from 0 to 1 is λ
- The transition rate from 1 to 0 is $\gamma + \mu_1$
- The reversed rate splits among the parallel transition proportionally to the forward rate

RCAT methodology (Harrison, 2002)

Conditions

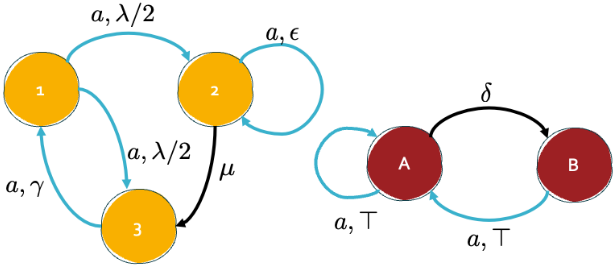
1. For each label a, if an agent is passive w.r.t. a, every state has an outgoing transition labelled a, named x_a
2. For each label a, if an agent is active w.r.t. a, every state has the same sum of reversed rates of the incoming transitions

Product-form: if the conditions are met, solve each process in isolation using the reversed rates x_a for each passive transitions a

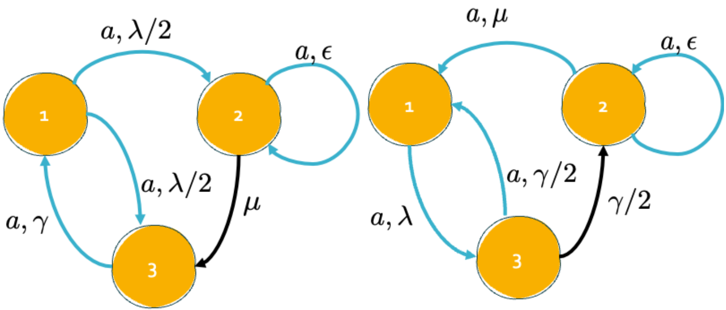
Tips:

- When verifying the second condition, assume the passive transitions have all the same rate
- Use Kolmogorov's criteria to simplify the reasoning

Find the product-form condition



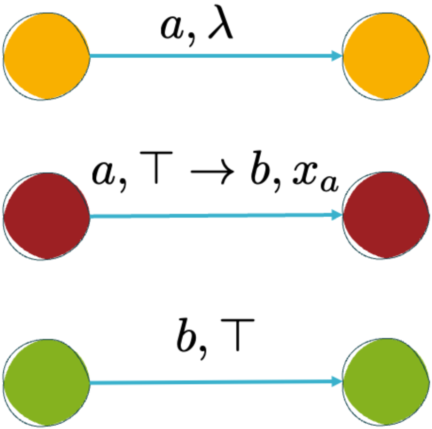
Find the product-form condition



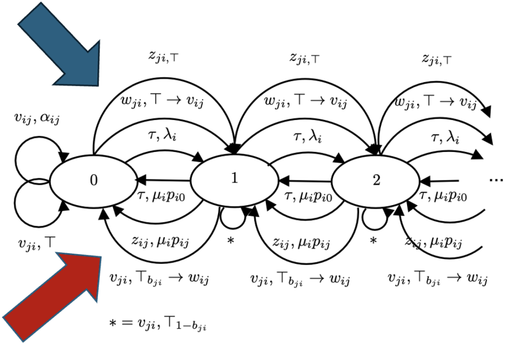
$$\lambda = \gamma/2 = \mu + \epsilon$$

Multiway synchronisations

Proposed in: *Peter G. Harrison, Andrea Marin: Product-Forms in Multi-Way Synchronizations. Comput. J. 57(11): 1693-1710 (2014)*



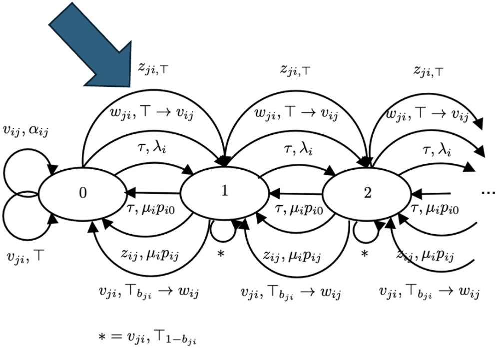
Proof of product-form for LB-networks



- for simplicity assume only station i and j
- Load factor:

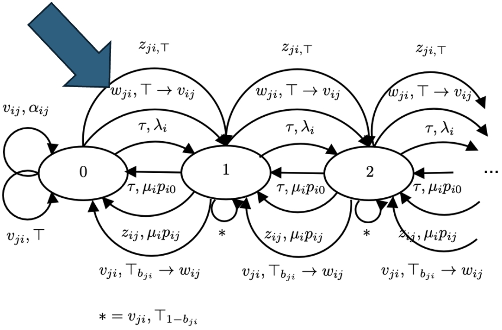
$$\rho_i = \frac{x^z_{ji} + x^w_{ji} + \lambda_i}{\mu_i + x^v_{ji} b_{ji}}$$

Proof of product-form for LB-networks



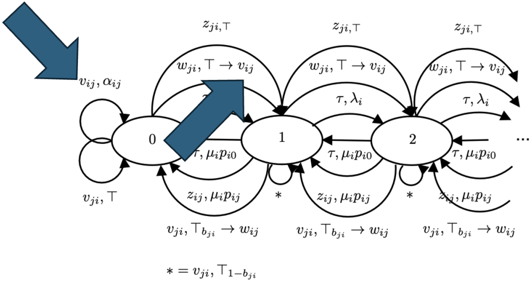
- z_{ij}: arrivals at station i from station j (passive)
- always enabled

Proof of product-form for LB-networks



- w_{ji}: arrivals at station i from migration from station j (passive)
- always enabled
- it propagates as v_{ij}, i.e., it requires another job if available

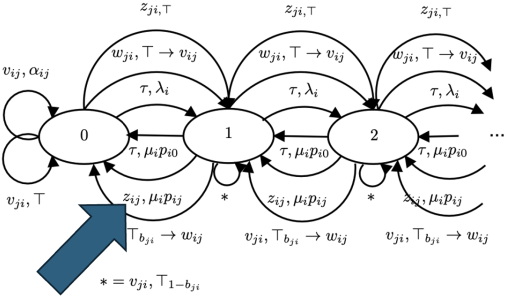
Proof of product-form for LB-networks



- v_{ij}: request for one job from i to j (active)
- reversed rate:

$$x^v_{ij} = \alpha_{ij} = \frac{x^w_{ji}}{\rho_i}$$

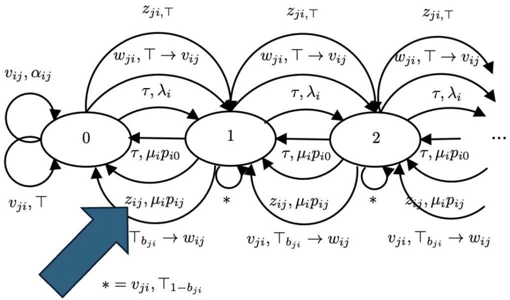
Proof of product-form for LB-networks



- z_{ij}: job completion at i moving to j (active)
- reversed rate:

$$x^z_{ij} = \mu_i p_{ij} \rho_i$$

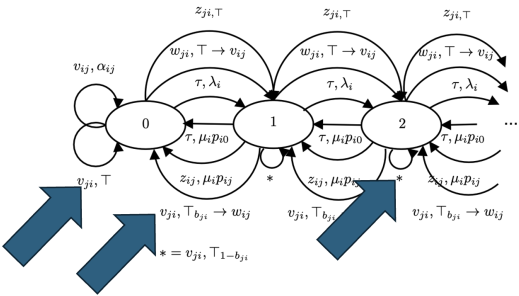
Proof of product-form for LB-networks



- z_{ij}: job completion at i moving to j (active)
- reversed rate:

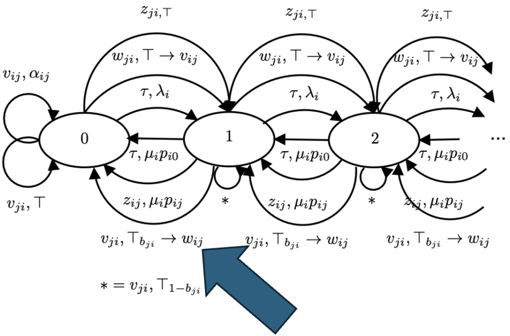
$$x^z_{ij} = \mu_i p_{ij} \rho_i$$

Proof of product-form for LB-networks



- v_{ji}: i receives a request of a job from j and decides to send it with probability b_{ji} (triggers w_{ij}) if any is present. If empty, it does not send anything. (passive)
- always enabled

Proof of product-form for LB-networks



- w_{ij}: i sends a job to j because of a request. (active)
- reversed rate:

$$x_{ij}^{w} = x_{ji}^{v} b_{ji} \rho_i$$

Section 6

Final remarks

Final remarks

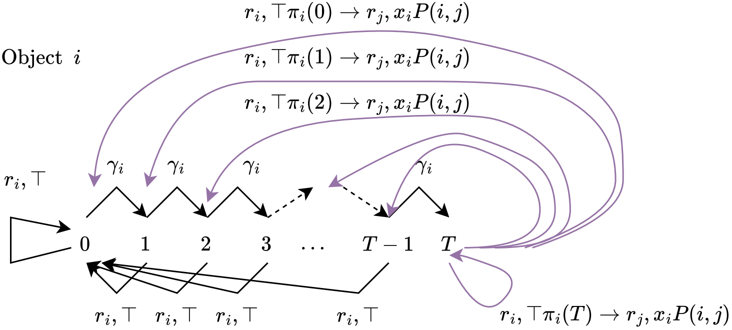
- Both queueing network models can be studied with Mean-Value approaches or convolution
- The MVA algorithms are obtained algebraically of thanks to the duality relation for fetching QNs
- RCAT proofs are very modular once you can model appropriately the interaction among the components
- Open problems: give formulations and proofs of the arrival theorems for LB-networks or fetching networks

THANKS FOR THE ATTENTION

A caching system

- N objects
- The object is live when the state is $0, \ldots, T-1$, it is evicted when state is T
- Object i state n_i represents the age of the object
- The object ages from state n_i to $n_i + 1$ with rate γ_i, for $n_i < T$
- When an object receives a request:
  - If there is a hit then the state (age) is reset to 0
  - If there is a miss, the request is forwarded to another object j with probability $P(i,j)$ (in another cache) and the age of this newly retried object is set to the steady-state distribution of object i

Process algebraic representation



Product-form result

The model with N objects has product-form steady-state distribution

$$\pi(n_1, \ldots, n_N) = \prod_{i=1}^{N} g_i(n_i)\,. \tag{1}$$

where:

- $$g_i(n_i) = \begin{cases} (1 - z_i) z_i^{n_i} & \text{if } 0 \le n_i < T \\ z_i^T & \text{if } n_i = T \end{cases},$$
- z_i is the minimum real root such that $0 \le z_i \le 1$ of the polynomial:
$$P_i(z) = x_i z^{T+1} - (x_i + \gamma_i) z + \gamma_i\,, \tag{2}$$
- x_i is the solution of the following system of rate equations:
$$x_i = \lambda_i + \sum_{\substack{j=1 \\ j \neq i}}^{N} z_j^T x_j P(j, i)\,. \tag{3}$$