Imperial College London

Department of Computing

Response time probability densities

Author:
Julianna Bor

Supervisor:
Peter Harrison

Submitted in partial fulfillment of the requirements for the MSc degree in Computing Science of Imperial College London

September 2017

Abstract

A Markovian queue with both batch arrivals and batch departures is shown to have a geometric equilibrium queue length probability distribution after adding special arrivals and departures to the system. From this, a similar result is obtained for a network of batch queues by the use of the reversed compound agent theorem. Using the generating function method, the Laplace-Stieltjes transform of the sojourn time random variable in the case of a tandem network of batch-queues is obtained. This result is derived by first considering the marginal sojourn time at the second queue and solving a recurrence for a vector of generating functions rather than a single generating function. Then the marginal sojourn time at the second queue and the reversed sojourn time at the first queue are considered jointly; however since those are only conditionally independent, a complex integral must be evaluated in order to pick out the desired coefficients.

Acknowledgments

I am grateful to my supervisor, Peter Harrison for introducing me to this topic and helping me throughout my work.

Contents

Chapter 1

Introduction

Performance modeling is an essential part of the design, optimization and development of communication and computer systems. Due to rapid technological development allowing faster transmission and bigger capacity, networks have become more and more complex, hence requiring a more sophisticated approach to understand their behavior.

There are two main types of model available for this purpose: simulation and stochastic. We focus on the latter. Communication and computer networks often evolve in time and stochastic processes are suitable for modeling such evolving systems by describing the states they enter.

We are using a queue-based model which is a common approach to describe systems where there is contention for some resource. Good examples of the possible applications of a queue-based model can be found in [2], where an M/M/1 discriminatory processor sharing queue is used to investigate performance-energy trade-offs of smartphone applications, and in [9], where the usage of replicas is considered in order to reduce the response time and improve fault tolerance in a network.

One of the many performance measures one could use is the response time. It is particularly important since it describes the waiting time users or participants face when using the system. Obtaining the response time with a queuing model has been used, for example, to minimize the waiting time in an emergency department [4, 3] or to optimize the scheduling of police cars [14]. However, obtaining the mean response time or even moments of the response time may not provide enough information. For instance, in a multi-user system a mean response time of one minute can be tolerable provided that the standard deviation is not too large, while a response time with a mean of five seconds which occasionally exceeds five minutes will probably lead to dissatisfied users. Furthermore, international standards for transaction processing (e.g TPC benchmarks) now include requirements for the 90% and 95% quantiles. That is, the given criterion – such as a response time of less then three seconds – has to be met 95% of the time. Hence, the probability distribution of the response time is required, which is significantly more expensive and difficult to obtain. In cases when such densities can be calculated, one often gets the Laplace transform of the density as a result which then needs to be inverted numerically.

In this thesis, Markovian batch queues and a path of two Markovian batch-queues in tandem are considered. The aim is to obtain the distribution of the sojourn time random variable corresponding to the whole network. Sojourn times in a tandem batch-network have been obtained in special cases in [1, 12] but the result for the general case appears to be novel.

First, we describe the conditions under which a Markovian batch queue has a geometric queue length probability distribution at equilibrium. Then we extend the theorem for a

network of queues using the reversed compound agent theorem (RCAT). These results can be found in [7]. A product form geometric equilibrium distribution makes it possible to use the generating function method to calculate the Laplace-Stieltjes transforms (LST) of the response time distributions.
Next, we calculate the LST of the response time distribution of both the normal and reversed process of a single queue. The idea for the tandem-network is to move the origin of the time-line to the instant at which the tagged task's batch leaves the first queue and then to consider the joint probability of the reversed process of the first queue back in time, and the normal process of the second queue forward in time, conditioned on the state at the origin. This is done by first calculating the LST of the marginal distribution of the response time at the second queue using the generating function technique for a vector of generating functions. Up to this point Harrison [5] is followed closely.
In order to obtain the LST of the joint distribution we would like to combine the results for the reversed process in the case of the first queue and the LST of the forward sojourn time at the second queue.
In the general case, the path is not overtake-free since the sojourn time of the tagged task at the second queue may be affected by later arrivals from the first queue. This means that, the sojourn time random variables are only conditionally independent; therefore using the generating function to obtain the unconditional probabilities is no longer straightforward.
Before addressing the difficulties caused by this conditional independence for the general problem, two special cases are considered, namely, the tandem network of two M/M/1 queues and a tandem network where only the first queue is M/M/1 and the second one is a general batch queue.
My most significant contribution to Harrison's work is to overcome the aforementioned difficulty in the general case by taking a complex integral in order to pick out specific coefficients from the generating function and therefore to obtain the LST of the unconditional joint probability distribution.
Finally, numerical results obtained by inverting the calculated LSTs are shown, all generated by Mathematica, and potential areas of application are outlined.

Chapter 2

Background

In this chapter, we cover the mathematical background material necessary for being able to state and prove results about sojourn times in tandem networks. Starting from the basic definitions and theorems of probability theory, we then move on to stochastic processes and finally introduce the notations, definitions and selected results of queuing theory. The material is based on the book: Performance modelling of communication and networks and computer architectures [8].

2.1 Basic mathematical background

This section provides a quick overview of the most common definitions and theorems of probability theory used throughout the thesis.

Definition 2.1.1 (Event space) $\mathcal{E} \subseteq 2^{\Omega}$ *is an event space if*

1. $\Omega \in \mathcal{E}$
2. $E \in \mathcal{E} \Rightarrow \overline{E} \in \mathcal{E}$
3. $E_i \in \mathcal{E} \;\; i = 1, 2 ... \Rightarrow \bigcup\limits_{i=1}^{\infty} E_i \in \mathcal{E}$

Definition 2.1.2 (Probablity measure) $P : \mathcal{E} \to \mathbb{R}$ *is a probability measure if*

1. $0 \leq P(E) \leq 1 \;\; \forall E \in \mathcal{E}$
2. $P(\Omega) = 1$
3. $P\left(\bigcup\limits_{i=1}^{\infty} E_i\right) = \sum\limits_{i=1}^{\infty} P(E_i); \;\; E_1, E_2 ... \in \mathcal{E} : \;\; E_i \cap E_j = \varnothing \;\; \forall i \neq j$

Proposition 2.1.1 (Law of total probability) $P(A) = \sum P(A|B_i)P(B_i)$ *where* $\{B_1, B_2, ...\}$ *is a partition of* A*.*

Definition 2.1.3 (Random variable) $X : \Omega \to \mathbb{R}, \;\; P(X \in I) = P(X^{-1}(I))$ *where* Ω *is the sample space and I is an interval. We assume that* $X^{-1}(I) \in \mathcal{E} \subseteq 2^{\Omega} \;\; \forall I \subset \mathbb{R}$*.*

Definition 2.1.4 (Distribution function) $F(X) : \mathbb{R} \to [0, 1], F(x) = P(X \leq x)$*.*

Definition 2.1.5 (Density function) *$f(x) : \mathbb{R} \to \mathbb{R}$ When X is continuous $f(x)$ the function for which: $P(X \in S) = \int\limits_S f(t)dt$. So in this case $F(x) = \int\limits_{-\infty}^{x} f(t)dt$. A necessary and sufficient condition for the existence of f is F to be everywhere differentiable.*

Proposition 2.1.2 *Properties of the distribution function:*

1. $x \leq y \Rightarrow F(x) \leq F(y)$
2. $\lim_{x\to\infty} F(x) = 1$
3. $\lim_{x\to-\infty} F(x) = 0$
4. *F is right continuous*

Definition 2.1.6 (Independence) *X and Y are independent if any of $1-3$ holds:*

1. $F(x, y) = F_X(x)F_Y(y) \ \forall x, y \in \mathbb{R}$
2. $f(x, y) = f_X(x)f_Y(y) \ \forall x, y \in \mathbb{R}$
3. $P(X \in A, Y \in B) = P(X \in A)P(Y \in B) \ \forall A, B \in \mathcal{E}$

Definition 2.1.7 (Expected value) *The expected value of a random variable is defined as follows:*
Discrete case: $E(X) = \sum\limits_{i=0}^{\infty} iP(X = i)$ where $X : \mathcal{E} \to \mathbb{Z}_0^+$
Continuous case: $E(X) = \int\limits_{-\infty}^{\infty} xdF(x) = \int\limits_{-\infty}^{\infty} xf(x)dx$
Furthermore, given two independent random variables: $E(XY) = E(X)E(Y)$

Definition 2.1.8 *Commonly occurring discrete and continuous random variables and their expectation:*

1. ***Geometric**: $X \sim$ Geo(p) then $P(X = i) = (1-p)p^i$, $E(X) = \frac{p}{1-p}$*
2. ***Poisson**: $X \sim$ Poisson(λ) then $P(X = i) = e^{-\lambda}\frac{\lambda^i}{i!}$, $E(X) = \lambda$*
3. ***Exponential**: $X \sim$ Exp(λ) then $f(x) = \lambda e^{-\lambda x}\mathbb{I}_{\{x\geq 0\}}$, $E(X) = \frac{1}{\lambda}$*
4. ***Erlang-n**: $X \sim$ Erlang-n(λ) then $f(x) = \frac{\lambda(\lambda x)^{n-1}e^{-\lambda x}}{(n-1)!}$, $E(X) = \frac{n}{\lambda}$*

Definition 2.1.9 (Conditional expectation) *$(\Omega, \mathcal{F}, P)$ probability space, $X : \Omega \to \mathbb{R}^n$, let $\mathcal{H} \subseteq \mathcal{F}$ be a σ-algebra. Then $E(X|\mathcal{H}) : \Omega \to \mathbb{R}^n$ is $\mathcal{H}$-measurable and satisfies the following equation $\forall H \in \mathcal{H}$*

$$\int\limits_H E(X|\mathcal{H})dP = \int\limits_H XdP$$

Or equally let $F_{X|Y}(x|y) = P(X \leq x|Y = y)$ be the conditional distribution, then
$E(X|Y = y) = \int\limits_{-\infty}^{\infty} xdF_{X|Y}(x|y) = \int\limits_{-\infty}^{\infty} xf_{X|Y}(x|y)dx$ where $f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)}$, $f_Y > 0$.

Proposition 2.1.3 *Basic properties of the conditional expectation:*

1. *if X and $\mathcal{H}$ are independent then $E(X|\mathcal{H}) = E(X)$*
2. *if X is independent of $\sigma(Y, \mathcal{H})$ then $E(XY|\mathcal{H}) = E(X)E(Y|\mathcal{H})$*
3. *if X is $\mathcal{H}$-measurable then $E(X|\mathcal{H}) = X$*
4. *if X is $\mathcal{H}$-measurable then $E(XY|\mathcal{H}) = XE(Y|\mathcal{H})$*
5. *Law of total expectation: $E(E(X|\mathcal{H})) = E(X)$*
6. *if $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{F}$ then $E(E(X|\mathcal{H}_2)\mathcal{H}_1) = E(X|\mathcal{H}_1)$*
7. *linearity: $E(aX + Y|\mathcal{H}) = aE(X|\mathcal{H}) + E(Y|\mathcal{H}), a \in \mathbb{R}$*
8. *positivity: if $X \geq 0$ then $E(X|\mathcal{H}) \geq 0$*

Point 5 above is used quite frequently backwards, meaning that, given a random variable X instead of calculating $E(X)$ directly we condition on another random variable and calculate the conditional expectation. For example X can be the number of arrivals during a service period of a queue and conditioning on the service time random variable makes it easier to obtain the expected value of it.

Definition 2.1.10 (Probability generating function) $\pi(z) = E(z^X)$. *In the discrete case, that is $\pi(z) = \sum\limits_{i=0}^{\infty} p_i z^i$ using the definition of expectation.*

As mentioned before, we usually obtain the Laplace transform of the density we are looking for, which then needs to be inverted. Inversion of the Laplace transform is in itself challenging, since it usually cannot be inverted analytically and numeric methods are often unstable or not precise enough, especially in the tail region.

Definition 2.1.11 (Laplace transform) *Let $F : \mathbb{R}_0^+ \to \mathbb{R}$, then the Laplace transform of F is a function itself $\mathcal{L}\{F\} : \mathbb{C} \to \mathbb{C}$*
$\mathcal{L}\{F\}(s) = \int\limits_0^{\infty} e^{-st} dF(t)$.
Using the definition of the expected value of a random variable X, if F is its distribution function and f is its density functions, then we can also write: $\mathcal{L}\{F\}(s) = E(e^{-sX}) = \int\limits_0^{\infty} e^{-st} f(t) dt$

Proposition 2.1.4 (Inversion of the Laplace transform) *If there are two functions with the same Laplace transform, they can only differ on a set which has Lebesgue measure zero. Hence, the inverse function is uniquely determined Lebesgue almost everywhere.*

Proposition 2.1.5 *Basic properties of the Laplace transform.*

1. ***linearity**: $\mathcal{L}\{af_1 + bf_2\} = a\mathcal{L}\{f_1\} + b\mathcal{L}\{f_2\}$*
2. ***derivative**: if f is differentiable, its Laplace transform is L and f' is the derivative of f and is of exponential order then the Laplace transform of $f^{'}$ is $sL(s) - f(0)$*
3. ***integral**: the Laplace transform of $\int\limits_0^t f(\tau) d\tau$ is $\frac{1}{s}F(s)$*
4. ***convolution**: the Laplace transform of $(f * g)(t) = \int\limits_0^t f(\tau)g(\tau - t) d\tau$ is $F(s)G(s)$ where F and G are the Laplace transforms of f and g, respectively.*

2.2 Stochastic processes

As stated before stochastic processes are suitable to model communication and computer networks mathematically due to their ability to describe an evolving system. Let us first define precisely a stochastic process.

Definition 2.2.1 (Stochastic process) *A family of random variables $\{X_t | t \in T\}$, where T is the parameter space and each X_t takes its values from some set S called the state space. Values of T are often referred to as times.*

Definition 2.2.2 $\{X_t\}$ *is **stationary** if*
$\forall t_1, t_2, ...t_n, t_1 + \tau, t_2 + \tau, ...t_n + \tau \in T, n \geq 1 : F_{X_{t_1}...X_{t_n}} = F_{X_{t_1+\tau}...X_{t_n+\tau}}$

Definition 2.2.3 $\{X_t\}$ *has **independent increments** if*
$\forall t_1 < t_2 < ... < t_n \in T, n \geq 1 : X_{t_1}, X_{t_2} - X_{t_1}, ..., X_{t_n} - X_{t_{n-1}}$ *are independent*

Definition 2.2.4 $\{X_t\}$ *has the **Markov property** if*
$\forall \tau_1 < \tau_2 < ... < \tau_m < t_1 < t_2 < ... < t_n \in T, n, m \geq 1$:
$F_{X_{t_1}...X_{t_n}|X_{\tau_1}...X_{\tau_m}}(x_1, .., x_n | y_1, .., y_m) = F_{X_{t_1}...X_{t_n}|X_{\tau_m}}(x_1, .., x_n | y_m)$

The Markov property means, at any given time t , the future will only depend on the state at t and not on the previous history before t.

2.2.1 Markov chains

Definition 2.2.5 (Markov chain) *A Markov chain is a discrete time stochastic process with a countable sample space which has the Markov property.*

Definition 2.2.6 (Time homogeneity) *A Markov chain is time homogeneous if*
$P(X_t = j | X_0 = i) = P(X_{t+\tau} = j | X_\tau = i), \forall \tau > 0; i, j \in S$.

Time homogeneity means that, given that the process is in state i at some point, the probability that it enters state j some t time after that, only depends on the elapsed time and not the time instant we started measuring from.
We will only consider time homogeneous Markov chains.

Definition 2.2.7 (Transition probabilities) *The n-step transition probabilities of a time homogeneous Markov chain is:* $p_{ij}^{(n)} = P(X_{m+n} = j | X_m = i)$.
When $n = 1$ *we use the notation:* p_{ij}

A Markov chain is characterized entirely by its one-step transitions.

Definition 2.2.8 (Transition matrix) $P = [p_{ij}]$ *which is a stochastic matrix.*

Proposition 2.2.1 (Chapman-Kolmogorov equation) *Let P^n, P^m be the n and m step transition matrices, respectively. Then $P^{m+n} = P^m P^n$.*

Definition 2.2.9 *Let T_{ii} be the first return time to i if $X_0 = i$.*

1. *i is a **transient** state if $P(T_{ii} < \infty) < 1$*
2. *i is a **null-recurrent** state if $P(T_{ii} < \infty) = 1$ but $E(T_{ii}) = \infty$*

3. *i is a **positive-recurrent** state if $P(T_{ii} < \infty) = 1$ and $E(T_{ii}) < \infty$*

So a positive recurrent state returns infinitely often with a finite expected recurrence time.

Definition 2.2.10 (Steady state) *If $\forall j : p_{ij}^{(n)}$ has a limit as $n \to \infty$ which is independent of i, then the chain is said to have a steady state or a state of **equilibrium**.*

When modeling a real-world system we usually interested in its behavior at equilibrium if there exists a steady state at all.
Probably the most important related theorem is Kolmogorov's theorem. In order to be able to state the theorem, two additional terms must be introduced.
We call a chain **irreducible** if the probability of getting from any state to any other state in n steps is positive for some n. In other words, it is possible to enter any state after some time from any given time t.
We call a chain **aperiodic** if the m step transition probabilities are positive for any m with finitely many exceptions.

Proposition 2.2.2 (Kolmogorov's theorem) *In an irreducible, aperiodic Markov chain:*

1. $p_{ij}^{(n)} \to \frac{1}{E(T_{ii})}$ *as* $n \to \infty$
2. *The chain is positive recurrent* $\Leftrightarrow \exists! \pi$ *stationary distribution, such that* $\pi = \pi P$ *with* $\pi_j = \frac{1}{E(T_{jj})}$

In practice, we can search for a solution of the equation $\pi = \pi P$ which always exists and then normalize it, so it sums up to 1 which is a necessary and sufficient condition for π to be a distribution. Therefore, using the theorem above, a nonzero normalizing constant is a necessary condition for the chain to be positive-recurrent.

2.2.2 Markov processes

Definition 2.2.11 *A stochastic process with continuous parameter space and discrete state space which has the Markov property is called Markov process.*

There is a special case of a Markov process which has great importance since it is the simplest, most mathematically well-behaved example of a Markov process. It also commonly occurs in real-world systems. This is the **Poisson process**.

Definition 2.2.12 (Poisson process) *Let $N_{t\tau}$ denote the number of arrivals in the time period $(t, \tau]$. $\{N_{0t} | t \geq 0\}$ is a Poisson process if $\forall t, h \geq 0$*

1. *it is time-homogeneous*
2. *it has independent increments*
3. *it is orderly:* $\frac{P(N_{t,t+h} \geq 2)}{h} \to 0$ *as* $h \to 0$

Or equivalently: $\exists \lambda > 0, \forall t, h \geq 0$

a $P(N_{t,t+h} = 0) = 1 - \lambda h + o(h)$

b $P(N_{t,t+h} = 1) = \lambda h + o(h)$

c $P(N_{t,t+h} \geq 2) = o(h)$

d $N_{0t}, N_{t,t+h}$ *are independent*

It is called a Poisson process since N_{0t} is a Poisson variable.
The equivalence of the above two definitions can be proven easily, so they can be used interchangeably. This property is used often in practice.

Proposition 2.2.3 *The two definitions above are equivalent.*

Sketch of the proof: First note that: $3. \Leftrightarrow (c)$ and $2. \Leftrightarrow (d)$. Furthermore, (b) is implied by (a) and (c). Since t is not present on the right sides of $(a), (b)$ and (c), time-homogeneity must follow. It only remains to prove that (a) follows from the first definition. For that we use the fact that $P(N_{0,s+t} = 0) = P(N_{0,s} = 0)P(N_{s,t} = 0)$ which follows from (1) and (2).

Definition 2.2.13 (Memoryless distribution) $P(T > t + \tau | T > \tau) = P(T > t)$

Proposition 2.2.4 *A continuous/discrete random variable is memoryless $\Leftrightarrow$ it is an exponential/geometric random variable*

One of the nice properties of a Poisson process is that the time to the first arrival and the time between two successive arrivals are both exponential variables. So by the memoryless property of the exponential random variable, no matter how much time has passed since the last arrival, the probability that the next arrival happens t time from now is still exponential with the same parameter. In other words, the forward recurrence time is also exponential and it is independent of past history.

Proposition 2.2.5 (Random observer property) *Let $\{X_t | t \geq 0\}$ be a process with Poisson arrivals. If there is an arrival at τ the state changes from X_{τ^-} to X_{τ^+}. (For example X_{τ^-} can be the length of a queue before and X_{τ^+} after the arrival of a new task.) The random observer property states that the distribution of X_{τ^-} is independent of the event that there is an arrival at τ. So the arriving task has the same information about the state that a random observer would have.*

The random observer property allows us, for example, to look at the length of the queue of waiting tasks at a server from the arriving job's perspective but still use the steady state probabilities of the queue length.
Now returning back to Markov processes in general, we use the same notations that we used for Markov chains. As before we only consider time homogeneous processes. We further assume the transition matrix $P(t) = [P_{ij}(t)]$ satisfies the following conditions:

1. $\forall t \in [0, \infty) : P(t)$ is stochastic
2. $P(t)$ is right continuous at $t = 0$
3. $P(s + t) = P(s)P(t)$

The third point means the Chapman-Kolmogorov equation has to hold, which is a consequence of the Markov property as well.
Now, in order to be able to use a steady state result similar to Kolmogorov's theorem in the discrete case, generators of a Markov process need to be introduced.

Definition 2.2.14 (State holding time) *If the process enters the state i at time t and the next transition is at $t+T$ then we call T the state holding time of state i.*

Proposition 2.2.6 *Denoting the state holding time with T in state i, it is exponential with parameter q_i. We call this the transition rate from state i.*

Definition 2.2.15 *We say that i is:*

1. ***stable*** *if* $0 \leq q_i < \infty$
2. ***instantaneous*** *if* $q_i = \infty$
3. ***absorbing*** *if* $q_i = 0$

Proposition 2.2.7 (Instantaneous transition rate matrix) *Let $Q = [q_{ij}]$ be a matrix for which $P(h) = I + Qh + o(h)$ so $Q = P^{'}(0)$. Such Q always exists due to the Chapman-Kolmogorov equation. It can be shown that $P(X_{t+h} = j|X_t = i) = q_{ij}h + o(h)$ $\forall i \neq j$ and $-q_{ii} = \sum\limits_{j \in S} q_{ij} = q_i$.*

In other words, q_{ij} is the instantaneous transition rate from state i to state j, and summing for all j we get the instantaneous transition rate from state i.
Q is also called the **generator** of the Markov process since it determines the evolution of the process entirely with the following equations:

Proposition 2.2.8 *For every Markov process with the previously defined conditions:*

1. ***Forward Kolmogorov equation****:* $\exists M < \infty$ $\forall i : q_i < M \Rightarrow P^{'}(t) = P(t)Q$
2. ***Backward Kolmogorov equation****:* $\forall i : q_i < \infty \Rightarrow P^{'}(t) = QP(t)$

Given a Markov process we can define a Markov chain by looking at the state right after each transition. We call it the **embedded Markov chain**. The precise definition is as follows:

Definition 2.2.16 (Embedded Markov Chain) *Let $\{X_t | t \geq 0\}$ be a Markov process, and let $\tau_1, \tau_2, ...$ denote the time instants at which the state changes. Then the chain defined as $\{Z_n | n = 0, 1, ...\}$ where $Z_0 = X_0$ and $Z_n = X_{\tau_n^+}$ has the Markov property and it is called the embedded Markov chain.*

We can use the properties defined earlier for Markov chains such as *irreducible, transient, null-recurrent and positive-recurrent* for Markov processes in the sense that the related EMC has that property.

Proposition 2.2.9 (Steady state result) *An irreducible Markov process with generator matrix Q is*

1. *transient or null-recurrent $\Rightarrow \pi = 0$, so there is no steady state*
2. *positive recurrent $\Leftrightarrow \pi Q = 0$ has a non-zero solution*

This is a similar steady state result to Kolmogorov's theorem. Using this result after some transformation we get the **balance equations**:

$$\sum_{j \neq i} \pi_j q_{ji} = -\pi_i q_{ii} = \sum_{j \neq i} \pi_i q_{ij}$$

Analyzing the above equation further, and writing it down for an arbitrary set of states A:

$$\sum_{i \in A} \sum_{j \neq i} \pi_j q_{ji} = \sum_{i \in A} \sum_{j \neq i} \pi_i q_{ij}$$

Note that all the pairs of states where both states are in A appear on both sides. Hence, we can omit them and obtain the **aggregated balance equations**:

$$\sum_{i \in A} \sum_{j \notin A} \pi_j q_{ji} = \sum_{i \in A} \sum_{j \notin A} \pi_i q_{ij}$$

If we look at this as a transition graph where we map the states to nodes and $\pi_i q_{ij}$ is the weight of the directed arc between node i and j, the above equation states that for an arbitrary set of nodes, the sum of all the weights corresponding to the arcs entering the set is equal to the sum of the weights corresponding to those leaving the set.
Choosing an appropriate set of states, this result can significantly simplify the equations we have to solve in order to find the steady state solution.

2.3 Queues

Using queues to model communication and other systems is quite common, because queues represent contention for some resource.
The resource subject to contention can be a police car awaited by a crime scene [14], doctors in an emergency department [4] or a disk in a storage system completing I/O requests [15].
This section briefly introduces the basic notations used, lists a few queuing disciplines and states Little's theorem, which is a simple but fundamental result in this area.

2.3.1 Basic terms and Little's result

A queue consists of three parts:

1. an **arrival process** which describes when customers arrive
2. the **queue** itself or, in other words, the waiting room where they wait to be served
3. the **service time** requirement for each customer

Queues are classified according to **Kendall's notation**: A/S/m.
Here **A** describes the arrival process, for example $A = M$ stands for Markovian, $A = G$ means general and $A = D$ deterministic. **S** is the service time distribution; again we use M for Markovian (that is, one with exponential distribution), and D for deterministic. Finally **m** denotes the number of servers serving the queue.

Queuing disciplines

1. **First come first served** (FCFS)

2. **Last come first served** (LCFS)
   (a) non-preemptive: the job currently being served is finished before a newly arrived task takes its place.
   (b) preemptive: a new job arriving to the queue immediately starts its serving time. They can be either preemptive resume or preemptive restart. In the former one the task which has been replaced will continue where it was left off while in the latter case it will start from the beginning again.
3. **Processor Sharing** (PS) Given a processor sharing discipline, the service capacity is equally divided amongst all the customers in the queue, so there is no real queuing. The rate at which a task is being served decreases as the number of the waiting tasks increases. Letting μ be the service rate of the server and n be the number of waiting customers, each customer sees an instantaneous service rate of $\frac{\mu}{n}$.

Proposition 2.3.1 (Little's result) *Suppose a queuing system which is in equilibrium. Let λ be the mean arrival rate, L be the mean number of tasks in the system and W be the mean time spent by a task in the system. The following equation holds:*

$$L = \lambda W$$

2.3.2 Simple Markovian queues: M/M/1

A simple Markovian queue is a **birth-death process**: a one-step transition can only change the current state by one. In this case either a task arrives at the queue or there is a departure when a task has completed its service. A Markovian queue has arrival rate $\lambda(n)$, and service rate $\mu(n)$, where n denotes the length of the queue in both cases.

Using the aggregated balance equations stated at the end of the previous section, we can find the steady state solution for the queue length. Solving the aggregated balance equations, we get:

$$\pi(j) = \frac{\rho_j}{\sum\limits_{k=0}^{\infty} \rho_k} \text{ where } \rho_0 = 1 \text{ and } \rho_j = \prod_{k=1}^{j} \frac{\lambda(k-1)}{\mu(k)} \text{ for } j > 0$$

Furthermore, π is a probability distribution if and only if $\sum\limits_{k=0}^{\infty} \rho_k < \infty$. In the case that π is a probability distribution, it is the unique stationary distribution of the Markov process which is positive recurrent.
Note that when the arrival and service rates are constant, – therefore, they do not depend on the queue length – this simplifies to $\pi(n) = (1-\rho)\rho^n$ where $\rho = \frac{\lambda}{\mu}$ which is a geometric random variable with parameter ρ.

2.3.3 Response time distribution in an M/M/1 queue

We now consider deterministic arrival (λ) and service (μ) rates. In an $M/M/1$ queue the mean waiting time can be obtained easily using Little's result as follows.
The mean arrival rate is λ and the mean queue length in equilibrium is $\frac{\rho}{1-\rho}$ which can be easily calculated since the distribution of the queue length is geometric as obtained above. Using Little's theorem we have $W = \frac{L}{\lambda} = \frac{1}{\mu-\lambda}$. This holds for all queuing disciplines.

As mentioned in the introduction, the mean response time does not give enough information about the waiting time in most cases. Therefore we focus on the LST of distribution of the waiting time from now on.
We give a proper proof in the case of an FCFS discipline, since this is used later in this thesis, and the outline of the proof for the PS discipline.

Proposition 2.3.2 (Waiting time distribution using FCFS discipline) *Consider an M/M/1 queue with FCFS discipline, and fixed arrival and services rates λ and μ , respectively. In this case, the generating function of the Laplace-Stieltjes transform of the waiting time distribution is*

$$G(x;\theta) = \frac{\mu}{\theta + \mu(1-x)}$$

We obtain the LST of the unconditional probability distribution by evaluating the generating function at ρ and multiplying it by $(1-\rho)$:

$$W^*(\theta) = (1-\rho)G(\rho;\theta) = \frac{\mu - \lambda}{\theta + \mu - \lambda}$$

Proof
Let I be the number of tasks the tagged task faces at its arrival. Conditioning on $I = i$, the sojourn time of the tagged task is the sum of i exponential variables with parameter μ and the remaining service time of the task that is currently being served. The remaining sojourn time is also exponential with the same parameter due to the memoryless property of the exponential distribution. Since the variables are independent, the LST of the conditional distribution is $W_i^*(\theta) = \left(\frac{\mu}{\theta+\mu}\right)^{i+1}$. Thus,

$$G(x;\theta) = \sum_{i=0}^{\infty} W_i^*(\theta)x^i = \sum_{i=0}^{\infty} \left(\frac{\mu}{\theta+\mu}\right)^{i+1} x^i = \frac{\mu}{\theta + \mu - \mu x}$$

Furthermore, $P(I = i) = (1-\rho)\rho^i$ by the random observer property. Consequently, evaluating the generating function at ρ and multiplying it by $(1-\rho)$ gives the desired probability distribution by the law of total probability. □

Proposition 2.3.3 (Waiting time distribution with PS) *In a PS M/M/1 queue with fixed arrival rate λ and service rate μ, the Laplace transform of the waiting time conditioned on the customer's service time is*

$$W^*(\theta|x) = \frac{(1-\rho)(1-\rho r^2)e^{-[\lambda(1-r)+\theta]x}}{(1-\rho r)^2 - \rho(1-r^2)e^{-(\mu/r - \lambda r)x}}$$

where r is the smaller root of the equation $\lambda r^2 - (\lambda + \mu + \theta)r + \mu = 0$ and $\rho = \frac{\lambda}{\mu}$.

Outline of the proof
Using the following notations:

- customer C arrives at 0
- faces a queue length N
- has service time X

- has waiting time T

We consider the time interval $(0, t+h)$ and an initial infinitesimal interval $(0, h)$. During that short interval two things can happen:

1. a new task arrives at the queue with probability λh
2. a task other than C is completed with probability $\frac{n\mu}{n+1}h$

We have

$$P(T \leq t+h|X=x, N=n) = \left[1 - \lambda h - \frac{n\mu}{n+1}\right] P(T \leq t|X = x - \frac{h}{n+1}, N=n) \; +$$

$$+ \; \lambda h \, P(T \leq t|X=x, N=n+1) + \frac{n\mu}{n+1}h \, P(T \leq t|X=x, N=n-1) + \; o(h).$$

We are taking advantage of the random observer property of the Poisson process which allows us to use the same time variable T on the right hand side.
Letting $F_n(t, x) = P(T \leq t|X = x, N = n)$, after taking the limit $h \to 0$ and some rearrangement we get:

$$(n+1)\frac{\partial F_n}{\partial t} = -\frac{\partial F_n}{\partial x} - ([n+1]\lambda + n\mu)F_n + (n+1)\lambda F_{n+1} + n\mu F_{n-1}.$$

Now let L_n be the Laplace transform of $\frac{\partial F_n}{\partial t}$. After differentiation by t, using the property of the Laplace transform for derivatives we obtain:

$$(n+1)\theta L_n = -\frac{\partial L_n}{\partial x} - ([n+1]\lambda + n\mu)L_n + (n+1)\lambda L_{n+1} + n\mu L_{n-1}.$$

We solve this recurrence relation by introducing the generating function

$$G(z; \theta, x) = \sum_{n=0}^{\infty} L_n(\theta; x)z^n.$$

Note that $W^*(\theta|x) = (1-\rho)G(\rho; \theta, x) = \sum_{n=0}^{\infty} L_n(\theta; x)(1-\rho)\rho^n$ by the law of total probability. Multiplying by z^n and summing from $n = 0$ to ∞ results in the following first-order partial differential equation:

$$[\mu z^2 - (\lambda + \mu + \theta)z + \lambda]\frac{\partial G}{\partial z} - \frac{\partial G}{\partial x} = (\lambda + \theta - \mu z)G.$$

We solve this by solving the auxiliary equations. The Laplace transform of the unconditioned waiting time can be obtained as an integral by using the fact that the service time of a customer is exponential with parameter μ.

$$W^*(\theta) = \int_0^{\infty} W^*(\theta|x)\mu e^{-\mu x}dx.$$

Chapter 3

Batch queues at equilibrium

The aim in this section is to state the product-form result for the equilibrium queue length probabilities in the case of a tandem pair of batch-queues. This result uses the reversed compound agent theorem, so we start by introducing the concept of reversing a stochastic process and look at a simple example of a single queue where the reversed process can be used to obtain the equilibrium queue length probabilities. Then we provide a proper definition of the batch-queue model in question, noting that so-called special arrivals and departures have to be added to the system; otherwise a product-form is known not to exist. We investigate how the effect of these somewhat artificial external arrivals and departures is minimized mathematically and what this means for the model. We then observe what reversing the batch queue means in this particular model and finally state the product-form theorem.

3.1 Reversed process

The analysis of a process can often be greatly simplified by looking at its reversed counterpart. We use this for both the product form result and later on to obtain the joint probability distribution of the tandem batch-network. The reversed process can be thought of as if we stopped the time in the original process and then rewound the tape. In the case if the reversed process is stochastically identical to the original one, we say the process is reversible. However, looking at the reversed process can still be useful even if the original process is not reversible. Formally, reversibility is defined as follows.

Definition 3.1.1 (Reversibility) $\{X_t\}$ *is reversible if* $\forall t_1, ..., t_n, \tau$:

$$F_{X_{t_1},...,X_{t_n}} = F_{X_{\tau-t_1},...,X_{\tau-t_n}}.$$

It is straightforward to see that a reversed process must be stationary, but not all stationary processes are reversible.

Proposition 3.1.1 *The reversed process of a stationary Markov process defined by state space S, instantaneous transition matrix Q, and equilibrium probabilities π is also a stationary Markov process with the same π and its instantaneous transition matrix Q' is defined by:* $q'_{ij} = \frac{\pi_j}{\pi_i} q_{ji}$

In practice, the above proposition can be used to find the equilibrium probabilities. First, one needs to guess q'_{ij} which can often be done by just imagining what the reversed process would look like, and then guessing a collection of positive real numbers π with a finite sum

P. If $q'_i = q_i$ and $q'_{ij} = \frac{\pi_j}{\pi_i} q_{ji}$ then it follows directly that π satisfies Kolmogorov's balance equations so (π/P) must be the equilibrium probability (the only one by uniqueness).
This leads to a necessary and sufficient condition for the reversibility of a stationary Markov process.

Proposition 3.1.2 *A process is reversible if and only if it satisfies the* ***detailed balance equations****:*
$\forall i, j \in S$

$$\pi_i q_{ij} = \pi_j q_{ji}$$

A related and more general result is Kolmogorov's theorem:

Proposition 3.1.3 (Kolmogorov's criteria) *A stationary Markov process with generator matrix Q and state spate S has a reversed process with Q' if and only if*

1. $g'_i = q_i \ \forall i \in S$
2. $q_{i_1,i_2} ... q_{i_{n-1},i_n} q_{i_n,i_1} = q_{i_1,i_n} ... q_{i_2,i_1} \ \forall i_1, ... i_n \in S$

In the case of a network of queues, reversibility is somewhat more complicated. We say that two processes P and Q are cooperating if they synchronize over a set of actions. Here, we always assume that there is an active and a passive participant, which means the active process determines the rate of the synchronized action.
The reversed compound agent theorem (**RCAT**) defines the necessary conditions under which a process can be reversed, and also provides an explicit formula to calculate the reversed rates. We omit stating and proving the theorem since it requires the introduction of an abbreviated PEPA syntax which is not particularly useful in the rest of the thesis. A detailed explanation can be found in [6].

3.2 Equilibrium probabilities

As stated at the beginning of this chapter, the aim is to calculate the response time probabilities in a tandem network of batch-queues. A batch-queue is an ordinary queue except that tasks arrive and leave the queue in batches of possibly different sizes. In a tandem network of batch-queues, there are two batch-queues after each other with given arrival and departure rates which can depend on the batch size. Both the arrival and departure processes are Markovian. When a batch leaves the first queue it proceeds to the second one, after a potential re-batching.
We wish to use the generating function method to obtain the LSTs of the sojourn time distributions in the next chapter, as we did in the simple case of an M/M/1 queue in Proposition 2.3.2. Therefore, the joint equilibrium queue length probabilities have to be in product form. In order to obtain a product form result in the tandem batch network mentioned earlier, we appeal to RCAT. One of the conditions of RCAT is that all occurrences of a reversed active action type must have the same rate. In our case, active actions are k-departures from the first queue proceeding to the second queue. Assuming that the rate of departing batches of size k does not depend on the local state of the node, the rate of a k-departure is: $\mu_k : i + k \to i \ \forall i \geq 0$. Now, using Proposition 3.1.1 to calculate the reversed rates of k-batch departures we have $\mu'_k = \frac{\pi_{i+k}}{\pi_i} \mu_k$. In the case of geometric equilibrium probabilities, this simplifies to $\mu'_k = \rho^k \mu_k$, which does not depend on i as required.

Therefore, we are seeking the conditions under which the equilibrium probabilities are geometric and the joint distributions have a product form. The other advantage of geometric equilibrium probabilities is that – in case of a single queue – the generating function can be directly used to obtain the unconditional probability distribution.
It is well known that no such product-form exists in a tandem batch-network with only normal batch arrivals and departures. Hence, we introduce special arrivals and departures at both queues. This approach in itself is not new and similar results in special cases were obtained in [10, 11].

The batch-queue model

Special arrivals can occur whenever the queue length is 0, and special departures empty the queue (in the case of the first queue this also means that tasks in that particular batch do not proceed to the second queue).

We are using the following notations from now on:

1. $A_i(z) = \sum_{s=1}^{n_{ai}} a_{i;s} z^s$ the generating function of the rates of normal arrivals at node i, where n_{ai} is the maximum batch size for the normal arriving batches
2. $D_i(z) = \sum_{s=1}^{n_{di}} d_{i;s} z^s$ the generating function of the rates of normal departures at node i, where n_{di} is the maximum batch size for the normal departing batches
3. $A_{i0}(z) = \sum_{s=1}^{n_{ai0}} a_{i;0s} z^s$ the generating function of the rates of special arrivals at node i, where n_{ai0} is the maximum batch size for the special arriving batches
4. $D_{i0}(z) = \sum_{s=1}^{n_{di0}} d_{i;s0} z^s$ the generating function of the special departures at node i, where n_{di0} is the maximum batch size for the special departing batches

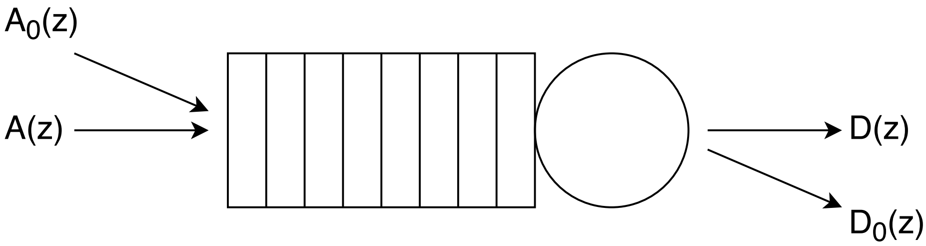

Figure 3.1: The batch queue model.

In the case of a single queue we usually omit the index i. We only consider the sojourn times in the case of finite maximum batch sizes, however in this section we only assume that $A(1), D(1), A_0(1) < \infty$. Figure 3.1 represents the model graphically.

Proposition 3.2.1 *The batch queue defined above has geometrically distributed equilibrium queue length probabilities, that is $\pi_n = (1-\rho)\rho^n$, if and only if*

$$(1-\rho z)[A_0(z) - D_0(\rho z)] = [A(1) - D(\rho)]\rho z - A(z) + D(\rho z) \tag{3.1}$$

for $|z| < min(\rho^{-1}, R)$ where R is the minimum of the radii of convergence of the four rate generating functions.

A queue that satisfies the above equation is called a **geometric batch queue**. This result is proved using the balance equations which appear in the background chapter. While A_0 and D_0 are not uniquely defined by (3.1), in practice we often aim to minimize the effect caused by these somewhat artificial arrivals and departures. The following corollary shows what minimizing the effect caused by the special arrivals and departures means mathematically.

Corollary 3.2.1 *Suppose* $A(z), D(z)$ *are given and* $A_0(z)$*,* $D_0(z)$ *are chosen to satisfy* (3.1).

1. $A_0(z) - D_0(\rho z)$ *has radius of convergence* $< \rho^{-1}$ *unless*

$$A(1) + D(1) = A(\rho^{-1}) + D(\rho) \tag{3.2}$$

which means it can only have an infinite radius of convergence if the above equation holds.

2. *If* $\exists r \in (0,1) : r^{-1}$ *is strictly less than the radius of convergence of* $A(z)$ *and* $A(1)+D(1)-D(r) < A(r^{-1})$ *then:*

$$A(x^{-1}) + D(x) = A(1) + D(1) \text{ has a unique root } x_0 \in (0,1) \Leftrightarrow A'(1) < D'(1)$$

In this case, the batch queue has a geometric equilibrium probability distribution with parameter $\rho = x_0$.

3. *Conversely, if* π *has a geometric distribution then* $A(\rho^{-1}) + D(\rho) = A(1) + D(1)$ *when* $A(\rho^{-1}) < \infty$.

An important special case of the above model is when we allow partial departing batches. That is, every time a departing batch size is larger than the current queue length, all the tasks are discarded. Formally this means

$$d_{l0} = \sum_{s=l+1}^{n_d} d_s$$

In this case, $D_0(z)$ and $A_0(z)$ are both determined entirely by $A(z)$ and $B(z)$:

$$D_0(z) = \sum_{l=1}^{n_d-1} \sum_{s=l+1}^{n_d} d_s z^l = \sum_{s=2}^{n_d} \sum_{l=1}^{s-1} d_s z^l \tag{3.3}$$

$$= \sum_{s=2}^{n_d} d_s \frac{z - z^s}{1-z} = \frac{zD(1) - D(z)}{1-z} \tag{3.4}$$

Now using this together with (3.1) we get:

$$A_0(z) = \frac{[A(1) + D(1) - D(\rho)]\rho z - A(z)}{1 - \rho z}$$

This is called a **discard batch-queue**.
When in addition (3.2) is satisfied too, the formula for $A_0(z)$ simplifies to:

$$A_0(z) = \frac{\rho z A(\rho^{-1}) - A(z)}{1 - \rho z}$$

in which case it is called a **minimal discard batch-queue**.
There are some additional properties of a minimal discard batch-queue as follows.

Proposition 3.2.2 *In a minimal discard batch-queue defined by $A(z), D(z)$:*

1. $A_0(z)$ *has finitely many terms* $\Leftrightarrow A(z)$ *has finitely many terms*
2. $A(z) = \sum_{i=1}^{n} a_i z^i \Rightarrow A_0(z) = \sum_{j=1}^{n-1} (\rho z)^j \sum_{i=j+1}^{n} a_i \rho^{-i}$
3. $A(z) = A(1)\frac{(1-\alpha)z}{1-\alpha z}$ *(it is geometric with parameter α)* $\Rightarrow A_0(z) = A(z)\frac{\alpha}{\rho-\alpha}$

Note that part two in the above proposition means that in the case of an M/M/1 queue with deterministic arrival and departure rates, the degree of the generating function of the special arrivals is 0. Furthermore, due to (3.4) the degree of the generating function of the special departure is also 0. Thus, in this case there are no need for special arrivals and departures in order to have a geometric probability distribution at equilibrium as expected.

The reversed batch-queue

Using Proposition 3.1.1, the rates of the reversed process can be easily determined for both the geometric batch-queue and the more special minimal discard batch-queue.

Proposition 3.2.3 *The reversed process of a geometric batch-queue with parameter ρ defined by A, A_0, D, D_0 is also a geometric batch queue with the same parameter and the following rate generating functions:*

$$A'(z) = D(\rho z) \qquad A'_0(z) = D_0(\rho z)$$
$$D'(z) = A(\rho^{-1} z) \qquad D'_0(z) = A_0(\rho^{-1} z)$$

The above proposition is easily proved by obtaining the individual reversed rates from Proposition 3.1.1. To check if the reversed process is also a geometric batch-queue, we apply Proposition 3.2.1 and check that equation (3.1) holds.

Proposition 3.2.4 *The reversed process of a minimal discard batch-queue with parameter ρ defined by A, D is also a minimal discard batch queue with the same parameter and rate generating functions:*

$$A'(z) = D(\rho z) \qquad D'(z) = A(\rho^{-1} z)$$

Proof
It is already known that the reversed process is a geometric batch-queue using the previous proposition. It is only left to prove that it is a minimal discard queue. We do that by proving that (3.4) holds.

$$D'_0(z) = A_0(\rho^{-1} z) = \frac{zA(\rho^{-1}) - A(\rho^{-1} z)}{1-z} = \frac{zD'(1) - D'(z)}{1-z}$$

We also have to check that (3.2) is satisfied, which follows easily using the equations we obtained above and the fact that the original process is a minimal discard batch queue.

$$A'(1) + D'(1) = D(\rho) + A(\rho^{-1}) = D(1) + A(1) = A'(\rho^{-1}) + D'(\rho)$$

□

We only consider minimal batch queues from here onwards; therefore the rate generating functions are determined uniquely by $A(z)$ and $D(z)$.

Product form result

Now we are ready to state the product-form theorem for a network of discard batch-queues. It is the generalization of equation (3.2) and therefore Proposition 3.2.1.

Theorem 3.2.1 *A network of M minimal discard batch-queues at equilibrium has $P(\boldsymbol{N} = \boldsymbol{n}) = \prod_{j=1}^{M} (1 - \rho_j)\rho_j^{n_j}$ where $\boldsymbol{N} = (N_1, ..., N_M), \boldsymbol{n} = (n_1, ..., n_M)$ and $(\rho_1, ..., \rho_M)$ are the solutions of:*

$$A_j(1) + D_j(1) + \sum_{k=1,k\neq j}^{M} B_{kj}(\rho_k, 1) = A_j(\rho_j^{-1}) + D_j(\rho_j) + \sum_{k=1,k\neq j}^{M} B_{kj}(\rho_k, \rho_j^{-1}) \tag{3.5}$$

The extra terms correspond to the arrivals at node j coming from another node k and are thus determined by RCAT.

$$B_{kj}(\rho_k, z) = \sum_{l=1}^{\infty} \sum_{i=1}^{\infty} p_{kijl}(\rho_k^i d_{k;i}) z^l$$

where p_{kijl} is the so-called routing probability, the probability that a batch of size i leaving node k will arrive at j as a batch of size l.

Chapter 4

Sojourn times

In the minimal discard batch-queue model, a task can be discarded both halfway – finishing its service at the first queue – and at the end – finishing its service at the second queue. We are interested in the case when it finishes its service normally, that is, goes along the path without being discarded.

It is particularly challenging to calculate sojourn times for a tandem networks of batch-queues because the path is not overtake free. This means that even after the tagged task arrives at the second queue its sojourn time is not independent of the events happening at the first queue. More specifically, certain arrivals from the first queue can influence whether the tagged task departs in a partial or complete batch.

In this chapter, we first consider sojourn times in the case of a single batch-queue and its reversed counterpart. Then we calculate the sojourn times for the tandem network without distinguishing between special and normal batches. Relaxing this condition results in overtake free paths, hence the LST of the joint sojourn time can be achieved more easily. Next, we move to the more interesting general case, and obtain the LST of the marginal distribution of the sojourn time at the second queue. This requires the solution of a recurrence for a vector of generating functions rather than a single generating function. Finally, we put everything together and calculate the LST of the joint probability distribution of the sojourn times at the first queue and the second queue.

4.1 Sojourn time in a single batch-queue

Before we start calculating the sojourn time itself, we look at two proposition corresponding to properties of the exponential random variable. Both are used several times later on.

Proposition 4.1.1 *$X_1, ..., X_n$ are independent exponential random variables with parameters $\lambda_1, ...\lambda_n$, respectively.*

$$P\left(\min(X_1, ..., X_n) > t\right) = \exp\left(-\sum_{i=1}^{n} \lambda_i t\right)$$

In other words, the minimum of n independent exponential variables is also exponentially distributed with parameter $\sum_{i=1}^{n} \lambda_i$.

Proof

$$P\left(\min(X_1, ..., X_n) > t\right) = P(X_1 > t, ..., X_n > t) = \prod_{i=1}^{n} e^{-\lambda_i t} = e^{-t\sum\limits_{i=1}^{n} \lambda_i}$$

□

Above we only calculate the distribution of the minimum, but we often need to know the probability that a specific variable has the smallest value.

Proposition 4.1.2 *$X_1, ..., X_n, Y$ are independent exponential random variables with parameters $\lambda_1, ..., \lambda_n, \mu$, respectively.*

$$P(X_1, ..., X_n > Y) = \frac{\mu}{\sum\limits_{i=1}^{n} \lambda_i + \mu}$$

Proof

$$\begin{aligned} P(X_1, ...X_n > Y) &= \int\limits_0^\infty P(X_1, ..., X_n > t | Y = t) f_Y(t) dt \\ &= \int\limits_0^\infty e^{-(\sum\limits_{i=1}^{n} \lambda_i)t} \mu e^{-\mu t} dt = \frac{\mu}{\sum\limits_{i=1}^{n} \lambda_i + \mu} \end{aligned}$$

□

As has been stated earlier we usually use the Laplace-Stieltjes transform to determine the probability distribution. The exponential distribution appears many times, so we calculate its LST in advance.

Proposition 4.1.3 *X is an exponential random variable with parameter λ.*

$$L^*(\theta) = \frac{\lambda}{\theta + \lambda}$$

Proof

$$\begin{aligned} L^*(\theta) &= E(e^{-X\theta}) \quad \text{by the definition of the LST} \\ &= \int\limits_0^\infty e^{-\theta t} \lambda e^{-\lambda t} dt = \frac{\lambda}{\theta + \lambda} \end{aligned}$$

as stated. □

Using the propositions above, we can immediately calculate the LST of the time to the next departure of either kind, since that is just a minimum of some exponential variables:

1. normal departure of size $k \sim \text{Exp}(d_k)$
2. special departure $\sim \text{Exp}(d_{l0})$, when the current queue length is l

The time to the next departure is the minimum of these exponential variables, so by Proposition 4.1.1 it is itself exponential with parameter $\sum_{i=1}^{l} d_i + d_{l0} = D(1)$.
Hence, by Proposition 4.1.3 the LST of the random variable is:

$$S^*(\theta) = \frac{D(1)}{\theta + D(1)} \tag{4.1}$$

Normal-to-any paths

We denote the remaining sojourn time of a task at position $m+1$ with R_m and the corresponding LST by R^*_m. Let $G_R(z,\theta)$ be the generating function of R^*_m.
In the easiest case, there is only one queue and we do not care about the type of the departing batch.

Proposition 4.1.4 *In an equilibrium minimal discard batch-queue defined by A, D*

$$G_R(z,\theta) = \frac{D(1) - D(z)}{(1-z)(\theta + D(1) - D(z))} \tag{4.2}$$

Proof
If the current queue length is l, by the law of total probability the remaining sojourn time is S if the next departing batch contains the tagged task or $(S + R_{n-j})$ if it is a j-departure without the tagged task:

$$R_n = \frac{\sum_{i=n+1}^{l} d_i + d_{l0}}{D(1)} S + \sum_{j=1}^{n} d_j (S + R_{n-j})$$

Now, taking the LST of both sides we have

$$R^*_n = \frac{S^*}{D(1)} \left[d_{n+1} + d_{n+1,0} + \sum_{j=1}^{n} d_j R^*_{n-j} \right]$$

Multiplying by z^n and summing from $n = 0$ to ∞:

$$\begin{aligned} G_R(z) &= \frac{S^*}{D(1)} \left[\sum_{n=0}^{\infty} d_{n+1} z^n + \sum_{n=0}^{\infty} d_{n+1,0} z^n + \sum_{n=0}^{\infty} \sum_{j=1}^{n} d_j R^*_{n-j} z^n \right] \\ &= \frac{S^*}{D(1)} \left[z^{-1}(D_0(z) + D(z)) + \sum_{j=1}^{\infty} \sum_{n=0}^{\infty} d_j R^*_n z^{n+j} \right] \end{aligned}$$

$$= \frac{S^*}{D(1)} \left[\frac{z^{-1}(zD(1) - D(z) + (1-z)D(z))}{1-z} + D(z)G_R(z) \right]$$

Thus

$$G_R(z)(D(1) - D(z)S^*) = \frac{S^*(D(1) - D(z))}{1-z}$$

Plugging in (4.1) for S^* we get the desired result. □

Since the reversed process is also a minimal discard batch queue as stated in Proposition 3.2.4, a similar result for it can be easily obtained:

$$\widetilde{G_R}(z, \theta) = \frac{A(\rho^{-1}) - A(\rho^{-1}z)}{(1-z)[\theta + A(\rho^{-1}) - A(\rho^{-1}z)]}$$

Now, in order to calculate the actual sojourn times, we have to consider three cases separately:

1. **First task in the batch**

   By the random observer property (2.2.5), the probability that a tagged task faces a queue length n at its arrival is geometric with parameter ρ. Thus, using the generating function we can easily obtain the sojourn time as

   $$T_F^*(\theta) = (1-\rho)G_R(\rho, \theta) = \frac{D(1) - D(\rho)}{\theta + D(1) - D(\rho)}$$

   which is an exponential random variable with parameter $D(1) - D(\rho)$.

   In case of the reversed process we similarly get an exponential variable with parameter $A(\rho^{-1}) - A(1)$. Furthermore, $A(\rho^{-1}) - A(1) = D(1) - D(\rho)$ by (3.2); hence we get the same exponential variable for the reversed process when we consider the first task in the batch.

2. **Last task in the batch**

   The last task in a batch joins the queue at position $M+K$ where M is the queue length at the arrival instant and K is the batch size. Hence, we have $P(M=n) = (1-\rho)\rho^n$ and $P(K=k) = \frac{a_k}{A(1)}$ and

   $$T_L^*(\theta) = \frac{(1-\rho)}{A(1)} \sum_{m=0}^{\infty} \sum_{k=1}^{\infty} a_k \rho^m R_{m+k-1}^*(\theta) = \frac{(1-\rho)}{A(1)} \sum_{m=0}^{\infty} \sum_{k=1}^{m+1} a_k \rho^{m-k+1} R_m^*(\theta)$$

   which in case $A(z)$ is geometric simplifies to

   $$T_L^*(\theta) = (1-\rho)(1-\alpha)\frac{\rho G_R(\rho) - \alpha G_R(\alpha)}{\rho - \alpha}$$

3. **Random task in the batch**

   The tagged task joins the queue at $M+H$ where M is the current queue length and H is the tagged task's position in the arriving batch. Now, $b_k = P(H=k) = \sum\limits_{j=k}^{\infty} \frac{a_j}{A'(1)}$ by a well-known result for the backwards recurrence time.

   $$T_R^*(\theta) = \sum_{m=0}^{\infty} \sum_{k=1}^{m+1} b_k \rho^{m-k+1} R_m^*(\theta)$$

   If $A(z)$ is geometric, we get the same result as for the last task in the batch as expected by the memoryless property of the geometric distribution.

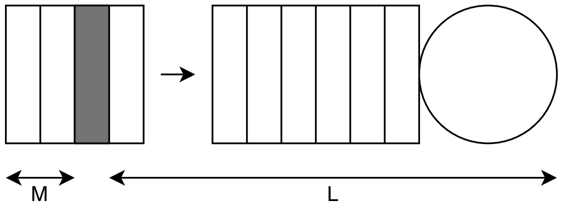

Figure 4.1: Single batch-queue at the arrival instant of the tagged task.

Normal-to-normal paths

A significantly harder case and also more interesting, when we only consider those tasks that depart in a normal batch. Up to this point, this section has mainly used results from [5]; however in the following proposition (4.1.5) when obtaining a recurrence for the generating function, we deviate from the article and do not take into account the tagged task's position in the departing batch since it is not needed for the joint case and omitting it makes the statement and the proof neater. Furthermore, we also prove that the matrix on the left hand side of equation (4.4) can indeed be inverted, and therefore that a solution for the vector of generating functions can be obtained, which was omitted from the article [5].

As shown in figure 4.1, we use L to denote the number of tasks in front of the tagged one and M for the number of tasks behind it.
We seek the quantities $\gamma_{lm}(t) = P(R < t, I = 1|L = l, M = m)$, where I is the indicator function of the event that *the departing batch is normal* and R is the response time random variable. We keep using the *-notation to mean the LST of a function, in this case $\gamma^*_{lm}(\theta)$.
Let $H_m(x;\theta) = \sum_{l=0}^{\infty} \gamma^*_{lm}(\theta)x^l$ be the generating function of the conditional probabilities. First, we obtain a recurrence relation for the generating functions.

Proposition 4.1.5 *H_m is as above but we omit θ from the argument for brevity.*

$$[A(1) + D(1) - D(x) + \theta]H_m(x) = \sum_{s=1}^{n_a} a_s H_{m+s}(x) + \sum_{i=0}^{n_d-1} \sum_{s=i+1}^{\min(i+m+1,n_d)} d_s x^i \tag{4.3}$$

Proof
If we consider an infinitesimal period of time, the following events can occur:

1. normal arrival
2. normal departure without the tagged task
3. normal departure containing the tagged task

Note, there cannot be any special arrivals, because the queue length is at least 1.
Since both the arrival and departure processes are Poisson, we know that the probability that an event with rate λ happens in an interval of size h is $\lambda h + o(h)$ by Definition 2.2.12.
Therefore, by using the memoryless property of the exponential distribution we have

$$\gamma_{lm}(t+h) = h\sum_{s=1}^{n_a} a_s\gamma_{l,m+s}(t) + h\sum_{s=1}^{\min(l,n_d)} d_s\gamma_{l-s,m}(t) + h\sum_{s=l+1}^{\min(l+m+1,n_d)} d_s + h[1-A(1)-D(1)]\gamma_{lm} + o(h)$$

After rearranging, dividing by h and taking the limit $h \to 0$, we take the LST of both sides,

$$[\theta + A(1) + D(1)]\gamma^*_{lm} = \sum_{s=1}^{n_a} a_s\gamma^*_{l,m+s} + \sum_{s=1}^{\min(l,n_d)} d_s\gamma^*_{l-s,m} + \sum_{s=l+1}^{\min(l+m+1,n_d)} d_s$$

Multiplying by x^l and then summing from $l = 0$ to ∞ we get

$$[\theta + A(1) + D(1)]H_m(x) = \sum_{s=1}^{n_a} a_s H_{m+s}(x) + \sum_{l=0}^{\infty}\sum_{s=1}^{\min(l,n_d)} d_s\gamma^*_{l-s,m}x^l + \sum_{l=0}^{\infty}\sum_{s=l+1}^{\min(l+m+1,n_d)} d_s x^l$$

where the second term has a closed form:

$$\sum_{l=0}^{\infty}\sum_{s=1}^{\min(l,n_d)} d_s\gamma^*_{l-s,m}x^l = \sum_{s=1}^{n_d}\sum_{l=s}^{\infty} d_s\gamma^*_{l-s,m}x^l = \sum_{s=1}^{n_d}\sum_{l=0}^{\infty} d_s\gamma^*_{l,m}x^{l+s} = D(x)H_m(x)$$

Hence, we get the stated equation. □

Moving forward, notice that as soon as there are at least $n_d - 1$ tasks behind the tagged one, a new arrival can no longer influence the tagged task's response time, which means $H_m = H_{n_d-1}$ for all $m \geq n_d - 1$.

The above proposition can be written in matrix form as follows:

$$\mathbb{L}(x)\boldsymbol{H}(x) = \mathbb{Y}\mathbb{D}\boldsymbol{v}(x) \tag{4.4}$$

where $\boldsymbol{v}(x) = (1, x, ..., x^{n_d-1})$, $\mathbb{L}(x) = (A(1) + D(1) - D(x) + \theta)\mathbb{I} - \mathbb{M} - \mathbb{K}$, and $\mathbb{D}, \mathbb{Y}, \mathbb{M}, \mathbb{K}$ are $n_d \times n_d$ matrices defined as follows:

$$\mathbb{D} = \begin{bmatrix} d_1 & \dots & d_{n_d-1} & d_{n_d} \\ d_2 & \dots & d_{n_d} & 0 \\ \vdots & \iddots & \iddots & \vdots \\ d_{n_d} & 0 & \dots & 0 \end{bmatrix} \qquad \mathbb{Y} = \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 1 & 1 & 0 & \dots & 0 \\ & \vdots & & \vdots & \\ 1 & & \dots & & 0 \\ 1 & 1 & \dots & & 1 \end{bmatrix}$$

$$\mathbb{M} : \begin{cases} m_{ij} = a_{j-i} & 0 \leq i \leq n_d - 1, i+1 \leq j \leq \min(n_a + i, n_d - 1) \\ 0 & \text{otherwise} \end{cases}$$ */upper triangular/*

$$\mathbb{K} : \begin{cases} k_{ij} = \sum_{s=n_d}^{n_a+i} a_{s-i} & 0 \leq i \leq n_d - 1, j = n_d - 1 \\ 0 & \text{otherwise} \end{cases}$$ */only the last column is non-zero/*

Assuming that $\mathbb{L}(x)$ can be inverted and therefore we can solve the equation for $\boldsymbol{H}(x)$, the vector of generating functions can be used to obtain the LST of the response time in the following way.

We start by looking at a special case when the maximum batch size is two for both the arriving and departing batches, then move on to the general case.

Proposition 4.1.6 *When $n_a = n_d = 2$ the LST of the sojourn time of a task arriving and departing in a normal batch considering a random task in the batch is:*

$$T^*(\theta)/T^*(0)$$

where

$$T^*(\theta) = \frac{(1-\rho)[(a_1 + \frac{a_2}{\rho})H_0(\rho;\theta) + a_2 H_1(\rho;\theta) - \frac{a_2}{\rho}H_0(0;\theta)]}{a_1 + 2a_2}$$

and

$$H_0(x;\theta) = \frac{(1-x)d_1^2 + [\theta + a_1 + a_2 + (1-x)(1+2x)d_2]d_1 + [x\theta + (1+x)(a_1+a_2) + x(1-x^2)d_2]d_2}{[\theta + a_1 + a_2 + (1-x)d_1 + (1-x^2)d_2][\theta + (1-x)d_1 + (1-x^2)d_2]}$$

$$H_1(x;\theta) = \frac{d_1 + (1+x)d_2}{\theta + (1-x)d_1 + (1-x^2)d_2}$$

In addition, the probability that a task arriving in a normal batch departs in a normal batch is $T^*(0)$.

Proof
Using the matrix form of Proposition 4.1.5 and omitting θ for brevity we have,

$$[a_1 + a_2 + d_1(1-x) + d_2(1-x^2) + \theta]H_0(x) = a_1 H_1(x) + a_2 H_1(x) + d_1 + d_2 x$$
$$[a_1 + a_2 + d_1(1-x) + d_2(1-x^2) + \theta]H_1(x) = a_1 H_1(x) + a_2 H_1(x) + d_1 + d_2(1+x)$$

Solving the above equation, we get the stated solution for $H_0(x)$ and $H_1(x)$.

Let π_{lm} be the probability that there are l tasks before and m tasks behind the tagged one at its arrival instant.
$\pi_{lm} = (1-\rho)\sum_{s=0}^{l} f_{sm}\rho^{l-s}$ by the random observer property, where f_{sm} is the probability that there are s tasks in front of and m tasks behind the tagged one in its arrival batch.

1. First task in the batch: $f_{sm} = \frac{a_{m+1}}{A(1)}\mathbb{I}_{\{s=0\}}$
2. Last task in the batch: $f_{sm} = \frac{a_{s+1}}{A(1)}\mathbb{I}_{\{m=0\}}$
3. Random task in the batch: $f_{sm} = \frac{a_{s+m+1}}{A'(1)}$

We carry on by looking at the most difficult case, when the tagged task is in a random position (the other two cases can be obtained similarly).

$$T^*(\theta) = \sum_{m=0}^{n_a-1}\sum_{l=0}^{\infty}\gamma_{lm}^*(\theta)\pi_{lm} = \sum_{m=0}^{1}\sum_{l=0}^{\infty}\gamma_{lm}^*(\theta)(1-\rho)\sum_{s=0}^{l} f_{sm}\rho^{l-s} =$$

$$(1-\rho)\sum_{l=0}^{\infty}\left[\gamma_{l0}^{*}(\theta)(f_{00}\rho^{l}+f_{10}\rho^{l-1}\mathbb{I}_{\{l\geq 1\}})+\gamma_{l1}^{*}(\theta)f_{01}\rho^{l}\right]=$$
$$(1-\rho)\frac{H_0(\rho;\theta)(a_1+\frac{a_2}{\rho})+H_1(\rho;\theta)a_2-H_0(0;\theta)\frac{a_2}{\rho}}{a_1+2a_2}$$

□

For the general case, following the same method and noting that $f_{sm}=0$ whenever $s+m\geq n_a$ we get,

$$T^{*}(\theta)=(1-\rho)\sum_{m=0}^{n_a-1}\sum_{l=0}^{n_a-1-m}\rho^{-l}f_{lm}\left[H_{\min(m,n_d-1)}(\rho;\theta)-\sum_{s=0}^{l-1}\frac{\rho^s}{s!}\left.\frac{\partial^s H_{\min(m,n_d-1)}(x;\theta)}{\partial x^s}\right|_{x=0}\right] \tag{4.5}$$

In order for $\mathbb{L}$ to be invertible, it cannot be singular inside the unit disk. Since it is an upper triangular matrix, this means the diagonal elements have to be nonzero.

1. All but the last diagonal element is equal to: $A(1)+D(1)-D(x)+\theta$

   $$|A(1)+D(1)-D(x)+\theta|\geq|A(1)+D(1)+\theta|-|D(x)|$$

   by the reverse triangle inequality. Assuming that $\mathrm{Re}(\theta)\geq 0$ and noting that $|D(x)|\leq D(1)$ inside the unit disk,

   $$|A(1)+D(1)+\theta|-|D(x)|\geq A(1)+D(1)-D(1)=A(1)>0$$

2. Last diagonal element: $D(1)-D(x)+\theta$

   Similarly,

   $$|D(1)-D(x)+\theta|\geq|D(1)+\theta|-|D(x)|\geq D(1)-D(1)=0$$

   However if both inequalities are in fact equalities, we have to assume that $\mathrm{Re}(\theta)\geq\varepsilon$ in order for it to be strictly greater than 0.

We showed for both cases that the absolute value of the diagonal elements is strictly larger than 0 inside the unit disk; therefore there cannot be any roots there.
These restrictions for θ are ubiquitous and do not prevent the inversion of the LST since it can be done along a vertical line.

4.2 Sojourn time in a tandem batch-network

The general idea for calculating the sojourn time for a tandem batch-network is the following. We move the origin of the time axis to the instant when the tagged task leaves node 1 and enters node 2. Then we look at the joint probability of the forward sojourn time at the second queue and the reversed sojourn time at the first queue given the middle state I,L and in-transit batch position J,K.

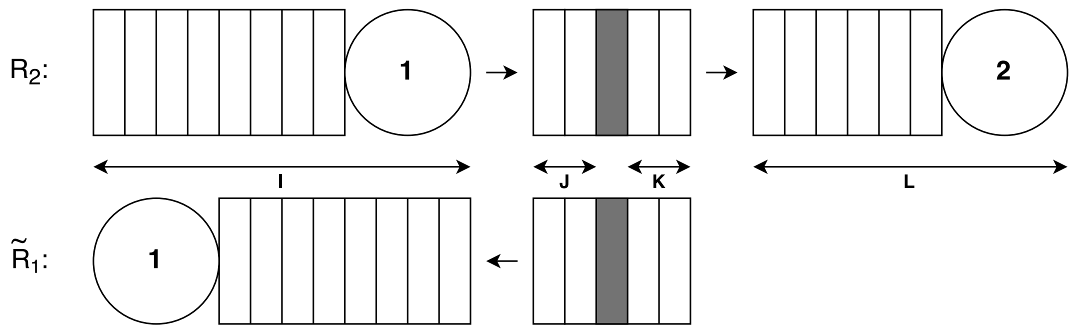

Figure 4.2: State at t=0, when calculating the joint probability distribution of the forward sojourn time at node 2 and the reversed sojourn time at node 1.

As is shown in figure 4.2, I is the number of tasks in the first queue, L is the number of tasks in the second queue. Note that we leave out the tagged tasks batch from both places. We denote the number of tasks behind and in front of the tagged task in its batch by J and K, respectively.
We start by looking at the middle state probabilities and the distribution of the position in the batch.

Proposition 4.2.1 *The joint probability of I and L above is:*

$$P(I = i, L = l) = (1 - \rho_1)(1 - \rho_2)\rho_1^i \rho_2^l$$

where ρ_1 and ρ_2 are the solutions of (3.5).

Proof
Let π_{il} be the joint probability of the queue length and $p_{ss'}$ be the probability that an s-batch leaving node 1 arrives to node 2 as a s'-batch.
Then the flux into the middle (I, L) is:

$$\sum_{s=1}^{n_{d1}} \sum_{s'=1}^{n_{a2}} \pi_{i+s,l} d_{1;s} p_{ss'} = \sum_{s=1}^{n_{d1}} \sum_{s'=1}^{n_{a2}} (1-\rho_1)(1-\rho_2)\rho_1^{i+s}\rho_2^l d_{1;s} p_{ss'} = (1-\rho_1)(1-\rho_2)\rho_1^i \rho_2^l D_1(\rho_1)$$

while the total departure flux is:

$$\sum_{i=0}^{\infty} \sum_{l=0}^{\infty} (1-\rho_1)(1-\rho_2)\rho_1^i \rho_2^l D_1(\rho_1) = D_1(\rho_1)$$

The middle state probability is the ratio of the flux into the middle state to the total flux, which is $(1 - \rho_1)(1 - \rho_2)\rho_1^i \rho_2^l$ as stated. □

From now on we assume that a normal batch proceeds to node 2 without changing its size.

Proposition 4.2.2 *The size of an in-transit batch at equilibrium is independent of the middle state and has probability generating function:*

$$\frac{D_1(\rho_1 z)}{D_1(\rho_1)}$$

In addition, when the tagged task is at a random position within its arriving batch the probability generating function of the joint probability of (J, K) *is:*

$$F(z_1, z_2) = \frac{D_1(\rho_1 z_1) - D_1(\rho_1 z_2)}{\rho_1(z_1 - z_2)D'(\rho_1)}$$

Proof
The flux corresponding to an in-transit batch of size n – when the middle state is (i, l) – is $(1-\rho_1)(1-\rho_2)\rho_1^{i+n}\rho_2^l d_{1;n}$ and the total flux is $(1-\rho_1)(1-\rho_2)\rho_1^i\rho_2^l D_1(\rho^1)$. Hence, the probability that an in-transit batch has size n is $\frac{d_{1;n}\rho_1^n}{D_1(\rho_1)}$.
When the tagged task is at a random position, the probability generating function of its position within the batch is:

$$\sum_{s=1}^{n_{d1}} \frac{d_{1;s}\rho_1^s}{\rho_1 D_1'(\rho_1)} \sum_{j=0}^{s-1} z_1^j z_2^{s-1-m} = \sum_{s=1}^{n_{d1}} \frac{d_{1;s}\rho_1^s}{\rho_1 D_1'(\rho_1)} \frac{z_1^s - z_2^s}{z_1 - z_2} = \frac{D_1(\rho_1 z_1) - D_1(\rho_1 z_2)}{\rho_1 D_1'(\rho_1)(z_1 - z_2)}$$

□

4.2.1 Overtake-free path

As stated several times before, the paths are not overtake-free in the general case, but under certain conditions they can be overtake-free. In this section we look at the special case when these conditions are met, and calculate the LST of the joint sojourn time with little effort.
In order to achieve overtake-free paths, a task at position k in the second queue cannot be affected by later arrivals. That is, $\sum_{s=k}^{l} d_s + d_{l0} = c_k$ where l is the current queue length, and c_k can depend on at most k (but not l). We may as well write $\sum_{s=k}^{l} d_s + \sum_{s=1}^{k-1} d_s + d_{l0} = c_k$ since the extra term only depends on k. So, $d_{l0} = c_k - D(1) + \sum_{s=l+1}^{n_d} d_s$. If we choose c_k to be $D(1)$ we get the discard model back except that we do not distinguish between special and normal batches.

By using proposition 4.1.4 to obtain the generating function for both the reversed sojourn time at node 1 and the forward sojourn time at node 2, we are now able to calculate an exact result for the LST of the joint sojourn time in the case of overtake-free paths.

Proposition 4.2.3 *In a tandem pair of discard batch-queues defined by* A_1, D_1, A_2, D_2*, the LST of the joint probability distribution of the sojourn times of tasks arriving and departing in either a special or a normal batch is:*

$$R^*(\theta_1, \theta_2) = \frac{[A_1(\rho^{-1}) - A_1(1)][D_2(1) - D_2(\rho_2)]F(\rho_1^{-1}, \rho_2^{-1})}{[\theta_1 + A_1(\rho_1^{-1}) - A_1(1)][\theta_2 + D_2(1) - D_2(\rho_2)]} - (1-\rho_1)(1-\rho_2)\sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=0}^{j-1}\sum_{l=0}^{k-1} \tilde{g}_{1;i} g_{2;l} f_{jk} \rho_1^{i-j}\rho_2^{l-k}$$

where ρ_1, ρ_2 *are the solutions of* (3.5)*,* f_{jk} *is the joint probability of the tagged task's position in its in-transit batch and* $F(x, y)$ *is the probability generating function corresponding to the* f_{jk}*s.*

Finally, $\tilde{g}_{1;i}$ is the LST of the remaining reversed sojourn time of a task at position $(i+1)$ at node 1 and $g_{1;l}$ is the remaining forward sojourn time of a task at position $(l+1)$ at the second node.

Proof
Using the same notations as before, by the properties of the conditional expectation,

$$R^*(\theta_1,\theta_2) = \mathbb{E}\left[\mathbb{E}\left(e^{-\theta_1\tilde{R}_1(I,J,K,L)-\theta_2 R_2(I,J,K,L)}|I,J,K,L\right)\right]$$
$$= (1-\rho_1)(1-\rho_2)\sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=0}^{\infty}\sum_{l=0}^{\infty} f_{jk}\rho_1^i\rho_2^l\tilde{g}_{1;i+j}g_{2;l+k}$$

where $\tilde{R}_1(I,J,K,L), R_2(I,J,K,L)$ are the sojourn times given the states I,J,K,L, which are independent, since the paths are overtake-free. Furthermore, the joint probability of (I,J,K,L) is $f_{jk}(1-\rho_1)(1-\rho_2)\rho_1^i\rho_2^l$ by proposition 4.2.1 and 4.2.2.
Thus, by proposition 4.1.4 and its reversed counterpart,

$$R^*(\theta_1,\theta_2) = (1-\rho_1)(1-\rho_2)\sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=j}^{\infty}\sum_{l=k}^{\infty} f_{jk}\rho_1^{i-j}\rho_2^{l-k}\tilde{g}_{1;i}g_{2;l}$$
$$= (1-\rho_1)(1-\rho_2)\left[\sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=0}^{\infty}\sum_{l=0}^{\infty} f_{jk}\rho_1^{i-j}\rho_2^{l-k}\tilde{g}_{1;i}g_{2;l}\right.$$
$$\left. - \sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=0}^{j-1}\sum_{l=0}^{k-1} f_{jk}\rho_1^{i-j}\rho_2^{l-k}\tilde{g}_{1;i}g_{2;l}\right]$$
$$= (1-\rho_1)(1-\rho_2)\left[\frac{F(\rho_1^{-1},\rho_2^{-1})[A_1(\rho^{-1})-A_1(1)][D_2(1)-D_2(\rho_2)]}{(1-\rho_1)(1-\rho_2)[\theta_1+A_1(\rho_1^{-1})-A_1(1)][\theta_2+D_2(1)-D_2(\rho_2)]}\right.$$
$$\left. - \sum_{j=0}^{n_{d1}}\sum_{k=0}^{n_{d1}}\sum_{i=0}^{j-1}\sum_{l=0}^{k-1} f_{jk}\rho_1^{i-j}\rho_2^{l-k}\tilde{g}_{1;i}g_{2;l}\right]$$

In the case of a random task in a batch, $F(z_1,z_2)$ is given by Proposition 4.2.2. □

This result provides a fairly efficient way to calculate the LST of the joint distribution, since the first term is in closed form and the second term is a finite sum.
We get the LST of the total sojourn time by evaluating the function at $\theta_1=\theta_2=\theta$, $R^*(\theta,\theta) = \mathbb{E}\left(e^{-\theta(\tilde{R}_1+R_2)}\right)$.

4.2.2 Non-overtake-free paths

Finally, arriving to the general case where we take into account that the paths are not overtake-free, we first consider the marginal sojourn time at node 2 given the middle state I,L and batch position J,K. The method is similar to what we did in the case of a single queue with normal-to-normal paths, but notice that in this case the sojourn time also depends on the state I at the first queue.

Proposition 4.2.4 *In a tandem pair of discard batch-queues defined by A_1,D_1,A_2,D_2, the LST of the marginal distribution at node 2 – only considering the tasks leaving in a normal batch –*

given I, J, K at its transition instant has generating function $G_j(x, z; \theta)$ with respect to I and K:

$$\mathbb{L}(x, z; \theta)\boldsymbol{G}(x, z; \theta) = \boldsymbol{E}(x, z; \theta)$$

where I, J is as before, K is the sum of the number of tasks at the second node and before the tagged one in its transition batch ($K \leftarrow K + L$). Furthermore,
$\boldsymbol{G}(x, z; \theta) = (G_0(x, y; \theta), ..., G_{n_{d2}-1}(x, y; \theta))$,
$\mathbb{L}(x) = (A_1(1) + A_2(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z) + \theta)\mathbb{I} - \mathbb{M}_1(x) - \mathbb{K}_1(x) - \mathbb{M}_2 - \mathbb{K}_2$,
and $\mathbb{M}_1(x), \mathbb{K}_1(x), \mathbb{M}_2, \mathbb{K}_2$ are $n_{d2} \times n_{d2}$ matrices defined as follows:

$$\mathbb{M}_1(x) := \begin{cases} m_{1;ij}(x) = d_{1;j-i}x^{j-i} & 0 \le i \le n_{d2} - 1, i + 1 \le j \le \min(n_{d1} + i, n_{d2} - 1) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{K}_1(x) := \begin{cases} k_{1;ij}(x) = \sum_{s=n_{d2}}^{n_{d1}+i} d_{1;s-i}x^{s-i} & 0 \le i \le n_{d2} - 1, j = n_{d2} - 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{M}_2 := \begin{cases} m_{2;ij} = a_{2;j-i} & 0 \le i \le n_{d2} - 1, i + 1 \le j \le \min(n_{a2} + i, n_{d2} - 1) \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbb{K}_2 := \begin{cases} k_{2;ij} = \sum_{s=n_{d2}}^{n_{a2}+i} a_{2;s-i} & 0 \le i \le n_{d2} - 1, j = n_{d2} - 1 \\ 0 & \text{otherwise} \end{cases}$$

Finally, $\boldsymbol{E} = (e_1, ...e_{n_{d2}-1})$

$$\begin{aligned} e_j = & \sum_{i=1}^{n_{a1}-1} \left[a_{1;0i} - x^i \sum_{l=i+1}^{n_{a1}} a_{1;l}x^{-l} \right] g_{ij}(z; \theta) + \\ & \left[\frac{D_1(1) - D_1(x)}{1 - x} - A_1(x^{-1}) - A_{10}(1) \right] g_{0j}(z; \theta) + \\ & \frac{1}{1 - x} \left[\frac{D_2(1) - D_2(z)}{1 - z} - \sum_{l=1}^{n_{d2}} d_{2;l} \sum_{k=0}^{l-j-2} z^k \right] \end{aligned}$$

where the $g_{ij}(z; \theta) = \frac{1}{i!} \left. \frac{\partial^i G_j(x,z;\theta)}{\partial x^i} \right|_{x=0}$ *are unknown functions of z.*

Proof
Let $\tau_{ijk} = \mathbb{P}(R_2 \le t, N = 1 | I = i, J = j, K = k)$, where N is the indicator function of the event: *the tagged task departs in a normal batch*. Following the previous method we take an infinitesimal step during which the following events can happen:

1. normal/special arrival/departure of size s at node 1
2. normal arrival of size s at node 2
3. normal departure of size s at node 2 without the tagged task
4. normal departure of size s at node 2 with the tagged task

Using the memoryless property of the exponential distribution,

$$\tau_{ijk}(t + h) = (1 - h[A_1(1) + \mathbb{I}_{\{i=0\}}A_{10}(1) + A_2(1) + D_1(1)\mathbb{I}_{\{i>0\}} + D_2(1)])\tau_{ijk} +$$

$$h\sum_{s=1}^{n_{a1}} a_{1;s}\tau_{i+s,jk}(t) + \mathbb{I}_{\{i=0\}}h\sum_{s=1}^{n_{a1}-1} a_{1;0s}\tau_{s,jk}(t) + h\sum_{s=1}^{n_{a2}} a_{2;s}\tau_{ij+s,k}(t) +$$
$$\mathbb{I}_{\{i>0\}}h\sum_{s=1}^{\min(i,n_{d1})} d_{1;s}\tau_{i-s,j+s,k}(t) + \mathbb{I}_{\{i>0\}}h\sum_{s=i+1}^{n_{d1}} d_{1;s}\tau_{0,jk}(t) +$$
$$h\sum_{s=1}^{\min(k,n_{d2})} d_{2;s}\tau_{i,j,k-s}(t) + h\sum_{s=k+1}^{\min(j+k+1,n_{d2})} d_{2;s} + o(h)$$

where we used that the degree of A_{i0} is $n_{ai}-1$ by proposition 3.2.2.
After rearranging, dividing by h, taking the limit $h \to 0$ and dropping θ for brevity we have

$$[\theta + A_1(1) + \mathbb{I}_{\{i=0\}}A_{10}(1) + A_2(1) + D_1(1) + D_2(1)]\tau^*_{ijk} = \sum_{s=1}^{n_{a1}} a_{1;s}\tau^*_{i+s,jk} +$$
$$\mathbb{I}_{\{i=0\}}\sum_{s=1}^{n_{a1}-1} a_{1;0s}\tau^*_{sjk} + \sum_{s=1}^{n_{a2}} a_{2;s}\tau^*_{i,j+s,k} + \sum_{s=1}^{\min(i,n_{d1})} d_{1;s}\tau^*_{i-s,j+s,k} + \sum_{s=i+1}^{n_{d1}} d_{1;s}\tau^*_{0jk} +$$
$$\sum_{s=1}^{\min(k,n_{d2})} d_{2;s}\tau^*_{ij,k-s} + \sum_{s=k+1}^{\min(j+k+1,n_{d2})} d_{2;s}$$

Note that we omit the indicator $\mathbb{I}_{\{i>0\}}$, but the extra terms cancel each other out. We once again use the fact that any arrivals later than $n_{d2}-1$ after the tagged task do not affect its sojourn time. Therefore, $G_j(x,z;\theta) = G_{n_{d2}-1}(x,z;\theta)$ when $j \geq n_{d2}-1$.

Next, we multiply by $x^i z^k$ and sum from $i=0$ to ∞ and $k=0$ to ∞.

Starting on the left hand side:

$$\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\mathbb{I}_{\{i=0\}}A_{10}(1)\tau^*_{ijk} = \sum_{i=0}^{\infty}\sum_{k=0}^{\infty}A_{10}(1)\tau^*_{0jk} = A_{10}(1)g_{0j}(z)$$

Now, looking at the right hand side and reducing it term by term:

1.

$$\begin{aligned}\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=1}^{n_{a1}} a_{1;s}\tau^*_{i+s,jk}x^i z^k &= \sum_{k=0}^{\infty}\sum_{s=1}^{n_{a1}}\sum_{i=s}^{\infty} a_{1;s}\tau^*_{ijk}x^{i-s}z^k \\ &= \sum_{k=0}^{\infty}\sum_{s=1}^{n_{a1}}\sum_{i=0}^{\infty} a_{1;s}\tau^*_{ijk}x^{i-s}z^k - \sum_{s=1}^{n_{a1}}\sum_{i=0}^{s-1} a_{1;s}x^{i-s}\sum_{k=0}^{\infty}\tau^*_{ijk}z^k \\ &= A_1(x^{-1})G_j(x,z) - \sum_{i=0}^{n_{a1}} g_{ij}(z)\sum_{s=i+1}^{n_{a1}} a_{1;s}x^{-(s-i)}\end{aligned}$$

The first term appears on the left hand side, while the second one is part of $\boldsymbol{E}$.

2.

$$\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\mathbb{I}_{\{i=0\}}\sum_{s=1}^{n_{a1}-1} a_{1;0s}\tau^*_{sjk}x^i z^k = \sum_{k=0}^{\infty}\sum_{s=1}^{n_{a1}-1} a_{1;0s}\tau^*_{sjk}z^k = \sum_{i=1}^{n_{a1}-1} a_{1;0s}g_{ij}(z)$$

This appears in $\boldsymbol{E}$ combined with the previous term.

3.

$$\begin{aligned}\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=1}^{n_{a2}} a_{2;s}\tau^*_{i,j+s,k}x^i z^k &= \sum_{s=1}^{n_{a2}} a_{2;s}G_{\min(j+s,n_{d2}-1)}(x,z)\\ &= \sum_{s=j+1}^{n_{a2}+j} a_{2;s-j}G_{\min(s,n_{d2}-1)}(x,z)\\ &= \sum_{s=j+1}^{\min(n_{a2}+j,n_{d2}-1)} a_{2;s-j}G_s(x,z) + \sum_{s=n_{d2}}^{n_{a2}+j} a_{2;s-j}G_{n_{d2}-1}(x,z)\end{aligned}$$

These terms are precisely the matrices $\mathbb{M}_2$, $\mathbb{K}_2$ on the left hand side multiplied by $\boldsymbol{G}$.

4.

$$\begin{aligned}&\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=1}^{\min(i,n_{d1})} d_{1;s}\tau^*_{i-s,j+s,k}x^i z^k = \sum_{k=0}^{\infty}\sum_{s=1}^{n_{d1}}\sum_{i=s}^{\infty} d_{1;s}\tau^*_{i-s,j+s,k}x^i z^k\\ &= \sum_{k=0}^{\infty}\sum_{s=1}^{n_{d1}}\sum_{i=0}^{\infty} d_{1;s}\tau^*_{i,j+s,k}x^{i+s} z^k = \sum_{s=j+1}^{j+n_{d1}} d_{1;s-j}x^{s-j}G_{\min(s,n_{d2}-1)}(x,z)\\ &= \sum_{s=j+1}^{\min(j+n_{d1},n_{d2}-1)} d_{1;s-j}x^{s-j}G_s(x,z) + \sum_{s=n_{d2}}^{j+n_{d1}} d_{1;s-j}x^{s-j}G_{n_{d2}-1}(x,z)\end{aligned}$$

These terms are precisely the matrices $\mathbb{M}_1(x)$, $\mathbb{K}_1(x)$ on the left hand side multiplied by $\boldsymbol{G}$.

5.

$$\begin{aligned}\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=i+1}^{n_{d1}} d_{1;s}\tau^*_{0jk}x^i z^k &= \sum_{k=0}^{\infty}\sum_{s=1}^{n_{d1}}\sum_{i=0}^{s-1} d_{1;s}\tau^*_{0jk}x^i z^k\\ &= \sum_{k=0}^{\infty}\sum_{s=1}^{n_{d1}} d_{1;s}\tau^*_{0jk}z^k\frac{1-x^s}{1-x} = \frac{D_1(1)-D_1(x)}{1-x}g_{0j}(z)\end{aligned}$$

These terms appear on the right hand side of the equation as part of $\boldsymbol{E}$.

6.

$$\begin{aligned}\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=1}^{\min(k,n_{d2})} d_{2;s}\tau^*_{ij,k-s}x^i z^k &= \sum_{i=0}^{\infty}\sum_{s=1}^{n_{d2}}\sum_{k=s}^{\infty} d_{2;s}\tau^*_{ij,k-s}x^i z^k\\ &= \sum_{i=0}^{\infty}\sum_{s=1}^{n_{d2}}\sum_{k=0}^{\infty} d_{2;s}\tau^*_{ij,k}x^i z^{k+s} = D_2(z)G_j(x,z)\end{aligned}$$

This is part of the left hand side.

7.

$$\sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\sum_{s=k+1}^{\min(j+k+1,n_{d2})} d_{2;s}x^i z^k = \frac{1}{1-x}\sum_{k=0}^{\infty}\sum_{s=k+1}^{\min(j+k+1,n_{d2})} d_{2;s}z^k$$
$$= \frac{1}{1-x}\sum_{s=1}^{n_{d2}} d_{2;s}\sum_{k=\max(s-j-1,0)}^{s-1} z^k = \frac{1}{1-x}\sum_{s=1}^{n_{d2}} d_{2;s}\left(\frac{1-z^s}{1-z} - \sum_{k=0}^{s-j-2} z^k\right)$$
$$= \frac{1}{1-x}\left(\frac{D_2(1)-D_2(z)}{1-z} - \sum_{s=1}^{n_{d2}} d_{2;s}\sum_{k=0}^{s-j-2} z^k\right)$$

These terms also appear on the right hand side of the equation.

□

The above proposition and corresponding proof is where Harrison's work has been left off. Although the proof is omitted in [5] it was provided for this project. The material that follows continues this work of Harrison's.

Determining the unknown $g_{ij}(z;\theta)$ and $G_j(x,z;\theta)$ functions

$\mathbb{L}(x,z)$ is upper triangular and the diagonal elements are all identical except the last one. We denote them by $f_1(x)$ and $f_2(x)$ respectively, treating z as a parameter. We find the unknown functions going row by row starting at the lowermost one.

1. Finding $G_{n_{d2}-1}$:

   $f_2(x)G_{n_{d2}-1}(x,z) = e_{n_{d2}-1}$ and only $g_{i,n_{d2}-1}$ $0 \le i \le n_{a1}-1$ appear on the right hand side. Claiming that $f_2(x)$ has n_{a1} roots within the unit disk, we can determine all $g_{i,n_{d2}-1}$s and then express $G_{n_{d2}-1}(x,z) = e_{n_{d2}-1}/f_2(x)$ for all x except the roots of $f_2(x)$ where it can be approximated.

2. Determining $G_k(x,z)$ when all $G_s(x,z)$, $s > k$ are already known:

   $f_1(x)G_k = e_k - (u_1G_{k+1} + ... + u_{n_{d2}-k-1}G_{n_{d2}-1})$ where the u_s coefficients are from the kth row of $\mathbb{L}$. Again, we claim that $f_1(x)$ has n_{a1} roots inside the unit disk, therefore all the unknown g_{ik} functions can be determined in e_k and G_k can be expressed as before. G_k has to be approximated at the roots of f_1 again.

We appeal to Rouche's theorem to show that there are enough roots in the unit disk in both cases.

Proposition 4.2.5 (Rouche's theorem) *Let f, g be holomorphic functions inside some region K and $|g| < |f|$ on ∂K, then $f+g$ and f have the same number of roots inside K.*

In our case K is the unit disk and by the definitions of $\mathbb{L}$:

$$f_1(x) = A_1(1) + A_2(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z) + \theta$$

First, we multiply it by $x^{n_{a1}}$ to get rid of the negative powers of x. Using the notations of the theorem above,

$$f(x) = [A_1(1) + A_2(1) + D_1(1) + D_2(1) - D_2(z) + \theta]x^{n_{a1}}$$

$$g(x) = -A_1(x^{-1})x^{n_{a1}} = \sum_{s=1}^{n_{a1}} -a_{1;s}x^{n_{a1}-s}$$

In order to use the theorem, we have to prove that f is strictly larger than g when $|x| = 1$. If this condition is met, $f_1(x)x^{n_{a1}} = f(x) + g(x)$ has the same number of roots in the unit disk as f which has at least n_{a1}, given that 0 is a root with n_{a1} multiplicity. Furthermore, evaluating $f_1(x)x^{n_{a1}}$ at 0 we get $-a_{1;n_{a1}} \neq 0$. Hence, $f_1(x)$ has at least n_{a1} roots in the unit disk as required.

$$\begin{aligned}
&|[A_1(1) + A_2(1) + D_1(1) + D_2(1) - D_2(z) + \theta]x^{n_{a1}}| \geq \text{ by the reversed triangle inequality}\\
&|A_1(1) + A_2(1) + D_1(1) + D_2(1) + \theta| - |D_2(z)| \geq \text{ assuming } \mathrm{Re}(\theta) \geq 0 \text{ and } |z| \leq 1\\
&A_1(1) + A_2(1) + D_1(1) + D_2(1) - D_2(1) > A_1(1) \geq |g(x)|
\end{aligned}$$

The same has to be proven for $f_2(x)$ too.

$$f_2(x) = A_1(1) + D_1(1) + D_2(1) - A_1(x^{-1}) - D_2(z) + \theta - D_1(x)$$

We follow the same steps. Making exactly the same argument, it is enough to prove that $f_2(x)x^{n_{a1}}$ has at least n_{a1} roots inside the unit disk.
Now, we have:

$$\begin{aligned}
f(x) &= [A_1(1) + D_1(1) + D_2(1) - D_2(z) + \theta]x^{n_{a1}}\\
g(x) &= (-A_1(x^{-1}) - D_1(x))x^{n_{a1}} = \sum_{s=1}^{n_{a1}} -a_{1;s}x^{n_{a1}-s} + \sum_{s=1}^{n_{d1}} -d_{1;s}x^{n_{a1}+s}
\end{aligned}$$

Similarly, we get

$$\begin{aligned}
&|f(x)| \geq |A_1(1) + D_1(1) + D_2(1) + \theta| - |D_2(z)| > \text{ assuming } \mathrm{Re}(\theta) \geq \varepsilon \text{ and } |z| \leq 1\\
&A_1(1) + D_1(1) + D_2(1) - D_2(1) \geq A_1(1) + D_1(1) \geq |g(x)|
\end{aligned}$$

Therefore, we have enough roots to determine all g_{ij}s and then solve the equations for G_j.

Expressing the probability distribution with $G_j(x, z)$

By the law of total probability and using proposition 4.2.1, the LST of the probability distribution $T_{jk}(t) = P(R_2 \leq t, N = 1|J = j, K = k)$ and both j and k are bounded by n_{d1}, which is the maximum batch size for the departure process at the first queue:

$$\begin{aligned}
&T^*_{jk}(\theta)\\
&= \sum_{i=0}^{\infty}\sum_{l=0}^{\infty}(1-\rho_1)(1-\rho_2)\rho_1^i\rho_2^l\tau_{ij,k+l}(\theta) = (1-\rho_1)(1-\rho_2)\rho_2^{-k}\sum_{i=0}^{\infty}\sum_{l=k}^{\infty}\rho_1^i\rho_2^l\tau_{ijl}(\theta)\\
&= (1-\rho_1)(1-\rho_2)\rho_2^{-k}\left(G_{\min(j,n_{d2}-1)}(\rho_1,\rho_2;\theta) - \sum_{l=0}^{k-1}\rho_2^l\frac{1}{l!}\left.\frac{\partial^l G_{\min(j,n_{d2}-1)}(\rho_1,z;\theta)}{\partial z^l}\right|_{z=0}\right)
\end{aligned}$$

Therefore we get the LST after unconditioning with respect to J, K, and normalizing with the probability that the tagged task departs in a normal batch is:

$$T^*(\theta)/T^*(0) = \sum_{j=0}^{n_{d1}-1} \sum_{k=0}^{n_{d1}-1-j} f_{jk} T^*_{jk}(\theta)/T^*(0)$$

In the case of the first or last task in the batch f_{jk} is determined by 4.2.2.

Joint probability distribution

Now, we have the generating function of both the reversed process at the first queue and the forward process at the second queue in closed forms. We will look at the LST of the joint probability distribution and see if it can be expressed by the aforementioned generating functions.

Note that since the process is Markovian, once we condition on the present state of I, J, K, L the forward and reversed sojourn time random variables are independent.

Hence, we are seeking a closed form for the following expression, omitting θ and the tildes from the top of the gamma functions (which we used before to denote that they corresponded to the reversed process) for brevity:

$$\begin{aligned}
T^*_{jk} &= \sum_{i=0}^{\infty} \sum_{l=0}^{\infty} \gamma^*_{i+j,k} \tau^*_{ij,k+l} (1-\rho_1)\rho_1^i (1-\rho_2)\rho_2^l && (4.6)\\
&= \sum_{i=0}^{\infty} \sum_{l=k}^{\infty} \gamma^*_{i+j,k} \tau^*_{ijl} (1-\rho_1)\rho_1^i (1-\rho_2)\rho_2^{l-k} && (4.7)\\
&= (1-\rho_1)(1-\rho_2)\rho_2^{-k} \left[\sum_{i=0}^{\infty} \sum_{l=0}^{\infty} \gamma^*_{i+j,k} \tau^*_{ijl} \rho_1^i \rho_2^l - \sum_{i=0}^{\infty} \sum_{l=0}^{k-1} \gamma^*_{i+j,k} \tau^*_{ijl} \rho_1^i \rho_2^l \right] && (4.8)\\
&= (1-\rho_1)(1-\rho_2)\rho_2^{-k} \left[F_{jk}(\rho_1,\rho_2) - \sum_{l=0}^{k-1} \frac{\rho_2^l}{l!} \left. \frac{\partial^l F_{jk}(\rho_1, z)}{\partial z^l} \right|_{z=0} \right] && (4.9)
\end{aligned}$$

where a closed form for $F^*_{jk}(x,y) = \sum_{i=0}^{\infty} \sum_{l=0}^{\infty} \gamma^*_{i+j,k} \tau^*_{ijl} x^i y^l$ is not yet known.

The obvious problem with this scenario is that the sojourn times are only conditionally independent. Therefore, we need to be able to extract infinitely many coefficients (namely each coefficient for which $i + j = i'$) from the expression below:

$$G_j(x,z)H_k(y) = \sum_{i=0}^{\infty} \sum_{l=0}^{\infty} \sum_{i'=0}^{\infty} \tau^*_{ijl} \gamma^*_{i'k} x^i z^l y^{i'}$$

This difficulty will be overcome shortly, but let us consider two special cases first, neither of which requires dealing with the issue of conditional independence.

M/M/1 case

First, we look at the case when both queues are M/M/1 queues with arrival and departure rates $\lambda_1, \lambda_2, \mu_1, \mu_2$ respectively. Thus, the reversed rates of the first queue are $\mu_1\rho_1$ and $\lambda_1\rho_1^{-1}$ by Proposition 3.2.3.
All batches have size one, hence $n_{a1} = n_{a2} = n_{d1} = n_{d2} = 1$. Using the matrix form equation (4.4) for the reversed part we have:

$$(\mu_1\rho_1 + \lambda_1\rho_1^{-1} - \lambda_1\rho_1^{-1}x + \theta - \mu_1\rho_1)H_0(x;\theta) = \lambda_1\rho_1^{-1}$$
$$H_0(x;\theta) = \frac{\lambda_1\rho_1^{-1}}{[\lambda_1\rho_1^{-1}(1-x) + \theta]}$$

Therefore, evaluating the generating function at ρ_1 and multiplying by $(1-\rho_1)$ we get

$$\widetilde{R}_1^*(\theta) = (1-\rho_1)H_0(\rho_1;\theta) = (1-\rho_1)\frac{\lambda_1\rho_1^{-1}}{(\lambda_1\rho_1^{-1}(1-\rho_1)+\theta)} = \frac{\mu_1-\lambda_1}{\mu_1-\lambda_1+\theta} \tag{4.10}$$

Note that in case of the first queue in the tandem network the parameter of the equilibrium queue length probability calculated by the product form result is the same as if we calculated it considering the first queue alone. Therefore, it is valid to use the $\rho_1 = \lambda_1/\mu_1$ identity.
By Proposition 4.2.4 in the case of the forward sojourn time at the second queue we get

$$\left[\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \frac{\lambda_1}{x} - \mu_2 z + \theta - \mu_1 x - \lambda_2\right] G_0(x,z;\theta)$$
$$= \left[\mu_1 - \frac{\lambda_1}{x}\right] G_0(0,z;\theta) + \frac{\mu_2}{1-x}$$

Plugging in x_0 for which $\lambda_1 + \mu_1 + \mu_2 - \frac{\lambda_1}{x_0} - \mu_2 z + \theta - \mu_1 x_0 = 0$

$$G_0(0,z;\theta) = \frac{\mu_2}{x_0-1}\frac{1}{\left[\mu_1 - \frac{\lambda_1}{x_0}\right]} = \frac{\mu_2}{x_0\mu_1 - \mu_1 - \lambda_1 + \frac{\lambda_1}{x_0}} = \frac{\mu_2}{\mu_2(1-z)+\theta}$$

Thus,

$$G_0(x,z;\theta) = \frac{\mu_2[(\mu_1 - \frac{\lambda_1}{x})(1-x) + \theta + \mu_2(1-z)]}{(1-x)[\theta + \mu_2(1-z)]}\frac{1}{(\lambda_1 + \mu_1 + \mu_2 - \frac{\lambda_1}{x} - \mu_2 z + \theta - \mu_1 x)}$$
$$= \frac{\mu_2}{(1-x)[\theta + \mu_2(1-z)]}$$

The sojourn time at the second queue is independent of the length of the first queue now; therefore $\tau_{ijl} = \tau_{0jl}$.
This can also be shown through the generating function, since

$$R_2^*(\theta) = (1-\rho_1)(1-\rho_2)G_0(\rho_1,\rho_2;\theta) = \frac{\mu_2(1-\rho_2)}{\theta + \mu_2(1-\rho_2)} \tag{4.11}$$

which is precisely what we get by ignoring the first queue and using $G_0(0,z;\theta)$ instead of the general generating function.
By formula 4.9 and keeping in mind that τ_{ijl} does not depend on i, the required LST is:

$$
\begin{aligned}
(1-\rho_1)(1-\rho_2)F^*_{00}(\theta) &= (1-\rho_1)(1-\rho_2)\sum_{i=0}^{\infty}\sum_{l=0}^{\infty}\gamma_{i0}\tau_{i0l}\rho_1^i\rho_2^l && (4.12)\\
&= (1-\rho_1)(1-\rho_2)H_0(\rho_1;\theta)G_0(0,\rho_2;\theta) && (4.13)\\
&= \frac{\mu_1-\lambda_1}{\theta+\mu_1-\lambda_1}\frac{\mu_2(1-\rho_2)}{\theta+\mu_2(1-\rho_2)} && (4.14)
\end{aligned}
$$

as expected. Since this is a well known result, it also acts as a verification of the reliability of the model.

Note that ρ_2 is different from the parameter of the equilibrium probability distribution if the second queue was considered alone. Hence, in this case we cannot simplify with it.

We can take one more step towards the general solution with little effort. If we leave the first queue as an M/M/1 queue but let the second one be a general batch queue we can still obtain a closed form for the required LST.
Considering the γ_{ij} coefficients individually and noting that each is the sum of both i exponential variables (with parameter $\lambda_1\rho_1$) and the remaining service time of the current task at the server which is also exponential with the same parameter due to the memoryless property of the distribution:

$$
\gamma_{ij}(\theta) = \left(\frac{\lambda_1\rho_1^{-1}}{\theta+\lambda_1\rho_1^{-1}}\right)^{i+1} = \left(\frac{\mu_1}{\theta+\mu_1}\right)^{i+1}
$$

Plugging this into the formula we get

$$
\begin{aligned}
(1-\rho_1)(1-\rho_2)F^*_{j0}(\theta) &= (1-\rho_1)(1-\rho_2)\sum_{i=0}^{\infty}\sum_{l=0}^{\infty}\gamma_{i+j,0}\tau_{ijl}\rho_1^i\rho_2^l\\
&= (1-\rho_1)(1-\rho_2)\sum_{i=0}^{\infty}\sum_{l=0}^{\infty}\left[\frac{\mu_1}{\theta+\mu_1}\right]^{i+j+1}\tau_{ijl}\rho_1^i\rho_2^l\\
&= (1-\rho_1)(1-\rho_2)\left[\frac{\mu_1}{\theta+\mu_1}\right]^{j+1}G_j\left(\frac{\mu_1\rho_1}{\theta+\mu_1},\rho_2;\theta\right)
\end{aligned}
$$

General case

Returning to the original problem, we have to extract the coefficients for which $i+j=i'$ from $R_{jk}(x,z,y) = G_j(x,z)H_k(y) = \sum_{i=0}^{\infty}\sum_{l=0}^{\infty}\sum_{i'=0}^{\infty}\tau^*_{ijl}\gamma^*_{i'k}x^iz^ly^{i'}$.
We achieve this by evaluating a complex integral.

Proposition 4.2.6

$$
\int_0^{2\pi} e^{ikt}dt = \begin{cases} 2\pi & k=0\\ 0 & k\neq 0\end{cases}
$$

where i is the imaginary unit and $k\in\mathbb{Z}$.

Using the proposition above, evaluating $R_{jk}(x,z,y)$ at $x = re^{it}, y = re^{-it}$ where $r \in \mathbb{R}, r < 1$ and then integrating from $t = 0$ to $t = 2\pi$ we have
$cx^m y^n = cr^{m+n} e^{i(m-n)} \begin{cases} 0 & m \neq n \\ c2\pi r^{2m} & m = n \end{cases}$
where c denotes everything except the powers of x and y in each term in the infinite sum. Hence, in the case $j = 0$

$$F_{0k}(r^2, z) = \frac{1}{2\pi} \int_0^{2\pi} R_{0k}(re^{it}, z, re^{-it}) dt$$

and in the general case,

$$F_{jk}(r^2, z) = \frac{1}{2\pi} \int_0^{2\pi} R_{jk}(re^{it}, z, re^{-it}) \left[re^{it}\right]^j dt \tag{4.15}$$

Thus, plugging (4.15) into (4.9) we have a closed formula for the LST of the joint probability distribution.

4.3 Numerical results and application

Both the single queue model and the more complicated one for the joint probability distribution have been programmed in Mathematica.
We use the following method when calculating the density functions:

1. Find the parameters ρ or (ρ_1, ρ_2) of the geometric equilibrium distribution. (When there are no solutions for ρ or (ρ_1, ρ_2) we cannot proceed any further.)
2. Calculate the generating functions.
3. Express the LST of the distribution with the generating functions.
4. Invert the LST numerically.

In the case of a single queue we use arrival rates: $a_1 = 2, a_2 = 1, a_3 = 1$ and $a_4 = 1$ and departure rates: $d_1 = 20$ and $d_2 = 10$.
Figures 4.3 and 4.4 show the density of the sojourn time considering the last and a random task in the batch, respectively. As can be seen considering a random task instead of the last task does not make a huge difference; the density function in the first case is slightly flatter.
Next, we consider a tandem batch-network with arrival rates: $a_{1;1} = 2, a_{1;2} = 1, a_{1;3} = 1, a_{1;4} = 1$; $a_{2;1} = 3, a_{2;2} = 1$ and departure rates: $d_{1;1} = 20, d_{1;2} = 10$; $d_{2;1} = 8, d_{2;2} = 4, d_{2;3} = 1, d_{3;4} = 1, d_{2;5} = 2$, for the first and the second processes respectively. We calculate the density of the sojourn time for the reversed process at the first node considering the last task in the batch. The results are shown in figure 4.5.
In the case of the joint probability distribution, numerical stability issues were encountered during the inversion of the LST. The most challenging part computationally is the determination of the vector of generating functions in the case of the forward sojourn time at the second node.

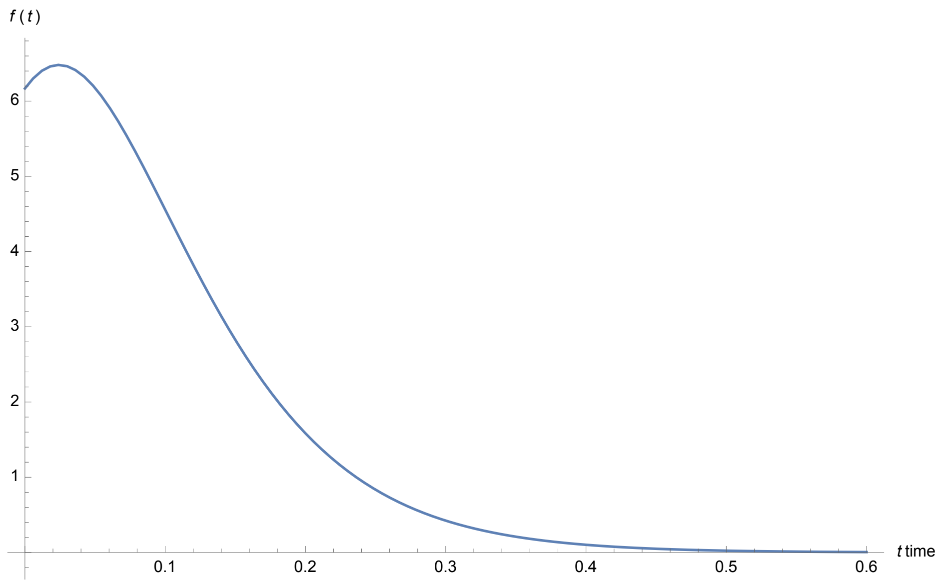

Figure 4.3: Density function of the sojourn time random variable in case of a single queue with arrival rates $a_1 = 2, a_2 = 1, a_3 = 1, a_4 = 1$ and departure rates $d_1 = 20, d_2 = 10$ considering the last task.

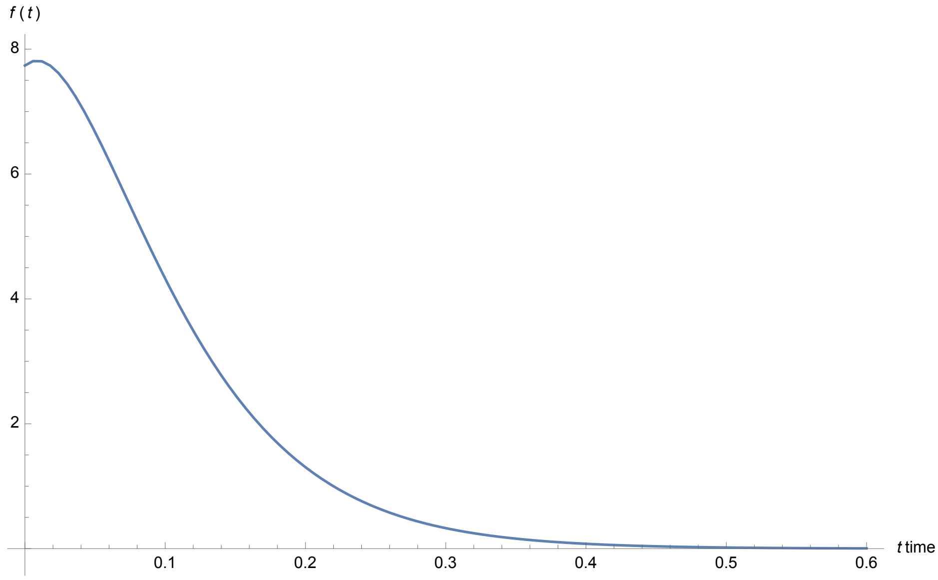

Figure 4.4: Density function of the sojourn time random variable in case of a single queue with arrival rates $a_1 = 2, a_2 = 1, a_3 = 1, a_4 = 1$ and departure rates $d_1 = 20, d_2 = 10$ considering a random task.

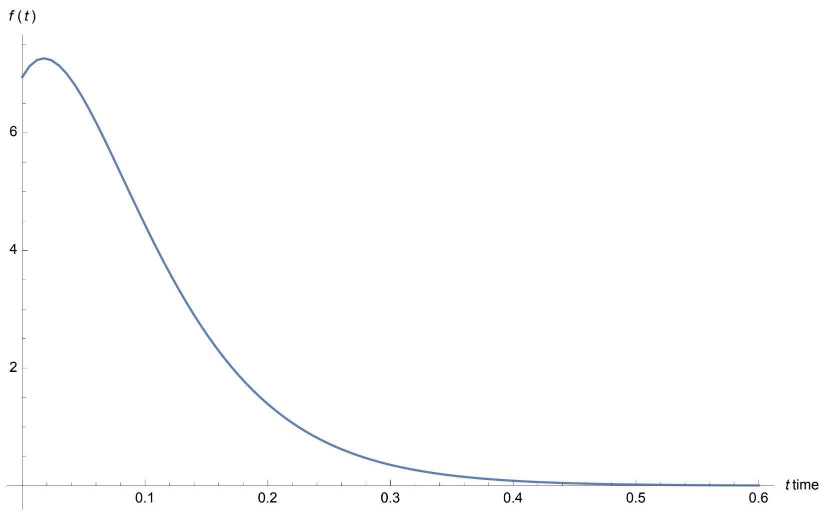

Figure 4.5: Density function of the sojourn time random variable of the reversed process at the first queue.

However, for the simple case of two M/M/1 queues in a tandem network the results are quite accurate. The arrival rates now are $a_{1;1} = 1.5, a_{2;1} = 1$ and the departure rates are $d_{1;1} = 3, d_{2;1} = 3$.

Figure 4.6 shows the results obtained by calculating the LST of the forward and reversed sojourn times and then inverting it numerically for 300 points between 0 and 10. For figure 4.7, we invert the exact LST given by equation (4.14). As can be seen, the density function calculated by the model is really close to the exact solution, which makes the method itself highly promising.

Finally, we consider a tandem network with rates $a_{1;1} = 2, a_{1;2} = 1$; $a_{2;1} = 3, a_{2;2} = 1$ and $d_{1;1} = 2, d_{1;2} = 10$; $d_{2;1} = 8, d_{2;2} = 1$.

Figure 4.8 shows the marginal density of the sojourn time at the second queue while figure 4.9 presents the density function of the joint distribution inverting the LST at $\theta_1 = \theta_2$. In other words, this figure shows the density function of the sum of the two sojourn time random variables. There are some spikes in the graphs due to the aforementioned numerical instability during the inversion of the Laplace transforms. This instability only appears in the case of complex θs which makes it difficult to fix the problem. Overcome the difficulties caused by the inversion, was not part of this project but it will be investigated in the future. We intend to try different available inversion methods and/or approximate the density function by the first 4-5 moments if necessary.

Application

Tandem batch-networks have been used to model wireless sensor networks to find the optimal centralized or decentralized sensor scheduling [16].

Furthermore, batch networks are suitable to model bursty traffic which occurs in several types of networks such as IP networks [13] or data transfer in storage systems [15]. Data centers consume a vast amount of energy, so it is essential to make them as efficient as possible in terms of energy usage. One possible way of saving energy is using devices with multiple power levels of operation: *on*, *off* and possibly an intermediate *sleep* state. This allows the devices to be switched off when they are not in use. However, in case of steady traffic – no long idle periods – switching off the device is not beneficial. In fact, since it

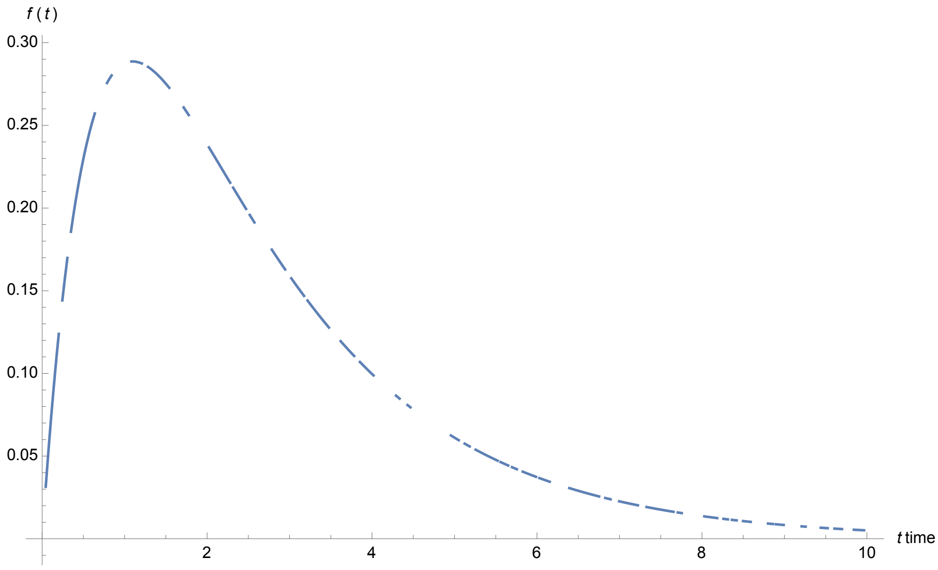

Figure 4.6: Density function calculated by the generating functions in the case of two M/M/1 queues with rates: $a_{1;1} = 1.5, a_{2;1} = 1$ and $d_{1;1} = 3, d_{2;1} = 3$.

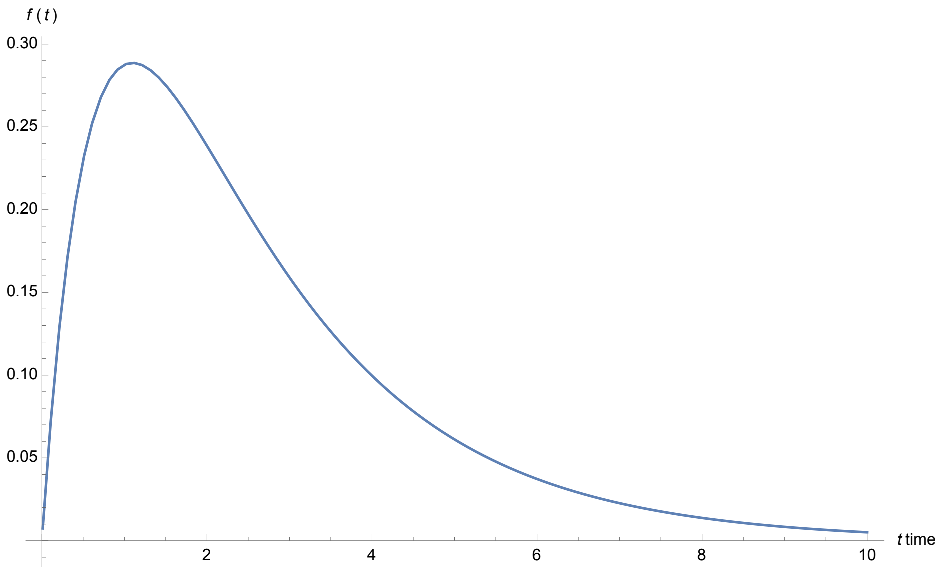

Figure 4.7: Density function inverting the exact LST in the case of two M/M/1 queues with rates: $a_{1;1} = 1.5, a_{2;1} = 1$ and $d_{1;1} = 3, d_{2;1} = 3$.

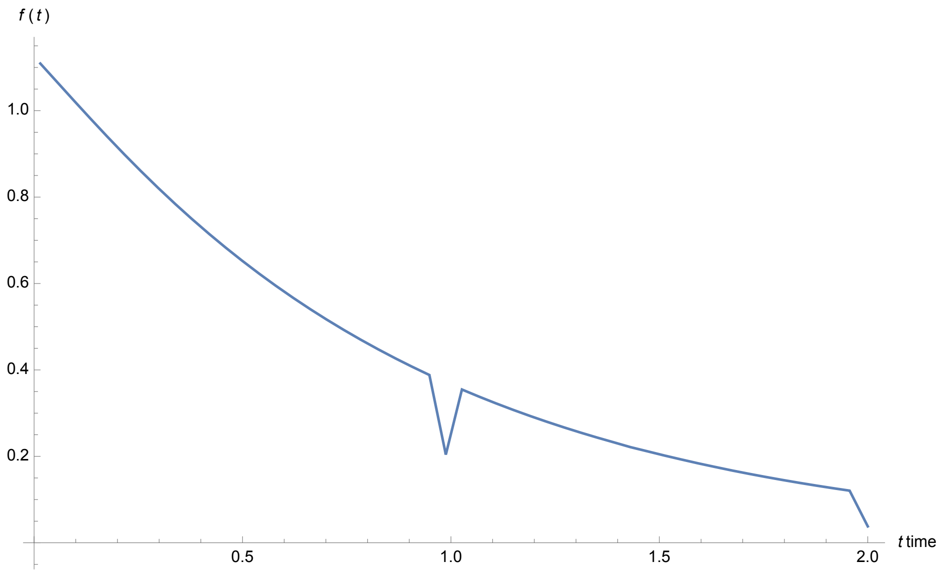

Figure 4.8: Density function of the marginal sojourn time at the second node. The rates are: $a_{1;1} = 2, a_{1;2} = 1$; $a_{2;1} = 3, a_{2;2} = 1$ and $d_{1;1} = 2, d_{1;2} = 10$; $d_{2;1} = 8, d_{2;2} = 1$.

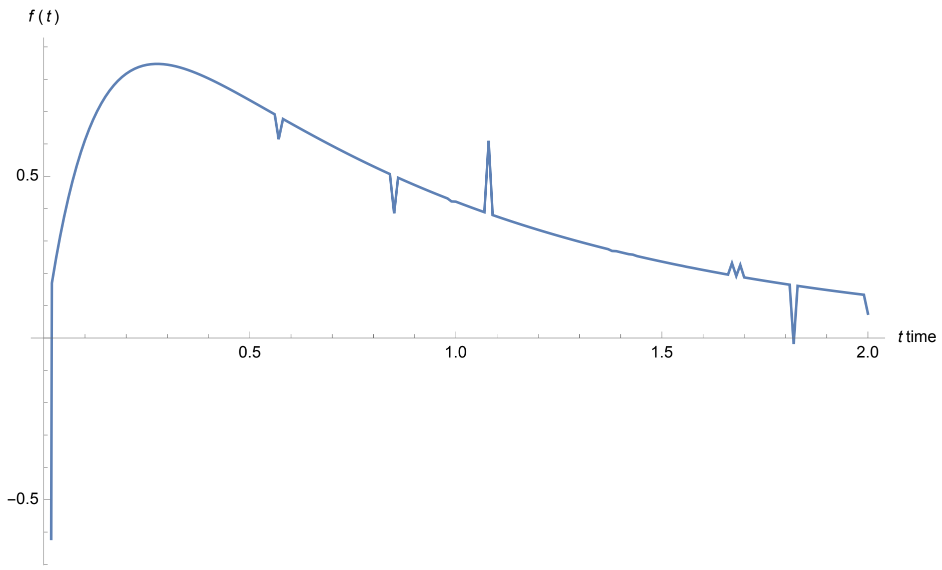

Figure 4.9: Density function of the sum of sojourn time random variables for rates $a_{1;1} = 2, a_{1;2} = 1$; $a_{2;1} = 3, a_{2;2} = 1$ and $d_{1;1} = 2, d_{1;2} = 10$; $d_{2;1} = 8, d_{2;2} = 1$ in a tandem network of batch-queues.

has to be powered up again quickly, switching off could potentially cause energy overheads, penalizing the unprovoked changing of states [17].
Therefore, it is important to model these types of networks in order to predict the load of traffic in the future and hence be able to construct an efficient scheduling algorithm. Using the batch queue model, regular traffic can be represented by normal batch arrivals. Furthermore, when a device is switched off some tasks already sent to the device might get lost, and similarly when it is switched on again there can be a backlog of work which causes a sudden burst of activity. This atypical behavior is well modeled by the special arrivals and departures.
One could argue that the odds that a real-world system satisfies the conditions of the product form theorem are low. However, modeling a network with exact parameters is too expensive in most cases, which means that a good approximation is important and useful.

Chapter 5

Summary

We started the thesis by laying down the necessary mathematical background in Chapter 2, which covered the most relevant definitions and theorems of stochastic processes and queuing theory. The aim in Chapter 3 was to state the product-form theorem, since an equilibrium queue length probability distribution in product form makes it possible to use the generating function method to obtain the LSTs of the unconditional sojourn time distribution for the whole network (as we do in Chapter 4). We start by describing the reversal of a stochastic process and give a proper definition of the batch-queue model, noting that special arrivals and departures must be allowed in order to be able to obtain a product-form equilibrium queue length distribution. Next, we look at what reversing the process means in the case of a single batch-queue. To obtain a product-form result for the tandem network we appeal to RCAT. We make the observation that a geometric queue length distribution at equilibrium ensures that the conditions of RCAT are satisfied. In addition, this also allows us to express the LST of the unconditional sojourn time distribution in terms of the generating functions. Hence, we seek conditions under which the equilibrium queue length distribution is geometric, first considering a single queue and then extending the result for a network of queues using RCAT.

Chapter 4 then focuses on the sojourn times of batch queues, first considering a single batch queue and its reversed counterpart. We then analyze the challenges faced in the case of the tandem network and obtain the LST of the marginal sojourn time distribution at the second node. This partial result required solving a recurrence of a vector of generating functions and Rouche's theorem was used to show that a solution can be obtained even though the matrix in the equation is singular.

After considering a few special cases with M/M/1 queues in the tandem network and overtake-free paths, we return to the original problem and use the results for the reversed process at the first node and the forward process at the second node together. We evaluate a complex integral to overcome the difficulties caused by the non-overtake-free paths or, in other words, the lack of unconditional independence between the two processes, and therefore obtain the LST of the joint sojourn time distribution. This completes the theoretical solution of the problem.

The thesis concludes by showing some numerical results generated by an initial implementation of the model in Mathematica. However, due to numerical instability issues during the inversion of the LST of the joint distribution, it is left for future work to make the implementation complete and stable by testing different inversion methods or using an approximation of the densities by moments of the sojourn time.

Appendix A

Mathematica code

```
(********** LT inversion ***********)

FT[F_, t_, M_:32] := Module[{np, r, S, theta, sigma}, np = Max[M, $MachinePrecision];
  r = SetPrecision[2 M / (5 t), np];
  S = r theta (Cot[theta] + I);
  sigma = theta + (theta Cot[theta] - 1) Cot[theta];
  (r / M) Plus @@ Append[Table[Re[Exp[t S] (1 + I sigma) F[S]], {theta, Pi / M, (M - 1) Pi / M, Pi / M}], (1 / 2) Exp[r t] F[r]]]

(********** Product-form solution ************)

(* Task proceeds from queue 1 to queue 2 and then finishes *)

BGen[rates_, z_] := rates.z ^ Range[Length[rates]];

ProdForm[as_, ds_] := Module[{x, z, na1, na2, nd1, nd2},
  Clear[x, z];
  A0[z_] := {(ρ1 z BGen[as[[1]], 1 / ρ1] - BGen[as[[1]], z]) / (1 - ρ1 z), (ρ2 z BGen[as[[2]], 1 / ρ2] - BGen[as[[2]], z]) / (1 - ρ2 z)} // Simplify;
  a0s = If[Length[#] == 0, {}, Drop[CoefficientList[#, z], 1]] & /@ A0[z];
  {na1, na2} = Length /@ as;
  {nd1, nd2} = Length /@ ds;
  NSolve[{BGen[as[[2]], 1 / ρ2] + BGen[ds[[1]], ρ1 / ρ2] + BGen[ds[[2]], ρ2] - BGen[as[[2]], 1] - BGen[ds[[1]], ρ1] - BGen[ds[[2]], 1]  == 0,
      BGen[as[[1]], 1 / ρ1] + BGen[ds[[1]], ρ1] - BGen[as[[1]], 1] - BGen[ds[[1]], 1] == 0, ρ1 > 0, ρ2 > 0, ρ1 < 1, ρ2 < 1}, {ρ1, ρ2}] // Flatten // N]

(* Equilibrium probability parameter in case of a single queue *)

ProdFormSingle[as_, ds_] := Module[{x, z, na, nd},
  Clear[x, z];
  A0[z_] := (ρ1 z BGen[as, 1 / ρ1] - BGen[as, z]) / (1 - ρ1 z) // Simplify;
  a0s = If[Length[A0[z]] == 0, {}, Drop[CoefficientList[A0[z], z], 1]];
  na = Length[as];
  nd = Length[ds];
  NSolve[{BGen[as, 1 / ρ1] + BGen[ds, ρ1] - BGen[as, 1] - BGen[ds, 1] == 0, ρ1 > 0, ρ1 < 1}, ρ1] // Flatten // N]
```

Figure A.1: Finding the equilibrium probabilities

```
(********** LT inversion ***********)

FT[F_, t_, M_:32] := Module[{np, r, S, theta, sigma}, np = Max[M, $MachinePrecision];
  r = SetPrecision[2 M / (5 t), np];
  S = r theta (Cot[theta] + I);
  sigma = theta + (theta Cot[theta] - 1) Cot[theta];
  (r / M) Plus @@ Append[Table[Re[Exp[t S] (1 + I sigma) F[S]], {theta, Pi / M, (M - 1) Pi / M, Pi / M}], (1 / 2) Exp[r t] F[r]]]

(********** Product-form solution ************)

(* Task proceeds from queue 1 to queue 2 and then finishes *)
BGen[rates_, z_] := rates.z ^ Range[Length[rates]];

ProdForm[as_, ds_] := Module[{x, z, na1, na2, nd1, nd2},
  Clear[x, z];
  A0[z_] := {(ρ1 z BGen[as[[1]], 1 / ρ1] - BGen[as[[1]], z]) / (1 - ρ1 z), (ρ2 z BGen[as[[2]], 1 / ρ2] - BGen[as[[2]], z]) / (1 - ρ2 z)} // Simplify;
  a0s = If[Length[#] == 0, {}, Drop[CoefficientList[#, z], 1]] & /@ A0[z];
  {na1, na2} = Length /@ as;
  {nd1, nd2} = Length /@ ds;
  NSolve[{BGen[as[[2]], 1 / ρ2] + BGen[ds[[1]], ρ1 / ρ2] + BGen[ds[[2]], ρ2] - BGen[as[[2]], 1] - BGen[ds[[1]], ρ1] - BGen[ds[[2]], 1]  == 0,
      BGen[as[[1]], 1 / ρ1] + BGen[ds[[1]], ρ1] - BGen[as[[1]], 1] - BGen[ds[[1]], 1] == 0, ρ1 > 0, ρ2 > 0, ρ1 < 1, ρ2 < 1}, {ρ1, ρ2}] // Flatten // N]
(* Equilibrium probability parameter in case of a single queue *)
ProdFormSingle[as_, ds_] := Module[{x, z, na, nd},
  Clear[x, z];
  A0[z_] := (ρ1 z BGen[as, 1 / ρ1] - BGen[as, z]) / (1 - ρ1 z) // Simplify;
  a0s = If[Length[A0[z]] == 0, {}, Drop[CoefficientList[A0[z], z], 1]];
  na = Length[as];
  nd = Length[ds];
  NSolve[{BGen[as, 1 / ρ1] + BGen[ds, ρ1] - BGen[as, 1] - BGen[ds, 1] == 0, ρ1 > 0, ρ1 < 1}, ρ1] // Flatten // N]
```

Figure A.2: Finding the generating functions


```
(********** LT inversion ***********)

FT[F_, t_, M_:32] := Module[{np, r, S, theta, sigma}, np = Max[M, $MachinePrecision];
  r = SetPrecision[2 M / (5 t), np];
  S = r theta (Cot[theta] + I);
  sigma = theta + (theta Cot[theta] - 1) Cot[theta];
  (r / M) Plus @@ Append[Table[Re[Exp[t S] (1 + I sigma) F[S]], {theta, Pi / M, (M - 1) Pi / M, Pi / M}], (1 / 2) Exp[r t] F[r]]]

(********** Product-form solution ************)

(* Task proceeds from queue 1 to queue 2 and then finishes *)
BGen[rates_, z_] := rates.z ^ Range[Length[rates]];

ProdForm[as_, ds_] := Module[{x, z, na1, na2, nd1, nd2},
  Clear[x, z];
  A0[z_] := {(ρ1 z BGen[as[[1]], 1 / ρ1] - BGen[as[[1]], z]) / (1 - ρ1 z), (ρ2 z BGen[as[[2]], 1 / ρ2] - BGen[as[[2]], z]) / (1 - ρ2 z)} // Simplify;
  a0s = If[Length[#] == 0, {}, Drop[CoefficientList[#, z], 1]] & /@ A0[z];
  {na1, na2} = Length /@ as;
  {nd1, nd2} = Length /@ ds;
  NSolve[{BGen[as[[2]], 1 / ρ2] + BGen[ds[[1]], ρ1 / ρ2] + BGen[ds[[2]], ρ2] - BGen[as[[2]], 1] - BGen[ds[[1]], ρ1] - BGen[ds[[2]], 1]  == 0,
      BGen[as[[1]], 1 / ρ1] + BGen[ds[[1]], ρ1] - BGen[as[[1]], 1] - BGen[ds[[1]], 1] == 0, ρ1 > 0, ρ2 > 0, ρ1 < 1, ρ2 < 1}, {ρ1, ρ2}] // Flatten // N]
(* Equilibrium probability parameter in case of a single queue *)
ProdFormSingle[as_, ds_] := Module[{x, z, na, nd},
  Clear[x, z];
  A0[z_] := (ρ1 z BGen[as, 1 / ρ1] - BGen[as, z]) / (1 - ρ1 z) // Simplify;
  a0s = If[Length[A0[z]] == 0, {}, Drop[CoefficientList[A0[z], z], 1]];
  na = Length[as];
  nd = Length[ds];
  NSolve[{BGen[as, 1 / ρ1] + BGen[ds, ρ1] - BGen[as, 1] - BGen[ds, 1] == 0, ρ1 > 0, ρ1 < 1}, ρ1] // Flatten // N]
```


Figure A.3: Expressing the LSTs with the generating functions and inverting them to get the density functions

Bibliography

[1] Jinho Baik and Raj Rao Nadakuditi. Batch latency analysis and phase transitions for a tandem of queues with exponentially distributed service times. *Queueing Systems*, 78(3):255–285, Nov 2014. pages 1

[2] TS Chis and PG Harrison. Performance-energy trade-offs in smartphones. pages 127–135, 2016. pages 1

[3] Eduardo Gonzalo-Almorox, Alexandru Lucian Gaianu, and Luke Vale. The role of queuing models in economic evaluation: lessons from an illustrative case study of emergency care. 2015. pages 1

[4] Hourvash Akbari Haghighinejad, Erfan Kharazmi, Nahid Hatam, Sedigheh Yousefi, Seyed Ali Hesami, Mina Danaei, and Mehrdad Askarian. Using queuing theory and simulation modelling to reduce waiting times in an iranian emergency department. *International journal of community based nursing and midwifery*, 4(1):11, 2016. pages 1, 10

[5] Peter Harrison. Sojourn time distributions in tandem batch-networks. *SIGMETRICS Perform. Eval. Rev.*, 41(3):37–39, January 2014. pages 2, 24, 34

[6] Peter G. Harrison. Turning back time in markovian process algebra. *Theor. Comput. Sci.*, 290(3):1947–1986, January 2003. pages 15

[7] Peter G. Harrison, Richard A. Hayden, and William J. Knottenbelt. Product-forms in batch networks: Approximation and asymptotics. *Perform. Eval.*, 70(10):822–840, October 2013. pages 2

[8] Peter G. Harrison and Naresh M. Patel. *Performance Modelling of Communication Networks and Computer Architectures (International Computer S)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1992. pages 3

[9] PG Harrison and Z Qiu. Performance enhancement by means of task replication. pages 191–205, 2013. pages 1

[10] W. Henderson and P. G. Taylor. Product form in networks of queues with batch arrivals and batch services. *Queueing Systems*, 6(1):71–87, Dec 1990. pages 16

[11] James R. Jackson. Jobshop-like queueing systems. *Management Science*, 50(12_supplement):1796–1802, 2004. pages 16

[12] Tsuyoshi Katayama and Kaori Kobayashi. Sojourn time analysis of a two-phase queueing system with exhaustive batch-service and its vacation model. *Mathematical and Computer Modelling*, 38(11):1283 – 1291, 2003. Stochastic models in engineering, technology, and management. pages 1

[13] Alexander Klemm, Christoph Lindemann, and Marco Lohmann. *Traffic Modeling of IP Networks Using the Batch Markovian Arrival Process*, pages 92–110. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. pages 41

[14] Peter J. Kolesar, Kenneth L. Rider, Thomas B. Crabill, and Warren E. Walker. A queuing-linear programming approach to scheduling police patrol cars. *Operations Research*, 23(6):1045–1062, 1975. pages 1, 10

[15] A. S. Lebrecht, N. J. Dingle, P. G. Harrison, W. J. Knottenbelt, and S. Zertal. Using bulk arrivals to model i/o request response time distributions in zoned disks and raid systems. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 23:1–23:10, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). pages 10, 41

[16] Mihaela Mitici, Jasper Goseling, Jan-Kees van Ommeren, Maurits de Graaf, and Richard J. Boucherie. On a tandem queue with batch service and its applications in wireless sensor networks. *Queueing Systems*, Jun 2017. pages 41

[17] Athanasios E. Papathanasiou and Michael L. Scott. Energy efficiency through burstiness. In *WMCSA*, 2003. pages 44