Queueing Systems with Heterogeneous Servers: On Fair Routing of Patients in Emergency Departments

Research Thesis

Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Operations Research and Systems Analysis

Yulia Tseytlin

Submitted to the Senate of the
Technion - Israel Institute of Technology

Nissan, 5769 Haifa April, 2009

Acknowledgements

This research thesis was done under the supervision of Professor Avishai Mandelbaum in the Faculty of Industrial Engineering and Management. I would like to express my deepest gratitude for the utmost effort and time he contributed to my research. I feel privileged to have been advised by this distinguished scientist, dedicated teacher and thoughtful person. I am thankful for his invaluable guidance and support throughout every stage of the work, for encouraging and believing in me.

The practical part of my research was done at the Rambam hospital. I am grateful to Kosta Elkin and Noga Rozenberg, whose undergraduate project at the hospital was my first acquaintance with the practical world of my research. I wish to thank all the people at Rambam hospital that I had the honor to work with, and first and foremost Mira Shiloach, nursing manager of the Emergency Department and Internal Wards, for her assistance and information sharing, for her time and help, and for acquainting me with the key people of the hospital. My deepest gratitude goes to the hospital management, especially to Doctor Yaron Bar-El, medical operations director, who opened the doors that had to be opened. I am thankful to Doctor Z. Azam, the head of Internal Ward B, and to the staff of the Emergency Department, for making my visits to the hospital exciting, for allowing me to interview them and for sharing their knowledge and thoughts. Special thanks go to Yossi Levy for his help in data provision. I am also extremely grateful to my contact people in five other hospitals: Doctor Haron in Naharia, Doctor Halpern in Tel-Aviv, Professor Stern in Hadassah - Har Hatzofim, Doctor Ashkar in Hilel-Yafe and Shoshi Levavi in Carmel, for kindly cooperating and providing me with information on routing processes in their institutions.

I would like to thank Professor Haya Kaspi whose help in theoretical analysis is greatly appreciated. I am also thankful to Professor Mor Armony and to Professor Petar Momcilovic for many helpful discussions. During my graduate studies, I had the honor and pleasure to be involved in two research projects: I am truly grateful to my project partners Yariv Marmor, Galit Yom-Tov and Asaf Zviran - I learnt from them a lot. I am also extremely thankful to other fellow graduate students who helped me, especially to Polina Khudyakov and Shimrit Maman. I wish to thank Professor Michal Penn for valuable guidance during the project in the course "System Analysis and Design".

Special thanks go to my boyfriend and my family for supporting and encouraging me.

Finally, the generous financial help of the Technion Graduate School is gratefully acknowledged. I am extremely thankful to the British Technion Society Graduate Fellowship, the Forchheimer Foundation Fellowship and the Jarndyce Foundation Fellowship. I am also grateful to to joint Rambam-IBM-IE&M OCR project for being a generous source of financial help and providing a nurturing environment for doing research. I wish to thank the Israel National Institute for Health Services and Health Policy Research for the generous financial support.

Contents

List of Tables

List of Figures

Abstract

We consider the process of patients' routing from an Emergency Department (ED) to Internal Wards (IWs) in Anonymous Hospital. We recognize two main problems in the process: patients' waiting times in the ED for a transfer to the IWs are long, and patients' allocation to the wards does not appear to be fair. This motivates us to model the "ED-to-IW process" as a queueing system with heterogeneous server pools, where the pools represent the wards and servers are beds. We analyze this system under various queue-architectures and routing policies, in search for fairness and good operational performance.

Our queueing system, with a single centralized queue and several server pools, forms an Inverted-V model. We introduce the Randomized Most-Idle routing policy (RMI): each arriving customer joins one of the available pools, with a probability that equals the proportion of idle servers in this pool out of the overall number of idle servers in the system. The Inverted-V system under RMI gives rise to a reversible Markov process which can be analyzed, in steady-state, in closed form. Its analysis yields interesting operational insights, for example: faster pools have lower average occupancy-rate than slower pools, but higher "flux" (average number of customers served by a server in this pool, per time unit) than slower pools.

We supplement the closed-form analysis by an asymptotic analysis in the QED (Quality and Efficiency Driven) regime, where efficiency is carefully balanced against service quality. The asymptotic analysis provides simplified expressions for some operational measures, e.g., the probability of delay in the system; it also yields insights on how idleness is shared among the pools, which adds more favorable "points" to the fairness of RMI routing. A disadvantage of RMI is its being randomized, which motivates us to study alternatives: a class of non-randomized policies - WMI (Weighted Most-Idle). We compare those policies in the QED regime, according to various fairness and performance criteria; our analysis is tested against computer simulations.

Another queue-architecture considered is Distributed Finite Queues (DFQ). We propose the RMI equivalent for such a system, which also gives rise to a reversible Markov process and is analyzed in steady-state. In order to achieve additional practical insights, by accommodating some analytically intractable features and testing various routing policies, we model the ED-to-IW process by a computer simulation. With the simulation, we compare various routing policies, according to some fairness and performance criteria, while also accounting for the availability of information in the system. Finally, we comment on the ED-to-IW process from a game-theoretic view point.

List of Symbols

λ	the arrival rate
K	the number of server pools
μ_i	the service rate of a server in pool i
N_i	the number of servers in pool i
N	the total number of servers in the system: $N = \sum_{i=1}^{K} N_i$
$\mathcal{I}_i(t)$	the number of idle servers in pool i at time t
$B_i(t)$	the number of busy servers in pool i at time t
$\mathcal{I}_i$	the stationary number of idle servers in pool i
B_i	the stationary number of busy servers in pool i
$\tilde{\rho}_i$	the stationary occupancy rate in pool i: $\tilde{\rho}_i = \dfrac{B_i}{N_i}$
$\bar{\rho}_i$	the mean stationary occupancy rate in pool i (servers' utilization in pool i)
γ_i	the average *flux* through pool i (average number of arrivals per pool i server per time unit): $\gamma_i = \mu_i \bar{\rho}_i$
X^+	$= max\{X, 0\}$
E	expectation
P	probability
W_q	the stationary waiting time in the centralized queue
L_q	the stationary centralized queue-length
ρ	the total traffic intensity
$\stackrel{d}{=}$	distributed as (for example, $X \stackrel{d}{=} Pois(\lambda)$ means that X is a random variable that is Poisson distributed with parameter λ).
$\Phi(\cdot)$, $\varphi(\cdot)$	the standard normal distribution and density functions

The Quality and Efficiency Driven (QED) Regime:

$\approx$	$a_n \approx b_n$ if $a_n/b_n \to 1$, as $n \to \infty$
o	$a_n = o(b_n)$ if $a_n/b_n \to 0$, as $n \to \infty$
a_i	the limiting service capacity proportion of pool i
q_i	the limiting fraction of servers in pool i out of the total number of servers
μ	the mean service rate (harmonic): $\mu = \left(\frac{a_1}{\mu_1} + \frac{a_2}{\mu_2}\right)^{-1}$
$\hat{\mu}$	the mean service rate (arithmetic): $\hat{\mu} = \mu_1 a_1 + \mu_2 a_2$
δ	the square root safety capacity coefficient
β	the Quality-of-Service (QoS) parameter: $\beta = \frac{\delta}{\sqrt{\mu}}$

The Distributed Finite Queues (DFQ) System:

b_i	the queue (buffer) capacity of pool i
b	the total queue capacity: $b = \sum_{i=1}^{K} b_i$
C_i	the total capacity of pool i: $C_i = N_i + b_i$
C	the total system capacity: $C = \sum_{i=1}^{K} C_i = N + b$
$\mathcal{E}_i(t)$	the number of empty places in the buffer of pool i at time t
$V_i(t)$	the total number of vacant places in pool i at time t: $V_i(t) = \mathcal{I}_i(t) + \mathcal{E}_i(t)$
W_p	the stationary waiting time in the distributed queues
W_t	the stationary total waiting time in the system
$a \wedge b$	$min\{a, b\}$

Game Theory:

$(\mathbb{K}, V)$	the transferable utility cooperative game
$\mathbb{K}$	the set of players: $\mathbb{K} = \{1, 2, \ldots, K\}$
$V(\cdot)$	the characteristic function of the game
$\mathbf{S}$	coalition (a group of cooperating players): $\emptyset \subset \mathbf{S} \subseteq \mathbb{K}$ ($\mathbb{K}$ is the grand coalition)
$\varphi_i(V)$	the Shapley value of player i in the game $(\mathbb{K}, V)$
λ_i	the arrival rate to pool i
$\lambda_{\mathbf{S}}$	the arrival rate to the system, formed by coalition $\mathbf{S}$: $\lambda_{\mathbf{S}} = \sum_{i \in \mathbf{S}} \lambda_i$
$N_{\mathbf{S}}$	the total number of servers in the system, formed by coalition $\mathbf{S}$: $N_{\mathbf{S}} = \sum_{i \in \mathbf{S}} N_i$
$E(L_q^{\mathbf{S}})$	the mean total queue length in the system, formed by coalition $\mathbf{S}$

List of Acronyms

ALOS	Average Length of Stay
a.s.	almost sure
B&D	Birth and Death
DFQ	Distributed Finite Queues
ED	Emergency Department
FCFS	First Come First Served
FSF	Fastest Servers First
i.i.d.	independent and identically-distributed
IW	Internal Ward
LIPF	Longest Idle Pool First
LISF	Longest-Idle Server First
LOS	Length of Stay
LWISF	Longest-Weighted-Idle Server First
MI	Most-Idle
MSP	Multiple-Server-Pools
NERMI	Non-random Equivalent to RMI
OB	Occupancy Balancing
PASTA	Poisson Arrivals See Time Averages
QED	Quality and Efficiency Driven
QIR	Queue-and-Idleness-Ratio
QoS	Quality-of-Service

RA Random Assignment

RMI Randomized Most-Idle

SSC State-Space Collapse

SSF Slow Server First

SSP Single-Server-Pools

WMI Weighted Most-Idle

Chapter 1

Introduction

During the last century and the beginning of the present one, the service sector has grown significantly and now accounts for over 70% of the economy in the United States, and similarly in many other Western countries. The service sector covers a wide spectrum of activities, e.g. professional, financial and government services. In this research we focus on a very important part of the service sector - the health care system, and in particular on *hospitals*.

A hospital is an institution for health care, which is able to provide long term patient stays. Green [23] states that hospitals are increasingly aware of the need to use their resources as efficiently as possible, in order to continue to assure their institutions' survival and prosperity. Over the years, hospitals have been successful in using medical and technical innovations to deliver more effective clinical treatments, while reducing patients' time spent in the hospital. However, hospitals are typically rife with inefficiencies and delays, thus present a propitious ground for many research projects in numerous science fields, and in the Operations Research field in particular.

Hospitals include numerous medical units specializing in different areas of medicine, for example, internal, surgery, intensive care, obstetrics, and so forth. In most large hospitals, there are several similar medical units operating in parallel. In our research, we focus on the **Emergency Department (ED)** and its interface with four **Internal Wards (IWs)** in a large Israeli hospital - we refer to it as **Anonymous Hospital**. The ED caters immediate threats to health and provides emergency medical services. Thus the proper functioning of the ED is of utter importance, and its overcrowding can cause an inability to admit new patients and ambulances diversions (see [15] for consequences of ED congestion).

A patient arriving to the ED undergoes registration, diagnostic testing, basic treatment and then is either dismissed or admitted to stay, the latter if doctors decide on hospitalization, in which case the patient is transferred to the appropriate medical unit. We focus on admitted internal patients, specifically on the process from the decision of hospitalization till admission to the IW. Two main problems could arise in the process: patients' waiting times in the ED for a transfer to the IWs could be **long**, and patients' allocation to the wards need not be **fair**.

The main goal of our research is to investigate the "ED-to-IW process", practically, theoretically and with the help of simulations. Practically, we study the process in Anonymous Hospital in great details, comparing it to other hospitals as well. Theoretically, we model the ED-to-IW process as a queueing system with heterogeneous server pools: the pools represent the wards and servers are beds. We analyze this system under various queue-architectures and routing policies, in search for fairness and good operational performance.

In order to achieve additional practical insights, by accommodating some analytically intractable features and testing various routing policies, we create a computer simulation model of the ED-to-IW process - in a joint project with A. Zviran. The project report may be found in [52]; we provide its summary in Chapter 8. Many interesting empirical issues are analyzed in a joint project with Y. Marmor and G. Yom-Tov [35]. We refer the reader to this project, during specific discussions throughout our work, for more details on the topic under consideration.

1.1 The Structure of the Thesis

This work is structured as follows: first, in Chapter 2, we provide background on the practical side of our research, namely on patients flow from the ED to IWs in hospitals. In Chapter 3 we provide background on the theoretical side of our research: we survey the literature on relevant queueing models, game theory and fairness. Next, in Chapters 4 and 5, we analyze a specific type of a queueing model - an *inverted-V model*, first in steady-state and then asymptotically. In Chapter 6 we address another queueing system: a *distributed finite-queues model*. In Chapter 7 we examine the ED-to-IW process from a game-theoretic point of view, and in Chapter 8 the simulation study of the ED-to-IW process is presented. Finally, we discuss our research limitations and propose ideas for further research in Chapter 9.

Chapter 2

Practical Background: the ED-to-IW Process

The practical side of our research studies the patients flow from the ED (Emergency Department) to the IWs (Internal Wards) of Anonymous Hospital. In this chapter, we provide background on the hospital and the medical units in question (Section 2.1), describe the process of patients' routing (Section 2.2) and the problems involved (Section 2.3). Then we describe how the "ED-to-IW process" is managed in five other Israeli hospitals (Section 2.4) and survey the relevant literature (Section 2.5).

We would like to acknowledge that a significant part of the background on the ED-to-IW process in Anonymous Hospital has been provided to us by K. Elkin and N. Rozenberg [18], who did an IE undergraduate project at this hospital in 2006-2007.

2.1 Hospital, ED and IWs

Anonymous Hospital is a large Israeli hospital with about 1000 beds, 45 medical units, and about 75,000 patients hospitalized yearly. Among its variety of medical sections, it has a large ED with an average arrival rate of 240 patients daily and a capacity of 40 beds; and five IWs which we denote from A to E. The ED is divided into two major subunits: Internal and Trauma (surgical and orthopedic patients); each of those is divided into "walking" and "lying" subunits, according to the state of patients treated there. An internal patient, whom the ED decides to hospitalize, is directed to one of the five Internal Wards according to a certain routing policy - and this routing process is precisely the focus of our research.

Internal Medicine Departments are responsible for the treatment of a wide range of internal disorders, providing inpatient medical care to thousands of patients each year. Wards A-D are more or less similar in their medical capabilities - each can treat multiple types of patients. Ward E, on the other hand, treats only "walking" patients, and the routing process from the ED to it differs from the one to the other wards (see Section 2.2). In our research we concentrate on the routing process to wards A-D only.

Although wards A-D provide similar medical services, they do differ in their *operational measures*. Below we elaborate on what we mean by operational measures. First of all, each medical unit is characterized by its *capacity*. Ward's capacity is measured by its number of beds (*static capacity*) and number of service providers - doctors, nurses, administrative staff and general workers (*dynamic capacity*). Generally (and in our hospital IWs in particular), the latter is determined proportionally to the former (see, however, discussions on the appropriateness of such "proportional" staffing in [29], [24] and [56]). As an example, in Anonymous hospital, an IW nurses-to-beds ratio in morning shifts is one to five-six. Hence a unit's operational capacity can be characterized by the number of its beds only - denoted as its *standard capacity*. *Maximal capacity* stands for standard capacity plus extra beds, that can be placed in corridors in overloaded periods.

In addition, medical units can be characterized by various performance measures: *operational* - average bed occupancy level, Average Length of Stay (ALOS), waiting times for various resources, number of patients hospitalized per bed per time unit (we call it *flux*); and *quality* - patients' return rate, patients' satisfaction, mortality rate, etc. Note that occupancy rate and flux are calculated relatively to wards standard capacities. Comparing two basic measures, ward capacity and ALOS, we observe in Table 2.1 that the wards differ on both. We note that Ward B is the smallest and the "fastest" (shortest ALOS) among wards A-D.

Table 2.1: Internal Wards Capacities and ALOS

	Ward A	Ward B	Ward C	Ward D	Ward E
Standard capacity (# beds)	45	30	44	42	24
Maximal capacity (# beds)	52	35	46	44	27
ALOS (days)	6.368	4.474	5.358	5.562	4.11

2.2 The Routing Process

The decision of routing a patient to-be-hospitalized to one of wards A-D is made on the basis of a computer program, referred to at the hospital as the "*Justice Table*". The program accepts a patient's category (described below) as an input parameter and returns a ward for the patient (A-D) as an output. As its name suggests, the algorithm's goal is doing justice with the wards: making the patients' allocation to the wards *fair*. Below we describe briefly the history of the Justice Table from its inception to present days.

2.2.1 The History of the Justice Table

Before 1997: Patients' allocation was decided according to a fixed "table of duty" of the wards (every day another ward was on duty and had to accommodate all incoming patients), but allocation was subject to wards' approval - each ward had the authority to refuse to admit a patient. Consequently, waiting times in the ED until transfer to the IWs were extremely long - 10.5 hours on average, with 12% of the patients forced to wait more than 24 hours (!) [41]. This was unbearable to patients, and caused a heavy overload on the ED and hence the department's malfunctioning. In 1995, as part of a hospital quality program, a dedicated team for improving processes in the ED was founded - its goal was, in particular, to reduce the ED-IW delays. The team [41] proposed a change in the existing routing policy: a patient's placement would be determined by an algorithm named the "Justice Table", and the authority for patients' routing would be taken away from the wards. Implementation of this change-of-authority was not easy but eventually successful.

Short description of the 'Justice Table" algorithm: The purpose of the "Justice Table" was to balance the load among the wards. It was decided to classify patients into three categories: *ventilated* - patients that required artificial respiration, *special care* - patients whose rate on the Norton scale (a table used to predict if a patient might develop a pressure ulceration) was below 14, and *regular* - all other patients. Length Of Stay (LOS) and complexity of treatment varied significantly among those categories, which is why the algorithm directed each category independently in order to ensure fair allocation. For each patients' category, there was *cyclical order* among the wards, namely each ward received one patient in its turn (Round-Robin). In addition to a patient's classification, the algorithm took into account the size of each ward, by allocating less patients to a smaller ward. It took into account neither the actual number of occupied beds at the time the routing decision was made, nor the discharge rate at the wards.

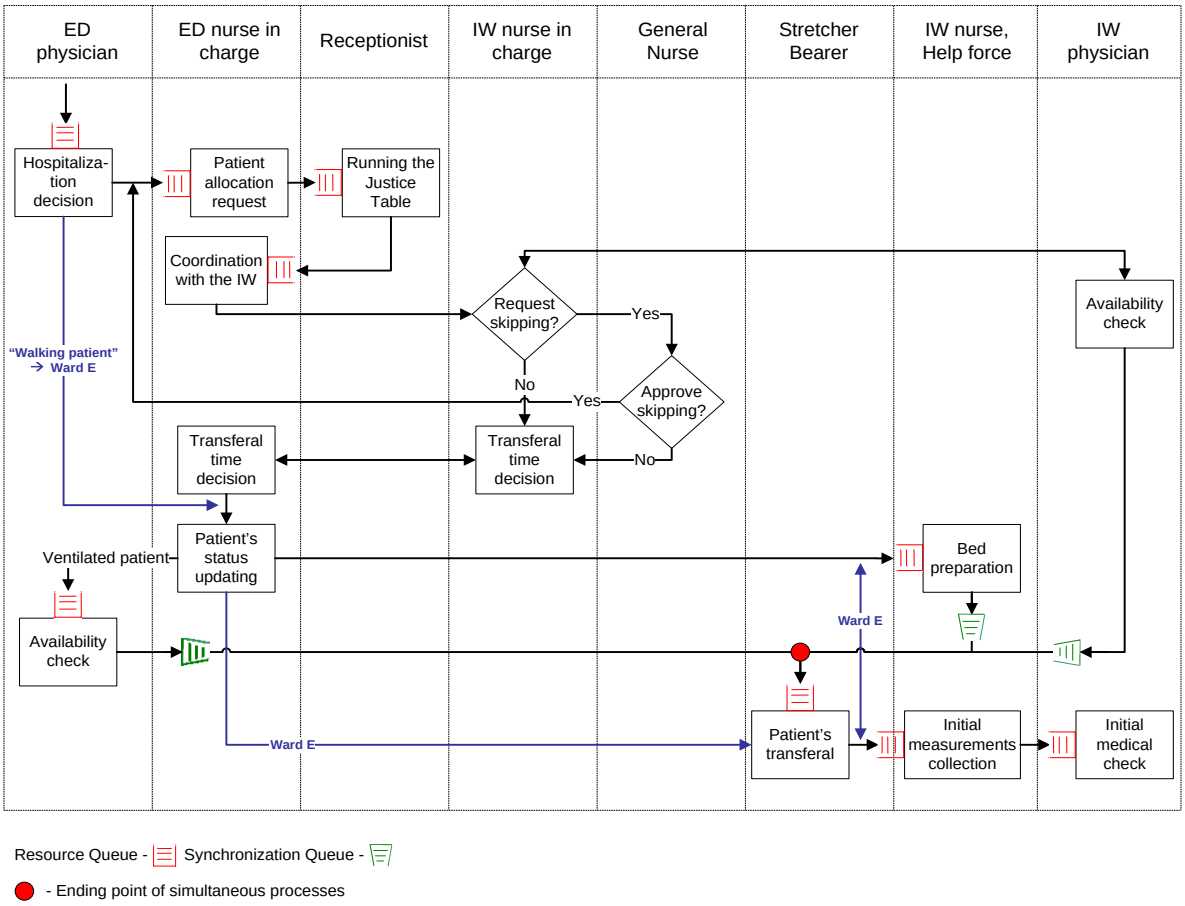
The current situation: The immediate results of implementing the new routing policy were very impressive - average waiting time from decision on hospitalization till moving the patient to a ward was reduced to 66 minutes (from over ten hours) [41]. In addition, significant improvements in other ED processes were measured as well (due to the overload reduction), along with a higher ward efficiency (more admissions, shorter LOS). But in 2004, the use of the Justice Table was discontinued, due to software changes in the hospital. In 2006, adapted to the new software, the Justice Table was reinstated with minor changes, but its influence grew smaller, as the medical staff had become used to making placement decisions without it. Thus, in the observations that [18] conducted, it turned out that a significant number of the patients transferred from the ED to the IWs were, in fact, not routed via the Justice Table (see also the concluding discussion of Section 2.3.2).

2.2.2 Patients' Routing

One can fully appreciate the complexity of the ED-to-IW process in the Integrated (Activities - Resources) Flow Chart (Figure 2.1) and other flow charts (see Appendix A). We provide here a short description: A patient, whom a physician in charge of the ED decides to hospitalize in the IWs, is assigned to one of the five wards in the following manner: If this is an "independent" walking patient, usually s/he is assigned to Ward E - in this case usually s/he is transferred to the ward almost without delay. Otherwise, a receptionist of the ED runs the Justice Table. She transfers the output (one of the wards A-D) to the nurse in charge of the ED, who starts a negotiation process with the chosen ward. If the ward refuses to admit the patient (usually for reasons of overloading), the two sides appeal to a General Nurse, who is authorized to approve the so-called "skipping" - allowing a ward to skip its turn. If skipping is granted, the receptionist runs the Justice Table again, and the process repeats itself until some ward agrees (or is forced) to admit the patient.

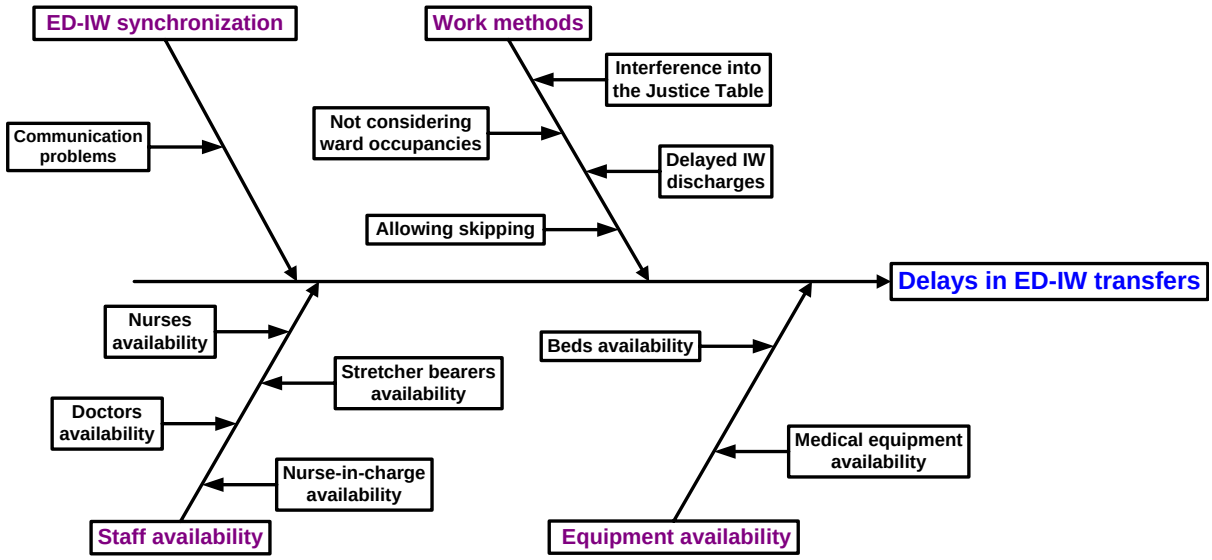
The next stage of the negotiations is agreeing upon the time at which the patient will be transferred to his ward. Here interests are conflicting: the ED seeks to discharge the patient as soon as possible in order to be able to accept new ones, and the IWs wish to have the move carried out at a time convenient for them. From conversations with nurses from both sides we learn that, when deciding on a patient's transfer time, the main issue taken into account (assuming there is an available bed in the ward) is nurses' and doctors' availability (they might be unavailable because of treating other patients, shifts changing or meals, various staff meetings or resuscitation). Another parameter is the availability

Figure 2.1: Integrated (Activities - Resources) Flow Chart



of necessary equipment and other logistic considerations: for example, preparation for a "complicated" patient who requires special bed/equipment, or placement near a nursing station, takes a longer time. Patients to-be hospitalized wait in the ED till transfer to their ward is carried out - sometimes these waiting times are extremely long (see the following Section 2.3.1). Through the Causes and Effects Chart (Fish-bone Diagram) in Figure 2.2 one sees the various causes of these long delays. We wish to emphasize that the delays are caused not only by beds unavailability: patients usually wait even when there are available beds (see Remark 6.4.1).

Figure 2.2: ED-IW Delays: Causes and Effects Chart



2.3 Problems in the ED-to-IW Process

We recognize two main problems in the process of patients routing from the ID to the IWs: waiting times in the ED for a transfer to the IWs could be **long**, and patients' allocation to the wards does not appear to be **fair**. In this section we describe and analyze each one of these problems.

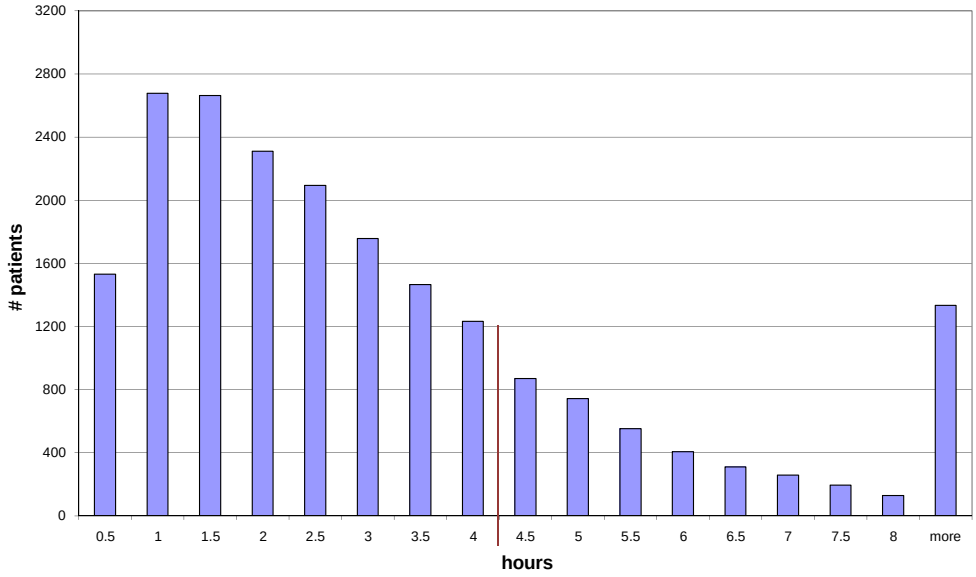
2.3.1 Long Delays

For the reasons described above (Section 2.2.2) and depicted in Figure 2.2, patients often wait a long time in the ED until they are transferred to their IW. There exists an agreement obliging the wards to admit patients within *four hours* from the decision of hospitalization, but in certain cases it takes even longer. Exact data on those waiting times are not kept in the hospital information systems, thus it is hard to tell exactly how long the patients wait. We now present two reasonably reliable estimations of these waiting times.

From 182 observations that [18] conducted in May 2007, one sees that the average waiting time from a decision of a patient's hospitalization till admission in the IWs was 97 minutes on average. The longest average waiting time was for admission in Ward B (112 minutes), waiting times for admission in the other wards were more or less similar (around

90 minutes). We note that 45% of the patients observed waited two hours or longer and 7% - four hours or longer. However, during our visit in the ED, we got the impression (from a conversation with the nurse in charge) that waiting times are longer. Indeed, when analyzing the time that passed from a decision of hospitalization in a certain ward till receiving the first treatment in this ward (these data were acquired from the hospital database), we observe far longer waits. For example, the average delay in 2006-2008 was 3.1 hours (for Wards A-D), and for 23% of the patients this time was longer than four hours. The waiting times histogram is depicted in Figure 2.3.

Figure 2.3: Waiting Times Histogram



* Data refer to period 1/05/06 - 30/10/08 (excluding the months 1-3/07 when Ward B was in charge of an additional sub-ward)

Long waiting times cause an overload on the ED, as beds remain occupied while new patients continue to arrive. They cause significant discomfort to the waiting patients as well: in the ED they suffer from noise and lack of privacy and hot meals. In addition, they do not enjoy the best professional medical treatment and dedicated attention as in the wards; hence the longer the patients wait in the ED, the lower their satisfaction and the higher the likelihood for clinical deterioration. Improving the efficiency of patients flow from the ED to the IWs, while shortening waiting times in the ED, will improve the service and treatment provided to patients. In addition, reducing the load on the ED will lead to a better response to arriving patients and it is likely to save lives.

2.3.2 Fairness

Before arguing that patients' allocation to the IWs does not appear to be fair, we should first understand what is meant by "fairness" (for a literature survey on the notion of "fairness", see Section 3.2). Possible criteria for fair patient allocation, from the points of view of *wards* - meaning medical and nursing staff, and *patients*, are abundant. We consider staff-fairness first. Our anchor point is the results of a survey conducted by [18], in which the staff (nurses, doctors and administration) were asked to grade the extent of fairness in different routing policies.

Staff - Fairness

When discussing fairness with wards staff, the consensus is that each nurse/doctor should have the same workload. Seemingly, this is the same as saying that each nurse/doctor should take care, at any given time, of an equal number of patients. As the number of nurses and doctors is usually proportional to standard capacity, this criterion is equivalent to keeping beds *occupancy rates* equal among the wards. However, if one maintains occupancy levels equal then, by Little's law, wards with shorter ALOS will have a higher turnover rate - admit more patients per bed. But the load on the ward staff is not spread uniformly over a patient's stay, as treatment during the first days of hospitalization requires much more time and effort from the staff than in following days [18]; in addition, patients' admissions and discharges consume doctors' and nurses' time and effort as well. Thus, even if the occupancy among wards is kept equal, the ward admitting more patients per bed ends up having a larger load on its staff. Hence, a natural alternative fairness criterion is balancing the incoming load, or *flux* - namely, the number of admitted patients per bed per certain time unit (for example, per month), among the wards.

Let us examine the proposed fairness criteria in our process, with the help of Table 2.2. Consider Ward B which, as we have seen already in Table 2.1, is both the smallest and the "fastest" (in the sense of short ALOS) out of the four wards. We observe that the average occupancy rate in this ward is high. In addition, the number of patients hospitalized per month in this ward equals about 90% of the number of patients hospitalized per month in the other wards, although its size is just about 2/3 of the others. And indeed, the flux in Ward B (6.25 patients per bed per month) is significantly higher than in the other wards, hence (from the discussion above) the load on its staff is the highest. Short LOS could result from a superior efficient clinical treatment; but it might alternatively be a consequence of too-early discharge of patients, which is clearly undesirable. One possible

Table 2.2: Internal Wards Operational Measures

	Ward A	Ward B	Ward C	Ward D
ALOS (days)	6.368	4.474	5.358	5.562
Mean Occupancy Rate	97.8%	94.4%	86.8%	91.1%
Mean # Patients per Month	205.5	187.6	210.0	209.6
Standard capacity	45	30	44	42
Mean # Patients per Bed per Month	4.57	6.25	4.77	4.77
Return Rate (within 3 months)	16.4%	17.4%	19.2%	17.6%

* Data refer to period 1/05/06 - 30/10/08 (excluding the months 1-3/07)

(and accessible) quality measure of clinical care is patients' *return rate* (proportion of patients who are re-hospitalized in the IWs within a certain period of time - in our case, three months). In Table 2.2 we see that the return rate in Ward B does not differ much from the other wards. Also, patients' satisfaction level in surveys - another measure of treatment's quality - does not differ in Ward B from the other wards (based on [18]). We conclude that the most efficient ward, instead of being rewarded, is exposed to the highest load; hence, the patients' allocation appears unfair, as far as the wards are concerned.

Fairness is further diminished when we examine admissions separately for each patients' category. As mentioned earlier, prior to routing, patients are classified into three categories: ventilated, special-care and regular. In Table 2.3 we see patients' allocations from different categories during the period 01/05/2006 - 01/09/2008 (again, excluding the months 1-3/2007 when Ward B was in charge of an additional sub-ward). The fraction of ventilated and special-care patients, allocated to Ward B (out of the total number of patients allocated to Ward B) is significantly higher than to other wards. Load inflicted on the ward's staff by such patients is higher than by regular patients. Besides, their LOS are generally longer - and the fact that ALOS in Ward B are the shortest is now even more impressive.

As a last observation, we mention that the exposed unfairness is caused not so much by the algorithm of the Justice Table but, rather, by interferences into the allocations it produces, as we see when analyzing the empirical data. Indeed, for 13.2% of the patients that were routed via the Justice Table during the period 01/05/2006 - 01/09/2008, the ward chosen originally by the program was actually changed. One of the reasons for those changes were wards' overload. We see that among the patients whose allocation was changed, Ward B had the least number of changes (445 changes versus 600, 696 and 972 for wards A, C and D respectively). Hence, one of the reasons that Ward B admits

Table 2.3: Justice Table Allocations by Patient Categories

IW\Type	Regular	Special-care	Ventilated	Total
Ward A	2,316 (50.3%)	2,206 (47.9%)	83 (1.8%)	4,605 (25.2%)
Ward B	1,676 (43.0%)	2,135 (54.7%)	90 (2.3%)	3,901 (21.4%)
Ward C	2,310 (49.9%)	2,232 (48.2%)	88 (1.9%)	4,630 (25.4%)
Ward D	2,737 (53.5%)	2,291 (44.8%)	89 (1.7%)	5,117 (28.0%)
Total	9,039 (49.5%)	8,864 (48.6%)	350 (1.9%)	18,253

* Data refer to period 1/05/06 - 1/09/08 (excluding the months 1-3/07)

relatively more patients, is that it exceeds its capacity less frequently than the other wards, plausibly, due to its efficiency (which is consistent with what we learnt through an interview with the head of Ward B). Furthermore, 19.2% of the patients (over the same time period) were eventually admitted to a ward, different from the one recorded in the Justice Table (*after* the changes described above). The number of patients eventually admitted to Ward B that were originally assigned to one of the wards A,C,D is the largest, and at the same time the proportion of patients originally assigned to Ward B but eventually admitted to other wards is the smallest. Readers are referred to [35] for additional empirical data and further discussion on this issue.

Patients - Fairness

Fairness criteria from the patients' point of view is concerned, first and foremost, with waiting time from a hospitalization decision till ward admission: one expects it to not vary by much among patients. Ensuring that each nurse/doctor takes care of an equal number of patients is important here as well: each patient should enjoy equal quality of care. An additional notable aspect: in our literature survey in Section 3.2, we shall see that the FCFS (First-Come-First-Served) queueing discipline is essential for customer justice perception while waiting. Consequently, a customer feels s/he is treated more fairly when s/he waits in a single queue than in multi-queues ([33], [42]). But what happens in our ED-to-IW process? Suppose that a decision of hospitalization of patient X was made prior to a decision of hospitalization of patient Y (assuming both of them are of the same type): say, patient X is directed to Ward A and patient Y to Ward B. Now, suppose that Ward B becomes ready to admit the patient faster than Ward A - hence patient Y starts receiving service *before* patient X, although s/he "arrived" later [18] - the FCFS principle is violated. It follows that there are practically four separate queues - one for each of the

wards A-D: a multi-queues system (schematically depicted in Figure 3.3).

We identified two reasons (at least) for this way of routing (not admitting patient X to Ward B that happened to become ready before Ward A). The first reason is that the ward informed about admitting a patient starts preparing for this *specific* patient: different patients, even if they fall under the same classification, might require different preparations. The second reason is psychological: a patient awaiting hospitalization (as well as family members) experiences high levels of stress as is - one thus does not wish to add distress by changing the ward to which s/he was originally assigned [18]. We return to the issue of multi-queues systems vs. single-queue systems in the theoretical part of the thesis, in particular in Chapter 6.

We see that the patients routing does not appear to be fair either towards wards' staff, or towards patients. Increasing fairness in the routing process will increase staff satisfaction, provide incentives for improved care and cooperation (see Section 3.2 for the importance of service providers' equity perception). This will also improve patients' satisfaction, in particular their perception of the quality of care.

2.4 Other Hospitals

We have addressed patients flow from the ED to the IWs in one specific hospital and exposed significant problems in the process. We were eager to learn how this process is managed in other hospitals and, in particular, what routing policies are being used, and how successful they are in terms of delays and fair allocation. We turned to a number of Israeli hospitals with a questionnaire (see Appendix B), which included both qualitative (detailed description of the process, fairness considerations) and quantitative (operational measures of an ED and IWs: capacity, ALOS, waiting times) questions. Five hospitals (denoted from 1 to 5) answered our questions, and we are truly grateful to them. A summary of their responses is presented in Table 2.4.

The hospitals differ in their functionality and geographical location. In Table 2.4 we also see that they differ in *size* (number of IWs and beds in them; number of treated patients), in the *load* IWs are subjected to (average number of transfers to IWs per bed; average occupancy rate - as reported by [55]) and by their *efficiency* (ALOS in the ED and in the IWs). Despite these differences, we recognize the same problems in the ED-to-IW process in all hospitals. First of all, none of the hospitals (except for Hospital 5) measures waiting times of patients to-be hospitalized in the IWs - we were given just a

Table 2.4: Hospitals Comparison

	Hospital 1	Hospital 2	Hospital 3	Hospital 4	Hospital 5	Anon. H.
Number of IWs	9	2	3	4	6	5
IW # beds	327	45	108	93	210	185
Average weekly # of arrivals to Internal ED	1050	350	637	630	1050	1050
Average weekly # of transfers from ED to IWs	525 (50%)	49 (14%)	266 (42%)	168 (26%)	469 (45%)	231 (22%)
Average weekly # of transfers per IW bed	1.606	1.089	2.463	1.806	2.233	1.249
IW Occupancy*	107.5%	118%	106.5%	116.4%	110%	93.8%
ED ALOS (hours)	2.2	6	2.83	6.8	2.5	4.2
IW ALOS (days)	3.9	3.9	3.5	6.1	3.5	5.2
Average waiting time in ED for IW (hours)	?	4	1	8	0.5	1.5-3
Wards differ**?	yes	yes	no	yes	no	yes
Routing Policy	cyclical order	last digit of id	cyclical order	vacant bed	cyclical order***	cyclical order***

* Based on internet article [55].

** Differ in their capacities and LOS.

*** Accounting for different patient types and ward capacities.

rough estimate. Those estimated waiting times are long (again, except for Hospital 5) - indeed, the situation in Anonymous hospital is not the worst.

The routing policies are intuitive and simple - "cyclical order" policies prevail. Although in four out of the six hospitals the wards are heterogeneous - differ in their capacities and ALOS, those differences play no role in routing: no hospital accounts for differences in ALOS and only Anonymous Hospital accounts for ward capacities. In addition, none of the hospitals, besides our hospital and Hospital 5, allocates patients of different categories separately. Surely this cannot be fair, as load inflicted on the ward

staff by patients of different categories varies significantly.

Hospital 2 has quite an original policy: routing is performed according to the one-before-last digit of a patient's identity number: if it is odd then the patient is assigned to Ward A, if even - to Ward B. This is equivalent to random assignment: each ward is chosen with probability 1/2. But ward capacities differ: the size of one ward is 2/3 of the other's, hence this method cannot be fair. Hospital 4 has an even "simpler" policy: sending to a ward that has a vacant bed (we were told that the wards were *always* full). Indeed, the load on the IWs in this hospital (average occupancy and flux) is very high. Due to such a policy, the wards decide when to admit their patients and they do not have any incentive to become more efficient and discharge patients faster - waiting times, ALOS in the ED and in the IWs are the longest in this hospital.

Hospital 5 presents an exception - patients to be hospitalized in the IWs are transferred from the ED almost without any delay, there are separate cycles for different patient categories and even the policy of cyclical routing appears fair as all the wards are the same (in terms of equal capacities and LOS). However, from a conversation with the ED receptionist during our visit to this hospital, we learnt that the routing process was managed by her manually and she received frequent complaints from the wards' staff on unfairness of the allocations.

Note: the discussion presented here is necessarily superficial, as it is based on questionnaires and interviews (if there were such) solely - no observations or data collection were performed in those hospitals. We are not acquainted with any evidences on how the ED-to-IW process is managed in hospitals abroad.

2.5 Literature Review

There is a vast amount of research literature on health care systems in numerous scientific fields, and in the Operations Research field in particular. We present here just some of it (the most relevant): additional references may be found in each of the papers mentioned below. Green ([23], [24]) describes the general background and issues involved in hospital capacity planning, and shows examples of how Operations Research methodologies can be used to provide important insights into operational strategies and practice. Some other examples can be found in [10], [16], [17], [22], [29], [38].

Both research papers ([15]) and newspaper articles ([40]) recognize the importance of ED proper functioning and present the consequences of its overcrowding. Patients' flow

from the ED to other medical units in hospital (not just the IWs) has received attention as well (it even became a subject for popular novel [53]). Ramakrishnan et al. [43] construct a two-time-scale model for a hospital system, where the wards operate on a time-scale of days and are modeled by a discrete time Markov chain, and the ED operates on a much faster time-scale and is modeled by a continuous time Markov chain. With the help of this model, [43] estimates expected occupancy of the wards and the probability of each ward to reach its capacity.

Simulation models are very powerful instruments of the Operations Research field in: (a) validating the correctness of theoretical models, (b) testing algorithms or policies that are too complicated to be solved analytically. Simulations can handle almost any model complexity and take into account the very small details. An additional possible advantage of medical systems simulations is that, due to their graphical interface, they are more likely to be understandable and useful to medical staff (and not only to OR researchers). An interesting example of ED simulations is Sinreich and Marmor [37], who created a generic simulation model of an ED. We present a simulation analysis of the ED-to-IW process in Chapter 8.

Chapter 3

Theoretical Background and Literature Review

The theoretical side of our research studies *queueing systems with heterogeneous servers*. In this chapter we provide the necessary theoretical background and survey relevant literature. We begin with presenting various queueing models with heterogeneous servers in Section 3.1. Then, as our study is concerned with *fair routing* question, we provide a literature review on fairness in service systems from the Behavioral Sciences point of view (Section 3.2). We conclude with some background on a game-theoretic approach of costs sharing in Section 3.3.

3.1 Queueing Models

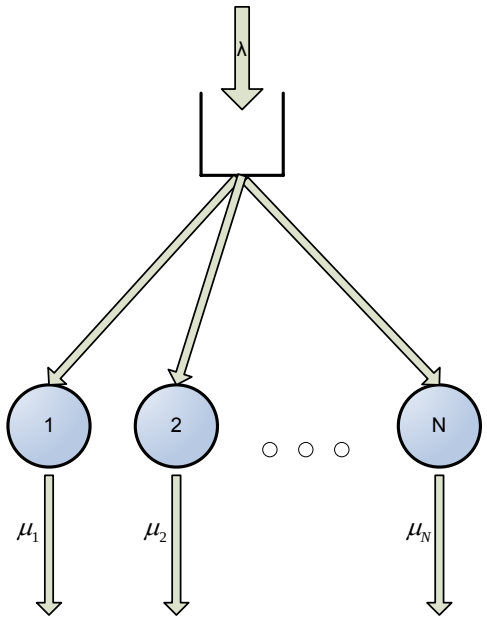
In this section we present several queueing models that will be used in our study. As part of the descriptions, we survey the literature related to each model. In addition, we introduce the QED (Quality and Efficiency Driven) asymptotic regime in Subsection 3.1.4.

3.1.1 The Slow Server Problem

Consider the following queueing system (see Figure 3.1): homogeneous customers arrive according to a Poisson process with rate $\lambda > 0$, N heterogeneous servers operate independently in parallel, service times for server i are i.i.d. exponential with rate μ_i,

and the system has one waiting line with infinite capacity. Each customer requires a service from any of the servers, and each server can serve only one customer at a time. The queueing discipline is FCFS (first-come-first-served), non-preemptive (the service of a customer cannot be interrupted once started), and work-conserving (there are no idle servers whenever there are delayed customers in the queue). In addition, we assume that all interarrival and service times are statistically independent.

Figure 3.1: Queueing System with Heterogeneous Servers



The literature surveyed below studies the utilization of the slow server in such queueing systems with heterogeneous servers. It addresses the problem of finding the best operating policy in order to minimize the steady-state *mean sojourn time* of the customers in the system, which is equivalent to minimizing the long-run *average number* of customers in the system, due to Little's law. Although systems with heterogeneous servers have been studied before, this problem is first posed (to the best of our knowledge) by Larsen in his Ph.D. dissertation in 1981 and published two years later with his Ph.D. advisor Agrawala [32]. Independently, it is addressed by Rubinovitch [45], who is the one to coin the terminology - "the slow server problem". Both [32] and [45] study a system with two servers: fast and slow (i.e. $N = 2,\ \mu_1 > \mu_2$). Larsen looks at the problem in the context of computer systems and communication networks, and Rubinovitch views the fast server as new equipment which arrives to replace an older (slower) equipment. Both note that in systems with human servers, it is quite usual to encounter heterogeneous performance.

Rubinovitch [45] looks at three different scenarios, according to the probability with which a customer who arrives when both servers are idle chooses a server. When at random, that was defined as the system with *uninformed customers*, in the sense that a customer does not know which of the servers is the fastest. Respectively, the case when the customer chooses the fast server is defined as the case of *informed customers*. Then there is a more general model of *partially informed customers*: a customer who arrives to an empty system joins the fast server with probability p and the slow one with probability $1-p$. For each of the three cases, Rubinovitch identifies a critical number ρ_c, as a function of the service rates, such that if the traffic intensity ρ of the system ($\rho = \frac{\lambda}{\mu_1+\mu_2}$) is below this number, the slow server should not be used; namely, the *sojourn time* without the slow server is shorter than the one with it. The single server system obtained when the slow server is removed (under this condition) is always stable ($\lambda < \mu_2$).

Significant attention has been given to *threshold* policies - policies which allocate a customer from the queue to the fast server whenever s/he becomes available, but use the slow server only once the queue length exceeds a certain threshold. Larsen [32] studies such policies for both infinite and finite queues, and shows that an optimal threshold level is deterministic rather than randomized. Rubinovitch, in his subsequent paper [46], finds the optimal threshold level. He assumes that customers are allowed to stall, i.e., wait for a busy fast server even when the slow server is free, and finds the optimal number of stalling customers, when the objective is to minimize the mean time spent in the system.

Lin and Kumar [34] show that in systems with unbounded queues, the optimal policy which minimizes the mean sojourn time of customers in the system is of a threshold type. Stockbridge [49] analyzes finite queue systems and shows that here the optimal control policy is not always of a threshold type - sometimes it is "better" not to use the slow server at all and lose customers through blocking. This might happen in conditions of heavy traffic, large queue and certain slow server's rate, when filling up the queue and rejecting additional customers reduce the number of customers in the system by more than accepting a customer for slow service (and the criterion favors reducing the number of customers). This suggests that the objective of minimizing the average sojourn time may not be always appropriate, for example, when one of the goals is to maximize throughput or to minimize the fraction of lost customers.

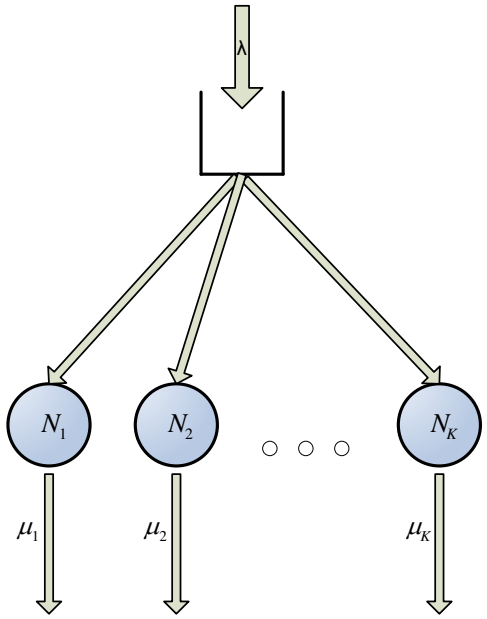
Cabral [12] extends the slow server problem from 2 to n heterogeneous servers. For the case with uninformed customers he shows that there exists a value of the arrival rate below which the slowest server (the one with the smallest μ_i) should not be used and above which should be used. Later, Cabral [13] proves that, for any two servers, the probability that the faster among them is busy is smaller than the probability that the

slower one is busy, and the effective service rate of the faster server is higher than that of the slower one (see our Theorem 4.4.1, which we proved independently). Except for [13], none of the studies presented above attends to the issue of *fairness* towards servers.

3.1.2 Inverted-V Model

Now let us look at the extension of the previously discussed model: the $\wedge$**-model**, or **Inverted-V model** (the terminology coined by [3]) - see Figure 3.2. The system consists of K server pools: pool i contains N_i statistically identical servers, each with exponential service time at rate μ_i (namely service rates are the same within each pool but vary among the pools). The total number of servers in the system is $N = \sum_{i=1}^{K} N_i$. Upon arrival, a customer is routed to one of the available pools (which has one or more idle servers), or joins the centralized queue if all the servers in all the pools are busy. All the other assumptions of previous subsection hold as well, and if we take $N_i = 1,\ \forall i = 1, \ldots, K$ we recover exactly the previously discussed model with $N = K$ servers - referred to from now on as the *Inverted-V Model with Single-Server Pools*.

Figure 3.2: Inverted-V System



Armony [3] proposes for inverted-V system the routing policy of *Fastest Servers First (FSF)*, which assigns customers to the fastest available pool. She shows that FSF is asymptotically optimal in the Quality and Efficiency Driven (QED) regime, in the sense

that it minimizes the steady state *mean waiting time*, as the arrival rate and number of servers grow large. (We introduce the QED regime in Section 3.1.4). Armony and Mandelbaum [4] extend this result to a system with abandonments. But the FSF policy could not be *fair* to the servers (we discuss the notion of "fairness" in Section 3.2), as asymptotically only the slowest servers have any idle time - thus Armony and Ward [5] propose alternative policy (see below). Additional references to papers that study the dynamic control of the $\wedge$-system in the QED regime can be found in [5]: prior to Atar [7], no research deals with the fairness-towards-servers issue [5].

Atar [7] studies a single-server pools model, where the number of servers N and their service rates $\{\mu_k\}$ are random variables. He analyzes two routing policies in the QED regime: the first one routes an arriving customer to the server that has been idle for the longest time among all idle servers (in a deterministic environment, this policy is called in [5] *Longest-Idle Server First (LISF)*). In the second policy, a customer joins the available server with highest service rate - in a deterministic environment this policy is equivalent to FSF. Atar [7] shows that, for the first policy, the duration of the last idle period is asymptotically balanced among servers that are idle at a given time - thus, this policy expresses a certain form of fairness towards servers.

Armony and Ward [5] analyze the case of $K = 2$ server pools. They address an extension of LISF routing: *Longest-Weighted-Idle Server First (LWISF)*, which is designed to solve the fairness issue by routing to the server that has idled the longest, while maintaining that the fraction of idle servers in pool k ($k = 1, 2$), out of total number of idle servers, equals f_k. Armony and Ward [5] propose a threshold policy that asymptotically achieves these target idleness ratios while minimizing steady-state expected waiting time, namely it outperforms LWISF.

Atar, Shaki and Shwartz [8] propose a policy that routes a customer to the pool with the *longest cumulative idleness* among the available pools, in an attempt to achieve fairness towards servers. The policy is referred to as the *Longest Idle Pool First (LIPF)*; the more general version of it is the longest *weighted* cumulative idleness policy. Atar et al.[8] show that, in the QED regime, the policy balances cumulative idleness among the pools. The authors present a terminology of a "blind policy" - a policy that requires none or minimal information on the parameters of the system or the system state at the time of routing decision; being a blind policy is one of the advantages of LIPF policy.

Tezcan [50] attends to the servers' fairness issue as well: he analyzes two routing policies for a *distributed parallel server system* with multiple server pools, and compares their performance to a corresponding inverted-V system. We survey his work, and distributed

systems in general, in the following Section 3.1.3.

The papers presented above (and our research as well) deal with a single customer class. However, some studies cover a more general class of models called *Parallel-Server Systems*: service systems with multiple server pools and multiple customer classes. The problem of dynamic control of such systems is often referred to as "skill-based routing", borrowing the terminology from the world of Call Centers. Gurvich and Whitt analyzed this problem in several papers; in the latest [25], they consider a family of routing policies called *Queue-and-Idleness-Ratio (QIR)* rules. These rules state that (1) a server that becomes available chooses a customer to-be served next from the class whose queue length exceeds by most a predetermined proportion of the total queue length; (2) an arriving customer is routed to the server pool whose number of idle servers exceeds by most a predetermined proportion of the total number of idle servers. QIR controls are shown to drive both the queue-length and the idleness process to the predetermined proportion, in the QED regime.

Alanyali and Hajek [1] studied similar systems (with several customer types and a finite set of *locations*) even before, in the context of load sharing networks. They analyze resource allocation strategies, that aim to achieve fairness by load balancing. The authors in [1] propose a *Least Load Routing (LLR)* policy, that assigns a customer to the location with the least load (where the *load* is measured by the number of customers assigned to this location). The LLR policy and its variants are demonstrated to lead to a fluid type limit and to be asymptotically optimal.

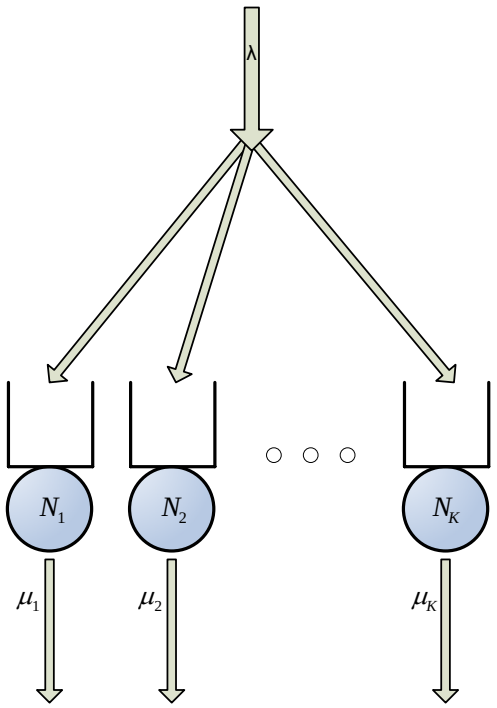
3.1.3 Distributed Systems

We proceed to review another type of queueing systems - a *distributed parallel server system*. The difference from the inverted-V system is that here each pool has its own dedicated queue, as opposed to a single centralized queue (see Figure 3.3). A customer, immediately upon arrival, must be routed to a server pool/queue, following some routing policy. The latter here is more complex than in the inverted-V system, as one should address the case when a customer arrives to the system with all servers busy (which queue to join?), while in the inverted-V system customers automatically join a centralized queue.

There exists quite an extensive literature on distributed systems and their load balancing (one can find references in [50]). Tezcan [50] analyzes two routing policies: the Minimum-Expected-Delay-Faster-Server-First (MED-FSF) and the Minimum-Expected-Delay-Load-Balancing (MED-LB). Under both policies, if a customer arrives to the system

when all the servers are busy, s/he is routed to the queue that has the minimum expected delay. If not all servers are busy, then under the MED-FSF s/he is routed to the fastest available pool and under the MED-LB to the least utilized available pool, where the utilization of the pool j, $1 \leq j \leq K$, at each moment is given by the total number of customers being served by servers in pool j divided by N_j (the total number of servers in pool j). Tezcan shows that MED-FSF asymptotically minimizes the stationary total queue length and delay probability in the QED regime, and MED-LB balances the average utilizations over all the server pools in the QED regime. Next he shows that, under both policies, a distributed system performs (asymptotically) as well as the corresponding inverted-V system, and both systems perform as well as a corresponding $M/M/N$ system.

Figure 3.3: Distributed System



3.1.4 The QED Asymptotic Regime

The Quality and Efficiency Driven (QED) regime (otherwise called the *Halfin-Whitt regime*), has been considered in the papers presented in Section 3.1.2 ([3], [7], [5], [50], [25] and [8]) - it is considered in our research as well (see Chapter 5). The QED regime was first discovered by Erlang [19]; it was mathematically formalized by Halfin and Whitt

[26] (further references can be found in the papers stated above). It can be informally defined as follows: A system with a large volume of arrivals (demand) and many servers (supply, or capacity) is operating in the QED regime if (a) the delay probability is neither near 1 nor near 0, or (b) its total service capacity is equal to its demand up to a safety capacity, which is of the same order of magnitude as the square root of the demand. Characterization (a) describes the quality aspect, and characterization (b) points at a high server efficiencies - thus the QED regime achieves high levels of both service quality and system efficiency, by carefully balancing between the two [3].

We note that (b) above, formulated in terms of arrival rate and service capacity, is suitable for the $\wedge$-design, while the prevalent QED definition for Erlang-C ($M/M/N$) systems is given through the following staffing level formula (known as "the square-root safety staffing principle"):

$$N \approx R_\lambda + \beta\sqrt{R_\lambda}, \quad 0 < \beta < \infty, \tag{3.1}$$

where $R_\lambda = \lambda/\mu$ denotes the average *offered load*, and β is a *Quality-of-Service (QoS)* parameter - the larger the value of β, the higher is the service quality. (If abandonments are allowed, for example in Erlang-A ($M/M/N+M$), $\beta \leq 0$ is allowed as well).

3.2 Fairness

In order to discuss *fair* routing policies we should first obtain some insight on the notion of *fairness* (*justice* and *equity* are alternative terms) in service systems, with the help of Behavioral Sciences. One can analyze fairness towards *customers* and fairness towards *servers*. There is a vast amount of literature on measuring fairness in queues from the customer point of view ([9], [33], [42] are a few examples). Different aspects are investigated (for example, single queue versus multi-queues, or FCFS versus other queueing disciplines), but all agree that FCFS policy is essential for justice perception. Consequently, customer satisfaction in a single queue is higher than in multi-queues [33]; and waiting in a multi-queue system produces a sense of lack of justice even when no objective discrimination exists [42].

The literature on justice from the server point of view is concerned with *Equity Theory* [27], according to which the workers perceive the level of justice they are treated with by comparing their and others ratios of the outcomes from the job to their inputs to the job. Specifically, if the outcome/income ratio of the individual is perceived to be unequal to others, then inequity exists. The larger the inequity the individual perceives (either

underreward or overreward), the more uncomfortable s/he feels and the harder s/he works to restore equity [27]. In [11] it is shown that in customer service centers, servers' equity perception has a positive influence on their performance and job satisfaction. References to additional studies on the importance of perceived justice amongst employees can be found in [5]. Fleurbaey and Maniquet [20] examine a production model with heterogeneous servers where the inequality of servers' skills is not attributed to their responsibility (but is due to the natural endowments distribution). Thus, the researchers aim to neutralize the inequalities in skills rather than reward the more "talented" or efficient server.

3.3 Game Theory

In Chapter 7 we shall attempt to examine our ED-to-IW process from the point of view of *Game Theory*, applying a *costs sharing* approach. Here we view the wards as "players" that pool their resources to serve the ED patients. There are several fair cost allocation approaches (some are surveyed in [51]): the most prominent one is the *Shapley value* method, first introduced by Shapley [47]. The case we study is when, targeting for efficiency and cost savings, servers *pool* their resources (service capacities) to serve the union of their customers. Examples of costs allocation after pooling in queueing systems, with the Shapley value approach, are treated in [21], [22] and [44]. Gonzáles and Herrero [22] pose a cost-allocation problem in sharing an operating-theater, where "players" are different medical disciplines (surgical procedures); Garcia-Sanz et al. [21] analyze several variations of the model in [22]. Finding a *core* of the game is another approach to allocate congestion costs among the players in a fair way - this method is used by Anily and Haviv in [2].

As a coalition's *cost* (or its *characteristic function*, in the terminology of cooperative games) one can choose some optimality criterion, which reflects the cost of congestion (this cost is reduced as a result of the players' cooperation). For example, [2] and [44] use the steady-state mean number of customers in the system, [22] - the average waiting time and [21] - both waiting time and sojourn time in the system, all in order to quantify congestion costs.

Maniquet [36] adopts the Shapley value approach as well, but in his problem the players are customers waiting for some service. The questions addressed in [36] are: how should a queue be organized, when one accounts for each customer's waiting cost (which is his or her impatience), and how to design monetary compensations for waiting customers in a fair way. An additional interesting games-theoretic research is [48], which studies the

aircraft landing problem. The authors in [48] introduce several heuristics that determine a landing order and feasible landing times for a set of flights at a runway: different heuristics provide schedules with different trade-offs between efficiency, cost, delay and fairness (several possible fairness measures are analyzed).

Chapter 4

Inverted-V Model: Exact Analysis

We model the process of patients routing from the ED to the IWs as a queueing system with heterogeneous pools with i.i.d. servers; arrivals to the system are patients to-be-hospitalized in the IWs, pools symbolize the IWs, which indeed have different service rates (1/ALOS) and the number of servers in each pool corresponds to the number of beds in each ward. In order to create a tractable Markovian system, we assume that arrivals to the wards occur according to a Poisson process and LOS in wards are exponentially distributed (both assumptions are important for analytical tractability, but they are inaccurate reality-wise - see [35] for empirical findings about arrivals and LOS). Although, as we saw in Chapter 2, patients to-be-hospitalized in the IWs are classified into several categories, we analyze here a single customer class model. This certainly presents a limitation for application of our theoretical results (see Chapter 9).

In Chapter 3 we presented two general queue-architectures for systems with heterogeneous server pools: a *centralized queue* and *distributed queues*. The system under centralized queue forms the inverted-V model ($\wedge$-model) - we analyze such a model in this chapter (in steady-state) and in the next chapter (in the QED regime). The distributed server system (each pool has its own dedicated queue) is analyzed in Chapter 6. The question of which model better reflects the reality of our ED-to-IW process will be discussed at the end of Chapter 6.

In Section 4.1 we introduce the *Randomized Most-Idle (RMI)* routing policy. We discuss its general queue structure in Section 4.2, and its other favorable properties in Sections 4.3 and 4.4. We prove some additional interesting results in Section 4.5 and present steady-state formulae of several performance measures in Section 4.6. Finally, in Section 4.7 we present several yet-unproved conjectures.

4.1 Introducing Randomized Most-Idle (RMI) Routing Policy

We consider the following routing policy: each arriving customer is assigned to one of the available pools, *with probability that equals the proportion of idle servers in this pool out of the overall number of idle servers in the system.* Formally, denote by $\mathcal{I}_i(t)$ the number of idle servers in pool i at time t. A customer that arrives at time t will be routed to pool i with probability $\frac{\mathcal{I}_i}{\sum_{j=1}^{K}\mathcal{I}_j}$ (unless $\mathcal{I}_i = 0, \forall i = 1, 2, \ldots, K$, in which case the customer joins the queue). We refer to this routing policy as **RMI** (for *Randomized Most-Idle*).

As the servers within each pool are homogeneous, RMI routing is equivalent to choosing a server out of all idle servers at random. To illustrate this let us consider the following example: assume a customer arrives to the system with two available pools: pool i has two available servers and pool j - three available servers. Thus the customer is routed to pool i with probability $2/5$ and to pool j with probability $3/5$. As in pool i there are two symmetric servers, each one of them will serve this customer with probability $1/5$; similarly, each server in pool j will serve the customer with probability $1/5$. Hence the customer is assigned to one of the five available servers with equal probability.

The reason that we take interest in the RMI policy is that it appears fair, as it chooses a server out of all idle servers at random (we address other properties that imply fairness later). An obvious advantage of RMI is that it is a *blind* policy (using the terminology of [8]) - it does not require any information on arrival rates to the system, pool sizes or service rates at the time of routing decision. The only information it does require is the number of idle servers in each pool at the moment of routing, which is even less than the information required by the LWISF policy proposed by [5] or the LIPF policy proposed by [8]. Both LWISF and LIPF require accumulation of certain system history: time that passed from each server's last service completion (LWISF) or even *cumulative* idleness time of each pool (LIPF) (we shall return to this comparison at the end of Chapter 5).

An analytically appealing feature of the RMI policy is that, when modeled as a Markov chain in continuous time, the system is *reversible* (actually, we conjecture that this is *the only* routing policy under which the $\wedge$-system forms a reversible Markov Jump Process - see Conjecture 4.7.1). Thus, we derive its steady state probabilities in a straightforward manner and provide an exact analysis in steady state (while for the other policies mentioned above this seems impossible - indeed they were analyzed only asymptotically).

4.2 General Queue Structure

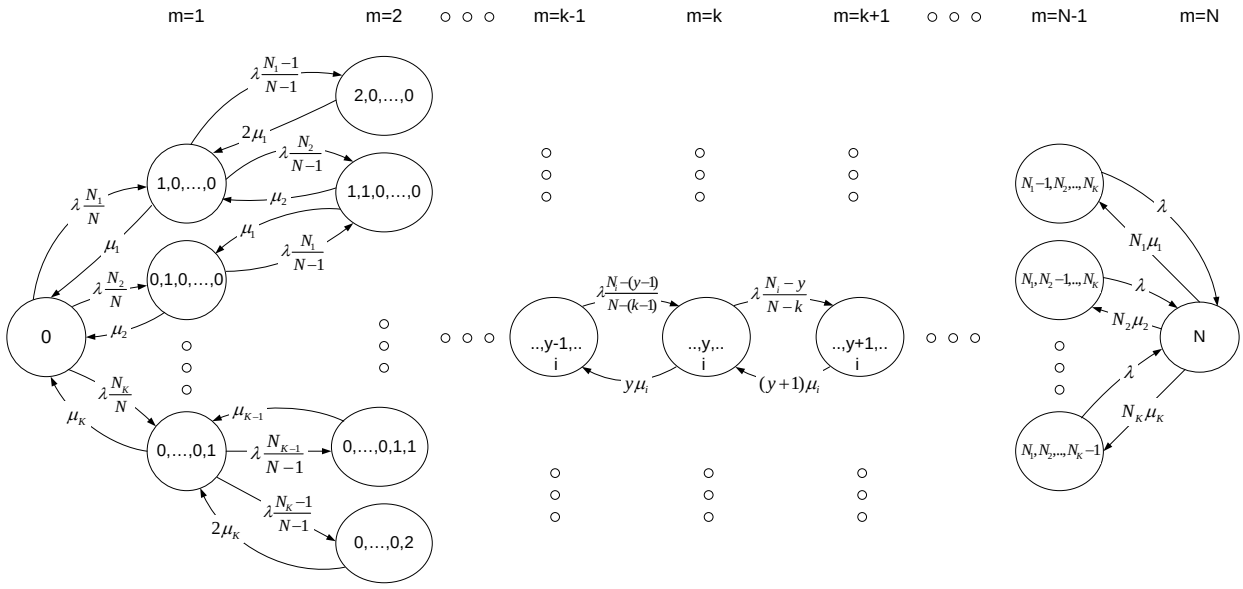
We discussed previously the inverted-V model with infinite queue - the *Delay model*. However, its analysis can be easily extended to a model in which the queue forms a general Markovian *Birth and Death* (B&D) process. In particular, these can be finite-queue and no-queue (*Loss*) models, or models with *inpatient customers* (namely, customers *abandon* the system if their service does not start within a time that is exponentially distributed). In addition, with the help of B&D models, one can capture queue-length dependent arrival rates (for example, in face-to-face queues, the longer the queue the less people tend to join), or state-effects on service rates (for example, servers may increase their service speed when system load is high - an example of such a phenomenon in the health-care world is shown in [17]).

We view such a generalized system as a concatenation of a *Loss model* and its queue. The loss-model means that no queueing is possible, namely, a customer to arrive at the system with all servers busy is "blocked" and leaves the system without receiving service. The queue is presented by a general Markovian B&D process. In general, under every work-conserving policy, the queue of inverted-V system forms a B&D process - as observed in [3]. However, this does not necessarily imply that one can analyze the loss-model and the queue separately and then deduce measures for the *combined* system. This is possible however for RMI routing, as its loss-model is *reversible* (for a definition see [30]) - as we shall show later, in Appendix C. Clearly, every ergodic B&D process is reversible as well; we connect then the two reversible processes via a single state (as will be explained below). It is easy to verify that such a connection forms a reversible process, and one can obtain its stationary distribution from the stationary distributions of its component processes.

First, we show that the inverted-V loss-model under RMI routing is reversible. We model our system as a Markov chain in continuous time $Y = \{Y_t,\ t \geq 0\}$. We characterize each state as a K-dimensional vector $y = (y_1, y_2, \ldots, y_K)$, where y_i is the number of busy servers in pool i ($y_i \in \{0, 1, \ldots, N_i\}$). The state $(N_1, N_2, ..., N_K)$, where all the servers are busy, is denoted as (N). Define $m_y = \sum_{i=1}^{K} y_i$ - the total number of busy servers at state y. The state space of the system is $S = \{0, 1, \ldots, N_1\} \times \ldots \times \{0, 1, \ldots, N_K\}$ ($|S| = \prod_{i=1}^{K} (N_i + 1)$). The corresponding transition rate diagram is shown at Figure 4.1. In order to prove reversibility of the process and find its stationary distribution, we use the following theorem proved in [30]:

Theorem 4.2.1 [30] *Consider an ergodic continuous-time Markov process X on a discrete state space S. Then X is reversible if and only if there exists a collection of positive*

Figure 4.1: Inverted-V Loss Model under RMI as a Markov Chain



numbers $\pi_{x_i},\ x_i \in S$*, summing up to unity, that satisfy the* ***detailed balance conditions****:*

$$\pi_{x_i} q_{x_i x_j} = \pi_{x_j} q_{x_j x_i}\ ,\ \forall x_i, x_j \in S. \tag{4.1}$$

When there exists such a collection $\pi_{x_i},\ x_i \in S$*, it is the unique stationary distribution of the process* X*.*

We write the *detailed balance equations* for our loss system in Appendix C. While solving them, we simultaneously verify the reversibility of the process Y and obtain its stationary distribution:

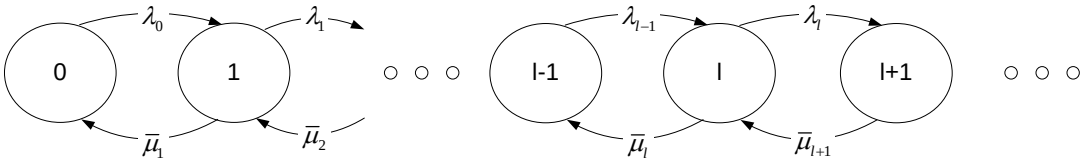
$$\pi_y = \pi_0 \prod_{i=1}^{K} \binom{N_i}{y_i} \frac{(N-m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}}, \qquad \forall y \in S, \tag{4.2}$$

where π_0 is obtained from $\sum_{y \in S} \pi_y = 1$ and equals to:

$$\pi_0 = \left[\sum_{y_1=0}^{N_1} \sum_{y_2=0}^{N_2} \cdots \sum_{y_N=0}^{N_K} \left(\prod_{i=1}^{K} \binom{N_i}{y_i} \right) \frac{(N-m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}} \right]^{-1}. \tag{4.3}$$

Next, let us attend to the *queue* part, which is presented by a general Markovian Birth and Death process: each state l (that represents the number of customers in queue) has birth rate λ_l and death rate $\bar{\mu}_l$ - see the transition rate diagram in Figure 4.2:

Figure 4.2: General Markovian Birth and Death process



Suppose the maximal number of customers in the queue is $c > 0$ (c may be equal to infinity as well). We find the stationary probability $\tilde{\pi}$ of the process, using, as before, its reversibility, by solving the following balance equations:

$$\tilde{\pi}_l \lambda_l = \tilde{\pi}_{l+1} \bar{\mu}_{l+1}, \qquad \forall l = 0, 1, 2, \ldots.$$

We obtain

$$\tilde{\pi}_l = \tilde{\pi}_0 \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}, \qquad \forall l = 1, 2, \ldots, \tag{4.4}$$

where $\tilde{\pi}_0$ is obtained from $\sum_{l=0}^{c} \tilde{\pi}_l = 1$ and equals to:

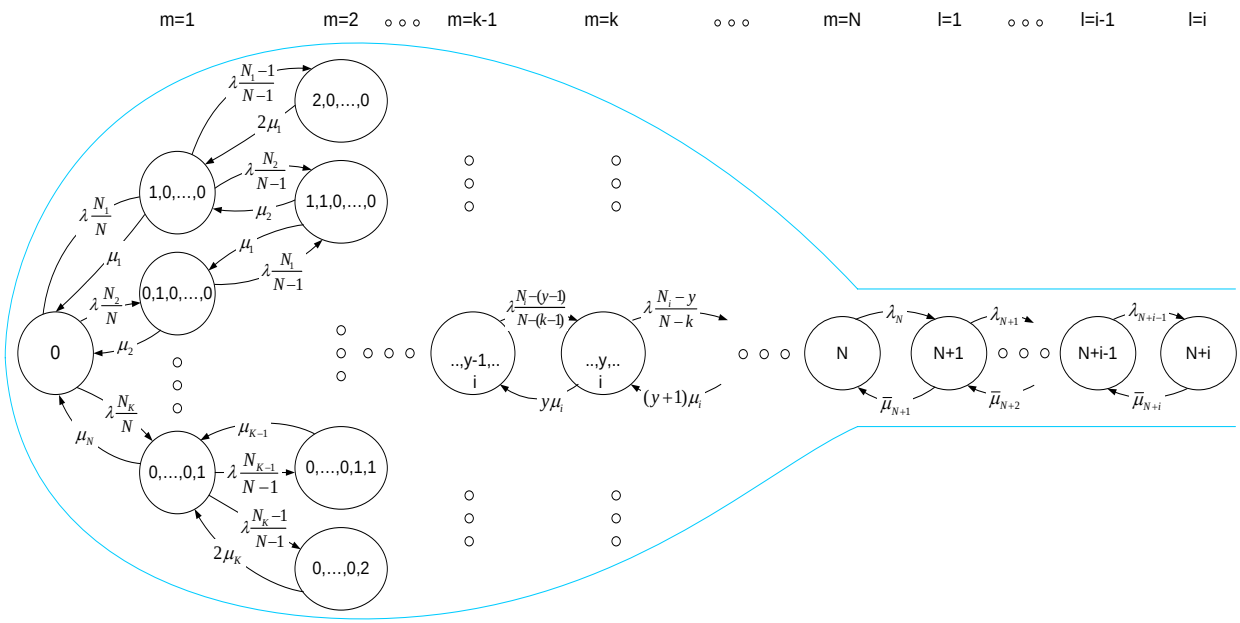
$$\tilde{\pi}_0 = \left[\sum_{l=0}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} \right]^{-1}. \tag{4.5}$$

Note that in case of $c = \infty$, one requires summability of the sum in (4.5) for ergodicity, in order to satisfy the conditions of Theorem 4.2.1.

To connect the described B&D process with our loss-system, we renumber its states: instead of $0, 1, 2, \ldots$ mark $N, N+1, N+2, \ldots$, to represent the total number of customers in the system. We merge the two processes via state (N), namely place the loss-model on the left and the queue process on the right and connect them by consolidating state (N) of the both - see the transition rate diagram of the combined process in Figure 4.3.

We see that the diagram resembles a *kite*, where the "body" of the kite is our loss-system and its "tail" is the B&D queue (the tail can be "empty", which corresponds to a loss-system). Clearly, the way the body and the tail are connected implies that the obtained process is reversible (as they are connected via a single state and this does not add any condition to the detailed balance equations of both). We characterize the states of the whole process in the following way: states with idle servers are K-dimensional vectors $y = (y_1, y_2, \ldots, y_K)$, where y_i is the number of busy servers in pool i; and states with all servers busy are denoted $(N + l)$, where l stands for the length of the queue.

Figure 4.3: General $\wedge$-Model under RMI as a Markov Chain



The stationary distribution of the system is obtained from individual processes stationary distributions. For states $y \in S$ of the loss system, the stationary distribution, as a function of π_0, remains the same as in (4.2), where π_0 will be *recalculated* in a way that *all* new system states' probabilities sum up to 1. For the queue states $(N+l)$, $l = 1, 2, \ldots$, the stationary distribution, as a function of $\tilde{\pi}_0$, remains the same as in (4.4), where $\tilde{\pi}_0 = \pi_N$: the state with an empty queue corresponds to state (N) in the new system. Thus we obtain:

$$\pi_y = \begin{cases} \pi_0 \prod_{i=1}^{K} \binom{N_i}{y_i} \frac{(N-m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}} & , \quad l = 0; \\ \pi_0 \frac{\lambda^N}{N! \prod_{i=1}^{K} \mu_i^{N_i}} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} & , \quad l > 0. \end{cases} \tag{4.6}$$

Here π_0 is deduced from $\sum_{y \in S} \pi_y + \sum_{l=0}^{c} \pi_{N+l} = 1$, and equals to:

$$\pi_0 = \left[\sum_{y_1=0}^{N_1} \sum_{y_2=0}^{N_2} \cdots \sum_{y_N=0}^{N_K} \left(\prod_{i=1}^{K} \binom{N_i}{y_i} \right) \frac{(N-m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}} + \frac{\lambda^N}{N! \prod_{i=1}^{K} \mu_i^{N_i}} \sum_{l=1}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} \right]^{-1}. \tag{4.7}$$

The results of the present section provide a general framework for an Inverted-V system under the RMI policy: with the stationary distribution in (4.6), we can analyze the system under any queue structure, as long as the latter forms an ergodic B&D process. In Sections 4.3 and 4.4, we address such a general model. In Section 4.5 we analyze a delay-model with an unbounded queue, i.e, the transition rates are $\lambda_l = \lambda,\ \forall l = 0, 1, 2, \ldots$, and $\bar{\mu}_l = \sum_{i=1}^{K} N_i \mu_i,\ \forall l = 1, 2, \ldots$ - we then explain how the analysis can be extended to other queue structures.

4.3 Equivalence to Single-Server-Pool Model

In this section, we examine the connection between the inverted-V model under RMI and the inverted-V model with single-server pools under *Random Assignment (RA)* routing. In the course of this section, the latter model is denoted as *SSP* (Single-Server-Pools) system and the former - as *MSP* (Multiple-Server-Pools) system. We analyze both systems in a general form as presented in Section 4.2, when the queue of both is the same B&D process. On the one hand, SSP under RA is a special case of MSP under RMI - we discuss this in Subsection 4.3.1. On the other side, in Subsection 4.3.2 we show that, MSP under RMI can always be represented as SSP under RA (with corresponding servers and their rates). Hence there exists a sort of "egg-hen" connection between those two models.

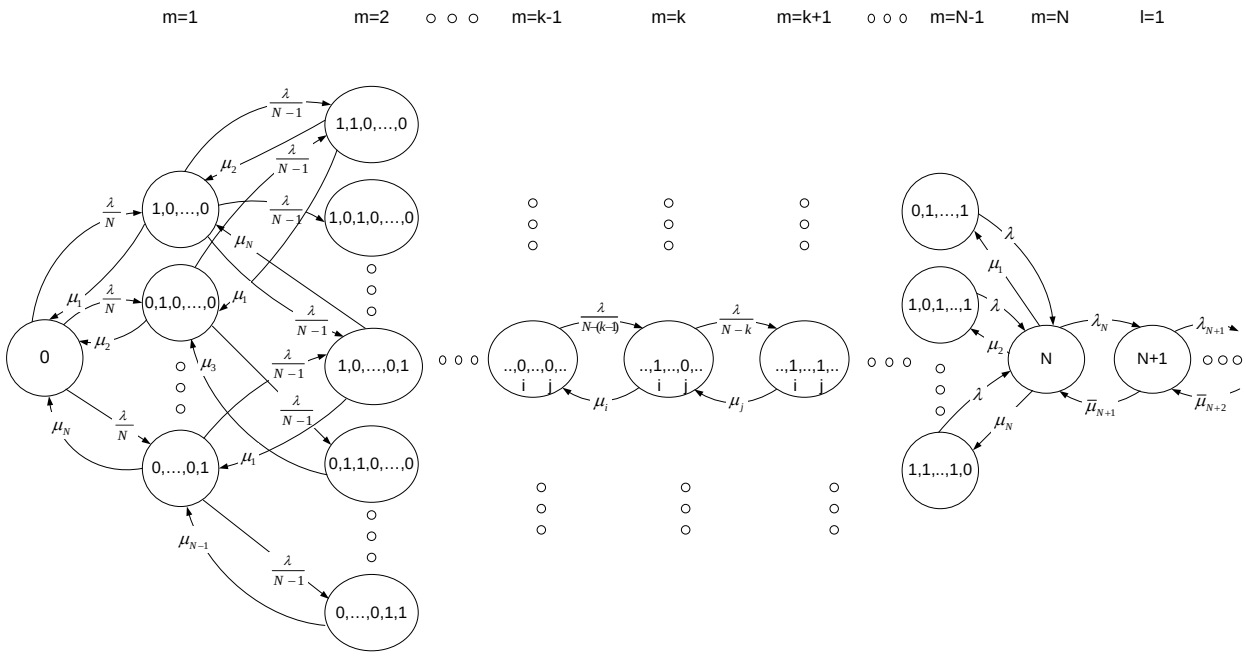
4.3.1 Single-Server-Pool Model

Let us consider a special case of the inverted-V model - the single-server-pool system SSP (discussed in Section 3.1.1): the number of pools equals the number of servers $K = N$, and $N_i = 1,\ \forall i \in \{1, 2, \ldots, K\}$ (in each pool there is just a single server). Clearly, RMI routing for such a model is, in fact, *random assignment* (RA) routing, or the case of *uninformed customers* (using the terminology of [45] and [12]): service assignment of a customer who arrives to a system with k $(1 \leq k \leq N)$ idle servers is at random: s/he joins each one of the idle servers with probability $1/k$. As noted earlier, Cabral [12] examined such systems (the case of the delay-models with unbounded queue) and calculated their steady-state distribution. In our research, we use a state-space and mathematical notations that differ from those in [12].

Similarly to Section 4.2, we model SSP as a Markov chain in continuous time $X = \{X_t,\ t \geq 0\}$. We characterize the states of X in a similar way as in Section 4.2: states with idle servers are denoted as N-dimensional vectors $x = (x_1, x_2, ..., x_N)$: x_i equals 1 if

server i is busy serving a customer and 0 otherwise; and states with all servers busy are denoted as $(N+l)$, where l represents the length of the queue. Again, $m_x = \sum_{i=1}^{N} x_i$ is defined as the total number of busy servers at state x. The corresponding transition rate diagram is shown at Figure 4.4.

Figure 4.4: SSP under RA as a Markov Chain



Due to results of the previous section, SSP is reversible for every ergodic B&D queue structure. Thus, by substituting into (4.6) $K = N$ and $N_i = 1$, $\forall i \in \{1, 2, \ldots, K\}$, we find that π_x - the stationary probability that process X is in state x, equals to:

$$\pi_x = \begin{cases} \pi_0 \dfrac{(N-m_x)!}{N!} \dfrac{\lambda^{m_x}}{\prod_{i=1}^{N} \mu_i^{x_i}} & , \quad l = 0; \\ \\ \pi_0 \dfrac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i} \dfrac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} & , \quad l > 0, \end{cases} \tag{4.8}$$

where π_0 is obtained from $\sum_x \pi_x = 1$ and equals to:

$$\pi_0 = \left[\sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_N=0}^{1} \frac{(N-m_x)!}{N!} \frac{\lambda^{m_x}}{\prod_{i=1}^{N} \mu_i^{x_i}} + \frac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i} \sum_{l=1}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} \right]^{-1}. \tag{4.9}$$

The stationary distribution found is consistent with the one obtained in [12] (for the special case of delay-models with an unbounded queue: the queue transition rates are $\lambda_l = \lambda,\ \forall l = 0, 1, 2, \ldots$, and $\bar{\mu}_l = \sum_{i=1}^{N} \mu_i,\ \forall l = 1, 2, \ldots$). In addition, by taking equal service rates $\mu_i = \mu,\ \forall i = 1, \ldots, N$, we obtain exactly the stationary probabilities of *homogeneous-servers* queueing models (for example, for the delay-models with an unbounded queue - of the *M/M/N* (Erlang-C) model).

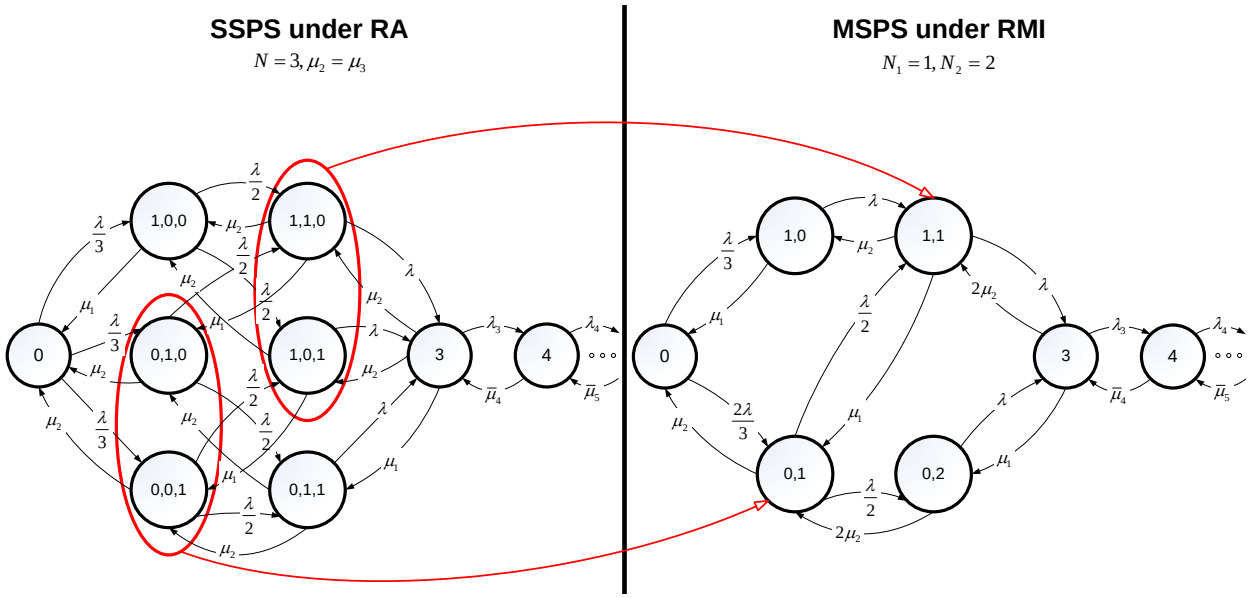
4.3.2 MSP-SSP Equivalence Proof

First, let us introduce some terminology to be used in this section. By *equivalent systems* (or models) we mean that one system can be represented as the other and visa-versa, without losing any information. As we consider Markovian stochastic processes, this implies that the state space of the first system can be reduced to the state space of the second, i.e., the states of the two systems are either *identical* (formally, the states are *identical* if they represent the same system state and have the same incoming and outgoing rates), or the first system states can be combined to form the second system states. In the latter case, we say that several identical states of the first system are *equivalent* to a state of the second system, if they all represent the same system state and the incoming/outgoing rates to/from every state in the second system are equal to a sum of incoming/outgoing rates to/from the states of the first system. The transitions rates of both systems are called *equivalent* in this case.

Theorem 4.3.1 *The two following models are equivalent: MSP with K pools, pool i consisting of N_i servers each with rate μ_i, total number of servers $N = \sum_{i=1}^{K} N_i$, under the RMI policy; and SSP with N servers, where N_i servers have rate μ_i, $(i = 1, \ldots, K)$, under the RA policy.*

Example: Before proceeding to the formal proof, let us illustrate how the theorem works on the following simple example: We take SSP with uninformed customers and three servers: server 1 has service rate μ_1 and servers 2 and 3 have service rate μ_2 each. We compare it to MSP under the RMI policy with two pools: pool 1 with one server of rate μ_1 and pool 2 with two servers of rate μ_2. In Figure 4.5, we compare the transition rate diagrams of the two systems. On the diagram of SSP we notice that there are two pairs of identical states: (0,1,0) and (0,0,1); (1,1,0) and (1,0,1), in the sense that each state within the pair has the same incoming and outgoing rates. If we replace each identical pair by one combined state with appropriate rates, we get exactly MSP. The proof below for the general case is based on similar considerations.

Figure 4.5: SSP vs. MSP



Proof At first, consider all the states of type $(N+l)$, $l \geq 0$: in both systems, all the servers are busy and the queue sizes are equal. As both queues are represented by the same B&D process, these states are clearly identical in both systems. For example, in case of the delay-model, the queues are B&D processes with birth rate λ and death rate $\sum_{i=1}^{N} \mu_i$ for SSP and $\sum_{i=1}^{K} N_i \mu_i$ for MSP, such that: $\sum_{i=1}^{N} \mu_i = \sum_{i=1}^{K} N_i \mu_i$.

The case of idle servers is somewhat more complicated since here the state space is different in the two models. We should show that states in SSP can be aggregated to produce states in MSP in a way that the incoming/outgoing rates to/from every state in MSP will be equal to a sum of incoming/outgoing rates to/from the corresponding SSP states. Let us look at SSP and order all servers according to their service rate (servers 1 to N_1 have rate μ_1, servers N_1+1 to N_1+N_2 have rate μ_2, etc). We notice that, as not all the servers are different, for a Markovian description it is redundant to keep information about each server whether busy or not - it is enough to know how many out of the N_i i.i.d. servers are busy. Hence we can reduce the N components of the state vector to just K components - component i denoting how many servers of type i (with service rate μ_i) are busy.

Consider the states where y_i servers with rate μ_i are busy ($y_i = \sum_{j:\mu_j=\mu_i} x_j$, $i = 1, \ldots, K$). There are several identical states like that, and we can combine them into a single state. If an arrival occurs to these states, one of the idle servers is chosen at random in both systems - the transition rates in both systems are equivalent (rates in MSP are

sum of rates in SSP). For example, in MSP, the transition rate from state $(y_1, y_2, \ldots, y_K)$ to state $(y_1, \ldots, y_j + 1, \ldots, y_K)$, for some $j \in \{1, 2, \ldots, K\}$, is $\frac{\lambda(N_j - y_j)}{N - \sum_{i=1}^{K} y_i}$. In SSP, there are $N_j - y_j$ corresponding transfers (for each idle server of type j), the transition rate of each is $\frac{\lambda}{N - \sum_{i=1}^{K} y_i}$ - the rates indeed sum up to $\frac{\lambda(N_j - y_j)}{N - \sum_{i=1}^{K} y_i}$. The transition rates of service completions are equivalent in both systems as well, as the same servers are busy. For example, the transition rate from state $(y_1, y_2, \ldots, y_K)$ to state $(y_1, \ldots, y_j - 1, \ldots, y_K)$, for some $j \in \{1, 2, \ldots, K\}$, is $y_j \mu_j$. In SSP, there are y_j corresponding transfers (for each busy server of type j), and the transition rate of each is μ_j - the rates indeed sum up to $y_j \mu_j$. ■

Remark 4.3.1 (Stationary Distribution Equivalence.)
When comparing (4.6) and (4.8), the latter for the case where N_i servers have rate μ_i, $(i = 1, \ldots, K)$, (total number of servers is $N = \sum_{i=1}^{K} N_i$ in both systems), we see that the stationary distributions of both systems are equal. For states with $l > 0$ this is trivial. For $l = 0$, states in SSP are aggregated to produce the states in MSP, and summing up their π_x, we get the corresponding π_y. For each state in MSP there are $\prod_{i=1}^{K} \binom{N_i}{y_i}$ corresponding states in SSP. Noting that $\prod_{i=1}^{N} \mu_i^{x_i} = \prod_{i=1}^{K} \mu_i^{\sum_{j:\mu_j = \mu_i} x_j} = \prod_{i=1}^{K} \mu_i^{y_i}$, we observe the equivalence of (4.6) and (4.8).

As a consequence of Theorem 4.3.1, the analysis of SSP under random assignment holds for MSP under the RMI policy, and visa versa. The equivalence between the two systems is useful: while SSP is easier to analyze, MSP represents a wider class of queueing systems - in particular, it pictures more accurately than SSP the reality of our hospital ED-to-IW process.

4.4 Fast Servers Work Less but Serve More

The researches mentioned in Chapter 3 ([45], [32], [46], [34], [49], [12], [3], [7]) show that it is preferable (under two criteria: minimal steady-state sojourn time and minimal steady-state waiting time) to use the faster servers more than the slower servers. But this is obviously unfair towards the fast servers (which get "punished" for being fast by working more) and it gives them an incentive to slow down - an undesirable result for the system as a whole. Thus, there exists a trade-off between operational optimality for the system and fairness towards servers [5]. In this section we investigate this issue for our RMI policy.

Let us fix some notations: denote by B_i the steady-state number of busy servers at pool i ($i = 1, 2, \ldots, K$). Each B_i is a random variable that attains values in $\{0, 1, \ldots, N_i\}$ ($i = 1, 2, \ldots, K$). Introduce another set of random variables: the steady-state *occupancy rate* of pool i is the number of busy servers in pool i divided by total number of servers in this pool: $\tilde{\rho}_i = \dfrac{B_i}{N_i}$ ($i = 1, 2, \ldots, K$). Next, denote by $\bar{\rho}_i$ the *mean* long-run (steady-state) occupancy rate in pool i. As the servers within each pool are symmetric, $\bar{\rho}_i$ also stands for the average *utilization* of servers in pool i - the proportion of time that each server is busy in steady-state. We define the *effective service rate* of server i in system state x ($\forall x : x_i = 1$) as the average number of customers served by this server at system state x per time unit: it equals $\mu_i \pi_x$, where π_x is the stationary probability to be in state x. Finally, by γ_i we denote the average *flux* through pool i (average number of arrivals per pool i server per time unit): $\gamma_i = \mu_i \bar{\rho}_i$, by Little's law. Clearly, γ_i stands also for the average effective service rate of server i.

In Theorem 4.4.1 below we prove that, comparing any two pools, servers' utilization in the faster pool is *lower* than servers' utilization in the slower pool, but the flux in the faster pool is *higher* than the flux in the slower pool. This implies that, taking any two servers, the faster between the two will work less than the slower but will serve more customers than the slower. The result suggests, first of all, some form of fairness: faster servers are "rewarded" by working less and slow servers are "punished" by working more. In addition, operational preferences of the system are accommodated as well: more customers are served by faster servers than by slower servers.

For notational convenience, we prove the theorem for the single-server-pool model under the RA policy (note that Cabral [13] proved this result independently) - by Theorem 4.3.1 the proof is also valid for the RMI policy in the inverted-V model. We also note that the results of the present section hold for every general queue structure (presented in Section 4.2), as the states where all the servers are occupied do not affect occupancy or flux comparison.

Theorem 4.4.1 *In the inverted-V model under the RMI policy, for any two pools j and k: if $\mu_j > \mu_k$, then:*

(1) $\bar{\rho}_j < \bar{\rho}_k$

(2) $\gamma_j > \gamma_k$.

In words, each server in the faster pool has lower utilization (works less) but has higher productivity, or flux (serves more customers) than each server in the slower pool.

Proof As mentioned above, the proof is shown for the single-server-pool model under RA. We take two servers, j and k, so that $\mu_j > \mu_k$. For convenience, in this proof

we denote system states in a slightly different manner than in Section 4.3.1: by listing the servers that are busy at this state. For example, state $(1,3)$ means that servers 1 and 3 are busy and all the other servers are idle (which corresponds to the original state $(1,0,1,0,\ldots,0)$). Now, denote by p_m^j the stationary probability that m servers are busy, including server j, and by p_m^k the stationary probability that m servers are busy, including server k (m takes values in $\{1,2,\ldots,N\}$). For example, $p_1^j=\pi_j$, $p_2^j=\sum_{i\neq j}\pi_{ji}$, $p_3^j=\sum_{i\neq j}\sum_{\substack{l\neq j\\ l\neq i}}\pi_{jil}$, etc. Note that $p_N^j=p_N^k$, as this is the stationary probability that all servers are busy. Since a server's utilization is the steady-state probability that s/he is busy, $\bar{\rho}_i=\sum_{m=1}^N p_m^i$ $(i=j,k)$. In the same manner, $\gamma_i=\mu_i\bar{\rho}_i=\sum_{m=1}^N \mu_i p_m^i$, $(i=j,k)$. In Lemma 4.4.1 below we prove that, for every $m\in\{1,2,\ldots,N-1\}$, p_m^j is smaller than p_m^k but $\mu_j p_m^j$ is larger than $\mu_k p_m^k$. Clearly, this implies:

$$\bar{\rho}_j=\sum_{m=1}^N p_m^j<\sum_{m=1}^N p_m^k=\bar{\rho}_k, \qquad \textit{and}$$
$$\gamma_j=\mu_j\bar{\rho}_j=\sum_{m=1}^N \mu_j p_m^j>\sum_{m=1}^N \mu_k p_m^k=\mu_k\bar{\rho}_k=\gamma_k.$$

■

Lemma 4.4.1 $p_m^j<p_m^k$ *and* $\mu_j p_m^j>\mu_k p_m^k$, $\forall m\in\{1,2,\ldots,N-1\}$.
In words, for every possible system occupancy (number of busy servers in the system), except for full occupancy (all servers are busy), the probability that the faster server is occupied is smaller than the probability that the slower server is occupied. However, the faster server's effective service rate is higher than the slower server's effective service rate in these states.

Proof Denote by $\mathcal{X}\subseteq\{1,2,\ldots,N\}\setminus\{j,k\}$ a possible set of busy servers, that are not j or k. State $(i,\mathcal{X})$ means that the busy servers are i and the servers of $\mathcal{X}$ $(i=j,k)$. State $(j,k,\mathcal{X})$ means that the busy servers are j, k and the servers of $\mathcal{X}$. Substituting the stationary probability that the system is in state $(i,\mathcal{X})$ $(i=j,k)$ from (4.8), we obtain that, for every possible subset $\mathcal{X}$ of $\{1,2,\ldots,N\}\setminus\{j,k\}$:

$$\pi_{j\mathcal{X}}=\pi_0\frac{(N-(|\mathcal{X}|+1))!}{N!}\frac{\lambda^{|\mathcal{X}|+1}}{\prod_{i\in\mathcal{X}}\mu_i\cdot\mu_j}<\pi_0\frac{(N-(|\mathcal{X}|+1))!}{N!}\frac{\lambda^{|\mathcal{X}|+1}}{\prod_{i\in\mathcal{X}}\mu_i\cdot\mu_k}=\pi_{k\mathcal{X}} \quad (4.10)$$

where the inequality follows from $\mu_j>\mu_k$. Next we note that:

$$p_m^j=\sum_{\mathcal{X}:|\mathcal{X}|=m-1}\pi_{j\mathcal{X}}+\sum_{\mathcal{X}:|\mathcal{X}|=m-2}\pi_{jk\mathcal{X}} \quad (4.11)$$
$$p_m^k=\sum_{\mathcal{X}:|\mathcal{X}|=m-1}\pi_{k\mathcal{X}}+\sum_{\mathcal{X}:|\mathcal{X}|=m-2}\pi_{jk\mathcal{X}}$$

The last sum is equal in both expressions, and (4.10) implies that $\sum_{\mathcal{X}:|\mathcal{X}|=m-1} \pi_{j\mathcal{X}} < \sum_{\mathcal{X}:|\mathcal{X}|=m-1} \pi_{k\mathcal{X}}$. Hence $p_m^j < p_m^k,\ \forall m \in \{1, 2, \ldots, N-1\}$.

Now we prove that $\mu_j p_m^j < \mu_k p_m^k,\ \forall m \in \{1, 2, \ldots, N-1\}$. Multiplying each side of the first equation of (4.11) by μ_j and each side of the second equation by μ_k, yields:

$$\mu_j p_m^j = \sum_{\mathcal{X}:|\mathcal{X}|=m-1} \mu_j \pi_{j\mathcal{X}} + \sum_{\mathcal{X}:|\mathcal{X}|=m-2} \mu_j \pi_{jk\mathcal{X}}$$
$$\mu_k p_m^k = \sum_{\mathcal{X}:|\mathcal{X}|=m-1} \mu_k \pi_{k\mathcal{X}} + \sum_{\mathcal{X}:|\mathcal{X}|=m-2} \mu_k \pi_{jk\mathcal{X}}$$

where

$$\mu_j \pi_{j\mathcal{X}} = \pi_0 \frac{(N - (|\mathcal{X}| + 1))!}{N!} \frac{\lambda^{|\mathcal{X}|+1}}{\prod_{i\in\mathcal{X}} \mu_i} = \mu_k \pi_{k\mathcal{X}}, \tag{4.12}$$

and

$$\mu_j \pi_{jk\mathcal{X}} = \pi_0 \frac{(N - (|\mathcal{X}| + 2))!}{N!} \frac{\lambda^{|\mathcal{X}|+2}}{\prod_{i\in\mathcal{X}} \mu_i \cdot \mu_k} > \pi_0 \frac{(N - (|\mathcal{X}| + 2))!}{N!} \frac{\lambda^{|\mathcal{X}|+2}}{\prod_{i\in\mathcal{X}} \mu_i \cdot \mu_j} = \mu_k \pi_{jk\mathcal{X}}, \tag{4.13}$$

where the inequality follows again from $\mu_j > \mu_k$. Equations (4.12) and (4.13) imply: $\mu_j p_m^j > \mu_k p_m^k,\ \forall m \in \{1, 2, \ldots, N-1\}$. ∎

Remark 4.4.1 (Strengthening Lemma 4.4.1.)
Lemma 4.4.1 amounts to an even stronger claim than originally intended: for every possible occupancy of the system, the probability that the faster server is busy is smaller than the probability that the slower server is busy, and the effective service rate of the faster server is higher than the one of the slower server. Equality is achieved only when *all* servers are occupied. This implies that, in each state, the faster server works less, and not only on average overall.

Remark 4.4.2 (Upper and lower bounds.)
The theorem provides upper and lower bounds on the ratio of server utilizations: $\frac{\mu_k}{\mu_j} < \frac{\bar{\rho}_j}{\bar{\rho}_k} < 1$. This suggests that the difference in utilizations of any two servers is more significant the more their service rates differ: for $\mu_k \approx \mu_j$, $\bar{\rho}_k \approx \bar{\rho}_j$, but as μ_k grows smaller than μ_j, the server utilizations' ratio may decrease. The bounds on the flux ratio are the following: $1 < \frac{\mu_j \bar{\rho}_j}{\mu_k \bar{\rho}_k} < \frac{\mu_j}{\mu_k}$, where the second inequality follows from $\frac{\bar{\rho}_j}{\bar{\rho}_k} < 1$. The upper bound is important in the following way: although the fact that faster servers serve more customers contributes to system performance, one should not forget that, in certain cases, higher flux actually implies higher workload. In particular, this is the case in our

ED-to-IW process, because service (treatment) complexity is not homogeneous during a customer (patient) stay (recall Section 2.3.2). For the RMI policy, the servers' flux ratio $\frac{\gamma_j}{\gamma_k}$ is bounded by their service rates ratio; thus, if servers' rates are comparable, the faster server's flux cannot be much higher than the slower's one.

Remark 4.4.3 (Fairness and operational preferences.)
The fact that the utilization decreases the faster the server gets, provides an incentive for servers to work faster, which is positive on one side but, on the other, might harm service quality, if one starts serving his/her customers *too* fast. Another issue is that if the slow server is not responsible for being slow (for example, a new server vs. a senior experienced server), "punishing" him/her for being slow appears quite unfair. However, as noted in the previous remark, higher flux may be considered in certain cases as a "punishment" as well. Hence, in order to decide which server is better off: the faster or the slower, or alternatively, what are servers' incentives (to increase or decrease his/her service rate?), one must account for the servers' utility functions (the ones they strive to optimize), which combine both criteria (utilization and flux) - we attempt to do so in Sections 7.2 and 8.3.

4.5 Queue Length Performance Criterion

Similarly to the previous section, in the present section we state and prove a result for single-server-pools under the RA policy; by Theorem 4.3.1, it is also valid for the RMI policy in the inverted-V model. We analyze here a delay-model with an unbounded queue - as presented in Figure 3.1. We discuss extensions of this analysis to other queue structures in Subsection 4.5.3.

As mentioned in Section 3.1.1, some researches have used *mean sojourn time in the system* as the performance criterion they strived to optimize: [45] shows that, in case of two heterogeneous servers, under some circumstances it is better to discard the slow server; [12] shows the same for n servers. The question we ask here is whether such a phenomenon prevails under the criterion of *mean waiting time in queue*, or equivalently by Little's law, *mean number of customers in queue* (see Remark 4.5.1 for the rational behind this alternative criterion).

It makes sense that, when minimizing average waiting time, it is never advantageous to discard the slow server. More than that, we prove that, via appropriate coupling, the queue length and waiting time in a system with N servers (*System-N*) are path-wise dominated by the queue length and waiting time in a system with $N-1$ servers

(*System-(N-1)*), at all times, when both systems operate under a random assignment policy (System-$(N-1)$ is System-N with one of the servers removed). In Subsection 4.5.1 we prove first queue lengths and then waiting times dominance, both by coupling the two systems in a way that produces a.s. (almost sure) dominance at all times. Thus we deduce stochastic order and the order of means in steady-state. In Subsection 4.5.2 we discuss the case of two servers, for which the result can be proved under more general conditions, and present two conjectures for the case of an arbitrary number of servers.

4.5.1 Coupling Proof

Definitions: We denote the queue length in System-i $(i = N-1, N)$ at time $t \geq 0$ by $X^i(t)$. For each server $j \in \{1, 2, \ldots, i\}$, $i = N-1, N$, denote:

$$l_j^i(t) = \begin{cases} 1 & \text{if server } j \text{ is idle in System-}i \text{ at time } t \geq 0 \\ 0 & \text{if server } j \text{ is busy in System-}i \text{ at time } t \geq 0 \end{cases}$$

We define then the sets of busy and idle servers in each system at time $t \geq 0$:

$$\mathbb{B}^i(t) = \{j \in \{1, 2, \ldots, i\} |\ l_j^i(t) = 0\}$$

$$\mathbb{I}^i(t) = \{j \in \{1, 2, \ldots, i\} |\ l_j^i(t) = 1\}$$

Denote the number of idle servers in System-i $(i = N-1, N)$ at time $t \geq 0$ by $\mathcal{I}^i(t)$: $\mathcal{I}^i(t) = |\mathbb{I}^i(t)|$. Due to work-conservation, it holds that $\mathcal{I}^i(t) > 0$ only if $X^i(t) = 0$ and $X^i(t) > 0$ only if $\mathcal{I}^i(t) = 0$ $(i = N-1, N)$.
Finally, we shall use the following two sets:

$$\mathbb{V}^N(t) = \mathbb{I}^N(t) \setminus (\mathbb{I}^N(t) \cap \mathbb{I}^{N-1}(t))$$

namely, the group of all the servers that are idle in System-N but busy in System-$(N-1)$ and server N if s/he is idle; and

$$\mathbb{V}^{N-1}(t) = \mathbb{I}^{N-1}(t) \setminus (\mathbb{I}^N(t) \cap \mathbb{I}^{N-1}(t))$$

namely, the group of all the servers that are idle in System-$(N-1)$ but busy in System-N.

Theorem 4.5.1 *Assuming initial states as in (I.1) and (I.2),the process $X^N(t)$ is stochastically dominated by the process $X^{N-1}(t)$ in the sense that:*

$$X^N(t) \underset{st}{\leq} X^{N-1}(t)\ ,\ \forall t \geq 0.$$

Proof The proof is based on sample-path coupling arguments, inspired by the proofs of Lemma 3 at [3] and Proposition 3.1 at [4]. We claim that our two systems can be coupled such that the following three properties hold a.s., for all $t \geq 0$.

$$X^N(t) \leq X^{N-1}(t) \tag{1}$$

$$\mathcal{I}^N(t) \geq \mathcal{I}^{N-1}(t) \tag{2}$$

$$\mathbb{V}^{N-1}(t) = \emptyset \tag{3}$$

Establishing property (1) will complete our proof.

Initialization: We start at time $t = 0$, at which we either assume that the properties hold, or, alternatively, start from the same state of both systems, namely:

I.1 If there is no queue, then the same servers are busy in both systems: $\mathbb{B}^N(0) \cap \{1, 2, \ldots, N-1\} = \mathbb{B}^{N-1}(0)$, and server N is either busy or idle (does not matter). Then $\mathcal{I}^N(0) \geq \mathcal{I}^{N-1}(0),\ \mathbb{V}^{N-1}(t) = \emptyset$.

I.2 If there is a queue, then it is equal in both systems: $X^N(0) = X^{N-1}(0)$

We fix a sample path $\{t_j\}_{j=1}^{\infty}$, where t_j is the time of arrival of the jth customer into both systems, i.e. the systems are coupled such that customers arrive into both systems at the same time. Let S be a set that consists of event points of both systems, namely S includes customers' arrival times and service completion times (service starting times coincide either with arrival times - if there is at least one idle server, or with service completion times - if there is a queue). We shall gradually construct S so that, ultimately, S will include all event times of both systems. We initialize S with the arrival times we have just drawn and, as mentioned, gradually add service completion times as we draw them during the procedure. At time 0 we draw service times for each of the busy servers - the same in both systems, namely, for each server $j \in (\mathbb{B}^N(0) \cap \mathbb{B}^{N-1}(0))$ we draw exponential random variable with rate μ_j to be its service time in both systems. (Due to the loss of memory of exponential distribution, it does not matter how long the server was busy before $t = 0$ in either system). We draw then service times for each of the servers in $\mathbb{V}^N(0)$, if there are any; calculate service completion times, add them to S and renumber the events.

Procedure: We prove (1)-(3) by induction on $t_n \in S$. From the initialization, at time $t_0 = 0$ the properties are satisfied. We assume that they are satisfied for all $t \leq t_n$, and show that they hold immediately after an event at t_{n+1} (at times $t_n < t < t_{n+1}$, both system states remain as at t_n, as nothing occurs). Let us go over all possible cases of both

system states at t_{n+1}, describe the procedure of coupling and show that, in each case, the properties are satisfied:

If t_{n+1} is an *arrival* (to both systems), then the following cases are possible:

A1. $\mathcal{I}^N(t_n) = 0$. From our induction assumption (2) it follows that $\mathcal{I}^{N-1}(t_n) = 0$. The customer joins the queue in both systems, hence (1) is satisfied at t_{n+1}: $X^N(t_{n+1}) = X^N(t_n) + 1 \leq X^{N-1}(t_n) + 1 = X^{N-1}(t_{n+1})$. As nothing changes for idle servers in both systems, (2) and (3) remain satisfied.

A2. $\mathcal{I}^N(t_n) > 0$ and $\mathcal{I}^{N-1}(t_n) = 0$. In System-$N$ we draw a server for the arrived customer: s/he joins server $j \in \mathbb{I}^N(t_n)$ with probability $\frac{1}{\mathcal{I}^N(t_n)}$ (to generate random assignment). For this server j we draw service time (exponential with rate μ_j) - which will be as well the new service time for server j in System-$(N-1)$ (unless $j = N$) instead of the service time that was drawn for him/her previously - again we are allowed to do so because of the exponential loss of memory property. Update S with the new service completion time. In System-$(N-1)$ the customer joins the queue. Properties (1), (2) are satisfied: $X^{N-1}(t_{n+1}) > 0 = X^N(t_{n+1})$, $\mathcal{I}^N(t_{n+1}) \geq 0 = \mathcal{I}^{N-1}(t_{n+1})$, and (3) is satisfied as well, as there are no idle servers in System-$(N-1)$.
Note that the opposite case ($\mathcal{I}^N(t_n) = 0$ and $\mathcal{I}^{N-1}(t_n) > 0$) is not possible because of the induction assumption (2).

A3. $\mathcal{I}^N(t_n) > 0$ and $\mathcal{I}^{N-1}(t_n) > 0$. Draw a server in System-$(N-1)$: the customer joins server $j \in \mathbb{I}^{N-1}(t_n)$ with probability $\frac{1}{\mathcal{I}^{N-1}(t_n)}$ (again, to generate random assignment). Due to the induction assumption (3) this server j is available in System-N as well.

(a) If $\mathbb{V}^N(t_n) = \emptyset$, our induction assumption (3) implies $\mathbb{I}^N(t_n) = \mathbb{I}^{N-1}(t_n)$. Then we send the customer to server j in System-N as well (the RA is preserved, as each idle server has an equal probability $\frac{1}{\mathcal{I}^N(t_n)} = \frac{1}{\mathcal{I}^{N-1}(t_n)}$ to be chosen). We draw exponential service time with rate μ_j for both systems and update S with the new service completion time. All the properties hold for t_{n+1}: $X^N(t_{n+1}) = X^{N-1}(t_{n+1}) = 0$, $\mathcal{I}^N(t_{n+1}) = \mathcal{I}^{N-1}(t_{n+1}) = \mathcal{I}^N(t_n) - 1$ and $\mathbb{V}^{N-1}(t_{n+1})$ is still empty.

(b) If $\mathbb{V}^N(t_n) \neq \emptyset$, we draw a server in System-N in the following manner: with probability $\frac{\mathcal{I}^{N-1}(t_n)}{\mathcal{I}^N(t_n)}$ we send our customer to j (note that due to (2) and (3) $\mathcal{I}^N(t_n) > \mathcal{I}^{N-1}(t_n)$) and with probability $\frac{1-\frac{\mathcal{I}^{N-1}(t_n)}{\mathcal{I}^N(t_n)}}{|\mathbb{V}^N(t_n)|} = \frac{1}{\mathcal{I}^N(t_n)}$ - to server $k \in$

$\mathbb{V}^N(t_n)$. (The RA is preserved: for servers in $\mathbb{V}^N(t_n)$ it is obvious; as for servers in $\mathbb{I}^N(t_n) \setminus \mathbb{V}^N(t_n)$: given that one of them is drawn (with probability $\frac{\mathcal{I}^{N-1}(t_n)}{\mathcal{I}^N(t_n)}$), the probability to choose each of them is $\frac{1}{\mathcal{I}^{N-1}(t_n)}$ - and this is exactly the probability with which server j was chosen in System-$(N-1)$.) If in both systems server j was chosen - draw exponential service time with rate μ_j for both systems. If in System-N server N was chosen - draw service times for j and for N. If in System-N some server $k \in \{1, 2, ..., N-1\} \setminus \{j\}$ was chosen, draw service times for j and for k. We know that in System-$(N-1)$ server k is busy as s/he is in $\mathbb{V}^N(t_n)$, thus we replace his/her service completion time with the service completion time we drew for k in System-N (again it is allowed due to the exponential loss of memory property). Update S with the new service completion times. Properties (1)-(3) are satisfied: $X^N(t_{n+1}) = X^{N-1}(t_{n+1}) = 0$, $\mathcal{I}^N(t_{n+1}) = \mathcal{I}^N(t_n) - 1 > \mathcal{I}^{N-1}(t_n) - 1 = \mathcal{I}^{N-1}(t_{n+1})$ and $\mathbb{V}^{N-1}(t_{n+1})$ is still empty as we chose the server in System-N only from those busy in System-$(N-1)$ (and server N).

If t_{n+1} is a *service completion* time

C1. in both systems, then:

(a) If there is a queue in both systems, the first customer in each queue joins the server that has just finished his/her service. We draw service time for this server (note: it is the same server in both systems, as we drew the same service times for the appropriate servers in both systems), and update S with the new service completion time. Each queue becomes one person smaller and there are no idle servers, hence the properties are satisfied at t_{n+1}.

(b) If there is a queue just in System-$(N-1)$, the first customer in it joins the server that has just finished his/her service. We draw service time for this server, and update S with the new service completion time. The properties are satisfied, as $X^N(t_{n+1}) = 0 \leq X^{N-1}(t_{n+1})$, $\mathcal{I}^N(t_{n+1}) = \mathcal{I}^N(t_n) + 1 > 0 = \mathcal{I}^{N-1}(t_{n+1})$ and $\mathbb{V}^{N-1}(t_{n+1})$ is still empty.
Note that the opposite case (a queue just in System-N) is not possible due the induction assumption (1).

(c) If the queue is empty in both systems, $X^N(t_{n+1}) = X^{N-1}(t_{n+1}) = 0, \mathcal{I}^N(t_{n+1}) = \mathcal{I}^N(t_n) + 1 \geq \mathcal{I}^{N-1}(t_n) + 1 = \mathcal{I}^{N-1}(t_{n+1})$ and $\mathbb{V}^{N-1}(t_{n+1}) = \emptyset$ as the server that has just finished his/her service is the same one in both systems.

C2. just in System-N: then the server that has finished his/her service must be server N, as every other server in $\mathbb{B}^N(t_n)$ is in $\mathbb{B}^{N-1}(t_n)$ due to the induction assumption (3), and they finish their service simultaneously (see the way of drawing service times in (A2) and (A3b) above). The properties are satisfied: if $X^N(t_n) > 0$ then $X^N(t_{n+1}) = X^N(t_n) - 1 \leq X^{N-1}(t_n) - 1 = X^{N-1}(t_{n+1}) - 1 < X^{N-1}(t_{n+1})$, otherwise $\mathcal{I}^N(t_{n+1}) = \mathcal{I}^N(t_n) + 1 > \mathcal{I}^{N-1}(t_n) = \mathcal{I}^{N-1}(t_{n+1})$.

C3. just in System-$(N-1)$: then the server that finished his/her service must be idle in System-N, otherwise they would have finished simultaneously. The properties are satisfied: $X^N(t_{n+1}) = 0 \leq X^{N-1}(t_{n+1})$, $\mathcal{I}^N(t_{n+1}) = \mathcal{I}^N(t_n) \geq \mathcal{I}^{N-1}(t_n) + 1 = \mathcal{I}^{N-1}(t_{n+1})$ (this is because at t_n the server that finished his/her service was busy in System-$(N-1)$ and idle in System-N), and $\mathbb{V}^{N-1}(t_{n+1}) = \emptyset$

■

The following lemma will help us in proving waiting times domination and in the proofs of Subsection 4.5.2.

Lemma 4.5.1 *Consider two systems (A and B), both having a single centralized queue under FCFS, non-preemptive and work-conserving discipline. The systems are coupled such that customers arrive into both at the same time. Then, the queue length of System-A is smaller than or equal to the queue length of System B* ***if and only if*** *each customer enters the service in System-B no earlier than in System-A. Formally, define $X^i(t)$ - the queue length in System-i ($i = A, B$); ξ_j^A and ξ_j^B - entrance to the service times of customer j in systems A and B respectively. Then $\xi_j^B \geq \xi_j^A$, $\forall j = 1, 2, \ldots$,* ***if and only if*** *$X^B(t) \geq X^A(t)$, $\forall t \geq 0$.*

Proof - first direction: $\xi_j^B \geq \xi_j^A$, $\forall j = 1, 2, \ldots$, implies $X^B(t) \geq X^A(t)$, $\forall t \geq 0$. Aiming for a contradiction, we assume that the queue in System-A is larger than the queue in System-B, and show that then there exists a customer who enters the service in System-B earlier than in System-A. Let us look at the first time t that the queue in System-A became larger than in System-B: $X^A(t) > X^B(t)$. If t is some customer arrival time, then prior to this arrival the queues were empty in both systems, and the customer enters the service in System-B immediately, but joins the queue in System-A. Then obviously, s/he enters the service in System-B earlier than in System-A. If t is a service completion time, then it occurred only in System-B, and prior to time t the queues in both systems were equal and not empty (as t is the first time that the queue in System-A became larger than in System-B). Then a customer, who at time t leaves the queue and enters the service in System-B, is still queueing in System-A.

Second direction: $X^B(t) \geq X^A(t),\ \forall t \geq 0$, implies $\xi_j^B \geq \xi_j^A,\ \forall j = 1, 2, \ldots$. Aiming for a contradiction, assume that there exists some customer j that at some time point t is waiting in the queue in System-A but is receiving service in System-B (i.e., that customer entered service in System-B earlier than in System-A). Assume that after this customer, $c \geq 0$ more customers arrived to both systems till time t. Then, due to the FCFS discipline, the queue in System-A at time t is at least $c+1$ but the queue in System-B is c at the most, contradicting the $X^B(t) \geq X^A(t)$ assumption. ∎

Corollary 4.5.1 *Under the same coupling setting as in Theorem 4.5.1, waiting times in queue in System-N are path-wise dominated by waiting times in System-$(N-1)$. Namely, denote by W_j^i the waiting time of customer j in System-i $(i = N-1, N)$. Then $W_j^N \leq W_j^{N-1},\ \forall j \in \{1, 2, \ldots\}$.*

Proof Follows directly from the second direction of Lemma 4.5.1 (we proved $X^{N-1}(t) \geq X^N(t),\ \forall t \geq 0$, in Theorem 4.5.1): if customers enter the service in System-N no later than in System-$(N-1)$, their waiting times in System-N are no longer than in System-$N-1$. ∎

Remark 4.5.1 (The rational behind choosing queue length as a criterion.)
There may be several reasons for choosing the queue-length (waiting time) criterion rather than number-in-system (sojourn time) criterion. In our ED-to-IW case, the objective is to minimize the "queue" for transfer to the wards, thus reducing the overload on the ED (reasons for the importance of minimizing waiting times in the ED are discussed in Section 2.3.1). In general service systems, it is plausible that, in certain cases, customers perceive their waiting as more disturbing than being served by a slow server: they might prefer (even unconsciously) to wait less, even at the cost of a longer sojourn time. We discuss this issue as a future research idea in Chapter 9.

Remark 4.5.2 (Steady-state stochastic dominance.)
The proof presented above shows stochastic order of the queue length processes *for all times*. We also proved, separately, stochastic order of the *steady state* queue lengths, namely that the steady state queue length in a system with N servers is stochastically dominated by the steady state queue length in a system with $N-1$ servers. The significance of this proof is that, in Theorem 4.5.1, we show only *weak* stochastic dominance: $X^N(t) \leq_{st} X^{N-1}(t)\ ,\ \forall t \geq 0$; however, for the steady state we verify *strong* stochastic dominance: $L_q^N <_{st} L_q^{N-1}$ (or $P(L_q^N > l) < P(L_q^{N-1} > l)$), where L_q^i is the stationary queue length in System-i $(i = N-1, N)$. The proof can be found in Appendix D.

Remark 4.5.3 (Importance of underlying assumptions for our proof.)
We note that the assumption of Poisson arrivals does *not* play any role in the proofs, thus this subsection results are correct for every general arrivals process. The assumption of exponentially distributed service times, however, is critical for the proof of Theorem 4.5.1: but is it critical for the result itself (i.e., does the theorem hold for other service times distributions)? In addition, Theorem 4.5.1 was proved for the Random Assignment policy, but does it hold for other routing policies? These questions are open - we were not able to prove the theorem for general service times or for other routings (see, however, Conjectures 4.5.1 and 4.5.2). Nevertheless, below we show that our assumptions can be relaxed for $N = 2$ number of servers: indeed, for two servers the theorem holds for every work-conserving non-preemptive FCFS routing policy, under a general service time distribution.

4.5.2 The Two Server Case

In this subsection we aim at proving that the queue length in a system with two servers, under every work-conserving non-preemptive FCFS policy, is path-wise dominated by the queue length in a corresponding *G/GI/1* system. Consequently, the steady-state stochastic order and order of means will be implied. Practically this means that, in the case of two servers, it is never worthwhile to discard the slow server, if the optimality criterion is *mean number of customers in queue*, or equivalently by Little's law, *mean waiting time in the queue*.

We compare the following two systems: the first has two independent servers with general service time distribution with rates μ_1 and μ_2, $\mu_1 > \mu_2$ W.L.O.G.; the second system has only the fast server - the *G/GI/1* model. Arrivals to both systems are according to a general process with rate λ; there is one waiting line with infinite capacity; the queueing discipline is FCFS, non-preemptive and work-conserving. In addition, we assume that all service times are statistically independent.

Let Π be the set of all non-preemptive work-conserving FCFS routing policies. We denote the queue length of the two-server system (*System-2*) operating under policy $\pi \in \Pi$ at time $t \geq 0$ by $X^2_\pi(t)$, and queue length of the one-server system (*System-1*) - by $X^1(t)$. In addition, denote the number of busy servers in System-2 at time $t \geq 0$ by $Z^2_\pi(t)$, and in System-1 - by $Z^1(t)$. Due to work-conservation, it holds that $Z^i(t) < i$ only if $X^i(t) = 0$ and $X^i(t) > 0$ only if $Z^i(t) = i$ $(i = 1, 2)$.

Theorem 4.5.2 *Assuming that at $t = 0$ both systems are empty, the process $X^2_\pi(t)$ is stochastically dominated by the process $X^1(t)$, for every non-preemptive work-conserving FCFS policy $\pi \in \Pi$, in the sense that:*

$$X^2_\pi(t) \underset{st}{\leq} X^1(t) \ , \ \forall t \geq 0.$$

Proof Before proceeding to the formal proof we offer an intuitive explanation: in heavy traffic, it is obvious that two servers can handle more customers than a single server - even a very slow server added to System-1 will reduce the queue load, not to mention that if $\lambda \geq \mu_1$ then System-1 "explodes". In light traffic, an arriving customer joins the fast server in System-1 and one of the servers in System-2. But if the next customer arrives before the previous has left, in 1-system s/he joins the queue while in System-2 s/he joins the other server. If this never happens (very light traffic) - the queue is essentially always empty in both systems, Theorem 4.5.2 still holds in equality.

The formal proof is based on sample-path coupling arguments, inspired by the proof of Lemma 3 at Armony [3] (we use similar notations as well). Let us consider System-2 (during the proof we omit the index "π"). Suppose the jth customer to arrive into the system arrives at time t_j and has a service requirement of η_j, meaning that if this customer is served by server k, $k = 1, 2$, the service duration is η_j/μ_k. Fix a sample path of $\{(t_j, \eta_j)\}_{j=1}^{\infty}$. We look at customers $j = 1, 2, \ldots, n$, for some finite number n (as the theorem holds for every n it holds for $n \uparrow \infty$ as well). Let d_j be the departure time of customer j from the system and D_n - the time at which the last of all the customers $j = 1, 2, \ldots, n$ left the system. Let $S_2 = \{0 \leq s_1 < s_2 < \ldots < s_M = D_n\}$ be the set of all event points; S_2 consists of all customers' arrival times, service starting and completion times, i.e. each s_i is either t_j or d_j, for some $j = 1, 2, \ldots, n$ (and as n is finite, M is finite). In addition, denote a customer j entrance to service time by ξ_j. Either $\xi_j = t_j$ (if $Z_2(t_j) < 2$) or $\xi_j = d_i$ for some $i < j$, due to work conservation. At time points that are not in S_2, the queue length does not change as no event occurs: for every $t : s_i < t < s_{i+1}$, $i \in \{1, 2, \ldots, M-1\}$: $X^2(t) = X^2(s_i)$. We construct the process $X^1(t)$ on the same sample path $\{(t_j, \eta_j)\}_{j=1}^{\infty}$, by adding to S_1 (the set of all event points in System-1) arrival, service starting and ending times, in a way that will now be described.

Initialization: We start at $t = 0$ with both systems empty (but it is easy to relax this assumption); $S_1 = \{\}$. $j = 1$.

Procedure: While $j \leq n$ do:
If $Z^1(t_j) = 0$, assign a customer to the server. Otherwise, add the customer to the queue.

Denote the customer's service starting time in the new system by ξ'_j, and the departure time by d'_j ($d'_j = \xi'_j + \eta_j/\mu_1$); $S_1 = S_1 \cup \{t_j, d'_j\}$; $j = j + 1$.

Now we show that from Procedure we obtain $\xi'_j \geq \xi_j, \ \forall j = 1, 2, \ldots, n$, namely each customer enters the service in System-1 no earlier than in System-2. By Lemma 4.5.1, this implies $X^1(t) \geq X^2(t), \ \forall t \geq 0$.

The proof is by induction on j: $j = 1$ - the first customer arrives to an empty system: $\xi'_1 = \xi_1 = t_1$. Assume that for $j = 2, ..., k-1$: $\xi'_j \geq \xi_j$. We show that $\xi'_k \geq \xi_k$: If $Z^2(t_k) < 2$ then in System-2 the customer enters the service immediately upon arrival. Otherwise, let us look at the interval $[max(t_k, \xi_{k-1}), \xi_k]$. During this interval, k is at the head of System-2 queue, and customers $k-1$ and $l \leq k-2$ are receiving service in System-2. Aiming for a contradiction, assume that customer k is receiving service in System-1 (i.e., s/he entered the service in System-1 earlier than in System-2). This means that the server in System-1 finished serving both l and $k-1$ before one of the System-2 servers became idle. One of these two customers (l and $k-1$) was served by the same server in both systems (the fast one), thus has the same service time. Due to the induction assumption, s/he entered the service in System-1 no earlier than in System-2, hence s/he should leave System-1 no earlier than System-2 - a contradiction. ∎

Remark 4.5.4 (Waiting times order.)
The waiting times order, namely, the proof that under the same coupling setting, waiting times in queue in System-2 (for every policy in Π) are path-wise dominated by waiting times in System-1, follows as in Corollary 4.5.1. Here is an alternative explanation for the mean waiting times order: if a customer encounters wait in a lightly loaded system, it is highly probable that s/he is the only one in queue. Then his/her average wait is $1/(\mu_1 + \mu_2)$ in System-2 and $1/\mu_1$ in System-1.

4.5.3 Several Conjectures

We now present two additional conjectures for systems with an arbitrary number of servers.

Conjecture 4.5.1 *Under the* FSF (Fastest Servers First) *routing policy, Theorem 4.5.1 holds for an arbitrary number of servers N, and for general service times.*

Conjecture 4.5.2 *Under the* SSF (Slowest Servers First) *routing policy, for every number of servers $N > 2$, Theorem 4.5.1* **does not hold** *- namely, it might be optimal to discard the slowest server, under the minimal queue length optimality criterion.*

Finally, we note that the results of this section were obtained for the delay-model with unbounded queue, but they can be extended (at least some of them) to other queue structures (as presented in Section 4.2). For example, we can reformulate Theorem 4.5.1 for finite-queue systems or/and for systems with abandonments. We expect that the proof would not change by much: for impatient customers, one could couple systems, for example, in a way that abandonments will occur simultaneously in both (similarly to [4]). An additional result for systems with abandonments will be an ordering between *abandonment probabilities*, namely:

Conjecture 4.5.3 *The process $Ab^N(t)$ is stochastically dominated by the process $Ab^{N-1}(t)$ in the sense that:*

$$Ab^N(t) \underset{st}{\leq} Ab^{N-1}(t) \ , \ \forall t \geq 0.$$

Here $Ab^i(t)$ is the total number of abandonments up to time t in System-i ($i = N-1, N$). In particular, this implies that the steady-state abandonment probability in System-N is smaller than the steady-state abandonment probability in System-$(N-1)$.

Lemma 4.5.1, however, should be clearly reformulated in order for it to apply to finite-queue or abandonment systems - and similarly, the other results that use this lemma. For loss-systems, the criterion analogous to our queue-length criterion can be the system *loss probability*. We believe that via coupling, similar to the one used in the proof of Theorem 4.5.1, one can prove the loss probabilities order between our two systems, namely:

Conjecture 4.5.4 *The process $Bl^N(t)$ is stochastically dominated by the process $Bl^{N-1}(t)$ in the sense that:*

$$Bl^N(t) \underset{st}{\leq} Bl^{N-1}(t) \ , \ \forall t \geq 0.$$

Here $Bl^i(t)$ is the total number of blocked customers up to time t in System-i ($i = N-1, N$). In particular, this implies that the steady-state loss probability in System-N is smaller than the steady-state loss probability in System-$(N-1)$.

4.6 Performance Measures

In this section we calculate several steady-state performance measures for the inverted-V model under RMI routing. We analyze the system with a general queue structure, as presented in Section 4.2: its stationary distribution is given by (4.6). Denote the stationary *queue length* (number of customers in queue in steady-state) by L_q, and the

stationary *waiting time* in the queue by W_q. First, let us find $P(W_q > 0)$ - the steady-state probability that all servers are busy, or, equivalently by PASTA (Poisson Arrivals See Time Averages), the steady-state probability that a customer arriving to the system is delayed:

$$P(W_q > 0) = \sum_{l=0}^{c} \pi_{N+l} \overset{(4.6)}{=} \pi_0 \frac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i{}^{N_i}} \sum_{l=0}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}, \tag{4.14}$$

where π_0 is given in (4.7). Then we obtain the queue length distribution:

$$P(L_q = l | W_q > 0) = \frac{\pi_{N+l}}{\sum_{i=0}^{c} \pi_{N+i}} \overset{(4.6)}{=} \frac{\frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}}{\sum_{k=0}^{c} \frac{\prod_{i=0}^{k-1} \lambda_i}{\prod_{i=1}^{k} \bar{\mu}_i}}, \qquad \forall l = 0, 1, \ldots, c. \tag{4.15}$$

$$P(L_q = l) = P(L_q = l | W_q > 0) P(W_q > 0) \overset{(4.15)}{=} \frac{\frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}}{\sum_{k=0}^{c} \frac{\prod_{i=0}^{k-1} \lambda_i}{\prod_{i=1}^{k} \bar{\mu}_i}} P(W_q > 0), \qquad \forall l = 1, 2, \ldots, c. \tag{4.16}$$

Next we attend to several special cases of the queue process, which will be used in the next chapter. We start with **Loss models** - no queueing is possible. In the terminology of Section 4.2, this model is the body of a "kite", or a kite with a null tail - its stationary probabilities were found in (4.2). The performance measure of most interest in the loss-model is the steady-state *loss probability*, namely the probability that the system is in state (N) where all the servers are busy. By PASTA, this equals the probability that an arriving customer is blocked. We find the following expression for the loss probability:

$$P(block) = \pi_N = \pi_0 \cdot \frac{\lambda^N}{N! \prod_{i=1}^{K} \mu_i^{N_i}} \overset{(4.3)}{=} \frac{\frac{\lambda^N}{N! \prod_{i=1}^{K} \mu_i^{N_i}}}{\sum_{y_1=0}^{N_1} \cdots \sum_{y_N=0}^{N_K} \left(\prod_{i=1}^{K} \binom{N_i}{y_i} \right) \frac{(N - m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}}}. \tag{4.17}$$

Another popular queue-structure is the **Delay model** with infinite queue, namely, the transition rates of the B&D queue are $\lambda_l = \lambda,\ \forall l = 0, 1, 2, \ldots$, and $\bar{\mu}_l = \sum_{i=1}^{K} N_i \mu_i,\ \forall l = 1, 2, \ldots$. We note that this queue process is similar to the one formed by $M/M/N$ (Erlang-C) with arrival rate λ and individual service rate $\mu = \frac{\sum_{i=1}^{K} N_i \mu_i}{N}$ (the average service rate of the $\wedge$-system servers). Hence, performance measures concerned with delay (queue-length and waiting time distributions) resemble the corresponding measures of this $M/M/N$ system, given a steady state delay. The same applies for finite queues (equivalent to $M/M/N/N+K$) and for models with abandonments (equivalent to $M/M/N+M$, namely Erlang-A).

For delay models with infinite queues, we denote the total *traffic intensity* of the system by

$$\rho := \frac{\lambda}{\sum_{i=1}^{K} N_i \mu_i}. \tag{4.18}$$

We assume that $\rho < 1$ (as noted in Section 4.2, we require this assumption, in the case of infinite queue, for ergodicity). Substituting the transition rates of the queue process in (4.14) and taking $c \to \infty$, we obtain the steady-state delay probability:

$$P(W_q > 0) \overset{(4.14)}{=} \pi_0 \frac{\lambda^N}{N! \prod_{i=1}^{N} {\mu_i}^{N_i}} \sum_{l=0}^{\infty} \frac{\lambda^l}{\left(\sum_{i=1}^{K} N_i \mu_i\right)^l} \overset{(4.18)}{=} \pi_0 \frac{\lambda^N}{N! \prod_{i=1}^{N} {\mu_i}^{N_i}} \left(\frac{1}{1-\rho}\right), \tag{4.19}$$

where π_0 is obtained from (4.7) and equals to:

$$\pi_0 = \left[\sum_{y_1=0}^{N_1} \sum_{y_2=0}^{N_2} \cdots \sum_{y_K=0}^{N_K} \left(\prod_{i=1}^{K} \binom{N_i}{y_i} \right) \frac{(N-m_y)!}{N!} \frac{\lambda^{m_y}}{\prod_{i=1}^{K} \mu_i^{y_i}} \left(\frac{1}{1-\rho}\right)^{(m_y - N + 1)^+} \right]^{-1}. \tag{4.20}$$

There exists a relation between the loss and delay probabilities, identical to the relation between the Erlang-B and Erlang-C formulae:

$$P(W_q > 0) = \frac{P(block)}{1 - \rho[1 - P(block)]}. \tag{4.21}$$

Finally, we deduce the queue length and waiting time distributions by substituting the transition rates to (4.15) and (4.16), and taking $c \to \infty$. The performance measures obtained are indeed similar to the measures of the corresponding $M/M/N$ system:

$$P(L_q = l | W_q > 0) \overset{(4.15)}{=} \frac{\frac{\lambda^l}{(\sum_{i=1}^{K} N_i \mu_i)^l}}{\sum_{k=0}^{\infty} \frac{\lambda^k}{(\sum_{i=1}^{K} N_i \mu_i)^k}} \overset{(4.18)}{=} \rho^l (1-\rho), \qquad \forall l \in \{0, 1, 2, \ldots\};$$

$$\Rightarrow \quad (L_q^N | W_q^N > 0) \overset{d}{=} Geo_0(1-\rho).$$

$$P(L_q = l) \overset{(4.16)}{=} \frac{\frac{\lambda^l}{(\sum_{i=1}^{K} N_i \mu_i)^l}}{\sum_{k=0}^{\infty} \frac{\lambda^k}{(\sum_{i=1}^{K} N_i \mu_i)^k}} P(W_q > 0) \overset{(4.18)}{=} \rho^l (1-\rho) P(W_q > 0), \qquad \forall l \in \{1, 2, \ldots\}; \tag{4.22}$$

$$\Rightarrow \quad EL_q = E(L_q | W_q > 0) P(W_q > 0) = \frac{\rho}{1-\rho} P(W_q > 0); \tag{4.23}$$

$$\Rightarrow \quad EW_q \overset{Little}{=} \frac{EL_q}{\lambda} = \frac{P(W_q > 0)}{\sum_{i=1}^{K} N_i \mu_i \cdot (1-\rho)}; \tag{4.24}$$

$$f_{W_q}(t) = P(W_q > 0) \left(\sum_{i=1}^{K} N_i \mu_i - \lambda\right) e^{(\sum_{i=1}^{K} N_i \mu_i - \lambda)t}, \qquad \forall t \geq 0. \tag{4.25}$$

Recursion for the loss probability:
We find a recursion expression for the loss probability of systems with two server pools. We start with N_1 servers in pool 1 and 0 servers in pool 2, and at each iteration we add a server to pool 2. We denote by A_i the reciprocal of the loss probability in the system with $N_2 = i$ servers in pool 2, namely:

$$A_i = [P(block)]^{-1} \overset{(4.17)}{=} \sum_{y_2=0}^{i} \binom{i}{y_2} \left(\frac{\mu_2}{\lambda}\right)^{i-y_2} \sum_{y_1=0}^{N_1} \binom{N_1}{y_1} \left(\frac{\mu_1}{\lambda}\right)^{N_1-y_1} (N_1 + i - (y_1 + y_2))!.$$

We start with

$$A_0 = \sum_{y_1=0}^{N_1} \binom{N_1}{y_1} \left(\frac{\mu_1}{\lambda}\right)^{N_1-y_1} (N_1 - y_1)! = \sum_{y_1=0}^{N_1} \frac{N_1!}{y_1!} \left(\frac{\mu_1}{\lambda}\right)^{N_1-y_1},$$

which is the reciprocal of the loss probability in $M|M|N_1|N_1$, and obtain the following expression:

$$A_i = \left(\frac{\mu_2(N_1 + i)}{\lambda} + 1 - \frac{\mu_2}{\mu_1}\right) \cdot A_{i-1} + (i-1)\left(\frac{\mu_2}{\lambda}\left(\frac{\mu_2}{\mu_1} - 1\right)\right) \cdot A_{i-2} + \frac{\mu_2}{\mu_1}. \tag{4.26}$$

This recursion is useful for calculating the loss probability in large systems (such as those analyzed in the next chapter). The expression for A_0 can be calculated via known recursions for the $M|M|N_1|N_1$ (Erlang-B) system.

4.7 Additional Conjectures

In this section, we present some additional interesting claims that we believe are to hold, but we do not provide a proof.

Conjecture 4.7.1 *RMI is the only work-conserving non-preemptive routing policy, under which the $\wedge$-system forms a* ***reversible*** *Markov Jump Process.*

In Theorem 4.4.1 we proved that, comparing any two pools, the mean occupancy rate in the faster pool is lower than the mean occupancy rate in the slower pool. We believe that, strengthening the order of the means, the following prevails: comparing any two pools, occupancy rate in the faster pool is *stochastically dominated* by the occupancy rate in the slower pool.

Conjecture 4.7.2 *In the inverted-V model under the RMI policy, for any two pools j and k: if $\mu_j > \mu_k$, then: $\tilde{\rho}_j \leq_{st} \tilde{\rho}_k$. Namely, $P(\tilde{\rho}_j > x) \leq P(\tilde{\rho}_k > x)$, $\forall x \in (0,1)$). In words, the occupancy rate in the faster pool is stochastically dominated by the occupancy rate in the slower pool.*

We wish to compare the performance of the inverted-V system with N heterogeneous single-server-pools, under RMI, with the performance of the corresponding Erlang-C system with N *homogeneous* servers, where service rate of each is an arithmetic average of the heterogeneous system rates: $\bar{\mu} = \frac{\sum_{i=1}^{N} \mu_i}{N}$. Thus, both systems have the same average servers' utility $\rho = \frac{\lambda}{N\bar{\mu}} = \frac{\lambda}{\sum_{i=1}^{N} \mu_i}$. We conjecture that the homogeneous system outperforms its heterogeneous counterpart in steady-state, when the criterion considered is *minimal mean waiting time in the queue*, or equivalently by Little's law, *minimal mean number of customers in queue*. Thus, the steady state queue length in the Erlang-C system is stochastically dominated by the steady state queue length in the corresponding $\wedge-$system under RMI. From here, order of the means is implied: the mean queue length in steady state is always smaller in Erlang-C system than in the corresponding $\wedge-$system under RMI. The claim is easily verified in the QED regime (see Remark 5.1.3), but is harder to prove in steady-state.

We denote the queue length (number of customers in queue in steady-state) in the heterogeneous server system by L_q^H, and the queue length in Erlang-C by L_q^C. Similarly, denote the steady-state queue waiting time in the two systems by W_q^H and by W_q^C respectively.

Conjecture 4.7.3 $P(L_q^C > l) < P(L_q^H > l)$, $\forall l \geq 0$.
In words, L_q^C is stochastically dominated by L_q^H.

As $P(L_q^C > l)) = \rho^{l+1} P(W_q^C > 0)$ and $P(L_q^H > l)) = \rho^{l+1} P(W_q^H > 0)$, the conjecture is in fact reducible to $P(W_q^C > 0) < P(W_q^I > 0)$: the steady-state probability that a customer arriving to the system is delayed, is smaller in Erlang-C than in the corresponding heterogeneous servers system.

Chapter 5

Inverted-V Model: QED Asymptotics

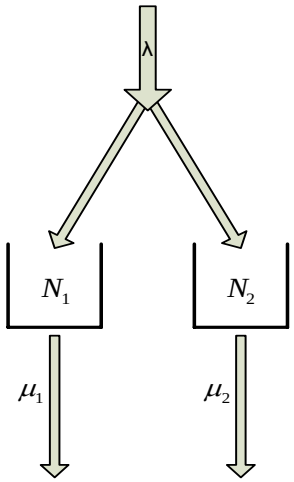
In this chapter, we continue the analysis of the inverted-V model under RMI routing: in Chapter 4 we provided its exact analysis: path-wise and steady state; now we proceed to asymptotic analysis in the QED (Quality and Efficiency Driven) regime. We introduced the QED regime in Section 3.1.4: we saw that it describes systems with many servers and a comparably large volume of arrivals. In QED Delay systems, waiting times are an order of magnitude shorter than service times. In QED Loss systems, the blocking probability is of the order of magnitude $1/\sqrt{N}$. We believe that the QED regime is relevant to our ED-to-IW process: first of all, in Table 2.1 we saw that the number of servers (beds) in each pool (ward) is around 30-50 - the system is large enough. In addition, servers' utilization (beds occupancy) is above 80% (Table 2.2), and the waiting times are indeed an order of magnitude shorter than the service times: hours versus days (Section 2.3.1). The issue of suitability of the QED regime to the ED-to-IW process is studied in [35] (see also Remark 6.4.1).

This chapter is structured as follows: in Section 5.1 we describe QED asymptotics for the RMI policy, in $\wedge$-systems with $K = 2$ server pools. We examine a loss-model (no queueing possible) and present approximation of the loss probability multiplied by $\sqrt{N}$, as found by Momcilovic [39]. In addition, we describe the phenomenon of *dimensionality reduction* (discovered by Momcilovic [39] as well): in the QED limit, the two-dimensional process, which describes the system state, is in fact equivalent to a one-dimensional process. We present some approximations for Delay models in Section 5.1.3. Finally, in Section 5.2, we attend to a class of non-randomized routing policies - *Weighted Most-Idle (WMI)*, and compare them to the RMI policy in the QED regime (also for the case of two server pools), using heuristic and simulation analysis.

5.1 QED Approximation for RMI Routing Policy

In this chapter we examine a Loss model with $K = 2$ server pools (see Figure 5.1). Arrivals to the system are Poisson with rate λ, there are two server pools, where the number of servers in pool i is N_i and the individual service rate in pool i is μ_i, $(i = 1, 2)$. Assume $\mu_1 > \mu_2$. The total number of servers in the system equals to $N = N_1 + N_2$. No queueing is possible: a customer arriving to the system with all servers busy is blocked. Otherwise, the customer is allocated to one of the available servers according to the Randomized Most-Idle (RMI) routing policy.

Figure 5.1: Loss System with Two Heterogeneous Server Pools



In Section 4.6 we found an expression for the steady-state *loss probability*, namely, the stationary probability that all the servers are busy - see (4.17). The expression in (4.17) is cumbersome; in particular, its calculation for large values of λ and N_i may even be impossible (though one might use the recursion relation in (4.26)). This motivates us to find an approximation for the loss probability in the QED regime. We use the following scaling, suitable for the $\wedge$-design, adopted from Armony [3]. We consider a sequence of systems, indexed by λ (to appear as a superscript) with increasing arrival rates $\lambda \to \infty$, and increasing total number of servers N^λ but with fixed service rates μ_1 and μ_2. Both λ and N^λ tend to ∞ simultaneously, in a way that the following conditions hold:

$$\lim_{\lambda\to\infty} \frac{N_1^\lambda \mu_1 + N_2^\lambda \mu_2 - \lambda}{\sqrt{\lambda}} = \delta, \quad or, \quad N_1^\lambda \mu_1 + N_2^\lambda \mu_2 = \lambda + \delta\sqrt{\lambda} + o(\sqrt{\lambda}),\ as\ \lambda \to \infty; \tag{C1}$$

$$\lim_{\lambda\to\infty} \frac{N_i^\lambda \mu_i}{\lambda} = a_i, \quad or, \quad N_i = a_i \frac{\lambda}{\mu_i} + o(\lambda), \quad as \quad \lambda \to \infty, \qquad i = 1, 2; \tag{C2}$$

here $-\infty < \delta < \infty$, and $a_i > 0,\ (i = 1, 2)$, $a_1 + a_2 = 1$.

Condition (C1) is equivalent to the classical square-root safety staffing rule given in (3.1): the *total service capacity*, $N_1^\lambda \mu_1 + N_2^\lambda \mu_2$, is equal to the arrival rate, λ, plus a square root safety capacity, $\delta\sqrt{\lambda}$, where δ is the square root safety capacity coefficient - some quality of service parameter. The scalar a_i denotes the limiting proportion of the service capacity of pool i $(i = 1, 2)$ out of the total service capacity. Due to (C2), the quantity $a_i\lambda/\mu_i$ can be considered as the *offered load* of pool i. Define $\mu := \left(\frac{a_1}{\mu_1} + \frac{a_2}{\mu_2}\right)^{-1}$; then λ/μ can be interpreted to be the *total offered load* of the system. Given this definition of μ, (C1) implies that:

$$\lim_{\lambda\to\infty} \frac{\lambda}{N^\lambda} = \mu, \quad or, \quad N^\lambda = \frac{\lambda}{\mu} + o(\lambda), \quad as \quad \lambda \to \infty. \tag{5.1}$$

Condition (C2) guarantees that the total traffic intensity converges to 1, as λ increases to infinity, namely:

$$\lim_{\lambda\to\infty} \left(\rho^\lambda := \frac{\lambda}{N_1^\lambda \mu_1 + N_2^\lambda \mu_2}\right) = 1. \tag{5.2}$$

Also, $\rho^\lambda \approx \dfrac{\lambda}{N^\lambda \mu}$, in the sense that $\lim\limits_{\lambda\to\infty} \dfrac{\rho^\lambda}{\lambda/N^\lambda\mu} = 1$. Finally, we define

$$\lim_{\lambda\to\infty} \frac{N_i^\lambda}{N^\lambda} = \frac{a_i}{\mu_i}\mu := q_i, \qquad i = 1, 2; \tag{5.3}$$

where q_i is the limiting fraction of pool i servers out of the total number of servers. We assume that the pools are of comparable size: $q_i > 0,\ i = 1, 2$. Clearly, $q_1 + q_2 = 1$ and

$$q_1\mu_1 + q_2\mu_2 = \mu. \tag{5.4}$$

5.1.1 P(block) Approximation

As noted earlier, the steady-state loss probability for a general number of servers was given in (4.17). Rewriting it for the case of two pools, we obtain:

$$P^\lambda(block) = \pi_0^\lambda \cdot \frac{\lambda^{N^\lambda}}{N^\lambda! \mu_1^{N_1^\lambda} \mu_2^{N_2^\lambda}} = \frac{\dfrac{\lambda^{N^\lambda}}{N^\lambda! \mu_1^{N_1^\lambda} \mu_2^{N_2^\lambda}}}{\displaystyle\sum_{y_1=0}^{N_1^\lambda} \sum_{y_2=0}^{N_2^\lambda} \frac{\binom{N_1^\lambda}{y_1}\binom{N_2^\lambda}{y_2}}{\binom{N^\lambda}{y_1+y_2}} \frac{\lambda^{y_1+y_2}}{(y_1+y_2)! \mu_1^{y_1} \mu_2^{y_2}}}. \tag{5.5}$$

Momcilovic [39] calculated QED approximations for the loss probability in (5.5) multiplied by $\sqrt{\lambda}$, under the scaling presented above. Specifically, he proved that

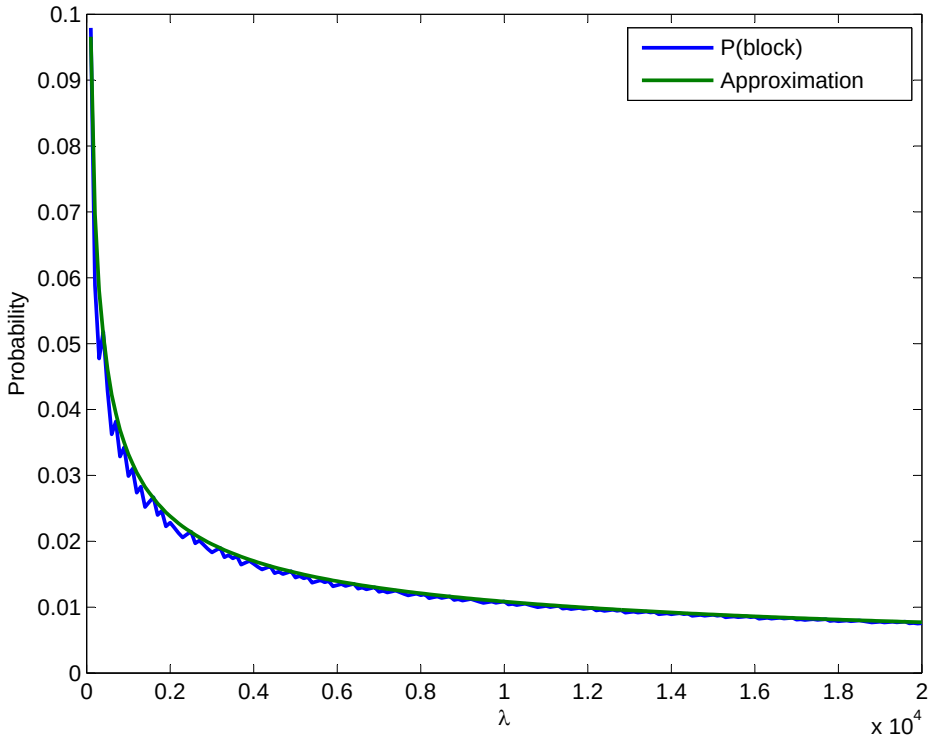
$$\lim_{\lambda\to\infty}\sqrt{\lambda}P^{\lambda}(block)=\sqrt{\hat{\mu}}\frac{\varphi(\delta/\sqrt{\hat{\mu}})}{\Phi(\delta/\sqrt{\hat{\mu}})}, \tag{5.6}$$

where: $\hat{\mu}:=\mu_1 a_1+\mu_2 a_2$ and $\varphi(\cdot),\Phi(\cdot)$ are the density and probability functions of the standard normal distribution, respectively. Using (5.1), we deduce:

$$\lim_{\lambda\to\infty}\sqrt{N^{\lambda}}P^{\lambda}(block)=\sqrt{\frac{\hat{\mu}}{\mu}}\frac{\varphi(\delta/\sqrt{\hat{\mu}})}{\Phi(\delta/\sqrt{\hat{\mu}})}. \tag{5.7}$$

The accuracy of the approximation can be seen in Figure 5.2, which pictures the loss probability and its approximation divided by $\sqrt{N^{\lambda}}$, as λ increases, for the numerical example of $\mu_1=15,\ \mu_2=7.5,\ q_1=1/3,\ q_2=2/3,\ \delta=3$.

Figure 5.2: $P^{\lambda}(block)$ versus $\sqrt{\frac{\hat{\mu}}{\mu}}\frac{\varphi(\delta/\sqrt{\hat{\mu}})}{\Phi(\delta/\sqrt{\hat{\mu}})}/\sqrt{N^{\lambda}}$



In (5.7) we see that $P^{\lambda}(block)$ is of the order of magnitude $1/\sqrt{N}$, which is consistent with the QED regime. In addition, we note that the limit of $\sqrt{N}P^{\lambda}(block)$ is a function of three parameters: δ, μ and $\hat{\mu}$, where δ is a quality of service parameter, and μ and

$\hat{\mu}$ are service rates' averages, in the following sense: As $\lambda \to \infty$, a_i is the proportion of customers served by pool i, out of the total number of customers served. This follows from:

$$\lim_{\lambda\to\infty} \frac{N_i\gamma_i^\lambda}{\lambda(1-P^\lambda(block))} = \lim_{\lambda\to\infty} \frac{N_i\mu_i\bar{\rho}_i^\lambda}{\lambda(1-P^\lambda(block))} = \lim_{\lambda\to\infty} \frac{N_i\mu_i}{\lambda} = a_i. \tag{5.8}$$

Then, if we record, for each customer, the service rate of the pool s/he was served at, and calculate the average over all customers, then $\hat{\mu} := \mu_1 a_1 + \mu_2 a_2$ is the *arithmetic average* and $\mu = \left(\frac{a_1}{\mu_1} + \frac{a_2}{\mu_2}\right)^{-1}$ is the *harmonic average* of service rates of the *customers*. In addition, in (5.4) we saw that $\mu = q_1\mu_1 + q_2\mu_2$, where q_i is the proportion of pool i servers out of the total number of servers, as $\lambda \to \infty$. Namely, μ is an *arithmetic average* of *servers'* rates. As we shall see in Subsection 5.1.2, dimensionality reduction result implies an additional meaning for a_1 and a_2: proportion of idle servers in pool 1 and 2 (out of the total number of idle servers in the system). Thus, μ and $\hat{\mu}$ may be also viewed as the harmonic and arithmetic averages of the *idle servers'* rates.

Remark 5.1.1 (Equal service rates.)
Let us examine the special case of equal service rates $\mu_1 = \mu_2$. Then $\mu = \hat{\mu} = \mu_1 = \mu_2$ and

$$\lim_{\lambda\to\infty} \sqrt{N^\lambda} P_\lambda(block) = \frac{\varphi(\delta/\sqrt{\mu})}{\Phi(\delta/\sqrt{\mu})} = \frac{\varphi(\beta)}{\Phi(\beta)},$$

where

$$\beta \overset{(C1)}{=} \frac{\delta}{\sqrt{\mu}} = \lim_{\lambda\to\infty} \frac{N_1^\lambda\mu_1 + N_2^\lambda\mu_2 - \lambda}{\sqrt{\lambda\mu}} \overset{(5.1),(5.2)}{=} \lim_{\lambda\to\infty} \sqrt{N^\lambda}(1-\rho^\lambda). \tag{5.9}$$

Thus, the approximation is consistent with the Erlang-B approximation, found by Jagerman [28].

5.1.2 Dimensionality Reduction

Momcilovic [39] establishes a dimensionality reduction result with respect to the *idle servers* process. Namely, he shows that asymptotically, a one-dimensional process becomes sufficient to describing the two-dimensional process of the number of idle servers in each pool. The dimensionality reduction result implies that, in the limit, the number of idle servers in each server pool can be expressed as a function of the total number of idle servers.

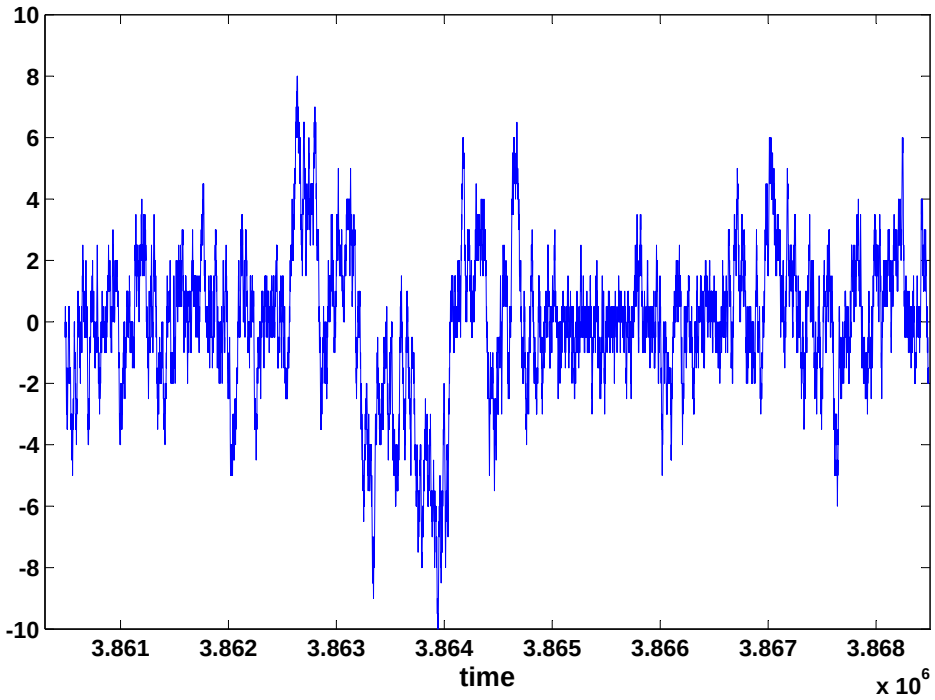
Denote $\mathcal{I}_i^\lambda$ to be the stationary number of idle servers in pool $i, i = 1, 2$. Given that $\mathcal{I}^\lambda := \mathcal{I}_1^\lambda + \mathcal{I}_2^\lambda = \gamma\sqrt{\lambda}$, $\mathcal{I}_i^\lambda$ deviates from $a_i\gamma\sqrt{\lambda}$ by $\Xi\sqrt[4]{\lambda}$, where $\Xi \Rightarrow Norm(0, \gamma a_1 a_2)$,

$as\ \lambda \to \infty$. Hence $\frac{\mathcal{I}_1^\lambda}{\mathcal{I}^\lambda} \approx a_1,\ as\ \lambda \to \infty$, or:

$$\frac{\mathcal{I}_1^\lambda}{\mathcal{I}_2^\lambda} \approx \frac{a_1}{a_2}, \qquad as\ \lambda \to \infty. \tag{5.10}$$

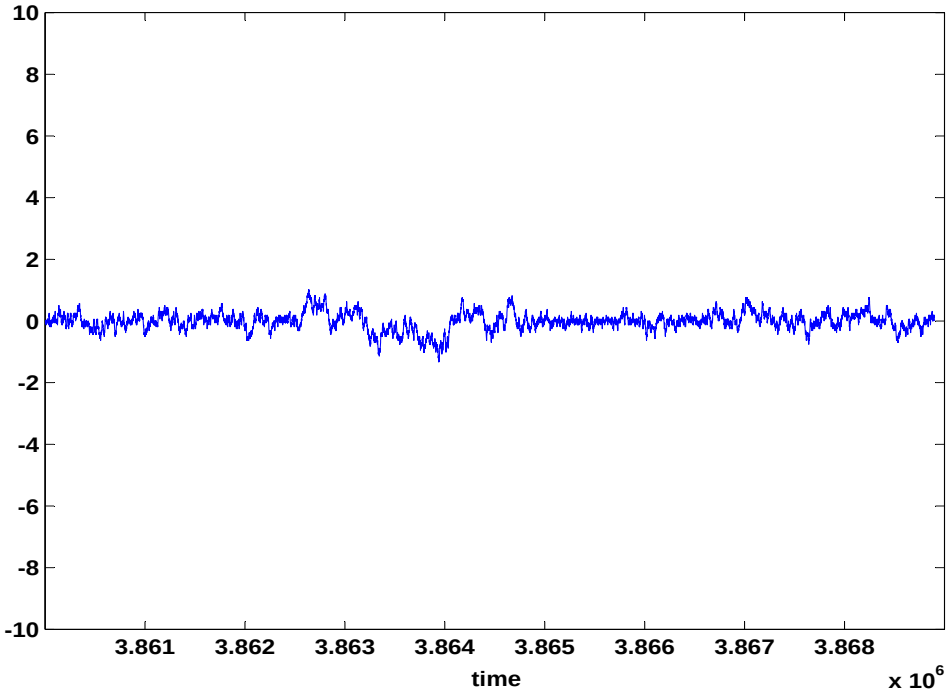
To illustrate this dimensionality reduction, let us consider the following example: we simulate the inverted-V loss system under RMI, with parameters that adhere to the QED scaling (C1) and (C2): $\lambda = 3950,\ \mu_1 = 15,\ \mu_2 = 7.5,\ N_1 = 138,\ N_2 = 276$ ($\delta = 3,\ a_1 = a_2 = 0.5$). In Figure 5.3, we draw a realization of the number of idle servers in pool 1 minus a_1 multiplied by the total number of idle servers in the system (i.e., $\mathcal{I}_1(t) - a_1\mathcal{I}(t)$). The realization is depicted a long time after the simulation started, thus the system is in steady-state. We see that the graph oscillates around 0, and its deviation from 0 is an order of magnitude of $\sqrt[4]{\lambda}$ (which in our case is 7.9). We thus observe a reduction in order of magnitude: while the total number of idle servers is an order of magnitude of $\sqrt{\lambda}$, the pictured variable is an order of magnitude of $\sqrt[4]{\lambda}$. Hence, when dividing $\mathcal{I}_1(t) - a_1\mathcal{I}(t)$ by $\sqrt[4]{\lambda}$ (see Figure 5.4), we anticipate dimensionality reduction.

Figure 5.3: $\mathcal{I}_1(t) - a_1\mathcal{I}(t)$



The dimensionality reduction result, that was demonstrated here, is not called state-space collapse (SSC), as the resulting one-dimensional process is not necessarily Markovian. The question whether it is Markovian is under investigation of Momcilovic [39].

Figure 5.4: $(\mathcal{I}_1(t) - a_1\mathcal{I}(t))/\sqrt[4]{\lambda}$



Remark 5.1.2 (Implication of the obtained idleness proportion.)
The fact that, for large λ, the *idleness proportion* of pool i (meaning, the proportion of idle servers in pool i) equals to a_i, adds an interesting insight on the RMI policy. Recall LISF (*Longest-Idle Server First*) routing, introduced by [7] for single-server pools systems and analyzed by [5] for general inverted-V systems (Section 3.1.2). If the LISF policy is applied to the $\wedge-$system, the diffusion limit found by [7] implies that the steady-state proportion of idle servers in pool i asymptotically (in the QED regime) equals $N_i\mu_i/\sum_{k=1}^{K} N_k\mu_k$ [5], namely, LISF and RMI asymptotically achieve the same idleness proportions. This adds more favorable "points" to the fairness of RMI routing, as LISF is commonly used in call centers and is considered to be fair [5]. In RMI, as in LISF, idleness is shared among the server pools proportionally to their service capacities and, in addition, proportionally to their effective service rates (as a_i is the proportion of customers served by pool i - see (5.8)). Additional discussion is provided in Remark 5.2.3.

5.1.3 $\mathbf{P}(W_q > 0)$, $\mathbf{E}W_q$, $\frac{EL_q}{\sqrt{N}}$ and $\frac{1}{\sqrt{\lambda}} f_{W_q|W_q>0}(\frac{t}{\sqrt{\lambda}})$ Approximation

We wish to obtain QED approximations for various performance measures in the *Delay system* with infinite queue (see the steady-state expressions in Section 4.6). Here we require $\delta > 0$ in Condition (C1) to ensure system stability. First, using the relation

between the loss and delay probabilities in (4.21), we find the delay probability limit, with the help of the $\sqrt{N}P^\lambda(block)$ approximation in (5.7):

$$\lim_{\lambda\to\infty} P^\lambda(W_q>0) \overset{(4.21)}{=} \lim_{\lambda\to\infty} \frac{P^\lambda(block)}{1-\rho^\lambda[1-P^\lambda(block)]} = \lim_{\lambda\to\infty} \left(\rho^\lambda + \frac{1-\rho^\lambda}{P^\lambda(block)}\right)^{-1} = \overset{(5.2),(5.9)}{=} \left(1+\frac{\beta}{\sqrt{N}P^\lambda(block)}\right)^{-1} \overset{(5.7)}{=} \left(1+\beta\sqrt{\frac{\mu}{\hat{\mu}}}\frac{\Phi(\delta/\sqrt{\hat{\mu}})}{\varphi(\delta/\sqrt{\hat{\mu}})}\right)^{-1}. \quad (5.11)$$

Next, with the help of the delay probability approximation, we find the approximation of EW_q (the exact formula is given in (4.24)):

$$\lim_{\lambda\to\infty} \sqrt{N}EW_q \overset{(4.24)}{=} \lim_{\lambda\to\infty} \frac{\sqrt{N^\lambda}}{N_1^\lambda\mu_1+N_2^\lambda\mu_2-\lambda}P(W_q>0) \overset{(C1),(5.1)}{=} \frac{\lim_{\lambda\to\infty}P^\lambda(W_q>0)}{\delta\sqrt{\mu}} \overset{(5.11),(5.9)}{=}$$
$$= \left(\beta\mu\left(1+\beta\sqrt{\frac{\mu}{\hat{\mu}}}\frac{\Phi(\delta/\sqrt{\hat{\mu}})}{\varphi(\delta/\sqrt{\hat{\mu}})}\right)\right)^{-1}. \quad (5.12)$$

Similarly, we obtain the approximation of EL_q:

$$\lim_{\lambda\to\infty} \frac{EL_q}{\sqrt{N}} \overset{(4.23)}{=} \lim_{\lambda\to\infty} \frac{\rho^\lambda}{\sqrt{N}(1-\rho^\lambda)}P(W_q>0) \overset{(5.9),(5.11)}{=} \left(\beta\left(1+\beta\sqrt{\frac{\mu}{\hat{\mu}}}\frac{\Phi(\delta/\sqrt{\hat{\mu}})}{\varphi(\delta/\sqrt{\hat{\mu}})}\right)\right)^{-1}. \quad (5.13)$$

Finally, we find the approximation of $\frac{1}{\sqrt{\lambda}}f_{W_q|W_q>0}(\frac{t}{\sqrt{\lambda}})$:

$$\lim_{\lambda\to\infty} \frac{1}{\sqrt{\lambda}}f_{W_q|W_q>0}\left(\frac{t}{\sqrt{\lambda}}\right) \overset{(4.25)}{=} \frac{N_1^\lambda\mu_1+N_2^\lambda\mu_2-\lambda}{\sqrt{\lambda}}e^{-\frac{N_1^\lambda\mu_1+N_2^\lambda\mu_2-\lambda}{\sqrt{\lambda}}t} \overset{(C1)}{=} \delta e^{-\delta t} \quad (5.14)$$

Note: in the case of equal service rates $\mu_1=\mu_2$, the above reduces to Erlang-C QED approximations.

Remark 5.1.3 (Comparison to the homogeneous server system.)
Following Conjecture 4.7.3, we would like to show that the homogeneous servers system, where the service rate of each server is μ, outperforms the $\wedge$−system under RMI. We conjectured the following for steady-state systems, but were unable to prove it; however, the proof in the QED regime turns out simple. In order to prove that the delay probability in (5.11) is larger than the delay probability of the corresponding Erlang-C model, it is enough to show that $\sqrt{\frac{\hat{\mu}}{\mu}}\frac{\varphi(\delta/\sqrt{\hat{\mu}})}{\Phi(\delta/\sqrt{\hat{\mu}})} > \frac{\varphi(\delta/\sqrt{\mu})}{\Phi(\delta/\sqrt{\mu})}$ (namely, the loss-probabilities order). Note that $\hat{\mu}>\mu$ (for heterogeneous pools), as $\hat{\mu}$ is the arithmetic average and μ is the harmonic average of the service rates of the customers (as discussed in Subsection 5.1.1). Then $\sqrt{\frac{\hat{\mu}}{\mu}}>1$, $\varphi(\delta/\sqrt{\hat{\mu}})>\varphi(\delta/\sqrt{\mu})$ and $\Phi(\delta/\sqrt{\hat{\mu}})<\Phi(\delta/\sqrt{\mu})$ (the latter two inequalities follow from the form of the standard normal density and distribution functions, as $\delta>0$).

5.2 Non-Random Routing

In the previous chapter and in the current one, we saw that the RMI routing policy enjoys some desirable properties: it forms a reversible Markov chain, thus can be analyzed both in steady-state (for any general B&D queue) and asymptotically. In addition, RMI appears *fair* towards servers: this follows both from its definition (routing to one of the idle servers at random) and from our analysis: exact (see Section 4.4) and asymptotic (see Remark 5.1.2). However, proposing such a routing for a hospital environment seems unnatural, due to its randomness; this inspires us to search for a non-random analogue. In this section we examine *non-random* routing policies which are equivalent in some ways to RMI, or even outperform RMI with regards to some operational or fairness criterion. Our analysis is for the loss model with two heterogeneous server pools in the QED regime.

5.2.1 Introducing Weighted Most-Idle (WMI) Routing

The naive non-random equivalent to RMI is **MI** - *Most-Idle* policy, which routes an arriving customer to the most vacant pool (the one with maximal number of idle servers). MI *appears* analogous to the RMI policy, as both regard the proportion of idle servers in each pool out of total number of idle servers in the system: RMI routes to each pool with probability equal to this proportion and MI routes to the pool with the maximal proportion. The MI policy aims to balance the number of idle servers in both pools: asymptotically (in the QED regime), as $N \to \infty$, $\mathcal{I}_1 \approx \mathcal{I}_2$ (this *dimensionality reduction* follows from [25], as will be explained in Remark 5.2.1). However, dimensionality reduction in RMI suggests: $a_2\mathcal{I}_1 \approx a_1\mathcal{I}_2$, as follows from (5.10). Thus, unless $a_1 = a_2$, RMI and MI clearly lead to different dynamics.

Hence we propose the natural extension of MI - **WMI** (*Weighted Most-Idle*) routing policy, which aims to balance the *weighted number of idle servers* among the pools, i.e. routes an arriving customer to the pool where the number of idle servers multiplied by the pool's weight is maximal. Formally, let us introduce a weight vector (w_1, w_2), $w_i \in (0, 1)$, $w_1 + w_2 = 1$. A customer arriving at time t, is routed to pool $i = argmax\{w_1\mathcal{I}_1(t), w_2\mathcal{I}_2(t)\}$ (or arbitrarily, if $w_1\mathcal{I}_1(t) = w_2\mathcal{I}_2(t)$ - see Remark 5.2.2). Hence, asymptotically ($as\ N \to \infty$),

$$w_1\mathcal{I}_1 \approx w_2\mathcal{I}_2. \tag{5.15}$$

(As with MI, this dimensionality reduction follows from [25] - see Remark 5.2.1).

We distinguish three interesting cases for weights in the WMI policy:

- $w_1 = w_2 = 1/2$: here asymptotically $\mathcal{I}_1 \approx \mathcal{I}_2$, and this is the *MI* policy.
- $w_1 = a_2, w_2 = a_1$: here $a_2\mathcal{I}_1 \approx a_1\mathcal{I}_2$, which is exactly what happens in RMI. Thus, we denote this policy as *NERMI - Non-random Equivalent to RMI.*
- $w_1 = q_2, w_2 = q_1$: here we route an arriving customer to pool $i = argmax\{q_2\mathcal{I}_1(t), q_1\mathcal{I}_2(t)\}$, which is equivalent to routing to the pool with maximal proportion of idle servers out of total number of servers in this pool, or routing to the least utilized pool (pool with the minimal occupancy rate). We denote this policy *OB* (*Occupancy-Balancing*).

Remark 5.2.1 (Dimensionality reduction for WMI.)
As described in Section 3.1.2, Gurvich and Whitt [25] address parallel-server systems (systems with multiple server pools and multiple customer classes) and propose *Queue-and-Idleness-Ratio (QIR)* routing rules that specify (1) what customer class to serve for a server, (2) what pool to join for a customer. When translated to the inverted-V settings, QIR rules are reduced to (2) - what pool to join for an arriving customer. Then, for the case of two server pools (for more than two pools, see Remark 5.2.4), QIR is similar to our WMI policy in the following sense. QIR stipulates sending to the server pool that maximizes $\mathcal{I}_j(t) - v_j(\mathcal{I}_1(t) + \mathcal{I}_2(t)),\ j = 1, 2$, where the v_j's are the idleness ratios ($v_1 + v_2 = 1$). If the v_j's are positive, the routing rule reduces to sending to the server pool with the highest value of $\mathcal{I}_j(t)/v_j$. By setting $v_j = 1/w_j,\ j = 1, 2$, and normalizing the w_j's such that $w_1 + w_2 = 1$, we obtain exactly WMI. Gurvich and Whitt [25] prove that, in the QED regime, $\mathcal{I}_j(t) - v_j(\mathcal{I}_1(t) + \mathcal{I}_2(t))$ converges to 0, when $\lambda \to \infty,\ j = 1, 2$, or, equivalently, $\mathcal{I}_1/v_1 \approx \mathcal{I}_2/v_2$. This clearly implies (5.15) for WMI.

Remark 5.2.2 (The case $w_1\mathcal{I}_1(t) = w_2\mathcal{I}_2(t)$.)
A question that arises, when introducing WMI, is what pool to choose, if at the time t of some customer's arrival holds $w_1\mathcal{I}_1(t) = w_2\mathcal{I}_2(t)$. Intuitively, one might route this customer to each one of the pools at random, or with some alternative probability that accounts for pool sizes or service capacities. We conjecture (and verify in simulations) that, although the policy of choosing a pool in such a case might be crucial for systems in light traffic and/or with small number of servers, in the QED regime with heavy traffic and large number of servers in both pools it plays no role. This is because (1) when the number of servers is large, the difference of ± 1 busy server is insignificant; (2) in heavy traffic, arrivals to the system are dense: if $w_1\mathcal{I}_1(t) = w_2\mathcal{I}_2(t)$ and we choose, for example, pool 1, then the next customer arriving at time t', will encounter $w_1\mathcal{I}_1(t') < w_2\mathcal{I}_2(t')$ and will be routed to pool 2 - the balance will be preserved. Indeed, simulations of WMI in the QED regime show that, even if one routes always to pool 1 or always to pool 2 in such a case, the difference in performance is negligible.

5.2.2 WMI versus RMI

We wish to analyze WMI routing from the aspects of fairness towards servers and operational performance, and compare it to RMI. The analysis is in the QED regime, based on the dimensionality reduction for RMI (5.10), due to [39], and on the dimensionality reduction for WMI, proved by [25] (see Remark 5.2.1). Our discussion is supported by computer simulations. When analyzing fairness towards servers, we address the following two already mentioned criteria: *occupancy balancing* and *flux balancing*. Server's utilization or, equivalently, pool's occupancy rate, is one of the prevalent measures of servers' workload. As the occupancy rates in both pools tend to 1 in the QED regime, we compare the ratio between the *proportions of idle servers* in the pools, referred to as the *Idleness-Ratio* $\mathbb{IR} = \dfrac{\bar{\mathcal{I}}_1/\mathbf{N_1}}{\bar{\mathcal{I}}_2/\mathbf{N_2}} = \dfrac{\mathbf{1}-\bar{\rho}_1}{\mathbf{1}-\bar{\rho}_2}$ (where $\bar{\mathcal{I}}_i$ stands for the average number of idle servers in pool i). The closer the ratio is to 1 the more balanced the routing is, according to this criterion.

In Section 2.3.2 we saw that, when addressing fairness towards servers, one should account for additional measure of workload - the average "flux" through the pools - namely, the number of customers served by a server per time unit. Hence, our second criterion is the *Flux-Ratio* $\mathbb{FR} = \dfrac{\gamma_1}{\gamma_2} = \dfrac{\bar{\rho}_1\mu_1}{\bar{\rho}_2\mu_2}$. The closer the ratio is to 1 the more balanced the routing is, according to this criterion. Concerning systems performance: when analyzing loss-systems, we consider the *loss probability* - $P(block)$ (for delay-systems we would have chosen $P(W_q > 0)$ or EL_q as our criterion). Clearly, a smaller loss probability implies better operational performance. Below we compare RMI and WMI routings with respect to the above three criteria. The summary of the comparison is presented in Table 5.1. During the comparison, we denote the policy in question ("R" for RMI and "W" for WMI) as a superscript of the measure.

Idleness-Ratio Criterion

For RMI routing, from the dimensionality reduction result (5.10) follows that:

$$\mathbb{IR}^R = \frac{1-\bar{\rho}_1^R}{1-\bar{\rho}_2^R} = \frac{\mathcal{I}_1/N_1}{\mathcal{I}_2/N_2} \overset{(5.10)}{\approx} \frac{N_2 a_1}{N_1 a_2} \overset{(5.3)}{\approx} \frac{q_2 a_1}{q_1 a_2} = \frac{\mu_1}{\mu_2}, \tag{5.16}$$

where the last equality follows from the fact that $a_i = q_i\mu_i/\mu$ $(i = 1, 2)$, which in turn follows from (5.3). We see that, asymptotically, the idleness-ratio of RMI depends only on the service rates. As we assume $\mu_1 > \mu_2$, it always holds that $\mathbb{IR}^R > 1$ (clearly, this

supports our finding in Theorem 4.4.1: the faster pool has lower occupancy rate than the slower pool, namely, $\bar{\rho}_1^R < \bar{\rho}_2^R$).

Similarly for WMI, from (5.15) follows:

$$\mathbb{IR}^W = \frac{1-\bar{\rho}_1^W}{1-\bar{\rho}_2^W} = \frac{\mathcal{I}_1/N_1}{\mathcal{I}_2/N_2} \overset{(5.15)}{\approx} \frac{w_2 N_2}{w_1 N_1} \overset{(5.3)}{\approx} \frac{w_2 q_2}{w_1 q_1}, \tag{5.17}$$

i.e., the idleness-ratio depends on the weights and pool capacities. We see that $\mathbb{IR}^W$ can be smaller, larger or equal to 1, depending on the ratio between $w_i q_i$. We examine these three conditions separately and deduce which policy is more balanced in each case, according to the idleness-ratio criterion (namely, where the idleness-ratio is closer to 1):

- $w_1 q_1 = w_2 q_2$: here $\mathbb{IR}^W = 1$ (and $\bar{\rho}_1^W = \bar{\rho}_2^W$: occupancy rates are balanced - this special case is the OB policy). Clearly, idleness-ratio of WMI is the most balanced in this case.
- $w_1 q_1 > w_2 q_2$: here $\mathbb{IR}^W < 1$. Hence we compare $\frac{w_1 q_1}{w_2 q_2}$ with $\frac{\mu_1}{\mu_2}$ and conclude that if $\frac{w_1 q_1}{w_2 q_2} > \frac{\mu_1}{\mu_2}$, RMI is more balanced; if $\frac{w_1 q_1}{w_2 q_2} = \frac{\mu_1}{\mu_2}$, they are equally balanced; and otherwise WMI is more balanced.
- $w_1 q_1 < w_2 q_2$: here $\mathbb{IR}^W > 1$. In this case, we compare $\frac{w_2 q_2}{w_1 q_1}$ with $\frac{\mu_1}{\mu_2}$ and conclude that if $w_1 a_1 < w_2 a_2$, RMI is more balanced, if $w_1 a_1 = w_2 a_2$ they are equally balanced, and otherwise WMI is more balanced.

Flux-ratio and P(block) criteria

Next we attend to the flux-ratio criterion. From Theorem 4.4.1 follows that, for RMI, the slower pool has lower flux than the faster one, namely $\mathbb{FR}^R = \frac{\gamma_1^R}{\gamma_2^R} > 1$. We analyze the flux-ratio of WMI and compare it to the flux-ratio of RMI in each one of the cases identified previously:

- $w_1 q_1 = w_2 q_2$: here $\mathbb{FR}^W = \frac{\bar{\rho}_1^W \mu_1}{\bar{\rho}_2^W \mu_2} = \frac{\mu_1}{\mu_2}$, as $\bar{\rho}_1^W = \bar{\rho}_2^W$. However, $\mathbb{FR}^R = \frac{\bar{\rho}_1^R \mu_1}{\bar{\rho}_2^R \mu_2} < \frac{\mu_1}{\mu_2}$ (as $\bar{\rho}_1^R < \bar{\rho}_2^R$) - RMI produces better balancing.
- $w_1 q_1 > w_2 q_2$: For both policies we have $\frac{\gamma_1}{\gamma_2} = \frac{\mu_1 \bar{\rho}_1}{\mu_2 \bar{\rho}_2} > 1$, while $\frac{\bar{\rho}_1^R}{\bar{\rho}_2^R} < 1$ and $\frac{\bar{\rho}_1^W}{\bar{\rho}_2^W} > 1$. Hence the RMI flux-ratio is smaller (and thus closer to 1) - again RMI produces better balancing.

- $w_1q_1 < w_2q_2$: This case is more complex, as we do not know whether $\mathbb{FR}^W$ is smaller or larger than 1 (as $\mathbb{IR}^W > 1$). However, in the QED regime, $\frac{\bar{\rho}_1}{\bar{\rho}_2}$ tends to 1, as $\lambda \to \infty$, while $\frac{\mu_1}{\mu_2}$ remains fixed; hence we conjecture that $\mathbb{FR}^W = \frac{\mu_1\bar{\rho}_1^W}{\mu_2\bar{\rho}_2^W} > 1$. As $\mathbb{FR}^R > 1$, the policy which has the smallest ratio $\frac{\bar{\rho}_1}{\bar{\rho}_2}$ is more balanced here. We can compare $\frac{\bar{\rho}_1}{\bar{\rho}_2}$ with the help of the idleness-ratio comparison, hence we examine separately the cases according to the idleness-ratio comparison:

  - $w_1a_1 < w_2a_2$: we saw that for the idleness-ratio criterion, RMI is more balanced, hence $\frac{1-\bar{\rho}_1^R}{1-\bar{\rho}_2^R} < \frac{1-\bar{\rho}_1^W}{1-\bar{\rho}_2^W}$. Multiplying numerators by denominators we get: $1+\bar{\rho}_1^R\bar{\rho}_2^W-\bar{\rho}_1^R-\bar{\rho}_2^W < 1+\bar{\rho}_2^R\bar{\rho}_1^W-\bar{\rho}_2^R-\bar{\rho}_1^W$. As in the limit $\bar{\rho}_1^R \approx \bar{\rho}_2^R$ and $\bar{\rho}_1^W \approx \bar{\rho}_2^W$, we obtain $\frac{\bar{\rho}_1^R}{\bar{\rho}_2^R} < \frac{\bar{\rho}_1^W}{\bar{\rho}_2^W}$ $\Rightarrow$ WMI is more balanced.
  - $w_1a_1 = w_2a_2$: here $\frac{1-\bar{\rho}_1^R}{1-\bar{\rho}_2^R} = \frac{1-\bar{\rho}_1^W}{1-\bar{\rho}_2^W}$. Hence, similarly as before: $\frac{\bar{\rho}_1^R}{\bar{\rho}_2^R} \approx \frac{\bar{\rho}_1^W}{\bar{\rho}_2^W}$ $\Rightarrow$ the policies are approximately equally balanced.
  - $w_1a_1 > w_2a_2$: here $\frac{1-\bar{\rho}_1^R}{1-\bar{\rho}_2^R} > \frac{1-\bar{\rho}_1^W}{1-\bar{\rho}_2^W}$. Hence, similarly as before: $\frac{\bar{\rho}_1^R}{\bar{\rho}_2^R} > \frac{\bar{\rho}_1^W}{\bar{\rho}_2^W}$ $\Rightarrow$ RMI routing is more balanced.

Finally, let us attend to the $P(block)$ criterion. Intuitively, the more customers are routed to the faster pool the smaller $P(block)$ gets (recall that for optimal system performance, one should route to the slower pool only when all fast servers are busy (FSF) - [3]). Thus, we conjecture that the higher the flux-ratio, the smaller the loss probability. Therefore, when comparing two policies, the policy with the least balanced flux-ratio (namely, the highest flux-ratio, as the flux ratio always exceeds 1) has smaller $P(block)$.

Note, that the flux-ratio and loss probability comparison is based mostly on heuristics (although verified by simulations) and does not follow from a formal proof. In Table 5.1 below, we summarize which policy performs better for each measure in different cases, where the closer the ratios are to 1 is better, and the smaller $P(block)$ is better. We conclude that, for different sets of parameters and different target functions, a different policy is superior. One can choose weights for WMI policy in a way that will optimize one of the criteria (or some integrated criterion).

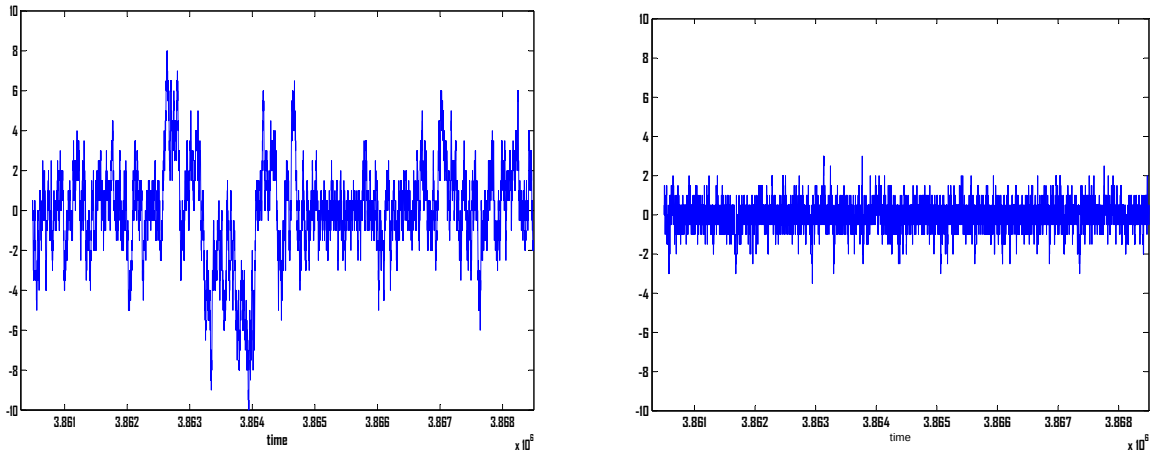
Table 5.1: Comparison: **RMI** vs. **WMI**

		Idleness-Ratio	Flux-ratio	P(block)
$w_1q_1 = w_2q_2$		WMI	RMI	WMI
$w_1q_1 > w_2q_2$	$\frac{\mu_1}{\mu_2} < \frac{w_1q_1}{w_2q_2}$	RMI	RMI	WMI
	$\frac{\mu_1}{\mu_2} = \frac{w_1q_1}{w_2q_2}$	equal		
	$\frac{\mu_1}{\mu_2} > \frac{w_1q_1}{w_2q_2}$	WMI		
$w_1q_1 < w_2q_2$	$w_1a_1 < w_2a_2$	RMI	WMI	RMI
	$w_1a_1 = w_2a_2$	equal	equal	equal
	$w_1a_1 > w_2a_2$	WMI	RMI	WMI

NERMI versus RMI

In Table 5.1 we see that, in the case of $w_1a_1 = w_2a_2$, the average performance of the both policies is equal. This is exactly the case of NERMI - Non-random Equivalent to RMI. However, the policies are not exactly the same, as one is randomized and the other - not. Their difference is shown in Figure 5.5: on the left we see a graph, already familiar to us from Section 5.1.2; it depicts the dimensionality reduction for RMI - a realization of the number of idle servers in pool 1 minus a_1 multiplied by the total number of idle servers in the system, namely, $\mathcal{I}_1(t) - a_1\mathcal{I}(t)$, which is an order of magnitude of $\sqrt[4]{\lambda}$. On the right we see the same graph, but for the NERMI policy (for the same system parameters): here, the magnitude order is 1. Although the policies clearly differ, we conjecture that they are the same on the diffusion scale.

Figure 5.5: $\mathcal{I}_1(t) - a_1\mathcal{I}(t)$: RMI vs. NERMI



MI policy

MI is a special case of WMI with $w_1 = w_2$, hence its idleness-ratio is obtained from (5.17): $\mathbb{IR}^{MI} = \frac{1-\bar{\rho}_1^{MI}}{1-\bar{\rho}_2^{MI}} \approx \frac{q_2}{q_1}$. We see that it depends only on pool capacities, and the larger pool has higher occupancy rate. This policy seems unfair towards the larger pool's servers, especially if the differences in capacities are significant. However, for relatively close pool capacities, higher occupancy rate for the larger pool does not necessarily imply higher load on the staff, as usually the size provides certain advantage in sharing the workload (due to economies of scale). One can compare MI to RMI using Table 5.1, by substituting $w_1 = w_2$. We believe that, in general, RMI makes more sense than MI, as it accounts for service rates, while MI accounts only for pool capacities.

Remark 5.2.3 (Connection to LWISF.)

In Remark 5.1.2 we noted the connection between RMI and LISF which, due to previous discussion, implies a relation between NERMI and LISF. A similar connection exists also for extensions of those policies: WMI and LWISF - the policy studied in [5]. LWISF (longest-weighted-idle server first) routes to the pool where the longest idleness time (time since latest service completion of the idle servers in this pool), multiplied by a weight related to this pool, is the largest. Formally, if i_k represents the idle time of the server in pool k that has been idle the longest, and w'_k is the weight of pool k ($w'_1 + w'_2 = 1$), the LWISF policy routes the customer to the pool $k = argmax\{w'_1 i_1, w'_2 i_2\}$ (assuming not all the servers are busy) [5]. Clearly, when the weights are adjusted appropriately, namely, $w_1 = \frac{a_2/w'_2}{a_1/w'_1 + a_2/w'_2}$ and $w_2 = \frac{a_1/w'_1}{a_1/w'_1 + a_2/w'_2}$, LWISF and WMI will lead to the same average results.

The obvious difference between the two policies is that WMI is based on the number of idle servers and LWISF - on the time that passed from each server's most recent service completion. However, Armony, Gurvich and Ward [6] are currently in the process of establishing that LWISF is asymptotically equivalent to QIR for $\wedge$-systems, and thus to WMI. This equivalence is important and insightful. While LWISF addresses individual idleness and thus is appropriate for systems with human servers, like call centers, it does not make much sense in the case when servers are beds. It could suite our hospital scenario if one addresses nurses or doctors as servers (instead of beds), but even then the information required by LWISF at the time of routing decision, namely time that passed since each server last service completion, is usually unavailable at the hospital. The policies' equivalence will imply that, asymptotically, it does not matter which policy to use, and one can choose the more suitable routing for each specific case. In addition,

an asymptotic equivalence of NERMI and LISF will imply an asymptotic equivalence of RMI and LISF.

Remark 5.2.4 (More than two servers pools.)
This chapter described asymptotic analysis for only two server pools. But what happens in the case of an arbitrary number of pools? We conjecture that for RMI the same dimensionality reduction result holds: the idleness proportion of pool i: $\frac{\mathcal{I}_i}{\sum_{j=1}^{K}\mathcal{I}_j}$ equals a_i where $a_i = \lim_{\lambda\to\infty}\frac{N_i^{\lambda}\mu_i}{\lambda}$, $\quad i = 1, 2, \ldots, K$ $(\sum_{j=1}^{K} a_j = 1)$. Then it is plausible that RMI will be asymptotically equivalent to QIR for $v_i = a_i$, $\forall i = 1, 2, \ldots, K$, and to WMI for the weights chosen such as $v_1 w_1 = v_2 w_2 = \ldots = v_K w_K$ (for these weights WMI is similar to QIR). These and other routing policies are analyzed for the case of four pools/wards with the help of computer simulations in [52] - see the summary in Chapter 8. Theoretical asymptotic analysis for the general number of pools presents an interesting direction for further research (see Chapter 9).

Chapter 6

Distributed Finite Queues System

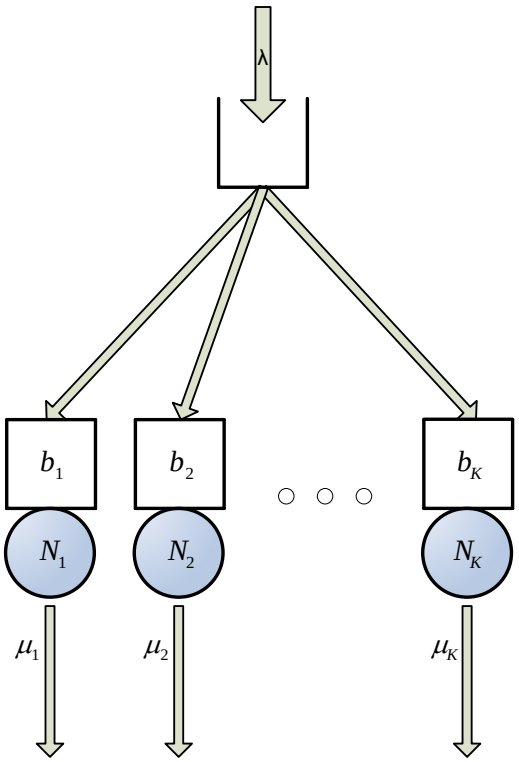
In this chapter, we study an additional type of queueing systems - a *distributed parallel server system* (as introduced in Section 3.1.3). In this system, each pool has its own dedicated queue (an alternative term is *buffer*), as opposed to the single centralized queue of the inverted-V system. The literature, surveyed in Section 3.1.3, addresses such systems with infinite queues (see Figure 3.3); here we analyze a slightly different setting. We assume that each pool has a queue with a finite capacity and, when all the queues are full, customers join a single centralized queue, modeled as a general Markovian Birth-and-Death process (the system is depicted in Figure 6.1). Thus, the system can be again viewed as a "kite", similar to the inverted-V system in Section 4.2: the "body" of the kite is a "loss-model" (in a sense that customers, arriving when all the pool queues are full, are blocked) and its "tail" is a centralized queue - the B&D process (the tail can be "empty"); see the transition rate diagram of this system, modeled as a Markov process, in Figure 6.2. We call such a system a *Distributed Finite Queues (DFQ)* System. DFQ presents a *generalization* of the inverted-V system: if the distributed queues have null capacity, the latter reduces to the former (the $\wedge-$system is a special case of DFQ).

One can think of several possible applications of DFQ systems. In our ED-to-IW process, after a patient is assigned to some IW, s/he waits in the ED for transfer to this ward - namely, in the dedicated "queue" of this ward (see Section 2.3.2). Clearly, as the number of beds in the ED is finite, the queue of each ward has finite capacity. The possible use of a centralized queue in the ED is for patients, that wait for their *assignment* to the IWs; but we believe that the "loss-model" is more suitable for modeling the ED-to-IW process (see further discussion on what model captures the ED-to-IW process better, in Section 6.4). One might also encounter DFQ systems in the general service sector: consider a crowded post office with several agents working in parallel and a limited space:

customers wait outside the office in a single queue and, after entering the office - join one of several queues. Another example can be Passport Queues when entering the U.S., which also operate as a DFQ system (with a human router into the separate queues).

The formal definition of the DFQ system is as follows: customers arrive according to a Poisson process with rate λ. There are K server pools: pool i has N_i i.i.d. exponential servers with service rates $\mu_i,\ i = 1, 2, \ldots, K$ $(\mu_i \neq \mu_j, \forall i \neq j)$. The total number of servers in the system is $\sum_{i=1}^{K} N_i = N$. Pool i has a waiting line with finite capacity b_i; the *total queue capacity* is $\sum_{i=1}^{K} b_i = b$. Denote the *total capacity of pool* i (number of servers plus queue capacity) by $C_i = N_i + b_i,\ i = 1, 2, \ldots, K$. Then $C = \sum_{i=1}^{K} C_i = N + b$ is the *total system capacity*. If a customer arrives when there are C other customers in the system, s/he joins a single centralized queue of the form of a general Markovian B&D process: possible structures of this queue are discussed in Section 4.2. The system is depicted in Figure 6.1.

Figure 6.1: Distributed Finite Queues (DFQ) System



6.1 RMI Routing Policy for Distributed Finite-Queues System

We propose the following routing policy for the DFQ system. As before, denote by $\mathcal{I}_i(t)$ the number of idle servers in pool i at time t. In addition denote by $\mathcal{E}_i(t)$ the number of *empty places* in the buffer of pool i at time t (we assume work conservation, namely: if $\mathcal{I}_i(t) > 0$ then $\mathcal{E}_i(t) = b_i$, or equivalently if $\mathcal{E}_i(t) < b_i$ then $\mathcal{I}_i(t) = 0$). By $V_i(t) = \mathcal{I}_i(t) + \mathcal{E}_i(t)$ we denote the total number of *vacant places* (idle servers + empty places in queue) in pool i at time t. A customer arriving at time t is routed to pool i with probability $\frac{V_i(t)}{\sum_{j=1}^{K} V_j(t)}$ (unless $V_i(t) = 0, \forall i = 1, \ldots, K$, in which case the customer leaves, or joins the centralized queue - depending on the structure of the system's "tail").

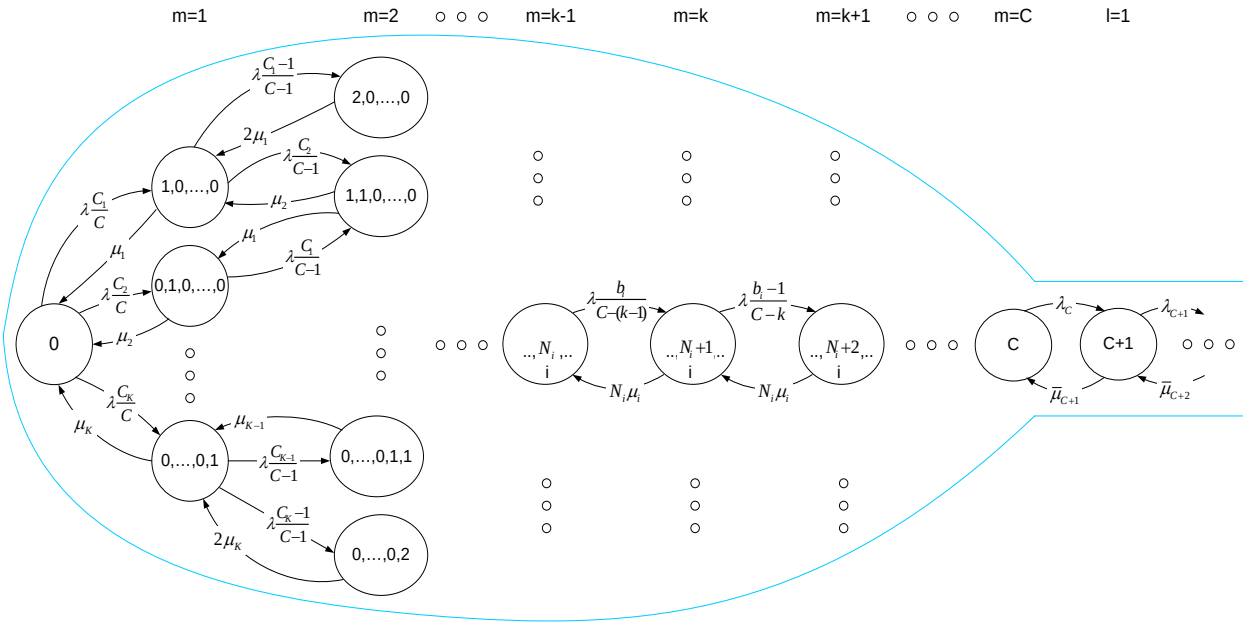
We refer to this routing policy as **RMI** (*Randomized Most-Idle*), as it resembles RMI routing in the $\wedge-$system (introduced in Section 4.1): in both systems (inverted-V and DFQ), routing is based on the proportion of *idleness* of each pool: in the $\wedge-$system, this idleness is represented by the number of *idle servers*; in the DFQ system - by the number of *vacant places*. An additional important similarity is that the DFQ system, like the inverted-V system, under the RMI policy, forms a *reversible* Markov Jump Process (and here we also conjecture that this is the only routing policy under which the DFQ system forms a reversible Markov Jump Process - see Conjecture 6.3.1). Indeed, if we take $b_i = 0$ ($C_i = N_i$), $\forall i \in \{1, 2, \ldots, K\}$, DFQ under RMI reduces to the inverted-V system under RMI. On the other hand, if $b_i \to \infty$, $\forall i \in \{1, 2, \ldots, K\}$ ($N_i << C_i$), the limit is the distributed system with infinite queues (depicted in Figure 3.3), where customers are routed to one of the pools at random (as the number of vacant places tends to infinity, thus is equal in all pools).

As the DFQ system under RMI is reversible, we can calculate the stationary distribution of the system. We do this similarly to Section 4.2: first, we analyze the *loss-model* (maximal number of customers in the system is C) - we prove that it is reversible and obtain its stationary distribution; then we connect the loss-model with the *queue* - a general Markovian B&D process (depicted in Figure 4.2; its stationary distribution is given in (4.4)). Similarly to the $\wedge-$system in Section 4.2, this concatenation of two reversible processes forms a reversible process, and we obtain its stationary distribution from the stationary distributions of the component processes.

We model the DFQ system under RMI routing as a Markov chain in continuous time $Z = \{Z_t,\ t \geq 0\}$. We characterize each state, with the centralized queue being empty, as a K-dimensional vector $z = (z_1, z_2, \ldots, z_K)$, where z_i is the total number of *customers*

in pool i - both in service and in queue ($z_i \in \{0, 1, \ldots, C_i\}$). Define $m_z = \sum_{i=1}^{K} z_i$ - the total number of customers at state z. The state $(C_1, C_2, ..., C_K)$, where all the distributed queues are full, is denoted by (C). States in which the number of customers in the system exceeds C by l customers, are denoted by $(C + l)$ (l is then the length of the centralized queue). The corresponding transition rate diagram is shown in Figure 6.2.

Figure 6.2: Distributed Finite Queues System under RMI as a Markov Chain



In Appendix E, we present the detailed balance equations, as in (4.1), for our loss-system (the "body" of the kite); we simultaneously verify reversibility of the process and obtain its stationary distribution - see (E.3). Next we connect the loss-system with the queue, modeled by a general Markovian Birth and Death process: each state $C + l$ (l customers in queue) has birth rate λ_l and death rate $\bar{\mu}_l$, and the maximal number of customers in the centralized queue is $c > 0$ (c may be equal to infinity) - see the transition rate diagram in Figure 4.2. As in Section 4.2, the two processes are connected via state (C); the resulting process is reversible and its stationary distribution is calculated from the individual processes stationary distributions, namely from (E.3) and (4.4):

$$\pi_z = \begin{cases} \pi_0 \prod_{i=1}^{K} \left(\frac{C_i!}{(C_i - z_i)!} \frac{N_i^{-(z_i - N_i)^+}}{(N_i \wedge z_i)! \cdot \mu_i^{z_i}} \right) \frac{\lambda^{m_z} (C - m_z)!}{C!} & , \quad l = 0; \\ \pi_0 \prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-b_i}}{N_i! \cdot \mu_i^{C_i}} \right) \frac{\lambda^C}{C!} \cdot \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} & , \quad l > 0. \end{cases} \tag{6.1}$$

Here π_0 is deduced from $\sum \pi_z = 1$ and equals to:

$$\pi_0 = \left[\sum_{z_1=0}^{C_1} \cdots \sum_{z_K=0}^{C_K} \left(\prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-(z_i - N_i)^+}}{(C_i - z_i)!(N_i \wedge z_i)! \mu_i^{z_i}} \right) \frac{\lambda^{m_z}(C - m_z)!}{C!} \right) + \prod_{i=1}^{K} \left(\frac{C_i! N_i^{-b_i}}{N_i! \mu_i^{C_i}} \right) \frac{\lambda^C}{C!} \sum_{l=1}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i} \right]^{-1}. \tag{6.2}$$

Note: By substituting $b_i = 0$ ($C_i = N_i$), we obtain the stationary distribution of the $\wedge-$system under RMI (as in (4.6)). On the other side, by taking $b_i \to \infty$, $\forall i \in \{1, 2, \ldots, K\}$ (but keeping N_i fixed), we converge to the stationary distribution of the distributed system with infinite queues, where customers are routed to one of the K pools at random (with probability $1/K$).

6.2 Performance Measures in Steady-State

In this section, we calculate several steady-state performance measures for the DFQ model under RMI routing. We analyze the system with a general queue structure, as presented in the previous section: its stationary distribution is given by (6.1). The measures, concerned with the centralized queue, are obtained similarly to Section 4.6. Denote the stationary *centralized queue length* (number of customers in the centralized queue in steady-state) by L_q, and the stationary *waiting time in the centralized queue* by W_q. First, we find $P(W_q > 0)$ - the steady-state probability that all distributed queues are full or, equivalently by PASTA, the steady-state probability that a customer arriving to the system is delayed in the centralized queue:

$$P(W_q > 0) = \sum_{l=0}^{c} \pi_{C+l} \overset{(6.1)}{=} \pi_0 \prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-b_i}}{N_i! \cdot \mu_i^{C_i}} \right) \frac{\lambda^C}{C!} \sum_{l=0}^{c} \frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}, \tag{6.3}$$

where π_0 is given in (6.2). Then we find the queue-length distribution:

$$P(L_q = l | W_q > 0) = \frac{\pi_{C+l}}{\sum_{i=0}^{c} \pi_{C+i}} \overset{(6.1)}{=} \frac{\frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}}{\sum_{k=0}^{c} \frac{\prod_{i=0}^{k-1} \lambda_i}{\prod_{k=1}^{l} \bar{\mu}_i}}, \qquad \forall l = 0, 1, \ldots, c; \tag{6.4}$$

$$P(L_q = l) = P(L_q = l | W_q > 0) P(W_q > 0) \overset{(6.4)}{=} \frac{\frac{\prod_{i=0}^{l-1} \lambda_i}{\prod_{i=1}^{l} \bar{\mu}_i}}{\sum_{k=0}^{c} \frac{\prod_{i=0}^{k-1} \lambda_i}{\prod_{i=1}^{k} \bar{\mu}_i}} P(W_q > 0), \quad \forall l = 1, 2, \ldots, c.$$

Expressions for special cases of the queue processes, such as *loss-models* and *delay-models with infinite queue*, can be found similarly to Section 4.6. For example, consider a

loss-model (no centralized queue), which stationary distribution is given in (E.3). We wish to find the steady-state *loss probability*, namely the probability that the system is in state (C), where all the distributed queues are full. By PASTA, this equals the probability that an arriving customer is blocked. We get the following expression for the loss probability:

$$P(block) = \pi_C \overset{(E.3)}{=} \pi_0 \prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-b_i}}{N_i! \cdot \mu_i^{C_i}} \right) \frac{\lambda^C}{C!} \overset{(E.4)}{=}$$

$$= \frac{\prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-b_i}}{N_i! \cdot \mu_i^{C_i}} \right) \frac{\lambda^C}{C!}}{\sum_{z_1=0}^{C_1} \cdots \sum_{z_K=0}^{C_K} \left(\prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-(z_i - N_i)^+}}{(C_i - z_i)!(N_i \wedge z_i)! \mu_i^{z_i}} \right) \frac{\lambda^{m_z} (C - m_z)!}{C!} \right)}. \tag{6.5}$$

Additional interesting performance measures relate to *delay in the distributed queues*. Denote the stationary *waiting time in the distributed queues* by W_p, and the stationary *total waiting time* in the system by W_t. The *probability of delay in the system* (in the distributed queues and/or the centralized queue) can be found in a following way:

$$P(W_t > 0) = P(W_p > 0, W_q = 0) + P(W_q > 0),$$

where $P(W_q > 0)$ is given in (6.3). By PASTA, $P(W_p > 0, W_q = 0)$ is the probability that a customer arrives to the system, when there are some vacant places, and there exists at least one pool with all servers busy, and the customer is routed to such a pool (with all servers busy). Formally, define the set of "busy" pools (pools with all servers busy) in state $z = (z_1, z_2, \ldots, z_K)$ by $\mathbb{B}(z) = \{i \in \{1, 2, \ldots, K\} |\ z_i \geq N_i\}$. In order to calculate $P(W_p > 0, W_q = 0)$, we go over all states z, with $\mathbb{B}(z) \neq \emptyset$, and sum up the probabilities that the system is in such a state in steady-state (namely, π_z), multiplied by the probability that a customer is routed to one of the states in $\mathbb{B}(z)$, i.e:

$$P(W_p > 0, W_q = 0) = \sum_{\substack{z_1 \ldots z_K: \\ \mathbb{B}(z) \neq \emptyset}} \pi_z \cdot \sum_{j \in \mathbb{B}(z)} \frac{C_j - z_j}{C - m_z}. \tag{6.6}$$

Here π_z are given in (6.1) and $m_z = \sum_{i=1}^{K} z_i$ is the total number of customers at state z.

One can also calculate the distributions of the stationary waiting time and queue length of pool i, and consequently - the distribution of the *total* waiting time and the total queue length (in the distributed queues plus a centralized queue).

6.3 Additional Issues

The RMI policy for the DFQ system suffers from the same main disadvantage as the RMI policy for the inverted-V system: its being *randomized*. In DFQ systems, this disadvantage is even amplified since the routing produced by RMI is not *work-conserving*, in a sense that a customer can be routed to a queue of some pool at the time when some other pool has idle servers. The problem can be attended to similarly as in the inverted-V system: by finding non-random policies analogous to RMI, like WMI in the $\wedge-$system (see Section 5.2). For DFQ systems, the WMI policy prescribes routing an arriving customer to the pool where the number of *vacant places*, multiplied by the pool's weight, is maximal. In order to find appropriate weights, for WMI to be equivalent to RMI, one must analyze DFQ systems under RMI in the *QED regime*. One can employ QED scaling, similar to the one used in the inverted-V model (see Section 5.1: in particular, Conditions (C1) and (C2)), with the additional conditions that the queue of each pool should be in the order of magnitude of *square-root of the pool's capacity*, namely: $b_i = \zeta_i\sqrt{N_i} + o(\sqrt{\lambda})$, *as* $\lambda \to \infty, \forall i \in \{1, 2, ..., K\}$ (as the scaling in [31]).

One can also extend the *steady-state analysis* of RMI in DFQ, and answer additional interesting questions, such as: comparing two servers, who works more: the faster or the slower, and whose productivity is higher? (as done for inverted-V systems under RMI, in Section 4.4). It will also be insightful to compare system's performance to a corresponding inverted-V system, and find under which conditions DFQ performs as well as an inverted-V system (or perhaps even outperforms it?) - see Subsection 6.3.1. A comparison to a corresponding homogeneous-servers system might be of interest as well. We believe that exploring these and other issues in DFQ systems presents a worthy direction for future research - see also Chapter 9. Finally, we expect that the following statement holds for DFQ systems (similar to Conjecture 4.7.1 for inverted-V systems):

Conjecture 6.3.1 *RMI is the only non-preemptive routing policy, under which the DFQ system forms a* ***reversible*** *Markov Jump Process.*

6.3.1 Distributed Systems vs. Inverted-V Systems

In Sections 2.3.2 and 3.2, we mentioned that a multi-queues system *appears to be less fair* towards waiting customers than a single-queue systems. In addition, the latter usually has an operational advantage over the former: in the case of work-conserving non-preemptive routing policies, an inverted-V system outperforms the corresponding

distributed system (for example, has smaller delay probability). This is because, in the distributed system, a customer might wait in a queue of some pool while, at the same time there is a server available in another pool; while in the inverted-V system, a customer never waits when there is an idle server (as we consider work-conserving policies). The following conjecture addresses distributed systems with unbounded queues, where routing is performed according to fixed probabilities, namely, immediately upon arrival to the system, a customer is routed to pool i with probability p_i, $0 < p_i < 1$, $\sum_{i=1}^{K} p_i = 1$.

Conjecture 6.3.2 *The average stationary total queue-length in the distributed system with probabilistic routing, considered above, exceeds the average stationary queue-length in the corresponding inverted-V system with unbounded queue, under the RMI policy.*

However, Tezcan [50] (and other researchers, mentioned in [50]) demonstrated that, *in the QED regime*, under *effective* routing policies, a distributed system performs as well as its corresponding inverted-V system (we surveyed [50] in Section 3.1.3). We conjecture that, comparing a distributed finite-queues system (DFQ) with a corresponding inverted-V system (with a centralized queue of maximal capacity b - equal to the total distributed queues' capacity in DFQ), the latter outperforms the former, when the minimal queue-length optimality criterion is considered. However, DFQ might perform asymptotically as well as the corresponding $\wedge-$system: this depends on the distribution of b among individual pools' queues, and on the routing policy (see also Remark 6.3.1 on the alternative optimality criterion).

Remark 6.3.1 (Minimal sojourn time performance criterion:)
Under the optimality criterion of minimal sojourn time in the system, DFQ might outperform the inverted-V system: Consider an extreme case, where the queue capacity of the fastest pool equals b (i.e., all other pools have no queues), and the routing policy sends customers to the fastest pool, as long as its queue is not full. We believe that the average sojourn time in such a system is shorter than in the corresponding inverted-V system under FSF, as the latter forms a sort of *threshold policy* - customers wait for busy fast servers even when there are available slow servers (see literature survey on the threshold policies on page 24 in Section 3.1.1). Clearly, this system is even less fair towards the fastest servers than the $\wedge-$system under FSF.

6.4 Application to the ED-to-IW Process

In Chapters 4, 5 and the present one, we analyzed various queueing models with heterogeneous server pools, when our original goal was to model the hospital ED-to-IW

process. The real-world process is quite complex (as we observed in Section 2.2.2 and in Figure 2.1), thus one cannot expect from a theoretical model to match it perfectly. However, we believe that the DFQ loss-model, presented in this chapter, captures the process quite well. In Section 2.3.2 we saw that, in the ED-to-IW process, we deal with a multi-queues system: a patient, assigned to a ward, waits in the ED for transfer to this specific ward - in a "queue" dedicated to this ward. In addition, the number of beds in the ED is finite, thus the queue of each ward must have finite capacity.

Clearly, the total queue capacity b is smaller than the number of beds in the Internal ED, but how is it distributed among the wards, namely, what are the b_i's? Determining the individual queue capacities presents an important question, as the b_i's influence the routing (for example, the dynamics of RMI routing clearly depends on them). A reasonable way is to take the b_i's proportionally to the pools' capacities (N_i's), or to the pools' *service* capacities ($N_i\mu_i$'s). Another method can be to use them as *incentives* for the wards: increase/decrease each ward's queue-capacity as a "reward"/"fine" for efficiency/inefficiency.

The main deficiency of the models discussed so far, as far as their fit to the real ED-to-IW process, is the fact that the servers in the models correspond to *beds*. However, in the real ED-to-IW process, the "servers" are also medical and nursing staff, stretcher-bearers, special equipment - unavailability of each resource causes patients' delays (see also Figure 2.2). This implies that a patient could wait even when there is an available bed in the designated ward - see Remark 6.4.1. (We attempted to model patients' waiting times in a computer simulation of the ED-to-IW process, created in [52] - see Section 8.1.) Thus, our theoretical systems model the delays caused *only* by the scarcity of beds, although most practical delays are caused by unavailability of nurses or doctors. A possible way to model the delays in each ward is with the help of an M/M/N (or GI/GI/N) model, where N is the number of nurses or doctors in this ward. An additional significant limitation of our theoretical systems - modeling a single customer/patient class instead of multiple classes - is discussed, as a future research idea, in Chapter 9.

Remark 6.4.1 (Operational regime.)
Finally, we discuss what operational regime suits the ED-to-IW process, with the help of delay probability. First, as servers in our models are beds, we estimate the probability of encountering an available bed upon hospitalization decision (relatively to the wards' standard capacities) from the empirical data (on period 1/05/06-30/10/08, excluding the months 1-3/07). We observe that proportion of patients, upon the assignment time of those there was an available bed in the designated ward, was 46%, 54%, 81%, 75%, for Wards A-D respectively. Thus, the *probability of encountering an available bed* is clearly

at QED levels. However, in Figure 2.3, we see that the probability to be admitted to the wards immediately (or within a short time) after the hospitalization decision is much smaller: thus, over the same period of time, only 2.7% of the patients were admitted to an IW within 15 minutes from their assignment to this ward. This reminds us more of the *Efficiency Driven* regime. We also note that, in evening shifts (when most of the patients are admitted - see [35]), there usually are one or two doctors in each ward; hence one expects that they operate under high loads. Then the *probability of having an available doctor* should be at Efficiency Driven levels (we do not have any empirical data on staff availability).

As a concluding remark, we note that although a $\wedge-$system is more fair towards waiting customers than DFQ and usually outperforms DFQ (Section 6.3.1), the ED-to-IW process operates under the latter system and not under the former. The reasons for maintaining distributed queues in the ED-to-IW process are listed in Section 2.3.2: they make sense indeed. However, we might "convince" the hospital to transfer to the centralized queue system, at least on a theoretical level, by showing what could be gained - such a setting is demonstrated in the next chapter.

Chapter 7

Game Theory

Hospitals provide numerous examples of conflicts of interests and thus present a propitious ground for *Game-Theoretic* research. One can easily find examples of "games" between various "players" in the hospital. For example, in our ED-to-IW process, there is a conflict of interests between a nurse in the ED, who wishes to transfer a patient to the IW as quickly as possible, and a nurse in the IW, who might wish to delay the transfer. A game-theoretic dilemma arises also among the wards: it might be beneficial for a ward to prolong patients' stay, while the other wards "cooperate" and strive for a quick discharge of patients (this way the non-cooperating ward receives less patients as it maintains high occupancy levels) - we discuss this further in Section 7.2.

In Section 7.1, we adopt a *costs sharing* game-theoretic approach (introduced in Section 3.3). We view the IWs as players, who could pool their resources to serve ED patients. As a coalition *cost* (*characteristic function*), we choose some optimality criterion that reflects the cost of congestion (e.g. steady-state mean number of customers in the queue), and this way define a cooperative game. We find a *fair* costs allocation with the *Shapley value* approach: the Shapley value of each player/ward reflects its relative "responsibility", with respect to the cost of congestion. Based on this approach, hospital administration could possibly make informed decisions on "fines" and "rewards", or even routing policies. Though our game is aimed at the hospital ED-to-IW process, it may well fit other service systems, where players are firms or service providers.

7.1 Allocating Delay Costs among IWs: the Shapley Value Approach

We define a *transferable utility cooperative game* $(\mathbb{K}, V)$, where $\mathbb{K} = \{1, 2, \ldots, K\}$ is a set of K players, and $V(\cdot)$ is a *characteristic function* of the game (as in [2]). Each player can choose to *cooperate* - a group of cooperating players is called *coalition* $\mathbf{S}$, $\emptyset \subset \mathbf{S} \subseteq \mathbb{K}$; $\mathbb{K}$ is called the *grand coalition*. The characteristic function $V : 2^K \to \mathbb{R}$ assigns to each nonempty coalition $\mathbf{S}$ a number $V(\mathbf{S})$, which represents the cost (or the profit) of this coalition; $V(\emptyset)$ is defined to be zero. Or goal is to allocate the cost of the grand coalition among the players in a *fair* manner. The *Shapley value* method for cost allocations is based on marginal costs in entering coalitions. It provides a "fair" allocation, in the sense that it is the only allocation with the following desirable properties: efficiency, additivity, symmetry and the dummy player property (see [51] for explanations of these properties). The Shapley value of player i in the game $(\mathbb{K}, V)$, $\varphi_i(V)$, is calculated through the following formula:

$$\varphi_i(V) = \sum_{\mathbf{S}: i \in \mathbf{S}} \frac{(|\mathbf{S}| - 1)!(K - |\mathbf{S}|)!}{K!} [V(\mathbf{S}) - V(\mathbf{S} \backslash i)]. \tag{7.1}$$

The formula can be justified by imagining the coalition being formed one player at a time, with each player demanding their contribution $V(\mathbf{S}) - V(\mathbf{S}\backslash i)$ as a fair compensation, and then averaging over the possible different permutations in which the coalition can be formed.

7.1.1 Game Definition

We define a cooperative game in our queueing system, which represents the ED-to-IW process. The players are pools (wards); as before, pool i contains N_i statistically identical servers (beds), each with exponential service time at rate μ_i $(i = 1, 2, \ldots, K)$. We assume that customers arrive to pool i according to a Poisson process with rate $\lambda_i > 0$, and each pool has an unbounded queue, i.e., pool i operates like an $M/M/N_i$ model $(i = 1, 2, \ldots, K)$. The whole system can be animated as a *distributed system* with infinite queues (see Figure 3.3), where routing occurs according to fixed (static) probabilities: the probability that an arriving customer is routed to pool i is $p_i = \dfrac{\lambda_i}{\sum_{i=1}^K \lambda_i}$ (clearly, $\sum_{i=1}^K p_i = 1$). A group of cooperating pools (*coalition*) $\mathbf{S}$, $\emptyset \subset \mathbf{S} \subseteq \mathbb{K}$, forms an *inverted-V system* (with a single centralized queue), with incoming stream of customers $\lambda_{\mathbf{S}} = \sum_{i \in \mathbf{S}} \lambda_i$, $|\mathbf{S}|$ pools and total number of servers $N_{\mathbf{S}} = \sum_{i \in \mathbf{S}} N_i$. For $\mathbf{S} = \mathbb{K}$ (grand coalition), we denote $\lambda = \sum_{i=1}^K \lambda_i$, and the total number of servers in the systems is

$N = \sum_{i=1}^{K} N_i$ (we omit the subscript for the grand coalition). We assume that $\lambda_i < N_i\mu_i,\ \forall i = 1,\ldots,K$, in order to guarantee stability of each $M/M/N_i$ system. These conditions imply $\rho_{\mathbf{S}} := \dfrac{\lambda_{\mathbf{S}}}{\sum_{i\in\mathbf{S}} N_i\mu_i} < 1,\ \forall \mathbf{S} \subseteq \mathbb{K}$, i.e., system stability for every possible coalition.

Define the characteristic function of the game $V(\cdot)$ as the **expected number of customers in queue in steady-state**. Denote the steady state mean queue length of pool i in the distributed system by $E(L_q^i)$; thus $V(\{i\}) = E(L_q^i)$ (which corresponds to the case of no-coalition). The characteristic function of coalition $\mathbf{S}$ is then $V(\mathbf{S}) = E(L_q^{\mathbf{S}})$, where $E(L_q^{\mathbf{S}})$ is the average total queue length in the system, formed by coalition $\mathbf{S}$; $V(\emptyset)$ is naturally defined as zero. Each coalition operates as a $\wedge$-system, under the *RMI* routing policy (introduced in Section 4.1): upon arrival, a customer is routed to one of the available pools, with probability that equals the proportion of idle servers in this pool, out of the overall number of idle servers in the system, or joins the centralized queue if all the servers in all the pools are busy.

The players' cooperation transfers the *distributed system*, where routing is done according to fixed probabilities, into an *inverted-V system* under RMI. In the previous chapter (Conjecture 6.3.2), we stated that the total mean queue length in the latter system is smaller than in the former system. Then, cooperation of all the pools (grand coalition) is optimal for the system, as it minimizes the total "cost" - mean queue length. An anticipated disadvantage of the present setting is that it does not account for the fact that, by joining a coalition, pool i may start serving less or more customers than λ_i (number of customers served by pool i in the distributed system). This can certainly affect the players' motivation to cooperate, and should be accounted for when calculating their costs. We return to this issue later on.

We seek the relative responsibility of each pool for delay in the system, namely allocate the average queue length of the grand coalition among the pools. In Section 4.6, we found the steady-state average queue length in the inverted-V system under RMI - see (4.23); using this formula, for each coalition $\mathbf{S}$, $\emptyset \subset \mathbf{S} \subseteq \mathbb{K}$, we obtain the coalition's characteristic function:

$$V(\mathbf{S}) = E(L_q^{\mathbf{S}}) = \frac{\rho_{\mathbf{S}}}{1-\rho_{\mathbf{S}}} P(W_q^{\mathbf{S}} > 0) \overset{(4.19)}{=} \pi_0^{\mathbf{S}} \frac{(\lambda_{\mathbf{S}})^{N_{\mathbf{S}}}}{N_{\mathbf{S}}! \prod_{i\in\mathbf{S}} {\mu_i}^{N_i}} \left(\frac{\rho_{\mathbf{S}}}{(1-\rho_{\mathbf{S}})^2} \right). \tag{7.2}$$

Here $\pi_0^{\mathbf{S}}$, the steady-state probability that the system, formed by coalition $\mathbf{S}$, is empty, is adopted from (4.20) and equals to:

$$\pi_0^{\mathbf{S}} = \left[\sum_{\substack{y_i=0 \\ i\in\mathbf{S}}}^{N_i} \left(\prod_{i\in\mathbf{S}} \binom{N_i}{y_i} \right) \frac{(N_{\mathbf{S}} - m_y)!}{N_{\mathbf{S}}!} \frac{(\lambda_{\mathbf{S}})^{m_y}}{\prod_{i\in\mathbf{S}} \mu_i^{y_i}} \left(\frac{1}{1-\rho_{\mathbf{S}}} \right)^{(m_y - N_{\mathbf{S}}+1)^+} \right]^{-1}. \tag{7.3}$$

(Recall that y_i is the number of busy servers in pool i ($y_i \in \{0, 1, \ldots, N_i\}$), and $m_y = \sum_{i\in\mathbf{S}} y_i$ - the total number of busy servers.)

We allocate the grand coalition costs among the pools, by finding a Shapley value of pool i, φ_i. We interpret φ_i as the part of the total queue length that pool i is "responsible" for. Note that φ_i can be negative, if pool i's joining the coalition has decreased the queue length dramatically, or, conversely, exceed $E(L_q)$, if pool i's joining the coalition has increased the queue length dramatically. In any case, we have $\sum_{i=1}^{K} \varphi_i = E(L_q)$, due to the efficiency property. The calculation of the Shapley value is based on (7.1): one substitutes the characteristic function expression for each coalition $\mathbf{S}$, as given in (7.2). The resulting expression turns out highly cumbersome and complex, even for a small number of players/pools. Calculations can be easily performed, though, for specific examples - such an example, for our hospital ED-to-IW case, is presented in the next subsection. More complex models could be accommodated in simulations which will generate values of the characteristic function. Clearly, our analysis and its conclusions are preliminary and *ad hoc*, and the idea requires and deserves further research. Alternatively, one can analyze the game in the QED asymptotic regime, with the help of approximations, which yields far simpler expressions; we discuss this further in Subsection 7.1.3.

7.1.2 Numerical Example

Below we present an example on how the Shapley game can fit our actual ED-to-IW process. Our game then has $K = 4$ players - server pools $1-4$, corresponding to IWs A-D. We estimate the required parameters from empirical data (from Tables 2.1 and 2.2): the arrival rate to pool i is estimated by the average number of patients hospitalized in Ward i per day; the number of servers in pool i is the *average* number of beds in Ward i (taken as the average between the ward's standard and maximal capacity), and individual service rates are 1/ALOS. The resulting system parameters are presented in Table 7.1.

First, for each possible coalition, we calculate through (7.2) its characteristic function - the mean stationary queue-length in the system that the coalition forms. The results are presented in Table 7.2 ("1st Example"): we observe that the average queue-length in the $\wedge-$system, formed by the grand coalition, is 0.298, which is much smaller than individual queue-lengths (and also smaller than other coalitions' queue-lengths).Next, we obtain the

Shapley values of the pools, through (7.1) - see Table 7.1. We see that the values are *negative* for Pools 2 and 3 (Wards B and C) and *positive* for Pools 1 and 4 (Wards A and D). This suggests that Pools 1 and 4 carry the main responsibility for congestion: this makes sense, as those are the slowest pools. Pool 2 has the highest service rate, thus it "helps" to reduce the queue-length and φ_2 is negative. Pool 3 has even smaller Shapley value, as it had lower *flux* in the original distributed system (arrival rate divided by the number of servers: $\lambda_3/N_3 < \lambda_2/N_2$), although $\mu_3 < \mu_2$. In general, φ_i's are determined by the combination of the pools' three parameters: λ_i's, N_i's and μ_i's. Higher service rates clearly lead to smaller responsibility for congestion and thus smaller Shapley values; and so does low flux in the original distributed system (λ_i/N_i).

Table 7.1: Hospital Data Example

	Pool 1	Pool 2	Pool 3	Pool 4
λ_i (per day)	6.85	6.25	7.00	6.95
N_i	49	33	45	43
μ_i (per day)	0.1570	0.2235	0.1866	0.1798
$E(L_q^i)$	2.664	1.503	0.848	3.512
φ_i	0.369	-0.184	-0.624	0.737
$\tilde{\lambda}_i$	6.737	6.315	7.262	6.743

In addition, we calculate the *actual throughput* of pool i (namely, the average number of customers served by pool i), in the system formed by the grand coalition; we denote it by $\tilde{\lambda}_i$: $\tilde{\lambda}_i = N_i\gamma_i$, where γ_i is the flux of pool i in the grand coalition ($i = 1, 2, 3, 4$). We compare $\tilde{\lambda}_i$'s with λ_i's - the number of customers served by pool i originally, in the distributed system. The last line of Table 7.1 shows that the number of served customers has increased for Pools 2 and 3, but grew smaller for Pools 1 and 4. Thus, the pools that are less (more) responsible for the queue-length now serve more (less) customers than prior to joining the coalition. Hence, in this game, the Shapley values do not tell the whole story, and are not sufficient when deciding on incentives for players: although some pools are "rewarded" for decreasing the queue length, they are made responsible for more customers, which in our ED-to-IW case may be considered as a "punishment" (higher flux may imply higher load on the staff, as we saw in Section 2.3.2). Alternatively, players could be interested in serving more customers: for example, if the players are commercial firms, more customers imply higher profits. We propose a change in the game, that will accommodate $\tilde{\lambda}_i$'s, in Remark 7.1.1.

Back to our example, in order to neutralize the difference between $\tilde{\lambda}_i$'s and λ_i's, we

assume that they are equal ($\lambda_i = \tilde{\lambda}_i,\ \forall i \in \{1,2,3,4\}$), namely the number of customers, served by each pool, is the same in the distributed system and in the $\wedge$-system. This way we attempt to "isolate" the influence of the queue length only. All other parameters (besides arrival rates) remain the same as previously (in Table 7.1). Characteristic functions of each coalition - the mean stationary queue-length in the queueing systems they form, are presented in Table 7.2 ("2nd Example"). Shapley values are calculated with (7.1), as before, and equal $(0.121,\ 0.055,\ -0.026,\ 0.148)$, for Pools $1-4$ respectively. We note that the dispersion of the values is much smaller than before, and still Pools 1, 4 have high values (high responsibility for the congestion) and Pools 2, 3 - low values. Thus, the cost allocation, obtained in the original setting, lead to reasonable *order* of responsibilities (which player is more responsible and which is less), although it did not account for a change in the number of customers served by each pool.

Table 7.2: Characteristic Function (Mean Stationary Queue-length) Values

Coalition **S**	$V(\mathbf{S})$ 1st Example	$V(\mathbf{S})$ 2nd Example
$\{1\}$	2.6642	1.9264
$\{2\}$	1.5029	1.7534
$\{3\}$	0.8478	1.6316
$\{4\}$	3.5120	1.9716
$\{1,2\}$	1.0367	0.9450
$\{1,3\}$	0.6588	0.8661
$\{1,4\}$	1.8202	0.9945
$\{2,3\}$	0.4841	0.8479
$\{2,4\}$	1.2548	0.9670
$\{3,4\}$	0.8132	0.8904
$\{1,2,3\}$	0.3569	0.4889
$\{1,2,4\}$	0.8088	0.5482
$\{1,3,4\}$	0.5523	0.5030
$\{2,3,4\}$	0.4255	0.5014
$\{1,2,3,4\}$	0.2976	0.2976

Remark 7.1.1 (Accounting for a change in the arrival rate to each pool.) As mentioned above, a disadvantage of the proposed game is that it does not account for the fact that a pool may start serving less/more customer once it joins the coalition

(although one can still gain insights from the game, one should treat these changes in arrival rates reasonably). One way to circumvent it is to assume that each player/pool has some *target cost function* that it wishes to minimize: in addition to queue costs, the function accounts for the number of served customers: $P_i = c_1\tilde{\lambda}_i + c_2\varphi_i$, where $\tilde{\lambda}_i$ is the actual throughput of pool i and φ_i is the part of the queue length that pool i is responsible for $(i = 1, 2, \ldots, K)$. If a pool is interested in serving more customers, c_1 is negative; otherwise both weights are positive. Each player then checks whether it is better off *in* or *out* of the coalition, namely, where its cost function is smaller. For example, in order to check whether the players prefer a distributed system or the grand coalition, we compare $c_1\lambda_i + c_2 E(L_q^i)$ with $c_1\tilde{\lambda}_i + c_2\varphi_i$, $\forall i \in \{1, 2, \ldots, K\}$. Considering our example and taking weights $c_1 = c_2 = 1/2$, we obtain the total costs *before cooperation* $(4.76, 3.88, 3.92, 5.23)$ and *after cooperation* $(3.52, 3.11, 3.33, 3.72)$. This means that all pools prefer the grand coalition, as they gain by reducing drastically their queue length. Clearly, the weights influence the results significantly; determining them is a challenging task for further research.

7.1.3 In the QED Regime

In Section 5.1.3, we found QED approximations for various performance measures in the inverted-V system under RMI, for the case of two pools: the approximation of $\frac{EL_q}{\sqrt{N}}$ is given in (5.13). Clearly, the expression in (5.13) is much simpler than the steady-state one (7.2), thus one can attempt to gain insights on pools' responsibilities for congestion, in the QED regime. We treat the case of $K = 2$ players (pools), as we have the average queue-length approximation for this case only. We find the characteristic functions of all coalitions (for two players, the possible coalitions are $\{1\}, \{2\}$ and $\{1, 2\}$). For the inverted-V model ($\mathbf{S} = \{1, 2\}$ - the grand coalition), we obtain:

$$V^N(\{1,2\}) = \lim_{\lambda\to\infty} \frac{EL_q}{\sqrt{N}} \overset{(5.13)}{=} \left(\beta\big(1 + \beta\sqrt{\frac{\mu}{\hat{\mu}}}\frac{\Phi(\delta/\sqrt{\hat{\mu}})}{\varphi(\delta/\sqrt{\hat{\mu}})}\big)\right)^{-1}. \tag{7.4}$$

For $M/M/N_i$ models ($\mathbf{S} = \{i\}$, $i = 1, 2$), we have:

$$V^N(\{i\}) = \lim_{\lambda\to\infty} \frac{EL_q^i}{\sqrt{N}} \overset{(5.3)}{=} \sqrt{q_i}\left(\beta_i\big(1 + \beta_i\frac{\Phi(\beta_i)}{\varphi(\beta_i)}\big)\right)^{-1}, \qquad i = 1, 2; \tag{7.5}$$

where $q_i = \lim_{\lambda\to\infty} \frac{N_i^\lambda}{N^\lambda}$ and $\beta_i = \lim_{\lambda\to\infty} \sqrt{N_i^\lambda}(1 - \rho_i^\lambda)$, $i = 1, 2$.

Using (7.1), we obtain: $\varphi_1 = \frac{1}{2}(V^N(\{1\}) + V^N(\{1,2\}) - V^N(\{2\})$ and $\varphi_2 = \frac{1}{2}(V^N(\{2\}) +$

$V^N(\{1,2\}) - V^N(\{1\})$ [1]. The Shapley values depend on difference between $V^N(\{1\}$ and $V^N(\{2\})$: the player with the smaller $V^N(\{i\})$ has a smaller φ_i. Thus, the Shapley values depend on q_i's - limiting fractions of pool i servers out of the total number of servers, and on β_i's - quality of service parameters. The pool with larger β_i has a smaller Shapley value, which makes sense since systems with higher level of service quality have smaller congestions. Thus, Shapley values are influenced by the size of the pool, its service rates and arrival rate, which is consistent with the numerical examples in Subsection 7.1.2.

7.2 Choosing the Service Rate: an Alternative Approach

In this section, we pursue a different game-theoretic approach. The players are still server pools, corresponding to the IWs, but the decision-makers are now *nurses* of the IWs. We aim to check whether they "prefer" to increase or decrease patients' discharge rate, where the conflict is between *occupancy-rate* and *flux*. If a ward prolongs patients' stay, it admits less patients (its flux is lower), but its occupancy rate is higher; and conversely, if it discharges patients faster, its occupancy rate is lower but the flux is higher. We consider the inverted-V system under RMI, where by Theorem 4.4.1, the faster pool does indeed have a lower occupancy rate but higher flux, when compared against the slower pool. A system designer wishes to provide players with the incentives to work faster (although serving customers too fast might harm service quality, as mentioned in Remark 4.4.3).

Consider two pools/wards (one can extend the game to more pools). Pool i has N_i servers, and it determines its service rate, μ_i, in a way that will minimize its *target load function* $f(\mu_i)$ $(i = 1, 2)$. *Load*, experienced by nurses in a ward, can be roughly divided into two parts: load implied by treating hospitalized patients, and load implied by patients' admissions and discharges (load definition is studied in [35]). Define t_{ini} - the average time required from a nurse for every admission/discharge, and t_{tr} - the average time of treatment required by every hospitalized patient per hour. Then the target function of pool i equals

$$f(\mu_i) = (2\gamma_i N_i t_{ini} + \bar{\rho}_i N_i t_{tr})/n_i, \qquad i = 1, 2 \tag{7.6}$$

where $\bar{\rho}_i$ is the mean steady-state occupancy rate in pool i, γ_i is the average flux through pool i (γ_i is multiplied by 2 in order to account for both admissions and discharges), and

[1] One can alternatively address $\lim_{\lambda\to\infty} V^N(\mathbb{S}) =: \hat{V}(\mathbb{S})$, $\mathbb{S} = \{\{1\}, \{2\}, \{1,2\}$, and calculate the Shapley values with those rescaled values $\hat{V}(\mathbb{S})$.

n_i is the number of nurses in ward i ($i = 1, 2$). By dividing the expression by n_i, we deduce that (7.6) is the average load per nurse in ward i (fraction of time the nurse is busy), $i = 1, 2$.

We construct a two-player *strategy game*, where we assume that each player/ward chooses its *strategy* - service rate, to be *high*, *medium* or *low*. Using the empirical data from Table 7.1, we assume that the *low* service rate is 0.16 (the rate of the slowest Pool 1) and the *fast* service rate is 0.22 (the rate of the fastest Pool 2); the *medium* rate is taken as the average between the two, namely 0.19. Our players are pools 1 and 2, corresponding to Wards A and B. The number of beds in the ward is, as in Table 7.1: $N_1 = 49,\ N_2 = 33$; and arrival rate to the system is the number of patients hospitalized in both wards: $\lambda = 13.1$. Ward A has, on average, 8 nurses per shift and Ward B has 6 nurses per shift (data is taken from [18]), namely $n_1 = 8,\ n_2 = 6$.

Finally, we must estimate t_{ini} and t_{tr}: t_{tr} is adopted from [54]: assuming 0.4 patients' requests per hour, multiplied by the average service time of each request (15 minutes), we get $t_{tr} = 0.01$ hours per patient per hour. We do not have any empirical data on t_{ini}: we estimate it as 0.2 hours, as we learnt from an IW nurse that admission of a patient does not take much longer than the regular treatment time. In Table 7.3, we present $(f(\mu_1), f(\mu_2))$ for every combinations of service speeds in both pools ($f(\mu_i)$'s are calculated by (7.6)).

Table 7.3: Strategy Game - the ED-to-IW Example

		Player II - Ward B		
		high	medium	low
Player I | Ward A	high	(0.8447, 0.7512)	(0.8857, 0.7582)	(0.9378, 0.7596)
	medium	(0.8733, 0.8046)	(0.9200, 0.8132)	(0.9806, 0.8148)
	low	(0.8999, 0.8804)	(0.9537, 0.8932)	(1.0255, 0.8974)

In order to identify the strategies (service rates) that are preferred by the players, we find the *Nash equilibrium*: a set of strategies, such that no player can do better by unilaterally changing its strategy. Namely, we search for a cell in Table 7.3, where the load is minimal for *both* players. We note that the value of such a cell is (0.8447, 0.7512) - namely, in our game, both players would choose to operate under high service rate. (Note that the setting where both players choose low rate is not even feasible, as then the first player has load higher than 1.)

There are several disadvantages of this game-theoretic approach. First, the equilibrium

depends on t_{ini} and t_{tr} - one must estimate these times empirically. In addition, we assume that the μ_i's do not depend on t_{ini} and t_{tr}, namely, that service rates are not influenced by the duration of clinical treatments, but are more administrational. Finally, in practice, nurses are not the ones to determine patients' discharge rate. Thus, the result only demonstrates what nurses would *prefer* under this setting but not what they can accomplish.

Chapter 8

Simulation Analysis

In previous chapters we modeled the process of patients routing from the ED to IWs as a queueing system, and analyzed it under various queue-architectures and routing policies. However, our theoretical models do not depict precisely the complex reality: we had to simplify the real-world process in order to be able to analyze it analytically. In addition, when choosing a routing policy, we had to restrict ourselves to those that can be theoretically tractable (for example, reversible policies). While analytical models do provide important practical insight, it is still useful to expand our modeling scope to accommodate some analytically intractable features, for example time-inhomogeneous Poisson arrivals, and routing that is based on partial information. And here *simulation* comes to the rescue, as a very powerful instrument for complex systems modeling and analysis.

In this chapter we present a summary of a joint project with A. Zviran, carried out within the framework of the "System Analysis and Design" Technion OR graduate course. (The full project report is available in [52].) The goal of the project was to create a computer simulation model of the ED-to-IW process in Anonymous Hospital, and with its help examine various routing policies, according to some fairness and performance criteria, while accounting for availability of information in the system.

A generic computer simulation, which modeled the ED-to-IW process described in Chapter 2 (and depicted in Figure 2.1 and Appendix A), was built in Matlab software. Simulation parameters were estimated from empirical quantitative data (obtained from the hospital database); its results were validated against the real data. We aimed to find out what routing algorithm might be "optimal": both fair and best-performing. Thus, we defined various fairness and performance measures to form a single *integrated*

criterion of quality; we proposed various routing policies, while accounting for availability of information in the system, and implemented them in the simulation. The algorithms were evaluated according to the optimality criteria and compared against each another. Below we highlight several interesting issues studied in the project, which have not been discussed yet in the thesis; we then summarize some related results.

8.1 Simulation Model

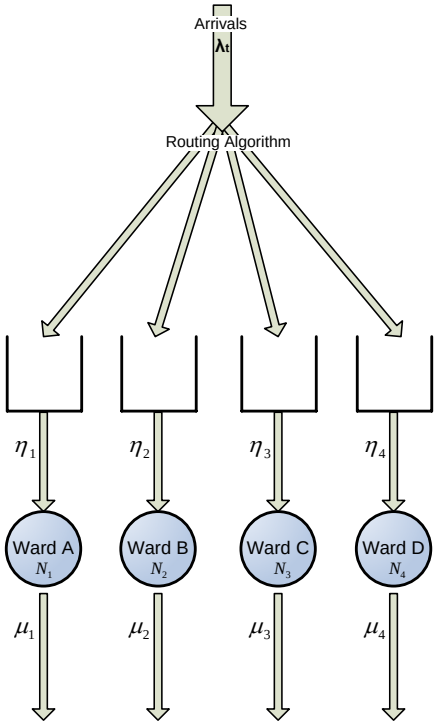
In general, the model underlying the simulation is similar to our theoretical model (described at the beginning of Chapter 4): four heterogeneous server pools represent the four IWs A-D, and each pool contains a number of i.i.d. servers corresponding to the number of beds in each ward. The main differences, namely the points where the simulation model is closer to reality than the theoretical model, are that arrivals to the system occur according to a *time-dependent* Poisson process and LOS in the wards are of a *log-normal* distribution (as opposed to homogeneous Poisson arrivals and exponential LOS, required for a Markovian system).

More importantly, we attempted to account for the fact that patients' waiting times in the ED are caused not only by lack of a bed in the designated ward (see Section 6.4), as in classical queueing systems. We thus divided the waiting times into two phases: delay when there is an available bed (a "boarding time" that ward staff requires for preparation), and waiting when all the beds are occupied. We modeled the boarding time of a ward as an exponentially distributed random variable with the mean depending linearly on the occupancy rate of this ward at the time of allocation (see detailed explanation in Section 2.1 of [52]). The simulation model is depicted in Figure 8.1. Its main limitation remains regarding a single customer/patient class (see Chapter 9). We also assumed that each pool has a dedicated queue of infinite capacity (*distributed system*), although the finite-queues model is more appropriate (see Section 6.4).

8.2 Quality Criteria

In order to evaluate different routing policies and compare them, we introduced various *fairness* and *performance* optimality criteria. Fairness criteria correspond to the discussion in Section 2.3.2: for wards' staff-fairness, we strived to balance the *occupancy rate* and the *flux* (number of patients per bed per time unit) of the wards. In order to measure how

Figure 8.1: Simulation Model



fair each policy is according to a certain criterion, we used the *standard deviation* of the parameter we wished to balance (see [48], that measured the fairness of airport schedules in a similar way). Thus, for the criterion of balanced occupancy rates, we strived to minimize the average standard deviation of wards' occupancy rates (standard deviation of occupancy rate at each moment tells us how large is the difference in occupancy rates among the wards), and for the criterion of balanced flux we strived to minimize the standard deviation of flux. In order to measure patients-fairness, we compared standard deviation of their waiting times (similar to [9]), produced by different policies. As the main operational performance measure, we considered the average delay time. The *integrated criterion of quality* was formed by aggregating those measures with appropriate weights (additional details can be found in Sections 4.1 and 4.2 of [52]).

8.3 Routing Algorithms

We analyzed several routing algorithms, some very intuitive and simple, and others more complex (for a full discussion, readers are referred to Sections 4.3 and 4.4 of [52]). We started with *Round Robin* algorithms - routing according to a certain cyclical order

(such algorithms are widely used in hospitals - see Section 2.4). In particular, the *Flux Balancing Algorithm* aimed to keep an equal number of patients per bed per time unit - it routed a patient to the ward that had not received patients for the longest time, while accounting for the wards' capacities so that the number of patients admitted by each ward would be proportional to its size. The algorithm indeed balanced the flux, but it lead to long waiting times. RMI and MI routing policies (familiar to us from Chapters 4 and 5, for the inverted-V models) produced better results in the simulation, but were not found optimal.

We proposed the *Occupancy Balancing Algorithm*, which aimed at balancing pools occupancies at each moment of routing. The algorithm worked in the following way: at the time of a customer arrival, one checked what would be the standard deviation of the occupancy rates among the pools if this customer would be allocated to a certain pool (out of all available pools). The customer was eventually allocated in the way that minimized the standard deviation of the occupancy rates after the allocation. The algorithm showed very good performance: low and balanced occupancy rates plus short waiting and sojourn times in the system. However, it allocated more patients to the fastest ward, which was unfair to this ward's staff. We proposed then the *Weighted Algorithm*, which combined the Occupancy Balancing and the Flux Balancing algorithms, by minimizing a convex combination of the two conflicting demands: balanced occupancy rates and balanced flux. The algorithm achieved both staff-fairness and good operational performance.

Another parameter, considered when proposing routing policies, was the availability of information in the system at the moment of the routing decision. We recognized three possibilities: *no information*, meaning that at the moment of routing decision only static parameters are known (number of beds, ALOS), and the system's state is unknown; *full information*, meaning that dynamic system parameters, such as occupancy rate, are available to us; and *partial information*, meaning that we know the system state only at some time points, and we estimate its state at the decision time, based on this information. In our Anonymous hospital, the ED relies on one update per day (in the morning) on the occupancy status in the IWs (thus the exact information on the number of available beds at the moment of routing is unavailable). We found that our algorithms' implementation under partial information, resulted in almost no deterioration of performance. We conclude that the Weighted algorithm achieves the best performance (in terms of fairness and minimal weighting time) and can be implemented in a partial-information environment, which is easier to design and implement in real-life hospital settings.

Chapter 9

Future Research

Facing the challenge of modeling a complex reality, one must compromise and allow relaxations and simplifying assumptions. Clearly, these present a limitation for the applications of our results. We address below some main items that we simplified or omitted, and which can be added and investigated within further research.

- **Extend theoretical and simulation analysis to several customer (patient) classes:** As mentioned in Chapter 2, patients to-be-hospitalized in the IWs are classified into several categories. When arriving to the ED, patients are classified as "walking" or "lying"; in addition, prior to running the Justice Table, they are classified as "regular", "special care" or "ventilated". The load, inflicted on the hospital by patients, varies significantly among different categories: in LOS, complexity of treatment and waiting times. In both theoretical and simulation analysis, we considered a single patient class - this simplifies the model, but presents a limitation as well.

- **Include Ward E in the theoretical study:**
  Ward E differs significantly from the other wards in several parameters: type of patients (only walking, independent patients), higher turnover rate (more admissions and discharges per day), more patients treated by a single nurse/physician. Thus, routing to Ward E is not done via the Justice Table, and we excluded it from our study. Nevertheless, it is interesting to examine more carefully fairness aspects concerning Ward E, especially as this ward operates under high loads - its average occupancy rate, relatively to standard capacity, is higher than 100%. This happens because the hospital strives to send to Ward E the maximal number of internal walking patients, and thus send them to Wards A-D only when Ward E is totally

full. Until recently, there were several beds in the Eye-Ward, which were designated for Ward E patients (now the ward has moved to a new location and its capacity increased). In view of this, incorporating Ward E in theoretical models can be interesting and insightful.

Next we discuss theoretical issues that can be further expanded.

- **Extend the QED analysis of $\wedge$-systems to an arbitrary number of server pools:** In Chapter 5, we analyzed the $\wedge$-system under RMI and WMI routings, in the QED asymptotic regime, for the case of *only two* server pools. In Remark 5.2.4, we speculated how the analysis would change for an arbitrary number of pools. Extending the QED study for more pools is important, in particular as there are more than two IWs in Anonymous hospital.

- **Analyze the RMI policy in distributed-finite-queues, in the QED regime:** In Chapter 6, we introduced the RMI policy in DFQ systems (each pool has its own dedicated finite-capacity queue) and analyzed it, quite superficially, in steady-state. In Section 6.3, we discussed how this analysis can be extended, in particular, in the QED regime. For example, the expressions of the steady-state performance measures (given in Section 6.2) are extremely complex; finding their QED approximations appears challenging and insightful. Extending the analysis of DFQ systems in general (not just under RMI routing), also seems as a worthy direction for further research.

- **Psychological study: waiting time versus sojourn time criterion:** As we saw in Chapter 4.5, optimizing the queue-length (waiting time) may differ from optimizing the number-in-system (sojourn time). As discussed in Remark 4.5.1, in different cases a different criterion is more appropriate. In particular, the following "behavioral studies" question arises: what does a customer perceive as more disturbing - waiting or being served by a slow server? "Rational" customers would prefer to wait, if that will minimize their total time in the system. This might be true in face-to-face queues, where customers can compare their sojourn time against their fellow-customers. However, in phone queues for example, getting served by a slow server might feel less disturbing than waiting. We believe this to be an interesting direction for further research - an example of the psychological cost of waiting was studied in [14].

- **Extend the Game-theoretic study:** In Chapter 7, we proposed several ways to approach the ED-to-IW process from a Game-Theoretic point of view. The ideas were analyzed *ad hoc* and should be pursued further.

Bibliography

[1] Alanyali M. and Hajek B., *Analysis of Simple Algorithms for Dynamic Load Balancing*, Mathematics of Operations Research **22** (1990), no. 4, 840–871. 3.1.2

[2] Anily S. and Haviv M., *Cooperation in Service Systems*, Manuscript under review. 3.3, 7.1

[3] Armony M., *Dynamic Routing in Large-Scale Service Systems with Heterogeneous Servers*, Queueing Systems **51** (2005), 287–329. 3.1.2, 3.1.2, 3.1.4, 4.2, 4.4, 4.5.1, 4.5.2, 5.1, 5.2.2

[4] Armony M. and Mandelbaum A., *Routing and Staffing in Large-Scale Service Systems: the Case of Homogeneous Impatient Customers and Heterogeneous Servers*, Manuscript under review. 3.1.2, 4.5.1, 4.5.3

[5] Armony M. and Ward A., *Fair Dynamic Routing Policies in Large-Scale Systems with Heterogeneous Servers*, Manuscript under review. 3.1.2, 3.1.4, 3.2, 4.1, 4.4, 5.1.2, 5.2.3

[6] Armony M., Gurvich I. and Ward A., *Private communication and work in process.* 5.2.3

[7] Atar R., *Central Limit Theorem for a Many-Server Queue with Random Service Rates*, The Annals of Applied Probability **18** (2008), no. 4, 1548–1568. 3.1.2, 3.1.4, 4.4, 5.1.2

[8] Atar R., Shaki Y.Y. and Shwartz A., *A Blind Policy for Equalizing Cumulative Idleness*, Working Paper. 3.1.2, 3.1.4, 4.1

[9] Avi-Itzhak B. and Levy H., *Measuring Fairness in Queues*, Advances in Applied Probability **36** (2004), 919–936. 3.2, 8.2

[10] Bekker R. and de Bruin A.M., *Time-dependent Analysis for Refused Admissions in Clinical Wards*, Submitted to AOR-ORAHS's special volume (2008). 2.5

[11] Ben-Zrihen M., Borsher J., Reiss A. and Tseytlin Y., *Behavioral Models in Customer Service Centers*, IE&M Undergraduate Project, Technion (2007). 3.2

[12] Cabral F.B., *The Slow Server Problem for Uninformed Customers*, Queueing Systems **50** (2005), no. 4, 353–370. 3.1.1, 4.3.1, 4.3.1, 4.4, 4.5

[13] ———, *Queues with Heterogeneous Servers and Uninformed Customers: who Works the Most?*, arXiv:0706.0560vl (2007). 3.1.1, 4.4

[14] Carmon Z., Shanthikumar J.G. and Carmon T.F., *A Psychological Perspective on Service Segmentation Models: The Significance of Accounting for Consumer's Perceptions of Waiting and Service*, Management Science **41** (1995), no. 11, 1806–1815. 9

[15] Cooper A.B., Litvak E., Long M.C. and McManus M.L., *Emergency Department Diversion: causes and Solutions*, Academic Emergency Medicine **8** (2001), 1108–1110. 1, 2.5

[16] de Bruin A.M., Bekker R., van Zanten L. and Koole G.M., *Dimensioning Hospital Wards Using the Erlang Loss Model*, Submitted to AOR-ORAHS's special volume (2008). 2.5

[17] Diwas K.C. and Terwiesch C., *The Impact of Work Load on Productivity and Quality: an Econometric Analysis of Hospital Operations*, Manuscript under review (2008). 2.5, 4.2

[18] Elkin K. and Rozenberg N., *Patients' Flow from the Emergency Department to the Internal Wards*, IE&M Undergraduate Project, Technion (2007). 2, 2.2.1, 2.3.1, 2.3.2, 2.3.2, 2.3.2, 2.3.2, 7.2

[19] Erlang A.K., *On the Rational Determination of the Number of Circuits*, in: The Life and Works of A.K. Erlang, eds E. Brockmeyer, H. L. Halstrom, A. Jensen, The Copenhagen Telephone Company, Copenhagen, Denmark, (1948). 3.1.4

[20] Fleurbaey M. and Maniquet F., *Cooperative Production with Unequal Skills: the Solidarity Approach to Compensation*, Social Choice Welfare **16** (1999), 569–583. 3.2

[21] Garcia-Sanz M.D., Fernandez F.R., Fiestras-Janeiro M.G., Garcia-Jurado I. and Puerto J., *Cooperation in Markovian Queueing Models*, European Journal of the Operational Research (2007). 3.3

[22] Gonzáles P. and Herrero C., *Optimal Sharing of Surgical Costs in the Presence of Queues*, Mathematical Methods of Operations Research **59** (2004), 435–446. 2.5, 3.3

[23] Green L.V., *Capacity Planning and Management in Hospitals*, Operations Research and Health Care (Brandwau et al editors) (2004), 14–41. 1, 2.5

[24] ———, *Using Operations Research to Reduce Delays for Healthcare*, Tutorials in Operations Research, Informs (2008). 2.1, 2.5

[25] Gurvich I. and Whitt W., *Queue-and-Idleness-Ratio Controls in Many-Server Service Systems*, Manuscript under review. 3.1.2, 3.1.4, 5.2.1, 5.2.1, 5.2.1, 5.2.2

[26] Halfin S. and Whitt W., *Heavy-traffic Limits for Queues with many Exponential Servers*, Operations research **29** (1981), 567–587. 3.1.4

[27] Huseman R.C., Hatfield J.D. and Miles E.W., *A New Perspective on Equity Theory: the Equity Sensitivity Construct*, The Academy of Management Review **12** (1987), no. 2, 222–234. 3.2

[28] Jagerman D.L., *Some properties of the Erlang loss function*, Bell Systems Technical Journal **53** (1974), no. 3, 525551. 5.1.1

[29] Jennings O.B. and Vericourt F.d., *Nurse-to-Patient Ratios in Hospital Staffing: a Queuing Perspective*, Working Paper, Duke University. 2.1, 2.5

[30] Kelly F.P., *Reversibility and Stochastic Networks*, Wiley, Chichester, 1979. 4.2, 4.2, 4.2.1

[31] Khudyakov P., *Designing a Call Center with an IVR (Interactive Voice Response)*, M.Sc. Thesis, IE&M, Technion (2006). 6.3

[32] Larsen R.L. and Agrawala A.K., *Control of a Heterogeneous Two-Server Exponential Queueing System*, IEEE Transactions on Software Engineering **July** (1983), pp. 522–526. 3.1.1, 4.4

[33] Larson R.C., *Perspectives on Queues: Social Justice and the Psychology of Queueing*, Operations Research **35** (1987), 895–905. 2.3.2, 3.2

[34] Lin W. and Kumar P.R., *Optimal Control of a Queueing System with Two Heterogeneous Servers*, IEEE Transactions on Automatic Control **28** (1984), no. 8, 696–703. 3.1.1, 4.4

[35] Mandelbaum A., Marmor Y., Tseytlin Y. and Yom-Tov G., *From the Emergency Department to Hospitalization: Using Theoretical Models, Empirical Models and Simulation for the Operational Analysis of the ED, IW, and their Interface*, Working Paper, Technion. 1, 2.3.2, 4, 5, 6.4.1, 7.2

[36] Maniquet F., *A Characterization of the Shapley Value in Queueing Problems*, Journal of Economic Theory **109** (2003), 90–103. 3.3

[37] Marmor Y. and Sinreich D., *Emergency Departments Operations: the Basis for Developing a Simulation Tool*, IIE Transactions **37** (2005), no. 3, 233–245. 2.5

[38] McManus M.L., Long M.C., Cooper A. and Litvak E., *Queuing Theory Accurately Models the Need for Critical Care Resources*, Anesthesiology **100** (2004), no. 5, 1271–1276. 2.5

[39] Momcilovic P., *Private communication.* 5, 5.1.1, 5.1.2, 5.1.2, 5.2.2

[40] New York Times, *1 in 3 Hospitals Say They Divert Ambulances.* 2.5

[41] Anonymous Hospital: Report of the Quality Promotion Team, *Reducing Waiting Times in the Emergency Department and LOS in the Internal Wards*, (1997). 2.2.1

[42] Rafaeli A., Barron G. and Haber K., *The Effects of Queue Structure on Attitudes*, Journal of Service Research **5** (2002), no. 2, 125–139. 2.3.2, 3.2

[43] Ramakrishnan M., Sier D. and Taylor P.G., *A Two-Time-Scale Model for Hospital Patient Flow*, IMA Journal of Management Mathematics **16** (2005), 197–215. 2.5

[44] Reinhardt G. and Dada M., *Allocating the Gains from Resource Pooling with the Shapley Value*, Journal of the Operational Research Society **56** (2005), 997–1000. 3.3

[45] Rubinovitch M., *The Slow Server Problem*, Journal of Applied Probability **22** (1983), 205–213. 3.1.1, 4.3.1, 4.4, 4.5

[46] ———, *The Slow Server Problem: a Queue with Stalling*, Journal of Applied Probability **22** (1985), 879–892. 3.1.1, 4.4

[47] Shapley L.S., *A Value for n-Person Games*, Annals of Mathematical Studies **28** (1953), 307–317. 3.3

[48] Soomer M.J. and Koole G.M., *Fairness in the Aircraft Landing Problem*, Working paper (2008). 3.3, 8.2

[49] Stockbridge R.H., *A Martingale Approach to the Slow Server Problem*, Journal of Applied Probability **28** (1991), 480–486. 3.1.1, 4.4

[50] Tezcan T., *Optimal Control of Distributed Parallel Server Systems under the Halfin and Whitt Regime*, Math of OR (2007). 3.1.2, 3.1.3, 3.1.4, 6.3.1

[51] Tijs S.H. and Driessen T.S.H., *Game Theory and Cost Allocation Problems*, Management Science **32** (1986), no. 8, 1015–1028. 3.3, 7.1

[52] Tseytlin Y. and Zviran A., *Simulation of Patients Routing from an Emergency Department to Internal Wards in Rambam Hospital*, OR Graduate Project, IE&M, Technion (2008). (available on http://iew3.technion.ac.il/serveng/ References/project_yulia_asaf.pdf). 1, 5.2.4, 6.4, 8, 8.1, 8.2, 8.3

[53] Wright J. and King R., *We All Fall Down: Goldratt's Theory of Constraints for Healthcare Systems*, North River Press, 2006. 2.5

[54] Yankovic N. and Green L.V., *A Queueing Model for Nurse Staffing*, Under review at Operations Research. 7.2

[55] ynet, *What Hospital Is the Most Crowded in Israel?*, http://www.ynet.co.il/articles/0,7340,L-3656123,00.html (2009). 2.4

[56] Yom-Tov G., *Queues in Hospitals: Semi-Open Queueing Networks in the QED Regime*, Ph.D. Proposal, IE&M, Technion (2007). 2.1

Appendix A

Flow Charts of the ED-to-IW Process: Activities, Resources, Information

In Section 2.2.2 we described the ED-to-IW routing process and depicted it in the Integrated (Activities - Resources) Flow Chart (Figure 2.1). Below we present additional charts, that highlight different aspects of the process.

Figure A.1: Activities Flow Chart

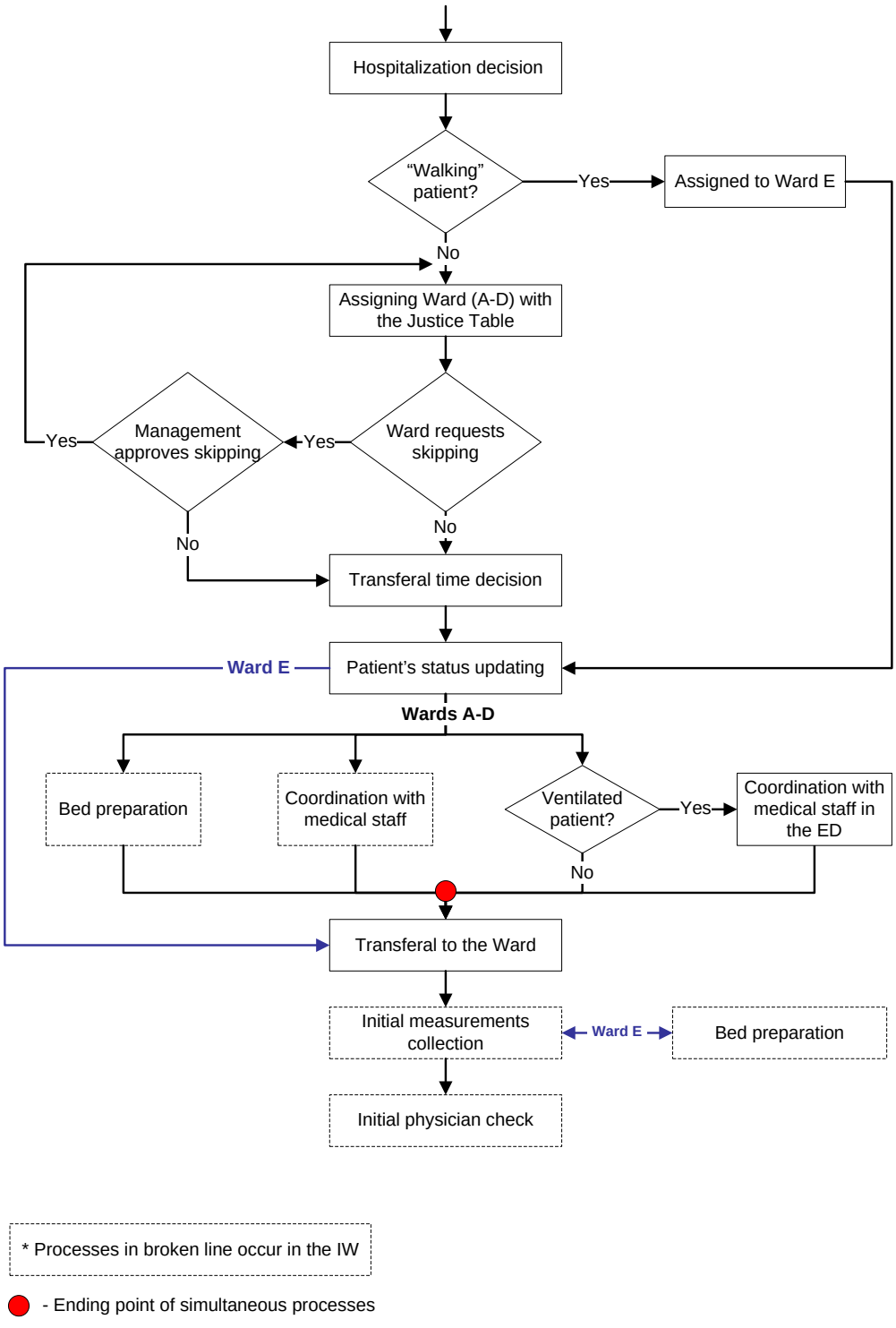


Figure A.2: Resources Flow Chart

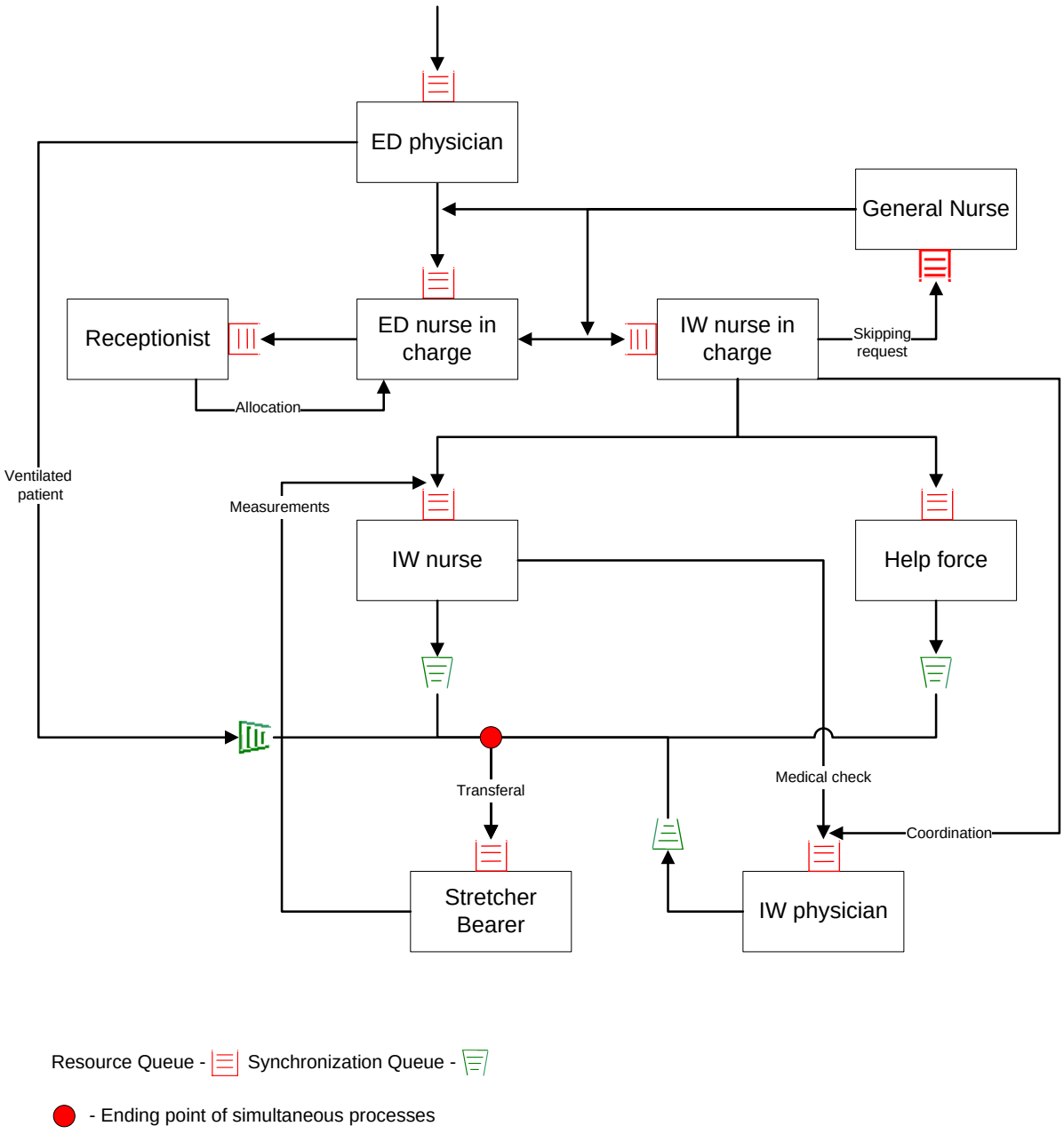
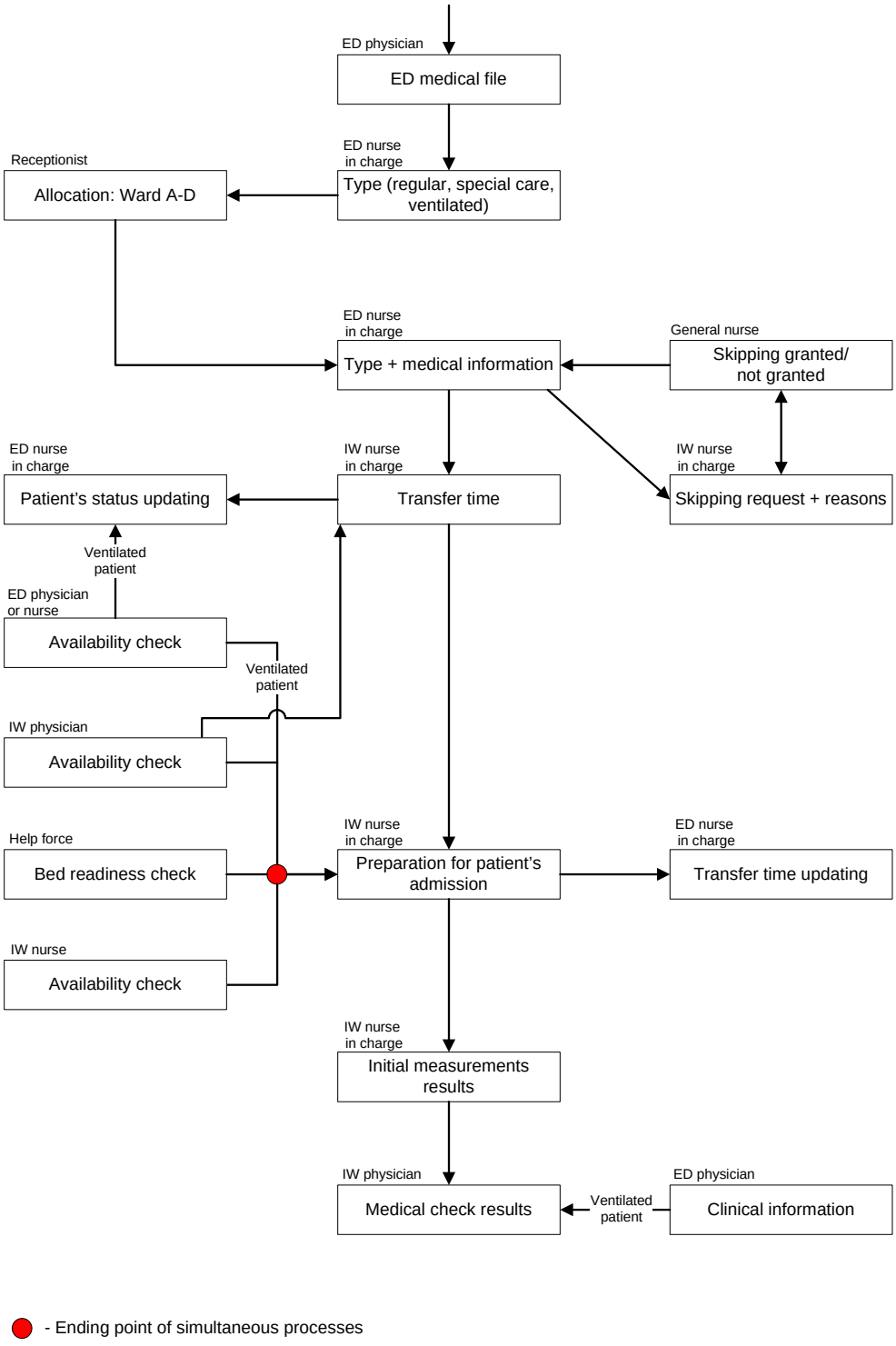


Figure A.3: Information Flow Chart



Appendix B

Questionnaire for Hospital EDs

Figure B.1: Questionnaire

הפקולטה להנדסת תעשיה וניהול
הטכניון מכון טכנולוגי לישראל

איש הקשר:

שם: _____ **תפקיד:** _____ **טלפון:**_____

I. שאלות כלליות:

1. מספר מחלקות האשפוז הפנימיות בבית החולים _____

2. כיצד מסווגים החולים הפנימיים במלר״ד (למשל סיעודי, מונשם,...)?

_____
_____
_____
_____

3. קיבולת המחלקות (מספר מיטות) ואחוז תפוסה ממוצעת (מספר חולים/ קיבולת ממוצעת) – [בממוצע על-פני שנה או פרק זמן אחר שברשותכם]:

מחלקה	קיבולת ממוצעת (מספר מיטות)	אחוז תפוסה ממוצעת
מלר״ד פנימי		
פנימית א׳		
פנימית ב׳		
פנימית ג׳		

II. מדיניות ניתוב החולים, שהוחלט לאשפזם, מהמלר"ד למחלקות הפנימיות.

1. **תיאור התהליך שעובר חולה מרגע ההחלטה על אשפוזו ועד להעברתו למחלקה פנימית ומדיניות בחירת המחלקה הפנימית בה יאושפז החולה** (בפרט: מי משתתף בתהליך; איך מתקבלת ההחלטה באיזו מחלקה יאושפז החולה (על-ידי גוף כלשהו, בעזרת סבבים קבועים בין המחלקות, על-ידי תהליך אוטומטי כלשהו); האם בידי המחלקה הנבחרת סמכות לסרב לקבל את החולה מסיבות כלשהן וכד'):

2. האם ישנה התייחסות מיוחדת לניתובם של "חולים חוזרים" (חולים אשר אושפזו בעבר במחלקה פנימית כלשהי)? אם כן, מהו אפיונם של החולים האלו ומהי מדיניות ניתובם?

III. נתונים תפעוליים של המלר"ד:

1. מספר ממוצע של הפונים הפנימיים למלר"ד ביממה_____
2. מספר ממוצע של החולים הפנימיים המועברים לאשפוז במחלקות הפנימיות ביממה_____
3. משך השהיה הממוצע במלר"ד של החולים הפנימיים _____
_____
4. **משך ההמתנה הממוצע במלר"ד מרגע ההחלטה על האשפוז ועד להעברת החולה למחלקה הפנימית** _____
_____
5. האם משך ההמתנה בסעיף הקודם **משתנה** בין המחלקות הפנימיות השונות? אם כן, מהו משך ההמתנה הממוצע לאשפוז בכל מחלקה? _____
_____
_____
_____

מהן לדעתך הסיבות לשוני במשכי ההמתנה בין המחלקות? _____
_____
_____
_____
_____
_____

IV. נתונים תפעוליים של מחלקות האשפוז הפנימיות:

1. מהו משך אשפוז ממוצע במחלקות הפנימיות?

	משך אשפוז ממוצע (ימים)
פנימית א'	
פנימית ב'	
פנימית ג'	

2. במידה וקיים שוני במשכי האשפוז בין המחלקות השונות, מהן לדעתך הסיבות לקיומו?_____
_____
_____
_____
_____

V. הוגנות בתהליך ניתוב החולים:

1. האם לדעתך שיטת ניתוב החולים הקיימת היא **הוגנת כלפי צוות מחלקות האשפוז (רופאים ואחיות)**? [קריטריונים אפשריים ל**הוגנות**: התחשבות במספר מיטות במחלקות, בתפוסת המחלקות, בזמן שעבר מאז שכל מחלקה קיבלה את החולה האחרון שלה, במשכי האשפוז במחלקות וכד']. אנא פרט/י. _____
_____
_____
_____
_____

איך היית משפר/ת את שיטת ניתוב החולים הקיימת כדי שתהיה הוגנת כלפי הצוות?
_____
_____
_____
_____

2. האם לדעתך שיטת ניתוב החולים הקיימת היא **הוגנת כלפי החולים**? [קריטריונים אפשריים ל**הוגנות**: שמירה על משך ההמתנה במלר"ד שווה לכל חולה, שמירה על כך כי חולה הממתין את הזמן הארוך ביותר לאשפוז יהיה הבא שיאושפז, התחשבות בסיווג הקליני של החולה וכד']. אנא פרט/י. _____
_____
_____
_____
_____

איך היית משפר/ת את שיטת ניתוב החולים הקיימת כדי שתהיה הוגנת כלפי החולים?
_____
_____
_____
_____

הערות נוספות:
_____
_____
_____
_____
_____
_____

תודה רבה על שיתוף הפעולה!!!

Appendix C

Detailed Balance Conditions for the Inverted-V Model under RMI

We first write the detailed balance equations, as in (4.1): this way we simultaneously prove reversibility of the process and obtain its stationary distribution. For convenience, we denote system states in a slightly different manner than in Section 4.2: for each state we list the pools that have busy servers (in the subscript) and note how many servers are busy in these pools (in the superscript). This way, π_{13}^{21} is the stationary probability of the state where pool 1 has two busy servers and pool 3 has one busy server; all other servers in all other pools are idle (in the original notations, this state corresponds to $(2, 0, 1, 0, \ldots, 0)$). With the help of the transition rate diagram in Figure 4.1, we obtain the equations in (C.1).

We start with state (0) (the system is empty) and proceed first to the states with one busy server, then to the states with two busy servers, etc. Recursively, we substitute, as we progress, previously-obtained expressions into newly-obtained ones, thus expressing all as a function of π_0. (This is similar to how one solves the steady-state equations of the one-dimensional Birth and Death Process.) For each state, we obtain several equations (the number of equations for each state equals the number of pools with busy servers at this state); from each equation we obtain the same expression as a function of π_0, where π_0 is calculated at the end through $\sum \pi_y = 1$. This resulting expression is the stationary probability of the state, due to Theorem 4.2.1. Thus the process is reversible.

$$
\begin{aligned}
&\pi_0\frac{\lambda N_i}{N}=\pi_i^1\mu_i && \Rightarrow\ \pi_i^1=\pi_0\frac{N_i\lambda}{N\mu_i} && \forall i\in\{1,2,\ldots,K\}\\
&\pi_i^1\frac{\lambda(N_i-1)}{N-1}=\pi_i^2\cdot 2\mu_i && \Rightarrow\ \pi_i^2=\pi_0\frac{N_i(N_i-1)\cdot\lambda^2}{N(N-1)\cdot 2!\mu_i^2} && \forall i\in\{1,2,\ldots,K\}\\
&\left.\begin{aligned}\pi_i^1\frac{\lambda N_j}{N-1}&=\pi_{ij}^{11}\mu_j\\ \pi_j^1\frac{\lambda N_i}{N-1}&=\pi_{ij}^{11}\mu_i\end{aligned}\right\} && \Rightarrow\ \pi_{ij}^{11}=\pi_0\frac{N_iN_j\cdot\lambda^2}{N(N-1)\mu_i\mu_j} && \forall i\neq j\in\{1,2,\ldots,K\}\\
&\pi_i^2\frac{\lambda(N_i-2)}{N-2}=\pi_i^3\cdot 3\mu_i && \Rightarrow\ \pi_i^3=\pi_0\frac{N_i(N_i-1)(N_i-2)\cdot\lambda^3}{N(N-1)(N-2)\cdot 3!\mu_i^2} && \forall i\in\{1,2,\ldots,K\}\\
&\left.\begin{aligned}\pi_{ij}^{11}\frac{\lambda(N_i-1)}{N-2}&=\pi_{ij}^{21}\cdot 2\mu_i\\ \pi_i^2\frac{\lambda N_j}{N-2}&=\pi_{ij}^{21}\mu_j\end{aligned}\right\} && \Rightarrow\ \pi_{ij}^{21}=\pi_0\frac{N_i(N_i-1)N_j\cdot\lambda^3}{N(N-1)(N-2)\cdot 2!\mu_i^2\mu_j} && \forall i\neq j\in\{1,2,\ldots,K\}\\
&\left.\begin{aligned}\pi_{ij}^{11}\frac{\lambda N_k}{N-2}&=\pi_{ijk}^{111}\mu_k\\ \pi_{ik}^{11}\frac{\lambda N_j}{N-2}&=\pi_{ijk}^{111}\mu_j\\ \pi_{jk}^{11}\frac{\lambda N_i}{N-2}&=\pi_{ijk}^{111}\mu_i\end{aligned}\right\} && \Rightarrow\ \pi_{ijk}^{111}=\pi_0\frac{N_iN_jN_k\cdot\lambda^3}{N(N-1)(N-2)\cdot\mu_i\mu_j\mu_k} && \forall i\neq j\neq k\in\{1,2,\ldots,K\}
\end{aligned}
\tag{C.1}
$$

Continuing this way for each number of busy servers, we obtain the result of (4.2) (we go back to the states' representation in terms of $y=(y_1,y_2,...,y_K)$: each y_i represents the number of busy servers in pool i, and $m_y=\sum_{i=1}^K y_i$ - the total number of busy servers at state y):

$$
\pi_y=\pi_0\prod_{i=1}^K\binom{N_i}{y_i}\frac{(N-m_y)!}{N!}\frac{\lambda^{m_y}}{\prod_{i=1}^K\mu_i^{y_i}} \qquad \forall y_i\in\{0,1,\ldots,N_i\},\ \forall i=1,2,\ldots,K
$$

Note, that the pools are assumed *heterogeneous*, namely $\mu_i\neq\mu_j,\ \forall i\neq j\in\{1,2,\ldots,K\}$. Otherwise, if there are several pools with equal service rate, we unite them into a single pool (the change is notational only).

Appendix D

Proof of Theorem 4.5.1 for Steady-State

We wish to prove that the steady state queue length in a system with N servers is stochastically dominated by the steady state queue length in a system with $N-1$ servers (recall that in Section 4.5.1 we showed stochastic order of the queue length processes *for all times*). Consequently the order of means will be implied: the mean queue length in steady state is always smaller in a system with N servers than in a corresponding system with $N-1$ servers. We denote the steady-state queue length in the system with j servers by L_q^j, and the steady-state waiting time in queue in the system with j servers by W_q^j $(j = N-1, N)$.

Theorem D.0.1 $P(L_q^N > l) < P(L_q^{N-1} > l),\ \forall l > 0.$

In words, L_q^N is stochastically dominated by L_q^{N-1}.

Proof First we should find an expression for $P(L_q > l)$. In Section 4.6, we found performance measures ($P(L_q = l)$, $P(W_q > 0)$) for the general inverted-V model with multiple-servers pools under RMI. Here, similarly to Section 4.5.1, we address a single-server-pools system under RA (and by Theorem 4.3.1, the result is also valid for the RMI policy in the multiple-servers pools). Hence, by substituting $K = j$ $(j = N-1, N)$ and $N_i = 1,\ \forall i \in \{1, 2, \ldots, K\}$, to the relevant expressions in Section 4.6, we obtain:

$$P(L_q^j > l) = \sum_{i=l+1}^{\infty} P(L_q^j = l) \overset{(4.22)}{=} \sum_{i=l+1}^{\infty} \rho_j^l (1-\rho_j) P(W_q^j > 0) = \rho_j^{l+1} P(W_q^j > 0) = \\ \overset{(4.19)}{=} \rho_j^{l+1} \pi_0^j \frac{\lambda^j}{j! \prod_{i=1}^{j} \mu_i} \left(\frac{1}{1-\rho_j} \right), \qquad \forall l > 0, \quad j = N-1, N; \tag{D.1}$$

where $\rho_j = \dfrac{\lambda}{\sum_{i=1}^{j} \mu_i}$ $(j = N-1, N)$, and π_0^j is obtained from (4.20), by substituting $K = j$ and $N_i = 1$, $\forall i \in \{1, 2, \ldots, K\}$, and equals to:

$$\pi_0^j = \left[\sum_{x_1=0}^{1} \sum_{x_2=0}^{1} \cdots \sum_{x_j=0}^{1} \frac{(j-m_x)!}{j!} \frac{\lambda^{m_x}}{\prod_{i=1}^{j} \mu_i^{x_i}} \left(\frac{1}{1-\rho_j} \right)^{(m_y - j + 1)^+} \right]^{-1}, \qquad j = N-1, N. \tag{D.2}$$

We wish to prove that $P(L_q^N > l) < P(L_q^{N-1} > l)$, $\forall l > 0$. Then, from (D.1), we should show that:

$$\rho_N^{l+1} \cdot \pi_0^N \frac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i} \left(\frac{\sum_{i=1}^{N} \mu_i}{\sum_{i=1}^{N} \mu_i - \lambda} \right) < \rho_{N-1}^{l+1} \cdot \pi_0^{N-1} \frac{\lambda^{N-1}}{(N-1)! \prod_{i=1}^{N-1} \mu_i} \left(\frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right).$$

Divide both sides of the inequality by the positive expression $\dfrac{\lambda^{N-1}}{(N-1)! \prod_{i=1}^{N-1} \mu_i}$:

$$\left(\frac{\lambda}{\sum_{i=1}^{N} \mu_i} \right)^{l+1} \pi_0^N \frac{\lambda}{N \mu_N} \left(\frac{\sum_{i=1}^{N} \mu_i}{\sum_{i=1}^{N} \mu_i - \lambda} \right) < \left(\frac{\lambda}{\sum_{i=1}^{N-1} \mu_i} \right)^{l+1} \pi_0^{N-1} \left(\frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right)$$

Reorganizing the last inequality leads to:

$$\frac{\pi_0^N \lambda}{\pi_0^{N-1} N \mu_N} < \frac{\sum_{i=1}^{N-1} \mu_i \cdot \left(\sum_{i=1}^{N} \mu_i - \lambda \right)}{\sum_{i=1}^{N} \mu_i \cdot \left(\sum_{i=1}^{N-1} \mu_i - \lambda \right)} \frac{\left(\frac{\lambda}{\sum_{i=1}^{N-1} \mu_i} \right)^{l+1}}{\left(\frac{\lambda}{\sum_{i=1}^{N} \mu_i} \right)^{l+1}} \tag{D.3}$$

Substituting π_0 from (D.2) leads to:

$$\frac{\pi_0^N \lambda}{\pi_0^{N-1} N \mu_N} = \frac{(\pi_0^{N-1})^{-1} \frac{\lambda}{N \mu_N}}{(\pi_0^N)^{-1}} =$$

$$= \frac{\overbrace{\left[\sum_{m=0}^{N-2} \sum_{\substack{x_1 \ldots x_{N-1}: \\ \sum x_i = m}} \frac{(N-1-m)!}{N!} \frac{\lambda^{m+1}}{\prod_{i=1}^{N-1} \mu_i^{x_i} \cdot \mu_N} \right]}^{a} + \overbrace{\frac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i} \left(\frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right)}^{b}}{\underbrace{\left[\sum_{m=0}^{N-1} \sum_{\substack{x_1 \ldots x_N: \\ \sum x_i = m}} \frac{(N-m)!}{N!} \frac{\lambda^m}{\prod_{i=1}^{N} \mu_i^{x_i}} \right]}_{c} + \underbrace{\frac{\lambda^N}{N! \prod_{i=1}^{N} \mu_i} \left(\frac{\sum_{i=1}^{N} \mu_i}{\sum_{i=1}^{N} \mu_i - \lambda} \right)}_{d}} \tag{D.4}$$

In order to show that (D.4) is smaller than the right side of (D.3), we employ the following observation:

Observation 1 *For any positive numbers e,f,g,h, it holds that:* $\frac{e+f}{g+h} < \frac{f}{h}$, *if* $h < f$ *and* $e < g$ *(iff* $eh < gf$*).*

We wish to show that $\frac{(a)+(b)}{(c)+(d)} < \frac{(b)}{(d)}$. Then we need to check that $(d) < (b)$ and $(a) < (c)$:

- The inequality $(d) < (b)$ follows straight from the following observation:

Observation 2 *For any positive numbers e,f,g, it holds that:* $\frac{e+f}{e+f-g} < \frac{e}{e-g}$,

where $e = \sum_{i=1}^{N-1} \mu_i$, $f = \mu_N$ and $g = \lambda$.

- To show $(a) < (c)$, let us perform variables change in (a), taking k=m+1:
(a)=$\sum_{k=1}^{N-1} \sum_{\substack{x_1 \ldots x_{N-1}: \\ \sum_{x_i}=k-1}} \frac{(N-k)!\lambda^k}{N! \prod_{i=1}^{N-1} \mu_i^{x_i} \cdot \mu_N}$. Then (c)=(a)+$\sum_{m=0}^{N-1} \sum_{\substack{x_1 \ldots x_{N-1}: \\ \sum_{x_i}=m}} \frac{(N-m)!\lambda^m}{N! \prod_{i=1}^{N-1} \mu_i^{x_i}}$.
Clearly, the last expression is positive, thus the inequality holds.

Returning to (D.4) and using Observation 1:

$$\frac{\pi_0^N \lambda}{\pi_0^{N-1} N \mu_N} < \frac{(b)}{(d)} = \frac{\frac{\lambda^N}{N! \prod_{i=1}^N \mu_i} \left(\frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right)}{\frac{\lambda^N}{N! \prod_{i=1}^N \mu_i} \left(\frac{\sum_{i=1}^{N} \mu_i}{\sum_{i=1}^{N} \mu_i - \lambda} \right)} = \frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N} \mu_i} \left(\frac{\sum_{i=1}^{N} \mu_i - \lambda}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right) <$$

$$< \frac{\sum_{i=1}^{N-1} \mu_i}{\sum_{i=1}^{N} \mu_i} \left(\frac{\sum_{i=1}^{N} \mu_i - \lambda}{\sum_{i=1}^{N-1} \mu_i - \lambda} \right) \frac{\left(\frac{\lambda}{\sum_{i=1}^{N-1} \mu_i} \right)^{l+1}}{\left(\frac{\lambda}{\sum_{i=1}^{N} \mu_i} \right)^{l+1}}$$

The last inequality is due to $\frac{\left(\frac{\lambda}{\sum_{i=1}^{N-1} \mu_i} \right)^{l+1}}{\left(\frac{\lambda}{\sum_{i=1}^{N} \mu_i} \right)^{l+1}} > 1$.

We thus obtain (D.3). ∎

Note: The above proof is valid for **any** N-1 servers (not necessarily the faster ones) and, as a special case, for the system with homogeneous servers (Erlang-C) as well.

Appendix E

Detailed Balance Conditions for the DFQ Model under RMI

We write the detailed balance equations, as in (4.1): this way we simultaneously prove reversibility of the process and obtain its stationary distribution. First, let us observe that for the states where $z_i \leq N_i,\ \forall i \in \{1, 2, \ldots, K\}$, the transition rate diagram and thus the balance equations, are similar to the $\wedge-$model under RMI, for $N_i = C_i$ (these equations appear in (C.1)). Hence, for π_z of these states, we obtain similar expressions as in Appendix C (substituting $N_i = C_i$):

$$\pi_z = \pi_0 \prod_{i=1}^{K} \binom{C_i}{z_i} \frac{(C - m_z)!}{C!} \frac{\lambda^{m_z}}{\prod_{i=1}^{K} \mu_i^{z_i}}, \qquad \forall z_i \in \{0, 1, \ldots, N_i\},\ \forall i = 1, 2, \ldots, K \quad \text{(E.1)}$$

For other states, where there exists an $i \in \{1, 2, \ldots, K\}$, such that $z_i > N_i$ (at least one of the pools has a non-empty queue), the balance conditions change. For convenience, we denote system states as in Appendix C: for each state, we list the pools that have busy servers (in the subscript) and note how many customers are in these pools (in the superscript). This way, π_{13}^{21} is the stationary probability of the state where pool 1 has two customers and pool 3 has one customer; all other servers in all other pools are idle (in our original notations, this state corresponds to $(2, 0, 1, 0, \ldots, 0)$). With the help of the transition rate diagram in Figure 6.2, we obtain the equations in (E.2).

For each pool i, we start with state (N_i) (all the servers in pool i are busy and its queue is empty, and all other pools have no customers) and gradually "add" customers to the system. Recursively, we substitute, as we progress, previously-obtained expressions into newly-obtained ones, thus expressing all as a function of π_0 (as in Appendix C). For

each state, we obtain several equations (the number of equations for each state equals the number of pools with busy servers at this state); from each equation we obtain the same expression as a function of π_0, where π_0 is calculated at the end through $\sum \pi_z = 1$. This resulting expression is the stationary probability of the state, due to Theorem 4.2.1. Thus the process is reversible.

$$\pi_i^{N_i} \frac{\lambda b_i}{C - N_i} = \pi_i^{N_i+1} N_i \mu_i \quad \Rightarrow \quad \pi_i^{N_i+1} = \pi_0 \frac{C_i!(C - N_i - 1)!}{C!(C_i - N_i - 1)! N_i! N_i} \frac{\lambda^{N_i+1}}{\mu_i^{N_i+1}} \quad *$$

$$\pi_i^{N_i+1} \frac{\lambda(b_i - 1)}{C - N_i - 1} = \pi_i^{N_i+2} N_i \mu_i \quad \Rightarrow \quad \pi_i^{N_i+2} = \pi_0 \frac{C_i!(C - N_i - 2)!}{C!(C_i - N_i - 2)! N_i! N_i^2} \frac{\lambda^{N_i+2}}{\mu_i^{N_i+2}} \quad *$$

$$\left.\begin{aligned} \pi_i^{N_i+1} \frac{\lambda C_j}{C - N_i - 1} &= \pi_{ij}^{(N_i+1)1} \mu_j \\ \pi_{ij}^{N_i 1} \frac{\lambda b_i}{C - N_i - 1} &= \pi_{ij}^{(N_i+1)1} N_i \mu_i \end{aligned}\right\} \quad \Rightarrow \quad \pi_{ij}^{(N_i+1)1} = \pi_0 \frac{C_i!(C - N_i - 2)! C_j}{C!(C_i - N_i - 1)! N_i! N_i} \frac{\lambda^{N_i+2}}{\mu_i^{N_i+1} \mu_j} \quad **$$

$$\begin{aligned} * \quad & \forall i \in \{1, 2, \ldots, K\} \\ ** \quad & \forall i \neq j \in \{1, 2, \ldots, K\} \end{aligned} \tag{E.2}$$

Continuing this way, for every possible number of customers in each pool, and then generalizing the expressions (E.1) and (E.2) (we go back to the states' representation in terms of $z = (z_1, z_2, ..., z_K)$: each z_i represents the number of customers in pool i, and $m_z = \sum_{i=1}^{K} z_i$ - the total number of customers at state z), we obtain:

$$\pi_z = \pi_0 \prod_{i=1}^{K} \left(\frac{C_i!}{(C_i - z_i)!} \frac{N_i^{-(z_i - N_i)^+}}{(N_i \wedge z_i)! \cdot \mu_i^{z_i}} \right) \frac{\lambda^{m_z}(C - m_z)!}{C!}, \tag{E.3}$$

where π_0 is obtained from $\sum \pi_z = 1$ and equals to:

$$\pi_0 = \left[\sum_{z_1=0}^{C_1} \cdots \sum_{z_K=0}^{C_K} \left(\prod_{i=1}^{K} \left(\frac{C_i! \cdot N_i^{-(z_i - N_i)^+}}{(C_i - z_i)!(N_i \wedge z_i)! \mu_i^{z_i}} \right) \frac{\lambda^{m_z}(C - m_z)!}{C!} \right) \right]^{-1}. \tag{E.4}$$

Note, that the pools are assumed *heterogeneous*, namely $\mu_i \neq \mu_j$, $\forall i \neq j \in \{1, 2, \ldots, K\}$. Otherwise, if there are several pools with equal service rate, we unite them into a single pool (the change is notational only).