Resource-Driven Activity-Networks (RANs): A Modelling Framework for Complex Operations

Petar Momčilović
Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA, petar@tamu.edu

Avishai Mandelbaum
Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel, avim@technion.ac.il

Nitzan Carmeli
Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel, nitzany@campus.technion.ac.il

Mor Armony
Stern School of Business, New York University, New York, NY 10012, USA, marmony@stern.nyu.edu

Galit Yom-Tov
Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel, gality@technion.ac.il

Key words: Fluid or macroscopic models, functional strong-law-of-large-numbers, asymptotic approximations, heavy-traffic; time-varying, dynamical systems, activity analysis in economics, SPN.
History: November 14, 2022

Abstract

We develop a data-based modelling framework that supports analysis and optimization of large and complex, congestion-prone service-operations. Drawing from many-server asymptotic regimes, all finite-capacity participants in the service process (e.g. customers, servers) are equally considered resources that are either busy or await each others' availability. Models are then activity-oriented, where each activity (e.g. exam in a hospital) first consumes a subset of the resources (e.g. patient, doctor, exam-room) at specific states (e.g. waiting patient, available doctor, idle exam-room); and then, upon activity completion, produces possibly other resources at other states (e.g. served patient, available doctor, exam-room that requires cleaning). We hence refer to our models as Resource-Driven Activity Networks, or RANs for short.

The RAN language is that of Linear Input-Output Economics, hence the RAN framework is parsimonious yet rich. Practically, it covers features such as multiple resource types, complex interactions among resources, highly variable and long processes, exchangeable fork-join constructs, all within time-varying environments. Theoretically, RANs cover dynamic- and static-models, closed- and open-networks, and both many- and single-server asymptotic regimes.

This first RAN paper is at the fluid (average, macroscopic) level. Our fluid RANs are thus deterministic models, yet they are also stochastic-aware in that activity durations are not negligible (as in conventional fluid models) – in fact, the distributions of these durations are model primitives. Fluid RANs could also enjoy an intrinsic modeling value, by *directly* approximating large complex operations (rather than indirectly, by approximating stochastic models of these operations). Notably, already the fluid level gives rise to ample research challenges that are either novel or originate from classical models when viewed through a RAN-lens.

Contents

1. Introduction

Service systems (e.g. telecommunication, healthcare, transportation), while prevalent in our lives, have become large, complex and congestion-prone as demand for their offering has proliferated. Service engineers thus face the challenge of designing operations with many resources (e.g. many servers and customers) that interact through complex processes. In turn, managers actualize these designs, while seeking to match their system capacity with its demand; indeed, a mismatch would result in excessive waiting of resources for each other (e.g. long waiting time of customers for servers – low quality, or long waiting time of servers for customers – low efficiency).

Consider, for example, emergency departments (EDs) or outpatient clinics. Such systems involve multiple resource types (physicians, nurses, technicians, etc.) and multiple patient types (differentiated by severity and medical speciality), all adhering to highly variable and complex processes that evolve within a queueing-rich time-varying environment (Armony et al. 2015). Moreover, due to technological advances (e.g., Real-Time Location Systems (RTLS), see Figure 1), one can mine features and processes of such systems from their data – data that have become accessible in a previously-unparalleled quantity, resolution and quality. This prompts the additional challenge of managing and modelling operational big-data.

The above challenges — design, management, data — call for modelling support that, traditionally, has been provided by *queueing networks*, e.g. Chen and Yao (2013): here *networks* depict complex multi-activity processes while *queues* capture contention for scarce resources. Yet queueing networks often suffer from the "curse of dimensionality" in their analytical scale (being intractable), or "curse of simplicity" in their modelling scope (unable to accommodate complex features). The scale-shortcoming has given rise to a flourishing research area: approximations of queueing networks that arise from their fluid and diffusion asymptotics. The scope-shortcoming, however, is still a persisting challenge, and a main goal of our research is to alleviate it to some extent. To this end, we borrow strength from *many-server asymptotic regimes* while following the seminal research path of Harrison (1988, 2002, 2003) (that, unlike here, was conceived in single-server or *conventional heavy-traffic*.)

More concretely, we develop a mathematical framework for operational models of service systems, which we refer to as Resource-Driven-Activity-Networks (RANs). RANs offer a symmetric viewpoint under which all scarce service-constituents (e.g., servers and customers (Newell 1973)) are equally considered as *resources*. Furthermore, RANs accommodate complex features (e.g. time-dependence, randomness), multiple operational-regimes (e.g. many-server or conventional heavy- or light-traffic) and elaborate interactions between resources, all via an inequality (3) that mathematically articulates a set of feasible "plans"; and much like Input-Output models

of economic systems (Leontief 1986, Koopmans 1951), RANs offer a unifying umbrella for models that are dynamic or static, closed or open, and that enjoy simultaneous consumption and production by and of multiple resources. The present paper is a first step at the fluid (laws of large numbers) level; hence Fluid RANs are deterministic models. Yet they are also stochastic-aware in that model-primitives for activity durations are their cumulative distribution functions (cdf's): if G is the cdf of a specific activity duration, and this activity started at time u, then RAN's articulation of its completion-time is $G(t-u)$ (1), that is the probability that the activity completed by time t, for all $t \geq u$ (Krichagina and Puhalskii 1997, Reed 2009, Liu and Whitt 2012).

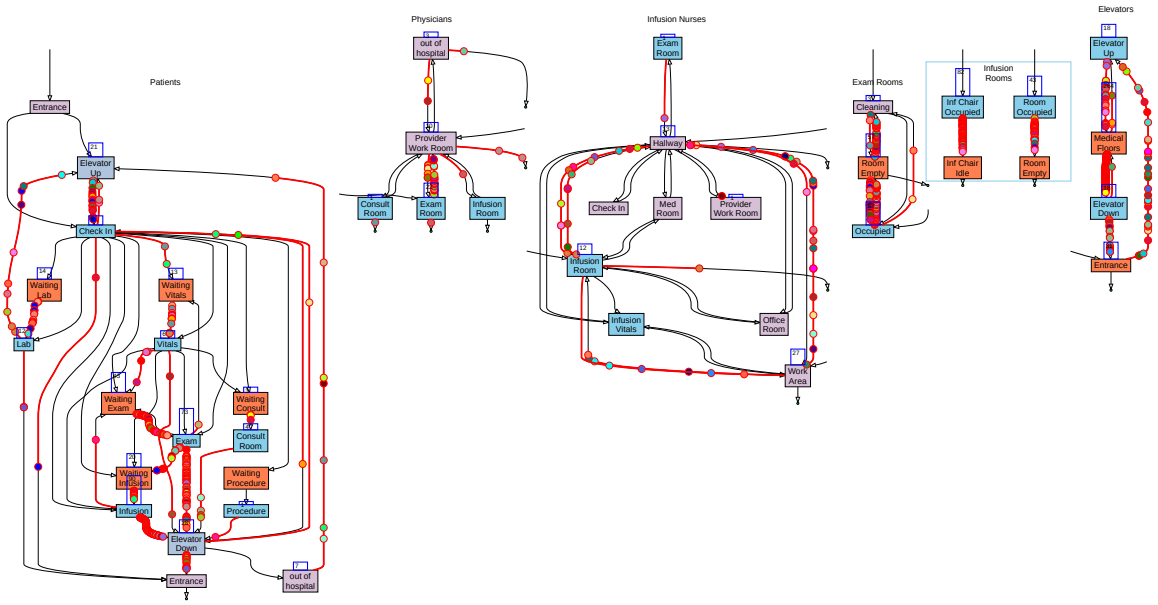
The case for a new framework better be well-justified. The rest of the Introduction is hence our attempt to motivate this case, both practically and theoretically, as it arises from data-based exploration of real-life service systems.

1.1. Practical Motivation: Complexity, Size; Resource-View, Long Services

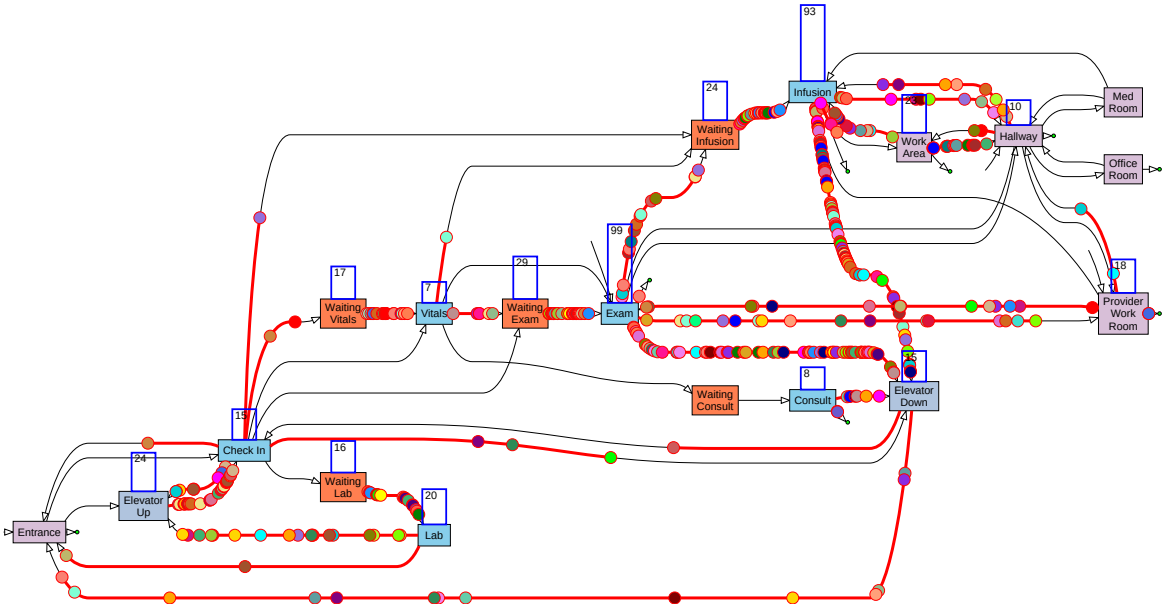
We start with introducing RAN's view of scarce service-constituents: all play an equal role of a resource that dynamically changes its state. (The simple setting of two resources – customers and servers – was captured by the Erlang-S model, "S" for "Servers", in Azriel et al. (2019).) Then, we contrast service systems in heavy-traffic via their capacities, more precisely few- vs. many-servers, which reveals an essential comparative characteristic: long service duration in the latter (long relative to queueing and idling time) vs. short in the former.

1.1.1. Symmetric Resource-View of Complex Systems (Scope). We now demonstrate the complexity of service operations, and our symmetric resource-view of them. To this end, consider Figure 1: it is based on data from our data-partner, the Dana-Farber Cancer Institute (DFCI), which is a large outpatient oncology center in Boston MA, USA. On average, 1000 patients visit DFCI every day, and they are catered to or receive treatment by 300–400 staff members. Each patient, staff member and equipment at DFCI is equipped with a badge that broadcasts badge locations almost continuously. These locations, in turn, are captured and archived by RTLS, which is the main source of our data. In addition, we also have the data of all planned itineraries at DFCI. Combining RTLS (the actual) and itineraries (the planned) enables the creation of a complete vast-quantity, high-quality transaction-level data, as depicted in our Figures.

To elaborate, Figure 1a presents a (time) snapshot of activity networks of the following resources at DFCI: patients, physicians, nurses, exam rooms, infusion beds/chairs and elevators. (Elevators are the only resource that has no itinerary). Our data is dynamically depicted in



(a) Activity networks of 6 resources in DFCI: patients, physicians, infusion nurses, exam rooms, infusion chairs/rooms, elevators. Data-animation is accessible at the following LINK1. In each network, the colored discs represent individual resources, while the animation recreates their flow through activities within the network, during a single day. Blue rectangles represent activities that involve more than one resource (for example: the activity *exam* involves a patient, a physician, and an exam room); Orange rectangles correspond to delays (either idling or queueing); Purple rectangles are activities that involve only a single resource.



(b) Integrative RAN-view of the above 6 Activity networks, with data-animation accessible at LINK2. Resources dynamically change their state, which is animated by their flow through activities. Activities, in turn, consume resources and, after an "activity-duration," produce resources that possibly support future activities. For example, activity *exam*, that consumes a waiting patient, idle doctor and available exam room, produces, for example, an idle doctor, a patient moving on to infusion, and a room that is to be cleaned.

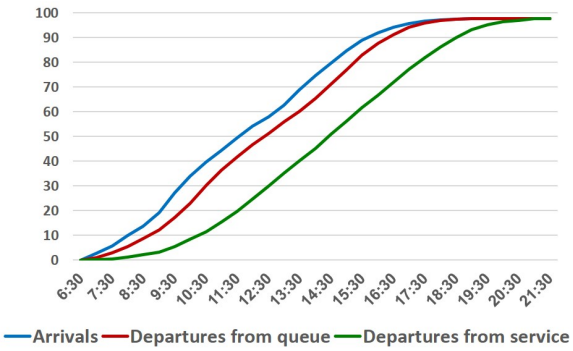
Figure 1 Two views of a service network: (a) Activity networks of individual resources (in DFCI: patients, physicians, infusion nurses, exam rooms, infusion chairs/rooms, elevators); vs. (b) Integrative dynamic RAN view, where resources interact as they are consumed and produced by activities.

LINK1, which is data-animation that vividly demonstrates the richness and high-resolution of our data. The animation also illustrates the complexity of DFCI service processes, complexity that is common to many service systems and healthcare in particular. Such operations require coordination and synchronization of ample resources that hence dynamically change their state over time.

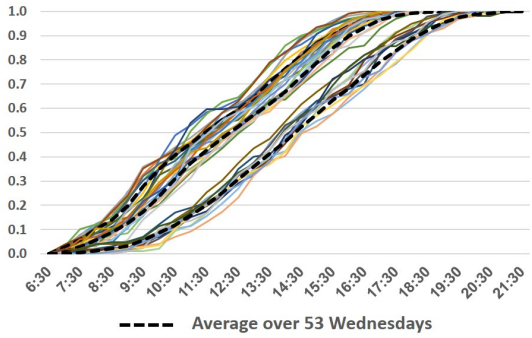
Figure 1b combines, under a single RAN-roof, the constituents of 1a, thus depicting our symmetric resource-view (data-animation is accessible in LINK2). Indeed, all resources are integrated into a RAN, that models interactions and flows of resources, as activities consume and produce them. For example, patients' rides in elevators are activities that require a synchronization of two resources—riders (patients, visitors, ...) and an available elevator, all simultaneously present on the same floor. (In fact, since the start of Covid, DFCI has imposed restrictions on the number of people who can share an elevator; hence elevators became a scarce resource for which people wait non-negligible times.)

1.1.2. Large Systems with "Long" Service Durations (Scale). An alternative characterization of few-server (conventional) heavy-traffic asymptotics versus many-server heavy traffic is in terms of relatively *short (negligible)* service durations (few-servers) vs. *long (non-negligible)* service durations (many-servers). Here short or long is relative to queueing time (alternatively sojourn time). This implies that, for queueing networks in conventional heavy traffic, customers' sojourn time within a station is essentially their queueing time (service duration negligible relatively to queueing). In contrast, for many-server queueing networks in heavy traffic, both queueing and service durations could be non-negligible; however, in a station that is critically loaded (demand is well-balanced against capacity), sojourn time is essentially service duration (queueing time negligible relative to service).

The above is captured by Figure 2, that focuses on the infusion process in a single disease center at DFCI: it presents cumulative arrivals, cumulative departures from queue (service initiations), and cumulative departures from service. Note that, although our dynamic fluid model arises from realizations (individual days) of a large system, it does also cover an average day ("average Wednesday" in our case), the shape of which resembles individual days. Assuming first-come-first-served (FCFS) policy, the horizontal difference (in Figure 2a) between the cumulative arrivals (in blue) and the cumulative departures from queue (in red) corresponds to customers' waiting time; the horizontal difference between the cumulative departures from queue and the cumulative departures from service (in green) corresponds to service durations (Hall 1991, Ch. 6). Thus, it is observed that service durations are much longer than waiting durations, in every individual Wednesday, and on average.



(a) Cumulative arrivals, departures from queue, and departures from service. Average over 53 Wednesdays, 02/2019 – 02/2020.



(b) Cumulatives, normalized w.r.t. daily totals: individual Wednesdays, 07/2019 – 09/2019, and average over all 53 Wednesdays 02/2019 – 02/2020.

Figure 2 Cumulative arrivals, service initiations, and departures from service at an infusion unit in DFCI.

Service Processes vs. Merely their Duration. Duration is but one feature of what is typically an elaborate service *process*. It does suffice in conventional heavy-traffic, where fluid-service "is" in fact its service-rate. However, models with "long" services are required to capture features beyond duration. Examples include Mandelbaum and Reiman (1998) (pooling networked servers into a single universal server); Carmeli (2015) (design of self-services, e.g. in answering machines); and Carmeli (2020) (multi-featured load-balancing, e.g. striving for "fair" allocation of workload).

1.2. Theoretical Motivation: Generalizing SPNs, Operational-Regimes, Taxonomies

In Section 3.2.5, we formally explain how our RANs generalize SPNs (Stochastic Processing Networks), the latter being a state-of-art modeling framework for complex operations (Harrison 1988, 2000, 2002, 2003). A comprehensive treatment of SPNs is provided in the recent Dai and Harrison (2020) where, following Dai (1995), fluid SPNs are used as tools for establishing stability of their originating Stochastic SPNs: fluid stability (fluid levels vanish within a finite time) implies stochastic stability (positive recurrence). Consequently, the simpler the fluid model the better (easier to prove stability), and it turns out that piecewise-linear fluid dynamics suffices. Fluid RANs, in contrast, serve as standalone models of complex realities, hence the broader their modelling scope the better, in particular RAN dynamics is time-varying.

Both SPNs and RANs accommodate complexity via the input-output view of activity analysis (Koopmans 1951). However, SPNs constitute resources (e.g. servers), materials (e.g., customers), activities (e.g., services), and then protocols for their interactions. RANs, on the other hand, acknowledge only resource and activities (e.g. customers and serves are both resources), which is a main enabler of their modelling parsimony.

1.2.1. Mixed Heavy-Traffic Regimes: Single-Resourcing. The RAN framework is motivated by many-server asymptotic regimes, and the SPN framework (Dai and Harrison (2020)) by conventional (single-server) heavy-traffic. Often enough, however, both regimes happen to *coexist* within a single system. For example, a typical patient-flow at DFCI (Figure 1b) starts with blood-test (single service-station with many-servers = nurses), continues with a medical exam (many service stations, each with a single-server = physician) and ends with infusion (single station many-servers = beds). Another example is vehicle rental systems, where vehicles can be rented from various locations. Such systems, when analyzed from the vehicle perspective, are closed systems with both single-server queues and infinite-server queues (George and Xia 2011, Benjaafar et al. 2022, Braverman et al. 2019); see §4.4.

In Section 3.2 we show that the RAN modeling framework also accommodates mixed heavy-traffic regimes. This is motivated via an asymptotic procedure, under which any resource in the system can become a "single-server" resource that operates in conventional heavy-traffic; we call this procedure *single-resourcing*. In the resulting *mixed RAN*, resources can thus operate in conventional heavy-traffic or in the many-server heavy-traffic regime, and any combination of single- or few-, many- or infinite-resources is allowed. The above implies that SPNs are special cases of RANs: both *practically* (SPNs are *approximately* RANs with very-short activity durations), as well as *theoretically* (Fluid SPNs are *special cases* of our RANs with only single-resources §3.2.5).

1.2.2. Taxonomy of Systems: Open, Closed and Time-Varying. In the queueing theory literature, the terms *open network* and *closed network* usually refer to the customer population. Specifically, a service system is *open* if customers arrive to it from the "outside world", where there is effectively an infinite customer population; they receive service within the system from servers awaiting them; and after completing their sojourn they leave (return to the "outside world"). While such systems are called "open", from the server perspective they are in fact closed since there is a constant population of servers that is an endogenous part of the system. Most queueing networks are therefore mixed, having "open"-customer and "closed"-server populations.

As already noted by Newell (Figure 1 in Newell (1973), the labels of "customers" and "servers" are often interchangeable. To elaborate, a service system is commonly called *closed* if, similarly to the servers above, customers also constitute a finite population, which circulates perpetually within the system while obtaining services. Classical models of such systems are machine-repairmen (Momčilović and Motaei 2018), and closed Jackson networks (Prisgrove 1987, Chen and Mandelbaum 1991c). But there are service systems in which, naturally, both customers and

servers are "open", e.g., ride hailing systems such as Uber or Lyft (Afeche et al. 2018, Özkan and Ward 2020, Aveklouris et al. 2021); or where customers are "closed" while servers are "open" (e.g., tele-medicine services provided to prison inmates (Schooley et al. 2019)).

The distinction between who-are-customers and who-are-servers thus reduces to merely a modelling terminology, while formally they play the symmetric role of resources. Then the RAN notion §2.2.6 of closed vs. open is associated with an individual resource (rather then a system), while addressing ambiguities that arise in time-varying systems (Massey 1985, Whitt 2018). For example, in a $G_t/GI/s_t$ RAN, the server-population constitutes a closed resource, even though servers need not stay in the system perpetually – s_t can both increase and decrease – which could model servers entering and leaving the system according to their shifts.

1.3. Contributions

In this paper, we propose a framework for modeling service operations: model-features arise from data (e.g. Figure 1), and large-scale asymptotics provides the theoretical foundation (Halfin and Whitt 1981, Garnett et al. 2002, Van Leeuwaarden et al. 2017). As a first step, we focus on fluid scaling, and similarly to Harrison (1988) *"we do not attempt a rigorous convergence proof to justify the proposed approximation, but the argument given in support of the approximation amounts to a broad outline for such a proof"*. Our outline is given in Section 2.3, but our main rigor-lack justification stems from how useful RANs are, even under the present macroscopic (fluid) view: they provide a platform for theoretical and practical modeling, analysis, and optimization of numerous queueing networks, and beyond.

Operations research of fluid approximations and models has been proliferating. For example, Zychlinski (2022) lists about 150 applications of fluid models (which excludes theoretical research such as single- and many-server FSLLNs, measure-valued models and piecewise-linear limits of stochastic GJNs and SPNs). Our humble view is that RANs add a timely useful step to the evolution of this "fluid-research": we support it in Appendix A, by cherry-picking (necessarily biased) steps in that evolution.

The value of a fluid model increases with the complexity of the features and processes of its origin, which is a stochastic model – as in Dai and Harrison (2020), or a real system – as in the present paper (e.g. DFCI in Figure 1, and the data-based example in §4.6). In the former case, the fluid model is a means for establishing stability of their originating stochastic model. In contrast, our fluid models capture the operational dynamics of their originating systems, they are hence bona fide models with a stand-alone value, as we now describe:

- The RAN framework provides a unifying umbrella for capacity-constrained models that are dynamic or static, closed or open; and they operate simultaneously in conventional and many-server heavy-traffic, while possibly alternating among operational regimes (e.g. under-, over- or critically-loaded). More concretely, we show in Section 4 that RANs capture a wide variety of queueing models, for example the $G_t/GI/s_t$ queue (Puhalskii and Reed 2010, Liu and Whitt 2012), open and closed generalized Jackson networks (Kaspi and Mandelbaum 1992, Chen and Mandelbaum 1994, Prisgrove 1987), parallel-server networks (Williams 2000), machine-repair systems (Momčilović and Motaei 2018), and re-balancing networks (Braverman et al. 2019).
- In contrast with earlier studies that focus on specific control policies or families of policies, our framework is based on a characterization of feasible plans under all potential control policies: this is the elegant and parsimonious RAN Master Inequality (RMI) (3), that charts the landscape of asymptotically feasible plans. Once desirable operating points are identified, one could employ more refined methods to evaluate or optimize their performance.
- RMI (3) expresses non-linear richness that far exceeds piecewise-linear dynamics, as in the fluid Equations (6.1)-(6.5) of Dai and Harrison (2020). In fact, the RAN framework is rich enough to model systems with complexity far beyond parallel-servers with skills-based-routing of customers (Atar, 2005, Mandelbaum and Stolyar, 2004, Chen et al., 2020), namely one-to-one matching between a waiting customer and an idle server. Specifically, activities could engage a set of resources, or resources could be simultaneously engaged in a set of multiple activities, and those sets or activity-resource correspondence could vary in time; this requires a collaboration of resources (Gurvich and van Mieghem 2015, 2018) as well as continuous dynamic synchronization.
- The RAN framework gives rise to novel approaches, insights and generalizations to classical concepts, with many driven by RANs symmetric resource-view (Figure 1). Examples include fluid models with activity durations that are characterized by cumulative distribution functions, and which can be "long" relative to idle times (Figure 2); time-varying sets of bottleneck associated with activities §3.3.2; offered plans §3.3.1, that generalize offered-loads or offered-capacities in the case of customers or servers, respectively; plans that are non-idling (work-conserving) or maximal (21); and specific designs (e.g. fork-join §5.2), controls (e.g. join-the-shortest-queue §4.5), and behaviors (e.g. abandonment §5.3). RANs are also flexible enough to accommodate mathematical extensions such as the following: single-resources §3.2, hence SPNs are special cases of RANs; long-run (steady-state) dynamics §5.1; measure-valued dynamics §2.2.4; resources (rather than networks) that are open or closed §2.2.6; and a wide variety, including some new, Skorohod problems §4.2.2, or more generally complementarity problems §3.4.

1.4. Organization

The rest of the paper is organized as follows. We start, in Section 2, by introducing the RAN framework, its building blocks, the RAN Master Inequality (RMI) (3), various basic RAN properties, and a description of pre-limits stochastic RANs. Then, in Section 3, we present additional analyses that can be carried out within the RAN framework, in particular static plans, single-resources (as in SPNs), bottlenecks and non-idleness. In Section 4, we introduce, as illustrative examples, RAN models of some classical and modern queueing networks; then, to appreciated RANs data-based potential, we describe how RAN has supported the management of online order-fulfillment by hundreds of robots §4.6. Directions for extending the basic RAN model are outlined in Section 5, which include Fork-Join constructs and abandonments. Future research is described in Section 6. Finally, the paper concludes with appendices, notably Appendix A that summarizes the historical evolution of relevant fluid research, and Appendix C that expands on how the parsimonious RMI in fact implies a measure-valued state-descriptor of a RAN.

1.5. Common Notation

Denote by D^m the space of all $\mathbb{R}^m$-valued functions on $[0,\infty)$ that are right-continuous with left limits (RCLL); for $x \in D^m$, let $\Delta x(t) := x(t) - x(t-)$ be its jump-size at time t. Define two subsets of D^m: $D^m_+ = \{x \in D^m : x_i \geq 0\}$ is the space of such functions that are non-negative, and $\bar{D}^m_\infty = \{x \in D^m_\uparrow : \lim_{t\to\infty} \frac{1}{t} x(t) \text{ exists}\}$ consists of functions that are asymptoticlaly linear "around infinity". For two functions $x, y \in D^1$, let $x \succeq y$ indicate that $(x-y)$ is a non-decreasing function in D^1. Then $D^m_\uparrow = \{x \in D^m_+ : x_i \succeq 0\}$ are functions in D^m_+, the m coordinates of which are one-dimensional non-decreasing non-negative functions. For $x, y \in D^m$, let $x * y = \{(x * y)(t),\, t \geq 0\} := (x_1 * y_1, \ldots, x_m * y_m)^T \in D^m$ be the multi-dimensional convolution operator, in which

$$(x_i * y_i)(t) = \int_{[0,t]} x_i(t-u)\, \mathrm{d}y_i(u), \quad i = 1, \ldots, m;$$

note that $\int_{[0,t]} \mathrm{d}y(u) = y(t)$, since the integral covers also $t = 0$. For a cumulative distribution function (cdf) G, let $\bar{G} := 1 - G \in D^1$ be its survival function. We then denote by G^m_* the operator of m-fold convolution, which could be defined recursively: $G^m_* * A := G^{m-1}_* * (G * A)$, with G^0 being the identity operator. The matrix $I_{K\times K}$ is the $K \times K$ identity matrix (we omit K when dimension is clear by context). Finally, for a set S, denote by $|S|$ its cardinality; and by $\mathbb{1}\{S\}$ its indicator function: $\mathbb{1}\{S\}(u) = 1$ if $u \in S$, and $\mathbb{1}\{S\}(u) = 0$ if $u \notin S$.

2. RAN Model

In this section, we introduce the RAN framework and its building blocks. In parallel, we make it all concrete via a model of an Emergency Department (ED), which is a complex service-system that has been operationally (and compromisingly) modelled as a queueing-network.

2.1. Building Blocks: Resources, Sub-Resources, Activities

A RAN model comprises K different *resource pools* and J different types of *activities*; resource *units* are dynamically *engaged* in activities, and the RAN is a dynamic model of these engagement processes (recall Figure 1). For example, in an ED, patients, nurses, doctors, X-ray machines and technicians, beds and rooms are all considered resource units. All ED nurses in a given shift constitute a resource pool, while the ED doctors during that shift could form another resource pool. Administrative reception, doctor exam, X-ray test, etc., are all considered activities. An ED RAN could then model patient flows, nurse workflows during a shift, etc.

2.1.1. Resources and Sub-Resources. A unit of resource can be in several *states* (one at a time). The sub-pool of units, of a specific resource at a specific state, or the pair *resource-state*, is formalized through the notion of a *sub-resource*. Any two units in a specific resource pool (similarly in a specific sub-resource pool) are interchangeable; and one unit of resource corresponds to, or is *involved* in exactly one unit of sub-resource at a time (this will be relaxed in Section 5.2).

Terminology-wise, a unit from "resource pool k" ("sub-resource pool l") will be mostly referred to as a unit of "resource k" (unit of "sub-resource l"). Then, a resource is *involved* in its sub-resources or, equivalently, a sub-resource involves its resource; similarly, a resource is *engaged* in activities or, equivalently, an activity engages resources.

Note that, in our formal RAN model, the notion of *state* was left vague; this is intentional as it allows a flexible verbal state-description. Instead, a matrix R will be introduced (momentarily), that articulates mathematically the relationships between resources and sub-resources. Sub-resources hence capture, through their state, the flow of resources through activities in the system. In addition to their internal flow within the system, sub-resources can enter and leave the system *spontaneously*, e.g., due to some external reason, regardless of system dynamics.

Formally, let L be the number of *sub-resource pools* in the system; L is at most the product of the number of resources by the number of states (not all resources can be in all states), which is assumed finite. Then R is a $K \times L$ matrix with values in $\{0, 1\}$: $R_{k,l} = 1$ whenever sub-resource l

involves resource k, otherwise $R_{k,l} = 0$. We assume that each resource is involved in at least one sub-resource, and that each sub-resource involves exactly one resource: $\sum_l R_{k,l} \geq 1$, for all k, and $\sum_k R_{k,l} = 1$, for all l. Note that the assumptions on R imply that $K \leq L$, and that $I_{K\times K}$ is a sub-matrix of R.

It is useful to define a map $r : \{1, \ldots, L\} \to \{1, \ldots, K\}$, such that $r(l)$ is the resource that involves sub-resource l, that is, $R_{r(l),l} = 1$. Then r has an inverse r^{-1}, where $r^{-1}(k)$ is the set of all sub-resources involved in resource k.

Let $\Lambda = \{\Lambda(t), t \geq 0\}$ be an L-dimensional function: each coordinate $\Lambda_l = \{\Lambda_l(t), t \geq 0\}$ is a function of bounded variation over finite intervals, which has the form $\Lambda_l(t) = \Lambda_l^+(t) - \Lambda_l^-(t)$, $t \geq 0$. Here $\Lambda_l^+ = \{\Lambda_l^+(t), t \geq 0\}$ and $\Lambda_l^- = \{\Lambda_l^-(t), t \geq 0\}$ are both in $D_\uparrow^1$, and they capture the cumulative processes of sub-resources (and hence also resources) that *arrive* to and *depart* from the system, respectively. Note that, as a difference between two flows, Λ is considered a *quantity* (negative values could be interpreted as backlogs); however, if this difference is monotone, it can also be considered as a *flow*: inflow if increasing or outflow if decreasing.

For example, in an ED, a patient is considered a unit of resource from the resource pool *"patients"*; and during a visit in an ED, patients may reside in different states that depend on their ED journey. Concretely, a *"patient after triage"*, thought of as a resource-state pair, is a sub-resource; one particular *patient after triage* is a unit from that sub-resource pool; and there could be several such units: their corresponding patients, constituting a sub-pool of the resource pool *"patients"*, are all patients right after triage. Similarly, we have units of sub-resources *"patient before reception"*, *"patient after blood-test"*, and so on. In analogy to patients, ED doctors may also be in different states. Therefore, there may be several sub-resources that involve the resource *"doctors"*, and there are units of these sub-resources, namely specific doctors that are at the corresponding states: e.g., *"available doctor"*, *"doctor occupied in interpreting patients' imaging results"*, *"doctor in hospital wards"*, etc. Let the resource *"doctor"* be denoted by k, and let the sub-resource *"available doctor"* be denoted by l; then $R_{k,l} = 1$. Doctors enter and leave the ED according to their shifts. Their entrance and departure processes are thus given by Λ_l^+ and Λ_l^-, respectively: $\Lambda_l^+(t)$ is the total number of doctors that entered the ED and started their shift during $[0, t]$, and $\Lambda_l^-(t)$ is the total number of doctors that ended their shift and left the ED during that time.

2.1.2. Activities. Every activity requires at least one sub-resource in order to take place; and, once completed, it releases (one or more) sub-resources (which may be different from the input sub-resources). We thus say that activities *consume* and *produce* sub-resources, and formalize it via a consumption matrix C and a production matrix P. The defining characteristic

of an activity is that the resources engaged in it are *unavailable* for other activities, until the activity is completed. Activities are hence considered actions that occupy sub-resource units, and hence indirectly resource units, for some non-negative duration – these activity durations are characterized by proper cumulative distribution functions (cdf's).

Formally, let $G = \{G(t), t \geq 0\}$ be a J-dimensional function, with coordinates indexed by j: $G_j = \{G_j(t), t \geq 0\}$ is the cdf of type j activity duration (non-negative). Denote by σ_j a generic random variable representing the duration of activity j, and let $\bar{G}_j := 1 - G_j$ be the corresponding survival function.

Let C be an $L \times J$ non-negative consumption (input) matrix: the element $C_{l,j}$ is the amount of sub-resource l consumed by (required for) a single type j activity; $C_{l,j} \geq 0$. In the same manner, let P be an $L \times J$ non-negative production (output) matrix: the element $P_{l,j}$ is the amount of sub-resource l produced upon completion of a single type j activity, and that *remains* in the system (immediately routed back); $P_{l,j} \geq 0$.

Canonical forms for C, P, R: It is sometimes useful to renumber resources and sub-resources so that R is a block-diagonal matrix, with the following structure: in row $k = 1$, the first $r^{-1}(1)$ elements are 1 and the rest are 0; in row 2, the first $r^{-1}(1)$ elements are 0, the next $r^{-1}(2)$ elements are 1, and the rest are 0; etc. Such an arrangement of resources and sub-resources endows C and P with a column-block structure: their top block, of order $r^{-1}(1) \times J$, has consumption (production) amounts associated with resource $k = 1$; the block below it, of order $r^{-1}(2) \times J$, is associated with $k = 2$, etc.

2.1.3. Relating Activities and Resources via Sub-Resources. Every unit of activity j consumes (produces) an amount $[RC]_{k,j}$ ($[RP]_{k,j}$) of resource k. Activities could naturally create resources: for example, a fork-join activity can produce several "copies" of the same resource (see Section 5.2). However, in most RAN applications, activities conserve resources, at the activity-resource level, in that the amount of resource *produced* and routed back to the network is no more than the amount of resource *consumed*. Therefore, such resource-conservation will form a standing assumption in what follows (with its exceptions noted explicitly). Formally:

ASSUMPTION 1. **Activities do not create resources:** $\boxed{RC \geq RP.}$
One can equivalently assume that activities only change states of resources. Formally, there exists a non-negative $L \times J$-matrix P^- such that $RC = R[P + P^-]$. The element $P^-_{l,j}$ is the amount of sub-resource l that leaves the system, once produced by (per) one unit of activity j. ($I_{K \times K}$ is a sub-matrix of R hence the equivalence.) ▲

For example, in an ED, the activity *doctor exam* (denoted by j) consumes one unit of sub-resource from type *"available doctor"* (denoted by l_1) and one unit of sub-resource from type *"patient after nurse admission"* (denoted by l_2). Assuming it requires no additional resources: $C_{l_1,j}=1$ and $C_{l_2,j}=1$, while the rest of the entries in C's jth column are 0. Once completed, activity j produces one unit of sub-resource from type *"available doctor"* that remains in the system, namely becomes available for the next patient; hence $P_{l_1,j}=1$. It also produces sub-resources involving the resource "patient", specifically fractions of a patient-unit that are routing proportions, after the exam, to various additional activities within the ED; these form the jth column of P, so that $\sum_{l=1}^{L} P_{l,j} \le 1$; with $1-\sum_{l=1}^{L} P_{l,j}$ being the fraction of patients leaving the ED after the exam, e.g. hospitalized or released. Going beyond classical queueing networks, the activity *X-ray exam* would require (consume) one sub-resource *"patient with X-ray order"*, one *"available X-ray machine"* and one *"available X-ray technician"*.

2.2. Feasible Plans: The RAN Master Inequality (RMI)

A *plan* $X=(X_j)\in D_{\uparrow}^{J}$ models a policy/scheme for operating RAN's originating system: it is a J-dimensional function of which $X_j(t)$, its jth coordinate at time t, represents the total amount (number) of activity j that *started* during the time interval $[0,t]$. (We shall sometimes emphasize that X is a *dynamic* plan, mainly to distinguish it from static plans that will be introduced later.) By definition, X_j is non-decreasing and right-continuous-with-left-limits (RCLL); it is convenient to assume $X(t)=0$, for $t<0$, noting that positive jumps of X_j's at $t=0$ are allowed. Indeed, we use *cumulative* number of started activities, rather than the (prevalent) instantaneous rate at which activities start, in order to accommodate discontinuous X: for activity j, a positive jump at t means that $X_j(t)-X_j(t-)>0$ activities j started at time t.

REMARK 1 (**On activities that started before $t=0$**). A plan X starts counting activity initiations from and including time 0. It turns out useful to allow activities that are already *in-progress* at time 0, which means that, in fact, they started *before* time 0. One must then account for these activities when describing the system-state at time 0. However, to facilitate model assimilation, we shall first assume that there are no activities in-progress at time 0; we shall later remove this assumption in Section 5.1. ▲

If $X_j(t)$ activities started during the time interval $[0,t]$, $t\ge 0$, then $(G*X)_j(t)$ of them ended by time t, where

$$(G*X)_j(t) := \int_{[0,t]} X_j(t-u)\,\mathrm{d}G_j(u) = \int_{[0,t]} G_j(t-u)\,\mathrm{d}X_j(u), \quad t\ge 0, \tag{1}$$

is convolution in terms of Lebesque-Stieltjes integrals. (Note that $\int_{[0,t]} \mathrm{d}X_j(u) = X(t) - X(0-) = X(t)$, $t \geq 0$.) Now, $X \in D_{\uparrow}^J$ implies $X \geq G * X = \{G * X(t),\, t \geq 0\} \in D_{\uparrow}^J$, as should be. It follows that the total amount of activities in progress (ongoing) at time t is the vector $(\bar{G} * X)(t)$; or, as a process: $\bar{G} * X = \{\bar{G} * X(t),\, t \geq 0\} \in D_{+}^J$ (more on that momentarily).

2.2.1. RAN Master Inequality (RMI). A plan $X \in D_{\uparrow}^J$ is *feasible* on $[0, T]$ if

$$\sum_{j=1}^{J} C_{l,j} X_j(t) \leq \sum_{j=1}^{J} P_{l,j} \left(G_j * X_j\right)(t) + \Lambda_l(t), \tag{2}$$

for all sub-resources $l = 1, ..., L$, and at all times $t \in [0, T]$; in matrix form:

$$\boxed{\text{RMI:} \qquad CX \leq P\, G * X + \Lambda.} \tag{3}$$

REMARK 2 (**On RMI notation**). The operation $*$ should be applied first, resulting in the column vector $G * X := (G_1 * X_1, \ldots, G_J * X_J)^\top$, which is then multiplied from the left by the matrix P. One could also interpret G as the matrix $\mathrm{diag}(G_1, ..., G_J)$, which results in the same $PG * X$, also if PG is performed first. ▲

We emphasize that RMI is at the sub-resources level (L inequalities between processes), and recall that sub-resources represent resources at specific states. RMI inequalities (2) can be interpreted both as *flow*- and *count*-inequalities. Starting with flow, RMI inequalities constrain the flow of resources through activities at all states: the left-hand side of (2) represents the cumulative amount of sub-resource l consumed by activities during $[0, t]$, which should not exceed the right-hand side, namely the amount of available sub-resource l. The latter constitutes the sum of two elements: first, the cumulative amount of sub-resource l that is endogenously produced by plan X during $[0, t]$, formally $[P\,(G * X)]_l(t)$; plus the net amount of sub-resource l within the system that is due to exogenous supply during $[0, t]$, namely $\Lambda_l(t) = \Lambda_l^+(t) - \Lambda_l^-(t)$ — see Figure 3. (Recall that $\Lambda_l^+(t)$ stands for the cumulative number of exogenous *arrivals* of sub-resource l, while $\Lambda_l^-(t)$ is the cumulative number of sub-resource l that *spontaneously exits* the system (not due to endogenous dynamics), both during $[0, t]$.)

However, RMI can be also interpreted as count inequalities since RMI slacks, namely the difference between the right- and left-hand sides of (2), represents the amount of sub-resources that are available to be engaged in activities. Thus, interpreted as either flow- or count-inequalities, RMI ensures that an activity can start only if there is a sufficient amount of all the sub-resources that it must consume.

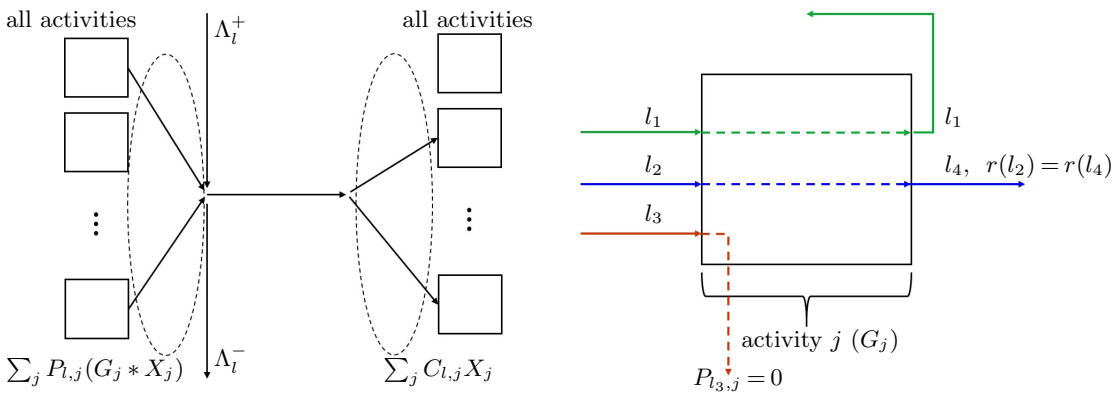


Figure 3 An example of the flow of sub-resources: exogenously, in-between activities and within activities.

REMARK 3 (**Linear parsimonious structure**). It is significant that RMI is, in fact, a parsimonious system of linear constraints in X, with the additional conic constraint $X \in D_\uparrow^J$. This renders feasible – modelling-wise, computationally and statistically – the process of starting with data from a complex system, then estimating appropriate C, P, Λ to create a computationally-tractable model, and ultimately applying the model in various manners. The linear structure will be further pursued in Section 3.3. ▲

REMARK 4 (**Finite time-horizon**). Note that RMI is formulated over finite horizons $[0,T]$, while plans are defined over $[0,\infty)$. Requiring feasibility over a finite horizon could be for practical reasons – e.g. only single-day modelling is relevant – in which case plans can be made flat beyond the horizon; it could be also for a mathematical reason, because the set of plans might become trivial (constituting only $X \equiv 0$) when requiring feasibility over $[0,\infty)$. (In general, the set of feasible plans is non-increasing in the length of the interval T.)

For an example, consider the following single-activity, single-resource RAN: $C = P = R = 1$; $\Lambda(t) = e^{-2t}$, $G(t) = 1 - e^{-t}$, for $t \geq 0$. Then, for a fixed horizon T, RMI amounts to $\bar{G} * X(t) = e^{-t} \int_0^t e^u \, \mathrm{d}X(u) \leq e^{-2t}$, over $t \in [0,T]$; and non-zero feasible plans X clearly exist. However, RMI over $[0,\infty)$ yields $e^{-2t} \geq e^{-t}X(t)$; hence $\lim_{t\to\infty} X(t) = 0$, implying that $X \equiv 0$ since $X \in D_\uparrow^J$. In words, RMI over $[0,\infty)$ requires completion of *all* activities ever initiated, which is infeasible since resource capacity depletes too fast (in the sense that the right tail of Λ decreases faster than that of G). ▲

The parsimonious RMI, jointly with its plan X, in fact characterize RAN dynamics at relatively high resolutions. This will be now demonstrated by first introducing RMI slacks in §2.2.2 – §2.2.3, and then using these slacks to develop refined measure-valued RAN dynamics in §2.2.4.

2.2.2. Busy Process: Ongoing vs. Completed Activities. Fix a plan $X \succeq 0$. Let $B(t)$ be the J-vector recording the amounts of *busy (ongoing) activities* at time $t \geq 0$. Then

$$B := X - G * X = \bar{G} * X = [I - G] * X, \tag{4}$$

which is to be interpreted coordinate-wise: for example, $B_j(t) = \bar{G}_j * X_j(t) = X_j(t) - G_j * X_j(t)$, where $G_j * X_j(t)$ is the cumulative amount of completed activities j during $[0,t]$ (1); and $[I - G]_j = I - G_j$ is coordinate j of the operator $I - G : D^J \to D^J$.

Exhaustive Busy Processes. Implicit in X and $G * X$ is the fact that an activity that started must be carried out to completion, without interruption. From this emerges the converse question that will arise again in the sequel (e.g. staffing $\mathrm{G}_t/\mathrm{G}/s_t$, in Appendix D): characterize the processes $B \in D^J_+$ that embody only complete uninterrupted activities (Liu and Whitt 2012, Ingolfsson et al. 2007). We now demonstrate that the natural mathematical environment for casting this question is Renewal Theory (e.g. Serfozo (2009), §2.4–2.7).

View $X = B + G * X$ as the renewal equation with data B and unknown X. Its solution is $X = [I - G]^{-1} * B$, again interpreted coordinate-wise, where $[I - G]^{-1}_j := [I - G_j]^{-1}$ is the renewal function, or operator, corresponding to cdf G_j. We call $B \geq 0$ *exhaustive* if $[I - G_j]^{-1} * B_j \succeq 0$, for all activities j; then $B_j = [I - G_j] * A_j = A_j - G_j * A_j$, for a uniquely determined $A_j \succeq 0$, which reveals two facts: $A_j(t)$ is the cumulative number of j-activities embodied in B that started by time t; and $G_j * A_j$ counts activity completions, which occur "exactly G_j" after starting. It follows that the renewal operator $[I - G_j]^{-1}$, when applied to exhaustive B_j, is extracting exactly the starts of activities embodied in B_j, namely A_j (or X_j for RANs).

More on Renewal Operators. The renewal operator can be represented as an infinite "geometric" sum $[I - G]^{-1} = I + G + G^2_* + G^3_* + \ldots$, which adds the following intuition to the above (subscript j is omitted for simplicity):

$$(I + G + G^2_* + \ldots + G^n_*) * B = \sum_{m=0}^{n} [G^m_* * A - G * (G^m_* * A)] = A - G^{n+1}_* * A.$$

In the last sum, each $G^m_* * A - G * (G^m_* * A) = G^m_* * B$ is exhaustive since $G^m_* * A \succeq 0$; in fact, it constitutes a right-shift of B by m iid activity durations. This is also the case for the left-hand and right-hand sides: the left is derived from B via operations that are exhaustive-preserving (sums, time-shifts by activity duration); and the right equals a sum in which activity completions telescope with successive activity starts. Observe that both the left- and right-hand sides are monotonic in n, and that $G^{n+1}_* * A \to 0$, as $n \uparrow \infty$, due to $G(0) < 1$ and the SLLN. Letting $n \uparrow \infty$ on both sides yields $[I - G]^{-1} * B = A$, i.e., the renewal operator indeed extracts activity starts from busy processes. Exhaustive $B \geq 0$ are precisely the functions that the renewal operator

maps to non-decreasing functions. These include $B \succeq 0$ since then $G_*^n * B \succeq 0$, and $[I-G]^{-1} = I + G + G_*^2 + G_*^3 + \ldots$. However, exhaustive B's can have points of decrease. For example, if G is exponential with rate μ, then $[I-G]^{-1} * B(t) = B(t) + \mu \int_0^t B(u)\,\mathrm{d}u$, corresponding to the renewal function $1+\mu t$, $t \geq 0$; then $[I-G]^{-1} * B \succeq 0$ if and only if $e^{\mu t}B(t), t \geq 0$, is non-decreasing. (Dermitzakis and Politis (2022) develops sufficient conditions for monotonicity, if B is differentiable.) Explicitly characterizing the exhaustive B's, for a given cdf G, poses an open problem.

2.2.3. Slacks: Idling vs. Active Resources, Decoupled vs. Coupled Flows. Under a concrete plan and at any time, some units of a specific sub-resource could be idle; by this we mean that these units are engaged in no activities; hence the involved resource is not utilized to its capacity. (Amounts of sub-resource idleness are quantified by RMI slacks, as will be formalized momentarily.) In that vein, an *activity is idle* if all the sub-resources that it consumes are idle; and a *plan is idle* if at least one activity is idle.

Consider, for example, servers and customers in a specific station of a queueing-network: some *"servers"* could be idle (waiting for a customer); and for *"customers"*, being idle means that they are queueing, or waiting, for a server (either by plan, or by necessity due to scarce resources). Thus, symmetrically, both customers and servers can be involved in sub-resources that are idle, namely engaged in no activities.

To be formal, let $Q(t)$ be an L-vector representing the amounts of *idle sub-resources* (not engaged in ongoing activities) at time t. Then, at all $t \geq 0$, $Q(t) = (\Lambda + P\,G * X - CX)\,(t)$, and in compact vector form:

$$Q := \Lambda + P\,G * X - CX, \tag{5}$$

where individual coordinates of Q are one-dimensional functions: jointly, they form a RAN state-description that is determined by X (as in Reed (2009), Puhalskii and Reed (2010)).

The slack process Q, counting the total number of idle sub-resources, is the difference between two *flow processes*: $(\Lambda + P\,G * X)$, namely the accumulated amounts of sub-resources produced or turning idle; and CX, the accumulated amounts consumed or becoming 'busy'. Analogously, B in (4), that counts the number of busy activities, is the difference between flow X (cumulative number of started activities) and flow $G * X$ (completed activities).

Both process Q and B are *netflow* processes, in the sense that they are differences between an *inflow* and an *outflow* process (recall Figure 2). They are non-negative for *different* reasons ($B \geq 0$ hold mathematically for all plans $X \succeq 0$, while $Q \geq 0$ is a constraint on X, namely RMI). Their non-negativity, however, models (mathematically articulates) the *same* netflow-physics, which we now explain in a broader context.

REMARK 5 (**On slacks and elapsed time between inflows and outflows**). Introduce two partial orders between functions $A, E \in D^1_\uparrow$: $A \geq E$ iff $A - E \geq 0$, and $A \succeq E$ iff $A - E \succeq 0$. Now view A as inflow to and E as outflow from some buffer ($A - E$ is netflow, or buffer-contents). Then $A \geq E$ means that an outflow/increase in E at time t can be supported by inflow/increases in A at times *prior* to t: E is decoupled from A; hence slack $A - E$ can be *stored* for a later use. (e.g. in B above, activity-completions follow their starts, after durations that are G-distributed; and $Q \geq 0$ ensures consumption of only the earlier-produced.)

On the other hand, $A \succeq E$ couples change in E with those of A: outflow E at time t must be supported by inflow A at the same time t; hence slack $A - E$ can be viewed as production *lost* (perished) unless consumed immediately upon becoming available (positive). This interpretation becomes obvious when A and E have derivatives $\dot{A}$ and $\dot{E}$; then $A \succeq E$ iff $\dot{A} \geq \dot{E}$, namely A and E are coupled via $\dot{A}(t) \geq \dot{E}(t)$, at all $t \geq 0$. (The partial order $\succeq$, which is stronger than $\geq$, will arise naturally in Section 3.2 – see Remark 9 – in a context related to conventional heavy-traffic and SPNs; there units of potential service-time are lost due to idling servers). As a final observation, allowing $A - E$ to become negative reflects backlog. ▲

REMARK 6 (**Refining protocols of resource engagement**). In anticipation of the next subsection, we note that Q counts idleness at an aggregated level which, for some purposes, must be refined. We shall do that by specifying some order in resource engagement, an example of which is the following Longest-Idle-First (LIF) policy: a starting activity will consume the sub-resources, from each required type, that have been idle for the longest time (within their type). Departures due to Λ^- could also occur based on the LIF policy. For example, if resources becoming idle are customers joining a queue then LIF is FCFS; and if they are servers becoming idle, it is longest-idle-first-serve. ▲

2.2.4. Measure-Valued (2D) RAN Dynamics. Both B and Q record, at each time $t \geq 0$, the total *number* of entities (activities and sub-resources, respectively) that 'started' a phase but have not yet 'completed' it. Of interest are also refined counts that correspond to the duration that activities have already been ongoing (e.g. time in service); or that sub-resources have been or will remain idle (e.g. servers). Such refinements, which we now derive, constitute 2-dimensional, or equivalently measure-valued, descriptions of RAN dynamics (in analogy to G/GI/s+GI, as modeled in Liu and Whitt (2012), Zhang (2013), Kang and Pang (2013), Kaspi and Ramanan (2011).

We start with pointing to a fundamental difference between B and Q. Specifically, duration of activities in B are 'known', albeit statistically, in that B averages out the fact that

'activity starts' (determined by X) trigger corresponding 'activity completions' ($G*X$) after a G-distributed duration. With Q, on the other hand, counting is too aggregated to determine, for specific sub-resources, how long these have been or will be idle (disengaged from activities). For example, Figure 2 is insufficient to disclose, as is, the duration between arrivals-to and departures-from queue (waiting time). Thus refinements are required to specify a correspondence between 'starts' and their 'completions', say via some policy that specifies an order-of-engagement of sub-resources in activities (e.g. LIF mentioned above, which is FCFS in the case of Figure 2).

Our first example of a measure-valued (or 2-dimensional) RAN process is $B^{\leftarrow}_t(\le u)$, $t \ge 0$: it stands for the number of ongoing *activities* at time $t \ge 0$, whose activity duration thus far is no more than $u \ge 0$. Its 2-dimensional representation is given by

$$B^{\leftarrow}_t(\le u) = \int_{[t-u,t]} \bar{G}(t-v)\, \mathrm{d}X(v),$$

since, considering an ongoing activity at time t, in order for its duration thus far to not exceed u, it should have started at time $v \in [t-u, t]$.

Now consider the process $Q^{\leftarrow}_t(\le u)$, $t \ge 0$, which stands for the number of delayed sub-resources at time $t \ge 0$, out of $Q(t)$, that have been engaged in no activity for $u \ge 0$ time-units or less (and similarly $Q^{\leftarrow}_t(> u)$). The function $Q^{\leftarrow}_t(\le \cdot)$ determines a measure on $[0, \infty)$, at each $t \ge 0$, with total mass $Q(t) = \lim_{u\to\infty} Q^{\leftarrow}_t(\le u)$. Hence $Q^{\leftarrow}_\cdot$ is a measure-valued (fluid) process; and for its complete characterization, one must specify a priority or a scheduling policy under which sub-resources are being consumed. (This is in contrast to $B^{\leftarrow}_\cdot$, for which X and G suffice.)

Assuming LIF, namely that sub-resources are consumed by activities according to their order of "arrival to delay", the 2-dimensional representations of $Q^{\leftarrow}_t(\le u)$ and $Q^{\leftarrow}_t(> u)$ are given by

$$Q^{\leftarrow}_t(\le u) = A(t) - A(t-u) \vee D(t) \quad \text{and} \quad Q^{\leftarrow}_t(> u) = [A(t-u) - D(t)]^+,$$

where $A := \Lambda^+ + PG * X$ and $D := \Lambda^- + CX$. In particular, $Q^{\leftarrow}_t(\ge 0)_l = Q_t$, and $Q^{\leftarrow}_t(\ge u) = 0$, for $u > t$. Figure 4 is a depiction of these $2D$-representations. In words, if $D_l(t) < A_l(t-u)$ then $A_l(t) - A_l(t-u)$ are waiting less than or exactly u, and $A_l(t-u) - D_l(t)$ are waiting more than u. If $D_l(t) \ge A_l(t-u)$ then no one waits more than u.

All in all, there are eight measure-valued processes associated with B and Q: $B^{\leftarrow}_t(\le u)$, $B^{\leftarrow}_t(> u)$; $Q^{\leftarrow}_t(\le u)$, and $Q^{\leftarrow}_t(> u)$ (out of which three were explained above); and $B^{\rightarrow}_t(\le u)$, $B^{\rightarrow}_t(> u)$; $Q^{\rightarrow}_t(\le u)$, and $Q^{\rightarrow}_t(> u)$. For example, $B^{\rightarrow}_t(\le u)$ is the number of ongoing activities at time t, whose *residual* activity duration (after t, hence $\rightarrow$) is no more than u; and $Q^{\rightarrow}_t(\le u)$ is the number of delayed sub-resources at time t, out of $Q(t)$, whose residual time-in-delay is no more

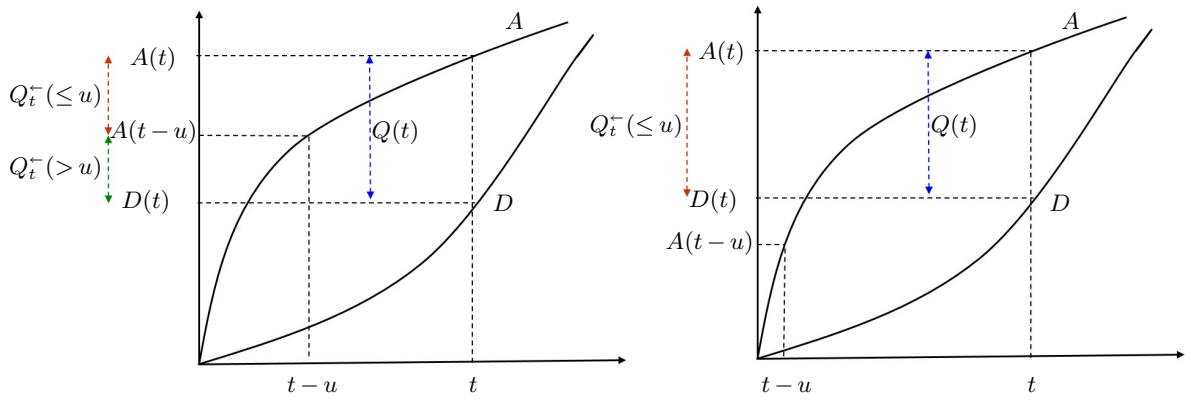


Figure 4 2-dimensional description of Q (for a single sub-resource l); $A \coloneqq \Lambda + PG * X$.

than u. A complete characterization of these eight measure-valued (2D) state-descriptors, of a fluid RAN, is given in Appendix C. Importantly, they are made possible by having cdf's of activity durations as RAN primitives: distributions as inputs enable distributions/measures as output.

2.2.5. Snapshot Inequality. This inequality is an aggregated consequence of RMI at the *resource* level. Nevertheless, it can contribute new information when RMI is taken to its extreme, as will be elaborated on below.

Multiplying by $R \geq 0$ both sides of RMI, then recalling Assumption 1, we get that

$$RC\bar{G} * X = RCX - RCG * X \leq R\Lambda - (RC - RP)G * X = R\Lambda^+ - [R\Lambda^- + RP^-G * X], \qquad (6)$$

at all $t \geq 0$. These are K inequalities at the resource level. In the kth one, the left-hand side represents the amount of resource k that is being used by *ongoing* activities at time t; the right-hand is the total amount of resource k within the system at time t: cumulative inflow of resource k during $[0,t]$, $(R\Lambda^+)_k(t)$, minus cumulative outflow during this interval, either due to plan X, that is $[RP^-G * X]_k(t)$, or spontaneously, $(R\Lambda^-)_k(t)$. For later application, a weaker version of (6) suffices, which is implied by its left- and right-hand sides: at all $t \in [0,T]$,

$$\boxed{\text{Snapshot:} \qquad RC\,\bar{G} * X \leq R\,\Lambda.} \qquad (7)$$

Both inequalities (6)–(7) constrain the amount of resources that are being used at any given time t; hence they will be referred to as *Snapshot Inequalities*. They are at the resource level (K inequalities), which is coarser than the sub-resource level (L, $L \geq K$) of RMI. For systems over finite time intervals in the many-server regime, the snapshot inequality does not contain additional information compared to RMI. However, when considering limiting behavior of RANs

(either in the long-run §3.1, or by operating some resources in single-server regimes §3.2), it turns out that information can be lost by RMI while still being recoverable from the Snapshot Inequalities.

2.2.6. Closed Resources. Recall that $[RC]_{kj}$ ($[RP]_{kj}$) is the amount of resource k that is consumed (produced) by a single unit of activity j. We call resource k *closed* if $\sum_j [RC]_{kj} = \sum_j [RP]_{kj}$. By Assumption 1, this is the same as $[RP^-]_{kj} = 0$, for all activities j. In words, resource k is closed if it does not leave the system through activities. Note that a closed resource k can both arrive or leave the system spontaneously, that is, $[R\Lambda]_k(t)$ can either increase or decrease (respectively $\mathrm{d}[R\Lambda^+]_k(t) > 0$ or $\mathrm{d}[R\Lambda^-]_k(t) > 0$, at some $t \geq 0$).

Resource k is *open* if it is not closed: $[RP^-]_{kj} > 0$, for some j. For example, customers in conventional open queueing networks constitute an open resource: after service, some, but not necessarily all, leave the system. On the other hand, each server-pool is a closed resource: after service, servers remain within the system until their next service; still, their amount can change in time to reflect time-varying staffing – see Section 4.3 for additional details.

2.3. The Universe of Pre-Limits

Fluid RANs, it was argued, have an intrinsic value as *direct* models of their originating *realities*. More traditionally, however, fluid models merely approximate stochastic *models* of these realities, rather than the realities themselves. The originating stochastic models are then *pre-limits*, since fluid models arise as their first-order functional strong limits. We now outline a possible universe of pre-limits for fluid RANs. It will enhance understanding of RANs' modeling role while, at the same time, revealing that a novel asymptotic/mathematical framework is required to accommodate convergence of pre-limit RANs – specifically, random *sets* of feasible plans converging to the RMI (3).

Recall that the primitives of a fluid RAN are $(C, P, R; G, \Lambda)$, and feasible plans are RCLL non-decreasing functions X that adhere to RMI (3). Its pre-limit stochastic RAN will have random consumption (production) with averages C (P), random activity durations with cdf's G; R playing the same role; and a random Λ^η, where $\eta \uparrow \infty$ drives the pre-limit to a *many-"resource"* asymptotic regime. Finally, feasible plans constitute a set of stochastic processes, as will be articulated below.

A Fluid RAN arises from a sequence of systems, indexed by $\eta \uparrow \infty$: P, C, R and G do not change with η, while Λ^η is such that $\eta^{-1}\Lambda^\eta \to \Lambda \in D^L$ a.s., as $\eta \uparrow \infty$; here, each Λ^η is random, but the limit Λ must be deterministic. (Our description of a fluid RAN will be informal in that we avoid rigorous specifications of convergence modes and measurability constraints – more

on that below.) A plan $X = \{X(t) : t \geq 0\}$ is a stochastic process with sample paths in $D^J_\uparrow$, measurable and adapted appropriately. Let $\tau_{j,n} \equiv \tau_{j,n}(X) = \inf\{t \geq 0 : X_j(t) \geq n\}$ and $\sigma_{j,n}$ be the start-time and the duration of the nth activity of type j, respectively. The nth activity of type j, "consumes" $\psi_{l,j,n}$ units of sub-resource l and "produces" $\varphi_{l,j,n}$ units of sub-resource l that remains in the network. It is assumed that $\{\psi_{l,j,n}\}_n$ and $\{\varphi_{l,j,n}\}_n$ are sequences of non-negative random-variables that are iid, with expectations $C_{l,j}$ and $P_{l,j}$ respectively. Let ψ^n and φ^n be $L \times J$-matrices with elements $\{\psi_{l,j,n}\}$ and $\{\varphi_{l,j,n}\}_{l,j}$, respectively; then a pre-limit version of Assumption 1 relates those matrices: $R\psi^n \leq R\varphi^n$ element-wise, for all n.

Example 1 (Consumption and Production Models). Different (joint) distributions of $\{\psi_{l,j,n}, \varphi_{l,j,n}\}_{l,j}$ (for fixed n) correspond to different models. For example, suppose that sub-resources $1, 2, \ldots, 5$ correspond to the same resource and that $[2\ 1\ 0\ 0\ 0\ \ldots]^\top$ and $[0\ 0\ 0.4\ 0.6\ 2\ 0\ \ldots]^\top$ are the jth columns of C and P, respectively (jth activity). In a system with $\mathbb{P}[\psi_{1,j,n} = 2] = \mathbb{P}[\psi_{2,j,n} = 1] = \mathbb{P}[\varphi_{5,j,n} = 2] = 1$, $\mathbb{P}[\varphi_{3,j,n} = 1] = 0.4$ and $\mathbb{P}[\varphi_{4,j,n} = 1] = 0.6$ with $\varphi_{3,j,n}, \varphi_{4,j,n} \in \{0, 1\}$ and $\varphi_{3,j,n} + \varphi_{4,j,n} = 1$, two units of sub-resource 1 and one unit of sub-resource 2 are consumed in a single j activity; every activity completion produces two units of sub-resource 5, and either a unit of sub-resource 3 or a unit of sub-resource 4, with probabilities 0.4 and 0.6, respectively. Alternatively, if $\mathbb{P}[\psi_{1,j,n} = 2] = \mathbb{P}[\psi_{2,j,n} = 1] = 1$, $\mathbb{P}[\varphi_{3,j,n} = 3] = 2/15$, $\mathbb{P}[\varphi_{4,j,n} = 3] = 1/5$ and $\mathbb{P}[\varphi_{5,j,n} = 3] = 2/3$ with $\varphi_{3,j,n}, \varphi_{4,j,n}, \varphi_{5,j,n} \in \{0, 3\}$ and $\varphi_{3,j,n} + \varphi_{4,j,n} + \varphi_{5,j,n} = 3$, then the consumption is as before, but each activity completion produces 3 units – of either sub-resource 3, 4 or 5, with probabilities $2/15$, $1/5$ and $2/3$, respectively. ▲

For system η and horizon T, the set of feasible plans over $[0, T]$ consists of stochastic processes X such that

$$\mathcal{S}^\eta_T := \left\{ X \in D^J_\uparrow[0,T] : \sum_{j=1}^{J} \sum_{n=1}^{\lfloor X_j(\cdot) \rfloor} \psi_{l,j,n} \leq \sum_{j=1}^{J} \sum_{n=1}^{\lfloor X_j(\cdot) \rfloor} \mathbb{1}\{\sigma_{j,n} + \tau_{j,n} \leq \cdot\} \varphi_{l,j,n} + \Lambda^\eta_l(\cdot),\ l = 1, \ldots, L \right\};$$

that is, under $X \in \mathcal{S}^\eta_T$, what can be consumed must not exceed what is available, at any time $t \in [0, T]$. The set $\mathcal{S}^\eta_T$ is random as it is created from realizations of $\{\sigma_{j,n}, \psi_{l,j,n}, \varphi_{l,j,n}\}$. Straightforward algebra yields

$$\mathcal{S}^\eta_T = \left\{ X \in D^J_\uparrow[0,T] : CX \leq P(G * X) + \eta^{-1}\Lambda^\eta + \Delta^\eta(X) \right\},$$

where

$$\Delta^\eta_l(X) := \sum_{j=1}^{J} \left(C_{lj} X_j - \eta^{-1} \sum_{n=1}^{\lfloor \eta X_j(\cdot) \rfloor} \psi_{l,j,n} \right) - \sum_{j=1}^{J} \left(P_{lj} (G * X)_j - \eta^{-1} \sum_{n=1}^{\lfloor \eta X_j(\cdot) \rfloor} \mathbb{1}\{\sigma_{j,n} + \tau_{j,n} \leq \cdot\} \varphi_{l,j,n} \right)$$

and $\tau_{n,j} = \inf\{t \geq 0 : \eta X_j(t) \geq n\}$. Note that, for a fixed $X \in D^J_\uparrow$, the following representation implies that $\Delta^\eta(X)$ vanishes, as $\eta \uparrow \infty$:

$$\begin{aligned}\sum_{n=1}^{\lfloor \eta X_j(\cdot)\rfloor} \mathbb{1}\{\sigma_{j,n} + \tau_{j,n} \leq \cdot\}\varphi_{l,j,n} = &\sum_{n=1}^{\lfloor \eta X_j(\cdot)\rfloor} \mathbb{1}\{\sigma_{j,n} + \tau_{j,n} \leq \cdot\}(\varphi_{l,j,n} - P_{l,j}) \\ &+ P_{l,j} \sum_{n=1}^{\lfloor \eta X_j(\cdot)\rfloor} (\mathbb{1}\{\sigma_{j,n} + \tau_{j,n} \leq \cdot\} - G_j(\cdot - \tau_{j,n})) \\ &+ P_{l,j} \int_0^\cdot G_j(\cdot - u)\, \mathrm{d}\lfloor \eta X_j(u)\rfloor;\end{aligned}$$

the first two sums on the right-hand side are sums of zero-mean random variables, and the last term converges to $P_{l,j}(G * X)_j$ (after dividing by η, and recalling that $\tau_{j,n}$'s are the "jump"-times of ηX_j). Now the road to pre-limit convergence is clear: fix $T > 0$, and apply an appropriate continuous-mapping theorem to the mapping

$$H(\Lambda) := \left\{X \in D^J_\uparrow[0,T] : CX \leq P(G * X) + \Lambda\right\},$$

where H maps elements $\Lambda \in D^J[0,T]$ to subsets of $D^J_\uparrow[0,T]$ (feasible sets). More precisely, defining

$$\Delta^\eta_+ := \sup_{X \in D^J_\uparrow[0,T]} \Delta^\eta(X) \quad \text{and} \quad \Delta^\eta_- := \inf_{X \in D^J_\uparrow[0,T]} \Delta^\eta(X),$$

and noting that H is monotone, yields $H(\eta^{-1}\Lambda^\eta + \Delta^\eta_-) =: \mathcal{S}^\eta_{T-} \subseteq \mathcal{S}^\eta_T \subseteq \mathcal{S}^\eta_{T+} := H(\eta^{-1}\Lambda^\eta + \Delta^\eta_+)$. This establishes strong convergence of our pre-limit, as $\eta \uparrow \infty$:

$$\mathcal{S}^\eta_T = H(\eta^{-1}\Lambda^\eta + \Delta^\eta_\pm) \to H(\Lambda) =: \mathcal{S}_T, \quad \text{a.s.},$$

in which the latter deterministic limit is the feasible set of the limiting Fluid RAN (RMI (3)).

Remark 7 (**Measure-valued representations**). Given $\{\sigma_{j,n}, \psi_{l,j,n}, \varphi_{l,j,n}\}$, a feasible (pre-limit) plan X suffices to characterize all two-dimensional functions in Section 2.2.4 (again, in terms of appropriate continuous mappings). For example, a feasible X defines activity start times $\{\tau_{j,n}\}$ which, in turn, defines quantities such as the number of ongoing activities that started at least a specific number of time units ago. ▲

The above outline, for establishing a.s. convergence of pre-limits to Fluid-RANs, lacks a rigorous framework and some accompanying technicalities. Completing these mathematical details, which is beyond our present scope, will entail two main steps (not unlike Inclusion Theory, e.g. Aubin and Cellina (1984)): first, identifying an appropriate metric for convergence of point-to-set mappings (e.g. Hausdorff metric); then generalizing this deterministic framework to random mappings (e.g. via measurable selections).

3. RAN Analyses

Having established the foundations for the RAN framework, we now turn to some analysis that it enables: introducing static RANs, allowing single-resources, and viewing RANs as linear or complementarity programs in function spaces.

3.1. Static Plans: Long-Run Limits, RMI, Snapshot

A RAN model is characterized by its RMI, which is a mathematical articulation of its dynamics via feasible plans. It is natural, and often useful (e.g. in analyzing long-run trends, specifically stability of the pre-limit, as in Dai et al. (2014)), to associate a feasible *static* plan with a RAN dynamic plan. This can be done in two ways: having the static be a steady-state (stationary version) of the dynamic plan, which we do in Section 5.1; or, letting the static arise as a long-run limit of the dynamic, which we do now. The RAN "language" is that of cumulative amounts, while the natural "language" of static models is *rates*. To this end, we consider now plans in $\bar{D}^J_\infty$, namely plans X for which $\lim_{t\to\infty} \frac{1}{t}X(t) = x \in \mathbb{R}^J_+$ exists. ($x = \dot{X}(\infty)$, when the latter limiting derivative exists.) We also assume finiteness of all a_j, the first moments of G_j, and let $\mathcal{A} := \mathrm{diag}(a_1, ..., a_J)$ denote their corresponding diagonal matrix.

3.1.1. Little's Law, Flow Conservation. Let $X_j \in \bar{D}^1_\infty$ as above. Then the following long-run version of Little's Law prevails, for all activities j: $(\bar{G}_j * X_j)(t) \to a_j x_j$, as $t \uparrow \infty$ which, in matrix-form, is

$$\text{Little's Law:} \qquad \lim_{t\to\infty} (\bar{G} * X)(t) = \mathcal{A}x. \tag{8}$$

This law directly implies that

$$\lim_{t\to\infty} \frac{1}{t}(G * X)(t) = x = \lim_{t\to\infty} \frac{1}{t}X(t), \tag{9}$$

namely, for every activity j, x_j is both its long-run start-rate and completion-rate.

To justify (8), define $\tilde{X}_j = \{\tilde{X}_j(t) = X_j(t) - x_j t,\, t \geq 0\}$, and note that, for $\varepsilon > 0$, there exists T_ε such that

$$|\tilde{X}_j(t)| \leq \begin{cases} \varepsilon t, & t \geq T_\varepsilon, \\ (\varepsilon + x_j)T_\varepsilon, & t \leq T_\varepsilon. \end{cases}$$

This inequality yields the following bound:

$$\begin{aligned} |(\bar{G}_j * X_j)(t) - a_j x_j| &= \left| \int_0^t \bar{G}_j(t-u)\, \mathrm{d}\tilde{X}_j(u) - x_j \int_t^\infty \bar{G}_j(u)\, \mathrm{d}u \right| \\ &= \left| \int_0^{(1-\varepsilon)t} \tilde{X}_j(t-u)\, \mathrm{d}\bar{G}_j(u) \right| + \left| \int_{(1-\varepsilon)t}^t \tilde{X}_j(t-u)\, \mathrm{d}\bar{G}_j(u) \right| + x_j \int_t^\infty \bar{G}_j(u)\, \mathrm{d}u \\ &\leq \varepsilon a_j + (\varepsilon + x_j)\varepsilon t \mathbb{P}[\sigma_j \geq (1-\varepsilon)t] + x_j \int_t^\infty \bar{G}_j(u)\, \mathrm{d}u, \end{aligned}$$

where σ_j is distributed according to G_j. Using Markov's inequality, letting first $t \to \infty$ and then $\varepsilon \to 0$, results in (8).

3.1.2. Static RMI and Snapshot Inequalities. Towards the static RMI, consider RANs with the following limit: $\frac{1}{t}\Lambda(t) \to \lambda \in \mathbb{R}^L$, as $t \uparrow \infty$. For their feasible plans $X \in \bar{D}^J_\infty$, with long-run limits $x \in \mathbb{R}^J_+$, divide both sides of RMI (3) by t, let $t \uparrow \infty$, and apply (9) to get

$$\boxed{\text{Static RMI:} \qquad Cx \leq Px + \lambda, \quad x \in \mathbb{R}^J_+.} \tag{10}$$

These are L *rate*-constraints on $x \in \mathbb{R}^J_+$, at the sub-resource level.

Towards the static Snapshot, consider RANs with the following limit: $\Lambda(t) \to \Lambda(\infty)$, as $t \uparrow \infty$. For $X \in \bar{D}^J_\infty$ as above, and recalling Assumption 1 and Little's Law (8), $t \uparrow \infty$ in the dynamic Snapshot (7) yields the inequality

$$\boxed{\text{Static Snapshot:} \qquad RC\mathcal{A}x \leq R\Lambda(\infty), \quad x \in \mathbb{R}^J_+.} \tag{11}$$

These are K *quantity*-constraints on $x \in \mathbb{R}^J_+$, at the resource level. Note that Assumption 1 implies $R\Lambda(\cdot) \geq 0$, hence also $R\lambda \geq 0$.

3.1.3. Static Plans, as Long-Run Limits. Feasible static plans, unlike dynamic plans, sometimes require *both* the Static RMI and Snapshot inequalities for their characterization. This happens when the Snapshot contributes binding constraints beyond RMI, as we now explain. Assume all that is required for Static RMI and Static Snapshot to prevail (existence of λ and $\Lambda(\infty)$, for which $R\Lambda(\infty) \geq 0$ and $R\lambda \geq 0$ must hold). To facilitate exposition, and without much loss, assume that in fact $\lambda \geq 0$ (no long-run backlog).

Now focus the discussion on a resource k. If $(R\lambda)_k > 0$ then $(R\Lambda)_k(\infty) = \infty$, hence the kth Static Snapshot inquality is redundant. This is also the case when $(R\lambda)_k = 0$ and $(R\Lambda)_k(\infty) = \infty$.

One is left with $(R\Lambda)_k(\infty) < \infty$, and $(R\lambda)_k = 0$: the former statement renders the kth static Snapshot (11) relevant, potentially binding; the latter is the same as $\lambda_l = 0$, for all l such that $r(l) = k$ (since $\lambda \geq 0$) which, we now prove, changes the corresponding static RMI (10) into the equalities $(Cx)_l = (Px)_l$. Denote $v = (Cx) - (Px)$, then the Static RMI implies $v_l \leq 0$, for l with $r(l) = k$, while $(Rv)_k \geq 0$ by Assumption 1, hence v_l vanishes if $r(l) = k$.

Special Cases: The static RAN, corresponding to a closed dynamic RAN (all resources closed), has $RC = RP$. When $\lambda = 0$, it follows that feasible plans require both static RMI and Snapshot for their characterization:

$$\text{Static closed RAN with } \lambda = 0: \quad Cx = Px, \quad RC\mathcal{A}x \leq R\Lambda(\infty).$$

On the other hand, for a RAN with $\lambda > 0$ (which is referred to in the literature as open – it is in fact open on entry), all static Snapshot inequalities are redundant and, hence, it is characterized by only static RMI. Concrete examples of such RANs are Generalized Jackson Networks (GJN), as introduced in Section 4.3. Their static RMI+Snapshot have maximal elements that are, in fact, solutions of their Traffic Equations (Chen and Mandelbaum 1991a).

3.2. Single-Resources in RANs, or Mixed RANs

The discussion and examples in §1.2.1 motivate an extension of RANs that accommodates also single-server "pools". Such RANs, that generalize SPNs, will be called *mixed RANs*, or simply *RANs with single-resources*. Their dynamics, it turns out, must be characterized by *both* RMI and Snapshop – with the latter applied to only closed single-resources. Hence denote by $\mathtt{H}\cup\mathtt{M}$ the single-resources, in which resources are closed in $\mathtt{H}$ and open in $\mathtt{M}$: formally $\sum_j (RC-RP)_{kj} = 0$, for all $k \in \mathtt{H}$ or, equivalently by Assumption 1, $(RC)_{kj} = (RP)_{kj}$, for all $k \in \mathtt{H}$, and all j; and $\sum_j (RC-RP)_{kj} > 0$, for all $k \in \mathtt{M}$ (where $\mathtt{M}$ is analogous to Materials in SPNs – see §3.2.5).

The building blocks of mixed RANs are (C, P, R, G, Λ), as in our regular RAN; plus one additional primitive: $\hat{\Lambda} \in D^{|\mathtt{H}|}$, with coordinates $\hat{\Lambda}_k \in D^1$ that are indexed by $k \in \mathtt{H}$; plus two additional structural constraints:

$$G_j = \mathtt{1}\{\cdot \geq 0\}, \quad \text{if } [RC]_{kj} > 0, \text{ for some } k \in \mathtt{H}\cup\mathtt{M}; \quad \text{and} \quad \Lambda_l = 0, \quad \text{for all } l \in r^{-1}(\mathtt{H}). \tag{12}$$

Thus activities that consume single-resources (indicator G_j's) are instantaneous (of zero-duration). Furthermore, amounts of $\mathtt{H}$ resources are and must be zero; indeed, $\Lambda_l \neq 0$, if $r(l) = k \in \mathtt{H}$, endows k with infinite processing capacity (it can support an unbounded number of activity starts within any finite time interval) and hence can be eliminated from the RAN. It follows that negligible amounts of closed single-resources have non-negligible "processing" power (due to the vanishing activity times). This is captured by $\hat{\Lambda}_k$ ($k \in \mathtt{H}$ is a closed single-resource), which can be interpreted as cumulative "capacity" of resource k; for example, suppose that activity duration are in seconds; then $\hat{\Lambda}_k(t)$ is the number of seconds that one unit of resource k is capable of "processing" during $[0, t]$ (which could also decrease in t).

The mixed RAN is then characterized first by RMI: $CX \leq PG * X + \Lambda$, that is identical to (3), as well as by

$$\boxed{\text{Snapshot for closed single-resources:} \qquad (RC\mathcal{A}X)_k \preceq \hat{\Lambda}_k, \quad k \in \mathtt{H}.} \tag{13}$$

It will also follow that, since G and Λ adhere to (12), the *single-resourced part of RMI* in fact becomes:

$$(CX)_l \le (PX)_l + \Lambda_l, \quad r(l) \in \mathtt{M}, \tag{14}$$

$$(CX)_l = (PX)_l, \qquad r(l) \in \mathtt{H}. \tag{15}$$

The dynamics of mixed RANs will be now derived via a procedure referred to as *single-resourcing*: its starting point is the usual many-resource RAN, in which we drive, asymptotically, the RAN resource in $\mathtt{H}\cup\mathtt{M}$ to conventional heavy-traffic. The outcome of single-resourcing is the above mixed-RAN model.

3.2.1. Single-Resourcing: RMI. We start with a base-case RAN $(C, P, R, \tilde{G}, \tilde{\Lambda})$, in which "tilde" is appended to the primitives that play a role in single-resourcing. Given any resource $k \in \mathtt{H} \cup \mathtt{M}$, we "single-resource" it by decreasing its amount in the system while simultaneously decreasing duration of all activities that involve it; and we do this via scaling that preserves "processing capacity", as will become clear momentarily. Denote by $\xi \uparrow \infty$ the parameter of this (conventional heavy-traffic) scaling. The scaling of single-resourcing is then

$$G^\xi_j(\cdot) = \begin{cases} \tilde{G}_j(\xi\cdot), & j : [RC]_{kj} > 0, \text{ for some } k \in \mathtt{H}\cup\mathtt{M}, \\ \tilde{G}_j(\cdot), & \text{otherwise}, \end{cases} \quad \text{and} \quad \Lambda^\xi_l = \begin{cases} \xi^{-1}\tilde{\Lambda}_l, & r(l) \in \mathtt{H}, \\ \tilde{\Lambda}_l, & \text{otherwise}. \end{cases} \tag{16}$$

Letting $\xi \uparrow \infty$ in RMI (3), with the latter G^ξ and Λ^ξ are used, yields in the limit $CX \le PG * X + \Lambda$, in which G and Λ have the structural properties of a mixed RAN:

$$G_j = \begin{cases} \mathbb{1}\{\cdot \ge 0\}, & j : [RC]_{kj} > 0, \text{ for some } k \in \mathtt{H}\cup\mathtt{M}; \\ \tilde{G}_j, & \text{otherwise}; \end{cases} \quad \text{and} \quad \Lambda_l = \begin{cases} 0, & r(l) \in \mathtt{H}; \\ \tilde{\Lambda}_l, & \text{otherwise}. \end{cases}$$

REMARK 8 (**Scaling intuition**). G^ξ-scaling is dividing service duration by $\xi \uparrow \infty$. For open resources $k \in \mathtt{M}$ and a given plan, this suffices to decrease the total amount of k, engaged in activities and averaged over time, by order ξ. However, the amounts of $k \in \mathtt{H}$, being resources that are closed, are completely determined by their $\tilde{\Lambda}$ (they are unaffected by durations) which, hence, must be reduced directly: we divide these $\tilde{\Lambda}$ by ξ, in order to preserve processing capacity (amount divided by average duration). For more concrete intuition, assume existence of $\dot{X}_j(t)$, then $\xi[X_j(t) - G^\xi_j * X_j(t)] \to a_j \dot{X}_j(t)$, as $\xi \uparrow \infty$; or $B^\xi_j(t) \approx \frac{a_j}{\xi}\dot{X}_j(t)$, in terms of ongoing activities (5); here a_j is the mean duration of activity j (having cdf $\tilde{G}_j$). This reveals that rates-of-change in planning provide the natural language of constraints involving instantaneous activities, which in fact we integrate (recall $\succeq$ in Remark 5) to relax differentiability; and Snapshot will turn out the natural supporting tool, which will culminate in (13) above. ▲

Returning to RMI, it is unchanged for sub-resources that are not consumed by activities-consuming-single-resources. For resources $k \in \mathtt{M} \cup \mathtt{H}$, however, the related activities become instantaneous ($G_j * X_j \equiv X_j$); and $\Lambda_l = 0$, for all $l \in r^{-1}(\mathtt{H})$. The *single-resourced part of RMI* then becomes (14) and (15) above: the latter equality follows from $(CX)_l \leq (PX)_l$, $l \in r^{-1}(\mathtt{H})$ (by taking $\xi \uparrow \infty$ in $(RPG^\xi * X + R\Lambda^\xi)_k \geq (RCX)_k = (RPX)_k$, $k \in \mathtt{H}$; the former inequality suggests that the scaled dynamics of closed resources nullifies their associated RMI constraints: a refined additional version of RMI is hence required to enforce plan-feasibility, which is where Snapshot comes to the rescue.

3.2.2. Single-Resourcing: Snapshot a Must for Closed Resources. Write the pre-limit version of (15) as a Snapshot slack $(R\Lambda^\xi - RC\bar{G}^\xi * X)_k \geq 0$, which (in the ξth system) is the amount of resource k that *idles* at time t: indeed, $(R\Lambda^\xi)_k(t)$ is the total amount of resource k at time t in the system (being closed), while $(RC\bar{G}^\xi * X)_k(t)$ represents the total amount of resource k that is being engaged (by any activity) at time t. Now the amount of that idling slack is order $\frac{1}{\xi}$ (e.g. when $\dot{X}(t)$ exists: $\xi(R\Lambda^\xi(t) - RC\bar{G}^\xi * X(t))_k \to (R\Lambda(t) - RC\mathcal{A}\dot{X}(t))_k$, as $\xi \uparrow \infty$). This explains both how RMI trivializes on $\mathtt{H}$ while, at the same time, also offers its refinement.

We start with an *integrated* form of Snapshot slacks:

$$\left\{ \int_0^t (R\xi\Lambda^\xi(u) - RC\xi(\bar{G}^\xi * X)(u))_k \, \mathrm{d}u, \, t \geq 0 \right\} \succeq 0, \quad k \in \mathtt{H}.$$

Next, for activity j that consumes resource $k \in \mathtt{H}$, the scaling of G_j^ξ yields, for a fixed $t \geq 0$, as $\xi \uparrow \infty$,

$$\begin{aligned} \xi \int_0^t (\bar{G}^\xi * X)_j(u) \, \mathrm{d}u &= \xi \int_0^t \int_z^t \bar{G}_j^\xi(u - z) \, \mathrm{d}u \, \mathrm{d}X_j(z) \\ &= \int_0^t \int_0^{\xi(t-z)} \bar{G}_j(u) \, \mathrm{d}u \, \mathrm{d}X_j(z) \to a_j X_j(t). \end{aligned}$$

Therefore, for *single-resources that are closed*, one supplements (15) with (13), in which

$$\hat{\Lambda}_k = \left\{ \int_0^t (R\tilde{\Lambda})_k(u) \, \mathrm{d}u, \, t \geq 0 \right\}, \quad k \in \mathtt{H}.$$

The functions $\hat{\Lambda}_k$ need be defined only for $k \in \mathtt{H}$, in which case $\tilde{\Lambda}_l \in D^1$ implies that $\hat{\Lambda}_k$ are continuous. This continuity, however, is not assumed in (13), as it can be relaxed by replacing Λ_l^ξ in (16), for $r(l) \in \mathtt{H}$, with

$$\left\{ \int_0^t (R\xi\Lambda^\xi)_k(u) \, \mathrm{d}u, \, t \geq 0 \right\} \to \hat{\Lambda}_k \in D^1,$$

as $\xi \uparrow \infty$, which directly allows $\hat{\Lambda}_k \in D^1$.

3.2.3. Mixed RANs and their Slacks. To summarize, activities involved in single-resources are instantaneous ($\tilde{G}_j * X_j \equiv X_j$). This implies that mixed RANs must be characterized by *both* RMI (with the special structure (14)-(15)) as well as by Snapshot (applied only to closed single-resources). Snapshot is necessary since, without it, there would exist plans that violate constraints on the amounts of single-resources used, while these constraints must prevail since they can be interpreted as limits of RMI constraints that must prevail.

REMARK 9 (**Slacks with single resources – continuing Remark 5**). Slacks for RANs with single resources are of two types: *idling* sub-resources awaiting processing, that are quantified by non-negative functions (analogues of Q's in (5)); and *lost processing capacities* of resources that are closed – these are non-decreasing functions, for which the ground was already prepared in Remark 5. Formally:

$$Q := PG * X + \Lambda - CX \geq 0, \tag{17}$$

$$Y_k := \hat{\Lambda}_k - (RC\mathcal{A}X)_k \succeq 0, \quad k \in \mathtt{H}. \tag{18}$$

Note that, in view of (15), implicit in (17) is that $Q_l \equiv 0$, for $r(l) \in \mathtt{H}$; and since activities involving single resources are instantaneous, their amounts (B's in (5)) trivially vanish at all times. Moreover, "transformations" between sub-resources involving closed resources are instantaneous, hence these sub-resources are tightly coupled. It follows that resource level is the right level to "express" slacks (as in (18)), and slack Y_k quantifies lost processing capacity of resource k (relative to planning), in units of (integrated) rates-of-loss (recall Remark 8). ▲

3.2.4. Static Mixed RANs. Corresponding to our RAN that also has single resources $k \in \mathtt{M} \cup \mathtt{H}$, and assuming existence of λ as in (10), and $\lim_{t\uparrow\infty} \frac{1}{t}\hat{\Lambda}(t) =: \hat{\lambda}$, there is a static RAN that is characterized by its RMI and Snapshot as follows: for $x \in \mathbb{R}^J_+$, one has

- *Static RMI* with single resources: $\boxed{Cx \leq Px + \lambda}$; for the single resources, this RMI takes the form: $(Cx)_l \leq (Px)_l + \lambda_l$, for $r(l) \notin \mathtt{H}$; and $(Cx)_l = (Px)_l$, for $r(l) \in \mathtt{H}$.
- *Static Snapshot* for closed many-resources: $\boxed{(RC\mathcal{A}x)_k \leq R\Lambda(\infty)_k}$, for $k \notin \mathtt{H}$, k closed.
- *Static Snapshot* for closed single-resources: $\boxed{(RC\mathcal{A}x)_k \leq \hat{\lambda}_k}$, for $k \in \mathtt{H}$.

Static RANs are thus characterized by flow constraints (RMI) and capacity constraints (Snapshot) which, in fact, jointly create the feasibility set of a standard Linear Program.

3.2.5. RANs generalizing SPNs via single-resourcing. SPNs consist of "resources" and "materials" that together engage in performing activities. Arising from conventional heavy traffic, they can be viewed as RANs having only single-resources, namely $\mathtt{M} \cup \mathtt{H} = \{1, 2, \ldots, K\}$, where the closed RAN resources $\mathtt{H}$ correspond to SPN resources, and the open RAN Resources $\mathtt{M}$

correspond to SPN Materials (hence the M). In fact, SPNs are only special cases of such RANs, which turns out easiest to explain by comparing against Harrison (2002), specifically static against (2.1) and dynamic against (4.8) (where, for simplicity, we let Harrison's $\mathcal{S}$ be the non-negative orthant). Note that we have been using SPNs to encompass Stochastic Processing Networks but also their static and dynamic fluid models – the latter two will be now compared to our RANs.

Starting dynamic, inequalities (14) and (13) correspond to (4.8) in Harrison (2002), specifically to material flow constraints ($Q(t) \geq 0$) and resources capacity-constraints (*integrated* version of $Ax(t) \leq b$). However, equalities (15), which can be viewed as resources' "itineraries", have no SPN analogues: indeed, resources have no itineraries in SPNs as only materials can change their state in the system. Therefore, already (single-resourced) RANs enrich the modeling scope of SPNs. For example, a Round-Robin policy can be easily modeled by a RAN but not by an SPN (without additional constraints).

Similarly for static, the first and second bullet in the above §3.2.4 (static RMI and Snapshot) correspond to (2.1) in Harrison (2002), respectively to $Ax = \lambda$ (relaxed to less-equal) and $Rx \leq b$. Thus, Harrison's R corresponds to our $RC\mathcal{A}$, his A to our $C - P$, and his b is our $\hat{\lambda}$. Finally recall that Harrison's units are rates and RANs use counts (our x is rates but Ax is counts).

3.3. RANs as Continuous Linear Programs (CLPs)

Static RMIs (10) are feasibility sets of Linear Programs (LPs) §3.2.4. Analogously, RMIs (3) can be viewed as feasibility-sets of Continuous LPs (CLPs), which are LPs in infinite-dimensions (continuous-time function spaces). CLPs originated in Bellman (1953), which was motivated by "bottleneck problems": §1.2.1 of Anderson and Nash (1987) explains why; Anderson and Philpott (1994) adds analysis; and recent research appears in Shindin and Weiss (2015, 2020), Shindin et al. (2021) and references therein. A CLP-view of RANs is useful. We now make this concrete via two examples: first offered-plans §3.3.1 and then bottlenecks §3.3.2. Further advantages will be discussed in §3.3.3, after RANs will be formulated as CLPs.

3.3.1. Offered-Plans (rather than Offered-Loads). Roughly speaking, the offered-load to a service station is the least number of servers that suffices to have all customers be served without wait; less servers will result in queueing/delays, more servers would create idleness. Offered-loads can be calculated as averages in steady-state (then they are scalars), or over time (deterministic functions). One can also define them sample-paths-wise, in which case they are stochastic processes (Reich 2012). In any case, traditional offered-loads are one-dimensional,

yet applications beg for a multi-dimensional notion (e.g. Carmeli (2020)). We now introduce such a notion for RANs, which is natural in view of the symmetry among resources.

For a subset of sub-resources $\mathcal{S} \subseteq \{l : r(l) \notin \mathtt{H}\} \cup \{r^{-1}(k) : k \in \mathtt{H}\}$, consider all feasible plans under the assumption that the amounts of sub-resources not in $\mathcal{S}$ are unbounded, i.e., there are no capacity constraints on sub-resources outside $\mathcal{S}$. Formally, the *offered-plans associated with sub-resources* $\mathcal{S}$, which we also call $\mathcal{S}$-feasible, constitute the following set of dynamic plans:

$$\begin{aligned} CX &\leq PG * X + \Lambda^{\mathcal{S}}, \\ (RC\mathcal{A}X)_k &\preceq \hat{\Lambda}^{\mathcal{S}}_k, \quad k \in \mathtt{H}, \end{aligned}$$

where

$$\Lambda^{\mathcal{S}}_l = \begin{cases} \Lambda_l, & l \in \mathcal{S} \cup r^{-1}(\mathtt{H}), \\ \infty, & \text{otherwise;} \end{cases} \quad \text{and} \quad \hat{\Lambda}^{\mathcal{S}}_k = \begin{cases} \hat{\Lambda}_k, & r^{-1}(k) \in \mathcal{S}, \\ \infty, & \text{otherwise;} \end{cases}$$

here $r^{-1}(\mathtt{H}) := \{l : r(l) \in \mathtt{H}\}$. The *offered-load* corresponding to a specific offered-plan is the set of minimal $\{\Lambda^{\mathcal{S}}_l, l \notin \mathcal{S} \cup r^{-1}(\mathtt{H})\}$ and $\{\hat{\Lambda}^{\mathcal{S}}_k, r^{-1}(k) \notin \mathcal{S}\}$, that leave this offered-plan $\mathcal{S}$-feasible; in fact, these minimal $\Lambda^{\mathcal{S}}_l$'s and $\hat{\Lambda}^{\mathcal{S}}_k$'s are given by $\Lambda^{\mathcal{S}}_l = (CX - PG * X)_l$, $l \notin \mathcal{S} \cup r^{-1}(\mathtt{H})$, and $\hat{\Lambda}^{\mathcal{S}}_k = (RC\mathcal{A}X)_k$, $r^{-1}(k) \notin \mathcal{S}$, respectively. (Note that letting either $\Lambda^{\mathcal{S}}_l = \infty$, for all l such that $r(l) = k$, or $\hat{\Lambda}^{\mathcal{S}}_k = \infty$ for single-resources, effectively creates a RAN without resource k – indeed, a resource must have finite capacity.)

Example 2 (Offered-load and offered-capacity). The conventional offered-load to a $\mathrm{G}_t/\mathrm{G}/s_t$ queue, as in Eick et al. (1993) and Whitt (2018), is the average number of busy servers in a corresponding $\mathrm{G}_t/\mathrm{G}/\infty$. It equals $\bar{G} * A$, with $A(\cdot)$ denoting the function of average cumulative arrivals to the latter two queues; and it plays a central role in applications as the skeleton of time-varying staffing, given arrivals (Whitt 2013). The function $\bar{G} * A$ is also the above offered-load in the $\mathrm{A}_t/\mathrm{G}/s_t$ RAN (Appendix D): it corresponds to plan A, which is the maximum when $\mathcal{S}$ consists of servers only. On the other hand, when $\mathcal{S}$ is only arriving customers, one obtains what can be naturally called *offered-capacity*: it could serve as the skeleton for time-varying appointments, given staffing (see Appendix D for details). ▲

Static offered-loads can be defined in a similar fashion. To this end, consider a static mixed-RAN as in §3.2.4. Then, the static offered-plans, associated with sub-resources $\mathcal{S}$, constitute the following set of static $\mathcal{S}$-feasible plans $x \in \mathbb{R}^J_+$:

$$\begin{aligned} Cx &\leq Px + \lambda^{\mathcal{S}}, \\ (RC\mathcal{A}x)_k &\leq \hat{\lambda}^{\mathcal{S}}_k, \quad k \in \mathtt{H}, \end{aligned}$$

where

$$\lambda_l^{\mathcal{S}} = \begin{cases} \lambda_l, & l \in \mathcal{S} \cup r^{-1}(\mathtt{H}), \\ \infty, & \text{otherwise;} \end{cases} \quad \text{and} \quad \hat{\lambda}_k^{\mathcal{S}} = \begin{cases} \hat{\lambda}_k, & r^{-1}(k) \in \mathcal{S}, \\ \infty, & \text{otherwise.} \end{cases}$$

The static *offered-load* corresponding to a specific static offered-plan is the set of minimal $\{\lambda_l^{\mathcal{S}}, l \notin \mathcal{S} \cup r^{-1}(\mathtt{H})\}$ and $\{\hat{\lambda}_k^{\mathcal{S}}, r^{-1}(k) \notin \mathcal{S}\}$, that leave this offered-plan $\mathcal{S}$-feasible; in fact, these $\lambda_l^{\mathcal{S}}$'s and $\hat{\lambda}_k^{\mathcal{S}}$'s are given by $\lambda_l^{\mathcal{S}} = (Cx - Px)_l$, $l \notin \mathcal{S} \cup r^{-1}(\mathtt{H})$, and $\hat{\lambda}_k^{\mathcal{S}} = (RC\mathcal{A}x)_k$, $r^{-1}(k) \notin \mathcal{S}$, respectively.

3.3.2. Bottleneck Sets. Analysis of bottlenecks has a long and rich history, with the concept being typically associated with obstacles for improving performance. In particular, fluid models determine bottlenecks and operational regimes in their originating pre-limit stochastic systems (Chen and Mandelbaum 1991c, Mandelbaum and Massey 1995). We now offer a particular definition for RANs, in which improving performance means increasing X; it will imply that if X_j can not be increased then a set of sub-resource, consumed by j, must have zero-slack. Formally, a set of sub-resources $\mathcal{B}(t) \subseteq \{1, 2, \ldots, L\}$ is a *bottleneck*, of activity j at time t if

$$\mathcal{B}(t) \subseteq \{l : C_{l,j} > 0, r(l) \notin \mathtt{H}\} \cup \{r^{-1}(k) : [RC]_{k,j} > 0, k \in \mathtt{H}\},$$

and for the slacks Q_l and Y_k in (17)–(18) we have

$$\begin{aligned} Q_l(t) = 0,\ l \in \mathcal{B}(t) \quad &\text{and} \quad Q_l(t) > 0,\ l \notin \mathcal{B}(t) \quad \text{if } C_{l,j} > 0,\ r(l) \notin \mathtt{H}; \\ \mathrm{d}Y_k(t) = 0,\ r^{-1}(k) \in \mathcal{B}(t) \quad &\text{and} \quad \mathrm{d}Y_k(t) > 0,\ r^{-1}(k) \notin \mathcal{B}(t) \quad \text{if } (RC)_{k,j} > 0,\ k \in \mathtt{H}. \end{aligned}$$

Observe that for each closed single-resource $k \in \mathtt{H}$, all or none of its sub-resources $r^{-1}(k)$ are in the bottleneck $\mathcal{B}(t)$. This definition, of $\mathcal{B}(t)$ being a bottleneck of activity j at time t, renders an increase in $X_j(t)$ feasible if and only if one increases *all* Λ's and $\hat{\Lambda}$'s corresponding to sub-resources in $\mathcal{B}(t)$. Let $\mathbb{B}(t) \subseteq 2^{\{1,\ldots,L\}}$ denote the set of bottlenecks at time t (set of sets).

Example 3 (Tandem network with flexible servers). Consider a two-station tandem network. The corresponding RAN consists of two activities $\{1, 2\}$ (services at the two stations), four resources (customers $\{1\}$; two dedicated server pools $\{2, 3\}$ that engage in the two activities, respectively; and one flexible pool $\{4\}$ that can be engaged in either of the activities), and five sub-resources (two sub-resources $\{1, 2\}$ involve customers at the two stations; and similarly $\{3, 4, 5\}$ involve servers). Dynamics is such that, initially, there are customers only at the first

station, with no further external arrivals; and service activity at either station requires 3 sub-resources: one customer plus one dedicated server (e.g. sub-resource 3 for activity 1) plus one flexible server (sub-resource 5). Formally, the model primitives are

$$C=\begin{bmatrix}1&0\\0&1\\1&0\\0&1\\1&1\end{bmatrix},\quad P=\begin{bmatrix}0&0\\1&0\\1&0\\0&1\\1&1\end{bmatrix},\quad R=\begin{bmatrix}1&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix},\quad \Lambda=\begin{bmatrix}4\\0\\2\\1\\2\end{bmatrix},\quad G(t)=\begin{bmatrix}\mathbb{1}\{t\geq 1\}\\1-e^{-t}\end{bmatrix}.$$

The following plan is feasible (and also maximal):

$$X(t)=\begin{bmatrix}2+\mathbb{1}\{t\geq 1\}+\mathbb{1}\{t\geq 2\}\\ \min\{\mathbb{1}\{t\geq 1\}+(t-1)^{+},4\}\end{bmatrix}.$$

Under this plan, all servers corresponding to sub-resources 3 and 5 are busy at the first station for the first time-unit. After that, the flexible servers (sub-resource 5) are divided equally between the two stations. For this particular plan X, we have $\mathbb{B}(t)=\{\{3,5\},\{2,5\}\}$ for $t\in[0,1)$, $\mathbb{B}(t)=\{\{5\},\{4,5\}\}$ for $t\in[1,2)$, $\mathbb{B}(t)=\{\{1,5\},\{4,5\}\}$ for $t\in[2,3)$, $\mathbb{B}(t)=\{\{1\},\{4\}\}$ for for $t\in[3,4)$, and $\mathbb{B}(t)=\{\{1\},\{2\}\}$ for $t\geq 4$.

When the three server pools are single resources, the following plan is feasible (and also maximal):

$$X(t)=\begin{bmatrix}\min\{t,4\}\\ \min\{t,4\}\end{bmatrix}.$$

In this case, $\mathbb{B}(t)=\{\{3,5\},\{2,4,5\}\}$ for $t\in[0,4)$ and $\mathbb{B}(t)=\{\{1\},\{2\}\}$ for $t\geq 4$.

The present example, both with and without single resources, enjoys two features that are not in concert with prevalent views. First, customers as well as servers could be bottlenecks. Second, bottlenecks do constitute sets of multiple sub-resources, which is here due to having flexible servers; with only dedicated servers, however, the bottlenecks would all be singletons. ▲

Remark 10 (**Bottleneck sub-network**). In SPNs, operating under a maximal plan, one can restrict attention to the so-called bottleneck sub-network of servers/activities (e.g. §1.7 in Chen and Mandelbaum (1991a)); note that customers are the bottlenecks in the eliminated part of the considered SPN. A similar sub-network exists in the case of RANs, which we now describe. Suppose RMI has a unique maximal plan X^m. Then, in general, there exists a simpler RMI (RAN) (in the sense that some resources are eliminated from the original model) such that its maximal feasible plan is also X^m (on some time interval). Specifically, as far as X^m is concerned, RMI inequalities are either binding or non-binding (at a given t). If, for non-binding inequalities, the corresponding $\Lambda_l(t)$'s or $\hat{\Lambda}_k(t)$'s are set to infinity, the maximal feasible plan does not change. Hence, if all RMI inequalities corresponding to a resource are non-binding, then this resource

can be "eliminated" from consideration (at that specific t in a given time interval, if true for all time instances within that interval). Equivalently, a resource can be eliminated if none of its sub-resources are part of a bottleneck.

In addition to resources, some activities can be eliminated as well. In general, activities with non-zero duration cannot be eliminated. Among zero-duration activities, one can eliminate an activity iff it consumes only sub-resources in the bottleneck associated with that activity. Such activities become merely flow-routers: they instantaneously transform consumed sub-resources into produced ones. When eliminating these activities, the same procedure (re-calculation of model parameters) used in Queueing Networks should be followed (e.g. §1.7 in Chen and Mandelbaum (1991a)). ▲

We end this "bottleneck section" with a discussion of *static* bottlenecks. Unlike queueing networks, however, where traffic equations determine static bottlenecks (since only servers are treated as bottlenecks), static *general* RANs present challenges that render our discussion incomplete, as an example will momentarily show.

To start, and for a given feasible static plan, one can attempt defining static bottlenecks (sets of sub-resources and single-resources) by the same logic used earlier for the dynamic case. However, two challenges arise. First, in static RANs, there is a fundamental difference between closed and open resources, which are described by counts and rates, respectively. Indeed, for closed resources, the λ_l's in static RMI (10) vanish, and Static Snapshot must hence be used to determine whether slacks are zero for such resources; in contrast, static RMI suffices for open resources. Second, zero slack of a sub-resource does not necessarily indicate that increasing the amount of the corresponding closed resource would lead to higher activity rates. To be concrete, consider a two-activity (exponential unit durations) RAN with three closed resources:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} b \\ b \\ 0 \\ 3b \end{bmatrix}.$$

In this example, increasing Λ_1 without increasing Λ_2 does not yield an increase in the maximum static plan. Similarly, increasing Λ_4 does not result in a higher maximum static plan even when both slacks at activity 1 vanish.

3.3.3. Continuous Linear Programs (CLPs). As mentioned, RMIs are feasibility-sets of CLPs (LPs in spaces of continuous-time functions). More accurately, RMIs conform to existing CLP formulations only under additional assumptions (e.g. existence of some derivatives). Yet relaxing these assumptions, which are apparently technical, reaches beyond our present scope.

(Theories of infinite-dimensional spaces are typically sensitive to such "technicalities," that are in fact mathematically fundamental.) Nevertheless, we do outline here two CLP views of RANs, and we do so for reasons to be elaborated on after the CLPs.

CLP 1: $\dot{X} \geq 0$ ($\mathrm{d}X \geq 0$). RMI (3) is the feasibility set of a (generalized) CLP, namely all $X \in D^J_\uparrow$ that satisfy

$$\int_{[0,t]} K(t-u)\, \mathrm{d}X(u) \leq b(t), \quad t \geq 0; \tag{19}$$

here $b(t) = \Lambda(t)$, and K is a *convolution kernel* of dimension $L \times J$, with components given by $K_{l,j}(t-u) = C_{l,j} - P_{l,j}G_j(t-u)$; or, in matrix form, $K = C - P \times \mathrm{diag}(G)$. To conform to a classical CLP (hence the "generalized"), one restricts attention to plans $X(\cdot)$ with derivative $\dot{X}(\cdot)$ almost everywhere. Then (19) becomes a CLP in $\dot{X} \geq 0$: $\int_{[0,t]} K(t-u)\dot{X}(u)\, \mathrm{d}u \leq b(t)$. ▲

CLP 2: $X \in D^J_\uparrow$. Assume that the cdf's of activity durations, G_j's, are absolutely continuous with densities g_j's. Then RMI (3) is the feasibility set of a (conic) CLP, namely all $X \in D^J_\uparrow$ that satisfy

$$B(t)X(t) + \int_{[0,t]} K(t,u)X(u)\, \mathrm{d}u \leq b(t), \quad t \geq 0. \tag{20}$$

Here $B(t) \equiv C$, $b(t) = \Lambda(t)$; and K is a convolution kernel of dimension $L \times J$, with components $k_{l,j}(t,u) = -P_{l,j}g_j(t-u)$, that arises after integration-by-parts. The constraint $X \in D^J_\uparrow$ renders (20) a conic CLP, that distinguishes it from classical CLPs in which $X \geq 0$. ▲

Despite the fact that, as is, neither CLPs fits the general model of a RAN, we included them for three reasons: exploration, exploitation and conceptual. Starting with the latter reason, a CLP view of RANs adds to RMI a natural optimization-view (seeking "optimal" plans), which complements our earlier motivating discussions of offered-loads and bottlenecks, as well as later discussion of non-idling and maximal plans §3.4. Next, it suggests *exploring* open research questions and directions (e.g. extending existing CLP theories to cover RMIs). Finally, ready-made research may turn out *exploitable* or extendable, for example Duality (Levinson 1966), Conic CLPs (Shapiro 2001), and Linear Complementarity (references below, in Section 3.4). Examples to explore and exploit are outlined in Appendix B.

Arguably, the two most important families of problems in optimization are Linear Programming (Dantzig 1963) and Complementarity Problems (Cottle et al. 1968). We related RMIs to the former, and we now continue to the latter (adding Snapshot due to adding single resources).

3.4. RANs as (Extended) Dynamic Complementarity Programs ((E)DCPs)

The CLP view of RANs opens up optimization paths, for example seeking feasible plans that maximize a given utility. Digging deeper, experience with SPNs (arising from queueing models)

suggests that in fact stronger versions of optimization can also prevail: specifically, there could be maximal feasible plans or even a maximum (largest) one. Furthermore, it then turns out that maximality could be equivalent to non-idleness of plans, often called work-conserving strategies in queueing contexts: concrete examples are RANs (dynamic or static) associated with $\mathrm{G}_t/\mathrm{GI}/s_t$ (Section 4.1 and Appendix D), Machine-Repair models §4.2, Generalized-Jackson Networks §4.3, and Parallel-Servers §4.5. In all these examples, one can view a non-idling activity as having zero-slack in at least one of the sub-resources it consumes. This viewpoint enables one to accommodate general RANs (unlike (Dai and Harrison 2020, §7.1), that rely on the special structure of a queueing network).

For SPNs, or rather RANs with *only* single resources, a framework that accommodates the above is Dynamic Complementarity Theory (DCP) (Mandelbaum 1989, Stewart 2009) of various forms (Linear, Differential, ...), also referred to as Skorohod or Reflection Problems (Whitt 2002, §§14.2-4,14.9). RANs, however, require adding to DCP also Convolution Complemntarity (Stewart (2006), to accommodate G-convolutions) and Extended Complementarity (De Schutter and De Moor (1995, 1998), with its more general complementarity conditions). All these complementarity theories clearly emerge from our definition of Non-idleness (Complementarity), which we offer next for general RANs.

3.4.1. Non-Idling Plans. Striving for generality, we consider a RAN with single resources, as in §3.2 (specifically RMI (3) with (12), plus (13)). Its plan X is *non-idling at time* t if

$$\sum_j \prod_{l:\, C_{l,j}>0,\, r(l)\notin \mathtt{H}} (PG*X+\Lambda-CX)_l(t) \prod_{k:\, C_{l,j}>0,\, r(l)=k\in \mathtt{H}} \mathrm{d}(\hat{\Lambda}-RC\mathcal{A}X)_k(t)=0; \tag{21}$$

here $\mathtt{H}$ is the set of single-resources (if any) that are closed; $Q_l=(PG*X+\Lambda-CX)_l\geq 0$ are RMI (3) slacks of sub-resources; and $Y_k=(\hat{\Lambda}-RC\mathcal{A}X)_k\succeq 0$ are Snapshot (13) slacks of single-resources. Articulated succinctly with the non-negative slacks Q_l, and the non-decreasing slacks Y_k:

$$\sum_j \prod_{l:\, C_{l,j}>0, r(l)\notin \mathtt{H}} Q_l(t) \prod_{k:\, C_{l,j}>0, r(l)=k\in \mathtt{H}} \mathrm{d}Y_k(t)=0.$$

A corresponding static version (Cottle et al. 2009), arising from the Static RMI (10) ($Cx\leq Px+\lambda$), is: $\lambda=q+[C-P]x$, in which λ, C, P are given, and $(x,q)\geq 0$ is to be determined so that $\sum_j x_j \prod_{l:\, C_{lj}>0} q_l=0$; this is precisely an Extended Complementarity Problem, as mentioned above.

3.4.2. Three Forms of Complementarity. To simplify the discussion, assume that (21) applies over all $t \geq 0$. Non-idling then gives rise to three forms of complementarity between the slack functions Qs and Ys:

1. $Q_1 \cdot Q_2 = 0$, for $Q_1 \geq 0$, $Q_2 \geq 0$: there is no time t at which *both* $Q_1(t) > 0$ and $Q_2(t) > 0$. This characterizes a non-idling (work-conserving) $\mathrm{G}_t/\mathrm{GI}/s$ (Appendix D).
2. $Q \cdot \mathrm{d}Y = 0$, for $Q \geq 0$, $Y \succeq 0$: e.g. if $Q(t) > 0$ then Y does not increase at time t. This is the Skorohod problem, which characterizes $\mathrm{G}_t/\mathrm{GI}/1$-SPN, or $\mathrm{G}_t/\mathrm{GI}/s$-RAN with servers single-resourced (Appendix D).
3. $\mathrm{d}Y_1 \cdot \mathrm{d}Y_2 = 0$, for $Y_1 \succeq 0$, $Y_2 \succeq 0$: there is no time at which Y_1 and Y_2 *both* increase. This characterizes a machine-repair SPN model: single machine and single repairman, or machine-repair RAN with its two resources single-resourced (end of Section 4.2).

REMARK 11 (**Vanishing slacks**). Complementarity entails some vanishing slacks, which in fact relates the concepts of offered-loads, non-idleness and bottlenecks. Indeed, the offered-load associated with a set of sub-resources $\mathcal{S}$ induces zero-slacks for sub-resources not included in $\mathcal{S}$. An activity is non-idling if and only if at least one of the sub-resources it consumes has zero slack. A bottleneck arises when a group of sub-resources, consumed by a given activity, all have zero slacks. It thus follows that a *feasible plan is non-idling if and only if a bottleneck is associated with every activity*. Going the reversed direction, let us start with a feasible *non-idling* plan (21). Then the (sub)-resources that correspond to binding RMI constraints are elements of bottlenecks, with one bottleneck per activity. Letting $\mathcal{S}$ *be the set of all non-bottleneck (sub)-resources, the offered-load of* $\mathcal{S}$ *then equals the amounts of bottleneck (sub)-resources* (those not in $\mathcal{S}$). It will be instructive to (re)view the above via the lens of CLP duality (complementarity slackness (36)), after the supporting theory will have been developed. ▲

3.4.3. Maximality, Complementarity, Non-Idleness. A plan that is *idling* at time t has at least one activity j that is idle at t, which entails positive slacks of all the resources that this j consumes: $Q_l(t) > 0$ for the sub-resources l that are not closed, and $\mathrm{d}Y_k(t) > 0$ for closed resources k. Having these positive slacks suggests that an idling X can be increased through some re-planning of consumption and production or, conversely, that a non-idling X is maximal: there is no plan $\tilde{X}$ such that $X(t) \leq \tilde{X}(t)$, at all times $t \geq 0$, and $X_j(\tau) < \tilde{X}_j(\tau)$, for some activity j at some time τ. However, and despite the examples mentioned at the start of this section ($\mathrm{G}_t/\mathrm{GI}/s_t$, Jackson Networks, ...), the relation between non-idleness and maximality turns out subtle, which we demonstrate via examples below. These reveal that, for general RANs, extant theories of CLP and Complementarity cover neither the concepts of, nor the

relations among, offered plans, bottlenecks, maximality and non-idleness – indeed, a complete understanding clearly lies beyond our present scope.

EXAMPLE 4 (NON-IDLING AND NOT-MAXIMAL). Consider a single-activity RAN with two resources: $C=[1,\ 1]^\top$, $P=[1,\ 2]^\top$, $\Lambda=[1,\ 0]^\top$, and $G(t)=\frac{1}{2}\mathbb{1}\{t\geq 0\}+\frac{1}{2}\mathbb{1}\{t\geq 1\}$; note that Assumption 1 does not hold in this case. This RAN has a maximum plan $\tilde{X}$, given by $\tilde{X}(t)=2(\lfloor t\rfloor+1)$, $t\geq 0$. Hence the plan $X(t)=\lfloor t\rfloor+1$, $t\geq 0$, is not maximal, yet it is non-idling as $(PG*X+\Lambda-CX)(t)=[\frac{1}{2},\ 0]^\top$, over $t\in[0,1)$. The reason, in this specific example, combines its two features: 0-activity duration with positive probability, and producing 2 units of resource 2 after consuming only 1 unit of it; this enables the production of any amount of resource 2 in 0-time, as long there is sufficient amount of resource 1. ▲

EXAMPLE 5 (IDLING AND MAXIMAL). Consider a single-activity RAN (G/M/s_t) with two resources: $C=[1,\ 1]^\top$, $P=[0,\ 1]^\top$, $\Lambda(t)=[2+\lambda t,\ (2-\mathbb{1}\{t\geq 1\})]^\top$, and $G(t)=1-e^{-\mu t}$, $t\geq 0$. For the case $\mu<\min\{\lambda,\ln 2\}$, the plan $\tilde{X}(t)=e^{\mu}+\mu t\mathbb{1}\{t\geq 1\}$ is maximal, yet it is idling for $t\in[0,1)$. The idleness is due to the "sudden drop" in the amount of servers (the second resource) at time $t=1$: $\tilde{X}$ hence must idle during all $t<1$, in order to not violate the (future) capacity constraint at time $t=1$. ▲

Despite the above two examples, it was demonstrated in some cases and conjectured in others that RANs do sometimes exhibit relations, even equivalence, between non-idleness (21) and maximality of plans. We already mentioned such RANs (G$_t$/GI/s in Appendix D, Machine Repair in Section 4.2, GJNs in Section 4.3, Parallel Servers in Section 4.5), all being models of queueing networks that enjoy special structure (e.g. every activity consumes exactly two resources – one server and one customer; see also (Dai and Harrison 2020, §2.6)). Equivalence when C and P are square matrices has been addressed in complementarity research: static by the classical LCP (Cottle and Veinott 1972) and dynamic by DCP (Yang 1993, Mandelbaum 1989); and in both cases, equivalence holds iff $(C-P)$ is related to Leontief matrices. We are unaware of static analogues for ELCPs, not to mention dynamic complementarity models that would cover our RANs. By focusing on but one aspect out of many, our next example helps appreciate the subtlety in relating non-idleness and maximality.

EXAMPLE 6 (NON-IDLING IFF MAXIMAL). Reconsider Example 5, by replacing its $\Lambda_2(t)=2-\mathbb{1}\{t\geq 1\}$ with any Λ_2 for which $\{e^{\mu t}\Lambda_2(t),\ t\geq 0\}$ is a non-decreasing function. It can then be shown that (21) is sufficient and necessary for maximality since there exists sufficient capacity to avoid activity interruptions (recall the last part of §2.2.2). Indeed, no idling is required to support future capacity constraints since the amount of on-going activities decreases exponentially, at rate μ, when no new activities are initiated. ▲

4. Illustrative Examples

In this section, we present several RAN examples that can be identified with known (queueing) models. In most examples, the RAN goes beyond the existing: for example, it could be time-varying while the standard is stationary, or its primitives need not be smooth while the standard requires densities. Indeed, even in simple models, the RAN view is often instructive and insightful.

4.1. Single-Activity RANs

Starting with the simplest, one-activity one-resource RANs cover the $G_t/GI/\infty$ model (delay station), with RMI: $X \le A$. If $A \succeq 0$ then it models an arrival process, and the non-idling maximum plan is trivially $X^m = A$, with its busy process $B = A - A * G$ being the offered load (Whitt 2013). Single-activity two-resources RANs include: (1) The Taxi model (resources depart upon service completion: $X \le \Lambda_1$, $X \le \Lambda_2$), with the maximal plan $X^m = \Lambda_1 \wedge \Lambda_2$, if the latter is in $D^1_\uparrow$; and (2) A model where the activity requires two servers from two finite server-pools (closed resources). In this case $X^m = [I - G]^{-1}(\Lambda_1 \wedge \Lambda_2)$, assuming that $\Lambda_1 \wedge \Lambda_2$ is exhaustive (Section 2.2.2).

A last single-activity example is $G_t/GI/s_t$, with RMI: $X \le A$, $X \le G * X + s$; here $A \in D^1_\uparrow$ is an arrival function, G is service cdf, and $s \in D^1_+$ is a staffing function. Upon service completion, customers depart (open resources), while servers turn ready for their next service (closed); servers do still leave at times when s decreases. In Appendix D, we provide a systematic summary of $G_t/GI/s_t$. Here we only demonstrate that, already for this classical simple model, the RAN view uncovers some interesting facts and themes:

1. **$G_t/GI/s_t$: Fixed-point characterization, exhaustive staffing.** Introduce the operator $T : D^1 \to D^1$ by

$$X \mapsto T(X) := A \wedge (G * X + s). \tag{22}$$

Then, due to RMI, $T(X) - X = (A - X) \wedge (s - \bar{G} * X) \ge 0$, for any feasible plan X. It follows that a non-idling plan must be a fixed point of T, in which case it is also maximal: $X^m = A \wedge (G * X^m + s)$; equivalently $Q \wedge I \equiv 0$, where $Q := A - X^m$ and $I := s - \bar{G} * X^m$ are, respectively, queue-length and idleness. There is, however, a hidden subtlety here: one can show that, when $G(0) < 1$, the transformation T always has a unique fixed point; however, this fixed-point is not necessarily non-decreasing, which is inconsistent with the requirement $X \succeq 0$ for feasible plans. For example, if $A \equiv \infty$ then the fixed-point $X \succeq 0$ iff the staffing function s is an exhaustive busy process §2.2.2. A finite A does generalize and elevate the difficulty of the open problem

to characterize exhaustive processes, which now reads: given $A \succeq 0$ and cdf G, characterize the staffing functions s for which the unique fixed-point $X = T(X)$ is in $D^1_\uparrow$ (Liu and Whitt (2012) address this problem for differentiable A and G).

EXAMPLE 7 (EXHAUSTIVE FOR $A \equiv \infty$ BUT NOT FOR A FINITE). Let $G(t) = \mathbb{1}\{t \geq 1\}$, $s(t) = \mathbb{1}\{0 \leq t < 1\}$, $A(t) = \mathbb{1}\{t \geq 1/2\}$; then $X(t) = \mathbb{1}\{1/2 \leq t < 1\}$ is the unique fixed point of T, because $\mathbb{1}\{t \geq 1/2\} \wedge (\mathbb{1}\{3/2 \leq t < 2\} + \mathbb{1}\{0 \leq t < 1\}) = \mathbb{1}\{1/2 \leq t < 1\}$; but X is not non-decreasing; indeed, in this case there is no non-idling feasible plan. If, on the other hand, $A \equiv \infty$ then $X \equiv 1$ is the unique fixed point, which is also a feasible plan. ▲

2. **Limit interchange: From many- to single-server heavy-traffic.** Let L be the amount of customers in the $\mathrm{G}_t/\mathrm{GI}/s_t$ RAN. Then $L := A - G * X = \bar{G} * A + G * (A - X)$; alternatively $L - s = (A - X) - (G * X + s - X) = Q - I$, and, for a non-idling plan X, one has $(L-s)^+ = A - X$ (and $(L-s)^- = I$, thus L contains information about both slacks Q and I.) Therefore, in that case, $L = \bar{G} * A + G * (L - s)^+$, which is in agreement with the fluid results in Reed (2009); $\bar{G} * A$ is the corresponding $\mathrm{G}_t/\mathrm{GI}/\infty$ process.

We now show that single-resourcing servers of the $\mathrm{G}_t/\mathrm{GI}/s_t$ RAN leads to the Skorohod problem of the corresponding single-server queue (Mandelbaum 1989, Williams 2017). Specifically, single-resourcing "$X \leq G * X + s$" leads to "$X \preceq \hat{s}$", where $\hat{s}(t) = \mu \int_0^t s(u)\, \mathrm{d}u$. Then, in the single-resource limit, the total number of customers in the system, $L = A - G * X = A - X - (G * X - X)$, converges to $L = N + Y$: here $N = A - \hat{s}$ is a netput process, which is given; $Y = \hat{s} - X$ is lost potential, to be planned; and Y is Skorohod-feasible iff $L \geq 0$ and $Y \succeq 0$. Similarly, the characterization of a non-idling X, namely $(A - X) \wedge (s - \bar{G} * X) \equiv 0$, converges to $L\, \mathrm{d}Y \equiv 0$. One can then show that X^* is a non-idling plan (for the single-resourced RAN) iff $Y^* = \hat{s} - X^*$ is a *solution* of the Skorohod problem with data N. (It is perhaps surprising that $X^* = \hat{s} - Y^* \succeq 0$ follows from Y^* being a Skorohod solution with data $N = A - \hat{s}$.) To summarize, through single-resourcing servers, one obtains heavy-traffic results for $\mathrm{G}_t/\mathrm{GI}/1_t$ from those of $\mathrm{G}_t/\mathrm{GI}/s_t$. ▲

4.2. Machine-Repair Model

The model consists of m machines and s repairmen; both $m(\cdot) \in D^1_+$ and $s(\cdot) \in D^1_+$ can be time-varying. A machine is operational (working) for a random amount of time, distributed according to F (with mean $1/\lambda$), before it breaks down. A broken machine must be repaired. A single repairman repairs one machine at a time, and repair times are distributed according to G (with mean $1/\mu$). Once a machine is repaired it starts working, and the cycle repeats. All repair and working durations are independent.

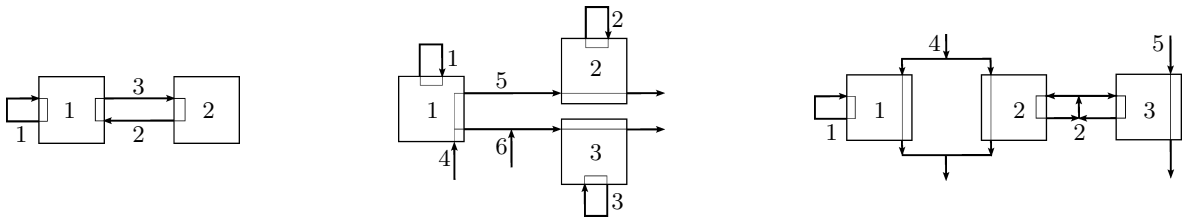


Figure 5 Three *activity-diagrams* of RANs: machine-repair §4.2 (left), three-node GJN §4.3 (center), and an N-model §4.5 (right). In the diagrams, nodes and arcs represent activities and sub-resources, respectively. The arc-arrows indicate whether specific sub-resources are consumed (inward) and/or produced (outward); for example, consider activity 1 (repair) in machine-repair: it both consumes and produces sub-resource 1 (idle server), and it also consumes sub-resource 2 (broken machine) and produces sub-resource 3 (working machine). One could also draw corresponding *resource-diagrams*, for example depicting flow of "customers" through "resources" (e.g. architectures such as V, N, W, M and X (Garnett and Mandelbaum 2000)). Resource-diagrams are simpler, more prevalent but, unlike activity-diagrams, resource roles are not symmetric .

The RAN of machine-repair has two activities (1: repairing, 2: working), two resources (1: s repairmen, 2: m machines), and three sub-resources (1: idle repairman, 2: broken machine, 3: repaired machine) – see Figure 5 (left). The corresponding RAN primitives are as follows:

$$R=\begin{bmatrix}1&0&0\\0&1&1\end{bmatrix},\quad \mathcal{A}=\begin{bmatrix}1/\mu&0\\0&1/\lambda\end{bmatrix},\quad C=\begin{bmatrix}1&0\\1&0\\0&1\end{bmatrix},\quad P=\begin{bmatrix}1&0\\0&1\\1&0\end{bmatrix},\quad \Lambda=\begin{bmatrix}s\\m_1\\m_2\end{bmatrix},\quad b=\begin{bmatrix}s\\m\end{bmatrix},$$

where $m_1+m_2=m$; here $m_1(0)$ and $m_2(0)$ are the amounts of machines that just started working and just broke down at time $t=0$, respectively. A plan $X=(X_1,X_2)^\top$ determines cumulative amounts of "repair-starts from broken" (X_1) and of "work-starts from repaired" (X_2). RMI is then

$$X_1\le G*X_1+s,\quad X_1\le F*X_2+m_1,\quad X_2\le G*X_1+m_2.$$

Slacks in the three flow inequalities can be interpreted, respectively, as the amounts of idle repairmen, of broken machines awaiting repair, and of repaired machines that are not (yet) working. The corresponding snapshot inequalities are $\bar{G}X_1\le s$ and $\bar{G}X_1+\bar{F}X_2\le m$.

When $m_2\succeq 0$, finding a non-idling X can be done as follows. Monotonicity and the third RMI inequality imply that the non-idling X satisfies $X_2=G*X_1+m_2$. Utilizing this equality in the other two yields that X_1, corresponding to the non-idling X, is a solution of the following fixed-point equation:

$$X_1=(G*X_1+s)\wedge(F*G*X_1+F*m_2+m_1). \tag{23}$$

provided $X_1\in D^1_\uparrow$. (Complete analysis of (23) is left for future research.) Note that the above expression generalizes (22), that solves for the maximal throughput of $\mathrm{G}_t/\mathrm{G}/s_t$. Specifically,

machine-repair reduces to $\mathrm{G}_t/\mathrm{G}/s_t$ when working durations are infinite ($F=0$); then m_1 is cumulative arrivals to the system, and (23) becomes $X_1=(G*X_1+s)\wedge m_1$ (22).

4.2.1. Static Machine-Repair. Constraints on long-run rates can be obtained when $m(t)\to m(\infty)$ and $s(t)\to s(\infty)$, as $t\uparrow\infty$. Indeed, for $X\in\bar{D}^2_\infty$, namely $t^{-1}X(t)\to x$, as $t\uparrow\infty$, (11) implies $x_1/\mu\le s(\infty)$ and $x_1/\mu+x_2/\lambda\le m(\infty)$ – the former due to repairmen and the latter to machines. Adding $Cx=Px$ (10), which implies $x_1=x_2$, identifies the components of the maximal x to be

$$x_1=x_2=\mu\, s(\infty)\wedge\frac{\lambda\mu}{\lambda+\mu}m(\infty).$$

4.2.2. Single-Resourcing Machine-Repair. There are three different ways to single-resource the Machines Repairmen Model §3.2: single repairman, single machine, and both being single. Each of the corresponding models, which we now describe, gives rise to either a new insight or an open problem:

1. *Single repairman:* This RAN is a cyclic network with two stations, one single-server and the other infinite-server. Its RMI is $X_1\preceq\mu\hat{s}=:\tilde{s}$, $X_1\le F*X_2+m_1$, $X_2\le X_1+m_2$. For a maximal plan, that is also non-idling, one clearly takes $X_2:=X_1+m_2$. This leaves one with maximizing X_1, subject to $X_1\preceq\tilde{s}$ and $\bar{F}*X_1\le m_1+F*m_2=:m^+$. We now outline how this maximum can be derived via solving a generalized version of the 1-dimensional Skorohod problem (Williams 2017).

Define $Y:=\tilde{s}-X_1$, $Z:=m^+-\bar{F}*X_1$, and $N:=m^+-\bar{F}*\tilde{s}$. Then the set of all $Y\succeq 0$ such that $Z=N+[I-F]*Y\ge 0$, is the feasibility set of a *generalized Skorohod problem* ($F\equiv 0$ is the classical problem). This set has a least Y^m (equivalently maximal X_1^m), which is also characterized by the complementarity condition $Z\,\mathrm{d}Y\equiv 0$. One can in fact solve for the least Y^m by writing $Z=[N-F*Y]+Y$, then deducing that $Y(t)=\sup_{0\le u\le t}[F*Y(u)-N(u)]$, $t\ge 0$, by the classical Skorohod problem: Y^m is hence the unique fixed-point of a contraction. (For $F\equiv 0$, Y is in fact explicitly given.) A final comment is that $X_1:=\tilde{s}-Y\in D^1_\uparrow$ must hold if (X_1,X_2) is to be a plan; a sufficient condition is that m_1 and m_2 are both in $D^1_\uparrow$.

2. *Single machine:* This RAN is a cyclic network with two infinite-server stations and a single machine (customer), where the capacity constraint is on the sum of the two processing capacities. (With only a single machine, one can interpret both stations as single servers). RMI takes the form $\frac{1}{\mu}X_1+\frac{1}{\lambda}X_2\preceq\hat{m}:=\hat{m}_1+\hat{m}_2$, and $X_1=X_2$; equivalently $X_1\preceq\frac{\lambda\mu}{\lambda+\mu}\hat{m}=:\tilde{m}$, and $X_2=X_1$. Then, for $t\ge 0$, the maximal plan, which is also non-idling, is given by $X_1(t)=X_2(t)=\inf_{u\ge t}\tilde{m}(u)$, namely the largest non-decreasing function that is dominated by $\tilde{m}$. In particular, if $\tilde{m}\succeq 0$ then $X_1=X_2=\tilde{m}$.

3. *Single machine and single repairman:* This RAN is the same as in the second case but with the additional constrain, as in the first case, on the processing capacity of the first station. RMI yields $X_1 \preceq \tilde{s}$ and $X_1 \preceq \tilde{m}$. The maximal solution, which is also non-idling, is then $X_1 = X_2 = \tilde{m} \curlywedge \tilde{s}$, where the minimum $\curlywedge$ is with respect to the partial order $\preceq$. One can in fact construct this minimum as follows. Represent the bounded-variation function $\tilde{s} - \tilde{m}$ as the difference between its increasing and decreasing parts: $\tilde{s} - \tilde{m} = a - b$, where $a, b \succeq 0$, $a(0) = b(0) = 0$ and $\mathrm{d}a \wedge \mathrm{d}b = 0$. The maximal solution is then given by $X_1 = X_2 = \tilde{s} - a \equiv \tilde{m} - b$.

4.3. Generalized Jackson Network (GJN)

We now describe fluid RANs that arise from time-varying Generalized Jackson Networks (GJNs). Specifically, we cover many-server (Liu and Whitt 2014), stationary and single-server (Chen and Yao 2013, Ch. 7) GJNs, that are either open or closed. (The introductory footprints of $\mathrm{G}_t/\mathrm{GI}/s_t$, in §4.1 and Appendix D, will be clearly recognized.)

GJN has J stations/nodes, such that each station j has an amount (possibly time-varying) of $s_j \in D_+^1$ servers, who provide iid services with cdf G_j; routing is Markovian, according to a $J \times J$ sub-stochastic matrix Π ($\Pi \geq 0$ and $\Pi \cdot \bar{1} \leq \bar{1}$): specifically, Π_{j_1,j_2} is the probability that a customer, having completed a service at station j_1, continues directly to station j_2; hence $1 - \sum_j \Pi_{j_1,j}$ is the probability of exiting the network after the j_1-service. Finally, for each j, let $\mathcal{E}_j \in D_+^1$ be (mean) cumulative amount of exogenous arrivals/departures (of customer) to station j: arrivals prevail when $\mathcal{E}_j$ increases and departures when it decreases.

The GJN fluid RAN has J activities, $J+1$ resources, and $2J$ sub-resources. Activity j is a service at station j, of one customer by one of the s_j servers there. Resources $1, \ldots, J$ correspond to servers at the J stations, and resource $J+1$ to customers. Sub-resource $j = 1, \ldots, J$ corresponds to an idle server at station j; and sub-resource $J + j \in \{J+1, \ldots, 2J\}$, is a customer at station j, either queueing up or being served there (see Figure 5 (center) for an example of a three-node GJN). It follows from the above that the RAN building blocks are

$$R = \begin{bmatrix} I_{J\times J} & \bar{0}_{J\times J} \\ \bar{0}_{1\times J} & \bar{1}_{1\times J} \end{bmatrix}, \quad C = \begin{bmatrix} I \\ I \end{bmatrix}, \quad P = \begin{bmatrix} I \\ \Pi^\top \end{bmatrix}, \quad \Lambda = \begin{bmatrix} s \\ \mathcal{E} \end{bmatrix};$$

here $I_{J\times J}$ (or simply I) is the identity matrix, $\bar{0}$ is a matrix of all zeros, $\bar{1}$ is a (row) vector of ones, $\mathcal{E} := (\mathcal{E}_1, \ldots, \mathcal{E}_J)^\top$, and $s := (s_1, \ldots, s_J)^\top$ (with the latter two being both in D_+^J).

Every activity consumes two (input) sub-resources – one sub-resource that involves the servers resource, and one that involves the customers resource. Activities may produce, however, more than just two (output) sub-resources: first a single sub-resource that involves the servers resource, but also one or several sub-resources that involve the customers resource (capturing

routing after a service to subsequent stations). Note that Assumption 1 prevails: $RC \geq RP$ since Π is sub-stochastic. The role of P^- can be played by any $P^- := \left[\bar{0}_{J\times J}\ (\tilde{\Pi} - \Pi)\right]^\top$, where $\tilde{\Pi}$ is any *stochastic* $J \times J$ matrix $\tilde{\Pi}$ ($\tilde{\Pi} \geq 0$ and $\tilde{\Pi} \cdot \bar{1} = \bar{1}$) such that $\tilde{\Pi} \geq \Pi$ element-wise. Then $RC = R[P + P^-]$, since $\tilde{\Pi}$ is stochastic. In particular, if Π itself is stochastic then customers do not leave the network.

RMI of GJN-RAN constitutes the following $2J$ inequalities:

$$X \leq G * X + s, \quad X \leq \Pi^\top G * X + \mathcal{E}, \qquad \text{for plans} \quad X \in D^J_\uparrow; \tag{24}$$

the first J inequalities are server constraints per station, and the other J are customer routing constraints. Generalizing (22), RMI can be summarized into merely J inequalities, written succinctly in terms of a (monotone) transformation $T : D^J \to D^J$ as follows:

$$X \leq T(X), \quad \text{where} \quad T(X) := (G * X + s) \wedge (\Pi^\top G * X + \mathcal{E}). \tag{25}$$

Non-idling Maximal Plans. The complementarity condition (21) reduces to J complementarities between RMI slacks (24) so that, at each station, there cannot be simultaneously idle servers and waiting customers (often referred to as work-conservation). Formally:

$$(s - \bar{G} * X) \wedge (\Pi^\top G * X + \mathcal{E} - X) \equiv 0,$$

which is equivalent to $T(X) \equiv X$, namely X is a fixed-point of T. Since $X \leq T(X)$ for all plans, a fixed-point X is plausibly the maximal plan if also $X \in D^J_\uparrow$.

An analysis of the transformation T and its fixed points (re. existence, uniqueness, feasibility: $X \in D^J_\uparrow$) is left for future research. It offers a significant generalization of (22) and (23)); and it is a special case of the Dynamic Complementarity Problem (re. non-idling, maximality) in §3.4.3. Note that a fixed point, if existing, is in $D^J_\uparrow$ when both $s \in D^J_\uparrow$ (e.g. constant) and $\mathcal{E} \in D^J_\uparrow$ (e.g. cumulative arrivals). This happens in Static GJNs, which we turn to next.

Static GJN §3.1. Assume that $s(\infty) := \lim_{t\uparrow\infty} s(t)$, and $\mathcal{E}(t) := \mathcal{E}(0) + \lambda t$, $t \geq 0$, with $\mathcal{E}(0), \lambda \in \mathbb{R}^J_+$. For a plan $X \in \bar{D}^m_\infty$, such that $\lim_{t\to\infty} \frac{1}{t} X(t) = x \in \mathbb{R}^J_+$, divide either (24) or (25) by t and let $t \uparrow \infty$: we get $x \leq \mathcal{A}^{-1} s \wedge (\Pi^\top x + \lambda)$, in view of (8) and (9), which combines Static RMI (10) with Static Snapshot (11). We now show that this combination leads to the classical traffic equations of GJNs. To this end, change variables from x to $y = \Pi^\top x + \lambda$, which yields $y \leq \lambda + \Pi^\top (y \wedge \mathcal{A}^{-1} s)$. It can now be shown (Chen and Mandelbaum 1991a,c) that a maximal y exists; it is in fact a maximal fixed-point of the traffic equation

$$y = \lambda + \Pi^\top (y \wedge \mathcal{A}^{-1} s), \tag{26}$$

in which $\mathcal{A}^{-1}s$ is the vector of processing capacities and y is the inflow vector, hence $y \wedge \mathcal{A}^{-1}s$ is throughput. Note that the mapping $x \mapsto y$ is not necessarily a bijection: there could be multiple x's that correspond to the same y. If y is a maximal fixed-point of (26), then all of the corresponding x's are maximal static plans.

The traffic equations can be also interpreted as a Linear Complementarity Problem (LCP), that is analyzed in Chen and Mandelbaum (1991a) separately for open vs. closed GJNs. While RANs offer a unified treatment, such a distinction does have its nuances, as we now outline.

Static Closed GJN. Our closed GJN has parameters Π stochastic and irreducible, $s(t) \equiv s(0) \geq 0$ (per-station fixed amounts of servers), and $\mathcal{E}(t) \equiv \mathcal{E}(0) \geq 0$ (network-total fixed amounts of customers); it follows that $R\Lambda \equiv [s(0),\ \sum_{j=1}^{J} \mathcal{E}_j(0)]^\top$. Static RMI (10) now implies $x = \Pi^\top x$, hence $y = x$ (inflow = ouflow = throughput, for a closed GJN); and Static Snapshot (11) yields the following $J+1$ inequalities

$$\begin{bmatrix} \mathcal{A}x \\ a^\top x \end{bmatrix} \leq \begin{bmatrix} s(0) \\ \bar{1}^\top \mathcal{E}(0) \end{bmatrix},$$

in which $\mathcal{A}$ is the diagonal matrix of mean service durations, and $\bar{1}$ is a row vector of ones; hidden in the above are the traffic equations $x = \Pi^\top x \leq \mathcal{A}^{-1}x$ (recalling $x = y$). By irreducibility of Π, x is proportional to the unique stationary distribution π: $x = c\pi$, for some scalar $c > 0$. The maximal plan then corresponds to

$$c = \frac{\bar{1}^\top \mathcal{E}(0)}{a^\top \pi} \bigwedge_{j=1}^{J} \frac{s_j(0)}{a_j \pi_j}.$$

This last expression is the minimum of $J+1$ elements, one for each of the $J+1$ resources. The elements achieving the minimum create a bottleneck set §3.3.2 and, in particular, resource "customers" can be a bottleneck (in which case, all sub-resources corresponding to customers are bottlenecks). In a balanced network, all $J+1$ resources are bottlenecks, which occurs when $\sum_j \mathcal{E}_j(0) = \sum_j s_j(0)$ and

$$\frac{s_1(0)}{a_1\pi_1} = \cdots = \frac{s_J(0)}{a_J\pi_J} = \frac{\sum_j \mathcal{E}_j(0)}{a^\top \pi} = \frac{\bar{1}^\top \mathcal{E}(0)}{a^\top \pi}.$$

Static Open GJN. Here $\mathcal{E}(t) = \lambda t$, with at least one $\lambda_j > 0$; and Π is sub-stochastic with spectral radius less than 1, hence at least one row-sum is less than one. The traffic equations now have a unique solution y (trivially maximal); and, contrary to the closed case, $y \neq x$ can happen, namelyl $y_j > s_j/a_j$ for some j (strict bottleneck).

Single-Resourcing GJN. We now show that single-resourcing all server-pools in GJN-RAN gives rise to an SPN-RAN; then solving for a non-idling plan amounts to solving a Skorohod (oblique reflection) problem; which, in the static case, again reduces to solving the traffic equations.

Servers are closed resources. Hence, by (3) and (13), single-resourcing all server pools gives rise to the following RMI, for $X \in D^J_\uparrow$: $X \preceq \hat{s}$, and $(I-\Pi^\top)X \leq \mathcal{E}$, where $\hat{s}_i(\cdot) := \mathcal{A}^{-1} \int_0^\cdot s_i(u)\,\mathrm{d}u$. A non-idling X is then determined by having at least one of the preceding two inequalities tight, for each of the stations; formally, $Q^\top \mathrm{d}Y \equiv 0$, where $Q := \mathcal{E} - [I-\Pi^\top]X$, and $Y := \hat{s} - X$ are RMI slacks (21). Now solving for a non-idling plan is in fact solving a Skorohod problem, in $\mathbb{R}^J_+$ for open and on a J-dimensional simplex for closed GJNs. The data is $N := \mathcal{E} + \Pi^\top \hat{s} - \hat{s}$ (netflow), and an oblique-reflection matrix $[I-\Pi^\top]$:

$$Q = N + [I-\Pi^\top]Y \geq 0, \qquad Q\,\mathrm{d}Y \equiv 0. \tag{27}$$

For proofs of existence, uniqueness and minimality of Y, readers are referred to §14.2 in Whitt (2002) for Π arising from open GJNs; and to Chen and Mandelbaum (1991a,b) for closed. Note that Y is minimal iff X is maximal. There is however one point that we leave open: due to the special structure of N, if Y is the maximal solution then $X := \hat{s} - Y \in D^J_\uparrow$.

Finally, when $\hat{s}(t) := \mu t$, $\mathcal{E}(t) := \mathcal{E}(0) + \lambda t$, and considering static plans, the non-idling condition simplifies to the traffic equations (26): $(\mu - x)^\top(\lambda - (I-\Pi^\top)x) = 0$, or $x = \mu \wedge (\Pi^\top x + \lambda)$, or $y = \lambda + \Pi^\top y \wedge \mu$ (where $y := \lambda + \Pi^\top x$).

4.4. Re-Balancing Networks

A ridesharing system was modeled as a closed queueing network, in Benjaafar et al. (2022) and Braverman et al. (2019). The model can be interpreted as a network consisting of N circulating customers, r single-server queues, and $2r^2$ infinite-server queues. The customers model cars in a ridesharing system, while single-server queues correspond to geographical regions. In this subsection, we slightly deviate from the "standard" RAN notation in order to make the model mode interpretable (e.g., we index activities with both single and double indices). Potential number of service completions in time interval $[0,t]$ at single-server queue i is $\Lambda_i(t)N$ – these model passenger arrivals; a passenger that does not encounter an available car abandons the system. A passenger matched with a car in region i has its destination in region j with probability $U_{i,j}$; the duration of such a ride is $G_{i,j}$ with mean $a_{i,j}$ (one of r^2 infinite-server queues modeling full-car rides). Once a passenger completes its ride, the empty car is moved to region

j with probability Q_{ij}, in order to pick up a next passenger (one of r^2 infinite-server queues modeling empty-car rides).

The system is in the large-demand regime $N \uparrow \infty$, which corresponds to a fluid RAN. Let X_i, $X_{i,j}$ and $Y_{i,j}$ be, respectively, the number of passengers matched to cars in region i during $[0,t]$, the number of full-car rides from region i to region j that start in $[0,t]$, and the number of full-car rides from region i to region j that start in $[0,t]$. The corresponding RAN model is then characterized by

$$X_{i,j} \leq U_{i,j} X_i, \tag{28}$$

$$Y_{i,j} \leq Q_{i,j} \sum_{k=1}^{r} G_{k,i} * X_{k,i}, \tag{29}$$

$$X_i \leq \sum_{j=1}^{r} G_{j,i} * Y_{j,i} + n_i, \tag{30}$$

$$(\Lambda_i - X_i) \succeq 0, \quad i = 1, \ldots, r; \tag{31}$$

here we assume that n_i (scaled) cars are awaiting passengers at time 0, and $n = n_1 + \cdots + n_r = 1$. Constraints (30) and (31) are due to single-server queues §3.2. The goal is to design the re-routing matrix Q such that the revenue by time t is maximized:

$$\max_{Q} \max_{X} \sum_{i=1}^{r} X_i(t) \sum_{j=1}^{r} c_{i,j} U_{i,j},$$

where c_{ij} is average revenue generated by a passenger travelling from region i to region j. Substituting (29) in (30), then combining it with (28), results in

$$X_i \leq \sum_{j=1}^{r} Q_{j,i} G_{j,i} * \sum_{k=1}^{r} G_{k,j} * X_{k,j} + n_i \leq \sum_{k=1}^{r} X_k * \sum_{j=1}^{r} (G_{k,j} U_{k,j} * G_{j,i} Q_{j,i}) + n_i;$$

note that (28) and (29) become equality for maximal plans. Summing up the preceding over all indices yields

$$\sum_{k=1}^{r} X_k \leq \sum_{k=1}^{r} X_k * \sum_{i,j=1}^{r} (G_{k,j} U_{k,j} * G_{j,i} Q_{j,i}) + n,$$

which is a constraint that relates the total amount of matched passenger-car pairs in $[0,t]$ (see the objective function) with the number of cars in the system n. Then, the long-run average version of the optimization problem is

$$\max_{Q} \max_{x} \sum_{i=1}^{r} \beta_i x_i,$$

$$\begin{aligned}
\text{subject to } & \sum_{k=1}^{r} \alpha_k x_k \le n, \\
& x_k \le \lambda_k, \quad k = 1, \dots, r, \\
& \alpha_k = \sum_{i,j=1}^{r} U_{k,j} Q_{j,i} \left(a_{k,j} + a_{j,i} \right), \quad k = 1, \dots, r, \\
& \beta_i = \sum_{j=1}^{r} c_{i,j} U_{i,j}, \quad i = 1, \dots, r.
\end{aligned}$$

4.5. Controlled Matching Models

In this subsection, we illustrate how the RAN framework can accommodate specific control strategies, here via two "matching" models: input-queues under static priorities, and output-queues under join-the-shortest-queue.

Input Queues with Static Priorities. Following the parallel-server model presented in Williams (2000), there are M customer classes that receive service from N server pools, according to some pre-designed topology (see Figure 5 (right) for an example of the so-called N-model with $M = 2$ and $N = 2$). In a RAN model of such a system, there are $J \le M \times N$ activities, where activity $j = (m, n)$ corresponds to a service of a customer from class m by a server from server pool n. Since in RANs, both customers and servers are resources, the model consists of a total of $K = M + N$ resources. Activity $j = (m, n)$ consumes an *arriving customer of class* m and an *available server from pool* n. Once completed, it produces an *available server from pool* n and the customer leaves the system. Thus, there is a single sub-resource (a resource-state pair) for every resource, implying that $R := I_{K \times K}$.

One can rearrange the matrices C, P and Λ such that their top part represents sub-resources that involve customers and the bottom part represents sub-resources involving servers:

$$C = \begin{bmatrix} C^M \\ C^N \end{bmatrix}, \quad P = \begin{bmatrix} 0 \\ C^N \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda^M \\ \Lambda^N \end{bmatrix};$$

here $C^M_{m,j} = 1$ if activity j engages a customer from class m, and 0 otherwise; and $C^N_{n,j} = 1$ if activity j engages a server from class n, and 0 otherwise. Class m customers arrive to the system according to $\Lambda^M_m \in D_\uparrow$; the number of servers in each pool is given by the non-negative function $\Lambda^N_n = s_n$. RMI then yields

$$C^M X \le \Lambda^M, \qquad C^N \bar{G} * X \le \Lambda^N.$$

Note that $Q := \Lambda^M - C^M X$ quantifies the amount of customers awaiting service, and $I := \Lambda^N - C^N \bar{G} * X$ is the amount of idle servers. Additional constraints are needed to specify a particular policy. For example, consider the following (non-preemptive) priority policy: customers

are routed to the lowest-index available servers, while idle servers are routed to the lowest-index available customers. In that case, for every activity $j=(m,n)$, we add

$$\mathrm{d}X_j \sum_{l<m:\, C^M_{l,n}=1} Q_l \equiv 0, \qquad \mathrm{d}X_j \sum_{l<n:\, C^N_{m,l}=1} I_l \equiv 0, \tag{32}$$

where sums over empty sets are assumed zero.

Output Queues with Join-the-Shortest-Queue (JSQ). Customers arrive according to $\Lambda \in D^1_\uparrow$ and, immediately upon arrival, they are routed to one of J infinite-capacity queues, indexed by $j=1,\ldots,J$; there are also dedicated exogenous arrivals to queue j, according to $\Lambda_j \in D^1_\uparrow$; and each customer in queue j is served by one member of server-pool j, which consists of s_j homogeneous servers.

RAN has J core activities, so that activity j is a service by pool-j server. To model routing, introduce J auxiliary activities, indexed by $J+1,\ldots,2J$, that have zero-duration. These activities "consume" arrivals, and activity $J+j$ "produces" customers routed to pool j. RMI then amounts to the following relations, which apply to an arbitrary routing policy:

$$\sum_{j=1}^{J} X_{J+j} = \Lambda,$$
$$X_j \le (X_{J+j}+\Lambda_j) \wedge (G_j * X_j + s_j), \quad j=1,\ldots,J.$$

The first equality indicates that, upon arrival, Λ-customers are routed to one of the queues. The second inequality holds as equality for non-idling plans: these are characterized by a fixed-point equation (compare with (25)), the analysis of which is an open problem. Finally, JSQ dynamics is enforced by the additional constraint

$$\sum_{j=1}^{J} \mathbb{1}\{Q_j(t) > \min_l Q_l(t)\}\, \mathrm{d}X_{J+j}(t) \equiv 0, \tag{33}$$

where $Q_j := X_{J+j} + \Lambda_j - X_j \ge 0$.

4.6. Data-based Example: Robots Fulfilling Online Orders.

This example demonstrates the applicability of the RAN framework to modelling and analyzing real complex systems. Specifically, we describe RAN models of a robotic system for online order-fulfillment, which consists of anywhere between tens to few hundreds robots. (Another example is an Executive Health Screening clinic, that operates as an appointment-based service-shop; see §5.4 of Carmeli (2022) and Chen et al. (2022).)

In our order-fulfillment center, inventory is stowed in thousands of totes that are racked in horizontal rows of many multiple levels (the more levels the less area). An arriving order

typically requires items from multiple totes, which are transported by robots from the shelving units to one of several *stations* on site. The stations are occupied by human operators or robotic arms – these collect the items required for the order, after which the totes are shelved back by robots.

The robotic system consists of two types of robots: *ground* robots that can move on the floor in all four directions, and *lift* robots that can retrieve and insert totes from and to the tall shelving units. The process of transporting a tote to a station starts with a lift robot retrieving the tote from the shelving unit and bringing it to a meetup point on the floor. In parallel, an empty ground robot is directed to that meetup point, where it fetches the tote and carries it to the station. Additional fulfilment processes, such as decanting inventory into totes, are performed similarly.

The system has been modeled as a complex RAN, with both closed resources (ground robots, lift robots, stations, meetups) and open resources (orders, inventory); resources that operate in the many-server regime (ground robots, lift robots, meetups, totes, empty cells within the shelving units) or single-server resources (stations, operators). Since activities may be racking-unit-dependent and station-dependent (e.g., transferring a tote from racking-unit 1 to station A), a RAN model of such a system can quickly blow up to hundreds of activities and sub-resources, depending on the application and hence the appropriate aggregation. (See Figure 7 in Appendix E for a simplified activity diagram of such a RAN.)

The RAN framework has played a key role in analyzing and optimizing robotic systems as above. It has been used for dimensioning – how many ground robots, lift robots and stations are required to achieve a desired throughput; to identify static and dynamic bottlenecks; evaluate novel features and architectural changes (specifically, when such changes are time- or effort-consuming to develop or implement or even simulate); and to determine how many ground and lift robots should be dynamically allocated to specific missions, for example to collecting items for newly arrived orders, decanting inventory into empty totes, dispatching full order totes, etc. Appendix E adds details on the ground-robots allocation problem and its RAN-based solution.

5. Extensions

In the previous section, we enriched the core RAN model with control policies (static priorities, JSQ). To further demonstrate RAN's modeling strength, we now describe, at least partially, three RAN generalizations: initiating activities prior to time zero, which enables steady-state dynamics; Fork-Join of non-exchangeable entities; and abandonents while idling. Additional generalization will be described in the next "Future Research" section.

5.1. General Initial Conditions and Stationary Plans

A plan $X = \{X(t), t \geq 0\}$ starts counting activity initiations from and including time 0. However, there could also be activities that are already in progress at time 0, which means that in fact they started *before* time 0. One must then account for these activities when describing the system state at time 0. We shall hereafter say that an activity is *in progress* at time 0 if it started strictly prior to time 0.

Define $V = \{V(t), t \geq 0\} \in D_{\uparrow}^{J}$, where $V(t)$ represents the amount of activities that are in progress (as opposed to starting) at time 0 and are completed by time t (before or at t); $V(0) = 0$ without loss of generality (which implies that $V(\infty)$ is the amount of activities in progress at time $t = 0$). Then, with a general initial state, RMI and Snapshot become

$$\boxed{\text{RMI:} \qquad CX \leq P\left(G * X + V\right) + \Lambda,}$$

$$\boxed{\text{Snapshot:} \qquad RC\bar{G}X \leq R(PV + \Lambda).}$$

Stationary plan. A feasible static plan can be related to a feasible dynamic plan by being either the latter's long-run limit or its steady-state (stationary) plan. The limit relation was considered in Section 3.1. Going the other direction, we start with a static feasible plan x, and we seek a dynamic plan X so that the static x is a steady-state of X. Formally, for a RAN with $\Lambda = \{\Lambda(t) = \lambda t, t \geq 0\}$, for some $\lambda \in \mathbb{R}^L$, one seeks a dynamic plan X and a process V such that, at all times $t \geq 0$:

$$(G * X + V)_j(t) = x_j t \quad \text{and} \quad (\bar{G} * X + \bar{V})_j(t) = a_j x_j; \tag{34}$$

or, in matrix form: $\frac{1}{t}(G * X + V)(t) \equiv x$, and $(\bar{G} * X + \bar{V})(t) \equiv Ax$; here $\bar{V} = \{\bar{V}(t) := V(\infty) - V(t), t \geq 0\}$, where $\bar{V}(t)$ is the number of activities that were in progress at time 0 and are still in progress at time $t > 0$. The two conditions (34) constitute a natural articulation of steady-state since x_j, being the jth component of the static feasible plan x, is the steady-state rate at which activity j is performed; $a_j = \mathbb{E}\sigma_j$, where σ_j is the (random) duration of activity j (with cdf G_j); and hence $a_j x_j$ is the (average) number of ongoing j-activities in steady-state.

Starting with a feasible plan x, namely x that satisfies (10), to specify a dynamic plan of which x is steady-state, let

$$X(t) = xt \quad \text{and} \quad V_j(t) = x_j \int_0^t \bar{G}_j(u)\, \mathrm{d}u,$$

for all $t \geq 0$; thus, there are $V(\infty) = \mathcal{A}x$ ongoing activities initially. Then we have $\left(G * X + V\right)_j(t) = \left(X - \bar{G} * X + V\right)_j(t) = x_j t$ and

$$\left(\bar{G} * X + \bar{V}\right)_j(t) = x_j \int_{[0,t]} \bar{G}(t-u)\, \mathrm{d}u + V_j(\infty) - V_j(t) = x_j \int_0^t \bar{G}(u)\, \mathrm{d}u + \int_t^\infty \mathrm{d}V_j(u) = a_j x_j.$$

Observe that the chosen X and V satisfy RMI: $Cxt \le Pxt + \lambda t$, since the corresponding x adheres to static RMI. We refer to the pair X and V as a stationary plan, not only because of (34), but also since the distribution of the remaining durations of on-going activities does not vary with time. Indeed, the amount of fluid initially engaged in activity j, with remaining activity duration more than $u > 0$ is $\bar{V}_j(u) = a_j x_j \mathbb{P}[\sigma_j^e > u]$, where σ_j^e is a random variable distributed according to the residual distribution of G_j. This persists also at $t > 0$ since, for $u \ge 0$, we have

$$\bar{V}_j(t+u) + \int_0^t \bar{G}_j(t-z+u)\, \mathrm{d}X_j(z) = \bar{V}_j(t+u) + x_j \int_u^{t+u} \bar{G}_j(z)\, \mathrm{d}z = \bar{V}_j(u).$$

REMARK 12 (**Stationarity**). Note that we only consider stationary plans, not stationaty RANs. For a RAN to be in stationarity, in addition to a plan X being stationary, one would need to require (at least) that the slacks are constants (and specified). Hence, the notion of stationarity does not exist for all RANs. For example, consider a single activity RAN with two resources: $C = [1\ 1]^\top$, $P = [0\ 0]^\top$ and $\Lambda = [\Lambda_1\ \Lambda_2]^\top$, where $\Lambda_1(t) = \lambda_1 t$ and $\Lambda_2(t) = \lambda_2 t$, $t \ge 0$. Unless $\lambda_1 = \lambda_2$, no stationary plan yields a stationary RAN. Moreover, for RANs that permit stationarity, the stationary "state" of the system need not be unique. To illustrate this point, let

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad \text{and} \quad R\Lambda = b \in (0, \infty).$$

Then, for a dynamic plan X defined by rate $x < b/(2a)$, any initial values of slacks do not change with time. ▲

5.2. Fork-Join (FJ) Constructs

A FJ network has of two (or more) activities that can be performed in parallel by different resources, where both (or all) of these activities must be completed in order for the process to continue. A refined machine-repair system can illustrate the concept: it caters to m iid machines, each working for a random amount of time until breaking down; immediately when this happens, the broken machine is disassembled into two parts, A and B; type A parts and type B parts are repaired in parallel and then, two repaired parts are assembled back into one operating machine. The parts are *exchangeable* in that any repaired A can be assembled with any repaired B. For illustration purposes, it suffices to assume that every repair/assembly station operates as an infinite-server queue (no queues). There are thus 4 activities, plus a single resource (machines) and 5 sub-resources (all involve the same resource) – these are marked in the *activity diagram* in Figure 6. The primitive matrices of the system are

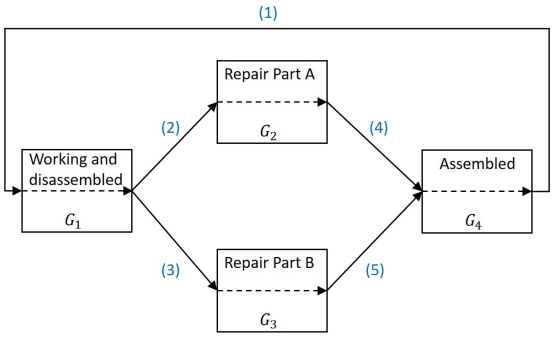


Figure 6 Activity diagram of a (refined) Machine-repair model, with a FJ (disassembly-assembly) structure.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Then RMI (3) can be written down in terms of C, P, G. Observe that the FJ construct is not expressed by RMI. Note also that Assumption 1 is violated: specifically above, the activity "working and disassembled" consumes a single unit of a resource (assembled machine) but produces multiple units of the same resource (Parts A, B). Therefore, Snapshot (7) with $R = [1\ 1\ 1\ 1\ 1]$ is not enough to ensure that the amount of resource engaged in activities is at most the amount of available resource.

Towards a Snapshot inequality that captures the FJ construct, introduce the following parametric family $\mathcal{R}$ of 1×5 matrices: $\mathcal{R} = \{R : \ R = [1\ \alpha\ (1-\alpha)\ \alpha\ (1-\alpha)],\ \alpha \in [0,1]\}$. Such an $\mathcal{R}$ encodes (i) sub-resources 2 and 3 are complementary parts of sub-resource 1, and (ii) sub-resources 4 and 5 correspond to sub-resources 2 and 3, respectively (i.e., 4 and 5 are complementary parts of 1). Let $\{R_v\}$ be the set of extreme points of $\mathcal{R}$. Then, in order to ensures that a plan X does not require more resources than are available, regardless of whether resources are decomposed into parts, Snapshot (7) should be replaced by <u>at least</u> the following inequality:

$$\bigwedge_v \left(R_v \Lambda - R_v C \bar{G} * X \right) \geq 0, \tag{35}$$

where $\wedge$ indicates that inequalities hold for all v; in fact, the last inequality can be rewritten as $\tilde{R} C \bar{G} * X \leq \tilde{R} \Lambda$, for some $\tilde{R}$, by eliminating redundant inequalities. In our Machine-Repair example, $\mathcal{R}$ has two extreme points (corresponding to $\alpha = 0$ and $\alpha = 1$, which are the two distinct paths of the FJ structure), and (35) reduces to

$$\bar{G}_1 * X_1 + \bar{G}_2 * X_2 + \bar{G}_4 * X_4 \leq \Lambda_1 + \Lambda_2 + \Lambda_4,$$
$$\bar{G}_1 * X_1 + \bar{G}_3 * X_3 + \bar{G}_4 * X_4 \leq \Lambda_1 + \Lambda_3 + \Lambda_5.$$

The last two inequalities form the FJ snapshot for our example. In general, however, obtaining RAN snapshot in the presence of FJ constructs, specifically characterizing the set $\mathcal{R}$, requires refined descriptions of activities (beyond C and P). For example, consider an activity that consumes two sub-resources and produces two sub-resources, with all four sub-resources involving the same resource. If one of the two consumed sub-resources is a "part" of the resource (i.e., underwent a fork operation prior to arriving to the present activity), then an activity description must specify whether one or both of the produced sub-resources are "parts" of the resource.

Finally, recall that our FJ constructs are *exchangeable*, which is natural for assembled parts. In service systems, however (e.g. healthcare), FJ constructs are often *non-exchangeable*, as "assembled parts" are usually entities belonging to the same customer (patient). A heuristic for non-exchangeable static FJ is offered in (Carmeli 2020, §5.3.2); it is based on (Lu and Pang 2017, Corollary 3.1), and applied to analyzing patient-flow in an Emergency Department.

5.3. Abandonments: First Steps and Challenges

In this section, we outline challenges in modeling RANs with abandonment of idling resources. Subtleties arise due to the RAN approach of characterizing all feasible plans, or all possible control policies, rather than specific ones. Thus, given that a certain amount of sub-resources should be engaged (X), and information available to the decision-maker, a control policy might specify prioritization of idle sub-resources, and these are precisely the potential abandonments. Different controls can thus lead to different system behavior, even under the same plan X. It follows that, with abandonments, a set of feasible plans should be considered under a specific prioritization policy.

To illustrate our point, it suffices to consider a RAN with a single activity and a single resource. (We shall abuse terminology and refer to members of this resource's pool also as resources.) One is given a cdf G of activity duration and, in addition, cdf F of resource-patience. For simplicity, let $\Lambda \in D_\uparrow$, interpreting it as exogenous resource-arrivals to the system – otherwise one would need to identify which specific resources should be removed from the system, hence decreases in Λ must be consistent with F.

For the above RAN, a prioritization policy and a plan X give rise to a decomposition $\Lambda = X + W + Q$, where plan $X \in D_\uparrow$ is as usual; $W \in D_\uparrow$ cumulatively counts arrivals that will eventually abandon ($F * W \in D_\uparrow$ is actual abandonments); and $Q \in D_+$ counts arrivals that will eventually be served. RMI is then simply $X \leq \Lambda + W$. (Note that $Q + \bar{F} * W$ cumulatively counts arrivals that are yet to be served or abandon.) In particular, consider a policy under which the resource either engages in the activity immediately upon arrival or, alternatively remains

unengaged until abandonment. For such a policy, $Q \equiv 0$ hence $X \preceq \Lambda$, which yields $W = \Lambda - X$. In general, however, determining W for an arbitrary policy might be challenging.

To further amplify some challenges in modeling abandonment, consider the following plan with two jumps, at times T_1 and T_2:

$$X(t) = \begin{cases} 0, & 0 \le t < T_1 \\ X(T_1), & T_1 \le t < T_2 \\ X(T_2), & t \ge T_2. \end{cases}$$

Prior to $t = T_1$, no resources are engaged, and the amount of available resources during time interval $t \in [0, T_1)$ is $(\bar{F} * \Lambda)(t)$, since $(F * \Lambda)(t)$ amount of resources abandon by time t. Thus, a necessary condition for X to be feasible is $X(T_1) \le (\bar{F} * \Lambda)(T_1)$. After $t = T_1$, the evolution depends on the control policy. To wit, introduce $\Lambda_{T_1} := \{\Lambda_{T_1}(t) = \Lambda(t)\mathbb{1}\{t \in [0, T_1]\}, t \ge 0\}$ and $\Lambda_{T_2} := \{\Lambda_{T_2}(t) = \Lambda(t)\mathbb{1}\{t \in (T_1, T_2]\}, t \ge 0\}$. Then under the random-order policy, the amount of available resources at time $t \in (T_1, T_2)$ is given by

$$\left[1 - \frac{X(T_1)}{(\bar{F} * \Lambda_{T_1})(T_1)}\right](\bar{F} * \Lambda_{T_1})(t) + (\bar{F} * \Lambda_{T_2})(t),$$

which limits the magnitude of the second jump in the considered X:

$$(\bar{F} * \Lambda)(T_2) - X(T_1)\frac{(\bar{F} * \Lambda_{T_1})(T_2)}{(\bar{F} * \Lambda_{T_1})(T_1)} \ge X(T_2) - X(T_1).$$

6. Future Research

We have developed a mathematical computation-friendly framework for modelling complex large-scale operations. Research-wise, however, this is only the first step for the "rubber to hit the road"; as suggested sporadically throughout the text, and will be now further outlined, already this step offers uncharted territories, novel directions, open problems, and an inviting horizon for applications.

- **Stochastic RANs §2.3.** Here activity durations are random, as are amounts of sub-resources consumed and produced. A rigorous justification of set-valued fluid and diffusion limits (including single-resourcing) is yet to be established. Such analyses would address multiple time-scales (due to single-resourcing), and various generalized Skorokhod problems (e.g. §4.2.2, (27)), that are special cases of Extended Dynamic Complementarity Problems §3.4.

- **Fluid RANs in support of stochastic RANs.** We envision fluid RANs becoming a natural useful element of the fluid toolbox, both theoretically and practically. For example, in relating fluid and stochastic RANs, operational regimes and bottlenecks of fluid RANs project into the same at the stochastic level; and desirable controls of fluid RANs will approximate

corresponding stochastic controls. Moreover, fluid RANs will provide meta-FSLLNs for the far-reaching variety of stochastic RANs, with the caveat that additional constraints might be needed (e.g. for non-idling controls, or specific policies such as static priorities (32) and JSQ (33)).

• **Non-idling and/or maximal plans/policies.** Maximal plans are often of interest in practice, and so are policies that implement such plans. The class of non-idling policies is appealing, but, as we argued, does not always correspond to maximal plans. On the other hand, for systems like GJNs with non-decreasing amounts of resources, non-idling policies do result in maximal plans. Characterizing RANs for which non-idling policies result in maximal plans remains an open problem, which is in the domain of LCPs §3.3 and DCPs §3.4.3. The problem is open even for basic cases such as $G_t/GI/s_t$ §3.2 or Machine Repair §4.2.

• **More on RANs as CLPs: Duality in finite- and infinite-dimension.** Finite-dimensional duality has often been found central in the analysis of SPNs (e.g. Harrison (2000), Atar et al. (2022)). It is also useful for RANs by enabling the following necessary and sufficient condition for RAN boundedness (see Appendix B for the proof): RMI is locally bounded iff RAN's primitives satisfy

$$\{\Delta \geq 0, [C-P]\Delta \leq 0\} = \{0\};$$

in particular, maximizing over feasible X's then makes sense. (The latter condition captures "no arbitrage opportunities", as in (70) of Harrison (2003)).

RANs also "beg" for infinite-dimensional duality theory. To wit, recall that RMI is the feasibility set of CLP1 in §3.3.3. Now add to the formulation some utility function $c(\cdot) \geq 0$ to maximize over. Then we have the following natural RAN-optimization problem as Primal, with its corresponding Dual right below:

$$\begin{array}{lllll} \text{Primal:} & \max\limits_{X \in D_\uparrow^J} \int_{[0,T]} c(u)\, \mathrm{d}X(u) & \text{s.t.} \int_{[0,t]} K(t-u)\, \mathrm{d}X(u) \leq b(t), & t \in [0,T]; \\ \text{Dual:} & \min\limits_{Y \in D_\uparrow^J} \int_{[0,T]} b(v)\, \mathrm{d}Y(v) & \text{s.t.} \int_{[t,T]} \mathrm{d}Y(v)\, K(v-t) \geq c(t), & t \in [0,T]. \end{array}$$

We conjectured the latter Dual problem from existing infinite-dimensional Duality theory. The conjecture is supported by the facts that the value of any primal-feasible X is upper-bounded by the value of any dual-feasible Y (proved via Fubini's theorem); it follows that if the two values are equal than the corresponding X and Y are both optimal, with the following complementary-slackness:

$$\mathrm{d}Y(t) > 0 \Rightarrow \int_{[0,t]} K(t-u)\, \mathrm{d}X(u) = b(t) \quad \& \quad \mathrm{d}X(t) > 0 \Rightarrow \int_{[t,T]} \mathrm{d}Y(v)\, K(v-t) = c(t), \quad (36)$$

for all $t \in [0, T]$. A complete duality theory, however, is yet to be developed, which should include conditions for strong Duality as in Levinson (1966), Shindin and Weiss (2014). Note that X is a maximum solution iff it is a solution of Primal, for all non-negative c's in a sufficiently rich family; and this has implications to Dual (as in Cottle and Veinott (1972)). Now note that the dual at time t is "peaking" into the future $[t, T]$, which is in concert with Example 5. Duality theory is hence potentially useful for understanding maximality of plans. It is also related to non-idleness since having maximal elements is related to solvability of complementarity problems as linear programs (Cottle and Pang 1978).

- **RAN inference: On mining RANs.** An ultimate goal of the RAN framework is the automatic creation of a RAN model, directly from data-logs, with the least structure pre-enforced. There are known techniques to automatically produce models that represent a given system architecture and its dynamics directly from event logs. Perhaps the most common one is *process mining* (Van Der Aalst 2012), which results in *Petri nets* (Haas 2006). We argue that one could map some of the elements in a Petri net to the RAN matrices R, C, and P, and by doing so, construct the RAN architecture from a Petri net that was automatically produced using process mining. (For example, Petri net transitions could be mapped to RAN activities, Petri net normal input places to RAN sub-resources, and Petri net colors to RAN resources. Note that some Petri nets elements do not have (at least for now) a correspondence in RAN, e.g., inhibitor input places.)

More details can be found in Senderovich et al. (2014, 2015). But regardless of which models are mined from data, they are likely to be mathematically intractable. Hence an accompanying computational framework will we required for generating useful insights.

It is noteworthy that, when fitting RANs to data, or vice versa, one need not worry about which resources are single-resources; rather, one would simply estimate cdf's (and if durations are "short" then be it); similarly for amounts. The value of single-resourcing is most pronounced in theoretical analysis.

- **Beyond present RAN dynamics.** The underlying assumption of the basic RAN is that activities are not interrupted and their duration does not depend on the state of the system. That excludes features such as abandonment, finite-buffers or processor-sharing service disciplines. Some of these can be incorporated with additional constraints relatively easily; for example, an upper bound on $PG * X + \Lambda - CX$ in the case of finite buffers. Others might require that specific operational policies are assumed (e.g. abandonment §5.3), while others still require further research (e.g. processor sharing).

Appendix A: Evolution of Research on Fluid Models/Approximations/Limits

1. **1960 – 70's: Time-varying Engineering Models.** Motivated by congestion problems in traffic flow theory, and abstracting to general congested systems, Newell (1971, 1982) argued that "complexity of the results in the existing literature is way out of proportion to its usefulness (including my own research)"; hence he sought to "turn queueing theory around and point it toward the real world". This led Newell to focus on time-dependent behavior (e.g. rush-hour), where congestion is most prominent (Newell 1968c,b,a, 1973).

2. **1980's: GJNs in heavy-traffic, the Birth of SPNs.** Piecewise-linear fluid approximations were established in Johnson (1983) in order to simplify proofs in Reiman (1984)'s pioneering heavy-traffic analysis of single-server open GJNs. In Chen and Mandelbaum (1991a,c), such fluid models (for both open and closed GJNs) turned into a research focus: first as models in their own right; and second by identifying bottleneck stations, which affected diffusion refinements. The 80's is also when the roots of SPNs were planted in Harrison (1988), which culminated in Dai and Harrison (2020).

3. **Role of fluid models, so far:** The fluid models of the 80's arose as FSLLN limits of processing networks (almost-sure convergence in conventional or single-server heavy-traffic), with parameters that do not vary with time (e.g. in GJN: constant arrival and service rates, as well as in SPNs: constant I/O matrices). The dynamics of such fluid limits is piecewise-linear; in fact linear after some finite time, which we refer to as *static* fluid processes (Chen and Mandelbaum 1991c, Harrison 2003).

Being FSLLN limits, static fluid processes capture first-order behavior (e.g. long-run average, identifying bottlenecks) of their originating stochastic systems. Diffusion refinements, therefore, arose via FCLT limits (weak convergence) that were centered around the static fluid processes; and diffusion parameters were also static in time (e.g. drift, covariance matrix).

4. **1990's: Stability vs. Time-varying Operational Regimes.** Fluid research then diverged into the following three main streams, all in a single-server setup:

(a) **Static fluid models**: Direct continuation of previous research on static fluid models. They now arise from increasingly complex operations (beyond GJNs), as well as in control applications, all in conventional heavy-traffic that, in fact, is characterized via fluid models (e.g. Williams (1998)). Examples include parallel servers (Harrison and López 1999), Brownian networks / SPNs (Harrison and van Mieghem 1997) and asymptotically-optimal control (Harrison and Wein 1990).

(b) **Stability**: Dai's paper (Dai 1995) paved the way for using static fluid models in proving stability of their stochastic origins: the stochastic model is stable (positive-recurrent) if its

corresponding fluid model is stable (fluid empties, that is, fluid paths vanish, within a finite time); note that the converse need not hold (Bramson 1999). The need for a new approach to establish stability arose from the challenge of understanding unstable sub-critical queueing networks (Rybko and Stolyar 1992, Bramson 1994); and this theory developed into Dai and Harrison (2020), that covers stability of almost the full scope of SPNs.

(c) **Time-varying dynamics**: Fluid processes with time-varying parameters arose from modeling service systems that inherently enjoy time-varying (transient) dynamics. They could serve as stand-alone models of such systems, capturing predictable variability in their originating stochastic realities and, hence, determining time-periods when the system is over-, under- or critically loaded (alternating operational regimes). Continuing and supplementing the engineering approach of the 60-70's, research became mathematical, following the path charted by Massey (1985) but within the framework of strong-approximations (Mandelbaum and Massey 1995, Mandelbaum et al. 1998); here fluid limits arise via uniform-acceleration (Massey 2002), in order to preserve time-varying dynamics in the limit (conventional scaling leads to static fluid models).

5. **2000's: Data-based many-server asymptotics/regimes.** The emergence of large service operations, notably telephone call centers with hundreds of agents (Gans et al. 2003, Aksin et al. 2007), and the availability of their data, in unprecedented quantity, quality and granularity (Brown et al. 2005), gave rise to a flourishing research area: many-server asymptotics of the relevant models, as first carried out in the forward-looking (Halfin and Whitt 1981).

Models were both static and time-varying (Feldman et al. 2008), in support of service operations-management (e.g. staffing in the case of the latter paper). Fluid models revealed first-order operational regimes (efficiency-, quality- and quality+efficiency-driven = QED), with their corresponding staffing-levels (Garnett et al. 2002); indeed, fluid models of infinite-server queues, referred to as offered-loads (Whitt 2013), provided the first-order terms of staffing, which were then refined to protect against stochastic variability (e.g. square-root safety staffing (Borst et al. 2004)). All the above many-server models had exponential service-durations; it took a breakthrough to abstract to general service duration (Reed 2009, Whitt 2006). Such a generalization was also data-driven since service-durations were often found to be log-normally distributed.

6. **2010 – 2022: Maturing theory and extensive applications.**

Fluid models have now become central in the toolbox of an operations researcher – in both theory and applications. Theory-wise, this is when the development of measure-valued fluid-models accelerated: following the very early footsteps of Grishechkin (1994), Doytchinov et al. (2001), Gromoll et al. (2002), it started with research such as (Kang and Ramanan 2010, Pang

and Whitt 2010, Kaspi and Ramanan 2011, Liu and Whitt 2012), and its advanced state-of-the-art comes out clearly by simply scholar-searching "queues fluid measure-valued 2022". (Recall that the measure-valued state in fact follows from our significantly simpler RMI (3).)

Application-wise, and again triggered by ample data, this is when healthcare operations started to take center stage (Shi et al. 2014, Armony et al. 2015, Mandelbaum et al. 2019). While such operations are typically not large in terms of their number of servers, many-server theory still remained applicable (de Véricourt and Jennings 2011, Yom-Tov and Mandelbaum 2014), which is due to the fast convergence-rate of many-server limits (Janssen et al. 2011).

Fluid applications have been taking a stronghold beyond healthcare, either breaking new ground or refining views of previous research. Ample examples appear in Zychlinski (2022) and Dai and Harrison (2020). Here are a few additional recent examples: to pool or not to pool (Cao et al. 2021); staffing to stabilize time-varying performance (Liu 2018); algorithms to calculate performance measures in a general time-varying fluid network (Liu and Whitt 2014) (e.g. general service durations, time-varying Markovian routing); fluid limits of parallel many-server queues with delayed information (Whitt 2021); Personalized queues (Mandelbaum and Momčilović 2017); metastability, that arises for example with service-slowdown or -speedup (Dong 2022); and a robust fluid approach to controlling processing networks (Bertsimas et al. 2014), serving later as a benchmark for controls via deep reinforcement learning (Dai and Gluzman 2022). The depth and scope of this body of research emerge clearly from scholar-searching "fluid queues 2022". ▲

This ends our brief historical survey. It helps put RAN in perspective by revealing that RANs constitute a natural timely step in the evolution of fluid research; and by placing RAN characteristics in their historical perspective. We now iterate some of these characteristics, for amplification and completeness:

- RAN features are based on high-resolution data from service operations, for example ACD (Automatic Call Distributor) in call centers (Brown et al. 2005) and RTLS (Real-Time Location Systems) in hospitals (Mandelbaum et al. 2019)). In the latter case, the data also includes appointment books.
- The core-RAN has time-varying dynamics; yet it accommodates also long-run and steady-state evolutions §3.1.
- The core-RAN corresponds to the many-resource operational regime, with general activity-durations; yet it accommodates also the single- or few-server regime (hence RANs generalize SPNs) §3.2.

- The core-RAN is governed by RMI (3), which is both parsimonious and rich-in-scope (e.g. it implies measure-valued state descriptors); with single-resources, their RMI constraints take the form (14) – (15); which is supplemented by Snapshot (13) in case there are single-resources that are closed.

Appendix B: More on RANs as Linear Systems (CLPs)

We viewed RANs as CLPs in §3.3.3, with their Duals conjectured in the fourth bullet point of §6. These discussions will be now supplemented with some observations and open questions:

- *Feasibility*: RANs with $\Lambda \geq 0$ are trivially feasible: the plan $X \equiv 0$ satisfies RMI (3), for all $T \geq 0$. But is there also a non-trivial feasible plan? The conic constraint $X \in D^J_\uparrow$ renders this question interesting, which becomes even more so when adding single-resources (RMI, with (12), is then supplemented by (13)); and what if Λ can go negative (backlog)? Feasibility can be also asked about non-idling (Mandelbaum 1989) or maximum plans (Yang 1993, Chen and Mandelbaum 1991b), with the former raising also questions of solution-uniqueness (Stewart 2009).

- *Static RANs (§3.1 and §3.2.4)*: The above existence and uniqueness problems have analogues for static RANs, in which case the framework for analysis is classical finite-dimensional Linear Programming and Complementarity. Here there is a large relevant body of research, for both LP (e.g. duality) and LCP or ELCP (for example, concerning existence and uniqueness of solutions (Habetler and Szanc 1995)).

- *Static within Dynamic:* Recall that the static model arose either as a limiting or as a stationary RAN. A third way, of computational and practical importance §4.6, would be to approximate a general RAN by a discrete-time RAN: it changes states at epochs $\{n\Delta, n = 1, ...\}$, for some $\Delta > 0$, so that state-changes are, in fact, static RANs (in the spirit of our pre-limit RANs §2.3). For RANs arising from single-class queueing networks, this was done in Mandelbaum (1989) and (Whitt 2002, §14.3) (the latter refers to static RANs as *instantaneous reflection maps*).

- *Redundancy*: Are there redundant constraints? activities? Must all activities be used by some feasible plan? What would be a canonical or minimal RAN representation? For example:

1. If $C_{l,j} = 0$ for every sub-resource l, then activity j does not consume anything. By Assumption 1, it could not produce any resource either, meaning that also $P_{l,j} = 0$ for every l. Activity j is therefore redundant and should be removed from the model.
2. If $[RC]_{k,j} = 0$ for every activity j, then resource k is not consumed by any activity. By Assumption 1, it could not be produced by any activity either. Therefore, resource k is redundant and could be removed.

3. If $\Lambda_l = \infty$ for all l such that $r(l) = k$, then resource k is, in fact, redundant. For example, $X \le A$ is in itself the RMI for the G/G/∞ RAN, and for G/G/s with customers being a single resource: in both RANs, servers are not a resource.
4. Activity j is redundant if $X_j \equiv 0$ for every feasible X. Assuming a static RAN with $\lambda > 0$, this is equivalent to $\{x_j : [C-P]x \le \lambda, x \ge 0\} = \{0\}$, which can be recast as an LP with a useful dual. What would be the analogue for dynamic RANs?

• *Bounded RMIs via Finite-dimensional duality:* A physically plausible property of feasible plans is boundedness, and duality helps establish a bound. Start with observing that the following three conditions (in the appropriate Euclidean spaces) are equivalent:

1. $\{\Delta \ge 0, [C-P]\Delta \le 0\} = \{0\}$ (in the spirit of (70) in Harrison (2003), we refer to this condition as "no arbitrage opportunities");
2. $\{x \ge 0, [C-P]x \le \lambda\}$ is bounded, for all $\lambda \ge 0$ (the bound depends on λ);
3. $\{y \ge 0, y[C-P] \ge e\} \ne \emptyset$ (via LP Duality). Indeed, any of the latter y provides a bound of the form $x^\top e \le y^\top \lambda$, for any x in Item 2.

Now for X satisfying RMI (3): $CX \le P * GX + \Lambda \le PX + \Lambda$ (since $G * X \le X$, for X non-decreasing). We deduce that $e^\top X(t) \le y^\top \Lambda^+(t)$, for all $t \ge 0$ (+ indicating positive part). It follows that RMI is locally bounded if and only if the RAN admits no arbitrage opportunity; in particular, maximizing over feasible X's then makes sense.

Appendix C: 2-Dimensional/Measure-Valued State Description

In Section 2.2.3, it was shown that a plan X determines, jointly with the primitives, the pair $(B(t), Q(t))$, for every $t \ge 0$: this constitute a 1-dimensional state description (Reed 2009, Puhalskii and Reed 2010). In Section 2.2.4, it was demonstrated that X in fact determines refined 2-dimensional, or measured-valued, descriptions (Kaspi and Ramanan 2011, Liu and Whitt 2012, Kang and Pang 2013, Zhang 2013). We now add the details of the descriptions that were left out, while repeating, for completeness, some of §2.2.4

For any $t \ge 0$ and $r \ge 0$, let

$Q_t^{\leftarrow}(\le r) =$ number of delayed sub-resources at time t, out of $Q(t)$,
whose time-in-delay (before t, hence $\leftarrow$) is no more than r time-units;

$Q_t^{\rightarrow}(> r) =$ number of delayed sub-resources at time t, out of $Q(t)$,
whose residual time-in-delay (after t, hence $\rightarrow$) exceeds r;

$B_t^{\leftarrow}(\le r) =$ number of ongoing activities, at time t,
with activity duration that is thus far (before t, hence $\leftarrow$) no more than r.

$B_t^{\rightarrow}(>r)=$ number of ongoing activities at time t,

with residual activity duration (after t, hence $\rightarrow$) that exceeds r.

Similarly, we also introduce $Q_t^{\leftarrow}(>r)$, $Q_t^{\rightarrow}(\leq r)$, $B_t^{\leftarrow}(\geq r)$, $B_t^{\rightarrow}(\leq r)$.

REMARK 13 (**On the above – notation and choice**). *Left-arrows* indicate events that started prior to time t: e.g. in $Q_t^{\leftarrow}$, this event is the start-of-delay for sub-resources that are still delayed (not engaged in activities) at t. *Right-arrows* indicate future events that end after t; e.g. in $Q_t^{\rightarrow}$, this event is the end-of-delay for sub-resources already delayed at t. We listed above 8 performance measures in total: calculating the first 4 (individually displayed) suffices to reveal the principles required for calculating all the others. ▲

Note that Q arises as a *difference of flows*: the total number of 'arrivals to delay' minus the total number of 'departures from delay'. This aggregated counting does not carry with it information about specific delay times between a sub-resource 'arrival' and 'departure'. The latter can be determined by adding order-of-engagement policy for activities, which establishes a bijection ($1-1$ mapping) between 'arrivals' and 'departures' (recall Figure 2). For example, consider a network with a single service activity that consumes two sub-resources: an *"arriving customer"* and an *"available server"*. Unengaged sub-resources from the first type are customers in queue, while unengaged sub-resources of the latter are idle servers. Clearly, a customer time-in-queue depends on the priority policy (e.g., FCFS, LCFS), and similarly, a server idle time depends on the scheduling policy (e.g., LISF, Round-Robin).

The process B can be considered as a *difference of counts*, which is determined by counting the total number of activities that started and where not yet completed. It is fundamentally different from Q in the sense that its corresponding activity durations are known albeit statistically: indeed, its 'activity initiations' X determine its 'activity completions' $G*X$ in a way that, roughly speaking, averages out the fact that each arrival gives rise to a departure after a G-distributed sojourn-time.

We shall now represent each of the above 2-dimensional (t,r)-measures in terms of X and model primitives. For notational convenience, denote $A\coloneqq\Lambda^{+}+PG*X$ and $D\coloneqq\Lambda^{-}+CX$. As explained above, the refined description of Q requires refined information. Here we focus on the following order-of-engagement policy: a starting activity will consume the sub-resources, from each required type, that have been unengaged (delayed) for the longest time (among their type). Moreover, the same criteria is applied to sub-resources that depart the system due to Λ^{-}. We thus call this policy Longest-Delayed-First (LDF). For example, considering again the above network with a single service activity, the LDF policy states that a new service will consume

the customer who has been waiting in queue for the longest time (FCFS), and the server who has been idle for the longest time (LISF).

- *Number of delayed sub-resources (delayed), under Longest-Delayed-First (LDF)*:

$$Q^{\leftarrow}: \quad Q_t^{\leftarrow}(\geq r) = [A(t-r) - D(t)]^+, \quad Q_t^{\leftarrow}(< r) = A(t) - A(t-r) \vee D(t); \tag{37}$$

in particular, $Q_t^{\leftarrow}(\geq 0)_l = Q_t$, and $Q_t^{\leftarrow}(\geq r) = 0$, for $r > t$.

$$Q^{\rightarrow}: \quad Q_t^{\rightarrow}(> r) = [A(t) - D(t+r)]^+, \quad Q_t^{\rightarrow}(\leq r) = D(t+r) \wedge A(t) - D(t); \tag{38}$$

in particular, $Q_t^{\rightarrow}(> 0) = Q_t$, $Q_t^{\rightarrow}(\leq 0) = 0$, and $Q_t^{\rightarrow}(\leq r) = Q_t$, for $r \geq r(t)$; here $r(t)$ is an L-vector representing the delay time of sub-resources that started the delay at time t: $r_l(t)$ is the largest r for which $D_l(t+r) = A_l(t)$, which is the longest delay among all sub-resources of type l that are not engaged by activities at time t (thus LDF).

Explanation for Q-calculations (see Figure 4 for a graphical demonstration), assuming LDF, for every sub-resource l:

—If $D_l(t) < A_l(t-r)$: $A_l(t) - A_l(t-r)$ are waiting less than r, $\Delta A_l(t-r)$ are waiting r, and $A_l(t-r) - D_l(t)$ are waiting r or more.

—If $D_l(t) = A_l(t-r)$: $A_l(t) - A_l(t-r)$ are waiting less than r, no one waits r or more.

—If $D_l(t) > A_l(t-r)$: All of $Q_l(t)$ are waiting less than r, no one waits r or more.

(We elaborated on only $Q^{\leftarrow}$ since $Q^{\rightarrow}$ is handled similarly.)

- *Number of ongoing activities*:

$$B_t^{\leftarrow}(\leq r) = \int_{[t-r,t]} \bar{G}(t-u)\, \mathrm{d}X(u)\,,$$
$$B_t^{\rightarrow}(> r) = \int_{[0,t]} \bar{G}(r+t-u)\, \mathrm{d}X(u)\,,\ B_t^{\rightarrow}(> 0) = B_t.$$

It follows that

$$B_t^{\leftarrow}(> r) = \int_{[0,t-r)} \bar{G}(t-u)\, \mathrm{d}X(u) = \int_{[0,t-r]} \bar{G}(t-u)\, \mathrm{d}X(u) - \Delta X(t-r)\, \bar{G}(r)\,,$$
$$B_t^{\rightarrow}(\leq r) = \int_{[0,t]} G(r+t-u)\, \mathrm{d}X(u) - (G*X)(t)\,,\ B_t^{\rightarrow}(\leq 0) = 0\,.$$

Explanation for B-calculations: first, why does $B_t^{\rightarrow}(\leq 0) = 0$? Ongoing activities with residual duration 0 are counted in X and in $G*X$, thus their number cancels out in $B = \bar{G}*X$. To explain $B_t^{\leftarrow}(\leq r)$ (other explanations are similar), consider an ongoing activity at time t: in order for its duration thus far to not exceed r, it should have started at time $u \in [t-r, t]$.

Appendix D: The $G_t/GI/s_t$ Queue as a RAN

This appendix constitutes a compact systematic account of the $G_t/GI/s_t$ fluid RAN. Most has already appeared above yet it is worthwhile including here. Indeed, it amplifies the (added) value of the RAN-viewpoint, by gathering abstract concepts and scattered facts under a concrete single "roof".

The $G_t/GI/s_t$ fluid RAN is conceptualized as an approximation to a stochastic $G_t/GI/s_t$ queue, in which the arrival process and the number of servers are scaled up (multiplied at all $t \geq 0$) by η, $\eta \uparrow \infty$, while the service cdf G is held fixed (Puhalskii and Reed 2010, Liu and Whitt 2012)[1]. Formally §2.3, random sets of pre-limit feasible plans converge, as $\eta \uparrow \infty$, to RMI (set of feasible plans) of the $G_t/GI/s_t$ RAN. This RAN has 2 resources (open customers, closed servers), 1 activity (service), 2 sub-resources (servers and customers, both at state "awaiting service"); and it is characterized by the primitives

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} s(\cdot) \\ A(\cdot) \end{bmatrix}, \quad \text{cdf } G(\cdot).$$

Here $s(\cdot) \in D^1_+$ is a staffing function (e.g. $s(t) = s\mathbb{1}\{t \in [0, H)\}$, $t \geq 0$, corresponds to $s \in \mathbb{R}^1_+$ servers that arrive, at $t = 0$, to a shift of length H, at the end of which all depart); $A \succeq 0$ is exogenous arrivals ($A(t)$ is the (average of the stochastic RAN) cumulative amount of arrivals during the time interval $[0, t]$); and G is the cdf of *service duration*, with $G(0-) = 0$ and, to avoid triviality, $G(0) < 1$.

We now list the main features of the $G_t/GI/s_t$ fluid RAN:

- RMI: $X \succeq 0$, s.t. $X \leq A$, $X \leq G * X + s$.
- *Slacks*: $Q := A - X =$ amount of queueing customers; $I := s - \bar{G} * X =$ amount of idling servers,
- *RMI via slacks*: $X \succeq 0$, s.t. $Q \geq 0$, $I \geq 0$.
- *Busy processes, Total in system*: Recall the busy process $B := X - G * X = \bar{G} * X =$ amount of servers (customers) engaged in service; $B \geq 0$ is implied by $X \succeq 0$. Introduce also $L := A - G * X =$ amount of customers within the system (waiting + served). Then note that $L - s = Q - I$, hence $Q \geq (L - s)^+$ and $I \geq (L - s)^-$. It follows that $L = \bar{G} * A + G * Q \geq Q_\infty + G * (L - s)^+$, where $Q_\infty = \bar{G} * A$ is the corresponding $G_t/GI/\infty$ RAN.
- *RMI via busy-processes*: A function B will be called *RMI-feasible busy-process* if $s \geq B$ and $A \geq [I - G]^{-1} * B \succeq 0$; in which case B is exhaustive. Then B is an RMI-feasible busy-process iff $X := [I - G]^{-1} * B$ is an RMI-feasible plan; in which case $X \succeq 0$ is a feasible plan.

[1] Fluid SPNs, on the other hand, arise from scaling (multiplying) the arrival rate $\lambda(t)$ and the service rate $\mu(t)$ by $\eta \uparrow \infty$, while maintaining the number of servers s fixed. This is referred to as *uniform acceleration* since it amounts to accelerating, uniformly at all times $t \geq 0$, both the arrival and service rates (Massey 1985, Mandelbaum and Massey 1995).

- Plan X is *non-idling* (work-conserving) iff $Q \wedge I \equiv 0$; iff $(A - X) \wedge (s - B) \equiv 0$, for B above. Such X exists iff it is maximum among RMI-feasible plans; iff B is maximum among exhaustive busy-processes; iff $Q \equiv (L - s)^+$ and $I \equiv (L - s)^-$, for L above; iff X is the unique fixed-point of the mapping $T(X) := A \wedge (s + G * X)$ which, furthermore, is in $D^1_\uparrow$; iff $L \equiv Q_\infty + G * (L - s)^+$, in concert with the fluid results in Reed (2009).
- The offered-plans associated with (waiting) customers are defined by $X \leq A$ (set $s = \infty$ in RMI), and the maximal plan is $X^m = A$ in that case. The *offered-load* associated with this plan is $s^* = A - G * A$ busy servers with no waiting and no idleness, which is exhaustive ($[I - G]^{-1} * s^* = A \succeq 0$); s^* is critical in that more staffing creates idleness and less gives rise to queueing. (Diffusion refinements, e.g. square-root staffing, use s^* as a QED staffing-skeleton (Borst et al. 2004).)
- The offered-plans associated with (idle) servers are defined by $X \leq s + G * X$ (set $A = \infty$ in RMI). If $[I - G]^{-1} * s \succeq 0$, the maximal plan is $X^m = [I - G]^{-1} * s$. The offered-load associated with this maximal plan (to be referred to as *offered capacity*) is $A^* = [I - G]^{-1} * s$ arrivals with no waiting and no idleness; it is critical in that less arrivals creates idleness and more gives rise to queues. (In analogy to server staffing, the offered-capacity A^* can be used as as skeleton for customer-appointments (Huang et al. 2022).)
- Fix a service distribution G with $G(0) < 1$ and a staffing function s. Then for any arrival function A, there exists a unique RCLL non-negative X that solves $X = A \wedge (s + G * X) =: T(X)$, denote it X_A. In other words, X_A is the unique fixed point of T which, furthermore, enjoys the following properties (complementing §4.1):
  - *Initial condition*: $X(0) = A(0) \wedge [1 - G(0)]^{-1} s(0)$.
  - *Monotone staffing*: If $s \succeq 0$ than X_A is a feasible plan for all A.
  - *Maximal*: If X_A is indeed feasible and X is another feasible plan, then $X_A \geq X$.
- *Single-resourcing customers* leads to the G/GI/∞ RAN: $X \leq A$, with zero-duration services.

Single-resourcing servers was already described in Section 4.1.

Appendix E: Dynamic Allocation of Ground-robots to Stations – Continuing §4.6

In Section 4.6, we described a RAN model that has been used to analyze and optimize a robotic order-fulfillment center. A fundamental challenge when operating such a center is the dynamic allocation of ground-robots (GRs) to stations: namely how many such robots should be allocated to each station, given specific site configurations (combinations of stations operating in different modes, for example picking items to fulfill new orders, decanting inventory into empty totes, dispatching full order totes, etc). Allocating too few GRs to a station would cause station

starvation, which reduces station throughput. Allocating too many, on the other hand, would result in long waiting time of GRs for the station: this increases robots' cycle time which reduces system throughput; it also hurts flexibility, for example responding to changes in configuration due to changes in the number of stations or their modes.

The first algorithm of GR allocation was static and mostly heuristic: for each station per each mode, and regardless of site configurations, it calculated a prefixed range of feasible GR allocations (interval of minimal to maximal number of GRs). This was replaced by a RAN-based control, which dynamically adapts GR allocations to changing site configurations.

RAN-based decisions on GR allocations must be made within milliseconds (upon need). This necessitates a trade-off between model complexity (catering to relevant details) vs. computational feasibility (solution in a timely manner). Here modeling amounts to specifying, from a superset of resources and activities, those constituting the currently-relevant RAN model (while the others are disregarded or aggregated). Computational constraints then render infeasible a dynamic complex RAN (hundreds to thousands activities and sub-resources), since it can not be optimized within milliseconds. The challenge is then addressed via approximating the dynamic RAN by a "piece-wise static RAN", which is in fact a sequence of static RANs (described momentarily) such that each is applied, consecutively, over a short rolling time-horizon. It is notable that already this sub-optimal approach shortened GR cycle times by over 30%, and increased total site throughput by up to 8%.

The Static RAN. The input for each static RAN is a site configuration, plus the numbers of available ground- and lift-robots (that are dynamically changing as some robots require charging, routine maintenance, etc.), plus activity durations (based on historical data that is updated periodically). The goal is to maximize stations' throughput (formalized below); and since each station is naturally modeled as a closed single-server resource ($\in \mathtt{H}$), this entails maximizing the utilization of its operator (either human or robotic-arm).

Figure 7 depicts a partial activity diagram of a RAN that models (to facilitate readability) only two types of stations and a single racking unit. Examples of activities (represented by rectangles in the figure) include loading an inventory tote for picking, or loading an empty tote for decanting, unloading a tote, returning a tote back to the shelving units, picking an item from a tote, etc.

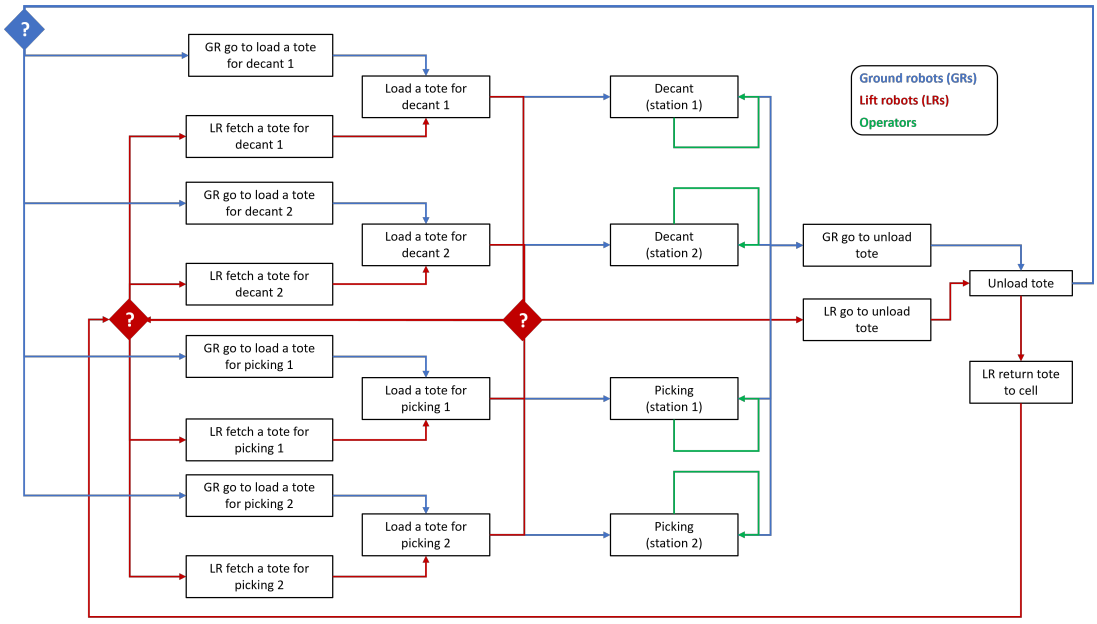


Figure 7 RAN diagram of an order-fulfilment robotic system (partial). Activities are represented by rectangles, arrows represent sub-resources, and different colors correspond to different resources. Question-mark rhombuses represent decision points.

Each static RAN, or rather its primitives C, P, $R, \mathcal{A}$ and $\Lambda(\infty)$, is created automatically from data (with different site configurations resulting in different RAN models). One then seeks to maximize the weighted sum of station throughput rates x, or formally

$$\begin{aligned} \max_{x} \; & w^{\top} x \\ \text{s.t.} \; & \left(RC\mathcal{A}x\right)_k \le R\Lambda(\infty)_k, \;\; \text{for } k \notin \mathtt{H},\, k \text{ closed}, \\ & \left(RC\mathcal{A}x\right)_k \le \hat{\lambda}_k, \;\; \text{for } k \in \mathtt{H}, \\ & (Cx)_l \le (Px)_l + \lambda_l, \;\; \text{for } \lambda_l > 0, \\ & (Cx)_l = (Px)_l, \;\; \text{for } \lambda_l = 0, \end{aligned} \tag{39}$$

where w is the non-negative vector of weights. For the single-server stations we have $\hat{\lambda}_k = 1$. GRs and lift robots (LRs) are modeled as closed, many-servers, resources, for which $R\Lambda(\infty)$ is the number of GRs/LRs available on site. All sub-resources involved with GRs and LRs, such as *"GR ready for load"* or *"LR ready for unload"*, have $\lambda_l = 0$. Finally, open resources (and their sub-resources) represent external demand (e.g., orders), for which the relevant λ_l's is positive.

Given an optimal x^*, $C\mathcal{A}x^*$ provides the number of sub-resources that participate in ongoing activities which, in particular, includes the number of ground-robots allocated to the different stations; in practice, and to hedge against uncertainty, one adds to this number some empirically-grounded square-root factor.

There are additional practical challenges that arose during implementations of the above, for example optimal solutions may turn out "unfair" in that the utilization of stations with human operators are not equal across stations. The RAN framework proved rich enough to accommodate this challenge (generate fair solutions), and ample others.

Acknowledgments

The RAN framework is data-based: our data-home is the Technion SEELab; its data has been collected, maintained and analyzed by Ella Nadjharov, Igor Gavako, and the late Valery Trofimov; and the specific data-source for the present paper is the Dana Farber Cancer Institute (DFCI), in which we are grateful to Ryan Leib, Sarah Kadish and Craig Bunnell for making our data-parnership possible.

The seeds of RANs were planted in a research proposal funded by the Binational Science Foundation (BSF) 2014 - 2017, granted to MA, AM, PM. It then developed into the PhD of Carmeli (2022), advised by AM, PM and GY, and generously supported by scholarships from the Technion. Further support of RAN research was provided to AM by the Israeli Science Foundation (ISF) 2018 - 2021, which continues in an ongoing ISF 2022 - 2025.

References

Afeche P, Liu Z, Maglaras C (2018) Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. *Columbia Business School Research Paper* 18–19. 11

Aksin Z, Armony M, Mehrotra V (2007) The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management* 16(6):665–688. 63

Anderson EJ, Nash P (1987) *Linear Programming in Infinite-Dimensional Spaces: Theory and Applications* (John Wiley & Sons). 34

Anderson EJ, Philpott AB (1994) On the solutions of a class of continuous linear programs. *SIAM Journal on Control and Optimization* 32(5):1289–1296. 34

Armony M, Israelit S, Mandelbaum A, Marmor YN, Tseytlin Y, Yom-Tov GB (2015) On patient flow in hospitals: A data-based queueing-science perspective. *Stochastic Systems* 5(1):146–194. 5, 64

Atar R (2005) Scheduling control for queueing systems with many servers: Asymptotic optimality in heavy traffic. *Annals of Applied Probability* 15(4):2606–2650. 12

Atar R, Castiel E, Reiman MI (2022) Parallel server systems under an extended heavy traffic condition: A lower bound, arXiv:2201.07855. 60

Aubin J, Cellina A (1984) *Differential Inclusions: Set-Valued Maps and Viability Theory*. Grundlehren der mathematischen Wissenschaften (Springer-Verlag). 27

Aveklouris A, DeValve L, Ward AR, Wu X (2021) Matching impatient and heterogeneous demand and supply. *arXiv preprint arXiv:2102.02710* . 11

Azriel D, Feigin PD, Mandelbaum A (2019) Erlang-S: A data-based model of servers in queueing networks. *Management Science* 65(10):4607–4635. 6

Bellman R (1953) Bottleneck problems and dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America* 39(9):947. 34

Benjaafar S, Liu H, Wu S (2022) Dimensioning on-demand vehicle sharing systems. *Management Science* 68(2):1218–1232. 10, 50

Bertsimas D, Nasrabadi E, Paschalidis IC (2014) Robust fluid processing networks. *IEEE Transactions on Automatic Control* 60(3):715–728. 64

Borst S, Mandelbaum A, Reiman MI (2004) Dimensioning large call centers. *Operations Research* 52(1):17–34. 63, 70

Bramson M (1994) Instability of fifo queueing networks. *Annals of Applied Probability* 4(2):414–431. 63

Bramson M (1999) A stable queueing network with unstable fluid model. *Annals of Applied Probability* 818–853. 63

Braverman A, Dai JG, Liu X, Ying L (2019) Empty-car routing in ridesharing systems. *Operations Research* 67(5):1437–1452. 10, 12, 50

Brown L, Gans N, Mandelbaum A, Sakov A, Shen H, Zeltyn S, Zhao L (2005) Statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association* 100(469):36–50. 63, 64

Cao P, He S, Huang J, Liu Y (2021) To pool or not to pool: Queueing design for large-scale service systems. *Operations Research* 69(6):1866–1885. 64

Carmeli N (2015) *Modeling and Analyzing IVR Systems, as a Special Case of Self-Services*. Master's thesis, Technion – Israel Institute of Technology. 9

Carmeli N (2020) *Data-Based Resource-View of Service Networks: Performance Analysis, Delay Prediction and Asymptotics*. Ph.D. thesis, Technion – Israel Institute of Technology. 9, 35, 58

Carmeli N (2022) *Data-Based Resource-View of Service Networks: Performance Analysis, Delay Prediction and Asymptotics*. Ph.D. thesis, Technion – Israel Institute of Technology, URL https://iew.technion.ac.il/serveng/References/Nitzan_PhD_Final.pdf. 53, 73

Chen H, Mandelbaum A (1991a) Discrete flow networks: Bottleneck analysis and fluid approximations. *Mathematics of Operations Research* 16(2):408–446. 30, 37, 38, 48, 49, 50, 62

Chen H, Mandelbaum A (1991b) Leontief systems, RBV's and RBM's. *Applied Stochastic Analysis* 1–43. 50, 65

Chen H, Mandelbaum A (1991c) Stochastic discrete flow networks: Diffusion approximations and bottlenecks. *Annals of Probability* 1463–1519. 10, 36, 48, 62

Chen H, Mandelbaum A (1994) Hierarchical modeling of stochastic networks, part I: Fluid models. *Stochastic Modeling and Analysis of Manufacturing Systems*, 47–105 (Springer). 12

Chen H, Yao DD (2013) *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, volume 46 (Springer Science & Business Media). 5, 47

Chen J, Dong J, Shi P (2020) A survey on skill-based routing with applications to service operations management. *Queueing Systems* 96(1):53–82. 12

Chen M, Baron O, Mandelbaum A, Wang J, Yom-Tov G, Arber N (2022) Waiting experience in open-shop service networks: Improvements via flow analytics & automation. *Available at SSRN* . 53

Cottle RW, Dantzig GB, et al. (1968) Complementary pivot theory of mathematical programming. *Mathematics of the Decision Sciences, part* 1(11). 39

Cottle RW, Pang JS (1978) On solving linear complementarity problems as linear programs. *Complementarity and Fixed Point Problems*, 88–107 (Springer). 61

Cottle RW, Pang JS, Stone RE (2009) *The Linear Complementarity Problem* (SIAM). 40

Cottle RW, Veinott AF (1972) Polyhedral sets having a least element. *Mathematical Programming* 3(1):238–249. 42, 61

Dai J, Dieker A, Gao X (2014) Validity of heavy-traffic steady-state approximations in many-server queues with abandonment. *Queueing Systems* 78(1):1–29. 28

Dai JG (1995) On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *Annals of Applied Probability* 5(1):49–77. 9, 62

Dai JG, Gluzman M (2022) Queueing network controls via deep reinforcement learning. *Stochastic Systems* 12(1):30–67. 64

Dai JG, Harrison J (2020) *Processing Networks: Fluid Models and Stability* (Cambridge University Press), ISBN 9781108488891. 9, 10, 11, 12, 40, 42, 62, 63, 64

Dantzig GB (1963) *Linear Programming and Extensions* (Princeton University Press). 39

De Schutter B, De Moor B (1995) The extended linear complementarity problem. *Mathematical Programming* 71(3):289–325. 40

De Schutter B, De Moor B (1998) The linear dynamic complementarity problem is a special case of the extended linear complementarity problem. *Systems & Control Letters* 34(1-2):63–75. 40

de Véricourt F, Jennings OB (2011) Nurse-to-patient ratios in hospital staffing: A queueing perspective. *Operations Research* 59(6):1320–1331. 64

Dermitzakis V, Politis K (2022) Monotonicity properties for solutions of renewal equations. *Statistics & Probability Letters* 180:109226. 21

Dong J (2022) Metastability in queues. *Queueing Systems* 100(3-4):413–425. 64

Doytchinov B, Lehoczky J, Shreve S (2001) Real-time queues in heavy traffic with earliest-deadline-first queue discipline. *Annals of Applied Probability* 332–378. 63

Eick SG, Massey WA, Whitt W (1993) The physics of the $M_t/G/\infty$ queue. *Operations Research* 41(4):731–742. 35

Feldman Z, Mandelbaum A, Massey WA, Whitt W (2008) Staffing of time-varying queues to achieve time-stable performance. *Management Science* 54(2):324–338. 63

Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management* 5(2):79–141. 63

Garnett O, Mandelbaum A (2000) An introduction to skills-based routing and its operational complexities. *Teaching notes* 114. 45

Garnett O, Mandelbaum A, Reiman M (2002) Designing a call center with impatient customers. *Manufacturing & Service Operations Management* 4(3):208–227. 11, 63

George DK, Xia CH (2011) Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research* 211(1):198–207. 10

Grishechkin S (1994) GI/G/1 processor sharing queue in heavy traffic. *Advances in Applied Probability* 26(2):539–555. 63

Gromoll HC, Puha AL, Williams RJ (2002) The fluid limit of a heavily loaded processor sharing queue. *Annals of Applied Probability* 12(3):797–859. 63

Gurvich I, van Mieghem JA (2015) Collaboration and multitasking in networks: Architectures, bottlenecks, and capacity. *Manufacturing & Service Operations Management* 17(1):16–33. 12

Gurvich I, van Mieghem JA (2018) Collaboration and multitasking in networks: Prioritization and achievable capacity. *Management Science* 64(5):2390–2406. 12

Haas PJ (2006) *Stochastic Petri Nets: Modelling, Stability, Simulation* (Springer Science & Business Media). 61

Habetler G, Szanc B (1995) Existence and uniqueness of solutions for the generalized linear complementarity problem. *Journal of Optimization Theory and Applications* 84(1):103–116. 65

Halfin S, Whitt W (1981) Heavy-traffic limits for queues with many exponential servers. *Operations Research* 29(3):567–588. 11, 63

Hall RW (1991) *Queueing Methods: For Services and Manufacturing* (Pearson College Division). 8

Harrison JM (1988) Brownian models of queueing networks with heterogeneous customer populations. *Stochastic Differential Systems, Stochastic Control Theory and Applications*, 147–186 (Springer). 5, 9, 11, 62

Harrison JM (2000) Brownian models of open processing networks: Canonical representation of workload. *Annals of Applied Probability* 75–103. 9, 60

Harrison JM (2002) Stochastic networks and activity analysis. *Translations of the American Mathematical Society-Series 2* 207:53–76. 5, 9, 34

Harrison JM (2003) A broader view of Brownian networks. *Annals of Applied Probability* 13(3):1119–1150. 5, 9, 60, 62, 66

Harrison JM, López MJ (1999) Heavy traffic resource pooling in parallel-server systems. *Queueing systems* 33(4):339–368. 62

Harrison JM, van Mieghem JA (1997) Dynamic control of Brownian networks: State space collapse and equivalent workload formulations. *Annals of Applied Probability* 747–771. 62

Harrison JM, Wein LM (1990) Scheduling networks of queues: Heavy traffic analysis of a two-station closed network. *Operations Research* 38(6):1052–1064. 62

Huang J, Mandelbaum A, Momčilović P (2022) Appointment-driven service systems with ample servers over alternating operational regimes (QED, ED, QD). *In Preparation* . 70

Ingolfsson A, Akhmetshina E, Budge S, Li Y, Wu X (2007) A survey and experimental comparison of service-level-approximation methods for nonstationary $M(t)/M/s(t)$ queueing systems with exhaustive discipline. *INFORMS Journal on Computing* 19(2):201–214. 20

Janssen A, Van Leeuwaarden JS, Zwart B (2011) Refining square-root safety staffing by expanding Erlang C. *Operations Research* 59(6):1512–1522. 64

Johnson DP (1983) *Diffusion approximations for optimal filtering of jump processes and for queueing networks* (PhD, The University of Wisconsin-Madison). 62

Kang W, Pang G (2013) Fluid limit of a many-server queueing network with abandonment. *Preprint* . 22, 66

Kang W, Ramanan K (2010) Fluid limits of many-server queues with reneging. *Annals of Applied Probability* 20(6):2204–2260. 63

Kaspi H, Mandelbaum A (1992) Regenerative closed queueing networks. *Stochastics: An International Journal of Probability and Stochastic Processes* 39(4):239–258. 12

Kaspi H, Ramanan K (2011) Law of large numbers limits for many-server queues. *Annals of Applied Probability* 21(1):33–114. 22, 64, 66

Koopmans T (1951) *Activity Analysis of Production and Allocation* (Wiley). 6, 9

Krichagina EV, Puhalskii AA (1997) A heavy-traffic analysis of a closed queueing system with a GI/∞ service center. *Queueing Systems* 25(1):235–280. 6

Leontief W (1986) *Input-Output Economics* (Oxford University Press). 6

Levinson N (1966) A class of continuous linear programming problems. *Journal of Mathematical Analysis and Applications* 16(1):73–83. 39, 61

Liu Y (2018) Staffing to stabilize the tail probability of delay in service systems with time-varying demand. *Operations Research* 66(2):514–534. 64

Liu Y, Whitt W (2012) The G_t/GI/s_t+GI many-server fluid queue. *Queueing Systems* 71(4):405–444. 6, 12, 20, 22, 44, 64, 66, 69

Liu Y, Whitt W (2014) Algorithms for time-varying networks of many-server fluid queues. *INFORMS Journal on Computing* 26(1):59–73. 47, 64

Lu H, Pang G (2017) Heavy-traffic limits for a fork-join network in the Halfin-Whitt regime. *Stochastic Systems* 6(2):519–600. 58

Mandelbaum A (1989) The dynamic complementarity problem. *Unpublished manuscript* . 40, 42, 44, 65

Mandelbaum A, Massey WA (1995) Strong approximations for time-dependent queues. *Mathematics of Operations Research* 20(1):33–64. 36, 63, 69

Mandelbaum A, Massey WA, Reiman MI (1998) Strong approximations for Markovian service networks. *Queueing Systems* 30(1-2):149–201. 63

Mandelbaum A, Momčilović P (2017) Personalized queues: the customer view, via a fluid model of serving least-patient first. *Queueing Systems* 87(1):23–53. 64

Mandelbaum A, Momčilović P, Trichakis N, Kadish S, Leib R, Bunnell CA (2019) Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. *Management Science* 66(1):243–270. 64

Mandelbaum A, Reiman MI (1998) On pooling in queueing networks. *Management Science* 44(7):971–981. 9

Mandelbaum A, Stolyar AL (2004) Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research* 52(6):836–855. 12

Massey WA (1985) Asymptotic analysis of the time dependent m/m/1 queue. *Mathematics of Operations Research* 10(2):305–327. 11, 63, 69

Massey WA (2002) The analysis of queues with time-varying rates for telecommunication models. *Telecommunication Systems* 21(2):173–204. 63

Momčilović P, Motaei A (2018) An analysis of a large-scale machine repair model. *Stochastic Systems* 8(2):91–125. 10, 12

Newell GF (1968a) Queues with time-dependent arrival rates. iii—a mild rush hour. *Journal of Applied Probability* 5(3):591–606. 62

Newell GF (1968b) Queues with time-dependent arrival rates. ii—the maximum queue and the return to equilibrium. *Journal of Applied Probability* 5(3):579–590. 62

Newell GF (1968c) Queues with time-dependent arrival rates i—the transition through saturation. *Journal of Applied Probability* 5(2):436–451. 62

Newell GF (1971) *Applications of Queueing Theory* (Chapman and Hall, London). 62

Newell GF (1973) *Approximate Stochastic Behavior of n-Server Service Systems with Large n*, volume 1 (Springer Berlin Heidelberg). 5, 10, 62

Newell GF (1982) *Applications of Queueing Theory, Second Edition* (Chapman and Hall, London). 62

Özkan E, Ward AR (2020) Dynamic matching for real-time ride sharing. *Stochastic Systems* 10(1):29–70. 11

Pang G, Whitt W (2010) Two-parameter heavy-traffic limits for infinite-server queues. *Queueing Systems* 65(4):325–364. 63

Prisgrove LA (1987) Closed queueing networks with multiple servers: Transient and steady-state approximations. Technical report, STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH. 10, 12

Puhalskii AA, Reed JE (2010) On many-server queues in heavy traffic. *Annals of Applied Probability* 20(1):129–195. 12, 21, 66, 69

Reed J (2009) The G/GI/N queue in the Halfin–Whitt regime. *Annals of Applied Probability* 19(6):2211–2269. 6, 21, 44, 63, 66, 70

Reich M (2012) *The offered-load process: Modeling, inference and applications*. Master's thesis, Technion-Israel Institute of Technology, Faculty of Industrial Engineering and Management. 34

Reiman MI (1984) Open queueing networks in heavy traffic. *Mathematics of Operations Research* 9(3):441–458. 62

Rybko AN, Stolyar AL (1992) Ergodicity of stochastic processes describing the operation of open queueing networks. *Problemy Peredachi Informatsii* 28(3):3–26. 63

Schooley B, Almont M, Moody B, Bischoff A, Mullins D, Patel N (2019) Telehealth decision support to optimize safety and quality prison inmate healthcare. *INFORMS Healthcare Conference*. 11

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014) Queue mining–predicting delays in service processes. *International Conference on Advanced Information Systems Engineering*, 42–57 (Springer). 61

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2015) Queue mining for delay prediction in multi-class service processes. *Information Systems* 53:278–295. 61

Serfozo R (2009) *Basics of Applied Stochastic Processes* (Springer Science & Business Media). 20

Shapiro A (2001) On duality theory of conic linear problems. Goberna M, Lopez M, eds., *Semi-infinite Programming*, volume 57 of *Nonconvex Optimization and Its Applications*, 135–165 (Springer). 39

Shi P, Dai J, Ding D, Ang SK, Chou M, Jin X, Sim J (2014) Patient flow from emergency department to inpatient wards: Empirical observations from a Singaporean hospital. *Available at SSRN 2517050* . 64

Shindin E, Masin M, Weiss G, Zadorojniy A (2021) Revised sclp-simplex algorithm with application to large-scale fluid processing networks. *arXiv preprint arXiv:2103.04405* . 34

Shindin E, Weiss G (2014) Symmetric strong duality for a class of continuous linear programs with constant coefficients. *SIAM Journal on Optimization* 24(3):1102–1121, URL http://dx.doi.org/10.1137/130921532. 61

Shindin E, Weiss G (2015) Structure of solutions for continuous linear programs with constant coefficients. *SIAM Journal on Optimization* 25(3):1276–1297. 34

Shindin E, Weiss G (2020) A simplex-type algorithm for continuous linear programs with constant coefficients. *Mathematical Programming* 180(1):157–201. 34

Stewart DE (2006) Convolution complementarity problems with application to impact problems. *IMA Journal of Applied Mathematics* 71(1):92–119. 40

Stewart DE (2009) Uniqueness for solutions of differential complementarity problems. *Mathematical Programming* 118(2):327–345. 40, 65

Van Der Aalst W (2012) Process mining. *Communications of the ACM* 55(8):76–83. 61

Van Leeuwaarden JS, Mathijsen BW, Zwart B (2017) Economies-of-scale in resource sharing systems: Tutorial and partial review of the QED heavy-traffic regime. *arXiv preprint arXiv:1706.05397* . 11

Whitt W (2002) *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues* (Springer Science & Business Media). 40, 50, 65

Whitt W (2006) Fluid models for multiserver queues with abandonments. *Operations Research* 54(1):37–54. 63

Whitt W (2013) Offered load analysis for staffing (OM Forum). *Manufacturing & Service Operations Management* 15(2):166–169. 35, 43, 63

Whitt W (2018) Time-varying queues. *Queueing Models and Service Management* 1(2):79–164. 11, 35

Whitt W (2021) On the many-server fluid limit for a service system with routing based on delayed information. *Operations Research Letters* 49(3):316–319. 64

Williams R (1998) Some recent developments for queueing networks. *Probability Towards 2000*, 340–356 (Springer). 62

Williams R (2017) Skorokhod Problems. URL: https://mathweb.ucsd.edu/~williams/courses/m28917/skorokhod32017.pdf. Last visited June 5, 2022. 44, 46

Williams RJ (2000) On dynamic scheduling of a parallel server system with complete resource pooling. McDonald D, Turner S, eds., *Analysis of Communication Networks: Call Centres, Traffic and Performance*, volume 28 of *Fields Institute Communications*, 49–71 (American Mathematical Society). 12, 52

Yang P (1993) Least controls for a class of constrained linear stochastic systems. *Mathematics of Operations Research* 18(2):275–291. 42, 65

Yom-Tov GB, Mandelbaum A (2014) Erlang-R: A time-varying queue with reentrant customers, in support of healthcare staffing. *Manufacturing & Service Operations Management* 16(2):283–299. 64

Zhang J (2013) Fluid models of many-server queues with abandonment. *Queueing Systems* 73(2):147–193. 22, 66

Zychlinski N (2022) Applications of fluid models in service operations management. URL https://noazy.net.technion.ac.il/files/2022/05/Applications_of_Fluid_Models_260522.pdf. 11, 64