Performance analysis of stochastic networks

Boxma, O.; Kapodistria, S.; Mandjes, M.

Publication date
2015

Document Version
Final published version

Published in
Nieuw Archief voor Wiskunde

Citation for published version (APA):
http://www.nieuwarchief.nl/serie5/toonnummer.php?deel=16&nummer=3&taal=0

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Performance analysis of stochastic networks

Queues are everywhere, and have significant impact on how we experience everyday’s life. The mathematical analysis of queueing, rooted in the interface of probability theory and operations research, is a strongly developed branch of research. Onno Boxma, Stella Kapodistria, Michel Mandjes give an overview.

In 1909 the Danish mathematician Agner Krarup Erlang published the paper ‘The theory of probabilities and telephone conversations’ [16]. In this paper, which is commonly viewed as the birth of queueing theory, Erlang studied dimensioning issues for traditional circuit-switched telephone systems. More specifically, a procedure was developed to determine the number of telephone lines which are needed between two villages so that the probability that, at some random time epoch, all lines are simultaneously busy is less than some specified small number.

The essential feature of Erlang’s model, and of queueing theory in general, is that there are customers who are competing for access to a scarce resource. In his model there was no waiting — if all lines are busy, a newly incoming call is ‘lost’. One comes across many situations in which models of this type apply, for instance in the context of wireless communication [6] and computer science [26]. One can, however, also think of variants in which customers who cannot be accommodated directly are sent to a ‘waiting room’, thus forming a genuine queue. Examples abound; one could think of the checkout counter of a supermarket, an elevator, a traffic light intersection, a machine that produces parts, a computer processor processing jobs, or a communication channel with a buffer for packets which still need to be transmitted. In some situations customers are initially willing to wait, but might become impatient at some

Figure 1 Queueing for on-the-day tickets at Wimbledon.
point — think for instance of the customers of a call center.

It is far from an easy task to model such a wide range of situations in which customers compete for access to a scarce resource. One has to model the service facility (the number of servers, their service speeds, the assignment of priorities, the size of waiting room, et cetera) as well as the customer behavior (the arrival process of customers at the service facility, the service requirements of the individual customers, the amount of patience they have, the choice which server to join, et cetera). Still, in the century following Erlang’s pioneering work, queuing theory has been remarkably successful in capturing the essential features of the congestion phenomena of a staggeringly wide range of extremely complicated real-life systems with relatively simple models — models which have shown to lend themselves to a detailed mathematical analysis. It has resulted in a set of techniques with which accurate predictions can be made of the global behavior of intricate stochastic systems, and which facilitate their optimization and control.

To a considerable extent, the success of queuing theory is due to the fact that one can distinguish a few basic building blocks, which have been studied in much detail and which time and again pop up in the analysis of new congestion phenomena. For instance, with the advent of wireless communications, sensor networks, and peer-to-peer networks, queuing models could be used to describe their performance. The building blocks most frequently used are the Erlang loss system (the system studied by Erlang in 1909; a system without queueing, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the construction and control of the building blocks we identified above, the arrival process and the service requirements.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.

The single server queue

Customers arrive at a service facility, where they would like to receive a certain amount of service. They are served customers in order of arrival (that is, First-Come-First-Served, usually abbreviated to FCFS). If a customer can not immediately serve customers, calls being lost when all lines are busy) and the single server queue. We shall describe the latter system in some detail, as it also plays a crucial role in the product-form networks that we shall discuss in the next section.
indicates Memoryless interarrival times, Generally distributed service requirements, and \( c \in \mathbb{N} \) servers.

A very attractive feature of the M/M/1 queue is that the stochastic process of numbers of customers \( X(t), t \geq 0 \), with \( X(0) \) the number of customers present (waiting plus in service) at time \( t \), is a Markov process. The powerful machinery of Markov processes now can be used, and readily yields elegant, explicit results. Suppose we assume that \( \lambda < \mu \), implying that the amount of work arriving per unit of time is smaller than the amount that can be served, thus guaranteeing that our queueing system is stable. Then it turns out that the steady-state distribution \( \lim_{t \to \infty} P(X(t) = n) \) exists, and is given by the geometric distribution:

\[
P(X = n) = (1 - \rho) \rho^n, \quad n = 0, 1, \ldots
\]

with \( \rho := \lambda/\mu < 1 \) representing the offered traffic load per unit time.

So far we have focused on single queueing facilities in isolation. In many practical contexts, however, the underlying stochastic systems can be seen as networks of multiple interrelated nodes. In the next section we consider these.

**Networks of queues**

**Open networks of queues**

Around 1950, the mathematical theory of stochastic processes had reached a certain maturity. Several monographs were published, including the landmark book of Feller [19], which not only gave a systematic discussion of a number of important stochastic processes, but also showed in a lucid way how to model many biological and physical phenomena by various stochastic processes like Markov chains and birth-and-death processes. The year 1954 saw the publication of the first study on networks of queues. R.R.P. Jackson [23] considered an M/M/1 queue \( Q_1 \), with arrival rate \( \lambda \) and service rate \( \mu_1 \), and assumed that each served customer immediately enters a second single server facility \( Q_2 \), again with infinite waiting room capacity and FCFS service, and again with independent, exponentially distributed service requirements; \( \mu_2 \) denotes the service rate in the downstream queue. He observed that the two-dimensional process of numbers of customers at \( Q_1 \) and \( Q_2 \), \( \{X_1(t), X_2(t), t \geq 0\} \), again is a Markov process — now a two-dimensional one. If \( \lambda < \mu_1 \) and \( \lambda < \mu_2 \), then this Markov process has a steady-state (limiting) distribution, and that distribution is unique. Jackson guessed that the steady-state distribution is given by

\[
\pi(n_1, n_2) = \pi_1^n \pi_2^{n_2}, \quad n_1, n_2 = 0, 1, \ldots
\]

with \( \pi_i := \lambda_i/\mu_i, \quad i = 1, 2 \). Then he set up the balance equations for this two-dimensional Markov chain: for \( n_1, n_2 = 1, 2, \ldots \),

\[
(\lambda + \mu_1 + \mu_2) \pi(n_1, n_2) = \lambda \pi(n_1 - 1, n_2) + \mu_1 \pi(n_1 + 1, n_2 - 1) + \mu_2 \pi(n_1, n_2 + 1),
\]

\[
(\lambda + \mu_1) \pi(n_1, 0) = \lambda \pi(n_1 - 1, 0) + \mu_1 \pi(n_1, 1),
\]

\[
(\lambda + \mu_2) \pi(0, n_2) = \mu_2 \pi(1, n_2 - 1) + \mu_1 \pi(0, n_2 + 1), \quad \pi(0, 0) = \mu_2 \pi(0, 1).
\]

Probably much to his surprise, Jackson observed that (4) indeed satisfies all the balance equations, and he had actually found the unique steady-state distribution!

Jackson’s results had a wide set of implications, of which we now mention a few.

i. The steady-state numbers of customers in \( Q_1 \) and \( Q_2 \) are independent, since the joint distribution is the product of the marginal distributions:

\[
P(X_1 = n_1, X_2 = n_2) = \pi_1^n \pi_2^{n_2}, \quad n_1, n_2 = 0, 1, \ldots
\]

ii. \( Q_2 \) actually behaves like an M/M/1 queue with Poisson(\( \lambda \)) arrival process (as follows by summing the expression in (4) over all \( n_1 = 0, 1, \ldots \)).

For obvious reasons, formula (4) has become known as a product-form result. Triggered by the above implications, Jackson’s results immediately gave rise to a frantick research effort. In 1956 Paul Burke [9], working at Bell Labs, proved what has later become known as the Output Theorem. This states that (i) the departure process of an M/M/c queue is again a Poisson process with (if the arrival rate \( \lambda \) is less than \( c \) times the service rate \( \mu \) the same rate as the arrival process, and (ii) the number of customers in an M/M/c queue at some arbitrary time \( t_0 \) is independent of the departure process before \( t_0 \).

Statement (i) immediately shows that \( Q_2 \) in Jackson’s two-queue model has a Poisson arrival process, and hence behaves like an M/M/1 queue. Statement (ii) readily implies that the steady-state numbers of customers in \( Q_1 \) and \( Q_2 \) are independent.

A year later Edgar Reich [32] gave a very simple proof of the output theorem, exploiting the observation that the queue length process in an M/M/c queue is reversible. Intuitively speaking, a reversible process is a stochastic process with the following property: if one would take a film of such a process and run the film backwards, then the resulting process is, statistically speaking, indistinguishable from the original process. Statement (i) of the output theorem immediately follows because, in the time-reversed process, the departure process becomes the arrival process — and hence is a Poisson process. Statement (ii) of the output theorem becomes after time reversal: the number of customers at some arbitrary time \( t_0 \) is independent of the arrival process after \( t_0 \). The memoryless property of that (Poisson) arrival process immediately implies that the latter statement is true.

J.R. Jackson [24], inspired by the results of R.R.P. Jackson, Burke and Reich, considered the following network of \( N \) single server queues \( Q_1, \ldots, Q_N \). New customers arrive at the queues according to independent Poisson processes, with rate \( \lambda_i \) at \( Q_i \). Service requirements at \( Q_i \) are independent, exponentially distributed with rate \( \mu_i, \quad i = 1, \ldots, N \). All servers operate under FCFS, and all waiting rooms have infinite capacity. If a customer has been served at \( Q_i \), then it is routed to \( Q_j \) with probability \( p_{ij} \) and leaves the network with probability \( p_{10}, \quad i, j = 1, \ldots, N \); obviously, one assumes that \( \sum_j p_{ij} = 1 \). All external interarrival times and service times are assumed to be independent.

Because of all the exponential, memoryless, assumptions, the process

\[
\{X_1(t), X_2(t), \ldots, X_N(t), t \geq 0\}
\]

of numbers of customers at \( Q_1, \ldots, Q_N \) is a Markov process. Jackson [24] verified that the balance equations for its steady-state distribution are satisfied by

\[
P(X_1 = n_1, \ldots, X_N = n_N) = \prod_{i=1}^N (1 - \rho_i) \rho_i^{n_i}, \quad n_1, \ldots, n_N = 0, 1, \ldots
\]
with, for $i = 1, \ldots, N$: $\rho_i := \lambda_i / \mu_i$, the offered load at $Q_i$, and where the so-called throughputs $\Lambda_i$ are the solution of the set of equations

$$\Lambda_i = \lambda_i + \sum_{j=1}^{N} \Lambda_j p_{ji}, \quad i = 1, \ldots, N$$

(in vector-matrix notation: $\Lambda = \lambda + \Lambda P$), which turns out to be unique as a direct consequence of the Perron-Frobenius theorem. The interpretation of $\Lambda_i$ is that it equals the external arrival rate $\lambda_i$ plus the sum of all the internal flows going into $Q_i$. It can be shown that the steady-state queue length distribution exists if and only if $\rho_i < 1$ for all $i = 1, \ldots, N$.

We observe that the steady-state distribution (6) exhibits a product form, which again implies that in steady state the numbers of customers at the various queues are independent, and again each queue behaves like an $M/M/1$ queue in isolation. Actually, Jackson [24] believed that the $M/M/1$ behavior of each $Q_i$ is not surprising, and can be seen as a direct consequence of the output theorem. He argued that if one merges two independent Poisson arrival processes, one obtains another Poisson process; and if one splits a Poisson process with fixed probabilities, a fraction $p_{ij}$ entering $Q_i$, then the resulting processes are independent Poisson processes. However, he overlooked the fact that his routing probabilities allow the possibility of feedback: a customer may visit a queue where it has been before. It is easily shown that the resulting dependency destroys the Poisson property of the flows. That makes the product-form result (6) all the more remarkable: the marginal queue length distribution at $Q_i$ is geometric, as if $Q_i$ were an $M/M/1$ queue (cf. (3)), but its arrival process does not have to be Poisson! Thanks to the work of Kelly [29] and others, much insight has been obtained into the phenomenon that each queue in the above-described Jackson network behaves as if it is an $M/M/1$ queue. The concept of quasi-reversibility plays a crucial role here: a service facility is quasi-reversible if it has the property that the departure process would be a Poisson process if the arrival process were a Poisson process.

**Closed networks of queues**

In 1963, J.R. Jackson [25] extended the results of his paper [24] to closed networks of $\cdot / M / 1$ (actually, $\cdot / M / 1$) queues. The only changes with respect to his above-described open network were that all external arrival rates $\lambda_i = 0$ and that all $p_{i0} = 0$, and that the system starts with $K$ customers. Since no customers can enter or leave, those $K$ customers stay in the network forever. At first sight this may seem not only cruel but also artificial, but actually it may well represent, e.g., having a fixed number $K$ of pallets in a factory, or having window flow control with window size $K$ in a communication network (i.e., at most $K$ packets may be transmitted without yet having received acknowledgment of receipt). Jackson [25] proved that the steady-state distribution of the numbers of customers at the various queues is once more given by a product form: for $n_1, \ldots, n_N = 0, 1, \ldots$ such that $n_1 + \cdots + n_N = K$,

$$P(X_1 = n_1, \ldots, X_N = n_N) = \frac{1}{G(N, K)} \prod_{i=1}^{N} \rho_i p_i$$

with $\rho_i := \lambda_i / \mu_i$ and $\Lambda_i = \sum_{j=1}^{N} \lambda_j p_{ji}$. The quantity $G(N, K)$ is a normalizing constant, obtained by summing the numerator of (7) over all possible combinations of $(n_1, \ldots, n_N)$, and realizing that the sum over all probabilities should equal 1. Notice that the $\Lambda_i$ are now determined up to a multiplicative constant (i.e., if $\Lambda_i$ is a solution, then so is $a \Lambda_i$ for any scalar $a$). The probability $P(X_1 = n_1, \ldots, X_N = n_N)$, however, still is uniquely determined. Indeed, multiplying all $\Lambda_i$ by $a$ amounts to multiplying both the numerator and denominator of (7) by $a^K$. It should also be observed that the product form now does not imply independence; in fact, the numbers of customers have an obvious dependence due to $X_1 + \cdots + X_N = K$.

**Generalizations**

Spurred by the elegance of the above product-form results, but also by the rapidly increasing need to study the performance of advanced computer and communication networks, a stream of papers was produced in the seventies and eighties, in which the product-form results of [23–25] were generalized. Some of the key publications are [5], [12] and [29]; several Dutch researchers have made important contributions to the field, including Boucherie, Cohen, van Dijk (who also published a monograph [15] on the topic) and Hordijk.

Thanks to all these efforts we now know that the steady-state joint queue length distribution in a small but significant class of queueing networks (open, closed, and mixed) has a product form. To mention some extensions: (i) Service facilities may have multiple servers; put differently, the service rate at a service facility may depend on the number of customers present. (ii) The service discipline at some nodes may be Last-Come-First-Served Preemptive-Resume, or Processor Sharing, instead of FCFS; here Processor Sharing is particularly relevant in a broad range of computer-communication applications. (iii) A network may have multiple classes of customers, with different routing probabilities for different classes (but not different service rates at FCFS nodes). For more details and information on the topic of queueing networks the interested reader is referred to [31, 34].

While these product-form results are of huge importance, as they allow a relatively simple performance analysis and optimization of a model that may reasonably accurately describe the behavior of a complex real-life system, they are also quite limited in the following sense. If one of the conditions for having a product-form network is violated, then most likely an exact analysis is extremely complicated, or — more often than not — completely out of reach. However, there is a class of, mainly, two-dimensional models — for example, two queues in series, or two queues and one arrival stream, customers joining the shorter queue — for which an exact analysis is possible. This is the topic we turn to in the next section. But beforehand, we first describe two interesting special systems.

**Remark 1.** A special case of a closed product-form network is a two-queue model with $K$ customers, $Q_1$ being an infinite server system (or, equivalently, a $K$-server system, as that would be sufficient to prevent any waiting; mean service time is $1 / \mu_1$ and $Q_2$ being a FCFS single server with exponentially($\mu_2$) distributed service times. In addition we assume that $p_{21} = p_{12} = 1$, meaning that the customers hop between both queues.

This model has become known as the computer-terminal model: $K$ active terminal users alternate between a ‘think mode’ in which they generate a job for the central processor, and a ‘wait mode’ in which they stay until the processor has handled the job. It also has become known as the machine-repair model: $K$ machines all alternate between an operational mode and a mode in which they are broken and stay in the repair shop $Q_2$, to be repaired by a single repairman. As observed in, e.g., [5], one has a product form even if the service times in $Q_2$ are generally distributed. Since $n_1 + n_2 = K$, the product-form results of [23–25] imply the results of (7)
uct form degenerates into a one-dimensional result: with $\nu := \mu_2/\mu_1$,

$$
\mathbb{P}(X_1 = n_1) = \frac{\nu^{n_1}}{n_1!} \prod_{i=1}^{K} \left( \frac{\nu^i}{i!} \right)^{\delta_{ii}},
$$

(8)

Interestingly, the same distribution holds for the so-called Erlang loss system, viz., calls arrive according to a Poisson($\mu_2$) process at a system of $K$ telephone lines, and the lengths of calls are generally distributed with mean $1/\mu_1$. In fact, it is not hard to see that the machine-repair model indeed is probabilistically equivalent with the Erlang loss system — a system that we introduced above as one of the basic building blocks of queueing.

**Remark 2.** In this second special case we consider the class of loss networks. In this model there are $N$ types of customers; customers of class $i$ arrive according to a Poisson process of rate $\lambda_i$ and remain in the system during a random time with mean $1/\mu_i$. The $N$ customer types use $R$ resources: a type $i$ customer uses an amount $A_{ir}$ at resource $r$. There is a total amount $C_r$ of resources of type $r$, entailing that a customer of type $i$ is blocked (and therefore lost) if upon arrival the remaining amount of resources available is less than the required $A_{ir}$. The resulting model is usually referred to as a loss network. Clearly, the numbers of customers in this system only attain values in the polyhedron

$$
H = \left\{ (n_1, \ldots, n_N) : \sum_{i=1}^{N} A_{ir} n_i \leq C_r \right\}.
$$

The steady-state distribution of the numbers of customers is again of product form: due to the detailed analysis in e.g. Kelly [30], with $\nu_i := \lambda_i/\mu_i$,

$$
\mathbb{P}(X_1 = n_1, \ldots, X_N = n_N) = \frac{1}{G} \prod_{i=1}^{N} \frac{\nu_i^{n_i}}{n_i!};
$$

where the normalizing constant $G = G(N, C_1, \ldots, C_K)$ is given by

$$
\sum_{(n_1, \ldots, n_N) \in H} \prod_{i=1}^{N} \frac{\nu_i^{n_i}}{n_i!},
$$

which can be efficiently computed using Buzen’s algorithm [30]. Because of the high relevance of this type of models, a substantial research effort was spent on developing computational techniques for loss networks, culminating in the elegant recursive techniques published essentially simultaneously by Kaufman [27] and Roberts [33].

The loss network attracted substantial attention in the 1990s, where it was used in the context of multi-service communication networks. Till then networks were service-specific: there was a telephone network, a separate network for data traffic, et cetera. From about 1990 on, however, networks were increasingly organized in such a way that they could support multiple services over a common infrastructure — as we know it from the current internet. For example, a voice call typically requires less of the network’s capacity (perhaps a few tens of kilobits, or even less) than a video connection (a few hundreds of kilobits), which can be nicely incorporated in the loss network model described above.

In a way the loss model can be considered as a very advanced version of the Erlang loss model that we introduced at the very beginning of this paper.

**Stability**

It was mentioned earlier that the steady-state distribution (6) exists if and only if the offered load of each station in the network is strictly less than one, i.e.

$$
\rho_i := \frac{\lambda_i}{\mu_i} < 1, \quad i = 1, 2, \ldots, N.
$$

Such conditions are, in the queueing context, referred to as stability conditions and can be viewed, informally speaking, as indications of whether a network has enough resources to handle incoming work. The stability analysis of queueing networks was perhaps thought to be a moot subject, in the sense that, based on the pioneering work of Jackson [24] and Kelly [28], it initially seemed that stability depends only on the offered load of each station in the network. Essentially, this simplistic analysis would imply that the stability of the network can be derived by looking individually at each station in the network.

However, a series of counterexamples demonstrated that the station traffic intensities may not be sufficient to determine the stability of the network. In [7], Bramson gave an example of a two-station network that is unstable, even though the offered load of each station in the network is strictly less than one. In particular, Bramson assumed a network consisting of two stations in tandem, to which customers arrive to station 1 according to a Poisson arrival process at rate 1 and follow a prescribed route $1 \rightarrow 2 \rightarrow 2 \rightarrow \cdots \rightarrow 2 \rightarrow 1$ at which point they exit the network. In total, any arriving customer to the network will visit station 1 twice and station 2 $J$ times according to the prescribed route. Furthermore, Bramson assumed that the service rate at each station depends on the number of times the customer has already visited the station, say $\mu_{i,j}$, where $i$ denotes the station (taking value 1 for station 1 and value 2 for station 2) and $j$ denotes the number of times this station has been visited up to then (taking values 1 and 2, if $i = 1$, and values 1, 2, $\ldots$, $J$, if $i = 2$). For instance, if one chooses

$$
\mu_{1,2} = \mu_{2,1} = \frac{400}{399}, \quad \mu_{1,1} = \mu_{2,2} = 10^{11},
$$

(9)

$j = 2, 3, \ldots, J$ and $J = 1600$,

then

$$
\rho_1 = \frac{1}{\mu_{1,1}} + \frac{1}{\mu_{1,2}} < 1 \quad \text{and} \quad \rho_2 = \frac{J}{\sum_{j=1}^{J} \frac{1}{\mu_{2,j}}} < 1.
$$

Bramson showed, see [7, Theorem 1], that for $\mu_{i,j}$ chosen according to (9), this system is unstable with the number of customers in the system growing unboundedly as $t \rightarrow \infty$.

This result, while looking counterintuitive at first sight, can be explained as follows, see [8, Section 3.2]. Assume that at time $t = 0$ there are $M$ customers (with $M$ a very large number) in station (1,1) and a few more in the rest of the system. Moreover, let $S_1$ denote the time at which the last of the original jobs i.e., the jobs present at time $t = 0$, at station 1 is served. Let $S_2, S_3, \ldots$ denote the successive times at which the last jobs at station 2 are served. Since $\mu_{1,1} \gg 1$, one has that $S_1 < M$ except on a set of small probability. Also, $\mu_{2,1} = 1/c \gg 1$, and so at time $S_1$ nearly all of the original jobs in the network are still at (2,1). Next, over this time interval $(S_1, S_2)$, the (approximately) $M$ jobs at (2,1) all move to (2,2). Since $\mu_{2,2} = 1/c$, the time it takes to serve these jobs is (approximately) $cM$. The time required to serve other jobs is minimal, so $S_2 - S_1 \approx cM$. During this time, (approximately) $cM$ new jobs enter the system, which quickly move to (2,1). Thus, at $t = S_2$, there are (comparatively) few jobs in the system except at (2,2) and (2,1), where there are (approximately) $M$ and $cM$ jobs, respectively.

Continuing our reasoning along the same lines, we observe that over $(S_2, S_3)$, the jobs at (2,1) and (2,2) advance to (2,2) and (2,3), respectively. Since $\mu_{2,2} \gg 1$, the time required to serve the jobs at (2,2) is negligible; the time required for the jobs at (2,1) is $c^2M$, so $S_3 - S_2 \approx c^2M$. Over this time,
new jobs enter the system, which quickly move to \((2, 1)\). So, at time \(S_J\), there are few jobs in the system except at \((2, 3), (2, 2),\) and \((2, 1)\), where there are \(M\), \(cM\) and \(c^2M\) jobs, respectively. Proceeding inductively, we obtain that at time \(S_J\), there are \(M\) jobs at \((2, J)\), \(cM\) jobs at \((2, J - 1)\), and so on down to \((2, 1)\), where there are \(c^{J-1}M\) jobs. At station 1, there are few jobs. The elapsed time is \(S_J - S_{J-1} \approx c^{J-1}M\). Note that \(c^J\) was chosen to be very small, and so there are about

\[
\sum_{k=0}^{J-1} c^k M \approx \frac{M}{1 - c}
\]

(jobs in the system. Likewise, \(S_J \approx cM/(1 - c)\). Over the short period of time \((S_J, S_{J-1})\), the evolution of the system changes. The \(M\) jobs from \((2, J)\) arrive at \((1, 2)\). Since \(\mu_{2,1} = 1/c\), these jobs require time \(cM\) to be served at station 1, during which time new arrivals at \((1, 1)\) will not be served. By time \(S_{2J}\), the jobs that were at station 2 at time \(S_J\) have already arrived at \((1, 2)\); because of (10), there are essentially \(M/(1 - c)\) such jobs. So, at time \(S_{2J}\), there are essentially \(M/(1 - c)\) jobs at \((1, 2)\) and no jobs elsewhere. Of course, here and elsewhere, we are taking liberties in ignoring ‘negligible’ quantities of jobs and probabilities. Let now \(T\) denote the time that these last jobs will exit the system. The time required to serve these jobs is \(cM/(1 - c)\). So, \(T - S_{2J} \approx cM/(1 - c)\). During this time, \(cM/(1 - c)\) jobs enter the system. These new jobs are obliged to remain at \((1, 1)\) until time \(T = S_{2J} + (T - S_{2J}) \approx 2cM/(1 - c)\). At this time, there are few jobs elsewhere in the system. So at time \(T\), the state of the system is a ‘multiple’, by the factor \(c/(1 - c)\), of the state at time \(0\).

Of course, since we are working with random events here, the above behavior is sometimes violated. However, such exceptional events occur with probabilities that are exponentially small in \(M\), and one can show they can be ignored without affecting the basic nature of the evolution of number of customers in the system. Needless to say, a rigorous proof requires accurate bookkeeping of such exceptional probabilities, but we were only interested in presenting here an intuitive argument with which the interested reader can grasp why this system is unstable.

Such counter examples inspired further investigations into the stability regions of queueing models under various scheduling policies and also spurred work on the development of a theory for the determination of the stability region for a wide range of queueing networks, see e.g. [21].

**Routing policies**

In the contexts previously described, we assume that customers are routed to the various stations of the network independently of the number of customers already waiting in these stations. However, in practice when a rational customer makes a decision on which station to join, then typically this decision is influenced by the number of customers waiting in queue in each one of the stations. Think for example of the structure of a supermarket: there are multiple cashiers each with their own waiting line, these constitute the various stations in our ‘supermarket’ network. In the context of supermarkets customers typically join the station with the smallest number of waiting customers. The steady-state distribution of this type of network has received the attention of various researchers and some of the area’s important contributions were achieved by several Dutch researchers, including Adan and Cohen (who also published two monographs [13–14] on the related topic of two-dimensional random walks). We will further elaborate on the topic of the steady-state analysis of the join the shortest queue (JSQ) policy in the case of two stations in the next section.

**Mathematical analysis of 2D models**

We have seen that single server queues and specific classes of multi-dimensional queueing systems, such as Jackson networks, can be analyzed in great detail. When slightly changing the mechanics, however, the analysis may become substantially harder. For example, in the case of two stations in parallel where customers are routed according to the JSQ policy, the steady-state distribution does not obey a product-form solution. The steady-state solution can still be found, as we demonstrate in this section.

**Model description, steady-state distribution**

In the basic version of the model customers arrive to the system according to a Poisson process at rate \(\lambda\). There are two queues; a new arrival is routed to the shorter one (in the case of a tie, the queue is selected at random). The service times at each of the queues are exponential with mean \(1/\mu\). It was argued that this system is stable if \(\rho := \lambda/2\mu < 1\). This model was first introduced by Haight [22], and was analyzed by Flatto and McKean [20] and Kingman [31]. We now describe the approach followed by the latter, identifying the probability generating function of the numbers of customers in steady-state.

First, with \(X_i\) denoting the number of customers in station \(i\) in stationarity, we define

\[
\pi(n_1, n_2) = P(X_1 = n_1, X_2 = n_2),
\]

\(n_1, n_2 = 0, 1, 2, \ldots\)

By symmetry, \(\pi(n_1, n_2) = \pi(n_2, n_1)\). Then, write the balance equations of the system: for \(n_1, n_2 = 0, 1, \ldots\) such that \(n_1 \leq n_2\),

\[
(2\rho + 1_{n_1 = 0} + 1_{n_2 = 0})\pi(n_1, n_2) = (2\rho 1_{n_2 = n_1} + \rho 1_{n_2 = n_1+1}) \pi(n_1, n_2 - 1) + 2\rho\pi(n_1 - 1, n_2) + \pi(n_1 + 1, n_2) + \pi(n_1, n_2 + 1),
\]

where \(1_{(A)}\) is the delta Kronecker taking value 1 when event \(A\) occurs and 0 otherwise. Let
be the bivariate probability generating function of the minimum queue length (represented by the variable $n_1$) and of the difference of the two queues (represented by the variable $n_2 - n_1$). Then, multiplying equation (11) with $x^{n_1}y^{n_2-n_1}$ and summing for all $n_1 \leq n_2$ yields a functional equation for the probability generating function:

$$P(x, y) := \sum_{n_1=0}^{\infty} \sum_{n_2=n_1}^{\infty} \pi(n_1, n_2)x^{n_1}y^{n_2-n_1},$$

$$|x|, |y| < 1,$$

be the bivariate probability generating function of the minimum queue length (represented by the variable $n_1$) and of the difference of the two queues (represented by the variable $n_2 - n_1$). Then, multiplying equation (11) with $x^{n_1}y^{n_2-n_1}$ and summing for all $n_1 \leq n_2$ yields a functional equation for the probability generating function:

$$P(x, y) := \sum_{n_1=0}^{\infty} \sum_{n_2=n_1}^{\infty} \pi(n_1, n_2)x^{n_1}y^{n_2-n_1},$$

$$|x|, |y| < 1,$$

be the bivariate probability generating function of the minimum queue length (represented by the variable $n_1$) and of the difference of the two queues (represented by the variable $n_2 - n_1$). Then, multiplying equation (11) with $x^{n_1}y^{n_2-n_1}$ and summing for all $n_1 \leq n_2$ yields a functional equation for the probability generating function:

$$P(x, y) := \sum_{n_1=0}^{\infty} \sum_{n_2=n_1}^{\infty} \pi(n_1, n_2)x^{n_1}y^{n_2-n_1},$$

$$|x|, |y| < 1,$$

be the bivariate probability generating function of the minimum queue length (represented by the variable $n_1$) and of the difference of the two queues (represented by the variable $n_2 - n_1$). Then, multiplying equation (11) with $x^{n_1}y^{n_2-n_1}$ and summing for all $n_1 \leq n_2$ yields a functional equation for the probability generating function:

$$P(x, y) := \sum_{n_1=0}^{\infty} \sum_{n_2=n_1}^{\infty} \pi(n_1, n_2)x^{n_1}y^{n_2-n_1},$$

$$|x|, |y| < 1,$$
References

23. R.R. Jackson, Queueing systems with phase-type service, Operational Research Quarterly 5 (1954), 109–120.