Performance Modelling — Lecture 5
Queueing Networks

Jane Hillston
School of Informatics
The University of Edinburgh
Scotland

30th January 2017

Single Queues

- The basic scenario for a single queue is that customers, who belong to some population arrive at the service facility.

Single Queues

- The basic scenario for a single queue is that customers, who belong to some population arrive at the service facility.
- The service facility has one or more servers who are capable of performing the service required by customers.

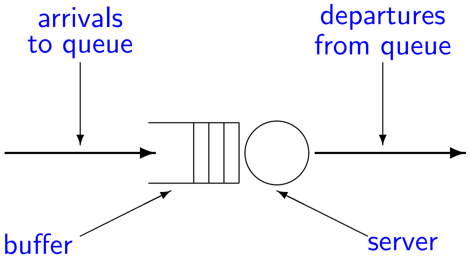
Single Queues

- The basic scenario for a single queue is that customers, who belong to some population arrive at the service facility.
- The service facility has one or more servers who are capable of performing the service required by customers.
- If a customer cannot gain access to a server it must join a queue, in a buffer, until a server is available.

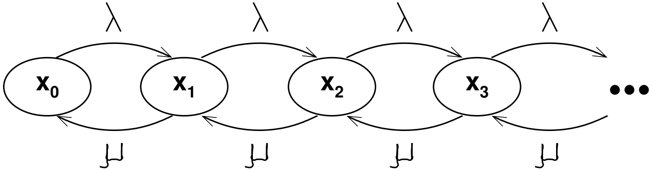
Single Queues

- The basic scenario for a single queue is that customers, who belong to some population arrive at the service facility.
- The service facility has one or more servers who are capable of performing the service required by customers.
- If a customer cannot gain access to a server it must join a queue, in a buffer, until a server is available.
- When service is complete the customer departs, and the server selects the next customer from the buffer according to the service discipline.

Simple Example



State space of a simple queue



Arrival Pattern of Customers

The ability of the service facility to provide service for an arriving stream of customers depends not only on the mean arrival rate λ, but also on the pattern in which they arrive, i.e. the distribution function of the inter-arrival times.

Arrival Pattern of Customers

The ability of the service facility to provide service for an arriving stream of customers depends not only on the mean arrival rate λ, but also on the pattern in which they arrive, i.e. the distribution function of the inter-arrival times.

We will primarily be considering queues in which the times between arrivals are assumed to be exponentially distributed.

Arrival Pattern of Customers

The ability of the service facility to provide service for an arriving stream of customers depends not only on the mean arrival rate λ, but also on the pattern in which they arrive, i.e. the distribution function of the inter-arrival times.

We will primarily be considering queues in which the times between arrivals are assumed to be exponentially distributed.

However you should be aware that many other possibilities exist and are studied, such as bulk or batch arrivals and train arrivals.

Service Time Distribution

The service time is the time which a server spends satisfying a customer.

Service Time Distribution

The service time is the time which a server spends satisfying a customer.

As with the inter-arrival time, the important characteristics of this time will be its average duration and the distribution function.

Service Time Distribution

The service time is the time which a server spends satisfying a customer.

As with the inter-arrival time, the important characteristics of this time will be its average duration and the distribution function.

We will mostly consider service times which obey an exponential distribution.

Service Time Distribution

The service time is the time which a server spends satisfying a customer.

As with the inter-arrival time, the important characteristics of this time will be its average duration and the distribution function.

We will mostly consider service times which obey an exponential distribution.

If the average duration of a service interaction between a server and a customer is $1/\mu$ then μ is the service rate.

Number of Servers

If there is more than one server in the service facility the processing of customers can go on concurrently: one customer at each server.

Number of Servers

If there is more than one server in the service facility the processing of customers can go on concurrently: one customer at each server.

It is assumed that the servers are indistinguishable from each other, so that it does not matter which server actually serves a customer—the results, and the service characteristics, will be the same.

Number of Servers

We can have

- a single server: the service facility only has the capability to serve one customer at a time; waiting customers will stay in the buffer until chosen for service; how the next customer is chosen will depend on the service discipline

Number of Servers

We can have

- a single server: the service facility only has the capability to serve one customer at a time; waiting customers will stay in the buffer until chosen for service; how the next customer is chosen will depend on the service discipline
- an infinite server: there are always at least as many servers as there are customers, so that each customer can have a dedicated server as soon as it arrives in the facility. There is no queueing, (and no buffer) in such facilities.

Multiple Servers

Between these two extremes there are multiple server facilities.

Multiple Servers

Between these two extremes there are multiple server facilities.

These have a fixed number of c servers, each of which can service a customer at any time.

Multiple Servers

Between these two extremes there are multiple server facilities.

These have a fixed number of c servers, each of which can service a customer at any time.

If the number of customers in the facility is less than or equal to c there will no queueing—each customer will have direct access to a server.

Multiple Servers

Between these two extremes there are multiple server facilities.

These have a fixed number of c servers, each of which can service a customer at any time.

If the number of customers in the facility is less than or equal to c there will no queueing—each customer will have direct access to a server.

If there are more than c customers, the additional customers will have to wait in the buffer.

Buffer Capacity

Customers who cannot receive service immediately must wait in the buffer until a server becomes available.

Buffer Capacity

Customers who cannot receive service immediately must wait in the buffer until a server becomes available.

If the buffer has finite capacity there are two alternatives for when the buffer becomes full:

Buffer Capacity

Customers who cannot receive service immediately must wait in the buffer until a server becomes available.

If the buffer has finite capacity there are two alternatives for when the buffer becomes full:

- either, the fact that the facility is full is passed back to the arrival process and arrivals are suspended until the facility has spare capacity, i.e. a customer leaves;

Buffer Capacity

Customers who cannot receive service immediately must wait in the buffer until a server becomes available.

If the buffer has finite capacity there are two alternatives for when the buffer becomes full:

- either, the fact that the facility is full is passed back to the arrival process and arrivals are suspended until the facility has spare capacity, i.e. a customer leaves;
- or, arrivals continue and arriving customers are lost (turned away) until the facility has spare capacity again.

Buffer Capacity

Customers who cannot receive service immediately must wait in the buffer until a server becomes available.

If the buffer has finite capacity there are two alternatives for when the buffer becomes full:

- either, the fact that the facility is full is passed back to the arrival process and arrivals are suspended until the facility has spare capacity, i.e. a customer leaves;
- or, arrivals continue and arriving customers are lost (turned away) until the facility has spare capacity again.

If the buffer capacity is so large that it never affects the behaviour of the customers it is assumed to be infinite.

Population: size

The characteristic of the population which we are interested in is usually the size.

Population: size

The characteristic of the population which we are interested in is usually the size.

Clearly, if the size of the population is fixed, at some value N, no more than N customers will ever be requiring service at any time.

Population: size

The characteristic of the population which we are interested in is usually the size.

Clearly, if the size of the population is fixed, at some value N, no more than N customers will ever be requiring service at any time.

When the population is finite the arrival rate of customers will be affected by the number who are already in the service facility (e.g. zero arrivals when all N are all already in the facility).

Population: size

The characteristic of the population which we are interested in is usually the size.

Clearly, if the size of the population is fixed, at some value N, no more than N customers will ever be requiring service at any time.

When the population is finite the arrival rate of customers will be affected by the number who are already in the service facility (e.g. zero arrivals when all N are all already in the facility).

When the size of the population is so large that there is no perceptible impact on the arrival process we assume that the population is infinite.

Population: classes

Often, members of the population are indistinguishable from each other.

Population: classes

Often, members of the population are indistinguishable from each other.

When this is not the case we divide the population into classes whose members all exhibit the same behaviour.

Population: classes

Often, members of the population are indistinguishable from each other.

When this is not the case we divide the population into classes whose members all exhibit the same behaviour.

Different classes differ in one or more characteristics, for example, arrival rate or service demand.

Population: classes

Often, members of the population are indistinguishable from each other.

When this is not the case we divide the population into classes whose members all exhibit the same behaviour.

Different classes differ in one or more characteristics, for example, arrival rate or service demand.

Identifying different classes is a workload characterisation task.

Service Discipline

When more than one customer is waiting for service there has to be a rule for selecting which of the waiting customers will be the next one to gain access to a server.

Service Discipline

When more than one customer is waiting for service there has to be a rule for selecting which of the waiting customers will be the next one to gain access to a server.

The commonly used service disciplines are

- FCFS first-come-first-serve (or FIFO *first-in-first-out*).
- LCFS last-come-first-serve (or LIFO *last-in-first-out*).
- RSS random-selection-for-service.
- PRI priority. The assignment of different priorities to elements of a population is one way in which classes are formed.

Service Discipline

When more than one customer is waiting for service there has to be a rule for selecting which of the waiting customers will be the next one to gain access to a server.

The commonly used service disciplines are

- FCFS first-come-first-serve (or FIFO *first-in-first-out*).
- LCFS last-come-first-serve (or LIFO *last-in-first-out*).
- RSS random-selection-for-service.
- PRI priority. The assignment of different priorities to elements of a population is one way in which classes are formed.

We will only consider solution of systems with the FCFS service discipline.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.
- S denotes the service rate and uses the distribution identifiers as above.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.
- S denotes the service rate and uses the distribution identifiers as above.
- c denotes the number of servers available in the service facility.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.
- S denotes the service rate and uses the distribution identifiers as above.
- c denotes the number of servers available in the service facility.
- m denotes the capacity of the **service facility** (buffer + server(s)), infinite by default.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.
- S denotes the service rate and uses the distribution identifiers as above.
- c denotes the number of servers available in the service facility.
- m denotes the capacity of the **service facility** (buffer + server(s)), infinite by default.
- N denotes the customer population, also infinite by default.

Kendall's notation

A shorthand notation for these six characteristics of a system is provided by Kendall's notation for classifying queues. In this notation a queue is represented as $A/S/c/m/N/SD$:

- A denotes the arrival process; usually M, to denote Markov (exponential), G, general, or D, deterministic distributions.
- S denotes the service rate and uses the distribution identifiers as above.
- c denotes the number of servers available in the service facility.
- m denotes the capacity of the **service facility** (buffer + server(s)), infinite by default.
- N denotes the customer population, also infinite by default.
- SD denotes the service discipline, FCFS by default.

Example

Consider a wireless access gateway:

- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered.

Example

Consider a wireless access gateway:

- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered.
- The gateway takes 2 milliseconds on average to transmit a packet.

Example

Consider a wireless access gateway:

- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered.
- The gateway takes 2 milliseconds on average to transmit a packet.
- The buffer currently has 13 places, including the place occupied by the packet being transmitted and packets that arrive when the buffer is full are lost.

Example

Consider a wireless access gateway:

- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered.
- The gateway takes 2 milliseconds on average to transmit a packet.
- The buffer currently has 13 places, including the place occupied by the packet being transmitted and packets that arrive when the buffer is full are lost.
- We wish to find out if the buffer capacity is sufficient to ensure that less than one packet per million gets lost.

Example

Consider a wireless access gateway:

- Measurements have shown that packets arrive at a mean rate of 125 packets per second, and are buffered.
- The gateway takes 2 milliseconds on average to transmit a packet.
- The buffer currently has 13 places, including the place occupied by the packet being transmitted and packets that arrive when the buffer is full are lost.
- We wish to find out if the buffer capacity is sufficient to ensure that less than one packet per million gets lost.

Making exponential assumptions about the arrival rate and the service rate we would model the gateway as an $M/M/1/13/\infty/FCFS$ queue with $\lambda = 0.125$ and $\mu = 0.5$ (/ms).

Traffic intensity

The two most important features of a single queue are the arrival rate of customers λ, and the service rate of the server(s), μ.

Traffic intensity

The two most important features of a single queue are the arrival rate of customers λ, and the service rate of the server(s), μ.

These are combined into a single parameter which characterises a single or multiple server system, the traffic intensity.

Traffic intensity

$$\rho = \frac{\lambda}{c \times \mu}$$

Traffic intensity and stability

For the system to be stable, ρ must be less than 1: that is the arrival rate of customers must be less than the rate at which they are served.

Stability

a queue is stable if, and only if, $\rho < 1$

Traffic intensity and stability

For the system to be stable, ρ must be less than 1: that is the arrival rate of customers must be less than the rate at which they are served.

Stability

a queue is stable if, and only if, $\rho < 1$

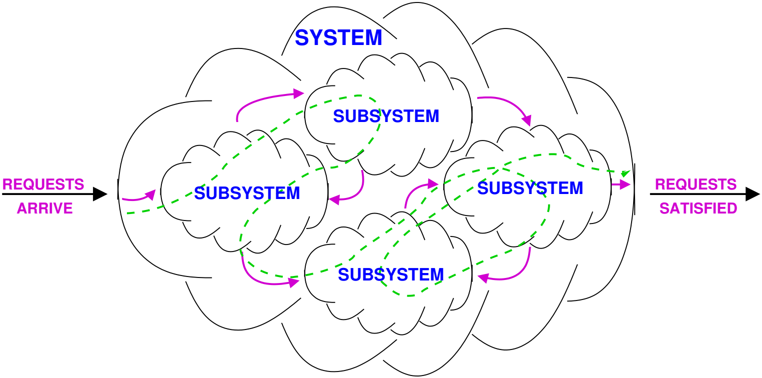
The alternative, if the arrival rate is greater than the service rate, would result in a queue which grew unboundedly.

Network of queues

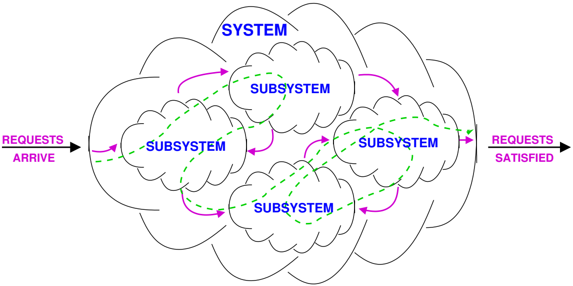
For many systems we can adopt a view of the system as a collection of resources and devices with customers or jobs circulating between them (cf. operational analysis).

Network of queues

For many systems we can adopt a view of the system as a collection of resources and devices with customers or jobs circulating between them (cf. operational analysis).

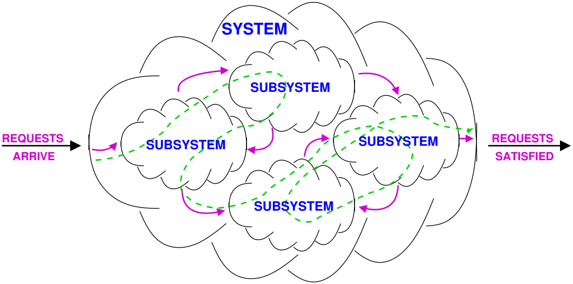


Network of queues



We can associate a service centre with each resource in the system and then route customers between the service centres.

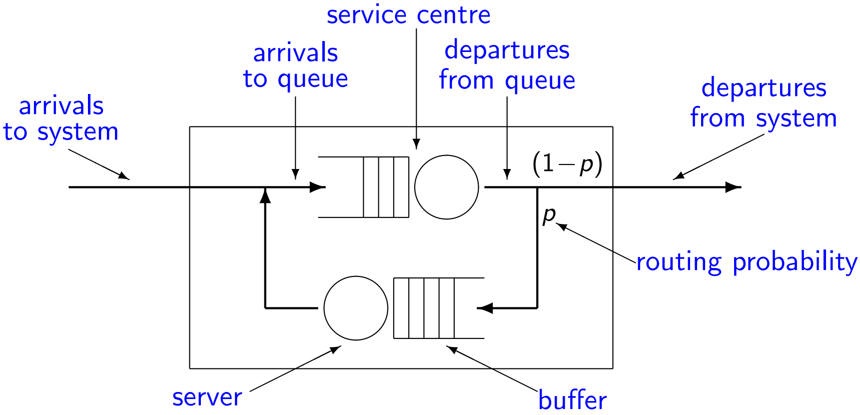
Network of queues



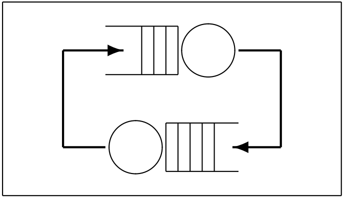
We can associate a service centre with each resource in the system and then route customers between the service centres.

After service at one service centre a customer may progress to other service centres, following some previously defined pattern of behaviour, corresponding to the customer's requirement.

Network of queues



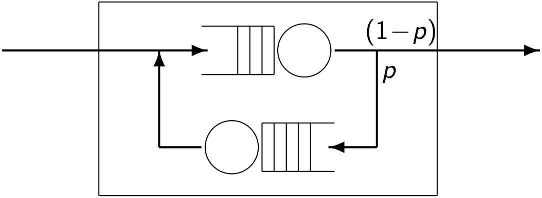
Network of queues



A network may be:

- closed, a fixed population of customers remain within the system;

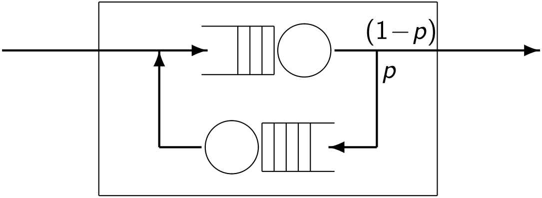
Network of queues



A network may be:

- closed, a fixed population of customers remain within the system;
- open, customers may arrive from, or depart to, some external environment; or

Network of queues



A network may be:

- closed, a fixed population of customers remain within the system;
- open, customers may arrive from, or depart to, some external environment; or
- mixed, there are classes of customers within the system exhibiting open and closed patterns of behaviour respectively.

State of a queueing network

The state of the system is typically represented as the number of customers currently occupying each of the service centres.

State of a queueing network

The state of the system is typically represented as the number of customers currently occupying each of the service centres.

If there a number of different classes of customers, the state is the number of customers of each class at each service centre.

State of a queueing network

The state of the system is typically represented as the number of customers currently occupying each of the service centres.

If there a number of different classes of customers, the state is the number of customers of each class at each service centre.

As well as differences in arrival rate and/or service demand, classes may be distinguished by the routes which customers take through the network.

Expressiveness

There are some features of contemporary computer and communication systems which are not easily represented by queueing networks.

Expressiveness

There are some features of contemporary computer and communication systems which are not easily represented by queueing networks.

Simultaneous resource possession In a computer system a job may continue to compute while its I/O requests are progressing in parallel. Thus, the job may be simultaneously using two or more resources of the system.

Expressiveness

There are some features of contemporary computer and communication systems which are not easily represented by queueing networks.

Simultaneous resource possession In a computer system a job may continue to compute while its I/O requests are progressing in parallel. Thus, the job may be simultaneously using two or more resources of the system.

Fork and Join primitives used in computer systems to create and synchronise subprocesses, cause the number of jobs in the system to change and invalidate the assumption of independence among jobs.

Expressiveness cont.

Bulk and train arrivals. In communication networks the arrival of packets may occur in batches (bulk arrivals) or in quick succession (train arrivals). This breaks assumptions of independence between customers, and exponential inter-arrival times.

Expressiveness cont.

Bulk and train arrivals. In communication networks the arrival of packets may occur in batches (bulk arrivals) or in quick succession (train arrivals). This breaks assumptions of independence between customers, and exponential inter-arrival times.

Load dependent arrivals. Computer networks and distributed systems have intelligent load-balancing policies that cause new jobs or packets to go to one of a number of devices depending upon the load observed in the recent past.

Expressiveness cont.

Bulk and train arrivals. In communication networks the arrival of packets may occur in batches (bulk arrivals) or in quick succession (train arrivals). This breaks assumptions of independence between customers, and exponential inter-arrival times.

Load dependent arrivals. Computer networks and distributed systems have intelligent load-balancing policies that cause new jobs or packets to go to one of a number of devices depending upon the load observed in the recent past.

Defections from the queue. Routers often set a maximum limit on the time that a packet or request is able to stay in a queue. Thus, packets may be dropped from the queue on the assumption that they may have already been retransmitted by higher protocol layers.