Queueing Theory

Peter Fenwick, July 2002

August 7, 2009

1 Preliminary note on mathematical models

Most of Computer Science has rather little contact with numbers, measurements and physical reality – it doesn't matter too much if things get a bit slower, or a bit faster.

Data Communications is not like that. It is full of physical quantities such as propagation velocities and delays, bit rates, message lengths, and so on. With real-world things like this we often set up mathematical pictures or "models" to describe and predict behaviour.

Now all three of the 742 lecturers (in 2001 at least) have a background in Physics, and Physics is largely about finding mathematical models or descriptions of some physical system or process. This means that we often work in terms of models, sometimes without even realising it, and this can lead to all sorts of confusions among Computer Science students. The queueing theory of these notes leads to quite typical mathematical models, with hidden or assumed implications.

Some of these aspects are –

1. A model is never any more than an approximation to reality; people can all to easily assume that their model is reality and then get into all sorts of problems. Sometimes a simple model exists only because we can't handle the mathematics of a more accurate one! This often applies to queueing theory.

2. When using a model you must know what simplifications and assumptions it makes. For example a very simple model of the flight of a ball or other projectile says that it is a parabola. More complex models progressively introduce air resistance, the spin of the ball, the rotation of the earth and even more, but at increasing complexity.

3. You must recognise the limitations of the model, when it works and when it doesn't. Knowing when it will not predict is at least important as knowing when it will predict.

4. You may find that somebody used to working with models will assume one for a while and then abruptly abandon it or move to another, when the original one is "clearly inappropriate". If you understand the model and its limitations as in the earlier points, the change may be natural. If you do not understand those aspects, it is totally confusing. Be warned – I have found this can be a very real problem!

5. Sometimes you can use a very simple, crude, model to see if something is feasible. For example if a communications protocol must transfer data at 1.5Mbit/s, and "quick and dirty"

calculation shows that it can never do better than 1.1Mbit/s then you probably need another approach. But if that calculation showed that it should work at 1.6 or even 1.4 Mbit/s then a more careful calculation is probably justified.

2 Queueing Theory – Introduction and Terms

Queueing Theory deals with the situation where customers(people, or other entities) wait in an ordered line or queue for service from one or more servers. Customers arrive on the queue according to some assumed distribution of interarrival times and, after waiting, take some service time to have the request satisfied. Within the environment of a computing system, queues apply to buffers in a communication system, the handling of I/O traffic (and especially disk traffic), to people awaiting access to a terminal, and failure rates and times to repair. In many cases there will be several cascaded queues,or several interacting queues. Several terms must be specified before we can discuss a general queueing system –

Source The population source may be finite or infinite. The essential point of a finite population is that the queue absorbs potential customers as it grows and the arrival rate falls in accordance with the population not in the queue. For a large population we often assume an infinite population to simplify the mathematics.

Arrival Process Assume that customers enter the queue at times $t_0 < t_1 < t_2 \dots t_n \dots$. The random variables $\tau_k = t_k - t_{k-1}$ (for $k \geq 1$) are the interarrival times, and are assumed to form a sequence of independent and identically distributed random variables. The arrival process is described by the distribution function A of the interarrival time $A(t) = P[\tau \leq t]$.

- If the interarrival time distribution is exponential (ie $P[t \leq \tau] = 1 - e - \lambda t$, where $\lambda = 1/\tau$), the probability of n arrivals in a time interval of length t is $e^{-\lambda t}(\lambda t)^n / n!$, for $n = 0, 1, 2, \dots$ and the average arrival rate is λ. This corresponds to the important case of a Poisson distribution where, in a very large population of n customers, the probability, P, of a particular customer entering the queue within a short time interval is very small, but there is a reasonable probability (nP) that some customer will arrive.
- Another important distribution is the Erlang-k distribution, defined by

$$E_k(x) = 1 - \sum_{k-1}^{j=0} \frac{(\lambda x)^j}{j!} e^{-\lambda x}$$

It applies to a cascade of servers with exponential distribution times, such that a customer cannot be started until the previous one has been completely processed.

Service Time Distribution Let s_k be the service time required by the kth arriving customer; assume that the s_k are independent, identically distributed random variables and that we can refer to an arbitrary service time as s, distributed as $W_s(t) = P[s \leq t]$. The most usual service-time distribution is *exponential*, defining a service called random service. If μ is the average service rate, then $W_s(t) = 1 - e^{-\mu t}$.

Maximum queueing system capacity In some systems the queue capacity is assumed to be infinite; all arriving customers can be accommodated, although the waiting time may be arbitrarily long. In others the queue capacity is zero (the customer is turned away if there is no free server). In other cases the queue may have a finite capacity, such as a waiting room with limited seating.

Number of servers The simplest queueing system is the single server system, which can serve only one customer at a time. A multiserver system has c identical servers and can serve up to c customers simultaneously.

Queue discipline The queue discipline, or service discipline, defines the rule for selecting the next customer. The most usual one is "first come first served" (FCFS), also known as "first in first out" or FIFO. Another one is "random selection for service" (RSS) or "service in random order" (SIRO). In some circumstances we deal with priority queues (essentially parallel queues where there is a preferred order of selecting customers for service), or with preemptive queues in which a new customer can interrupt a customer being served.

Traffic Intensity The traffic intensity ρ is the ratio of the mean service time $E[s]$ to the mean interarrival time $E[\tau]$, for an arrival rate λ and service rate μ; it defines the minimum number of servers to cope with the arriving traffic.

$$\rho = \frac{E[s]}{E[t]} = \lambda E[s] = \frac{\lambda}{\mu}$$

Server utilisation The traffic intensity per server or *server utilisation* $u = \rho/c$ is the approximate probability that a server is busy (assuming that traffic is evenly divided among the servers). Note that some authors interchange ρ and u so that ρ is the server utilisation and u is the traffic intensity – with single server systems the two have the same value.

A queue may be specified by the Kendall notation, of the form

$$A/B/c/K/m/Z$$

Here A specifies the interarrival time distribution, B the service time distribution, c the number of servers, K the system capacity, m the number in the source, and Z the queue discipline. The shorter notation $A/B/c$ is often used for no limit on queue size, infinite source, and FIFO queue discipline. The symbols used for A and B are –

GI General independent interarrival time

G General service time, usually assumed independent

E_k Erlang-k time distribution

M Exponential time distribution (Markov, or random times)

D Deterministic or constant interarrival or service time

mean arrival rate	λ		
mean service rate/server	μ		
mean interarrival time	t	$=$	$1/\lambda$
number of servers	c		
time in queue	q		
time at server	s		
average time in queue	$E[q]$		
average time at server	$E[s]$		
traffic intensity	ρ	$=$	$\lambda E[s] = \frac{\lambda}{\mu}$
per-server utilisation	u	$=$	$\frac{\lambda}{c\mu} = \frac{\rho}{c}$
time in system (queue+server)	w	$=$	$q + s$
mean time in (queue+server)	W	$=$	$E[w] = E[q] + E[s]$
Number in queueing system	N	$=$	$N_q + N_s$
Mean number in system	L	$=$	$E[N] = \lambda W = E[N_q] + E[N_s]$
Mean number in queue	L_q	$=$	λW_q

Table 1: Important Relations

3 Some Important Relations

In the examples of Table **??** we speak of the combination of (queue+server) as being the "system", ie between arrival at the queue and departure after service.

The following sections will derive some of the queueing equations for the more important queueing strategies and present some other results for each case.

4 The Random Arrival Process

This is the simplest queueing model and assumes that an arrival process is a combination of independent events; the probability of any particular customer arriving is small, but the population is large and the probability of some customer arriving is finite. This situation is described by the Poisson distribution; for an arrival rate λ, the probability of n customers arriving in a time t is

$$P_n[t] = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

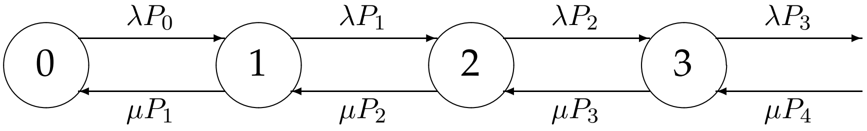
The probability $a(t)\delta t$ that the interarrival time is between t and $t + \delta t$ is simply the probability of no arrivals in time t, followed by one arrival in the time δt. Thus $a(t)\delta t = P_0(t)P_1(\delta t) = e^{-\lambda t}\lambda \delta t e^{-\lambda \delta t}$. As $\delta t \to 0, e^{-\lambda \delta t} \to 1$ and $a(t)\delta t = \lambda e^{-\lambda t}$. The inter-arrival times follow a negative exponential distribution. An exponential arrival distribution is usually plausible, especially with large customer populations. The usual assumption of an exponential service distribution is usually questionable and is often defensible only on the grounds that a solution is otherwise impossible.

An underlying assumption is that the random arrival model has no memory; each arrival (or service) is a separate event which is independent of what has happened before. The probability of its occurrence is independent of the time which that customer was away from the system or was being serviced. In contrast, for the important case of a constant or deterministic service time the probability of service completion is mostly zero except for one time since service commenced – the system has memory and the mathematics is far more complex, if it is indeed possible.

5 Single Server Model $M/M/1$, or $M/M/1/\infty/\infty/$FIFO

This model assumes exponential interarrival and service time distributions, a single server, and no limit on queue lengths. For this case the traffic intensity ρ is equal to the server utilisation u.

Consider a queueing system with an arrival rate λ, service rate μ, and a probability $P_j(t)$ of having j customers (including that being served at time t). We may represent the system by a state diagram where state j corresponds to having j customers in the system. Movement between the states is by customers arriving (moving to a "higher" state), or completing service (moving to a "lower" state). For the present assume that the arrival and service rates, λ and μ, are independent of the state. Later models do not have this simplification. [*A subtle point is that the arrival and service rates are assumed to be constant (in the steady state) so that a state transition is an independent event, which does not depend on the time spent in the state; this condition is satisfied only for random arrivals, or exponentially distributed inter-arrival times.*] Note also that $\Sigma P_k = 1$. The steady state solution must be averaged over a time which is large compared with both of $1/\lambda$ and $1/\mu$. The state S_k corresponds to the queueing system containing k customers and occurs with a probability P_k.



Consider first the two states S_0 (empty system) and S_1 (a single customer). The state moves from S_0 to S_1 by a customer arriving, and the change occurs with frequency λP_0. Similarly the state moves from S_1 to S_0 by the customer completing service, and the change occurs with frequency μP_1. In equilibrium the two must be equal and

$$\lambda P_0 = \mu P_1$$

Considering the probabilities of entering and leaving S_1, we have that

$$\lambda P_0 + \mu P_2 = \mu P_1 + \lambda P_1$$

But as $\lambda P_0 = \mu P_1$, we have $\lambda P_1 = \mu P_2$ and in general $\lambda P_k = \mu P_{k+1}$. Setting $\rho = \lambda/\mu$, and solving in turn for each P_k, we find that

$$P_k = \rho^k P_0$$

These probabilities must total 1, giving

$$\sum_{k=0}^{\infty} \rho^k P_0 = 1$$

As P_0 is the probability that the system is idle and ρ is the probability that the system is busy, it is clear that $P_0 = 1 - \rho$. Using the sum of a geometric series, we then find that

$$P_k = (1-\rho)\rho^k$$

The mean number of customers in the system, N, is then

$$N = \sum_{k=0}^{\infty} k P_k = (1-\rho) \sum_{k=0}^{\infty} k \rho^k$$

From which $\quad N = \frac{\rho}{1-\rho} \quad$ **(number in system)**

As the average number actually being served is ρ, the average number waiting in the queue is $N - \rho$,

giving
$$\begin{aligned} L_q &= \tfrac{\rho}{1-\rho} - \rho \\ &= \tfrac{\rho^2}{1-\rho} \end{aligned}$$
(number in queue)

A very important result which is intuitively obvious, but very hard to prove for the general case, is "Little's formula". This states that if the N customers are in the system for an average time T,

then $\quad N = \lambda T \quad$ **(Little's formula)**

As the number waiting is $\quad L_q = \frac{\rho^2}{1-\rho}$,

the time spent waiting is
$$\begin{aligned} W_q &= \tfrac{L_q}{\lambda} \\ &= \tfrac{\rho}{1-\rho} \tfrac{\lambda}{\mu} \tfrac{1}{\lambda} \\ &= \tfrac{\rho}{1-\rho} \tfrac{1}{\mu} \end{aligned}$$
(time in queue)

Average time in the system $\quad W = \frac{N}{\lambda} = \frac{\rho}{1-\rho}\frac{1}{\lambda}$

multiplying by $\mu \quad W = \frac{1}{\mu - \lambda} \quad$ **(time in system)**

Some basic results for the $M/M/1$ queue are shown in Table **??**

6 Scaling Effect

An important phenomenon in queueing systems is the "scaling effect". It may be assumed that if we have a single computer shared among n users, and replace it with n computers, each of $1/n$

Some basic relations for the M/M/1 queue are

Prob of n customers in system	P_n	$=$	$(1-\rho)\rho^n$
Prob of no customers in system	P_0	$=$	$(1-\rho)$
Prob of more than n cust. in system	$P[N>n]$	$=$	ρ^{n+1}
Avg no. of customers in system	$E[N]$	$=$	$\frac{\rho}{1-\rho}$
Average queue length	L_q	$=$	$\frac{\rho^2}{1-\rho}$
Waiting time distribution	$w(t)$	$=$	$\rho(\mu-\lambda)e^{-t(\mu-\lambda)}dt$
Average waiting time	W_q	$=$	$\frac{\rho}{1-\rho}\frac{1}{\mu}$
Prob. of waiting time $> t$		$=$	$\rho e^{-t(\mu-\lambda)}dt$
Average time in system	W	$=$	$E[w] = \frac{1}{(\mu-\lambda)}$
Variance of number in system		$=$	$\frac{\rho}{(1-\rho)^2}$
Probability of spending longer than t in system		$=$	$e^{-t(\lambda-\mu)}$
r-th percentile of waiting time —	$\pi_w(r)$	$=$	$\frac{E[s]}{1-\rho}\log_e\frac{100}{100-r}$
(r% of customers wait less than this time)		$=$	$E[w]\log_e\frac{100}{100-r}$
90th percentile	$\pi_w(90)$	$=$	$2.303E[w]$
95th percentile	$\pi_w(95)$	$=$	$2.996E[w]$
r-th percentile of time in queue	$\pi_q(r)$	$=$	$E[w]\log_e\frac{100\rho}{100-r}$
		$=$	$\frac{E[q]}{\rho}\log_e\frac{100\rho}{100-r}$
90-th percentile of time in queue	$\pi_q(90)$	$=$	$E[w]\log_e(10\rho)$
95-th percentile of time in queue	$\pi_q(95)$	$=$	$E[w]\log_e(20\rho)$

Table 2: Important results for the $M/M/1$ queue

the power, that the overall response time is unchanged and we have more conveniently located computers. This plausible argument is wrong.

Assume that we have old values of λ and μ, and new values of λ/n and μ/n, then the expected times in queue and times in system are –

$$\begin{array}{lrcl}
\text{Old time in queue} & E[q]_{old} & = & \frac{\rho}{\mu}\frac{1}{1-\rho} \\
\text{New time in queue} & E[q]_{new} & = & \frac{\rho}{\mu/n}\frac{1}{1-\rho} \\
 & & = & n\frac{\rho}{\mu}\frac{1}{1-\rho} \\
\text{Old time in system} & E[w]_{old} & = & \frac{1}{\mu}\frac{1}{1-\rho} \\
\text{New time in system} & E[w]_{new} & = & \frac{1}{\mu/n}\frac{1}{1-\rho} \\
 & & = & n\frac{1}{\mu}\frac{1}{1-\rho}
\end{array}$$

The mean number waiting in the queue and the mean number waiting in the system are unchanged, but we find that

$$\frac{E[q]_{new}}{E[q]_{old}} = \frac{E[w]_{new}}{E[w]_{old}} = n$$

The waiting times have therefore increased in inverse ratio to the computer power. The general rule is that separate queues to slower servers should be avoided where possible. It is better to have a single fast server.

7 Example of an M/M/1 situation

An office has one workstation which is used by an average of 10 people per 8-hour day, with the average time spent at the workstation exponentially distributed, and a mean time of 30 minutes. Assume an 8-hour day.

The arrival rate is λ = 10 per day = 1/48 per minute, giving a server utilisation of $\rho = 30/48 = 0.625$; the workstation should therefore be idle for 37.5% of the time.

However the full situation is shown in Figure **??**.

Thus, the average waiting time is 50 minutes, but for those who do not get immediate access the waiting time is 80 minutes! More complete calculations show that one third of the customers must spend over 90 minutes in the office for 30 minutes of useful work, and 10% must spend over 3 hours.

Providing 2 workstations decreases the average waiting time to 3.25 minutes, with only 10% having to wait more than 8.67 minutes.

8 Multiple Server Model $M/M/c$, or $M/M/c/\infty/\infty/$FIFO

This is the situation of a queue at a bank counter, where there are c servers. If there are fewer than c customers an arriving customer can be serviced immediately; if more than c customers arrive

Probability of more than 1 customer in system	$P[N \geq 2]$	$=$	ρ^2	$=$	0.391
Mean steady-state number in system	$L = E[N]$	$=$	$\dfrac{\rho}{1-\rho}$		
		$=$	$\dfrac{0.625}{1-0.625}$	$=$	1.667
Mean time customer spends in system	$W = E[w]$	$=$	$\dfrac{1}{\mu-\lambda}$	$=$	80 minutes
Mean number of customers in queue	L_q	$=$	$\dfrac{\rho^2}{1-\rho}$	$=$	1.04
Mean length of non-empty queue	$E[N_q \mid N_q > 0]$	$=$	$\dfrac{1}{1-\rho}$	$=$	2.67
Mean time in queue	$E[q]$			$=$	50 minutes
Mean time in queue for those who wait	$E[q \mid q > 0]$	$=$	$E[w]$	$=$	80 minutes

Figure 1: Example of M/M/1 Queueing system

they must wait for the next available server. The analysis follows the general approach taken earlier for the $M/M/1$ queue.

Assuming that all c servers are identical, with service rate μ, we have as before that $\lambda P_0 = \mu P_1$. Considering now the transitions between P_1 and P_2, the "upward" rate is governed entirely by the arrival statistics and is still λP_1, but with two servers active in the P_2 state, the "downward" probability is now doubled; the equation is now $\lambda P_1 = 2\mu P_2$. In general, we have that $\lambda P_{k-1} = k\mu P_k$, for all values of k up to c (while there is no waiting queue and all arrivals can be serviced immediately). Beyond that all servers are busy, the input queue builds up and the downward rate remains at $c\mu$.

Solving for P_j gives	P_j	$=$	$\frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j P_0$	for $j = 0, 1, \ldots, c$
or, letting $\rho = \lambda/\mu$	P_j	$=$	$\frac{\rho^j}{j!} P_0$	if not all servers are busy

The states when all c servers are busy may be modelled as a queue with arrival rate λ and service rate $c\mu$. If state c, with no customers waiting but all servers busy, occurs with probability P_c, then $0, 1, 2, 3, \ldots$ customers will be queued with probabilities

$$P_c, \left(\frac{\lambda}{c\mu}\right) P_c, \left(\frac{\lambda}{c\mu}\right)^2 P_c, \left(\frac{\lambda}{c\mu}\right)^3 P_c, \ldots$$

and a total probability of
$$\frac{P_c}{1-\lambda/c\mu} = P_0\frac{\rho^c}{c!}\frac{1}{1-\lambda/c\mu} = \frac{\rho^c}{c!}\frac{c\mu}{c\mu-\lambda}P_0$$

By normalising, we get
$$\frac{1}{P_0} = \sum_{j=0}^{c-1}\frac{\rho^j}{j!} + \frac{\rho^c}{c!}\frac{c\mu}{c\mu-\lambda}$$

If $c \gg \rho$, this is nearly the series expansion for the exponential function, giving $P_0 = e^{-\rho}$, and

$$P_j = \frac{\rho^j}{j!}e^{-\rho}$$

More usually, c is finite and the approximation is inappropriate, giving

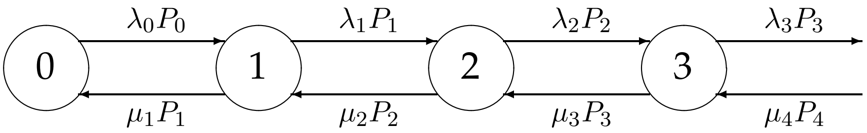
$$P_j - \frac{\rho^j/j!}{\sum \rho^i/i!}$$

Summarising the important formulae for the $M/M/c$ system –

Probability of no customers in system	$\frac{1}{P_0} = \sum_{j=0}^{c-1}\frac{\rho^j}{j!} + \frac{\rho^c}{c!}\frac{c\mu}{c\mu-\lambda}$	
Probability of n customers in system	$P_n = P_0\frac{\rho^n}{n!}$	if $n \le c$
	$P_n = P_0\frac{\rho^c(\lambda/c\mu)^{n-c}}{c!}$	if $n \ge c$
Average no. in queue	$L_q = \frac{P_0\lambda\mu\rho^c}{(c-1)!(c\mu-\lambda)^2}$	
Average no. in system	$L = L_q + \lambda/\mu$	
Average waiting time	$W_q = P_0\frac{\mu\rho^c}{(c-1)!(c\mu-\lambda)^2}$	
Average time in system	$W = W_q + 1/\mu$	
Waiting time distribution	$W_q(t)dt = \frac{P_0 c\rho^c}{c!}e^{-(c\mu-\lambda)t}dt$	
Prob of waiting longer than t	$= \frac{P_0 c\rho^c}{c!(c\mu-\lambda)}e^{-(c\mu-\lambda)t}$	

9 Solution of the general queueing equations

In the more general case, λ and μ may depend on the state (for example, in a finite population λ must decrease as each customer enters the queue and increase as each customer completes service). The argument follows the line of the earlier cases, but is rather more complex. Remember though that the earlier restriction still applies – within a given state, the values of λ and μ must be independent of the time already spent in that state.



Consider the equilibrium conditions for state j. State j is entered at rate $\lambda_{j-1}P_{j-1}$ by arrivals from state $(j-1)$ and at rate $\mu_{j+1}P_{j+1}$ by completion from state $(j+1)$. State j is left at rate $\lambda_j P_j$ by arrivals and at rate $\mu_j P_j$ by departures. The general equilibrium condition is then

$$\lambda_{j-1}P_{j-1} - (\lambda_j + \mu_j)P_j + \mu_{j+1}P_{j+1} = 0$$

For an empty queue, $\lambda_{-1} = \mu_0 = 0$, and

$$0 = -\lambda_0 P_0 + \mu_1 P_1$$

$$P_1 = \frac{\lambda_0}{\mu_1} P_0$$

and, in general

$$P_{j+1} = \frac{\lambda_j + \mu_j}{\mu_{j+1}} P_j - \frac{\lambda_{j-1}}{\mu_{j+1}} P_{j-1}$$

whence, substituting for values of j,

$$\begin{aligned} P_j &= \frac{\lambda_0 \lambda_1 \ldots \lambda_{j-1}}{\mu_1 \mu_2 \ldots \mu_j} P_0 \\ &= \frac{\lambda_0}{\mu_j} \prod_{i=0}^{j-1} \frac{\lambda_i}{\mu_i} P_0 \end{aligned}$$

As the sum over all j = 1, we get

$$\frac{1}{P_0} = 1 + \frac{\lambda_0}{\mu_1} + \sum_{j=2}^{n} \left[\frac{\lambda_0}{\mu_1} \prod_{i=1}^{j-1} \frac{\lambda_i}{\mu_i} \right]$$

The mean queue length L is then

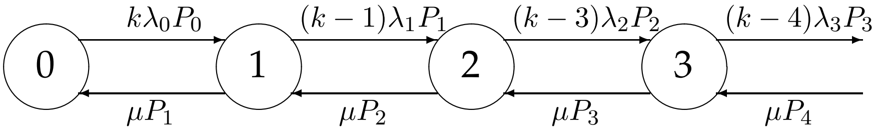
$$\begin{aligned} L &= \sum j P_j \\ &= \left[\frac{\lambda_0}{\mu_1} + 2\frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} + 3\frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2 \mu_3} + \ldots \right] P_0 \end{aligned}$$

Note that this equation is quite general – it relates the queue sizes to the arrival rates λ and service rates μ. We can get different queueing models by choosing different behaviours for the λ and μ. In most cases μ will be independent of the queue size (although the $M/M/c$ queue can regarded as a case with varying μ), but for a finite population we may find that λ decreases as customers enter the queue. Similarly, if customers are deterred by a long queue, we may find that λ decreases for large queues. The special case of multiple servers has been dealt with already, and another one is described in the next section. Other situations can be handled, provided only that the values of λ and μ can be calculated.

10 The Machine-repair model M/M/1/k/k/FIFO (or Machine-interference model)

The machine-repair is an extreme example of a finite population queueing system; the entire population may be in the system and the arrival rate zero. Some examples are –

- a machine-shop with a number of machines which work for a while and then need attention; the time to failure follows an exponential distribution (surprise!). A single maintenance worker has the job of repairing the failed machines; the repair time is again exponentially distributed. (The machine-repair model is an extreme case of a finite-population queueing system.) This model yields the extremely important concept of "walk time", which arises when a service worker (or computer, etc) visits or examines units in sequence. The walk time is the time to move from one unit to the next and is essentially non-productive or wasted time.
- a multi-processor computer with a shared memory. The processors work for a time before they need data from the memory (ie they "fail") and enter the memory queue for "servicing". They then resume operation as soon as the shared memory responds.
- a small population of users of a computer, where each user does other work for a while and then queues for the computer, thus removing one potential computer user.
- a polled or sequential access computer network where users work preparing input and then need service from the central computer (ie they "fail") and the computer polls or visits each in turn.
- A client-server system, where clients make requests of a central server and must wait for the response before they can proceed.



For this model we have –

number of machines		$=$	k
average time to machine failure		$=$	$E[o]$
average time to repair		$=$	$E[s]$
average service rate	μ	$=$	$1/E[s]$
probability of no machine needing service		$=$	P_0
serviceman utilisation	ρ	$=$	$\lambda/\mu = E[s]/E[o]$
failure rate per active machine	λ	$=$	$1/E[o]$
average number of failed machines		$=$	L
The failure rate with N machines under repair (ie. $k-N$ in service) is	λ_N	$=$	$(k-N)\lambda$
Then, putting $\rho = \lambda/\mu$	P_1	$=$	$k\rho P_0$
	P_2	$=$	$k(k-1)\rho^2 P_0 \qquad \ldots$ etc
The total probability is	1	$=$	$P_0\left[1 + k\rho + k(k-1)\rho^2 ... + k!\rho^k\right]$
or	P_0	$=$	$\left[1 + k\rho + k(k-1)\rho^2 + \ldots + k!\rho^k\right]^{-1}$
The operator utilisation (probability that the operator is busy)		$=$	$1 - P_0$
and the machine utilisation		$=$	$\dfrac{1-P_0}{k\rho}$
The avg time a machine is broken	W	$=$	$\dfrac{k}{\mu(1-P_0)} - \dfrac{1}{\lambda}$
Alternatively	W	$=$	$k/\lambda - E[o] - E[s]$
and to calculate P_0	$\dfrac{1}{P_0}$	$=$	$\displaystyle\sum_{n=0}^{k} \frac{k!}{(k-n)!}\left(\frac{E[s]}{E[o]}\right)^n$
		$=$	$\displaystyle\sum_{n=0}^{k} \frac{k!}{(k-n)!}\rho^n$

In many computer situations a more realistic model is the $M/D/1/k/k/$FIFO, the machine repair model with constant service time. Unfortunately this does not seem to be a standard result, if indeed the results are obtainable at all.

11 More general models

The models given so far are generally simple, but the assumption of exponentially distributed service time is often inappropriate. For example, some computing situations have a constant

the average number waiting	$L_q = E[n_q] = \frac{\lambda^2\sigma_s^2 + \rho^2}{2(1-\rho)} = \frac{\lambda^2 E[s^2]}{2(1-\rho)} = \rho^2 \frac{1+\sigma_s^2\mu^2}{2(1-\rho)}$
the average number in the system	$L = E[N] = L_q + \rho = \rho + \rho^2 \frac{1+\sigma_s^2\mu^2}{2(1-\rho)}$
the average time waiting	$W_q = E[q] = L_q/\lambda$
average time in non-empty queue	$E[q \mid q > 0] = W_q/\rho$
Standard deviation of time in queue	$\sigma_q^2 = E[q^2] - W_q^2$
average time in system	$W = E[w] = L/\lambda$
mean-square time in system	$E[w^2] = E[q^2] + \frac{E[s^2]}{1-\rho}$
variance of time in system	$\sigma_w^2 = E[w^2] - W^2$
variance of number in system	$\sigma_N^2 = \frac{\lambda^2 E[s^3]}{3(1-\rho)} + \left(\frac{\lambda^2 E[s]^2}{2(1-\rho)}\right)^2 + \frac{\lambda^2(3-2\rho E[s^2])}{2(1-\rho)} + \rho(1-\rho)$

Table 3: Results for the $M/G/1$ system

service time (many types of transaction servicing), but others have a much larger "tail" than the exponential distribution. The analysis is now much more difficult because λ and μ are time and history dependent and the simple state transition models do not apply. When going to the more general service distributions it is often impossible to get the exact distribution functions, but it is possible to get the mean and standard deviations of some of the variables if the first three moments of the service time are known.

12 The $M/G/1$ system

For these formulæ shown in Table **??** we introduce the standard deviations of the time in queue, service time and time in system, denoted by σ_q, σ_s and σ_w. The first equation, for the number of customers in the queue, is a fundamental equation for all queueing systems, known as the Pollaczek-Khintchine equation.

Two approximate results for the percentiles of response times are that

$$p_{90}(w) = E[w] + 1.3\sigma_w$$

and that $p_{95}(w) = E[w] + 2\sigma_w$

13 The $M/E_k/1$ queueing system

An important case of the $M/G/1$ system is the $M/E_k/1$ system, for the Erlang-k service time distribution (a cascade of exponential servers). The earlier equation is repeated, but now writing μ instead of λ, to emphasise that it describes a service distribution rather than an arrival distribution. The average service rate is μ.

$$E_k(x) = 1 - \sum_{j=0}^{k-1} \frac{(\mu x)^j}{j!} e^{-\mu x}$$

For large values of k, the Erlang-k distribution tends towards a rectangular distribution with a cut-off of 2μ. The moments of the service time, for substitution into the Table **??** results for the $M/G/1$ system, are –

$$\begin{array}{lrcl} \text{second moment} & E[s^2] & = & \dfrac{(k+1)}{k}\mu^2 \\ \text{third moment} & E[s^3] & = & \dfrac{(k+1)(k+2)}{k^2}\mu^3 \end{array}$$

14 The $M/D/1$ system

With a constant service time s, we use the $M/G/1$ model with $\sigma_s = 0$, $E[s] = s$, $E[s^2] = s^2$, $E[s^3] = s^3$, etc. The simplest important result is that the average number waiting is half that waiting with exponentially distributed service.

$$\begin{array}{lrcl} & L_q & = & \dfrac{\rho^2}{2(1-\rho)} \\ \text{the total number in the system} & N & = & \dfrac{\rho^2}{2(1-\rho)} + \rho \\ \text{the mean time in the system} & W & = & \dfrac{(2-\rho)}{2\mu(1-\rho)} \\ \text{std devn of number in system} & \sigma_N & = & \dfrac{1}{1-\rho}\sqrt{\rho - \dfrac{3\rho^2}{2} + \dfrac{5\rho^3}{6} - \dfrac{\rho^4}{12}} \end{array}$$

Other results can be derived from the $M/G/1$ equations.

15 The Erlang B and Erlang C formulæ

A telephone exchange normally has a number of incoming lines, served by a number of switches where any switch can service any one of a group of lines. (There are often about 1000 lines to a group.) If at least one switch is free the call can be accepted immediately. If all switches are busy the result depends on the design of the exchange.

- If there is no explicit input queue, an incoming call must be abandoned forthwith. The corresponding queue model is $M/M/c/c$, ie with c servers and a system capacity of c. The probability of a call being lost is variously known as *Erlang's lost call formula* or the *Erlang B formula*, or the *first Erlang function* and is

$$P_N = \frac{\rho^N/N!}{\sum_{i=0}^{N} \frac{\rho^i}{i!}}$$

The average number of occupied servers is

$$L = \rho(1 - P_N)$$

The equation and terminology arise from a telephone exchange where a limited number of switches are available to handle incoming calls, and calls are lost if all switches are busy.

- An alternative design allows the incoming call to wait until a server is free; the corresponding model is a $M/M/c$ queue (ie $M/M/c/\infty$, with unlimited capacity). The difference from the preceding case is that there may now be a queue of waiting calls; the Erlang B function assumed that calls which were accepted immediately were lost. The probability of a call having to wait is given by the Erlang C formula

$$\begin{aligned} P_n &= P_0 \frac{c\rho^c}{c!(c-\rho)} \\ \text{where} \quad \frac{1}{P_0} &= \sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{c\rho^c}{c!(c-\rho} \end{aligned}$$

These formulæ have obvious application to many computing applications. For example, given a multi-user computer with known session statistics, how many user processes should be made available to ensure that logons are accepted with certain probability?

16 Communications Buffers

If we assume a situation where we have a concentrating node which receives messages from a number of sources and transmits them over a single aggregate output channel, we clearly have a situation where queueing is important – the messages are queued in a buffer for transmission.

If there are many data sources (terminals etc) we can probably assume Poisson statistics and an exponential input distribution. The server situation is slightly more complex – the output channel sends data at a constant rate (bit/s or byte/s), but if we assume an exponential distribution of message length, the distribution of server time per message becomes exponential and the $M/M/1$ queue model will apply.

$$
\begin{array}{lrcl}
\text{The mean number in the system is} & N & = & \dfrac{\rho}{1-\rho} \\
\text{the average delay} & W & = & \dfrac{N}{\lambda} = \dfrac{\rho}{1-\rho}\dfrac{1}{\lambda} \\
 & & = & \dfrac{1/\mu}{1-\rho} = \dfrac{1}{\mu-\lambda} \\
\text{and the average queue length} & L_q & = & \dfrac{\rho^2}{1-\rho}
\end{array}
$$

In many cases we have messages of constant length; the $M/D/1$ queue model is then applicable. If the message transmission time is μ, then the service time $\lambda = 1/\mu$. Then,

$$
\begin{array}{lrcl}
\text{the mean number in the system} & N & = & \dfrac{\rho}{1-\rho}\left(1-\dfrac{\rho}{2}\right) \\
\text{the mean time in the system} & W & = & \dfrac{1/\mu}{1-\rho}\left(1-\dfrac{\rho}{2}\right) \\
 & & = & \dfrac{1}{\mu-\lambda}\left(1-\dfrac{\rho}{2}\right)
\end{array}
$$

In both cases the change to constant service time yields the old result (M/M/1 queue) with the multiplier $(1-\rho/2)$, which is always less than 1.0. The difference is entirely due to the variation in service times with the M/M/1 queue discipline. There is little change at light loading, but as the traffic intensity approaches 1, the delay for the $M/D/1$ queue tends to half the delay for the $M/M/1$ queue.

The number in queue and the delay (for $\mu = 1.00$) are -

	number & delay	
ρ	$M/M/1$	$M/D/1$
0.1	0.111	0.106
0.2	0.250	0.225
0.3	0.429	0.364
0.4	0.667	0.533
0.5	1.000	0.750
0.6	1.500	1.050
0.7	2.333	1.517
0.8	4.000	2.400
0.9	9.000	4.950
0.95	19.000	9.975
0.99	99.000	49.995

The normal rule of thumb is that an $M/M/1$ queue becomes overloaded for $(\rho > 0.6)$. The overload point for fixed service time ($M/D/1$ queue) occurs at a rather higher traffic intensity, at about $\rho = 0.7$. We can also calculate the number of buffers to ensure an upper limit of message rejection due to buffer overload.

16.1 Example 1

Assume that a multiplexer must accept 100 messages per second and that the user can tolerate a loss of no more than 10 messages in an 8-hour day. How many buffers must provided to guarantee this service?

There are $100 \times 3600 \times 8 = 2,880,000$ messages in a day, of which no more than 10 may be lost. The probability of all buffers being full may not exceed $10/2,880,000 = 3.5 \times 10^{-6}$. Given that the probability of there being n customers in the system is ρ^{n+1}, we must have that

$$\begin{array}{rrcl} & \rho^{n+1} & < & 3.5 \times 10^{-6} \\ \text{or} & (n+1)\log_e \rho & < & -12.57 \\ & n & > & -12.57/\log_e \rho - 1 \end{array}$$

The table of n as a function of ρ, is

ρ	Number of buffers
0.2	7
0.3	10
0.4	13
0.5	18
0.6	24
0.7	35
0.8	56
0.9	119

This is another argument for ensuring that a system is operating well within its apparent capacity. Not only do delays increase with loading, but so does the probability of buffer overflows. If the multiplexer operates at only 50% of its nominal load, a pool of at least 20 buffers should be provided to queue the input messages. (If the average message length is 100 bytes so that the multiplexer must handle 10,000 bytes per second, it should be rated at 20,000 bytes per second on its output line and have at least 20 message buffers.) Most actual multiplexers can invoke flow control procedures to inhibit traffic as overload approaches, but this example indicates just how much buffering may be needed in high speed data logging with asynchronous inputs.

16.2 Example 2

An 8 channel multiplexer has a nominal capacity of 500 char/s and has an input buffer of 50 characters for each channel. Each channel may tolerate no more than one lost character per day (8 hours). What is the maximum utilisation of the multiplexer?

Each channel has a nominal capacity of 1,800,000 characters in 8 hours, giving a permitted error probability of 0.556×10^{-6} and $\rho^{51} < 0.556 \times 10^{-6}$, whence $\rho < 0.75$. Increasing the buffer to 100 characters allows ρ to reach 0.87.

17 Performance of Sequential access Networks

This analysis applies to all networks in which the right to transmit cycles in a regular manner among the stations, including token ring and token bus. The machine repair model applies when customers need only occasional service and the service agent may be idle; this analysis is more applicable where queues exist at most stations and there is continuous traffic. It also shows an alternative approach to a queueing problem based on physical arguments as much as mathematical ones.

Consider a network of N nodes or stations, with an average walk time w for data (or control) to pass from one node to its successor. Thus the time for control to pass around the entire network is $L = N.w$, the *system walk time*. In some cases there may be different walk times for data transfer and control transfer the choice is usually obvious.

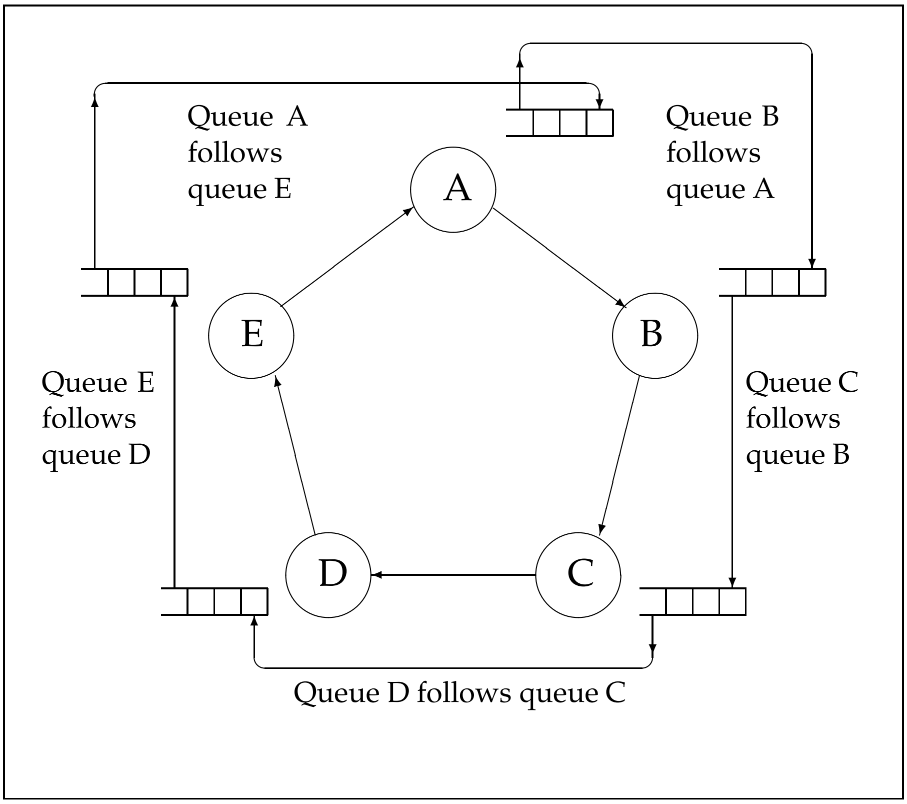
A user who wishes to transmit sees an "access time" from submitting the packet until the packet is finally sent. It has two components

- the "walk time" as control circulates to the node, and
- the queueing time, behind preceding messages (this includes the transmission time)

The access time may be derived by a simple physical argument. The time for control to circulate around the network, in the absence of any user traffic, is the walk time, L. A station which receives a message and wishes to transmit must wait, on average, for half this time before receiving the right to transmit, or a time of $L/2$. If, however, the network is busy with a utilization ρ, the right to transmit can circulate only when the network is idle; the circulation speed is reduced by the factor $(1-\rho)$ and the system walk time then becomes

$$\frac{L}{2}\frac{1}{(1-\rho)}$$

For the queueing delay, note that there are potential queues at each node and that these queues are served sequentially – the queue for the next node effectively continues on from the tail of the queue for the current node, and so on. Thus there is really just a single circular queue which is broken among the nodes and which needs transitions between nodes at appropriate times. However, messages which arrive at an arbitrary node will on average arrive at the mid-point of the effective queue and have half the usual waiting time.



The queueing delay is then half that for a single queue. The normal waiting time in a queue is

$$W_q = \frac{\rho}{1-\rho}\frac{1}{\mu}$$

For a sequential network with an arrival rate of λ per node ($N\lambda$ overall) the network utilisation is $\rho = \lambda N\overline{m}$, where $\overline{m}$ is the average message service time. For an exponential distribution of message lengths, we have that $\overline{m} = 1/\mu$, giving

$$\begin{aligned} \rho &= \frac{\lambda N}{\mu} \\ \text{then} \quad W_q &= \frac{N\lambda}{\mu^2(1-\rho)} \\ \text{Thus} \quad E(D) &= \frac{L}{2}\frac{1}{1-\rho} + \frac{N\lambda}{2\mu^2(1-\rho)} \\ &= \frac{n}{2}\frac{1}{(1-\rho)}\left(w + \frac{\lambda}{\mu^2}\right) \end{aligned}$$

Writing ρ in terms of λ and μ, the expected delay is

$$E(D) = \frac{N\mu}{2(\mu - N\lambda)}\left(w + \frac{\lambda}{\mu^2}\right)$$

For packets of constant size, the queueing delay is halved and the expected delay is

$$E(D) = \frac{N\mu}{2(\mu - N\lambda)}\left(w + \frac{\lambda}{2\mu^2}\right)$$

The delay is therefore dependent on the difference of the arrival and service rates (equivalent to the term $1/(1-\rho)$ and on the relative values of the walk time between nodes (w) and the time between message arrivals ($1/\mu$), weighted by the ratio of service time to arrival time. Given that λ and μ are fixed by the desired user traffic and the network transmission speed, it is clear that w must be as small as possible for good performance.

More exact models give slightly different results. For each station having a packet arrival rate λ, the same frame statistics (the second moment of frame length = $\overline{m^2}$ and the same walk time w, one model gives the average access delay as

$$\begin{aligned} E(D) &= \frac{t_c}{2}\left(1-\frac{\rho}{N}\right)+\frac{N\lambda\overline{m^2}}{2(1-\rho)} \\ &= \frac{L}{2}\frac{(1-\rho/N)}{1-\rho}+\frac{N\lambda\overline{m^2}}{2(1-\rho)} \end{aligned}$$

In these formulæ $\overline{m^2}$ is the second moment of the frame length and $\rho = N\lambda m$ is the network utilisation. The first term is related to the circulation of the transmission right and for low utilisation is just half the system walk time. The numerator is related to the utilisation of each node, and the denominator to the overall traffic intensity. The second term is in fact the average waiting time for an $M/G/1$ queue.

Another analysis gives a similar result, but with $(1+\rho/N)$ in the numerator. As the simple analysis presented here gives the average of these two "more exact" results, it seems to be just as good as either of the "better" approaches.

As both terms are dominated by the factor $1/(1-\rho)$ we must minimise the system walk time L to ensure performance at high utilizations. In most cases this means minimising the token latency at each node. We will also see that the token ring is much better in this regard than the token bus, simply because of the token-passing overheads.