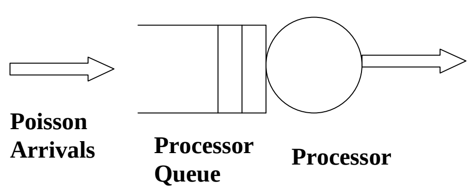
Basics of Queuing Network Models

Basics of Queuing Networks

- **Single Server Networks**
  - **Open Networks**
  - **Closed Networks**
- **Multi-Server Networks**
  - **Routing**
    - **Probabilistic**
    - **Deterministic**
  - **Open Networks**
  - **Closed Networks**
- **Some Example Problems**

Basics of Queuing Network Models

Single Server Open Networks



Parameters

- **Service Order - Scheduling Discipline**
- **Arrival Rate - λ – number of jobs arriving for service per unit time**
- **Interarrival Time - Average time between arrivals = $1/\lambda$**
- **Service Time -(S) - Random Variable - Time spent by server to service a job.**
- **Mean Service Time - Expected value of S.**
- **Average Service Rate - $\mu = 1/E[S]$**
- **Size of a job - Service requirement for a job.**

Basics of Queuing Network Models

Performance Metrics for Single Server Open Networks

Response Time - T_S = t(departure) - t(arrival)
Mean response time $E[T_S]$

Waiting Time - T_Q **- time in queue**

Throughput of Server - X - Rate at which server completes jobs

Number of Jobs in System - N_S **- the number of jobs in the queue plus the job, if any, in the server.**

Number of Jobs in Queue - N_Q

Basics of Queuing Network Models

Throughput for a Single Server

U = Fraction of time that server is busy
= (average service time required by a job)/ (average time between arrivals)
$= (1/\mu)/(1/\lambda) = \lambda/\mu$

$X = U * \mu = (\lambda/\mu) * \mu = \lambda.$

Throughput is independent of service rate!

How can this be?

Basics of Queuing Network Models

Properties of Single Server Open Networks

1. As λ increases, all of the performance metrics increase.

2. As μ increases, all of the performance metrics decrease.

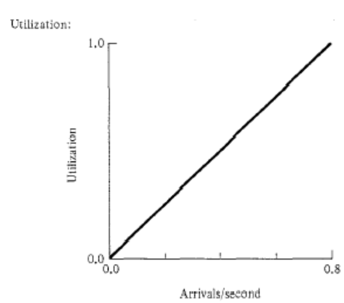
3. If $\lambda > \mu$, then N_Q grows without bound over time so we require $\lambda < \mu$.

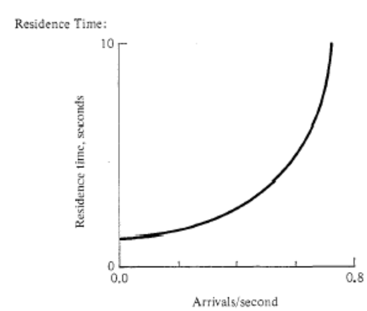
4. If λ and μ are taken from constant distributions $T_Q = 0$ and $T_S = S$.

5. Therefore queuing arises from variability in arrival and service distributions.

Basics of Queuing Network Models

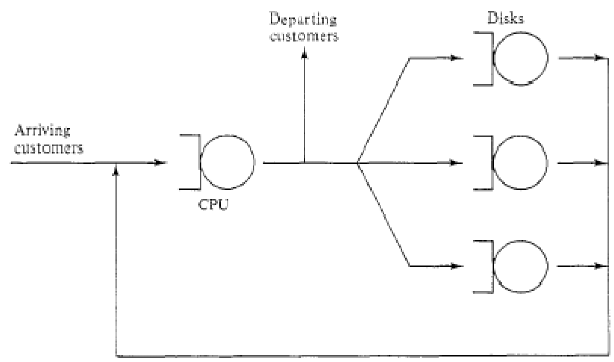
Let one customer arrive every two seconds, or 0.5 customers/second.
Let the the average service requirement of a customer be 1.25 second).





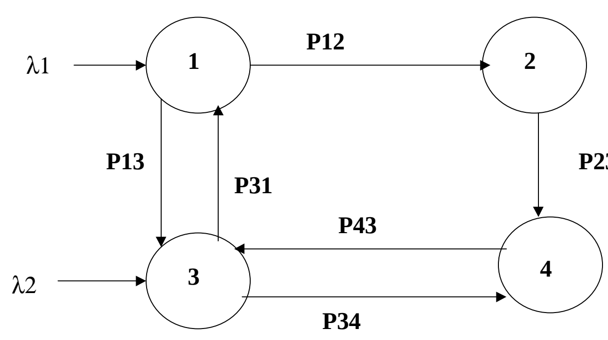
Basics of Queuing Network Models

Queuing Networks with Multiple Servers - CPU/Disk Configuration



Basics of Queuing Network Models

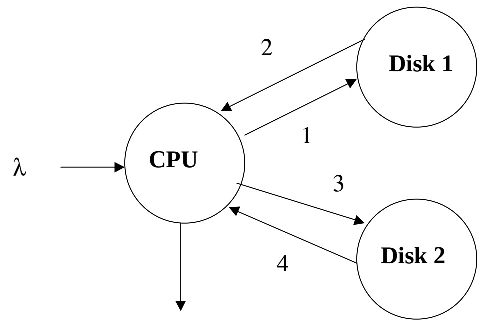
Open Networks of Servers



Probabilistic Routing

Basics of Queuing Network Models

Open Networks of Servers



Deterministic Routing

Basics of Queuing Network Models

Performance Metrics for Multi-Server Networks

Expected Values by Server

$E[N_i]$ - Number of Jobs at Server i.
$E[T_i]$ - Expected time a job spends at Server i.

Device Utilization - Let T be the observation period and B the busy fraction.

U_i = B/T for the ith device.

Device Throughput - Let T be the observation period and C the total number of jobs completed during the period.

$X_i = C/T = \mu_i * U_i$

Basics of Queuing Network Models

Throughput for a Multiple Server Network with Probabilistic Routing

$\mathbf{X} = \Sigma\, \mathbf{r_i}$

$\mathbf{X_i} = \lambda_i = \mathbf{r_i} + \Sigma\, \lambda_i * \mathbf{P_{ji}}$

This relation defines a set of similtaneous linear equations.

Throughput for a Finite Capacity Queue

X = U * μ **- but U is not defined by** λ/μ.

Why?

Basics of Queuing Network Models

Closed Networks - The Arrival Rate at the System is Defined by a Network of Queues.

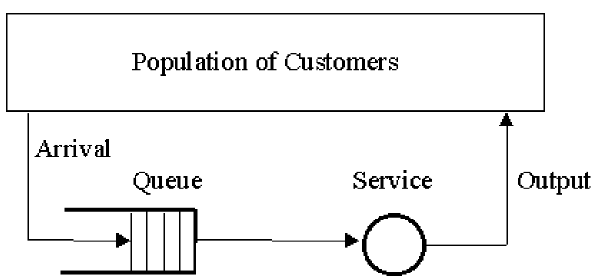
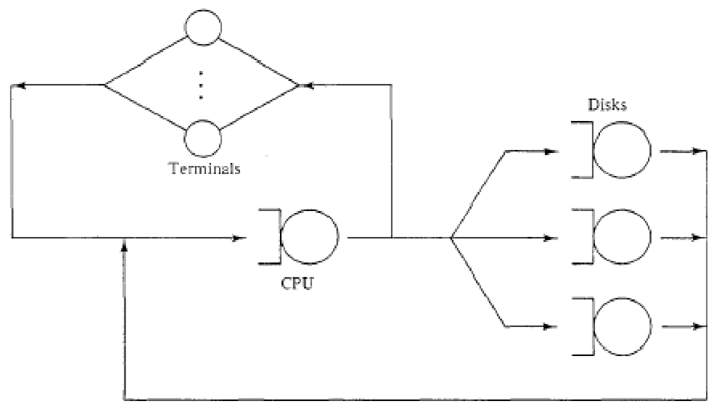


Figure 1

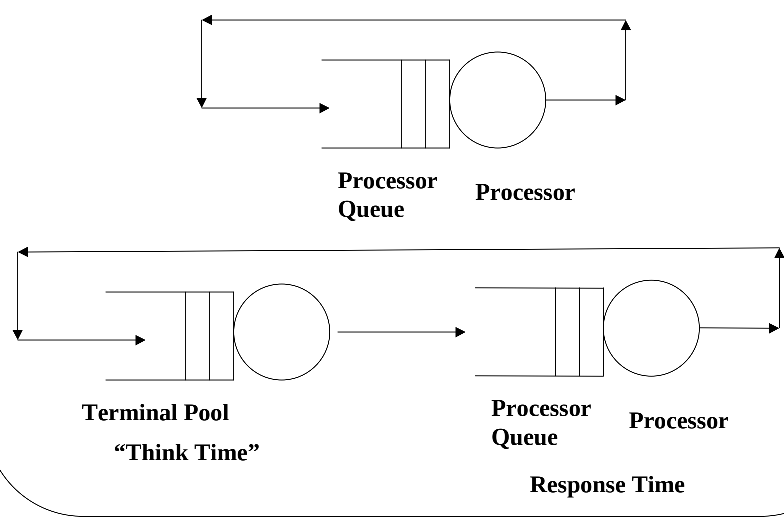
Basics of Queuing Network Models

Each user is modeled as a server with a server time which represents user "think time."



Basics of Queuing Network Models

Closed Networks



General Behaviors of Closed Networks

N - number of jobs in system.

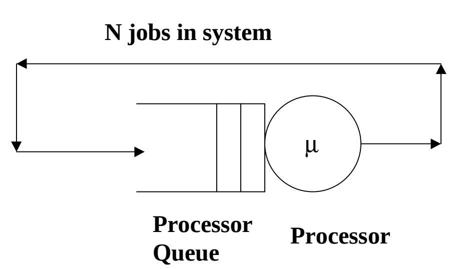
Z - Think time - delay between request for service.

Design Goal - Maximize N subject to upper bound on E[Response Time].

Question - Would you model a web server as an open network or as a closed network?

Design Goal - Maximize throughput of system

Question - What strategies will be different for these two goals?



Throughput $X = \mu$
So closed networks are different!

Open Systems
- **X is independent of service rate**
- **Throughput and response time are not related.**

Closed Systems
- **X depends on service rates of servers.**
- **Higher Throughput = Lower Response Time**

Little's Law

The average number of customers in a queueing system N is equal to the average arrival rate of customers to that system λ, times the average time a customer spends in that system T:

$$N = \lambda * T$$

This law which can be understood intuitively is a formal result of queueing theory and does not depend on any specific assumptions regarding the arrival time distribution, the service time distribution, the number of servers or the queuing discipline.

Differences Between Open and Closed Networks

Little's Law for Open Systems

$$E\{N_S\} = \lambda\, E\{T_S\}$$

$E\{N_S\}$ - Expected number of jobs in the system.

$E\{T_S\}$ - Expected mean time in the system of a job.

Little's Law is independent of almost everything!

1. Job arrives and sees $E\{N_S\}$ jobs in the system (including self.)
2. Expected time for each customer to complete is $1/\lambda$.
3. So expected time until customer completes is

$$E\{T_S\} = (1/\lambda) * E\{N_S\}$$

Basics of Queuing Network Models

Little's Law for Closed Systems

$N = X * E\{T_S\}$

N = number of jobs in system.

X is throughput

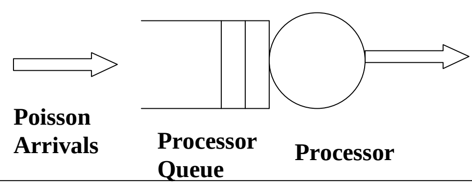
$E\{T_S\}$ - is mean time in system of a job. (Consider definition of system.)

Basics of Queuing Network Models

Example Problem - Processor Power

We are told that the arrival rate of requests for work for a single server process will double in the near future. The system has a satisfactory response time given the present load. We can assume that all other resources are sufficient for the doubled workload. The problem is to predict the speed of the CPU necessary to maintain the same response time. The arrival rate for units of work is reasonably well represented by a Poisson distribution with arrival rate, λ. We will not alter the scheduling algorithm for the CPU.

The problem statement suggests a simple model should be sufficient. In fact, this sounds like a single queue problem.



Basics of Queuing Network Models

Assuming this simple model is reasonable, what additional information do we need?

Assuming you have all the information you need, estimate the required processor speed.

Are there informal or intuitive arguments that lead you to the answer?

How would you go about solving the model mathematically or by simulation?

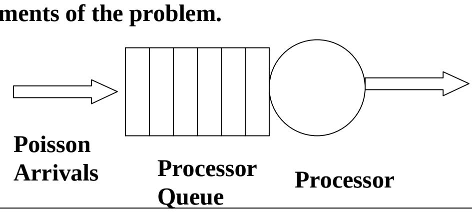
Basics of Queuing Network Models

Example Problem - Finite Buffer Capacity Problem

Consider a system which is processing requests for storing large files. It has a finite buffer capacity, specifically room for just six jobs. So if the queue for service exceeds six jobs an incoming job is lost. The variance in arrival rate (Poisson with rate λ) is causing some loss of jobs. When should we:

a) Buy more memory to double the buffer space?
b) Buy a faster processor to double the processor speed?
c) Do some mixture of the above?

This also sounds like a simple single queue model might be capture the essential elements of the problem.



Basics of Queuing Network Models

Assuming this simple model is reasonable, what additional information do we need?

What factors are we ignoring in choosing this simple model?

Are there informal or intuitive arguments that lead you to the answer?

How would you go about solving the model mathematically or by simulation?

Basics of Queuing Network Models

Example Problem - Closed Interactive System - Clerical Data Entry

The staff working for a telephone market survey company enters data online. The company is becoming very successful and wants to add to the number of call-persons and increase productivity of its staff. Therefore an increase in throughput of the processing system is needed. The data entry clerks enter a job, wait for a response from the data base, look at the data returned and then submit another job.
The current processing system includes some terminals connected by a network to a data base server. The data base server includes a CPU, a fast disk, a slow disk and some other devices.

We have done some measurements on the system and know the following things:

A) Data was collected over 900 seconds.
B) The total busy fraction of each device is known.
C) The number of completions at each device is known.
D) The total number of jobs completing is known.
E) The mean user delay between submitting jobs is known.

Basics of Queuing Network Models

The goal is to decide among options for increasing the through put of the system. Should we:

a) Buy a faster CPU?
b) Balance the load across the disks?
c) Buy a second fast disk?
d) Or something else.

Is this question complete and well-posed?

What other information do we need in order to determine the most robust path to enhance throughput?

What kind of model is needed to resolve this problem?

Basics of Queuing Network Models

One or Several Processors?

1. The jobs to be run on a processor configuration are non-preemptible. You want to minimize the mean response time for the jobs. Should you use one CPU of speed s or n slower CPUs of speed s/n?

What additional information do you need to construct a model which can be used to resolve this question?

2. The jobs to be run on a processor configuration are preemptible. You want to minimize the mean response time for the jobs. Should you use one CPU of speed s or n slower CPUs of speed s/n?

What additional information do you need to construct a model which can be used to resolve this question?

Task Assignment on a Configuration of Distributed Servers

Examples:

- **Web site load balancers**
- **Distributed compute servers**

Question:

What algorithm do we use to assignment jobs to resources in order to minimize mean response time?

Assume all jobs need only one server.

There are many possibilities:

- **random, round-robin, shortest local queue, least work in local queue, etc.**

What information do you need to resolve this problem?