Reversibility

Fact: the output of an *M/M/1* **(or** *M/M/m*, *M/M/∞***)**

queue with arrival rate λ is a Poisson

Process of rate λ. (Burke's Thm)

Follows from reversibility:

A Markov chain has the property that

P **[future | present, past] =** *P* **[future | present]**

i.e., conditional on the present state, futures states and past states are independent.

Thus

P **[past | present, future] =** *P* **[past | present]**

$$P[X_{n-1} = j \mid X_n = i, X_{n+1} = i_2, \ldots.] = P[X_{n-1} = j \mid X_n = i] \stackrel{def}{=} P^*_{ij}$$

The state sequence, run backward in time, in steady state, is also a Markov chain

$$P_{ij}^{*} = P[X_{n-1} = j \mid X_n = i] =$$

$$\underbrace{P_{ij}^{*}} = \frac{P_{ji}p_j}{p_i} = \frac{P[X_n = i \mid X_{n-1} = j]P[X_{n-1} = j]}{P[X_n = i]}$$

transition probabilities for reverse chain

A Markov chain is reversible if $P_{ij}^{*} = P_{ij}$

[All birth-death processes are reversible, since $p_i P_{ij} = p_j P_{ji}$]

$$P[X_{n-1} = j \mid X_n = i, X_{n+1} = i_1, \ldots., X_{n+k} = i_k$$

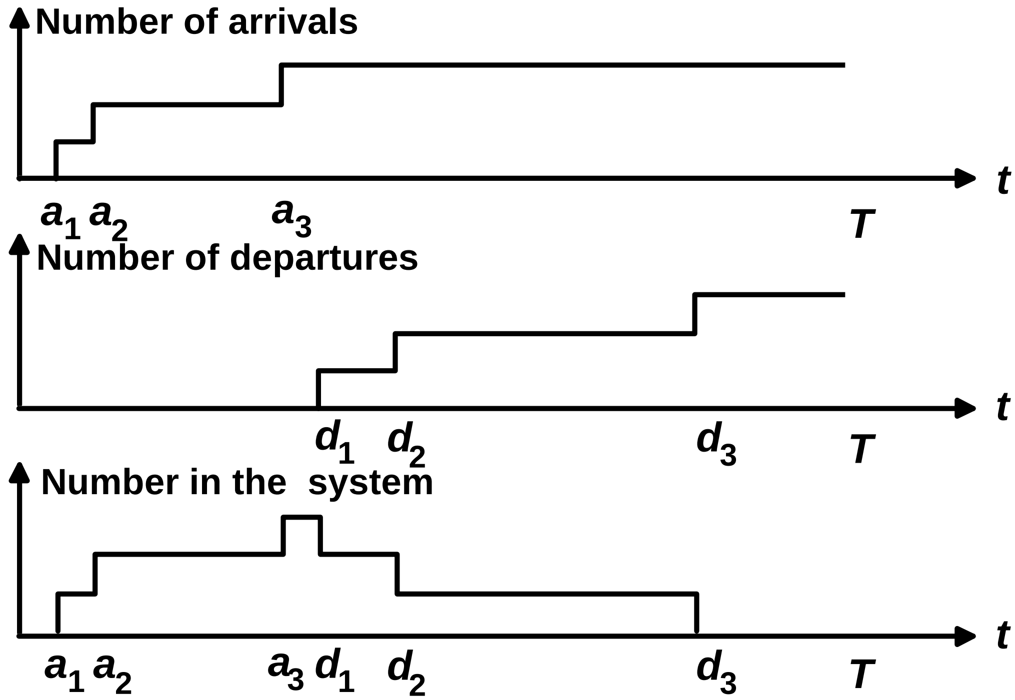
$$= \frac{P[X_{n-1} = j, X_n = i, X_{n+1} = i_1, \ldots., X_{n+k} = i_k]}{P[X_n = i, X_{n+1} = i_1, \ldots., X_{n+k} = i_k]}$$

$$= \frac{P[X_{n-1} = j, X_n = i]P[X_{n+1} = i, \ldots., X_{n+k} = i_k \mid X_{n_j} = j, X_n = i]}{P[X_n = i]P[X_{n+1} = i_1, \ldots., X_{n+k} = i_k \mid X_n = i]}$$

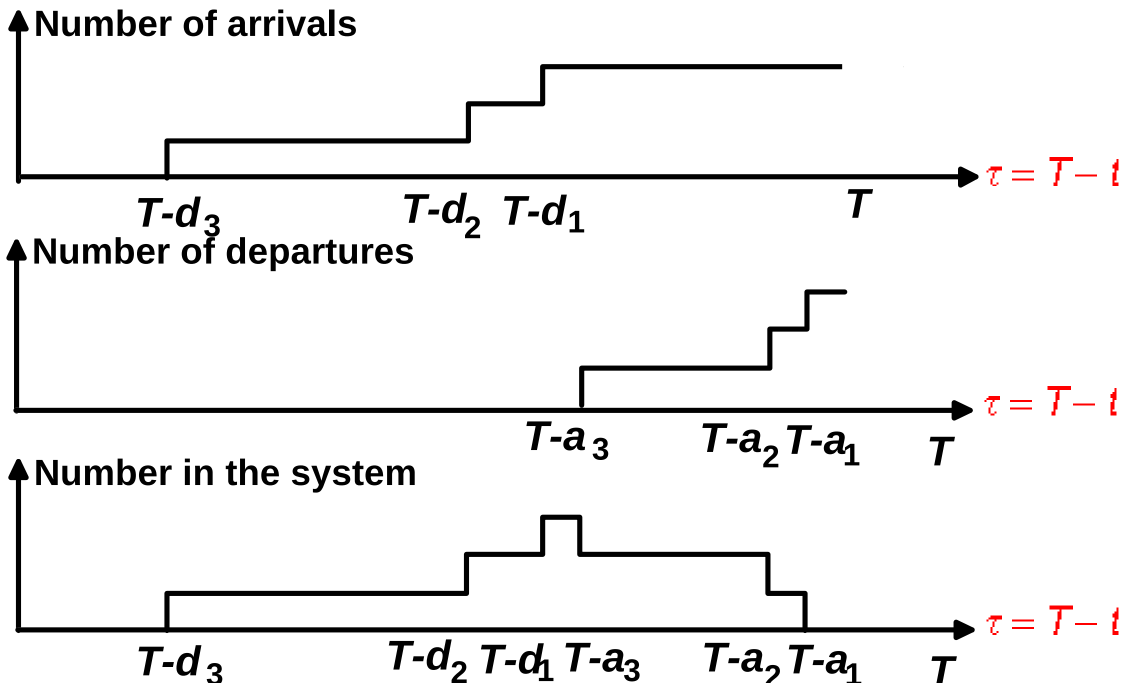
$$= \frac{P[X_{n-1} = j, X_n = i]}{P[X_n = i]}$$

$$= \frac{P[X_{n-1} = j]P[X_n = i \mid X_{n-1} = j]}{P[X_n = i]}$$

$$= \frac{p_j P_{ji}}{p_i}$$

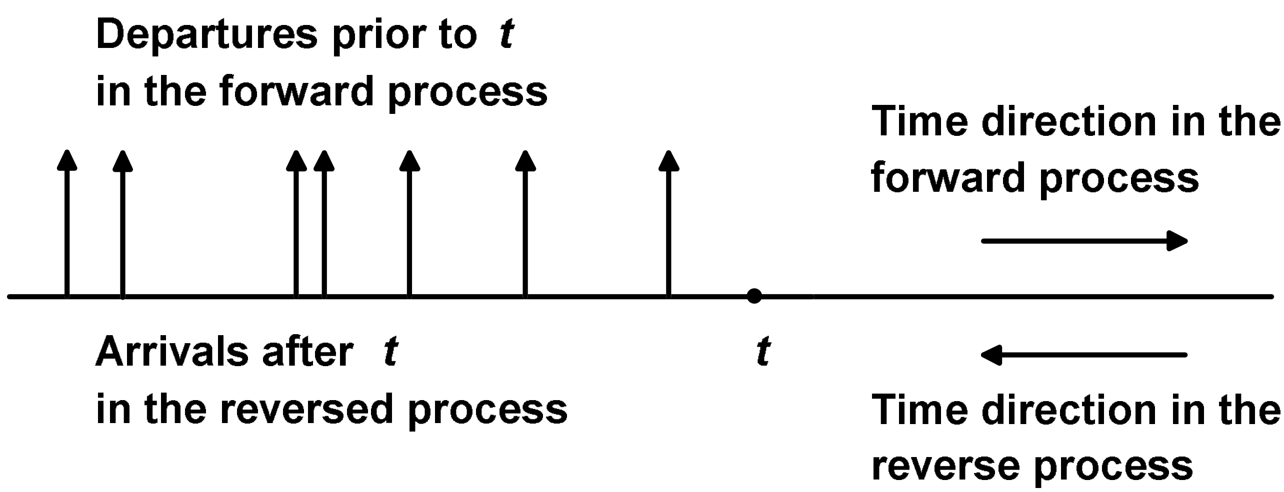


(a)



(b)

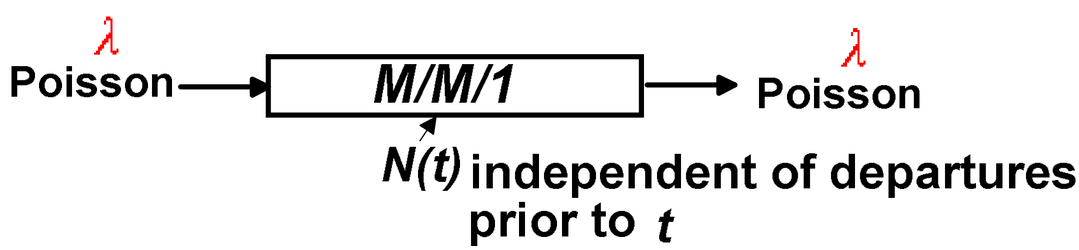
Since the backward process is statistically the same as the forward process, and the arrival process in the forward system is Poisson, the arrival process in the reverse system is also Poisson.
Since the arrivals of the reverse system corresponds to the departures of the forward system, the departure process of the forward system is Poisson.



Customer departures prior to time *t* in the forward system become customer arrivals after time *t* in the reversed system

Also the state of the system is independent of past departures (this is the second part of Burke’s Thm)

e.g.



$N(t)$ = # of customers

Note:

Conditional on the system being nonempty, the output has exponentially distributed interdepartures at rate $\mu > \lambda$.

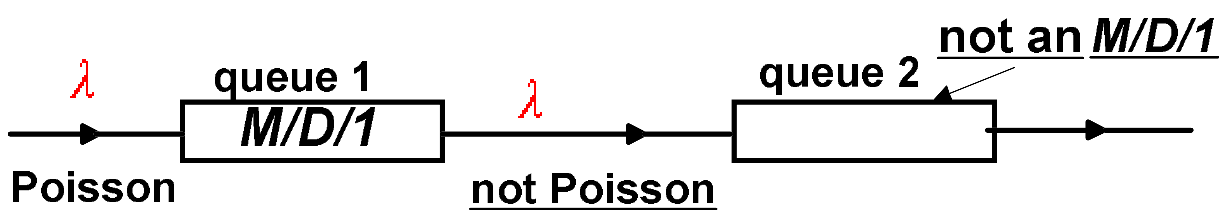
Conditional on the system being empty, the system has no output.

Thus it is plausible that the unconditional output is Poisson, and reversibility shows it.

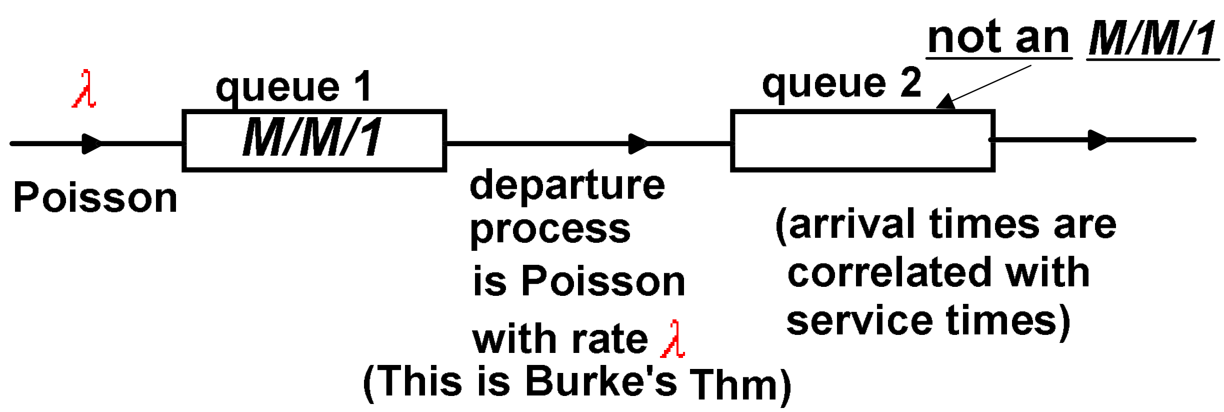
Networks of queues

There are different of carrying over single queue results:

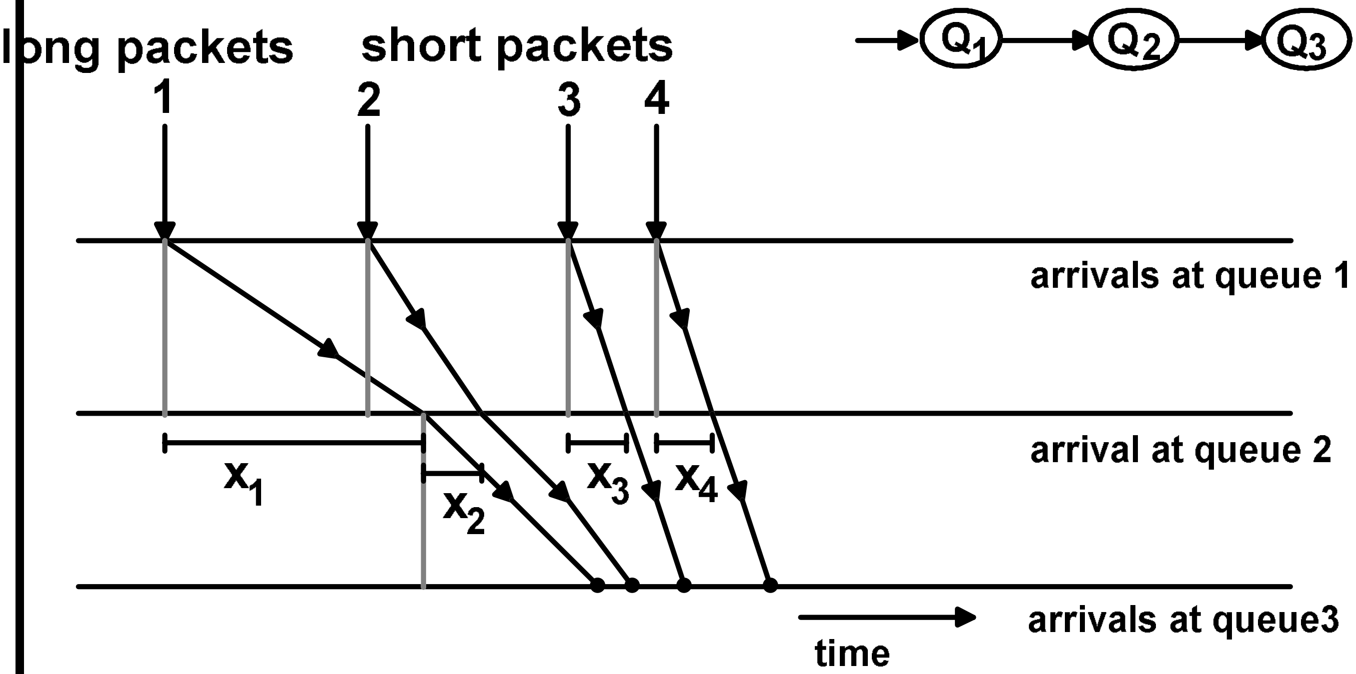
Example 1:



Example 2:



Effect of long packets



Arrivals at subsequent queue between become bursty (like many cars following behind a slow truck in a highway)

A long packet will typically wait less time at the second queue than short packets will.

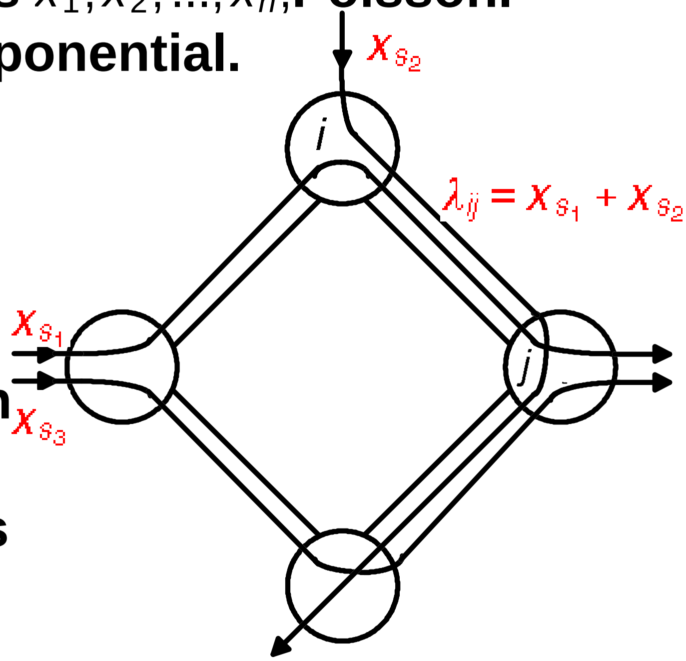
Analysis using Kleinrock's independence approximating assumption

n **traffic streams with rates** $x_1, x_2, ..., x_n$,**Poisson.**

μ_{ij} **:service rate on** *(i,j)*, **exponential.**

$\lambda_{ij} = \sum x_s$

all packet stream s crossing link *(i,j)*



Assume:

a) no feedback

b) Kleinrock's assumption

Then each queue behaves like an *M/M/1* **queue:**

$P(n_{ij}$ **customers in queue** $(i,j)\ ) = (1-\rho_{ij})\rho_{ij}^{n_{ij}}, \rho_{ij} = \frac{\lambda}{\mu}$

Average # of customers in queue or in service at link *(i,j)* **:**

$$N_{ij} = \frac{\rho_{ij}}{1-\rho_{ij}}\ ,\quad T_{ij} = \frac{1}{\mu_{ij}-\lambda_{ij}}$$

Tatol # of customers in the network:

$$N = \sum_{(i,j)} \frac{\rho_{ij}}{1-\rho_{ij}}\ ,\quad T = \frac{N}{\sum_s x_s}$$

Note: for datagram, $\lambda_{ij} = \sum x_s \cdot f_{ij}(s)$

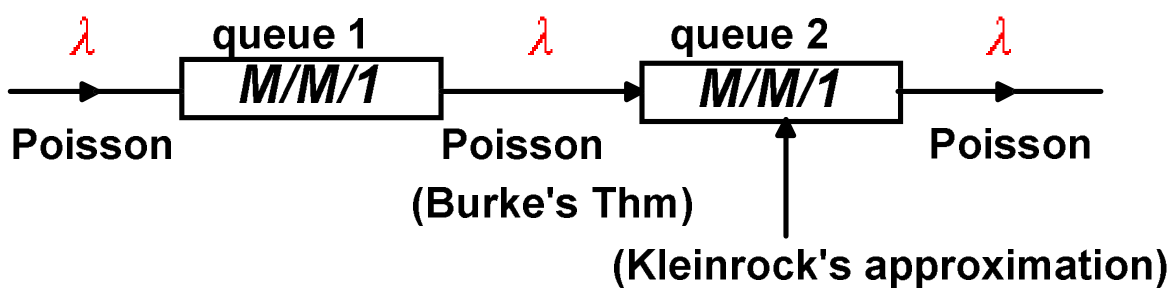
all packet stream s crossing link *(i,j)*

where $f_{ij}(s)$**= fraction of streams s packets that use** *(i,j)*

Example:

Assume that the service times of packets at queue 1 and queue 2, are independent and independent of the arrival process.

Under this approximating assumption the second queue is also *M/M/1*



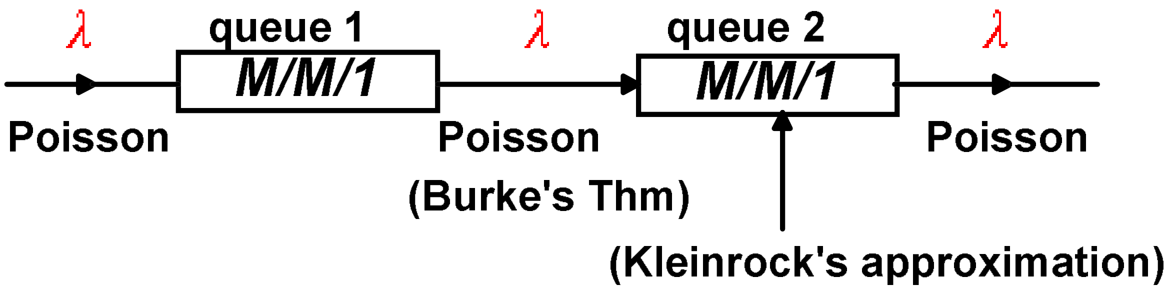
queue 1 : $P(X_n = i) = \rho_1^i \cdot (1 - \rho_1)\ ,\ \rho_1 = \frac{\lambda}{\mu_1}$

queue 1 : $P(Y_n = j) = \rho_2^j \cdot (1 - \rho_2)\ ,\ \rho_2 = \frac{\lambda}{\mu_2}$

under Kleinrock's approximation

$$P(X_n = i, Y_n = j) = \rho_1^i \cdot (1 - \rho_1) \times \rho_2^j \cdot (1 - \rho_2)$$

Proof:



- **The state X of queue 1 is independent of past departures (Burke's Thm)**
- **The state Y of queue 2 is determined by past departures from queue 1 and by service times in queue 2.**
- **assuming service times in queue 2 are independent of queue 1 service times (Kleinrock's approximation), the state Y of queue 2 is independent of X.**

So

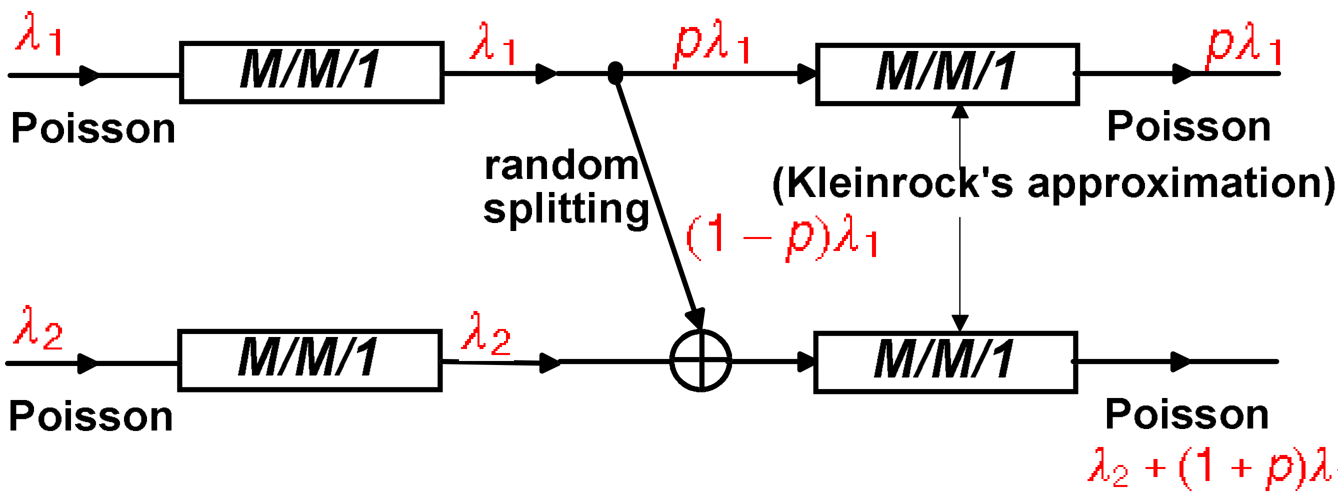
$$P(X_n = i, Y_n = j) = P(X_n = i) \cdot P(Y_n = j) = \rho_1^i \cdot (1-\rho_1) \times \rho_2^j \cdot (1-\rho_2)$$

Note:

Independence is by no means trivial (we used Burke's theorem in addition to Kleinrock's assumption)

This can be generalized to network with no feedback if input streams are independent and output are split independently.

Ex.

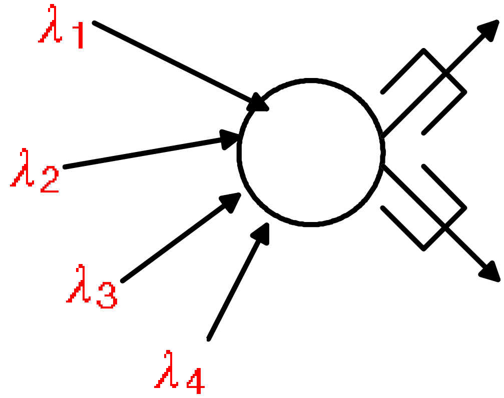


However, in practice:

1) Different sessions take different paths, so departures are not split independently.

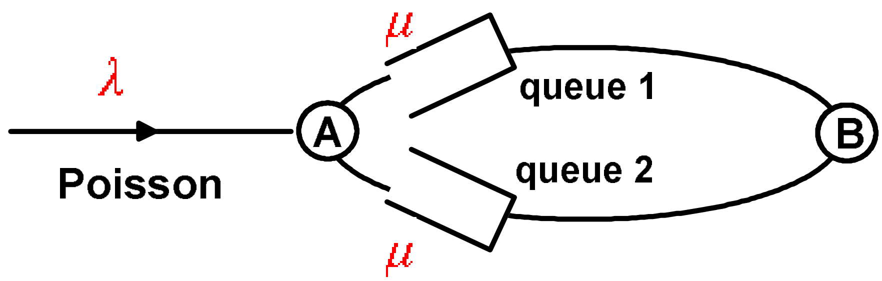
2) service time of a packet depends on its length, which does not change from queue to queue.

Kleinrock's assumption tends to be fairly good in dense networks, with queues receiving traffic from many other queues, and where no metering is used.



Then correlation between interarrival times and packet lengths decreases.

Example of metering



Randomization: flip a coin to assign a packet to a queue. Then we have 2 independent *M/M/1* queues:

$$T_R = \frac{1}{\mu - \frac{\lambda}{2}} = \frac{2}{2\mu - \lambda}$$

(consistent with Kleirock's Approximation)

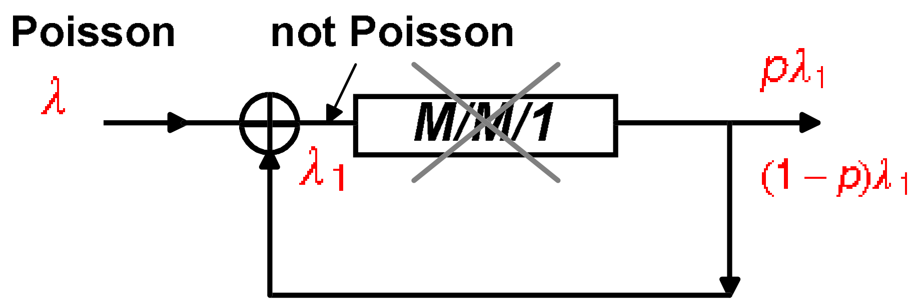
Metering: assign packet to queue that currently has smallest total backlog (and will therefore empty first)

This is equivalent to an *M/M/2* queue

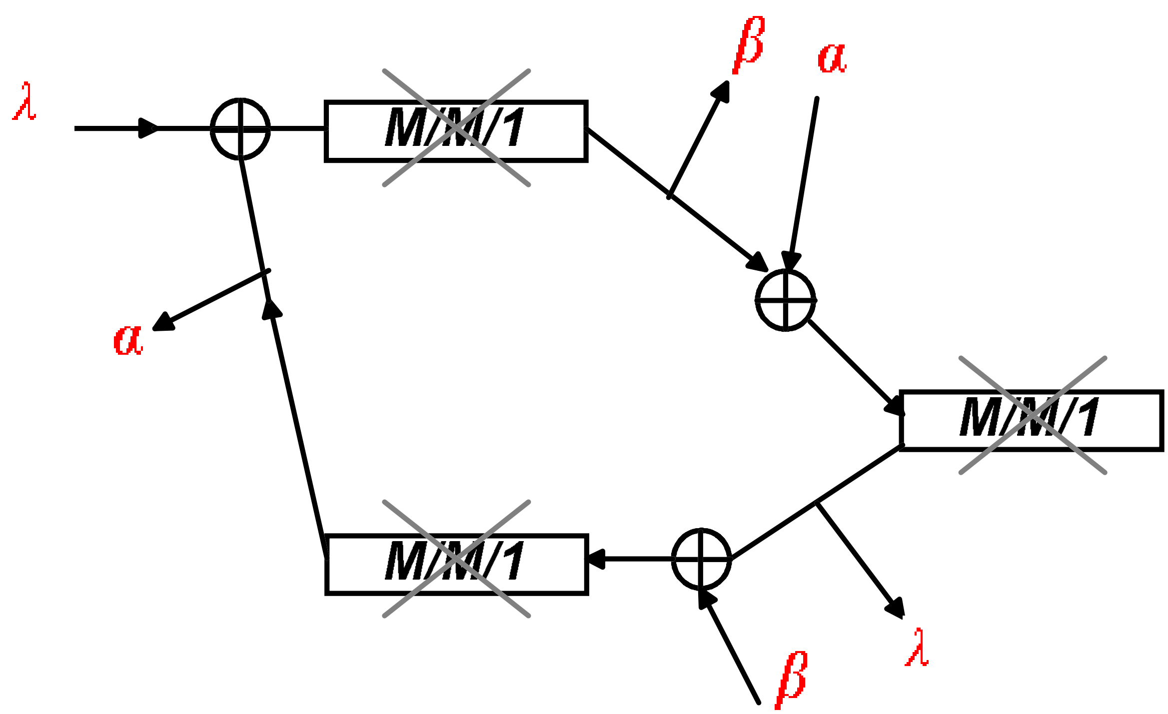
$$T_M = \frac{2}{(2\mu - \lambda)(1+\rho)}, \quad \rho = \frac{\lambda}{2\mu}$$

Metering is popular in datagram nets, and it tends to degrade the accuracy of Kleinrock's approximation.

Examples of feedback:



$$\lambda_1 = (1-p)\lambda_1 + \lambda \Rightarrow \lambda_1 = \frac{\lambda}{p}$$

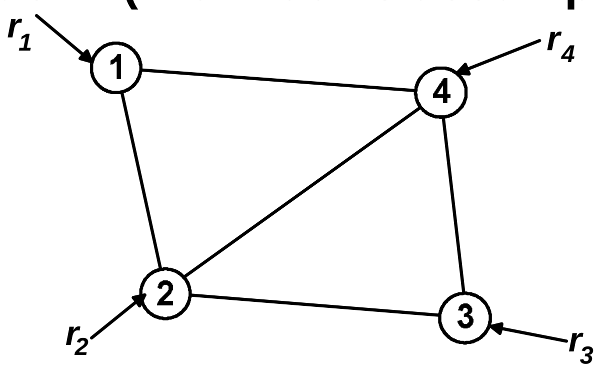


For systems with feedback, arrivals at a queue are not Poisson.

Jackson Networks

A Jackson networks is a network with K **queue,** $j=1, 2, ..., K$

- **Independent exogenous Poisson inputs**
- **Exponentially distributed services**
- **independent routing of packets**
- **service times of a packet at different nodes are independent. (Kleinrock's assumption)**



r_j **= exogenous input rate at node** j

μ_j **= service rate at queue** j

P_{ij} **= probability that a packet leaving node** i **goes to node** j

(with probability $1-\sum_{j=1}^{K} P_{i,j}$ **it leaves the network)**

λ_j **= combined rate at node** j $(< \mu_j)$

$$\lambda_j = r_j + \sum_{i=1}^{K} \lambda_k P_{ij}$$

[λ_j 's **are uniquely determined from** r_j 's **and** P_{ij} 's]

n_j **= # of customers at queue** j

$\overline{n} = (n_1, n_2, ..., n_k)$: **combined state of the system**

<u>Jackson's Thm</u>:

$$P(\overline{n}) = \prod_{j=1}^{k} P_j(n_j) \text{ , } P_j(n_j) = (1-\rho_j)\rho_j^{n_j} \text{ , } \rho_j = \frac{\lambda_j}{\mu_j}$$

i.e.,

(a) the state of queue *j* (at a given time in steady state) is independent of the states of all other queue at that given time

(b) the state of a queue is given by the *M/M/1* formula

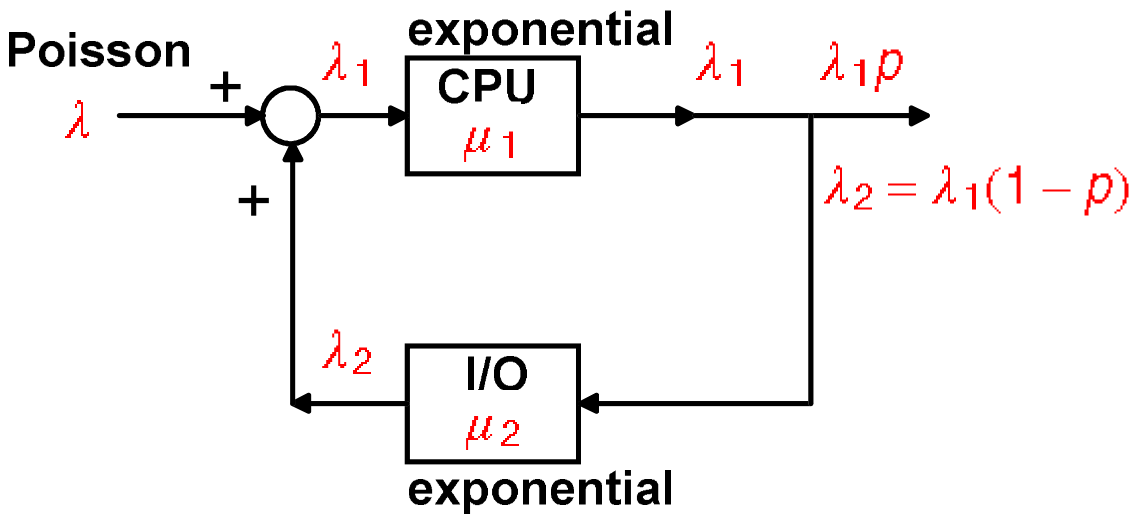
Jackson's theorem is not valid for data networks because of

(1) The session routing

(2) The fact that the service time of a packet at a queue depends on its length, which does not change from queue to queue

It is a reasonable approximation if queue receives from many streams.

Example: Computer system with feedback loop for I/O



$$\lambda_1 = \frac{\lambda}{p}\ ,\ \rho_1 = \frac{\lambda}{p\mu_1}$$

$$\lambda_2 = \frac{\lambda(1-p)}{p}\ ,\ \rho_2 = \frac{\lambda(1-p)}{p\mu_2}$$

$P(n_1, n_2) = (1-\rho_1)\rho_1^{n_1}(1-\rho_2)\rho_2^{n_2} \leftarrow$ **from Jackson's Thr**

n_1**: job in CPU,** n_2**: job in I/O**

$$N_1 = \frac{\rho_1}{1-\rho_1}\ ,\ N_2 = \frac{\rho_2}{1-\rho_2}$$

$$N = N_1 + N_2 = \frac{\rho_1}{1-\rho_1} + \frac{\rho_2}{1-\rho_2}\ ,\ T = \frac{N}{\lambda} = \frac{\rho_1}{\lambda(1-\rho_1)} + \frac{\rho_2}{\lambda(1-\rho_2)}$$

Note: neither the CPU nor the I/O system is actually *M/M/1*