Advanced Computer Networks

Dr Sudipta Saha
Associate Professor
Dept of Computer Science & Engineering
Indian Institute of Technology Bhubaneswar

Birth and Death Markov Process

DSSRG: Decentralized Smart Systems Research Group

https://sites.google.com/iitbbs.ac.in/dssrg
Or Google dssrg iitbbs

The *Birth* and *Death* Markov processes

- **Definition:**
- A **birth/death Markov process** is a **Markov process** where:

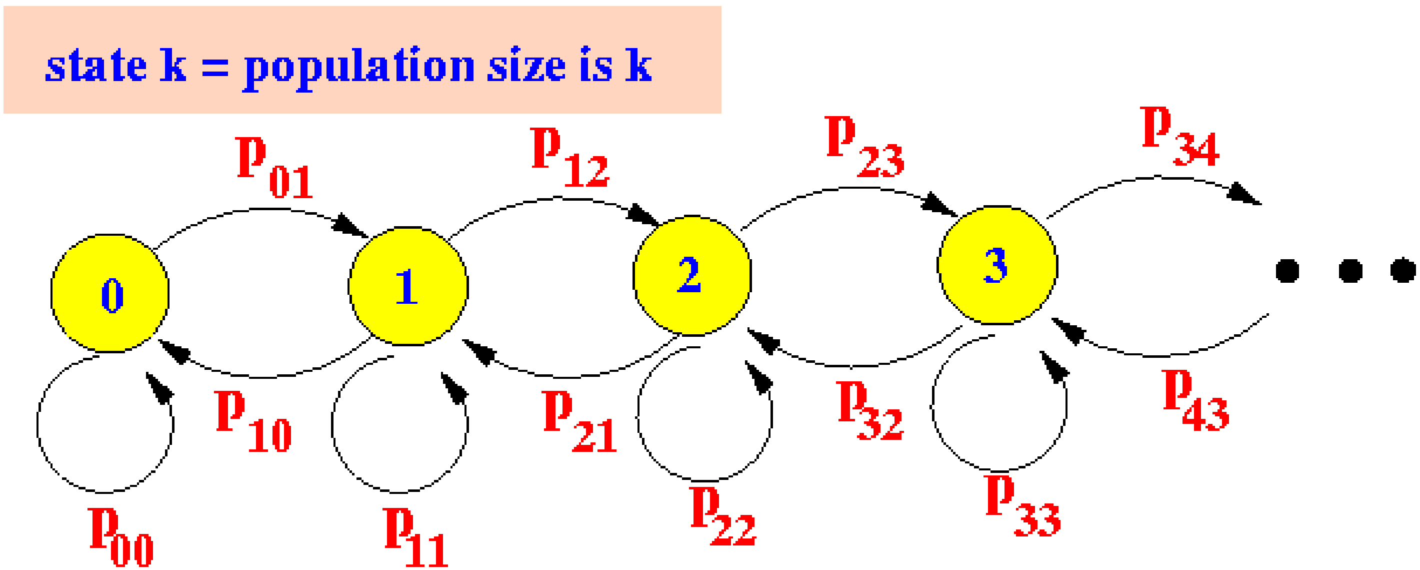
$$p_{i,i+1} = b_i \qquad \text{(birth)}$$

$$p_{i,i-1} = d_i \qquad \text{(death)}$$

$$p_{i,i} = 1 - b_i - d_i$$

$$p_{i,k} = 0 \qquad \text{for } k \le i-2 \text{ or } k \ge i+2$$

In other words

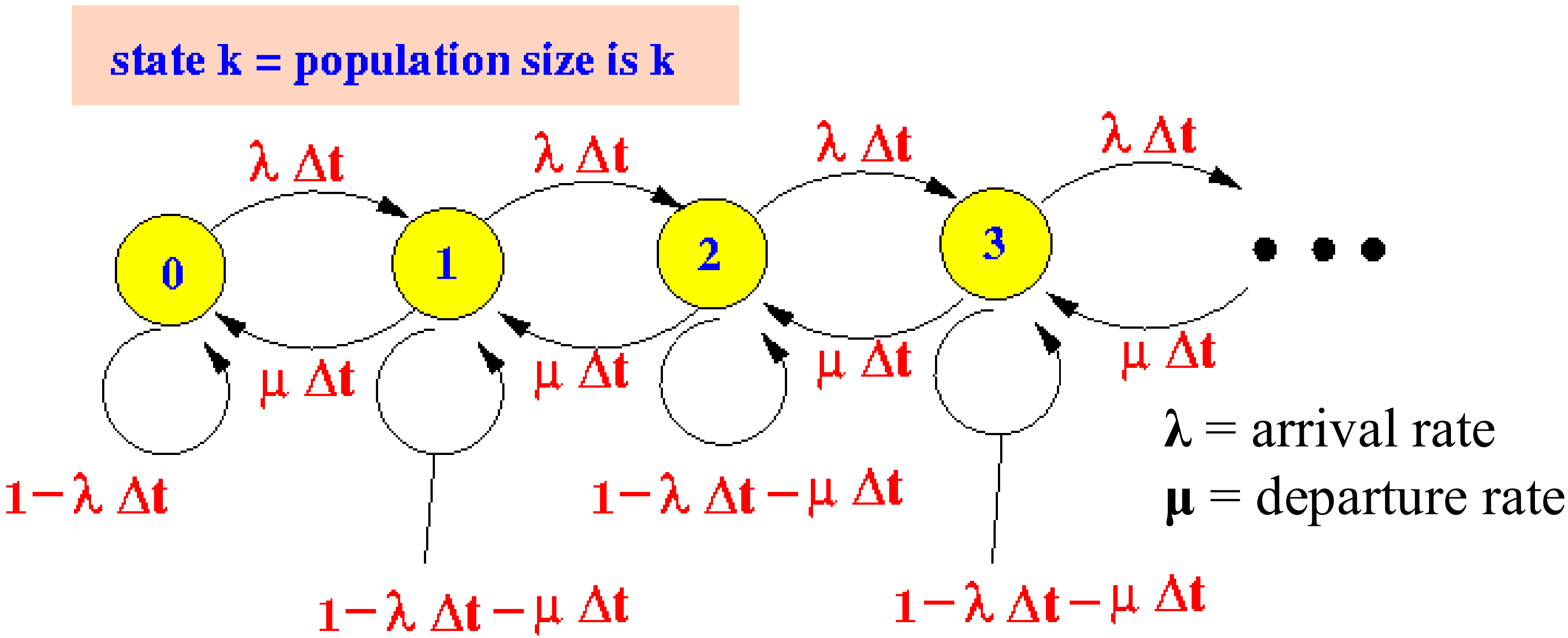
- A process can either go to i-1 or i+1
- All other transition probabilities are zero
- The **transition diagram** of a **birth/death process** looks like this:
- 

The *possible* **events** in a **birth/death Markov process** are:

- **Exactly one birth**
- **Exactly one death**
- The event that there is **no birth and no death** is the case that **no event occurs**

The *Poisson* Birth/Death process

- P[an *arrival* occurs in time interval Δt] = $\lambda \times \Delta t$
- P[a *departure* occurs in time interval Δt] = $\mu \times \Delta t$



Resulting state equations:

$$p_0 = p_0\,(1 - \lambda\Delta t) + p_1\,\mu\Delta t$$
$$p_1 = p_0\,\lambda\Delta t + p_1\,(1 - \lambda\Delta t - \mu\Delta t) + p_2\,\mu\Delta t$$
$$p_2 = p_1\,\lambda\Delta t + p_2\,(1 - \lambda\Delta t - \mu\Delta t) + p_3\,\mu\Delta t$$
$$p_3 = p_2\,\lambda\Delta t + p_3\,(1 - \lambda\Delta t - \mu\Delta t) + p_4\,\mu\Delta t$$
$$\ldots$$

Or:

$$p_0\,\lambda\Delta t = p_1\,\mu\Delta t$$
$$p_1(\lambda\Delta t + \mu\Delta t) = p_0\,\lambda\Delta t + p_2\,\mu\Delta t$$
$$p_2(\lambda\Delta t + \mu\Delta t) = p_1\,\lambda\Delta t + p_3\,\mu\Delta t$$
$$p_3(\lambda\Delta t + \mu\Delta t) = p_2\,\lambda\Delta t + p_4\,\mu\Delta t$$
$$\ldots$$

Or:

$$p_0 \lambda = p_1 \mu$$
$$p_1(\lambda + \mu) = p_0 \lambda + p_2 \mu$$
$$p_2(\lambda + \mu) = p_1 \lambda + p_3 \mu$$
$$p_3(\lambda + \mu) = p_2 \lambda + p_4 \mu$$

...

Or:

$$p_0 \lambda = p_1 \mu$$
$$p_1 \lambda + p_1 \mu = p_0 \lambda + p_2 \mu$$
$$p_2 \lambda + p_2 \mu = p_1 \lambda + p_3 \mu$$
$$p_3 \lambda + p_3 \mu = p_2 \lambda + p_4 \mu$$

...

Or:

$$p_0 \lambda = p_1 \mu$$
$$p_1 \lambda = p_2 \mu$$
$$p_2 \lambda + p_2 \mu = p_1 \lambda + p_3 \mu$$
$$p_3 \lambda + p_3 \mu = p_2 \lambda + p_4 \mu$$

...

.....

Or:

$$p_0 \lambda = p_1 \mu$$
$$p_1 \lambda = p_2 \mu$$
$$p_2 \lambda = p_3 \mu$$
$$p_3 \lambda + p_3 \mu = p_2 \lambda + p_4 \mu$$

...

Final set of equations …

- $$p_0 \lambda = p_1 \mu$$
  $$p_1 \lambda = p_2 \mu$$
  $$p_2 \lambda = p_3 \mu$$
  $$p_3 \lambda = p_4 \mu$$
  ...

Or:

$$p_1 = \lambda/\mu \times p_0$$
$$p_2 = \lambda/\mu \times p_1$$
$$p_3 = \lambda/\mu \times p_2$$
$$p_4 = \lambda/\mu \times p_3$$
...

```
Or:

  p1 = λ/μ × p0                    .... (1)
  p2 = (λ/μ)^2 × p0                .... (2)
  p3 = (λ/μ)^3 × p0                .... (3)
  p4 = (λ/μ)^4 × p0                .... (4)
  ...
```

- We can **express** every $\mathbf{p_i}$, **i = 1, 2, 3, ...** in terms on $\mathbf{p_0}$
- **Unfortunately**, we **do not (yet)** know the value of $\mathbf{p_0}$...

- We need **one more equation** to solve this system, which is:

$$p_0 + p_1 + p_2 + p_3 + \ldots = 1$$

- Substituting $\mathbf{p_i}$ in

$$p_0 + (\lambda/\mu)^1 \times p_0 + (\lambda/\mu)^2 \times p_0 + (\lambda/\mu)^3 \times p_0 + \ldots. = 1$$

$$p_0 \times \left(1 + (\lambda/\mu)^1 + (\lambda/\mu)^2 + (\lambda/\mu)^3 + \ldots \right) = 1$$

$$\begin{array}{rcl}
S & = & 1 + x^1 + x^2 + x^3 + x^4 + \ldots \\
xS & = & \phantom{1 +{}} x^1 + x^2 + x^3 + x^4 + \ldots \qquad - \\
\hline
(1-x)S & = & 1
\end{array}$$

Therefore:

$$1 + x^1 + x^2 + x^3 + x^4 + \ldots = \frac{1}{1 - x}$$

$$p_0 \times \frac{1}{1-(\lambda/\mu)} = 1$$

$$p_0 = 1 - (\lambda/\mu) = 1 - \rho$$

$$\rho = \lambda/\mu$$

- **Steady state probability distribution of a Poisson birth/death process with arrival rate λ and departure rate μ:**
- $$p_0 = (\lambda/\mu)^0 \times (1-(\lambda/\mu))$$
  $$p_1 = (\lambda/\mu)^1 \times (1-(\lambda/\mu))$$
  $$p_2 = (\lambda/\mu)^2 \times (1-(\lambda/\mu))$$
  $$p_3 = (\lambda/\mu)^3 \times (1-(\lambda/\mu))$$
  ....

Or:

$$p_0 = \rho^0 \times (1 - \rho)$$
$$p_1 = \rho^1 \times (1 - \rho)$$
$$p_2 = \rho^2 \times (1 - \rho)$$
$$p_3 = \rho^3 \times (1 - \rho)$$

....

$$P[\text{ k customers in system }] = p_k = \rho^k \times (1 - \rho)$$

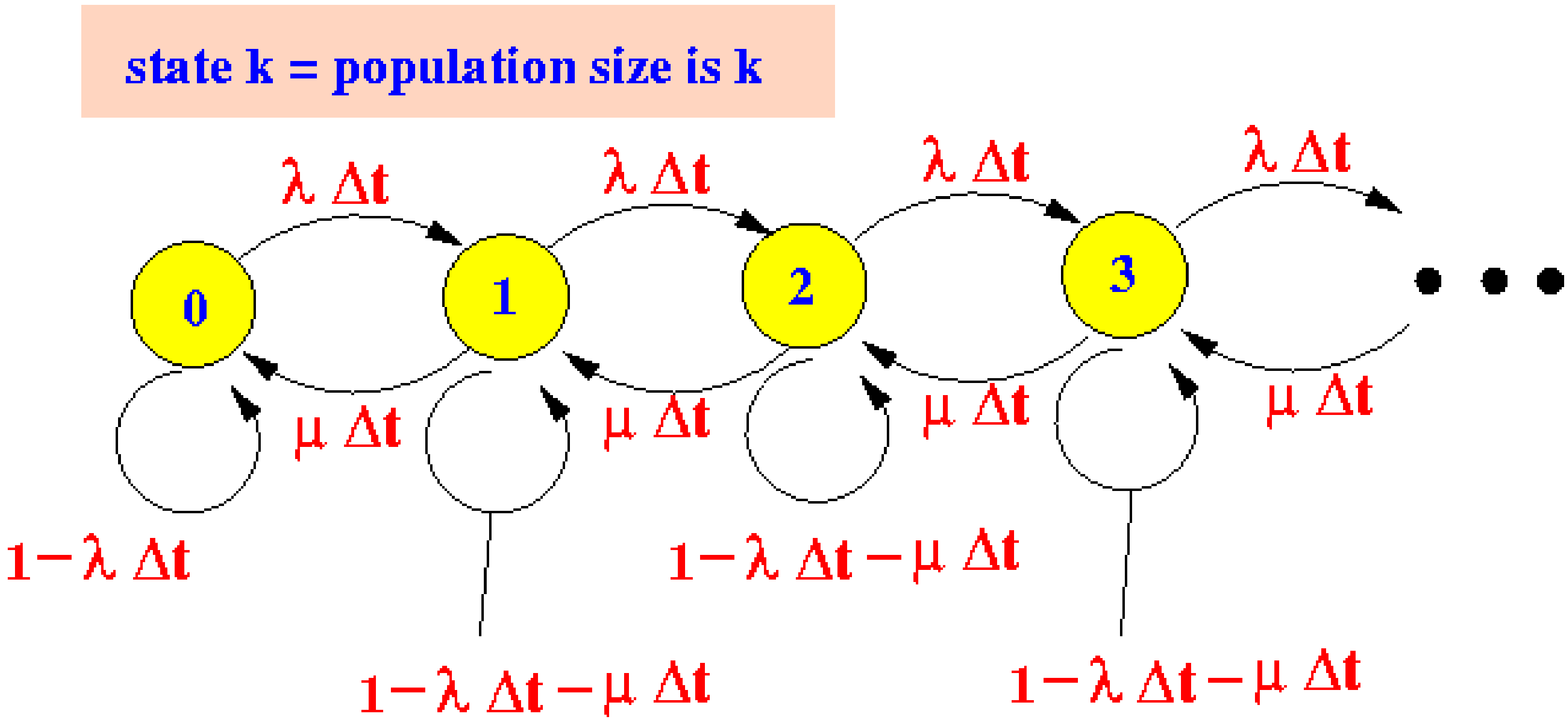
Solving Markov chain using *rate transition* diagram

- A **popular method** for finding the **equilibrium (steady state) probability distribution** of a **Markov chain** is using **rate transition diagrams**

- **Rate transition diagram:**

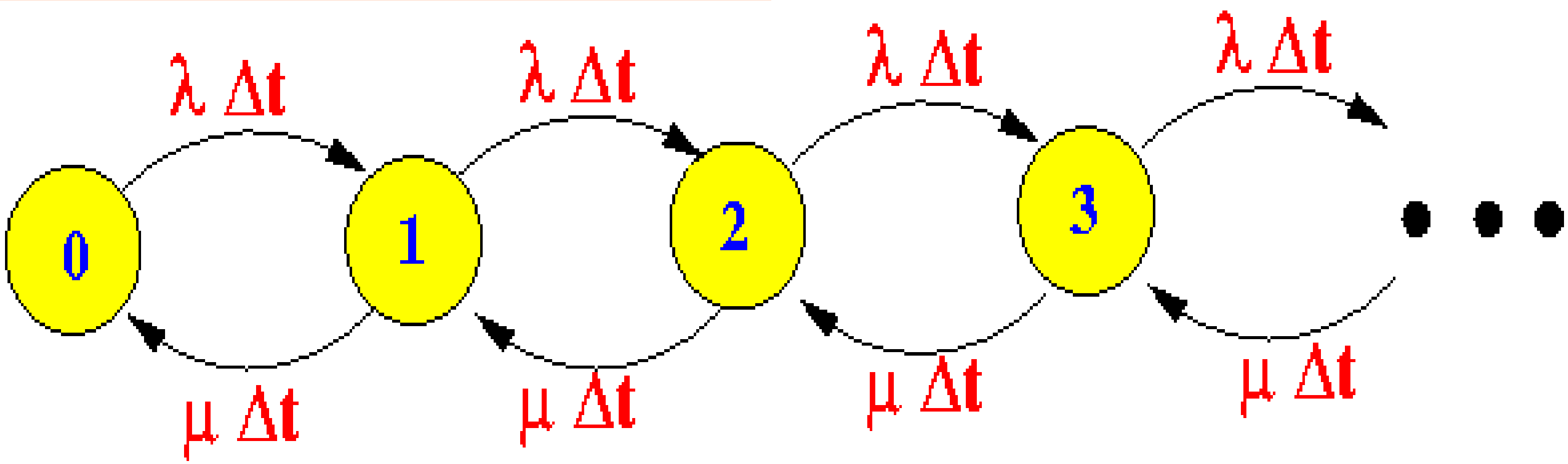
- A **rate transition diagram** is obtained by removing the **transitions** that goes from a **state** into the **same state** from a **state transition diagram**

State transition diagram:

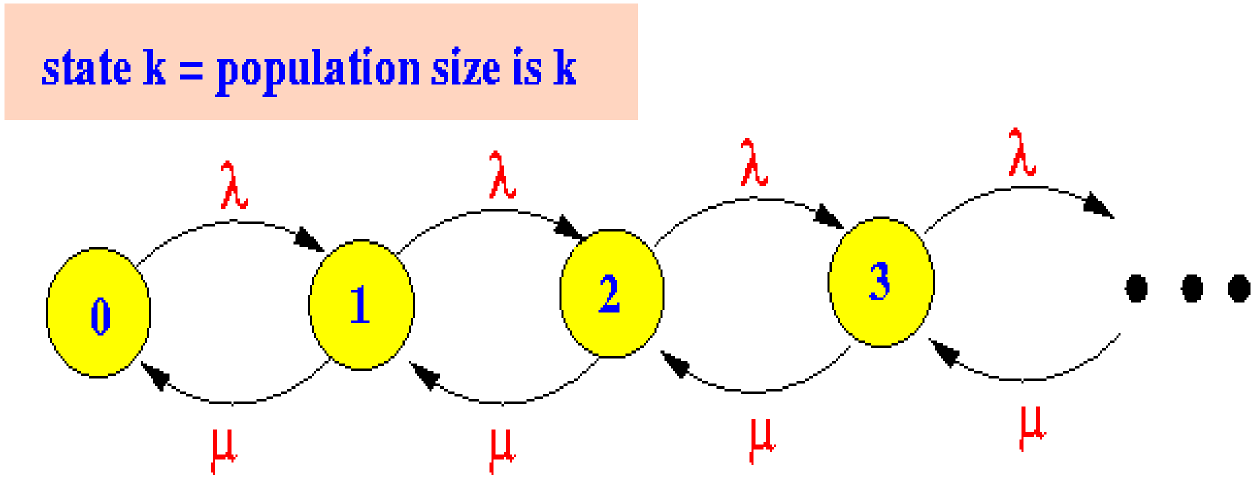


State diagram *without* loops to itself:

state k = population size is k



After *normalization* by dividing by Δt:

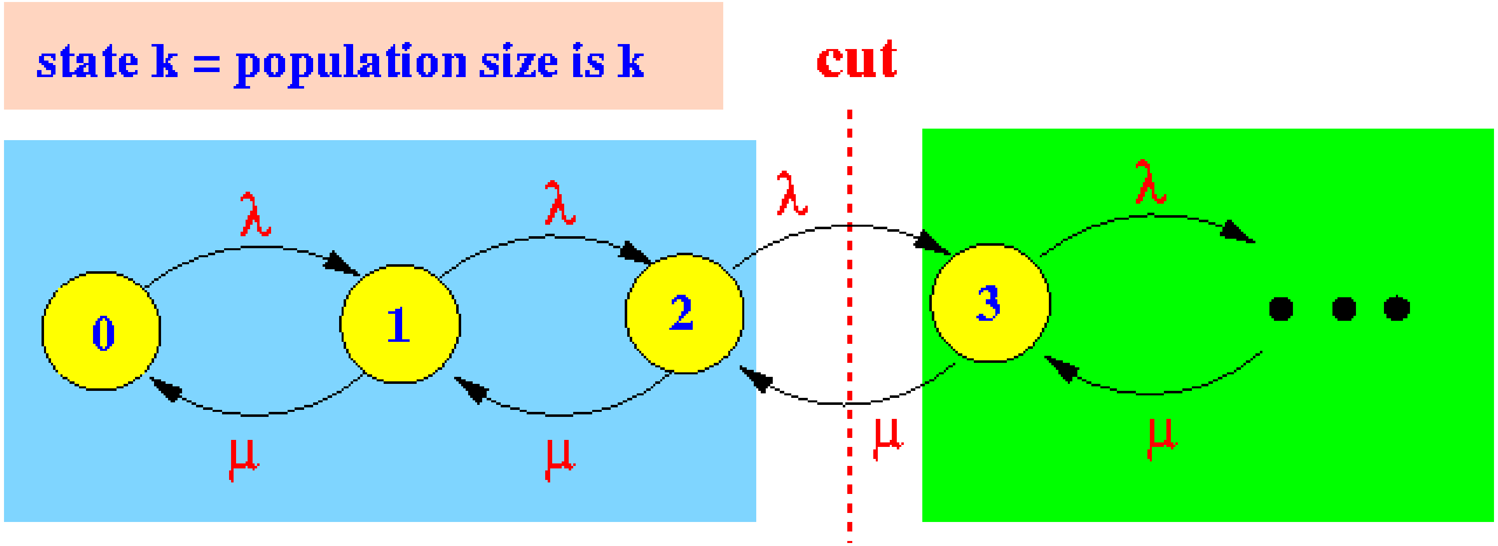


- λ = **arrival** *rate* of clients
- μ = **departure** *rate* of clients

- The **weights** on the **arcs** in the **diagram** are the **rates** of **arrival** and **departure**
- Hence the **name**: ***rate* transition diagram**

Setting up equilibrium equations using a *rate* transition diagram

- Find a **cut** in the **Markov chain** that divide the **Markov chain** into **2 disjoint pieces**
- In the **equilibrium state**, the **number of transitions** from **one side** the **cut** to the **other side** must be **equal** to the **reverse direction**
-

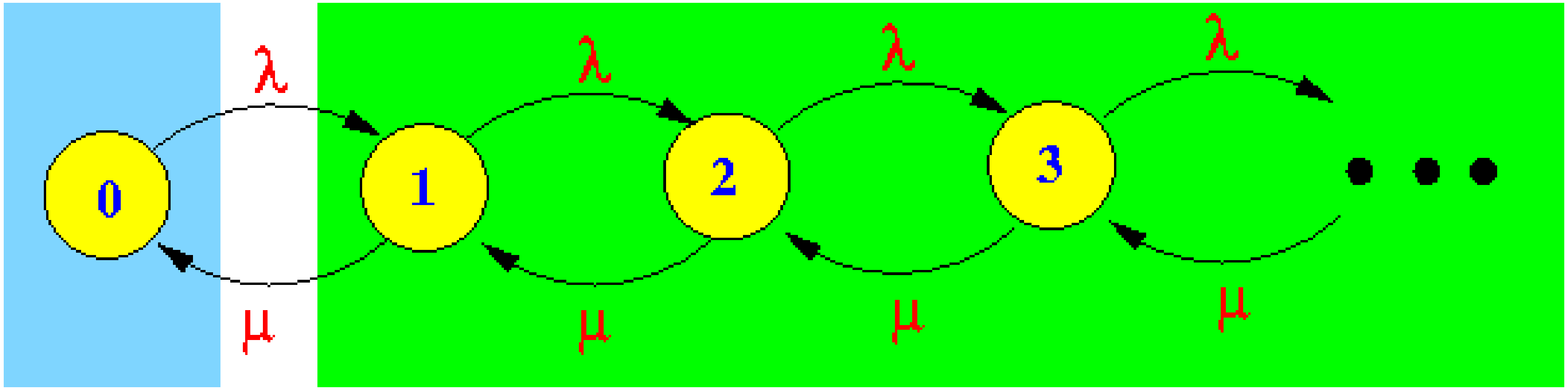


Equilibrium equation:

- Flow from left to right: $p_2 \times \lambda$
- Flow from left to right: $p_3 \times \mu$
- Equilibrium: $p_2 \times \lambda = p_3 \times \mu$
-

Complete example: equilibrium equations for the M/M/1 queuing system

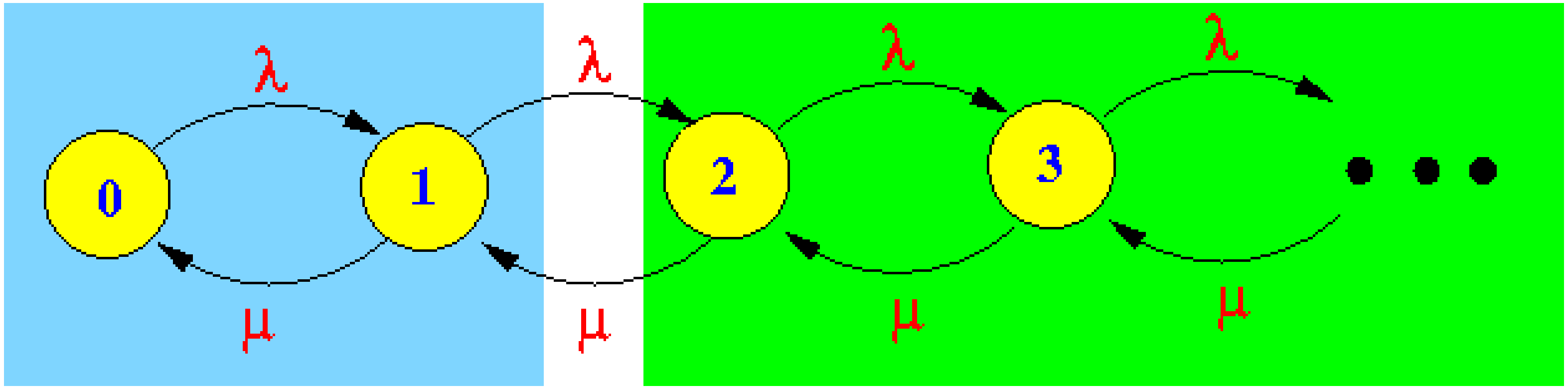
Cut 1:



Equilibrium equation:

Flow from left to right: $\lambda \times p_0$ Flow from right to left: $\mu \times p_1$ Equilibrium: $\lambda \times p_0 = \mu \times p_1$

- **Cut 2:**



Equilibrium equation:

Flow from left to right: $\lambda \times p_1$

Flow from right to left: $\mu \times p_2$

Equilibrium: $\lambda \times p_1 = \mu \times p_2$

- **Resulting set of equation for the equilibrium state:**
- $p_0 \lambda = p_1 \mu$
- $p_1 \lambda = p_2 \mu$
- $p_2 \lambda = p_3 \mu$
- $p_3 \lambda = p_4 \mu$

- This is the **same set** of equation as before...

Introduction to queueing theory

- **Queue**
- A **queue** is a **waiting line**...
- The **behaviour** of a **queue** is **characterized** by following parameters:

- the **arrival process** (commonly used: **Poisson process**)
- the **service (departure) process** (commonly used: **Poisson process**)
- the **number of servers** in the **system**
- The **queueing discipline** (often **FIFO**)
- The **capacity** of the **queue** (buffer space)
- The **size of the client population** (commonly used value: **infinite**)

Kendall notation for a queueing system

- **Kendall's notation** (or sometimes **Kendall notation**) the **standard system** used to **describe and classify** the **queueing model** that a queueing system corresponds to.
- The notation was first suggested by

  D. G. Kendall in **1953**
-

Long Kendall notation:

```
A   |   B   |   c   |   D   |   E   |   F
^       ^       ^       ^       ^       ^
|       |       |       |       |       |
|       |       |       |       |       +--- queueing discipline (default: FIFO)
|       |       |       |       +--- population size (default: infinite)
|       |       |       +--- queue capacity (default: infinite)
|       |       +--- number of servers
|       +--- server process
+--- arrival process
```

Abbreviated Kendall notation:

```
A   |   B   |   c
^       ^       ^       ^       ^       ^

|       |       |       |       |       |
|       |       |       |       |       +--- queueing discipline is FIFO
|       |       |       |       +--- population size is infinite
|       |       |       +--- queue capacity is infinite
|       |       +--- number of servers
|       +--- server process
+--- arrival process
```

Further notations:

- **Abbreviations** for **arrival and service processes**:

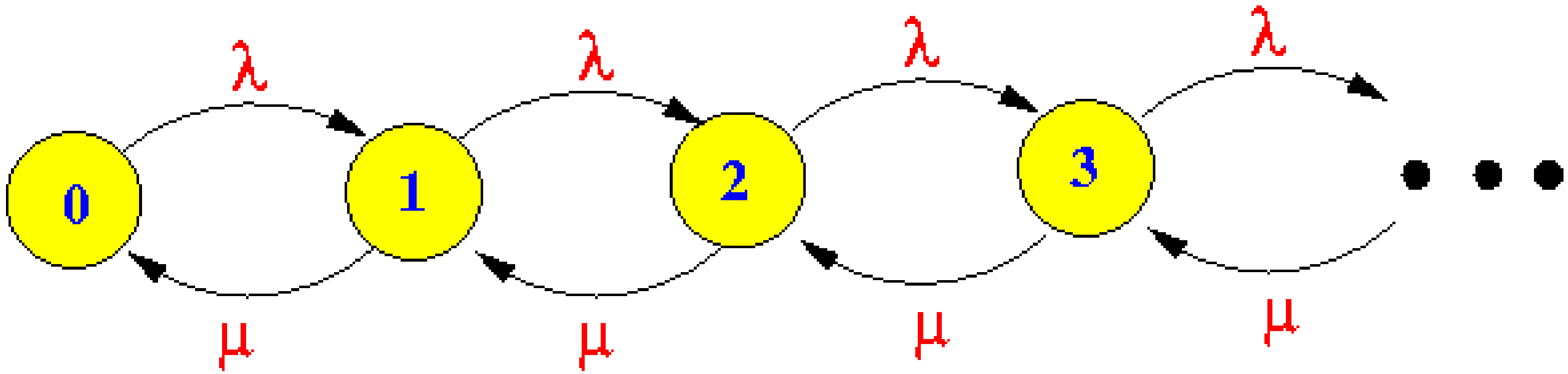
  - **M** = **Poisson process**
  - **D** = **Deterministic process** (fixed time between 2 consecutive events)
  - **G** = **general process**

The *classic* M/M/1 queue

- The **M/M/1** queue is a **short hand** notation for the **M/M/1/∞/∞/FIFO** queue:

- **M** = **arrival process** is **Poisson** (with some parameter **lambda;**)
- **M** = **service (departure) process** is **Poisson** (with some parameter **mu;**)
- **1** = there is **1 server** in system
- ∞ = **infinite queue capacity**; **no arriving client** will be **rejected**
- ∞ = **infinite population size** (the **arrival process** will be **unaffected** by the number of clients already in the system)
- **FIFO** = first in first out service
-

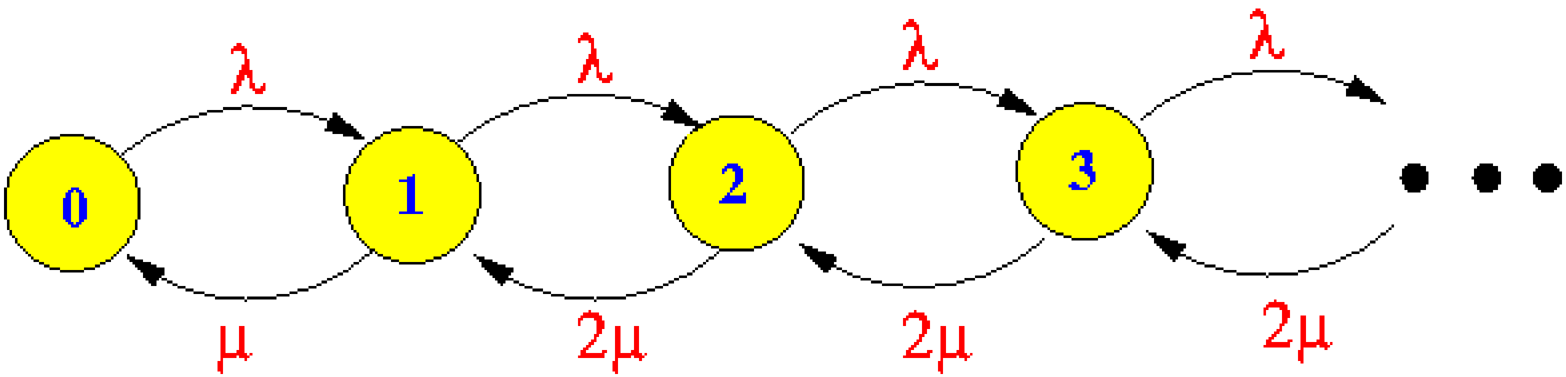
The **rate transition diagram** for the **M/M/1** queueing system is:



•We have already studied this system...

One more servers: the M/M/2 system

The **rate transition diagram** for the **M/M/2** system is:



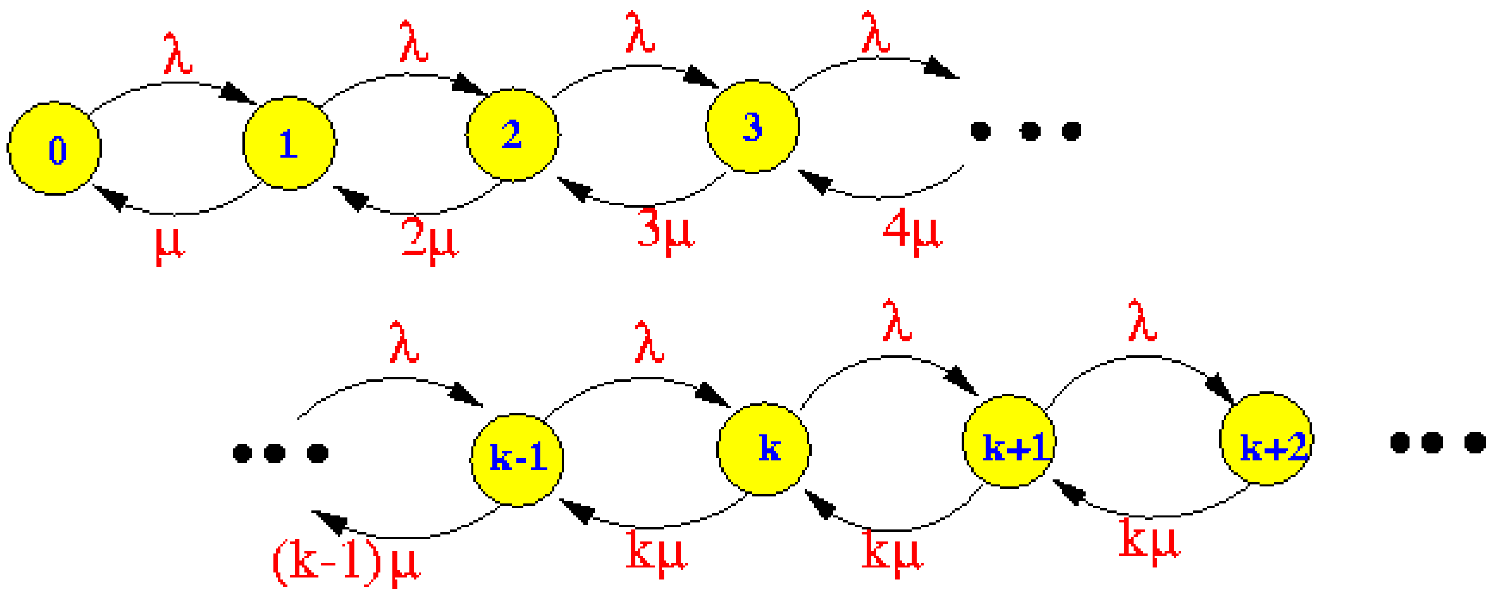
Note: A single service system where the maximum departure rate is now 2mu.

Equilibrium equations for M/M/2:

- $p_0 \times \lambda = p_1 \times \mu$
- $p_1 \times \lambda = p_2 \times 2\mu$
- $p_2 \times \lambda = p_3 \times 2\mu$
- $p_3 \times \lambda = p_4 \times 2\mu$
- ....

More servers in general: M/M/k

- The **rate transition diagram** for the **M/M/k** system is:
-



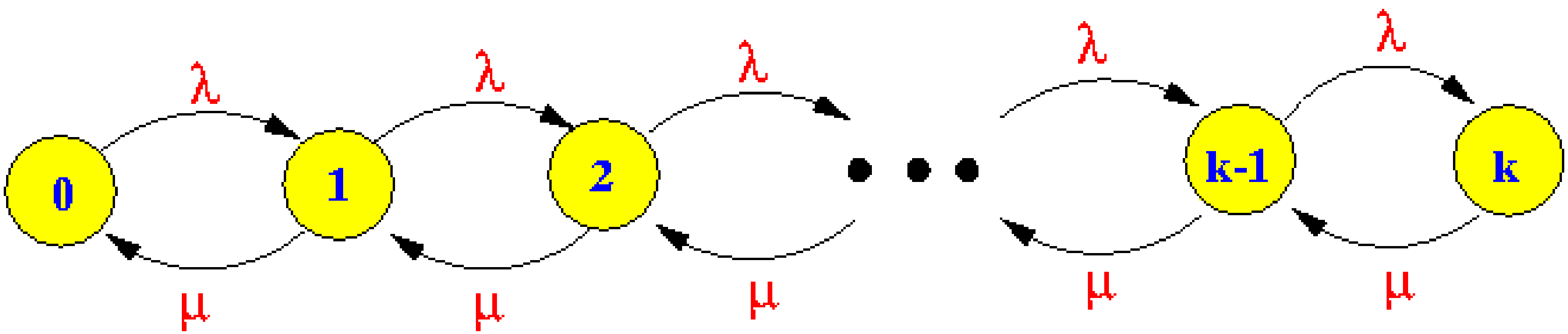
Equilibrium equations for M/M/k:

- $p_0 \times \lambda = p_1 \times \mu$
- $p_1 \times \lambda = p_2 \times 2\mu$
- $p_2 \times \lambda = p_3 \times 3\mu$
- ....
- $p_{k-2} \times \lambda = p_{k-1} \times (k-1)\mu$
- $p_{k-1} \times \lambda = p_k \times k\mu$
- $p_k \times \lambda = p_{k+1} \times k\mu$
- $p_{k+1} \times \lambda = p_{k+2} \times k\mu$
- $p_{k+2} \times \lambda = p_{k+3} \times k\mu$ ....

Finite buffer capacity: M/M/1/k

- The **M/M/1/k** queue is a **short hand** notation for the **M/M/1/k/∞/FIFO** queue:
-
  - **M** = **arrival process** is **Poisson** (with some parameter **lambda;**)
  - **M** = **service (departure) process** is **Poisson** (with some parameter **lambda;**)
  - **1** = there is **1 server** in system
  - **k** = **queue capacity**; **the (k+1)th arriving client** will be ***rejected***
  - **∞** = **infinite population size** (the **arrival process** will be **unaffected** by the number of clients already in the system)
  - **FIFO** = first in first out service

The **rate transition diagram** for the **M/M/1/k** queueing system is:



Notice that:

The **number of customers** in the **system** is **at most k**
Hence, the states of this **Markov chain** are **0, 1, 2, ..., k**

Equilibrium equations for M/M/1/k:

$p_0 \times \lambda = p_1 \times \mu$

$p_1 \times \lambda = p_2 \times \mu$

$p_2 \times \lambda = p_3 \times \mu$

$p_3 \times \lambda = p_4 \times \mu$

….

$p_{k-1} \times \lambda = p_k \times \mu$

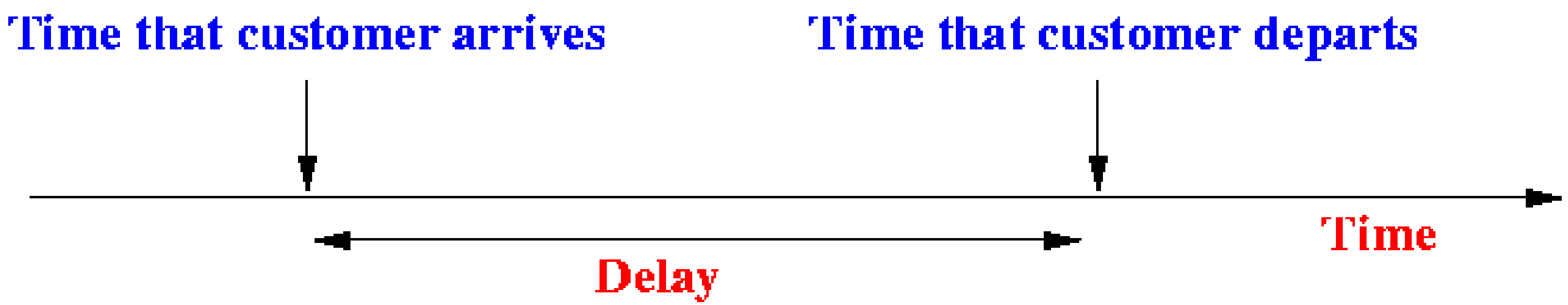
Performance analysis: Average queue length (average number of customers in system)

- **Purpose of queueing models:**
- **Analysis** of **queueing models** can provide **answers to performance related questions** through **mathematical analysis**
- **Commonly used performance measures**

- **Average queue length N**
- This is the **average number of customers** in the system.
- (Better: This is the **average number of customers waiting** in the system to get service....)

Average delay time <u>T</u>

- The **delay** is defined as:



- The **average delay time** is the **average amount of time** that a customer spends in the system.

Computing the *average queue length* (in general)

- Consider the following simple example:
- **50% of the time**, the system is **empty**
- **50% of the time**, the system has **1 customer**
- Then the **average number of customers** in the system is the **weighted sum**
- $\underline{N} = 0.5 \times 0 + 0.5 \times 1 = 0.5$

Average queue length

- **In general**, the **average queue length** (or the **average number of customers in system**) is equal to:

N = Mean (expected) number of customer

= 0 × P[0 customers in system]

\+ 1 × P[1 customer in system]

\+ 2 × P[2 customers in system] + ...

$= \sum_{\{k = 0, 1, .., \infty\}} k \times P[\text{ k customers in system}]$

(Definition of "expected value")

$= \sum_{\{k = 0, 1, .., \infty\}} k \times p_k$ .... (1)

Example: Average queue length in the M/M/1 queue

Recall that the **state probability i** of the **M/M/1** queue is

P[k customers in system] = $\rho^k (1 - \rho)$

- What is the the **average (expected) queue length <u>N</u>** of the **M/M/1** queue ?

- Let us derive this …

Contd…

$$\underline{N} = \sum_{\{k = 0, 1, .., \infty\}} k \times p_k$$

$$= \sum_{\{k = 0, 1, .., \infty\}} k \times \rho^k (1 - \rho)$$

$$= (1 - \rho) \times \sum_{\{k = 0, 1, .., \infty\}} k \times \rho^k \quad .... (2)$$

$$U = \sum_{\{k = 0, 1, .., \infty\}} \rho^k = 1/(1 - \rho)$$

Or:

$$U = \sum_{\{k = 0, 1, .., \infty\}} \rho^k = (1 - \rho)^{-1}$$

...

Thus:

$$\frac{dU}{d\rho} = \sum_{\{k = 0, 1, .., \infty\}} k\, \rho^{k-1} = (1 - \rho)^{-2}$$

Therefore:

$$\sum_{\{k = 0, 1, .., \infty\}} k\, \rho^{k} = \rho \times (1 - \rho)^{-2} \quad \text{.........} (3)$$

Finally…

Substitute (3) in (2):

$$\underline{N} = (1 - \rho) \times \sum_{\{k = 0, 1, .., \infty\}} k \times \rho^k$$

$$= (1 - \rho) \times \rho \times (1 - \rho)^{-2}$$

$$= \rho \times (1 - \rho)^{-1}$$

So...

Therefore, the mean number of customers in an M/M/1 queue is equal to:

$$\underline{N}(M/M/1) = \frac{\rho}{1 - \rho} \qquad \ldots\ldots (4)$$

Performance analysis: Average delay

- **Unlike** ***average queue length***, the **average delay cannot** be **directly derived**

- **[We don't have the distribution]**

- **Fortunately**, there is a **simple indirect way** to compute the **average delay**

- **Little's law (Little's formula)**
- **John Little** proved the following ***famous*** **formula**
- The **long-term average number of customers** in a **stable system is equal to: the long-term average arrival rate multiplied by the long-term average time a customer spends in the system**

In other words:

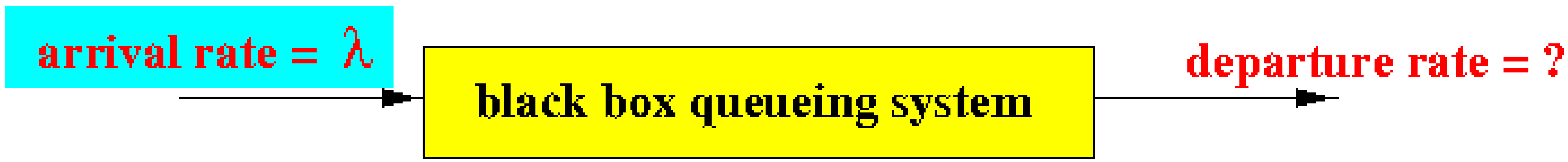
- Avg. # customers in system = λ × Avg. delay of customers

- **More intuitive form -**

$$= \frac{\text{Avg. \# customers in system}}{\text{Avg. delay of customers}} = \lambda \text{ (Little's formula)}$$

Proof:

- Consider a **stable system** that has an **average arrival rate** of λ:



Claim: The **long term departure rate** is **also** equals to **λ**

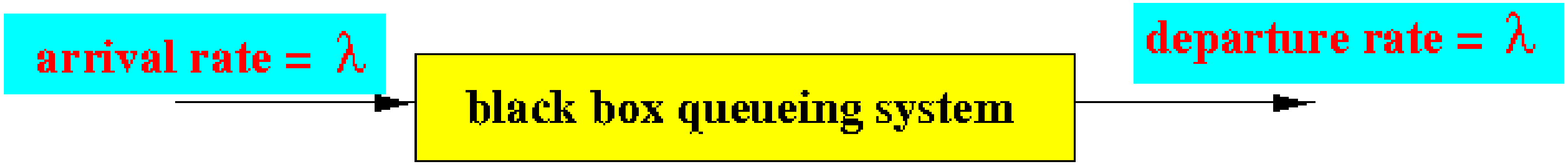
Reason:

If the ***long*** **term departure rate** is **less (<)** the ***long*** **term arrival rate λ**, then the **number of customers** in the **system** will **keep growing to *infinity*** and the **system** will ***not* be stable**

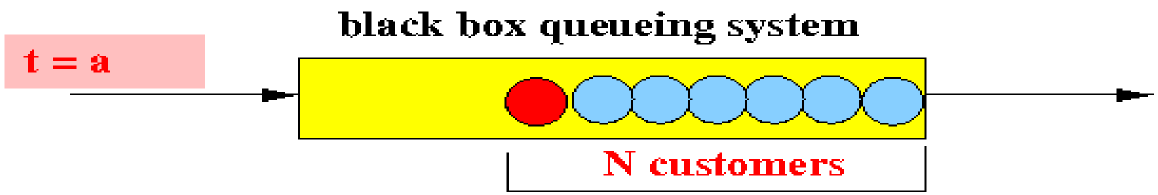
On the other hand, if the ***long*** **term departure rate** is **greater (>)** the ***long*** **term arrival rate λ**, then the **number of customers** in the **system** will **becomes equal to *0***.Such a **system** is ***not* be stable**

So…

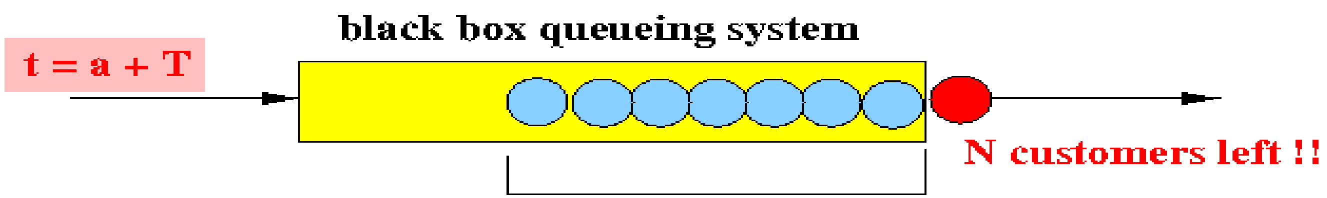
- **Conclusion:** in a ***stable*** **system**, ***long term*** **departure rate** = ***long term*** **arrival rate**



We can **calculate** the *long term* **departure rate** as follows:



When red customer arrives,
he finds an average of N customers
in system (including himself !)



The red customer spends avg. T sec in system

Hence, the *long term* **departure rate** is equal to:

```
                                       Avg. # customers in system
long term departure rate = ---------------------------------------
                                Avg. time a customer spends in system
```

which is the **Little's formula** !!!

Example: Average service time of a customer in the M/M/1 queue

- According to a previous analysis the **average queue length** in the **M/M/1 queueing system** is:
-

$$\underline{N}(M/M/1) = \frac{\rho}{1 - \rho}$$

According to the **Little's formula**:

$\underline{T} = \underline{N} / \lambda$ (Little's formula in another form)

So…

- Therefore, the **average service time** of the **M/M/1** queue is
- $$\underline{T} = \frac{\underline{N}}{\lambda}$$
  $$= \frac{1}{\lambda} * \frac{\rho}{1 - \rho}$$
  $$= \frac{1}{\lambda} * \frac{\lambda/\mu}{1 - \rho}$$
  $$= \frac{1}{\mu} * \frac{1}{1 - \rho} \qquad ..... (3)$$

Other performance measures: Idle time/busy time

How often do you waste system resources:

- **Idle system**: The system is **idle** when there are **zero (0) customers** in the system
- The **fraction of time** that the **system is idle** is equal to the **probability that the system is *empty***
- **Therefore:**
- Fraction of time that system is idle =

$$P[\text{ 0 customer in system }] = p_0$$

The **fraction of time** that an **M/M/1** queue is **idle** is equal to:

- fraction of time that M/M/1 queue is idle = 1 - ρ
- **Busy system: (busy = not idle)**
- The system is **idle** when there are **not zero (0) customers** in the system
- The **fraction of time** that the **system is idle** is equal to the **probability that the system is *not* empty**
- **Therefore:** fraction of time that system is idle = 1 - P[0 customer in system] = 1 - p_0

So…

- The **fraction of time** that an **M/M/1** queue is **busy** is equal to:

- **fraction of time that M/M/1 queue is idle = ρ**

Other performance measure: Loaded system

- The **M/M/*/k** queueing model is used to model system with **finite waiting capacity** (the **maximum number of customers in the system** is **at most k**)

- We can ask the following **capacity question** on system with **finite waiting capacity**:

- **How often** is the system running **at system capacity** i.e., **how often** is the **number of customers** in the system **equal to its *maximum***

- The **fraction of time** that an **M/M/*/k** queueing system is running at **maximum capacity** can be computed as: fraction of time that M/M/*/k system is at max. cap.

  = P[k customers in system] = p_k

Summary: Visualizing the relations

Inputs:

Arrival Rate
Departure Rate
No of Servers
Queue Capacity
…

Outputs:

Probability of staying at a state (Steady state probability)
Average length of the queue
Average delay of a customer
Idle time
Busy time
…