# ERP 420

## Research Project

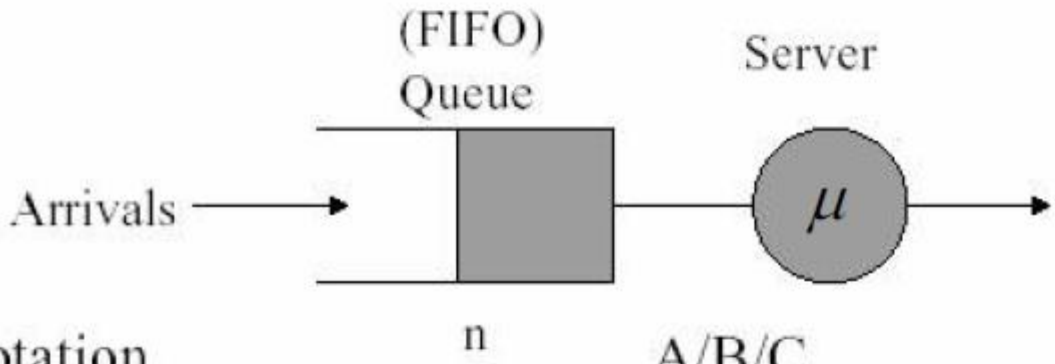
## Single Station Queuing Systems II


Dr A. De Freitas

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

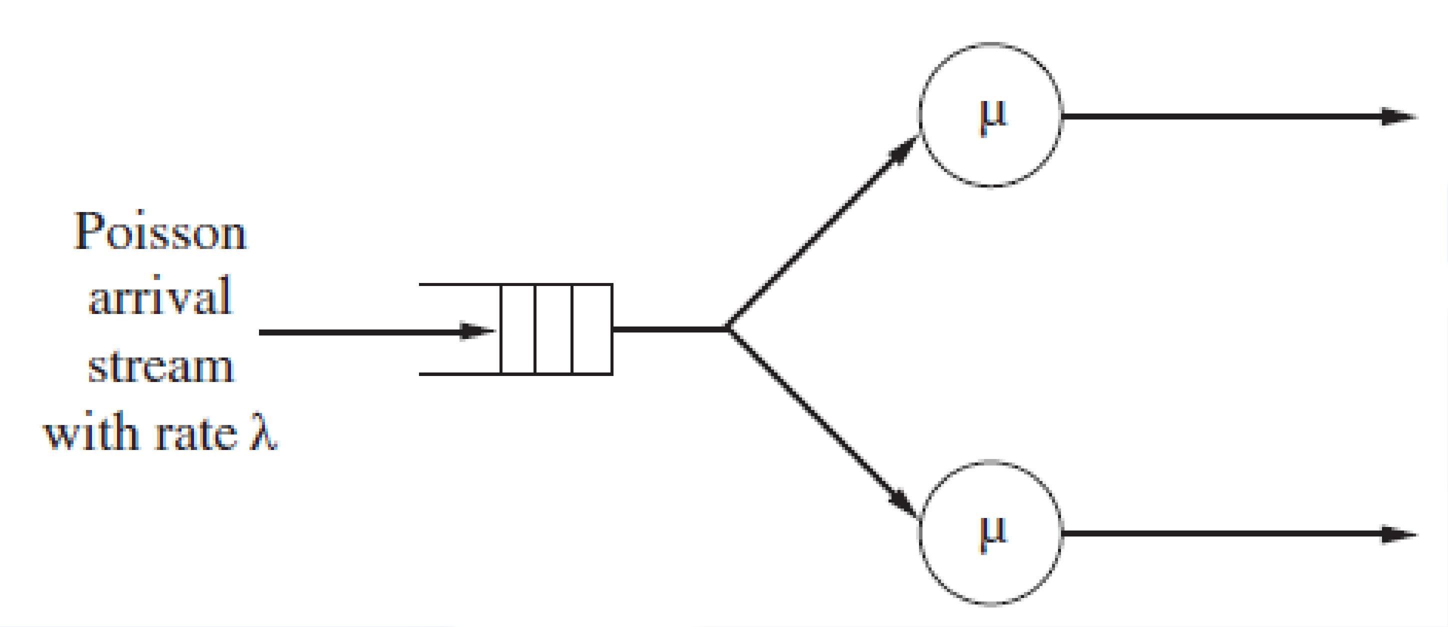
# Markov Queues – M/M/1

## M/M/1 Queue

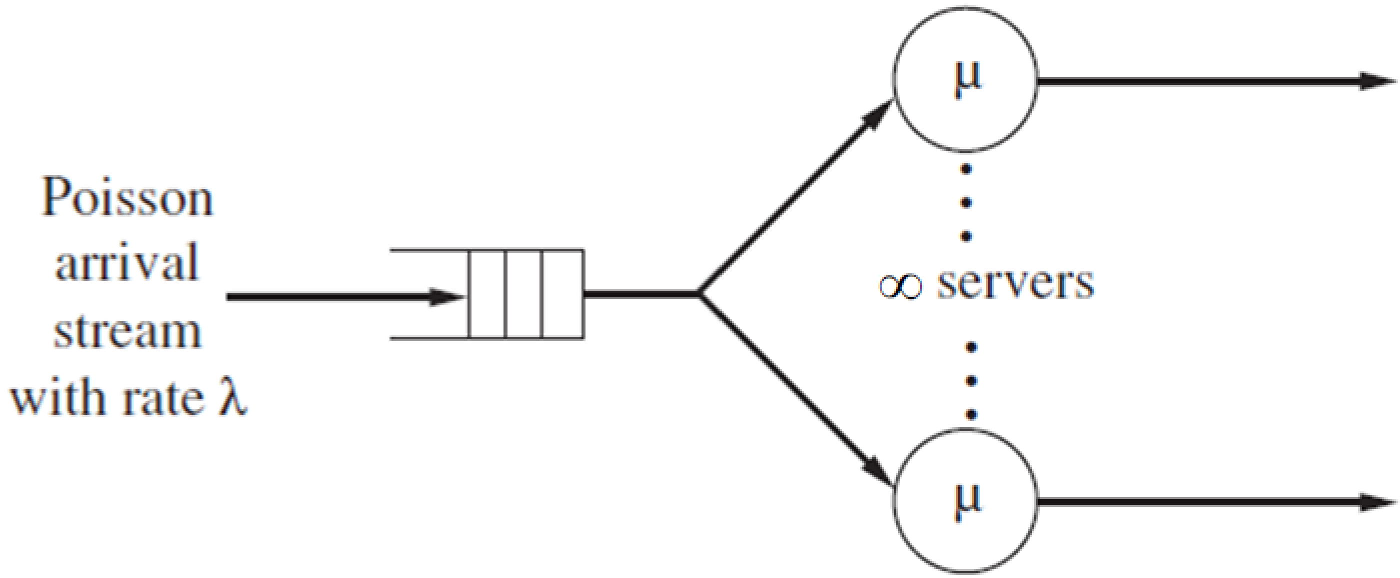



- Notation — A/B/C
  - Arrivals process → A
  - Service Distribution → B
  - Number of Servers → C
- M ≡ Markov process. Transition probabilities depend only on current state. Memoryless.
- Poisson/Exponential is special case of Markov.

# Markov Queues – M/M/2

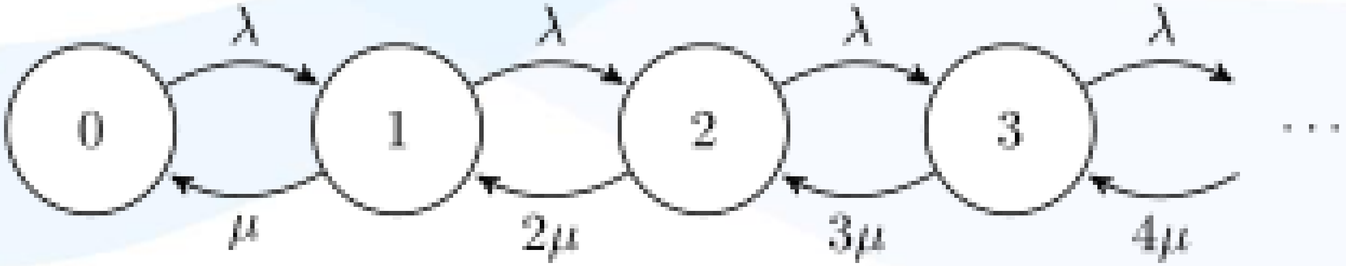


# Markov Queues – M/M/∞



# Markov Queues – M/M/∞

In an M/M/∞ we have a Poisson arrival process with arrival rate $\lambda$ and an infinite number of servers with service rate $\mu$ each. If there are $k$ jobs in the system, then the overall service rate is $\boldsymbol{k}\mu$ because each arriving job immediately gets a server and does not have to wait. Once again, the underlying CTMC is a birth-death process.




UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

# Markov Queues – M/M/∞

- we obtain the steady-state probability of $k$ jobs in the system:

$$\pi_k = \pi_0 \cdot \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}, k \geq 1 \quad (3-11) \qquad \pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \pi_0 (\frac{\lambda}{\mu})^k \frac{1}{k!}$$

- we obtain the steady-state probability of no jobs in the system:

$$\pi_0 = \frac{1}{1 + \sum_{k=1}^{\infty} (\frac{\lambda}{\mu})^k \frac{1}{k!}} = e^{-\frac{\lambda}{\mu}}$$

- And finally

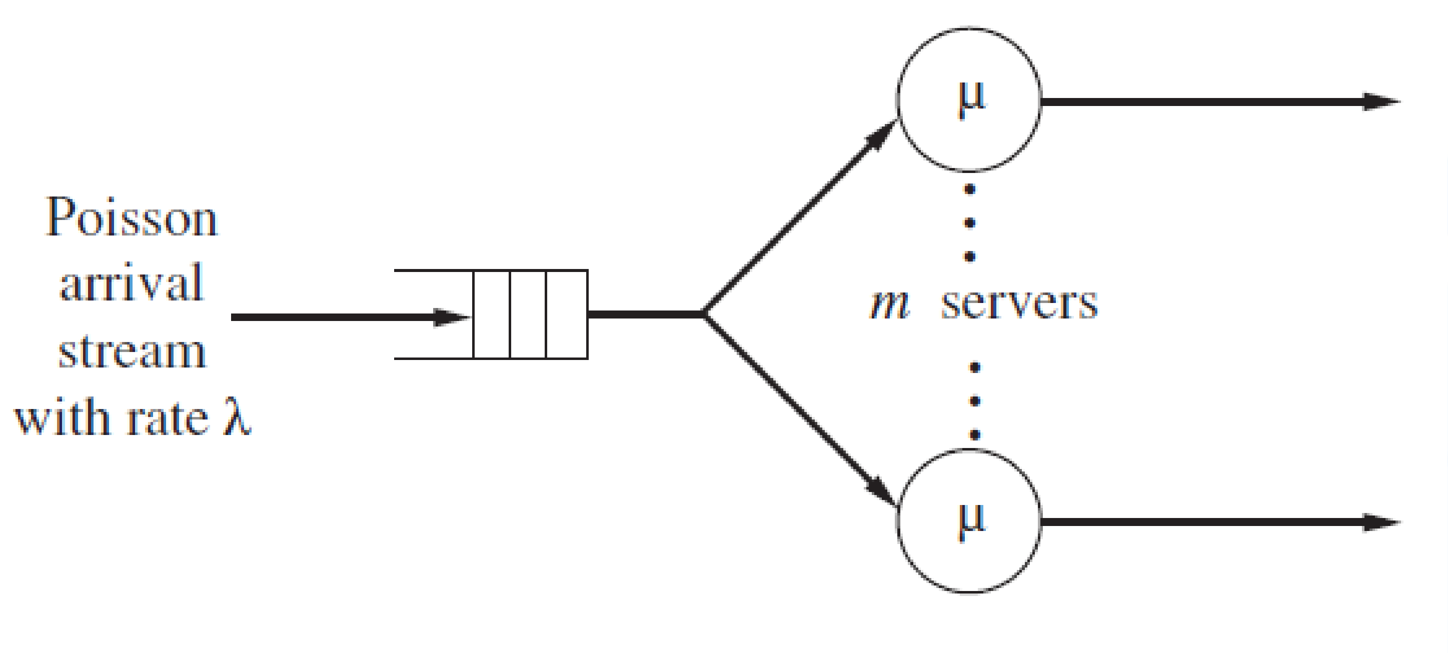
$$\pi_k = \frac{(\frac{\lambda}{\mu})^k}{k!} . e^{-\frac{\lambda}{\mu}}$$

- This is the Poisson pmf(probability function), and the expected number of jobs in the system is: $\overline{K} = \frac{\lambda}{\mu}$

# Markov Queues – M/M/∞

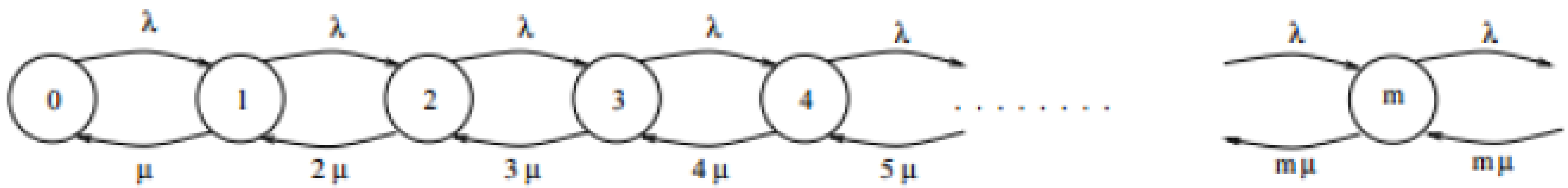
With Little's theorem the mean response time as expected is: $\overline{T} = \frac{1}{\mu}$

# Markov Queues – M/M/m



# Markov Queues – M/M/m

The M/M/m-Queue ($m > 1$) has the same interarrival time and service time distributions as the M/M/1 queue, however, there are m servers in the system and the waiting line is infinitely long. As in the M/M/1 case a complete description of the system state is given by the number of customers in the system (due to the memoryless property). The state-transition-rate diagram of the underlying CTMC is shown in the following Fig. The M/M/m system is also a pure birth-death system.



# Markov Queues – M/M/m

- **An** M/M/m queueing system with arrival rate $\lambda$ and service rate $\mu$ for each server can also be modeled as a birth-death process with

$$\lambda_k = \lambda, \quad k \geq 0,$$

$$\mu_k = \begin{cases} k\mu, & 0 \leq k \leq m, \\ m\mu, & m \leq k. \end{cases}$$

- it is routed to any idle server
- it joins the waiting queue – all servers are busy
- The condition for the queueing system to be stable (underlying CTMC to be ergodic) is $\lambda < m\mu$. The steady-state probabilities are given by (from Eq. (3.11))

$$\pi_k = \begin{cases} \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \cdot \frac{1}{k!}, & 0 \leq k \leq m, \\ \pi_0 \prod_{i=0}^{m-1} \frac{\lambda}{(i+1)\mu} \cdot \prod_{i=m}^{k-1} \frac{\lambda}{m\mu}, & k \geq m. \end{cases}$$

# Markov Queues – M/M/m

- With an individual server utilization, $\rho = \lambda / m\mu$ we obtain

$$\pi_k = \begin{cases} \pi_0 \dfrac{(m\rho)^k}{k!}, & 0 \leq k \leq m, \\ \pi_0 \dfrac{\rho^k m^m}{m!}, & k \geq m, \end{cases}$$

- and from Eq. (3.12) we obtain:

$$\pi_0 = \left[ \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}$$

# Markov Queues – M/M/m

- What is the probability of an arriving job finding all servers are busy? i.e. $P(K \geq m)$

# Markov Queues – M/M/m

- Probability of an arriving job finding that all servers are busy and an arriving customer has to wait in the queue is called Erlang-C formula and is given by:

$$P_m = \sum_{k=m}^{\infty} \pi_k = \frac{p_m}{1-\rho}$$

$$P_m = \frac{(m\rho)^m}{m!} \frac{P_0}{1-\rho}$$

- The above is a call queueing system in a telephone network with an infinite buffer. It's also called Erlang's delayed-call formula.

# Markov Queues – M/M/m

Determine the mean number of jobs in the system?

$$\overline{K} = mp + \frac{\rho}{1-\rho}.P_m$$

## Poisson Arrivals See Time Averages (PASTA)!

### Let us define the following:

- $p_k(t)$ be the probability that the system is in the state $k$ at time $t$;
- $a_k(t)$ be the probability that the arrival at time $t$ finds the system in state $k$;
- $A(t, t + \Delta t)$ be the event of an arrival in the interval $(t, t + \Delta t)$;
- $N(t)$ be the actual number in the system at time $t$.

### The PASTA property claims:

- if the arrival process is Poisson (M/-/-/- queuing systems);
- the state distribution as seen by a new arrival is the same as time-averaged:

$$a_k(t) = p_k(t), \qquad k = 0, 1, \ldots, \qquad t \geq 0.$$

**For $a_k(t)$ we have:**

$$a_k(t) = \lim_{\Delta t \to 0} Pr\{N(t) = k | A(t, t + \Delta t)\}$$
$$= \lim_{\Delta t \to 0} \frac{Pr\{A(t, t + \Delta t) | N(t) = k\} Pr\{N(t) = k\}}{Pr\{A(t, t + \Delta t)\}}$$

**Note the following:**

- arrival process is Poisson and interarrival times are exponential;
- exponential distribution is memoryless;
- number of arrivals in $(t, t + \Delta t)$ does not depend on the state of the system at $t$.

**It leads to:**

$$Pr\{A(t, t + \Delta t) | N(t) = k\} = Pr\{A(t, t + \Delta t)\}.$$

**Substituting we get:**

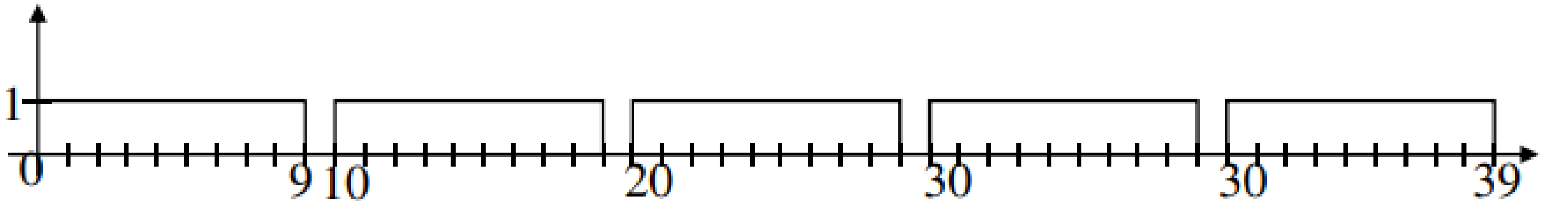
$$a_k(t) = \lim_{\Delta t \to 0} Pr\{N(t) = k\} = p_k(t),$$

**The PASTA property does not hold:**

- if the arrival process is not **homogenous** Poisson;
- if the arrival process depends on something (e.g. state of the system).

## Doesn't PASTA apply for all arrival processes?

- Deterministic arrivals every 10 sec
- Deterministic service times 9 sec
- Upon arrival: system is always empty $a_1$=0
- Average time with one customer in system: $p_1$=0.9



# Markov Queues – M/M/1/K

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

$$\lambda_k = \lambda, 0 \le k \le K$$

$$0, \ \mathrm{k} \ge \mathrm{k}$$

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0 \qquad \pi_0 = ? \Rightarrow \sum_{k=0}^{\infty} \pi_k = 1$$

$$\pi_0\left[1 + \frac{\lambda}{\mu} + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3 + \ldots\left(\frac{\lambda}{\mu}\right)^k\right] = 1$$

$$\pi_0 = \frac{1}{K+1} \qquad \lambda = \mu$$

$$\frac{1-\rho}{1-\rho^{K+1}} \qquad \lambda \ne \mu$$

$$\pi_k = \left(\frac{\lambda}{\mu}\right)^k \pi_0 \Rightarrow$$

$$\pi_k = \rho^k \frac{1-\rho}{1-\rho^{K+1}} \qquad 0 \le k \le K \qquad \lambda \ne \mu$$

$$\frac{\rho^k}{k+1} \qquad 0 \le k \le K \qquad \lambda = \mu$$

# Markov Queues – M/M/1/K

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

$$\lambda_k = \lambda, \; 0 \le k \le K$$

$$0, \; \mathrm{k} \ge \mathrm{k}$$

$$\pi_k = (\frac{\lambda}{\mu})^k \pi_0 \qquad \pi_0$$

For $\sum_{k=m}^{n} r^k$, $(r \neq 1)$

$$(\text{first summand}) \times \frac{1 - r^{\text{number of summands}}}{1-r} = r^m \frac{1 - r^{n-m+1}}{1-r}$$

(and, if $r = 1$, it is simply number of summands $= (n - m + 1)$).

$$\pi_0[1 + \frac{\lambda}{\mu} + (\frac{\lambda}{\mu})^2 + (\frac{\lambda}{\mu})^3 + \ldots (\frac{\lambda}{\mu})^k] = 1$$

$$\pi_0 = \frac{1}{K+1} \qquad \lambda = \mu$$

$$\frac{1-\rho}{1-\rho^{K+1}} \qquad \lambda \neq \mu$$

$$\pi_k = (\frac{\lambda}{\mu})^k \pi_0 \Rightarrow$$

$$\pi_k = \rho^k \frac{1-\rho}{1-\rho^{K+1}} \qquad 0 \le k \le K \qquad \lambda \neq \mu$$

$$\frac{\rho^k}{k+1} \qquad 0 \le k \le K \qquad \lambda = \mu$$

# Markov Queues – M/M/1/K

Average number of packets in M/M/1/K:

$$
\begin{aligned}
\overline{N} &= \sum_{k=1}^{K} k\rho^k P_0 = \rho P_0 \sum_{k=1}^{K} k\rho^{k-1} \\
&= \rho P_0 \left( \sum_{k=1}^{K} \rho^k \right)' = \rho P_0 \left( \rho \frac{1-\rho^K}{1-\rho} \right)' = \rho P_0 \left( \frac{\rho - \rho^{K+1}}{1-\rho} \right)' \\
&= \left( \left(1-(K+1)\rho^K\right)(1-\rho) + \rho - \rho^{K+1} \right) \cdot \frac{\rho P_0}{(1-\rho)^2} \\
&= \frac{\rho P_0 \left(1-(K+1)\rho^K - \rho + (K+1)\rho^{K+1} + \rho - \rho^{K+1}\right)}{(1-\rho)^2} \\
&= \frac{\rho P_0 \left(1-(K+1)\rho^K + K\rho^{K+1}\right)}{(1-\rho)^2} \\
&= \frac{\rho \left(1-(K+1)\rho^K + K\rho^{K+1}\right)}{(1-\rho)(1-\rho^{K+1})}.
\end{aligned}
$$

# Markov Queues – M/M/1/K

- What is the probability of loss i.e. Probability of an arriving packet finding the system in state K (using PASTA theorem):

$$P\{\text{loss}\} = P\{N(t) = K\} = \frac{\rho^K (1-\rho)}{1-\rho^{K+1}}$$

# Markov Queues – M/M/1/K

- **Utilisation:**
  - M/M/1: $$1 - P_0 = \rho = \frac{\lambda}{\mu}$$
  - M/M/1/K: $$1 - P_0 = \rho \frac{1 - \rho^K}{1 - \rho^{K+1}} < \rho$$

# Scaling the arrival and service rate (M/M/1)

- If arrival rate is increased from $\lambda$ to k.$\lambda$

$$\rho^{'} = \frac{\lambda^{'}}{\mu} = \frac{k\lambda}{\mu} = k\rho$$

$$\bar{K}_2 = \frac{\rho}{1-\rho} = \frac{k\rho}{1-k\rho} \geq \frac{k\rho}{1-\rho} = k.\bar{K}_1$$

$$\bar{W}_2 = \frac{1}{\mu - k\lambda} \geq \frac{1}{\mu - \lambda} = \bar{W}_1$$

- Thus, increasing the arrival rate will increase both average number of packets in the system (By a factor of k) and average delay.

# Scaling the arrival and service rate (M/M/1)

- If service rate is increased from $\mu$ to k $\mu$

$$\rho' = \frac{\lambda}{\mu'} = \frac{\lambda}{k\mu} = \frac{1}{k}\rho$$

$$\bar{K}_2 = \frac{\rho}{1-\rho} = \frac{\frac{1}{k}\rho}{1-\frac{1}{k}\rho} \leq \frac{\rho}{k-\rho} = \frac{1}{k}.\bar{K}_1$$

$$\bar{W}_2 = \frac{1}{k\mu-\lambda} \geq \frac{1}{k\mu-k\lambda} = \frac{1}{k}\bar{W}_1$$

- Thus, increasing the service rate by a factor of K will decreases both average number of packets in the system, and average delay by a factor of1/k.

# Scaling the arrival and service rate (M/M/1)

- If arrival rate is increased from $\lambda$ to k.$\lambda$
- And service rate is increased from $\mu$ to k. $\mu$

$$\rho' = \frac{\lambda'}{\mu'} = \frac{k\lambda}{k\mu} = \rho$$

$$\bar{K}_2 = \frac{\rho}{1-\rho} = \bar{K}_1$$

$$\bar{W}_2 = \frac{1}{k\mu - k\lambda} = \frac{1}{k}\frac{1}{\mu-\lambda} = \frac{1}{k}\bar{W}_1$$

- Thus, increasing arrival rate and service rate by a factor of k, will not change average number of packets in the system, however, the average delay is decreased by a factor of 1/k.