Introduction to Markov Chains, Queuing Theory, and Network Performance

Marceau Coupechoux
Telecom ParisTech, departement Informatique et Réseaux
marceau.coupechoux@telecom-paristech.fr
IT.2403 Modélisation et Performance des Réseaux

February 8, 2011

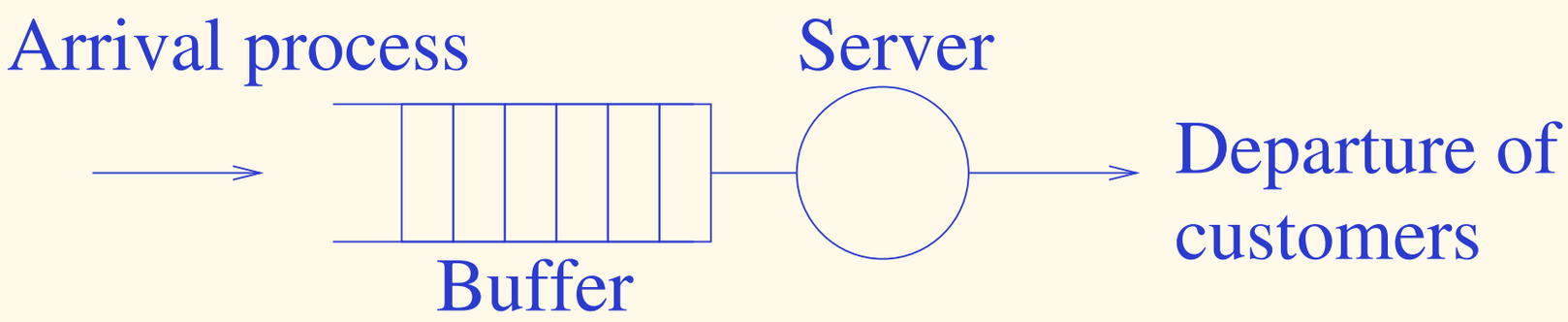
Outlines

1. Network Performance
2. Mathematical Background
3. Markov Chains
4. Formalism of Queuing Theory
5. Simple Queues
6. Erlang B and C laws
7. Networks of Queues

Formalism of Queuing Theory

1. Arrival process and service time
2. Buffer mechanism
3. Kendall notation
4. Multi-class queues
5. Networks of queues
6. Performance parameters
7. Stability
8. Little law
9. Distributions at arrival and departure instants

Simple Queue

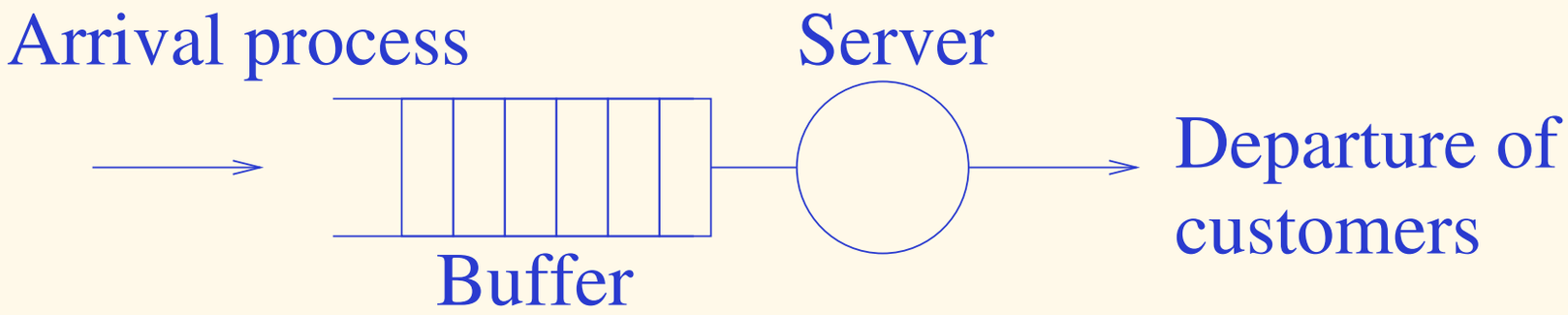


Arrivals

- How do customers arrive ?
- Example: the inter-arrival times are constant
- Simple case: the inter-arrival times between customers are iid random variables
- Poisson arrivals: inter-arrival times are exponentially distributed

Formalism of Queuing Theory

Simple Queue

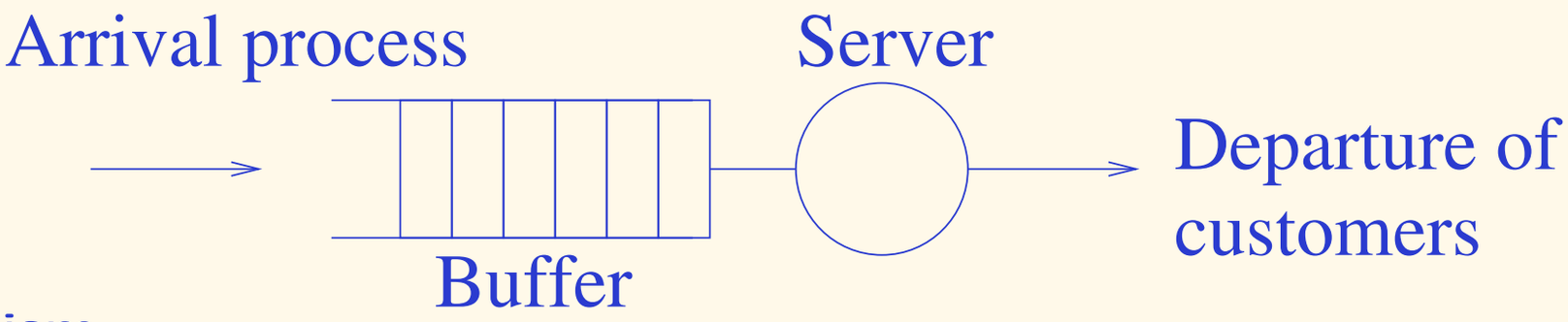


Service time

- How are customers served ?
- Are there more than one server ?
- How much time will the service take (service time distribution) ?
- Simple case: service times are iid random variables
- Exponential service: service time is exponentially distributed

Formalism of Queuing Theory

Simple Queue



Buffer mechanism

- How customers are stacked in the buffer and how customers are taken out of the buffer by the server

- FIFO: first in first out = FCFS : first come first served

- LIFO: last in first out

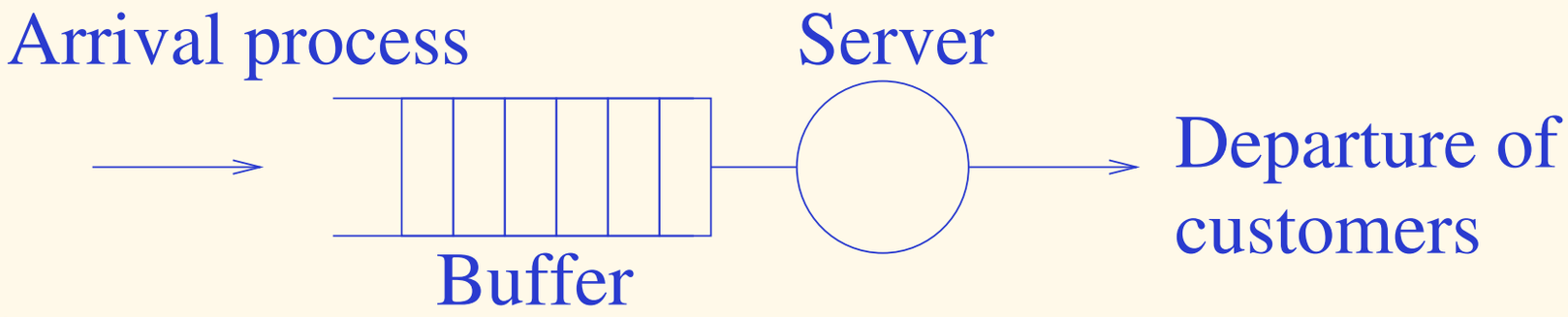
- RANDOM

- Round Robin: cyclic mechanism

- etc

Formalism of Queuing Theory

Simple Queue

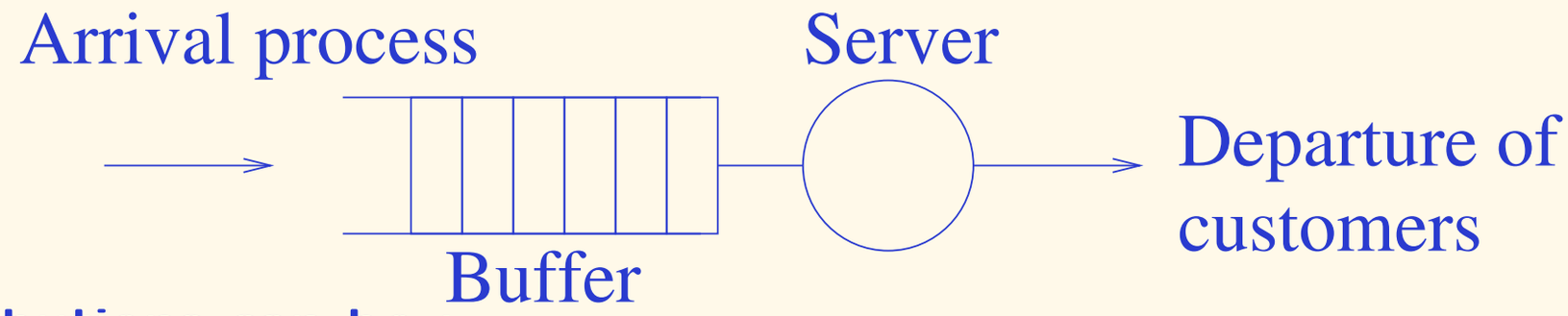


Kendall notation

- Queue A/B/C/K/m/Z
- A: inter-arrival distribution
- B: service distribution
- C: number of servers
- K: buffer capacity (default ∞)
- m: customer population (default ∞)
- Z: buffer mechanism (default FIFO)

Formalism of Queuing Theory

Simple Queue

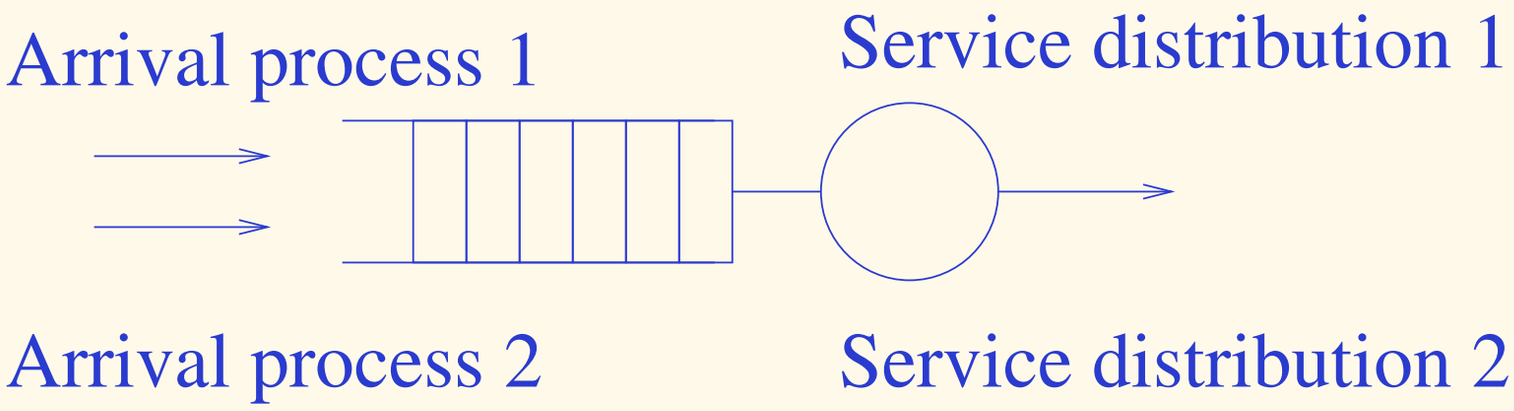


A and B distributions can be :

- M: exponential
- G: generic
- D: constant
- E_k: Erlang-k
- H_k: Hyperexponential-k
- C_k: Cox-k
- etc

Multi-class queues

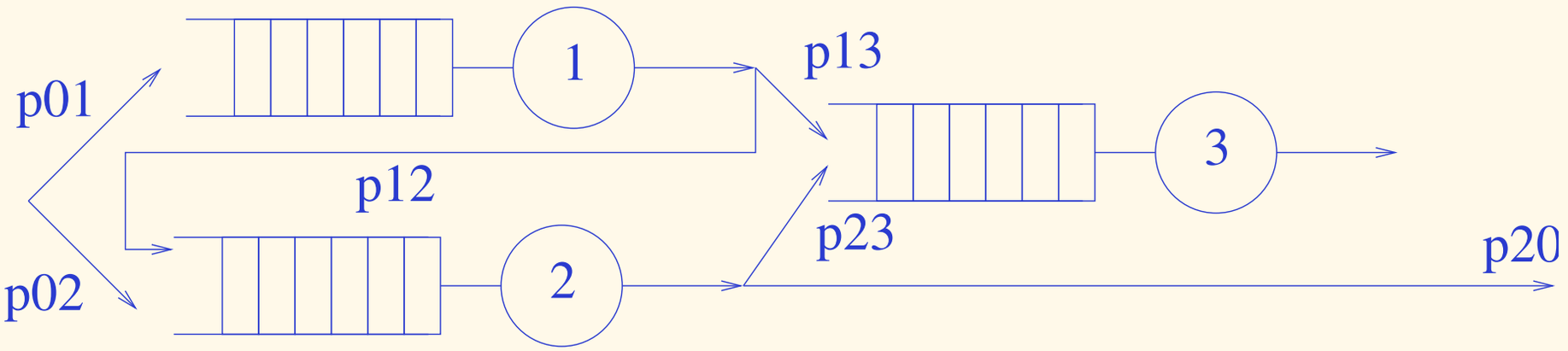
- Customers of different classes can experience different arrival process, or service times...



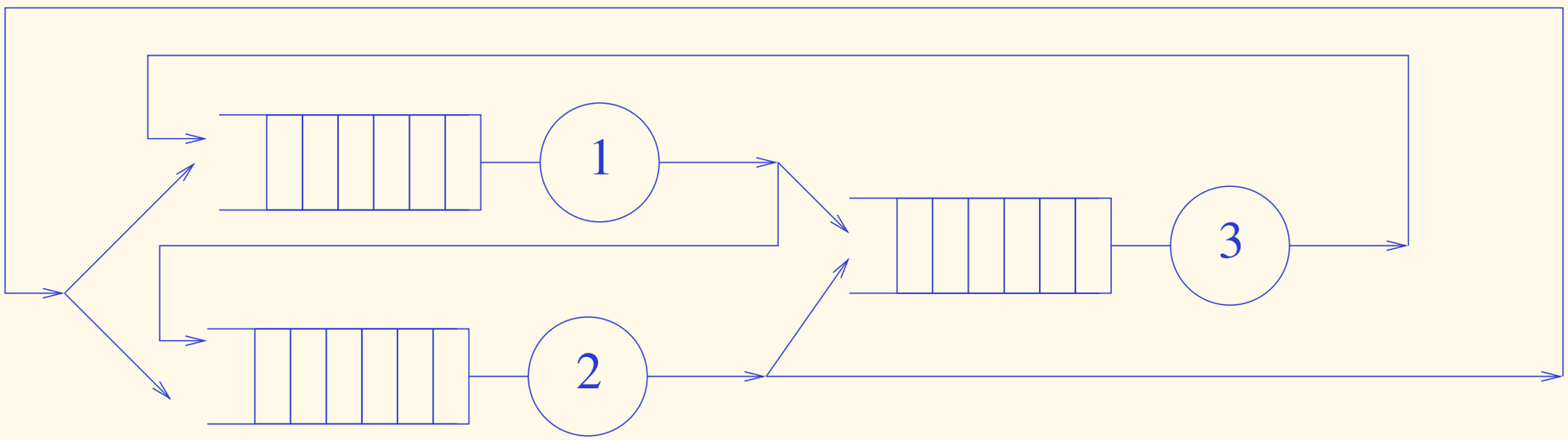
Network of queues

- Several connected simple queues

- Open networks: customers can leave the system and are coming from the outside



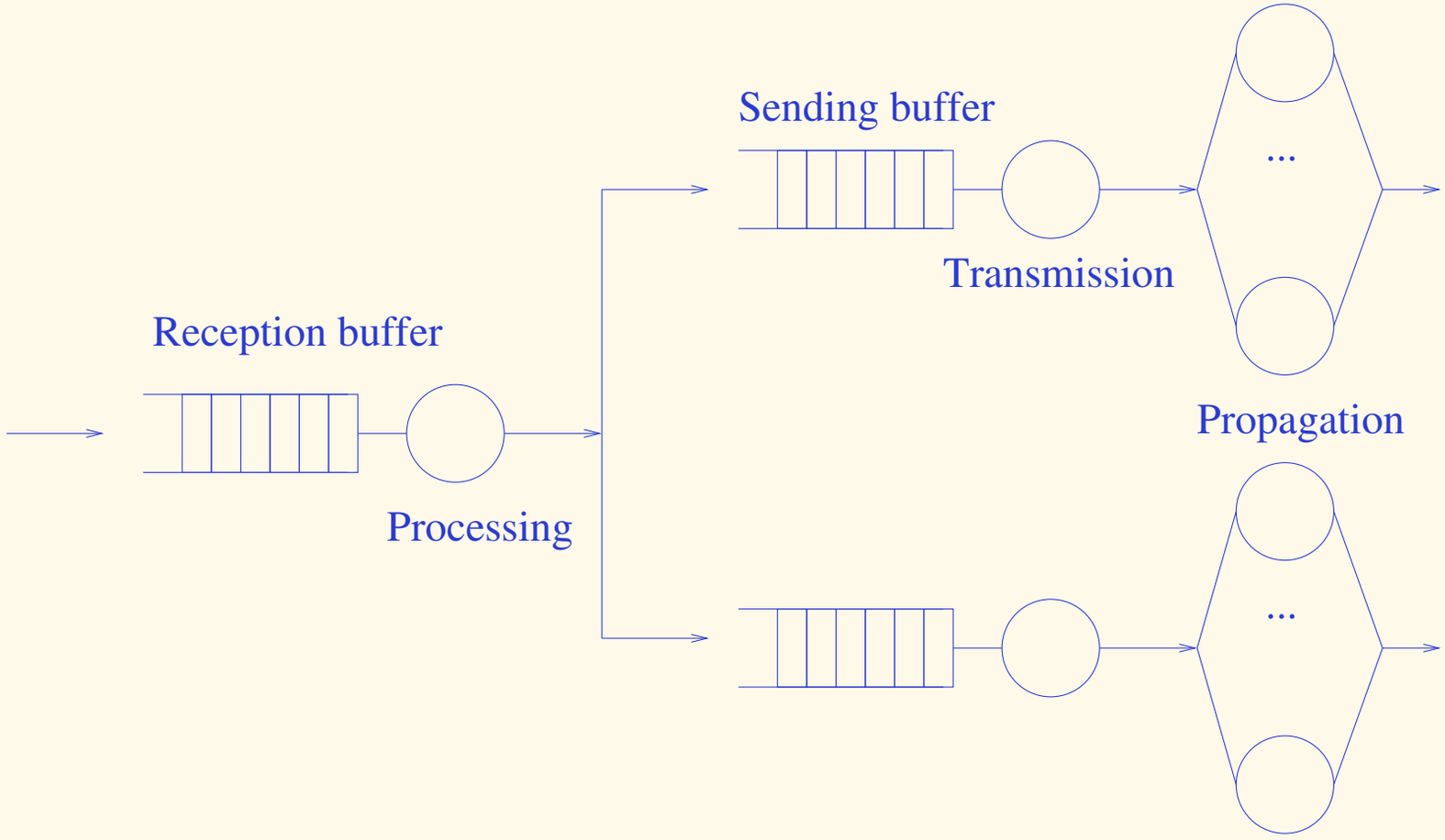
- Closed networks: finite population, customers can't leave the system



Formalism of Queuing Theory

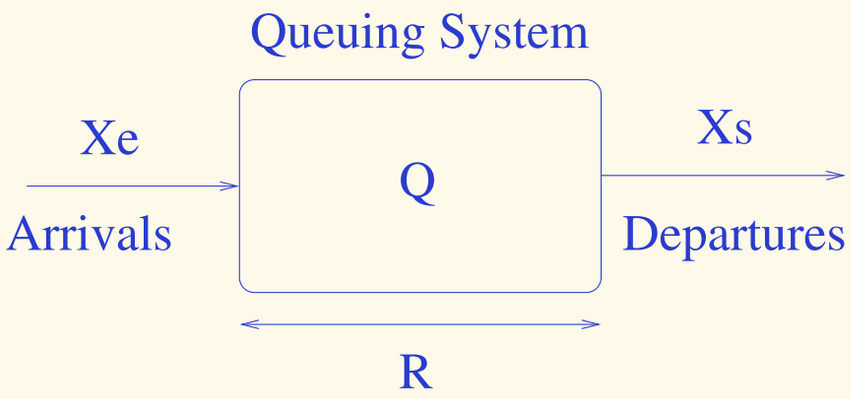
Packet network

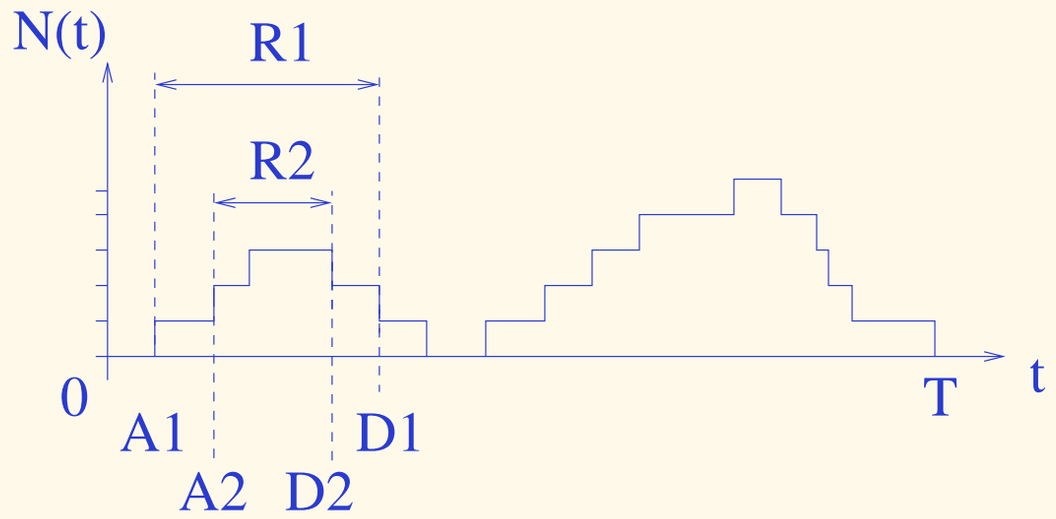
- Is modeled by a network of queues
- A network node: arrival buffer + processing time + sending buffers
- A transmission link: transmission time + propagation time



Formalism of Queuing Theory

Performance parameters





- A_k: arrival instant of the kth customer in the system
- D_k: departure instant of the kth customer in the system
- $R_k = D_k - A_k$: time spent in the system by the kth customer
- $T(n,T)$: time during which the system has n customers : $\sum_{n=0}^{\infty} T(n,T) = T$
- $A(T)/D(T)$: number of arrivals/departures in $[0;T]$

Performance parameters

- Input load: $X_e(T) = A(T)/T$
- Throughput: $X_s(T) = D(T)/T$
- Mean number of customers: $Q(T) = \frac{1}{T}\sum_{n=0}^{\infty} nT(n,T)$
- Mean time spent in the system: $R(T) = \frac{1}{A(T)}\sum_{k=1}^{A(T)} R_k$
- Utilization ratio: $U(T) = 1 - T(0,T)/T$

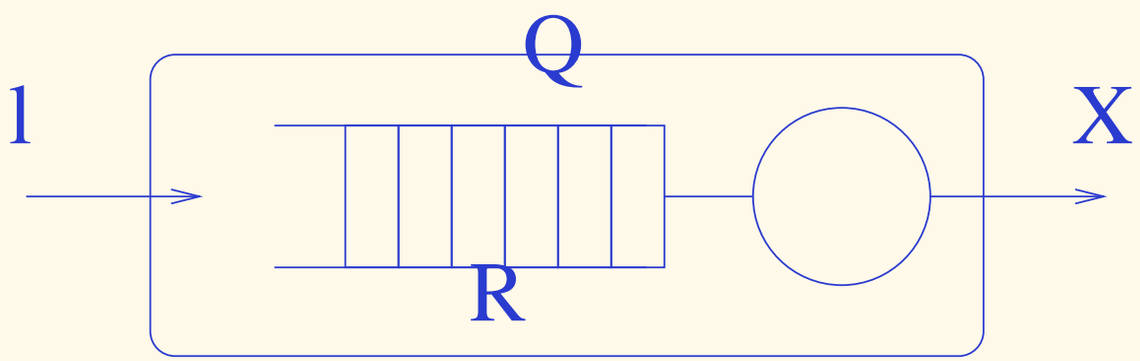
Stability

- A system is stable iff the input load equals the throughput in the stationary regime
- i.e. $\lim_{T\to\infty} X_e(T) = \lim_{T\to\infty} X_s(T)$
- or $\lim_{T\to\infty} \frac{D(T)}{A(T)} = 1$
- A mono-class open networks of queues is stable iff all queues of the network are stable
- A finite population system is always stable

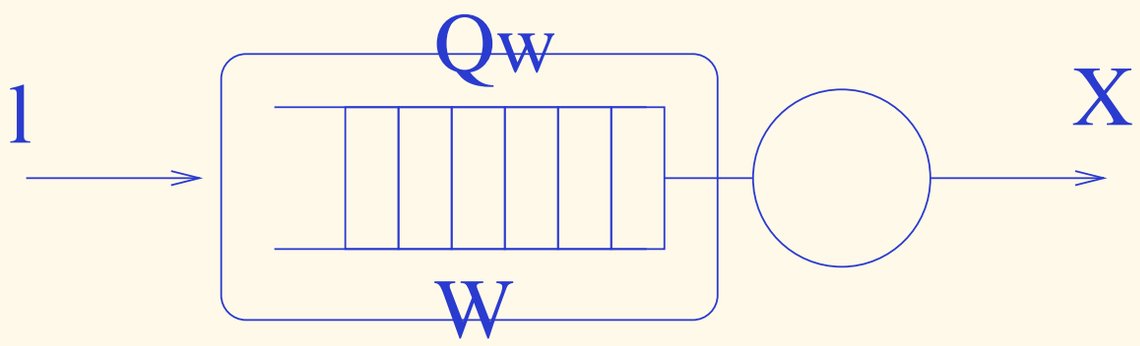
Little law

- For a stable system in stationary regime :

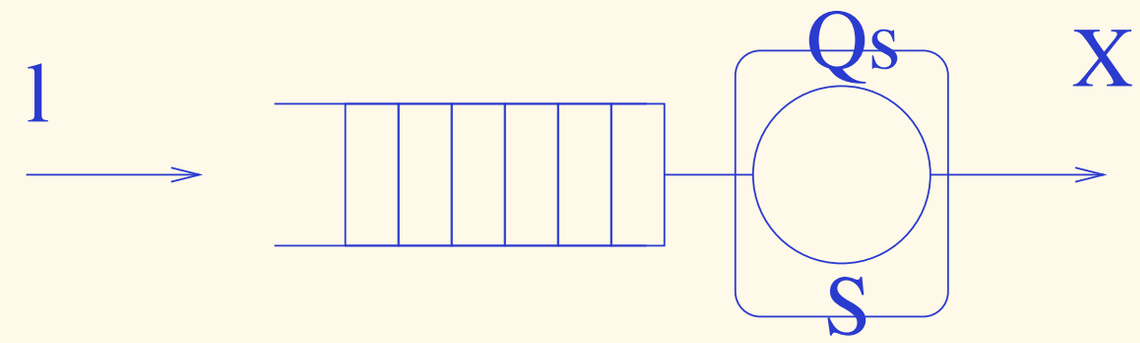
$$Q = RX$$



Q=RX=Rl



Qw=WX=Wl



Qs=U=SX=Sl

Distributions at arrival and departure instants

- $A(n,T)$: number of customers who find n customers in the system at their arrival
- $D(n,T)$: number of customers who leave n customers in the system at their departure
- $P(n,T) = T(n,T)/T$: proportion of time during which there are n customers in the system
- $P_a(n,T) = A(n,T)/A(T)$: proportion of clients that find n customers in the system
- $P_d(n,T) = D(n,T)/D(T)$: proportion of clients that leave n customers in the system
- $p(n) = \lim_{T\to\infty} P(n,T)$: stationary probability to have n customers in the system
- $p_a(n) = \lim_{T\to\infty} P_a(n,T)$: stationary probability for a client to find n customers
- $p_d(n) = \lim_{T\to\infty} P_d(n,T)$: stationary probability for a client to leave n customers

Properties

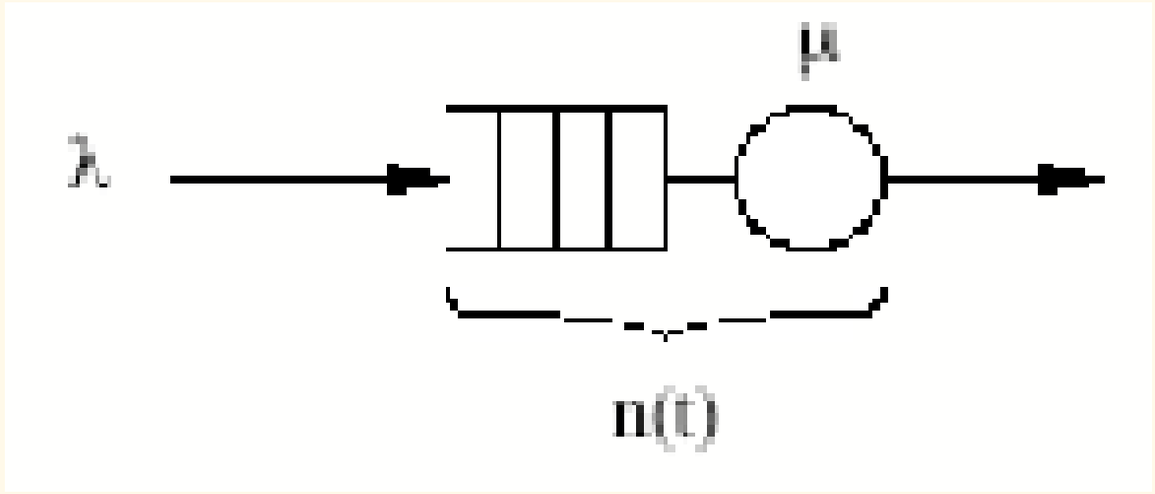
- In a (stable and ergodic) system where customers arrive or leave one by one:

$$p_a(n) = p_d(n)$$

- In a system with Poisson arrivals:

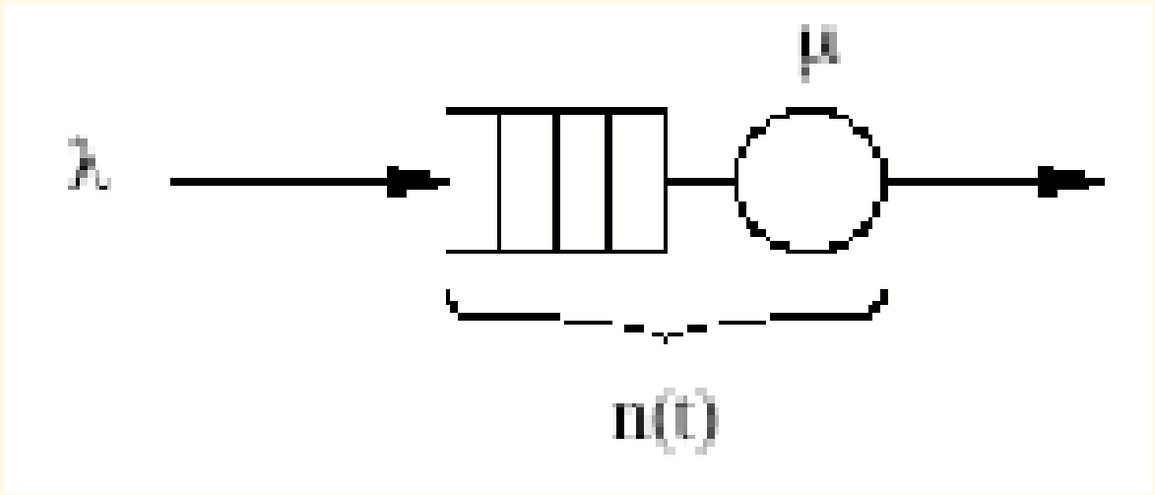
$$p_a(n) = p(n)$$

1. M/M/1
   (a) Definition
   (b) Associated CTMC
   (c) Stability
   (d) Analysis
2. M/M/1/K
   (a) Definition
   (b) Associated CTMC
   (c) Stability
   (d) Analysis
3. Markovian queues



M/M/1, Definition

- Infinite capacity
- A single server
- Poisson arrivals of rate λ
- Exponential service time of rate μ

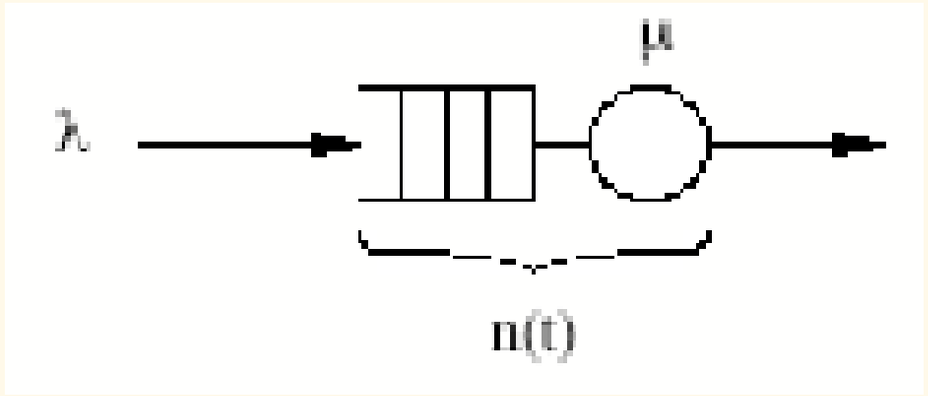


M/M/1, Associated DTMC

- Generic method: Find the "right" state description → stochastic process → analysis → performance parameters

- What is a "right" state description ? The knowledge of the present state is sufficient to predict the evolution of the system = the generated stochastic process is a Markov chain

- State description for the M/M/1: $\{n(t)\}_{t\geq 0}$

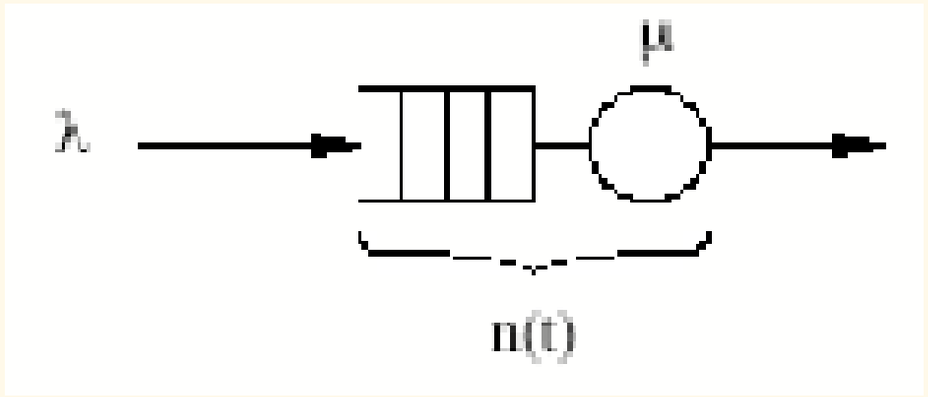


M/M/1, Associated DTMC

- Stochastic process with discrete state space and continuous time

- Memoryless property of the service time: it is not necessary to know for how long time the service has started to predict the remaining service time

- Memoryless property of the inter-arrival law: it is not necessary to know when the last customer arrived to predict the arrival of the next one



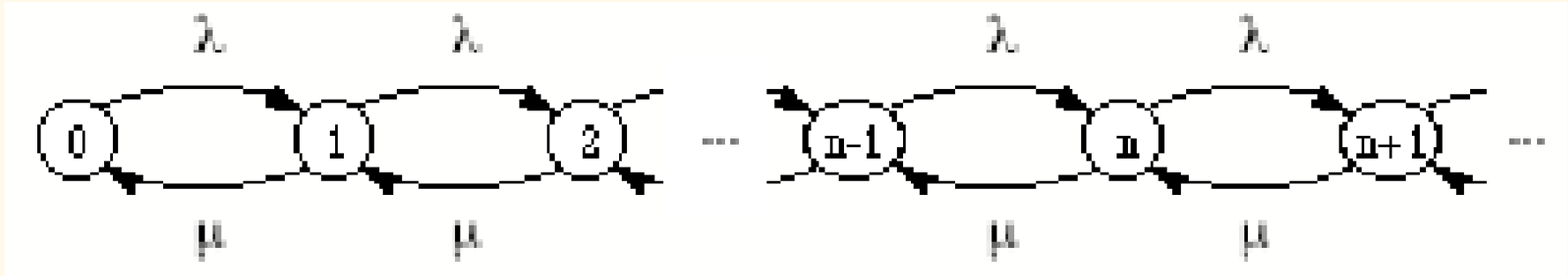
M/M/1, Stability

- Stability condition:

$$\lambda < \mu$$

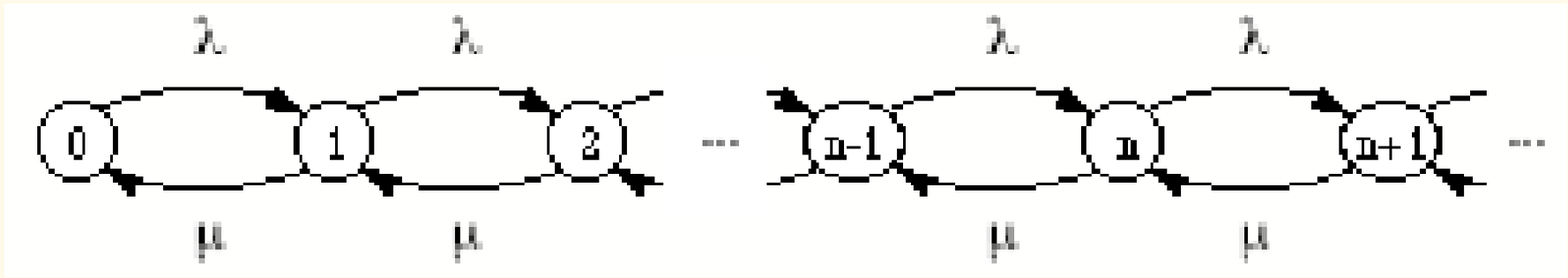
$$\rho = \frac{\lambda}{\mu} < 1$$

- ρ is referred to as the traffic intensity



M/M/1, Analysis

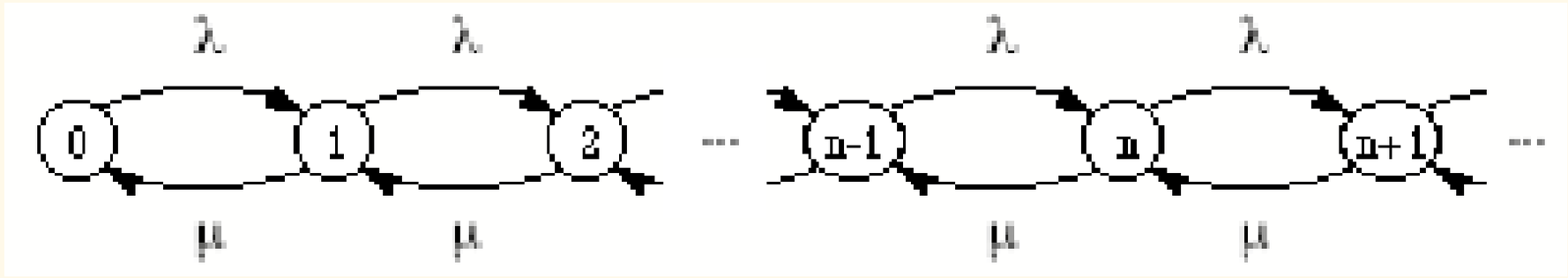
- We assume that the system is stable
- Three methods to derive the stationary (or steady-state) probabilities:
- (1) Solve $pQ = 0$ and $\sum_{n=0}^{\infty} p(n) = 1$
- (2) Solve the state equations: for each state, input flow=output flow
- (3) Solve the border equations



M/M/1, Analysis

- $pQ = 0$

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & ... & ... & ... \\ \mu & -(\lambda+\mu) & \lambda & 0 & ... & ... \\ 0 & \mu & -(\lambda+\mu) & \lambda & 0 & ... \\ ... & 0 & \mu & -(\lambda+\mu) & \lambda & 0 \\ ... & ... & 0 & \mu & -(\lambda+\mu) & \lambda \\ ... & ... & ... & 0 & \mu & ... \end{pmatrix}$$

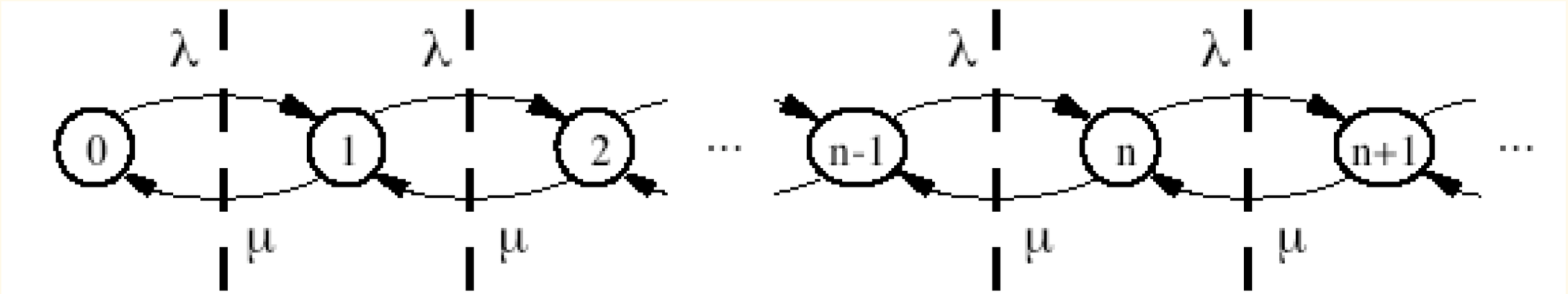


M/M/1, Analysis

- state equations: input flow=output flow

$$\begin{aligned} p(0)\lambda &= p(1)\mu \\ p(1)(\lambda+\mu) &= p(0)\lambda + p(2)\mu \\ &\ldots \\ p(n)(\lambda+\mu) &= p(n-1)\lambda + p(n+1)\mu \text{ for all } n \geq 1 \end{aligned}$$

Simple Queues



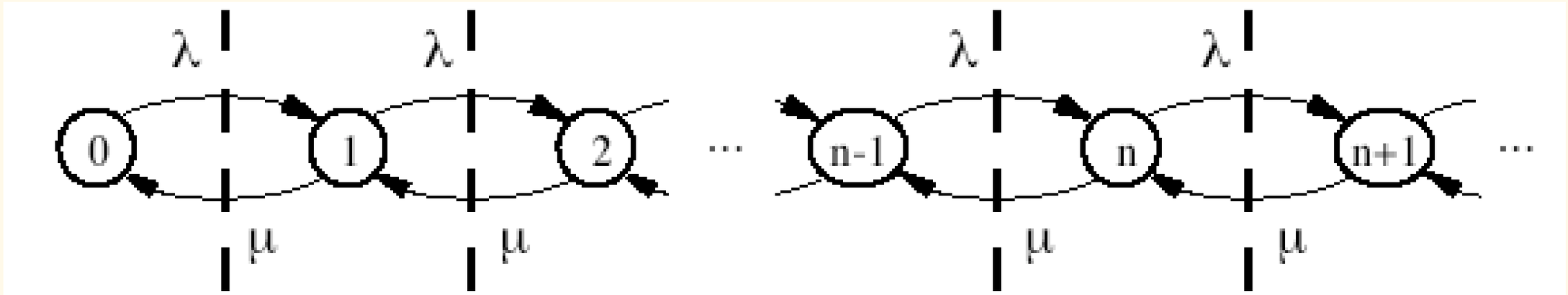
- border equations:

$$\begin{aligned} p(0)\lambda &= p(1)\mu \\ p(1)\lambda &= p(2)\mu \\ &\dots \\ p(n)\lambda &= p(n+1)\mu \text{ for all } n \geq 0 \end{aligned}$$

- $p(n) = \rho p(n-1)$

- $p(n) = \rho^n p(0)$

Simple Queues



- border equations:

$$\begin{aligned} p(0)\lambda &= p(1)\mu \\ p(1)\lambda &= p(2)\mu \\ &\ldots \\ p(n)\lambda &= p(n+1)\mu \text{ for all } n \geq 0 \end{aligned}$$

- Normalization : $p(0) = \frac{1}{\sum_{n=0}^{\infty} \rho^n} = 1 - \rho$

- Thus, for $\rho < 1$, $\forall n \geq 0\ p(n) = (1-\rho)\rho^n$

M/M/1, Performance parameters

- Throughput:

$$X = Proba[\text{queue non empty}]\mu = \sum_{n=1}^{\infty} p(n)\mu = \rho\mu = \lambda$$

- Server utilization:

$$U = 1 - p(0) = \rho = \frac{\lambda}{\mu}$$

- Average number of customers:

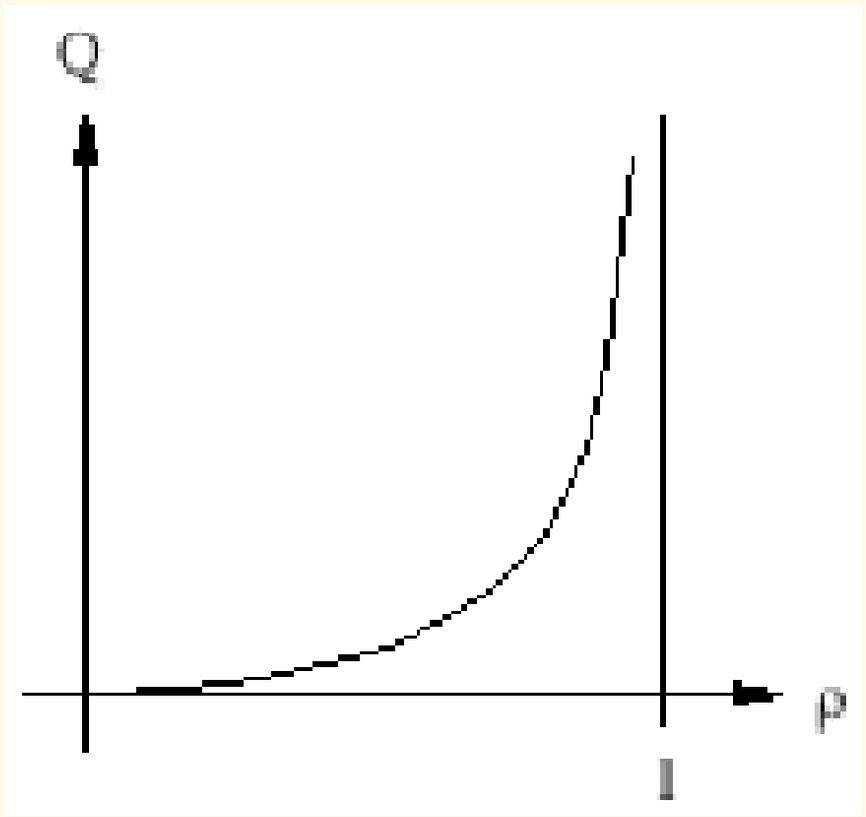
$$Q = \sum_{n=1}^{\infty} np(n) = \frac{\rho}{1-\rho}$$

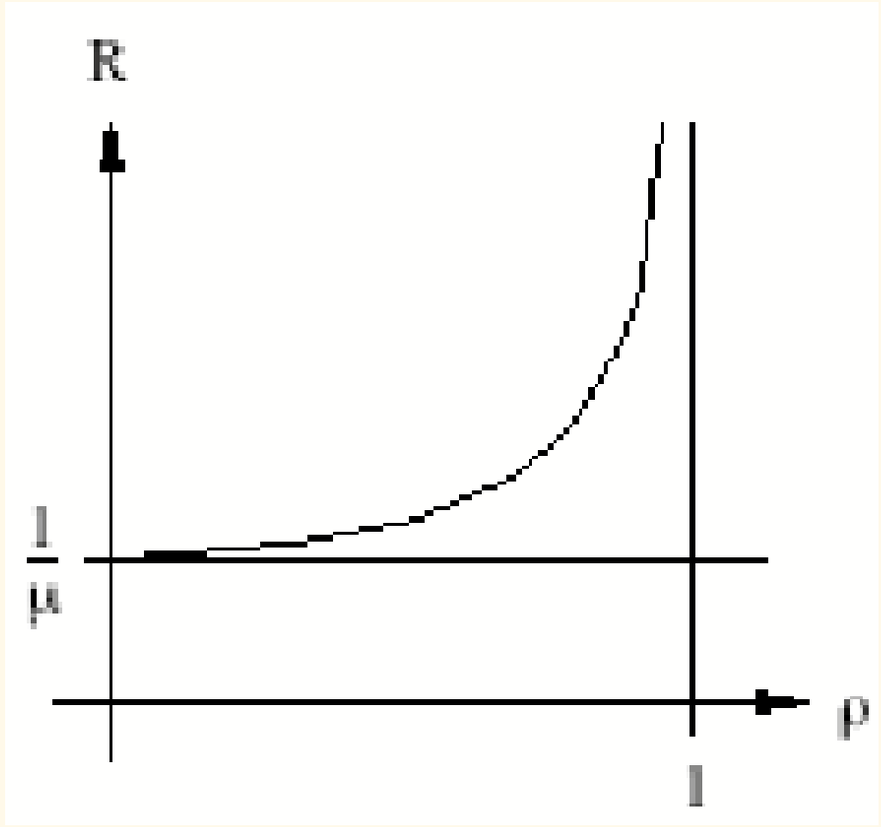
- Average time spent by a customer in the system (Little law):

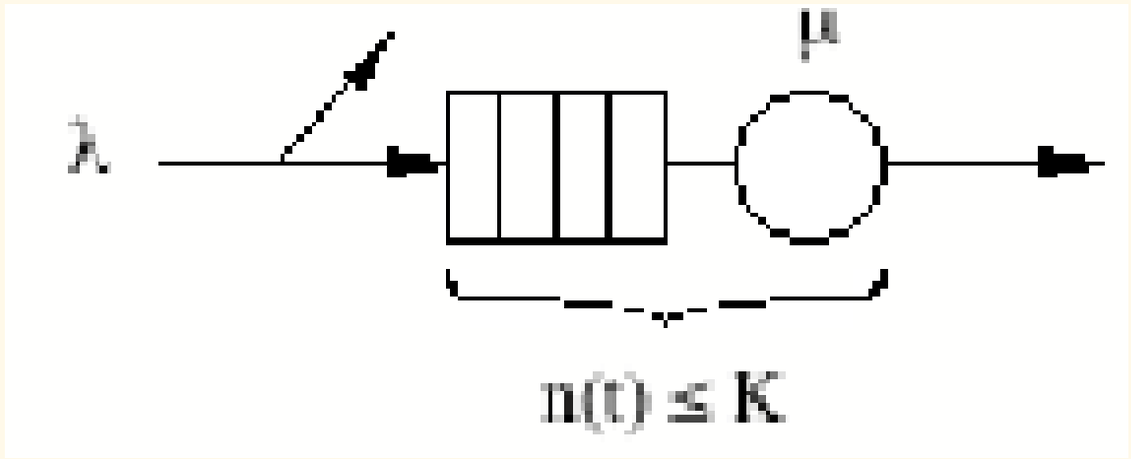
$$R = \frac{Q}{X} = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu} + \frac{\rho}{\mu(1-\rho)}$$

M/M/1, Performance parameters

- Q and R:

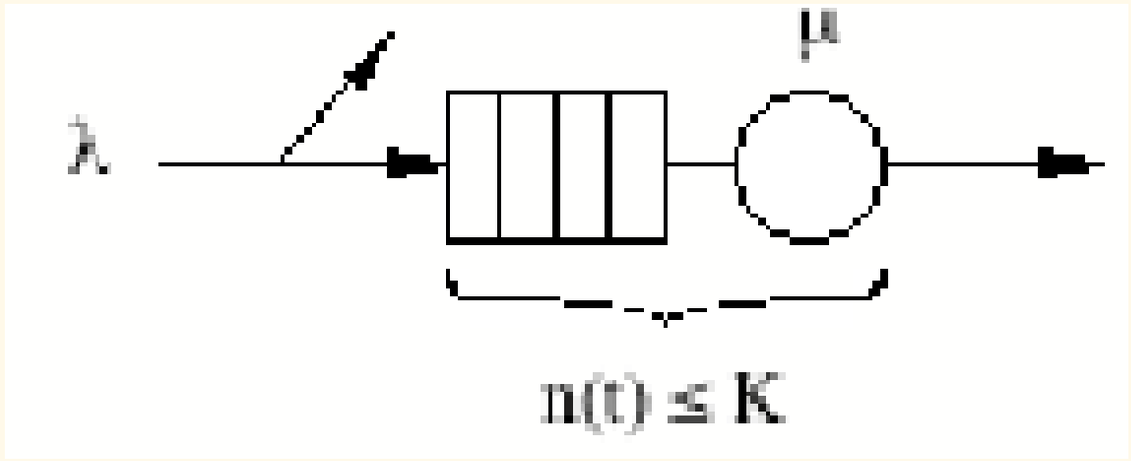






M/M/1/K, Definition

- The system (buffer and server) is limited to K customers
- A single server
- Poisson arrivals of rate λ
- Exponential service time of rate μ

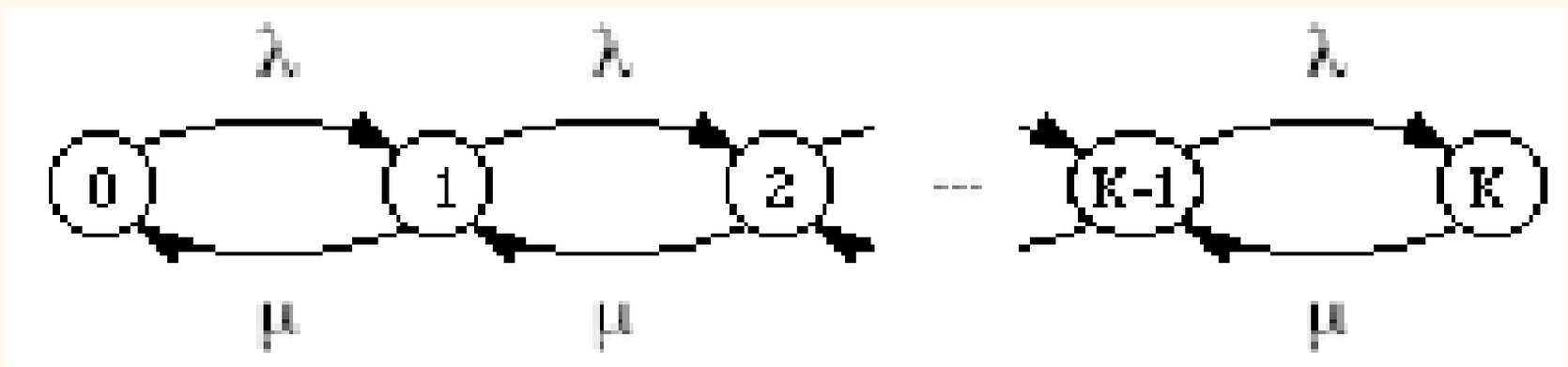


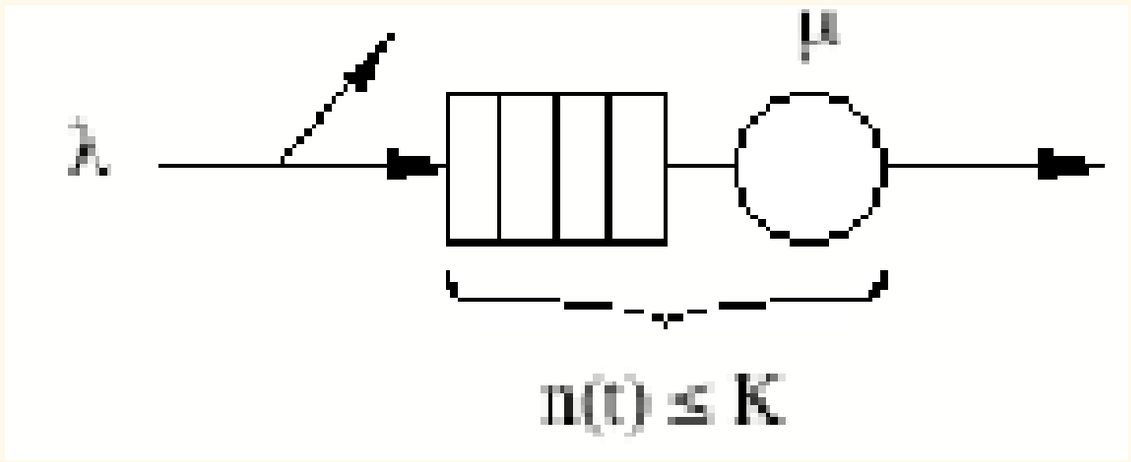
M/M/1/K, Associated CTMC

- State description: $\{n(t)\}_{t\geq 0}$

- Stochastic process with continuous time and discrete state space

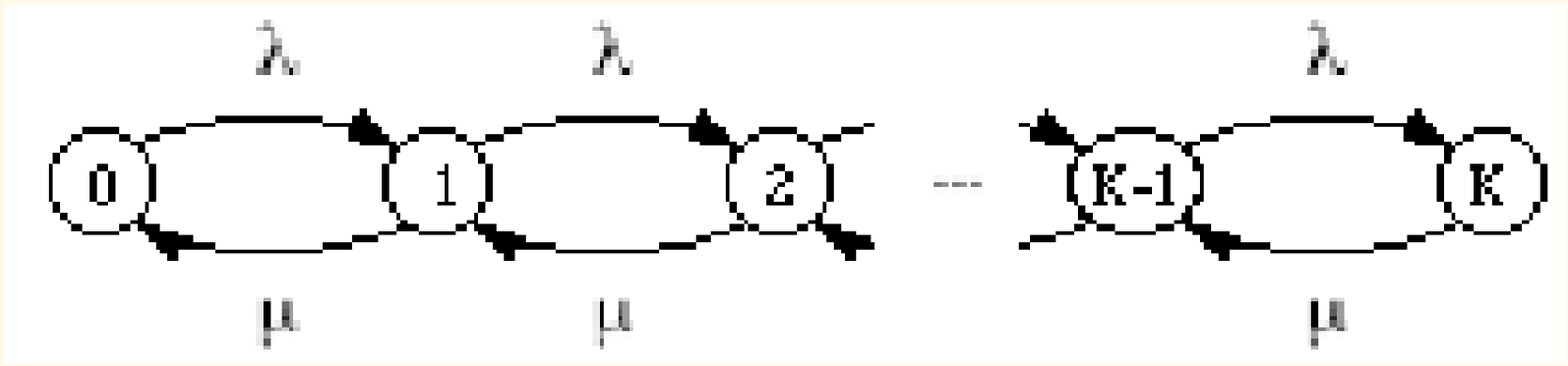
- At a given time, when there is $0 < n(t) < K$ customers in the system, both inter-arrival and service times are memoryless

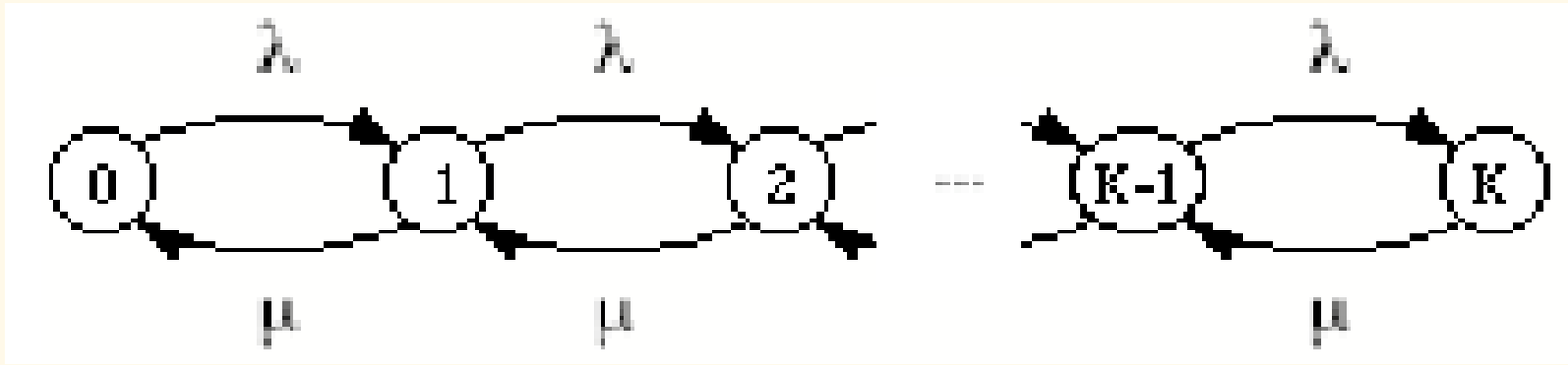




M/M/1/K, Stability

- The system is limited → the system is always stable





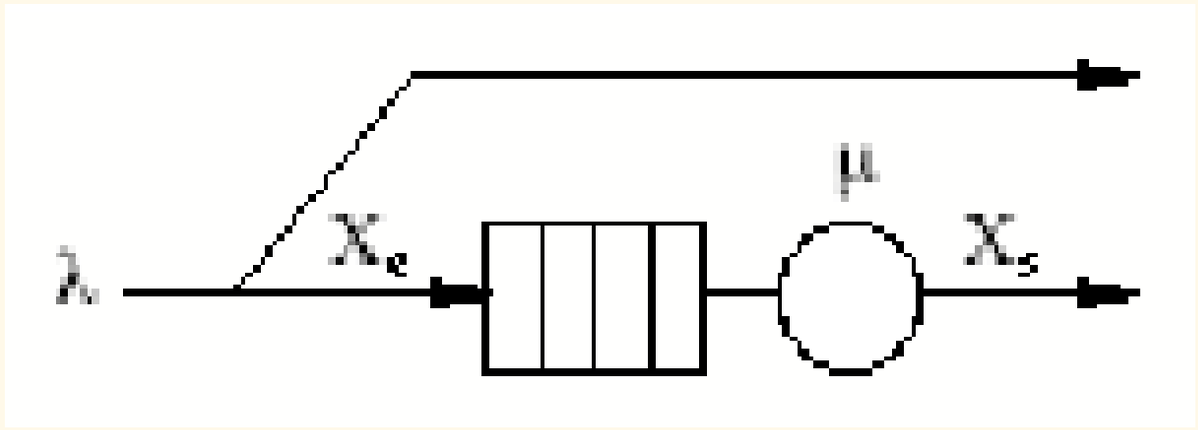
M/M/1/K, Analysis

- Border equations:

$$\begin{aligned} \forall n,\ 1 \leq n \leq K,\ p(n-1)\lambda &= p(n)\mu \\ \forall n,\ 1 \leq n \leq K,\ p(n) &= \rho p(n-1) \\ \forall n,\ 0 \leq n \leq K,\ p(n) &= \rho^n p(0) \end{aligned}$$

- Normalization:

$$p(0) = \frac{1}{\sum_{n=0}^{K} \rho^n} = \frac{1-\rho}{1-\rho^{K+1}}$$

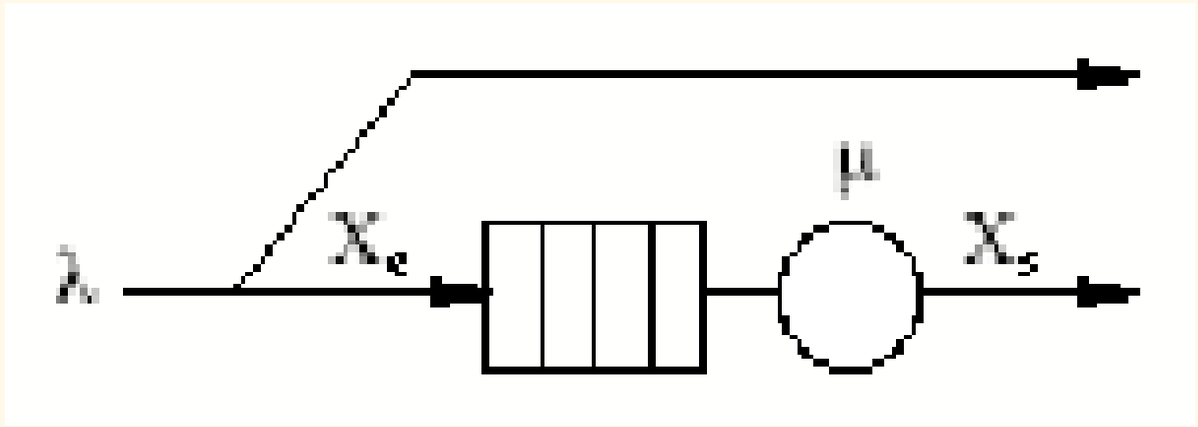


M/M/1/K, Performance parameters

- Throughput X: two methods

- (1) Departure rate from the server:

$$X_s = Proba[\text{queue not empty}]\mu = (1 - p(0))\mu = \frac{\rho - \rho^{K+1}}{1 - \rho^{K+1}}\mu$$



M/M/1/K, Performance parameters

- Throughput X: two methods
- (2) Arrivals in the system (not lost):

$$\begin{aligned} X_e &= Proba[\text{queue not full when a customer arrives}]\lambda \\ &= \sum_{n=0}^{K-1} p_a(n)\lambda \\ &= \sum_{n=0}^{K-1} p(n)\lambda \\ &= \frac{1-\rho^K}{1-\rho^{K+1}}\lambda \end{aligned}$$

M/M/1/K, Performance parameters

- Server utilization U:

$$U = 1 - p(0) = \rho \frac{1 - \rho^K}{1 - \rho^{K+1}}$$

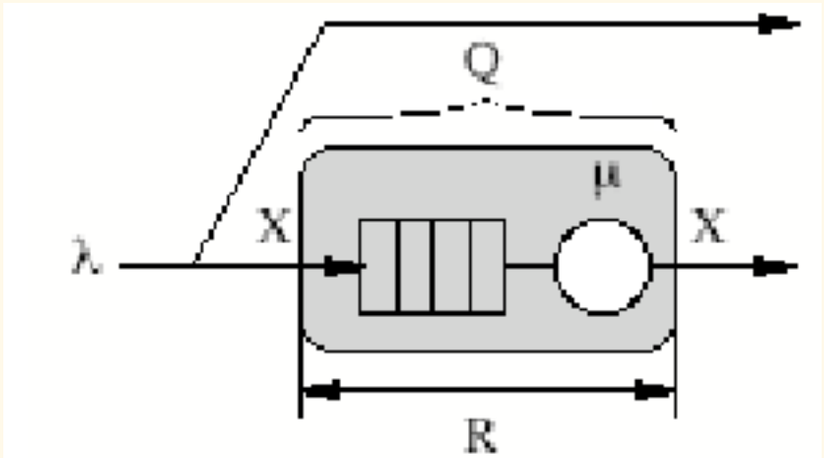
- Average number of customers Q:

$$Q = \sum_{n=0}^{K} np(n) = \frac{\rho}{1 - \rho} \times \frac{1 - (K+1)\rho^K + K\rho^{K+1}}{1 - \rho^{K+1}}$$

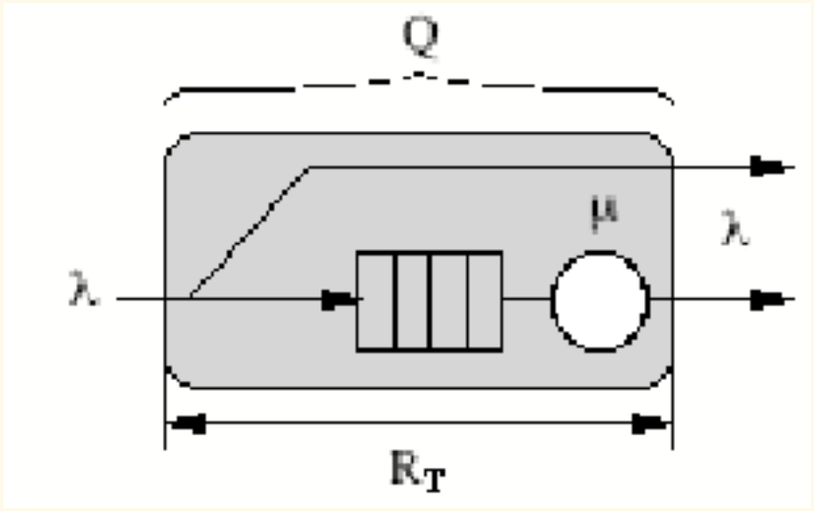
M/M/1/K, Performance parameters

- Average time spent by a customers in the system: 2 possible parameters

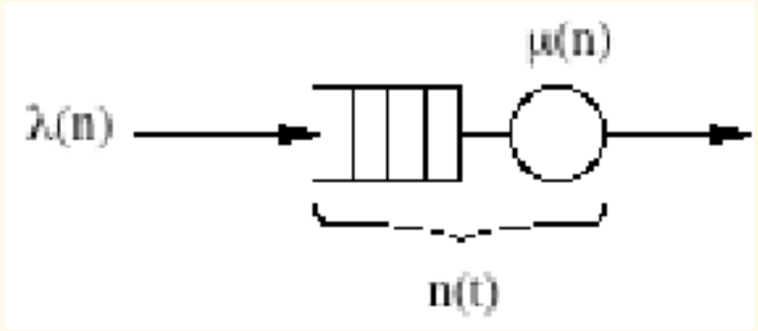
- (1) Time spent by a customers admitted in the system: $R = Q/X$



- (2) Time spent by all customers trying to enter the system: $R_T = Q/\lambda$

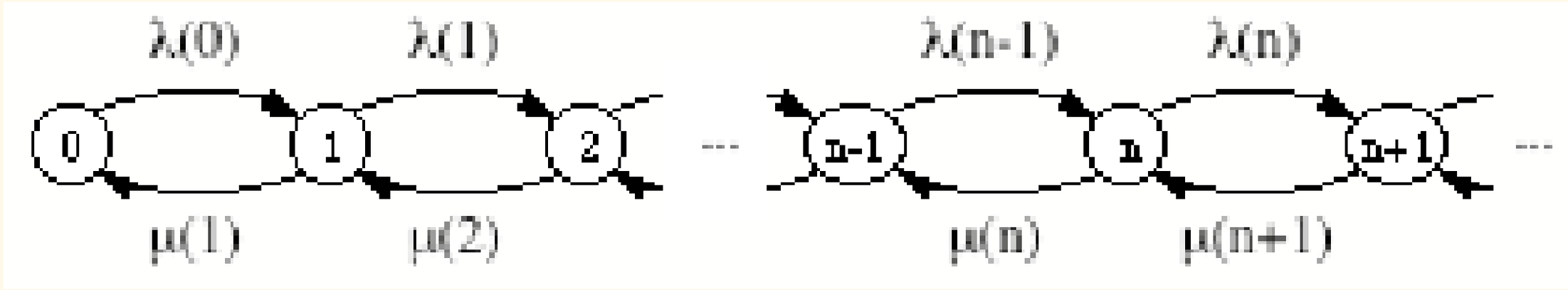


Simple Queues



Markovian queues, Definition

- Infinite buffer
- A single server
- Inter-arrival and service times law are memoryless and depends on the number of customers in the system
- When there are n customers in the system at a given time instant:
- - the arrival time of the next customer follows an exponential law of rate $\lambda(n)$
- - the service time follows an exponential law of rate $\mu(n)$

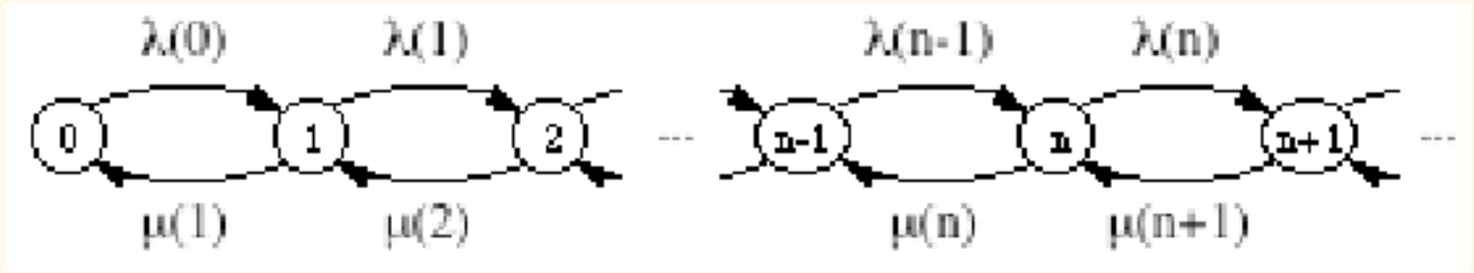


Markovian queues, Associated CTMC

- State description: $\{n(t)\}_{t\geq 0}$

- Stochastic process with continuous time and discrete state space

Simple Queues



Markovian queues, Stability

- Stability condition:

$$p(0) > 0$$

$$\frac{1}{1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda(k-1)}{\mu(k)} \right)} > 0$$

$$\sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda(k-1)}{\mu(k)} \right) < \infty$$

Markovian queues, Analysis

- Border equations:

$$\forall n > 0,\ p(n-1)\lambda(n-1) = p(n)\mu(n)$$

$$p(n) = \prod_{k=1}^{n} \frac{\lambda(k-1)}{\mu(k)} p(0)$$

- Normalization:

$$p(0) = \frac{1}{1 + \sum_{n=1}^{\infty} \left(\prod_{k=1}^{n} \frac{\lambda(k-1)}{\mu(k)} \right)}$$

Markovian queues, Performance parameters

- Throughput X: 2 methods

- (1) At the exit of the system:

$$X = \sum_{n=1}^{\infty} p(n)\mu(n)$$

- (2) At the entrance of the system:

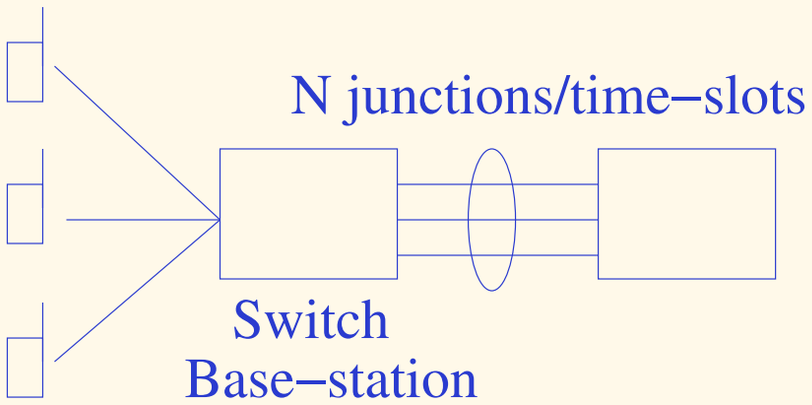
$$X = \sum_{n=0}^{\infty} p_a(n)\lambda(n) = \sum_{n=0}^{\infty} p(n)\lambda(n)$$

- Average number of customers in the system Q:

$$Q = \sum_{n=1}^{\infty} np(n)$$

- Average time spent by a customer in the system R: $R = Q/X$

Telephonic traffic



- Main issue: find the right number of servers associated to a given quality of service (QoS)

- Traffic a of a server (ITU-T E-100) : average proportion of occupation time of the server (t is the occupation time between 0 and T, T is the observation time) :

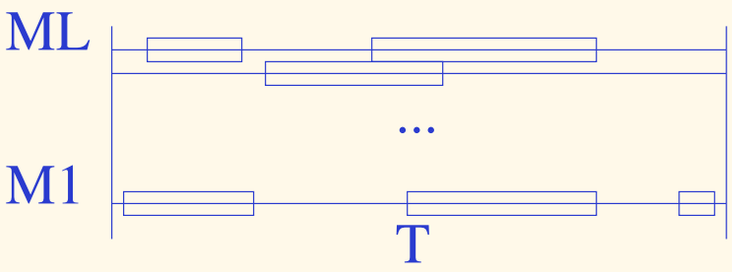
$$a = \frac{t}{T}$$

- a is in Erlangs

- US : Cent Call Second (CCS) is the occupation of a machine during 100s per hour

$$1CCS = 100/3600 = 1/36 Erlangs$$

Telephonic traffic



- Traffic of a group of L servers:

$$A = \frac{\sum_i t_{M_i}}{T} \leq L Erlangs$$

- Identical servers, with n: mean number of occupation periods in T and τ: mean occupation time of a server:

$$A = \frac{n\tau}{T}$$

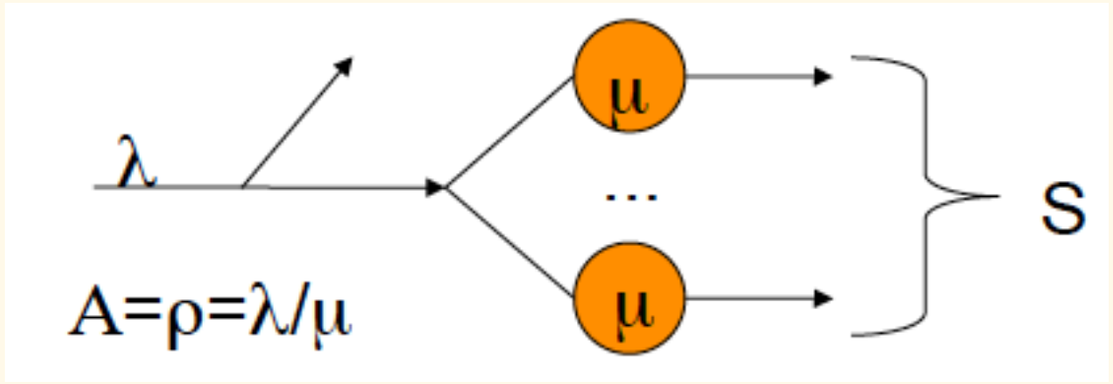
- Ergodicity: statistics of the group at a given time and statistics of a given server in time are equal

- with $\bar{x}$: mean number of occupied servers at a given time and the ergodicity property:

$$\frac{\bar{x}}{L} = a$$

Erlang B model, assumptions

- We consider S servers serving phone calls
- Calls arrive according to a Poisson law of rate λ, λ does not depend on the number of occupied servers (infinite population assumption)
- Calls that cannot be served are lost
- Service time is distributed according to a decreasing exponential law of rate μ



Erlang B, markov chain

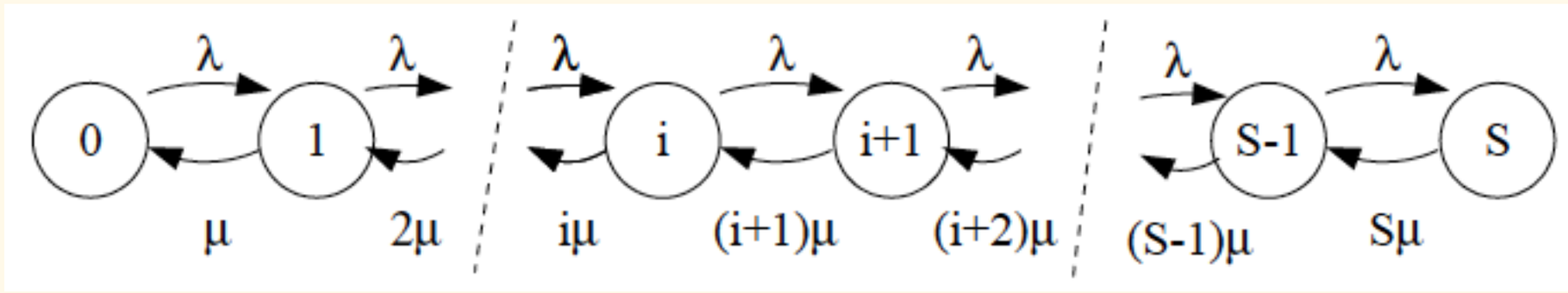
- The system is modeled with a M/M/S/S.

- Stationary probabilities are given by :

$$p(n) = \rho^n/n!p(0) \text{ and } p(0) = \left(\sum_{k=0}^{S} \rho^k/k!\right)^{-1} \text{ with } \rho = \lambda/\mu$$

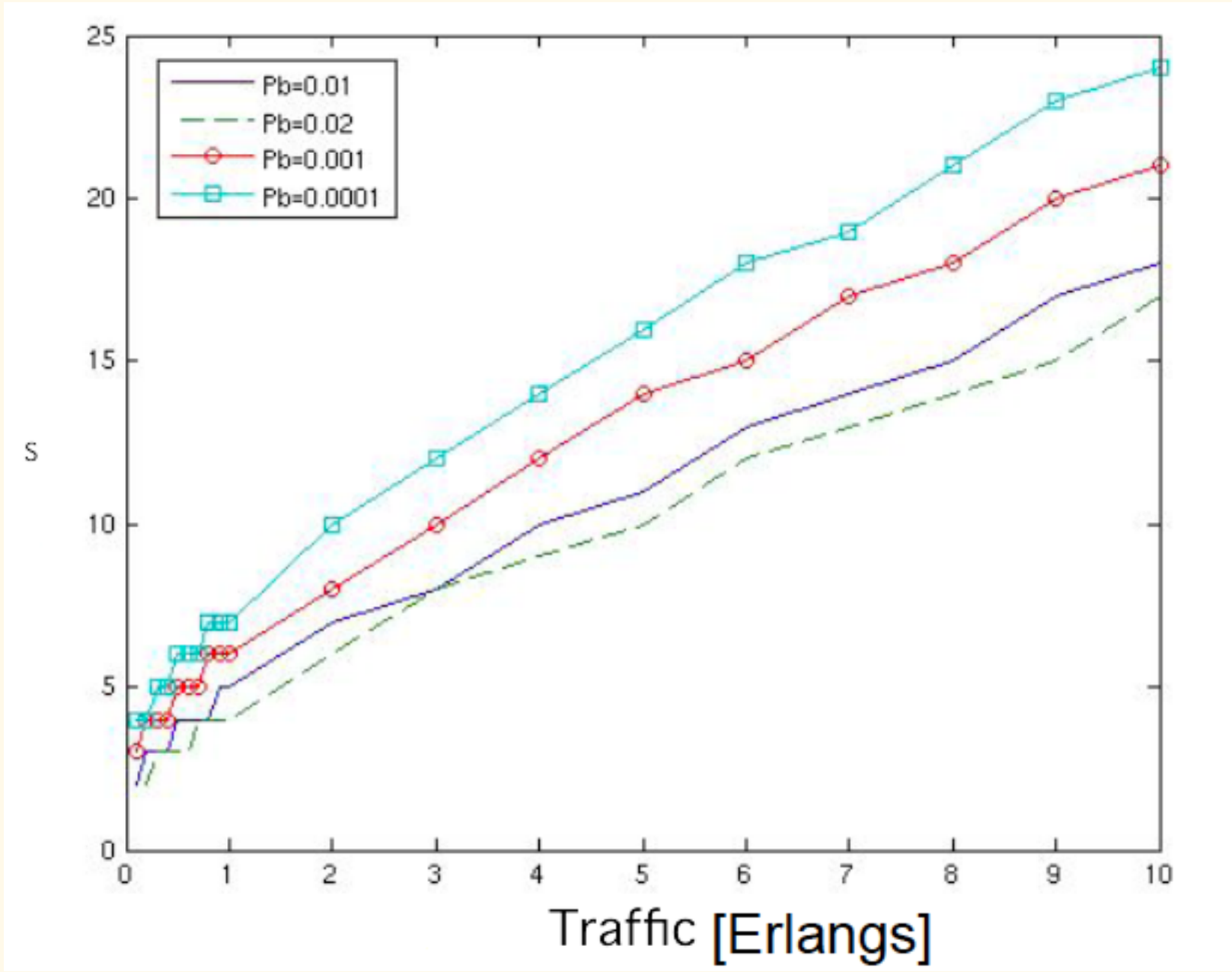
- Blocking probability is given by :

$$P_b = p(S) = \frac{\rho^S/S!}{\sum_{k=0}^{S} \frac{\rho^k}{k!}}$$



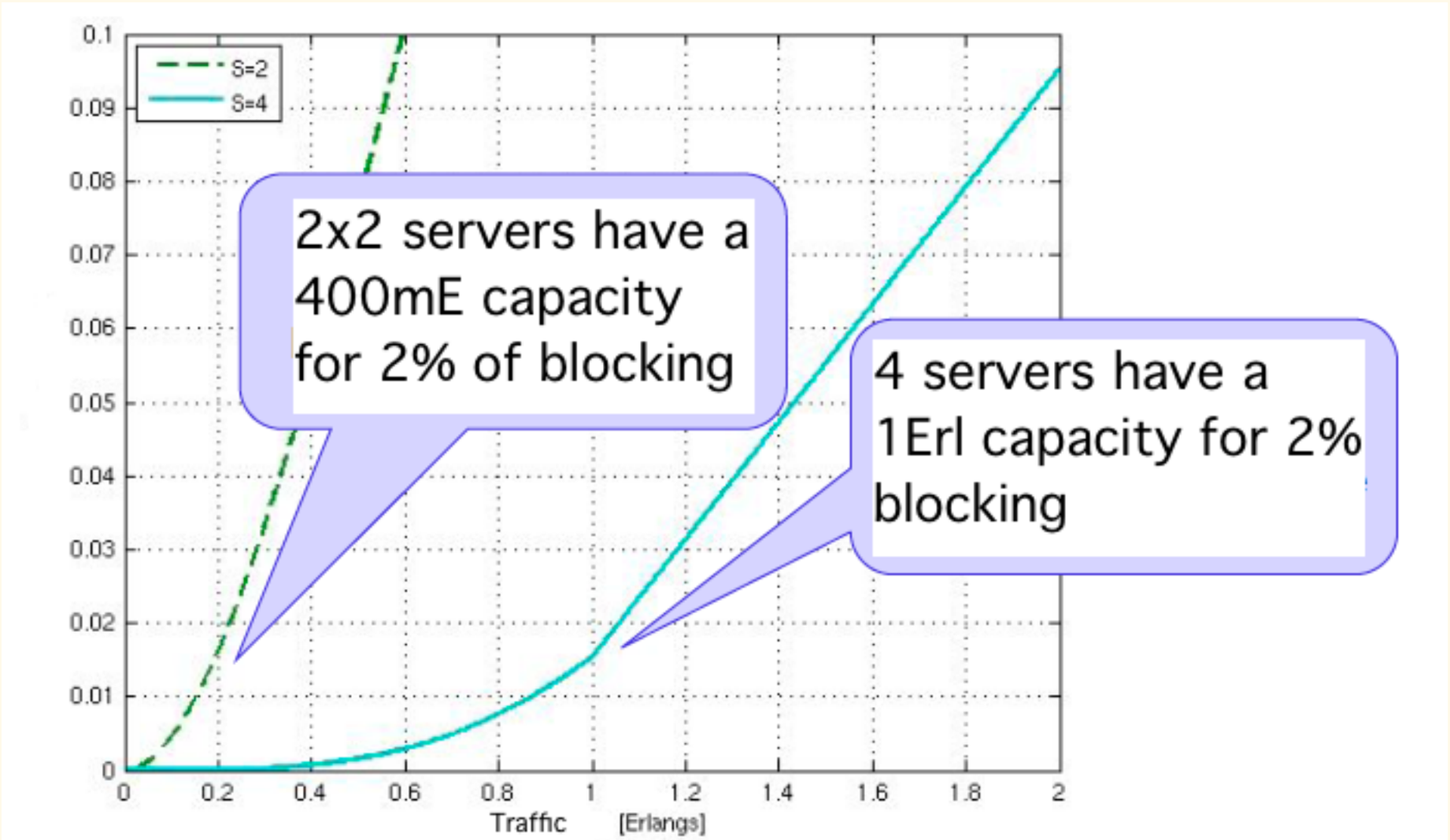
Erlang B and C laws

Erlang B law is not linear:



Erlang B and C laws

Erlang B law is not linear:



A recursive definition of the blocking probability:

$$P_b(S+1) = \frac{\rho P_b(S)}{S+1+\rho P_b(S)}.$$

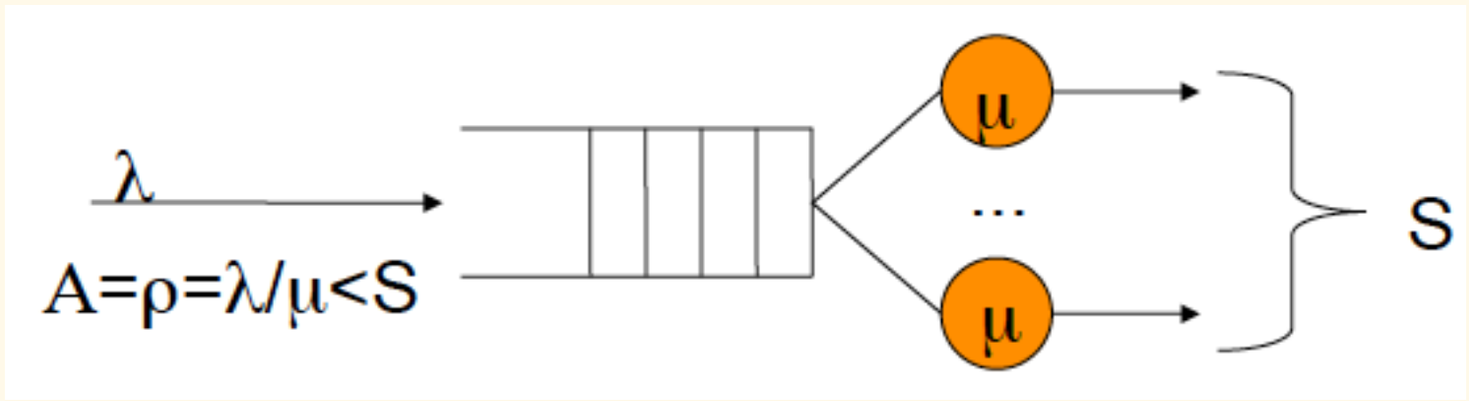
An approximation of the inverse of the Erlang B law (Claude Rigault ENST) :

If $E_{1,S}(A) = \epsilon = 10^{-k}$ then $S \approx A + k\sqrt{A}$

- Example: consider a traffic of $A = 100$ Erlangs
- What is the number of junctions to install in order that the probability of lost call is less than $\epsilon = 10^{-4}$?
- $N \approx 140$ servers

Erlang C model, assumptions

- We consider S servers serving phone calls
- Calls arrive according to a Poisson law of rate λ, λ does not depend on the number of occupied servers (infinite population assumption)
- Calls that cannot be served are put in an infinite buffer, discipline is FIFO
- Service time is distributed according to a decreasing exponential law of rate μ



Erlang C, markov chain

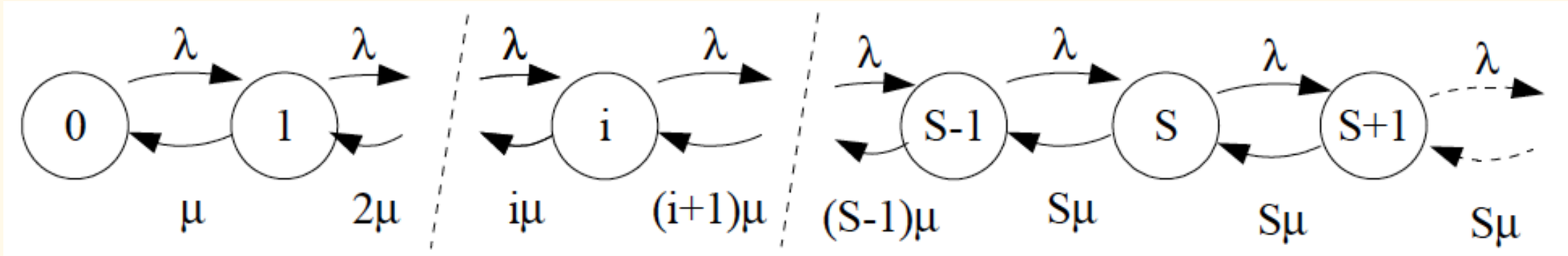
- The system is modeled with a M/M/S/∞.

- Stationary probabilities are given by :

$$p(n) = \rho^n/n!p(0) \text{ for } 0 \leq n \leq S-1,$$

$$p(n) = \rho^n/(S!S^{n-S})p(0) \text{ for } n \geq S,$$

$$p(0) = \left(\sum_{k=0}^{S-1} \rho^k/k! + \frac{\rho^S/S!}{1-\rho/S} \right)^{-1} \text{ with } \rho = \lambda/\mu$$



Erlang C, waiting probability

$$
\begin{aligned}
P_w &= \sum_{n \geq S} p(n) \\
&= \sum_{n > S} \frac{\rho^S}{S!} \frac{\rho^{n-S}}{S^{n-S}} p(0) \\
&= \frac{\rho^S}{S!} \frac{S}{S-\rho} p(0) \\
&= \frac{\frac{\rho^S}{S!} \frac{S}{S-\rho}}{\sum_{k=0}^{S-1} \rho^k/k! + \frac{\rho^S/S!}{1-\rho/S}}.
\end{aligned}
$$

Erlang C, some useful results

- Mean number of customers:

$$Q = P_w \frac{\rho/S}{1-\rho/S}$$

- Mean waiting time

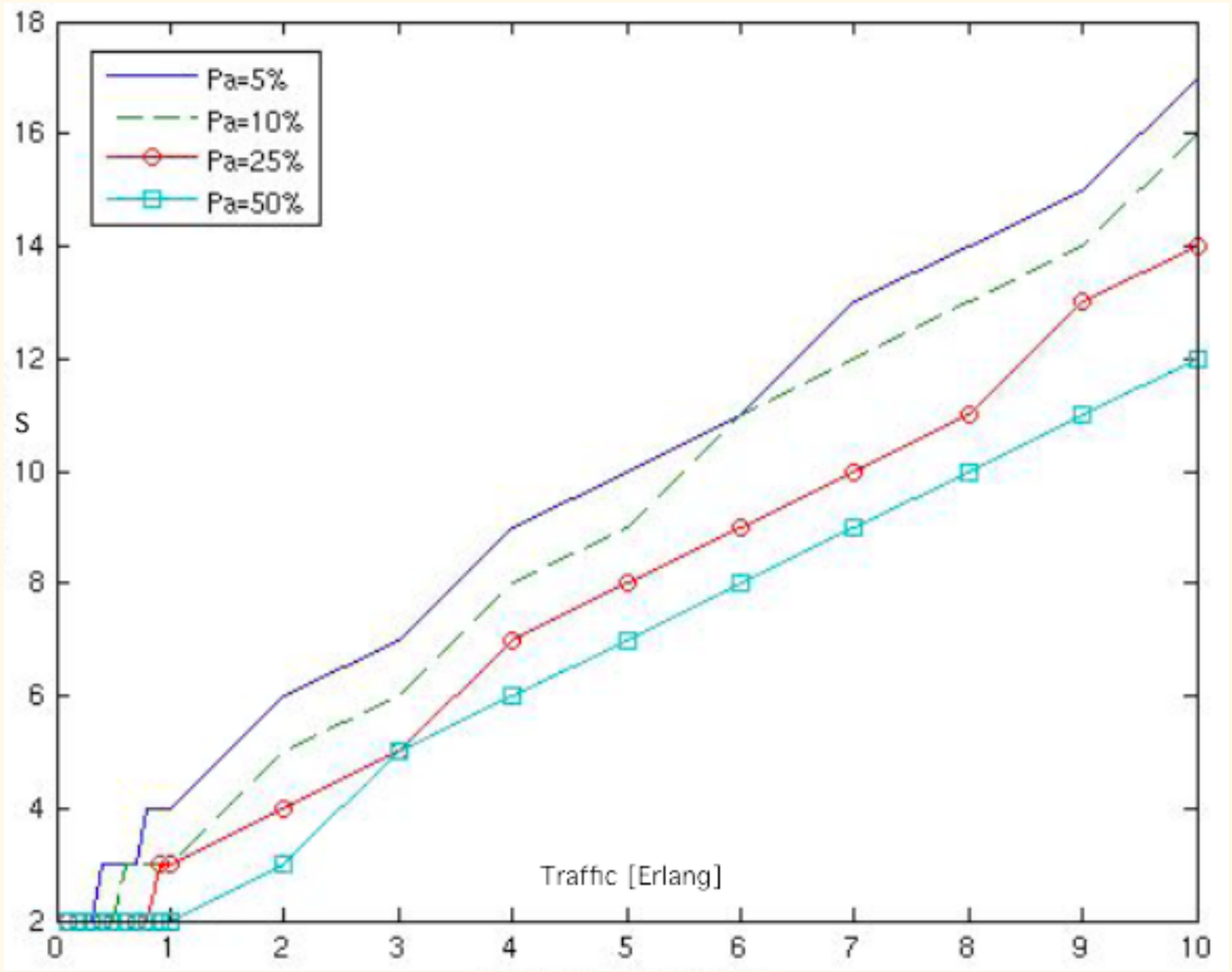
$$W = \frac{P_w}{\lambda} \frac{\rho/S}{1-\rho/S}$$

- Distribution of the waiting time:

$$P[T > t] = P_w \exp(-S\mu t(1-\rho/S))$$

Erlang B and C laws

Erlang C law is not linear:



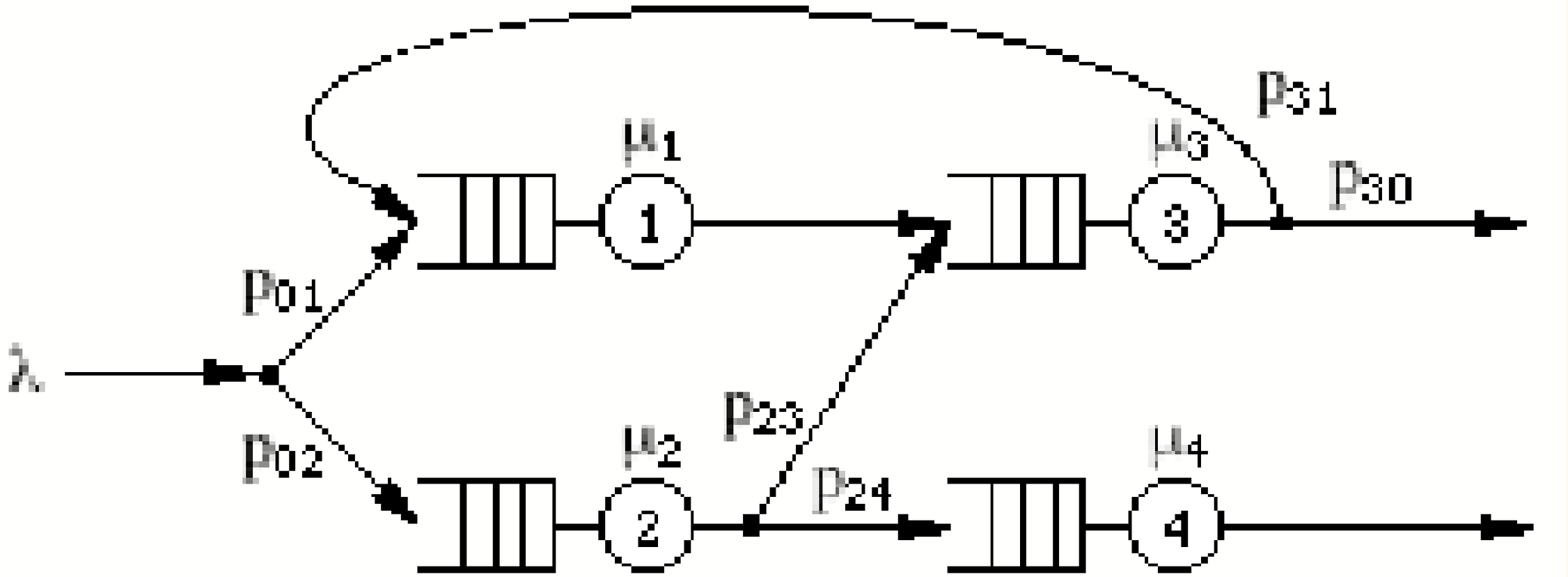
1. Definition of the Jackson networks
2. Stability
3. Visit rates
4. Analysis and Jackson theorem
5. Performance parameters

Definition of the Jackson networks

- Open network
- M stations
- A single class of customers
- Poisson arrivals of rate λ
- For each station: a single server, an unlimited FIFO buffer, an exponential service time of rate μ_i
- Probabilistic routing: p_{0i}=probability that a customers arriving in the network goes to station i; p_{ij}=probability that a customer leaving station i goes to station j; p_{i0}=probability that a customer leaving station i goes out of the system

$$\forall i,\ 0 \leq i \leq M,\ \sum_{j=0}^{M} p_{ij} = 1$$

Example of Jackson network



Stability

- Let e_i the average number of times that a customer visits station i during its journey in the system

- e_i is the visit rate of station i

- Arrival rate of customers in the network: λ

- Arrival rate of customers in station i: $\lambda_i = e_i \lambda$

- Stability condition:

$$\forall i = 1, ..., M,\ \lambda_i < \mu_i$$

Visit rates

- The traffic λ_i at station i is made of :
- (1) the outside traffic: λp_{0i}
- (2) the traffic coming from station j: $\lambda_j p_{ji}$ for all stations $j = 1, ..., M$

$$\lambda_i = \lambda p_{0i} + \sum_{j=1}^{M} \lambda_j p_{ji}$$

- With $\lambda_i = e_i \lambda$:

$$e_i = p_{0i} + \sum_{j=1}^{M} e_j p_{ji}$$

Analysis

- State description: $\{n(t) = [n_1(t), ..., n_M(t)]\}$
- Stochastic process with continuous time and discrete state space
- At a given instant, there is $n_i(t)$ customers in station i
- - arrivals are Poisson $\rightarrow$ memoryless
- - service times are exponential $\rightarrow$ memoryless
- - routing is probabilistic $\rightarrow$ memoryless

Jackson theorem

- For a stable Jackson network:

$$p(n) = \prod_{i=1}^{M} p_i(n_i)$$

- $p_i(n_i)$ is the steady-state probability of a M/M/1 with arrival rate λ_i and service rate μ_i

- $p_i(n_i) = (1 - \rho_i)\rho_i^{n_i}$

- with $\rho_i = \lambda_i/\mu_i$

Performance parameters

- For station i:

$$X_i = \lambda_i \; Q_i = \frac{\rho_i}{1-\rho_i} \; R_i = \frac{Q_i}{X_i} = \frac{1}{\mu_i - \lambda_i} \; U_i = \rho_i$$

- For the network:

$$X = \lambda \; Q = \sum_{i=1}^{M} Q_i$$

$$R = \frac{Q}{X} = \frac{Q}{\lambda} = \sum_{i=1}^{M} \frac{Q_i}{\lambda} = \sum_{i=1}^{M} e_i \frac{Q_i}{\lambda_i} = \sum_{i=1}^{M} e_i R_i$$