TELECOM ParisTech

Institut Mines-Télécom

Networks performance evaluation

Queueing Theory Introduction

Claude Chaudet

Claude.Chaudet@enst.fr

Queueing theory

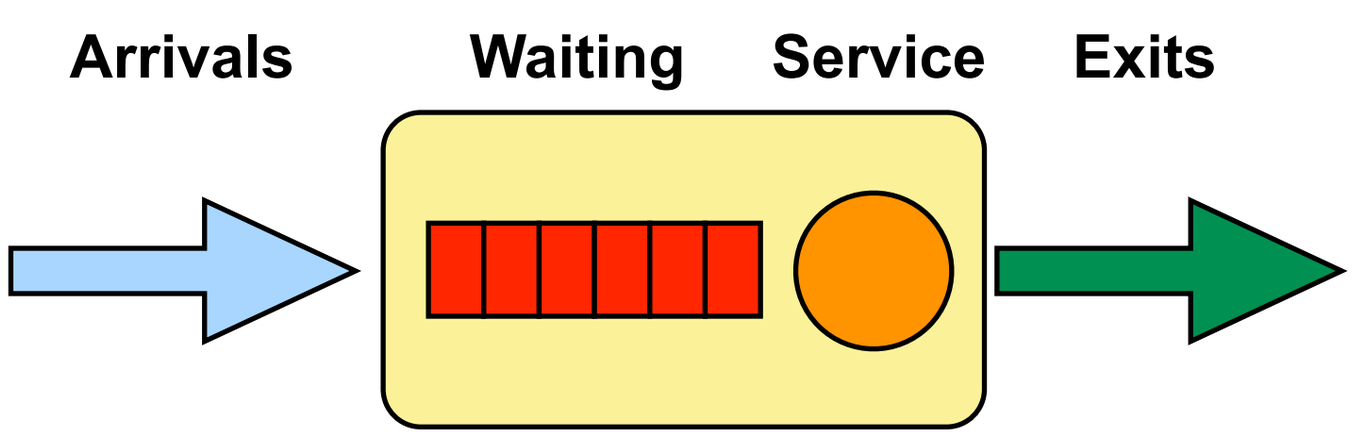
- **A mathematical model for access to a shared resource**
  - Networks / telecom examples: router buffer, telephone lines
  - Everyday life: road networks, supermarket counters

- **Purpose: estimate certain parameter values…**
  - Waiting time
  - Size of filling level of a waiting line
  - Rejection probability (saturated system, ...)
- **... in function of some system parameters**
  - Number of demands/load
  - Processing speed
  - Size/organization of the waiting line

General model

General process

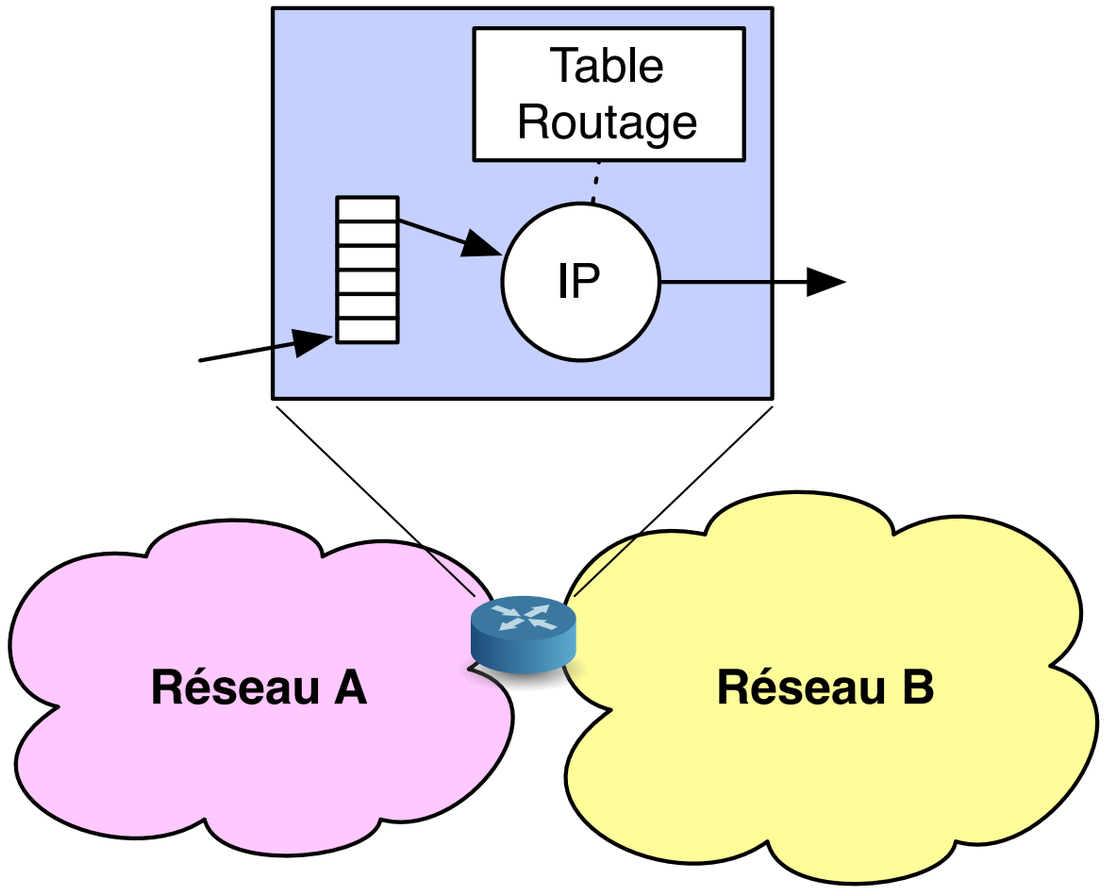
- Clients **arrive** un a system — modeled as a random process
- Clients **wait** (waiting room, buffer, ...)
- Clients **are served** by the system — random duration
- Clients **exit the system**



- We are interested in the system behavior and in its influence on the clients flow (how is the incoming flow transformed in the exit flow)

Example: network interconnection device

- **A packet can arrive while the CPU processes another one**
  - How much time will each packet wait, on average?
  - How much memory is necessary for a 100 Gb/s router with a CPU capable of processing packets in 1 ms (on average)?
    - Which table size can we allow to avoid delaying packets too much, knowing that search happens in O(log n) time?

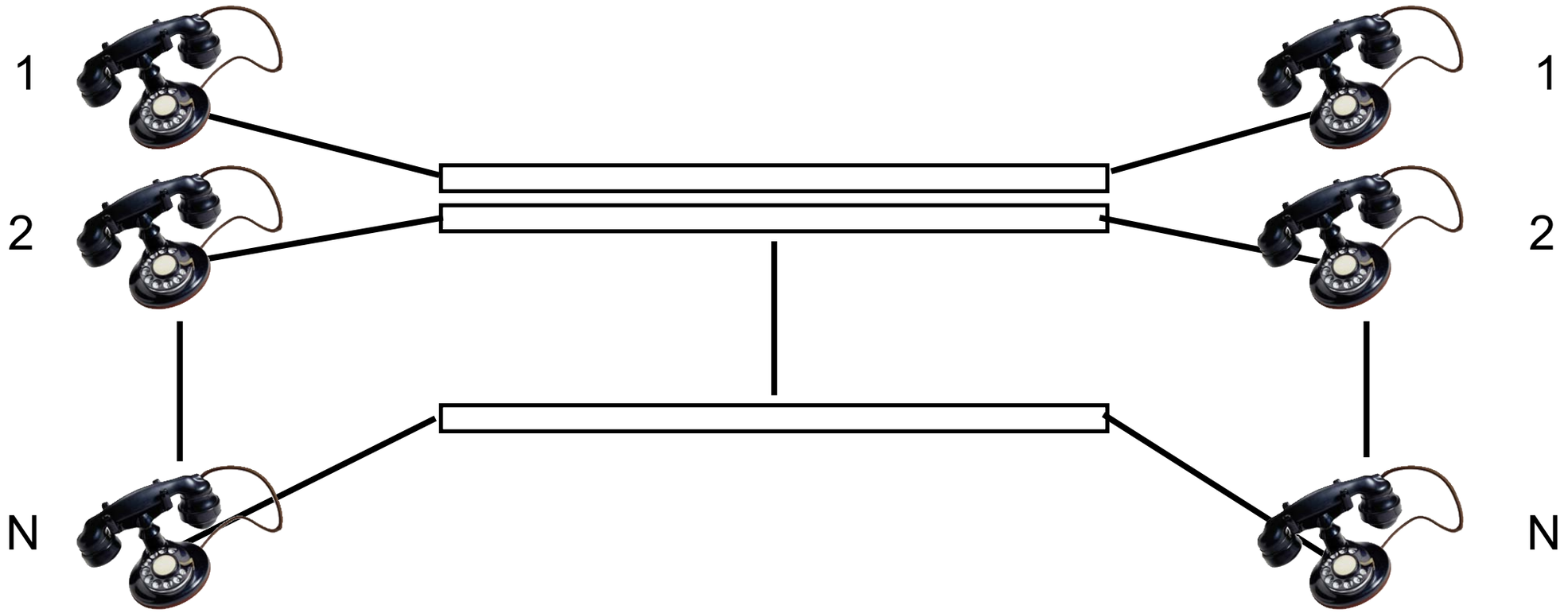


Example: supermarket

- **In a supermarket, there are several counters:**
  - Is it better to have one or multiple queueing lines?
    - Based on which criterion (average waiting time; max waiting time? ; ...)
  - How many people are necessary to have a waiting time inferior to a given value?

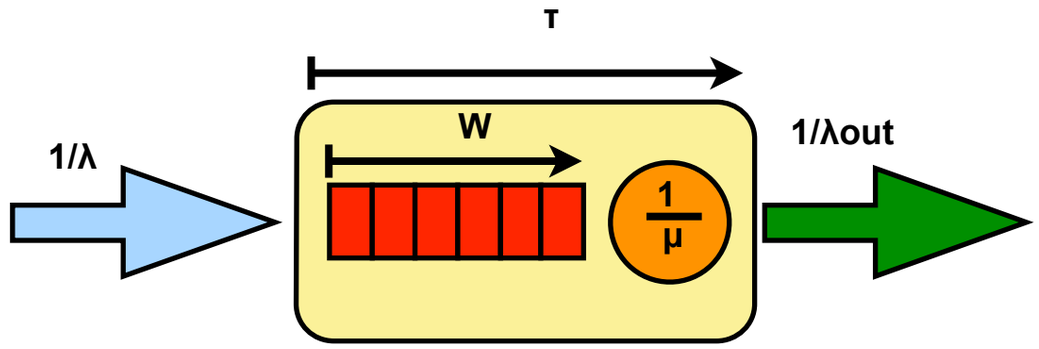
Example: telephone network

- **How many lines are necessary to interconnect N users?**
  - Too many: high cost for the operator
  - Too few: bad quality of service (rejection probability high)



The model

- **Problem data:**
  - Arrival process (inter-arrivals distribution)
    - parameters (average, variance,...) are known
  - Service process (distribution of service time)
    - parameters (average, variance,...) are known



- **We want to characterize how the queue behaves**
  - Average number of clients in the queue — Q
  - Average waiting time — W
  - Average sojourn time (total time passed in the system) — T
  - Probability to find a full/empty queue for an arbitrary client
  - Exit process in function of the entry process and the queue behavior

Different models: Kendall Taxonomy

- **Queueing systems are classified in classes**

- **Class name: T/X/C/K/m/Z**
  - T : Inter-arrivals process
  - X : Service process
  - C : Number of servers
  - K : Queue length (including servers ; optional, default = +∞)
  - m : population (optional, default = +∞)
  - Z : queueing discipline (optional, default = FIFO)

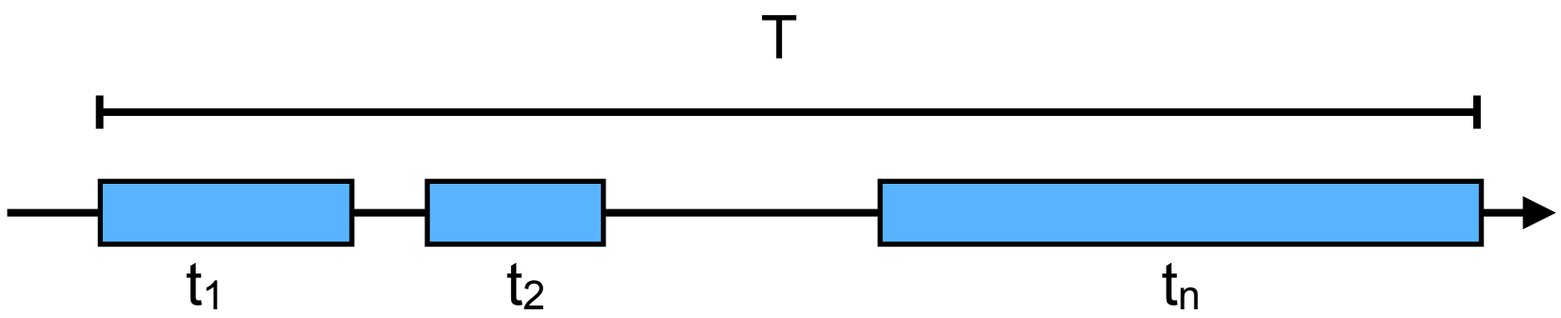
T / X : examples	
M	Exponential (Markov)
G	General (arbitrary)
D	Deterministic (constant)
H	Hyperexponential
E	Erlang (sum exponential)

- **Exemples**
  - M/M/1 ; M/M/C ; M/M/C/C
  - M/G/1 ; G/M/1

Z : examples	
FIFO	classical queue
LIFO	stack
PS	processor sharing
RANDOM	random selection

Definitions: traffic and load

- ***Traffic* represents the occupation of a resource (server, communication link, …)**
  - Traffic on a resource is the occupation proportion (or occupation probability) of this resource: $\alpha \in [0 ; 1]$



$$a = \frac{\sum_i t_i}{T}$$

- **The traffic unit is an "Erlang", named after the Danish engineer (1917)**
  - One erlang represents a 100% occupation
  - Convention: On N servers, traffic varies between 0 and N Erlang

Definitions: traffic and load (2)

- **Traffic represent the ratio between arrivals intensity and the service speed**
  - $\alpha = \lambda / \mu$ with λ arrivals rate (client / s)
    μ service rate (avg. service time = $1/\mu$)

- **The *load* is the ratio between intensity of the arrivals and the *global* service rate of the system**
  - $\rho = \lambda / (m.\mu)$ for *m* servers
  - $\rho = \alpha$ for a single server

Definitions: system stability

- **If the arrival rate is greater than the service capacity, the system cannot process the requests**
  - Problem is solved; the queue is overloaded, it is qualified of **unstable**
  - We are only interested in stable systems

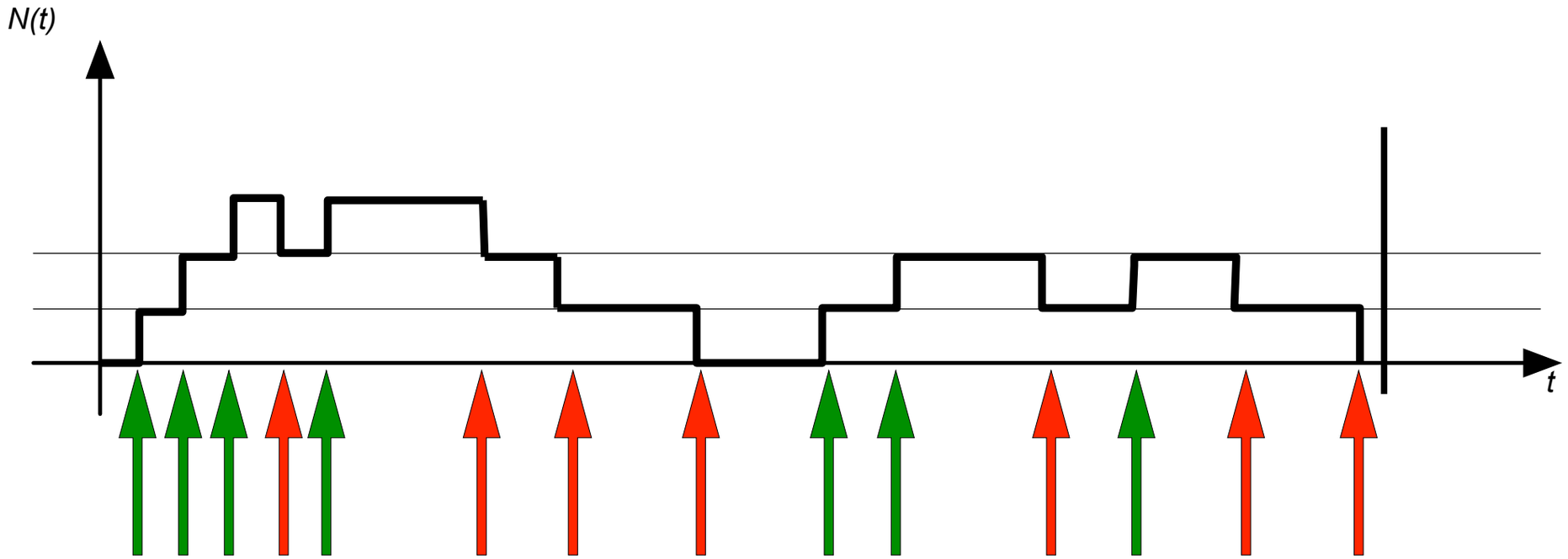
- **If there is no clients creation or destruction within the system, the system is said to be *stable* iff:**
  - Clients do not arrive faster than the system can process
  - ⇔ ρ < 1 : load is (strictly) inferior to 1
  - ⇔ λ < μ : for a single server ; λ < m.μ for *m* servers

- **In a stable system, $\lambda = \lambda_{out}$**

Queueing systems analysis

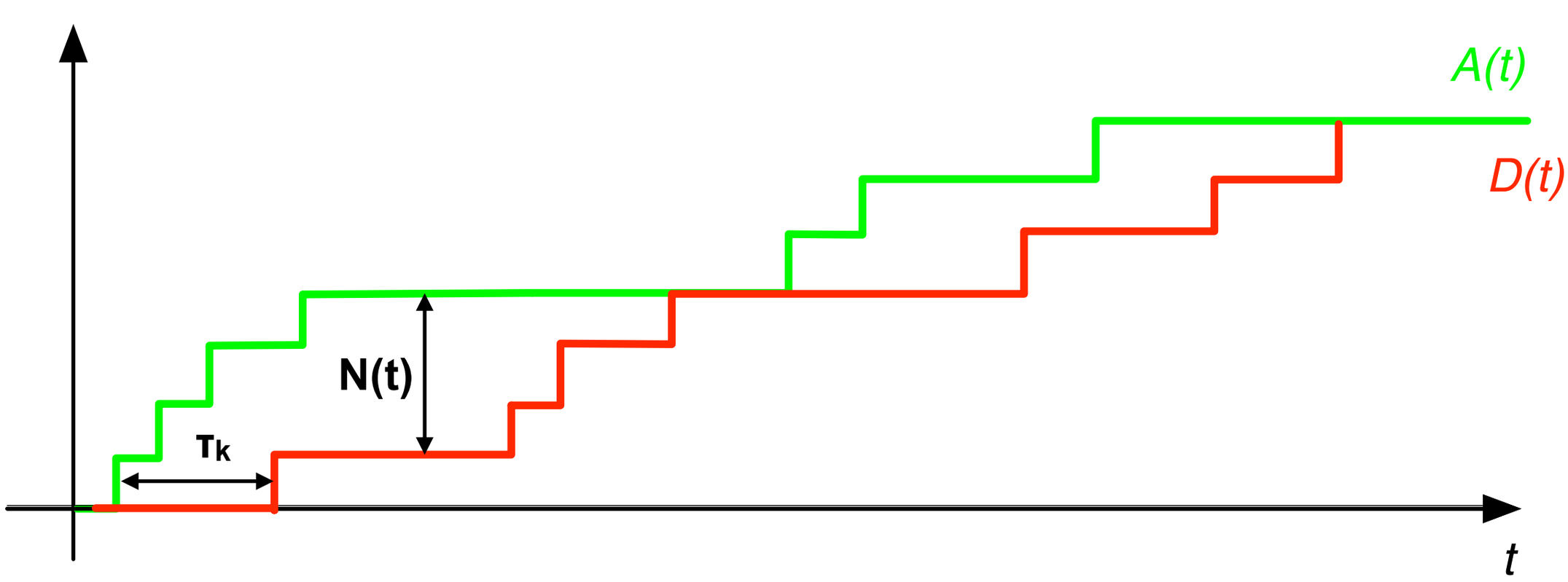
Evolution of the number of clients in queue

- **Observing the system, it is possible to draw the number of clients in the queue.**
  - Log arrivals and departures times



System usage

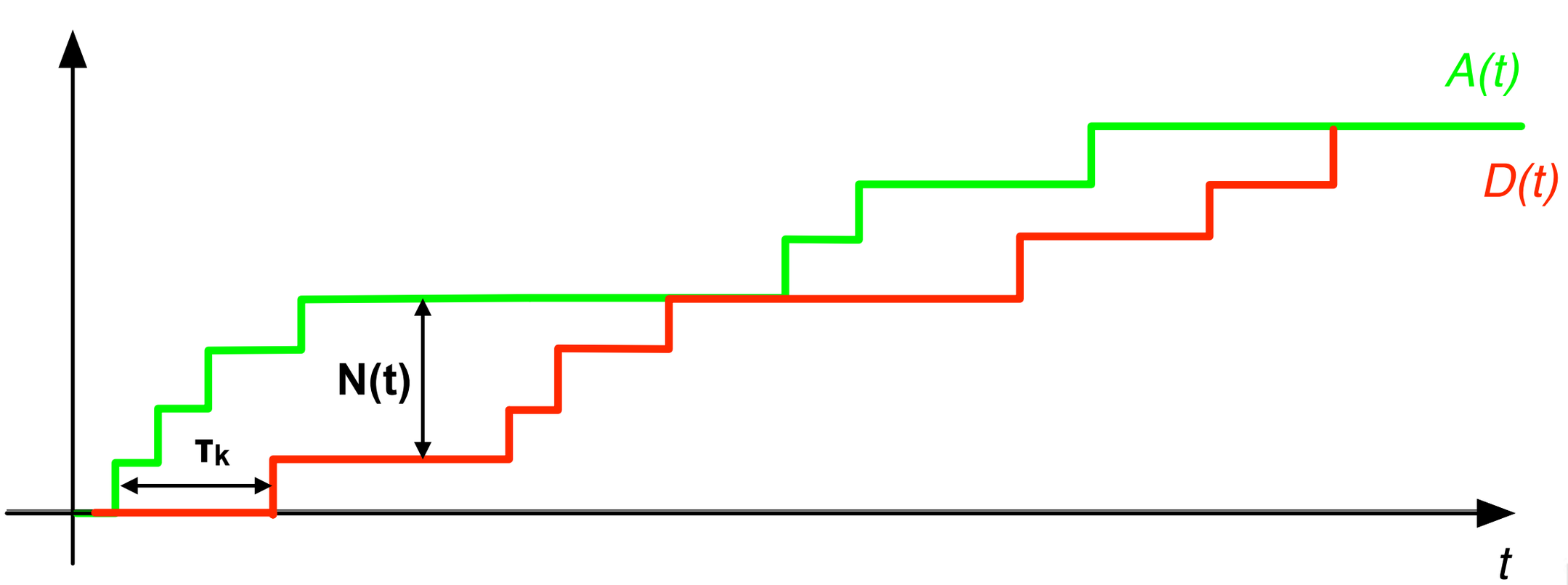
- **Comparing the arrival process A(t) and the departure process D(t), we can see:**
  - Services times (under FIFO discipline) : τ_k
  - Number of clients in the system at every moment : N(t)



System usage (2)

- **The area between both curves (S) can be calculated by two different ways:**

$$S = \int_{t=0}^{T} \left(A(t) - D(t)\right).dt = \int_{t=0}^{T} N(t).dt \qquad \text{and} \qquad S = \sum_{k=0}^{A(T)} t_k$$



System usage (3)

- **Both expressions are equal :**

$$\int_{t=0}^{T} N(t).dt = \sum_{k=0}^{A(T)} \tau_k$$

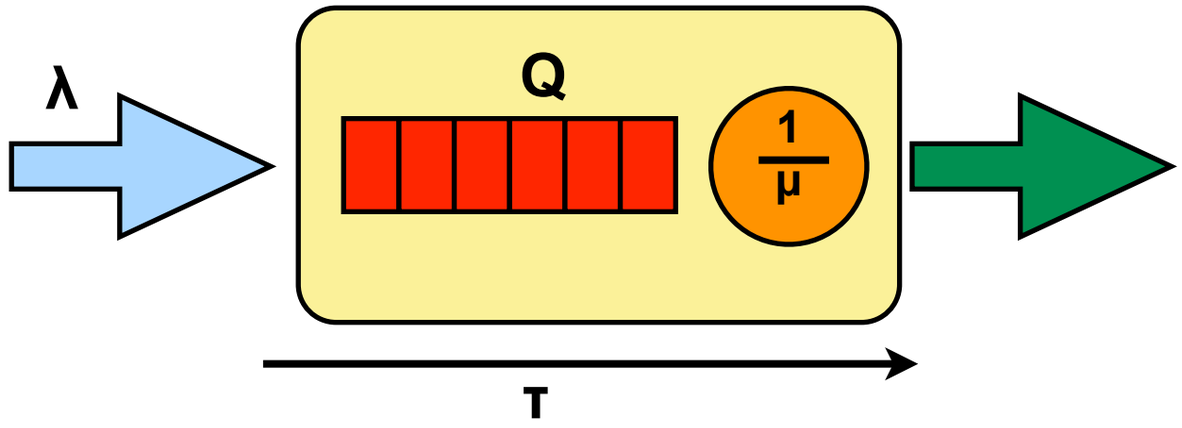
- **Therefore :**

$$\frac{\int_{t=0}^{T} N(t).dt}{T} = \frac{\sum_{k=0}^{A(T)} \tau_k}{T} = \frac{\sum_{k=0}^{A(T)} \tau_k}{N(T)} \cdot \frac{N(T)}{T}$$

$\bar{N}(T)$ $\bar{\tau}(T)$ $\bar{\lambda}(T)$

Little's Formula

- **Assumptions:**
  - The systems is stable
  - No creation or destruction of clients within the system



- **Little's formula :** $\boxed{Q = \tau \cdot \lambda}$
  - Q = average number of clients in the system
  - τ = average sojourn time
  - λ = throughput (incoming or outgoing)

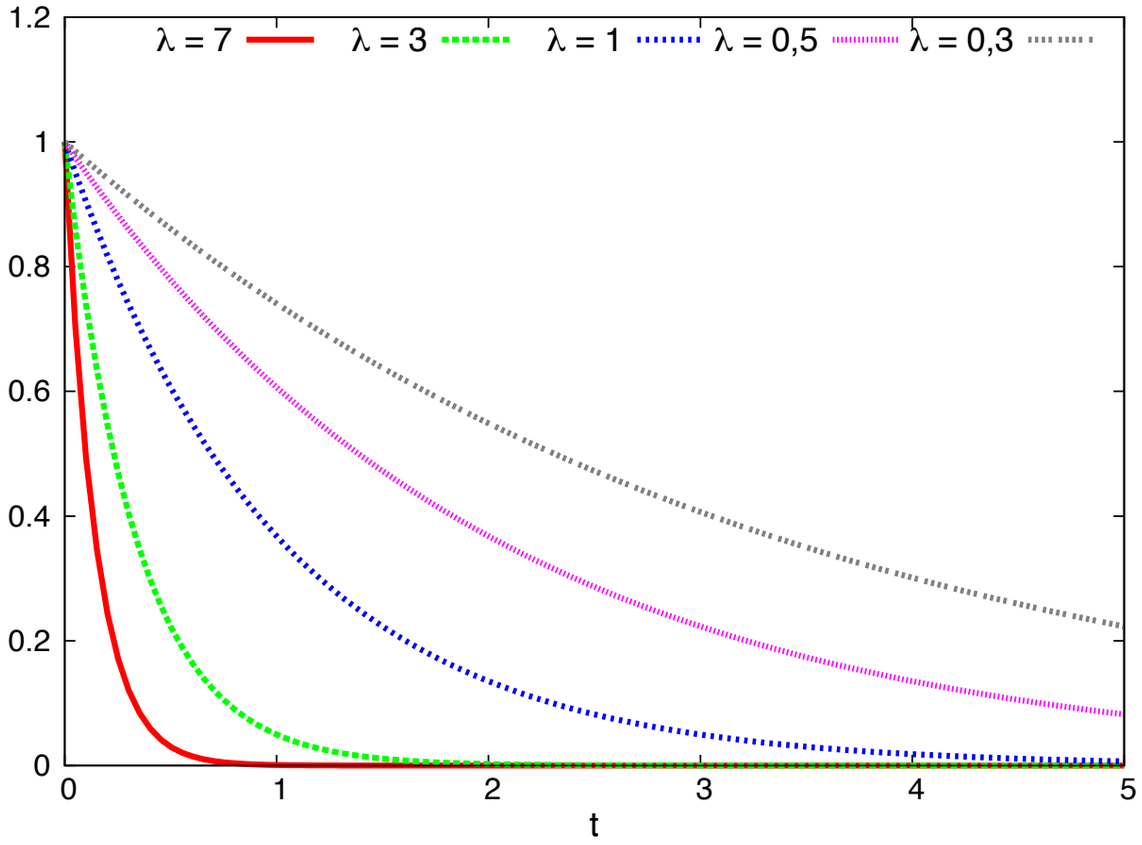
- **Does not depend on the arrival process / service time distribution**

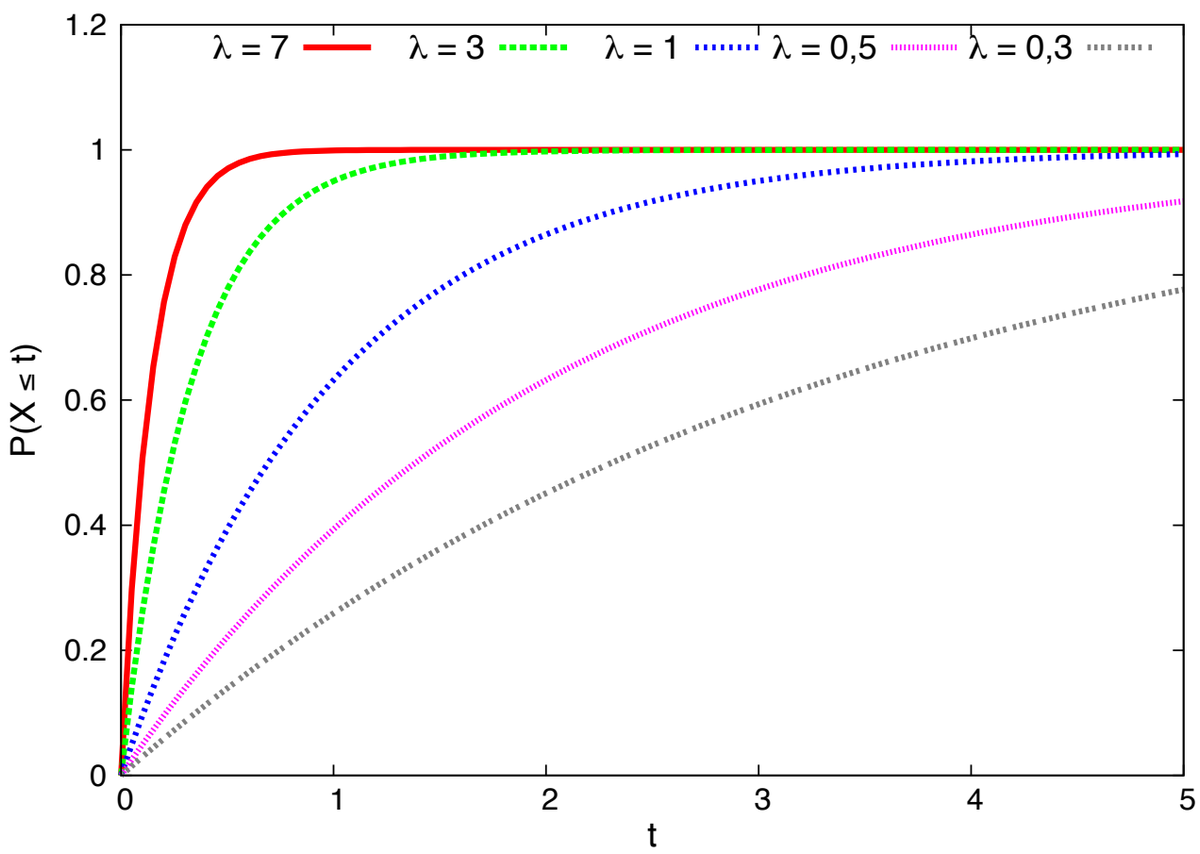
Exponential random variables
Poisson processes

Exponential law: definition

- **A random variable follows an exponential law, with parameter λ if its cumulative distribution function (CDF) is:**

$$P(X > t) = e^{-\lambda.t} \quad \Leftrightarrow \quad P(X \leq t) = 1 - e^{-\lambda.t}$$

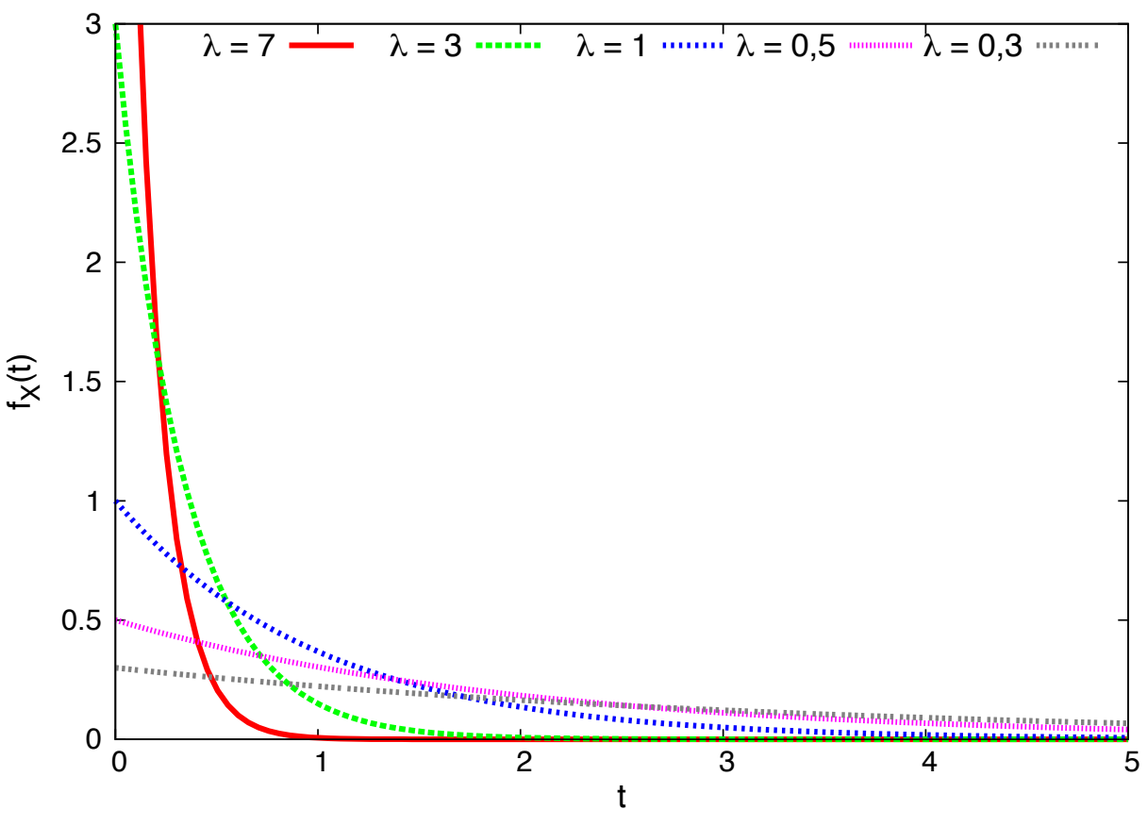


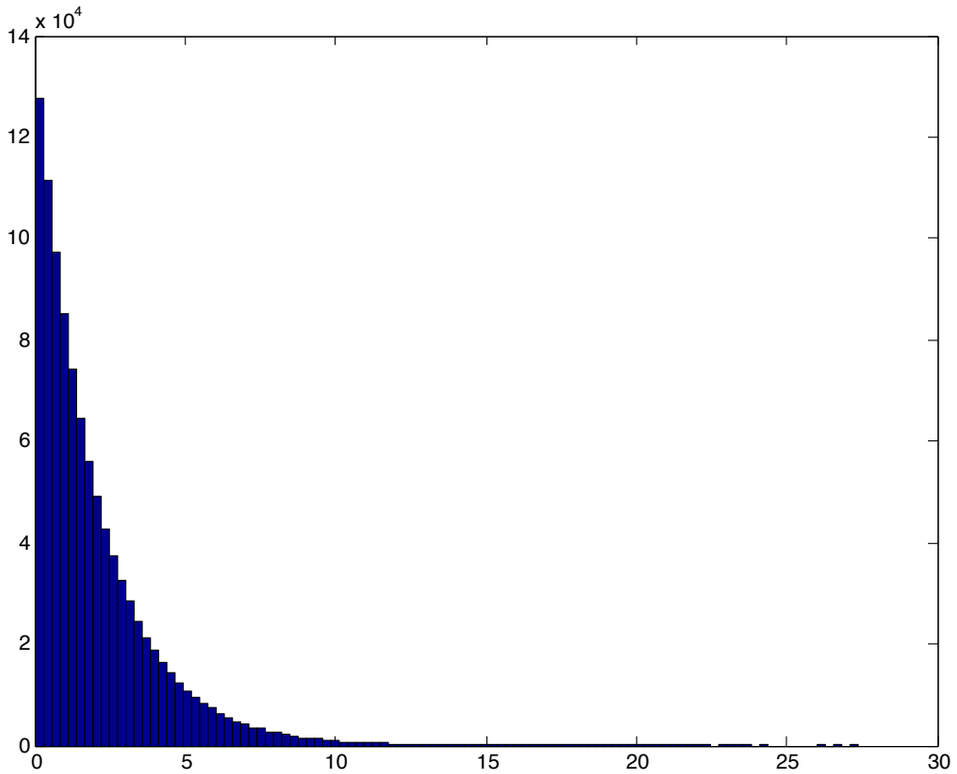


Exponential law: probability density function

- **The probability density function (PDF) is the derivative of the CDF:**

$$f_X(t) = \frac{d\left(P(X \leq t)\right)}{dt} = \lambda.e^{-\lambda.t}$$





Exponential law: average and variance

- **Average (expected value) :**

$$E(X) = \int_0^{+\infty} t.\left(\lambda.e^{-\lambda.t}\right) dt = \frac{1}{\lambda}$$

- **Variance :**

$$var(X) = \left(\int_0^{+\infty} t^2.\left(\lambda.e^{-\lambda.t}\right) dt\right) - E(X)^2 = \frac{1}{\lambda^2}$$

- **Standard deviation:**

$$\sigma = \sqrt{var(X)} = \frac{1}{\lambda}$$

Exponential law: fundamental properties

- **Exponential law is memory-less:**

$$P(X > s + t | X > s) = P(X > t)$$

- **Proof**
  - Using the conditional probabilities definition: $P(A \cap B) = P(A|B).P(B)$

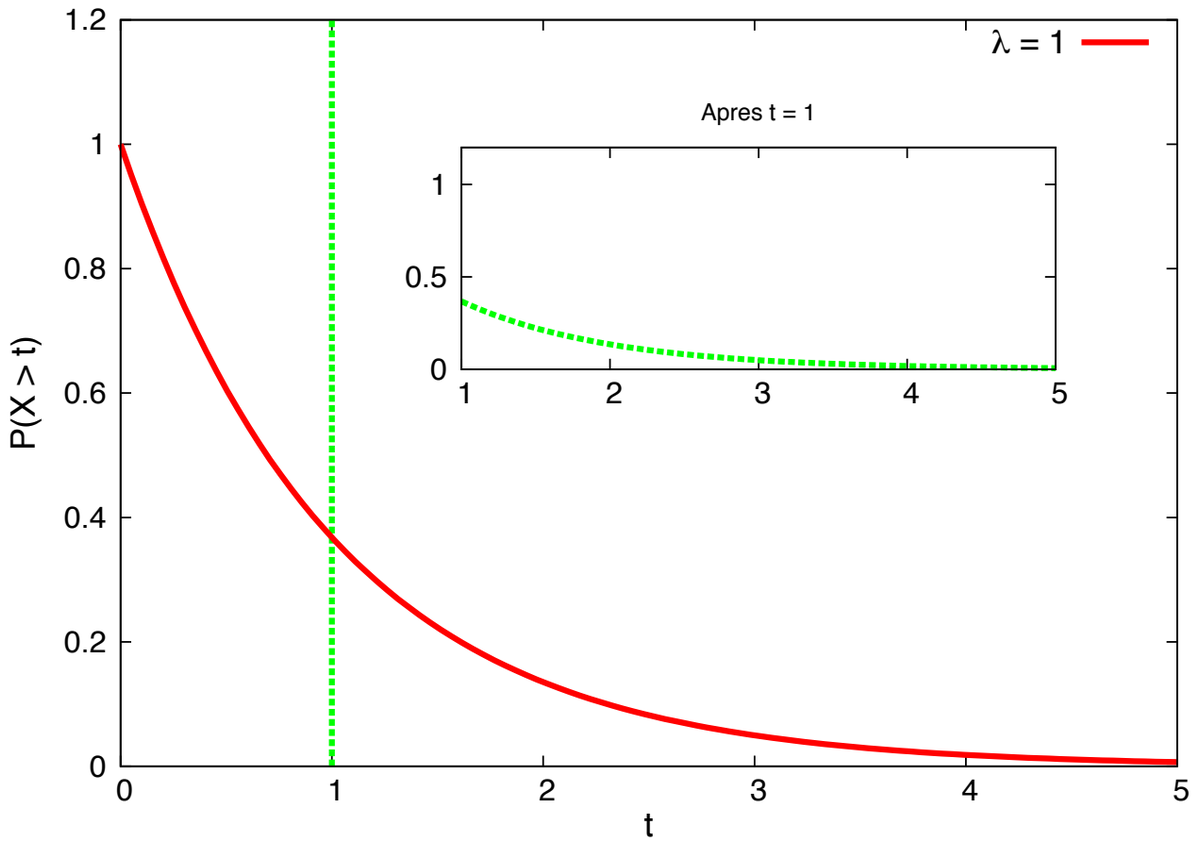
$$P(X > s + t | X > s) = \frac{P\left((X > s + t) \wedge (X > s)\right)}{P(X > s)} = P(X > s + t)$$

$$= \frac{e^{-\lambda.(s+t)}}{e^{-\lambda.s}} \qquad \text{(CDF)}$$

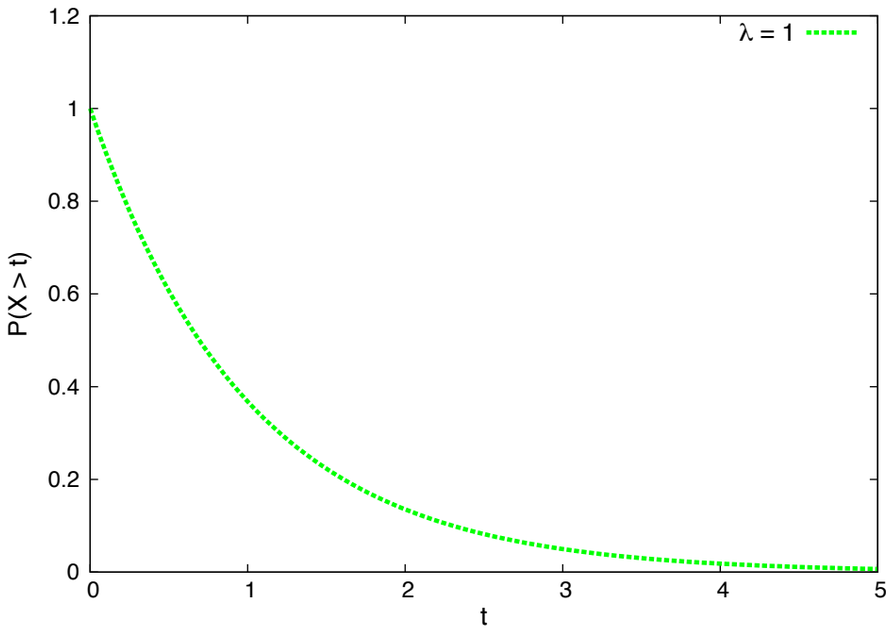
$$= e^{-\lambda.t} = P(X > t) \qquad \text{(exponential function property)}$$

Exponential law: memory-less property

- **What it means:**
  - Observe a phenomenon whose occurrence follows an exponential law
  - The time passed observing has no effect on the probability that the event occurs after 1s, 2s, …



Normalization



In practice

- **We observe a phenomenon and make the assumption that it can be characterized by an exponential law**

- **Steps to verify the hypothesis**
  - The frequency histogram should look like a geometric distribution
    - Be careful, the shape of the histogram does not provide any proof, just intuition

  - Compute average and variance of the samples
    - If exponential law, average = $1/\lambda$; variance = $1/\lambda^2$

  - We check the hypothesis with a **statistical test**
    - Pearson's X^2 test, Kolmogorov-Smirnov test, ...

Kolmogorov-Smirnov fitting test

- **Samples (measured values) : {x_1, x_2, ..., x_n}**
- **We build the empirical CDF that corresponds to our samples**
  - $F_n(x) = \frac{Card(\{i|x_i \leq x\})}{n}$
- **We then consider the theoretical CDF:**
  - $F(x) = P(X \leq x) = 1 - e^{-\lambda x}$
- **We then study the function | F - F_n |**
  - Find its maximum value and compute the following indicator:

$$D_n = \sqrt{n}.\sup_{x \in \mathbb{R}} |F_n(x) - F(x)|$$

  - Compare the Dn value with Kolmogorov table

Kolmogorov table

Expected certainty level

Nb samples

Maximum D_n value

	10 %	5 %	1 %
1	0,95	0.9750	0.9950
2	0,7764	0.8419	0.9293
3	0,636	0.7076	0.8290
4	0,5652	0.6239	0.7342
5	0,5095	0.5633	0.6685
6	0,468	0.5193	0.6166
7	0,4361	0.4834	0.5758
8	0,4096	0.4543	0.5418
9	0,3875	0.4300	0.5133
10	0,3697	0.4092	0.4889
11	0,3524	0.3912	0.4677
12	0,3381	0.3754	0.4491
13	0,3255	0.3614	0.4325
14	0,3142	0.3489	0.4176
15	0,304	0.3376	0.4042
16	0,2947	0.3273	0.3920
17	0,2863	0.3180	0.3809
18	0,2785	0.3094	0.3706
19	0,2714	0.3014	0.3612
20	0,2647	0.2941	0.3524
21	0,2586	0.2872	0.3443
22	0,2528	0.2809	0.3367
23	0,2475	0.2749	0.3295
24	0,2424	0.2693	0.3229
25	0,2377	0.2640	0.3166

	10 %	5 %	1 %
25	0,2377	0.2640	0.3166
30	0,2176	0.2417	0.2899
35	0,2019	0.2242	0.2690
40	0,1891	0.2101	0.2521
45	0,1786	0.1984	0.2380
50	0,1696	0.1884	0.2260
60	0,1551	0.1723	0.2067
70	0,1438	0.1598	0.1917
80	0,1347	0.1496	0.1795
90	0,1271	0.1412	0.1694
100	0,1207	0.1340	0.1608
n>100	1,223 / √n	1,358 / √n	1,629 / √n

Poisson Process

- **Counting process**
  - Probability that k events occur in a time interval T:

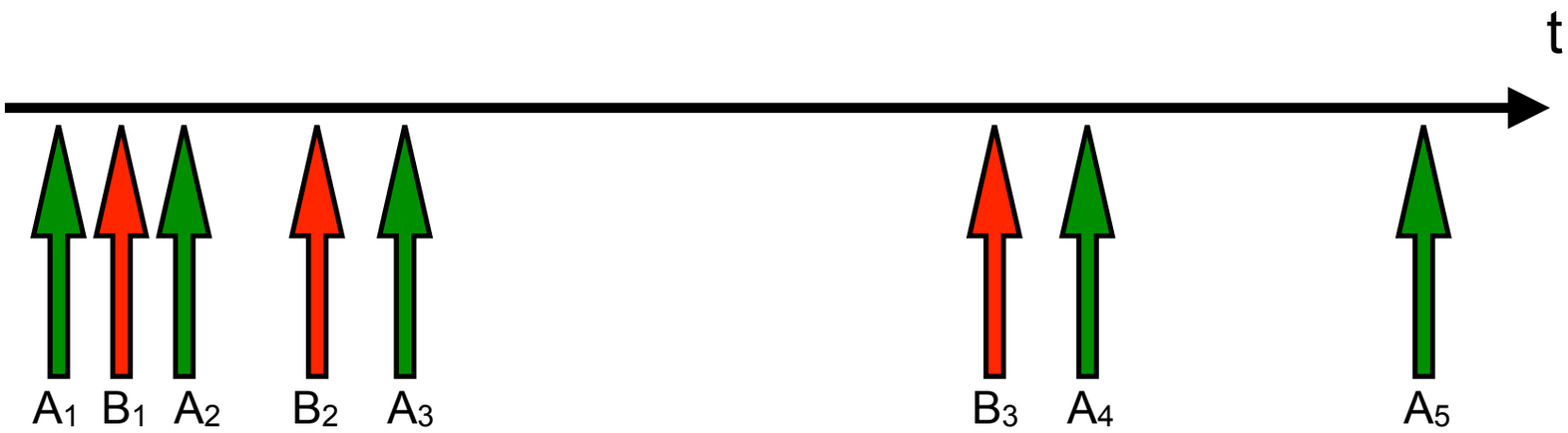
$$P[N_T = k] = \frac{(\lambda \cdot T)^k}{k!} \cdot e^{-\lambda \cdot T}$$

- $P[N_{T+dt} = k+j \mid N_T = k]$ = $\lambda.dt + o(dt)$ if $j = 1$
  = $o(dt)$ if $j > 1$
  = $1-\lambda.dt + o(dt)$ if $j < 1$

  - The occurrence of more than one event in an infinitesimal time interval is negligible.

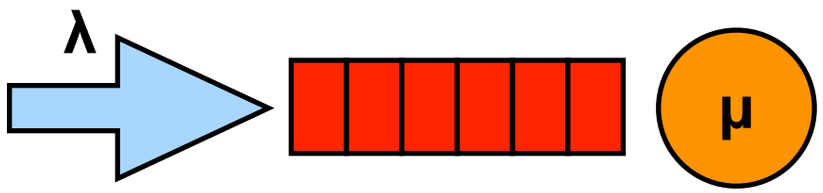
Poisson processes - properties

- **Inter-event times ($X_i = A_{i+1} - A_i$) follow an exponential law**
  - Time between two events: $1 - e^{-\lambda t}$



- **Superposition of two Poisson processes with parameters λ_1 and λ_2 is a Poisson process with parameter $\lambda_1 + \lambda_2$**

The M/M/1 queue



Arrivals: Poisson process (Parameter λ)

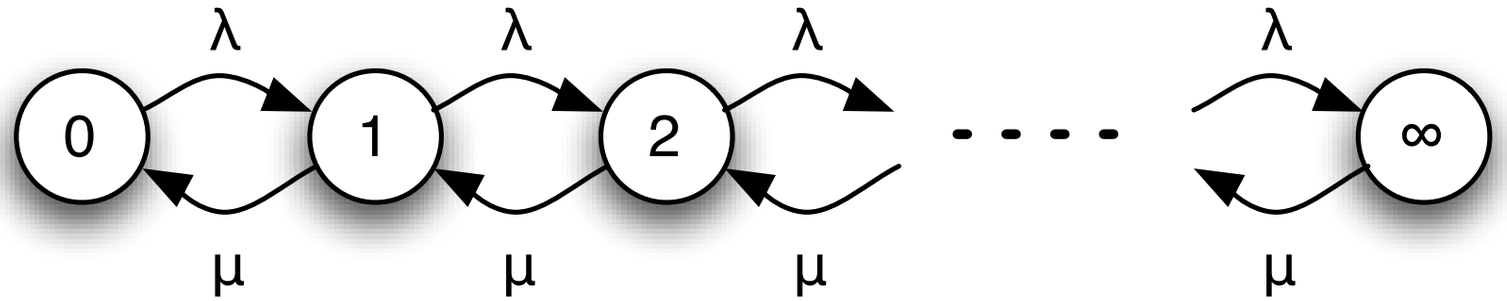
Service time : exponential (Parameter μ)

Single server

FIFO queue ; infinite length

M/M/1 queue model

- **To solve the problem, we examine how the number of clients in the system evolves**
  - Automaton with an infinite number of states (0, 1, 2, ..., ∞ clients in the system)
  - The system passes from state to state with a probability that depends on the arrival rate / service rate



- **We want to calculate the probability that the system functions in each state (i.e. probability that it contains *k* clients)**
  - We compute the expected number of clients from these probabilitis

- **This representation is called a Markov process**

Markov Chains and Markov Processes

What is a Markov chain/process?

- **A mathematical formalism to represent the random behavior of a system.**
  - Used for performance calculation

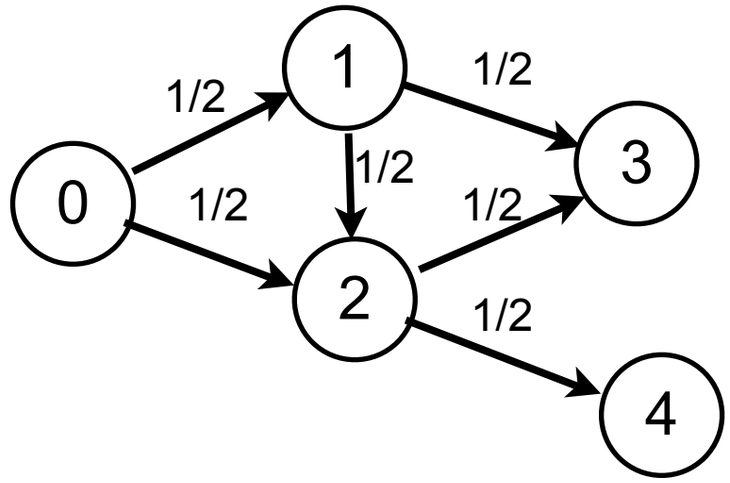
- **The system is represented as an automat**
  - List all potential states of the system
  - Compute probabilities to pass from one state all others

- **Applies to discrete event systems**
  - The states form a countable set (can be infinite)

- **Two types of Markov processes**
  - Discrete time (Markov chain)
  - Continuous time (Markov process)

A toy example: heads or tails

- **Play two times "heads or tails"**
- **Compute a score**
  - Heads => score + 1
  - Tails => score +2



- **What is the probability to have a score equal to k after the two rounds?**

Heads - Heads	2
Heads - Tails	3
Tails- Heads	3
Tails-Tails	4

Score = 2	0.25
Score = 3	0.5
Score = 4	0.25

Markov Chains

- **Stochastic system evolution**
  - The system passes from state to state at arbitrary moments
  - Time is counted in *number of steps*

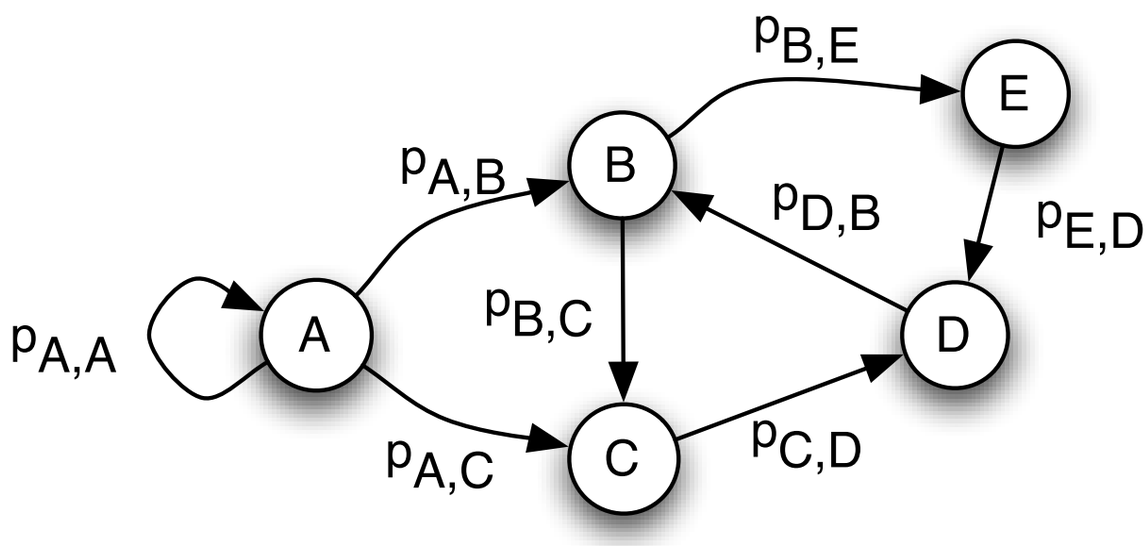
- **Transition probability between two states: probability to arrive in a state *knowing that we leave the current state*.**
  - Probability to remain in the same state is not necessarily equal to 0
  - Important property: the path is memory-less.

$$P\left(X_n = j | X_{n-1} = i_{n-1} \left(\wedge X_{n-2} = i_{n-2} \wedge \ldots \right)\right) = P\left(X_n = j | X_{n-1} = i_{n-1}\right)$$

- **Property: homogeneity**
  - A Markov chain is *homogeneous* if $P(X_n = j)$ does not depend on *n*
    - *i.e.* the transition probabilities do not change over time
  - We consider only homogeneous chains

Graphical and matrix representations

- **Transition graph**
  - Oriented graph
  - We do not represent zero probabilities
  - Probabilities to remain in the same state are sometimes not represented
    - 1 - sum of others



- **Transition matrix**
  - probability to pass from one state to the other

$$P = \begin{pmatrix} p_{A,A} & p_{A,B} & p_{A,C} & p_{A,D} & p_{A,E} \\ p_{B,A} & p_{B,B} & p_{B,C} & p_{B,D} & p_{B,E} \\ p_{C,A} & p_{C,B} & p_{C,C} & p_{C,D} & p_{C,E} \\ p_{D,A} & p_{D,B} & p_{D,C} & p_{D,D} & p_{D,E} \\ p_{E,A} & p_{E,B} & p_{E,C} & p_{E,D} & p_{E,E} \end{pmatrix}$$

Transient probabilities

- **We want to represent the probabilities of evolution of the system**
  - What is the probability that after *n* steps we arrive in state *i* ?

- **We define the probabilities vector after *n* steps:**
  - $\pi^{(n)} = (\pi_1{}^{(n)}, \pi_1{}^{(n), \ldots})$

- **It is computed from:**
  - The initial probabilities vector : $\pi^{(0)}$
  - The transition matrix : P

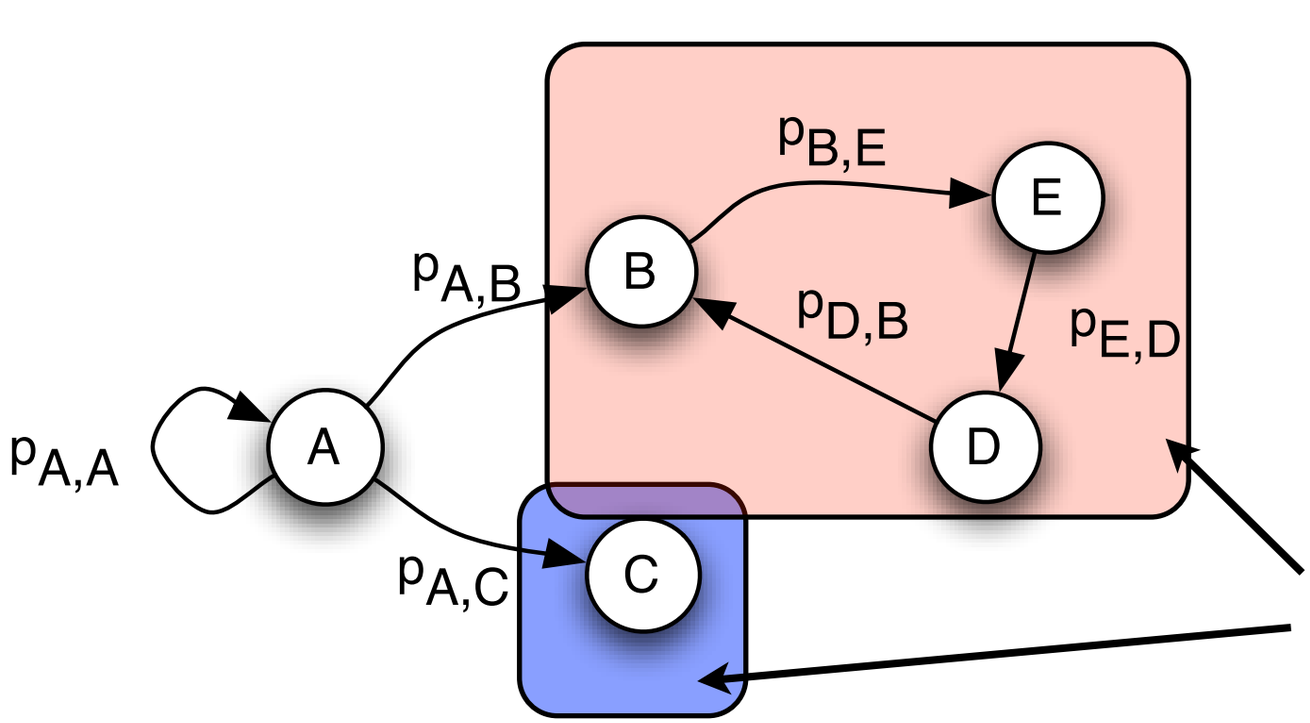
- **Iteratively:**
  - $\pi^{(n)} = \pi^{(n-1)}.P$
  - $\Rightarrow \pi^{(n)} = \pi^{(0)}.P^n$

Properties: Irreducibility

- **The system can evolve from any state to any other state in a finite number of steps**

$$\forall (i,j), \exists m \geq 1 \text{ tq } p_{i,j}^{(m)} \neq 0$$

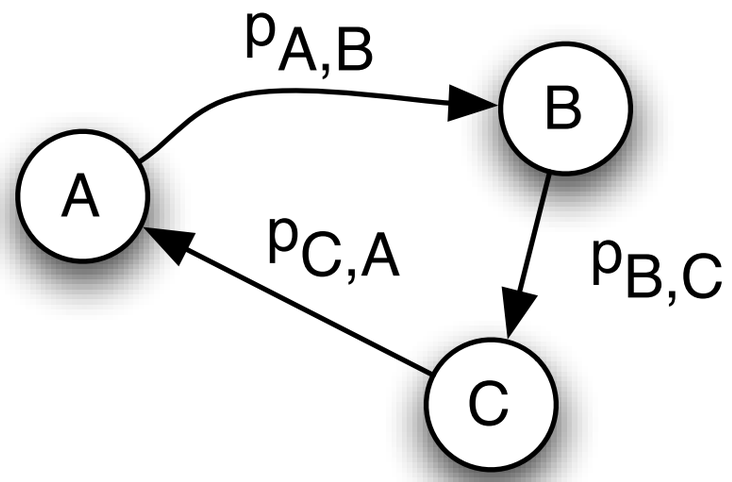
Probability to pass from state *i* to state *j* in *m* transitions



There are two subset of states which are absorbing => the Markov chain is not irreducible

States classification: periodicity

- **A state is said to be *periodical* (period *k*) si :**
  - ∀m non-multiple of k, $p_{j,j}^{(m)} = 0$



- **The period of a Markov chain is the GCD of all its states periods**
  - It is also the GCD of the circuits lengths in the transition graph
  - If there is a loop (∃i s.t. $p_{i,i} \neq 0$), the chain is *non-periodical*

States classification: transient states

- Let us represent by $f_{i,i}^{(m)}$ the probability that the first return to a state *i* happens *m* steps after leaving it

- The probability to come back to a state after leaving it is:

$$f_{i,i} = \sum_{m=0}^{+\infty} f_{i,i}^{(m)}$$

- The average number of steps necessary to come back is:

$$M_i = \sum_{m=0}^{+\infty} m.f_{i,i}^{(m)}$$

- A state *i* is said to be:
  - Transient if : $f_{i,i} < 1$
  - Recurrent null if : $f_{i,i} = 1$ and $M_i = +\infty$
  - Recurrent non-null if : $f_{i,i} = 1$ and $M_i < +\infty$

States classification: transient states

- **If the Markov Chain is irreducible:**
  - All states have the same nature (transient, recurrent null or recurrent non-null)
  - If states are periodical, all states have the same period

- **If the Markov Chain is irreducible and finite:**
  - All states are recurrent non-null

Stationary distribution

- **We want to compute the limit behavior, when m→+∞**
  - Does the limit probabilities vector exist?
  - Is it unique?
  - What is its expression?

$$\pi = \lim_{m \to +\infty} \pi^{(m)}$$

- **Fundamental theorem: a Markov chain that is irreducible and non-periodical has one and only one stationary probability vector, that does not depend on the initial state.**

- **To compute π :**
  - π is a fixed point: $\pi = \pi.P$
  - Sum of probabilities is equal to 1 : $\sum_i \pi_i = 1$

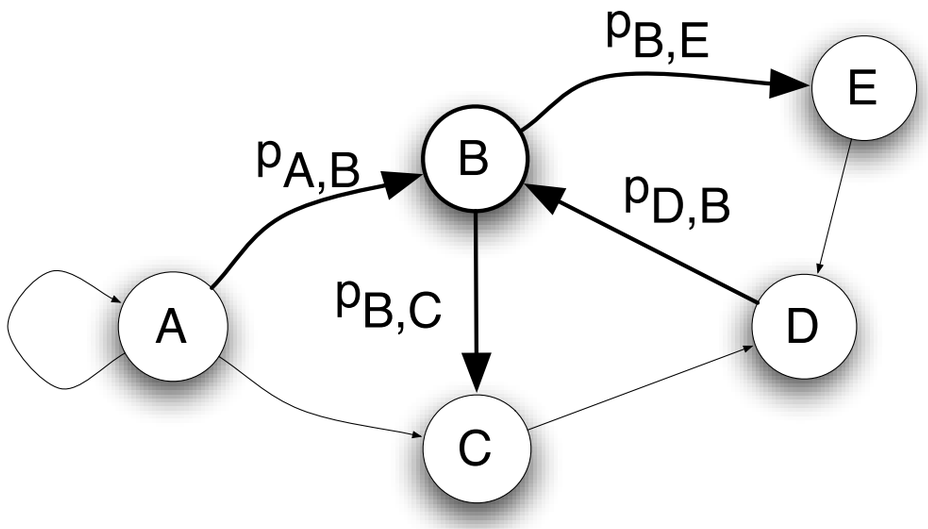
Resolution method #1

Another resolution method

Flow conservation equations

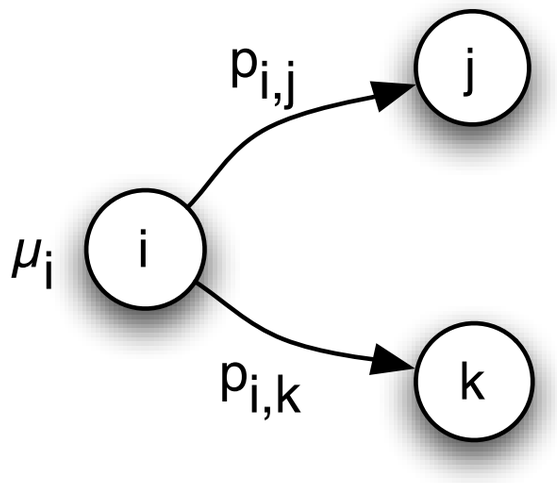
- For each state, the incoming flow is equal to the outgoing flow

$$\left.\begin{aligned} \pi_j &= \sum \pi_i \cdot p_{i,j} \\ \sum p_{j,i} &= 1 \end{aligned}\right\} \quad \Rightarrow \sum \pi_i \cdot p_{i,j} = \sum \pi_j \cdot p_{j,i}$$



From Markov Chains to Markov Processes

- **Markov processes are the continuous time version of Markov Chains:**
  - There is no notion of *step* anymore
  - We spend a certain time in each state *i*, distributed exponentially (parameter μ_i)
  - Transition probabilities (p_{ij}) define the possible arrival states *when leaving a state*
    - We suppose $p_{ii} = 0$



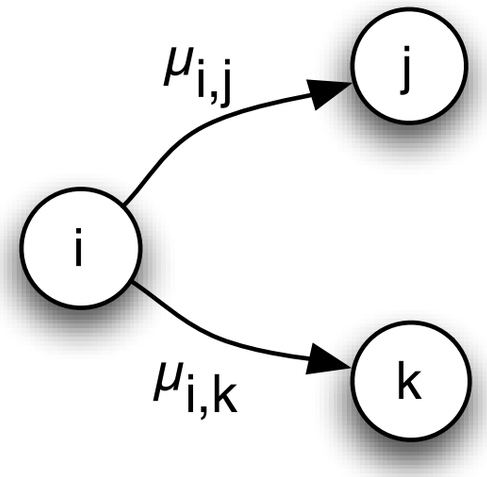
Markov Processes: system evolution

- **Probability to leave state *i* between t and t+dt:**
  - $$P\left(X(t+dt) \neq i | X(t) = i\right) = 1 - e^{-\mu_i.dt}$$
    $$= 1 - (1 - \mu_i.dt + o(dt))$$
    $$= \mu_i.dt + o(dt)$$

- **Thus, the probability, during time dt, to go from state *i* to *j*:**

$$p_{i,j}(dt) = P\left(X(t+dt) = j | X(t) = i\right)$$
$$= \left(\mu_i.dt + o(dt)\right).p_{i,j}$$
$$= \mu_i.p_{i,j}.dt + o(dt)$$

$\mu_{i,j}$



Markov Processes: infinitesimal generator

Infinitesimal generator

- Equivalent to the transition probability matrix in discrete-time
- Matrix: Q = ($q_{i,j}$) s.t.:
  - $\forall i \neq j, q_{i,j} = \mu_{i,j}$
  - $\forall i, q_{i,i} = -\sum_{j \neq i} \mu_{i,j} = -\mu_i$

Suppose that $p_{i,i} = 0$
Hence $\Sigma\ p_{i,j} = 1$

Example:

$$Q = \begin{pmatrix} -\mu_1 & \mu_{1,2} & \mu_{1,3} \\ \mu_{2,1} & -\mu_2 & \mu_{2,3} \\ \mu_{3,1} & \mu_{3,2} & -\mu_3 \end{pmatrix}$$

Markov Processes: system evolution

- The system evolution is characterized by the following differential equation:

$$\frac{d\pi(t)}{dt} = \pi(t).Q$$

- Who admits the following solution:

$$\pi(t) = e^{Q.t}$$

Finding the stationary distribution

- In an irreducible recurrent not null CTMC, π is the only solution of the system.
- As the function converges, the derivate is null at the limit:

$$\forall i, \sum \pi_i . q_{i,j} = 0$$

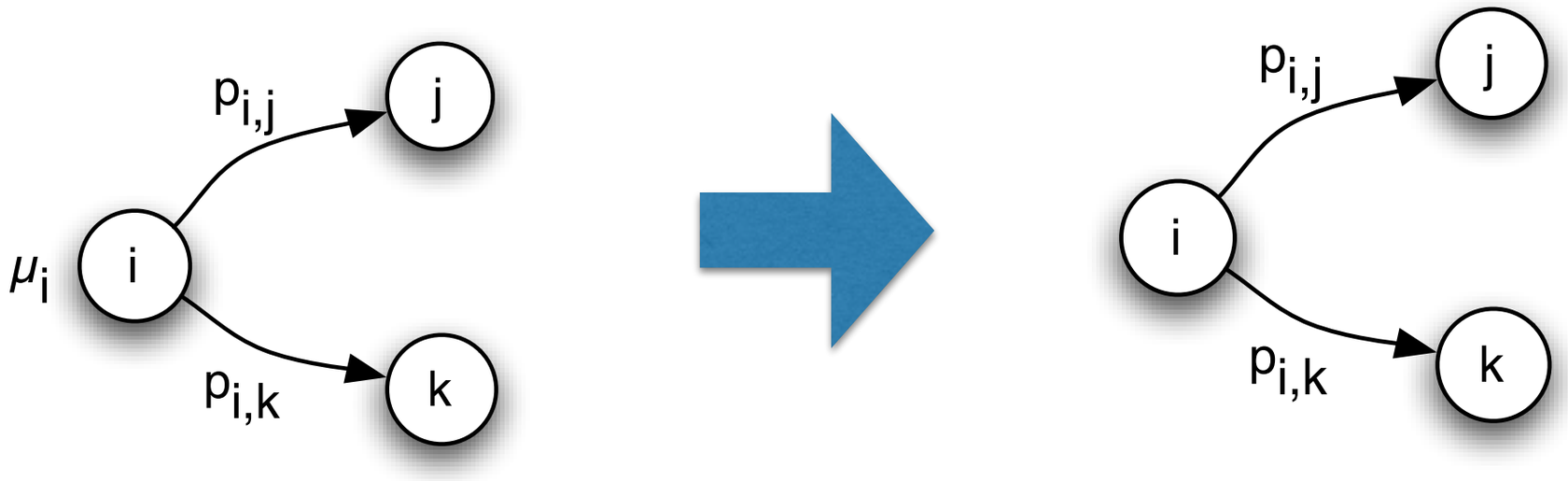
- Flow conservation equations:

$$\forall i, \sum \pi_i . q_{i,j} = 0 \quad \Rightarrow \sum_{i \neq j} \pi_i . q_{i,j} + \pi_j . q_{j,j} = 0$$

$$\Rightarrow \sum_{i \neq j} \pi_i . \mu_{i,j} = \sum_{i \neq j} \pi_j . \mu_{j,i}$$

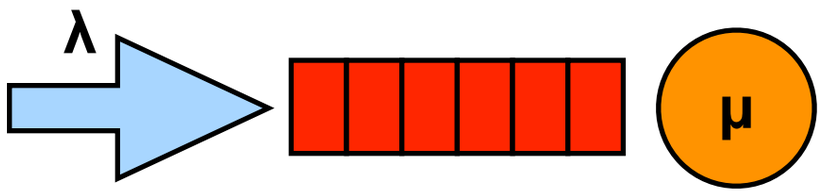
Embedded Markov Chain

- **There is an equivalent Markov Chain (discrete time) for a Markov Process**



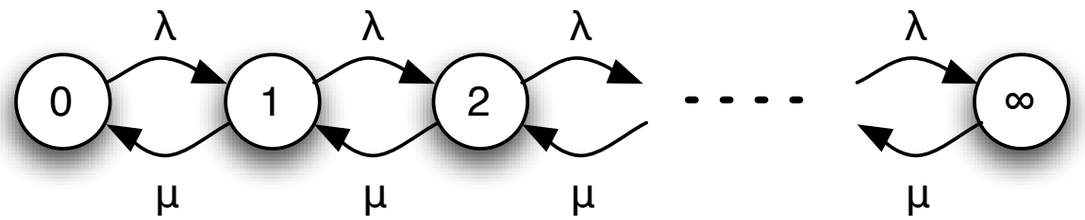
- **Properties:**
  - Markov Chain irreducible ⇔ Markov Process irreducible
  - Markov Chain transient ⇔ Markov Process transient
  - Markov Chain recurrent ⇔ Markov Process recurrent

M/M/1 analysis



Analysis: Markov Process

- **Representation of the number of clients in the system as a Markov process**
  - Valid, as we have a Poisson arrival process & exponential service time



- **It is a birth and death process**
  - Equilibrium equations:

$$\lambda.\pi(i-1) + \mu.\ \pi(i+1) = \lambda.\ \pi(i) + \mu.\ \pi(i)$$

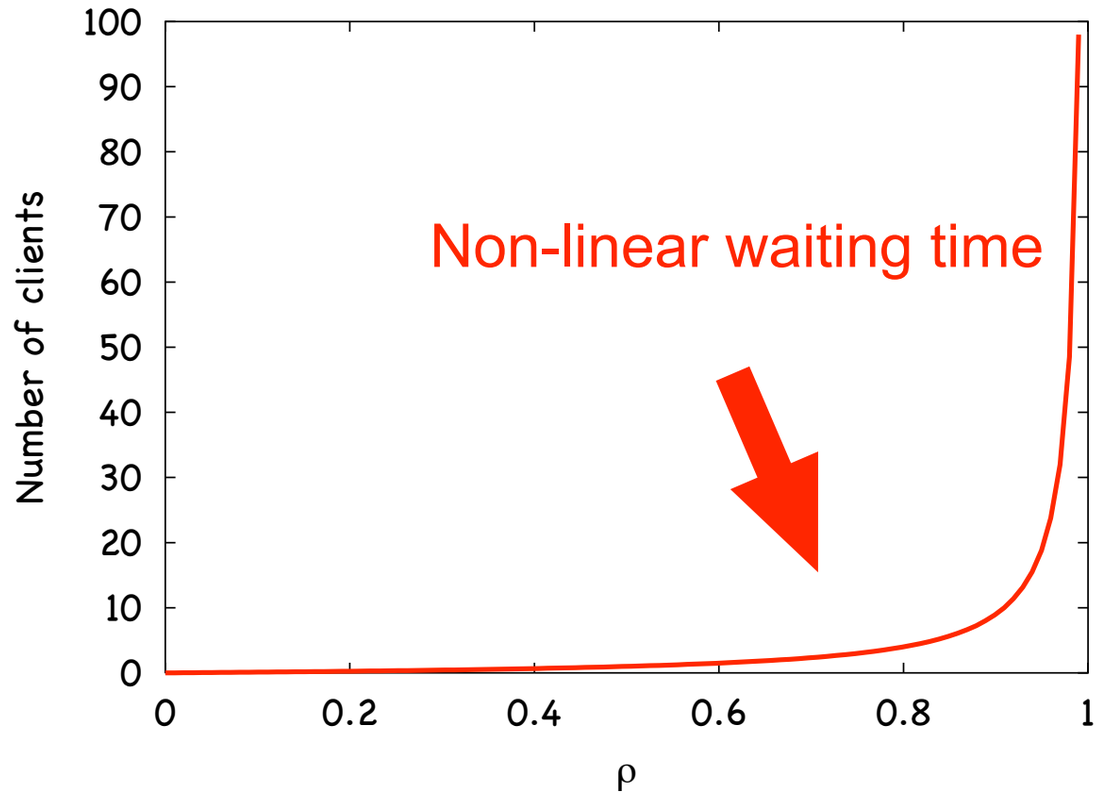
$$\lambda.\ \pi(0) = \mu.\ \pi(1)$$

  - Hence:
    - $\pi(i) = \rho^i.\ \pi(0) = \rho^i.(1 - \rho)$
    - $\pi(0)$ computed with $\sum \pi(i) = 1$, then using the sum of the terms of a geometric series

Main results

- **Queue load: ρ = λ/μ**
  - System is stable iff $\rho < 1 \Leftrightarrow \lambda < \mu$
- **Average number of clients in the system:**
  - Expected value of the number of clients:

$$Q = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1 - \rho}$$



- **Average Sojourn time:**
  - Via Little's formula ($Q = \tau.\lambda$)

$$\tau = \frac{1}{\mu - \lambda}$$

PASTA property

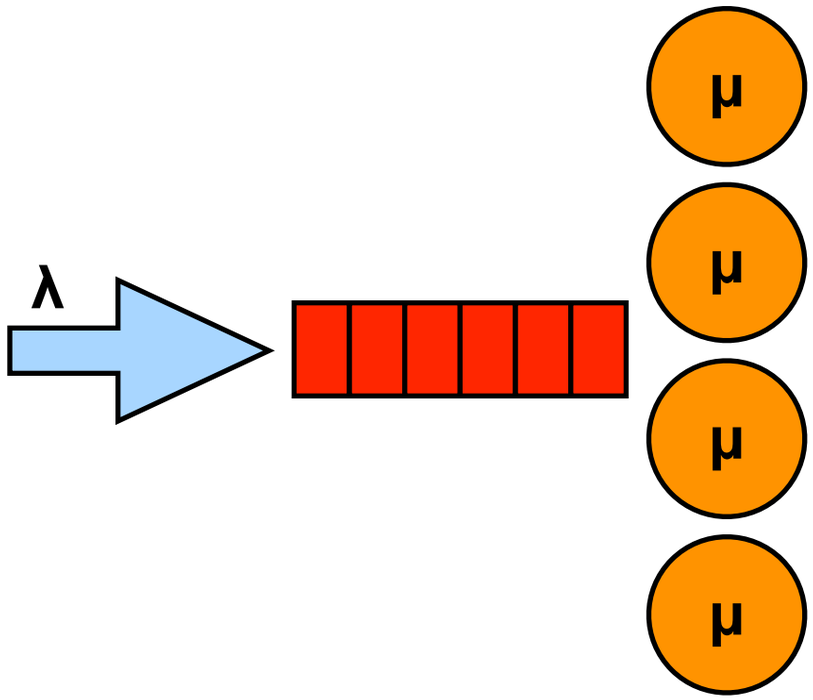
- **Expression of the sojourn time:**

$$\tau = \frac{1}{\mu - \lambda} = \frac{1}{\mu} \cdot \frac{1}{1-\rho} = \frac{1}{\mu} \cdot \left(\frac{\rho}{1-\rho} + 1 \right)$$

$$\boxed{\tau = \frac{\rho}{1-\rho} . \frac{1}{\mu} + \frac{1}{\mu}}$$

- **The PASTA property (*Poisson Arrivals See Time Averages*)**
  - In the case of exponential arrivals, the probability that a client finds N clients in the queue is equal to the stationary probability that the system contains N clients
  - Does not depend on the client, on the arrival moment, …

M/M/N queue



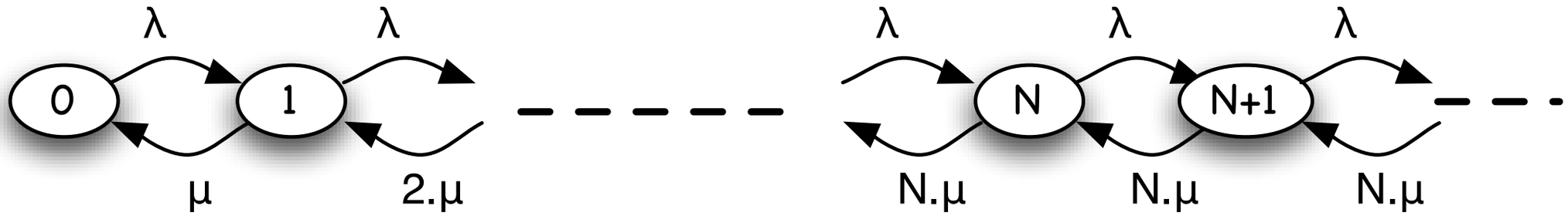
Arrivals: Poisson process (Parameter λ)

Service time : exponential (Parameter μ)

N servers, all identical

FIFO queue ; infinite length

M/M/N queue (Erlang-C)



Equations d'équilibre :

- Pour k = 0: $\lambda . \pi(0) = \mu . \pi(1)$
- Pour k < N: $\lambda.\pi(k-1) + (k+1).\mu . \pi(k+1) = \lambda . \pi(k) + k.\mu . \pi(k)$
- Pour k ≥ N: $\lambda.\pi(k-1) + N.\mu . \pi(k+1) = \lambda . \pi(k) + N.\mu . \pi(k)$

Solution:

- Pour k ≤ N: $$\pi(k) = \frac{\rho^k}{k!} \cdot \pi(0)$$
- Pour k ≥ N: $$\pi(k) = \left(\frac{\rho}{N}\right)^{k-N} .\pi(N)$$

M/M/N queue (Erlang-C)

- **Summing the probabilities, it is possible to express π(0) :**

$$\pi(0) = \frac{1}{\frac{\rho^N}{(N-1)! \cdot (N-\rho)} + \sum_{k=0}^{N-1} \frac{\rho^k}{k!}}$$

M/M/N queue (Erlang-C)

- **Probability to wait:**
  - D = P [Q ≥ N] = $\sum_{k=0}^{+\infty} \frac{\rho^k}{N^k} \cdot \pi(N)$

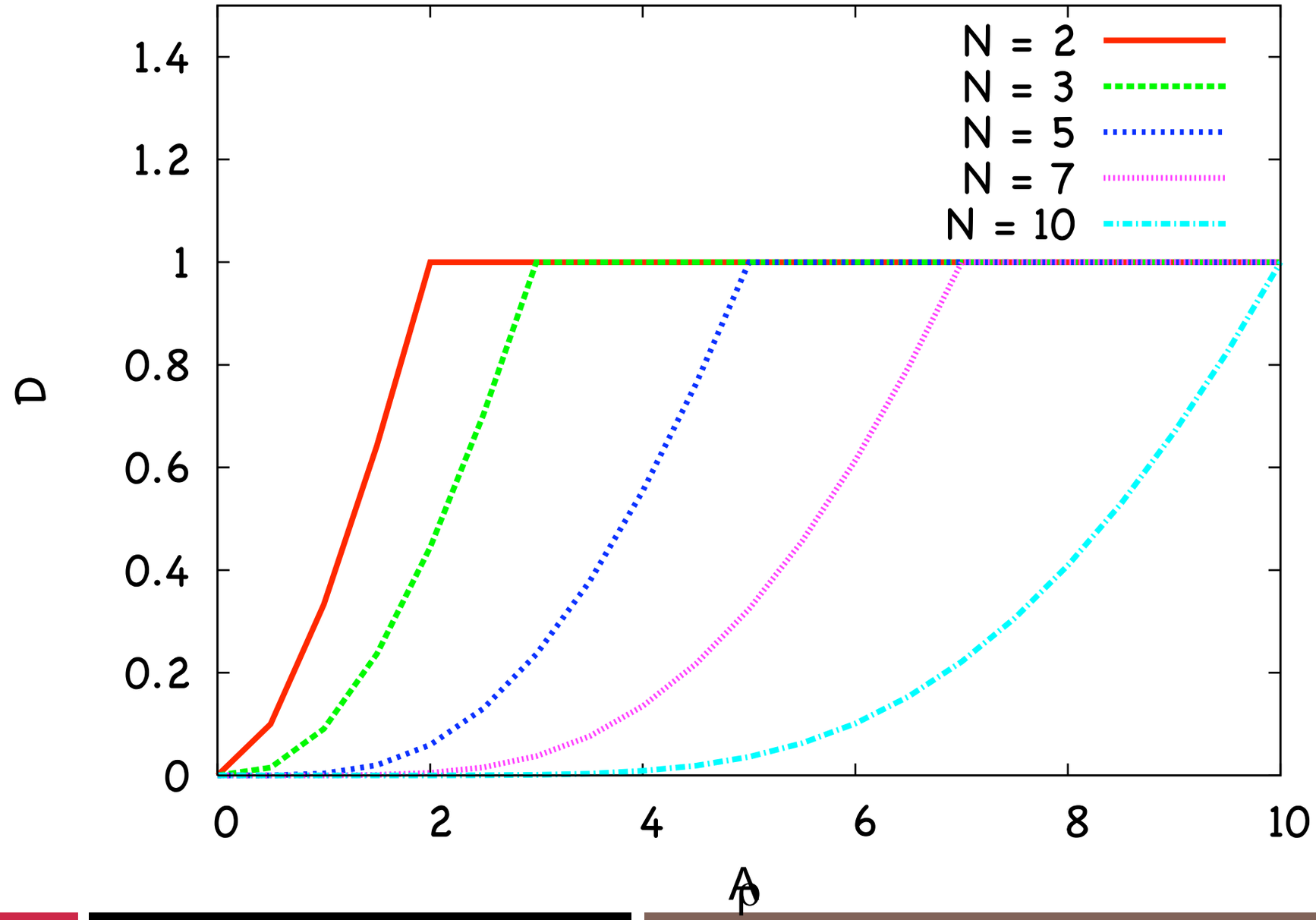
    = $\frac{N}{N-\rho} \cdot \pi(N)$

  - $$D = P[n \geq N] = \frac{\frac{\rho^N}{N!}}{\frac{\rho^N}{N!} + \left(1 - \frac{\rho}{N}\right) \sum_{k=0}^{N-1} \frac{\rho^k}{k!}}$$

- **It is the "Second Erlang law": $E_{2,N}(\rho)$**

M/M/N queue (Erlang-C)

- **Probability to wait in function of the load for different N:**



M/M/N queue (Erlang-C)

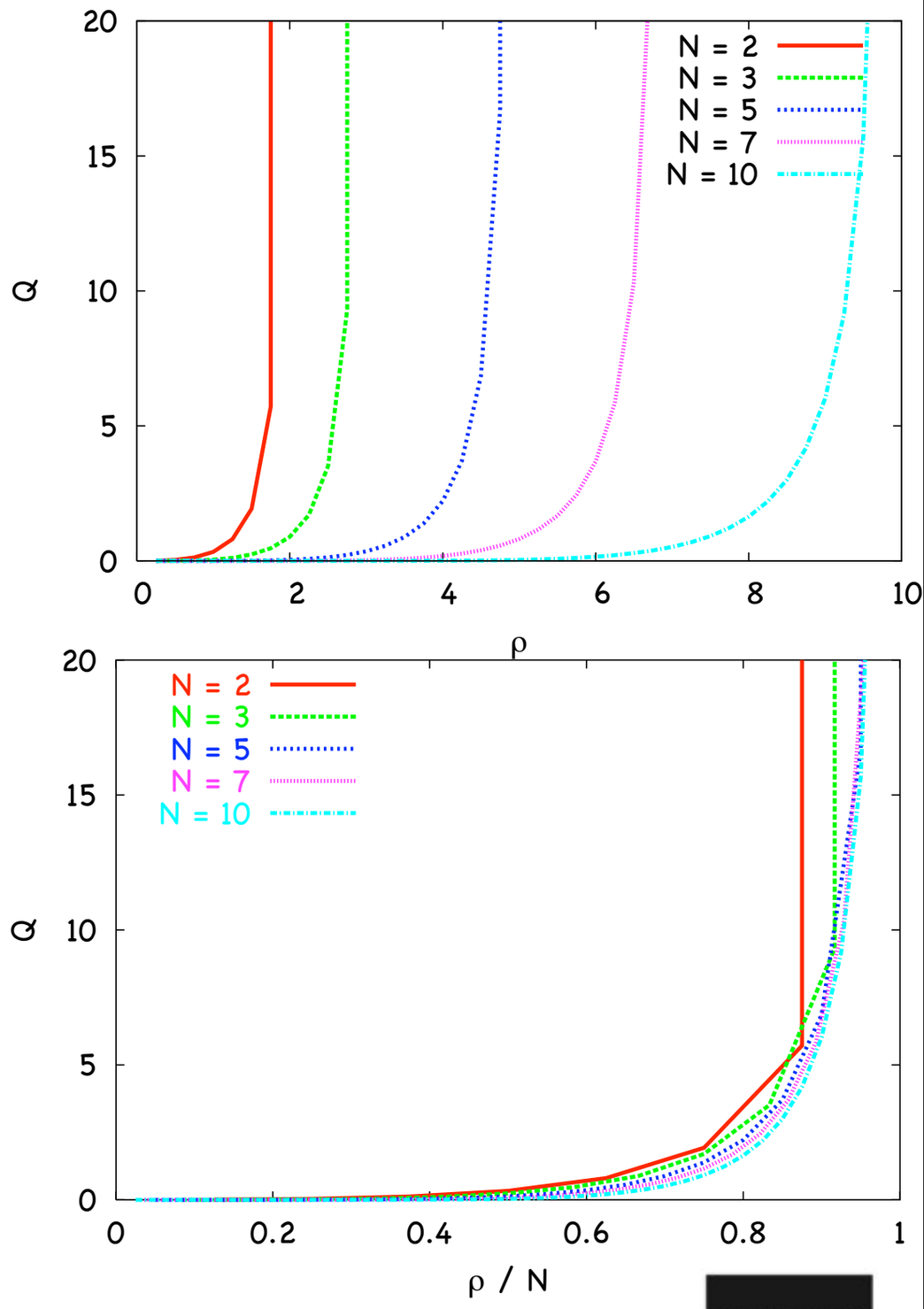
- **Average number of clients waiting:**

$$Q = \sum_{k=0}^{+\infty} k \cdot \pi(N+k)$$

$$= \sum_{k=0}^{+\infty} k \cdot \frac{\rho^k}{N^k} \cdot \pi(N)$$

$$= \frac{\frac{\rho}{N}}{\left(1 - \frac{\rho}{N}\right)^2} \cdot \pi(N)$$

$$\boxed{Q = \frac{\rho}{N - \rho} \cdot D}$$



M/M/N queue (Erlang-C)

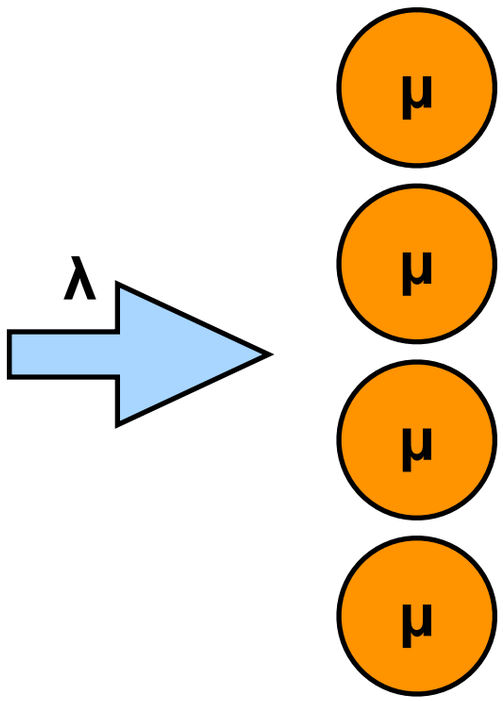
- **Average waiting time (in the queue) :**
  - Apply Little's formula to the queue (without servers):
  - Q = W . λ

$$\Rightarrow \boxed{W = \frac{Q}{\lambda} = \frac{\rho}{\lambda \cdot (N - \rho)} \cdot D}$$

- **Adding a service time, we get the sojourn time:**

$$\tau = W + \frac{1}{\mu}$$

M/M/N/N queue



Arrivals: Poisson process (Parameter λ)

Service time : exponential (Parameter μ)

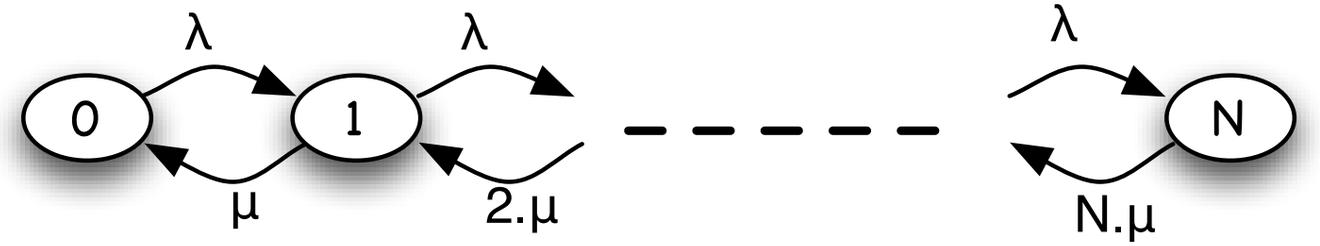
N servers, all identical

No queue (system full => client rejected)

Always stable system

M/M/N/N queue (Erlang-B)

- **Only N clients can enter the system**
  - Finite Markov process with N states



$$\sum_{k=0}^{N} \frac{\rho^k}{k!} \cdot \pi(0) = 1$$

$$\pi(0) = \frac{1}{\sum_{k=0}^{N} \frac{\rho^k}{k!}}$$

No simple expression for π(0)

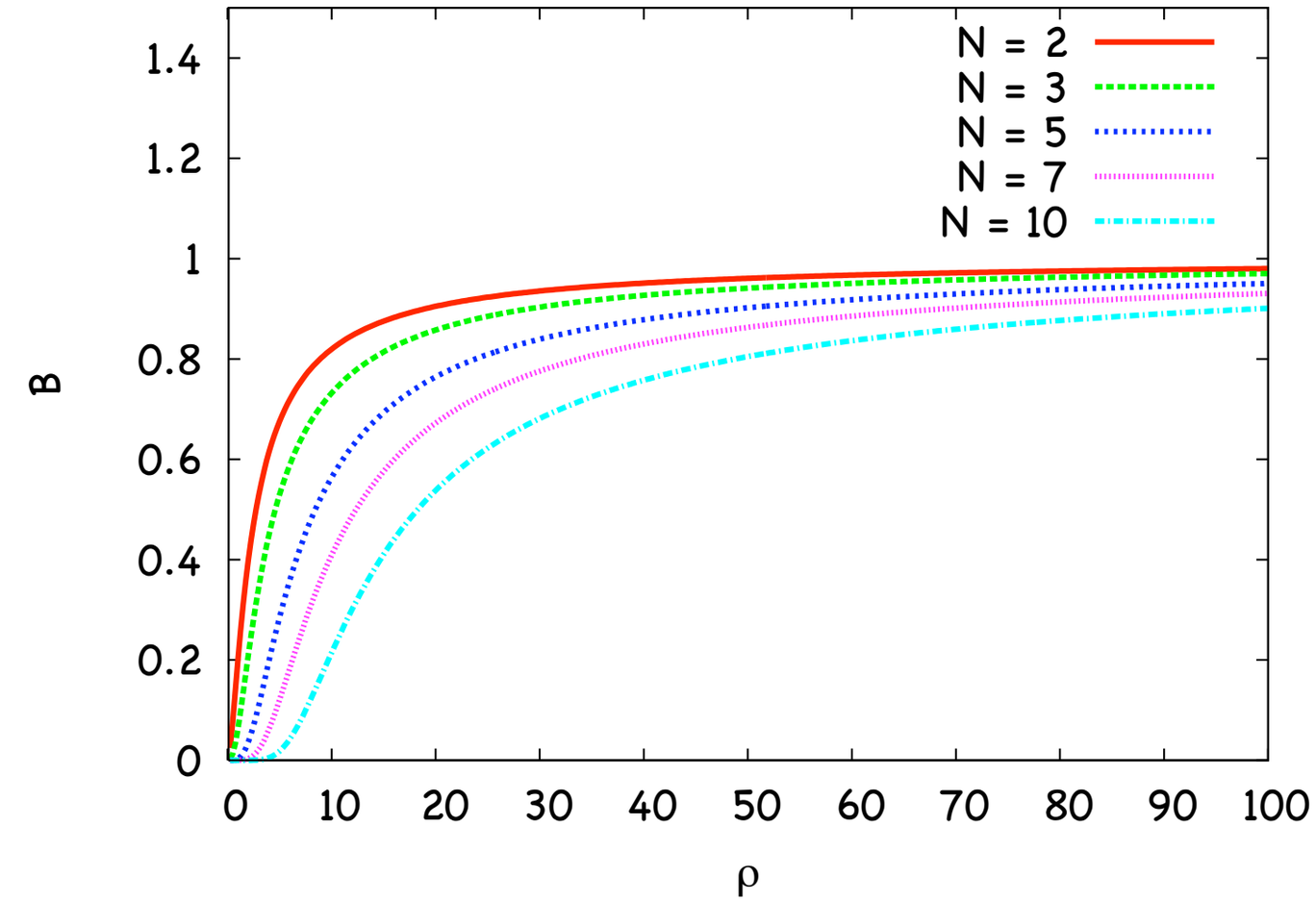
$$\pi(i) = \frac{\frac{\rho^i}{i!}}{\sum_{k=0}^{N} \frac{\rho^k}{k!}}$$

M/M/N/N queue (Erlang-B)

- **Loss probability: B = π[N]**

- **=**

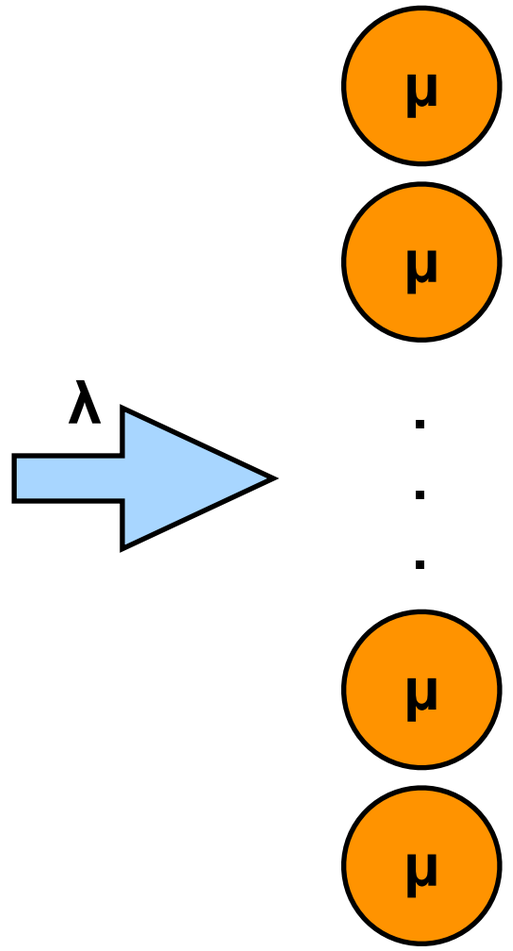
$$\frac{\frac{\rho^N}{N!}}{\sum_{k=0}^{N} \frac{\rho^k}{k!}}$$



- **"First Erlang Law" : $E_{1,N}(\rho)$**
  - Use abacus, tables, or a computer program to find the number of resources to deploy for satisfying a bounded rejection probability

Exercise

Analyze the M/M/∞ queue



Conclusion

Definitions / Vocabulary

(to complete)

- **Parameters**
  - Arrival process
  - Process intensity
  - Service time
  - Traffic
    - Unit = Erlang
  - Load
- **Usual metrics**
  - Average number of clients in the queue
  - Waiting time
  - Sojourn time
  - Exit process
- **Kendall Taxonomy**

Mathematical basis

(to complete)

- **Exponential distributions**
  - Definition, properties
  - Fitting test

- **Poisson process**
  - Relationship with exponential distributions

- **Markov Chains and Processes**
  - Principle
  - State graph, transition matric
  - Resolution methods

Queues properties

(to complete)

- **General properties**
  - System stability
  - Little's formula
- **M/M/1**
  - Average number of clients
  - Waiting and sojourn time
- **M/M/N**
  - Waiting probability
  - Average number of clients in the queue
  - Sojourn time
- **M/M/N/N**
  - Loss probability
- **M/M/∞**
  - Poisson process