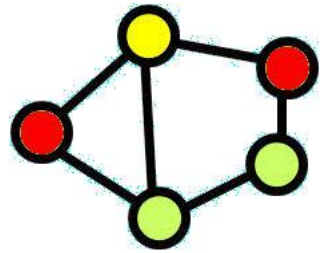
Slide supporting material



Lesson 6: Queues and Markov Chains

Giovanni Giambene

Queuing Theory and Telecommunications: Networks and Applications
2nd edition, Springer

Introduction to Queuing Systems

Queuing Systems

- Queuing systems are everywhere.
  - For example, airplanes "queue up", waiting for a runway so they can land. Then, they line up again to take off.
  - People line up for tickets, to buy groceries, etc.
- The Danish engineer A. K. Erlang founded queuing theory by studying telephone switchboards in Copenhagen for the Danish Telephone Company in the early 1900s.

Queuing Systems (cont'd)

- The interest is here to study queuing systems and related analytical methods for the study of telecommunication networks.
- **In telecommunication networks, queuing theory is used every time a network resource is shared by competing 'service requests'**.

Queues in Telecommunications

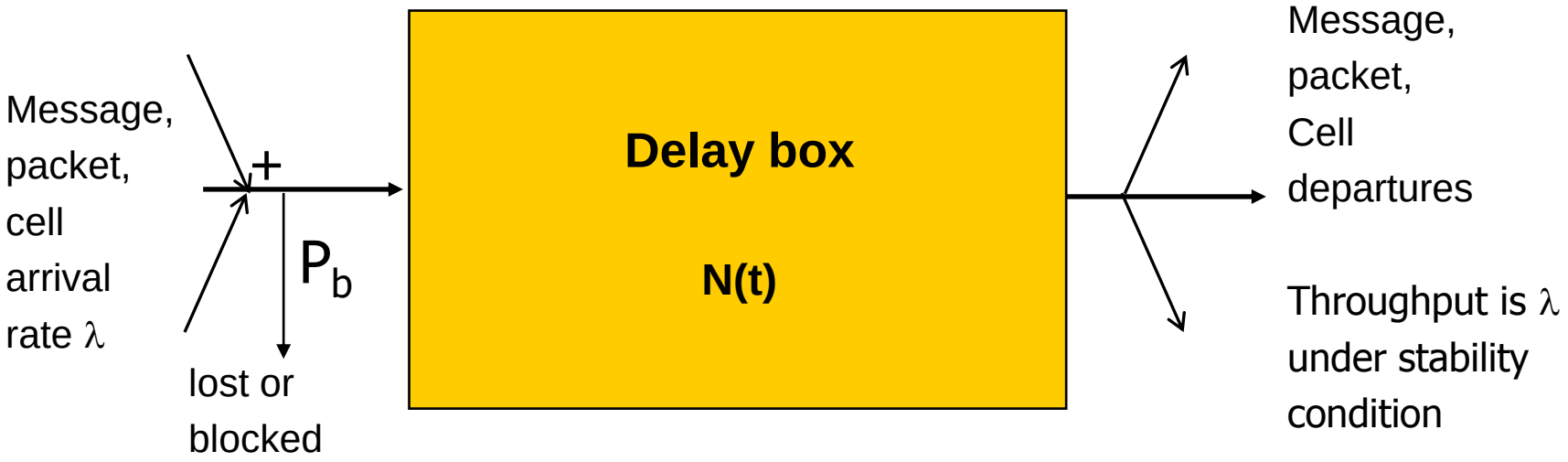
- Every protocol in every node of a telecommunication network can be modeled through an appropriate queuing process.
- Queues can be applied at different OSI levels:
  - **OSI Layer 1**: Blocking phenomena of a traffic flow (i.e., a call) due to unavailable resources in at least one link in the path from source to destination.
  - **OSI Layer 2**: Queuing is generated by different packets sharing the transmission resources of a link connecting two adjacent nodes (MAC, multiplexing);
  - **OSI Layer 3**: There are layer 3 buffers for IP-level QoS support (e.g., DiffServ).

Queuing Systems: Terminology

- In the study of queuing systems, there are also **different terms that have the same meaning and can be used interchangeably**.
- Some interesting examples are:
  - Client/customer/service request/job/packet/message/call/etc.
  - Service/transmission/etc.
  - Server/transmitter/etc.
  - Queue/buffer/memory/etc.

Queuing System: Basic Notations

- A basic model for a delay/loss system (**node or link**) in telecommunications:

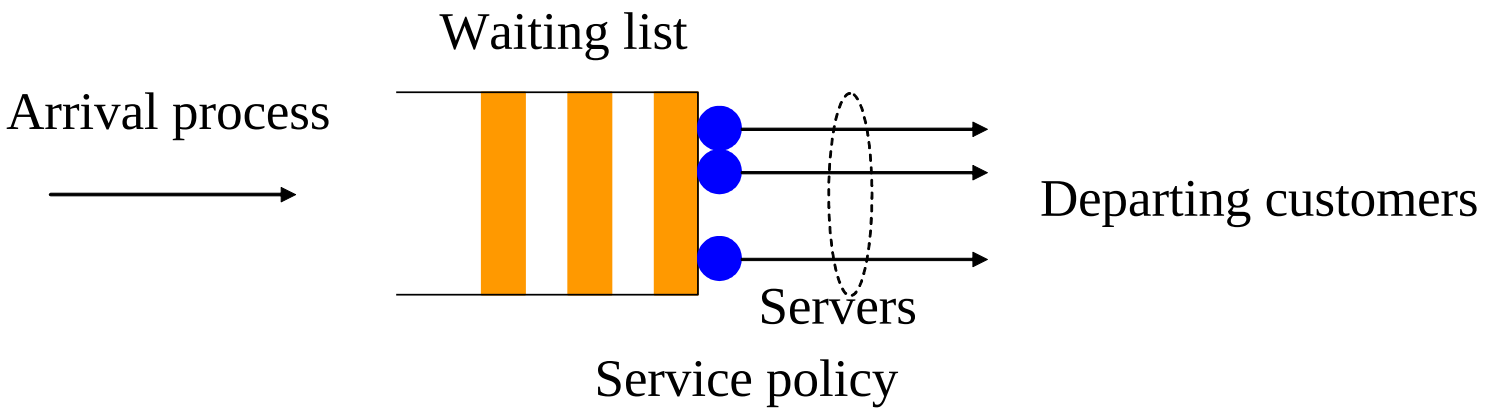


- Mean time spent in system by a customer (service request) = T
- Number of customers in the system at time t = N(t)
- Fraction of arriving customers that are lost or blocked (congestion) = P_b
- Long-term mean arrival rate of customers = λ
- Average number of customers/second that pass through the system = throughput

Queuing System: Basic Characteristics

- Queues are special cases of stochastic processes, which are represented by a state N(t) with discrete values, for instance denoting the number of queued 'entities' (called below 'requests'). **Queues are modeled by 'chains'.**

- A queue is characterized by:
  - An **arrival process** of service requests (mean arrival rate denoted with λ),
  - A **waiting list** of requests to be processed,
  - A **discipline** according to which requests are selected in the queue to be served,
  - A **service process**.



Kendall's Notation for Queuing Systems 'A/B/S/Δ/E - *service*'

Integer numbers

Letters or acronyms

A/B/S/Δ/E - ***service*** :

A = interarrival time distribution or arrival process

G = **General** (i.e., not specified); M = **Memoryless** (Poisson);

B = service time distribution

G = **General** (i.e., not specified); M = **Memoryless** (exponential);
D = **Deterministic**

S = number of parallel servers, S = 1, 2, … (S ≥ 1)

Δ = number of available places in the queue (wait+service), Δ ≥ S

E = limit on population, E ≥ S

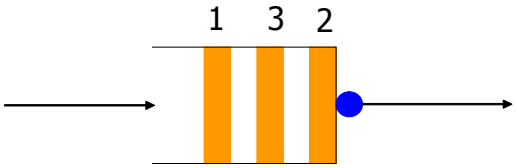
Service = denotes the discipline adopted to service the requests in the queue; for instance, First Input First Output (FIFO), Last Input First Output (LIFO), and Service In Random Order (SIRO).

Note: Δ and E are omitted if they are infinity.

D. G. Kendall, the English mathematician who first used the term 'queuing system' in the paper: "Some Problems in the Theory of Queues", *Journal Royal Statistical Society*, 1951

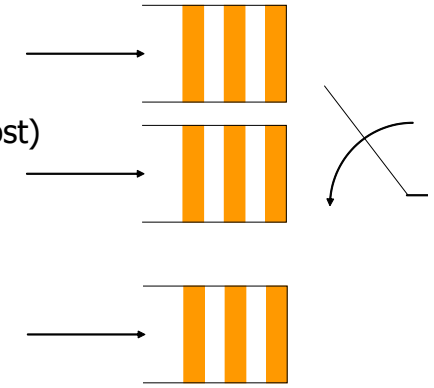
Service Policy vs. Scheduling

- **Service policy refers to the order according to which requests are serviced in a queue.** This order can also be dynamic if newly-arriving requests can change the service order of previous ones in the queue.



Head-Of-Line (HOL) packet

- **Scheduling** refers to the case where many queues share a given server (multiplexing context); the **task of the scheduler is to select the next request to be serviced among those in the queues**.
  - Example: Round Robin (RR) etc.
  - The service order can be static or dynamic.
  - Each queue may represent a different traffic class (in an end-host) or different end-hosts within a class.
  - Overheads (e.g., headers, dead times) can be needed in switching the server from one queue to the next one.
  - All queues sharing a given server behave globally as a single queue with a suitable service policy.



Main Performance Parameters for a Queue

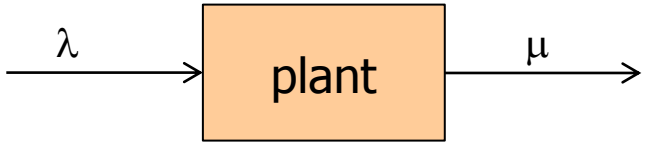
- The distribution of the number of requests in the queue (queue state distribution probability)
  - **Mean number of requests in the queue, N**
- The distribution of the time spent from the arrival of a request to the queue to the instant when the service of this request completes.
  - **Mean time spent to cross the queue (i.e., mean queue delay), T.**

Queue Stability (Steady-State)

- A single-server queue system is **stable** if

  `arrival rate of requests < service rate of requests`

  - For instance in an industrial production plant modeled as a global queue, stability requires that the frequency λ of the arrival of product requests **be lower than** the rate of product completion, μ:



  - $\rho = \lambda/\mu$ denotes the traffic intensity offered to the queue. **In a single server queue, stability requires that $\lambda < \mu$ or $\rho < 1$ Erl**. ρ also denotes the 'server utilization factor', that is the percentage of time (between 0 and 1) that the sever is busy.

- In an **unstable queue**:
  - Packets accumulate in the queue without a bound (packet delays increase continuously).
  - Flow/admission control may be used to limit the packet arrival rate.
  - Prioritization of flows keeps delays bounded for the important traffic.

Little Formula (1961)

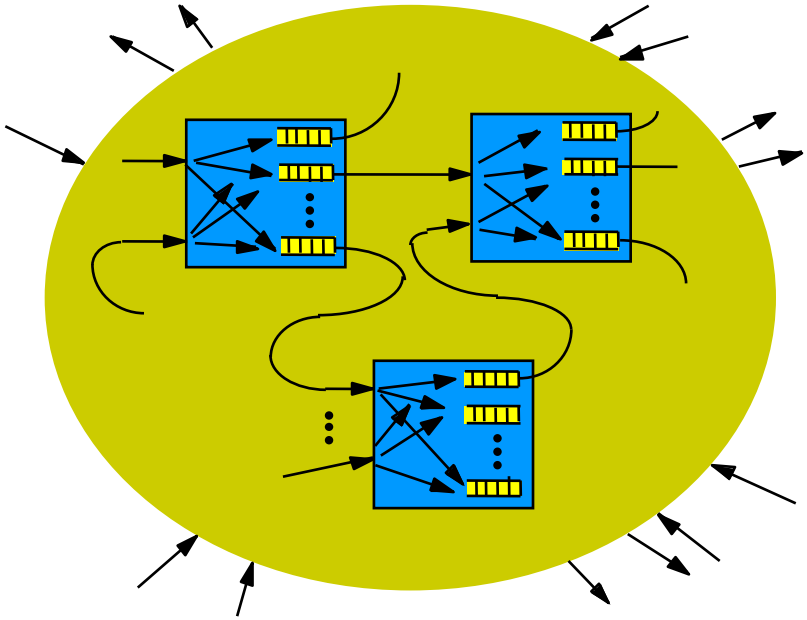
- Assumptions (the queuing system as a "black box"):
  - **General G/G/X/Y queuing system**
  - **Boundary condition**: The queue must become empty at some time instants (this is assured if the queue is stable, as we assume below).
  - **Conservation of customers**: All arriving customers (requests) will eventually complete their service and will leave the system (i.e., there are no customers lost).
  - The queuing system admits a **steady-state**: (i) the queue becomes empty from time to time; (ii) the queuing system is described by an ergodic process (time averages are equal to the corresponding statistical averages).
- The Little formula relates T and N quantities for a queue ($\overline{\lambda}$ denotes the 'mean rate of requests accepted into the system'):

$$T = \frac{N}{\overline{\lambda}} \quad \Leftrightarrow \quad N = \overline{\lambda} T$$

J. C. C. Little, "A Proof for the Queueing Formula: Λ = XW," *Operations Res.*, Vol. 9, pp. 383-387, 1961

Little Formula (cont'd)

- The Little theorem can also be applied to a packet-switched telecommunication network as a whole.
  - Note that a telecommunication network is formed of nodes and links. **Each node can be modeled as a set of queues representing the transmission buffers** (collecting different input traffic contributions) on different output links.



Little Formula (cont'd)

- The Little formula can be used to relate the mean delay experienced by a message (or packet) from the entrance to the exit from the network, T, and the mean number of messages (or packets) that are in the network, N, by means of the mean arrival rate λ of messages entering the network: $T = N/\lambda$

- Intuitively: since the Little formula is valid under very general assumptions on the queuing discipline and since the state model of a queue is unaffected by typical scheduling schemes, many queuing disciplines (e.g., FIFO, SIRO, LIFO, PS, etc.) achieve the same mean queue delay (while other moments of the delay do depend on the queuing discipline); this intuitive results is supported by the **insensitivity property**:

  - If the service discipline fulfills the insensitivity property assumptions, the queue state distribution and the mean delay do not depend on the service type. Hence, $E(\mathbf{T}_{FIFO}) = E(\mathbf{T}_{SIRO}) = E(\mathbf{T}_{LIFO})$. However, $Var(\mathbf{T}_{FIFO}) < Var(\mathbf{T}_{SIRO}) < Var(\mathbf{T}_{LIFO})$.

Little Formula (cont'd)

The insensitivity property holds when (hypotheses):

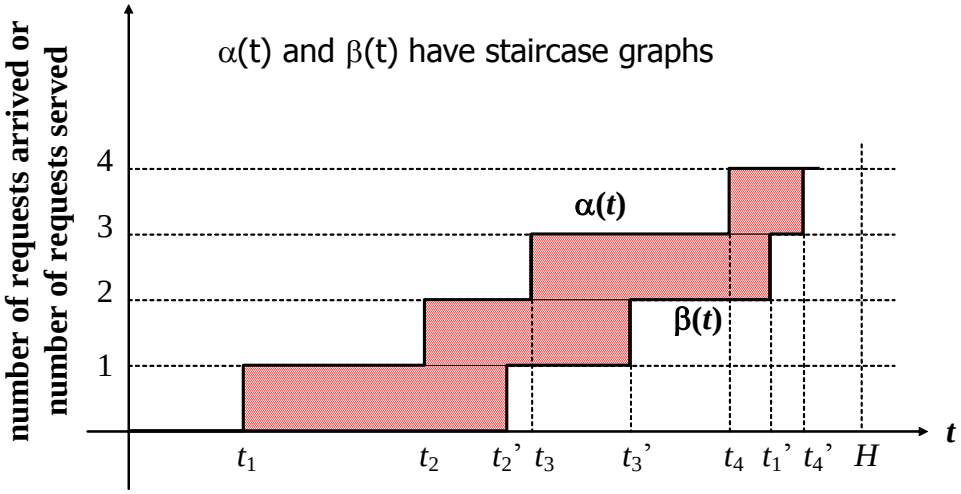
- The **service policy is independent of the service time**.
- The service policy is **non-preemptive** (a job that has started service will remain in service until completes).
- The service policy is **work-conserving** (there are not server vacations).
- **The queue is stable**.

- T
  e to the exit
  f s (or
  p ean arrival
  r
- I eneral
  a te model of
  a any queuing
  c same
  mean que ne other moments of the delay do depend on the queuing di ne); this intuitive results is supported by the **insensitivity property**:

  - If the service discipline fulfills the insensitivity property assumptions, the queue state distribution and the mean delay do not depend on the service type. Hence, $E(\mathbf{T}_{FIFO}) = E(\mathbf{T}_{SIRO}) = E(\mathbf{T}_{LIFO})$. However, $Var(\mathbf{T}_{FIFO}) < Var(\mathbf{T}_{SIRO}) < Var(\mathbf{T}_{LIFO})$.

Little Formula Proof

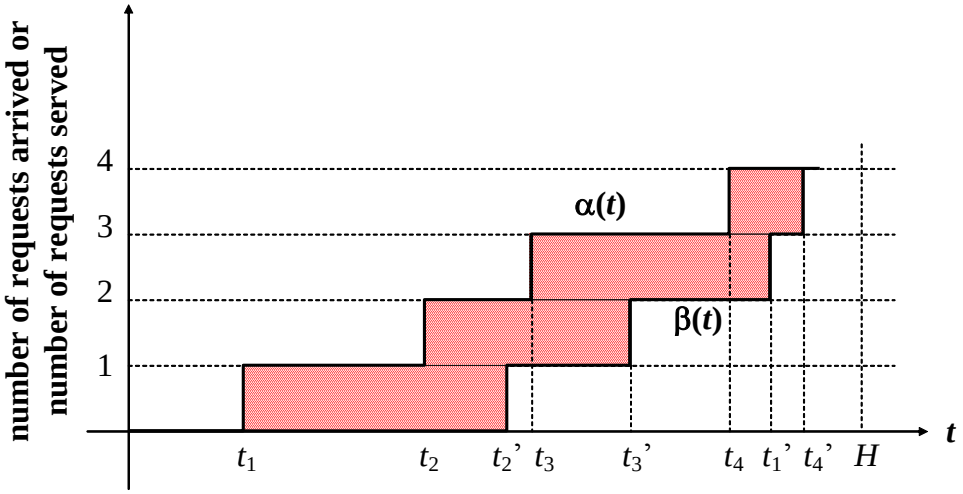
- We consider that at time t = 0 the queue is idle.
- Let us denote:
  - $\alpha(t)$ = number of requests arrived in the interval (0 , t);
  - $\beta(t)$ = number of requests that complete service in the interval (0 , t);
  - t_i = arrival instant of the i-th request;
  - t_i' = departure instant (i.e., service completion) of the i-th request.



- We neglect the cases of multiple arrivals (or departures) → both $\alpha(t)$ and $\beta(t)$ have variations of value 1 corresponding to arrival or departure instants, respectively.
- Note that $t_1 < t_2 < t_3 < \ldots$. Whereas, the ranking of the instants t_1', t_2', t_3', … depends on the adopted queuing policy (in the FIFO case $t_1' < t_2' < t_3' < \ldots$, but this is not necessary for the Little theorem).

Little Formula Proof (cont'd)

- The following relationships will be used:
  - $T_i = t_i' - t_i$ represents the time spent in the system by the i-th request;
  - $N(t) = \alpha(t) - \beta(t)$ is the number of requests in the queue at the instant $t \geq 0$.
- Let us consider a generic instant H, where $\alpha(t) = \beta(t)$, so that the system is empty (i.e., $N(H) = 0$). The time average of the delay experienced by a request arrived at the queue in the interval (0 , H) is:



$$\overline{T_H} = \frac{\sum_{i=1}^{\alpha(H)} T_i}{\alpha(H)} = \frac{\sum_{i=1}^{\alpha(H)} \left(t_i' - t_i\right)}{\alpha(H)} = \frac{\sum_{i=1}^{\alpha(H)} t_i' - \sum_{i=1}^{\alpha(H)} t_i}{\alpha(H)}$$

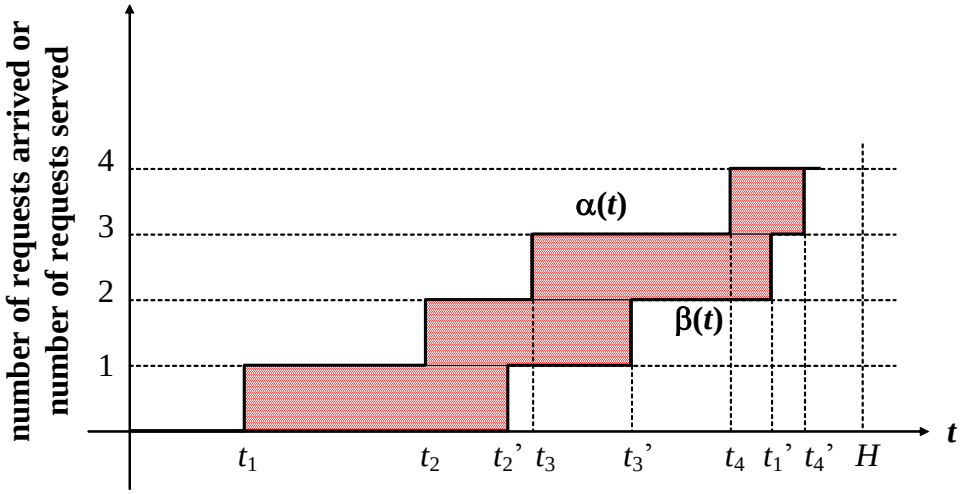
Little Formula Proof (cont'd)

- The difference

$$\sum_{i=1}^{\alpha(H)} t_i' - \sum_{i=1}^{\alpha(H)} t_i$$

is the highlighted area that can also be expressed as

$$\int_0^H [\alpha(t) - \beta(t)]dt = \int_0^H N(t)dt$$



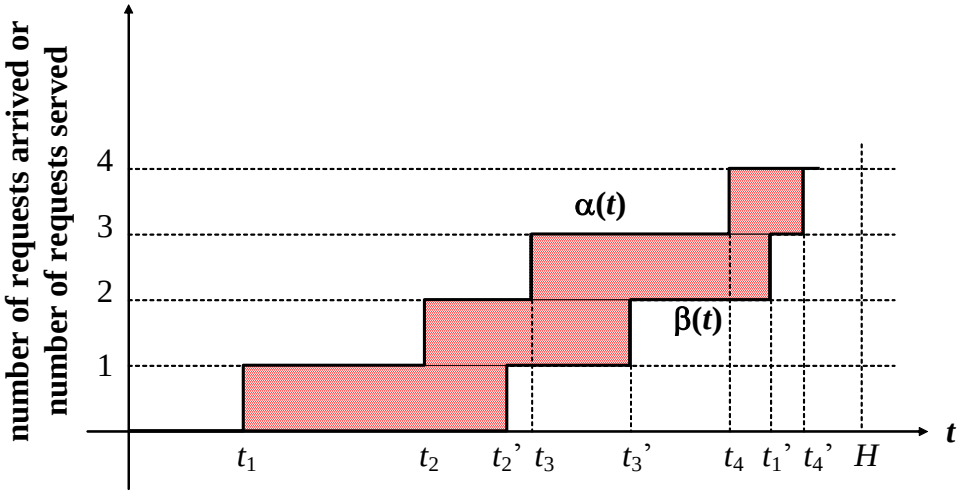
- $\overline{N_H} = \frac{1}{H}\int_0^H N(t)dt$ represents the time average of the number of requests in the queue in the interval (0 , H)

- $\overline{\lambda_H} = \alpha(H)/H$ represents the average arrival rate in the interval (0 , H).

Little Formula Proof (cont'd)

- We have:

$$\overline{T_H} = \frac{\int_0^H N(t)dt}{\alpha(H)} = \frac{H}{\alpha(H)} \times \frac{1}{H}\int_0^H N(t)dt = \frac{\overline{N_H}}{\overline{\lambda_H}}$$



- By employing the **ergodicity assumption**, we have that time averages can be substituted by statistical ones, here denoted as T, N and λ, respectively, thus obtaining the Little formula (QED).

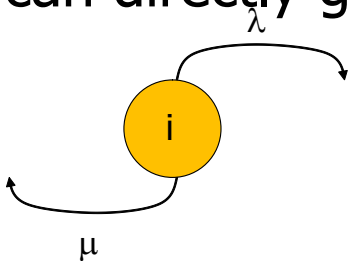
Markov Chains for Queues Analysis

A Markov Chain

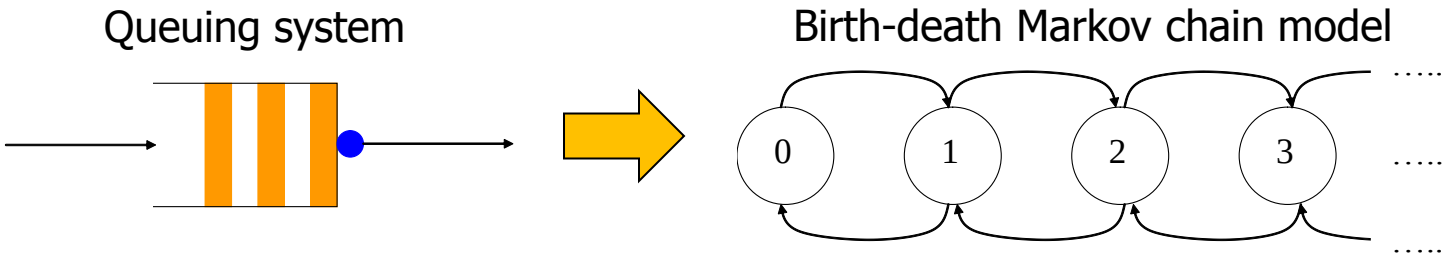
- The chain state at instant t_{n+1}, $X(t_{n+1})$, depends only on the state at the previous instant t_n, $X(t_n)$.
  - The stochastic process evolution is characterized only by its state value at the present instant, but not on the time already spent in this state. This memoryless characteristic is guaranteed only by **state sojourn times exponentially-distributed** in the case of a continuous-time chain (geometric distribution for a discrete-time chain).

Birth-Death Markov Chains

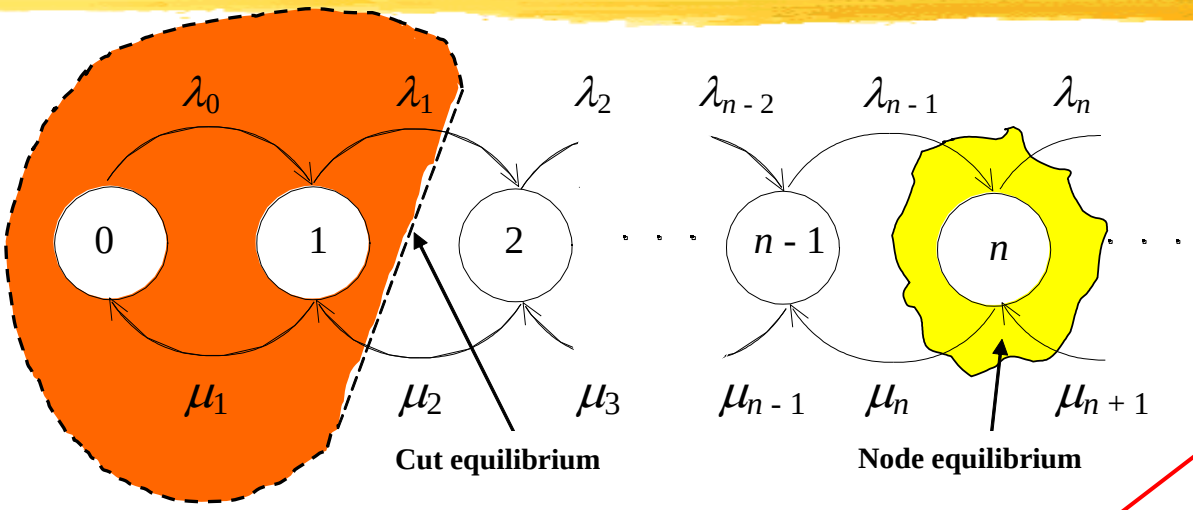
- A Markov chain represents a **birth-death process** when the state transitions in the chain occur only among adjacent states: from state i we can directly go only to state i+1 or i–1.



  - A Markov chain of the birth-death type can be used to **model** a queue with Poisson arrivals and exponential service times. These types of queues are denoted by symbols like **M/M/.../...** , according to the Kendall's notation.



Birth-Death Markov Chains: State Probability Distribution



We should write and solve **differential equations** modeling the system dynamics and then to take the limit for t to infinity to study the regime condition. This leads to the flow equilibrium conditions written below.

- **Ergodic condition**: $\exists\ k\ \ so\ \ that: \forall\ n \geq k,\ \lambda_n < \mu_n$ ⇒ Sufficient condition for stability
- **If the state space is finite (i.e., finite rooms in the queue), the queue is always stable for any traffic intensity value.**
- If the ergodic condition is fulfilled, the chain admits a **steady state**; at regime, state probabilities $P_n(t)$ (= probability that the system is in state n at time t) do not depend on time ⇒ $dP_n(t)/dt = 0$ and $P_n(t) = P_n$.
- Then, the following **cut equilibrium conditions can be written:**

$$\lambda_i P_i = \mu_{i+1} P_{i+1} \quad \text{for } i = 0, 1, \ldots$$

Birth-Death Markov Chains: State Probability Distribution (cont'd)

- Under stability condition (we impose ergodicity), we solve **recursively** the **cut equilibrium conditions** for i = 0, 1, …:

$$\begin{cases} \text{cut 1 balance}: \ \lambda_0 P_0 = \mu_1 P_1 \Rightarrow P_1 = \dfrac{\lambda_0}{\mu_1} P_0 \\ \text{cut 2 balance}: \ \lambda_1 P_1 = \mu_2 P_2 \Rightarrow P_2 = \dfrac{\lambda_1}{\mu_2} P_1 = \dfrac{\lambda_1}{\mu_2}\dfrac{\lambda_0}{\mu_1} P_0 \\ \ldots\ldots \\ \text{cut } i \text{ balance}: \ \lambda_{i-1} P_{i-1} = \mu_i P_i \Rightarrow P_i = \dfrac{\lambda_{i-1}}{\mu_i} P_{i-1} = P_0 \prod_{n=1}^{i} \dfrac{\lambda_{n-1}}{\mu_n} \end{cases}$$

$$\forall i \geq 1$$

- All state probabilities are expressed as functions of both the transitional rates and the probability of state '0', P_0. Therefore, we impose the **normalization condition** in order to obtain P_0

$$\sum_{i=0}^{\infty} P_i = 1 \Rightarrow P_0 \sum_{i=0}^{\infty} P_i / P_0 = 1 \Rightarrow P_0 \left(1 + \sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n} \right) = 1 \Rightarrow P_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n}}$$

Queue Performance Parameters

In case of a queuing system with S servers, ρ still denotes the mean number of busy servers → **$\rho < S$ [Erl] is the stability condition.**

- If the Markov chain models a queuing system, we can determine the following quantities once the state probability distribution has been solved obtaining P_i values:
  - Mean number of requests in the queue → $N = \sum_{n=0}^{\infty} nP_n = P'(1)$
  - Mean delay to cross the queue (Little theorem) → $T = \frac{N}{\bar{\lambda}}$

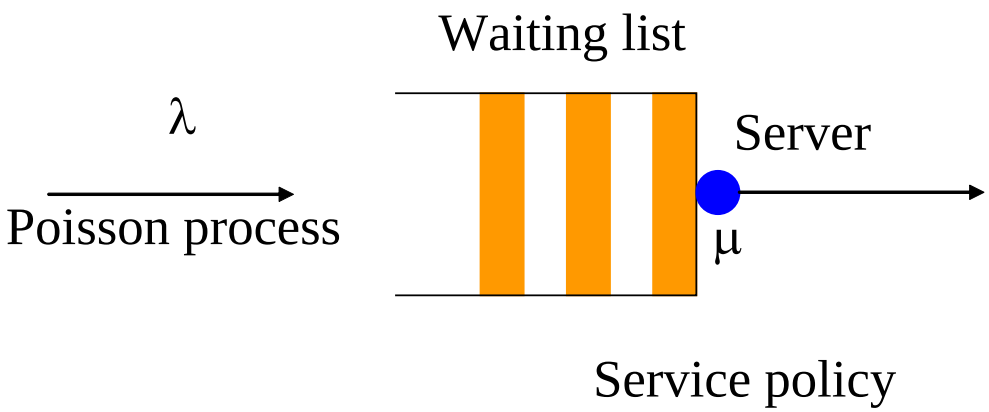
    where $\bar{\lambda}$ denotes the mean rate of requests entering the system, obtained as $\Sigma\lambda_i P_i$
  - Since T is the sum of mean service time (x) and mean waiting time (w), T = x + w, we multiply both members by $\bar{\lambda}$:

$$\bar{\lambda}T = \bar{\lambda}x + \bar{\lambda}w$$

Mean number of requests in service = mean number of busy servers = traffic intensity ρ in absence of blocking

The M/M/1 Queue

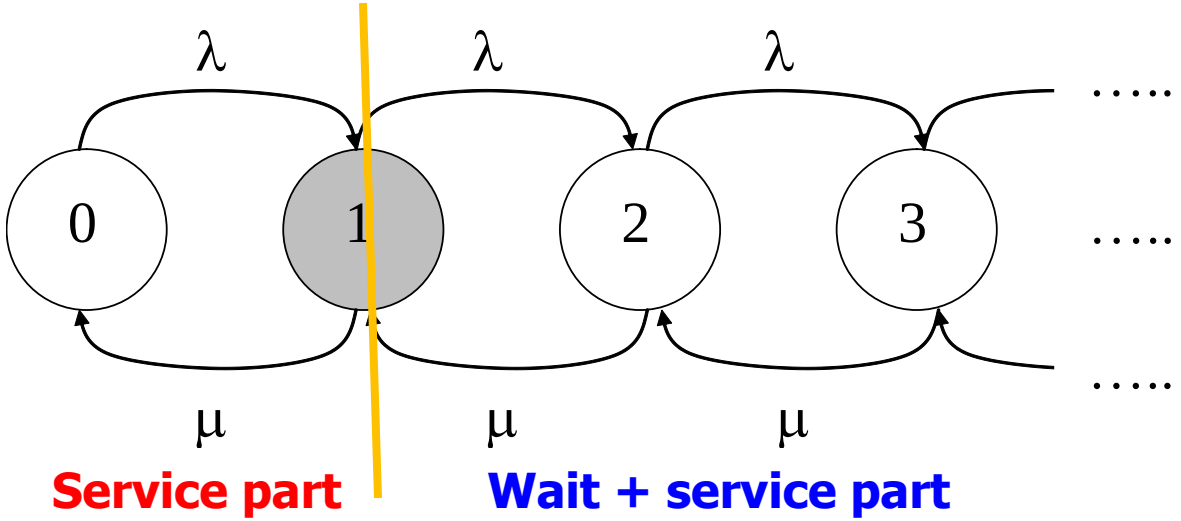
- M/M/1 queue: Poisson arrivals of requests (mean rate λ), exponentially-distributed service time (mean rate μ), single server, infinite rooms, and infinite population of users.



- **The state of the system is the number of requests in the queue (including the served one).**
- We can model the M/M/1 queue as a special case of a birth-death Markov chain with $\lambda_i \equiv \lambda$ and $\mu_i \equiv \mu$. Stability is assured by the ergodicity condition: $\rho = \lambda/\mu < 1$ Erlang.

The M/M/1 Queue (cont'd)

- Markov chain model for the M/M/1 queue:



- From cut equilibrium and normalization conditions, we have:

Geometric distribution

$$P_i = P_0\left(\frac{\lambda}{\mu}\right)^i = P_0\rho^i, \text{ for } i = 1, 2, \ldots.$$

$$P_0 = \frac{1}{1+\sum_{i=1}^{\infty}\rho^i} = \frac{1}{\sum_{i=0}^{\infty}\rho^i} = 1-\rho \quad (normalization)$$

ρ = (1-P_0) is valid in general for G/G/1
P_0 > 0 is a stability condition

Let **P(z)** denote the PGF of the state probability distribution, P_i:

$$P(z) = \sum_{i=0}^{\infty}(1-\rho)\rho^i z^i = \frac{1-\rho}{1-z\rho}$$

The M/M/1 Queue (cont'd)

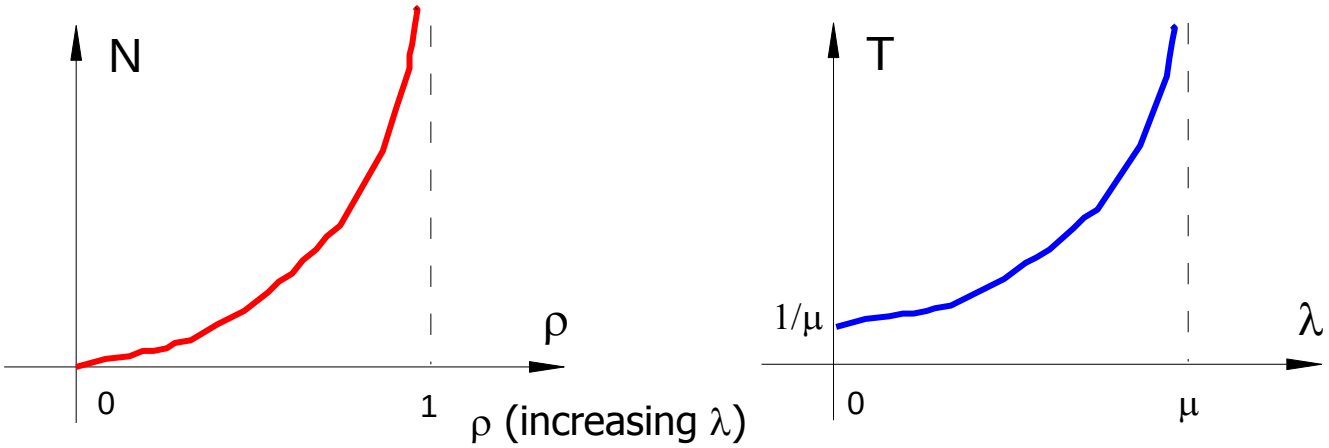
- **The state probability is geometrically distributed**.
- **The ergodicity condition that assures stability ($\rho < 1$ Erlang) entails $P_0 > 0$**: if the queue is stable it must be empty sometimes.
- The mean number of requests in the queue and the mean delay (Little theorem) are:

$$N = \sum_{i=0}^{\infty} i(1-\rho)\rho^i = \left.\frac{dP(z)}{dz}\right|_{z=1} = \frac{\rho}{1-\rho} \qquad T = \frac{N}{\lambda} = \frac{1}{\mu - \lambda}$$

- The ergodicity condition for stability allows that both N and T have finite values.
- The state probability distribution (and, hence, N) as well as T do not depend on the service discipline (insensitivity property).

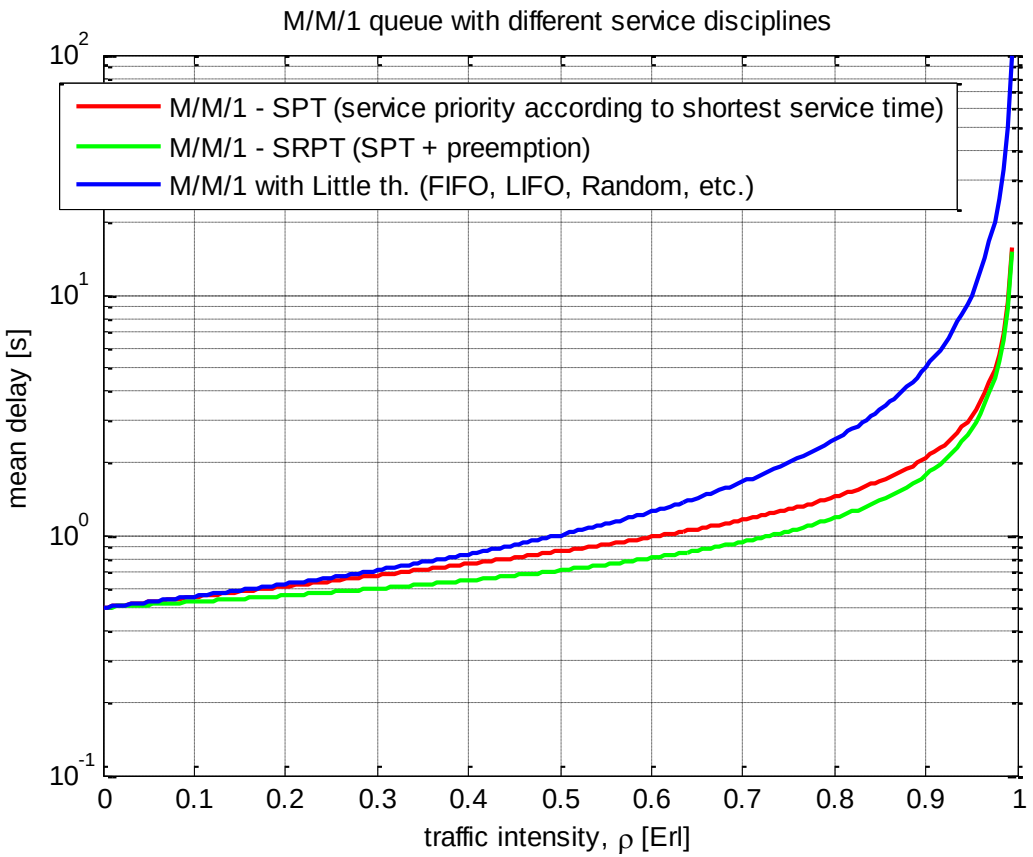
The M/M/1 Queue (cont'd)

Behaviors of N and T:



- When the traffic intensity approaches the maximum (1 Erl), the mean number of requests in the system and the mean system delay tend to infinity.
- When the traffic intensity tends to 0, the system tends to be empty and the mean system delay approaches $1/\mu$ ($\equiv$ mean service time).
- **Important consideration (performance – efficiency trade-off)**: increasing the utilization of the resources we have necessarily an increase of (buffer) congestion and delays.

M/M/1 Queue with Different Service Disciplines



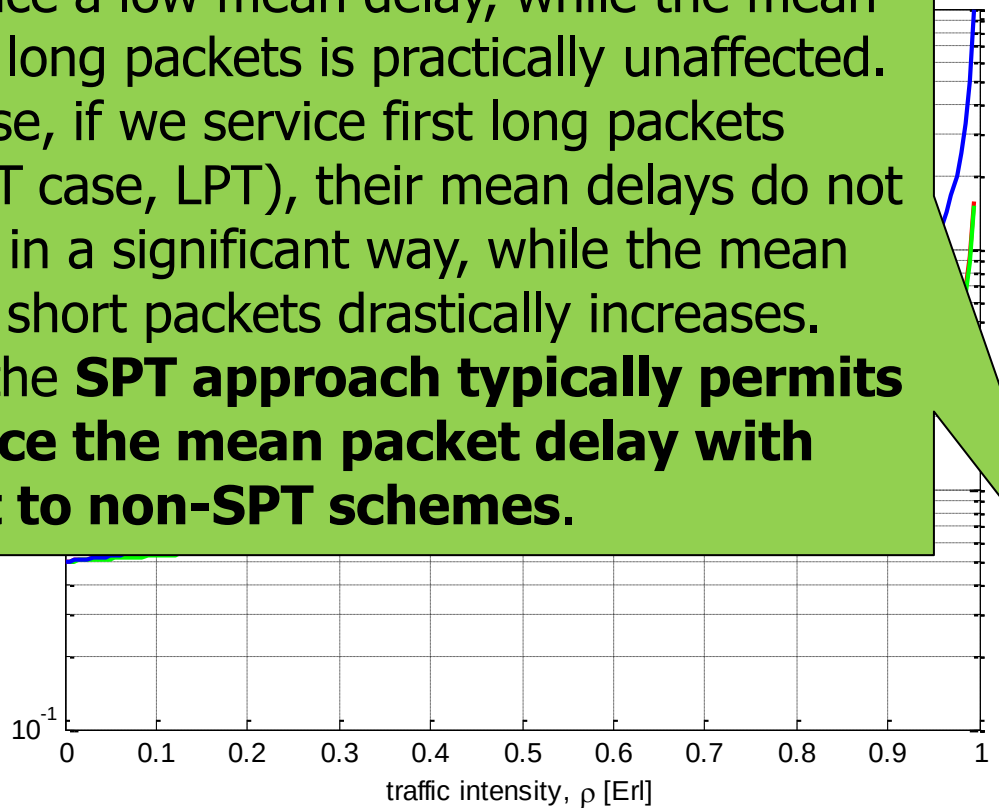
SPT = *Shortest Processing Time*

SRPT = *Shortest Remaining Processing Time* (preemptive)

For SPT and SRPT service disciplines (based on the knowledge of the service duration of each request) **the insensitivity property cannot be applied: M/M/1-SPT has a different state probability distribution (and mean number of requests) than M/M/1-FIFO.**

M/M/1 Queue with Different

Let us consider a queue where packets of two sizes arrive: very short packets (*non-congestive traffic*) and very long packets. If we service first short packets (SPT case), they experience a low mean delay, while the mean delay of long packets is practically unaffected. Otherwise, if we service first long packets (non-SPT case, LPT), their mean delays do not improve in a significant way, while the mean delay of short packets drastically increases. Hence, the **SPT approach typically permits to reduce the mean packet delay with respect to non-SPT schemes**.



SPT = *Shortest Processing Time*

SRPT = *Shortest Remaining Processing Time* (preemptive)

For SPT and SRPT service disciplines (based on the knowledge of the service duration of each request) **the insensitivity property cannot be applied: M/M/1-SPT has a different state probability distribution (and mean number of requests) than M/M/1-FIFO.**

Intuitive Comparison of Different Scheduling Schemes

Hp) We have some packets in a buffer and they have two different sizes (transmission times): 1 and 10. Different service orders can be applied to these packets.

FIFO order

- 1
- 11
- 12
- 13
- 14

SPT order

- 1
- 2
- 3
- 4
- 14

LPT order

- 10
- 11
- 12
- 13
- 14

Mean packet delay for the 5 packets according to their service order in 3 different cases.

Mean delay =10.2

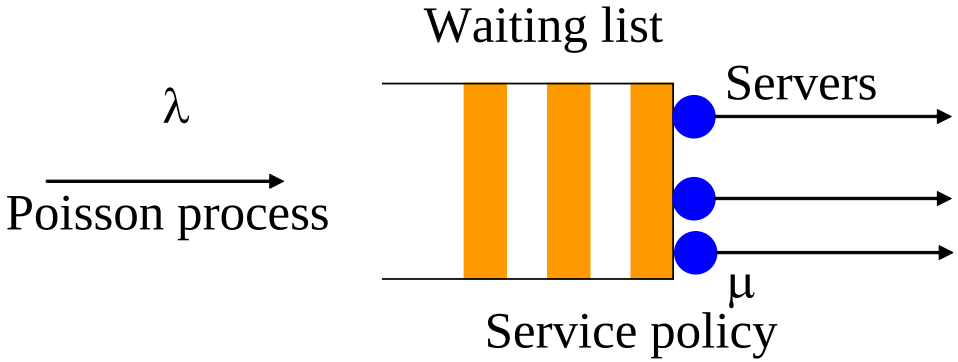
Mean delay =4.8

Mean delay = 12

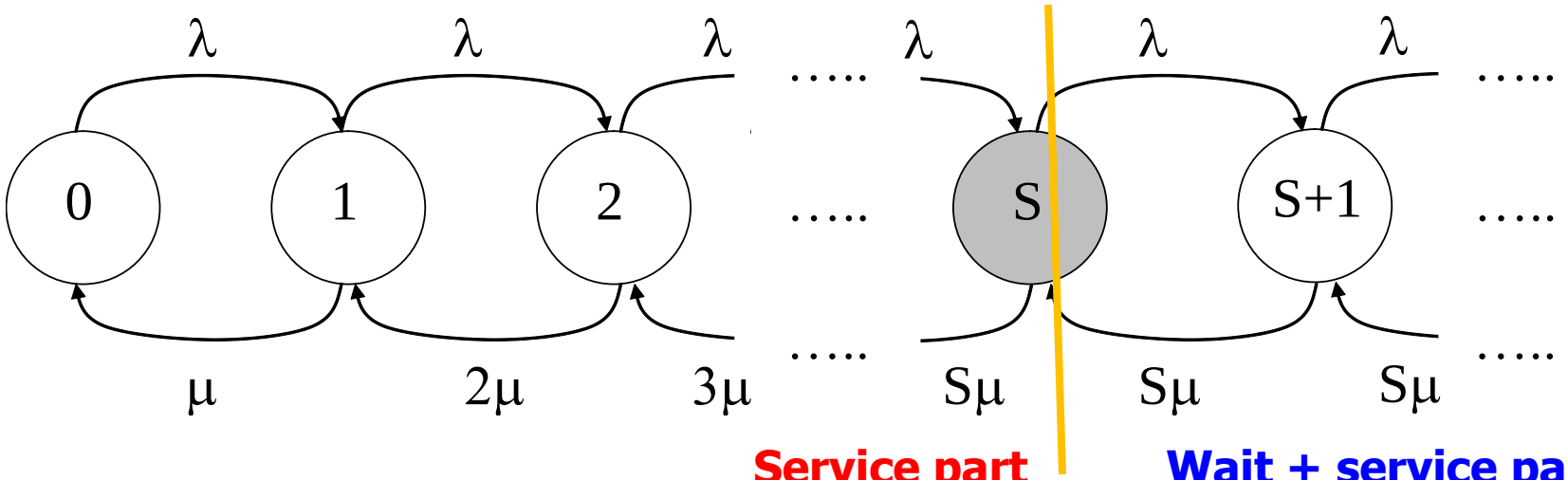
The M/M/S Queue

- We consider a queue with a Poisson arrival process (mean rate λ), exponential service time (mean rate μ) and S servers. The birth rate is always equal to λ ($\forall$ i), while the death rate depends on the state.
  - For the generic state $i \leq S$, there are i simultaneously-served requests; by invoking the memoryless property of the exponential distribution, each served request has a residual duration exponentially-distributed with mean rate μ. Therefore, the time for the death transition to the state $i - 1$ is the minimum among i times exponentially-distributed with mean rate μ; this minimum is still exponentially-distributed with mean rate $\mu_i = i\mu$.
  - For a generic state with $i > S$, the mean completion rate is μ_i equal to $S\mu$ (we have saturated the capacity for states $i \geq S$).

The M/M/S Queue (cont'd)



- Markov chain model for the M/M/S queue:



The M/M/S Queue (cont'd)

- The ergodicity condition for the stability of the queue is $\lambda/(S\mu) < 1 \Rightarrow$ traffic intensity $\rho = \lambda/\mu < S$ Erlangs; note that ρ/S is the utilization factor of a single server.
- From cut equilibrium and normalization conditions, we have:

$$\text{cut } 1 \text{ balance}: \ \lambda P_0 = \mu P_1 \Rightarrow P_1 = \frac{\lambda}{\mu} P_0 = \rho P_0$$

$$\text{cut } 2 \text{ balance}: \ \lambda P_1 = 2\mu P_2 \Rightarrow P_2 = \frac{\lambda}{2\mu} P_1 = \frac{\rho^2}{2} P_0$$

.....

$$\text{cut } S \text{ balance}: \ \lambda P_{S-1} = S\mu P_S \Rightarrow P_S = \frac{\lambda}{S\mu} P_{S-1} = \frac{\rho^s}{S!} P_0$$

$$\text{cut } S+1 \text{ balance}: \ \lambda P_S = S\mu P_{S+1} \Rightarrow P_{S+1} = \frac{\lambda}{S\mu} P_S = \frac{\rho^{s+1}}{S\,S!} P_0$$

…

$$P_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \prod_{n=1}^{i} \frac{\lambda_{n-1}}{\mu_n}} = \frac{1}{\sum_{i=0}^{S-1} \frac{\rho^i}{i!} + \sum_{i=S}^{\infty} \frac{\rho^i}{S!\,S^{i-S}}} = \frac{1}{\sum_{i=0}^{S-1} \frac{\rho^i}{i!} + \frac{S\rho^S}{S!(S-\rho)}}$$

The M/M/S Queue (cont'd)

- State probabilities P_n need to be calculated in an **iterative way** due to both the presence of factorial terms and, in general, the ratios of very high numbers when n is sufficiently high. The recursive process starts by computing P_1/P_0; this result is used to compute P_2/P_0 , and so on. The terms P_i/P_0 are progressively summed together to derive P_0 by means of the normalization condition.

- The probability that a new arrival finds all the servers busy (thus it is queued), P_C, is given by:

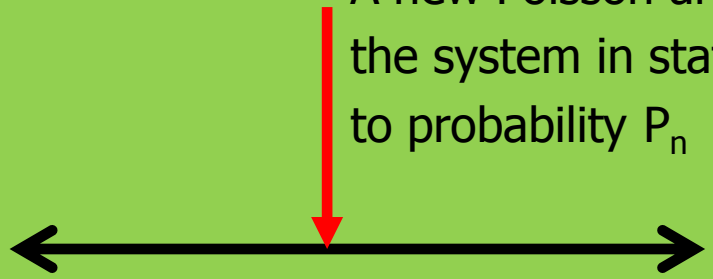
$$P_C = \sum_{i=S}^{\infty} P_i = P_0 \sum_{i=S}^{\infty} \frac{P_i}{P_0} = \frac{\dfrac{S\rho^S}{S!(S-\rho)}}{\displaystyle\sum_{i=0}^{S-1} \frac{\rho^i}{i!} + \frac{S\rho^S}{S!(S-\rho)}}$$

Erlang-C formula

This formula depends on the application of an important property for Poisson processes, that is the PASTA property described in next slide.

The PASTA Property (Only for Poisson Arrivals)

- The PASTA (Poisson Arrivals See Time Averages) property was defined by R. W. Wolff.
  - For M/-/-/- queues where the arrival process is Poisson, the **probability** that an arrival finds the chain in the state n is equal to the **time percentage** that the chain is in the state n (this is equal to the steady state probability P_n due to the ergodicity).



Time spent in the generic state n:

The percentage of time for which the system is in the state n is equal to the state probability P_n

March-April 1982.

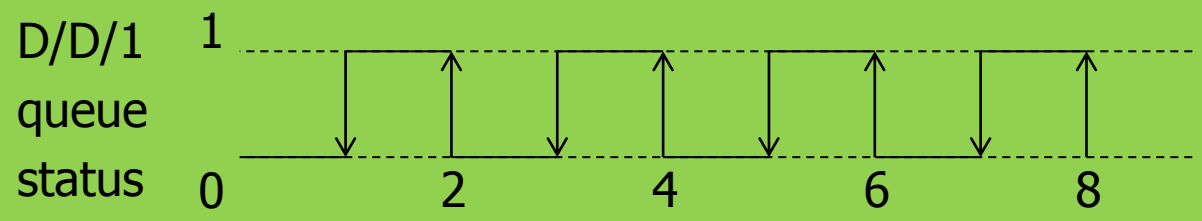
The PASTA Property (Only for Poisson Arrivals)

- The PASTA (Poisson Arrivals See Time Averages) property was defined by R. W. Wolff.
  - For M/-/-/- queues where the arrival process is Poisson, the **probability** that an arrival finds the chain in the state n is equal to the **time percentage** that the chain is in the state n (this is equal to the steady state probability P_n due to the ergodicity).
  - The PASTA property does not apply to state-dependent Poisson arrival processes or to non-Poisson arrival processes.
  - **The PASTA property is not generally true.** For instance, let us consider a D/D/1 queuing system, which is empty at time 0, with arrivals at times 1, 3, 5 s and with service times 1 s (there is a cycle length of 2 s): every arriving customer finds an empty system, whereas the fraction of time the system is empty is ½.

R. W. Wolff, "Poisson Arrivals See Time Averages", *Operational Research*, Vol. 30, No. 2, March-April 1982.

The PASTA Property (Only for Poisson Arrivals)

- The PASTA (Poisson Arrivals See Time Averages)



The new arrivals find an empty system so that for them it is like $P_0 = 1$. However, the queue is empty for 50% of time, thus yielding $P_0 = 0.5$.

ual to
qual to
son

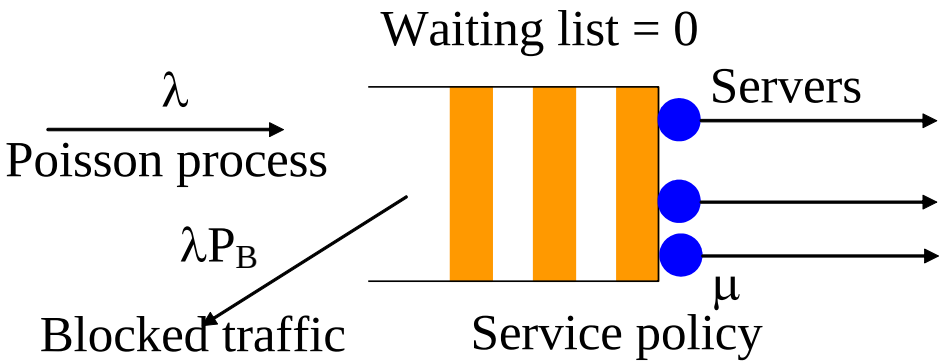
- **The PASTA property is not generally true.** For instance, let us consider a D/D/1 queuing system, which is empty at time 0, with arrivals at times 1, 3, 5 s and with service times 1 s (there is a cycle length of 2 s): every arriving customer finds an empty system, whereas the fraction of time the system is empty is ½.

R. W. Wolff, "Poisson Arrivals See Time Averages", *Operational Research*, Vol. 30, No. 2, March-April 1982.

The M/M/S/S Queue

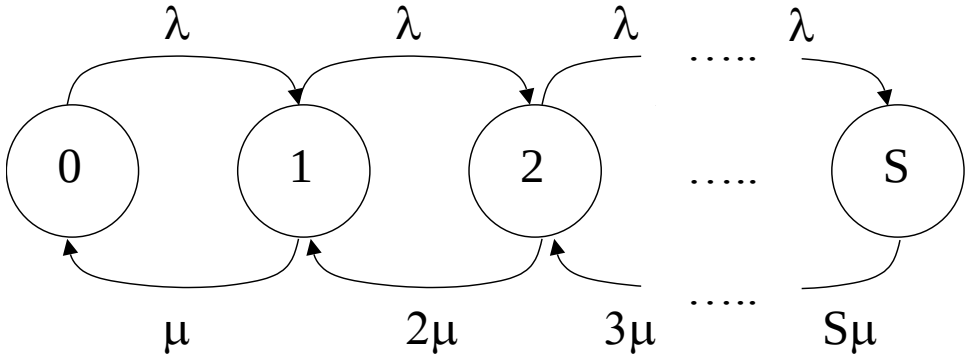
- In this queue we have only S rooms in the system and S servers; there are no waiting rooms in this queue. If a new arrival finds the system busy (i.e., with S requests in service) it is not admitted (**blocked**) in the system.

- Let P_B denote the probability that a new arrival finds the system busy and is blocked. Hence, we can prove that λP_B denotes the 'refused' traffic flow and $\lambda(1-P_B)$ denotes the 'admitted' traffic flow into the queue.



The M/M/S/S Queue (cont'd)

- This queue has S+1 states from i = 0 to S. Birth and death rates are derived from the M/M/S queue. **The ergodicity condition for the queue stability is always fulfilled** since there is a finite number of states.



- By exploiting the same derivations made in the M/M/S case, we can obtain the following state probability distribution:

$$P_i = \frac{\rho^i}{i!} P_0, \text{ where } P_0 = \frac{1}{\sum_{i=0}^{S} \frac{\rho^i}{i!}}$$

This is a truncated Poisson distribution.

The M/M/S/S Queue (cont'd)

- Since the mean arrival rate does not depend on the state, applying the PASTA property we obtain the probability that a new request is blocked and refused due to the unavailability of rooms in the queue, P_B, as the **probability that the queue is in the state S, P_S**:

$$P_B \equiv P_S = \frac{\rho^S}{S! \sum_{i=0}^{S} \frac{\rho^i}{i!}}$$

Erlang-B formula

Iterative Method for Erlang-B Computation

- The Erlang-B formula depends on S and ρ. This formula cannot be directly computed when the number of servers, S, is high due to the presence of factorial terms.

- This is the reason why an **iterative method** has been developed to compute the Erlang-B formula for increasing number of resources S:

$$P_b(\rho, 0) = 1$$

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

The M/M/S/S Queue (cont'd)

- The mean arrival rate (arrivals accepted into the system) is obtained as:

$$\bar{\lambda} = \sum_{i=0}^{S-1} \lambda_i P_i = \lambda \sum_{i=0}^{S-1} P_i = \lambda(1 - P_S)$$

- Hence, there is difference between the mean input arrival rate λ and the mean rate $\bar{\lambda}$ of arrivals accepted into the system (this is the rate to be used in the Little formula).

- The mean number of requests N in the system can be derived as:

$$N = \sum_{i=1}^{S} iP_i = \sum_{i=1}^{S} i \frac{\rho^i}{i!} P_0 = \rho \sum_{i=0}^{S-1} \frac{\rho^i}{i!} P_0 = \rho \sum_{i=0}^{S-1} P_i = \rho(1 - P_S)$$

- We can apply the Little theorem as:

$$T = \frac{N}{\bar{\lambda}} = \frac{\rho(1 - P_S)}{\lambda(1 - P_S)} = \frac{1}{\mu}$$

T coincides with the mean service time since there is no waiting time in this queue.

Erlang-B Table and its Use in Traffic Engineering

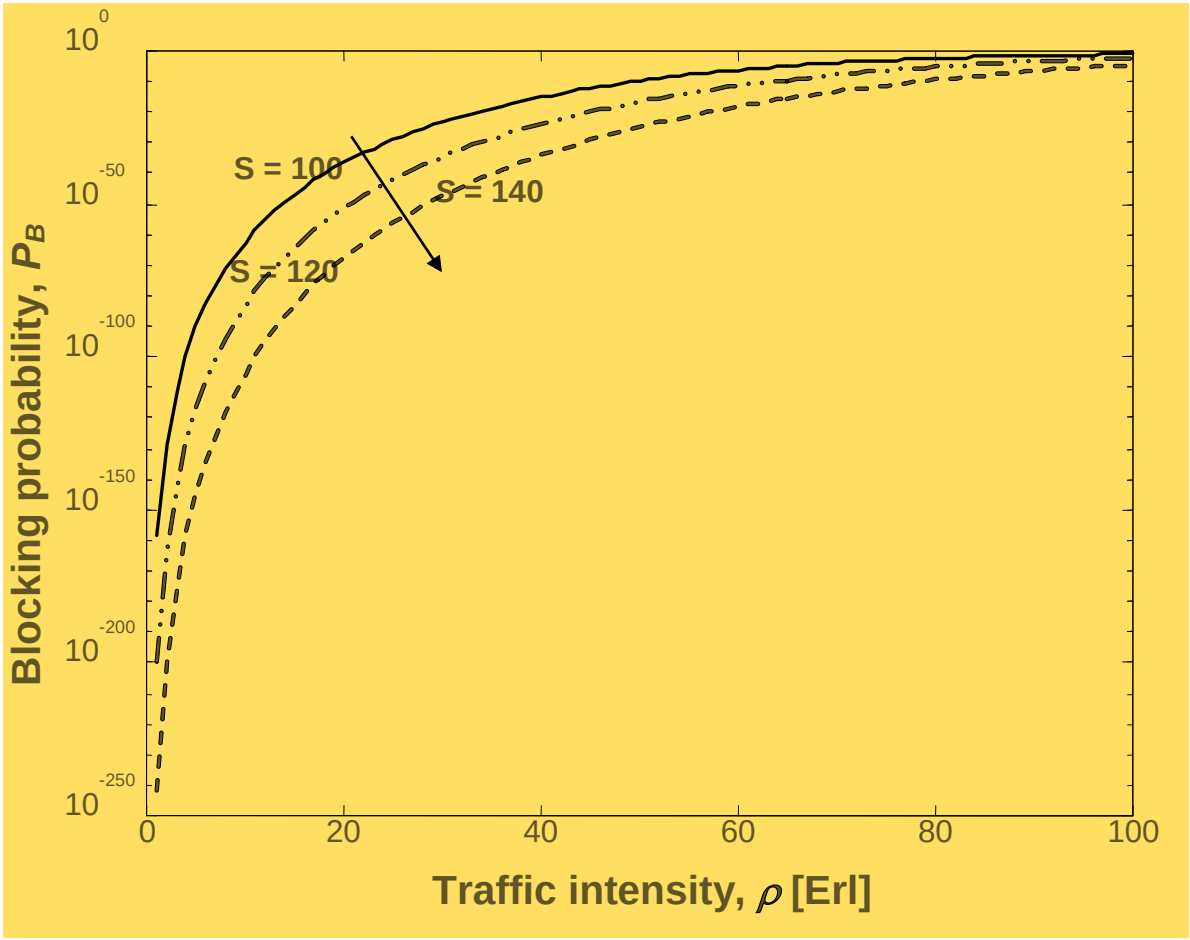
Servers	Bloking probability												
S	1.0%	1.2%	1.5%	2%	3%	5%	7%	10%	15%	20%	30%	40%	50%
1	.0101	.0121	.0152	.0204	.0309	.0526	.0753	.111	.176	.250	.429	.667	1.00
2	.153	.168	.190	.223	.282	.381	.470	.595	.796	1.00	1.45	2.00	2.73
3	.455	.489	.535	.602	.715	.899	1.06	1.27	1.60	1.93	2.63	3.48	4.59
4	.869	.922	.992	1.09	1.26	1.52	1.75	2.05	2.50	2.95	39	5.02	6.50
5	1.36	1.43	1.52	1.66	1.88	2.22	2.50	2.88	3.45	4.01	5.19	6.60	8.44
6	1.91	2.00	2.11	2.28	2.54	2.96	3.30	3.76	4.44	5.11	6.51	8.19	10.4
7	2.50	2.60	2.74	2.94	3.25	3.74	4.14	4.67	5.46	6.23	7.86	9.80	12.4
8	3.13	3.25	3.40	3.63	3.99	4.54	5.00	5.60	6.50	7.37	9.21	11.4	14.3
9	3.78	3.92	4.09	4.34	4.75	5.37	5.88	6.55	7.55	8.52	10.6	13.0	16.3
10	4.46	4.61	4.81	5.08	5.53	6.22	6.78	7.51	8.62	9.68	12.0	14.7	18.3
11	5.16	5.32	5.54	5.84	6.33	7.08	7.69	8.49	9.69	10.9	13.3	16.3	20.3
12	5.88	6.05	6.29	6.61	7.14	7.95	8.61	9.47	10.8	12.0	14.7	18.0	22.2
13	6.61	6.80	7.05	7.40	7.97	8.83	9.54	10.5	11.9	13.2	16.1	19.6	24.2
14	7.35	7.56	7.82	8.20	8.80	9.73	10.5	11.5	13.0	14.4	17.5	21.2	26.2
15	8.11	8.33	8.61	9.01	9.65	10.6	11.4	12.5	14.1	15.6	18.9	22.9	28.2
16	8.88	9.11	9.41	9.83	10.5	11.5	12.4	13.5	15.2	16.8	20.3	24.5	30.2
17	9.65	9.89	10.2	10.7	11.4	12.5	13.4	14.5	16.3	18.0	21.7	26.2	32.2
18	10.4	10.7	11.0	11.5	12.2	13.4	14.3	15.5	17.4	19.2	23.1	27.8	34.2
19	11.2	11.5	11.8	12.3	13.1	14.3	15.3	16.6	18.5	20.4	24.5	29.5	36.2
20	12.0	12.3	12.7	13.2	14.0	15.2	16.3	17.6	19.6	21.6	25.9	31.2	38.2
21	12.8	13.1	13.5	14.0	14.9	16.2	17.3	18.7	20.8	22.8	27.3	32.8	40.2
22	13.7	14.0	14.3	14.9	15.8	17.1	18.2	19.7	21.9	24.1	28.7	34.5	42.1
23	14.5	14.8	15.2	15.8	16.7	18.1	19.2	20.7	23.0	25.3	30.1	36.1	44.1
24	15.3	15.6	16.0	16.6	17.6	19.0	20.2	21.8	24.2	26.5	31.6	37.8	46.1
25	16.1	16.5	16.9	17.5	18.5	20.0	21.2	22.8	25.3	27.7	33.0	39.4	48.1
26	17.0	17.3	17.8	18.4	19.4	20.9	22.2	23.9	26.4	28.9	34.4	41.1	50.1
27	17.8	18.2	18.6	19.3	20.3	21.9	23.2	24.9	27.6	3			
28	18.6	19.0	19.5	20.2	21.2	22.9	24.2	26.0	28.7	3			
29	19.5	19.9	20.4	21.0	22.1	23.8	25.2	27.1	29.9	3			

Problem:

- To determine the number of servers S with input traffic intensity of 7 Erlang and requirement of blocking probability lower than or equal to 2%.
- Using the table, S = 13 servers are needed.

$$P_b(\rho, S) = \frac{\rho^S}{S! \sum_{n=0}^{S} \frac{\rho^n}{n!}}$$

Erlang-B Formula Behavior



The Erlang-B Formula in Extended Cases

- It is possible to prove that the M/M/S/S state probability distribution is also valid for an **M/G/S/S queue** with the same input traffic intensity; this is another **insensitivity property** concerning the statistics of the service time (only the mean value has impact through the input traffic intensity ρ).
  - **The Erlang-B formula can also be adopted in the general M/G/S/S case**. This is an important generalization of the Erlang-B formula, since in current systems sessions arrive according to Poisson processes, but their duration is not exponentially distributed.

S. M. Ross. *Stochastic Processes*. John Wiley and Sons, 1983.

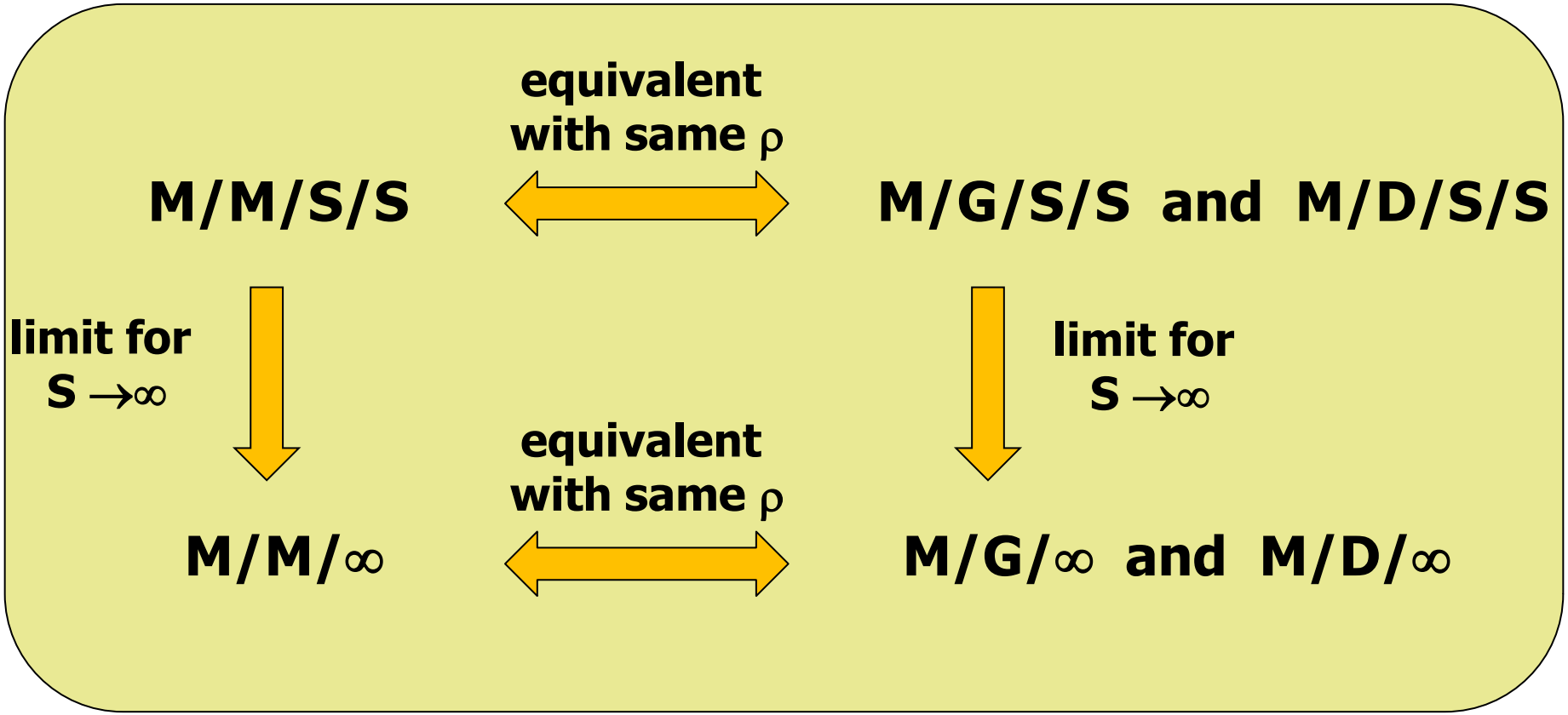
M/M/∞ and M/G/∞ Queues

- The M/M/∞ queue is the limiting case of the M/M/S/S queue (or the M/M/S case) for S $\rightarrow \infty$. Similarly, the M/G/∞ queue can be seen as the limiting case of the M/G/S/S queue for S $\rightarrow \infty$, and, therefore, can be studied by means of the equivalent M/M/∞ queue (i.e., with the same traffic intensity ρ).
- We use the state probability distribution of the M/M/S/S case and we take the limit for S $\rightarrow \infty$ so that we solve the M/M/∞ queue as:

$$P_i = \frac{\rho^i}{i!} P_0 \quad \text{where} \quad P_0 = \frac{1}{\sum_{i=0}^{\infty} \frac{\rho^i}{i!}} = e^{-\rho}$$

- The state probability of the M/M/∞ (M/G/∞) queue is **Poisson-distributed**.
- These queues are suitable to model traffic sources.

Equivalencies for M/M/S/S, M/M/∞,M/G/S/S, and M/G/∞



The M/D/∞ Example

- Let us consider an **S-Aloha case**. There is a (total) Poisson process with mean rate Λ and a fixed service duration T (= packet transmission time). We know that the number of arriving packets on a slot is according to a **Poisson distribution** with parameter G = ΛT; this is consistent with an **M/D/∞ model** of the system where G = ρ. Then, the probability distribution of the number of arriving packets on a slot results as:

$$P_i = \frac{\rho^i}{i!} e^{-\rho}$$

Queue Examples and Special Cases: a Summary

- Multiplexer models (single server): M/M/1/K, M/M/1, M/G/1, M/D/1
- Trunking models (classical telephony, S servers): M/M/S/S, M/G/S/S
- User 'application' traffic (infinite servers): M/M/∞, M/G/∞, M/D/∞
- Special cases:
  - **Bulk arrivals**: more than one arrival can occur at a given instant (compound arrival process). The symbol denoting the arrival process has an exponent, describing the statistics of the bulk arrivals. For instance, $\mathbf{M^{[Geom]}}$/G/1 for a geometrical number of 'objects' arriving together. This could be true in the following cases: (i) **IP packets fragmented** to fit layer 2 frame format (MAC layer queue); (ii) **Web page** with many objects.
  - **Batched service**: some arrived objects are serviced together (e.g., TDMA transmissions). The letter of the service process has an exponent describing the length of the batch. For instance, M/$\mathbf{D^{[b]}}$/1, for a deterministic service with b 'objects' together. This is the case of a **TDMA transmission system** with b slots per frame allocated to the service of packet arrivals.

Queue Examples and Special Cases: a Summary

Node of a telephone network or PBX

- Multiplexer models (single server): M/M/1/
- Trunking models (classical telephony, S servers): M/M/S/S, M/G/S/S
- User 'application' traffic (infinite servers): M/M/∞, M/G/∞, M/D/∞
- Special cases:
  - **Bulk arrivals**: more than one arrival can occur at a given instant (compound arrival process). The symbol denoting the arrival process has an exponent, describing the statistics of the bulk arrivals. For instance, $\mathbf{M^{[Geom]}}$/G/1 for a geometrical number of 'objects' arriving together. This could be true in the following cases: (i) **IP packets fragmented** to fit layer 2 frame format (MAC layer queue); (ii) **Web page** with many objects.
  - **Batched service**: some arrived objects are serviced together (e.g., TDMA transmissions). The letter of the service process has an exponent describing the length of the batch. For instance, M/$\mathbf{D^{[b]}}$/1, for a deterministic service with b 'objects' together. This is the case of a **TDMA transmission system** with b slots per frame allocated to the service of packet arrivals.

Queue Examples and Special Cases: a Summary

- Multiplexer models (single server): M/M/1, [illegible]
- Trunking models (classical telephony, S servers): [illegible]
- User 'application' traffic (infinite servers): M/M/∞, M/G/∞, M/D/∞
- Special cases:
  - **Bulk arrivals**: more than one arrival can occur at a given instant (compound arrival process). The symbol denoting the arrival process has an exponent, describing the statistics of the bulk arrivals. For instance, $\mathbf{M^{[Geom]}}$/G/1 for a geometrical number of 'objects' arriving together. This could be true in the following cases: (i) **IP packets fragmented** to fit layer 2 frame format (MAC layer queue); (ii) **Web page** with many objects.
  - **Batched service**: some arrived objects are serviced together (e.g., TDMA transmissions). The letter of the service process has an exponent describing the length of the batch. For instance, M/$\mathbf{D^{[b]}}$/1, for a deterministic service with b 'objects' together. This is the case of a **TDMA transmission system** with b slots per frame allocated to the service of packet arrivals.

S-Aloha access protocol

Exercises

Exercise #1

A radio link adopts four equivalent parallel transmitters for redundancy reasons. The operational characteristics of the transmitters require that each of them be switched off (for maintenance or recovery actions) according to a Poisson process with a mean interarrival time λ^{-1} of 1 month. The technician that performs maintenance and recovery actions requires an exponentially-distributed time with mean duration μ^{-1} of 12 hours in order to fix the problem. We consider that two technicians are available. This exercise requires:

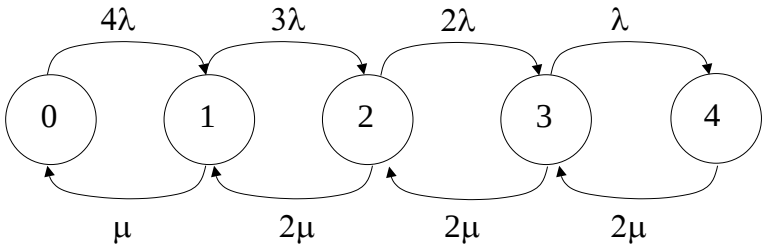
- To define a suitable model for the system;
- To determine the probability distribution of the number of non-working transmitters at a generic instant;
- To express the probability that no transmitter is working on this radio link.

Solution of Exercise #1

The system can be modeled as a Markov chain with **five states denoting the number of non-working transmitters**: 0, 1, …, 4. We exploit the **memoryless property of the exponential distribution** for both the interarrival times of the recovery actions for a transmitter with mean rate λ (= 1 action/month) and the repairing times with mean rate μ (= 1/12 repairing/hour). The transition from the generic state j ($0 \leq j < 4$) to the state with j+1 non-working transmitters is the minimum among 4–j **independent times** with exponential distribution and mean rate λ; such time is still exponentially distributed with mean rate $(4-j)\lambda$. As for the transitions from states j ($1 < j \leq 4$) to states with j–1 non-working transmitters, these are performed after time intervals that are the minimum between two independent, exponentially distributed times with mean rate μ (i.e., the times required by the two technicians to fix their problems); hence, these transitions occur after a time interval exponentially-distributed with mean rate 2μ. Of course the transition from state j = 1 to state j = 0 has an exponentially-distributed time with mean rate μ.

Solution of Exercise #1 (cont'd)

We obtain a Markov chain model like that used for an M/M/2/4/4 queue:



Since we consider a finite-state chain, there are no stability problems.

We can state cut equilibrium conditions:

$$\text{cut 1 balance}: \; 4\lambda P_0 = \mu P_1 \Rightarrow P_1 = 4\frac{\lambda}{\mu}P_0$$

$$\text{cut 2 balance}: \; 3\lambda P_1 = 2\mu P_2 \Rightarrow P_2 = \frac{3\lambda}{2\mu}P_1 = \frac{4\times 3}{2}\left(\frac{\lambda}{\mu}\right)^2 P_0$$

$$\text{cut 3 balance}: \; 2\lambda P_2 = 2\mu P_3 \Rightarrow P_3 = \frac{2\lambda}{2\mu}P_2 = \frac{4\times 3\times 2}{2^2}\left(\frac{\lambda}{\mu}\right)^3 P_0$$

$$\text{cut 4 balance}: \; \lambda P_3 = 2\mu P_4 \Rightarrow P_4 = \frac{\lambda}{2\mu}P_3 = \frac{4\times 3\times 2\times 1}{2^3}\left(\frac{\lambda}{\mu}\right)^4 P_0$$

$$\text{in general} \quad P_n = \frac{4!}{2^{n-1}(4-n)!}\left(\frac{\lambda}{\mu}\right)^n P_0 \quad \text{for } 0 < n \le 4$$

Solution of Exercise #1 (cont'd)

Finally, we impose the normalization condition:

$$P_0 = \frac{1}{1 + \sum_{n=1}^{4} \frac{4!}{2^{n-1}(4-n)!}\left(\frac{\lambda}{\mu}\right)^n}$$

The percentage of time for which no transmitter is working, is given by P_4.

Exercise #2

We have a transmission line to send the messages that arrive at a buffer with infinite capacity. Each message can wait for service for a maximum time (deadline); otherwise it is discarded from the buffer. We model the maximum waiting time of a message as a random variable with exponential distribution and mean rate γ. Messages arrive according to a Poisson process with mean rate λ and their transmission time is exponentially distributed with mean rate μ. We need to determine:

- A suitable queuing model for the system;
- The mean number of messages in the transmission buffer.

Solution of Exercise #2

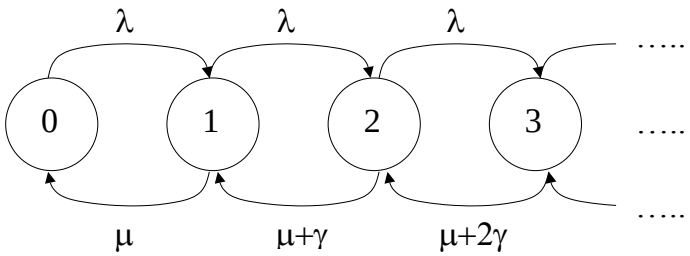
If messages have no deadline, this system can be described by a classical M/M/1 queue with mean arrival rate λ and mean completion rate μ.

In our case we model the system with a chain where the state denotes the number of messages in the system. The mean arrival rate is λ; but some considerations have to be made for the transitions from state j to state $j - 1$.

When there is a served message and another is in the waiting list, this message can wait for receiving service for a time exponentially distributed with mean rate γ. Therefore, **the transition from state j = 2 to state j = 1 is characterized by the minimum between two times exponentially distributed with mean rates μ (due to a service completion) and γ (due to a deadline expiration), respectively**. Hence, such transition occurs after an exponentially-distributed time with mean rate $\mu + \gamma$. In general, the transition from state j to state $j - 1$ occurs with mean rate $\mu + (j - 1)\gamma$.

Solution of Exercise #2 (cont'd)

We obtain a Markov chain model of the M/M/... type:



This Markov chain is **always stable**, since the ergodicity condition is definitely verified: $\lambda/[\mu + (j-1)\gamma] < 1$ Erl for any j greater than a given value. By means of the cut equilibrium conditions, we have:

$$\text{cut 1 balance}: \ \lambda P_0 = \mu P_1 \Rightarrow P_1 = \frac{\lambda}{\mu} P_0$$

$$\text{cut 2 balance}: \ \lambda P_1 = (\mu + \gamma) P_2 \Rightarrow P_2 = \frac{\lambda}{\mu + \gamma} P_1 = \frac{\lambda}{\mu} \frac{\lambda}{\mu + \gamma} P_0$$

$$\text{cut 3 balance}: \ \lambda P_2 = (\mu + 2\gamma) P_3 \Rightarrow P_3 = \frac{\lambda}{\mu + 2\gamma} P_2 = \frac{\lambda}{\mu} \frac{\lambda}{\mu + \gamma} \frac{\lambda}{\mu + 2\gamma} P_0$$

$$\text{in general} \ P_n = P_0 \frac{\lambda^n}{\prod_{i=0}^{n-1} (\mu + i\gamma)}$$

Solution of Exercise #2 (cont'd)

Finally, the normalization condition is:

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} \frac{\lambda^n}{\prod_{i=0}^{n-1} (\mu + i\gamma)}}$$

The mean number of messages in the buffer can be expressed as:

$$N = \sum_{n=1}^{\infty} nP_n = P_0 \sum_{n=1}^{\infty} n \, P_n / P_0$$

Exercise #3 on the Erlang-B Formula

- An *Internet Service Provider* (ISP) has to dimension a *Point of Presence* (POP) in the territory which can manage up to S simultaneous Internet connections (due to the limited number of available IP addresses or due to a limited processing capability). If a new Internet connection is generated by a user towards that POP and there are already S other connections in progress, the new connection request is blocked. We have to determine S guaranteeing that the blocking probability $P_B \leq 3\%$. We know that:
  - Each user generates Internet connections according to a Poisson process with mean rate λ
  - Internet sessions have a duration that is generally-distributed
  - Each POP subscriber is connected on average for 1 hour a day (thus contributing a load of about 41 mErlang)
  - We consider 100 subscribers/POP.
- Users are finite, but we apply the **conservative approximation of an infinite number of users at a parity of (max) offered load** ρ. Hence, we consider the queuing system of the M/G/S/S type that can be studied by the equivalent M/M/S/S queue with the same ρ: by means of the PASTA property, P_B is given by the Erlang-B formula.

$$P_b(\rho, S) = \frac{\rho^S}{S! \sum_{n=0}^{S} \frac{\rho^n}{n!}} \quad \text{where} \quad \rho = 100 \left[\frac{\text{users}}{\text{POP}}\right] \times 41 \left[\frac{\text{mErlang}}{\text{user}}\right] = 4.1 \,\text{Erlang}$$

According to the Erlang-B table S = 9

Exercise #3 on the Erlang-B Formula

- An *Internet Service Provider* (ISP) has to dimension a *Point of Presence* (POP) in the territory which can manage up to S simultaneous Internet connections (due to the limited number of available IP addresses or due to a limited processing capability). If a new Internet connection is generated by a user towards that POP and there are already S other connections in progress, the new connection request is blocked. We ...bability $P_B \leq 3\%$. We know

...ocess with mean rate λ

...s contributing a load of about 41

The M/G/S/S/P queue is approximated as M/G/S/S/∞ with peak load ρ.

Then, M/G/S/S/∞ is studied by means of the equivalent M/M/S/S queue with the same load ρ.

...ation of an infinite

...ρ. Hence, we consider the queuing system of the M/G/S/S type that can be studied by the equivalent M/M/S/S queue with the same ρ: by means of the PASTA property, P_B is given by the Erlang-B formula.

$$P_b(\rho, S) = \frac{\rho^S}{S! \sum_{n=0}^{S} \frac{\rho^n}{n!}} \quad \text{where} \quad \rho = 100 \left[\frac{\text{users}}{\text{POP}} \right] \times 41 \left[\frac{\text{mErlang}}{\text{user}} \right] = 4.1 \, \text{Erlang}$$

According to the Erlang-B table S = 9

Thank you!

giovanni.giambene@gmail.com