ECS455: Chapter 3

Poisson process and Markov chain

Office Hours:
BKD 3601-7
Tuesday 9:30-10:30
Friday 14:00-16:00

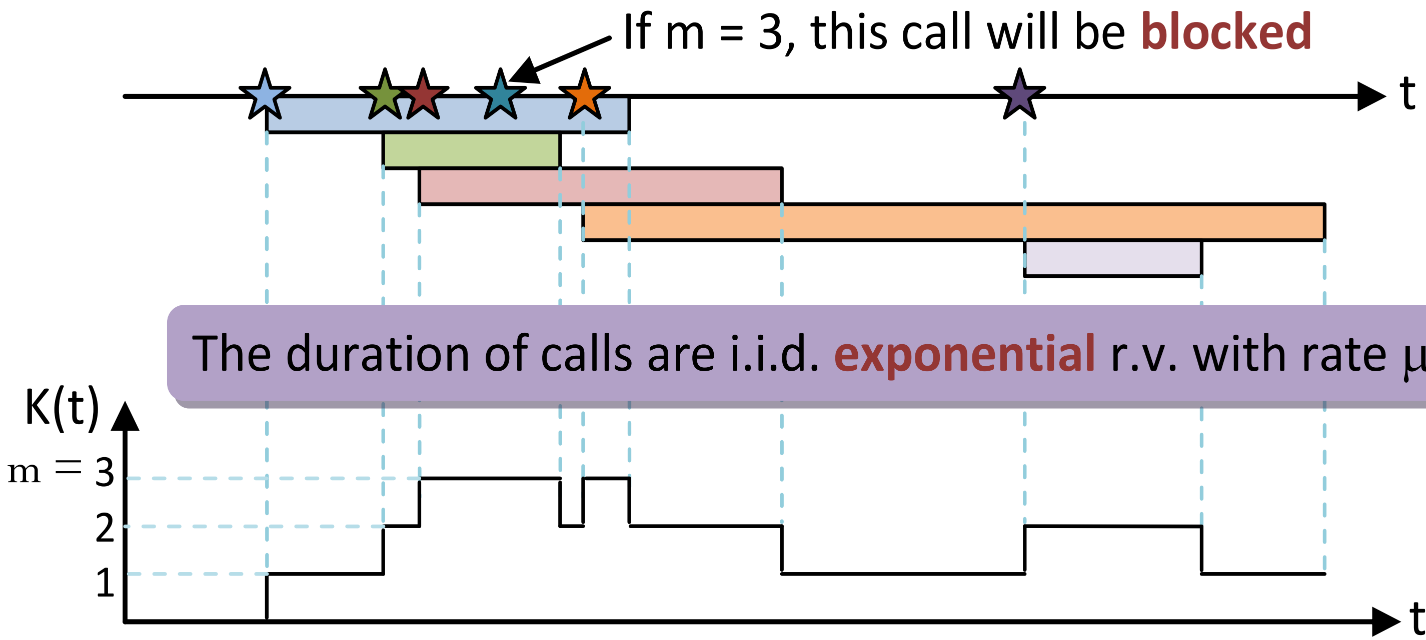
Dr.Prapun Suksompong
prapun.com/ecs455

M/M/m/m Assumption (Recap)

- **Blocked calls cleared**
  - Offers no queuing for call requests.
  - For every user who requests service, it is assumed there is no setup time and the user is given immediate access to a channel if one is available.
  - If no channels are available, the requesting user is blocked without access and is free to try again later.
- **Calls arrive as determined by a *Poisson process*.**
- There are memoryless arrivals of requests, implying that all users, including blocked users, may request a channel at any time.
- There are an infinite number of users (with finite overall request rate).
  - The finite user results always predict a smaller likelihood of blocking. So, assuming infinite number of users provides a conservative estimate.
- **The duration of the time that a user occupies a channel is *exponentially distributed***, so that longer calls are less likely to occur.
- There are m channels available in the trunking pool.
  - For us, m = the number of channels for a cell or for a sector

M/M/m/m Assumption (Con’t)

The call request process is **Poisson** with rate λ



The duration of calls are i.i.d. **exponential** r.v. with rate μ.

K(t) = “state” of the system

= the number of used channel at time *t*

We want to find out what proportion of time the system has $K = m$.

ECS455: Chapter 3

Poisson process and Markov chain

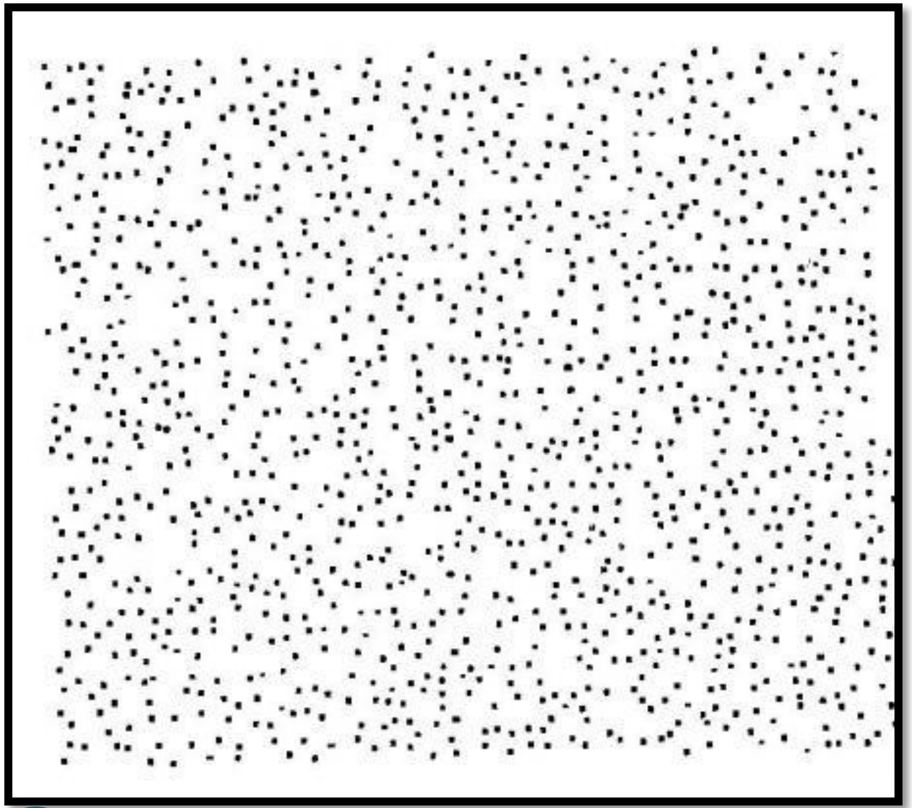
3.1 Poisson Process

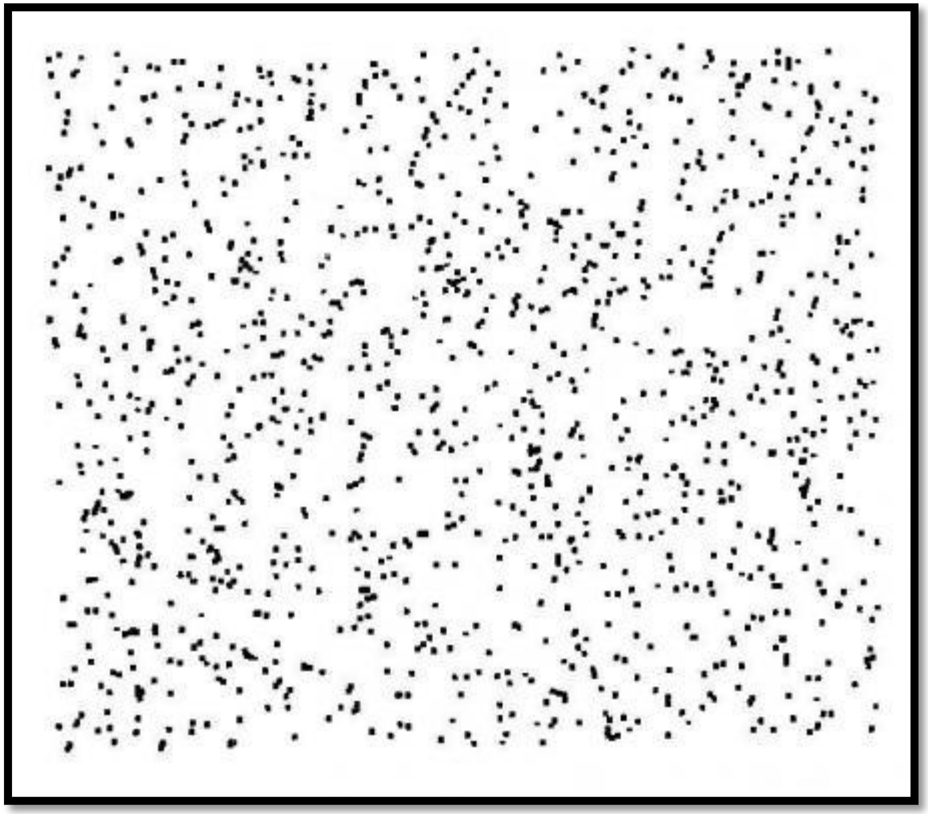
Dr.Prapun Suksompong
prapun.com/ecs455

Office Hours:
BKD 3601-7
Tuesday 9:30-10:30
Friday 14:00-16:00

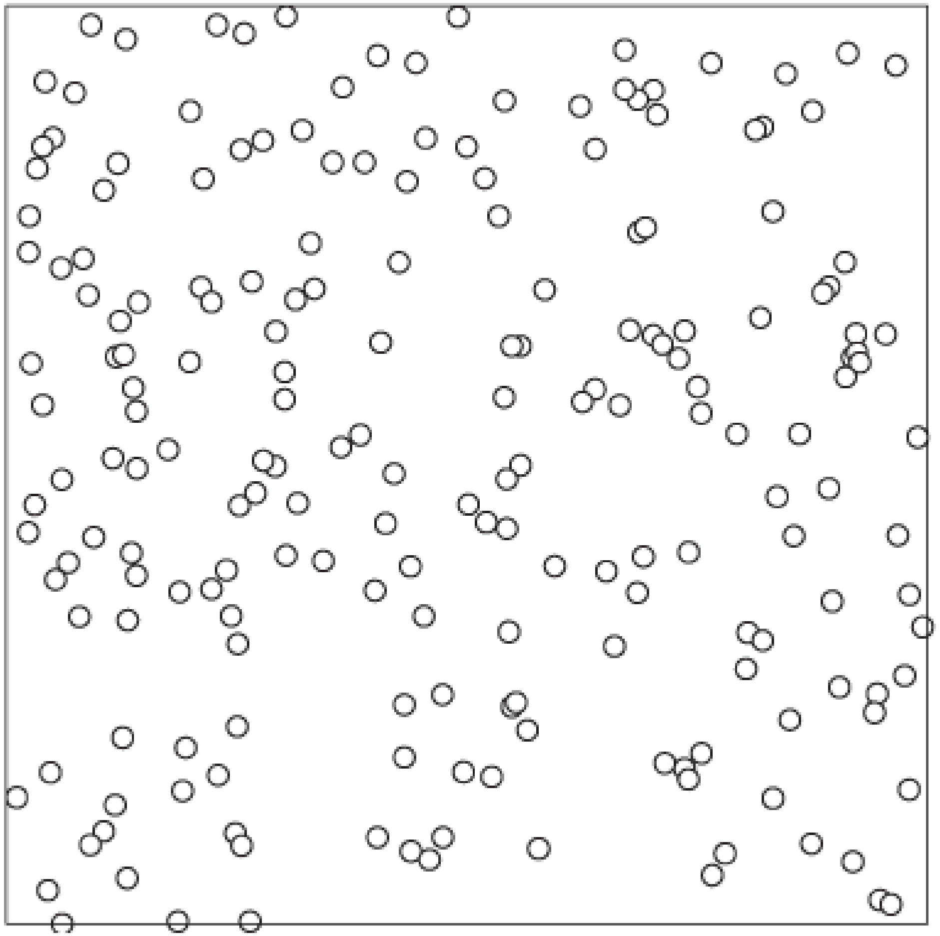
Poisson Process?

One of these is a realization of a two-dimensional Poisson point process and the other contains correlations between the points. One therefore has a real pattern to it, and one is a realization of a completely unstructured random process.





Poisson Process



All the structure that is visually apparent is imposed by our own sensory apparatus, which has evolved to be so good at discerning patterns that it finds them when they're not even there!

Poisson Process: Examples

- Sequence of times at which lightning strikes occur or mail carriers get bitten within some region
- Emission of particles from a radioactive source
- Arrival of
  - telephone calls at a switchboard or at an automatic phone-switching system
  - urgent calls to an emergency center
  - (filed) claims at an insurance company
  - incoming spikes (action potential) to a neuron in human brain

Poisson Process: Examples (2)

- Occurrence of
  - serious earthquakes
  - traffic accidents
  - power outages

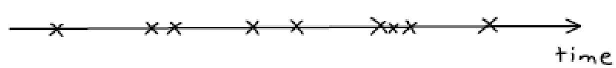
  in a certain area.
- Page view requests to a website
- Rainfall

Handout #3: Poisson Process

Poisson Process (with probability review)

In this note, we will consider an important random process called Poisson process. This process is a popular model for customer arrivals or calls requested to telephone systems.

We start by modeling Poisson Process as a random arrangement of "marks" (denoted by "x") on the time line. These marks may indicate the time that customers arrive or the time that call requests are made.



We will focus on one kind of Poisson process:

homogeneous Poisson process

From now on, when we say "Poisson process", what we mean is "homogeneous Poisson process".

The first property of Poisson process that you should remember is that

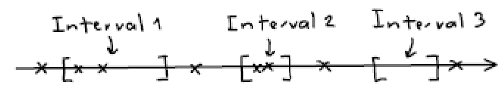
there is only <u>one</u> parameter for Poisson process. This parameter is the rate or intensity of arrivals (the average number of arrivals per unit time).

We use λ to denote this parameter.

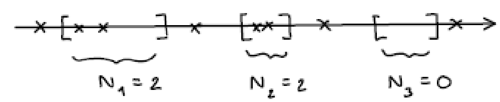
If λ is a constant, the Poisson process is homogeneous.

Our λ is constant because we focus on homogeneous Poisson process.

So, how can this λ control the Poisson Process? The key idea is that the Poisson process (PP) is as random/unstructured as a process can be. Therefore, if we consider many nonoverlapping intervals on the time-line shown below,



and count the number of arrivals in these intervals.



Then, the numbers N_1, N_2, N_3 should be independent; that is knowing the value of N_1 does not tell us anything at all about what N_2 and N_3 will be. This is what we are going to take as a vague definition of the "complete randomness" of the Poisson process.

So now, we have one more property of PP:

The number of arrivals in non-overlapping intervals are independent.

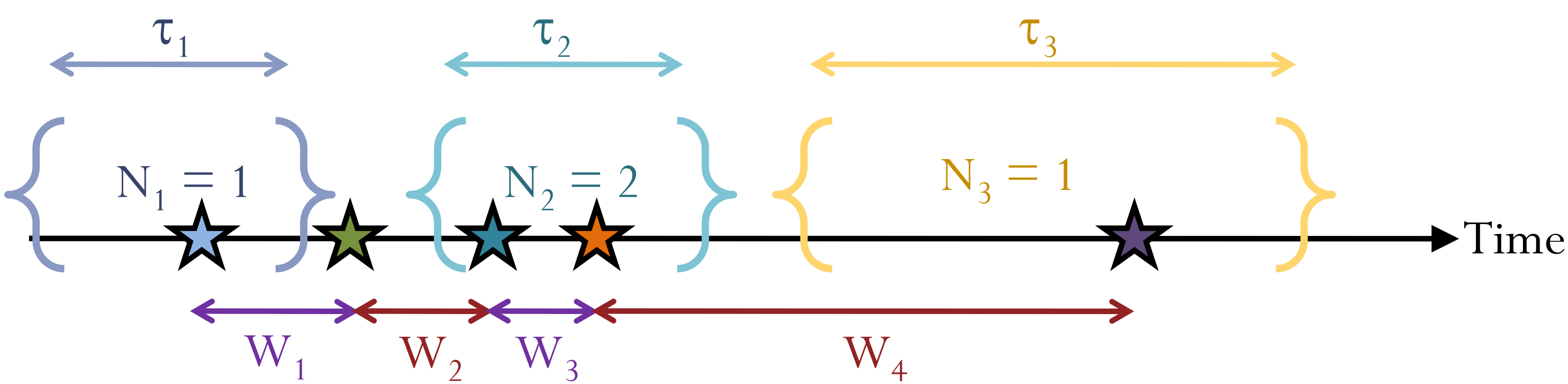
By saying something are independent, of course we mean it in terms of probability. Note that the numbers N_1, N_2, N_3 above are <u>random</u>. Because they are counting the number of arrivals, we know that they can be any nonnegative integers:

$0, 1, 2, 3, \ldots\ldots$

Because we don't know their exact values, we

Poisson Process

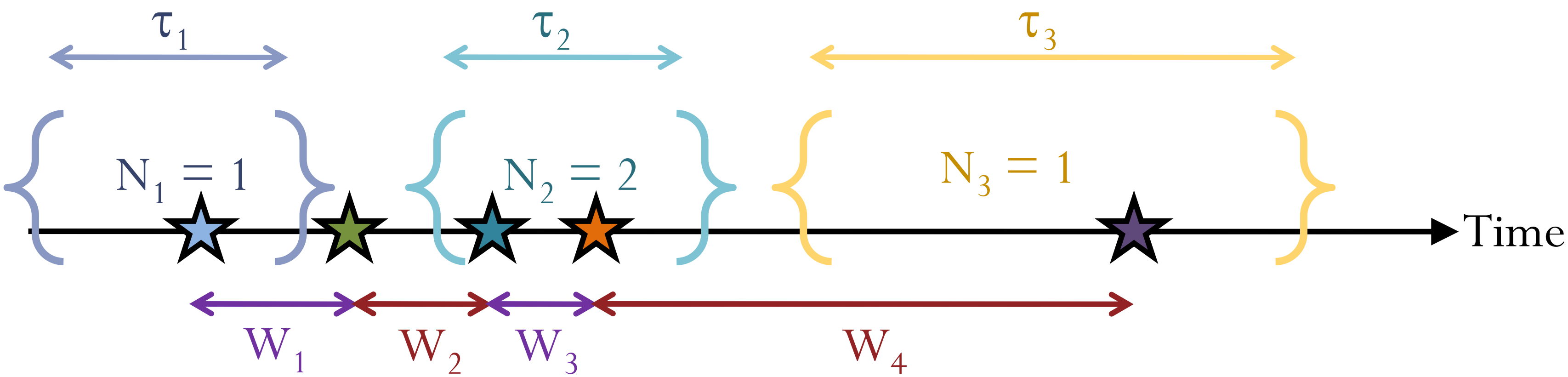
The number of arrivals N_1, N_2 and N_3 during non-overlapping time intervals are independent Poisson random variables with mean $= \lambda \times$ the length of the corresponding interval.



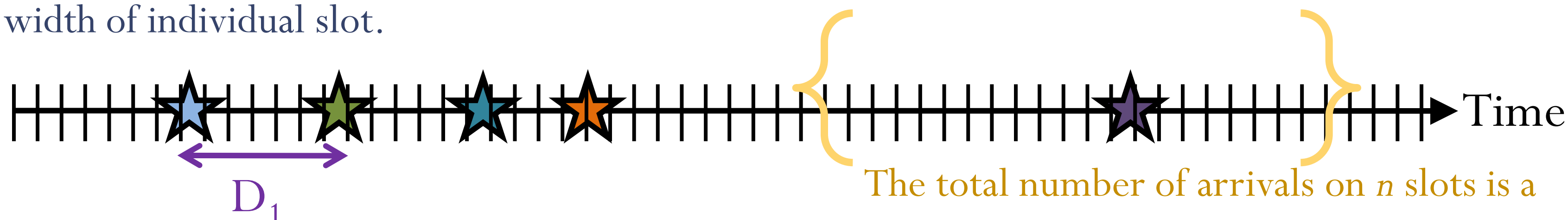
The lengths of time between adjacent arrivals $W_1, W_2, W_3 \ldots$ are i.i.d. exponential random variables with mean $1/\lambda$.

Small Slot Analysis (Poisson Process)

(discrete time approximation)



In the limit, there is at most one arrival in any slot. The numbers of arrivals on the slots are i.i.d. Bernoulli random variables with probability p_1 of exactly one arrivals $= \lambda\delta$ where δ is the width of individual slot.



The number of slots between adjacent arrivals is a geometric random variable.

The total number of arrivals on *n* slots is a binomial random variable with parameter (n,p_1)

In the limit, as the slot length gets smaller, geometric $\longrightarrow$ exponential

binomial $\longrightarrow$ Poisson

Poisson Process (Recap)

- Model call arrivals in M/M/m/m queue (which gives Erlang B formula).
- Along the way, we review many facts from probability theory.
  - pmf – Binomial, Poisson, Geometric
  - pdf - Exponential
  - Independence
  - Expectation, characteristic function
  - Sum of independent random variables and how to analyze it by characteristic functions
- You have seen that Poisson process connects many concepts that you learned from introductory probability class.

Handout #4: Erlang B & Markov Chain

Erlang B formula and its corresponding Markov Chain

Sunday, November 29, 2009
9:36 PM

Erlang B formula

In this note, we return to the Erlang B formula.

$$P_b = \frac{\frac{A^m}{m!}}{\sum_{i=0}^{m} \frac{A^i}{i!}}$$

call blocking probability

A = traffic intensity $= \frac{\lambda}{\mu}$

λ = average # of call attempts/requests per unit time

This λ is the rate for the entire trunked system. (not per user)

$\frac{1}{\mu} = H$ = average call length.

This blocking probability is a measure of the grade of service for a trunked system that provides no queuing for blocked calls.

We will see how we get this formula from a Markov chain.

We will again consider small time intervals.

Recall that one of the assumption we made to get the Erlang B formula is that traffic requests are described by a Poisson process (PP) which implies

(1) exponentially distributed call interarrival time and

(2) independence among interarrival times of call request.

There are facts that we proved when we talked about PP.

By saying that the arrival process is Poisson, I don't need to talk about these properties; they automatically follow.

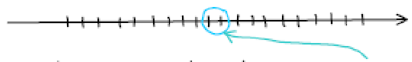
Now, in this system, there are M channels available in the trunking pool. Therefore, the probability that a call requested by a user will be blocked is given by

$$P_b = P[\text{None of the } m \text{ channels are free}]$$

this is my notation for "the probability that"

We will consider the long-term behavior of this system, i.e. the system is assumed to have been operating for a long time. In which case, at the instant that somebody is trying to make a call, we don't know how many of the channels are currently free.

To analyze this system, let's first divide the time into small slots all of which occupy the same length (as we have alway done.)



Then, consider a particular slot, say, this one. Suppose that at the begining of this slot, there are K channels that is currently used. We want to find out how this number K changes as we move forward one slot time.

This random variable K will be called the "state" of the system.

This is the same "state" concept that you have studied in digital circuits class.

There are many possible values for the "state" and the system moves from one state to another one as we move forward in time by one slot.

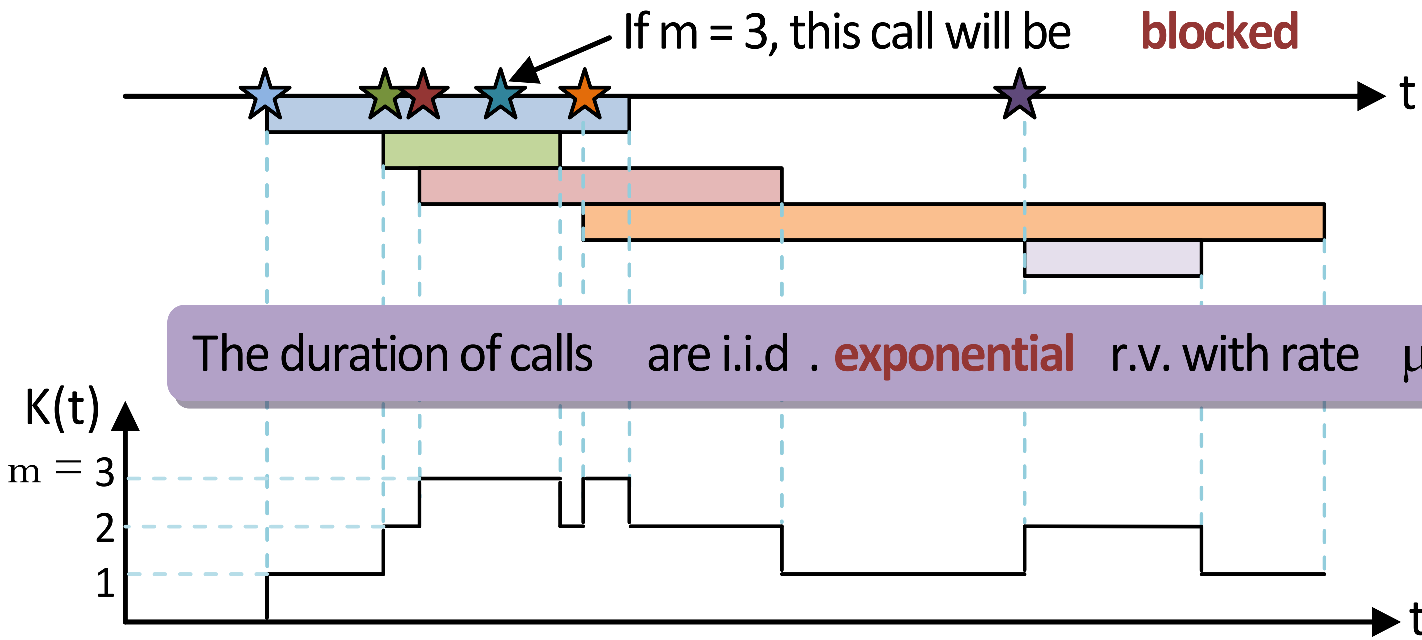
For example, suppose there are 5 persons using the channels at the beginning of the slot. Then K=5. Suppose by the end of the slot, none of these 5 persons finish their calls. Suppose also that there is one more person want to make a call at some moment of time during this slot. Then, at the end of the slot, the number of channels that are used becomes

$$5 - 0 + 1 = 6.$$

So, the state K of the system changes from 5 to 6 when we reach the end of slot which is

M/M/m/m Assumption (Recap)

The call request process is **Poisson** with rate λ



The duration of calls are i.i.d. **exponential** r.v. with rate μ.

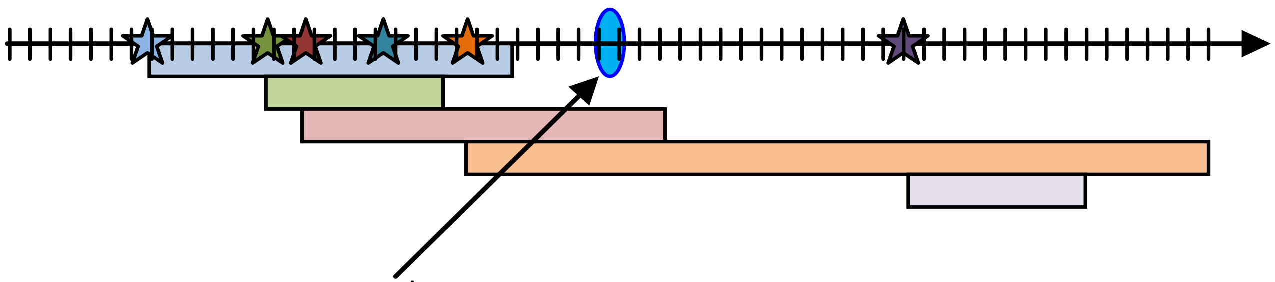
$K(t)$ = "state" of the system

= the number of used channel at time t

We want to find out what proportion of time the system has $K = m$.

Small Slot Analysis (Erlang B)

Suppose each slot duration is δ.



- Consider the i^{th} small slot.
- Let $K_i = k$ be the value of K at the beginning of this time slot.
- $k = 2$ in the above figure.
- Then, K_{i+1} is the value of K at the end of this slot which is the same as the value of K at the beginning of the next slot.
- P[0 new call request] ≈ 1 - $\lambda\delta$
- P[1 new call request] ≈ $\lambda\delta$
- P[0 old-call end] ≈ $(1-\mu\delta)^k \approx 1-k\mu\delta$
- P[1 old-call end] ≈ $k\mu\delta(1-\mu\delta)^{k-1} \approx k\mu\delta$

How do these events affect K_{i+1}?