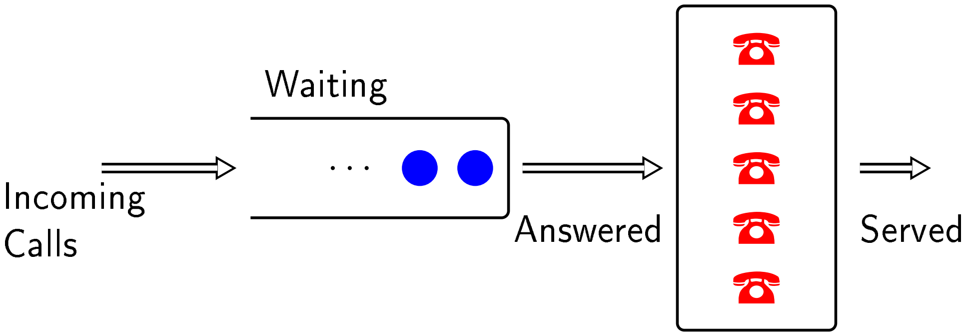
Multi-Server Systems

Waiting Systems – M/M/C/∞

The Phone Call Example – Waiting System



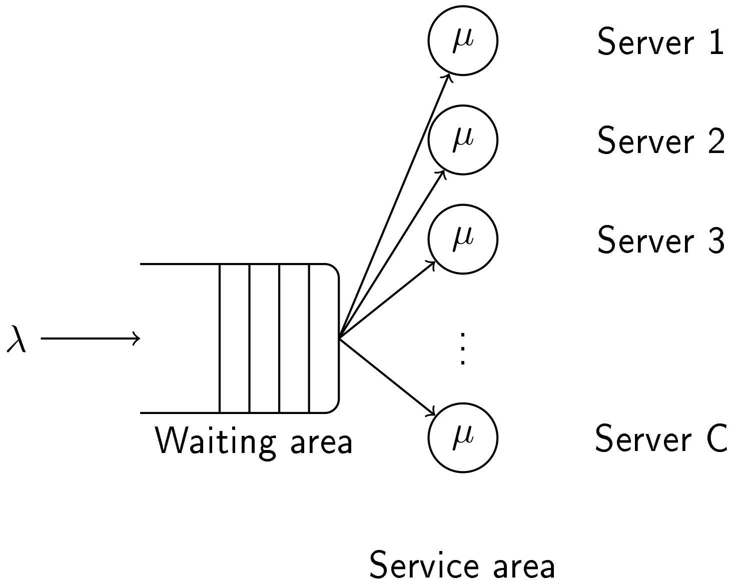
- Calls that find no available agent wait
- Infinite waiting area
- No retries

We will now study a pure wait system

- Let's consider a call center, as in our previous illustration, red agents mean busy agent, and green ones mean available ones
- Calls that arrive and find an available agent, are served. And calls that arrive and find no available agent, are put on hold
- We assume that there is an infinite waiting capacity, and a finite number of agents. We also assume that there are no retries

Formalization: M/M/C/∞

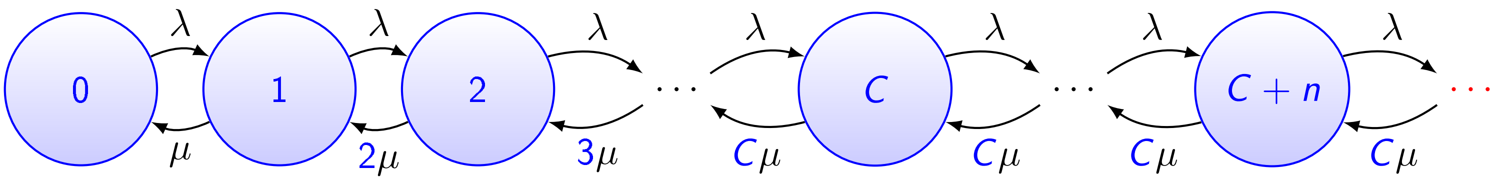
- Calls arrive according to a Poisson process of intensity λ
- There are C agents or servers
- Call durations are exponentially distributed with mean $1/\mu$



- To formalize our example, we need to make some assumptions. We will assume that call arrivals occur according to a Poisson process of an average of λ calls per unit of time. There are C servers which correspond to the agents in our call center example. Call durations are exponentially distributed, and on average calls last $1/\mu$ units of time. If arriving calls find all agents busy, they are put on hold, the waiting area has infinite capacity
- With these assumptions we can model the system as an M/M/C queue

M/M/C/∞ – State Transition Diagram

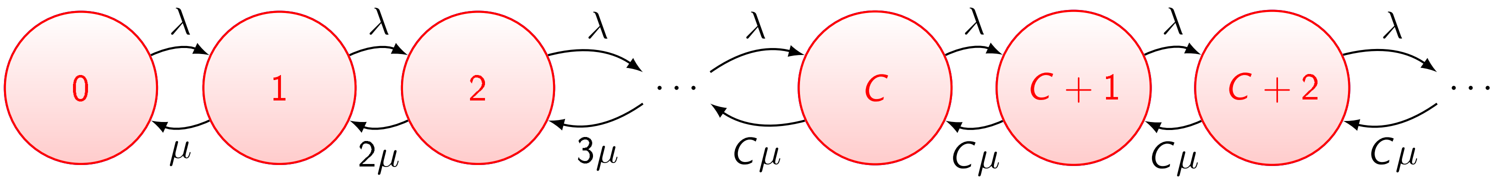
- We define the state as the number of customers in the system $\Rightarrow$ infinite states



- Customers arrive with a rate λ
- Servers are freed with a rate proportional to the number of customers being served
- Stability if and only if $\rho < C$ - from now on we shall assume this holds

In order to study the system, we proceed as we did before, first drawing the state transition diagram, and later on we'll compute the steady-state distribution. We define the state as the number of customers in the system. So we have an infinite number of states. Customers arrive with a rate lambda, which is the transition rate from state i to state $i+1$. Servers are freed with a rate proportional to the number of customers being served, like with the M/M/C/C queue, which is a maximum of C customers being served at a time. In this case, we have infinite states, and the queue will be stable if and only if $\rho < C$, which we shall assume holds from now on.

M/M/C/∞ – Steady-State Distribution



1. Local Balance Equations
   $\lambda\pi_0 = \mu\pi_1 \lambda\pi_1 = 2\mu\pi_2 \lambda\pi_{j-1} = j\mu\pi_j$, $j = 1, \dots, C$
   $\lambda\pi_C = C\mu\pi_{C+1} \lambda\pi_{C+1} = C\mu\pi_{C+2} \lambda\pi_{j-1} = C\mu\pi_j$, $\forall j > C$
2. Recursion $\Rightarrow$
   $\pi_j = \frac{\rho^j}{j!}\pi_0$, $j = 0, 1, \dots, C$
   and
   $\pi_j = \pi_0 \frac{\rho^C}{C!}(\frac{\rho}{C})^{j-C}$, $\forall j > C$
3. Normalization $\Rightarrow$
   $$\pi_0 = \frac{1}{\sum_{j=0}^{C} \frac{\rho^j}{j!} + \frac{\rho^C}{C!}\frac{\rho}{C-\rho}}$$

We are now ready to compute the steady state distribution.

- First we state the local balance equations. We have, for instance, that the mean flow from state 0 to one is equal to the mean flow from state one to zero. That is, $\pi_0\lambda$ is equal to $\pi_1\mu$. Similarly, if we consider states 1 and 2, $\lambda\pi_1$ is equal to two times $\mu\pi_2$. In general, we have, for j smaller than C, that $\lambda\pi_j$ is equal to j times $\mu\lambda\pi_{j+1}$.
- For states greater than or equal to C, we have for instance, $\pi C\lambda = C\mu\pi_{C+1}$. And similarly if we consider states $C+1$ and $C+2$. So, in general we have, for j greater than C, that $\lambda\pi_{j-1}$ is equal to $C\mu\pi_j$.
- At a second step, we observe that by recursion we can express each π_j as a function of π_0.
- And finally, we apply the Normalization condition, that is the fact that the sum of all probabilities must be equal to one, and so we obtain the value of π_0
- Note that this sum is an infinite one, in order for it to converge we need ρ/C to be strictly smaller than 1, which is the stability condition that we stated in the previous slide.