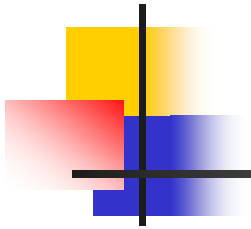
CS 5229: Advanced Compute Networks

Basic Queuing Model

Dr. Chan Mun Choon

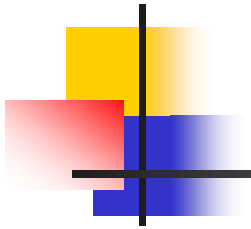
School of Computing, National University of Singapore

Aug 19, 2010



Reference

- Bertsekas and Gallager, “Data Networks”, 2nd Edition, Chapter 3: Delay Models in Data Network, Prentice Hall.



Motivation

- Analyzing network performance is difficult even for a single networking node
- However, if we restrict ourselves to certain set of traffic models, one can obtain valuable qualitative results and worthwhile intuition
  - For example, traffic engineering in the telephone network has been effective
  - The M/M/* queuing analysis is a simple and elegant way to perform basic traffic engineering

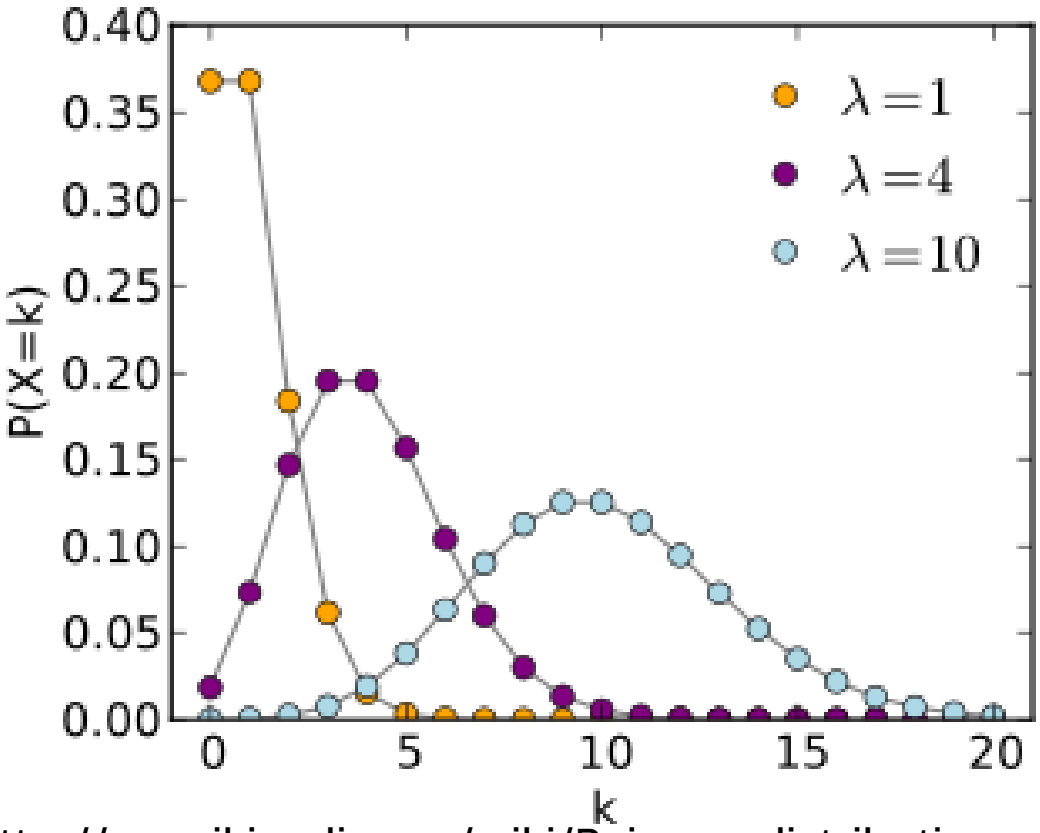
What is a Poisson Process?

- A Poisson Process A(t)
  1. A(t) is a **counting process** that represents the total number of arrivals that have occurred from 0 to t, A(t) – A(s) equals the number of arrivals in the interval (s,t]
  2. Number of arrivals that occur in disjoint intervals are **independent**
  3. Number of arrivals in any interval τ is Poisson distributed with parameter $\lambda\tau$

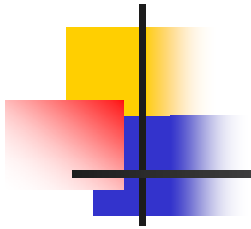
$$P\{A(t+\tau) - A(t) = n\} = e^{-\lambda\tau}\frac{(\lambda\tau)^n}{n!}$$

Cont'd

- Mean = Variance = λ



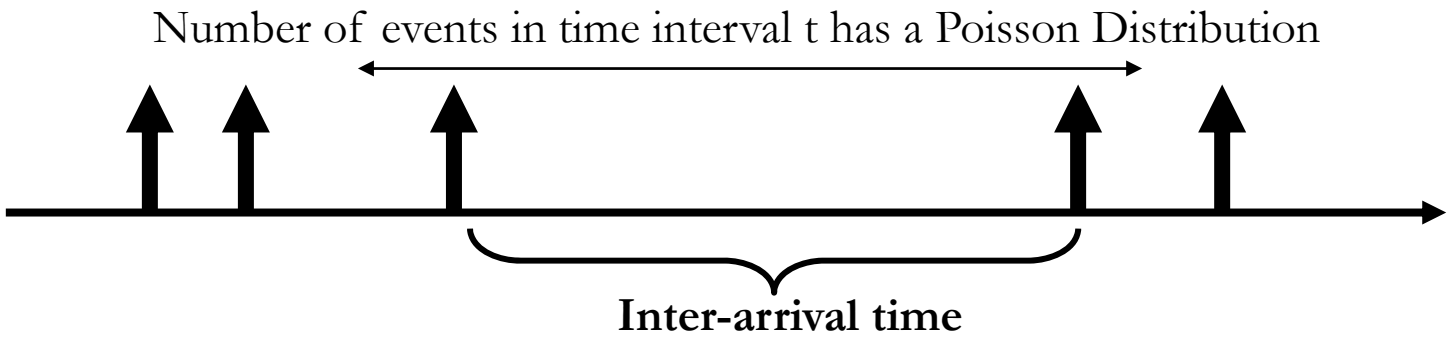
From: http://en.wikipedia.org/wiki/Poisson_distribution



Poisson Process

- **Merging:** if two or more independent Poisson process are merged into a single process, the merged process is a Poisson process with a rate equal to the sum of the rates
- **Splitting**: if a Poisson process is split probabilistically into two processes, the two processes are obtained are also Poisson

Inter-arrival Time



- Based on the definition of Poisson process, what is the inter-arrival time between arrivals?
- The distribution of inter-arrival time, t, can be computed as P{A(t) = 0}. Let T_n be the arrival time of the n^{th} event.

$$P\{A(t+\tau)-A(t)=0\}=P\{A(\tau)=0\}=e^{-\lambda\tau}$$

$$P\{T_{n+1}-T_n\le\tau\,|\,T_o,\ldots,T_n\}=1-e^{-\lambda\tau}$$

Exponential Distribution

- Cumulative Density Distribution

$$P\{\tau \le s\} = 1 - e^{-\lambda s}$$

- Probability Density Distribution

$$p\{\tau\} = \lambda e^{-\lambda \tau}$$

- Mean

$$E\{\tau\} = \frac{1}{\lambda}$$

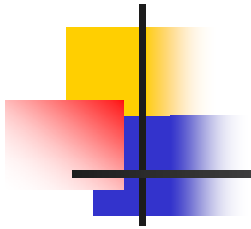
- Variance

$$Var\{\tau\} = \frac{1}{\lambda^2}$$

Memoryless Property

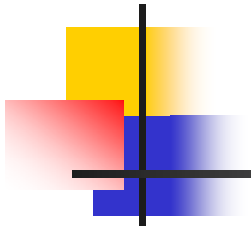
- For service time with exponential distribution, the additional time needed to complete a customer's service in progress is independent of when the service started

$$P\{\tau_n > r + t \mid \tau_n > t\} = \frac{e^{-\lambda(r+t)}}{e^{-\lambda t}} = e^{-\lambda r} = P\{\tau_n > r\}$$



Question

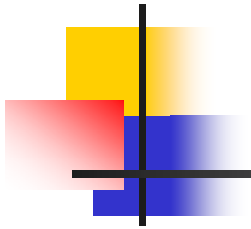
- Inter-arrival time of bus arriving at a bus stop has an exponential distribution.
  - A random observer arrives at the bus stop and a bus just leave t seconds ago. How long should the observer expects to wait?



Applications of Poisson Process

- Poisson Process has a number of “nice” properties that make it very useful for analytical and probabilistic analysis
- Has been used to model a large number of physical occurrences [KLE75]
  - Number of soldiers killed by their horse (1928)
  - Sequence of gamma rays emitting from a radioactive particle
  - Call holding time of telephone calls
  - **In many cases, the sum of large number of independent stationary renewal process will tend to be a Poisson process**

[KLE75] L. Kleinrock, “Queuing Systems,” Vol I, 1975.



Little's Theorem

- Given customer arrival rate (λ), service rate (μ)
  - What is the average number of customers (N) in the system and what is the average delay per customer (T) ?

Cont'd

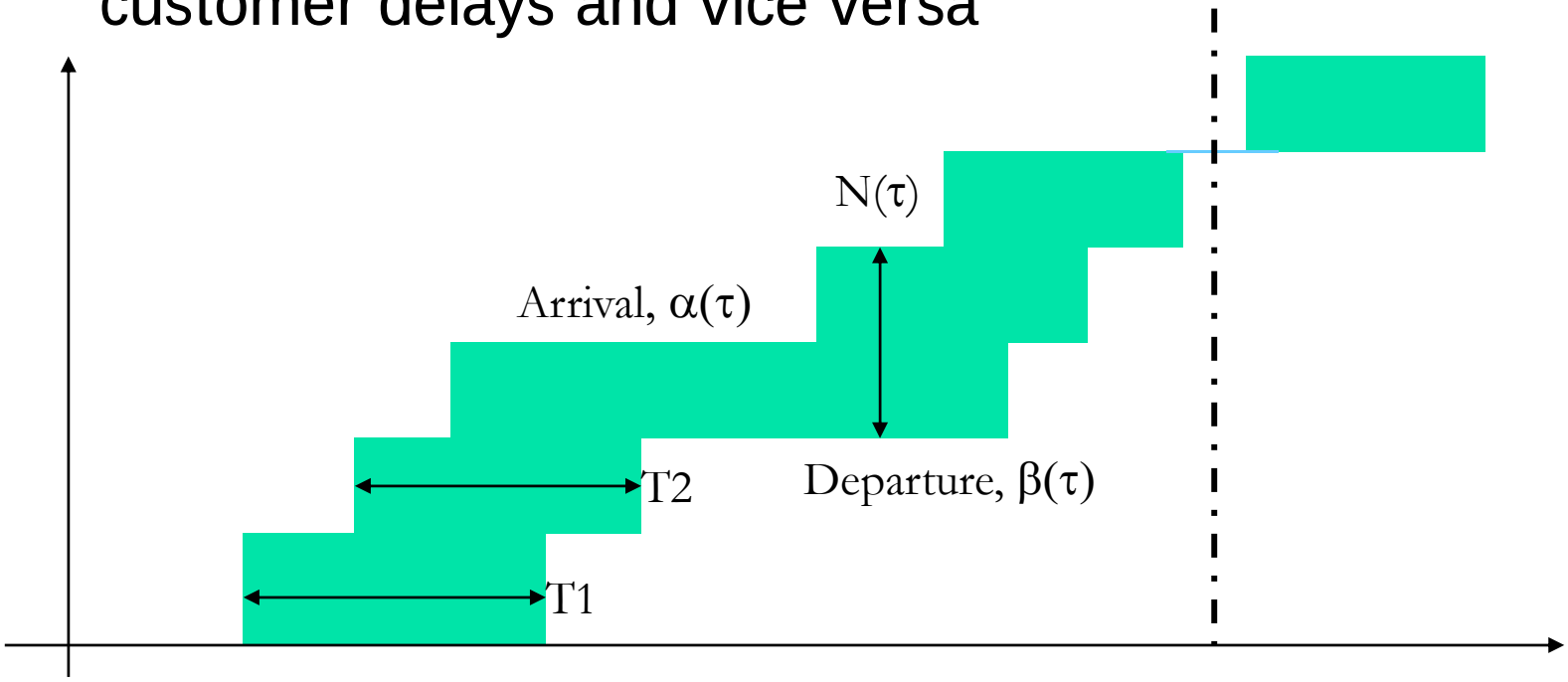
- Let
  - N(t) = # of customers at time t
  - α(t) = # of customers arrived in the interval [0,t]
  - T_i = time spent in system by i^{th} customer
- N_t : "typical" # of customers up to time t is

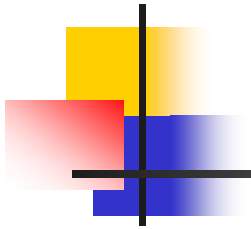
$$\frac{1}{t}\int_0^t N(\tau)d\tau$$

$$N = \lim_{t->\infty} N_t \qquad \lambda = \lim_{t->\infty} \lambda_t \qquad T = \lim_{t->\infty} T_t$$

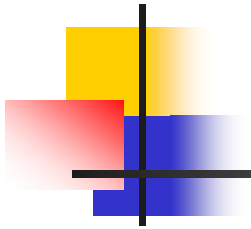
Little's Theorem

- Little's Theorem: $N = \lambda T$
  - Average # of customers = average arrival rate * average delay time of a customer
  - Crowded system (large N) are associated with long customer delays and vice versa



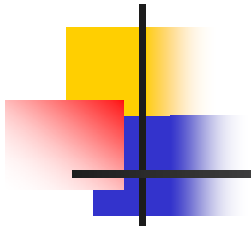


Derivation of Little's Theorem



Little's Theorem (cont'd)

- Little's Theorem is very general and holds for almost every queuing system that reaches statistics equilibrium in the limit

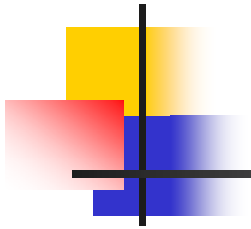


Example

- BG, Example 3.1
  - L is the arrival rate in a transmission line
  - N_Q is the average # of packets in queue (not under transmission)
  - W is the average time spent by a waiting packet (exclude packet being transmitted)
  - From LT, $N_Q = \lambda W$
  - Furthermore, if X is the average transmission time,
    - $\rho = \lambda X$
    - where ρ is the line's utilization factor (proportion of time line is busy)

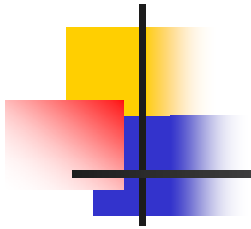
Example

- BG, Example 3.2
  - A network of transmission lines where packets arrived at n different nodes with rate λ_1 λ_2 ,...,λ_n
  - N is total number of packets in network
  - Average delay per packet is $$T = \frac{N}{\sum_{i=1}^{n} \lambda_i}$$
  - independent of packet length distribution (service rate) and routing



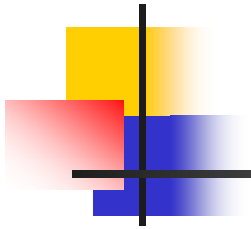
A Question ...

- Waiting time at two fast-food stores MD and BK
  - In MD, a queue is formed at each of the m servers (assume a customer chooses queue independently and does not change queue once he/she joins the queue)
  - In BK, all customers wait at a single queue and served by m servers
  - Which one is better?



Multiplexing of Traffic

- Traffic engineering involves the sharing of resource/link by several traffic streams
- Time-Division Multiplexing (TDM)
  - Divide transmission into time slots
- Frequency Division Multiplexing (FDM)
  - Divide transmission into divide frequency channels
- For TDM/FDM, if there is no traffic in a data stream, bandwidth is wasted



Statistical Multiplexing

- In statistical multiplexing, data from all traffic streams are merged into a single queue and transmitted in a FIFO manner
- One big advantage moving from circuit switching to packet switching is that statistical multiplexing can be exploited
- Benefits statistical multiplexing
  - has smaller delay per packet than TDM/FDM
  - can have larger delay variance
  - Results can be shown using queuing analysis

Basic Queuing Model

Departure Process
Exponential with mean 1/μ

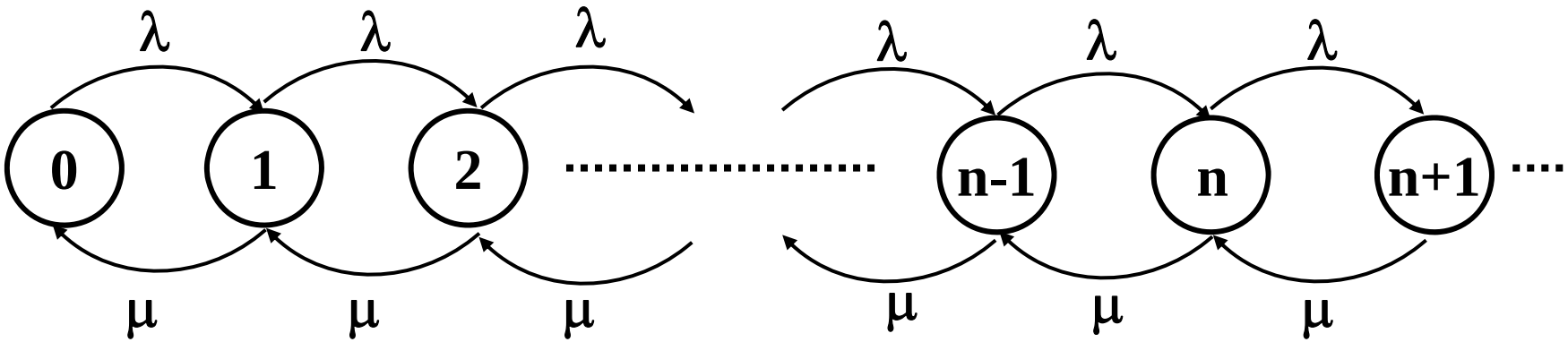
M/M/1

Number of servers

Arrival Process
Memoryless (or Poisson process with rate λ)

- **Default N is infinite**
- **D - deterministic, G - General**

Birth-Death Process



- **Model queue as a discrete time Markov chain**
- **Let P_n be the steady state probability that there are n customers in the queue**
- **Balance equation: at equilibrium, the probability a transition out of a state is equal to the probability of a transition into the same state**

Derivation of M/M/1 Model

- **Balance Equations:**
  - $\lambda P_0 = \mu P_1, \lambda P_1 = \mu P_2, \ldots, \lambda P_{n-1} = \mu P_n$
- **Let** $\rho = \lambda/\mu$
- $\rho P_0 = P_1, \rho P_1 = P_2, \ldots, \rho P_{n-1} = P_n$

$$P_n = \rho^n P_0$$

Derivation of M/M/1 Model

$$P_n = \rho^n P_0$$

$$\Sigma_n P_n = \Sigma_n \rho^n P_0 = P_0 / (1 - \rho) = 1 \ (\rho < 1)$$

$$P_0 = (1 - \rho)$$

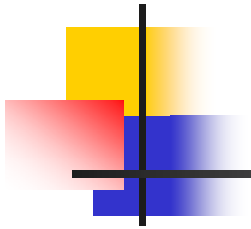
$$P_n = \rho^n (1 - \rho)$$

Average Number of Customers in System, N

$$N = \Sigma_n nP_n = \rho / (1 - \rho) = \lambda / (\mu - \lambda)$$

Properties of M/M/1 Queue

- **N =** $\rho / (1 - \rho) = \lambda / (\mu - \lambda)$
  - ρ can be interpreted as the utilization of the queue
  - System is unstable if $\rho > 1$ or $\lambda > \mu$ as N is not bounded
- In M/M/1 queue, there is no blocking/dropping, so waiting time can increase without any limit
  - Buffer space is infinite, so customers are not rejected
  - But there are “infinite number” of customers in front



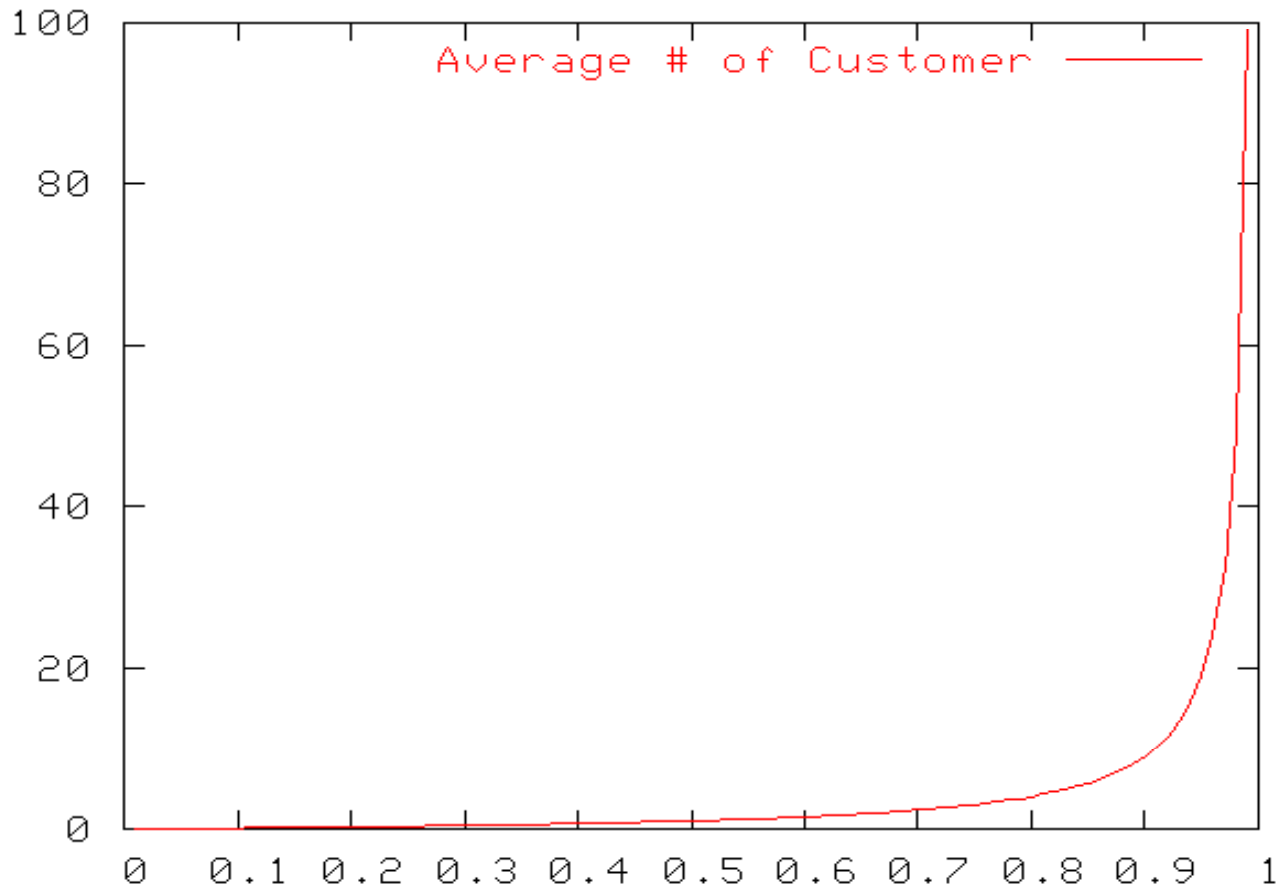
M/M/1

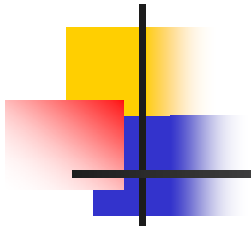
- From Little's Theorem,

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

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

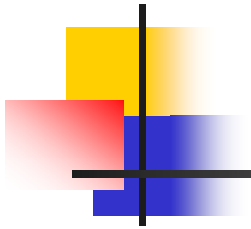
More properties of M/M/1





Example

- BG, Example 3.8 (Statistical Multiplexing vs. TDM)
  - Allocate each Poisson stream its own queue (λ,μ) or shared a single faster queue (kλ,kμ)?
  - Increase λ and μ or a queue by a constant k > 1
  - ρ = kλ/kμ = λ/μ (no change in utilization)
  - N = ρ / 1–ρ = λ / μ – λ (no change)
  - What changes?
    - T = 1/k(μ – λ)
    - Average transmission delay decreases by a factor k
  - **Why?**

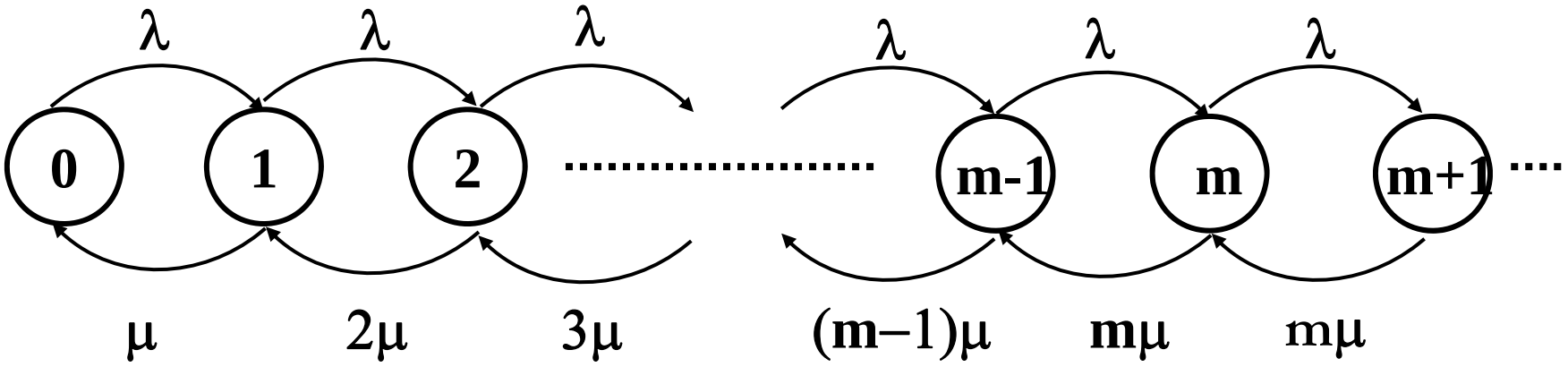


Example

- BG, Example 3.9
  - Consider k TDM/FDM channels
  - From previous example, merging k channels into a single (k times faster) will keep the same N but reduces average delay by k
  - So why use TDM/FDM ?
    - Some traffic are not Poisson. For example, voice traffic are “regular” with one voice packet every 20ms
    - Merging multiplexing traffic streams into a single channel incurs buffering, “queuing delay” and jitter

Extension to M/M/m Queue

- **There are m servers, a customer is served by one of the servers**
- $\lambda p_{n-1} = n\mu p_n$ **(n <= m)**
- $\lambda p_{n-1} = m\mu p_n$ **(n > m)**



Derivation of M/M/m Model

- **Balance Equations:**
  - $\lambda P_0 = \mu P_1, \lambda P_1 = 2\mu P_2, \ldots, \lambda P_{n-1} = n\mu P_n$
- **Let $\rho = \lambda/m\mu$**

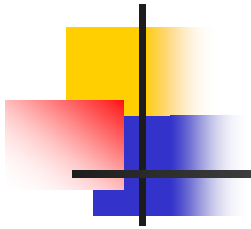
$$p_n = p_0 \frac{(m\rho)^n}{n!}, n \le m$$

$$p_n = p_0 \frac{m^m \rho^n}{m!}, n > m$$

Derivation of M/M/m Model

$$\sum_{n=0}^{\infty} p_n = 1$$

In order to compute P_n, $\mathbf{P_0}$ must be computed first.

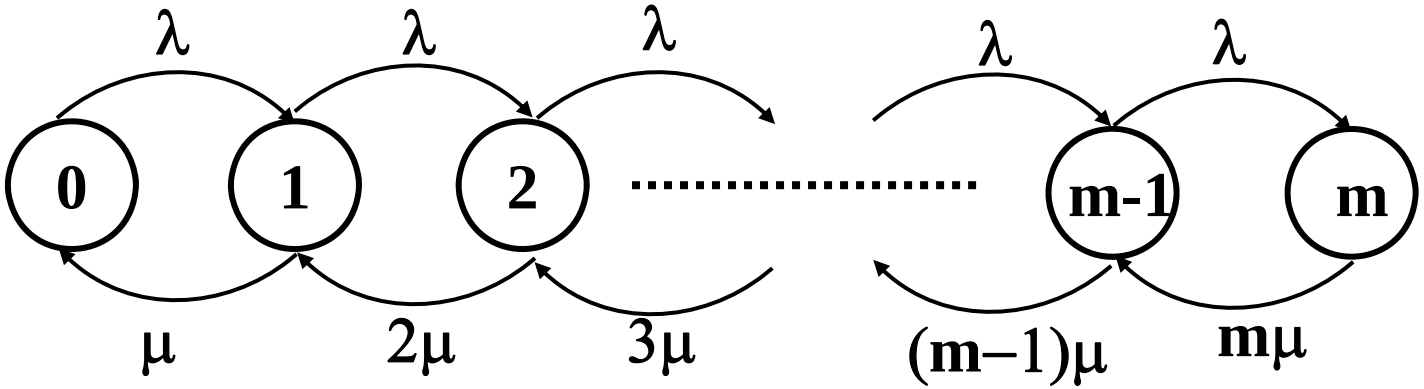


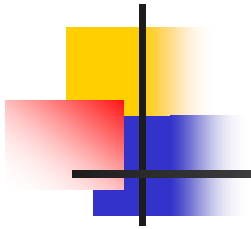
BK vs. MD

- BK – M/M/m
- MD – m * M/M/1
- Let m = 5,
  - λ of BK is 3, μ be 1
  - λ of each server in MD is 3/5 = 0.6
- What is the expected delay?

Extension to M/M/m/m Queue

- **There are m servers and m buffer size**
- **This is no buffering**
- **Calls are either served or rejected, calls rejected are lost**
- **Common model for telephone switching**





M/M/m/m Queue

Balanced Equations:

$$\lambda P_0 = \mu P_1,\ \lambda P_1 = 2\mu P_2,\ \ldots,\ \lambda P_{n-1} = n\mu P_n$$

$$P_n = P_0 (\rho^n) / n!$$

$$\Sigma^m_n P_n = \Sigma^m_n P_0 (\rho^n) / n! = 1$$

$$P_0 = (\Sigma^m_n (\rho^n) / n!)^{-1}$$

When does loss happens?

Loss happens when a customer arrives and see m customers in the system

M/M/m/m Queue

- PASTA: Poisson Arrival see times averages
  - P_m is time average
  - Use time averages to compute loss rate
- Loss for M/M/m/m queue is computed as the probability that there are m customers in the system:

$$(\rho^m / m!) \left(\Sigma^m_{n=0} (\rho^n / n!) \right)^{-1}$$

- The above equation is known as Erlang B formula and widely used to evaluate blocking probability

What is an Erlang?

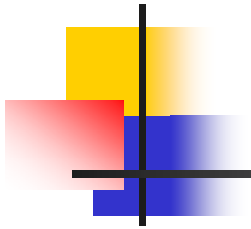
- An *Erlang* is a unit of telecommunications traffic measurement and represents the continuous use of one voice path
  - Average number of calls in progress
- Computing Erlang
  - Call arrival rate:λ
  - Call Holding time is: $1/\mu$, call departure rate = μ
  - System load in Erlang is λ/μ
- Example:
  - λ = 1 calls/sec, $1/\mu$ = 100sec, load = 1/0.01 = 100 Erlangs
  - λ = 10 calls/sec, $1/\mu$ = 10sec, load = 10/0.1 = 100 Erlangs
- Load is function of the ratio of arrival rate to departure rate, independent of the specific rates

Erlang B Table

Capacity (Erlangs) for grade of service of				
# of Servers (N)	P=0.02	P=0.01	P=0.005	P=0.001
1	0.02	0.01	0.005	0.001
5	1.66	1.36	1.13	0.76
10	5.08	4.46	3.96	3.09
20	13.19	12.03	11.1	9.41
40	31.0	29.0	27.3	24.5
100	87.97	84.1	80.9	75.2

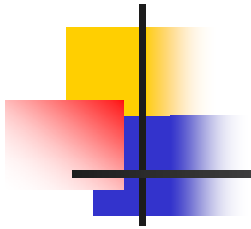
- **For a given grade of service, a larger capacity system is more efficient (statistical multiplexing)**

- **A larger system incurs a larger changes in blocking probability when the system load changes**



Example

- If there are 40 servers and target blocking rate is 2%, what is largest load supported?
  - P=0.02, N = 40
  - Load supported = 31 Erlang



Example

- Calls arrived at a rate of 1calls/sec and the average holding time is 12 sec. How many trunk is needed to maintain call blocking of less than 1%?
  - Load = 1*12 = 12 Erlang
  - From Erlang B table, if P=0.01, N >= 20