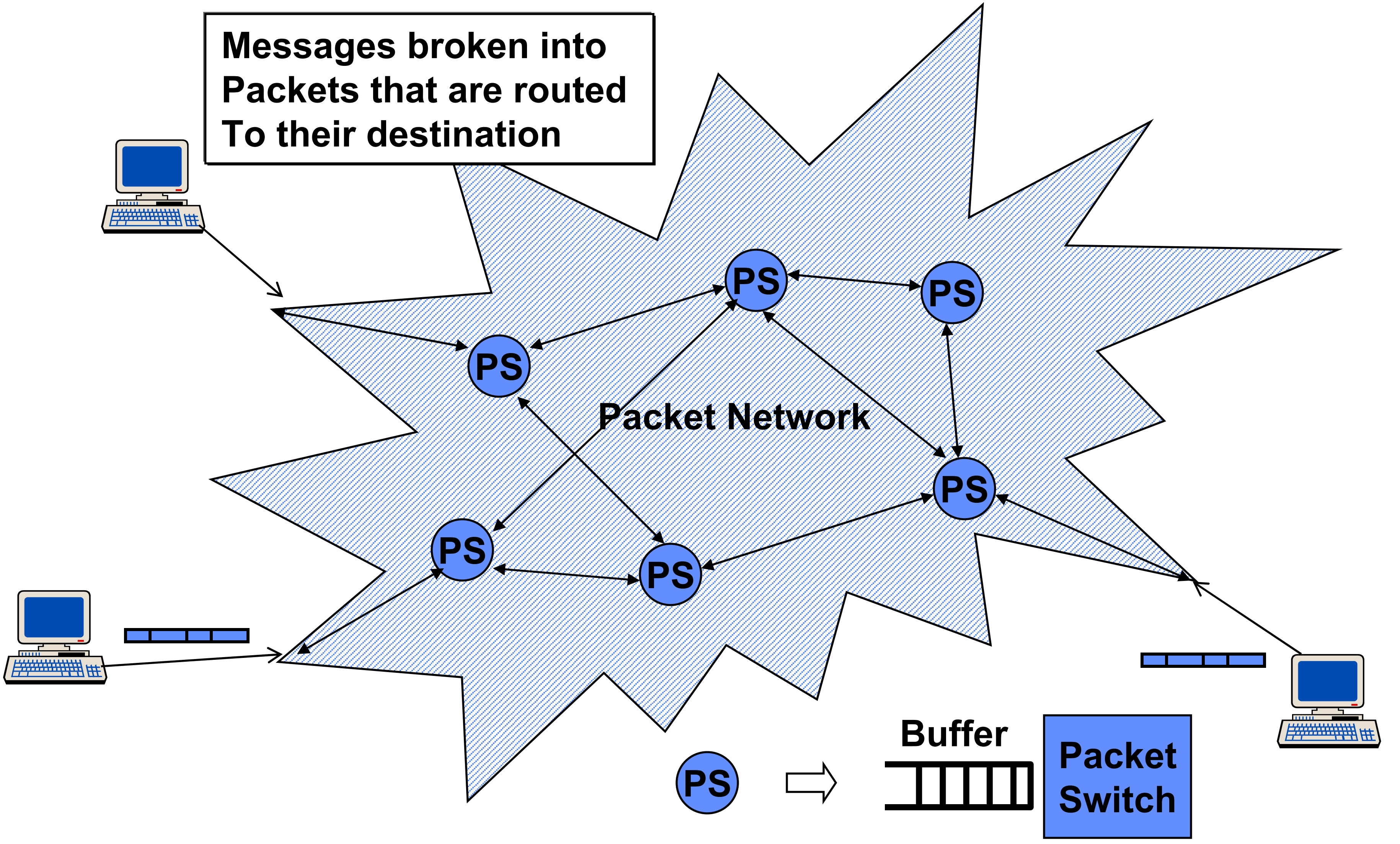
Lectures 5 & 6

6.263/16.37

Introduction to Queueing Theory

Eytan Modiano
MIT, LIDS

Packet Switched Networks



Queueing Systems

- **Used for analyzing network performance**
- **In packet networks, events are random**
  - **Random packet arrivals**
  - **Random packet lengths**
- **While at the physical layer we were concerned with bit-error-rate, at the network layer we care about delays**
  - **How long does a packet spend waiting in buffers ?**
  - **How large are the buffers ?**
- **In circuit switched networks want to know call blocking probability**
  - **How many circuits do we need to limit the blocking probability?**

Random events

- **Arrival process**
  - **Packets arrive according to a random process**
  - **Typically the arrival process is modeled as Poisson**

- **The Poisson process**
  - **Arrival rate of λ packets per second**
  - **Over a small interval δ,**

    P(exactly one arrival) = $\lambda\delta + o(\delta)$
    P(0 arrivals) = 1 - $\lambda\delta + o(\delta)$
    P(more than one arrival) = $0(\delta)$

    Where $0(\delta)/\delta \to 0$ □□ $\delta \to 0$.

  - **It can be shown that:**

$$\text{P(n arrivals in interval T)} = \frac{(\lambda T)^n e^{-\lambda T}}{n!}$$

The Poisson Process

$$P(\text{n arrivals in interval T}) = \frac{(\lambda T)^n e^{-\lambda T}}{n!}$$

n = number of arrivals in T

It can be shown that,

$$E[n] = \lambda T$$

$$E[n^2] = \lambda T + (\lambda T)^2$$

$$\sigma^2 = E[(n - E[n])^2] = E[n^2] - E[n]^2 = \lambda T$$

Inter-arrival times

- Time that elapses between arrivals (IA)

$P(IA <= t) = 1 - P(IA > t)$
$= 1 - P(0 \text{ arrivals in time } t)$

$= 1 - e^{-\lambda t}$

- This is known as the exponential distribution
  - Inter-arrival CDF = $F_{IA}(t) = 1 - e^{-\lambda t}$
  - Inter-arrival PDF = $d/dt\, F_{IA}(t) = \lambda e^{-\lambda t}$
- The exponential distribution is often used to model the service times (I.e., the packet length distribution)

Markov property (Memoryless)

$$P(T \le t_0 + t \mid T > t_0) = P(T \le t)$$

$\Pr oof:$

$$P(T \le t_0 + t \mid T > t_0) = \frac{P(t_0 < T \le t_0 + t)}{P(T > t_0)}$$

$$= \frac{\int_{t_0}^{t_0+t} \lambda e^{-\lambda t} dt}{\int_{t_0}^{\infty} \lambda e^{-\lambda t} dt} = \frac{-e^{-\lambda t} \Big|_{t_0}^{t_0+t}}{-e^{-\lambda t} \Big|_{t_0}^{\infty}} = \frac{-e^{-\lambda(t+t_0)} + e^{-\lambda(t_0)}}{e^{-\lambda(t_0)}}$$

$$= 1 - e^{-\lambda t} = P(T \le t)$$

- **Previous history does not help in predicting the future!**
- **Distribution of the time until the next arrival is independent of when the last arrival occurred!**

Example

- Suppose a train arrives at a station according to a Poisson process with average inter-arrival time of 20 minutes

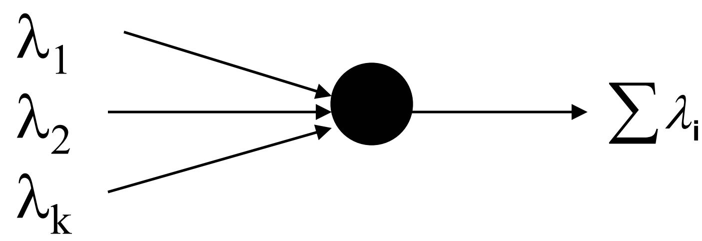
- When a customer arrives at the station the average amount of time until the next arrival is 20 minutes
  - Regardless of when the previous train arrived

- The average amount of time since the last departure is 20 minutes!

- Paradox: If an average of 20 minutes passed since the last train arrived and an average of 20 minutes until the next train, then an average of 40 minutes will elapse between trains
  - But we assumed an average inter-arrival time of 20 minutes!
  - What happened?

Properties of the Poisson process

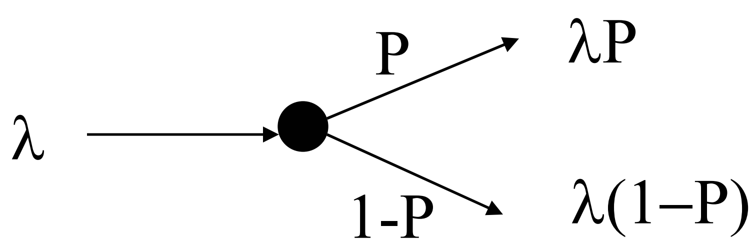
- **Merging Property**



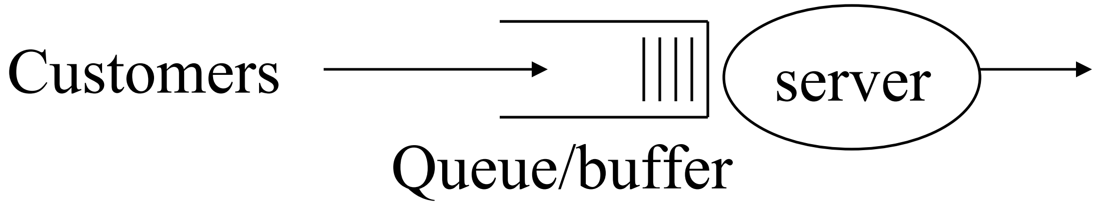
Let A1, A2, … Ak be independent Poisson Processes of rate $\lambda 1$, $\lambda 2$, …λk

$$A = \sum A_i \text{ is also Poisson of rate } = \sum \lambda_i$$

- **Splitting property**
  - **Suppose that every arrival is randomly routed with probability P to stream 1 and (1-P) to stream 2**
  - **Streams 1 and 2 are Poisson of rates $P\lambda$ and $(1-P)\lambda$ respectively**



Queueing Models

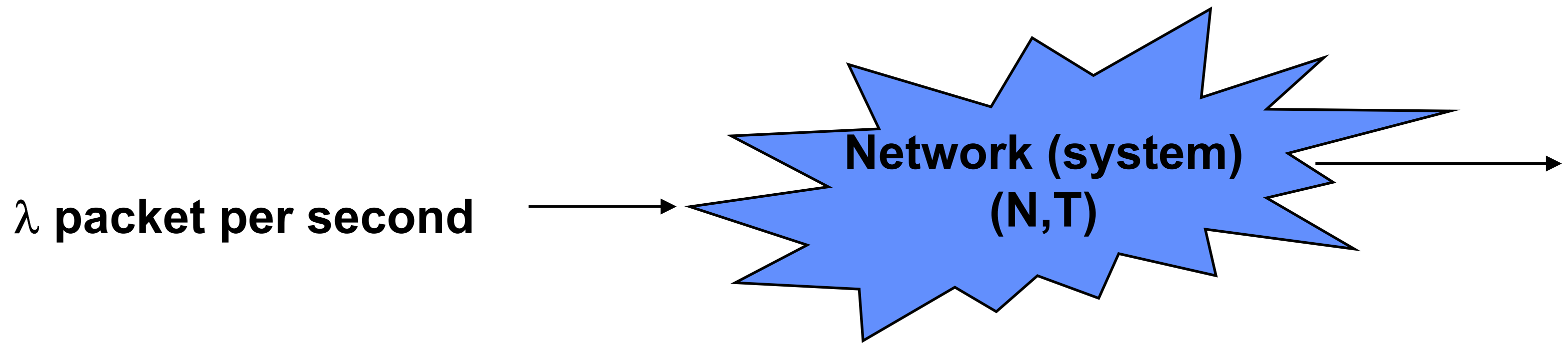


- **Model for**
  - Customers waiting in line
  - Assembly line
  - Packets in a network (transmission line)

- **Want to know**
  - Average number of customers in the system
  - Average delay experienced by a customer

- **Quantities obtained in terms of**
  - Arrival rate of customers (average number of customers per unit time)
  - Service rate (average number of customers that the server can serve per unit time)

Little's theorem

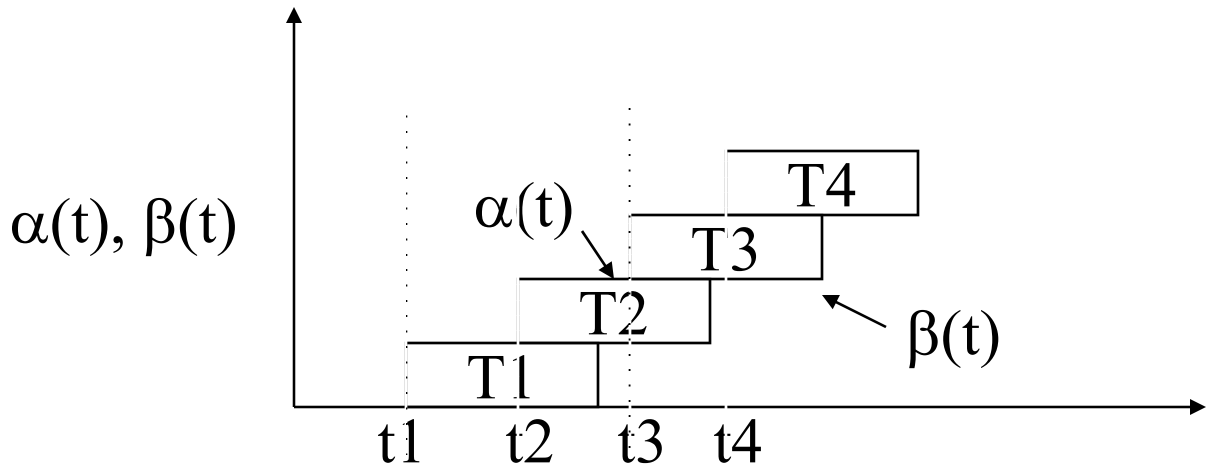


- N = average number of packets in system
- T = average amount of time a packet spends in the system
- λ = arrival rate of packets into the system (not necessarily Poisson)

- Little's theorem: $N = \lambda T$
  - Can be applied to entire system or any part of it
  - Crowded system -> long delays

    On a rainy day people drive slowly and roads are more congested!

Proof of Little's Theorem



- $\alpha(t)$ = number of arrivals by time t
- $\beta(t)$ = number of departures by time t
- t_i = arrival time of i^{th} customer
- T_i = amount of time i^{th} customer spends in the system
- $N(t)$ = number of customers in system at time t = $\alpha(t) - \beta(t)$

- Similar proof for non First-come-first-serve

Proof of Little's Theorem

$$N_t = \frac{1}{t}\int_0^t N(\tau)d\tau = \text{time ave. number of customers in queue}$$

$$N = Limit_{t\to\infty} N_t = \text{steady state time ave.}$$

$$\lambda_t = \alpha(t)/t,\ \lambda = Limit_{t\to\infty}\lambda_t = \text{arrival rate}$$

$$T_t = \frac{\sum_{i=0}^{\alpha(t)} T_i}{\alpha(t)} = \text{time ave. system delay},\ T = Limit_{t\to\infty} T_t$$

- **Assume above limits exists, assume Ergodic system**

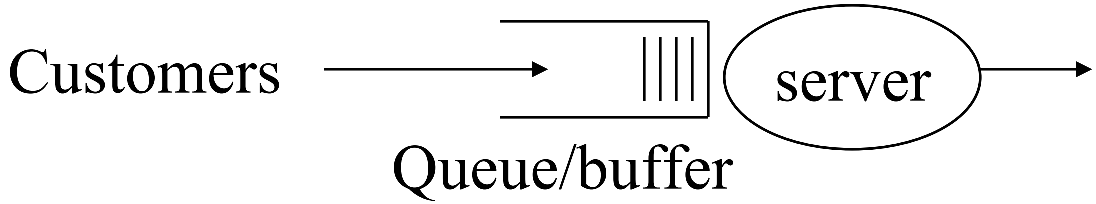
$$N(t) = \alpha(t) - \beta(t) \Rightarrow N_t = \frac{\sum_{i=1}^{\alpha(t)} T_i}{t}$$

$$N = \lim_{t\to\infty}\frac{\sum_{i=1}^{\alpha(t)} T_i}{t},\quad T = \lim_{t\to\infty}\frac{\sum_{i=1}^{\alpha(t)} T_i}{\alpha(t)} \Rightarrow \sum_{i=1}^{\alpha(t)} T_i = \alpha(t)T$$

$$N = \frac{\sum_{i=1}^{\alpha(t)} T_i}{t} = \left(\frac{\alpha(t)}{t}\right)\frac{\sum_{i=1}^{\alpha(t)} T_i}{\alpha(t)} = \lambda T$$

Application of little's Theorem

- **Little's Theorem can be applied to almost any system or part of it**
- **Example:**



1) The transmitter: D_{TP} = packet transmission time

- **Average number of packets at transmitter = $\lambda D_{TP} = \rho$ = link utilization**

2) The transmission line: D_p = propagation delay

- **Average number of packets in flight = λD_p**

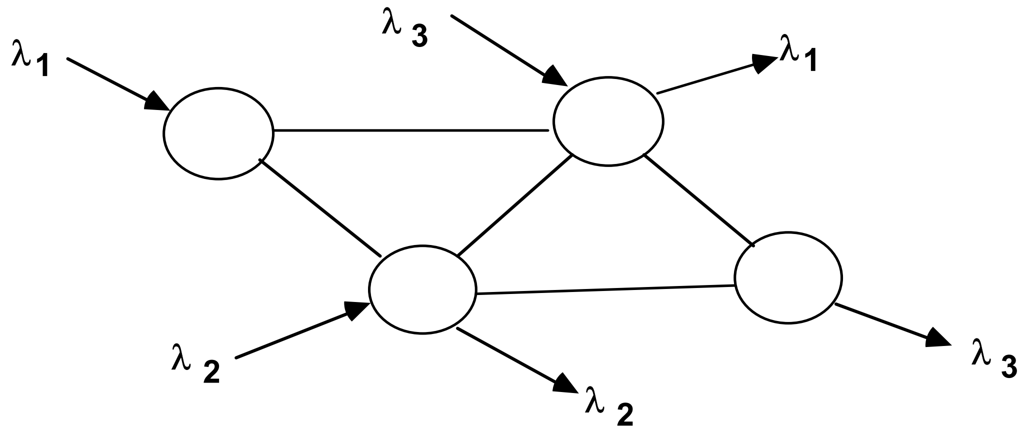
3) The buffer: D_q = average queueing delay

- **Average number of packets in buffer = $N_q = \lambda D_q$**

4) Transmitter + buffer

- **Average number of packets = $\rho + N_q$**

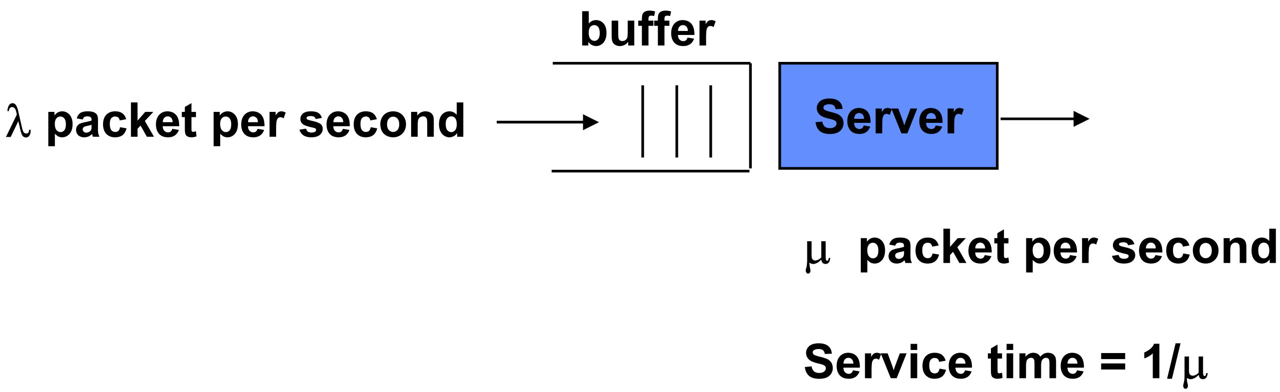
Application to complex system



- We have complex network with several traffic streams moving through it and interacting arbitrarily
- For each stream i individually, Little says $N_i = \lambda_i T_i$
- For the streams collectively, Little says $N = \lambda T$ where
- $N = \sum_i N_i$ & $\lambda = \sum_i \lambda_i$
- From Little's Theorem:

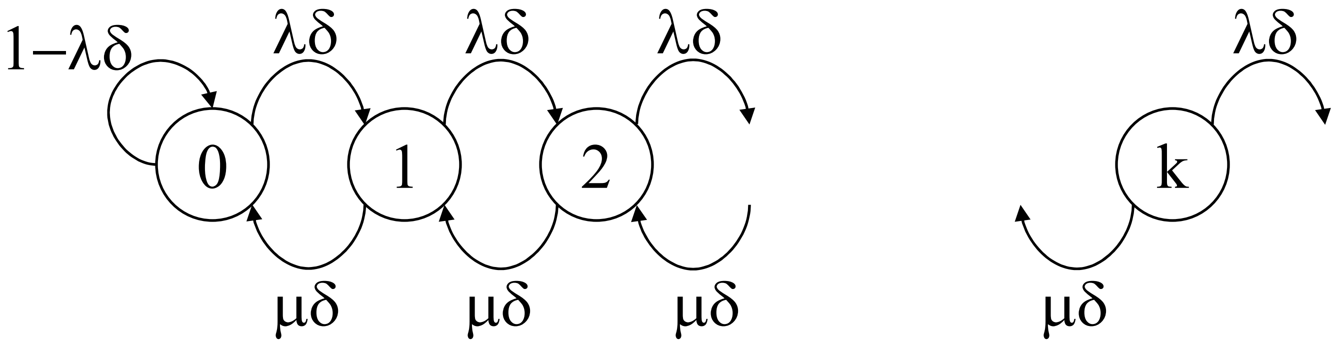
$$T = \frac{\sum_{i=1}^{i=k} \lambda_i T_i}{\sum_{i=1}^{i=k} \lambda_i}$$

Single server queues



- **M/M/1**
  - Poisson arrivals, exponential service times
- **M/G/1**
  - Poisson arrivals, general service times
- **M/D/1**
  - Poisson arrivals, deterministic service times (fixed)

Markov Chain for M/M/1 system



- **State k => k customers in the system**
- **P(I,j) = probability of transition from state I to state j**
  - **As δ => 0, we get:**

    P(0,0) = 1 - $\lambda\delta$,
    P(j,j) = 1 - $\lambda\delta - \mu\delta$

    P(j,j+1) = $\lambda\delta$
    P(j,j-1) = $\mu\delta$

    P(I,j) = 0 for all other values of I,j.

- **Birth-death chain: Transitions exist only between adjacent states**
  - $\lambda\delta$, $\mu\delta$ **are flow rates between states**

Equilibrium analysis

- **We want to obtain P(n) = the probability of being in state n**
- **At equilibrium λP(n) = μP(n+1) for all n**
  - **Local balance equations between two states (n, n+1)**
  - **P(n+1) = (λ/μ)P(n) = ρP(n),** $\rho = \lambda/\mu$
- **It follows: P(n) = ρ^n P(0)**
- **Now by axiom of probability:**

$$\sum_{i=0}^{\infty} P(n) = 1$$

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

$$\Rightarrow P(0) = 1 - \rho$$

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

Average queue size

$$N = \sum_{n=0}^{\infty} nP(n) = \sum_{n=0}^{\infty} n\rho^n (1-\rho) = \frac{\rho}{1-\rho}$$

$$N = \frac{\rho}{1-\rho} = \frac{\lambda / \mu}{1-\lambda / \mu} = \frac{\lambda}{\mu - \lambda}$$

- **N = Average number of customers in the system**
- **The average amount of time that a customer spends in the system can be obtained from Little's formula (N=λT => T = N/λ)**

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

- **T includes the queueing delay plus the service time (Service time = D_{TP} = 1/μ)**
  - **W = amount of time spent in queue = T - 1/μ =>**

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

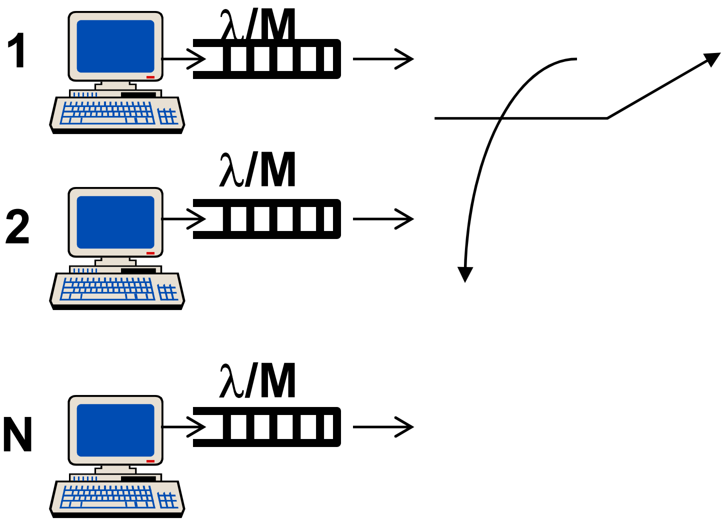
- **Finally, the average number of customers in the buffer can be obtained from little's formula**

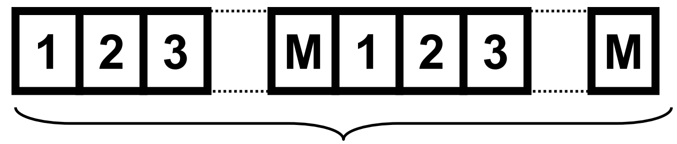
$$N_Q = \lambda W = \frac{\lambda}{\mu - \lambda} - \frac{\lambda}{\mu} = N - \rho$$

Example (fast food restaurant)

- Customers arrive at a fast food restaurant at a rate of 100 per hour and take 30 seconds to be served.
- How much time do they spend in the restaurant?
  - Service rate = μ = 60/0.5=120 customers per hour
  - T = $1/\mu-\lambda$ = 1/(120-100) = 1/20 hrs = 3 minutes
- How much time waiting in line?
  - W = T - $1/\mu$ = 2.5 minutes
- How many customers in the restaurant?
  - N = λT = 5
- What is the server utilization?
  - $\rho = \lambda/\mu$ = 5/6

Packet switching vs. Circuit switching



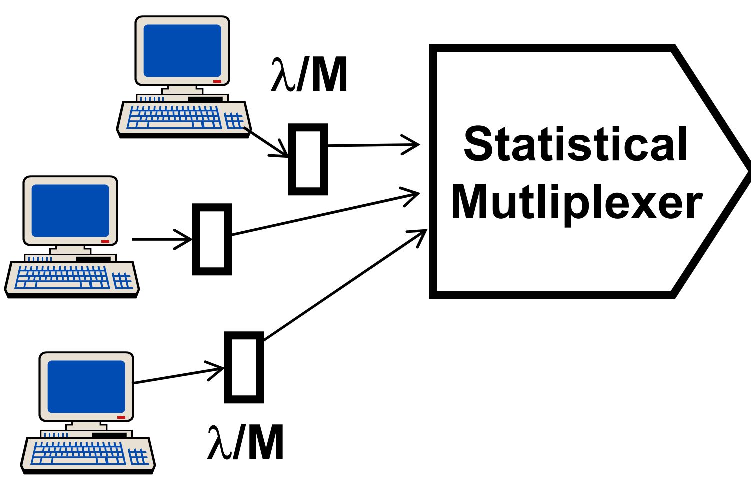


TDM, Time Division Multiplexing
Each user can send μ/N packets/sec and has packet arriving at rate λ/N packets/sec

$$D = M/\mu + \frac{M(\lambda/\mu)}{(\mu-\lambda)}$$

M/M/1 formula

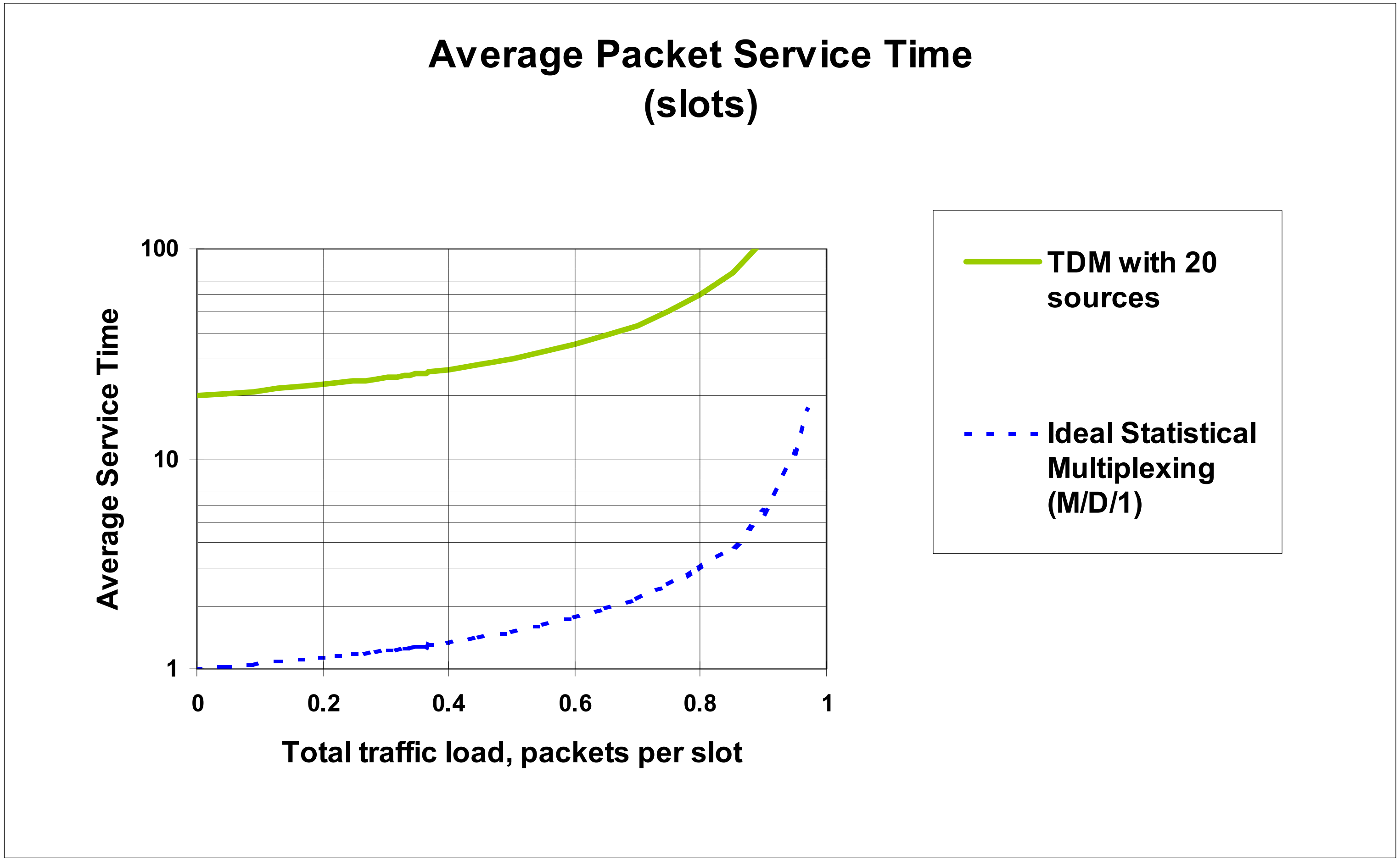
Packets generated at random times



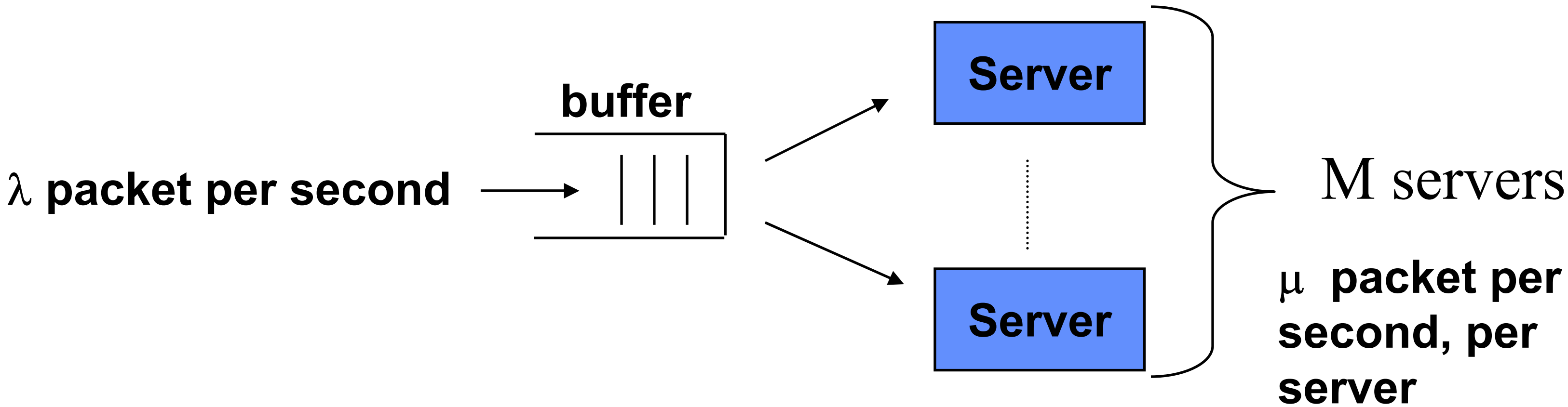
$$D = 1/\mu + \frac{(\lambda/\mu)}{(\mu-\lambda)}$$

M/M/1 formula

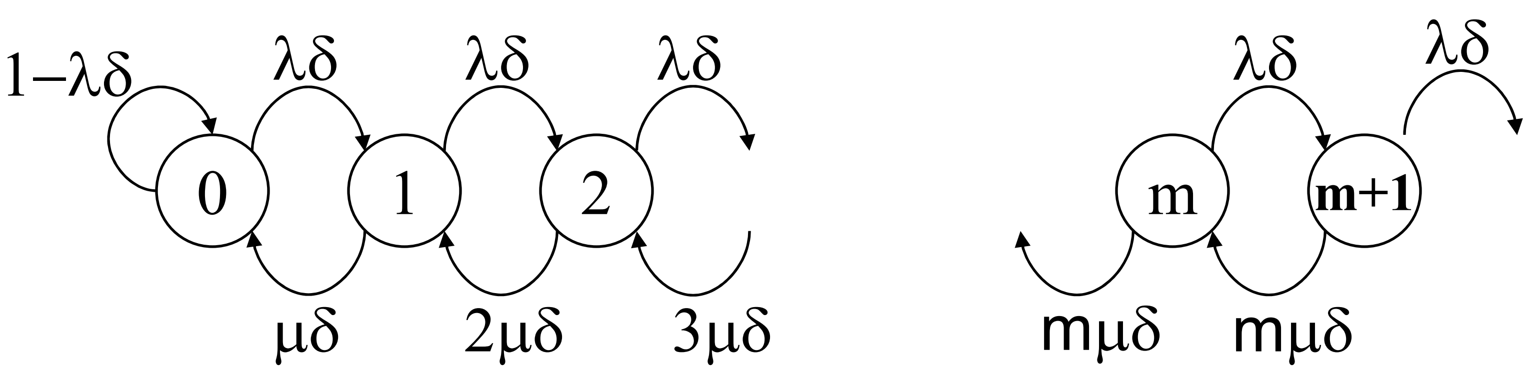
Circuit (tdm/fdm) vs. Packet switching



M server systems: M/M/m



- Departure rate is proportional to the number of servers in use
- Similar Markov chain:



M/M/m queue

- **Balance equations:**

$$\lambda P(n-1) = n\mu P(n) \quad n \le m$$

$$\lambda P(n-1) = m\mu P(n) \quad n > m$$

$$P(n) = \begin{cases} P(0)(m\rho)^n / n! & n \le m \\ P(0)(m^m \rho^n) / m! & n > m \end{cases}, \qquad \rho = \frac{\lambda}{m\mu} \le 1$$

- **Again, solve for P(0):**

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

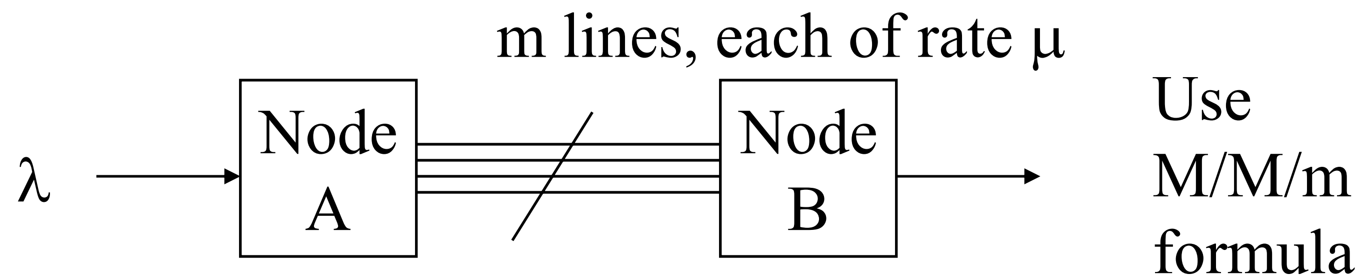
$$P_Q = \sum_{n=m}^{n=\infty} P(n) = \frac{P(0)(m\rho)^m}{m!(1-\rho)}$$

$$N_Q = \sum_{n=0}^{n=\infty} nP(n+m) = \sum_{n=0}^{n=\infty} nP(0)\left(\frac{m^m \rho^{m+n}}{m!}\right) = P_Q\left(\frac{\rho}{1-\rho}\right)$$

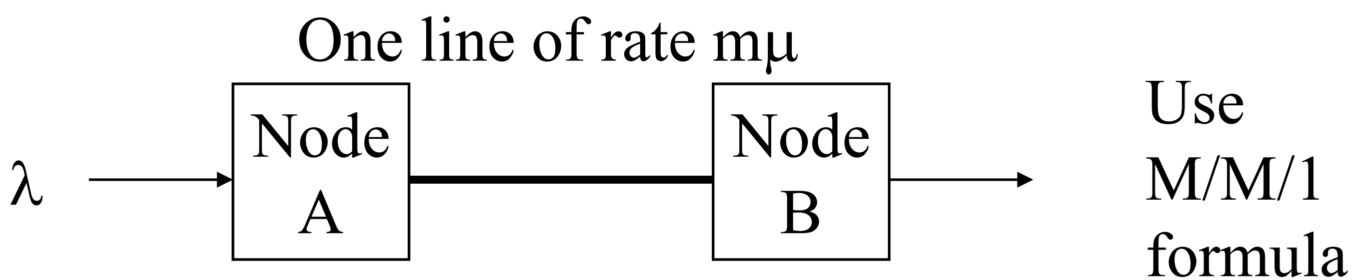
$$W = \frac{N_Q}{\lambda}, \; T = W + 1/\mu, \; N = \lambda T = \lambda/\mu + N_Q$$

Applications of M/M/m

- Bank with m tellers
- Network with parallel transmission lines



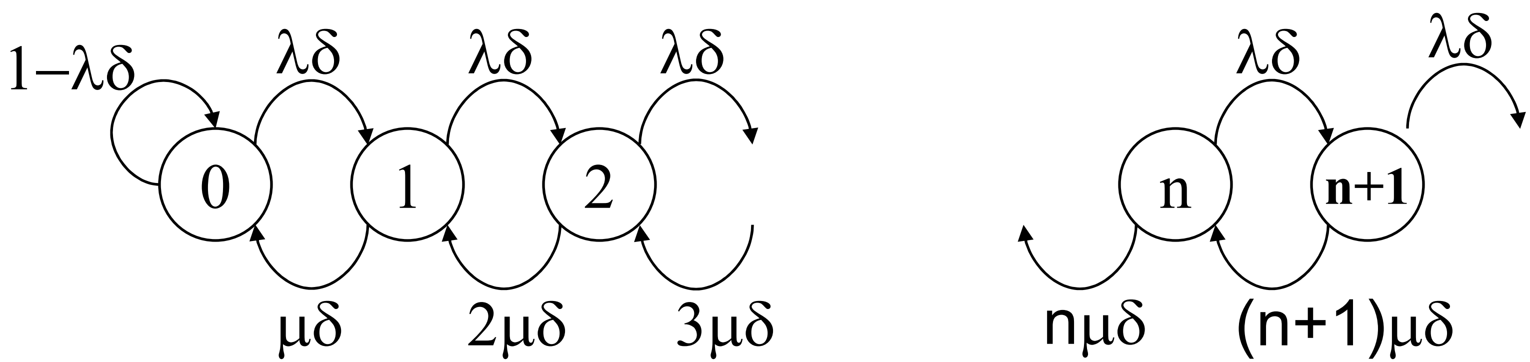
VS



- When the system is lightly loaded, PQ~0, and Single server is m times faster
- When system is heavily loaded, queueing delay dominates and systems are roughly the same

M/M/Infinity

- **Unlimited servers => customers experience no queueing delay**
- **The number of customers in the system represents the number of customers presently being served**



$$\lambda P(n-1) = n\mu P(n),\ \forall n > 1,\ \Rightarrow P(n) = \frac{P(0)(\lambda/\mu)^n}{n!}$$

$$P(0) = \left[1 + \sum_{n=1}^{\infty} (\lambda/\mu)^n / n!\right]^{-1} = e^{-\lambda/\mu}$$

$$P(n) = (\lambda/\mu)^n e^{-\lambda/\mu} / n! \Rightarrow Poisson\ distribution!$$

$$N = Average\ number\ in\ system = \lambda/\mu,\ T = N/\lambda = 1/\mu = service\ time$$

Blocking Probability

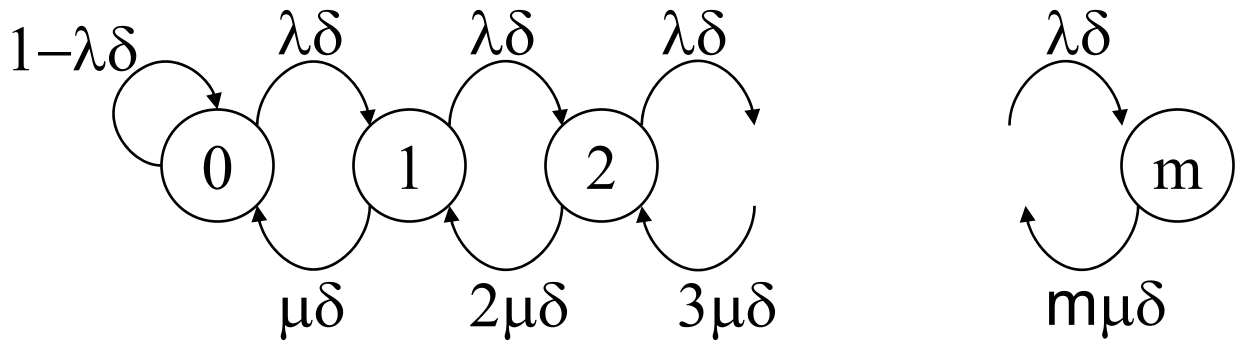
- **A circuit switched network can be viewed as a Multi-server queueing system**
  - Calls are blocked when no servers available - "busy signal"
  - For circuit switched network we are interested in the call blocking probability

- **M/M/m/m system**
  - m servers => m circuits
  - Last m indicated that the system can hold no more than m users

- **Erlang B formula**
  - Gives the probability that a caller finds all circuits busy
  - Holds for general call arrival distribution (although we prove Markov case only)

$$P_B = \frac{(\lambda / \mu)^m / m!}{\sum_{n=0}^{m} (\lambda / \mu)^n / n!}$$

M/M/m/m system: Erlang B formula



$$\lambda P(n-1) = n\mu P(n),\ 1 \le n \le m,\ \Rightarrow P(n) = \frac{P(0)(\lambda/\mu)^n}{n!}$$

$$P(0) = \left[\sum_{n=0}^{m} (\lambda/\mu)^n / n!\right]^{1}$$

$$P_B = P(Blocking) = P(m) = \frac{(\lambda/\mu)^m / m!}{\sum_{n=0}^{m} (\lambda/\mu)^n / n!}$$

Erlang B formula

$$P_B = \frac{(A)^m / m!}{\sum_{n=0}^{m} (A)^n / n!}$$

- **System load usually expressed in Erlangs**
  - **A= λ/μ = (arrival rate)*(ave call duration) = average load**
  - **Formula insensitive to λ and μ but only to their ratio**
- **Used for sizing transmission line**
  - **How many circuits does the satellite need to support?**
  - **The number of circuits is a function of the blocking probability that we can tolerate**
    Systems are designed for a given load predictions and blocking probabilities (typically small)
- **Example**
  - **Arrival rate = 4 calls per minute, average 3 minutes per call => A = 12**
  - **How many circuits do we need to provision?**
    Depends on the blocking probability that we can tolerate

Circuits	P_B
20	1%
15	8%
7	30%

Multi-dimensional Markov Chains

- **K classes of customers**
  - **Class j: arrival rate λ_j; service rate μ_j**
- **State of system: n = (n1, n2, …, nk); nj = number of class j customers in the system**

- **If detailed balance equations hold for adjacent states, then a product form solution exists, where:**

  - **P(n,.n2, …, nk) = P_1(n1)*P_2(n2)*…*P_k(nk)**

- **Example: K independent M/M/1 systems**

$$P_i(n_i) = \rho_i^{n_i}(1-\rho_i),\ \ \rho_i = \lambda_i / \mu_i$$

- **Same holds for other independent birth-death chains**

  - **E.g., M.M/m, M/M/Inf, M/M/m/m**

Truncation

- **Eliminate some of the states**
  - **E.g., for the K M/M/1 queues, eliminate all states where n1+n2+…+nk > K1 (some constant)**
- **Resulting chain must remain irreducible**
  - **All states must communicate**

Product form for stationary distribution of the truncated system

- **E.g., K independent M/M/1 queues**

$$P(n_1, n_2, \ldots n_k) = \frac{\rho_1^{n1}\rho_2^{n2}\ldots\rho_K^{nK}}{G}, \quad G = \sum_{n \in S} \rho_1^{n1}\rho_2^{n2}\ldots\rho_K^{nK}$$

- **E.g., K independent M/M/inf queues**

$$P(n_1, n_2, \ldots n_k) = \frac{(\rho_1^{n1}/n_1!)(\rho_2^{n2}/n_2!)\ldots(\rho_K^{nK}/n_k!)}{G}, \quad G = \sum_{n \in S} (\rho_1^{n1}/n_1!)(\rho_2^{n2}/n_2!)\ldots(\rho_K^{nK}/n_k!)$$

  - **G is a normalization constant that makes P(n) a distribution**
  - **S is the set of states in the truncated system**

Example

- **Two session classes in a circuit switched system**
  - M channels of equal capacity
  - Two session types:
    - Type 1: arrival rate λ1 and service rate μ1
    - Type 2: arrival rate λ2 and service rate μ2
- **System can support up to M sessions of either class**
  - If μ1= μ2, treat system as an M/M/m/m queue with arrival rate λ1+ λ2
  - When μ1=! μ2 need to know the number of calls in progress of each session type
  - Two dimensional markov chain state = (n1, n2)
  - Want P(n1, n2): n1+n2 <=m
- **Can be viewed as truncated M/M/Inf queues**
  - Notice that the transition rates in the M/M/Inf queue are the same as those in a truncated M/M/m/m queue

$$P(n_1, n_2) = \frac{(\rho_1^{n1} / n_1!)(\rho_2^{n2} / n_2!)}{G}, \qquad G = \sum_{i=0}^{i=m} \sum_{j=0}^{j=m-i} (\rho_1^i / i!)(\rho_2^j / j!), \quad n1 + n2 \le m$$

  - Notice that the double sum counts only states for which j+i <= m

PASTA: Poisson Arrivals See Time Averages

- The state of an M/M/1 queue is the number of customers in the system
- More general queueing systems have a more general state that may include how much service each customer has already received
- For Poisson arrivals, the arrivals in any future increment of time is independent of those in past increments and for many systems of interest, independent of the present state S(t) (true for M/M/1, M/M/m, and M/G/1).
- For such systems, $P\{S(t)=s|A(t+\delta)-A(t)=1\} = P\{S(t)=s\}$
  - (where A(t)= # arrivals since t=0)
- In steady state, arrivals see steady state probabilities

Occupancy distribution upon arrival

- **Arrivals may not always see the steady-state averages**

- **Example:**
  - **Deterministic arrivals 1 per second**
  - **Deterministic service time of 3/4 seconds**

λ = 1 packets/second T = 3/4 seconds (no queueing)

N = λT = Average occupancy = 3/4

- **However, notice that an arrival always finds the system empty!**

Occupancy upon arrival for a M/M/1 queue

a_n = Lim $_{t\text{-> inf}}$ (P (N(t) = n | an arrival occurred just after time t))
P_n = Lim $_{t\text{-> inf}}$ (P(N(t) = n))

For M/M/1 systems $a_n = P_n$

Proof: Let A(t, t+δ) be the event that and arrival occurred between t and t+δ

a_n (t) = Lim $_{t\text{-> inf}}$ (P (N(t) = n| A(t, t+δ))
= Lim $_{t\text{-> inf}}$ (P (N(t) = n, A(t, t+δ))/P(A(t, t+δ))
= Lim $_{t\text{-> inf}}$ P(A(t, t+δ)| N(t) = n)P(N(t) = n)/P(A(t, t+δ))

- Since future arrivals are independent of the state of the system,

  P(A(t, t+δ)| N(t) = n)= P(A(t, t+δ))

- Hence, a_n (t) = P(N(t) = n) = P_n(t)

- Taking limits as t-> infinity, we obtain $a_n = P_n$

- Result holds for M/G/1 systems as well