King Fahd University of Petroleum & Minerals Computer Engineering Dept

CSE 642 – Computer Systems Performance

Term 091

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

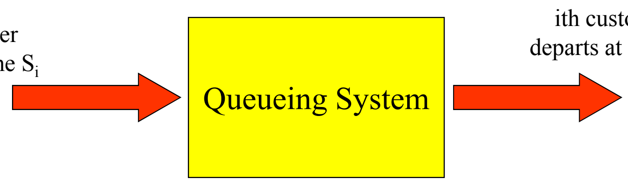
Email: ashraf@kfupm.edu.sa

Queuing Model

- **Consider the following system:**

A(t) $N(t) = A(t) - D(t)$ D(t)

ith customer arrives at time S_i



ith customer departs at time D_i

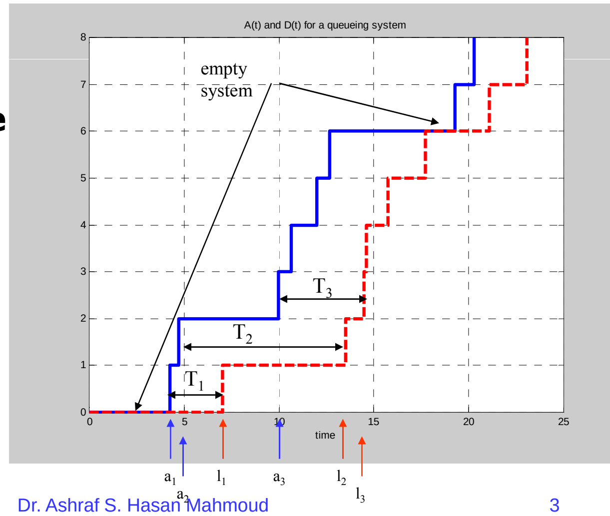
$T_i = D_i - A_i$

$W_i = T_i - S_i$
$= D_i - A_i - S_i$

A(t) – number of arrivals in (0, t]
D(t) – number of departures in (0, t]
N(t) – number of customers in system in (0,t]
T_i – duration of time spent in system for ith customer
W_i – duration of time spent waiting for service for ith customer

Example: Queueing System

- a_i and l_i arrival and departure instances
- $T_i = l_i - a_i$ is time spent in the system
- If A(t) = D(t) ➔ system is empty
- The graph is shown for FCFS service



Little's Formula

- Consider the time average of the number of customers in the system N(t) during (0,t],

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

i.e. average area under the curve for N(t)

$\langle N\rangle_t$ is also given by

$$\langle N\rangle_t = \frac{1}{t}\sum_{i=1}^{A(t)} T_i$$

Little's Formula – cont'd

- The average arrival rate $\langle\lambda\rangle_t$ is given by

$$\langle\lambda\rangle_t = \frac{A(t)}{t}$$

- Combining the previous equations we get:

$$\langle N\rangle_t = \langle\lambda\rangle_t \frac{1}{A(t)} \sum_{i=1}^{A(t)} T_i$$

- Let the quantity $\langle T\rangle_t$ be the average time a customer spends in the system, then

$$\langle T\rangle_t = \frac{1}{A(t)} \sum_{i=1}^{A(t)} T_i$$

Little's Formula – cont'd

- Combining the last two equations:

$$\langle N\rangle_t = \langle\lambda\rangle_t \langle T\rangle_t$$

- Which relates the time averages of the arrival rate, the number of customers in the system and the average time spent in the system
- Let t →∞, then one can write:

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

Under what conditions will $\langle N\rangle_t \rightarrow E[N]$ for $t \rightarrow \infty$?

Little's Formula – cont'd

- **Little's formula:**

E[N] = λE[T]

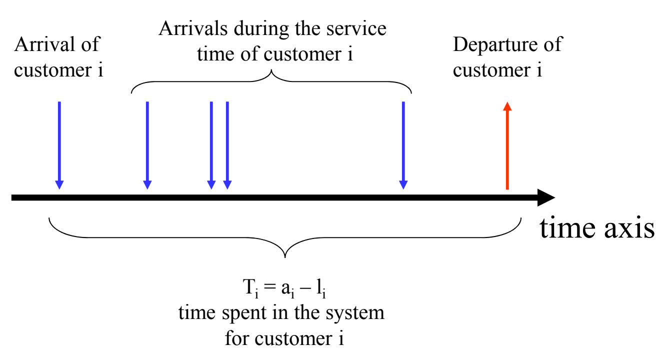
Holds for many service disciplines and for systems with arbitrary number of servers. It holds for many interpretations of the system as well

Note: $\sum_{i=1}^{A(t)} T_i = \sum_{i=1}^{A(t)} d_i - l_i = \sum_{i=1}^{A(t)} d_i - \sum_{i=1}^{A(t)} l_i$ **does not depend on the service order**

Intuitiveness of Little's Formula

- **Little's formula:**

E[N] = λE[T]



- **Formula applies to many interpretations of "system"!**

Example 1:

- **Problem: Let Ns(t) be the number of customers being served at time t, and let τ denote the service time. If we designate the set of servers to be the "system" then Little's formula becomes:**

$$E[Ns] = \lambda E[\tau]$$

where E[Ns] is the average number of busy servers for a system in the steady state.

Example 1: cont'd

Note: for a single server Ns(t) can be either 0 or 1 ➔ E[Ns] represents the portion of time the server is busy. If p_0 = Prob[Ns(t) = 0], then we have

$$1 - p_0 = E[Ns] = \lambda E[\tau], \text{ Or}$$
$$p_0 = 1 - \lambda E[\tau]$$

The quantity $\lambda E[\tau]$ is defined as the utilization for a single server. Usually, it is given the symbol ρ

$$\rho = \lambda E[\tau]$$

For a c-servers system, we define the utilization (the fraction of busy servers) to be

$$\rho = \lambda E[\tau] / c$$

Poisson Process

- Refer to the Summation process example in the Random Processes package
- Def: Poisson process to be the point process for which the number of events (successes), X(t), in a t-second interval is given by the Poisson distribution

$$P_k(t) = P(X(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \qquad k = 0,1,\ldots$$

where λ is the average rate of success per time unit

Poisson Process - Properties

- The random process X(t) is a Markov Process. For arbitrary times: $t_1<t_2<\ldots<t_k<t_{k+1}$

$$\text{Prob}[X(t_{k+1}) = x_{k+1}/X(t_k)=x_k, \ldots, X(t_1)=x_1]$$

$$= \text{Prob}[X(t_{k+1}) = x_{k+1}/X(t_k)=x_k]$$

- Independent increments
- Stationary increments

Poisson Process – Interarrival Time

- Let T be the random time between two consecutive events
- The distribution function is given by

$$
\begin{aligned}
F_T(t) &= P(T \le t) \\
&= P(\text{at least one arrival in } t \text{ seconds}) \\
&= 1 - P(0 \text{ arrivals in } t \text{ seconds}) \\
&= 1 - P_0(t) \\
&= 1 - e^{-\lambda t}
\end{aligned}
$$

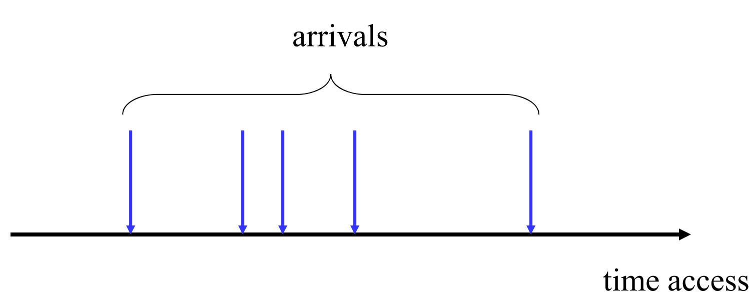
Therefore $f_T(t)$ is equal to $\lambda e^{-\lambda t}$ for $t \ge 0$

- Poisson Process ≡ interarrival times are independent and exponentially distributed

11/17/2009 Dr. Ashraf S. Hasan Mahmoud 13

Uniformity Property

- Def – give a number of arrivals in an interval, the arrivals are uniformly distributed throughout the interval!



11/17/2009 Dr. Ashraf S. Hasan Mahmoud 14

Uniformity Property – cont'd

- **Proof:**

Suppose that we are given than one arrival occurs in the interval [0,t],
Let Y be the arrival time of the single customer → 0 < y < t
Let X(y) be the number of events up to time y → X(t) – X(y) is the increment in the interval (y, t]

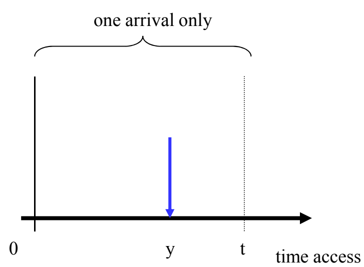
$$P(Y \le y) = P(X(y) = 1 \,/\, X(t) = 1]$$

$$= \frac{P(X(y) = 1 \text{ and } X(t) - X(y) = 0]}{P(X(t) = 1)}$$

$$= \frac{P(X(y) = 1)\, P(X(t) - X(y) = 0)}{P(X(t) = 1)}$$

$$= \frac{\lambda y e^{-\lambda y}\, e^{-\lambda(t-y)}}{\lambda t e^{-\lambda t}}$$

$$= y / t$$

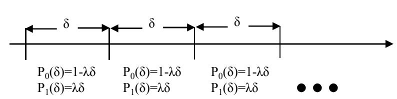


Kolmogorov Forward Differential Equations

- **Consider the incremental time interval δ, so small that λδ << 1 for all λ**
- **Using the Poisson density function and knowing that $e^{-\lambda\delta} \approx 1 - \lambda\delta + O(\delta)$ – where O(δ) are higher order terms of δ (i.e. lim O(δ)/ δ = 0 as δ → 0)**
- **One can write:**

$$P_0(\delta) = 1 - \lambda\delta + O(\delta)$$
$$P_1(\delta) = \lambda\delta + O(\delta)$$
$$P_i(\delta) = O(\delta) \quad \text{for } i \ge 2$$

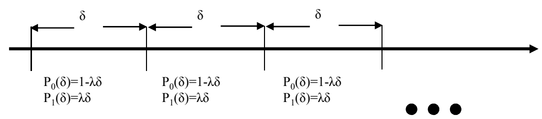
This means, we choose δ small such that the likelihood of more than one arrival during δ is close to zero



Sequence of iid Bernoulli experiments

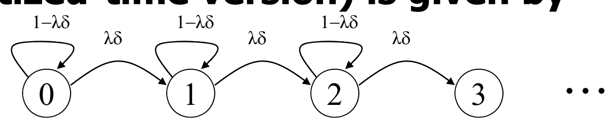
Kolmogorov Forward Differential Equations – cont'd

- **This means, we choose δ small such that the likelihood of more than one arrival during δ is close to zero**



Sequence of iid Bernoulli experiments

- **The corresponding state diagram (for the discretized-time version) is given by**



Kolmogorov Forward Differential Equations – cont'd

- **Let us study the evolution of $P_n(t)$ with respect to time, t**
  - **Remember $P_n(t)$ is the probability of n arrivals in an interval t**
- **Consider the change in $P_n(t)$ in the incremental interval (t, t+ δ)**

Kolmogorov Forward Differential Equations – cont'd

- Case n = 0

$P_0(t+\delta) = P(\text{no arrivals in } (0,t+\delta))$
$= P(\text{no arrivals in } (0,t))\, P(\text{no arrivals in } (t,t+\delta))$
$= P_0(t)(1 - \lambda\delta)$

- Case n > 0

$P_n(t+\delta) = P(n \text{ arrivals in } (0,t+\delta))$
$= P(n \text{ arrivals in } (0,t))\, P(\text{no arrivals in } (t,t+\delta))$
$+ P(n\text{-}1 \text{ arrivals in } (0,t))\, P(1 \text{ arrival in } (t,t+\delta))$
$= P_n(t)(1 - \lambda\delta) + P_{n-1}(t)(\lambda\delta)$

- The above equations can be written as

$[P_0(t+\delta) - P_0(t)]/\delta = -\lambda P_0(t)$, and

$[P_n(t+\delta) - P_n(t)]/\delta = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad n > 0$

Kolmogorov Forward Differential Equations – cont'd

- Take the limit as $\delta \to 0$, the previous equations can be written as:

$dP_0(t)/dt = -\lambda P_0(t)$, and

$dP_n(t)/dt = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad n > 0$

- Verify that $P_k(t)$ given by

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \qquad k = 0,1,\ldots$$

is a solution for the Kolmogorov Forward differential equations

Kolmogorov Forward Differential Equations – cont'd

- **Another form for the Kolmogorov D.E. is as follows:**

$$\frac{d\tilde{P}(t)}{dt} = \Lambda \tilde{P}(t)$$

where $\tilde{P}(t) = [P_0(t) \quad P_1(t) \quad P_2(t) \quad ....]^T$

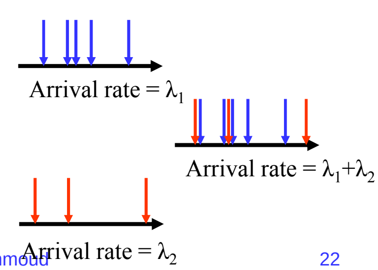
$$\Lambda = \begin{bmatrix} -\lambda & 0 & 0 & 0 & \cdots \\ \lambda & -\lambda & 0 & 0 & \cdots \\ 0 & \lambda & -\lambda & 0 & \cdots \\ 0 & 0 & \lambda & -\lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Λ is the infinitesimal generator matrix

Note sum of columns = zero

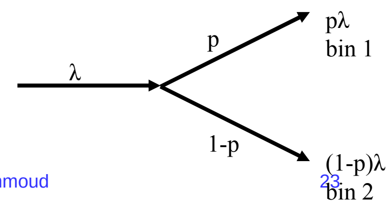
Adding Poisson Processes

- **Sum of two INDEPENDENT Poisson processes**
- **Consider an incremental interval δ**
  - **The probability of an arrival from either source is $\lambda_1\delta(1-\lambda_2\delta)+(1-\lambda_1\delta)\lambda_2\delta \approx (\lambda_1+\lambda_2)\delta$**
  - **The probability of arrivals from both source is $\lambda_1\delta\,\lambda_2\delta = \lambda_1\lambda_2\delta^2 \approx 0$**
- **Therefore, the sum is a Poisson process with rate $(\lambda_1+\lambda_2)$**



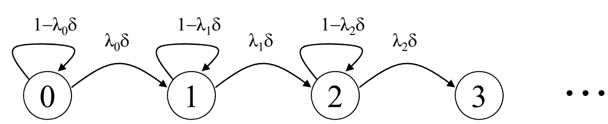
Splitting Poisson Processes

- Splitting of a Poisson processes
- Consider an incremental interval δ
  - The probability of an arrival to bin 1: λδp
  - The probability of an arrival to bin 1: λδ(1-p)
  - Since subsequence arrivals to either bins are independent and identically distributed
- Therefore, the arrivals processes to bin 1 and 2 Poisson with rate pλ and (1-p)λ, respectively



Pure Birth Processes

- Poisson process is a member of a wider class of "pure birth processes"
- In general the probability of an arrival in an interval δ can be function of the number in the system, $\lambda_n\delta$
- The corresponding state diagram will be



Pure Birth Processes – cont'd

- **In the same manner, you can show that the corresponding Kolmogorov D.E are given by**

$$dP_0(t)/dt = -\lambda_0 P_0(t), \text{ and}$$

$$dP_n(t)/dt = -\lambda_n P_n(t) + \lambda_{n-1} P_{n-1}(t), \quad n > 0$$

Subject to the condition $\sum_{n=0}^{\infty} P_n(t) = 1$

Pure Birth Processes – cont'd

- **Putting the Kolmogorov D.E.s in a matrix form:**

$$\frac{d\widetilde{P}(t)}{dt} = \Lambda \widetilde{P}(t)$$

Necessary and sufficient condition for stability is $\sum 1/\lambda_n = \infty$

where $\widetilde{P}(t) = [P_0(t) \quad P_1(t) \quad P_2(t) \quad \ldots]^T$

$$\Lambda = \begin{bmatrix} -\lambda_0 & 0 & 0 & 0 & \cdots \\ \lambda_0 & -\lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_1 & -\lambda_2 & 0 & \cdots \\ 0 & 0 & \lambda_2 & -\lambda_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Λ is the infinitesimal generator matrix

Note sum of columns = zero

Example: Yule-Furry Process

- For Yule-Furry process, $\lambda_n = n\,\lambda$ – linear rate with system population
- The evolution equations are then given by

$$dP_n(t)/dt = -n\lambda P_n(t) + (n-1)\lambda P_{n-1}(t); \quad n \geq k$$

- For the initial condition $P_k(0)=1$ for some k > 0, show that

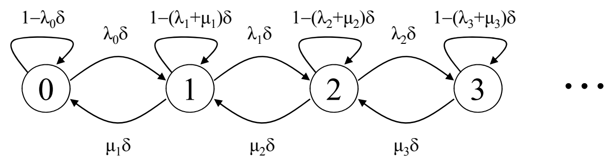
$$P_n(t) = \binom{n-1}{k-1} e^{-n\lambda t}\left(1-e^{-\lambda t}\right)^{n-k} \qquad n \geq k, t \geq 0$$

is a solution

Poisson Arrivals See Time Averages (PASTA)

Birth And Death Processes

- **The corresponding state diagram is as shown:**



- **The Kolmogorov D.E are given by**

$dP_0(t)/dt = -\lambda_0 P_0(t) + \mu_1 P_1(t)$, and

$dP_n(t)/dt = -(\lambda_n+\mu_1)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t)$, $n > 0$

Subject to the condition $\sum_{n=0}^{\infty} P_n(t) = 1$

Birth And Death Processes – cont'd

- **Putting the Kolmogorov D.E.s in a matrix form:**

$$\frac{d\tilde{P}(t)}{dt} = M\tilde{P}(t)$$

+ve solution exists if $0 \le \lambda_n < \mu_n$

where $\tilde{P}(t) = [P_0(t) \quad P_1(t) \quad P_2(t) \quad \ldots]^T$

$$M = \begin{bmatrix} -\lambda_0 & \mu_1 & 0 & 0 & \ldots \\ \lambda_0 & -\lambda_1-\mu_1 & \mu_2 & 0 & \ldots \\ 0 & \lambda_1 & -\lambda_2-\mu_2 & \mu_3 & \ldots \\ 0 & 0 & \lambda_2 & -\lambda_3-\mu_3 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Global Balance Equations

- **Steady state solution → dP(t)/dt = 0**
- **The resulting set of equations:**

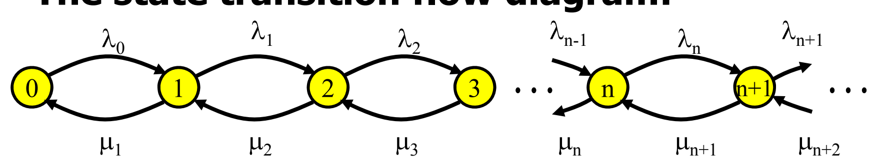
$$\lambda_0 P_0 = \mu_1 P_1, \text{ and}$$

$$(\lambda_n + \mu_n) P_n = \lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1}, \quad n > 0$$

In addition to the normalizing condition $\sum_{n=0}^{\infty} P_n = 1$

Global Balance Equations – cont'd

- **The state transition flow diagram:**



- **We can show the solution for the global balance equation is given by**

$$P_n = P_0 \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i}$$

and

$$P_0 = \left[1 + \sum_{n=1}^{\infty} \prod_{i=1}^{n} \frac{\lambda_{i-1}}{\mu_i} \right]^{-1}$$

The basis for all queueing formula to come!!

Queueing Models: M/M/1

- **Making the substitutions: $\lambda_n = \lambda$ and $\mu_n = \mu$, and defining $\rho = \lambda/\mu$, one can write**

$$P_n = (1-\rho)\rho^n \qquad n = 0,1,2,...$$

or

$$P(z) = \frac{1-\rho}{1-z\rho}$$

- **The mean and variance of number of customers in system, E[N] and Var[N] are given by**

$$E[N] = \frac{\rho}{1-\rho} \qquad Var[N] = \frac{\rho}{(1-\rho)^2}$$

- **The mean delay in the M/M/1 queue can be obtained through the application of Little's formula:**

$$E[D] = E[N]/\lambda = \frac{1}{\mu-\lambda}$$

M/M/1- Delay Distribution

- **The probability of n customers as a departing customer departs after spending t seconds in system is given by**

$$P[n \text{ customers in system/delay of departing customer} = t] = \frac{(\lambda t)^n \exp(-\lambda t)}{n!}$$

or

$$P_n = \int_0^\infty \frac{(\lambda t)^n e^{-\lambda t}}{n!} d(t)dt \qquad n = 0,1,...$$

$$P(z) = \sum_{n=0}^\infty p_n z^n = \sum_{n=0}^\infty z^n \int_0^\infty \frac{(\lambda t)^n}{n!} e^{-\lambda t} d(t)dt$$

$$P(z) = \int_0^\infty e^{-\lambda t(1-z)} d(t)dt = D(\lambda(1-z))$$

Note this probability is the same as the probability of n customers in system –

$$P(z) = \frac{(1-\rho)}{1-\rho z}$$

M/M/1- Delay Distribution

- **Also equal to the probability of finding n customers in system by an arriving customer (refer to PASTA property)**

$$\frac{(1-\rho)}{1-\rho z} = D\left(\lambda(1-z)\right)$$

Since d(t) is the PDF for the total delay time
Therefore, D(s) is given by

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

i.e. the delay for M/M/1 queue is exponentially distributed with mean 1/(μ- λ),

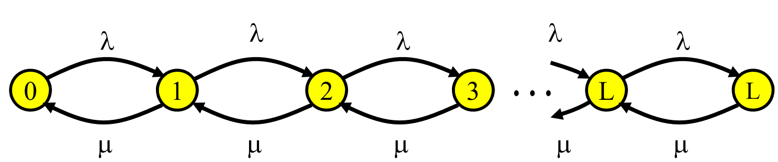
$$d(t) = (\mu-\lambda)e^{-(\mu-\lambda)t} \qquad t \geq 0$$

Queueing Models: M/M/1/L

- **Finite Capacity Case: $\lambda_j = \lambda$ for $j<L$**
  0 for $j \geq L$

  also $\mu_j = \mu$

- **The state-transition flow diagram of M/M/1/L queue is as shown below**



Queueing Models: M/M/1/L – cont'd

Steady-state pmf is given by

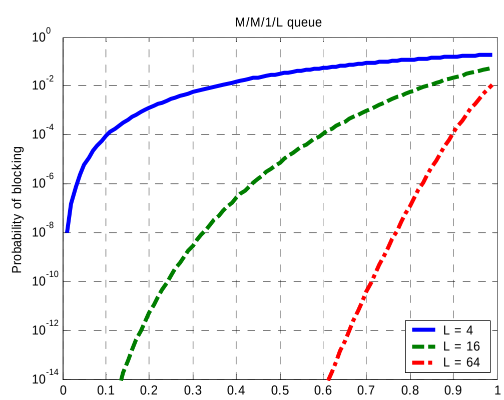
$$P_n = \begin{cases} \dfrac{(1-\rho)\rho^n}{1-\rho^{L+1}} & n \le L \\ 0 & n > L \end{cases}$$

- **What is P(z) equal to?**
- **In particular, the blocking probability, P_L, is given by the relation above for n = L**

Queueing Models: M/M/1/L – cont'd

- **In particular, the blocking probability, P_L, is given by the relation above for n = L**

$$P_L = \frac{(1-\rho)\rho^L}{1-\rho^{L+1}}$$



Example: M/M/1/L – cont'd

- **Problem: A voice signal is digitized at a rate of 8000 bps. The average length of a voice message is 3 min. Messages are transmitted on a DS-1 line, which has the capacity of 1.344 Mbps. While waiting for transmission, the messages are stored in a buffer which has a capacity of 10^7 bit. Plot the blocking probability versus the voice message arrival rate.**

Example: M/M/1/L – cont'd

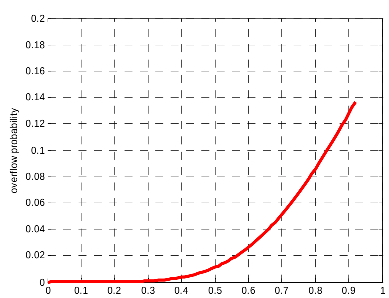
- **Solution:**

```
%
% Example 3.7 - voice multiplexing - page 91
clear all
LineWidth = 3;

DS1_Capacity     = 1.344e6; % bits/sec
BuffSizeBits     = 1e7; % different than textbook
BPSPerVoiceMsg   = 8000;% bps per voice msg
VoiceMsgDuration = 3*60; % second;
VoiceMsgSizeBits = VoiceMsgDuration * BitsPerVoiceMsg;
ServiceTime      = VoiceMsgSizeBits / DS1_Capacity;
% # of msgs buffer can fit
BufferSizeMsgs   = floor(BuffSizeBits/VoiceMsgSizeBit

Step  = 0.01;
Lamda = [0:Step:(1-Step)/ServiceTime];
Rho   = Lamda * ServiceTime;
PB    = (1-Rho).*Rho.^BufferSizeMsgs./(1-Rho.^(BufferSizeMsgs+1));
%
% Plot results
figure(1)
h = plot(Lamda, PB,'-r');
set(h, 'LineWidth', LineWidth);
xlabel('voice message arrival rate'); grid
ylabel('overflow probability');
axis([0 1 0 0.2]);
```

Note since voice message size is 1440000 bits, then buffer size can not be 10^6 bits as stated in the textbook. Here we use buffer size of 10^7 bits which means, buffer can accommodate 6 voice messages before it overflows.
Refer to example 3.7 page 91 in textbook



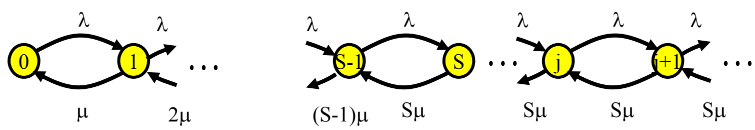
Queueing Models: M/M/S – Multiserver Systems

- **Assume S servers system, therefore:**

$$\mu_j = j\mu \text{ for } j \le S$$
$$\quad S\mu \text{ for } j > S$$

and $$\lambda_j = \lambda \text{ for all } j$$

- **The state-transition flow diagram of M/M/S queue is as shown below**



Erlang C model – Blocked calls are QUEUED

Queueing Models: M/M/S – Multiserver Systems – cont'd

- **Solving the balance equations, results in**

$$P_j = \begin{cases} \dfrac{P_0 \rho^j}{j!} & j \le S \\ \dfrac{P_0 \rho^j}{S! S^{j-S}} & j > S \end{cases}$$

P_0 is calculated as $$P_0 = \left[\sum_{j=0}^{S-1} \frac{\rho^j}{j!} + \frac{S\rho^S}{S!(S-\rho)} \right]^{-1}$$

- **The traffic utilization, $\rho = \lambda / \mu$**
- **Note the condition for solution validity is $\rho/S < 1$**

i.e. in the S-server case, the traffic load ranges 0 to S.

Queueing Models: M/M/S – Multiserver Systems – cont'd

- **The probability of queueing is equal to the probability of finding all S servers busy, therefore,**

$$P_c(S,\rho)=\sum_{j=S}^{\infty}P_j=P_0\frac{\rho^S}{S!}\frac{S}{(S-\rho)}$$

- **The mean number of customers in queue, E[Nq], is given by**

$$\overline{Q}=E[Nq]=\sum_{j=0}^{\infty}jP_{j+S}=P_0\frac{\rho^S}{S!}\frac{S\rho}{(S-\rho)^2}$$

Queueing Models: M/M/S – Multiserver Systems – cont'd

- **Therefore, the relation between average number of customers in queue and probability of queueing is given by**

$$\overline{Q}=\frac{P_c\rho}{(S-\rho)}$$

- **Applying Little's formula to compute the average queue delay**

$$\overline{D}_Q=\frac{\overline{Q}}{\lambda}=\frac{P_c\rho}{\lambda(S-\rho)}$$

Exercise: M/M/S/∞

- **Show that the waiting time distribution is given by**

$$F_W(x) = 1 - \frac{p_c S}{S - \rho} e^{-\mu(S-\rho)x} \qquad x > 0$$

Example: M/M/S/∞

- **<u>Problem</u>: a 160 kb/s line is used for data transmission. Two options are provided**

  a) Implement a 16-channel TDM scheme where every channel provides 10 kb/s.

  b) Use the overall trunk as one *fat* data transmission pipe.

  Assume data frames arrive at a Poisson rate λ and are exponentially distributed in length with average of 2000 bits per frame.

  Which scheme provides less delay?

Example: M/M/S/∞ - cont'd

- **Solution:**

  a) S = 16 servers – Model M/M/S

  R_c = 10 kb/s → E[τ] = 1/ μ = 2000/10 = 200 msec

  ρ = λ/μ = λ E[τ] = 200 λ

  E[T] = E[W] + E[τ] = E[Nq]/ λ + E[τ]

  = P_c (1/μ) /(S- ρ) + E[τ]

  b) S = 1 server – Model M/M/1

  R_c = 160 kb/s → E[τ] = 1/ μ = 2000/160 = 1.25 msec

  ρ = λ/μ = λ E[τ] = 1.25 λ

  E[T] = E[W] + E[τ] = E[Nq]/ λ + E[τ]

  = 1/ (μ – λ)

Example: M/M/S/∞ - cont'd

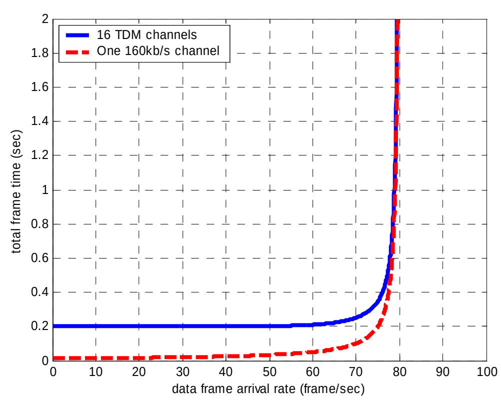
- **Solution:**

  For option (a)
  - minimum service time is equal to 200 msec

  For option (b)
  - minimum service time is equal to 1.25 msec

  Option (b) provides better (less) system

  Note: The x-axis in the textbook graph is not correct (Example 3.8 page 94). Verify?



Example: M/M/S/∞ - cont'd

```
%
% Example 3.8 - voice multiplexing - page 94
clear all
LineWidth = 3;

Line_Capacity    = 160e3; % bits/sec
NoOfChannels     = 16; % No of TDM channels
RateTDMChannel   = Line_Capacity/NoOfChannels;% bps per channel
AvgFrameSizeBits = 2000; % bits
%
% option (a) - 16 TDM channels - M/M/S queue
ServiceTime_a = AvgFrameSizeBits / RateTDMChannel;
S             = NoOfChannels;
Step          = 0.05;
Lamda_a       = [Step:Step:S/ServiceTime_a - Step];

Rho_a         = Lamda_a * ServiceTime_a;
[P0 PS Pc]    = Get_M_M_S(S, Rho_a);
W_a           = Pc.*Rho_a./(S-Rho_a)./Lamda_a;
T_a           = W_a + ServiceTime_a;
%
% option (b) - 1 160 kb/s channel - M/M/1 queue

ServiceTime_b = AvgFrameSizeBits / Line_Capacity;
Step          = 0.05;
Lamda_b       = [Step:Step:1/ServiceTime_b-Step];
Rho_b         = Lamda_b * ServiceTime_b;
T_b           = 1./(1./ServiceTime_b - Lamda_b);
%
% Plot results
figure(1)
h = plot(Lamda_a, T_a,'-', Lamda_b, T_b,'--r');
set(h, 'LineWidth', LineWidth);
xlabel('data frame arrival rate (frame/sec)'); grid
ylabel('total frame time (sec)');
legend('16 TDM channels', 'One 160kb/s channel', 2);
axis([0 100 0 2]);
```

```
function [P0, PS, Pc]    = Get_M_M_S(S, Rho);
% compute P0, PS, and Pc for an M/M/S queue given S and Rho
P0 = zeros(size(Rho));
PS = zeros(size(Rho));
Pc = zeros(size(Rho));

temp = zeros(size(Rho));
for i=0:S-1
    temp = temp + Rho.^i./factorial(i);
end
temp = temp + S.*Rho.^S./(factorial(S).*(S - Rho));

P0 = 1./temp;
PS = P0 .* Rho.^S./factorial(S);
Pc = PS .* S./(S - Rho);
```

Code to generate key probabilities (P_0, P_S, P_c) for M/M/S system

Code to compare between options (a) and (b)

Queueing Models: M/M/S/L

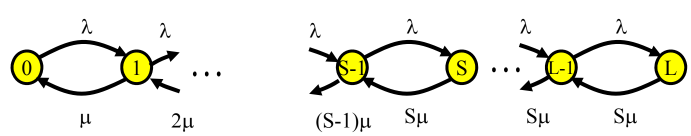
- **S server model with finite waiting room**
- **Assuming L ≥ S, we have**

$$\mu_j = j\mu \text{ for } j \le S$$
$$\quad S\mu \text{ for } j > S$$

and

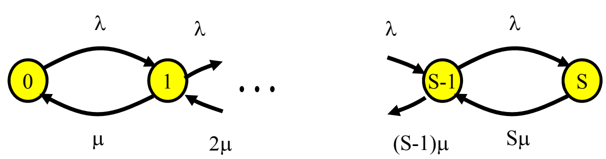
$$\lambda_j = \lambda \quad \text{for } j < L$$
$$\quad 0 \quad \text{for } j \ge L$$

- **The state transition flow diagram M/M/S/L queue**



Queueing Models: M/M/S/S

- **Special case of M/M/S/L where L = S;**
- **The state transition flow diagram M/M/S/S queue**



Erlang B model – Blocked calls are CLEARED

Queueing Models: M/M/S/S – cont'd

- **Solving the balance equation yields:**

$$P_n = \frac{P_0 \rho^n}{n!} \qquad n = 0,1,2,\ldots,S$$

and

$$P_0 = \left[\sum_{n=0}^{S} \frac{\rho^n}{n!} \right]^{-1}$$

- **When an arrival finds all S servers busy, it is blocked or dropped (no waiting room) – Probability of blocking is given by**

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

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

where $P_B(0,\rho) = 1$

- **Insensitivity Property of Erlang-B formula: Blocking probability does NOT DEPEND on the distribution of the service time, but rather its mean!!**

Example: M/M/S/S

- **Problem: constant length frames of 1000 bit each arrive an a multiplexer which has 16 output lines, each operating at a 50 kb/s rate. Suppose that frames arrive at an average rate of 1,440,000 frame per hour. There is no storage; thus if a frame is not served immediately it lost.**

  Calculate the blocking probability at the multiplexer.

Example: M/M/S/S – cont'd

- **Solution:**

  frame arrival rate, λ = 1,440,000 frame/hour
  = 400 frame/sec
  frame service time, $1/\mu$ = 1000 / 50 kb/s
  = 0.02 sec
  Traffic intensity, ρ = λ / μ = 8
  Number of servers, S = 16 (verify $\rho/S < 1$)

  Using the iterative formula ➔

S	1	2	3	4	5	6	7	8
$P_B(S, \rho)$	0.8889	0.7805	0.6755	0.5746	0.4790	0.3898	0.3082	0.2356
S	9	10	11	12	13	14	15	16
$P_B(S, \rho)$	0.1731	0.1217	0.0813	0.0514	0.0307	0.0172	0.0091	0.0045

our answer

M/M/S/S – Infinite Servers Case

- **Special case of the M/M/S/S queue**
- **Let S → ∞, i.e. an arriving customer always has a server available**
- **The probability of system in state zero is given by**

$$P_0 = \left[\sum_{n=0}^{\infty} \frac{\rho^n}{n!} \right]^{-1} = e^{-\rho}$$

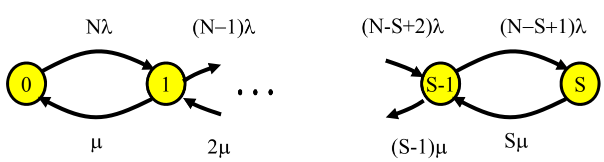
- **Therefore, the probability of system in state n ≥ 0 is computed as**

$$P_n = \frac{\rho^n}{n!} e^{-\rho}$$

Which is the Poisson distribution!!

Finite Source Queueing – Engset Distribution

- **Assume a finite population of N – each generate a message with rate λ (or with probability λδ in the interval (t, t+δ)). The next message is not transmitted till the prior one is served. Assume no storage case, i.e. if a source generates a message when no server is available, the message is lost and the source returns to idle state immediately.**
- **The state transition flow diagram is as shown:**



Finite Source Queueing – Engset Distribution – cont'd

- **The departure and arrival rates are**

$$\mu_n = n\mu \qquad n \le S$$
$$\lambda_n = (N-n)\lambda \qquad n \le S-1$$

- **You can show that the pmf is given by**

$$P_n = P_0 \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n \qquad n = 0,1,\ldots,S$$

and

$$P_0 = \left[\sum_{n=0}^{S} \binom{N}{n} \left(\frac{\lambda}{\mu}\right)^n\right]^{-1}$$

- **Remember P_S is the probability of blocking**

Finite Source Queueing – Engset Distribution – cont'd (2)

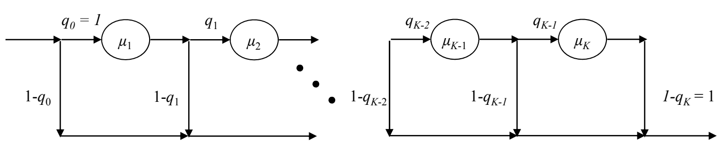
- **Consider the case for N ≤ S**
  - **Derive Pn and P0**
  - **Is there a blocking probability?**

Exercise

- **The textbook provides the final answers – but you need to show the solutions!!**

More Generalization – Cox Network

- **Consider the network of stages shown – Cox Network**
- **Prob of going through exactly i stages:** $\prod_{j=0}^{i-1} q_j (1-q_i)$
- **Assume q_0 = 1, q_K = 0, then** $\sum_{i=1}^{K} \prod_{j=0}^{i-1} q_j (1-q_i) = 1$



Characterization of Cox Network – con't

- **The Laplace transform of the service time if i stages are used:**

$$M_{T/i}(s) = \prod_{j=1}^{i} \frac{\mu_i}{s+\mu_i}$$

- **The Laplace transform for the service time in K-stages network:**

$$M(s) = q_0(1-q_1)\frac{\mu_1}{s+\mu_1} + q_0 q_1 (1-q_2)\frac{\mu_1}{s+\mu_1}\frac{\mu_2}{s+\mu_2}$$
$$+ q_0 q_1 q_2 (1-q_3)\frac{\mu_1}{s+\mu_1}\frac{\mu_2}{s+\mu_2}\frac{\mu_3}{s+\mu_3} + \cdots + q_0 q_1 q_2 \cdots q_{K-1}(1-q_K)\frac{\mu_1}{s+\mu_1}\frac{\mu_2}{s+\mu_2}\cdots\frac{\mu_K}{s+\mu_K}$$

$$M(s) = \sum_{i=1}^{K}\prod_{j=0}^{i-1} q_j (1-q_i) \prod_{k=1}^{i} \frac{\mu_k}{s+\mu_k}$$

Characterization of Cox Network – con't

- **M(s) given by**

$$M(s)=\sum_{i=1}^{K}\prod_{j=0}^{i-1}q_j(1-q_i)\prod_{k=1}^{i}\frac{\mu_k}{s+\mu_k}$$

is known as the Coxian distribution

- **For many service time distributions, that can be represented by a rational function of s, they can be put in the form of M(s)**
  - **Therefore, the method of stages provides a method to solve for generalized service time distribution**
  - **However, solving for the coefficients of μk/(s+μk) is not trivial.**

Characterization of Cox Network – con't

- **You can show (refer to textbook), the mean is given by**

$$E[T]=\sum_{i=1}^{K}\prod_{j=0}^{i-1}q_j(1-q_i)\sum_{k=1}^{i}\frac{1}{\mu_k}$$

$$=\sum_{i=1}^{K}\frac{\prod_{j=0}^{i-1}q_j}{\mu_i}$$

Characterization of Cox Network – con't

- Note that for $q_i = 1$, and $\mu_i = \mu$ for all i, then the expression for M(s) reduces to

$$M(s) = \left(\frac{\mu}{s+\mu}\right)^K$$

which the K-stage Erlang-distribution previously discussed on slide 56

- The expected delay in this case is given by

$$E[T] = \frac{K}{\mu}$$

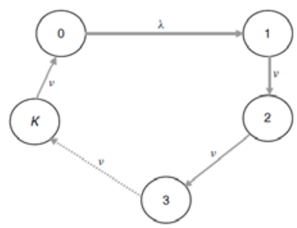
Insensitivity Property of Erlang B

- Example 3.10 – Consider a single server system with K identical stages. Assume a pure blocking system in which there is no queueing. I.e. an arriving message that finds the server busy is lost, otherwise it enters the first stage. Assume Poisson arrivals with rate λ message per second.

Insensitivity Property of Erlang B – cont'd

Solution:

- **The state of the system is the stage where the message is being served**
- **The state transition flow diagram is as shown**
- **The equilibrium equations are given by**
  - $P0\lambda = \nu PK$,
  - $P1\, \nu = \lambda P0$
  - $P2\, \nu = \nu\, P1 \ldots$
  - $PK\, \nu = \nu\, PK\text{-}1$
- **Since $\Sigma\, Pi = 1$ ➔**
  - $P0 = 1/[K(\lambda/\nu) + 1)$, and
  - $Pn = (\lambda/\nu)/[K(\lambda/\nu) + 1]$, n = 1, 2, …, K



Insensitivity Property of Erlang B – cont'd

Solution:

- **The blocking probability is equal to P1 + P2 + … + PK, i.e.**

$$P_B = \sum_{i=1}^{K} P_i = \frac{K\,\lambda/\nu}{1 + K(\lambda/\nu)}$$

- **Note that K/ν is the mean service time.**

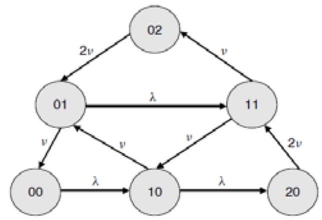
$$P_B = \frac{(\text{arrival rate})\times(\text{average service time})}{1+(\text{arrival rate})\times(\text{average service time})}$$

- **Compare this result to the M/M/S/S case where S = 1 ➔ same form**
  - **Here 1/ν is the mean service time**

Insensitivity Property of Erlang B – cont'd

Example 3.11:

- **M/E(2)/2/2 – system with two servers, each with two identical stages**
- **No storage place**



- **The equilibrium equations**
- **Therefore, the blocking probability, PB is given by p11+p02+p20. Hence**

$$P_B = \frac{2\left(\lambda/\nu\right)^2}{1+2\left(\lambda/\nu\right)+2\left(\lambda/\nu\right)^2}$$

- **Here mean service time = 2/ν**

$$\lambda P_{00} = \nu P_{01}$$
$$(\lambda+\nu)P_{01} = 2\nu P_{02} + \nu P_{01}$$
$$2\nu P_{02} = \nu P_{11}$$
$$(\lambda+\nu)P_{10} = \lambda P_{00} + \nu P_{11}$$
$$2\nu P_{20} = \lambda P_{10}$$
$$2\nu P_{11} = \lambda P_{01} + 2\nu P_{20}$$

Example: M/G/N/N

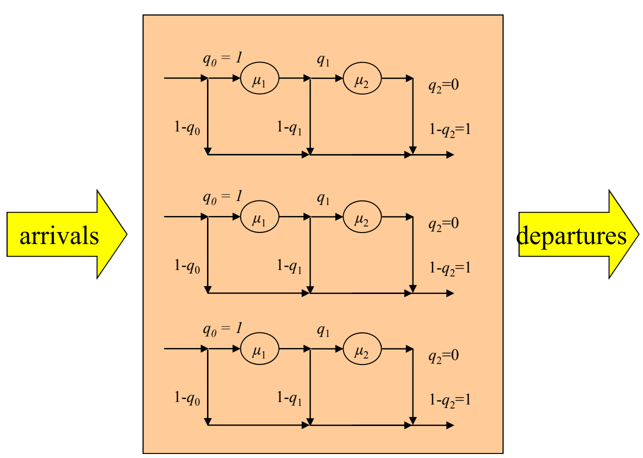
- **Consider a queueing system where**
  - **Arrivals are Poisson with rate λ**
  - **N servers and no waiting room**
  - **Each server is a Coxian server with K stages**
- **Objective: compute blocking probability? And show that it depends only on the mean service rate and the mean arrival rate (i.e. no dependence on the probability distribution of the service time – the *insensitivity* property of the Erlang-B formula)**

Example: M/G/N/N – cont'd

- **System state: K-dimensional vector**
  - **i.e. state = (k_1, k_2, …, k_K) – where k_i; i=1,2, …, K is the number of customers in stage I**
  - **Obviously, sum of k_is should be less or equal to N. Note it is equal to N if all servers are busy – remember too that only one customer can be in any server!!**

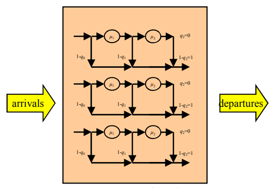
Example: M/G/N/N – cont'd

- **Consider a case where N = 3 and K = 2.**

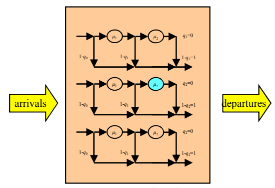


Example: M/G/N/N – cont’d

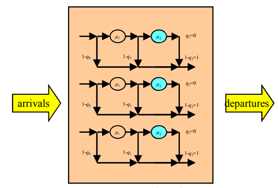
- **System States: examples**



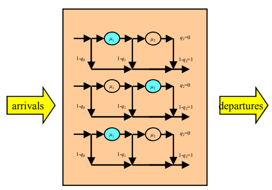
State = (0,0)



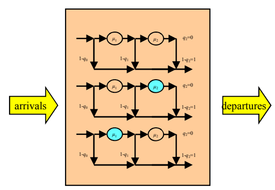
State = (0,1)



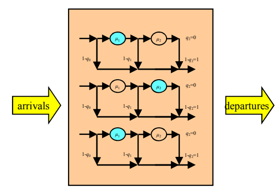
State = (0,3) – system full



State = (2,1) – system full



State = (1,1)



State = (2,1) – system full

Example: M/G/N/N – cont’d

- **Exercise: For the K= 2, N = 3 case explained before**
  - **A) draw the state transition diagram**
  - **B) show that the state equilibrium equations (3.76 and 3.77) are satisfied**
  - **C) derive the detailed balance equation 3.78**
- **Deliver a soft copy in power point of the detailed solution**

Example: M/G/N/N – Blocking Probability

- **Blocking probability is equal to the probability of system being in states where the sum of k_is is equal to N. i.e.**

$$P_B = \Pr ob\left(\sum_{i=1}^{K} k_i = N\right)$$

- **The textbook shows that the blocking probability is given by**

$$P_B = \frac{P(0)}{N!}\left[\lambda \sum_{i=1}^{K} \frac{\prod_{j=0}^{i-1} q_j}{\mu_i}\right]^N$$

where P(0) is a constant term found through the normalization equation

- **Refer for textbook for derivation details.**