Lecture 7

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

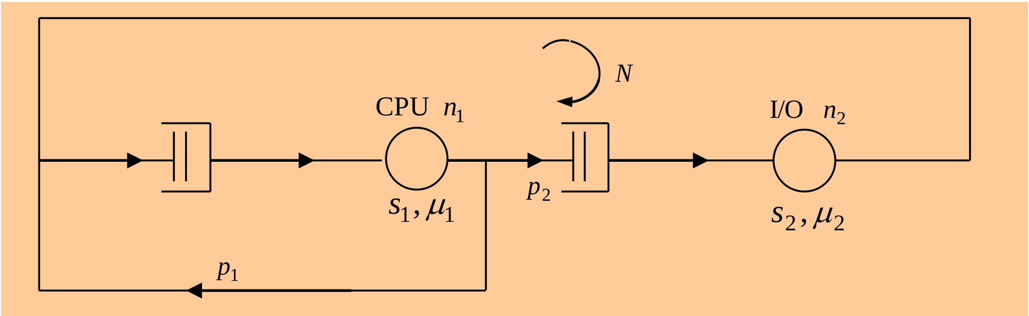
EE 336
Stochastic Models for the Analysis of Computer Systems and Communication Networks

Overview

- Summary of Lecture 6
- Single-class Closed Queuing Networks
- Computational Algorithms
  - Convolution Algorithm
  - Mean Value Analysis
- Applications
  - Computer Systems
  - Flow Control in Communication Networks

Single-class Closed Queuing Networks -1

- Consider a two node network



$n_1 + n_2 = N$

- The visit ratio equations are

$$\left.\begin{aligned} v_1 &= p_1 v_1 + v_2 \\ v_2 &= p_2 v_1 \end{aligned}\right\} \text{ have infinite \# of solutions}$$

v_i ~ Relative Throughput

- Choices

1) $v_1 = \dfrac{1}{p_1} \quad \Rightarrow v_2 = \dfrac{p_2}{p_1}$ — ***visit interpretation***

2) $v_1 = \dfrac{\mu_1}{s_1} \quad \Rightarrow v_2 = \dfrac{\mu_1 p_2}{s_2}$ — all utilizations will be scaled by CPU utilization (i.e., node 1)

3) $v_1 = 1 \quad \Rightarrow v_2 = p_2$ — CPU is the reference node

4) $v_1 + v_2 = 1$ — Probability interpretation

Single-class Closed Queuing Networks -2

- The probability distributions has *product form*

$$p(n_1, n_2) = \frac{1}{G(N)} Y_1(n_1) Y_2(n_2)$$

where $Y_i(n_i) = \left(\frac{v_i s_i}{\mu_i}\right)^{n_i} \quad i = 1, 2$

➲ $G(N)$ is the convolution of $Y_1(N)$ and $Y_2(N)$

$$\boxed{G(N) = \sum_{n_1=0}^{N} Y_1(n_1) Y_2(N - n_1) = Y_1(N) * Y_2(N)}$$

➲ Throughput: $X(N) = \frac{G(N-1)}{G(N)}$

- Nodal throughput: $X_i(n) = v_i X(n) \quad \Rightarrow \frac{X_i(n)}{X_j(n)} = \frac{v_i}{v_j}$

relative throughput

- Mean value analysis (MVA) equation

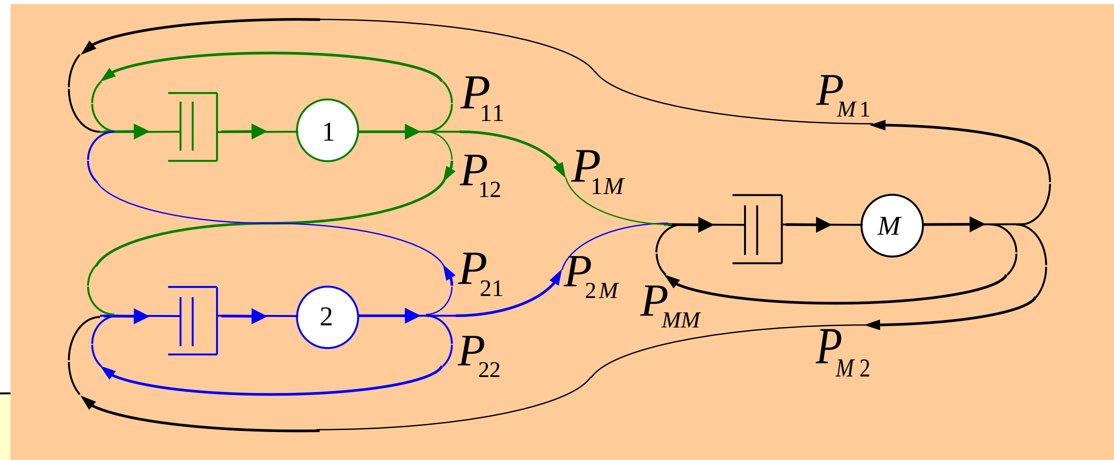
$$\boxed{p_i(n/N) = \frac{v_i X(N) s_i}{\mu_i} p_i(n-1/N-1) \qquad i = 1, 2;\ n = 1, 2, \ldots\ldots, N}$$

$p_i(n/N) \Rightarrow$ marginal probability that there are *n* customers at node *i* given *N* customers in the network

➲ Response time equation for node *i*: $\boxed{R_i(N) = \frac{v_i s_i}{\mu_i}\left[1 + Q_i(N-1)\right]}$

❖ The results are valid for general networks as well with minor changes for various node types

General Structure of Closed Network - 1



- Know
  - The number of customers in the network is a constant, N
  - Since the N customers are distributed among M nodes,

$$n_1 + n_2 + \ldots\ldots + n_M = N$$

  - The service demand at node i is exponentially distributed with mean s_i
  - The node types can be of one of the four types

$$\mu_i(n) = \begin{cases} \mu_i & \text{Single-server} \\ n\mu_i & \text{Infinite-server} \\ \min(n, m_i)\mu_i & \text{Multi-server} \\ \{\mu_i(1), \ldots\ldots, \mu_i(m_i)\} & \text{State-dependent node} \end{cases}$$

General Structure of Closed Network - 2

- The routing of a customer is governed by a Discrete-time Markov chain
- Any work conserving queuing discipline at each node

  $\Rightarrow$ when a customer is present, the server will not turn him away or become idle

- Want to determine

  $Q_i =$ Mean number of customers at node i (queue length)

  $R_i =$ Response time

  $U_i =$ Utilization of node i

  $X(n) =$ Network (system) throughput

  $X_i(n) =$ Nodal throughput at node i

  $p_i(n/N) =$ Marginal prob. that there are n customers at node i

General Structure of Closed Network - 3

- Method
  - Visits computed from

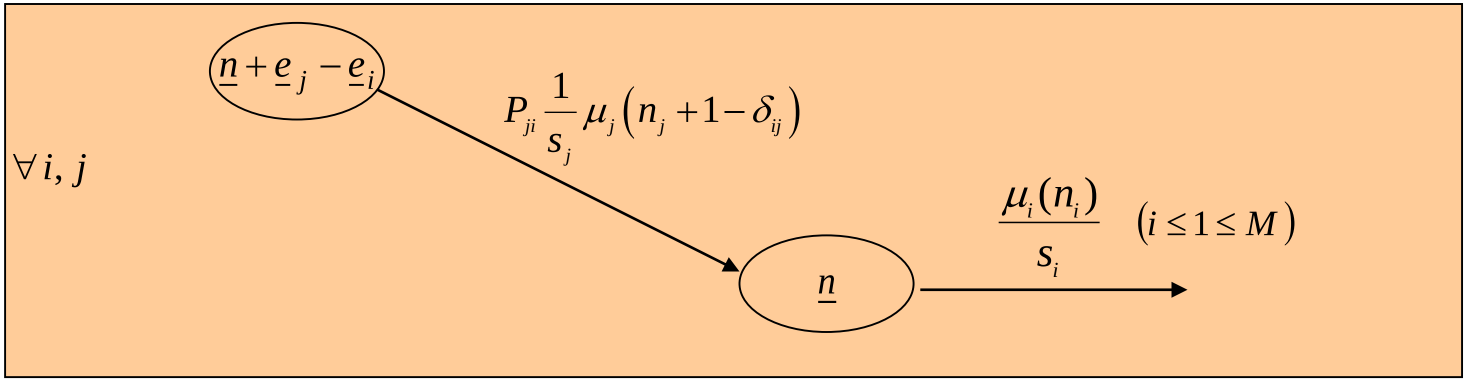
$$v_i = \sum_{j=1}^{M} P_{ji} v_j \quad \text{or} \quad 1 = \sum_{j=1}^{M} P_{ji} \frac{v_j}{v_i} \qquad \forall\, i = 1, 2, \ldots\ldots, M \qquad \cdots \quad \cdots \quad \cdots (1)$$

$$\text{state } (n_1, n_2, \ldots\ldots, n_M) \ni n_1 + n_2 + \ldots\ldots + n_M = N$$

➲ Number of possible states: Distribution of *N* objects among *M* *Nodes*

$$\binom{N+M-1}{M-1}$$

Homework problem



Global and Local Balance Equations

- Global balance equations

$$\sum_{i=1}^{M}\sum_{j=1}^{M} P_{ji}\frac{\mu_j(n_j+1-\delta_{ij})}{s_j}p(\underline{n}+\underline{e}_j-\underline{e}_i)=\sum_{i=1}^{M}\frac{\mu_i(n_i)}{s_i}p(\underline{n})$$

$$\text{(or)}\quad \sum_{i=1}^{M}\sum_{j=1}^{M} P_{ji}\left[\frac{\mu_j(n_j+1-\delta_{ij})}{s_j}p(\underline{n}+\underline{e}_j-\underline{e}_i)-\frac{\mu_i(n_i)v_j}{s_i v_i}p(\underline{n})\right]=0$$

use (1) here

- Local balance equations

$$\frac{\mu_j(n_j+1-\delta_{ij})}{s_j v_j}p(\underline{n}+\underline{e}_j-\underline{e}_i)=\frac{\mu_i(n_i)}{s_i v_i}p(\underline{n})$$

$$\Rightarrow p(\underline{n})=\frac{1}{G_M(N)}\prod_{i=1}^{M}Y_i(n_i)\qquad \Rightarrow Y_i(n_i)=\frac{(v_i s_i)^{n_i}}{\prod_{k=1}^{n_i}\mu_i(k)}$$

$G_M(N)\rightarrow$ Normalization constant

Specialization to Each Node Type

- Special Cases
  - Single-server nodes: $$Y_i(n_i) = \left(\frac{v_i s_i}{\mu_i}\right)^{n_i}$$
  - Infinite-server nodes: $$Y_i(n_i) = \left(\frac{v_i s_i}{\mu_i}\right)^{n_i} \frac{1}{n_i!}$$
  - Multi-server case: $$Y_i(n_i) = \begin{cases} \left(\frac{v_i s_i}{\mu_i}\right)^{n_i} \frac{1}{n_i!}; & n_i \le m_i \\ \left(\frac{v_i s_i}{\mu_i}\right)^{m_i} \frac{1}{m_i!} \left(\frac{v_i s_i}{m_i \mu_i}\right)^{n_i - m_i}; & n_i > m_i \end{cases}$$
  - State-dependent node case: $$Y_i(n_i) = \begin{cases} \frac{(v_i s_i)^{n_i}}{\prod_{k=1}^{n_i} \mu_i(k)}; & n_i \le m_i \\ \frac{(v_i s_i)^{n_i}}{\prod_{k=1}^{n_i} \mu_i(k)} \frac{1}{\left[\mu_i(m_i)\right]^{n_i - m_i}}; & n_i > m_i \end{cases}$$

Properties of Normalization Constant -1

- Properties

$$G_M(N)=\sum_{n_1=0}^{N}\sum_{n_2=0}^{N-n_1}\sum_{n_3=0}^{N-n_1-n_2}\cdots\cdots\sum_{n_{M-1}=0}^{N-\sum_{i=1}^{M-2}n_i}Y_1(n_1)Y_2(n_2)\ldots\ldots Y_{M-1}(n_{M-1})Y_M\left(N-\sum_{i=1}^{M-1}n_i\right)$$

$$=Y_1(N)*Y_2(N)*\ldots\ldots*Y_M(N)$$

Convolution of Y_i's

In z-domain

$$G_M(z)=\prod_{i=1}^{M}Y_i(z)=G_{M-1}(z)Y_M(z)$$

If we let

$$G_l(z)=\prod_{i=1}^{l}Y_i(z)$$

$$G_l(z)=G_{l-1}(z)Y_l(z),\quad G_0(z)=1,\quad G_0(n)=\delta_{n0}=\begin{cases}1 & n=0\\ 0 & else\end{cases}$$

$$\boxed{G_l(n)=\sum_{k=0}^{n}G_{l-1}(n-k)Y_l(k)}$$

Basis of convolution algorithm
$[G_l(0)=1, G_o(0)=1]$

In particular,

$$G_M(n)=G_{M-\{i\}}(n)*Y_i(n)\qquad \forall i=1,2,\ldots\ldots,M$$

Properties of Normalization Constant -2

- Special Cases

1. Single-server case

$$G_l(n)=\sum_{k=0}^{n}G_{l-1}(n-k)\left(\frac{v_l s_l}{\mu_l}\right)^k \Rightarrow G_l(z)=G_{l-1}(z)Y_l(z); \qquad Y_l(z)=\left(1-\frac{v_l s_l}{\mu_l}z\right)^{-1}$$

$$\Rightarrow G_l(z)-\frac{v_l s_l}{\mu_l}z\,G_l(z)=G_{l-1}(z)$$

(or) $$\boxed{G_l(n)=G_{l-1}(n)+\frac{v_l s_l}{\mu_l}G_l(n-1)}$$

2. Infinite-server case

$$Y_i(z)=\sum_{n_i=0}^{\infty}\left(\frac{v_i s_i}{\mu_i}\right)^{n_i}\frac{z^{n_i}}{n_i!}=e^{\left(\frac{v_i s_i}{\mu_i}\right)z}$$

$$G_{l+M_I-1}(n)=\sum_{k=0}^{n}G_{l-1}(n-k)G_{M_I}(k)$$

Suppose, had M_I infinite server nodes. Then

$$G_{M_I}(z)=\exp\left(\left[\sum_{i=1}^{M_I}\frac{v_i s_i}{\mu_i}\right]z\right)$$

Can combine infinite server nodes into a ***'single equivalent node'***

$$G_{M_I}(n)=\frac{1}{n_i!}\left(\sum_{i=1}^{M_I}\frac{v_i s_i}{\mu_i}\right)^{n_i}$$

(or) $$\boxed{G_{M_I}(n)=\frac{1}{n}\left(\sum_{i=1}^{M_I}\frac{v_i s_i}{\mu_i}\right)G_{M_I}(n-1); \qquad G_{M_I}(0)=1}$$

Properties of Normalization Constant -3

3. **Multi-server case**

$$G_l(n) = \sum_{k=0}^{n} G_{l-1}(n-k) Y_l(k)$$

$$\boxed{Y_l(k) = \begin{cases} \left(\dfrac{v_l s_l}{k\mu_l}\right) Y_l(k-1); Y_l(0) = 1; & k \le m_l \\ \left(\dfrac{v_l s_l}{m_l \mu_l}\right) Y_l(k-1); & k > m_l \end{cases}}$$

2. State-dependent server case

$$G_l(n) = \sum_{k=0}^{n} G_{l-1}(n-k) Y_l(k)$$

$$\boxed{Y_l(k) = \begin{cases} \left(\dfrac{v_l s_l}{\mu_l(k)}\right) Y_l(k-1); Y_l(0) = 1; & k \le m_l \\ \left(\dfrac{v_l s_l}{\mu_l(m_l)}\right) Y_l(k-1); & k > m_l \end{cases}}$$

Can we get a better Recursion like the way We did for M|M|m and S-D queues?....YES

New Recursion for Multi-server and SD Nodes -1

$$G_l(n) = \sum_{k=0}^{n} G_{l-1}(n-k) Y_l(k) \Rightarrow G_l(z) = Y_l(z) G_{l-1}(z) = [\sum_{k=0}^{\infty} Y_l(k) z^k] G_{l-1}(z)$$

$$G_l(z).\left(1 - \frac{v_l s_l}{\mu_l(m_l)} z\right) = \sum_{k=0}^{\infty}\left[Y_l(k) z^k - Y_l(k) \frac{v_l s_l}{\mu_l(m_l)} z^{k+1} \right] G_{l-1}(z)$$

$$= \left[\sum_{k=0}^{m_l-1} Y_l(k) z^k - \sum_{k=0}^{m_l-2} Y_l(k) \frac{v_l s_l}{\mu_l(m_l)} z^{k+1} \right] G_{l-1}(z)$$

$$= G_{l-1}(z) + \sum_{k=1}^{m_l-1} \left[Y_l(k) - Y_l(k-1) \frac{v_l s_l}{\mu_l(m_l)} \right] z^k G_{l-1}(z)$$

$$= G_{l-1}(z) + \sum_{k-1}^{m_l-1} \frac{(v_l s_l)^k}{\prod_{q=1}^{k-1} \mu_l(q)} \left[\frac{1}{\mu_l(k)} - \frac{1}{\mu_l(m_i)} \right] z^k G_{l-1}(z)$$

$$\boxed{G_l(n) = G_{l-1}(n) + \frac{v_l s_l}{\mu_l(m_l)} G_l(n-1) + \underbrace{\sum_{k=1}^{\min(m_l-1,n)} \frac{(v_l s_l)^k}{\prod_{q=1}^{k-1} \mu_l(q)} \left[\frac{1}{\mu_l(k)} - \frac{1}{\mu_l(m_i)} \right] G_{l-1}(n-k)}_{\sum_{k=1}^{\min(m_i-1,n)} Y_l(k) \left[1 - \frac{\mu_l(k)}{\mu_l(m_i)} \right] G_{l-1}(n-k)}}$$

New Recursion for Multi-server and SD Nodes -2

1. Single-server case ($m_l = 1$)

$$G_l(n) = G_{l-1}(n) + \frac{v_l s_l}{\mu_l(m_l)} G_l(n-1)$$

2. Infinite-server case ($m_l = \infty$)

$$G_l(n) = G_{l-1}(n) + \sum_{k=1}^{n} Y_l(k) G_{l-1}(n-k) = \sum_{k=0}^{n} Y_l(k) G_{l-1}(n-k); Y_l(k) = \frac{1}{k}\frac{v_l s_l}{\mu_l} Y_l(k-1);\; Y_l(0) = 1$$

3. Multi-server case

$$G_l(n) = G_{l-1}(n) + \frac{v_l s_l}{\mu_l(m_l)} G_l(n-1) + \sum_{k=1}^{\min(m_l-1,n)} Y_l(k)\left[1 - \frac{k}{m_l}\right] G_{l-1}(n-k)$$

$$Y_l(k) = \begin{cases} \left(\dfrac{v_l s_l}{k\mu_l}\right) Y_l(k-1); Y_l(0) = 1; & k \le m_l \\ \left(\dfrac{v_l s_l}{m_l \mu_l}\right) Y_l(k-1); & k > m_l \end{cases}$$

Computation of Normalization Constant

- Algorithm

$$G_0(0) = 1$$

$$G_l(0) = 1 \quad \forall\, l = 1, 2, \ldots\ldots, M$$

Do $n = 1, 2, \ldots\ldots, N$

  Do $l = 1, 2, \ldots\ldots, M$

    Evaluate Convolution Sum

  End Do

End Do

Key: It turns out that all performance measures Q_i, U_i, R_i, etc. are functions of $G(N)$

Performance Measures from *G(N)* -1

1. Marginal Probabilities at each node i

k at node $i \Rightarrow N$-k at all other nodes

Know

$$G_M(N) = \sum_{k=0}^{N} G_{M-\{i\}}(N-k)Y_i(k)$$

$$1 = \sum_{k=0}^{N} \frac{G_{M-\{i\}}(N-k)Y_i(k)}{G_M(N)} = \sum_{k=0}^{N} p_i(k/N)$$

$$\boxed{\Rightarrow p_i(k/N) = \frac{G_{M-\{i\}}(N-k)Y_i(k)}{G_M(N)}}$$

For single-server nodes the result simplifies to:

$$\boxed{p_i(k/N) = \left[\frac{G_M(N-k) - \frac{v_i s_i}{\mu_i} G_M(N-k-1)}{G_M(N)}\right]\left(\frac{v_i s_i}{\mu_i}\right)^k}$$

Performance Measures from $G(N)$ -2

2. Throughput of node i

$$X_i(N) = \sum_{k=1}^{N} p_i(k/N)\frac{\mu_i(k)}{s_i}$$

$$= \sum_{k=1}^{N} \frac{G_{M-\{i\}}(N-k)Y_i(k)}{G_M(N)}\frac{\mu_i(k)}{s_i}$$

$$= v_i \sum_{k=1}^{N} \frac{G_{M-\{i\}}(N-k)Y_i(k-1)}{G_M(N)} = v_i \frac{\sum_{n=0}^{N-1} G_{M-\{i\}}(N-1-n)Y_i(n)}{G_M(N)}$$

$$= v_i \frac{G_M(N-1)}{G_M(N)}$$

$$= v_i X(N)$$

Network Throughput

Performance Measures from $G(N)$ -3

3. Queue length

$$Q_i(N) = \sum_{k=1}^{N} k\, p_i(k/N)$$

For single-server nodes

$$Q_i(N) = \sum_{k=1}^{N} k\, p_i(k/N) = \sum_{k=1}^{N} k \left[\frac{G_M(N-k)Y_i(k) - G_M(N-k-1)Y_i(k+1)}{G_M(N)} \right]$$

$$= \sum_{k=1}^{N} \frac{G_M(N-k)Y_i(k)}{G_M(N)}$$

$$= \sum_{k=1}^{N} \left[\left(\prod_{l=1}^{k} \frac{G_M(N-l)}{G_M(N-l+1)} \right) \frac{v_i s_i}{\mu_i} \right] = \sum_{k=1}^{N} \prod_{l=0}^{k-1} \left(X(N-l) \frac{v_i s_i}{\mu_i} \right)$$

$$= \sum_{k=1}^{N} \left[\prod_{l=0}^{k-1} U_i(N-l) \right] = \sum_{k=1}^{N} \prod_{l=k}^{N} U_i(l)$$

As $N \to \infty$ $\quad Q_i(N) \to \dfrac{U_i}{1-U_i}$ Like in M/M/1 queue

4. Response time

$$R_i(N) = \frac{Q_i(n)}{X(n)}$$ Over all visits

Mean Value Analysis Recursion

Marginal probabilities, $p_i(k/N) = \dfrac{G_{M-\{i\}}(N-k)Y_i(k)}{G_M(N)}$

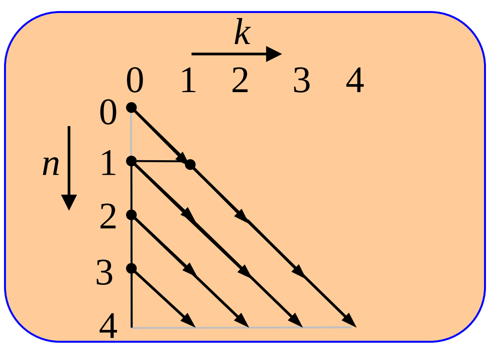
$$p_i(k-1/N-1) = \frac{G_{M-\{i\}}(N-k)Y_i(k-1)}{G_M(N-1)}$$

$$\Rightarrow p_i(k/N) = \frac{Y_i(k)}{Y_i(k-1)}\frac{G_M(N-1)}{G_M(N)}p_i(k-1/N-1)$$

$$\boxed{\Rightarrow p_i(k/N) = \frac{X(N)v_i s_i}{\mu_i(k)}p_i(k-1/N-1) \qquad k=0,1,2,\ldots\ldots,N}$$

- Probability distribution at node *i* with *N* customers is related to the probability distribution at the same node with *(N-1)* customers

- **More Generally,**



$$p_i(k/n) = \frac{X(n)v_i s_i}{\mu_i(k)}p_i(k-1/n-1); \qquad k=0,1,2,\ldots\ldots,n; \quad n=0,1,2,\ldots\ldots,N; \quad p_i(0/0)=1$$

$$\boxed{p_i(0/n) = 1 - \sum_{k=1}^{n} p_i(k/n)} \Rightarrow$$

Often results in negative probabilities

MVA Equations for Response Time -1

$$Q_i(n) = \sum_{k=1}^{n} k\, p_i(k/n)$$

$$= \sum_{k=1}^{n} \frac{k v_i s_i}{\mu_i(k)} X(n)\, p_i(k-1/n-1)$$

$$= v_i s_i X(n) \left[\sum_{k=1}^{m_i-1} \frac{k}{\mu_i(k)} p_i(k-1/n-1) + \sum_{k=m_i}^{n} \frac{k}{\mu_i(m_i)} p_i(k-1/n-1) \right]$$

$$= \frac{v_i s_i X(n)}{\mu_i(m_i)} \left[\underbrace{\sum_{k=1}^{m_i-1} k \left[\frac{\mu_i(m_i)}{\mu_i(k)} - 1 \right] p_i(k-1/n-1)}_{\gamma_i(n-1)} + 1 + Q_i(n-1) \right]$$

$$\frac{\mu_i(m_i)}{\mu_i(k)} = \frac{m_i}{k} \text{ for multi-server node}$$

$$\boxed{Q_i(n) = \frac{v_i s_i X(n)}{\mu_i(m_i)} \left[1 + \gamma_i(n-1) + Q_i(n-1) \right]}$$

Note: $Q_i(0) = 0$

(or) $$\boxed{R_i(n) = \frac{v_i s_i}{\mu_i(m_i)} \left[1 + \gamma_i(n-1) + Q_i(n-1) \right]}$$

MVA Equations for Response Time -2

- Special cases
  1. Single-server case

$$R_i(n) = \frac{v_i s_i}{\mu_i}\left[1 + Q_i(n-1)\right]$$

  2. Infinite-server case

$$R_i(n) = \lim_{m_i \to \infty} \frac{v_i s_i}{\mu_i}\left[\sum_{k=1}^{m_i-1}\left(1-\frac{k}{m_i}\right)p_i(k-1/n-1) + \frac{1}{m_i} + \frac{Q_i(n-1)}{m_i}\right]$$

$$= \frac{v_i s_i}{\mu_i}$$

- Expected network response time: $\sum_{i=1}^{M} R_i(n) = R(n)$
- Throughput: $\frac{n}{R(n)} = X(n)$
- Queue length: $Q_i(n) = X(n)R_i(n)$

MVA Equations for Response Time -3

- Probability distribution at state-dependent nodes

$$p_i(k/n) = \frac{v_i s_i X(n)}{\mu_i(k)} p_i(k-1/n-1); \qquad k = 1, 2, \ldots\ldots, n_i; \qquad n_i = \min(m_i - 1, n)$$

$$p_i(\geq m_i/n) = \frac{v_i s_i X(n)}{\mu_i(m_i)} p_i(\geq m_i - 1/n-1)$$

$$p_i(0/n) = 1 - \sum_{k=1}^{m_i - 1} p_i(k/n) - p_i(\geq m_i/n)$$

- Utilization of each node (needs to be computed for final population only)

$$U_i(N) = 0$$

Infinite-server nodes

$$U_i(N) = \frac{X(N) v_i s_i}{\mu_i}$$

Single-server nodes

$$U_i(N) = \frac{X(N) v_i s_i}{\mu_i(m_i)}$$

Multi-server & state-dependent nodes

$$\frac{\text{Throughput}}{\text{max. Service Capacity}}$$

MVA Equations for Response Time -4

- Computation of ***queue length*** and ***utilization at each node***
  - Algorithm

$Q_i(0) = 0$

$p_i(0/0) = 1$

Do $n = 1,2,\ldots\ldots,N$

Do $i = 1,2,\ldots\ldots,M$

Update response time $R_i(n)$

End Do

$$X(n) = n / \sum_{i=1}^{M} R_i(n)$$

Do $i = 1,2,\ldots\ldots,M$

Compute $Q_i(n)$

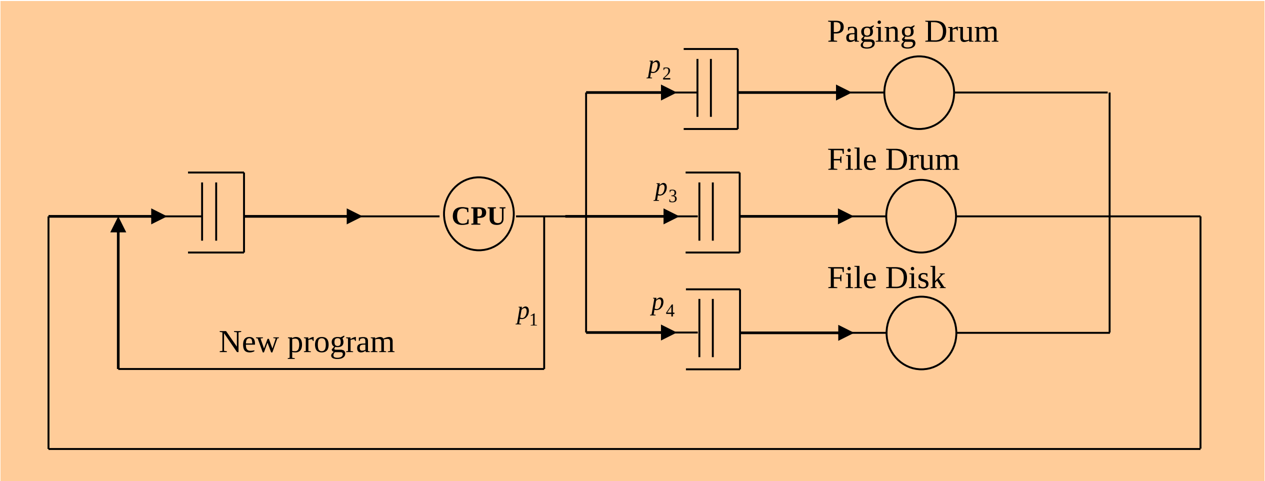
End Do

Compute $p_i(k/n)$ at multi-server & state-dependent nodes

End Do

Compute $U_i(N)$ $i = 1,2,\ldots\ldots,M$

Application Example: *"what if studies"* - 1



Number of visits: $v_1 = \dfrac{1}{p_1}, \quad v_2 = \dfrac{p_2}{p_1}, \quad v_3 = \dfrac{p_3}{p_1}$ & $v_4 = \dfrac{p_4}{p_1}$

$$v_1 = 1 + v_2 + v_3 + v_4$$

- Each program requires 2.1 billion instructions to be executed: $v_1 s_1 = 2.1 \times 10^9$ Ins
- CPU speed is 700 MIPS : $\mu_1 = 0.7 \times 10^9$ Ins/sec
- CPU time per program : $\dfrac{v_1 s_1}{\mu_1} = 3$ sec/job (Over all visits)

Application Example: *"what if studies"* - 2

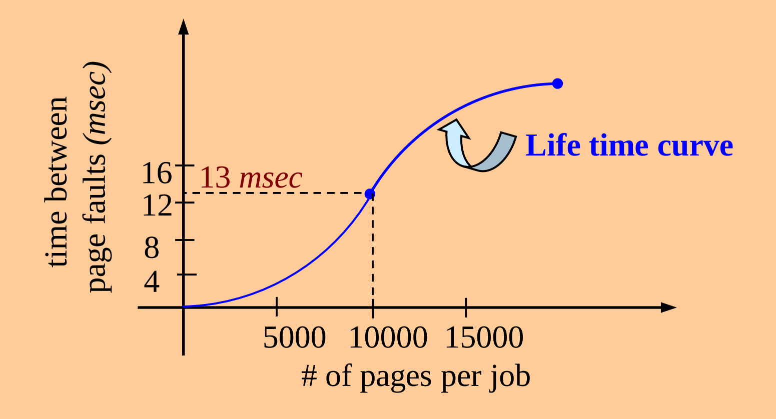
- Given that 150 file drum accesses & 250 file disk accesses : $v_3 = 150, \quad v_4 = 250$
- To get v_2, following information is given
  - Limited main memory of 30000 pages, divided equally among N jobs

    Per job we have *30000/N* pages

    Larger the # of jobs $\Rightarrow$ Smaller memory partition per job

    $\Rightarrow$ More often CPU has to interrupt to get the next segment of the program from the page drum

If $N = 3 \Rightarrow$ Allotted 10000 pages per program



Average time between page faults =13 msec

# of page faults = # of visits to page drum, $v_2 = \frac{3}{13} \times 10^3 \cong 230$ visits

Application Example: *"what if studies"* - 3

- Paging drum has revolution speed of 3600 rpm = 60 rps

  Transfers at 4μs/word (each page size is 1024 words)

  $$\Rightarrow s_2 = 1, \quad \mu_2 = \left(\frac{1}{2 \times 60} + 1024 \times 4 \times 10^{-6} \right)^{-1} = 80 \text{ pages/sec}$$

  $$v_2 = 230$$

  $$v_1 = v_2 + v_3 + v_4 + 1 = 1 + 230 + 150 + 250 = 631$$

- File drum

  $$s_3 = 1, \quad \mu_2 = 80 \text{ pages/sec}, \quad v_3 = 150$$

- File disk: 2400 rpm, t_{seek} =11ms, transfer time = 6μs

  $$s_4 = 1, \quad \mu_4 = \left(\frac{1}{2 \times 40} + t_{seek} + 1024 \times 6 \times 10^{-6} \right)^{-1} = 33.7 \text{ pages/sec}, \qquad v_4 = 250$$

- CPU $\dfrac{v_1 s_1}{\mu_1} = 3 \quad \Rightarrow s_1 = \dfrac{2.1 \times 10^9}{631}$; $v_1 = 630, \quad \mu_1 = 7 \times 10^8$ Ins/sec

	CPU (1)	Paging Drum (2)	File Drum (3)	File Disk (4)
v_i	631	230	150	250
s_i	3.17×10^6	1page	1page	1page
μ_i	7×10^8 Ins/sec	80 pages/sec	80 pages/sec	33.7 pages/sec
$v_i s_i / \mu_i$	3	2.86	1.86	7.5

Bottleneck

Application Example: *"what if studies"* - 4

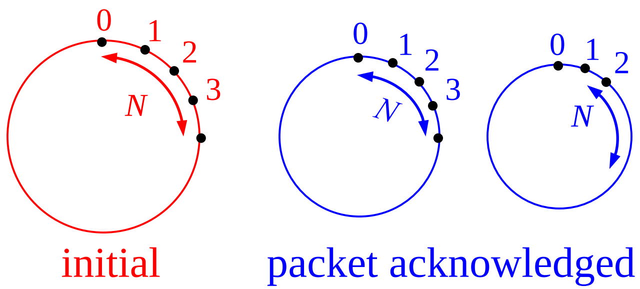
- Using MVA or convolution algorithm, one can explore, for example, the following
  1. Performance measures
  2. What if CPU is replaced by one that is 25% faster?
  3. Replace file drum with one having 4500 rpm
  4. Faster paging drum 4500 rpm
  5. Balance utilization to file and disk drums

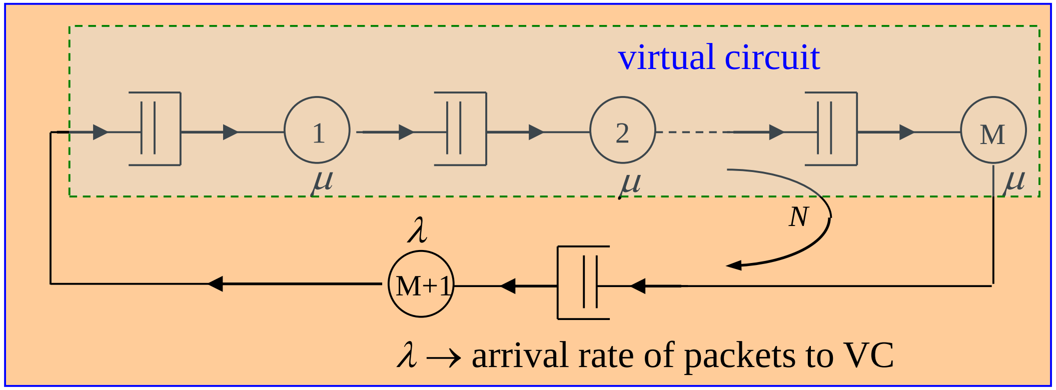
$$\Rightarrow \frac{v_3 s_3}{\mu_3} = \frac{v_4 s_4}{\mu_4} \quad \ni v_3 + v_4 = 400$$

  6. Increase memory size from 30000-40000 pages

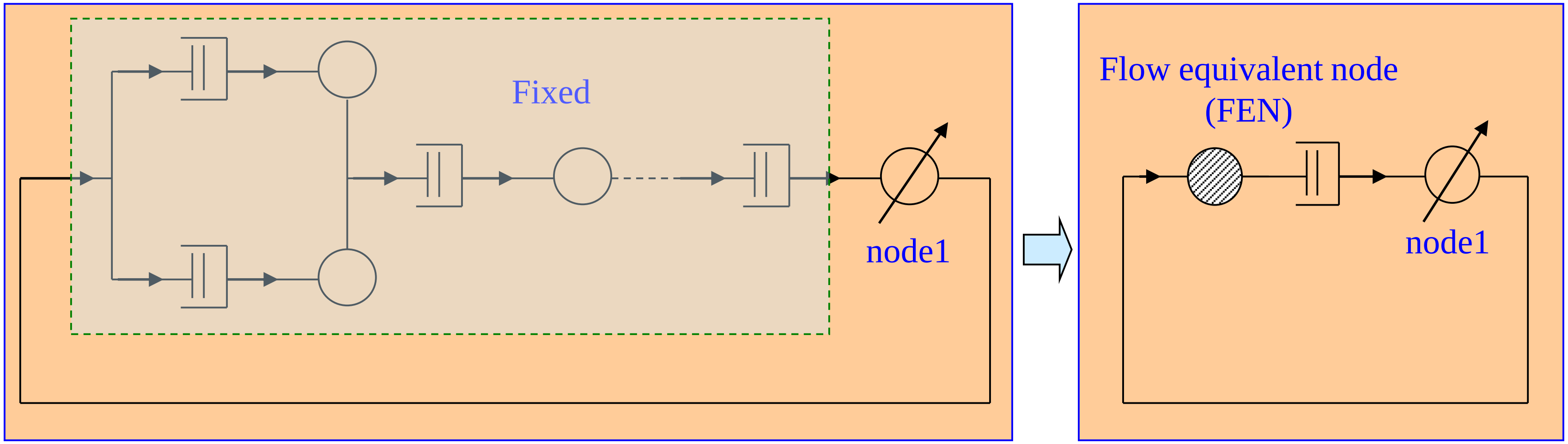
$\Rightarrow$ time per page fault =15ms

Application Example: *"sliding window flow control"* - 1





- Norton's theorem (or) Chandy-Herzog-Woo's theorem



Application Example: *"sliding window flow control"* - 2

- Know

$$p_1(k/N) = \frac{G_{M-\{1\}}(N-k)Y_1(k)}{G_M(N)}$$

$$G_M(N) = \sum_{k=0}^{N} G_{M-\{1\}}(N-k)Y_1(k)$$

- The network is equivalent to a 'two node network', if

$$G_{M-\{1\}}(n) = Y_{eq}(n) = \left(\prod_{k=1}^{n} \mu_{eq}(k)\right)^{-1} \qquad s_{eq} = 1, \quad v_{eq=1}$$

- Know

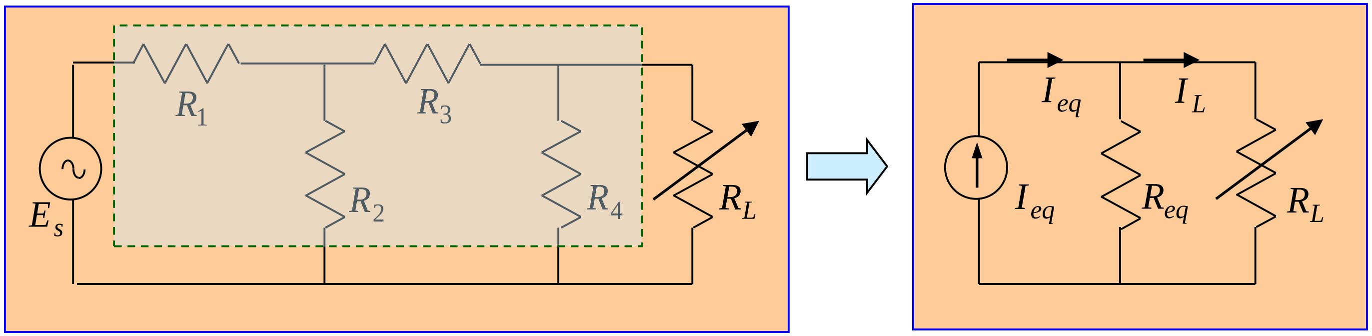
$$X_{eq}(n) = \frac{G_{M-\{1\}}(n-1)}{G_{M-\{1\}}(n)} = \frac{Y_{eq}(n-1)}{Y_{eq}(n)} = \mu_{eq}(n)$$

The service rate of flow eqv. node= Throughput of the boxed network

- $X_{eq}(n)$ can be found by "shorting" node 1
  - This is similar to Norton's theorem of electrical circuits

Application Example: *"sliding window flow control"* - 3

- Norton's theorem: Replace all sources by an equivalent current source and equivalent resistance (or impedance)



  - Similarly, we short node 1 (subsystem of interest)
  - Found throughput of the remaining network for all $n = 1,2,\ldots\ldots,N$
  - Set service rate of flow equivalent node $= \mu_{eq}(n), \quad n = 1,2,\ldots\ldots,N$

- For the sliding window flow control problem:
  - Short node M+1, all nodes have same service rate
  - Know v = 1, s = 1

$$R_i(n) = \frac{1}{\mu}\left[1 + Q_i(n-1)\right]$$

Since the network is balanced,

$$Q_i(n-1) = \frac{n-1}{M} \qquad i = 1,2,\ldots\ldots,M$$

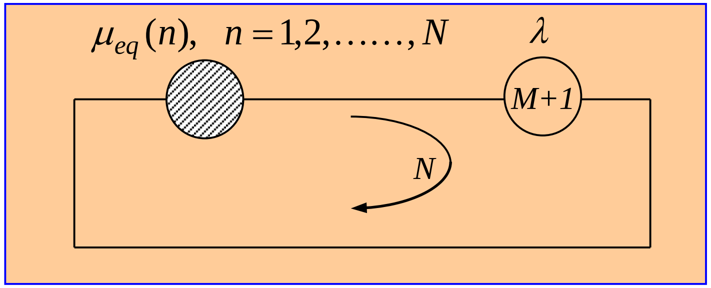
Application Example: *"sliding window flow control"* - 4

$$\Rightarrow R_i(n) = \frac{1}{\mu}\left[\frac{n+M-1}{M}\right]$$

$$R(n) = \sum_{i=1}^{M} R_i(n) = \frac{n+M-1}{\mu}$$

$$X_{eq}(n) = \frac{n\mu}{n+M-1} = \mu_{eq}(n)$$

So, the equivalent network is



- Yes, can solve via MVA!!

Special Cases

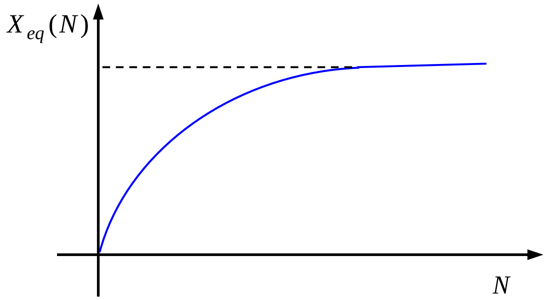
- **Case 1**

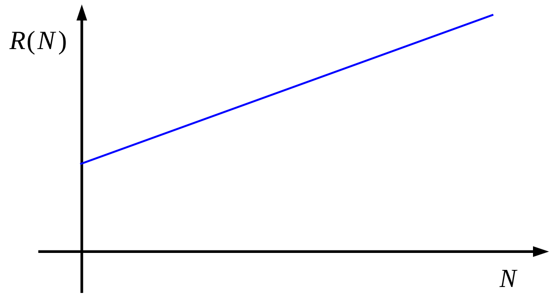
As $\lambda \to \infty$, all N packets will be at the equivalent network

$$\text{Throughput } = X_{eq}(N) = \frac{N\mu}{N+M-1}$$

$$\& \quad R(N) = \left[\frac{N+M-1}{\mu}\right] = \frac{M-1}{\mu - X_{eq}(N)} \; because \; \mu - X_{eq}(N) = \mu \frac{M-1}{N+M-1}$$

Application Example: *"sliding window flow control"* - 5





Case 2:

$$\lambda = \mu \quad \Rightarrow \quad X_{eq}(N) = \frac{N\mu}{(N+M)}, \qquad R_{eq}(N) = \frac{N+M}{\mu}\,\frac{M}{M+1}$$

— What should we choose as N, the window length? $N \approx M$

"Maxwell Demon"

— How to optimally control λ as a function of n?

See Lazar, IEEE T-AC, 1983, pp 1001-1009

"Bang Bang Control"

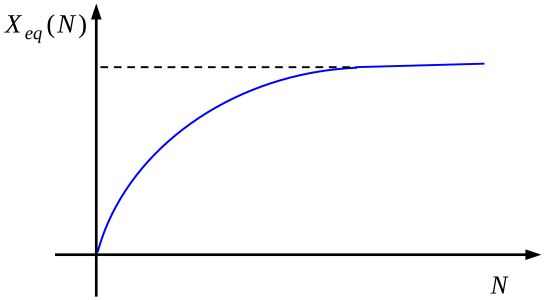
$$R(N) = \frac{M-1}{\mu - X_{eq}} \qquad \text{as } \lambda \to \infty$$

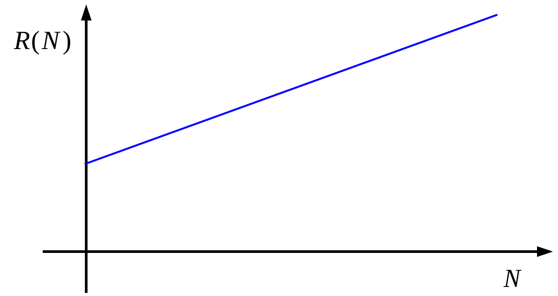
Power

$$\frac{X_{eq}}{R(N)} = \frac{X_{eq}(\mu - X_{eq})}{M-1} \quad \Rightarrow \max \ \text{ at } X_{eq} = \mu/2 \qquad N = M - 1$$

$$\text{at } X_{eq} = 0, \quad R(N) = \frac{M-1}{\mu}$$

Application Example: *"sliding window flow control"* - 6





$$R(N) = \frac{M-1}{\mu - X_{eq}} \qquad \text{as } \lambda \to \infty$$

$$\mu R(N) = \frac{M-1}{1 - \frac{X_{eq}}{\mu}} \Rightarrow Normalized\ Delay = \frac{M-1}{1 - Normalized\ Throughput}$$

$$Find\ X_{eq} \ni \mu R(N) = \frac{d(\mu R(N))}{X_{eq}} X_{eq}$$

$$\Rightarrow X_{eq} = \frac{\mu}{2} \Rightarrow N = M - 1 \,\&\, \mu R(N) = 2(M-1)$$

Summary

- Single-class Closed Queuing Networks
- Computational Algorithms
  - Convolution Algorithm
  - Mean Value Analysis
- Applications
  - Computer Systems
  - Flow Control in Communication Networks