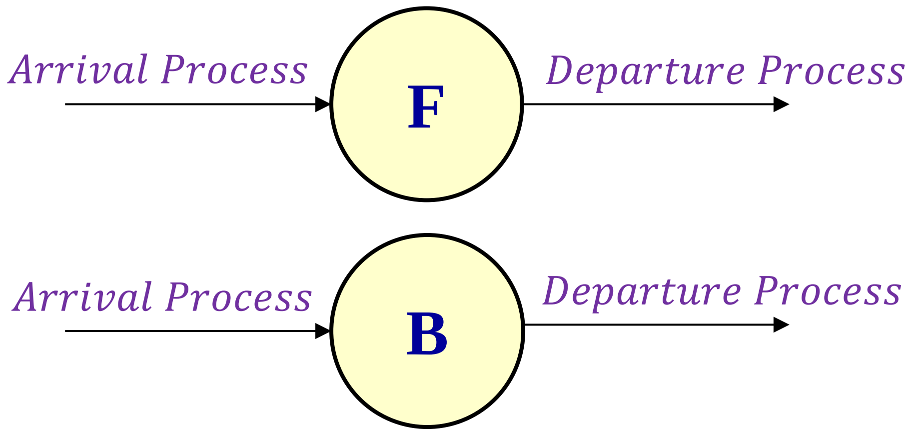
Queuing Networks

Wha Sook Jeon

Mobile Computing & Communications Lab.

Time Reversibility (1)

- Time reversibility
  - Statistical characteristic of forward process is the same as that of backward process
  - The arrival process of the forward process is the arrival process of the backward process, which is the departure process of the forward process $\Rightarrow$ The arrival process of time reversible process has the same statistical characteristic as its own departure process



arrival process $_{\mathrm{F}}$ = arrival process $_{B}$ = departure process $_{F}$ = departure process $_{B}$

Time Reversibility (2)

DTMC

- Forward process
  - Transition probability from state *i* to state *j*: p_{ij}

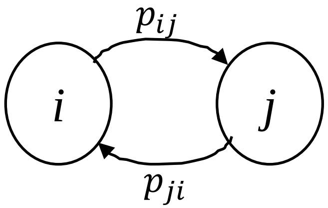
    $$p_{ij} = \Pr\{X_{n+1} = j | X_n = i\}$$

- Backward process
  - Transition probability from state *i* to state *j*: q_{ij}

    $$q_{ij} = \Pr\{X_n = j | X_{n+1} = i\}$$

  - $q_{ij} = \frac{\Pr\{X_n=j, X_{n+1}=i\}}{\Pr\{X_{n+1}=i\}} = \frac{\Pr\{X_{n+1}=i|X_n=j\}\Pr\{X_n=j\}}{\Pr\{X_{n+1}=i\}} = \frac{\pi_j p_{ji}}{\pi_i}$

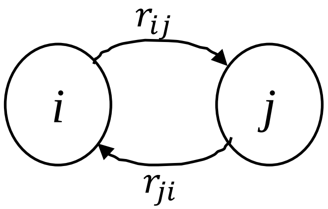
- Necessary and sufficient condition for time reversibility: $q_{ij} = p_{ij}$
  - $q_{ij} = \frac{\pi_j p_{ji}}{\pi_i} = p_{ij} \quad \Rightarrow \pi_i p_{ij} = \pi_j \, p_{ji}$



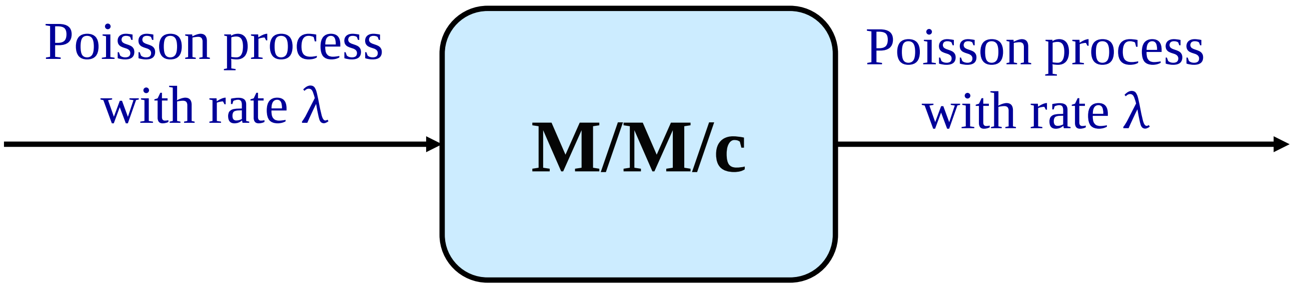
- Time reversible DTMC, $\pi_i p_{ij} = \pi_j p_{ji}$

Time Reversibility (3)

Time reversible CTMC : $\pi_i r_{ij} = \pi_j r_{ji}$



- Birth & death process is time reversible
  - Since M/M/c queuing system is a special case of birth & death process, M/M/c is time reversible
  - Arrival process of M/M/c queuing system is the same as its departure process. Thus, departure process of M/M/c is a Poisson process

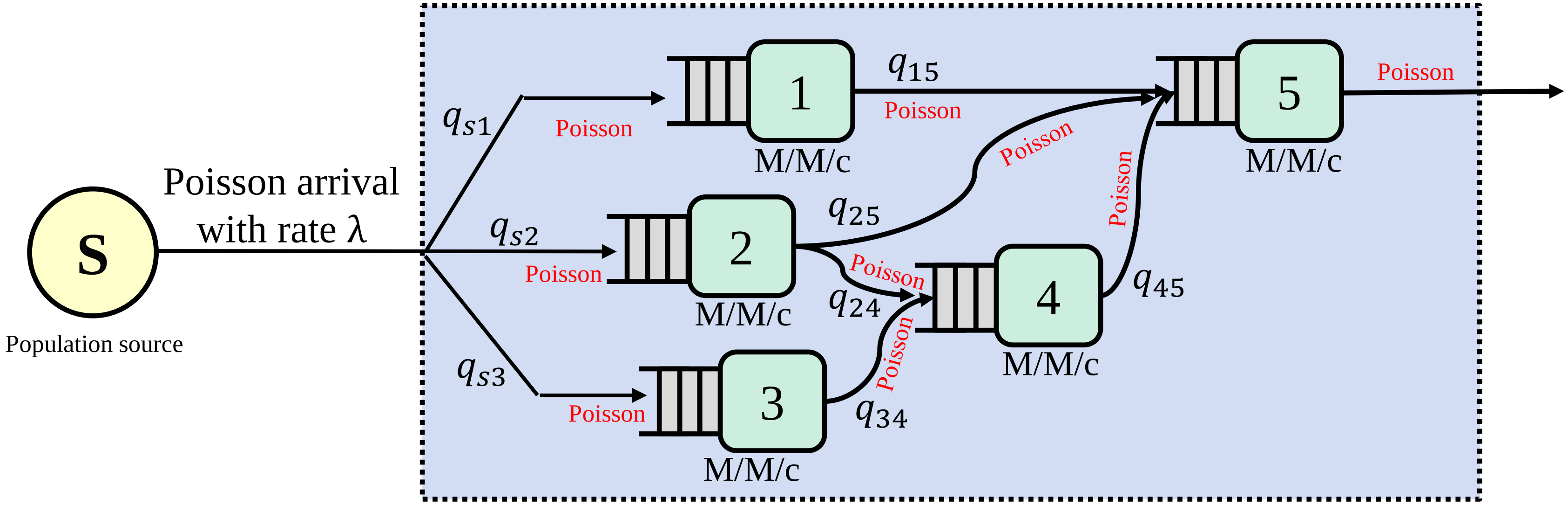


Open Queuing Networks (1)

- Open Queuing networks with product form solution

<Assumption>

- Poisson arrivals from outside source
- All servers have exponentially distributed service time
- A job from device *i* joins device *j* with (routing) probability q_{ij}



- Each device is modeled as M/M/c, being independent of each other.

Open Queuing Networks (2)

- System state: $(n_1, n_2, n_3, n_4, n_5)$
  - n_i: number of jobs in device *i*

- Jackson's decomposition theorem

$$P(n_1, n_2, n_3, n_4, n_5) = P_1(n_1)\, P_2(n_2) P_3(n_3)\, P_4(n_4) P_5(n_5)$$

  - $P(n_1, n_2, n_3, n_4, n_5)$: System state probability
  - $P_i(n_i)$: Probability of n_i jobs in device *i*

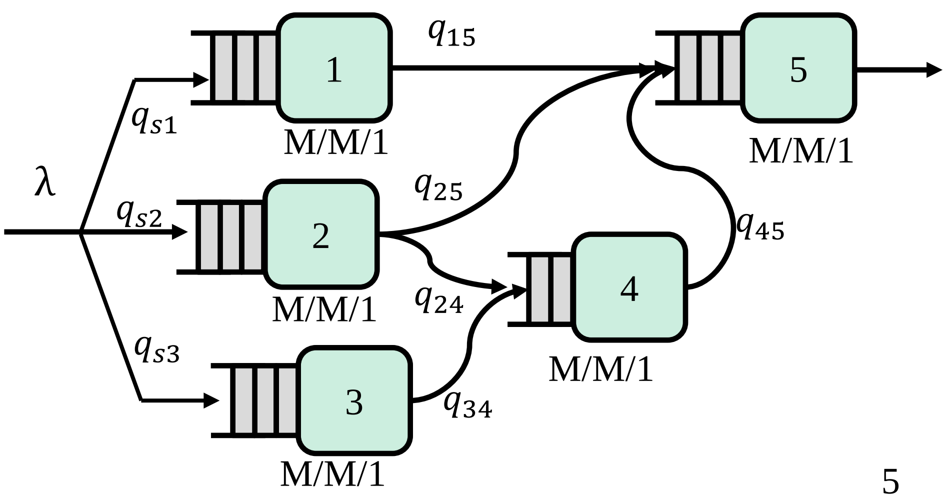
<example>

- When all devices are M/M/1

$P_i(n_i) = {\rho_i}^{n_i}(1 - \rho_i)$

$P(n_1, n_2, n_3, n_4, n_5) = \prod_{i=1}^{5} {\rho_i}^{n_i}(1 - \rho_i)$

- $\rho_i = \dfrac{\lambda_i}{\mu_i}$
- $\lambda_1 = \lambda q_{s1}, \quad \lambda_2 = \lambda q_{s2}, \quad \lambda_3 = \lambda q_{s3},$
- $\lambda_4 = \lambda_3 + \lambda_2 q_{24}, \quad \lambda_5 = \lambda_1 + \lambda_2 q_{25} + \lambda_4$



Open Queuing Networks (3)

- Performance measure

  < Device i >

  – Utilization of device i: $\rho_i = \frac{\lambda_i}{\mu_i}$

  – Mean number of jobs in device i: $\bar{N}_i = \frac{\rho_i}{1-\rho_i}$

  < System >

  – Mean number of jobs: $\bar{N} = \sum_{i=1}^{M} \bar{N}_i$

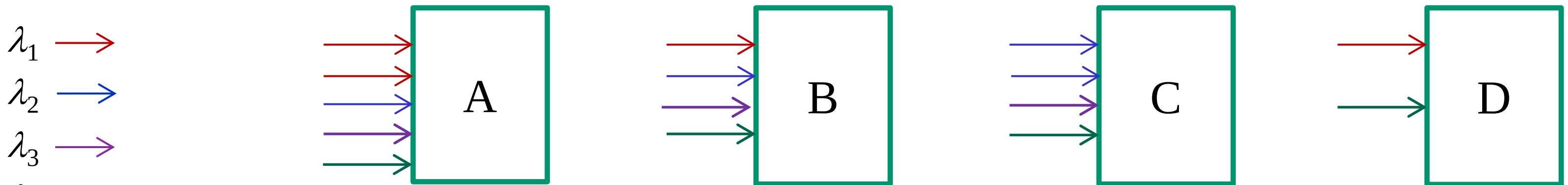
    - M: the number of devices in the network

  – Mean sojourn time of a job in the network: $\bar{T} = \frac{\bar{N}}{\lambda}$

Exercise1: Open Queuing Networks

A machine shop has four machines, A, B, C, and D. The numbers of servers in the machines A, B, C and D are one, one, two, and three, respectively. Service time distributions of the servers in the machines A, B, C and D are exponential at their respective rates μ_A, μ_B, μ_C, and μ_D. The shop gets four types of jobs, numbered 1 through 4, where each type requires service on machines in a particular sequence; type 1: ABDA, type 2: CABC, type 3: ACB, type 4: BCAD. The arrival process of type *i* jobs is Poisson at rate λ_i.

1) Under what conditions is this system stable?
2) What is the joint stationary distribution of the number of jobs at each machine?



$\lambda_A=2\lambda_1+\lambda_2+\lambda_3+\lambda_4$ $\lambda_B=\lambda_1+\lambda_2+\lambda_3+\lambda_4$ $\lambda_C=2\lambda_2+\lambda_3+\lambda_4$ $\lambda_D=\lambda_1+\lambda_4$

1) $\lambda_A < \mu_A$, $\lambda_B < \mu_B$, $\lambda_C < 2\mu_C$, $\lambda_D < 3\mu_D$

2) Note that the machines A, B, C, and D are an M/M/1. M/M/1, M/M/2, and M/M/3 respectively.

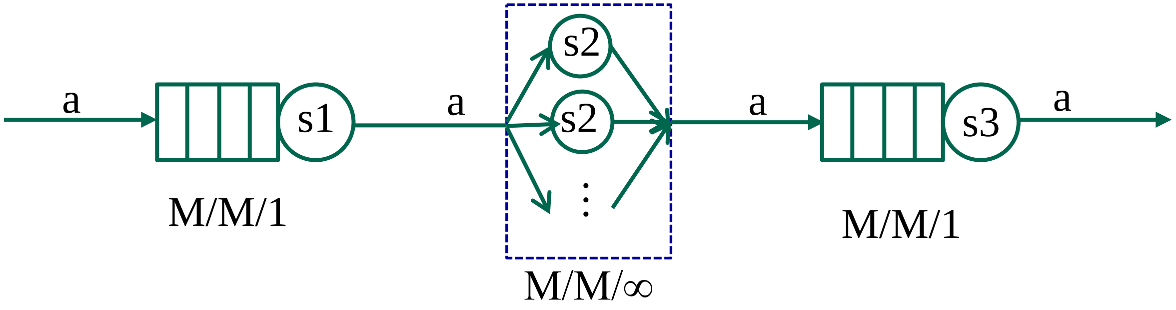
$$P(n_A, n_B, n_C, n_D) = \rho_A{}^{n_A}(1-\rho_A)\,\rho_B{}^{n_B}(1-\rho_B)\,2\rho_C{}^{n_C}P_c(0)\,\frac{9}{2}\rho_D{}^{n_D}P_D(0)$$

$n_A = \frac{\lambda_A}{\mu_A-\lambda_A}$, $n_B = \frac{\lambda_B}{\mu_B-\lambda_B}$, $n_C = \frac{4\mu_C\lambda_C}{4\mu_C{}^2-\lambda_C{}^2}$,

n_D : calculate by yourselves, using performance measure equation of an M/M/3 system

Exercise2: Open Queuing Networks

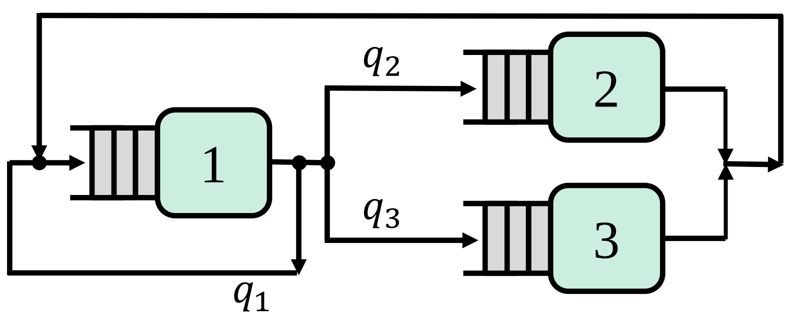
- 자장면 전문체인점에 자장면을 먹으러 오는 고객들이 도착율 a의 포아송 프로세스로 도착한다. 고객은 도착하자마자 주문을 한 후 자장면이 만들어질 때까지 기다린 후 자장면이 만들어지면 먹고 계산대에서 계산을 한 뒤 떠난다. 따라서 임의의 고객은 기다리든지 먹든지 계산을 하든지 셋 중 하나의 상태에 있다. 자장면을 만드는 요리사의 수와 는 계산원은 각각 1명이다. 요리사가 자장면을 만드는 시간과 먹는 시간, 그리고 계산하는데 걸리는 시간은 각각 평균이 s1, s2, s3인 지수분포를 따른다. 체인점은 충분히 넓다고 가정하자. 고객이 체인점에 머무는 시간을 구하라.



- $n_1 = \frac{a \times s1}{1 - a \times s1}$, $T_1 = \frac{s1}{1 - a \times s1}$, $T_2 = s2$, $n_3 = \frac{a \times s3}{1 - a \times s3}$, $T_3 = \frac{s3}{1 - a \times s3}$
- $T = T_1 + T_2 + T_3$

Closed Queuing Networks (1)

- ❖ Queuing network with no jobs from the outside
- ❖ The total number of jobs within the system is fixed.



  - N: the total number of jobs in the network
  - M: the number of devices in the network
- System state: $(n_1, n_2, \dots, n_M)$
  - n_i: number of jobs in device i
  - $N = \sum_{i=0}^{M} n_i$

Closed Queuing Networks (2)

- Assumptions for product form solution
  - The system is in steady state
  - All servers have exponentially distributed service time
  - Jobs are stochastically independent of each other
  - A job from device *i* joins device *j* with the (routing) probability q_{ij}

❖ Gordon and Newell's decomposition theorem

$$P(n_1, n_2, \dots, n_M) = \frac{1}{G} F_1(n_1)\, F_2(n_2) \dots F_M(n_M)$$

- $\sum_{\mathbf{n} \in \boldsymbol{S}(M,N)} P(n_1, n_2, \dots, n_M) = 1$
  - ✓ $\mathbf{n} = (n_1, n_2, \dots, n_M)$
  - ✓ $\boldsymbol{S}(M,N) = \{(n_1, n_2, \dots, n_M) | n_1 + n_1 + \cdots + n_M = N\}$
- Normalization factor $G = \sum_{\mathbf{n} \in \boldsymbol{S}(M,N)} \prod_{i=1}^{M} F_i(n_i)$

Closed Queuing Networks (3)

- $P(n_1, n_2, \ldots, n_M) = \frac{1}{G} F_1(n_1) F_2(n_2) \ldots F_M(n_M)$
- $F_i(n_i) = \begin{cases} 1 & , \quad n_i = 0 \\ V_i \times s_i(n_i) \times F_i(n_i - 1) & , \quad n_i \geq 1 \end{cases}$
  - V_i : Visit ratio of device *i* (relative input rate)
    - X_i :Throughput (Input rate) of device *i*
  - $s_i(n_i)$: the service time of device *i* when there are n_i jobs in device *i*

Insight: Remind that, for open queueing network,

$$P(n_1, n_2, \ldots, n_M) = P_1(n_1) P_2(n_2) \ldots P_M(n_M)$$

$$P_i(n_i) = \begin{cases} P_i(0) & , \quad n_i = 0 \\ \lambda_i \times s_i \times P_i(n_i - 1), & \quad n_i \geq 1 \end{cases} \quad \text{in M/M/1}$$

Closed Queuing Networks (4)

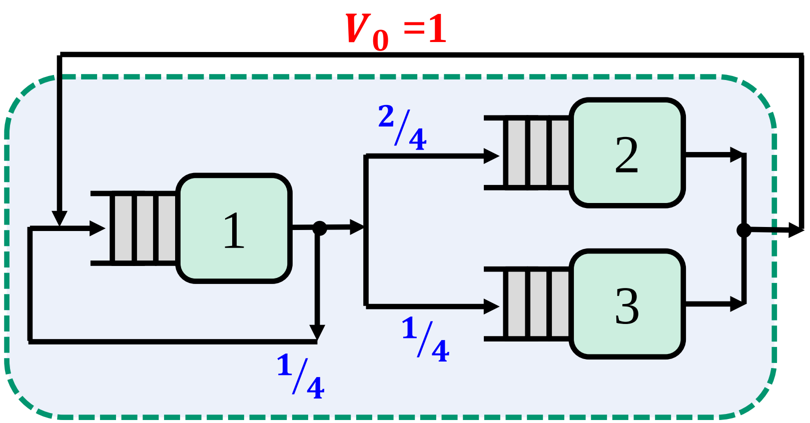
- Derivation of V_i,
  - For any appropriate link, V_0 =1.
  - Then, calculate other V_i values : $V_i = \sum_{j=0}^{M} V_j \, q_{ji}$

< Example >

$V_0 = 1 \,, V_0 = V_2 + V_3 \,,$

$V_1 = \frac{1}{4} V_1 + V_0 \,, V_2 = \frac{2}{4} V_1 \,, V_3 = \frac{1}{4} V_1$

$\Rightarrow \quad V_1 = \frac{4}{3} \,, V_2 = \frac{2}{3} \,, V_3 = \frac{1}{3}$



Closed Queuing Networks (5)

❖ Derivation for the product form solution

< Notation >

- $\mathbf{n} := (n_1, n_2, \dots, n_i, \dots, n_j, \dots, n_M)$
- $\mathbf{n}_{ij} := (n_1, n_2, \dots, n_i - 1, \dots, n_j + 1, \dots, n_M)$
- $r(\boldsymbol{n} \rightarrow \boldsymbol{k})$: transition rate from state $\boldsymbol{n}$ to state $\boldsymbol{k}$

1. Steady State Assumption (in-rate = out-rate)
   - $\sum_{\boldsymbol{l}} P(\boldsymbol{l}) r(\boldsymbol{l} \rightarrow \mathbf{n}) = P(\boldsymbol{n}) \sum_{\boldsymbol{k}} r(\boldsymbol{n} \rightarrow \boldsymbol{k})$

2. Exponential Server Assumption (in Steady State)
   - $\sum_{\mathbf{n}_{ij}} P(\mathbf{n}_{ij}) r(\mathbf{n}_{ij} \rightarrow \mathbf{n}) = P(\boldsymbol{n}) \sum_{\mathbf{n}_{ij}} r(\boldsymbol{n} \rightarrow \mathbf{n}_{ij})$

Closed Queuing Networks (6)

3. Independent Routing of each job

- $r(\boldsymbol{n} \to \mathbf{n}_{ij}) = q_{ij} \frac{\delta(n_i)}{s_i(n_i)}$ where $\delta(n_i) = \begin{cases} 1 & n_i > 0 \\ 0 & n_i = 0 \end{cases}$

$n_i > 0$

❖ Exponential Server and Independent Routing of each job

$$P(\boldsymbol{n}) \sum_{\mathbf{n}_{ij}} r(\boldsymbol{n} \to \mathbf{n}_{ij}) = \sum_{\mathbf{n}_{ij}} P(\mathbf{n}_{ij}) r(\mathbf{n}_{ij} \to \mathbf{n})$$

$n_i - 1 \geq 0, \quad n_j + 1 > 0$

$$\to \quad P(\boldsymbol{n}) \sum_i \sum_j q_{ij} \frac{\delta(n_i)}{s_i(n_i)} = \sum_i \sum_j \frac{\delta(n_i)\delta(n_j+1)}{s_j(n_j+1)} q_{ji} P(\mathbf{n}_{ij})$$

$$\boxed{P(\boldsymbol{n}) \sum_i \frac{\delta(n_i)}{s_i(n_i)} = \sum_i \sum_j \frac{\delta(n_i)}{s_j(n_j+1)} q_{ji} P(\mathbf{n}_{ij})}$$

Closed Queuing Networks (7)

- Define
  - $F_i(n_i) = \begin{cases} 1 & , \quad n_i = 0 \\ y_i \times s_i(n_i) \times F_i\,(n_i - 1) & , \quad n_i \geq 1 \end{cases}$

    where y_i's are unknown parameters

- Assume: $P(n_1, n_2, \ldots, n_M) = C\, F_1(n_1)\, F_2(n_2) \ldots F_M(n_M)$

From $P(\boldsymbol{n}) \sum_i \frac{\delta(n_i)}{s_i(n_i)} = \sum_i \sum_j \frac{\delta(n_i)}{s_j(n_j+1)}\, q_{ji}\, P(\mathbf{n}_{ij}),$

$$\cancel{C\, F_1(n_1)\, F_2(n_2) \ldots F_M(n_M)} \sum_i \frac{\delta(n_i)}{s_i(n_i)}$$

$F_i(n_i-1)$

$$= \sum_i \sum_j \frac{\cancel{\delta(n_i)}}{\cancel{s_j(n_j+1)}}\, q_{ji}\, \cancel{C\, F_1(n_1)\, F_2(n_2)} \ldots \frac{\delta(n_i)}{y_i\, s_i(n_i)}\, \cancel{F_i(n_i)}$$

$$\ldots y_j\, \cancel{s_j(n_j+1)}\, F_j(n_j)\, \cancel{\ldots F_M(n_M)}$$

$F_j(n_j+1)$

Closed Queuing Networks (8)

- $\sum_i \frac{\delta(n_i)}{s_i(n_i)} = \sum_i \sum_j \frac{\delta(n_i)}{s_i(n_i)} q_{ji} \frac{y_j}{y_i}$

  $\sum_i \frac{\delta(n_i)}{s_i(n_i)} \left(1 - \sum_j \frac{y_j}{y_i} q_{ji}\right) = 0$

  Thus, $1 = \sum_j \frac{y_j}{y_i} q_{ji}$

$$\boxed{y_i = \sum_{j=1}^{M} y_j q_{ji}}$$

- Note that the y_i's can be anything as long as they satisfy the above equation.
  - Applying to the throughput, $X_i = \sum_{j=1}^{M} X_j q_{ji}$
  - Applying to the visit ratio $V_i := X_i / X_0$, $V_i = \sum_{j=1}^{M} V_j q_{ji}$

Closed Queuing Networks (9)

- In summary

$$P(n_1, n_2, \dots, n_M) = \frac{1}{G} F_1(n_1)\, F_2(n_2) \dots F_M(n_M)$$

- $G = \sum_{\mathbf{n} \in \boldsymbol{S}(M,N)} \prod_{i=1}^{M} F_i(n_i)$
- $F_i(n_i) = \begin{cases} 1 & , \quad n_i = 0 \\ V_i \times s_i(n_i) \times F_i\,(n_i - 1) & , \quad n_i \geq 1 \end{cases}$
- $V_i = \sum_{j=0}^{M} V_j\, q_{ji}$ (For any appropriate link, $V_0 = 1$)

Note that the number of feasible states can be too many to calculate *G*

Closed Queuing Networks (10)

❖ Buzen's Recursive Algorithm for simply calculating G

– Let $g_m(n) \coloneqq \sum_{\mathbf{n} \in \boldsymbol{S}(m,n)} \prod_{i=1}^{m} F_i(n_i)$

where $\mathbf{n} = (n_1, n_2, \ldots, n_m)$, $\boldsymbol{S}(m,n) = \{(n_1, n_2, \ldots, n_m) | n_1 + n_1 + \cdots + n_m = n\}$

– $G = g_M(N)$

– $g_1(n) = F_1(n)$

– $g_m(0) = \prod_{i=1}^{m} F_i(0) = 1$

– $g_m(n) = \sum_{k=0}^{n} F_m(k) \sum_{(n_1 \ldots n_{m-1}) \in \boldsymbol{S}(m-1,n-k)} \prod_{i=1}^{m-1} F_i(n_i)$, $(n > 0, m > 1)$

$= \sum_{k=0}^{n} F_m(k) g_{m-1}(n-k)$

$g_m(n)$ can be calculated in a recursive fashion

Closed Queuing Networks (11)

- Calculation of $g_m(n)$:

$$g_m(n) = g_{m-1}(0)F_m(n) + g_{m-1}(1)F_m(n-1) + g_{m-1}(2)F_m(n-2) + \cdots + g_{m-1}(n-1)F_m(1) + g_{m-1}(n)F_m(0)$$

	1	2		m-1	m		M	
0	1	1	…	1 $\times F_m(n)$	1	…	1	
1	$F_1(1)$	$g_2(1)$	…	$+$ $g_{m-1}(1) \times F_m(n\text{-}1)$				
⋮	⋮	⋮		⋮				
n-1	$F_1(n\text{-}1)$	$g_2(n\text{-}1)$	…	$+$ $g_{m-1}(n\text{-}1) \times F_m(1)$				
n	$F_1(n)$	$g_2(n)$	…	$+$ $g_{m-1}(n) \times F_m(0)$	$= g_m(n)$			
⋮	⋮	⋮					$g_M(N\text{-}1)$	
N	$F_1(N)$	$g_2(N)$					$g_M(N)$	$= \boldsymbol{G}$

Closed Queuing Networks (12)

- When the service rate of each device is constant (a single server)
  - $s_i(n_i) = s_i$, $\forall n_i \geq 1 \quad \Rightarrow \quad F_m(k) = V_m s_m F_m(k-1)$
- $g_m(n) = F_m(0) g_{m-1}(n) + \sum_{k=1}^{n} F_m(k) g_{m-1}(n-k)$

$$= g_{m-1}(n) + V_m s_m \underbrace{\sum_{k=1}^{n} F_m(k-1) g_{m-1}(n-k)}_{\sum_{a=0}^{n-1} F_m(a) g_{m-1}(n-1-a)}$$

$$= g_{m-1}(n) + V_m s_m \, g_m(n-1)$$

	1	2		m-1	m		M
0	1	1	…	1	1	…	1
1	$F_1(1)$	$g_2(1)$	…	$g_{m-1}(1)$	$g_m(1)$		
⋮	⋮	⋮		⋮			
n-1	$F_1(n\text{-}1)$	$g_2\,(n\text{-}1)$	…	$g_{m-1}\,(n\text{-}1)$	$g_m(n\text{-}1)$ × $V_m s_m$		
n	$F_1(n)$	$g_2(n)$	…	$g_{m-1}(n)$	+ = $g_m(n)$		
⋮	⋮	⋮					
N	$F_1(N)$	$g_2(N)$					

Closed Queuing Networks (13)

- Performance measure
  - Throughput of device M: X_M
    - $X_M = \sum_{k=1}^{N} P_M(k) \frac{1}{s_M(k)}$
      - ✓ $P_M(k)$: Probability that there are k jobs in the device M
      - ✓ $P_M(k) = \sum_{(n_1,n_2,\dots,n_{M-1}) \in \boldsymbol{S}(M-1,N-k)} P(n_1, \dots, n_{M-1}, k)$

$$= \underline{\sum_{(n_1,n_2,\dots,n_{M-1}) \in \boldsymbol{S}(M-1,N-k)} \frac{1}{G} F_1(n_1) \dots F_{M-1}(n_{M-1})} F_M(k)$$

$$= \frac{1}{G} F_M(k) \underline{g_{M-1}(N-k)}$$

$$\Rightarrow X_M = \sum_{k=1}^{N} \frac{1}{G} F_M(k) g_{M-1}(N-k) \frac{1}{s_M(k)}$$

$$= \sum_{k=1}^{N} \frac{1}{G} V_M \cancel{s_M}(k) F_M(k-1)\, g_{M-1}(N-k) \frac{1}{\cancel{s_M}(k)}$$

$$= \frac{1}{G} V_M \underbrace{\sum_{k=1}^{N} F_M(k-1)\, g_{M-1}(N-k)}_{\sum_{m=0}^{N-1} F_M(m)\, g_{M-1}(N-1-m)} = \frac{1}{G} V_M\, \underline{g_M(N-1)}$$

Closed Queuing Networks (14)

Since $\frac{X_i}{X_j} = \frac{V_i}{V_j}$ for any devices i , j

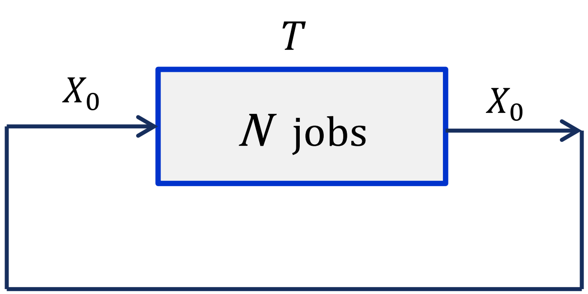
– System throughput : X_0

$$X_0 = \frac{X_M}{V_M} = \frac{g_M(N-1)}{G}$$

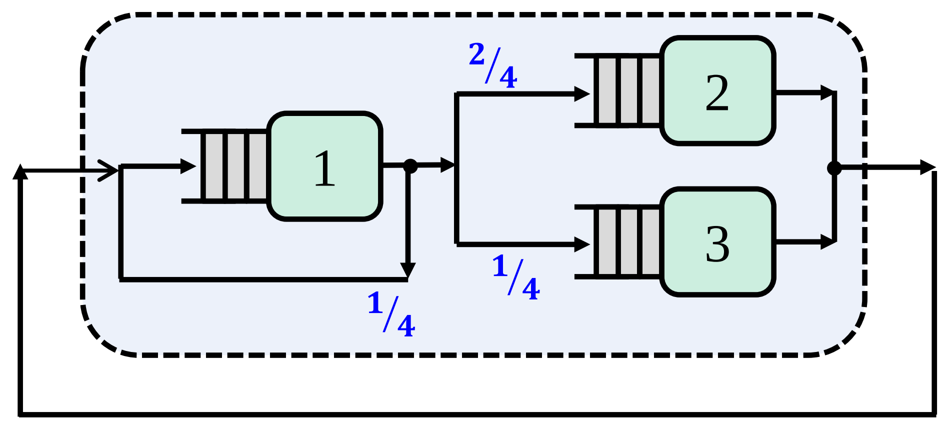
– Throughput of any device i

$$X_i = V_i X_0$$

– System response time: T



By Little's Law, $T = \frac{N}{X_0}$



$V_0 = 1$