Queueing Network

Chang-Gun Lee

M/M/1 Queueing Review

- The number of arrivals follows the Poisson distribution
- The service time follows the Exponential distribution
- The number of servers is ONE

$$\lambda_n = \lambda \text{ for all } n = 0,1,2,\cdots$$

$$\mu_n = \mu \text{ for all } n = 1,2,3,\cdots$$

$$\text{Let } \frac{\lambda}{\mu} = \rho$$

$$S = 1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu^2} + \cdots = 1 + \rho + \rho^2 + \cdots = \frac{1}{1-\rho}$$

Thus,

$$P_0 = 1 - \rho$$

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

M/M/1 Queueing

- L=E(N): Expected number of customers in the system

$$L = E(N) = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho} \text{ (see geometric dist.)}$$

- W: Average waiting time of a customer in the system
  - Use Little's Law

Little's Law
(by John D. C. Little)

$$L = \lambda W$$

Avg. # of customers in the system

Customer arrival rate

Avg. time a customer spends in the system

Proof is difficult. See Stidham's paper.

M/M/1 Queueing

- L=E(N): Expected number of customers in the system

$$L = E(N) = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho} \quad \text{(see geometric dist.)}$$

- W: Average waiting time of a customer in the system
  - Use Little's Law

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

M/M/1 Queueing

- L_q: Expected number of customers in the queue
- W_q: Average waiting time of a customer in the queue

$$W_q = W - W_s = \frac{W_s}{1-\rho} - W_s = \frac{\rho W_s}{1-\rho}$$

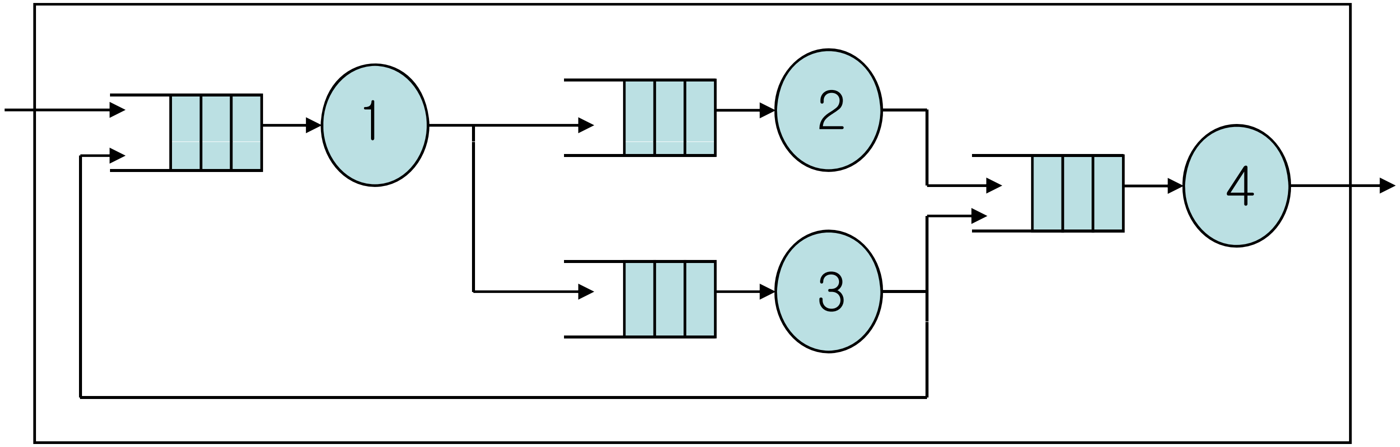
$$L_q = \lambda W_q = \lambda \frac{\rho W_s}{1-\rho} = \frac{\rho^2}{1-\rho}$$

- Probability that the server is busy?

$$L_s = \lambda W_s = \lambda \frac{1}{\mu} = \rho$$

Basics

- Queueing Network
  - A computer system is essentially a network of queues



- Forced flow law

Average system throughput

Average throughput of resource k

$$\lambda_k = \lambda V_k$$

# of visits to resource k per job

- Total service demand

Total service demand of a job for resource k

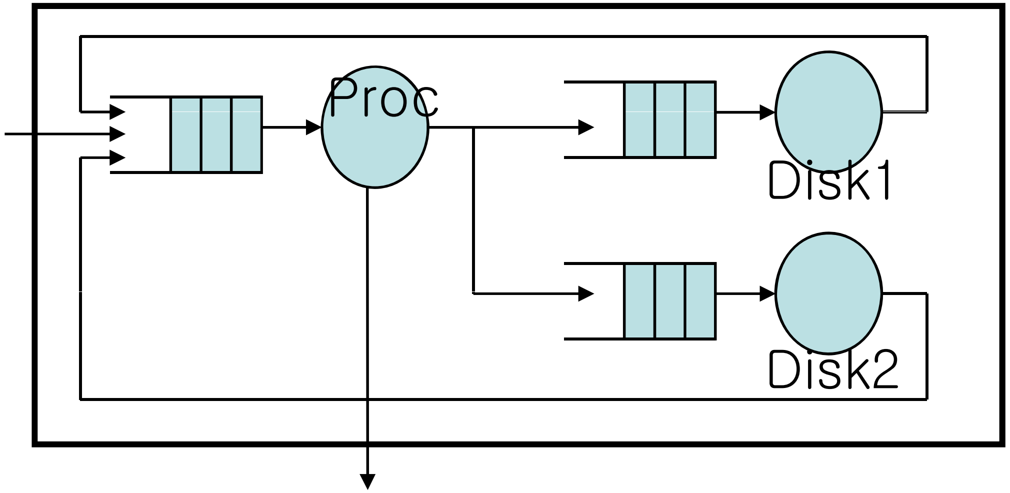
$$D_k = V_k S_k$$

Service time of a job per visit to resource k

# of visits to resource k per job

Example 1

- We made measurements on a batch processing machine. These indicate that the average number of visits each job makes to Drive 1 is five and that the disk throughput for Drive 1 is 10 requests per second. What is the system throughput?

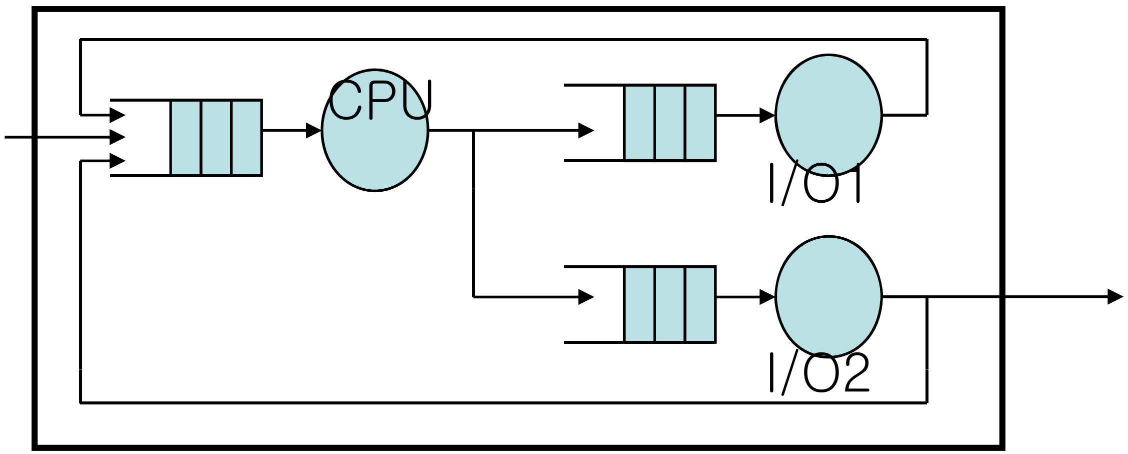


$$\lambda_k = \lambda V_k$$

$$\lambda = \lambda_k / V_k = 10/5 = 2 \text{ requests/second}$$

Example 2

- We measure a small batch processing computer system. We find that CPU has a visit ratio V_1=20 with S_1=0.05 seconds, the first I/O device has V_2=11 and S_2=0.08 seconds, while the other I/O device has V_3=8 and S_3=0.04 seconds. What is the bottleneck resource?



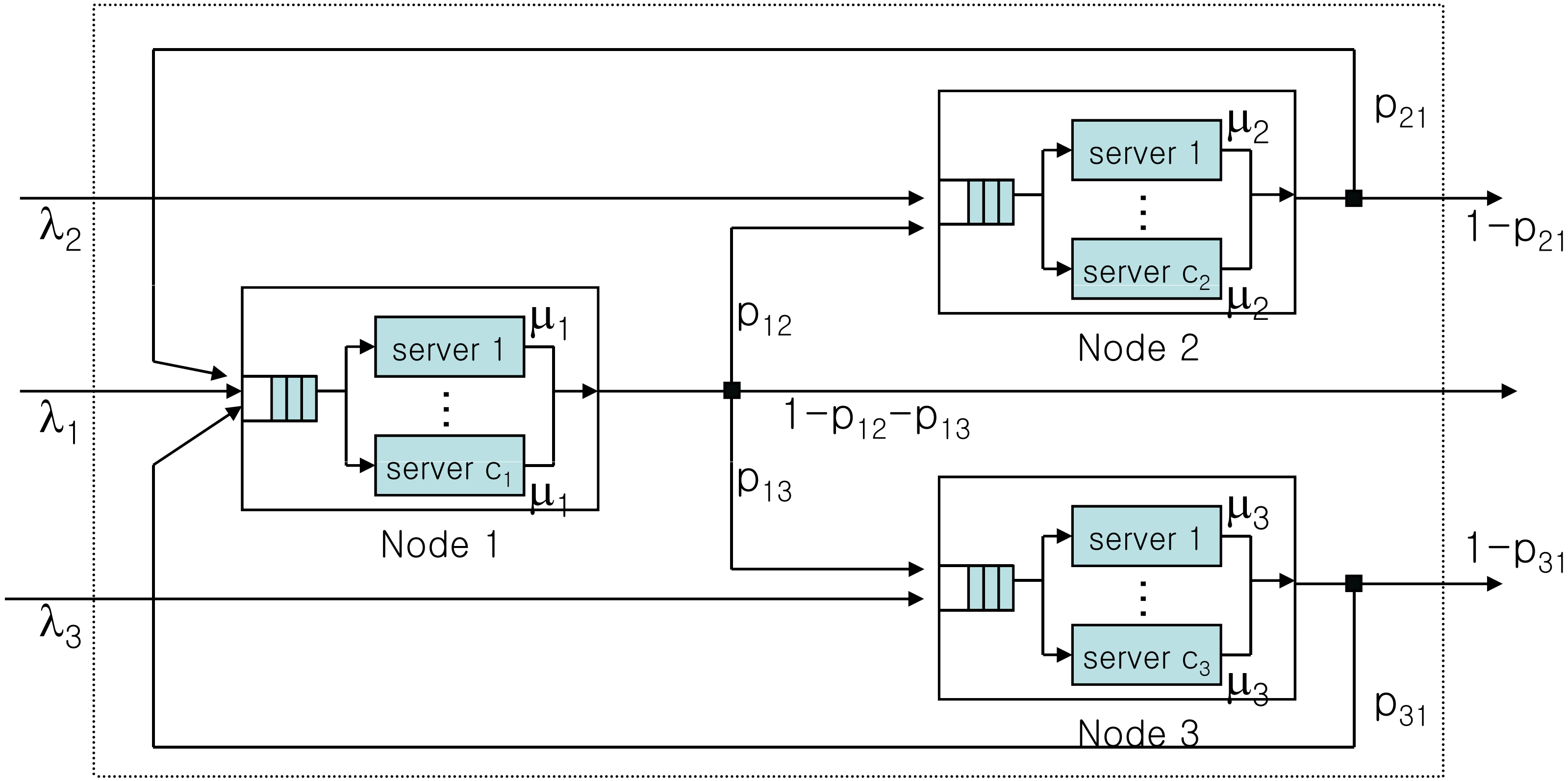
$$D_1 = V_1 S_1 = 1$$

$$D_2 = V_2 S_2 = 0.88$$

$$D_3 = V_3 S_3 = 0.32$$

Jackson Networks

- A queueing network consisting of K nodes
  - Each node consists of c_k identical exponential servers, each with average service rate μ_k
  - Customers arriving at node k from outside the system arrive in Poisson pattern with the average arrival rate λ_k
  - Once served at node k, a customer goes (instantly) to node j (j=1,2,…,K) with probability p_{kj} or leaves the network with probability $1 - \Sigma^{K}_{j=1} p_{kj}$.



Jackson's Theorem

- For each node k, the average arrival rate to the node, Λ_k, is given by

$$\Lambda_k = \lambda_k + \sum_{j=1}^{K} p_{jk}\Lambda_j$$

- Let $(n_1, n_2, \cdots, n_K)$ be the system state. Steady state prob?
- If $\Lambda_k < c_k\mu_k$ for $k=1,2,\cdots K$, there is a steady state distribution of the system state $(n_1, n_2, \cdots, n_K)$:
  - $p(n_1, n_2, \cdots, n_K)$ = steady state probability that there are n_k customers in the k-th node for $k=1, 2, \cdots, K$.

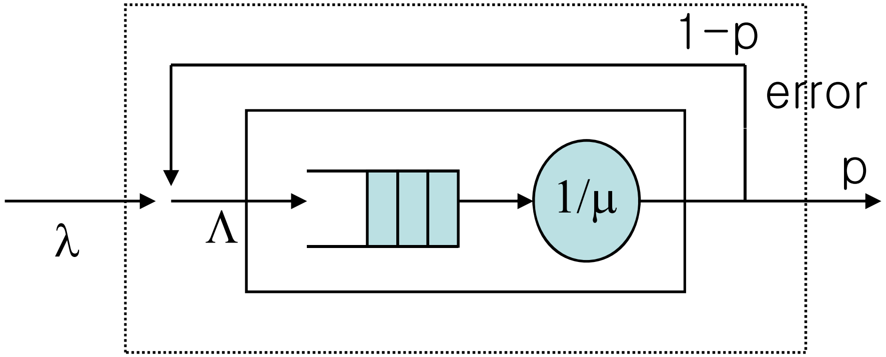
- The steady state probability that the system is in state $(n_1, n_2, \cdots, n_K)$:

$$p(n_1, n_2, \cdots n_K) = p_1(n_1)p_2(n_2)\cdots p_K(n_K)$$

that is, each node k behaves as if it were an independent $M/M/c_k$ queueing system with average arrival rate Λ_k.

Example

- Error recovery transmission system



$$\Lambda = \lambda + (1-p)\Lambda$$

$$\Lambda = \lambda / p$$

$$\text{server utilization} : \rho = \frac{\lambda / p}{\mu} = \frac{\lambda}{p\mu}$$

M/M/1 sytem

$$L = E(N) = \frac{\rho}{1-\rho} = \frac{\lambda}{p\mu - \lambda}$$

$$W_{small} = \frac{L}{\Lambda} = \frac{\frac{\lambda}{p\mu-\lambda}}{\frac{\lambda}{p}} = \frac{p}{p\mu - \lambda}$$

$$W_{big} = \frac{L}{\lambda}$$

$\lambda = 4$ messages per second

$1/\mu = 0.22$ sec

(time for transmitting a MSG and receiving ACK)

$p = 0.99, 1-p = 0.01$

$$\Lambda = 4.0404 \text{ messages per sec}$$

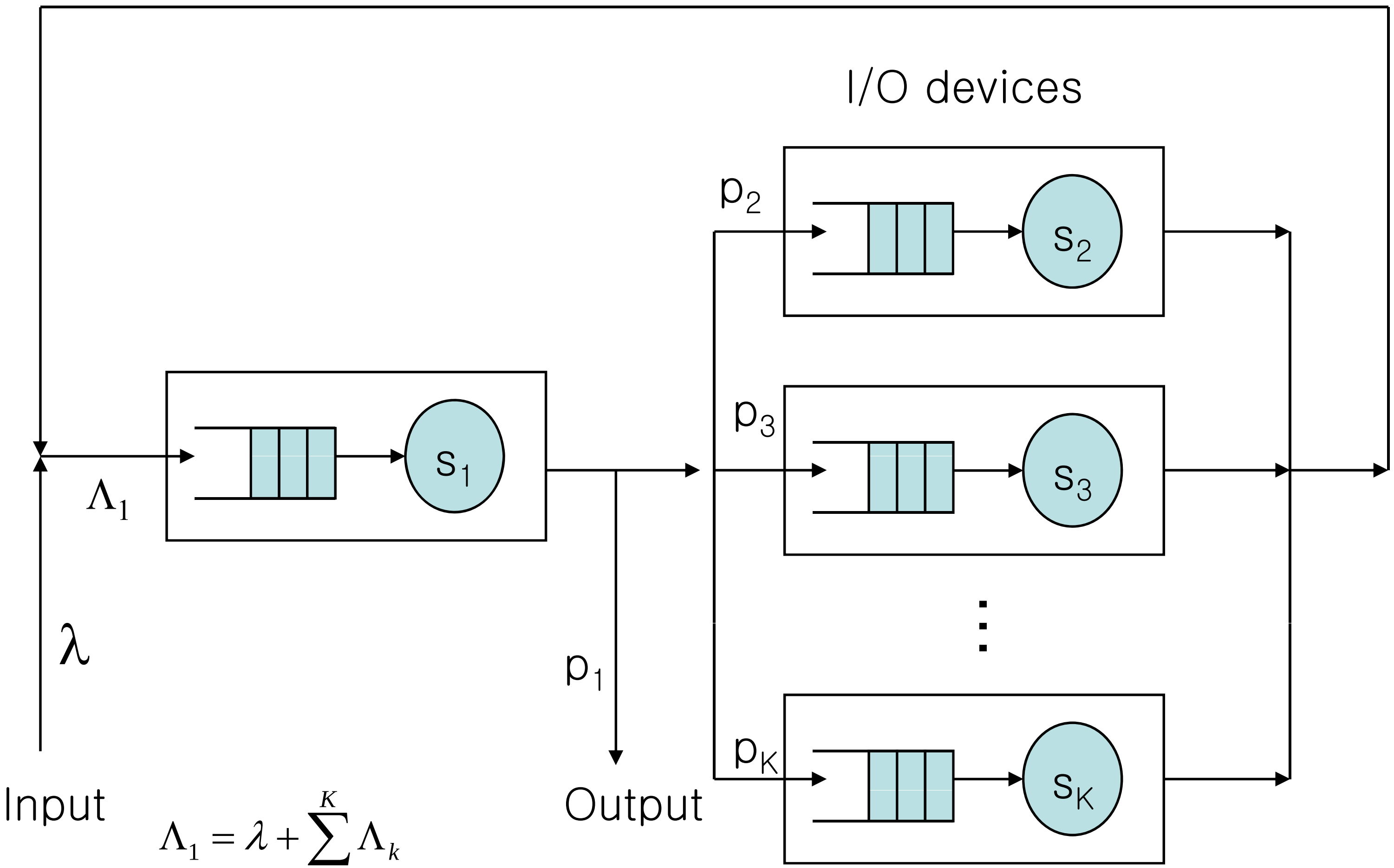
$$\rho = 0.8889$$

$$L = 8 \text{ messages}$$

$$W_{small} = 1.98 \text{ sec}$$

$$W_{big} = 2.0 \text{ sec}$$

Open central server system

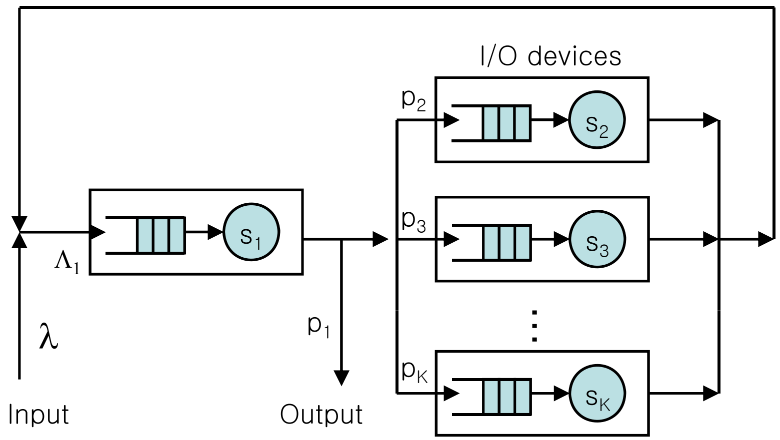


$$\Lambda_1 = \lambda + \sum_{k=2}^{K} \Lambda_k$$

$$\Lambda_k = p_k \Lambda_1$$

$$\Lambda_1 = \frac{\lambda}{p_1}, \Lambda_k = p_k \frac{\lambda}{p_1}$$

Open central server system



$$\Lambda_1 = \lambda + \sum_{k=2}^{K} \Lambda_k$$

$$\Lambda_k = p_k \Lambda_1$$

$$\Lambda_1 = \frac{\lambda}{p_1}, \Lambda_k = p_k \frac{\lambda}{p_1}$$

$$\rho_1 = \frac{\Lambda_1}{1/s_1} = \frac{\lambda s_1}{p_1},$$

$$\rho_k = \frac{\Lambda_k}{1/s_k} = \frac{\lambda p_k s_k}{p_1} (\text{when } 2 \le k \le K),$$

L_k ? W_k ?

$$L_k = \frac{\rho_k}{1-\rho_k}$$

$$W_k = \frac{L_k}{\Lambda_k} = \frac{\rho_k}{1-\rho_k} \frac{p_1}{\lambda p_k} = \frac{\frac{\lambda p_k s_k}{p_1}}{1-\rho_k} \frac{p_1}{\lambda p_k} = \frac{s_k}{1-\rho_k}$$

$$L = \sum_{k=1}^{K} L_k, W = \frac{L}{\lambda}$$

Example

1 CPU and 3 I/O devices

λ=1/5 jobs per second

CPU service time s_1 = 0.75 sec

I/O service times s_2, s_3, s_4 = 1, 2, 4 sec

$p_1=p_2=p_3=p_4=0.25$

$$\rho_1 = 0.6, \rho_2 = 0.2, \rho_3 = 0.4, \rho_4 = 0.8$$

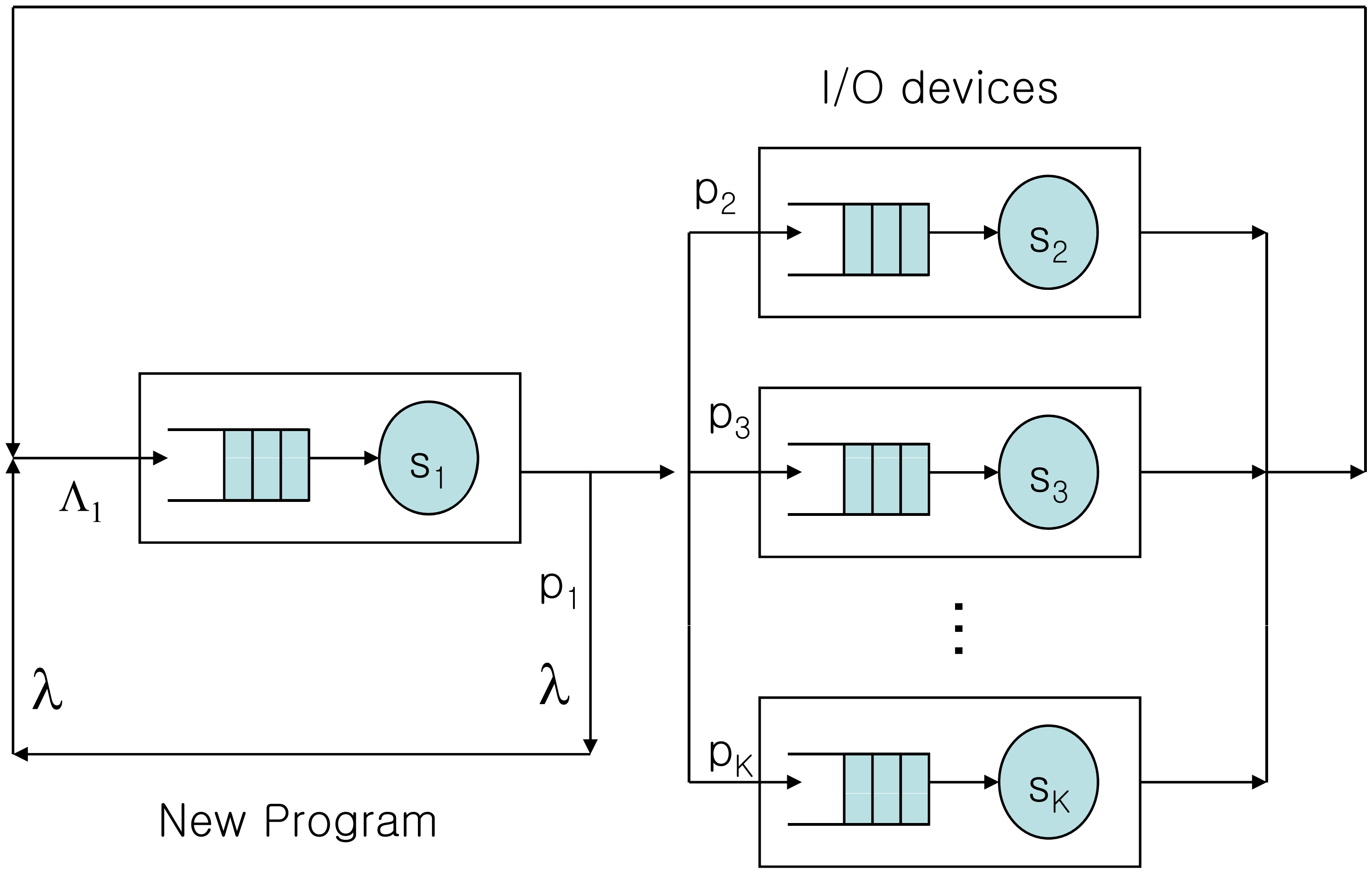
$$L_1 = 1.5, L_2 = 0.25, L_3 = 0.667, L_4 = 4$$

$$W_1 = 1.875, W_2 = 1.25, W_3 = 3.33, W_4 = 20.00$$

$$L = 6.417$$

$$W = L / \lambda = 32.085$$

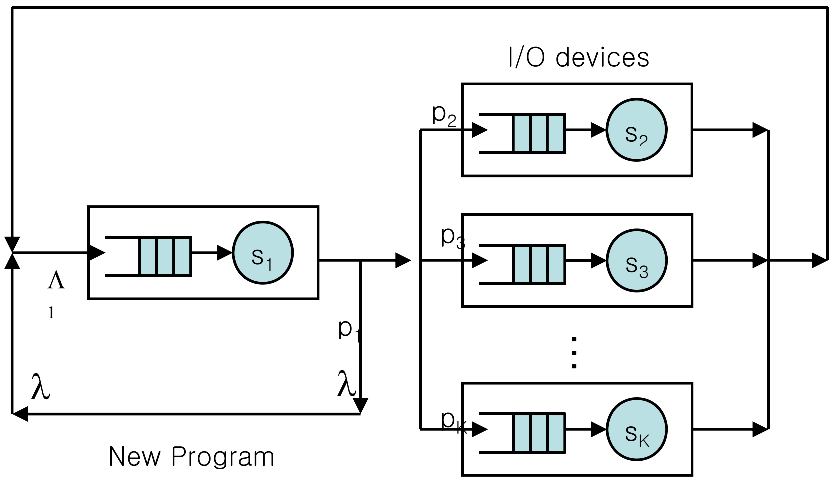
Closed central server system



N programs are running concurrently: multiprogramming level=N

If a program completes, a new program starts immediately.

Closed central server system



D_k : mean total resource requirement per job

$= V_k S_k$

$L_k(n)$: when multiprogramming level is n, average no. of jobs in resource k

$W_k(n)$: when multiprogramming level is n, average total time of a program in resource k

step 1 : $L_k(0) = 0$ for all $k = 1, 2, \cdots, K$

step 2(1) : $W_k(1) = D_k(1+0), k = 1, 2, \cdots, K$

$$W(1) = \sum_{k=1}^{K} W_k(1)$$

$$\lambda(1) = \frac{n}{W(1)}$$

$V_k W'_k$

$$L_k(1) = \lambda(1) W_k(1), k = 1, 2, \cdots, k$$

step 2(n) : $W_k(n) = D_k(1 + L_k(n-1)), k = 1, 2, \cdots, K$

$$W(n) = \sum_{k=1}^{K} W_k(n)$$

$$\lambda(n) = \frac{n}{W(n)}$$

$$L_k(n) = \lambda(n) W_k(n), k = 1, 2, \cdots, K$$

step 3 : $W = W(N)$

$$\lambda = \lambda(N)$$

$$\rho_k = \lambda D_k, k = 1, 2, \cdots, K$$

Example

- Suppose we have a computer system that consists of a CPU and one I/O device, i.e., K=2. The multiprogramming level is 2, i.e., N=2.The average CPU time needed per job is 0.4 seconds while an average of 0.6 seconds of I/O service is required (i.e., D_1=0.4, D_2=0.6).

$$\text{step } 1: L_1(0) = L_2(0) = 0$$

$$\text{step } 2(1):$$

$$W_1(1) = D_1(1+0) = 0.4, W_2(1) = D_2(1+0) = 0.6$$

$$W(1) = 1.0, \lambda(1) = \frac{1}{1.0} = 1$$

$$L_1(1) = \lambda(1)W_1(1) = 0.4, L_2(1) = \lambda(1)W_2(1) = 0.6$$

$$\text{step } 2(1):$$

$$W_1(1) = D_1(1+0) = 0.4, W_2(1) = D_2(1+0) = 0.6$$

$$W(1) = 1.0, \lambda(1) = \frac{1}{1.0} = 1$$

$$L_1(1) = \lambda(1)W_1(1) = 0.4, L_2(1) = \lambda(1)W_2(1) = 0.6$$

$$\text{step } 2(1):$$

$$W_1(2) = D_1(1+L_1(1)) = 0.56,$$

$$W_2(2) = D_2(1+L_2(1)) = 0.96$$

$$W(2) = 0.56 + 0.96 = 1.52, \lambda(2) = \frac{2}{1.52} = 1.3158$$

$$\text{Step } 3:$$

$$W = W(2) = 1.52, \lambda = \lambda(2) = 1.3158$$

$$\rho_1 = \lambda D_1 = 0.5263, \rho_2 = \lambda D_2 = 0.7895$$

How do we know D_k?

$$V_1 = \frac{1}{p_1}$$

$$V_k = p_k V_1, k = 2,3,\cdots,K$$

$$D_k = V_k S_k, k = 1,2,\cdots,K$$