HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

Các mô hình hàng đợi

ONE LOVE. ONE FUTURE.

Nội dung

- Định lý Little
- Kí hiệu Kendal
- Quá trình Sinh – Tử
- Hàng đợi M/M/1
- Hàng đợi M/M/1/K
- Hàng đợi M/M/c
- Hàng đợi M/M/c/c

Định lý Little (1)

- Xét một mạng đóng bất kỳ với các tham số sau:
- *N(t)*- Số yêu cầu nằm trong hệ thống tại thời điểm *t*.
- $\alpha(t)$ - Số yêu cầu đi đến hệ thống trong khoảng thời gian từ (*0,t*).
- $\beta(t)$ - Số yêu cầu rời khỏi hệ thống trong khoảng thời gian từ (*0,t*).

Định lý Little (2)

- Nếu gọi N_t là số lượng yêu cầu trung bình nằm trong hệ thống trong (*0,t*), ta có:

$$N_t = \frac{1}{t}\int_0^t N\ \ (t)dt$$

Định lý Little (3)

- T_i là thời gian mà yêu cầu *i* lưu lại trong hệ thống.
- T_t là thời gian lưu lại trung bình của yêu cầu trong hệ thống trong khoảng thời gian (*0,t*):

$$T_t = \frac{1}{\alpha(t)} \sum_{i=1}^{\alpha(t)} T_i$$

Định lý Little (4)

- λ_t là mật độ cuộc gọi trong khoảng thời gian $(0,t)$ ta có

$$\lambda_t = \frac{\alpha(t)}{t}$$

Định lý Little (4)

- Giả sử các giới hạn sau tồn tại

$$N = \lim_{t\to\infty} N_t \quad ; \quad \lambda = \lim_{t\to\infty} \lambda_t \quad ; \quad T = \lim_{t\to\infty} T_t$$

Định lý Little (5)

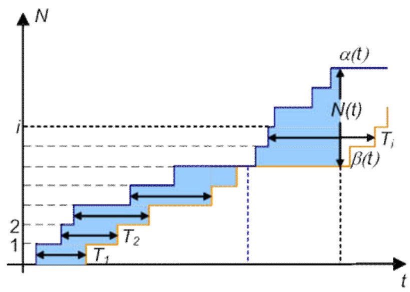
Định lý Little: Số yêu cầu trung bình nằm trong hệ thống bằng tích của tốc độ tới trung bình với thời gian lưu lại hệ thống trung bình của yêu cầu:

$N = \lambda T$;

Chứng minh (1)

- Xét trong khoảng (0,t) :
- Diện tích phần màu xanh:

$$S_t = \int_0^t N(t)dt = \int_0^t [\alpha(t) - \beta(t)]dt$$



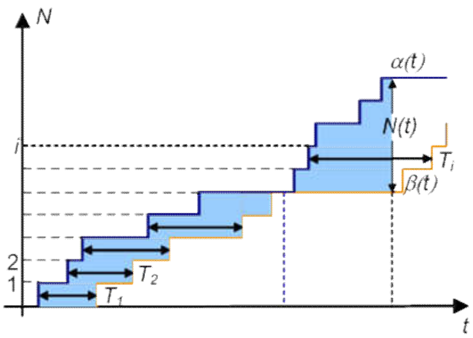
Chứng minh (2)

- Mặt khác diện tích này cũng bằng

$$S_t = 1 \times \sum_{i=1}^{\alpha(t)} Ti$$

- Suy ra

$$\int_0^t N(t)dt = \sum_{i=1}^{\alpha(t)} Ti$$



Chứng minh (3)

$$\frac{1}{t}\int_o^t N(t)dt = \frac{\alpha(t)}{t}\frac{\sum_{i=1}^{\alpha(t)} T_i}{\alpha(t)}$$

$$N_t = \lambda_t T_t$$

Giả sử các giới hạn sau tồn tại ta có

$$N = \lim_{t\to\infty} N_t \quad ; \quad \lambda = \lim_{t\to\infty} \lambda_t \quad ; \quad T = \lim_{t\to\infty} T_t$$

$\rightarrow N = \lambda$t (đpcm)

Kí hiệu Kendall (1)

- Ký hiệu Kendall được sử dụng để mô tả hệ thống hàng đợi đơn, do David George Kendall phát triển. Hệ thống hàng đợi được mô tả bằng một chuỗi ký tự như sau:
- *A/S/C/[K]/[P]/[D]*
- *A/S/m/[B]/[K]/[SD]*
- …..

Kí hiệu Kendall (2)

- *A/S/m/[B]/[K]/[SD]*
- ***A: Tiến trình tới*** (arrival process)
  - *M*: Tiến trình mũ (là tiến trình Markov hay tiến trình không nhớ)
  - *Er*: Tiến trình Erlang bậc *r*
  - *Hr*: Tiến trình siêu số mũ bậc *r*
  - *D*: Tiến trình tất định (deterministic)
  - *G*: Tiến trình chung

Kí hiệu Kendall (3)

- *A/S/m/[B]/[K]/[SD]*
- *S: Tiến trình phục vụ* (service process)
  - *M*: Tiến trình mũ (là tiến trình Markov hay tiến trình không nhớ)
  - *Er*: Tiến trình Erlang bậc *r*
  - *Hr*: Tiến trình siêu số mũ bậc *r*
  - *D*: Tiến trình tất định (deterministic)
  - *G*: Tiến trình chung

Kí hiệu Kendall (4)

- *A/S/m/[B]/[K]/[SD]*
- *M: number of servers: Số trạm phục vụ*
  - *1*
  - *N*
  - ∞

Kí hiệu Kendall (5)

- *A/S/m/[B]/[K]/[SD]*
- *B: Number of Buffers (Số lượng vị trí trong hàng đợi / System capacity (dung lượng hệ thống)*
- *Ví dụ m=1*
  - Buffer=∞, *Capacity= ∞*
  - Buffer=*N-1, Capacity=N*
  - Buffer=*0, Capacity=1*
- *Giá trị mặc định* Buffer=∞, *Capacity= ∞*

Kí hiệu Kendall (6)

- *A/S/m/[B]/[K]/[SD]*
- *K: Population size: Số lượng yêu cầu đi vào hệ thống*
- *Giá trị mặc định =∞*

Kí hiệu Kendall (7)

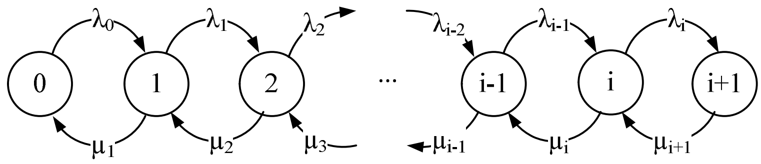
- *A/S/m/[B]/[K]/[SD]*
- *SD: Service discipline: Quy tắc phục vụ*
  - *FCFS*
  - *LCFS*
  - *Round Robin*
  - *SIRO*
- *Giá trị mặc định* =FCFS

Quá trình sinh – tử (1)

- Quá trình Sinh-Tử (Birth-Death Process)
- Trạng thái của hệ thống được biểu diễn bằng số các khách hàng n trong một hệ thống.
  - Khi có một khách hàng mới đến thì trạng thái của hệ thống sẽ thay đổi sang n+1
  - khi có một khách hàng ra đi thì trạng thái hệ thống sẽ thay đổi sang n-1,

Quá trình sinh – tử (2)

- .

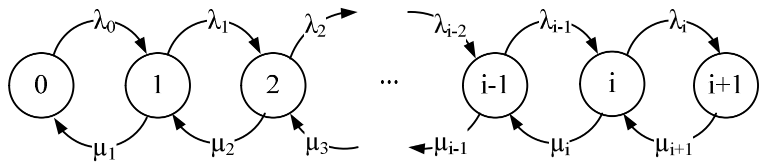


- Chuỗi Markov của một quá trình sinh-tử

λ_n : Tốc độ của lần đến *n*

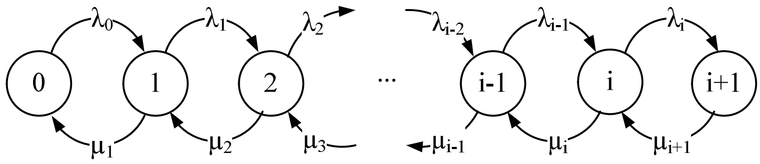
μ_n : Tốc độ của lần đi

Quá trình sinh – tử (3)



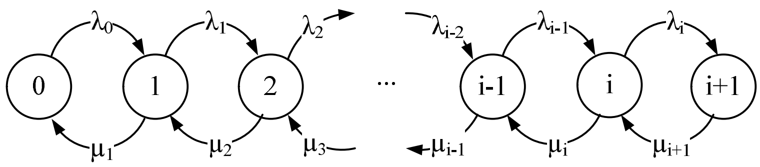
- Khi hệ thống ở trạng thái ổn định ta có
- $p_0\lambda_0 = p_1\mu_1$
- $\rightarrow p_1 = \frac{\lambda_0}{\mu_1} p_0$

Quá trình sinh – tử (4)



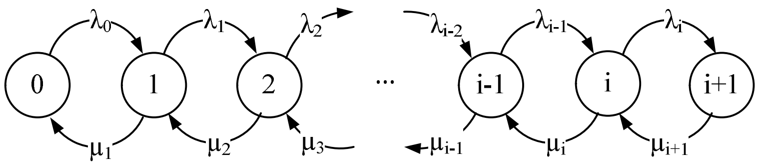
- Khi hệ thống ở trạng thái ổn định ta có
- $p_1\lambda_1 = p_2\mu_2$
- $\rightarrow p_2 = \frac{\lambda_1}{\mu_2} p_1$

Quá trình sinh – tử (5)



- Làm tương tự ta có
- $p_{i-1}\lambda_{i-1} = p_i\mu_i$
- $\rightarrow p_i = \frac{\lambda_{i-1}}{\mu_i} p_{i-1}$

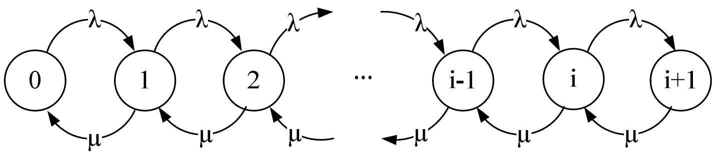
Quá trình sinh – tử (6)



- $\rightarrow p_i = \frac{\lambda_0 \lambda_1 \ldots \lambda_{i-1}}{\mu_1 \mu_2 \ldots \mu_i} p_0$

Hàng đợi M/M/1 (1)

- Tất cả các tốc độ đến đều là λ
- Tất cả các tốc độ đi đều là μ



- $p_i = \left(\frac{\lambda}{\mu}\right)^i p_0$
- Đặt ρ $= \frac{\lambda}{\mu}$ $ta\ có\ p_i = (\rho)^i p_0$

M/M/1(2)

- Ta có

$$\sum_{i=0}^{\infty} p_i = 1 \qquad \Rightarrow \sum_{i=0}^{\infty} p_0\rho^i = 1 \qquad \Rightarrow p_0\sum_{i=0}^{\infty} \rho^i = 1$$

- Mà ta có

$$\sum_{i=0}^{\infty} \rho^i = \frac{1}{1-\rho}\ với\ \rho < 1 \qquad \Rightarrow p_0 = 1-\rho$$

$$p_{N+1} = \rho^{N+1}(1-\rho)$$

Hàng đợi M/M/1 (3)

- *Số lượng trung bình của khách hàng trong hệ thống*

$$E(N) = \sum_{i=0}^{\infty} ip_i = \sum_{i=0}^{\infty} i\rho^i(1-\rho) = \rho(1-\rho)\sum_{i=1}^{\infty} i\rho^{i-1}$$

- Mà ta có

$$\sum_{i=0}^{\infty} \rho^i = \frac{1}{1-\rho}\ với\ \rho < 1 \qquad \Rightarrow \sum_{i=1}^{\infty} i\rho^{i-1} = \frac{1}{(1-\rho)^2}\ với\ \rho < 1$$

$$\Rightarrow E(N) = \frac{\rho}{1-\rho}$$

Hàng đợi M/M/1 (4)

- *Thời gian trung bình của khách hàng trong hệ thống*
- $Theo\ định\ lý\ Little\ ta\ có$
- $E(T) = \frac{E(N)}{\lambda}$
- $\Longrightarrow E(T) = \frac{\rho}{\lambda(1-\rho)}$
- $\Longrightarrow E(T) = \frac{1}{\mu-\lambda}$

Hàng đợi M/M/1 (5)

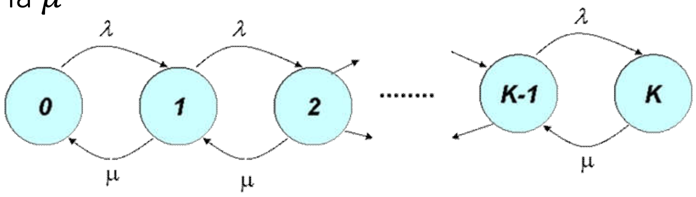
- *Thời gian phục vụ trung bình của khách hàng*
- $E(T_S) = \frac{1}{\mu}$
- *Thời gian trung bình của khách hàng trong hàng đợi*
- $E(T_Q) = E(T) - E(T_S) = \frac{1}{\mu-\lambda} - \frac{1}{\mu}$
- $\Longrightarrow\ E(T_Q) = \frac{\lambda}{(\mu-\lambda)\mu}$
- $\Longrightarrow E(T_Q) = \frac{\rho^2}{\lambda(1-\rho)}$

Hàng đợi M/M/1 (6)

- *Số lượng trung bình của khách hàng trong hàng đợi*
- Theo định lý Little ta có
- $E(N_Q) = \lambda E(T_Q)$
- $\Rightarrow E(N_Q) = \frac{\rho^2}{(1-\rho)}$

Hàng đợi M/M/1/K (1)

- Tất cả các tốc độ đến đều là λ
- Tất cả các tốc độ đi đều là μ



- $p_i = \left(\frac{\lambda}{\mu}\right)^i p_0$
- Đặt $\rho = \frac{\lambda}{\mu}$ ta có $p_i = (\rho)^i p_0$

M/M/1/K(2)

- Ta có

$$\sum_{i=0}^{K} p_i = 1 \quad \Rightarrow \sum_{i=0}^{K} p_0\rho^i = 1 \quad \Rightarrow p_0 \sum_{i=0}^{K} \rho^i = 1$$

- Mà ta có

$$\sum_{i=0}^{K} \rho^i = \frac{1-\rho^{K+1}}{1-\rho} \qquad \Rightarrow p_0 = \frac{1-\rho}{1-\rho^{K+1}}$$

$$\Rightarrow p_i = \frac{(1-\rho)\rho^i}{1-\rho^{K+1}}$$

M/M/1/K(3)

- Số yêu cầu trung bình trong hệ thống:

$$N = \sum_{i=0}^{K} i p_i$$

$$\Rightarrow N = p_0 \sum_{i=0}^{K} i\rho^i \quad \Rightarrow N = p_0\rho \sum_{i=0}^{K} i\rho^{i-1}$$

$$\Rightarrow N = \frac{(1-\rho)\rho}{1-\rho^{K+1}} \sum_{i=0}^{K} i\rho^{i-1} \ (1)$$

M/M/1/K(4)

- Mà ta có

$$\sum_{i=0}^{K}\rho^i = \frac{1-\rho^{K+1}}{1-\rho}$$

- Đạo hàm cả 2 hai vế ta có

$$\sum_{i=1}^{K} i\rho^{i-1} = \frac{-(K+1)\rho^K(1-\rho)+(1-\rho^{K+1})}{(1-\rho)^2}$$

- Thay vào công thức (1)

$$N = \frac{(1-\rho)\rho}{1-\rho^{K+1}}\sum_{i=0}^{K} i\rho^{i-1} \ (1)$$

$$\Longrightarrow N = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}}$$

M/M/1/K(5)

- Khi ρ $\rightarrow 1$ *thì vì* $p_i = (\rho)^i p_0 \rightarrow \ p_i = p_0$
- *Vì* $\rightarrow \ p_i = \frac{1}{K+1}$
- Khi đó số yêu cầu trung bình trong hệ thống là

$$\sum_{i=0}^{K} p_i = 1$$

$$N = \sum_{i=0}^{K} i p_i = \frac{K}{2}$$

M/M/1/K(6)

- Số yêu cầu trung bình trong hàng đợi:

$$N_Q = \sum_{i=1}^{K} (i-1)p_i$$

$$\Rightarrow N_Q = \sum_{i=1}^{K} ip_i - \sum_{i=1}^{K} p_i \qquad \Rightarrow N_Q = N - (1-p_0) \;\; \Rightarrow N_Q = N - 1 + p_0$$

$$\Rightarrow N_Q = \frac{\rho}{1-\rho} - \frac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} - 1 + \frac{1-\rho}{1-\rho^{K+1}}$$

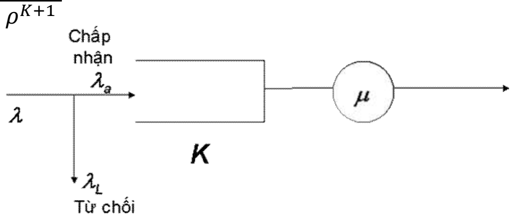
$$\Rightarrow N_Q = -\frac{1}{1-\rho} + \frac{1-\rho-(K+1)\rho^{K+1}}{1-\rho^{K+1}}$$

M/M/1/K(7)

- Xác suất để một yêu cầu bị từ chối

$$P_{loss} = p_K = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}}$$

- Tốc độ đến thực tế

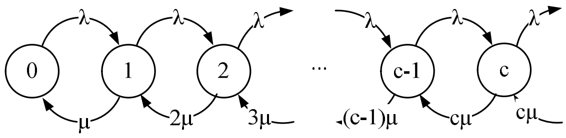


$$\lambda_a = (1-p_K)\lambda$$

M/M/1/K(8)

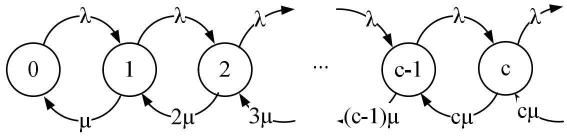
- Thời gian trung bình để một yêu cầu lưu lại trong hệ thống
  - $T = \frac{N}{\lambda_a} = \frac{1}{\lambda} \times \frac{1}{1-p_K} \times N$
- Thời gian trung bình của một yêu cầu trong hàng đợi
  - $T_Q = \frac{N_Q}{\lambda_a} = \frac{1}{\lambda} \times \frac{1}{1-p_K} \times N_Q$

Hàng đợi M/M/c (1)



- Tất cả các tốc độ đến đều là λ
- $p_1 = \left(\frac{\lambda}{\mu}\right) p_0$
- $p_2 = \left(\frac{\lambda}{2\mu}\right) p_1 = \frac{1}{2!}\left(\frac{\lambda}{\mu}\right)^2 p_0$
- $\rightarrow\ p_i = \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i p_0 \text{ với } i \le c$

Hàng đợi M/M/c (2)



- $Khi\ i \geq c\ thì$
  - $p_{c+1} = \left(\frac{\lambda}{c\mu}\right) p_c = \left(\frac{\lambda}{c\mu}\right) \frac{1}{c!} \left(\frac{\lambda}{\mu}\right)^c p_0$
  - $\rightarrow p_i = \left(\frac{\lambda}{c\mu}\right)^{i-c} \frac{1}{c!} \left(\frac{\lambda}{\mu}\right)^c p_0$
  - $\rightarrow p_i = \frac{1}{c!c^{i-c}} \left(\frac{\lambda}{\mu}\right)^i p_0$

M/M/c(3)

- Mà ta có

$$\sum_{i=0}^{\infty} p_i = 1$$

$$\sum_{i=0}^{c-1} p_i + \sum_{i=c}^{\infty} p_i = 1\ (*)$$

- Thay $p_i = \frac{1}{c!c^{i-c}} \left(\frac{\lambda}{\mu}\right)^i p_0$ vào $\sum_{i=c}^{\infty} p_i$ ta có

$$\sum_{i=c}^{\infty} p_i = \sum_{i=c}^{\infty} \frac{1}{c!\,c^{i-c}} \left(\frac{\lambda}{\mu}\right)^i p_0$$

M/M/c(4)

$$\text{Từ} \sum_{i=c}^{\infty} p_i = \sum_{i=c}^{\infty} \frac{1}{c!\, c^{i-c}} \left(\frac{\lambda}{\mu}\right)^i p_0 \ ta\ có$$

$$\rightarrow \sum_{i=c}^{\infty} p_i = \sum_{i=c}^{\infty} \frac{c^c}{c!} \left(\frac{\lambda}{c\mu}\right)^i p_0$$

Đặt $\rho = \frac{\lambda}{c\mu}$ $\quad \rightarrow \sum_{i=c}^{\infty} p_i = \sum_{i=c}^{\infty} \frac{c^c}{c!} \rho^i p_0$ $\quad$ Thay $i = j + c\ ta\ có$

$$\rightarrow \sum_{i=c}^{\infty} p_i = \sum_{j=0}^{\infty} \frac{c^c}{c!} \rho^{j+c} p_0 \quad \rightarrow \sum_{i=c}^{\infty} p_i = \frac{(c\rho)^c}{c!} p_0 \sum_{j=0}^{\infty} \rho^j$$

M/M/c(5)

- Mà ta có $\quad \sum_{j=0}^{\infty} \rho^j = \frac{1}{1-\rho}\ với\ \rho < 1$

$$\rightarrow \sum_{i=c}^{\infty} p_i = \frac{(c\rho)^c}{c!\,(1-\rho)} p_0\ (**)$$

- Thay $p_i = \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i p_0$ vào $\sum_{i=0}^{c-1} p_i$

$$\sum_{i=0}^{c-1} p_i = \sum_{i=0}^{c-1} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i p_0\ (***)$$

M/M/c(6)

- Thay (**) và (***) vào (*) ta có

$$\sum_{i=0}^{c-1} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^{i} p_0 + \frac{(c\rho)^c}{c!\,(1-\rho)} p_0 = 1$$

$$\rightarrow p_0 = [\sum_{i=0}^{c-1} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^{i} + \frac{(c\rho)^c}{c!\,(1-\rho)}]^{-1}$$

M/M/c(7)

- Xác suất xuất hiện hàng đợi

$$P_Q = \sum_{i=c}^{\infty} p_i$$

- Từ (**) ta có

$$\rightarrow P_Q = \sum_{i=c}^{\infty} p_i = \frac{(c\rho)^c}{c!\,(1-\rho)} p_0$$

Công thức Erlang C

$$\rightarrow P_Q = \sum_{i=c}^{\infty} p_i = \frac{1}{c!\,(1-\rho)}\left(\frac{\lambda}{\mu}\right)^{c} p_0$$

M/M/c(8)

- Số yêu cầu trung bình trong hàng đợi $N_Q = \sum_{i=c}^{\infty}(i-c)p_i$

Thay $i = j + c$ $ta\ có$ $N_Q = \sum_{j=0}^{\infty} j p_{j+c}$

- Từ $p_i = \frac{1}{c!c^{i-c}}(c\rho)^i p_0$ ta có $p_{j+c} = \frac{1}{c!c^j}(c\rho)^{j+c} p_0$
- $p_{j+c} = \frac{(c\rho)^c}{c!}\rho^j p_0$

$$\rightarrow N_Q = \frac{(c\rho)^c}{c!} p_0 \sum_{j=0}^{\infty} j\rho^j$$

M/M/c(9)

- Số yêu cầu trung bình trong hàng đợi (cont.)

$$Vì \sum_{i=1}^{\infty} \rho^{i-1} = \frac{1}{(1-\rho)^2}\ với\ \rho < 1\ thay\ vào \quad N_Q = \frac{(c\rho)^c}{c!} p_0 \sum_{j=0}^{\infty} j\rho^j\ ta\ có$$

$$\rightarrow N_Q = \frac{(c\rho)^c}{c!} p_0 \frac{\rho}{(1-\rho)^2}$$

- Mà ta có $\rightarrow N_Q = \frac{(c\rho)^c}{c!(1-\rho)} p_0 \frac{\rho}{(1-\rho)}$

$$P_Q = \frac{(c\rho)^c}{c!\,(1-\rho)} p_0$$

$$\rightarrow N_Q = P_Q \frac{\rho}{(1-\rho)}$$

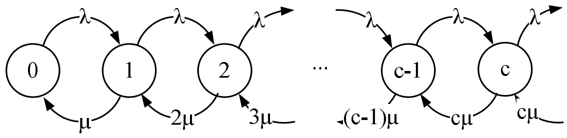
M/M/c (10)

- Thời gian trung bình của yêu cầu trong hàng đợi
- $T_Q = \frac{N_Q}{\lambda} = \frac{\rho P_Q}{\lambda(1-\rho)} = \frac{P_Q}{c\mu-\lambda}$
- Trễ trung bình của yêu cầu
- $T = \frac{1}{\mu} + T_Q = \frac{1}{\mu} + \frac{\rho P_Q}{\lambda(1-\rho)} = \frac{1}{\mu} + \frac{P_Q}{c\mu-\lambda}$

M/M/c (11)

- Số yêu cầu trung bình trong hệ thống
- $N = \lambda$T
- Thay T=$\frac{1}{\mu} + \frac{P_Q}{c\mu-\lambda}$ vào công thức trên ta có
- N=$\frac{\lambda}{\mu} + \frac{\lambda P_Q}{c\mu-\lambda}$= c$\rho + \frac{\rho P_Q}{1-\rho}$

Hàng đợi M/M/c/c (1)



- Tất cả các tốc độ đến đều là λ
- $p_1 = \left(\frac{\lambda}{\mu}\right) p_0$
- $p_2 = \left(\frac{\lambda}{2\mu}\right) p_1 = \frac{1}{2!}\left(\frac{\lambda}{\mu}\right)^2 p_0$
- $\rightarrow\ p_i = \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i p_0 với\ i \leq c$

M/M/c/c(2)

- Mà

$$\sum_{i=0}^{c} p_i = 1$$

$$\rightarrow\rightarrow p_0 \sum_{i=0}^{c} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i = 1$$

$$\rightarrow p_0 = [\sum_{i=0}^{c} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i]^{-1}$$

M/M/c/c(3)

- Xác suất tắc nghẽn
- $P_{tắc\ nghẽn} = p_c = \frac{1}{c!}\left(\frac{\lambda}{\mu}\right)^c p_0$
- Thay giá trị

$$p_0 = [\sum_{i=0}^{c}\frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i]^{-1} \text{ vào ta có}$$

- Công thức Erlang B

$$P_{tắc\ nghẽn} = \frac{\frac{1}{c!}\left(\frac{\lambda}{\mu}\right)^c}{\sum_{i=0}^{c}\frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i} \quad \rightarrow P_{tắc\ nghẽn} = \frac{\frac{1}{c!}\left(\frac{\lambda}{\mu}\right)^c}{1+\sum_{i=1}^{c}\frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i}$$

Hàng đợi có ưu tiên

ONE LOVE. ONE FUTURE.

Hàng đợi có ưu tiên(1)

- Trong hệ thống khách hàng có thể có các mức ưu tiên khác nhau.
- Ví dụ trong cửa hàng cắt tóc thì nam giới và phụ nữ là hai lớp ưu tiên khác nhau.
- Tại bưu điện sử dụng dịch vụ gửi thư và gửi bưu phẩm là hai mức ưu tiên khác nhau.

Hàng đợi có ưu tiên(2)

- Với mỗi mức ưu tiên:
  - Miêu tả của tiến trình đến
  - Miêu tả của tiến trình phục vụ
  - Số lượng server
  - Số lượng các vị trí đợi
  - Các quy tắc hàng đợi
  - Phản ứng của khách hàng khi bị trễ, tắc nghẽn, ...

Quy tắc phục vụ(1)

- FCFS (First Come First Served) nó thường được gọi là hàng đợi công bằng hay hàng đợi gọi và qui tắc này thường xuất hiện trong cuộc sống hàng ngày của chúng ta.
- Nó được xem như là FIFO, chú ý là FIFO chỉ sử dụng trong hàng đợi không sử dụng cho toàn hệ thống.

Quy tắc phục vụ(2)

- LCFS (Last Come First served) đó là chu trình ngăn xếp, như việc xếp hàng trên giá của cửa hàng.v.v
- Qui tắc này cũng xem như LIFO (Last In First Out)

Quy tắc phục vụ(3)

- SIRO (Sevice In Random Order) tất cả các khách hàng đang đợi trong hàng đợi có xác suất để được chọn phục vụ như nhau.
- Nó còn được gọi là RANDOM hay RS (Random Selection).

Quy tắc phục vụ(4)

- Nhận xét:
- Hai qui tắc đầu tiên chỉ sử dụng trong lần đến mà được xét,
- Qui tắc thứ 3 không được xem như tiêu chuẩn và không yêu cầu nhớ.

Quy tắc phục vụ(5)

- Liên quan đến thời gian phục vụ
- SJF (Shortest Job First): Việc đầu tiên ngắn nhất.
- SJN (Shortest Job Next): Việc tiếp theo ngắn nhất.
- SPF (Shortest Processing Time First): Thời gian xử lý đầu tiên ngắn nhất.

Quy tắc phục vụ(6)

- Có qui tắc bị ảnh hưởng bởi thời gian đến hoặc thời gian phục vụ. →thoả hiệp giữa các qui định
- RR (Round Robin): một khách hàng được phục vụ cho trong một khoảng thời gian cố định (Time slice). Nếu dịch vụ không hoàn thành trong khoảng thời gian này, thì khách hàng trở lại hàng đợi là FCFS.

Quy tắc phục vụ(7)

- FB (Foreground-Background):
- qui tắc này cố gắng thực hiện SJF mà không biết đến thời gian phục vụ sau này.
- Server sẽ cung cấp dịch vụ để khách hàng có thời gian phục vụ ít nhất.
- Khi tất cả các khách hàng có được thời gian phục vụ giống nhau, FB được xác định như là PS.

Quy tắc phục vụ(8)

- PS (Processor Sharing): tất cả khách hàng chia sẻ dung lượng dịch vụ bằng nhau.
- Qui tắc động : qui tắc hàng đợi phụ thuộc vào lượng thời gian sử dụng trong hàng đợi.

Thuật toán xếp hàng

- Xếp hàng vào trước ra trước (FIFO Queuing).
- Xếp hàng theo mức ưu tiên (PQ - Priority Queuing).
- Xếp hàng tuỳ biến (CQ - Custom Queuing).
- Xếp hàng theo công bằng trọng số (WFQ - Weighted Fair Queuing).

Xếp hàng vào trước ra trước(1)

- *FIFO Queuing*
  Trong dạng đơn giản nhất, thuật toán vào trước ra trước liên quan đến việc lưu trữ gói thông tin khi mạng bị tắc nghẽn và rồi chuyển tiếp các gói đi theo thứ tự mà chúng đến khi mạng không còn bị tắc nữa.
- FIFO trong một vài trường hợp là thuật toán mặc định vì tính đơn giản và không cần phải có sự thiết đặt cấu hình.

Xếp hàng vào trước ra trước (2)

- Nhược điểm:
  - FIFO không đưa ra sự quyết định nào về tính ưu tiên của các gói
  - FIFO không có sự bảo vệ mạng nào chống lại những ứng dụng (nguồn phát gói) có lỗi. Một nguồn phát gói lỗi phát quá ra một lưu lượng lớn đột ngột có thể là tăng độ trễ của các lưu lượng của các ứng dụng thời gian thực vốn nhạy cảm về thời gian.
- FIFO là thuật toán cần thiết cho việc điều khiển lưu lượng mạng trong giai đoạn ban đầu nhưng với những mạng thông minh hiện nay đòi hỏi phải có những thuật toán phức tạp hơn, đáp ứng được những yêu cầu khắt khe hơn.

Xếp hàng theo mức ưu tiên PQ(1)

- *PQ – (Priority Queuing)* :đảm bảo rằng những lưu lượng quan trọng sẽ có được sự xử lý nhanh hơn. Thuật toán được thiết kế để đưa ra tính ưu tiên nghiêm ngặt đối với những dòng lưu lượng quan trọng.
- PQ có thể thực hiện ưu tiên căn cứ vào giao thức, giao diện truyền tới, kích thước gói, địa chỉ nguồn hoặc địa chỉ đích

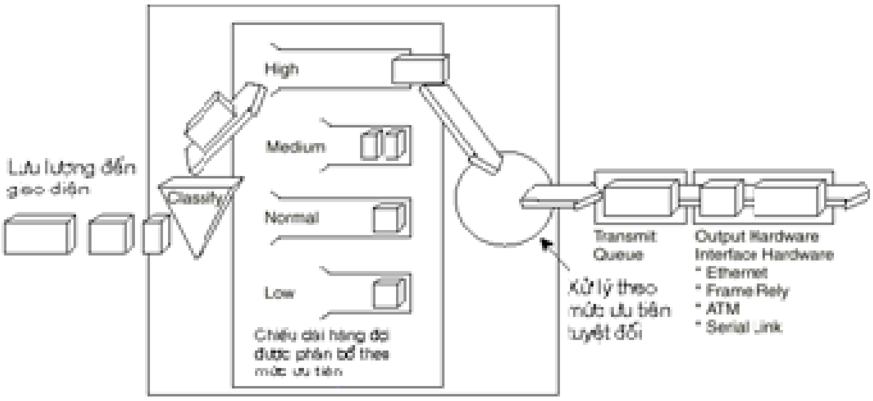
Xếp hàng theo mức ưu tiên PQ(2)

- PQ có thể thực hiện ưu tiên căn cứ vào
  - giao thức
  - giao diện truyền tới
  - kích thước gói
  - địa chỉ nguồn
  - địa chỉ đích

Xếp hàng theo mức ưu tiên PQ(3)

- Các gói được đặt vào 1 trong các hàng đợi có mức ưu tiên khác nhau dựa trên các mức độ ưu tiên được gán
- Các gói trong hàng đợi có mức ưu tiên cao sẽ được xử lý để truyền đi trước.

Xếp hàng theo mức ưu tiên PQ(4)



Xếp hàng theo mức ưu tiên PQ(5)

- PQ được cấu hình dựa vào các số liệu thống kê về tình hình hoạt động của mạng
- Nhược điểm: không tự động thích nghi khi điều kiện của mạng thay đổi.

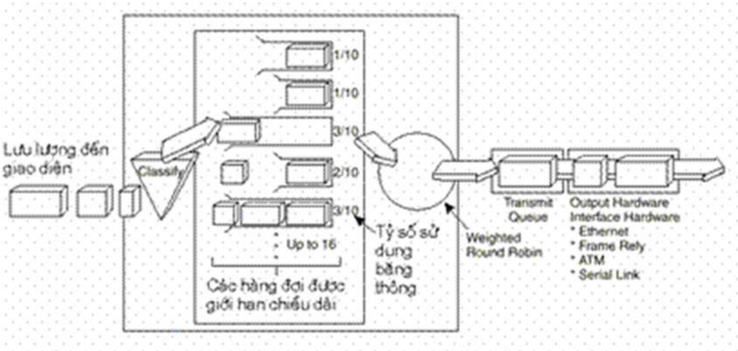
Xếp hàng tuỳ biến CQ (1)

- CQ *(Custom Queuing)*
  - Các ứng dụng khác nhau cùng chia sẻ mạng với các yêu cầu tối thiểu về băng thông và độ trễ.
  - Băng thông phải được chia một cách tỉ lệ cho những ứng dụng và người sử dụng.

Xếp hàng tuỳ biến CQ (2)

- CQ xử lý lưu lượng bằng cách gán cho mỗi loại gói thông tin trong mạng một số lượng cụ thể không gian hàng đợi và phục vụ các hàng đợi đó theo thuật toán round -robin (round-robin fashion).

Xếp hàng tuỳ biến CQ (3)



Xếp hàng tuỳ biến CQ (4)

- Nhược điểm: Cũng giống như PQ, CQ không tự thích ứng được khi điều kiện của mạng thay đổi.

Xếp hàng công bằng trọng số WFQ (1)

- *WFQ - Weighted Fair Queuing:* cung cấp được thời gian đáp ứng không đổi trong những điều kiện lưu lượng trên mạng thay đổi
- Thuật toán WFQ tương tự như CQ nhưng các giá trị tỷ lệ sử dụng băng thông gán cho các loại gói không được gán một số cố định bởi người sử dụng mà được hệ thống tự động điều chỉnh thông qua hệ thống báo hiệu Qos.

Xếp hàng công bằng trọng số WFQ (2)

- WFQ được thiết kế để giảm thiểu việc cấu hình hàng đợi và tự động thích ứng với sự thay đổi điều kiện lưu lượng của mạng.

Độ ưu tiên của khách hàng trong hàng đợi ưu tiên

- Khách hàng được chia thành p lớp ưu tiên. Khách hàng ở lớp ưu tiên k có độ ưu tiên cao hơn so với khách hàng ở lớp ưu tiên k+1. Hàng đợi ưu tiên lại đựoc chia thành các nhóm sau:
  - *Không ưu tiên phục vụ trước* (Non-preemptive hay là HOL - Head of the Line):
  - *Ưu tiên phục vụ trước* (preemptive):

Không ưu tiên phục vụ trước

- *Không ưu tiên phục vụ trước* (Non-preemptive hay là HOL - Head of the Line):
- Khách hàng đến với mức độ ưu tiên cao hơn so với khách hàng đang được phục vụ thì vẫn phải chờ cho đến khi server phục vụ xong khác hàng này (và phục vụ xong tất cả các khách hàng khác có mức độ ưu tiên cao hơn nó).

Ưu tiên phục vụ trước (preemptive)

- *Ưu tiên phục vụ trước:*
- Việc phục vụ khách hàng có quyền ưu tiên thấp sẽ bị ngừng lại khi có một khách hàng mà quyền ưu tiên của nó cao hơn đến hệ thống.

Ưu tiên phục vụ trước (cont.)

- Preemptive có thể chia thành các nhóm nhỏ sau:
  - Phục hồi ưu tiên (preemptive resume), khi mà sự phục vụ được tiếp tục từ thời điểm mà nó bị ngắt quãng trước đó.
  - Ưu tiên không lấy mẫu lại (preemptive without resampling), khi mà sự phục vụ bắt đầu lại từ đầu với khoảng thời gian phục vụ không đổi.
  - Ưu tiên lấy mẫu lại (preemptive with resampling), khi mà sự phục vụ bắt đầu lại với khoảng thời gian phục vụ mới.

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

HUST

hust.edu.vn fb.com/dhbkhn

TRÂN TRỌNG CẢM ƠN!