

Laboratori de Telemàtica 3

Exercicis previs de la pràctica 3

• Exercici 1

Les fórmules per a un sistema M/M/N són, tal com vam veure a XSSC, les següents:

$$\text{Probabilitat de demora: } P_d = \frac{\frac{(N \cdot \rho)^N}{n! \cdot (1 - \rho)}}{\sum_{k=0}^{N-1} \frac{(N \cdot \rho)^k}{k!} + \frac{N^N}{N!} \cdot \frac{\rho^N}{1 - \rho}}$$

$$\text{Factor d'utilització: } \rho = \frac{\lambda}{N \cdot \mu}$$

$$\text{Temps d'espera: } T_w = \frac{\lambda}{\mu \cdot (\mu - \lambda)}$$

$$\text{Nombre d'unitats al sistema: } N_w = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)}$$

Per al cas de l'exercici 2 tenim que el temps entre arribades és 10s, que vol dir que el nombre d'arribades per segon serà de 0,1 paquets/s. La utilització és del 75%, així fent ús de la fórmula del factor d'utilització trobem la taxa de servei, que és 0,134 paquets/s.

Calculem el temps d'espera que és de 22,5 segons i el nombre mitjà de paquets al sistema que és de 2,25 unitats.

Finalment la probabilitat de demora es redueix a l'expressió (N=1) següent:

$$P_d = \rho = 0,75 \equiv 75$$

• Exercici 2

Per a guardar valors de la forma en que ho fa Omnet (en el fitxer de vectors amb el nom i el timestamp corresponent) cal fer ús de la classe *cOutVector* (manual, pàgina 167). Es crearà una instància d'aquest objecte a la classe *Server* (o allà on sigui necessari) com a variable privada. En el constructor de la classe *server* s'inicialitzarà l'objecte fent ús del mètode *initialize* que pren com a paràmetre el nom de la variable (per al fitxer de vectors). Es pot veure com a la classe *Server* es fa això amb la variable *queueOccupancy*.

Omnet dóna la possibilitat de triar un fitxer de vectors diferent per a cada conjunt de variables i també la possibilitat d'especificar un temps de transitori (warm-up) on els valors de les variables són descartats.

Exercici 3

Per a realitzar una tongada de simulacions només cal compilar el programa fent ús de *cmdenv* (Command Line Environment) i activar l'*express mode* per defecte. D'aquesta manera no s'obrirà l'interfície gràfica de Omnet++ i directament s'engegarà la simulació de la xarxa.

Fent ús del shell script *mrunch* podem automatitzar molt el procés. Aquest script fa el següent:

- Genera *seeds* aleatoris per a cada un dels experiments.
- Prepara el fitxer de *seeds.ini* on afegeix el *seed* a fer servir i especifica el fitxer de vectors a usar (un diferent per cada experiment).
- Executa cada una de les proves fent ús d'un *seed* i un fitxer de sortida diferents.
- Guarda una còpia de les dades que ha emprat (per a poder reproduir una simulació)

Finalment cal filtrar la variable que ens interessa dels fitxers de vectors (com fèiem la pràctica 2) i per a fer-ho de forma automatitzada per a tots els fitxers emprarem el *fvector.sh* que simplement realitza el filtre *awk* per a tots els fitxers que li especifiquem.

• Exercici 4

```
function [mean,std] = time_weighted_mean(W)
    TD = W(2:size(W,1),1:1) - W(1:size(W,1)-1,1:1);
    raw_avg = TD'*W(1:size(W,1)-1,2:2);
    SM = TD'*power(W(1:size(W,1)-1,2:2),2);
    mean = raw_avg./W(size(W,1):size(W,1),1:1);
    std = sqrt(SM./W(size(W,1):size(W,1),1:1)- mean*mean);
end
```

• Exercici 5

```
function result = confidence_interval(path)
    % Afegir el / final
    if (path(size(path,2)) ~= '/')
        path = [path '/']
    end
    % Matriu de mitjes i desviacions
    result = zeros(size(fitxers),2);
    % Per cada fitxer calcular
    fitxers = dir(path);
    for i=1:size(fitxers);
        vector=load([path fitxers(i).name]);
        [mean,std] = time_weighted_mean(vector);
        % Guardar el resultat
        result(i,1) = mean;
        result(i,2) = std;
    end
end
```