

TAREA 3

Entrega: 8 de Julio de 2024

Resultados de Aprendizaje

- Crea modelos de simulación para procesos físicos de redes u otro que efectivamente capturen el funcionamiento de los procesos a simular
- Crea programas computacionales que evalúan numéricamente los modelos de simulación obteniendo resultados que se ajustan a resultados teóricos conocidos
- Utiliza técnicas de visualización de datos en lenguajes de propósito general que permitan apreciar cuantitativamente las variaciones de las variables observadas

Problema

Asignación de canales con usuarios finitos vs usuarios infinitos

- Al igual que en la tarea 1, suponga que existe un servidor cuya capacidad es igual a **C** (por ejemplo, sistema telefónico, servidor web, etc.), donde el número de usuarios está representado ahora por **M**.
- Suponga, que el proceso de llegada es aleatorio, y el tiempo de servicio es aleatorio.
- Las solicitudes de conexión (llegadas) y las salidas se generan en base a una variable aleatoria independiente e idénticamente distribuida de distribución exponencial con parámetro λ y μ respectivamente.
- No existe cola de espera (sin buffers).

¿Cuál es la probabilidad de que un usuario no pueda conectarse por congestión en el enlace?

Para evaluar la probabilidad de bloqueo del enlace en cuestión, programe un simulador Python (modificación simulador tarea 1), el cual represente el modelo definido del sistema.

Como salida de cada simulación, el simulador debe obtener:

- La probabilidad de bloqueo obtenida por el simulador, en base al número de llegadas totales de usuarios ejecutadas (LLEGADAS), cuya capacidad obtenga un valor en el orden de 10⁻³, de tal manera que se obtenga un error relativo del 5% con un intervalo de confianza de 95%. (Criterio de parada).

A continuación, considere el simulador de la tarea 1 con usuarios infinitos, y modifique su código de manera que el criterio de parada sea cumpliendo un intervalo de confianza (error relativo del 5% con un intervalo de confianza de 95%), y la distribución de las llegadas y salidas sea exponencial.

EL tráfico en el simulador de esta tarea se debe representar por un modelo ON-OFF, esto significa que mientras se encuentra el usuario en modo ON este se encuentra transmitiendo a su máxima tasa de bits, mientras que en modo OFF el usuario no transmite. El tiempo ON y OFF se encuentran representados por variables aleatorias exponenciales, cuyo tiempo medio de los períodos ON y OFF se denominan t_{ON} y t_{OFF} respectivamente:

$$\rho = \frac{t_{ON}}{t_{ON} + t_{OFF}}$$

, donde t_{ON} es igual a 0.001.

Para el tráfico de la Tarea 1, el simulador consideraba que todos los usuarios tienen la misma carga de tráfico ρ (tráfico homogéneo), es decir, los mismos valores medios t_{ON} y t_{OFF} .

$$\rho = \frac{t_{ON}}{t_{ON} + t_{OFF}} = \frac{\lambda}{\mu}; \quad \mu = \frac{1}{t_{ON}}; \quad \lambda = \frac{1}{t_{ON} + t_{OFF}}$$

Los simuladores deben incluir lo siguiente:

1. Definición de variables, estados y sistema (El código debe ser autocontenido y comentado con las explicaciones pertinentes).
2. Estado inicial del sistema. (No puede haber salidas si no hay llegadas)
3. Gestión de los eventos en la línea de tiempo, para así reconocer si los eventos son salidas o llegadas.
4. Cálculo de las métricas al finalizar la simulación. (probabilidad de bloqueo)

Declare un método para definir el avance de tiempo (scheduler). En este sentido puede aprovechar las funciones de python para crear y utilizar una línea de tiempo, o cree un medio propio para representar el avance de tiempo de la simulación. Los simuladores deben poder ejecutarse desde Jupyter Notebook, para así desplegar los datos y su posterior análisis, basándose en las librerías de visualización de datos de Python (Idee como hacerlo).

En Jupyter Notebook se hará la comparación de los resultados de ambos simuladores (usuarios finitos e infinitos). El simulador de usuarios infinitos se debe ejecutar una vez para una **carga de tráfico media igual a 25**, cuya capacidad es la suficiente como para obtener una probabilidad de bloqueo de 10^{-3} . Mientras que el simulador de usuarios finitos se debe ejecutar 10 veces, con los valores de $M = 30i$ usuarios, con $i = \{1, 2, 3, \dots, 10\}$, donde cada usuario tiene una carga de tráfico igual a $\rho = \frac{25}{M}$, para la misma capacidad del de usuarios infinitos. En base a lo anterior se debe incluir el siguiente gráfico utilizando las bondades de Python y Jupyter Notebook.

1. Un gráfico que indique las probabilidades de bloqueo (eje Y) por número de usuarios M (eje X) para el caso de usuarios finitos. A este gráfico se le debe agregar una línea constante con el valor obtenido por el simulador de usuarios infinitos.
2. Responder, ¿Cuántos usuarios debe tener el sistema para considerar que asintóticamente se comportan como un sistema con infinitos usuarios? Si el número de usuarios definidos por el criterio previamente propuesto no es suficiente, siga aumentando el número de usuarios.

La asignación es individual, donde se debe entregar a través de aula un archivo comprimido denominado de la siguiente forma "TEL341 Nombre Apellido T3". Este archivo comprimido debe contener todo lo necesario para su evaluación, es decir, el Notebook de Jupyter con los archivos asociados para su ejecución, el código en Python y los gráficos obtenidos.