

Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática

CI-1453 INVESTIGACIÓN DE OPERACIONES

Grupo 01

Prof. Ileana Alpízar

***Documentación del
Proyecto de simulación***

Elaborado por:

César Vargas Arce,
Bryan Sibaja Granados,

II Semestre 2016

Manual de instalación

Para correr el programa se necesita un windows de 64-bits y al correrlo de momento se despliega una corrida de la simulación.

Descripción del sistema a simular

Suponga que hay una red de 3 servidores para la recepción y envío de mensajes de correo electrónico (mensajes de ahora en adelante), así como para la recepción y envío de archivos separados en paquetes. Esta red funciona así:

1. La computadora o servidor No.1, recibe mensajes y paquetes desde fuera de la red. En esta computadora de varios procesadores o núcleos, **un proceso** siempre activo (corriendo en uno de sus procesadores o núcleos), recibe los mensajes, cuyo tiempo entre arribos tiene distribución normal con una media de **4 segundos** y una **varianza de 0.01 segundos cuadrados**. El tiempo que tarda este proceso receptor **enviando el mensaje a la cola de mensajes, o a alguno de los procesos encargados de atenderlos**, si es que alguno está libre, se asumirá como 0 para efectos de simplificación.

Otro proceso, activo siempre, se encarga de **recibir los paquetes o partes de archivos** que llegan a este servidor con un tiempo entre arribos **uniforme entre 3 y 7 segundos**. Este proceso enumera cada paquete recibido en secuencia de 0 a 100 (reiniciando luego en 0), y lo coloca en la cola, en la ventana si corresponde (esto se explicará más adelante), o de una vez con **el proceso encargado de atenderlo** si alguno de los dos procesos está libre-estos son los mismos procesos que atienden a los mensajes-. El proceso que recibe los paquetes tarda, para efectos prácticos, 0 segundos.

Los paquetes son enviados desde el servidor **No.1** al **No. 3**, el cual es el encargado de almacenar los archivos completos. Cada paquete lleva información de a qué archivo pertenece y su número de paquete, de un total de paquetes para cada archivo, **independientemente del número de secuencia** indicado anteriormente.

Los dos procesos que se encargan de atender tanto a los mensajes como a los paquetes trabajan de la siguiente manera:

cuando se atiende **un mensaje**, se prepara para ser enviado y se realiza la transferencia de este a la línea de transmisión (se pone bit por bit) . El tiempo que se tarda en esto tiene distribución de probabilidad cuya función de **densidad es: $f(x) = 2x/3$** con x entre 1 y 2 segundos. Desde el momento en el que se puso el último bit en la línea, el tiempo que tarda en llegar este último bit al servidor No.2 (**tiempo de propagación**) **es de 3 segundos**.

cuando se atiende **paquete**, también debe "armarse" para ser enviado y se realiza la transferencia de este a la línea de transmisión (se pone bit por bit). El tiempo promedio que se tarda en esto es de **1/2 segundo, distribución exponencial**. Cada paquete tiene un tiempo de propagación de 2 segundos y tiene una probabilidad **0.05 de perderse**. Es por esto último que cuando se envía un paquete, este no se desecha de la cola hasta que se sepa si se recibió en el servidor No. 3. Este servidor 3, cuando recibe correctamente un paquete, el número i, por ejemplo, le envía al ser No. 1 un ACK (acknowledge) con el

número de paquete que espera recibir luego del i , es decir envía un **ACK = $i+1$** . Como es un poco tedioso esperar siempre un ACK antes de enviar el siguiente paquete, lo que se hace es enviar, sin esperar ACK, un máximo de 4 paquetes. **A estos primeros 4 paquetes de la cola, se le llama la "ventana"**, la cual se va corriendo al ir desechando los que ya se sabe que llegaron al ser. **No.3**

Otro proceso en este servidor se encarga de ir **recibiendo los ACK** que llegan desde el No. 3. Al llegar un ACK, este proceso verifica que se trata de la confirmación de que llegó el primero de la cola, en ese caso, **corre la ventana un campo**, y si había un paquete ahí y si uno de los procesos encargados de hacer envíos está libre, comienza a prepararlo para el envío. **Si el ACK no es el correcto**, es decir, pide que se envíe el paquete que está de primero en la 2 cola, significa que este se perdió cuando se envió anteriormente y **el servidor No. 3 no aceptará los siguientes que le lleguen sin la secuencia correcta**, entonces el servidor No. 1 debe volver a enviar los mensajes de la ventana, comenzando con el primero. Por supuesto que este proceso tarda **0 segundos** por cada ACK que recibe.

2. La computadora o servidor No.2 es el encargado de enviar a su destino los mensajes que recibe del **No.1** Acá hay un proceso (del que asumiremos para esta simulación que tarda 0 segundos) que se encarga de ir recibiendo los mensajes e ir poniéndolos en cola, o, si el proceso que los atiende está libre, pasarlos a servicio. Cuando se atiende un mensaje, se revisa si viene con algún error, en ese caso se devuelve al servidor 1 (Note que en el servidor 1 debe haber un proceso aparte del que recibe los mensajes nuevos, que es el que recibe estos desde el 2), esto ocurre con **el 20% de los mensajes que recibe** , y ocurre igual con los que llegan por primera vez a la computadora 1, o con los que han sido devueltos desde la 2 por n -ésima vez. Al resto de los mensajes los envía a su destino. En esta máquina, el proceso que se encarga de atender a los mensajes, **tarda un promedio de 1 segundo, (t. exponencial)**, ya sea que los envíe a su destino, o que los devuelva a la computadora 1. Acá no preocupa si desde la No. 1 a la 2 se perdió algún mensaje. El tiempo de propagación de un mensaje devuelto a la comp 1, es de **3 segundos**.

3. La computadora o servidor No. 3 recibe los paquetes que le envía el servidor No.1 para almacenarlos en disco formando los archivos de los que son parte.

Como los paquetes cuando son transmitidos pueden perderse con una **probabilidad de 0.05**, el proceso que los atiende, debe verificar que le llegan en secuencia correcta. Para ello, cada vez que atiende un paquete correcto, suponga que se trata del paquete enumerado como i , lo envía almacenarse en disco y le devuelve a la computadora **No.1**, una señal (**ACK**) indicando que va bien la secuencia y que ahora espera el número de paquete $i+1$. Si se esperaba el paquete i , pero en su lugar le corresponde atender a un paquete con un **número de secuencia mayor, desecha este paquete** y le devuelve a la computadora No.1 una señal indicando que sigue esperando el paquete i (**ACK**) Suponga, para simplificar el proyecto, que los **ACK no se pierden en la línea**, que su tiempo de propagación es de **2 segundos**. El proceso encargado de atender cada paquete, tarda un tiempo con distribución normal, en segundos **$n(\mu=1.5, \sigma^2=0.012)$**

Para el caso en el que se pierdan todos los paquetes de la ventana que enviara el servidor No. 1, en este servidor 3 debe establecerse un **"timer" de 20 segundos**, para que si en ese tiempo no se recibe paquete alguno, se reenvíe el ACK correspondiente al paquete que se

espera. Suponga que el tiempo que se tarda preparando y poniendo en la línea el ACK es igual al que se tarda atendiendo un paquete.

Pseudocódigo

Eventos:

Arribo de paquete a la comp_1:

(Si se recibe un paquete)

Enumerar el paquete

Colocar el paquete en la cola de paquetes de la comp_1

(Si el tamaño de la cola < 2 && (procesos está libre))

enviar el paquete a la comp_3

Arribo de mensaje a la comp_1

(Si se recibe un paquete)

(Si el tamaño de la cola < 2 && (procesos está libre))

enviar el mensaje a la comp_3

(si no)

Colocar el mensaje en la cola de mensajes de la comp_1

Se libera servidor de la comp_1 (hay 2 servidores)

(Si la cantidad de servidores libres > 0)

(si se quiere enviar un mensaje)

Se envía el mensaje y se borra de la cola de mensajes

Se disminuye la cantidad de servidores libres

(si se quiere enviar un paquete)

Se envía el paquete

Se disminuye la cantidad de servidores libres

Llega un ACK a la comp _1

(Si el número del ack == al primero de la ventana)

se envía la ventana

(Si no)

Se corre la ventana hasta que el primero sea igual al numero del ack

Y se envía la ventana

Llega un mensaje dañado de la comp_2 a la comp_1

(si llega un mensaje dañado)

Se pone en la cola de mensajes

Llega un mensaje a la comp_2

(llega un mensaje a la comp_2)
 (si la cola de msj de la comp_2 está vacía)
 Procesar el mensaje
 (si no)
 Se pone en la cola de msjs de la comp_2

Se libera el servidor de la comp_2

 Se obtiene la probabilidad de que el mensaje traiga errores
 (si la probabilidad es < 20)
 se devuelve el msj a la comp_1
 (si no)
 se enva el msj a su destino

Llegó un paquete a la comp_3

 Se pone el paquete en la cola de la comp_3
 (si la cola de paquetes de la comp_3 < 2)
 (si el paquete está en la secuencia correcta)
 (si el proceso está libre)
 En envía el paquete a disco
 Se modifica el próximo paquete a esperar
 Y se envía un ack con el próximo paquete a esperar
 (si no)
 se envía un ack con el paquete que se espera

Se libera el servidor de la comp_3

 (si hay paquetes en la cola de la comp_3)
 (si el paquete a procesar está en la secuencia correcta)
 En envía el paquete a disco
 Se modifica el próximo paquete a esperar
 Y se envía un ack con el próximo paquete a esperar
 (si no)
 se envía un ack con el paquete que se espera

Se vence el timer

El timer está en la comp_1, y se vence cuando no se recibe un ack por mucho tiempo

 (si se vence el timer)
 Se envía la ventana a la comp_3

Estadísticas obtenidas:

Estadísticas obtenidas: impresión de la pantalla con las estadísticas obtenidas para los valores de prueba:

correr la simulación 10 veces, correrlo por 2000 segundos cada vez.

Análisis del sistema

No se pudo efectuar un análisis adecuado debido a que la simulación presenta fallos.

Problemas NO RESUELTOS

El problema principal se encuentra en los eventos de liberación de servidores de la computadora número 1. Esto ocurre ya que algunas condiciones se encuentran mal definidas y no se puede programar en el tiempo los eventos de llega paquete a la computadora número tres ni el de llega mensaje a la computadora número 2. La ejecución de esos dos eventos es fundamental para poder realizar el análisis del sistema. La solución de este problema consiste en redefinir las condiciones para que ocurran los eventos de forma adecuada.