1. Paso de testigo en anillo (Token ring). Diseñe una red unidireccional en anillo, donde existe un conjunto de procesos $P_1..P_6$, y un canal de comunicaciones de cada proceso P_i al proceso siguiente, P_{i+1}. P_6, como último proceso del anillo, estará conectado a P_1. Un mensaje `token` circula por esta red. Cuando un proceso P_i recibe el mensaje `token` de su vecino anterior, puede:

- Enviar el mensaje `token` a su vecino siguiente, o
- Enviar un mensaje `datos(d,t)` a su vecino siguiente; donde `d` es el proceso de destino (`d` está en el rango 1..n, y d ≠ i) y `t` es una cadena de caracteres que representa los datos a enviar.

En el segundo caso, el mensaje `datos(d,t)` da una vuelta completa al anillo. Cuando el mensaje alcanza al proceso P_d, el proceso copia los datos `t` del mensaje. Cuando el mensaje `datos(d,t)` alcanza al proceso P_i que lo envió, dicho proceso envía el mensaje `token` a su vecino siguiente.

Asuma que en el estado inicial de la red, un estímulo externo hace que el proceso P_0 genere el mensaje `token`.

- Desarrolle primero el problema, asumiendo que no existen errores de comunicación.
- Considere luego qué errores de comunicación se pueden presentar, y con base en sus consideraciones, rediseñe el protocolo para que sea tolerante a errores.

2. Votación tolerante a errores

Considere una red consistente en un arreglo de procesos $P_1..P_5$ y un proceso `c`. La red trabaja por rondas, de la siguiente manera: En cada ronda, el proceso `c` envía un mensaje solicitud a cada uno de los procesos P_i y espera recibir un mensaje `respuesta(v)` de cada uno de ellos, donde `v` puede ser TRUE o FALSE. Si en la ronda `n`, `c` recibe un mensaje `respuesta(TRUE)` de todos los P_i, asigna un valor de 1 a `votos[x]`; pero en caso de llegar a recibir al menos un mensaje `respuesta(FALSE)` de alguno de los P_i, asigna un valor de 0 a `votos[x]`. Si entre rondas llega un mensaje `encuesta` (del ambiente) al proceso `c`, éste debe contar el número de casillas iguales a cero e iguales a uno en el arreglo `votos`, y comunicar estos resultados al ambiente.

Diseñe uno de los procesos P_i y el proceso `c` en SDL, bajo el supuesto de que pueden ocurrir errores de transmisión.

ANÁLISIS CÓDIGO TCP/IP

El objetivo de este trabajo es estudiar un aspecto particular de la implementación de la pila TCP/IP en un sistema operativo real. Para ello se empleará la comparación entre Xinu (descrito por Douglas Comer en el segundo libro de su serie sobre el protocolo TCP/IP) y la implementación de Linux, en su última versión de kernel.

Como bibliografía se sugieren los libros de Comer, Cesati y Benvenuti (ver bibliografía del curso), además de abundante material disponible en la Internet.

Son 11 trabajos, que se desarrollarán en forma individual:

Cap 6, Tabla y algoritmo de enrutamiento en IP (Pedro Pablo Mora)
Cap 7, Fragmentación y reensamble en IP (Alejandro Hurtado)
Cap 8, ICMP (Julieth López)
Cap 9, IGMP (Darío Murillo)
Cap 10, UDP (Leonardo Vargas)
Cap 11, Estructuras de datos y entrada en TCP (Vanessa Zorrilla)
Cap 12, Implementación de la máquina de estados de TCP (Paola Camargo)
Cap 13, Procesamiento de la salida en TCP (Tatiana Minnig)
Cap 14, Manejo de timers en TCP (Saint Jaramillo)
Cap 15, Control de flujo y retransmisión en TCP (Juan Camilo Restrepo)
Cap 17, Interfaz de nivel de sockets (Juan Manuel Eschebach)

Nota: El número de capítulo (cap) indica el capítulo del tomo 2 de Comer que debe consultarse.

El trabajo deberá tratar los siguientes puntos:

- Importancia del módulo estudiado en el funcionamiento de TCP/IP
- Estructuras de datos empleadas (en Comer y en Linux)
- Funciones y/o procedimientos que se emplean, y su relación con las estructuras de datos (en Comer y en Linux)
- Comparación entre los dos enfoques
- Conclusiones

El entregable de este trabajo será un resumen ejecutivo de máximo 10 páginas. Este material será subido a Moodle, para que todos los estudiantes puedan consultarlos.

Disponible en: Monday, 27 de April de 2009, 14:00
Fecha de entrega: Thursday, 18 de June de 2009, 12:00

PRESENTACIONES Y PROYECTOS DE PROTOCOLOS DE APLICACIÓN

El presente trabajo consiste en hacer una presentación y análisis de un protocolo de nivel de aplicación de la suite TCP/IP. Los turnos de exposición serán los siguientes:

Día 1 (Lunes 7 de abril):
Turno 1: Protocolo HTTP (Tatiana Minnig, Juan Camilo Restrepo)
Turno 2: Protocolo POP3 (Darío Murillo, Juan Manuel Eschebach)
Turno 3: Protocolo MSN Messenger (Leonardo Vargas, Saint Jaramillo)

Día 2 (Jueves 10 de abril):
Turno 1: Protocolo TELNET (Pedro Pablo Mora)
Turno 2: Protocolo FTP (Vanessa Zorilla, Paola Camargo)
Turno 3: Protocolo SSH (Julieth López, Alejandro Hurtado)

Cada exposición deberá tratar lo siguiente:

- Breve descripción del propósito del protocolo
- Máquina de estados del protocolo y comandos principales
- Captura de una sesión del protocolo con Wireshark, mostrando el diálogo entre cliente y servidor
- Conclusiones

La exposición deberá ser de máximo 40 minutos; preferiblemente, de 30.
Debe entregarse al profesor una copia de la exposición (archivo Powerpoint). El material será subido a Moodle, para que todos los estudiantes puedan consultarlo.

Cada uno de los equipos de trabajo desarrollará también un proyecto de

software, relacionado con su exposición. Los requierimientos mínimos de los proyectos se describen a continuación.

Para el equipo de trabajo del protocolo POP3:
Cliente de correo electrónico. En este proyecto debe implementarse un programa cliente sencillo de correo electrónico. Debe emplear el protocolo SMTP para enviar el correo, y el protocolo POP3 para recibirlo. El programa debe ser capaz de enviar y recibir correos en texto plano, o por lo menos con un anexo (attachment).

Para el equipo de trabajo del protocolo MSN Messenger:
Cliente MSN Messenger. El objetivo de este proyecto es implementar un cliente sencillo para el sistema de mensajería MSN Messenger de Microsoft. El programa debe ser capaz de conectarse a los servidores de Messenger, obtener la lista de contactos, determinar el estado de los mismos, e iniciar una sesión de chat con un usuario activo.

Para el equipo de trabajo del protocolo SSH:
Cliente SSH. En este proyecto debe crearse un programa emulador de Terminal que emplee el protocolo SSH para establecer sesión remota con un equipo. El cliente empleará el protocolo SSH v2, y empleará únicamente autenticación por password.

Para el equipo de trabajo del protocolo FTP:
Servidor FTP. En este proyecto debe crearse un servidor sencillo para el protocolo FTP. El programa debe ser capaz de recibir conexiones de clientes FTP estándar, y soportar las operaciones básicas: Autenticación del usuario, cambio de directorio, transferencia de archivos desde y hacia el cliente, borrado de archivos, renombrado de archivos, transferencia en modalidad ASCII y binaria. También debe registrar información acerca de las operaciones que realice en un archivo de log, en formato de texto plano.

Para el equipo del protocolo HTTP:
Servidor Web elemental. El objetivo de este proyecto es crear un servidor que reciba conexiones HTTP y permita despachar páginas web sencillas (código HTML y gráficos) a un cliente web cualquiera. El cliente solamente soportará el método GET de HTTP; no manejará cookies, método POST ni CGIs. Se puede crear una aplicación multihilo, que permita manejar varias conexiones de clientes al mismo tiempo. Todas las operaciones del servidor deben ser registradas en un archivo de log, en formato texto plano.

Para el equipo del protocolo Telnet:
Cliente Telnet. En este proyecto debe crearse un programa emulador de terminal (estilo Hyperterminal) que emplee el protocolo Telnet para establecer sesión remota con un equipo. El emulador de identificarse ante el host como una terminal compatible con la VT100 de DEC, y entender las secuencias básicas para control de la pantalla (posicionamiento del cursor, borrado de la pantalla, atributos de video, etc.)

La fecha de entrega y sustentación de los proyectos es el 27 de mayo de 2008, a las 3 PM.

UNIVERSIDAD ICESI
FACULTAD DE INGENIERÍA
DISEÑO DE PROTOCOLOS
TEORÍA DE COLAS – TALLER #3

PRIMERA PARTE

En esta primera parte del taller, se empleará el multiplexor estadístico que se desarrolló en la clase anterior. Normalmente, no se simula un sistema cuando se conocen los valores "verdaderos" de desempeño, pero lo haremos aquí, para ver cómo la simulación estima dichos valores "verdaderos".

a. Calcule la cantidad de tiempo necesaria para que la fuente genere 5000 paquetes. Ejecute 10 corridas de simulación para este tiempo, sin emplear tiempo de estabilización (run-in). En una hoja electrónica, registre la demora media del paquete para cada una de las 10 corridas, así como el número de paquetes que se emplearon en cada corrida para calcular la media. Emplee la hoja electrónica para calcular la media y la varianza de las 10 demoras, y de los 10 números de paquetes. Compare con el resultado analítico obtenido en la clase pasada. Nota: No hay razón para preocuparse si no existe una buena coincidencia entre alguno de los resultados de una corrida y el valor analítico. Sin embargo, si el promedio de las 10 corridas difiere del resultado analítico en 5%, puede haber algún error en el modelo; revíselo para saber si es correcto.

b. Repita la parte a, pero esta vez calcule la cantidad de tiempo necesaria para que la fuente genere 5500 paquetes. Emplee un tiempo de estabilización equivalente a 500 paquetes.

c. Ahora, calcule la cantidad de tiempo necesaria para generar 50500 paquetes. Emplee un tiempo de estabilización equivalente a 500 paquetes. Ejecute 5 corridas de la simulación, y compare la media y la varianza de estas 5 corridas con la media y la varianza de los puntos a y b.

d. Corra una simulación de 500500 paquetes, y observe el comportamiento del sistema.

e. Resuma lo que ha aprendido hasta ahora acerca de la interpretación de resultados de la simulación, basándose en los resultados obtenidos en los puntos a, b c, y d del taller.

SEGUNDA PARTE

En esta segunda parte, se resolverán los problemas 4 y 6 (a y b) del taller #2, empleando simulación. Se dará crédito extra si se resuelve el problema 5.

NOTA: En el problema 4, calcule la probabilidad inicial de bloqueo mediante simulación, y luego vaya agregando líneas telefónicas hasta alcanzar la meta propuesta.

Se debe preparar un reporte escrito de la solución de este taller, y presentarlo el lunes 24-OCT-05. No hay formato específico para el reporte, pero debe hacerse bien organizado, conciso y completo.

No se ha intentado realizar esta tarea

Por este enlace debe subirse el informe correspondiente al examen final de la materia. Las asignaciones de tema son las siguientes:

- Pedro Pablo Mora: Token Ring
- Juan Camilo Restrepo: Switching en LANs
- Tatiana Minnig: Diseño de redes
- Julieth López: Diseño de redes
- Juan Manuel Eschebach: ATM
- Darío Murillo: RIP
- Alejandro Hurtado: OSPF
- Vanessa Zorrilla: Control de congestión en TCP
- Paola Camargo: Disciplinas de encolamiento
- Leonardo Vargas: RSVP
- Saint Jaramillo: Firewalls y VPNs

Disponible en: Monday, 15 de June de 2009, 08:00
Fecha de entrega: Thursday, 25 de June de 2009, 12:00

Moodle Docs para esta página

09660_1