



FIUBA

75-43 Introducción a los Sistemas Distribuidos

75-33 Redes y Teleprocesamiento I

1° Cuatrimestre 2004

Trabajo Práctico: Protocolo de Capa de Red

Prof. María Feldgen

Requerimiento general del trabajo práctico.

El trabajo práctico es la implementación de un protocolo de capa de red o enlace utilizando programación con formularios CGI, en grupos de 2 personas, según el detalle de asignación en la página de la materia.

La implementación del protocolo debe seguir las siguientes pautas:

1. La interacción con los programas comienza desde una página general en la cual se encuentran todos los trabajos de todos los grupos. Cada grupo tiene 2 renglones. El primer renglón contiene una descripción y un botón que da control al primer programa que muestra la primera página que contiene el formulario. El segundo renglón contiene una descripción y un botón, y da control al programa que reproduce cualquier secuencia de acciones que se haya realizado con el protocolo. (Ver trabajos de cuatrimestres anteriores). Ejemplo de como es una pantalla inicial de los trabajos de los cuatrimestres anteriores

Grupo	Trabajo
Juan López, Ana Perez y Oscar Garcia	Protocolo SDLC (ABM) Disciplina de línea GO BACK-N con ventana deslizante entre 2 y 4. <input type="button" value="SDLC_1"/> Listado de frames intercambiados del Protocolo SDLC (ABM) <input type="button" value="Listar"/>
Julio Gonzalez, Luis Perez, Adriana Fernandez y José Garcia	Protocolo HDLC (ABM) GO BACK-N Con selección de ventana deslizante entre 2 y 4. <input type="button" value="Conectar"/> Protocolo HDLC (ABM) Selective Reject/Reject Con ventana deslizante entre 2 y 4 <input type="button" value="Comienzo"/>
Prof. María Feldgen	Protocolo STOP & WAIT Es una aplicación ejemplo de uso de un formulario con CGI <input type="button" value="Empezar"/>

2. La interacción con el/los programa/s debe permitir que la persona elija, por medio de un formulario, distintos tipos de unidades, los complete y se lo envíe al programa. El programa es la otra mitad del protocolo (en un sistema peer to peer), cumpliendo con las siguientes condiciones:
 - a) La persona es la que inicia la conexión y es la que termina la conexión siempre.
 - b) La persona debe completar las unidades adecuados en cada momento, sin ayuda del programa.
 - c) El programa debe verificar que las unidades correspondan a algún escenario posible de interacción y debe responder con alguna de las alternativas posibles (no debe ser siempre la misma).

- d) Se debe tomar en cuenta que el protocolo es full duplex, por lo que deben intercambiarse datos en ambos sentidos (el programa puede utilizar un archivo de texto para enviar información en secuencia).
 - e) Se debe tomar en cuenta que el protocolo es con ventana (SE RECUERDA QUE LA VENTANA MÍNIMA EN TODOS LOS PROTOCOLOS ASIGNADOS ES DOS (2)).
 - f) Se debe tomar en cuenta que todos los protocolos tienen números de secuencia y se debe respetar el protocolo.
 - g) Se deben implementar las unidades de datos con o sin piggyback dependiendo si el protocolo lo tiene o no. Y NO debe ser la única opción de intercambio de unidades.
 - h) Se deben implementar todas las unidades tanto para controlar el intercambio como para hacer el control de flujo necesario.
 - i) Se deben implementar las unidades de control correspondientes a la conexión, negociación (sí tiene) y liberación de la conexión y rechazo de unidades erróneas. Para rechazar una unidad sin significado en la interacción del protocolo, se usará el rechazo del protocolo o se rechazará por pantalla (queda libre al implementador).
 - j) Cada interacción muestra que envió la persona y cual es la respuesta del/los programa/s correspondiente. La interacción va a determinar como sigue el diálogo, NO SE DEBE INDICAR CUAL ES LA ACCIÓN QUE DEBE TOMAR LA PERSONA A CONTINUACIÓN.
3. Para el/los programas que reproducen las interacciones guardadas, debe permitir elegir el archivo o alguna de las conexiones desde el comienzo hasta el fin de la conexión. Debe permitir elegir una conexión de las que se hicieron de una lista y mostrar cada intercambio en forma similar a cuando fue realizado, de a uno o de a varios renglones (no una sabana, sino que abarque una cantidad razonable de renglones a mirar en una pantalla).
4. Deben tener en cuenta que deben guardar el estado actual, para controlar números de secuencia y uso de la ventana para una secuencia de interacción desde que se abre la conexión hasta que se cierra.
5. Todos los protocolos tienen campos con datos binarios, la persona y el programa completan esos campos en formato "humano" (NO en binario). Para los códigos de comandos se sugiere usar los acrónimos de los comandos. Para los números de secuencia u otros números, se usará el esquema decimal normal. No se toma en cuenta el CRC. Se deben usar todos los campos incluso los de dirección que tengan un significado y uso en el protocolo.
6. Escenarios posibles:
- 1. La persona ingresa una unidad de datos solamente (con la ventana vacía) y el programa contesta con una unidad de control:
 - a) con un ACK esperando el próximo
 - b) haciendo control de flujo y cerrando la ventana (la persona solo puede hacer ACK's, hasta que el programa abra la ventana)
 - c) con una unidad de datos con piggyback si lo tiene o sin él.
 - d) con varias unidades de datos, tantos como admite el espacio disponible en la ventana.
 - 2. La persona ingresa varias unidades de datos (tantos como permite la ventana del momento) y el programa contesta con:
 - a) con el ACK del último recibido
 - b) con el ACK del primero o de alguno anterior al último
 - c) con un rechazo por fuera de secuencia (como si faltara el primero)
 - d) haciendo control de flujo y cerrando la ventana confirmando solamente el primero
 - e) ídem a o b pero con una unidad de datos
 - f) ídem a o b pero con varias unidades de datos, tantos como admite la ventana
 - 3. La persona ingresa unidades de control
 - a) se contesta con otra unidad de control
 - b) se envía una unidad de datos con o sin piggyback dependiendo del protocolo y si la persona no cerró la ventana
 - c) se envía mas de una unidad de datos ídem b.

4. Apertura y cierre de la conexión
5. Rechazo de unidades por errores o fuera de protocolo
6. Situaciones de errores de llenado de unidades (dependiente de la implementación).

Ver ejemplos de cuatrimestres anteriores.

Entrega del trabajo práctico:

1. En la fecha de la primera entrega, se deben completar los datos para incluir el trabajo en el menú de selección de los trabajos en el *servidor que se indique*, desde un **WWW browser** utilizando cualquiera de las otras máquinas: la URL correspondiente es la **dirección del server/isd041rg.htm**, seguir las instrucciones y completar los datos correspondientes a su grupo. Una vez completados deben avisar a la profesora para que active el grupo en el servidor.
2. Cada grupo utilizará una máquina en la cual debe configurar el Apache, para hacer las pruebas, generando una página inicial (solo al efecto de las pruebas) que invoque al primer programa de cada renglón indicado antes. Recuerde las máquinas son públicas, no se garantiza que permanezca igual de un día al otro.
 - a) En el servidor definitivo, cada grupo tendrá un usuario en el *servidor*, llamado **isd041nn**, donde **nn** es el número de grupo (número de orden en la tabla de asignación de trabajos). A través de este usuario cada grupo puede instalar el trabajo práctico que debe cumplir con las siguientes condiciones:
 - b) los ejecutables de los programas deben tener el sufijo **cgi**
 - c) todas las referencias a archivos deben ser relativas, NO se aceptan referencias completas
 - d) los programas y los archivos que utilizan esos programas deben copiarse al subdirectorio **cgi-bin** dentro de su usuario (que es un link al **cgi-bin/isd041/isd041nn**, del *web server*) y así cada grupo es independiente de los demás, con respecto a nombres de archivos y/o programas
 - e) gráficos, imágenes, logos, etc., que sean parte de las páginas de presentación deben almacenarse en el subdirectorio **html** (que es un link al **directorio principal/isd04/isd041nn** del *web server*) y así también se mantiene la independencia entre los grupos.
 - f) se pueden crear subdirectorios dentro del **cgi-bin** o **html**, en este caso, los permisos deben ser cambiados para dar acceso al web server, ídem se debe proceder con los archivos.
 - g) todos los archivos que se requieran para trabajar que no sean solo archivos de texto, deben crearse y cargarse en el LINUX.
 - h) en los directorios **html** y **cgi-bin** solo deben estar los archivos y programas que requiere el web server, todos los demás archivos que requieran para las pruebas, compilación etc., deben estar en el directorio principal del usuario o en cualquier otro subdirectorio creado por el grupo.
 - i) el manual y la explicación de los comandos deben ser accesibles desde el browser, mientras se opera con los programas, al igual que el informe con las explicaciones.
3. Se debe entregar el trabajo en formato informe en papel (NO MANUSCRITO) y en html. El informe en papel y los diskettes se deben entregar en un folio como se detalla mas adelante El informe (en papel y en html) debe contener:
 - a) Breve explicación de la función del protocolo
 - b) Interacción con la capa superior: Explique que servicios provee a su capa superior y que debe hacer la capa superior (limitaciones que debe cumplir, etc.)
 - c) Explique brevemente el protocolo, formatos y funciones.
 - d) Hacer un diagrama de interacción para determinar cuales secuencias de comandos son válidas y existen en una aplicación de este tipo:
 - i) indicando el servicio o función que desencadena la secuencia de acciones
 - ii) comando que se envía con sus parámetros mas relevantes
 - iii) servicio o acción que desencadena en el servidor (si corresponde)
 - e) Descripción de la implementación del protocolo simulando la interacción.

- f) Explicación de la configuración del Apache, Web server, para las pruebas, antes de instalarlo en el servidor definitivo.
 - g) Lista de archivos que usa cada programa y su función (1 renglón)
 - h) Manual de operación que indica como debe llenarse el mensaje para los distintos tipos de mensajes y que sea accesible desde los programas que simulan el protocolo.
 - i) Juego de prueba de una secuencia de comandos para probar el programa, accesible desde los programas que simulan el protocolo. (planteo de diferentes escenarios)-
4. Entrega del informe y diskettes en un folio conteniendo:
- a) La carátula debe contener: N° de grupo, N° de padrón y nombre y apellido de cada integrante del grupo; y nombre del protocolo.
 - b) Una explicación de cómo se deben instalar en el servidor y como se debe recompilar los programas (indique cual compilador usó), bibliotecas, etc.
 - c) Un diskette que contiene
 - i) un subdirectorio por cada subdirectorio del web server: **cgi-bin y html**, como mínimo y todos los subdirectorio que correspondan dentro de estos.
 - ii) un subdirectorio **progs** que contiene los fuentes de los programas y todo lo necesario para crear el trabajo práctico en el web server.

Recuerden: Para compilar en las máquinas nuevas:

- ◆ Para C los programas deben tener extensión *cpp*, y se compila con el *make* que automáticamente invoca al *g++* (*make nombre programa sin extensión*)
- ◆ Para Pascal: usar el *GPC*, programas con extensión *pas*. se compila:
gpc --automake programa.pas -o programa.cgi
- ◆ Todos los ejecutables deben tener extensión *cgi*.