

Introducción a las Redes de Computadores

Obligatorio 4 – 2010

Transporte Confiable sobre UDP

Introducción a las Redes de Computadoras – Transporte Confiable sobre UDP versión 2 (TCsUv2) – Segunda Parte

Facultad de Ingeniería
Instituto de Computación
Departamento de Arquitectura de Sistemas

Se desea finalizar la implementación del protocolo TCsU descrito en el obligatorio 3, llegando a obtener un protocolo de transporte confiable.

Nota previa - IMPORTANTE

Se debe cumplir íntegramente el “Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios”, disponible en <http://www.fing.edu.uy/inco/pm/uploads/Ense%flanza/NoIndividualidad.pdf>

En particular está prohibido utilizar documentación de otros grupos, de otros años, de cualquier índole, o hacer público código a través de cualquier medio (news, correo, papeles sobre la mesa, etc.).

Forma de entrega

La entrega debe realizarse mediante el formulario correspondiente en el sitio web del curso <http://www.fing.edu.uy/inco/cursos/redescomp/>. El sistema de entregas soporta múltiples entregas por grupo, llevando un histórico de las mismas. Se recomienda realizar una entrega vacía con tiempo, a los efectos de verificar que su sistema le permite entregar correctamente. Se considerará como válida la última entrega dentro del plazo asignado.

Se debe entregar un solo archivo 'ob4.tar.gz' que contenga los ítems descritos en el apartado **Entregable**. Dicho archivo deberá ser generado utilizando la herramienta GNU TAR y compresión gzip. Otros formatos (bz2, rar, zip, cab, jar, etc.) no son válidos y serán rechazados.

Fecha de entrega

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente antes del domingo 30 de mayo de 2010 a las 23:30 horas, sin excepciones. No se aceptará ningún trabajo pasada la citada fecha. En particular, no se aceptarán trabajos enviados por mail a los docentes del curso, ni entregados en medios magnéticos en el instituto.

Observaciones

La corrección se hará en la máquina virtual del curso tal como se distribuyó, sin cambios o agregados, por lo que debe probarse su funcionamiento en ese ambiente previo a la

entrega.

Descripción del problema

Se desea implementar un protocolo de transporte confiable. El protocolo garantiza que las TPDU (Transport Protocol Data Unit) serán entregados en su destino sin errores, sin pérdidas y en el mismo orden en que se entregaron a la entidad transporte en el origen.

El esquema general de funcionamiento es el propuesto en el obligatorio 3.

El protocolo debe permitir la conexión y desconexión de los equipos. Además debe asegurar los siguientes elementos:

1. La conexión debe realizarse entre un extremo pasivo que espera conexiones (*aceptarTCsU*) y un extremo activo que la inicia (*conectarTCsU*).
2. Comunicar datos en ambos sentidos.
3. Permitir el cierre de conexiones, en el formato especificado en la máquina de estados.
4. Respetar el orden de entrega. Los datos se deben entregar como un flujo continuo de bytes recibidos, independientemente de los mecanismos utilizados para la transmisión.
5. En caso de falla en la red, es responsabilidad del protocolo garantizar la entrega.
6. Descarte de datos recibidos en forma duplicada.
7. Los datos recibidos con errores en la transmisión deben ser reenviados.

El protocolo debe implementar las siguientes funciones, las que serán invocadas por los procedimientos que utilicen el TCsU, especificadas en el obligatorio 3:

- `int crearTCsU()`
- `int aceptarTCsU(unsigned short localUDPport)`
- `int conectarTCsU(unsigned short peerUDPport, struct in_addr peerIPaddr)`
- `int escribirTCsU(const void *buf, size_t len)`
- `int leerTCsU(void *buf, size_t len)`
- `int cerrarTCsU()`

Estas funciones habilitarán el uso de una sola conexión TCsU por vez, y las primitivas invocadas posteriormente a un *crearTCsU()*, harán referencia a ésta única conexión.

El protocolo deberá contar con:

- Un mecanismo go-back-N de ARQ (Automatic Repeat reQuest) con piggybacking (los ACKs también transportan datos).
- Deberá implementar una ventana de recepción (receive window) para control de flujo. Es decir, el extremo emisor debe recibir información sobre el espacio disponible en el buffer de recepción de forma similar a como lo hace TCP.
- Esta información será enviada por el extremo receptor en el campo "*ventana*" del cabezal de los ACKs o datos que envíe. La ventana de recepción será la cantidad de TPDU que pueden ser almacenadas en el buffer de recepción en el peor caso, o sea, cuando las TPDU transportan de 1400 bytes de datos (ésta es una diferencia con TCP). El protocolo no enviará más TPDU que las que quepan en la ventana de recepción.
- Cuando la ventana de recepción sea menor a 1400 bytes, el extremo emisor detendrá el envío de TPDU.
- Ambos extremos deberán mantener informado al extremo opuesto del estado de la ventana de recepción, enviando periódicamente TPDU, a modo de sonda, con dicha información (tal como lo hace TCP).

Se contará con un canal con distorsiones para pruebas, sobre el cual debe probarse el transporte confiable TCsU.

Se pide

Incluya en la API resultado del obligatorio 3, los nuevos requerimientos solicitados en el presente obligatorio. El resultado debe implementar un *Transporte Confiable sobre UDP versión 2 (TCsUv2)*.

Entregable

Se deberá entregar:

1.Documentación, con énfasis en el protocolo (descripción exhaustiva en idioma Español). Especificación de ARQ implementado, así como manejo de ventana de recepción implementada y pseudo-código del protocolo implementado.

2.Implementación:

- a)Código fuente (implementando el pseudocódigo).
- b)*Makefile* para la compilación del mismo.

Insumos

Se entregarán programas que invocan las funciones planteadas, y que linkeditados con la biblioteca creada, deberá realizar la aplicación indicada a través de TCsU.

Se entregará un emulador de red, que genere distorsión en la misma. Se plantea que la comunicación se realice a través de la interfaz *lo* (loopback), la que se identifica con la dirección IP *127.0.0.1*, o con el nombre *localhost* y un puerto cualquiera. Se provee de un script que configura la distorsión en la red, donde podrá incluirse retardo y jitter, pérdida, duplicación y daño a paquetes.

Para modificar los parámetros de distorsión se cuenta en el script con seis variables,

- INTERFACE Indica a que interfaz se le aplica la distorsión
- DELAY Retardo a aplicar en la red.
- LOSS Porcentaje de pérdida.
- REORDER Porcentaje de paquetes entregados fuera de orden.
- DUPLICATE Porcentaje de paquetes duplicados en la red.
- CORRUPT Porcentaje de paquetes corruptos en la red.

El script se invoca con parámetro up o down, para levantar o bajar la distorsión.

Nota

La documentación del obligatorio debe incluirse dentro del archivo de la entrega. La misma debe entregarse como un único archivo tipo PDF de nombre **informegrupoXX.pdf**. El mismo deberá respetar la numeración de secciones acorde a los requerimientos anteriores.

Es necesario que en el informe figuren el nombre y la cédula de identidad de cada integrante del grupo. En caso que esto no se cumpla el obligatorio no será corregido, con la consecuente pérdida del curso de sus autores. NO se aceptarán otros formatos de informe. (ver <http://www.universidad.edu.uy/odfpdf/>).