

# Introducción a las Redes de Computadores

## Obligatorio 3 – 2010

### Transporte Confiable sobre UDP

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

**Facultad de Ingeniería  
Instituto de Computación  
Departamento de Arquitectura de Sistemas**

*Durante los obligatorios 3 y 4 se desea implementar un protocolo de transporte confiable y orientado a conexión sobre el protocolo no orientado a conexión UDP, que denominaremos Transporte Confiable sobre UDP versión 2 (TCsUv2). En este obligatorio 3 nos concentraremos en el diseño e implementación de un soporte de red básico para el protocolo TCsUv2, buscando resolver los siguientes ítems: implementación del intercambio de unidades de información y la máquina de estados básica del mismo.*

#### **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 'ob3.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 lunes 10 de mayo de 2010 a las 00:30 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 a resolver en los obligatorios 3 y 4**

Durante los obligatorios 3 y 4 se desea implementar el protocolo de transporte confiable *TCsUv2*. Éste protocolo transmite datos recibidos como un flujo secuencial de bytes, entre dos aplicaciones utilizando un enlace implementado sobre UDP, devolviendo en el otro extremo el mismo flujo secuencial de octetos. La conexión se debe resolver en ambos sentidos.

El protocolo debe ser confiable y los datos serán entregados a la aplicación destino sin errores ni pérdidas y en el mismo orden en el que fueron entregadas por la aplicación de origen.

### ***Transporte Confiable sobre UDP versión 2***

El *TCsUv2* gestiona una conexión establecida entre dos aplicaciones. La gestión de la conexión implica implementar el establecimiento y cierre de conexión, así como el chequeo de pérdida de paquetes, solicitud de retransmisión de los mismos y control del flujo en caso de saturación de *buffers*.

El esquema general de funcionamiento propuesto para el protocolo, se presenta en la figura 1.

Las etiquetas de las transiciones entre estados tienen el siguiente formato: ***entidad\_IN:señal\_IN/entidad\_OUT:señal\_OUT***, donde *entidad\_IN* y *entidad\_OUT* indican respectivamente el origen y destino de la señal o señales (Aplicación o Capa de Red). Para limitar el tiempo de permanencia en un estado existe la señal ***TIME\_OUT***, que no necesita etiqueta *entidad\_IN*.

El protocolo debe permitir la conexión y desconexión entre dos aplicaciones. Además, la conexión debe cumplir con los siguientes requerimientos:

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 (figura 1).
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 de los datos perdidos durante la falla.
6. Descartar los datos recibidos en forma duplicada.

7. Descartar y notificar al extremo opuesto en caso de saturación de los buffers de recepción del TCsUv2 (control de flujo).

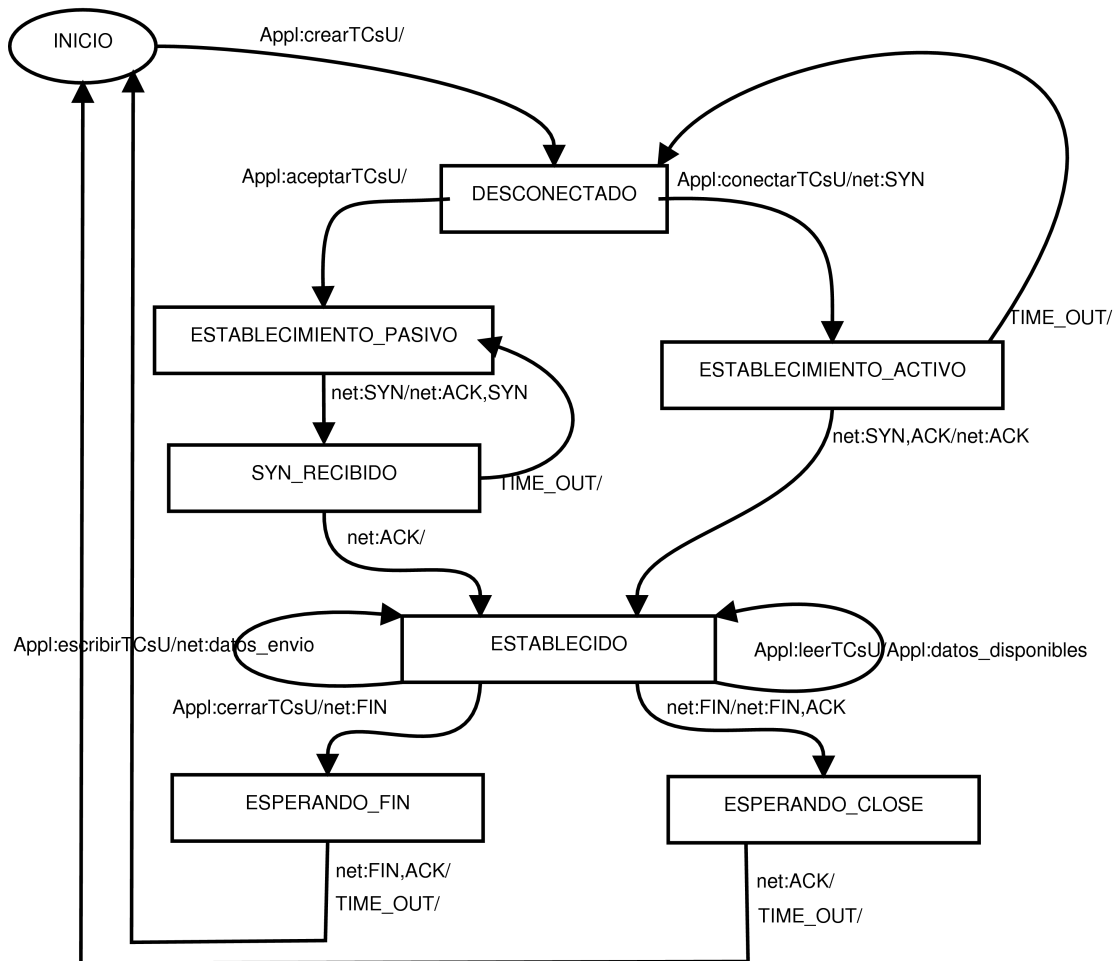


Fig.1 Máquina de Estados del protocolo TCsUv2

En éste obligatorio se pide la implementación de los primeros 3 puntos, y para una sola conexión. Asimismo, las estructuras de datos deben ser diseñadas e implementadas teniendo en cuenta los requerimientos 4 a 7 que serán implementados en el próximo obligatorio.

El protocolo debe implementar las siguientes funciones, las que serán invocadas por los procedimientos que utilicen el TCsUv2:

- **int crearTCsU()** Crea las estructuras de datos, y levanta los procedimientos necesarios para el control del *Transporte Confiable sobre UDP*. Devuelve un identificador (IdTCsU) mayor o igual a 0. En caso de error devuelve -1.
- **int aceptarTCsU(unsigned short localUDPport)** Queda esperando en forma pasiva la conexión de otro equipo a través del puerto UDP especificado (*localUDPport*). Esta función bloquea la aplicación hasta recibir una conexión. En caso de realizar en forma satisfactoria el procedimiento de conexión devuelve 0, en caso contrario devuelve un -1.
- **int conectarTCsU(unsigned short peerUDPport, struct in\_addr peerIPaddr)** Inicia en forma activa la conexión con otro equipo (*peerIPaddr*) a través del puerto UDP especificado (*peerUDPport*). Esta función bloquea la aplicación hasta establecer la conexión. En caso de realizar en forma satisfactoria

el procedimiento de conexión devuelve 0, en caso contrario devuelve un -1.

- **int escribirTCsU(const void \*buf, size\_t len)** Envía al otro extremo de la conexión, los datos pasados en *buf* de largo *len*. Esta función dependiendo de las posibilidades del transporte resuelve el envío de hasta *len* bytes pasados en la variable *buf*. En caso de no sea posible, ya sea por falta de buffers u otros motivos, intenta enviar la mayor cantidad de bytes. El procedimiento devuelve cantidad de bytes aceptados para envío, en caso de error devuelve un -1. En caso de no haberse establecido conexión devuelve -1.
- **int leerTCsU(void \*buf, size\_t len)** Entrega como máximo *len* bytes recibidos por la conexión. En caso de no contar con datos devolverá 0 byte. Los datos se devuelven a través del parámetro *buf*. El procedimiento devuelve cantidad de bytes devueltos en *buf*, y en caso de error devuelve -1. En caso de no haberse establecido conexión devuelve -1.
- **int cerrarTCsU()** Cierra la conexión en forma ordenada y destruye las estructuras de datos, y los procedimientos necesarios para el control del Transporte Confiable sobre UDP. En caso de error devuelve -1.

El intercambio de información entre los dos extremos se debe realizar a través de TPDU's (*Transport Protocol Data Unit*).

Las TPDU's contienen un cabezal presentado en la figura 2 y un campo de datos con largo entre 0 y 1400 bytes.

La semántica de los campos que integran el cabezal se indican a continuación:

- *tipo\_TPDU*, este campo contiene dos valores posibles 0x00 que indica que el TPDU transferido es de control, por lo tanto no se contabilizará ni actualizarán los números de secuencia, y 0x01 que indica que el TPDU es para la transmisión de datos.
- *Flags\_TPDU* con el mismo criterio utilizado por TCP y con banderas de SYN, ACK, RST y FIN.
- *TPDU\_nroSecuencia* contiene el número de secuencia de la TPDU. Se debe numerar por TPDU y no por byte, y en caso de pérdida deberá repetirse la TPDU entera.
- *TPDU\_nroACK* contiene el número de secuencia esperado por el extremo origen del TPDU.
- *Ventana*, campo utilizado por control por *TCsUv2*, indica cantidad de TDPU's que puede aceptar.
- *Checksum*, campo de chequeo de la integridad de toda la TPDU.

	<b>Bits 0-7</b>	<b>Bits 8-15</b>
<b>0</b>	<b>tipo_TPDU</b>	<b>Flags_TPDU</b>
<b>16</b>	<b>TPDU_nroSecuencia</b>	<b>TPDU_nroACK</b>
<b>32</b>	<b>Ventana</b>	<b>Checksum</b>

Fig. 2 Cabezal TPDU

Se debe implementar un mecanismo de *byte stuffing*, que permita la detección del comienzo y fin de las TPDU's, y que la TPDU transmita cualquier byte en su campo de datos (transparencia).

El mecanismo de *byte stuffing* coloca previo al comienzo de la TPDU la secuencia especial de bytes DLE\_BSTUFF (*data link escape*), STX\_BSTUFF (*start of text*) y finaliza la TPDU con la secuencia especial DLE\_BSTUFF, ETX\_BSTUFF (*end of text*).

Cuando aparece en el contenido de la TPDU el byte DLE\_BSTUFF, se procede agregando otro byte DLE\_BSTUFF. El receptor procesa la secuencia de byte, buscando el comienzo y fin de la TPDU por las secuencias especiales agregadas y eliminando los bytes DLE\_BSTUFF agregados para lograr la transparencia en el envío.

### Se pide

Implementar una API en C/C++, que contenga las funciones definidas anteriormente e implemente parte del *Transporte Confiable sobre UDP (TCsUv2)* correspondiente a los requerimientos 1 a 3. Dicha API se se linkeditará con programas que utilicen las mencionadas funciones. Deben resolverse los siguientes elementos del protocolo:

- Implementar mecanismo de detección de TPDU's transmitidas por UDP.
- Establecimiento de conexión del *TCsUv2*, generando las estructuras sobre UDP requeridas para la conexión real.
- Desconexión del *TCsUv2*.

### Entregable

Se deberá entregar:

1.Documentación. La documentación debe contener:

- a)Descripción exhaustiva en idioma Español del protocolo.
- b)Maquina de estados propuesta para la resolución de *byte stuffing*.
- c)Propuesta de implementación del mecanismo de transporte confiable.
- d)Pseudocódigo detallado y comentado del módulo *TCsUv2* entregado.

2.Implementación:

- a)Código fuente.
- b)*Makefile* para la compilación del mismo.

### Insumos

Se entregará *tcsu.h* conteniendo el cabezal de las funciones a implementar, constantes pre-establecidas, y formato del cabezal de las TPDU.

Se entregarán programas que utilizaran la API especificada para transmitir información utilizando la implementación de *TCsUv2* realizada durante este obligatorio.

### 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/>).