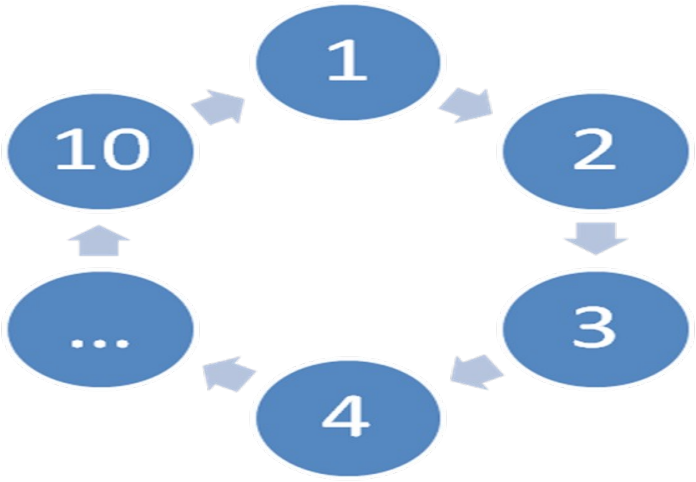
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN (UPCT)
DEPARTAMENTO DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

LABORATORIO DE ARQUITECTURA DE REDES DE COMUNICACIONES
(Ingeniero Técnico de Telecomunicación, Esp. Telemática)

Examen Final: Convocatoria de septiembre. Fecha: 16 de septiembre de 2010.

NOTA: El enunciado del examen NO hay que entregarlo. Los problemas se entregarán en hojas separadas. Responder el test en la hoja de respuestas del test. Escribir el nombre y DNI en todas las hojas que se entreguen. Numerar todas las páginas que se entreguen.

PROBLEMA 1. – En el contexto del laboratorio asuma que los equipos se han configurado con la siguiente topología en anillo:



TOPOLOGIA DE LA RED

Es decir, que el equipo N sólo puede transmitir paquetes al equipo (N+1), si N<10, o desde el equipo 10 al 1.

En esta configuración se considera el desarrollo de un protocolo de envío/recepción de mensajes que se construirá sobre libpaquete. Es decir, el protocolo a construir hará uso de las funciones enviarpkt(), recibirpkt(), localhost() y éstas exhibirán el mismo comportamiento "no fiable" (duplicados, pérdidas, retardos) que en el desarrollo de las prácticas de la asignatura.

El protocolo a desarrollar debe permitir:

- El envío de mensajes entre dos pares cualesquiera de la red (origen, destino), teniendo en cuenta las restricciones topológicas. Es decir, si por ejemplo 1 desea enviar un mensaje a 3, este debe dar un salto a través de 2.
- Entrega confiable de mensajes. Es decir, los paquetes deberán ser asentidos mediante un mensaje de ACK. En el ejemplo anterior, el ACK deberá viajar de 3 a 1 a través de los nodos 4, 5, 6, 7, 8, 9 y 10.

No es necesario solucionar los problemas de duplicados.

El sistema funcionará siguiendo un esquema síncrono. Cada 30 segundos (tiempo de *timeslot*) se enviará el contenido de una variable *buffer* como mensaje *i*-ésimo. El nodo destino estará indicado en una variable llamada *destino*. Para implementar la entrega confiable de mensajes, si un equipo no recibe confirmación de este mensaje *i*-ésimo, retransmitirá el mensaje en el siguiente *timeslot,* así hasta recibir el ACK correspondiente. En ese momento, llamará a la función *actualizarbuffer()* **(debe asumir que esta función ya existe)** que actualizará *buffer* con el siguiente mensaje y la variable *destino*, y en el siguiente *timeslot* enviará el mensaje (*i+1*)-ésimo.

Asimismo, durante un *timeslot* el sistema debe atender los mensajes entrantes:

- Si son para el propio nodo debe entregarlos llamando a la función *entregarbuffer*(char *buffer) **(debe asumir que esta función ya existe)**
- Si son para otro nodo debe enviarlo al siguiente nodo del anillo
- Si es un ACK para el propio nodo debe procesarlo
- Si es un ACK para otro nodo debe enviarlo al siguiente nodo del anillo

Se le solicita que:

a) Indique cuántos y qué formato tendrán los mensajes intercambiados en el sistema.
b) Codifique mediante pseudo-código el esquema propuesto. Nótese que todo el proceso puede implementarse mediante un único proceso secuencial.
c) Codifique en C el pseudo-código anterior.