Protocolos de Comunicación
y Sistemas Embebidos

Tarea 3: "Tabla de ruta en red anillo"

Objetivos

Crear e implementar un algoritmo capaz de crear tablas de ruta en una red anillo con puertos bidireccionales basado en la tarea anterior.

Grupos de Trabajo

Se debe trabajar en grupos de 2 personas donde ambos deberán trabajar en misma medida.

Programa

Se pide desarrollar **un único programa** llamado "nodo", el cual debe ejecutarse en el entorno de una Raspberry PI. El programa debe ser capaz de enviar y recibir mensajes dentro de una red, cada ejecución del programa corresponderá a un nodo. El mensaje por enviar debe ser encapsulado con los protocolos SLIP e IPv4 Modificado (mostrado en tabla 1).

- Al momento de ejecutar el programa en la consola, se debe especificar en la misma línea de comandos la IP del nodo, y sus 2 puertos de comunicación bidireccionales (usar nomenclatura *../tmp/p2*) *Ejemplo: ./nodo 192.168.0.1 tmp/p1 tmp/p10*. Para facilitar la ejecución incluir estos comandos en el archivo makefile.

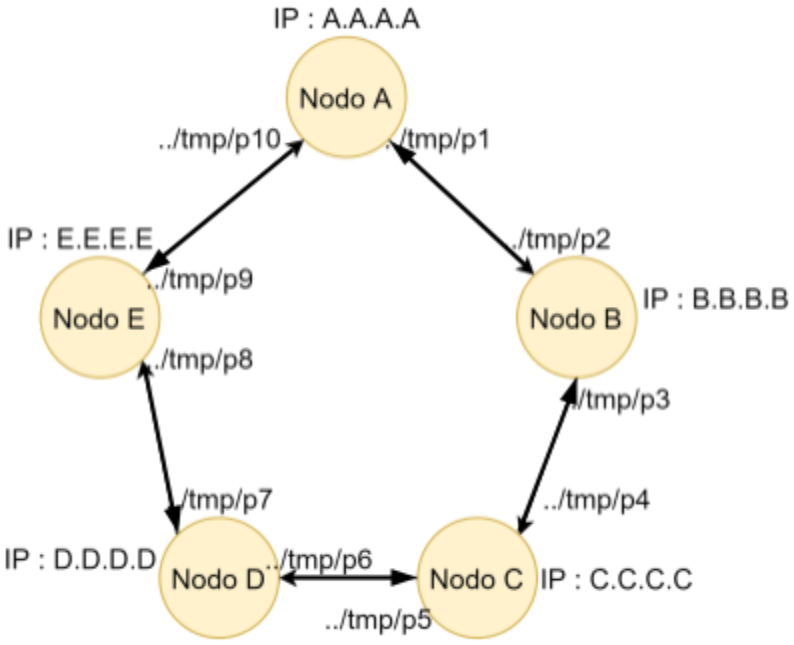

Figura 1: Red anillo a formar.

- Cada nodo utilizará 2 puertos, puerto 1 y puerto 2 el cual puede ser definido internamente dependiendo de los parámetros de entrada al ejecutar el programa. Con el fin de generar las siguientes interconexiones virtuales mostradas en la figura 1 en donde se observa la IP (opcional a cada grupo, pero debe indicar la utilizada en el informe) de cada nodo y sus respectivos puertos de comunicación.
- Todos los nodos tienen que poder mandar 2 tipos de mensajes:
    - Broadcast: Este tipo de mensaje tiene una dirección de destino con todos sus bits en alto, sin incluir el identificador de red, aunque para este caso se utilizará la dirección de destino F.F.F.F. Al recibir un mensaje con esta dirección, el programa debe reconocer que recibió un broadcast, reducir en 1 su TTL y reenviar el mensaje por el otro puerto. Al momento de recibir el mensaje el nodo debe evaluar la distancia en saltos que existe entre el emisor del mensaje y sí mismo, considerando el puerto donde se recibió el mensaje. Esta información debe quedar registrada en una tabla de ruta que se debe mostrar cada vez que reciba un broadcast. Si el nodo recibe en otro momento un mensaje de un nodo ya registrado con una distancia menor de saltos en otro puerto, debe actualizar su tabla de ruta, indicando con un mensaje que la tabla se actualizó. La estructura de una tabla de ruta simplificada se muestra en la tabla 1. Si un nodo recibe un mensaje tipo broadcast con su dirección, el mensaje debe ser descartado.

Dirección IP	Puerto	TTL
B.B.B.B	1	1
D.D.D.D	2	2

Tabla 1: Ejemplo de tabla de ruta.

    - Unicast: Este tipo de mensajes tiene una dirección de destino de un nodo en particular. Estos mensajes deben enviarse sólo si el nodo destino está registrado en la tabla de ruta del nodo emisor, de lo contrario indicar por consola que el usuario destino no es accesible. Al enviar este mensaje los nodos que no tengan la IP de destino deben simplemente reducir en 1 el TTL del paquete y enviarlo por su puerto emisor. Al momento del nodo destino recibir este mensaje debe anunciar por pantalla que recibió un mensaje tipo unicast y mostrar en la consola el contenido del mensaje.
- Todos los mensajes enviados deben ser encapsulados en el protocolo IPv4 modificado (Capa 3) y luego empaquetados en SLIP (Capa 2) para su posterior envío.
- Todos los menús por consola deben ser auto explicativos.
- Agregar en el menú la opción de mostrar la tabla de ruta.

Información Adicional:

Protocolos de Comunicación y Sistemas Embebidos

En la tabla 1 se muestra el Protocolo IPv4 modificado en donde se observan diferentes campos, con sus respectivos pesos en bits, para este caso el Flag de fragmento, el offset de Fragmento se consideran para uso en la sección de puntos extras, en donde el primer campo menciona indica con un 0 que el paquete no es fragmentado, con un 1 que es un fragmento y con un 2 que es último fragmento (el campo "Identificación" es el mismo para los fragmentos del dato fragmentado), los demás campos deben ser utilizados para la tarea de la siguiente manera, la longitud de datos permite una capacidad de 2^{16} - 1 bytes originalmente pero para este caso solo se limita a 1500 Bytes, el TTL será el campo en donde se guarde el Tiempo de vida del paquete, la identificación es única por dato y es ingresada como un contador por el emisor (para paquetes fragmentados este identificador es el mismo), la Suma de verificación es solo de la cabecera, la IP de origen y destino indican el direccionamiento de los nodos y los datos son el mensaje a enviar.

Tabla 1: Protocolo IPv4 Modificado.

<table>
<tr><td>0</td><td>7</td><td>8</td><td>15</td><td>16</td><td>31</td></tr>
<tr><td>Flag de fragmento</td><td colspan="3">Offset de Fragmento</td><td colspan="2">Longitud de los datos</td></tr>
<tr><td colspan="2">TTL</td><td colspan="2">Identificación</td><td colspan="2">Suma de verificación</td></tr>
<tr><td colspan="6">IP origen</td></tr>
<tr><td colspan="6">IP destino</td></tr>
<tr><td colspan="6">Datos</td></tr>
<tr><td colspan="6">...</td></tr>
</table>

Para la fácil comprensión de su código es necesario comentar por secciones, describiendo qué es lo que se está llevando a cabo. En caso de usar funciones, describir qué hace, sus salidas y entradas.
La tarea debe incluir un archivo Makefile para compilar ambos programas. En caso de que los programas no compilen, la tarea no será revisada.

Informe

Para este trabajo el informe debe ser solo un reporte del programa y un manual de usuario.

Entrega

Debe ser subida a Campus Virtual la fecha indicada con el texto. Recuerde que en el archivo comprimido deben estar todos los archivos necesarios para el correcto funcionamiento de la tarea, este debe tener los apellidos de los integrantes en formato "Apellido1_Apellido2.zip". Además, se debe incluir una carpeta "doc" con el informe solicitado en latex y pdf.

Fechas

Entrega: 3 de Julio de 2024.

Protocolos de Comunicación y Sistemas Embebidos

Observación

Se descontará **1 punto** por día de atraso. **La copia será sancionada con un 1.0.** Recuerde documentar adecuadamente el código fuente, de lo contrario tendrá descuentos. **Se descontará puntaje si la tarea no cumple con los formatos solicitados.**

Referencias útiles

https://www.geeksforgeeks.org/command-line-arguments-in-c-cpp/