**MONOHILO & MULTIHILO**

**Freddy L. Abad Leon**

**4.2. Preguntas UDP**

1. **¿Qué sucede si reservamos un array de menor tamaño (p.e. 5 bytes) que el tamaño del mensaje que estamos recibiendo?**

Al momento de reservar un array de menor tamaño que el mensaje que recibimos, el mensaje llegaría incompleto, ya que al no tener la memoria suficiente para guardar la información receptada tendería a perderse, es decir el array se trunca.

1. ¿**Qué sucede si queremos lanzar dos servidores UDP simultáneamente en el mismo número de puerto? ¿Por qué?**

Revisando la información de los puertos UDP y la arquitectura de los paquetes enviados, se puede notar que sin tener una conexión establecida, los paquetes se pueden enviar, porque tienen en cabecera la siguiente información: Puerto de origen, puerto de destino, suma de verificacion, longitud del mensaje y los datos.

Tomando en cuenta esta estrcutura de paquetes, si se conectan dos servidores UDP al mismo puerto, el primer servidor reserva el puerto y no se puede utilizar por otro servidor.

1. **¿Qué sucede si queremos lanzar varios clientes para que se comuniquen con el mismo servidor? ¿Existe algún tipo de bloqueo en los clientes que impida la comunicación?**

Si se envían varios clientes a conecarse con el mismo servidor no habría problema con la conexión, ya que los paqeutes se enviaría debido a que no es necesario una conexión preestablecida para el intercambio de datos, pero esta practica estaría a decisión del programador, ya que al ser UDP un protocolo orientado a la no conexión no es fiable en el envio de datos secuencialmente con confirmaciones ACK o NACK (caso del TCP/IP donde se crea un hilo para cada conexión y no hay restrccion).

1. **¿Qué sucede si no está vacío el buffer de recepción al llegar al método necesario para recibir un DatagramPacket?**

El mensaje es puesto en una cola por el *socket* que está asociado al puerto de destino. El mensaje puede ser recogido de la cola por una excepción o llamadas futuras de recepcion sobre ese *socket*. Los mensajes son descartados en el destino si ningún proceso tiene asociado un *socket* al puerto de destino.

5.2. Preguntas TCP

1. En la clase ServidorTCP, ¿cuál es la diferencia entre invocar el método start()e invocar al método run()de la clase ThreadServidor?

El método run() contiene el código a ser ejecutado “asincrónicamente” en otro thread, mientras que el método start() es el que crea al Thread y en algún punto hace que ese Thread ejecute lo que esta en run(). Este método devuelve el control inmediatamente. Pero si mezclamos todo y ejecutamos directamente el run(), el código se ejecutara en el Thread actual.

El método start() devuelve el control inmediatamente… mientras tanto, el nuevo Thread inicia su recorrido por el método run(). ¿Hasta cuando? Hasta que termina ese método, cuando sale, termina el Thread.

1. ¿En dónde se cierra el nuevo socket creado en el ServidorTCP cuando se acepta una nueva conexión? ¿Por qué?