

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Prof. Moreno Cervantes Axel Ernesto

Tarea envío de datagramas

I N T E G R A N T E S

Salinas Monroy America Joana

Sanchez Barragan Rodrigo

Introducción

Definición

Un datagrama es un paquete de datos que constituye el mínimo bloque de información en una red de conmutación por datagramas. Este tipo de red es uno de los dos protocolos de comunicación por conmutación de paquetes utilizados para encaminar unidades de información entre nodos de una red. A diferencia de la conmutación por circuito virtual, que está orientada a conexión, la conmutación por datagramas no establece conexiones previas.

Estructura del Modelo

Los datagramas se componen de dos partes:

1. Cabecera: Contiene información de control.
2. Datos: Los propios datos que se desean transmitir.

Cada paquete se trata de forma independiente gracias a que puede contener en la cabecera la dirección de origen y destinatario.

Mediante un encaminador (también conocido como enrutador o router), la red puede encaminar cada fragmento hacia el receptor o Equipo Terminal de Datos (ETD) por rutas diferentes.

Funcionamiento:

- En la técnica de datagramas, cada paquete se trata de manera independiente.
- La dirección de origen y destinatario en la cabecera permite el enrutamiento hacia el receptor.
- La flexibilidad de los datagramas permite:
 - a. Control del tráfico para aprovechar la capacidad de canal de cada tramo de red.
 - b. Adaptarse ante congestiones y caídas de nodos intermedios.
 - c. Abaratar costes al ajustar el ancho de banda y número de líneas precisados.
- Sin embargo, también impide garantizar:
 - a. Velocidad constante del flujo de datos.
 - b. Recepción de paquetes en el orden original.
 - c. Que todos los paquetes lleguen a su destino.

Uso en Internet:

Internet utiliza protocolos tanto orientados a conexión como no orientados a conexión.

- TCP (Transmission Control Protocol) garantiza que todos los paquetes lleguen correctamente y en orden.
- UDP (User Datagram Protocol) no garantiza la entrega de los datagramas.
- Los datagramas IP son las unidades principales de información en Internet.

En resumen, los datagramas son fundamentales para la comunicación en redes de conmutación de paquetes como Internet, ofreciendo flexibilidad y adaptabilidad, aunque a expensas de algunas limitaciones en la garantía de entrega y orden de los datos.

Desarrollo

Ciente

Esta clase se encarga de enviar un datagrama que contiene un objeto serializado a través de una red. Crea un objeto Persona y se encapsula junto con los otros datos en un objeto Objeto, después se serializa mediante un arreglo de bytes, una vez hecho esto se configuró el DatagramPacket a la dirección IP y el puerto especificado del servidor, por último una vez enviado el datagrama el socket se cierra.

Funcionalidad

Creación de un objeto Persona y un objeto Objeto que lo contiene:

- `Persona persona = new Persona("Rodrigo", "Sanchez", 22);`
- `Objeto mensaje = new Objeto(1000, 2000, persona);`

Inicialización y configuración del mensaje:

- `mensaje.inicializarIpOrigen(dir, dir);`
- `mensaje.mensaje = "Hola soy un mensaje enviado desde el socket";`

Serialización del objeto:

- Uso de `ByteArrayOutputStream` y `ObjectOutputStream` para convertir el objeto en una secuencia de bytes.

Envío del datagrama:

- `DatagramSocket cl = new DatagramSocket();`
- `cl.setBroadcast(true);`
- Creación y envío del `DatagramPacket` con el objeto serializado.

Clase Objeto

Esta clase representa el mensaje que se envía a través de la red y es serializable.

Métodos

- `inicializarIpOrigen(String ipOrigen, String ipDestino)`: Inicializa las IPs de origen y destino y graba la fecha de envío.
- Métodos getter y setter para los atributos.
- `int suma()`: Retorna la suma de a y b.
- `void grabarFecha()`: Graba la fecha y hora actuales.

Clase Persona

Esta clase representa a una persona y es parte del objeto Objeto.

Método:

- `toString()`: Retorna una representación en cadena de la persona.

Servidor

Esta clase se encarga de recibir el datagrama que contiene el objeto serializado, deserializarlo y mostrar la información recibida. Primero se coloca el puerto y se espera la recepción de datagramas de manera indefinida, después se deserializa para obtención del objeto Objeto original, se extrae y se muestra la información del objeto Objeto.

Recepción de datagramas en un bucle infinito:

- Creación de un `DatagramPacket` para recibir los datos.
- Recepción del paquete con `s.receive(p)`;
- Deserialización del objeto recibido usando `ByteArrayInputStream` y `ObjectInputStream`.

Mostrar información del objeto recibido:

- Imprimir el contenido del mensaje, IPs de origen y destino, números, fecha de envío y datos de la persona.