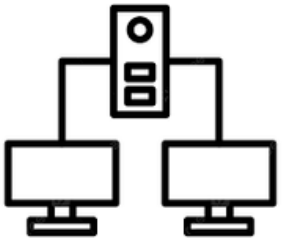
APLICACIONES PARA COMUNICACIONES EN RED

METAINFORMACIÓN

PARA SOCKETS DE DATAGRAMA



INTEGRANTES:

Jiménez Xala Arath

Torres Larios Andres Emiliano

Metainformación para sockets de datagrama

Para cuando enviamos paquetes a través de sockets de datagrama es necesario separarlo en varios paquetes de acuerdo con el tamaño del paquete, es por lo que es necesario concatenarle metainformación el cual contenga, tamaño total del paquete, total de paquetes segmentados y el número de paquete que se está entregando.

Al concatenar esta metainformación podemos asegurarnos que el paquete llegue completo, haciendo verificaciones con el número de paquetes y el tamaño total del paquete, en caso de que el tamaño sea menor o mayor y/o el número de paquetes sea menor o mayor podemos saber que el paquete se ha recibido de una forma incorrecta.

La forma que el equipo piensa para concatenarle esta metainformación al paquete usando Java es:

Emisor:

```
public class Emisor {
    public static void main(String[] args){
        try {
            String mensaje = "msg";
            byte[] tmp = mensaje.getBytes();

            // Definimos el total de segmentos. En este ejemplo se segmenta en 2
partes.
            int totalSegmentos = 2;
            int tamSegmento = tmp.length / totalSegmentos;

            DatagramSocket cl = new DatagramSocket();
            for (int ii = 0; ii < totalSegmentos; ii++) {
                // Usamos ByteArrayOutputStream para construir el paquete.
                ByteArrayOutputStream baos = new ByteArrayOutputStream();
                DataOutputStream dos = new DataOutputStream(baos);

                // Se escriben los metadatos:
                dos.writeInt(tmp.length);       // Tamaño total del mensaje
                dos.writeInt(totalSegmentos);   // Total de segmentos que se
enviarán
                dos.writeInt(ii);               // Número de paquete actual

                // Calculamos el segmento a enviar.
                int inicio = ii * tamSegmento;
                int fin = (ii == totalSegmentos - 1) ? tmp.length : inicio +
tamSegmento;
                byte[] btmp = Arrays.copyOfRange(tmp, inicio, fin);

                // Se escribe la longitud del segmento y luego los datos.
```

```
                dos.writeInt(btmp.length);      // Tamaño del segmento
                dos.write(btmp);                // Datos del segmento

                dos.flush();
                byte[] b = baos.toByteArray();

                // Se configura el datagrama y se envía.
                InetAddress dir = InetAddress.getByName("127.0.0.1");
                DatagramPacket p = new DatagramPacket(b, b.length, dir, 5555);
                cl.send(p);

                System.out.println("Enviando el paquete " + ii
                        + " de " + totalSegmentos
                        + " con el mensaje: " + new String(btmp));
                System.out.println("Mensaje enviado con un paquete de " +
b.length + " bytes.");

                dos.close();
                baos.close();
            }
            cl.close();
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

Receptor:

```
public class Receptor {
    public static void main(String[] args){
        try {
            DatagramSocket s = new DatagramSocket(5555);
            System.out.println("Servidor esperando datagrama...");
            for(;;) {
                DatagramPacket p = new DatagramPacket(new byte[65535], 65535);
                s.receive(p);
                DataInputStream dis = new DataInputStream(new
ByteArrayInputStream(p.getData()));

                // Se leen los metadatos enviados:
                int totalMensaje = dis.readInt();    // Tamaño total del mensaje
original
```

```
                int totalSegmentos = dis.readInt();    // Número total de
segmentos
                int numPaquete = dis.readInt();        // Número de este paquete
(segmento)

                // Se lee la longitud del segmento actual y el contenido:
                int tamSegmento = dis.readInt();       // Tamaño del segmento
                byte[] b = new byte[tamSegmento];
                dis.readFully(b);
                String mensajeSegmento = new String(b);

                System.out.println("Paquete recibido: número " + numPaquete
                        + " de " + totalSegmentos
                        + ", tamaño total del mensaje: " + totalMensaje
                        + ", segmento de " + tamSegmento + " bytes. Mensaje: " +
mensajeSegmento);

                dis.close();
            }
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```