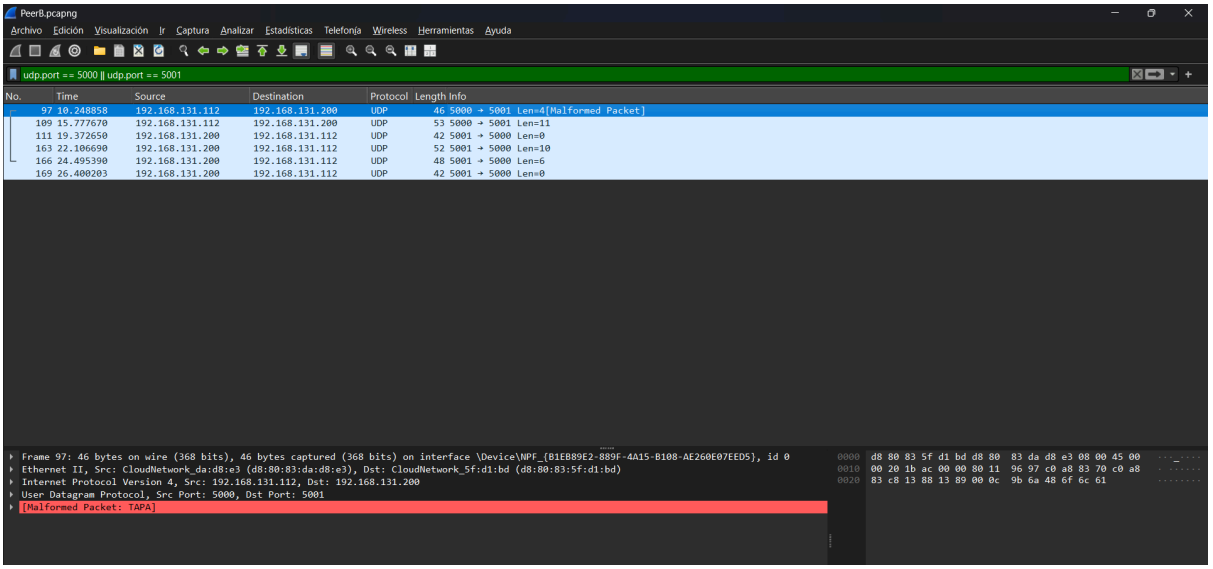
Melissa Dayana Mora Ballesteros
Repositorio: https://github.com/Meli-Ballesteros/Compunet_peer

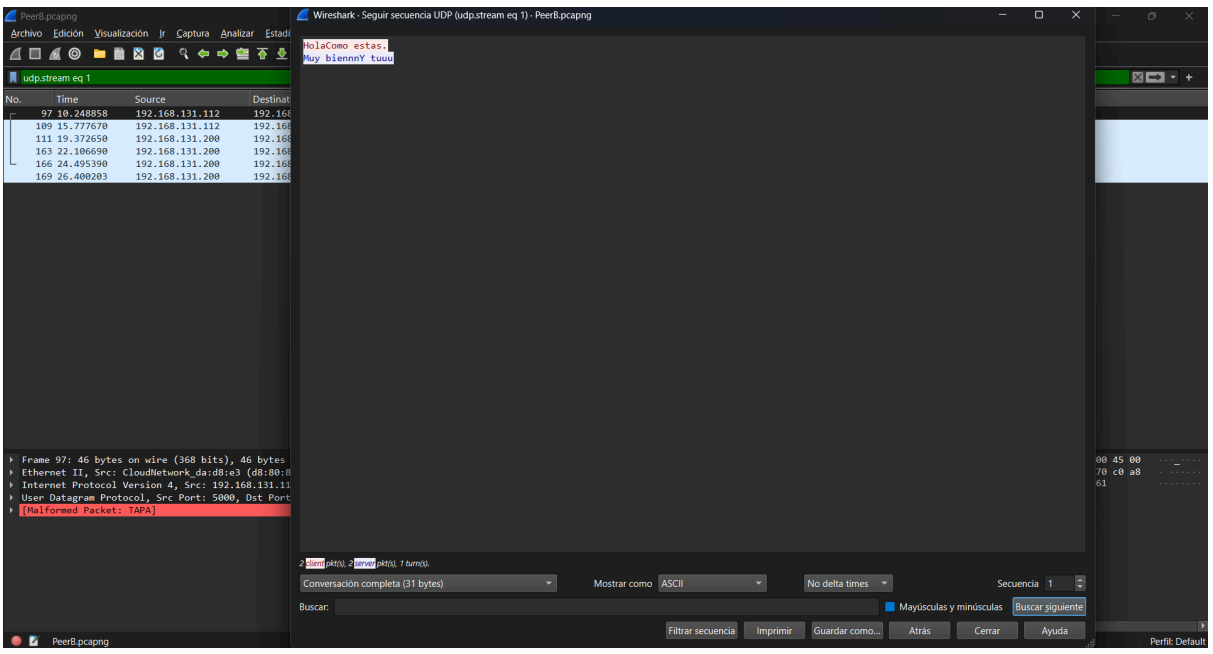
¿Es posible ver en la captura de Wireshark el contenido del mensaje enviado?

R//:

Sí, es completamente posible ver el contenido de los mensajes UDP en Wireshark. Debido a que UDP no cifra ni protege los datos de aplicación. Esto significa que cualquier mensaje que se envíe queda visible en el paquete UDP.



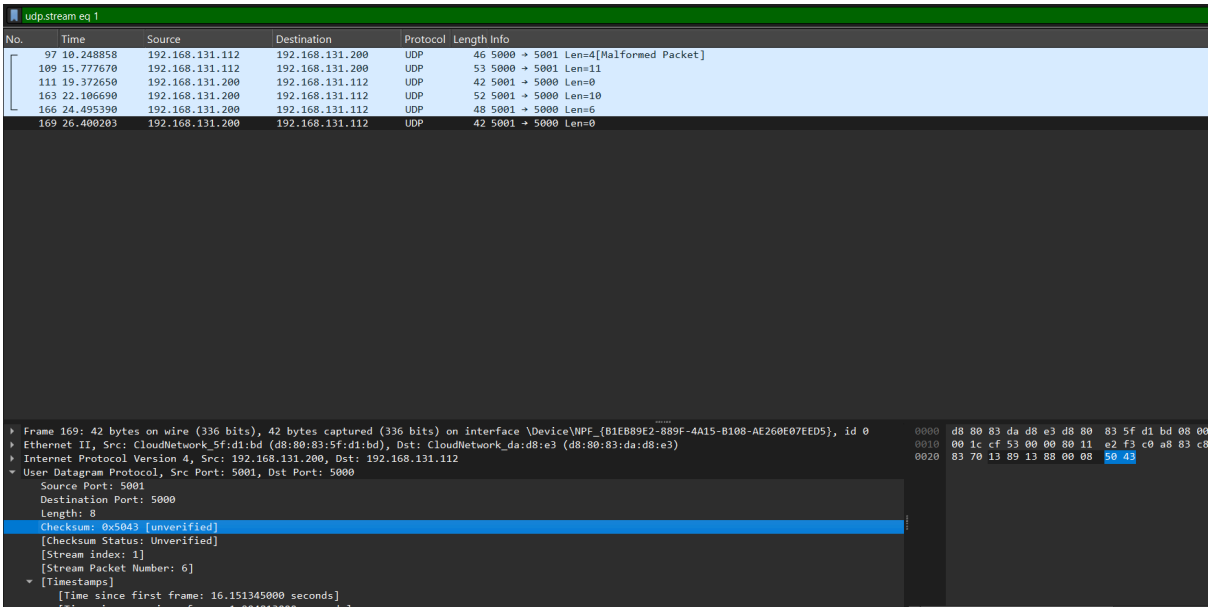
Dandole en click derecho y Seguir se puede ver el contenido.



Melissa Dayana Mora Ballesteros
Repositorio: https://github.com/Meli-Ballesteros/Compunet_peer

¿Cuál es el checksum de la captura? ¿Explique/investiguen porque este checksum?

R//:



El checksum 0x5043 en hexadecimal es un valor de 16 bits que verifica la integridad de los datos enviados en un datagrama UDP. Este checksum permite que el receptor valide si el contenido fue alterado durante la transmisión.

Wireshark calcula este valor a partir de:

- IP de origen: 192.168.131.200
- IP de destino: 192.168.131.112
- Protocolo: UDP
- Longitud total del mensaje

¿Qué patrones de diseño/arquitectura aplicará al desarrollo de un programa basado en red como este?

- Singleton
- Thread Multihilo
- Single Responsibility
- Mejoraría aplicar Observer Pattern para notificar eventos de llegada de mensaje a una GUI, dividiendo en capas MVC si se implementa JavaFX. Con su UI, eventos y conexión UDP.

Melissa Dayana Mora Ballesteros
Repositorio: https://github.com/Meli-Ballesteros/Compunet_peer

Investiguen que modificaciones son necesarias para implementar este mismo sistema pero para la comunicación TCP en java

R//: Se necesita una gestión explícita de conexiones, como usar Sockets, establecer conexiones reales si no, no funciona. Algo así:

“Servidor”
```
ServerSocket serverSocket = new ServerSocket(5000);
Socket clientSocket = serverSocket.accept();
BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
System.out.println("Mensaje recibido: " + in.readLine());
```

“Cliente”
```
Socket socket = new Socket("192.168.131.112", 5000);
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
out.println("Hola desde TCP");
```

¿Qué utilidades de codificación o seguridad agregarías al código?

R//: Aplicaría Advanced Encryption Standard en modo simétrico. Al igual. siento que compartiría una clave secreta entre peers. También, permitiría solo recibir paquetes de IPs autorizadas, rechazando inmediatamente IPs ajenas a mi conocimiento. Igualmente, limitaría el tamaño de los mensajes que se envían, teniendo asimismo un filtro para ver los mensajes que se envían clasificándolos entre buenos y malos según mi criterio. Por último, agregaría una autenticación ya sea con password o token.