

Sistema de Descubrimiento de Servicios en Red con UDP

BONO

Mateo López - 202220119

Angélica Ortiz Ballesteros – 202222480

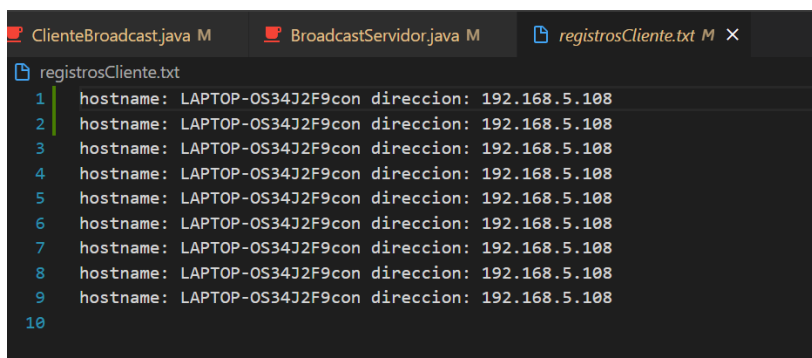
Explicación del diseño e implementación de cliente y servidor:

Cliente:

Para la implementación del cliente se utilizaron la librería DatagramSocket para poder enviar datagramas a través de un socket específico de la máquina que ejecuta el programa. Para poder hacer que sea un broadcast, no se le asignó una dirección ni puerto específico al DatagramSocket, y se configuró en modalidad de broadcast con el comando `socket.setBroadcast(true)`. Posterior a eso, se creó un objeto DatagramPacket con dirección IP de destino siendo 255.255.255.255 (broadcast) al puerto UDP 4445, y se envió este paquete. Esto se envía cada 5 segundos para facilitar la captura de paquetes en Wireshark.

Servidor:

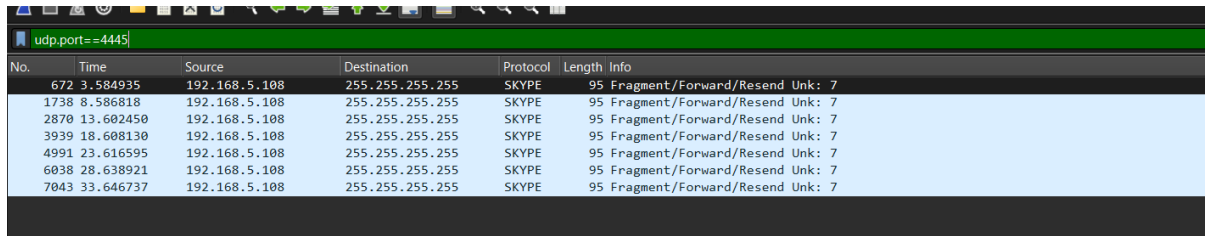
Para la implementación del servidor, se creó un DatagramSocket en el puerto 4445 (ya que el cliente va a enviar a ese puerto en todos los dispositivos conectados a la red) con broadcast encendido. Cuando el cliente envía el paquete, el servidor va a escuchar y recibirlo, tomando la información guardada dentro, la cual es el hostname y dirección IP del cliente, y la va a almacenar en el archivo registrosCliente.txt, el cual se ve así después de ejecutar el cliente:



```
registrosCliente.txt
1 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
2 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
3 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
4 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
5 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
6 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
7 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
8 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
9 hostname: LAPTOP-OS34J2F9con direccion: 192.168.5.108
10
```

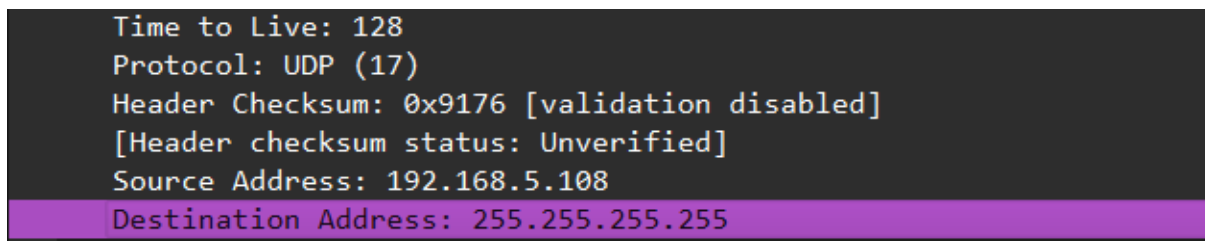
Archivos capturados en Wireshark:

Al filtrar únicamente los archivos escuchados en el puerto UDP 4445 desde un tercer computador, se pudo ver que los paquetes también fueron recibidos:

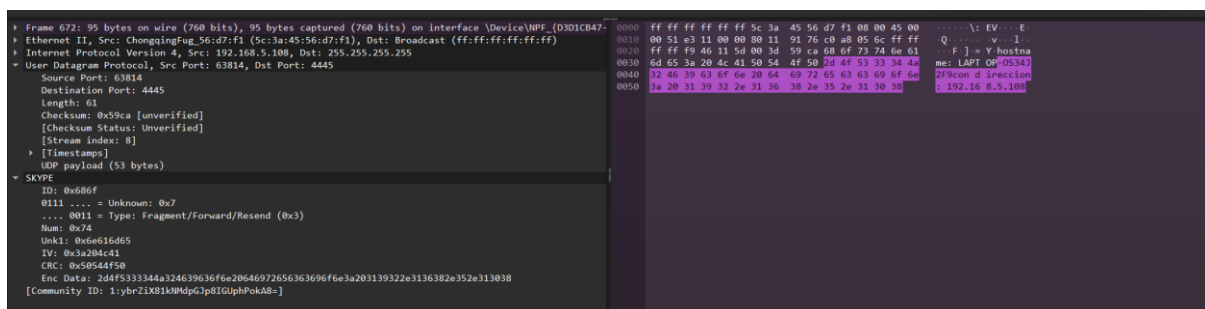


No.	Time	Source	Destination	Protocol	Length	Info
672	3.584935	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
1738	8.586818	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
2870	13.602450	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
3939	18.608130	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
4991	23.616595	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
6038	28.638921	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7
7043	33.646737	192.168.5.108	255.255.255.255	SKYPE	95	Fragment/Forward/Resend Unk: 7

Aunque el protocolo mostrado dice Skype, según un foro de Wireshark esto es porque "The current heuristic for Skype is 'a UDP packet with 3 or more bytes, and with the lower 4 bits of the 3rd byte being one of 0x2, 0x3, 0x5, 0x7, 0xd, or 0xf, is assumed to be a Skype packet'. That's a rather weak heuristic; perhaps it could be strengthened - the code has the comment 'FIXME: Extend this by minimum or exact length per message type', por lo que igual se sabe que es un paquete enviado a través del protocolo de transporte UDP.



De la capa de red se puede evidenciar que se está enviando como broadcast a la dirección 255.255.255.255, y que el protocolo de transporte usado es UDP.



En la trama del mensaje se puede ver en texto plano la información de hostname y dirección del cliente siendo parte del paquete entregado.

Foro de wireshark:

<https://ask.wireshark.org/question/9109/why-is-wireshark-reporting-skype-traffic-on-a-network-with-no-skype-traffic/>