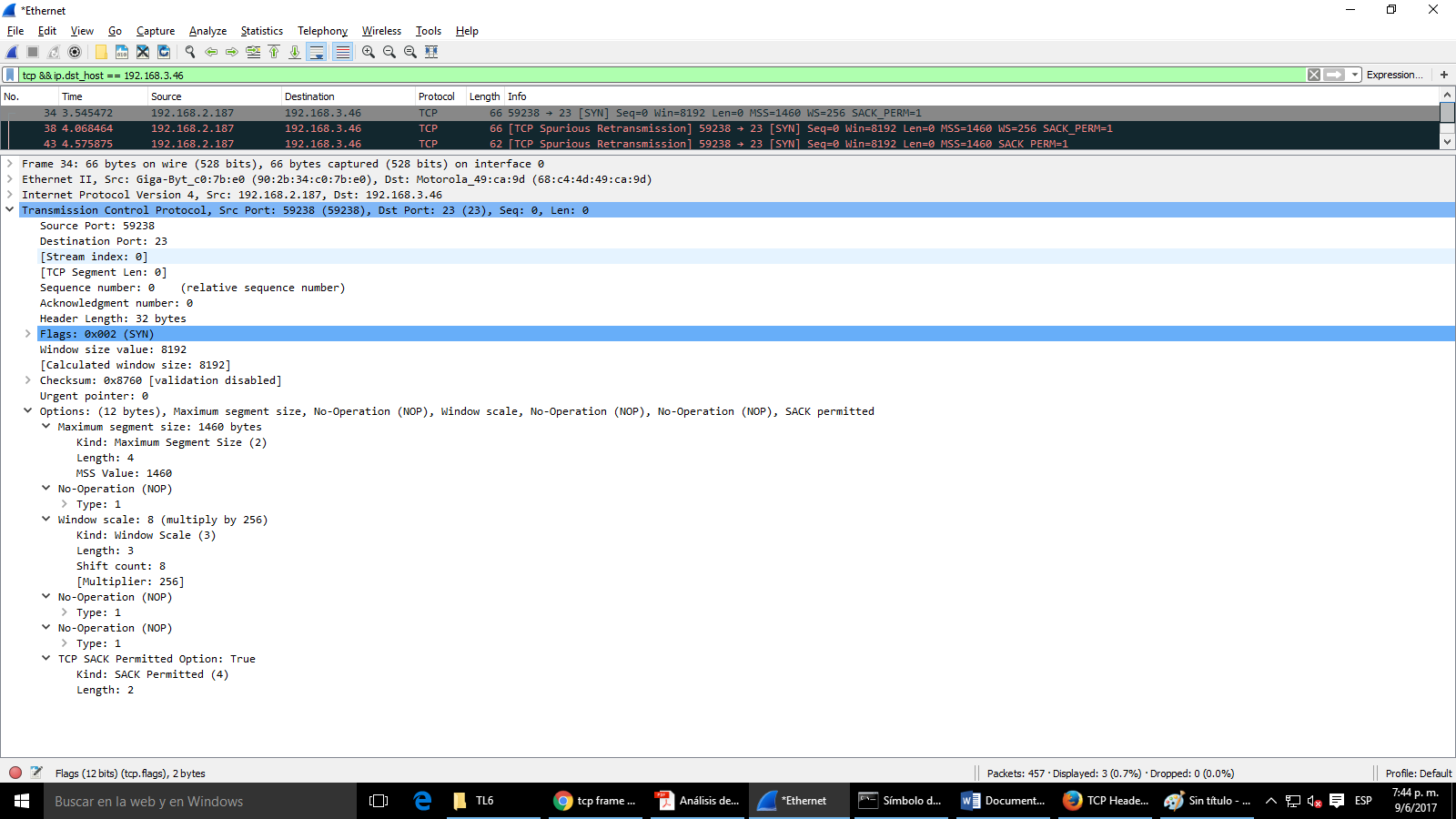
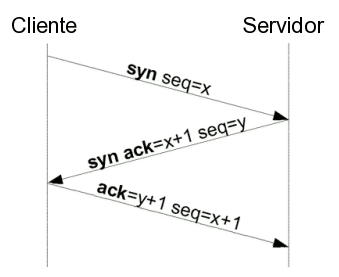
1. **a. Análisis del tráfico que produce un protocolo orientado a la conexión.**
2. Usamos telnet que usa tcp.



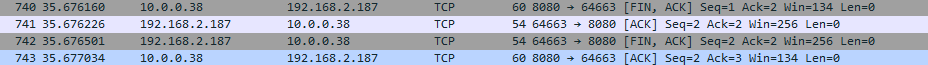
1. Puede ser empleado el protocolo.
2. Ya esta activado el sniffer.
3. telnet 192.168.3.46 (direccion destino)
4. Establecimiento de conexión tcp, utiliza 3 segmentos:



Gráfico:



Para liberación utilizamos 4 segmentos, FIN ACK y ACK en cada extremo:



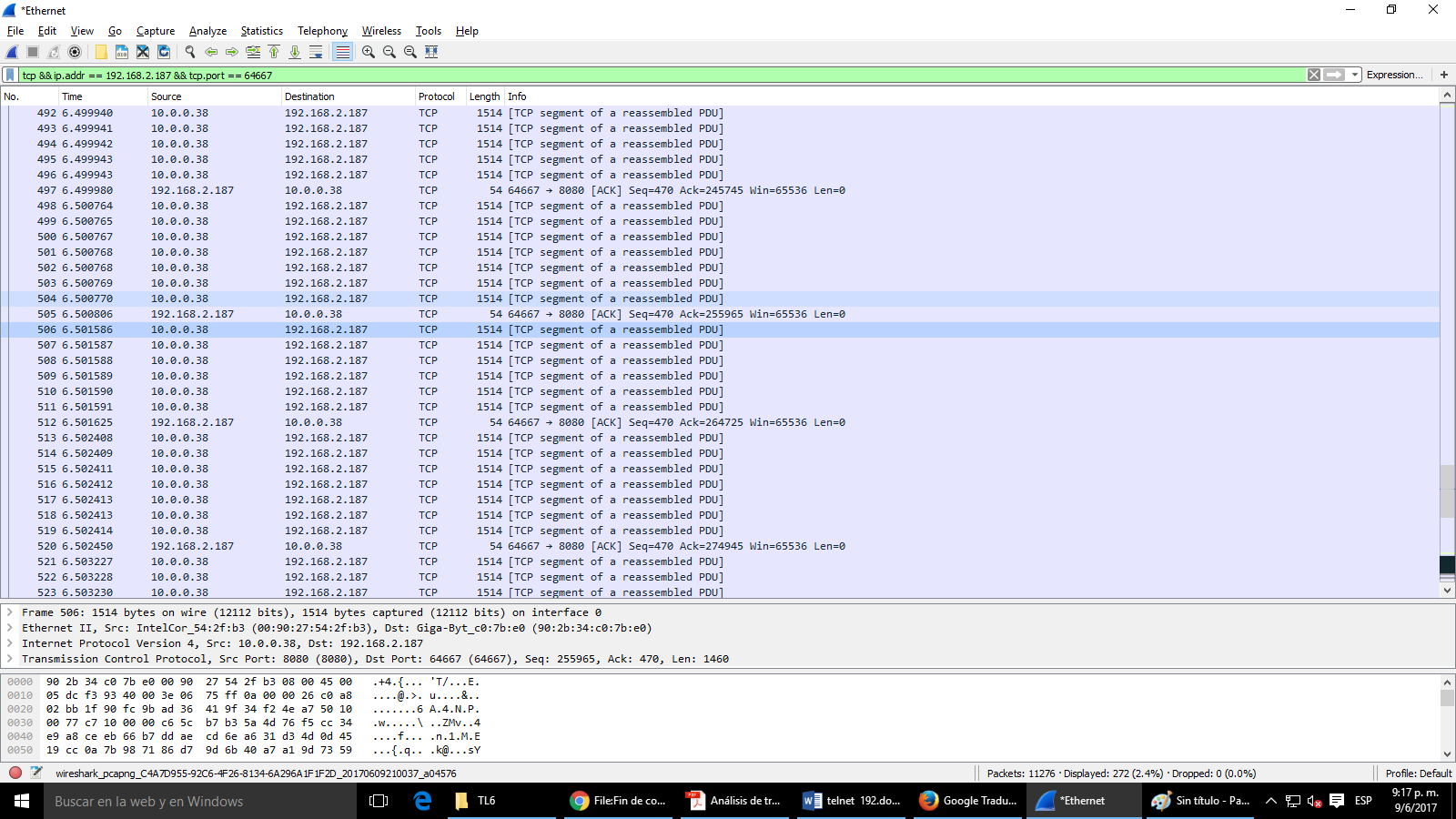
b) Conexiones lógicas. La confiabilidad y los mecanismos de control de flujo descritos anteriormente requieren que TCP inicializa y mantenga cierta información de estado para cada "flujo de datos". La combinación de este estado, incluyendo sockets, números de secuencia y tamaños de ventana, se llama conexión lógica. Cada conexión se identifica unívocamente por la pareja de sockets usados por los procesos emisor y receptor.

Según <https://es.stackoverflow.com/questions/7170/cu%C3%A1l-es-la-diferencia->entre-socket-tcp-ip-websockets-y-request-http :

Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

Socket = (IP local, IP remota, protocolo transporte, Puerto local y Puerto remoto).

c) Se usan 254 tramas (544 - 290). Algunas de las tramas son:



La clave es entender que se usan 3 segmentos para el establecimiento de la conexión, 4 segmentos para la liberación de la conexión y todos los demás segmentos son para datos.

d) -Ordenamiento y reensamble: se usa número de secuencia y ACK.

-Confiabilidad: TCP asigna un número de secuencia a cada byte transmitido y espera por un reconocimiento positivo (ACK) del receptor TCP. Si el ACK no se recibe en un intervalo fijado, los datos se retransmiten. Como los datos se transmiten en bloques (segmentos TCP) sólo el número de secuencia del primer byte de datos en los segmentos se envían al host destino.

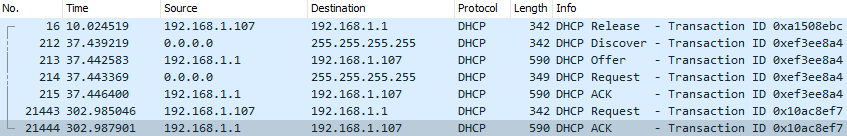
-Fiabilidad y control de errores: mediante CRC-16 y ARQ para retransmisión

-Control de flujo con ACK y NAK: El receptor TCP, cuyo envía un ACK de vuelta al emisor, también indica al emisor el número de bytes que puede recibir más allá del último segmento TCP recibido sin causar ni overrun ni desbordamiento en sus búferes internos. Este se envía en el ACK de forma of the highest sequence number it puede recibir sin problemas. Este mecanismo también se denomina mecanismo ventana y se discutirá con más detalle más tarde en este mismo capítulo.

-Multiplexacion: Se logra usando puertos como en udp.

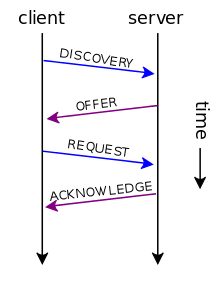
-Half dúplex porque transmite primero en un sentido, y después en el otro, full dúplex no puede ser ya que sino se chocarían cuando transmite.

e) 4 segmentos necesitamos, el print screen es el que se encuentra en el punto 2a).

1. **b. Análisis del tráfico que produce un protocolo no orientado a la conexión.**
2. 1) UDP: hay que tener en cuenta que es no orientado a la conexión, puede haber muchos caminos para llegar de un extremo a otro, usa datagramas.
3. Hago ipconfig /release en línea de comandos y me fijo el protocolo DHCP, filtro por los puertos que están involucrados, 67 y 68 en este caso, haciendo udp.port == 67 && up.port==68. Despues hago ipconfig /renew y obtengo lo siguiente:
4. 
5. Tuvimos en cuenta este video:

<https://www.youtube.com/watch?v=avCdgBtP5uU>

1. El grafico de lo que ocurre es lo siguiente:

[](https://commons.wikimedia.org/wiki/File:DHCP_session_en.svg)

DHCP Discovery es una solicitud DHCP realizada por un cliente de este protocolo para que el servidor DHCP de dicha red de computadoras le asigne una dirección IP y otros [Parámetros DHCP](https://es.wikipedia.org/wiki/Par%C3%A1metros_DHCP) como la máscara de red o el nombre DNS.

DHCP Offer es el paquete de respuesta del Servidor DHCP a un cliente DHCP ante su petición de la asignación de los Parámetros DHCP. Para ello involucra su dirección MAC (Media Access Control).

El cliente selecciona la configuración de los paquetes recibidos de *DHCP Offer*. Una vez más, el cliente solicita una dirección IP específica que indicó el servidor

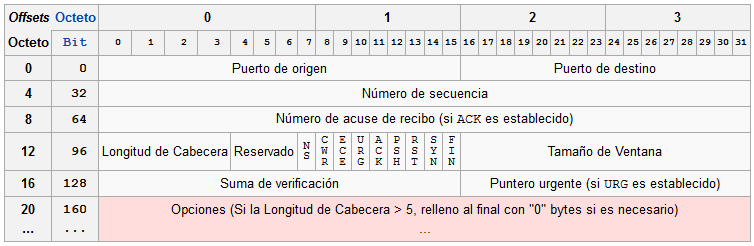
El servidor reconoce la solicitud y la envía acuse de recibo al cliente del cliente, se inicia la fase final del proceso de configuración. Esta fase implica el reconocimiento con el envío de un paquete al cliente.

1. Conexión: No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.

Fiabilidad y control de errores:  [suma de verificación](https://es.wikipedia.org/wiki/Suma_de_verificaci%C3%B3n) de la cabecera (checksum 16 bits).  El checksum también es opcional, aunque generalmente se utiliza en la práctica.

Control de flujo: no tiene, tampoco ordena paquetes.  
  
Multiplexación: posee multiplexado de aplicación, tiene puerto de origen y destino (el origen es opcional)

1. Encapsulamiento TCP:

[](https://commons.wikimedia.org/wiki/File:CabeceraTCP.png)

**Encapsulado de UDP:**



**C y D)**

|  |  |  |
| --- | --- | --- |
| Puerto | Nombre | Descripcion |
| 67 | bootps | Servicios del Protocolo Bootstrap o de inicio (BOOTP); también usado por los servicios del protocolo de configuración dinámica de host (DHCP). |
| 68 | bootpc | Cliente bootstrap (BOOTP); también usado por el protocolo de configuración dinámica de host (DHCP) |
| 546 | dhcpv6-client | Cliente DHCP, Protocolo de configuración dinámica de host, versión 6 |
| 547 | dhcpv6-server | Servicio DHCP, protocolo de configuración dinámica de host, versión 6 |
| 80 | http | Protocolo de transferencia de hipertexto (HTTP) para los servicios del World Wide Web (WWW) |
| 488 | gss-http | Servicios de seguridad genérica (GSS) para HTTP |
| 443 | https | Protocolo de transferencia de hipertexto seguro (HTTP) |
| 20 | ftp-data | Puerto de datos FTP |
| 21 | ftp | Puerto del Protocolo de transferencia de archivos (FTP); algunas veces utilizado por el Protocolo de servicio de archivos (FSP). |
| 245 | link | Servicio LINK / 3-DNS iQuery |

Links usados: <http://personales.upv.es/rmartin/tcpip/cap02s12.html>

<https://blog.smaldone.com.ar/2006/11/21/tutorial-sobre-tcpip/>

<https://es.wikibooks.org/wiki/Redes_inform%C3%A1ticas/Protocolos_TCP_y_UDP_en_el_nivel_de_transporte>