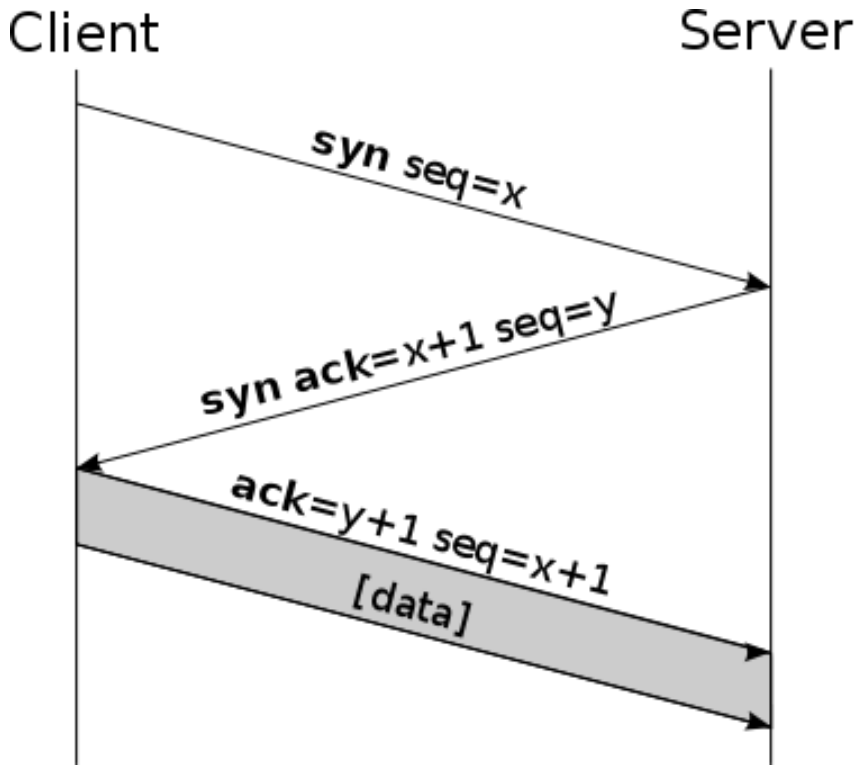
Capa de transporte - Explicación de práctica

Redes y Comunicaciones 2015

1. **TCP**
   - **Apertura del canal**
   - **Análisis Numero de Secuencia**
   - **Cierre del canal**
2. UDP
3. Servicios

TCP - Apertura del canal

Análisis con Wireshark

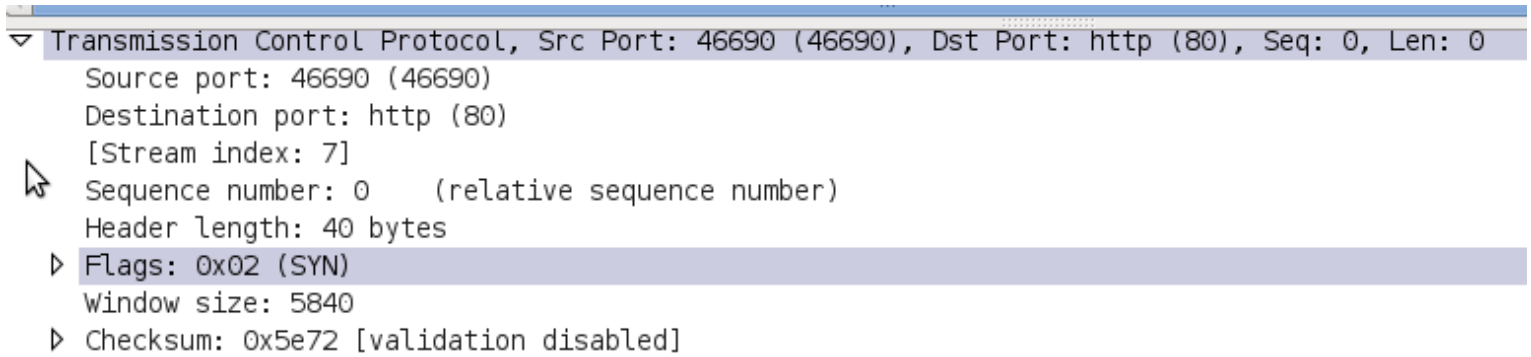


163.10.10.96	163.10.5.91	TCP	46690 > http [SYN] Seq=0 Win=5840 Len=0 MS
163.10.5.91	163.10.10.96	TCP	http > 46690 [SYN, ACK] Seq=0 Ack=1 Win=57
163.10.10.96	163.10.5.91	TCP	46690 > http [ACK] Seq=1 Ack=1 Win=5888 Le
163.10.10.96	163.10.5.91	HTTP	GET / HTTP/1.1
163.10.5.91	163.10.10.96	TCP	http > 46690 [ACK] Seq=1 Ack=543 Win=6912
163.10.10.96	199.59.149.230	TCP	56841 > https [SYN] Seq=0 Win=5840 Len=0 M

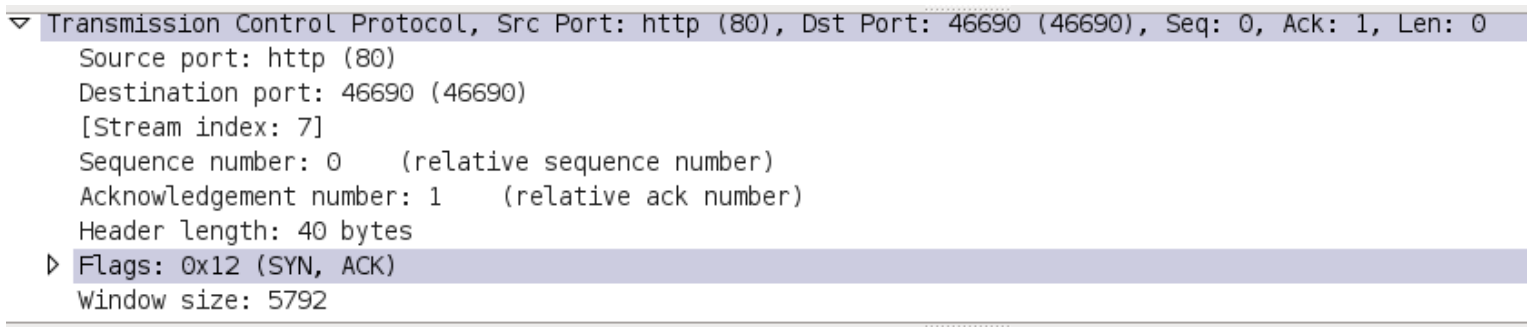
TCP - Apertura del canal

Análisis con Wireshark

1. Seq=0 PuertoOrigen=46690 PuertoDestino=80



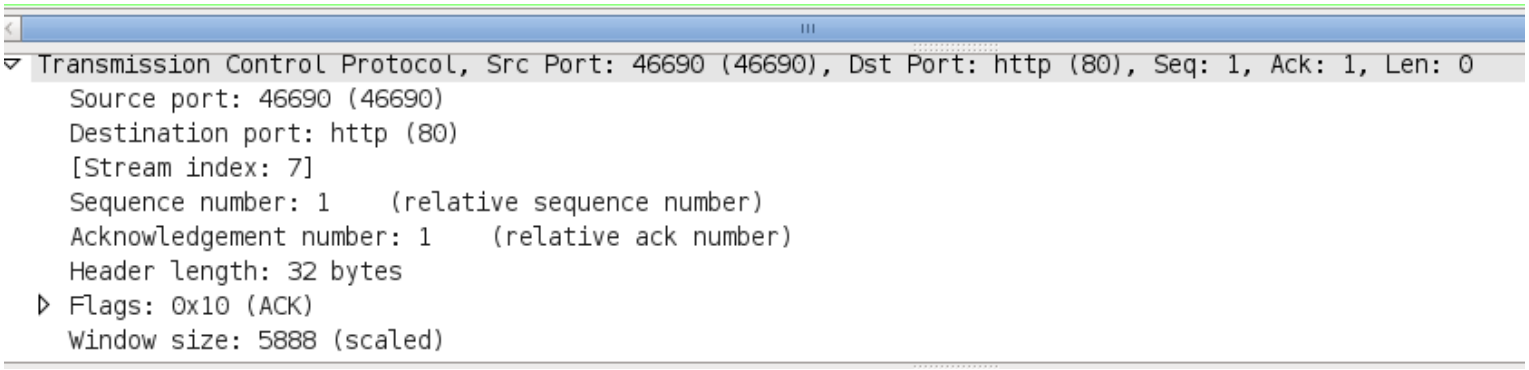
2. Seq=0 Ack=1 PuertoOrigen=80 PuertoDestino=46690



TCP - Apertura del canal

Análisis con Wireshark

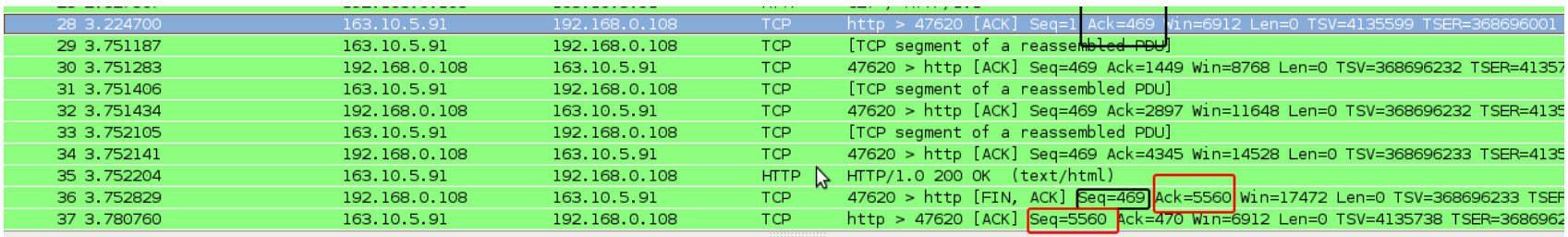
Seq=1 Ack=1 PuertoOrigen=46690 PuertoDestino=80



Análisis Numero de Secuencia TCP

Análisis con Wireshark

Cuando se inicia la conexión TCP, el Número de Secuencia inicial es un número impredecible. Para mejorar la claridad y el análisis, Wireshark muestra tales números de forma relativa al inicio de la conexión.



N° SEQ = ACK anterior

ACK = 469 ======>>>>> N° SEQ = 469

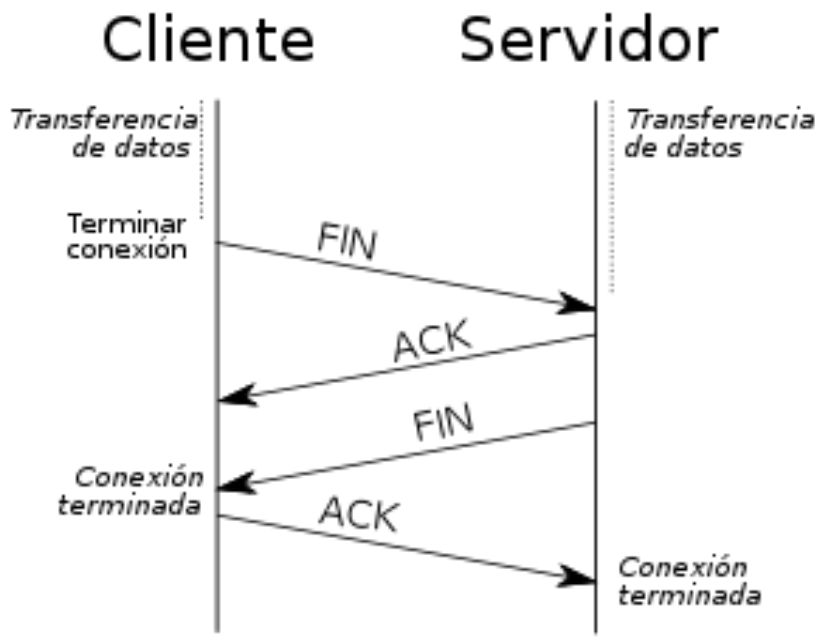
TCP utiliza FLAGS como mecanismo de señalización.

SYN, RST, ACK, FIN

TCP - Cierre del canal

La fase de finalización usa una negociación en cuatro pasos (four-way handshake)

- Conexión FullDuplex. Cada lado debe cerrarse independientemente.
- Para cerrar la conexión, se envía un FIN y se espera un ACK.
- La conexión puede quedar "medio abierta"si uno de los lados finaliza la conexión pero el otro no. El lado que ha dado por finalizada la conexión no puede enviar más datos pero la otra parte si podrá.



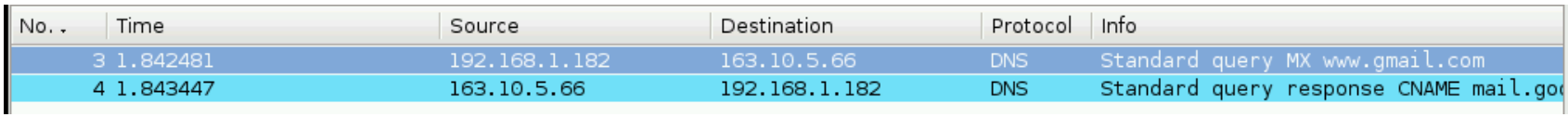
Explicación Capa de transporte

1. TCP
   - Apertura del canal
   - Análisis Numero de Secuencia
   - Cierre del canal
2. **UDP**
3. Servicios

UDP

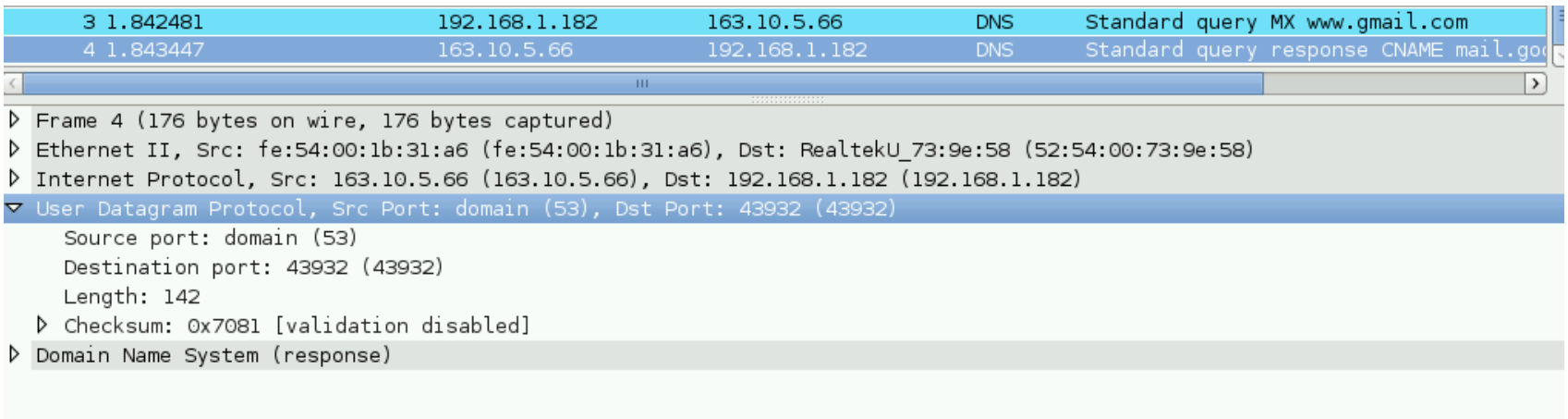
- UDP NO es orientado a la conexión.
- No hay flags ni mecanismos de señalización como en TCP.
- UDP no es un protocolo de transporte confiable

Análisis consulta DNS con Wireshark



No.	Time	Source	Destination	Protocol	Info
3	1.842481	192.168.1.182	163.10.5.66	DNS	Standard query MX www.gmail.com
4	1.843447	163.10.5.66	192.168.1.182	DNS	Standard query response CNAME mail.goo

Puerto origen - puerto destino



3 1.842481	192.168.1.182	163.10.5.66	DNS	Standard query MX www.gmail.com
4 1.843447	163.10.5.66	192.168.1.182	DNS	Standard query response CNAME mail.goo

```
▷ Frame 4 (176 bytes on wire, 176 bytes captured)
▷ Ethernet II, Src: fe:54:00:1b:31:a6 (fe:54:00:1b:31:a6), Dst: RealtekU_73:9e:58 (52:54:00:73:9e:58)
▷ Internet Protocol, Src: 163.10.5.66 (163.10.5.66), Dst: 192.168.1.182 (192.168.1.182)
▽ User Datagram Protocol, Src Port: domain (53), Dst Port: 43932 (43932)
    Source port: domain (53)
    Destination port: 43932 (43932)
    Length: 142
  ▷ Checksum: 0x7081 [validation disabled]
▷ Domain Name System (response)
```

Explicación Capa de transporte

1. TCP
   - Apertura del canal
   - Análisis Numero de Secuencia
   - Cierre del canal
2. UDP
3. **Servicios**

Estado de las conexiones

Las conexiones TCP se pueden examinar con el comando: netstat -nat

```
root@lihuen:~# netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.182:34459     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:34452     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:36471     173.194.37.49:80        ESTABLISHED
tcp        1      0 192.168.1.182:50450     209.191.122.70:80       CLOSE_WAIT
tcp        0      0 192.168.1.182:52471     184.169.70.33:80        ESTABLISHED
tcp        0      0 192.168.1.182:34456     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:40853     72.21.211.176:80        ESTABLISHED
tcp        0      0 192.168.1.182:53692     67.196.156.65:80        ESTABLISHED
tcp        0      0 192.168.1.182:34460     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:34455     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:47286     173.194.37.37:443       ESTABLISHED
tcp        0      0 192.168.1.182:52470     184.169.70.33:80        ESTABLISHED
tcp        0      0 192.168.1.182:34458     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:56089     66.135.210.181:80       ESTABLISHED
tcp        0      0 192.168.1.182:50381     208.80.154.225:80       ESTABLISHED
tcp        0      0 192.168.1.182:46814     199.59.148.201:80       ESTABLISHED
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
```

Conexiones escuchando: LISTEN

Conexiones establecidas: ESTABLISHED

1. ¿Como se pueden examinar las conexiones UDP?

Estado de las conexiones

Puertos Origen y Destino

Conexiones ESTABLECIDAS con 163.10.5.91

```
root@lihuen:~# netstat -nat | grep 163.10.5.91
tcp        0      0 192.168.1.182:41009     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:41013     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:41008     163.10.5.91:80          TIME_WAIT
tcp        0      0 192.168.1.182:41012     163.10.5.91:80          ESTABLISHED
tcp        0      0 192.168.1.182:41011     163.10.5.91:80          ESTABLISHED
```

¿Qué puerto origen y destino tiene cada una de las conexiones establecidas?

Puertos y Procesos asociados a las conexiones

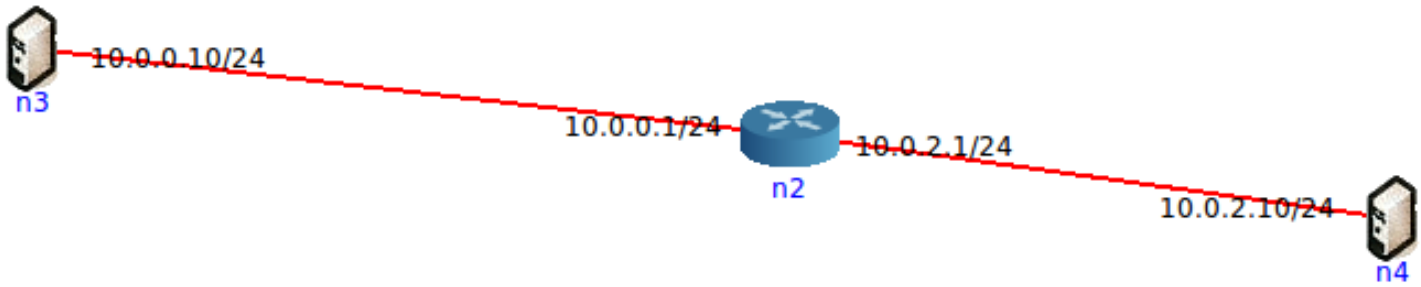
También se puede observar el proceso asociado a las conexiones con el parámetro -p de netstat

```
root@lihuen:~# netstat -nat -p
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1405/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      1316/cupsd
tcp        0      0 192.168.1.182:44454     173.194.37.36:80        ESTABLISHED 24528/firefox-bin
tcp6       0      0 :::22                   :::*                    LISTEN      1405/sshd
tcp6       0      0 ::1:631                 :::*                    LISTEN      1316/cupsd
root@lihuen:~#
```

- ¿Qué procesos representan a servicios que están corriendo? Los que están con estado LISTEN
- ¿Qué puerto está asociado con el servicio sshd?
- ¿Qué puerto está asociado con el servicio cupsd?
- ¿Cuál es el puerto origen y el puerto destino asociado a la conexión establecida?

Puede utilizar el archivo /etc/services para examinar el puerto por defecto que utilizan los distintos protocolos de aplicación

Utilizando CORE



- Crear topología
- Probar conectividad
- Levantar servicios con ncat
- Utilizar ncat como cliente