

# INF-256 Redes de Computadores

## Laboratorio 1

Gabriel Carmona    Jorge Ludueña  
201773509-8        201773507-4

4 de mayo de 2020

1. Se espera encontrar los dos protocolos de transportes utilizados, el TCP y UDP. Por otro lado también esperamos ver el protocolo HTTP, puesto que realizamos una llamada a una página web. Al momento de utilizar Wiresharks, nos encontramos con que efectivamente se utilizan estos protocolos, pero también se observó el protocolo IP que viene de la capa de red, lo cual tiene bastante sentido puesto que se envía información a través de la capa de transporte y para esto debe pasar por la capa de red, link y física (bits) para que llegue al receptor.

```
▼ Frame 106772: 93 bytes on wire (744 bits), 53 bytes captured (424 bits) on interface \Device\NPF_{...}_Loopback, id 0
  > Interface id: 0 (\Device\NPF_{...}_Loopback)
    Encapsulation type: NULL/Loopback (15)
    Arrival Time: May 3, 2020 18:19:17.260932000 Hora est. Sudamérica Pacífico
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1588544357.260932000 seconds
    [Time delta from previous captured frame: 0.108466000 seconds]
    [Time delta from previous displayed frame: 4.256228000 seconds]
    [Time since reference or first frame: 2045.727026000 seconds]
    Frame Number: 106772
    Frame Length: 93 bytes (744 bits)
    Capture Length: 53 bytes (424 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: null:ip:tcp:data]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
```

Aquí hay un ejemplo de un frame donde se pueden ver esos 2 protocolos IP y data.

2. Si bien el cliente y el servidor tienen asignados puertos distintos, a lo largo del tiempo en las interacciones vía TCP, ellos ocuparán los mismos puertos. Esto se debe principalmente al hecho de que se realice el handshaking, ya que eso mantiene la comunicación. Sí, coincide con lo visto en Wireshark, ya que se ve que al momento de realizar consultas los puertos en interacciones TCP se mantienen, como se aprecia en la imagen de abajo.

723	18.861806	127.0.0.1	127.0.0.1	TCP	84	50537 → 50366 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
784	22.791569	127.0.0.1	127.0.0.1	TCP	98	50537 → 50366 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=14
785	22.791619	127.0.0.1	127.0.0.1	TCP	84	50366 → 50537 [ACK] Seq=1 Ack=15 Win=2619648 Len=0
794	22.947938	127.0.0.1	127.0.0.1	TCP	89	50366 → 50537 [PSH, ACK] Seq=1 Ack=15 Win=2619648 Len=5
795	22.947981	127.0.0.1	127.0.0.1	TCP	84	50537 → 50366 [ACK] Seq=15 Ack=6 Win=2619648 Len=0
796	22.952632	127.0.0.1	127.0.0.1	UDP	62	50149 → 49152 Len=2
797	22.953223	127.0.0.1	127.0.0.1	UDP	792	49152 → 50149 Len=732
1779	44.472791	127.0.0.1	127.0.0.1	TCP	100	50537 → 50366 [PSH, ACK] Seq=15 Ack=6 Win=2619648 Len=16
1780	44.472836	127.0.0.1	127.0.0.1	TCP	84	50366 → 50537 [ACK] Seq=6 Ack=31 Win=2619648 Len=0
1813	45.230919	127.0.0.1	127.0.0.1	TCP	89	50366 → 50537 [PSH, ACK] Seq=6 Ack=31 Win=2619648 Len=5
1814	45.230956	127.0.0.1	127.0.0.1	TCP	84	50537 → 50366 [ACK] Seq=31 Ack=11 Win=2619648 Len=0
1815	45.231809	127.0.0.1	127.0.0.1	UDP	62	62026 → 49152 Len=2
1816	45.232349	127.0.0.1	127.0.0.1	UDP	155	49152 → 62026 Len=95

3. Los contenidos de los mensajes no son legibles para un humano, debido a que se encuentran en hexadecimal, Wireshark nos permite obtener dicho número y con algún programa o página web que traduzca de hexadecimal a ASCII se puede obtener el mensaje en texto.

En este caso los programas de servidor y cliente son capaces de transformarlos a texto gracias al metodo de decodificaion que posee python.

▼ Data (14 bytes)  
Data: 7777772e676f6f676c652e636f6d  
[Length: 14]

En la imagen, se observa un mensaje entre aplicaciones, este está en hexadecimal y corresponde a cuando el cliente le pregunto por www.google.com al servidor.

4. Wireshark permite ver el mensaje enviado por la pagina web, en la cual se observa que el header posee diferencias mínimas con el header que recibe el cliente. Algunas de estas diferencias son que hay ciertos detalles especificado como para dejar más claro, se puede ver en la imagen podemos ver como se especifica a que es 301, status code, o HTTP/1.1, que es la version de respuesta. Pero si no en si entregan el mismo header.

```
Hypertext Transfer Protocol
▼ HTTP/1.1 301 Moved Permanently\r\n
  > [Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
    Response Version: HTTP/1.1
    Status Code: 301
    [Status Code Description: Moved Permanently]
    Response Phrase: Moved Permanently
    Date: Sun, 03 May 2020 22:34:09 GMT\r\n
    Server: Apache\r\n
    Location: http://81dojo.com/\r\n
  ▼ Content-Length: 290\r\n
    [Content length: 290]
    Connection: close\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.302526000 seconds]
    [Request in frame: 2962]
    [Request URI: http://www.81dojo.com/]
    File Data: 290 bytes
  Use based text data: text/html (0 lines)
```

```
HTTP/1.1 301 Moved Permanently
Date: Sun, 03 May 2020 22:34:09 GMT
Server: Apache
Location: http://81dojo.com/
Content-Length: 290
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

5. Bonus: En las interacciones con un cliente y muchos clientes a la vez, la mayor diferencia que se puede observar es la presencia de más puertos, puesto que cada puerto corresponde a un cliente diferente.

75668	1442.280373	127.0.0.1	127.0.0.1	TCP	84 51886 → 50366 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
76274	1459.229778	127.0.0.1	127.0.0.1	TCP	100 50537 → 50366 [PSH, ACK] Seq=31 Ack=11 Win=2619648 Len=16
76275	1459.229821	127.0.0.1	127.0.0.1	TCP	84 50366 → 50537 [ACK] Seq=11 Ack=47 Win=2619648 Len=0
76304	1459.469118	127.0.0.1	127.0.0.1	TCP	89 50366 → 50537 [PSH, ACK] Seq=11 Ack=47 Win=2619648 Len=5
76305	1459.469158	127.0.0.1	127.0.0.1	TCP	84 50537 → 50366 [ACK] Seq=47 Ack=16 Win=2619648 Len=0
76306	1459.470344	127.0.0.1	127.0.0.1	UDP	62 58561 → 49152 Len=2
76307	1459.471117	127.0.0.1	127.0.0.1	UDP	375 49152 → 58561 Len=315
76308	1459.525883	127.0.0.1	127.0.0.1	TCP	98 51886 → 50366 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=14
76309	1459.525923	127.0.0.1	127.0.0.1	TCP	84 50366 → 51886 [ACK] Seq=1 Ack=15 Win=2619648 Len=0
76310	1459.526366	127.0.0.1	127.0.0.1	TCP	89 50366 → 51886 [PSH, ACK] Seq=1 Ack=15 Win=2619648 Len=5
76311	1459.526389	127.0.0.1	127.0.0.1	TCP	84 51886 → 50366 [ACK] Seq=15 Ack=6 Win=2619648 Len=0
76312	1459.527227	127.0.0.1	127.0.0.1	UDP	62 58562 → 49152 Len=2
76313	1459.527699	127.0.0.1	127.0.0.1	UDP	792 49152 → 58562 Len=732

En la imagen, se puede observar como están interacciones correspondientes a un cliente (color amarillo) y a otro cliente (color rojo).