

Guía de Ejercicios - 2

Nivel de Aplicación

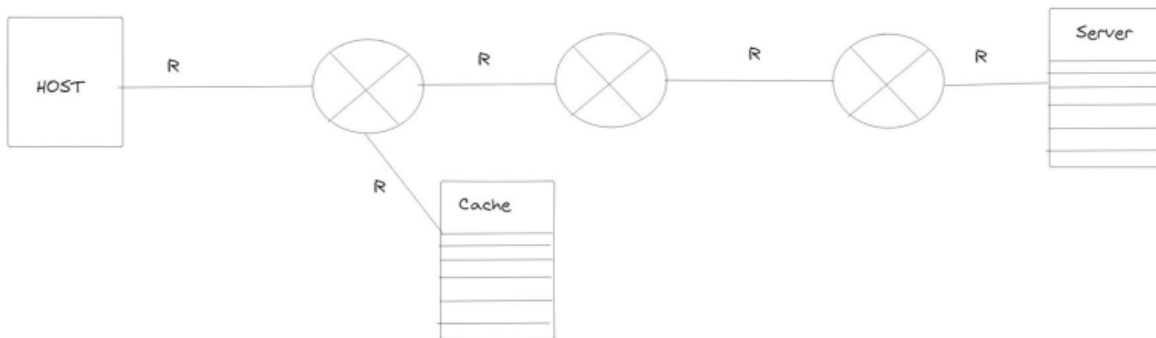
- I. Recurrir a la especificación de HTTP/1.1 ([RFC 2616](#)) para responder brevemente las siguientes preguntas:
 - a. Explicar el mecanismo que se utiliza para indicar el cierre de una conexión persistente. ¿Quiénes pueden cerrar una conexión? ¿El cliente? ¿El servidor? ¿Ambos?
 - b. ¿Qué servicios de encriptación provee HTTP?
 - c. ¿Puede un cliente abrir 3 o más conexiones simultáneas con un mismo servidor?
 - d. Tanto un servidor como un cliente pueden cerrar una conexión a nivel transporte si alguno de ellos detecta que la conexión ha estado inactiva durante algún tiempo. ¿Es posible que uno de los hosts comience a cerrar una conexión mientras que el otro está transmitiendo datos a través de esa misma conexión? Explicar.
 - e. Considerando la evolución del protocolo a la versiones posteriores a HTTP/1.1, describir de qué se trata el problema de *HOL (head-of-line) blocking* y cómo se aborda en HTTP/2.
2. El siguiente texto muestra la respuesta enviada desde el servidor a un mensaje HTTP GET. Responder las siguientes preguntas, indicando en qué parte del mensaje se encuentra la respuesta.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46

GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>

Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public
"-//w3c//dtd html 4.0transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows
NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550ASpring 2005 homepage</title><lf></head><lf>
<...>
```

- a. ¿El servidor pudo encontrar con éxito el documento o no?
 - b. ¿A qué hora el servidor envió la respuesta?
 - c. ¿Cuándo se modificó el documento por última vez?
 - d. ¿Cuántos bytes hay en el documento que se devuelve?
 - e. ¿El servidor aceptó establecer una conexión persistente?
3. Tenemos una red en la cual, cuatro personas en horarios distintos del día quieren acceder al mismo archivo. El archivo tiene tamaño L y los paquetes de request y consulta tienen tamaño k . (Ignorar tiempos de encolamiento, propagación y procesamiento)
 - a. Calcular cuánto es la demora de las cuatro personas sin usar caché.
 - b. Calcular el tiempo que tardaron si todas las personas primero van a preguntar a la caché. Si la caché tiene el archivo, lo devuelve. Si no, la persona va a buscarlo y cuando vuelva, va a dejar una copia a la caché.



4. Supongamos que Alina, con una cuenta de correo electrónico basada en la Web (como Hotmail o Gmail), envía un mensaje a Benjamín, quien accede al mensaje en su servidor de correo mediante IMAP. Discutir cómo llega el mensaje del host de Alina al host de Benjamín. Prestar especial atención a enumerar los distintos protocolos de capa de aplicación que se utilizan para mover el mensaje entre los dos hosts.
5. ¿Es posible que el servidor web y el servidor de correo electrónico de una organización tengan exactamente el mismo alias para un nombre de host (por ejemplo, `foo.com`)? ¿Cuál sería el tipo de *resource record (RR)* de DNS que contiene el nombre de host del servidor de correo?

-
6. Supongamos que Alina se une a la red BitTorrent y todavía no posee piezas (o *chunks*) del archivo que desea descargar. Sin *pieces* disponibles, Alina no podrá convertirse en uno de los 4 *peers* principales de ninguno de los otros *peers*, dado que no tiene nada que enviar. ¿Cómo podrá conseguir Alina su primera *piece*?
 7. Luego del parcial de TD4, Valentín toma su celular en el colectivo camino a casa y se dispone a mirar un streaming de YouTube de su influencer favorito. Para ello, abre el Chrome, ingresa a www.youtube.com y luego presiona sobre la imagen en miniatura del video que aparece en la home del sitio.
 - a. Mencionar por lo menos tres protocolos de aplicación que hayan intervenido a lo largo de este proceso (incluyendo la reproducción del contenido multimedia). ¿Qué función cumple cada uno de ellos?
 - b. Durante los primeros 40 segundos del streaming, Valentín nota que las imágenes se ven muy pixeladas y que la calidad del video deja mucho que desear. No obstante, esta situación cambia repentinamente y por el resto del viaje el video se reproduce con una excelente calidad. Explicar en detalle a qué puede deberse esto.
 8. Supongamos que, dentro de nuestro navegador web, hacemos click en un enlace para obtener una página. La dirección IP de la URL solicitada no está almacenada en la caché local, por lo que es necesario hacer una búsqueda DNS. En esa búsqueda, se visitan sucesivamente n servidores DNS antes de obtener la dirección IP. Cada una de esas visitas incurren en diferentes RTTs que denominaremos $RTT_1, RTT_2, \dots, RTT_n$. Además, debemos suponer que la página que solicitamos contiene exactamente un objeto: una pequeña cantidad de texto HTML. Si RTT_0 es el *round-trip time* entre el host local y el servidor que contiene el objeto, y asumiendo que el tiempo de transferencia del objeto es 0, ¿Cuánto tiempo pasa desde que el cliente hace click en el enlace hasta que recibe el objeto?
 9. Supongamos que un archivo HTML hace referencia a ocho objetos muy pequeños en el mismo servidor. Despreciando los tiempos de transmisión, calcular cuánto tiempo (medido en RTT al servidor) transcurre si usamos:
 - a. HTTP no persistente sin conexiones TCP paralelas.
 - b. HTTP no persistente con el navegador configurado para permitir 6 conexiones paralelas.

c. HTTP persistente.

10. Consideremos un enlace corto de 10 metros, sobre el cual se puede transmitir a una velocidad de 150 bits/seg en ambas direcciones. Supongamos que los paquetes que contienen datos tienen una longitud de 100.000 bits y los paquetes que solo contienen información de control (por ejemplo, ACK o *handshaking*) tienen una longitud de 200 bits. Supongamos, además, que N conexiones paralelas obtienen una fracción $1/N$ del ancho de banda del enlace cada una. Considerando el protocolo HTTP y suponiendo que cada objeto descargado tiene un tamaño de 100 kb (*kilobits*) y que el objeto descargado inicial contiene referencias a otros 10 objetos en el mismo servidor,

- ¿Tendrían sentido en este caso las descargas paralelas a través de instancias paralelas de HTTP no persistente?
- ¿Y con HTTP persistente? ¿Se pueden esperar ganancias significativas respecto al caso no persistente? Explicar y justificar la respuesta.

11. En cierto host se utilizó el comando `dig` para realizar una consulta DNS. En dicho host, además, estaba en ejecución Wireshark. Una vez finalizada la ejecución de `dig`, se encontró una traza de paquetes que detallamos a continuación. Analizar la traza para responder las siguientes preguntas:

- ¿Qué tipo de registro DNS se solicitó en la consulta?
- ¿Cuál es el servidor DNS local del host?
- ¿Cuántos servidores TLD para .com se ofrecieron al proceso de `dig`? ¿Cuál fue elegido?
- ¿Cuántos servidores autoritativos posee el dominio consultado?

No.	Time	Source	Destination
Protocol Length			
1051	1.420013	192.168.0.10	181.30.140.195
DNS	82		

Frame 67: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd), Dst: 76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01)
Internet Protocol Version 4, Src: 192.168.0.10, Dst: 181.30.140.195
User Datagram Protocol, Src Port: 47680, Dst Port: 53
Domain Name System (query)
Transaction ID: 0xf0cf
Flags: 0x0120 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 0
Queries
 <Root>: type NS, class IN
Additional records

No.	Time	Source	Destination
Protocol Length			
1052	1.435881	181.30.140.195	192.168.0.10
DNS	567		

Frame 69: 567 bytes on wire (4536 bits), 567 bytes captured (4536 bits) on interface \Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: 76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01), Dst: RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd)
Internet Protocol Version 4, Src: 181.30.140.195, Dst: 192.168.0.10
User Datagram Protocol, Src Port: 53, Dst Port: 47680
Domain Name System (response)
Transaction ID: 0xf0cf
Flags: 0x81a0 Standard query response, No error
Questions: 1
Answer RRs: 14
Authority RRs: 0
Additional RRs: 14
Queries
Answers
 <Root>: type NS, class IN, ns b.root-servers.net

Licenciatura en Tecnología Digital

TD4: Redes de Computadoras

```

<Root>: type NS, class IN, ns e.root-servers.net
<Root>: type NS, class IN, ns l.root-servers.net
(...)
Additional records
  b.root-servers.net: type A, class IN, addr 199.9.14.201
  e.root-servers.net: type A, class IN, addr 192.203.230.10
  l.root-servers.net: type A, class IN, addr 199.7.83.42
  (...)
---
No.      Time      Source      Destination
Protocol Length
1057 10.642201 192.168.0.10 199.7.83.42
DNS      100

Frame 1057: 100 bytes on wire (800 bits), 100 bytes captured (800
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd), Dst:
76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01)
Internet Protocol Version 4, Src: 192.168.0.10, Dst: 199.7.83.42
User Datagram Protocol, Src Port: 37356, Dst Port: 53
Domain Name System (query)
  Transaction ID: 0x80e8
  Flags: 0x0020 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    guerrillamail.com: type MX, class IN
  Additional records

---
No.      Time      Source      Destination
Protocol Length
1101 10.674204 199.7.83.42 192.168.0.10
DNS      1219

Frame 1101: 1219 bytes on wire (9752 bits), 1219 bytes captured (9752
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: 76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01), Dst:
RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd)
Internet Protocol Version 4, Src: 199.7.83.42, Dst: 192.168.0.10
User Datagram Protocol, Src Port: 53, Dst Port: 37356
Domain Name System (response)
  Transaction ID: 0x80e8
  Flags: 0x8000 Standard query response, No error
  Questions: 1
  Answer RRs: 0
  Authority RRs: 15
  Additional RRs: 27
  Queries
  Authoritative nameservers
    com: type NS, class IN, ns a.gtld-servers.net
    com: type NS, class IN, ns b.gtld-servers.net
    com: type NS, class IN, ns c.gtld-servers.net
    (...)
  Additional records
    a.gtld-servers.net: type A, class IN, addr 192.5.6.30
    b.gtld-servers.net: type A, class IN, addr 192.33.14.30
    c.gtld-servers.net: type A, class IN, addr 192.26.92.30
    (...)
[Time: 0.032003000 seconds]

---
No.      Time      Source      Destination
Protocol Length
1108 10.679169 192.168.0.10 192.33.14.30
DNS      100

Frame 1108: 100 bytes on wire (800 bits), 100 bytes captured (800
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd), Dst:
76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01)
Internet Protocol Version 4, Src: 192.168.0.10, Dst: 192.33.14.30
User Datagram Protocol, Src Port: 54089, Dst Port: 53
Domain Name System (query)
  Transaction ID: 0x9dea
  Flags: 0x0020 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    guerrillamail.com: type MX, class IN
  Additional records

---
No.      Time      Source      Destination
Protocol Length
1364 10.864175 192.33.14.30 192.168.0.10
DNS      953

Frame 1364: 953 bytes on wire (7624 bits), 953 bytes captured (7624
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: 76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01), Dst:
RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd)
Internet Protocol Version 4, Src: 192.48.79.30, Dst: 192.168.0.10
User Datagram Protocol, Src Port: 53, Dst Port: 54089
Domain Name System (response)
  Transaction ID: 0x9dea
  Flags: 0x8000 Standard query response, No error
  Questions: 1
  Answer RRs: 0
  Authority RRs: 2
  Additional RRs: 13
  Queries
  Authoritative nameservers
    guerrillamail.com: type NS, class IN, ns
rick.ns.cloudflare.com
    guerrillamail.com: type NS, class IN, ns
sara.ns.cloudflare.com
  Additional records
    rick.ns.cloudflare.com: type A, class IN, addr
172.64.33.139
    sara.ns.cloudflare.com: type A, class IN, addr
172.64.32.144
    (...)
[Time: 0.185006000 seconds]

---
No.      Time      Source      Destination
Protocol Length
1389 10.867911 192.168.0.10 172.64.32.144
DNS      100

Frame 1389: 100 bytes on wire (800 bits), 100 bytes captured (800
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd), Dst:
76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01)
Internet Protocol Version 4, Src: 192.168.0.10, Dst: 172.64.32.144
User Datagram Protocol, Src Port: 45544, Dst Port: 53
Domain Name System (query)
  Transaction ID: 0xd016
  Flags: 0x0020 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    guerrillamail.com: type MX, class IN
  Additional records

---
No.      Time      Source      Destination
Protocol Length
1690 10.902005 172.64.32.144 192.168.0.10
DNS      153

Frame 1690: 153 bytes on wire (1224 bits), 153 bytes captured (1224
bits) on interface
\Device\NPF_{770EDE64-4DF3-4B5A-B3EE-BE4EF66B103C}, id 0
Ethernet II, Src: 76:54:7d:f9:9f:01 (76:54:7d:f9:9f:01), Dst:
RivetNet_8c:66:cd (9c:b6:d0:8c:66:cd)
Internet Protocol Version 4, Src: 172.64.32.144, Dst: 192.168.0.10
User Datagram Protocol, Src Port: 53, Dst Port: 45544
Domain Name System (response)
  Transaction ID: 0xd016
  Flags: 0x8400 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 3
  Queries
  Answers
    guerrillamail.com: type MX, class IN, preference 0, mx
mail.guerrillamail.com
  Additional records
    mail.guerrillamail.com: type A, class IN, addr
168.119.142.36
    (...)
[Time: 0.034094000 seconds]

```

Ejercicios *hands-on*

1. Utilizar *Wireshark* para capturar un paquete HTTP. Sobre el paquete capturado responder:
 - e. ¿De qué tipo de mensaje se trata? ¿Es un *request*?
 - f. ¿Qué puertos de origen y destino se observan en el paquete?
 - g. Si el paquete era un *request*, buscar en la herramienta la respuesta del mismo.

2. *netcat* es una herramienta de networking de línea de comandos que permite enviar y recibir datos a través de la red. A modo de ejemplo, mostramos a continuación cómo abrir una conexión al puerto 80 de `www.utdt.edu` y acto seguido enviarle un *request* GET al servidor:

```
$ nc www.utdt.edu 80
GET / HTTP/1.1
Host: www.utdt.edu
```

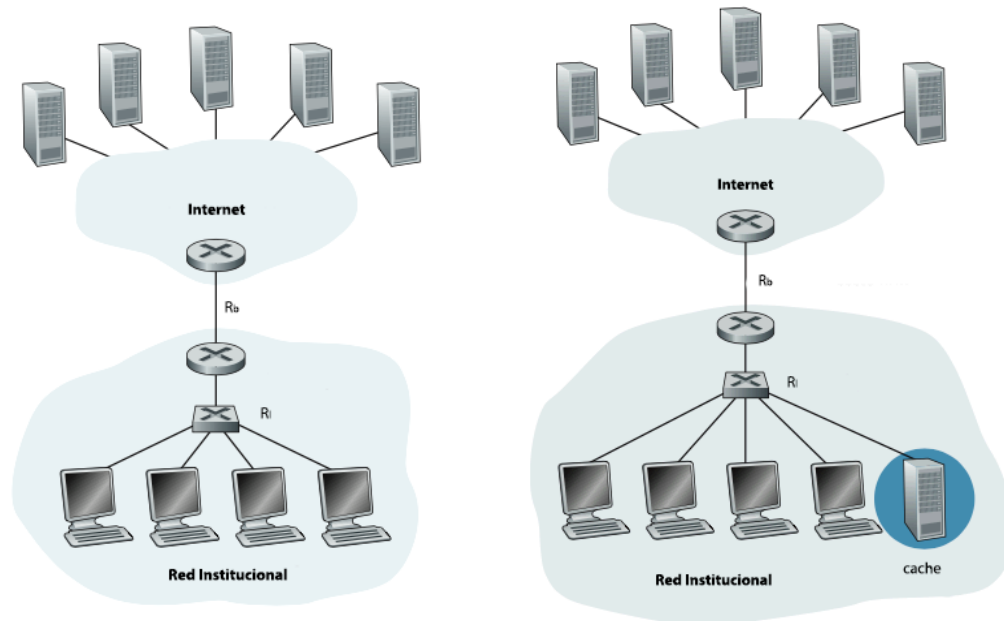
 - a. Ejecutar el *request* anterior en *netcat*. ¿Qué se observa en Wireshark? ¿Qué respuesta HTTP se obtiene? ¿Cómo se interpreta?
 - b. Inspeccionar el source HTML del sitio `httpforever.com` y utilizar *netcat* para solicitar un archivo referenciado desde el HTML alojado en dicho servidor. ¿Qué respuesta HTTP se obtiene en este caso?

3. Escribir un programa en Python para instanciar un socket, enviar el request del ejercicio anterior al mismo host y recibir la respuesta. ¿Qué se observa en Wireshark?

4. Ensamblar una *query* DNS en Scapy para consultar los registros de tipo A del dominio `www.utdt.edu`.
 - a. Investigar cómo obtener la dirección del servidor DNS local de tu computadora. Enviar luego el mensaje anterior a dicho servidor y esperar la respuesta del mismo. ¿Qué campos DNS contiene? Imprimir el paquete con la primitiva `show2` de Scapy para responder esta pregunta.
 - b. Repetir el punto anterior utilizando como resolvedor DNS el host 8.8.8.8 en vez del servidor DNS local configurado en tu computadora.

Ejercicios Opcionales

I. Supongamos los siguientes escenarios en una red institucional:



- La tasa de transmisión en la red institucional es R_i
- La tasa de transmisión del enlace de acceso a Internet es R_b
- La red institucional tiene N clientes que solicitan, uno tras otro, un mismo archivo grande de tamaño L via HTTP.

¿Para qué valores de R_i el tiempo de transmisión total es más chico cuando se instala una caché en la red institucional (figura de la derecha)?