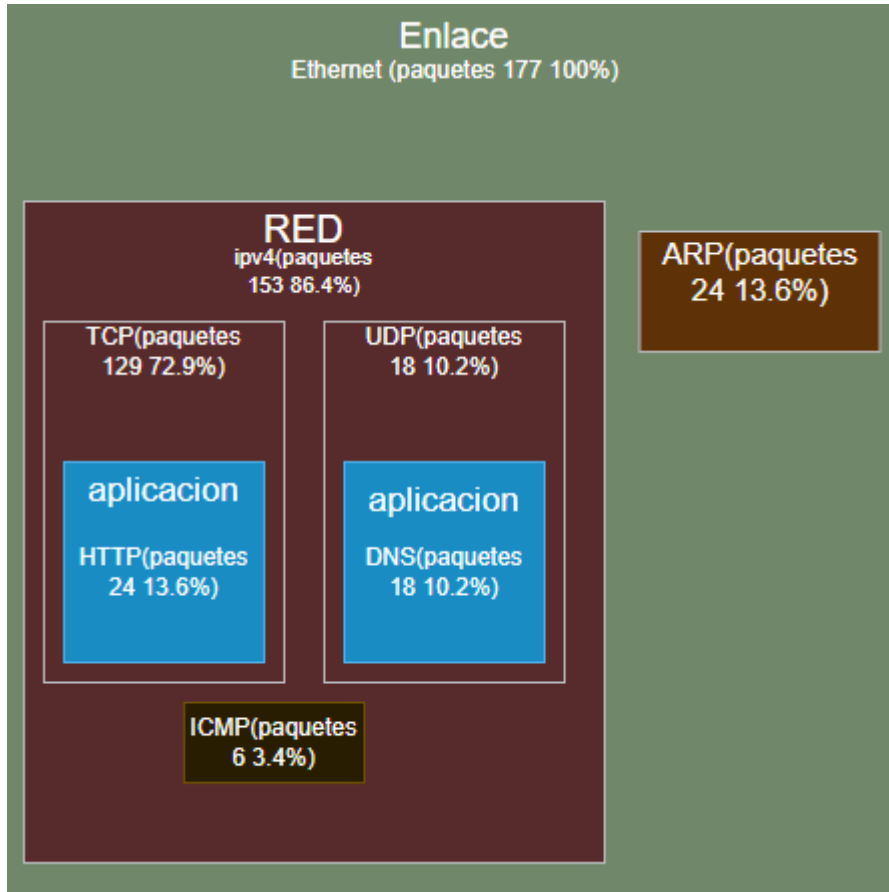


## Trabajo Práctico Final Integrador

### Configuración de Redes y Análisis de protocolos

2. Confeccione cuadros con la distribución de mensajes por protocolo por capa (Indique claramente qué protocolo/s ubica en cada capa y por qué).



**Capa de Enlace (1):** En esta capa se utiliza el protocolo Ethernet ya que este se utiliza para enviar y recibir tramas en el ámbito de redes locales (LAN). Es el protocolo estándar en la actualidad ya que se provee flexibilidad, alta velocidad y capacidad de datos, además de que es una opción rentable en comparación a otras tecnologías de red. Define el formato de las tramas, direcciones MAC y diferentes técnicas de detección de colisiones.

En estas tramas se encapsulan PDUs de capas superiores y protocolos auxiliares como ARP. ARP se encapsula en capa de enlace y trabaja en conjunto con la capa de Red ya que permite resolver direcciones IP a direcciones MAC. Estas direcciones son las que permiten la comunicación dentro de una red local.

**Capa de Red (2):** En esta capa se utiliza el protocolo IPv4. Este es el estándar del funcionamiento de Internet. Se manejan a través de direcciones IP que identifican unívocamente a cada dispositivo de Internet y se encarga del enrutamiento de paquetes (datagramas) a través de conmutadores llamados routers.

Utiliza el protocolo complementario ICMP. Este protocolo se encarga de enviar mensajes de control y errores por la red IP.

**Capa de Transporte (3):** En esta capa solo hay dos protocolos: TCP y UDP.

TCP se encarga de mantener una conexión fiable entre dos dispositivos finales y es orientado a la conexión. Provee control de flujo, errores y congestión mediante diversos mecanismos. Cada conexión se identifica por un par de sockets, estos son la concatenación de Dirección IP y puerto origen y Dirección IP y puerto destino. Los puertos se utilizan para identificar un servicio y así poder multiplexar varios de estos en una misma conexión.

UDP es un protocolo no orientado a la conexión y es utilizado por aplicaciones que necesitan una entrega rápida pero son pocos fiables ya que no realiza control de ningún tipo, a excepción del checksum.

**Capa de Aplicación (4):** Se encuentran los diferentes protocolos y servicios que permiten interactuar en la red. En esta captura utilizamos HTTP que utiliza TCP para el envío y recuperación de páginas web, y DNS que utiliza UDP para resoluciones de nombres.

**3. Identifique todas las conexiones TCP. Por cada una, indique: finalidad, sockets de cliente y servidor, segmentos de apertura y cierre. Muestre los parámetros intercambiados durante la fase de apertura.**

- Conexión entre pcusuario y proxy sockets<10.10.0.10:45896–10.10.0.1:3128>  
apertura:3-4-5 cierre:74-77  
**cliente:** W = 64240; ISN =2948180827; MSS = 1460; WS = 7;SACK: NOP.  
**servidor:** W = 65160; ISN =3026690716; MSS = 1460; WS = 7;SACK: NOP.  
La finalidad de esta conexión es que pcusuario pida un recurso a [www.tpfinal-tyr.com](http://www.tpfinal-tyr.com), con una solicitud HTTP GET a través del proxy y espera la respuesta.
- Conexión entre proxy y servidor www socket<10.10.0.1:38672–12.12.0.11:80>  
apertura: 30 a 38 cierre: 57-58-59-64-65-67-68-69-70  
**cliente:** W = 64240; ISN =1442436874; MSS = 1460; WS = 7;SACK: NOP.  
**server:** W = 65160; ISN =3516665645; MSS = 1460; WS = 7;SACK: NOP.  
Esta conexión es para que el proxy pida el recurso en el servidor web [www.tpfinal-tyr.com](http://www.tpfinal-tyr.com) con HTTP GET y esperar la respuesta del servidor.
- Conexión entre pcusuario y proxy sockets<10.10.0.10:45910–10.10.0.1:3128>  
apertura: 73-74-75 cierre: 120-124-125  
**cliente:** W = 64240; ISN =936995111; MSS = 1460; WS = 7;SACK: NOP.  
**server:** W = 65160; ISN =4040416614; MSS = 1460; WS = 7;SACK: NOP.  
En esta conexión pcusuario necesita el recurso index.html de datos.tpfinal-tyr.com y se lo pide al proxy con un HTTP GET.
- Conexión entre proxy y el servidor de datos  
sockets<10.10.0.1:51200–12.12.0.12:80>  
apertura: 84- de 87 a 94 cierre: de 110 a 115 y 121 a 123

**cliente:** W = 64240; ISN =2273985962; MSS = 1460; WS = 7; SACK:NOP.

**server:** W = 65160; ISN =1070407495; MSS = 1460; WS = 7;SACK:NOP.

En esta conexión el proxy le pide el recurso index.html al servidor web con un HTTP GET.

- conexión entre pcusuario y proxy sockets<10.10.0.10:45924–10.10.0.1:3128>

apertura: 126-127-129 cierre: 159-160-161

**cliente:** W = 64240; ISN =2319620458; MSS = 1460; WS = 7;SACK: NOP.

**server:** W = 65160; ISN =3144779218; MSS = 1460; WS = 7;SACK: NOP.

En esta conexión pcusuario pide el recurso [datos.pl](http://datos.pl) al proxy con un HTTP GET.

- Conexión entre proxy y el servidor de datos socket<10.10.0.1:51200–12.12.0.12:80>  
apertura: 131 a 139 cierre: 152-153-154-156

**cliente:** W = 64240; ISN =3820717764; MSS = 1460; WS = 7;SACK: NOP.

**server:** W = 65160; ISN =18576225592; MSS = 1460; WS = 7; SACK:NOP.

En esta conexión el proxy le pide el recurso [datos.pl](http://datos.pl) al servidor web con un HTTP GET.

#### 4. Para la conexión TCP utilizada para recuperar el archivo .html entre usuario y proxy indique la finalidad de cada PDU intercambiada a nivel de transporte y aplicación.

73: pcusuario empieza una conexión tcp mandando un SYN al servidor proxy

74: servidor proxy contesta con un SYN y ACK confirmando el anterior SYN

75: pcusuario confirma el SYN del proxy iniciando una nueva conexión (3-way-handshake)

76: petición HTTP GET de pcusuario para recuperar el recurso index.html al dominio datos.tpfinal-tyr.com para que resuelva el proxy

77: ACK del proxy hacia pcusuario para confirmar la petición fue recibida

116: el servidor proxy recuperó la página web y envía la cabecera de esta misma con un PSH así la capa de aplicación lo recibe directamente

117:pcusuario envía un ACK confirmando lo mandado por el servidor proxy

118: servidor proxy envía la respuesta de la petición GET con el contenido de index.html

119: pcusuario envía un ACK confirmando la pdu anterior

120: el servidor proxy confirma el ACK anterior y manda la bandera de FIN para empezar a terminar la conexión

124: pcusuario manda el FIN también y un ACK confirmando el fin del lado del proxy.

125: servidor proxy manda el último ACK confirmando el cierre de los dos lados de la conexión TCP

#### 6. Compare los headers HTTP del requerimiento del cliente al proxy con respecto del realizado por el proxy al servidor web. Muestre y explique las diferencias.

Petición de pcusuario al servidor proxy

```
> GET http://datos.tpfinal-tyr.com/index.html HTTP/1.0\r\n
User-Agent: w3m/0.5.3+git20230121\r\n
Accept: text/html, text/*;q=0.5, image/*, application/*\r\n
Accept-Encoding: gzip, compress, bzip, bzip2, deflate\r\n
Accept-Language: en;q=1.0\r\n
Host: datos.tpfinal-tyr.com\r\n
Referer: http://www.tpfinal-tyr.com/\r\n
\r\n
```

## Petición del servidor proxy al servidor web

```
> GET /index.html HTTP/1.1\r\n
User-Agent: w3m/0.5.3+git20230121\r\n
Accept: text/html, text/*;q=0.5, image/*, application/*\r\n
Accept-Encoding: gzip, compress, bzip, bzip2, deflate\r\n
Accept-Language: en;q=1.0\r\n
Referer: http://www.tpfinal-tyr.com/\r\n
Host: datos.tpfinal-tyr.com\r\n
Via: 1.0 proxy.tpfinal-tyr.com (squid/5.7)\r\n
X-Forwarded-For: 10.10.0.10\r\n
Cache-Control: max-age=259200\r\n
Connection: keep-alive\r\n
\r\n
```

La diferencia principal es en la petición del proxy al servidor web, solo se pide el recurso de la forma "GET /recurso". Esto se realiza porque el proxy puede estar conectado a varios servidores y necesita saber a que servidor específico hizo la solicitud el cliente.

en la petición GET del cliente al proxy se especifica el nombre del servidor en el cual conseguir el recurso y tiene la forma: "GET nombre-servidor/recurso".

Además en la cabecera de la solicitud desde el proxy se agregan algunos campos: "X-Forwarded-For" tiene la dirección IP que originalmente hizo la solicitud, en este caso es 10.10.0.10 que es el pcusuario. "Via" tiene la dirección del proxy que está haciendo la consulta junto al servicio que está ejecutando, en este caso squid. "Referer" contiene la dirección web de la que se redirigió a la página actual, redirección desde el servidor [www.tpfinal-tyr.com](http://www.tpfinal-tyr.com).

También "Cache-Control" que tiene instrucciones para el caching temporal, y "Connection" que controla si la conexión se mantendrá abierta luego de que la transacción actual se termine.

## Ejercicio 7 y 5 adjunto en los archivos