



Instituto Politécnico Nacional

IPN

Escuela Superior de Cómputo

ESCOM

6CV2

Práctica 6

“Servicio UDP de la capa de transporte”

Academia de Redes Aplicaciones para comunicaciones  
en red

Alumna:

Navarrete Becerril Sharon Anette

Fecha de entrega: “22-Octubre-2024”

Profesor: Ojeda Santillan Rodrigo

## OBJETIVO

Servicio UDP de la capa de transporte La capa de transporte es la cuarta capa del modelo de referencia OSI (Interconexión de Sistemas Abiertos). Su principal función es proporcionar una transferencia de datos confiable y eficiente entre dos dispositivos en una red. Esta capa se encarga de segmentar los datos recibidos de la capa de sesión en porciones manejables, y reensamblar estos datos en el destino. La capa de transporte también ofrece servicios de control de errores, control de flujo y, en algunos casos, control de congestión.

## INTRODUCCIÓN

UDP (User Datagram Protocol) es un protocolo de la capa de transporte que se utiliza para la transmisión de datos. Es uno de los principales protocolos de la capa de transporte en el modelo TCP/IP, junto con TCP (Transmission Control Protocol). UDP es un protocolo simple, sin conexión y no fiable, lo que significa que no establece una conexión antes de enviar datos, no garantiza la entrega de los datos, y no asegura el orden de los paquetes. UDP proporciona un mecanismo para detectar datos corruptos en paquetes, pero no intenta resolver otros problemas que surgen con paquetes, como cuando se pierden o llegan fuera de orden. Por eso, a veces UDP es conocido como el protocolo de datos no confiable. UDP es simple pero rápido, al menos en comparación con otros protocolos que funcionan sobre IP. A menudo se usa para aplicaciones sensibles al tiempo (como "streaming" de vídeo en tiempo real) donde la velocidad es más importante que la precisión.

## DESARROLLO

Para esta práctica se desarrollará un cliente y un servidor al que se le pasará un mensaje estático a través del protocolo UDP y que el servidor lo pueda escuchar y recibir. Inicialmente se debe de instalar Node JS en tu ordenador de la siguiente manera: Comenzaremos por instalar las dependencias necesarias para que se pueda crear el proyecto, para ello se puede utilizar un entorno de desarrollo libre como es el caso de Visual Studio Code o el que sea de la preferencia del lector. Comenzaremos por descargar node.js

### Código del servidor:

```
const dgram = require('dgram');
const PORT = 8080;
const HOST = '127.0.0.1';

const servidor = dgram.createSocket('udp4');

servidor.on('listening', () => {
  const address = servidor.address();
  console.log(`Servidor UDP escuchando en ${address.address}:${address.port}`);
});

servidor.on('message', (mensaje, rinfo) => {
```

```

console.log(`Mensaje recibido: ${mensaje} de ${rinfo.address}:${rinfo.port}`);
// Responder al cliente (opcional)
const respuesta = Buffer.from('Mensaje recibido correctamente');
servidor.send(respuesta, rinfo.port, rinfo.address, (err) => {
  if (err) {
    console.error('Error al enviar la respuesta:', err);
  } else {
    console.log('Respuesta enviada al cliente');
  }
});
});

// Manejador de errores
servidor.on('error', (err) => {
  console.error(`Error en el servidor: ${err.stack}`);
  servidor.close();
});

servidor.bind(PORT, HOST);

```

C:\Windows\System32\cmd.e x + ^

```

C:\Users\israe\Desktop\ADR\PracticasRedes2\Practica6>node P_udp_servidor.js
Servidor UDP escuchando en 127.0.0.1:8080
Mensaje recibido: Mando paquete UDP de 127.0.0.1:60205
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63330
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63331
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63332
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63333
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63334
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63335
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63336
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:63337
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:51083
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:51084
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:51085
Respuesta enviada al cliente
Mensaje recibido: Mando paquete UDP de 127.0.0.1:51086
Respuesta enviada al cliente
^C
C:\Users\israe\Desktop\ADR\PracticasRedes2\Practica6>

```

## Cliente

Por otro lado, el cliente también utiliza el módulo `net` junto con la biblioteca `readline-sync` para permitir la entrada de datos desde la línea de comandos, y posteriormente envía estos datos al servidor. Ambos scripts permiten observar en tiempo real el intercambio de datos entre el cliente y el servidor. Para obtener una visión más detallada de los paquetes de datos transmitidos

### Código del cliente:

```
const dgram = require('dgram');
const PORT = 8080;
const HOST = '127.0.0.1';

setInterval(() => {
    const cliente = dgram.createSocket('udp4');

    // Enviar mensaje al servidor
    const mensaje = Buffer.from('Mando paquete UDP');
    cliente.send(mensaje, PORT, HOST, (err) => {
        if (err) {
            console.error('Error al enviar el mensaje:', err);
        } else {
            console.log('Mensaje UDP enviado');
        }
    });

    // Escuchar respuesta del servidor
    cliente.on('message', (respuesta, rinfo) => {
        console.log(`Respuesta del servidor: ${respuesta} de
${rinfo.address}:${rinfo.port}`);
        cliente.close();
    });

    // Manejador de errores en el cliente
    cliente.on('error', (err) => {
        console.error('Error en el cliente:', err);
        cliente.close();
    });
}, 1000); // Envía un mensaje cada segundo
```

## Análisis de tráfico con Wireshark

Se utilizó la herramienta Wireshark, que permite analizar los paquetes TCP y evaluar la calidad de la transmisión.

Capturando desde Adapter for loopback traffic capture						
Archivo	Edición	Visualización	Jr Captura	Analizar	Estadísticas	Teléfono
No.	Time	Source	Destination	Protocol	Length	Info
37	12.878387	127.0.0.1	127.0.0.1	UDP	49	63337 + 8088 Len=17
38	12.879958	127.0.0.1	127.0.0.1	UDP	62	88880 + 63337 Len=30
39	13.889821	127.0.0.1	127.0.0.1	UDP	49	51083 + 8088 Len=17
40	13.890001	127.0.0.1	127.0.0.1	UDP	62	88880 + 51083 Len=30
41	14.901195	127.0.0.1	127.0.0.1	UDP	49	51084 + 8088 Len=17
42	14.901886	127.0.0.1	127.0.0.1	UDP	62	88880 + 51084 Len=30
43	15.903162	127.0.0.1	127.0.0.1	UDP	49	51085 + 8088 Len=17
44	15.903929	127.0.0.1	127.0.0.1	UDP	62	88880 + 51085 Len=30
45	15.904136	127.0.0.1	127.0.0.1	UDP	49	51086 + 8088 Len=17
46	15.914264	127.0.0.1	127.0.0.1	UDP	62	88880 + 51086 Len=30
47	21.578449	127.0.0.1	127.0.0.1	TCP	50	49789 + 1042 [PSH, ACK] Seq=13 Ack=5 Win=8410 Len=6
49	21.578491	127.0.0.1	127.0.0.1	TCP	46	1042 + 49789 [ACK] Seq=5 Ack=19 Win=8347 Len=6
49	21.578821	127.0.0.1	127.0.0.1	TCP	46	1042 + 49789 [PSH, ACK] Seq=5 Ack=19 Win=8347 Len=2
50	21.578841	127.0.0.1	127.0.0.1	TCP	44	49789 + 1042 [ACK] Seq=6 Ack=20 Win=8341 Len=6
51	25.012736	127.0.0.1	127.0.0.1	TCP	53	51111 + 49769 [PSH, ACK] Seq=10 Ack=4 Win=8441 Len=9
52	25.012781	127.0.0.1	127.0.0.1	TCP	49	49769 + 56511 [ACK] Seq=4 Ack=19 Win=8438 Len=9
53	25.014130	127.0.0.1	127.0.0.1	TCP	47	49769 + 56511 [PSH, ACK] Seq=4 Ack=19 Win=8438 Len=3
54	25.041456	127.0.0.1	127.0.0.1	TCP	54	56511 + 49769 [ACK] Seq=19 Ack=7 Win=8441 Len=6
55	31.578341	127.0.0.1	127.0.0.1	TCP	50	49789 + 1042 [ACK] Seq=6 Ack=20 Win=8341 Len=6
55	31.578354	127.0.0.1	127.0.0.1	TCP	44	1042 + 49789 [PSH, ACK] Seq=7 Ack=11 Win=8347 Len=6
57	31.588353	127.0.0.1	127.0.0.1	TCP	46	1042 + 49789 [PSH, ACK] Seq=7 Ack=25 Win=8347 Len=2
59	31.588394	127.0.0.1	127.0.0.1	TCP	44	49789 + 1042 [ACK] Seq=25 Ack=9 Win=8410 Len=6
59	34.423688	127.0.0.1	127.0.0.1	TCP	52	17532 + 49742 [PSH, ACK] Seq=105 Ack=57 Win=8306 Len=8
60	34.423727	127.0.0.1	127.0.0.1	TCP	44	49742 + 17532 [ACK] Seq=105 Ack=57 Win=8306 Len=8
61	34.423730	127.0.0.1	127.0.0.1	TCP	47	17532 + 49742 [PSH, ACK] Seq=112 Ack=57 Win=8306 Len=96
62	42.423753	127.0.0.1	127.0.0.1	TCP	44	49742 + 17532 [ACK] Seq=57 Ack=209 Win=8405 Len=9
63	42.424899	127.0.0.1	127.0.0.1	TCP	52	49742 + 17532 [PSH, ACK] Seq=57 Ack=209 Win=8405 Len=8
64	34.428252	127.0.0.1	127.0.0.1	TCP	44	17532 + 49742 [ACK] Seq=289 Ack=65 Win=8306 Len=8
65	34.428591	127.0.0.1	127.0.0.1	TCP	52	49742 + 17532 [PSH, ACK] Seq=65 Ack=209 Win=8306 Len=48
66	34.428596	127.0.0.1	127.0.0.1	TCP	44	17532 + 49742 [ACK] Seq=65 Ack=11 Win=8306 Len=6
67	34.428668	192.168.5.61	127.0.0.1	NDNS	71	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
68	34.428734	172.108.78.89	224.0.0.251	NDNS	71	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
69	34.431708	192.168.5.61	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
70	34.431943	172.108.78.89	224.0.0.251	NDNS	109	Standard query 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
71	34.432001	192.168.5.61	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
72	34.432669	fe80::9d12:3004a1:2c0:ff:fb	127.0.0.1	NDNS	91	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
73	34.434034	fe80::9d12:3004a1:2c0:ff:fb	127.0.0.1	NDNS	129	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
74	34.434196	fe80::9d12:3004a1:2c0:ff:fb	127.0.0.1	NDNS	129	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
75	34.434895	172.108.78.89	224.0.0.252	LLNMR	85	Standard query 0x774c ANY EdwardsSantiago
76	34.434901	172.108.78.89	224.0.0.252	LLNMR	85	Standard query 0x774c ANY EdwardsSantiago
77	34.434991	192.168.5.61	224.0.0.251	NDNS	71	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
78	34.442251	172.108.78.89	224.0.0.251	NDNS	71	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
79	34.444747	fe80::73d:8861%e18:1554	224.0.0.251	NDNS	91	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
80	34.444938	fe80::73d:8861%e18:1554	224.0.0.251	NDNS	91	Standard query 0x0000 ANY EdwardsSantiago.local, "QDN" question
81	34.445139	192.168.5.61	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
82	34.445155	fe80::73d:8861%e18:1554	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
83	34.445157	fe80::73d:8861%e18:1554	224.0.0.251	NDNS	85	Standard query? ANY EdwardsSantiago
84	34.452975	192.168.5.61	224.0..252	LLNMR	65	Standard query 0x84ea ANY EdwardsSantiago
85	34.454949	fe80::73d:8861%e18:1554	224.0.0.252	NDNS	129	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
86	34.462795	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
87	34.462802	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
88	34.462811	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
89	34.462815	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
90	34.462819	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
91	34.462821	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
92	34.462825	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
93	34.462829	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
94	34.462833	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
95	34.462837	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
96	34.462841	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
97	34.462845	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
98	34.462849	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
99	34.462853	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
100	34.462857	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
101	34.462861	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
102	34.462865	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
103	34.462869	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
104	34.462873	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
105	34.462877	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
106	34.462881	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
107	34.462885	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
108	34.462889	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
109	34.462893	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
110	34.462897	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
111	34.462901	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
112	34.462905	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
113	34.462909	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
114	34.462913	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
115	34.462917	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
116	34.462921	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
117	34.462925	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
118	34.462929	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
119	34.462933	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
120	34.462937	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
121	34.462941	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
122	34.462945	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
123	34.462949	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
124	34.462953	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
125	34.462957	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
126	34.462961	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
127	34.462965	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
128	34.462969	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
129	34.462973	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
130	34.462977	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
131	34.462981	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
132	34.462985	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
133	34.462989	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
134	34.462993	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
135	34.462997	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
136	34.463001	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
137	34.463005	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
138	34.463009	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861%e18:1554 A 192.168.56.1
139	34.463013	172.108.78.89	224.0.0.251	NDNS	109	Standard query response 0x0000 AAAA fe80::73d:8861

## CONCLUSIONES

Colin Garcia Beatriz Adriana:

**Implementación práctica del modelo cliente-servidor:** La práctica descrita en el documento enseña a configurar un sistema cliente-servidor utilizando UDP en Node.js. A través de la creación de scripts simples para enviar y recibir mensajes, se demuestra la facilidad con la que se puede implementar comunicación en red utilizando este protocolo, haciendo énfasis en su capacidad para manejar eventos y errores

Mendoza Munguia Braulio Adriel:

**Simplicidad y eficiencia del protocolo UDP:** El protocolo UDP es una opción ideal para aplicaciones donde la rapidez es prioritaria sobre la confiabilidad, como en transmisiones en tiempo real. Su diseño sin conexión y sin mecanismos de corrección de errores lo hace más rápido que otros protocolos como TCP, aunque a costa de no garantizar la entrega o el orden de los paquetes

Navarrete Becerril Sharon Anette:

**Uso de herramientas de monitoreo como Wireshark:** La práctica nos enseña el uso de herramientas como Wireshark para observar y analizar el tráfico de red generado por las comunicaciones UDP. Esto no solo ayuda a entender cómo funcionan los paquetes a nivel técnico, sino que también fomenta buenas prácticas en el análisis y depuración de la red

## Bibliografía

Stevens, W. Richard. Programación en red con Unix: La API de Sockets. Addison-Wesley Professional, 2003.

Forouzan, Behrouz A. Comunicaciones de Datos y Redes. McGraw-Hill, 2012. Tanenbaum, Andrew S., y David J. Wetherall. Redes de Computadoras. Pearson, 2010.

Comer, Douglas E. Internetworking con TCP/IP Volumen Uno. Prentice Hall, 2006. Kurose, James F., y Keith W. Ross. Redes de Computadoras: Un Enfoque Descendente. Pearson, 2017.

Beazley, David, y Brian K. Jones. Python Cookbook: Recetas para Dominar Python 3. O'Reilly Media, 2013. Begg, A. Sockets TCP/IP en C: Guía Práctica para Programadores. Morgan Kaufmann, 2000