Chapter 1: Introducing Networks and Protocols

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2. Data Link
1. Physical
Routed Connection

Router 1

Router 2

Router 3

Router 4

Router 5

---

**Windows PowerShell**

```
PS C:\> tracert example.com
Tracing route to example.com [93.184.216.34] over a maximum of 30 hops:
1  <1 ms  <1 ms  <1 ms  192.168.50.1
2  *       *       * Request timed out.
3  *       *       * Request timed out.
4  2 ms  2 ms  1 ms my.jetpack [192.168.1.1]
5  119 ms 47 ms 41 ms 172.26.36.169
6  56 ms 39 ms 38 ms 107.79.227.124
7  *       *       * Request timed out.
8  58 ms 79 ms 70 ms 12.83.186.145
9  51 ms 40 ms 41 ms eger1401igenp.ip.att.net [12.122.133.33]
10 76 ms 38 ms 39 ms dcr2-so-4-0-0.atlanta.savvis.net [192.205.32.118]
11 116 ms 198 ms 47 ms 192.229.225.133
12 76 ms 40 ms 37 ms 93.184.216.34
Trace complete.
PS C:\>
```
Windows PowerShell

PS C:\Users\honp> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix . : localdomain
  Link-Local IPv6 Address . . . . . : fe80::cd70:e700:5486:fb1a%5
  IPv4 Address . . . . . . . . . . . : 192.168.182.133
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . : 192.168.182.2

Tunnel adapter isatap.localdomain:
  Media State . . . . . . . . . . . : Media disconnected
  Connection-specific DNS Suffix . : localdomain

Tunnel adapter Local Area Connection# 2:
  Connection-specific DNS Suffix . : 2001:0:9d38:6ab8:8ba:2c5a:5950:c03c
  IPv6 Address . . . . . . . . . . . : 2001:0:9d38:6ab8:8ba:2c5a:5950:c03c%2
  Link-Local IPv6 Address . . . . . : fe80::8ba:2c5a:5950:c03c%2
  Default Gateway . . . . . . . . . : ::

PS C:\Users\honp>
bob — bash — 80×20

Last login: Mon Sep 17 19:22:53 on ttys000
m1:~ honp$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=3<RXCSUM,TXCSUM>
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=b<RXCSUM,TXCSUM,VLAN_HWTAGGING>
    ether 00:0c:29:59:17:6f
    inet6 fe80::20c:29ff:fe59:176f%en0 prefixlen 64 scopeid 0x4
    inet 192.168.182.128 netmask 0xffffff00 broadcast 192.168.182.255
    media: autoselect (1000baseT <full-duplex>)
    status: active
m1:~ honp$

honp@ubby18:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qdisc qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qdisc qlen 1000
    link/ether 00:0c:29:74:ba:ce brd ff:ff:ff:ff:ff:ff
    inet 192.168.182.145/24 brd 192.168.182.255 scope global dynamic noprefixroute ens33
        valid_lft 1511sec preferred_lft 1511sec
    inet6 fe80::ff60:954e:211:7ff0/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
honp@ubby18:~$ http://api.ipify.org/

107.77.85.86
Chapter 2: Getting to Grips with Socket APIs
m1:Desktop honp$ gcc time_server.c -o time_server
m1:Desktop honp$ ./time_server
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connection...
Client is connected... 127.0.0.1
Reading request...
Received 320 bytes.
Sending response...
Sent 79 of 79 bytes.
Sent 25 of 25 bytes.
Closing connection...
Closing listening socket...
Finished.
m1:Desktop honp$

Local time is: Fri Oct 19 08:42:05 2018
m1:Desktop honp$ gcc time_server.c -o time_server
m1:Desktop honp$ ./time_server
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connection...
Client is connected... ::1
Reading request...
Received 316 bytes.
Sending response...
Sent 79 of 79 bytes.
Sent 25 of 25 bytes.
Closing connection...
Closing listening socket...
Finished.
m1:Desktop honp$ 

Local time is: Fri Oct 19 09:29:29 2018
m1:Desktop honp$ gcc time_server_dual.c -o time_server_dual
m1:Desktop honp$ ./time_server_dual
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connection...
Client is connected... ::ffff:127.0.0.1
Reading request...
Received 320 bytes.
Sending response...
Sent 79 of 79 bytes.
Sent 25 of 25 bytes.
Closing connection...
Closing listening socket...
Finished.
m1:Desktop honp$ 

http://127.0.0.1:8080/

Local time is: Fri Oct 19 16:19:39 2018
Chapter 3: An In-Depth Overview of TCP Connections

```
getaddrinfo()
    ↓
socket()
    ↓
connect()

Has stdin input?
    Yes → fgets(stdin)
    No  → send()

recv()

Has socket input?
    Yes  →
    No   →
Socket closed by peer?
    Yes  → close()
    No   →

printf()
```
root@ubby16:/home/lv/chap03# ./tcp_client example.com http
Configuring remote address...
Remote address is: 93.184.216.34 http
Creating socket...
Connecting...
Connected.
To send data, enter text followed by enter.
GET / HTTP/1.1
Sending: GET / HTTP/1.1
Sent 15 bytes.
Host: example.com
Sending: Host: example.com
Sent 18 bytes.
Sending:
Sent 1 bytes.
Received (1592 bytes): HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Tue, 30 Oct 2018 19:59:46 GMT
Etag: "1541025663+ident"
Expires: Tue, 06 Nov 2018 19:59:46 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (ord/4CD5)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270

<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8"/>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
    }  
  </style>

</head>
<body>

</body>
</html>
root@ubby16:~/lv/chap03

root@ubby16:~/lv/chap03# ./tcp_server_toupper
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connections...
New connection from 127.0.0.1

lv@ubby16:~/.chap03

lv@ubby16:~/.chap03$ ./tcp_client 127.0.0.1 8080
Configuring remote address...
Remote address is: 127.0.0.1 http-alt
Creating socket...
Connecting...
Connected.
To send data, enter text followed by enter.
Hello World!
Sending: Hello World!
Sent 13 bytes.
Received (13 bytes): HELLO WORLD!
root@ubby16:/home/lv/chap03
  $ ./tcp_server_chat
  Configuring local address...
  Creating socket...
  Binding socket to local address...
  Listening...
  Waiting for connections...
  New connection from 127.0.0.1
  New connection from 127.0.0.1

lv@ubby16:~$ ./tcp_client 127.0.0.1 8800
  Configuring remote address...
  Remote address is: 127.0.0.1 http-alt
  Creating socket...
  Connecting...
  Connected.
  To send data, enter text followed by enter.
  Received (23 bytes): Hello from terminal 1.
  Sending: Hello from a different terminal!
  Sent 33 bytes.

lv@ubby16:~$ ./tcp_client 127.0.0.1 8800
  Configuring remote address...
  Remote address is: 127.0.0.1 http-alt
  Creating socket...
  Connecting...
  Connected.
  To send data, enter text followed by enter.
  Received (23 bytes): Hello from terminal 1.
  Received (33 bytes): Hello from a different terminal!
Chapter 4: Establishing UDP Connections
$ gcc udp_recvfrom.c -o udp_recvfrom.exe -lws2_32

$ udp_recvfrom.exe
Configuring local address...
Creating socket...
Binding socket to local address...
$ gcc udp_recvfrom.c -o udp_recvfrom.exe -lws2_32

$ udp_recvfrom.exe
Configuring local address...
Creating socket...
Binding socket to local address...
Received (11 bytes): Hello World
Remote address is: 127.0.0.1 55476
Finished.

$

$ gcc udp_sendto.c -o udp_sendto.exe -lws2_32

$ udp_sendto.exe
Configuring remote address...
Remote address is: 127.0.0.1 8080
Creating socket...
Sending: Hello World
Sent 11 bytes.
Finished.

$
$ gcc udp_server_toupper.c -o udp_server_toupper.exe -lws2_32

$ udp_server_toupper.exe
Configuring local address...
Creating socket...
Binding socket to local address...
Waiting for connections...

$ gcc udp_client.c -o udp_client.exe -lws2_32

$ udp_client.exe 127.0.0.1 8080
Configuring remote address...
Remote address is: 127.0.0.1 8080
Creating socket...
Connecting...
Connected.
To send data, enter text followed by enter.
Example string to convert.
Sending: Example string to convert.
Sent 27 bytes.
Received (27 bytes): EXAMPLE STRING TO CONVERT.
Chapter 5: Hostname Resolution and DNS

2. Client A receives the IP address xxx.xxx.xxx.xxx.
5. Root DNS server directs query to the .COM DNS server.
6. .COM DNS server queries the EXAMPLE.COM DNS server.
7. EXAMPLE.COM DNS server returns the IP address xxx.xxx.xxx.xxx.
8. DNS server returns the IP address to Client A.

Client B queries www.example.com.
2. DNS server returns the IP address xxx.xxx.xxx.xxx to Client B.

I remember from the last request.
```bash
$ gcc lookup.c -o lookup.exe -lws2_32

$ lookup example.com
 Resolving hostname 'example.com'
 Remote address is:
  93.184.216.34

$ 
```
## Header Format

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QR</td>
<td>Opcode</td>
<td>AA</td>
<td>TC</td>
<td>RD</td>
<td>RA</td>
</tr>
</tbody>
</table>

- ID
- QDCOUNT
- ANCOUNT
- NSCOUNT
- ARCOUNT

## Question Format

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- QTYPE
- QCLASS
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'w'</td>
<td></td>
<td></td>
<td>'w'</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'w'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'w'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'e'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>'x'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'a'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'m'</td>
<td></td>
<td></td>
<td></td>
<td>'p'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'l'</td>
<td></td>
<td></td>
<td>'e'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>'c'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>'o'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>NAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDLENGTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
m1:Desktop honp$ gcc dns_query.c -o dns_query
m1:Desktop honp$ ./dns_query example.com a
Configuring remote address...
Creating socket...
Sent 29 bytes.
ID = AB CD
QR = 0 query
OPCODE = 0 standard
AA = 0
TC = 0
RD = 1 recursion desired
QDCOUNT = 1
ANCOUNT = 0
NSCOUNT = 0
ARCOUNT = 0
Query 1
    name: example.com
    type: 1
    class: 1

Received 45 bytes.
ID = AB CD
QR = 1 response
OPCODE = 0 standard
AA = 0
TC = 0
RD = 1 recursion desired
RCODE = 0 success
QDCOUNT = 1
ANCOUNT = 1
NSCOUNT = 0
ARCOUNT = 0
Query 1
    name: example.com
    type: 1
    class: 1
Answer 1
    name: (pointer 12) example.com
    type: 1
    class: 1
    ttl: 20576
    rdlen: 4
    Address 93.184.216.34
m1:Desktop honp$ ./dns_query gmail.com mx
Configuring remote address...
Creating socket...
Sent 27 bytes.
ID = AB CD
QR = 0 query
OPCODE = 0 standard
AA = 0
TC = 0
RD = 1 recursion desired
QDCOUNT = 1
ANCOUNT = 0
NSCOUNT = 0
ARCOUNT = 0
Query 1
   name: gmail.com
type: 15
class: 1

Received 150 bytes.
ID = AB CD
QR = 1 response
OPCODE = 0 standard
AA = 0
TC = 0
RD = 1 recursion desired
rcode = 0 success
QDCOUNT = 1
ANCOUNT = 5
NSCOUNT = 0
ARCOUNT = 0
Query 1
   name: gmail.com
type: 15
class: 1
Answer 1
   name: (pointer 12) gmail.com
type: 15
class: 1
ttl: 3599
rdlen: 32
pref: 30
MX: alt3.gmail-smtp-in.l.google. (pointer 18) com
Answer 2
   name: (pointer 12) gmail.com
type: 15
Chapter 6: Building a Simple Web Client

[Diagram showing the flow of a simple web client request and response]

HTTP Request
GET / HTTP/1.1
Host: example.com

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1255
...

HTTP Response

[Table showing protocol, hostname, port, path, and hash]

- Protocol: http
- Hostname: www.example.com
- Port: 80
- Path: /res/page1.php
- Query String: ?user=bob#account
m1:Desktop honp$ gcc web_get.c -o web_get
m1:Desktop honp$ ./web_get example.com
URL: example.com
hostname: example.com
port: 80
path: Configuring remote address...
Remote address is: 93.184.216.34 http
Creating socket...
Connecting...
Connected.

Sent Headers:
GET / HTTP/1.1
Host: example.com:80
Connection: close
User-Agent: honpwc web_get 1.0

Received Headers:
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Mon, 17 Dec 2018 03:17:29 GMT
Etag: "1541025663+ident"
Expires: Mon, 24 Dec 2018 03:17:29 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (ord/4CB8)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270
Connection: close

Received Body:
<!doctype html>
<html>
<head>
    <title>Example Domain</title>
Example Form

You can fill out this form to make a POST request.

Name: Alice
Comment: Well Done!

submit
Chapter 7: Building a Simple Web Server

HTTP Request

GET /document.htm HTTP/1.1
Host: example.com
...

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1255
...

HTTP Response

```
m1:chap07 honp$ file --mime-type public/index.html
public/index.html: text/html
m1:chap07 honp$ file --mime-type public/smile.png
public/smile.png: image/png
m1:chap07 honp$ file --mime-type public/test.txt
public/test.txt: text/plain
m1:chap07 honp$ ...
```

```
m1:chap07 honp$ gcc web_server.c -o web_server
m1:chap07 honp$ ./web_server
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
New connection from 127.0.0.1.
serve_resource 127.0.0.1 /index.html
New connection from 127.0.0.1.
serve_resource 127.0.0.1 /smile.png
```

[38]
Example Page

This page is only an example.

Click here to see another page, or click here to download a text file.
Chapter 8: Making Your Program Send Email
Alice sends an email to Bob using SMTP. The email is then delivered to Gmail Server. Gmail Server then forwards the email to Yahoo Server via HTTP.
m1:Desktop honp$ dig mx gmail.com

; <<< DiG 9.7.3 <<< mx gmail.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18189
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
gmail.com.

;; ANSWER SECTION:
; IN  MX
gmail.com. 5 IN  MX 5 5 gmail-smtp-in.l.google.com.
gmail.com. 5 IN  MX 10 alt1.gmail-smtp-in.l.google.com.
gmail.com. 5 IN  MX 20 alt2.gmail-smtp-in.l.google.com.
gmail.com. 5 IN  MX 30 alt3.gmail-smtp-in.l.google.com.
gmail.com. 5 IN  MX 40 alt4.gmail-smtp-in.l.google.com.

;; Query time: 51 msec
;; SERVER: 192.168.182.2#53(192.168.182.2)
;; WHEN: Tue Jan 15 15:17:09 2019
;; MSG SIZE  rcvd: 150

m1:Desktop honp$ 

m1:Desktop honp$ dig mx yahoo.com

; <<< DiG 9.7.3 <<< mx yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30161
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
yahoo.com.

;; ANSWER SECTION:
; IN  MX
yahoo.com. 5 IN  MX 1 mta7.am0.yahoodns.net.
yahoo.com. 5 IN  MX 1 mta5.am0.yahoodns.net.
yahoo.com. 5 IN  MX 1 mta6.am0.yahoodns.net.

;; Query time: 63 msec
;; SERVER: 192.168.182.2#53(192.168.182.2)
;; WHEN: Thu Jan 17 20:08:12 2019
;; MSG SIZE  rcvd: 106

m1:Desktop honp$
m1:Desktop honp$ gcc smtp_send.c -o smtp_send
m1:Desktop honp$ ./smtp_send
mail server: gmail-smtp-in.l.google.com
Connecting to host: gmail-smtp-in.l.google.com:25
Configuring remote address...
Remote address is: 74.125.124.26 smtp
Creating socket...
Connecting...
Connected.
S: 220 mx.google.com ESMTP z5si775139ioi.6 - gsmt
C: HELO HONPWC
S: 250 mx.google.com at your service
from: alice@handsonnetworkprogramming.com
C: MAIL FROM:<alice@handsonnetworkprogramming.com>
S: 250 2.1.0 OK z5si775139ioi.6 - gsmt
to: [email_address]@gmail.com
C: RCPT TO:<[email_address]@gmail.com>
S: 250 2.1.5 OK z5si775139ioi.6 - gsmt
C: DATA
S: 354 Go ahead z5si775139ioi.6 - gsmt
subject: Test Email
C: From:<alice@handsonnetworkprogramming.com>
C: To:<[email_address]@gmail.com>
C: Subject:Test Email
C: Date:Sat, 19 Jan 2019 00:02:46 +0000
C:
Enter your email text, end with "." on a line by itself.
> Hi,
C: Hi,
> This email will be in the book!
C: This email will be in the book!
> .
C: .
S: 250 2.0.0 OK 1547856174 z5si775139ioi.6 - gsmt
C: QUIT
S: 221 2.0.0 closing connection z5si775139ioi.6 - gsmt
Closing socket...
Finished.
m1:Desktop honp$
Chapter 9: Loading Secure Web Pages with HTTPS and OpenSSL
**Diagram 1:**
- **Public Key:**
  - e244dbfa
  - 57c32c21
  - d38f9213

- **Private Key:**
  - 60ad6bc3
  - 2a655e14
  - 23c1c0be

**Diagram 2:**
- **Private Key:**
  - 60ad6bc3
  - 2a655e14
  - 23c1c0be

- **Public Key:**
  - e244dbfa
  - 57c32c21
  - d38f9213

**Diagram 3:**
- **Command Output 1:**
  - $ openssl version
  - OpenSSL 1.1.0g 2 Nov 2017

- **Command Output 2:**
  - $ gcc openssl_version.c -o openssl_version -lcrypto
  - $ ./openssl_version
  - OpenSSL version: OpenSSL 1.1.0g 2 Nov 2017
$ gcc https_simple.c -o https_simple -lssl -lcrypto -Wall
$ ./https_simple example.org 443
Configuring remote address...
Remote address is: 93.184.210.34 https
Creating socket...
Connecting...
Connected.

SSL/TLS using ECDHE-RSA-AES128-GCM-SHA256
subject: /C=US/ST=California/L=Los Angeles/O=Internet Corporation for Assigned Names and Numbers/OU=Technology/CN=www.example.org
issuer: /C=US/O=DigiCert Inc/CN=DigiCert SHA2 Secure Server CA
Sent Headers:
GET / HTTP/1.1
Host: example.org:443
Connection: close
User-Agent: https_simple

Received (341 bytes): 'HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html; charset=UTF-8
Date: Thu, 24 Jan 2019 23:50:51 GMT
Etag: "1541025663+ident"
Expires: Thu, 31 Jan 2019 23:50:51 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (ord/4CDAA)
Vary: Accept-Encoding
X-Cache: HIT
Content-Length: 1270
Connection: close

Received (1270 bytes): '<!doctype html>
<html>
<head>
    <title>Example Domain</title>
    ...
</head>
<body>
    ...
</body>
</html>

Connection closed by peer.
Closing socket...
Finished.
Chapter 10: Implementing a Secure Web Server

```
$ openssl req -x509 -newkey rsa:2048 -nodes -sha256 -keyout key.pem -out cert.pem -days 36500
Generating a 2048 bit RSA private key
............+++.
+++ Writing new private key to 'key.pem'
-----
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

```
$ openssl x509 -text -noout -in cert.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 
      d5:e8:c8:77:6a:18:c6:80
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = AU, ST = Some-State, O = Internet Widgets Pty Ltd
    Validity
      Not Before: Feb 7 21:10:06 2019 GMT
      Not After : Jan 14 21:10:06 2119 GMT
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        f0:c4:3e:94:ad:d7:3e:dc:7f:55:ce:b8:ed:85:f5:
        cc:19:18:1d:54:1c:d0:4c:b3:91:3d:44:de:fb:4d:
$ gcc tls_time_server.c -o tls_time_server -lssl -lcrypto
$ ./tls_time_server
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connection...
Client is connected... 127.0.0.1
SSL connection using ECDHE-RSA-AES128-GCM-SHA256
Reading request...
Received 0 bytes.
Sending response...
Sent 79 of 79 bytes.
Sent 25 of 25 bytes.
Closing connection...
Waiting for connection...
Client is connected... 127.0.0.1
SSL connection using ECDHE-RSA-AES128-GCM-SHA256
Reading request...
Received 0 bytes.
Sending response...
Sent 79 of 79 bytes.
Sent 25 of 25 bytes.
Closing connection...
Waiting for connection...
Your connection is not secure

The owner of 127.0.0.1 has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

Learn more...

Report errors like this to help Mozilla identify and block malicious sites

Go Back  Advanced

Local time is: Thu Feb  7 15:04:33 2019
**Example Page**

This page is only an example.

Click [here](#) to see another page, or click [here](#) to download a text file.
Chapter 11: Establishing SSH Connections with libssh

```
$ gcc ssh_version.c -o ssh_version -llssh
$ ./ssh_version
libssh version: 0.8.6/openssl/zlib
$
```

```
$ gcc ssh_connect.c -o ssh_connect -llssh
$ ./ssh_connect
localhost
[2019/02/15 19:44:44.037146, 2] ssh_connect: libssh 0.8.6 (c) 2003-2018 Aris Adamantiadis, Andre as Schneider and libssh contributors. Distributed under the LGPL, please refer to COPYING file for information about your rights, using threading threads_pthread
[2019/02/15 19:44:44.037397, 2] ssh_socket_connect: NonBlocking connection socket: 3
[2019/02/15 19:44:44.037414, 2] ssh_connect: Socket connecting, now waiting for the callbacks to work
[2019/02/15 19:44:44.037431, 1] socket_callback_connected: Socket connection callback: 1 (0)
[2019/02/15 19:44:44.275512, 1] ssh_client_connection_callback: SSH server banner: SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
[2019/02/15 19:44:44.275561, 1] ssh_analyze_banner: Analyzing banner: SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
[2019/02/15 19:44:44.275594, 1] ssh_analyze_banner: We are talking to an OpenSSH client version: 7.6 (76000)
[2019/02/15 19:44:44.342304, 1] ssh_known_hosts_read_entries: Failed to open the known_hosts file '/etc/ssh/ssh_known_hosts': No such file or directory
[2019/02/15 19:44:44.387545, 2] ssh_packet_dh_reply: Received SSH_KEXDH_REPLY
[2019/02/15 19:44:44.402617, 2] ssh_client_curve25519_reply: SSH_MSG_NEWKEYS sent
[2019/02/15 19:44:44.402662, 2] ssh_packet_keys: Received SSH_MSG_NEWKEYS
[2019/02/15 19:44:44.408837, 2] ssh_packet_keys: Signature verified and valid
Connected to localhost on port 22.
Banner:
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
$
```
$ gcc ssh_auth.c -o ssh_auth -lssh
$ ./ssh_auth localhost 22 alice
Connected to localhost on port 22.
Banner:
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
Host public key hash:
SHA1:UWXxDb3ArslAr7i2YBW07rLugtE
Checking ssh_session_is_known_server()
Host Known.
Password: password123
Authentication successful!
$

$ gcc ssh_command.c -o ssh_command -lssh
$ ./ssh_command localhost 22 alice
Connected to localhost on port 22.
Banner:
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
Host public key hash:
SHA1:UWXxDb3ArslAr7i2YBW07rLugtE
Checking ssh_session_is_known_server()
Host Known.
Password: password123
Authentication successful!
Remote command to execute: ls -l
total 24
-rw-rw-r-- 1 alice alice 312 Feb 15 07:52 credits.txt
-rw-r--r-- 1 alice alice 8980 Apr 16 2018 examples.desktop
-rw-r--r-- 1 alice alice 2124 Feb 15 07:51 schedule.txt
-rw-rw-r-- 1 alice alice 35 Feb 15 07:51 test.txt
ssh_channel_read() failed.
$
```
$ gcc ssh_download.c -o ssh_download -lssh
$ ./ssh_download localhost 22 alice
Connected to localhost on port 22.
Banner:
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.2
Host public key hash:
SHA1:UWXxDb3ArslAr7l2YBW07rLugtE
Checking ssh_session_is_known_server()
Host Known.
Password: password123
Authentication successful!
Remote file to download: test.txt
Downloading file test.txt (48 bytes, permissions 0664)
Received test.txt:
Hello World!
This file exists only for testing.
```

Chapter 12: Network Monitoring and Security

```
$ ping example.com
PING example.com (93.184.216.34) 56(84) bytes of data.
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=1 ttl=128 time=49.2 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=2 ttl=128 time=355 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=3 ttl=128 time=39.3 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=4 ttl=128 time=151 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=5 ttl=128 time=34.6 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=6 ttl=128 time=66.8 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=7 ttl=128 time=65.0 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=8 ttl=128 time=62.9 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=9 ttl=128 time=186 ms
64 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=10 ttl=128 time=134 ms
^C
--- example.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9018ms
rtt min/avg/max/mdev = 34.696/114.537/355.360/94.093 ms
$ 
```
$ ping -s 1000 example.com
PING example.com (93.184.216.34) 1000(1028) bytes of data.
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=1 ttl=128 time=39.4 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=2 ttl=128 time=82.0 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=3 ttl=128 time=64.4 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=4 ttl=128 time=270 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=5 ttl=128 time=253 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=6 ttl=128 time=53.1 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=7 ttl=128 time=48.9 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=8 ttl=128 time=48.5 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=9 ttl=128 time=50.3 ms
1008 bytes from 93.184.216.34 (93.184.216.34): icmp_seq=10 ttl=128 time=49.5 ms

--- example.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 39.400/96.117/270.959/83.954 ms
$  

$ traceroute -n example.com
traceroute to example.com (93.184.216.34), 30 hops max, 60 byte packets
 1 23.92.28.3 0.584 ms 0.683 ms 0.807 ms
 2 74.207.239.24 0.701 ms 74.207.239.20 0.687 ms 74.207.239.6 0.758 ms
 3 198.32.132.86 1.927 ms 74.207.239.8 0.733 ms 0.720 ms
 4 198.32.132.86 1.888 ms 152.195.80.131 3.866 ms 198.32.132.86 1.860 ms
 5 93.184.216.34 0.347 ms 0.345 ms 152.195.80.131 3.815 ms
 6 93.184.216.34 0.353 ms 0.379 ms 0.362 ms
$  

[ 58 ]
Source

Intermediate Routers

Destination

Echo Request, TTL=1
Time Exceeded

Echo Request, TTL=2
Time Exceeded

Echo Request, TTL=3
Echo Reply

$h = netstat -nt
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address
    Foreign Address     State
tcp  0        0 127.0.0.53:53  0.0.0.0:*  LISTEN  434/systemd-resolve
 tcp  0        0 0.0.0.0:22  0.0.0.0:*  LISTEN  72522/sshd
 tcp  0        0 0.0.0.0:80  0.0.0.0:*  LISTEN  92531/cupsd
 tcp  0        0 0.0.0.0:1:631  :::*  LISTEN  20762/apache2
 tcp  0        0 0.0.0.0:22  0.0.0.0:*  LISTEN  72522/sshd
 tcp  0        0 0.0.0.0:1:631  :::*  LISTEN  92531/cupsd
\$

$s = sudo netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address
    Foreign Address     State     PID/Program name
tcp  0        0 127.0.0.53:53  0.0.0.0:*  LISTEN  434/systemd-resolve
 tcp  0        0 0.0.0.0:22  0.0.0.0:*  LISTEN  72522/sshd
 tcp  0        0 0.0.0.0:80  0.0.0.0:*  LISTEN  92531/cupsd
 tcp  0        0 0.0.0.0:1:631  :::*  LISTEN  20762/apache2
 tcp  0        0 0.0.0.0:22  0.0.0.0:*  LISTEN  72522/sshd
 tcp  0        0 0.0.0.0:1:631  :::*  LISTEN  92531/cupsd
\$
C:\tshark -i 5 host 8.8.8.8
Capturing on 'Wire'   
  1 0.000000 192.168.50.119 -> 8.8.8.8  DNS 71 Standard query 0xabcd A example.com
  2 0.146698 8.8.8.8 -> 192.168.50.119 DNS 87 Standard query response 0xabcd A example.com A 93.184.216.34
2 packets captured
C:\>

C:\tshark -i 5 host 93.184.216.34 and port 80
Capturing on 'Wire'   
  1 0.000000 192.168.50.119 -> 93.184.216.34 TCP 66 5521 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  2 0.054634 93.184.216.34 -> 192.168.50.119 TCP 66 80 -> 5521 [SYN, ACK] Seq=0 Ack=1 Win=27800 Len=0 MSS=1370 SACK_PERM=1
  3 0.054765 192.168.50.119 -> 93.184.216.34 TCP 54 5521 -> 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
  4 0.054902 192.168.50.119 -> 93.184.216.34 HTTP 145 GET / HTTP/1.1
  5 0.095022 93.184.216.34 -> 192.168.50.119 TCP 60 00 -> 5521 [ACK] Seq=1 Ack=92 Win=27888 Len=0
  6 0.099856 93.184.216.34 -> 192.168.50.119 TCP 1424 HTTP/1.1 300 OK [TCP segment of a reassembled PDU]
  7 0.099957 93.184.216.34 -> 192.168.50.119 HTTP 300 HTTP/1.1 300 OK [text/html]
  8 0.698857 192.168.50.119 -> 93.184.216.34 TCP 60 00 -> 5521 [FIN, ACK] Seq=1617 Ack=92 Win=27888 Len=0
  9 0.098140 192.168.50.119 -> 93.184.216.34 TCP 54 5521 -> 80 [ACK] Seq=92 Ack=1618 Win=65536 Len=0
11 packets captured
C:\>
Chapter 13: Socket Programming Tips and Pitfalls
call connect()

Client

 Server

SYN Message

SYN + ACK Message

ACK Message

connect() returns
Client

DATA (Request)

ACK Message

DATA (Response)

Server
Client

wait for acknowledgment

resume sending

Server

delay ACK

data

ACK Message

DATA

ACK + DATA (Response)
close() is called

ESTABLISHED

FIN-WAIT-1

ACK Message

FIN-WAIT-2

FIN Message

delay

TIME-WAIT

CLOSED

Peer

ESTABLISHED

CLOSE-WAIT

LAST-ACK

CLOSED

FIN Message

ACK Message
```
honp@ubby18:~$ gcc server_noreuse.c -o server_noreuse
honp@ubby18:~$ ./server_noreuse
Configuring local address...
Creating socket...
Binding socket to local address...
Listening...
Waiting for connection...
Client is connected.
Waiting for connection...
^C
honp@ubby18:~$ ./server_noreuse
Configuring local address...
Creating socket...
Binding socket to local address...
bind() failed. (98)
honp@ubby18:~$ 
```

```
C:\>gcc setsize.c -o setsize.exe
C:\>setsize
FD_SETSIZE is 64.
C:\>
```
Chapter 14: Web Programming for the Internet of Things

Step 1: IoT device acts as Wi-Fi access point during configuration

Step 2: IoT device connects to network as client for Internet access

Smart Watch

Smartphone

Router

ZigBee Devices

ZigBee Coordinator/Hub