

Networking and HTTP

Martin Pola

Foundations of the Internet

Many smaller networks that are linked together: *inter-net*.

- Devices on the internet are called *hosts*
- All hosts have at least one IP address
- Multiple IP addresses necessary if a host has multiple *network interfaces* (e.g. both wired and wireless)

There are different *versions* of IP addresses.

- IP version 4 – 192.168.10.2
- IP version 6 – 2001:0db8:85a3:0000:0000:8a2e:0370:7334

IP version 6 (IPv6) was introduced to mitigate the issue of IP address exhaustion.

Find your IP address

- `ipconfig` on Windows
- `ifconfig` on Unix-like systems (abbreviation for *interface config*)

You can always reach your own computer (or host, more generally) via the *loopback interface*:

`127.0.0.1`

Hosts, routers and switches

It would be impossible for every host to have direct connections to all other hosts.

- Switches are used *inside* networks
- Routers are used *between* networks

Two-way communication between (a pair of) hosts on the internet.

- Client-server – client makes requests, server responds
- Peer-to-peer – hosts are both clients and servers

Services

Sockets provide a foundation for *services* on the Internet.

Through sockets, hosts exchange sequences of bytes. For hosts to understand each other, services use *protocols* – pre-defined, standardized languages.

Services

Sockets provide a foundation for *services* on the Internet.

Through sockets, hosts exchange sequences of bytes. For hosts to understand each other, services use *protocols* – pre-defined, standardized languages.

Most services use the client-server model.

- Hypertext Transfer Protocol (HTTP) – our *web*
- Internet Message Access Protocol (IMAP) – reading e-mail
- Simple Mail Transfer Protocol (SMTP) – sending e-mail
- Skype

Services

Sockets provide a foundation for *services* on the Internet.

Through sockets, hosts exchange sequences of bytes. For hosts to understand each other, services use *protocols* – pre-defined, standardized languages.

Most services use the client-server model.

- Hypertext Transfer Protocol (HTTP) – our *web*
- Internet Message Access Protocol (IMAP) – reading e-mail
- Simple Mail Transfer Protocol (SMTP) – sending e-mail
- Skype

There are also services using the peer-to-peer model, e.g. Bittorrent.

HTTP

- ❶ The client (your web browser) makes a request
- ❷ The server responds to the request

Requires a host to run a program that acts like a server. A host can run multiple server programs at the same time.

How does the host know which server program an incoming connection is trying to reach? Each server program *listens* on a separate port number.

HTTP runs on port 80, by default, but the port number can be specified in the URL.

- `http://example.net/` – default port
- `http://example.net:8080/` – port 8080

HTTP request

```
GET / HTTP/1.0  
Host: example.net
```

The slash (/) comes from the last part of the URL:
`http://example.net/`

We can use `telnet` to connect to a remote server:

```
$ telnet example.net 80
```

HTTP response

A response consists of *headers* and a *body*. The two sections are separated by two line breaks.

```
HTTP/1.1 200 OK
```

```
Content-Type: text/html; charset=UTF-8
```

```
Expires: Wed, 12 Feb 2020 09:01:17 GMT
```

```
Last-Modified: Thu, 17 Oct 2019 07:18:26 GMT
```

```
Server: ECS (nyb/1D04)
```

```
Content-Length: 1256
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
[...]
```