Connectivity 2

Internet refresher …

TCP/IP

IP

Private & public addresses

Routing, NAT and Firewalls

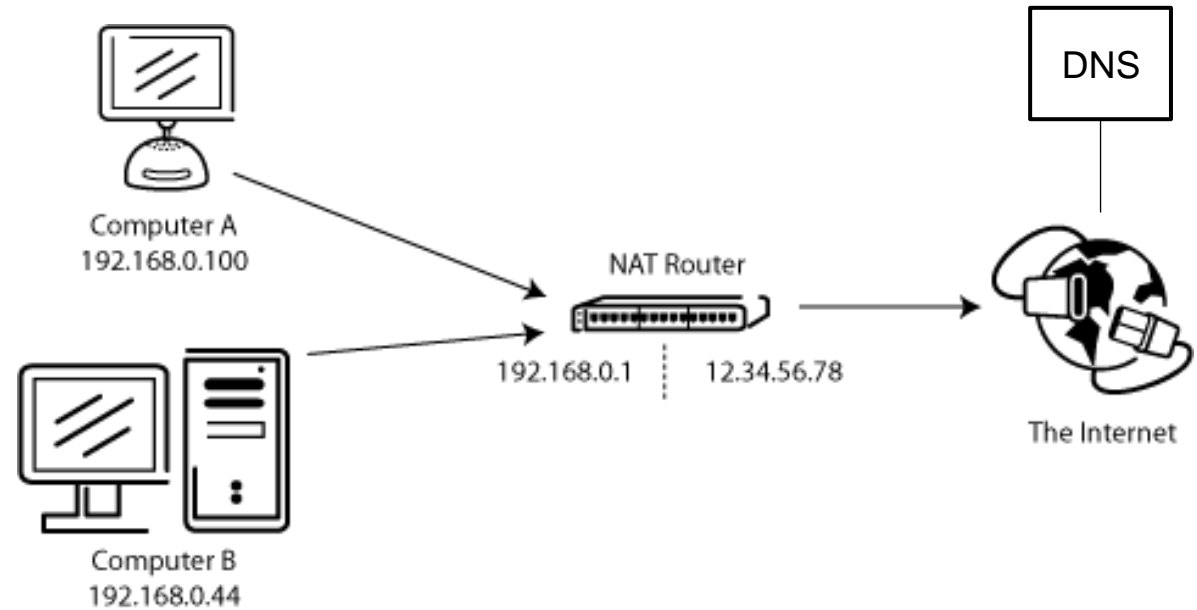Host names (DNS)

Load balancing

- DNS round robin
- Virtual IP

TCP & UDP

Ports (services)

Error control & ordering

Computer A
192.168.0.100

Computer B
192.168.0.44

NAT Router

192.168.0.1   12.34.56.78

DNS

The Internet

HTTP Request

Method

GET, POST, PUT, DELETE …

Headers

Host

Accept (content type, encoding ….)

Authorization

Cache-Control
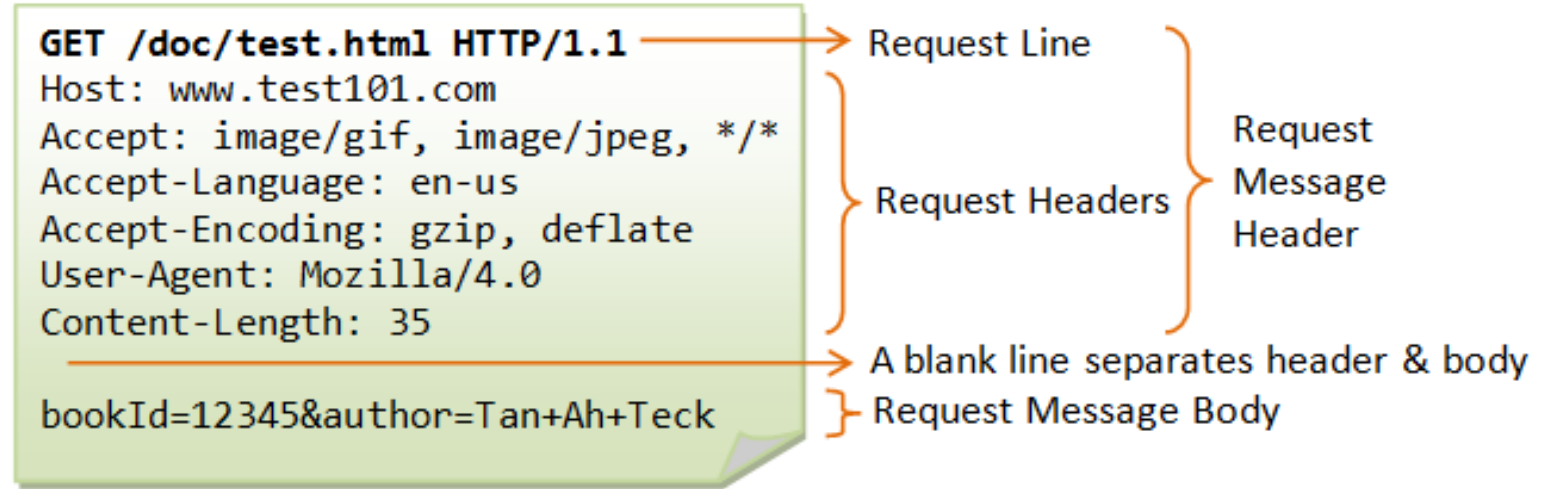
Cookies

Content-Type

Body

Application specific (e.g. JSON, XML …)

POST and PUT methods only

```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request Message Header

A blank line separates header & body

Request Message Body

HTTP Response

Status line

Protocol version

Status

Headers

Access-Control-Allow-Origin

Cache-Control

Content-Type

Set-Cookie

…

Body

Application specific (e.g. JSON …)

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Status Line

Response Headers

Response Message Header

A blank line separates header & body

Response Message Body
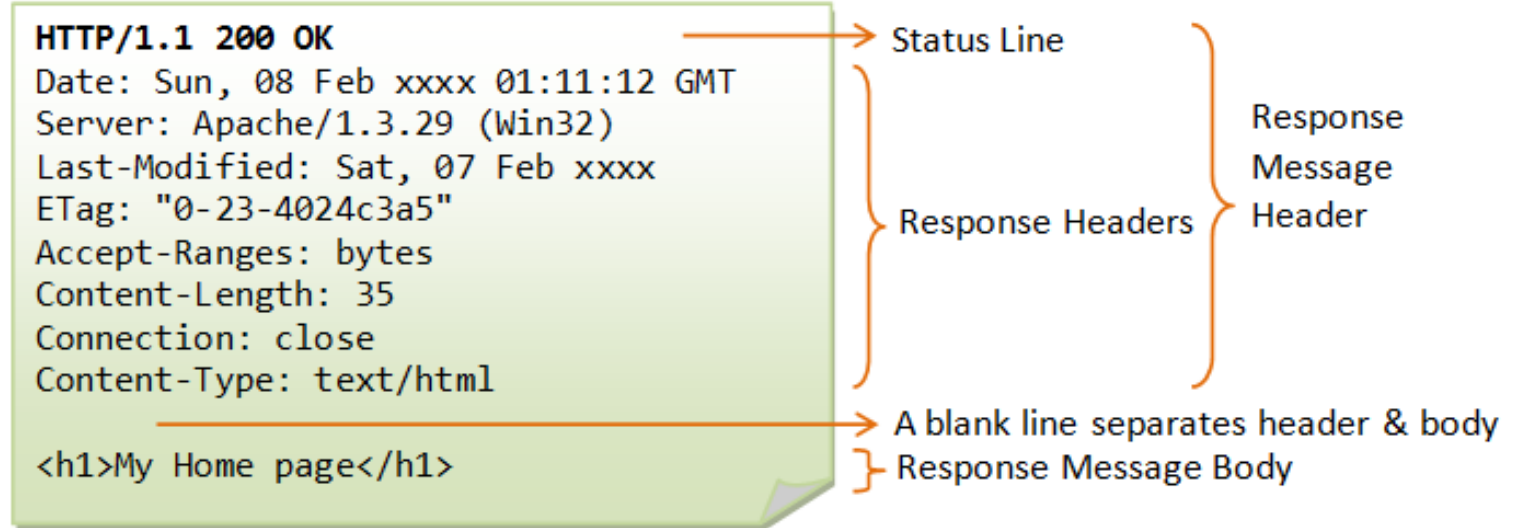
MQTT

MQTT (ISO/IEC PRF 20922)

Overview

TCP/IP based

- MQTT-SN (UDP)

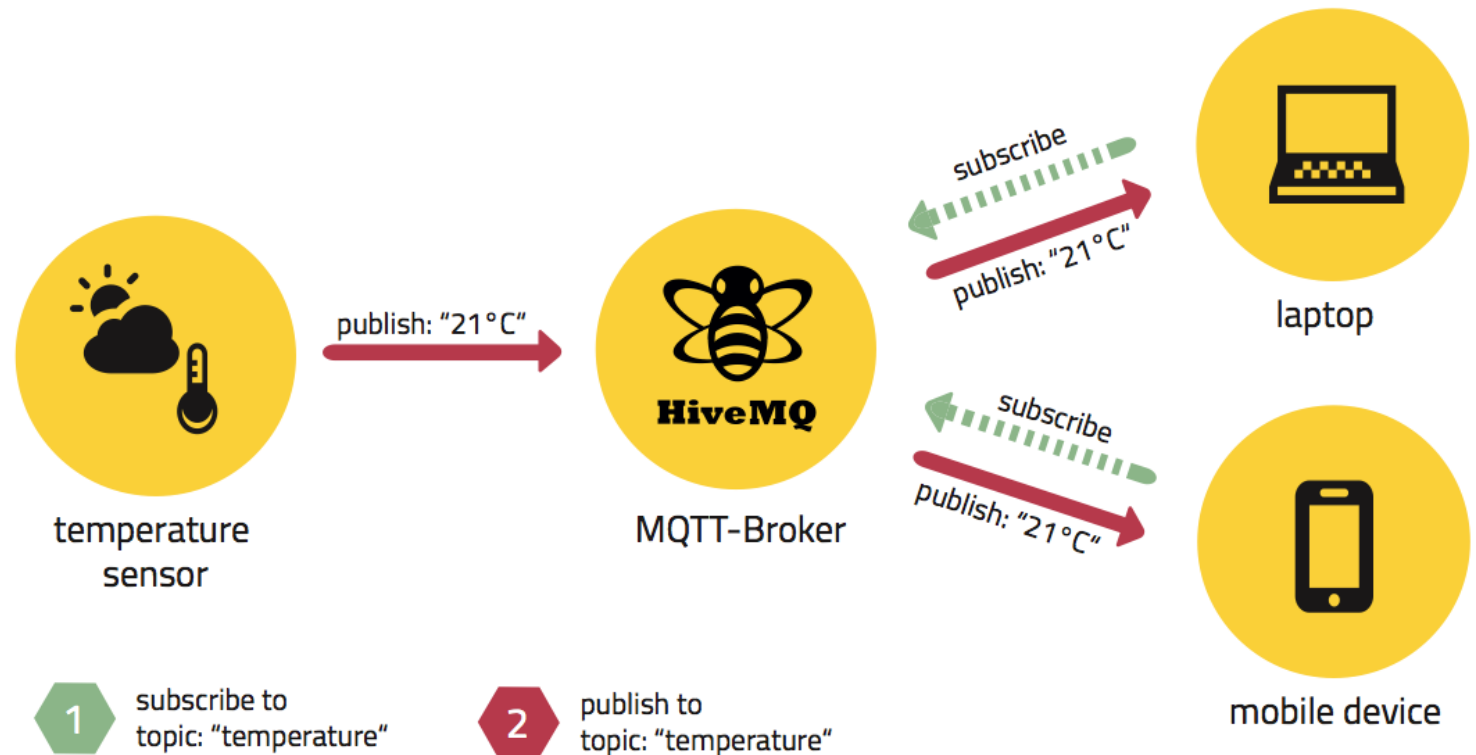**Small footprint / low bandwidth**

e.g. compared to HTTP

Pub/sub

Client connect to **broker**

And publish to **topics**

Or subscribe to **topics**

- Wildcard subscription
- Retained messages

MQTT implementations

Servers
Open source: Eclipse Mosquitto, HiveMQ …
Cloud: AWS, Azure, GCloud …

Clients
ESP32: pubsubclient, lwmqtt, Paho
Linux: Paho

…

MQTT QoS

QoS = Quality of service

Does not apply to TCP, only to the client-client connection

Three levels:
- 0: fire and forget (at most once)
- 1: resend until acknowledged (at least once)
- 2: exactly once delivery (exactly once)

WARNING

Cloud providers may deviate from the MQTT QoS specification

Practical tips

Which level to use with ESP32?
- Use 0 when you don't care too much about lost messages
- Use 1 when you can afford duplicate messages (e.g. deduplicating on the server side)

What to do when the server is not accessible (e.g. connection refused, no route to host)?
- Buffer locally (preferably in-memory)

MQTT security

Encryption

TLS on top of TCP/IP

Authentication

Username / password

Access key

Certificate / PK

MQTT 5

MQTT.ORG

MQTT 5 History

2010
MQTT 3.1
Royalty-free release

2014
MQTT 3.1.1
OASIS standard

2018
MQTT 5
Initial release

2019
HiveMQ MQTT Client
Public release

2019
HiveMQ
Open source edition

1999
MQTT
Invention

2013
HiveMQ 1.3
Initial public release

2016
MQTT 3.1.1
ISO standard

2018
HiveMQ 4
100% MQTT 5 compability

2019
MQTT 5
OASIS standard

https://www.hivemq.com/blog/mqtt5-essentials-part1-introduction-to-mqtt-5/

Exercises

WiFi modes

Station + HTTP client

ESP Access point + HTTP server -> fire an LED upon HTTP request

MQTT

ESP client to mosquito / thingsboard server