

Connectivity 2

Internet refresher ...

TCP/IP

IP

Private & public addresses

Routing, NAT and Firewalls

Host names (DNS)

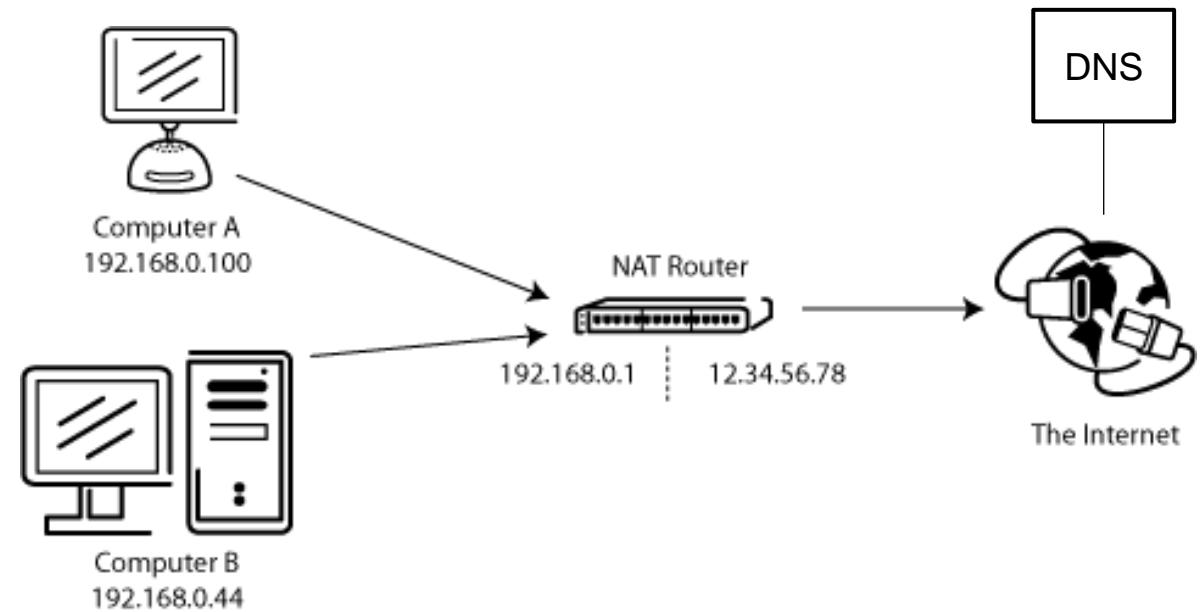
Load balancing

- DNS round robin
- Virtual IP

TCP & UDP

Ports (services)

Error control & ordering



WiFi

WiFi

Modes

Station (WIFI_STA)

Access Point (WIFI_AP)

Relay (WIFI_AP_STA)

Other terminology

SSID (Service Set Identifier)

RSSI (Received Signal Strength Indicator)

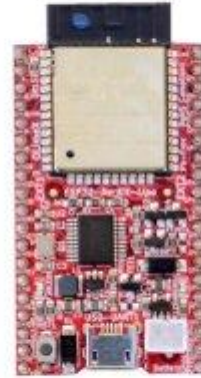
Authentication

WPA2-PSK, WPA2-ENT (RADIUS)

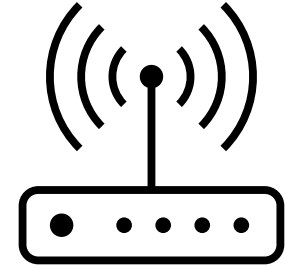
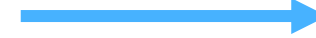
WPA3: PSK->SAE, longer keys ...

Outdated: WEP, WPA

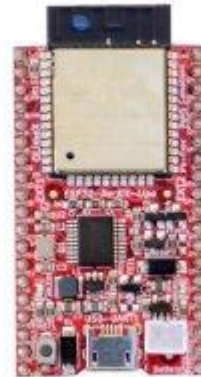
192.168.0.57



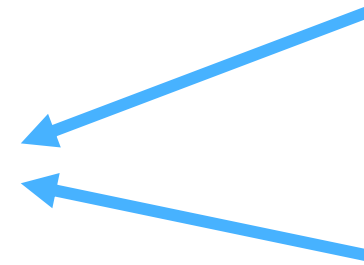
STA



192.168.0.1



AP



HTTP

HTTP Request

Method

GET, POST, PUT, DELETE ...

Headers

Host

Accept (content type, encoding)

Authorization

Cache-Control

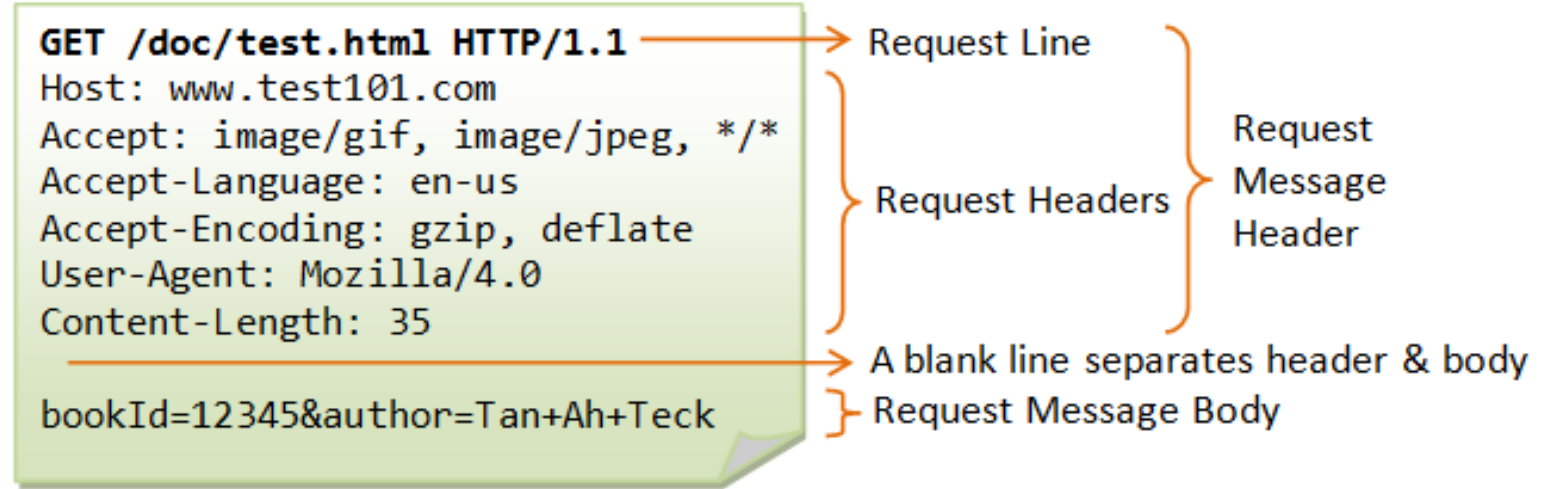
Cookies

Content-Type

Body

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

POST and PUT methods only



HTTP Response

Status line

Protocol version

Status

Headers

Access-Control-Allow-Origin

Cache-Control

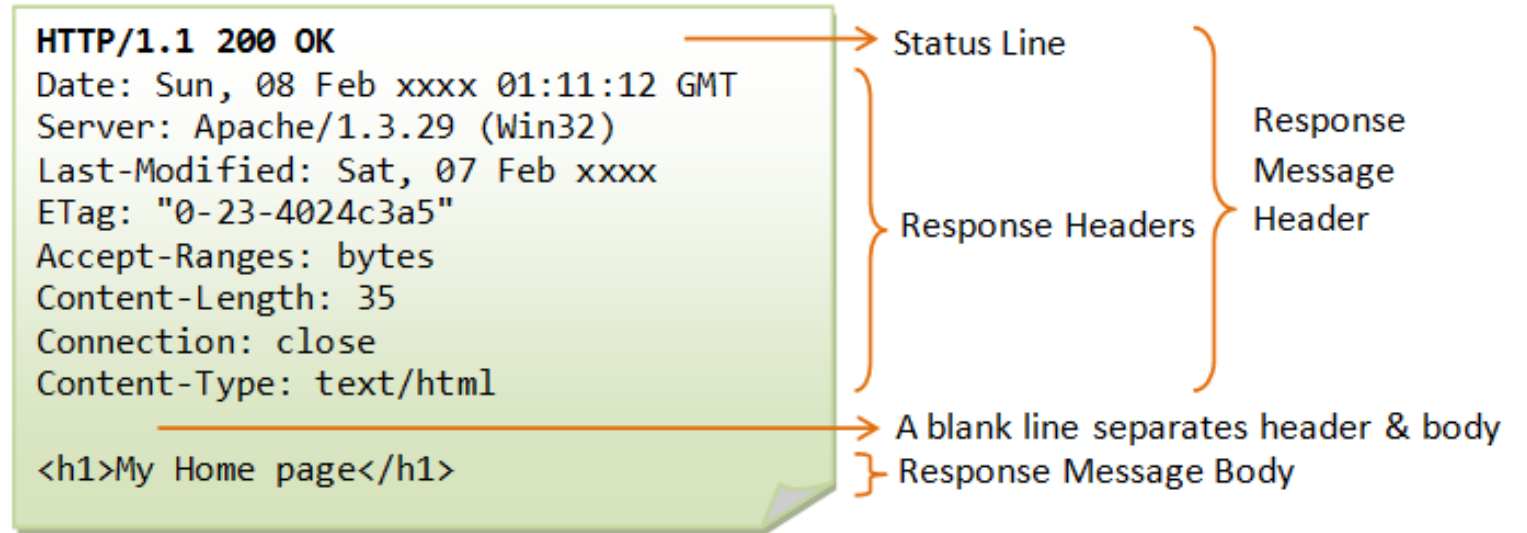
Content-Type

Set-Cookie

...

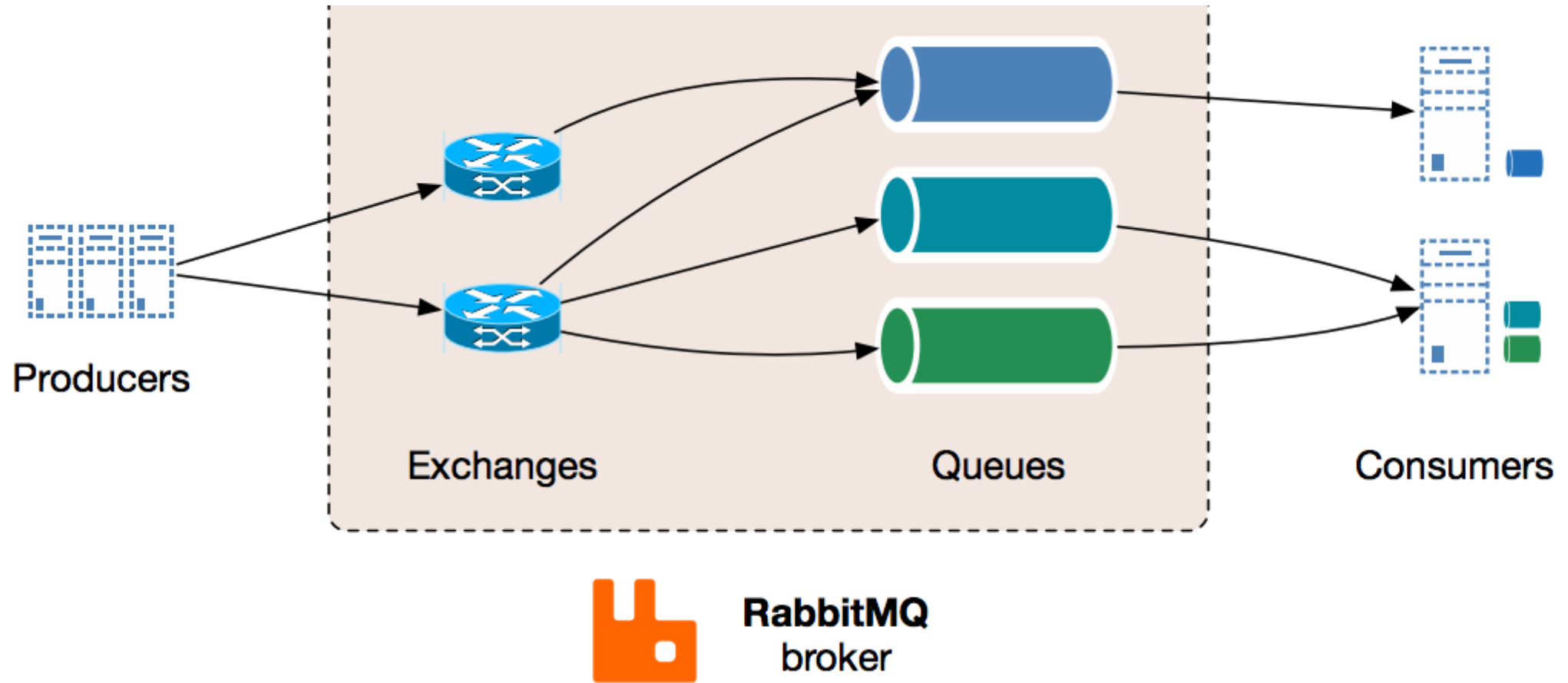
Body

Application specific (e.g. JSON ...)



MQTT

Message queues



See: <https://medium.com/must-know-computer-science/system-design-message-queues-245612428a22>

MQTT (ISO/IEC PRF 20922)



Overview

TCP/IP based

- MQTT-SN (UDP)

Small footprint / low bandwidth

e.g. compared to HTTP

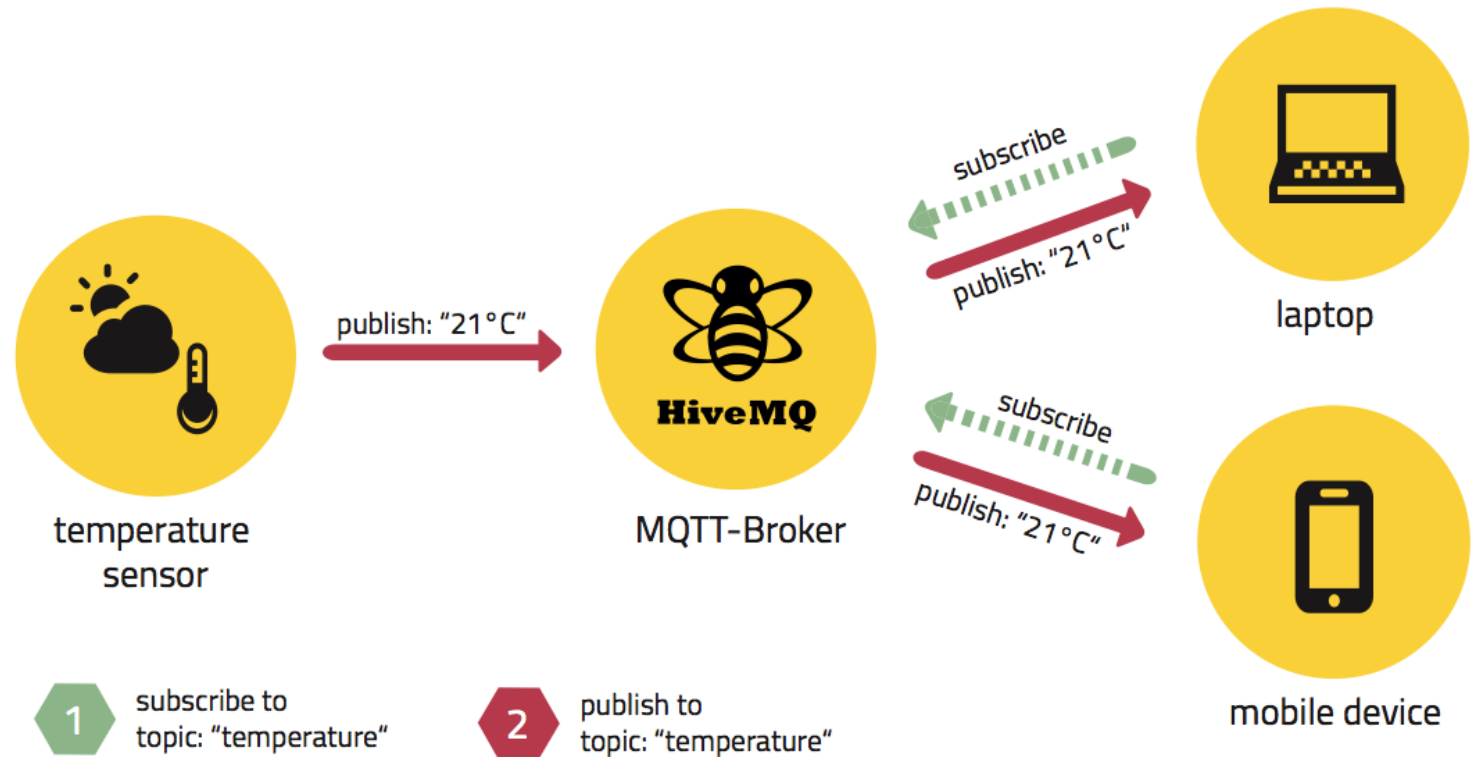
Pub/sub

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 Quality of Service (QoS)



QoS Level

Sender defines message QoS

Subscriber defines its supported QoS during subscription

Does not apply to TCP, only to sender-receiver connection

Levels

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

Practical tips

- 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)
- Avoid 2 - buffer locally (preferably in-memory) when server is not accessible (e.g. no route to host)

WARNING

Cloud providers may deviate
from the MQTT QoS
specification

MQTT security



Encryption

TLS on top of TCP/IP

Authentication

Username / password

Access key

Certificate / PK

