Aalto University
School of Electrical
Engineering

Communication, coupling & modularity

25.1.2019

Contents

- **Communication**
  - Characteristics of physical and protocol layers
  - Models, protocols and coordination
- **Communication in microservice architectures**
  - Failures
- **Coupling and modularity**
  - Types of coupling in different architectural approaches
  - Achieving modularity at different quanta

Communication networks

- **Even "virtualized" networks operate in physical reality**
  - Sometimes can assume locality e.g. loopback speeds (pods)
  - Distribution and decentralization may hide physical aspects
    - *Translation: System might be placed to straddle a buggy or oversubscribed router/switch*
  - Physical latencies: speed of light & electric signals, processing delays in switches and routers; retransmits

~ 40 000 km
* ½
/ 300 000 km/s
/ 2/3
= 100 ms

+ amplification delay
+ routing delay

Communication networks

- **Assumption of “infinite network capacity” in cloud may fail**
  - Loss of 50% of network capacity in a datacenter (backhoe)
  - Limits at virtual machines and physical servers (AWS ENA 25 Gbps)

  → even if you cannot saturate an IaaS PoP, you can closer to your service

- **Further delays and failures from protocols and OS**
  - Number of concurrent TCP connections (OS)
  - Bugs in protocol implementations (usually non-OS)

Network protocols

- **TCP almost universal, but**
  - In some situations UDP may be more suitable (within a service)
  - SCTP tends to keep popping up (alpha in K8S v1.12)
- **IPv6**
  - Getting more common, but still mostly user-side requirement
- **Deep magic**
  - TCP slow start algorithm — originally for congestion control
  - Anycast, multicast and broadcast (if you control subnets and/or routers)
  - VPNs and tunnels sometimes for integration (island hopping)

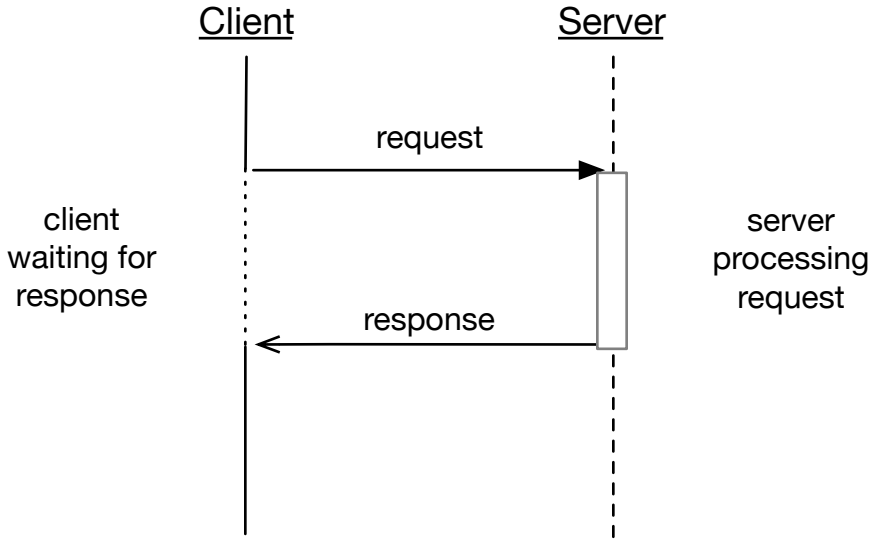
Application protocols

- **Almost all service interactions occur at application level protocols**
  - HTTP and HTTPS primary (QUIC in the future?)
    - *HTTP(S) used to transport other application level protocols*
    - *SOAP, REST, …*
  - gRPC, Thrift, AQMP, etc.
- **Operate on top of TCP**
  - Sometime work around TCP issues (such as slow start, with Keep-Alive connections)
  - TCP is connection-oriented: connect → transmit → close
  - Usually client-server, e.g. specific listener address and port

Communication models

- **Synchronous response**
  - Request-response pattern
  - Reply expected immediately (after processing)
- **Asynchronous response**
  - Processing started by request
  - Immediate response provides a handle or identifier
  - Response methods
    - *Polling by client (known endpoint or part of response)*
    - *Callback from server (agreed-upon endpoint or part of request)*
    - *Response publish (message queue, pubsub, blackboard, …)*
- **Message-passing**
  - Request itself asynchronous

Synchronous request



Asynchronous communication models