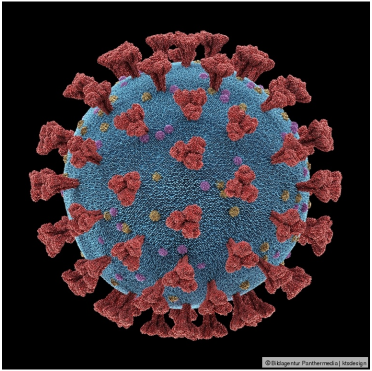
# COVID-19 Measures

- Wear a mask (medical or FFP2) until you have taken a seat
- When seated you may take off the mask if you can maintain an interpersonal distance of 1,5 m
- Open the windows periodically whenever possible
- Behave reasonable and use common sense



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# Distributed Systems
## Network Services

Prof. Dr. Oliver Hahm
Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

# Agenda

1. Motivation
2. Internet Protocols
3. Error, Flow, and Congestion Control
4. Server Architecture

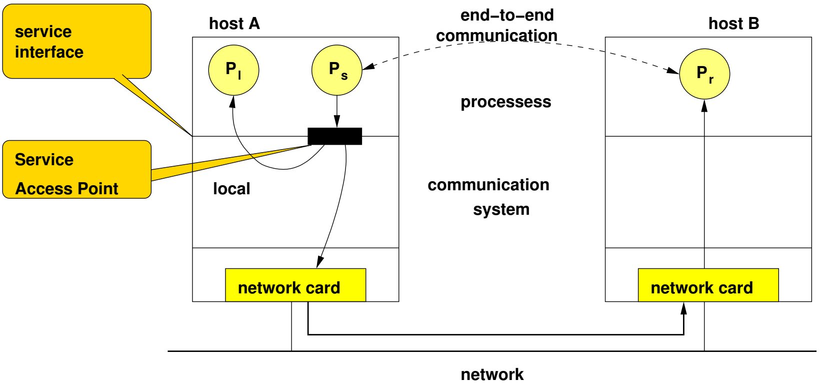
# Agenda

1. Motivation
2. Internet Protocols
3. Error, Flow, and Congestion Control
4. Server Architecture

# Typical Properties of a Communication Service

- End-to-end communication between processes
- Quality-of-Service (QoS) parameter
  - QoS characteristics, e.g.,
    - Performance characteristics (response time, throughput)
    - Real-time properties
    - Failure characteristics
- Transparency wrt network topology and transmission method
- Important internal aspects:
  - Error control
  - Flow control
  - Routing

# Generic Model for System wide Inter Process Communication (IPC)



# Agenda

1 Motivation

2 Internet Protocols

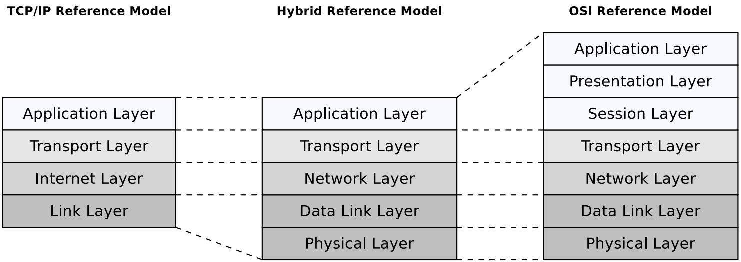
3 Error, Flow, and Congestion Control

4 Server Architecture

# OSI Reference Model

Central concepts of the OSI model are:

- **Services**: Define what the layer does, i.e., its semantics
- **Interfaces**: Define how to access it
- **Protocols**: Describe how the layer is implemented



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

What are the tasks, devices, and protocols for each layer?

# Tasks of Layer 1

- Layer 1: **Physical Layer (PHY)**
  - Definition of mechanical/electrical/optical specifications and procedures required for the Bit transmission on the carrier.
  - The Physical Layer can be subdivided into
    1. **PMD Sublayer**: Physical Media Dependent (mechanical/electrical/optical) specifications
    2. **PHY Sublayer**: Physical (Signals $\leftrightarrow$ bits: en/de-coding)
  - Transmits the ones and zeros

# Tasks of Layer 2

- Layer 2: **Data Link Layer (DL)**:
  - Provides mechanisms how two nodes on the same physical network guarantee a safe communication.
  - For this purpose, the information (= bits) are grouped together in **frames** and are complemented by checksum as final part of the frame. The checksum allows an error detection (and potentially correction).
  - The Data Link Layer can be subdivided two sublayers as well:
    1. **MAC Sublayer**: Media Access Control defines the access to the physical carrier and how nodes are addressed
    2. **LLC Sublayer** Logical Link Control provides an additional safety and control functionality

# Tasks of Layers 3 and 4

- Layer 3: **Network Layer**
  - The task of this layer is to transmit the data in **blocks** between the (network) nodes.
  - Those data blocks on layer 3 are called **packets**.
  - Forwards packets between logical networks (over physical networks)
  - Thus, layer 3 can be understood as packet exchange layer.
- Layer 4: **Transport Layer**
  - The transport layer provides a (potentially reliable) virtual **end-to-end connection** for the data transport between the end nodes
  - Acts as a **multiplexer** between the various processes on the hosts via ports
  - Transport protocols can implement connection-oriented or connectionless communication

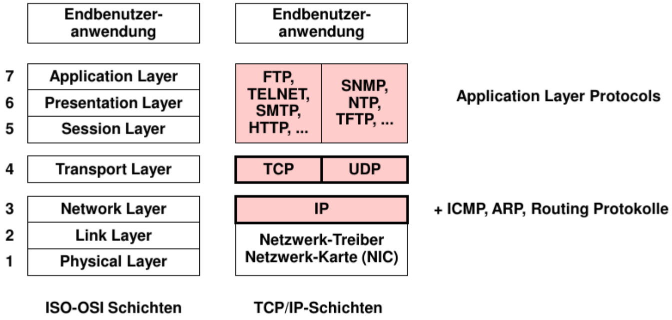
# Tasks of Layers 5 and 6

- Layer 5: **Session Layer**
  - This is the lowest application-specific layer and is responsible to raise, maintain, and gracefully terminate communication relationships between the end nodes: a **Session**.
  - It's particular scope is to provide dialog-functionalities among the communication partners, in particular to allow a synchronization between the involved communication processes.
- Layer 6: **Presentation Layer**
  - The conversion and translation of different data representations (e.g. character set families like ASCII and EBCDIC) to a common format prior of sending, is main task of the Presentation Layer.
  - This layer may include functionalities which allow compression of data, conversion, and encryption.
  - Known presentation schemes here were ASN.1 (**Abstract Syntax Notation No. 1**) and XDR (**eXternal Data Representation**); however XML (**eXtensible Markup Language**) is now mostly used instead.

# Tasks of layers 7

- Layer 7: **Application Layer**
  - Contains all protocols, that interact with the application programs (e.g., browser or email program)
  - Here is the actual payload (e.g., HTML pages or emails), formatted according to the used application protocol
  - Comprises the biggest variety of protocols

# Classification of Internet Protocols




| | | | |
|---|---|---|---|
| **IP:** | Internet Protocol | **SMTP:** | Simple Mail Transfer Protocol |
| **TCP:** | Transmission Control Protocol | **HTTP:** | HyperText Transfer Protocol |
| **UDP:** | User Datagram Protocol | **SNMP:** | Simple Network Management Prot. |
| **FTP:** | File Transfer Protocol | **NTP:** | Network Time Protocol |
| **TELNET:** | Remote Terminal Protocol | **TFTP:** | Trivial File Transfer Protocol |

# Main Characteristics of the TCP/IP Architecture

- Connectionless protocol (IP) on the network layer
- Packet switching via network nodes
- Static and dynamic routing
- Transport protocol with reliable end-to-end transmissions (TCP)

# Circuit versus Packet switching

- **Circuit switching**
  - A sustained (virtual) connection is present between both communication partners (end nodes) as long as the data transmission lasts
  - → End nodes must be attached to a specific exchange node to provide
  - Drawback: The communication breaks down, in case the connection fails
- **Packet switching**
  - The data to be exchanged will be encapsulated in packets as complete information units
  - These packets are dropped on the network and exchanged between the communication partners
    - Packets do include an address-information about the sender and the recipient node
    - Packets can be buffered on the transmission path
  - Drawback: Some per-packet overhead for long-lasting connections

# Summary IP

**Essential Properties of IP**

- **Connectionless** protocol
- **"‘best-effort"’** transport of individual messages (Datagram, =Packet)
- Addressing of hosts via 32 bit (IPv4) or 128 bit (IPv6) **IP addresses**
- **Fragmentation** if necessary (optional in IPv6)
- IPv4 contains a checksum for the header, but not the payload; IPv6 does not contain any checksum
- IPv4 packets may contain optional header fields, IPv6 uses extension headers
- The **lifetime** of a packet in the network is limited

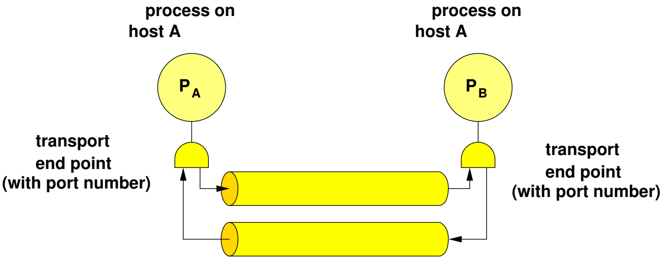
# TCP, UDP, and the Transport Layer

- Tasks of the **Transmission Control Protocols (TCP)**:
  - Reliable bidirectional point-to-point transport of a bytestream between two hosts on the endpoints
- Tasks of the **User Datagram Protocols (UDP)**:
  - Best-effort datagram service of IP layer is accessible for the processes on the endpoints
- Classification:

| | | |
|---|---|---|
| Application Layer (OSI Layer 5-7) | FTP<br>TELNET<br>SMTP<br>HTTP<br>rsh<br>rlogin<br>... | NTP<br>SNMP<br>TFTP<br>... |
| Layer 4 | TCP | UDP ⟸ |
| Layer 3 | IP | |

# TCP Communication Model

- connection oriented
- virtual, bidirectional, full-duplex capable connection between endpoints (used by their processes)
- Addressing of transport layer endpoints via 16 bit port numbers (in addition to the node's IP addresses)
- Bytestream oriented, not blockwise
- A single packet containing a chunk of the bytestream is called segment



# Summary TCP

**Further Characteristics of TCP:**

- Reliable transmission due to ...
  - Sequence numbers
  - Checksum calculation (same algorithm as IPv4)
  - Reception receipts (acknowledgements) and timeouts
  - Retry after timeout
- Sliding window principle for flow control
- Urgent Data and Push Function for high-priority data

# Summary UDP

**Essential Properties of UDP:**

- Connectionless protocol
- Addressing of the user via 16 bit port numbers
- "‘best-effort"’ transmission of datagrams (individual messages), IP services from the network layer are made accessible for application processes $\rightarrow 1:1$ mapping
- Multicast/Broadcast capable (1:n communication), direct application of multicast capable networks like, for instance, Ethernet
- Integrity check via an optional checksum
- No receipts or other guarantees, i.e., datagrams can get lost, arrive in a different order, or getting duplicated
- No flow control, i.e, on the receiver site datagrams may get discarded in case of full or missing buffers
- Well suited for the implementation of simple request/response protocols

For which type of application would you use TCP and for which UDP?

# Agenda

# Network Performance

- Data and messages, which are transmitted over networks may be lost and/or distorted:
  - Insufficient quality of the transmission carrier
  - External distortion impacts (e.g., because of electromagnetic fields)

$\hookrightarrow$ Both impacts causing data corruption or even data loss.

- The communication protocol has to deal with these cases and has to provide:
  - **Error/Fault Control**:
    Identification and compensation for transmission errors/failures
  - **Flow Control**:
    Adaptive means to adjust the amount of data to be send w.r.t. the recipient's (announced) capacity
  - **Congestion Control**:
    Additional means, to reduce the potential lost of data (packets) on the network

# Failure Causes

### Signal Transmission Errors

During the transmission of bit sequences on the physical layer errors may occur

They are typically caused by...

- Signal deformation
  - Attenuation of the transmission medium
- Noise
  - Thermal or electronic noise
- Crosstalk
  - Interference by neighboring channels
  - Capacitive coupling increases with increasing frequency
- Short-time disturbances
  - Cosmic radiation
  - Defective or insufficient insulation

### Error Types

Burst errors are more common than single bit errors

### Typical BER values

| | |
|---|---|
| POTS | $2 * 10^{-4}$ |
| Radio link | $10^{-3} - 10^{-4}$ |
| Ethernet | $10^{-9} - 10^{-10}$ |
| Fiber | $10^{-10} - 10^{-12}$ |

The LLC sublayer tries to detect and handle bit errors that occur during signal transmission

# Checksum

## Checksum

The checksum is calculated by a pre-defined algorithm for a block of data. They are typically used for the verification of the data integrity.

- For error detection, the sender attaches a checksum at each frame
- The receiver can now detect erroneous frames and discard them
- Possible checksums:
  - Parity-check codes
  - The polynomial code – Cyclic Redundancy Checks (CRCs)

# Error Control

- In order to detect transmission errors on the upper layers positive $A^+$ and negative **Acknowledgments** $A^-$ are feasible.
  - However, Acknowledgments can be compromised as well (worst case).
  - The Sender has to consider a **deferment period**, until the Acknowledgment has been finally received.
  - In addition, the data blocks (or even the transmitted Byte) can be **enumerated** (bookkeeping).
- The Sender has to keep the transmitted data in his **sending buffer** until he finally has received the Acknowledgment.

# Flow control

- Flow control enables the adaption of the transmission rate of the sender with respect to
  - the recipient or
  - any network component which is responsible for the data transfer (Gateway).
- Typical flow control methods:
  - Messages **hold and continue** (`XON/XOFF`) issued by the recipient also know as Ready-for-Reception/Clear-to-Send (RFR/RTS+CTS),
  - By issuing **credits**
  - **Window mechanism**, where the communication partners mutually tell their reception buffer to each other and adjust the data to transmit according to the provided value.

# Congestion control

- Any physical network as only a certain capacity to transmit only a certain amount of data (data packets) during a certain time period.
- Congestion will be encountered, if
  - the recipient buffer in any network component is exaggerated and incoming data packets need to be dropped,
  - the sender is required to build up additional send buffers (**queues**), without being able to transmit the data packets on the network.
- congestion avoidance is task of congestion control, since any congestion will impact
  - the data **throughput** and
  - the **transfer latency** (delay)

  negative.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# Agenda

1 Motivation

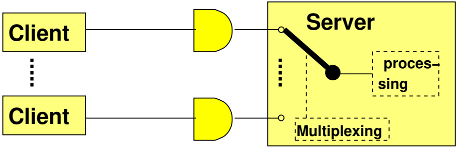
2 Internet Protocols

3 Error, Flow, and Congestion Control

4 Server Architecture

# Server Architecture

- Architectural principles for the internal structure of server processes
  - **Problem:** A server typically needs to communicate with multiple clients at once



# Models

- Simple sequential server
- Sequential server as state machine
- Parallel server processes
- Multithreaded server

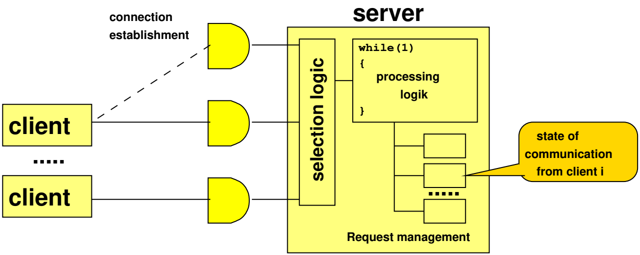
# Simple Sequential Server

- **One** process handle the requests of all clients one after another

→ Problem if the server acts as a client towards another server while processing a request: ⇒ the whole server gets blocked!

- **Drawbacks:**
  - No concurrency in the server
  - No use of (a potentially) underlying multicore architecture by a single server process
- This approach is hardly acceptable for productive applications in the traditional Internet, but may be applicable for very constrained devices

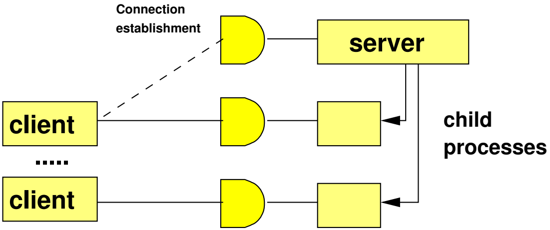
# Sequential Server as State Machine




- No internal blocking:
  multiple requests can be handled in an overlapping manner
- Multiplexing "'by hand"' ⇒ complex to program
- Selection logic in UNIX:
  - non-blocking requests (Option `O_NDELAY`) and polling
  - `select()`

# Parallel Server Processes

## Architecture:




- Child processes preserve the current state of communication per remote peer in memory
- Advantage: Multicore architecture can be used
- Problem: Expensive process handling ($\rightarrow$ context switches)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

# Multithreaded Server

- Automated resolution of the multiplexing problem
  - A thread is permanently assigned to each request at the start of processing
  - Each single thread of the server may block at any point of time without affecting the overall concurrency

  → Thread pool is required
- Applicable for all paradigms of distributed applications
- Requires synchronisation!

# Current State of Multithreading

- All modern operating systems and runtime environments support threading
- Even many embedded operating systems (like RIOT) support multithreading by now
- Typical APIs
  - pthreads POSIX 1003.4 (C/C++)
  - Boost threads (C++)
  - Java Concurrency since SE 5: `java.util.concurrent`

Important takeaway messages of this chapter

- The Internet's TCP/IP architecture provides a flexible and generic communication system for many types of higher layer services
- The network has to manage transmission errors and control the data flow
- Depending on the use case various architectures to design a server application are possible