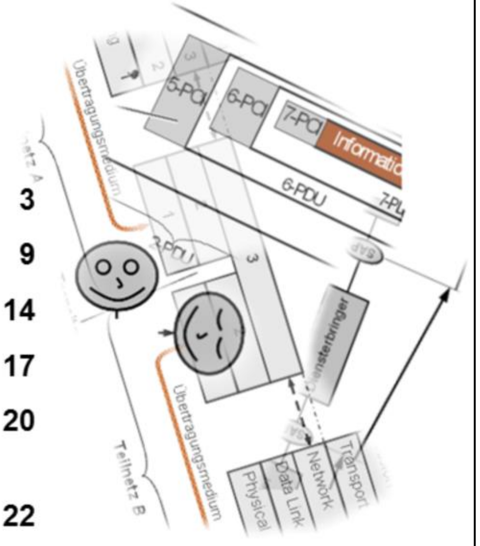
Chapter 1:

Fundamentals Review

List of Contents for this Chapter:

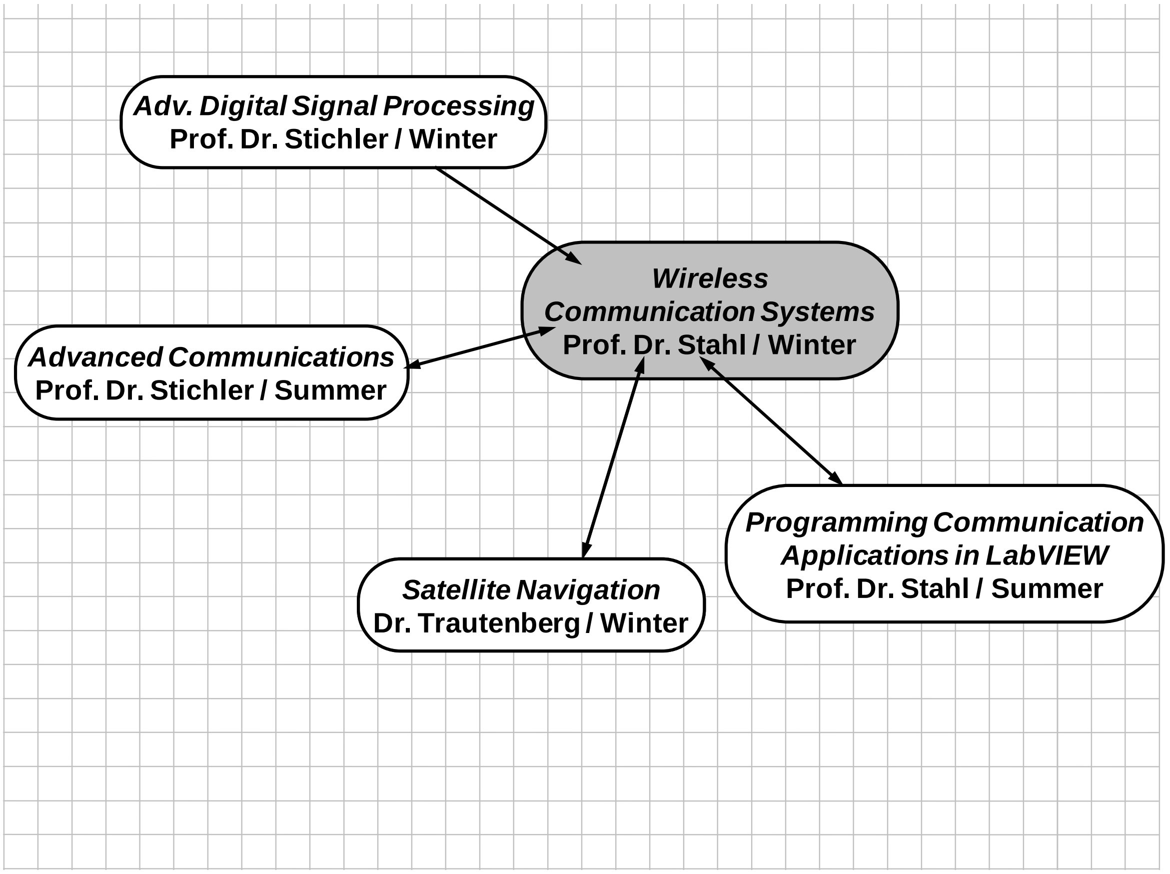


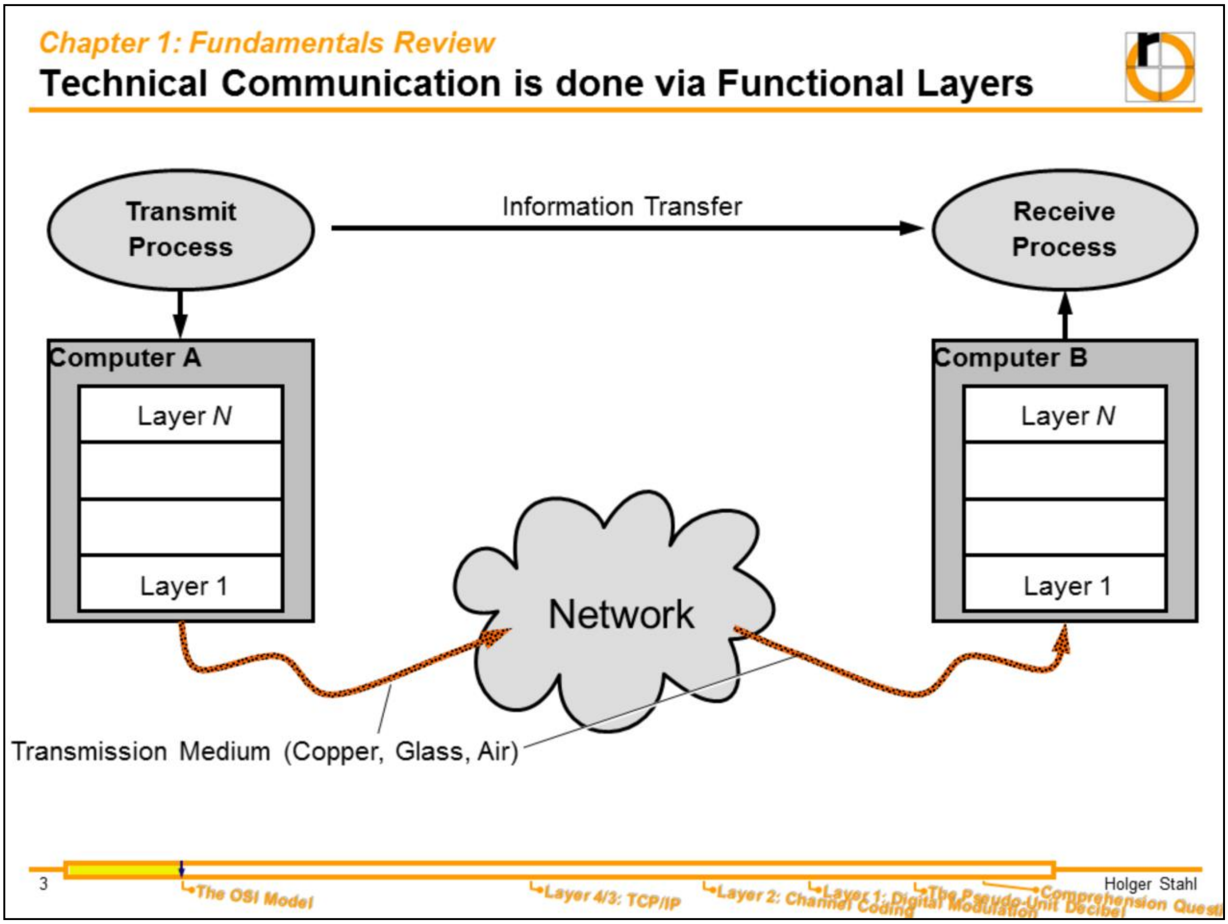
This chapter tries to review some important fundamentals for this class.

Prerequisites for this Lecture

- In the fields of ***Communication Protocols***:
  - The ISO/OSI reference model
    - ⇒ Basic concepts of *Protocols*, *Interfaces* and *Services*
  - Protocols:
    - ⇒ The TCP/IP protocols
    - ⇒ Error Coding, especially CRC (*Cyclic Redundancy Check*)
    - ⇒ FEC and ARQ Protocols (like *Stop-and-Wait* or *Sliding Window*)
    - ⇒ Understanding of a 2G and/or 3G communication standard
- In the fields of ***System Theory***:
  - Representation of signals and systems in the *time and frequency domain*
  - Sampling and reconstruction of signals
  - *Analogue* ↔ *Digital* Conversion of signals
  - Usage of the pseudo-unit "Decibel"
- In the fields of ***Communications***:
  - Digital modulation techniques like *QAM, PSK*, etc.
  - *Low Pass* ↔ *Band Pass Transformation*

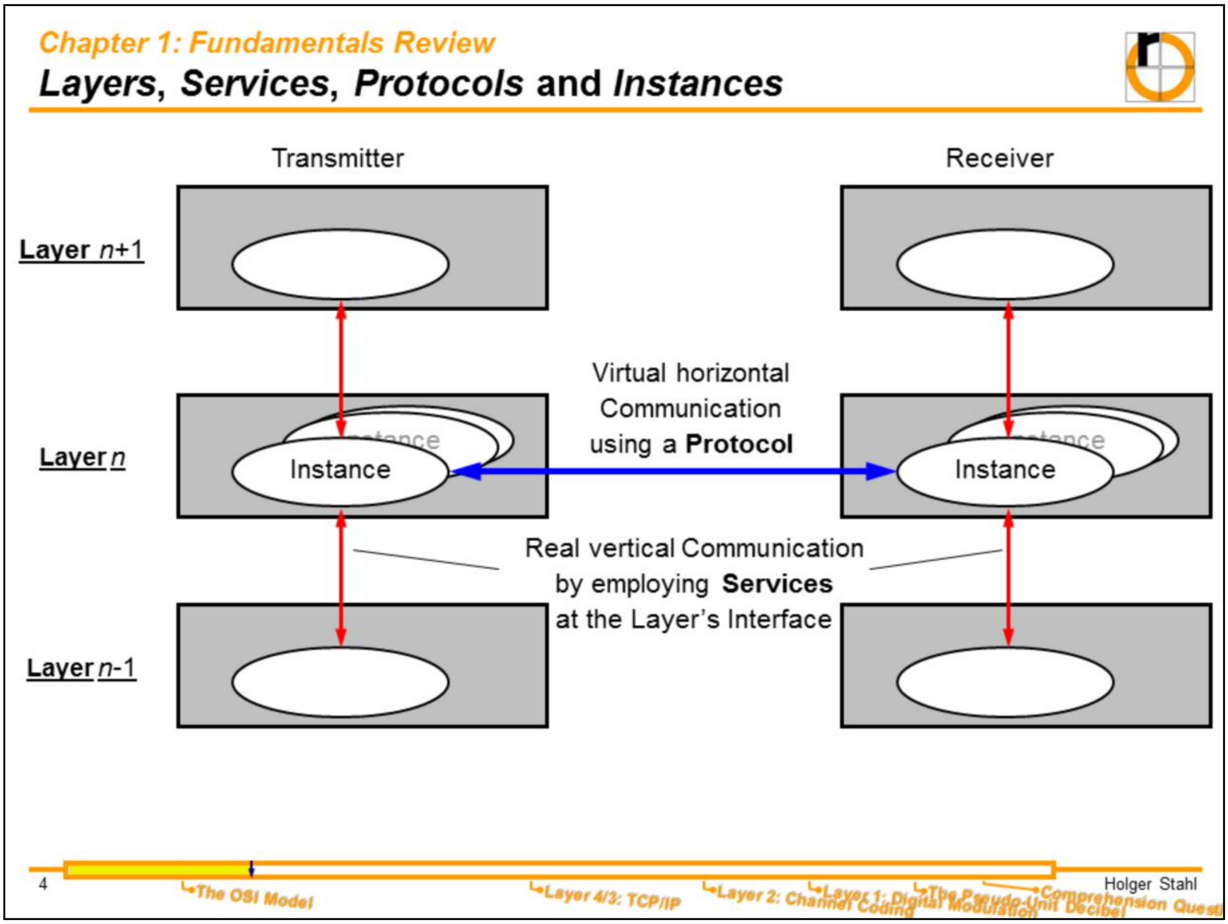
Related Lectures at Rosenheim University of Applied Sciences





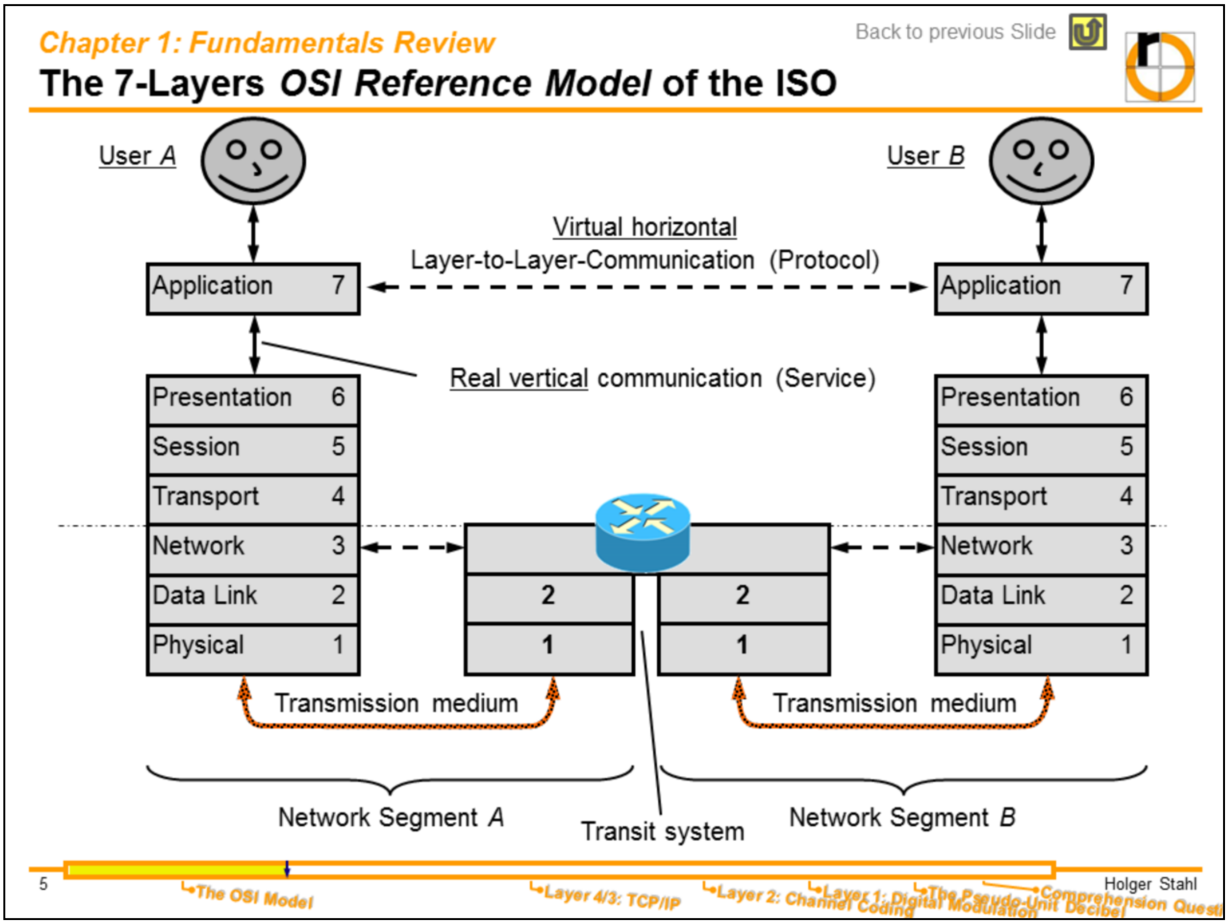
The technical process of information transfer between two computer systems (*Communication Systems*) is usually split up into several small sub-tasks. Therefore, so-called *Layers* are defined which prepare the information to be sent over a physical medium. Vice versa, the receiver has to restore the original information after the transmission. The principle of breaking up the tasks into Layers represents a <u>hierarchical modularisation</u> of the communication process.

This practice is applied in a similar way in software development. The aim is a reduction of the development effort: Each module to be developed stays manageable, because each Layer defines upper/lower interfaces with a strongly limited functionality. The functionality of the lower layers can be used by the upper layers, thus avoiding ‘reinventing the wheel’.



In the following, we look at the OSI (*Open Systems Interchange*) Reference Model for Communication, which strictly distinguishes the following principles:

- ***Layers*** = Modularisation of the communication tasks:
  - Each Layer comprises a functionality, which is strictly separated and independent from the upper and the lower Layer.
  - Within each Layer, instances communicate exclusively with instances of the neighboured Layers.
  - By bringing identical and similar functions together in one Layer, the interfaces between the layers can be kept slim.
- ***Protocols*** of each Layer = virtual horizontal Communication:
  - Each Layer implements a protocol, that features the communication between instances of one layer.
  - Each of the involved protocol instances virtually communicates with its communication peer of the same Layer, i.e. instances of the same Layer communicate using the same ‚language‘.
- ***Services***, offered at the *Interfaces* = real vertical Communication:
  - In reality, only the lowest layer can actually exchange messages with other systems. All other data transport within the protocol stack happens vertically, by offering services towards the next higher layer.
  - The service offered by Layer *n*, employs and includes the services and the functionality of all lower Layers.



The OSI-Model:

The architecture model for *Open Systems Interconnection* (OSI) was founded in 1983 of the *ISO* (*International Standardisation Organisation*). The model aimed a free Communication between Devices of various manufacturers. The ISO/OSI reference model is based on the following principles:

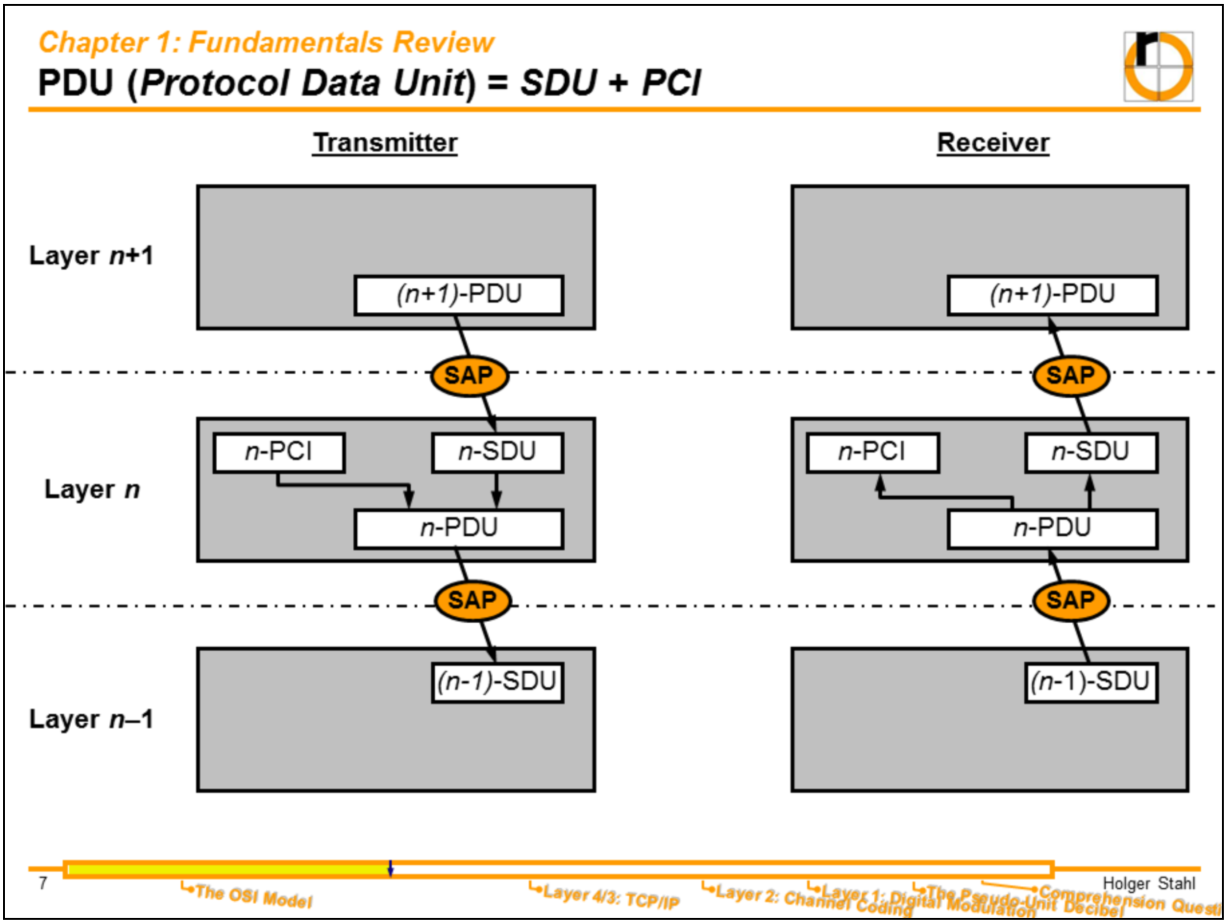
- Organisation of the functionalities necessary for communication in a hierarchical *Stack*.
- Each lower layer *N*-1 in the stack offers *Services* for the next higher layer *N*
- Communication between identical Layers *N* on different systems is done by the so-called *Protocols*.

Tasks of the 'Networking' Layers 1…3

- **Layer 1 – *Physical Layer***
  - Transfer of frames of bits over a physical medium
  - Parameters for the conversion bits ↔ symbols: *Power*, *Frequency*, *Modulation Scheme, etc.*
- **Layer 2: – *Data Link Layer***
  - Error-free transfer of Layer-3 data over a network segment
  - Flow Control, Error Recovery
  - Addressing of systems (computers) at a common medium
  - Medium Access Control (MAC)
- **Layer 3: – *Network Layer***
  - World-wide addressing of systems (computers)
  - Choice of the route that the data will take ⇒ World-wide Routing, Switching

Tasks of the 'End-to-End' Layers 4…7

- **Layer 4 – *Transport Layer***
  - Addressing of applications (programs) within a system (computer)
  - Provision of an error-free End-to-End link with constant quality
- **Layer 5 – *Session Layer***
  - Arrangement and control of sessions
  - Access control (*Authentication*, *Authorisation*)
- **Layer 6 – *Presentation Layer***
  - Ciphering
  - Data Compression (by removing redundancy and irrelevance)
- **Layer 7 – Application Layer**
  - Applications are the source/sink of the information to be exchanged
  - Some Applications offer services that are directly available to the user, e.g. eMail, File Transfer, Web-Browser

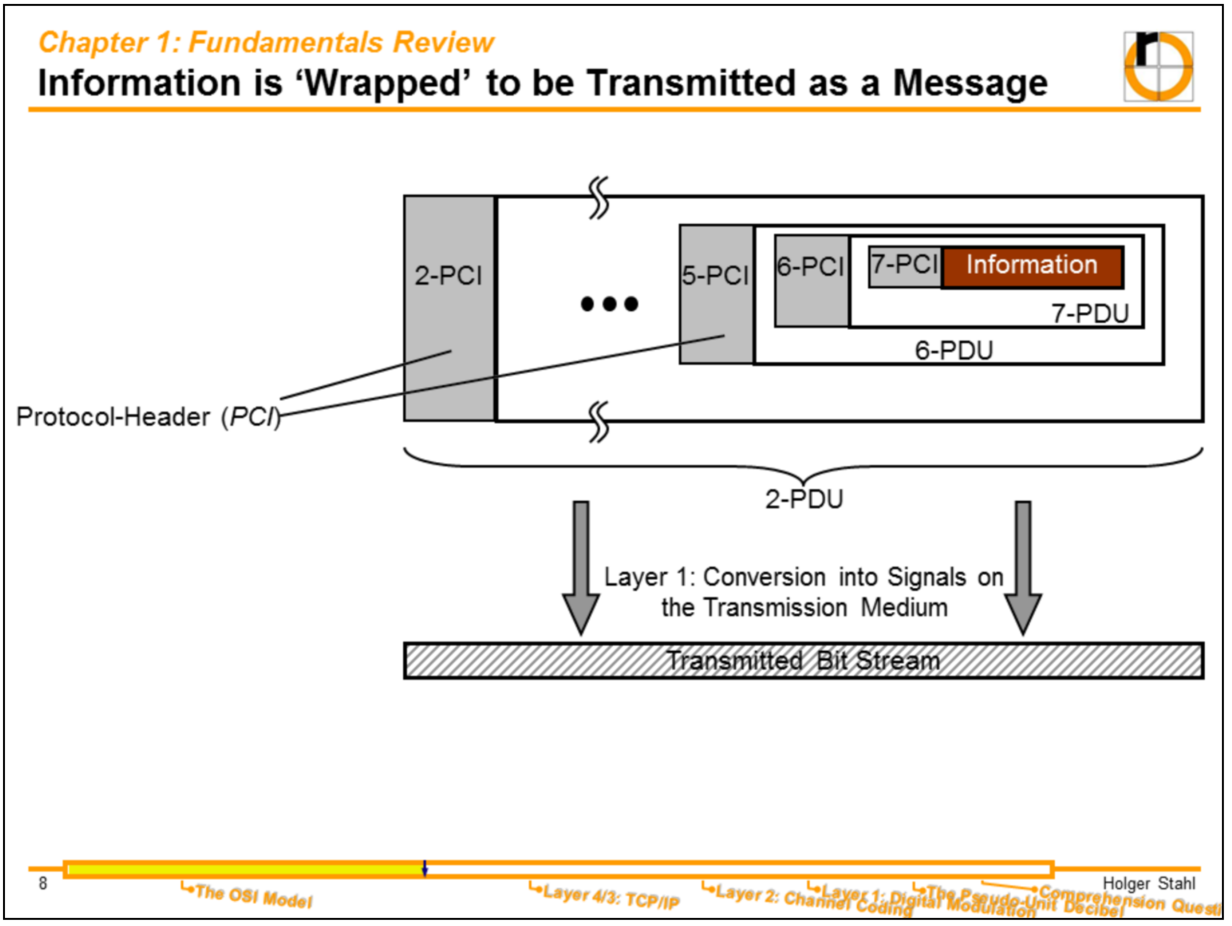


Within the OSI (*Open Systems Interchange*) Reference Model, each layer adds a certain Protocol Control Information (PCI) to the message that it gets from the layer above:

PDU = SDU + PCI

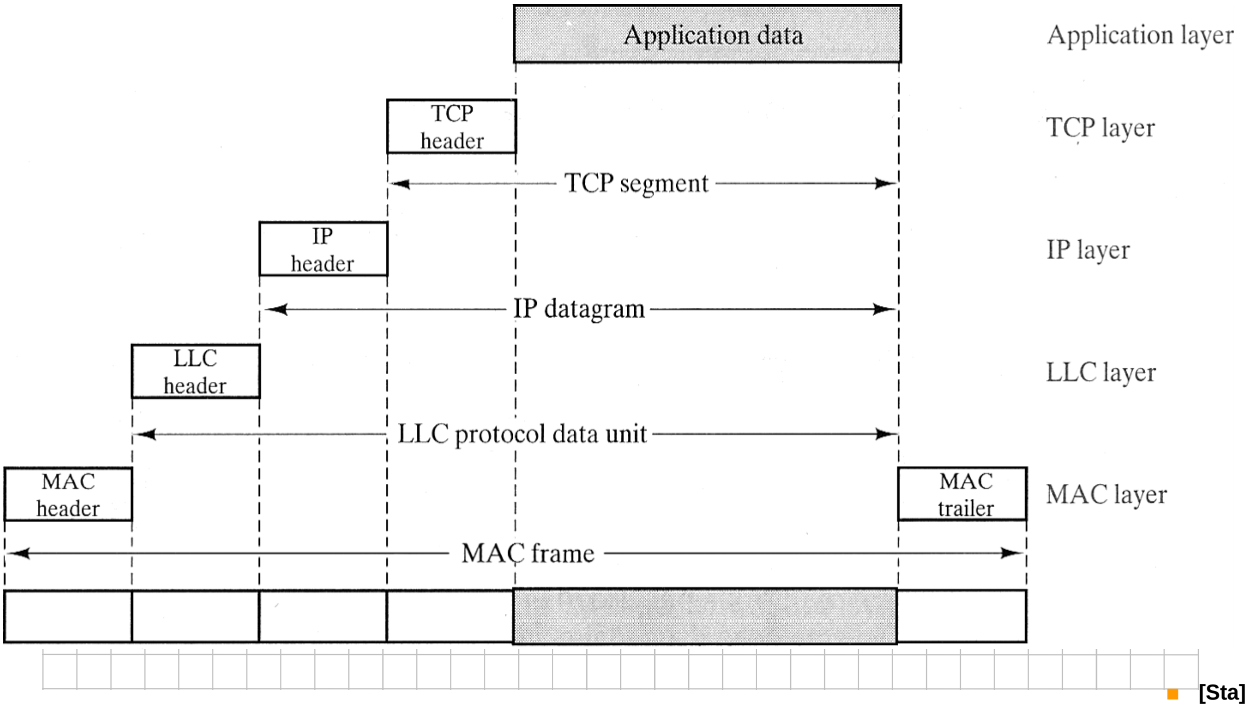
- ***Protocol Data Unit* (PDU)**: The data unit exchanged between peer entities of the same protocol layer. On the downward direction, it is the data unit generated for the next lower layer. On the upward direction, it is the data unit received from the previous lower layer.
- ***Service Data Unit* (SDU)**: The data unit exchanged between two adjacent protocol layers. On the downward direction, it is the data unit received from the previous higher layer. On the upward direction, it is the data unit sent to the next higher layer.

The Interfaces between two Layers are called *Service Access Points* (SAP). Each SAP leads to exactly one single protocol instance in the upper and in the lower layer.



The picture above illustrates, how information is wrapped by protocol control information of each Layer of the protocol stack.

<u>Contents of a MAC-Frame inside a LAN (*Local Area Network*):</u>

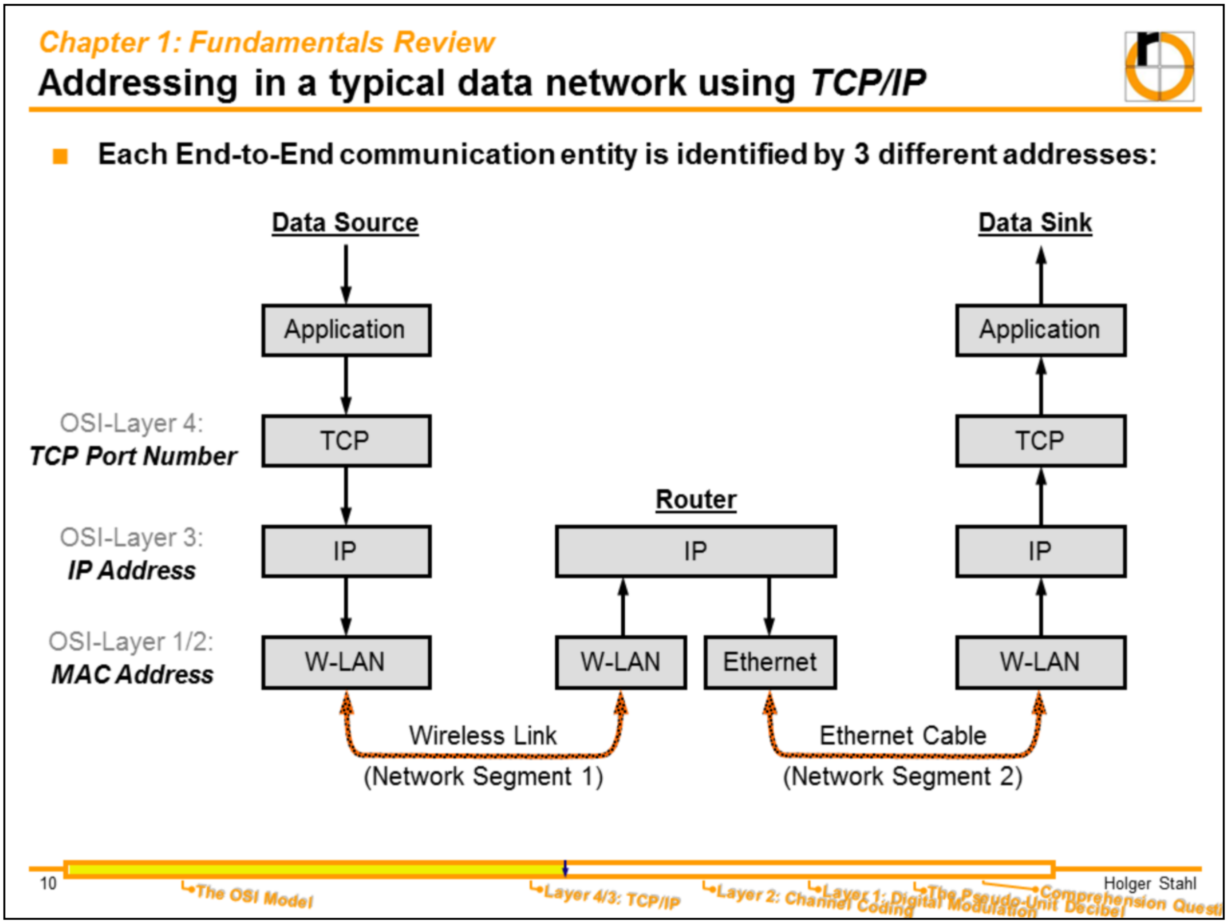


[Sta]

On OSI-Layer 4/3: Transport and Switching by *TCP/IP*

Services:

- **TCP (*Transport Control Protocol*)**
  - Represents OSI Layer 4: *Transport* (see figure before)
  - Supports virtual connection-oriented exchange of data
  - Supports error control & flow control, data is ordered in the correct sequence
  - Supports addressing of applications on the local system by using the *TCP port number*
- **IP (*Internet Protocol*)**
  - Represents OSI Layer 3: *Networking* (see figure before)
  - Supports unreliable packet-switched exchange of datagrams
  - No error control, no flow control, no guarantee of the sequence order
  - Supports World-wide addressing of systems by using the *IP address*



<u>**The three pairs of addresses identifying each End-to-End communication Session**</u>

- **TCP Port Number:**
  - Identifies <u>an Application</u> within <u>a host system</u>
  - Format: Ports are represented by a 16-bit number, usually represented decimal
  - For the source port, the client selects an arbitrary local port number that is not in use. The destination port number is fixed by the application to be addressed on the server.
  - Many port numbers in the range 1…1,024 are assigned fixed (*well-known ports*):

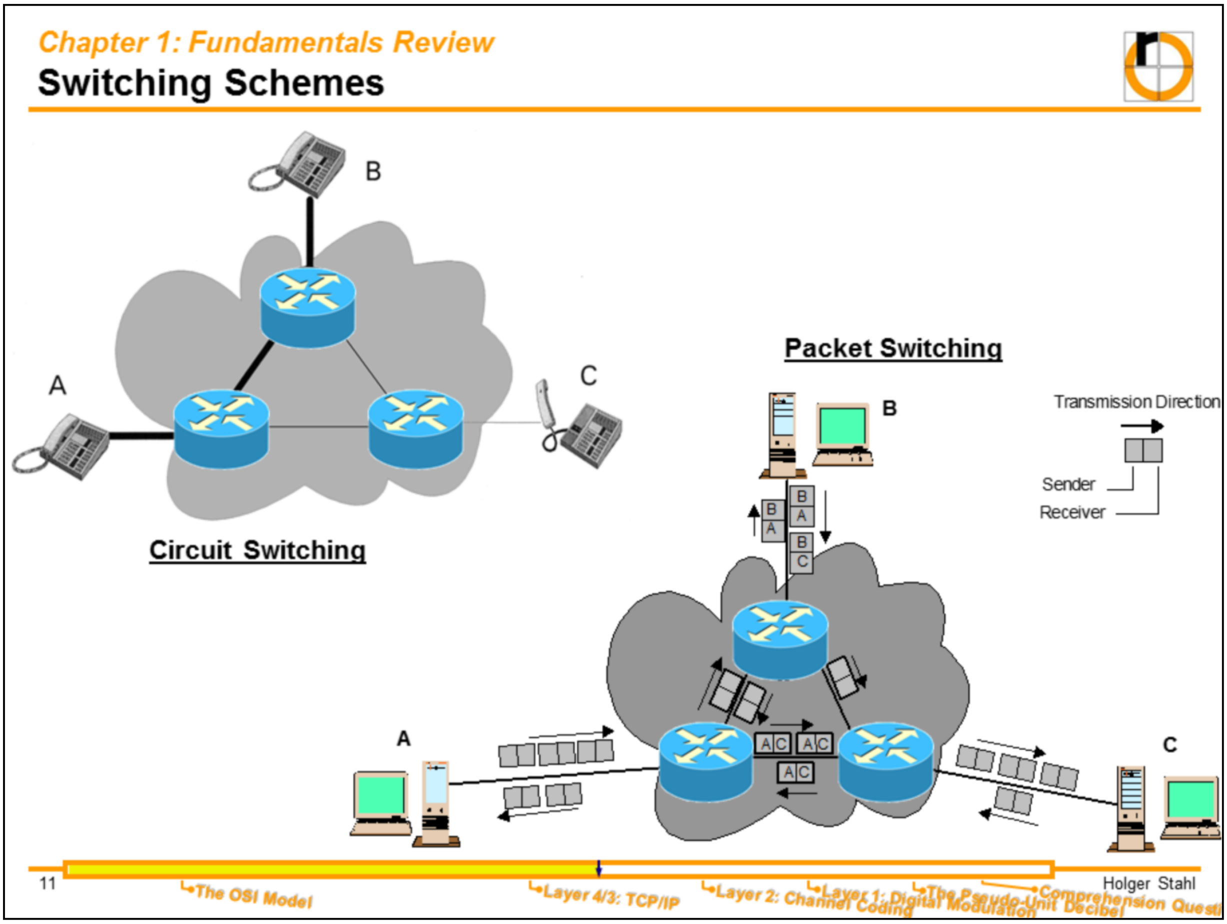
application	port number
Day time	13
FTP Control	20
FTP Data	21
Telnet	23

application	port number
SMTP	25
DNS	53
WWW	80
POP3	110

- **IP Address:**
  - Identifies <u>a network adapter</u> of a specific host <u>world-wide</u>
  - Format: Source and destination IP address are a 32-bit number, which usually is written as a combination of 4 decimal numbers *a*, *b*, *c*, and *d* in the range 0…255

aaa . bbb . ccc . ddd

- **MAC Address:**
  - Identifies <u>a network adapter</u> within <u>a Network Segment</u>
  - Format: 48-bit, usually being written as a combination of 6 hexadecimal numbers



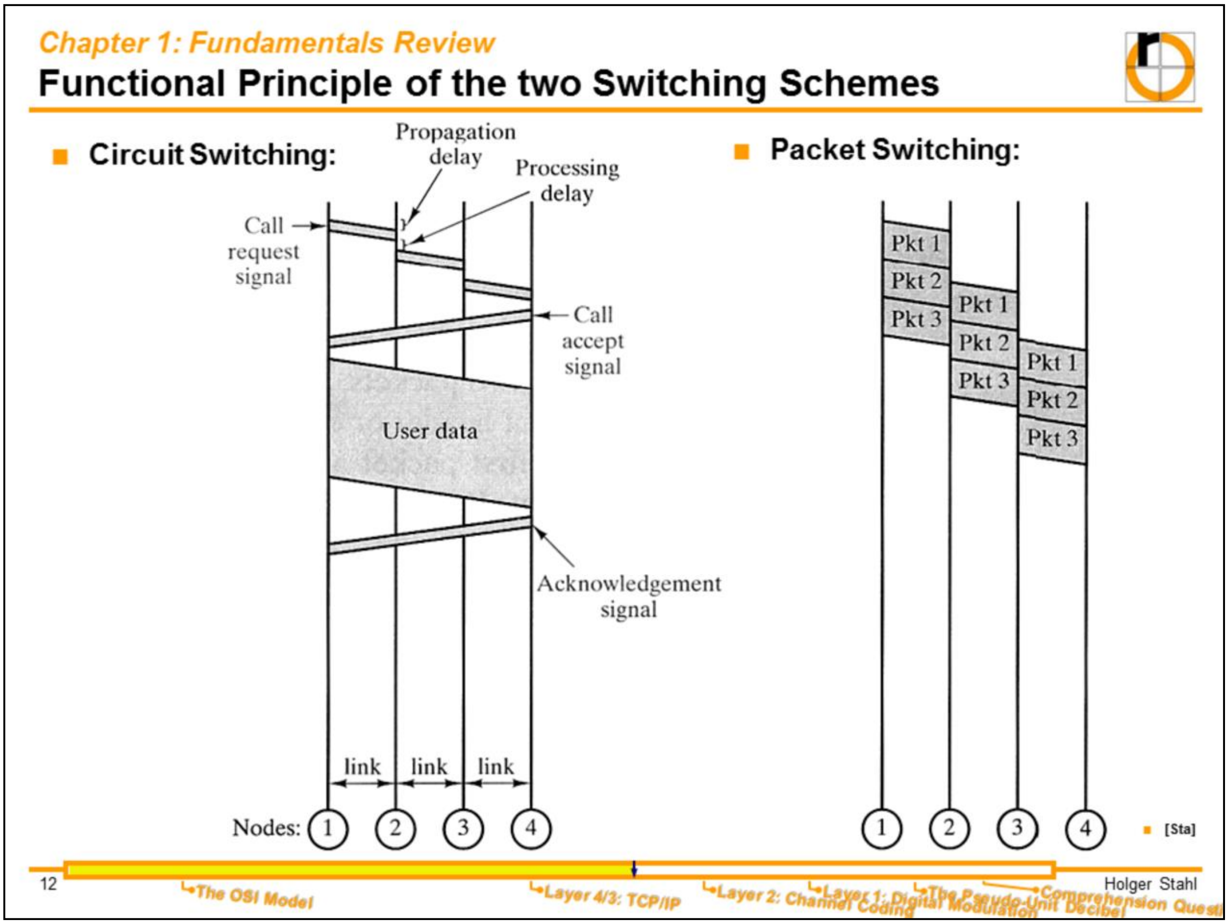
Two basic Switching Schemes are distinguished:

- **Circuit Switching** provides an exclusive physical link during the entire duration of the communication session.
- **Packet Switching** transports the data in packets with a limited size. Each of these packets is stamped with the destination address and then transported individually.

Circuit Switching	*Packet Switching*
Permanent physical Connection Link is exclusively available for the user	**Datagram Transfer** Service similar to the traditional mail, i.e. each packet will be transported individually to the recipient.
Connection-Oriented Example Telephone: Connection Setup, Data Transfer, Connection Shutdown	**Connection-Less** Example *Internet Protocol*: Data is exchanged block-wise without Connection Setup/Shutdown

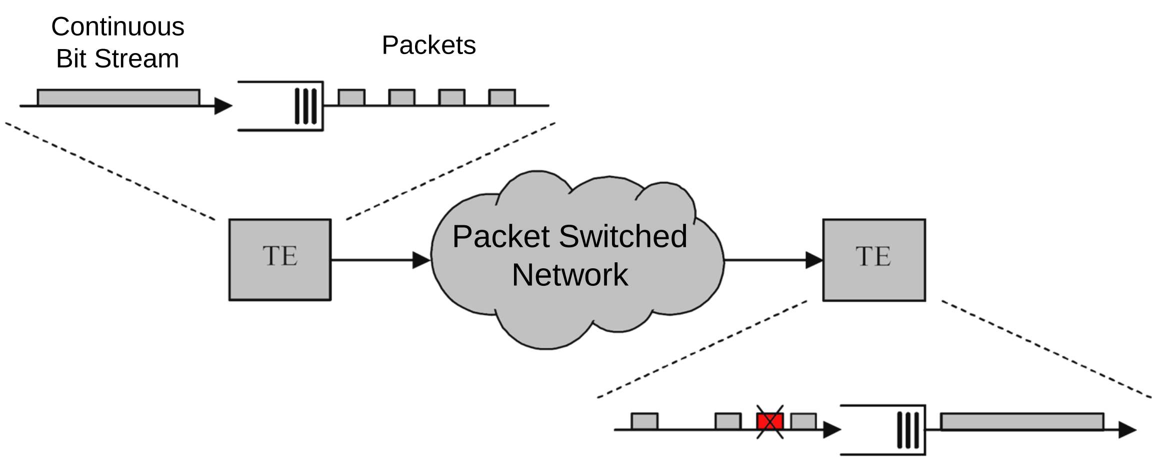
Properties of *Packet Switching* compared to *Circuit Switching*:

+ Transmission bandwidth is only needed and reserved on demand
+ Typically, the accounting is done based on the amount of traffic (i.e. the transmitted data) rather than the link time.
− Non-transparent Transmission ⇒ Each *Switching Node* (= *Router*) uses individual link parameters for the next hop
− Continuous data streams must be segmented into pieces before it can be transmitted in form of packets. The receiver needs to reassemble the original data stream in the right order again.

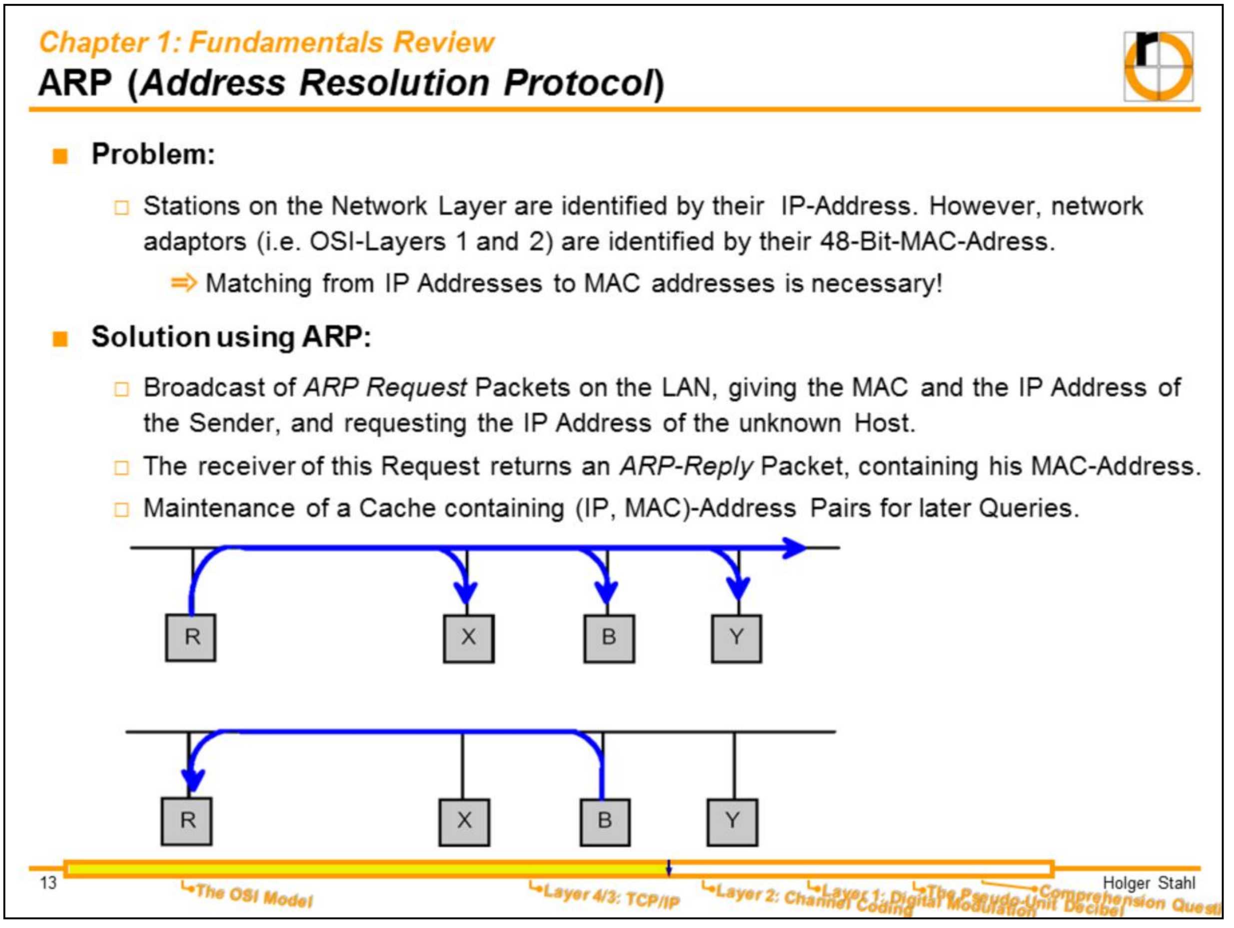


How is Continuous Data transmitted over a Packet-Switched Network?

- Simulation of Circuit Switching by Buffering and Packetization
  - ⇒ Delay of the data by Buffering and Switching
  - ⇒ Possible errors caused by lost packets and wrong order of the packets
- Example: Voice-over-IP:



- Figure modified from [Carsten Roppel: *Übertragungstechnik – Signalverarbeitung – Netze*. Hanser, München]



In order to deliver an IP Datagram, at first a matching from the 32-Bit IP address to the 48-Bit MAC address has to be performed. This is done by a dynamic table (the so-called *ARP Cache*) with known assignments of IP addresses und MAC addresses, which every node (this can be the source host or a router) separately manages. At first, the IP-Adresse of the next node. This can also be a router! Subsequently, the node checks the ARP Cache, if an entry for the MAC address of the destination node is contained:

- If this **search was successful**, the packet is sent directly over the LAN (i.e. Layer 1 and 2!) to the MAC address that has been found.
- If the IP address of the next node could not be found in the *ARP Cache*, an *ARP-Request* packet with the IP address of that node will be generated, and broadcasted into the LAN. All nodes of the respective network segment will receive this ARP broadcast, so comparing the requested IP address with their own IP address.
- That node, the IP address of which matches, answers by using an *ARP Reply* packet. Beside the requested sender's address, the answer contains the requested MAC address as well.
  - Now the sender of the *ARP Request* knows the MAC address, to which the data packed has to be forwarded to. The address pair contained in the ARP-Reply will be added to the ARP Cache.
  - In order to reduce the frequency of ARP Requests, also all other nodes, which were not involved in the ARP process, add the pair of addresses to their Cache.

In order to react on dynamic situations (i.e. change of addresses) in the network, all entries in the ARP Cache carry an expiry date, which is 5 min per default. Hence ARP broadcasts represent a significant part of the network load, since broadcast packets are forwarded to all nodes of a network segment.

Chapter 1: Fundamentals Review

On OSI-Layer 2: *We observe Bit Errors of the Physical Layer*

- Definition of an 'Error':

The received data differ from the original.

- Typical BER (*Bit Error Rates*) p_E of usual Transmission Media:

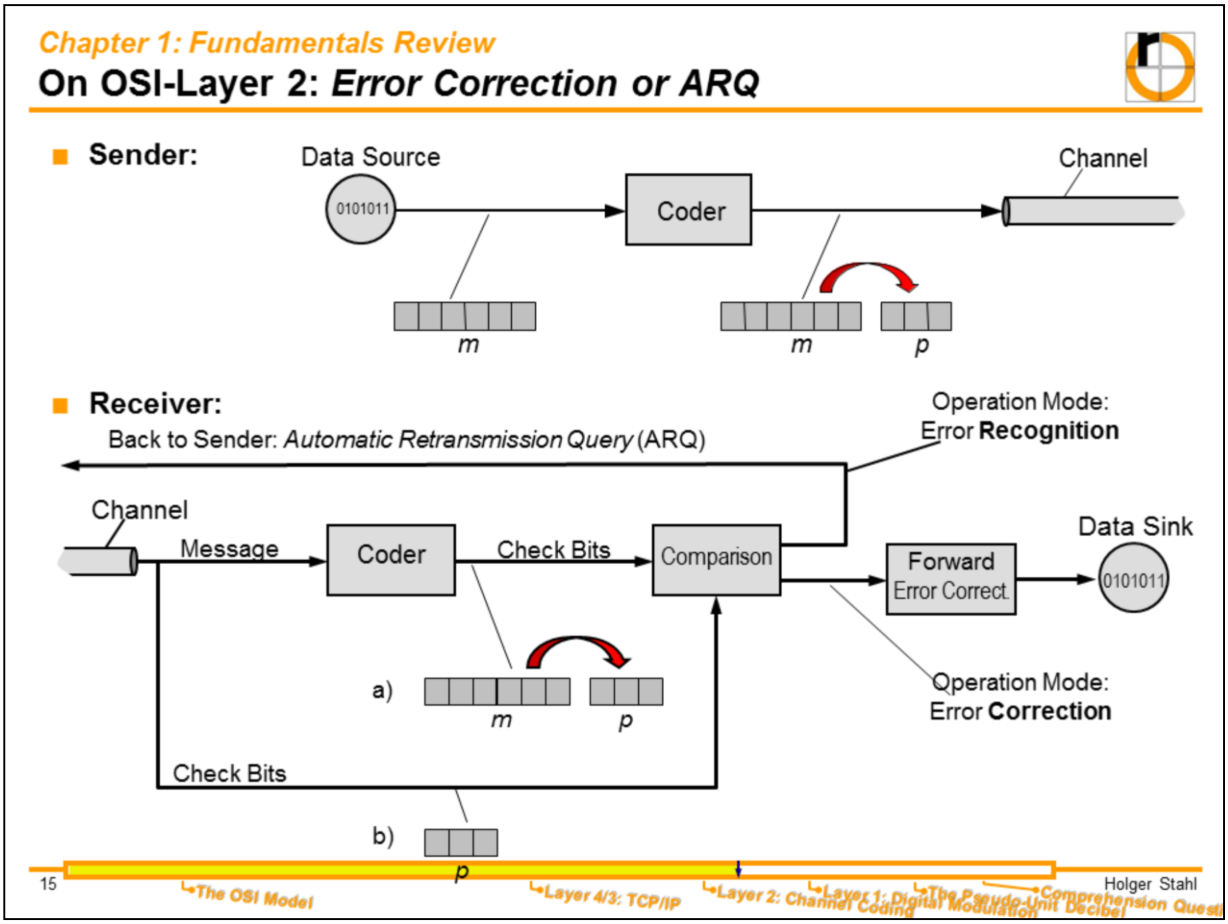
Medium	Error Rates per Bit p_E	
ISDN Basic Rate Subscriber Interface	10^{-6}	
Ethernet (Twisted Pair Cable)	10^{-8}	
Fibre Glass Channel	10^{-12}	
Wireless Channel (GMSK with 10dB SNR)	10^{-4}	
Wireless Mobile Channel with Rayleigh Fading	10^{-1}	

Possible Reasons for Errors:

- Noise & Interference:
  - Thermal Noise together with a low signal power results in a small SNR (*Signal-to-Noise Ratio*).
  - Crosstalk
  - Impulses (Thunderstorms, Processes of switching on/off devices)
- Frequency-dependent (linear) Distortion on the channel
- Nonlinear Distortion

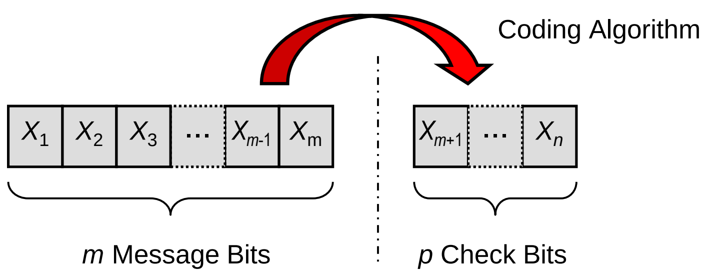
Error Recovery by adding Redundancy

For error recovery, the message is supplemented by control information (redundancy). Depending on the amount of redundancy, a certain number of errors can be detected or even corrected.



Systematic Block Coding

Assume that binary data has to be transmitted. Therefore blocks are composed from m message bits which are taken out from the continuous data stream. Each block is complemented by p *Check Bits* or *Redundancy Bits* (also called *FCS*, *Frame Check Sequence*). The message block of $n = m+p$ bits including the check bits is also called (n,m)-*Block-Code*. Codes, which transfer the original message bits over the channel are called *Systematic Codes*:



The ratio of the size if the message m and the total size n of the block is called ***Efficiency*** or *Code Rate*:

$$\boxed{r = m/n}$$

In the **Sender** (i.e. in the channel coder), p redundant check bits are calculated for each block of m message bits, which will afterwards be transmitted together with the message.

The **Receiver** (i.e. the channel decoder) compares the received and the newly generated check bits. Depending on the operation mode of the error recovery, either a retransmission of the erratic block is requested (*Error Recognition with ARQ*) or the decoder tries to correct (*Error Correction*) the errors.

Operation Modes: *Error Recognition* or *Error Correction*

As already mentioned, the error recovery can be obtained by two different strategies:

- **ARQ (*Automatic Retransmission Query*)**

  Recognition of errors and – if necessary – request of a retransmission (*ARQ*).
  - Effective reduction of remaining errors with relatively small redundancy.
  - The ARQ-scheme requires a reverse channel; this is available in case of duplex links, but not for broadcast transmissions.
  - If ARQ is used together with high error rates (e.g. mobile communication) and large blocks, which have to be repeated in case of errors, two negative effects apply:
    - ⇒ The block error rate (probability for an error within the block) becomes very high.
    - ⇒ The amount of data that has to repeated, becomes high.

- **FEC (*Forward Error Correction*)**

  Blocks that have been detected as erratic will be "corrected" by replacing them with the next correct block. It is impossible to guarantee that this process is done correctly!
  - Many check bits are necessary, needing additional transmission bandwidth
  - Powerful algorithms are necessary for the correction.

Forward Error Correction usually is only worth it in case of channels with high error rates (mobile channels), or in case of continuous, synchronous data streams, where no ARQ is possible. Examples: *Digital TV*, *Voice-over-IP*

Block Coding vs. *Convolutional Coding*

Block coding always saves a number of bits, before the check bits are calculated. It is also possible to add the redundancy continuously – bit for bit: For this so-called *Convolutional Coding*, there is no need to intermediately save a block of message bits, which reduces the signal delay time (e.g. for mobile telephony).

Example: The popular block code "CRC-32"

- *Cyclic Redundancy Check* with $p = 32$ check bits
- Recognition performance:
  - More than $m = 4$ billion message bits can be protected by only $p = 32$ check bits.
  - Secure recognition of up to 2 errors
  - Most important: Recognition of error bundles of $l \leq 32$ bit
- Used for error recognition in many standards, e.g.:
  - *Ethernet*
  - *zip*-File Compression
  - *MPEG-2* (Transport Stream Format used for *DVB-T*, *mp3*), etc.

On OSI-Layer 1: *Analogue and Digital Modulation Schemes*

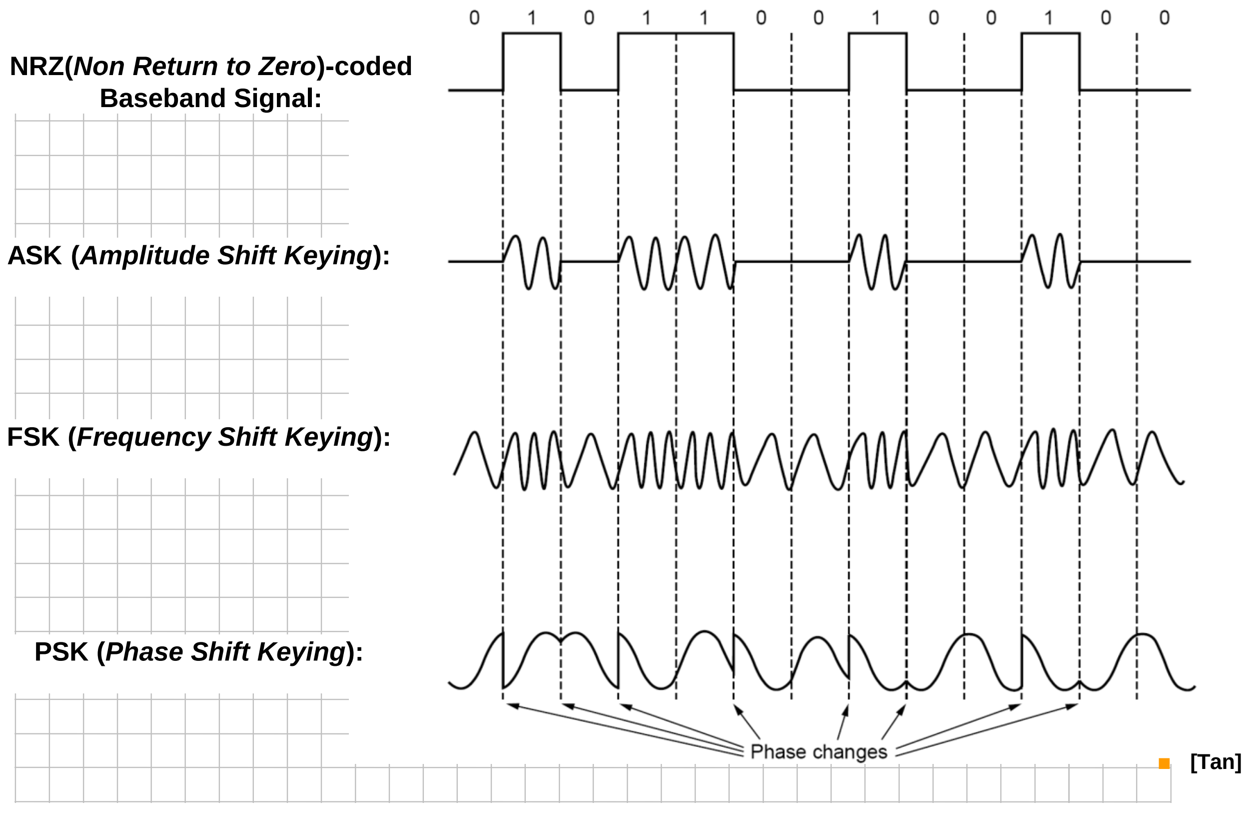
$$s(t) = \hat{U} \cdot \sin(2\pi \cdot f_0 t + \varphi)$$

<u>Modification of the Amplitude:</u>
Analog: *Amplitude Modulation* (AM)
Digital: *Amplitude Shift Keying* (ASK)

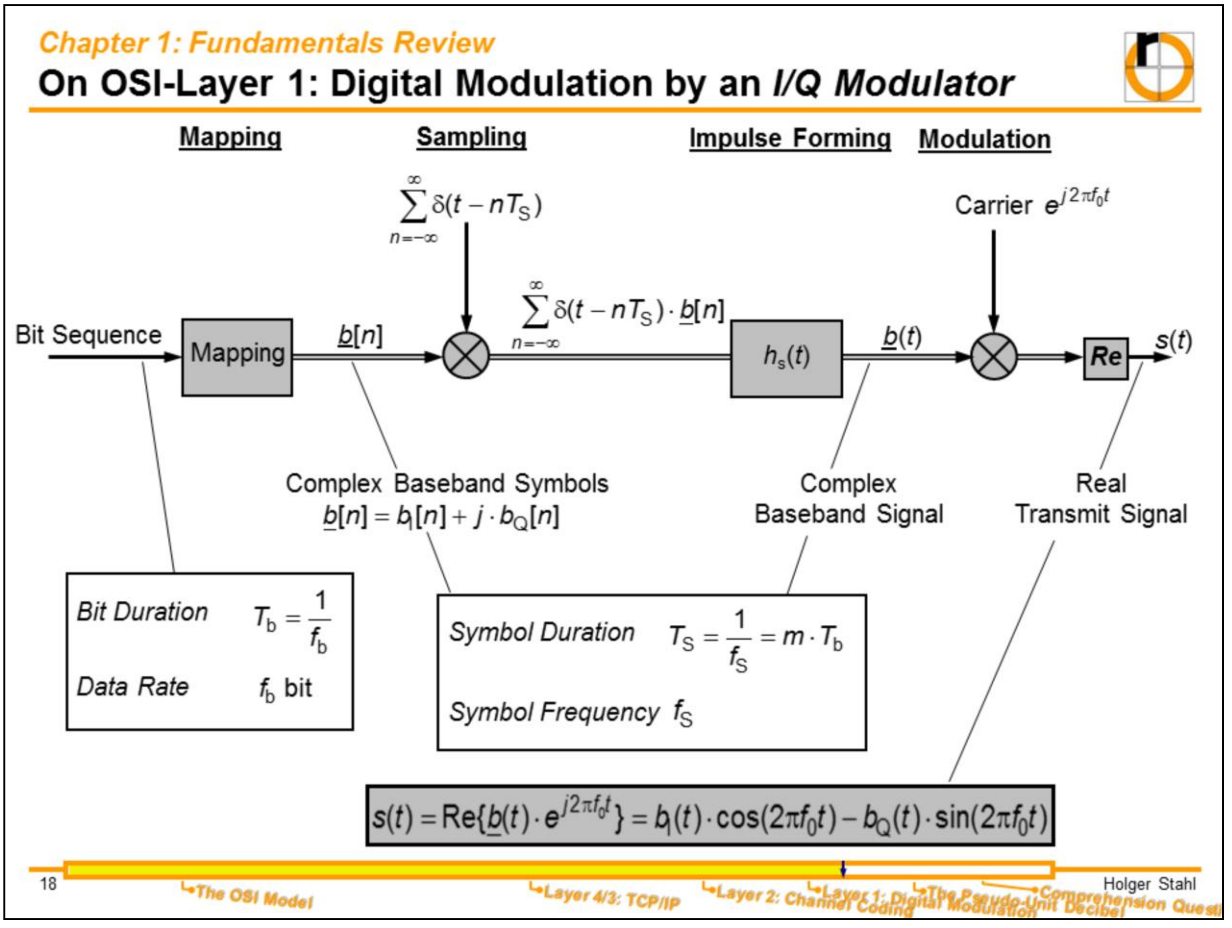
<u>Modification of the Phase:</u>
Analog: *Phase Modulation* (PM)
Digital: *Phase Shift Keying* (PSK)

<u>Modification of the (current) Frequency:</u>
Analog: *Frequency Modulation* (FM)
Digital: *Frequency Shift Keying* (FSK)

<u>Example Signals for Elementary Digital Modulation Schemes:</u>



[Tan]



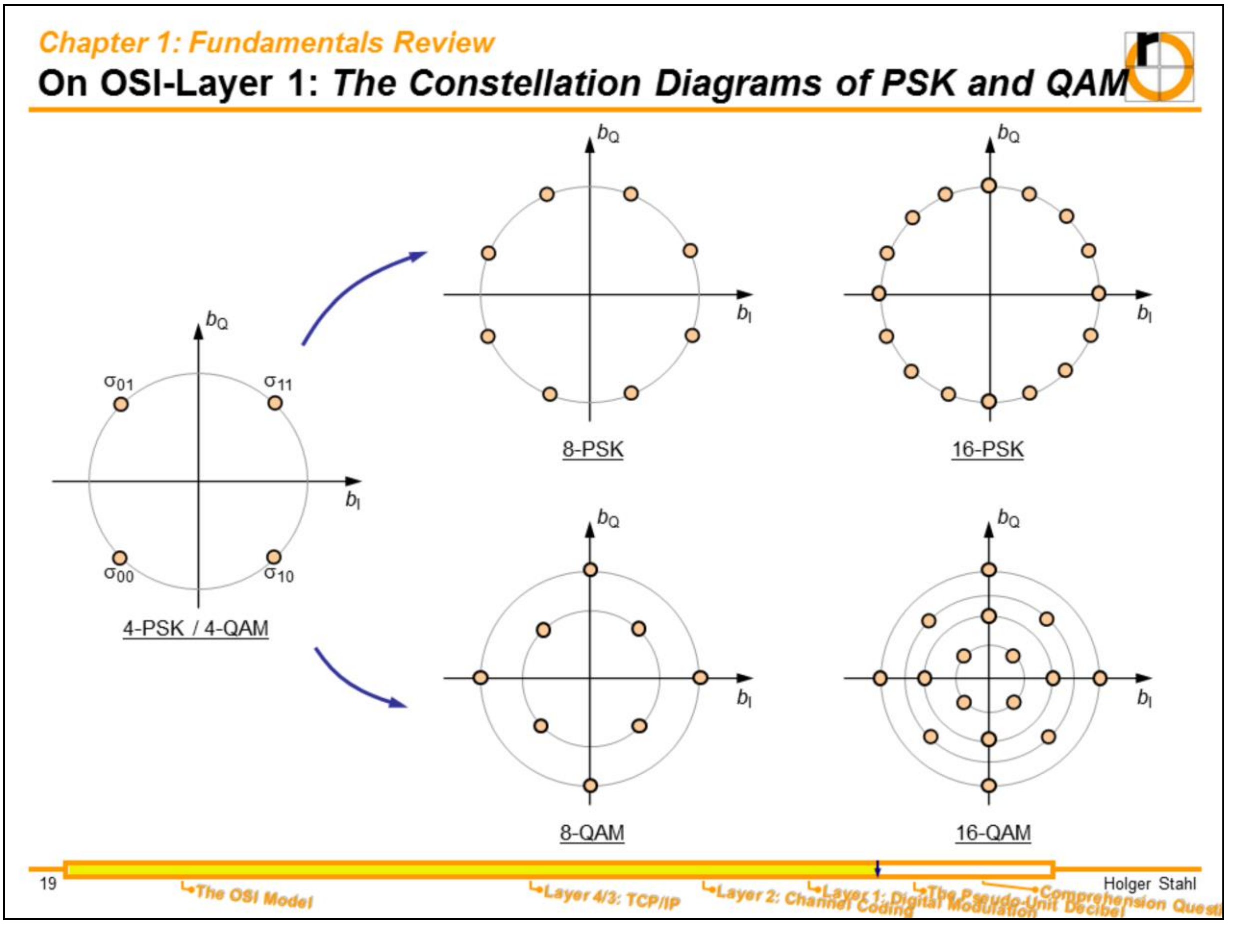
From the elementary modulation schemes elucidated on the previous slide, a big variety of combined and further developed digital modulation principles exist. In the following, an I/Q modulator will be observed, which is able to perform *Quadrature Amplitude Modulation* (QAM), which stands for a combination of ASK and PSK (extended to a possibly higher number of amplitude and phase values).

How does an I/Q modulator work?

Two baseband signals, a real (in-phase) one and an imaginary (quadrature-phase) one, are modulated simultaneously*) onto a sine and a cosine carrier. Since sine and cosine are orthogonal to each other, the two signals can be transmitted together without any interference on their way to the receiver.

---

*) In the block diagram above, we use one complex modulation (carrier $e^{j2\pi f_0 t}$) instead of the separate sine and cosine carrier.

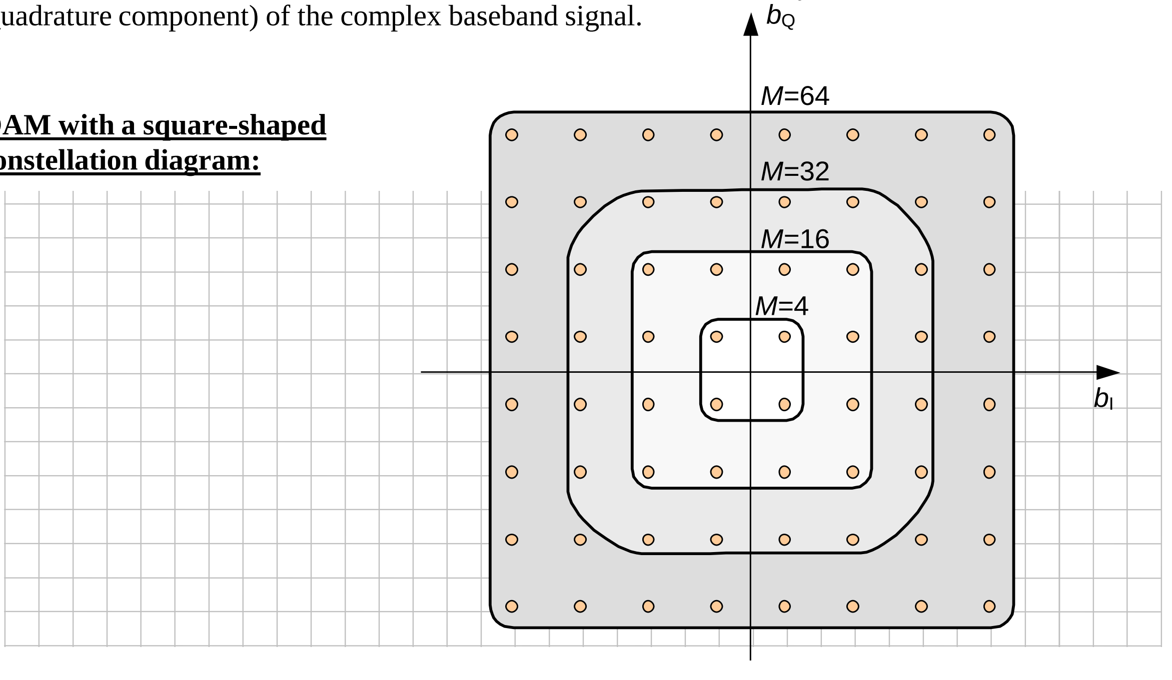


<u>Constellation Diagram</u>

Depending on the *Mapping* function, a combination of *m* bits is coded by one complex baseband symbol. Hence, each of the $M = 2^m$ symbol values represents one carrier state, a combination of amplitude and phase. The figure above shows all possible carrier states in a so-called *Constellation Diagram*.

The symbol values appear as real and imaginary parts (b_I and b_Q, I for the Inphase-, Q for the Quadrature component) of the complex baseband signal.

<u>QAM with a square-shaped constellation diagram:</u>



The Pseudo-Unit *Decibel*

- **Reason for introducing the unit *Decibel* (‘dB’)**
  - Wireless Communication engineers often work with extremely high or extremely small power values, amplitudes, and amplification factors.
    - ⇒ Instead of juggling these extremely large/small numbers, we use their logarithm.
    - ⇒ Multiplications can be replaced by simple additions.

- **Expressing Relations (e.g. Cable Attenuation, *Signal-to-Noise* Ratio):**

Power Ratio in ‘dB’

$$\boxed{\frac{P_2}{P_1} = 10 \cdot \log\left(\frac{P_2}{P_1}\right) \mathrm{dB}}$$

Amplitude (e.g. Voltage) Ratio in ‘dB’

$$\frac{U_2}{U_1} = 20 \cdot \log\left(\frac{U_2}{U_1}\right) \mathrm{dB}$$

- **Expressing absolute values for Power or Amplitude [1)]**

Power in ‘dBm’

$$\boxed{P = 10 \cdot \log\left(\frac{P}{\mathbf{mW}}\right) \mathbf{dBm}}$$

Amplitude in ‘dBμV’

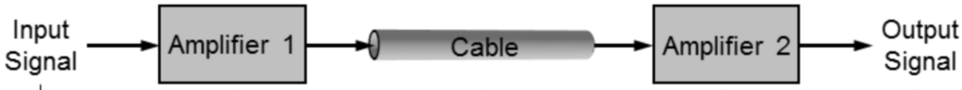
$$U = 20 \cdot \log\left(\frac{U}{\mathbf{\mu V}}\right) \mathbf{dB\mu V}$$

[1)] Note: The rules given in DIN/IEC/ISO recommend, not to combine the Pseudo-Unit ‘dB’ with physical units (like ‘mW’), because they easily can be confused with a product of both ‘dB · mW’. According to these rules, an absolute power ***P* = 20 dBm** has to be written as ***L*$_P$(re 1 mW) = 20 dB** .

However, to keep the formulas compact in this lecture, we will interpret the value together with the pseudo-unit ‘dB’ as a pure factor, which then (after de-logarithmization) can be multiplied with the unit. This way of thinking is clarified in the following by putting braces around the ‘20 dB’:

$$P = 100\,\mathrm{mW} = (20\,\mathrm{dB}) \cdot \mathrm{mW} = 20\,\mathrm{dBm}$$

Example for using the ‘dB’ calculation scheme



Input Signal	Amplifier 1	Cable	Amplifier 2	Output Signal
P_1	a	b	c	$P_2 = P_1 \cdot a \cdot b \cdot c$
$P_1 = 10\log\left(\frac{P_1}{\text{mW}}\right)\text{dBm}$	$10\log(a)\,\text{dB}$	$10\log(b)\,\text{dB}$	$10\log(c)\,\text{dB}$	

$$P_2 = \left[10\log\left(\frac{P_1}{\text{mW}}\right) + 10\log a + 10\log b + 10\log c\right]\text{dBm}$$

A. The *OSI Reference Model for Communication*

1. General Questions about the *OSI Reference Model for Communication*:
   a) Which OSI-Layers are defined by a LAN (*Local Area Network*)?
   b) Which protocol layers are affected by a *Transit System*?
   c) Which protocol layers implement *End-to-End* functions?
   d) What for are SAPs (*Service Access Points*) in the OSI Reference Model?
   e) Which Layers of the OSI Reference Model might implement a *Flow Control* Mechanism?

2. In the following you find some types of user data. Specify for each of these cases, if it would be better to transport them by a packet-switched or by a circuit-switched network:
   a) Telephony
   b) Live-Video
   c) a Facsimile
   d) an eMail
   e) "Surfing" in the World-Wide-Web

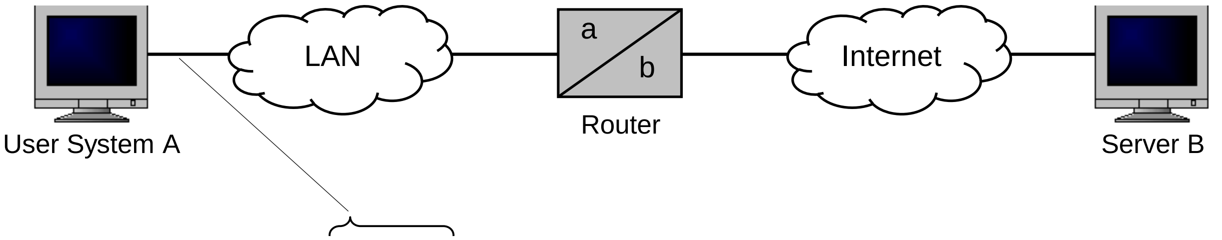
3. List one protocol standard for each of the following (combination of) OSI Layers:
   a) Physical
   b) Physical & Data Link
   c) Network
   d) Transport
   e) Application

4. Addresses in the *OSI Reference Model*
   a) What is identified by a MAC address? What is the scope of this address?
   b) What is identified by a Layer-3 address? What is the scope of this address?
   c) What is identified by a Layer-4 address? What is the scope of this address?
   d) Give a practical example for a Layer-3 address.
   e) Give a practical example for a Layer-4 address.

B. *TCP/IP Basics*

1. General Questions:
   a) List two services that are offered by IP (*Internet Protocol*).
   b) Why do VoIP systems usually not use the TCP?
   c) Why is the MAC address not sufficient for world-wide addressing of a host computer?
   d) Which protocol instance on which layer is addressed by the TCP port number?

2. A Web browser on computer system **A** sends a request to a HTTP-Server **B** via TCP/IP:



Look at the details of the data frame leaving the sender:

a) Which destination MAC address will be chosen by the sender (Circle the right answer):
…of the router interface **a**, …of the router interface **b**, …or of the destination server **B** ?

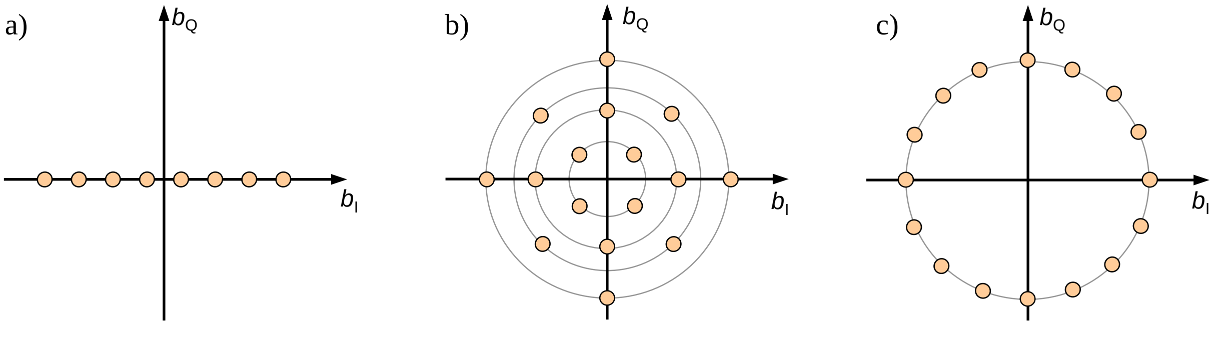
b) Which destination IP address will be chosen by the sender (Circle the right answer):
…of the router interface **a**, …of the router interface **b**, …or of the destination server **B** ?

c) Specify the exact destination TCP port number.

C. OSI-Layer 1: *Digital Modulation Basics*

1. General questions concerning digital modulation:
   a) Briefly explain, what is a base band signal!
   b) What is the advantage of PSK (*Phase Shift Keying*) compared to QAM (*Quadrature Amplitude Modulation*)?
   c) What is the advantage of QAM compared to PSK?
   d) Sketch the constellation diagram for 8-PSK.
   e) Sketch the constellation diagram for 16-QAM.
   f) Why does the modulator need an impulse forming filter?

2. The following three diagrams show the constellation diagrams of an I/Q modulator:



Specify for each of the three cases, …

i. how many bits are coded per symbol, and…

ii. whether *PSK* (*Phase Shift Keying*), pure *ASK* (*Amplitude Shift Keying*) or *QAM* (combined *ASK* and *PSK*) was used.

D. OSI-Layer 2: *Block Coding and Error Correction*

1. Error Probabilities:

   A frame of n = 1000 bit is received with a BER (*Bit Error Rate*) of $p_E = 10^{-5}$.

   a) Calculate the probability $p_E(1000)$ of the frame being received with one or more errors.

   Now the frame is protected with a single (i.e. p = 1) parity bit:

   b) Estimate the remaining error probability $p_{RE}(1001)$ for error recognition, i.e. the probability of an erratic frame, which is not recognized.

2. Miscellaneous questions about Channel Coding:

   a) List one advantage and one disadvantage of ARQ (*Automatic Retransmission Query*) compared to FEC (*Forward Error Correction*).

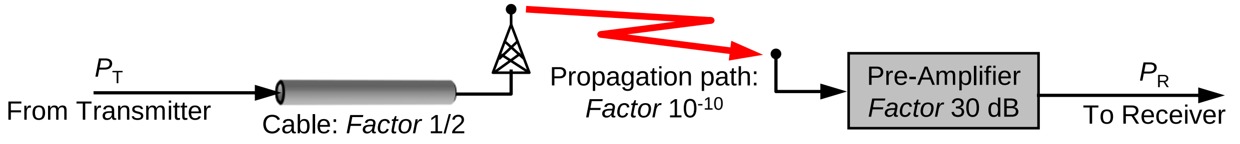
   b) List two standards which employ “CRC-32” for protecting data against errors.

   c) How many check bits are used in “CRC-32” ?

   d) How many message bits can be protected against errors by “CRC-32” ?

E. The Pseudo-Unit *Decibel*

1. A radio transmitter produces an output signal with the power P_T = 10 W. Before the signal reaches the input of the receiver, it suffers from cable attenuation, propagation path loss, and a final amplification with the given power amplification factors:



   Calculate the received power P_R in the unit ‘dBm’!

2. The input of a receiver has an impedance of 50 Ω. The signal power is P_R = 1 μW.

   a) Convert the signal power in the unit ‘dBm’.

   b) Calculate the signal amplitude in the unit ‘dBμV’.