Karlsruhe Institute of Technology

Institute for Information
Processing Technologies

Introduction

Communication Systems and Protocols – Session 1

Who we are

- Lecturer
  - Dr.-Ing. Jens Becker
    - Email: csp@itiv.kit.edu
    - Room: 322
    - Consultation hours: by arrangement

- Advisor Exercises
  - M. Sc. Nidhi Anantharajaiah
    - csp@itiv.kit.edu
    - Room: 226.2
    - Consultation hours: by arrangement

Ilias Learning platform

- Address: https://ilias.studium.kit.edu/ilias.php?ref_id=1435350&cmdClass=ilrepositorygui&cmdNode=uk&baseClass=ilRepositoryGUI

- No Password at the moment, later “CSP_SS2021”

- Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials.

- The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable.

- If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums.

Ilias Learning platform

- Contents:
  - Video Lecture and Video Tutorials
  - Lecture and Exercise slides
  - Forum
  - Student surveys
  - Taskbook
  - Old examinations
  - Calendar
  - General announcements

- Please visit regularly

Disseminating Recordings?

Quote from the Mission Statement of KIT (KIT 2025 Umbrella Strategy)

“Our working together and our management culture are characterized by respect, cooperation, confidence, and subsidiarity.”

- Please observe the personal rights of your teachers and show your respect by not recording the lectures and not disseminating the recordings.
- Any recording requires the express approval by KIT. This approval must be obtained in writing via the teacher responsible for the lecture/course.

Schedule CSP

Monday: 10:00 – 11:30 (online)

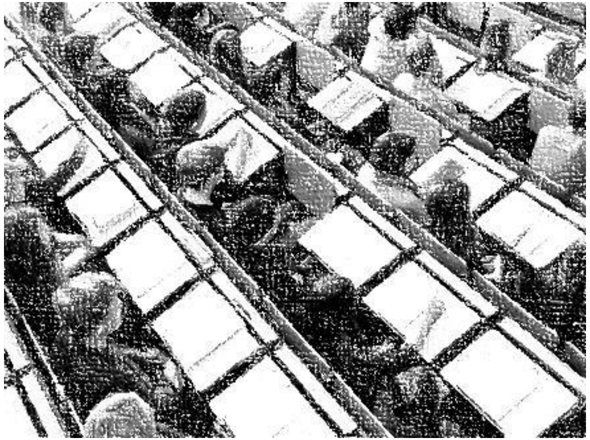
- 12.04.2021 – Lecture 1
- 19.04.2021 – Lecture 2
- 26.04.2021 – Lecture 3
- 03.05.2021 – Lecture 4
- 10.05.2021 – Lecture 5
- 17.05.2021 – Lecture 6
- 24.05.2021 – Holiday
- 31.05.2021 – Lecture 7
- 07.06.2021 – Lecture 8
- 14.06.2021 – Lecture 9
- 21.06.2021 – Lecture 10
- 28.06.2021 – Lecture 11
- 05.07.2021 – Exercise 6
- 12.07.2021 – Lecture 12
- 19.07.2021 –

Thursday: 10:00 – 11:30 (online)

- 15.04.2021 –
- 22.04.2021 – Exercise 1
- 29.04.2021 –
- 06.05.2021 – Exercise 2
- 13.05.2021 – Ascension Day
- 20.05.2021 – Exercise 3
- 27.05.2021 – Holiday
- 03.06.2021 – Corpus Christi
- 10.06.2021 – Exercise 4
- 17.06.2021 –
- 24.06.2021 – Exercise 5
- 01.07.2021 –
- 08.07.2021 – Exercise 7
- 15.07.2021 – Exercise 8
- 22.07.2021 – Question time

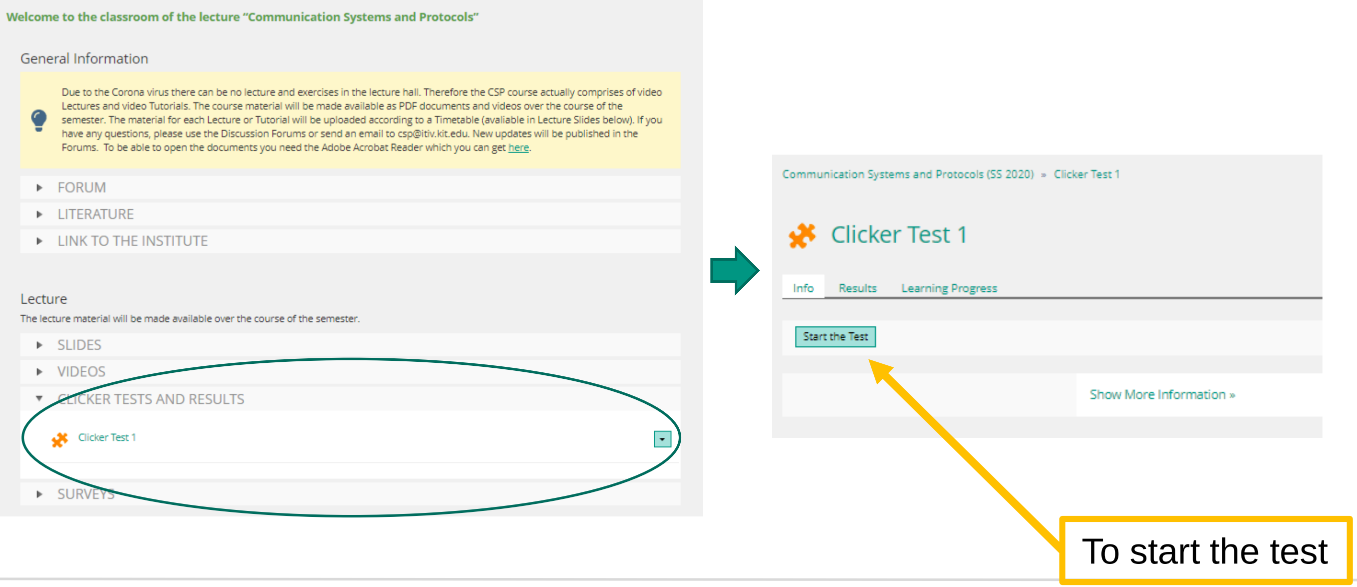
Written Examination

- Date: Monday, 26.07.2021 (Can change)
- Time: TBA
- Place: TBA
- No examination aids are permitted, except for:
  - A ruler
  - A non-programmable calculator
  - a single sheet of A4 paper with self- and hand-written notes, writing may be on both sides OR two sheets of A4 paper with writing on one side each.
  - A dictionary
- The language of the examination is English



Clicker Tests

- We want you to become active!
- Participate in Clicker Tests in ILIAS
- Information on when to take Clicker Tests will be included in the slides
- Anonymous Test



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

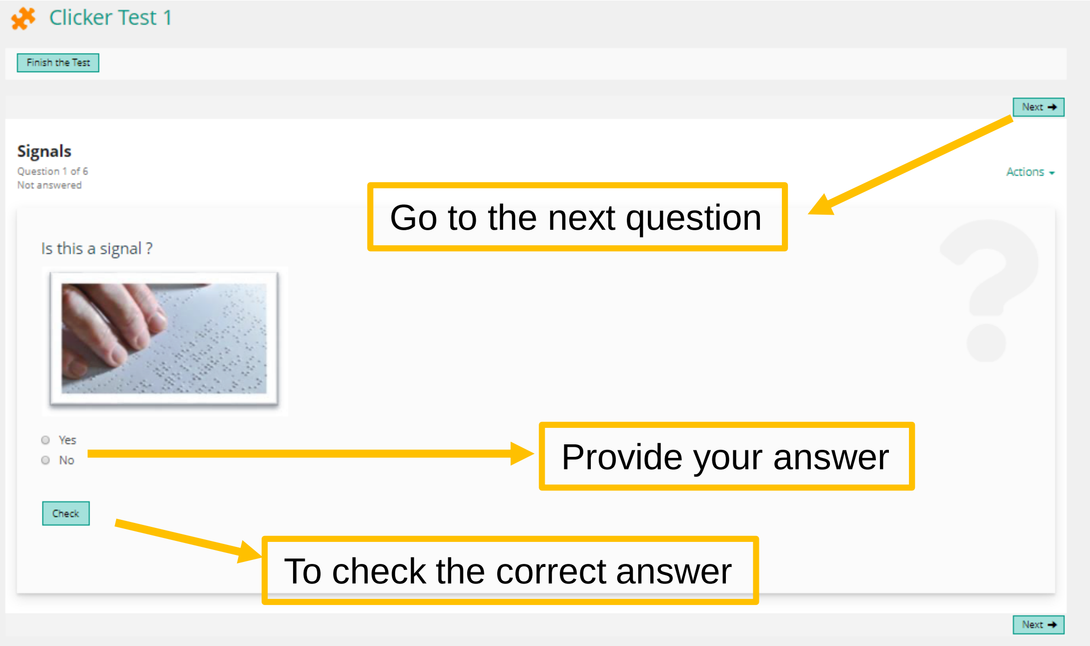
Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (available in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

To start the test

Clicker Tests : Guidelines



Signals

Question 1 of 6

Not answered

Is this a signal ?

○ Yes

○ No

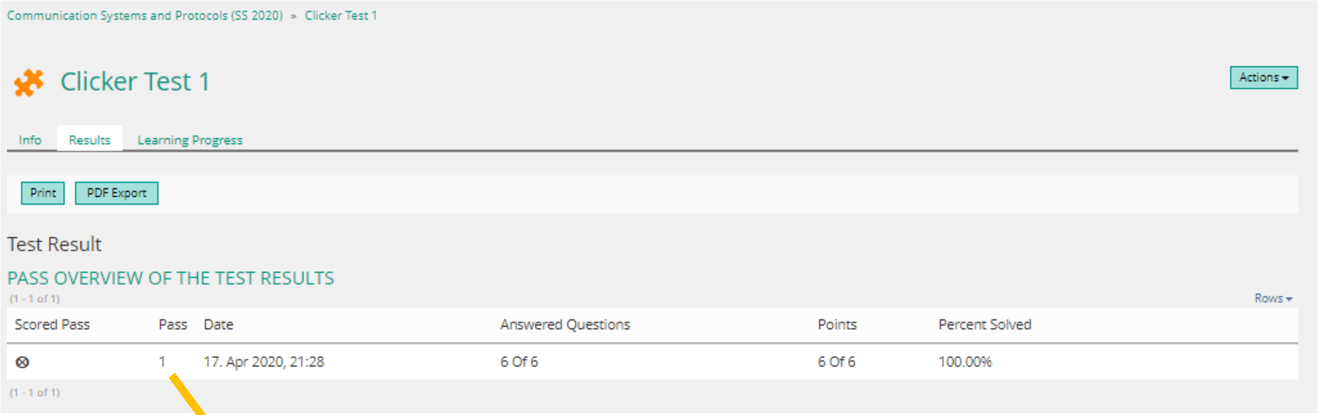
Go to the next question

Provide your answer

To check the correct answer

Clicker Tests : Results Overview

- Each clicker test has around 5-6 questions
- Each having one point
- Anonymous Test



Communication Systems and Protocols (SS 2020) » Clicker Test 1

Clicker Test 1

Test Result

PASS OVERVIEW OF THE TEST RESULTS

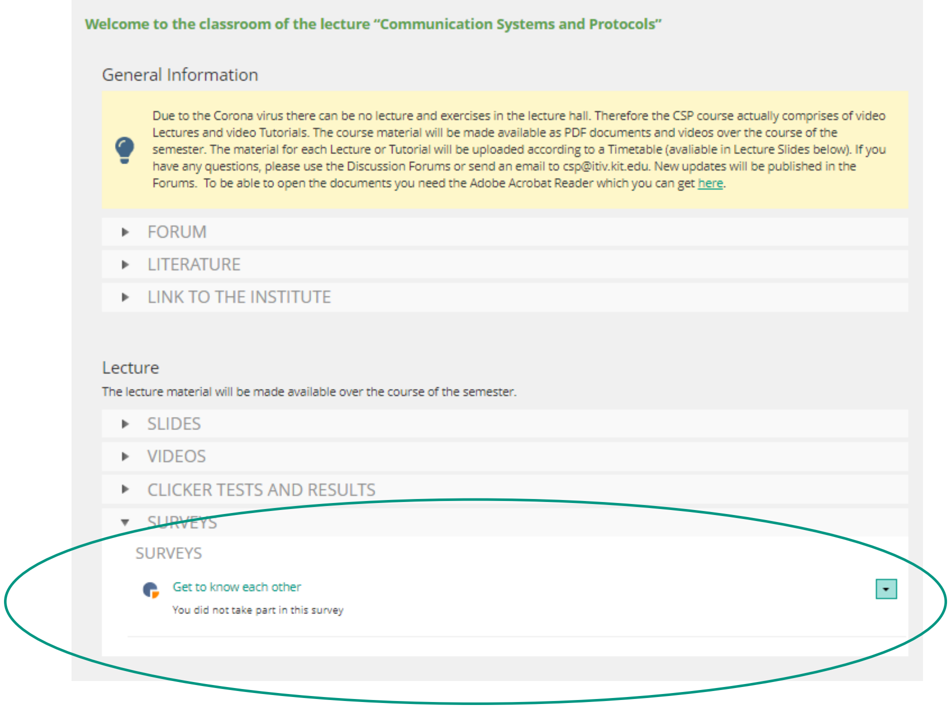
(1 - 1 of 1)

Scored Pass	Pass	Date	Answered Questions	Points	Percent Solved
⊗	1	17. Apr 2020, 21:28	6 Of 6	6 Of 6	100.00%

(1 - 1 of 1)

Number of Attempts

Get to know each other Survey



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (available in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

- ▸ FORUM
- ▸ LITERATURE
- ▸ LINK TO THE INSTITUTE

Lecture

The lecture material will be made available over the course of the semester.

- ▸ SLIDES
- ▸ VIDEOS
- ▸ CLICKER TESTS AND RESULTS
- ▾ SURVEYS

SURVEYS

Get to know each other

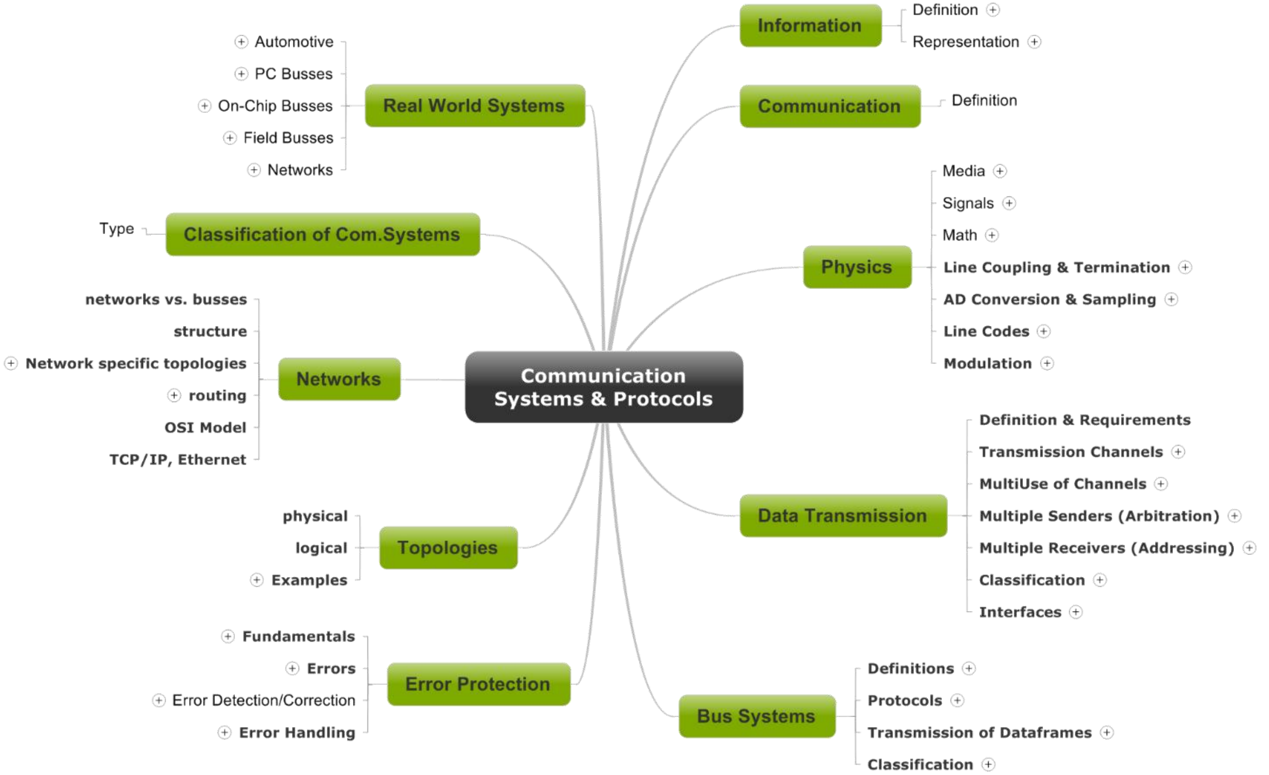
You did not take part in this survey

Prerequisites

- Digitaltechnik
  - Signals
  - Information Theory
  - Error Protection
- Lineare Elektrische Netze
  - Quadripole
  - Transmission Line Theory
- Nachrichtentechnik/Nachrichtenübertragung
  - Channel Capacity

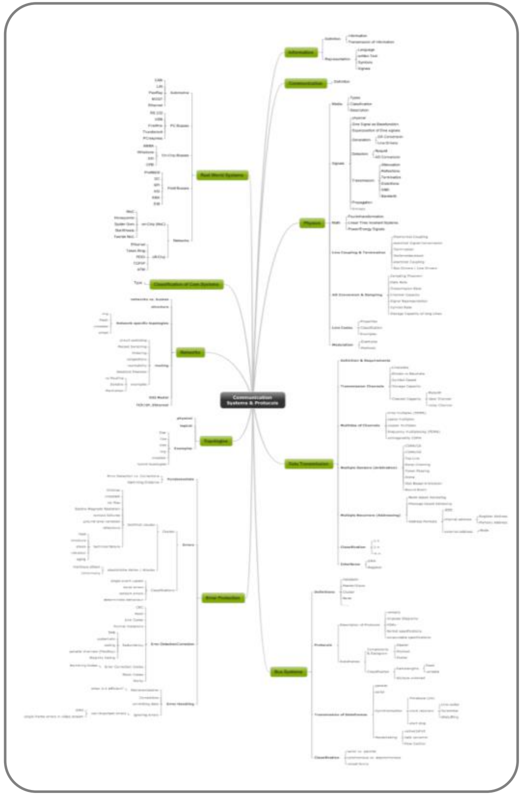
- Focus of Communication Systems and Protocols
  - Communication Architectures
  - Data Transmission (Formats, Protocols, etc.)
  - Embedded busses and related communication entities

Parts of Communication Systems



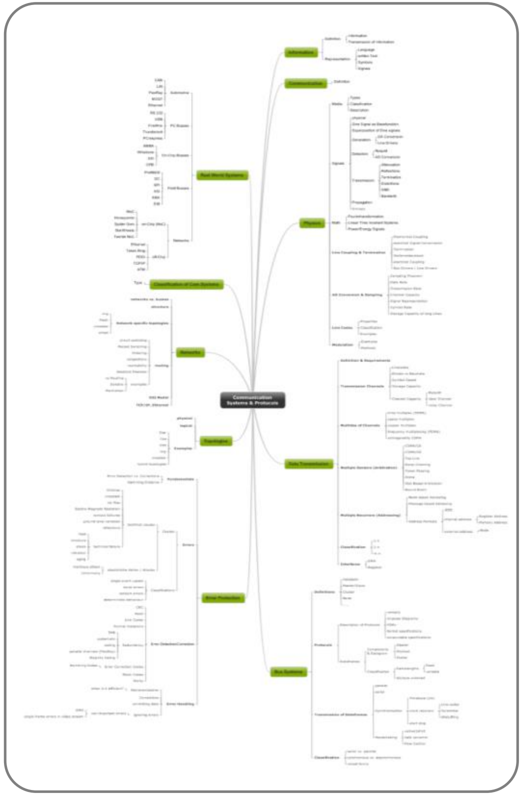
Parts of Communication Systems

- Communication Systems comprises many fields
- We focus on dedicated aspects



After this Lecture you should…

- …know basic communication systems and be able to name them.
- …be able to categorize different communication systems in regards to possible constraints.
- …be able to name basic mechanisms of communication systems.
- … be able to carry out these mechanisms.
- … be able to choose valid methods suitable under given constraints.
- … be able to design a communication system adhering to constraints, specifications and be able to choose suitable methods, components, and subsystems.



Calculation of work for course

- Credit points for CSP: 5
  - Each credit point corresponds to 25-30h of work -> 125h – 150h
  - 20 lectures/exercises -> 30h plus 2h of examination
  - Preparing for the examination: 25h
  - 13 weeks of lecture -> **5-7h per week for preparation**

Literature

- Behrouz A. Forouzan, Data Communications and Networking, McGraw-Hill, 2013, ISBN: 978-0-07-337622-6
- Schürmann, B.: Grundlagen der Rechnerkommunikation, Vieweg Verlag 2004, ISBN 3-528-15562-0
- Wittgruber, F.: Digitale Schnittstellen und Bussysteme, 2. Auflage, Vieweg Verlag 2002, ISBN 3-528-17436-6
- Zimmermann, W.: Bussysteme in der Fahrzeugtechnik, 1. Auflage, Vieweg Verlag 2006, ISBN-13: 978-3834801661
- Schnell, G.: Bussysteme in der Automatisierungs- und Prozesstechnik, 7. Auflage, Vieweg Verlag 2008, ISBN-13: 978-3834804259
- Wörn, H.: Echtzeitsysteme: Grundlagen, Funktionsweisen, Anwendungen, 1. Auflage, Springer Verlag 2009, ISBN-13: 978-3540205883
- Stallings, W.: Data and Computer Communications, 8 edition, Prentive Hall 2006, ISBN-13: 978-0132433105
- Tanenbaum A.: Computer Networks, 2nd edition, Edison Wesley Longman 2004, ISBN-13: 978-0131629585
- Halsall, F.: Data Communications, Computer Networks and Open Systems, 4th edition, Addison Wesley 1996, ISBN-13: 978-0201422931
- Mahalik, N.: Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control, Springer 2010, ISBN-13: 978-3642072840
- Dietmar, D.: Fieldbus systems and their applications, 1st edition, Elsevier Science 2004, ISBN-13: 978-0080442471
- Anderson, D.: FireWire system architecture, 2nd edition, Edison Wesley Professional 1998, ISBN-13: 978-0201485356

Thank you for your attention

Prof. Dr.-Ing. Dr. h. c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Introduction to Communication Systems

Contents

- Examples for communciation systems
- Topics covered in the lecture
- Layered model of a communication system
  - OSI reference model

Why Communication Systems?

People

- Communication is the Backbone of today’s Societies
- Interaction keeps Things running
- More and more specialized entities / work distribution

Technology

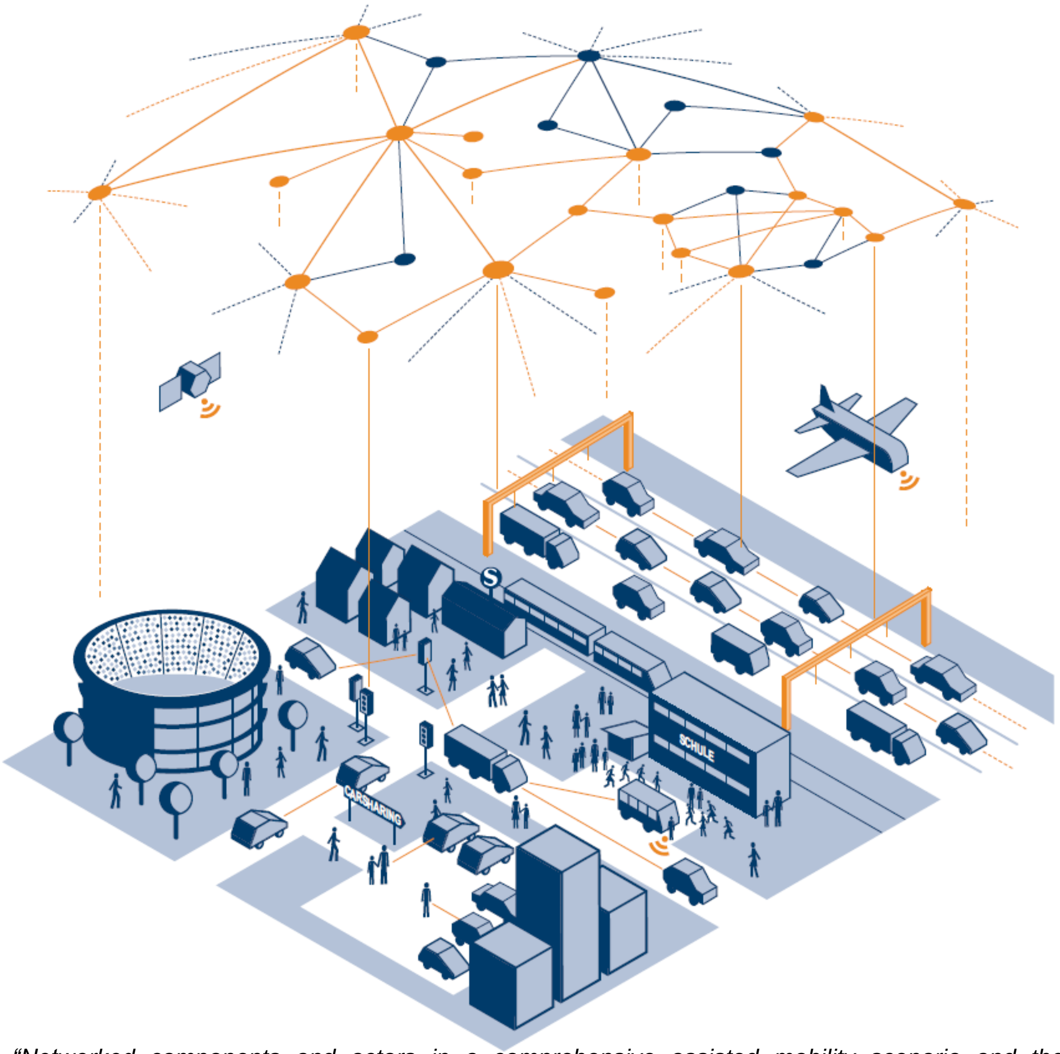
- Same concepts apply
- specialized machines interact with each other
- Networked Devices
- Internet of Things

Scenarios & Examples

- Smart traffic
- Ambient Assisted Living

Smart Traffic

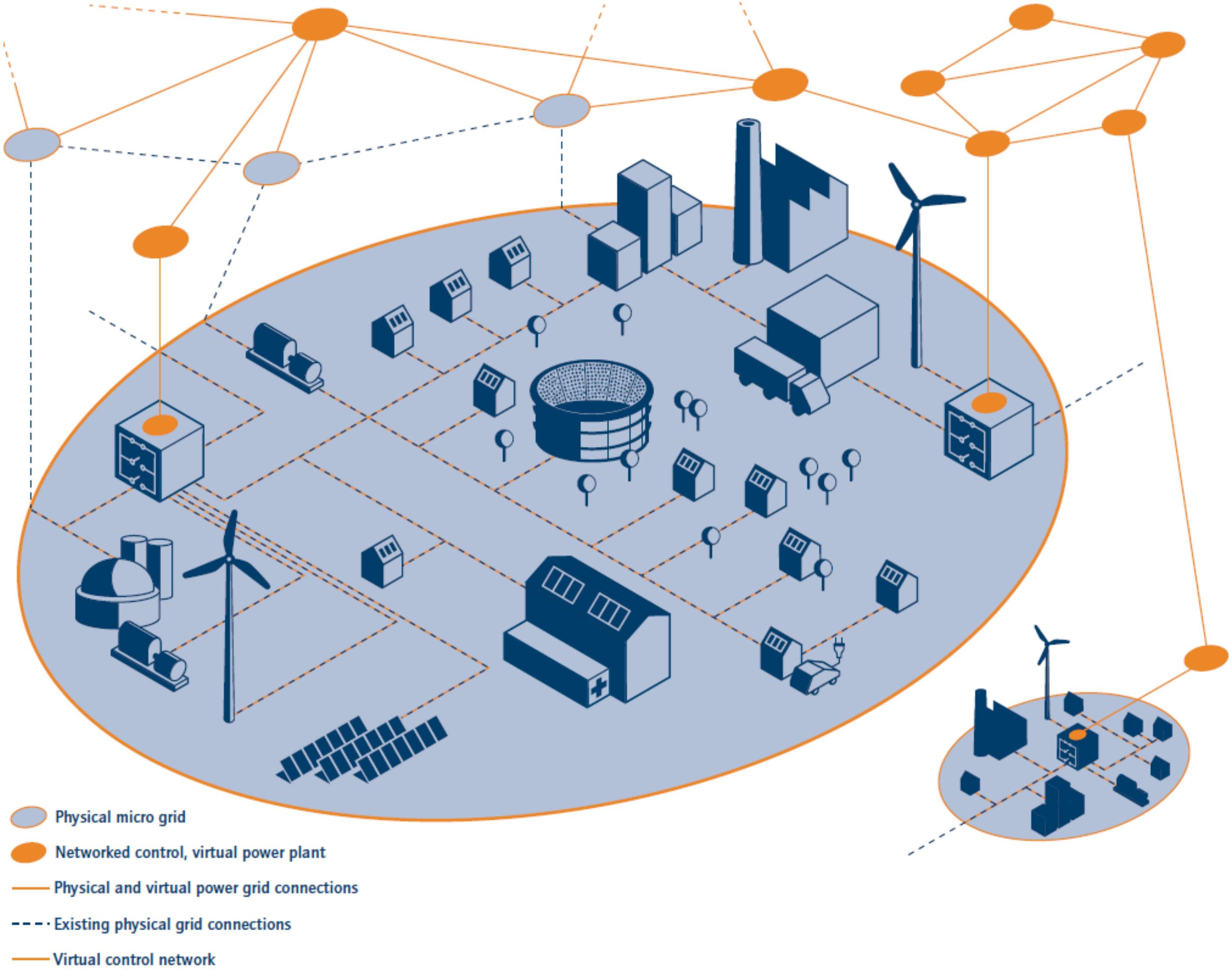
- Car2X Communication
- Traffic Flow Control
- Smart Cities
  - Public Transport
  - Individual Transport



"Networked components and actors in a comprehensive assisted mobility scenario and their situation-specific coordination, symbolized by direct connections between actors and with globally networked CPS services" , CPS : Cyber-Physical Systems

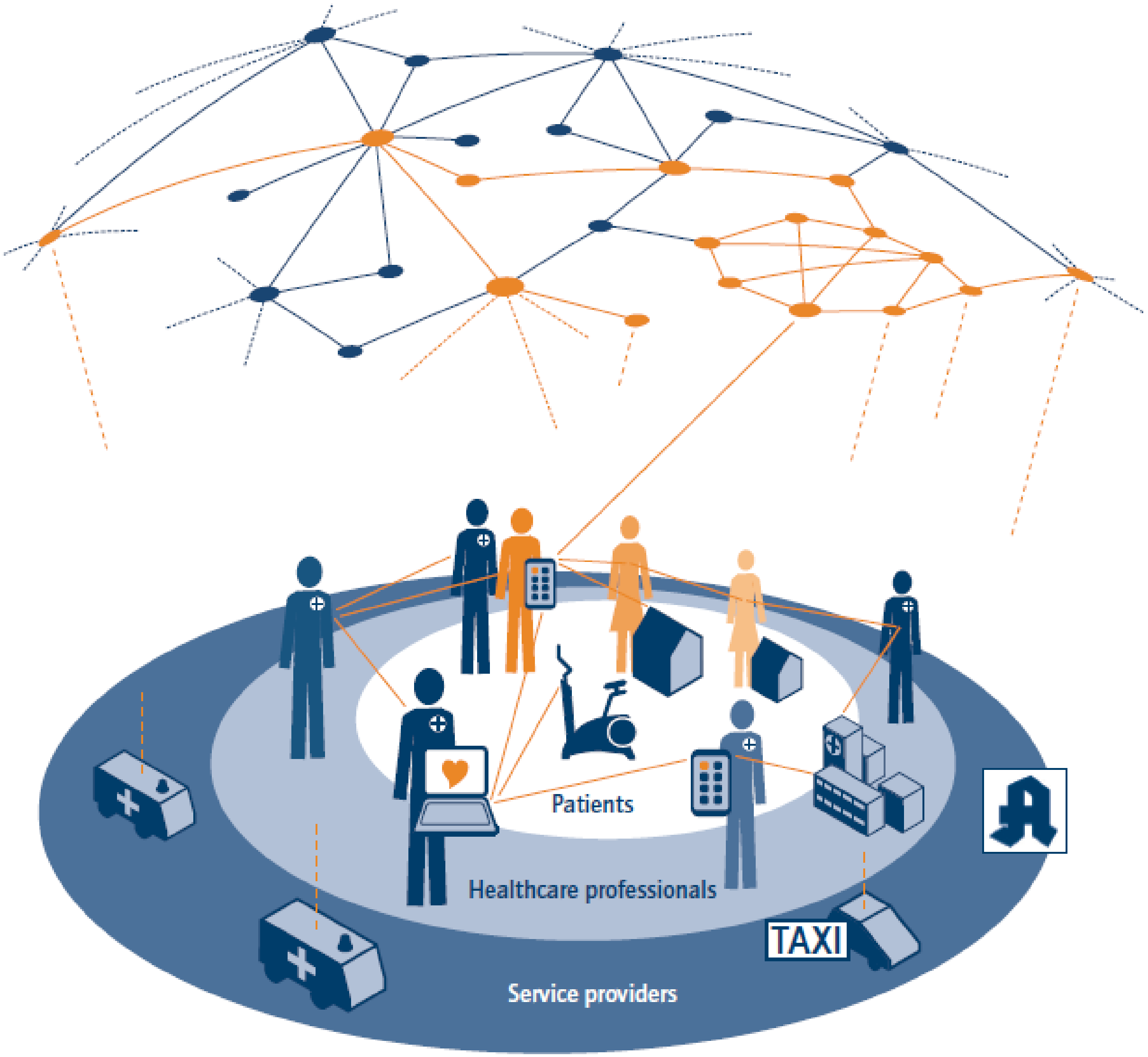
Source : Eva Geisberger/Manfred Broy (Eds.) ,"Living in a networked world", Integrated research agenda Cyber-Physical Systems (agendaCPS), acatech STUDY, March 2015

Power Grids


Physical micro grid
Networked control, virtual power plant
Physical and virtual power grid connections
Existing physical grid connections
Virtual control network

"Schematic representation of a micro grid in the networked energy supply scenario (smart grid)", Source: acatech Studie 2015

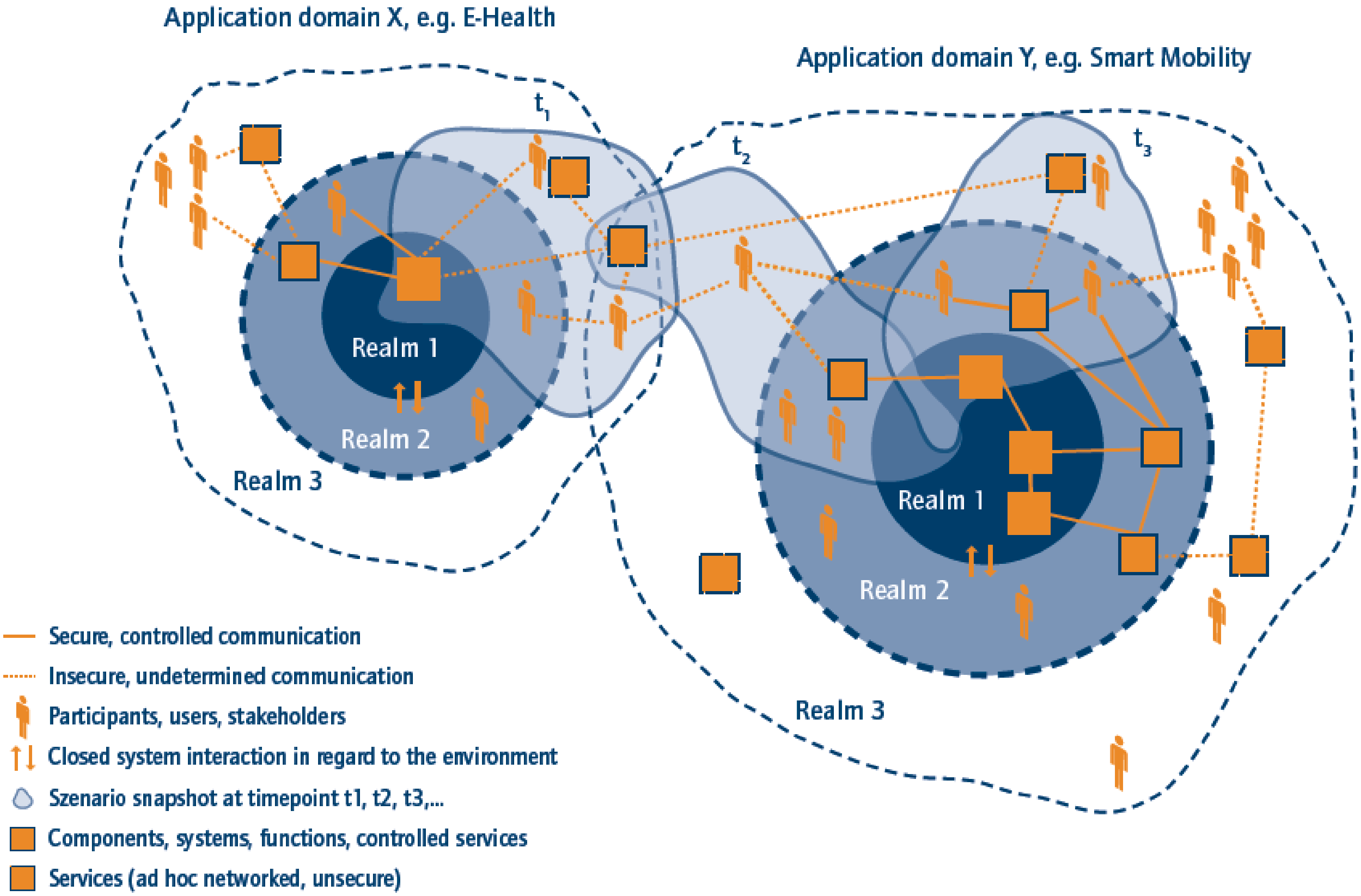
Health Care



“Networked actors in an integrated telemedicine healthcare scenario”

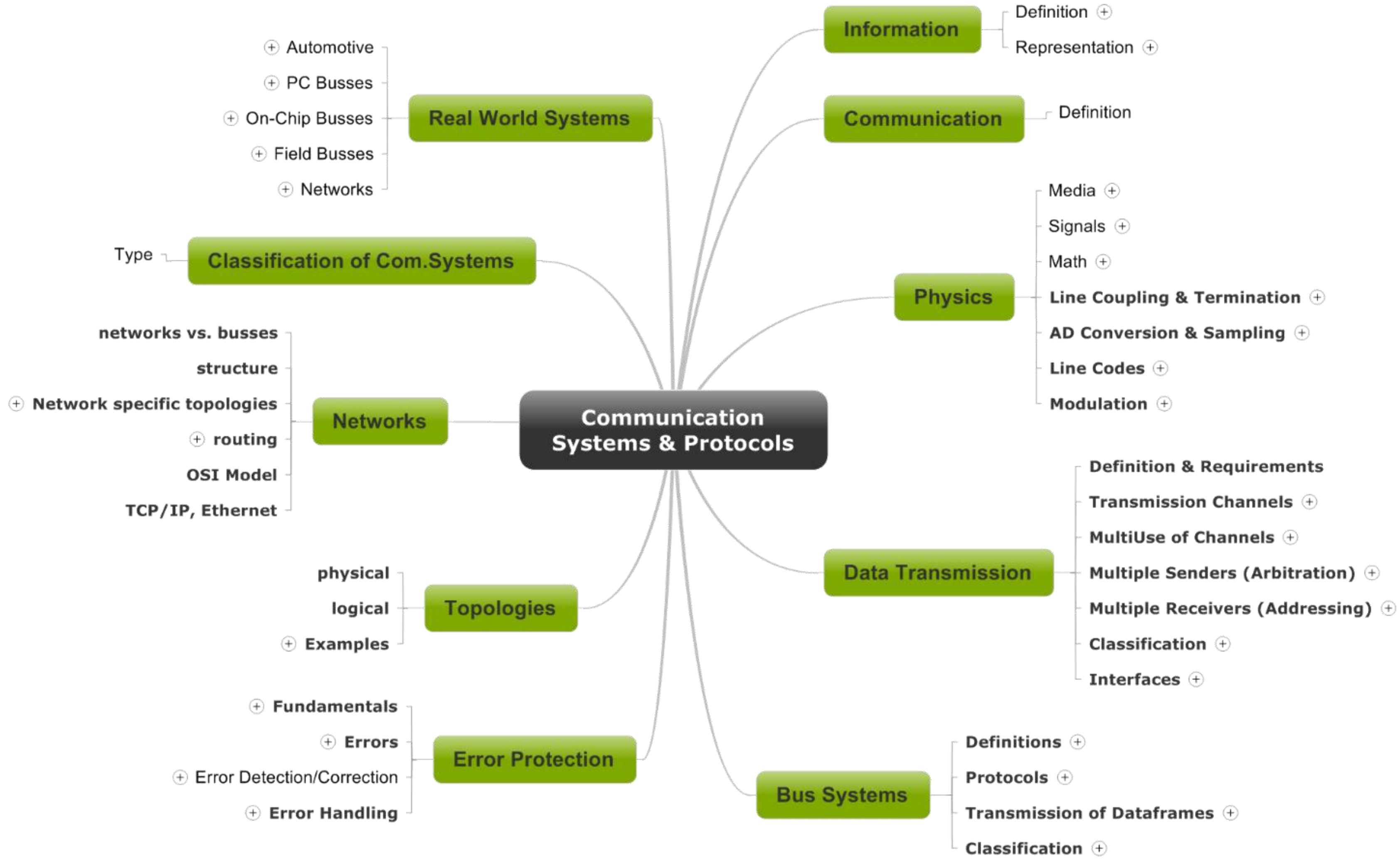
Source: acatech Studie 2015

Cross Domain Communication

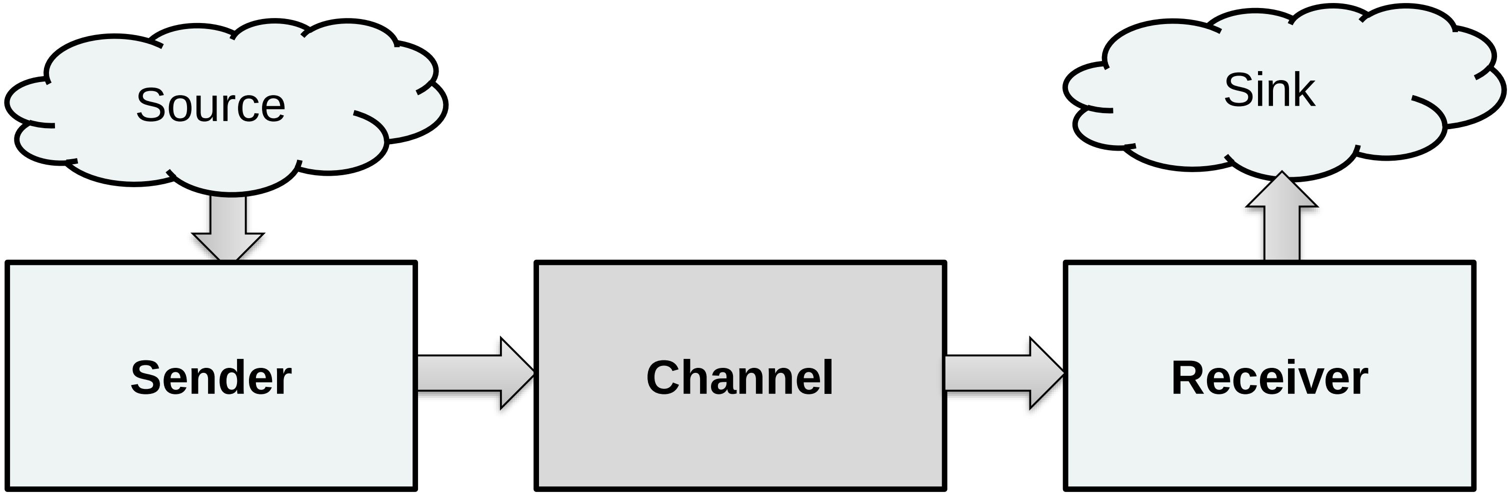

Application domain X, e.g. E-Health
Application domain Y, e.g. Smart Mobility
t1
t2
t3
Realm 1
Realm 2
Realm 3
Realm 1
Realm 2
Realm 3
Secure, controlled communication
Insecure, undetermined communication
Participants, users, stakeholders
Closed system interaction in regard to the environment
Szenario snapshot at timepoint t1, t2, t3,...
Components, systems, functions, controlled services
Services (ad hoc networked, unsecure)

“Overview of the domain structure based on the level of predictability of the behaviour of the systems and human beings involved“,
Source: acatech Studie 2015

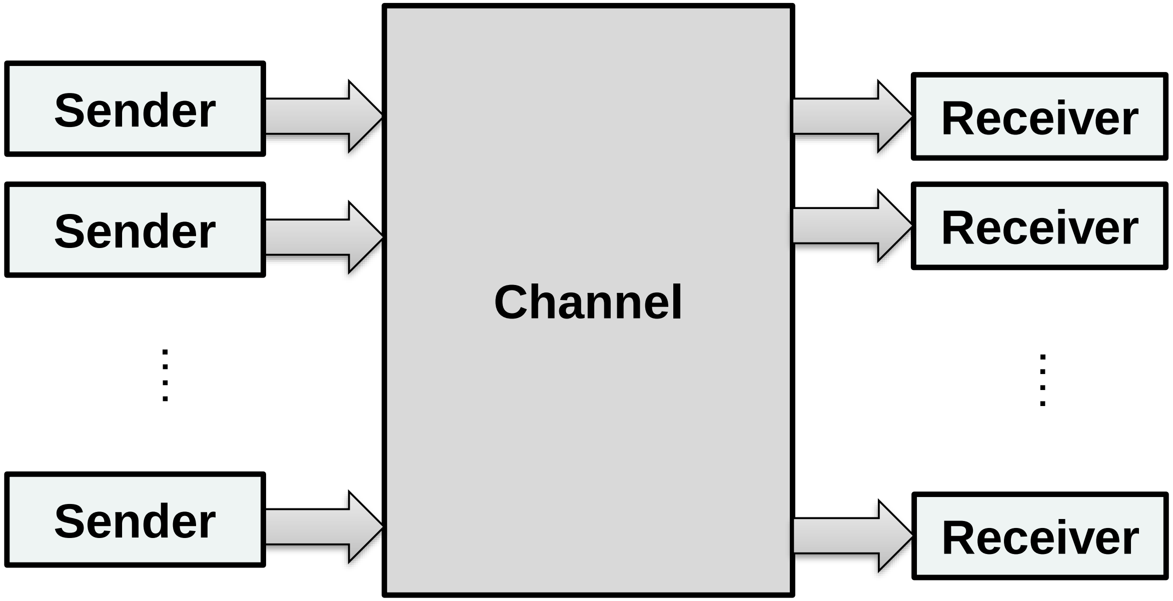
Parts of Communication Systems



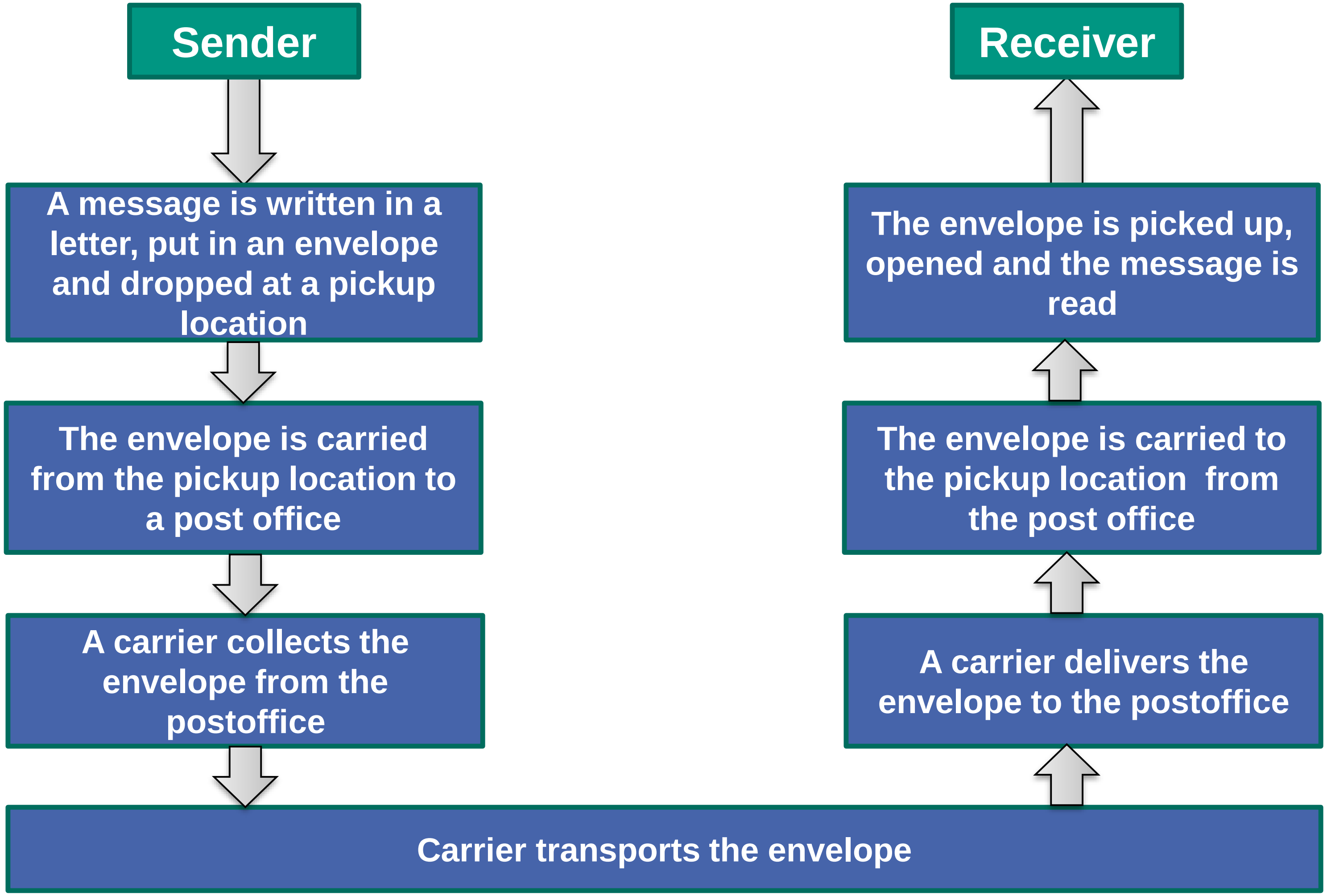
Technical Communication System



Technical Communication System :
Multiple Senders and Receivers



Real world example : Sending a letter



Sender

A message is written in a letter, put in an envelope and dropped at a pickup location

The envelope is carried from the pickup location to a post office

A carrier collects the envelope from the postoffice

Carrier transports the envelope

A carrier delivers the envelope to the postoffice

The envelope is carried to the pickup location from the post office

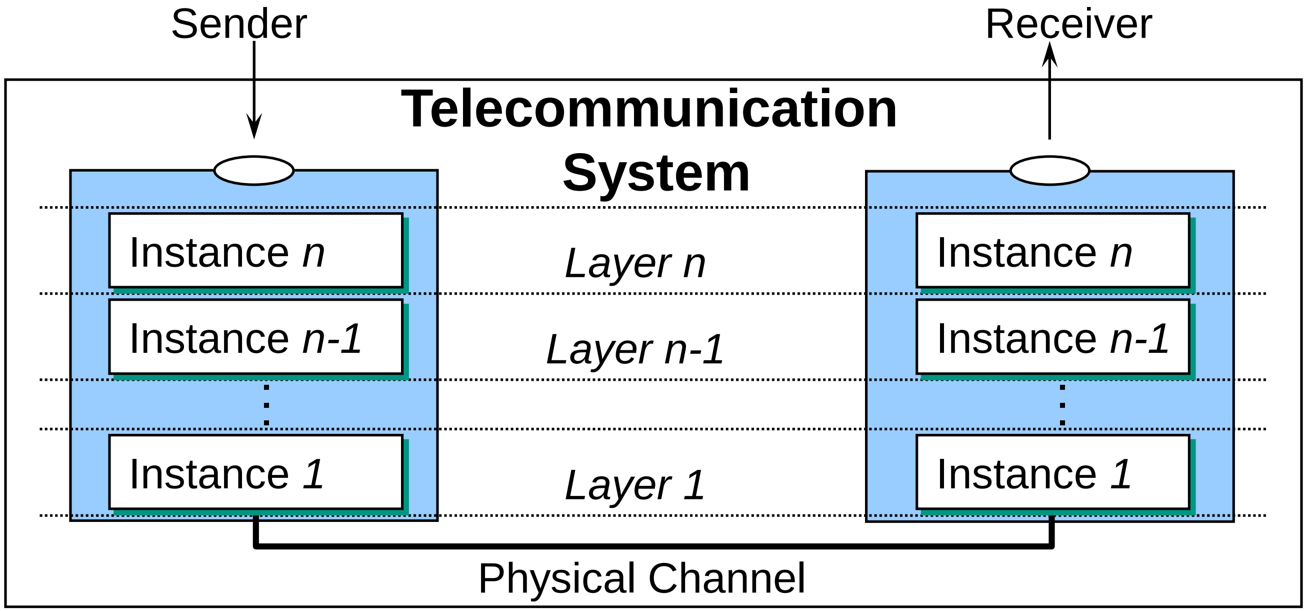
The envelope is picked up, opened and the message is read

Receiver

Examples : DHL, Hermes, DPD, UPS etc

Layer Model

- Segmentation into oversee-able functions
- Modular approach
- Every layer provides a service to the layer above
- The service is a collaboration of the layer’s instances and adheres to a specified protocol
- Highly complex concept, therefore layers are partially merged together



Open Systems Interconnection (OSI) Model

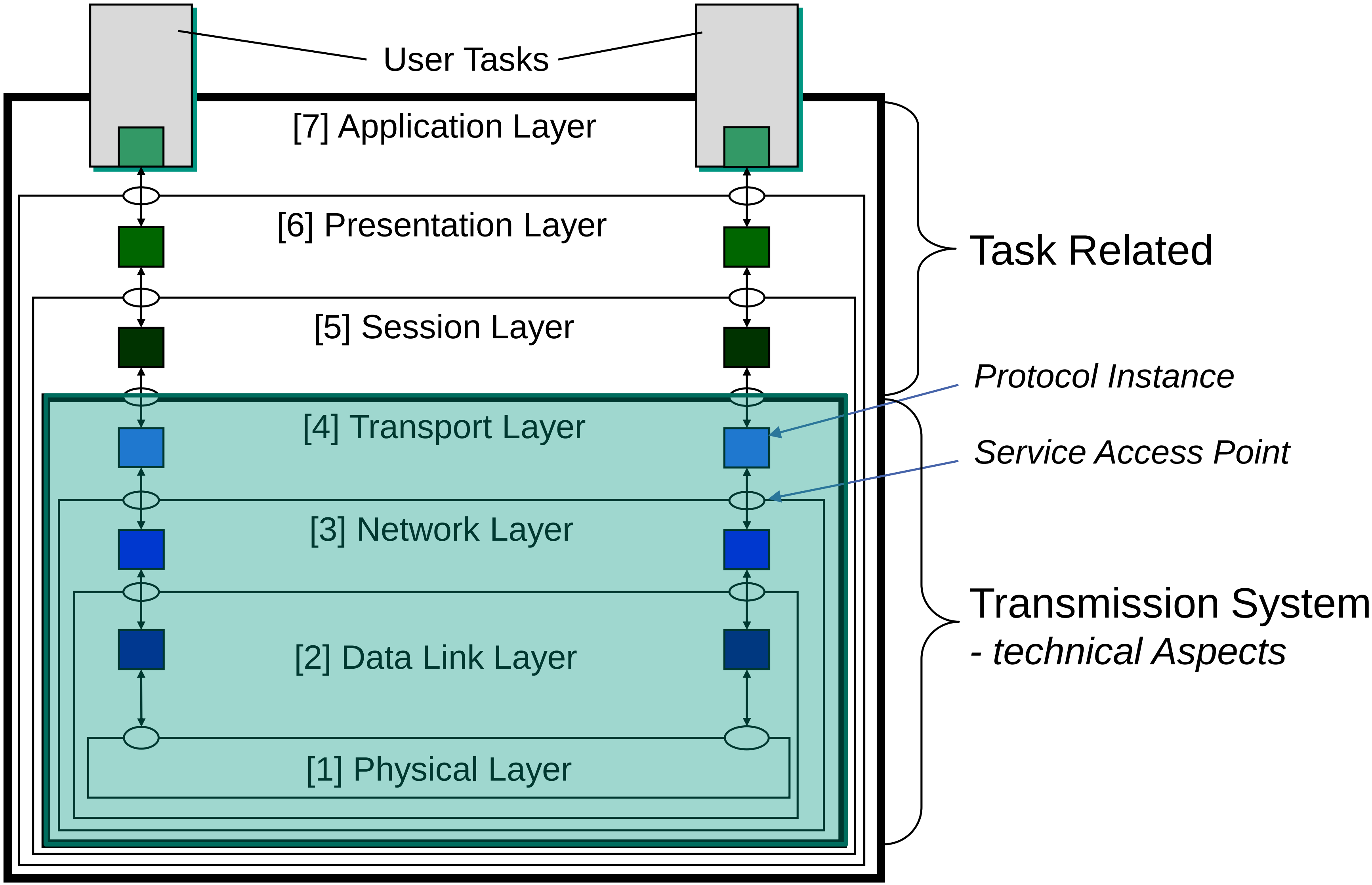
- Most important reference model for computer communication published by International Standards Organization (ISO) 1984 following the preliminary work of CCITT
  - OSI-7-layer model (Basic Reference Model for Open Systems Interconnection)
    - ISO 7498-Standard, later adapted by CCITT as X.200-standard)

- The model partitions the communication system into abstract layers.

- “The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software.
- The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.”

Source: *Behrouz A. Forouzan, “Data Communications and Networking”, Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007*

Logic Architecture: Basic OSI Reference Model



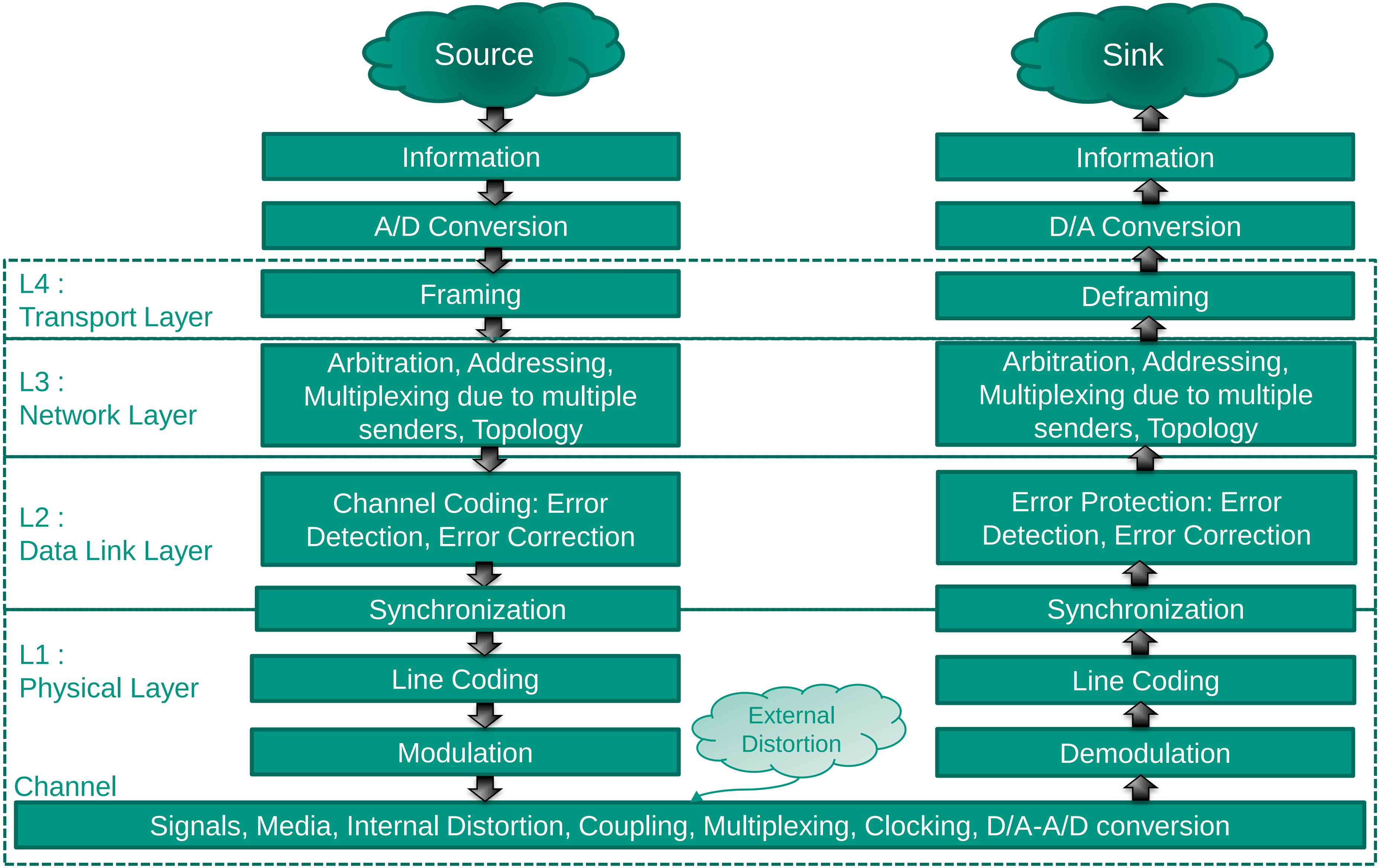
Layer Description

- Physical Layer, Layer 1:
  - Methods and schemes to transmit bit sequences
  - Defines the transmission medium (electrical and mechanical properties)
  - Line Coding
- Data Link Layer, Layer 2:
  - defines formats of data packets
  - defines error detection/correction schemes on logical level
- Network Layer, Layer 3:
  - Addressing of nodes
  - Routing and path finding
  - Flow Control
- Transport Layer, Layer 4:
  - Establish logic connections between nodes (channels)
  - provides stable channels

Layer Description

- Session Layer, Layer 5:
  - Control of Connections/Sessions between two communication partners:
    - Permissions
    - Structure of transmission
    - Opening, management, and closing of connection/session
- Presentation Layer, Layer 6:
  - Defines structure of application data including formatting, encryption, data representation, …
- Application Layer, Layer 7 :
  - Provides functions for an application such as:
    - eMail
    - FTP
    - …
  - Data Input and Output is done on this layer

Technical Communication System


Source
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
External Distortion
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

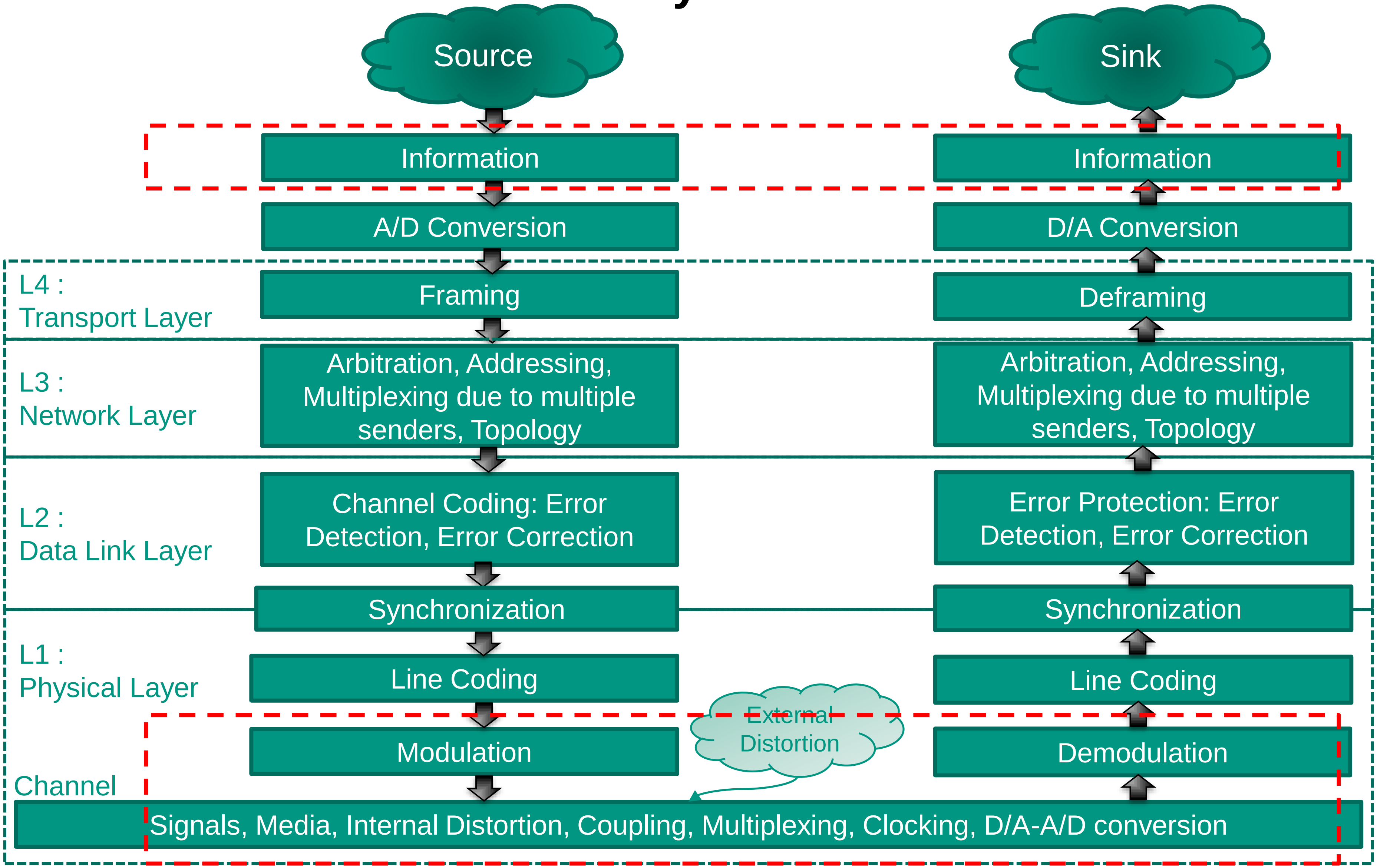
Information and Signals

Contents

- Information
  - Definition
  - Examples

- Signals
  - Representation of Signals

Technical Communication System



Source
Sink
Information
Information
A/D Conversion
D/A Conversion
L4 :
Transport Layer
Framing
Deframing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Error Protection: Error Detection, Error Correction
Synchronization
Synchronization
L1 :
Physical Layer
Line Coding
Line Coding
External Distortion
Modulation
Demodulation
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Definitions for Information:

The communication or reception of knowledge or intelligence

Knowledge obtained from investigation, study, or instruction *(2)* **:** intelligence, news *(3)* **:** facts, data

A signal or character (as in a communication system or computer) representing data *(2)* **:** something (as a message, experimental data, or a picture) which justifies change in a construct (as a plan or theory) that represents physical or mental experience or another construct

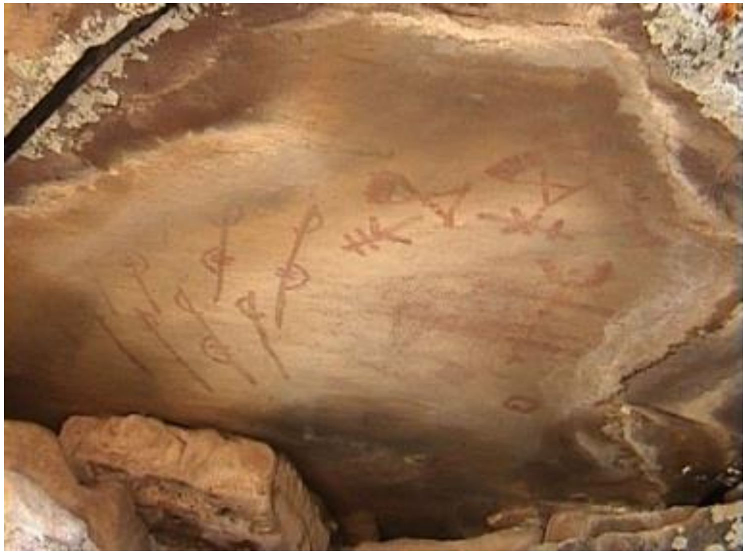
Data that is (1) accurate and timely, (2) specific and organized for a purpose, (3) presented within a context that gives it meaning and relevance, and (4) can lead to an increase in understanding and decrease in uncertainty.

Sources: Merriam-Webster dictionary
http://www.businessdictionary.com/

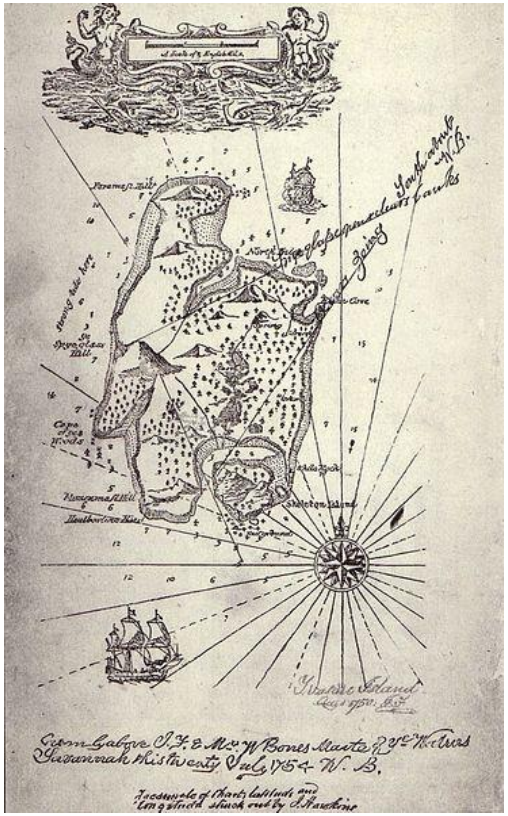
Representation of Information

- Examples:

(www.lavoilotte.de)



(Wikimedia Commons)



(Wikimedia Commons)

(www.abenteuer-informatik.de)

Signals

- Definition:

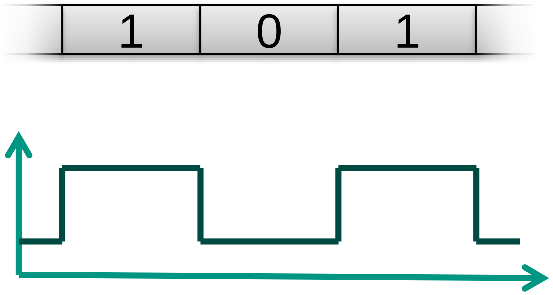
  A **signal** is a function that conveys information about the behavior or attributes of some phenomenon.

  Roland Priemer (1991). Introductory Signal Processing. World Scientific.

  - Remarks:
    - Every physical quantity can form a signal
    - Physical quantity has to be measurable and controllable in order to become a signal for communication purpose
      - Random signals that cannot be influenced, are not usable
    - Assignment of information to a signal is arbitrary
    - For this lecture, we focus on electrical signals

Definition : Bit

- Bit
  - A **bit** (a contraction of **binary digit**) is the basic unit of information in computing and telecommunications. A bit can only have two values: either 1 or 0. A bit can be implemented in hardware by means of a two state device.
  - The symbol for bit, as unit of information, is either simply “bit” or lowercase “b”.



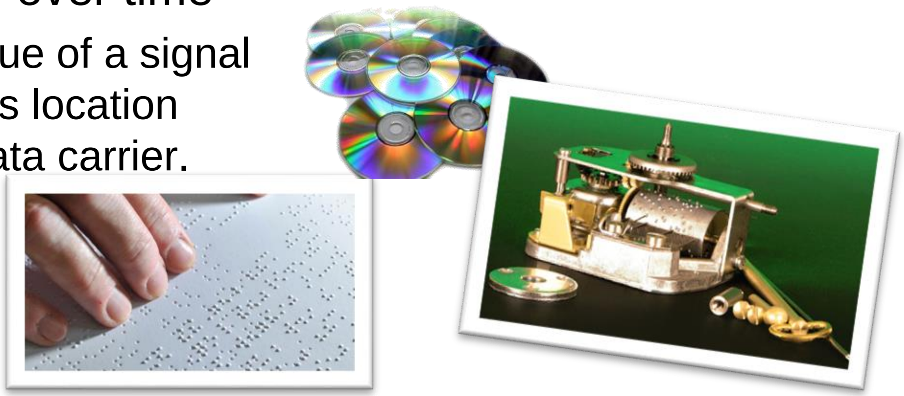
Data

Digital Signal

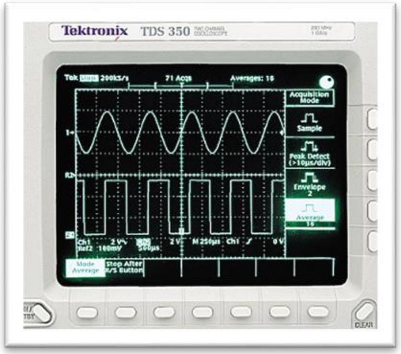
- Bit rate
  - In telecommunications and computing, **bit rate** is the number of bits that are conveyed or processed per unit of time.
  - The bit rate is quantified using the bits per second (bit/s or bps) unit.

Signals and their Parameters

- Signal parameters
  - Physical value of a signal representing information, either through its direct value or its variance of value over time
    - With spatial signals, the value of a signal parameter is a function of its location e.g. location of data on a data carrier.



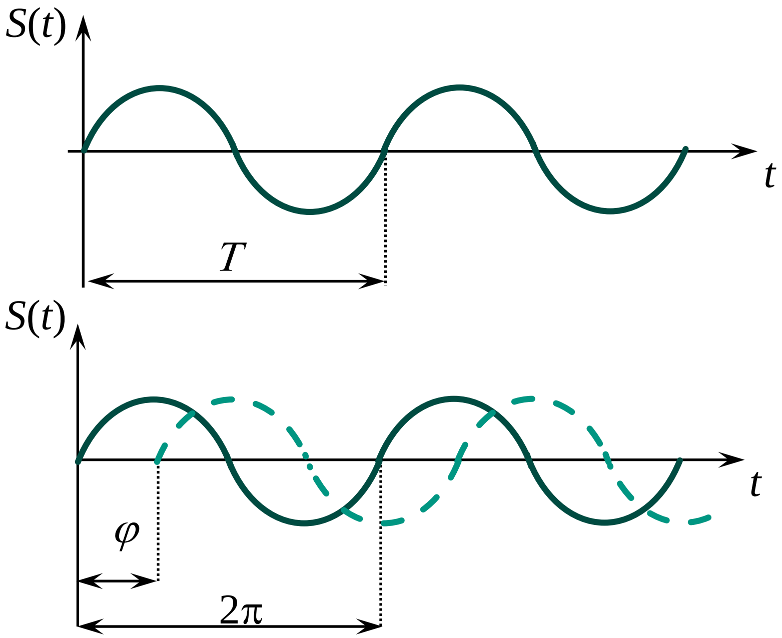
    - With time variant signals, the value of a signal parameter S is a function of time S=S(t)



Periodic and digital Signals

- Parameter of periodic signals: Period T, Frequency 1/T, Amplitude S(t), Phase φ

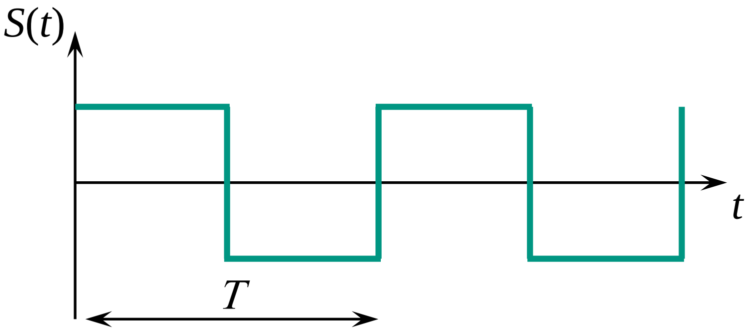
- Examples:
  - Sine Wave
  - Phase difference φ



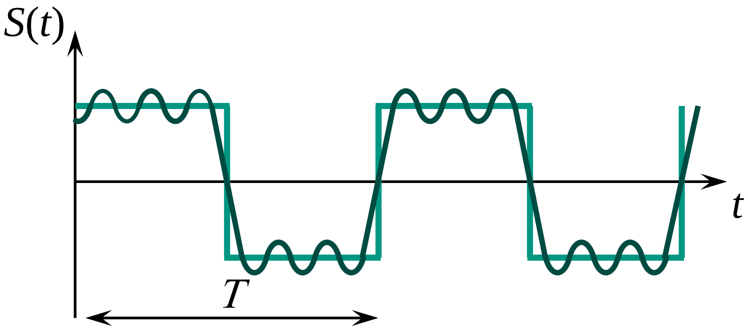
Beyond sinus

- Problem: Many signals are not sinus-formed. How do we model them?

- Example: Square wave („idealized“)



- Solution: Approximation of other signals by sum of sinus signals → Fourier



Signal Representation via Fourier Series

Joseph Fourier, 1822: Every *periodic signal s(t)* can be described as:

- Sum of cosine- and sine-waves of varying frequencies

$$s(t)=\frac{A_0}{2}+\sum_{n=1}^{\infty}\left(A_n\cos(n\omega t)+B_n\sin(n\omega t)\right)$$

- Sum of cosine functions of varying frequencies and phase positions

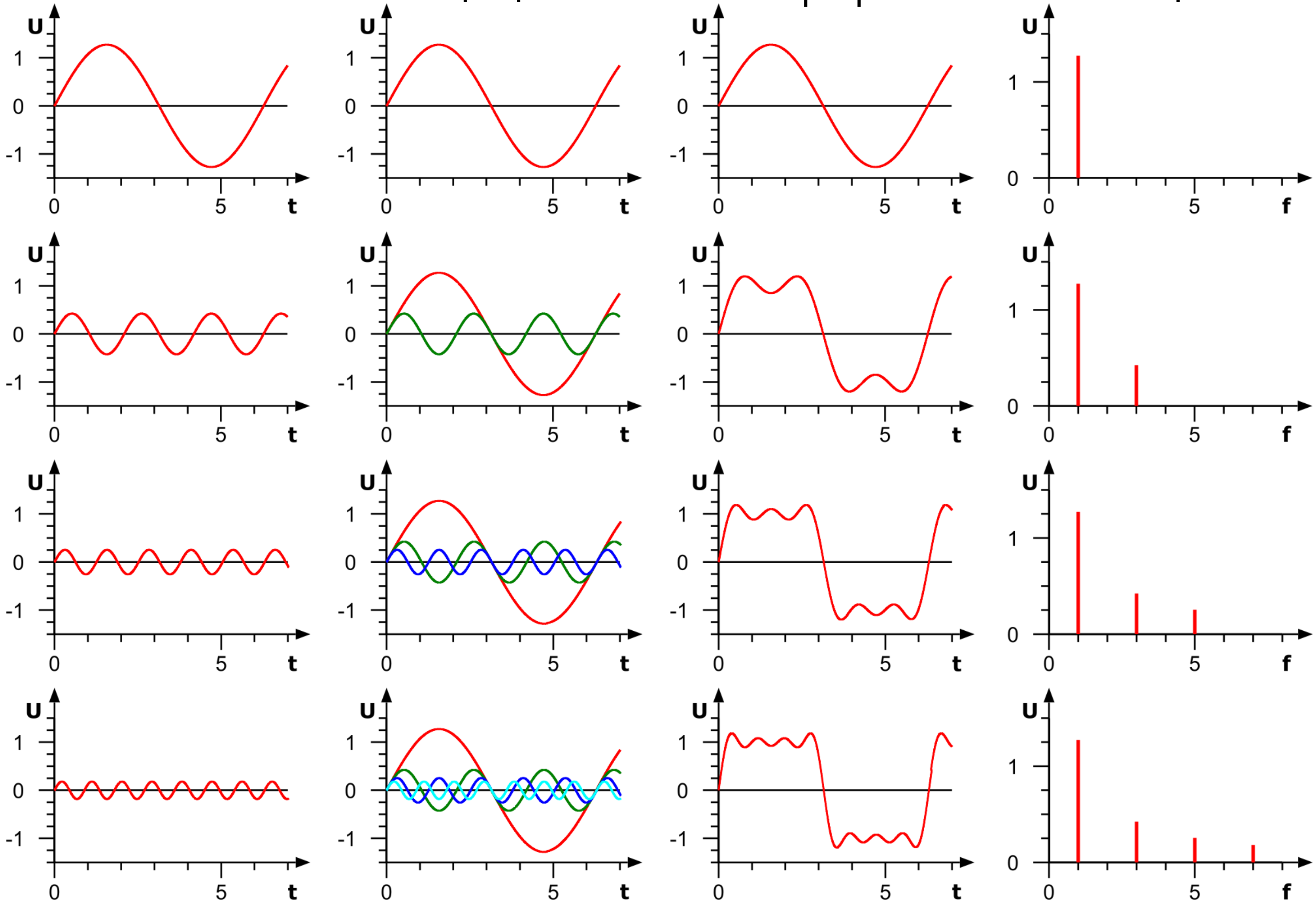
$$s(t)=\frac{A_0}{2}+\sum_{\mathrm{n}=1}^{\infty}C_n\cos(n\omega t+\varphi_n)\quad with\; C_n=\sqrt{A_n^2+B_n^2}$$

$$\varphi_n=\arctan(B_n/A_n+k\pi)$$

$$\begin{array}{lll} & \frac{A_0}{2} & \text{signal offset} \\ \text{with} & \omega=2\pi f_0 & \text{fundamenta l or first harmonic} \\ & \omega_{\mathrm{n}}=n2\pi f_0 & \text{n - } th \text{ harmonic} \end{array}$$

Superposition of Sines

Sine Waves | Superposition | Result of Superposition | Spectrum



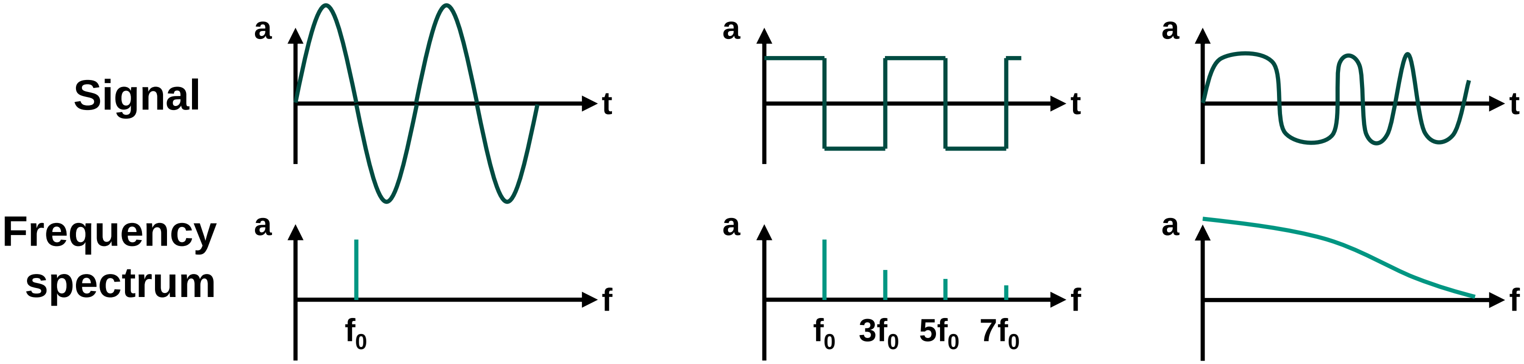
Source: wikimedia commons

Application to non-periodic Signals

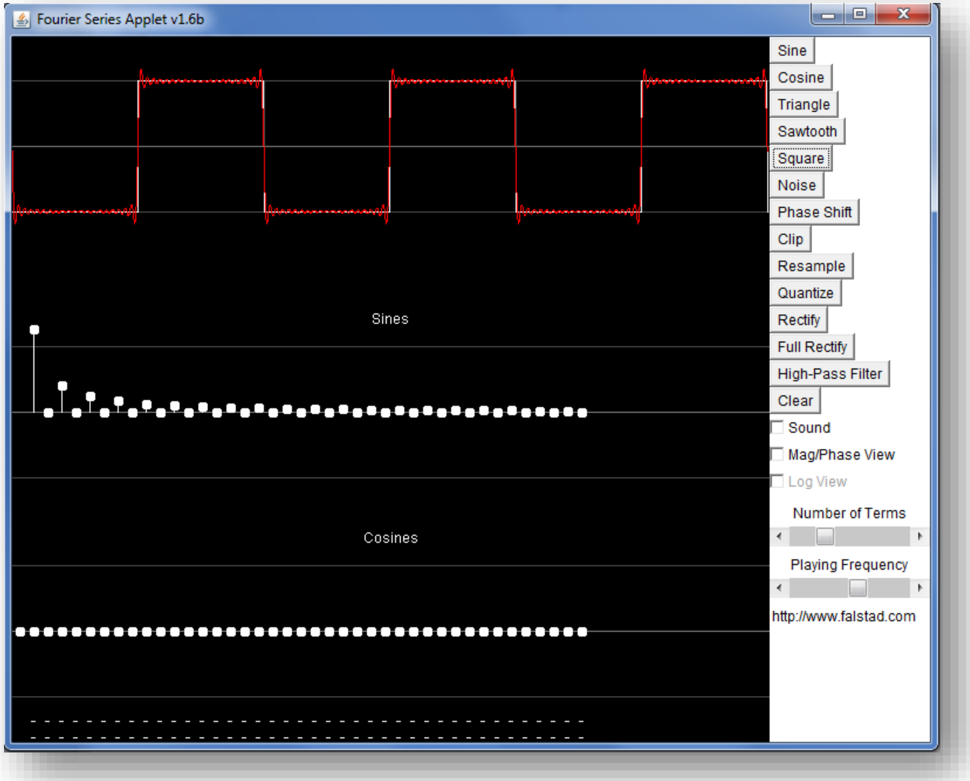
- A non-periodic, finite signal can be seen as *one* period of a periodic signal with an infinite cycle time
  - The discrete line spectrum is turned into a continuous spectrum
  - Summation of the series decomposition is transformed to an integration

$$S(f)=\int_{-\infty}^{\infty} s(t)\cdot e^{-j2\pi ft}dt \quad \text{Fourie r Transform}$$

$$s(t)=\int_{-\infty}^{\infty} S(f)\cdot e^{j2\pi ft}df \quad \text{Fourier Reverse Transform}$$

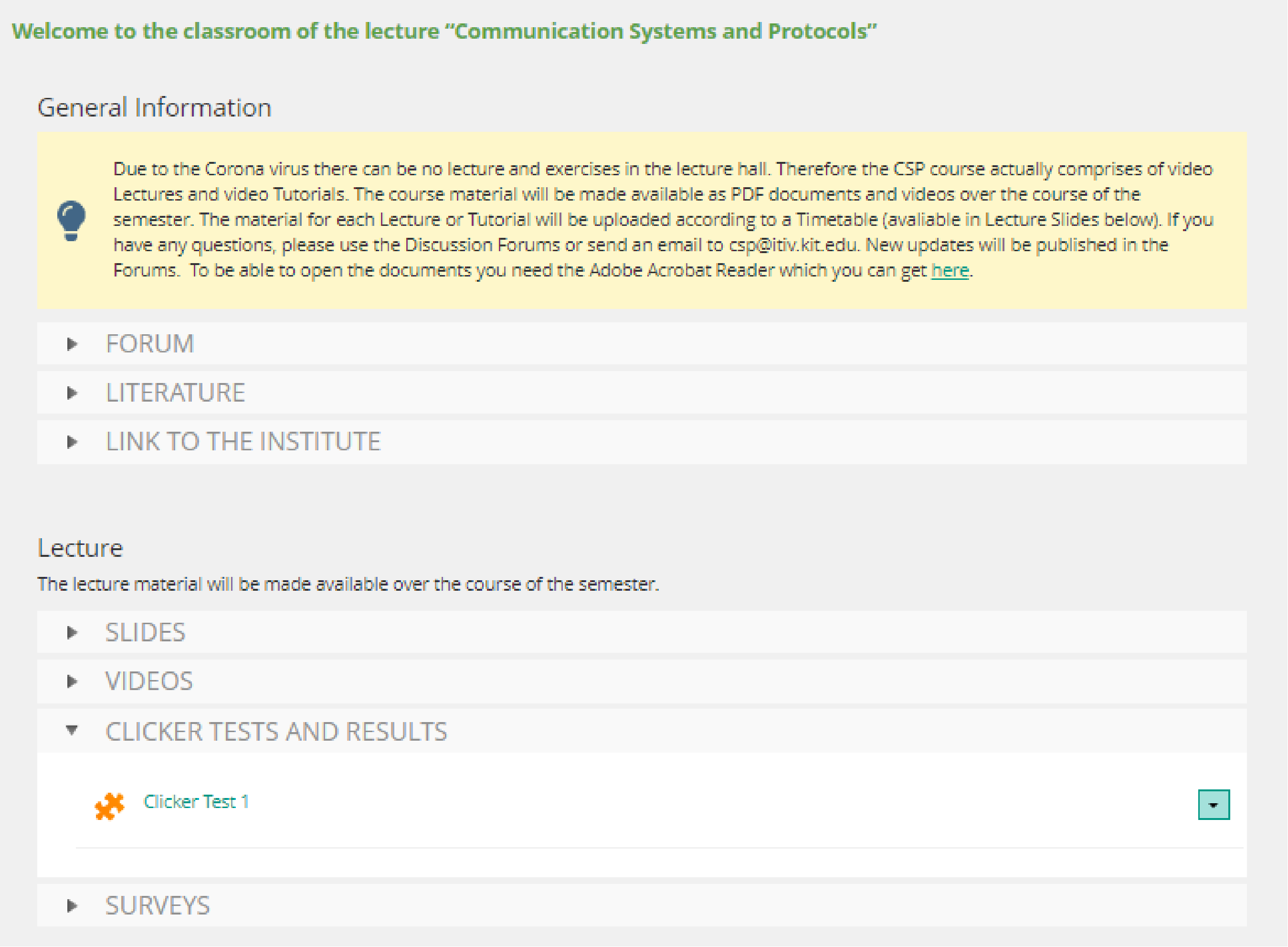


Example Applet



- http://www.falstad.com/fourier/

Participate in Clicker Test 1 : Signals



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Thank you for your attention

Institut für Technik der
Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

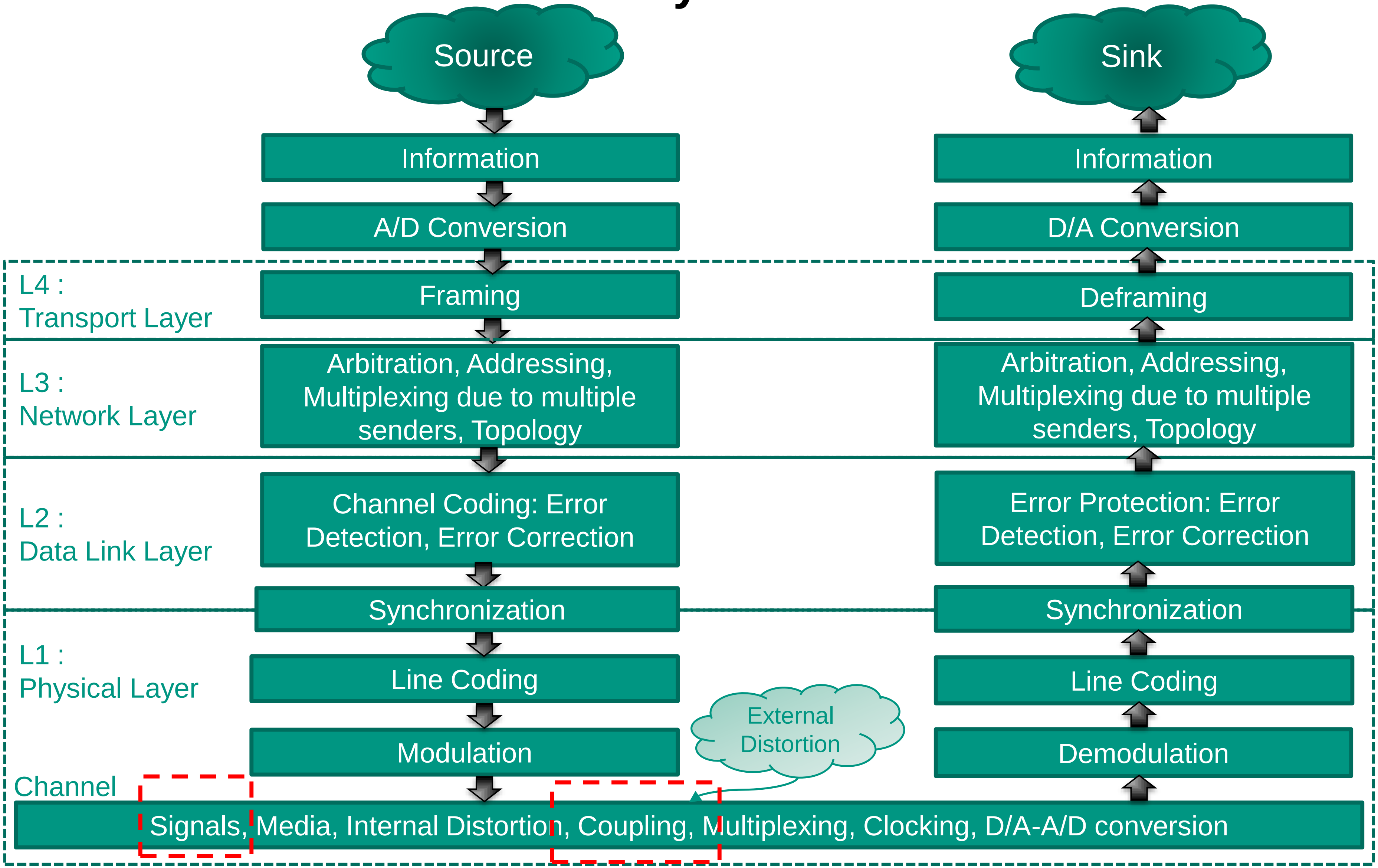
Communication Systems and Protocols

Media

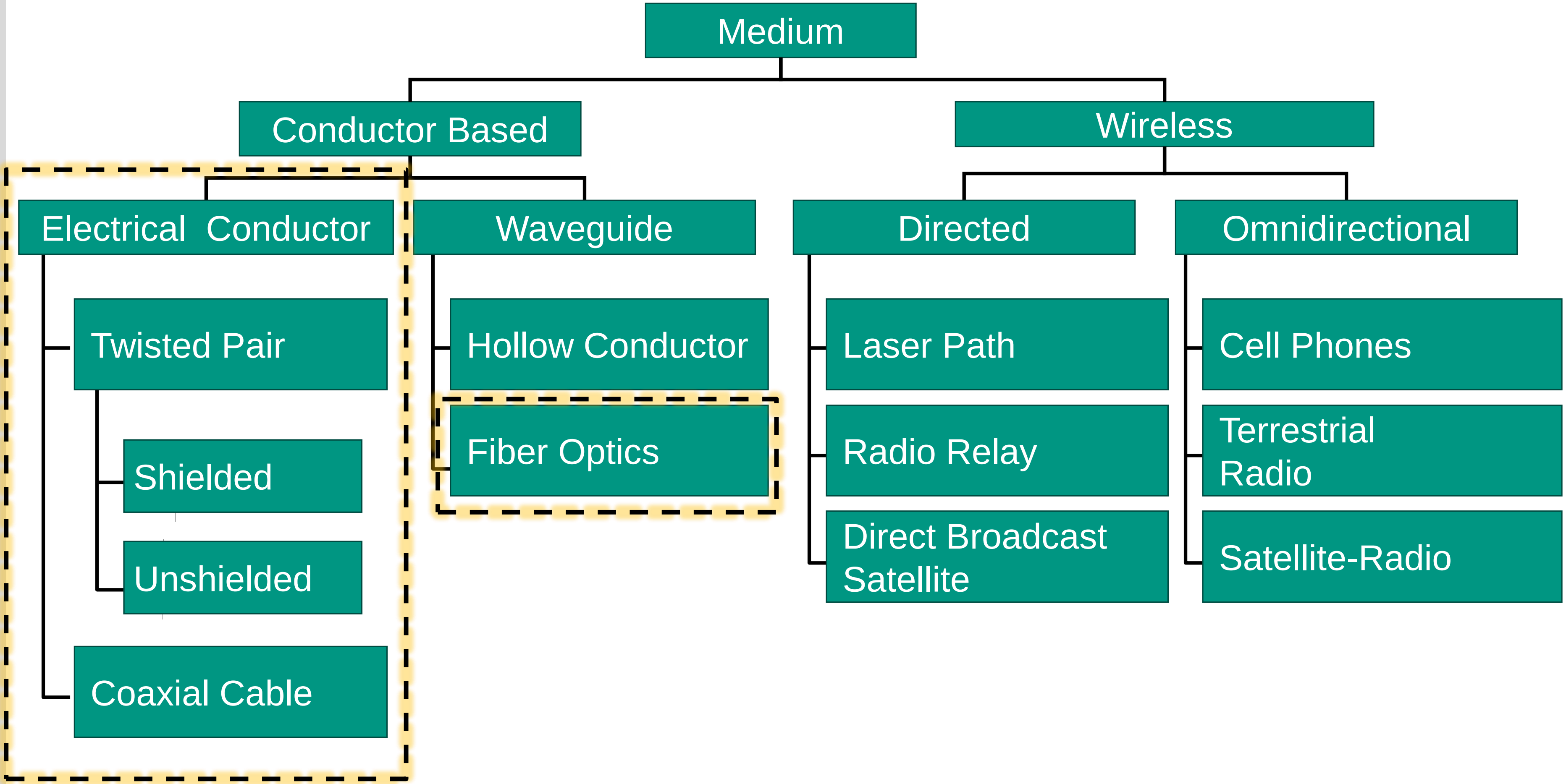
Contents

- Media
  - Classification
  - Signal lines

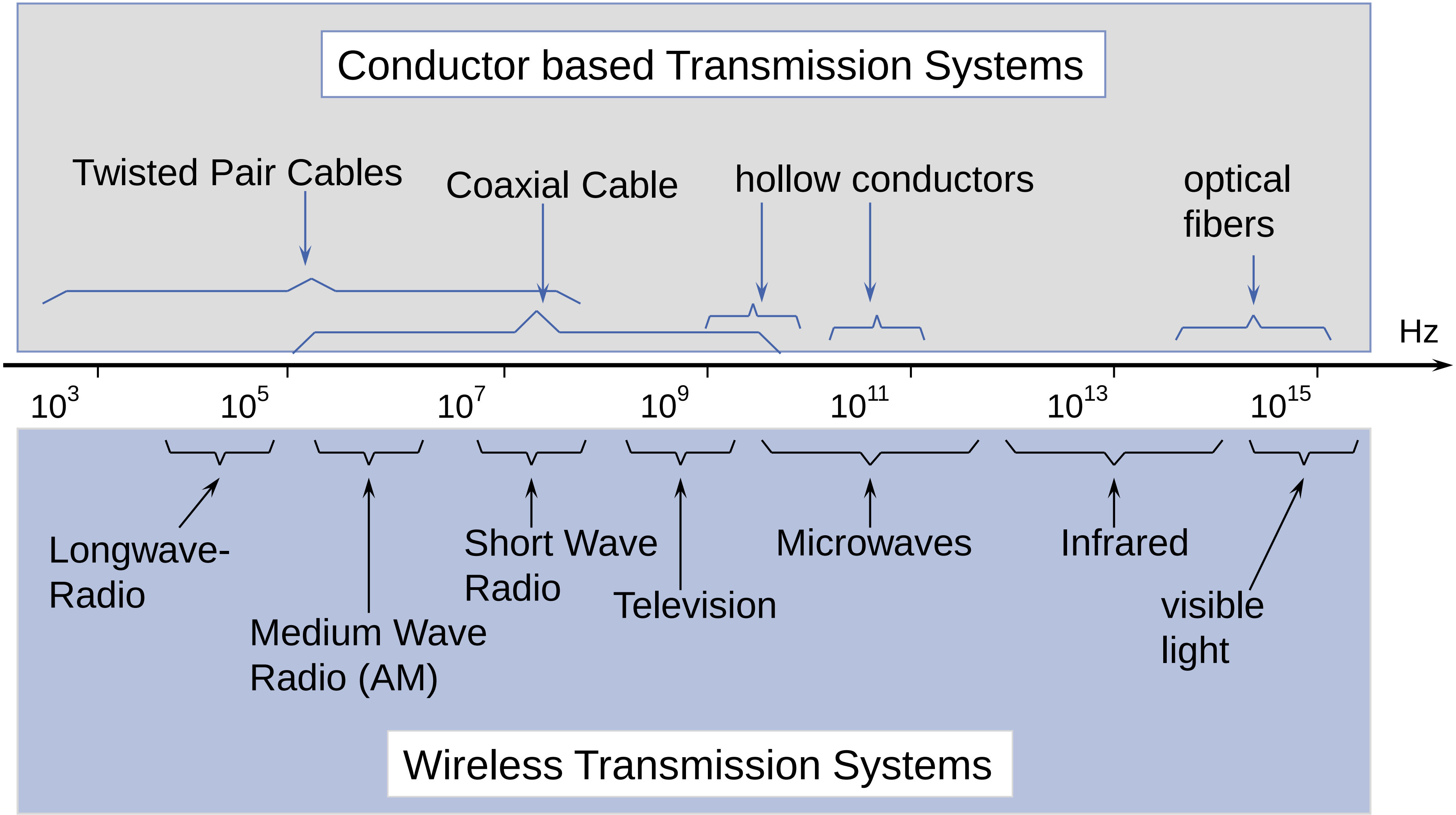
Technical Communication System


Source
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
Channel
External Distortion
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

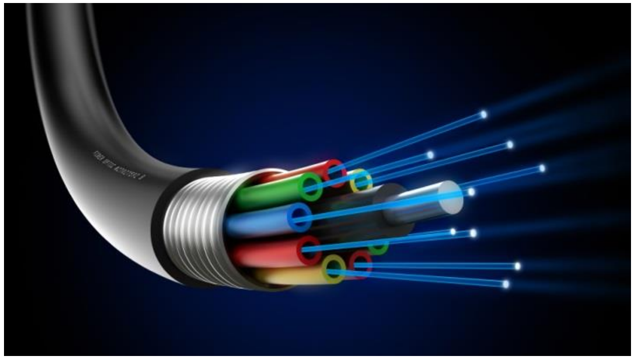
Medium: Classification



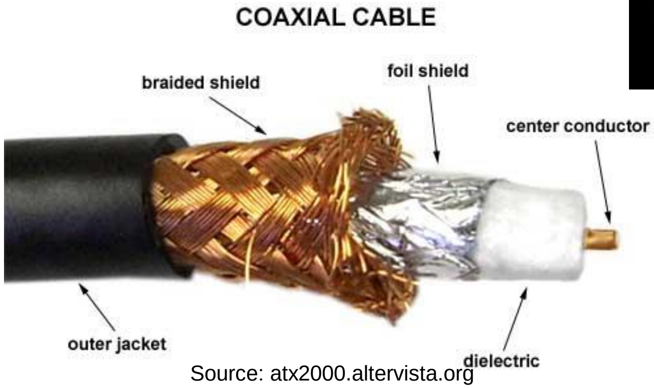
Usage of the electromagnetic Spectrum



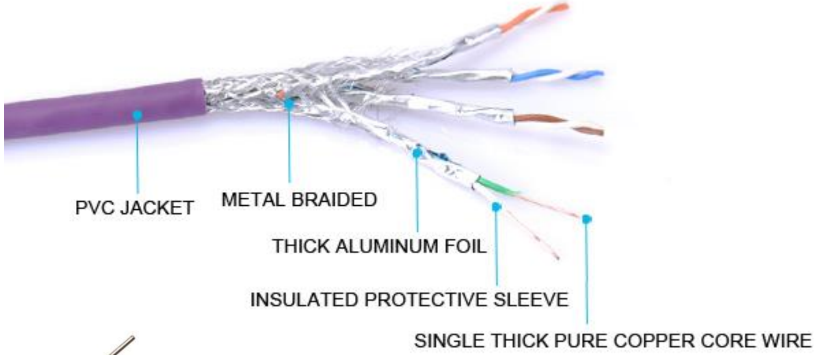
Examples for media



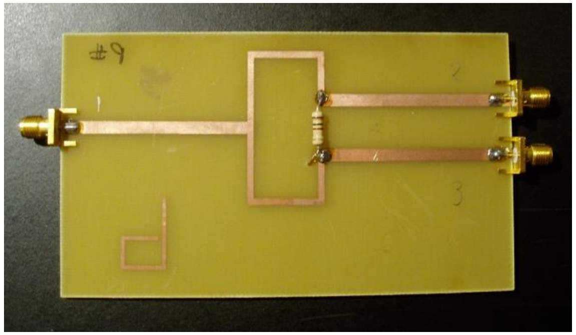
Source: datatalk.co



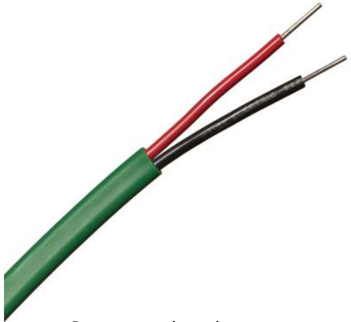
Source: atx2000.altervista.org


PVC JACKET
METAL BRAIDED
THICK ALUMINUM FOIL
INSULATED PROTECTIVE SLEEVE
SINGLE THICK PURE COPPER CORE WIRE

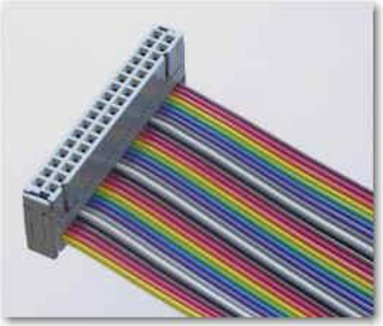
Source: vention.en.alibaba.com



Source: individual.utoronto.ca/georgas



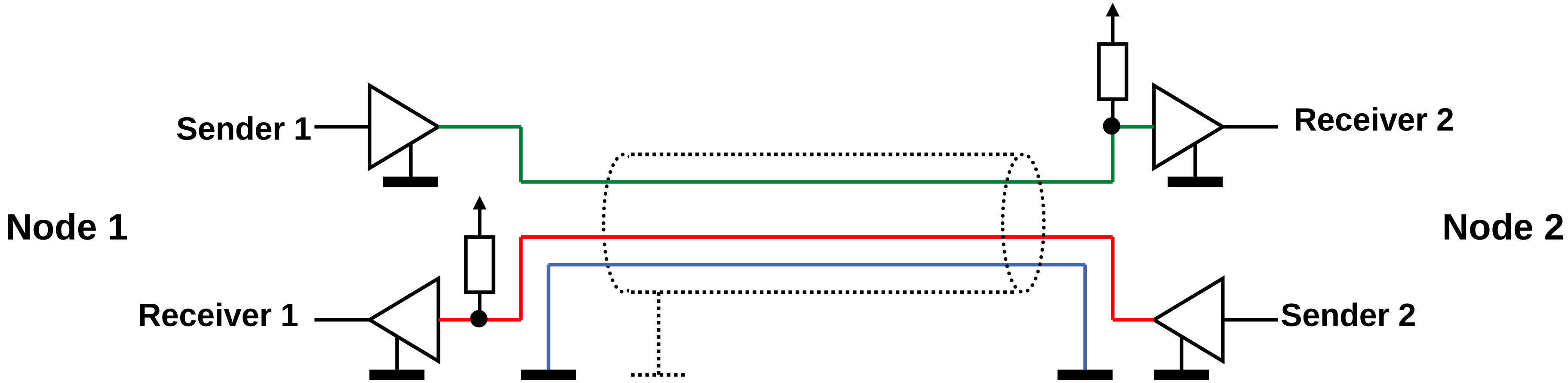
Source: paigewire.com



Source: jitmfg.com

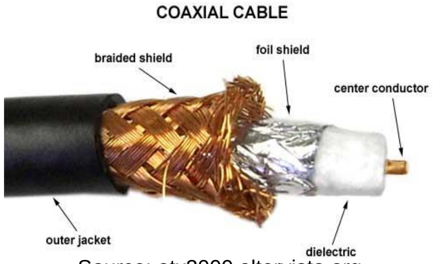
Unbalanced lines

- Unbalanced lines are transmissions lines whose conductors have unequal impedances with respect to ground
- One dedicated conductor per signal
- Second conductor is shared (or ground)
- Increased interference liability, e.g. cross talk
    - Additional shielding can minimize interference effects



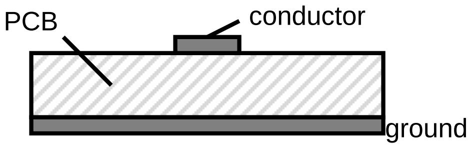
Examples

- Co-axial cable
  - One central conductor
  - Braided shielding around the conductor as second conductor
  - Mostly used for higher frequencies

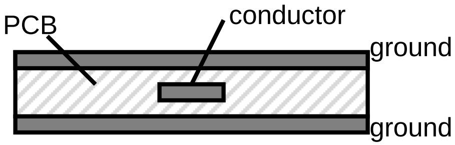


Source: atx2000.altervista.org

- Microstrip
  - Can be made using established manufacturing techniques of PCBs with no additional manufacturing cost
  - On PCBs and Backplanes

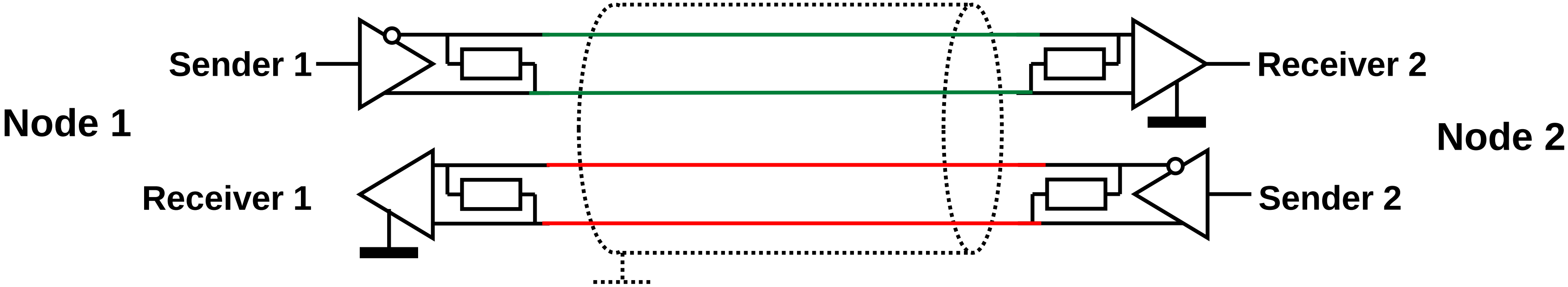


- Stripline
  - Flat conductor with ground plane above and below the conductor
  - On PCBs and Backplanes, robust



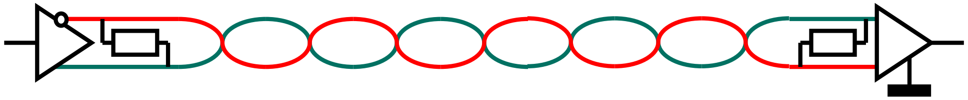
Balanced Lines

- Balanced lines have two conductors of the same type
  - Equal impedances along theirs lengths
  - Equal impedances to ground and other circuits
- Advantage:
  - Noise has the same effects on both conductors
  - External noise can be canceled out when using a differential amplifier as receiver

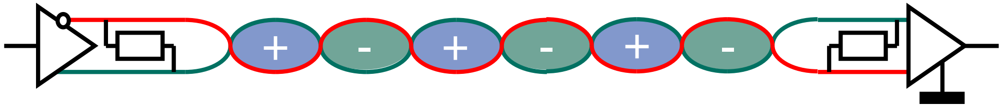


Examples: Twisted-Pair-Cable

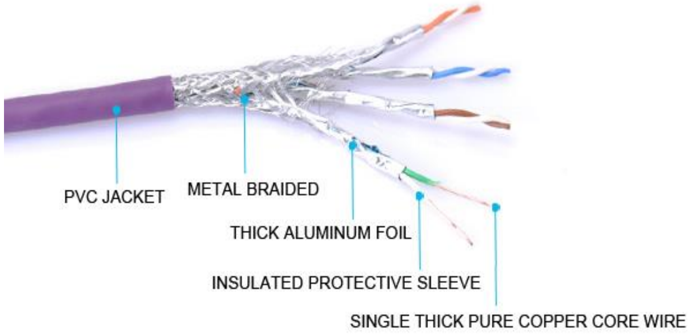
Two conductors of a single circuit are twisted together



- Purpose: Protection against electromagnetic interference from external sources
- Principle: voltage induced in one conductor loop is canceled by induced voltage in next inverted loop



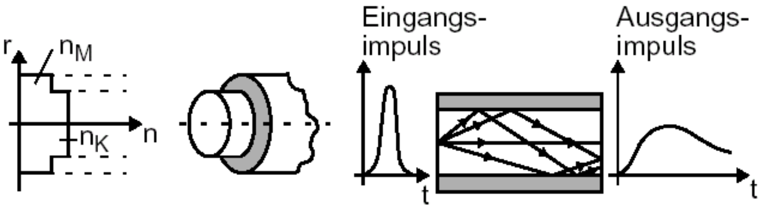
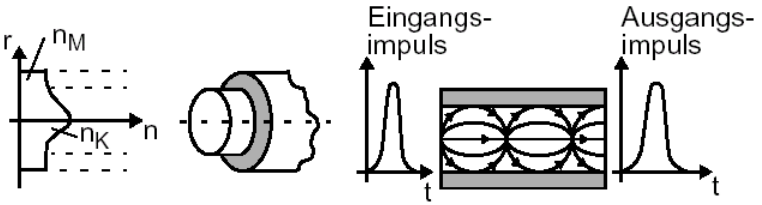
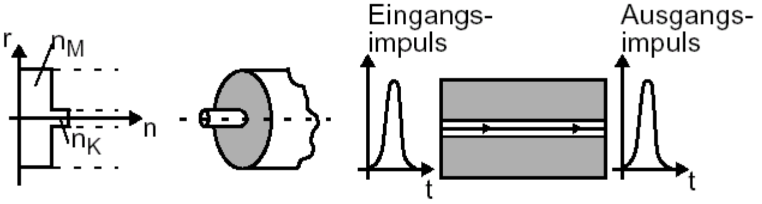
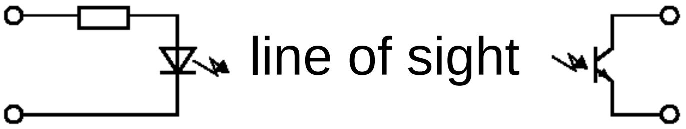
- Often used together with symmetric signaling



PVC JACKET
METAL BRAIDED
THICK ALUMINUM FOIL
INSULATED PROTECTIVE SLEEVE
SINGLE THICK PURE COPPER CORE WIRE

Source: vention.en.alibaba.com

Optical Carrier Media

Type	Principle und Transmission Behavior	Characteristics
Stepindex-Multimode-Fibers	 r nM nK n Eingangs-impuls Ausgangs-impuls t t	Large mode delays result in substantial pulse spreading
Gradientindex-Multimode-Fibers	 r nM nK n Eingangs-impuls Ausgangs-impuls t t	Small mode delays result in minor pulse spreading
Stepindex-Monomode-Fibers	 r nM nK n Eingangs-impuls Ausgangs-impuls t t	Very small mode delays result in almost accurate pulse transmission
Infrared Light	Application of infrared-LEDs und phototransistors 	Short ranged and direct line of sight is mandatory

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

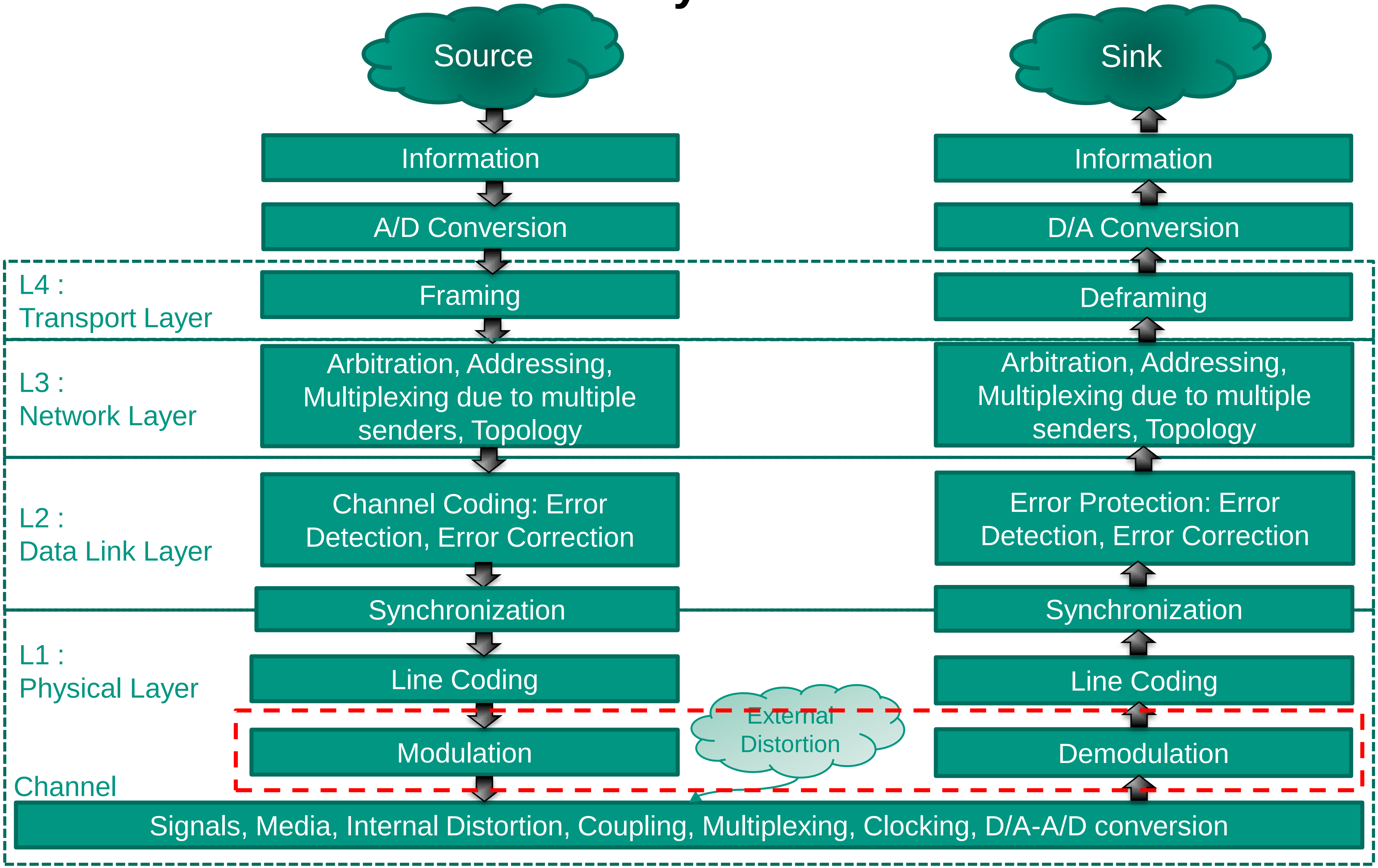
Communication Systems and Protocols

Modulation

Contents

- Modulation and keying shemes
  - Amplitude shift keying
  - Frequency shift keying
  - Phase shift keying
  - Quadrature amplitude modulation

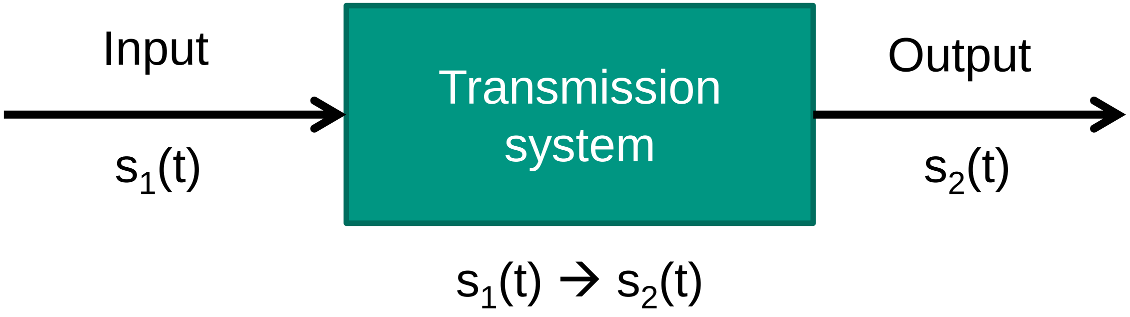
Technical Communication System


Source
Information
A/D Conversion
L4 :
Transport Layer
Framing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Synchronization
L1 :
Physical Layer
Line Coding
Modulation
Channel
External Distortion
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

System theory: Transmission system

- Definition: Transmission system

  A **transmission system** is a mathematical model that describes the transmission behavior of a complex arrangement. It provides a mathematical unique mapping of an output signal to an input signal.
  The mapping is often called *transformation*.



Modulation of an Analog Signal

- With band limited channels it is generally not possible to use an (ideal) base band transmission

- Modulation:
  - Changing characteristics of a carrier frequency
  - Changes are proportional to control signal
  - Receiver measures characteristics and reconstructs the initial control signal from it

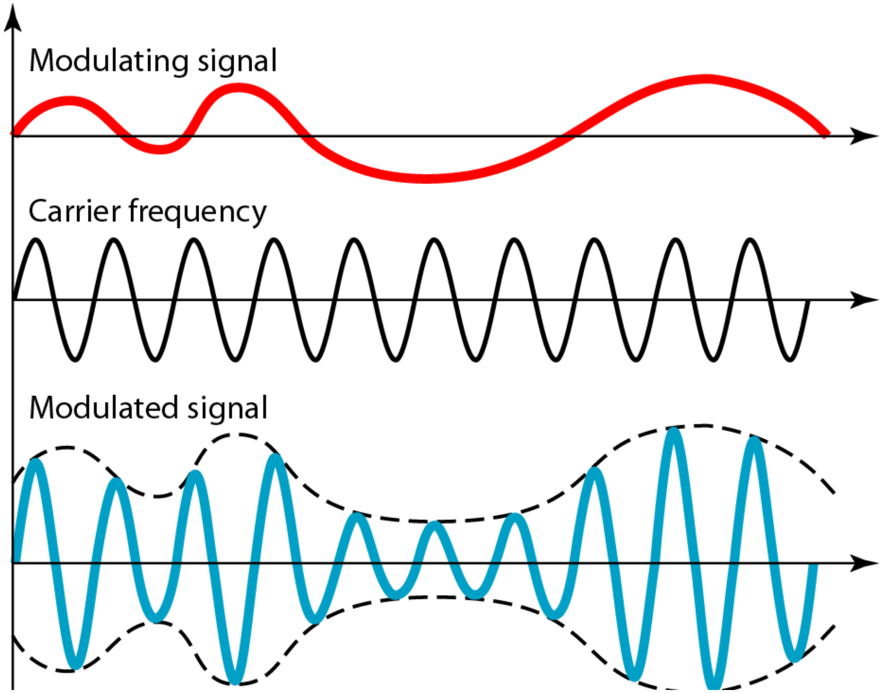


Image Source: *Behrouz A. Forouzan, "Data Communications and Networking", Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007*

Analog-to-Analog conversion

- Modulation in the analog domain
  - unlimited number of signal values
  - continuous variation of signal values

- Why?
  - “Modulation is needed if the medium is bandpass in nature or if only a bandpass channel is available to us”

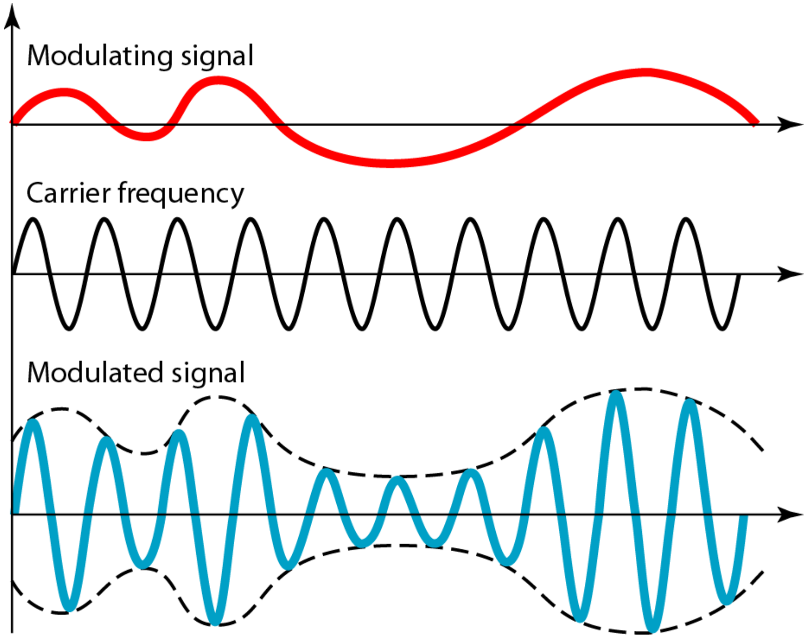
Source: *Behrouz A. Forouzan, “Data Communications and Networking”, Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007*

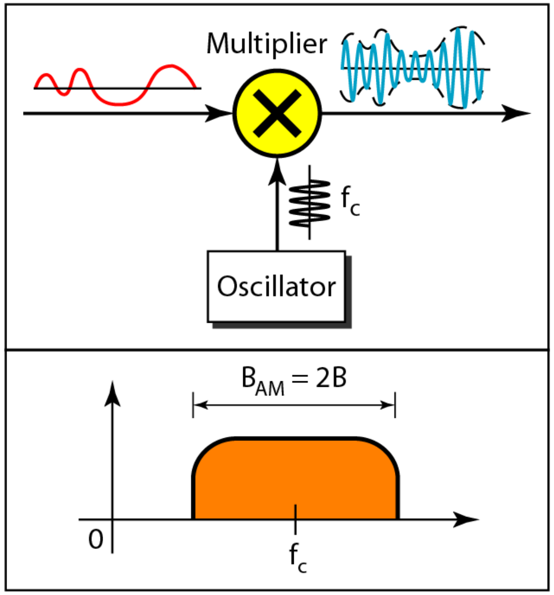
- Types
  - Amplitude Modulation
  - Frequency Modulation
  - Phase Modulation

Amplitude Modulation

- Amplitude of the carrier signal is modulated so that its amplitude varies with the changing amplitudes of the modulating signal.

- Frequency and phase of carrier remain same



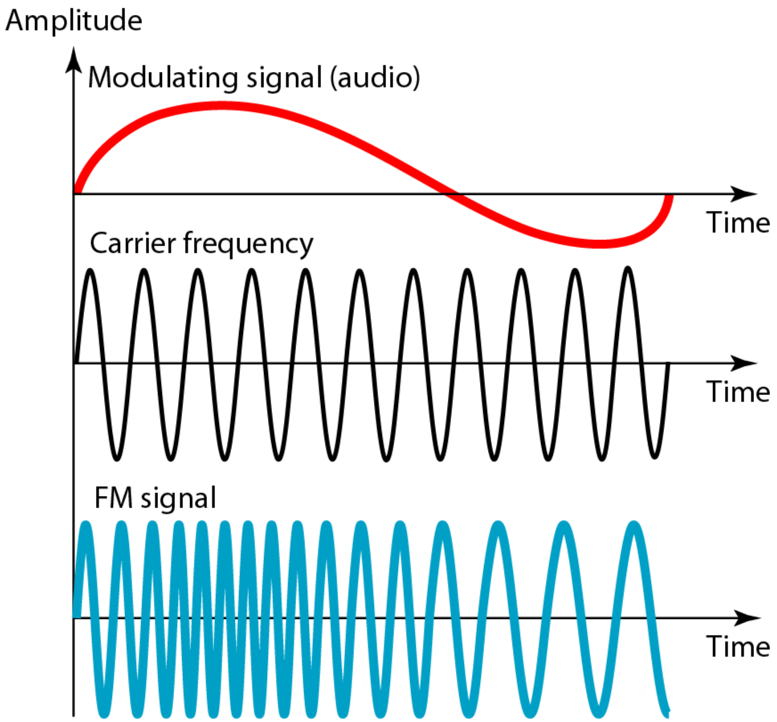


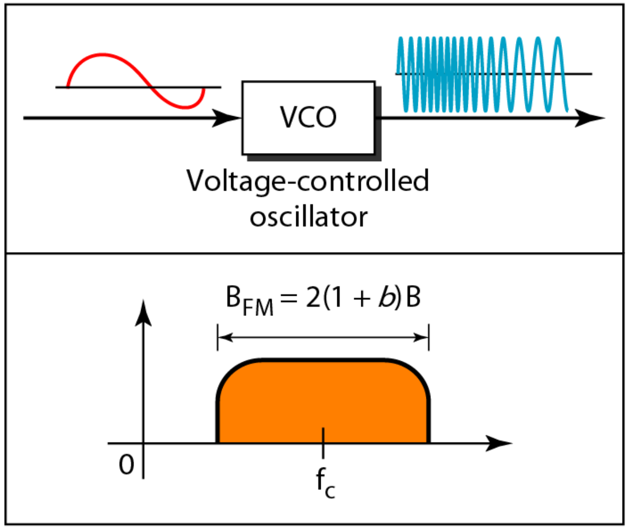
B = Bandwidth of modulating signal

Image Source: *Behrouz A. Forouzan, “Data Communications and Networking”, Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007*

Frequency Modulation

- Frequency of the carrier signal is modulated so that it follows the changing amplitude of the modulating signal.
- Peak amplitude and phase of carrier signal remain constant

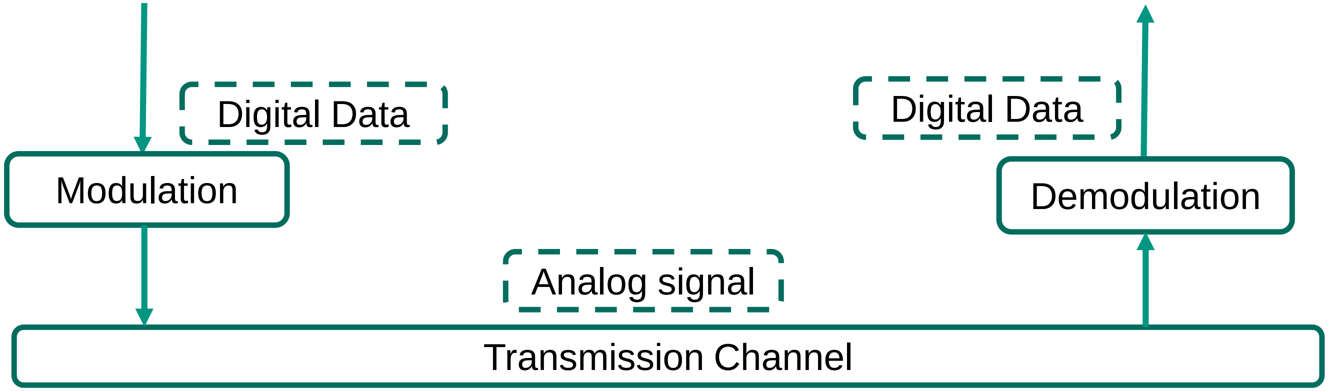




B = Bandwidth of modulating signal
b = factor which depends on modulation technique

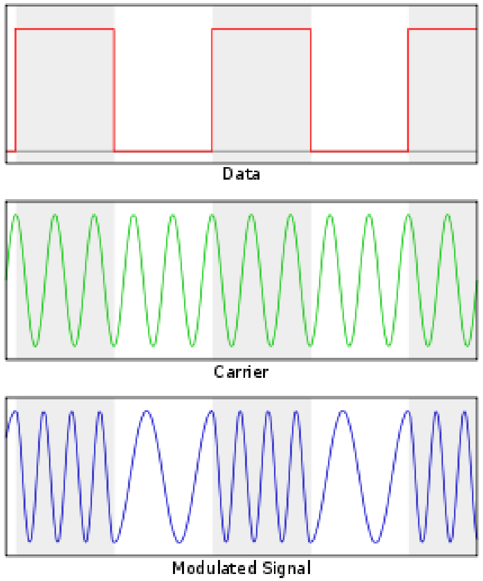
Image Source: *Behrouz A. Forouzan, "Data Communications and Networking", Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007*

Digital-to-Analog Conversion



- Shift-Keying in the digital domain
  - only discrete signal values possible
  - switches from one signal value to the next

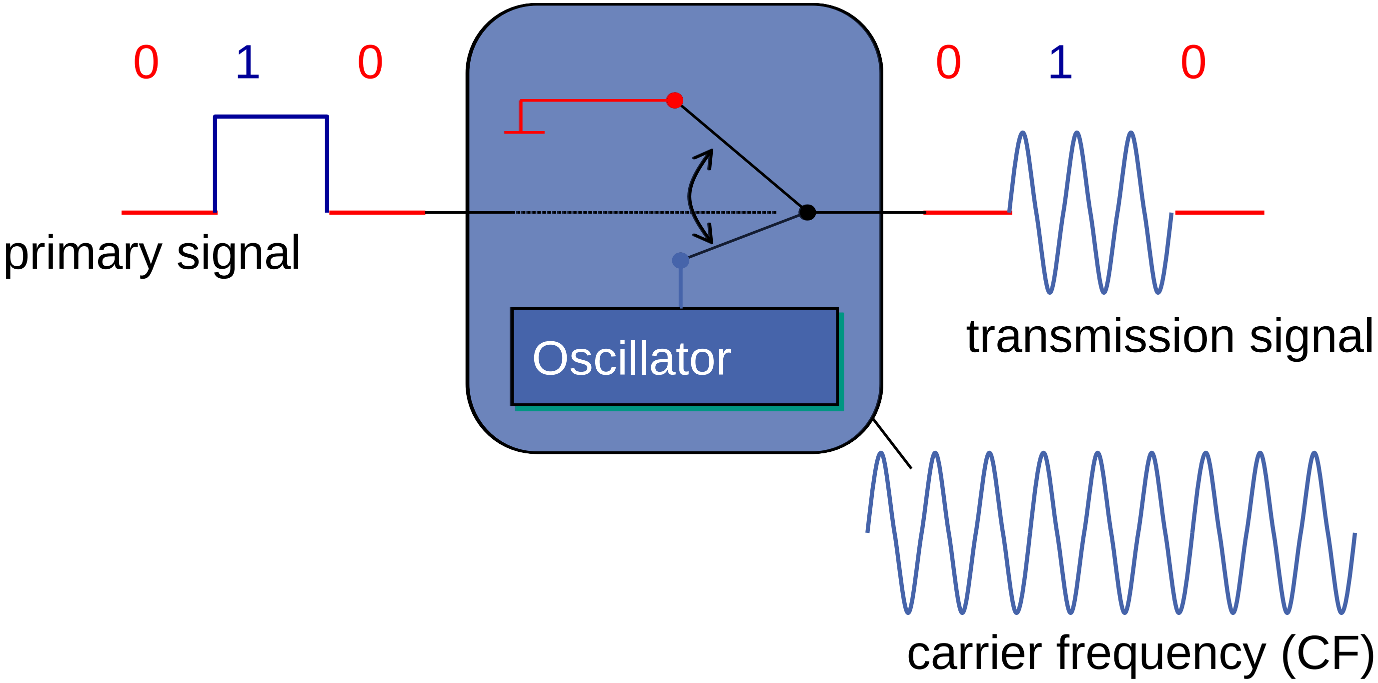
- Digital-to-Analog Conversion
  - Amplitude Shift Keying (ASK)
  - Frequency Shift Keying (FSK)
  - Phase Shift Keying (PSK)
  - Quadrature Amplitude Modulation (ASK and PSK)



Source: http:// en.wikipedia.org

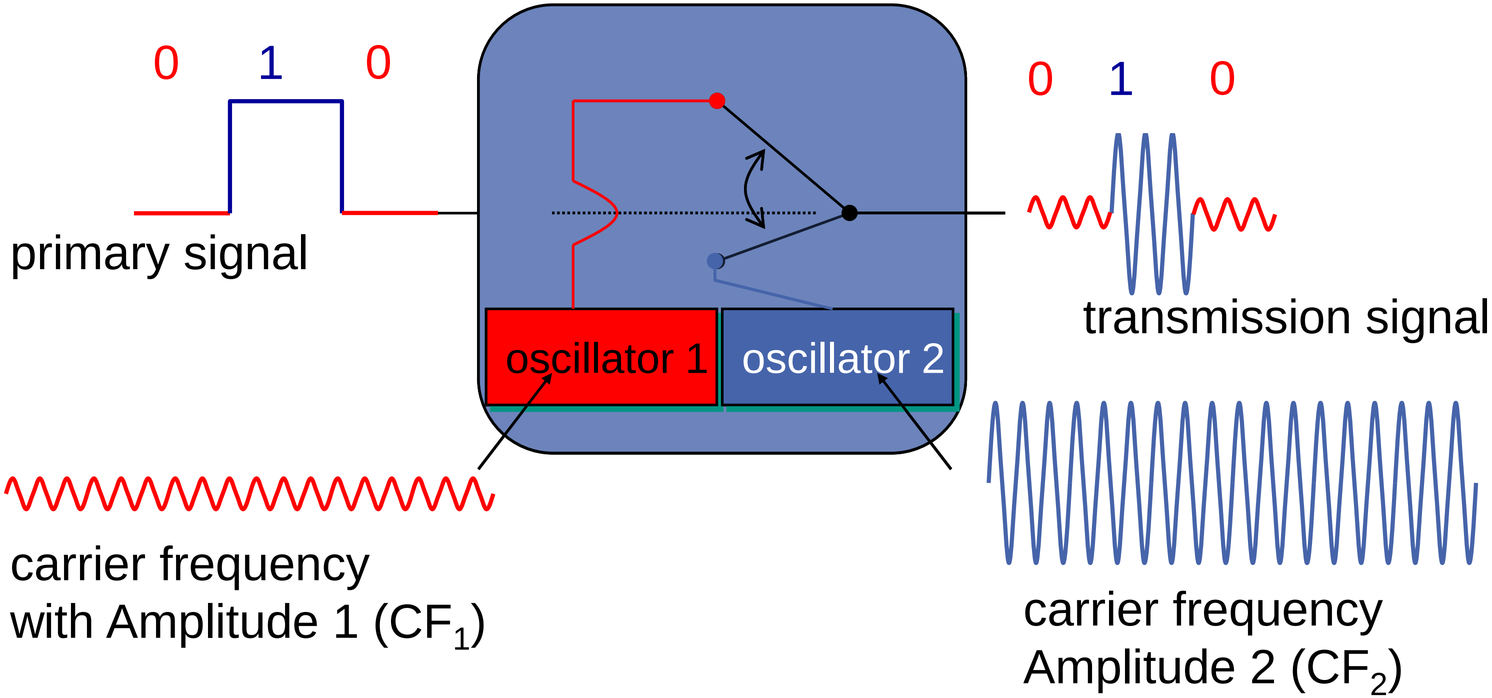
FSK

Amplitude Shift Keying I



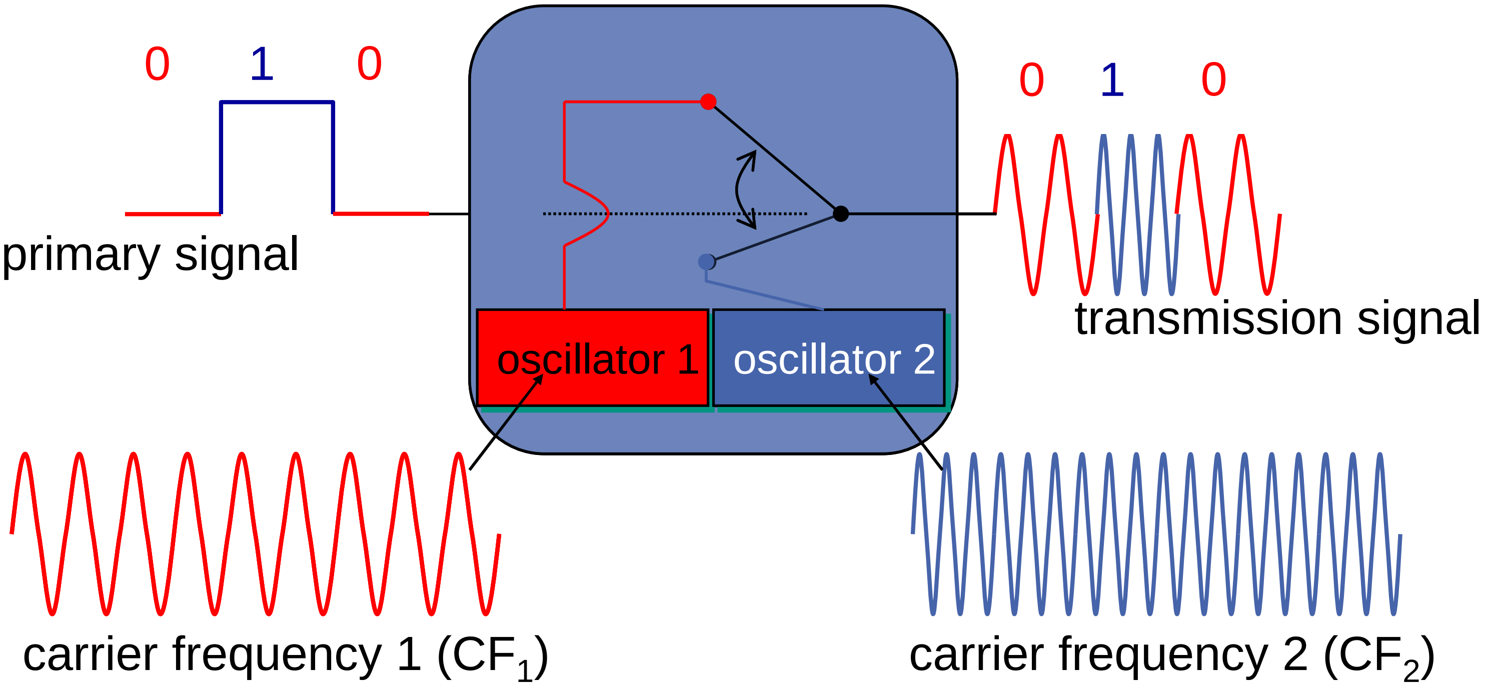
- Primary signal is modulated onto the carrier frequency by altering the amplitude.
- Amplitude modulation is highly susceptible.
- Example: Morse Code

Amplitude Shift Keying II



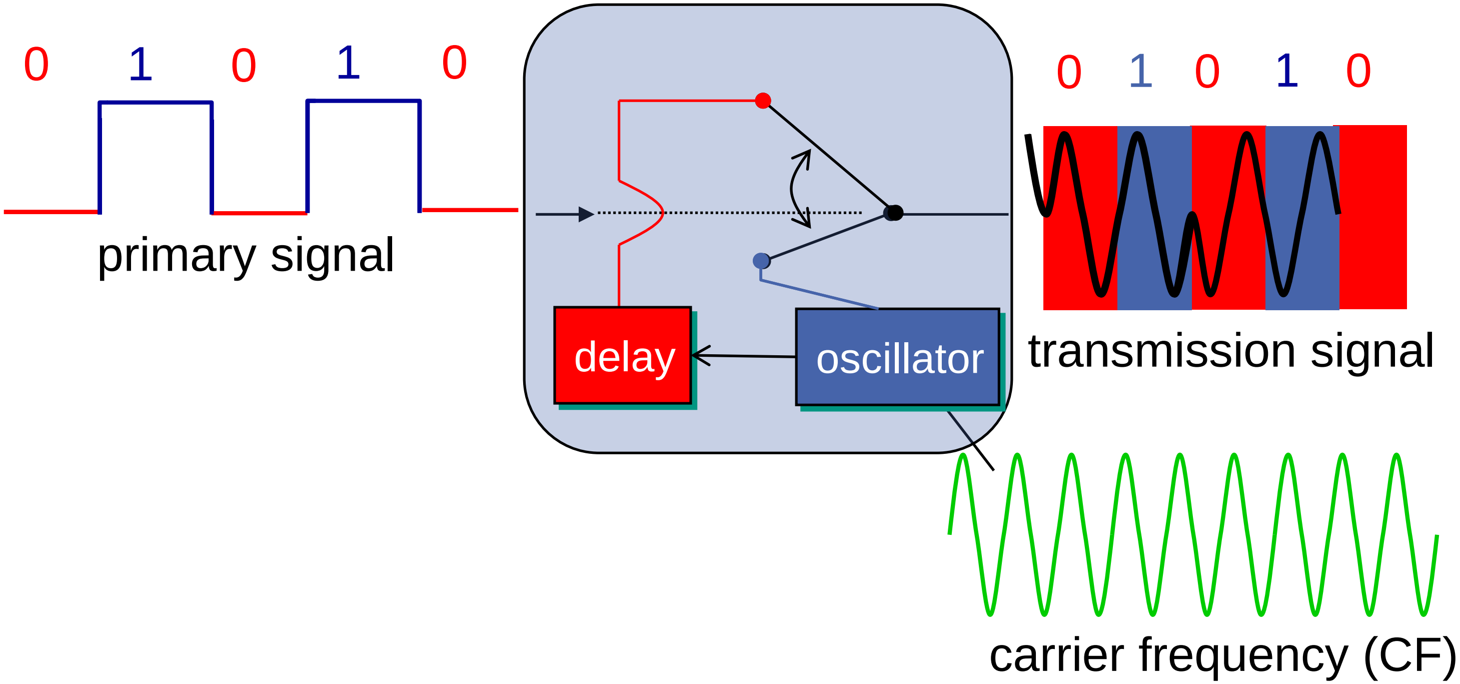
- Primary signal is modulated onto the carrier frequency by altering the amplitude.
- Amplitude modulation is highly susceptible.
- Detection of long runs of 1s requires stable clock generators

Frequency Shift Keying



- Primary signal is modulated via dedicated changes of the carrier frequency
- One of the uses of frequency modulation is the scheme used for VHF terrestrial radio

Phase Shift Keying

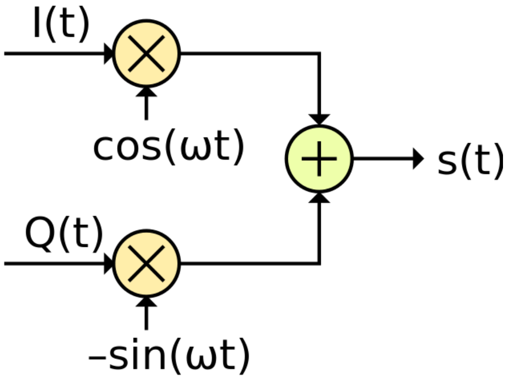


- primary signal is modulated using dedicated jumps in phase of the transmission signal
- Two methods possible
  - Absolute PSK: ‘0’: no phase shift (in comparison to a reference phase)
    ‘1’: phase shift (in comparison to a reference phase)
  - Relative PSK : ‘0’: no phase shift (in comparison to phase of previous bit)
    ‘1’: phase shift (in comparison to phase of previous bit)

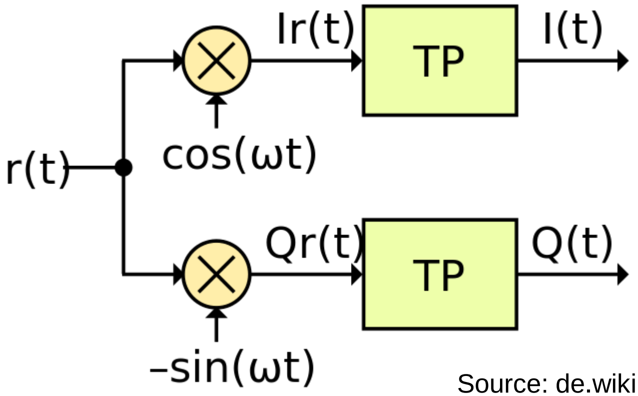
Quadrature Amplitude Modulation (QAM)

- Usage of two sine carrier signals that are shifted 90° to each other → Signals do not interfere with each other
  - One signal is called *In-phase component (I)*
  - Other signal is called *Quadrature Component (Q)*
- These two signals are added up to form the sender signal *(IQ-Modulation)*
- Demodulation requires same phase in sender and receiver → additional measures required

QAM-Modulation



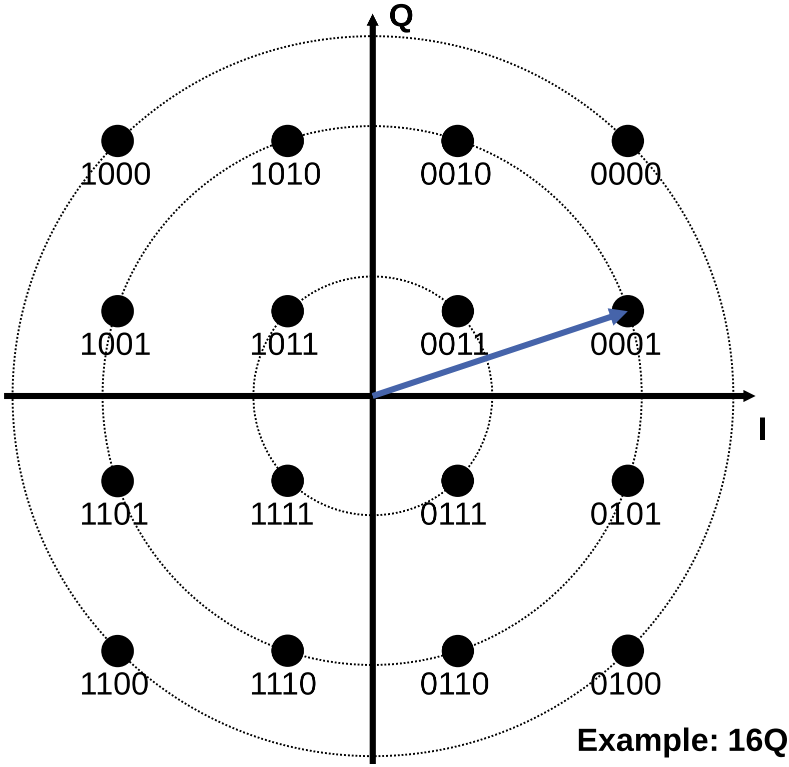
QAM-Demodulation



Source: de.wikipedia.org

Quantizied QAM: Constellation diagram

- The two QAM carriers can be represented in a two-dimensional diagram → **constellation diagram**
- When using discrete and independent signals, each point in the constellation diagram can represent more then one bit



Example: 16QAM

QAM: placement of constellation points

- Signal is influenced by noise during transmission
  - Received signals are not at the ideal positions
  - Acceptance radius is required to define valid ranges for each point

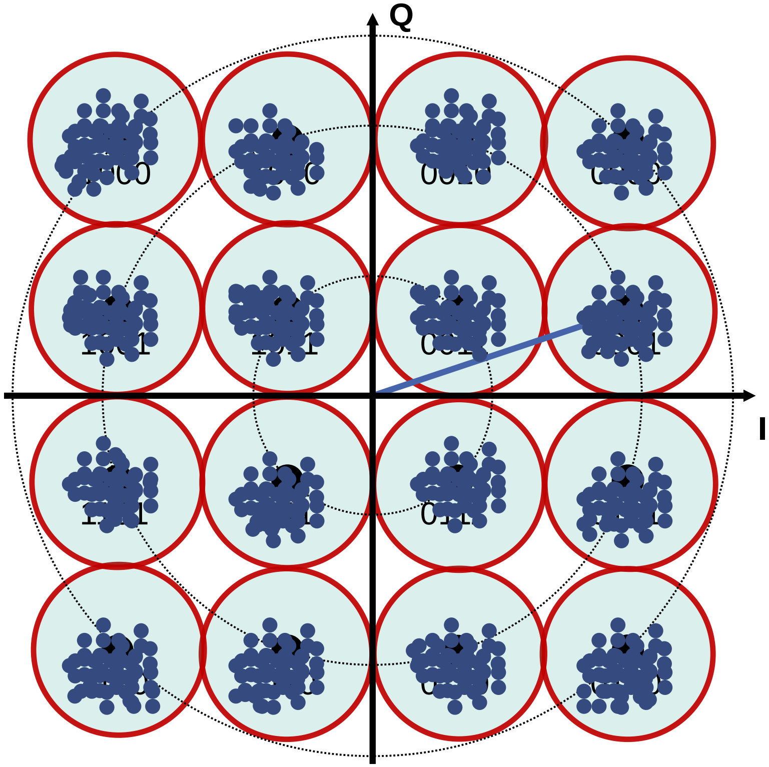
Positioning of signal points:

- Maximize distance of points to avoid misinterpretation of a signal

In General:

- More points allow more bits per signal (higher data rates) but require better signal-to-noise ratios

→ Trade-off required



Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

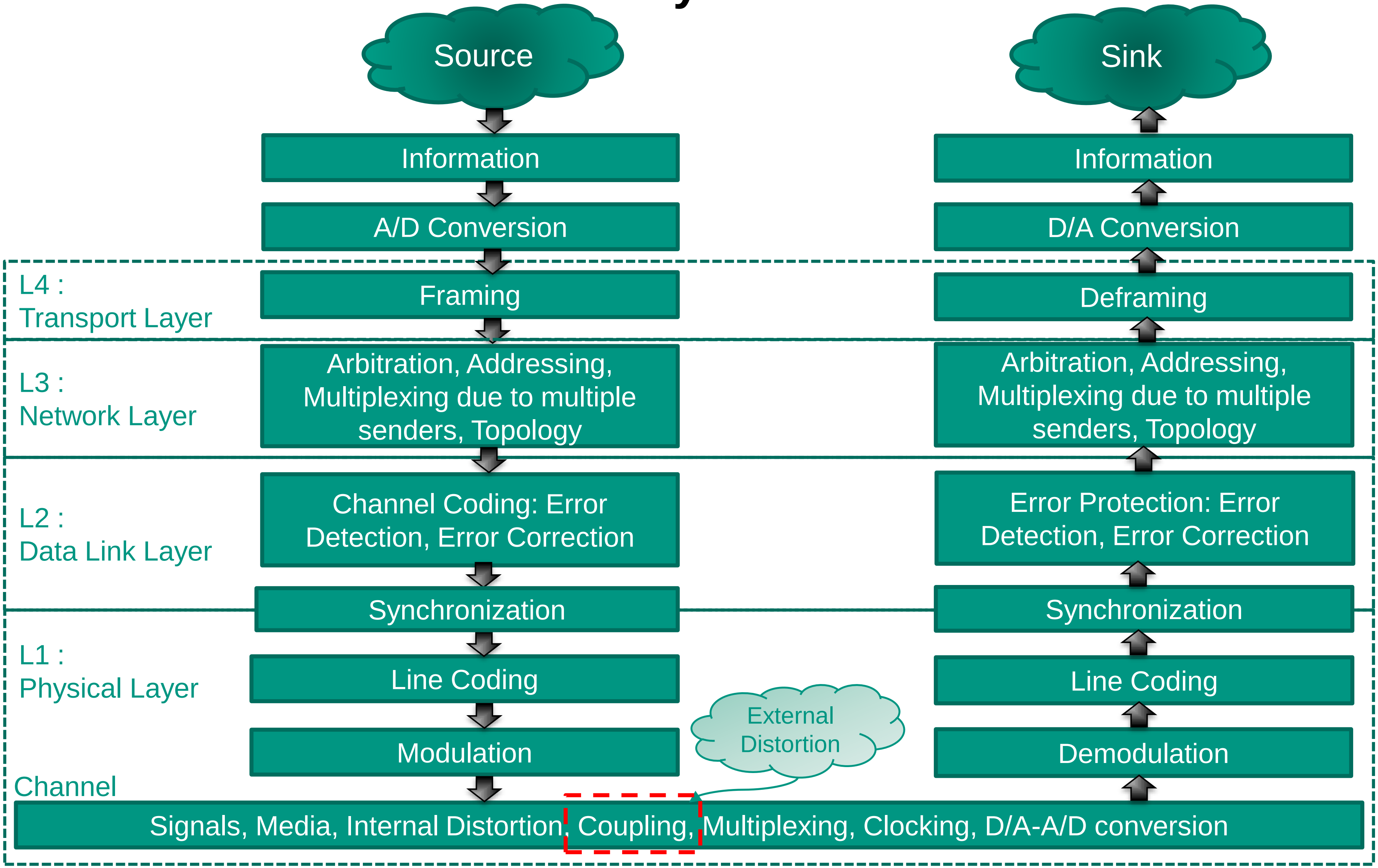
Communication Systems and Protocols

Electrical drivers

Contents

- Electrical drivers
  - Transistor Transistor Logic (TTL)
  - Tristate drivers
  - Open collector drivers
  - Emitter Coupled Logic (ECL)

Technical Communication System


Source
Information
A/D Conversion
L4 :
Transport Layer
Framing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Synchronization
L1 :
Physical Layer
Line Coding
Modulation
External Distortion
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation

Motivation electrical drivers

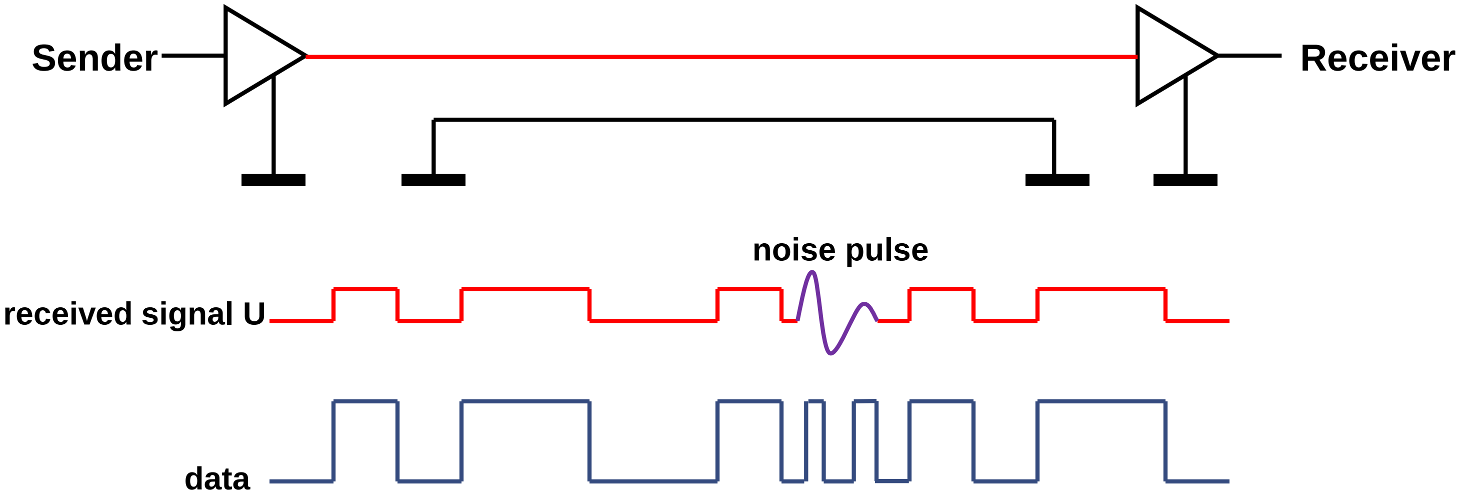
- Purpose of electrical drivers:
  - Different voltage levels for circuit and signal line
  - Higher currents needed for long signal lines
  - Adding additional noise resistance (e.g. differential signaling)

  - Busses require multiple senders and receivers to be coupled using one shared media

Single-ended signaling

- Sender generates a single voltage
- Receiver compares this signal with a fixed reference voltage
- Sender and receiver share the same ground

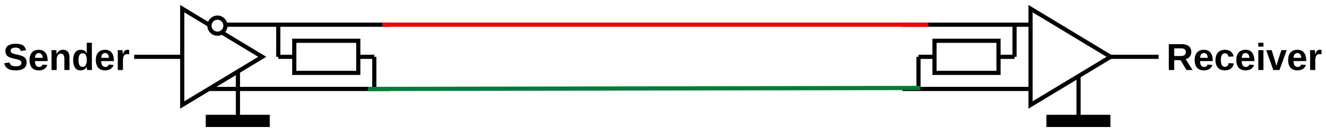
- Sensitive to noise
  - Floating ground
  - Interferences

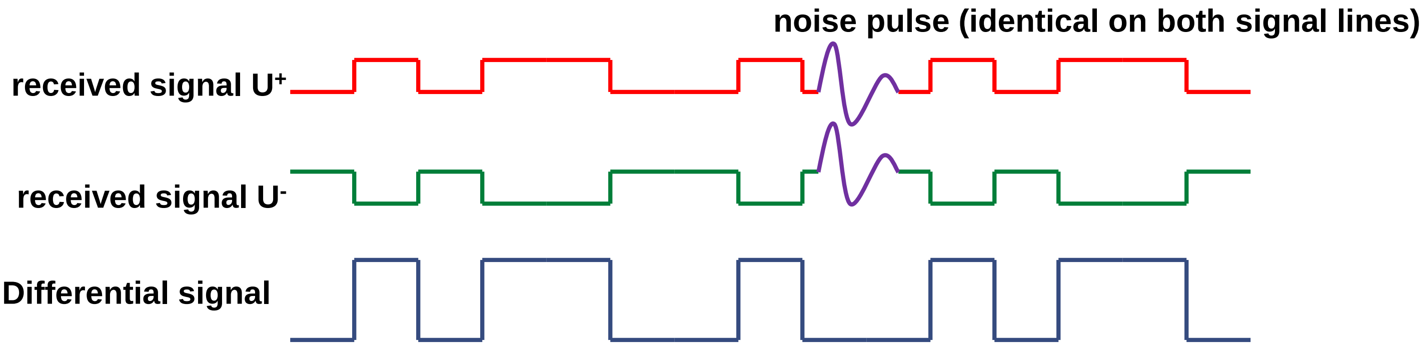


Differential signaling

- Transmission of each signal over dedicated signal paths using inverse signal voltages

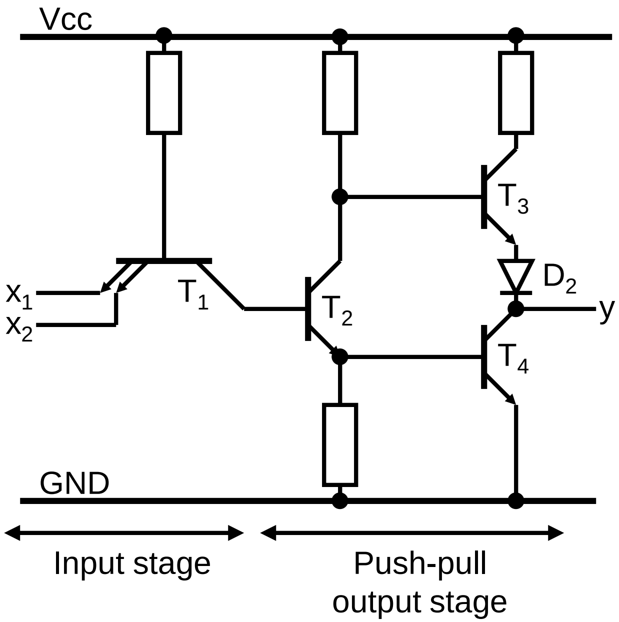
- Electric potential between signal lines is doubled
  - Longer distances or smaller voltage levels possible
- Inherent compensation of additive disturbances





Electrical Output Driver (single ended)

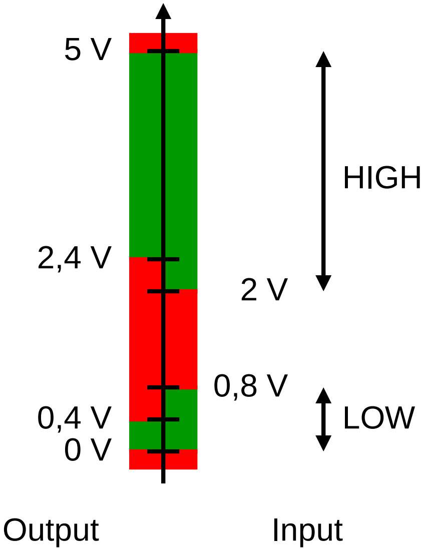
- Transistor-Transistor-Logic (TTL)
  - In earlier times the most common used technology
  - Inverted Transistor 1 at input provides high input resistance
    → low load for previous circuit
  - Output transistors T3 and T4 are only conducting one at a time
  - Advantages:
    - High current delivery at the output
    - Little energy is consumed while static



TTL basic gate (i.e. SN 7400)

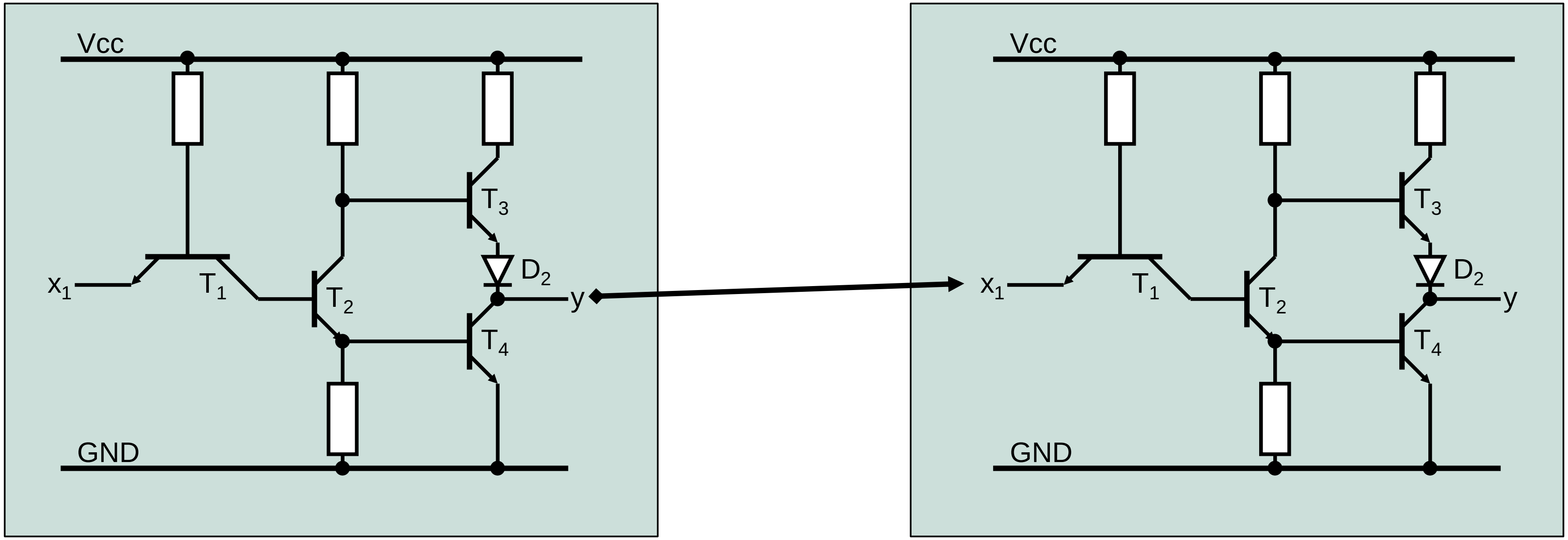
TTL levels

- No exact voltage but a voltage band is definied as HIGH or LOW signal
  - Take into account possible disturbances
- Valid HIGH and LOW areas are wider at the input due to possible voltage drops on the lines
- Asymmetric division since the HIGH level drops under load which does not happen to the LOW level



TTL levels

Transmission Setup with TTL Technology

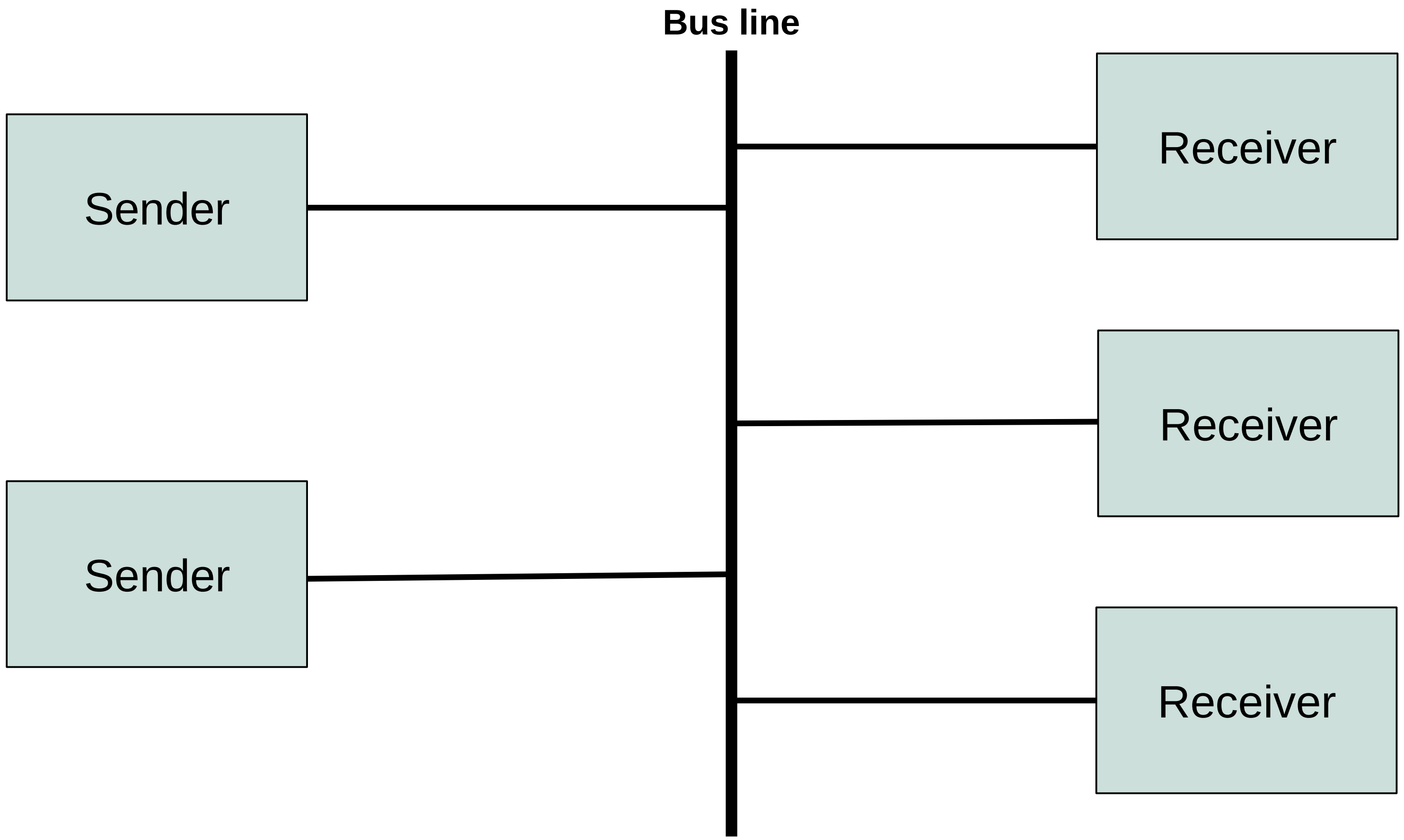


Sender

Receiver

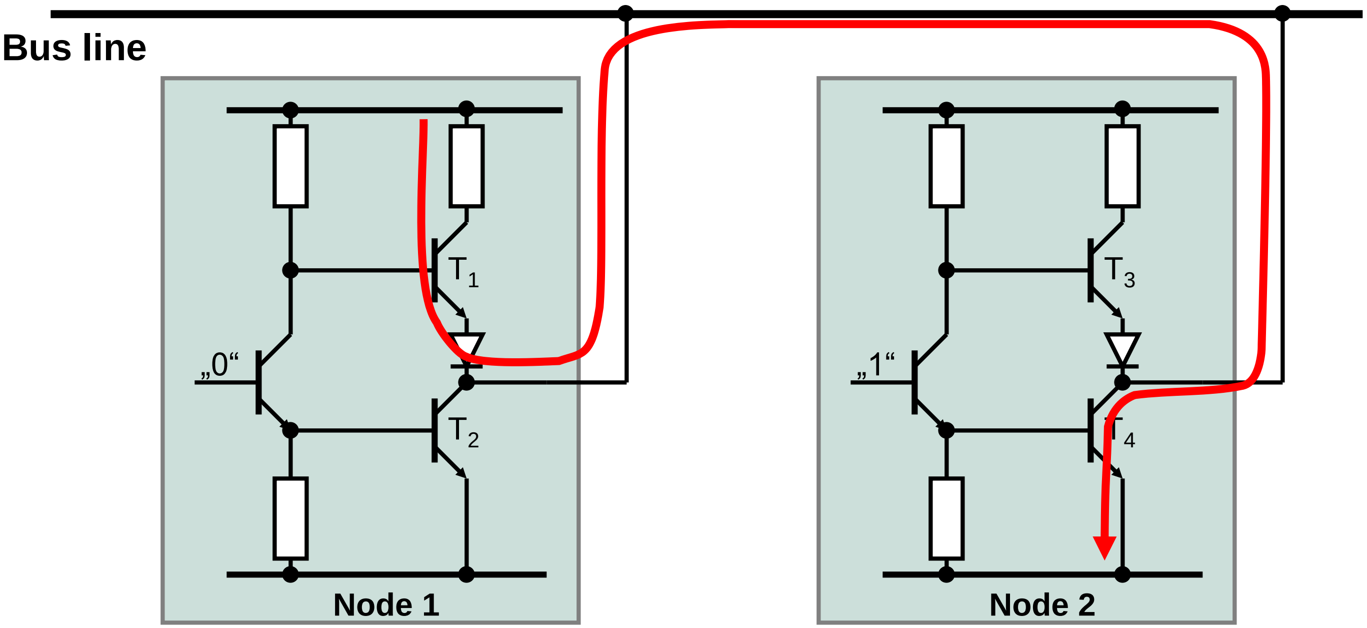
Idea of a Bus System

- Multiple Senders and receivers are connected together to the same channel for communication



Connection of multiple TTL drivers

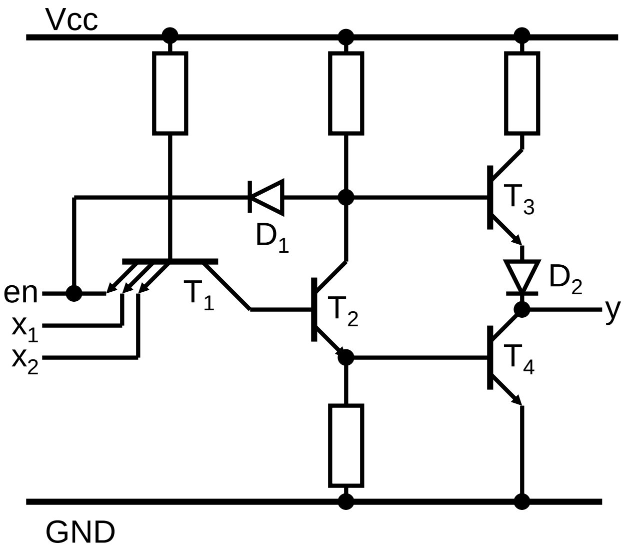
- Danger for short-circuit if the outputs of several TTL gates are connected to each other
  - If one gate is on HIGH level and another on LOW level a short-circuit current can flow via the open transistors (T1 and T4 in the figure)
- Modification of the bus coupling circuit is necessary when using driver on bus lines!



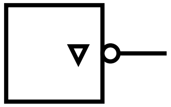
Solution 1: Tristate driver

- Parallel switching of several outputs is not possible with simple TTL gates → danger of short-circuit
- By turning off unused output stages, several of such circuits can be connected to one bus
- Third state (Tristate): high-resistance (Z)
- In each case only one output is allowed to drive the bus, all others must show high-resistance

Example: NAND gate



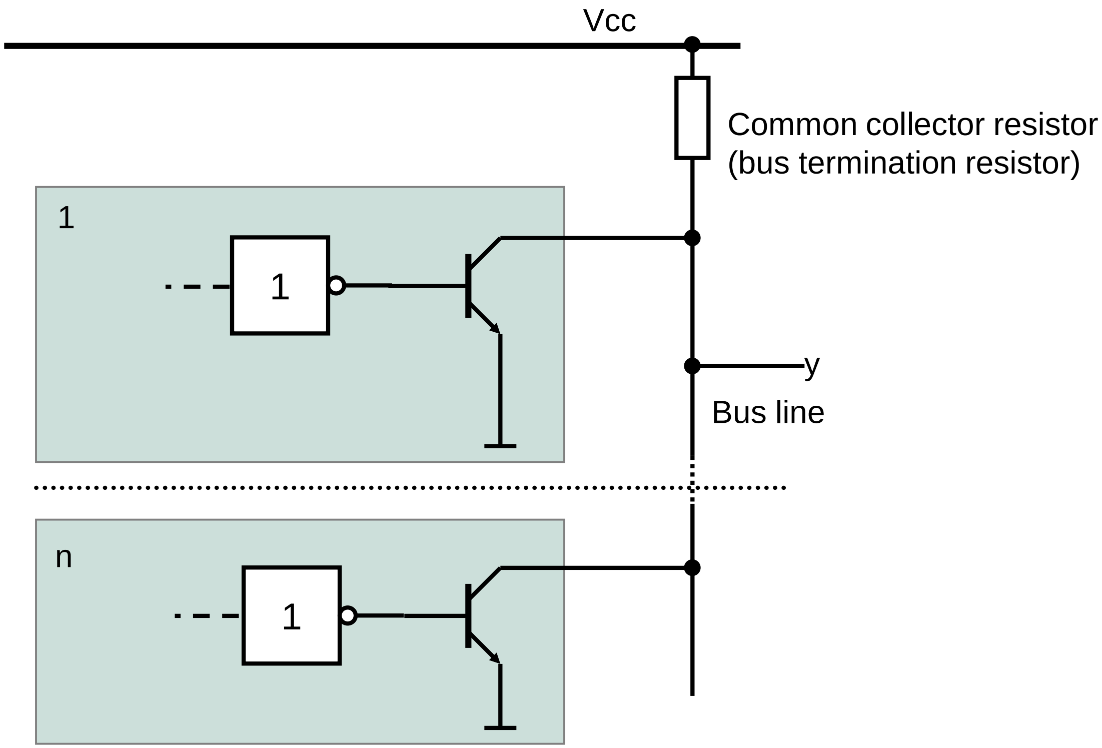
en	x_1	x_2	y
HIGH	LOW	LOW	HIGH
HIGH	LOW	HIGH	HIGH
HIGH	HIGH	LOW	HIGH
HIGH	HIGH	HIGH	LOW
LOW	-	-	Z



Switch symbol of a Tristate output

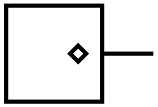
Solution 2: Open collector bus

- The collector of each of the output transistors remains unconnected
- All bus members share one collector resistor
- The output is only HIGH if all transistors cut off
  - low value (GND) on the bus line is dominant value



Common collector resistor
(bus termination resistor)

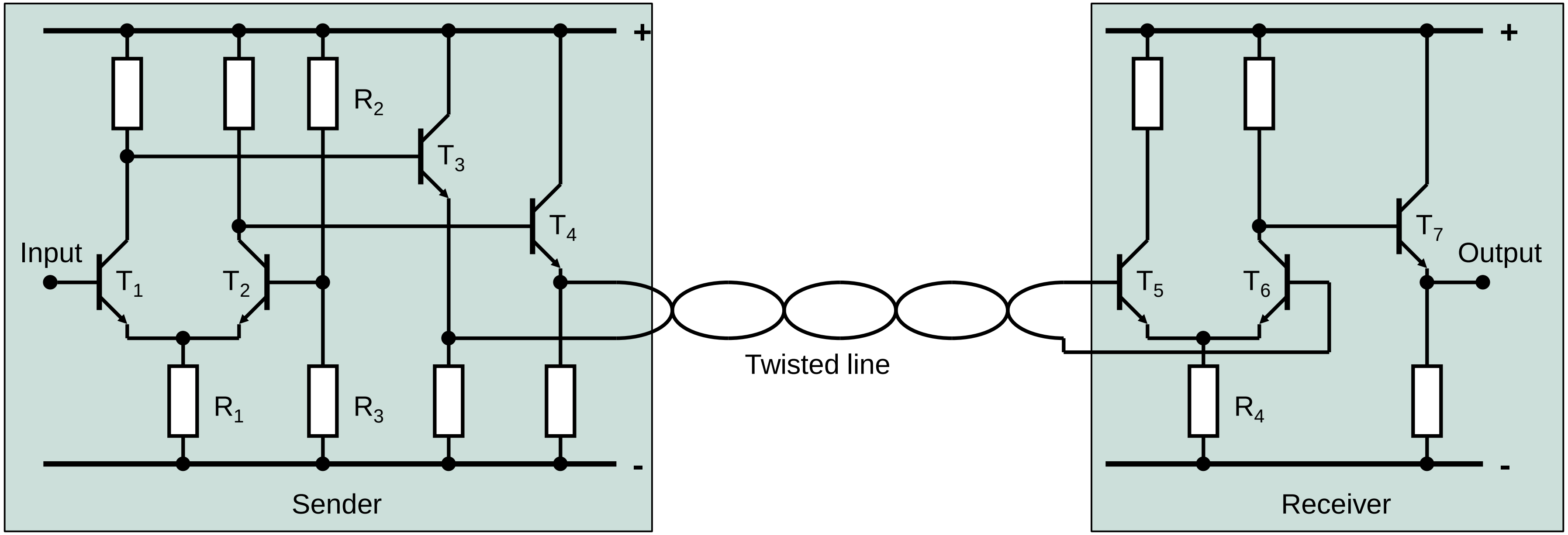
Bus line



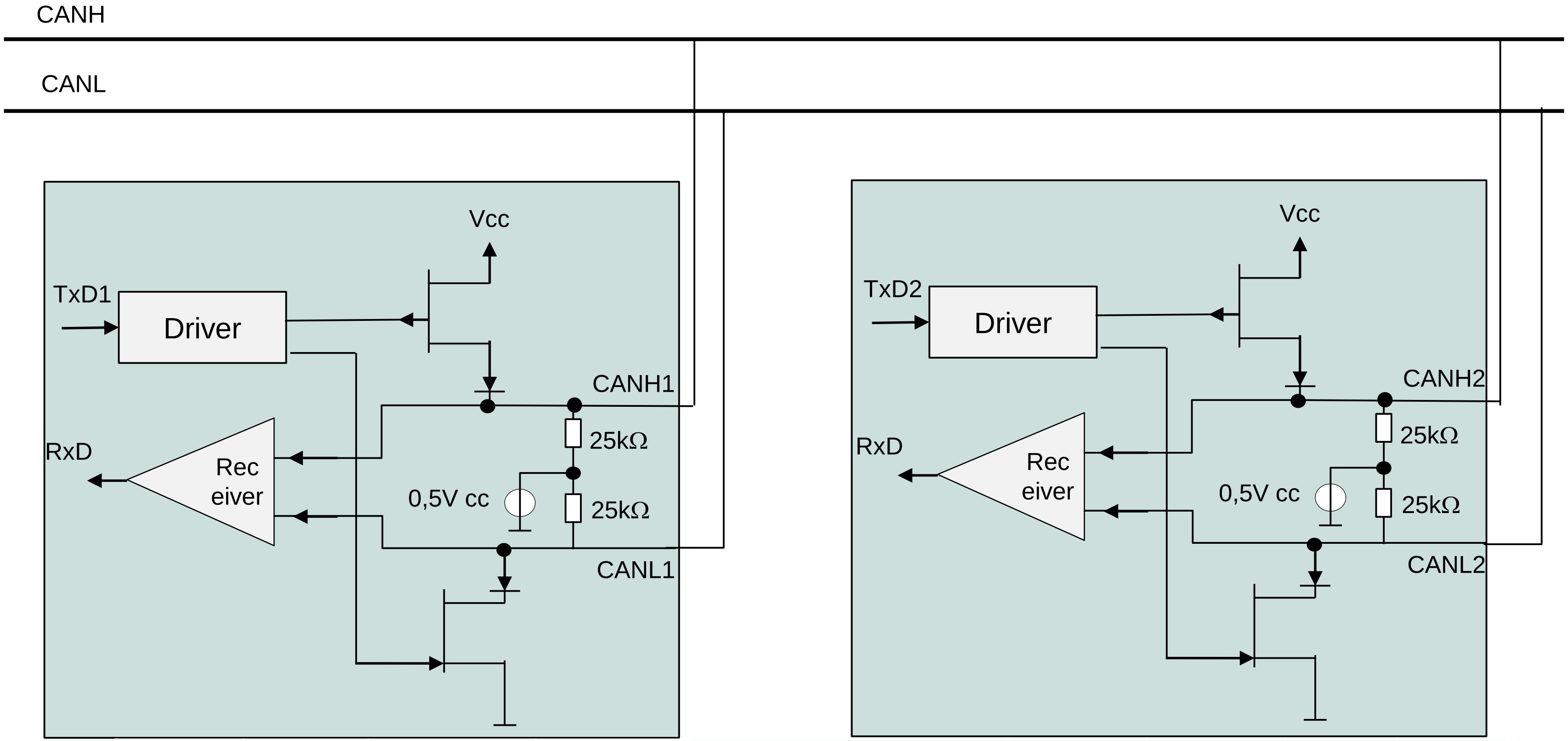
Switch symbol of a
open collector output

Emitter Coupled Logic (ECL) (differential signal)

- Higher speed since transistors do not go to saturation
- With the help of the asymmetric input signal a differential signal for transmission is generated
- The differential amplifier inside the receiver reconstructs the input signal



Example for differential bus driver (CAN-Bus)



TxD1	TxD2	CANH1	CANL1	CANH2	CANL2	CANH	CANL	Diff	RxD
1	1	0.5	0.5	0.5	0.5	0.5	0.5	0	1
1	0	0.5	0.5	Vcc	Gnd	Vcc	Gnd	Vcc	0
0	1	Vcc	Gnd	0.5	0.5	Vcc	Gnd	Vcc	0
0	0	Vcc	Gnd	Vcc	Gnd	Vcc	Gnd	Vcc	0

Figure Reference: CAN Transceiver, 2010-2017 Vector Informatik GmbH

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

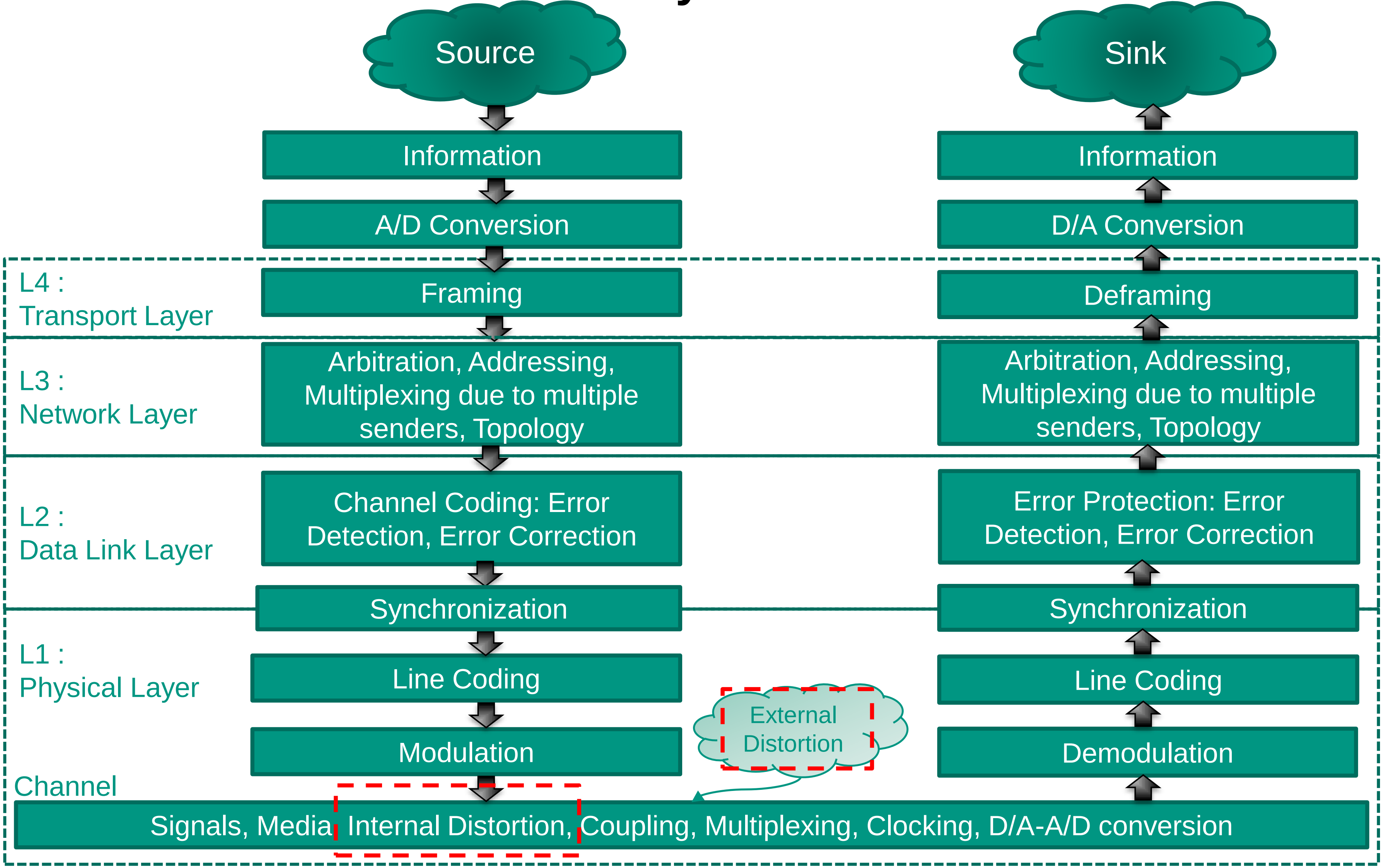
Line Model, Signal Distortions

Contents

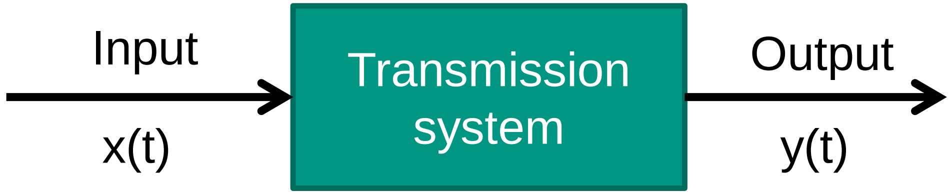
- Line Model
  - Telegrapher‘s equation
  - Reflection on wires

- Signal Distortions
  - Reflections
  - Attenuation
  - Bandwidth

Technical Communication System

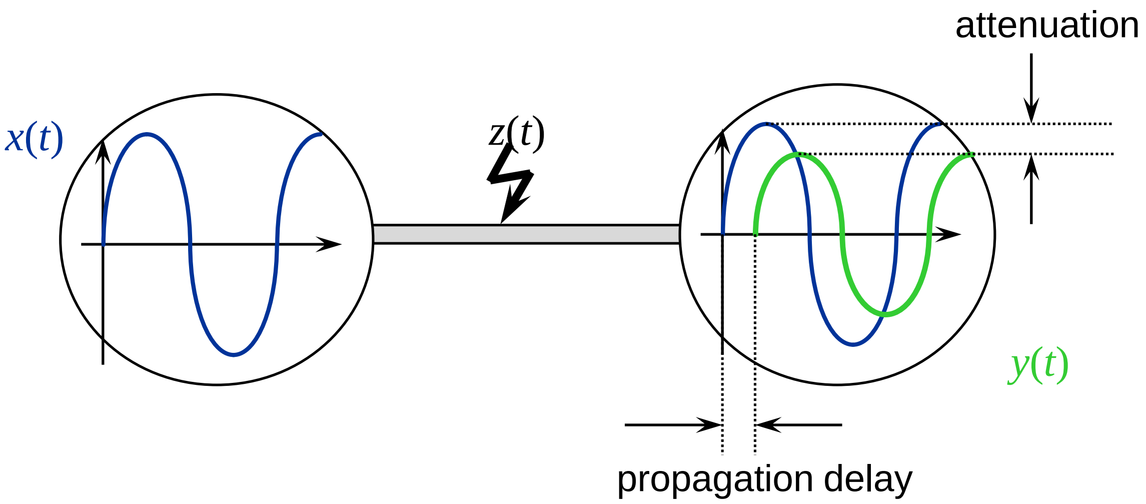

Source
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
Channel
External Distortion
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Signal Transmission over a Medium



$y(t) = F(x(t), z(t))$

- with
  - *x*(*t*): Input signal
  - *y*(*t*): Output signal
  - *z*(*t*): Interference

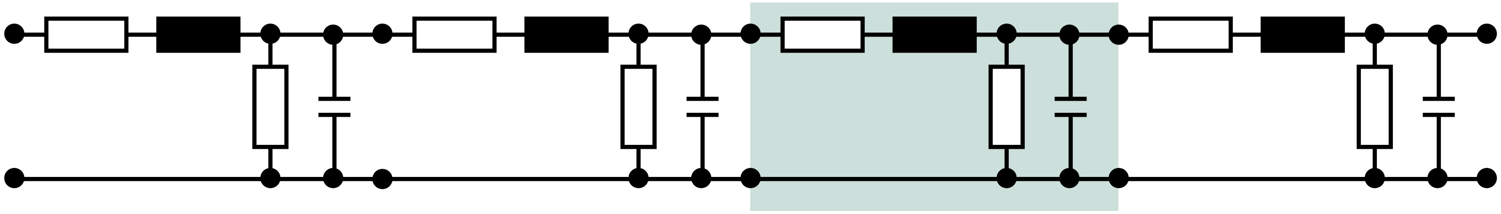


- Interferences can be caused by external sources as well as by media itself

Model of Signal Lines

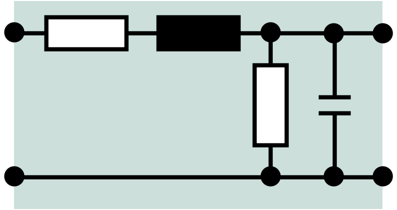
- Problem: When taking into account wave effects, signal lines can not be described by single elements any more

- Solution: Description using a distributed waveguide model → cascade of RLC quadripole



- Each segment represents an infinitesimally short segment of a transmission line

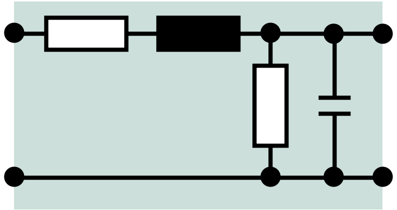
Components of the line model



- Distributed resistance **R’**, caused by resistance of the conductors is represented by a series resistor
- Distributed inductance **L’**, caused by the magnetic field around the wires is represented by a series inductor
- Distributed capacitance **C’** between the conductors is represented by a shunt capacitor
- Distributed conductance **G’**, caused by the dielectricum between the conductors is represented by a shunt resistor between the conductors

Idea: Derivation of the telegrapher‘s equation

- Boundary conditions: long, straight signal line, propagation in x-direction



- Using the laws of Kirchhoff, one can describe this system using the following equations:

$$\frac{\partial U(x,t)}{\partial x} = -L'(x)\frac{\partial I(x,t)}{\partial t} - R'(x)I(x,t)$$
$$\frac{\partial I(x,t)}{\partial x} = -G'(x)U(x,t) - C'(x)\frac{\partial U(x,t)}{\partial t}$$

Solution of the telegrapher‘s equation

- Under the assumption of constant distributed elements one can find:

$$U(x,t) = u_1 e^{i\omega t - \gamma x} + u_2 e^{i\omega t + \gamma x}$$
$$I(x,t) = i_1 e^{i\omega t - \gamma x} + i_2 e^{i\omega t + \gamma x}$$

with $\gamma = \sqrt{(R' + i\omega L')(G' + i\omega C')}$

- In the lossless case (R’=0, G’=0) the equations can be simplified:

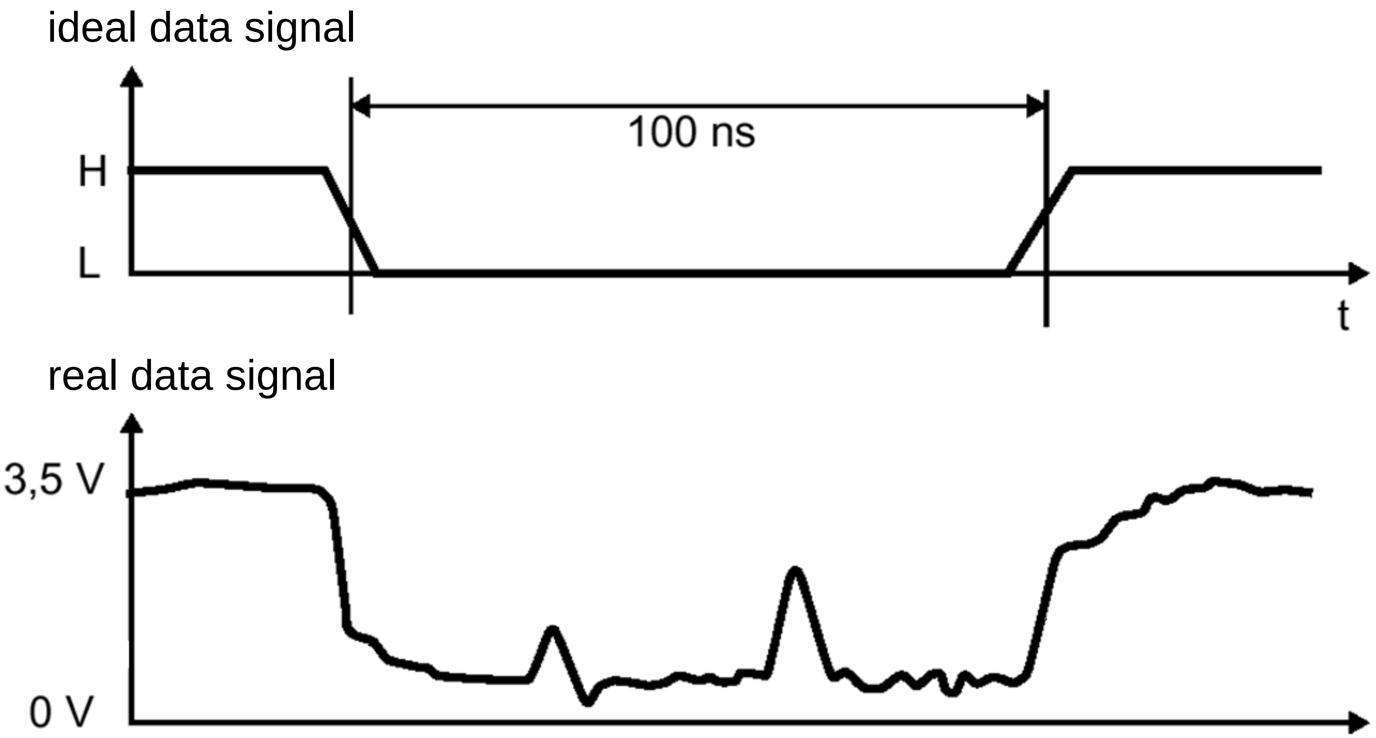
$$U(x,t) = u_1 e^{i\omega t - \gamma x} + u_2 e^{i\omega t + \gamma x}$$
$$I(x,t) = \frac{u_1}{Z_0} e^{i\omega t - \gamma x} - \frac{u_2}{Z_0} e^{i\omega t + \gamma x}$$

with $Z_0 = \sqrt{\frac{L'}{C'}}$, that is called **characteristic impedance**

→More details see lecture “Grundlagen der Hochfrequenztechnik”

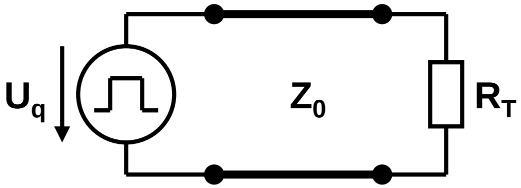
Possible signal distortions

- **Reflections**
- Attenuation
- Band limitation
- Glitches
- Noise
- Cross Talk



Reflection at the wire end

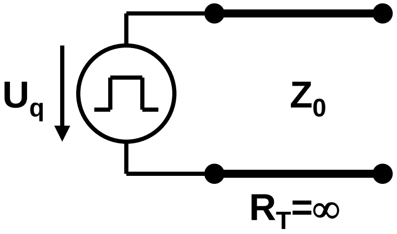
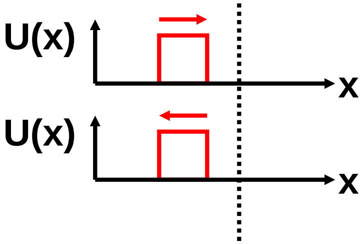
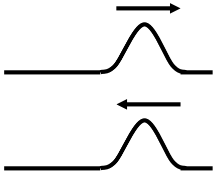
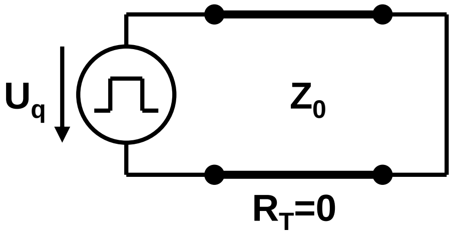
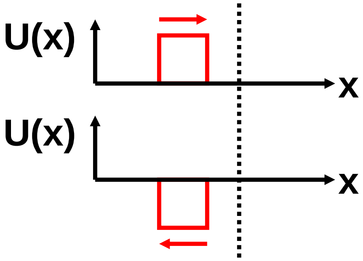
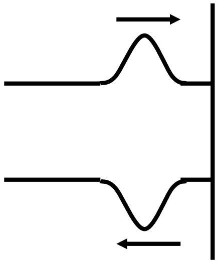
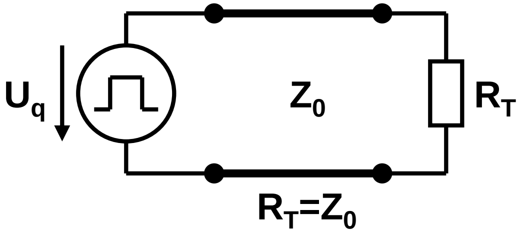
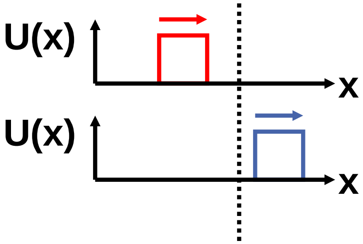
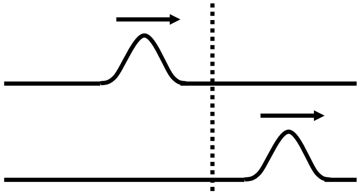
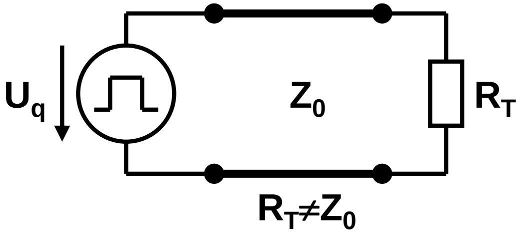
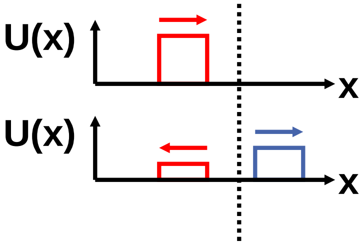
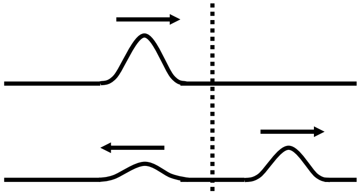
- Signals spread wavelike within long lines/wires
- There exists a forward and eventually a backward running wave that interfere on the wire
- The amplitude of the backward running wave depends on
  - The impedance of the line
  - The terminating impedance at the end of the line



- Definition of the reflection factor: $r = \frac{R_T - Z_0}{R_T + Z_0}$

R_T: Terminal Resitance
Z_0: Characteristic impedance

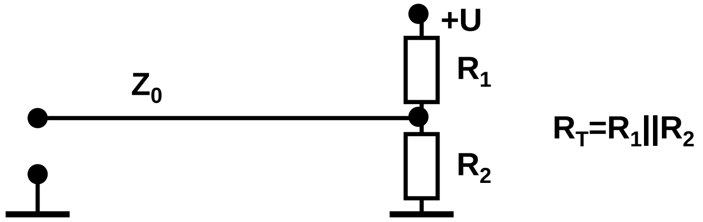
Reflection on wires

Line	Reflection factor r	Voltage curve	mechanical analogue
	r=1		 loose end
	r=-1		 solid end
	r=0		 Coupling joint
	-1<r<1		 Coupling joint

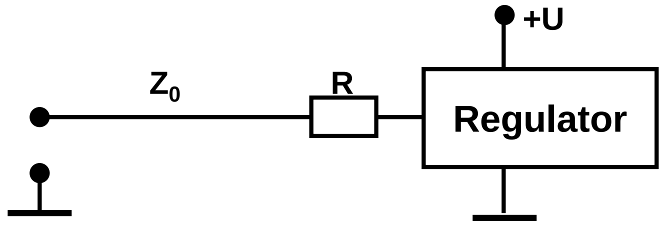
Line terminations

- Termination by using the characteristic wave resistance Z_0 to avoid reflections
- The utilization of several resistors switched in parallel is advantageous since usage of resistors of a relatively high value which provide for a lower DC charge allow for a approximation to the relatively low resistance of the line

- Asymmetric line



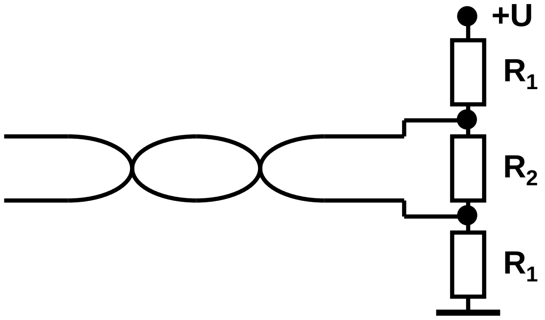
passive



active

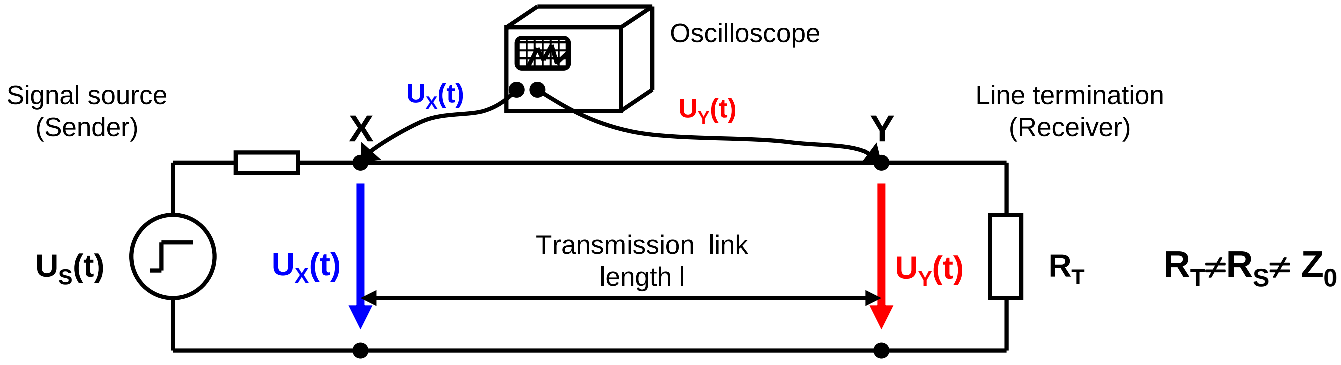
Consists of voltage regulators to maintain voltage of the terminating resistor at a constant level

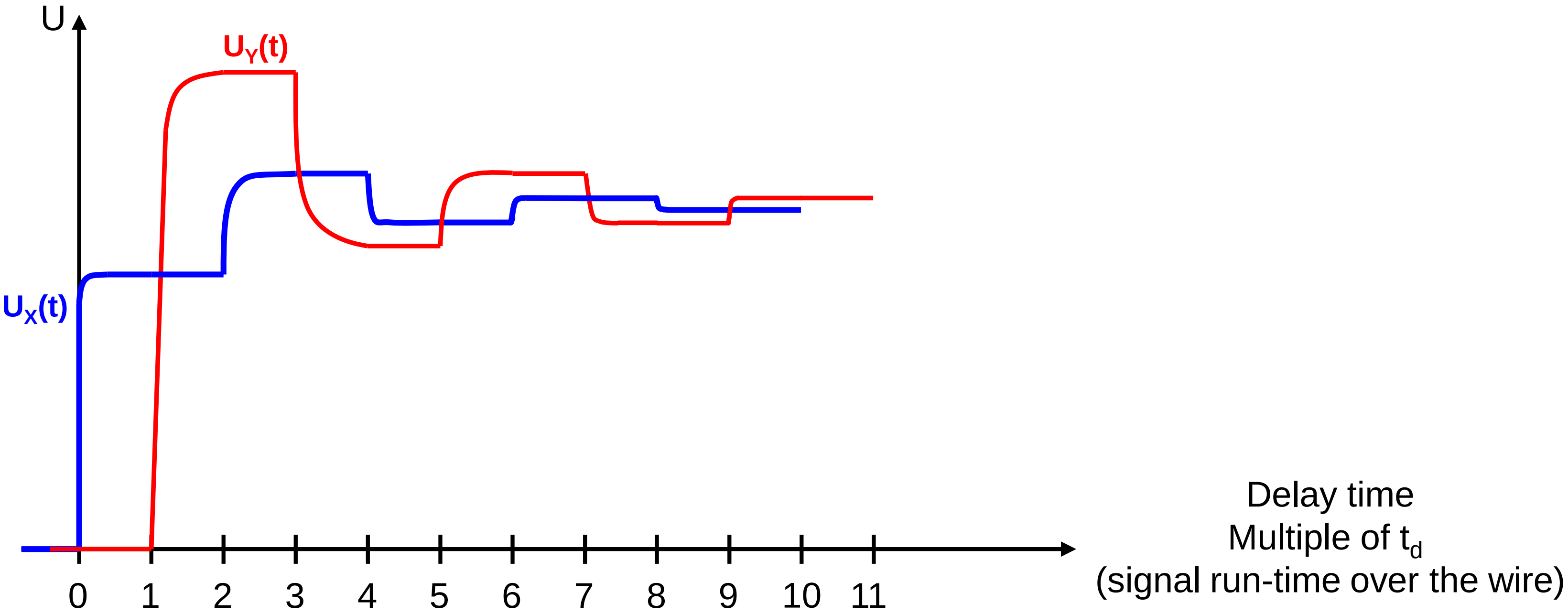
- Symmetric line



Outlook to exercises: Reflection on wires

- Measurement setup for the determination of the reflection on the lines:

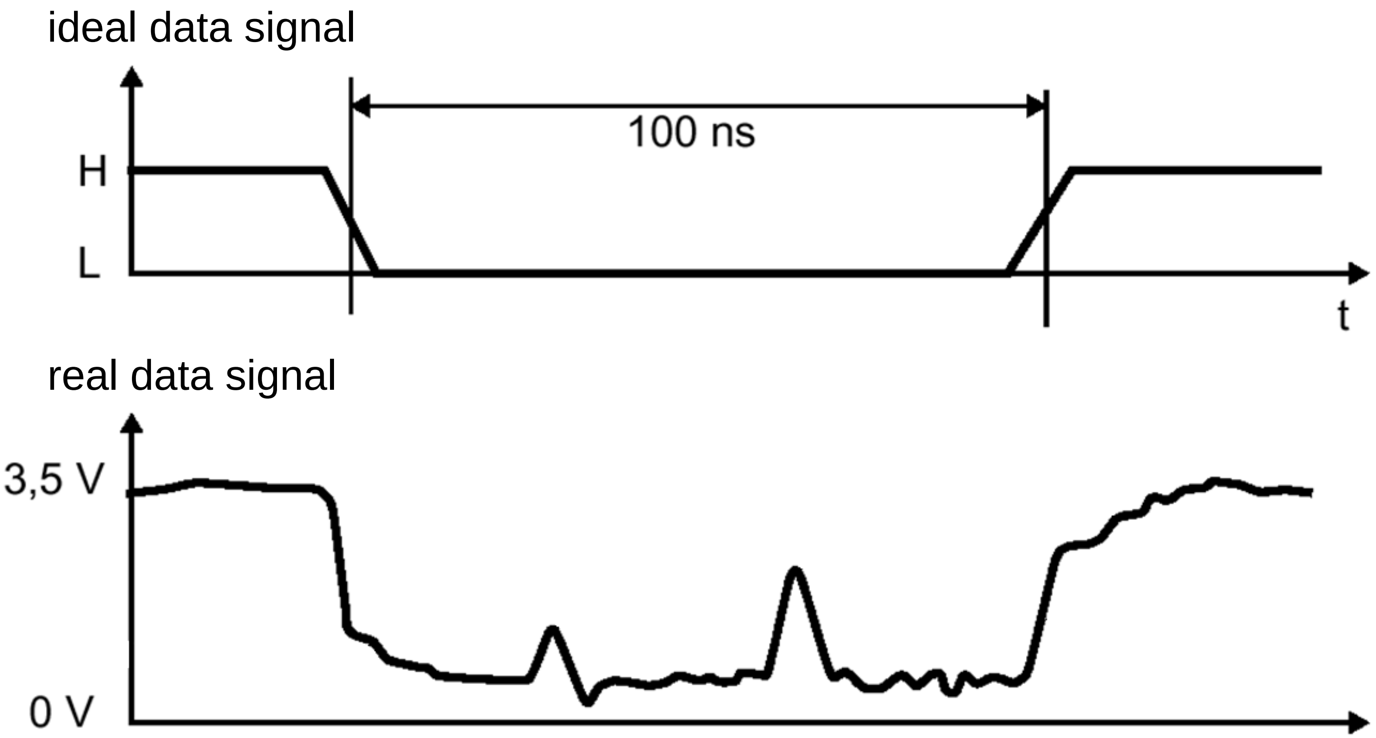




Possible signal distortions

- Reflections
- **Attenuation**
- Band limitation

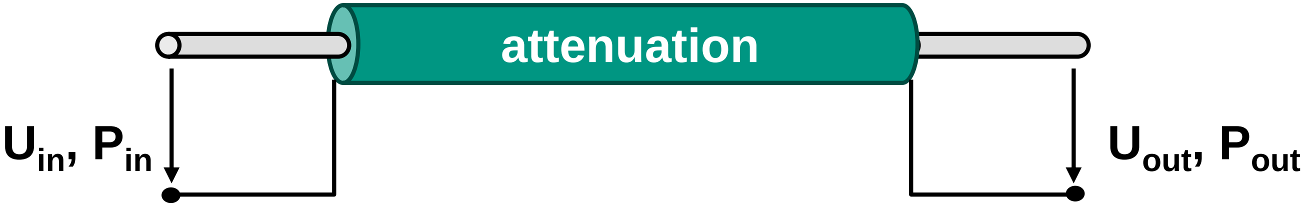
- Glitches
- Noise
- Cross Talk



Attenuation (media dependend)

- Attenuation of signals is caused by:
  - Ohmic loss
  - Dielectric loss
  - Radiation loss
  - Reflections

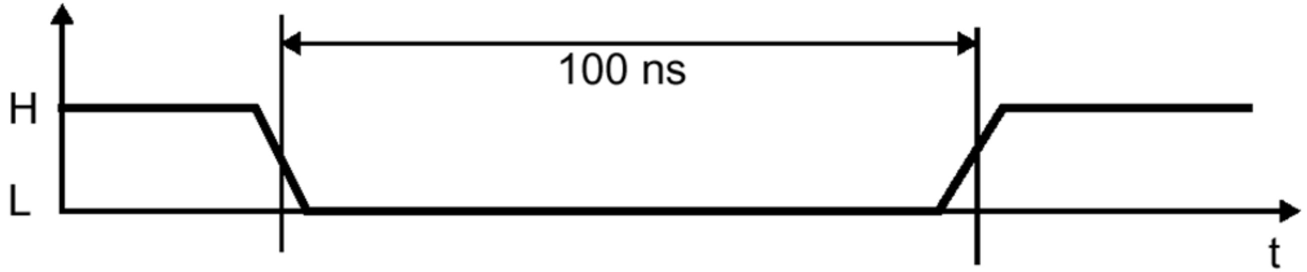
- Attenuation $G = 10\log\left(P_{in}/P_{out}\right) = 20\log\left(U_{in}/U_{out}\right)\left[\mathrm{dB}\right]$
  - additive properties of single cable sections
  - Depends on length/distance
  - Depends on frequency of transmitted signal



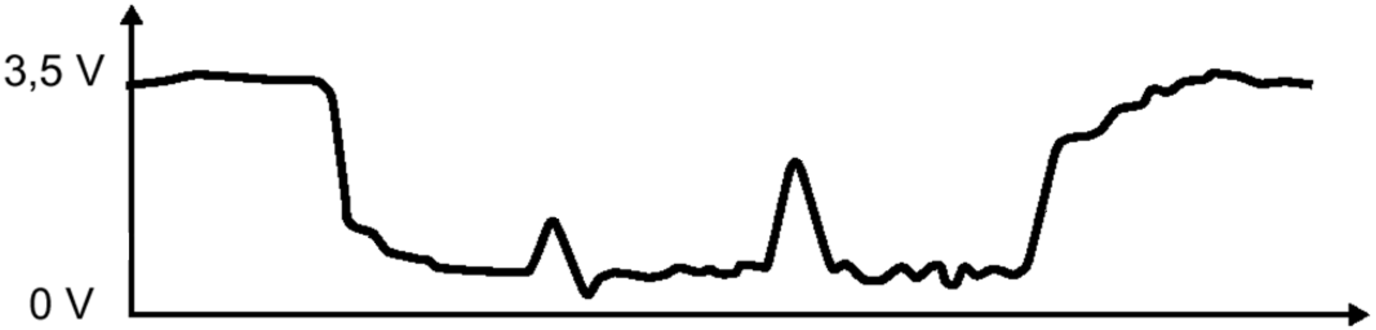
Possible signal distortions

- Reflections
- Attenuation
- **Band limitation**
- Glitches
- Noise
- Cross Talk

ideal data signal

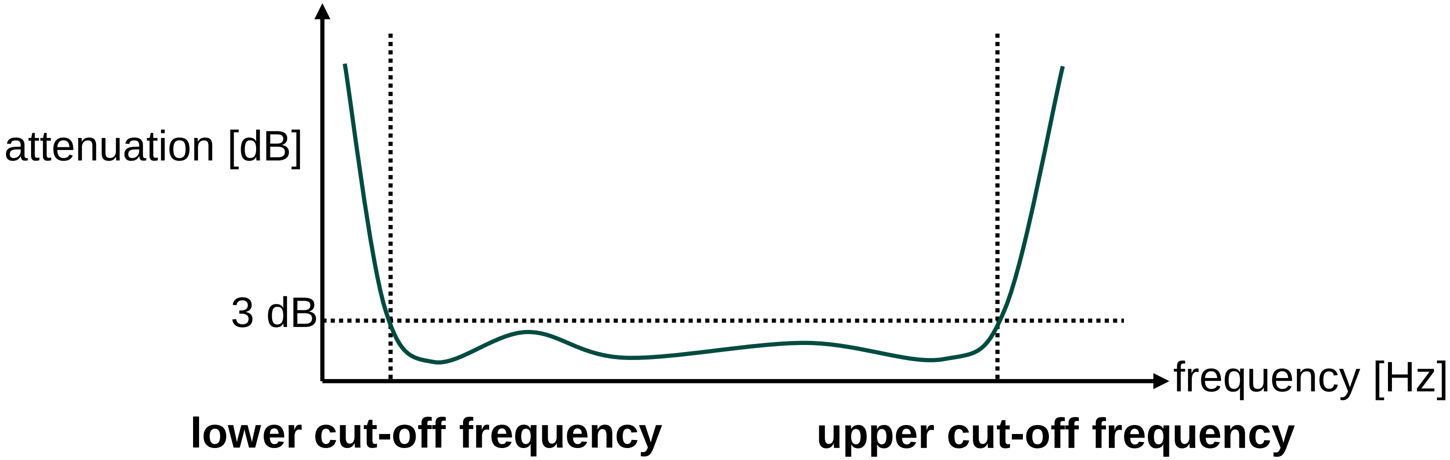


real data signal



Bandwidth

- Between the frequencies almost uniform attenuation

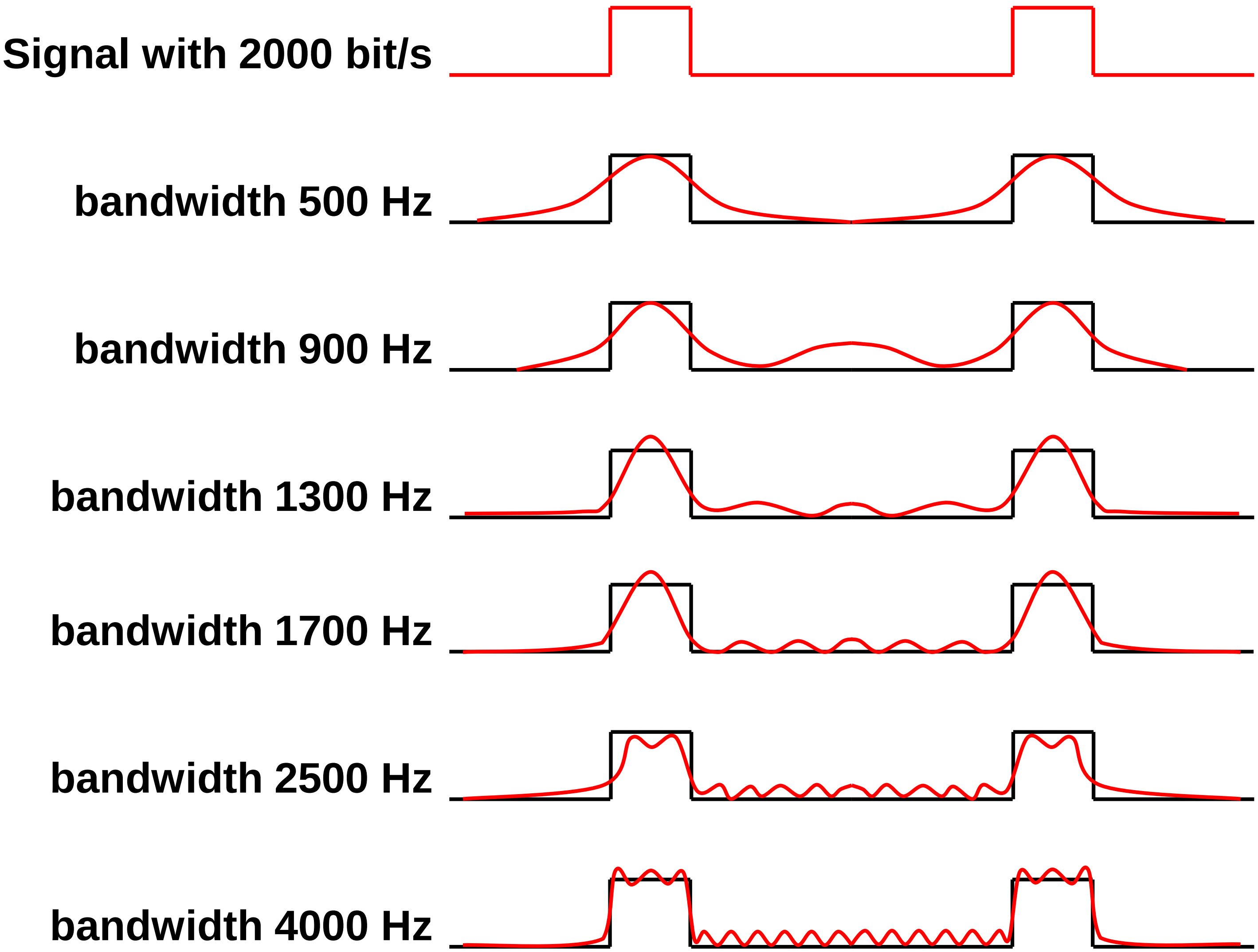


- Limits not ideal
  - No vertical signal edges
  - Frequencies are attenuated at different degrees

Bandwidth in Hertz

- Cut Off frequency
  - Frequency at with the signal amplitude has dropped by 3dB compared to the input value
- Difference between the highest and lowest frequencies in a composite signal or range of frequencies a channel can pass.
- Passband bandwidth : Interval between upper and lower cut-off frequency

Influence of the Bandwidth on a digital Signal



Signal with 2000 bit/s

bandwidth 500 Hz

bandwidth 900 Hz

bandwidth 1300 Hz

bandwidth 1700 Hz

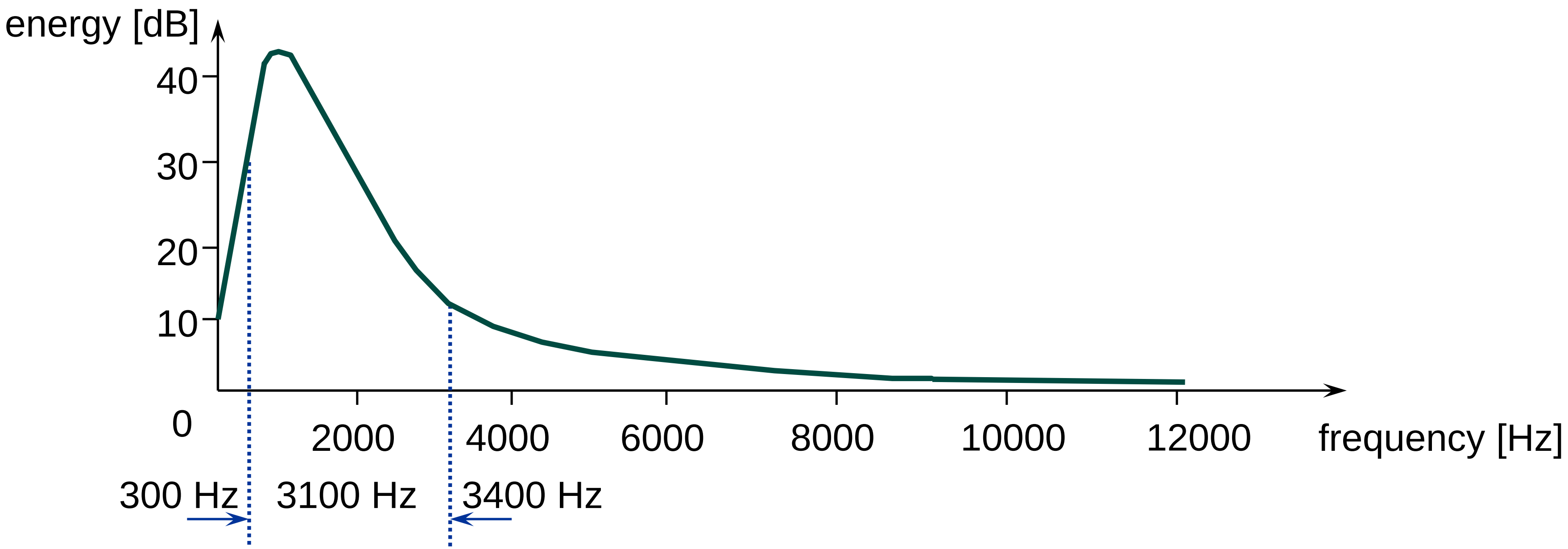
bandwidth 2500 Hz

bandwidth 4000 Hz

source: University Ulm

Example: Frequency Spectrum of a Signal

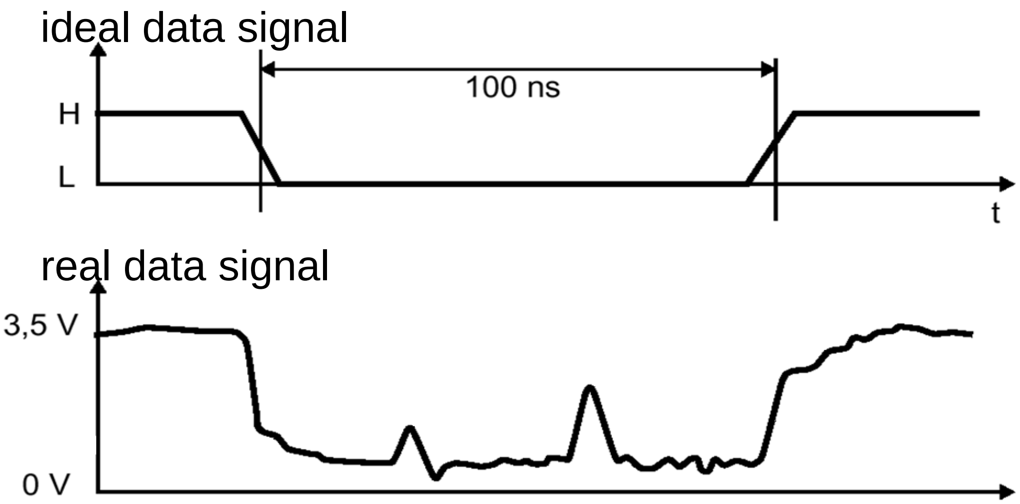
- Band limited signal: signals can comprise a “naturally” limited frequency spectrum or can be limited to a section of a spectrum through technical means (bandwidth)
  - Example: ITU-Standard telephone channel
    Frequency spectrum of the human voice



Possible Signal Distortions

- Reflections
- Attenuation
- Band limitations

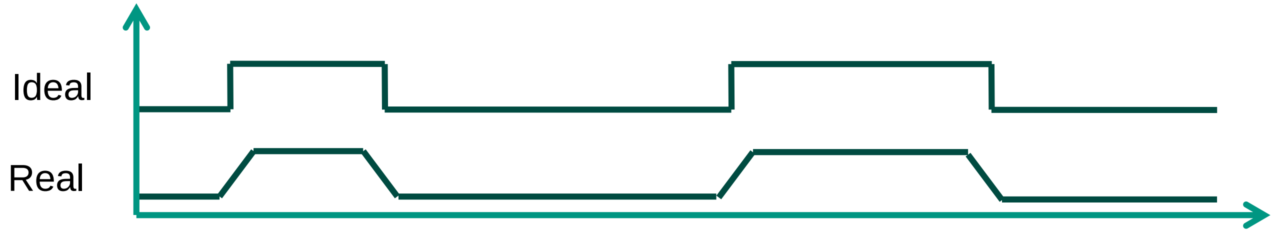
- **Glitches**
- **Noise**
- **Cross Talk**



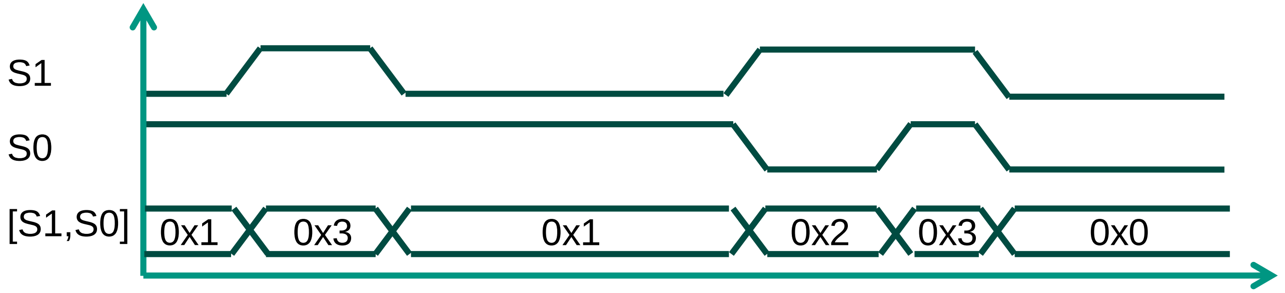
- Glitches
  - An electric pulse of a short duration. Sometimes it can be an undesirable result of a fault or design error.
- Noise
  - Several types like thermal noise, induced noise (motors and appliances), may corrupt the signal.
- Cross talk
  - Effect of one wire over another. One wire acts as a sending antenna and the other as the receiving antenna.

Excursus: Representation of Signals

- Ideal vs. realistic edges
  - Raise and fall time is approximated by slanted edges

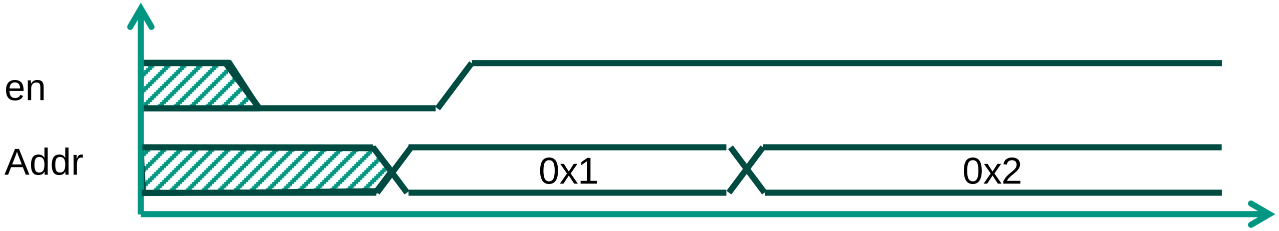


- Combination of individual lines (busses)
  - Signal is drawn as a ribbon with the combined value as text
  - Every change on one of the signals results in a change of the bus



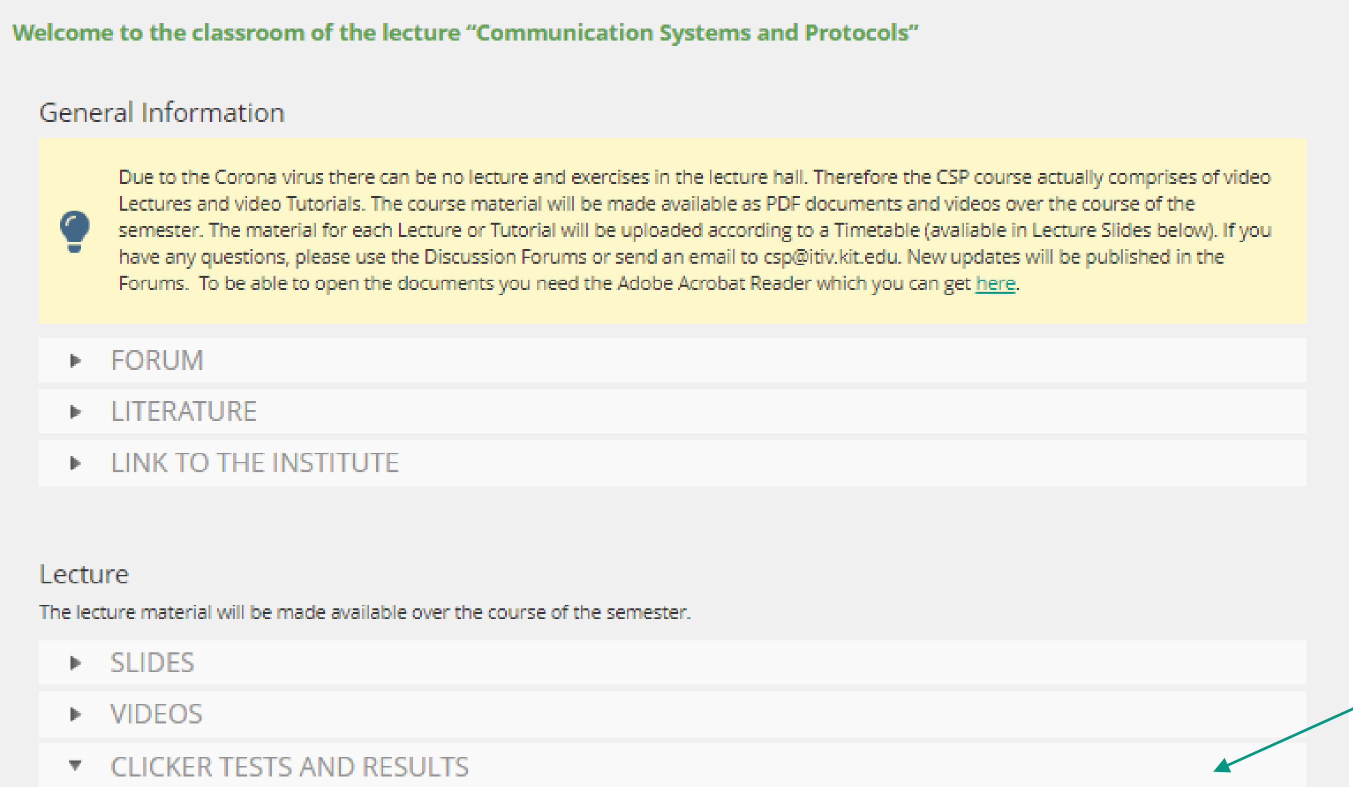
Excursus: Representation of Signals II

- Undefined value
  - In specifications, a signal might not be determined for every point in time
  - Undefined region is depicted as shaded area



- Inverted Logic (LOW active)
  - Signals with inverted logic can be marked as follows:
    - nSignal
    - *Signal
    - $\overline{\text{Signal}}$

Participate in Clicker Test 2b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Sampling

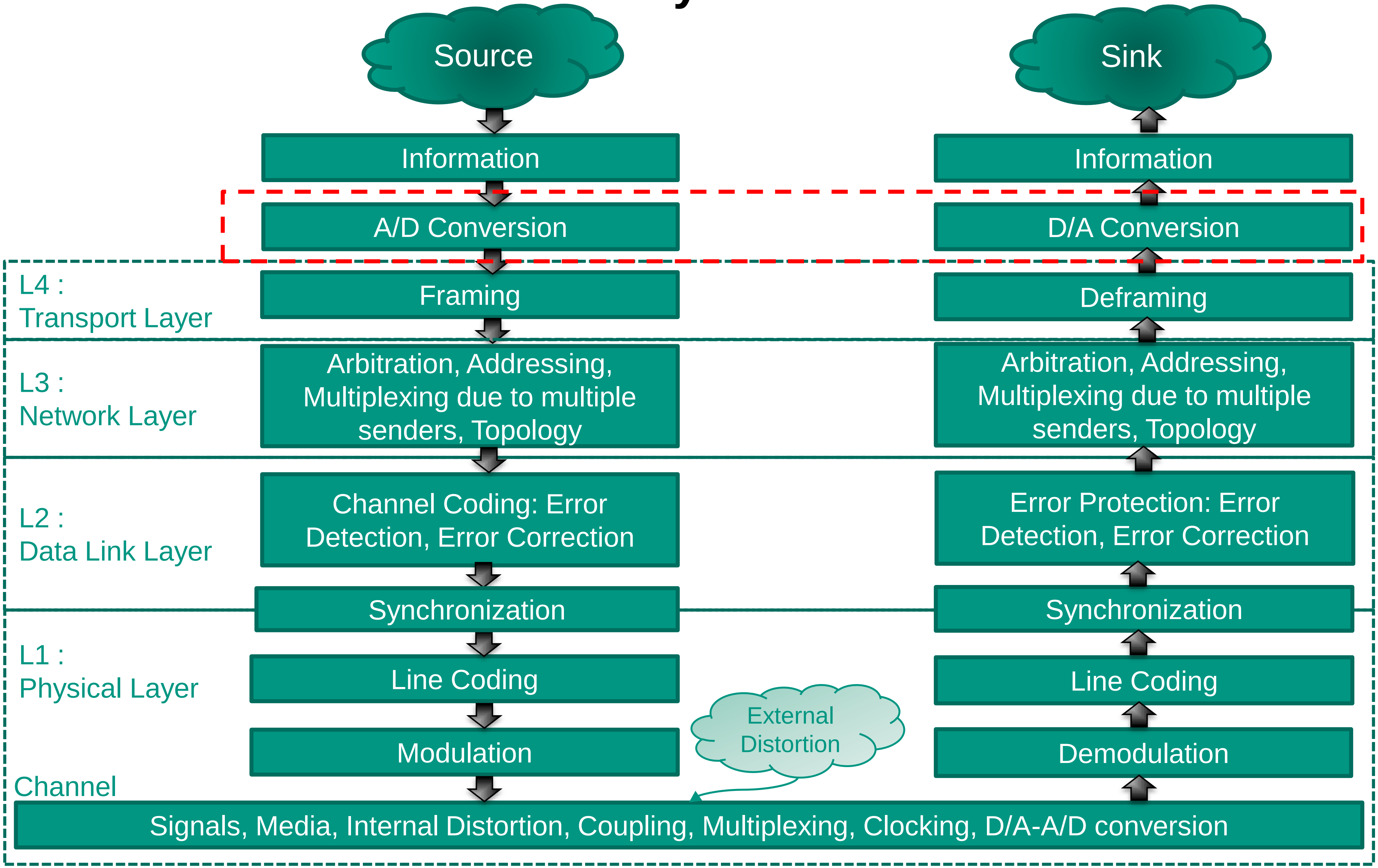
Contents

- Signal Classes

- Analog-Digital conversion
  - Sample & Hold
  - Digitalization
  - Discretization

- Oversampling of digital signals

Technical Communication System



Source

Information

A/D Conversion

L4 : Transport Layer

Framing

L3 : Network Layer

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

L2 : Data Link Layer

Channel Coding: Error Detection, Error Correction

Synchronization

L1 : Physical Layer

Line Coding

Modulation

Channel

Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

External Distortion

Sink

Information

D/A Conversion

Deframing

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

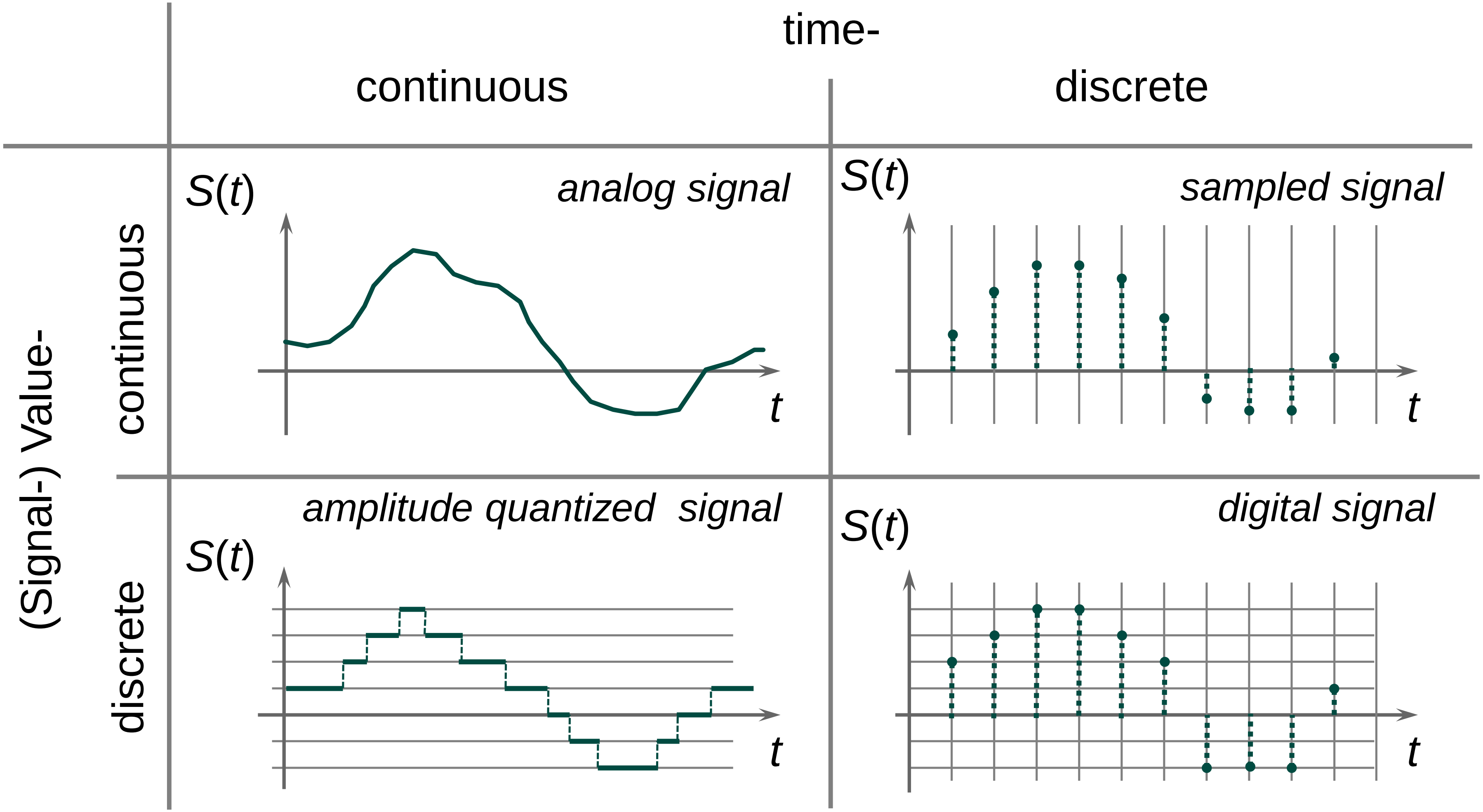
Error Protection: Error Detection, Error Correction

Synchronization

Line Coding

Demodulation

Classes of Signals



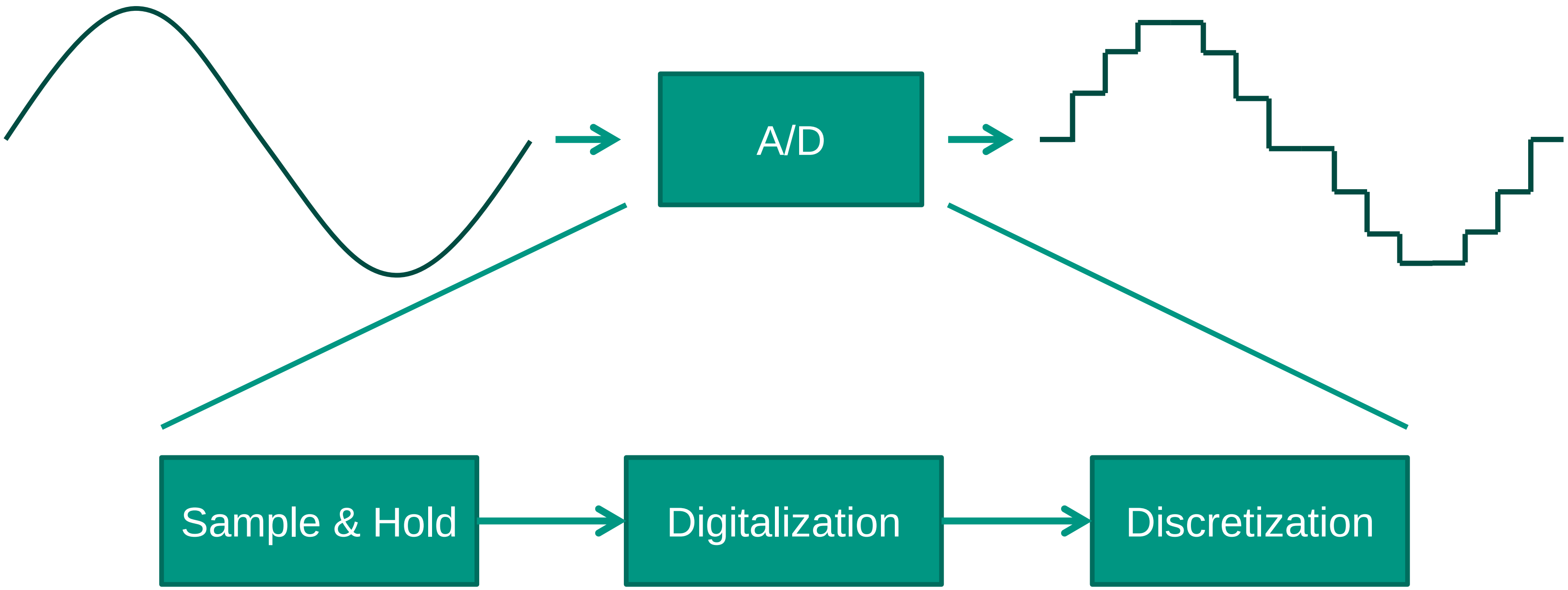
time-

continuous | discrete

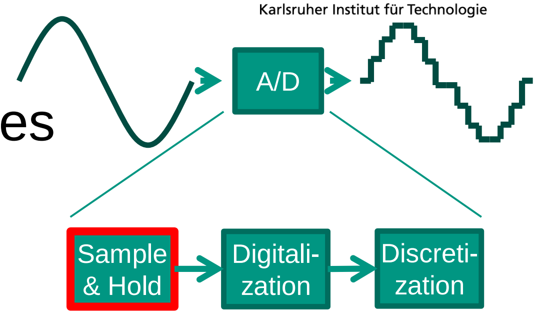
(Signal-) Value-

continuous | discrete

Analog-Digital Conversion



Sampling of Signals



- Transforming a continuous signal into discrete values
  - Signal shall be reconstructable unambiguously
- Prerequisites:
  - Signal limited in frequency with a maximum frequency f_{max}
  - Equidistant sampling points
- Nyquist-Shannon-Sampling Theorem

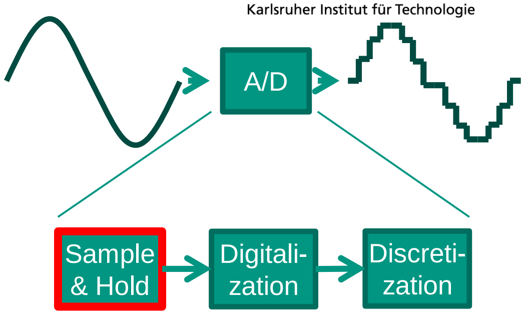
  A continuous signal has to be sampled with a frequency more then double the maximum frequency of the signal itself f_{max} in order to be able to unambiguously reconstruct the original signal.

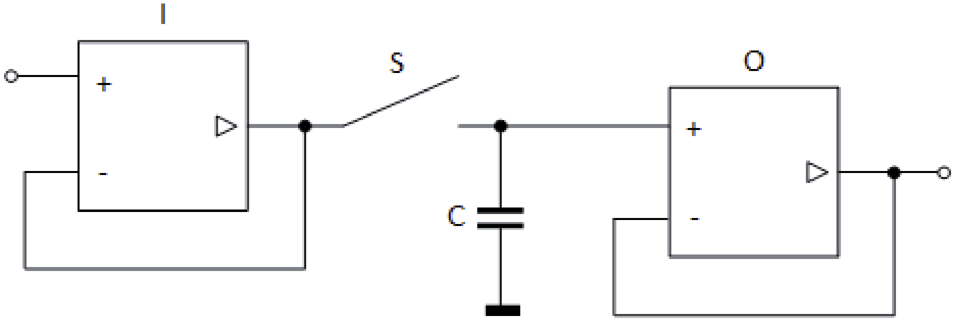
$$f_{Sample} > 2 \cdot f_{max}$$

To keep this condition it may be necessary to filter the signal prior to sampling with a low-pass filter. This prevents artefacts otherwise caused by high frequencies.

Sample & Hold

- Analogue signals change over time
- Analogue signals have to be kept constant for duration of analog-digital conversion

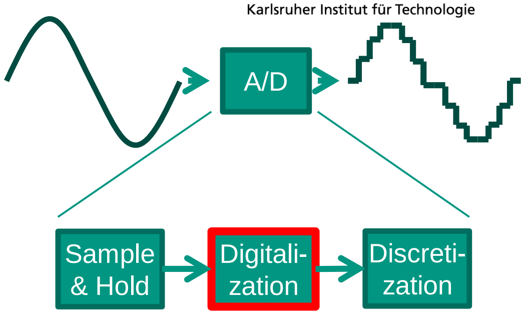




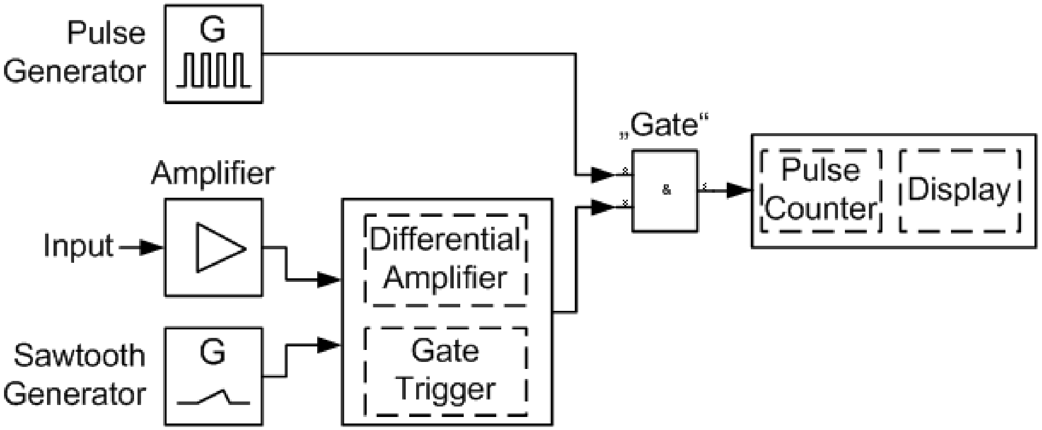
Impedance Converter

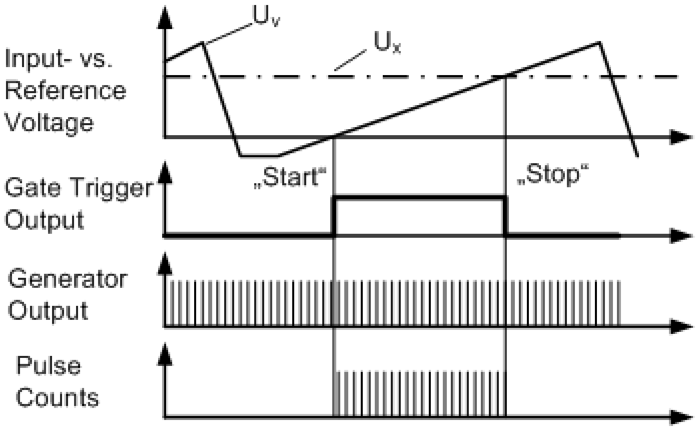
Voltage Follower

Digitalization



- **Single-Slope ADC (counter-ramp)**
  - Sawtooth Signal rises slowly from GND to U_e.
  - Counter is incremented until U_e is reached
  - Counter value is proportional to U_e
  - At the end of the sampling phase Counter value is stored into a register



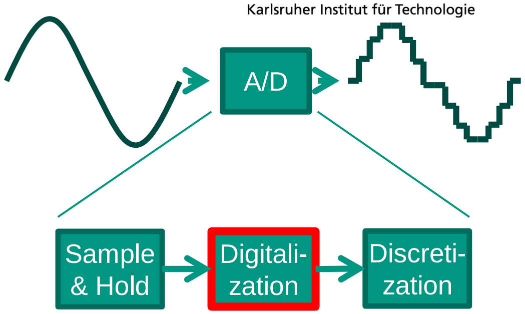


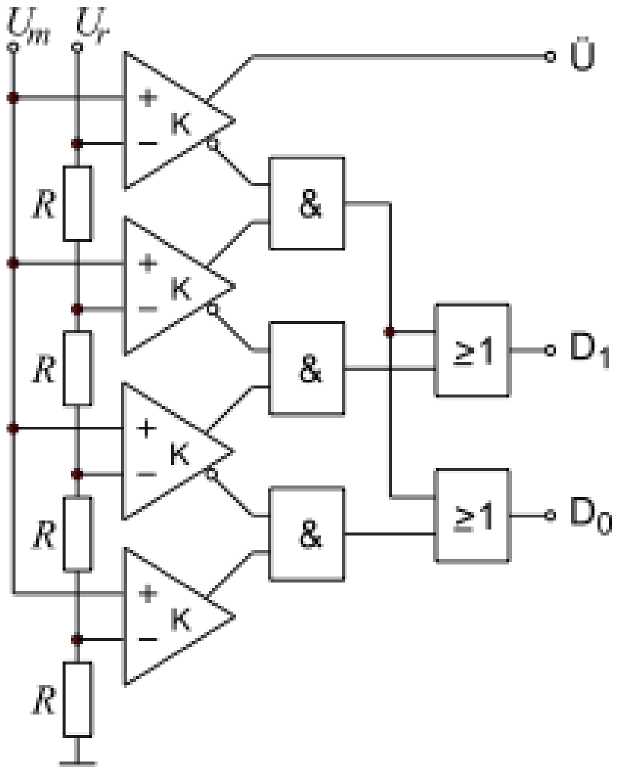
Source: http://www.alte-messtechnik.de/technik/ad-wandlung.php

→ More information see lecture “Elektronische Schaltungen”

Digitalization II

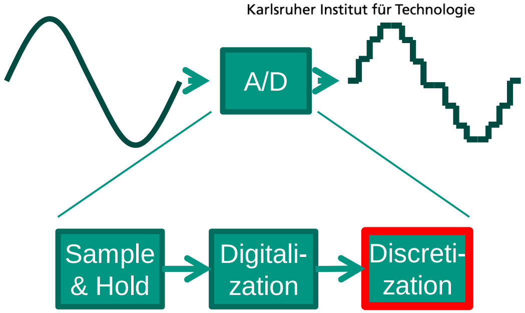
- Flash Converter
  - Direct conversion of analogue input signal into digital signal
  - Here: 2-bit full parallel Converter with binary encoding
  - Comparison of signal with reference voltage(s)
    - resistor cascade
    - Comparator "Cascade"
  - Digital Glue Logic for generation of binary value





Source: http://de.wikipedia.org

Discretization



In theory:

- After the first two steps, the signal can still be reconstructed perfectly
  - Signal value is sampled with unlimited precision

In reality:

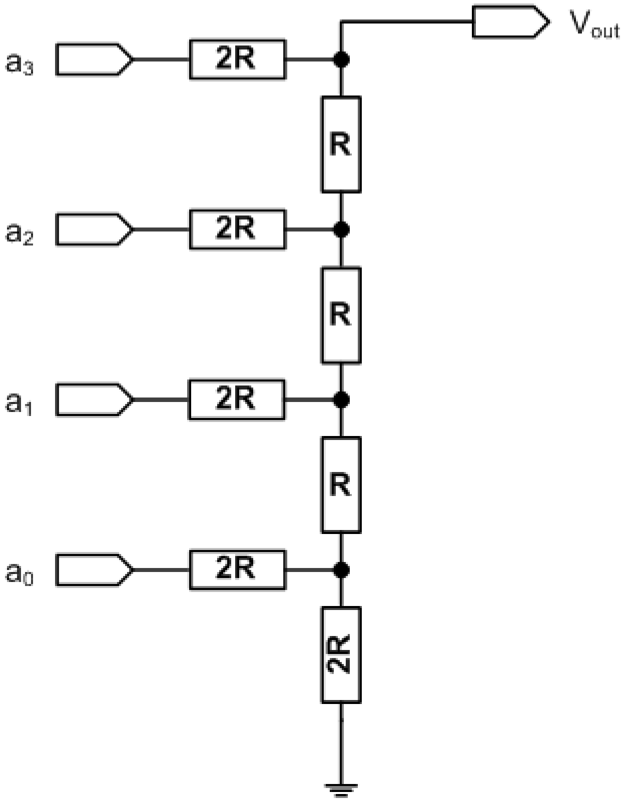
- Technical realization is restricted in the number of possible signal values
  - Quantization required to fit existing system
  - Only limited precision available
    - Cost
    - Amount of data
- Discretization step inherently included in AD Converters (see previous examples)

→ Signal can not be reconstructed perfectly because of errors introduced by quantification

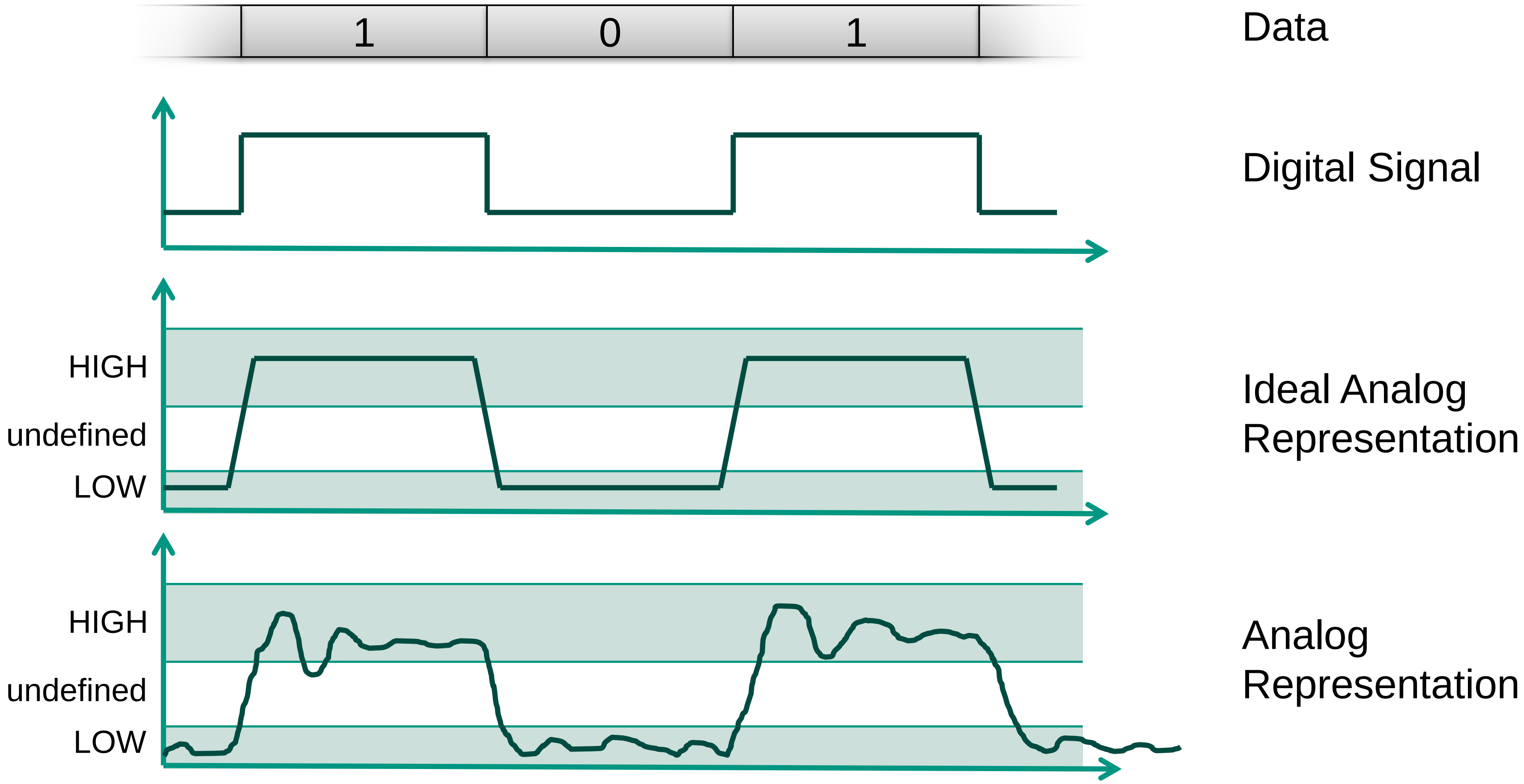
Digital Analog Conversion (DAC) – Example

- Digital Values are converted into analog signal
- Reconstruction Filters can be used to reconstruct original analog signal in some cases (reconstruction filters) using interpolation
  - Bandwidth has to be large enough
- Quantization Errors
  - Discretization step in ADC always introduces noise (relatively low)

R-2R Ladder

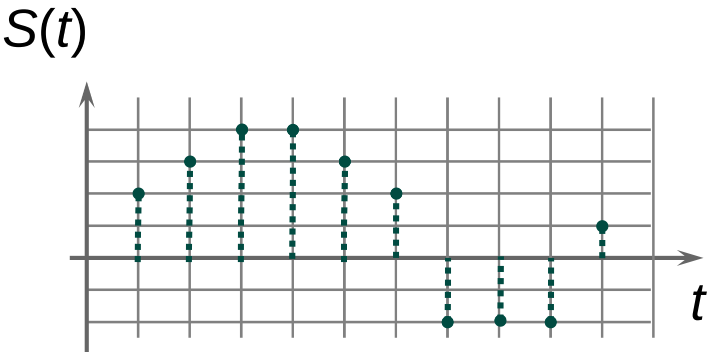


Representation of digital Signals and Sampling

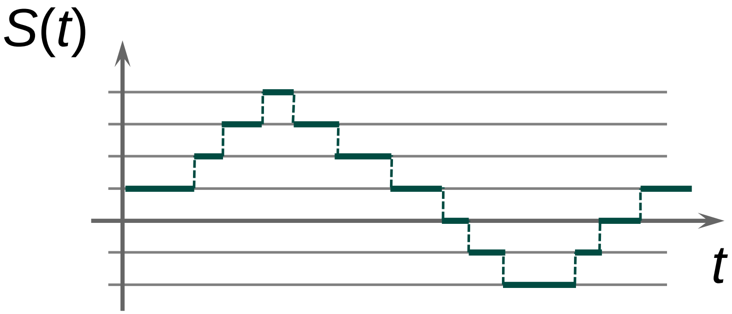


- Only interested in the value of a bit
- Decreases sampling frequency to one sample per bit
- In theory one sampling point per bit is enough

Motivation Oversampling



- Digital Signals are assumed to have discrete values only at discrete points in time
  - High/Low
  - -1 , 0 , +1



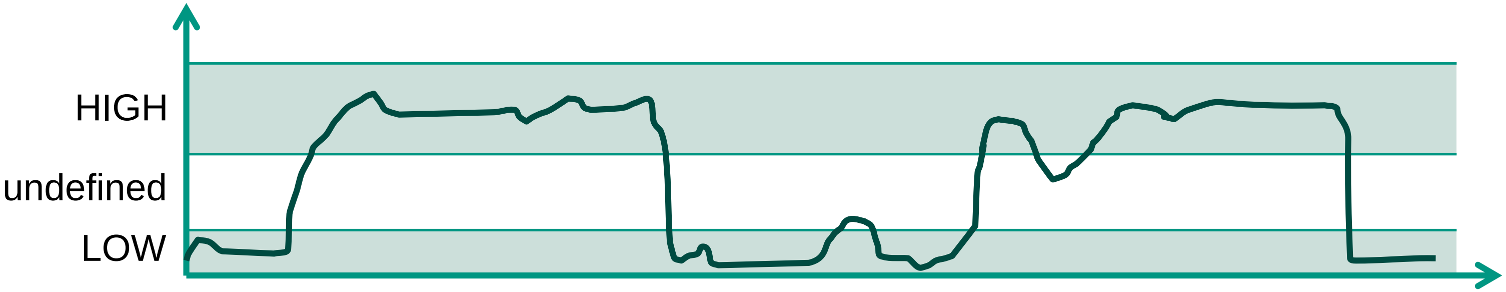
- Digital signals are often represented by analog signals
  - discrete values
  - time continuous

- Under ideal conditions, a single sample during a given interval is sufficient to reconstruct the original digital signal

Acceptance Bands for Digital Signals

- Absolute value of the analog representation is not of interest only the signal interval is of importance
- Distortion of digital signals is possible if the analog signal is distorted

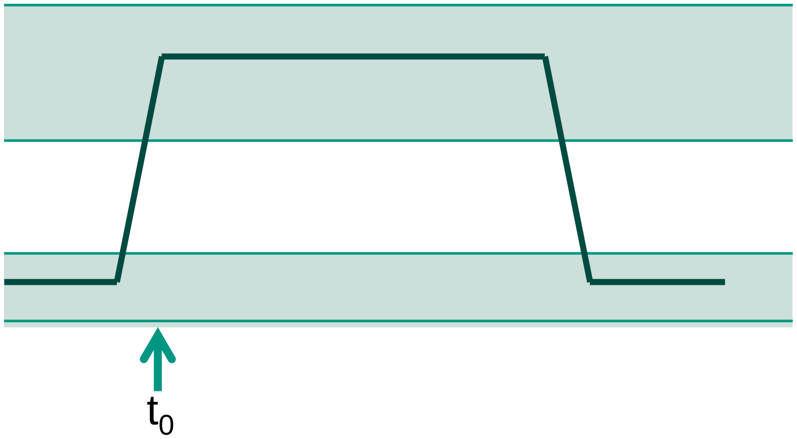
  → Definition of Acceptance Bands

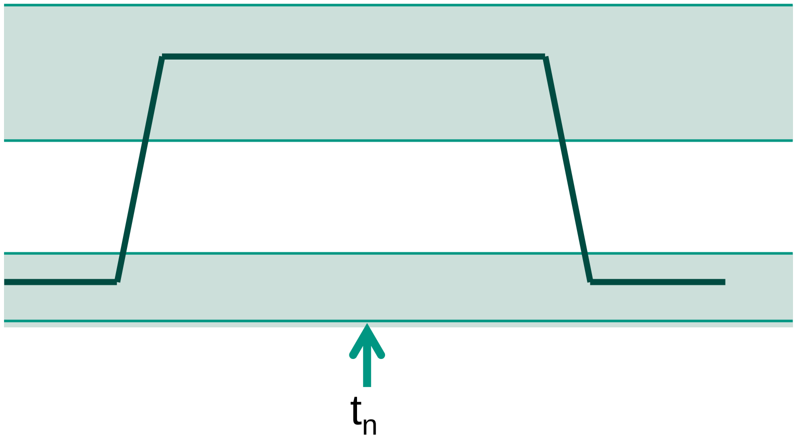


- Reconstruction of digital signal is possible as long as analog signal is within the acceptance bands
- If signal drops into undefined:
  - interpret a signal change only if and when a valid acceptance band is entered
  - Hysteresis (see Schmitt Trigger in the analogue domain)

Sample Points for digital Signals

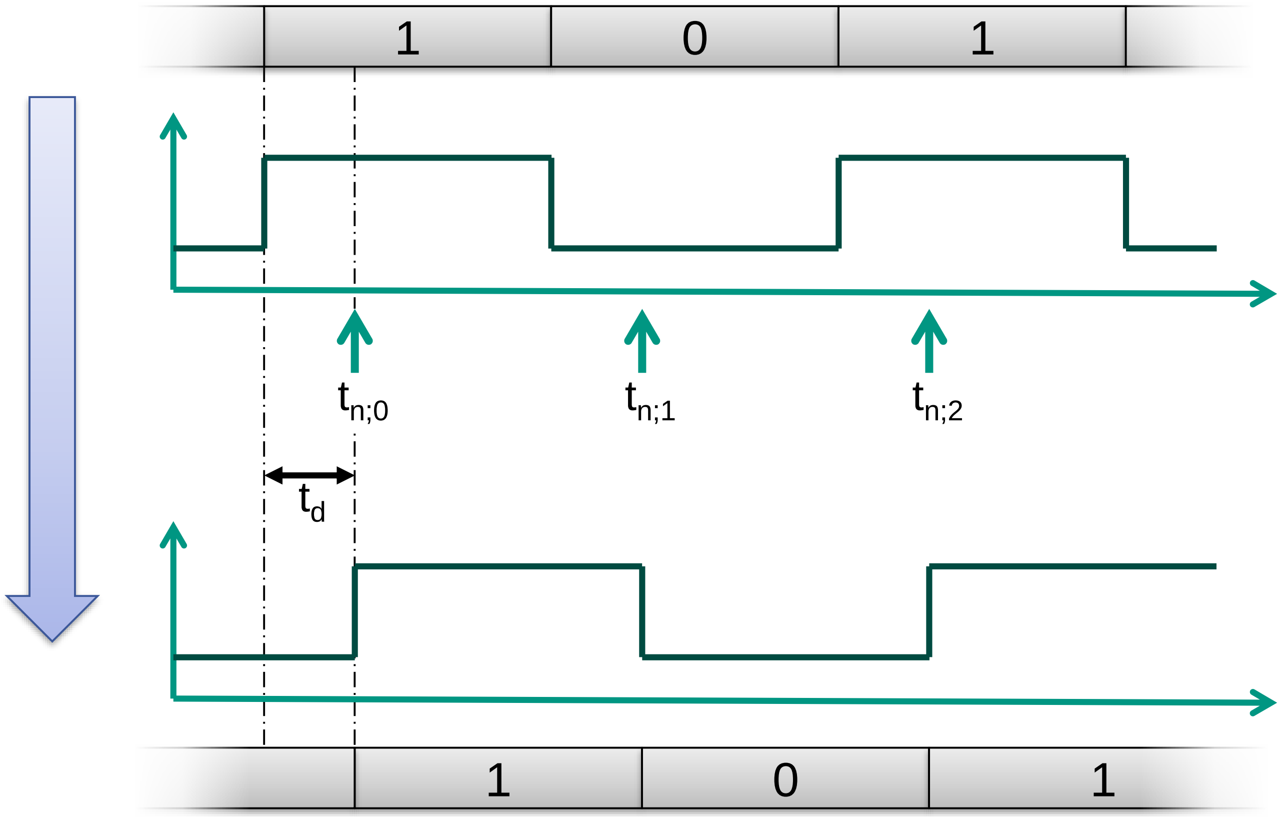
- Goal in a communication system: get valid data as early as possible
  → Sampling of signals as early as possible





- Sampling at t_0 (beginning of a bit)
  - + no latency
  - − possibly high error rate
    - over-/underswing
    - slopes
    - timing (detection of bit start)
- Sampling at t_n
  - + Signal is stable
  - − high latency with late t_n

Time Delay of Samples



Data (Sender)

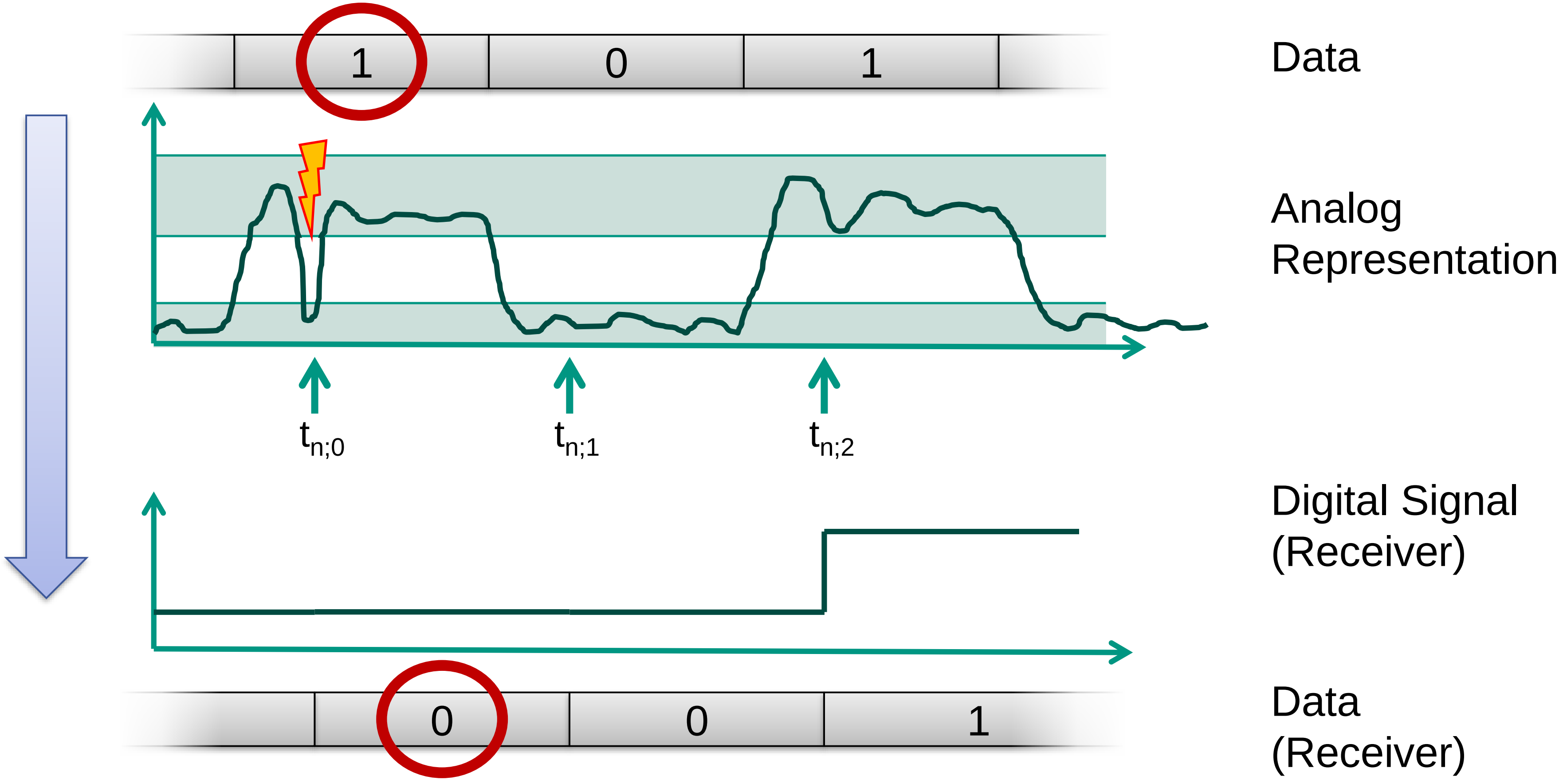
Digital Signal (Sender)

Digital Signal (Receiver)

Data (Receiver)

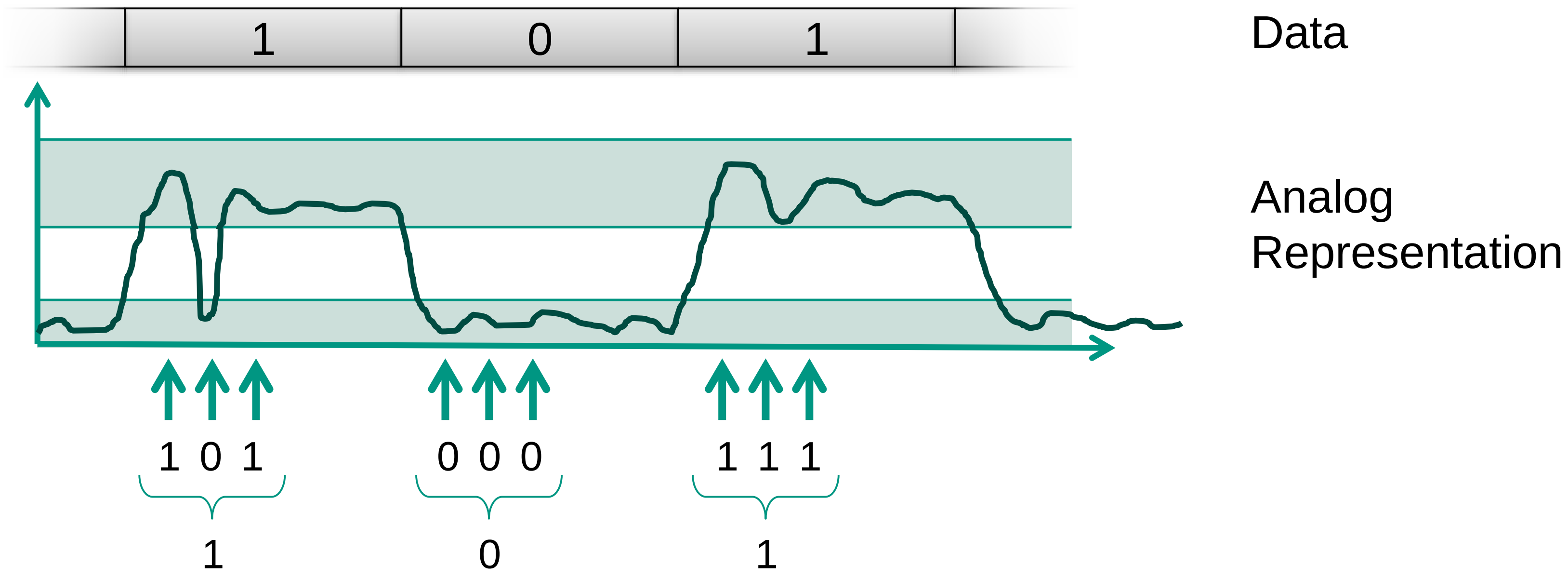
- Sampling at later time points increases latency (t_d) on receiver side

Error Case: Sampling of Digital Signals



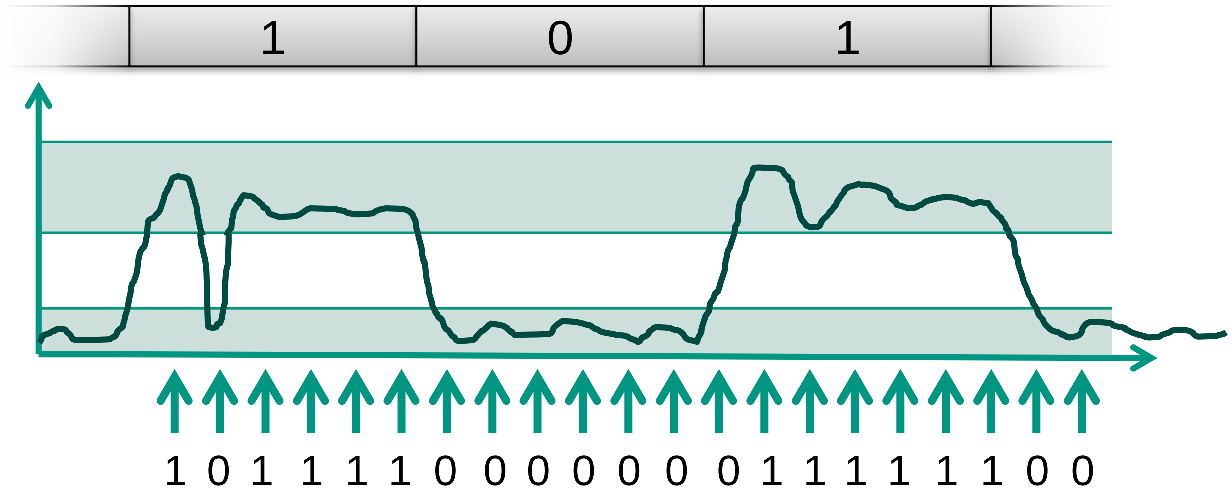
- Problem: Interferences (glitches) could change value of this single sample → wrong signal value will be assumed

Oversampling of Digital Signals



- Oversampling:
  - n sample points per bit (n-times oversampling)
  - Majority voting among the samples → value that appears most is taken
- Advantages:
  - Filtering of short distortions is possible
- Disadvantages:
  - Final signal is available only after the last sample has been taken

Oversampling of Digital Signals



1 0 1

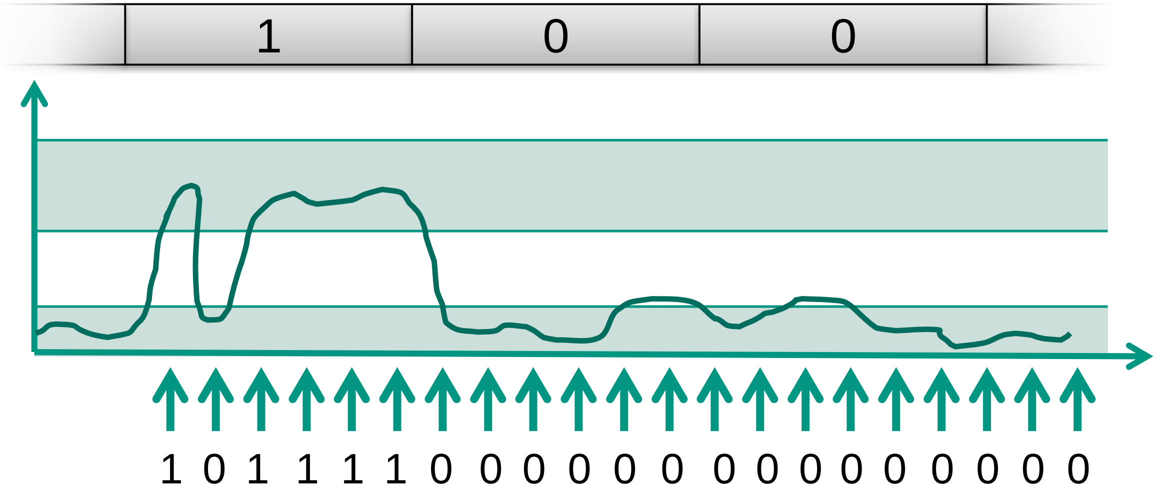
1 0 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0

Data

Analog Representation

- Oversampling can be done continuously
  - only first n sample points of a bit are used to read value
  - remaining samples are not needed
- If no information about the start of a bit is available:
  - sliding window
  - each sample value is used
  - allows for edge detection

Ambiguity of Digital Signals

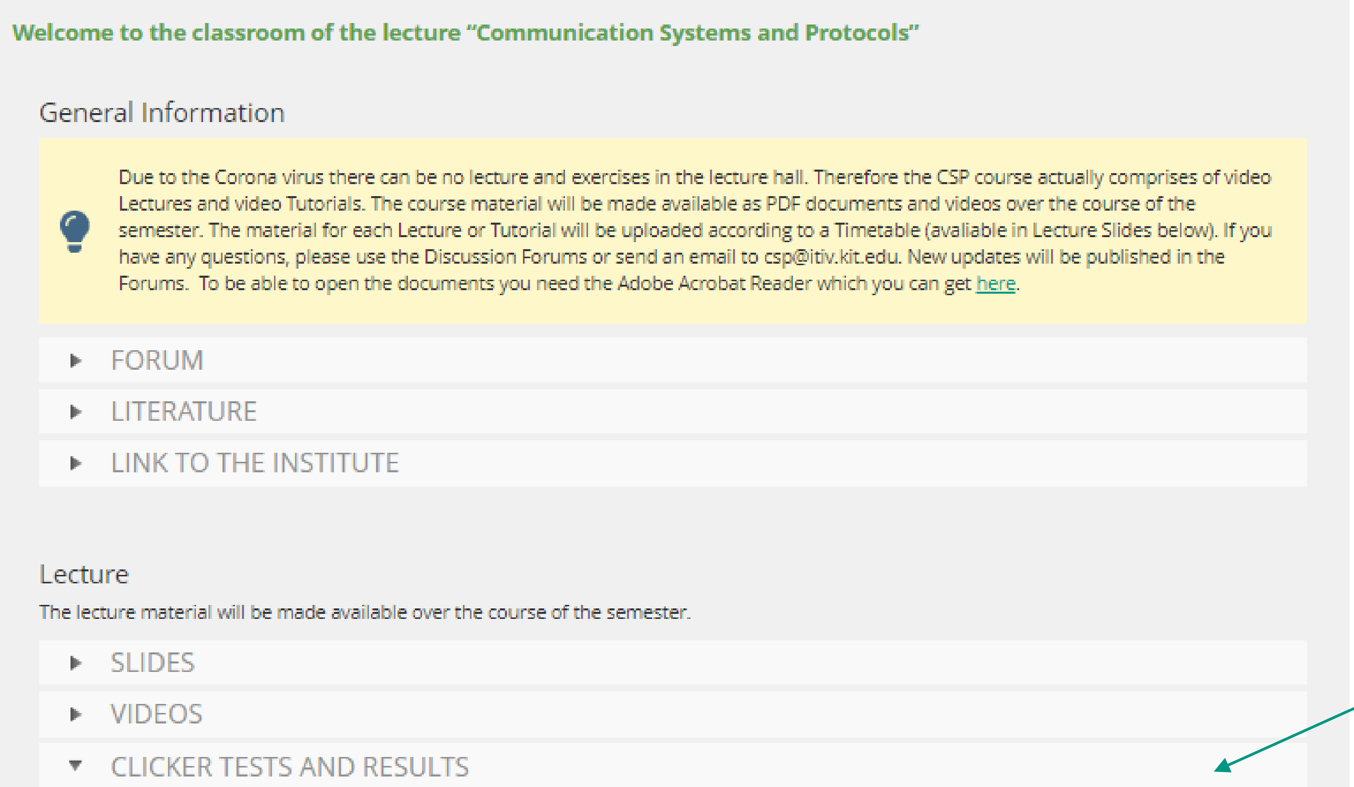


Data

Analog
Representation

- What happens with series of equal data?
  - Where does a bit begin and where does it end?
  - Additional information needed!

Participate in Clicker Test 3b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

▸ FORUM

▸ LITERATURE

▸ LINK TO THE INSTITUTE

Lecture

The lecture material will be made available over the course of the semester.

▸ SLIDES

▸ VIDEOS

▾ CLICKER TESTS AND RESULTS

Clicker Tests are added here

Thank you for your attention

Institut für Technik der
Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

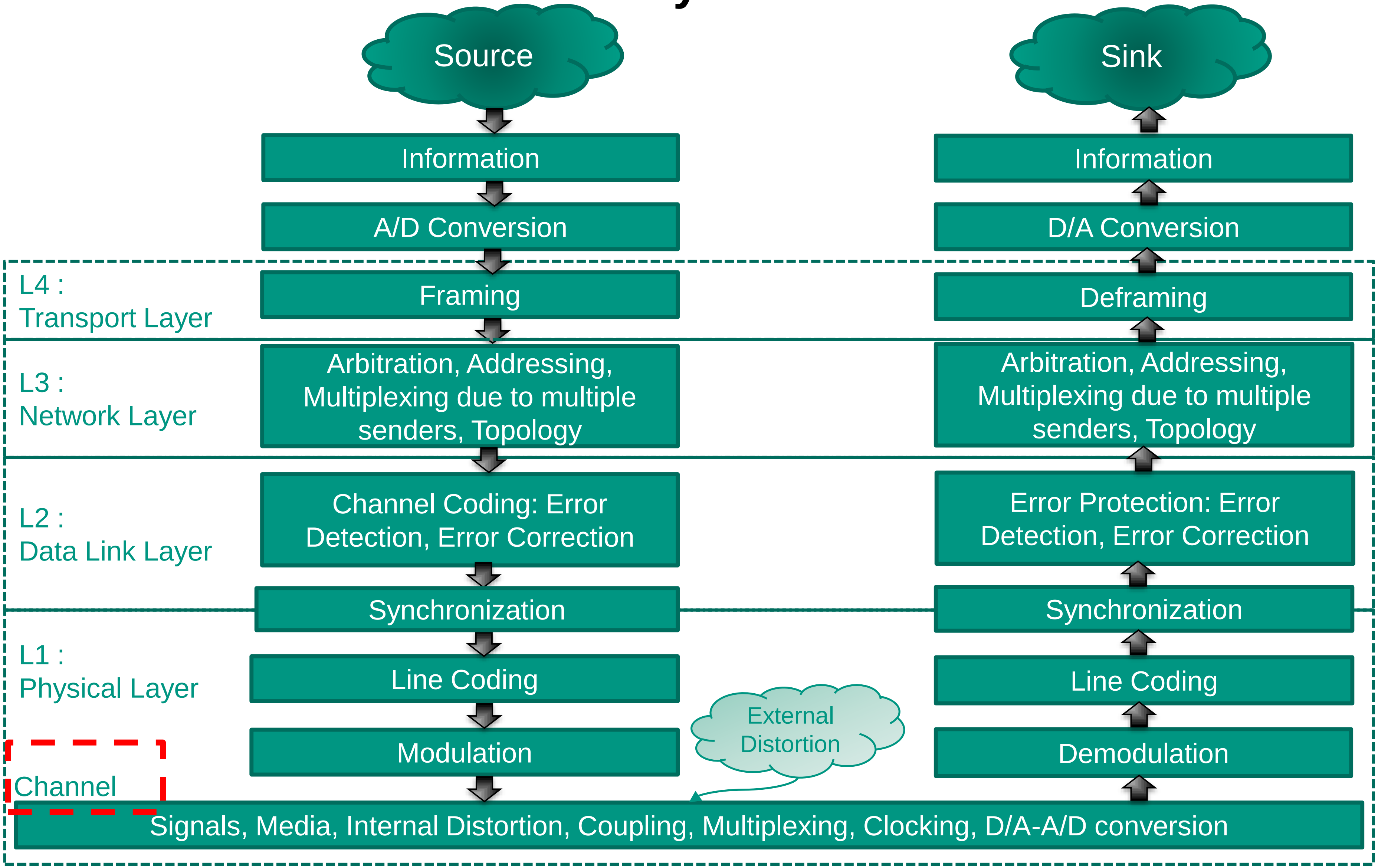
Channel Capacity

Contents

- Bit rate vs. Symbol rate

- Transmission capacity of a channel
  - Ideal channel
  - Real channel with noise

Technical Communication System


Source
Information
A/D Conversion
L4 :
Transport Layer
Framing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Synchronization
L1 :
Physical Layer
Line Coding
Modulation
Channel
External Distortion
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation

Definitions (recap)

- Bit
  - A **bit** (a contraction of **binary digit**) is the basic unit of information in computing and telecommunications. A bit can only have two values: either 1 or 0. A bit can be implemented in hardware by means of a two state device.
  - The symbol for bit, as unit of information, is either simply “bit” or lowercase “b”.

- Bit rate
  - In telecommunications and computing, **bit rate** is the number of bits that are conveyed or processed per unit of time.
  - The bit rate is quantified using the bits per second (bit/s or bps) unit.

Source: Wikipedia

Definitions II

- Symbol
  - A **symbol** can be described as either a pulse (in digital baseband transmission) or a “tone” (in passband transmission using modems) representing an integer number of bits. There may be a direct correspondence between a symbol and a small unit of data (for example, each symbol may encode one or several binary digits or 'bits') or the data may be represented by the transitions between symbols or even by a sequence of many symbols.

- Symbol rate
  - In digital communications, **symbol rate** (also known as baud or modulation rate) is the number of symbol changes (waveform changes or signaling events) made to the transmission medium per second using a digitally modulated signal or a line code.
  - The Symbol rate is measured in baud (Bd) or symbols/second.

Source: Wikipedia

Relation Between Bit Rate and Symbol Rate

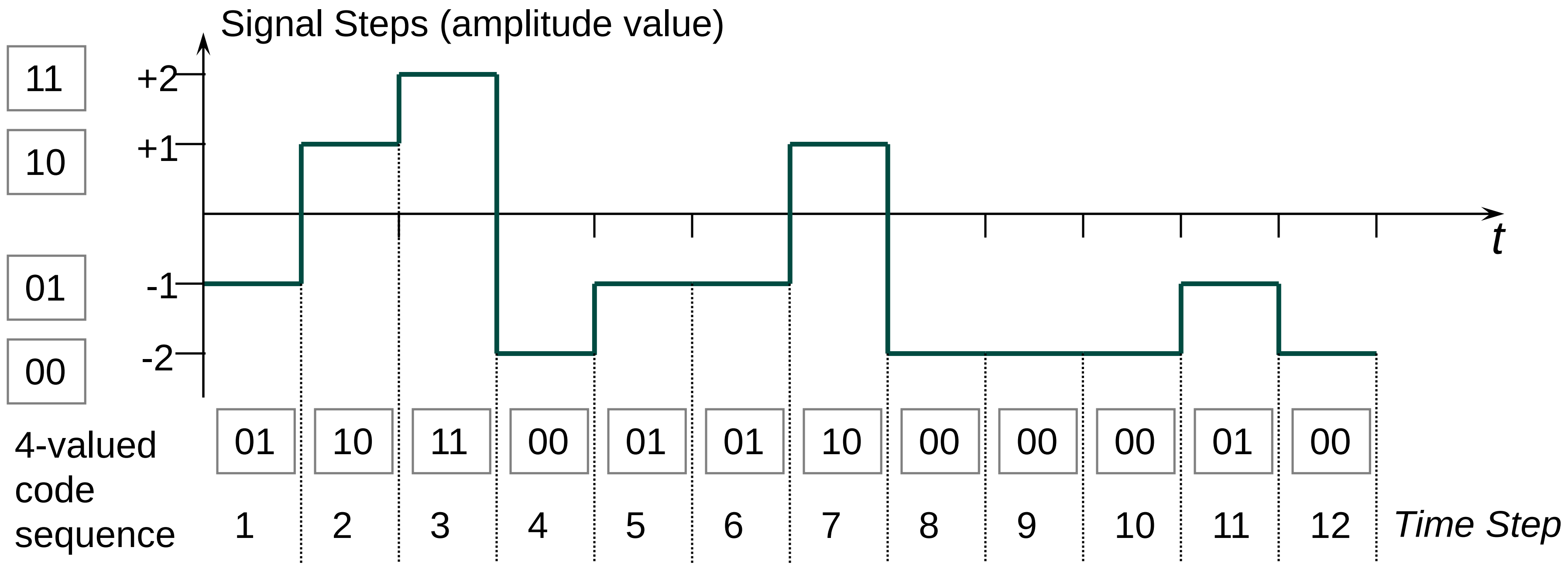
Bit rate is related to, but should not be confused with the symbol rate!

- Only equal when there are two levels per symbol, representing 1 and 0. → binary signals
- For most line codes and modulation methods:
    - Symbol rate <= Bit rate
    - A transmission scheme with 2^N signal levels can transfer N bit/symbol
        - Bit rate = symbol rate * N
- For some self-synchronizing line codes, where each bit is represented by two pulses
    - Bit rate = symbol rate / 2

Example: Multi-Value digital Signal

- The (discrete) coordinate of a signal can have more than two values
  - Example: DIBIT: two bit per coordinate value (4-valued Signal)

- One Symbol encodes two bits



Transmission Capacity of an ideal channel

- How many bits can be transmitted error free over a band limited channel?

- Following the sampling theorem:

  → a channel limited in frequency to f_{limit} can transmit at maximum $2 \cdot f_{limit}$ symbols per second.

- H. Nyquist, 1924:
  - Noise-free channel
  - V= amount of discrete signal steps

  → Maximum data rate: $2 \cdot f_{\mathrm{limit}} \cdot \log_2 V \; [bit/s]$

  - Data rate increases with the number of signal steps.

- Example:
  - F_{limit}= 3000 Hz, 8 signal steps
  - D_{limit}= 2 * 3000 * ld(8) = 18000 bit/s

Transmission Capacity of a real channel

- C. Shannon, 1948:
  - Channel with random noise
  - How many different signal steps can be distinguished correctly?

- Signal/Noise ratio S/N limits the number of bits per symbol

- Number of transmittable bits per second:
  - Shannon Limit = $W \cdot \log_2(1+S/N)\,[bit/s]$
    - S/N=0: no information (infinite amount of noise)
    - S/N=1: ~1 bit/s
    - S/N>1: multiple bits/s

- Example: analog telephone line
  - S_{norm}=-10dB, N_{norm}=-34dB, W=3000Hz

  → 24,000bit/s

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

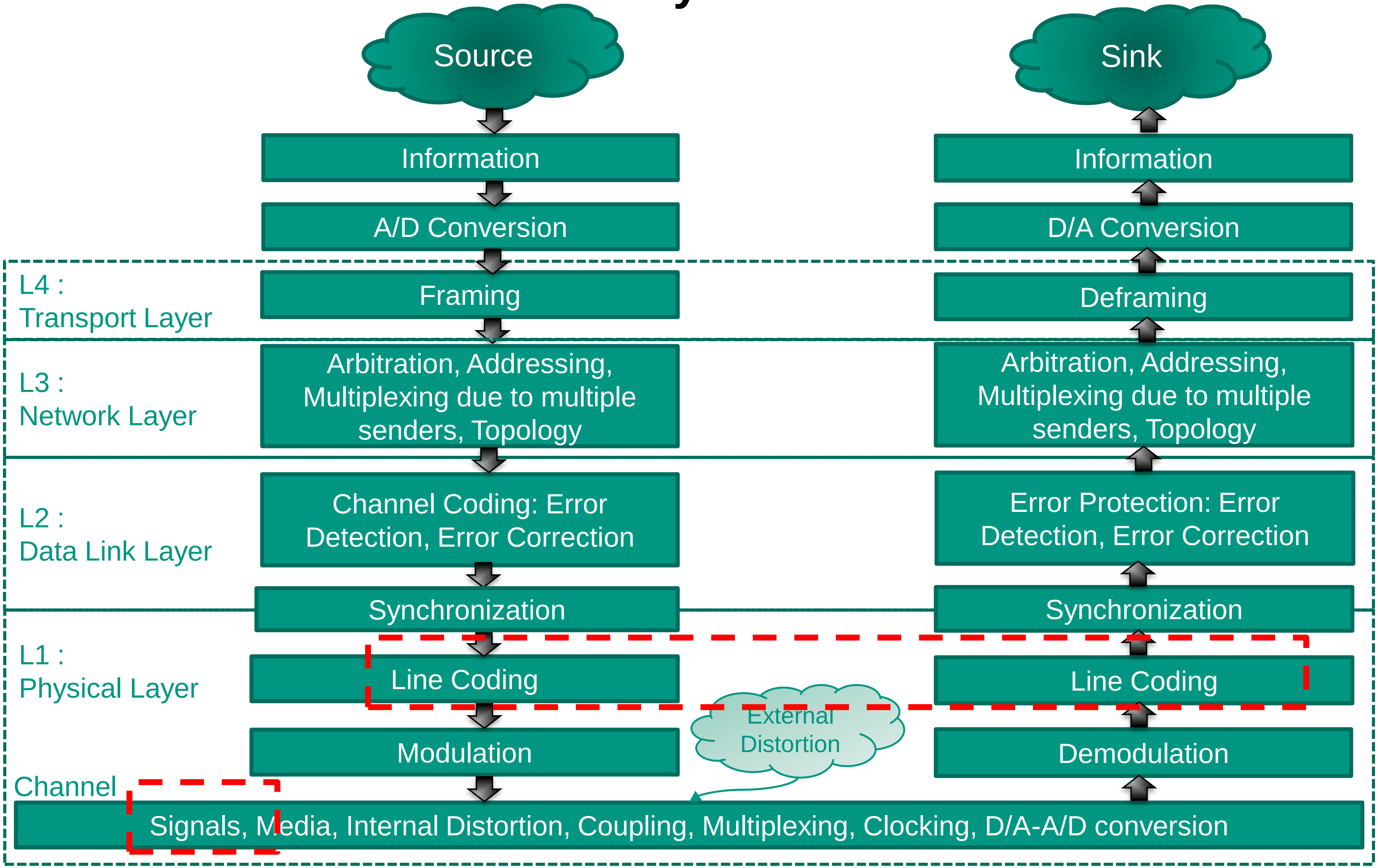
Communication Systems and Protocols

Line Codes

Contents

- Representation of Signals
- Line Codes
  - Non-Return-to-Zero (NRZ)
  - Return-to-Zero (RZ)
  - NRZI
  - Manchester
  - AMI
- Classification of Line Codes

Technical Communication System

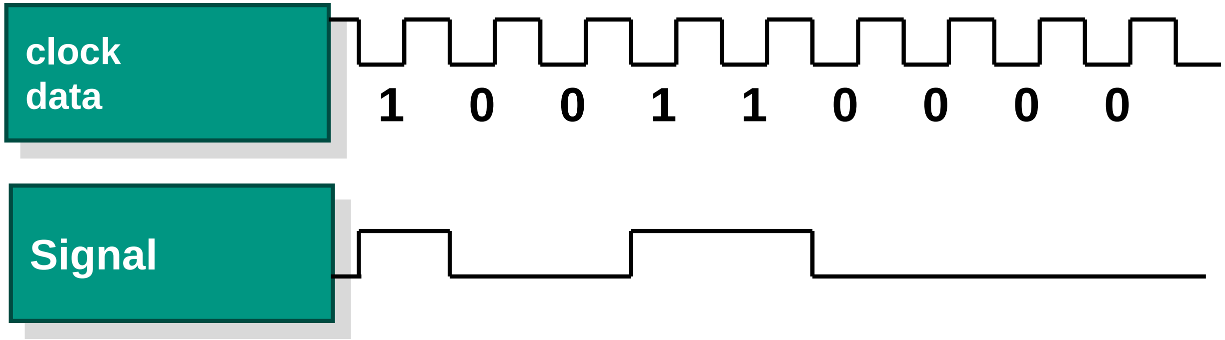

Source
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
External Distortion
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Representation of Signals in bus systems

- What is an electrical representation of binary signals?
  - High voltage, low voltage (e.g. amplitude)
  - High frequency, low frequency
  - current, no current
  - bright light, dim light
- Digital Baseband Representation
  - Signals contain no modulation, in general it contains f=0. Baseband signal = sequence of (square-)Impulses
- Modulation Representation
  - Signals are modulated onto a carrier frequency
  - it does not need to contain f=0

Baseband Representation

- Baseband Representation
  - Signals contain no modulation, in general it contains f=0.
    Baseband signal = sequence of (square-)pulses



- Intuitive representation of data
- Requires theoretically unlimited bandwidth (due to square-pulses)
- Capacitive or inductive coupling of nodes infeasible

Motivation: Line Codes

- Is pure baseband representation of bits sufficient?

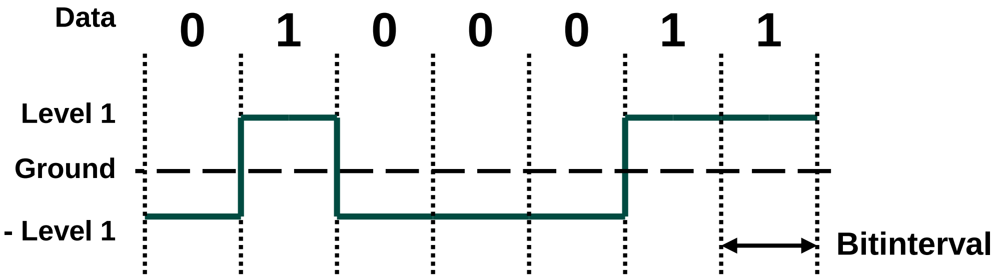
- Value of bit can be encoded, but how about other features?
  - Clock recovery
  - Multi-valued signals
  - DC freeness
  - Simple signal reconstruction
  - Error detection

- There should be other encoding schemes besides baseband representation → Line codes

Non-Return-to-Zero (NRZ)

- ‘1’ is represented by a HIGH-signal and ‘0’ as a LOW-Signal
- Invariant signal level during a bit interval
- Signal transitions occur at the interval boundary
- NRZ is the de-facto standard in digital systems

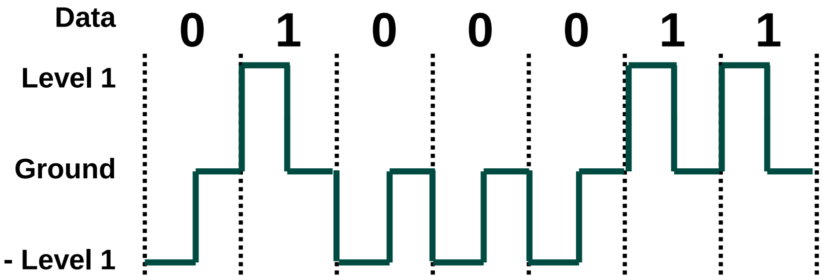
  - Easy to implement
  - DC Component can be substantial
  - Unsuitable for timing recovery



Return-to-Zero (RZ)

- Logic ‘1’ is represented by a HIGH-signal in the first half of the bit interval
- Logic ‘0’ is represented by a LOW-signal in the first half of the bit interval
- In the second half of the bitinterval code returns to the basic state (Zero)

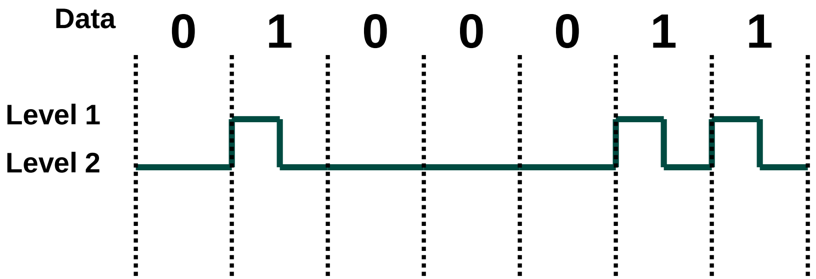
  - Easy Clock Recovery possible
  - The baud rate is double the bit rate → doubled bandwidth is necessary
  - DC Component can be substantial



Return-to-Zero (RZ) Unipolar

- ‘1’ is represented by a HIGH-signal and ‘0’ as a LOW-Signal
- Square pulse within the first half of a bit interval stands for a ‘1’
- Afterwards code falls back to the basic state (zero)

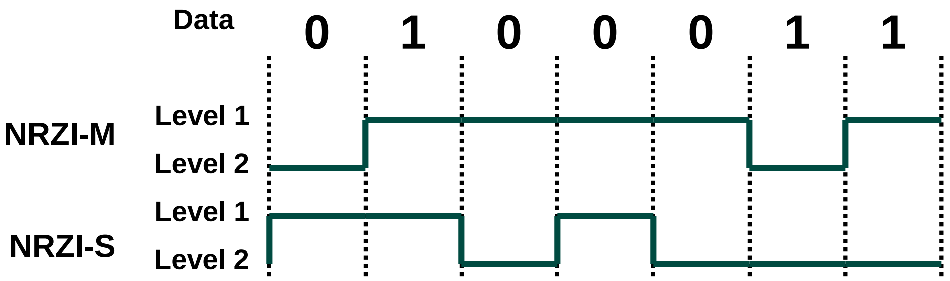
  - Simple implementation in digital hardware
  - In the extreme case (long sequences of ‘1’s) the baud rate is double the bit rate → doubled bandwidth is necessary
  - No timing recovery possible for long sequences of ‘0’s
  - DC Component can be substantial



NRZI Coding (Differential coding)

- Not the absolute signal value is used for coding, but instead a signal value depending on the polarity of the previous signal element.
- Two variants of NRZI codes
  - NRZI-M (Mark)
    - Signal transition to opposite signal value when transmitting ‘1’
    - No signal transition when transmitting ‘0’
  - NRZI-S (Space)
    - Identical to NRZI-M, but signal transitions when sending a ‘0’

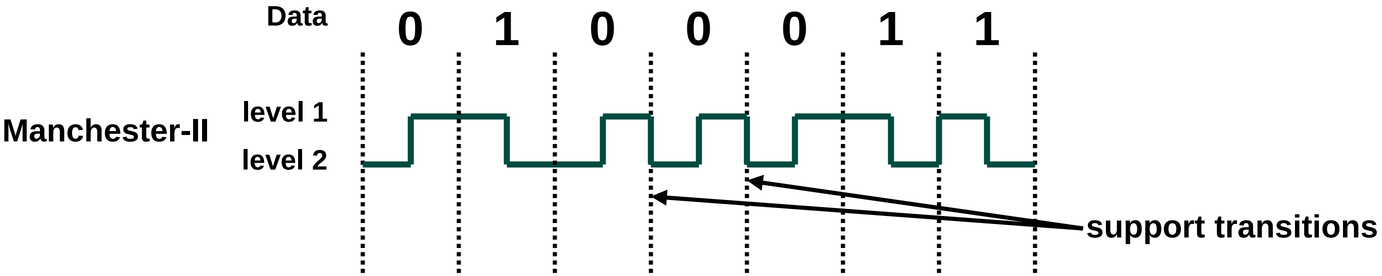
  - For both variants:
    - Edges more easily detectable than absolute levels
    - DC Component can be substantial
    - Timing recovery not always possible



Manchester Code

- At least one signal transition per bit interval, maximum two transitions per bit
- One possible definition (Manchester-II):
  - ‘1’= Signal transition from HIGH level to LOW level in the middle of the bit interval
  - ‘0’= Signal transition LOW level to HIGH level in the middle of the bit interval
  - Can be created via XORing of NRZ data and clock signal

    - Easy timing recovery, because at least one signal transition per bit interval exists
    - no DC component exists **for bipolar encoding**
    - Error detection on signal level: Missing an expected signal transition
    - Support transitions increase baud rate → higher band width necessary

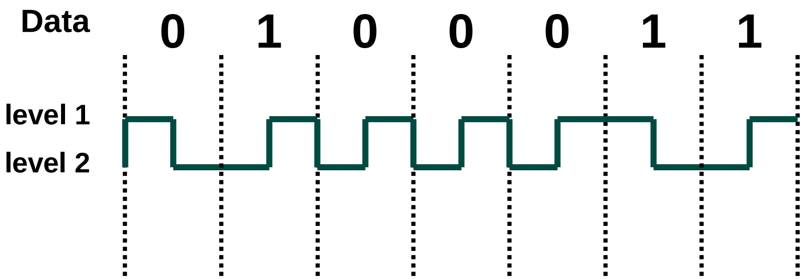


Differential Manchester Code

- Signal transitions in the middle of each bit interval
- Signal transition at the start of a bit interval only when encoding ‘0’

- This can be seen as a phase change when encoding ‘1’ and no change in phase when encoding a ‘0’

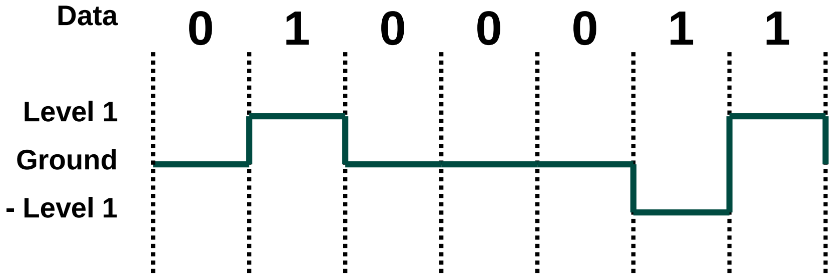
- Output signal dependent on initial signal level at t_0
- Independent of polarity



AMI-Code (Alternate Mark Inversion)

- Encoding with more than two signal values
- ‘1’ is represented by alternating positive or negative pulses

  - No DC component
  - Long ‘0’-sequences impede timing recovery



Classification of Line Codes

- Level orientated Line Codes
  - Symbols are represented by signal value
- Phase orientated Line Codes
  - Symbols are coded through phase jumps
- Unipolar Line Codes
  - only one active Signal level is used in addition to GND level
- Bipolar Line Codes
  - two different active Signal levels are use, GND is not used for coding
- Ternary Line Codes
  - Both symbol values 0 und 1 are encoded using three signal values (level 1, ground, level 2)
- Multi-valued Line Codes
  - Multiple bits are assembled within a single coding symbol

Important properties of Line Codes I

- Number of collectively coded bits
  - In a single value more than one bit can be encoded

- Direct Current Component
  - Sum V_{DC} of voltages on a wire over a longer period of time

    $V_{DC} = 0$ → DC balanced (or: zero-DC, zero-bias, DC equalized)
    $V_{DC} \neq 0$ → a DC current is transported over the line

  - Problem, if not DC balanced:
    - When transmitting data and power supply voltage over the same line, the supply voltage is affected
    - When using transducers as couplers, varying DC affects data interpretation on receiver's end

  - In practice, DC balancing is only achievable in the statistic average of long transmission sequences.

Important properties of Line Codes II

- Error detection
  - Signal errors shall be detectable on the signal level itself.
- Required Bandwidth
  - How many signal changes are required to encode one bit?
  - Is correlated to attenuation
  - Higher frequencies are attenuated more than lower frequencies
- Clock Recovery
  - From received signals the value and clock of the send data can be recovered.
  - Clock recovery is needed if no separate clock line is available.
  - The clock portion shall not be correlated to the data content of the transmission.

Homework

- Classify the Line Codes according to the properties given in the last two slides (+, ++ , - , -- , O)

Line Code	DC Current	Required Bandwidth	Clock Recovery
NRZ			
RZ			
NRZI			
AMI			
Manchester			

Thank you for your attention

Homework

- Classify the Line Codes according to the properties given in the last two slides (+, ++ , - , -- , O)

Line Code	DC Current	Required Bandwidth	Clock Recovery
NRZ	- -	+	-
RZ	- -	O / -	+ / -
NRZI	- -	+	-
AMI	+ +	+	O
Manchester	+ +	O / -	+ +

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

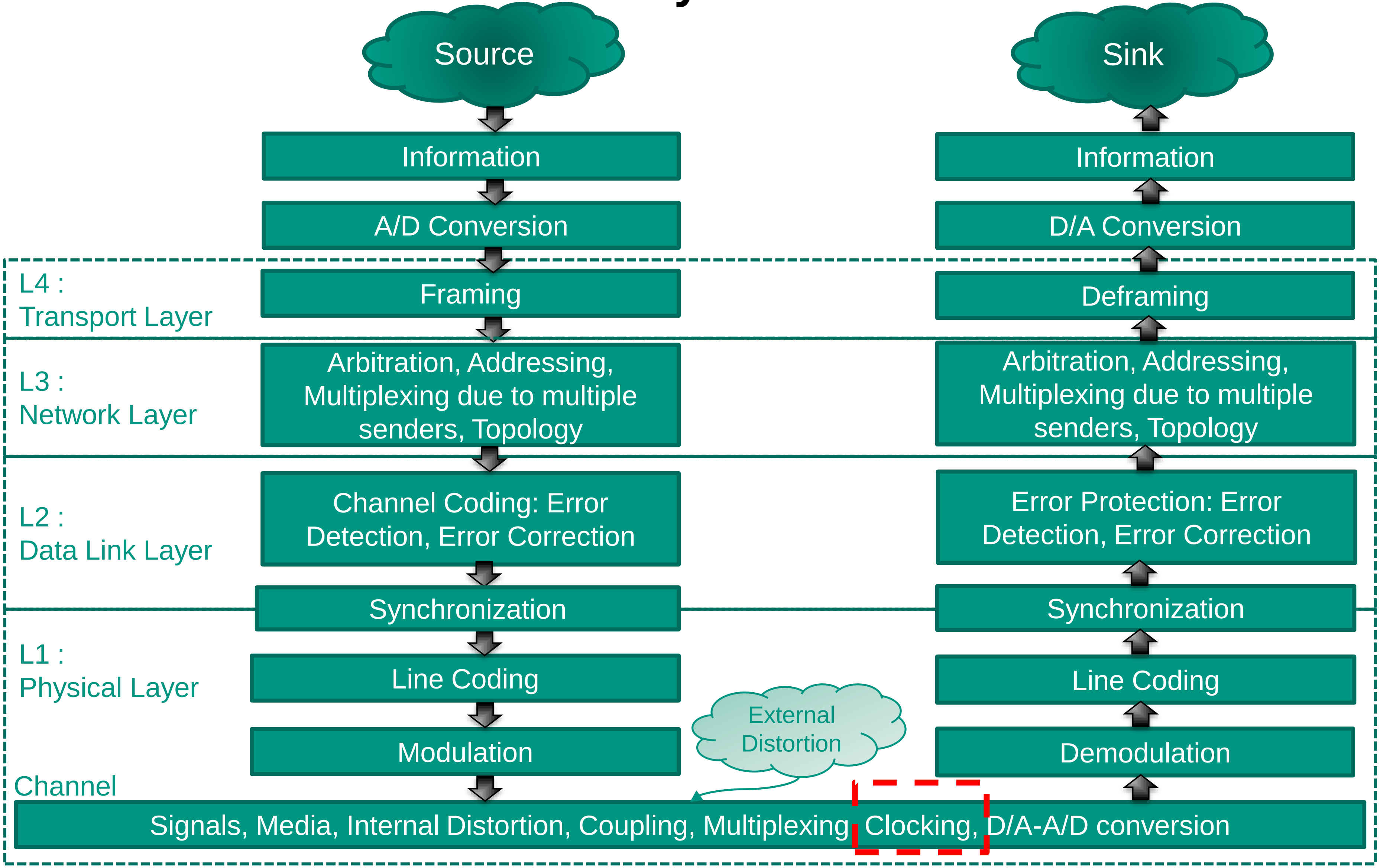
Clocking

Contents

- What is Clocking?

- Methods for clocking
  - Clock Signal
  - Scrambler

Technical Communication System



Source

Information

A/D Conversion

L4 :
Transport Layer

Framing

L3 :
Network Layer

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

L2 :
Data Link Layer

Channel Coding: Error Detection, Error Correction

Synchronization

L1 :
Physical Layer

Line Coding

Modulation

External Distortion

Channel

Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Sink

Information

D/A Conversion

Deframing

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

Error Protection: Error Detection, Error Correction

Synchronization

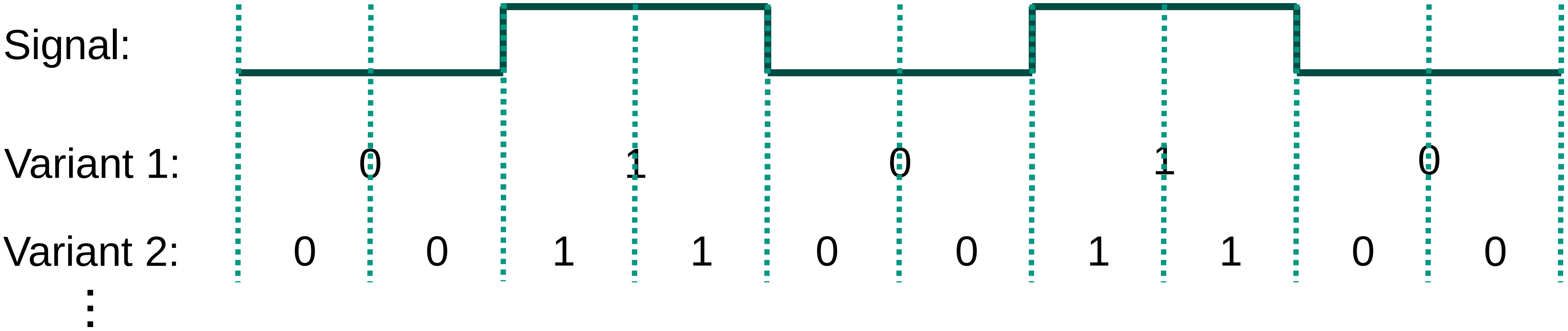
Line Coding

Demodulation

Clocking

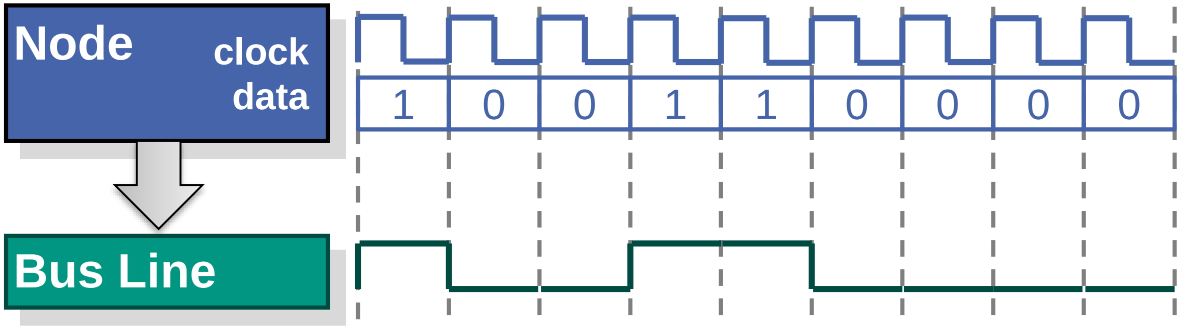
- For data transmission it is important to separate individual bits
  - Where is the end of a bit, when does a new one start?
  - Possible ambiguity

→ Clocking is used to determine to borders of a bit

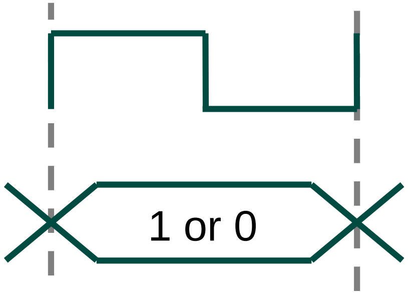


Using a Clock Signal

- Transmission of information is based on time slots (clock cycles)

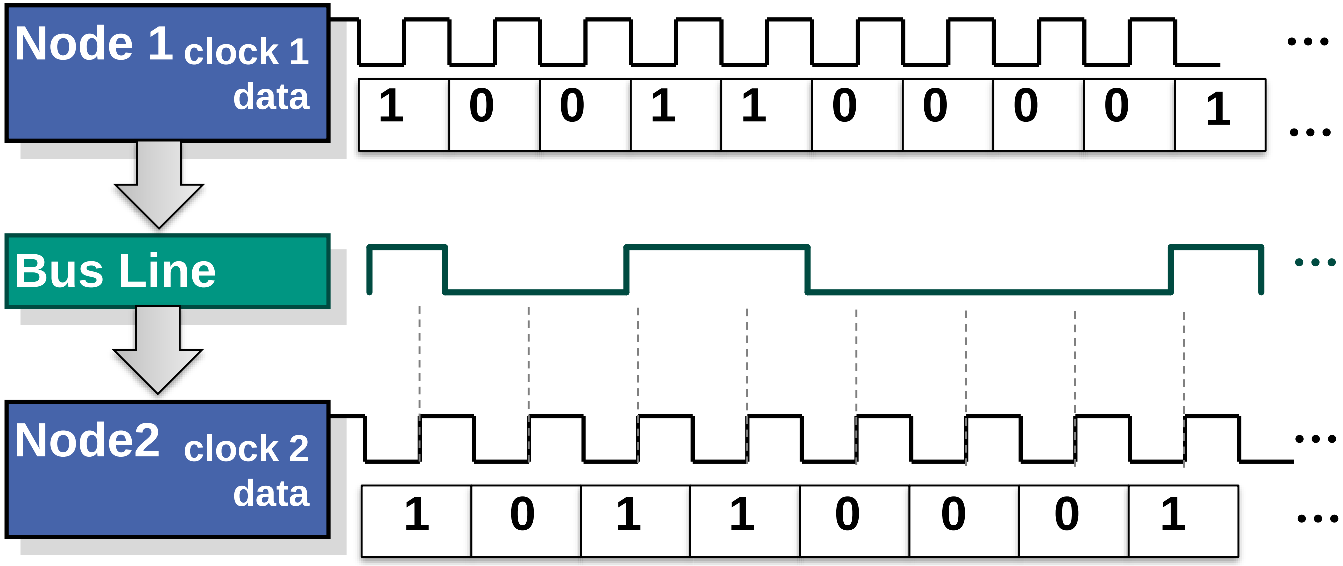


- Clock signal acts as time measure



Problems for transmission without clocksignal

- For correct transmission of data sender and receiver have to have a common understanding of the current time (clock)



- Clock Recovery from Data allows
  - Detection of correct bit intervals
  - Synchronization

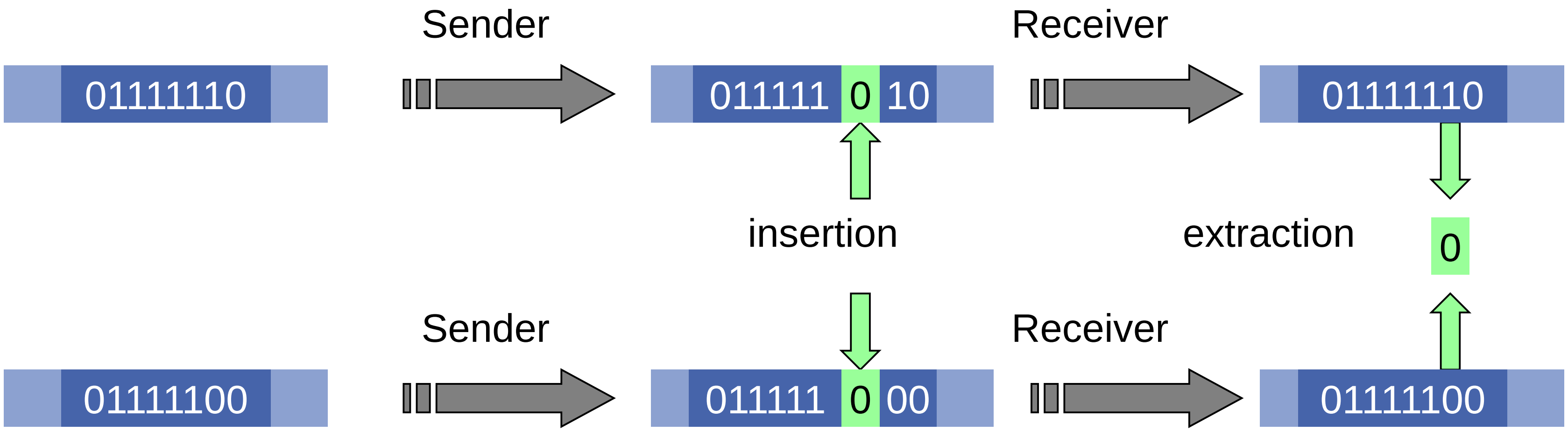
Block Codes

- 4B/5B Code
  - Avoiding the inefficiency of Manchester encoding
  - No long sequences of ‚0‘s or ‚1‘s
  - Addition of an extra bit to avoid such sequences
  - 4-Bit data is encoded in 5-Bit blocks
    - Not more than one leading ‚0‘s
    - Not more than two trailing ‚0‘s
  - Transmission is done using NRZ-I-Code
  - 80% efficiency

4-Bit Data	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

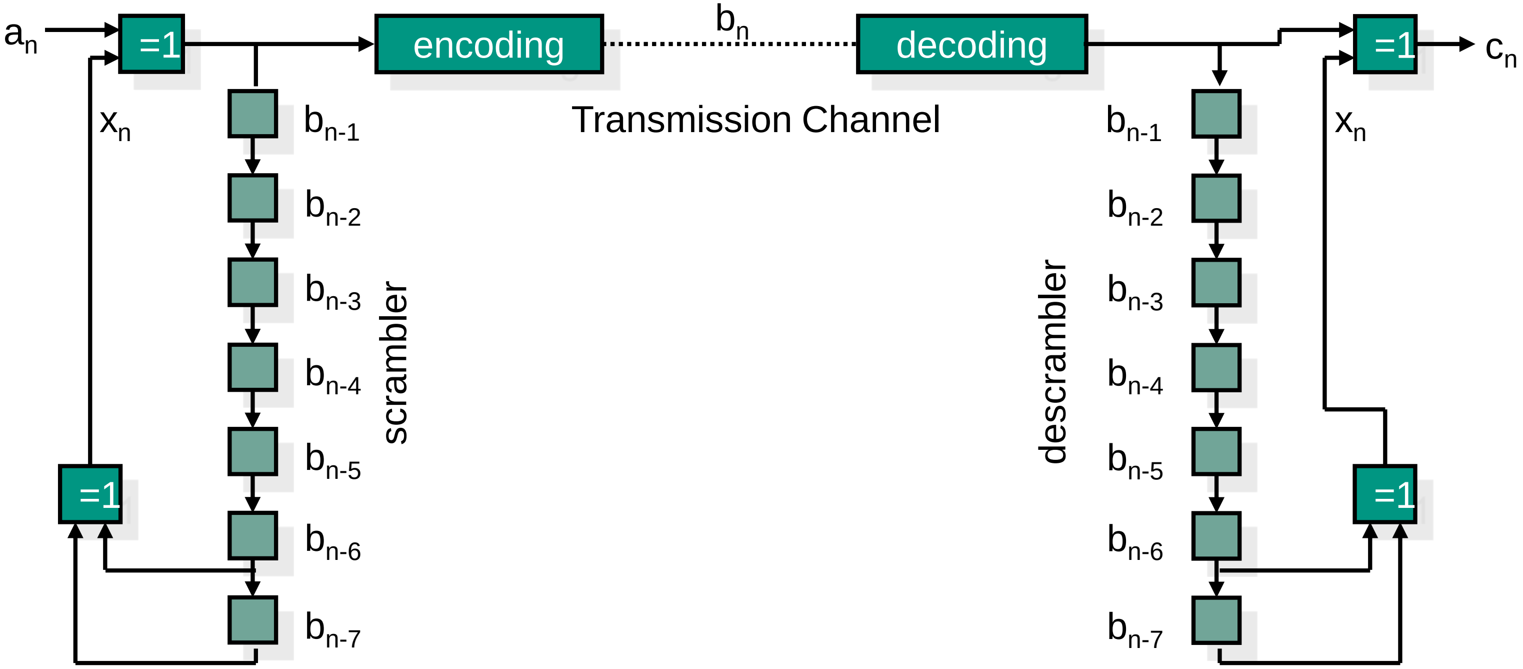
Scrambler: Bitstuffing

- Goal: Ensure timing recovery using frequent signal transitions, if no proper line encoding is being used.
- Bit Stuffing
  - Sender always inserts a ‚0' after *n* ‚1's (or vice versa)
  - Receiver interprets every sequence of 111...110 as 111...11 (or vice versa)
  - Example: *n=5*



Scrambler: Use of Linear Feedback Shift Register

- Goal: Creating a seemingly chaotic sequence of bit values from any given data stream (including long sequences of ‘1’s or ‘0’s)
- Data stream is linked with sequence of a linear feedback shift register (LFSR)
- Resulting bit is send to receiver and also fed back into LFSR



- Descrambler build symmetrically to scrambler
- It holds: $c_n = b_n \otimes x_n = (a_n \otimes x_n) \otimes x_n = a_n$ with $x_n = b_{n-6} \otimes b_{n-7}$ (on both sides)

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

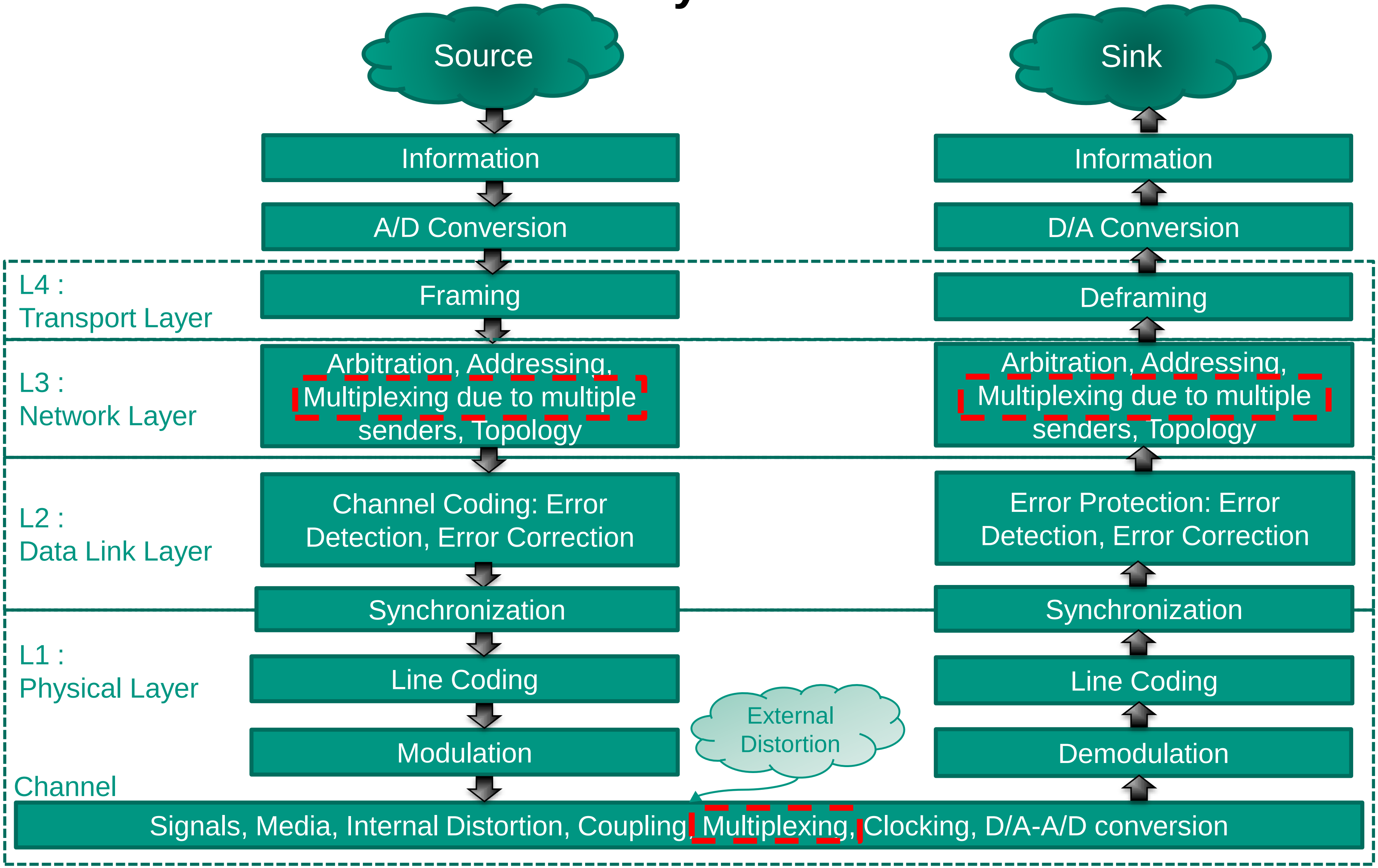
Multiplexing

Contents

- Types of Transmission

- Multplexing
  - Space division
  - Time divison
  - Frequency division
  - Code division

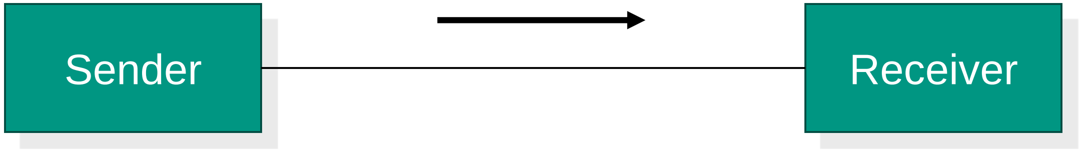
Technical Communication System


Source
Sink
Information
Information
A/D Conversion
D/A Conversion
L4 :
Transport Layer
Framing
Deframing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Error Protection: Error Detection, Error Correction
Synchronization
Synchronization
L1 :
Physical Layer
Line Coding
Line Coding
External Distortion
Modulation
Demodulation
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Types of Transmissions

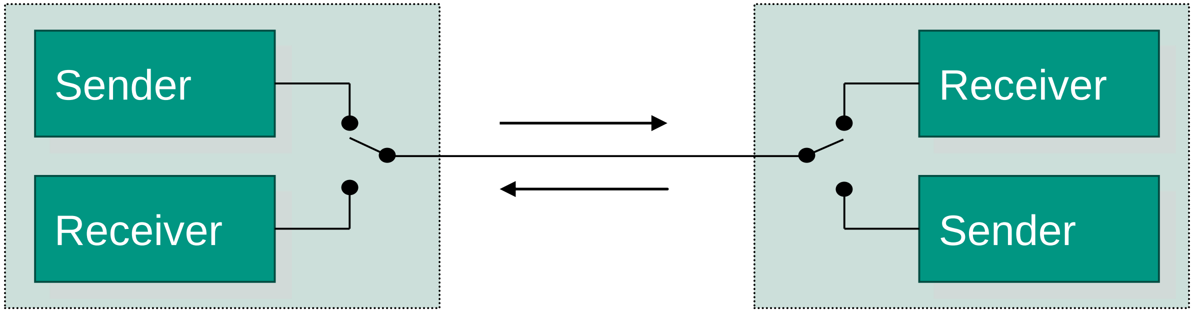
A transmission on a channel can be classified according to the direction of the data transmitted, as well as according to the degree of parallelism.

- Simplex transmission (respectively directional operation)
  - Only a single direction
  - Application: television, terrestrial radio, process data acquisition

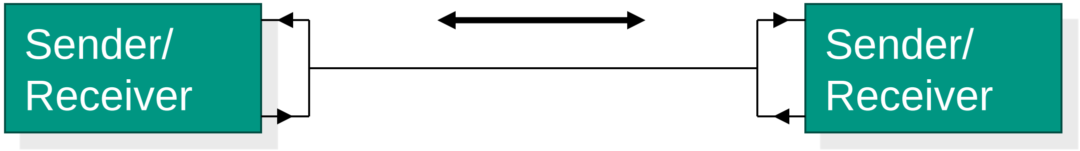


Types of Transmissions

- Half duplex transmission (also: alternating operation)
  - In both directions, but not simultaneously
  - Switching in terminal equipment
  - Application: radio communication, 10base-T Ethernet , ...

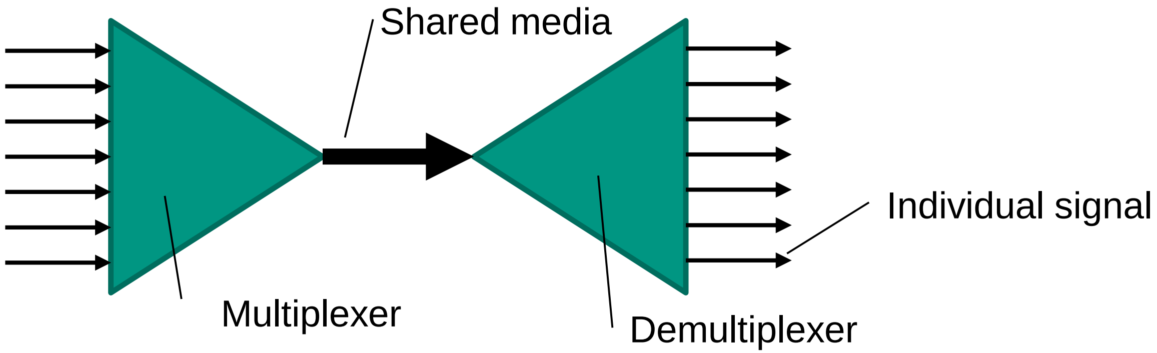


- Full Duplex transmission
  - In both directions simultaneously
  - Application: Telephone, RS 232, 100Mbit Ethernet,...



Multiplexing

- Multiplexing is a way to combine multiple signals into one shared media

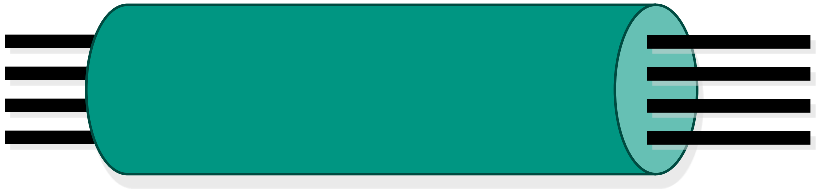


- Subdivision of a physical channel into logical channels
  - The individual signals are assigned to the logical channels

- Different solutions for multiplexing possible

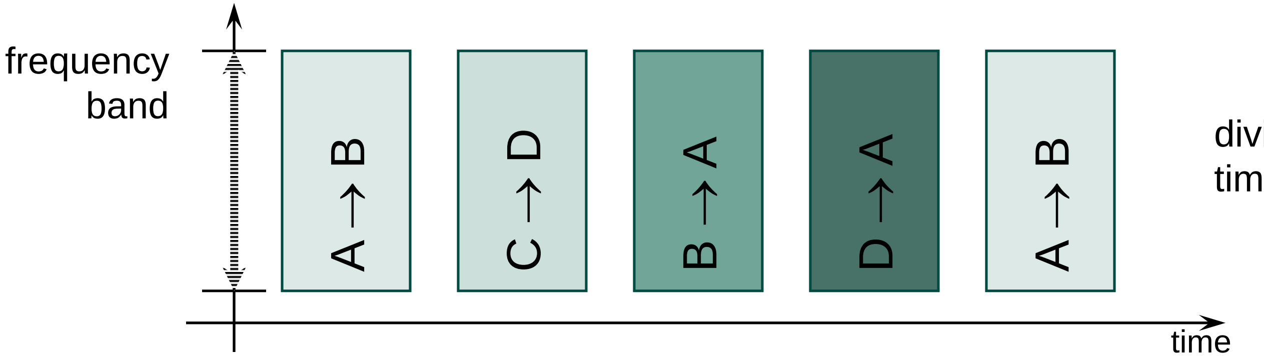
Examples of multiplexing

- **Space division („copper multiplex“)**
  - Representation of signals on multiple lines, baseband



bundle of
wires (pairs)

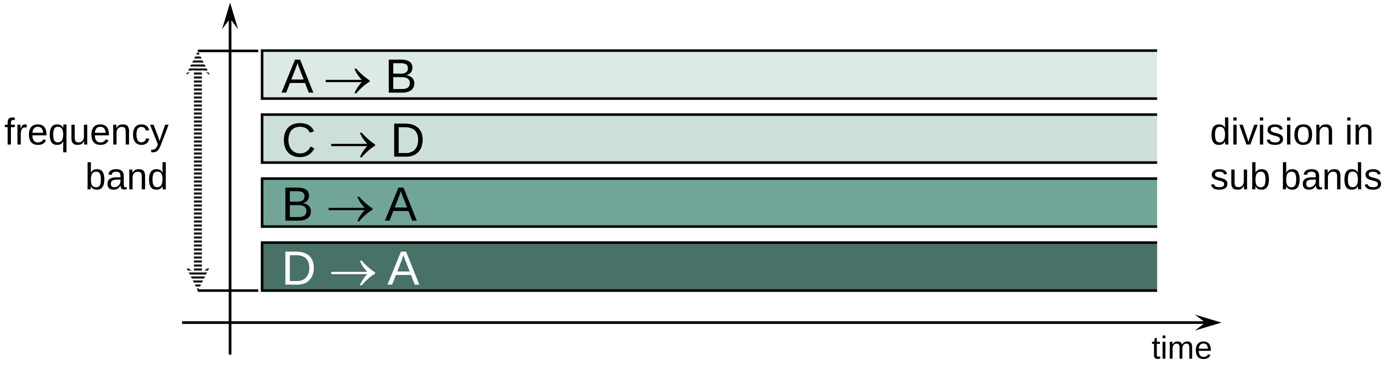
- **Time division multiple access (TDMA)**
  - Representation using temporal shifts (slots), base band



division into
time slots

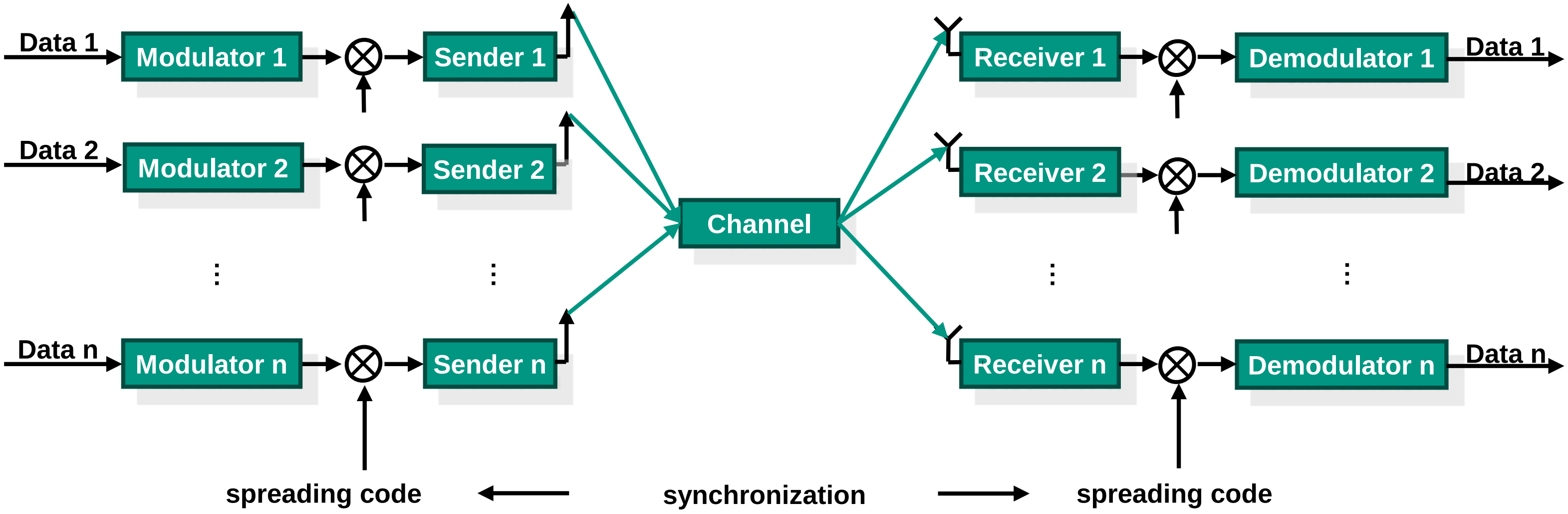
Examples of multiplexing II

- **Frequency division multiple access (FDMA)**
  - Representation using multiple frequencies at the same location, at the same time, modulation



- **Wavelength Division multiplex (special form of frequency division**)
  - transmission of multiple wave lengths (colors) over a single optical waveguide.
  - currently up to 160 individual wavelengths possible
  - especially suitable for wide area networks (WAN)
  - within a single wavelength time multiplex possible – currently data rates of 100 Gbits/s feasible (theoretically resulting in 16 Tbit/s per fiber)

Code Division Multiple Access (CDMA)



- All senders use the same frequency band and send at the same time
  - Distinction needed → Spreading codes

- Different nodes can be distinguished by their individual spreading codes.
  - Logic combination with the correct spreading code will extract the original data
  - Transmissions of other nodes will appear similar to white noise and can be filtered out.

Spreading codes

- Senders use *orthogonal sequences* during the encoding stage, to ensure separation of signals.
- Consider an example where an orthogonal sequence contains N elements (where N is the number of nodes). The properties of orthogonal sequences are
  - The inner product of two equal sequences is N
  - The inner product of two different sequences is 0.
- The signal received is correlated with a sequence present at the desired receiver. If the receiver has the same sequence as the sender, there is high correlation.
- Example for Spreading Codes: Walsh functions (Details will be explained in exercises)

Formation rule for Spreading codes: Walsh functions

Orthogonal sequences can be generated using Walsh Table. Rules to create Walsh functions are

1. Start with “+1” as element a_{11} of a 2x2 matrix

$$\begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{matrix} = \begin{matrix} +1 & a_{12} \\ a_{21} & a_{22} \end{matrix}$$

2. The element a_{11} is repeated at position a_{12} and a_{21}
3. At position a_{22} the inverse of a_{11} is inserted.

$$\begin{matrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{matrix} = \begin{matrix} a & a \\ a & \bar{a} \end{matrix} = \begin{matrix} +1 & +1 \\ +1 & -1 \end{matrix}$$

4. This matrix now becomes element a_{11} of a new 2x2 matrix, where the steps 2. and 3. are repeated
5. The steps 2. to 4. are repeated until the required length of the Walsh function is reached

Formation rule for Walsh functions

+1	+1	+1	+1	Walsh Function f_0
+1	-1	+1	-1	Walsh Function f_1
+1	+1	-1	-1	Walsh Function f_2
+1	-1	-1	+1	Walsh Function f_3

In general

Walsh Function f_0

Walsh Function f_1

⋮

Walsh Function f_n

Properties of Walsh functions

- All Walsh functions (except function 0) contain the same amount of +1 and -1
- The functions are orthogonal to each other, the cross correlation of two different functions equals to 0
  - Example: Walsh function 1 and 2

+1 -1 +1 -1

times

+1 +1 -1 -1

+1 -1 -1 +1 Sum = 0

- The autocorrelation of a function equals to eight (in this example)
  - Example: Walsh function 1

+1 -1 +1 -1

times

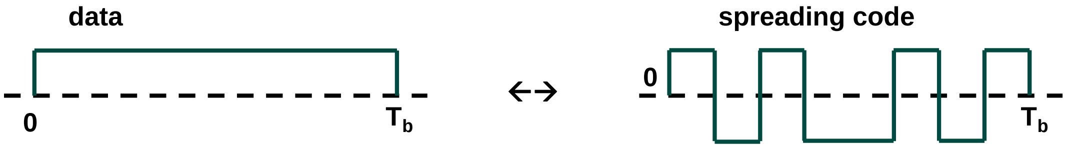
+1 -1 +1 -1

+1 +1 +1 +1 Sum = 4

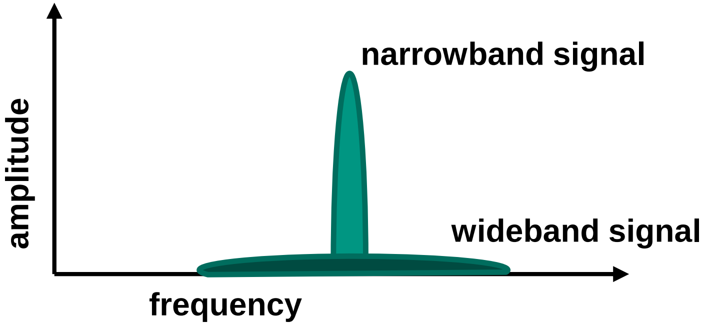
CDMA : What does spreading mean?

- Spreading Code
  - One bit is encoded with multiple sub-bits (Chips)

  → Increase of bandwidth by spreading factor (SF)



- frequency spread:
  - Transmitted signal is distributed over a wide spectrum using the *spreading code (e.g. Walsh function).*



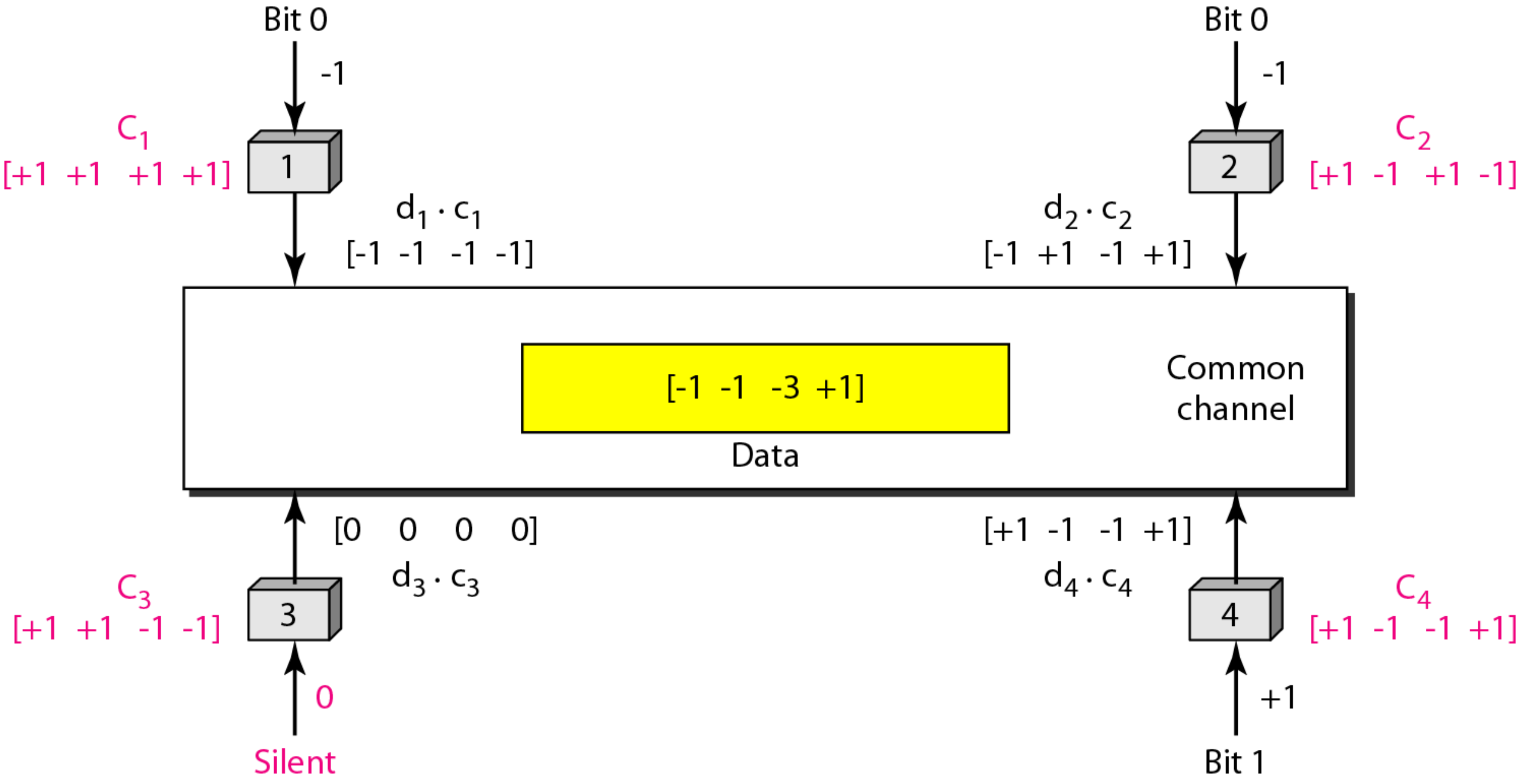
Data transmission with CDMA

- Multiple nodes want to send data at the same time using the same frequency band
  - Every node has a unique spreading code assigned to it

- Every bit to be send is encoded with the spreading code assigned to the corresponding sender
  - Transmission of „1“: Spreading sequence is send as is
  - Transmission of „0“: Spreading sequence is inverted

- Spreading sequences of all nodes are superimposed and form **one** signal on the media
  - Attention: The signal on the media is now multi-valued and not binary any more!

Data transmission with CDMA : Example



“Sharing channel in CDMA“

Source: *Behrouz A. Forouzan, “Data Communications and Networking”, Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007 Chapter 12, Figure 12.26*

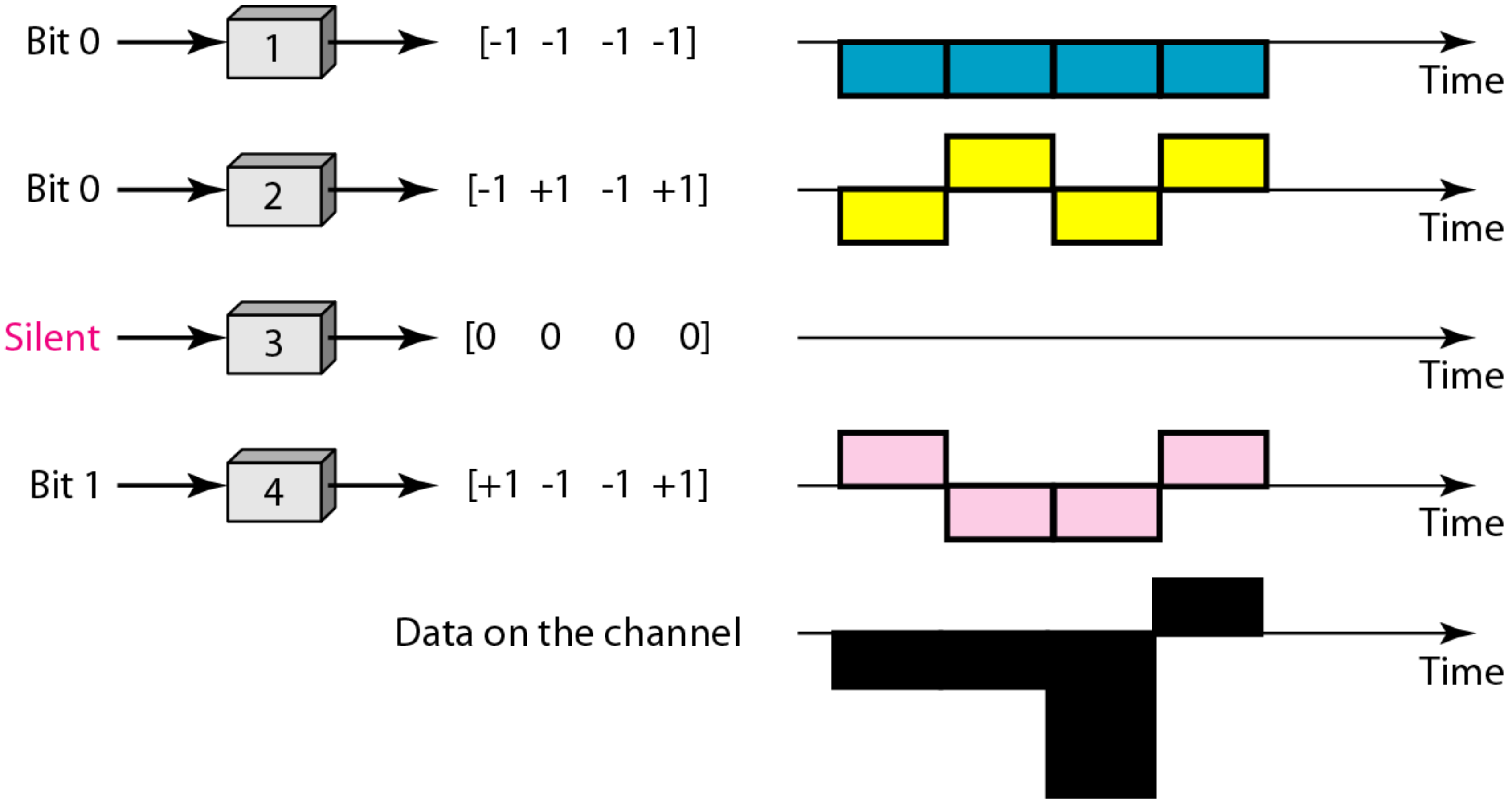
Reception with CDMA

- On receiver side the transmitted signal is correlated with the Walsh function that has been assigned to the node

- As the Walsh functions of the other nodes are orthogonal, only the signal that is meant for the receiver is filtered
  - If the result of the correlation is +N, a “1” has been send
  - If the result of the correlation is -N, a “0” has been send

- For real data transmission a decision based on the exact value is not possible any more because of interferences and other disturbances that distort the signal. Therefore the signal has to be within in a predefined interval (acceptance band) to be interpreted as “1” or “0”.

Data on the Channel

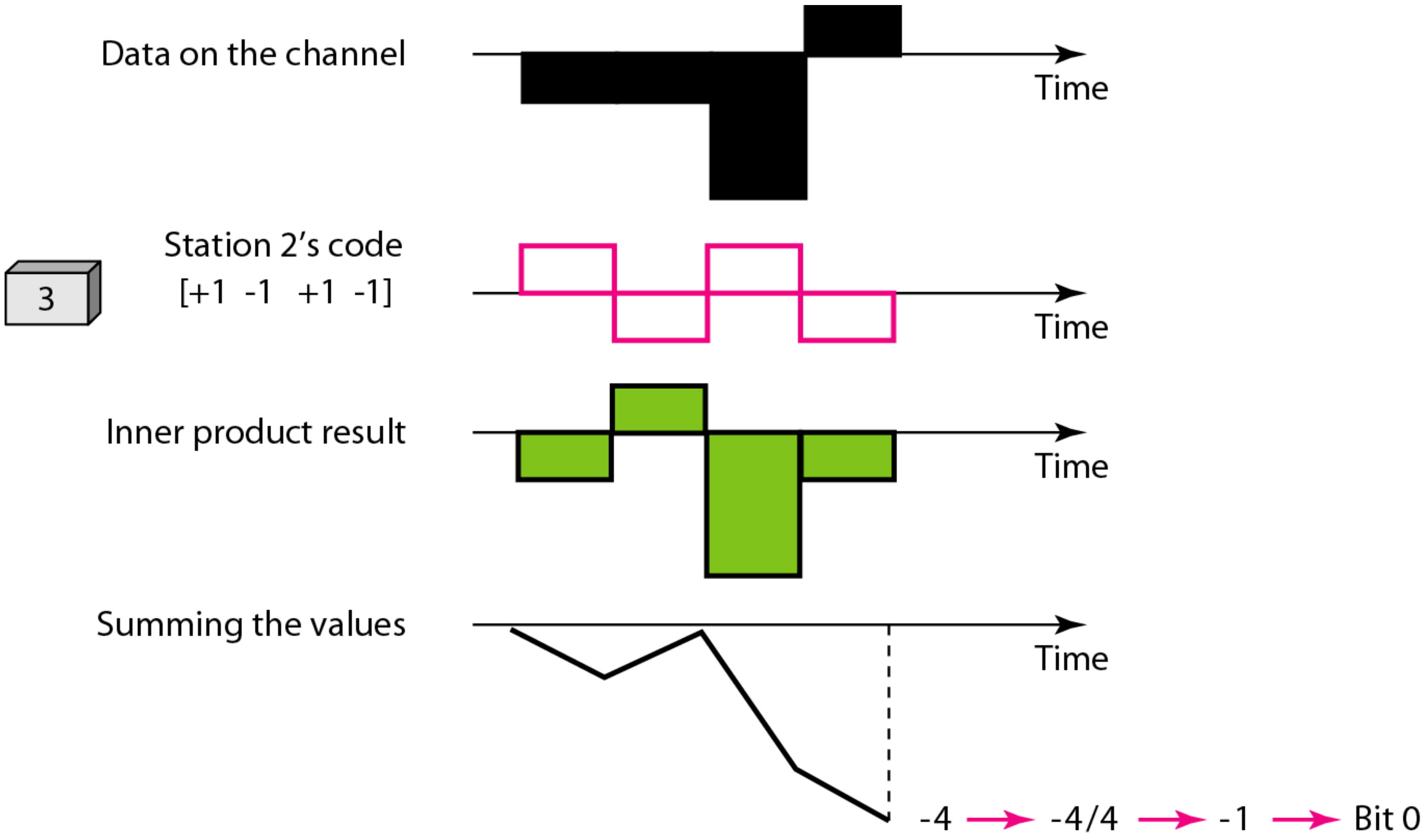


“Digital signal created by four stations in CDMA“

Source: *Behrouz A. Forouzan, “Data Communications and Networking”, Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007 Chapter 12, Figure 12.27*

Reception with CDMA

- Station 3 is listening to Station 2



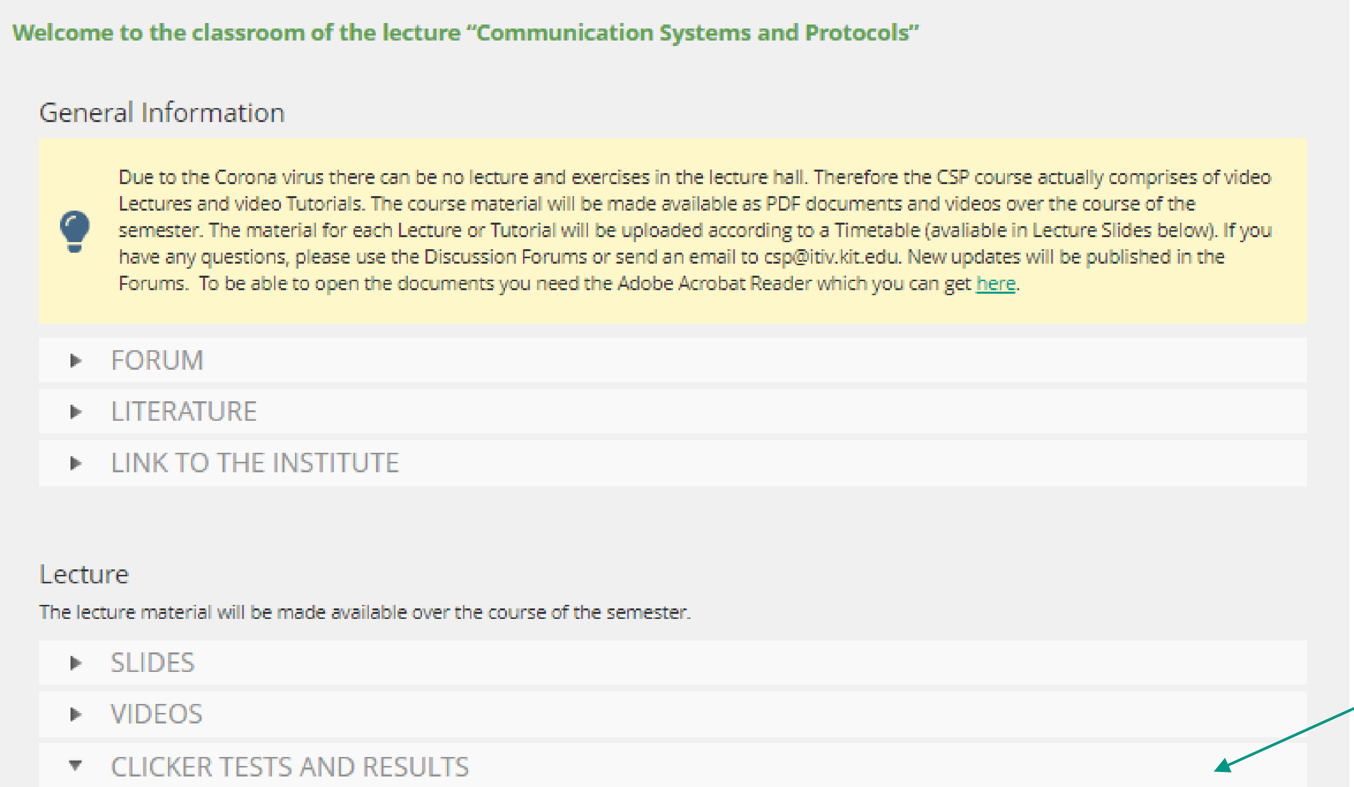
"Decoding of the composite signal for one in CDMA"

Source: *Behrouz A. Forouzan, "Data Communications and Networking", Fourth Edition, ISBN: 0072967757 ,Copyright year: 2007 Chapter 12, Figure 12.28*

Advantages and Disadvantages

- Pro:
  - More robust against narrowband disturbances because signal energy is distributed on a broader spectrum
  - Lower sensitivity against interferences from other nodes
  - No guard intervalls required → better usage of spectrum
- Contra:
  - More effort required on sender and receiver side
  - Requires channel with larger bandwidth

Participate in Clicker Test 4a



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

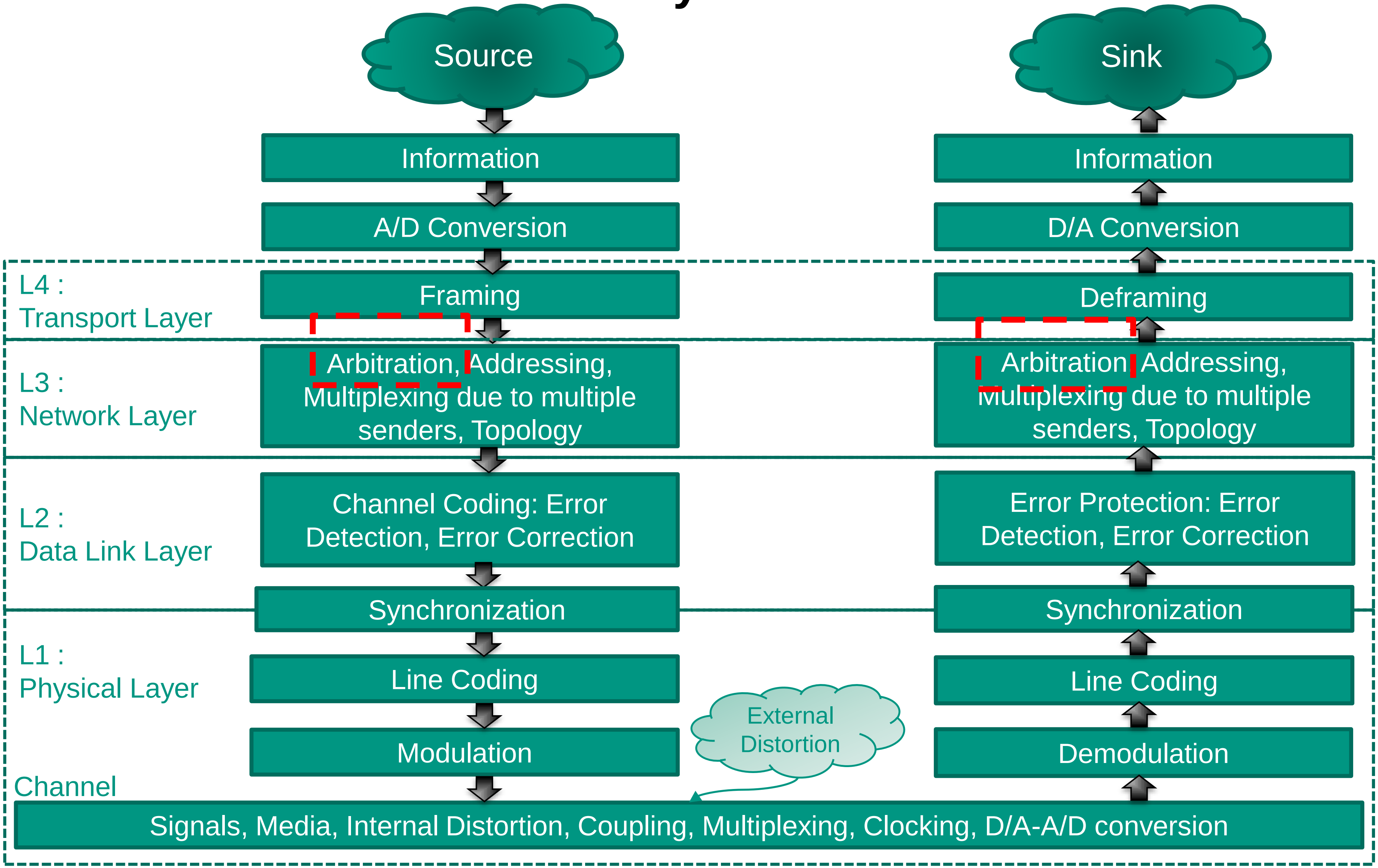
Arbitration

Contents

- Arbitration with additional signal lines (parallel busses)
  - Daisy-Chain
  - Polling
  - Tap Line
  - Self Selection

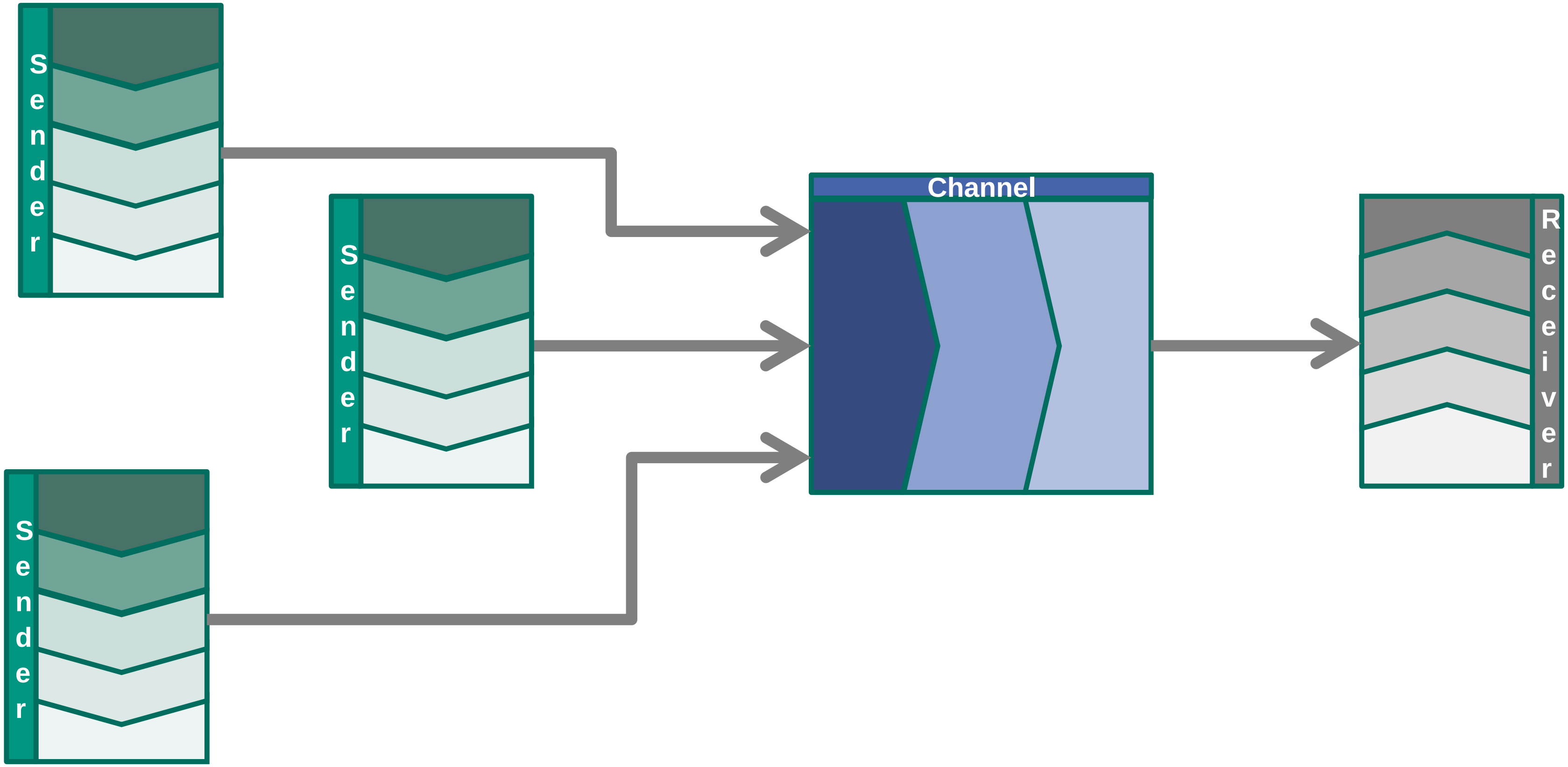
- Arbitration without additional signal lines (serial busses)
  - Token passing
  - Aloha
  - CSMA/CD
  - CSMA/CA

Technical Communication System

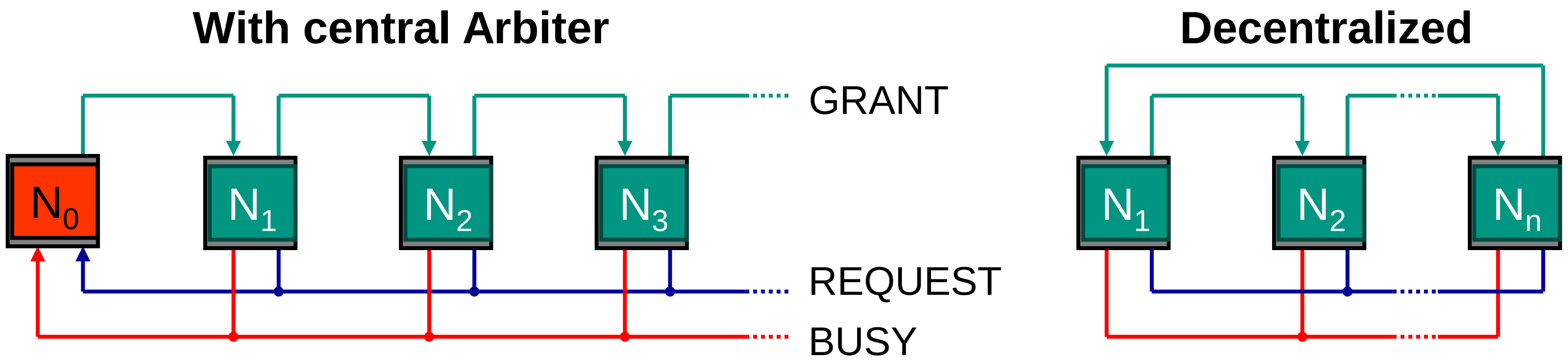

Source
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
Channel
External Distortion
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Dynamic Multiplexing: Arbitration

- Static multiplexing schemes can make bad use of available channel if transmissions are not equally distributed
- Dynamic assignment of channel to one sender → Arbitration



Daisy-Chain

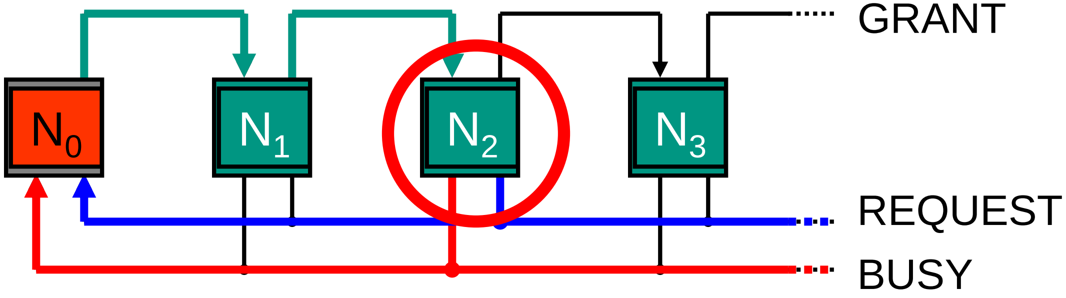


- All nodes that are "ready-to-send" activate the *REQUEST*-line
- If a node N_i receives the *GRANT*-signal it can take over the control of the bus
  - If N_i has data to send, N_i activates the *BUSY*-line and keeps the *GRANT*-signal
  - If N_i has no data to send, it passes the *GRANT*-signal to the next node N_{i+1}
- Centralized: One central arbiter N_0. Chain always starts with N_0.
- Decentralized: Chain always starts with the node that was the last bus master
- Properties:
  - relatively slow
  - easy extensions (scalable)
  - low hardware footprint

→ Centralized daisy-chaining unfair, as priorities are fixed by the wiring

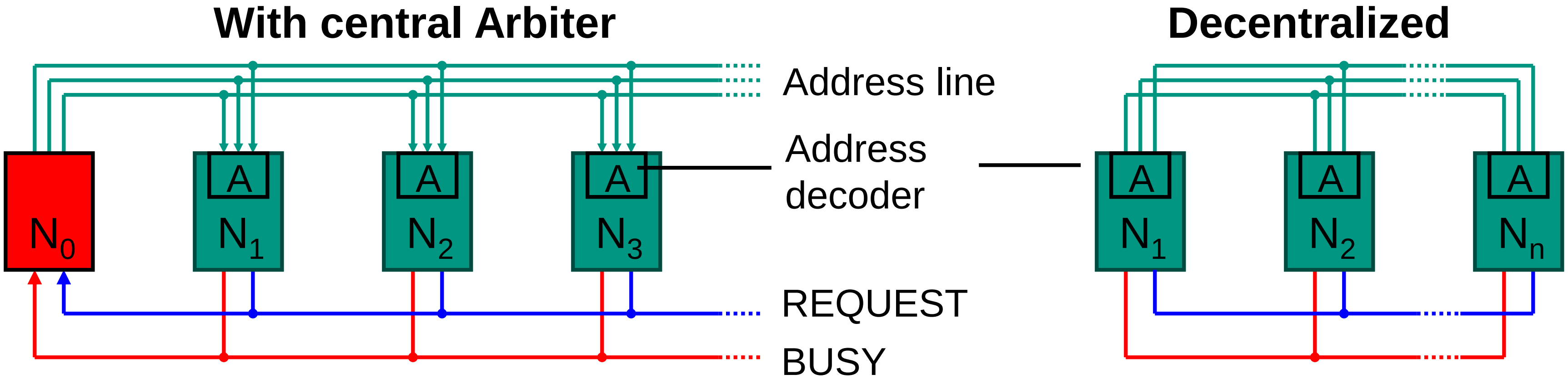
Daisy-Chain

- Example:



- N_2 has a transmit request (wants to send data)
- N_2 activates *REQUEST*-line
- Arbiter N_0 asserts *GRANT*-signal
- N_1 has no data to send and passes *GRANT*-signal
- N_2 asserts *BUSY*-signal, removes *REQUEST* and takes control of the bus
- N_2 de-asserts *BUSY*-signal thus freeing the bus for other transmissions

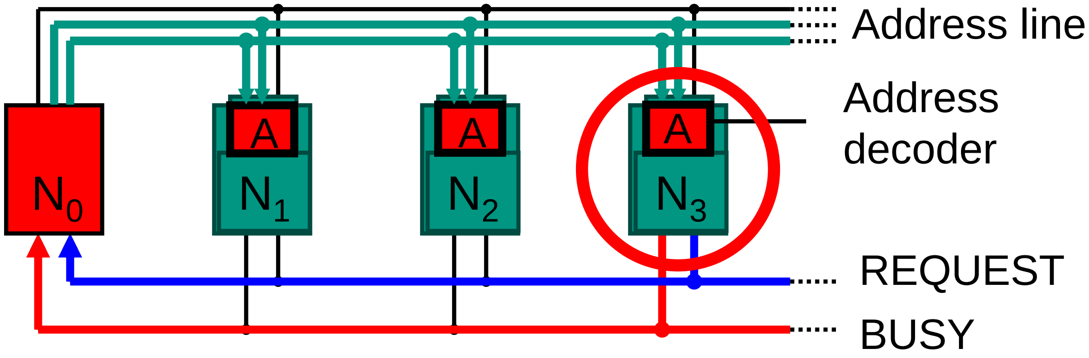
Polling



- Transmit requests are forwarded to an arbiter over a common signal line
- Arbiter then addresses the bus nodes sequentially following their priority
  - decentralized solution: last bus master acts as arbiter
- If the address of a node with a transmit request is set, the node asserts a *BUSY*-signal and becomes bus master.
- Properties:
  - fair and priority based arbitration possible
  - medium hardware resources
  - slow arbitration

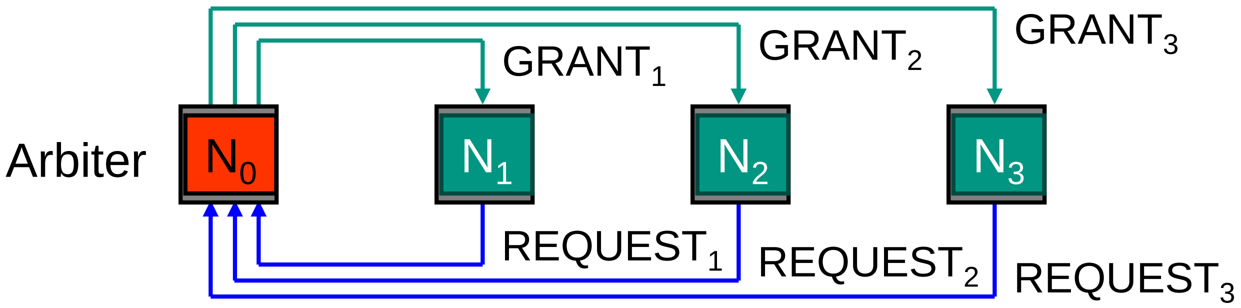
Polling

- Example:



- N_3 has a transmit request (wants to send data)
- N_3 activates *REQUEST*-line
- Arbiter N_0 polls N_1 (checks for transmit request)
- Arbiter N_0 polls N_2
- Arbiter N_0 polls N_3
- N_3 has issued a REQUEST and asserts now the BUSY-signal to take over the bus

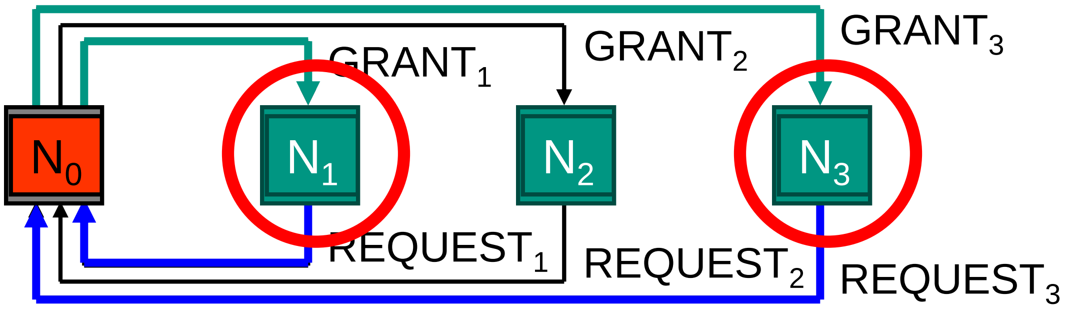
Tap Line



- All nodes are connected with a central arbiter over individual tap lines
- For a transmit request dedicated *REQUEST*-lines are asserted with an according signal
- The central arbiter issues *GRANT*-signals according to an arbitrary scheme
  - Priority table
- Properties:
  - fair and priority based arbitration possible (priorities can be changed easily)
  - high demand of hardware resources (expensive)
  - fast arbitration due to dedicate *REQUEST*- and *GRANT*-lines
  - Scalability to an arbitrary number of nodes somewhat limited

Tap Line

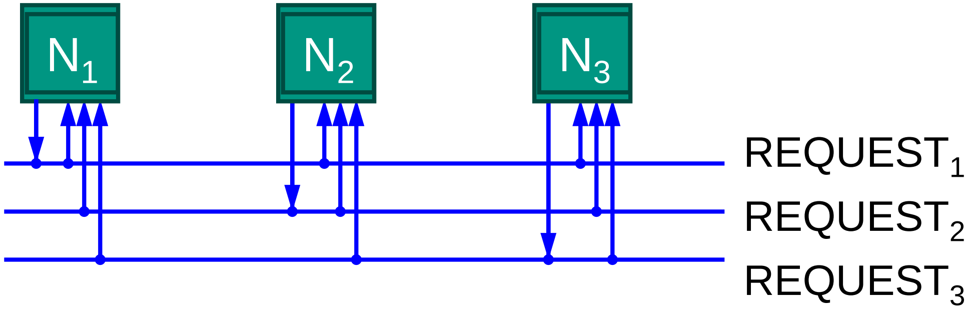
- Example:



- N$_1$ and N$_3$ each have a transmit request
- Both activate their *REQUEST*-line
- Arbiter N$_0$ signals a *GRANT* first to N$_1$
- N$_1$ takes over the bus to transmit data and subsequently removes its *REQUEST* signal
- Arbiter signals a *GRANT* to N$_3$
- N$_3$ takes control of the bus

Self-Selection

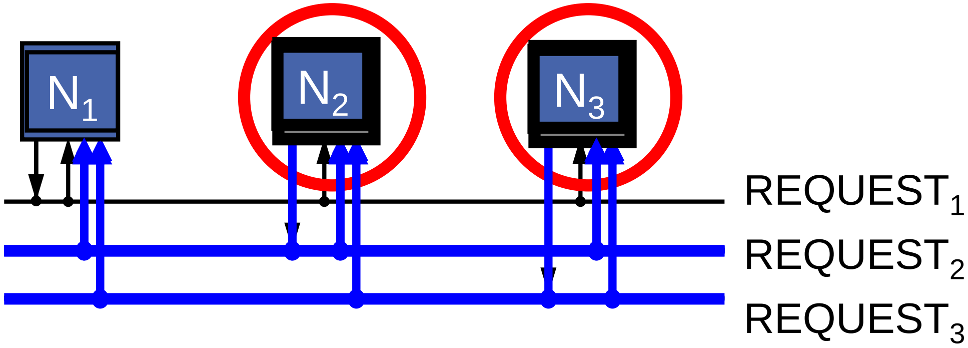
- Decentralized Arbitration by self-selection



- One REQUEST line per node that can be read/detected by all other nodes
- Shared arbitration function is implemented in every node (redundancy)
  - all nodes agree on what node takes control over the bus
- Properties:
  - Fair and priority based scheme possible
  - Fast due to dedicated REQUEST-lines
  - more expensive than daisy-chaining, cheaper than centralized arbitration over tap lines
  - Scalability to an arbitrary number of nodes somewhat limited

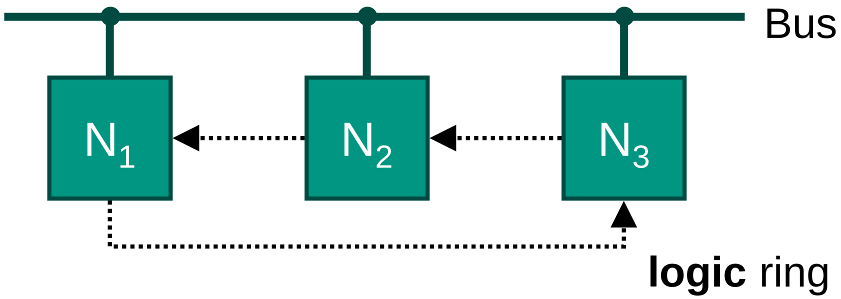
Self-Selection

- Example:



- N_2 and N_3 each have a transmit request
- Both activate their *REQUEST*-line
- All internal arbiters decide to grant the bus to N_3 first
- N_3 takes over the bus to transmit data and subsequently removes its *REQUEST* signal
- Afterwards all nodes have the common understanding that N_2 is granted the bus
- After finishing the data transfer N_2 removes its *REQUEST*-signal from the *REQUEST*-line

Token-Passing

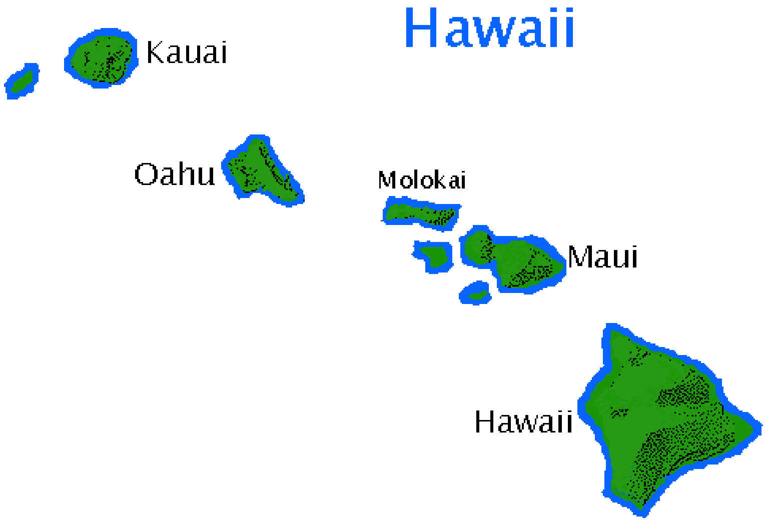


- All nodes are connected to the same (serial) bus
- Every node has a unique address
- Every node has stored the address of its (logic) successor, thus creating a logic ring
- A “free-token” is rotating on the ring (by sending a “Token Packet”)
- A node holding the token is allowed to send data
- If a node is holding the token, but has no data to send, it will pass the token to its successor
- Procedures to inject and remove nodes have to exist

- Properties:
  - flexible, fair, and real-time capable scheme because each node can transmit after a maximal interval of (($\texttt{n-1}$) $\cdot \texttt{max\_data\_length}$) time units

Aloha

- Properties: - random bus access
  - not real-time capable (undefined idle time)
- First media access scheme based on a randomness-strategy
  → first use in a (terrestrial radio) network on Hawaii
- Each node with a transmit request will output data on the bus immediately (without being granted permission and without checking if medium is free)
- Problem: signal distortion if two nodes are sending data simultaneously
  - Detection of conflicts: receiver does not send an acknowledge at the end of a message
  - Correction of conflicts: data packet is resend until an acknowledge is received
  - → large latencies possible
  - → channel utilization very high due to multiple data transmissions and acknowledge



→ Scheme is only adequate if channel utilization is lower than 20%

CSMA (Carrier Sense Multiple Access) I

- Extension to Aloha: before transmitting data the channel is sensed for ongoing
- Two slightly varying schemes exits:

1. Non-persistent Scheme

- For each Transmit Request the channel is checked to be free:
  - Channel free: transmit immediately
  - Channel occupied: wait for a random time span and retry (this reduces the probability of two nodes starting a transmission simultaneously)
- Every iteration increases the random waiting time
- Channel might be unused during the node's waiting time

CSMA (Carrier Sense Multiple Access) II

- Extension to Aloha: before transmitting data the channel is sensed
- Two slightly varying schemes exits:

2. **P-persistent Scheme**
   - For each Transmit Request the channel is checked to be free:
     - Channel free: send immediately with a probability *p*, delay transmission for a time span *t* with a probability *1-p*

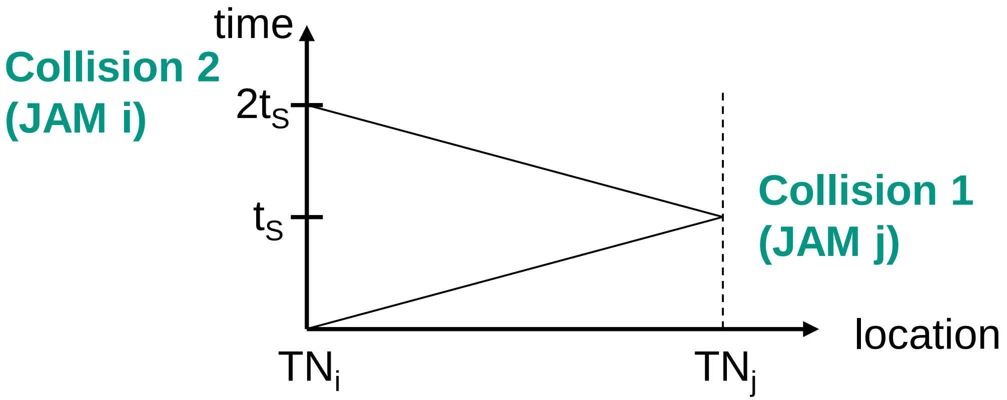
       t is the time for one bit to pass through the communication channel for a node to detect if a second node is trying to send as well
     - Channel occupied: wait until channel is free and restart from the beginning

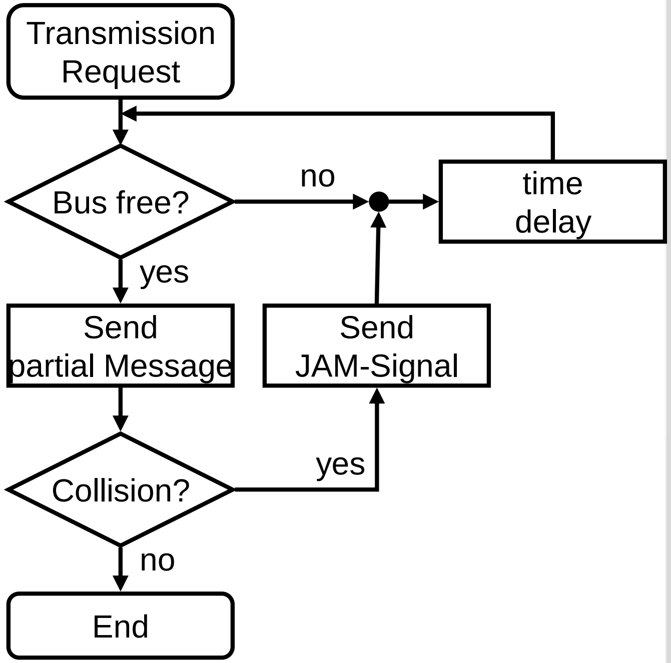
- Disadvantages of both schemes (as with Aloha):
  - No direct collision detection, collisions can only be identified through a missing acknowledge (only after a full data packet has been sent).

CSMA/CD (CD = Collision Detect)

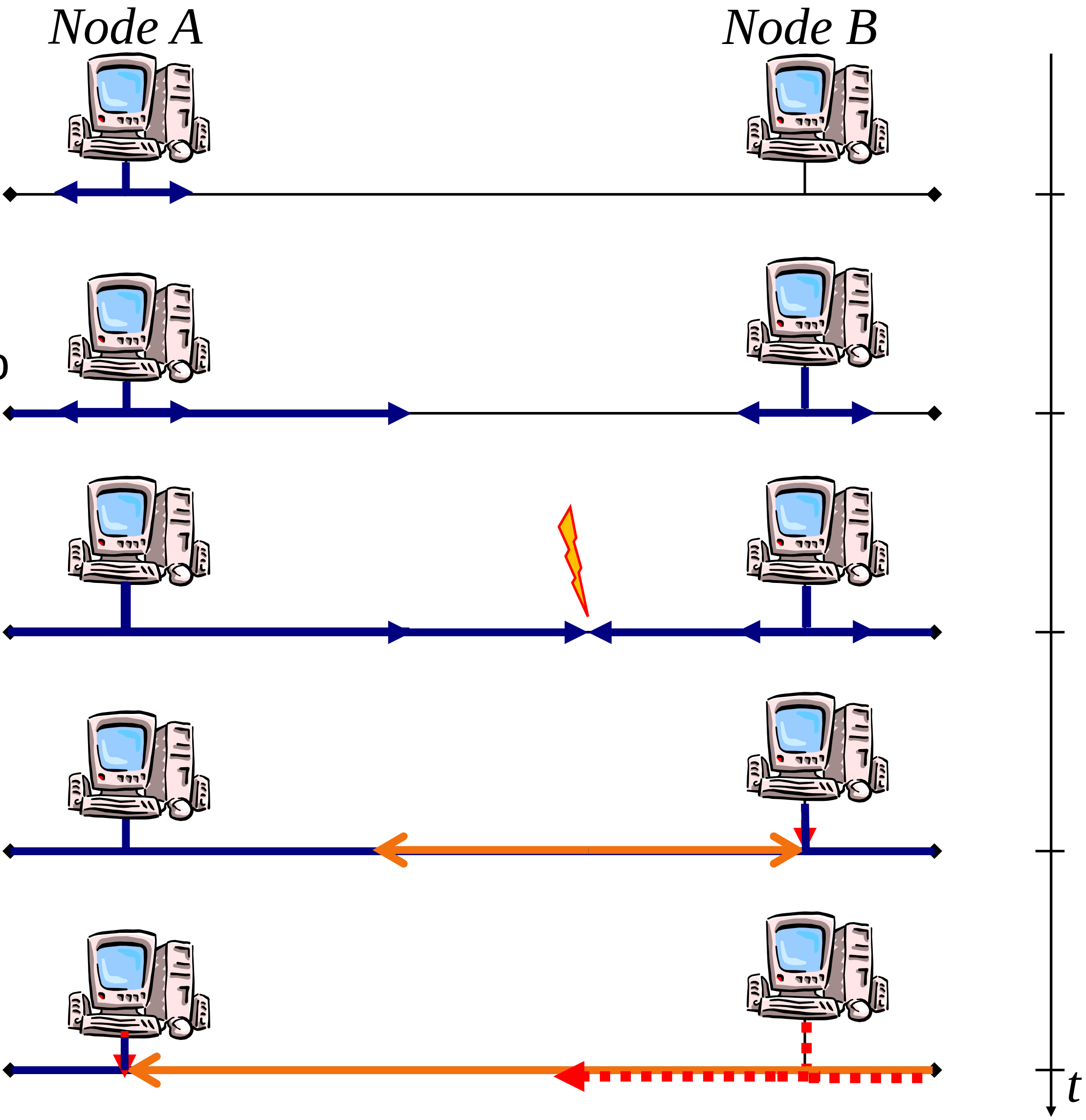
- Direct detection of collisions
- Sender is always reading the channel and checks if the signal sent is identical to the one being read.
  - Fast detection if a collision has occurred
- Sender detects a collision:
  - Transmission of a JAM signal, transmission is ceased
  - Node that is transmitting simultaneously is detecting the JAM signal and ceases its transmission as well
- Receivers will discard data as soon as JAM signal occurs

- Collision is detected after a maximal propagation time t_S
  - Minimal packet length: 2 t_S, maximum wire length must be defined.





Exemplary Sequence of CSMA/CD



- Node A starts to send, because Medium is free.
- Node B starts to send, because Medium seems to be free.
- A collision of data packets occurs.
- Node B detects the collision, ceases data transmission and sends JAM-signal.
- Node A detects collision and also sends a JAM-signal.

Restrictions for cable length in CSMD/CD

- Sender has to recognize a collision in order to retransmit data

- Length of the media and duration of sending are related

$$2 \cdot s_{max} = v \cdot t_{Frame} \qquad with \qquad \begin{aligned} s_{max}&: Maximum\ length\ of\ media \\ v&: signal\ propagation\ speed \\ t_{Frame}&: Duration\ of\ data\ transmission \end{aligned}$$

- In short: The data to be send has to be long enough for the signal to travel twice the media during sending time.

- JAM Signal: Sequence of 1s and 0s (hex A)
  - Purpose: make sure that all receivers recognize the collision
  - Is send after a collision has been detected
  - Has to be long enough to be realized by all receivers

Properties of CSMA/CD

- Easy to be extended, no configuration needed
- Data destruction possible
- Need to discard already sent data after collision has occurred
- Bad channel utilization
  - Rule of thumb: 30%-70%
- No guaranteed Real-Time Capability

CSMA/CA (CA = Collision Avoidance)

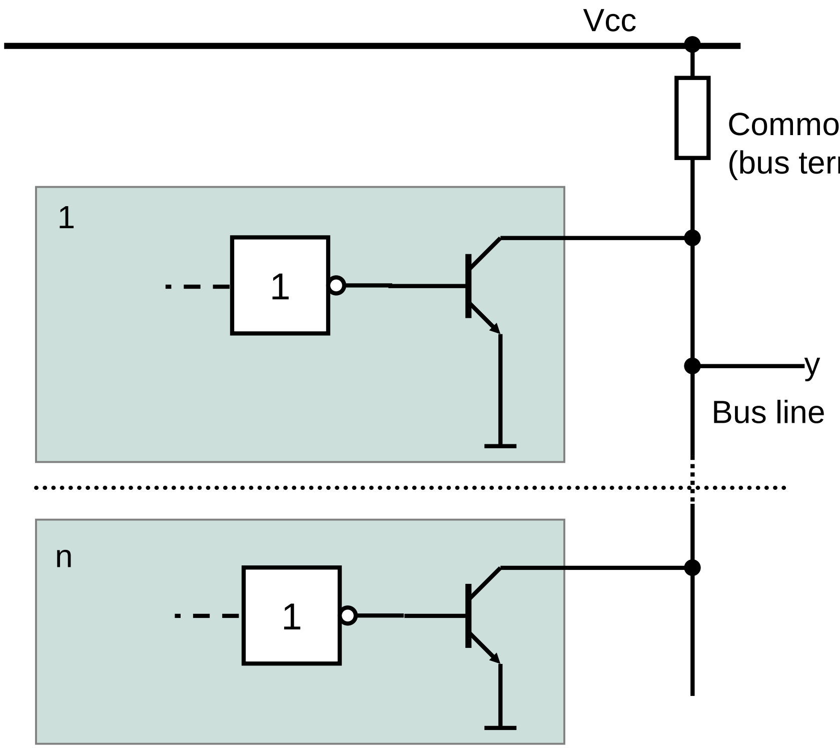
- Avoidance of collisions by priority controlled bus arbitration.
- Every node is assigned an identifier (ID) that equals its priority.

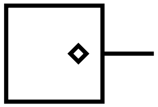
- After completing a transmission on the bus, all nodes with a transmission request start to send their ID. All nodes are connected via wired-OR or via wired-AND respectively.
  - wired-OR: „1“ dominant, „0“ recessive
  - wired-AND: „0“ dominant, „1“ recessive

- A transmission starts with the most significant bit (MSB).
  - Each sender monitors the bus level during each bit being send
  - As soon as the bit currently being read from the bus is not identical with the bit send by the node, the node retreats and retries the transmission later.

Recapitulation: Open collector driver

- The collector of each of the output transistors remains unconnected
- All bus members share one collector resistor
- The output is only HIGH if all transistors cut off
  - low value (GND) on the bus line is dominant value



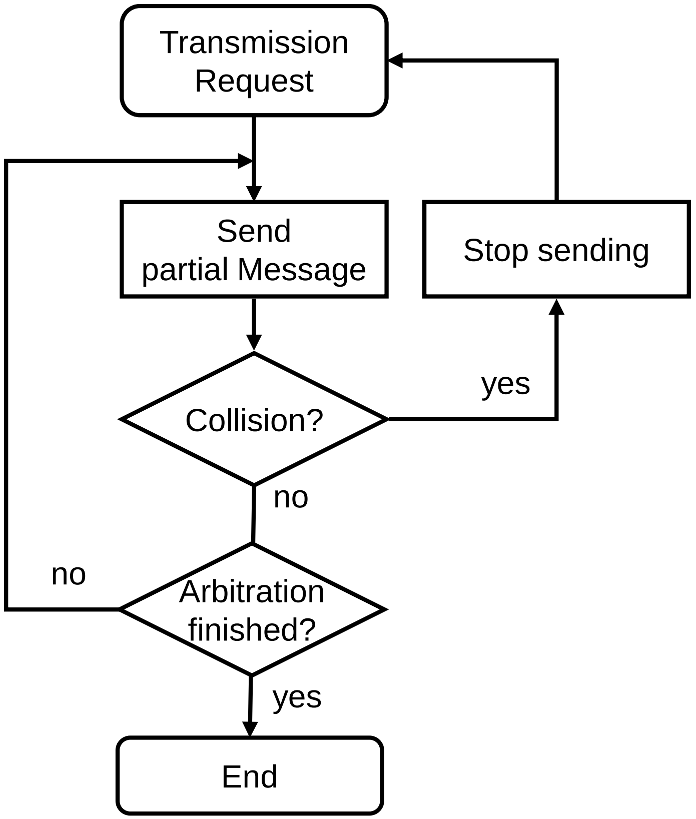


Switch symbol of a open collector output

Arbitration process for CSMA/CA

- Requirements
  - Unique ID per message/node
  - All nodes start arbitration at the same time
  - Bitwise arbitration
  - Bit is long enough, so that all nodes can read it

- Procedure
  - Send arbitration ID bit per bit
  - If data on bus is the same as sending data → continue arbitration
  - If data is different → arbitration lost, withdraw from bus



Simultaneity

- **Problem:** It is necessary for all bus nodes in the cluster to read the bit being send (independent of the distance to the sending node), before (!) a new bit is put onto the bus.

- **Resulting requirement:** signal propagation time t_S is negligible small compared to the digit length (bit time) t_B:

$$\left[t_s = \frac{l}{v}\right] << \left[t_B = \frac{1}{TR}\right]$$

$$l = \text{length of the bus line}$$

$$v = \text{propagation velocity}$$

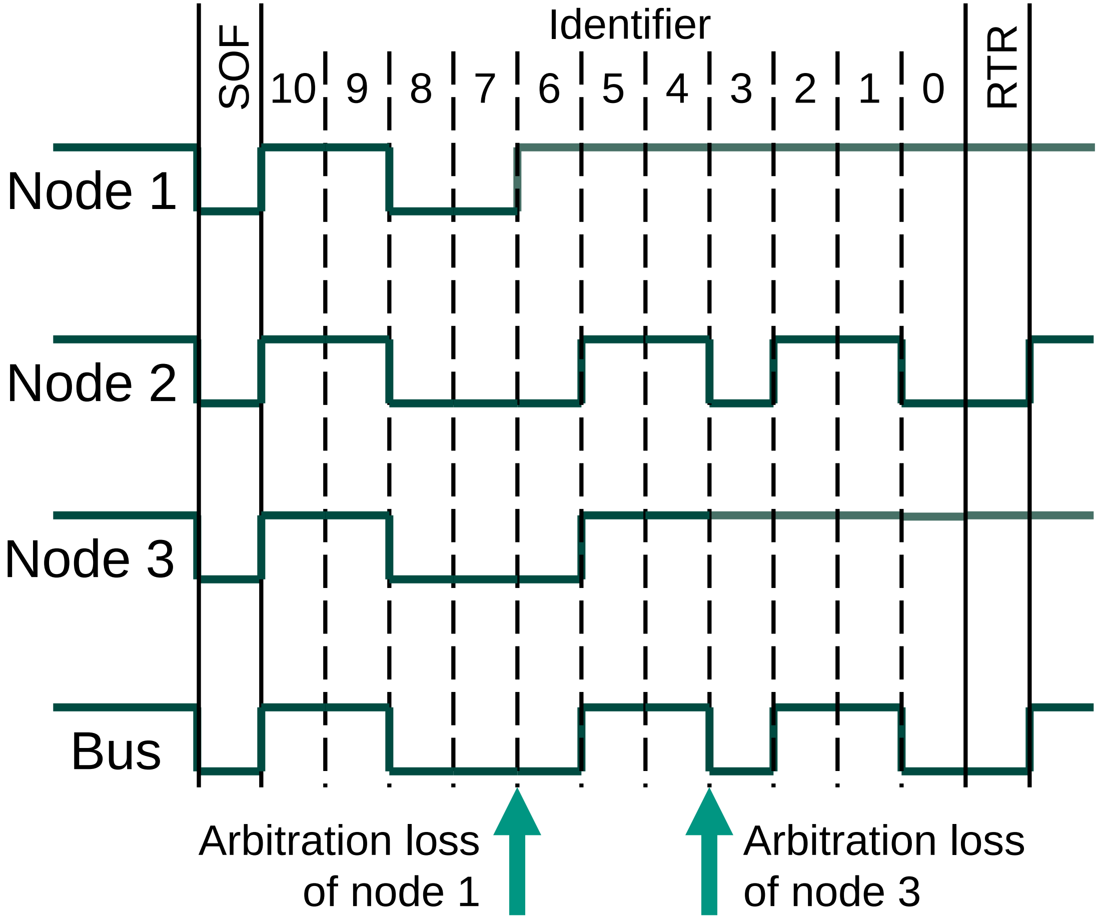
$$TR = \text{transmission rate}$$

Properties of CSMA/CA

- No data destruction
- No need to discard already sent data
- 100% channel utilization is possible
- Limited length of bus and/or transmission rate
  - due to simultaneity
- Very limited Real-Time Capability
  - If the packet length is finite, the node with the highest priority can adhere to real-time constraints.
  - Bus can be blocked if node with highest priority is constantly transmitting.
    - In general each node has to wait after a transmission for a predefined time before transmitting a new message.
  - Other nodes can adhere to real-time constraints as well if waiting time is long enough.

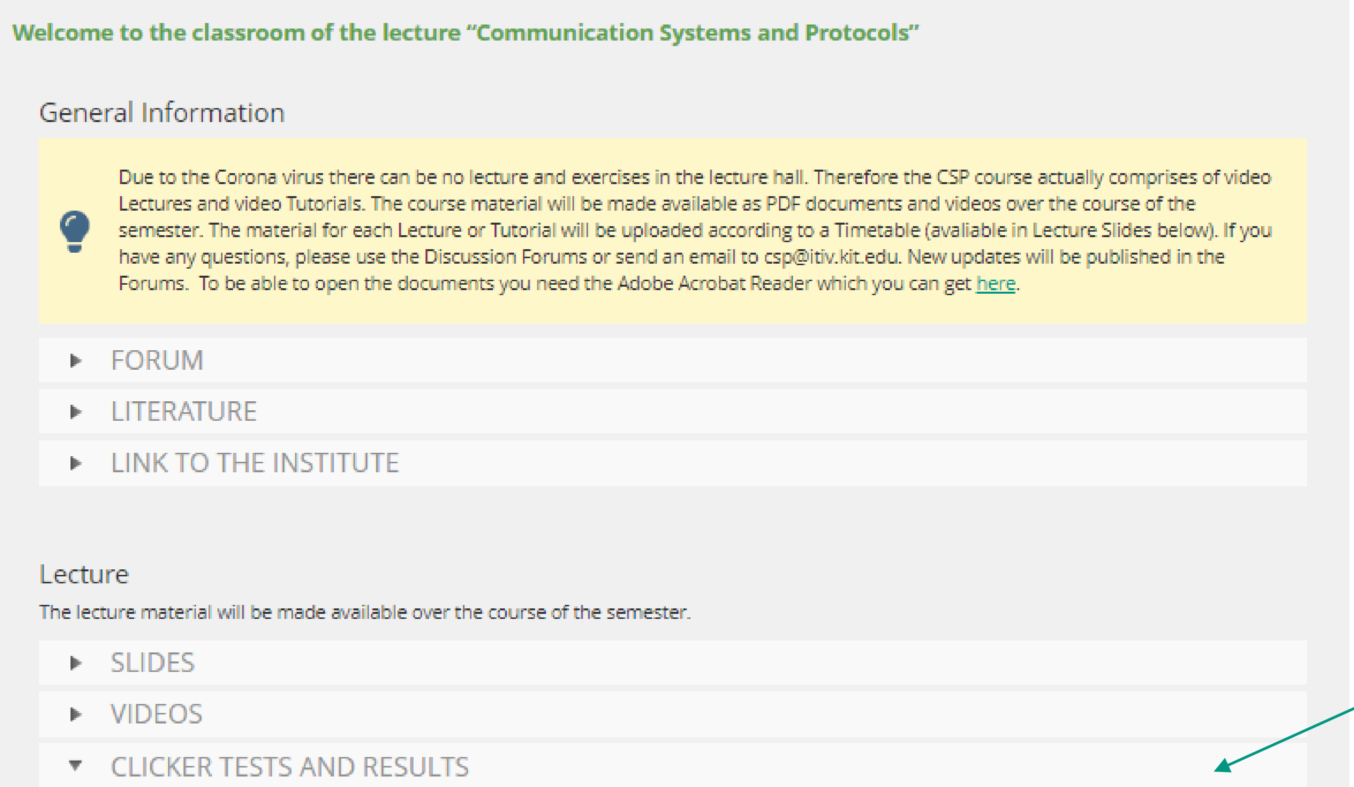
Example: CSMA/CA (CAN-Bus)

- Concurrent bus access of 3 bus nodes, dominant „0“



Node 2 is granted bus access sends its complete message

Participate in Clicker Test 4b



Welcome to the classroom of the lecture “Communication Systems and Protocols”

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

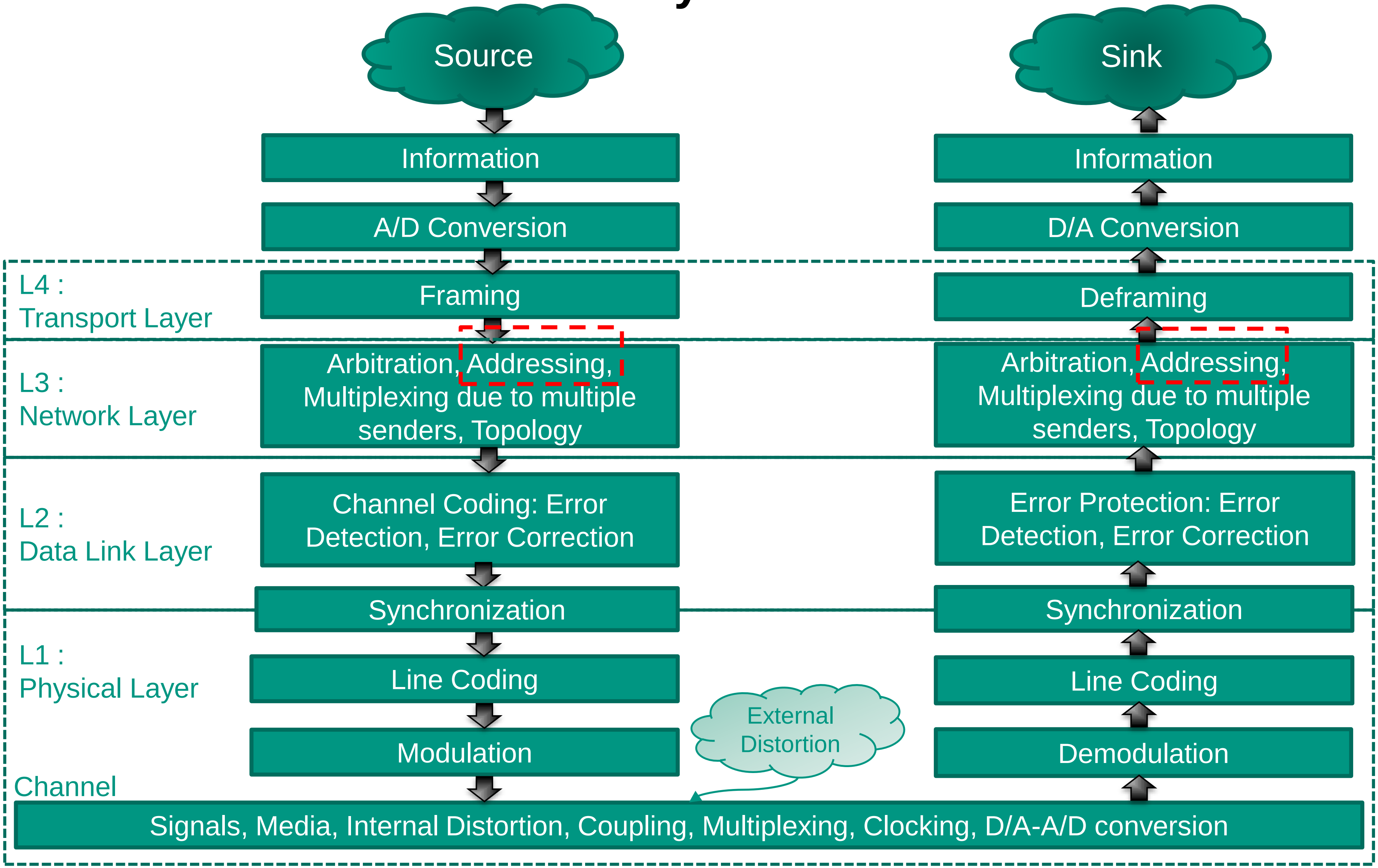
Communication Systems and Protocols

Addressing

Contents

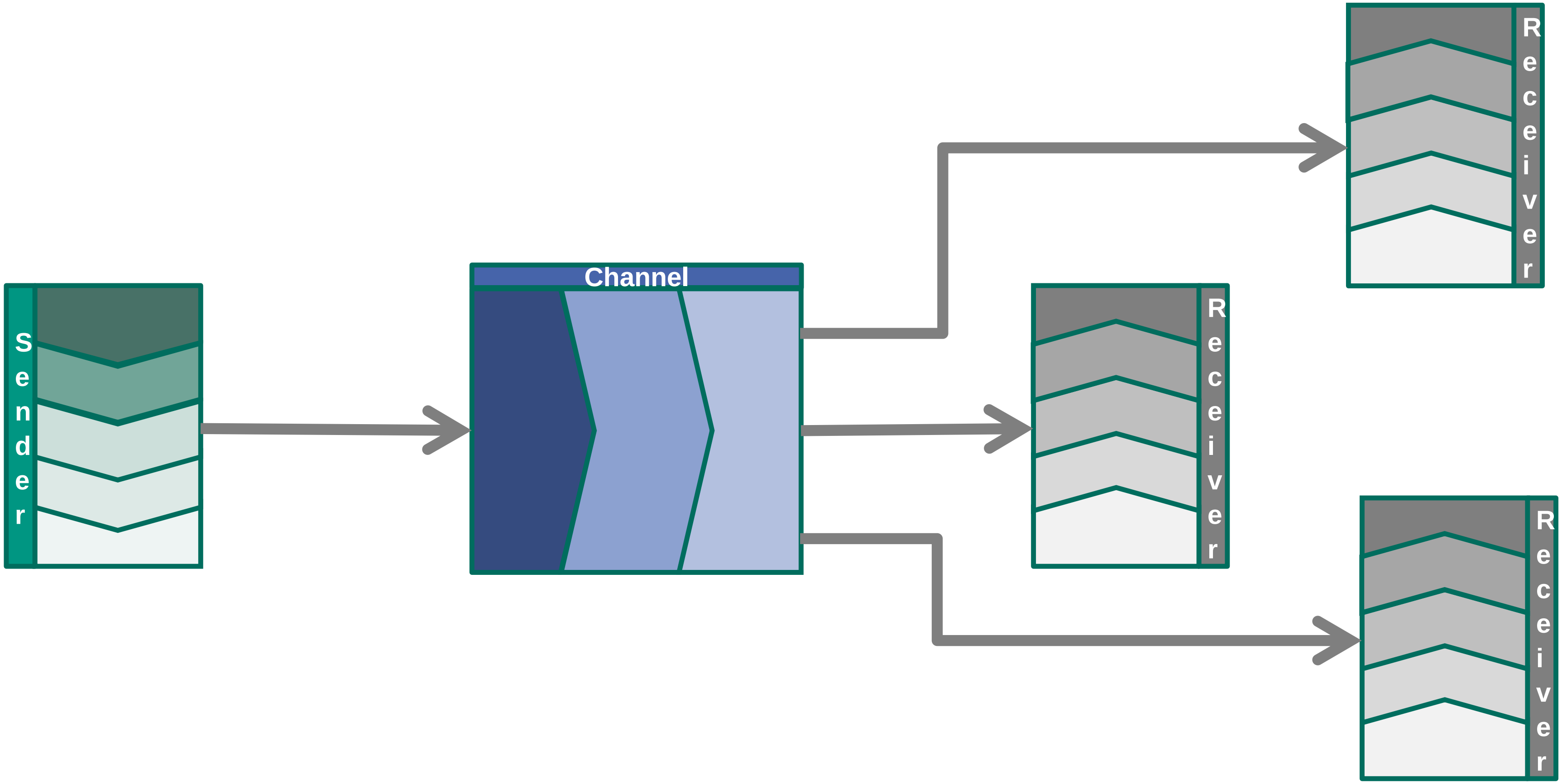
- Node based addressing
- Message based addressing

Technical Communication System


Source
Information
A/D Conversion
L4 :
Transport Layer
Framing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Synchronization
L1 :
Physical Layer
Line Coding
Modulation
Channel
External Distortion
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion
Sink
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation

Multiple Receivers : Addressing

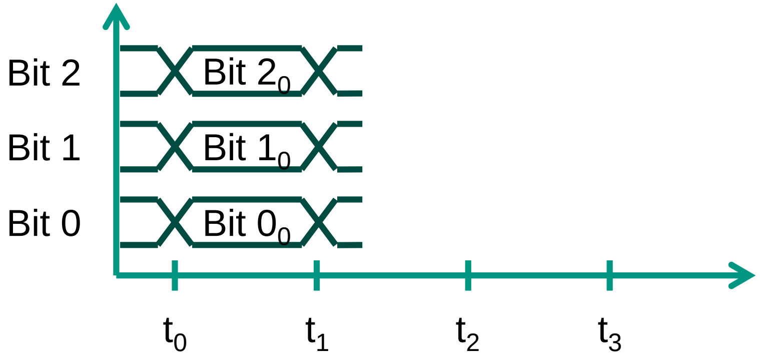
- Multiple receivers can be listening at the same channel
- It has to be distinguished which receiver should receive the actual transmission → Addressing



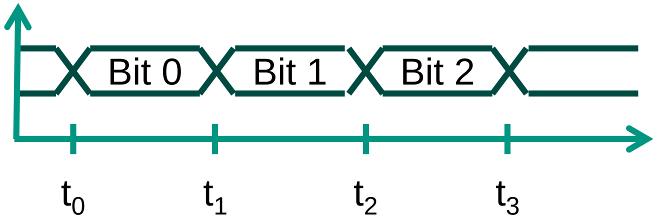
Node based addressing

- Every receiver in the communication system is assigned an individual and unique Identifier (ID, address)

- A transmission is initiated by sending the receiver’s address via
  - Additional signal lines (address lines)

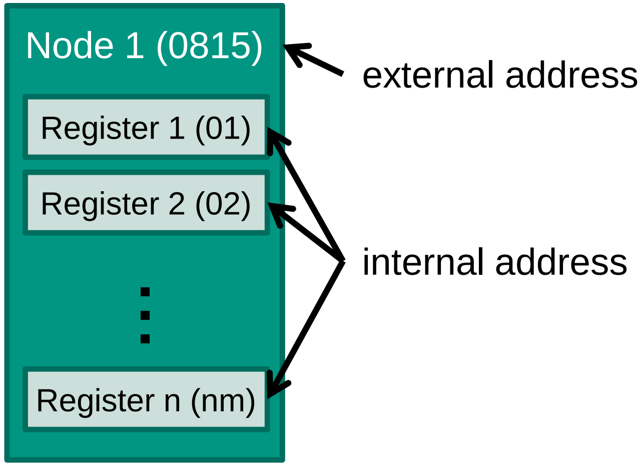


  - Sending the receivers address before sending the data if only one data line is available



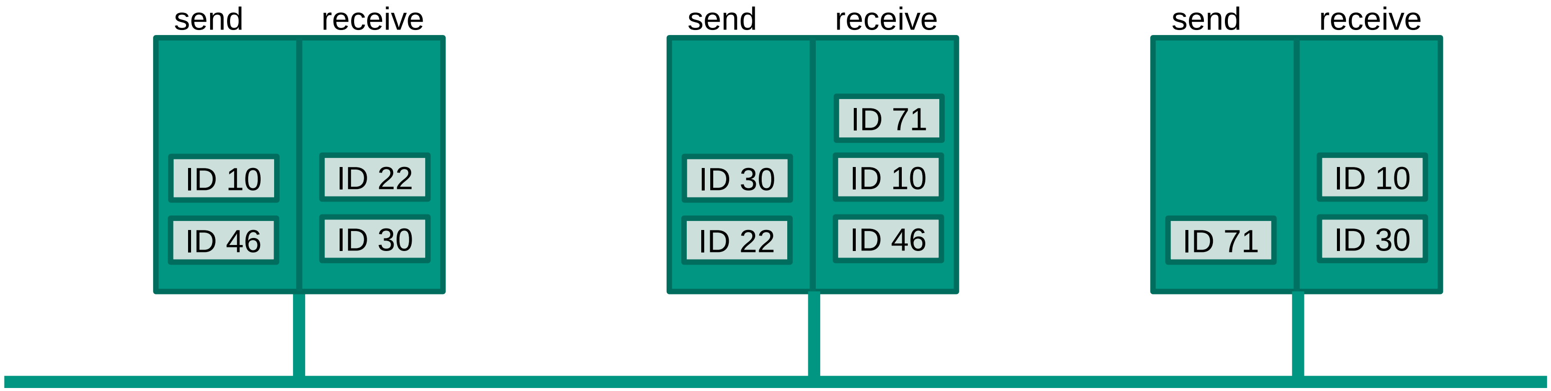
Internal node addressing

- There can also be additional internal addresses within the receivers that are also transmitted over the communication system
  - External Address: Identify the receiver
  - Internal Address: Identify receiver-internal memory or registers



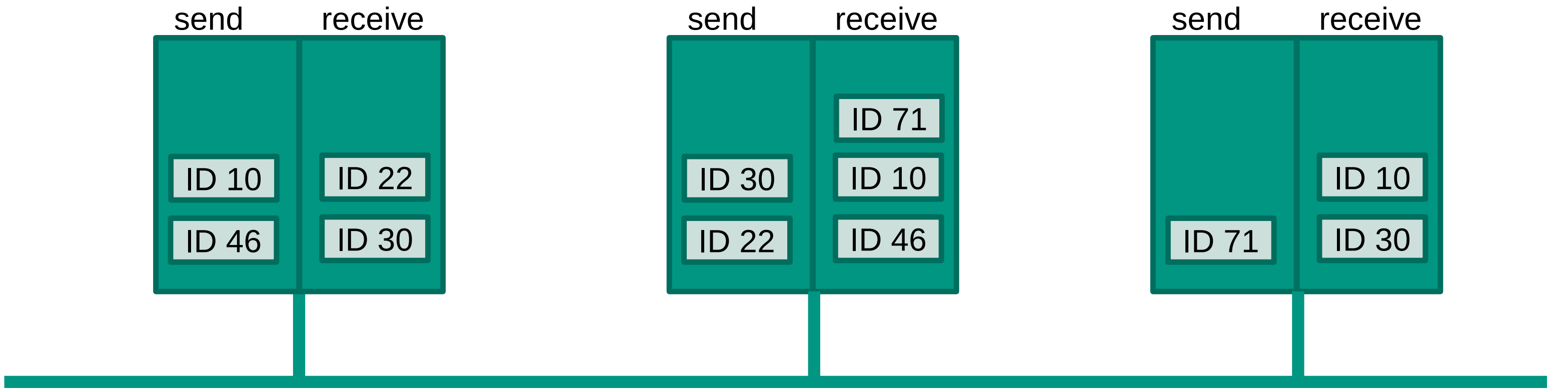
Message based addressing I

- The address is assigned to individual messages with defined content
  - E.g. temperature reading of a sensor, speed of a car
- Message ID is unique in a communication system
  - Each message can only be send by one single sender
  - Each sender can send an arbitrary number of different messages



Message based addressing II

- Every receiver can decide upon this message ID if the message is of interest to him or not
- No dedicated addressing of a receiver required/possible
  - Everyone in the communication system shares the same information → Broadcast



- Example: CAN

Karlsruher Institut für Technologie

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

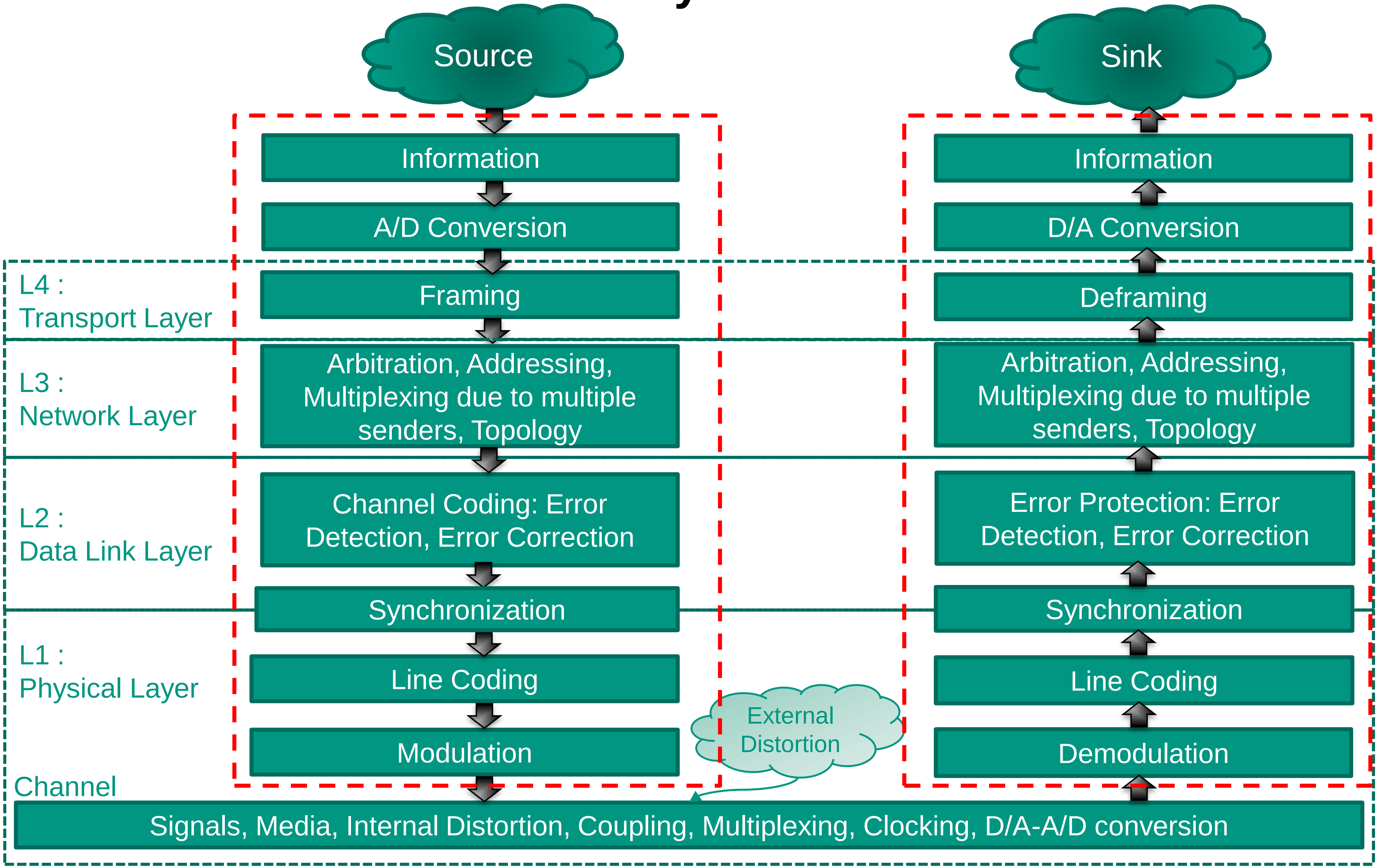
Communication Systems and Protocols

Bus systems

Contents

- Components of Bus Systems
  - Node
  - Interface
  - Bus
  - Cluster
- Parallel vs. Serial Busses
- Topologies

Technical Communication System


Source
Sink
Information
A/D Conversion
Framing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Channel Coding: Error Detection, Error Correction
Synchronization
Line Coding
Modulation
Information
D/A Conversion
Deframing
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Error Protection: Error Detection, Error Correction
Synchronization
Line Coding
Demodulation
L4 : Transport Layer
L3 : Network Layer
L2 : Data Link Layer
L1 : Physical Layer
External Distortion
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

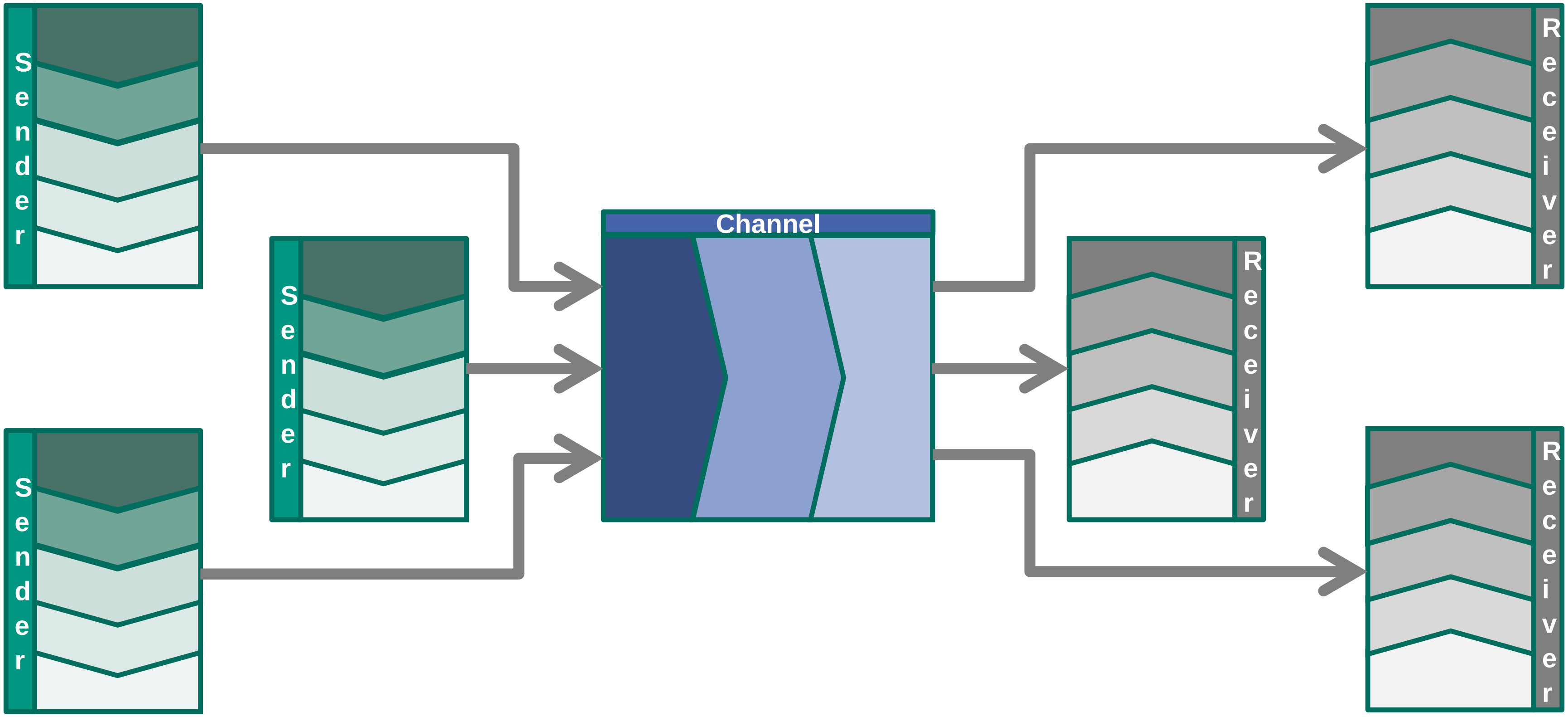
Possible setups for Communication Systems

- 1:1 Direct communication between two partners
  - Examples: RS-232, Telephone
- 1:n Only one sender, multiple Receivers
  - Examples: Radio, Television
- m:1 Multiple senders but only one receiver
  - Examples: Network printer
- m:n Multiple Senders, multiple Receivers
  - Examples: Telephone Conference

m: Number of senders
n: Number of receivers

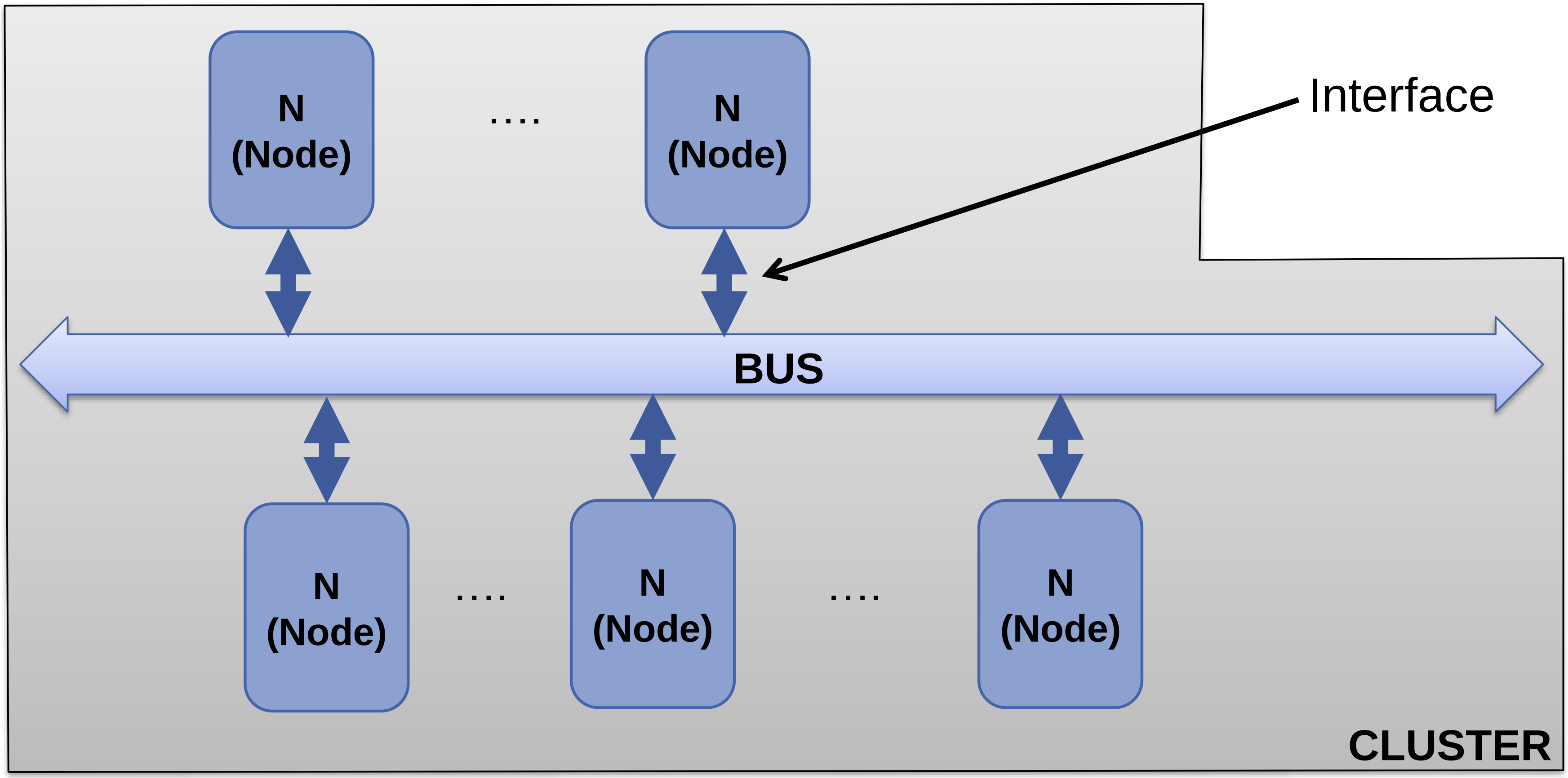
Bus Systems

- Multiple Senders and receivers share the same channel for communication
    - Arbitration and Addressing required



Simplified Overview

- Multiple Transmission Nodes (N) connected over the same wire bundle.



Node

- Definition: Node

  (1): **:** a point at which subsidiary parts originate or center

  (Merriam-Webster dictionary)

  (2): a centering point of component parts.

  (Dictionary.com)

- Pragmatic Definition:
  - A terminal in a communication system

Interface

- Definition: Interface

  a : the place at which independent and often unrelated systems meet and act on or communicate with each other
  b : the means by which interaction or communication is achieved at an interface

  (Merriam-Webster dictionary)

- Pragmatic definition:
  - Defined data transmission connection between exactly two entities
  - Required Specifications:
    - Mechanical coupling
    - Electrical signals and timing
    - logic signals (coding, temporal sequence)
    - A complete definition is not always given!

Bus

- Definition: Bus

  (1) A collection of wires through which data is transmitted from one part of a computer to another. […] When used in reference to personal computers, the term *bus* usually refers to *internal bus*. This is a bus that connects all the internal computer components to the CPU and main memory.

  All buses consist of two parts -- an address bus and a data bus. The data bus transfers actual data whereas the address bus transfers information about where the data should go. (webopedia)

  A set of parallel conductors in a computer system that forms a main transmission path (Merriam-Webster dictionary)

Cluster

- Definition: Cluster

  (1): a number of similar things that occur together: as

  (Merriam-Webster dictionary)

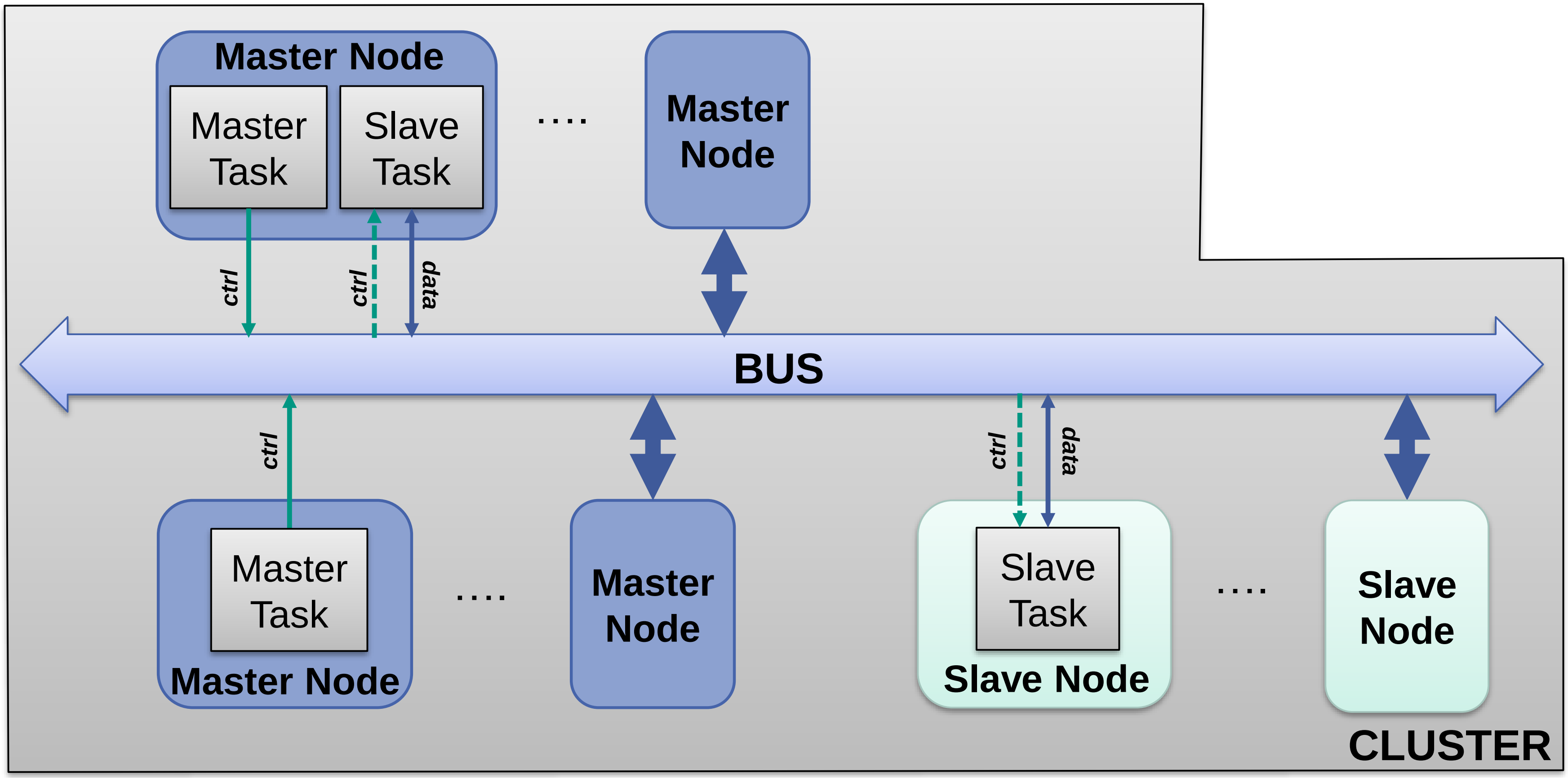
  (2): a group of loosely coupled computers that work together closely

  (webopedia)

- Pragmatic Definition:
  - A bus with a set of nodes connected over it.
  - Within the cluster it is possible to transmit information

General Bus Structure (1)

- Multiple Nodes connected over the same wire bundle
- One wire bundle is allotted to more than a single node



- Address of a node: coded with ld *n* control signals

Bus Nodes

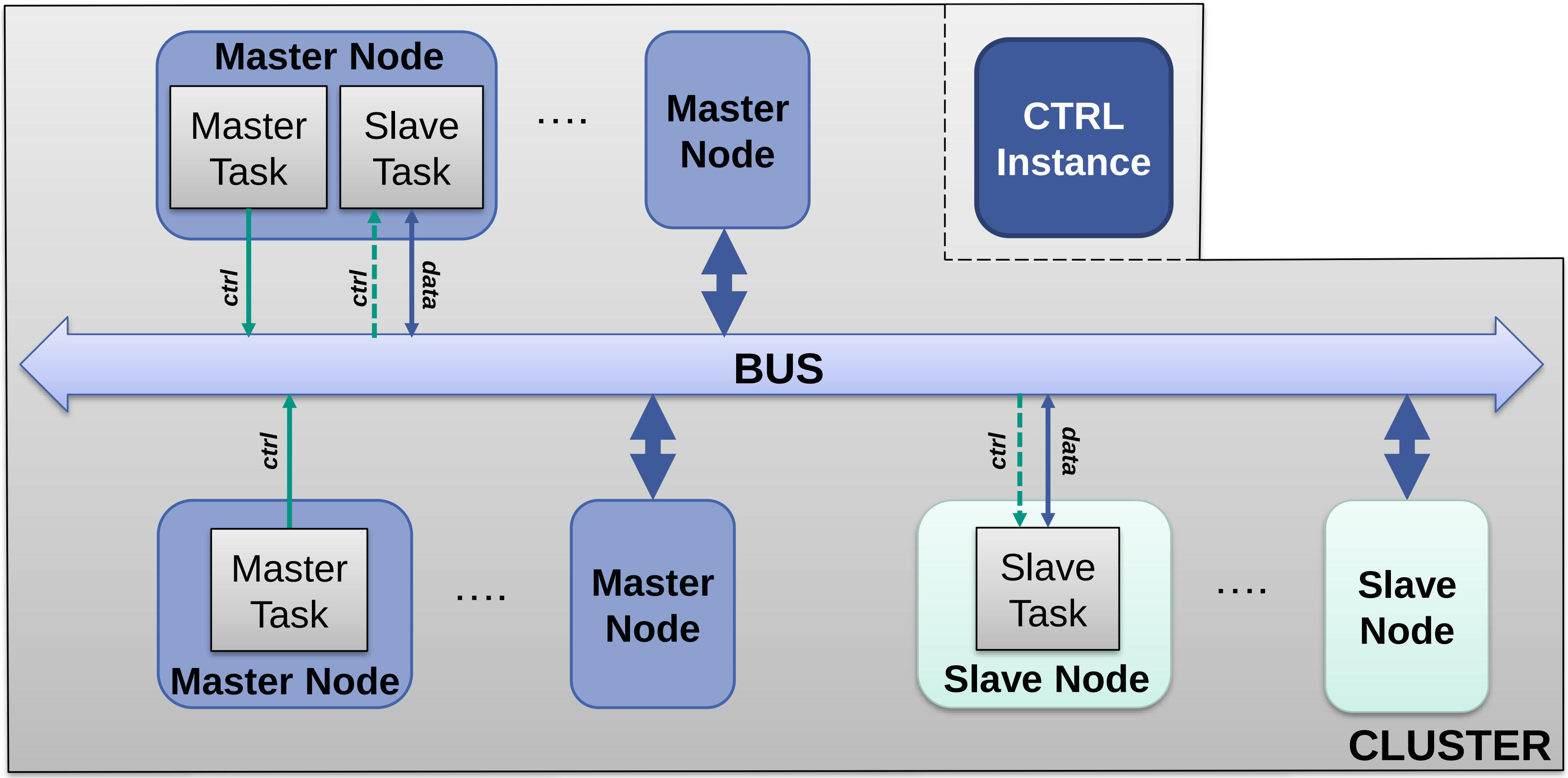
- Master Task
  - active entity
  - controls the bus
  - triggers communication
  - Arbitration

- Slave Task
  - passive entity
  - is controlled by master
  - reads data off the bus
  - puts data on the bus, when triggered by a master task

- Master Task / Slave Task in the same Node (package) e.g. LIN Bus
- Possible Bus Types
  - Single Master / Multi Slave
  - Multi Master / Multi Slave

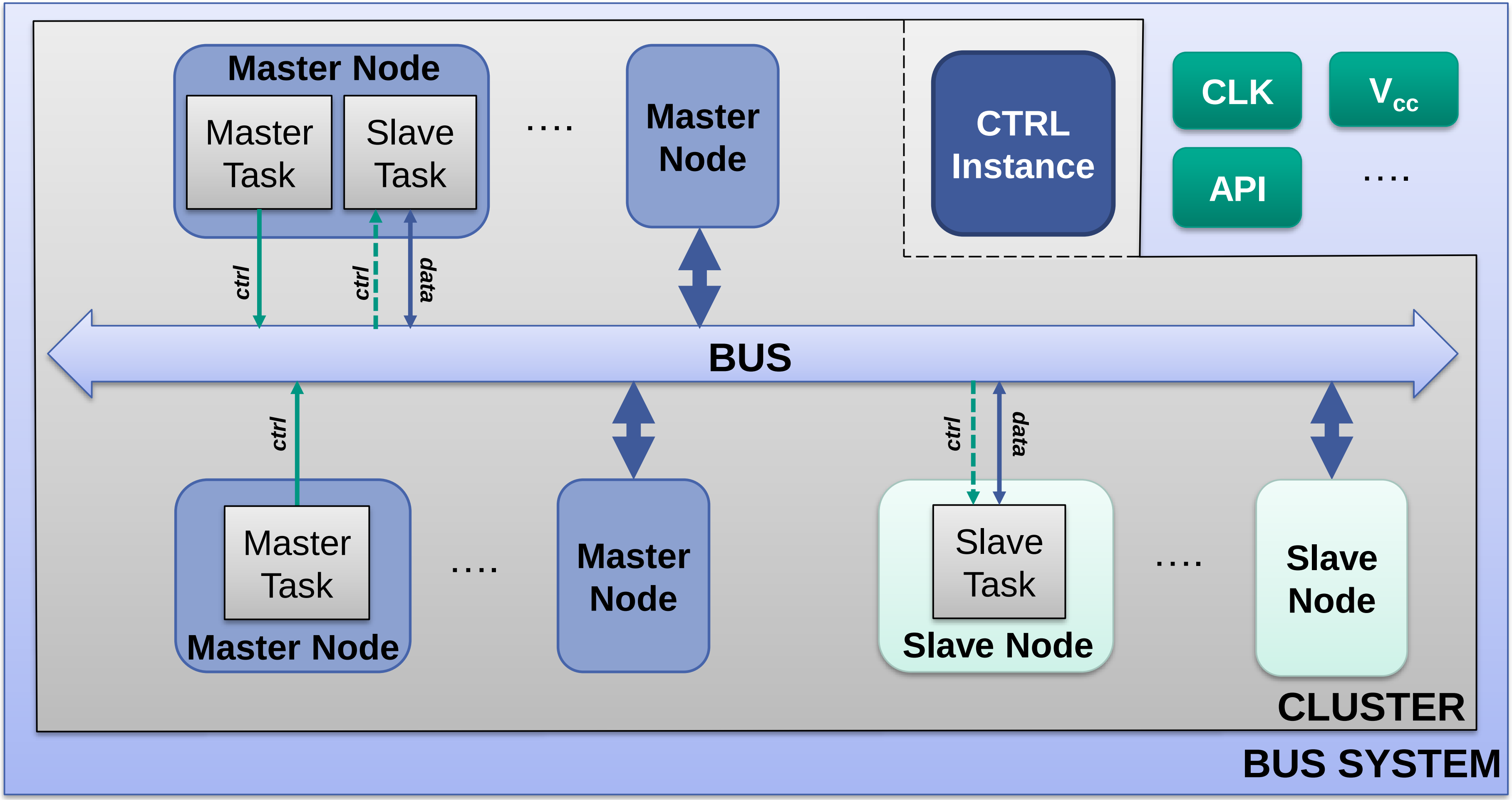
General Bus Structure (2)

- Multi-Master Busses require control instances (e.g. Arbiter) to regulate bus access
- Can be a node within the bus or an external controller

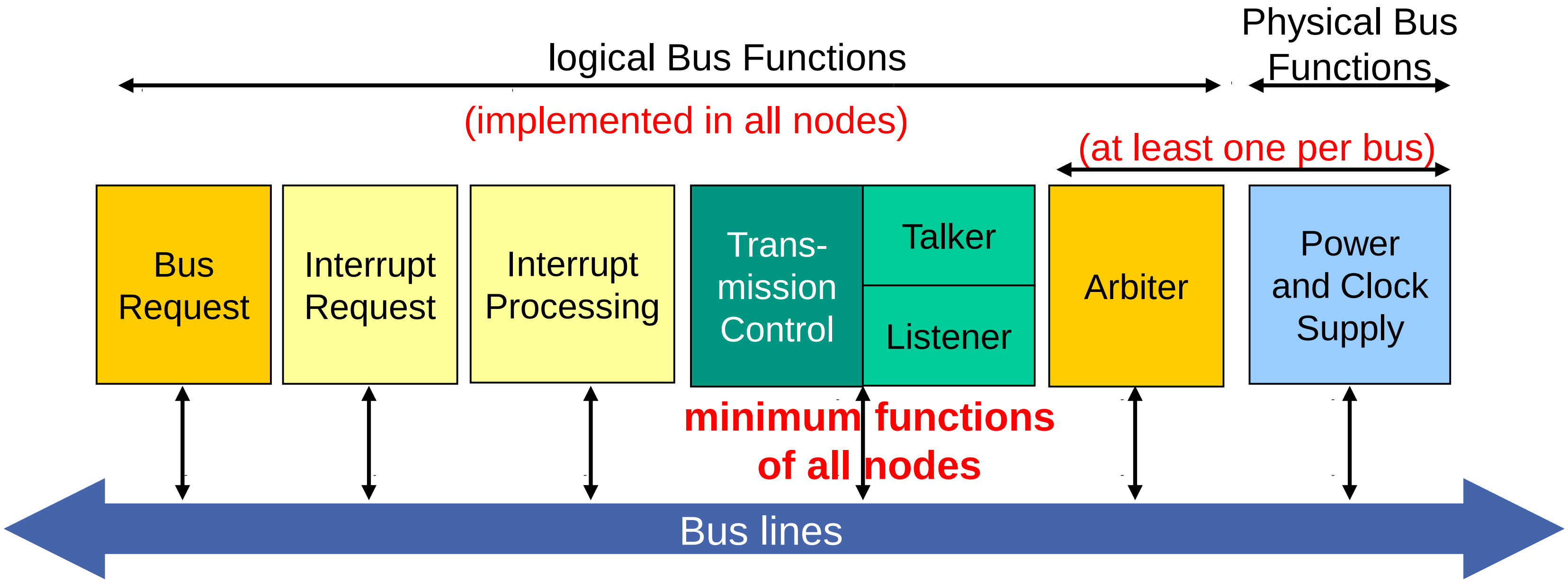


General Bus Structure (3)

- Additional components (clock, power supply, Application interfaces, … form the **bus system**.

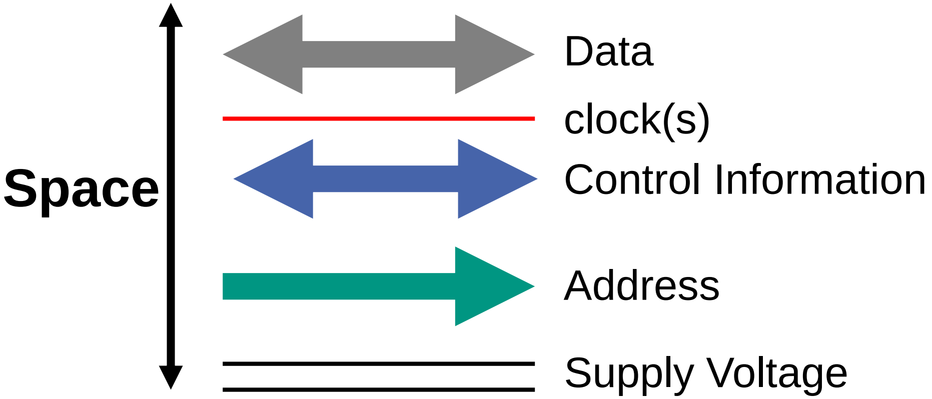


Functional Setup of a Node

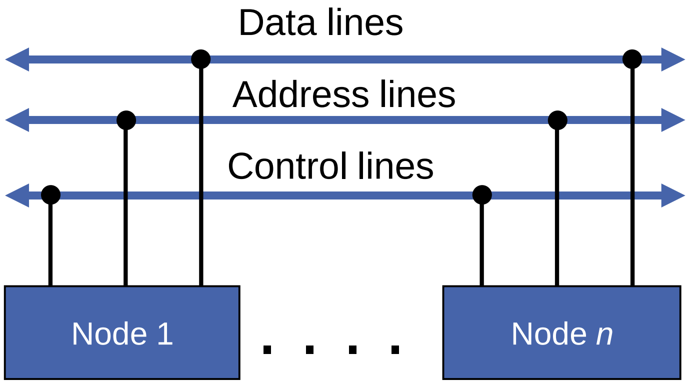


Bus Request:	Node wants to become bus master
Bus Arbitration:	Processing of Bus Request(s)
Interrupt Request:	Forwarding of a node’s interrupt
Interrupt Processing:	Reception and processing of an interrupt
Transmission Control:	Control of the actual data transmission either as master or slave. A node can send data (Talker) or receive data (Listener)

Parallel Bus



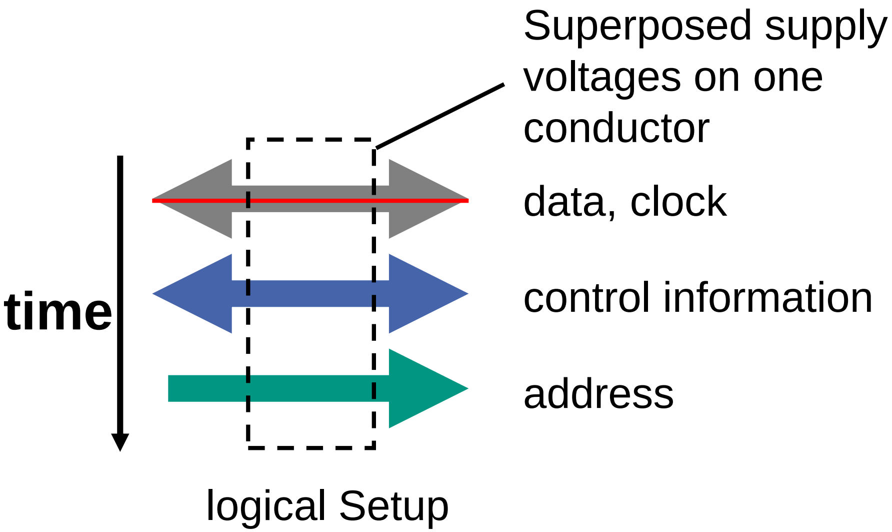
logical Setup

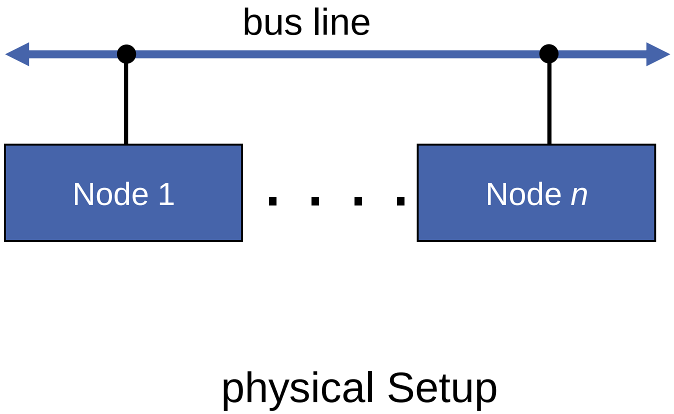


physical Setup

- Data Bus: Actual Data Transmission
- Address Bus: Selection of specific nodes and addresses → sometimes data and addresses are transmitted over them same line at different points in time
- Control Bus: Bus requests, Arbitration, Interrupts
- Supply Bus: Power Supply and Clock Lines

Serial Bus





- Only a single line that is set up in a bus structure.
- Functions that are implemented via dedicated wires in parallel bus systems, are implemented in serial bus systems via (software-) protocols

Comparison Serial vs. Parallel Communication

Parallel Communication

- Pros:
  - High throughput possible
- Cons:
  - Expensive
  - Skew of individual signal lines
  - Protocol changes become difficult because of predefined signal lines

Serial Communication

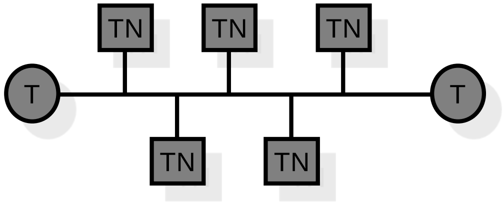
- Pros:
  - Cheap
  - High data rates with low effort
  - Flexible in terms of changes
- Cons:
  - less bits per clk can be transmitted

Topology

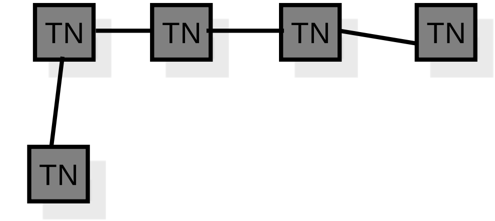
> *A (network)* ***topology*** *describes the arrangement of nodes within a communication systems. It defines how the computers, or nodes, within the network are arranged and connected to each other.*

- Characteristics of divers topologies
  - Normal operation: resources and cost, throughput, range, maximum number of nodes, expandability
  - Exceptional operation: susceptibility, start-up operation, restart, node connection during runtime, Security (malicious misuse, eavesdropping,…)

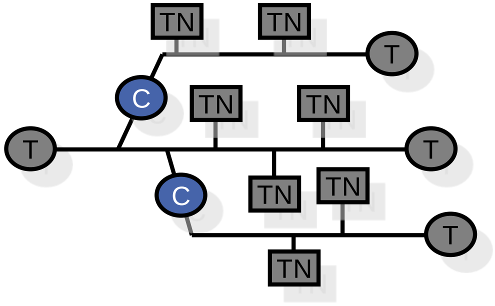
Common types of Topologies - Overview



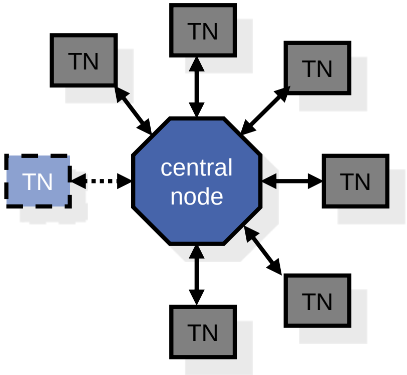
Bus



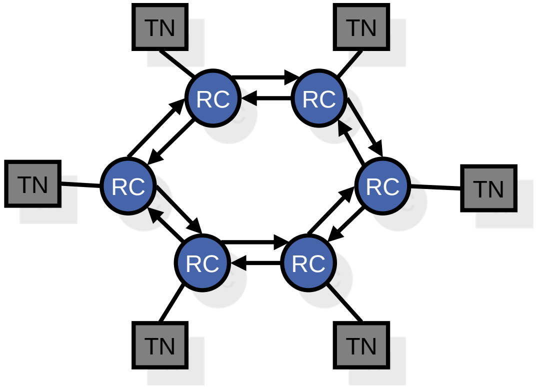
Line



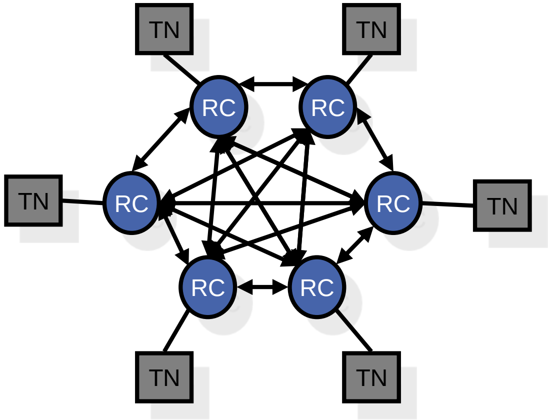
Tree



Star



Ring



Fully connected Mesh

TN Node

T Termination

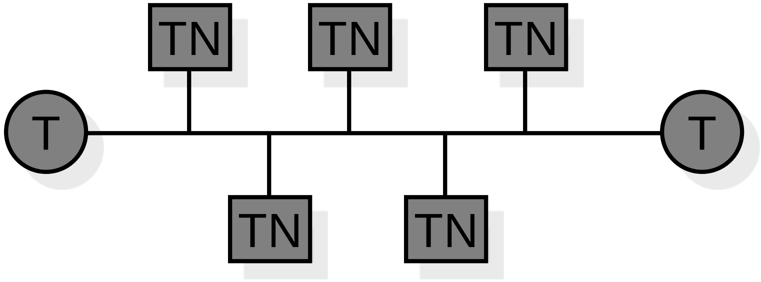
C Coupler

RC Ring Coupler

Bus Topology

Each node is connected to a central bus that runs along the entire network. All information transmitted across the bus can be received by any system in the network.

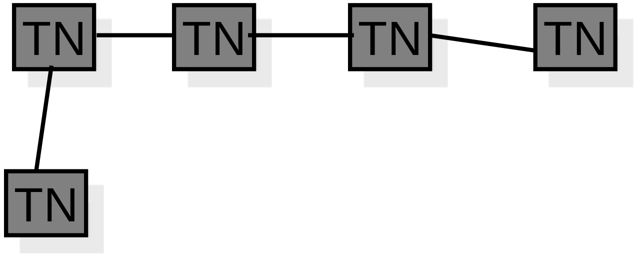
- Linear Bus
- One shared line
- Easy connection of nodes with tap lines
- Only one master a time
- Danger of collisions, complex management



Line Topology

Nodes are arranged in a line, where most nodes are connected to two other nodes. However, the first and last node are not connected like they are in a ring.

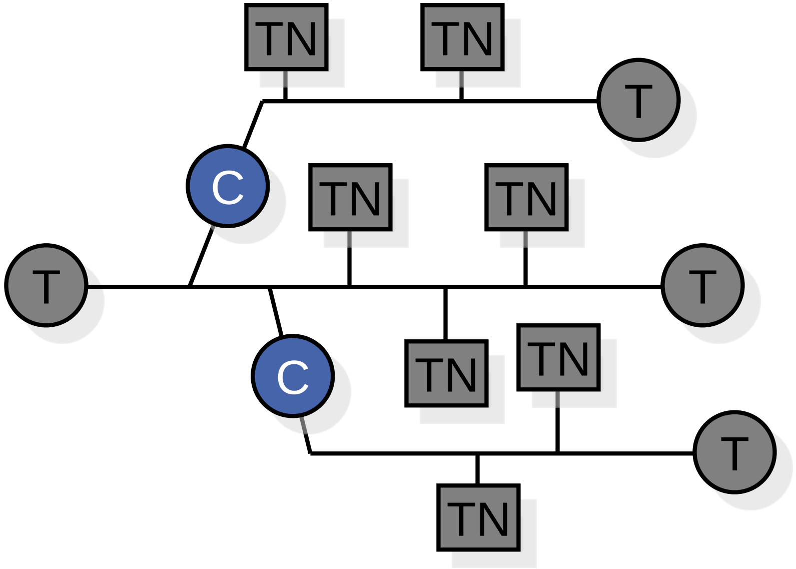
- Point-to-Point Connection
- Easy management, easily extensible
- Example: FireWire in some cases



Tree Topology

One "root" node connects to other nodes, which in turn connect to other nodes, forming a tree structure. Information from the root node may have to pass through other nodes to reach the end nodes.

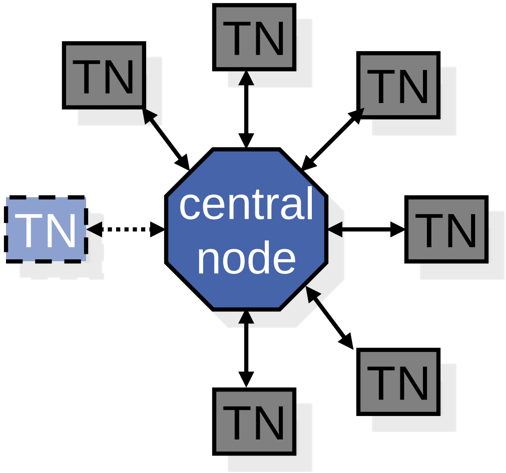
- Tree Structure
- Suitable couplers connect individual linear bus segments
- Long distances possible
- Increased data throughput with local data exchange and intelligent couplers



Star Topology

One central node is connected to each of the other nodes on a network. Similar to a hub connected to the spokes in a wheel.

- Central Node, no communication when inactive
- Point-to-Point Connection
- Easy management, easily extensible
- Example: Telephone switchboard
  Ethernet-Switch

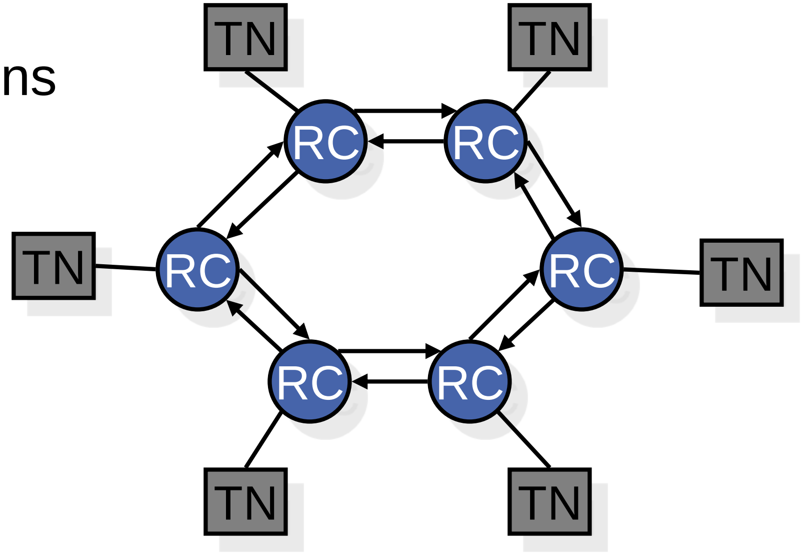


Ring Topology

Each node is connected to exactly two other nodes, forming a ring. Can be visualized as a circular configuration. Requires at least three nodes.

- Ring Structure
  - Cycle of Point-to-Point connections
  - Access over ring coupler
  - susceptible to failure

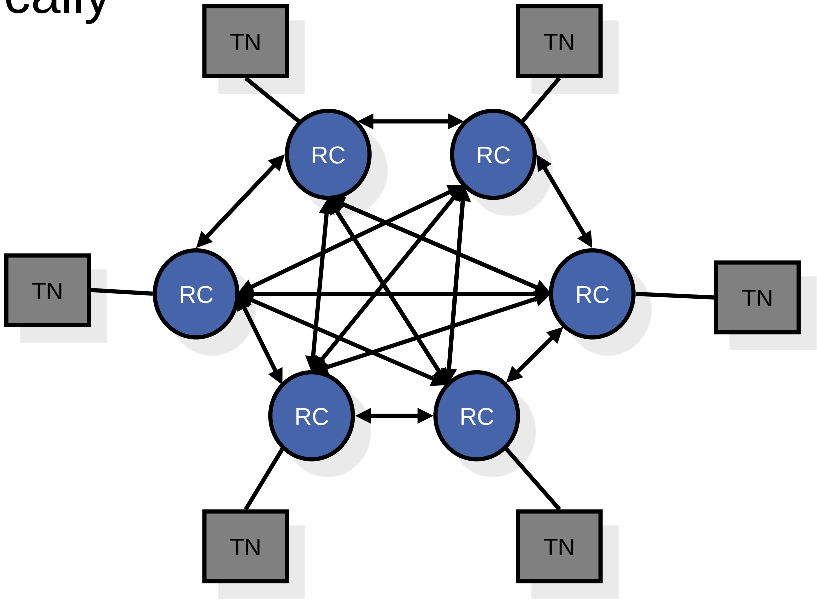
- Double Ring
  - Redundancy within the ring
  - Shutdown of damaged nodes possible



Fully Connected Mesh Topology

Each node is connected to all other nodes.

- Number of connections grow quadratically
  - c = n(n-1)/2 where
    - n is number of nodes
    - c is a uni-directional connection
- Highly reliable due to redundant links between nodes.
- Mostly used in military applications.

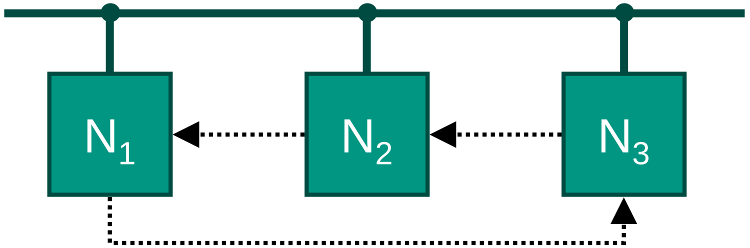


Logic Structure vs. Physical Structure

- Topology is the spatial relation of (**physical**) nodes,
- Is determinable by examining the sequence of access to the (**logic**) nodes.

- Difference between logic and physical structure are possible
  - Physical bus with a logic ring e.g. Profibus, TokenBus
  - Physical star with logic bus e.g. Ethernet

- Example: TokenBus
  - Physical structure: Bus
  - Logical structure: Ring

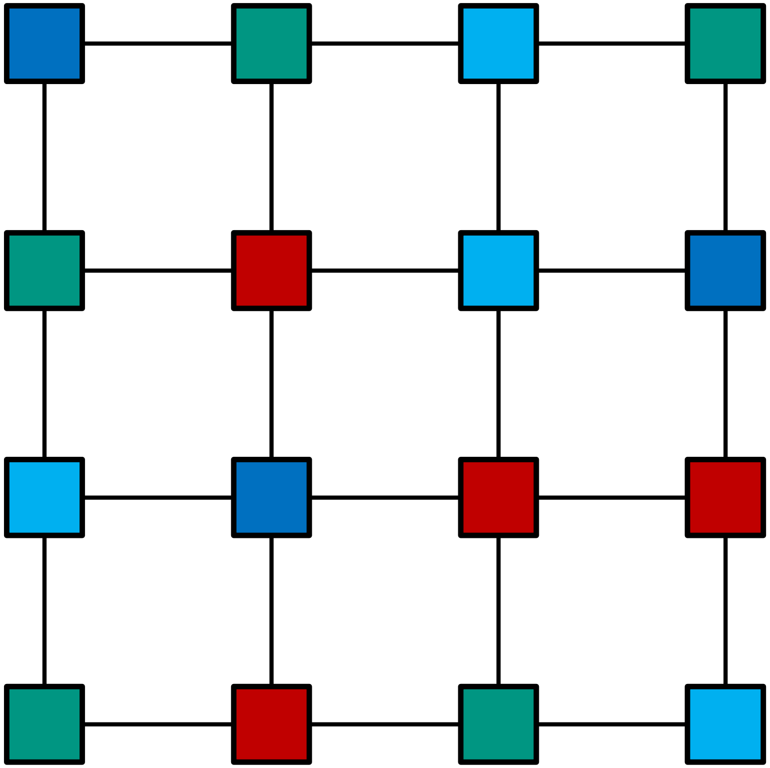


Physical Bus

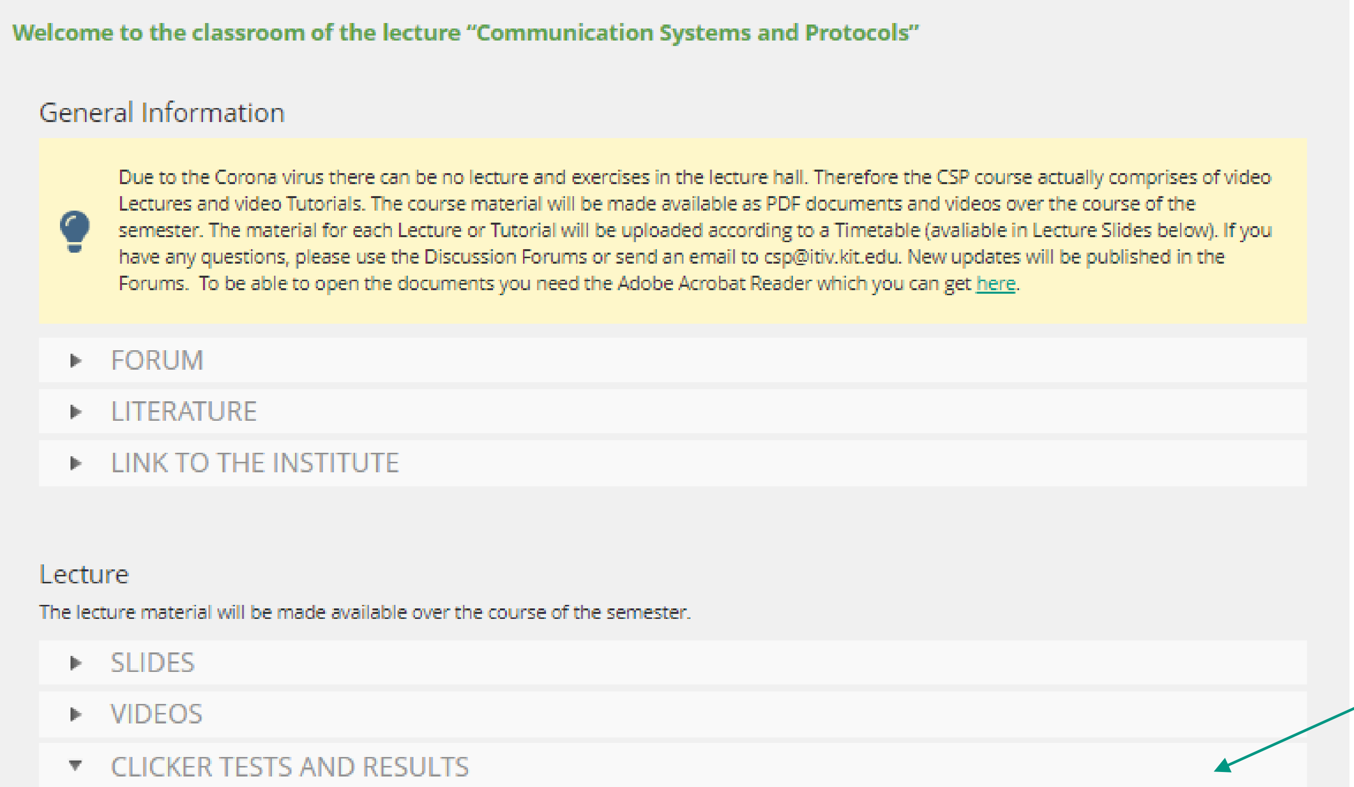
Logical ring

Network

- Special form of a topology
- More connections per node as compared to standard topologies
- Details will be presented later in this lecture



Participate in Clicker Test 5a



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

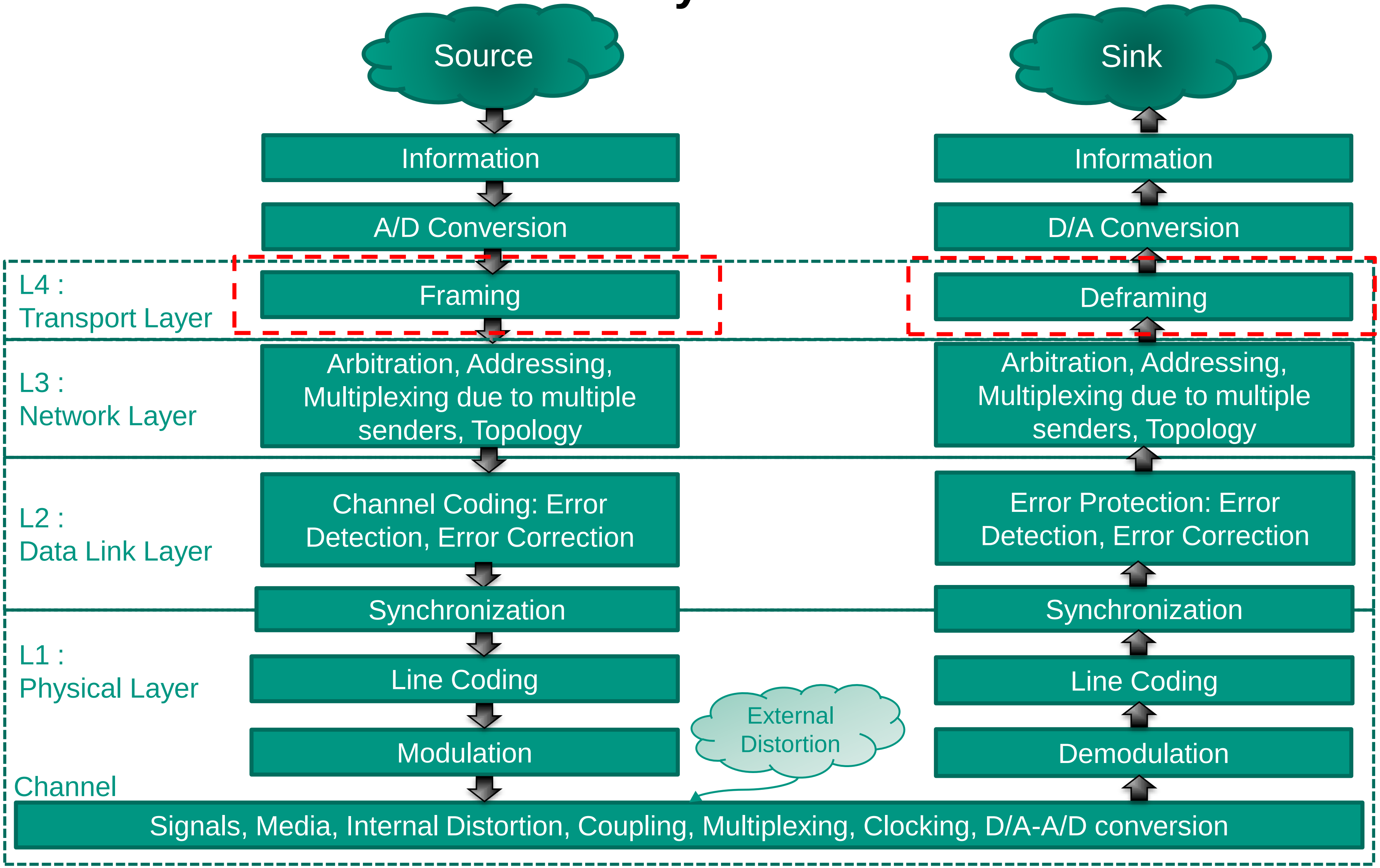
Communication Systems and Protocols

Protocols

Contents

- Protocol specification
- Date Frames

Technical Communication System



Source

Information

A/D Conversion

L4 :
Transport Layer

Framing

L3 :
Network Layer

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

L2 :
Data Link Layer

Channel Coding: Error Detection, Error Correction

Synchronization

L1 :
Physical Layer

Line Coding

Modulation

External Distortion

Channel

Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Sink

Information

D/A Conversion

Deframing

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

Error Protection: Error Detection, Error Correction

Synchronization

Line Coding

Demodulation

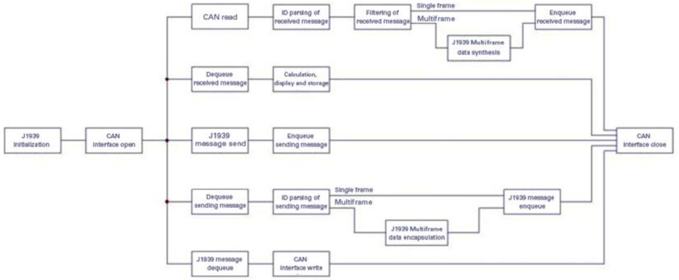
Specification of a Bus System

- Technical parameters (on Technology Level)
  - Signals
  - Signal Lines
  - Levels
  - Interconnects (i.e. Plugtype, Footprints)

Source: http://www.etas.com

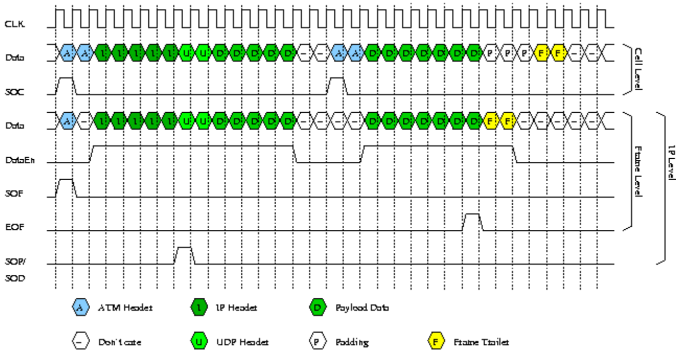
Source: http://www. leib-engineering.de

- Protocol
  - behavioral description
  - what happens, when does it happen
  - timely behavior



Source: http://sine.ni.com

- Dataframe
  - Interpretation of signals (lines, bits, positions of bits)
  - Representation of Information



→ All of the above is forming the specification

Protocols

Definition:

*A **protocol** is a set of rules for communication amongst a set of communication partners.*

- Requirements:
  - Completeness
  - Uniqueness / Unambiguousness
  - Transparency
  - Documentable
  - Adaptability

- A defined and eventually standardized protocol is the basis for save and reliable (and sometimes legally binding) functionality of open systems.

Protocol Specification

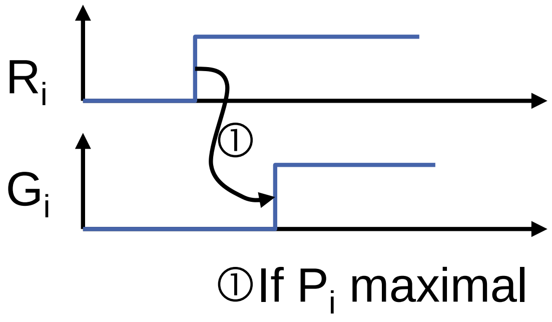
- Verbal representation
  „.. TN$_i$ would like to be master and asserts signal R$_i$ with 1. Subsequently the arbiter grants access to the bus G$_i$ if priority of i is currently the highest...“

- Problems:
  - Representation is not unique
  - unclear, sometimes even confusing
  - Interdependencies hard to grasp

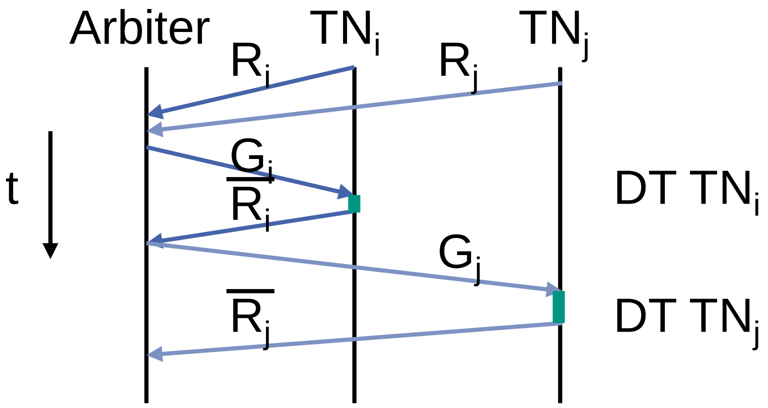
- → Representation in a formulized procedure is useful

Protocol Specification

- Pulse diagram
  - Representation of temporal characteristics of signals
  - Cause and effects are marked with additional arrows



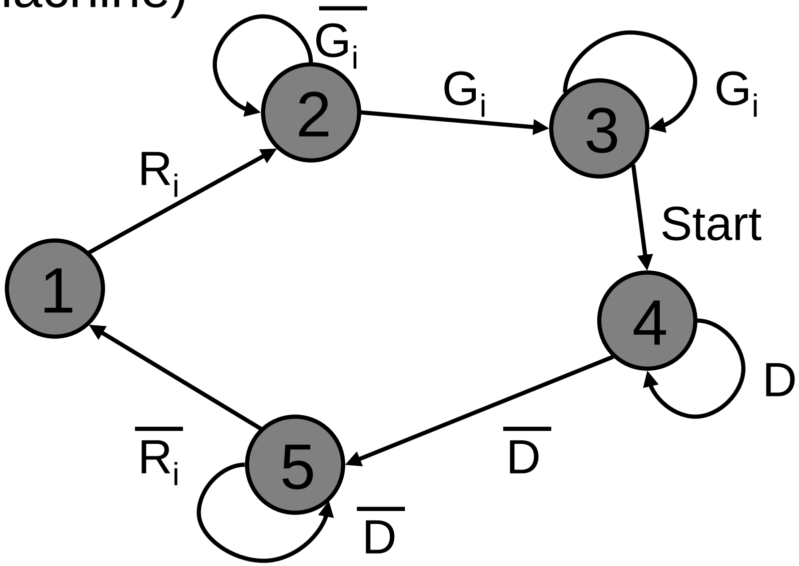
- Sequence diagram
  - Communication partners are marked with vertical lines
  - Exchange of messages are marked with descending lines



TN_i: Node i
DT: Data Transmission

Protocol Specification

- State Transition Diagram (finite state machine)
  - Graphical representation
  - Easy to read and understand



- Syntax notation in BNF-Notation (Backus-Naur-Form)

  Example: ARINC Protokoll

```
<language> ::= <prologue><frame_table><epilogue>
<prologue> ::= <gap_spec><delta_spec><ver_spec><other_spec>
<gap_spec> ::= GAP <digit><eol>
...
<binary> ::= <bit_value>[<binary>]
<bit_value> ::= 0|1
```

Dataframe - General Structure

control information <header>	data field <body>	data validation <trailer>

- Synchronization
- Addresses
- Instructions
- Arbitration

- Fixed or arbitrary length
- Bits, Bytes, Symbols

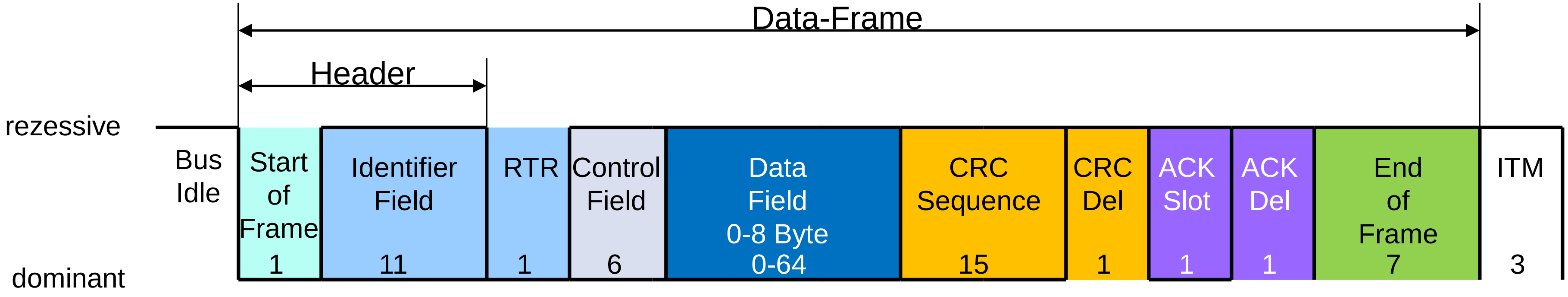
- Correction information
- Error detection
- End notification

Header of a dataframe

- Header is send as first part of a dataframe it contains
  - Information for synchronization (see session 4)
  - Address of target node (see session 6)
  - Information for Arbitration (see sessions 5 and 6)
  - Additional control information (e.g. priorities, change between commands and data, type of packet, etc.)

- Information fields are usually distinguished by their position (length) in the header

- Example: CAN Dataframe



Dataframe – Length of information field

control information <header>	data field <body>	data validation <trailer>

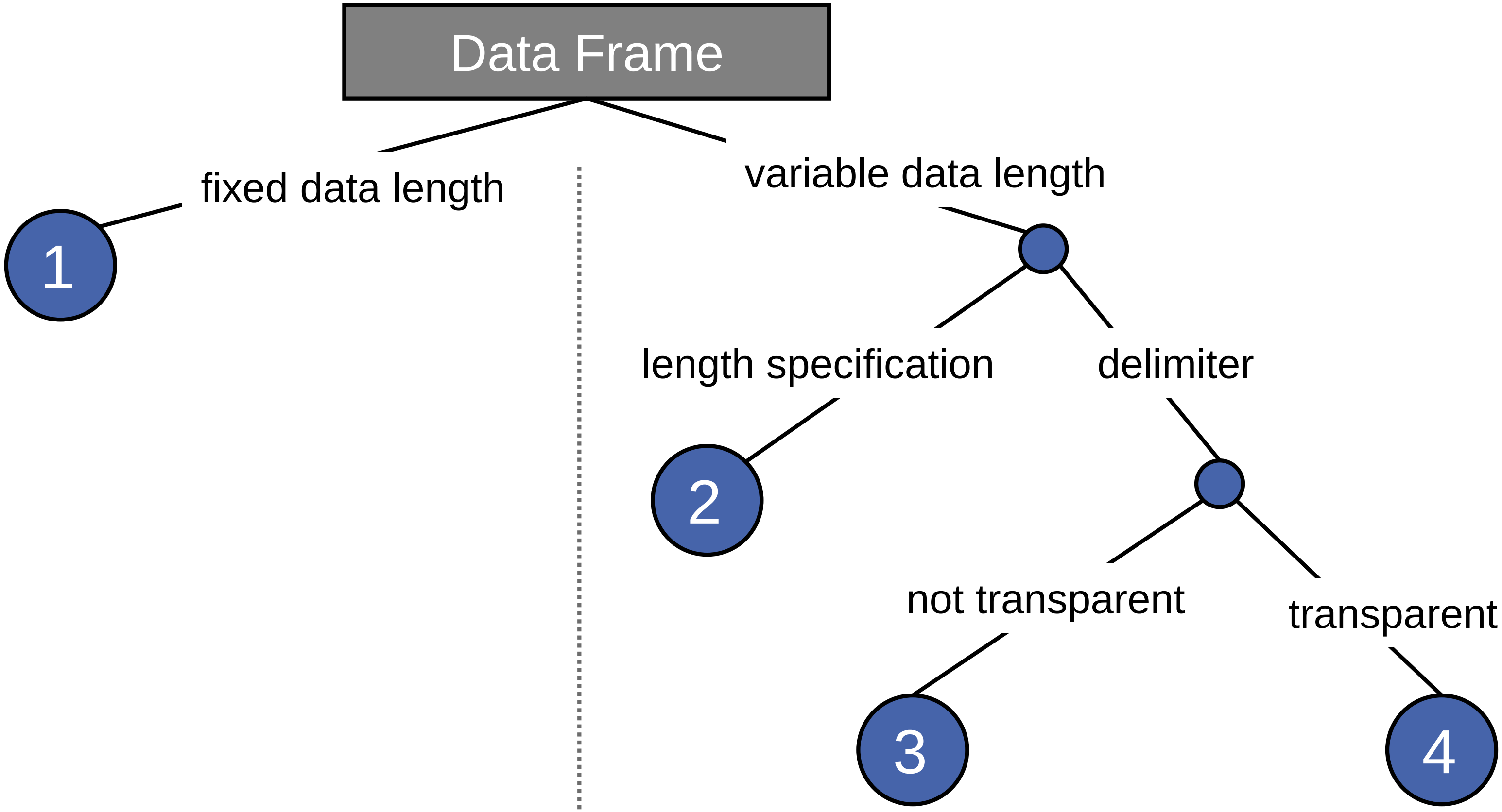
- Synchronization
- Addresses
- Instructions
- Arbitration

- Fixed or arbitrary length
- Bits, Bytes, Symbols

- Correction information
- Error detection
- End notification

- The meaning of a bit in a dataframe is usually determined by ist position
- What happens if data of variable length is to be transmitted?

Determination of data field length



Data Frame

fixed data length

1

variable data length

length specification

2

delimiter

not transparent

3

transparent

4

+ : Explicit field boundaries

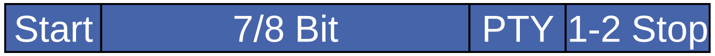
– : Eventually „empties“

+ : Efficient utilization

– : field boundaries

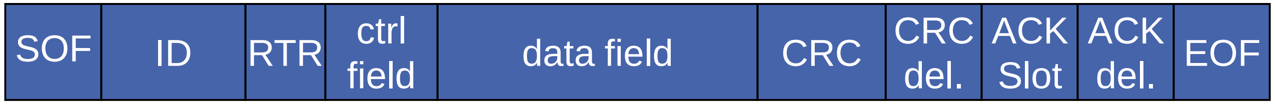
Examples for Dataframes

1. Fixed data length
   - Serial interface, RS-232



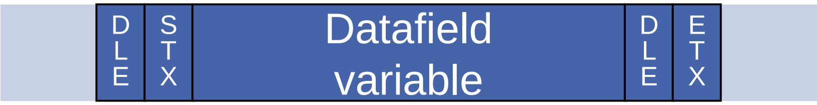
Start	7/8 Bit	PTY	1-2 Stop

2. Field length in header
   - CAN-Bus Data packet: data field length 0-8 Bytes
   - Length denoted in control field



SOF	ID	RTR	ctrl field	data field	CRC	CRC del.	ACK Slot	ACK del.	EOF

3. Reserved symbols, data field without such symbols
   - unusual
4. Arbitrary data, symbol stuffing necessary
   - Predefined symbols for start and end of transparency, e.g. DLE, STX, ETX (as used in ASCII)



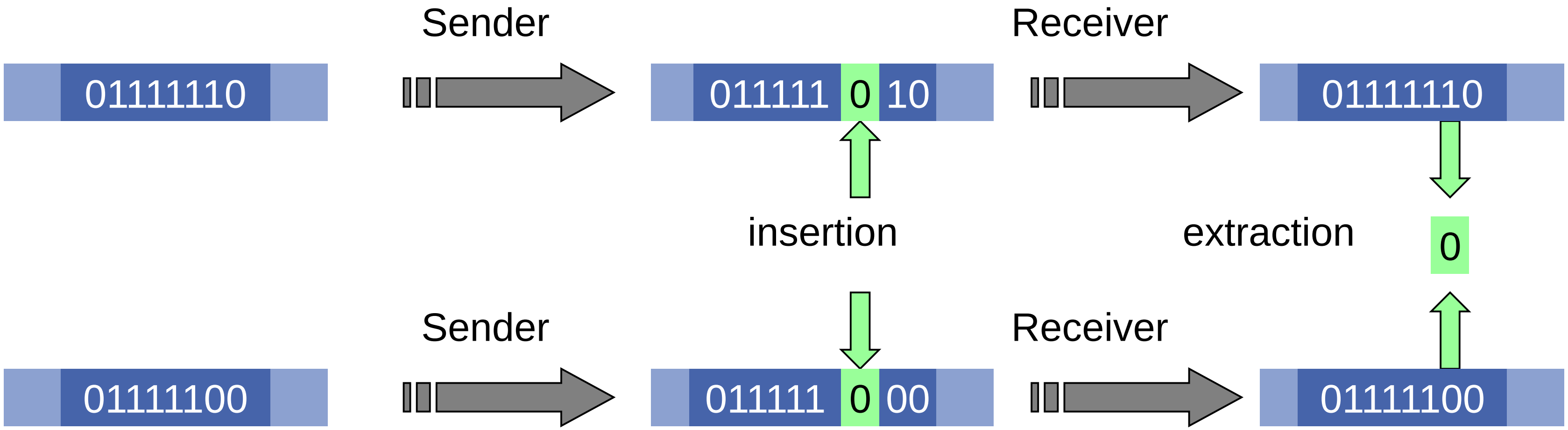
DLE	STX	Datafield variable	DLE	ETX

Bit Stuffing (Transparency)

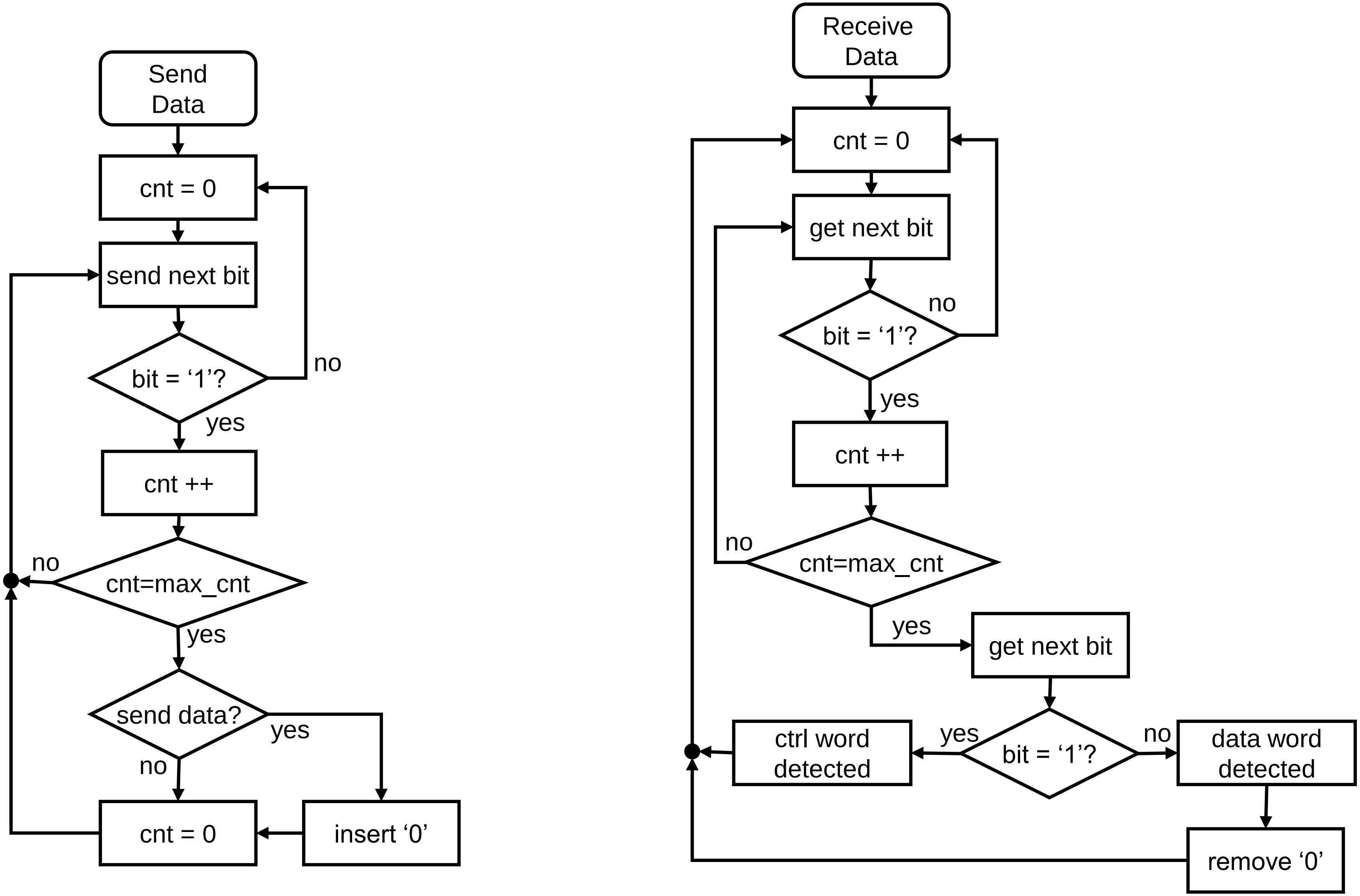
- Bit Stuffing
  - A sequence of *n* ‚0‘s or ‚1‘s is defined as control symbol for Data Start/End. This must not appear within the data stream.
  - Insertion of a ‚1‘ after *n-1* ‚0‘s and vice versa

Example:

- after five ‚1‘s in a data word a ‚0‘ is inserted. The receiver will remove every ‚0‘ after a sequence of five ‚1‘s.



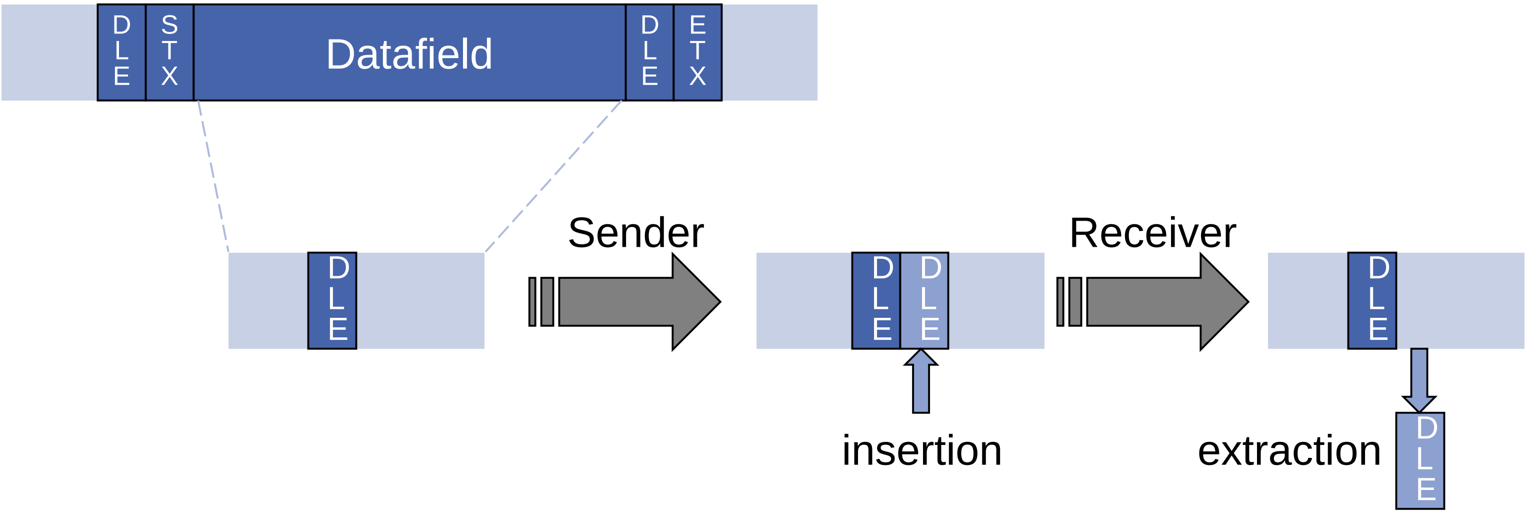
Bit Stuffing – Algorithm (Example)



Symbol Stuffing (Transparency)

- Symbol Stuffing
  - When using frame formats with variable data field lengths, a symbol is necessary that marks the start and end of a data field:
    Data Link Escape (DLE)
  - This symbol can also be a regular symbol to be transmitted

    → Masking necessary
  - If DLE is part of the data field, the symbol is doubled and receiver removes the second DLE symbol.

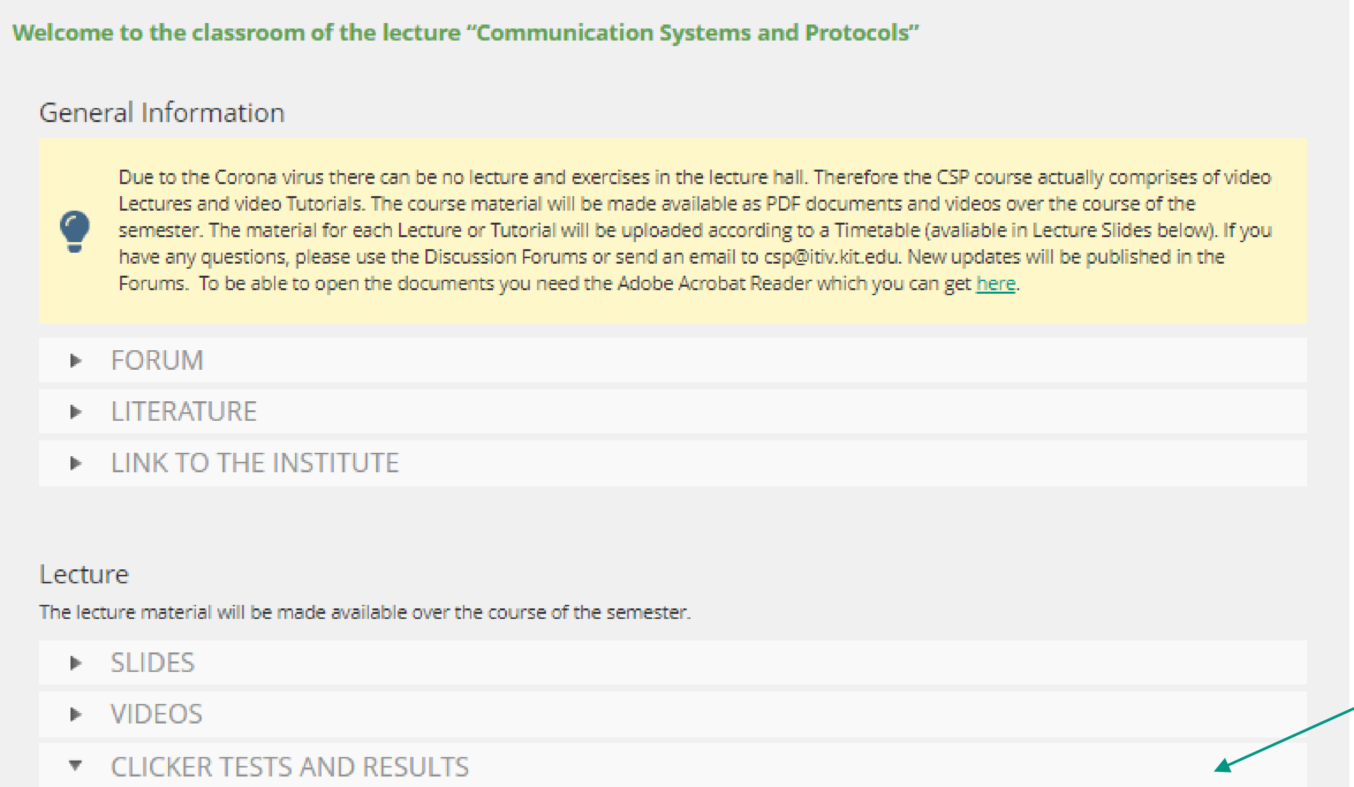


Dataframe – Data validation

control information <header>	data field <body>	data validation <trailer>
- Synchronization - Addresses - Instructions - Arbitration	- Fixed or arbitrary length - Bits, Bytes, Symbols	- Correction information - Error detection - End notification

- Data might get corrupted during transmission
- How do we ensure that correct data has been received?

Participate in Clicker Test 5b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

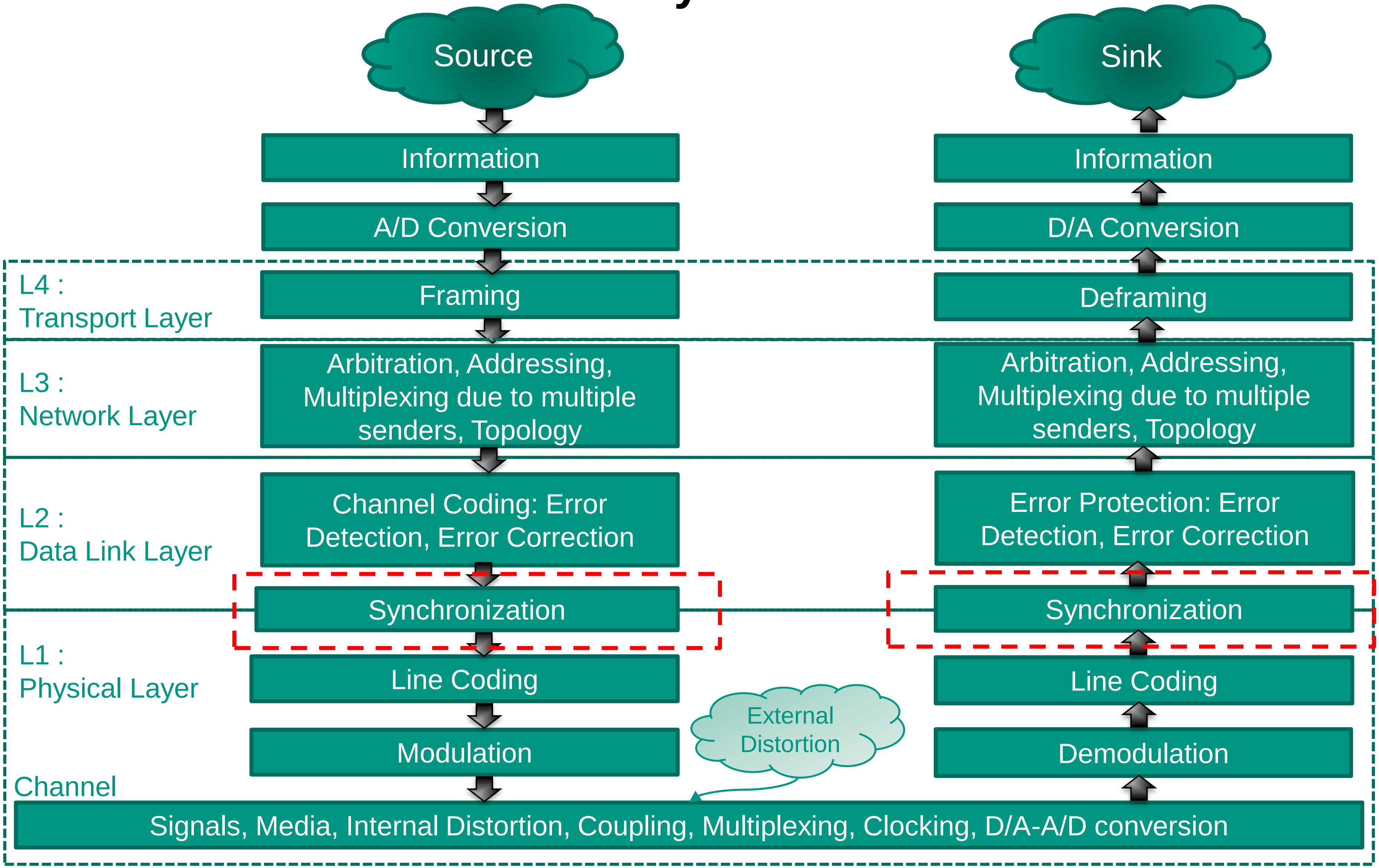
Synchronisation

Contents

- Synchronous transmission vs. Asynchronous transmission

- Synchronisation schemes
  - Start-Stop mode
  - Flow control

Technical Communication System



Source

Information

A/D Conversion

L4 : Transport Layer

Framing

L3 : Network Layer

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

L2 : Data Link Layer

Channel Coding: Error Detection, Error Correction

Synchronization

L1 : Physical Layer

Line Coding

Modulation

Channel

Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

External Distortion

Sink

Information

D/A Conversion

Deframing

Arbitration, Addressing, Multiplexing due to multiple senders, Topology

Error Protection: Error Detection, Error Correction

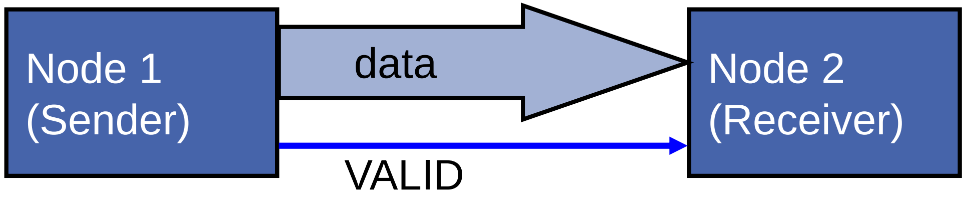
Synchronization

Line Coding

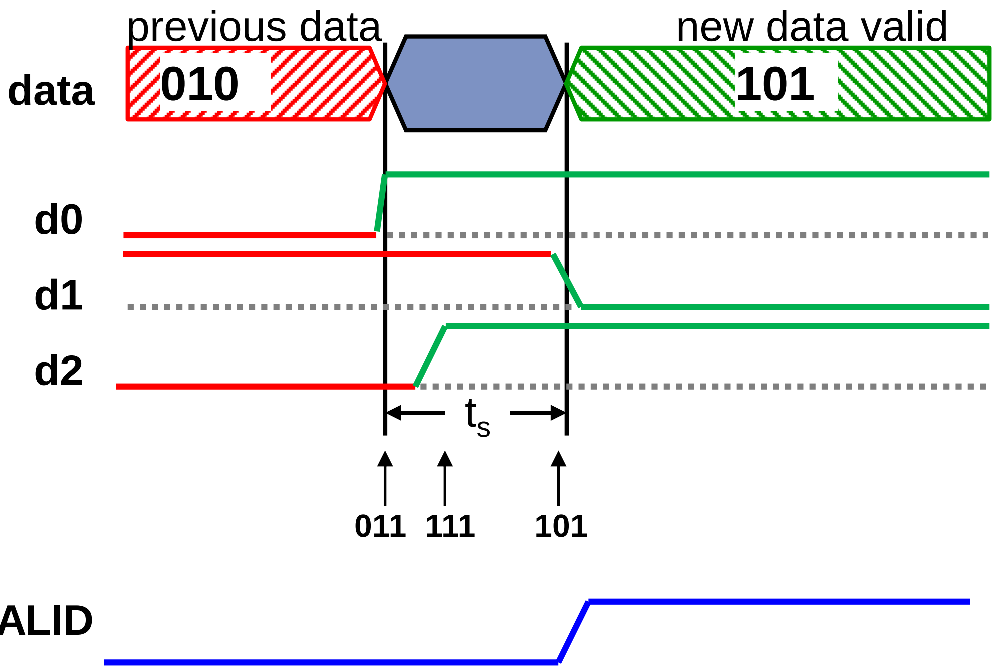
Demodulation

Data-Skew: Need for synchronisation

- Sender outputs new information to data lines
- Due to varying signal delays on different (parallel) signal paths the data on the bus may be invalid for a period of time t_s



- Example:
  Signal change 010 → 101



- Sender marks valid new data with a VALID signal

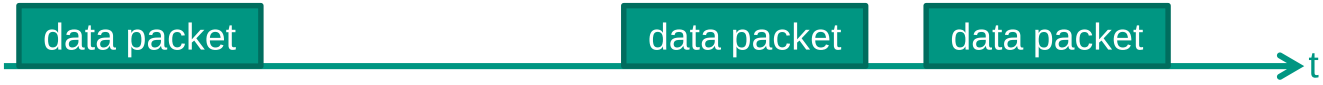
Different types of Synchronisation

- Low-level Synchronization (clocking)
  - Communication partners have to be able to separate individual bits
  - Common time base using
    - Clock line
    - Suitable line code
    - synchronized local clocks
    - …

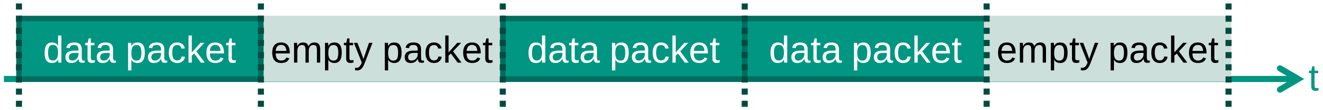
- High-Level Synchronization (e.g. flow control)
  - Logical Synchronization of communication process
    - When does a data frame start/end?
    - Is the receiver ready for reception?
    - …

Synchronous vs. Asynchronous Transmission

- Asynchronous:
  - Transmission can occur at any time
  - Mark begin and end of a transmission
  - There can be periods where „nothing“ is being transmitted



- Synchronous:
  - Transmission at dedicated time points
  - Synchronization even without payload data being send
  - Even if there is no data transmission required, empty packets will be send

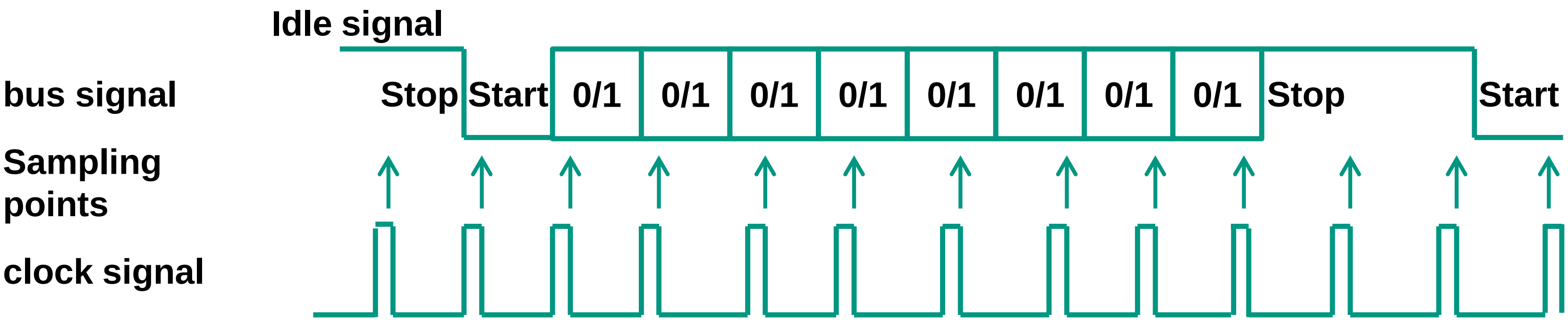


Synchronisation Schemes

	Synchronous Transmission	Asynchronous Transmission
Parallel Transmission	Shared clock line *(see Session 11)*	Flow-Control
Serial Transmission	Suitable line code or scrambler (shared clock line) *(see Session 10)*	Start-Stop-mode

Start-Stop mode

- When no transmission occurs, bus is in idle state → no signal changes occur on the bus
- Begin of a transmission is signaled using a well defined edge on the bus signal → start bit
- This edge is used to synchronize sender and receiver clock
- Every bit has a well defined step size and thus can be separated using the internal clocks → baud rate
- At the end, it has to be ensured that the bus goes to idle level in order to be able to detect a new transmission → stop bit

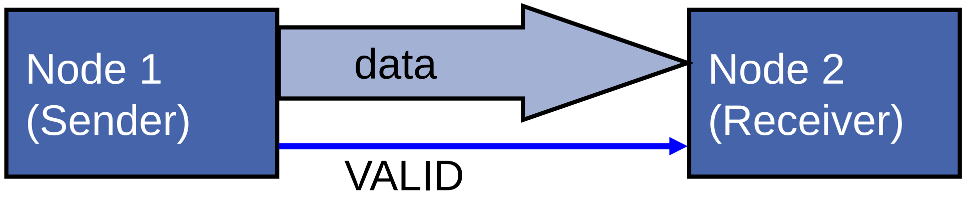


Start-Stop mode: Format definition

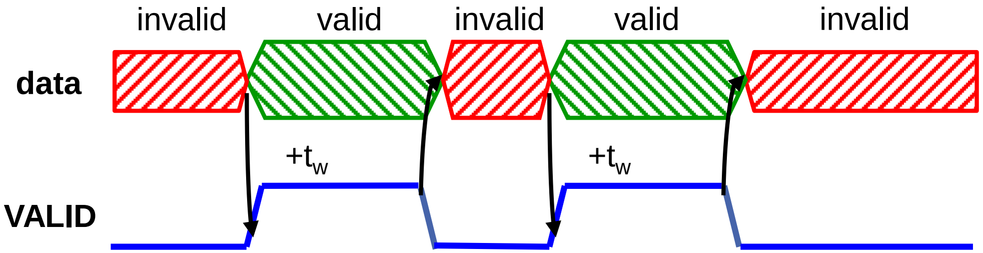
- Format definition for serial transmission using start-stop mode
  - **X-Y-Z**
- With:
  - X: number of data bits (normaly 5-8)
  - Y: parity check (none, even, odd)
  - Z: number of stop bits (1, 2)
- Notes:
  - There is always on start bit (denotes begin of transmission)
  - Parity checks will be explained in session 18
  - The number of stop bits in the definition is the **minimum** number. There can be more „stop bits“ if no transmission is going on

Open-loop Flow Control

If data is transmitted in multiple subsequent cycles, it has to be regulated when data of one cycle is processed and when a new cycle may be started.



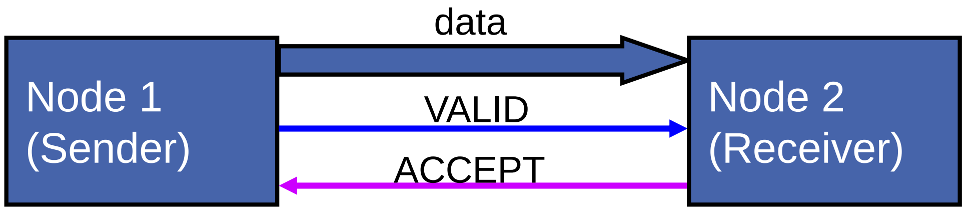
- One always waits for a fixed period of time t_w



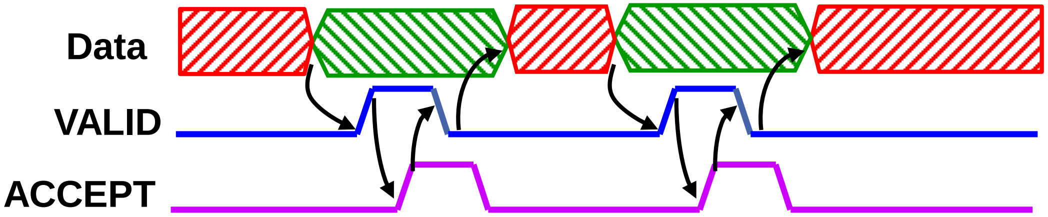
- Disadvantage: Waiting time t_w is dependent on the slowest node on the bus and is always invariant in length (synchronous mode)

Level-triggered Closed-loop Flow Control

Compared to open-loop flow control, closed-loop flow control uses additional signals from receiver to signal the sender its state.



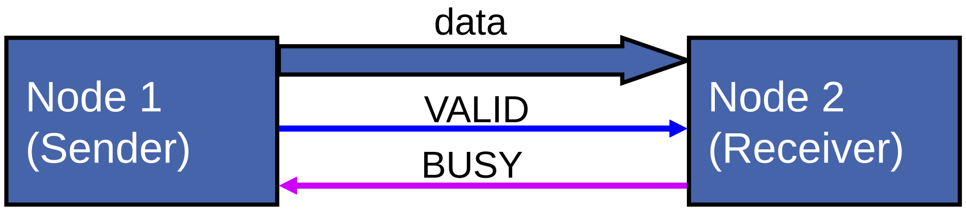
- Receiver asserts ACCEPT signal if data is not needed any longer.



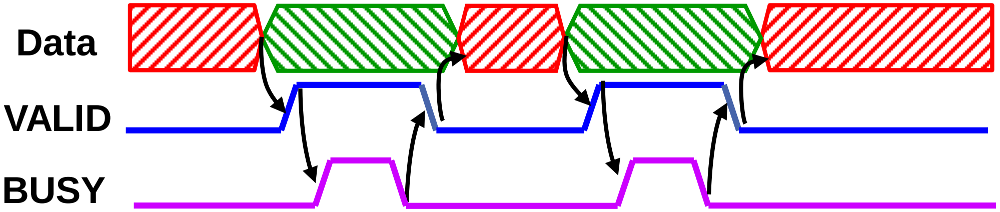
- Disadvantage: Problems with node synchronization if ACCEPT signal is asserted for too long (sender will remove subsequent data from the bus before receiver has read it)

Level-triggered Closed-loop Flow Control II

If data is transmitted in multiple subsequent cycles, it has to be regulated when data of one cycle is processed and when a new cycle may be started.



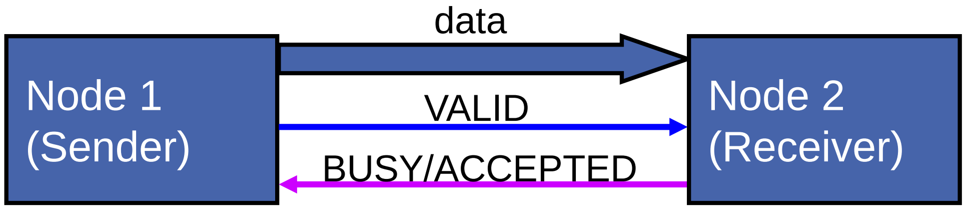
- Receiver asserts BUSY-Signal as long as data has to be available.

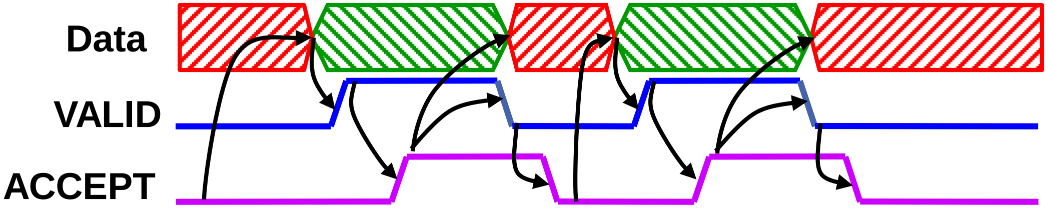


- Disadvantage: Problems with node synchronization if BUSY signal is asserted too late (data will be removed from the bus too soon)

Edge-triggered Closed-loop Flow Control

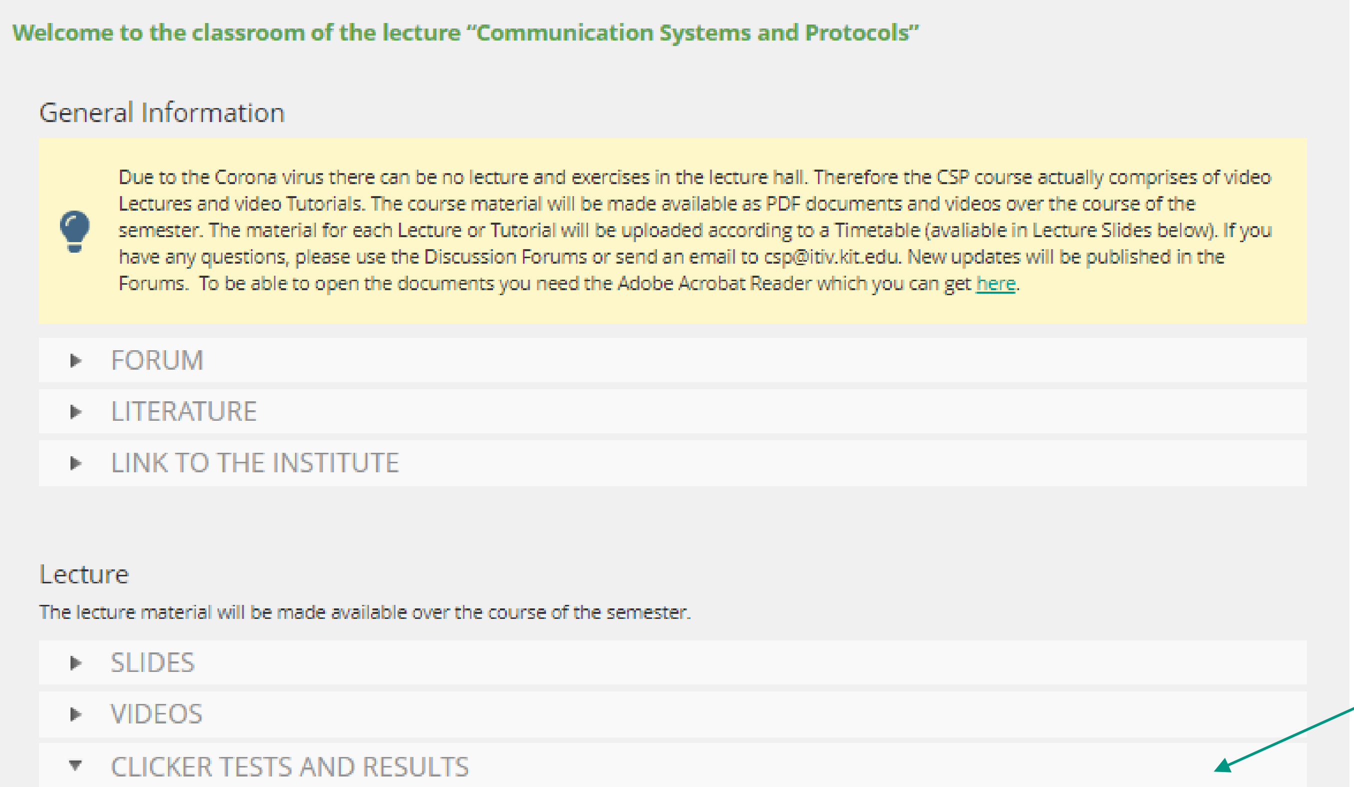
- Safe synchronization by regarding all edges of the control signals
  1. Sender is only allowed to put data on the bus if ACCEPT=0. Valid data on the bus is signaled with VALID=1.
  2. Via VALID=1 the receiver recognizes new valid data and processes it. As soon as the data has been consumed, the receiver asserts ACCEPT=1.
  3. Only if the receiver has acknowledged the data (ACCEPT=1), the sender is allowed to remove the data from the bus and sets VALID=0.
  4. When the receiver detects the de-asserted VALID, it sets the signal ACCEPT=0 as well. Only after this the sender is allowed to start a new cycle.





- Always safe independent of the sender's or receiver's speed.

Participate in Clicker Test 6a



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

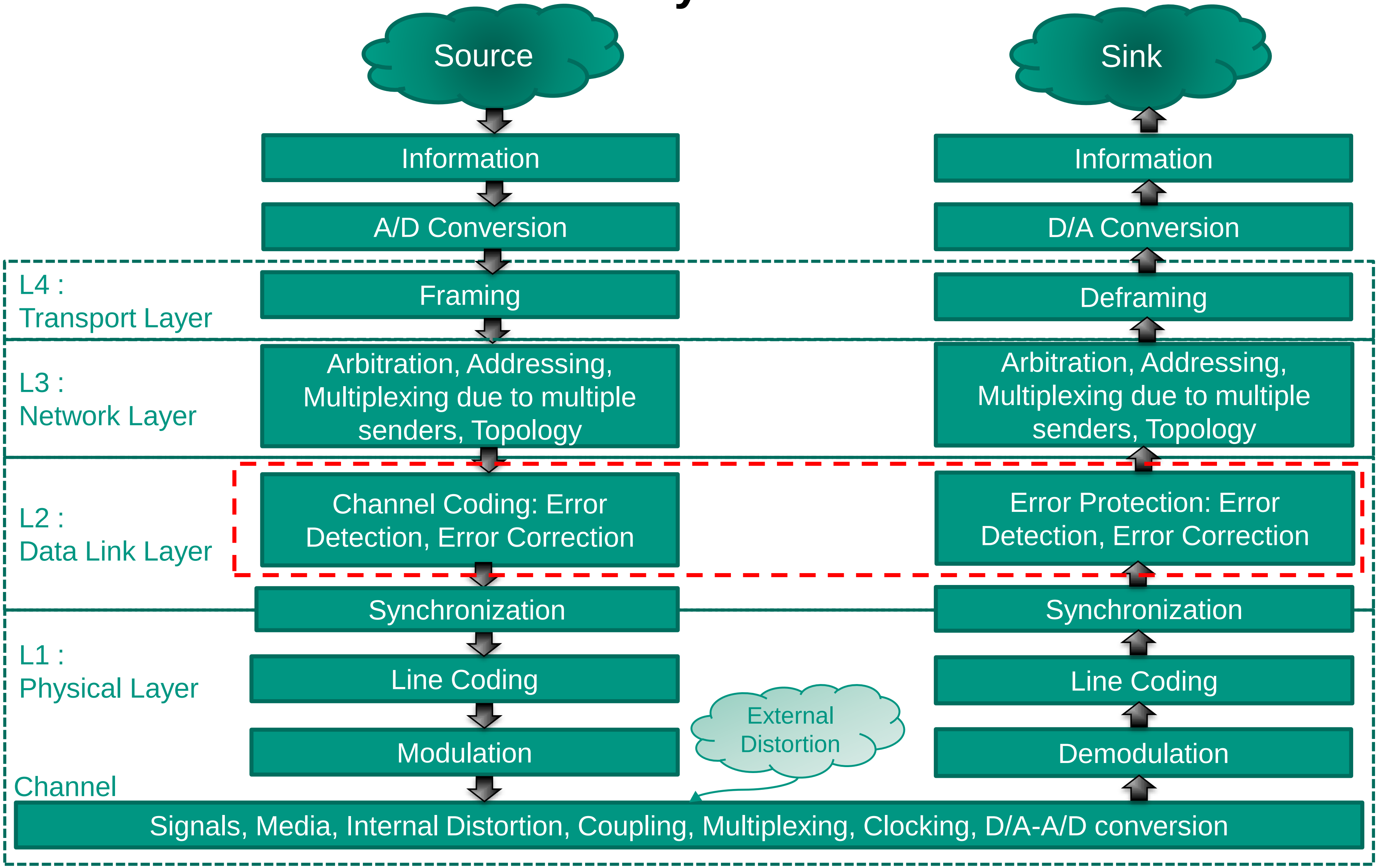
Communication Systems and Protocols

Error Detection

Contents

- Redundancy
- Hash functions
- Cyclic Redundancy Check (CRC)

Technical Communication System



Source
Sink
Information
Information
A/D Conversion
D/A Conversion
L4 :
Transport Layer
Framing
Deframing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Error Protection: Error Detection, Error Correction
Synchronization
Synchronization
L1 :
Physical Layer
Line Coding
Line Coding
External Distortion
Modulation
Demodulation
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

The difference between Safety and Security

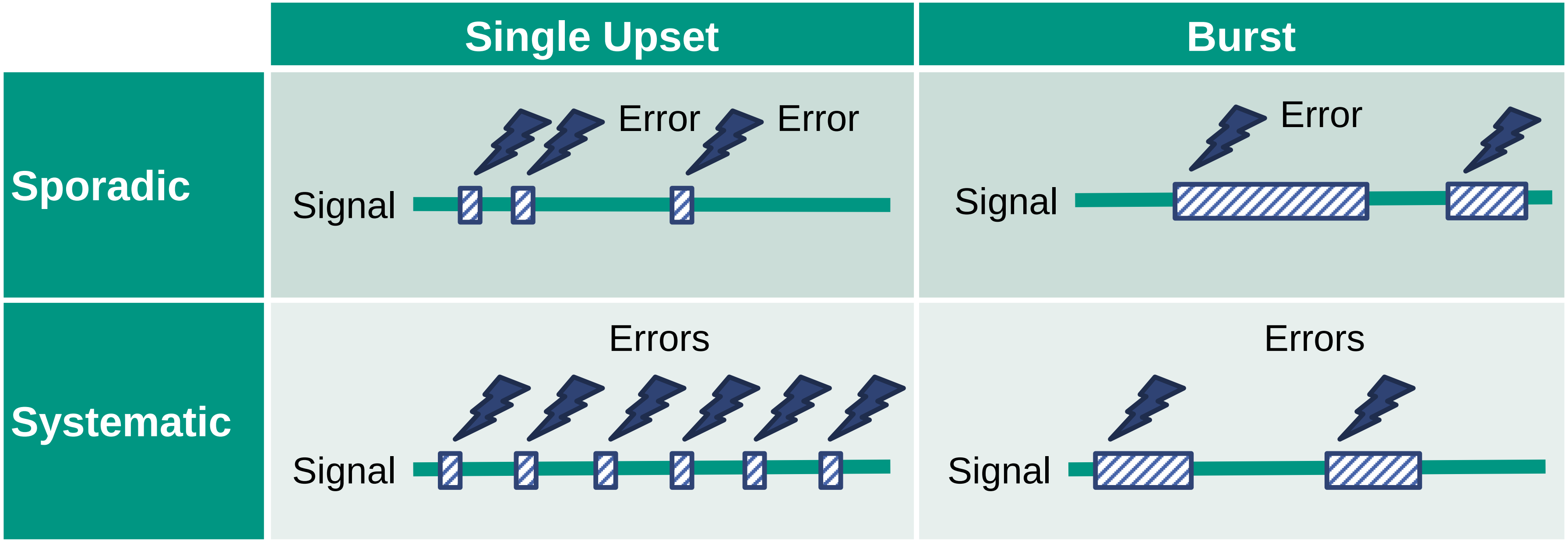
Safety:
Protection of the environment from failures of the system.

- In our case:
  - Changes in data during transmission have to be detected, so that wrong data cannot cause harm to others.
  - Protection against unwittingly errors

Security:
Protection of the information against attacks from the environment.

- In our case:
  - Willful change of information by attackers
- Security is not handled in this lecture

Defect classes



- Single upset: Only one bit is changed
- Burst: Multiple bits in sequence are changed

- Sporadic: Changes happen randomly
- Systematic: Changes follow a certain pattern

How to Deal with Errors/Defect Classes

- Errors can only be processed if they are known
  - Analysis of possible error sources is part of the design process
  - Error avoidance, error detection and/or error correction are part of the specification of a communication system

- Error avoidance can be done on physical level, e.g.:
  - Proper media (shielding, twisted pairs)
  - Signaling (differential signaling, voltage levels)
  - System design (termination)

- Nevertheless, still errors can occur
  - Error detection/correction on data level is required
  - Adding redundancy to the data, to detect/correct errors
    - Hamming codes (see lecture “Digitaltechnik”)
    - Parity checking
    - Cyclic Redundancy Check (CRC)

Redundancy

Redundancy:

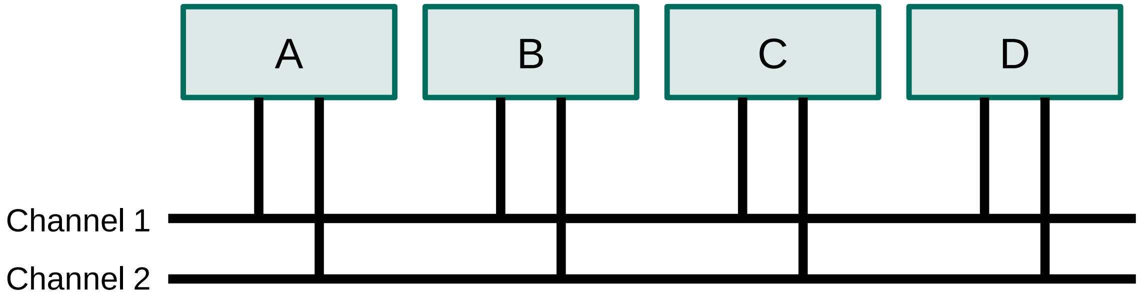
The number of bits used to transmit a message minus the number of bits of actual information in the message.

(Source: wikipedia.org)

- Redundancy can be external or in the information itself
- External (Hardware) redundancy
  - Implement the same functionality multiple times
  - E.g. transmission of the same information over two different lines
- Information redundancy
  - Natural redundancy: „wasted“ capacity of a channel that does not contain new information
  - Constructed redundancy: additional data, that has been added to protect the information

External Redundancy

- Add additional redundancy by implementing multiple instances of the communication system as a whole or in parts
  - See lecture „Systems and Software Engineering“
- Example: FlexRay
  - Two independent communication lines that can be used to transmit the same data twice, each on one line
  - An interference on one line should not have exact the same effects on the other line
  - If the received packets for these two lines differ, they are rejected and have to be transmitted again



Constructed vs. “Natural” Redundancy

- In “natural” information inherent redundancy can exist
  - Speech, music, pictures, …
  - it includes unneeded data (e.g. voice can be transmitted with small bandwidths)

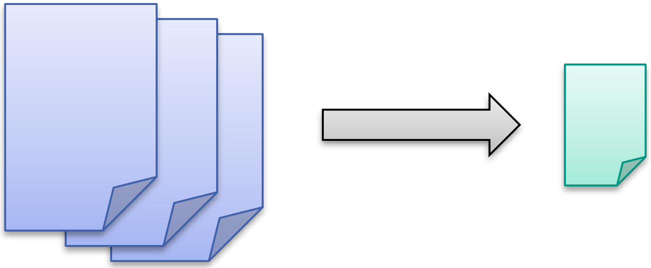
- “Natural” redundancy is…
  - … usually created randomly and often distributed uniformly
  - … has no dependable information value

- Remove “natural” redundancy and add systematically constructed redundancy

- Processing of data for transmission:
  1. Sample natural information
  2. Reduce inherit redundancy to reduce amount of data (compression)
  3. Add artificial redundancy, e.g. to add error detection capabilities

Constructed Redundancy: Hash functions

- Hash function maps large data sets (keys) to smaller data sets of fixed length (hash values):



- Properties of good hash functions (for communication purposes):
  - Different inputs should lead to different hash values (reduce number of collisions)
  - Efficient calculation of hash
  - Small change in input values → large change in hash value

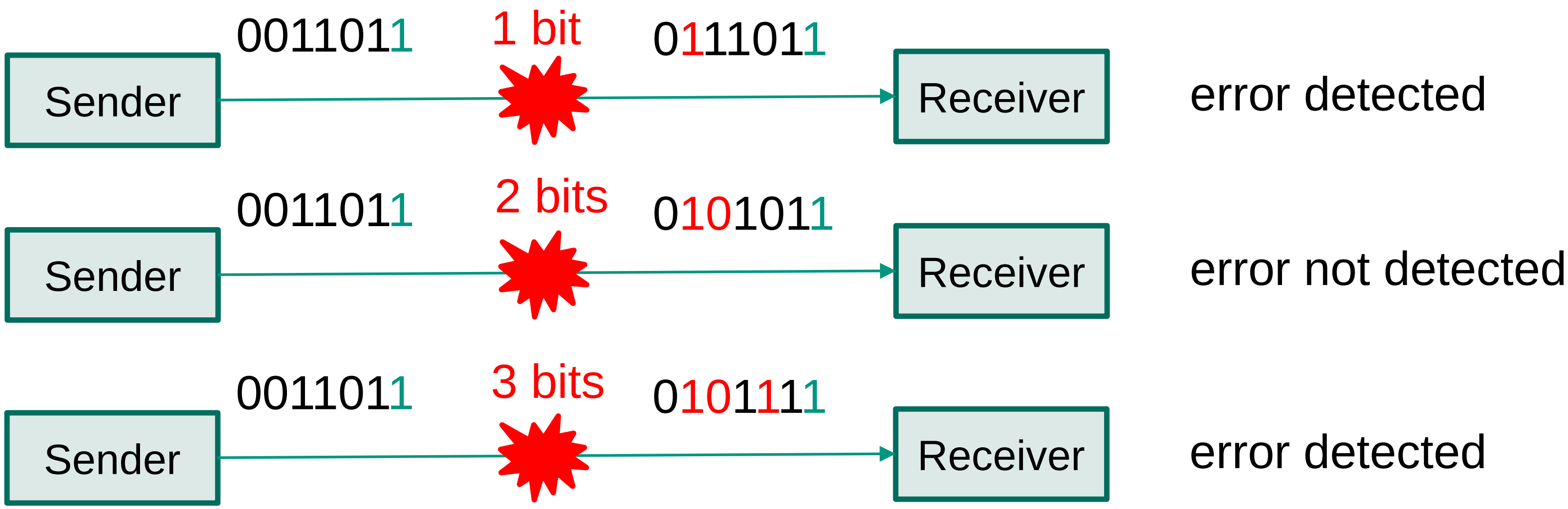
- Hash used as checksum to detect errors:
  - Cross sum
  - Parity
  - Cyclic Redundancy Check (CRC)

Methods for Error Detection based on Redundancy (recap)

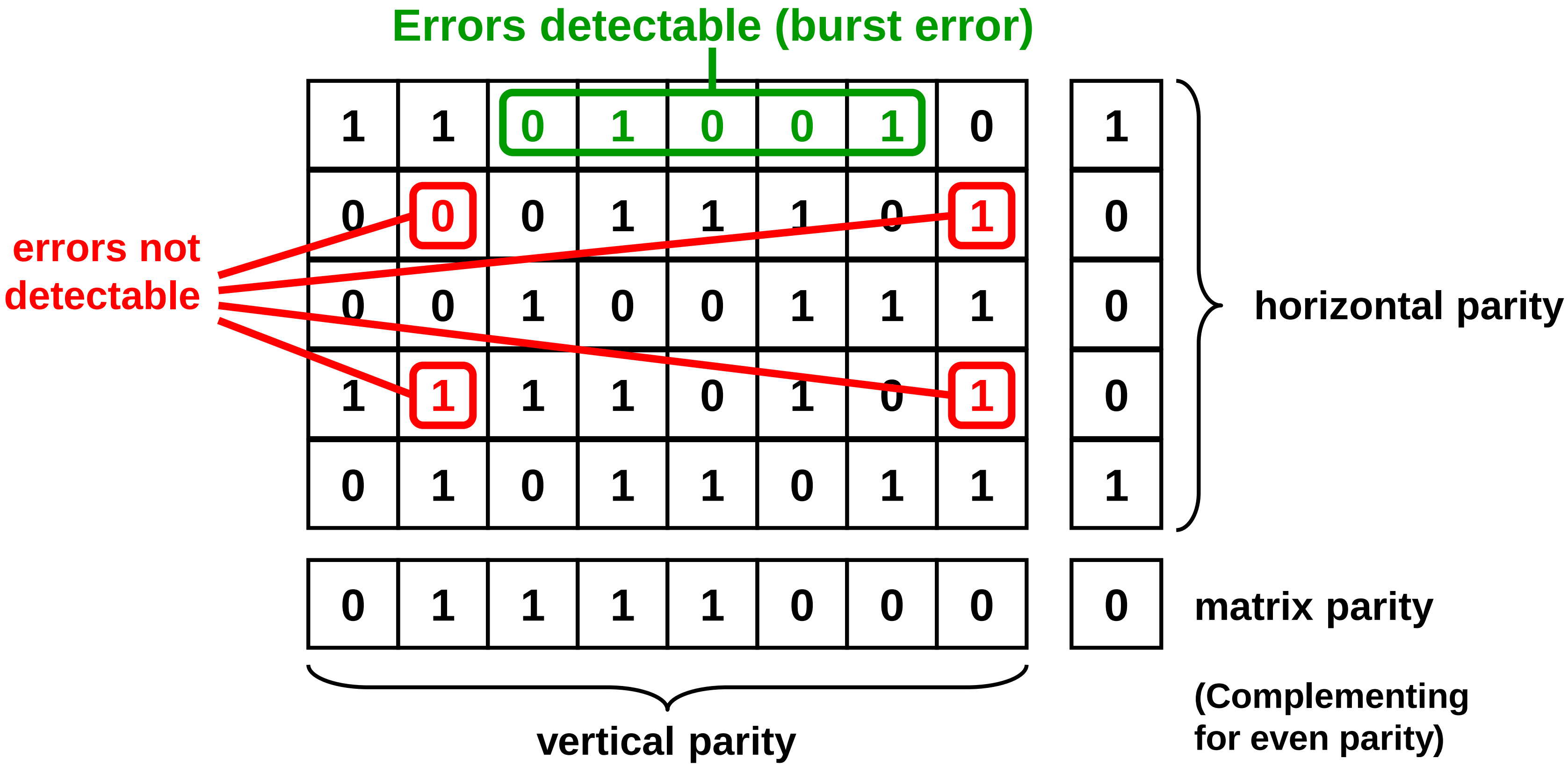
- **Parity**
  - Complementing a data word with an extra (single) bit to make the number of bits within the data word carrying ‚1‘s <even> or <odd>
  - Can be used column-wise (vertical parity) or row-wise (horizontal parity)
  - Detection of an odd number of errors (1, 3,..).
- **Block Check**
  - A set of n data words is protected by vertical parity and horizontal parity
  - Detection of burst errors of length n is possible
- **XOR Block Check**
  - For every 2 data words their XOR concatenation is send as a third data word. These 3 data words allows to reconstruct all data from any two data words. It has to be known which data word is erroneous!
  - Used in RAID arrays.
- **Hamming Code**
  - Exploiting the hamming distance between valid code words, meaning that not all code words that can be generated with the available bits are valid code words

Example: Parity

- Parity : Considering the ‘even’ case
  - Sender: 001101
    - to make the number of 1s ‘even’ a parity bit ‘1’ is added -> 0011011
  - Receiver checks parity. If the number of 1s is
    - Even = no error detected, data is extracted
    - Odd = error detected, no data extracted
- Detects odd number of errors (1, 3,..).



Example: Block Check



Errors detectable (burst error)

errors not detectable

1	1	0	1	0	0	1	0	1
0	0	0	1	1	1	0	1	0
0	0	1	0	0	1	1	1	0
1	1	1	1	0	1	0	1	0
0	1	0	1	1	0	1	1	1
0	1	1	1	1	0	0	0	0

horizontal parity

matrix parity

(Complementing for even parity)

vertical parity

- Detectable bit errors:
  - All odd number of errors
  - All even number of errors with odd number of bit errors per row/column
  - All 2-Bit errors
  - Most 4-Bit errors (except “rectangular” bit errors)

Example: XOR Block Check

- Using RAID 4 as an example, Assume four drives contain the information and the XOR of them is placed in the fifth drive

Drive	**Data**							
1	1	1	0	1	0	0	1	0
2	0	1	1	0	0	0	0	1
3	1	1	1	1	1	1	1	1
4	1	1	1	0	0	0	0	1
5	1	0	1	0	1	1	0	1

xor

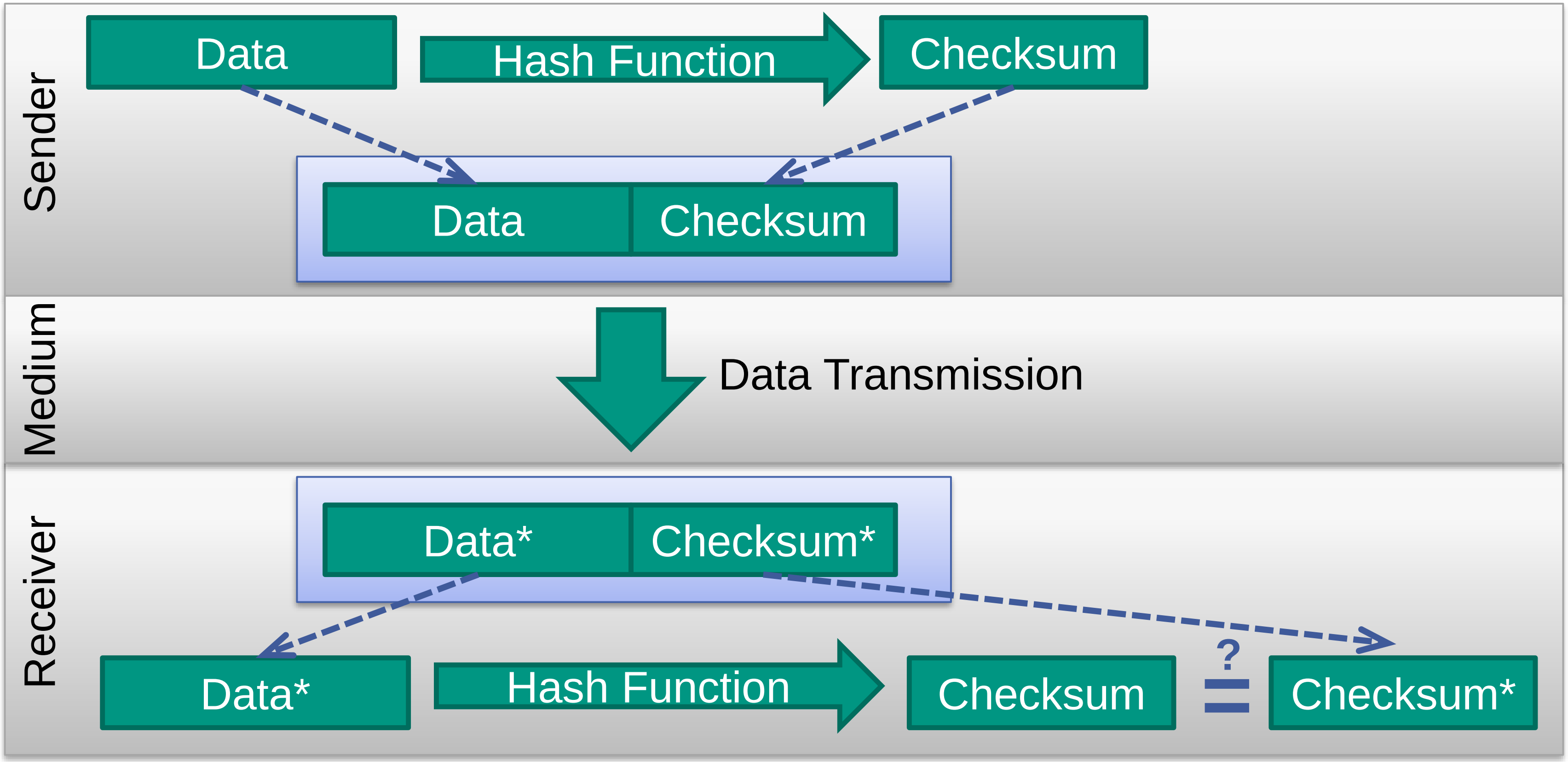
- If any drive fails, the data can be recovered using information from the other four drives. If drive 4 fails then
  - Drive4 = Drive1 XOR Drive2 XOR Drive3 XOR Drive5

Drive	**Data**							
1	1	1	0	1	0	0	1	0
2	0	1	1	0	0	0	0	1
3	1	1	1	1	1	1	1	1
5	1	0	1	0	1	1	0	1
4	1	1	1	0	0	0	0	1

xor,
data is recovered

Cyclic Redundancy Check (CRC) – Approach

- Generate a checksum (Hash)
- Send checksum together with raw data
- Receiver checks received data and received checksum match



CRC –mathematical description

- Generation of checksum uses a generator polynomial *G(x)* on sender and receiver side
  - most significant bit (MSB) and least significant bit (LSB) must be ‚1'

- Calculation on sender side
  - Data vector *M(x)* of length *m* bit (exceeding length of *G(x)*)
  - Checksum equals the modulo of the division $(x^r M(x))/G(x)$
    - *r:* degree of generator polynomial
    - $x^r M(x)$ adds *r* zero bits to the end of the data vector
  - Checksum is appended to the data
    - Corresponds to the addition of the modulo: $x^r M(x) + R$

- Calculation on receiver side
  - Division of received data by *G(x)*
    - If the modulo of the division is not equaling zero an error has occurred
    - If the modulo of the division equals zero, no error has been detected

CRC – Hardware Realization

- Can be easily and efficiently realized in hardware.
- Polynomials are expressed as bit strings
  - e.g.: $M(x)=x^5+x^4+x^0$ corresponds to 110001

- Calculations based on division modulo-2 arithmetic, no carries
  - Corresponds to a bitwise XOR operation (⊕)

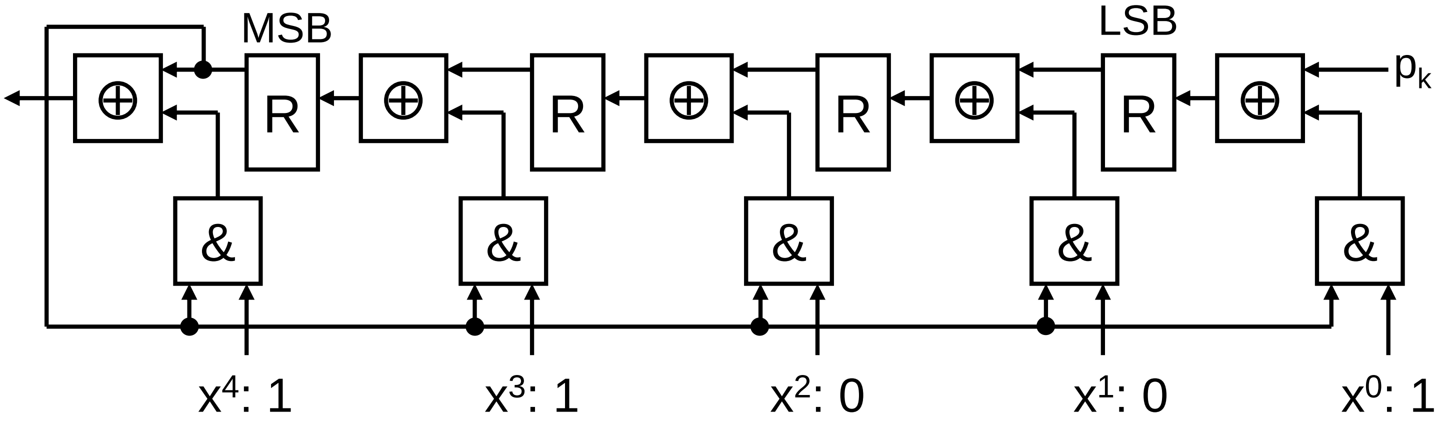
⊕	0	1
0	0	1
1	1	0

  - Example for bitwise XOR operation:

$$\begin{array}{r} 10011011 \\ \oplus\,\underline{11001010} \\ 01010001 \end{array}$$

Hardware Implementation

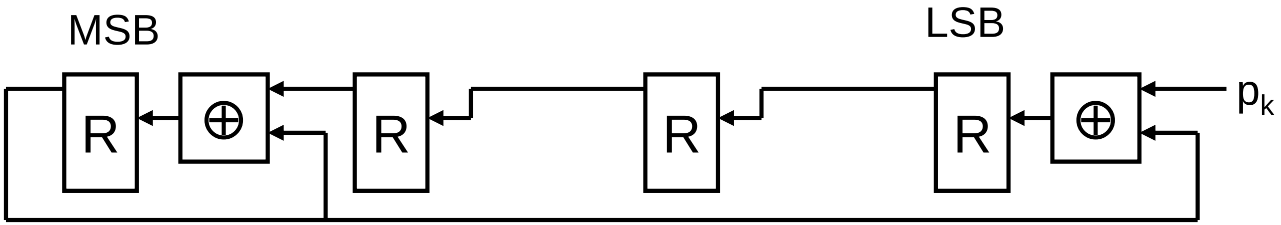
- Utilization of linear feedback registers with XOR-operations
  - Example: Generator polynomial $\mathbf{x^4+x^3+1 = 11001}$



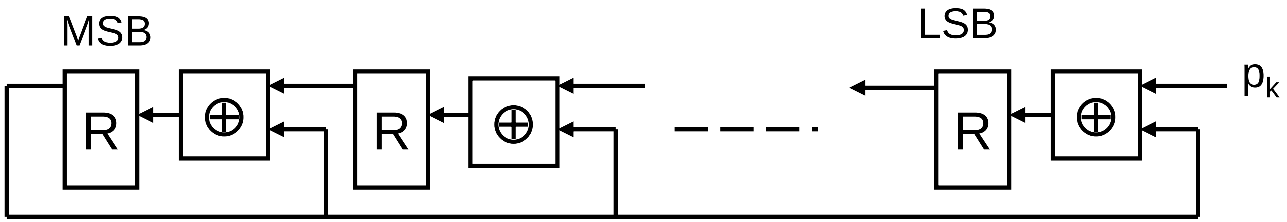
- Data vector is shifted bitwise through the register
- Content of MSB denotes the current quotient bit
- If MSB=1, the generator polynomial is added (XOR gate) and the dividend is shifted left (corresponds to a right-shift of the generator)
- If MSB=0, only the shift is executed
- Input data is shifted until a logic 1 is available in the MSB

Hardware Implementation

- Simplification for fixed generator polynomial:
- Example: Generator polynomial **x^4+x^3+1 = 11001**
  - AND-Gates can be omitted
  - XOR-gates only required for factors ≠ 0 in the generator polynomial
  - last XOR-gate can be omitted as the output is not used



- General representation



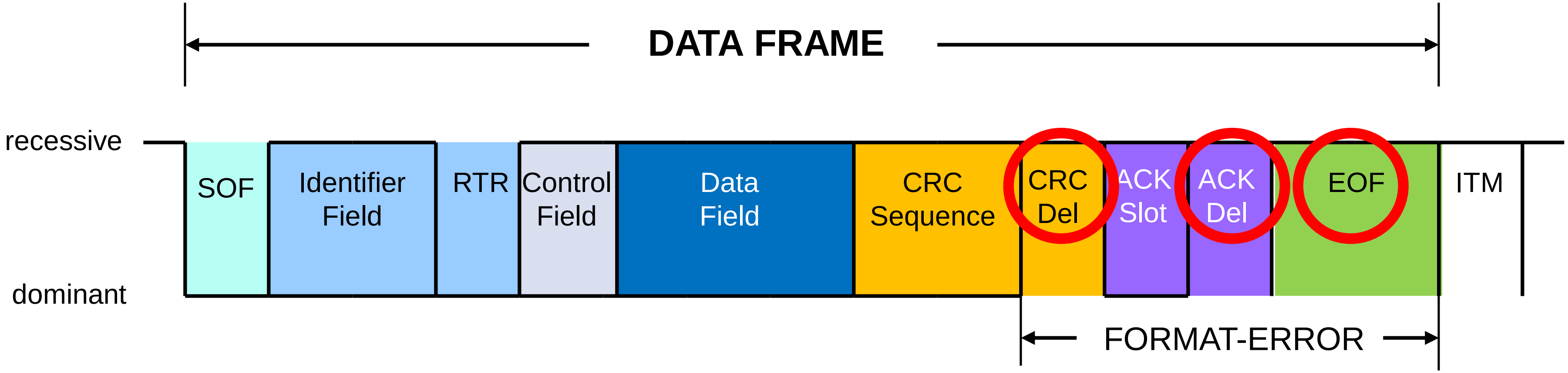
CRC Properties

- CRC can detect the following errors (without proof):
  - All single-bit errors
  - Two single-bit errors if *(x^k+1)* is not divisible by the generator polynomial (k $\leq$ frame size)
  - Odd number of bit errors if *G(x)* contains factor *(x+1)*
  - All error bursts $\leq$ degree of generator polynomial

- Examples for standardize polynomials:
  - CRC-12 $= x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16 $= x^{16}+x^{15}+x^2+1$
  - CRC-CCITT $= x^{16}+x^{12}+x^5+1$
  - CRC-32 $= x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Format Error Detection

- Example: Error detection mechanism for the CAN Bus



- Detection of Format Errors:
  - CRC Delimiter as well as ACK Delimiter and EOF are bits with a fixed value and a fixed position within the frame
  - If these bits have changed, an Format Error is detected

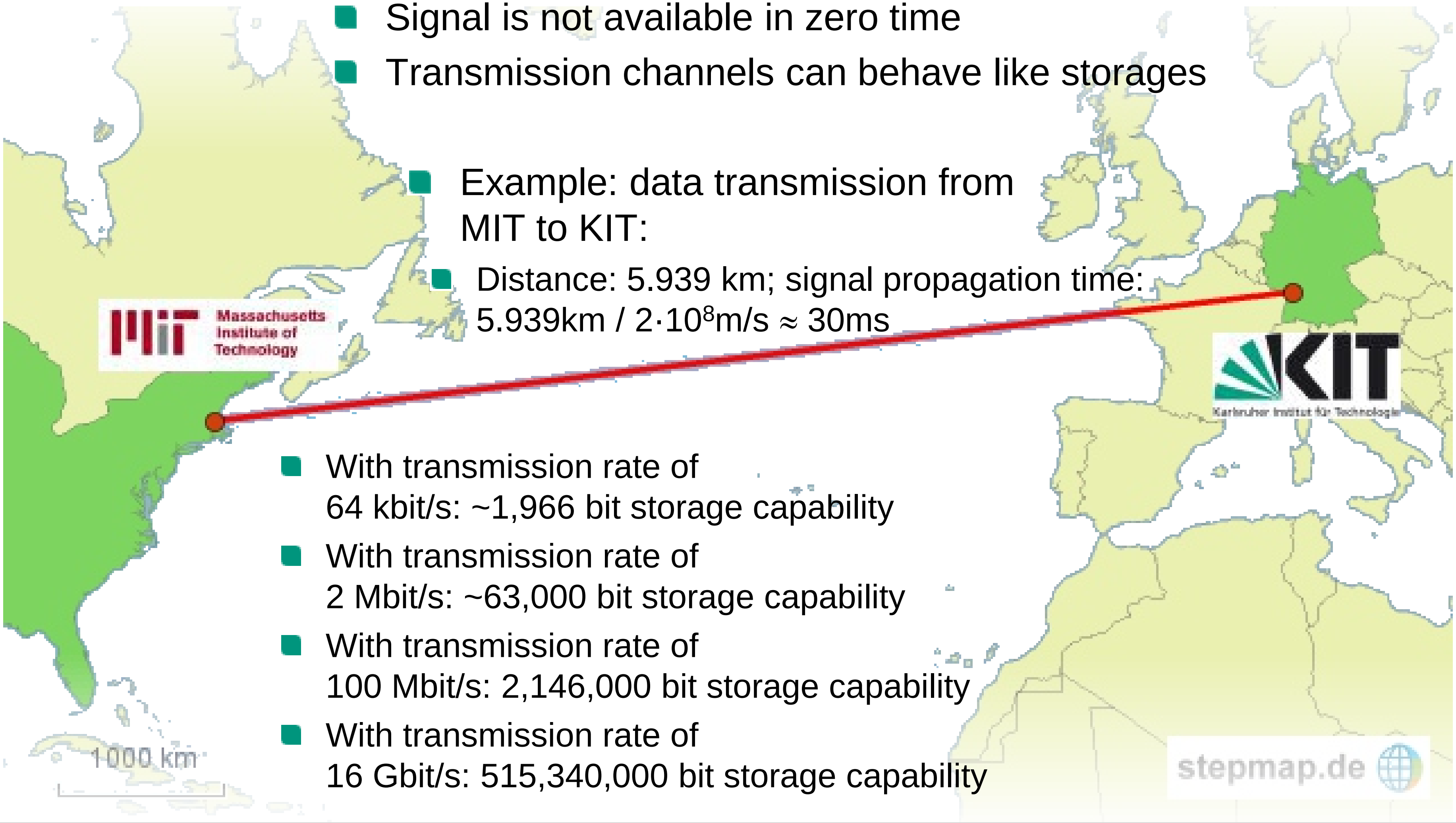
Error Handling

There are several options how to deal with detected errors

- Resend
  - the faulty data packet is send once more
  - increases communication time
  - can clog the system
  - 100% correct data transmission is possible
- Correction
  - Faulty data is reverted to correct data
  - More redundancy necessary (internal or external)
  - Not always guaranteed that errors are correctable
- Discard of faulty data packet
  - data is lost
  - allows maximum throughput
  - often used in systems with isochronous data transfers (e.g. video streaming applications)

Signal propagation on Media

- Signal propagation speed is limited (approx. 0.6*c)
  - Signal is not available in zero time
  - Transmission channels can behave like storages


Massachusetts Institute of Technology
Karlsruher Institut für Technologie
1000 km
stepmap.de

- Example: data transmission from MIT to KIT:
  - Distance: 5.939 km; signal propagation time: 5.939km / $2 \cdot 10^8$m/s ≈ 30ms

- With transmission rate of 64 kbit/s: ~1,966 bit storage capability
- With transmission rate of 2 Mbit/s: ~63,000 bit storage capability
- With transmission rate of 100 Mbit/s: 2,146,000 bit storage capability
- With transmission rate of 16 Gbit/s: 515,340,000 bit storage capability

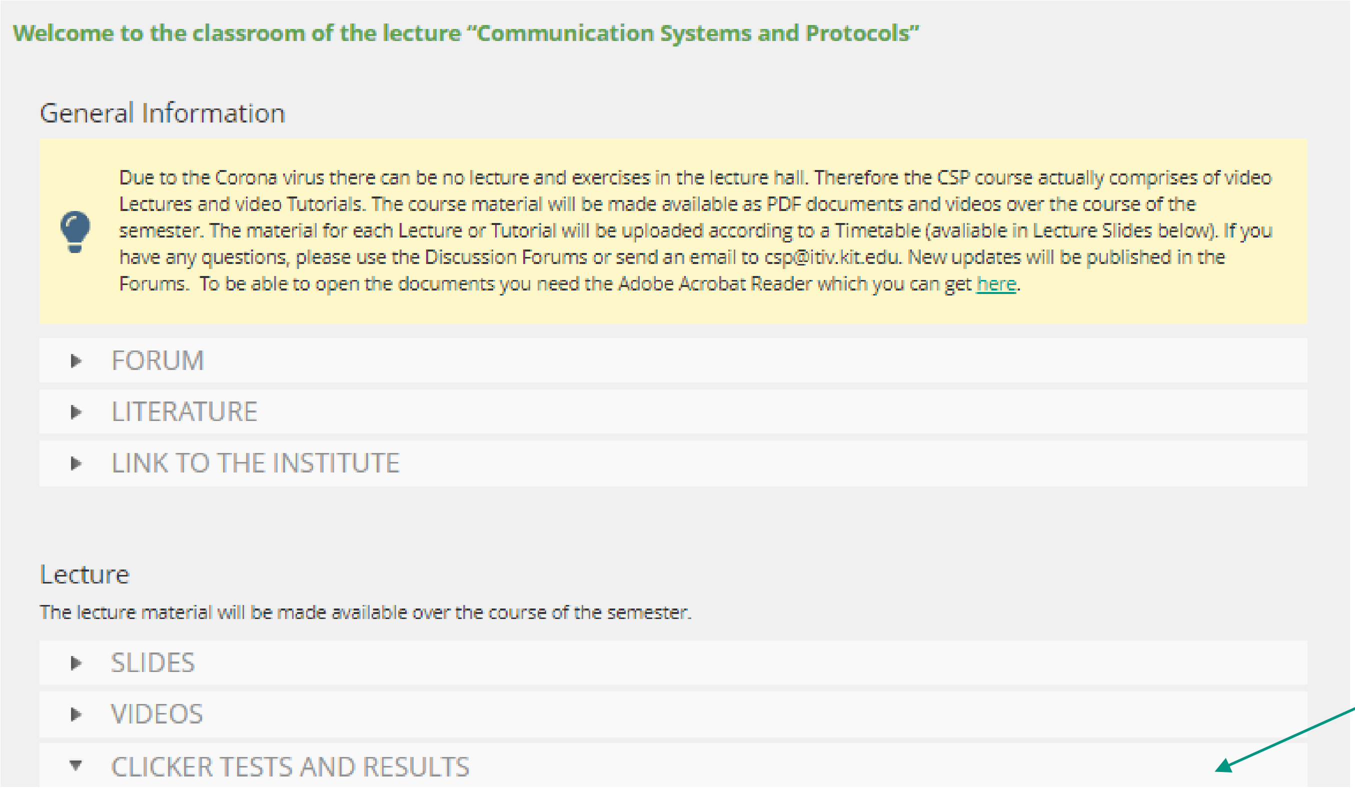
Error Management

- System Analysis how many errors shall be handled
  - prior to implementation
  - type of errors
  - statistical analysis
  - no guarantees given that real system always adheres to the assumed constraints

- Errors on System level
  - faulty nodes
  - faulty lines
  - wrong system assumptions about error rate

- Global error management
  - assures that errors do not affect the complete system
  - Separation of faulty nodes from the bus
  - Status reports to higher authorities (system monitor, users, etc.)

Participate in Clicker Test 6b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

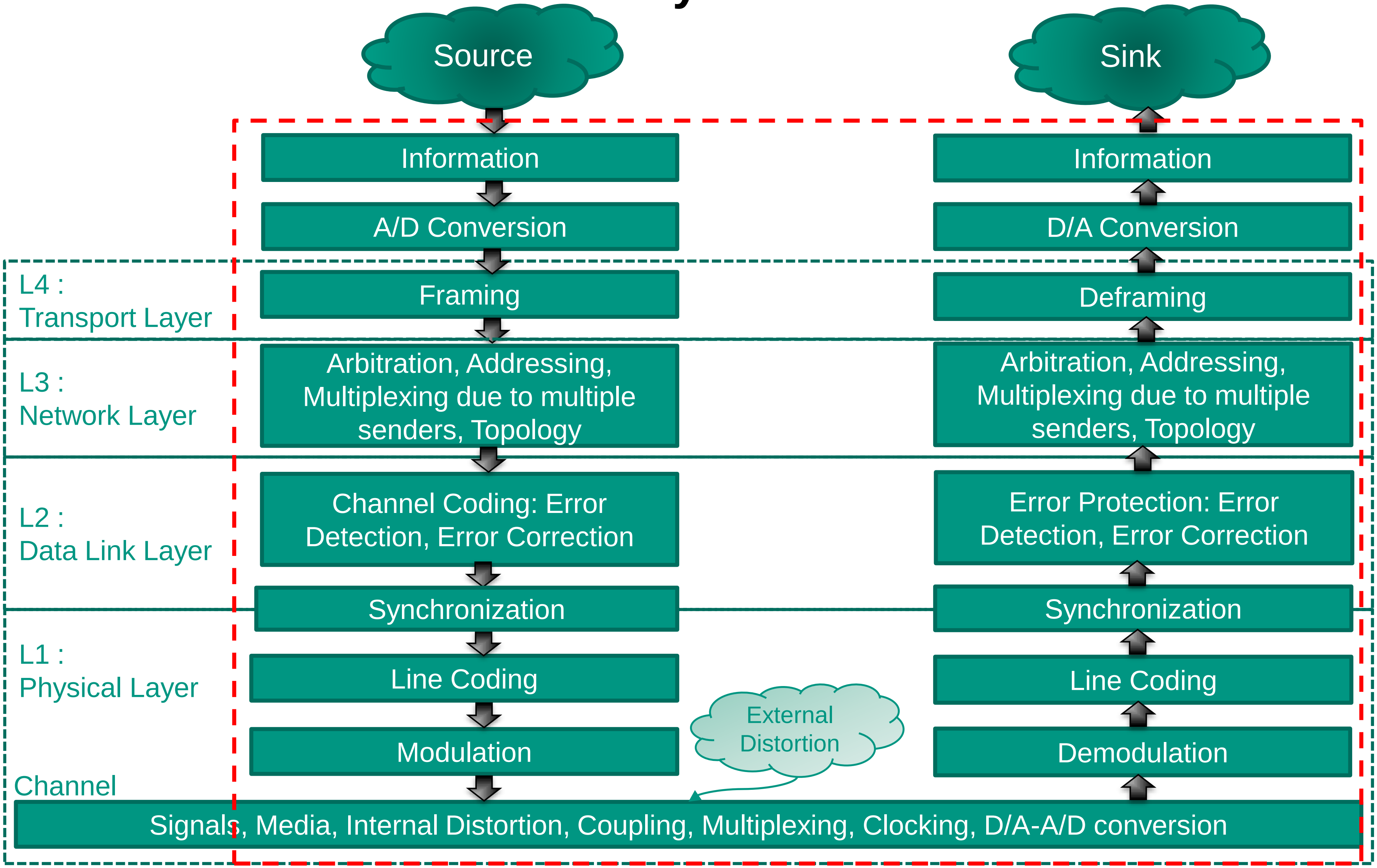
Classification of Communication Systems

Contents

- Properties of Communication Systems
  - Technical
  - Physical
  - Econimical
  - Usability

- Classes of Communication Systems
  - System Busses
  - Peripheral Busses
  - Field Busses
  - Automotive Busses
  - Networks

Technical Communication System


Source
Sink
Information
A/D Conversion
Information
D/A Conversion
L4 :
Transport Layer
Framing
Deframing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Error Protection: Error Detection, Error Correction
Synchronization
Synchronization
L1 :
Physical Layer
Line Coding
Line Coding
Modulation
Demodulation
External Distortion
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

Properties of communication systems

- All Features presented up to now have effects on the properties of communication systems
- What are more general properties of such systems?
- How can we try to compare them?

Time for Transmission

- Transmission Rate
  - how many bits can be transferred per time unit
  - influenced by clock rate and/or degree of parallelism

- Latency
  - Latency is the time between start and fulfillment of a request
  - Is highly related to the time the bus master has to wait to gain control over the bus
  - Minimizing the latency is crucial for real-time applications

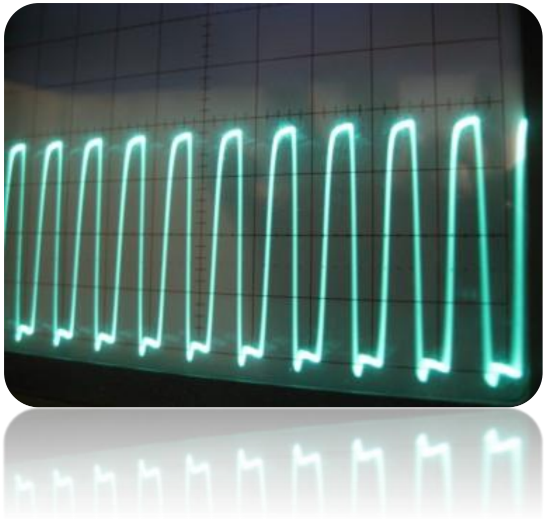
Request

Answer

Source: meshIP Blog

Clock Rate

- Transmission rate is growing with the clock rate.
  - But not all devices depend on increasing clock rates.
- High clock rates are hard to implement from a technical point of view.
- In summary this would result in uneconomical bus couplers in regard to many applications.



Real-Time Capability

Definition of Real-Time:

In a system it has to be guaranteed that any communication can be fully performed within an arbitrarily chosen but fixed time span.

- Any node wanting to transmit data is able to do so within the given time span.
- Time span can be any defined time
  - typical values in the order of milliseconds
  - denotes the maximum time span for a message to be transferred from sender node to receiving node
  - start and end of transmission has to lie within this time span

Error Handling

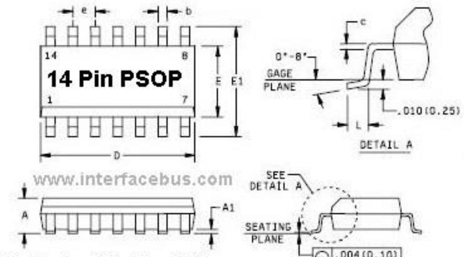
- Error Detection Capabilities
- Error Correction Capabilities

- Redundancies
  - physical redundancy, e.g. TMR (triple modular redundancy)
  - logic redundancy, e.g. encoding, CRC, protocols, …

- Robustness
  - how tolerant is the system against errors or failures
  - even under bad circumstances the system will still work correctly

Physical Properties

- Dimensions
  - Smaller dimensions of bus components also demand smaller scaled bus connectors, but also open new perspectives:
    - Bus lines get shorter, resulting in smaller capacities and enabling higher clock rates
    - More functionality/volume, resulting in cheaper bus components



Plastic Small Outline Package

Dimensions							
Symbol	Inches		Millimeters		Symbol	Inches	
	Min	Max	Min	Max		Min	Max
A		.069		1.75	E	.150	.157
A1	.004	.010	0.10	0.25	E1	.228	.244
b	.014	.020	0.35	0.51	e	.050 BSC	
c	.008 NOM		0.20 NOM		L	.016	.044
D	.337	.344	8.55	8.75			

Source: interfacebus.com

- Low-Power
  - Decrease of energy consumption
  - Decrease of temperature and thus cooling problems
  - With decreasing energy consumption, increase of clock frequency is possible

Economic Properties

- Cost
  - Important factor for high-volume systems
  - Cost reduction improves the chances of a bus system to penetrate the market
  - Can be achieved in various ways:
    - Usage of standardized components
    - Simple and easy to use standard protocols
    - Off-the-shelf libraries, tools, development flow etc.

- Licensing Issues
  - license costs?
  - Legal questions (who is allowed to use it)?

Source: aktivgegenabschiebung.de

System Setup

- Size of the system
    - Number of nodes
    - Number of parallel transmissions
    - Coverage area
- Scalability
    - Extension of the communication system possible?
    - Variation in data rate
- Compatibility
    - Integration of older components
    - Evolving specifications
    - Interaction of different bus systems

Usability

- Hot-Plug Capability
- Configuration
- Mechanic Issues
  - Plugs
  - Housing
  - Cabling

- Quality-of-Service
  - Guarantee of bandwidth
  - Real-Time Capability
  - Response Time
  - Error Free-ness
  - Isochronous Transfer
    - Communication in equidistant time intervals
    - Each time interval carries a communication

Behavioral Requirements for Busses

- Functional Requirements
  - A bus system shall be easy to use and shall be easily extendable
  - Userfriendly
  - Easy to implement protocols
- Flexibility
  - A bus system shall have a wide range of possible uses. This results in different (sometimes contradictory) requirements:

Low-Cost-Domain:	Cheap, small, compatible
High-End-Domain:	Performance
Mobile Devices:	Small, Low-Power

- Performance
  - Bus system shall not be the limiting factor of the system
  - Different components have different requirements
  - Transmission speed can not only be designed for the fastest component as this would lead to very high cost without significant advantages

Definition of Classes

- System Busses
- Peripheral Busses
- Field Busses
- Automotive Busse
- Networks

System Busses

- Often used within computers
- Connect internal devices (memory, IO-devices, processor)
- Usually limited to PCB
- Coverage area typically in the order of centimeters
- Data rate ranges from kByte/s to Gbyte/s

- Examples:
  - I^{2}C-Bus
  - SPI
  - PCI
  - PCI-Express

Peripheral Busses

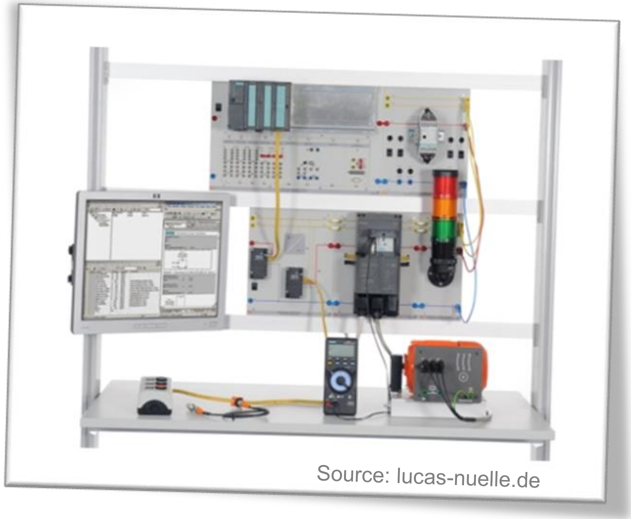
- Used to connect peripheral devices to a computer system
- Simple installation and configuration desirable
- Coverage area typically in the order of centimeters to meters
- Data rates from kByte/s to Mbyte/s

- Examples:
  - RS-232
  - USB
  - FireWire

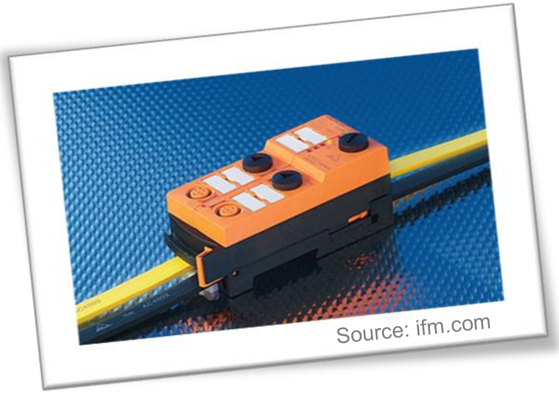
Process and Field Busses

- Used in industrial environments for process automation
- Increased demands in robustness and safety
- Often includes real-time demands
- Coverage area typically hundreds of meters with many nodes
- Data rates from Byte/s up to MByte/s

- Examples:
  - ASI
  - ProfiBus
  - KNX
  - EIB



Source: lucas-nuelle.de



Source: ifm.com

Automotive Busses

- Used to couple embedded control units (ECU) and sensors/actuators in cars
- High demands in terms of robustness and safety
- Cost sensitive
- Coverage area in the order of meters
- Data rates from Byte/s to MByte/s

- Examples:
  - LIN
  - CAN
  - FlexRay

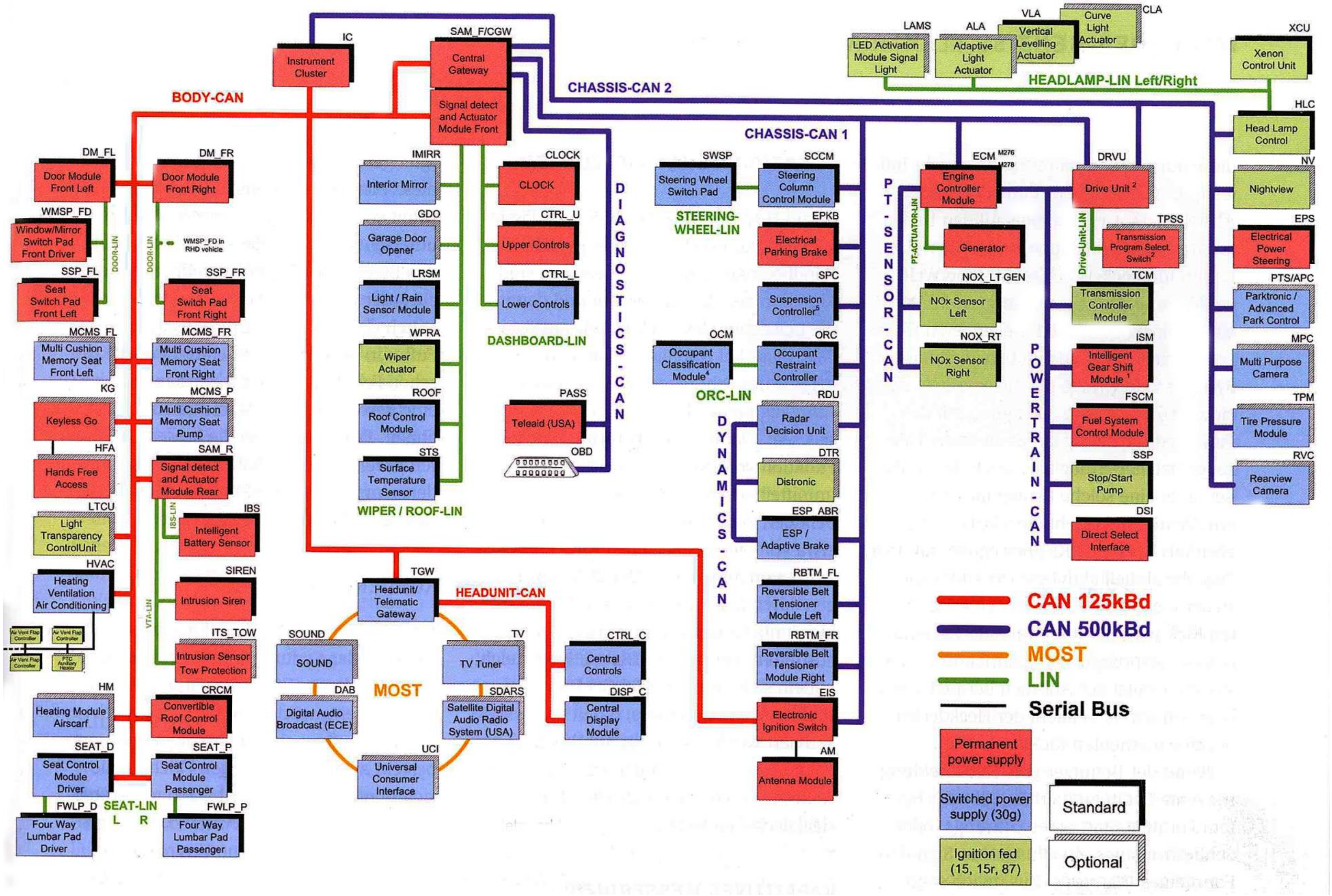
Source: melexis.de

Hierarchy of bus systems

- There is no one perfect communication system for every use case
- Different Applications/Tasks have different requirements
  - real-time capabilities
  - bandwidth
  - prioritization
  - …

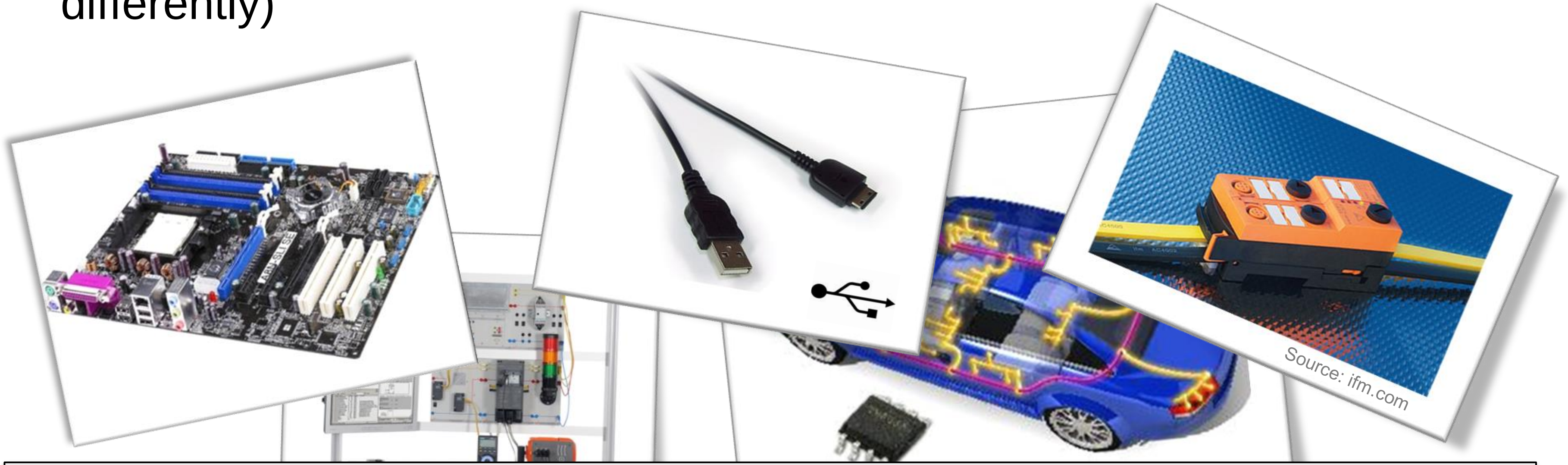
- Solution:
  - Clustering of communication nodes with similar demands onto one bus (bus domain)
  - Coupling of different busses and bus types within a system
    - Allows cross-domain information transfer (Gateway)

Hierarchy of an automotive bus system



Conclusion

- There is no perfect one solution for every use case
- Advantages and Disadvantages…
  - … differ from one system to another
  - … are of different importance under different circumstances
- Classification is not always sharp (some busses can be classified differently)



Source: ifm.com

Conclusion: Different bus systems for different applications.

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

I^{2}C-Bus

I^{2}C-Bus (Inter-Integrated-Circuit-Bus)

- Serial Bus System
- Multi Master / Multi Slave
- CSMA/CA Arbitration Scheme
- Low Cost Bus System

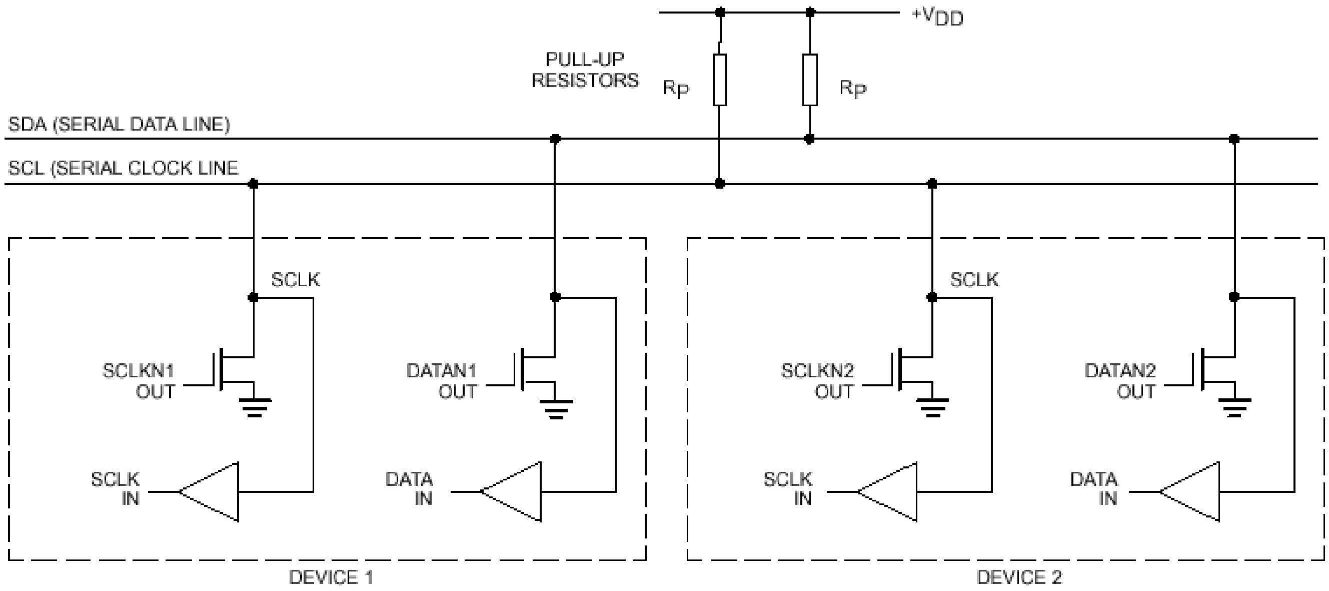
I^{2}C-Bus

- Developed by Philips for interconnecting ICs on printed circuit boards
- Only two wires for data transmission
  - Clock line SCL (serial clock)
  - Data line SDA (serial data)
  - Lower cost and less error-prone because of low pin-count
- Multi-Master, Multi-Slave
- Data transmission in packets of 8bit
  - 7bit for addresses → 128 addresses
  - 1bit for toggling between reading/writing
- Transfer rate:
  - 100kbit/s in standard mode
  - 400kbit/s in fast mode
  - 3.4Mbit/s in high-speed mode

Source of I^{2}C images and diagrams: *NXP: UM10204, I2C-bus specification and user manual, Rev. 4 — 13 February 2012*

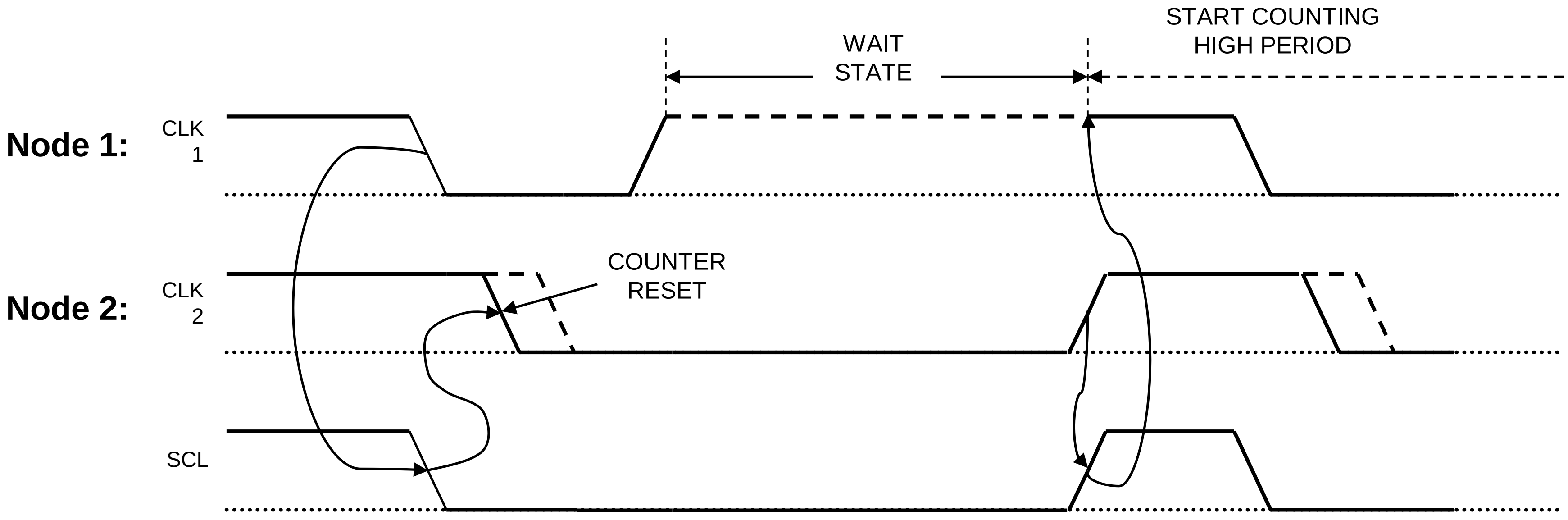
I²C Bus Connection

- Open-Collector outputs for every bus subscriber
  - Wired-AND with dominant ‚LOW‘ on the bus
  - Pull-Up resistors externally connected
- In idle mode, both SCL and SDA are ‚HIGH‘
- Concurrent monitoring of the bus at every subscriber



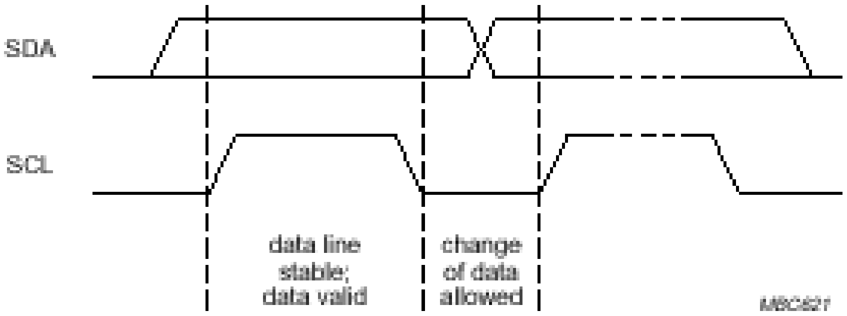
Clock Synchronization

- Clock signal is the sum generated by all nodes
- A slower node can pull down the bus to ‚LOW‘ in order to insert wait states
- The next cycle is not started before SCL is back to ‚HIGH‘

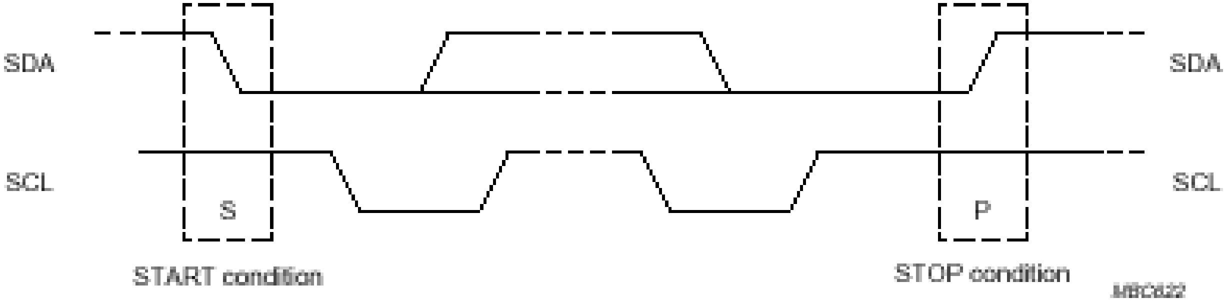


Data Validity

- Data on the bus is only valid when SCL is assigned ‚HIGH‘



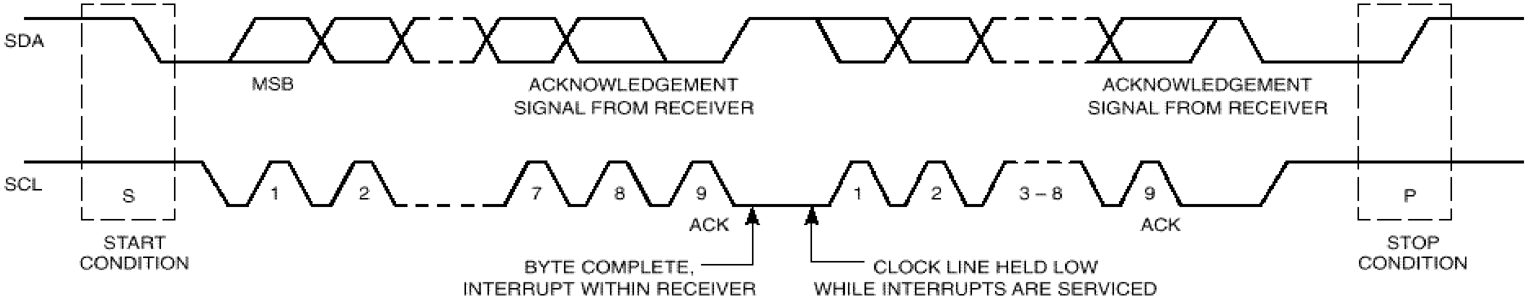
- A switch at SCL = ‚HIGH‘ has a special meaning
  - A switch of SDA from ‚HIGH‘ to ‚LOW‘ is interpreted as start of a transmission (start condition)
  - A switch from ‚LOW‘ to ‚HIGH‘ marks the end of a transmission (stop condition)



Data Transmission

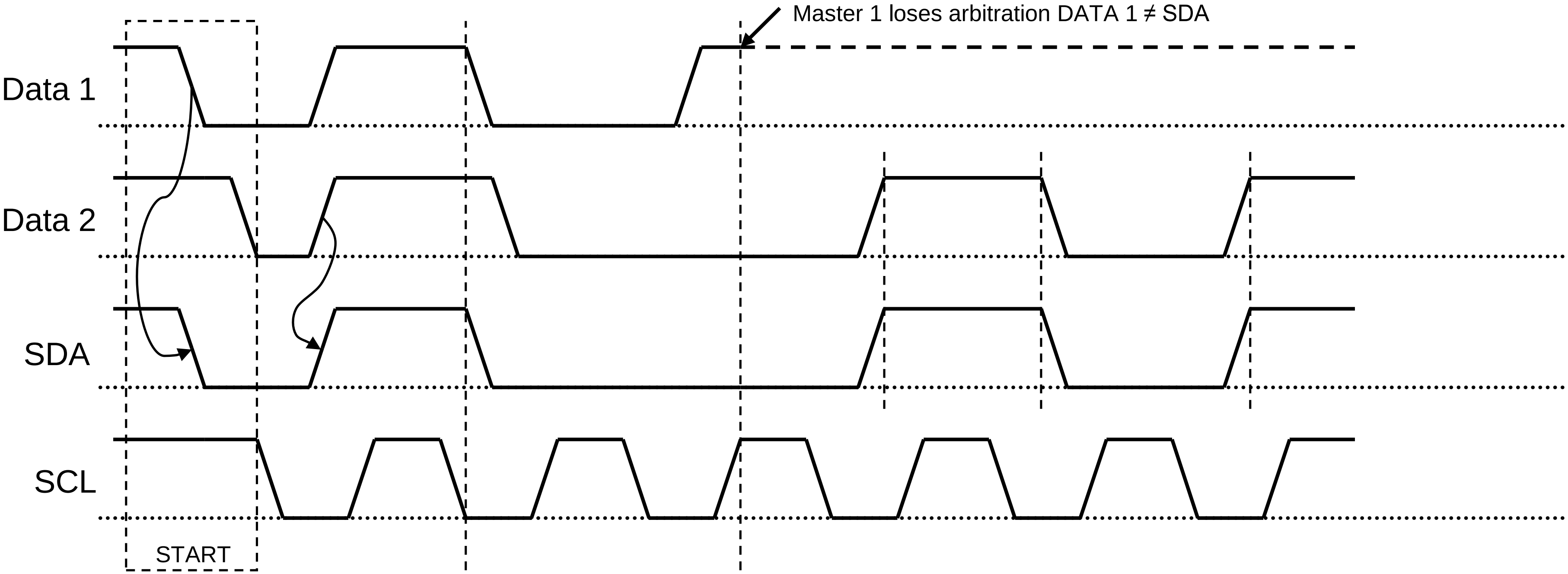
- Data is accepted with the rising edge of the SCL-line
- If a slave can not accept data in time it pulls the SCL-line down to ‚LOW' in order to insert wait states

- 8 bits per packet, MSB first
- One additional acknowledge bit per packet
  - Receiver pulls SDA to ‚LOW' for one clock cycle after data reception
  - If the bus stays ‚HIGH', the transfer has to be redone or canceled


SDA
MSB
ACKNOWLEDGEMENT SIGNAL FROM RECEIVER
ACKNOWLEDGEMENT SIGNAL FROM RECEIVER
SCL
S
1
2
7
8
9
ACK
1
2
3 – 8
9
ACK
P
START CONDITION
BYTE COMPLETE, INTERRUPT WITHIN RECEIVER
CLOCK LINE HELD LOW WHILE INTERRUPTS ARE SERVICED
STOP CONDITION

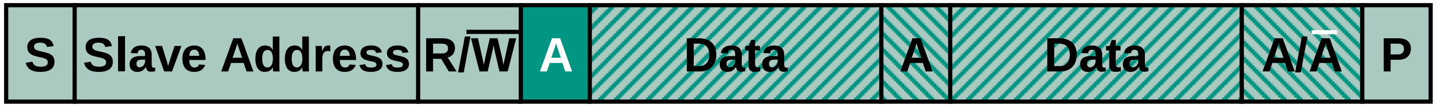
Arbitration

- A master is only allowed to send if the bus is free
  - SCL and SDA are ‚HIGH‘
- Several masters can send at the same time
- While sending, all subscribers read back the bus
- If a master detects that data on the bus is not matching the data it has send, it gets passive and does not send further data for this cycle
- No data loss on the bus → CSMA/CA



Framing

- 7bit address of the selected slave is send first
- The 8th bit denotes whether it is a read or write access
- Next, data is transmitted in packets of 8 bit
- After 8 bits there is one acknowledge bit



S | Slave Address | $R/\overline{W}$ | A | Data | A | Data | $A/\overline{A}$ | P

A= acknowledge (SDA ,LOW‘)

□ From Master to Slave

*A= not acknowledge (SDA ,HIGH‘)

■ From Slave to Master

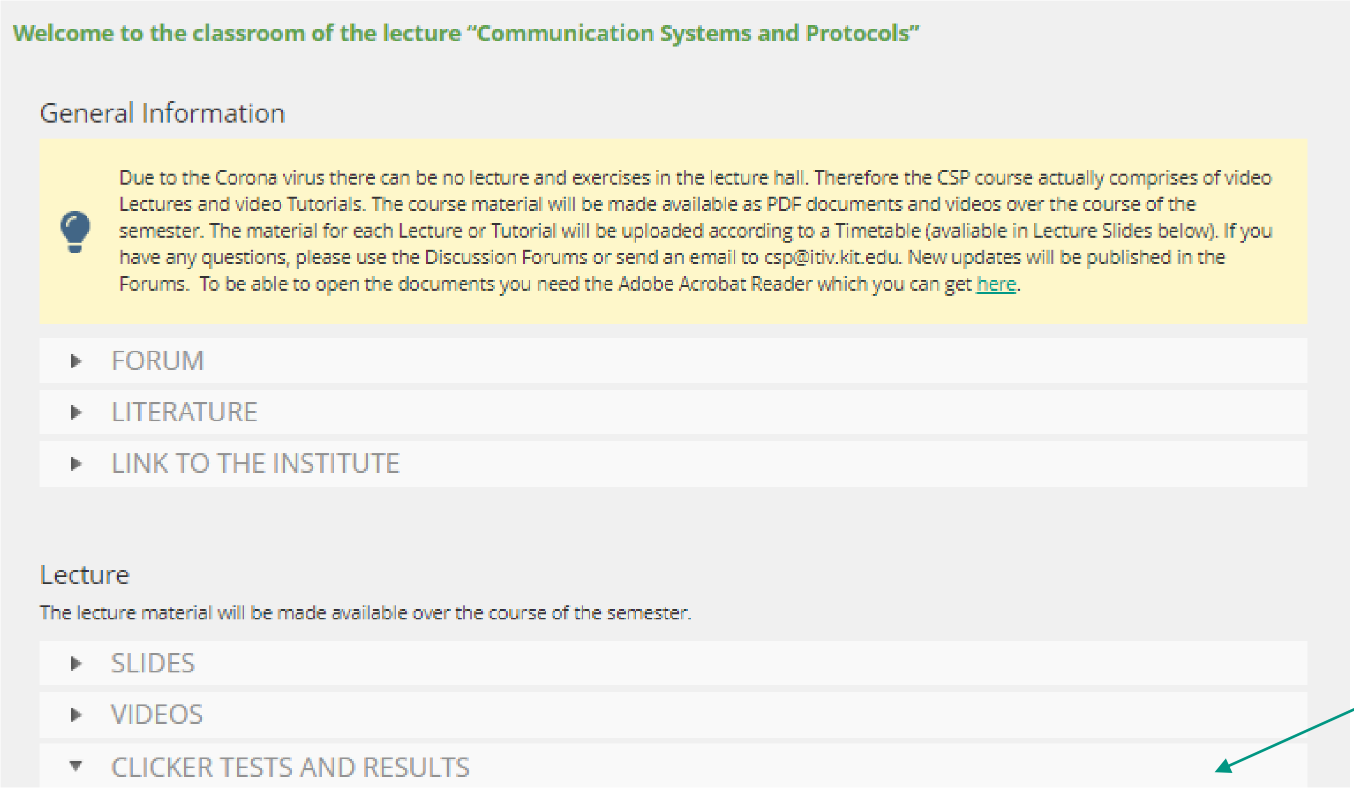
S=START condition

P=STOP condition

Addressing

- Problem:
  - Address of bus node can be hard coded by vendor
    → Only one device possible per design
- Solution:
  - Parts of the address can be selected by external wiring
    → DIP-Switches, solder bridge, ...
  - Address can be initialized via the bus
    → Programming using *General Call Address*

Participate in Clicker Test 7a



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

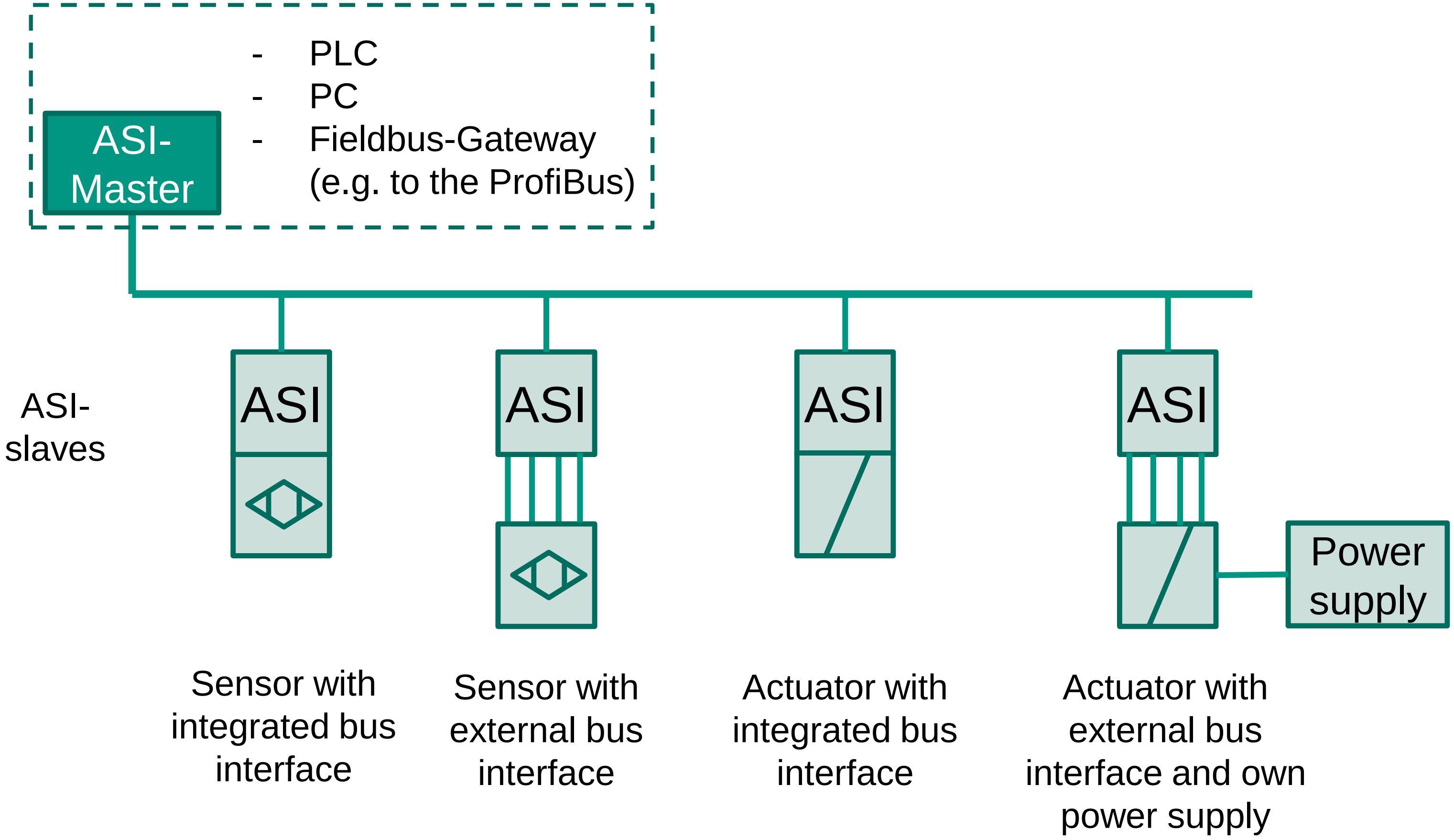
Actuator Sensor Interface (ASI)

The Actuator Sensor Interface (ASI)

- The goal is to directly connect sensors/actuators to the control via a bus
- Requirements:
  - Easy installation
  - Easy commissioning and maintenance
  - Low-priced (because of many bus connections)
- Data and energy should be transmitted on a two-conductor cable for all sensors and most actuators
- Simple and robust transmission procedure without limitations concerning net topology
- Compact and inexpensive bus connection

Concept of the ASI

- Master slave concept using a single master


ASI-Master
- PLC
- PC
- Fieldbus-Gateway (e.g. to the ProfiBus)
ASI-slaves
ASI
ASI
ASI
ASI
Power supply
Sensor with integrated bus interface
Sensor with external bus interface
Actuator with integrated bus interface
Actuator with external bus interface and own power supply

Transmission Medium

- As transmission medium, unshielded two-wire lines have been specified (→ max. 2A at 24V)

- Two variants:
  - 1.5mm² flat ribbon conductor (→budget-priced)
  - 1.5mm² ASI-specific flat cable (→ advantageous during installation)

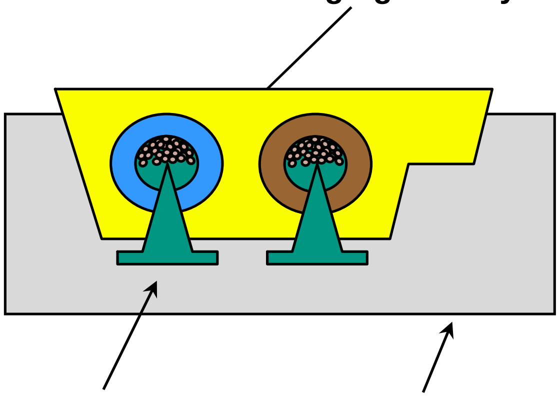
- Features:
  - Entire length of the line: 100m (→ extendable by a repeater to 300 m)
  - Max. 100mA current consumption per slave
  - Max. 1 master und 31 slaves (→ max. 124 sensors/actuators) per bus but several ASI-lines can be used in parallel

- Arbitrary topologies (star, tree, line) realizable by a coupling module

Coupling module

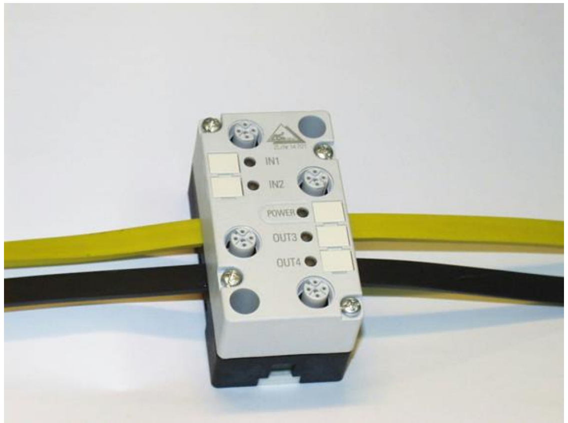
- In the ASI flat cable coupling module the contacting is realized in the form of a penetration technique
- Installation by clipping in the ASI cable without cutting or removing insulation
- Every coupling module can take 2 cables and connect them electrically
- Two types of user modules inside the lid:
  - Active user module: includes electronics for the slave connection → up to 4 conventional sensors/actuators can be connected
  - Passive user module: Without own electronics → for further branching of the ASI line

Mechanical coding through geometry



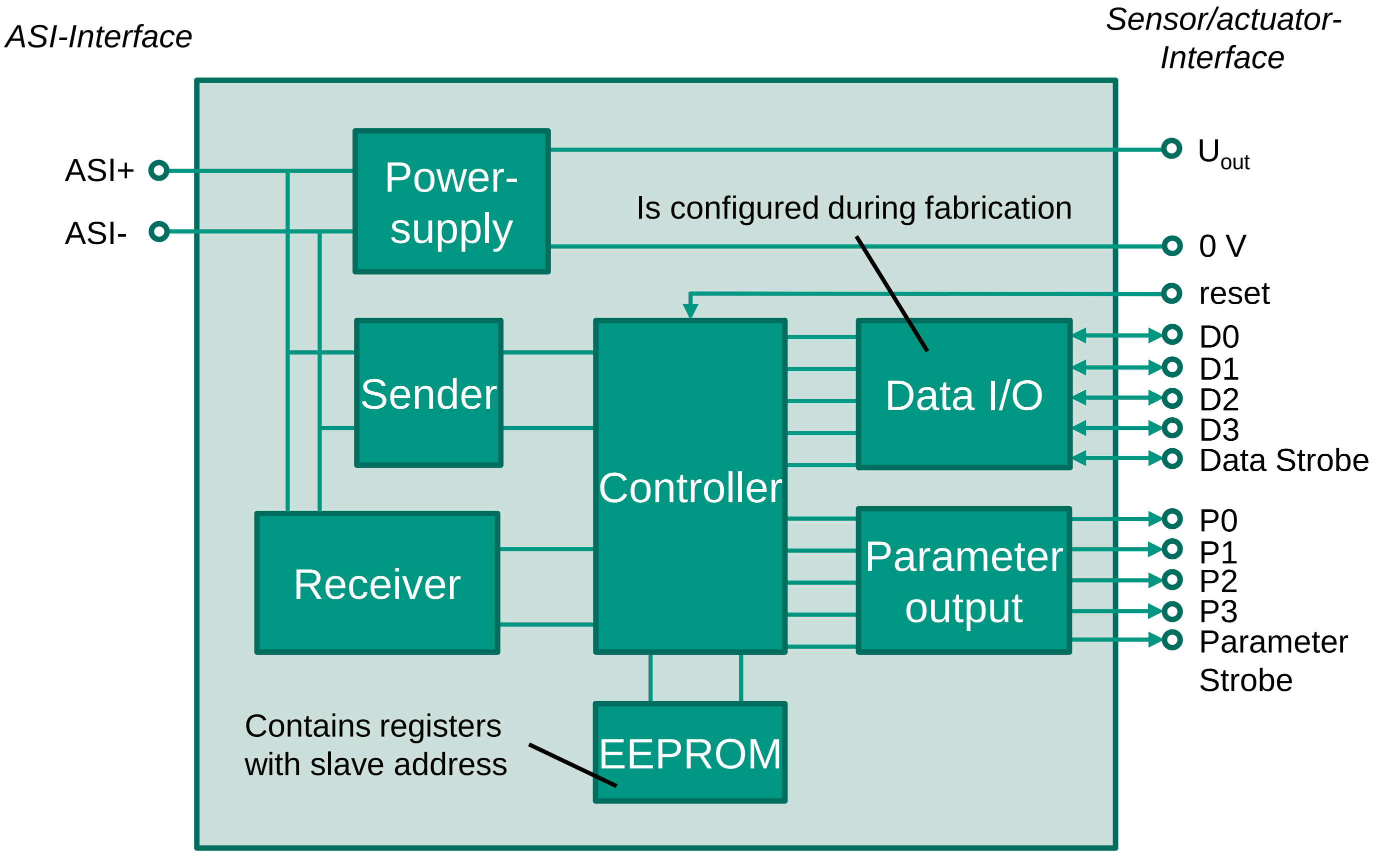
Penetrating contact

AS-Interface electro-mechanical interface



Structure of an ASI slave

- No complex software needed



Is configured during fabrication

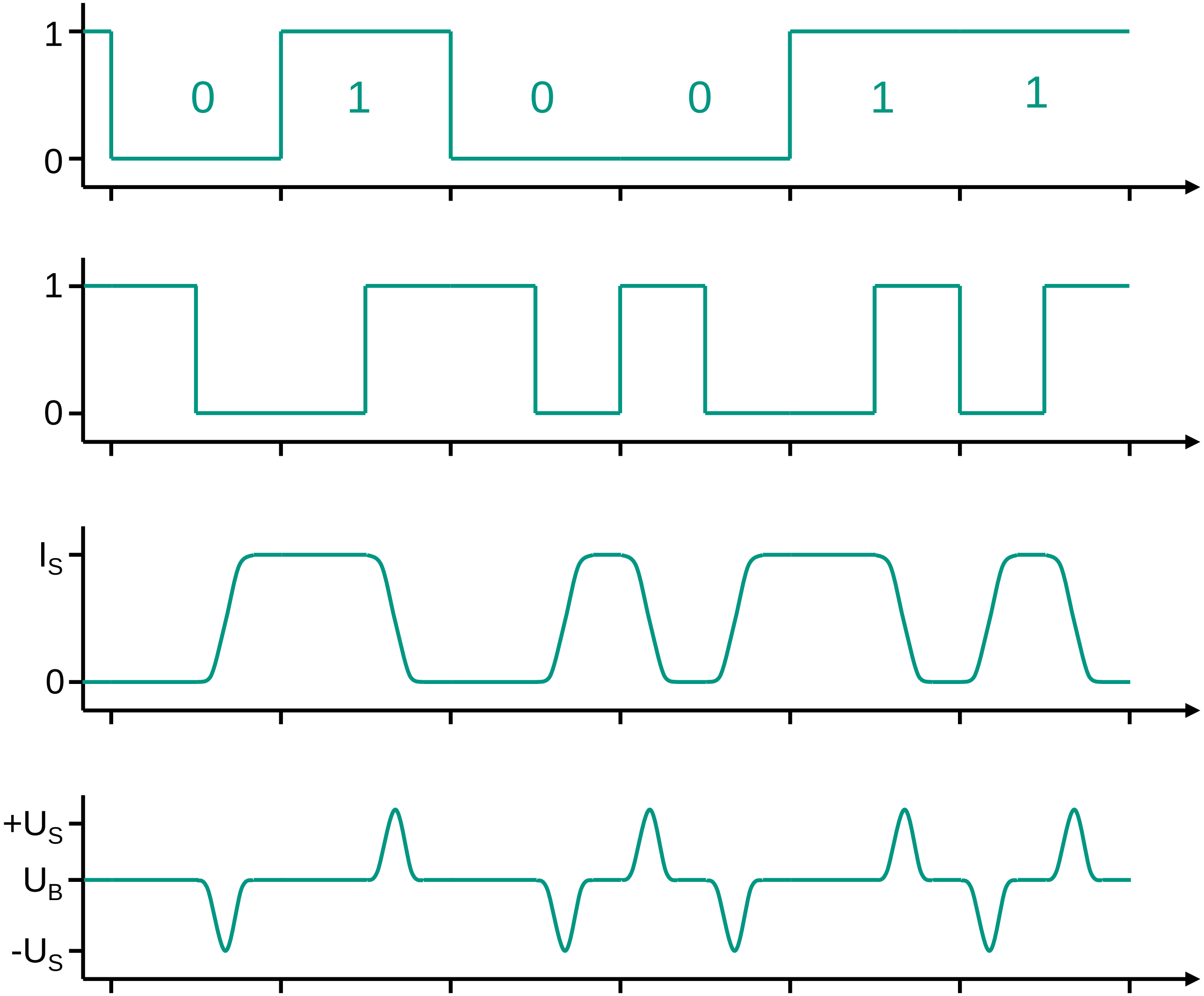
Contains registers with slave address

Modulation scheme

- The modulation scheme must satisfy the following requirements
  - Free of continuous current, since the data signal is modulated on top of the power supply
  - Master and slave must be able to easily generate the signal
  - Narrow-band since the attenuation of the cables rapidly increases with frequency

- Application of the Alternating Pulse Modulation (APM)
  1. Manchester coding of the raw data (→ phase change with every transition of the sender signal
  2. Generation of an appropriate sender current
  3. Sender current induces a voltage level within an inductivity that is existent only once in the system. The induced voltage can be larger than the supply voltage of the sender (→ neg. voltage at current increase, pos. voltage at descent
     - Low cut-off frequency when voltage pulses are formed as $\sin^2$-pulses
     - On specified lines, bit times of 6µs (→167 kBit/s) realizable

Examples for modulation



Sender bit string

Manchester coded bit string

Sender current

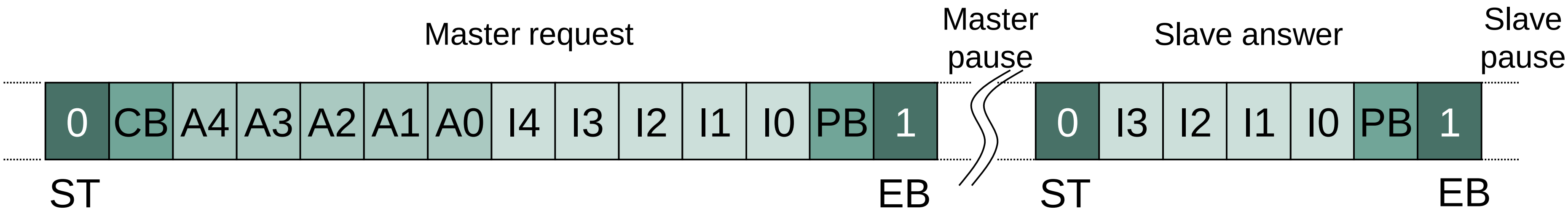
Voltage on signal line

Transmission method

- Bus access: Master-Slave with cyclic polling → real-time capable
- Master sends frame with address of a slave plus data (14 bit with 6µs each)
- Slave answers (7 bit with 6µs each) within a given time
  - Master pause: in general 3, max. 10 bit times (after that, the master assumes that no answer will arrive any more and sends the next request)
  - Slave pause: 1 bit time (6 µs)
- Only 5 bit for information within the frame to keep frames short

→ 14+3+7+1 = 25 bit per cycle → 150 µs per cycle

→ 5 ms overall cycle time with 31 slaves (sufficient for PLC controls)



ST:	Start bit	I4..I0:	Information bits
CB:	control bit	PB:	Parity bit
A4..A0:	Slave address	EB:	End bit

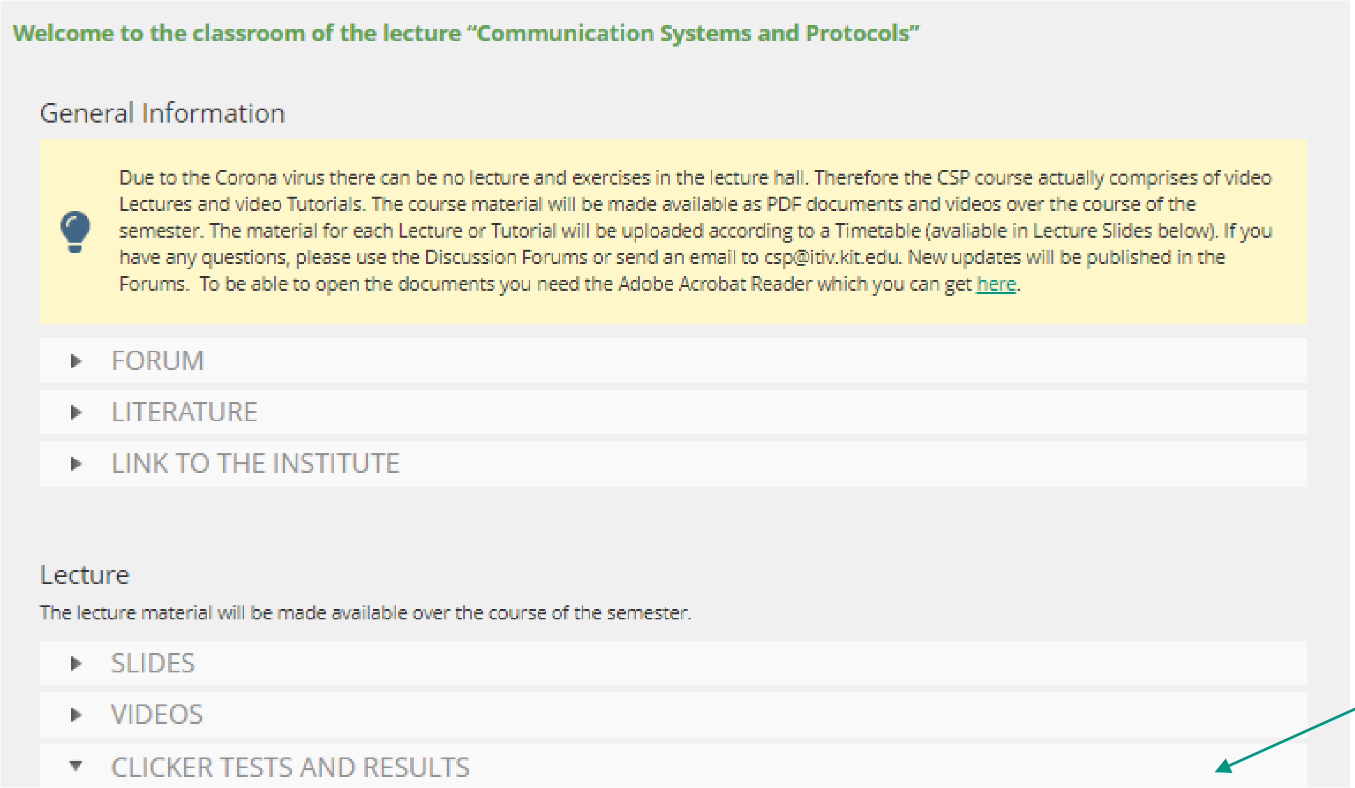
Data Integrity (I)

- Integrity checking uses different criteria than other so far considered bus systems
- Due to the shortness of the frames the checksum overhead would be too large
- ASI tests the signal waveform on physical layer (16 times sampling during one bit time)
- The following rule set is tested by the slave module:
  - Start and stop bits: first pulse must be negative, last pulse must be positive
  - Successive pulses must have different polarity
  - Between two pulses of a frame only one pulse is allowed to be missing
  - No pulse during pause time
  - Even parity

Data Integrity (II)

- High safety:
  - All one- and twofold pulse errors are recognized
  - 99.9999% of all three- and fourfold pulse errors are recognized
  - Parity check takes effect from threefold pulse error on
- Theoretical estimation:
  - Having a bit error rate of 100 errors/s only every 10 years a erroneous frame is not recognized
- Erroneous frames are repeated but do hardly increase the overall cycle time due to their shortness

Participate in Clicker Test 7b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruhe Institute of Technology

Institute for Information Processing Technologies

Communication Systems and Protocols

Session 22: FireWire

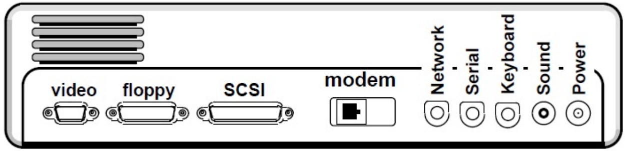
IEEE 1394: Standard for a High-Performance Serial Bus

- Brief History
  - Apple led the initiative in 1987
  - The standard is the result of the collaboration of several companies including Apple, IBM and Sony.
  - Apple called the interface **FireWire**. It is also known by the brands **i.LINK** (Sony), and **Lynx** (Texas Instruments).
  - Approved as IEEE standard in 1995: IEEE 1394-1995
  - Superseded by version in 2008: IEEE 1394-2008

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Goals

- Low cost, high performance ergonomic system interconnect
- Compatible architecture with other IEEE busses
- Peer-to-peer read/write memory model for asynchronous services
- Provide Isochronous service



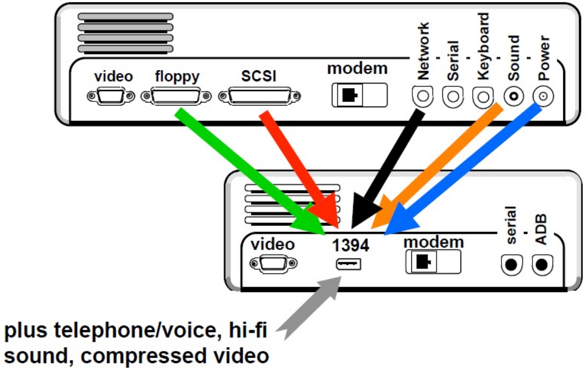


Fig Source : https://grouper.ieee.org/groups/802/802_tutorials/04-July/NewTechIntroTo1394.pdf

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

IEEE Std 1394-2008

- Three primary applications have driven the design and architecture of the serial bus:
  - **an alternate for a parallel backplane bus**
  - **a low-cost peripheral bus and**
  - **a bus bridge between architecturally compatible 32-bit buses.**

- Version:
  - The description provided here is based on IEEE Std 1394-2008

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Asyncronous and Isochronous transfers

- This standard provides specifications for a high-speed serial bus that supports both **asynchronous and isochronous communication**

- The asynchronous (*asyn* = any, *chronous* = time) data transfer service provides a packet delivery protocol for variable-length packets to an explicit address and return of an acknowledgment.

- The isochronous (*iso* = same, *chronous* = time) data transfer service provides a broadcast packet delivery protocol for variable length packets that are transferred at regular intervals.

  - Asynchronous transport
    - Guaranteed delivery
    - Reliability more important than timing
    - Retries are OK

  - Isochronous transport
    - Guaranteed timing
    - Late data is useless
    - Never retry

Source : https://grouper.ieee.org/groups/802/802_tutorials/04-July/NewTechIntroTo1394.pdf

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

General Features

- Intended to provide a low-cost interconnect between cards on the same backplane, cards on other backplanes, and external peripherals.

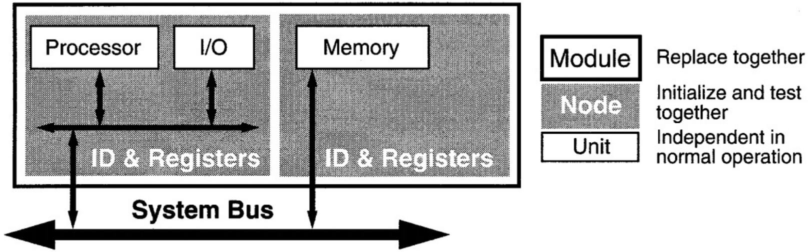
- This standard follows the command and status register (CSR) architecture to be compatible with other IEEE busses.
  - The CSR architecture defines the serial bus addressing structure

- The cable environment supports multiple data rates of around 100 Mbit/s, 200 Mbit/s, and 400 Mbit/s. (The lowest speed is known as the “base rate.”)

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Node and Module Architecture

- The serial bus architecture is defined in terms of **nodes**.
- A node is a logical entity with a unique address. It can be independently configured and identified.
- More than one node may reside on a single module, and more than one unit may reside in a single node.
- Note that modules are a physical packaging concept and nodes are a logical addressing concept.
- The address space provided by a node can be directly mapped to one or more units. A unit is a logical entity, such as a disk controller, which corresponds to unique I/O driver software

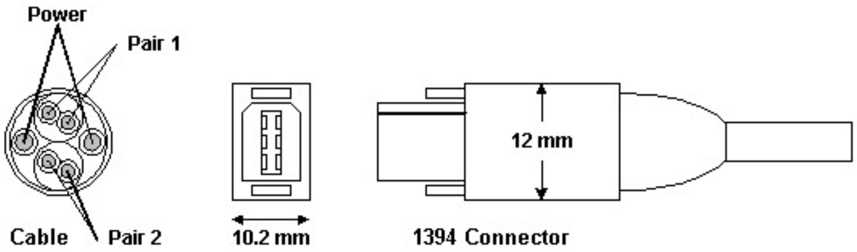


Module Architecture

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Port Interface

- A port consists of two twisted pair interfaces, TPA/TPA* and TPB/TPB*, and an optional power distribution pair, VP/VG.
- A node may have from one to sixteen of such ports.
- Each port has associated circuitry that provides separate signals for arbitration or for packet data reception and transmission

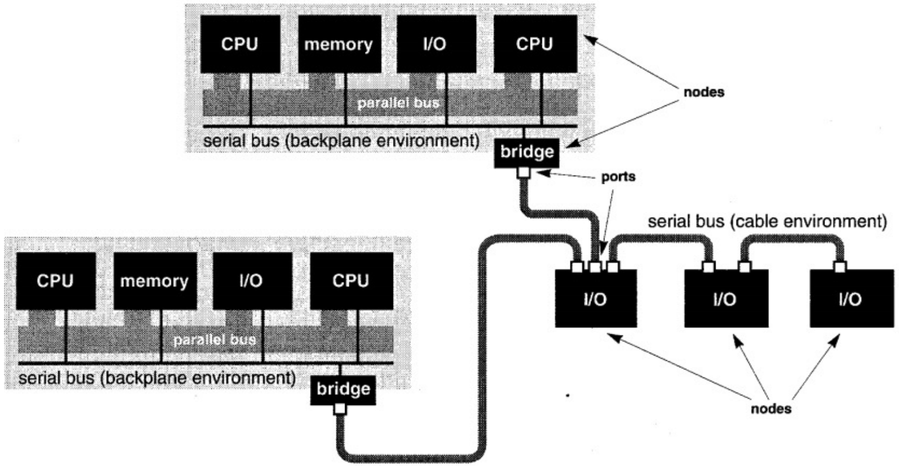


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Topology

- The physical topology of the serial bus is divided into two “environments”
  - **Backplane environment**
  - **Cable environment**

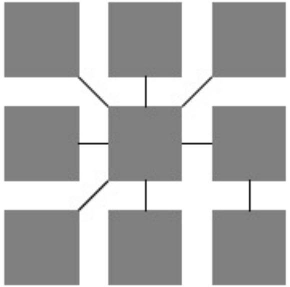
- Interconnected nodes may reside in either environment without restriction.

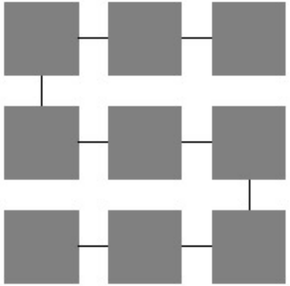

CPU
memory
I/O
CPU
parallel bus
serial bus (backplane environment)
bridge
nodes
ports
serial bus (cable environment)
CPU
memory
I/O
CPU
parallel bus
serial bus (backplane environment)
bridge
I/O
I/O
I/O
nodes

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable environment

- The physical topology for the cable environment is a noncyclic network with finite branches and extent.
- **"Noncyclic" means that closed loops are unsupported.**
- The medium consists of two conductor pairs for signals and one pair for power and ground that connect together ports on different nodes.
- Each port consists of terminators, transceivers, and simple logic. The cable and ports act as bus repeaters between the nodes to simulate a single logical bus





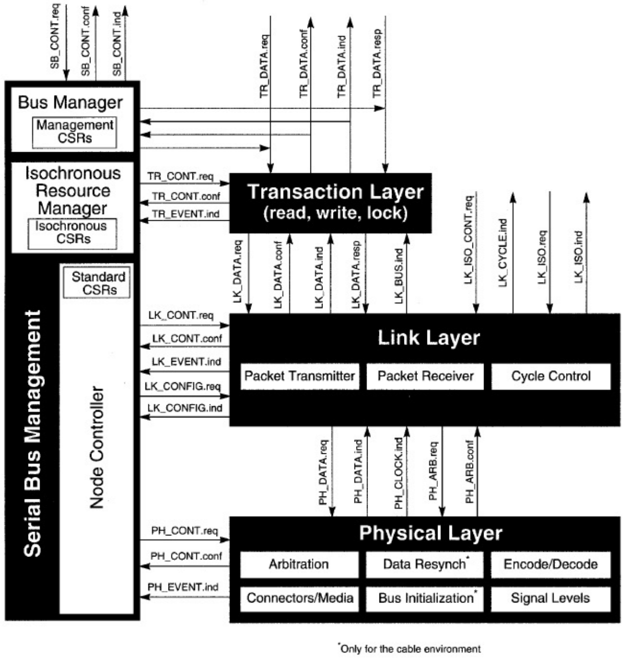
Backplane environment

- The physical topology of the backplane environment is a multidrop bus.
- The media consists of two single ended conductors running the length of the backplane.
- Connectors distributed along the bus allow nodes to “plug into” the bus.

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Serial Bus Protocol (SBP) Architecture

- Consists of a set of three stacked layers
  - **Transaction layer**
  - **Link layer**
  - **Physical Layer (PHY).**
- Serial Bus Management (SBM): Monitors and controls physical, link, and transaction layers.



SBP Stack

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Transaction layer

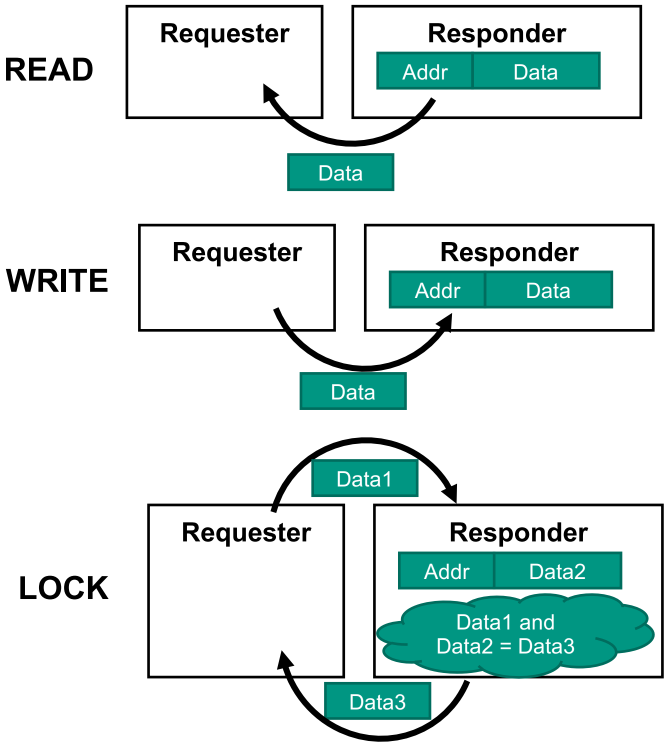
- Defines a complete request-response protocol to perform the bus transactions required to support the command and status register (CSR) architecture
  - The CSR architecture defines the serial bus addressing structure, which allows almost 2^{16} nodes.

- Transactions are multithreaded, in that more than one transaction can be started by a requester before the corresponding response is returned. These are called a split-response transactions.

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Transaction layer

- Data are transferred between nodes on the serial bus by three different transaction types:
  - **Read**: Data at a particular address within a responder are transferred back to a requester.
  - **Write:** Data are transferred from a requester to an address within one or more responders.
  - **Lock:** Data are transferred from a requester to a responder, processed with data at a particular address within the responder, and then transferred back to the requester.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Link Layer

- Provides an acknowledged datagram service to the transaction layer
- Provides a **half-duplex** data packet delivery service
  - A one-way data transfer with confirmation of request
- It provides addressing, data checking, and data framing for packet transmission and reception
- Packets which are used here are called **primary packets and acknowledge packets**

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Link Layer

- The process of delivering a single packet is called a "**subaction**," and there are two types:
  - **An asynchronous subaction**: a variable amount of data and several bytes of transaction layer information are transferred to an explicit address and an acknowledge is returned.
  - **An isochronous subaction or "channel":** a variable amount of data is transferred on regular intervals with simplified addressing and no acknowledge.

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Physical Layer (PHY)

- The physical layer has three major functions:

  1. It translates the logical symbols used by the link layer into electrical signals on the different serial bus media.
  2. It guarantees that only one node at a time is sending data by providing an arbitration service.
  3. It defines the mechanical interfaces for the serial bus.

- The cable and backplane environments have different PHYs
- Nonetheless, both PHYs share two fundamental concepts:
  - **Data-Strobe (DS) encoding for data bits, and**
  - **A simple method for ensuring fair access to the bus.**

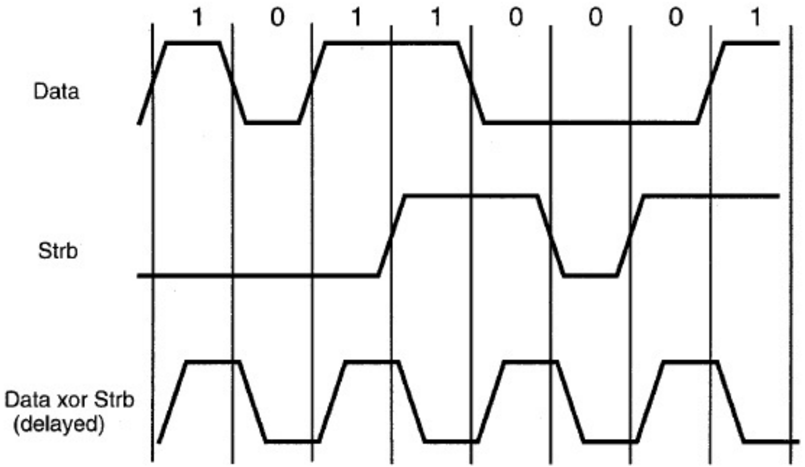
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Data Transmission: Data-Strobe (DS) encoding

- During packet transmission, there is only a single node transmitting on the bus, so the entire media can operate in a half-duplex mode using two signals: **Data and Strb.**

- NRZ data are transmitted on Data and is accompanied by the Strb signal, which changes state whenever two consecutive NRZ data bits are the same, ensuring that a transition occurs on either Data or Strb for each data bit.

- A clock that transitions each bit period can be derived from the exclusive-or of Data with Strb as shown in Figure



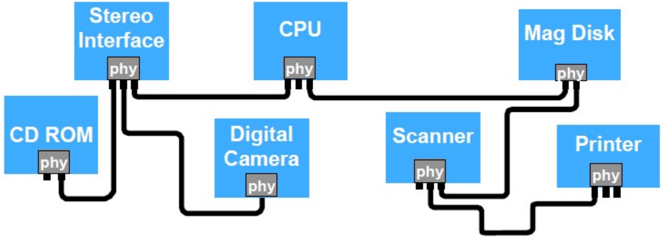
DS Encoding

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable environment

- Here we focus on the arbitration in the cable environment.

- First describe the cable environment in detail:
  - The cable environment is a network of nodes connected by point-to-point links called physical connections.
  - The physical connection consists of a port on the PHY of each node and the cable between them.
  - The primary restriction is that nodes have to be connected together as an acyclic graph (no loops).



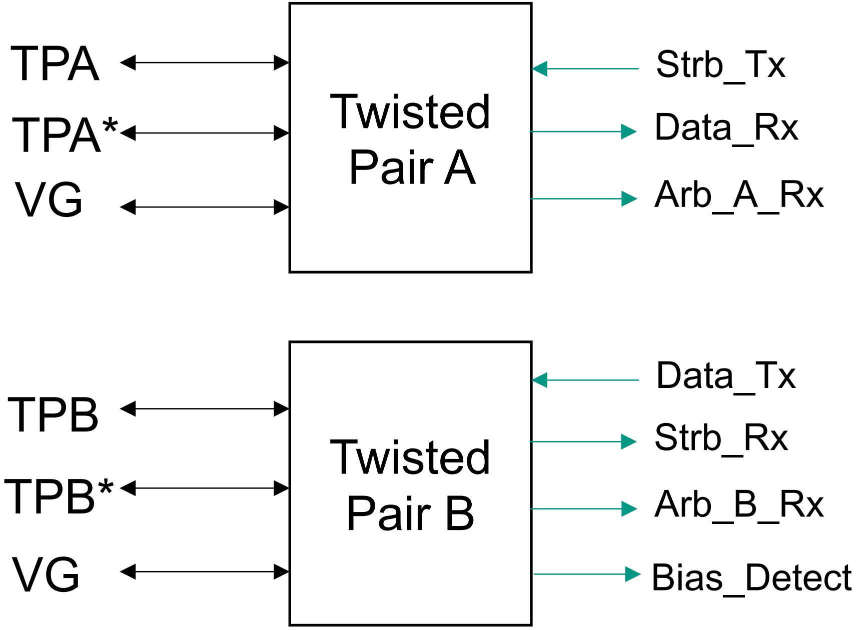
Cable Environment

Fig Source : https://grouper.ieee.org/groups/802/802_tutorials/04-July/NewTechIntroTo1394.pdf

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable Physical Layer (PHY): Port Interface

- A port consists of two twisted pair interfaces, TPA/TPA* and TPB/TPB*, and an optional power distribution pair, VP/VG.
- Each port has associated circuitry that provides separate signals for arbitration or for packet data reception and transmission
- **Intrestingly Strb_Tx, Data_Tx signals are used to generate the arbitration signals Arb_A_Tx and Arb_B_Tx.**



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable Physical Layer (PHY)

- The cable PHY logically consists of four major components
  1. **Ports**, which provide the cable media interface.
  2. **Arbitration logic,** which provides access to the bus.
  3. **Resynchronizer,** which takes received Data-Strobe(DS) encoded data bits and generates data bits synchronized to a local clock.
  4. **Encoder,** which takes either the data being transmitted by the node (if the node has won arbitration) or the data received by the resynchronizer and encodes it in DS format.

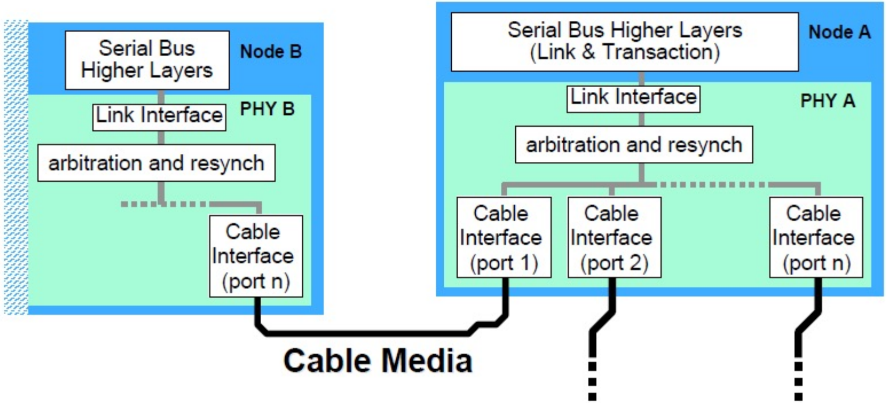
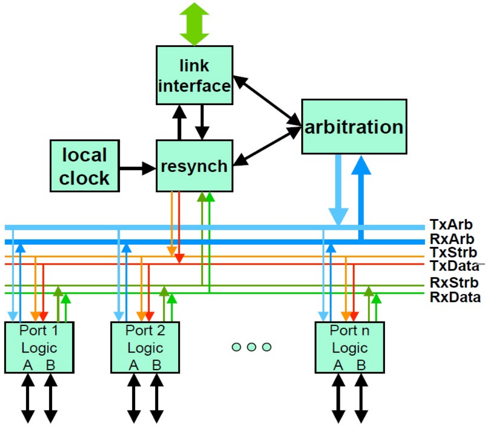


Fig Source : https://grouper.ieee.org/groups/802/802_tutorials/04-July/NewTechIntroTo1394.pdf



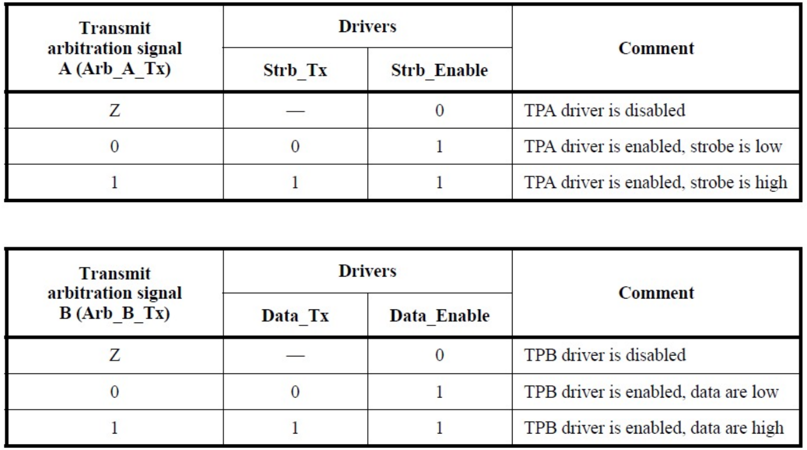
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable Physical Layer (PHY): Internal Signals

- During cable arbitration, the arbitration signals (Arb_A_Tx and Arb_B_Tx) take three different values (“1”, “0”, and “Z”) when TPA/TPB meet certain min and max voltage requirements

- The NRZ data are transmitted and received as Data_Tx and Data_Rx and is accompanied by a strobe signal, Strb_Tx and Strb_Rx.

- Example:
  - The logical values for arbitration Arb_A_Tx and Arb_B_Tx are generated by the drivers according to the table.



Transmit arbitration signal A (Arb_A_Tx)	Drivers		Comment
	Strb_Tx	Strb_Enable	
Z	—	0	TPA driver is disabled
0	0	1	TPA driver is enabled, strobe is low
1	1	1	TPA driver is enabled, strobe is high

Transmit arbitration signal B (Arb_B_Tx)	Drivers		Comment
	Data_Tx	Data_Enable	
Z	—	0	TPB driver is disabled
0	0	1	TPB driver is enabled, data are low
1	1	1	TPB driver is enabled, data are high

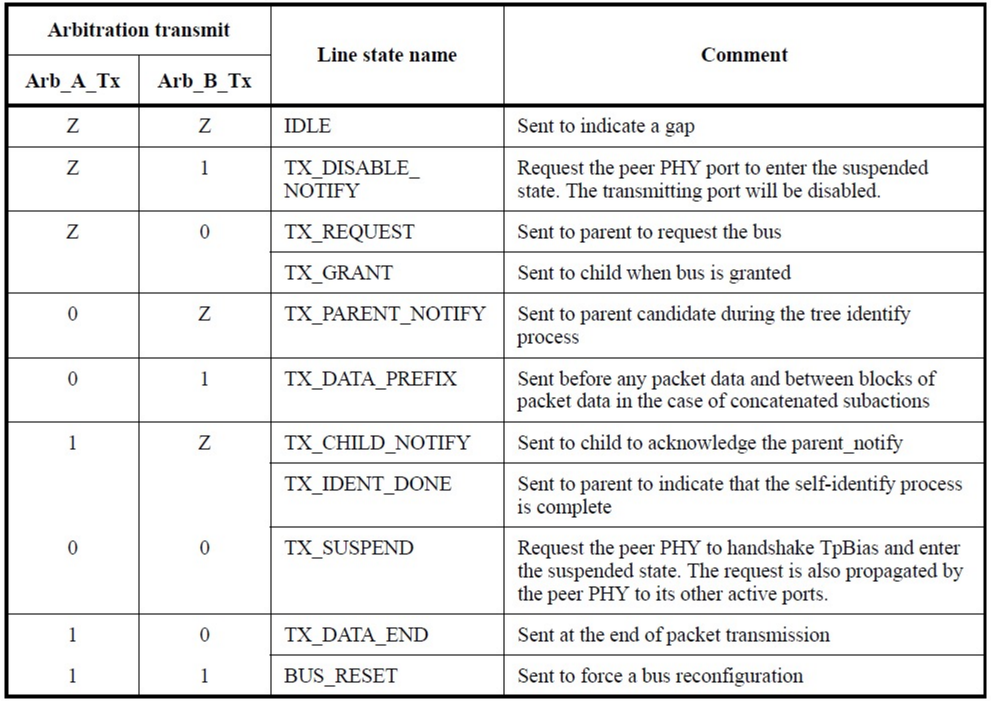
Arbitration signal generation rules

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Cable Physical Layer (PHY): Line states

- The combination of Arb_A_Tx and Arb_B_Tx yield different **Line States** having different meaning.

- Example: Some of the Line states that are transmitted during arbitration is shown here



Arbitration transmit		Line state name	Comment
Arb_A_Tx	Arb_B_Tx		
Z	Z	IDLE	Sent to indicate a gap
Z	1	TX_DISABLE_ NOTIFY	Request the peer PHY port to enter the suspended state. The transmitting port will be disabled.
Z	0	TX_REQUEST	Sent to parent to request the bus
		TX_GRANT	Sent to child when bus is granted
0	Z	TX_PARENT_NOTIFY	Sent to parent candidate during the tree identify process
0	1	TX_DATA_PREFIX	Sent before any packet data and between blocks of packet data in the case of concatenated subactions
1	Z	TX_CHILD_NOTIFY	Sent to child to acknowledge the parent_notify
		TX_IDENT_DONE	Sent to parent to indicate that the self-identify process is complete
0	0	TX_SUSPEND	Request the peer PHY to handshake TpBias and enter the suspended state. The request is also propagated by the peer PHY to its other active ports.
1	0	TX_DATA_END	Sent at the end of packet transmission
1	1	BUS_RESET	Sent to force a bus reconfiguration

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

System Configuration

- The cable arbitration takes advantage of the point-to-point nature of the cable environment by having each node handshake with its immediate neighbors to determine ownership of the media.

- **Prior to normale operation the system has to be configured.**

- System configuration is done in three phases
  1. **Bus initialize**
  2. **Tree identify**
  3. **Self-identify.**

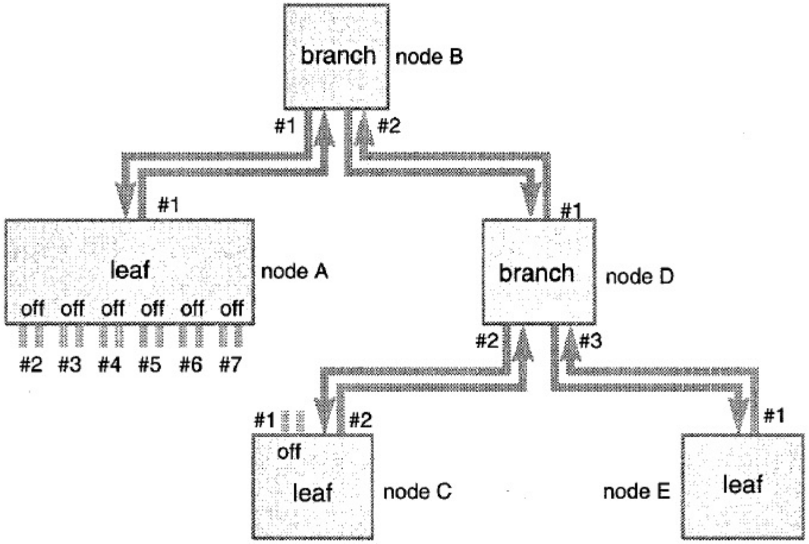
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

1. Bus Initialize

- Whenever a node joins the bus, a bus reset signal forces all nodes into a special state that clears all topology information.

- After the bus initialization process, the only information known to a node is whether it is a
  - **Branch:** more than one directly connected neighbor or
  - **Leaf:** only a single neighbor or
  - **Isolated:** unconnected

- The eventual root may be either a branch or a leaf!



After Bus Initialization

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

- The tree identify process translates the general network topology into a tree, where one node is designated a root and all of the physical connections have a direction associated with them pointing towards the root node.

- The direction is set by labeling each connected port as a
  - **Parent:** connected to a node closer to the root or
  - **Child:** port connected to a node further from the root

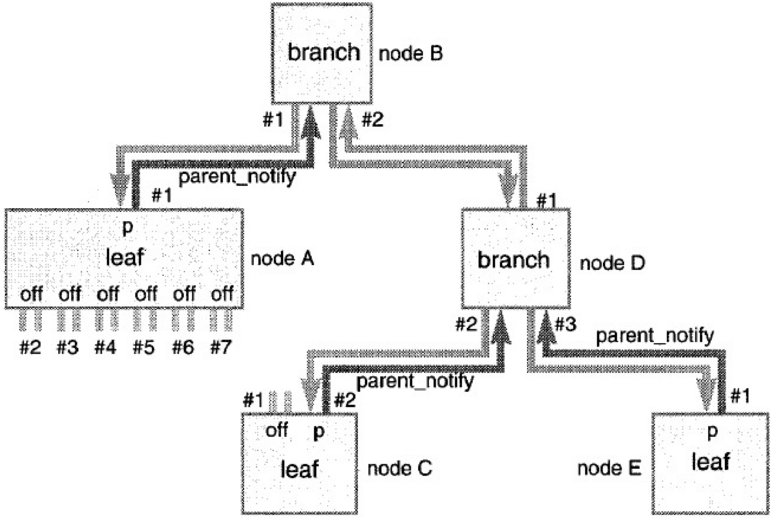
- Any unconnected ports are labeled “off” and do not participate in further arbitration processes.

- Any loop in the topology is detected by a timeout in this process.

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

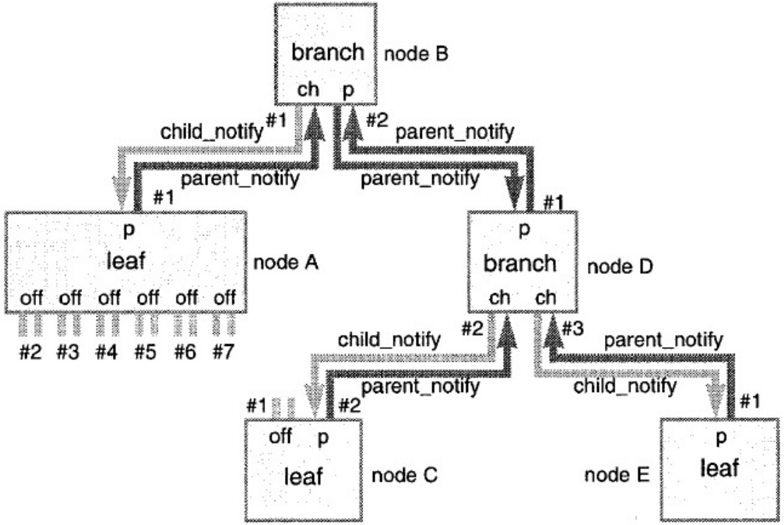
- The first step is for all leaf nodes to notify their probable parents. This is done by sending a *parent_notify* packet (signal)
- In this example, nodes A, C, and E send a *parent_notify* to their single connected port.
- This is the start of the parent-child handshake process.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

- The nodes internally recognize the *parent_notify* signals and mark the ports receiving them as child ports.
- If these nodes have one unidentified port, they send *parent_notify* up to their probable parents.
- At the same time the nodes send down *child_notify* signals to their child ports.
- Example here: Nodes B and D have one port remaining that is connected but not yet identified as child or parent. Therefore a *parent_notify* signal is sent.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

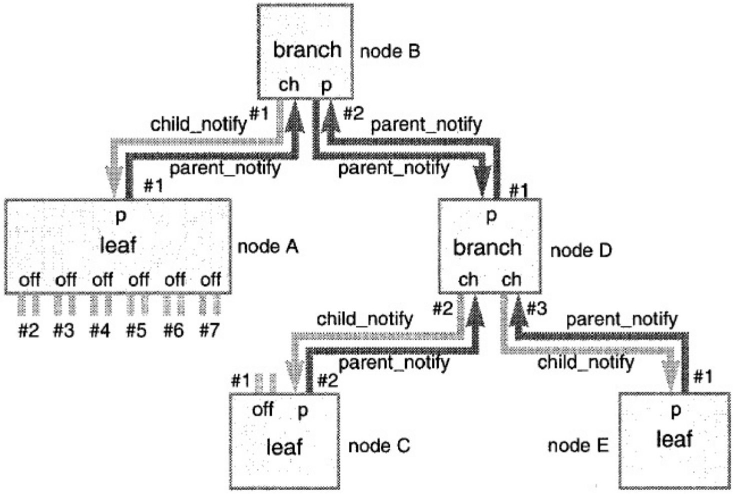
- When the leaf nodes receive the *child_notify* signal, that port is assigned as their parent port. Once assigned, their *parent_notify* signal is withdrawn.

- **Nodes having only child ports are assigned as the root.**

- It can occur that multiple nodes are receiving only *parent_notify* signals. This leads to a process called “root contention”

- Example:
  - Both nodes B and D discover that they are receiving the *parent_notify* port
  - Since one of the two nodes has to become the parent of the other, this collision of intentions starts a process called “root contention”.

- This is resolved by withdrawing the *parent_notify* signals. A timer is assigned on each node with a random duration and the signals are resent after the time elapses.

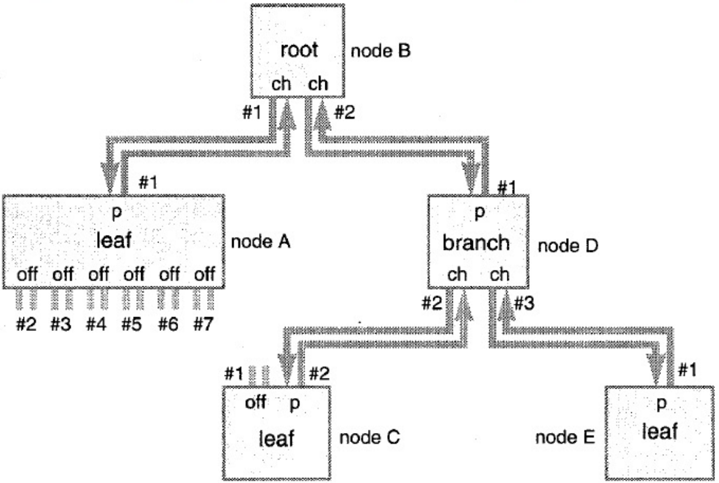


Start of root contention

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

2. Tree identify process

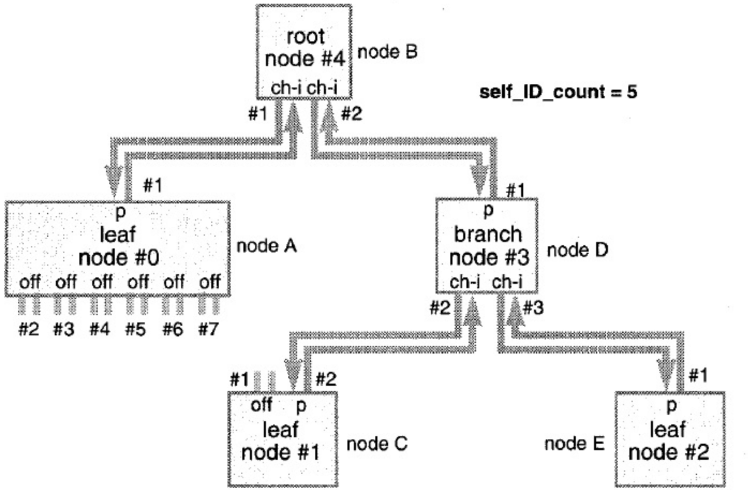
- When a node has all ports labeled as children, it takes the root function for itself.
- In this example:
  - Node D resends the signals first due to root contention
  - Node B has all ports as children and is assigned as the root.
- Note that the selection of the root node is not topology dependent. It is completely acceptable that the root node also be a leaf.
- The node that waits the longest after the bus reset to start participating in the tree identify process becomes the root.
- A particular node can be forced to wait a longer time by setting a *force_root* bit



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

3. Self-Identify Process

- The next step is to give each node a unique physical_ID
- The self-identify process uses a deterministic selection process, where the root node passes control of the media to the node attached to its lowest numbered connected port first. The child nodes then passes control in a recursive manner
- The first leaf node to receive control, during its self-identify process selects 0. The next node which receives control selects 1 and so on
- The root then passes control to the next lowest numbered port. When the nodes attached to all the ports of the root are finished, the root itself does a self-identify process.
- **Note that each port of the node is individually numbered. There is no particular order to the numbering, it is just a way to give each port a unique label**

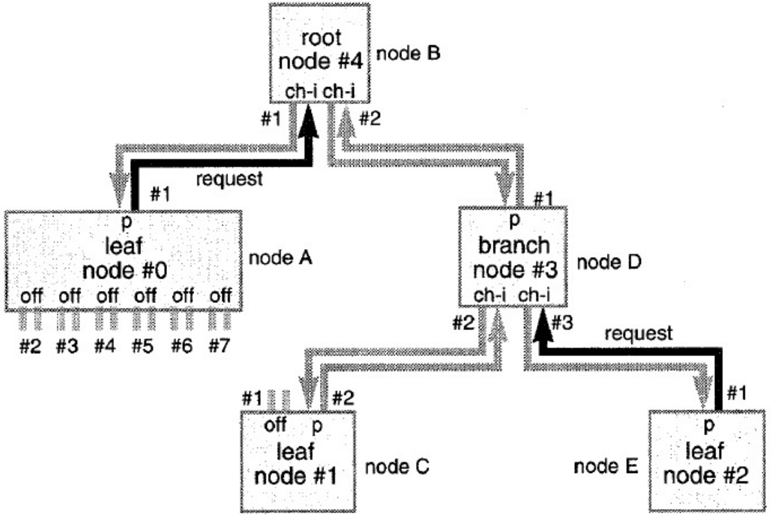


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Arbitration

- Once the self-identify process is complete, nodes can use the normal arbitration method to send packets

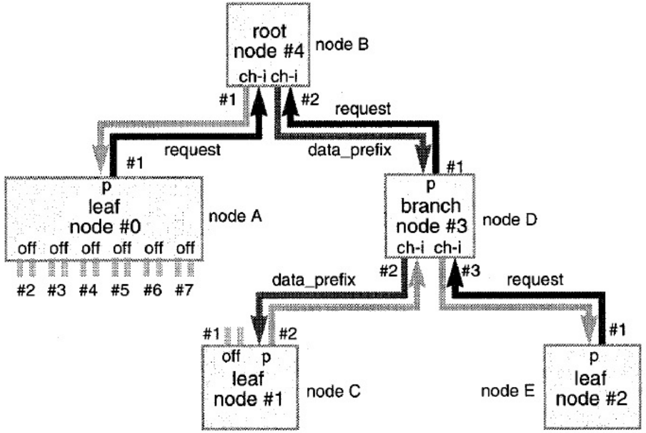
- Example:
    - Node A and E begin arbitrating at the same time by sending **request** to their parents

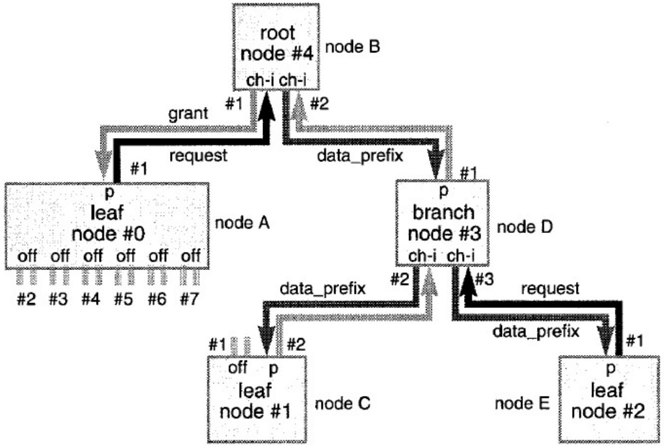


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Normal Arbitration

- Node D forwards the request and denies access to its other children by sending a *data_prefix*.
- At the same time, node B receives request from node A and denies access to its other children.
- Since node B is root, it does not forward the request further and grants access to Node A.





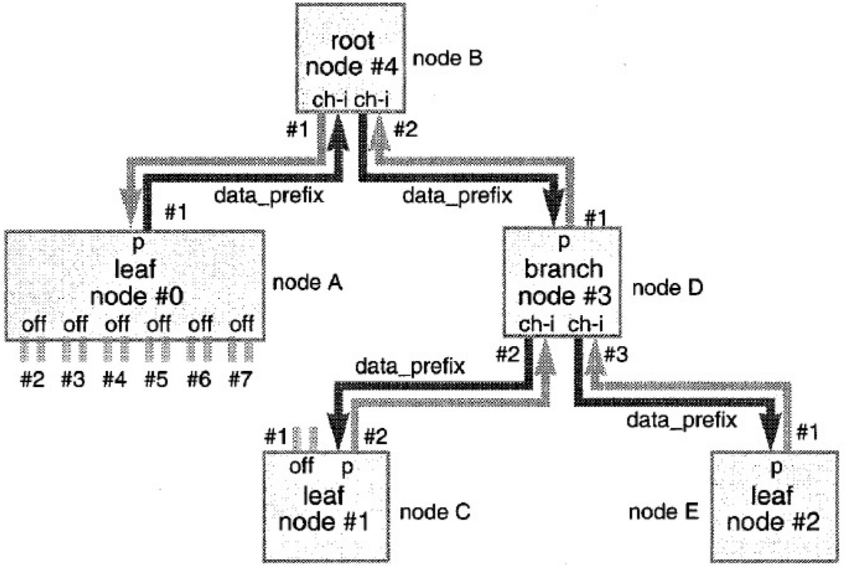
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Normal Arbitration

- The granted node sends a *data_prefix* signal to all other nodes to warn other nodes that data is about to be sent.

- When the parent (in the example it is the root) receives this *data_prefix* signal, it stops sending the *grant*.

- At this point, the physical connections between all the nodes are now in the same state and pointed away from the node that won the arbitration.



Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Fair Arbitration

- The normal cable and backplane arbitration methods guarantee that **only one node will be transmitting at the end of the arbitration period.**

- These methods only provide a strict priority access; the node with the highest natural priority will always win.
  - Example: For cable environment, closest node to the root always wins

- The normal **asynchronous arbitration** for the serial bus adds a simple scheme that splits the access opportunities evenly among competing nodes.
  - A **fairness protocol** is used which is based on the concept of a **fairness interval.**

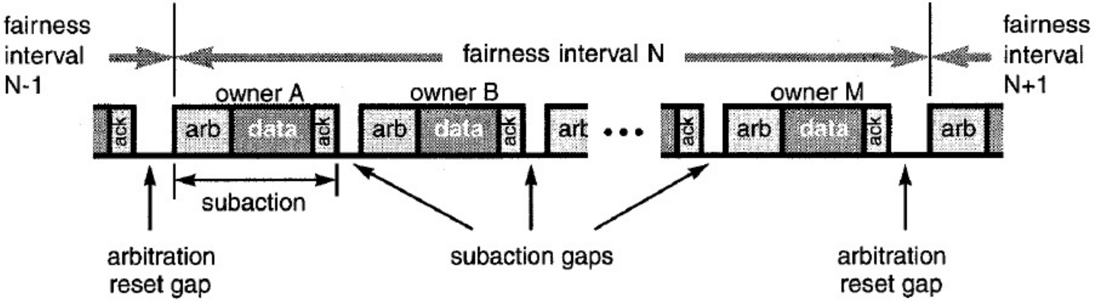
Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Fairness Protocol

- A fairness interval consists of one or more periods of bus activity separated by short idle periods called **subaction gaps** and is followed by a longer idle period known as an **arbitration reset gap**.

- At the end of each subaction gap, **bus arbitration** is used to determine the next node to transmit an asynchronous packet.

- When using fair arbitration, an active node can initiate sending an asynchronous packet exactly once in each fairness interval.

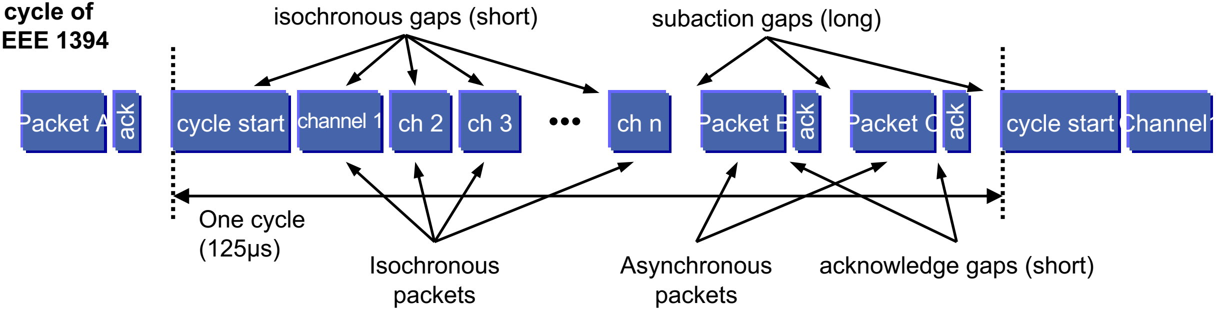


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Isochronous transfers

- The cycle master (root) controls the transfer in frames of 125µs
- An asynchronous packet can have a maximum length of 62µs
- Addressing is done using channel numbers that have been negotiated at the beginning
- Nodes that want to send isochronous packets apply directly for the bus without waiting for the subaction gap
  - Thus isochronous transfers have higher priority then asynchronous transfers
  - If no other node wants to start an isochronous transfer there will arise a subaction gap so that other nodes can apply for the bus

Bus cycle of the IEEE 1394



isochronous gaps (short)

subaction gaps (long)

Packet A | ack | cycle start | channel 1 | ch 2 | ch 3 | ••• | ch n | Packet B | ack | Packet C | ack | cycle start | Channel1

One cycle (125µs)

Isochronous packets

Asynchronous packets

acknowledge gaps (short)

Source : Figure Q.16, FireWire™ Reference Tutorial (An Informational Guide), January 22, 2010, 1394 Trade Association

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Thank you for your attention!

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4659233

Institut für Technik der
Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Universal Serial Bus (USB)

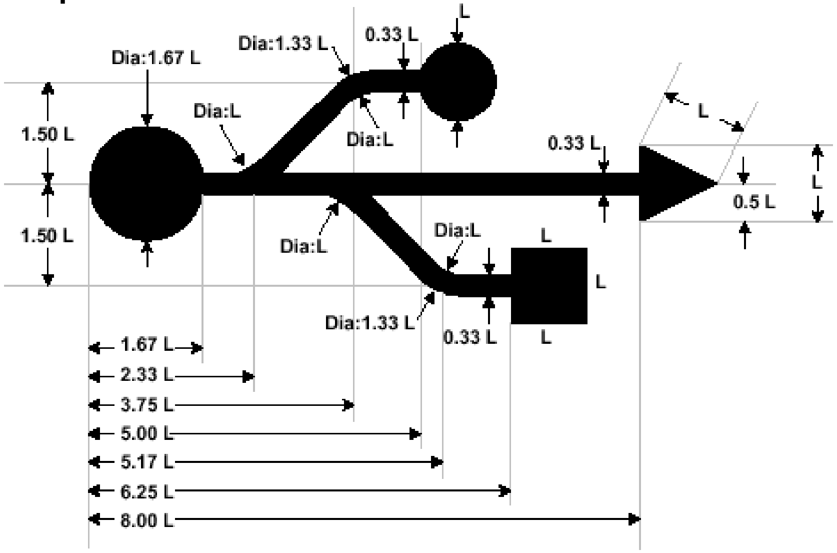
Universal Serial Bus (USB)

- Serial Bus
- NRZ-I Encoding
- Single Master / Multi Slave
- Hot-Plug Capability

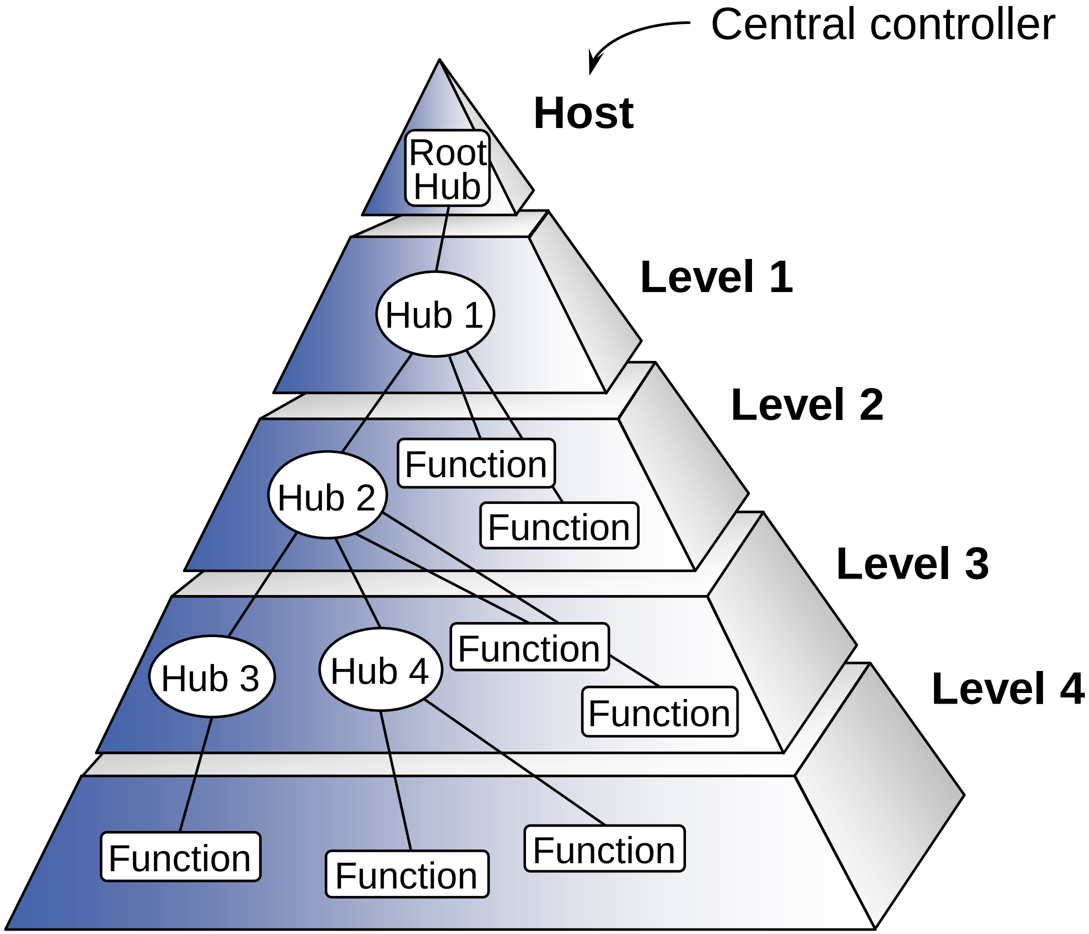
Universal Serial Bus (USB)

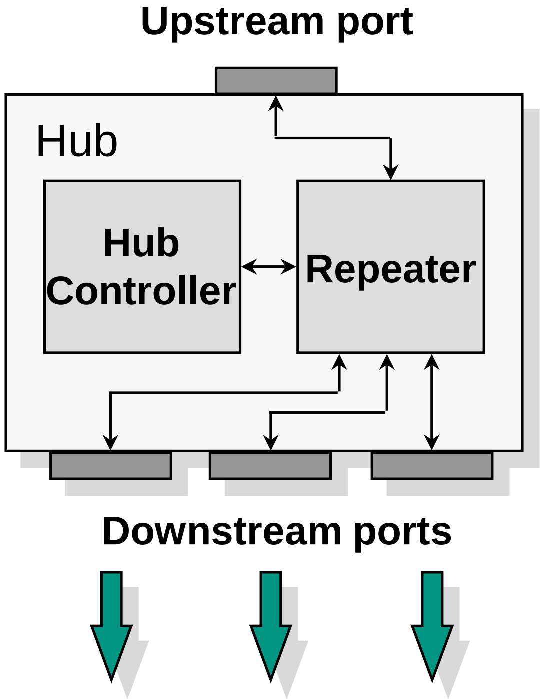
- Bus system used to connect peripheral devices like mouse, keyboard and printer to the PC
- Serial bus
  - Hot-plugging, auto configuration
  - At most 127 attached devices possible
  - Tree topology, branching is done in hubs
  - Point-to-point connections between individual devices
  - Central control through one single host (PC)
  - Polling is done by host, no arbitration required
  - Different speeds:
    1.5 Mbit/s, 12 Mbit/s,
    480 Mbit/s (USB 2.0),
    5000 Mbit/s (USB 3.0)
- Electrical characteristics
  - Differential signaling
  - Integrated power supply

All dimensions are ± 5%



Topology

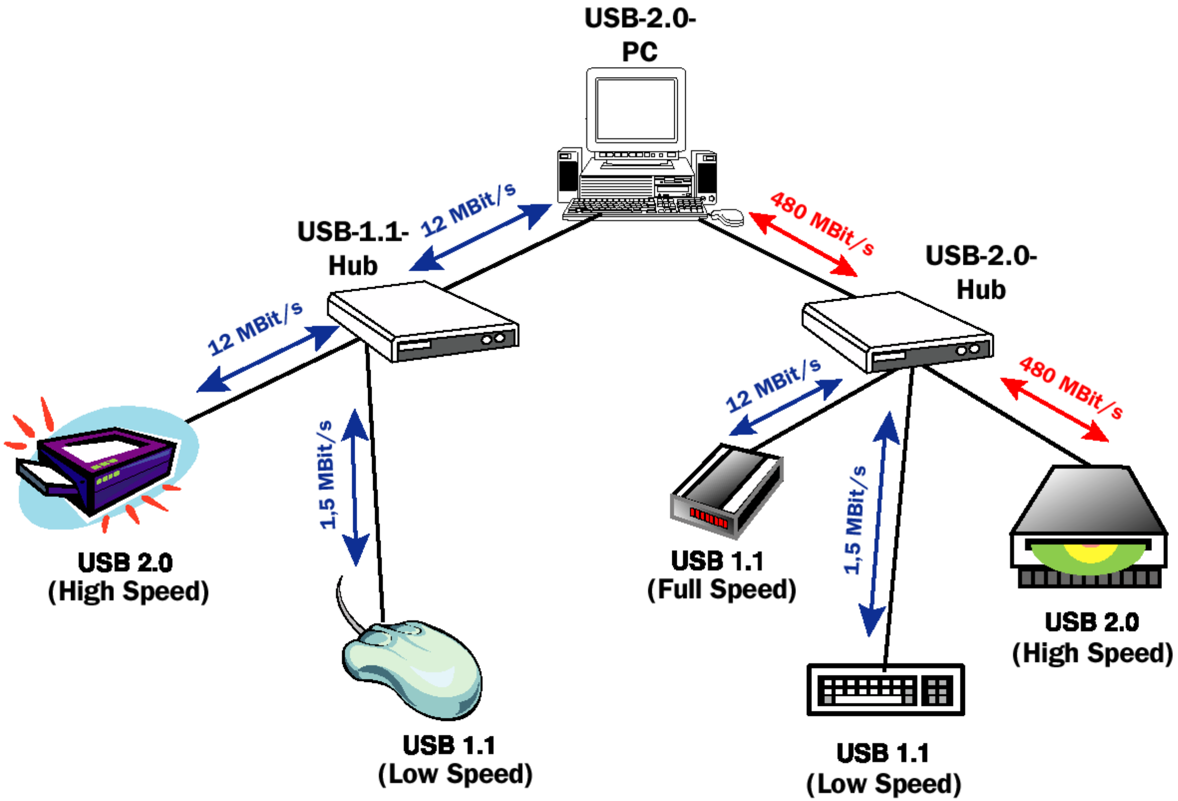




active: max. 500 mA
passive: max. 100 mA

Topology in hybrid mode with USB 1.1 and 2.0

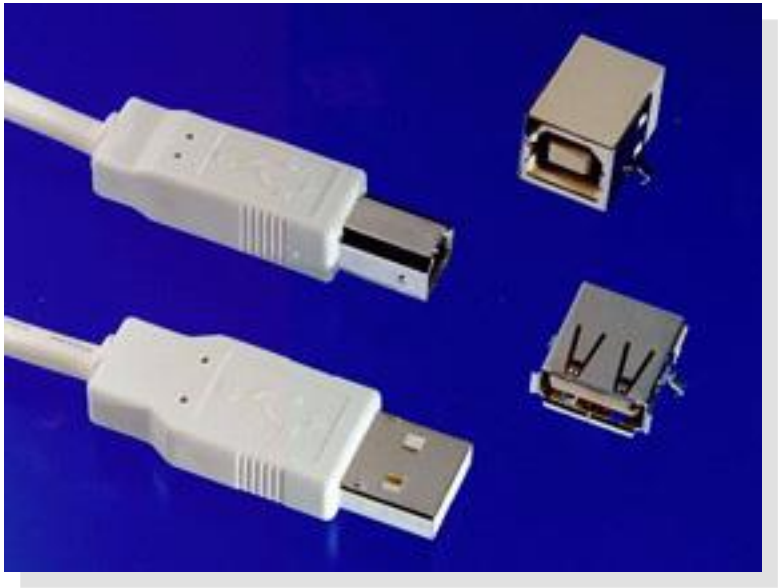
- Hybrid mode
  - USB 2.0 is compatible to USB 1.1
  - High Speed mode is only possible if root hub as well as all intermediate stations support USB 2.0



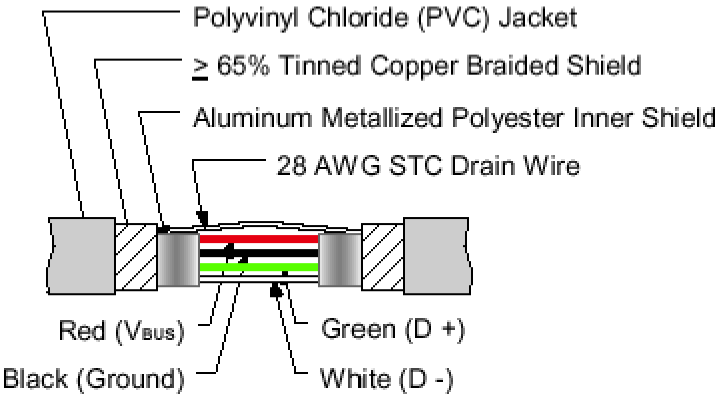
Mechanical connection

- In order to avoid illegal connections, two different types of connectors exist: Series A and Series B
  - Series B: pins in pairs one upon the other
  - Power supply pins are 1mm longer (hot plugging)

Bottom: serie A, top: series B



- Cable
  - 4 wires
  - 2 data wires
  - Integrated power supply


Polyvinyl Chloride (PVC) Jacket
≥ 65% Tinned Copper Braided Shield
Aluminum Metallized Polyester Inner Shield
28 AWG STC Drain Wire
Red (VBUS)
Black (Ground)
Green (D +)
White (D -)

Energy consumption

- Devices can be supplied with energy (5V) over the bus

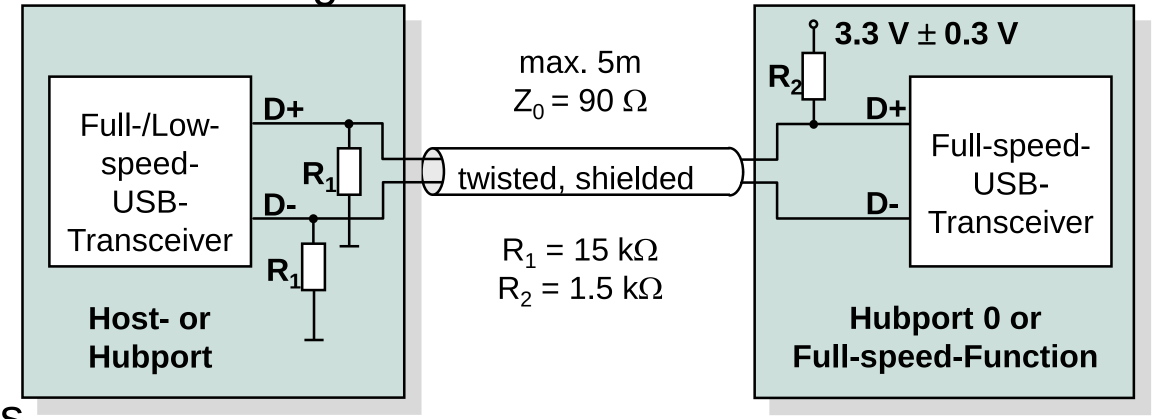
- 3 different classes have to be distinguished
  - Low-power devices: max 100mA
  - High-power devices: max 500mA
  - Self-powered devices: max 100mA from the bus, the rest has to be fetched from their own power supply during operation → otherwise they can be disabled by the host

- Devices have to announce their power consumption at configuration time and are not allowed to go beyond this limit

- Devices have to support standby mode
  - Maximum power consumption of 500µA
  - Devices have to go into standby after 3ms of inactivity on the bus or by dedicated command of the host

Transmission speed detection

- Devices announce their transmission speed using different pull-up resistors at the data wires
- Example for resistor configuration at 12Mbit/s



- At 1.5Mbit/s
  - Low-Speed Transceiver
  - R_2 at D-
  - Cable does not need to be twisted/shielded
  - Z_0 arbitrary
  - Max. 3m wire length
- High-speed devices announce themselves as Full-speed devices (12Mbit/s) first
  - At configuration time, the host is informed about the higher transmission rate possible
  - In normal operation, the resistor is disabled in order to provide symmetrical wire behavior

Differential signaling

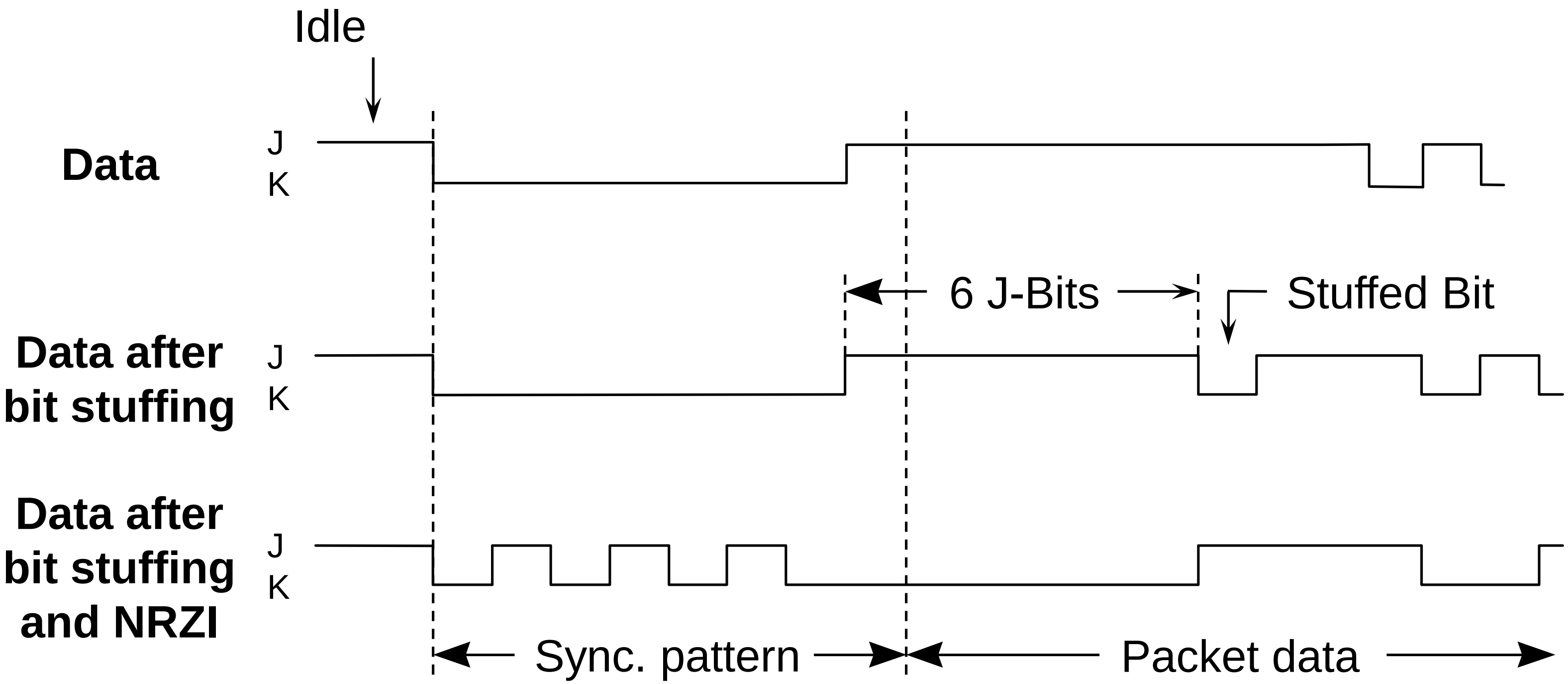
- Voltage on the wires 0-3.3V
- Differential transmission V_{D+}, V_{D-}
  - Differential ONE: $V_{D+} > V_{D-}$
  - Differential ZERO: $V_{D+} < V_{D-}$
- Additional single ended signals are possible
  - V_{D+} and V_{D-} < 0.3V → single ended Zero (SE0)
  - Marks the end of a packet

- Different interpretation of the states depending on transmission speed
  - State ‚J'
    - Differential ZERO at Low-Speed (1.5Mbit/s)
    - Differential ONE at Full-Speed (12Mbit/s)
  - State ‚K'
    - Differential ONE at Low-Speed (1.5Mbit/s)
    - Differential ZERO at Full-Speed (12Mbit/s)

- In High-Speed mode (480Mbit/s) signaling is done using constant currency of 17.78mA

Coding

- Individual bits are encoded using NRZ-I code: Level changes only in K-state
- Bit stuffing: after 6 J-Bits, one K-Bit is inserted



Data transmission

- Bus is controlled only by host

- Host is polling every subscriber und sends or requests data
  - No arbitration scheme and collision avoidance necessary
  - High burden on host because of polling → intelligent controller required
  - Prioritization is determined by host only through order of polling

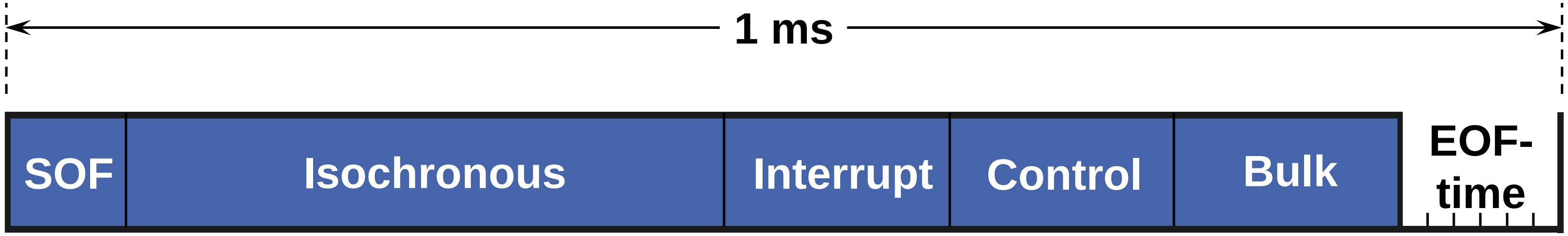
- Isochronous transmission
  - Fixed data rate, no error handling, e.g. for video transmission

- Asynchronous transmission

Control-Transfer	Interrupt-Transfer	Bulk-Transfer
♦ Used for commands and status information ♦ Initialization	♦ Requires a certain service period ♦ Only for IN direction	♦ For large amounts of data ♦ Long waiting time possible ♦ Only at 12 Mbit/s

Data format

- Every frame has a length of 1ms
- Composition of the frame
  - Start-Of-Frame (SOF) packet for synchronization and identification
  - Transfers of the different categories, depending on available time budget
    - Prioritization is as follows:
      - Isochronous, Interrupt, Control, Bulk
    - Max. 90% of the bandwidth can be used for isochronous and interrupt transfers
    - Max. 10% can be used for control transfers
    - Remaining bandwidth can be used for control and bulk transfers, where control transfers have higher priority
  - End-Of-Frame (EOF) time for the rest of the cycle length of 1ms



Transmission packets I

- The packets used for data transmission are composed of the following fields
  - Sync (synchronization)
    - All packets start with a sync field
    - It consists of 7 K-bits and 1 J-bit and is used for synchronization of the senders and receivers clock
  - PID (Packet ID)
    - Determines the type of the packet
  - ADDR (Address)
    - The target address of the packet
    - Length 7bit → 127 devices can be addressed, address 0 is reserved for initialization

Transmission packets II

- ENDP (Endpoint)
  - Sub address within a device
  - Length of 4bit, in Low-Speed devices, only 2 sub addresses are allowed
- CRC (Cyclic Redundancy Check)
  - Checksum for error detection
  - In token packets 5bit, in data packets 16bit long
- EOP (End-Of-Packet)
  - Marks the end of a packet
  - Signalling is done issuing SE0 for 2 bit times followed by one J-bit

Packet types I

- Start-Of-Frame packet
  - Marks the start of a new transfer frame

8 Bits	8 Bits	11 Bits	5 Bits	3 Bits	
Sync	**PID**	**Frame number**	**CRC5**	**EOP**	SOF-Token

- Token packet
  - Determines type and direction of the data to be transmitted
    - IN: Transfer from device to host
    - OUT: Transfer from host to device
    - SETUP: Marks a control transfer

8 Bits	8 Bits	7 Bits	4 Bits	5 Bits	3 Bits	
Sync	**PID**	**ADDR**	**ENDP**	**CRC5**	**EOP**	Token

Packet types II

- Data packet
  - User data of the transmission

8 Bits	8 Bits	0-1023 *Bytes*	16 Bits	3 Bits
Sync	**PID**	**DATA**	**CRC16**	**EOP**

Data

- Handshake packet
  - Used to acknowledge transferred data
  - ACK: Packet has been received successfully
  - NAK: At the moment, device is not able to send/receive data
  - STALL: Error case, intervention of the host is required

8 Bits	8 Bits	3 Bits
Sync	**PID**	**EOP**

- Handshake / Low-Speed-Preamble
  - Switch to Low-Speed mode

PID field

- Using the PID field, the device is able to detect the type of packet
- The PID has a length of 4bit, for error detection purpose, these bits are additionally appended in inverted form
  - If the bits do not match, the packet is dropped

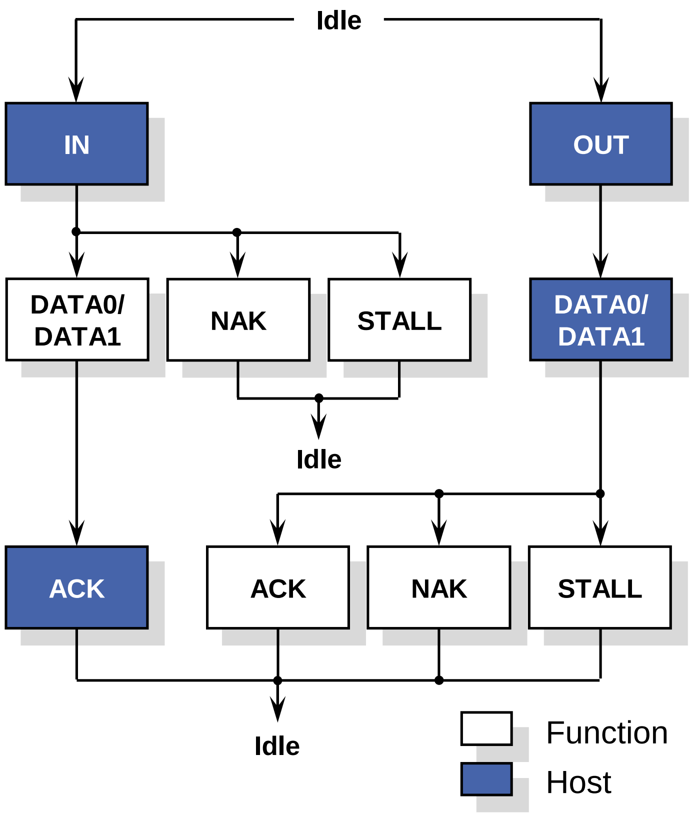
(LSB)							(MSB)
PID_0	PID_1	PID_2	PID_3	$\overline{PID}_0$	$\overline{PID}_1$	$\overline{PID}_2$	$\overline{PID}_3$

Paket type	PID name	PID [bit3:bit0]	Description
Token	OUT	0001	Transfer from host to device
	IN	1001	Transfer from device to host
	SOF	0101	Start of a new frame, includes frame number
	SETUP	1101	Device is to be configured by the host
Data	DATA0	0011	Changes after each successful transmission
	DATA1	1011	
Handshake	ACK	0010	Reception without errors
	NAK	1010	Device is not able to send/receive data
	STALL	1110	Same as NAK, but action from the host required
Special	PRE	1100	Used to access a low-speed function

Transfer process

- The host starts a transfer by sending a corresponding token packet
- Then one ore more data packets are transmitted
- Each packet is acknowledged by a handshake packet
- If errors have occurred during a transmission, no handshake packet is send. The sender has to retransmit the packet
  - For isochronous transfers there is no handshake. Defective data is not transmitted again!

Exemplary Bulk Transfer:



Adding a device to the bus (Enumeration)

1. The hub/host detects a new device via the pull-up resistors
2. A reset is being performed in order to set the device to a well defined state. The device then uses the reserved address 0.
3. The host queries the configuration parameters of the device
4. The host assigns an address to the device and configures the device using control transfers
5. The device is now ready to operate on the bus

Pictures: http://www.yatego.com

USB on-the-go (OTG)

- Extension of USB 2.0 used to connect two devices directly
  - No host required
  - Can be used e.g. to connect digital camera and printer

- Requirements for such a dual-role device
  - Simple host capabilities
  - List of supported devices (not all possible USB devices have to be supported)
  - A micro-AB connector (new type of connector)
  - Has to be able to drive 8mA to the bus

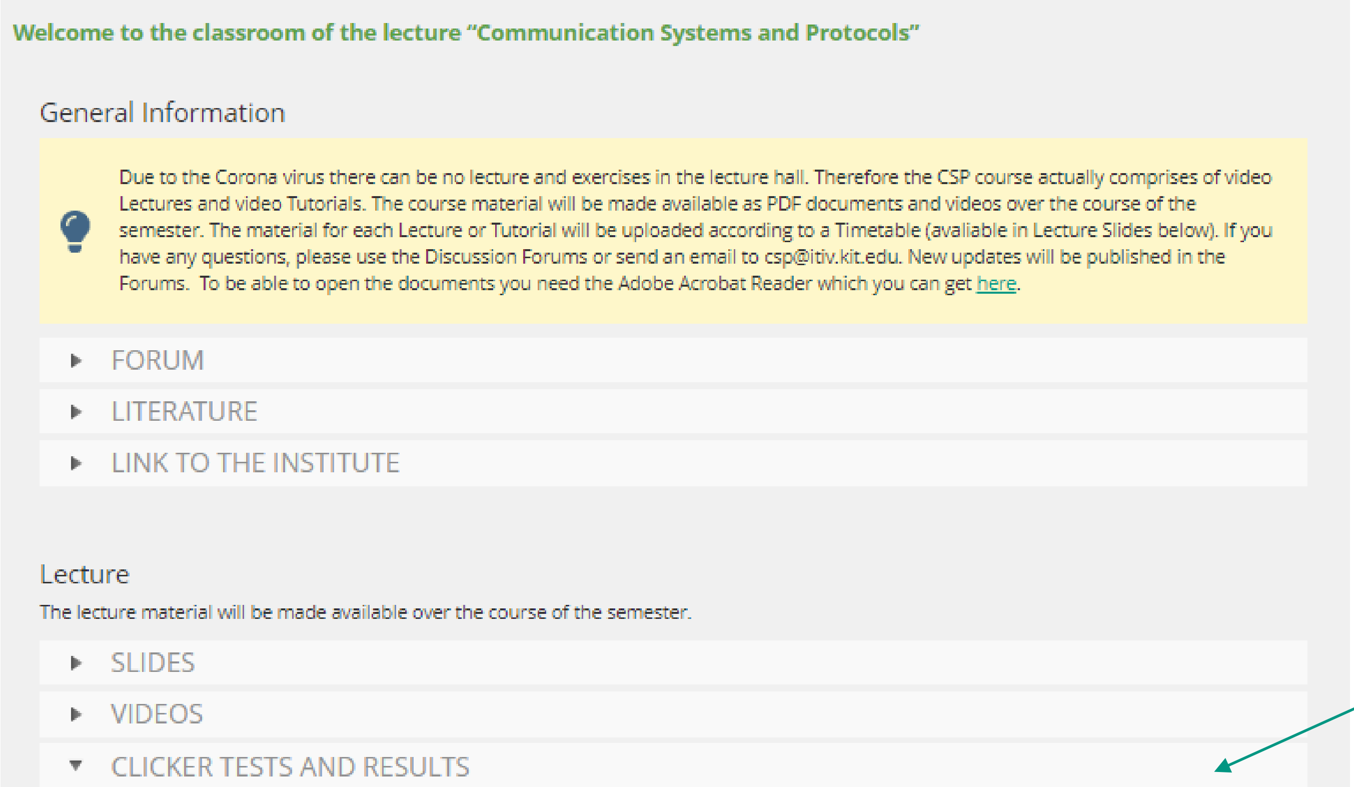
USB 3.0

- Utilizes a dual bus architecture which is backward compatible with USB 2.0
- Includes SuperSpeed and non-SuperSpeed (USB 2.0 speeds) information exchange
- Tiered Star topology similar to USB 2.0

Characteristic	SuperSpeed USB	USB 2.0
Data Rate	**SuperSpeed : 5.0 Gbps**	Low-Speed : 1.5 Mbps Full-Speed : 12 Mbps High Speed : 480 Mbps
Data Interface	Dual-Simplex, four wire differential Signalling. Simultaneous bi-directional data flows.	Half-duplex two-wire differential signaling. Unidirectional data flow with negotiated directional bus transitions
Cable signal count	Six : Four for SuperSpeed data path. Two for non-SuperSpeed datapath	Two : Two for low-speed/full-speed/high-speed data path.
Bus transition protocol	Host directed, asynchronous traffic flow. Packet is explicitly routed.	Host directed, polled traffic flow. Packet traffic is broadcast to all devices.

Source : Universal Serial Bus 3.0 Specification, Revision 1.0, November 12,2008

Participate in Clicker Test 8b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Karlsruher Institut für Technologie

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Session 24: CAN

Controller Area Network (CAN)

- 1981 developed by Bosch and Intel
- Goal: Cross-linkage of complex controllers and control units
- International dissemination in the automobile area, the sector of home appliance, in textile machines, in medical engineering devices, …
- Low priced bus transceivers through high volume number

CAN Characteristics

- Multi master architecture
- High transmission rate (max. 1 Mbit/s)
- Message distribution system
- Message prioritization
- Network wide data consistency
- Priority controlled and collision free bus access
- Very high failure recognition capability
- Powerful failure treatment
- Node monitoring
- NRZ coding with bit stuffing

Field of Application

- Factory automation
  - Networking of control units of the lowest automation level (field level) in order to be able to capture and affect process values
- Building automation
  - Automation of important operating technological functions like heating, ventilation, illumination, securing, observing
- Mobile Systems
  - Networking of components of body and comfort electronics as well as control units of the powertrain
- Machines, facilities, devices
  - Networking of actuators and sensors in medical- or naval technological devices, printing-, textile- and paper machines

Special Requirements on CAN

- Sending
  - Every node is allowed to send as soon as the physical transmission medium is free
  - Transmission of data carried out on demand or by initiation of the transmitter node
  - Provides cyclic communication, request-based communication, and event-based communication
- Reception
  - Message shall be available to every node for reception (receiver selective system)

Special Requirements on CAN

- High reliability
  - High safety and availability
  - Fault tolerance

- Time critical data transmission
  - Efficient utilization of the transmission bandwidth
  - Short bus access latency and transmission time
  - Short failure recovering phase

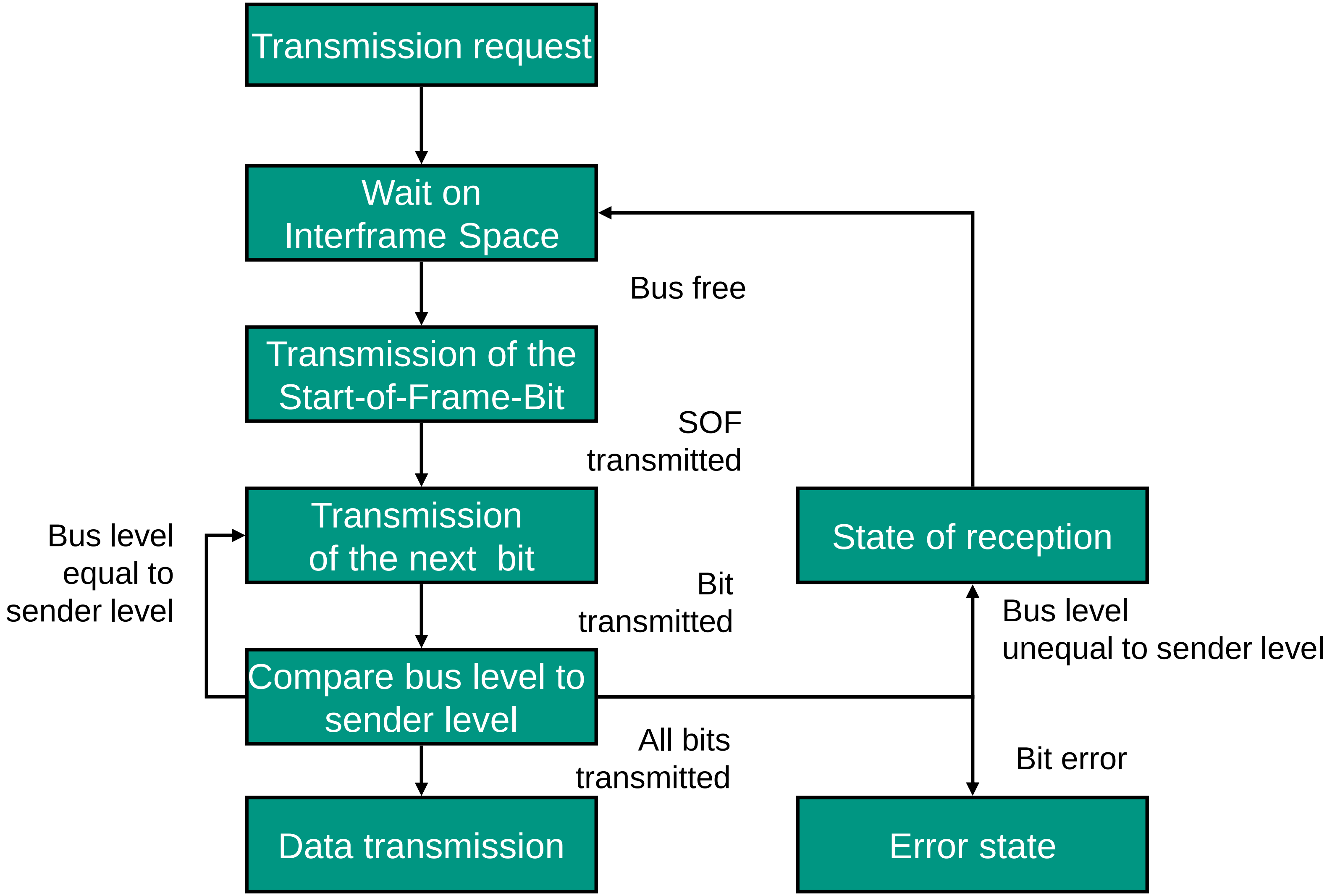
Bus Arbitration

- Participants are connected to the bus via open collector drivers
  - Dominant LOW, logical ‚0‘
  - Recessive HIGH, logical ‚1‘
    → „Wired-AND“

- Arbitration scheme: CSMA/CA

- Arbitration is done with the help of the identifiers of a message
  - Identifier with the lowest value has got highest priority (dominant ‚0‘)
  - Every identifier can exist only once within the whole system

Procedure of bit wise bus arbitration


Transmission request
Wait on Interframe Space
Bus free
Transmission of the Start-of-Frame-Bit
SOF transmitted
Transmission of the next bit
Bus level equal to sender level
Bit transmitted
Compare bus level to sender level
All bits transmitted
Data transmission
State of reception
Bus level unequal to sender level
Bit error
Error state

Data Transmission

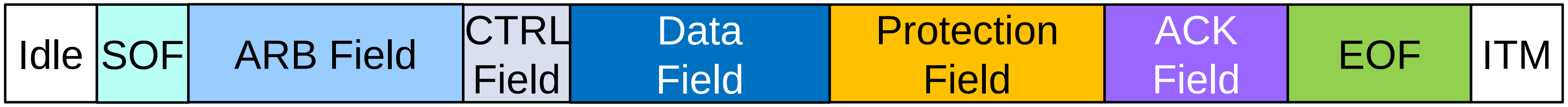
- As coding method the NRZ code is used for the CAN bus
  - Long chains of equivalent bits do not provide enough information for clock recovery

- Bit stuffing to generate enough clock information
  - After five bits of the same polarity an inverse stuff bit is inserted
  - The stuff rule is broken by a special error frame which can be recognized

Message Formats

- There exist four different types of frames:
  - **Data frame** for data transmission from a sender to one or several receivers initiated by the sender
  - **Remote frame** to request a data frame with the same ID from another participant
  - **Error frame** for signaling an error which has been recognized by a participant
  - **Overload frame** provides the possibility of a delay between two data or remote frames

CAN Data Frame

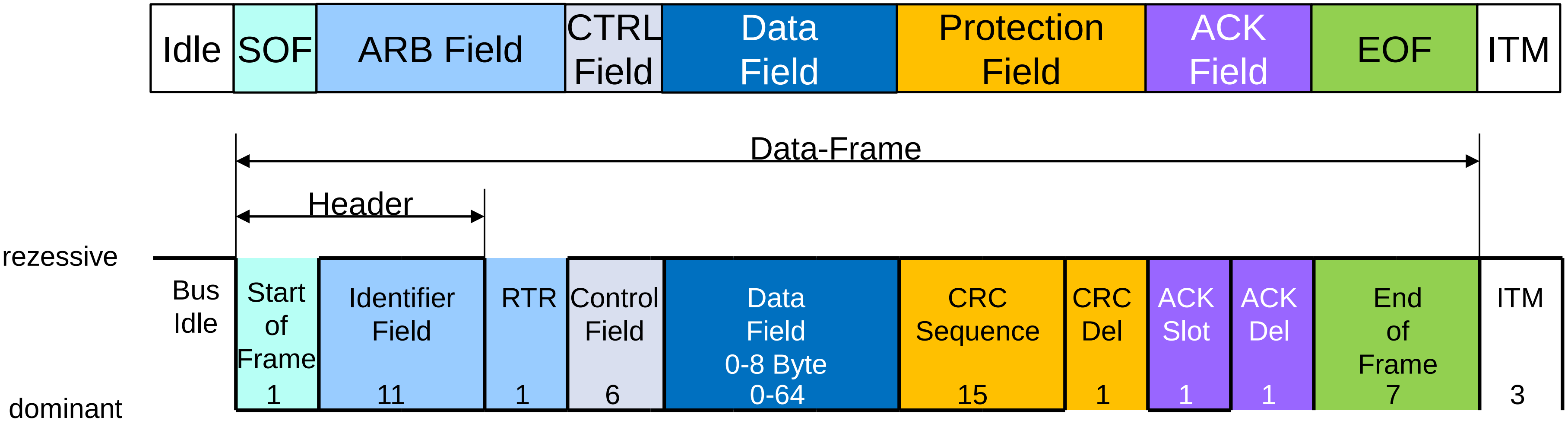


Idle	SOF	ARB Field	CTRL Field	Data Field	Protection Field	ACK Field	EOF	ITM

- Transmission only if bus is idle
- CSMA/CA performed with Arbitration Field (ARB)
- Data field has variable length which is defined by the Control Field
- Bitstuffing is performed up to the last bit of the Protection Field (excluding its delimiter)
- Within the ACK Field, receivers acknowledge the successful reception of a message
- End-of-Frame (EOF) enables participants which are unsynchronized to synchronize again
- Intermission (ITM)

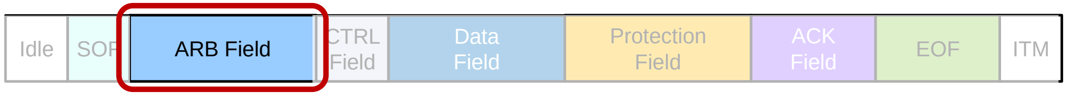
Data Frame I

- General structure of a data frame:

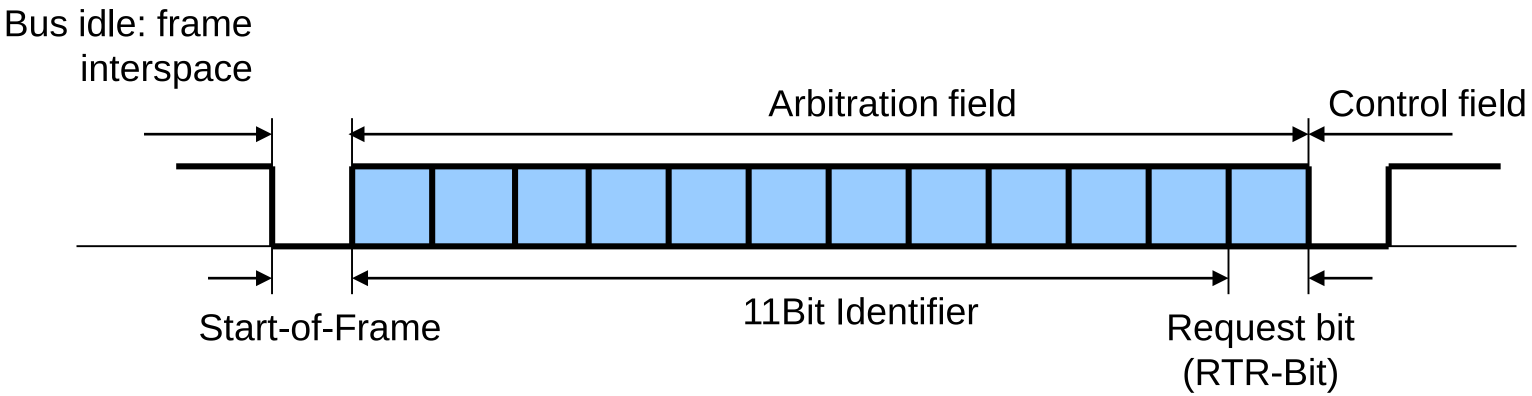


- Start label (Start-of-Frame)
  - The participant is only allowed to send if the bus is idle, that means that it has been at recessive level for at least 11 bit times
  - The first signal edge of the start bit synchronizes all participants on the participant that started the arbitration process

Data Frame II

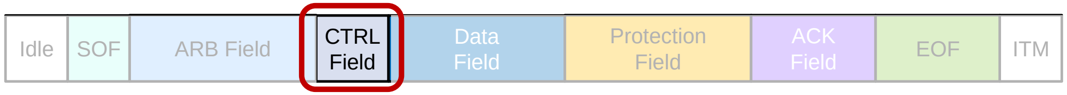


- Arbitration field

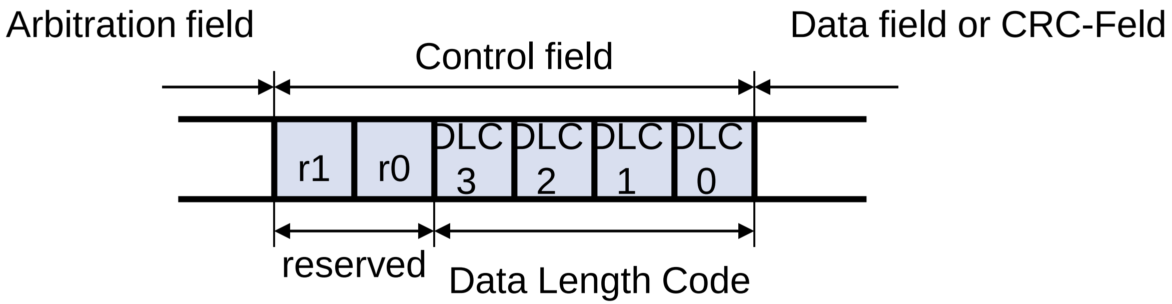


- Identifier marks a message and defines its priority
- MSB is transmitted first
- In a remote frame the RTR bit is set to recessive level in a data frame to dominant level
  → Transmission of a message has got higher priority than requesting a message

Data Frame III



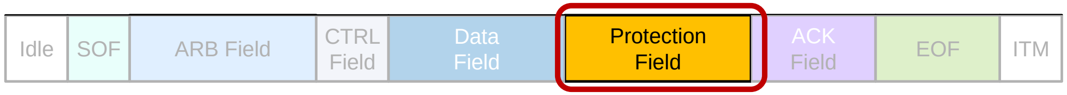
- Control field



  - Length specification of the data field with the help of the four least significant bits (Data Length Code, DLC)
  - Most significant bits are reserved (r1,r0)

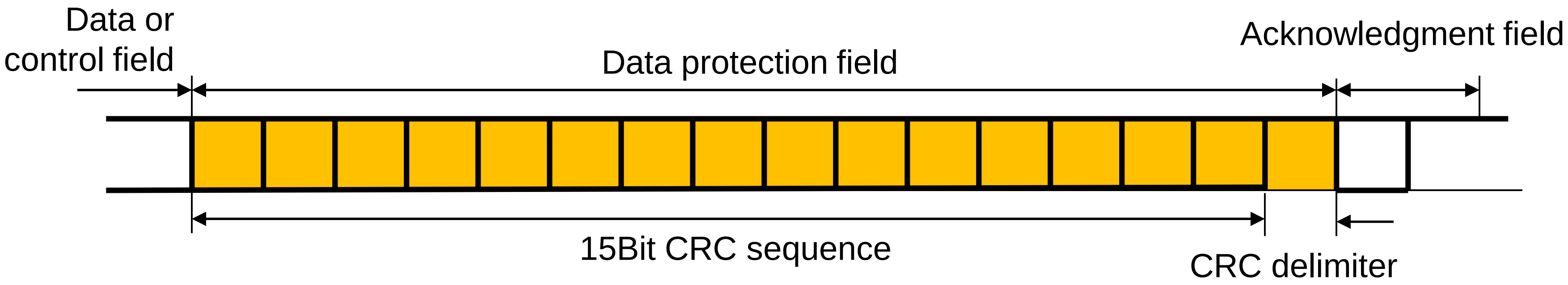
- Data field
  - User information of a message
  - Length between 0 and 8 Byte

Data Frame IV



Idle | SOF | ARB Field | CTRL Field | Data Field | Protection Field | ACK Field | EOF | ITM

- Data protection field



Data or control field

Data protection field

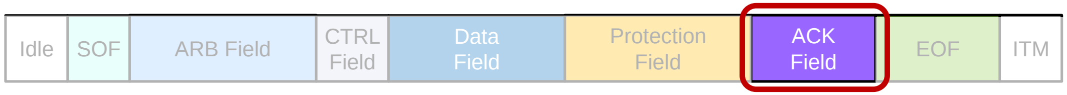
Acknowledgment field

15Bit CRC sequence

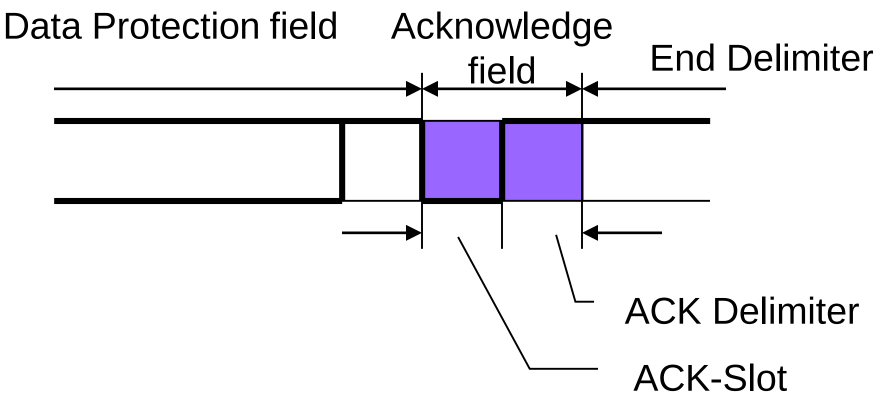
CRC delimiter

- 15 bit CRC sequence for the recognition of transmission errors
- CRC polynomial: $x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1$
- One recessive delimiter bit

Data Frame V

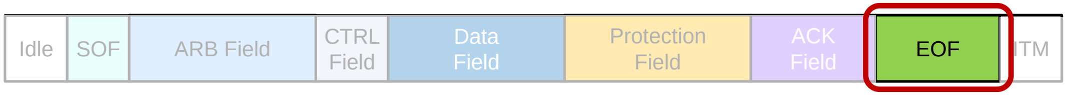

Idle | SOF | ARB Field | CTRL Field | Data Field | Protection Field | ACK Field | EOF | ITM

- Acknowledgement field

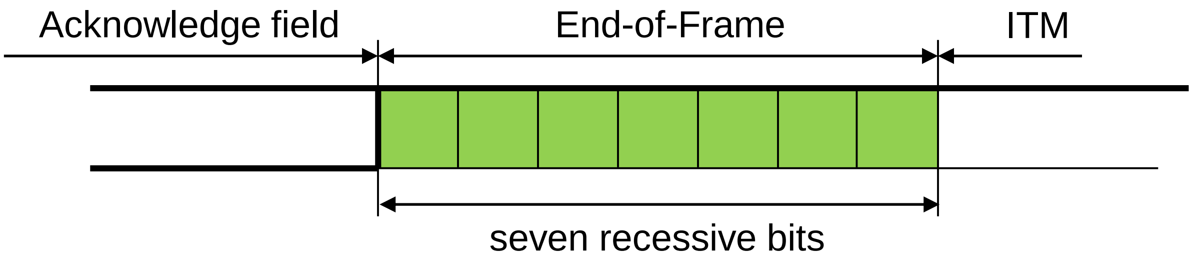


- Within the ACK slot every receiver acknowledges successful reception of a message by transmitting a dominant bit
  - Since that way the recognition of errors within single participants is not possible the error frame is utilized
- The field is always delimited by a recessive delimiter bit in order to be able to differentiate between a positive reception acknowledgement and an error flag that is starting at the same time

Data Frame VI



- End-of-Frame (EOF)
  - Every data or remote frame is delimited by a sequence of seven recessive bits

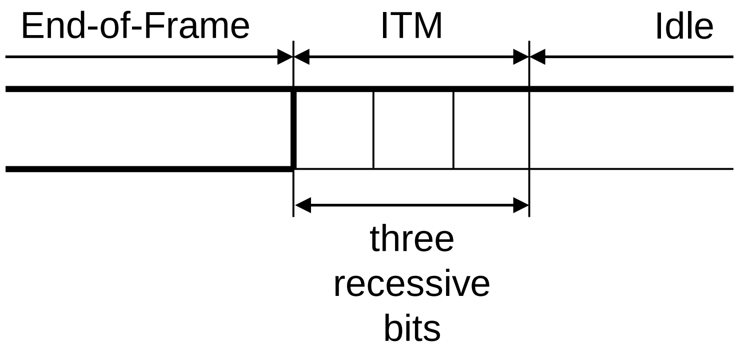


  - This also enables participants which are completely unsynchronized to synchronize again
    - The participant interprets EOF as stuff error
    - It transmits an error frame
    - It waits for a new initiation of a transmission

Data Frame VII

Idle	SOF	ARB Field	CTRL Field	Data Field	Protection Field	ACK Field	EOF	ITM

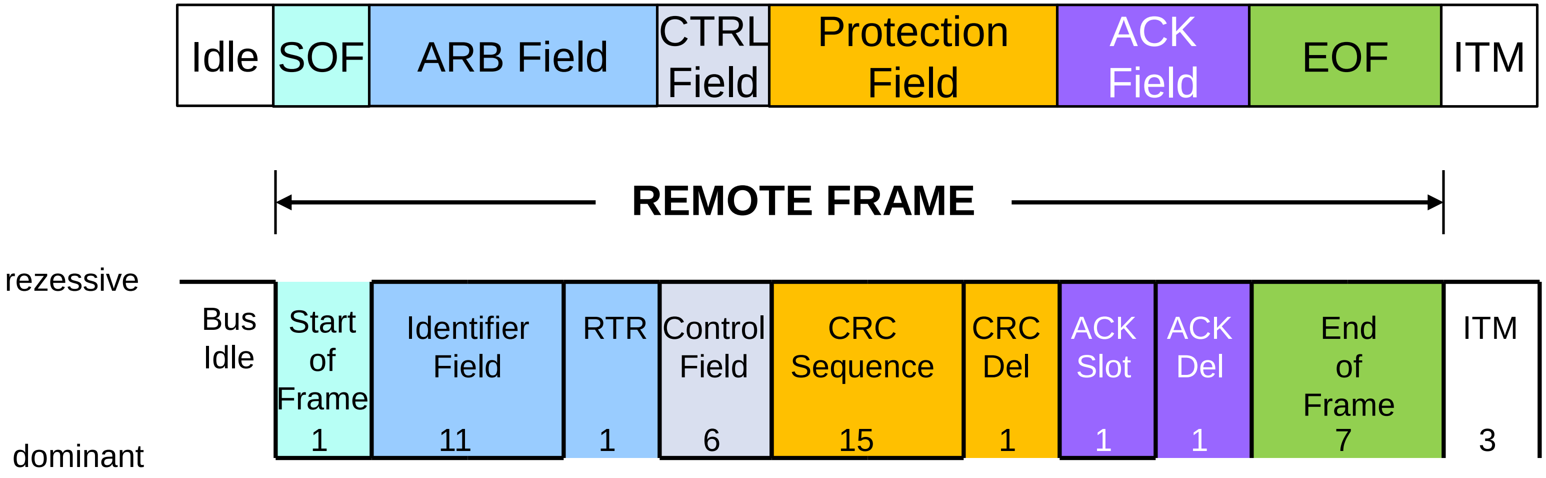
- Intermission (ITM)
  - Data Frames (and Remote Frames) are separated from preceding frames using Interframe Spacing (ITM followed by Idle)



  - ITM consists of three recessive bits
  - During an ITM no bus node is allowed to start a transmission of a new data frame or remote frame
  - Within the first bit of the ITM an Overload Condition may be signalled

Remote Frame (standard format)

- Data request



Idle	SOF	ARB Field	CTRL Field	Protection Field	ACK Field	EOF	ITM

REMOTE FRAME

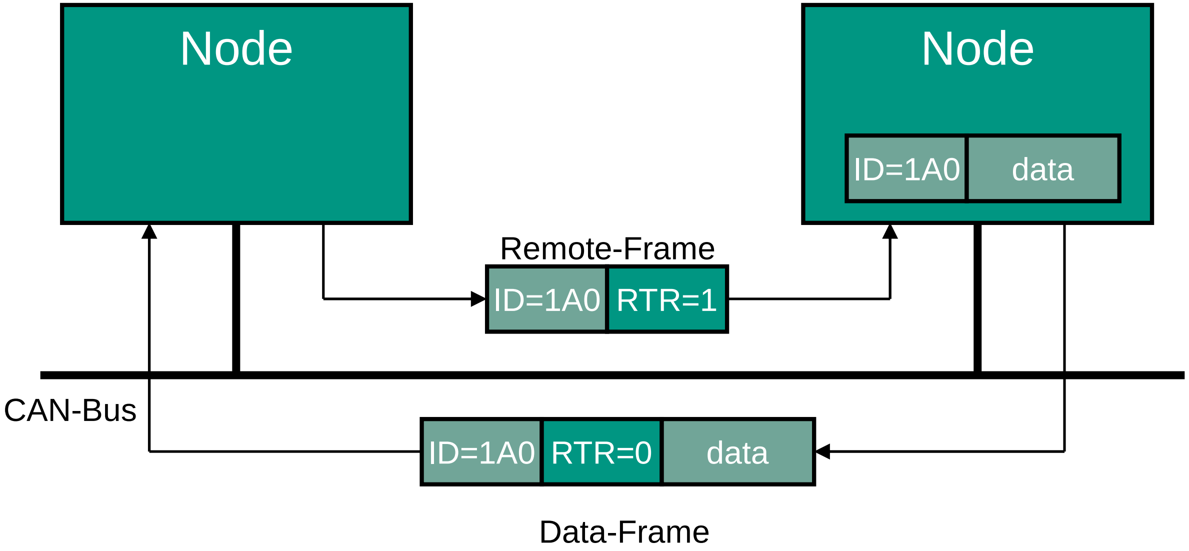
rezessive dominant	Bus Idle	Start of Frame	Identifier Field	RTR	Control Field	CRC Sequence	CRC Del	ACK Slot	ACK Del	End of Frame	ITM
		1	11	1	6	15	1	1	1	7	3

- Structure is similar to the data frame with missing data field
- Request for a message with the same identifier
  - Transmission of a message has got higher priority as request of it

  → RTR bit is recessive for a request
- At most there is one sender for a message while several receivers are permitted!

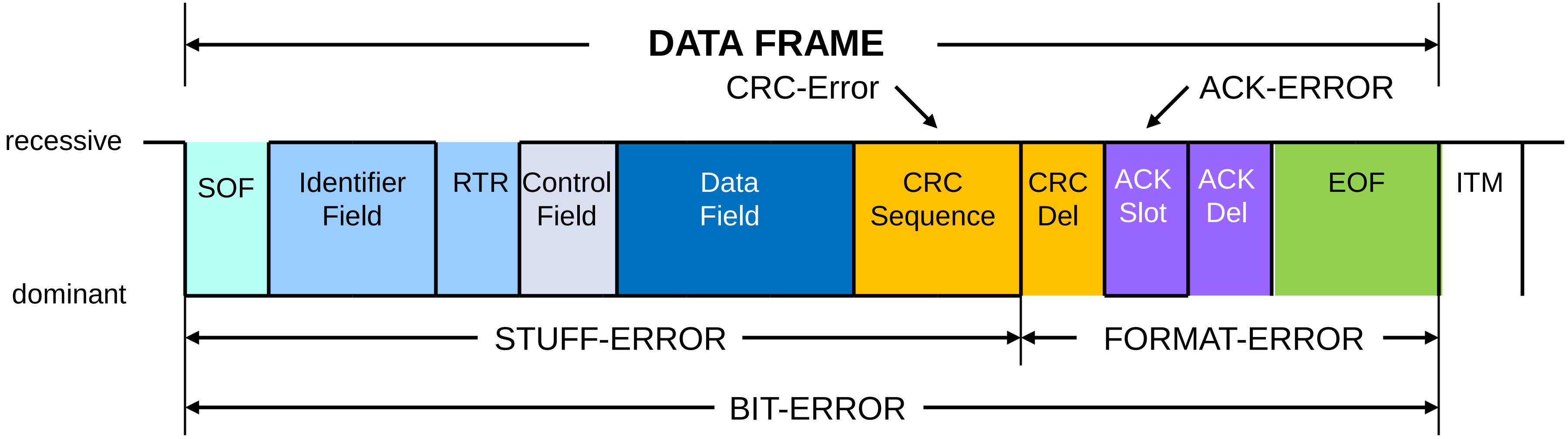
Message transmission

- Every participant can transmit messages initiated by itself
- A participant can request a message by transmitting a request frame with the desired ID
- The responsible node transmits requested frame with desired data



Error Detection Mechanisms 1

- For error detection, the following mechanisms are used:



- Detecting an error leads to the transmission of an error frame and that way to a termination of data transmission
- Data transmission will be repeated afterwards

Error Detection Mechanisms 2

- Bit Monitoring
  - Every participant checks whether the level that it puts on the bus also appears on the bus
  - Enables the detection of all globally effective errors
  - Additional detection of all errors that are local at the sender
- Monitoring of the frame format (Message Frame Check)
  - Violation of defined frame format elements (e.g. recessive delimiter bits)
- Cyclic Redundancy Check (CRC)
  - Protection of a part of the transmission frame by a 15 bit long CRC field
  - Error message after the acknowledge delimiter
  - Polynomial used: $x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1$
    - Enables the recognition of at most 5 arbitrarily distributed errors or burst errors of up to 15 bits length

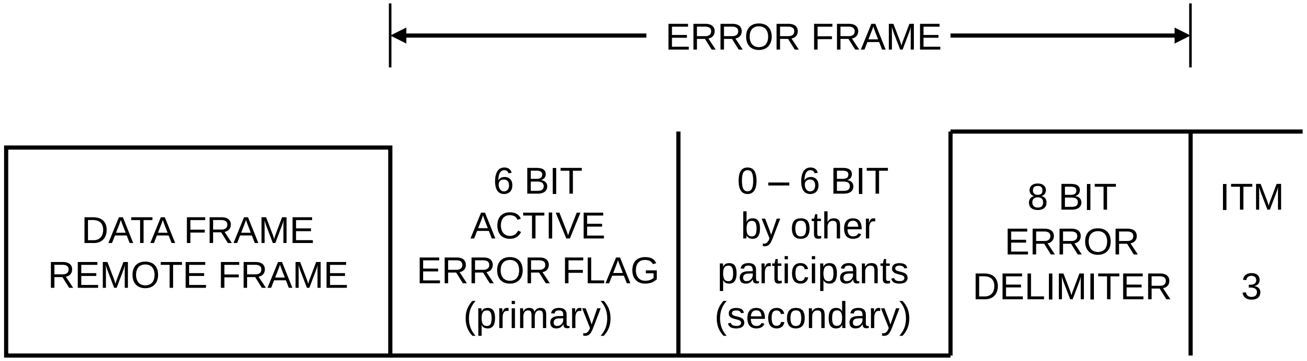
Error Detection Mechanisms 3

- Monitoring of acknowledgement by the sender
  - Sender waits for a confirmation of error free reception inside the ACK slot
  - A missing confirmation (recessive level in the ACK slot) is interpreted as an error that was caused by the sender himself

- Monitoring of the bit stuffing rule
  - Bit stuffing is used in the fields from SOF to CRC delimiter (not including the CRC delimiter)
  - After 5 bits of the same level a bit of the inverse polarity must follow
  - A violation of this rule is interpreted as error

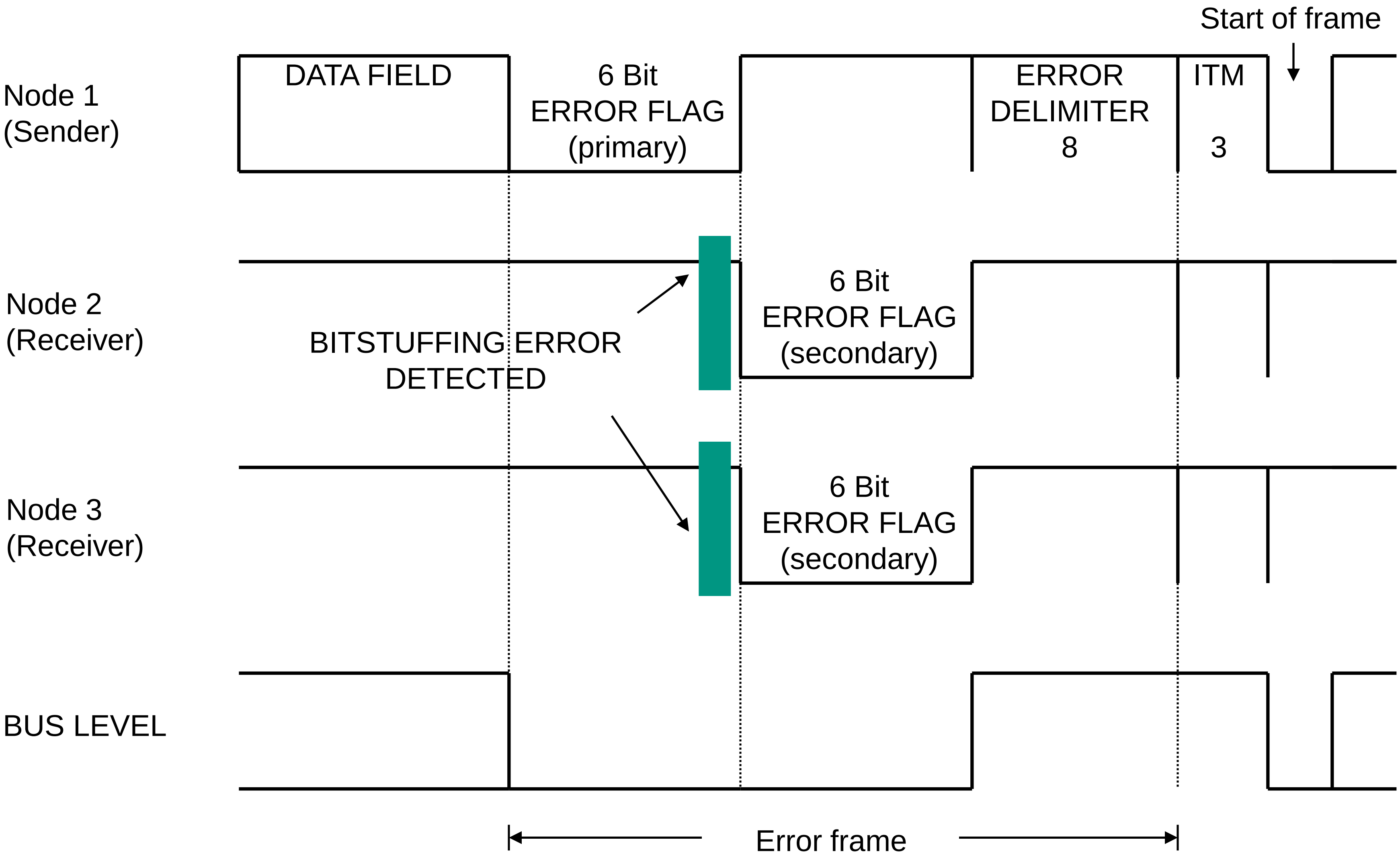
Error Frame

- Used to cancel a faulty transmission by performing an intended code violation

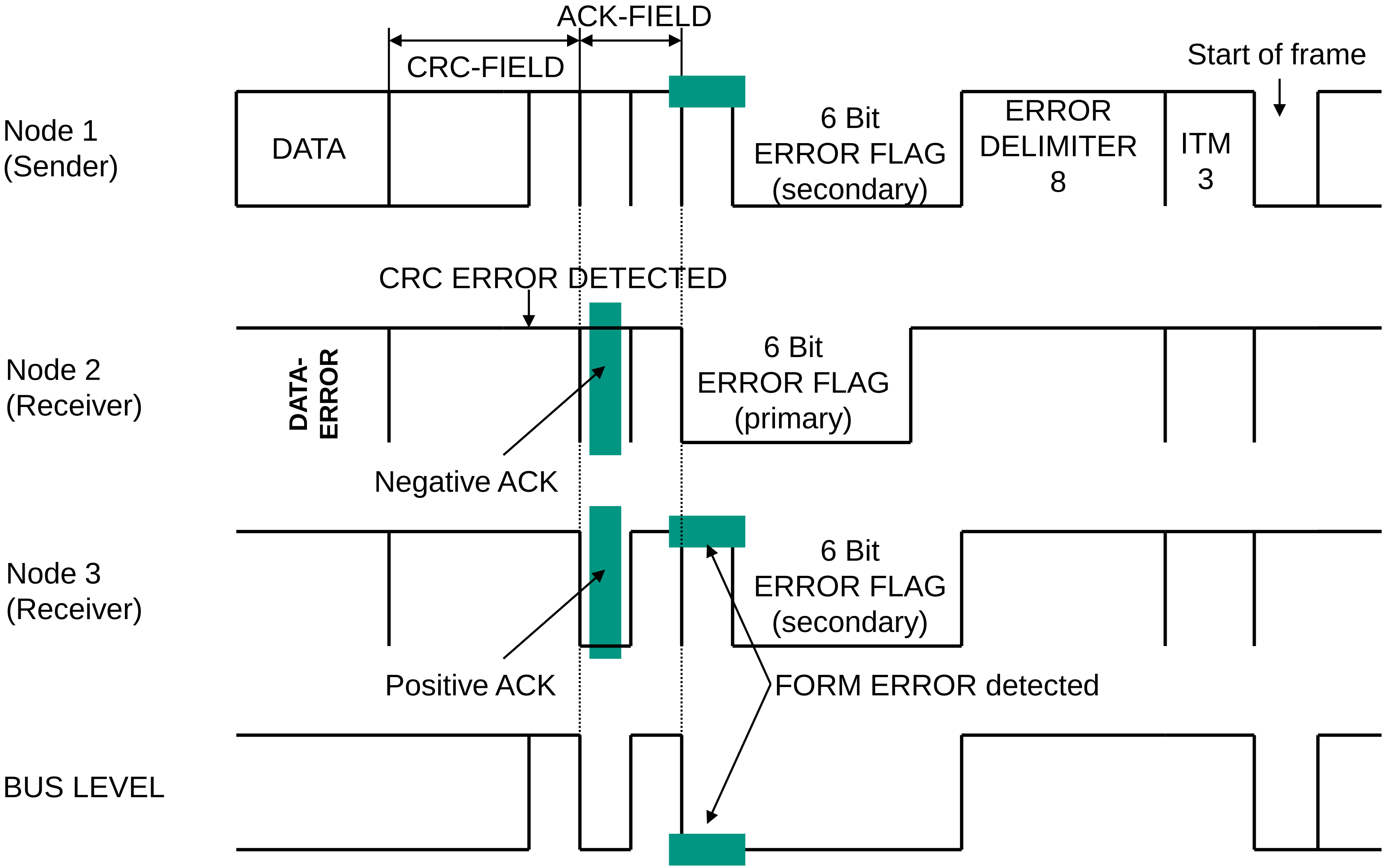


- Transmission of 6 bits with same polarity (error flag)
  → Violation of the stuffing rule
- After error recognition, other participants also transmit an error flag
  → Length of error frames can vary
- Closed by transmission of 8 recessive bits to mark end of transmission

Example: Local failure in the sender



Example: Local failure in the receiver



Error Isolation

- Malfunctioning participants can render the bus unusable by persistent transmission of error flags
  - Erroneous participant must be recognized and deactivated

- CAN nodes include send and receive counters
  - If an erroneous transmission or reception is recognized the counters are incremented
  - Successful transmissions and receptions decrement the counters

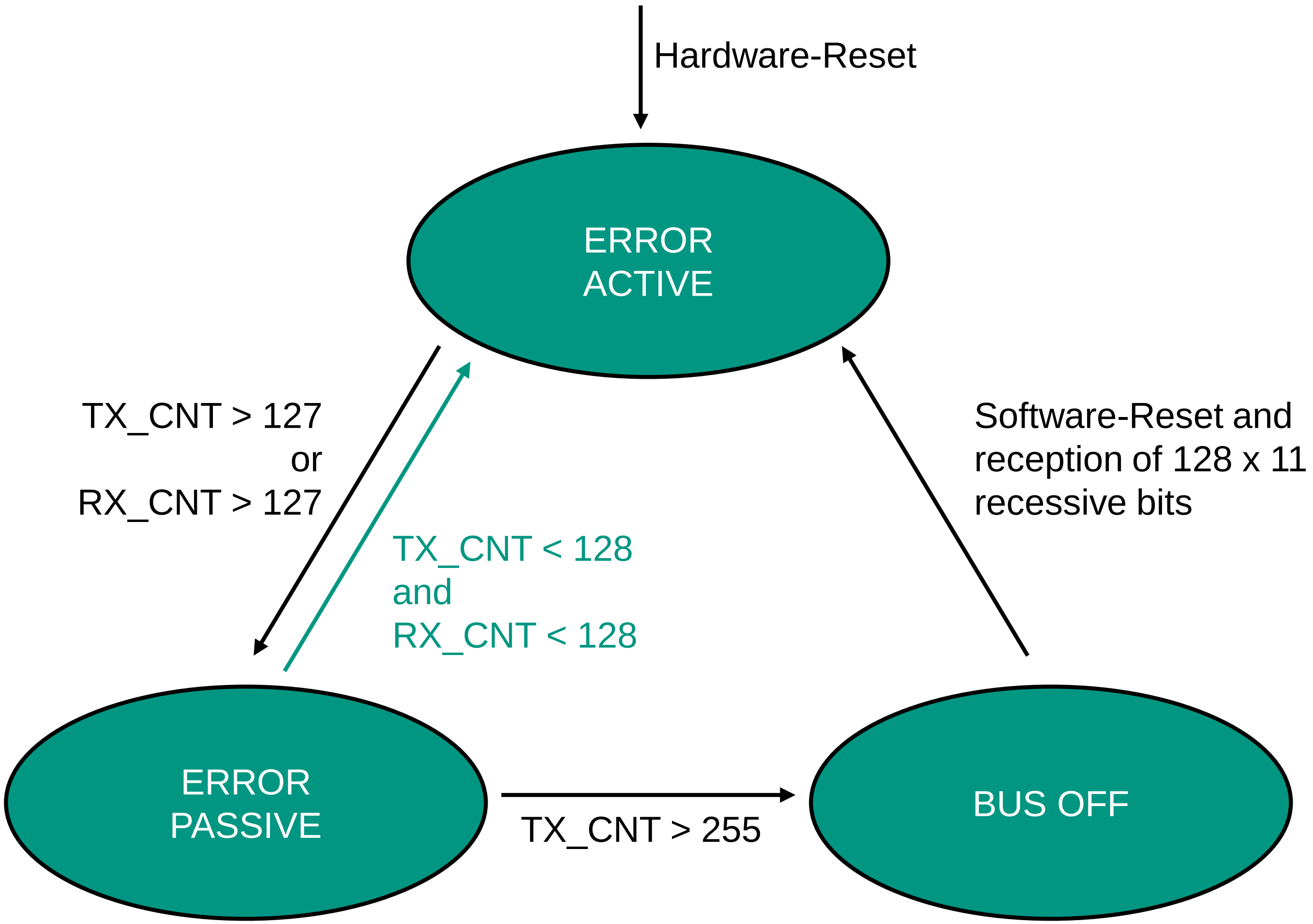
Error Isolation

- Starting state ‘error active’, full-valued participant.
  - Sends an Active Error Flag when error has been detected.
    - Active Error Flag → Consists of six consecutive ‘dominant’ bits

- If the value 127 is exceeded the node becomes ‘error passive’.
  - It is no more allowed to generate a Active Error Flag.
  - Sends Passive Error Flag when an error has been detected.
    - Passive Error Flag → Consists of six consecutive ‘recessive’ bits unless it is overwritten by ‘dominant’ bits from other nodes

- If the error counter further increases the node has to be disconnected from the bus (bus off)

Error states



RX_CNT: RECEIVE ERROR COUNT

TX_CNT: TRANSMIT ERROR COUNT

Rules for Modification of the TX_CNT

- Sender transmits an error flag (this is the case if a bit error is recognized or no ACK is received)
  - TX_CNT => TX_CNT + 8

- After successful transmission of a message
  - TX_CNT => TX_CNT – 1

- If an error passive sender recognizes an ACK error (without a following error flag) the TRANSMIT ERROR COUNT is not incremented any more
  - This shall prevent a node in a network without other nodes to turn itself off again, since it does not receive an ACK (e.g. if different nodes have different start up times)

Rules for Modification of the RX_CNT

- Receiver sends an error flag
  - RX_CNT => RX_CNT + 1

- If the receiver has transmitted an error flag first
  - RX_CNT => RX_CNT + 8

- After successful reception of a message
  - RX_CNT => RX_CNT – 1, if current count between 1 and 127
  - RX_CNT => [119..127], if current count > 127

- A node can never be set to “bus off“ by the RECEIVE ERROR COUNT

Error Handling

1. Detection of an error

2. Immediate transmission of an error frame, for CRC errors only after the ACK delimiter (if a node is error passive he can only terminate the message that he transmitted himself)

3. The message is discarded within all nodes

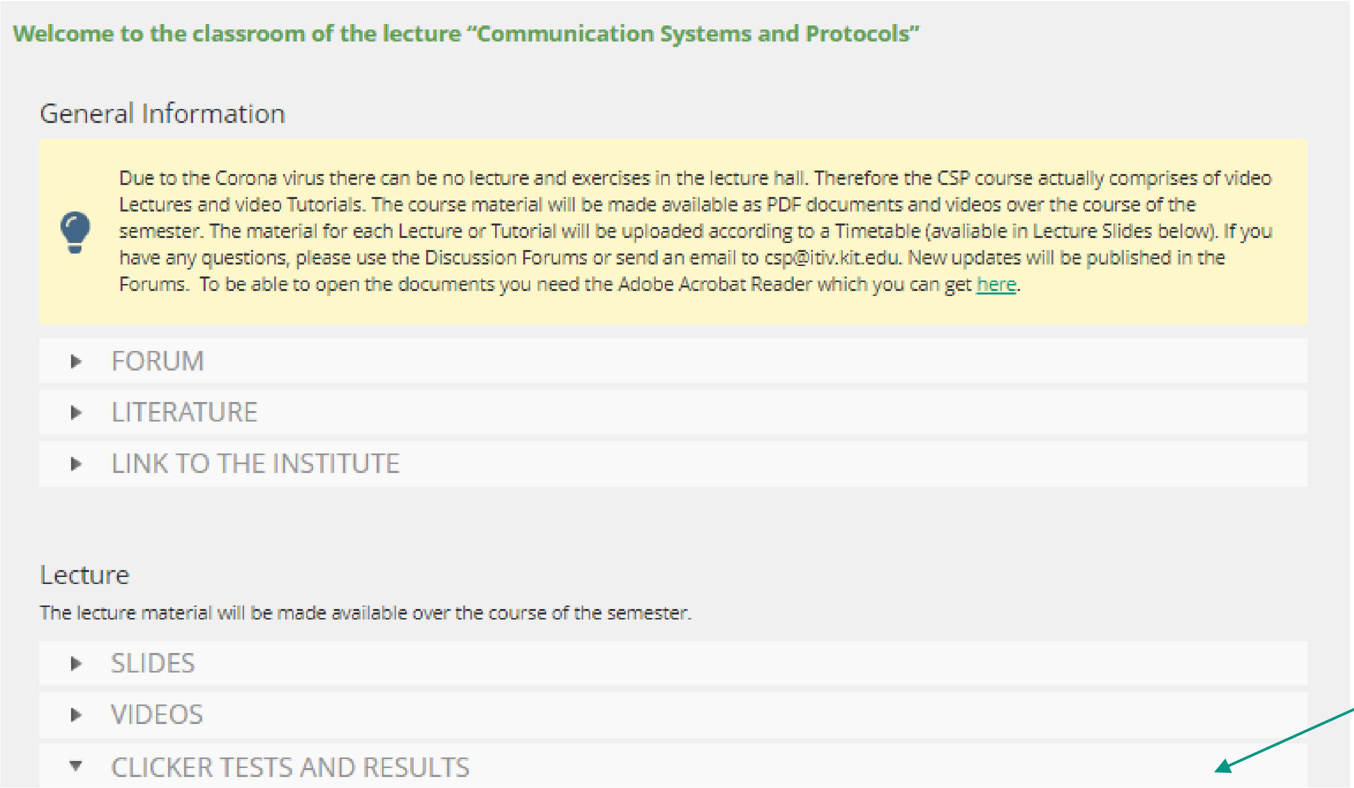
4. The error counters within all nodes are appropriately incremented

5. The message is automatically repeated

Overload Frame

- Used for delaying data transmissions
- Can be interpreted as special error frame
- An overload frame may only start within the first bit of the frame interspace
- Transmission of 6 dominant bits within the frame interspace
  - Other participants recognize this and also send an overload frame
- Closed by a chain of 8 recessive bits
- At most two frames in a row allowed

Participate in Clicker Test 9a



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Karlsruher Institut für Technologie

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Session 25: LIN

Local Interconnect Network (LIN)

- Cost-sensitive sub-bus for CAN
- Non safety-critical applications
- LIN consortium has been founded by well-known car manufacturers and suppliers as well as semiconductor and tool providers
- OEM-independent standard for communication in sensor/actuator field
- Criteria:
  - Simple and cost-efficient physical layer
  - Lean communication protocol (up to 20 kbit/s)
  - Methodology for automated design of LIN nodes and LIN networks

> ***"LIN is a cost-competitive serial communication system designed for localized vehicle electrical networks"*** (www.lin-subbus.org)

LIN Specification

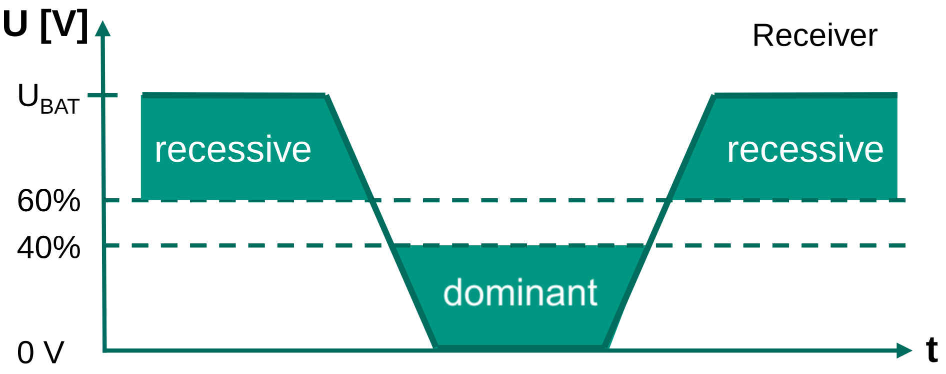
- First specification in 1999
  - LIN-Protocol-Specification
  - LIN-Configuration-Language-Specification
  - LIN-Application-Interface-Specification
- Version 2.0 in 2003
  - Adaptation to demands of the automotive industry
    - “Off-the-shelf” nodes (nodes that can be adapted to specific conditions in a LIN cluster)
    - Automated Design and generation of LIN networks
- Version 2.1 in 2006
  - Better understandability of specification
  - Removed inconsistencies
- Standardized by OSI with version 2.2A → ISO 17987-1

LIN-Bus Level

- Uses single wire transmission with common ground of the vehicle (no differential signaling as known from CAN)

- Open-collector drivers with Pull-up resistor
  - LOW state is referred as dominant level
  - HIGH state is referred as recessive level

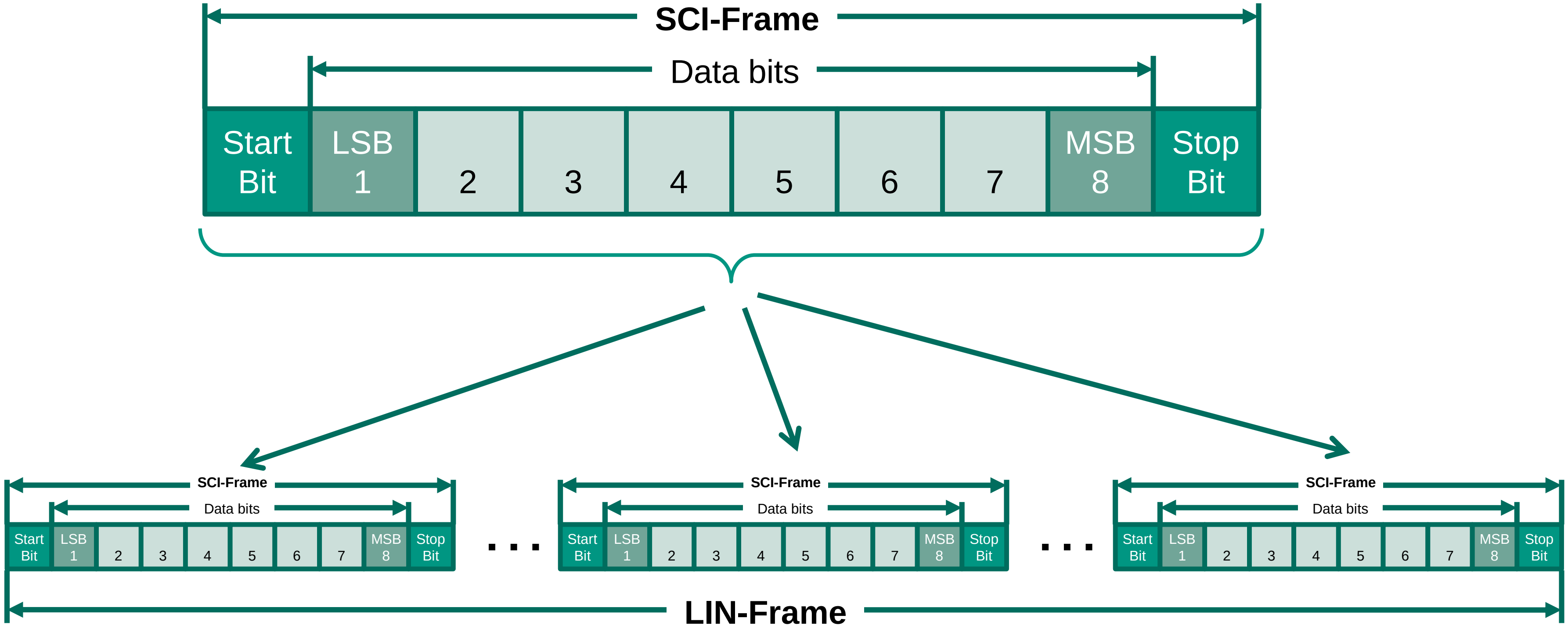
  - Level below 40% of supply voltage is interpreted as logical “0”
  - Level above 60% of supply voltage is interpreted as logical “1”



LIN Data Format

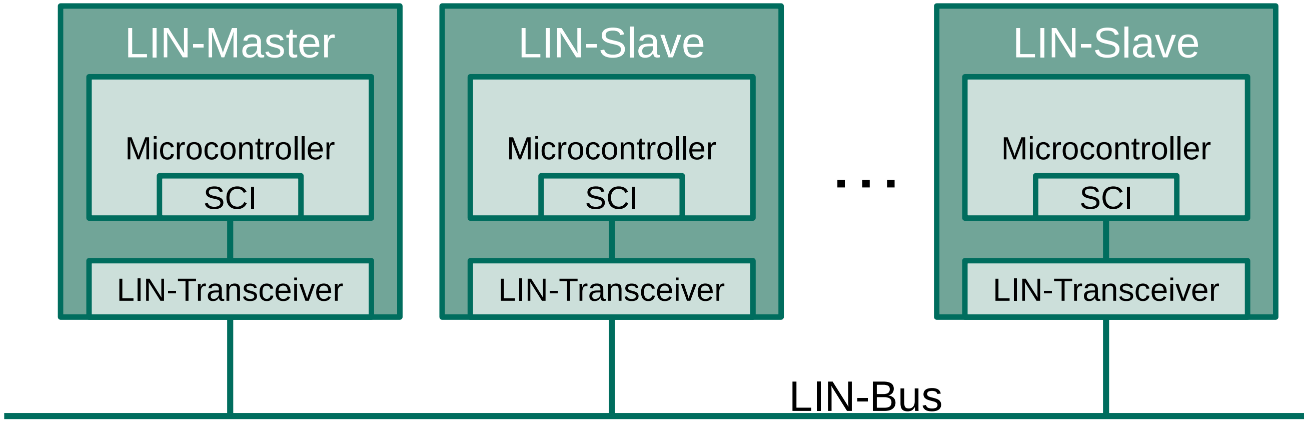
- Byte oriented communication
  - Byte is transmitted with least significant bit (LSB) first
  - Every byte is completed by a start and a stop bit (SCI-Frame)

- In principle, a LIN-Frame consists of multiple SCI-Frames



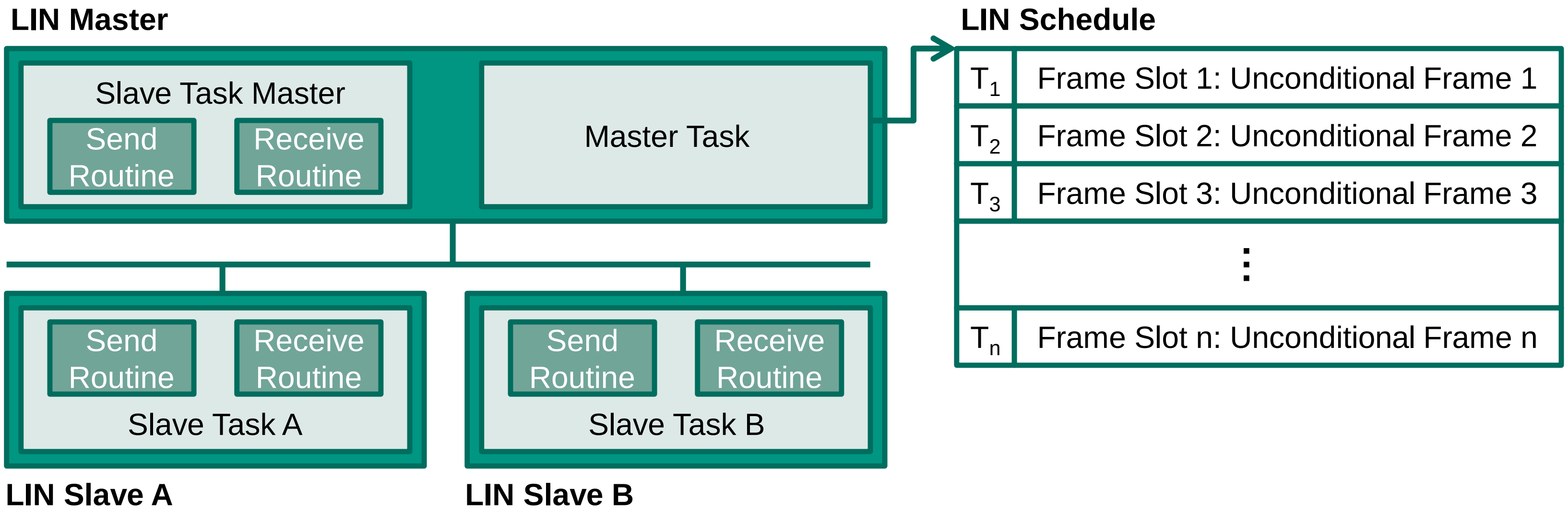
LIN Network

- A LIN cluster consists of several LIN nodes connected via a physical media (LIN-bus)
- No dedicated communication controller, microcontroller with Serial Communication Interface (SCI) is sufficient → low cost
- Physical connection to the bus is done in LIN-transceiver
  - Transformation of logical level to physical level on bus
  - Output driver
  - Receiver
  - Sleep and wakeup logic



LIN Communication Principle

- Master-Slave architecture
  - One single node is master (LIN Master) that coordinates all communication using cyclic polling
  - At least one (or more) slaves nodes (LIN Slave)
  - Each node in a cluster (including LIN master) has a slave task that handles communication
  - The LIN master has an additional master task to control the schedule of the communication

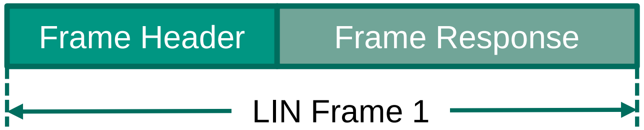


T_1	Frame Slot 1: Unconditional Frame 1
T_2	Frame Slot 2: Unconditional Frame 2
T_3	Frame Slot 3: Unconditional Frame 3
⋮	
T_n	Frame Slot n: Unconditional Frame n

Data communication

- Deterministic data communication by periodic execution of LIN schedule through LIN master

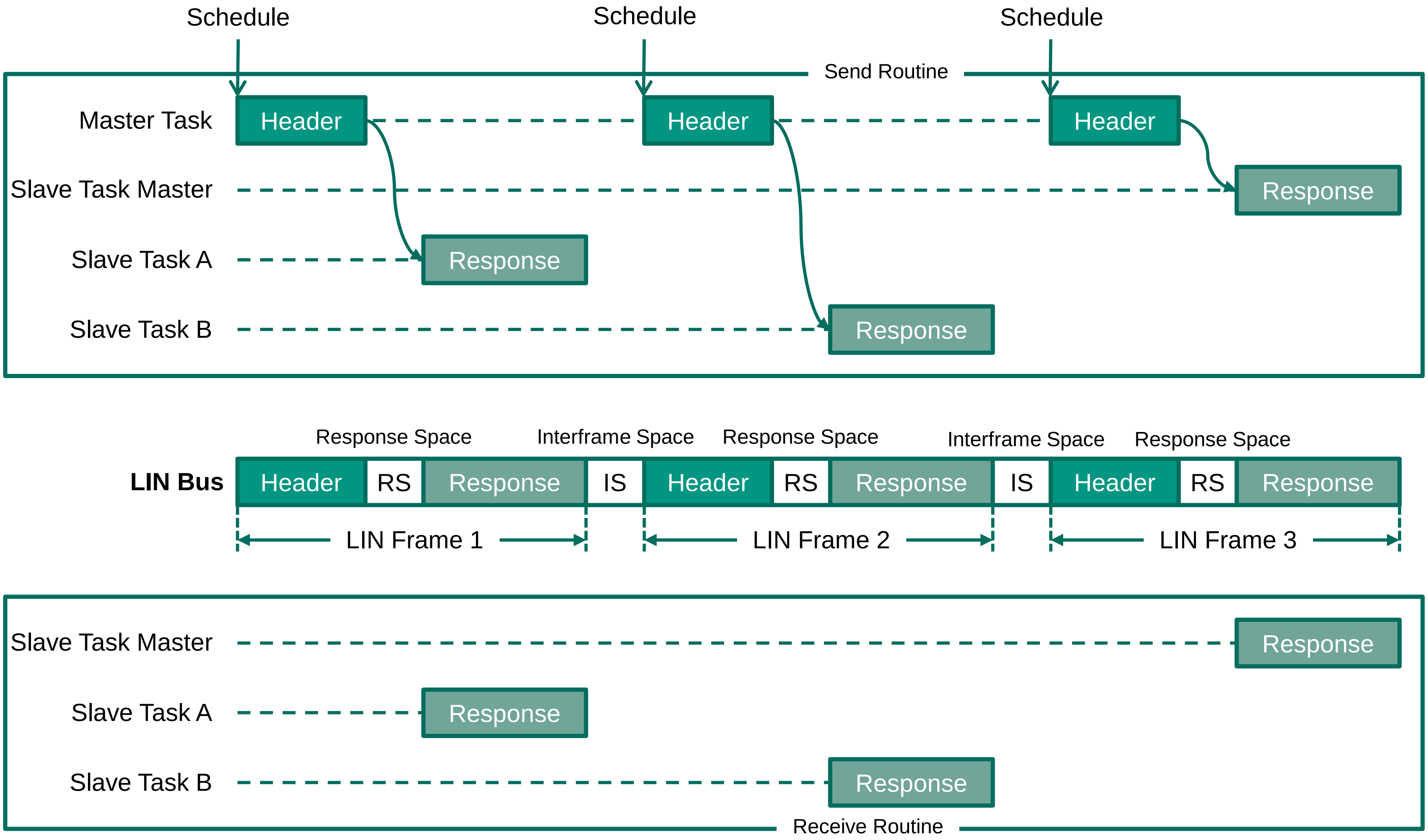
- Master starts transmission by sending a frame header
- A frame response is appended by a slave task to complete the frame



- Each frame response is available to be received by any other slave task

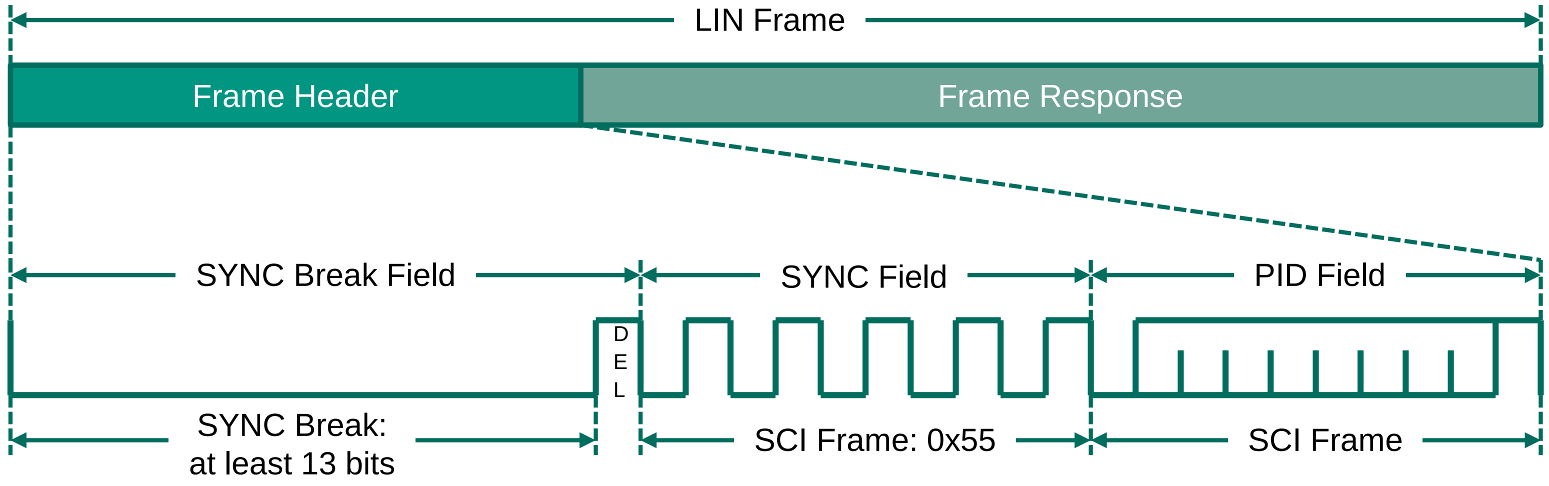
- LIN Master essentially just initiates and delegates communication in a LIN cluster → delegated token bus access

Example for LIN communication

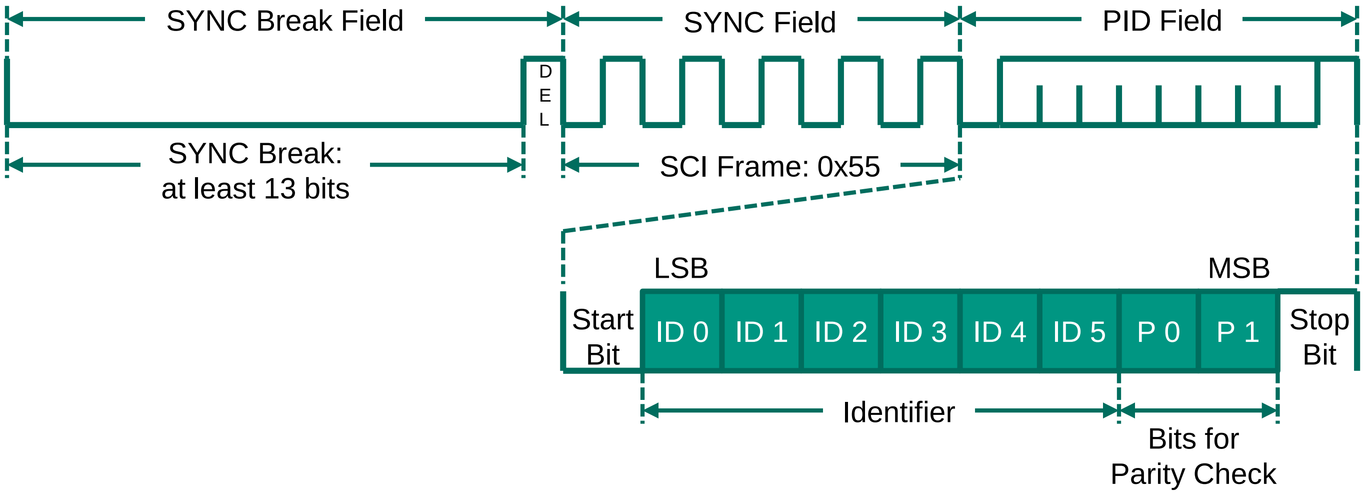


Frame Header

- Synchronization via sync break field and sync field
  - Sync break field denotes the start of a new message transmission
  - Sync field is used to synchronize the LIN slaves to the master’s transmission speed
- Protected Identifier (PID) determines the behavior of the LIN slaves



Protected Identifier (PID)



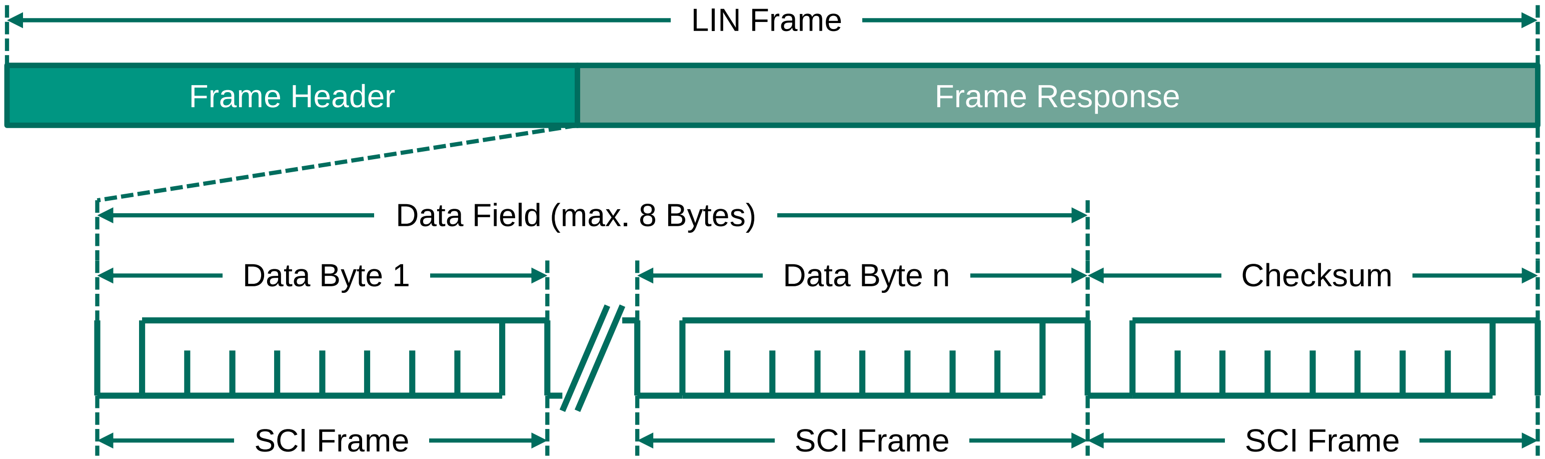
- Up to 64 identifiers may be allocated in a LIN cluster

Identifier	Usage
0 – 59	Message identifier
60 (0x3C), 61 (0x3D)	Configuration and diagnostics
62 (0x3E)	OEM-specific communication
63 (0x3F)	Reserved for future extensions

- Computation of parity bits:
  - P0 = ID0 ⊕ ID1 ⊕ ID2 ⊕ ID4
  - P1 = ¬(ID1 ⊕ ID3 ⊕ ID4 ⊕ ID5)

Frame Response

- Frame response is the answer of the LIN slave that has been addressed by the identifier in the frame header
  - Consists of 1 to 8 data bytes plus 1 byte checksum
  - Frame response is available to any LIN node (broadcast)
  - Least significant bit (LSB) is send first
  - Word transmission begins with low byte (little endian, Intel mode)



Checksum

- Data protection via checksum
  - Classic checksum: only protects user data
  - Enhanced checksum: protects protected identifier as well as user data (used in LIN 2.x for identifiers 0 to 59)
  - Diagnostic frames are always protected using classic checksum
- Checksum calculation:
  - Inverted eight bit sum with carry over all bytes

    Checksum = INV(data byte 1 $\oplus$ data byte 2 $\oplus$ … $\oplus$ data byte n)
- Less powerful than block coding, hamming code or CRC coding

Error detection

- Error detection mechanisms
  - Bit monitoring
    - Compare send bit with actual bus level
  - Checksum
    - Check arriving data bytes
  - Parity Check
    - Two parity bits to protect the identifier
  - Slave Responding Check
    - Responding frame has to be send directly after a frame header
  - Sync Field Check
    - Check whether the data transmission clock can be deducted

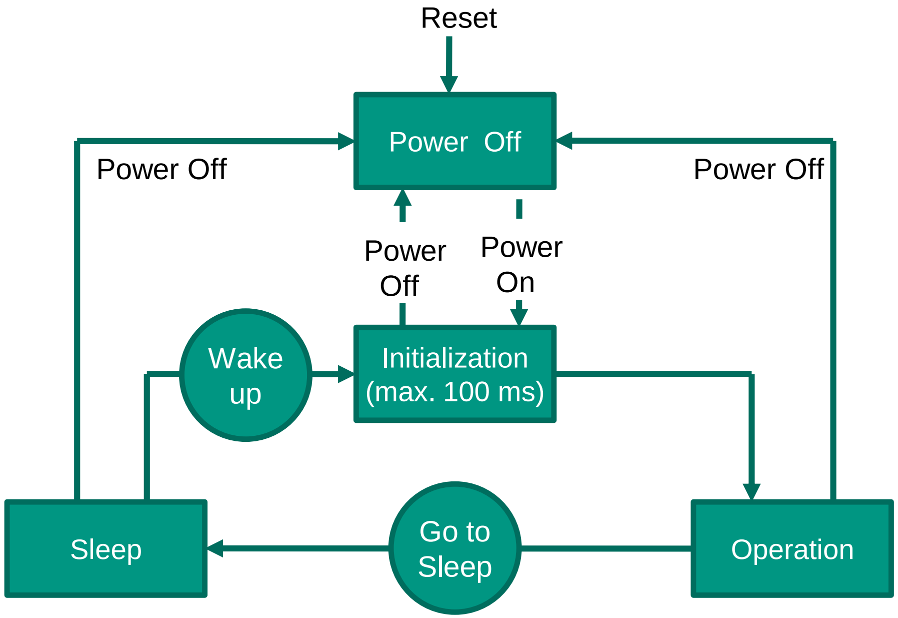
- No error handling described in LIN specification!

LIN Network Management

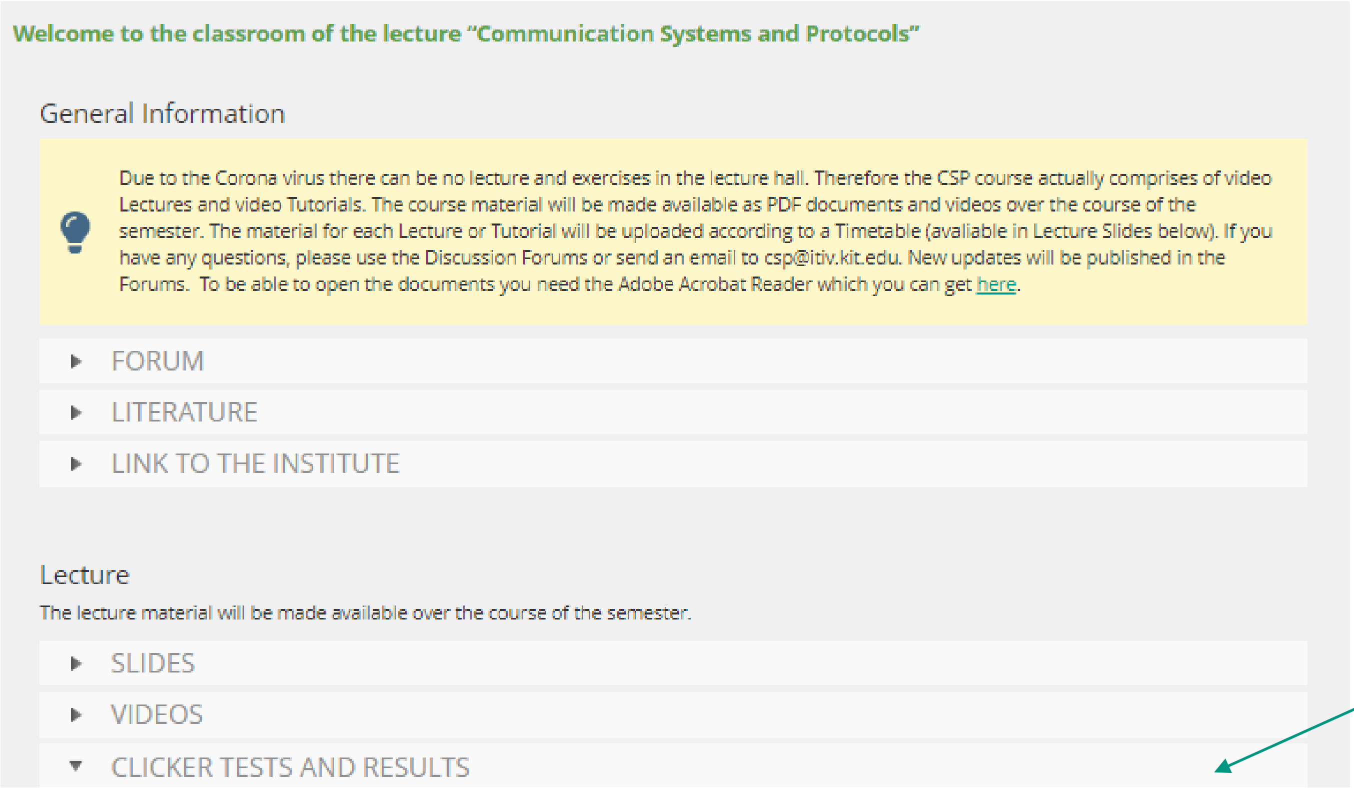
- On Power-on, a LIN switches to initialization phase
  - After having initialized (max. 100 ms), the LIN slave autonomously switches to the operating state

- By transmitting a so called go to sleep command, the LIN master can put all LIN slaves in a cluster to sleep mode

- All nodes can wake up a cluster by sending a wake-up signal on the LIN bus
  - Dominant pulse that lasts between 250µs and 5ms



Participate in Clicker Test 9b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Karlsruher Institut für Technologie

Thank you for your attention

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Session 26: FlexRay

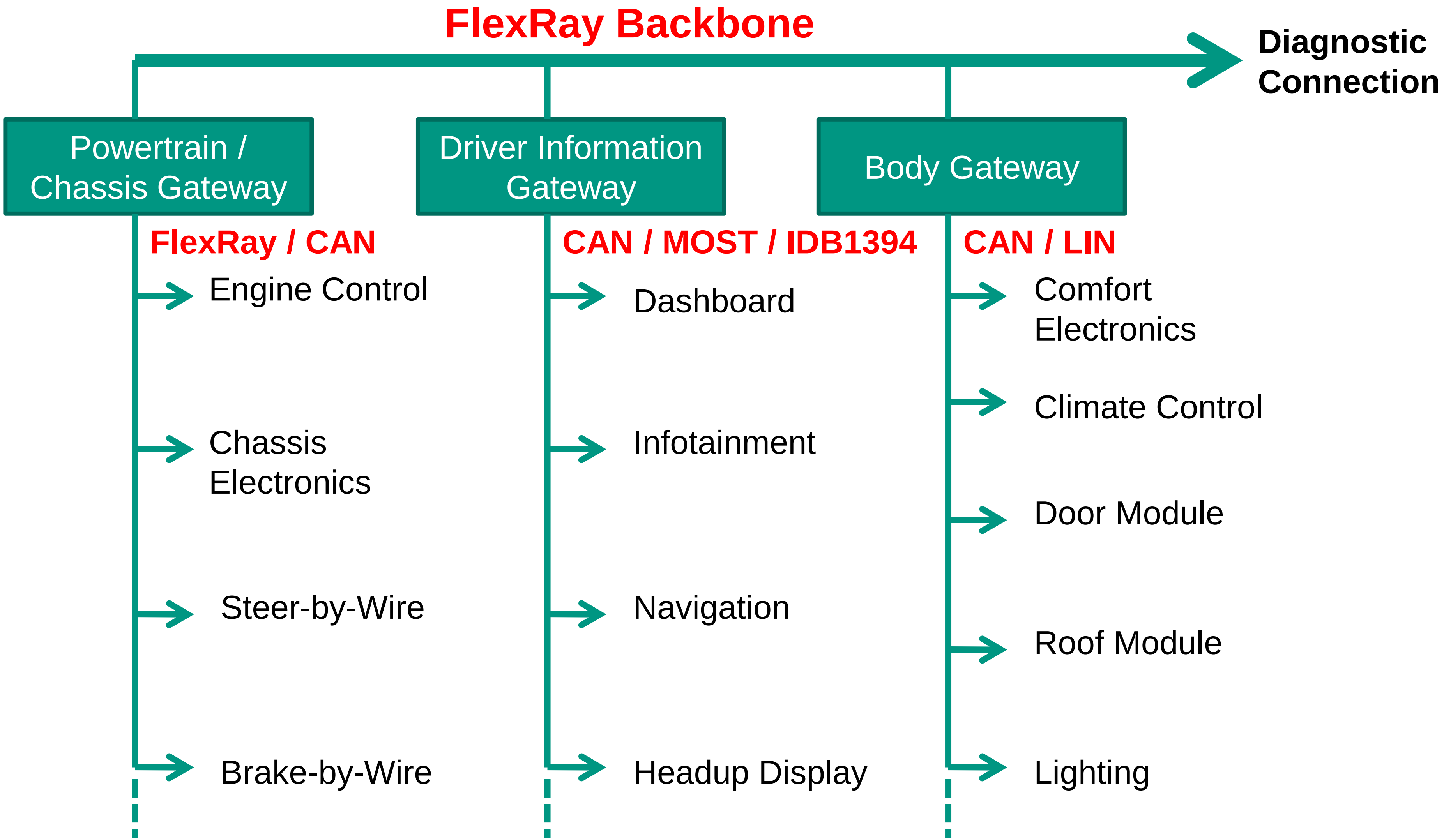
FlexRay

- Consortium founded in 1999
- Motivation: increasing requirements concerning bandwidth and transmission safety within automobiles
- Goal: Development of deterministic, fault tolerant communication system with 10 MBit/s bandwidth
- Basis for X-by-Wire systems
- Basically the protocol can be used for optical as well as electrical media.

Basic features

- Multi-master system
- Beside basic star like structure also line and bus structures possible
- Allows operation in single channel mode with (10 MBit/s) or double channel mode (20 MBit/s)
  - Single channel mode to increase reliability. The second channel is used as a redundancy
  - Double channel mode to raise throughput but less reliability
- Mechanisms to increase fault tolerance
  - Deactivation of sub-networks in the case of an error
- Deterministic message transmission by using communication cycles
  - Communication cycle is flexibly configurable
  - Global time base for synchronization
- Physical layer: Differential signalization on a twisted pair
- In contrast to CAN „logic 1“ and „logic 0“ are equitable

Possible Application of FlexRay



FlexRay Backbone

Diagnostic Connection

Powertrain / Chassis Gateway

FlexRay / CAN

- Engine Control
- Chassis Electronics
- Steer-by-Wire
- Brake-by-Wire

Driver Information Gateway

CAN / MOST / IDB1394

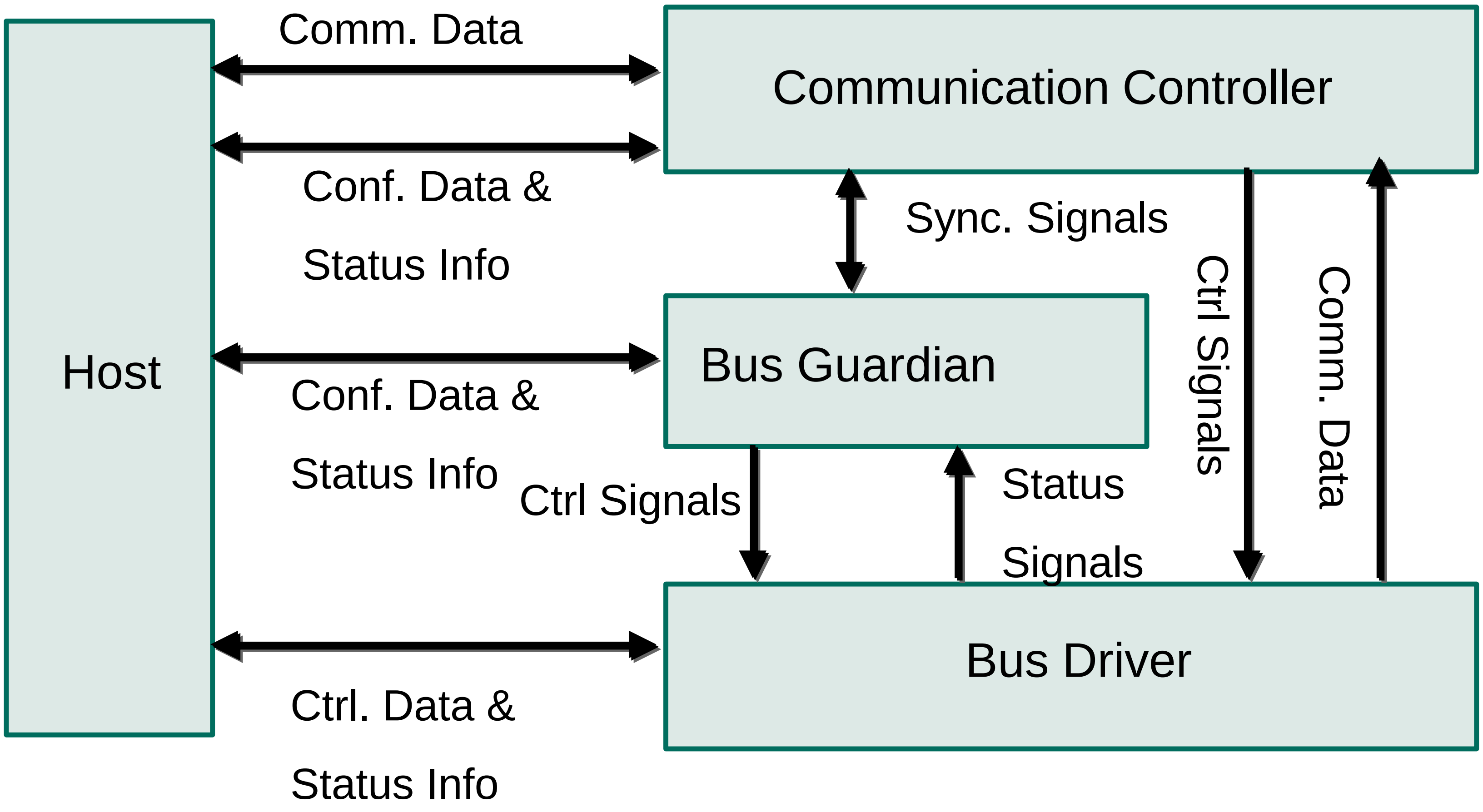
- Dashboard
- Infotainment
- Navigation
- Headup Display

Body Gateway

CAN / LIN

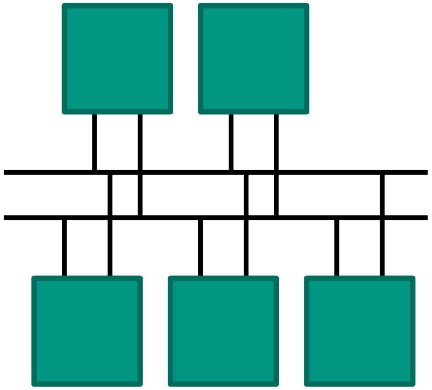
- Comfort Electronics
- Climate Control
- Door Module
- Roof Module
- Lighting

System Structure of a FlexRay Node

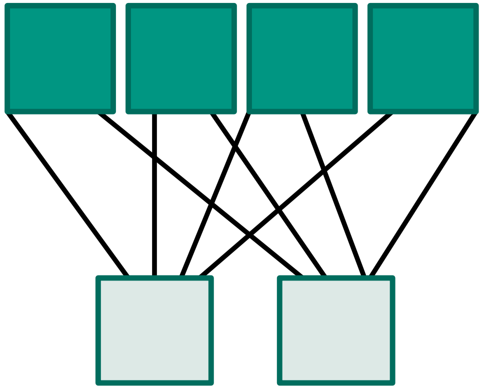


Possible Topologies (Examples)

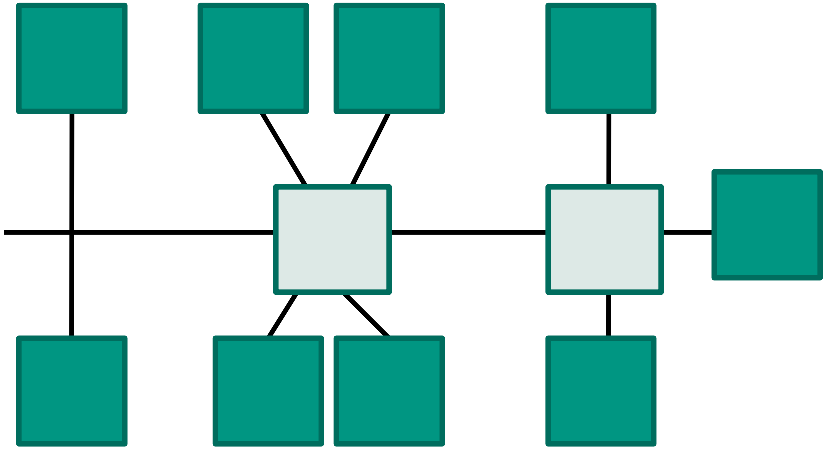
- Double channel bus topology (A)
- Double channel active star topology (B)
  - Point-to-point connections
  - Star coupler actively drives the signals to the connected nodes
  - Without such repeaters a FlexRay segment is not allowed to be longer than 24 m
- Hybrid topology (C)
  - Combination of bus and star



(A)

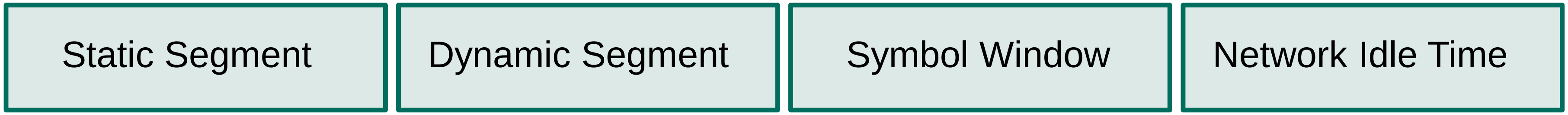


(B)



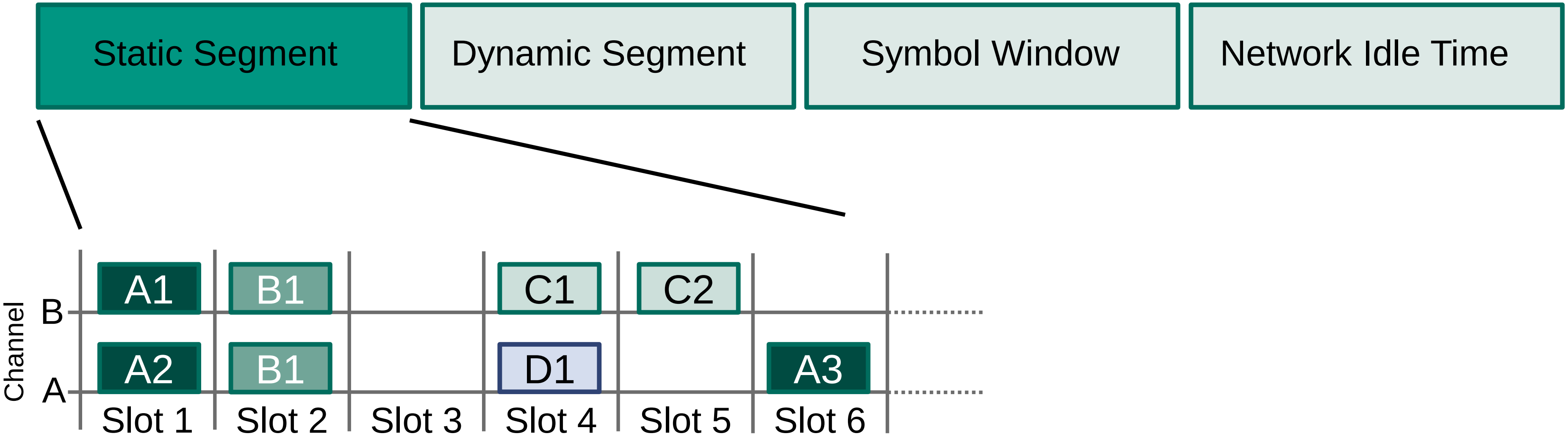
(C)

FlexRay Communication Cycle



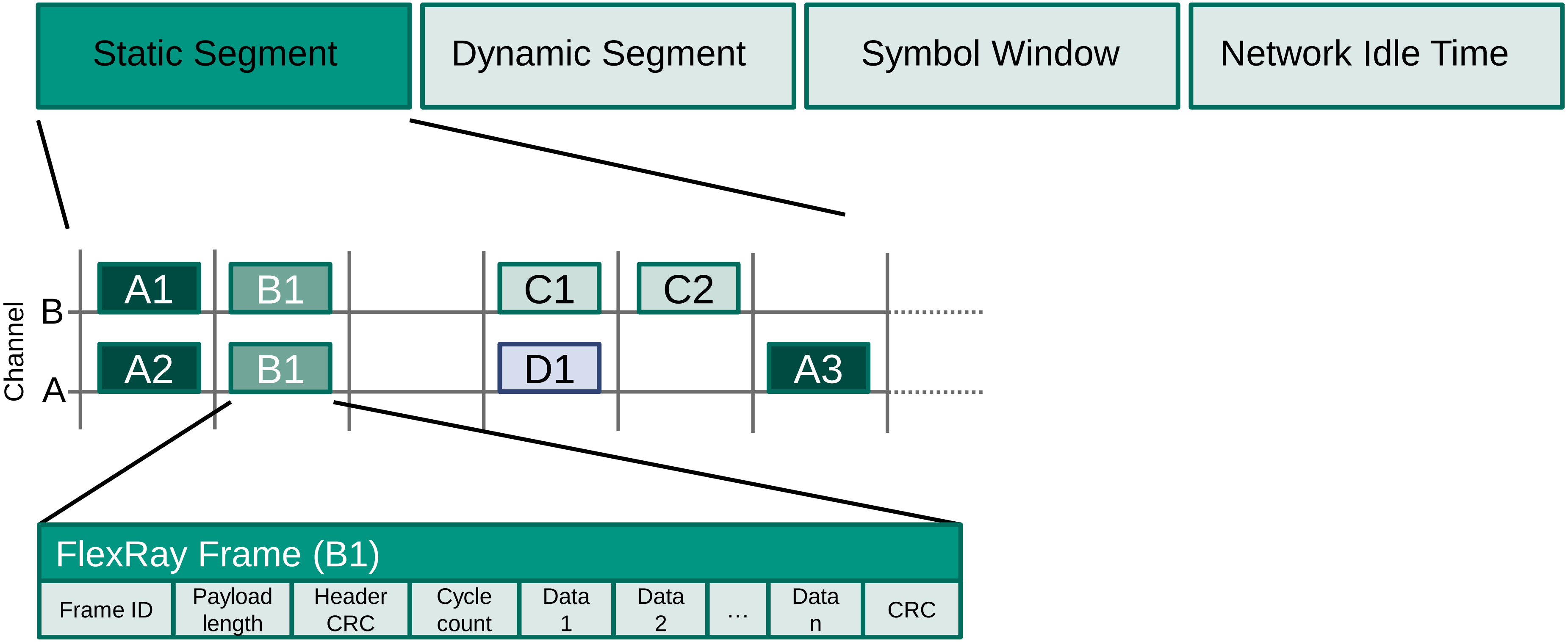
- Frames are transmitted within a communication cycle
- The communication cycle allows deterministic as well as non deterministic transmission. It is divided into
  - Static segment
  - Dynamic segment (optional)
  - Symbol window (optional)
  - Network idle time
- The cycle can be adapted to different use cases (e.g. pure TDMA or pure event based communication)

Static Segment (1)



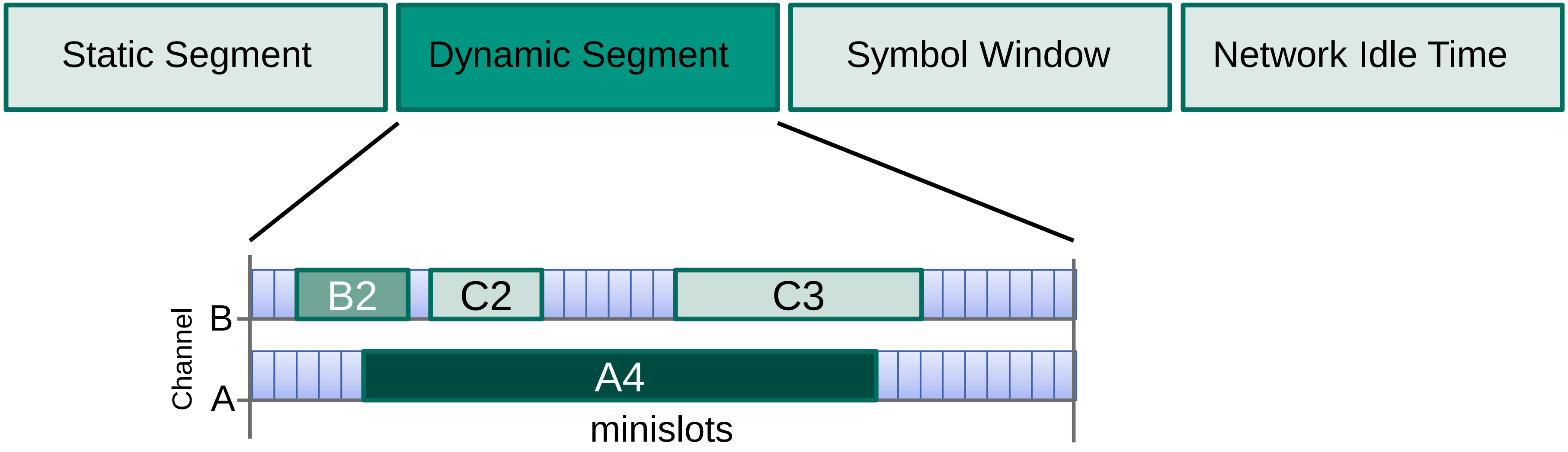
- Nodes are assigned to fix time windows in which frame transmission can be performed
- Bus access within the static segment via TDMA
  - Realization of realtime capability within this segment (deterministic)
  - Every message within this segment is transmitted at defined points in time

Static Segment (2)



- Each frame is assigned to slots in which only that frame is allowed to be send
- If using two channels, frames can be transmitted redundantly

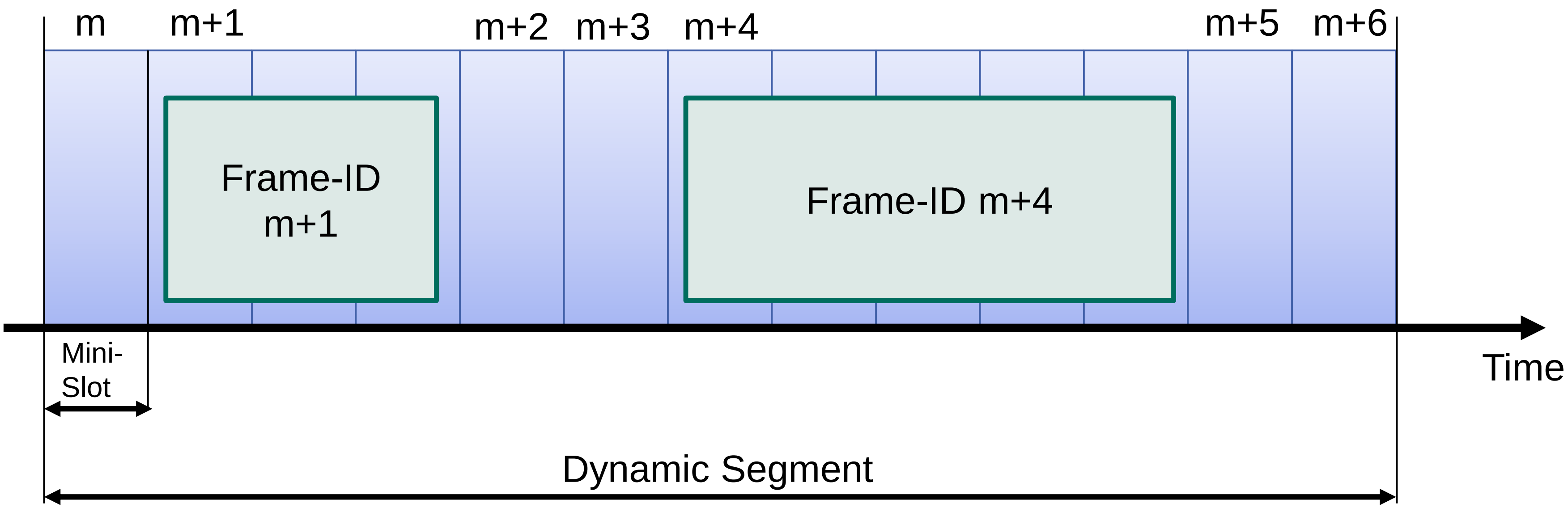
Dynamic Segment



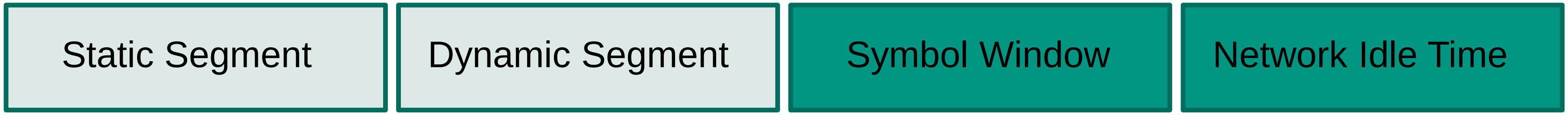
- For frames with lower requirements concerning predictable latency
- Bus access via FTDMA (also called Mini-Slotting)
- Used for event based transmission: Event based data to be transmitted does not depend on a static slot

Mini-Slotting (FTDMA)

- Prioritized bus access within the dynamic segment controlled with a **slot count**
  - If a Minislot is not used the slot counter is incremented and the Minislot elapses unused
  - If the actual slot count corresponds to the Slot-ID of the frame, the frame occupies a certain number of Minislots. After the next free Minislot the slot counter is immediately incremented
- Space within the dynamic segment can be used efficiently that way



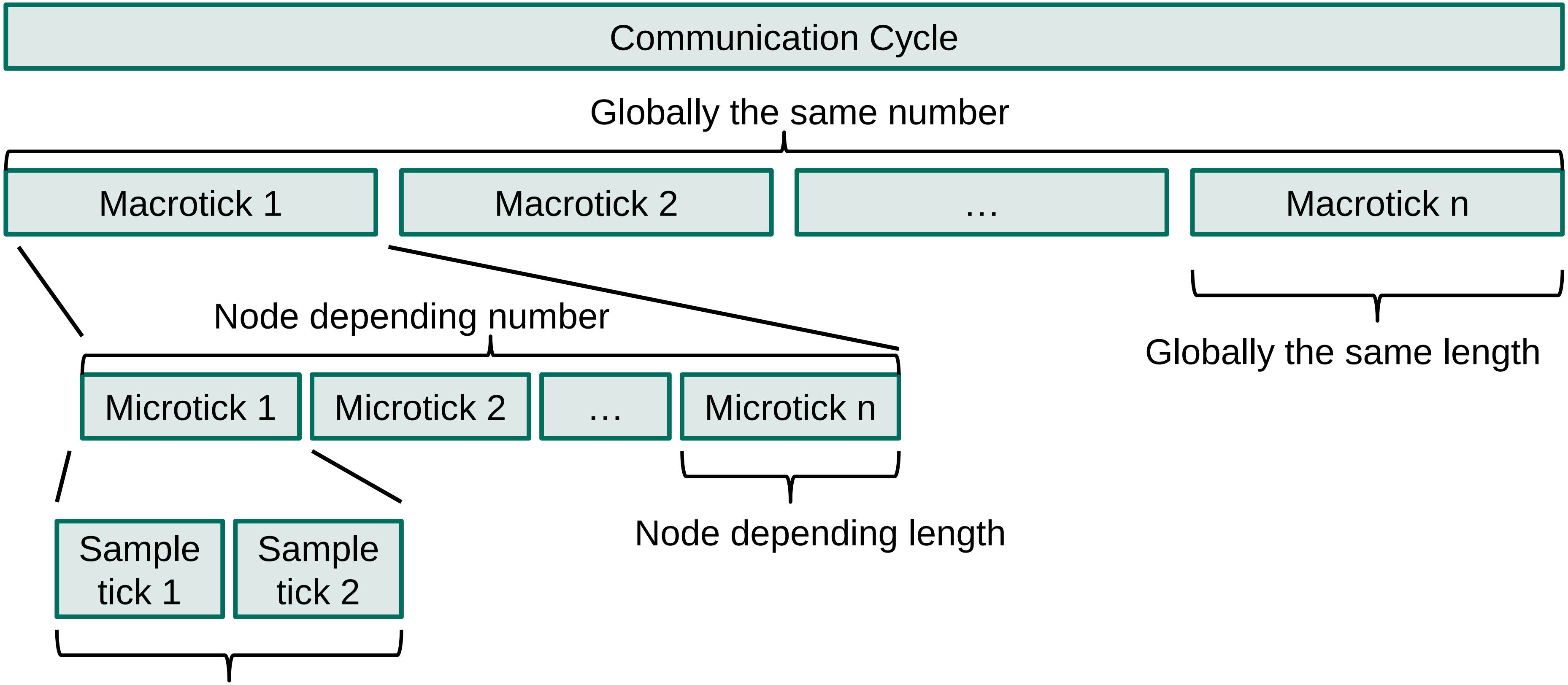
Symbol Window and Network Idle Time



- Symbol Window
  - Transmission of Flexray internal control information
  - If the system is started up the nodes wake up each other by transmitting special symbols within the symbol window
- Network Idle Time
  - Necessary for time synchronization
  - During the NIT the nodes can compensate their individual „speed“ of their oscillators and can agree to a unique system time

Time Synchronization

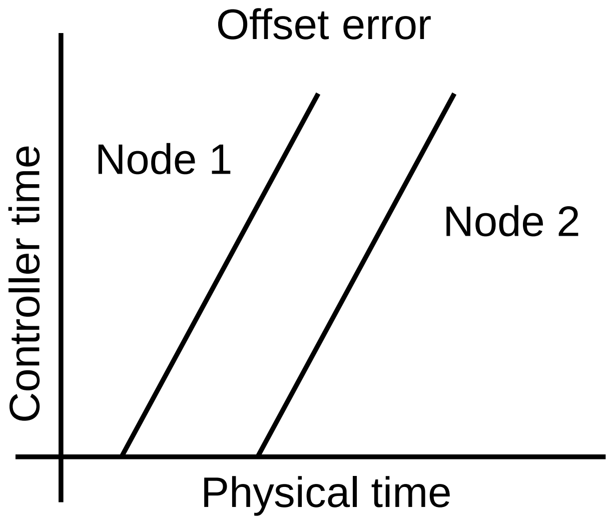
- Hierachically organized time system
- No central clock available
- Each node has a local time. The global view of time must be determined by exchange and adjustment of time information

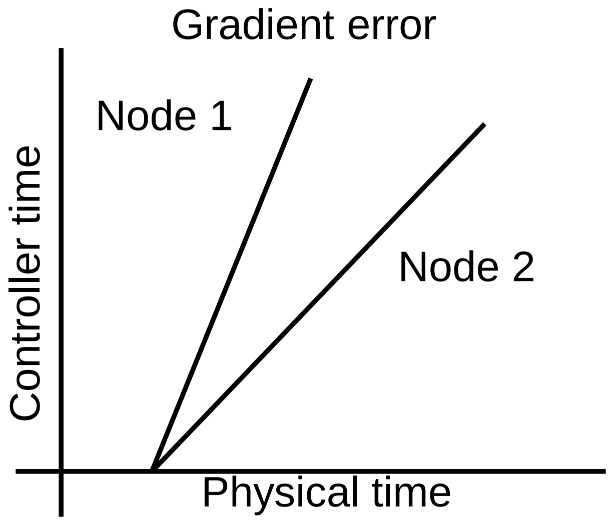


Each microtick consists typically of 2 sampleticks of about 0.0125us length

Timing Errors

- Differences in the local time bases of the nodes find expression in the form of **offset error** and **gradient error**
- Offset error: Can be determined by a one-time measurement within the static segment, it describes the absolute deviation between two nodes
- Gradient error: Indicates how two nodes timely depart from each other per cycle and requires two measurements
- Within the FlexRay protocol a combination of offset and gradient error correction is applied





Offset Error Correction

- The nodes measure the cycle length with the help of the time stamps of two consecutive cycles and compare it to their own cycle length
- Again the difference is calculated
- Deviation is determined by the FTM (Fault Tolerant Midpoint) algorithm
- The offset error (e.g. given by run time delays) is eliminated by modifying the number of microticks at the end of each cycle (during the network idle time

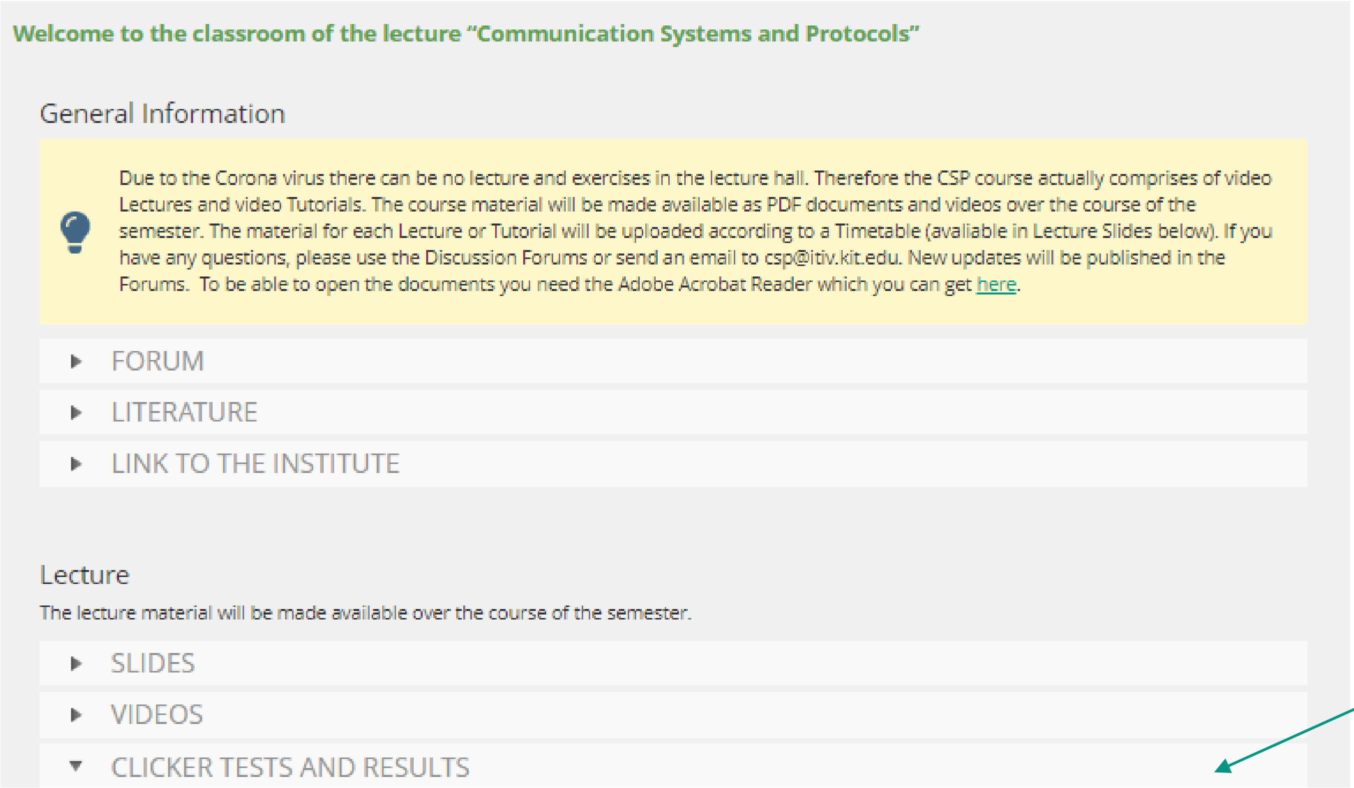
Gradient Error Correction

- The determination of the correction values is done by special sync nodes which are a subset of all available nodes
- Gradient error correction
  - Every sync node transmits in every cycle in one of its static slots a sync frame which is a normal data frame with the sync bit set
  - All nodes (including the sync nodes) compare the actual point in time of reception of each sync frame with the expected reception time and so get a list of differences with which they determine the offset correction by using a special algorithm (FTM)
- The gradient correction then adds or removes microticks per macrotick to get the same cycle duration within all nodes

Precision

- Precision means the maximum deviation in time of two arbitrary nodes despite the usage of synchronization
- Has great influence on the data rate
- The higher the **difference P** of the clocks of two nodes the smaller is the window in which data can be transmitted
- Additionally to difference P there is the **signal runtime D** on the lanes, transceivers and star couplers.
- In the worst case when the sender is maximum retarded and when having maximum signal delay the receiver must be able to perceive the frame completely within the appropriate slot
- The **Network Idle Time** therefore has to be configured as **P+D as a minimum**

Participate in Clicker Test 9c



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Session 27: Networks - LANs

Definition: Network

An arrangement of intersecting horizontal and vertical lines: *a spider constructs a complex network of several different kinds of threads*

A number of interconnected computers, machines, or operations: *a computer network*

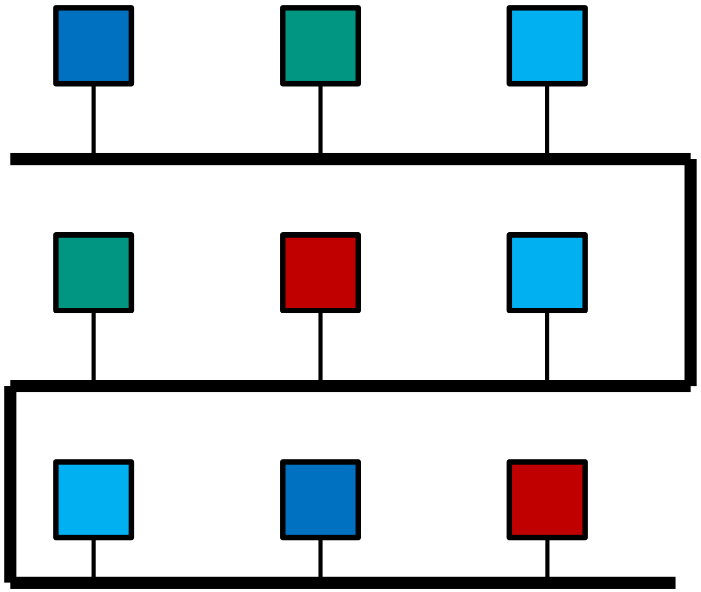
1. Computers: A group of interconnected (via cable and/or wireless) computers and peripherals that is capable of sharing software and hardware resources between many users. The Internet is a global network of networks. See also local area network and wide area network.

2. Communications: A system that enables users of telephones or data communications lines to exchange information over long distances by connecting with each other through a system of routers, servers, switches, and the like.

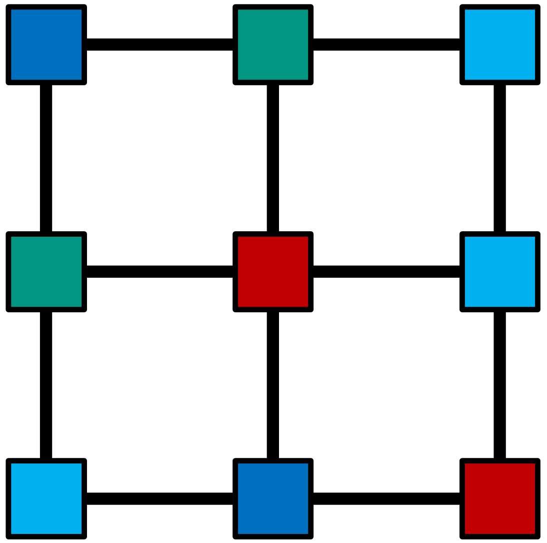
Sources: Oxford Dictionaries
http://www.businessdictionary.com/

Benefits of Networks

System Setup using a bus



System Setup using a Network



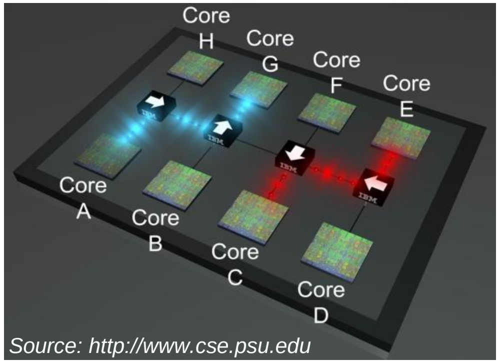
- Less and shorter communication lines
- Multiple connections at the same time possible
- Very flexible
- Efficient use of bandwidth

Classification

Classification of networks by coverage area

- **WAN** (Wide Area Network)
  - spans over large geographical areas
  - carries aggregated global network traffic.
- **MAN** (Metropolitan Area Network)
  - Coverage up to 60 km
  - serves as access point to the global WAN
- **LAN** (Local Area Network)
  - Coverage up to 900 m
  - typically privately owned smaller networks
- **PAN** (Personal Area Network)
  - Coverage of ~10 m
  - network of personal items
- **NoC (Network on Chip)**
  - **Coverage on chip (some millimeters)**
  - **connect computational units on a single chip**

Source: http://www.techstagram.com



Source: *http://www.cse.psu.edu*

OSI Model, TCP/IP, Ethernet

Overview

- Communication Models (OSI-Layer Model)
- Local Area Networks (LANs)
  - Couplers
  - Example : Ethernet
  - TCP/IP

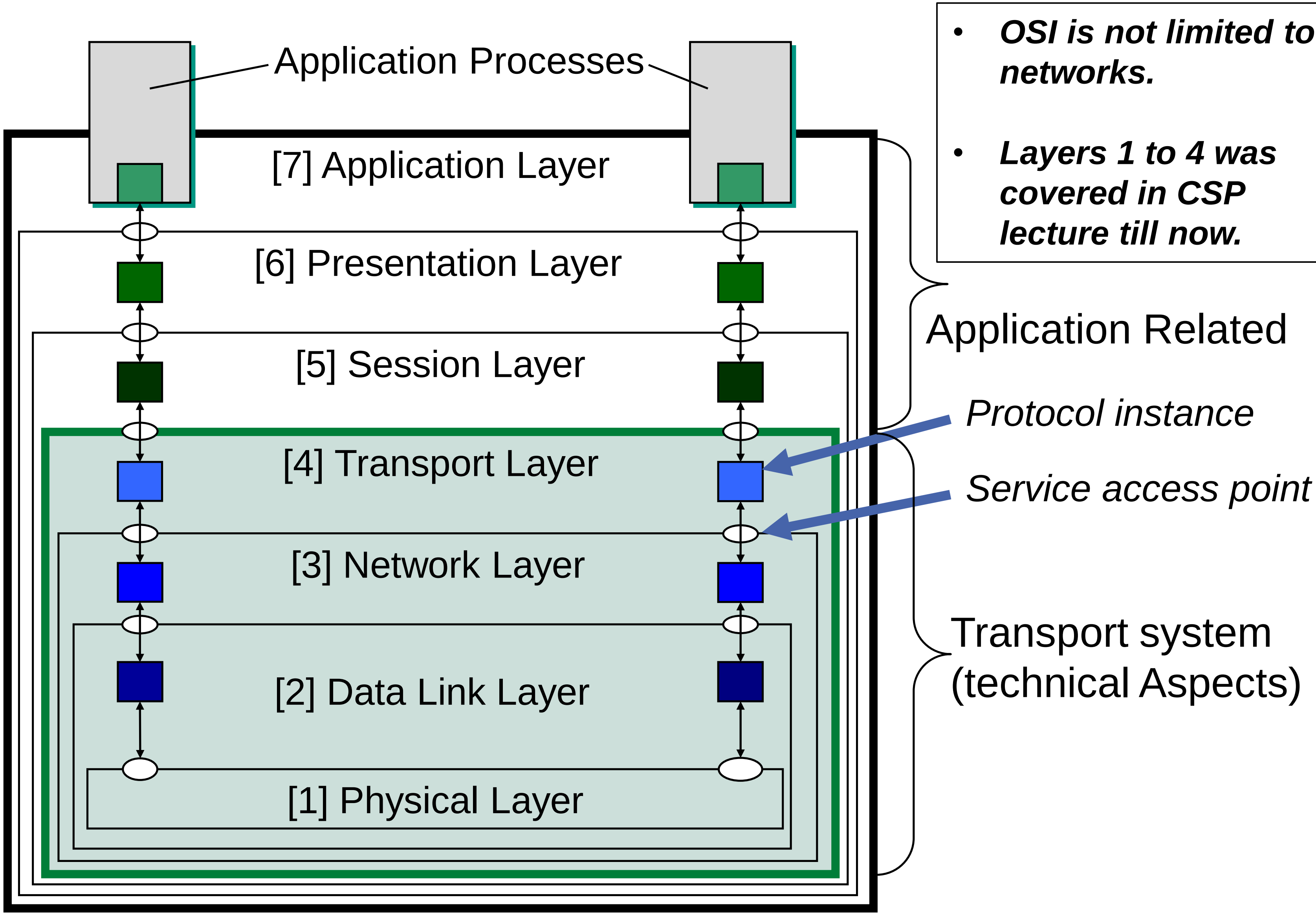
For more in-depth information visit the lecture „Telematik“ (Prof. Dr. Martina Zitterbart)

History of Communication Models

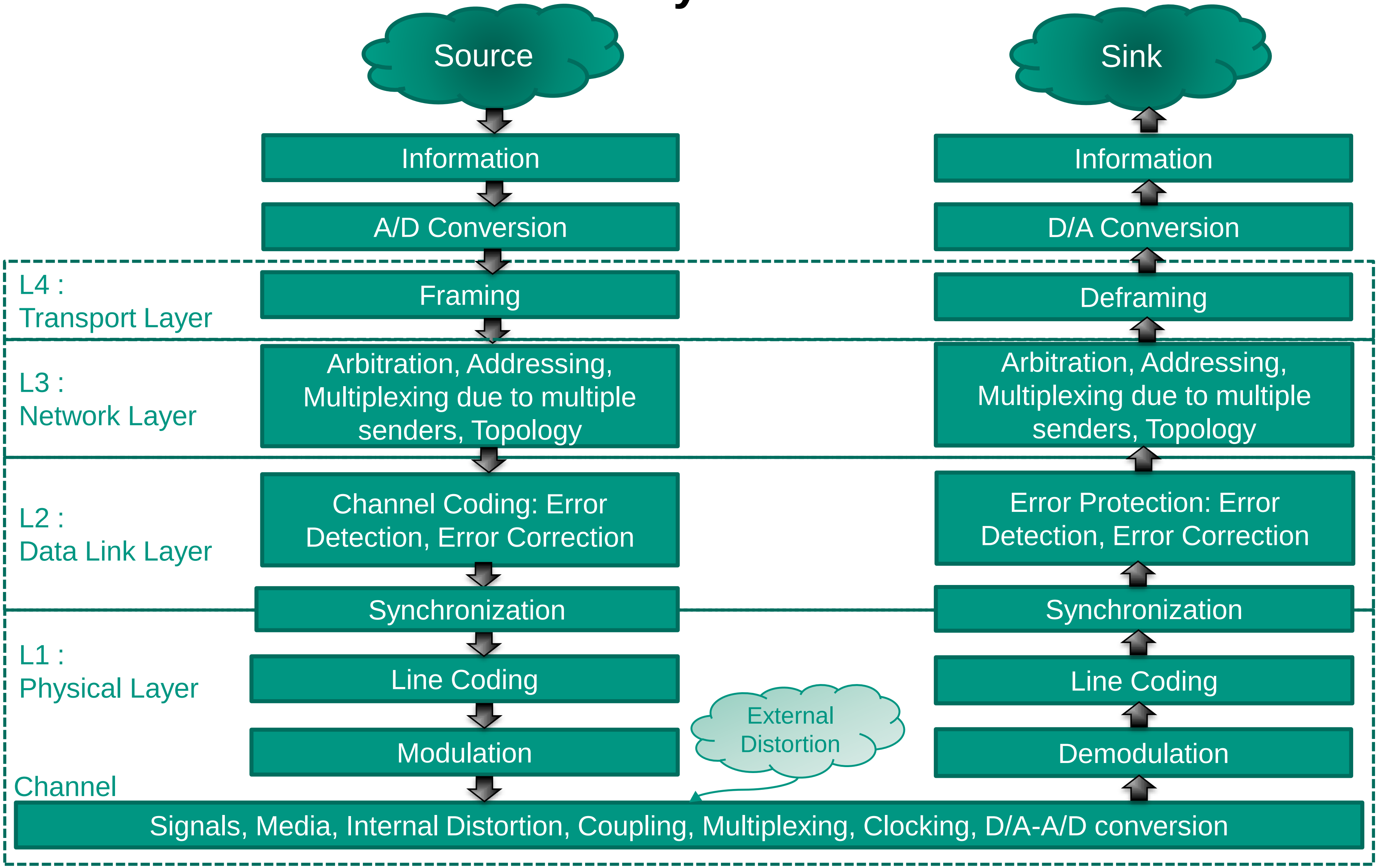
- First computer networks were vendor specific and closed networks
  - Today: mainly open networks
- Most important Wide Area Network (WAN):
  - 1969: ARPA-Net (Advanced Research Projects Agency)
  - Successor: Internet
- Computer communication requires mutual understanding of how the communication is supposed to function
  - Protocols
- All current communication architectures: Layer models
  - OSI-Reference Model, TCP/IP-Protocols
  - Every layer implements certain functions
  - Independency of layers due to standardized interfaces
  - Dedicated mapping of functionalities and services to certain layers.
- Most important reference model for computer communication published by ISO 1984 following the preliminary work of CCITT
  - OSI-7-layer model (Basic Reference Model for Open Systems Interconnection)
    - ISO 7498-Standard, later adapted by CCITT as X.200-standard)

Review
Logic Architecture: OSI Reference Model

- ***OSI is not limited to networks.***
- ***Layers 1 to 4 was covered in CSP lecture till now.***

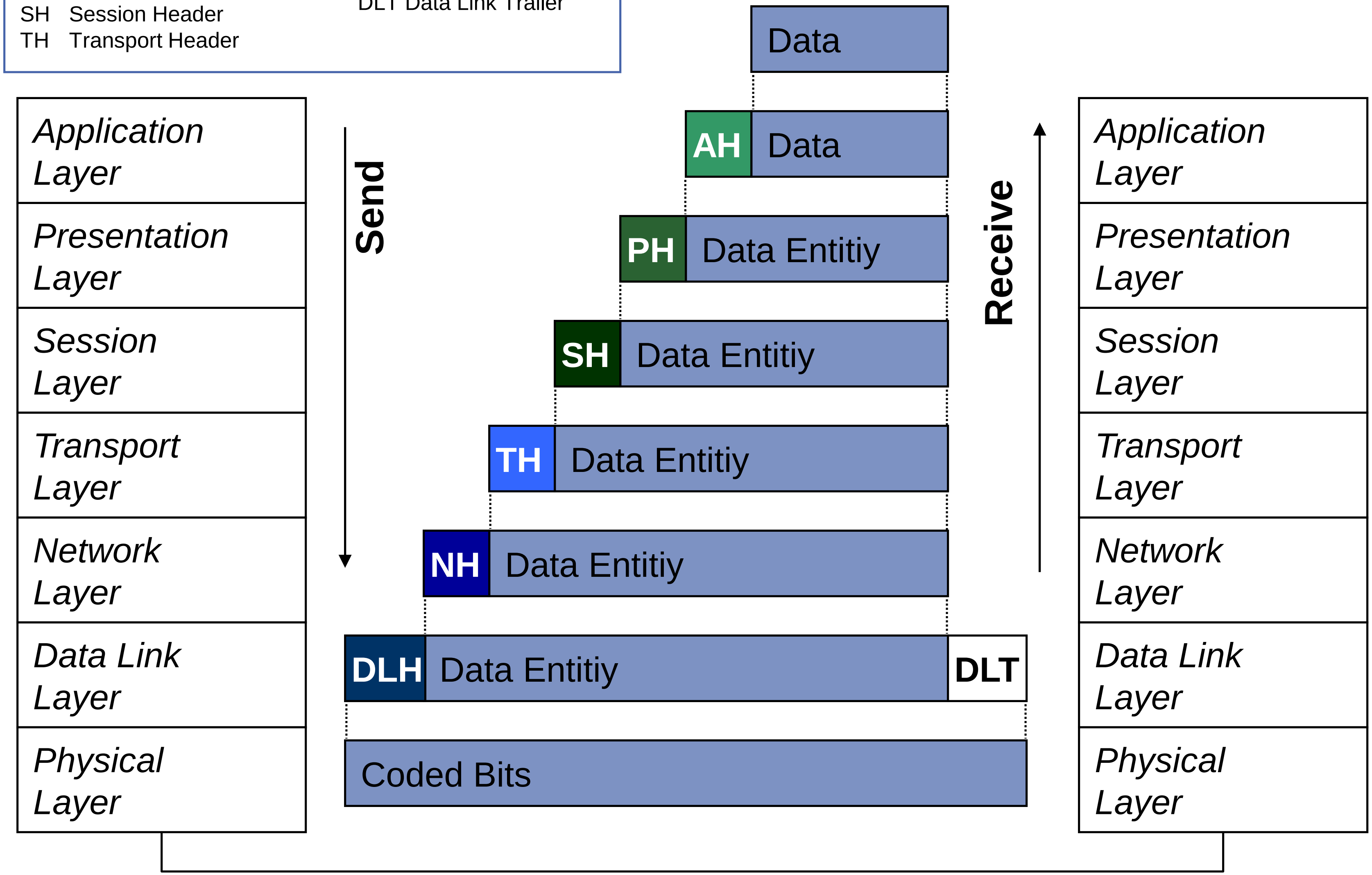


Technical Communication System


Source
Sink
Information
Information
A/D Conversion
D/A Conversion
L4 :
Transport Layer
Framing
Deframing
L3 :
Network Layer
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
Arbitration, Addressing, Multiplexing due to multiple senders, Topology
L2 :
Data Link Layer
Channel Coding: Error Detection, Error Correction
Error Protection: Error Detection, Error Correction
Synchronization
Synchronization
L1 :
Physical Layer
Line Coding
Line Coding
External Distortion
Modulation
Demodulation
Channel
Signals, Media, Internal Distortion, Coupling, Multiplexing, Clocking, D/A-A/D conversion

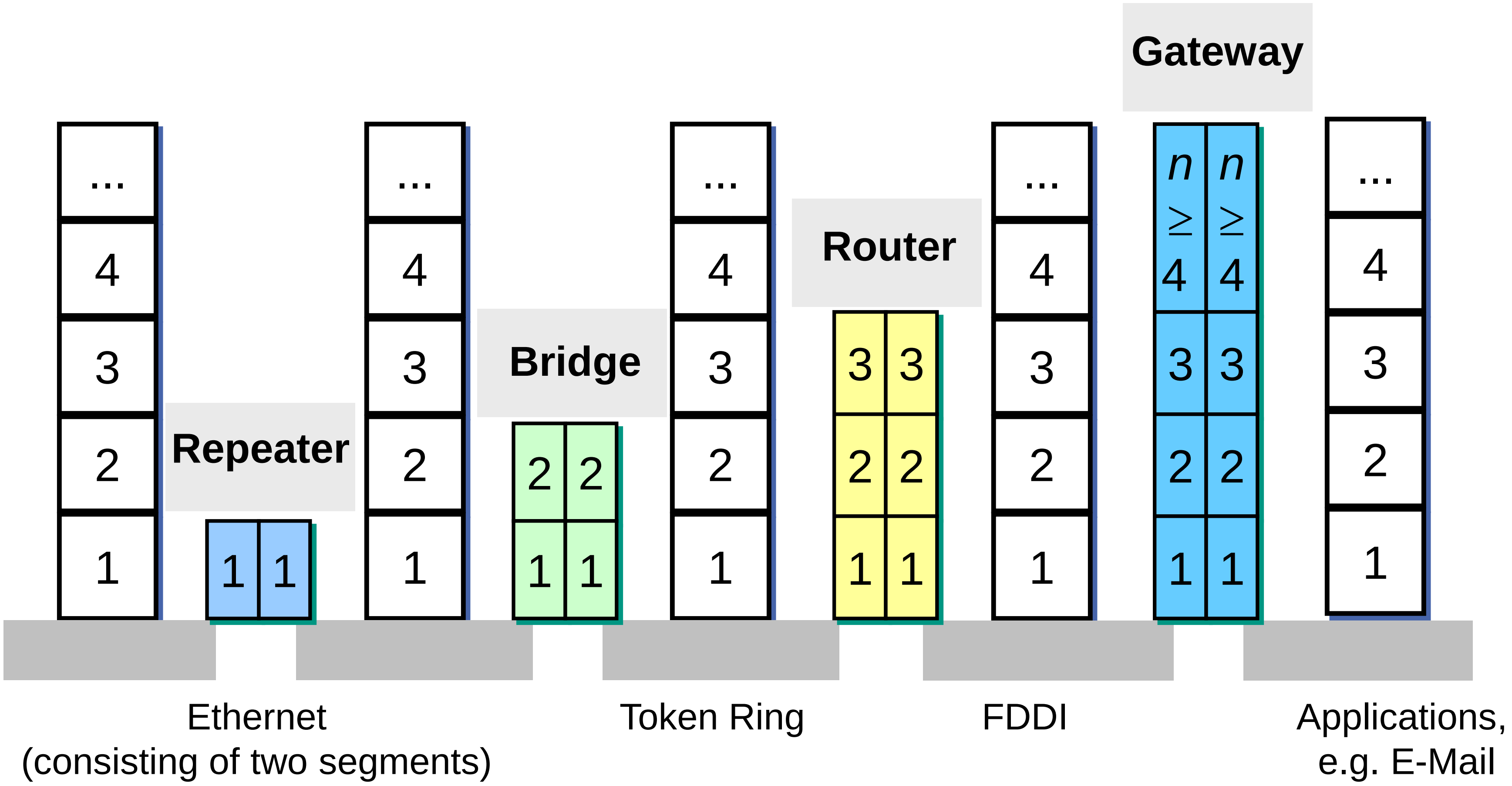
Data Encapsulation

AH	Application Header	NH	Network Header
PH	Presentation Header	DLH	Data Link Header
SH	Session Header	DLT	Data Link Trailer
TH	Transport Header		



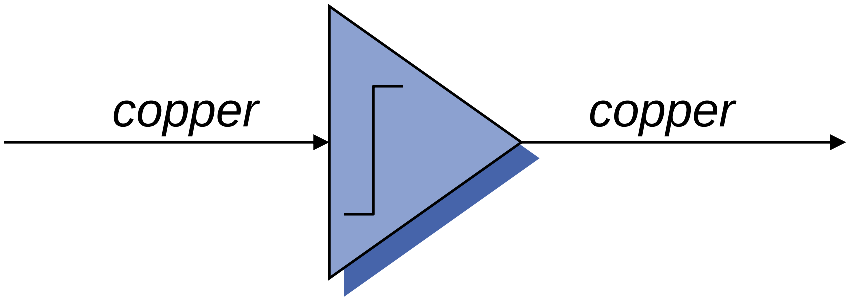
Network Coupling

Coupling of Networks– Internetworking

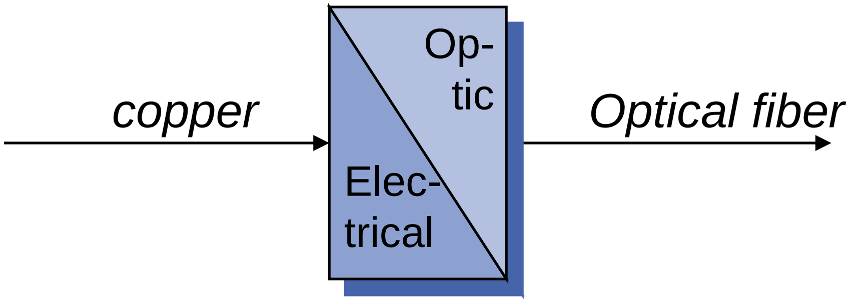


Repeater

- Coupling of physical media using signal regeneration / -amplification
  - No storage of intermediate data (buffering)
  - No packet processing
  - Media can differ, but protocol on layer 2 has to be equal



Electrical Repeater (Amplifier)



Electro-optical Repeater

Repeater – Characteristics

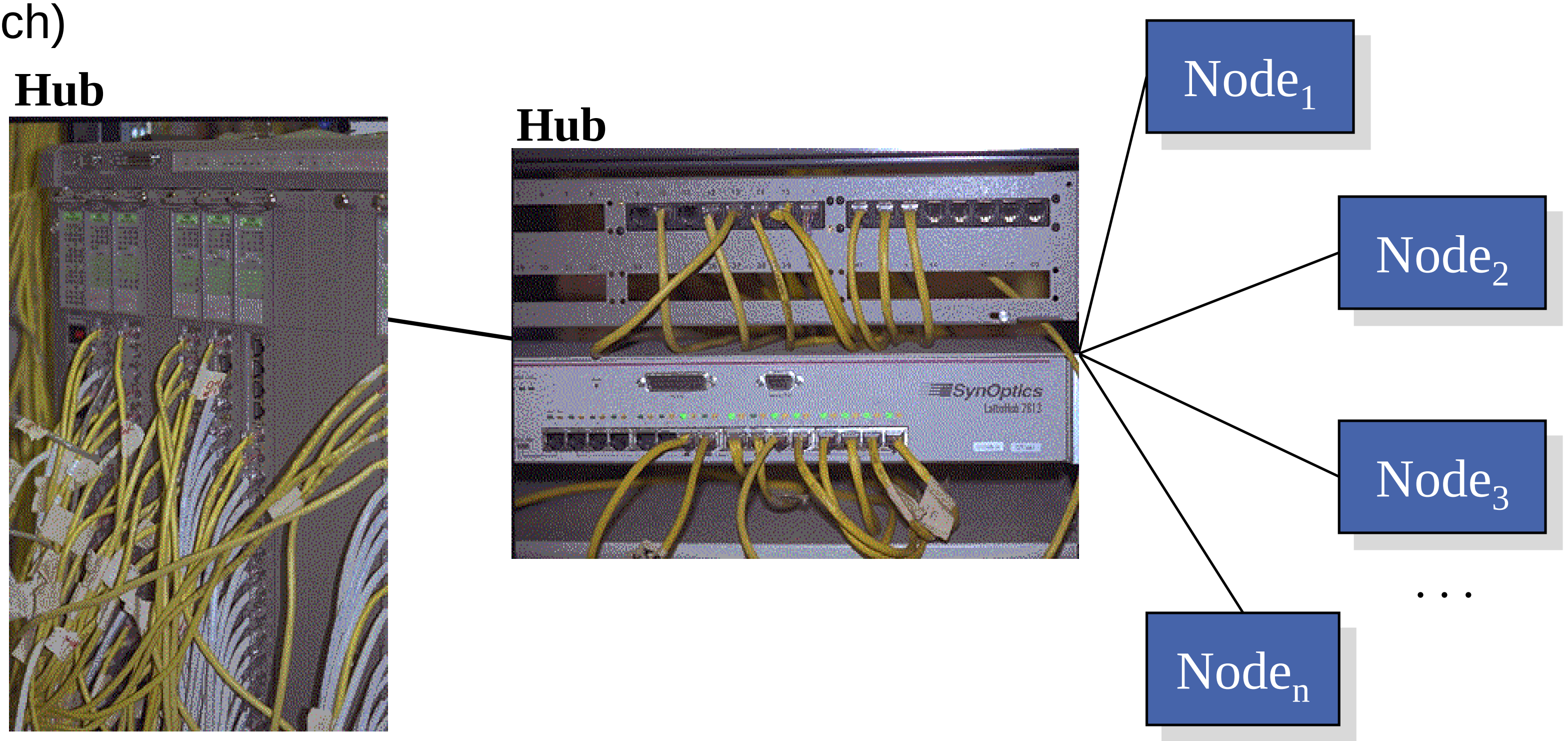
- Network coupling on layer 1

- Field of application
  - Coupling of local networks to increase physical length of network segment
  - Generation of multiple outgoing signals in branching points
  - Change of transmission media

- + Simple, cheap technology
- + No performance penalties because of data processing
- + Very long communication links can be realized (e.g. transoceanic cables)
- − No intelligence; all data is passed on
- − No increase in network capacity because of partitioning

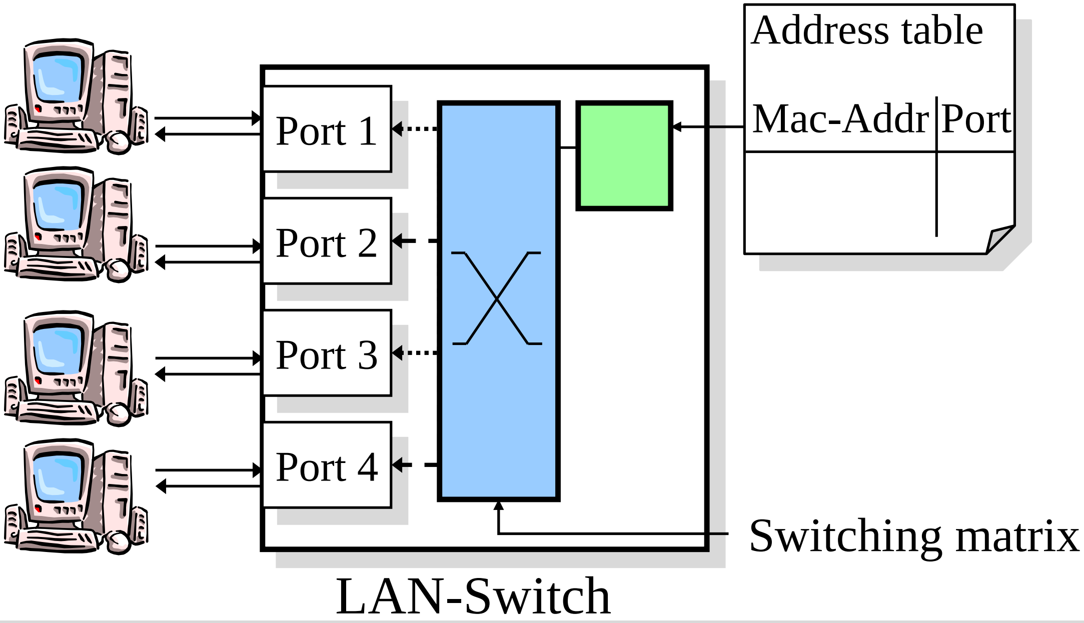
Hubs

- Nodes are connected to a hub using star topology
- Hubs provide similar functionality as repeaters (layer 1)
- Hubs can be cascaded
- Packets are forwarded to all connected nodes
  - ⇨ Data throughput of the whole network is not increased (compared to switch)



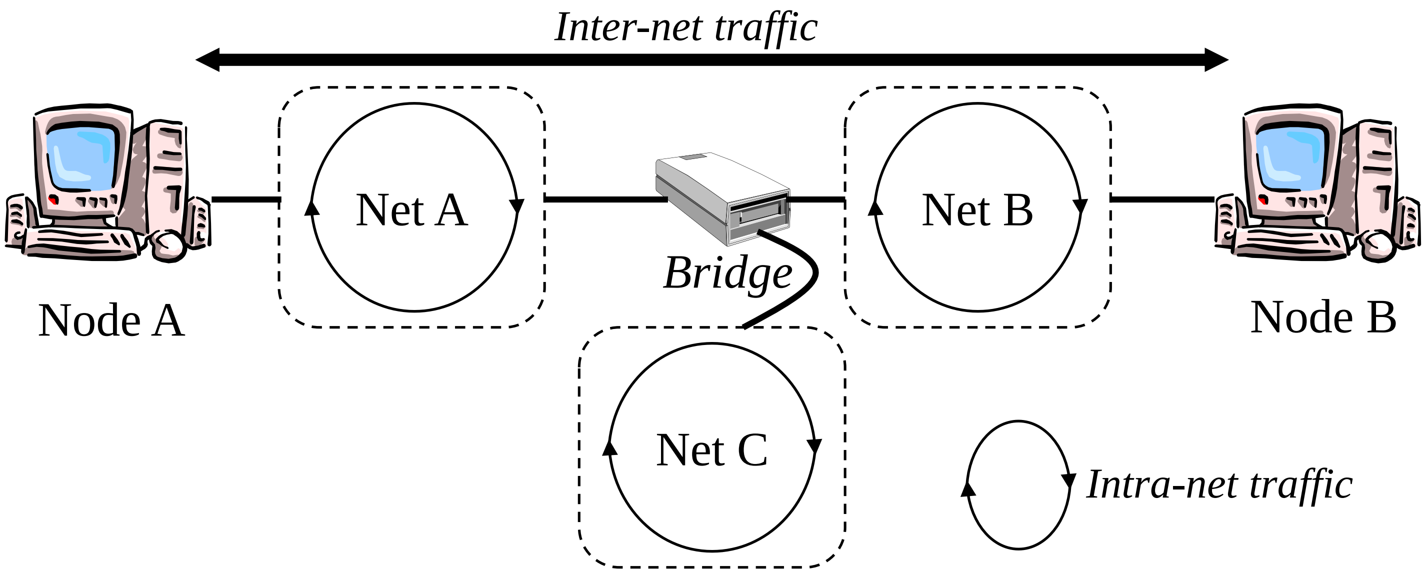
Switched LANs

- Star topology with one central component = *Switch*
- Every connected node can use the full network bandwidth
- Usually separation of sending and receiving direction (full duplex transmission)
- Parallel processing/forwarding of multiple incoming data packets
- Input and output are connected using an address table



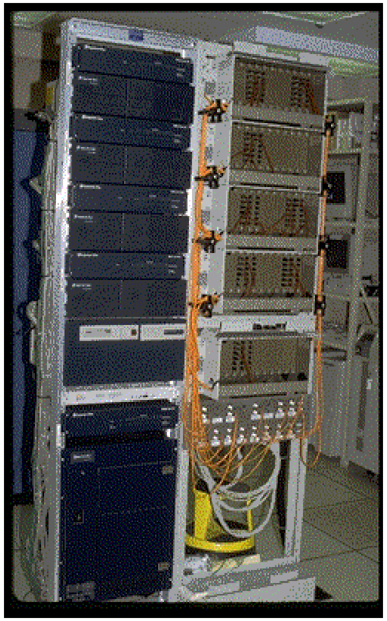
Bridge

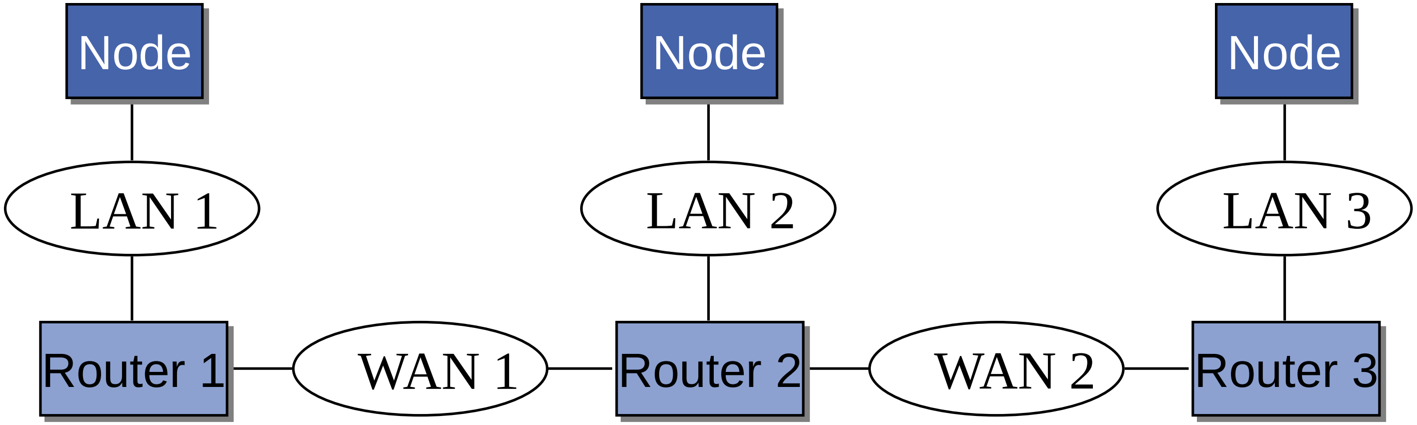
- Coupling of networks on layer 2
  - Networks of same type (e.g.. 802.x with 802.x) (non-translating)
  - Networks of different type (e.g. 802.x with 802.y (x^{1}y)) (translating)
- Functions:
  - Separation of intra-net traffic from inter-net traffic (filter functionality)
  - Increase of network capacity of large networks through partitioning (each partition with full number of nodes/network length)
  - Realization of simple routing functionality (routing on layer 2)



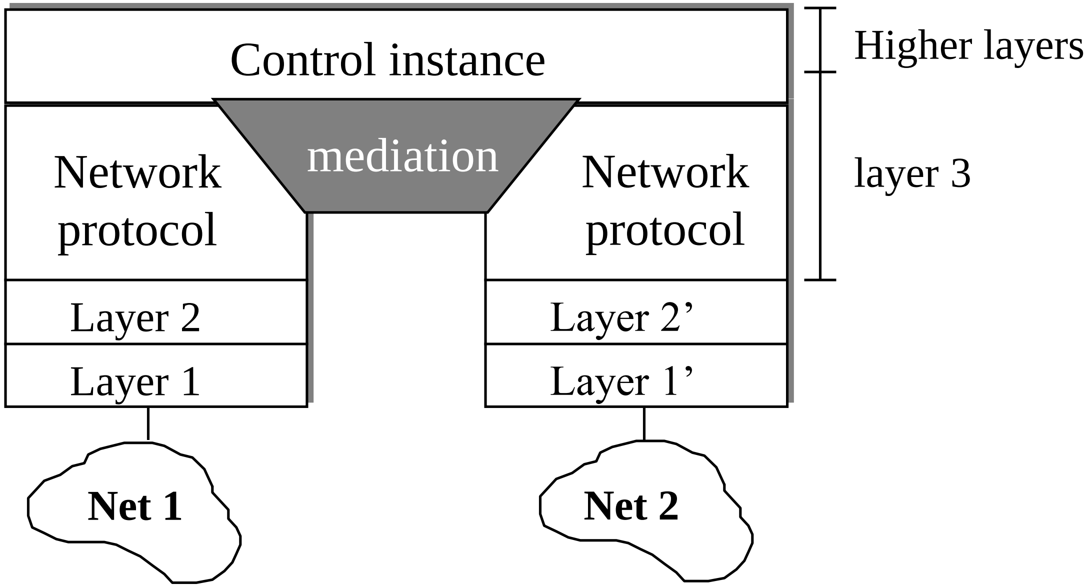
Router

- Network coupling on layer 3
- Functions:
  - Communication between distant nodes over one or more WANs
  - Routing using global unique and preferable hierarchical network addresses (e.g. IP address, ISO address)
  - Segmentation/reassembling of layer 3 packets to adapt to differing maximum packet sizes in layer 2
  - Security measures to control network access based on network address (“Firewall”)
  - Automatic limit of layer 2 broadcasts





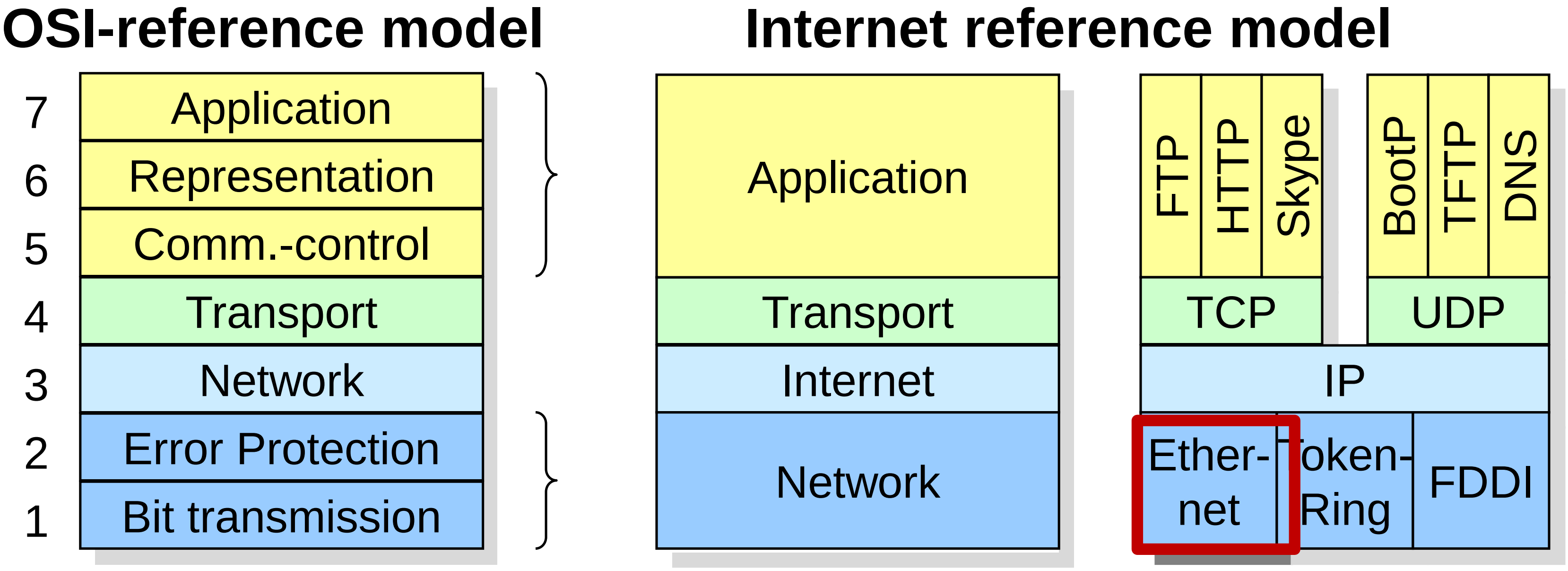
Architecture of a router



- Main characteristics:
  - Each network has its own layer 1 and layer 2 instances
  - Generally the network protocol is the same for all networks (e.g. IP-Router)
  - Choice of path is done based on unique network addresses
  - Mediation component connects the two network protocol instances, realizing the forwarding functionality
  - Control instance may implement routing protocols, protocols for error monitoring and management protocols

Examples for Networks

OSI Model in the real world: Ethernet

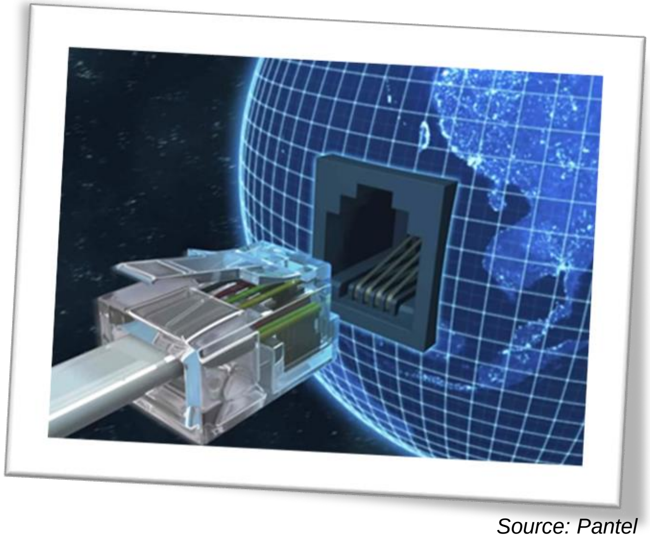


Ethernet

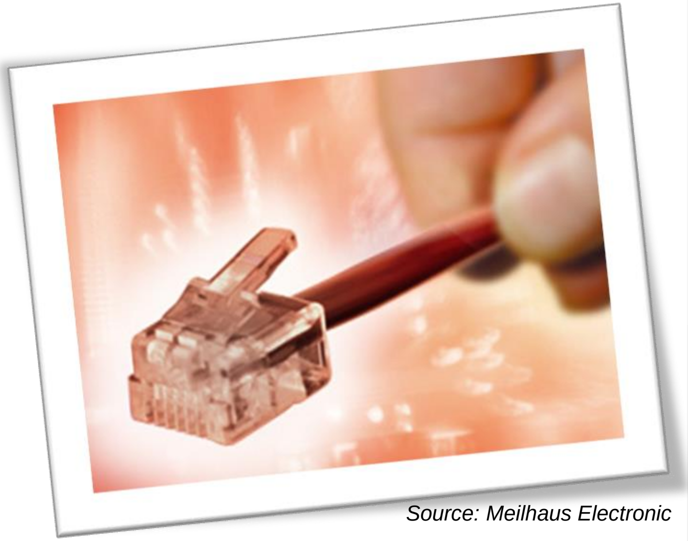
- Physical Layer, developed 1970‘s by Xerox
- Twisted Pair Wiring

- Dominant LAN technology due to
  - Easy to understand, implement, manage, and maintain
  - Allows low cost network implementations
  - Highly flexible topology for network installation
  - manufacturer independent interoperability of network components is guaranteed

- Uses CSMA/CD protocol for arbitration



Source: Pantel



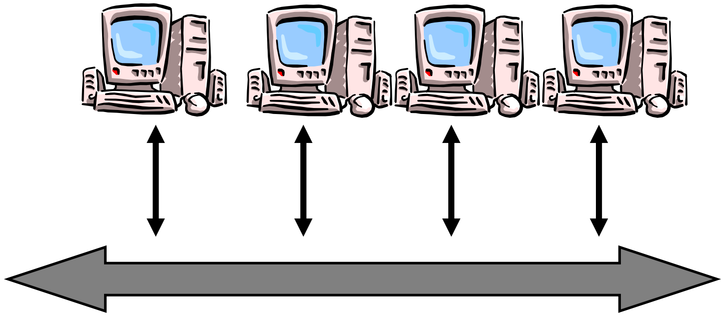
Source: Meilhaus Electronic

CSMA/CD (Ethernet)

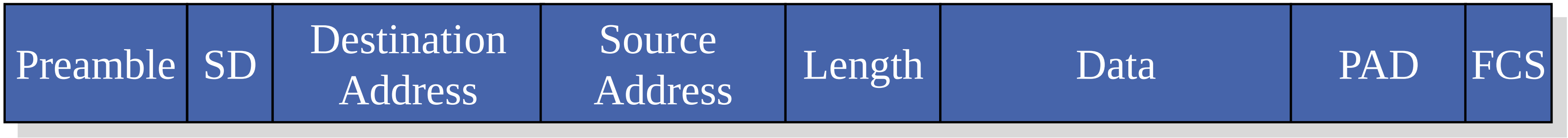
- Topology
  - all stations are coupled over a common bus
  - every station can send at any point in time

  ⇨ Collision of multiple transmissions destroys data!

- Arbitration Scheme:
  Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
  - prior to sending: check medium (Listen Before Talk)
  - if medium free: start to send
  - while sending: check data on bus (Listen While Talk)
  - upon detection of a collision: abort send process and notify all connected stations



Packet format CSMA/CD according to IEEE 802.3



Preamble	SD	Destination Address	Source Address	Length	Data	PAD	FCS

PR = Preamble for Synchronization [56 bit]

SD = *Start-of-frame Delimiter* shows beginning of a block (10101011) (AB_H) [8 bit]

DA = *Destination Address* [16/48 bit]

SA = *Source Address* [16/48 bit]

Length = Number of octettes within data field [16 bit]

Data = Data field [0 - 1.500 byte]

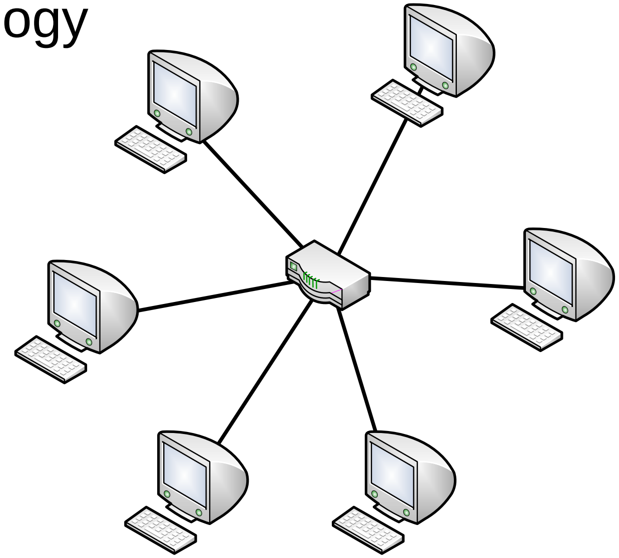
PAD = *Padding*, to append short data fields to reach necessary length of a data field (Collision detection) [0 - 368 bit]

FCS = *Frame Check Sequence*, Polynomial division via CRC32 polynomial for error detection [32 bit]

Important: Some implementations of CSMA/CD (e.g. Ethernet 1.0, Ethernet 2.0 or IEEE 802.3) use some fields with slightly different meaning!

Fast Ethernet (IEEE 802.3u)

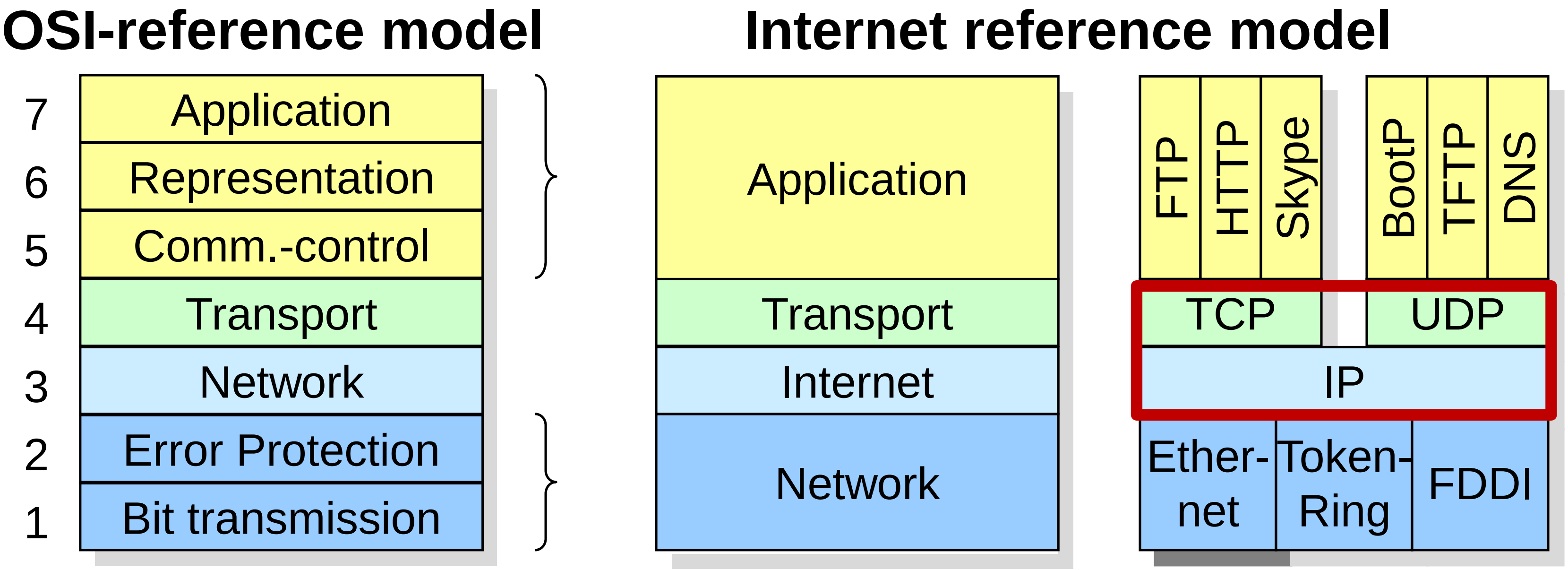
- Star topology



- Transmission capability 100 Mbit/s

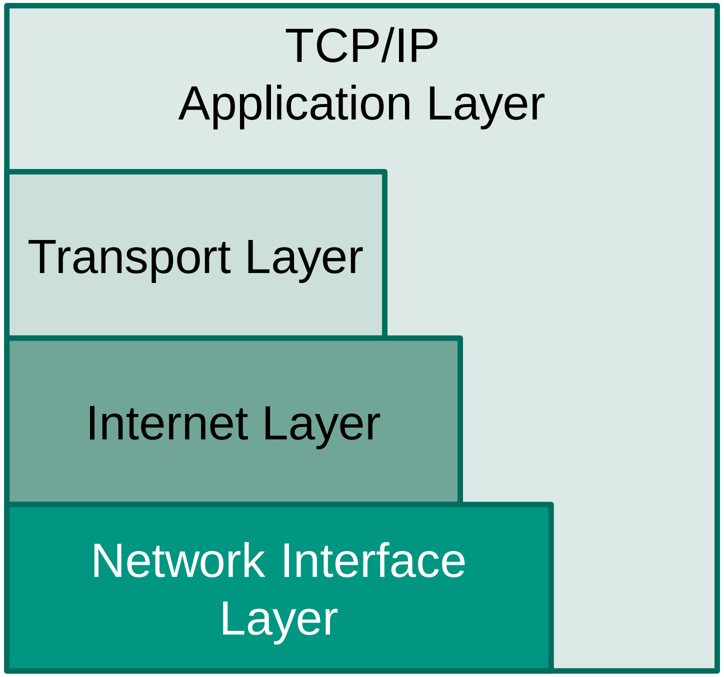
Description	Cable	Segment Length	Advantages
100Base-T4	Four twisted pairs	100m (max.)	common cables
100Base-TX	Two twisted pairs	100m (max.)	full duplex at 100 Mbit/s
100Base-F	optic fibers	800m (max.)	full duplex, long distance

OSI Model in the real world: The Internet



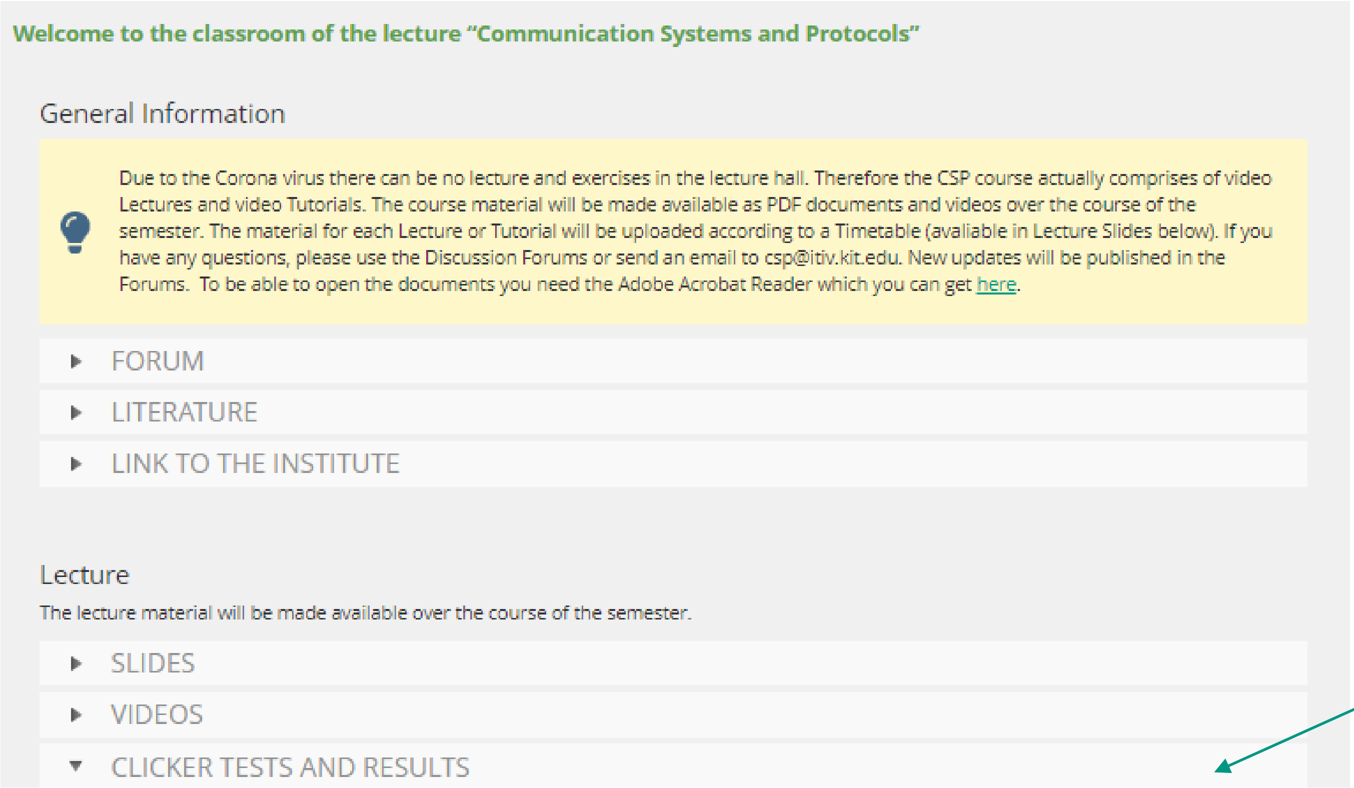
TCP/IP

- Protocol suite comprising
  - Transmission Control Protocol (TCP)
  - Internet Protocol (IP)
  - Other related protocols: User Datagram Protocol (UDP), Address Resolution Protocol (ARP), etc. …



For more in-depth information visit the lecture „Telematik“ (Prof. Dr. Martina Zitterbart)

Participate in Clicker Test 10b



Welcome to the classroom of the lecture "Communication Systems and Protocols"

General Information

Due to the Corona virus there can be no lecture and exercises in the lecture hall. Therefore the CSP course actually comprises of video Lectures and video Tutorials. The course material will be made available as PDF documents and videos over the course of the semester. The material for each Lecture or Tutorial will be uploaded according to a Timetable (avaliable in Lecture Slides below). If you have any questions, please use the Discussion Forums or send an email to csp@itiv.kit.edu. New updates will be published in the Forums. To be able to open the documents you need the Adobe Acrobat Reader which you can get here.

Lecture

The lecture material will be made available over the course of the semester.

Clicker Tests are added here

Thank you for your attention

Karlsruher Institut für Technologie

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)

Communication Systems and Protocols

Session 28: Networks-on-Chip - Basics

System-on-Chip (SoC)

- SoC is an integrated circuit that incorporates all the components of an electronic system. The components include processor, memory and input/output ports.

- Multi-processor System-on-Chip (MPSoC) : Contains more than one processing elements. They can be
  - Heterogeneous MPSoC : containing different processing elements like Digital Signal Processors (DSP), accelerators, peripherals, interconnect.
  - Homogeneous MPSoC: containing multiple general purpose or dedicated processors, memory, interconnect and peripherals.

Motivation

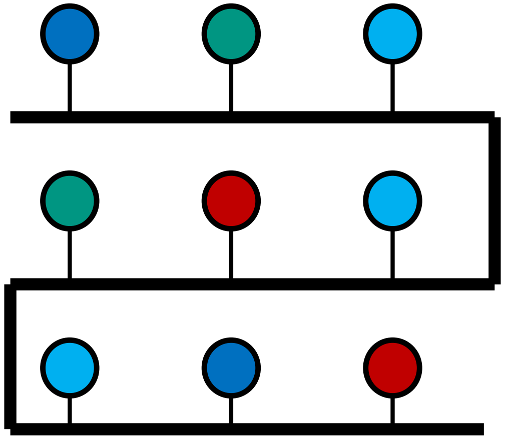
- System-on-Chip (SoC) are used in different domains like telecommunications, consumer electronics and multimedia. Examples include mobile phones, automotive systems, digital televisions, set-top boxes, video games, …

- The number of cores on a chip computing information is increasing and future integrated systems will contain thousands and more of such cores.

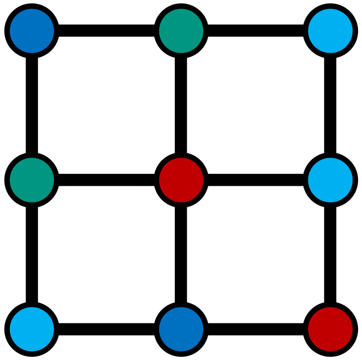
- For such systems, a bus based on-chip communication would become a critical performance and power bottleneck.

Network vs. Bus (Need for NoC)

- Bus:
  - dedicated and fixed physical communication channel
  - bandwidth is limited by bus line
  - Communication bandwidth is constant for growing number of nodes → **bad scalability**



- Networks:
  - different and multiple communication channels are possible
  - Error resistance can be higher due to alternative routing
  - variable Bandwidth due to multiple, parallel channels



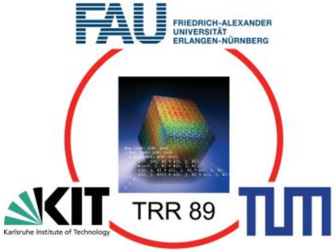
NoC vs LAN

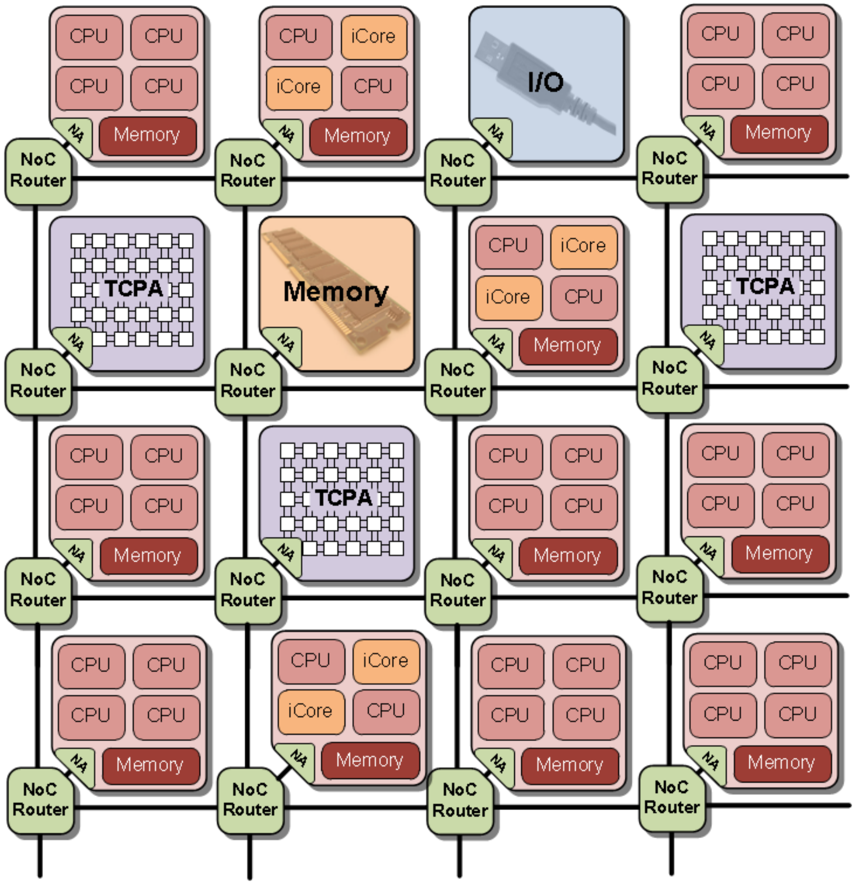
- Local Area Networks (LAN)
  - high computation power (PC)
  - wide coverage area
  - varying topology
  - very large number of connected nodes
- Network-on-Chip (NoC)
  - coverage area limited to the chip
  - topology usually fixed
  - number of nodes limited by chip size
  - High performance
  - Simple routers, low complexity
- But basic network and routing mechanics are similar for both types of networks.

Example for NoC: iNoC

- In the project **Invasive Computing (InvasIC)** we are investigating a novel paradigm for the design and resource-aware programming of future parallel computing systems.

- One of the research areas at ITIV
  - **Invasive NoCs (iNoC)** – Autonomous, Self-Optimising Communication Infrastructures for MPSoCs


FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
KIT Karlsruhe Institute of Technology
TRR 89
TUM



Upto 4x4 Tiles and 80 cores.

Building Blocks of NoC

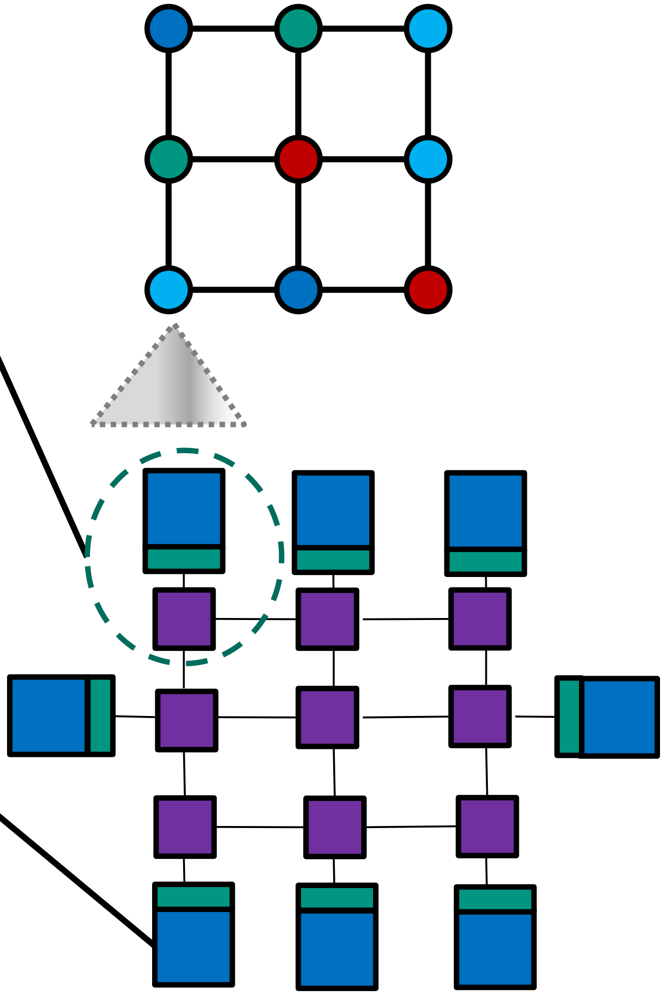
Basic Definitions

Node

- Basic communication entity that gets connected in a network
- A network node (normally) contains Ports, Routing Unit, Network Interface and a Tile

Tile

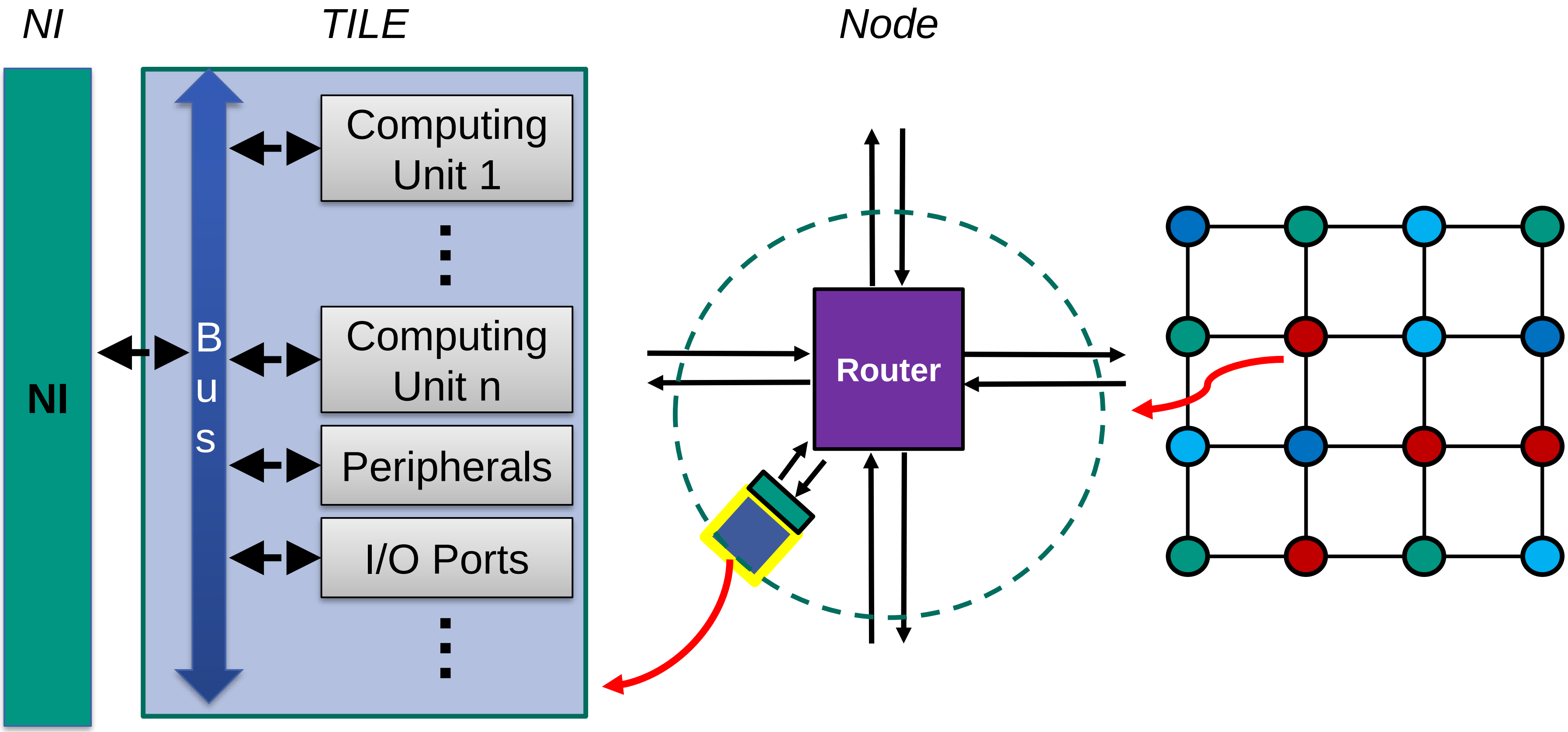
- A Tile is the unit that produces and consumes the data that is to be transported over a network
- Computing Units, Peripherals, Memory, Input/Output ports, etc.



🔵 Node 🟦 Tile 🟩 Network Interface 🟪 Routing Unit/Router

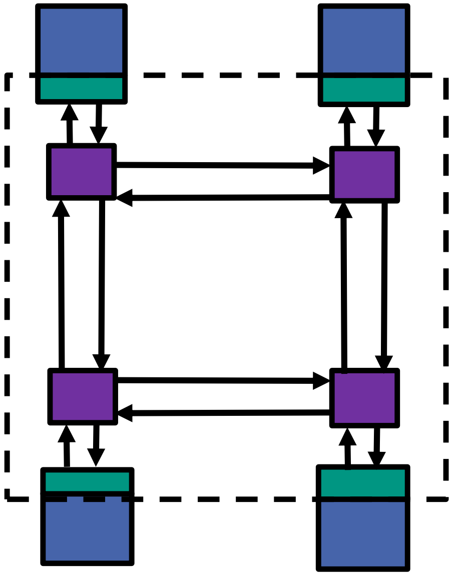
Node : Extended Setup

- A network node can contain:
  - multiple Computing Units
  - Peripherals (such as memory)
  - Input/Output Ports
  - …



Building blocks of NoC

- ***Link*** ←
- ***Router*** 🟪
- ***Network Interface (NI) / Network Adapter (NA)*** ▬
- ***Tile / Computing Unit*** 🟦



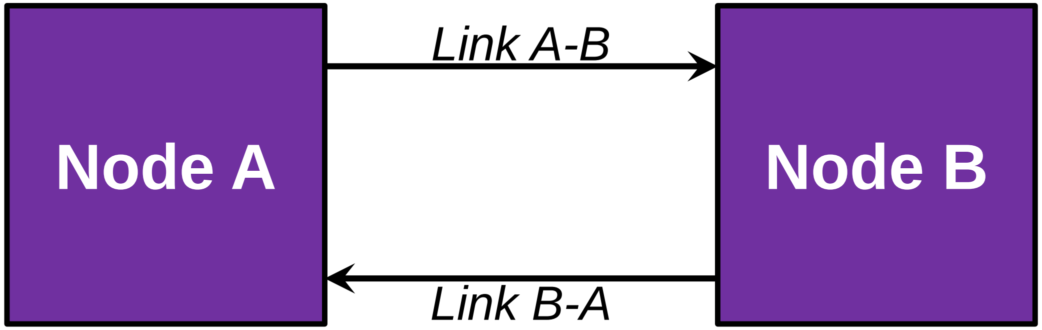
Chapter 2, NoC Basics .É. Cota et al., *Reliability, Availability and Serviceability of Networks-on-Chip*, 11
DOI 10.1007/978-1-4614-0791-1_2, © Springer Science+Business Media, LLC 2012

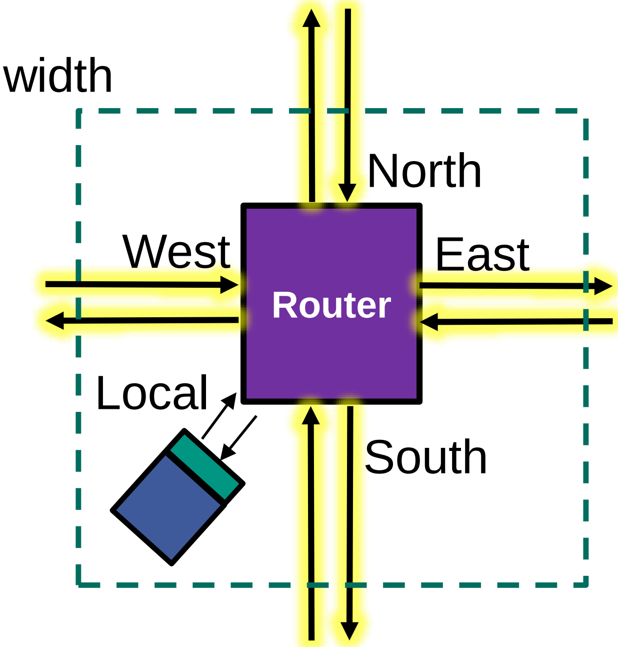
Network Link

- Physical communication channel between neighboring nodes
- Unidirectional
- Link has fixed data width
- Multiple links between two nodes can exist (in either direction)
  - concurrent transmissions

  or

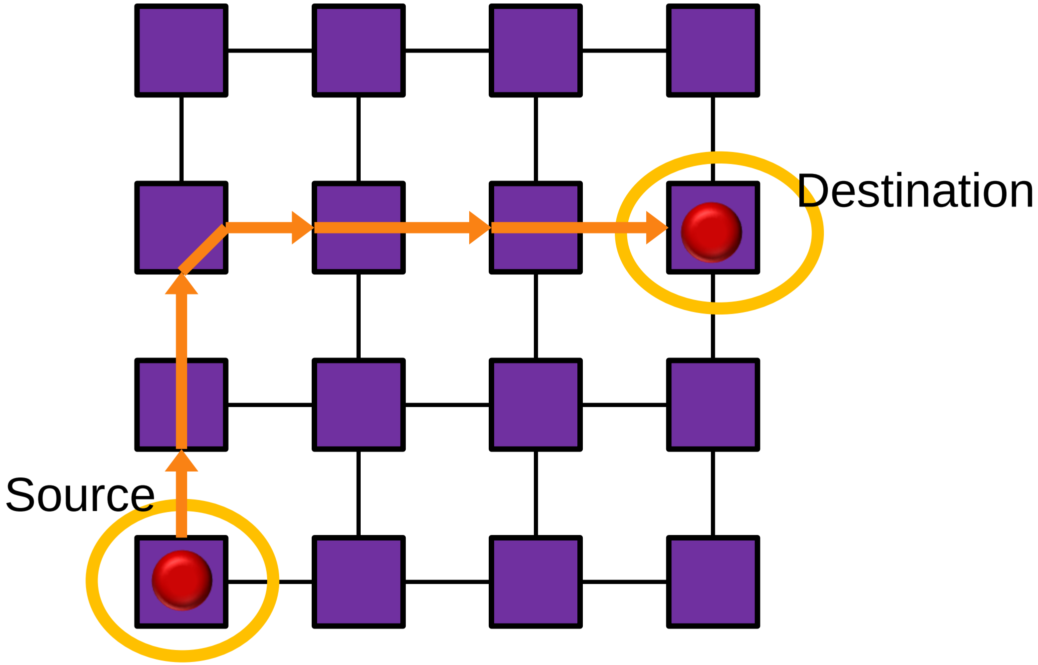
  - aggregation to achieve higher throughput/bandwidth

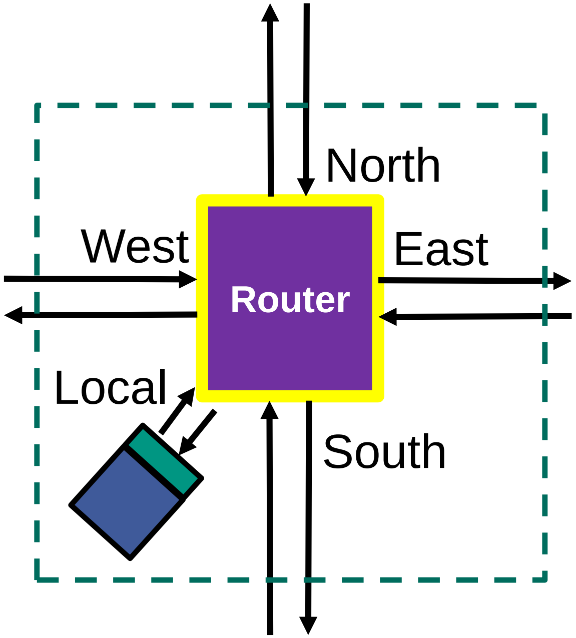




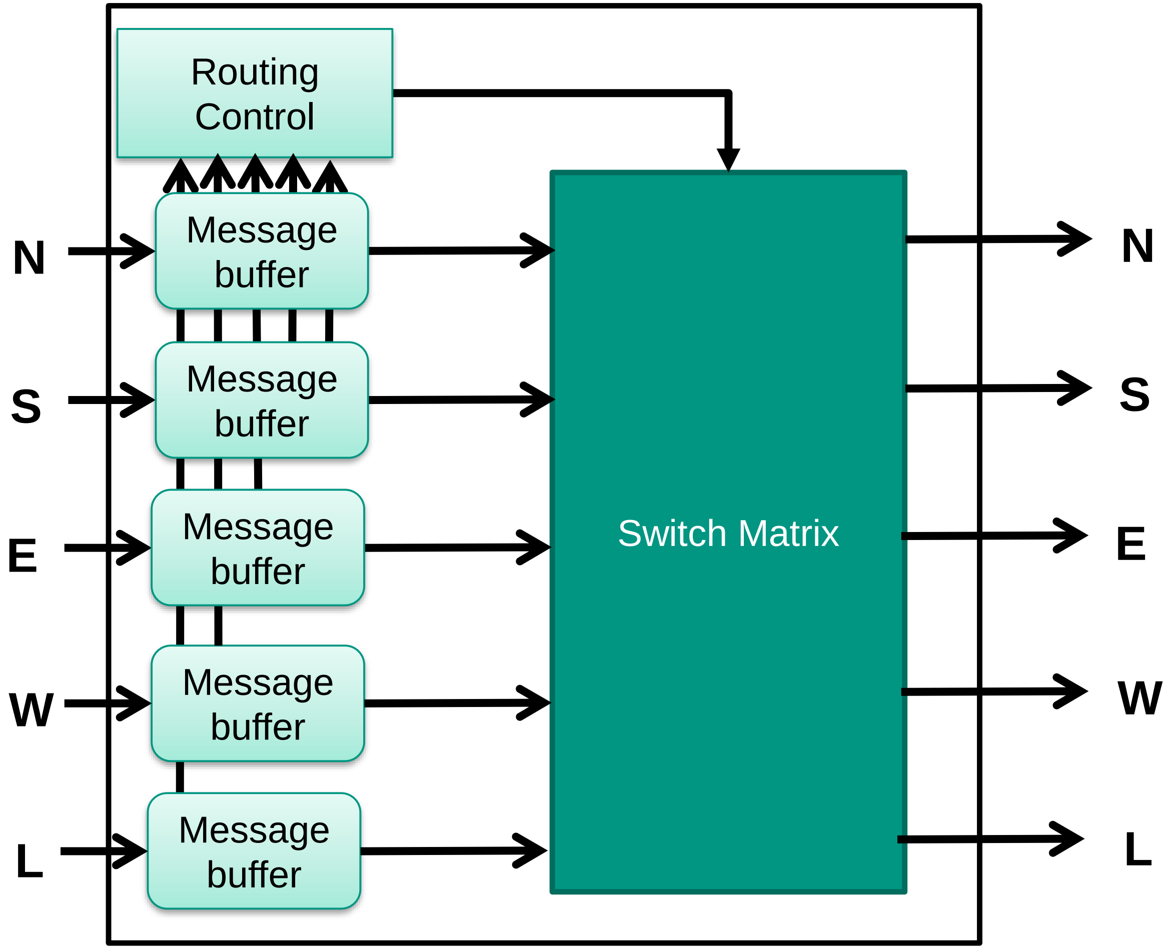
Router

- Switches Paths
- Embedded intelligence that decides on the direction of the data (routing algorithm)
- Refresh signal
- Additional functionality possible (local storage, Error Correction, …)



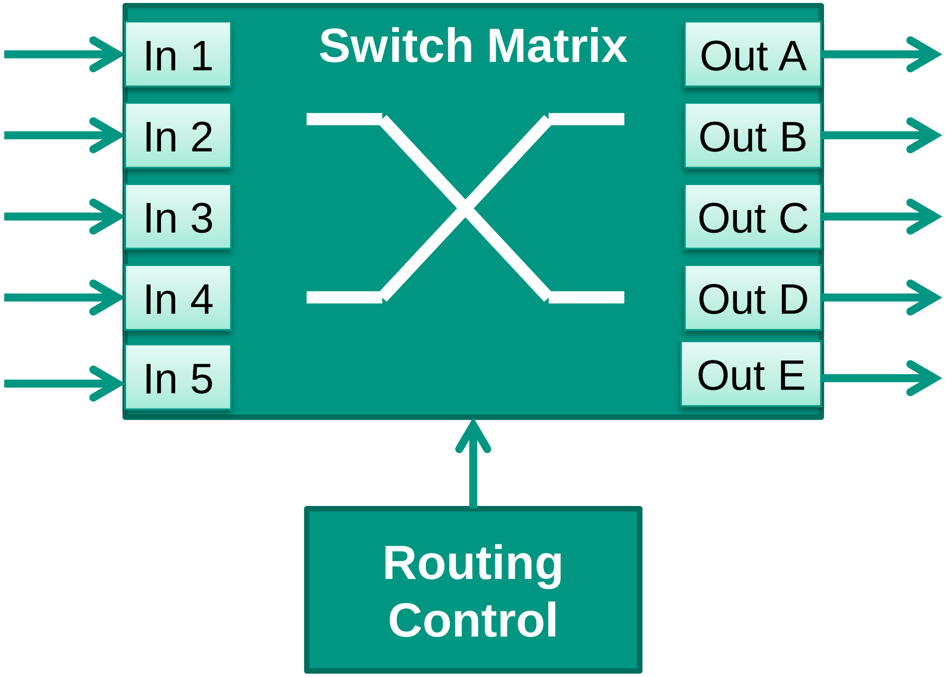


Typical components in a Router



Typical Components in a Router

- Switch matrix
  - connects one input to one output
  - multiple connections at the same time can exist
- Routing Control
  - evaluates data address
  - implements routing connections according to routing protocol
  - determines what input is connected to what output

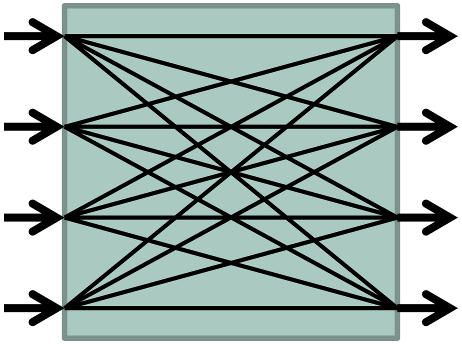


Examples for Switching matrix

- Switching matrixes are used to electrically connect inputs with outputs

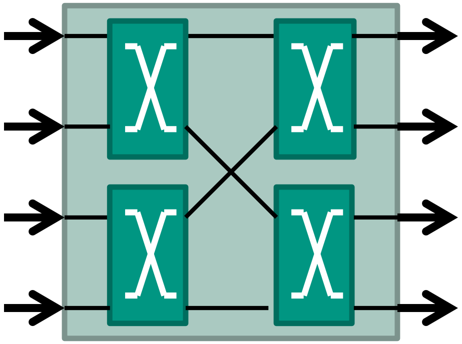
- Full Crossbar
  - Any input can be connected to every output
  - No limitations for routing
  - Expensive in terms of hardware resources

Full Crossbar



- Reduced Crossbar
  - Number of concurred connections is limited
  - Fewer resources required
  - Routing algorithm has to take care of individual characteristics of matrix

Reduced Crossbar

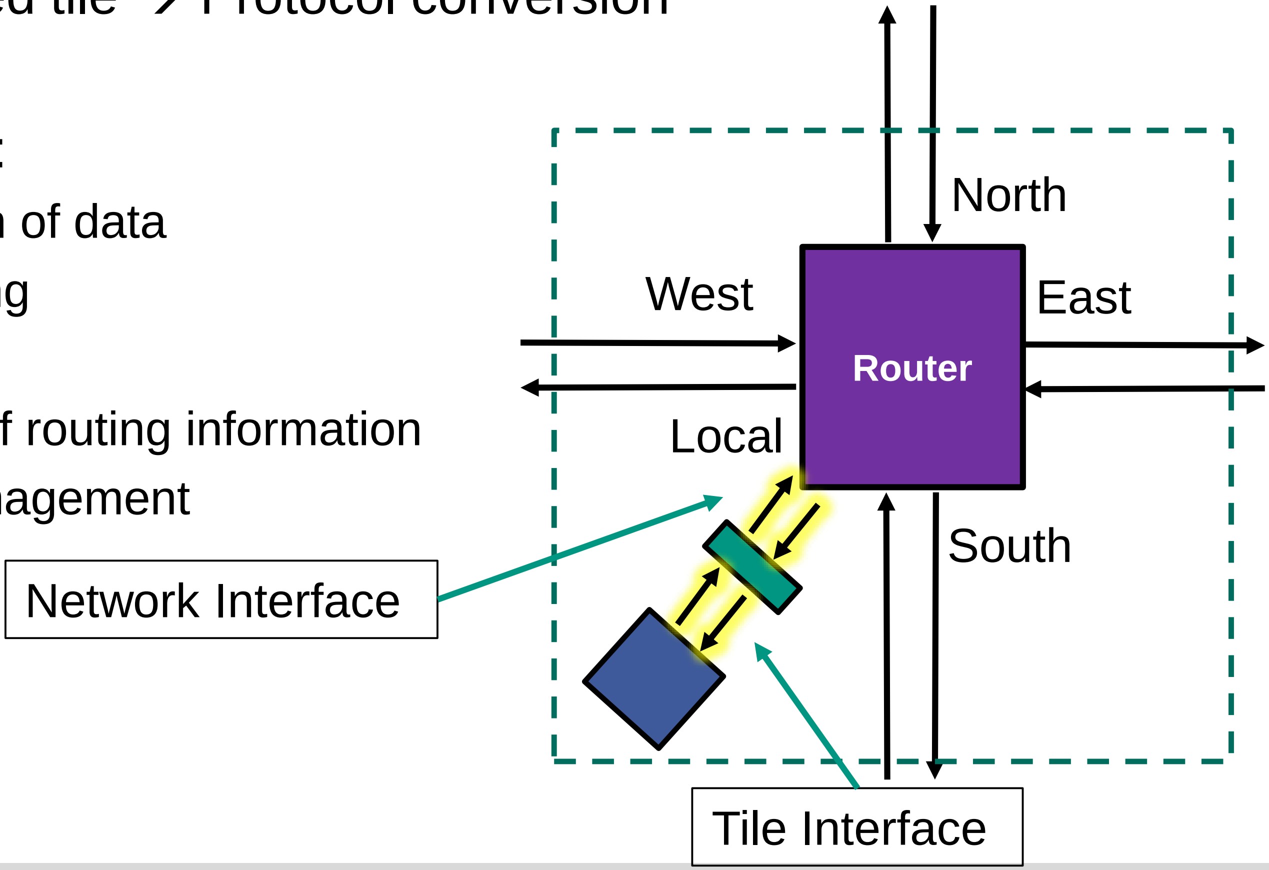


Network Adapter / Network Interface

- Used to attach system components to the network

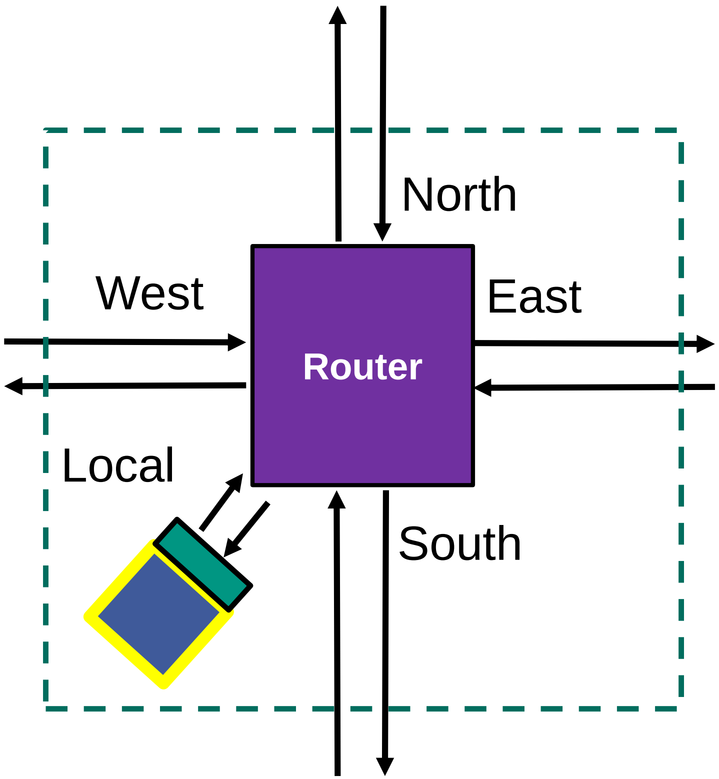
- Can act as a wrapper to hide the network specific behavior from the attached tile → Protocol conversion

- Responsible for:
  - Packetization of data
  - Error Handling
  - Flow control
  - Generation of routing information
  - Network management



Computing Unit

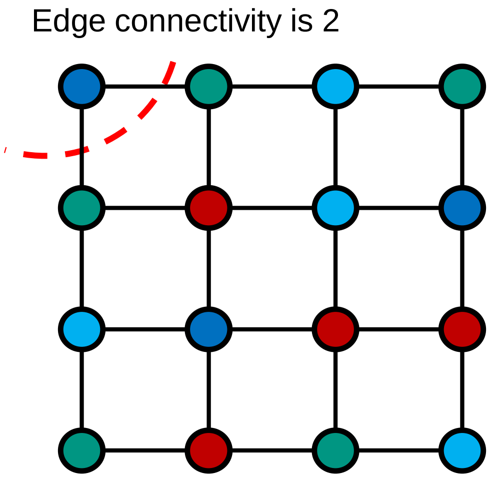
- Computational Entity
  - Processor/Microprocessor
  - Memory
  - Infrastructure
  - I/O
- Runs an application or part of an application
- Is producing and consuming data
- Is connected to the network via the network interface



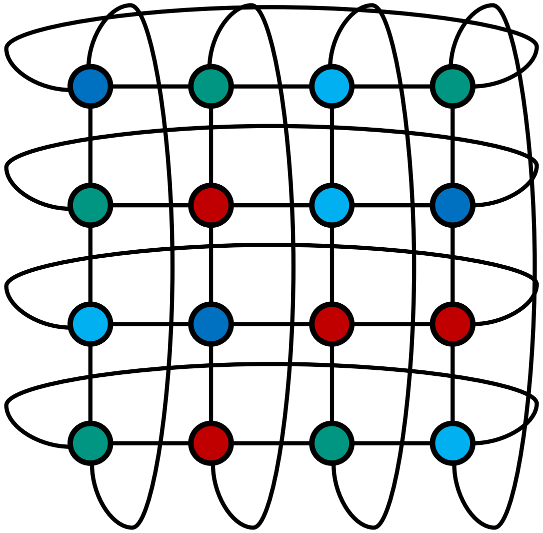
Physical Structure

Networks – Physical Structure

- Geometric topologies can be mesh, torus and any combinations



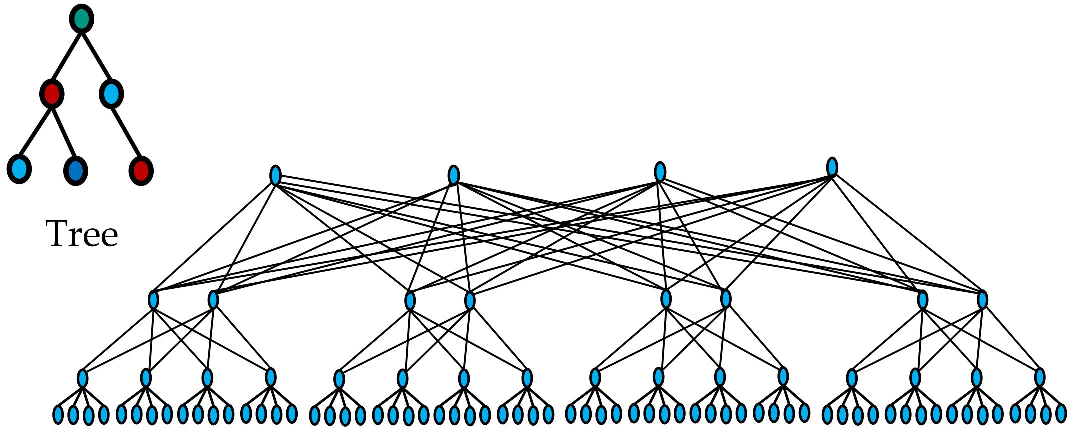
4x4 Mesh Topology



4x4 Torus Topology

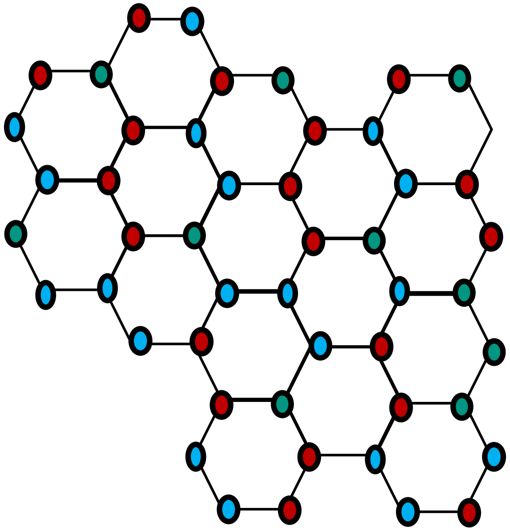
Edge connectivity
- Minimum number of edges to cut to disconnect a node
- Gives information of how resilient the network is
- Example: Mesh = 2

More examples

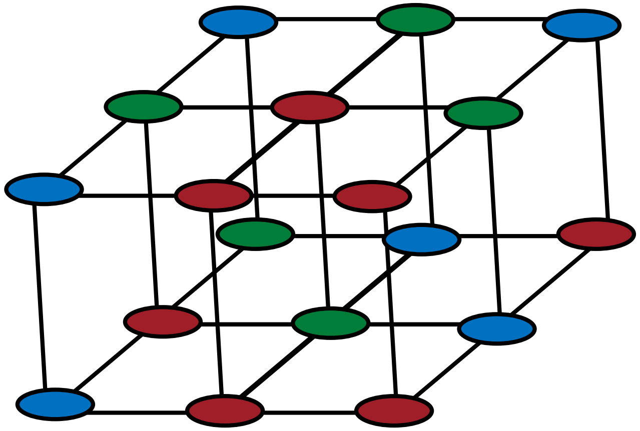


Tree

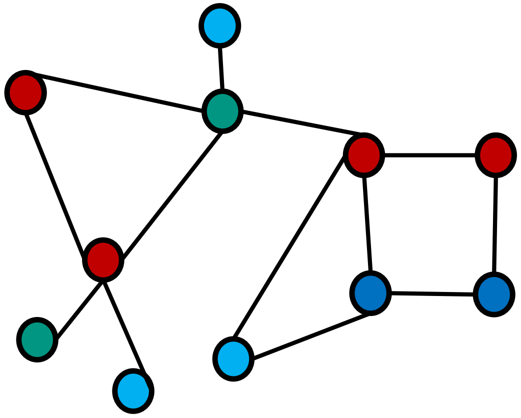
Butterfly Fat Tree



Honeycomb Mesh



3 D Mesh



Irregular

Definitions II

Hop

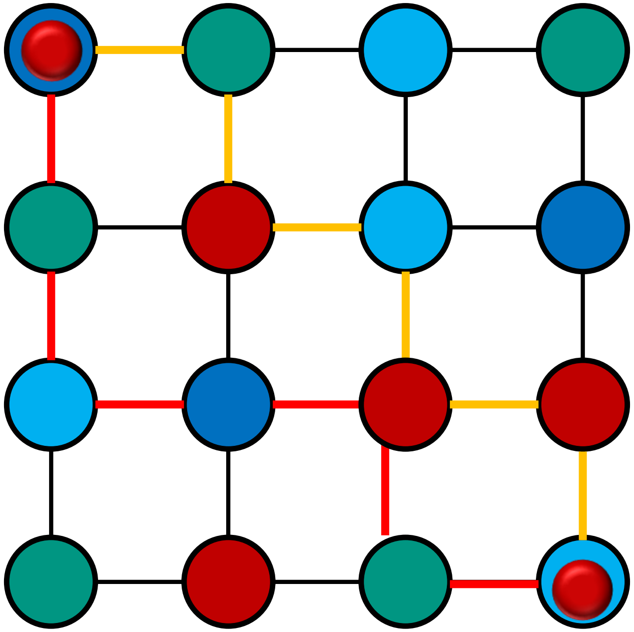
- The transmission of data from one node to the neighboring node

Diameter

- The Diameter of a network is the largest, minimal hop count over all pairs of nodes.

Manhattan Distance

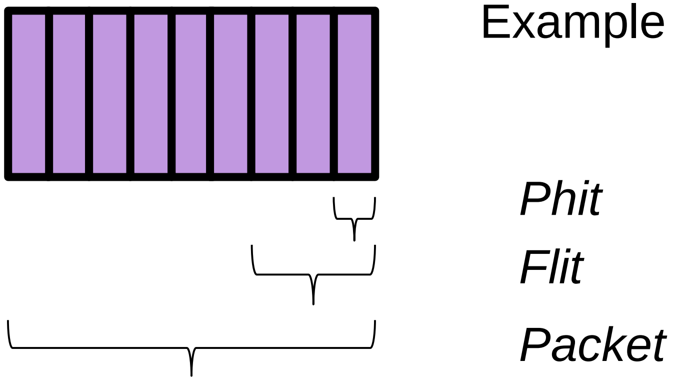
- Layout of the network must be a mesh based grid
- physical distance following the edges
- all paths that always move towards the target have same distance



Message

- Data is transferred on a link. A link has fixed width. The data is measured in bits.
- Unit of data transferred in a single clock cycle on a link is called the ***phit*** (physical unit).
- Two routers synchronize each data transfer (for example to ensure buffers do not overflow). Link-level flow control is used (can be based on handshaking).
- The unit of synchronization is called a ***flit*** (flow control unit). Is at least as large as a phit.
- One or more phits = ***flit***
- Multiple flits = ***packet***
- Multiple packets = ***message***

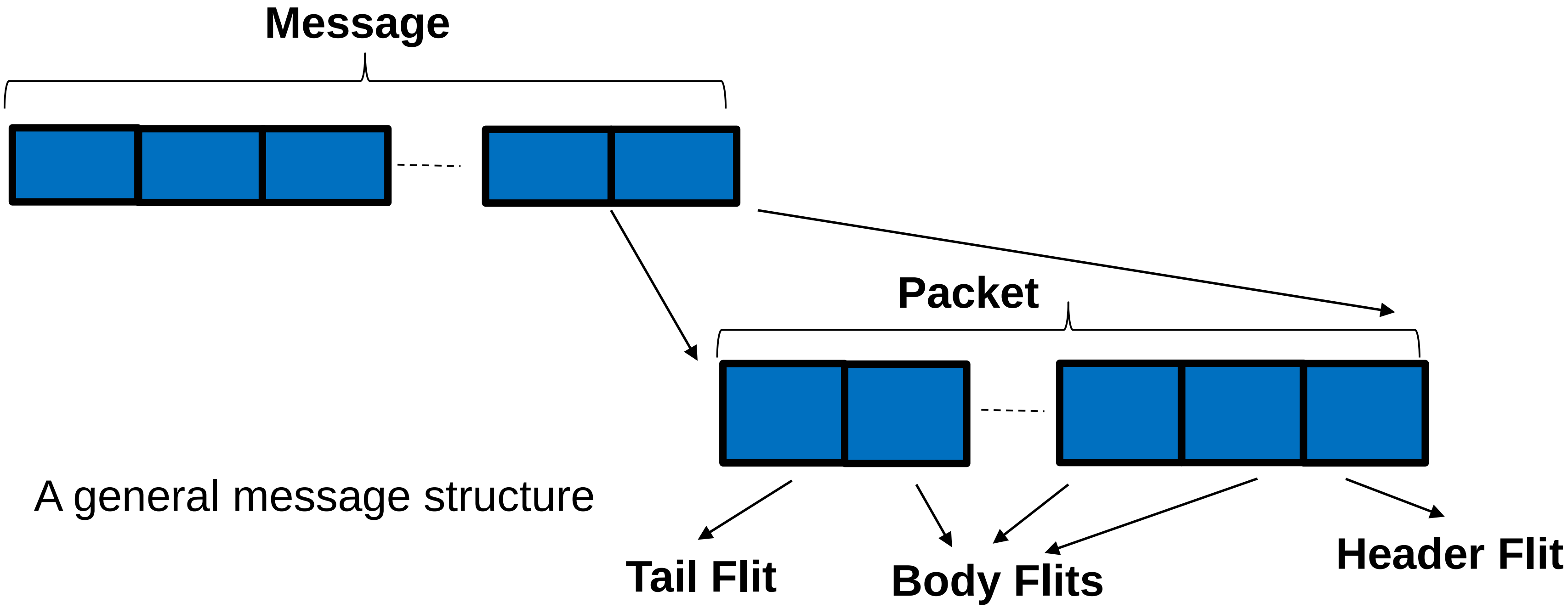
Example



Page 160, Networks on Chips: Technology and Tools, De Micheli, G. and Benini, L.ISBN 9780080473567, Systems on Silicon, https://books.google.de/books?id=IHHTmSBcoGIC, 2006,Elsevier Science

Message

- Different NoCs use different sizes for phit, flit, packet and message sizes.
- Examples
  - **iNoC** in InvasIC , Flit and phit size is equal and is 32 bits



A general message structure

Karlsruher Institut für Technologie

Thank you for your attention

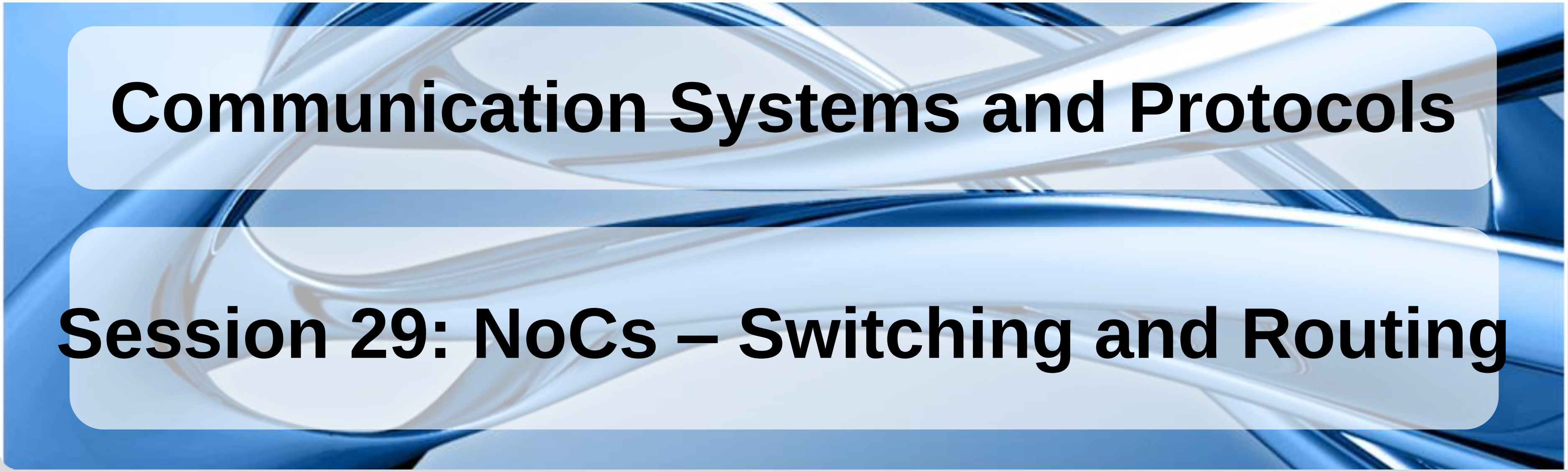
Karlsruher Institut für Technologie

Institut für Technik der
Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker *(becker@kit.edu)*

Dr.-Ing. Jens Becker *(jens.becker@kit.edu)*

Institut für Technik der Informationsverarbeitung (ITIV)



Communication Systems and Protocols

Session 29: NoCs – Switching and Routing

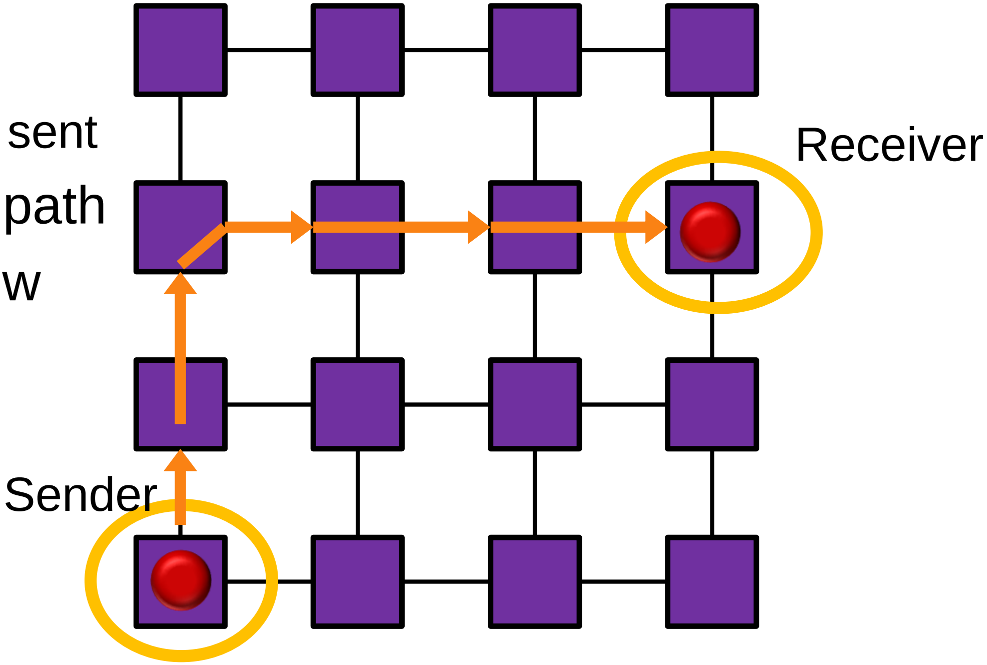
Switching vs Routing

- **Switching: How data flows in the network**
  - Circuit switching
  - Packet switching
- **Routing: which route the data should take** (eg. XY routing)

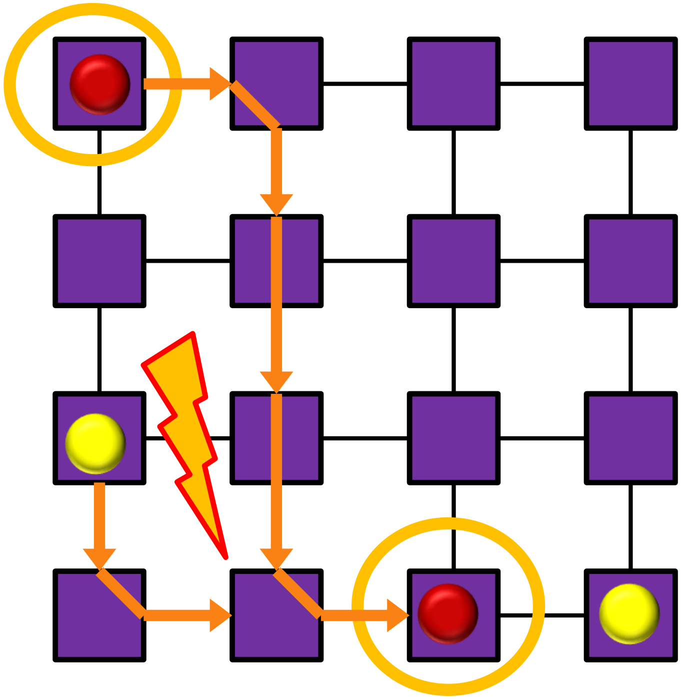
Switching

Circuit switching

- Messages are sent in their entirety when using circuit switching.
- A path is established between sender and receiver.
- Logically, the **head flit** travels from the sender reserving the links along the way.
- If the header arrives at the receiver, an acknowledgement is sent to the sender.
  - If links are not obtained, negative acknowledgement is sent
- Data transfer occurs, when the path is reserved. The **body flits** follow the head flit.
- After data transfer, the **tail flit** releases the path.



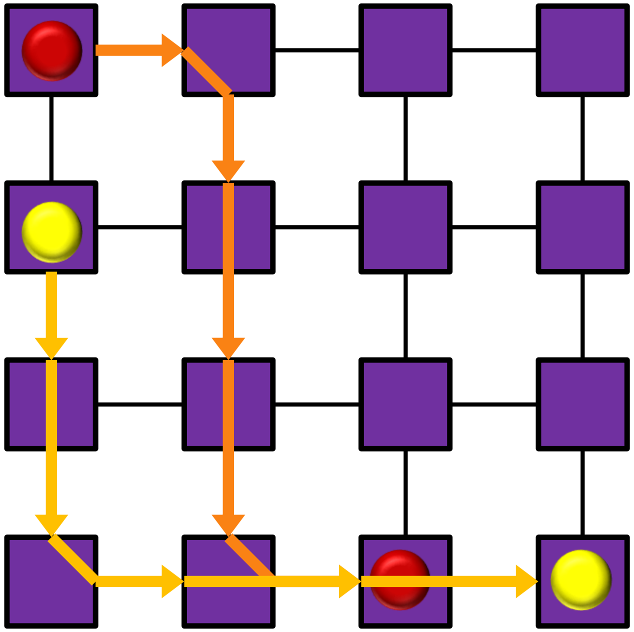
Circuit Switching



- **Pros**
  - No interference (on logic level)
  - Guaranteed throughput possible
  - Data need not be buffered
- **Cons**
  - High latency due to setting up the circuit initially.
  - Other communication paths may be blocked off
  - Does not scale well
- **Example:**
  - Analog telephone network
  - In NoCs: SOCBus

Packet Switching

- Data is send in consecutive packets
- Path is established for the duration needed to send one packet
- Path can be used in a time multiplexed fashion by multiple communication processes

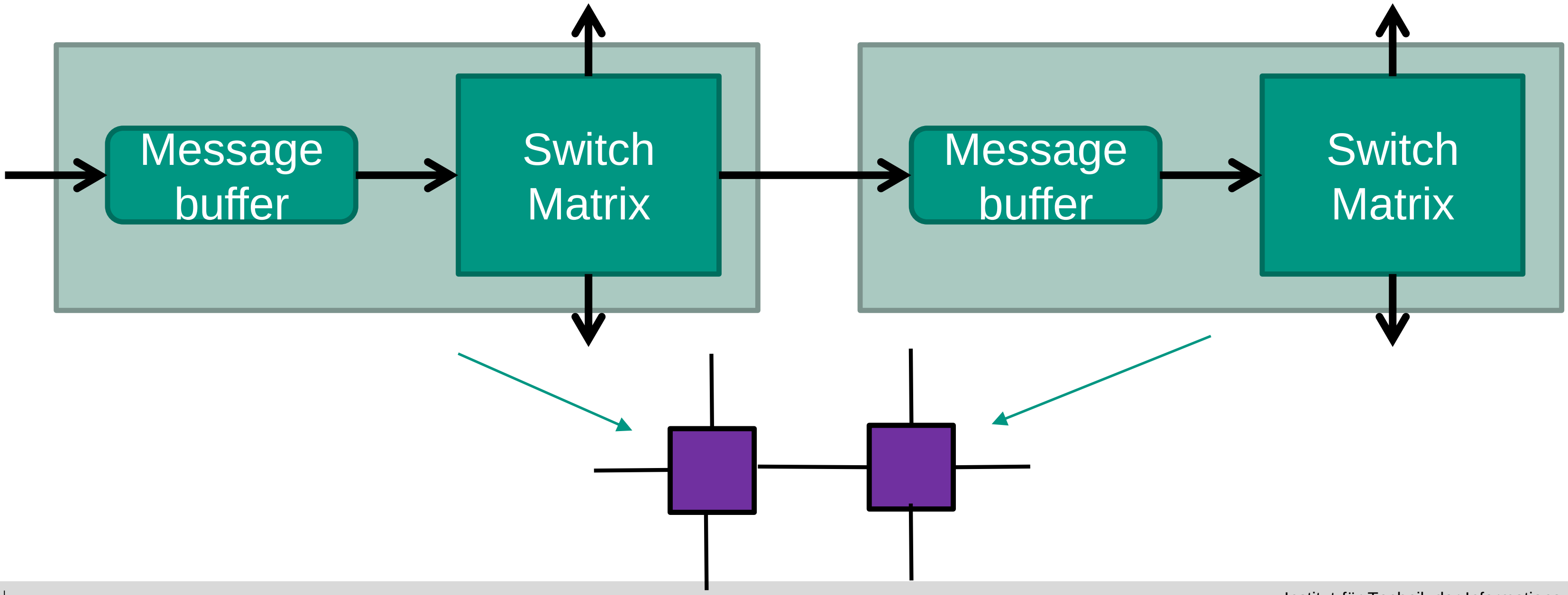


- **Pros**
  - Less blocking of communication paths
- **Cons**
  - Different routes of packets can lead to different arrival sequence
    - Ordering of packets may be necessary (packet ID)
  - Guarantees for network behavior difficult to give

- Examples: iNoC

Network link for packet switching

- For circuit switching, data can be directly forwarded from one node to next node
- For packet switching, messages have to be stored if next node is not ready to take data → message buffers required
  - Message is stored in buffer until next node is ready for reception
  - Multiple messages can be stored in buffer



Comparison

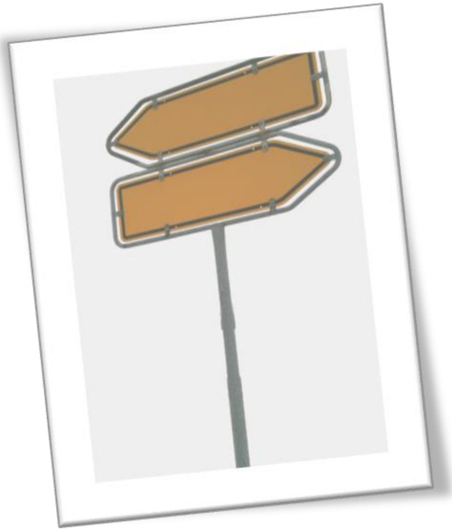
Circuit Switching		Packet Switching	
Pros	Cons	Pros	Cons
• good for streaming data • guaranteed throughput • fast • simple routers • high throughput • real-time capable	• can block other communication • inefficient use of communication resources (for sporadic communication) • Setup required prior to communication start	• good for sporadic data transfers • flexible • multiplexing of paths possible • efficient usage of communication resources (links)	• complex routers • additional control overhead • additional local storage needed • varying latencies possible • behavior hard to predict

Routing

The idea behind Routing

- Routing Problem
  - Connect data source with data sink
  - Decision on what paths to use
  - has to be integrated into the communication system (i.e. one or more network nodes, router,…)

- Possible Goals for Routing Algorithm:
  - Find short routes
  - Reduce routing overhead
  - Adaptability to…
    - changes in topology (node or link failures)
    - varying traffic loads
  - Avoid routing loops
  - …



Definitions

Loop

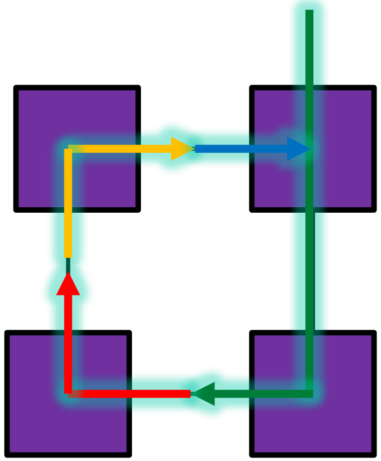
- A *Loop* is a path in a network that leads back to its starting point

Deadlock

- A situation where a link is blocked by one transmission that is waiting for the other transmission to finish is called a deadlock.
- No communication is possible in case of a deadlock

Livelock

- Data is forwarded through a network without reaching its destination



Routing Requirements

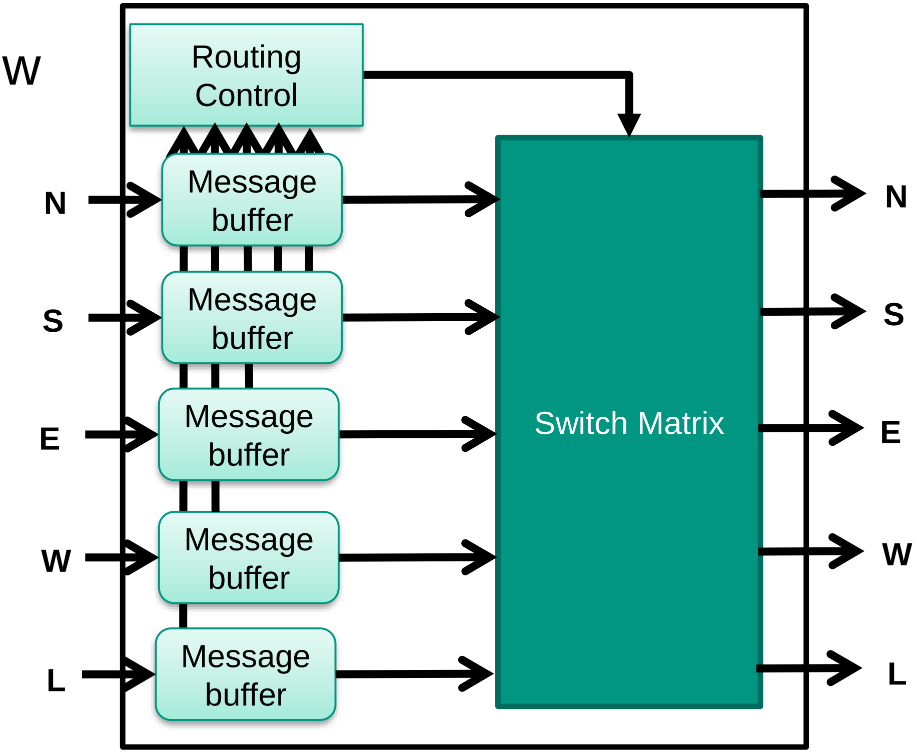
Usually multiple messages m_i are exchanged by multiple node pairs A_j and B_j

Routing goals

- Even Load Distribution over the network
- Minimization of *communication path or time (latency)*
- Prevention of *Deadlocks*
- Minimization of *Contention* (delay of messages due to competing requirements)
- Minimization of *Congestions* (deletion of messages, for example caused by buffer overflows)

Routing

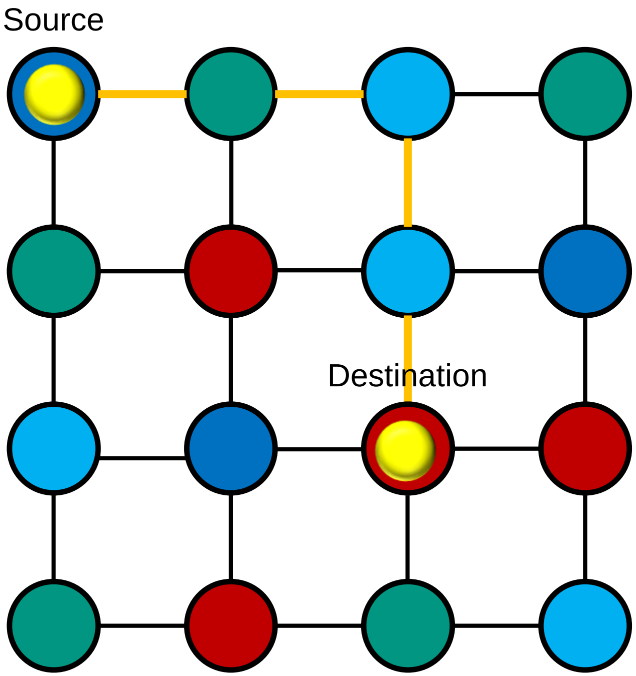
- Routing Unit implements a (pre-)defined behavior
  - connects correct IN/OUT
  - implements the communication paths
- Routing has to be defined somehow
  - Routing algorithms
- Routing Optimization Goals
  - minimal latencies
  - short routing path
  - balanced network load
- Examples of Routing Algorithms
  - X-Y Routing
  - Dijkstra-Algorithm



Routing Schemes

- Routing schemes can be classified into different categories.

- For example:
  - **Static (Deterministic) and Dynamic (Adaptive)**
  - **Distributed and Source routing**
  - **Minimal Routing and Non-Minimal Routing**



Static vs Dynamic

Static Routing

- predefined during design time
- generates fixed routing information for nodes
- Example : Dijkstra

- + simple nodes
- + fast pathfinding due to precomputation
- − not flexible
- − susceptible to defective nodes
- − no runtime adaptation

Dynamic Routing

- routing information is computed during runtime of the system
- the best path to a desired destination can change during operation
- knowledge of network status is necessary
- Example : Deflection Routing / “Hot Potato” Routing

- + highly flexible
- + scalable
- − computationally expensive
- − route finding slow due to internal computation

Source vs Distributed

Source Routing

- Sending node is defining the complete route for packet
- Information is part of the header of the packet
- Example : Used in Æthereal NoC

+ Optimal routings possible

+ Low complexity for intermediate routers

− Protocol overhead, complete routing information is transported to the network

− Overhead depends on path taken through the network

Distributed Routing

- routing computations is done in nodes
- cooperation of nodes by means of message exchanges possible
- Example: XY Routing used in iNoC

+ Easy adaptation to changing routing requirements during runtime possible

+ Constant protocol overhead → better scalability

− Intermediate routers get more complex

Minimal vs non-minimal

Minimal Routing

- always the shortest path is chosen toward the destination
- Example: XY

➕ Simple and can have lesser time cost in deciding paths

➖ If there is no balanced traffic, some links are overused and other links remain idle

Non-Minimal Routing

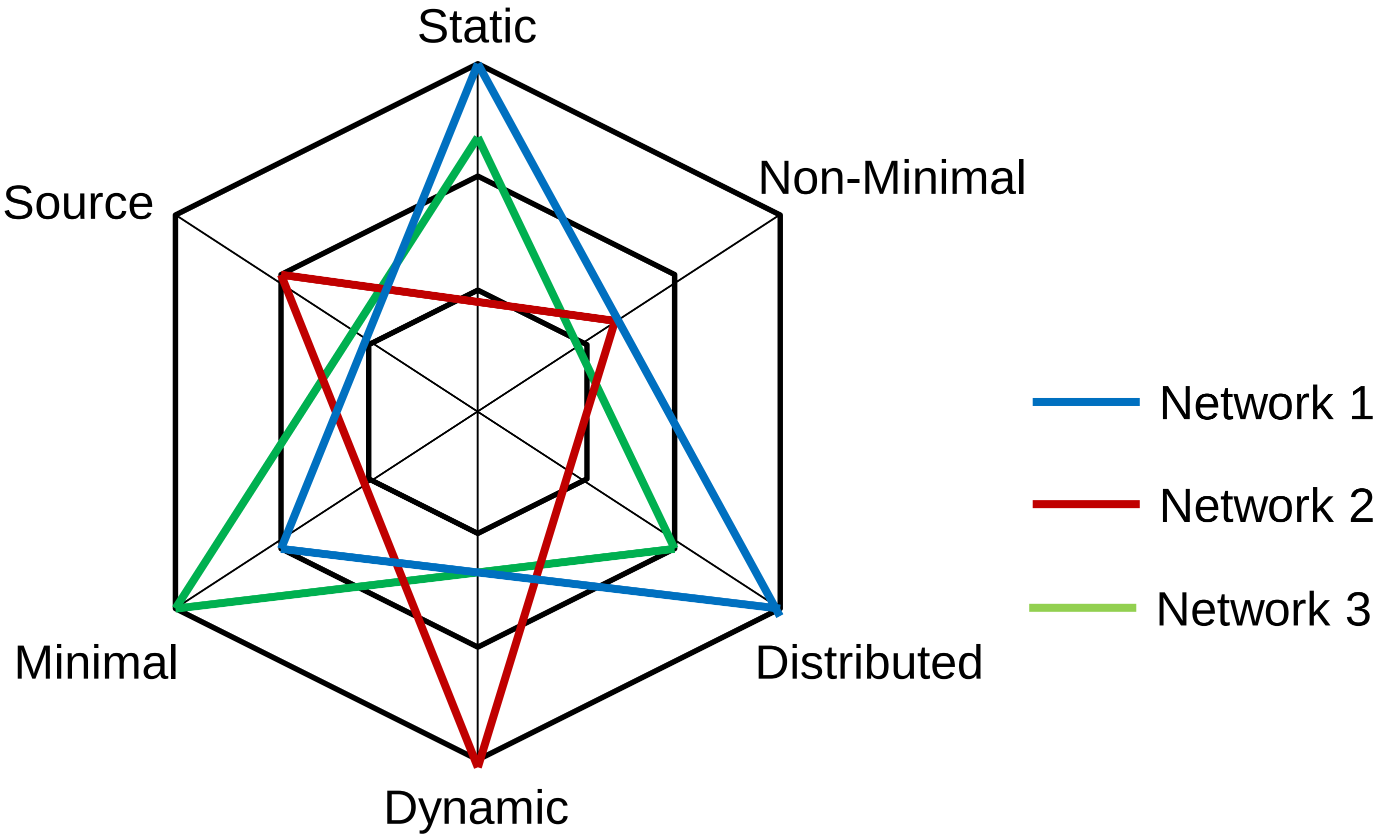
- not always shortest path is chosen
- Example: Deflection Routing

➕ Can choose alternate paths due to congestion or to ensure load balancing

➖ Can be more expensive due to additional power consumption

Routing

- Real world systems often encompass features or attributes of different categories



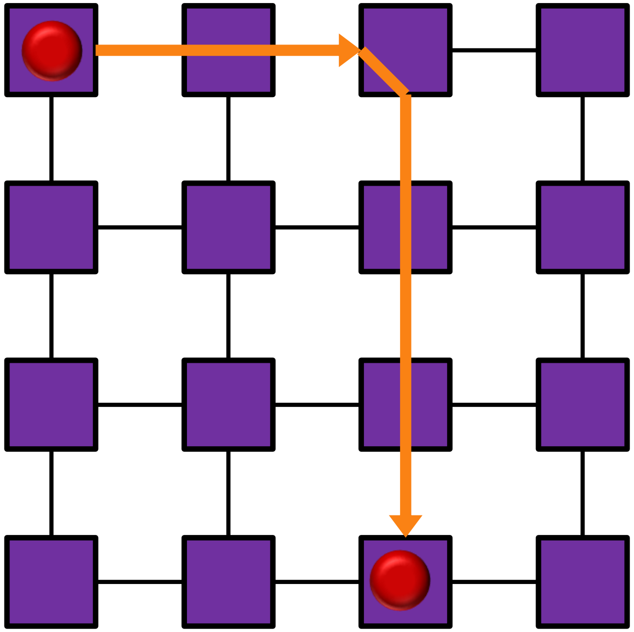
Evaluation criteria for routing algorithms

- Number of hops
  - How many links are used to reach the destination?
- Latency
  - Different links can have different bandwidth
  - Can be correlated with number of hops if same time per hop
- Guaranties on bandwidth, throughput, real-time capability, etc.
  - Latency has to be known
  - Waiting time inside router has to be limited
- Robustness, failure tolerance
  - System should be still operational if one or more links are broken
  - Flow control to make sure that packets reach destination

Examples for Routing Algorithms

X-Y Routing

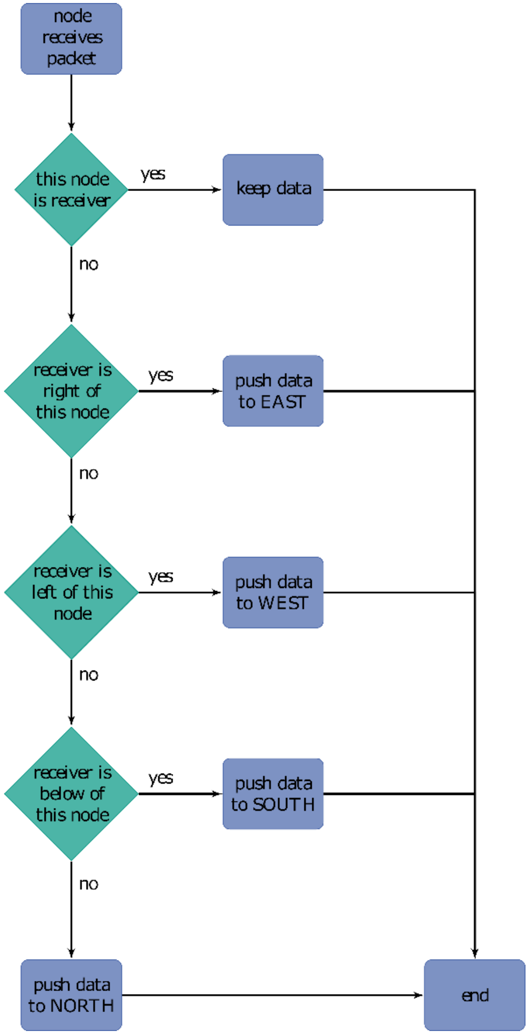
- Only applicable for mesh based networks
- Routing always towards destination
- First in X direction, then in Y direction

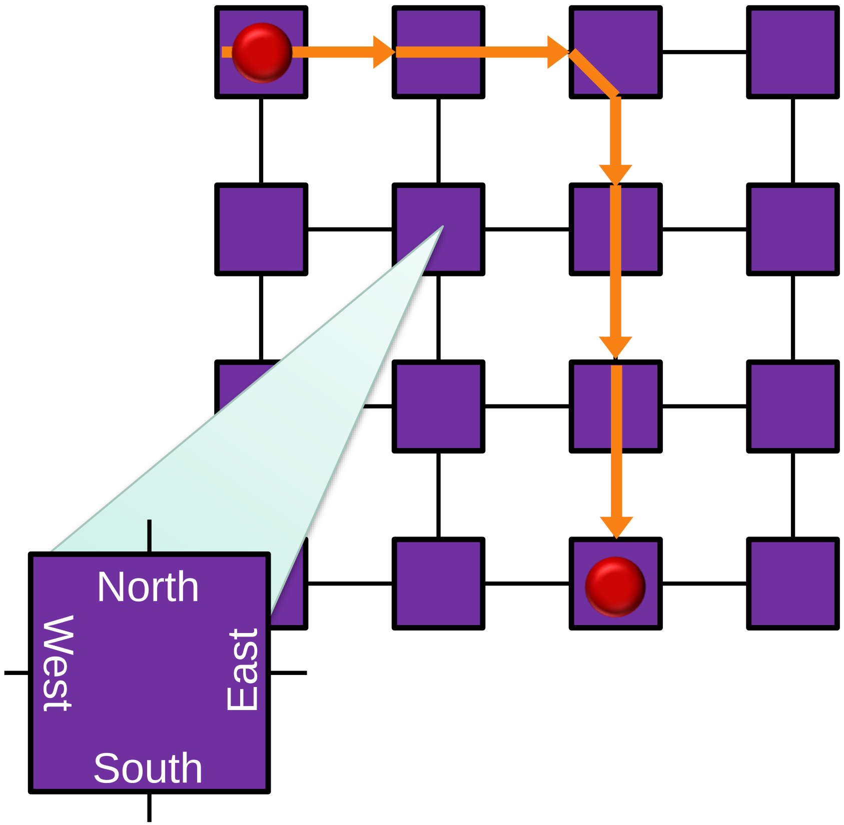


- Nodes have to know their position within the network
- Address has to be mapable to the location within the network (coordinates)

+ Simple and small algorithm
+ Deadlock free
− Only works for meshes (rectangular)

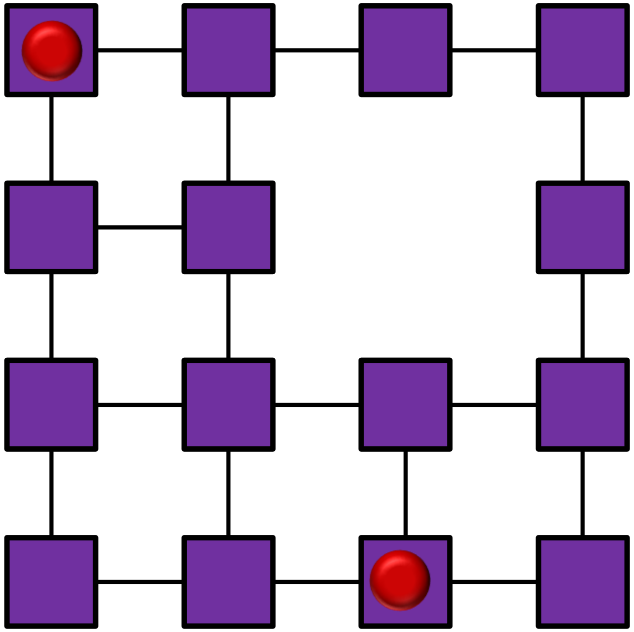
XY-Routing Implementation





X-Y Routing: Limitations

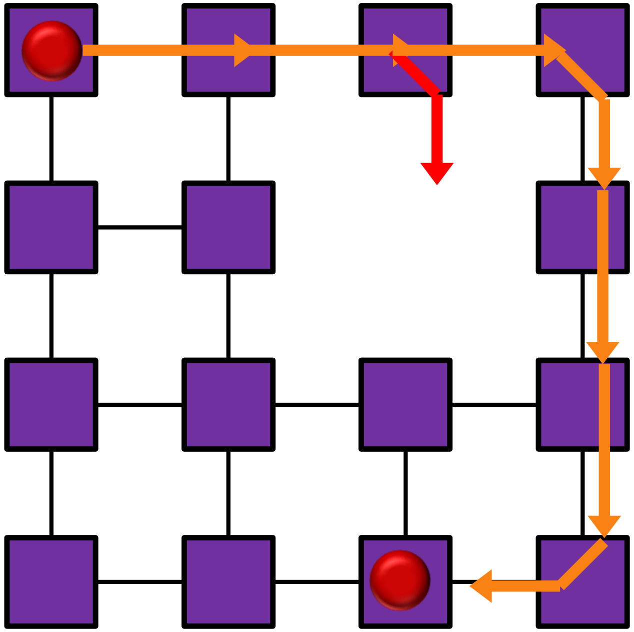
- How can the destination be reached in case of a missing or faulty node?



- Use Deflection Routing / “Hot Potato” Routing
  - Dynamic, Non-Minimal

Deflection Routing / "Hot Potato" Routing

- Packet will be routed to an output port according to a algorithm.
- If that is busy or not available another port is selected.

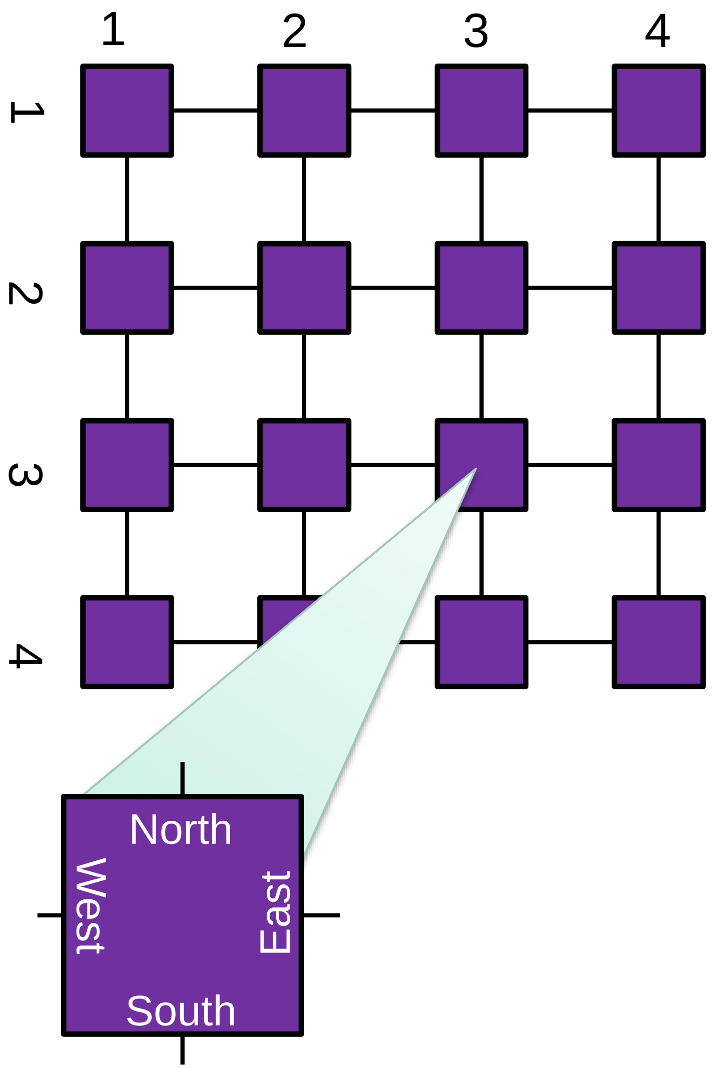


- Packet is not stored in the next router. It is constantly forwarded until it reached the destination
  - Packet is bounced around in the network like a "hot potato"

+ When certain paths are busy/ unavailable, packets can choose alternate paths
+ Deadlock Free
− Packet can travel away from the destination. Livelocks can occur.

- Priority can be assigned to certain paths to avoid livelocks.

Odd-Even Turn Routing



- Specialized XY Routing
- Columns and Rows are numbered
- Two rules apply:
  1. No East-North Turns in even columns
     No East-South Turns in even columns
  2. No South-West Turns in odd columns
     No North-West Turns in odd columns

- Adaptive routing:
  - Deadlock free
  - Load Balancing is possible
  - Error Tolerant

- Routing algorithm has to take care of reachability

Dijkstra-Algorithm: Properties

- Presented 1959 by Edsger W. Dijkstra (1930-2002)

- Finds the shortest path to every node from a given starting node

- Application for non-mesh-based networks
  - Distance may not be the only criteria

- Djikstra can compute routes that are optimal regarding given constraints (weights) other than pure Manhattan distance
  - Length
  - Bandwidth
  - Cost
  - …

Dijkstra-Algorithm

- Finds the shortest path from a starting node to all other nodes in a known topology

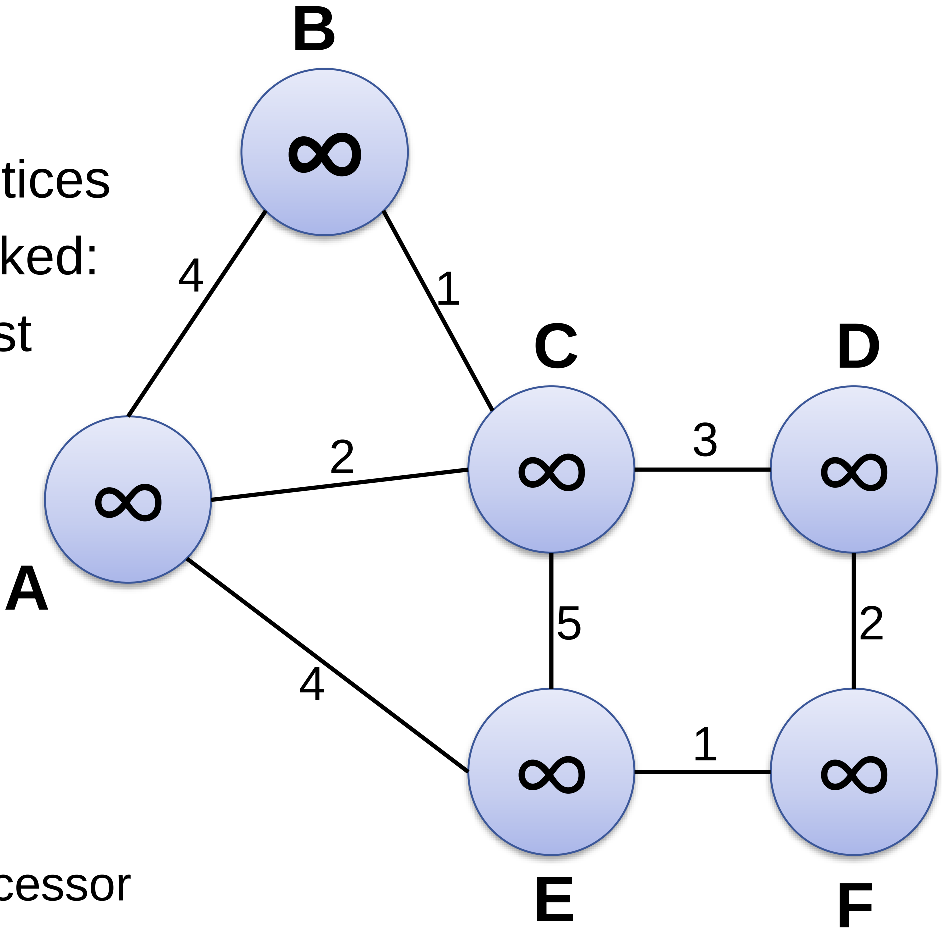
A. Set distance to infinity for all vertices

B. Repeat until all vertices are marked:

   1. Mark the vertex with smallest distance
   2. For all neighbors add current distance and edge weight
   3. If sum is smaller than current distance

      → update distance

      → set marked vertex as predecessor

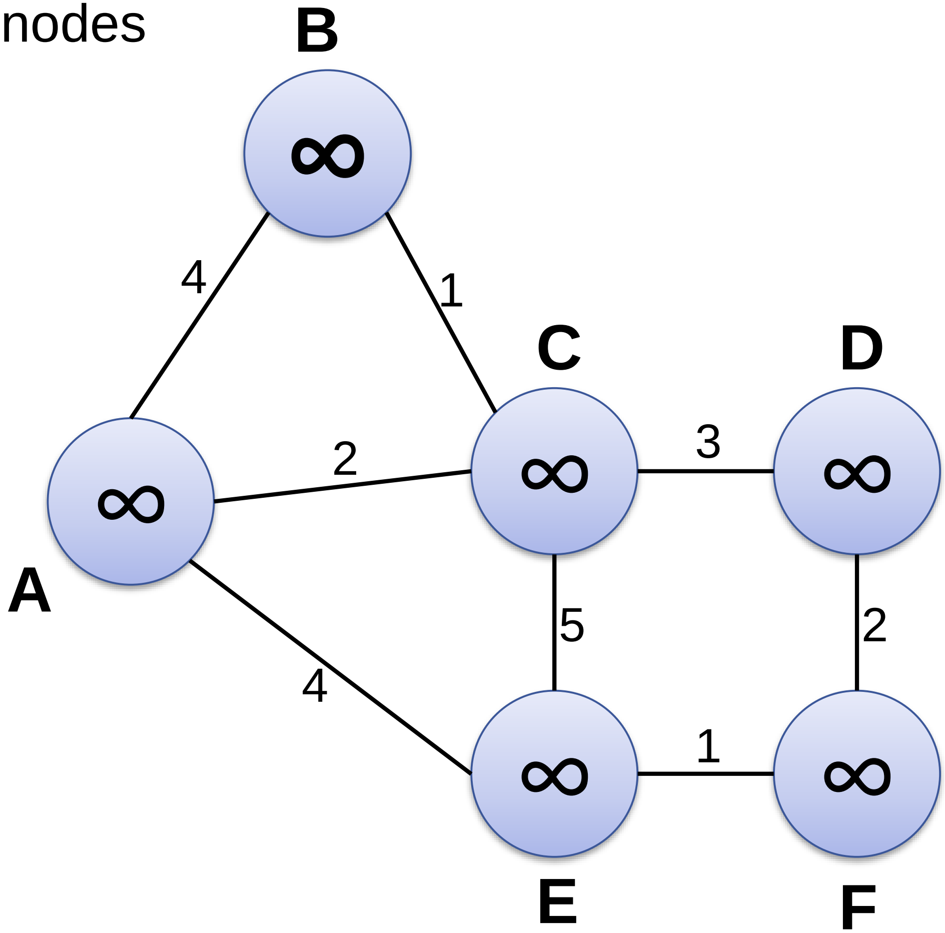


Dijkstra-Algorithm - Example

Step 1

Vertex	Dist.	Pred.
A	0	A
B	∞	-
C	∞	-
D	∞	-
E	∞	-
F	∞	-

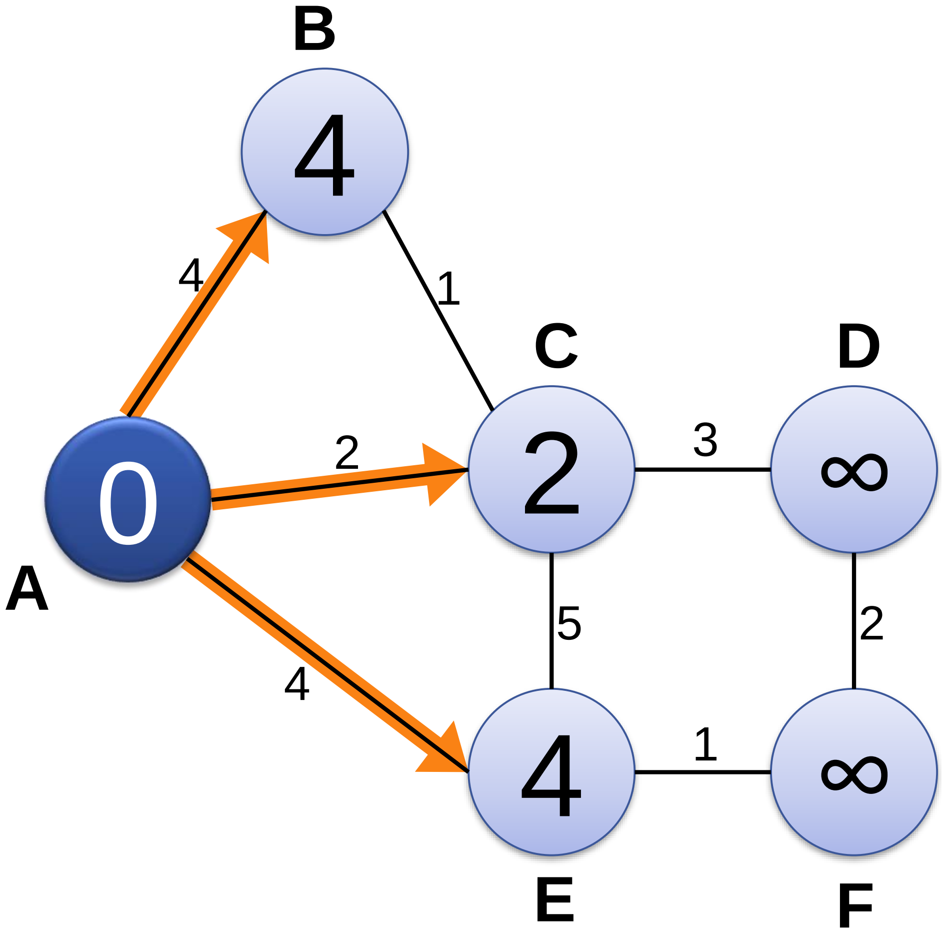
- Find the shortest path from node A to all other nodes



Dijkstra-Algorithm - Example

Step 2

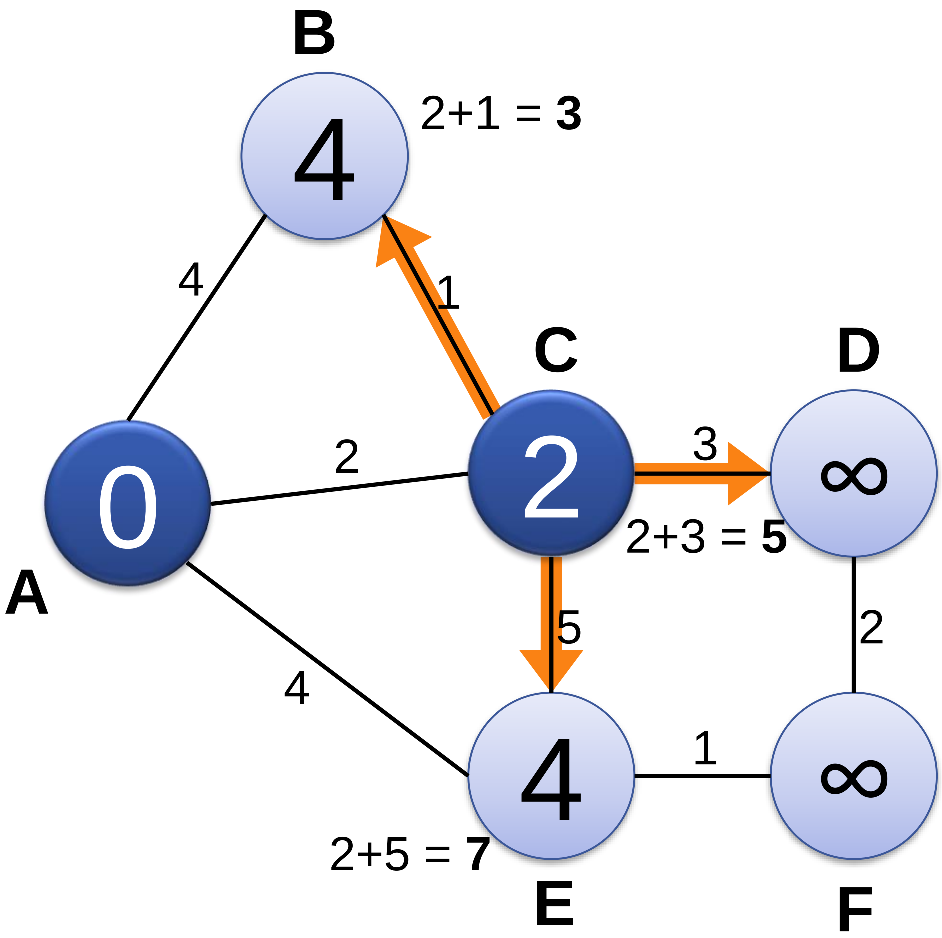
Vertex	Dist.	Pred.
A	0	A
B	4	A
C	2	A
D	∞	-
E	4	A
F	∞	-



Dijkstra-Algorithm - Example

Step 3

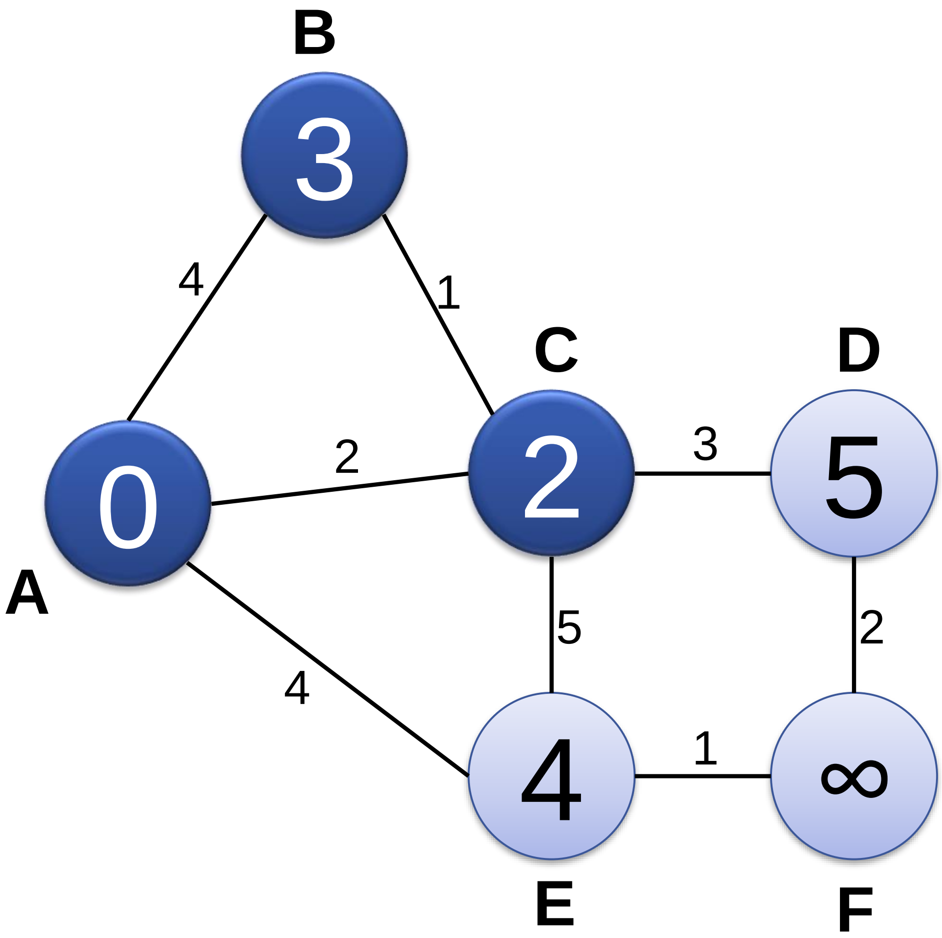
Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	∞	-



Dijkstra-Algorithm - Example

Step 4

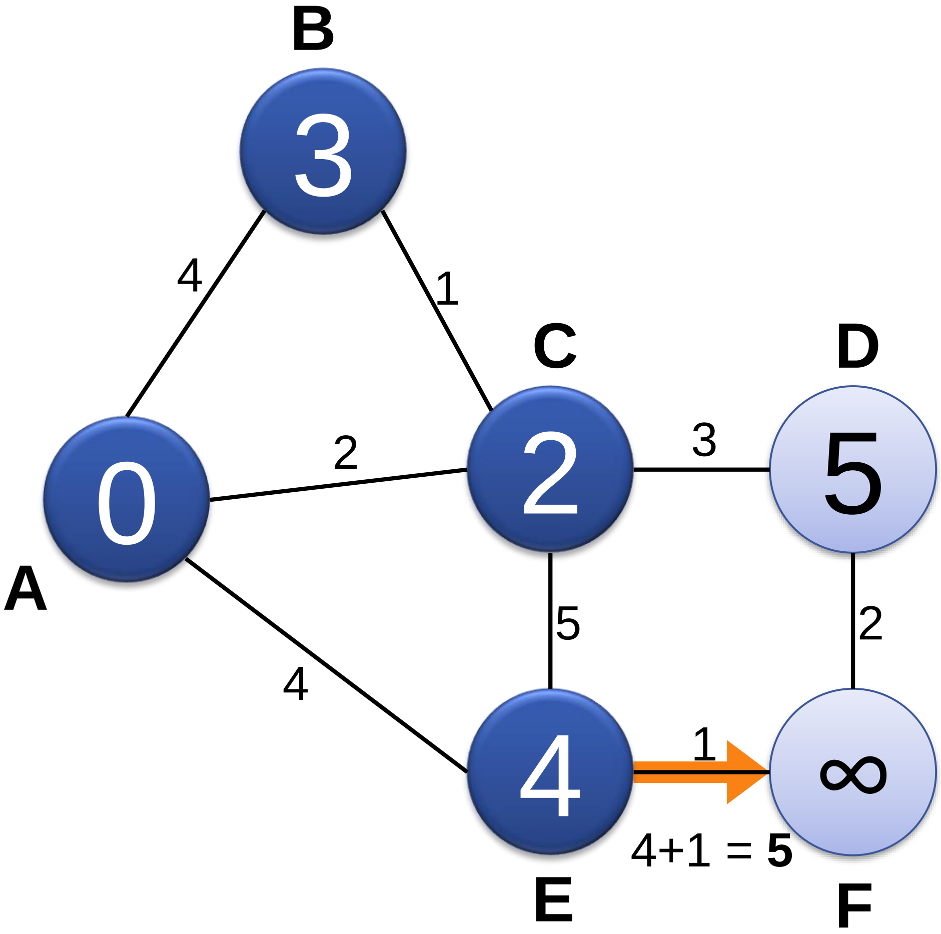
Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	∞	-



Dijkstra-Algorithm - Example

Step 5

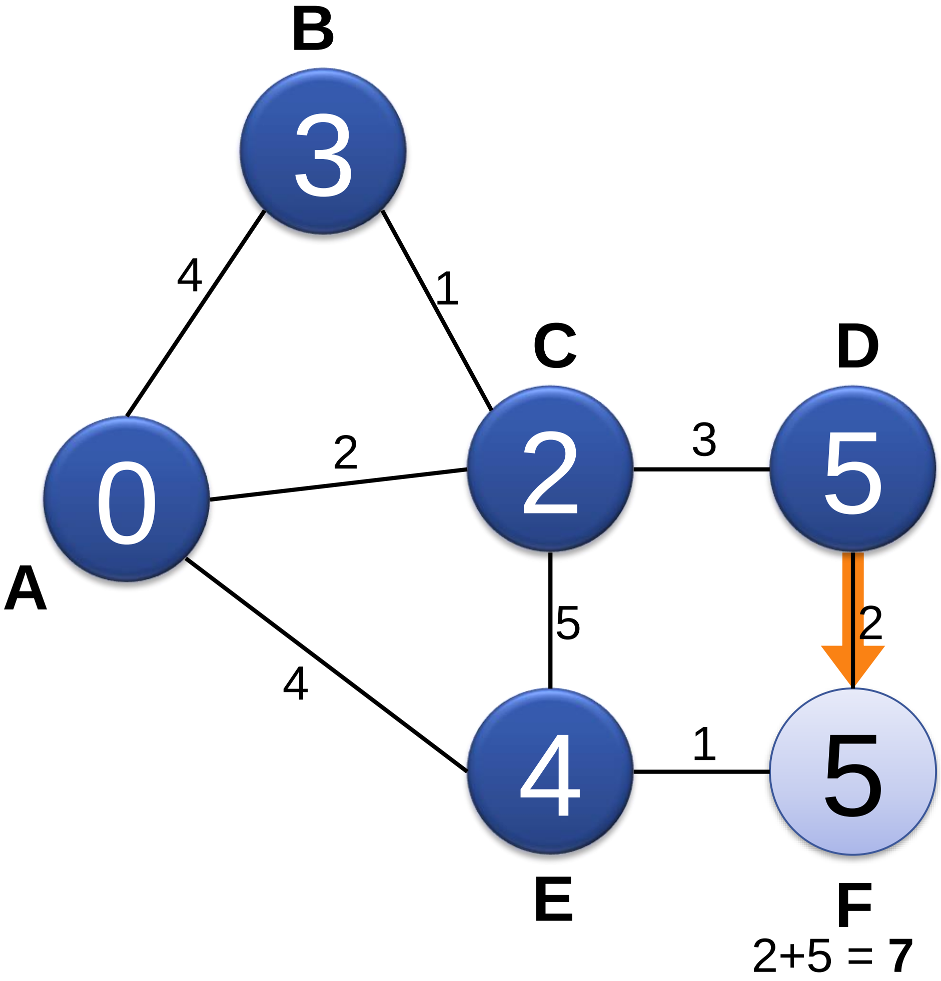
Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Dijkstra-Algorithm - Example

Step 6

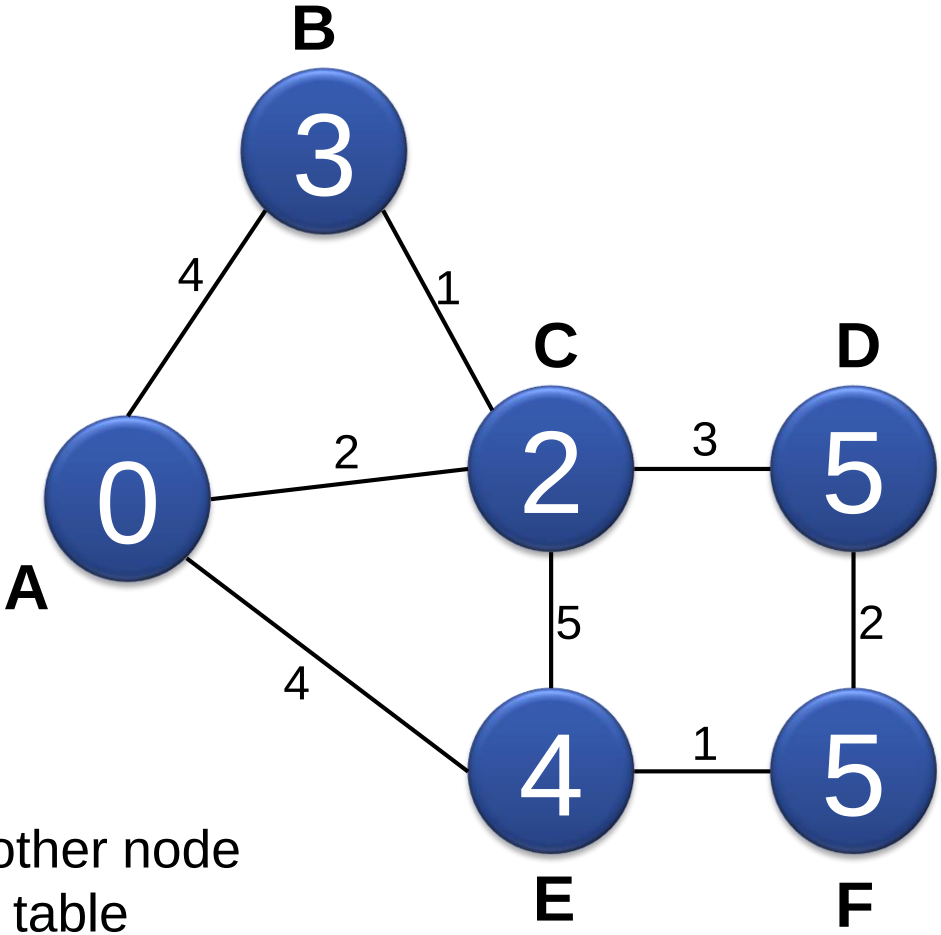
Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



Dijkstra-Algorithm - Example

Step 7

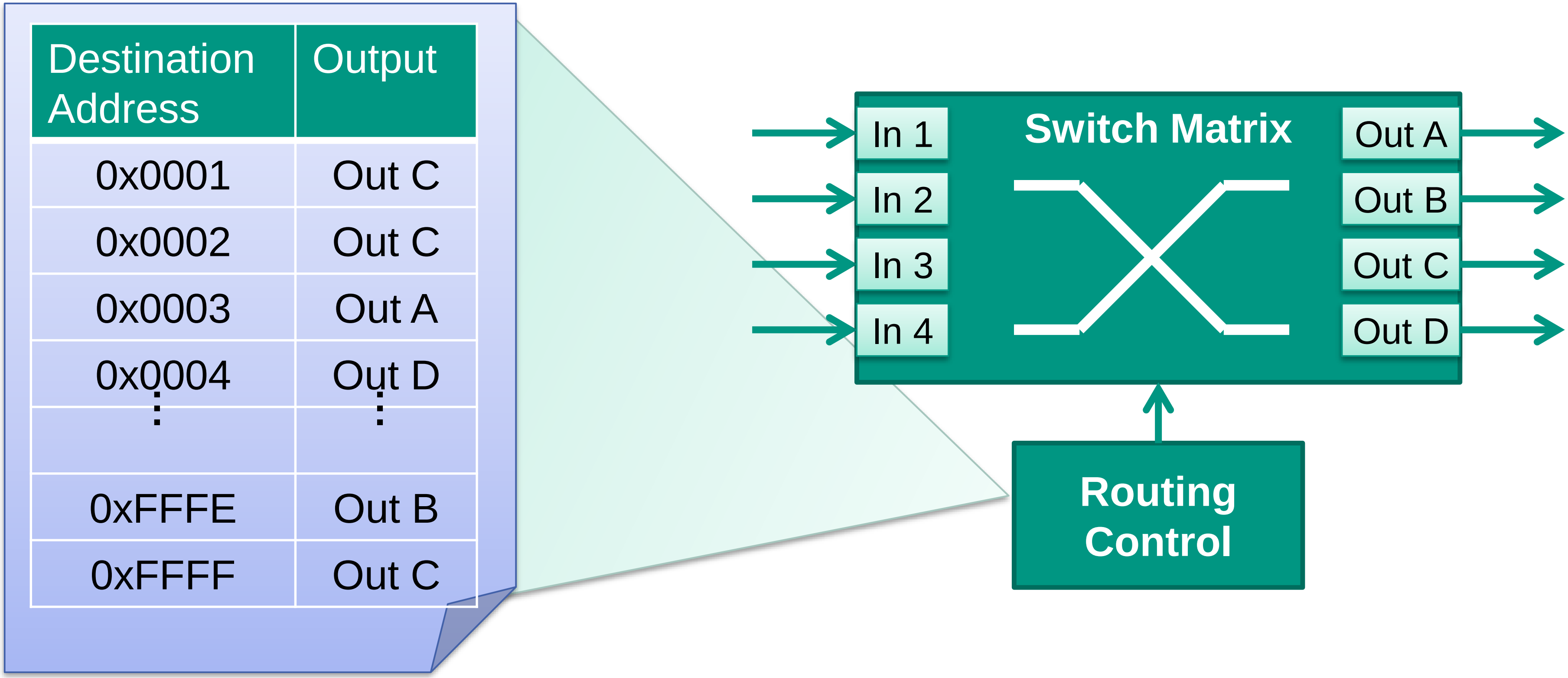
Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E



- The shortest path from A to any other node can be constructed now from the table
- For complete routing table, algorithm is repeated for all nodes

Example for Routing Control – Routing Table

- One row per destination address
- Content stores output port for a given destination
- Table can be static or dynamic
  - Example algorithm: Dijkstra



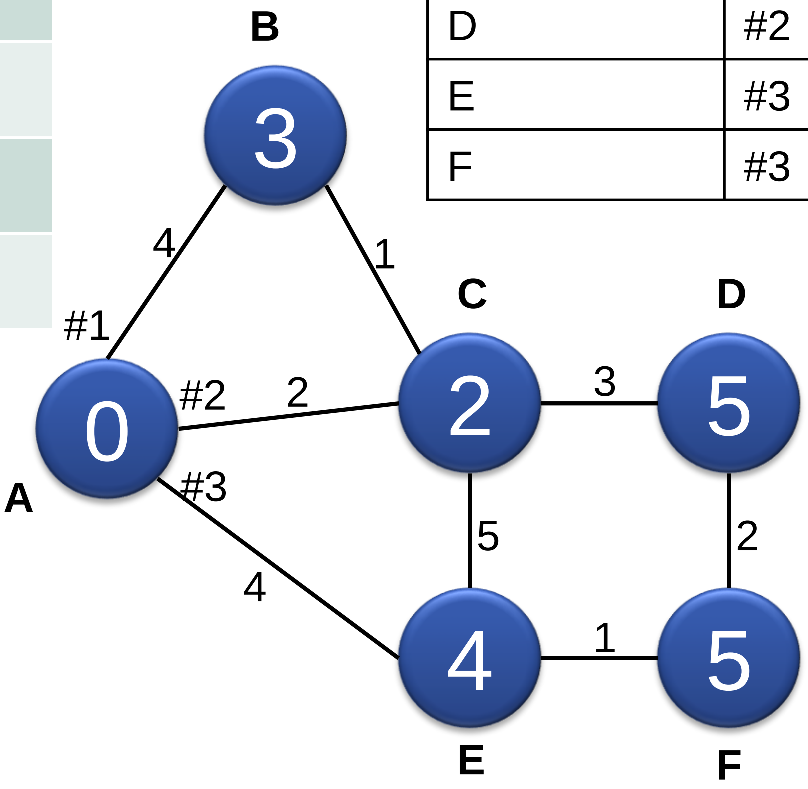
Destination Address	Output
0x0001	Out C
0x0002	Out C
0x0003	Out A
0x0004	Out D
⋮	⋮
0xFFFE	Out B
0xFFFF	Out C

Dijkstra-Algorithm – Example

Vertex	Dist.	Pred.
A	0	A
B	3	C
C	2	A
D	5	C
E	4	A
F	5	E

Routing table for node A

Destination	Output port	Path
B	#2	A -> C -> B
C	#2	A -> C
D	#2	A -> C -> D
E	#3	A -> E
F	#3	A -> E -> F



Virtual Channels

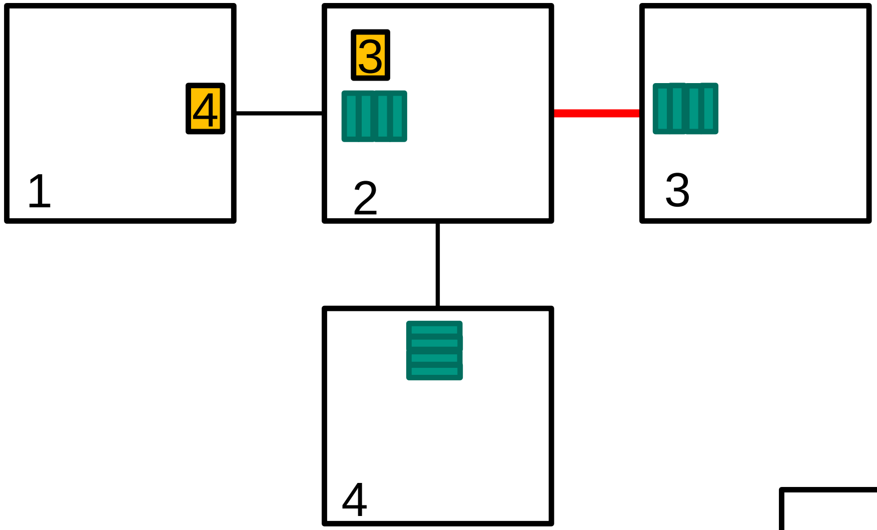
Motivation for virtual channels

- In larger networks, messages might have to wait for other messages to be transported
  - Link is blocked for circuit switching
  - Message buffer is blocked for packet switching

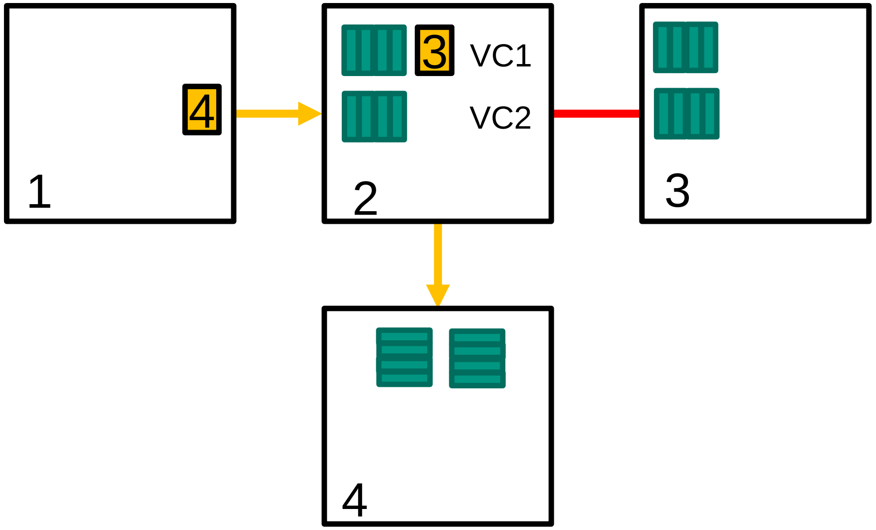
- Bad link utilization

- Solution: Share physical link between multiple communications

Example



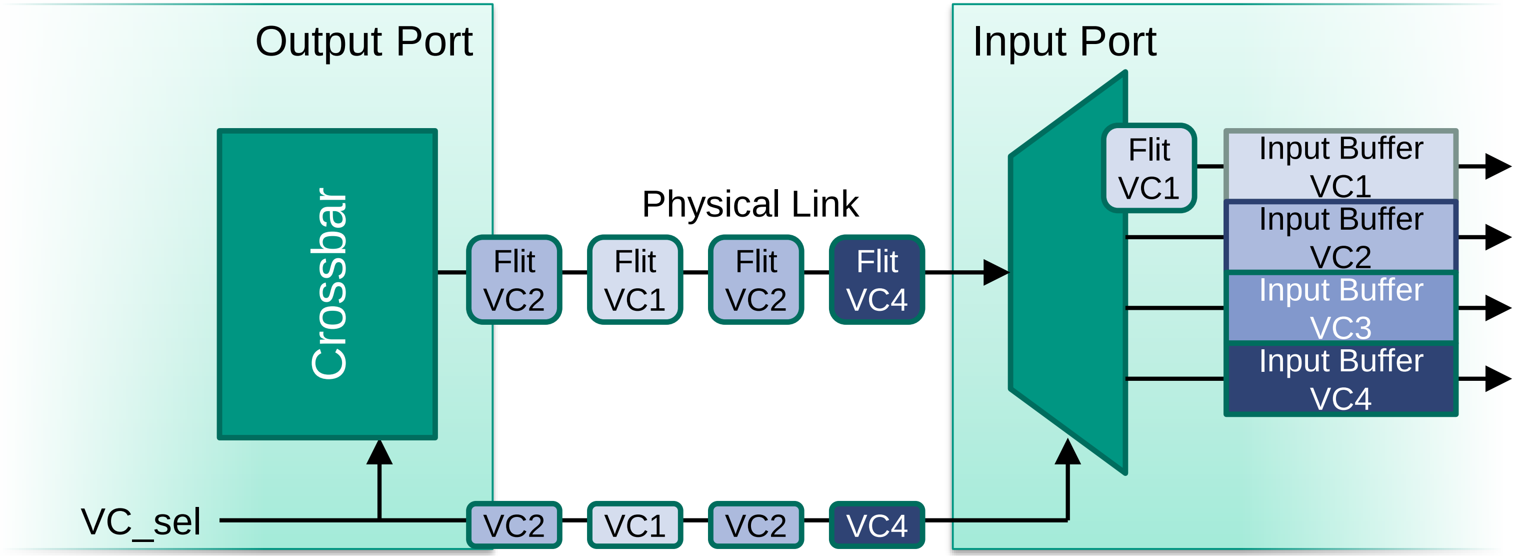
- Packet with destination ‘3’ cannot move forward as the link is ‘busy’
- Packet with destination ‘4’ cannot move forward even though the links it requires are free.



- Two Virtual Channels are now present.
- Packet with destination 4 uses VC2, which is free and reaches the destination

▥ Buffer □ Router

Virtual Channels (VC)

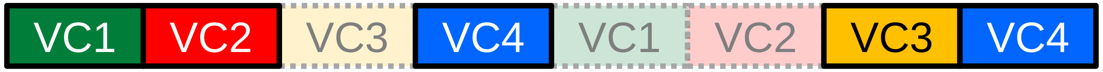


- Time Multiplexing of physical link
- Independent data transmissions on independent virtual channels (VC)
- Internal storage (buffers) in routers/tiles required
- Each node can take independent routing decisions (priorities, buffer levels)
- additional control logic within the routers/node as well as additional control information between nodes

VC-Arbitration - Scheduling

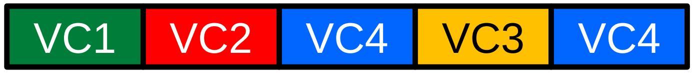
- Network nodes have multiple shared components
  - Routing unit, if packet arrive at different ports and have to leave at same output port
  - Buffers
  - Output ports

  - Arbitration is required → Scheduling
    - Time Division Multiple Access (TDMA)
      - Fixed Slots for every packet



VC1 VC2 VC3 VC4 VC1 VC2 VC3 VC4

    - Round Robin (RR)
      - Fixed order (if no access required, slot can be taken by next request



VC1 VC2 VC4 VC3 VC4

    - Priority
      - Assign priorities to messages

Detailed discussion of scheduling see lecture Systems and Software Engineering (Prof. Sax)

Thank you for your attention