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 6: Bus systems

Clicker Session: Recapitulation

- https://arsnova.eu/mobile/#id/33969518

Recapitulation

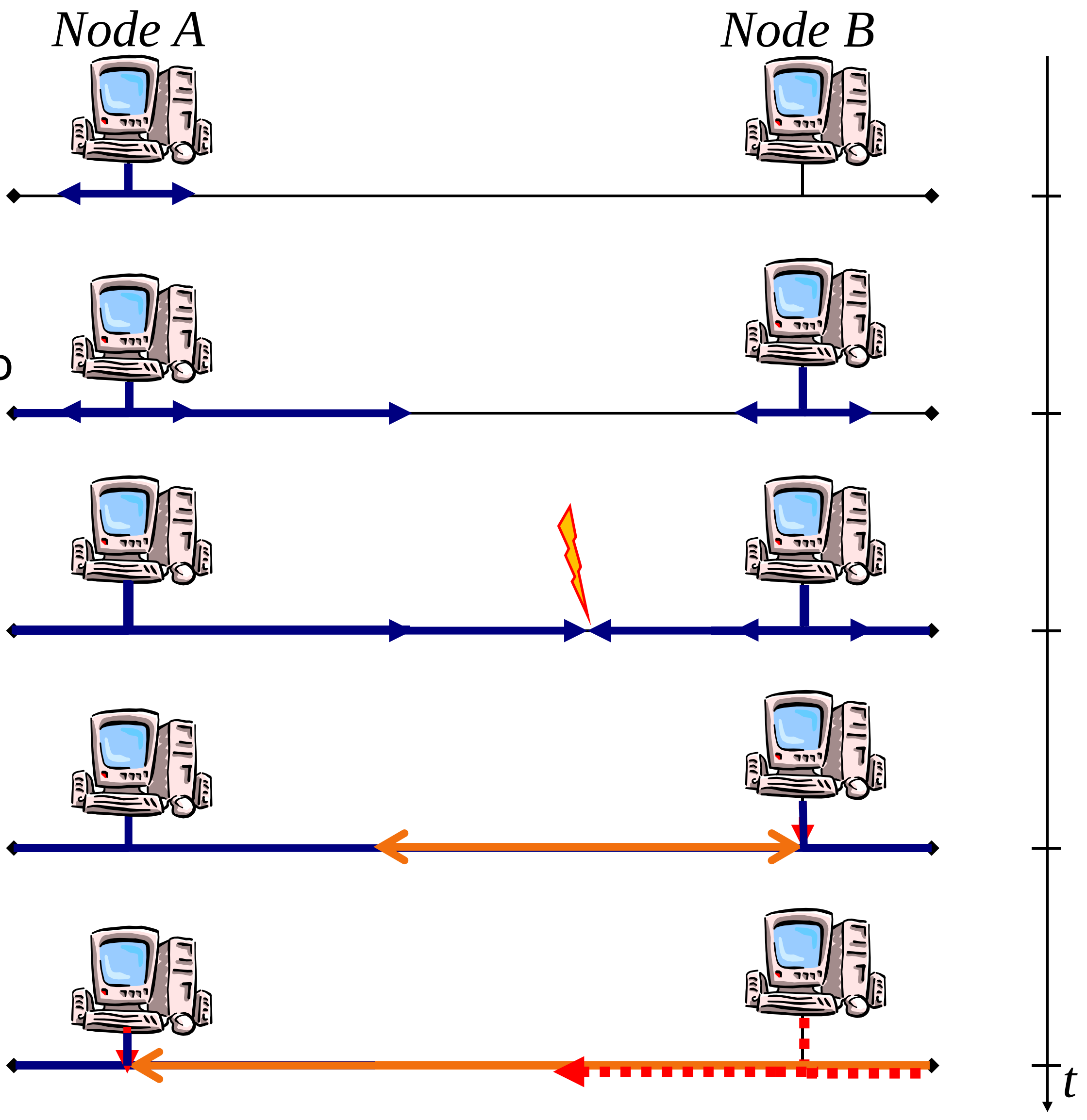
- Multiplexing
  - Walsh functions

- Arbitration schemes
  - Schemes, that require additional wires
    - Daisy chain
    - Polling
    - Tap line
    - Token passing

  - Schemes without additional wires
    - Aloha
    - CSMA / CD

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)

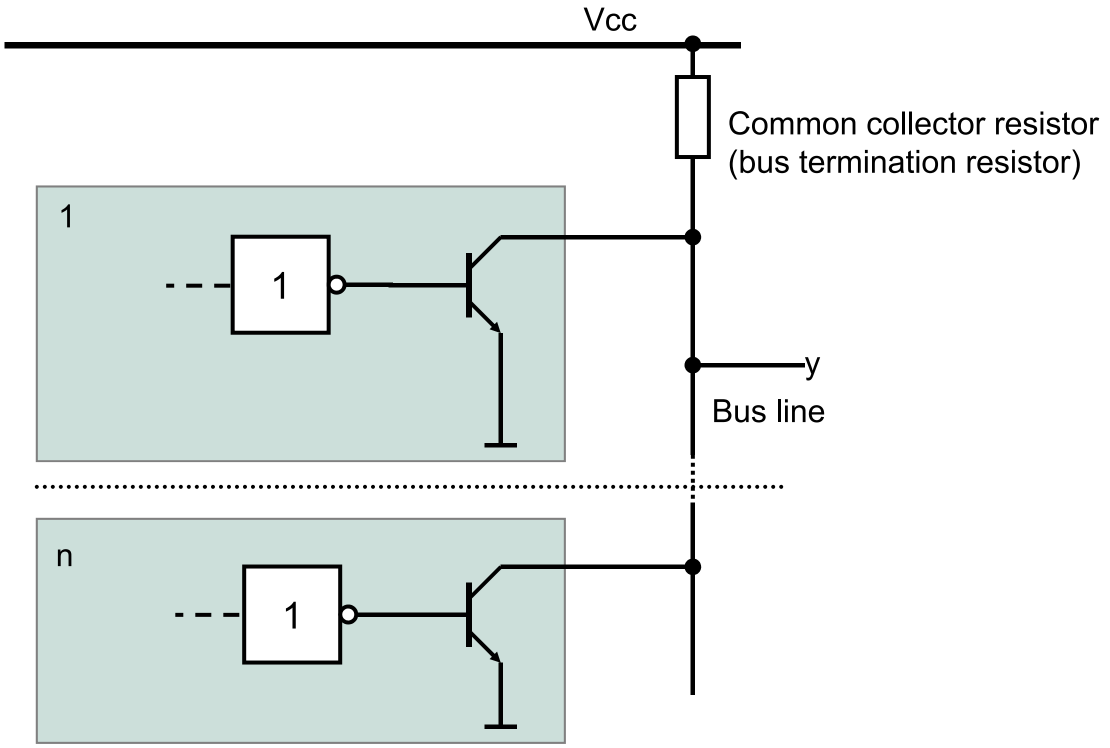
- Avoidance of collisions by priority controlled bus arbitration.
- Every node is assigned an identifier (ID) that equals its priority.

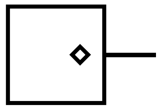
- After completing a transmission on the bus, all nodes with a transmission request start to send their ID. All nodes are connected via wired-OR or via wired-AND respectively.
  - wired-OR: „1“ dominant, „0“ recessive
  - wired-AND: „0“ dominant, „1“ recessive

- 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

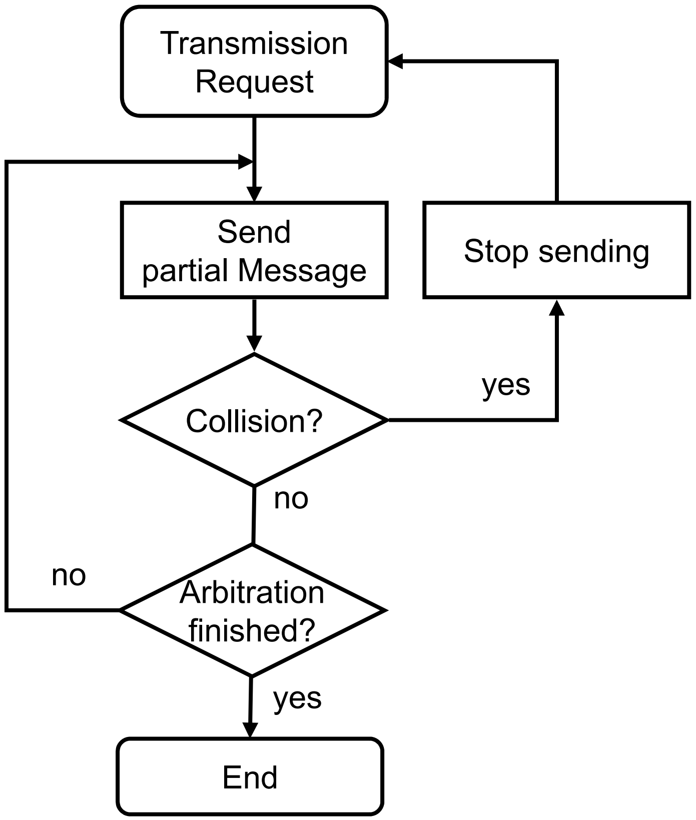




Switch symbol of a
open collector output

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:

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

$l =$ length of the bus line

$v =$ propagation velocity

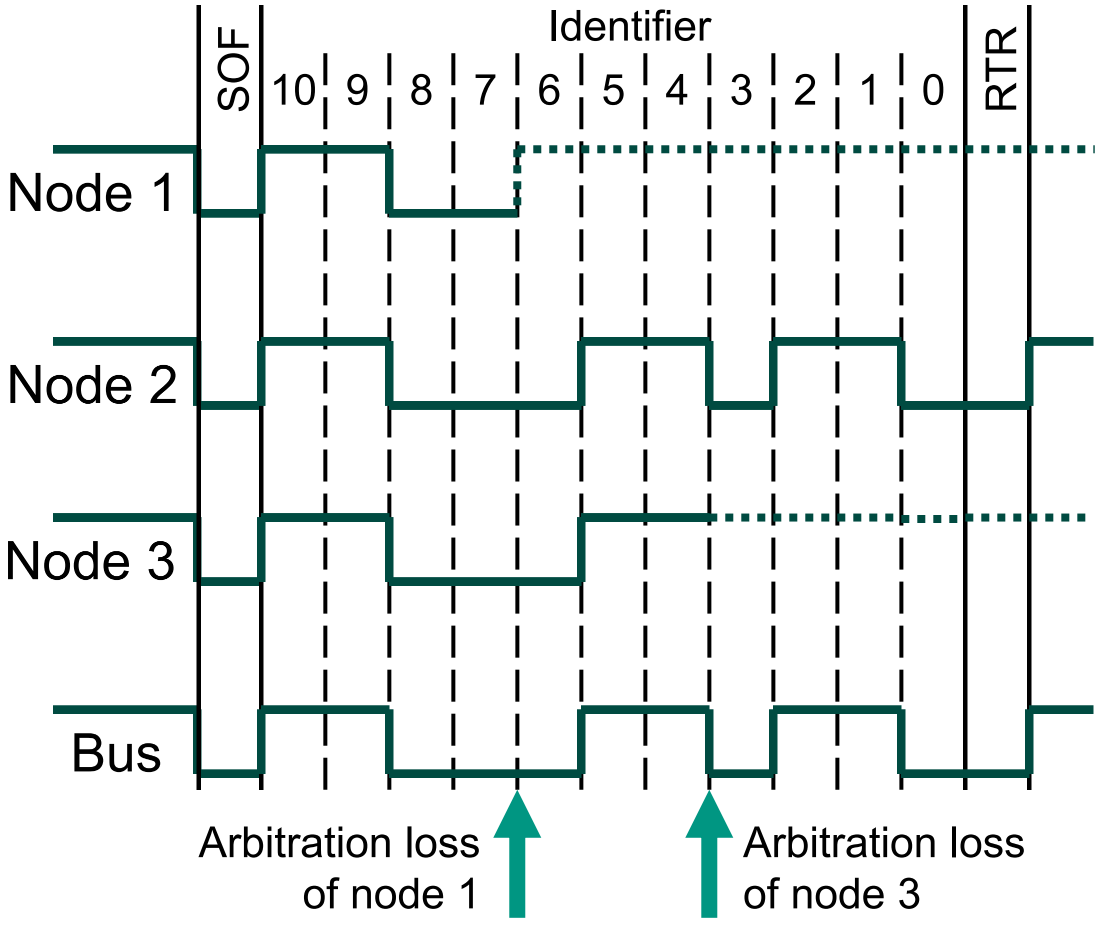
$TR =$ transmission rate

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.

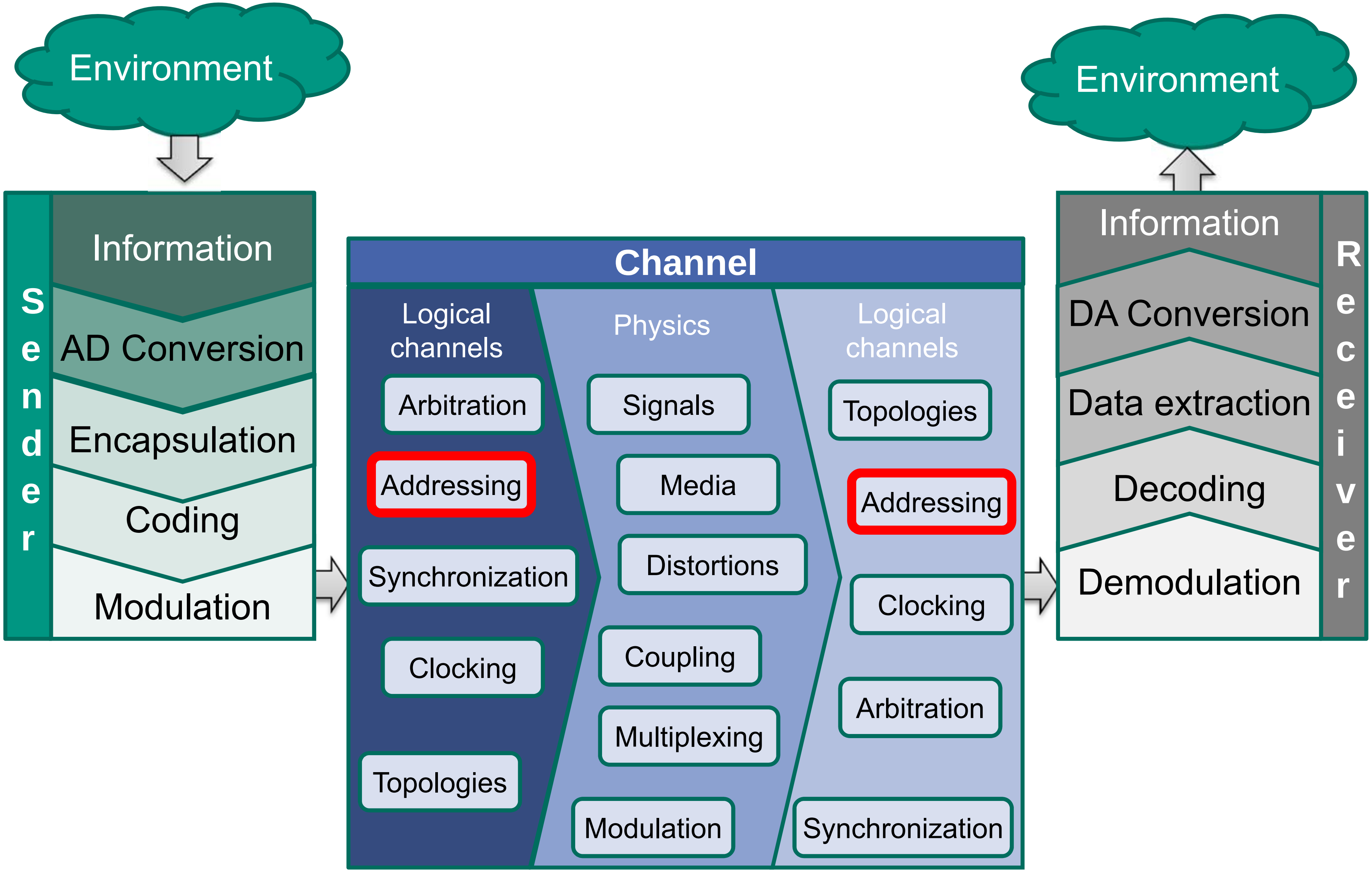
Example: CSMA/CA (CAN-Bus)

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



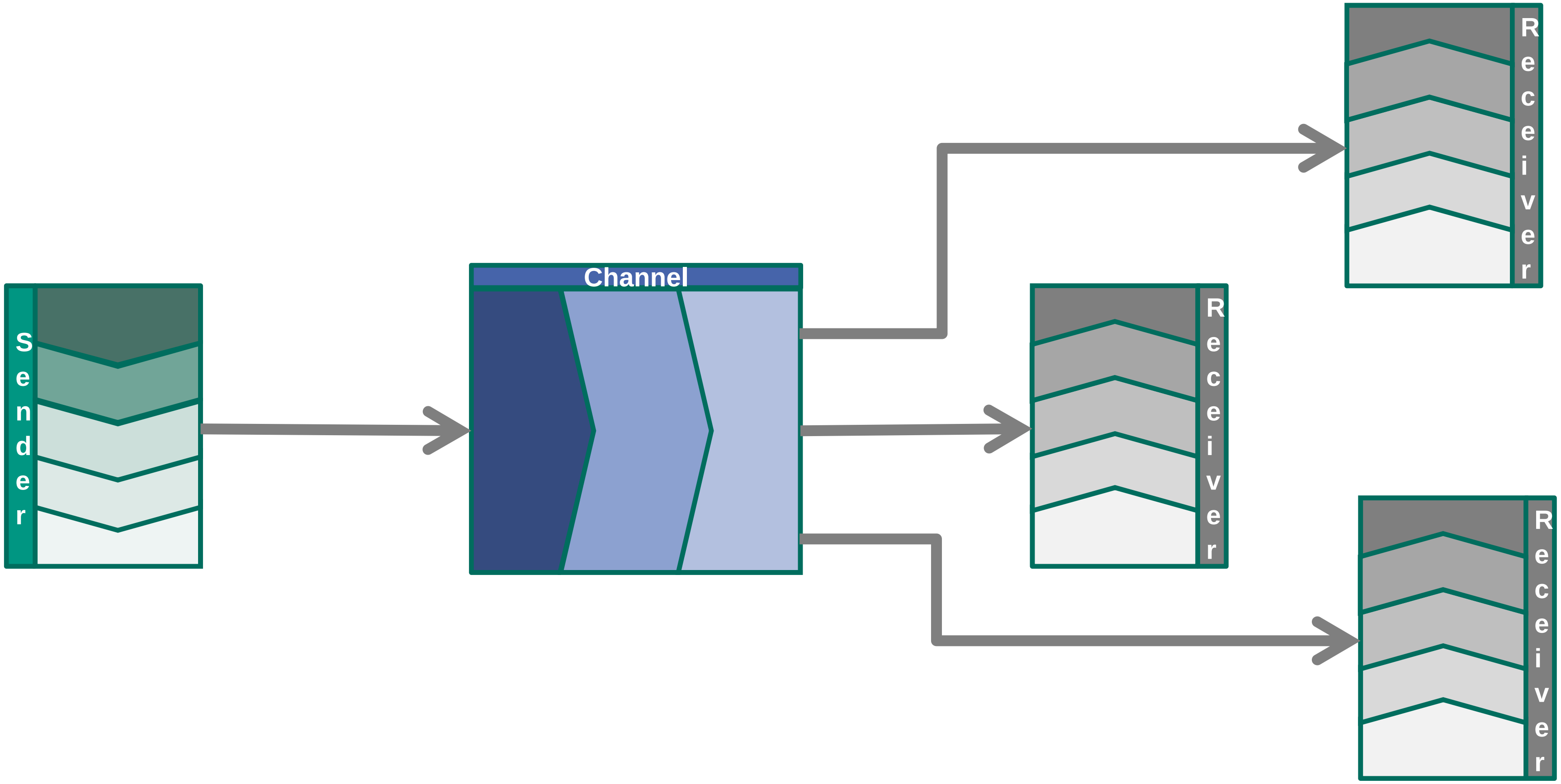
Node 2 is granted bus access sends its complete message

Transmission System – Addressing



Multiple Receivers : Addressing

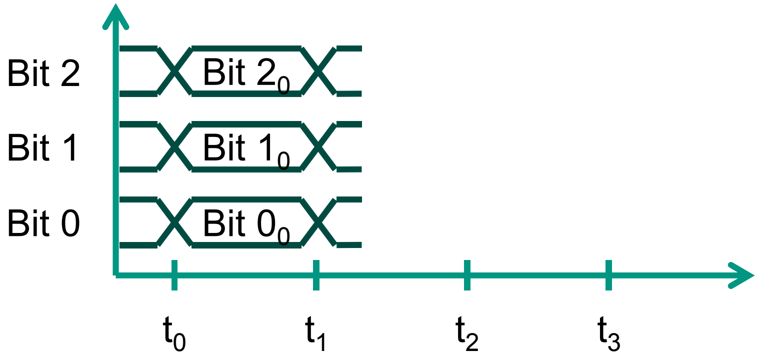
- Multiple receivers can be listening at the same channel
- It has to be distinguished which receiver should receive the actual transmission Addressing



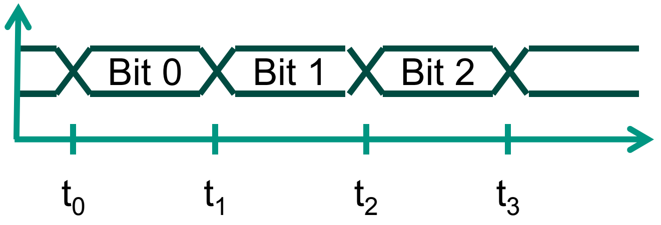
Node Addressing

- 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)

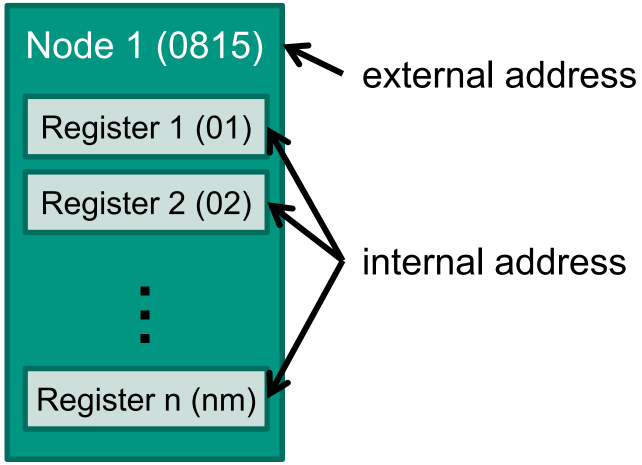


  - Sending the receivers address before sending the data if only one data line is available



Internal Node Addressing

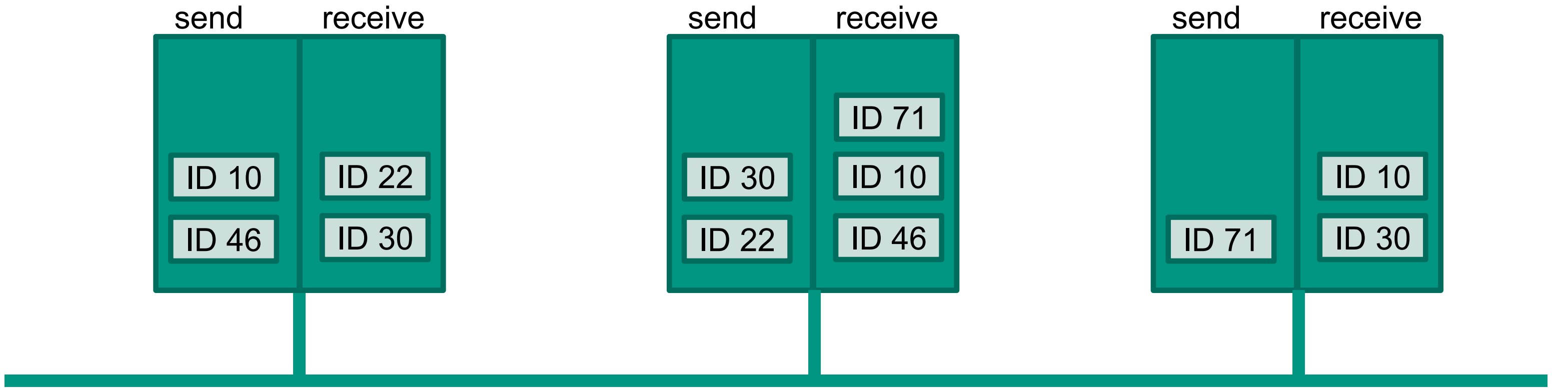
- There can also be additional internal addresses within the receivers that are also transmitted over the communication system
  - External Address: Identify the receiver
  - Internal Address: Identify receiver-internal memory or registers



Message based addressing I

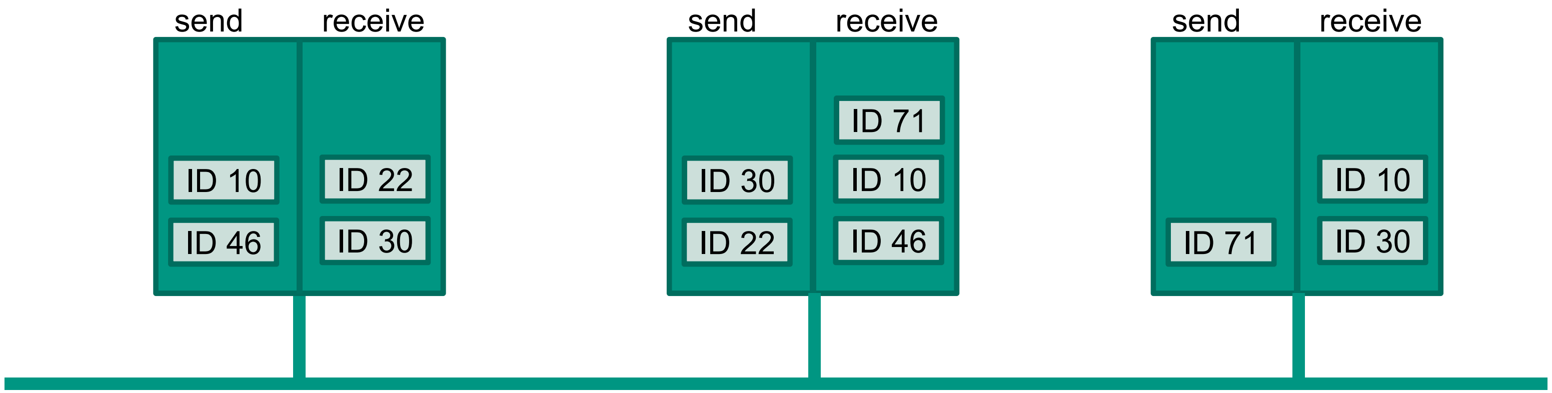
- The address is assigned to individual messages with defined content
  - E.g. temperature reading of a sensor, speed of a car

- 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



send receive
ID 10 ID 22
ID 46 ID 30

send receive
ID 71
ID 30 ID 10
ID 22 ID 46

send receive
ID 10
ID 71 ID 30

- Example: CAN

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

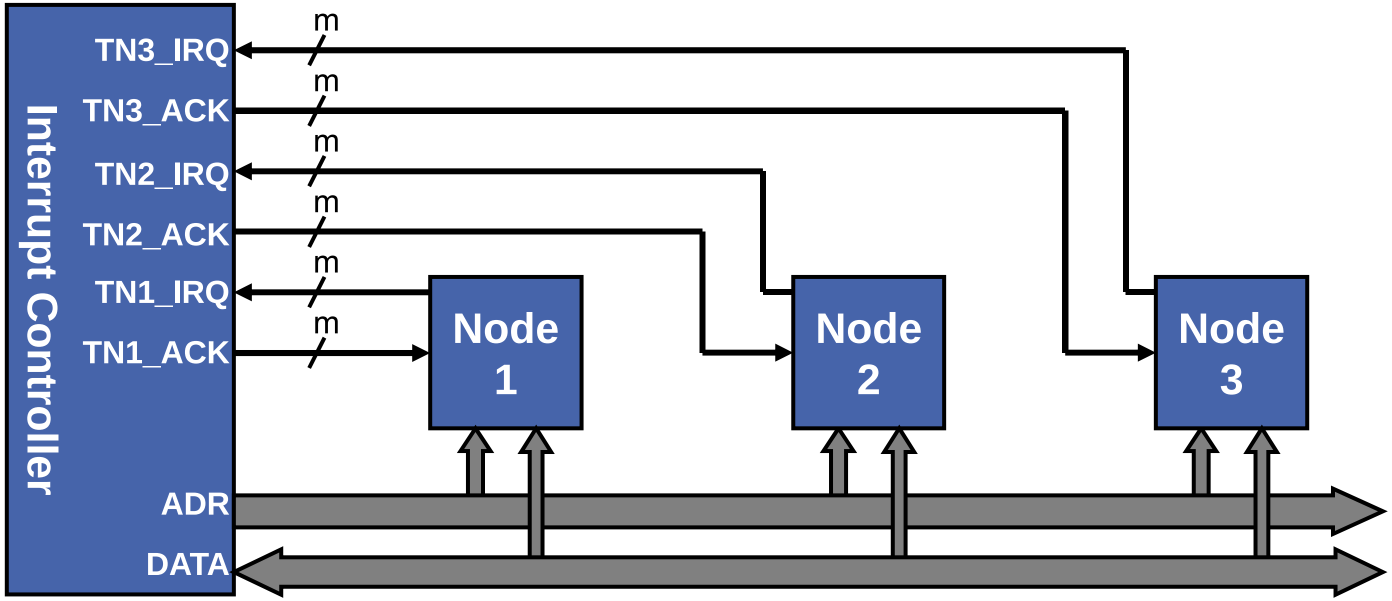
m:	Number of senders
n:	Number of receivers

Interrupts

- In a master-slave bus system, only the masters can actively apply for the bus
- Problem: If a slave has to send important information, it has to wait for a master to be polled
- Solution: Interrupts
  - Dedicated lines that go from the slaves to the masters
  - For serial communication, interrupt handling has to be emulated by cyclic polling of every slave

Examplary Implementation

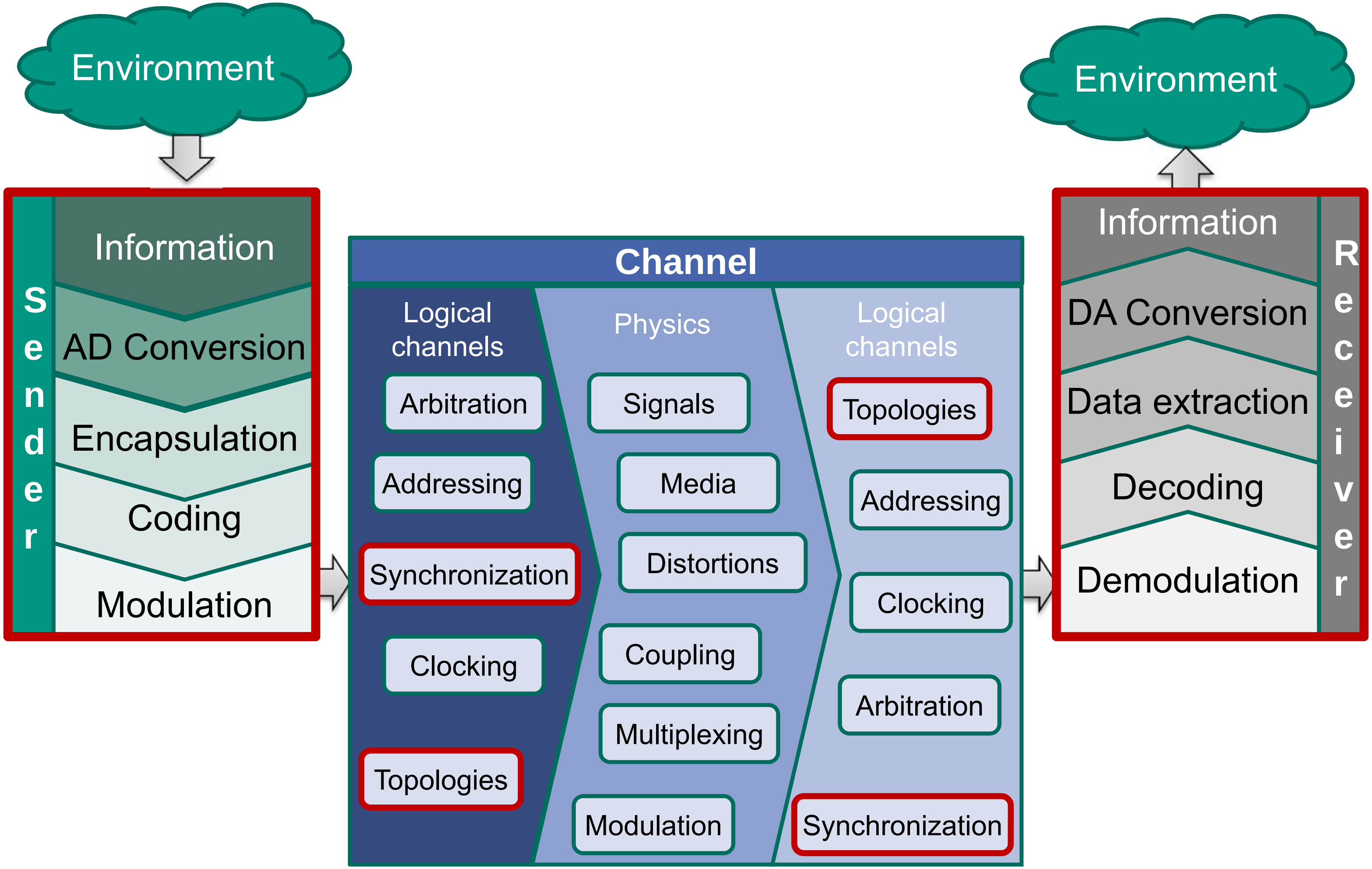
- If a slave has a communication request, it asserts an interrupt
- Interrupt Controller (can be implemented within Master Node)
  - starts a communication session with the slave to service interrupt request
  - acknowledges Interrupt to node
- Multiple interrupt priorities (levels) possible



Clicker Session: Arbitration

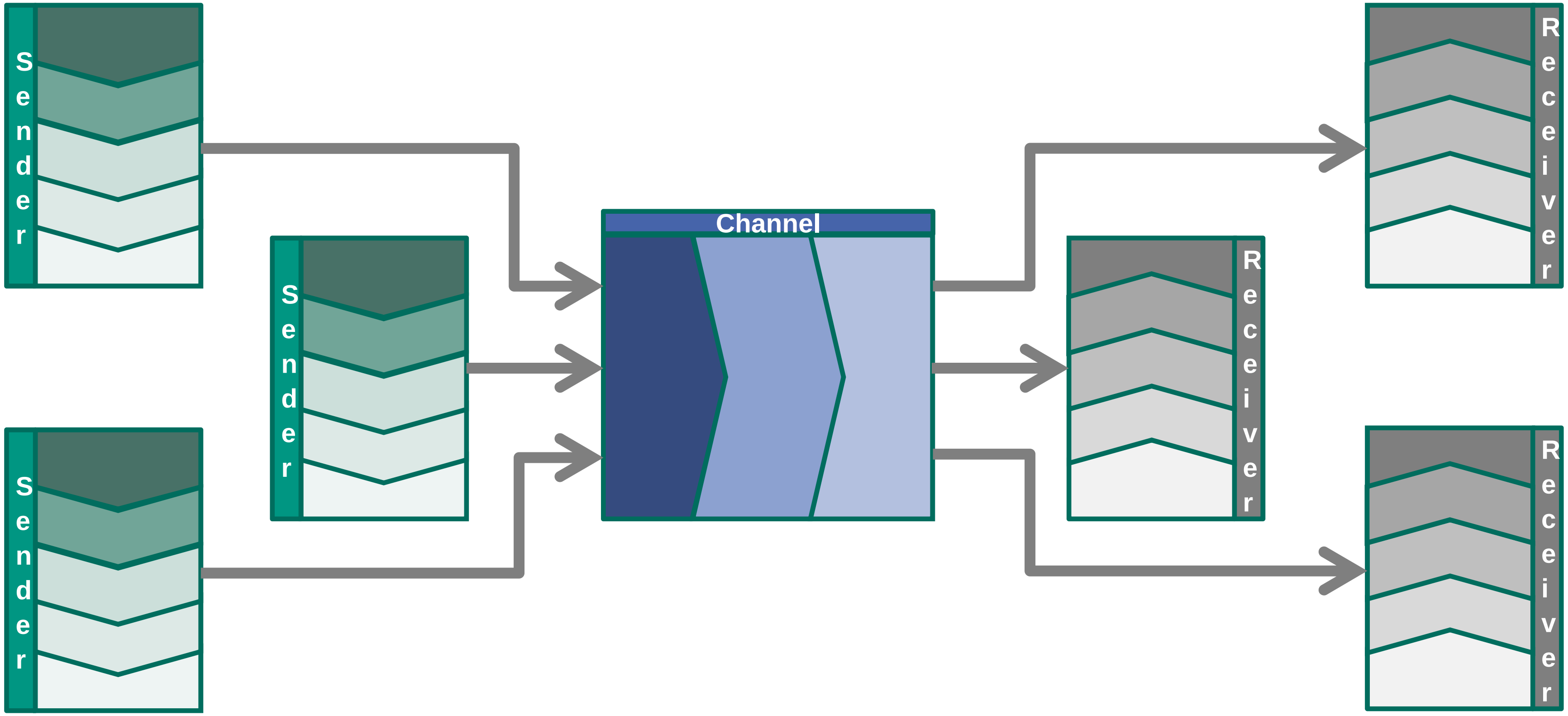
- https://arsnova.eu/mobile/#id/33969518

Transmission System



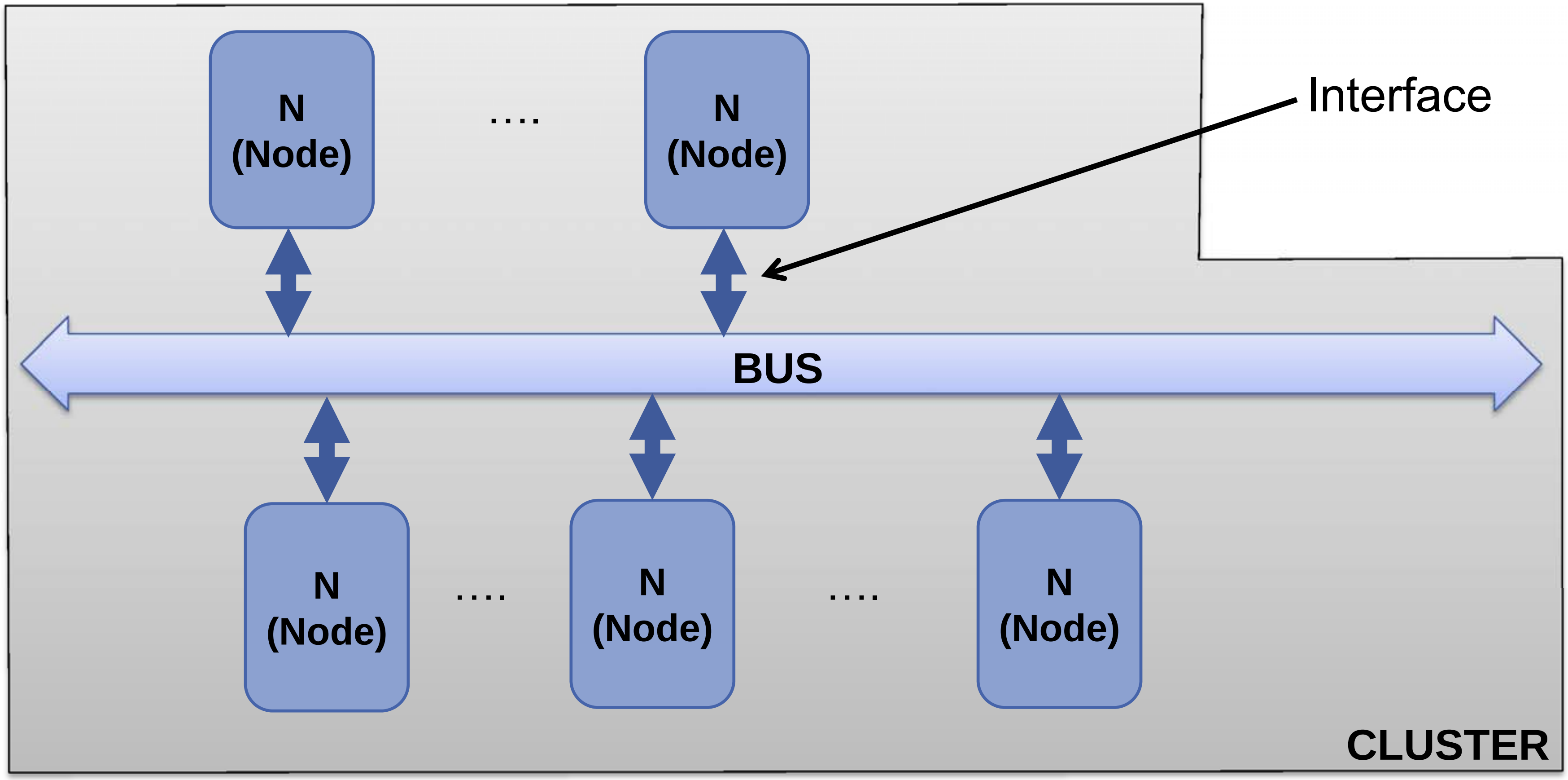
Bus Systems

- 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)

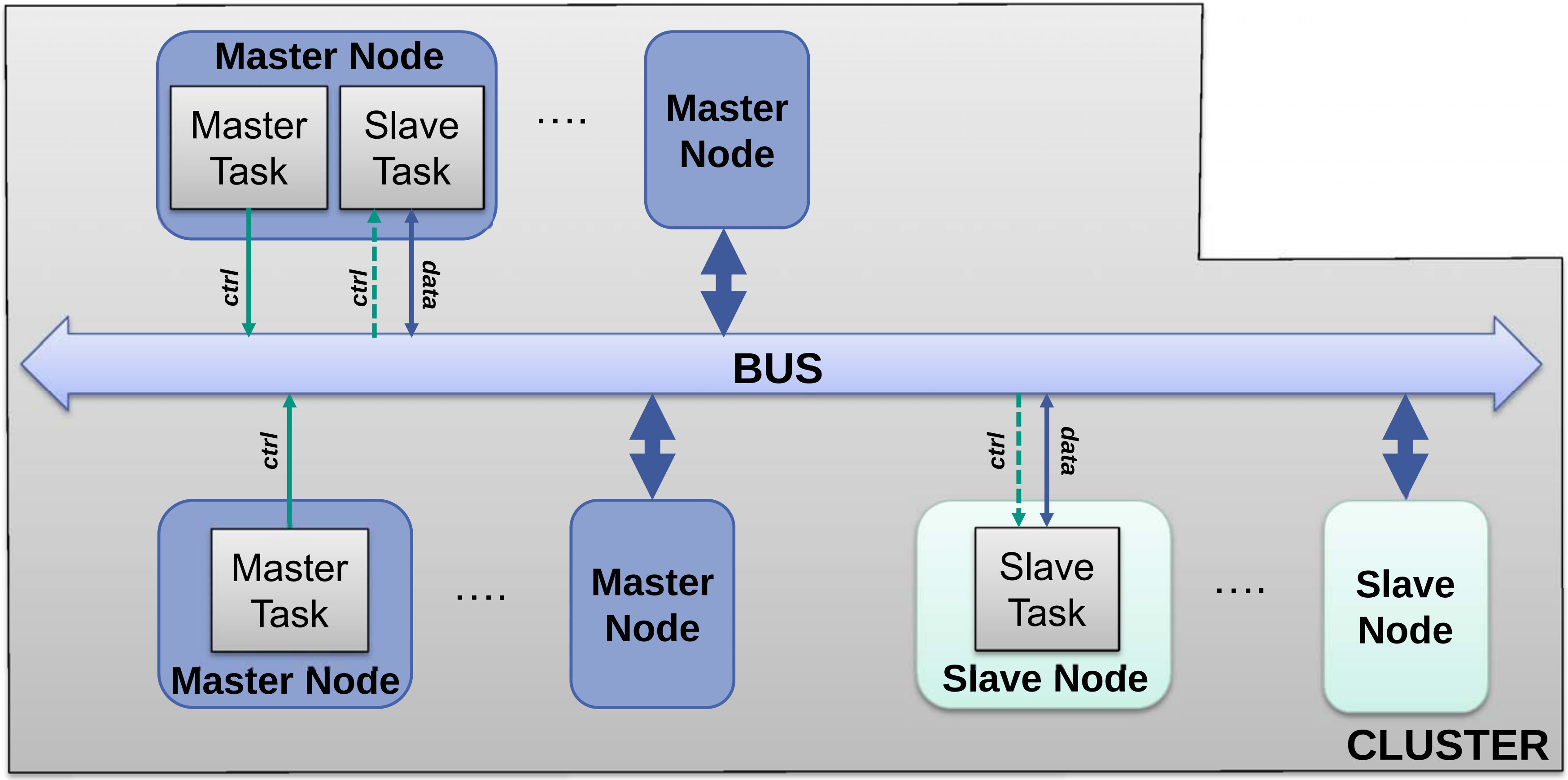
  (2): a group of loosely coupled computers that work together closely

  (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

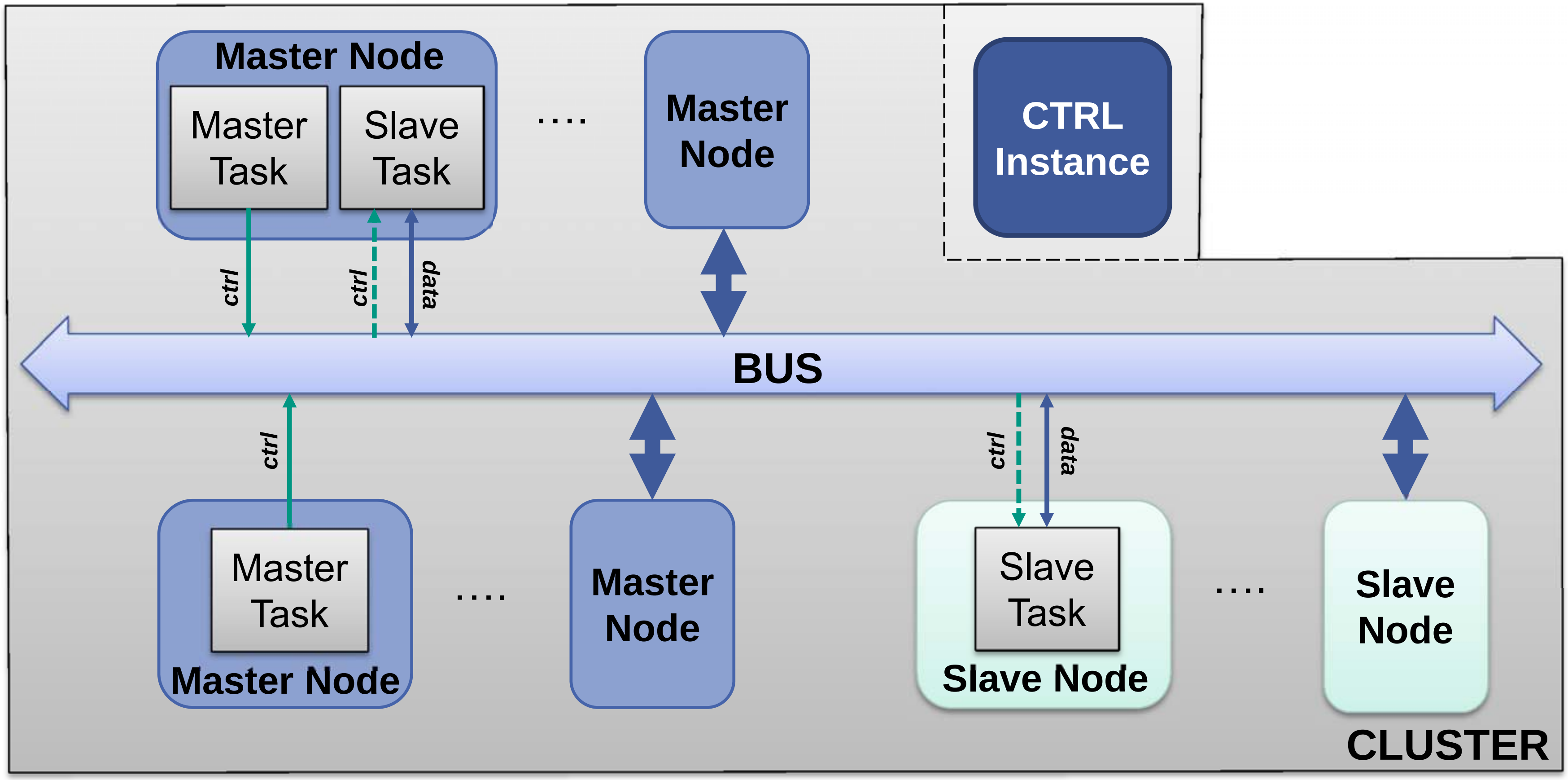
- Master Task
  - active entity
  - controls the bus
  - triggers communication
  - Arbitration

- 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

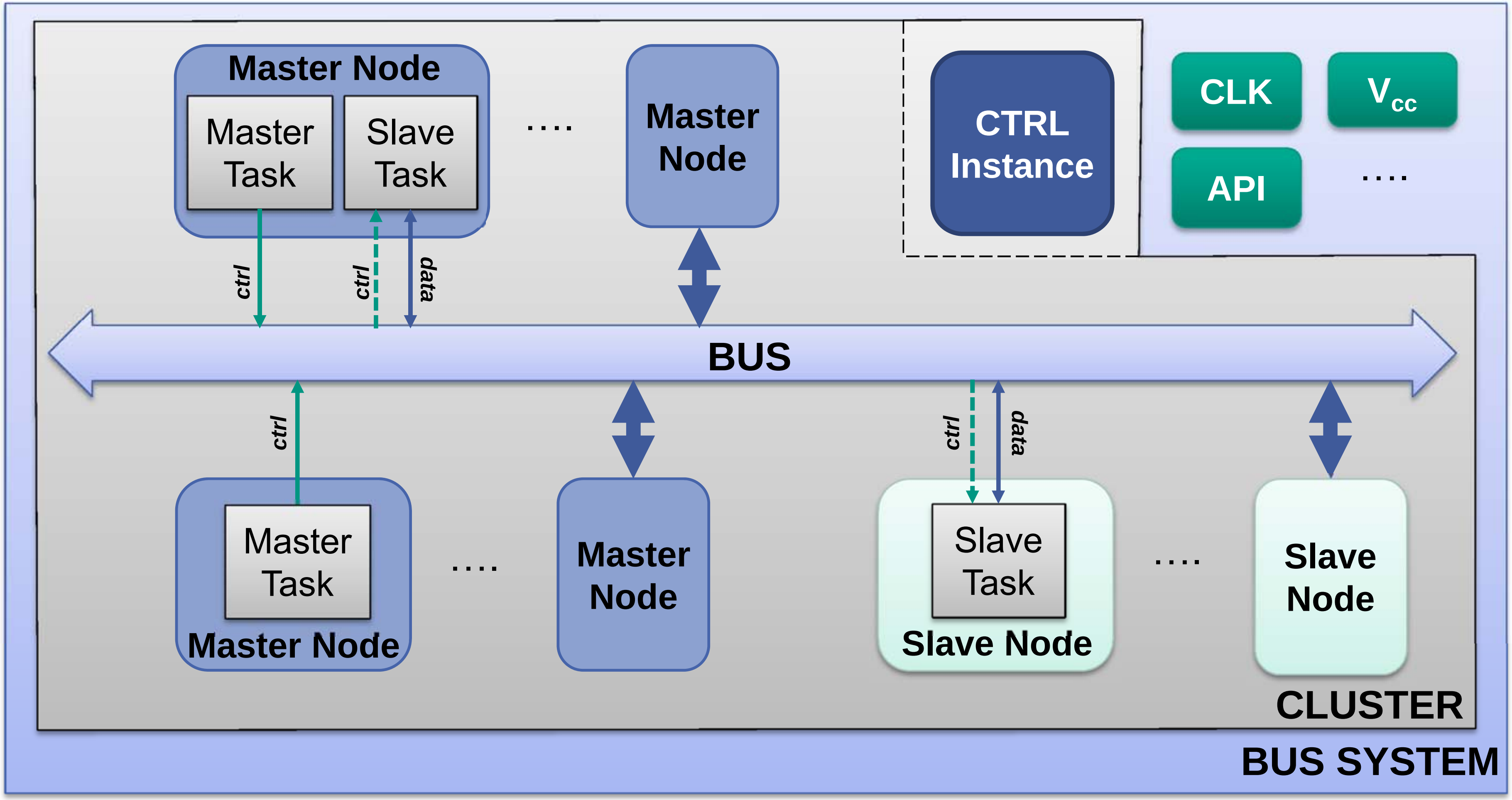
General Bus Structure (2)

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

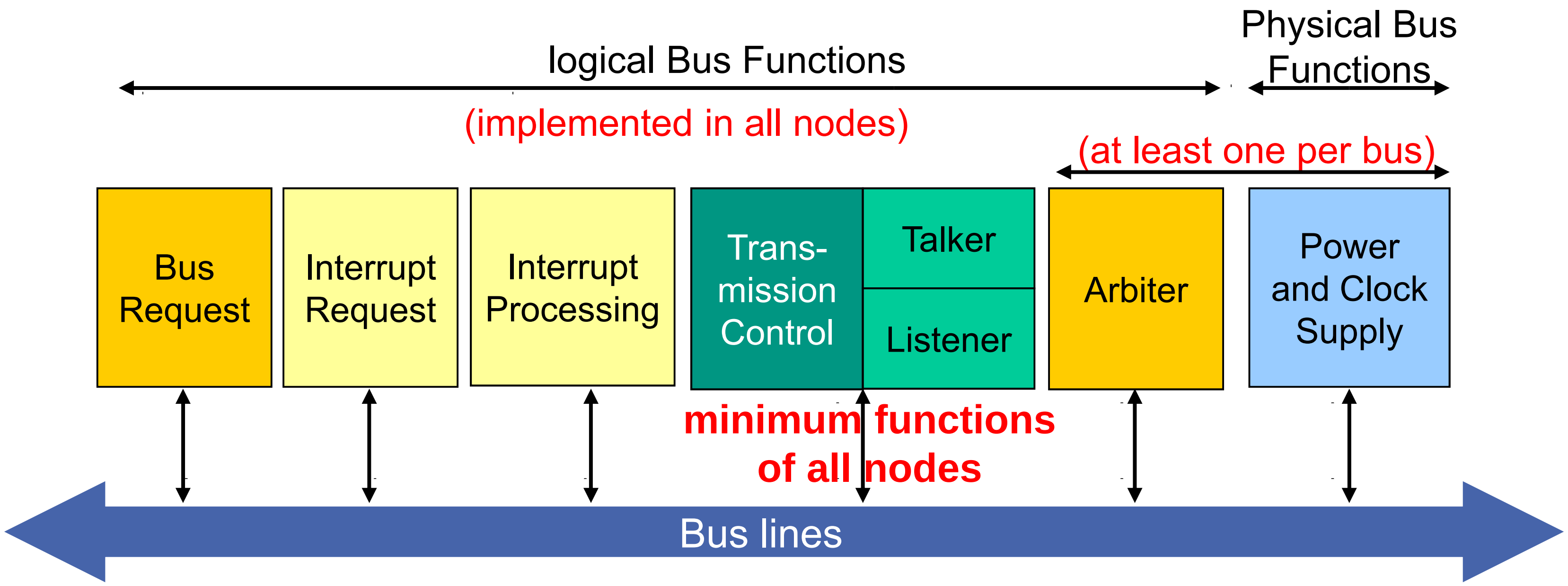


General Bus Structure (3)

- 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)

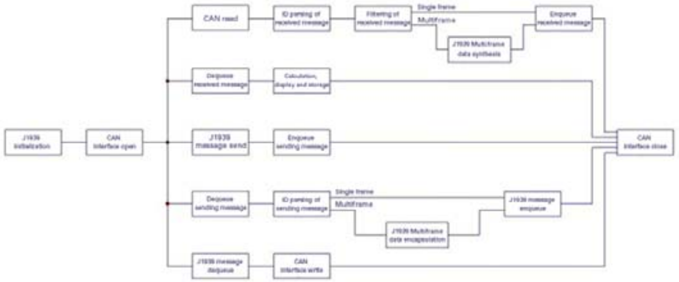
Specification of a Bus System

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

Source: http://www.etas.com

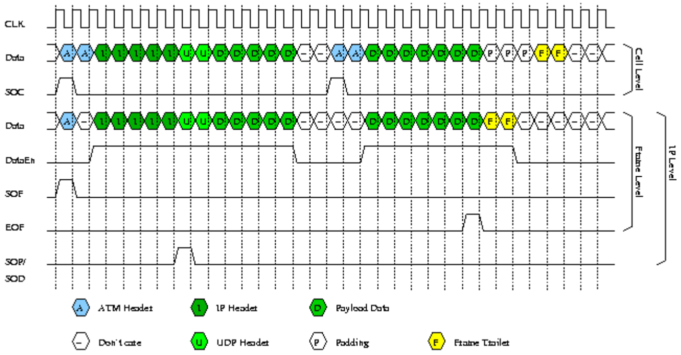
Source: http://www. leib-engineering.de

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



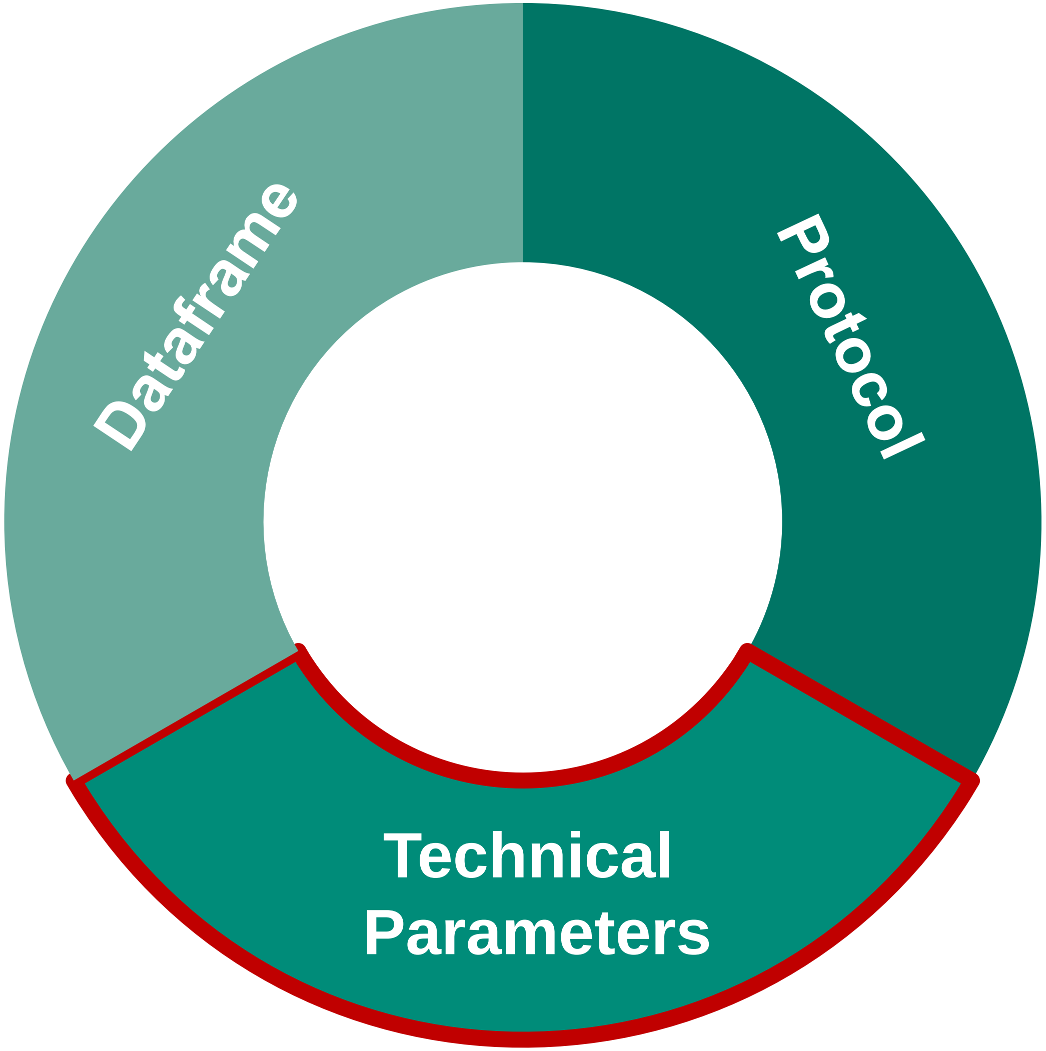
Source: http://sine.ni.com

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

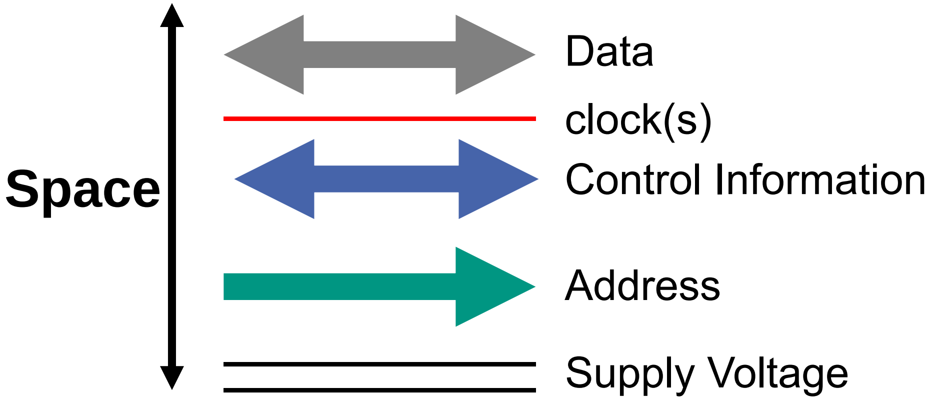


□ All of the above is forming the specification

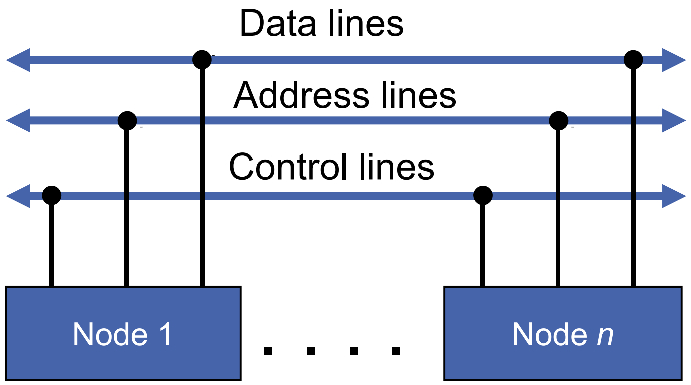
Parts of a Specification



Parallel Bus



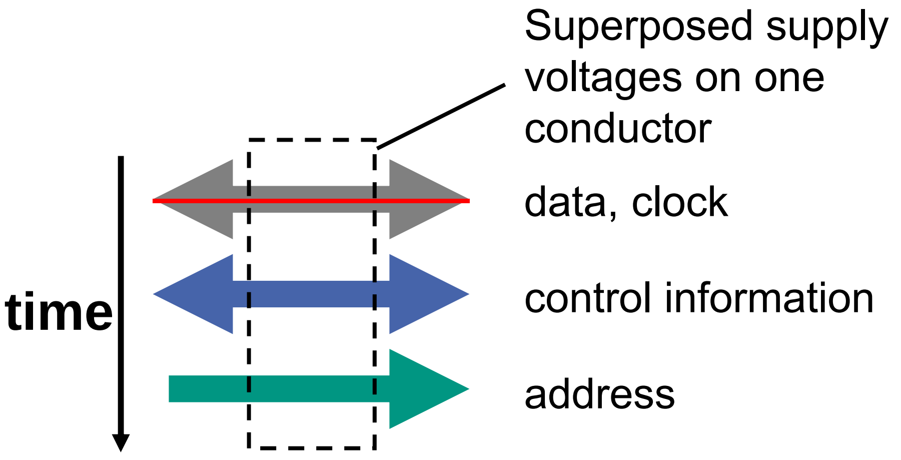
logical Setup



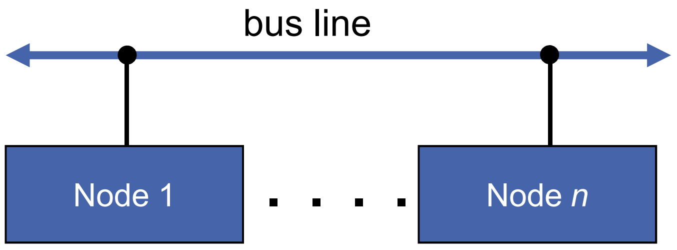
physical Setup

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



logical Setup



physical Setup

- 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

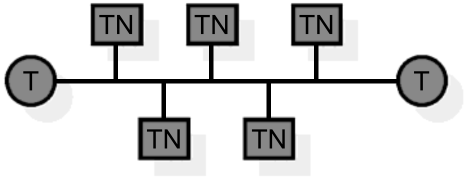
Topology

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.*

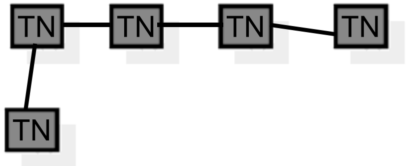
- Characteristics of divers topologies
  - Normal operation: resources and cost, throughput, range, maximum number of nodes, expandability
  - Exceptional operation: susceptibility, start-up operation, restart, node connection during runtime, Security (malicious misuse, eavesdropping,…)

Common types of Topologies - Overview

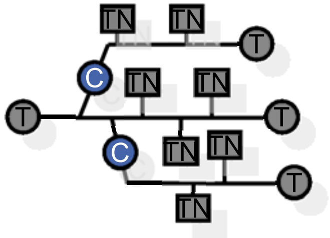
- **Bus**



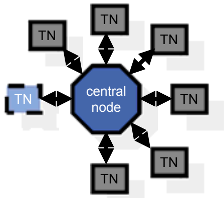
- **Line**



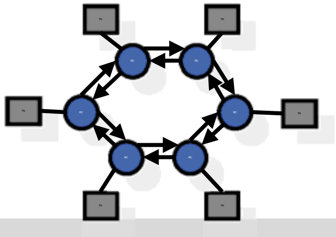
- **Tree**



- **Star**



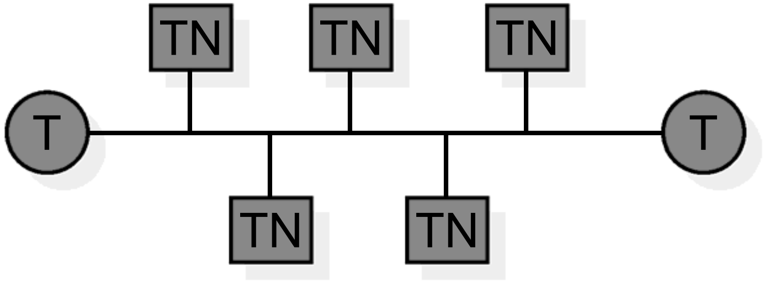
- **Ring**



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.

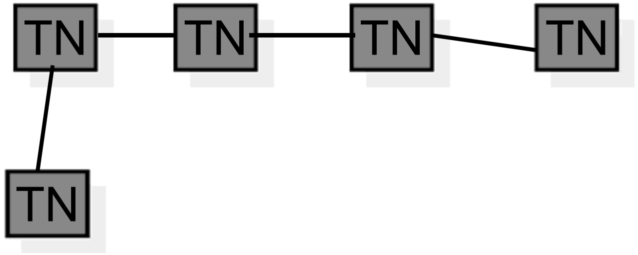
- Linear Bus
- One shared line
- Easy connection of nodes with tap lines
- Only one master a time
- Danger of collisions, complex management



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.

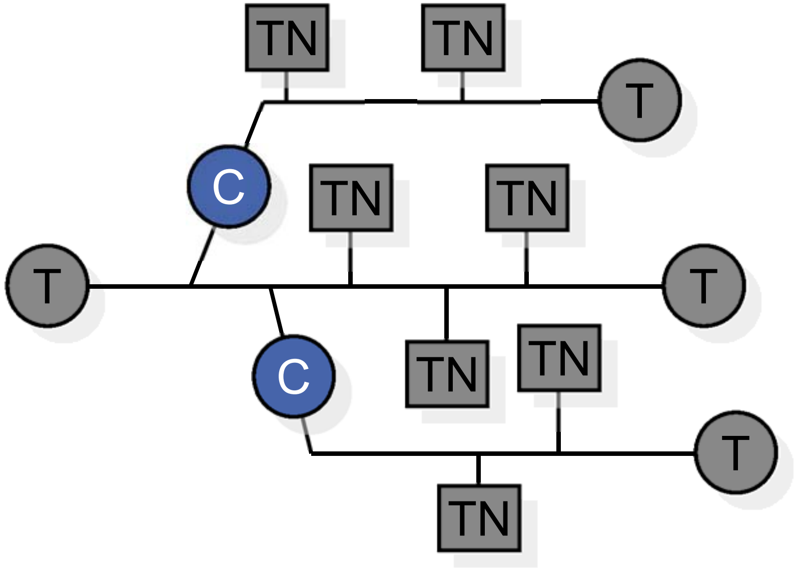
- Point-to-Point Connection
- Easy management, easily extensible
- Example: FireWire in some cases



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.

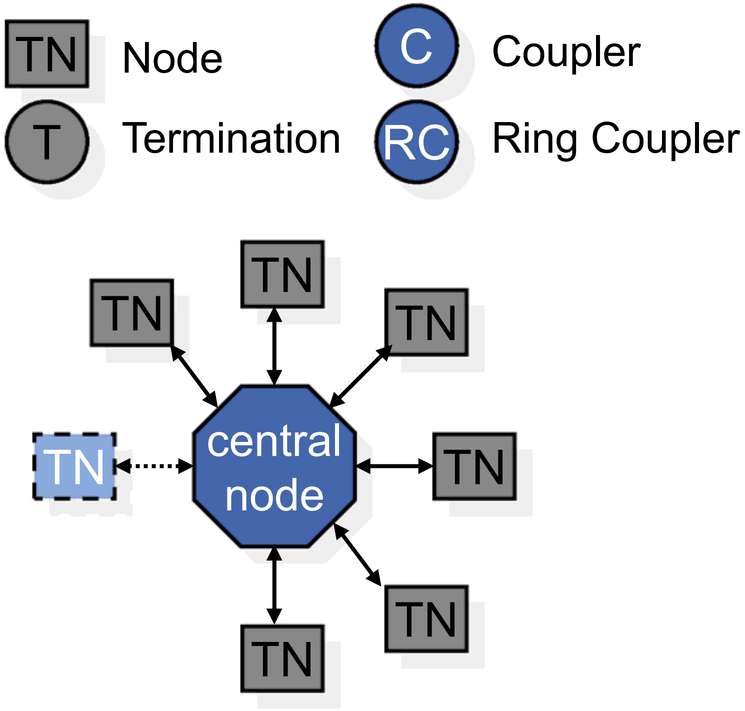
- Tree Structure
- Suitable couplers connect individual linear bus segments
- Long distances possible
- Increased data throughput with local data exchange and intelligent couplers



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.

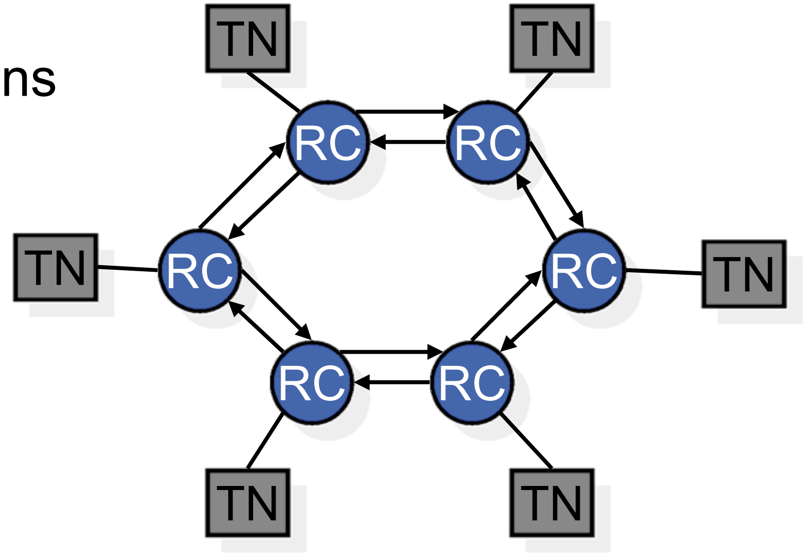
- Central Node, no communication when inactive
- Point-to-Point Connection
- Easy management, easily extensible
- Example: Telephone switchboard
  Ethernet-Switch



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

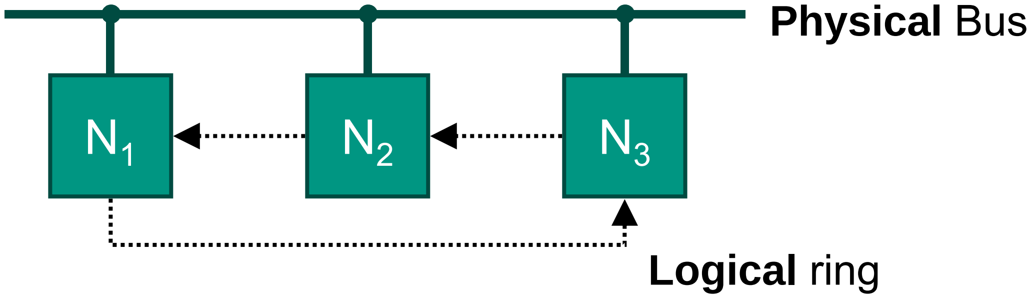


Logic Structure vs. Physical Structure

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

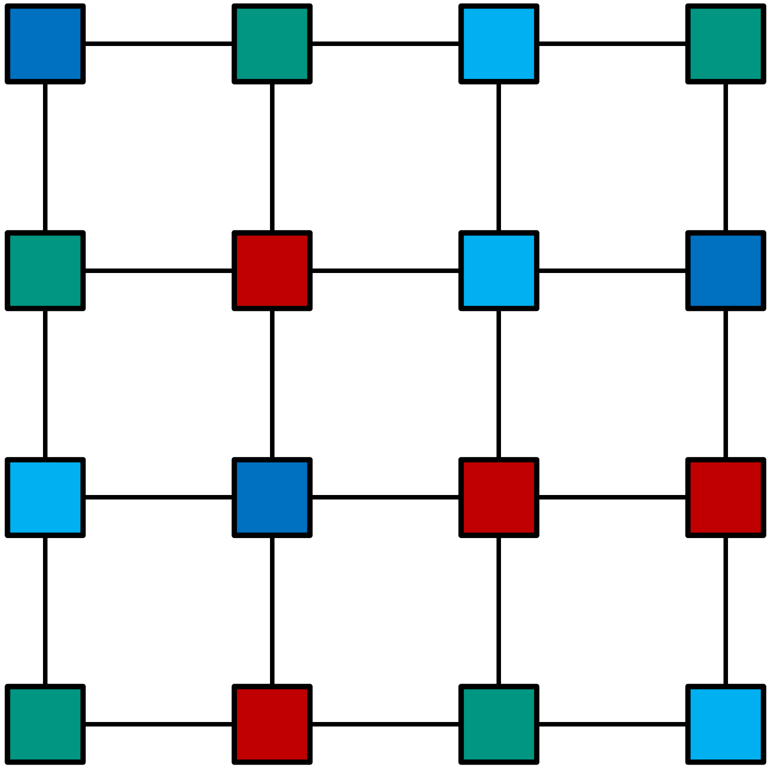
- Difference between logic and physical structure are possible
  - Physical bus with a logic ring e.g. Profibus, TokenBus
  - Physical star with logic bus e.g. Ethernet

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

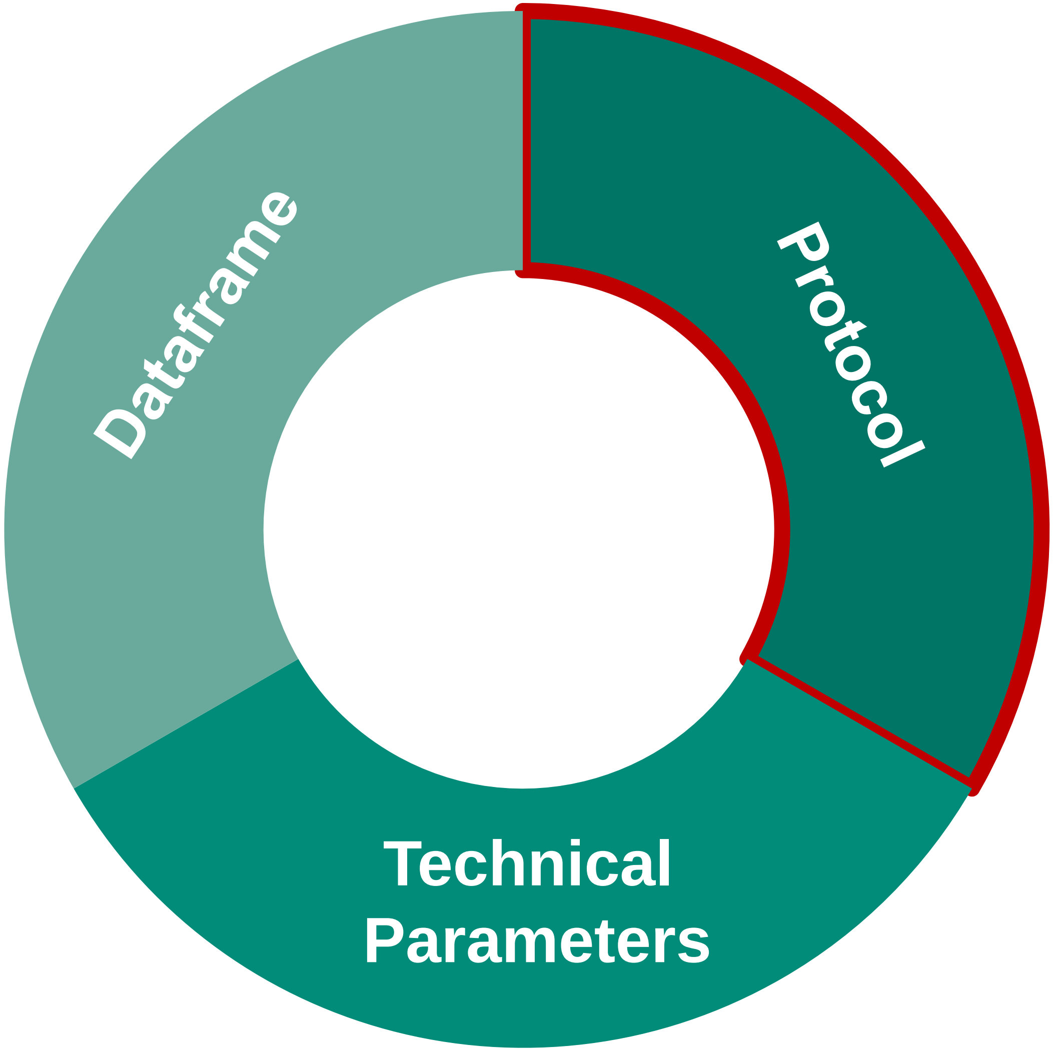


Network

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



Parts of a Specification



Think-Pair-Share: Requirements for Protocols

- There is an existing communication system
- All technical parameters (wiring, voltages, connectors, etc.) are already defined
- You have to describe the behavior of the communication partners, so that someone else is able to implement it

- What are the general properties your specification has to fulfill?

1. Discuss with your neighbor
2. Share your ideas with the audience

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

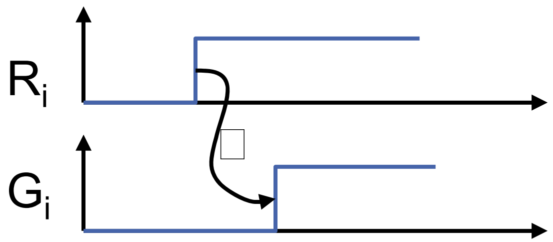
- Verbal representation
  „.. TN_i would like to be master and asserts signal R_i with 1. Subsequently the arbiter grants access to the bus G_i if priority of i is currently the highest...“

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

- □ Representation in a formulized procedure is useful

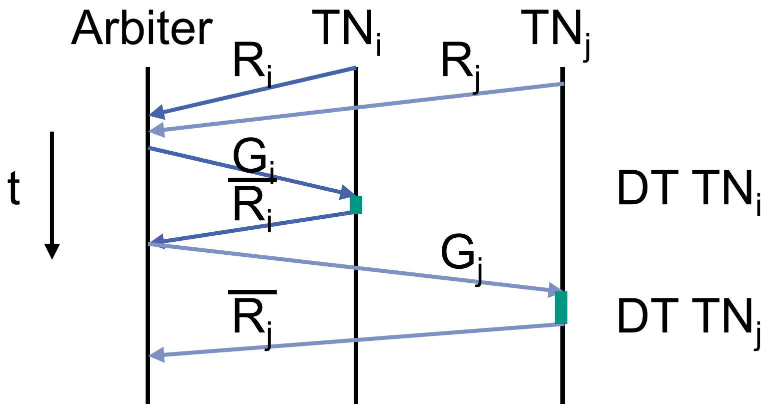
Protocol Specification

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



□ If P_i maximal

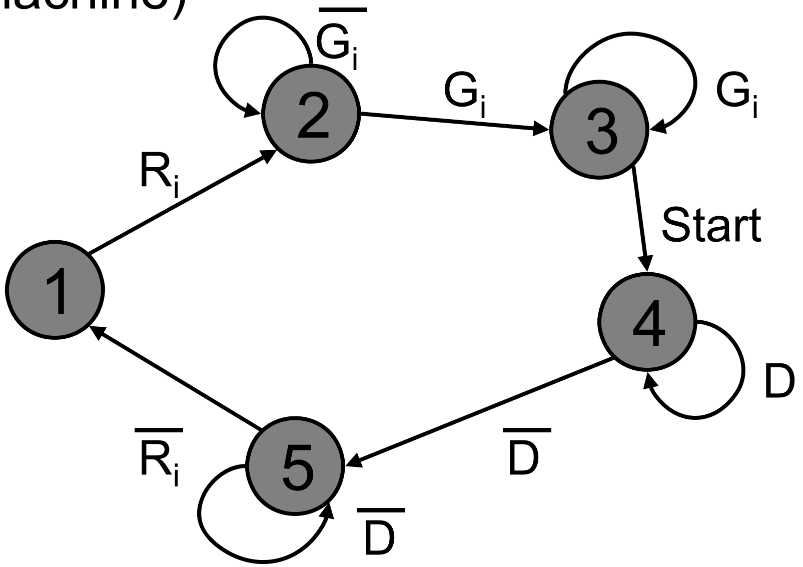
- Sequence diagram
  - Communication partners are marked with vertical lines
  - Exchange of messages are marked with descending lines



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
```

Outlook

- Datagram
- Error protection
- Classification