Bus Systems

Automotive Bus Systems – LIN, MOST, Flexray

Prof. Dr. Reinhard Gotzhein, Dr. Thomas Kuhn

Automotive Bus Systems

Learning objectives

- Understand more automotive bus systems
  - LIN Bus
  - MOST Bus
  - Flexray Bus

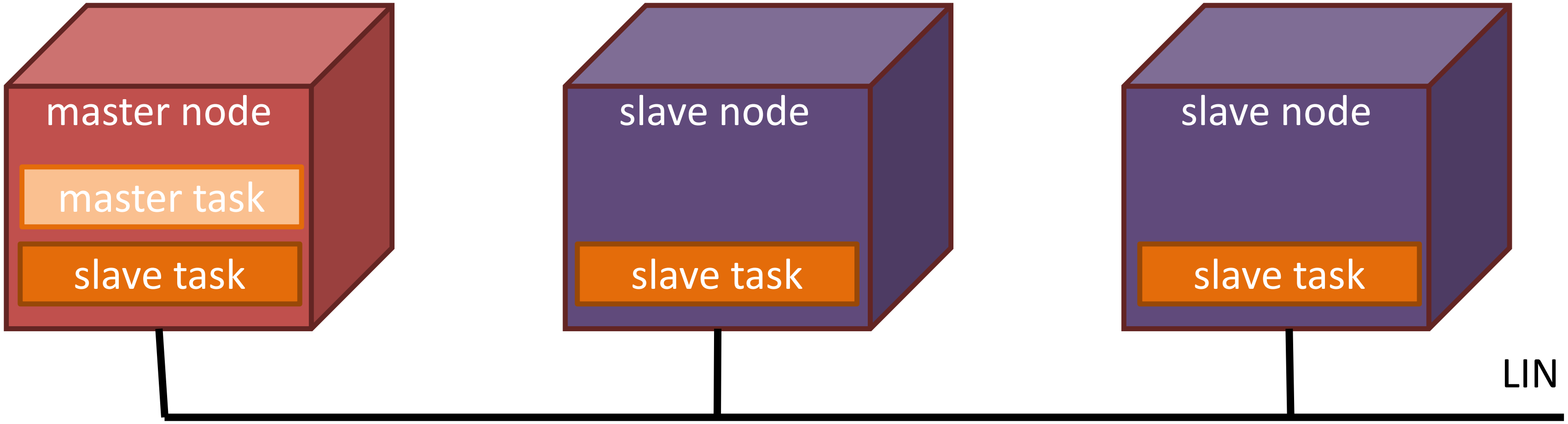
- Know their operation priciples, predictablility, and application areas

Local Interconnect Network (LIN)

LOCAL INTERCONNECT NETWORK

- Development and standardization
  - Since end of 90‘s by industry consortium
  - de-facto-Standard; most recent: LIN 2.1 (2006); USA: SAE J2602 (2005)
- Application areas
  - Cheap communication bus for coupling simple automotive devices
  - Doors, seats, mirrors, windows, light…
  - Nested bus, used to connect devices to a single CAN bus master
- Properties
  - One wire bus, NRZ coding
    - 12V: logic 1 (recessive)
    - 0V: logic 0 (dominant)
  - Bytewise transmission using start and stop bits
  - Transmission rate up to 19,2 Kbps; useable data rate up to 800 Byte/s
  - Bus length up to 40m

Local Interconnect Network (LIN)



- Only one wire between nodes
  - One reason for low transmission speeds – why?

- Medium access is controlled by central master node
  - Single-Master/Multiple-Slave
  - Nodes may implement master and slave software tasks
  - Communication is controlled by master task
  - Master node realizes gateway to other networks (e.g. CAN)

Local Interconnect Network (LIN)

- LIN Bus - Scalability
  - 1 Master node, up to 16 slave nodes
  - Periodic polling of slave nodes by master

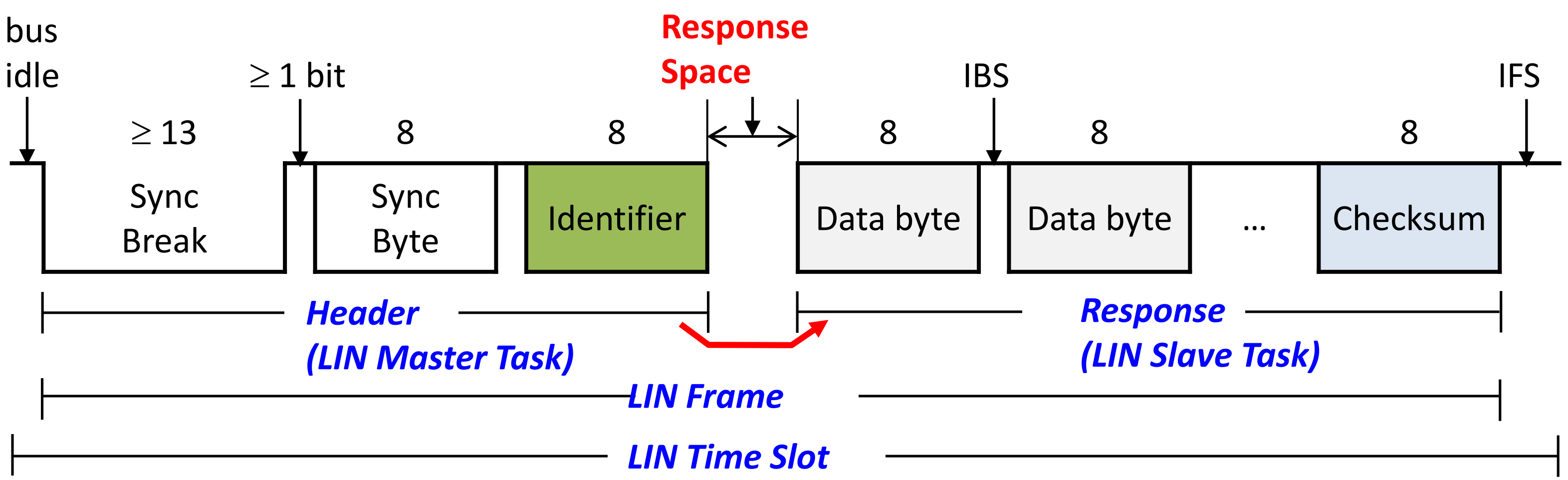
- Addressing
  - Each message is itentified by unique identifier that is assiged to one fixed node (master or slave)
  - A sender may have several IDs assigned
  - LIN nodes decide based on received message ID whether to receive the message or not
  - LIN does not use node addressing (this is similar to CAN bus)

Local Interconnect Network (LIN)

Regular LIN transmission

- Master node divides communication medium into time slots of fixed, predefined length (Similar to TT-CAN)
- LIN master node polls slave nodes by transmitting a request frame with defined frame ID
    - Polling frame is transmitted in predefined transmission slot
- LIN slave node that is responsible for ransmitted frame ID responds in same transmission slot
- One or multiple receivers receive the transmitted frame
    - Master and/or slave nodes
    - Is this communication reliable?
    - Is this communication predictable?
    - Does LIN bus scale?

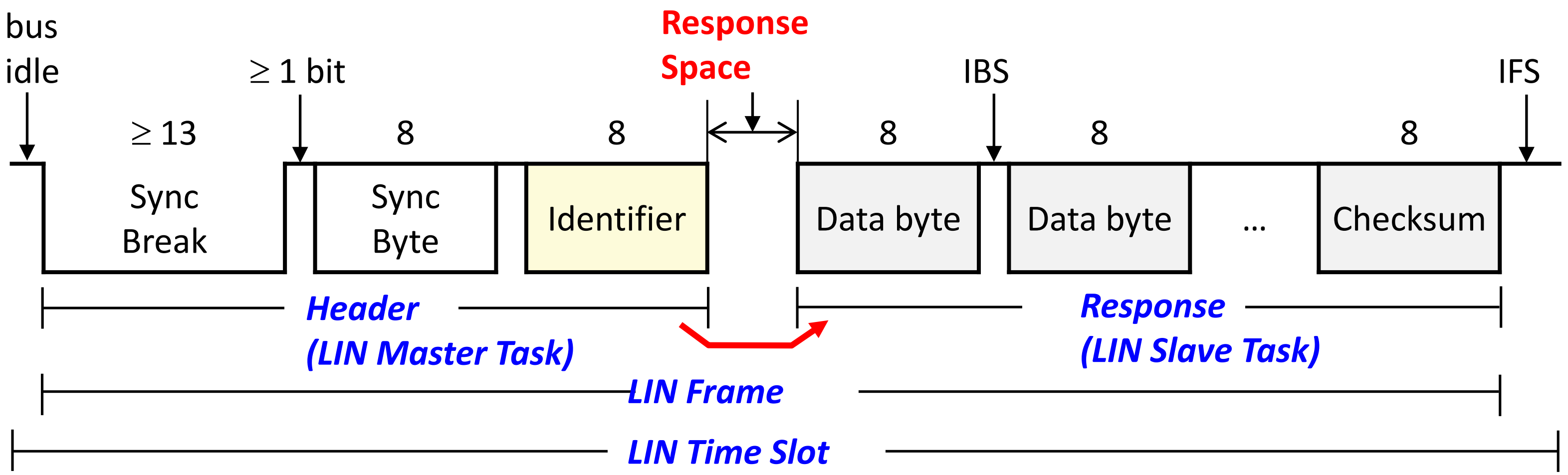
Local Interconnect Network (LIN)



IBS: Inter-Byte Space; IFS: Inter-Frame Space

- **Sync Break:** Sequence of at least 13 dominant Bits for signalling bus arbitration by Master node, followed by at least one recessive bit. This unique bit sequence is recognized by all slaves , all nodes are informed about frame start
- **Sync Byte:** Bit sequence 01010101 for synchronizing bit clock rate of slaves
- **Data bytes:** Message length is between 1 and 8 bytes and is statically configured for each message type
- **Checksum:** Calculated from data bytes (classic checksum) or identifier and data bytes (enhanced checksum)

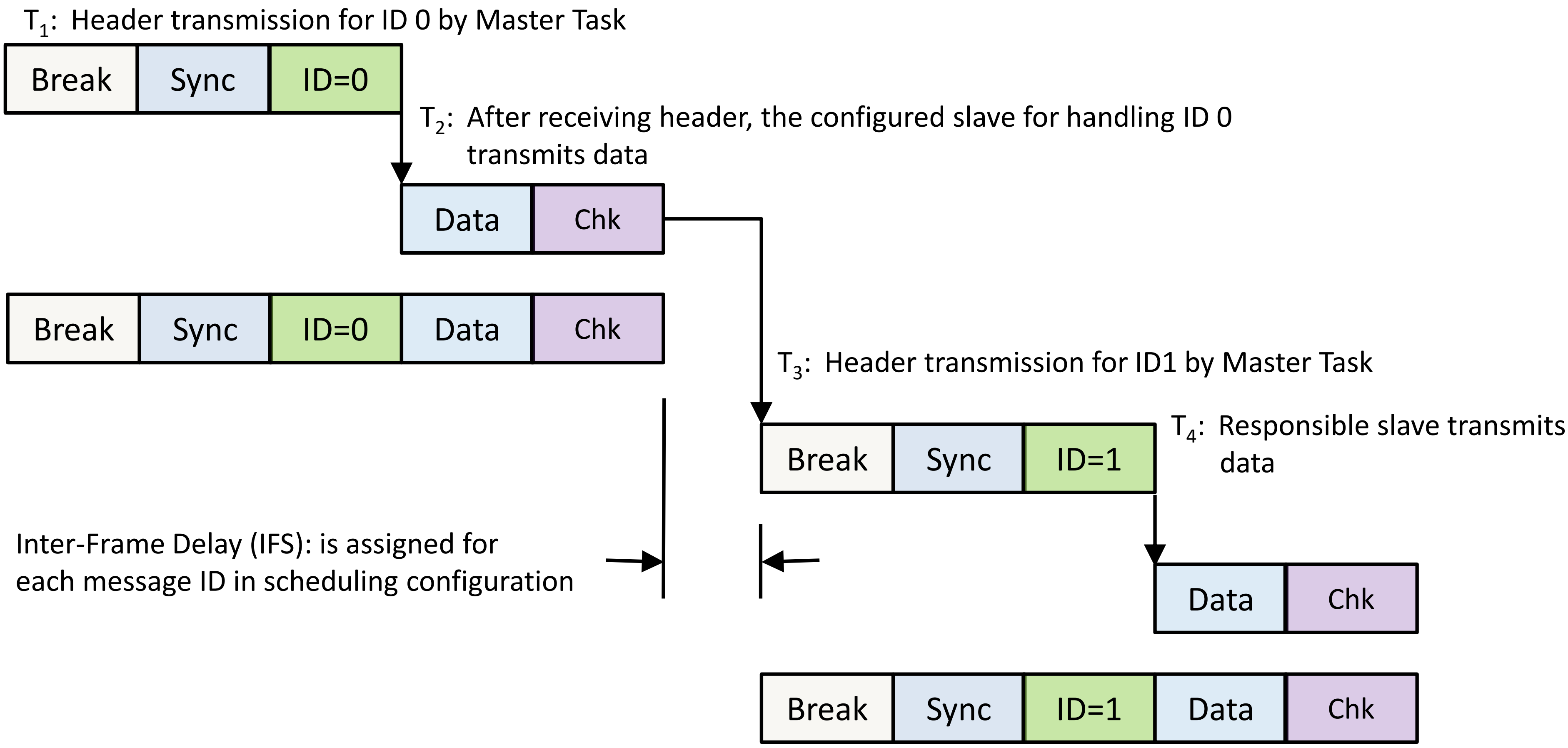
Local Interconnect Network (LIN)



IBS: Inter-Byte Space; IFS: Inter-Frame Space

- **Identifier:** 6 Bit ID of transmitted frame type, followed by 2 parity bits
  - 0x00 .. 0x3B: freely useable by developer
  - 0x3C, 0x3D: Diagnostic frames for maintenane purposes
    - Master-Request-Frame (MRF): Diagnostincs and configuration
    - Slave-Response-Frame (SRF): Querying of slave data
  - 0x3E, 0x3F: reserved for extensions

Local Interconnect Network (LIN)



Local Interconnect Network (LIN)

- Master node splits medium into fixed time slots
  - d_{IFS} (d_{RS}): Inter-Frame (Response) Space duration
  - Factor 1,4: Permitted synchronization tolerance
  - Assumptions
    - R_T = 19,2 Kbps → d_{bit} = 0,052ms
    - $n_{dataBytesMax}$ = 8

- Polling according to statically configured scheduling plan by master node
  - Static definition of message order and message frequency
    - Exclusive medium access at defined times
    - Deterministic upper bounds for transmission delays (assuming no errors)
  - Master node may switch between scheduling tables depending on operation state (e.g. normal operation or diagnostics)

Local Interconnect Network (LIN)

LIN Frame types

- Standard frames / unconditional frames
  - Exclusive assignment of time slots
  - Polling by master
  - Collission free transmissions
- Sporadic frames
  - Assignment of time slot to multiple identifiers
  - Definition of identifier priority order
  - Master transmites identifier with (slot local) highest priority
    - Master decides which frame type will be transmitted
  - Collission free transmissions

Local Interconnect Network (LIN)

- Event triggered frames
  - Use of same identifier for polling multiple frame types
  - Slaves respond when an event did occur
    - Decentral slot usage decision

  → Collissions are possible

    → Transmission delays are no longer deterministic

  - Collission avoidance strategies
    - Use events that are mutually exclusive
    - Use bit monitoring to detect collissions
    - Use checksum to detect collissions

  - Collission handling
    - Stop transmsission
    - Use unconditional frames to poll nodes individually

Local Interconnect Network (LIN)

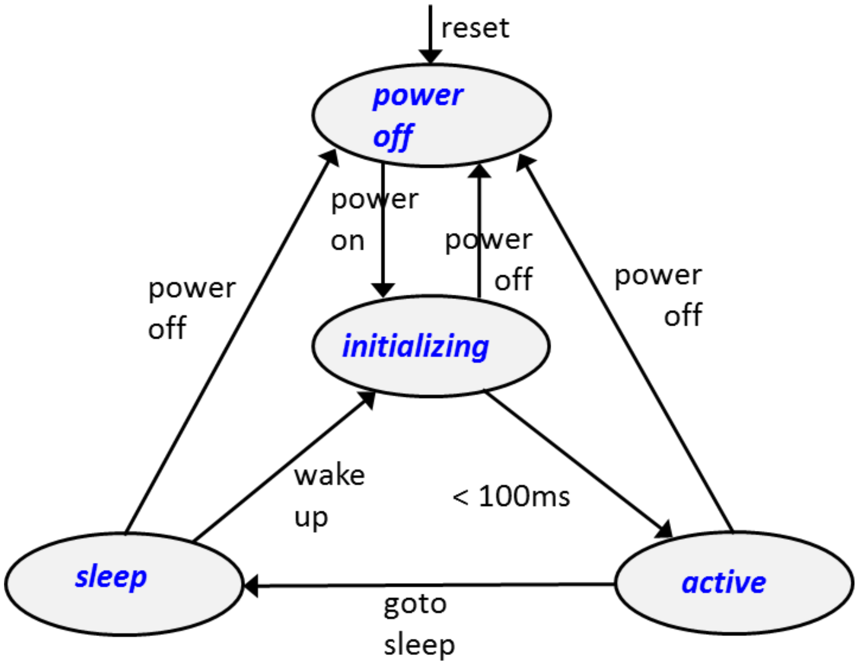
Error detection and error handling

- Bit errors
  - Detection: Monitor transmitted bit value with bit value on bus by sampling voltage level
  - Handling: Stop transmissions (at end of byte)
- Checksum error
  - Detection: Receiver validates transmitted checksum with calculated checksum
  - Handling: Discarding of received frame, no error signaling
- Wrong identifier
  - Detection: Parity check of identifier byte by receiver(s)
  - Handling: Ignore frame, no error signaling
- Slave-Not-Responding
  - Detection: No frame data is transmitted after master did transmit header
  - Handling: Not part of LIN bus specification

Local Interconnect Network (LIN)

Energy management

- State graph of transceiver
  - *power off:* Switched off
  - *active:* Normal operation
  - *sleep:* Energy saving mode



- State transitions
  - Goto sleep
    - Explicit signaling by master
    - Autonomously due to bus inactivity (after 4 sec at earliest, after 10 sec at latest)
  - Wake up
    - Signaling by master or slave
    - Dominant bus level for 250 – 500 µsec

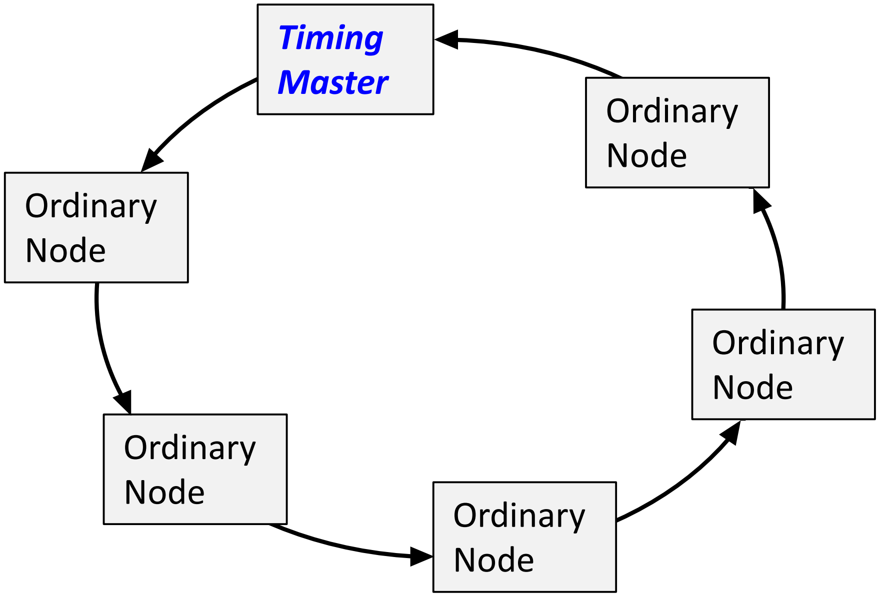
Media-Oriented Systems Transport (MOST)

- Development and standardization
  - Since end of 90s by industry consortium: MOST Cooperation (amongst others: Audi, BMW, Daimler, Harman/Becker, SMSC)
  - MOST Specification Rev. 3.0 (2008)
  - Predecessor: Domestic Data Bus (D2B), Philips

- Application areas
  - Telematics- and multimedia applications (Infotainment-Bus)
  - Car radio, CD changer, Cellphone, TV, Navigation system

- Properties
  - Fibre-Wire or twisted pair wires
  - Transmission rates: 25 Mbps (MOST25), 50 Mbps (MOST50), 150 Mpbs (MOST150) [since 2012: A3, Mercedes S-Class]
  - Ring topology, up to 64 nodes

Media-Oriented Systems Transport (MOST)



- Physical ring topology
  - Unidirectional PtP connections
  - Collission free

- MAC with distributed medium access
  - Reservations for transmission of streaming data
  - Token Passing for sporadic (Packet) data
  - Priority based fair contention for management data (control data)

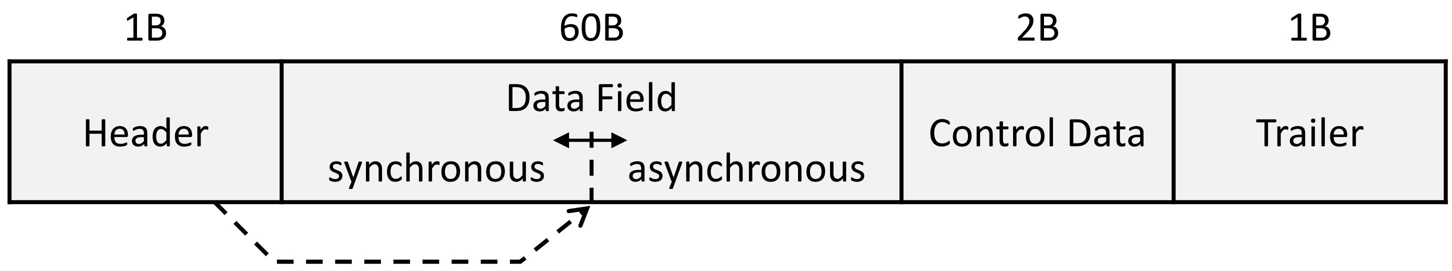
Media-Oriented Systems Transport (MOST)

- Time divided medium
  - Time slots of fixed duration, called MOST blocks
  - Each MOST block is subdivided into 16 MOST frame slots

- Duration of frames (and blocks) depends on frame rate
  - 44,1 kHz (Sampling rate of Audio-CD)
    Frame duration = 22,67 µsec, Block duration = 363 µsec
  - 48 kHz (Sampling rate of Audio-DVDs):
    Frame duration = 20,83 µsec, Block duration = 333 µsec

- All nodes in a MOST ring must use same frame rate

Media-Oriented Systems Transport (MOST)



MOST25 Frame format (Frame size = 64 bytes)

- Header
  - Preamble (4b): Synchronization (SoF, SoB)
  - Boundary Descriptor (4b): Subdivision into synchronous and asynchronous area
    - (Value x 4)
  - Data Field
    - Synchronous (24-60 Byte): Contention free transmission based on reservations
    - Asynchronous (0-36 Byte): Contention based transmission
  - Control Data: Management data
  - Trailer: Status information, checksum

Media-Oriented Systems Transport (MOST)

Synchronous data (24 - 60 Bytes per Frame)

- Split into 24 - 60 time slots, 8 Bits (1 Byte) per time slot
  - One time slot is one virtual (physical) channel
  - Channels are reserved individually by applications

- Synchronous part of MOST frame is usually not completely assigned to one application
  - Transmission rate of one virtual (physical) channel:
    - 44.100 Bytes/s if 44.100 MOST frames are transmitted per second
    - 48.000 Bytes/s if 48.000 MOST frames are transmitted per second

Media-Oriented Systems Transport (MOST)

Transmission of streaming data

- Logic channels (Streaming Channels) created by grouping virtual physical channels
- Administration of streaming channels via explicit control messages
  - Reservation
  - Definition of sender and receiver
- Example
  - Transmission of Audio CD (Stereo, 16 Bit per sample) requires 4 physical channels
  - Maximum of 15 concurrent uncompressed Audio CD streams (MOST25)

Media-Oriented Systems Transport (MOST)

Transmission of streaming data

- Supports streams up to 2,646 MB/s
- Ring topology - delay for message forwarding:
  - 2 frame transmission times per node
  - $2 \cdot 22{,}67$ µsec, 45 µsec if 44,1 kHz frame rate is used (MOST25)
  - MOST50 guarantees < 1 µsec

Media-Oriented Systems Transport (MOST)

Transmission of sporadic data

- Using asynchronous data area ($0 \leq n \leq 36$ Bytes per Frame)
  - Asynchronous part of MOST frame is completely assigned to one sender
  - Transmission rate of one physical channel:
    - $0 \leq n \leq 36$ Bytes per MOST-Frame
    - $n \cdot 44.100$ Byte/sec (Assuming system frequency of 44.1 kHz)

- Longer frames may be split to multiple MOST frames (fragmentation)
  - Example: Map data for navigation systems require short data bursts
  - Token passing mechanism is used for medium arbitration

Media-Oriented Systems Transport (MOST)

1B	2B	1B	2B	0..48B (0..1014B)	4B
Arbitration Field	Target Address	Data Length	Source Address	Data	CRC Checksum

MOST25 Data frame format

- Arbitration field: Detect busy medium
  - Target/Source Address
    - Physical address is calcutaed based on node position in MOST ring,Master address is 0x0400
    - Logic address: assigned by master
  - Data Length: Length in quadlets (4 Byte-Units)
  - Data: Payload, up to 48 Bytes (or 1014 Bytes) length
  - CRC Checksum: Checksum

Media-Oriented Systems Transport (MOST)

Transmission of control data (management data)

- Using of control channel in MOST frame
  - 2 Bytes per Frame
  - Transmission rate: 2 · 44.100 Byte/sec (assuming frame rate of 44.1 kHz)
- Management data frames are split accross **16** MOST-Frames of MOST-Blocks
- Priority based, fair contention for communication channel
  - 0 (low) .. 15 (high)
  - Node checks arbitration field and overwrites if necessary
  - Initial sender removes control data from ring when it passes again

Media-Oriented Systems Transport (MOST)

4B	2B	2B	1B	17B	2B	4B
Arbitration Field	Target Address	Source Address	Message Type	Data	CRC Checksum	Trailer

- Format of MOST25 Management frames (Frame length = 32 Byte)
  - Arbitration Field: Priority of control message
  - Target/Source Address: see above
  - Message Type:
    - Ressource Allocation / Deallocation: Request/release logic channels
    - Remote Read / Write: read/write configuration of other nodes
  - Data, CRC Checksum: see above
  - Trailer: e.g. Acknowledgement from receiver

Media-Oriented Systems Transport (MOST)

Error detection and handling

- Streaming Data
  - No error detection and handling on MAC level
  - Repetition of falsely received data usually not necessary
- Sporadic data
  - Checksum error
    - Detection: CRC-Checksum
    - Handling: Dropping of frame, no error signaling

Media-Oriented Systems Transport (MOST)

Error Detection and handling

- Management data (Control Data)

  - Checksum error
    - Detection: CRC-Checksum
    - Handling: Drop frame, do not send positive acknowledgement

  - No receiver
    - Detection: No positive acknowledgement was received by sender
    - Handling: Automatic re-transmission (Low Level Retry)

FlexRay

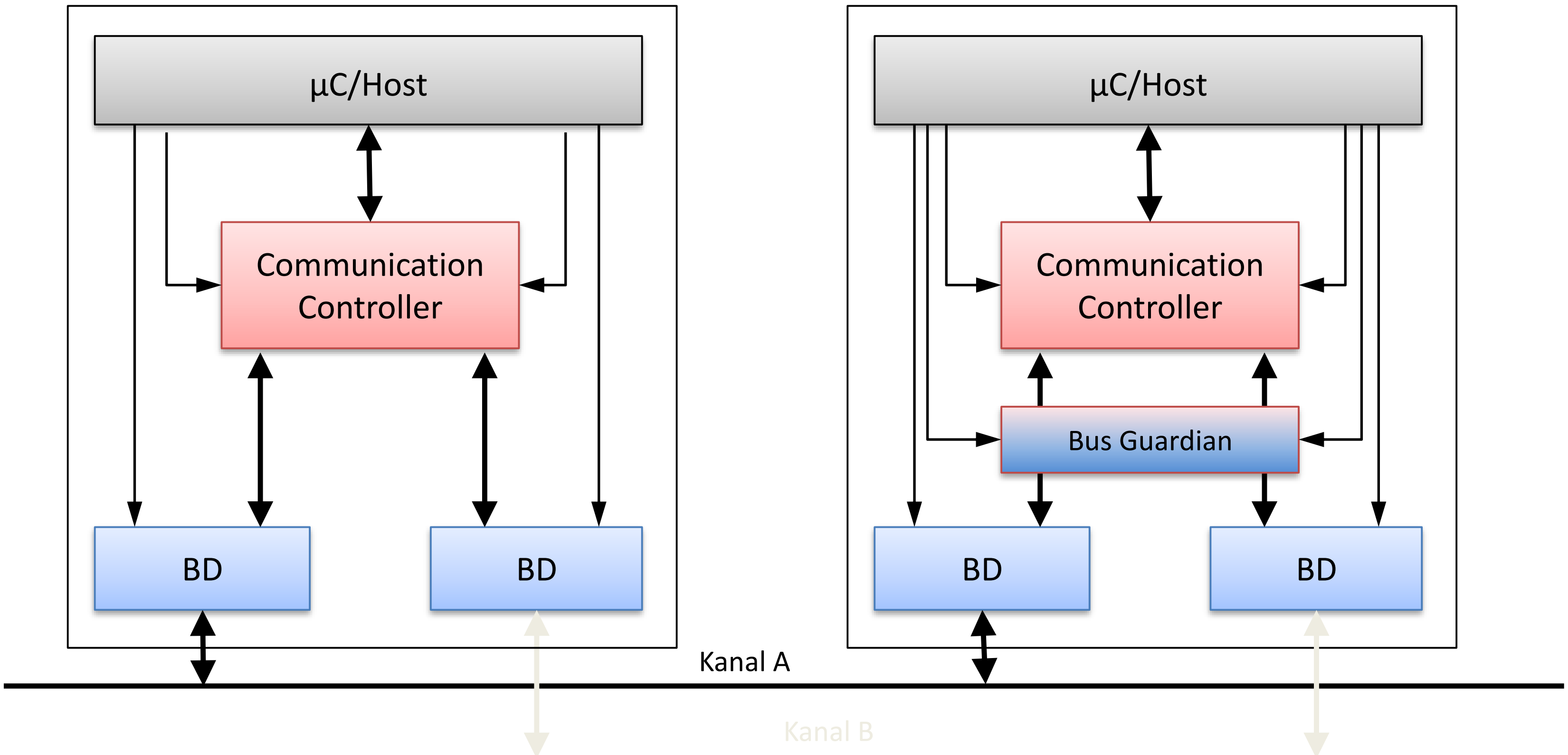
- Development and standardization
  - 2000-2009 by industry consortium (BMW, Daimler, VW, General Motors, Freescale, Philips, Bosch, …)
  - De-facto-Standard; most recent: FlexRay Specification V2.1 Rev. A (2005); Move towards ISO-Standard (IS IS 10681) in preparation
- Application
  - Similar to CAN Bus
  - Exchange of real-time control data
  - x-by-Wire-Systems

FlexRay

Properties

- Transmission rates up to bis 10 Mbps per physical channel (Twisted pair cable)
  - Bus topology (Line, Star)
  - Up to 64 nodes per segment
  - Max. 24m distance between nodes
- Robust due to 2 physiclly separated communciation channels
  - Time and event triggered communciation protocol
  - Supports both contention based and contention free access
  - Supports hard real-time guarantees
- Flexray depends on time synchronization between nodes

FlexRay


µC/Host
Communication Controller
BD
BD
µC/Host
Communication Controller
Bus Guardian
BD
BD
Kanal A
Kanal B

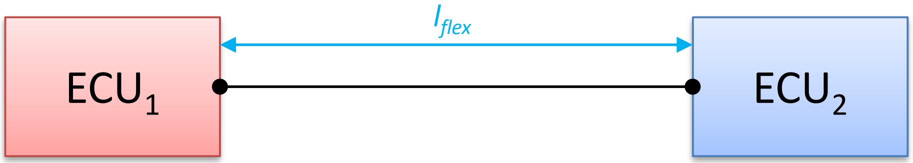
FlexRay

Topologies

- Passive bus topologies
  - Point-to-point connections
  - Linear passive bus
  - Passive star
- Active bus topologies
  - Active star

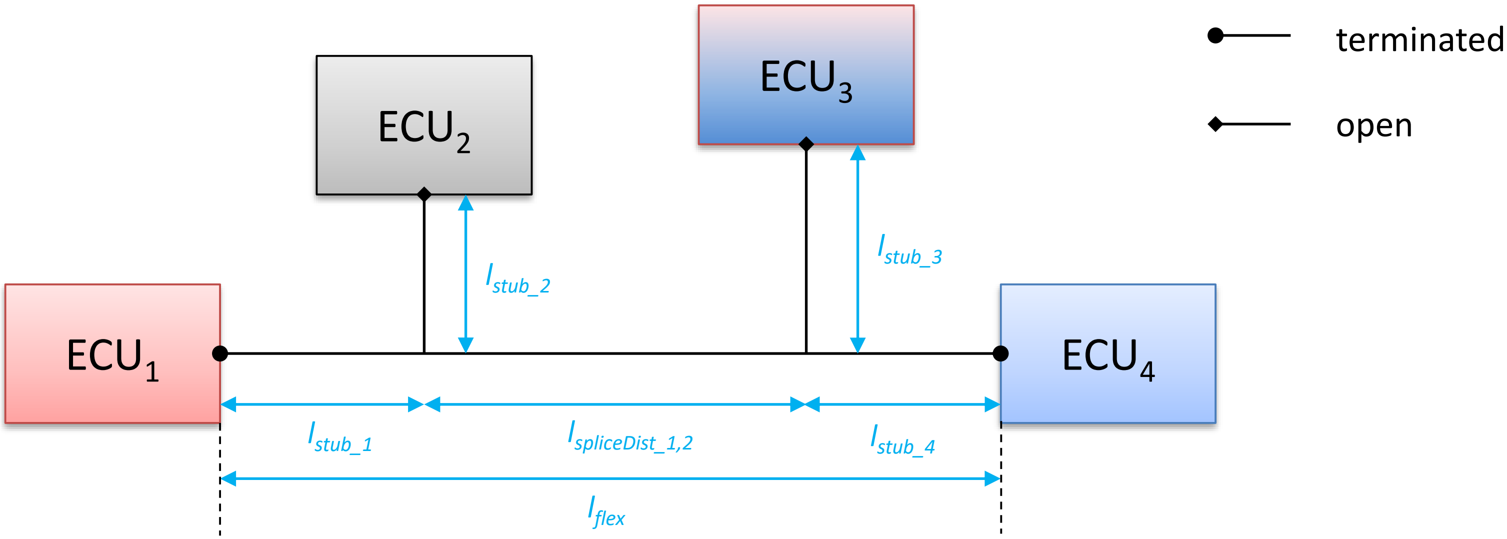
FlexRay

Point-to-point connection



- Most simplistic bus topologie
- Maximum bus length l_{flex} = 24 m
- Both ends terminated with resistor

FlexRay

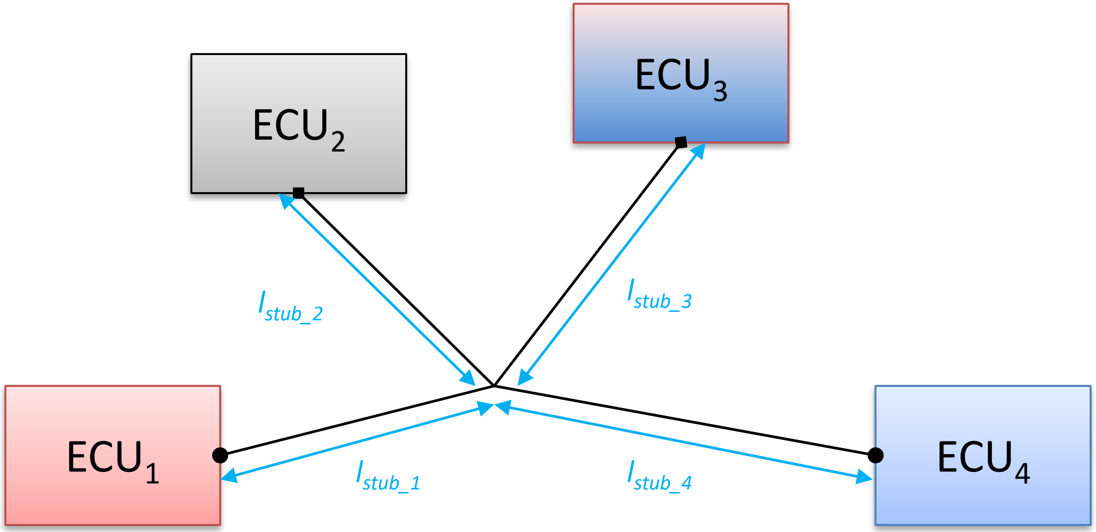


- Extension of point to point topology
  - Supports 4 to maximum of 22 nodes
  - Maximum permitted length $l_{flex} = l_{stub\_n} + l_{spliceDist\_n,m} + l_{stub\_m} = 24$ m

- Lines to main bus (l_{stub_2} *and* l_{stub_3})
  - Typical length: few cm
  - Must not contain any forks

FlexRay

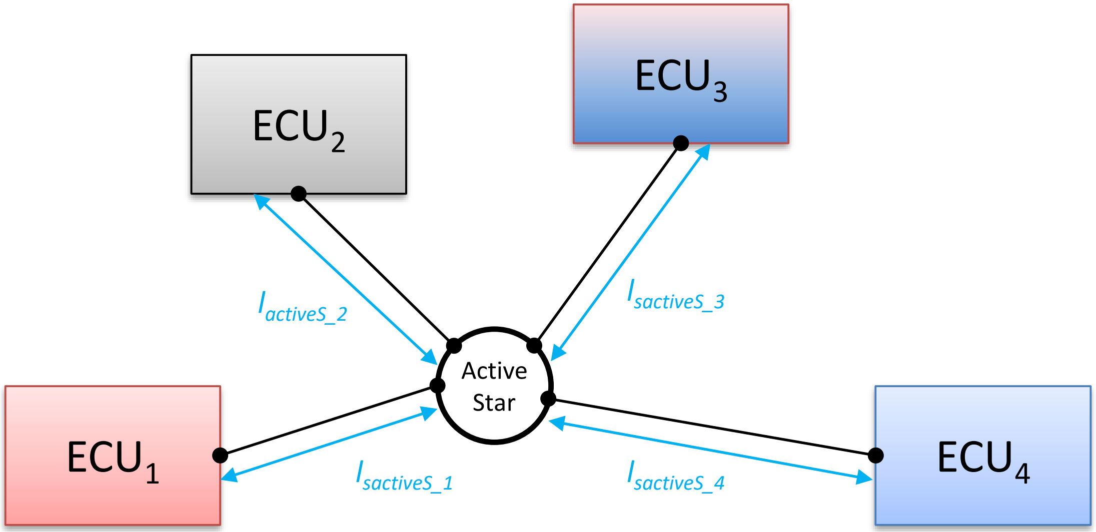
Passive star



- No „main“ line
- 3 to up to 22 nodes
- Maximum permitted bus length is $l_{flex} = l_{stub\_n} + l_{stub\_m}$ = 24 m

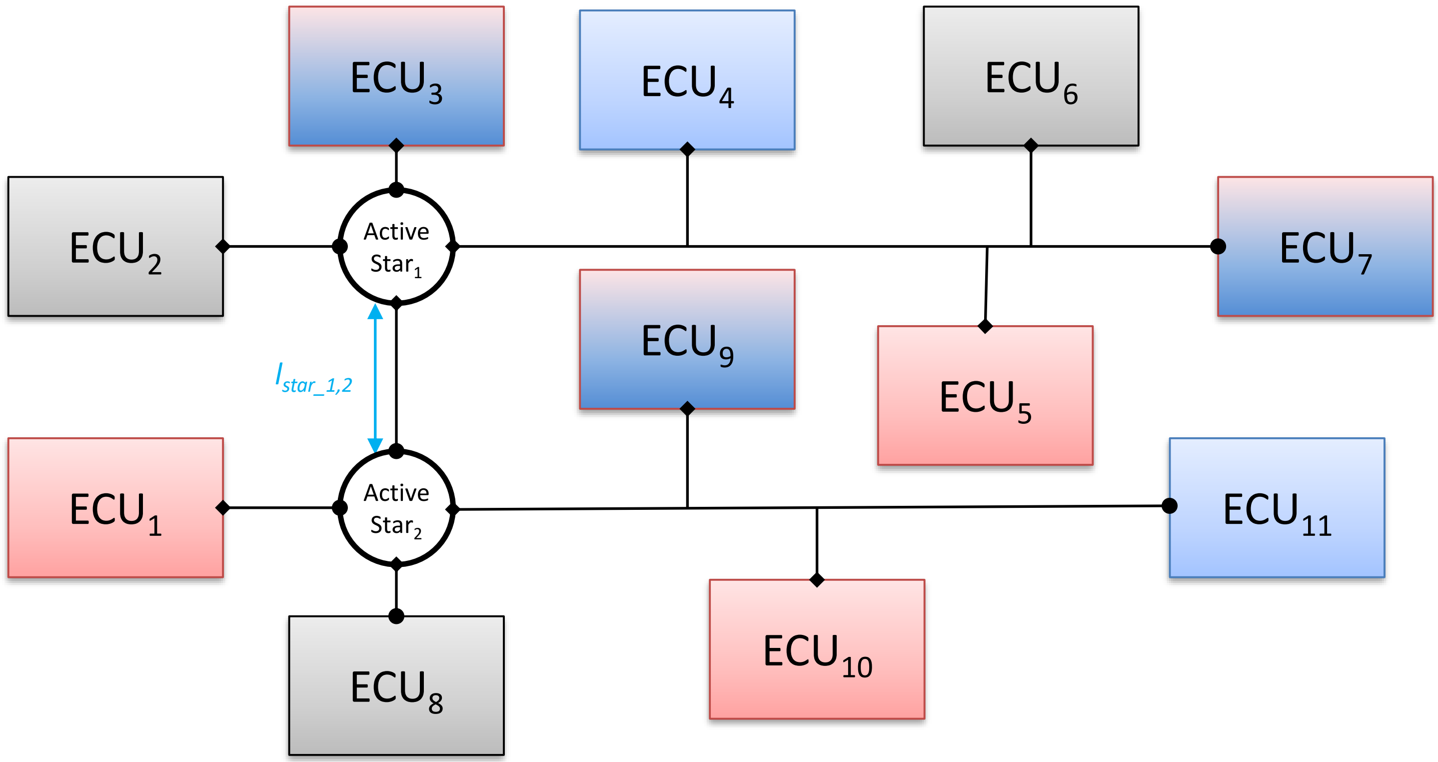
FlexRay

Active star



- No main line
  - Supports 3 to 22 nodes
  - Significantly increases maximum bus length ($l_{activeS\_n}$ = 24m)

FlexRay



- Mesh networks
  - What are maximum permitted bus lengths?

FlexRay

Toplogies

- FlexRay supports flexible topologies (compare to CAN)

- Bus speed may be traded for bus length
  - 10Mbit/s → l_{flex} = 24m
  - 5Mbit/s → l_{flex} = 48m
  - etc.

- Two independent physical communication channels
  - Doubles data rate or provides redundancy

FlexRay

Flexray MAC

- MAC with concurrent access
  - Time slotted medium
  - Requires time synchronization of all bus participants
  - Divided between contention based, contention free access, management periods

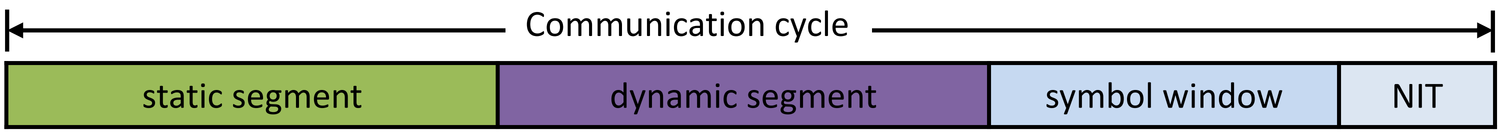
- Contention free access
- Time slots (TDMA)
- Periodic data, preconfigured schedule

- Contention based access
- Priority based access
- Flexible TDMA (FTDMA)

FlexRay

Time slotting

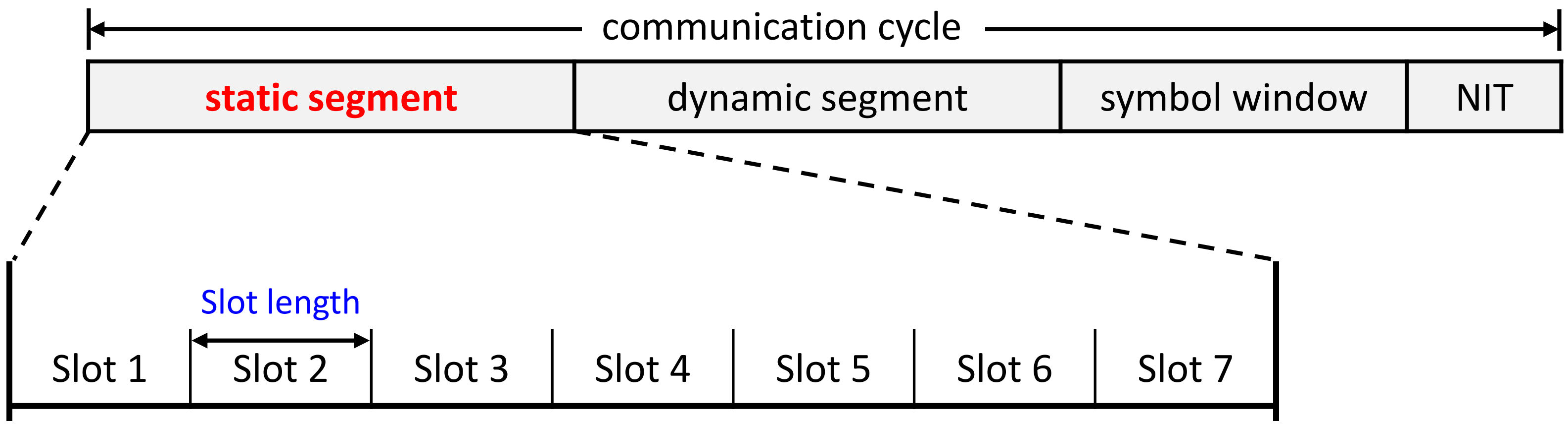
- **Communication cycles**: Dividing time into fixed size cycles (max. 16ms)
- Grouping of 64 communication cycles
- Dividing of communication cycles into fixed length segments



- **static segment**: contention free access
- **dynamic segment** (optional): contention based access
- **Symbol Window** (optional): network management
- **Network Idle Time** (**NIT**): time synchronization

FlexRay

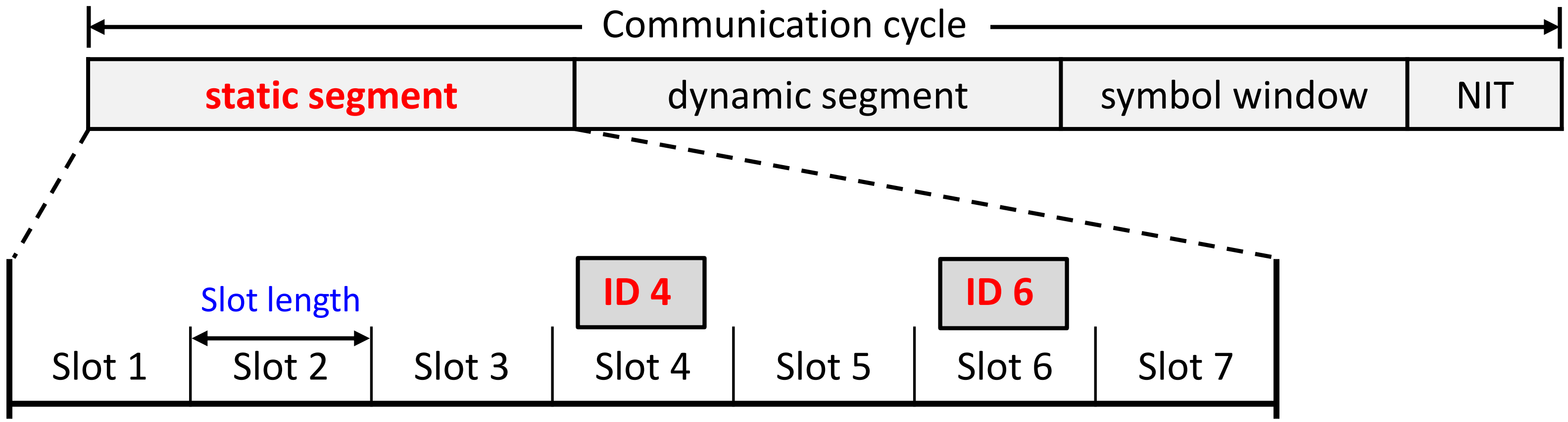
Static segment



- Static segment is subdivided into 2..1023 static slots of equal length
  - Both physical channels are divided in same way
- Static and exclusive assignment of slots to at most one node
  - Different assignments per physical channel are permitted
  - If slots of both channels are assigned to same node, redundant transmissions are possible

FlexRay

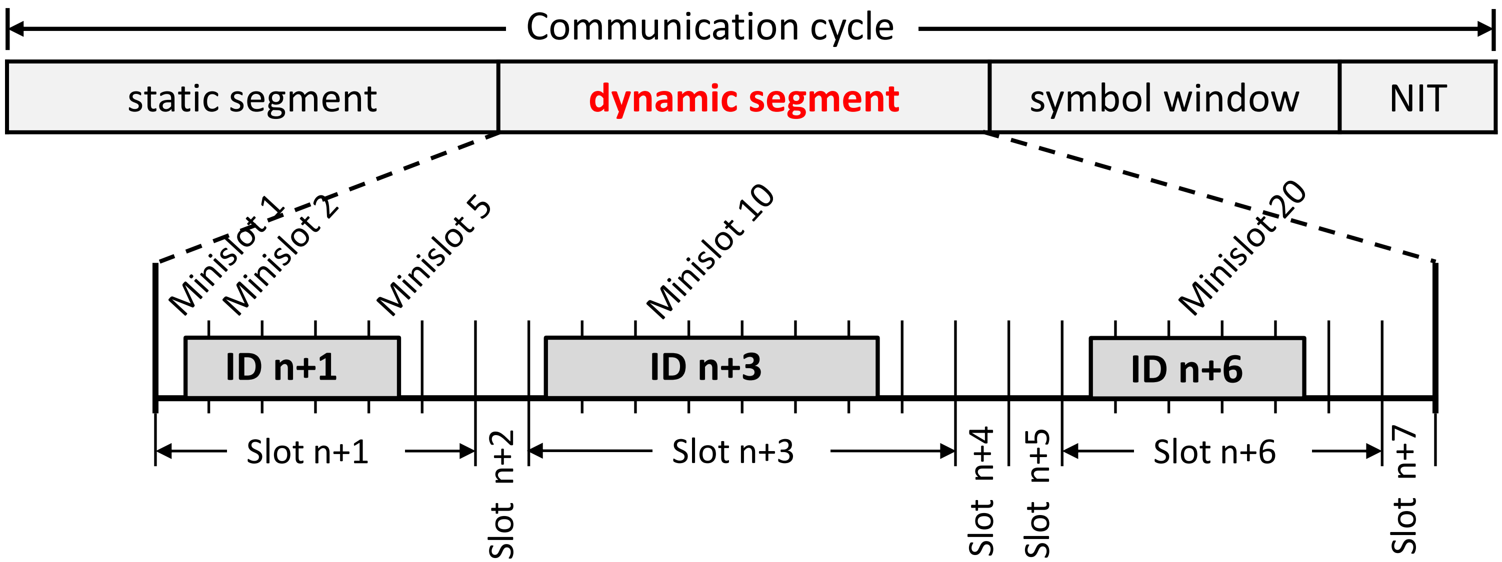
Static segment



- Identification of static slots by continuous number and channel
- Identification of frame by frame ID, channel ID, cycle counter
  - **Frame ID**: identifies slot of static segment that should be used for transmission
  - **Channel ID**: identifies FlexRay channel (A, B) for frame transmission
  - **Cycle Counter**: optionally identifies communication cycle
  - Frame ID must be globally unique to avoid collissions

FlexRay

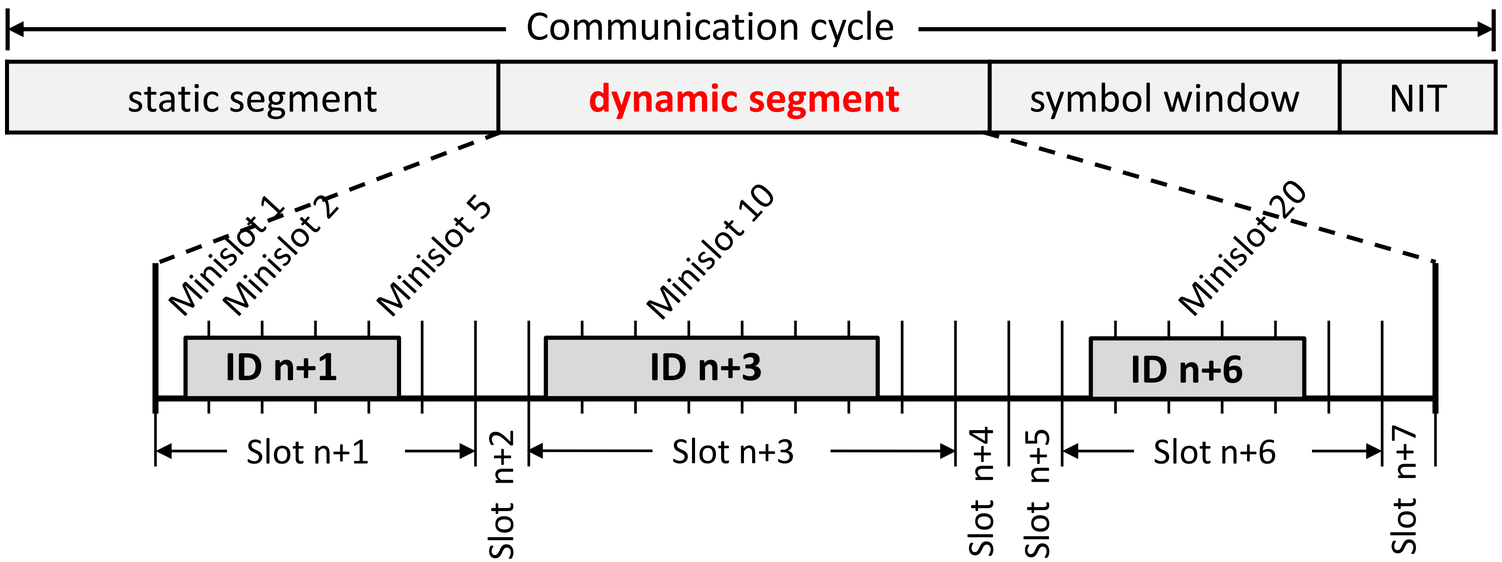
Dynamic segment



- Splitting dynamic segment into **Minislots** of equal size (0 up to 7986 Minislots)
  - Each data transmission requires multiple data transmissions
  - Length of data transmissions are variable, tranmsissions must not exceed dynamic segment length
- Frame is identified by frame ID, channel ID and cycle counter
  - Continues numbering of static frames (maximum slot-ID = 2047).
  - Frame ID defines frame priority (lowest slot is equals highest priority).
  - Frame ID must be unique

FlexRay

Dynamic segment



- Medium arbitration (per channel) : **Flexible TDMA (FTDMA)**
  - All nodes initialize slot counter with n+1 (n is ID of last static slot)
  - Slot counter is incremented when no transmission occurs in minislot
  - If value of slot counter equals a waiting frame ID, this slot starts a transmission
  - Sender must guarantee that his transmission ends within dynamic segment

FlexRay

Slot timing

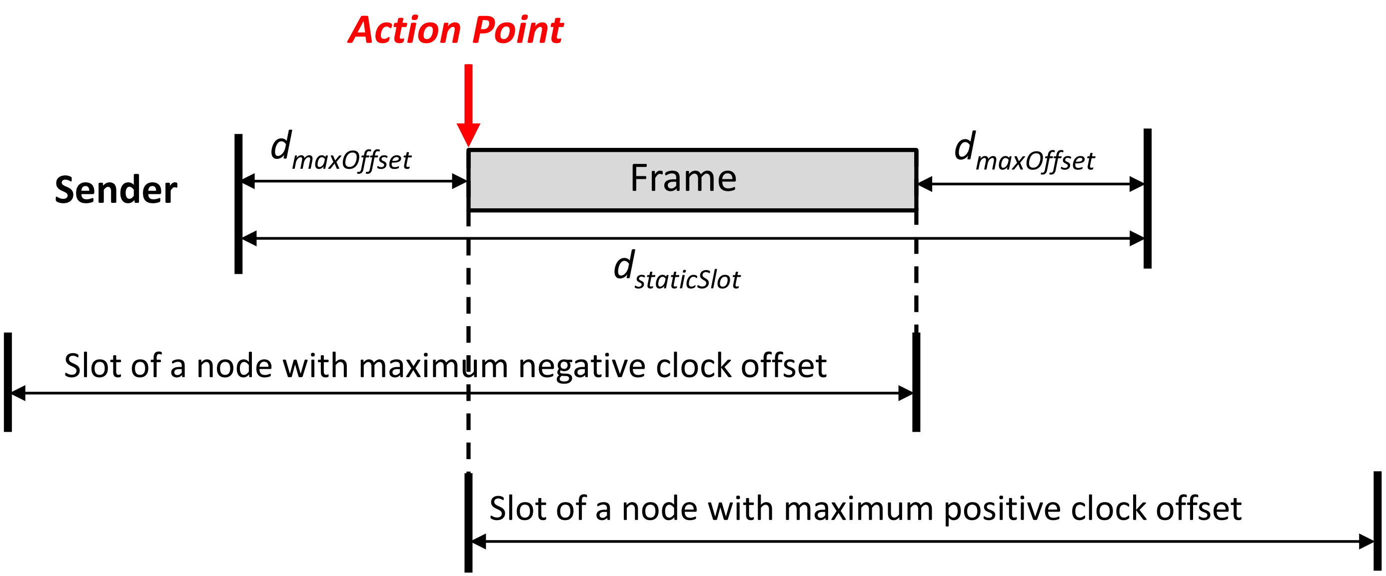
- Frame transmission must start within time slot and also end within that time slot if a static slot is used
- All network nodes (sender, receiver) must perceive this in that way
  - → Avoiding of collisions
  - → Correct assigment of FrameID to SlotID

- Problems
  - Exact time synchronization is not possible, clock offset between nodes remains
  - Clock skew changes (increments) this offset between clocks of network nodes over time

FlexRay

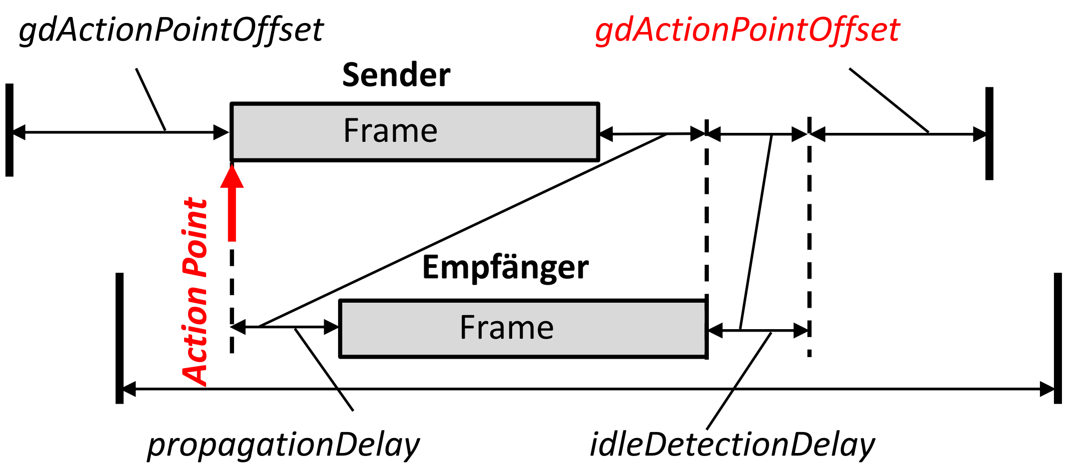
Time synchronization

- Time synchronization with maximum permitted clock offset $d_{maxOffset}$
  - Definition of maximum permitted upper bound for clock skew (minimum HW quality)
  - Consideration of clock offset and clock skew when using time slots
  - **Reduces useable bandwidth**



FlexRay

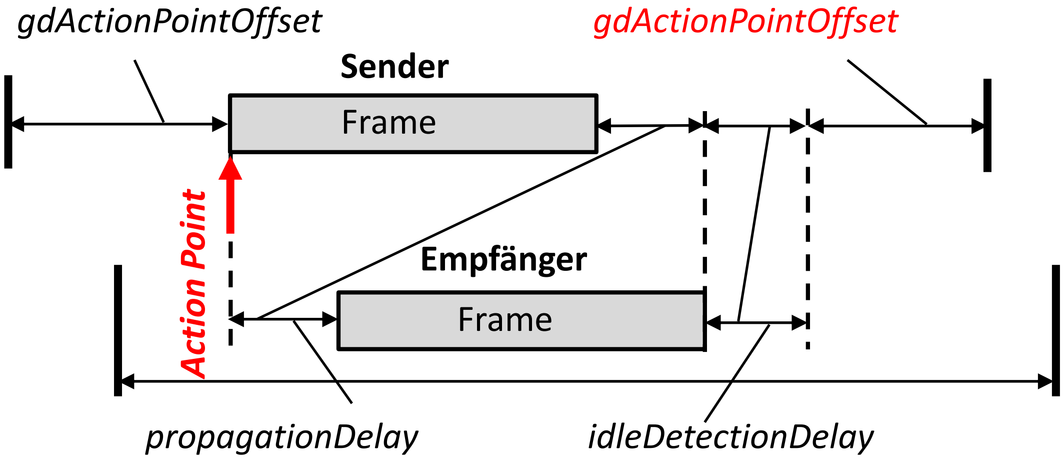
Time synchronization – static segment



- *progagationDelay* [µs]: Propagation of bit in physical medium
- *idleDetectionDelay* [µs]: Duration required for detecting end of transmission

FlexRay

Time synchronization – static segment



- *gdActionPointOffset* [MacroTicks]:
  Difference between start of slot and start of transmission in macro ticks

- Restriction

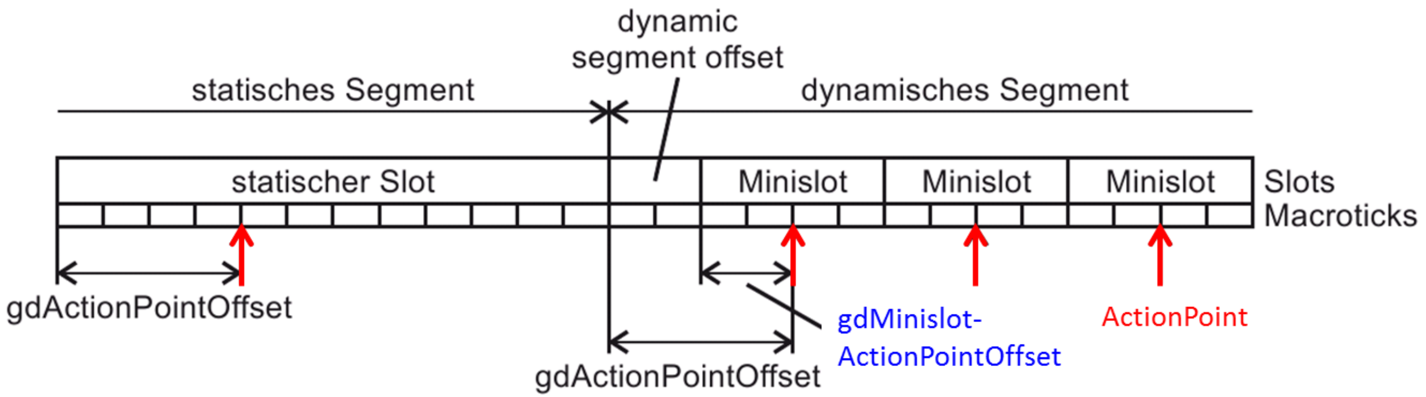
$$gdActionPointOffset \geq \left\lceil \frac{aWorstCasePrecision\ - gdMinPropDelay}{gdMacroTick \cdot (1 - clockDeviationMax)} \right\rceil$$

  - *aWorstCasePrecision*: maximum clock offset (6,675-18,4 µs)
  - *gdMinPropDelay*: minimum propagation delay between two nodes
  - *gdMacroTick*: Macro tick duration (1-6 µs)
  - *cClockDeviationMax*: maximum permitted clock skew (e.g. 1500 ppm, 1ppm = 0,0001%)

FlexRay

aus: M. Rausch, *FlexRay*, Hanser-Verlag, 2008

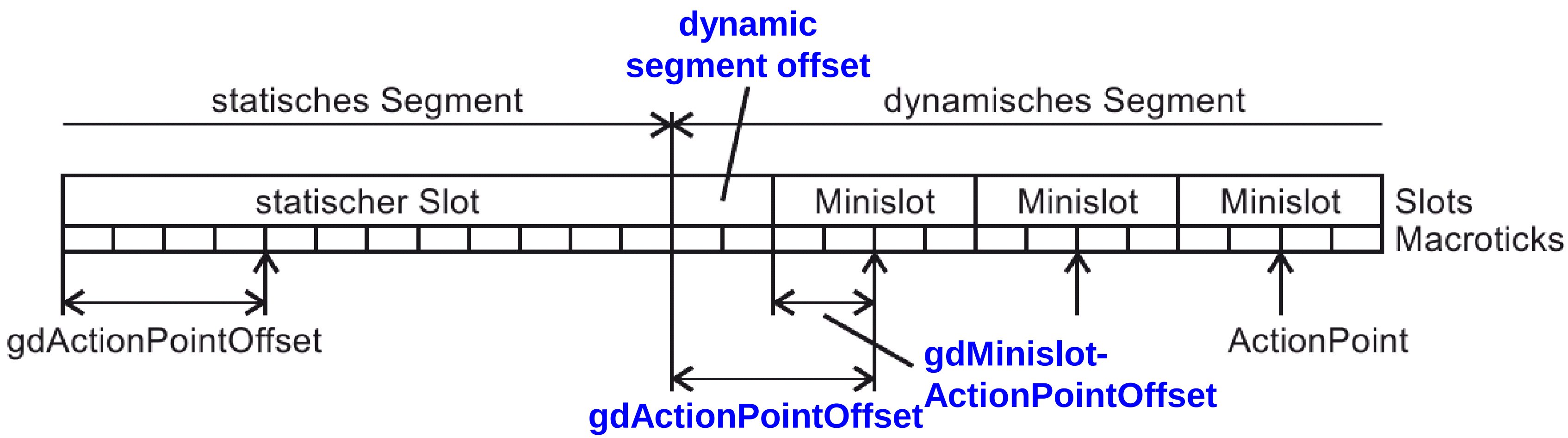
Time synchronization – dynamic segment



- Rationale
  - Start of transmission (Action Point) needs to be assigned similar by all nodes
  - End of transmission must have same value on all nodes as well (relative to Action point)
- Use of parameter *gdMinislotActionPointOffset*
  - Either identical or smaller than *gdActionPointOffset*
  - Smaller value yields less robustness but better bandwidth use
  - *Dynamic Trailing Sequence* is attached to frame to make frame end at mini slot boundary

aus: M. Rausch, *FlexRay*, Hanser-Verlag, 2008

Time synchronization – dynamic segment



- Important
  - Distance between two frames in static segment is $2 \cdot gdActionPointOffset$
  - If *gdMinislotActionPoint* < *gdActionPointOffset*, this does not hold for last static segment
  - Add leading *Dynamic Segment Offset* to start of dynamic segment to compensate

FlexRay

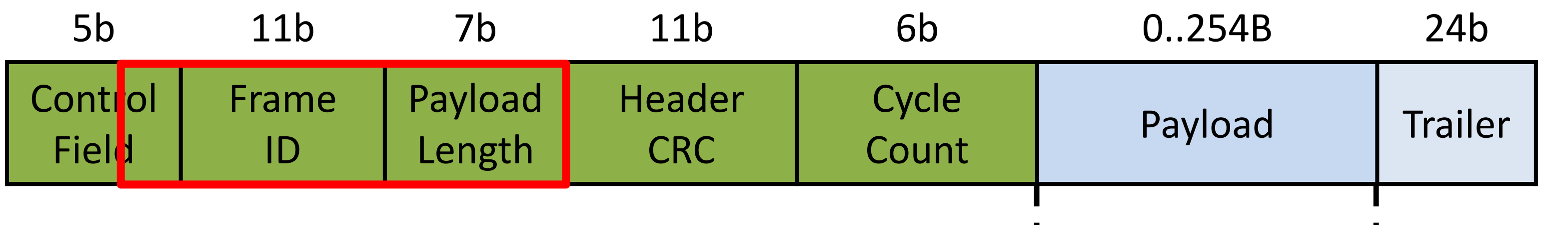
Frame format

5b	11b	7b	11b	6b	0..254B	24b
Control Field	Frame ID	Payload Length	Header CRC	Cycle Count	Payload	Trailer

- Header (first 40 Bit)
  - *Control Field*: Control bits
    - Bit 1: *Reserved*
    - Bit 2: *Payload Preamble Indicator* to tag management data
    - Bit 3: *Null Frame Indicator* to tag empty frame
    - Bit 4: *Sync Frame Indicator* for synchronization
    - Bit 5: *Startup Frame Indicator* for synchronization
  - *Frame ID*: Number of slots for frame transmission (1..2047)
    - Value 0 is reserved for invalid frame
  - *Payload Length*: Number of payload words, 2 Byte per word

FlexRay

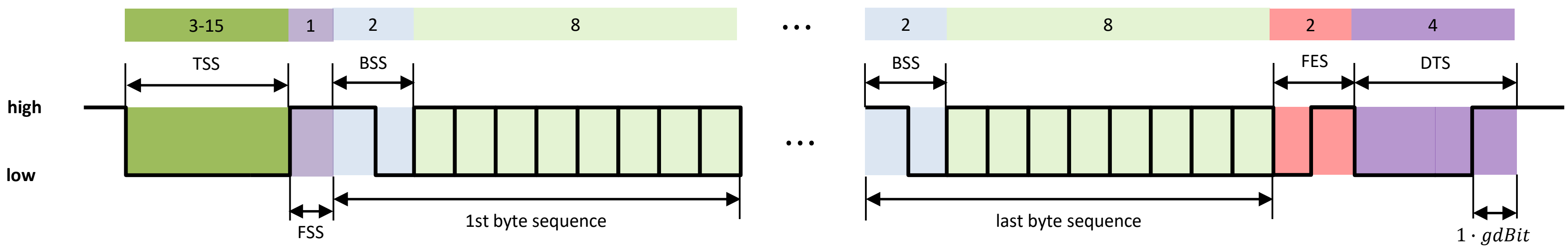
Frame format



- Header (first 40 Bit) (cont'd)
  - *Header CRC*: Checks bits 4 & 5 of Control Field, Frame ID, Payload length
  - *Cycle Count*: Number of communication cycle in which frame is transmitted
- Payload
  - Payload (unit is words, i.e. 2 Bytes)
  - Transmits up to 127 words → 254 Byte
- Trailer
  - *Frame CRC*: Checksum of whole frame, different generator polynoms for physical channels A and B

FlexRay

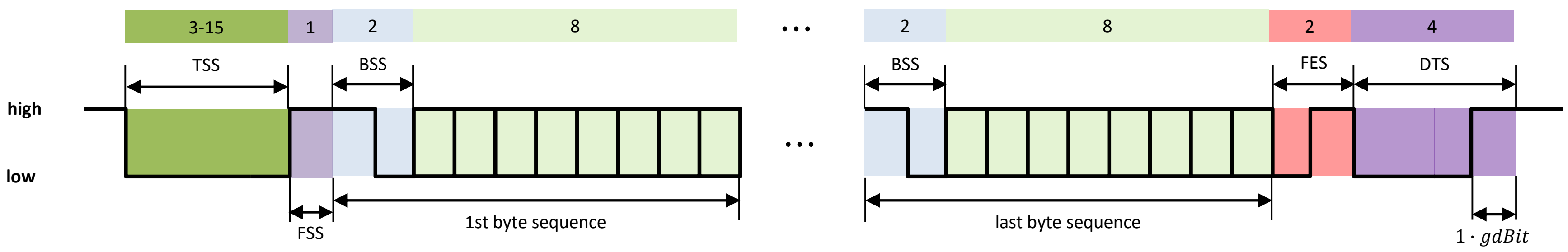
Frame format



- PHY-Preamble (Bit synchronization)
  - Transmission Start Sequence (TSS) activates bus drivers 3-15 Data_0-Bits
  - Frame Start Sequence (FSS): Safety buffer to prevent quantisation errors when sampling (Data_1-Bits)
- Bit resynchronisation: Byte-Start-Sequence (BSS): 1-0-Bit sequence before each Byte → 20% Overhead
- PHY-Trailer
  - Frame End Sequence (FES): 0-1-Bit sequence

FlexRay

Frame format



- Frame in dynamic segment
  - Dynamic Trailing Sequence (DTS):
    - Low-Phase of variable length to fill up to next action point
    - Ensures exact mini slot ID assignment
    - End of dynamic frame: last bit of DTS is high bit

FlexRay

Startup phase

- Goal: Initial time synchronization of all bus participants (TDMA)
  - Create common time base (Makroticks) before message exchange
  - Synchronization to beginning of communication cycle 0
  - Distributed synchronization increases reliability

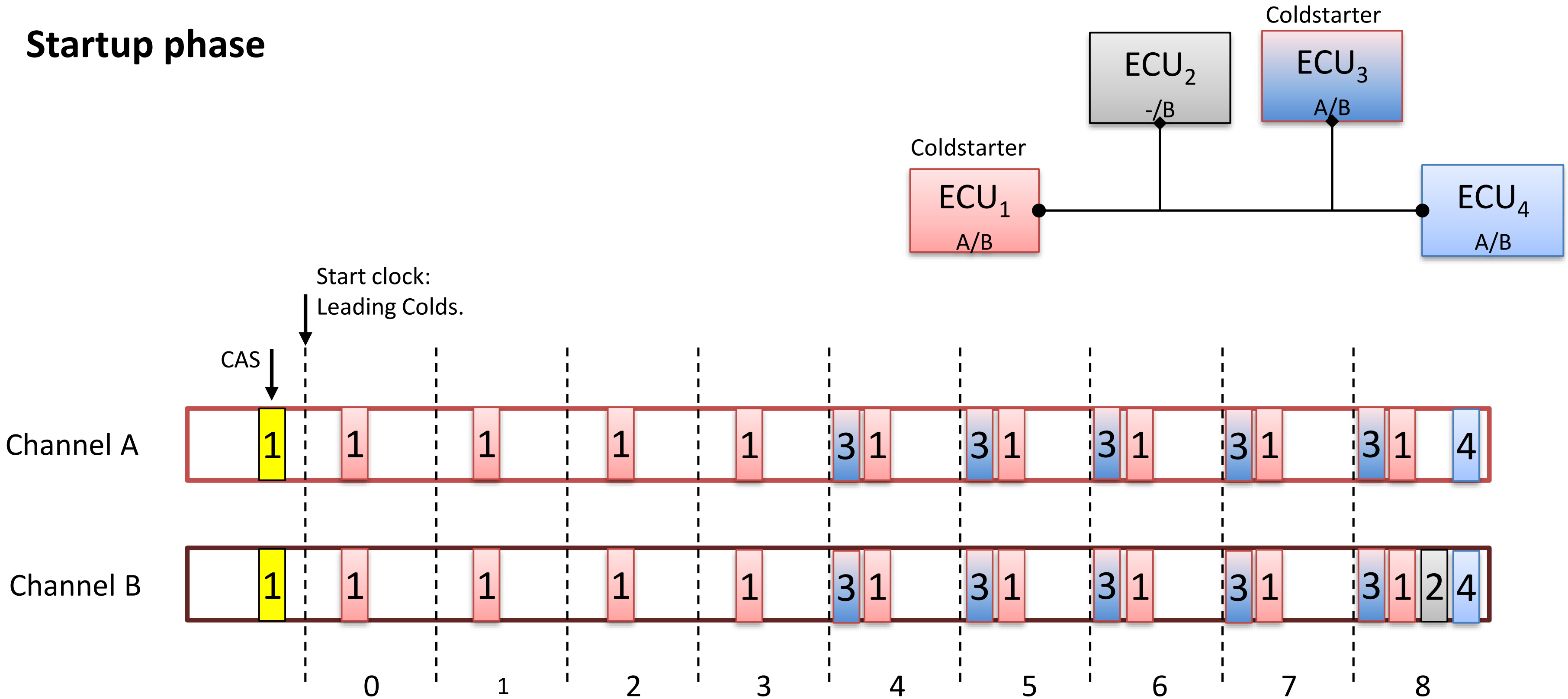
FlexRay

Startup phase

- Startup phase
  - Definition of at least two (redundnacy) cold-start nodes for initial synchronization
  - Each cold start node monitors medium for at least two cycles
    - If inactive, node becomes *leading coldstarter*
    - If active, node becomes *following coldstarter*
  - Leading coldstarter sends unique *Collision Avoidance Symbol* (*CAS*) on both channels
  - Signals communication cycle 0
  - After CAS, leading cold starter sets local clock to 0

FlexRay

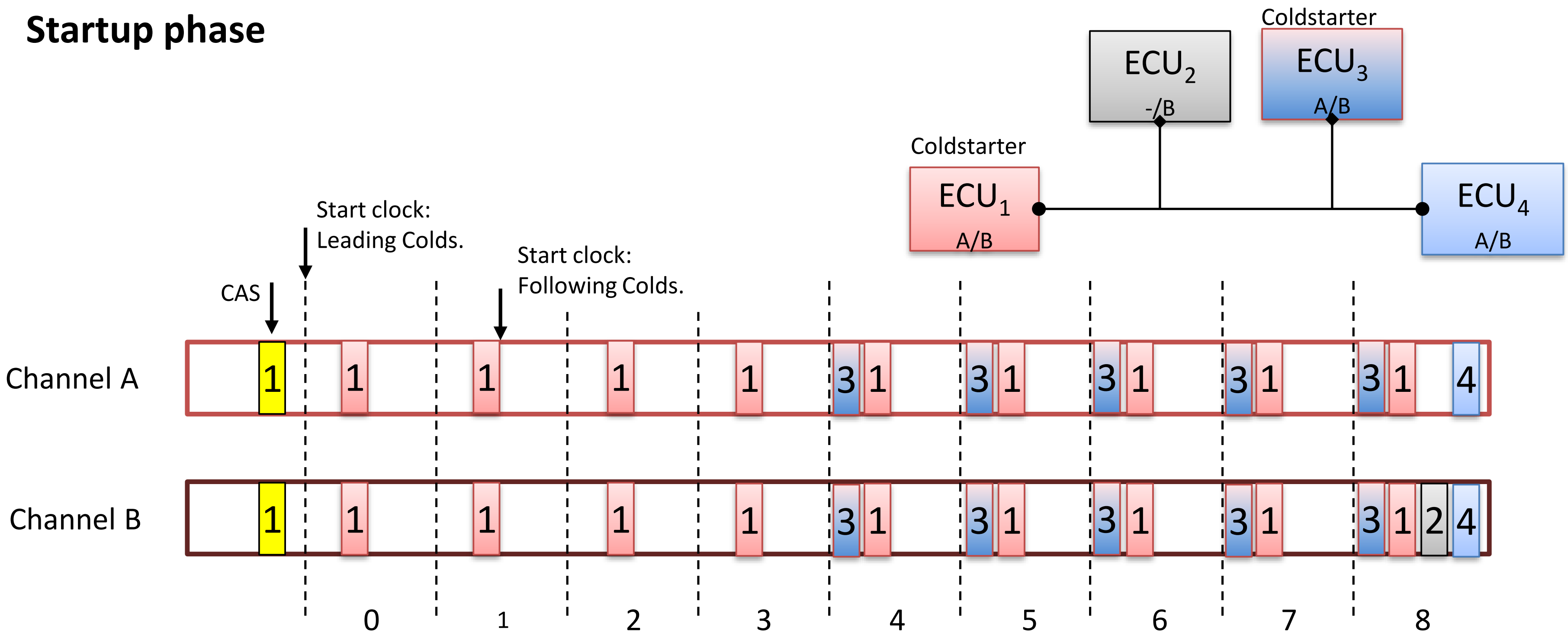
Startup phase



- Leading cold starter sends Startup Frame (SF) in assigned static slot of following communication cycle
  - *Frame has Startup Frame Indicator Bit* and *Sync Frame Indicator Bit* set to high value
  - *Frame ID* and *Cycle Count* define offset to start of communication cycle 0

FlexRay

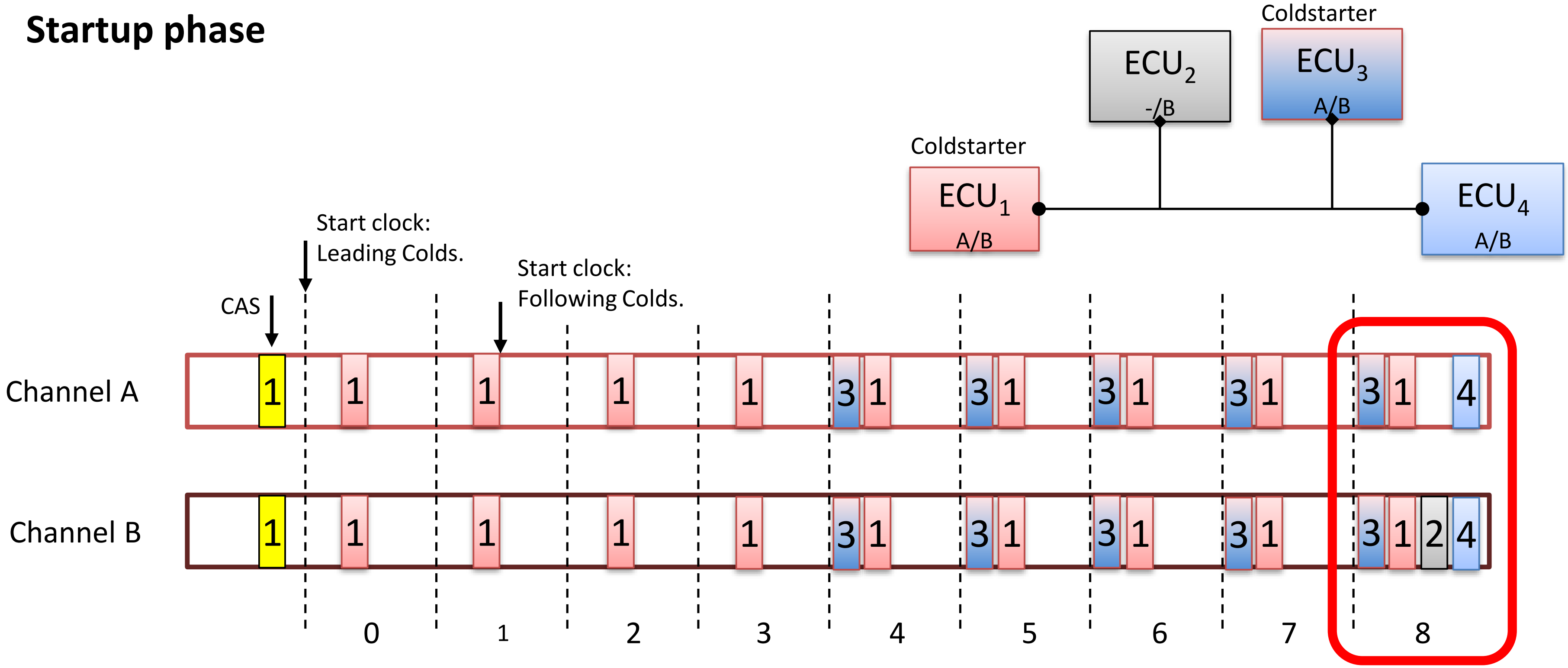
Startup phase



- *Following Coldstarter* waits for reception of 2 SF, then it starts ist local clock
  - Initial clock value is set based on offset to start of cycle 0
  - Clock correction value is calculated by measuring difference between measured and expected distance between received SF
- Waits for reception of two more SF, then starts active synchronization by sending additional SF in his slots

FlexRay

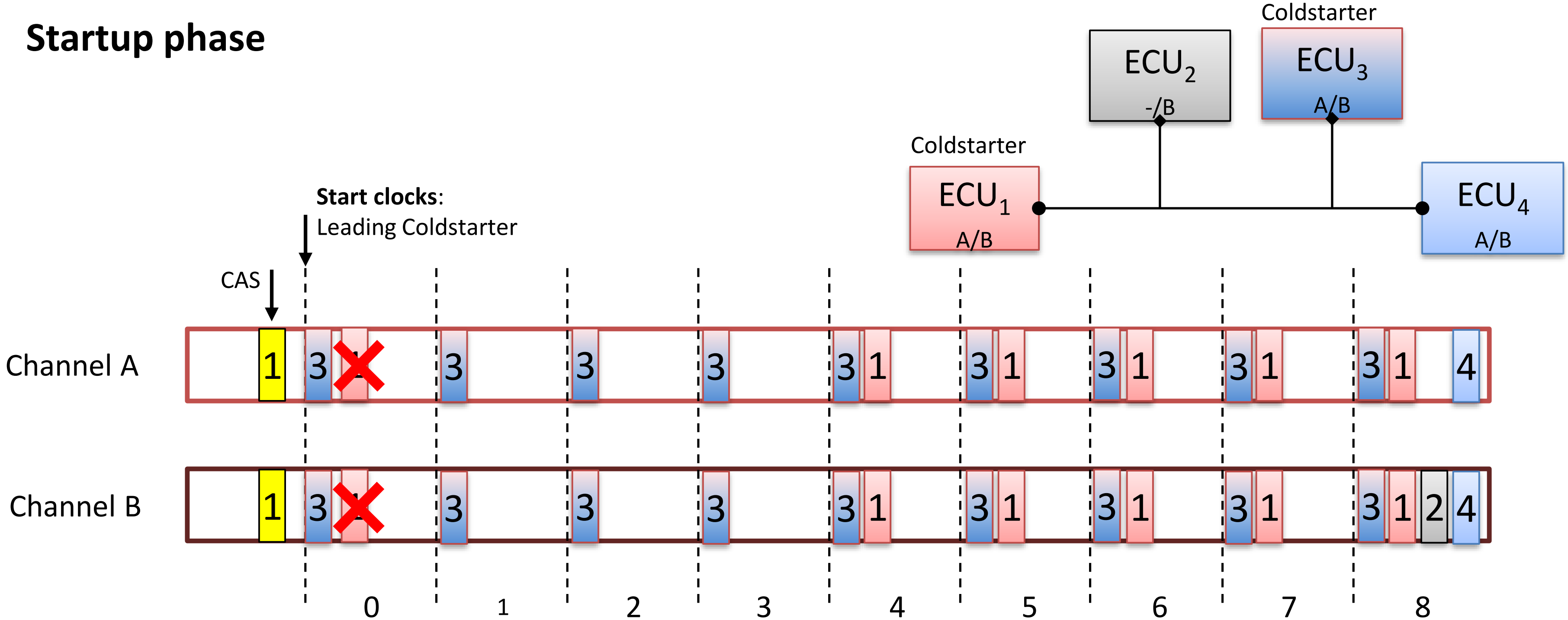
Startup phase



- Repeats startup-sequence if no following cold starter was detected after 6 cycles
  - From cycle 8: normal FlexRay communication

FlexRay

Startup phase



- Scenario: 2 Coldstarter send (almost) concurrently a *CAS*
  - *Both* nodes become leading coldstarters and start their local clocks
  - One of both nodes transmits his startup frame first (fixed static slot)
  - Other node detects that, terminates ongoing startup phase and restarts
  - Becomes following coldstarter after restarting startup sequence

FlexRay

Startup phase

- Integration of additional (non-coldstart) nodes
  - Start and initialization of local clocks when receiving first startup frames
  - Calculation of clock correction value based on difference between distance of measured and expected start of Startup-Frames
  - Active communication after consecutive reception of four startup-frames of at least two coldstart nodes

- Measures for periodic re-synchronization
  - 2..15 synchronization nodes (incl. coldstart nodes)
  - Periodic transmission of sync frames (regular Frames with sync. Indicator bit set)
  - Transmitted synchronously on both channels in reserved static slots
  - Frames correct clock values in NIT-Segment (Network Idle Time)

FlexRay

Failure detection and handling

- **Bit failure**
  - Detection: Header CRC, Frame CRC
  - Handling: Dropping frame
    - No automatic repetition to ensure deterministic operation

- **Synchronization failure**
  - Detection: Checking of time based behavior
    - Local cycle counter ≠ Cycle Count in received frame
    - Local slot counter ≠ Frame ID in received frame
    - Clock control values exceed permitted limits
  - Handling: Resynchronization
    - Changing clock state to „normal passive“
    - No active participation on bus (only receiving data)

FlexRay

- **„Babbling Idiot“ Problem**

  - Detection: Checking of bus transmissions / bus use
    - Station tries to transmit at wrong times
    - Bus Guardian enables medium access for transmission only in specific slots

  - Handling: Prevented by **Bus Guardian**
    - Bus Guardian prevents bus access if node behavior is incorrect (redundancy)
    - Node moves itself to „halt“ state