Field buses (part 2): time triggered protocols

Nico Fritz
Universität des Saarlandes

CAN and LIN

	LIN	CAN
Type	Serial communication protocol event triggered	
Arbitration	Master/slave	Decentral by message priorities
Transfer rate	20kbps	20kbps – 1Mbps
Relative costs per node	0.5	1
Average number of nodes	8-16	20-40

Distributed real-time systems

- Safety-critical system demands:
  - Determinism required: reception time of messages must be known
  - Fault tolerance, i.e. redundant bus
  - Higher bit rate required (faster message cycling)
- Example: Break-by-wire
  - Sensor at the break pedal sends to all 4 breaks
  - Each break sends its understanding of the sensor data to all other breaks
  - Each brake applies the brake force appropriately
  - Inconsistencies must be recognized and handled
- Some needs can‘t be matched by the event-triggered protocols

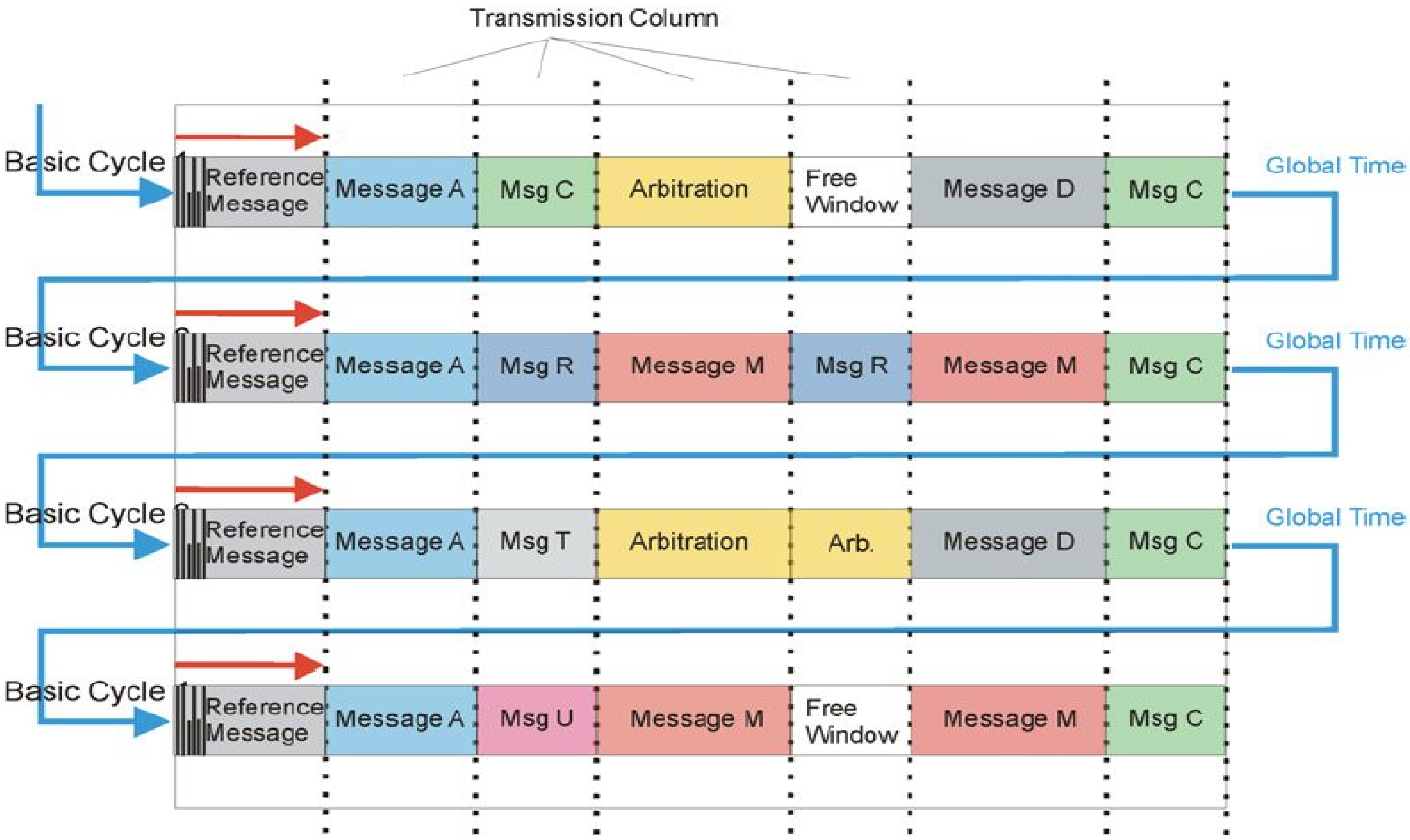
Time Triggered Protocols

- Global time
- Exact time point of a certain message is known (determinism)
- Real time capable, for safety-critical systems
- Each node gets a time slot in the transmission loop where only it can send a message
  - No arbitration necessary
  - Less flexible
- Examples:
  - TTCAN
  - Flexray
  - TTP/C

TTCAN

- Enhancement of CAN to time triggered model
- All nodes have a global time, set by a reference message sent by a time master
  - More than one time master with different priorities
- Each node knows when it should send or receive a message
- Each message gets its own time window
- A system matrix determines the time windows

TTCAN system matrix



TTCAN

- Possibility to include the event triggered CAN in so called arbitration windows
- No redundancy support
- Data rates must not exceed the CAN limitations
- Suitable for the first generation of x-by-wire systems
- CAN based monitoring and analyzing tools can be used

FlexRay

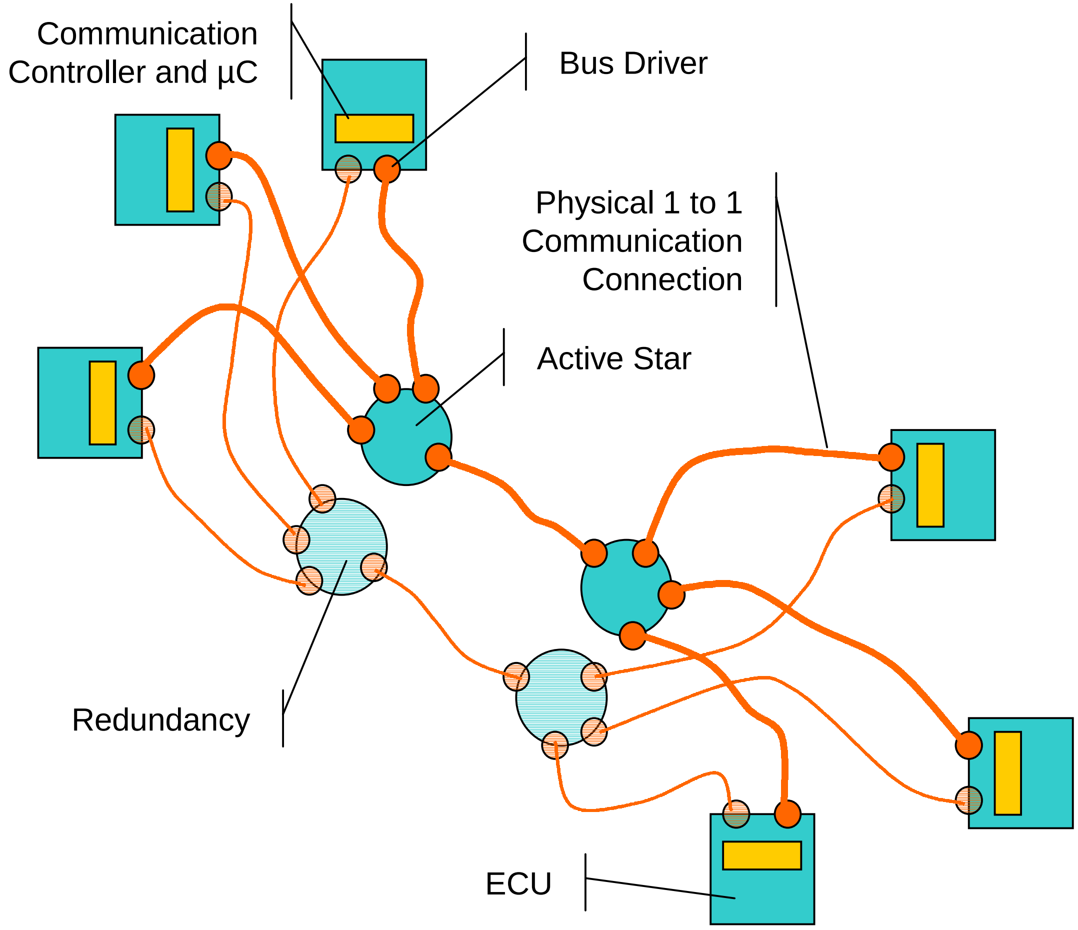
- Developed by Daimler-Chrysler, BMW and some semi-conductor manufacturers in 1999
- Free, not commercial
- Standardized
- Real time capable
- Target transfer rate: 10Mbps
- Synchronous and asynchronous (based on Byteflight) transmission
- Time triggered, no arbitration
- To fulfil future demands

FlexRay - Features

- Scalability
- Bus access via TDMA (Time Division Multiple Access)
- Deterministic data transmission, guaranteed message latency
- Support of optical and electrical physical layer
- Support of redundant transmission channels
- Error containment on the physical layer through independent “Bus Guardians”
- Fast error detection and signalling
- Up to 64 nodes on the bus
- Various topologies

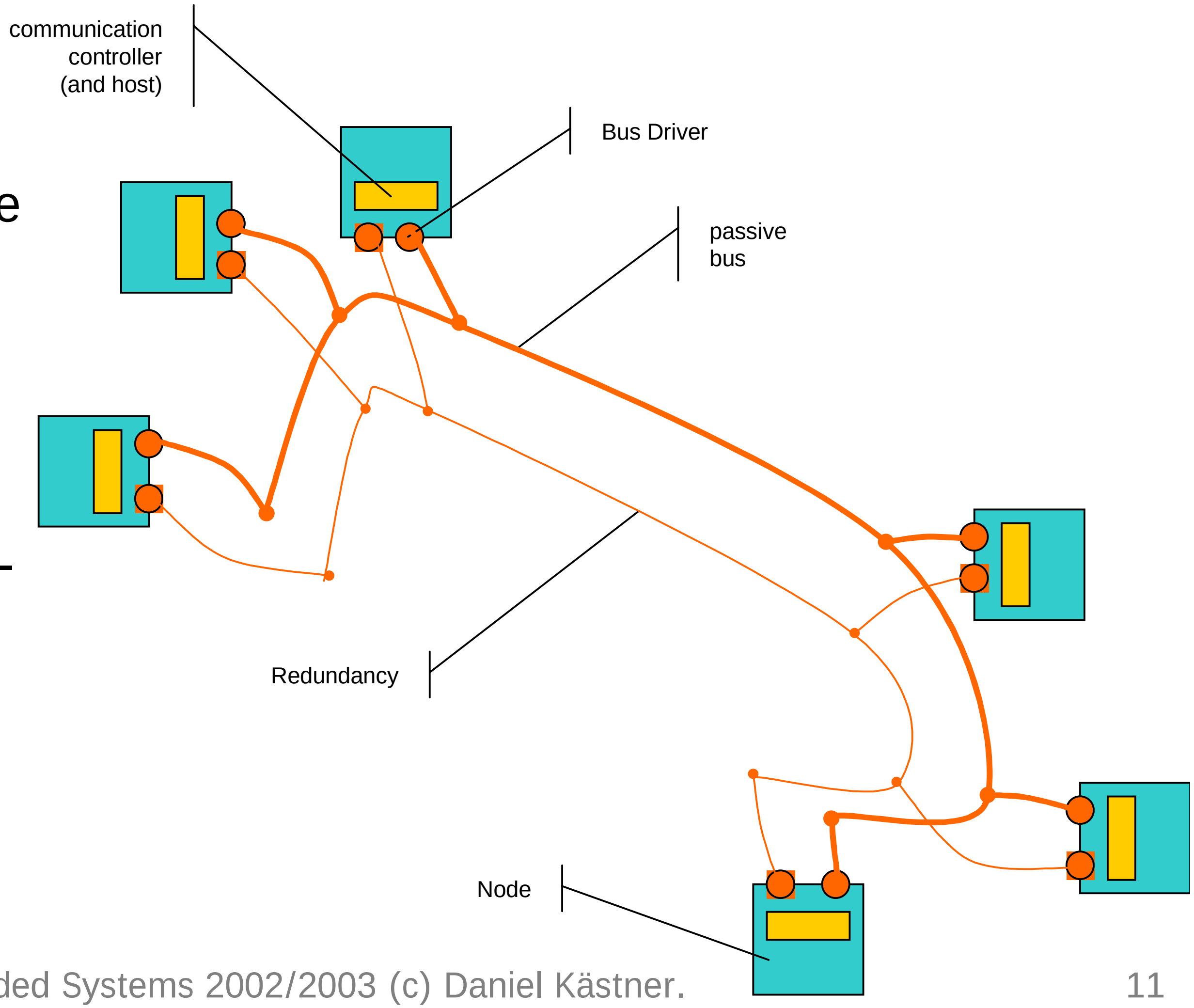
FlexRay - Topologies

- Active Star:
  - Optional redundant communication channels
  - 1 to 1 communication connections in combination with active stars
  - Support of “wake-up” via bus
  - Support of net data rates up to 5 Mbit/sec



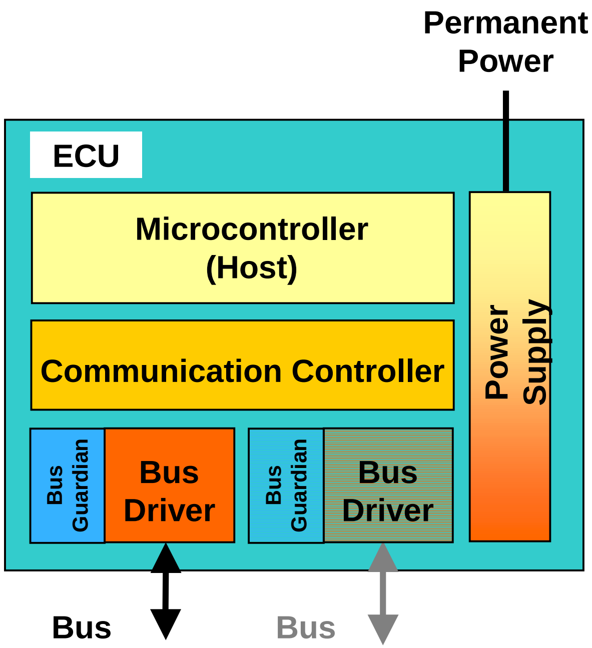
FlexRay - Topologies

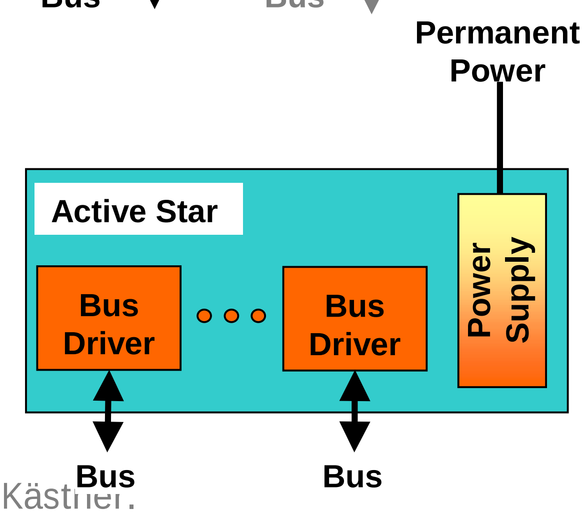
- Passive bus
  - Solution with restrictions (reuse of available Physical Layer):
  - Optional redundant communication channels
  - Support of “wake-up” via bus, depends on the Physical Layer
  - low gross data rate (similar to CAN)



FlexRay – ECU & Active Star

- ECU (Electrical Control Unit):
  - CC regulates the communication between host and the bus
  - BG is controlled by the CC and watches the time slots and other safety relevant functions
  - Each ECU is connected to a power supply which is connected to a permanent power (vehicle battery)
- Active Star:
  - No host or CC required
  - Many bus driver needed



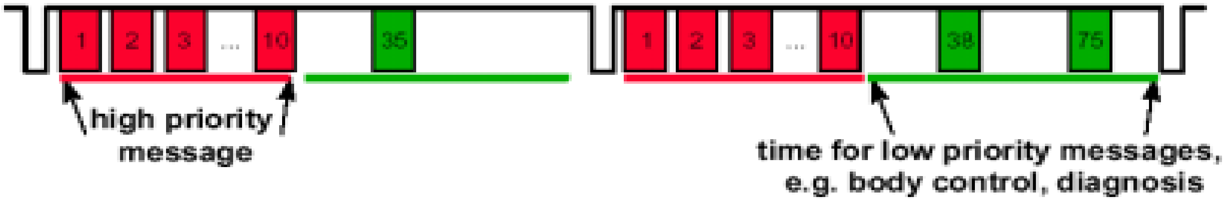


FlexRay – Data transmission

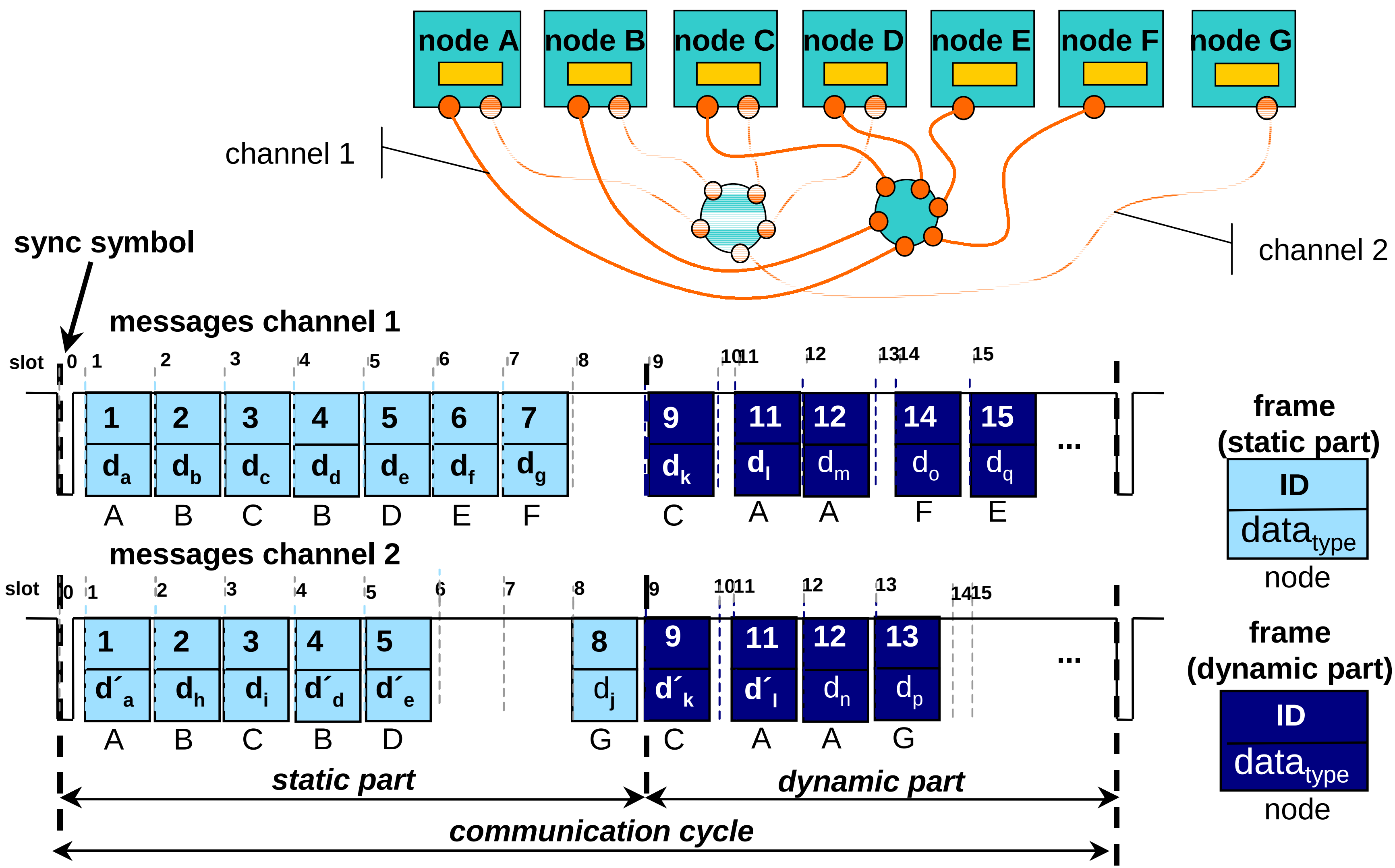
- Communication cycle is divided in an static and a dynamic part
- Static part:
  - For high priority messages
  - Each node gets his time slot(s)
  - Nodes connected to both channels and send synchronously on both
- Dynamic part:
  - For lower priority messages
  - Messages are sent according to the Byteflight protocol
  - Variable slots
  - Flexible TDMA

FlexRay – dynamic part

- Byteflight protocol
  - Some node is declared synch master
    - The synch master determines the cycle time, so the cycle time is fixed.
    - Current implementation: cycle time 250 µs, data rate 10Mbps
  - Each node has one or more unique Ids
  - A slot counter tells each node when it may send
    - The counter is stopped during transmission
    - Increasement after transmission or some waiting time t_wx



FlexRay – Data transmission



FlexRay – Frame format

ID	MUX	SYNC	LEN	DATA	CRC
10 bit	1 bit	1 bit	4 bit	0...12 byte	16 bit

ID: Identifier, 10 bit, range: (1_{10} ... 1023_{10}), defines the slot number in the static part or the priority in the dynamic part

MUX: Multiplex Field, 1 bit, enables a node to transmit different messages with the same ID

SYNC: Synchronisation Field, 1 bit, tags the frames which will be used for the clock synchronisation

LEN: Length field, 4 bit, LEN = number of used data bytes (0_{10} ... 12_{10})

D0 ... D11: Data bytes, 0-12 bytes

CRC: Cyclic Redundancy Check Field, 16 bit

FlexRay – Frame format

ID	MUX	SYNC	LEN	CYCLE	DATA	CRC
10 bit	1 bit	1 bit	4 bit	8 bit	0...11 byte	16 bit

CYCLE: Cycle Counter, 8 bit, range: (0_{10} ... 255_{10}). The CYCLE-field will be used as cycle counter or as a data byte. The cycle counter will be incremented consistently in all communication controllers at the beginning of each communication cycle

SYNC: Synchronisation field, 1 bit, tags the frames which contain the cycle counter

D0 ... D10: Data bytes, 0-11 bytes

FlexRay - Synchronisation

- Only possible in the static part of the transmission cycle
- Pure dynamical systems are synchronized by a bus master
  - A failing bus master must be handled on application level
- At start-up a node waits for timeout and sends its messages. As soon as another node synchronizes on that message communication begins

TTP/C

- TTP/C = Time Triggered Protocol Class C (hard real-time system)
  - Part of the Time-Triggered Architecture (TTA)
    - Designed as a solution for safety-critical applications
  - Scaled-down version called TTP/A (soft real-time systems)
- Designed by the TTA-Group
  - Development begun in the early 80‘s (university of Vienna)
  - Airbus, Audi, VW, Renault, TTTech, Delphi, Esterel, …
- Goal
  - Safety critical protocol with low costs
  - Target applications are for example X-by-wire
- Data rate: 25Mbps with today's controllers

TTP/C - Features

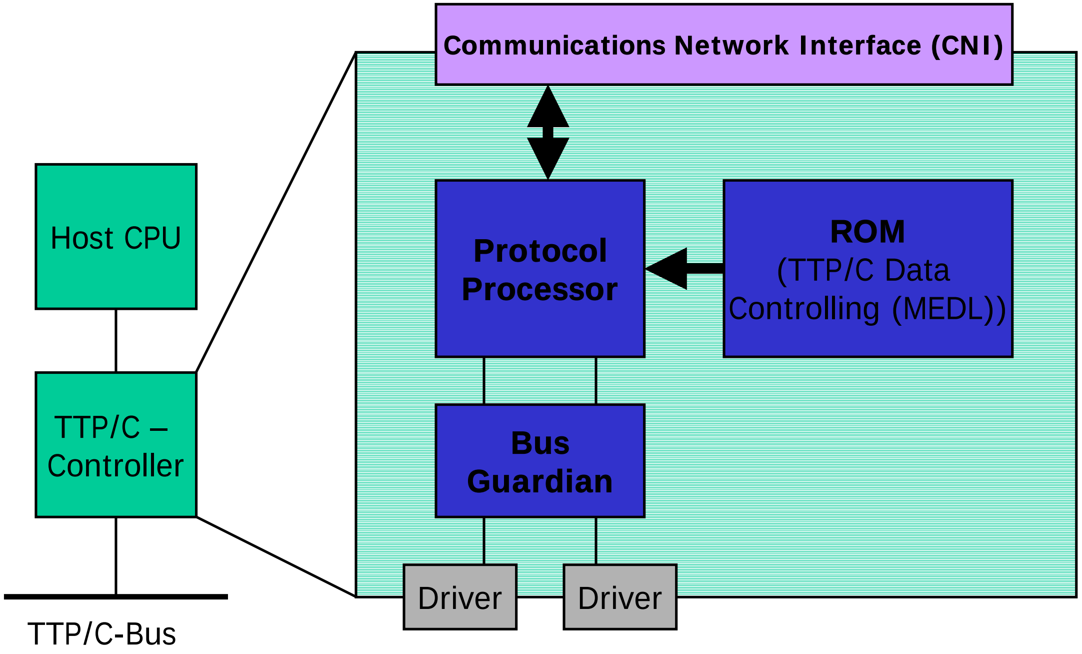
- Autonomous fault-tolerant message transport with known delay between nodes of a cluster
  - Messages are send host independent, the host only provides the data
- Membership for nodes:
  - Every node is informed about the health state of every other node
  - Defective nodes are excluded from membership until they restart with a correct protocol state
- Clique avoidance
  - Detect and eliminate formation of cliques
- Fault tolerant clock synchronization

TTP/C - Features

- Topologies (same as mentioned with FlexRay)
  - Active star
  - Passive bus
  - Combinations
- Shadow nodes:
  - Can take the place of former failed active nodes
  - So if an active node fails it can be deactivated and replaced by its shadow node

TTP/C - Node

- Any host only communicates through TTP/C-controllers
- Controller synchronizes on its own by comparing with the MEDL and adjusting

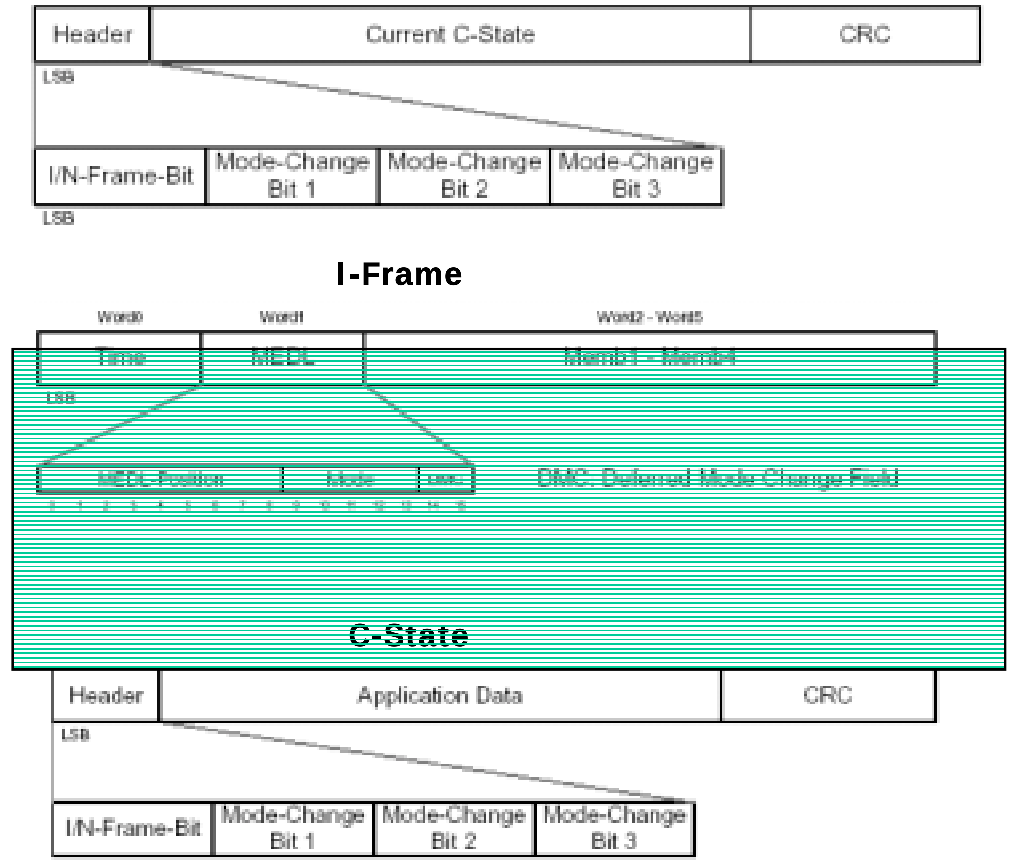


TTP/C - MEDL

- MEDL = Message Descriptor List
- Control information are saved there at initialisation, so that a controller can work without getting control signals by the host
- Data:
  - For each message to send the time point and the CNI address
  - For reception the time point and the CNI address
  - Additional control flow information

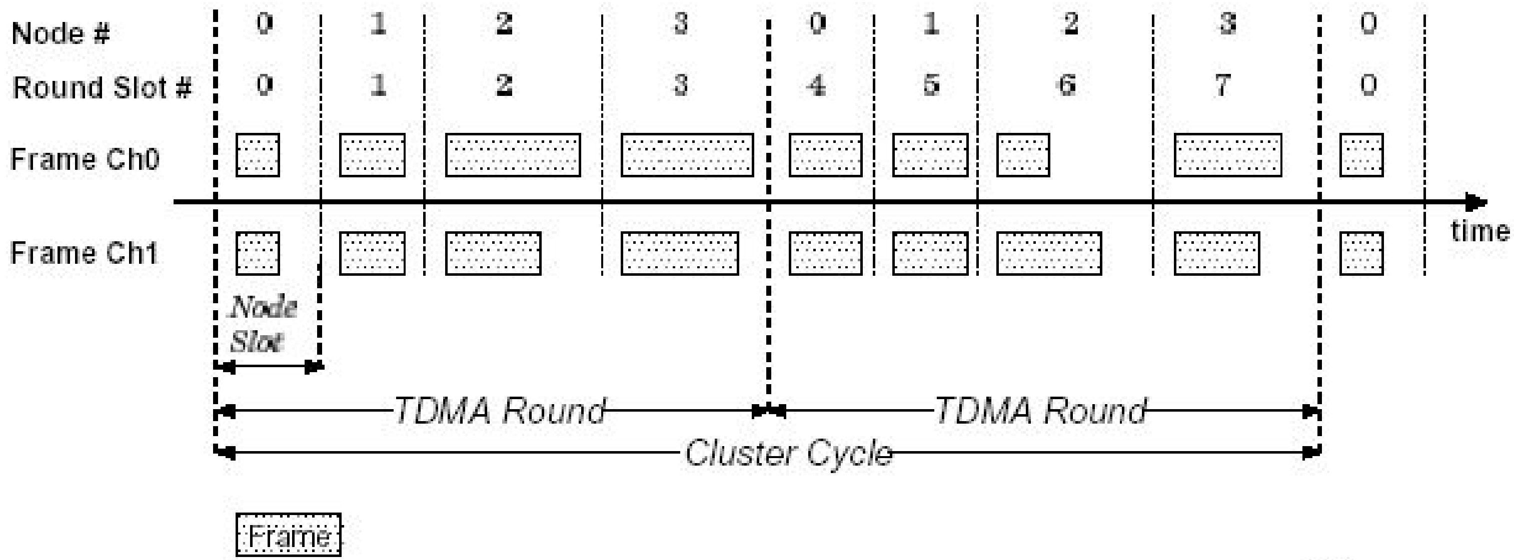
TTP/C - Frames

- I-Frames are used for initialisation and resynchronisation
- C-State is the current controller state including global time, MEDL position and current mode
- N-Frames are used to send data
- All frames contain a CRC field for error detection

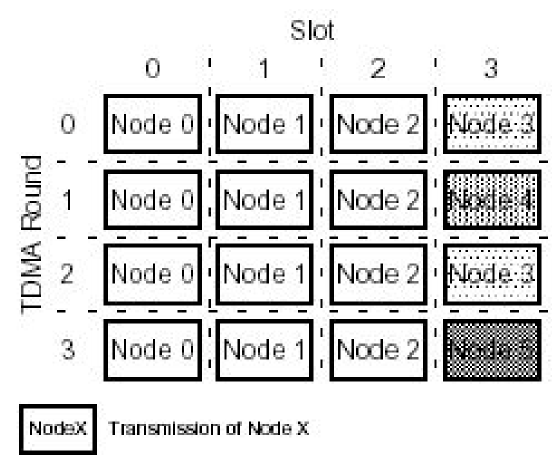


N-Frame

TTP/C – Cluster Cycle



- A cluster cycle contains several TDMA rounds
- A TDMA round contains several node slots
- The message a node sends in its slot varies in different TDMA rounds
- A node slot can be multiplexed, i.e. used by different nodes in different TDMA rounds



FlexRay vs. TTP/C

- FlexRay:
  - More flexible (free specification of synch. and asynchronous part)
  - Some safety features not within the controller but left to the application
  - Still under development
- TTP/C:
  - More integrated safety features
  - Less flexible (no asynchronous transmission)
  - More mature
    - Already 20 years of development and testing
    - 2nd generation controllers available
  - Lower implementation costs (only one guardian per node)

Summary

- For safety-related systems time-triggered protocols are needed because of their determinism
- Future applications demand an increase in band with, so high data rates must be supported
- TTCAN is only a temporary solution
- FlexRay is more flexible than its competitor TTP/C but is still under development
- TTP/C is the most mature protocol for future safety-critical systems at the moment

References

- Bussyteme im Automobil, Thomas Dohmke
- TTP High-Level Specification Document, TTTech
- The Time-triggered architecture, H.Kopetz, G.Bauer
- A Comparison of Bus Architectures for Safety-critical Embedded Systems, J.Rushby
- Time Triggered Communication on CAN, Bosch
- www.flexray-group.com (docs and presentations)
- www.tta-group.org