CAN - Byteflight – Flexray – TTP/C

Technical comparison of protocol properties with a focus on safety related applications

CAN: application tells communication controller to send/receive messages with a certain identifier (11 or 29 bits), ie, priority; the communication controller tries to send immediately, but can be delayed by current traffic, and by arbitrating against higher-priority messages currently transmitted by other nodes.
→ Messages have a transmission latency that depends on the concurrent higher-priority messages; changing the message set/priority/transmit rate results in changes of the communication timing that are difficult to predict.

Identifiers can be added to an existing network by only changing the sender and relevant receivers, but the message timing changes for all nodes.
→No composability on the communication level.

Arbitration between concurrent messages is non-destructive and therefore efficient, but limits the bit rate (< 1 Mbit/s) and the topology (no star or ring, no repeaters) and has severe safety issues ("babbling idiot" possible).

CAN – Byteflight – Flexray – TTP/C

Byteflight: each node is assigned a set of cluster-wide unique telegram IDs to use for transmission (same as for CAN, but 8 bits per ID). A master starts a minislotting rounds with a SYNC pulse every 250 microseconds.

For each ID value a minislot counter runs on all nodes; during minislot “x” the owner of ID “x” must send (“static telegrams”) or can send (“dynamic telegrams”). Unused IDs or IDs from failed nodes result in short minislots (depends on propagation delays, ca. 1-2 µs), while used IDs result in the transmission of a telegram from one node.

Except for telegram “1”, the exact time of transmission in a round is therefore unknown; the protocol is event-triggered, protocol error containment should be provided by an intelligent star coupler node.

Composability is based on the idea of “fast oversampling”, the application does not have (need?) information about the exact communication timing or communication errors.

CAN – Byteflight – Flexray – TTP/C

Flexray: Communication is separated between a “TDMA” part and an “event” part. The TMDA part is defined during the startup sequence by all participating nodes, based on transmission of unique slot identifiers. In the event part, the nodes perform the Byteflight protocol.

Flexray performs distributed clock synchronization and allows mixing single- and dual-channel nodes in one cluster, but provides no consistency between them.

Membership and acknowledgement services are not included in the protocol and are implemented in the application if needed.

No public protocol specification is available at present. The information about Flexray presented here is based on the data given on www.flexray.de and in discussions and may be subject to change (status: August 2002)

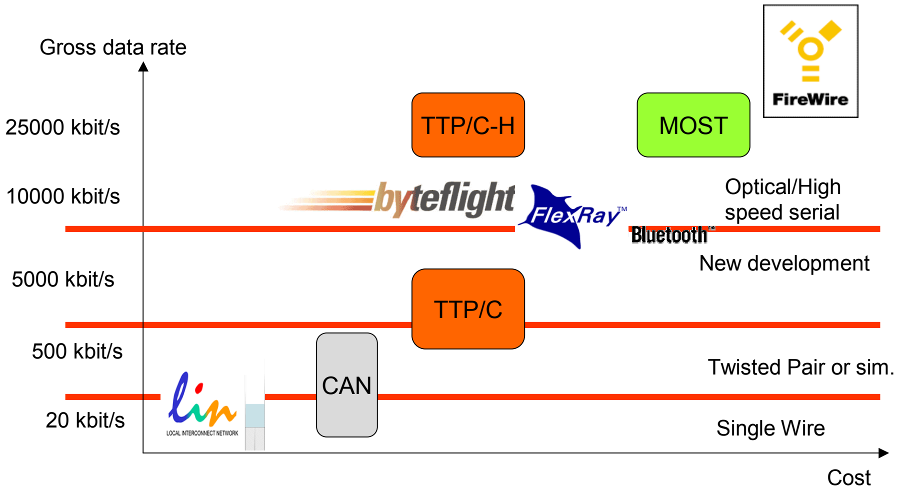
CAN – Byteflight – Flexray – TTP/C

TTP/C: each node is assigned a portion of the available bandwidth (“sending slot”) and transmits once in each TDMA round; all transmission and reception is controlled by a global time-base which is autonomously established by the communication controllers using a distributed fault-tolerant synchronization algorithm. The bandwidth assignment is statically stored in each TTP/C controller (“MEDL”) and cannot be changed or violated by any node even in a fault scenario.

Fault tolerance and composability are integral components of the protocol design, based on the “Single Fault Hypothesis” and fully deterministic communication behaviour of all nodes.

TTP/C provides a consistent message base (“atomic broadcast”) using a membership service and continuous acknowledgement. The protocol autonomously detects communication faults even outside of the fault hypothesis by C-state agreement.

Protocol Comparison



Protocol Groups

Class A, -20 kbit/s: LIN, L-CAN

Class B, 50-500 kbit/s: CAN, J1850

MMedia, > 20 Mbit/s: MOST, Firewire

Wireless: GSM, Bluetooth

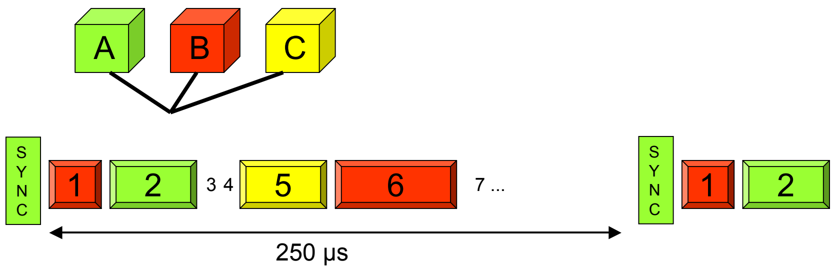
Safety: Byteflight, TTP/C, Flexray

Byteflight

BMW, already in production in the ISIS safety system

Minislotting (round oriented), telegrams with identifiers and 1-12 bytes of application data

Global SYNC-Pulse (time-base) by Clock Master

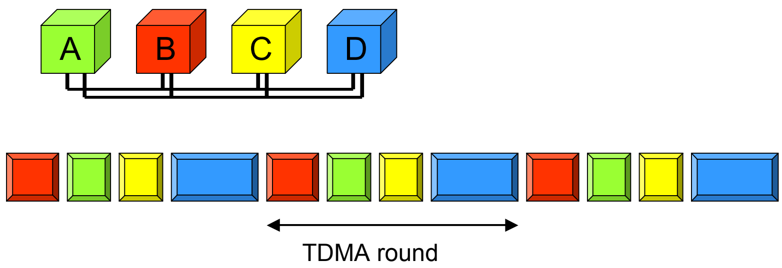


TTP/C

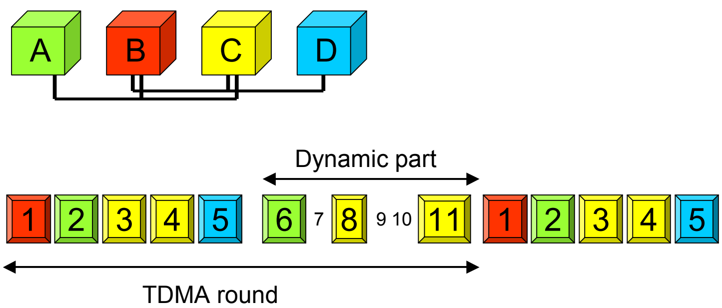
Prof. Kopetz / DaimlerChrysler / TTTech

Strict TDMA with fault tolerant distributed clock synchronization and consistent membership.

Design criteria: No Single Point of Failure



Flexray



Daimlerchrysler, BMW and development partners

Mixture of TDMA and Byteflight

Design criteria: Safety AND Flexibility

“Typical Application” (real or presented):

CAN: any embedded network with 2-10 ECUs, soft real-time requirements and loop times of 5-50 milliseconds

Byteflight: airbag inflation system with ca. 12 ECUs and fast response time requirements

Flexray: brake-by-wire (concept shown) with safety requirements

TTP/C: x-by-wire system (e.g. “FILO car”, avionics safety systems) with 4-32 ECUs and high safety requirements

Physical Layer and Topology:

CAN: up to 1 Mbit/s over twisted pair (ISO 11898-2), bus (star)

Byteflight: 10 Mbit/s over plastic optical fiber, star

Flexray: up to 10 Mbit/s, twisted pair/bus or star, optical fiber/star

TTP/C: up to 2 Mbit/s over twisted pair (ISO 11898-2), bus or star
up to 5 Mbit/s over twisted pair (RS-485), bus or star
up to 5 Mbit/s over optical fiber, star
25 Mbit/s over Ethernet PHY (100BaseT/F), star
1 Gbit/s over Ethernet (research project), star

Note: Due to the lack of any priority based arbitration, TTP/C is the only protocol which can handle long propagation delays efficiently, e.g., in multi-star configurations.

CAN – Byteflight – Flexray – TTP/C

Packet Size, Protocol Overhead and typical efficiency:

CAN: - 8 bytes, ca 4-6 bytes overhead (message identifier/size, start/stop/stuff bits, message CRC), ca. 25-35% typ. efficiency

Byteflight: - 12 bytes, ca. 4 bytes overhead (message identifier/size, message CRC) plus start/stop/stuff bits, no data available

Flexray: - 12 or -246 bytes, ca. 4 - 8 bytes overhead (message identifier/size, message CRC) plus start/stop/stuff bits?

TTP/C: - 240 bytes, ca. 4 bytes overhead (frame header 4-8 bit, frame CRC 24 bit) plus inter-frame gap, ca. 60-80% typ. efficiency

Note: Time-triggered protocols do not need identifiers on the protocol level, they can still be implemented (with any size) on a higher level if needed.

CAN – Byteflight – Flexray – TTP/C

Protocol Services (in addition to message multicast and checksum):

CAN: acknowledgement bit and immediate retransmission (not safe for real-time systems)

Byteflight: SYNC pulse “clock” from master node as global time-tick

Flexray: distributed clock synchronization (state + rate correction), bus guardian for TDMA transmissions

TTP/C: distributed clock synchronization (verified), consistent membership and clique detection, consistent broadcast, implicit (=efficient!) acknowledgement, consistent mode changes, transparent shadowing (“hot spares”), bus guardian for all transmissions

CAN – Byteflight – Flexray – TTP/C

Fault Hypothesis and Fault Tolerance Strategy:

CAN:	**"tolerates" communication fault by retransmission, no error containment or support for higher level fault tolerance**
Byteflight:	**optical physical layer and fast retransmission, error containment by star coupler is application specific**
Flexray:	**TDMA slot scheme protected by node local bus guardians, no error containment for "dynamic" part**
TTP/C:	**"single fault" hypothesis, strong error containment by bus guardian, full predictability allows maximal error containment. Formal verification for core mechanisms of the protocol.**

CAN – Byteflight – Flexray – TTP/C

System Design Guidelines for Safety Related Applications:

CAN:	**not intended for safe systems, redundancy and fault tolerance are very difficult to achieve if at all**
Byteflight:	**passive safety (for fail-safe systems) possible with intelligent star coupler (application responsibility)**
Flexray:	**mixed-channel architecture intended for electronic replacement of current two-way hydraulic braking systems, no concepts for other safety related applications known**
TTP/C:	**time-triggered architecture (TTA) systematically deals with fault tolerant units and replica determinism, fail-silence, composability (reduces probability for design faults!)**

CAN – Byteflight – Flexray – TTP/C

Availability and Support:

CAN:	**world standard for automotive electronics**
Byteflight:	**components available, development tools unknown/not widely available, used by BMW.** www.byteflight.com
Flexray:	**under development, supported by Flexray-Group (Daimler-Chrysler, BMW, Motorola, Philips)** www.flexray-group.com
TTP/C:	**components available, development tools available from TTTech, used by several automotive and aerospace development groups (e.g., Audi, Honeywell).** www.ttagroup.org, www.tttech.com