CIS 721 - Real-Time Systems

Lecture 19: Time-Triggered Protocols

Mitch Neilsen
neilsen@cis.ksu.edu

* some slides from H. Kopetz, "Real-Time Systems: Design Principles for Distributed Embedded Applications", 2nd Ed., 2011

Outline

- **Real-Time Communication**
  - Controller Area Network
  - CAN Higher Layer Protocols -- CAN Kingdom, J1939, etc.
  - **Time-Triggered Protocols**
    - **State and Event Information**
    - **Why Time-Triggered Communication?**
    - **Examples of Time-Triggered Protocols**
    - **Integration of Event-Triggered and Time-Triggered Services**
- Next Time: FreeRTOS

Safety Critical Applications

- An ***embedded system*** is part of a larger system that performs a safety-critical service.
- Failure of the system can cause harm to human life or extensive financial loss.
- In most cases, tight interaction with the environment: real-time response of the embedded system required.
- System must perform predictably, even in the case of a failure of individual components within the system.
- No single point of failure implies the need for a distributed computer architecture.

Essential Characteristics of RT Systems

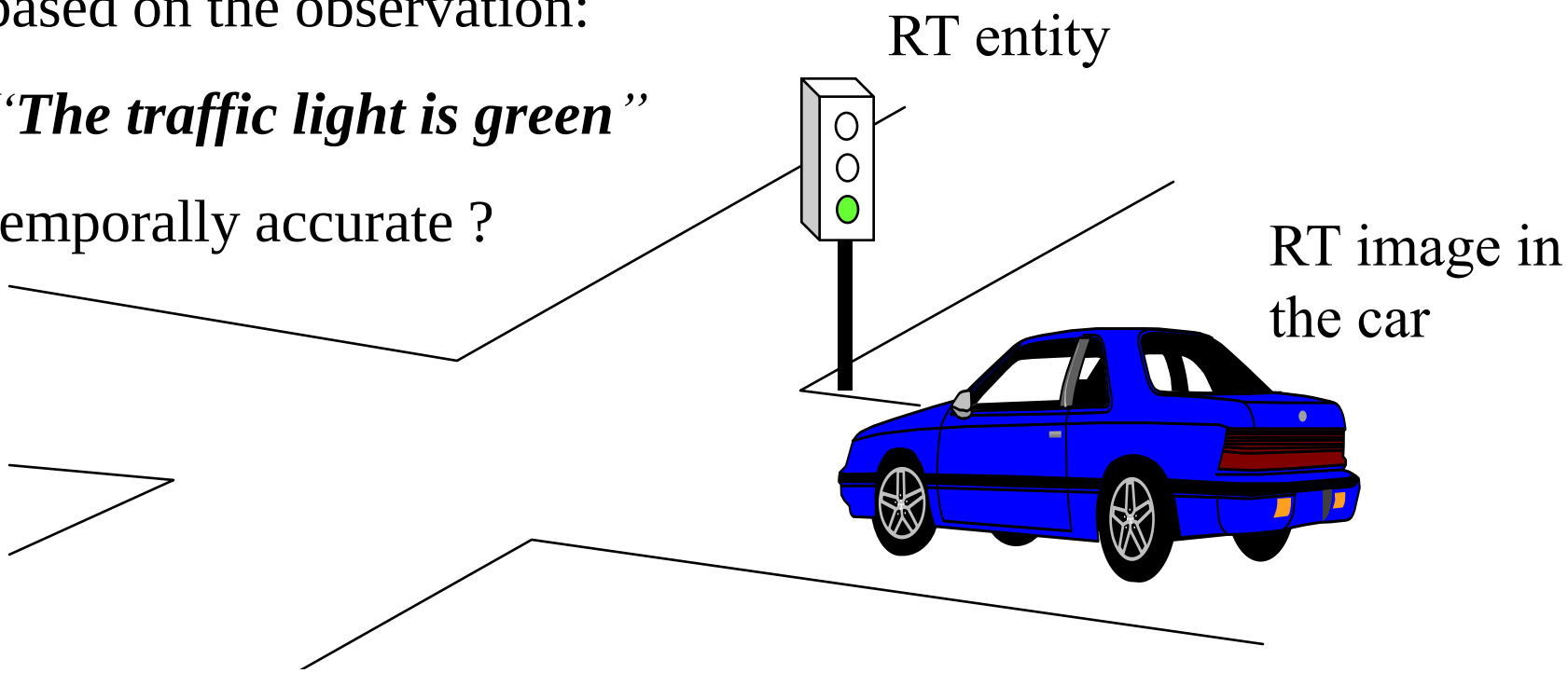
- **Physical time is a first order concept**: There is only one physical time in the world and it makes a lot of sense to provide access to this physical time in all nodes of a distributed real-time system.
- **Time-bounded value of real-time data:** The validity of real-time data is invalidated by the progression of real-time.
- **Existence of deadlines:** A real-time task must produce results before the deadline--a known instant on the timeline--is reached.
- **Inherent distribution:** Smart sensors and actuators are nodes of a distributed real-time computer system.
- **High dependability:** Many real-time systems must continue to operate even after a component has failed.

Temporal Accuracy of Real-Time Information

How long is the RT image,
based on the observation:

*"**The traffic light is green**"*

temporally accurate ?

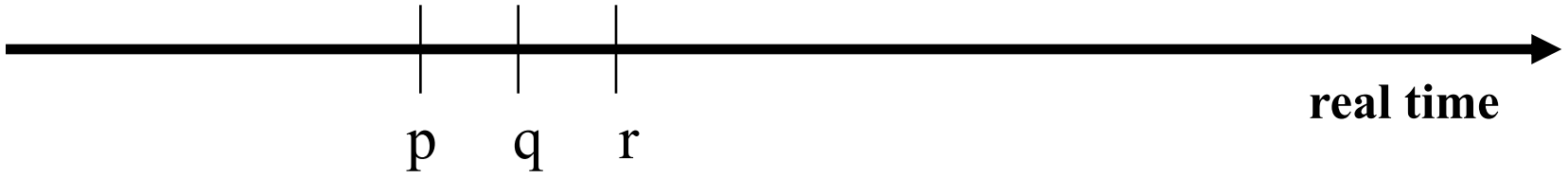


If the correct value is used at the wrong time,
it's just as bad as the opposite.

Model of Time (Newton)--Temporal Order

- The continuum of real time can be modeled by a directed timeline consisting of an infinite set {T} of ***instants*** with the following properties:
  - {T} is an **ordered set**, i.e., if p and q are any two instants, then either (1) p is simultaneous with q or (2) p precedes q or (3) q precedes p, and these relations are mutually exclusive.
  - {T} is a **dense set**. This means that, if p ≠ r, there is at least one q between p and r.

The order of instants on the timeline is called the ***temporal order***.



Durations and Events

- A section (interval) of the time line is called a ***duration***.
- An ***event*** is a happening at an instant of time.
- An event does not have a duration. If two events occur at an identical instant, then the two events are said to occur ***simultaneously***.
- Instants are totally ordered; however, ***events are only partially ordered***, since simultaneous events are not in the order relation.

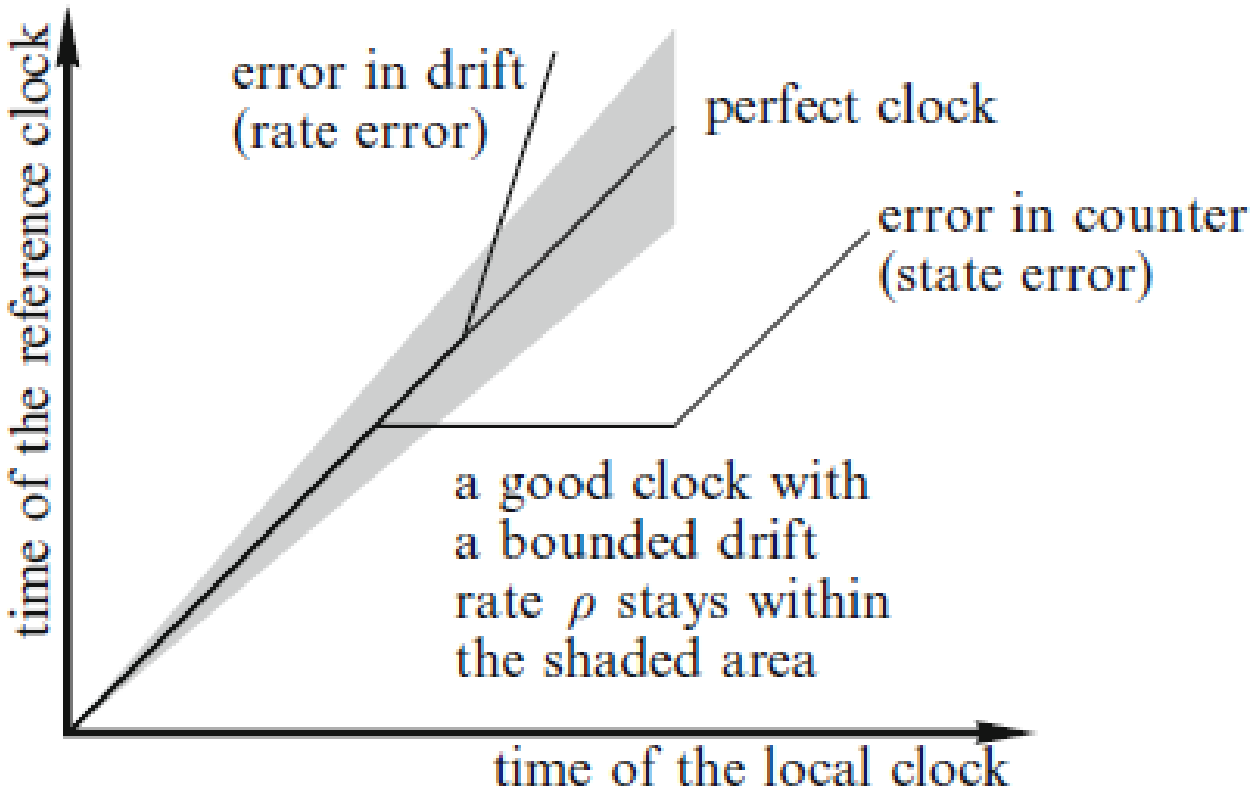
Causal Order

- Reichenbach [Rei57,p.145] defined ***causality*** by a mark method without reference to time: "If event e1 is a cause of event e2, then a small variation (a mark) in e1 is associated with small variation in e2, whereas small variations in e2 are not necessarily associated with small variations in e1."
- **Example**: Suppose there are two events e1 and e2:
  - e1 = Somebody enters a room.
  - e2 = The telephone starts to ring in a room.
- Consider the following two cases
  - (i) e2 occurs after e1 **- not causally related**
  - (ii) e1 occurs after e2 **- probably causally related**

Clocks

- A **(digital physical) clock** is a device for measuring time. It contains a counter and a physical oscillation mechanism that periodically generates an event to increase the counter. The periodic event is called the microtick of the clock.
- Granularity: The duration between two consecutive microticks of a digital physical clock is called a ***granule*** of the clock.

Clock Drift

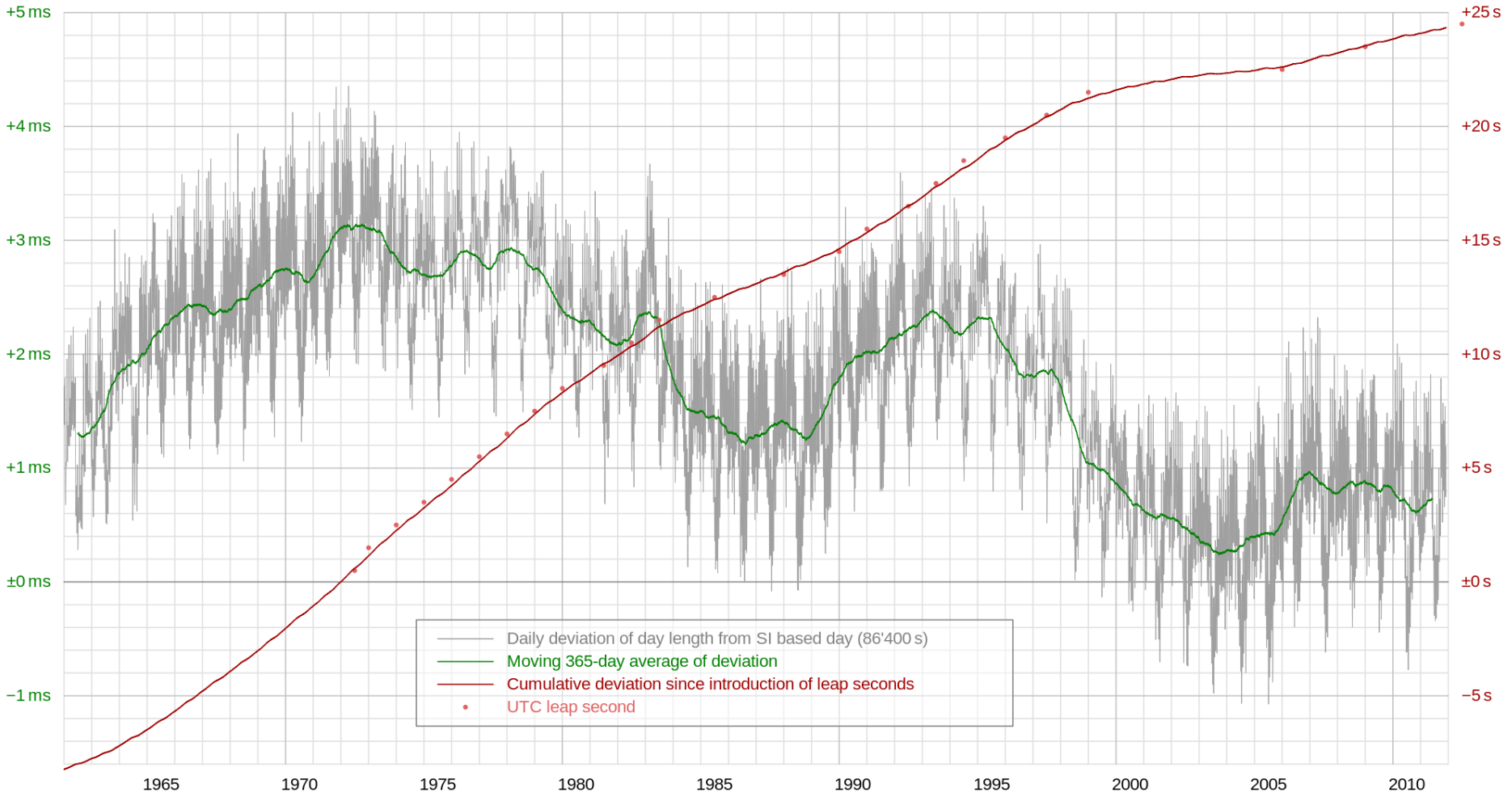

error in drift
(rate error)
perfect clock
error in counter
(state error)
a good clock with
a bounded drift
rate ρ stays within
the shaded area
time of the reference clock
time of the local clock

Time Standards

International Atomic Time (TAI – Temps Atomique Internationale): The need for a time standard that can be generated in a laboratory gave birth to the International Atomic Time (TAI). TAI defines the second as the duration of 9,192,631,770 periods of the radiation of a specified transition of the cesium atom 133. The intention was to define the duration of the TAI second so that it agrees with the second derived from astronomical observations. TAI is a **chronoscopic** timescale, i.e., a timescale without any discontinuities (e.g., leap seconds). The epoch of TAI starts on January 1, 1958 at 00:00 h Greenwich Mean Time (GMT). The time base of the global positioning system GPS is based on TAI with the epoch starting on January 6, 1980 at 00:00 h.

Universal Time Coordinated (UTC): UTC is a time standard that has been derived from astronomical observations of the rotation of the earth relative to the sun. It is the basis for the time on the wall-clock. The UTC time standard was introduced in 1972, replacing the Greenwich Mean Time (GMT) as an international time standard.

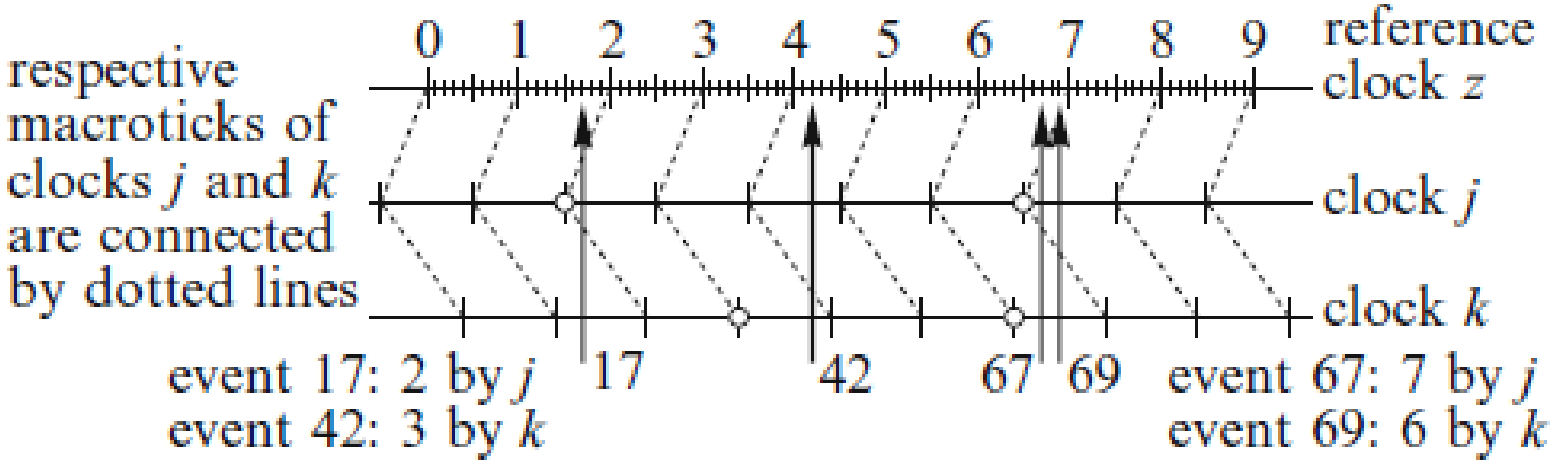
Deviation in Day Length [wikipedia]



Leap Seconds

Because the rotation of the earth is not smooth, but slightly irregular, the duration of the GMT second changes slightly over time. In 1972, it was internationally agreed that the duration of the second should conform to the TAI standard, and that the number of seconds in an hour would have to be modified occasionally by inserting a leap second into the UTC to maintain synchrony between the UTC (wall-clock time) and astronomical phenomena, like day and night. Because of this leap second, the UTC is **not a chronoscopic** timescale, i.e., it is not free of discontinuities. It was agreed that on January 1, 1958 at midnight, both the UTC and the TAI had the same value. Currently, UTC deviates from TAI by about 34 s. The point in time when a leap second is inserted into the UTC is determined by the Bureau International de l'Heure and publicly announced, so that the current offset between the UTC and the TAI is always known.

Temporal Order



Although the duration between event 17 and event 42 is 25 microticks, and the duration between event 67 and event 69 is only 2 microticks, both durations lead to the same measured difference of one granule. The global time-stamp for event 69 is smaller than the global timestamp for event 67, although event 69 occurred after event 67.

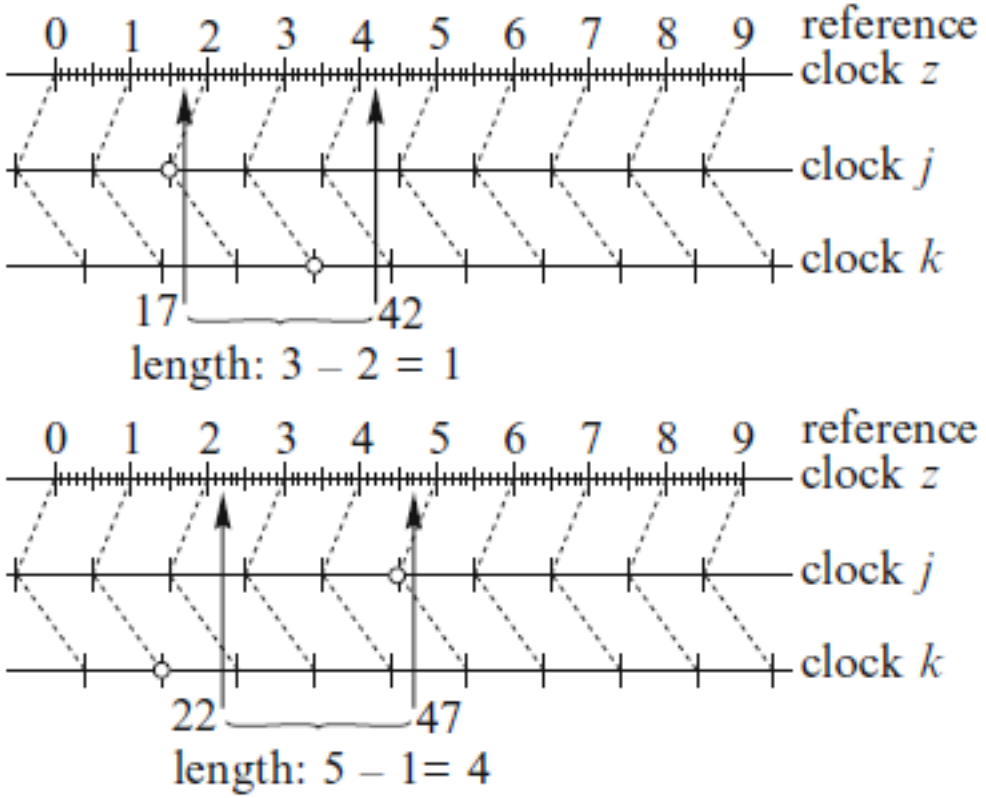
Because of the accumulation of the synchronization error and the digitalization error, it is not possible to reconstruct the temporal order of two events from the knowledge that the global time-stamps differ by one tick.

Faithfulness in Time Measurement

This fundamental limitation in time measurement limits the ***faithfulness*** of the digital computer model of a controlled physical subsystem. The time-base in the physical part of a cyber-physical system is dense, while the time base in the computer system is discrete.

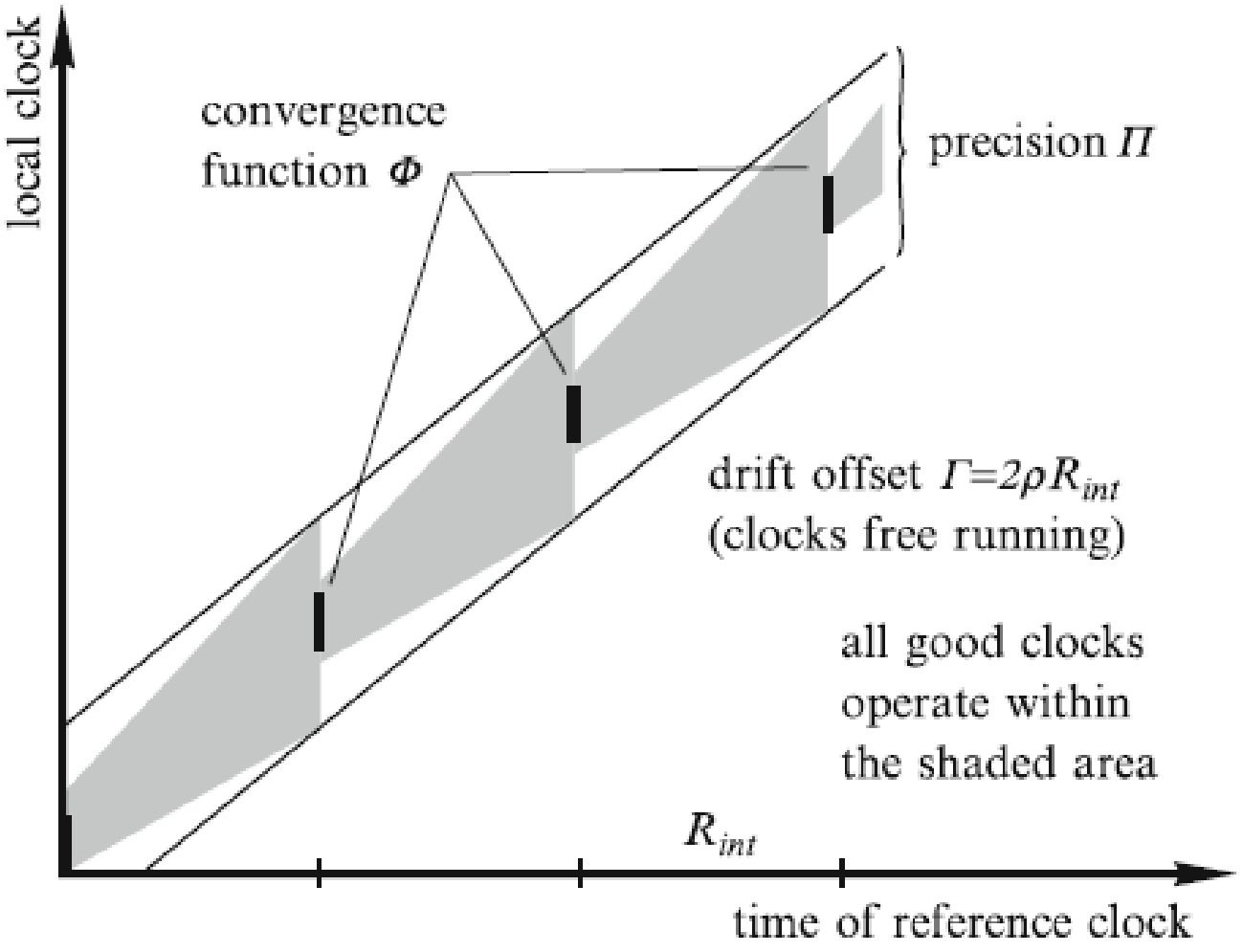
Whenever two events in the physical subsystem occur close together, compared to the granularity of the global time, it is not possible to reconstruct the physical temporal order of the events in the computer system faithfully. The only way out of this dilemma is the provision of **a global time base with a smaller granularity**, such that temporal errors are reduced [Kop09].

Errors in Interval Measurement



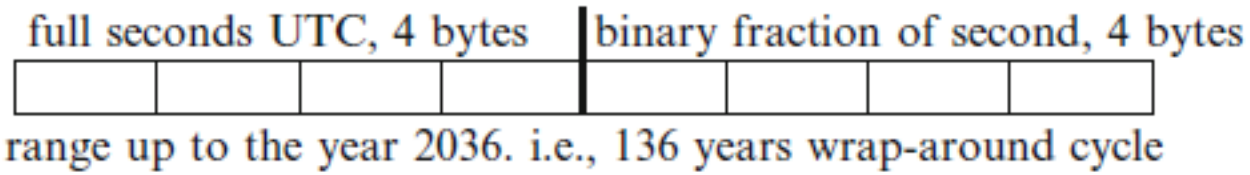
$$(d_{obs} - 2g) < d_{true} < (d_{obs} + 2g)$$

Clock Synchronization Condition



Clock Synchronization

- Internal Clock Synchronization
  - Example: CAN nodes synchronize with each other
- External Clock Synchronization
  - Example: Clocks are synchronized with an external clock base; e.g., using Network Time Protocol (NTP)



NTP Time Format

Real Time (RT) Entity

- A Real-Time (RT) Entity is a state variable of interest for the given purpose that changes its state as a function of real-time.
- We distinguish between:
  - Continuous RT Entities
  - Discrete RT Entities
- Examples of RT Entities:
  - Flow in a Pipe (Continuous)
  - Position of a Switch (Discrete)
  - Setpoint selected by an Operator
  - Intended Position of an Actuator

Observation

- Information about the state of a RT-entity at a particular point in time is captured in the concept of an **observation**.
- An *observation* is an atomic triple

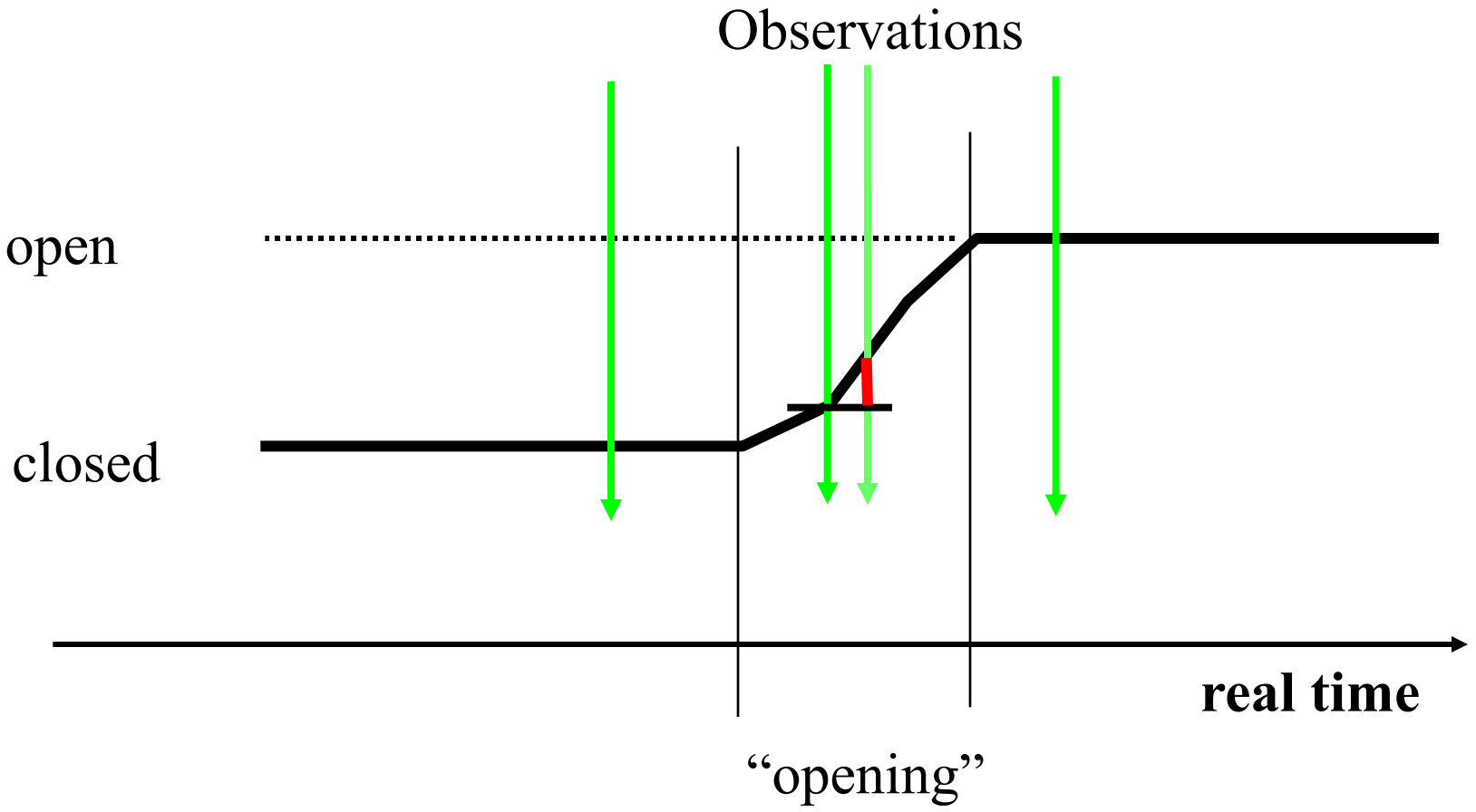
  Observation = <Name, Time, Value>

  consisting of:
  - The name of the RT-entity
  - The point in real-time when the observation has been made
  - The values of the RT-entity

Observations are transported in messages:

If the time of message arrival is taken as the time of observation, then delaying a message changes the contained observation.

Observation of a Valve



State and Event Observation

- An observation is a ***state observation***, if the value of the observation contains the full or partial state of the RT-entity. The time of a state observation denotes the point in time when the RT-entity was sampled.

- An observation is an ***event observation***, if the value of the observation contains the difference between the "old state" (the last observed state) and the "new state". The time of the event information denotes the point in time of observation of the "new state".

What is the Difference?

	State	Event
Time of Observation	periodic	after event occurrence
Trigger of Observation	**Time**	**Event**
Content	Full state	Difference (new - old)
Required Semantics	**at-least once**	**exactly once**
Loss of observation	short blackout	loss of state synchronization
Idempotency	yes	no

Event Triggered (ET) vs. Time Triggered (TT)

- A Real-Time system is ***Event Triggered*** **(ET)** if the control signals are derived solely from the occurrence of events, e.g.,
  - termination of a task
  - reception of a message
  - an external interrupt
- A Real-Time system is ***Time Triggered*** **(TT)** if the control signals, such as
  - sending and receiving of messages
  - recognition of an external state change

  are derived solely from the progression of a (global) notion of time.

Time-Triggered vs. Event-Triggered

- Time-triggered control system
  - All activities are carried out at certain points in time know a priori.
  - All nodes have a common notion of time, based on approximately synchronized clocks.
- Event-triggered control system
  - All activities are carried out in response to relevant events external to the system.

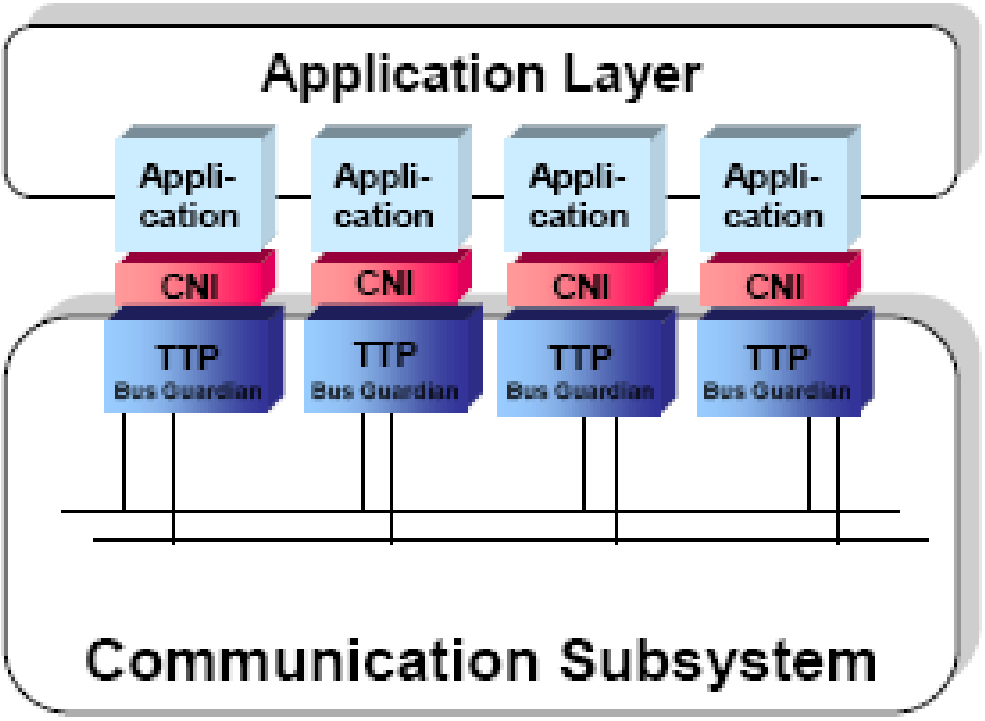
Event-Triggered Approach

- Event-triggered approach
  - Model using timed automata
  - Example: CAN (Controller Area Network)
  - Example: Meeting of 3 people
    - Everyone speaks whenever he/she has something to say.
    - Must wait for the currently speaker to finish before a new speaker can start.
    - Imagine a meeting of 200 people!

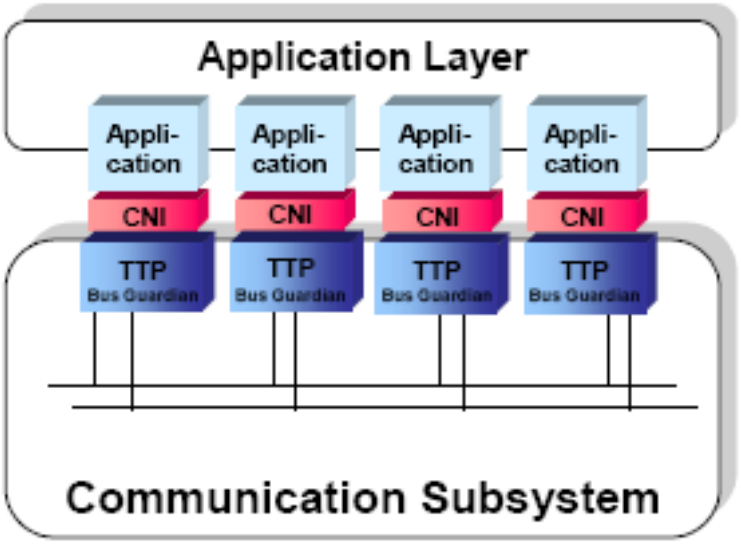
Time-Triggered Approach

- Example: Meeting of 3 people
  - Every speaker is assigned a predetermined time slot.
  - After one round, the speaker gets a slot again.
  - Also, a topic-schedule is worked out in advance.
    - Topic1, Topic2, Topic4 in the first round.
    - Topic1, Topic3 and Topic5 in the second round
    - Topic2, Topic4 and Topic5 in the third round.
  - The **protocol** ensures that no one breaks the rules!

Time-Triggered Architecture

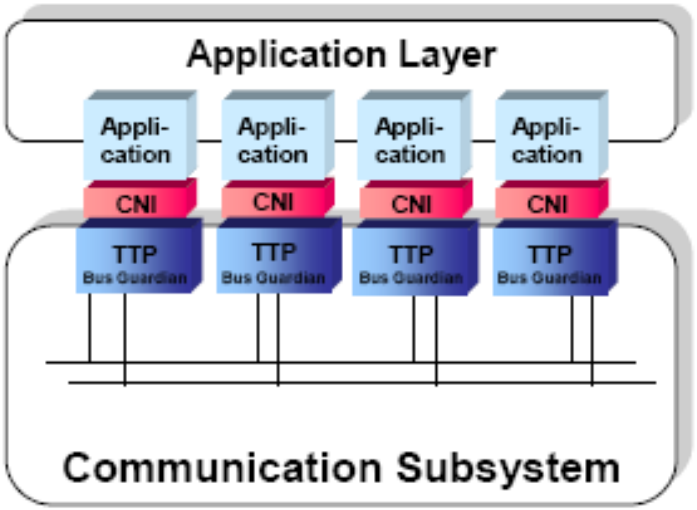


Time-Triggered Architecture



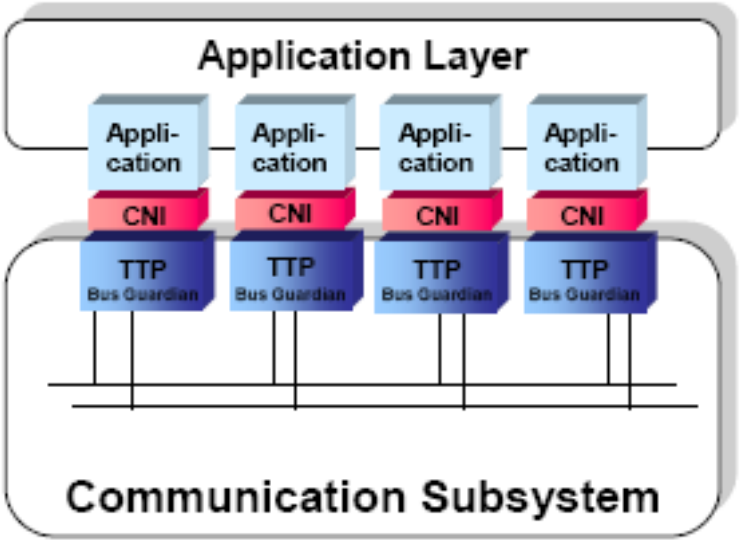
- Basic unit: NODE
- Node:
  - A processor with memory
  - I/O subsystem
  - Operating system
  - Application software
  - Time-triggered communication controller

Time-Triggered Architecture



- **Communication (TTA Protocol)**
  - **Nodes connect to each other via two independent channels.**
  - **The communication subsystem executes a periodic Time Division Multiple Access (TDMA) schedule.**
  - **Read a data frame + state information from one CNI (Communication Node Interface) at predetermined fetch instant and deliver to the CNIs of all receiving nodes at predetermined delivery instants.**

Time-Triggered Architecture



- **Communication**
  - **All the TTPs in a cluster know this schedule.**
  - **All nodes of a cluster have the "same" notion of global time.**
  - **Fault-tolerant clock synchronization.**
  - **TTA BUS topology.**

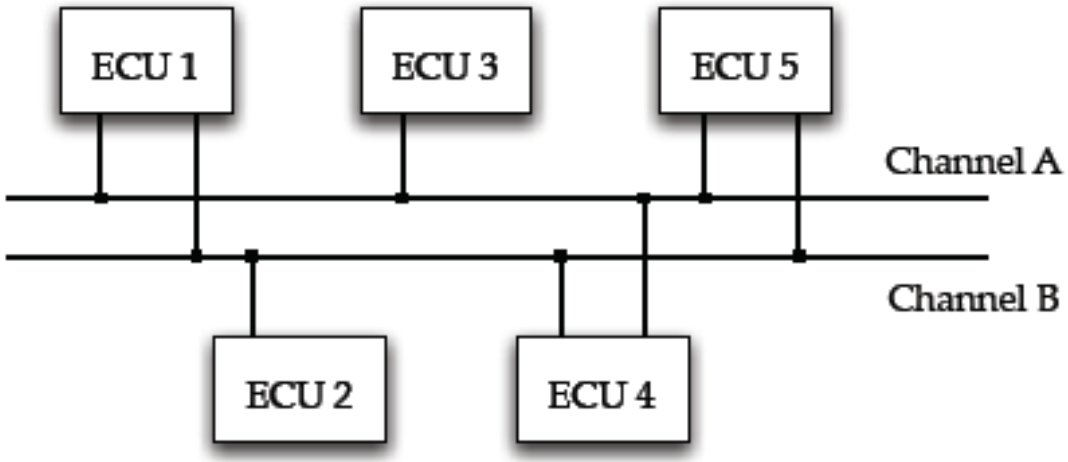
Features of the TTP

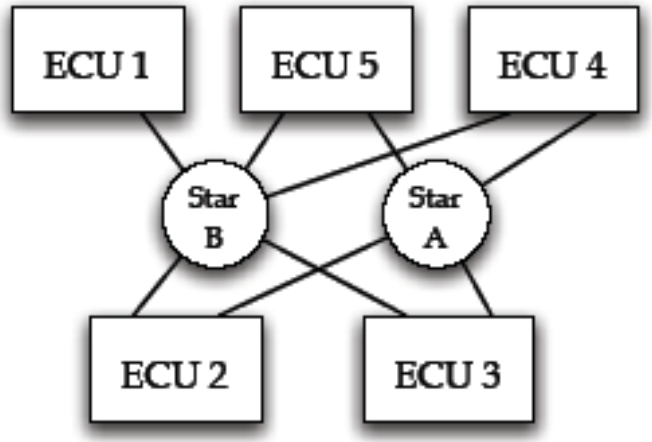
- Fault-tolerance
- Small overhead
- ***Only data signals (and no control signals) cross interfaces***.
- TTP integrates numerous services
  - Predictable message transmission
  - Message acknowledgement in group communication
  - Clock synchronization
  - Group membership

Assumptions

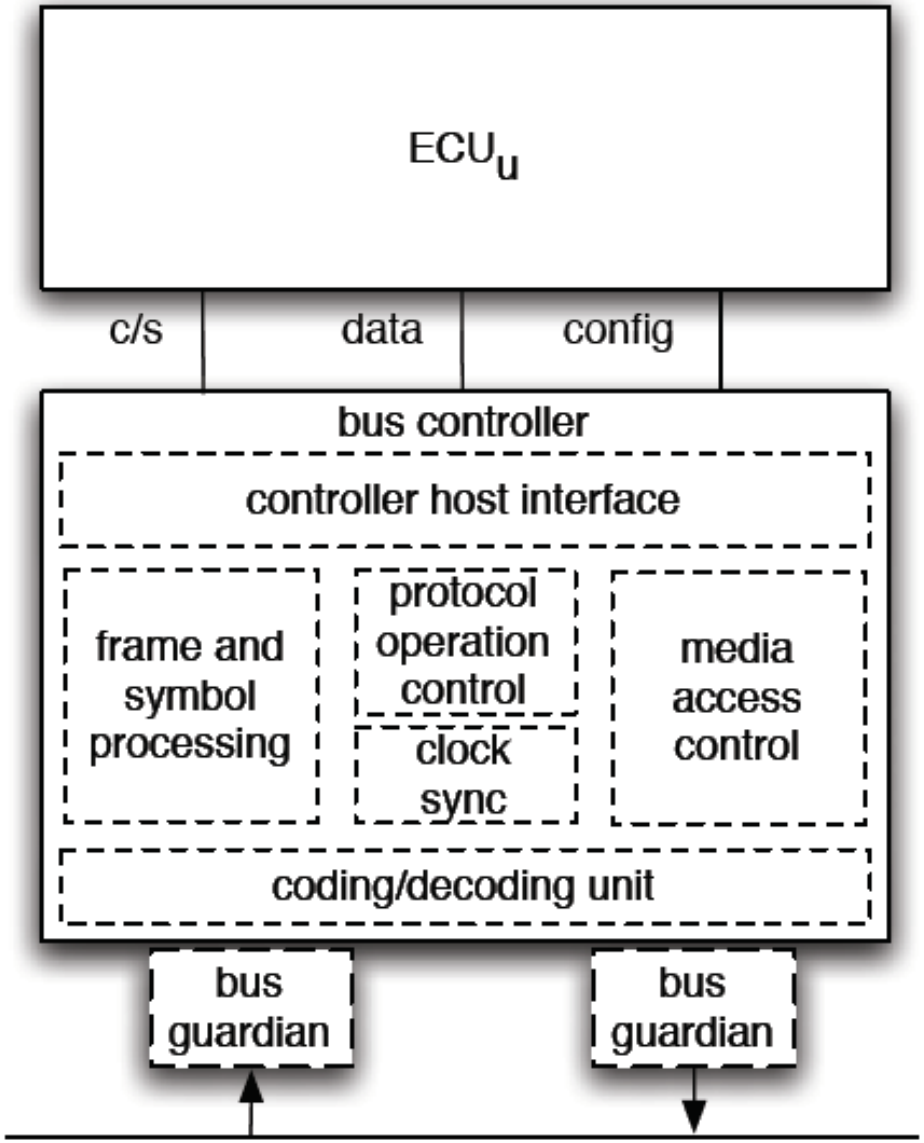
- Fail-silence
  - Communication channels only have omission failures.
  - Nodes either deliver correct results or no results
    - Internal failures are detected and node turned off

Network Topologies

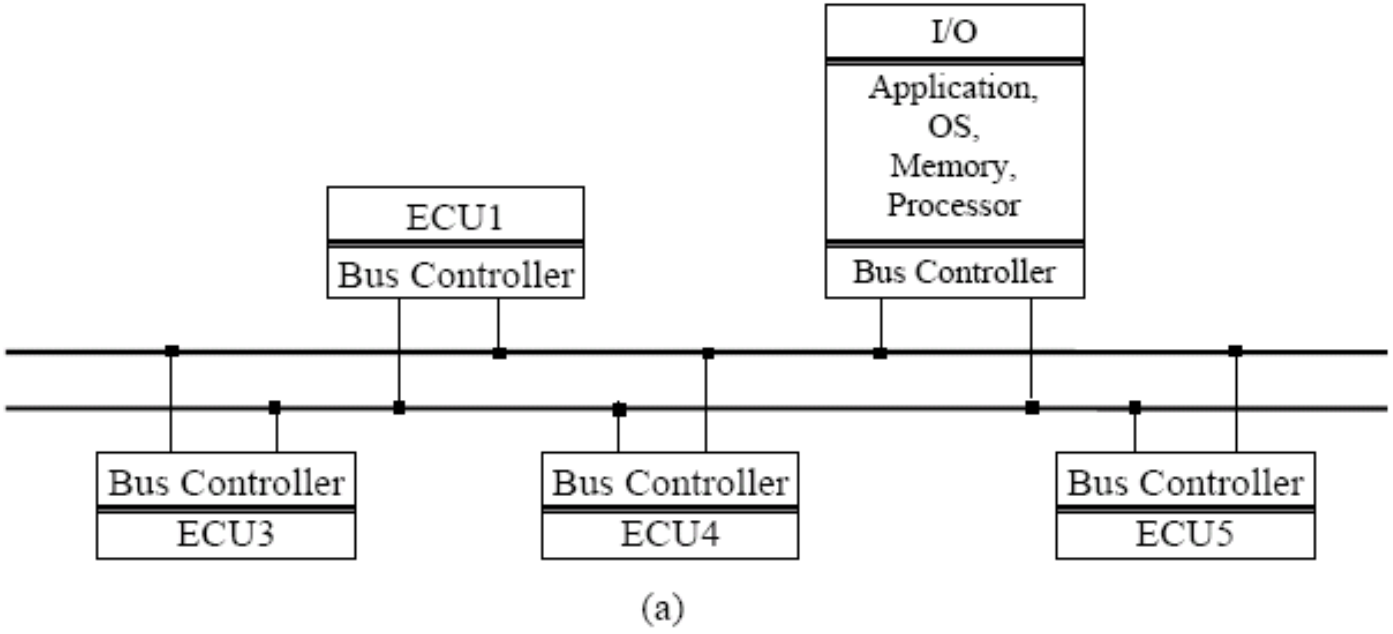




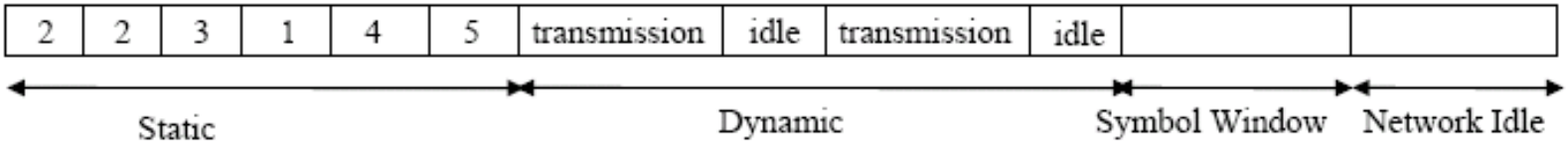
ECU + The Bus Controller



The TDMA Schedule (FlexRay)



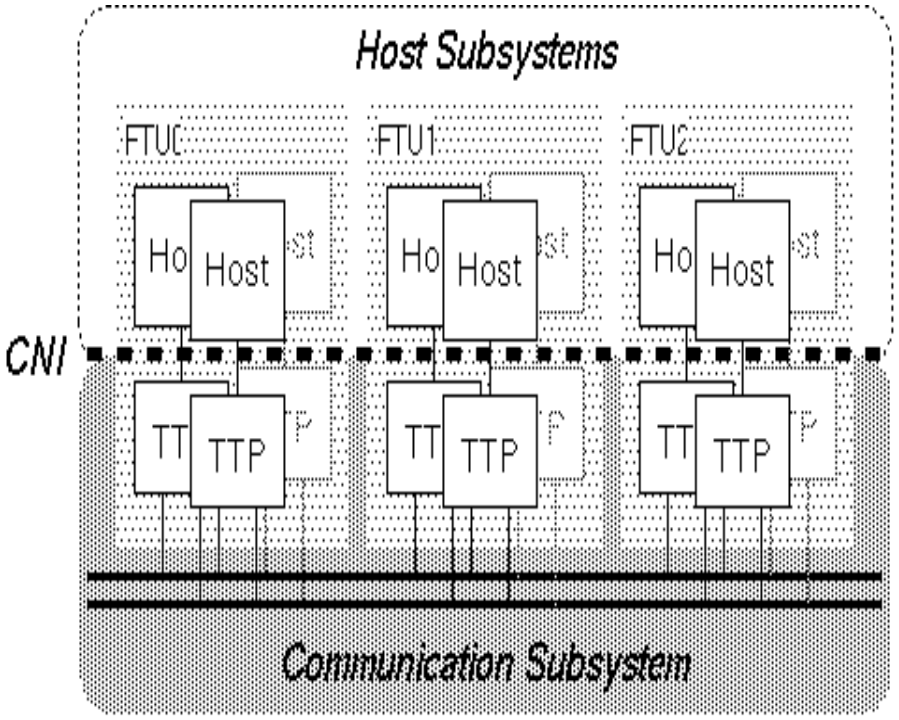
(a)



(b)

System Overview

- **Replicated communication channels**
- **The channel is a broadcast bus**
- **Access is by TDMA driven by progression of *global time***
- **Local nodes time synchronized by TTP**
- **Communication by rapid and periodic message exchanges**



CNI: Communication Network Interface

TTP: TTP Communication Controller

FTU: Fault Tolerant Unit

TTP Design Rationale

- Sparse time base
  - Messages are sent only at statically designated intervals
  - Inflexible compared to Event-Triggered (ET) model, but easier to test
- Use of ***a priori*** knowledge
  - All nodes are aware of when each node is scheduled to transmit
  - Sender node information need not be included in frame
  - Reduced overhead
- Broadcast
  - Correctness of transmitted message can be concluded as soon as one receiver acknowledges message delivery (broadcast medium)

Protocol Highlights

- Bus access
  - A Fault-Tolerant Unit (FTU) will have one or two time slots depending on class of fault-tolerance
  - Number of slots in a TDMA round given to an FTU may also be different
- Membership Service
  - If a message from a sending node does not occur in designated interval, its membership is set to 0 in other nodes
  - Membership checked before transmission. A node is alive if
    - Its internal error detection mechanism has not indicated error, and
    - At least one of its transmitted frames has been correctly acknowledged.

Protocol Highlights

- Temporary blackout handling
  - Correlated failure of a number of nodes
  - Identified by sudden drop in membership
  - Nodes send I-messages and perform local emergency control
  - After membership has stabilized, mode changed to global emergency service

Protocol Highlights

- Temporal encapsulation of nodes
  - Communication bandwidth assigned statically
  - Time base is sparse- every input can be observed and reproduced exactly
- Testability
  - Easy to test the implementation in comparison to ET
  - Easy to simulate –finite number of execution scenarios
    - Uncontrolled interactions between nodes are prevented
    - Determinism: can replicate states of nodes

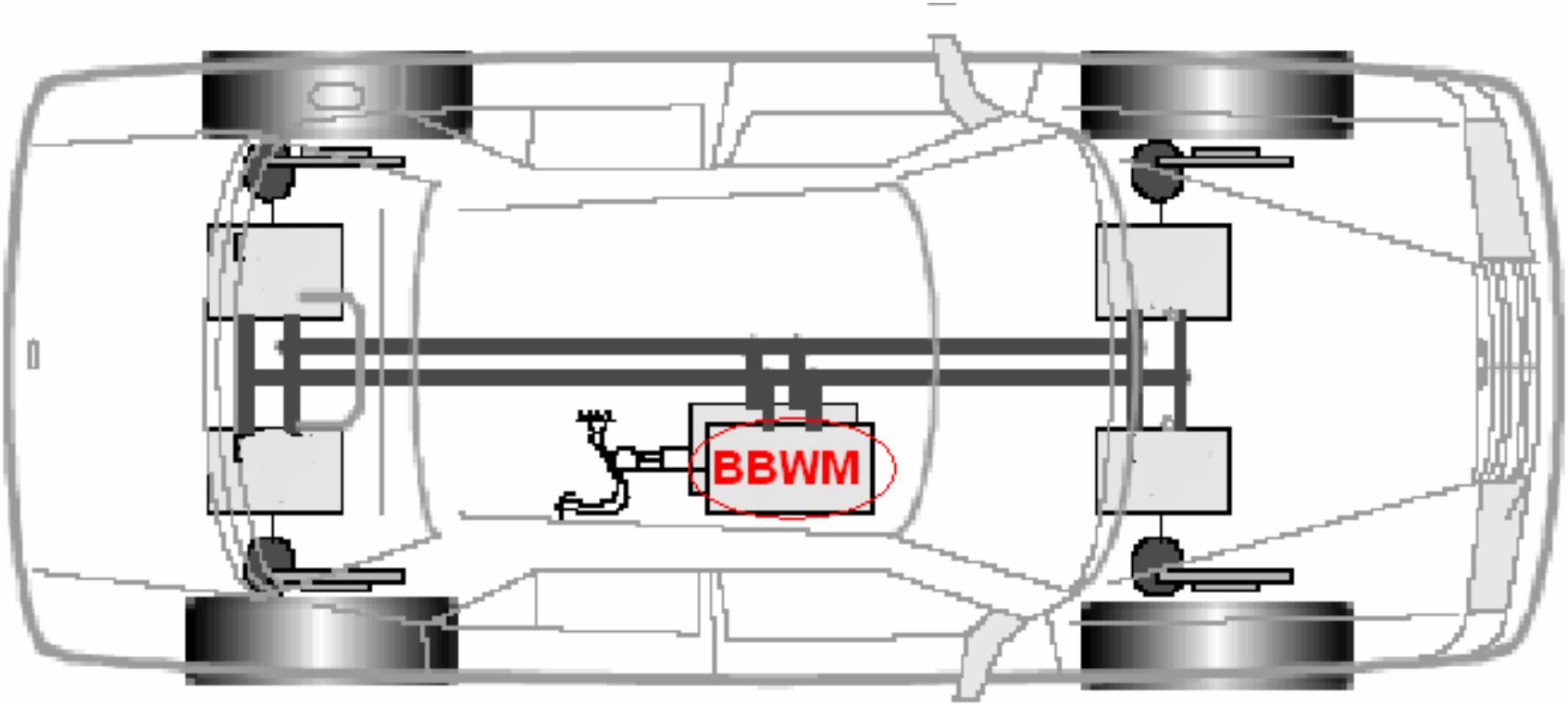
Strengths

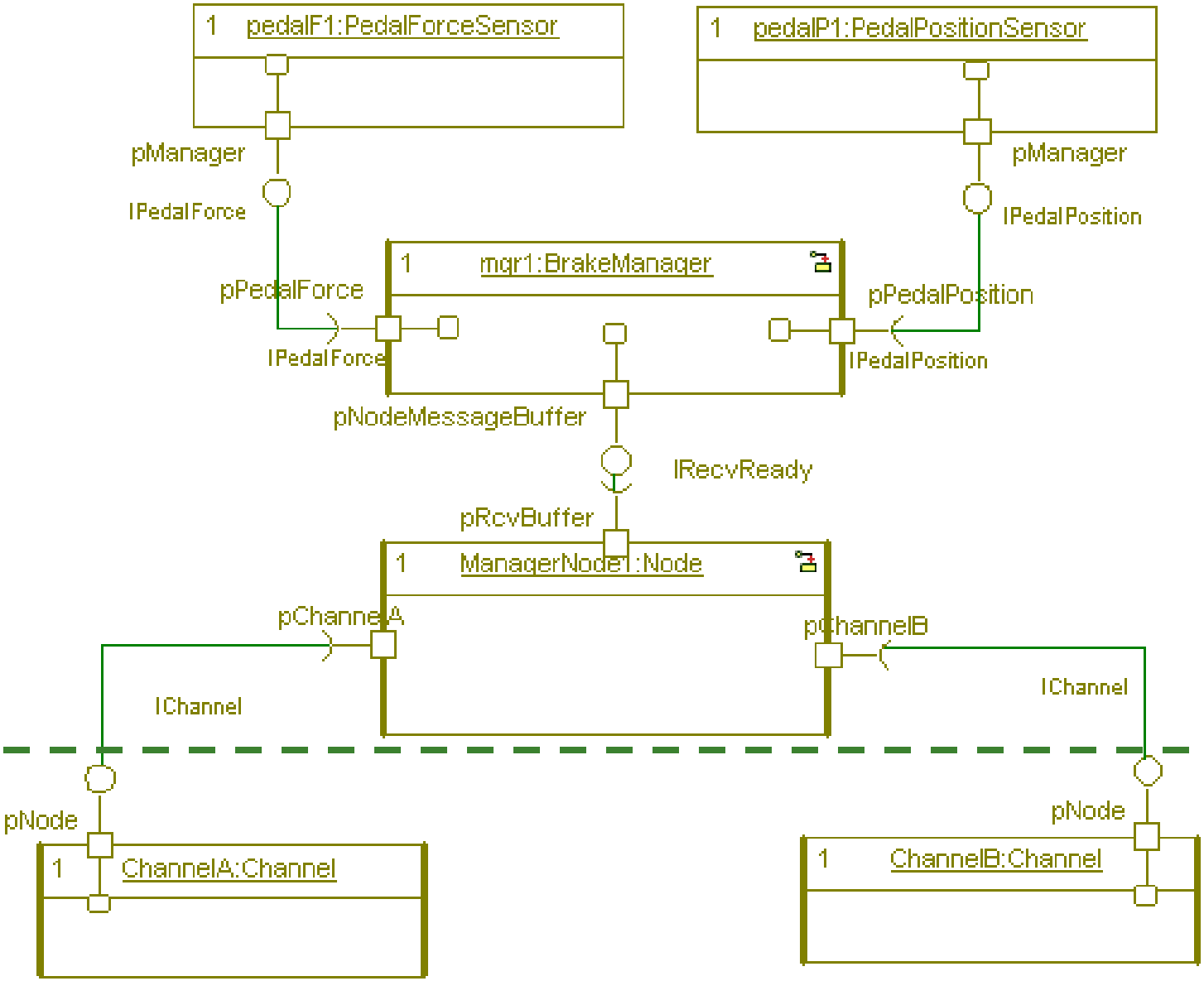
- Can provide fault-tolerant real-time performance
- Practical, efficient, and scalable
  - Can be implemented using available hardware, signalling mechanisms
  - Low overhead
  - High data rates, used in both twisted fiber and optical channels
- Reusability, composability, and testability

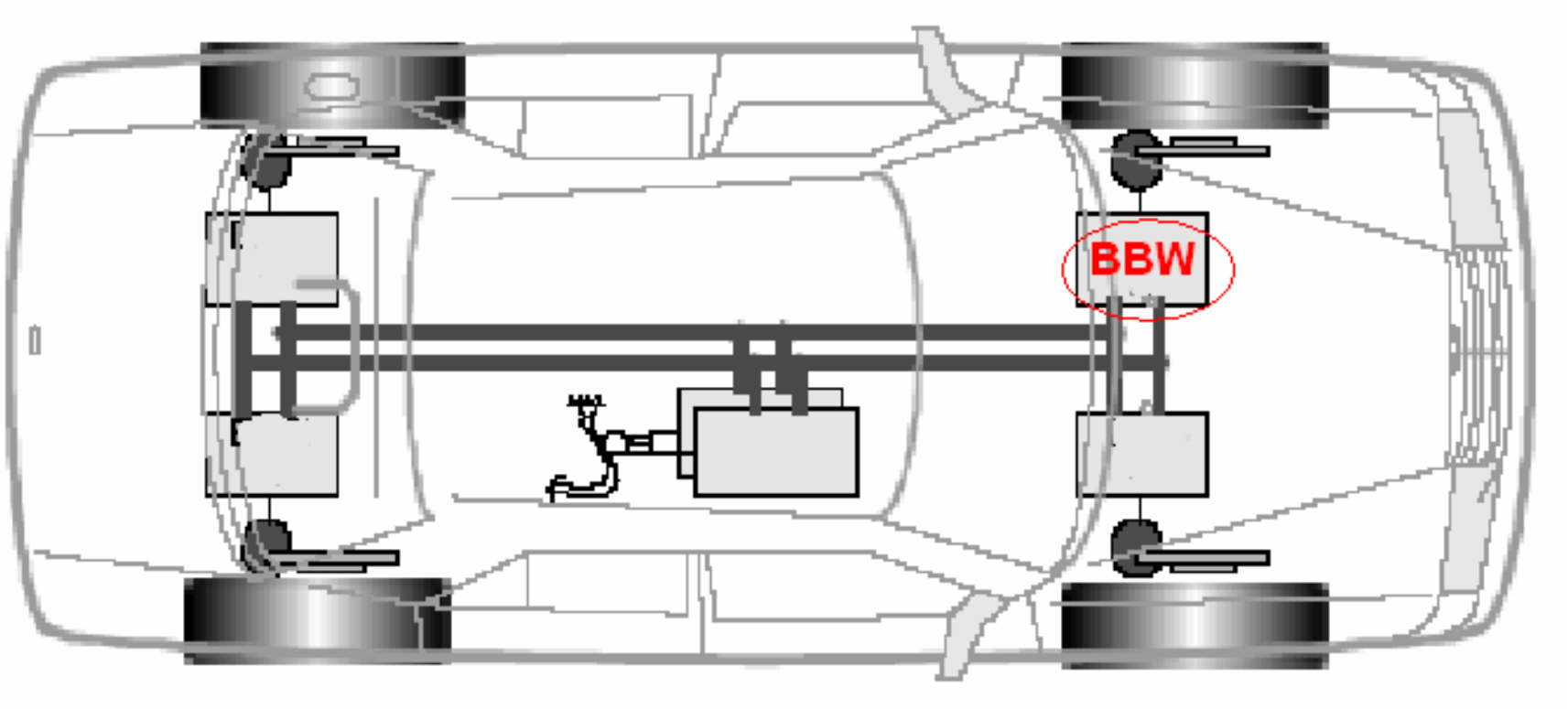
Weaknesses

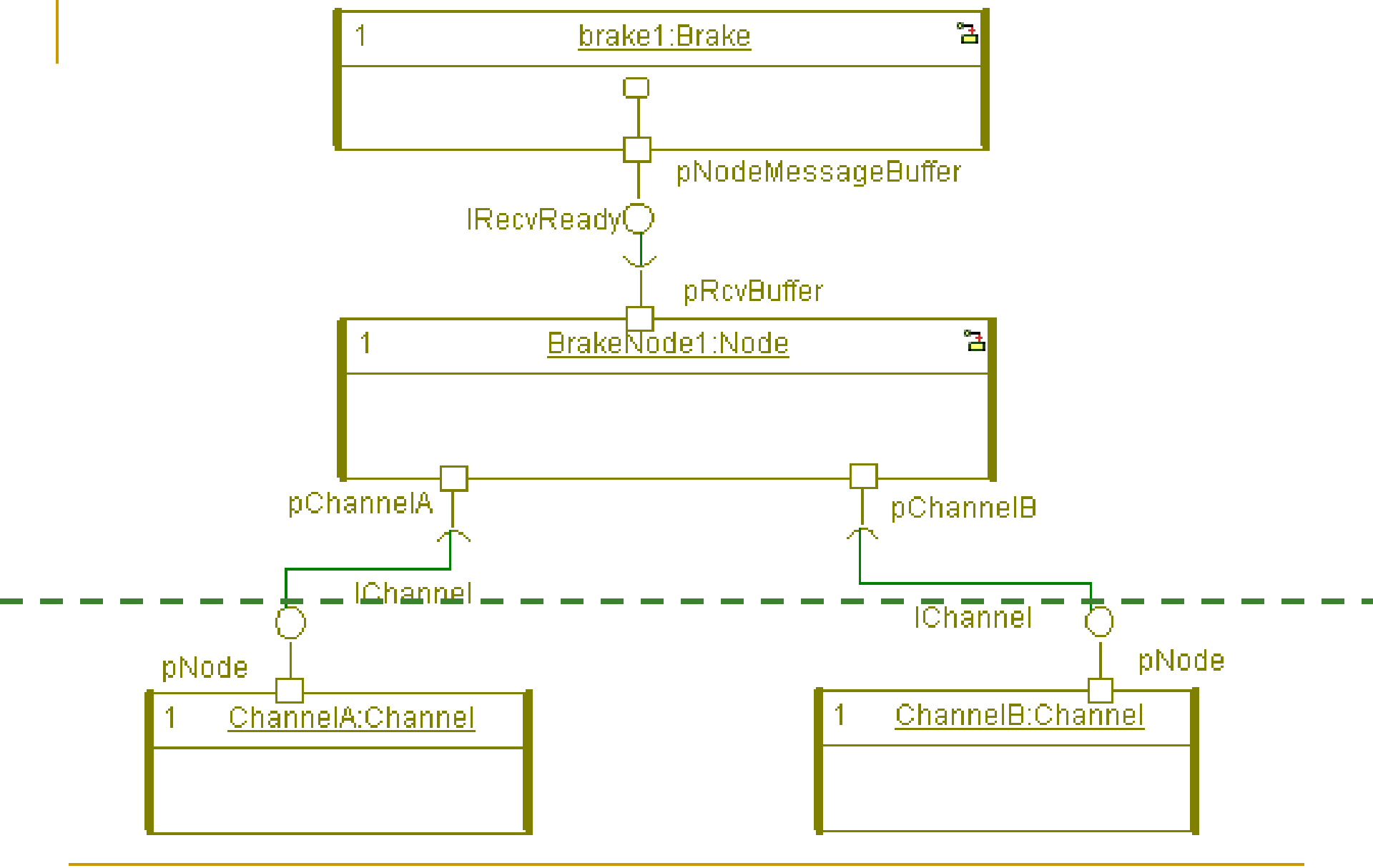
- The schedule is fixed so there is no bandwidth allocated for alarms and other spontaneous messages
- All fault-tolerance mechanism is implemented at system level, this means that very little “freedom” is left for application specific implementations
- Addition of nodes affects the existing system (although not the application)

Block diagram of Automobile Brakes




1 pedalF1:PedalForceSensor
1 pedalP1:PedalPositionSensor
pManager
pManager
IPedalForce
IPedalPosition
1 mgr1:BrakeManager
pPedalForce
pPedalPosition
IPedalForce
IPedalPosition
pNodeMessageBuffer
IRecvReady
pRcvBuffer
1 ManagerNode1:Node
pChannelA
pChannelB
IChannel
IChannel
pNode
pNode
1 ChannelA:Channel
1 ChannelB:Channel


BBW


1
brake1:Brake
pNodeMessageBuffer
IRecvReady
pRcvBuffer
1
BrakeNode1:Node
pChannelA
pChannelB
IChannel
IChannel
pNode
pNode
1
ChannelA:Channel
1
ChannelB:Channel

TTP (**T**ime-**T**riggered **P**rotocol)

TTP – more than just a protocol

- Network protocol
- Operating system scheduling philosophy
- Fault tolerant approach

Time-Triggered approach

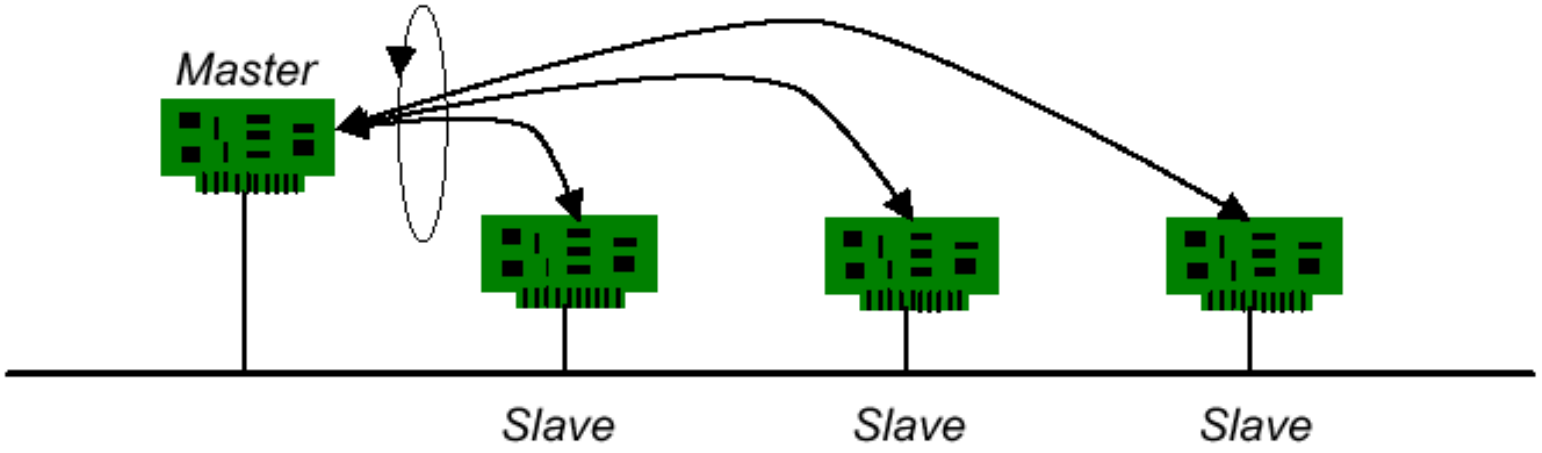
- Stable time base
- Simple
- Cyclic schedules

Two Versions

- TTP/A (Automotive Class A = soft real time)
  - A scaled-down version of TTP
  - A cheaper master/slave variant
- TTP/C (Automotive Class C = hard real time)
  - A full version of TTP
  - A fault-tolerant distributed variant

TTP/A - Polling

- Operation
  - Master polls the other nodes (slaves).
  - Slave nodes only transmit messages when they are polled.
  - Inter-slave communication through the master.



Polling Tradeoffs

- Advantage
  - Simple protocol to implement
  - Historically very popular
  - Bounded latency for real-time applications
- Disadvantage
  - Single point of failure from centralized master
  - Polling consumes bandwidth
  - Network size is fixed during installation (or master must discover nodes during reconfiguration)

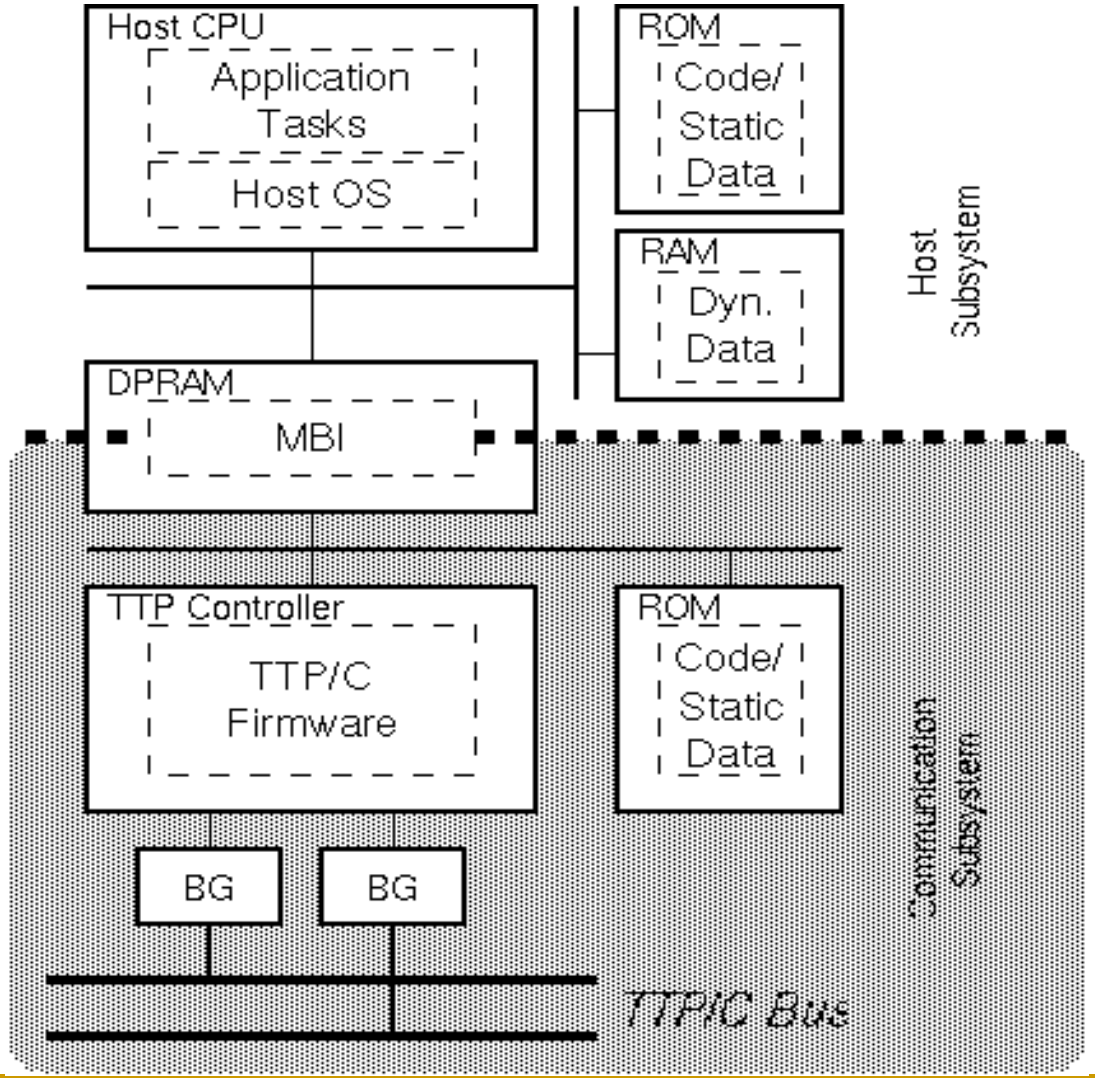
TTP/C

- TTP/C
  - A time-triggered communication protocol for safety-critical (fault-tolerant) distributed real-time control systems.
  - Based on a TDMA (Time Division Multiple Access) media access strategy.
  - Based on synchronized clocks.

Single Node Configuration

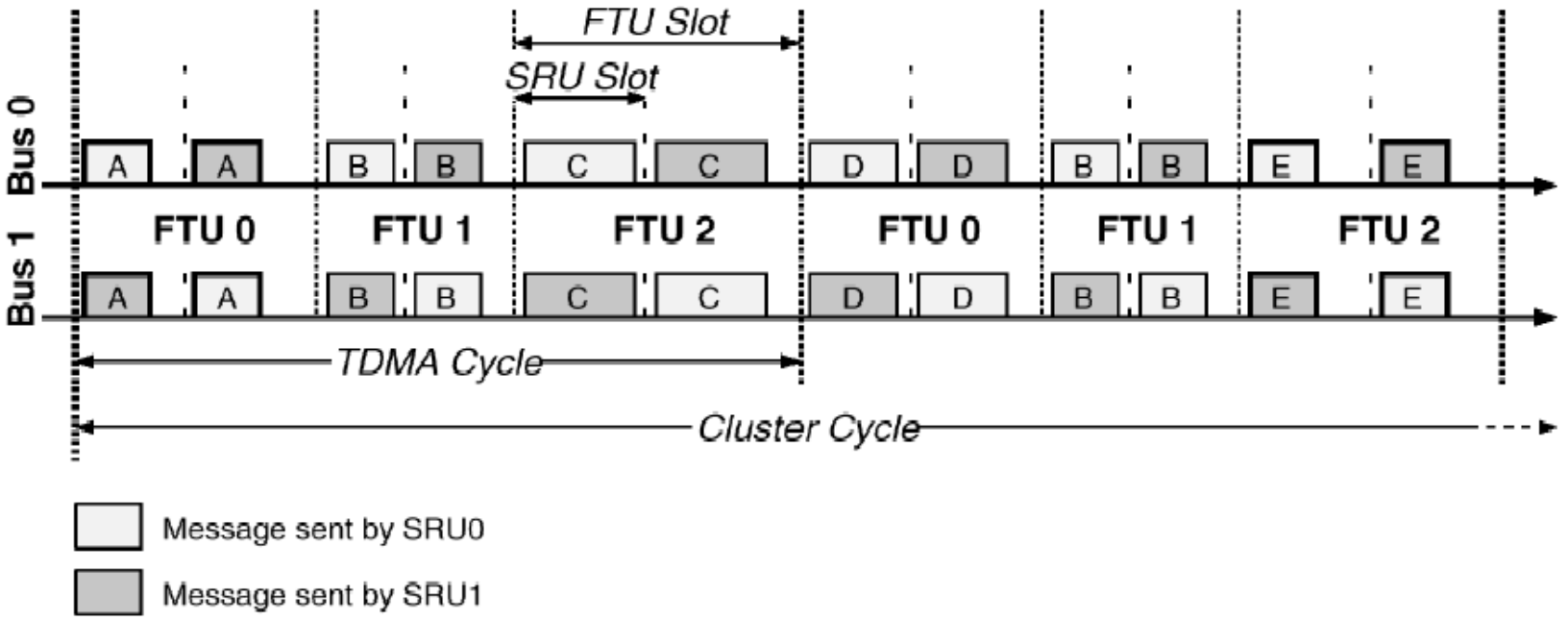
- Includes controller to run protocol
- DPRAM (dual ported RAM)
  - To implement memory-mapped network interface
- Bus Guard (BG)
  - Hardware watchdog to ensure “fail silent” operation
- Real chips must use highly accurate time sources
  - For reliability, even dual redundant crystal oscillators as used in DATAC for Boeing 777

TTP/C Architecture

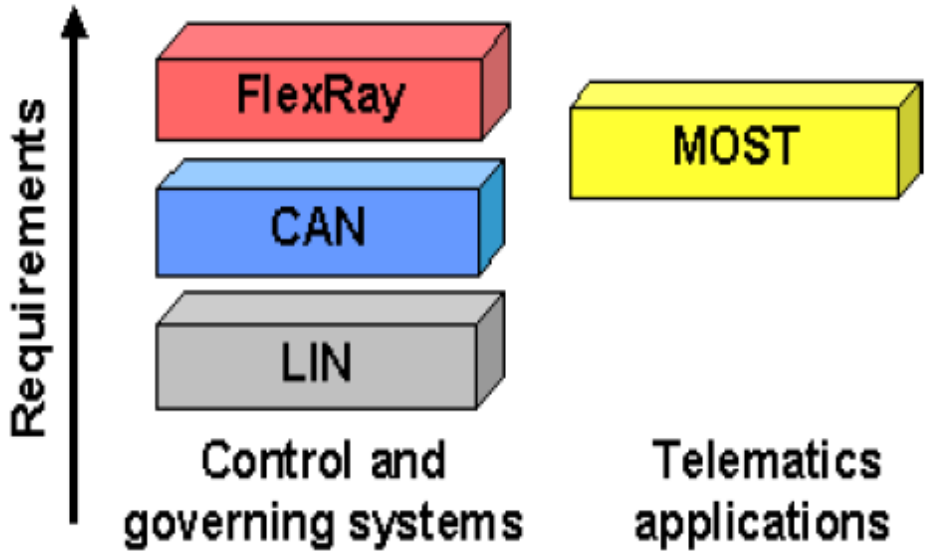


TTP/C - Cyclic Schedule

- TDMA Cycle
  - One unit sends results twice. Then, the next unit sends its results twice; and so on, until cycling back to the next message from the first unit.
- Cluster Cycle
  - A cluster cycle involves scheduling all possible messages and tasks.

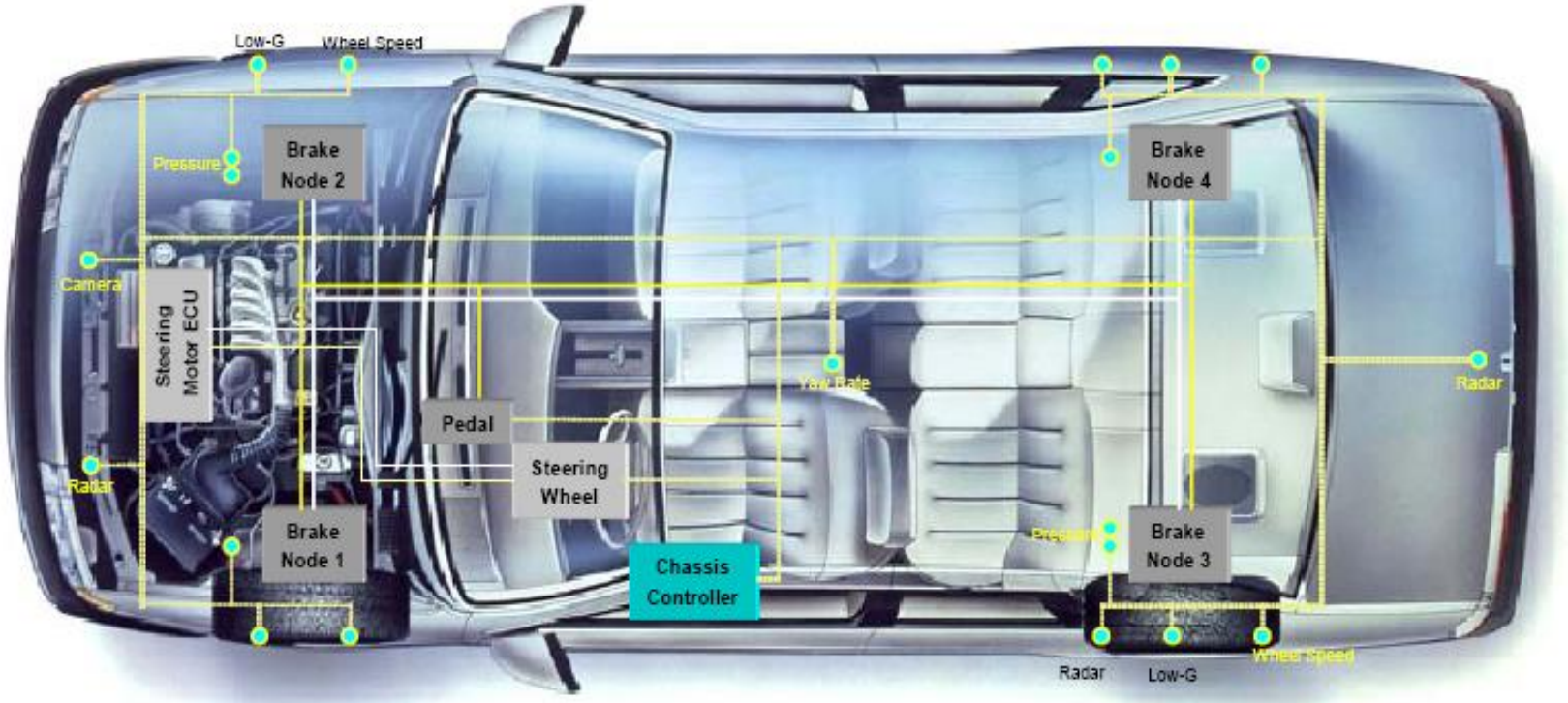


Major In-vehicle Networking Standards



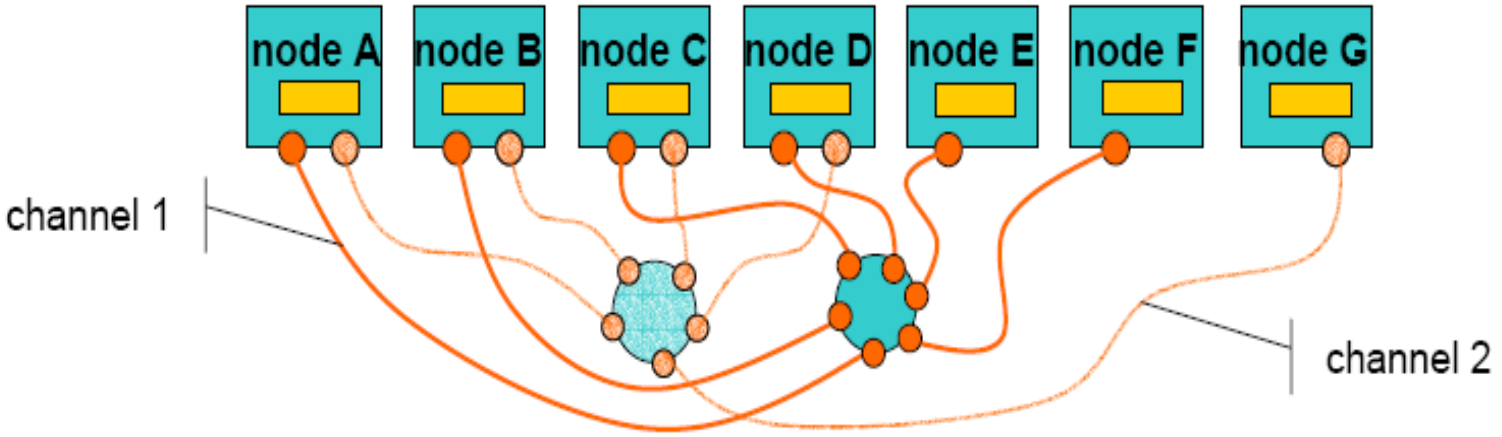
FlexRay is designed to meet key automotive requirements like dependability, availability, flexibility, and a high data rate to complement the major in-vehicle networking standards - CAN, LIN, and MOST.

What is FlexRay?



- FlexRay is a fault-tolerant, deterministic, time triggered communication protocol
- Intended to be a global standard for advanced automotive control
- Required to enable next generation, “by-wire” applications
- See www.flexray.com for more information

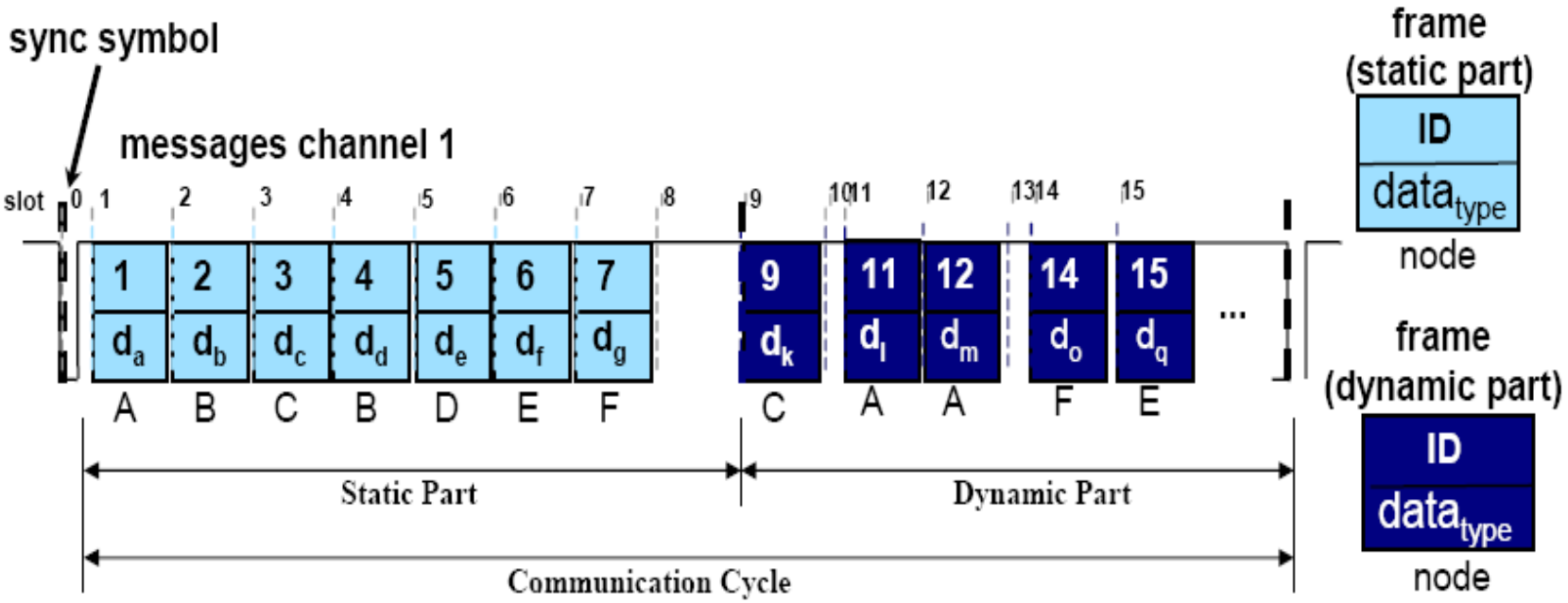
Integration Flexibility



FlexRay controller network in mixed connectivity, "active star" configuration

- **Multiple operating modes**
  - Single channel, dual (redundant) channel, or mixed connectivity
- **Supports bus, star, and multiple star topologies**
  - Active, passive, or mixed
- **Fault-tolerant and time triggered services implemented in hardware**
- **Can be used with optical and/or electrical physical layers**
- **Supports system integration at vehicle manufacturer level**

FlexRay Protocol Basics



- Data rate of 10 Mbit/sec per channel
- Scalable communication system, allowing synchronous and asynchronous data transmission
- Configurable to provide a mix of deterministic and dynamic data transmission
- Fast error detection and signalling
- Support of a fault tolerant synchronized global time base
- Error containment on the physical layer through an independent “Bus Guardian”
- Arbitration free transmission

Summary

- Time-Triggered architectures and protocols are becoming more important.
- ***Global time and clock synchronization play a fundamental role***.
  - But this also incurs overhead.
- The (TDMA) schedule is ***static***.
  - Can't do application specific optimizations.
- Used for time-deterministic, robust distributed systems.