Real-Time Communication

Prof. Dr. Jian-Jia Chen

LS 12, TU Dortmund

06 June 2017

Outline

Introduction

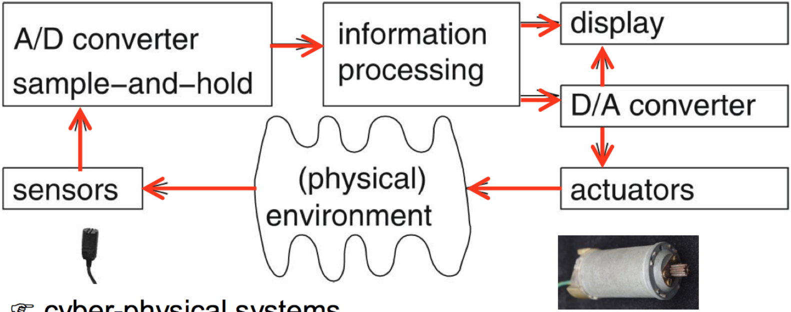
Analysis of TDMA

CAN (Controller Area Network)

Flexray

Communication

is everywhere ....



A/D converter

sample–and–hold

information processing

display

D/A converter

sensors

(physical) environment

actuators

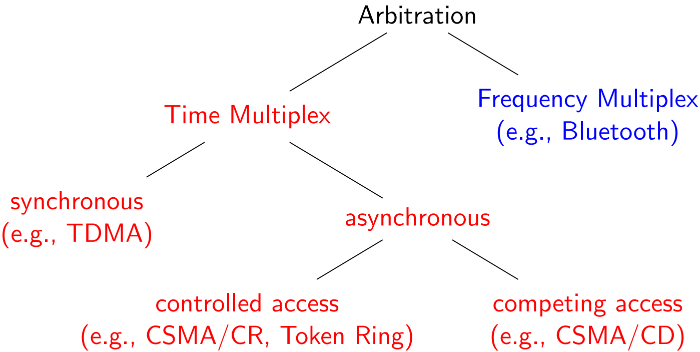
☞ cyber-physical systems

Materials are based on slides from Prof. Wang Yi, Prof. Lothar Thiele, and Prof. Peter Marwedel.

Communication: Requirements

- Performance
  - bandwidth and latency
  - guaranteed behavior (real-time)
- Efficiency
  - cost (material, installation, maintenance)
  - low power
- Robustness
  - fault tolerance
  - maintainability, diagnoseability
  - security, safety

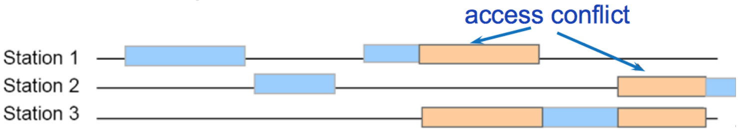
Protocol Classification



- Arbitration
  - Time Multiplex
    - synchronous (e.g., TDMA)
    - asynchronous
      - controlled access (e.g., CSMA/CR, Token Ring)
      - competing access (e.g., CSMA/CD)
  - Frequency Multiplex (e.g., Bluetooth)

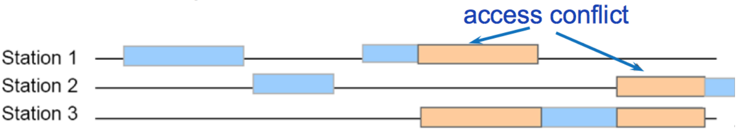
Random Access

- no access control, requires low medium utilization

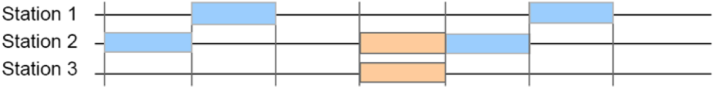


Random Access

- no access control, requires low medium utilization

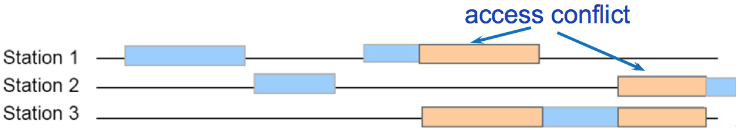


- improved variant: slotted random access

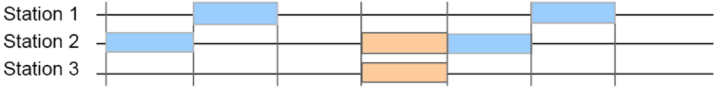


Random Access

- no access control, requires low medium utilization



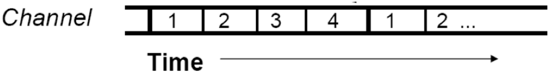
- improved variant: slotted random access



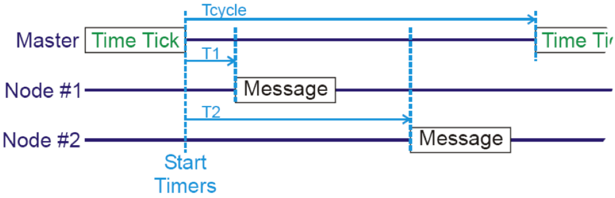
- What is the optimal sending rate p in case of n stations?
  - probability that a slot is not taken by others: $(1-p)^{n-1}$
  - probability to send successfully: $p \cdot (1-p)^{n-1}$
  - the maximum probability with respect to p happens when $d(p \cdot (1-p)^{n-1})/dp = 0$, i.e., $p = 1/n$.

TDMA (Time Division Multiple Access)

- Communication in statically allocated time slots
- Synchronization among all nodes necessary:
  - periodic repetition of communication frame or



  - master node sends out a synchronization frame
- Examples: TTP, static portion of FlexRay, satellite networks

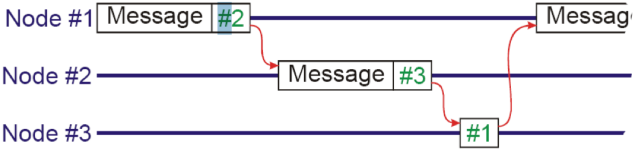


CSMA/CD

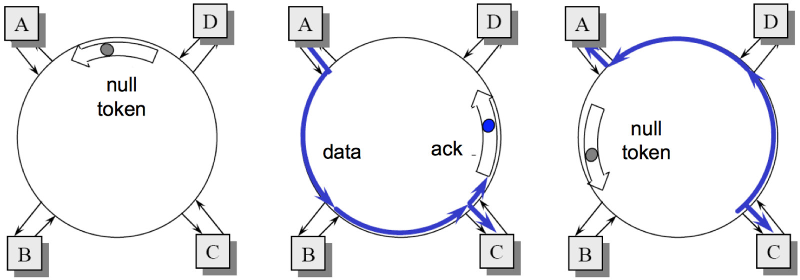
- CSMA/CD (Carrier Sense Multiple Access / Collision Detection)
- Try to avoid and detect collisions:
  - before starting to transmit, check whether the channel is idle
  - if a collision is detected (several nodes started almost simultaneously): wait for some time (backoff timer)
  - repeated collisions result in increasing backoff times
- Examples: Ethernet, IEEE 802.3
- Stochastic behavior, and problematic in general for real-time systems without any treatments

Token Protocols

- Token value determines which node is transmitting and/or should transmit next
  - Only the token holder may transmit
  - Master/slave polling is a special form
  - Null messages with tokens must be passed to prevent network from going idle
- Examples: IEEE 802.4, Profibus, TokenRing



Token Ring



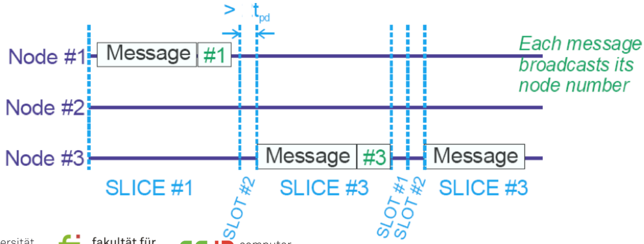
A requests transmission

A is token owner
A sends data to C
C sends acknowledge

A send null token

CSMA/CA

- Carrier Sense Multiple Access / Collision Avoidance
- Operation:
  - reserve s slots for n nodes; note: slots are normally idle - they are (short) time intervals, not signals; if slot is used it becomes a slice.
  - nodes keep track of global communication state by sensing
  - nodes start transmitting a message only during the assigned slot
  - If $s = n$, no collisions; if $s < n$, statistical collision avoidance
- Examples: 802.11, part of FlexRay



Each message broadcasts its node number

CSMA/CR

- Carrier Sense Multiple Access / Collision Resolution
- Operation:
  - Before any message transmission, there is a global arbitration

  - Each node (or each message type) is assigned a unique identification number
  - All nodes wishing to transmit compete by transmitting a binary signal based on their identification value
  - A node drops out the competition if it detects a dominant state while transmitting a passive state
  - Thus, the node with the lowest identification value wins
- Example: CAN Bus

Outline

Introduction

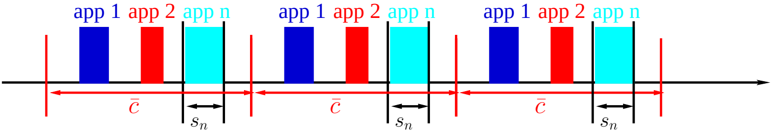
Analysis of TDMA

CAN (Controller Area Network)

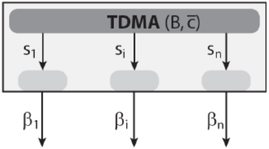
Flexray

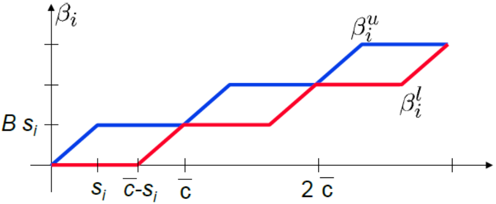
TDMA Resource

- Consider a real-time system consisting of n applications that are executed on a resource with bandwidth B that controls resource access using a TDMA (Time Division Multiple Access) policy.
- Analogously, we could consider a distributed system with n communicating nodes, that communicate via a shared bus with bandwidth B, with a bus arbitrator that implements a TDMA policy.
- TDMA policy: In every TDMA cycle of length $\bar{c}$, one single resource slot of length s_i is assigned to application i.



TDMA Resource





$$\beta^u_{TDMA}(\Delta) = B \min\left\{\left\lceil\frac{\Delta}{\bar{c}}\right\rceil s_i, \Delta - \left\lfloor\frac{\Delta}{\bar{c}}\right\rfloor(\bar{c} - s_i)\right\}$$

$$\beta^l_{TDMA}(\Delta) = B \max\left\{\left\lfloor\frac{\Delta}{\bar{c}}\right\rfloor s_i, \Delta - \left\lceil\frac{\Delta}{\bar{c}}\right\rceil(\bar{c} - s_i)\right\}$$

Schedulability Test for Sporadic Tasks under TDMA

The time-demand function $W_i(t)$ of the task τ_i is defined as follows:

$$W_i(t) = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil C_j.$$

Theorem

A system $\mathcal{T}$ of periodic, independent, preemptable, and constrained-deadline tasks is schedulable on one processor by a fixed-priority scheduling policy under a TDMA if

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i \ \ and \ \ W_i(t) \leq \beta^l_{TDMA}(t)$$

holds.

Exercise: How do we handle non-preemptive cases and minimum slice length for the TDMA schedule? That is, each packet has a fixed-length, and must be sent within one time slot.

Challenge for Utilization-Based Analysis

Suppose that the cycle length $\bar{c}$ is $f \cdot T_{\min}$, where $T_{\min}$ is the minimum period of the sporadic tasks and $f < 1$. Given $U_{TDMA} = \frac{\bar{s}}{\bar{c}}$ of the TDMA schedule, what is the least utilization upper bound to serve the sporadic tasks by using Rate-Monotonic scheduling under this TDMA schedule?

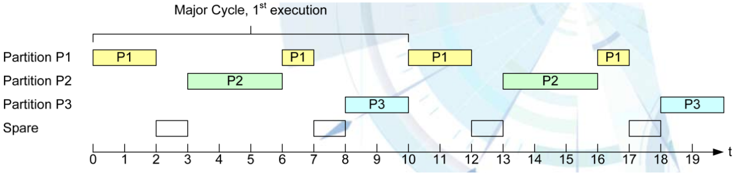
Challenge

The special case $f \rightarrow 0$ is easier. The general case which uses f and U_{TDMA} together is left as a challenge for you to explore.

Additional Remarks

Why is TDMA interesting?

- Integrated Modular Avionics (IMA) exactly partitions the functions by using a *flexible* TDMA, and uses fixed-priority preemptive scheduling within each partition
- Partitions are scheduled according to Time Division Multiple Access (TDMA)
- Execution times, number of partitions windows and offsets are defined in the Major Cycle



`https://www.symtavision.com/downloads/success-stories/06_IMA_SchedulingIssues_EADS_Breunig.pdf`

Outline

Introduction

Analysis of TDMA

CAN (Controller Area Network)

Flexray

CAN (Controller Area Network)

- Initiated in the late 70's to connect a number of processors over a cheaper shared serial bus
- From Bosch (mid 80's) for automotive applications
- De facto standard for in-vehicle communications.
- Fair cost
- Shared broadcast bus (one sender/many receivers) (CSMA/CR)
- CAN bus is a twisted wire
- Medium speed:
  - Max: 1Mbit/sec; typically used from 35 Kbit/sec up to 500Kbit/sec
- Highly robust (error mechanisms to overcome disturbance on the bus) and
- Real-time guarantees can be made about CAN performance

Bit Transmission on CAN

- Fundamental requirement: Everyone on the bus sees the current bit before the next bit is sent
    - This permits a very clever arbitration scheme (later)
- Global time is assumed and maintained
- Bits per second (depending on the length of CAN bus):
    - 1 Mbps CAN bus → 1 micro sec per bit: a bit can travel 100 m per 1000ns (thus max bus length 40∼50 m)
    - 40 Kbps CAN bus → 25000 ns per bit: A bit can travel 2500 m per 25000 ns (thus max bus length 1000∼1250 m)
- Bandwidth
    - 1 Mbps up to 40∼50 m (normal)
    - 0.5 Mbps upto 80∼100 m
    - 40 Kbps up to ∼1000 m
    - 5 Kbps up to ∼10,000 m (maximum)

CAN Frame

- Small sized frames (messages): 0 to 8 bytes
  - perfect for many embedded control applications
- Relatively high overhead: a frame size of more than 100 bits to send 64 data bits
  - do not use this for bulk data transfer
- Interrupt only after an entire message is received

CAN Addressing

- CAN bus can have an arbitrary number of nodes
  - Nodes do not have proper addresses
  - Each message has an 11-bit “field identifier”
  - Everyone interested in a message type listens to it
  - Works like this: “I’m passing a ball”
  - Not like this: “I’m passing a ball to Reus”
- Designer should allocate the message identifiers to the stations (different nodes send different messages!)
- Each node has a queue for messages ordered by priorities/identifiers

The CAN Arbitration Mechanism

- Shared broadcast bus
- Bus behaves like a large AND-gate - if all nodes sends 1 the bus becomes 1, otherwise 0.
    - 0: dominant bit (in fact, sending 0 by high voltage)
    - 1: recessieve bit
- A frame is tagged by an identifier
    - indicates contents of frame
    - most importantly, it is used for arbitration as priority
- Bit-wise arbitration
    - Each message has unique priority ⇒ node with message with lowest id wins arbitration
- Lowest id = highest priority!
- The CAN bus is a fixed-priority-based scheduled resource
- What happens if a CAN node goes crazy/haywire and transmits too many high priority frames?
    - This can make the bus useless
    - Assumed not to happen

CAN Message Scheduling and Analysis

- Each frame should be non-preemptive
- This is a non-preemptive fixed-priority scheduling
- The maximum bits per frame is 135 bits (by considering all the overheads and bitstuffing)
- This results in a maximum blocking time due to a frame of 135 bits in CAN
- For a CAN with 1Mbit/s, the blocking time is up to 135 μs

CAN Message Scheduling and Analysis Cont.

Theorem

A system $\mathcal{T}$ of periodic, independent, preemptable, and constrained-deadline message-passing tasks is schedulable on a CAN bus if

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i - C_i \text{ and } B_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil C_j \leq t$$

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i \text{ and } C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t}{T_j} \right\rceil C_j \leq t$$

where the higher-priority message types have lower indexes and B_i is the blocking time of message type i.

CAN Bus Utilization

- We remember from non-preemptive scheduling:
  - General utilization bounds are not possible
  - Utilization bounds based on computation time ratios are possible ($\gamma = \max_{\tau_i \in lp(\tau_k)} \left\{ \frac{C_i}{C_k} \right\} = \frac{B_k}{C_k}$)
- For CAN:
  - Fixed number of bits for arbitration
  - Maximum length of messages
- Previous results: the utilization bound of RM-NP for control area network (CAN) 2.0A is 25.8% and for CAN 2.0B is 29.5% (Andersson and Tovar, 2009)
- Current results: the utilization bound of RM-NP with respect to γ can still be up to 50.0%, if none of the lower priority tasks has a larger execution time, i.e., $\gamma \leq 1$ (von der Brüggen, Chen, and Huang, 2015)

A Typical Scenario

Pseudo-code

```
at each timer interrupt
do
    • perform analog-to-digital conversion to get $y$;
    • compute control output $u$;
    • send output $u$ to CAN;
od
```

- There is a set of tasks computing values/results of some functions
  - the results will be the sent over the CAN Bus
- There is a set of periodic tasks sending CAN messages
  - they will be released by the computing tasks
- The CAN messages will trigger some other computing tasks on the other CAN nodes

A Typical Scenario

Pseudo-code

```
at each timer interrupt
do
    • perform analog-to-digital conversion to get $y$;
    • compute control output $u$;
    • send output $u$ to CAN;
od
```

- There is a set of tasks computing values/results of some functions
  - the results will be the sent over the CAN Bus
- There is a set of periodic tasks sending CAN messages
  - they will be released by the computing tasks
- The CAN messages will trigger some other computing tasks on the other CAN nodes
- A task has variant response times, which cause jitters for the message transmission task
- A message may trigger a computation task on another node, which causes jitters for the computation task due to the variant transmission delays/queuing times

Adding Jitter to Analysis

Theorem

A system $\mathcal{T}$ of periodic, independent, preemptable, and constrained-deadline message-passing tasks is schedulable on a CAN bus if

$$\forall \tau_i \in \mathcal{T} \ \exists t \text{ with } 0 < t \leq D_i - C_i - J_i \text{ and } B_i + \sum_{j=1}^{i-1} \left\lceil \frac{t + J_j}{T_j} \right\rceil C_j \leq t$$

$$\forall \tau_i \in \mathcal{T} \ \exists t \text{ with } 0 < t \leq D_i - J_i \text{ and } C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t + J_j}{T_j} \right\rceil C_j \leq t$$

where the higher-priority message types have lower indexes and B_i is the blocking time of message type i and J_j is the jitter of a task τ_j.

Adding Jitter to Analysis

Theorem

A system $\mathcal{T}$ of periodic, independent, preemptable, and constrained-deadline message-passing tasks is schedulable on a CAN bus if

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i - C_i - J_i \text{ and } B_i + \sum_{j=1}^{i-1} \left\lceil \frac{t + J_j}{T_j} \right\rceil C_j \leq t$$

$$\forall \tau_i \in \mathcal{T}\ \exists t \text{ with } 0 < t \leq D_i - J_i \text{ and } C_i + \sum_{j=1}^{i-1} \left\lceil \frac{t + J_j}{T_j} \right\rceil C_j \leq t$$

where the higher-priority message types have lower indexes and B_i is the blocking time of message type i and J_j is the jitter of a task τ_j.

Why do we need $-J_i$ here?

Outline

Introduction

Analysis of TDMA

CAN (Controller Area Network)

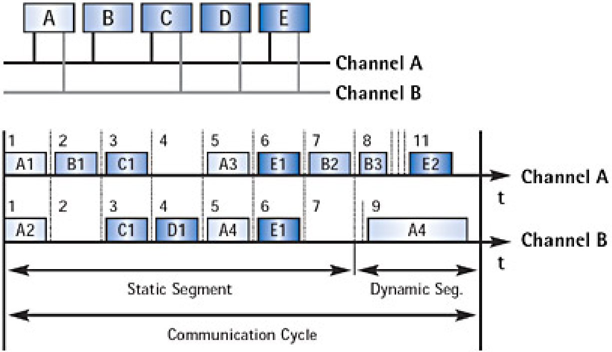
Flexray

Flexray

- Developed by the FlexRay consortium (BMW, Ford, Bosch, DaimlerChrysler, . . .)
- Meets requirements with transfer rates $\gg$ CAN standard
  - High data rate can be achieved:
  - initially targeted for ~10Mbit/sec;
- Design allows much higher data rates
- Improved error tolerance and time-determinism
- Flexible TDMA protocol
- Cycle subdivided into a static and a dynamic segment.
  - Static segment is based on a fixed allocation of time slots to nodes.
  - Dynamic segment for transmission of ad-hoc communication with variable bandwidth requirements.

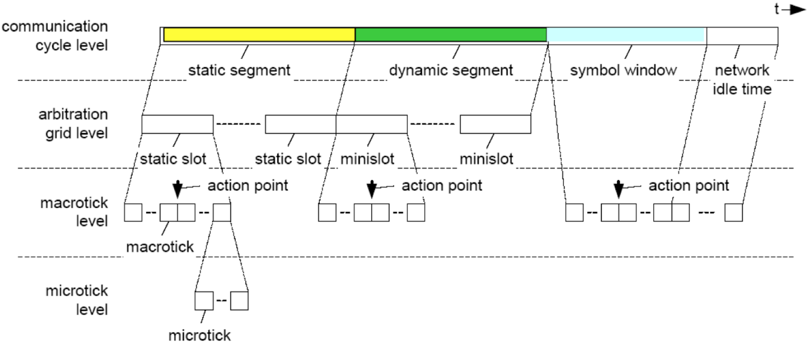
Flexray

- Use of two independent channels to eliminate single-point failures and to allow flexibility of different channel configurations
- Bandwidth in the dynamic segment is used only when it is actually needed.



http://www.ixxat.de/introduction_flexray_de.html?markierung=flexray

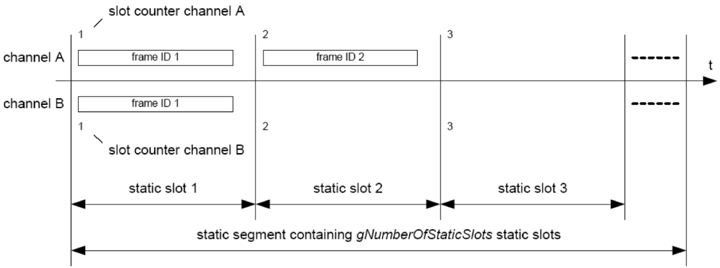
FlexRay Message Cycle



Static Segment

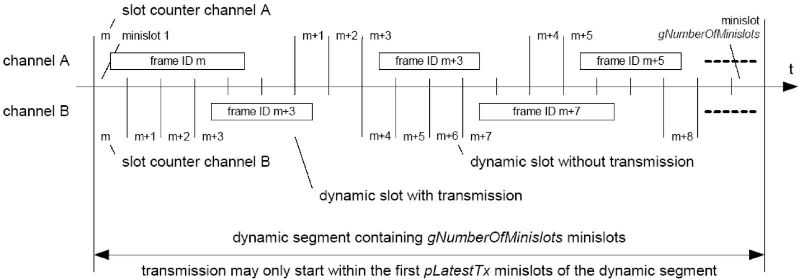
TDMA messages, most likely used for critical messages

- All static slots are the same length
- All static slots are repeated in order every communication cycle
- All static slot times are expended in the cycle whether they are used or not



Dynamic Segment

- Each minislot is an opportunity to send a message
- If message is not sent, minislot elapses unused (short idle period)
- All nodes watch whether a message is sent (they can count minislots)



Response-Time Analysis

- The worst-case response time analysis in static segments is similar to the TDMA with multiple time slots in a cycle.
- The worst-case response time analysis in the dynamic segments is similar to CAN, but now requires further considerations of the offsets due to the static segments.

Traian Pop, Paul Pop, Petru Eles, Zebo Peng, Alexandru Andrei: Timing Analysis of the FlexRay Communication Protocol. ECRTS 2006: 203-216