Disclaimer

- The material provided in this document is not my original work and is a summary of some one else's work(s).
- A simple Google search of the title of the document will direct you to the original source of the material.
- I do not guarantee the accuracy, completeness, timeliness, validity, non-omission, merchantability or fitness of the contents of this document for any particular purpose.
- Downloaded from najeebkhan.github.io

Greedy Heuristic for Dynamic Segment Scheduling

Presented By

Greedy Heuristic for Dynamic Segment Scheduling

Presented By
Najeeb
2013 – 6 – 20

Outline

- Introduction
- Message Schedule for the DS
- Message Grouping
- Proposed Heuristic Algorithm

Introduction

Introduction

- We discuss the scheduling of messages in the dynamic segment of the FlexRay

Introduction

- We discuss the scheduling of messages in the dynamic segment of the FlexRay
- In [1] bounds on the generation times and the timing requirements of the signals is taken into consideration to propose a reservation-based scheduling approach that preserves the flexible medium access of the DS

Introduction

- We discuss the scheduling of messages in the dynamic segment of the FlexRay
- In [1] bounds on the generation times and the timing requirements of the signals is taken into consideration to propose a reservation-based scheduling approach that preserves the flexible medium access of the DS
- [1] uses ILP which is computationally very complex, we formulate a fast heuristic algorithm for the scheduling of the dynamic segment

FlexRay Protocol

FlexRay Protocol

- FC → SS + DS + SW +NIT

FlexRay Protocol

- FC → SS + DS + SW +NIT
- The smallest time unit in the DS is the minislot (MS) with a duration of T_{MS} (in milliseconds), and the DS contains a fixed number of N_{DS} MS

FlexRay Protocol

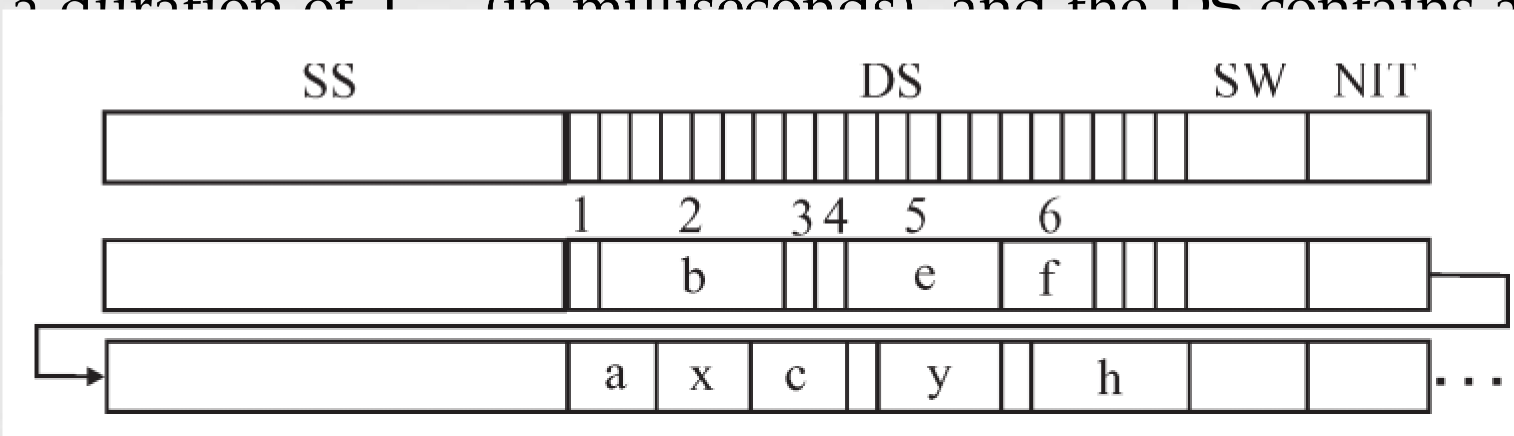
- FC → SS + DS + SW +NIT
- The smallest time unit in the DS is the minislot (MS) with a duration of T_{MS} (in milliseconds), and the DS contains a fixed number of N_{DS} MS
- If a message is transmitted in a DYS, then the length of the DYS is equal to the number of MS needed for message transmission. Otherwise, the length of the DYS is one MS

FlexRay Protocol

- FC → SS + DS + SW +NIT
- The smallest time unit in the DS is the minislot (MS) with a duration of T_{MS} (in milliseconds), and the DS contains a fixed number of N_{DS} MS
- If a message is transmitted in a DYS, then the length of the DYS is equal to the number of MS needed for message transmission. Otherwise, the length of the DYS is one MS
- The arbitration procedure ensures that only frames with a FID that equals the current value of the slot counter can be transmitted

FlexRay Protocol

- FC → SS + DS + SW +NIT
- The smallest time unit in the DS is the minislot (MS) with a duration of T (in milliseconds), and the DS contains a



-
- FID that equals the current value of the slot counter can be transmitted

Previous Work

Previous Work

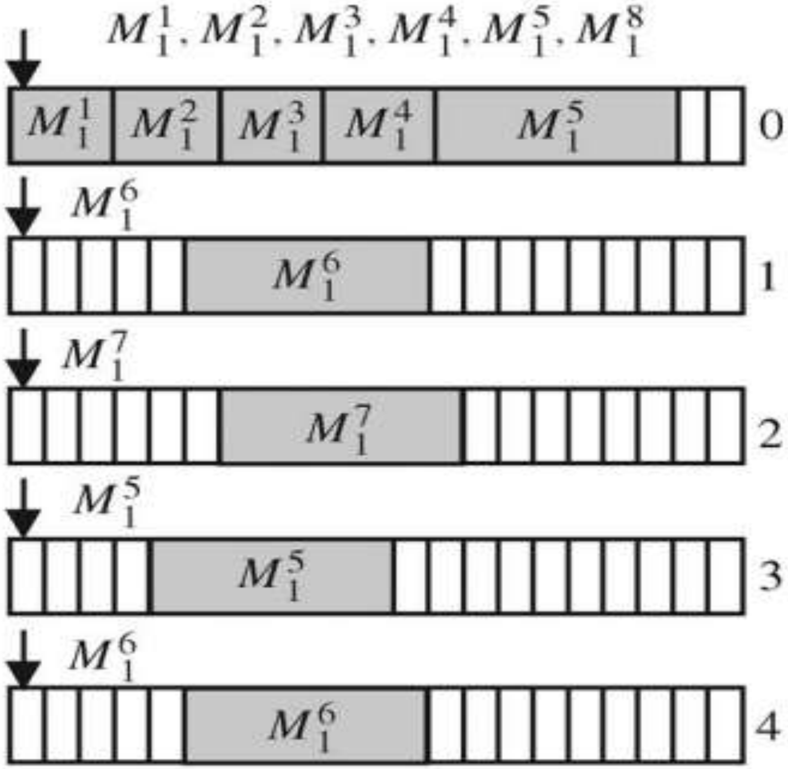
- Previous work on the scheduling of DS uses DM approach and also assumes that T_C, T_{SS} and T_{DS} are pre determined

Previous Work

- Previous work on the scheduling of DS uses DM approach and also assumes that T_C, T_{SS} and T_{DS} are pre determined
- How DM fails?

Previous Work

M_1^n	lm_1^n [MS]	dm_1^n [ms]	FID
M_1^1	2	15	1
M_1^2	2	15	2
M_1^3	2	15	3
M_1^4	2	15	4
M_1^5	40	15	5
M_1^6	40	15	6
M_1^7	40	15	7
M_1^8	112	25	8



Message Schedule For The DS

Message Schedule For The DS

- A reservation R for a node n is a 4-tuple (n,rp,w,l) with the reservation period rp $\in$ N, the offset w $\in$ {0,...,rp−1} and the reservation length l $\in$ N

Message Schedule For The DS

- A reservation R for a node n is a 4-tuple (n,rp,w,l) with the reservation period rp $\in$ N, the offset w $\in$ {0,...,rp−1} and the reservation length l $\in$ N
- R stands for l MS that are reserved at all FCs (z*rp+w), z $\in$ N_0, while 1 MS is reserved in the remaining FCs

Message Schedule For The DS

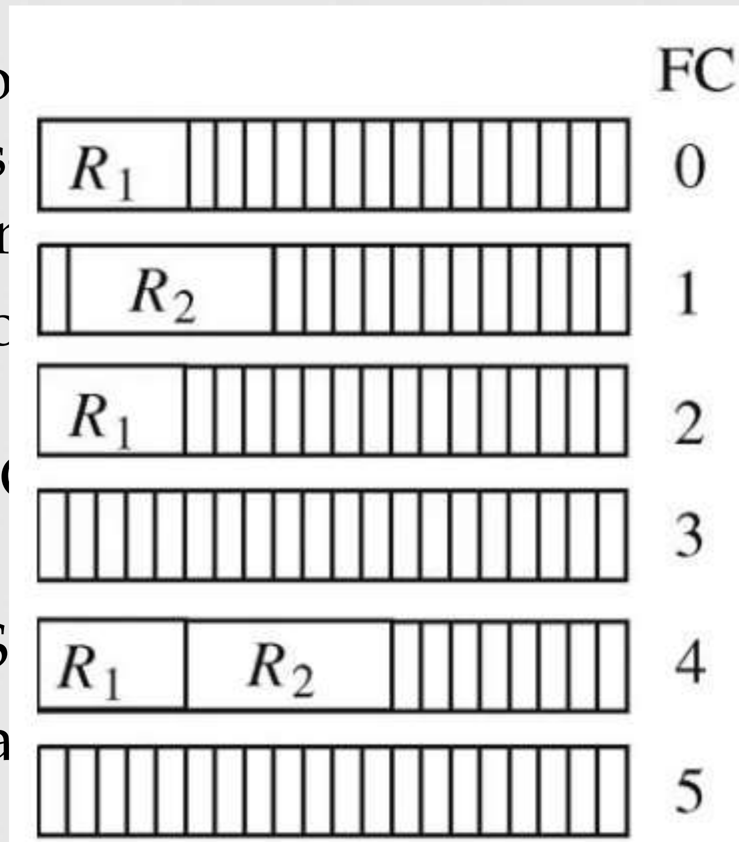
- A reservation R for a node n is a 4-tuple (n,rp,w,l) with the reservation period rp $\in$ N, the offset w $\in$ {0,...,rp−1} and the reservation length l $\in$ N
- R stands for l MS that are reserved at all FCs (z*rp+w), z $\in$ N_0, while 1 MS is reserved in the remaining FCs
- Bandwidth reservation per FC for a given R is B_R= l/rp MS

Message Schedule For The DS

- A reservation R for a node n is a 4-tuple (n,rp,w,l) with the reservation period rp $\in$ N, the offset w $\in$ {0,...,rp−1} and the reservation length l $\in$ N
- R stands for l MS that are reserved at all FCs (z*rp+w), z $\in N_0$, while 1 MS is reserved in the remaining FCs
- Bandwidth reservation per FC for a given R is B_R= l/rp MS
- Each reservation is associated with a FID

Message Schedule For The DS

- A reservatio … tuple (n,rp,w,l) with the res … the offset $w \in \{0,\ldots,rp-1\}$ ar … $\in N$
- R stands fo … ed at all FCs $(z*rp+w)$, z … eserved in the remaining F…
- Bandwidth … a given R is $B_R = l/rp$ MS…
- Each reserva … ID



Performance Metrics

Performance Metrics

- Cycle Load L_j : The maximum number of MS that is reserved for message transmission in FC j for an arbitrary assignment of FIDs, considering that at most one FID can be assigned per message

Performance Metrics

- Cycle Load L_j : The maximum number of MS that is reserved for message transmission in FC j for an arbitrary assignment of FIDs, considering that at most one FID can be assigned per message
- L_j includes both the case where no message is transmitted for an FID (duration of 1 MS) and the case where a message is transmitted

Performance Metrics

- Cycle Load L_j : The maximum number of MS that is reserved for message transmission in FC j for an arbitrary assignment of FIDs, considering that at most one FID can be assigned per message
- L_j includes both the case where no message is transmitted for an FID (duration of 1 MS) and the case where a message is transmitted

$$L_j = \sum_{R \in \mathcal{R}_j} l + \left(|\mathcal{M}_{\mathrm{S}}| - \sum_{R \in \mathcal{R}_j} 1 \right) = \sum_{R \in \mathcal{R}_j} (l-1) + |\mathcal{M}_{\mathrm{S}}|.$$

Where $j = (z \cdot rp + w)$

Performance Metrics

Performance Metrics

- We define the maximum cycle load as

Performance Metrics

- We define the maximum cycle load as

$$L_{max} = \max_{j \in \{1,\ldots,rp\}} (Lj)$$

Performance Metrics

- We define the maximum cycle load as

$$L_{max} = \max_{j \in \{1,\ldots,rp\}} (Lj)$$

then, we choose $N_{DS}=L_{max}$ and minimize L_{max} to determine a feasible schedule with the shortest possible DS

Performance Metrics

Performance Metrics

- Bandwidth Reservation: Indicate the number of MS reserved per FC for each node $n \in N$ and for all of the nodes

Performance Metrics

- Bandwidth Reservation: Indicate the number of MS reserved per FC for each node n∈N and for all of the nodes

$$B^n = \sum_{R \in R^n} (l/rp)$$

$$\sum_{n=1}^{N} B^n$$

Optimal Scheduling of Messages

Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups have been computed by the algorithms in [1].

Optimal Scheduling of Messages

$\mathcal{G}_1 = \{M_1^1\},\ pm_1^1 = 3,\ dm_1^1 = 5$ $R_1 = (1, 2, w_1, 20)$	$\mathcal{G}_2 = \{M_1^1, M_2^1\},$ $R_2 = (1, 2, w_2, 30)$
$\mathcal{G}_3 = \{M_2^1\},\ pm_2^1 = 5,\ dm_2^1 = 7$ $R_3 = (1, 4, w_3, 30)$	$\mathcal{G}_4 = \{M_3^1\},\ pm_3^1 = 4, dm_3^1 = 6$ $R_4 = (1, 3, w_4, 10)$
$\mathcal{G}_5 = \{M_1^2\},\ pm_1^2 = 3,\ dm_1^2 = 7$ $R_5 = (2, 2, w_5, 22)$	$\mathcal{G}_6 = \{M_1^2, M_2^2\}$ $R_6 = (2, 2, w_6, 48)$
$\mathcal{G}_7 = \{M_1^2, M_2^2, M_3^2\}$ $R_7 = (2, 2, w_7, 48)$	$\mathcal{G}_8 = \{M_1^2, M_3^2\}$ $R_8 = (2, 2, w_8, 30)$
$\mathcal{G}_9 = \{M_1^2, M_4^2\}$ $R_9 = (2, 2, w_9, 42)$	$\mathcal{G}_{10} = \{M_2^2\}, pm_2^2 = 7, dm_2^2 = 9$ $R_{10} = (2, 6, w_{10}, 48)$
$\mathcal{G}_{11} = \{M_3^2\}, pm_3^2 = 7, dm_3^2 = 9$ $R_{11} = (2, 6, w_{11}, 30)$	$\mathcal{G}_{12} = \{M_4^2\}, pm_4^2 = 5, dm_4^2 = 5$ $R_{12} = (2, 4, w_{12}, 42)$

Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups have been computed by the algorithms in [1].

Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups have been computed by the algorithms in [1].
- Our goal is now to determine G_S and the offsets wi of the corresponding reservations such that Lmax(and, thus, the required duration T_{DS} of the DS) is minimized

Heuristic Algorithm

```
G=set of groups for each node
gsel=[]
for each node n:
    for each group gi:
        GT=Largest(G)
        if NoOverlap between gsel and GT:
            {
            gsel = gsel U GT
            G = G - GT
            }
```

Heuristic Algorithm

- In contrast to the ILP approach used in [1], we propose a fast heuristic algorithm for the selection of optimal groups

```
G=set of groups for each node
gsel=[]
for each node n:
    for each group gi:
        GT=Largest(G)
        if NoOverlap between gsel and GT:
            {
            gsel = gsel U GT
            G = G - GT
            }
```

Heuristic Algorithm

```
w=0
W=[]
for each reservation R:
#1      if w is in W:
            w++
        if (w < period of R)
            offset=w
            W = W U w
        else
            w=0
            goto 1
```

Heuristic Algorithm

- And for the selection of offsets

```
w=0
W=[]
for each reservation R:
#1     if w is in W:
          w++
       if (w < period of R)
          offset=w
          W = W U w
       else
          w=0
          goto 1
```

References

References

1. Message Scheduling for the FlexRay Protocol: The Dynamic Segment

Thank You