# 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.
- 

# Message Scheduling for the FlexRay protocol

The Dynamic Segment

Presented By

# Message Scheduling for the FlexRay protocol

The  Dynamic Segment

Presented By
Najeeb
2013 – 5 – 30

# Outline

- Introduction
- FlexRay protocol
- Message Schedule for the DS
- Message Grouping
- Optimal Scheduling of Messages
- Results
- Conclusion

# 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 this paper 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

# 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

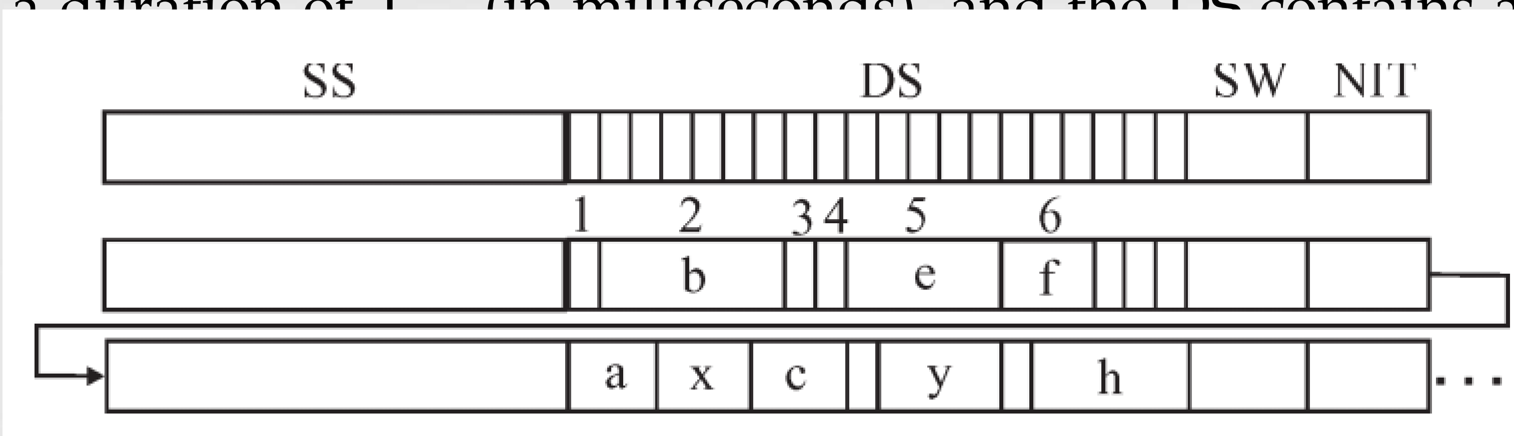
- FC → SS + DS + SW +NIT
- The smallest time unit in the DS is the minislot (MS) with a duration of $T_{MS}$ (in milliseconds), and the DS contains a fixed number of $N_{DS}$ MS
- If a message is transmitted in a DYS, then the length of the DYS is equal to the number of MS needed for message transmission.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

# FlexRay Protocol

- The length of a sporadic message can be computed, including the signal data s in multiples of two-byte words, the FlexRay framing overhead s*4bits + OF, and the communication-free DYS idle phase as

# FlexRay Protocol

- The length of a sporadic message can be computed, including the signal data s in multiples of two-byte words, the FlexRay framing overhead s*4bits + OF, and the communication-free DYS idle phase as

$$l_m^n = \lceil (s_m^n \cdot 16 \text{ bits} + s_m^n \cdot 4 \text{ bits} + O_{\mathrm{F}})\, \tau_{\mathrm{bit}} / T_{\mathrm{MS}} \rceil$$

# Previous Work

# Previous Work

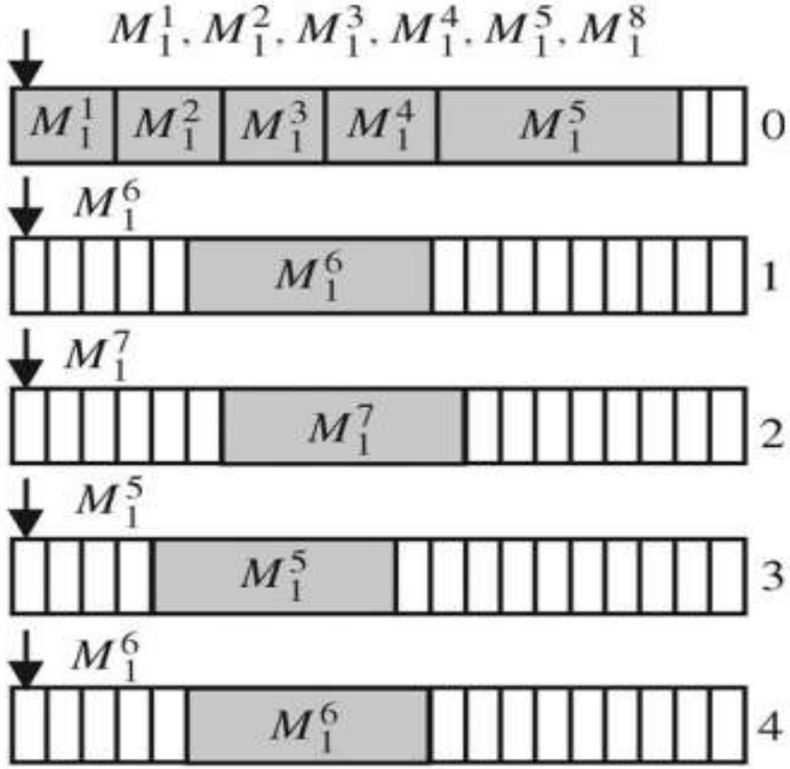
- Previous work on the scheduling of DS uses DM approach and also assumes that $T_C$, $T_{SS}$ and $T_{DS}$ are pre determined

# 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?

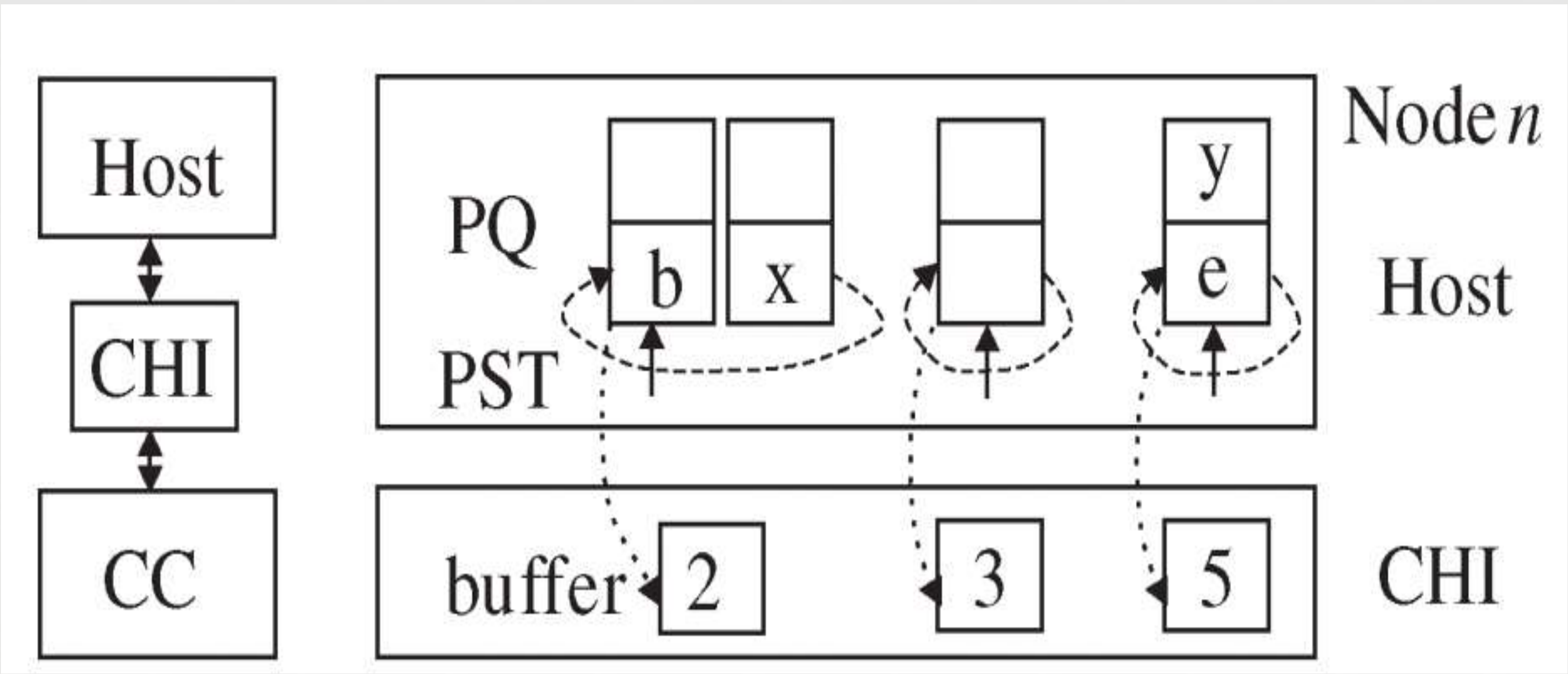
# Previous Work

| $M_1^n$ | $lm_1^n$ [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 |



# FlexRay Software Architecture

# FlexRay Software Architecture



# 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

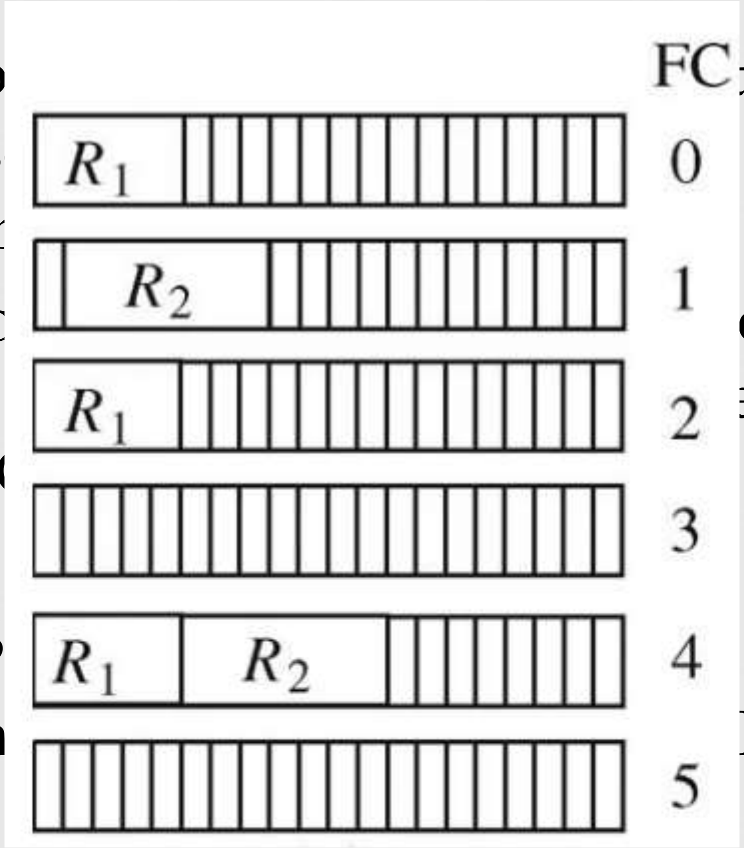
- A reservation R for a node n is a 4-tuple (n,rp,w,l) with the reservation period rp $\in$ N, the offset w $\in$ {0,...,rp−1} and the reservation length l $\in$ N
- R stands for l MS that are reserved at all FCs (z*rp+w), z $\in N_0$, while 1 MS is reserved in the remaining FCs
- Bandwidth reservation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



# FC Time

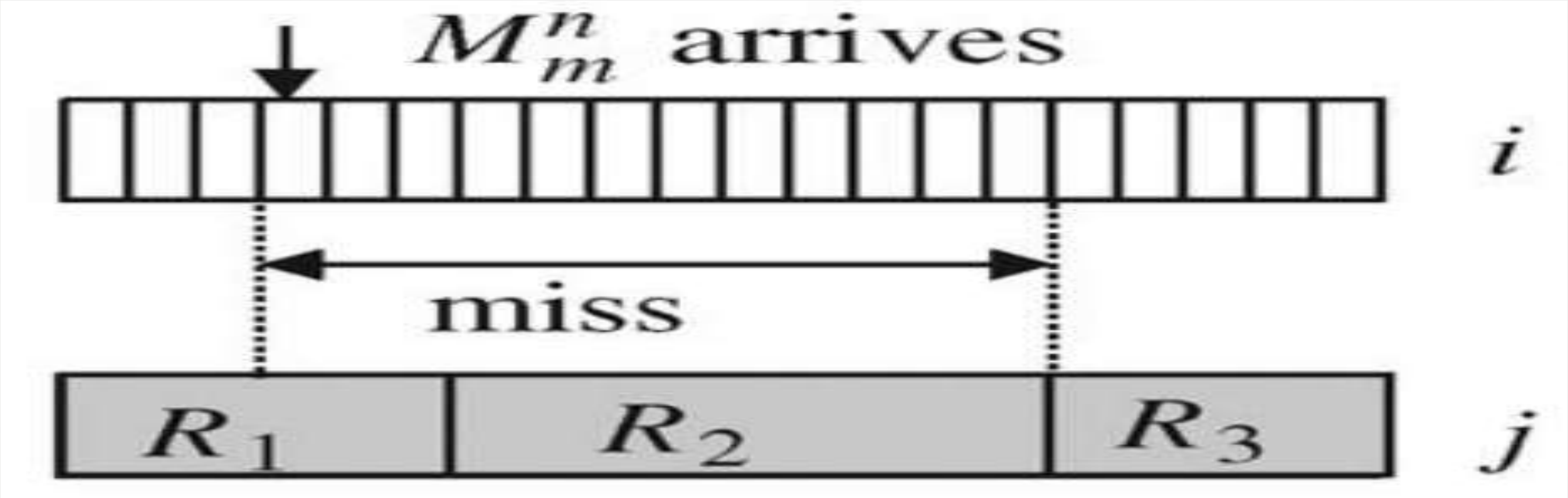
# FC Time

- If Tc is chosen larger than d, transmitting M multiple times in the same FC does not guarantee that M meets the deadline

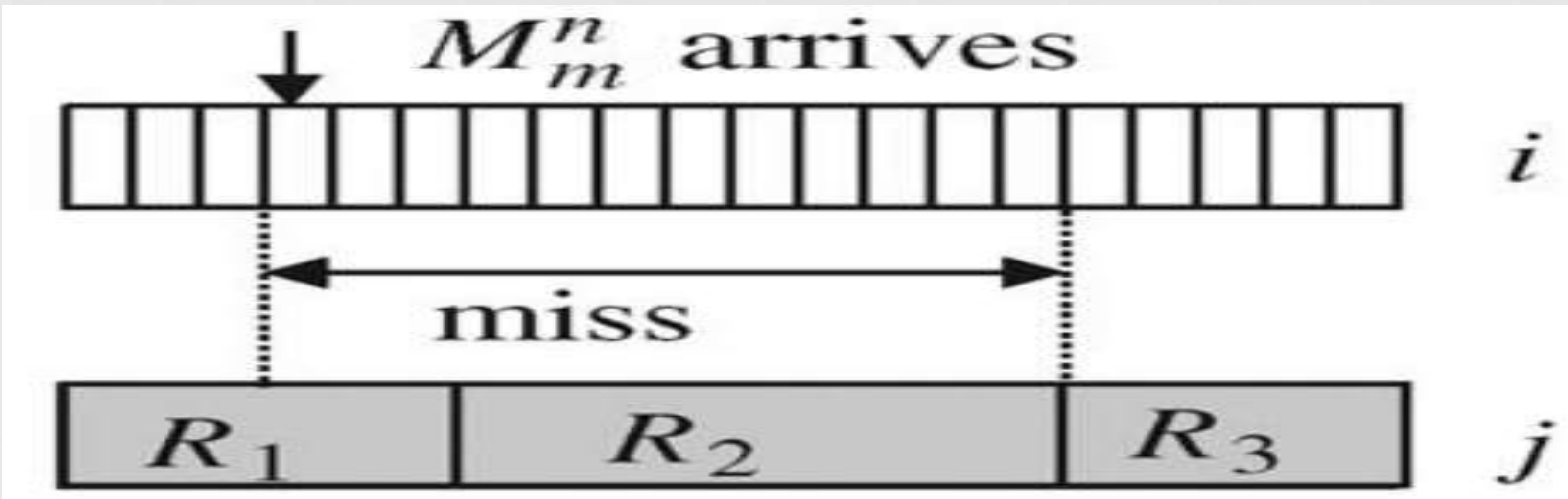
# FC Time

- If Tc is chosen larger than d, transmitting M multiple times in the same FC does not guarantee that M meets the deadline
- Hence, it must hold that $T_c \leq d_{min}$, where $d_{min}$ is the minimum deadline among all sporadic message deadlines.

# FC Time

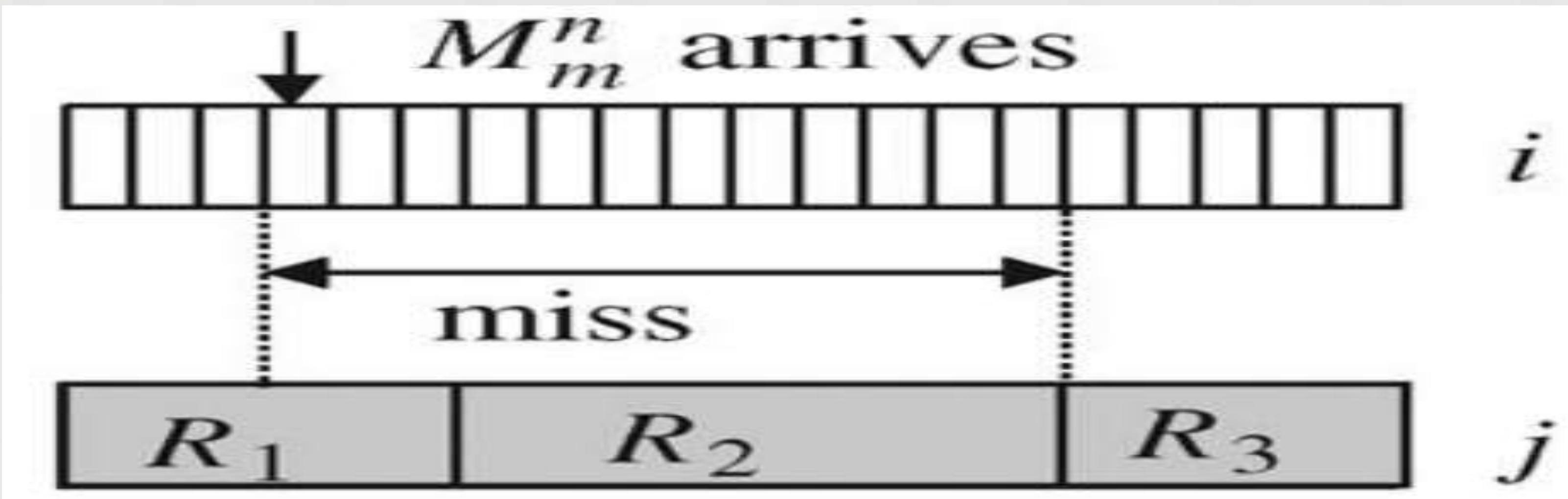


# FC Time




- Additionally (rp+1) * Tc ≤ dm and hence rp=dm/Tc−1

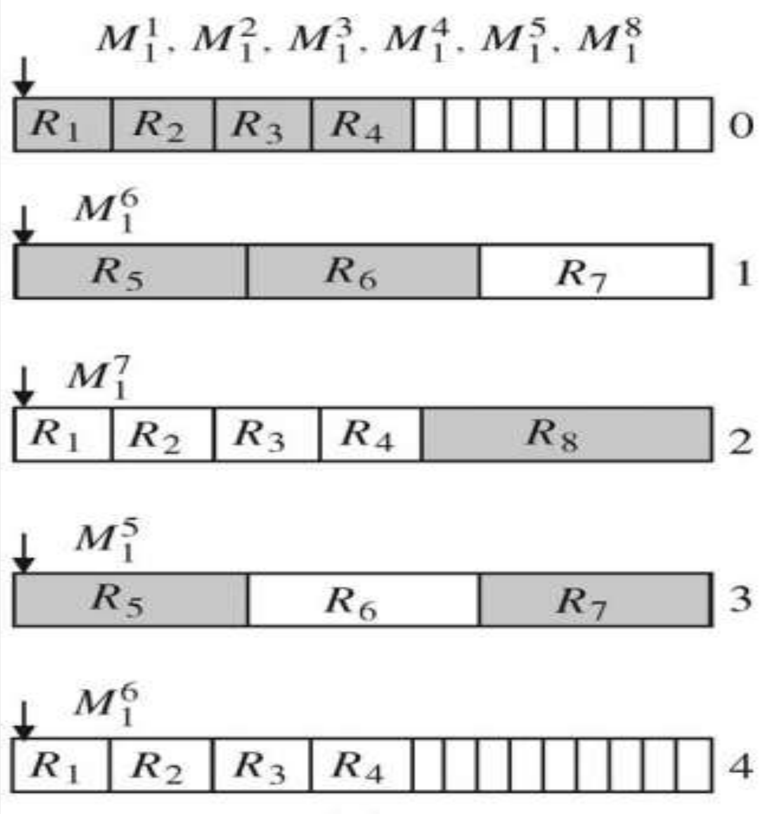
# FC Time




- Additionally (rp+1) * Tc ≤ dm and hence rp=dm/Tc−1
- Having determined the message reservation period for each message in MS, a good choice for the FC time Tc is the gcd of all rp

# Example Revisited

# Example Revisited



# 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

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

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

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

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

# Message Grouping

**Algorithm 4.1 (Check and Add)**
Input: $M_l^n$, $\mathcal{G}_q^n$, $\mathcal{G}^n$.
Init: result $= more\_fit$
**if** ($P_{q,l}^n < 0$ or $RM_{q,l}^n < 1$)
  result $= no\_fit$
**else**
  $\mathcal{G}_q^n := \mathcal{G}_q^n \cup \{M_l^n\}$ and $\mathcal{G}^n = \mathcal{G}^n \cup \mathcal{G}_q^n$
  **if** ($RM_{q,l}^n = 1$)
    result $= last\_fit$
**return** result

# Message Grouping

**Algorithm 4.2 (Group)**

Input: $LM^n_{\mathrm{S}}, M^n_{\mathrm{c}}, \mathcal{G}^n_q, \mathcal{G}^n$

(**while** $M^n_{\mathrm{c}} < \mathbf{last}(LM^n_{\mathrm{S}})$)

  $M^n_{\mathrm{c}} = \mathbf{next}(M^n_{\mathrm{c}}))$

  $temp\mathcal{G}^n_q = \mathcal{G}^n_q$

  result $=$ $\mathbf{Check\ and\ Add}(M^n_{\mathrm{c}}, temp\mathcal{G}^n_q, \mathcal{G}^n)$

  **if** (result $= more\_fit$ and $M^n_{\mathrm{c}} \neq \mathbf{last}(LM^n_{\mathrm{S}})$)

    $tempM^n_{\mathrm{c}} = M^n_{\mathrm{c}}$

    **Group** $(LM^n_{\mathrm{S}}, tempM^n_{\mathrm{c}}, temp\mathcal{G}^n_q, \mathcal{G}^n)$

# Optimal Scheduling of Messages

# Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups in G have been computed as

# Optimal Scheduling of Messages

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

# Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups in G have been computed as

# Optimal Scheduling of Messages

- Let N={1,2}. We assume that the groups in G have been computed as
- 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

# Optimization Contd…

# Optimization Contd…

- We formulate integer programming problems with two components to find the optimal message schedule
  - The first component addresses the selection of the message groups and the corresponding reservations. $gi \in \{0,1\}$ takes the value of 1 if $Gi \in G_S$ and is 0 otherwise
  - The second component is determining the reservation offsets. $x_{i,k} \in \{0,1\}$ takes the value of 1 if $w_i = k$ and is 0 otherwise, where $k=0,\ldots,rp_i-1$

# Optimization Contd…

Consider a reservation $R_i$ that corresponds to $\mathcal{G}_i \in \mathcal{G}$. The contribution of $R_i$ to $L_j$, $j = 1, \ldots, G_{\mathrm{RP}} - 1$, is given as follows.

1) $g_i = 0$: Then, $\mathcal{G}_i \notin \mathcal{G}^S$ and $R_i$ does not add to $L_j$.
2) $g_i = 1$ and $x_{i,k} = 1$ for $k = j \bmod rp_i$: Then, $w_i = k$ and $l_i$ MS are reserved for $R_i$ in $L_j$.
3) $g_i = 1$ and $x_{i,k} = 0$ for $k = j \bmod rp_i$: Then, $w_i \neq k$ and one MS is reserved for $R_i$ in $L_j$.

Accordingly, we can express the cycle load $L_j$ as follows:

$$L_j = \sum_{\mathcal{G}_i \in \mathcal{G}} g_i \cdot (x_{i,k} \cdot l_i + (1 - x_{i,k}) \cdot 1)$$

# Optimization Contd…

# Optimization Contd…

- Assuming without loss of generality that $L_{max} = L_0$, the expression to be minimized is

# Optimization Contd…

- Assuming without loss of generality that $L_{max} = L_0$, the expression to be minimized is

$$\min_X L_0 = \min_X \sum_{\mathcal{G}_i \in \mathcal{G}} g_i \cdot x_{i,0} \cdot l_i + g_i(1 - x_{i,0}) \cdot 1$$

# Optimization Contd…

- Assuming without loss of generality that $L_{max} = L_0$, the expression to be minimized is

$$\min_X L_0 = \min_X \sum_{\mathcal{G}_i \in \mathcal{G}} g_i \cdot x_{i,0} \cdot l_i + g_i(1 - x_{i,0}) \cdot 1$$

- With the constraints

# Optimization Contd…

- Assuming without loss of generality that $L_{max} = L_0$, the expression to be minimized is

$$\min_X L_0 = \min_X \sum_{\mathcal{G}_i \in \mathcal{G}} g_i \cdot x_{i,0} \cdot l_i + g_i(1 - x_{i,0}) \cdot 1$$

- With the constraints

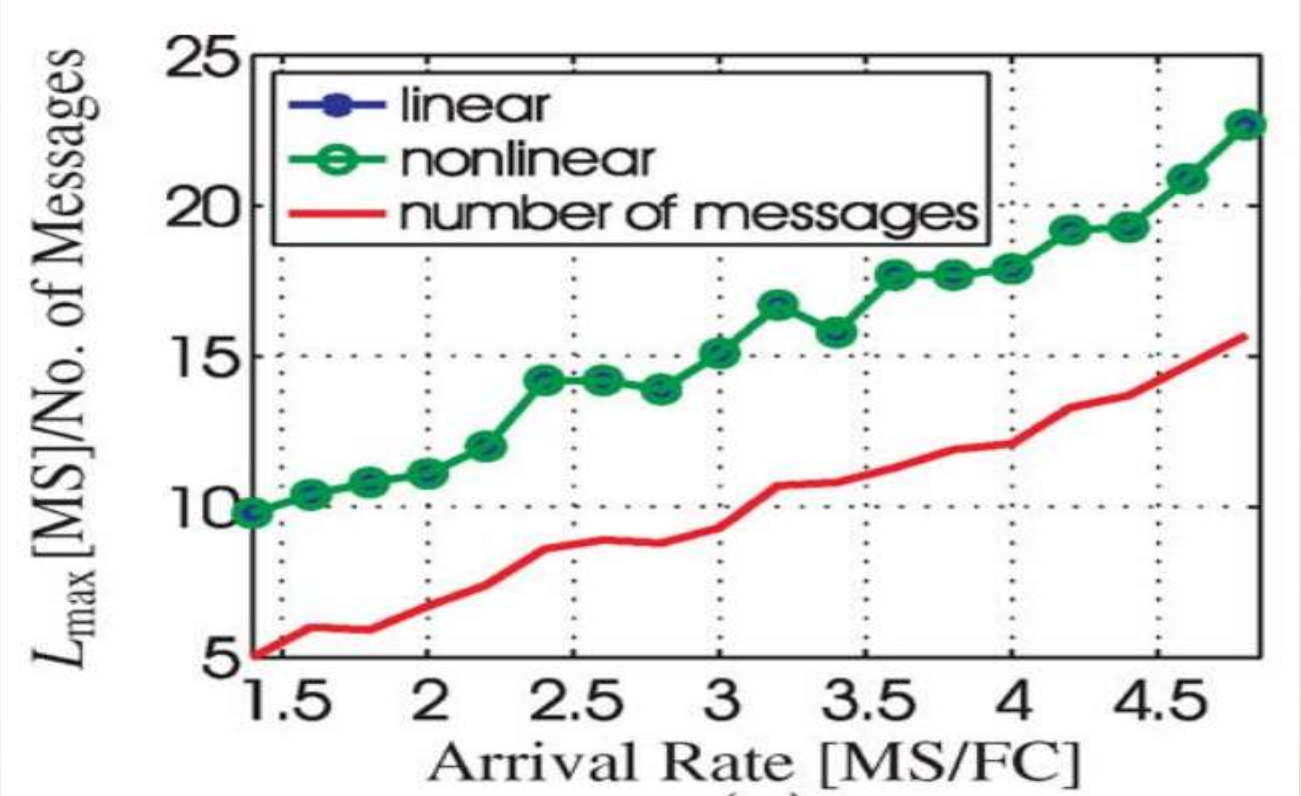
$$\forall M_m^n, \qquad \sum_{i, M_m^n \in \mathcal{G}_i} g_i = 1$$

$$\text{for } i = 1, \ldots, |\mathcal{G}|, \qquad \sum_{k=0}^{rp_i - 1} x_{i,k} = g_i$$

$$\text{for } j = 1, \ldots, G_{\mathrm{RP}} - 1, \qquad L_j \leq L_0.$$

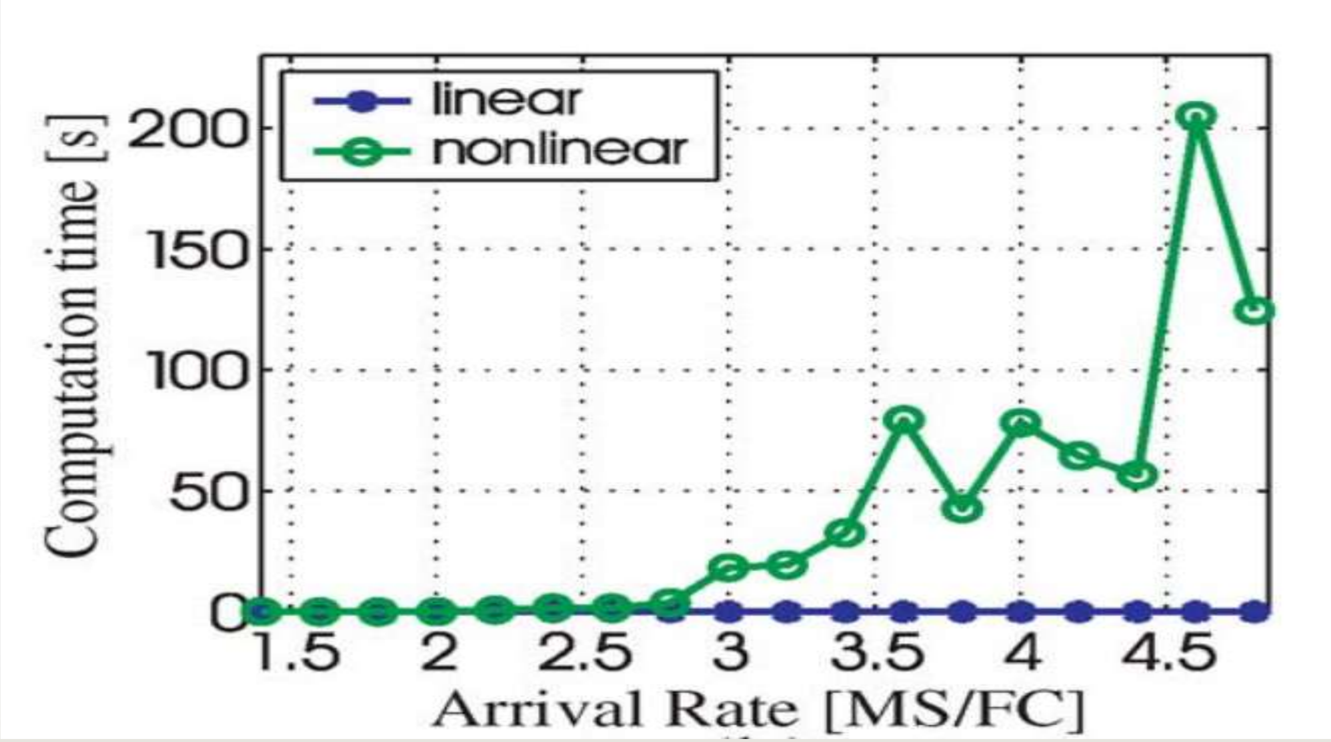
# Results

- Plot of the maximum cycle load Lmax, against the arrival rate



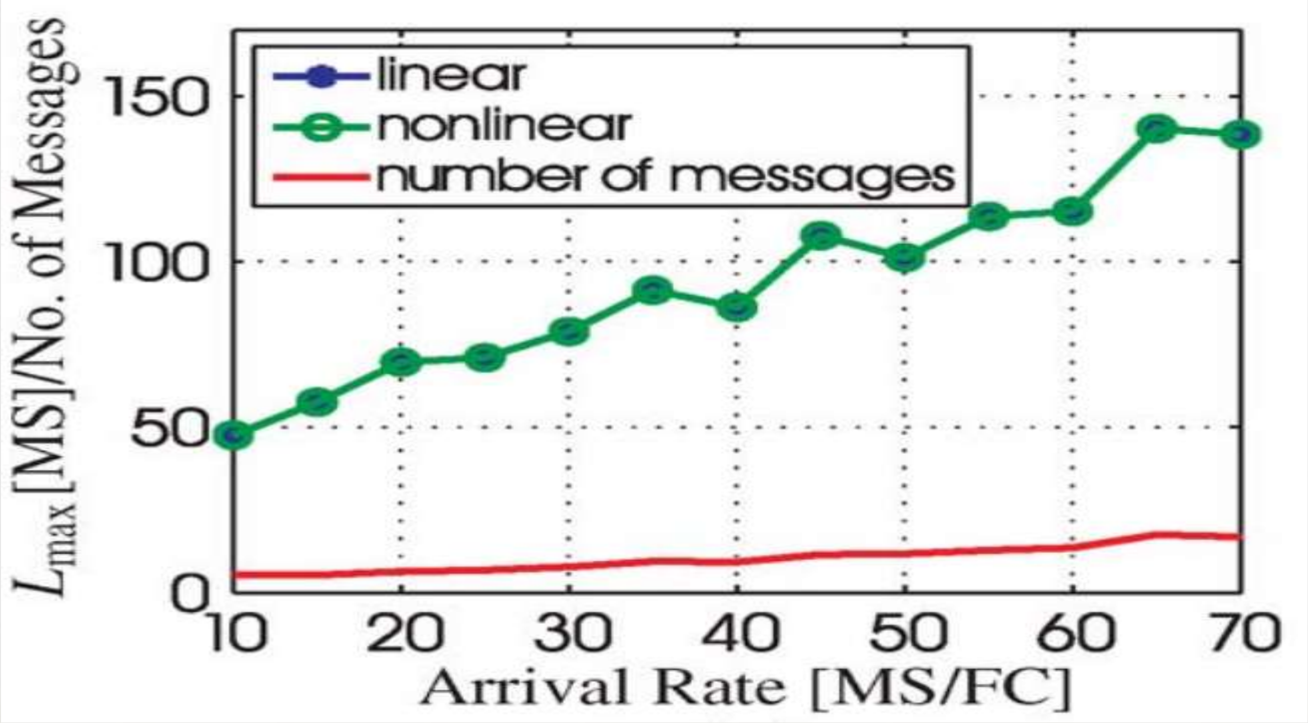
# Results

- Plot of the computational time, against the arrival rate



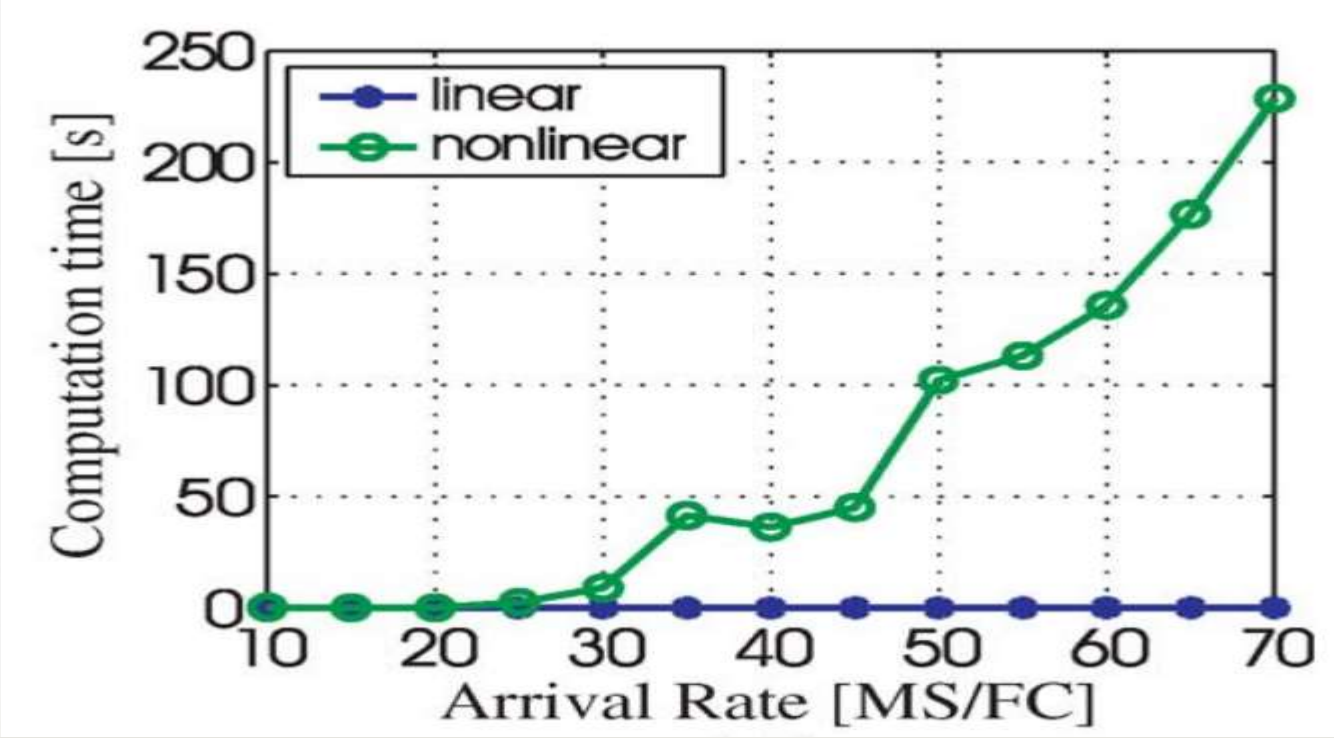
# Results

- Plot of the maximum cycle load Lmax, against the arrival rate(Extended Message Set)



# Results

- Plot of the computational time, against the arrival rate (Extended Message Set)



# Thank You