Selective Real-Time Data Emission in Mobile Intelligent Transport Systems

WMC 2017

Laurent George[1]
Damien Masson[1]
Vincent Nelis[2]

[1]Université Paris-Est, ESIEE Paris, Laboratoire d'informatique Gaspard-Monge (LIGM) UMR CNRS 8049

[2]CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto

Tuesday, December 5th 2017

Outline

1. Use Case Description
2. Open Issues
3. Solution
4. Future Works

Outline

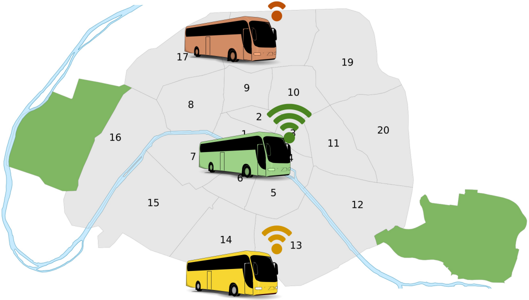
1 Use Case Description

2 Open Issues

3 Solution

4 Future Works

Use Case

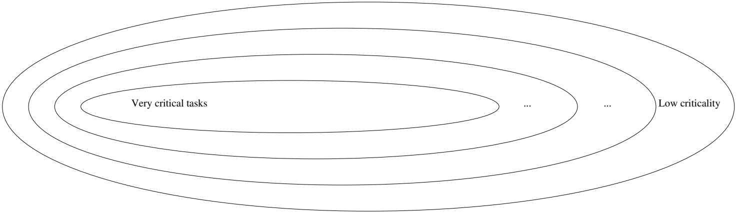


- A vehicle (e.g. a bus) which collects a lot of data
  - speed
  - doors state
  - video recording
  - ...
- Must send the data in real-time to a central station
- But the network speed fluctuates (congestion, link quality etc.)

Use Case

We want to schedule data transmission in order to offer different level of service guarantees according to the data criticality.

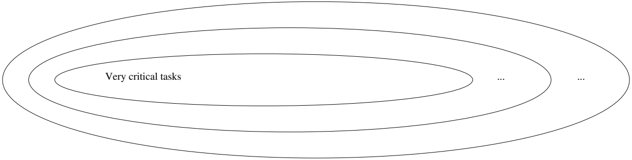
- all the data can be transmitted at a given minimum transmission speed S
- the available transmission speed lowers bellow S:
  - reduce emission frequencies – Future works ?
  - drop less critical tasks



Use Case

We want to schedule data transmission in order to offer different level of service guarantees according to the data criticality.

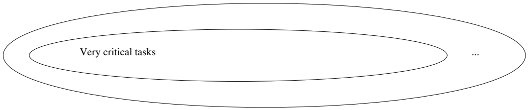
- all the data can be transmitted at a given minimum transmission speed S
- the available transmission speed lowers bellow S:
  - reduce emission frequencies – Future works ?
  - drop less critical tasks



Use Case

We want to schedule data transmission in order to offer different level of service guarantees according to the data criticality.

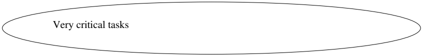
- all the data can be transmitted at a given minimum transmission speed S
- the available transmission speed lowers bellow S:
  - reduce emission frequencies – Future works ?
  - drop less critical tasks



Use Case

We want to schedule data transmission in order to offer different level of service guarantees according to the data criticality.

- all the data can be transmitted at a given minimum transmission speed S
- the available transmission speed lowers bellow S:
    - reduce emission frequencies – Future works ?
    - drop less critical tasks



Outline

1 Use Case Description

2 Open Issues

3 Solution

4 Future Works

Open Issues

- Determine the optimal number of criticality groups
- Determine the optimal priority assigment
- Once criticality groups are formed, determine the minimum speed at which each group can be guarantee (more formally stated in a few slides)
    - Solved in this paper

Open Issues

- Determine the optimal number of criticality groups
- Determine the optimal priority assigment
- Once criticality groups are formed, determine the minimum speed at which each group can be guarantee (more formally stated in a few slides)
  - Solved in this paper

Formally

Sending each kind of data is a non preemptive task characterized by:

- a period T_i,
- a cost C_i which is the necessary time to send the data at a given arbitrary reference speed,
- a relative deadline D_i,
- a criticality level.

Formally

- The system is composed by a set of criticality-uniform tasksets: $\tau(1), \tau(2), ..., \tau(x)$ where x is the number of criticality levels.
- Priorities of tasks in $\tau(i)$ are higher than in $\tau(j)$ when $i < j$ (maybe not necessary)
- Priorities inside a criticality group can follow any fixed priority assignment rule
- The problem is then to determine minimal speeds thresholds $sp_1, sp_2, ..., sp_x$ where at speed sp_l, the union of $\tau(1), \tau(2), ...\tau(l)$ is feasible.

Outline

1 Use Case Description

2 Open Issues

3 Solution

4 Future Works

The formula

For any $sp(x)$ containing n tasks labelyzed $\tau_1, \tau_2, ..., \tau_n$ by priority order,

$$sp(x) = \max_{i\in[1,n]}\left(\max_{j\in[0,n_i-1]}\left(\min_{t\in S_{i,j}}\left(\max\left(\frac{w_{i,j}}{t-\epsilon}, \frac{w_{i,j}+C_i}{jT_i+D_i}\right)\right)\right)\right) \quad (1)$$

where

- n_i is the number of τ_i jobs released in the level-i busy window
- $S_{i,j}$ is the set of time instants between a job release and its deadline at which higher priority tasks release a job, plus its own activation and deadline.

$$S_{i,j} = \bigcup_{k\in \mathrm{hp}(\tau_i)} \left\{rT_k \mid r \in \mathbb{N}^+ \text{ and } jT_i < rT_k < jT_i + D_i\right\} \bigcup \left\{jT_i, jT_i + D_i\right\} \quad (2)$$

The formula

$$sp(x) = \max_{i\in[1,n]}\left(\max_{j\in[0,n_i-1]}\left(\min_{t\in S_{i,j}}\left(\max\left(\frac{w_{i,j}}{t-\epsilon},\frac{w_{i,j}+C_i}{jT_i+D_i}\right)\right)\right)\right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time
  - from the finishing time

The formula

$$sp(x) = \max_{i\in[1,n]}\left(\max_{j\in[0,n_i-1]}\left(\min_{t\in S_{i,j}}\left(\max\left(\frac{w_{i,j}}{t-\epsilon},\frac{w_{i,j}+C_i}{jT_i+D_i}\right)\right)\right)\right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time
  - from the finishing time

The formula

$$sp(x) = \max_{i\in[1,n]}\left(\max_{j\in[0,n_i-1]}\left(\min_{t\in S_{i,j}}\left(\max\left(\frac{w_{i,j}}{t-\epsilon},\frac{w_{i,j}+C_i}{jT_i+D_i}\right)\right)\right)\right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time
  - from the finishing time

The formula

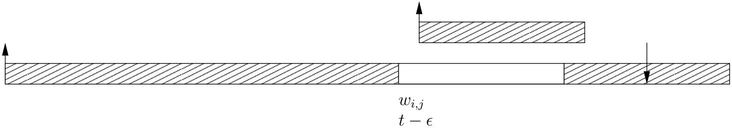
$$sp(x) = \max_{i \in [1,n]} \left(\max_{j \in [0, n_i - 1]} \left(\min_{t \in S_{i,j}} \left(\max \left(\frac{w_{i,j}}{t - \epsilon}, \frac{w_{i,j} + C_i}{jT_i + D_i} \right) \right) \right) \right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time
  - from the finishing time

The formula

$$sp(x) = \max_{i\in[1,n]}\left(\max_{j\in[0,n_i-1]}\left(\min_{t\in S_{i,j}}\left(\max\left(\frac{w_{i,j}}{t-\epsilon},\frac{w_{i,j}+C_i}{jT_i+D_i}\right)\right)\right)\right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time

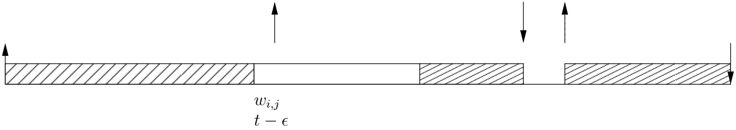


  - from the finishing time

The formula

$$sp(x) = \max_{i\in[1,n]} \left(\max_{j\in[0,n_i-1]} \left(\min_{t\in S_{i,j}} \left(\max \left(\frac{w_{i,j}}{t-\epsilon}, \frac{w_{i,j}+C_i}{jT_i+D_i} \right) \right) \right) \right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time

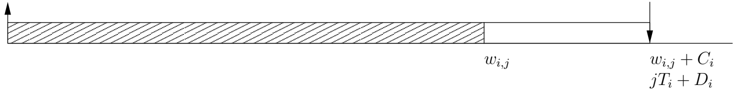


  - from the finishing time

The formula

$$sp(x) = \max_{i \in [1,n]} \left(\max_{j \in [0, n_i - 1]} \left(\min_{t \in S_{i,j}} \left(\max \left(\frac{w_{i,j}}{t - \epsilon}, \frac{w_{i,j} + C_i}{jT_i + D_i} \right) \right) \right) \right) \quad (3)$$

- for a taskset, the minimum speed is the biggest limit that comes by iterate on the tasks
- for a task, we have to iterate on the jobs
- for a job, we look at some key instants
- amongst these instants, the limit come either
  - from the starting time
  - from the finishing time



Outline

1. Use Case Description
2. Open Issues
3. Solution
4. Future Works

Future Works

- Compute the minimal factor we have to apply to the periods that makes a system feasible
- proove that the formula also apply to a non feasible taskset to find the minimal speed that make it feasible (straight forward ?)
- Investigate on how priority assigment can reduce the number of necessary criticality groups
- Compare the deadline missed rate with the setting where we let tasks reach their deadline before dropping them on a real case scenario