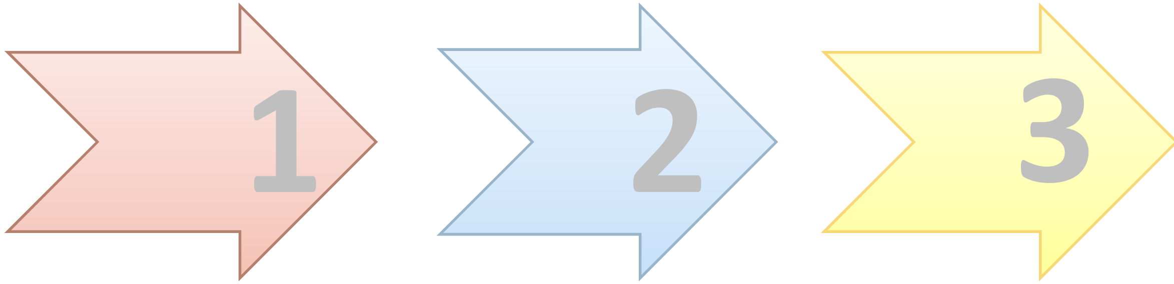
Slotted WiDom: Schedulability Analysis and its Experimental Validation

Maryam Vahabi and Björn Andersson

CISTER Spring Seminar Series 2011

Research Center in
Real-Time Computing System
FCT Research Unit 608

Flow of the talk



- Motivation
- Background on schedulability analysis
- Background on WiDom protocol
- Proposed schedulability analysis for Slotted-WiDom
- Experimental results

Introduction

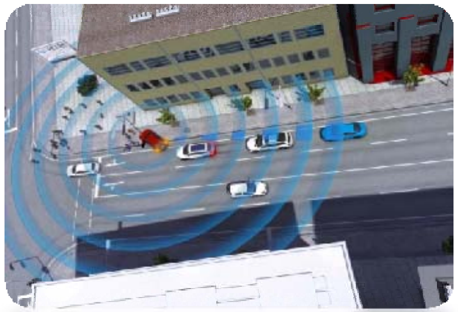
Environmental Monitoring

Collaborative Robatics

Healthcare Monitoring

Industrial Automation

Vehicular Network



Introduction

Real-Time Requirements

Generalized Rate-Monotonic Analysis

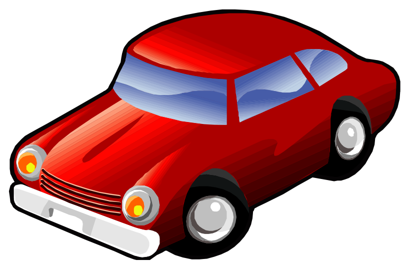
- Uniprocessors
- Wired networks

- Wireless networks

Introduction

RM Analysis Requirements for ~~Wireless~~ :

- R1. Prioritized medium access control (MAC)
- R2. Slow growth in arbitration overhead (by increasing priority levels)
- R3. Low arbitration overhead
- R4. Provide schedulability analysis



CAN-bus

Dominance / Binary-countdown protocol

Designed for *wired domain*

Introduction

RM Analysis Requirements for **Wireless** :

- ✔ R1. Prioritized medium access control (MAC)
- ✔ R2. Slow growth in arbitration overhead (by increasing priority levels)
- ✘ R3. Low arbitration overhead
- ✔ R4. Provide schedulability analysis

WiDOM

Wireless version of CAN bus

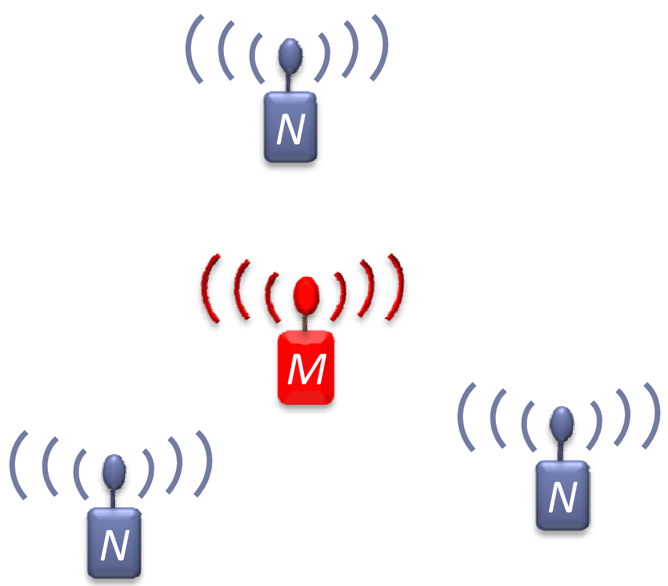
Introduction

RM Analysis Requirements for **Wireless** :

- ✓ R1. Prioritized medium access control (MAC)
- ✓ R2. Slow growth in arbitration overhead (by increasing priority levels)
- ✓ R3. Low arbitration overhead
- ✗ R4. Provide schedulability analysis

Slotted-WiDOM

Out-of-band Synch. Signal



Motivation

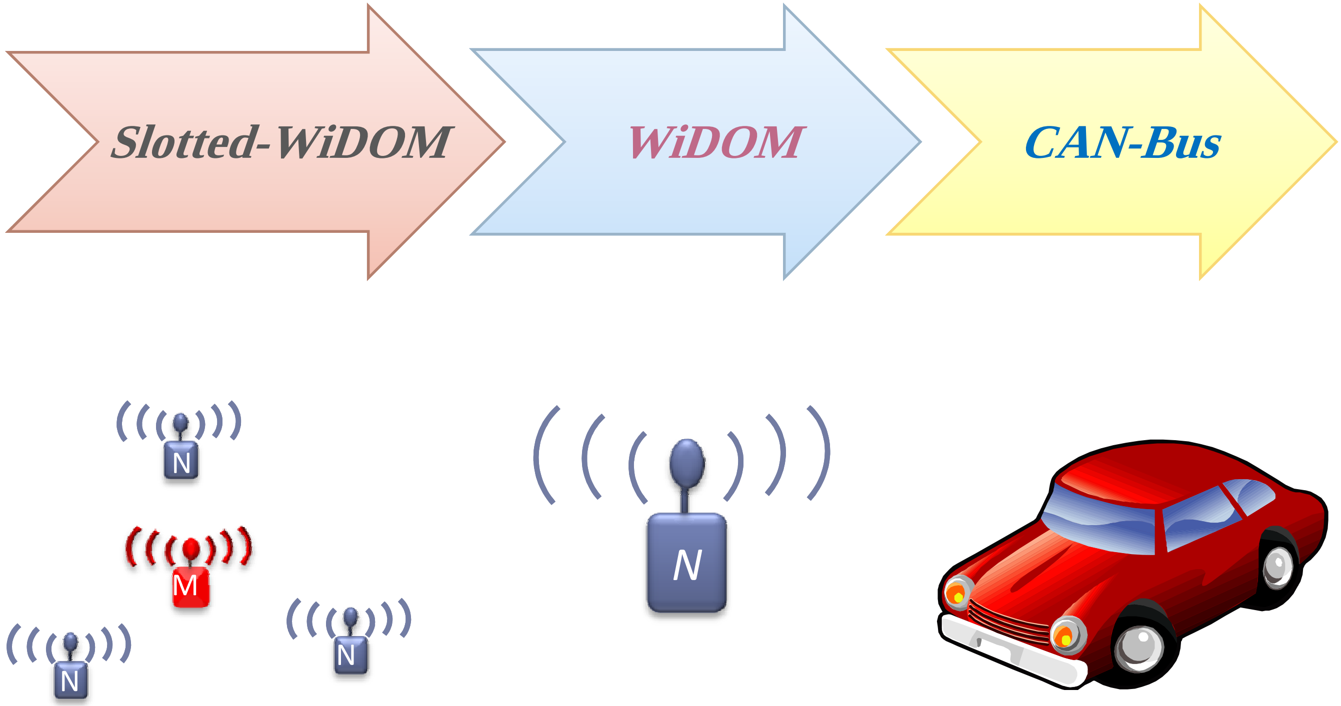
RM Analysis Requirements for **Wireless** :

- ✓ R1. Prioritized medium access control (MAC)
- ✓ R2. Slow growth in arbitration overhead (by
- ✓ R3. Low arbitration overhead
- ✓ **R4. Provide schedulability analysis**

Slotted-WiDOM

Out-of-band Synch. Signal

Background on schedulability analysis



Schedulability analysis of *CAN-bus*

- First proposal (1994)

preemptive static-priority scheduling

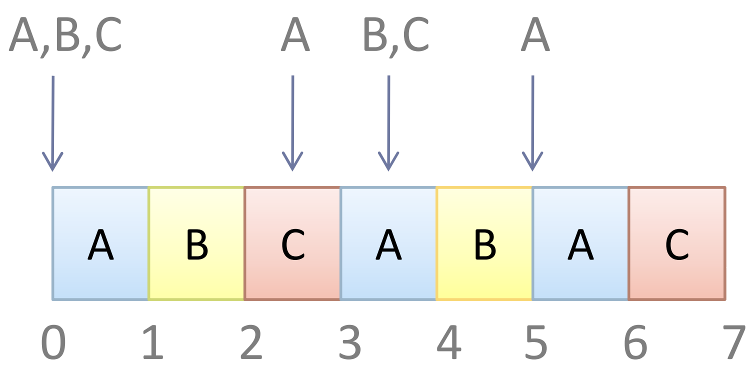
↓ Non-preemptive ↓

Primitive CAN-bus Analysis ✗

- First Correct proposal (2007)

Busy period of message *i* > period of message *i*

Schedulability analysis of *CAN-bus*



$C_A = C_B = C_C = 1$

$D_A = D_B = D_C = 3$

$T_A = 2.5$

$T_B = 3.5$

$T_C = 3.5$

- First Correct proposal (2007)

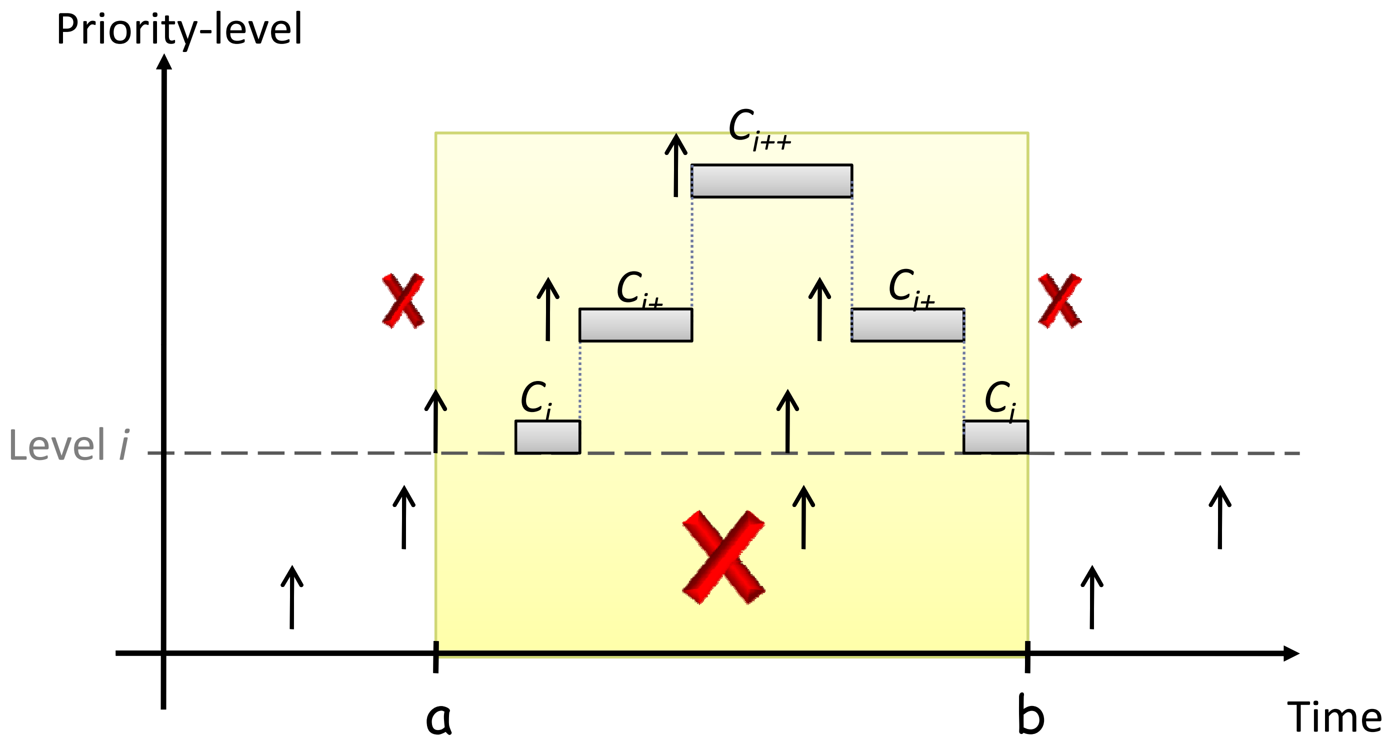
Busy period of message *i* > period of message *i*

Schedulability analysis of *CAN-bus*

Busy period:

Lehoczky (1990)

A level-*i* busy period is a time interval [a,b] within which jobs of priority *i* or higher are processed through [a,b] but no jobs of level *i* or higher are processed in (a-ε,a) or (b,b+ ε) for sufficiently small ε>0.



Research Center in Real-Time Computing System
FCT Research Unit 608

Schedulability analysis of *CAN-bus*

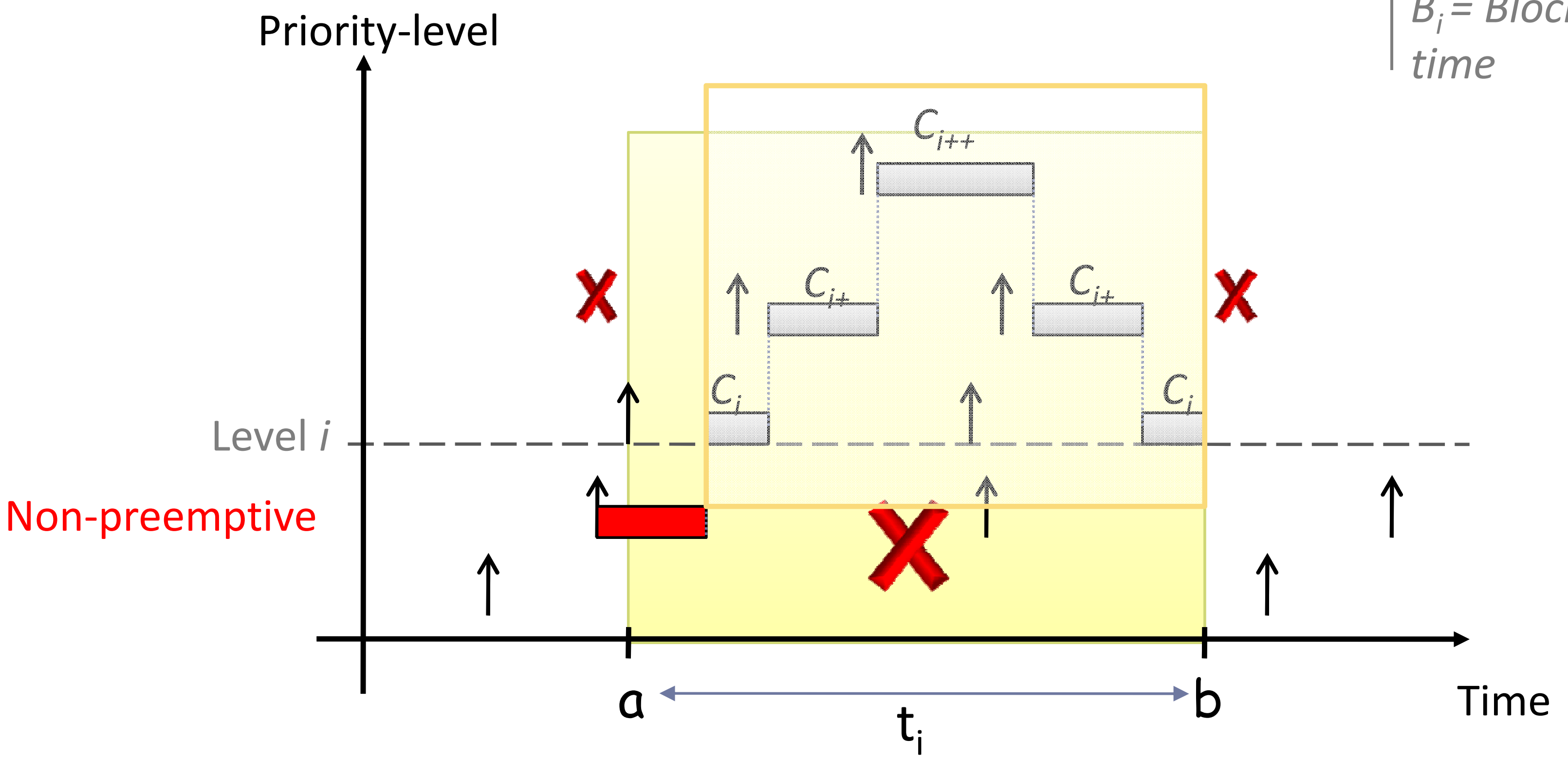
Busy period:

$$t_i = B_i + \sum_{\forall m \in hp(i) \cup i} \left\lceil \frac{t_i}{T_m} \right\rceil \times C_m$$

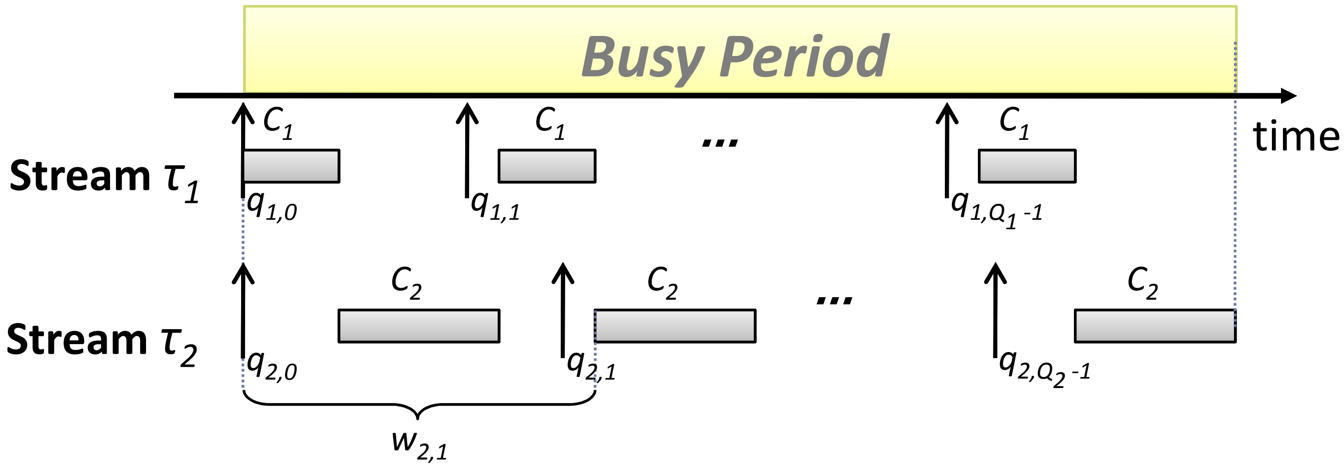
C_m = Transmission time

T_m = Message Period

B_i = Blocking time



Schedulability analysis of *CAN-bus*



$$R_i = \max_{q=0,\ldots,Q_i-1} (R_{i,q})$$

$$Q_i = \left\lceil \frac{t_i}{T_i} \right\rceil$$

$$R_{i,q} = w_{i,q} - q \times T_i + C_i$$

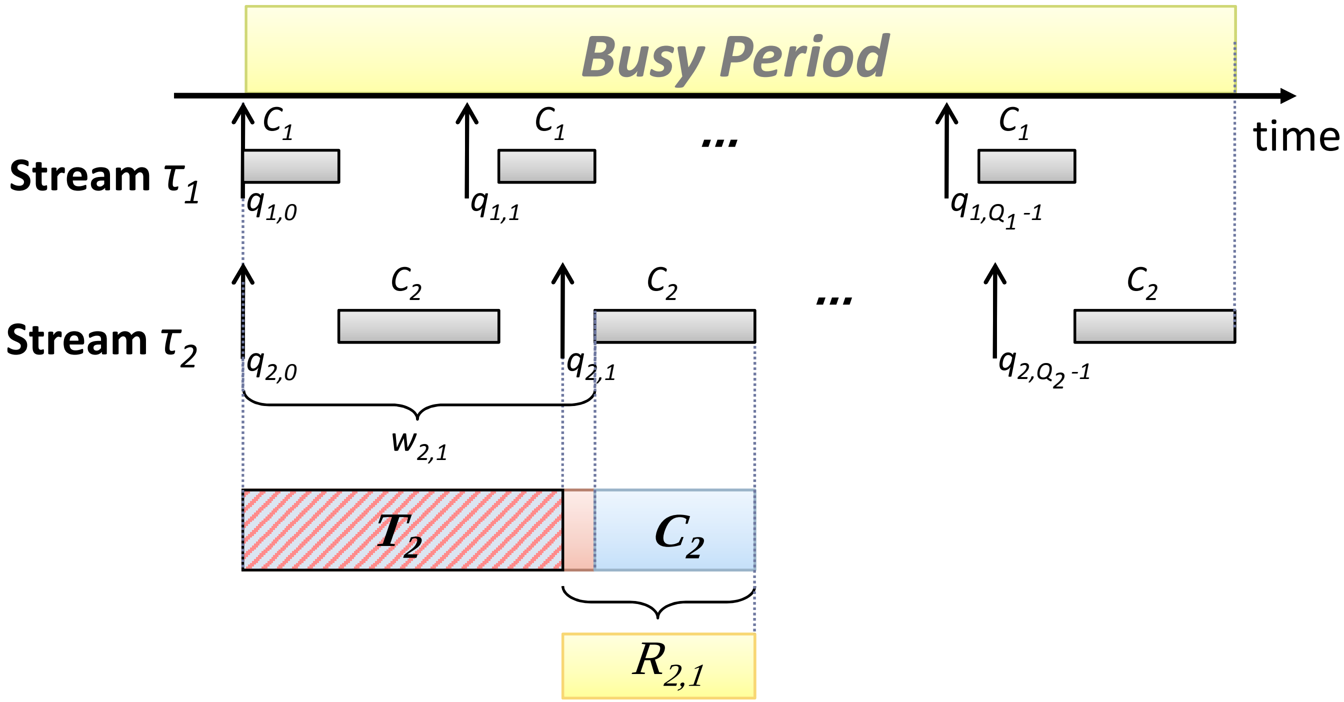
R_i = WCRT of message stream i

Q_i = No. of instances of message stream i located in level-i busy period

T_i = Periodicity of message stream i

$w_{i,q}$ = ?

Schedulability analysis of *CAN-bus*



$$R_{2,1} = w_{2,1} - 1 x T_2 + C_2$$

Background on WiDom protocol

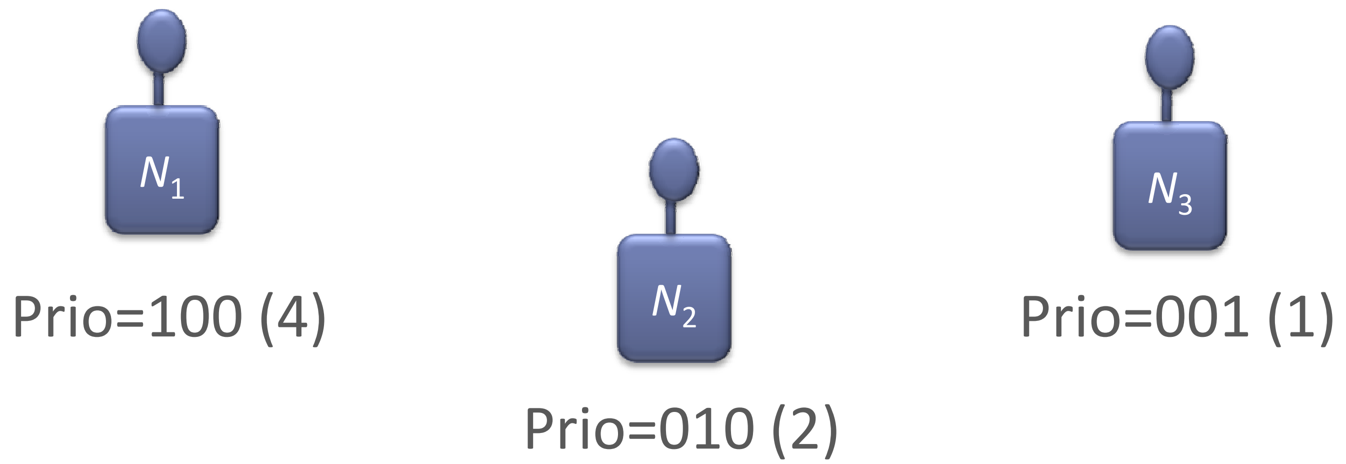
Dominance / Binary-countdown protocol

- Conflict-free tx by exploiting *bitwise arbitration*
- Provides large number of priority levels
- *Lower* value → *higher* priority
- Three phases:
  - Synchronization phase
  - Tournament phase
  - Receive/transmit phase

Tournament phase (an example)

0: Dominant bit

1: recessive bit



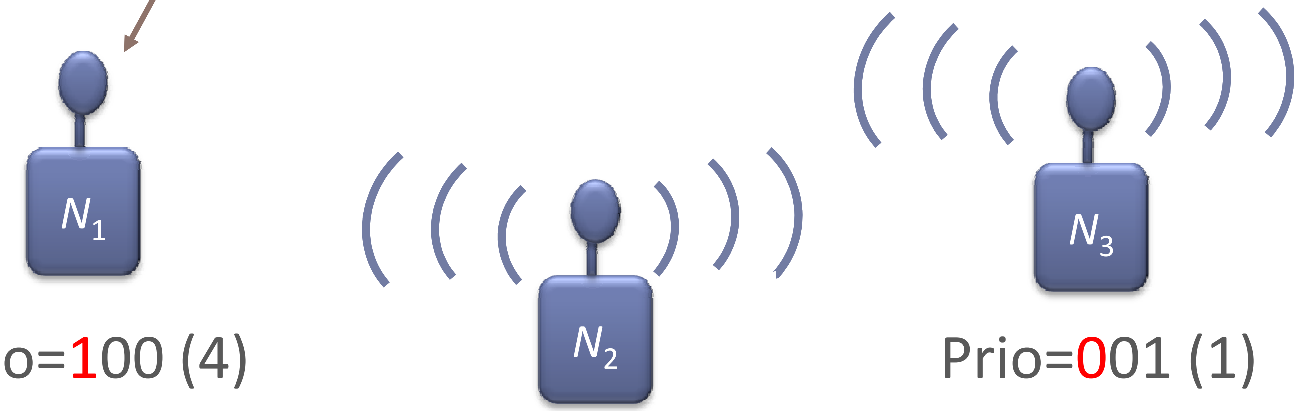
Tournament phase

0: Dominant bit

1: recessive bit

N_1 detects a dominant bit.

N_1 loses the arbitration.



Prio=100 (4)

Prio=010 (2)

Prio=001 (1)

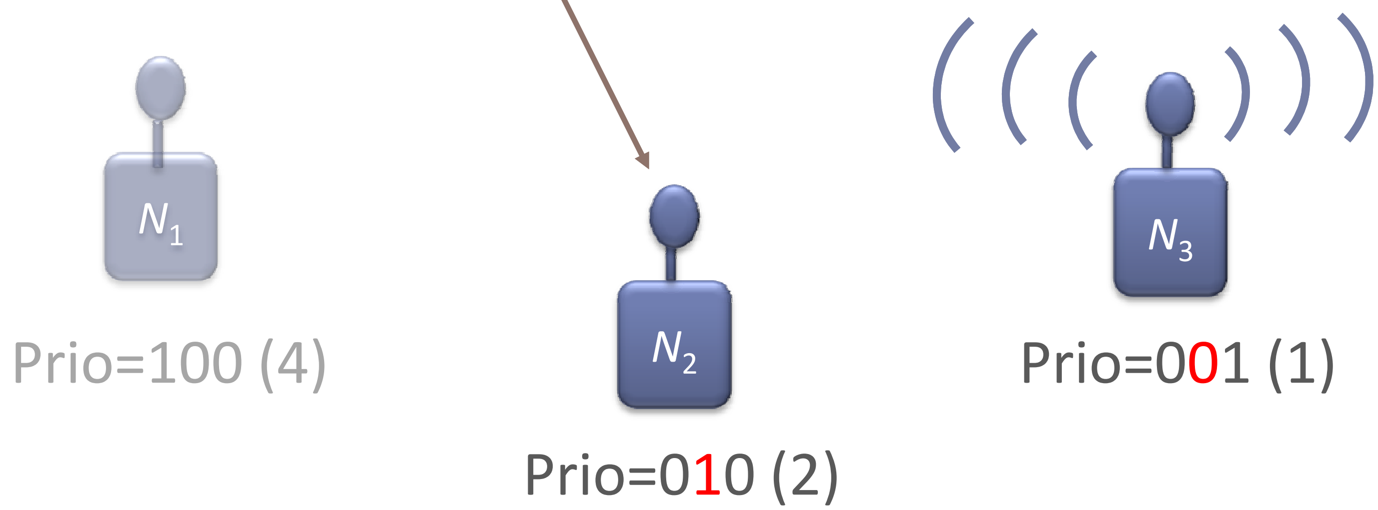
Node with *Dominant* bit, transmit a carrier

Nodes with *recessive* bit, listen

Tournament phase

N_2 detects a dominant bit.

N_2 loses the arbitration.



Node with *Dominant* bit, transmit a carrier

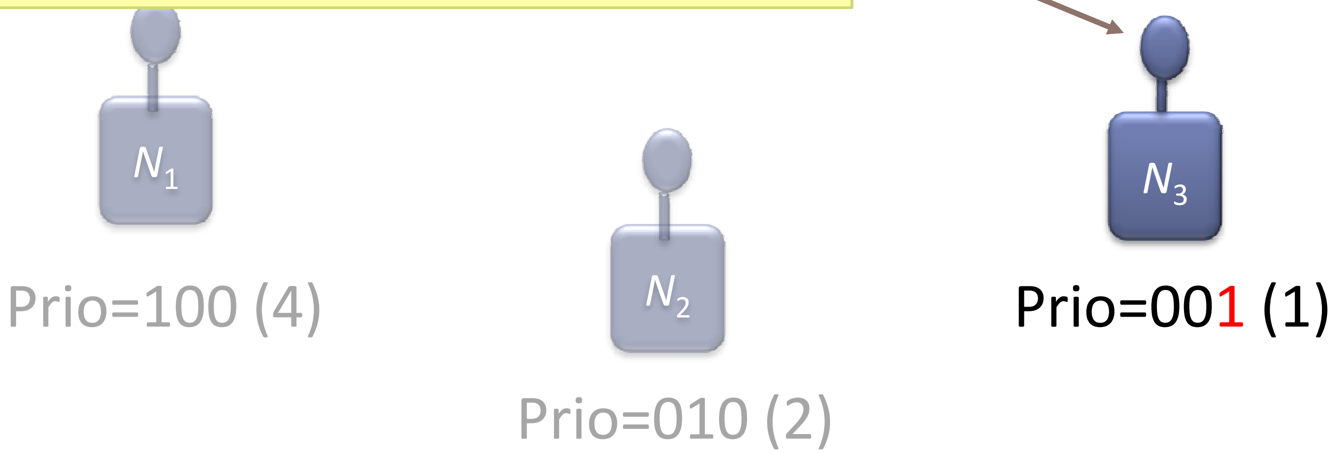
Nodes with *recessive* bit, listen

Tournament phase

N_3 finishes transmitting the priority bits.

N_3 reaches the end of the arbitration…

…and proceeds to transmit the message.



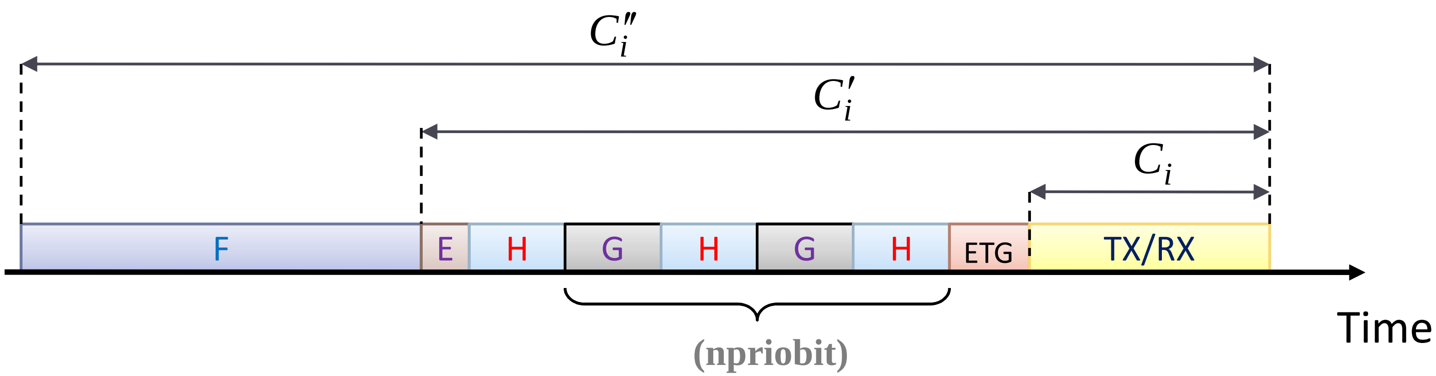
Node with *Dominant* bit, transmit a carrier

Nodes with *recessive* bit, listen

WiDom synchronization

Initial WiDom

Long period of silence : F



E: clock drift compensation
H: carrier pulse transmission
G: guarding time
ETG: end of tournament gap
npriobit: No of priority bits

WiDom synchronization

Slotted WiDom

Out-of-band signalling

WiFLEX add-on board

- WiFLEX_main
- WiFLEX_rxsync

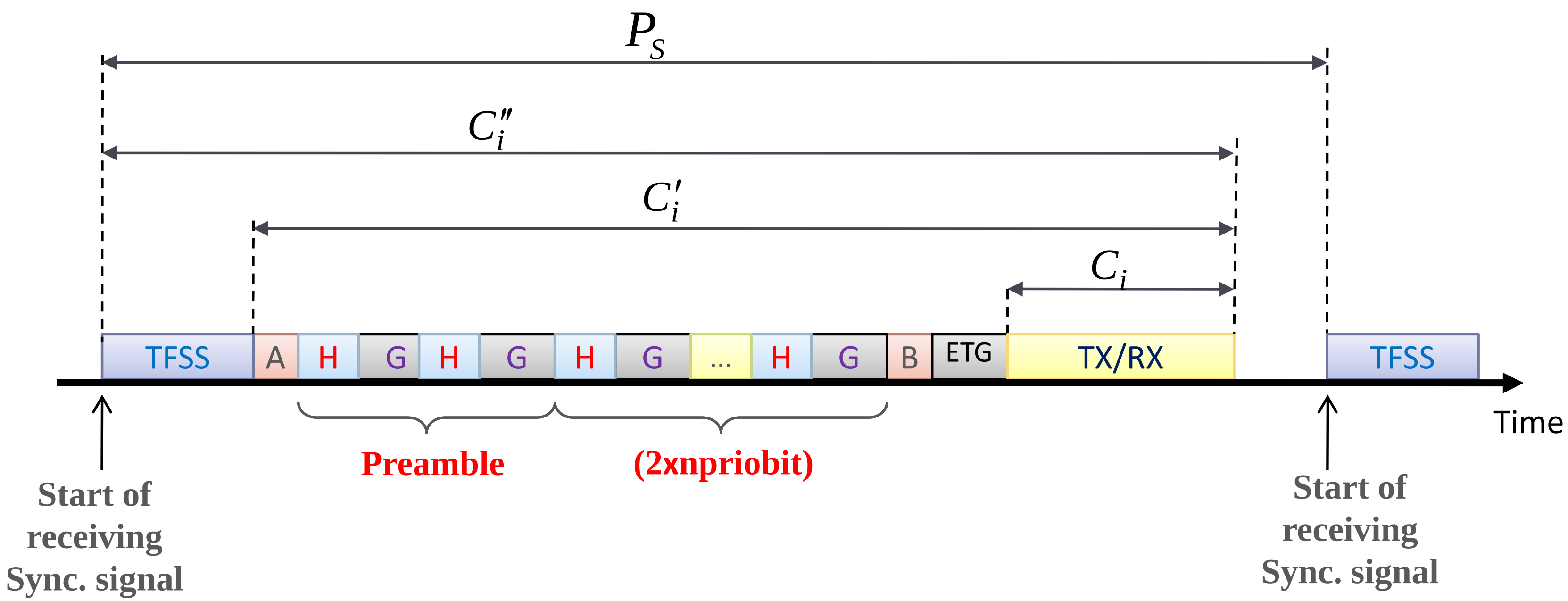
WiFLEX platform
stacked on FireFly mote

WiFLEX platform
stacked on MICAz mote

WiDom synchronization

Slotted WiDom

Out-of-band signalling



A: Transferring priority from MicaZ to WiFLEX (PRIO_TRA)

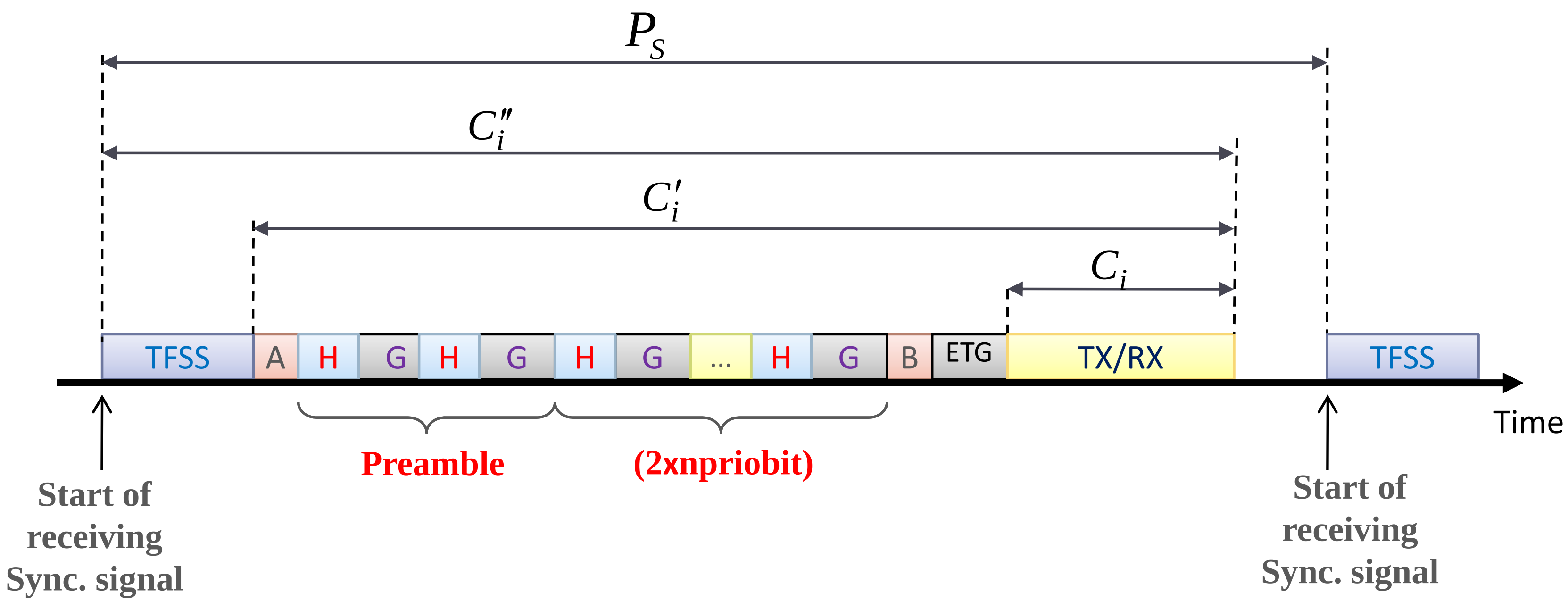
B: Transferring winner priority from WiFLEX to MicaZ (WIN_PRIO)

P_S: sync. Signal periodicity / Slot duration

WiDom synchronization

Slotted WiDom

Out-of-band signalling



$$P_S \geq TFCS + PRIO\_TRA + 2(H+G)(npriobits+1) + ETG + WIN\_PRIO + \max(C_i)$$

New schedulability analysis

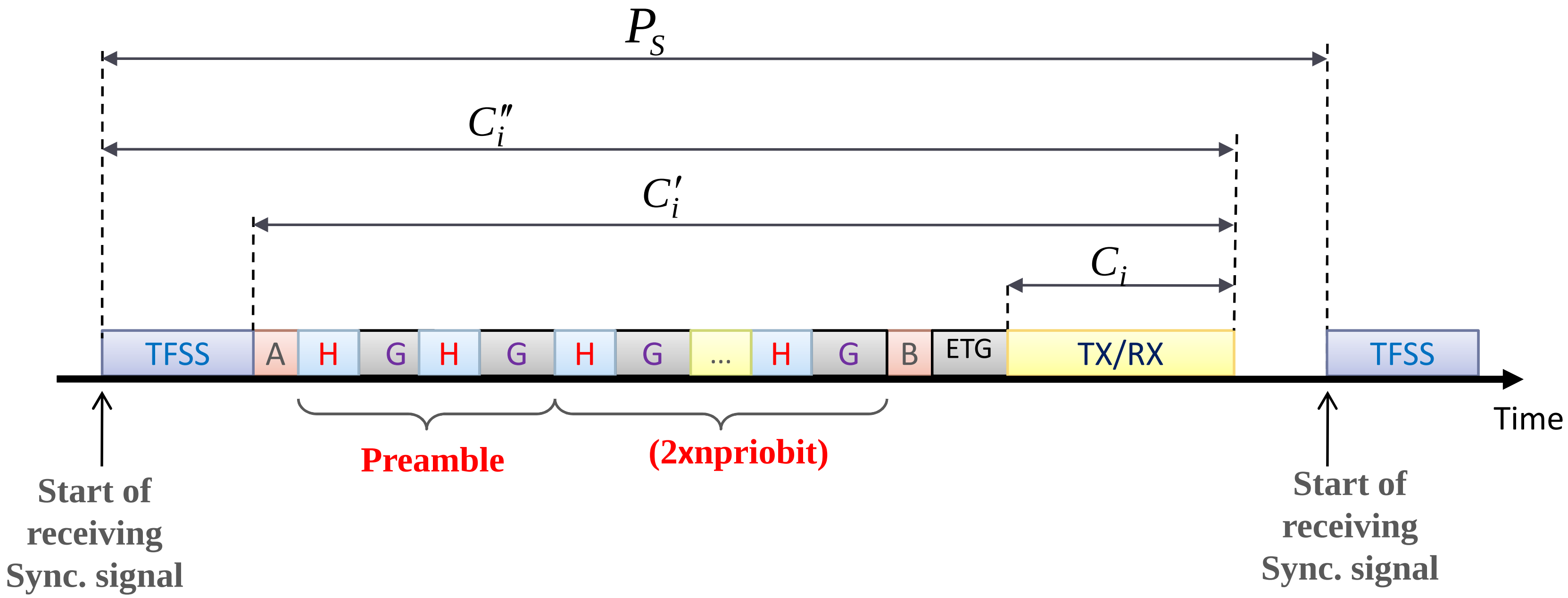
One packet per slot

- Level-*i* busy period .... $L_i = P_S + \sum_{j \in hp(i) \cup i} \left\lceil \frac{L_i + J_j}{T_j} \right\rceil \times P_S$

- Finding no of instances .... $Q_i = \left\lfloor \frac{L_i + J_i}{T_i} \right\rfloor + 1$

- Finding WCRT .... $R_i = \max_{q=0 \ldots Q_i - 1} \left(w_{i,q} + J_i + C_i'' - q \times T_i \right)$

New Schedulability analysis



$$C_i'' = C_i' + TFCS$$

$$C_i' = C_i + 2(H+G) + PRIO\_TRA + 2(H+G)(npriobits) + WIN\_PRIO + ETG$$

Analytical results

- Packet length .... 128 Bytes
- Data rate .... 250 Kb/s

$$\forall i \in \{1 \ldots n\} : C_i = 128 \times 8 \times \frac{1}{250000} = 4096\ \mu s$$

$$\forall i \in \{1 \ldots n\} : C_i' = 8545\ \mu s$$

$$\forall i \in \{1 \ldots n\} : C_i'' = 8845\ \mu s$$

$$P_S \geq 8845\,\mu s$$

$$P_S = 9560\,\mu s$$

(npriobits) 15

(H+G) .... 110 µs

(TFSS) .... 300 µs

(PRIO_TRA) .... 139 µs

(WIN_PRIO) .... 235 µs

Analytical result (first scenario)

- Packet length .... 128 Bytes
- Data rate .... 250 Kb/s
- Release jitter .... 1 ms
- No of Nodes 6

i	1	2	3	4	5	6
T_i (μs)	30,000	80,000	150,000	300,000	700,000	1,800,000
R_i (μs)	18,405	27,965	37,525	56,645	66,205	85,325

$$\forall i \in \{1 \ldots n\} : R_i \leq T_i$$

Analytical result (second scenario)

- Packet length .... 128 Bytes
- Data rate .... 250 Kb/s
- Release jitter .... 1 ms
- No of Nodes 10

i	1	2	3	4	5
T_i (μs)	30,000	70,000	120,000	300,000	900,000
R_i (μs)	18,405	27,965	37,525	56,645	66,205

i	6	7	8	9	10
T_i (μs)	1,900,000	3,700,000	5,400,000	5,400,000	5,400,000
R_i (μs)	94,885	114,005	123,565	171,365	180,925

$$\forall i \in \{1 \ldots n\} : R_i \leq T_i$$

Experimental setup

- Hardware platform .... MICAz + WiFLEX

- Software .... Nano-RK OS + WiDom

```
Send-Task(){
      generated-packet++;
}

send_pkt(){
      transmitted-packet++;
}
```

generated-packet

≠

transmitted-packet

Deadline miss

Experimental setup

- Hardware platform .... MICAz + WiFLEX

- Software .... Nano-RK OS + WiDom

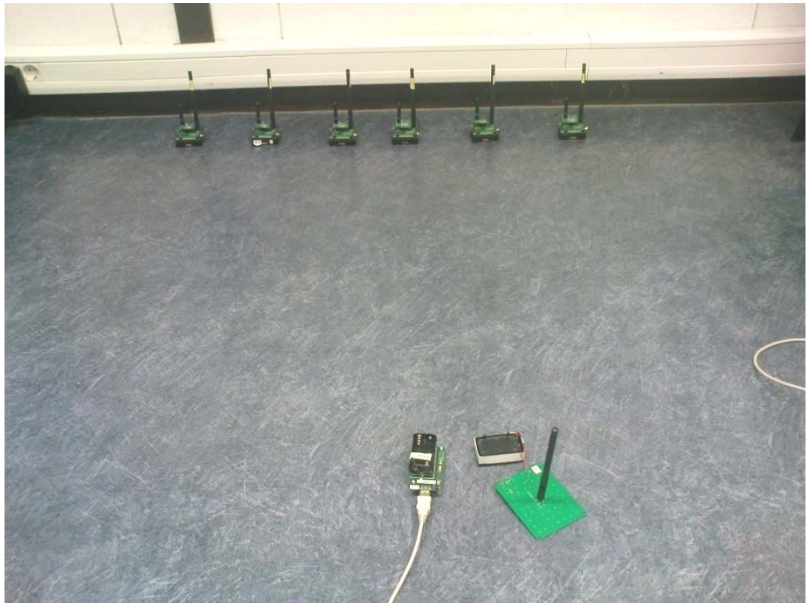
```
Send-Task(){
     Set Timer;
}

rf_tx_packet(){
    Wi=Read Timer;
    pPayload=Wi;
}
```

Receiver Side:

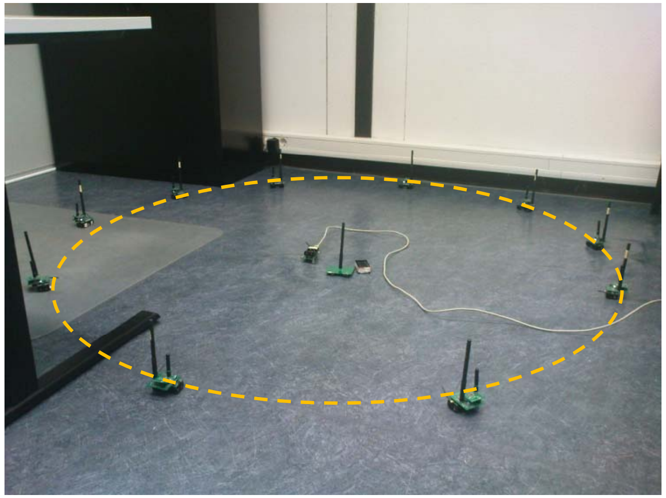
$$R_i=W_i+C_i$$

Experimental result (first Scenario)



i	1	2	3	4	5	6
T_i (μs)	30,000	80,000	150,000	300,000	700,000	1,800,000
R_i (μs)	18,405	27,965	37,525	56,645	66,205	85,325
R'_i (μs)	18,348	27,583	37,128	55,982	59,184	64,834

Experimental result



i	1	2	3	4	5
T_i (μs)	30,000	70,000	120,000	300,000	900,000
R_i (μs)	18,405	27,965	37,525	56,645	66,205
R'_i (μs)	18,343	27,584	37,147	56,225	55,428

i	6	7	8	9	10
T_i (μs)	1,900,000	3,700,000	5,400,000	5,400,000	5,400,000
R_i (μs)	94,885	114,005	123,565	171,365	180,925
R'_i (μs)	58,019	39,509	62,490	34,464	62,403

Experimental result

- No deadline miss

  generated-Packet == transmitted-packet

- Very small packet loss rate (≤ 1%)

- Validate the calculated upperbound

  Calculated RT ≥ Measured RT

Next steps

- Implementing Reliable WiDom
- Provide a real-time constraint-free model for WiDom
- Find appropriate priority relationship assignment for multi-hop message streams

Thank You!