Modeling Real-Time Networks with MAST2

WATERS 2011

July 2011, Porto, Portugal


mgh@unican.es
www ctr unican es
Modeling Real-Time Networks with MAST2

1. Introduction
2. Overview of the MAST2 model
3. Networks
4. Network switches and routers
5. Modelling AFDX networks
6. Conclusions
1. Introduction

MAST (Modeling and Analysis Suite for Real-Time Applications)
- defines a model to describe the timing behavior of real-time systems
- model is analyzable via schedulability analysis techniques
- sensitivity analysis
- assignment of scheduling parameters
- support for different scheduling policies

The MAST model is very similar to the model defined in MARTE:
- the standardized UML profile for real-time embedded systems
Some of the new elements being defined in MAST2 are network switches and routers

- IEEE 802.1p switches with prioritized traffic
- The Avionics Full-Duplex Switched Ethernet (AFDX) defines a hard real-time network based on switched ethernet
Communications network defined in the ARINC-644, Part 7 standard

- based on the use of point to point full duplex ethernet links
- and special purpose switches

The routing of messages is preconfigured

- there is no delay in the discovery of routing addresses

Two redundant hardware communication channels

The switches can filter erroneous traffic

The switches have two FIFO queues at each output port

- high and low priority traffic
Virtual Links

Traffic regulation at the sending end via *Virtual Links*

- communication objects with a dedicated maximum bandwidth

Each virtual link (VL) has two parameters

- **$L_{\text{max}}$:** the largest Ethernet frame in bytes
- **$BAG$:** the Bandwidth Allocation Gap ($BAG$)
  - a power of the value 2 in the range $[1,128]$ ms
  - the $BAG$ is the minimum interval between Ethernet frames transmitted on the VL

Each virtual link has a FIFO queue for all the fragmented packets to be transmitted through it

The priority is configured on a VL basis
2. Overview of the MAST2 model

Software Modules

- Mutual Exclusion Resources

Operation

- Step

Platform

- Processing Resource

- Scheduler

- Schedulable Resource

Schedulable Entities

- Scheduling Parameters

End-to-end flow

Real-time situation

- Event

- Event Handler

- Timing Requirement

- Reference
A simple distributed example

CPU1  CAN Bus  CPU2
Sender Task  Data Message  Receiver Task
Platform View

CPU1: Regular_Processor
  Speed_Factor = 1.0
  CPU1_Sch: Primary_Scheduler
    Policy = Fixed_Priority_Policy
  Host

CAN_Bus: Packet_Based_Network
  Throughput = 1.0e6
  CAN_Bus_Sch: Primary_Scheduler
    Policy = Packet_Based_FP_Policy
  Host

CPU2: Regular_Processor
  Speed_Factor = 1.0
  CPU2_Sch: Primary_Scheduler
    Policy = Fixed_Priority_Policy
  Host
Concurrent architecture view

CPU1_Sch: Primary_Scheduler
Policy= Fixed_Priority_Policy

CPU2_Sch: Primary_Scheduler
Policy= Fixed_Priority_Policy

CAN_Bus_Sch: Primary_Scheduler
Policy= Packet_Based_FP_Policy

Sender: Thread
... Scheduler
Sender: Thread
... Scheduler
Sender: Thread
... Scheduler

Msg_Stream: Comm_Channel
... Scheduler
Msg_Stream: Comm_Channel
... Scheduler
Msg_Stream: Comm_Channel
... Scheduler

Receiver: Thread
... Scheduler
Receiver: Thread
... Scheduler
Receiver: Thread
... Scheduler

SP: Fixed_Priority_Params
Priority = 12

M: Fixed_Priority_Params
Priority = 1200

RP: Fixed_Priority_Params
Priority = 14
## Operations view

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Send:</strong> Simple Operation</td>
<td><strong>Data:</strong> Message</td>
<td><strong>Receive:</strong> Simple Operation</td>
</tr>
<tr>
<td>WCET = 1.23e-3</td>
<td>Max_Message_Size = 1600</td>
<td>WCET = 1.23e-3</td>
</tr>
</tbody>
</table>
Real-Time Situation view

Sender: Thread

Msg_Stream: Comm_Channel

Receiver: Thread

Step_Schedulable_Resource

E1: Periodic_Event
Period = 1.0e-2

Sendr: Step
Input_Event
Output_Event

DataStep: Step
Input_Event
Output_Event

Recvr: Step
Input_Event
Output_Event

Step_Operation

Send: Simple_Operation

Data: Message

Receive: Simple_Operation

TR: Hard_Global_Deadline
Deadline = 0.8e-2

Observer/List
3. Networks

- Processing_Resource
  - Speed_Factor
  - Throughput
  - Synchronization_Source
    - Clock_Synchronization_Object
  - Driver_List
    - Driver
  - RTP_E_Network
    - 
    - Max_Blocking
4. Network switches and routers

- **Processing_Resource**
  - Speed_Factor

- **Network_Switch**
  - Max_Fixed_Fork_Latency
  - Max_Variable_Fork_Latency
  - Max_Delivery_Latency

- **Network_Router**
  - Max_Fixed_Branch_Latency
  - Max_Variable_Branch_Latency

- **Clock_Synchronization_Object**
  - Synchronization_Source

- **Regular_Switch**
  - ...

- **AFDX_Switch**
  - ...

Also Min, Avg
Message Delivery Event Handlers

Event_Handler

Network_Switch

Message_Event_Handler

Message_Delivery

Message_Fork

Event

Scheduling_Parameters

Input_Event

Output_Event

Output_Event_List

Sched_Params

Switch

1

1

1
5. Modelling AFDX networks

- Network
  + Throughput
  - Driver List

- Packet-Based Network

- AFDX Link
  - Max_Packet_Size
  - Min_Packet_Size
  - Max_HW_Tx_Latency
  - Max_W_Rx_Latency
  - Ethernet_Overhead
  - Protocol_Overhead

Also Min, Avg
AFDX Virtual Links

- **Scheduling_Parameters**
  - ...

- **AFDX_Virtual_Link**
  - Lmax : Bit_Count
  - BAG : Time
A Simple Example

Sender Task → CAN Bus → Receiver Task

Sender Task → Ethernet → Switch → Ethernet → Receiver Task
Concurrent architecture view

Link1_Sch: Primary_Scheduler
Policy = AFDX_Policy

Stream1: Communication_Channel

M1: AFDX_Virtual_Link
Lmax = 1000
BAG = 4.0-3

Scheduler

Link2_Sch: Primary_Scheduler
Policy = AFDX_Policy

Stream2: Communication_Channel

M2: Fixed_Priority_Params
Priority = 2
Operations view

Data: Message

Max_Message_Size = 1600
Real-Time Situation view

Stream1: Communication_Channel

Step_Schedulable_Resource

E2: Internal_Event...

S1: Step
Input_Event Output_Event

Step_Operation

MD: Message_Delivery
Input_Event Output_Event

Data: Message

Switch: AFDX_Switch

E22: Internal_Event...

S2: Step
Input_Event Output_Event

Step_Operation

E23: Internal_Event...

S3: Step
Input_Event Output_Event

Step_Operation

Stream2: Communication_Channel

Step_Schedulable_Resource

E3: Internal_Event...

© Michael González Harbour
5/Jul/11
Alternatives for prioritized switch

a) Add the priority to the message event handler

b) Add the priority to the output communication channel
6. Conclusions

We have proposed new modelling elements to support

- network switches
- network routers
- AFDX real-time networks

These elements:

- are being implemented in MAST2 together with their associated analysis techniques
- will be proposed for a future version of the MARTE UML profile for real-time embedded systems