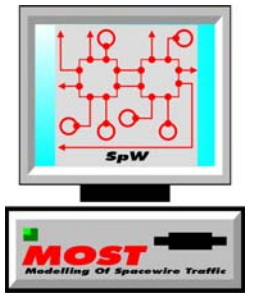
scisys 4Links ThalesAlenia Space
A Thales / Finmeccanica Company



MOST presentation

SpaceWire Working group
14th of December 2011
At ESTEC

THALES

Agenda

- **Context of the study**
- **Study presentation and objectives**
  - People involved in the study
  - MOST tool, a support for all project phases
- **MOST major interests**
- **SpaceWire protocol coverage**
- **A tool based on OPNET Modeler 16.0**
  - Benefits and statistics available
- **SpaceWire building block overview**
  - Processes involved
- **MOST library description**
  - Generic nodes
  - RMAP protocol and CCSDS PTP
  - SpW 10X router
  - SMCS116SPW
  - Remote Terminal Controller
- **MOST utilization**
- **MOST demonstration and validation**
  - Study case 1 : Earth observation mission
  - Study case 2 : Robotic mission
- **Perspectives**
- **Conclusion**

Context of the study: What is MOST?

- **In the frame of SpaceWire technology standardisation and development, ESA supports the need for a SpaceWire network simulation:**
  - To support SpW network design and optimization
  - To allow SpW networks performances analyse from the beginning, without waiting for system testing phase
  - To offer a progressive tool for SpW experts who would like to integrate specific SpW components, or, to update existing library with regard to standard upgrades
- **The ESA study "SpW Traffic simulator" objective is to provide such a simulating tool called *MOST* (Modeling of SpaceWire Traffic)**
- **As a background for this study, MOST was initially developed in year 2006 by TAS-F and was based on OPNET toolkit 12.0**

Study objectives

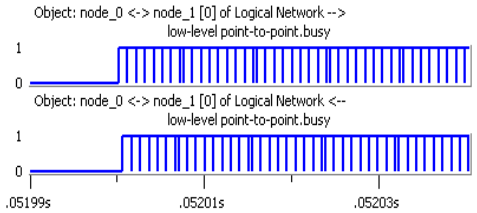
- **<u>Main objectives :</u> Specification, implementation and validation of MOST**

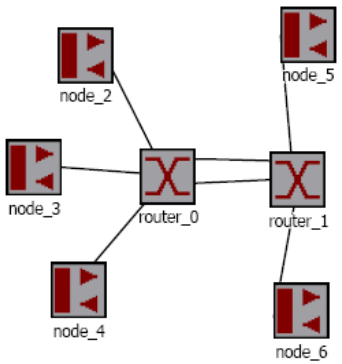
- **The study was built on five main steps:**
  - The specification of the simulator itself
  - The simulator development
  - The Building Block validation, based on simple test cases
  - The demonstration based on two representative study cases
  - The simulator delivery and installation

- **People involved in the study:**
  - ESA:
    - David Jameux
  - 4Links:
    - Barry Cook
    - Paul Walker
  - Scisys:
    - Peter Mendham
    - Stuart Fowell
  - TAS-F Cannes:
    - Emmanuelle Liebgott (replaced by Brice Dellandrea)
    - Philippe Fourtier
    - Loic Parent

A support all along the project life

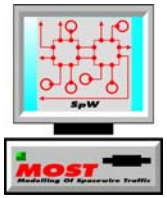
- **MOST simulator is dedicated to the following users:**
  - System engineers who have to design network topology and to perform validation tests
  - Developers who would need to test new component features or protocol





MOST can be used during all phases of a project:

- **During early steps of projects, MOST mainly plays a part in the following design activities :**
  - **Phase A and before : performs evaluations, starting from a preliminary specification of network and nodes**
  - **Phase B : consolidate design by enhancing and completing nodes models behavior in terms of data provider and consumer**
- **During development steps of a project, MOST participates to :**
  - **Phase C, D : design, validation and investigation**
- **During maintenance step of a project, MOST takes part to :**
  - **Phase E : investigations, support to very specific operations**



MOST major interests

- It offers the possibility to build SpW network model, and, to test defined design without waiting for HW,
- It offers a large choice of representative SpW components, with set of attached attributes for configuration purpose,
- It allows to keep control on traffic load and identify weak parts of the network topology,
- It gives load margins and traffic performances,
- It allows failures simulation and give the possibility to run various scenarios
- MOST supports design risks, decreases and secures planning thanks to early verification,
- In addition, MOST is a progressive tool which allows testing of SpW standard evolutions.

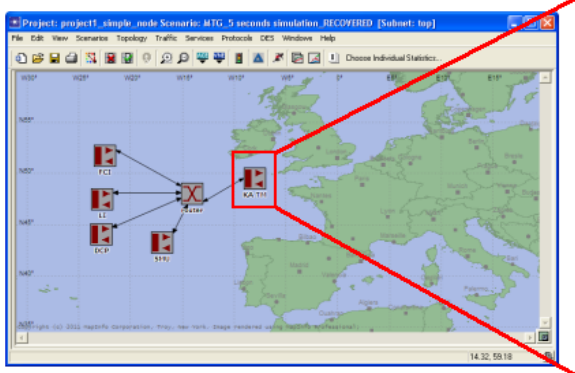
SpaceWire protocol coverage

- **SpaceWire – Links, Nodes, Routers & Networks [ECSS-E-ST-50-12C, 31st July 2008]**
- **Handle sequence of characters, representative of characters length:**
  - Data characters
  - Control characters (FCT, EOP, EEP)
  - Control Codes (NULL, Time code)
- **Sequence of initialization, disconnection/reconnection**
- **Apply the control flow mechanism, on packet emission & reception**
- **Packets have the structure as stated in ECSS ST-50-12C :**
  - Destination address - Cargo - EOP
- **Handle the following anomaly cases:**
  - Source node and destination node stalled
  - Elephant message
  - Erroneous packet
- **Packet addressing:**
  - Logical addressing
  - Physical addressing
- **Time code**

OPNET Modeler 16.0

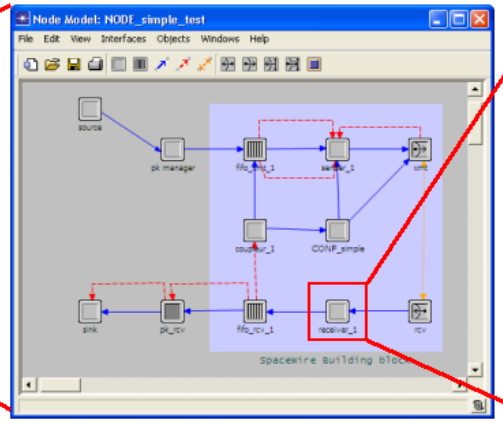
- **MOST simulator is based on OPNET Modeler 16.0**
- **OPen NETwork Modeler**

▪ Finite State Machine
▪ C-Code in each state



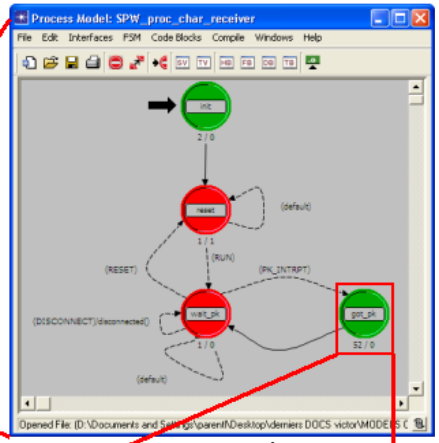
Project or Network domain

▪ Design of the network
▪ Drag & drop from library
▪ Attributes attached to each node are determined

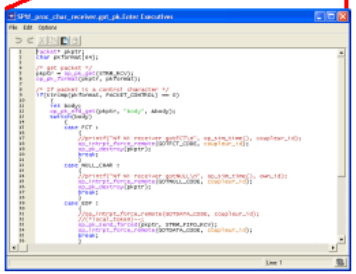


Node domain

▪ Node architecture
▪ Modules interconnection

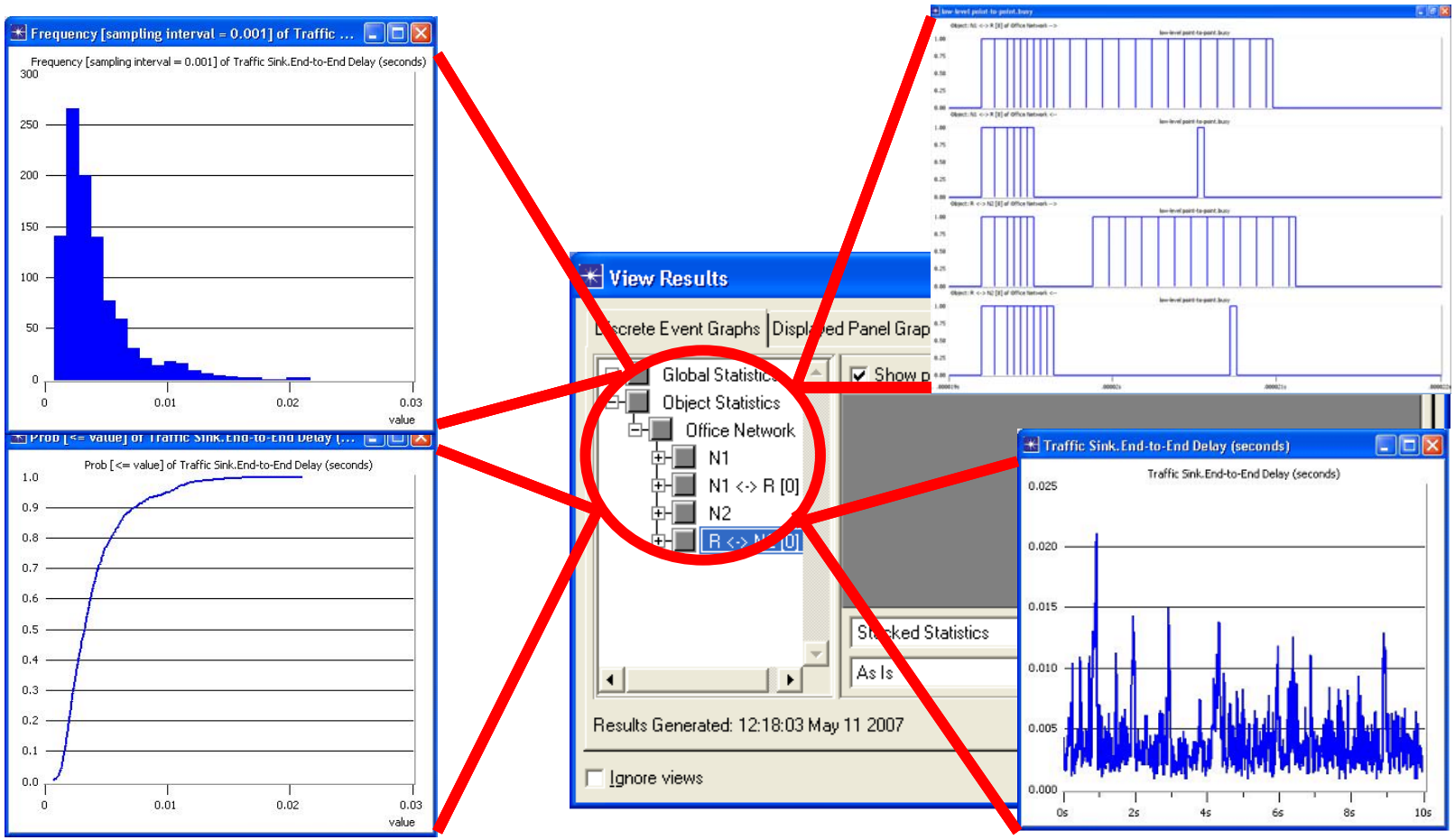


Process domain



OPNET Source code

OPNET outputs and analysis



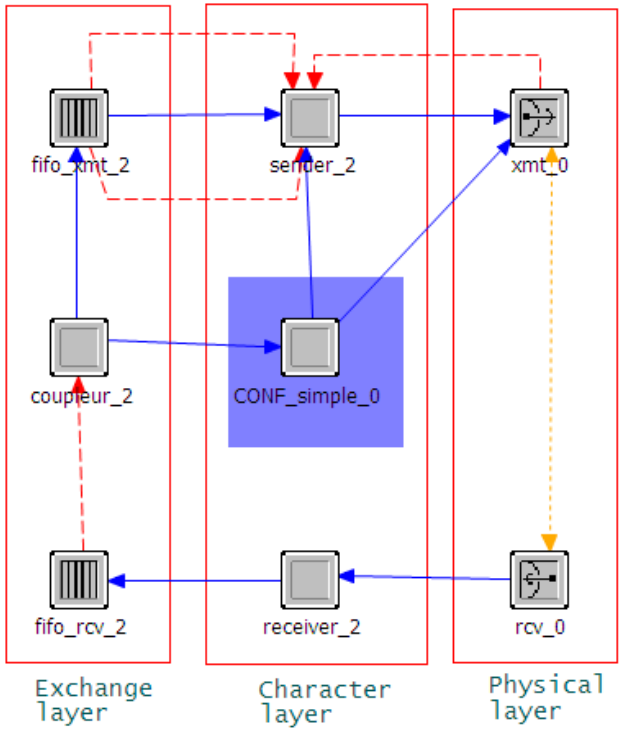
Example of statistics available

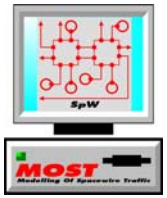
- The buffer size of a module (in bytes)
- Output port attribution of the router :
  - This shows which output port of the router has been used to route a packet in function to the simulation time.
- Transmission of each character on a link :
  - This statistics shows the traffic flow on a link, showing each byte in function to the simulation time.
- End-to-End delay (seconds):
  - This shows the value in seconds for a packet to transit from the source node to the destination node. Possible to obtain ETE per packet type (SC, CMD, HK, TM…)
- End-to-End delay round trip (seconds):
  - This statistics enables to know the time taken by a node to receive the response of its command.

SpaceWire building block overview

- The SpW building block, is composed of 3 layers
  - Physical
  - Character
  - Exchange

- Emission and reception is asynchronous, so it is handle by two different processes.
- A coupler connects the emission and reception on the exchange level, to initialize the link correctly.





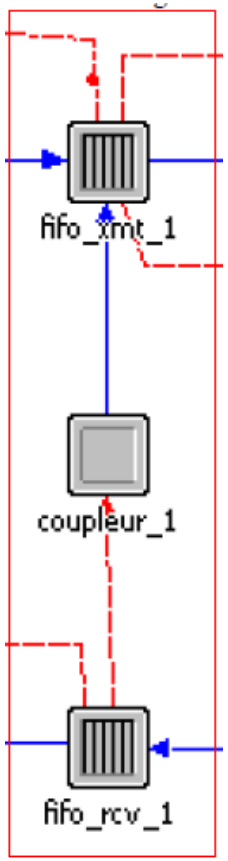
Processes involved (1)

- **Physical layer:**
  - Connect packets flow coming from the emitter towards the receiver to the SpW link.
  - Emitter and receiver are point-to-point type.

- **Character layer:**
  - The Sender:
    - Sends data and control characters stored in the emission buffer if the SpW link is available.
  - The Receiver:
    - Handles characters coming from the physical receiver
      - If data character, it sends it to the receiver buffer.
      - If command character, it notifies the coupler and sinks the character.
      - It detects a disconnection with notification to the distant emitter.
  - The Configuration process:
    - Configuration specific to each node

Processes involved (2)

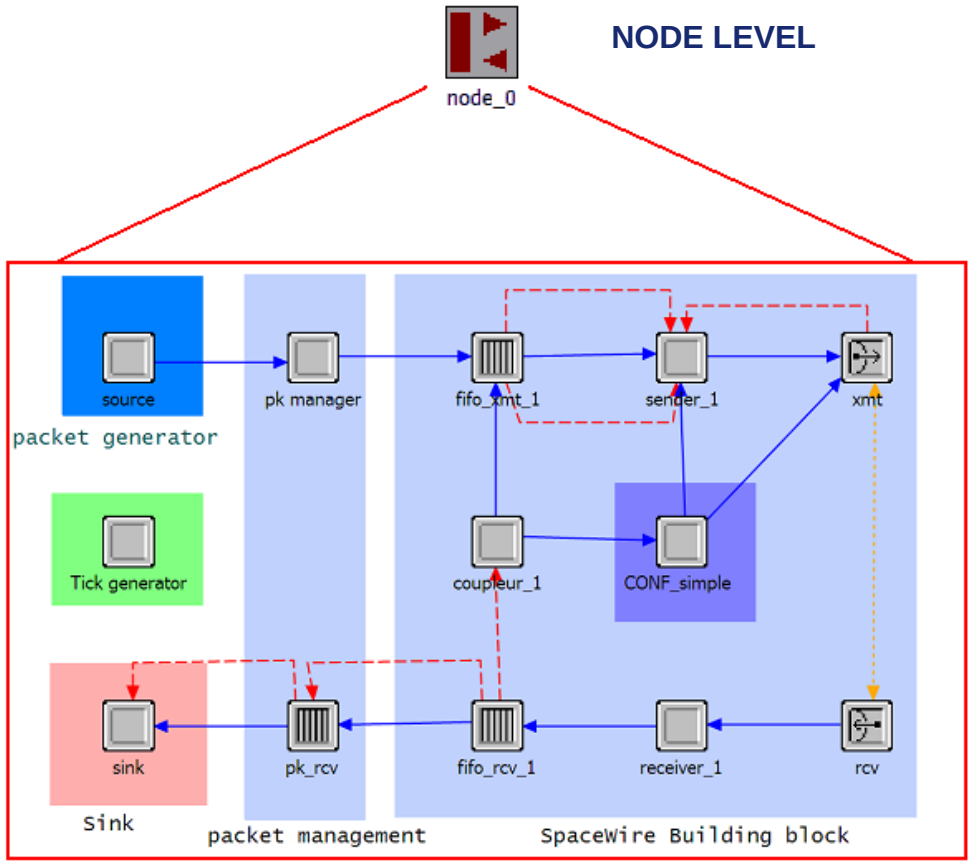
- **Exchange layer:**
  - The coupler: manages the state of character layer and permit to initialize the link interface and the flow control. Communication with emission and reception modules is done with distant interrupts.
  - Emission buffer:
    - Receives packets from both the pk_sender and the coupler.
      - SpaceWire packets from the application
      - Control characters from the coupler (FCT)
    - The emission buffer sends the highest priority to the character layer
  - Reception buffer:
    - Receives the packets coming from the receiver.
    - Stores data type characters and transmit them to the application
    - Transmits packet on request (flux access) on the output stream.

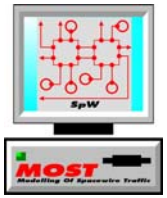


MOST library

- **Nodes**
  - Generic nodes to test protocols developed
- **Protocols**
  - SpaceWire
  - RMAP
  - CCSDS Packet Transfer Protocol
- **Components**
  - SPW 10X Router
  - SMCS116SPW
  - SMCS332SPW
  - Remote Terminal Controller
- **Link**
  - SpaceWire link

Generic node





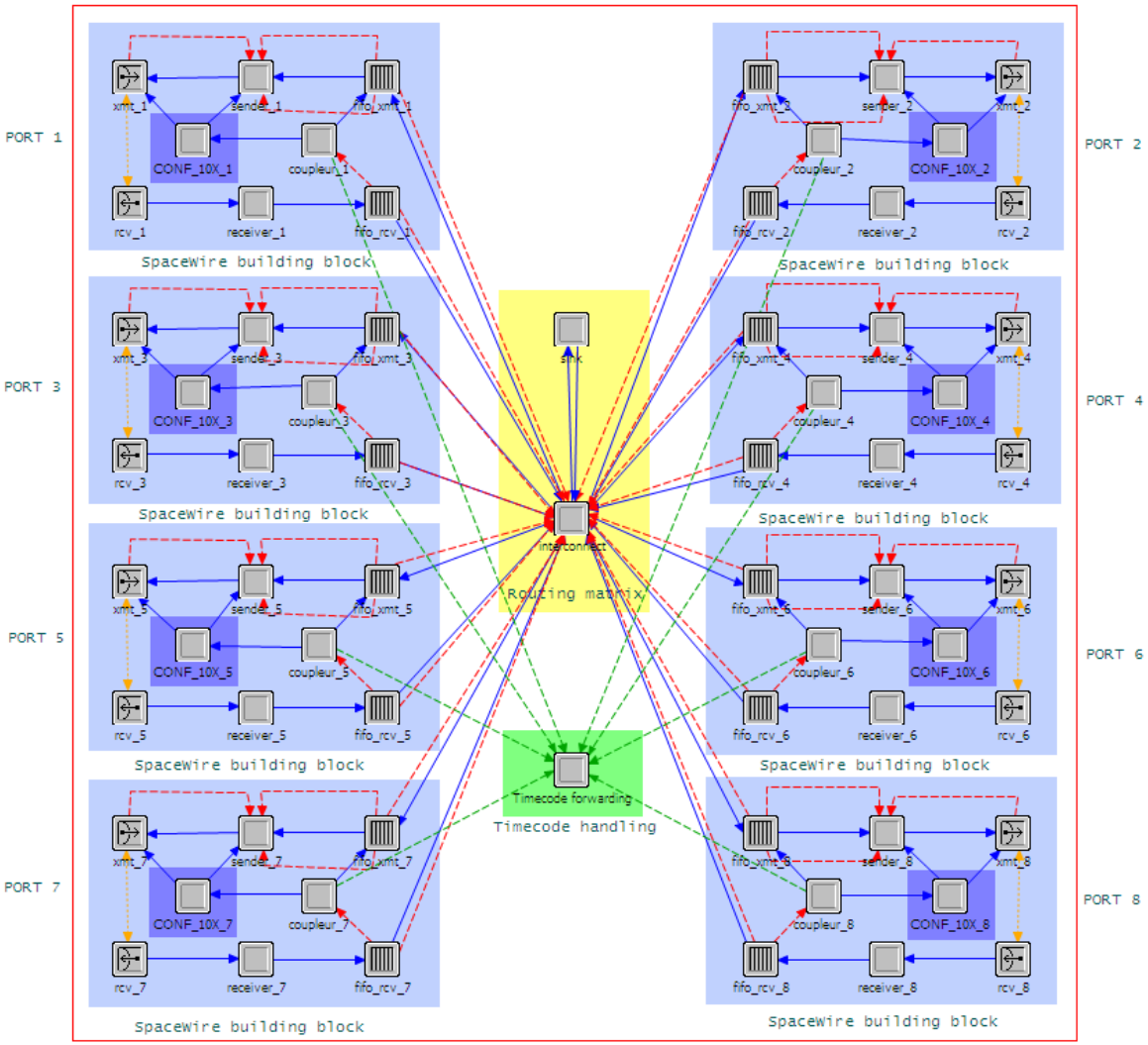
SpW - 10X Router

- The 10X is composed of 8 SpW interfaces
- The interfaces are connected by an interconnection matrix and a sink:
  - The matrix assigns the correct output ports to the incoming packets. For each incoming packet the matrix creates a sub-process to handle the routing mechanism
  - The sink enables to destroy packets consumed by the router.

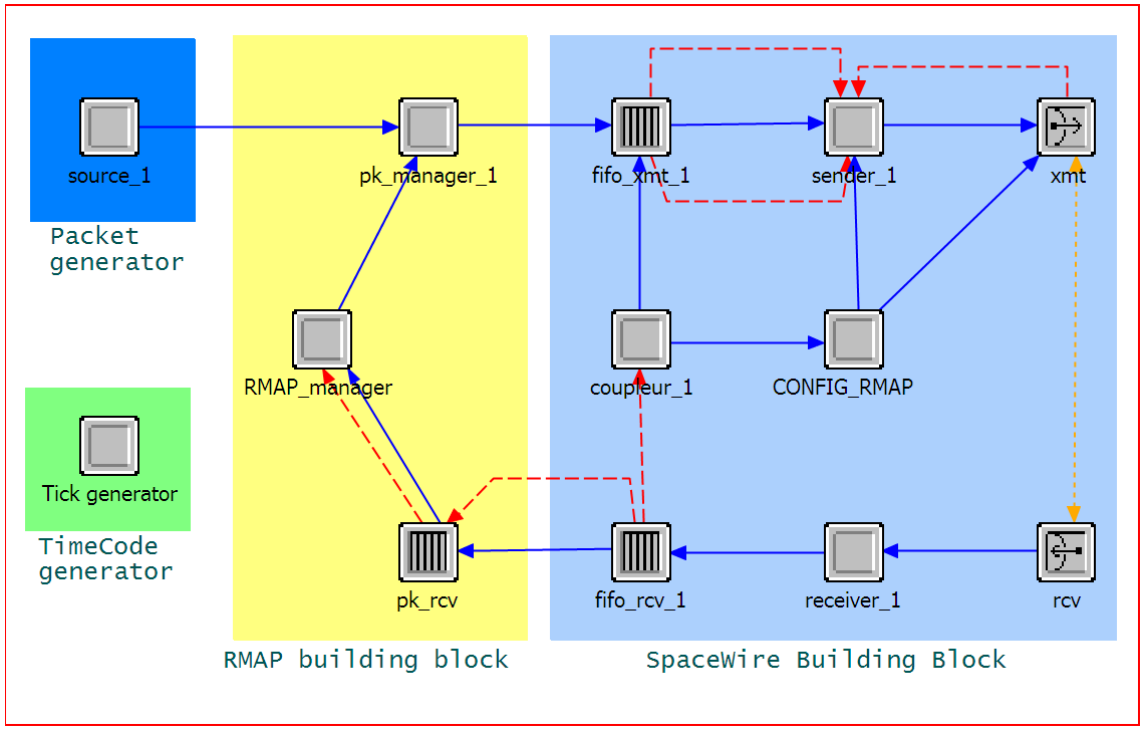
- The “Routing table” tells the router the output port to transfer a packet to, for each possible address.
  - Logical addressing
  - Group adaptive routing

- Round robin: priorities management
- Data rates corresponding to the datasheet of the 10X router
- Functionalities that have to be implemented: 2 FIFOs ports

SpW 10X architecture



RMAP building block

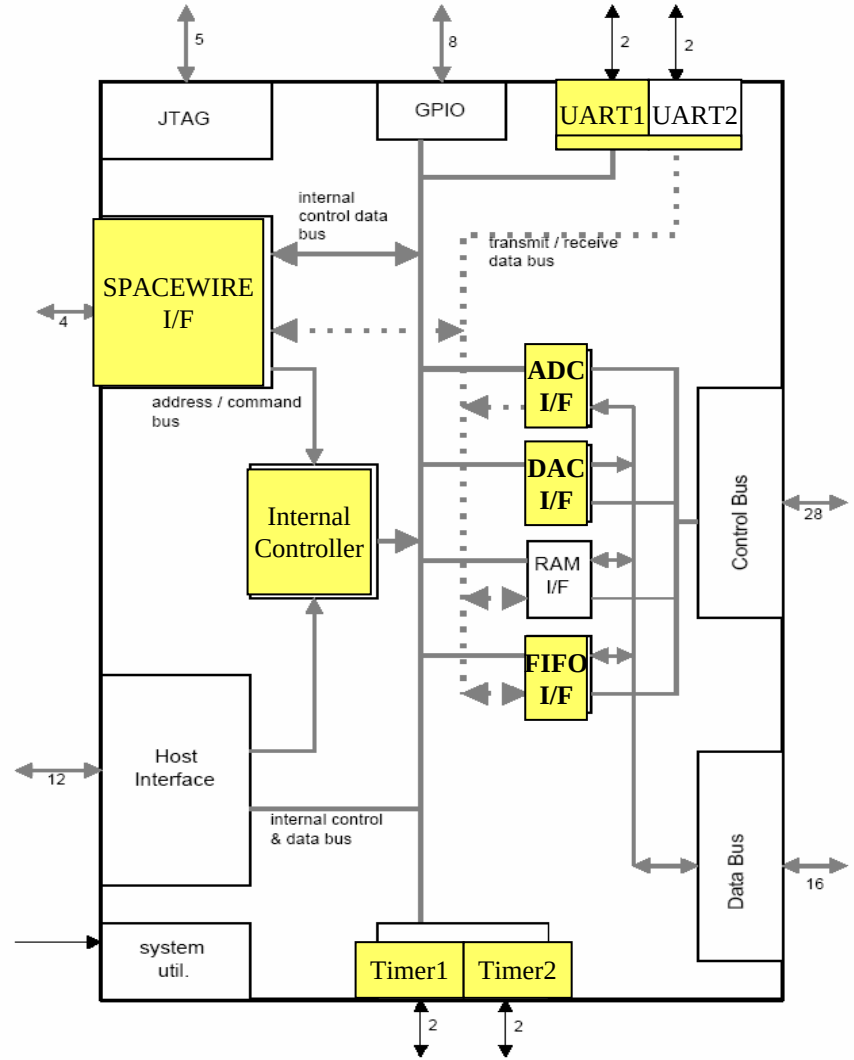


CCSDS PTP building block

- **This node has the same structure as a SpaceWire node.**
- **But it consists of other information to be representative of the CCSDS PTP as defined in :**
  - SpaceWire - CCSDS Packet Transfer Protocol, ECSS-E-ST-50-53C, 5th February 2010

- **Information that are inserted in the packet are Source APID and Packet type TM or TC. The encapsulation of CCSDS packets is done by adding APID and packet type, so it can be retrieved by the destination node.**

SMCS116SpW – Description



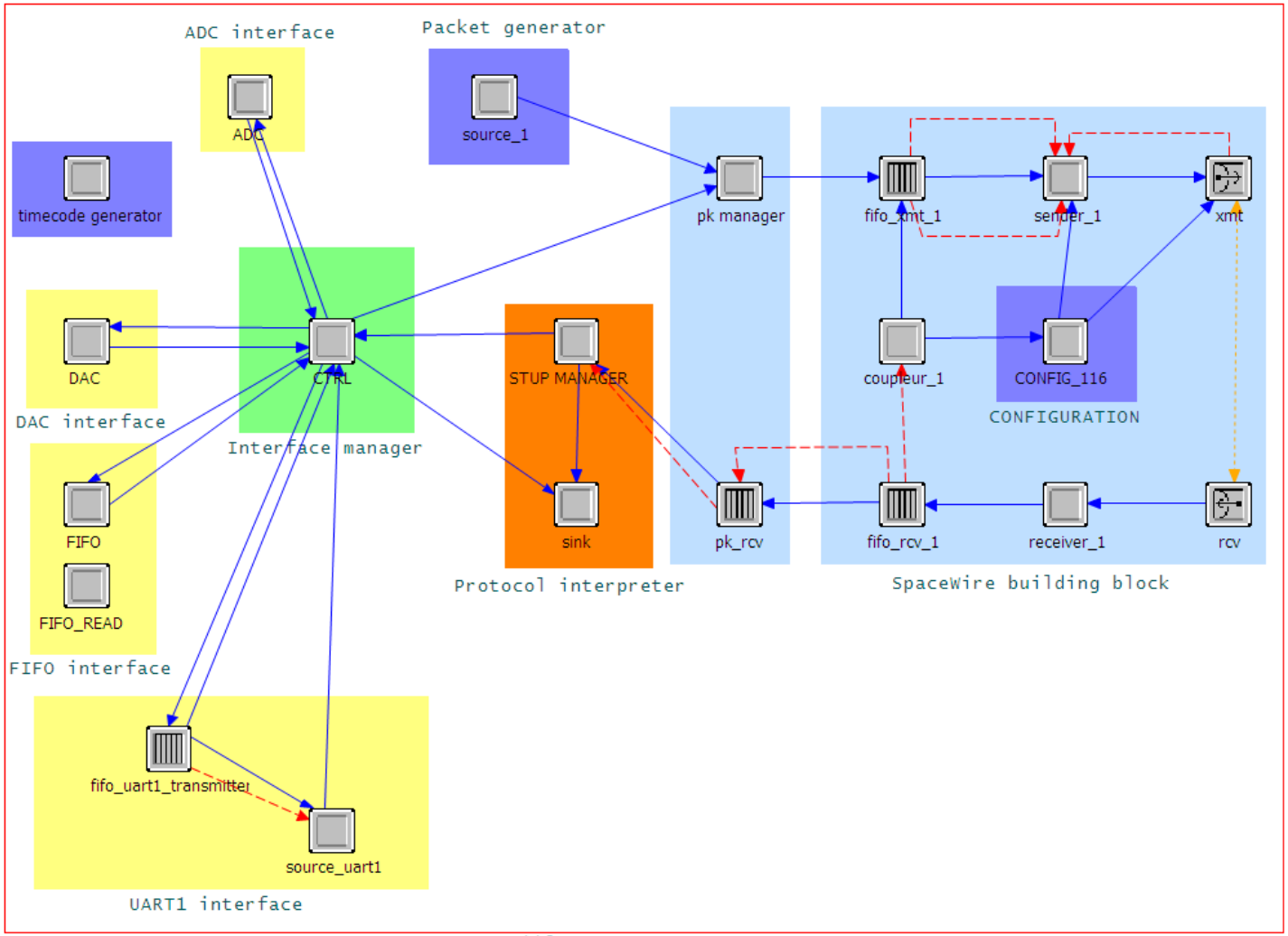
Component model coverage:

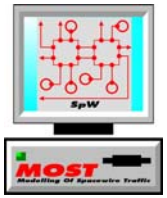
- SpW/STUP
- Functions related to traffic (source/consumer)
- R/W registers

- ADC as source
- DAC as consumer
- FIFO as source/consumer
- UART1 as source/consumer
- Timers (involved in ADC processing)

- *RAM, GPIO: not covered*

SMCS116SPW - architecture

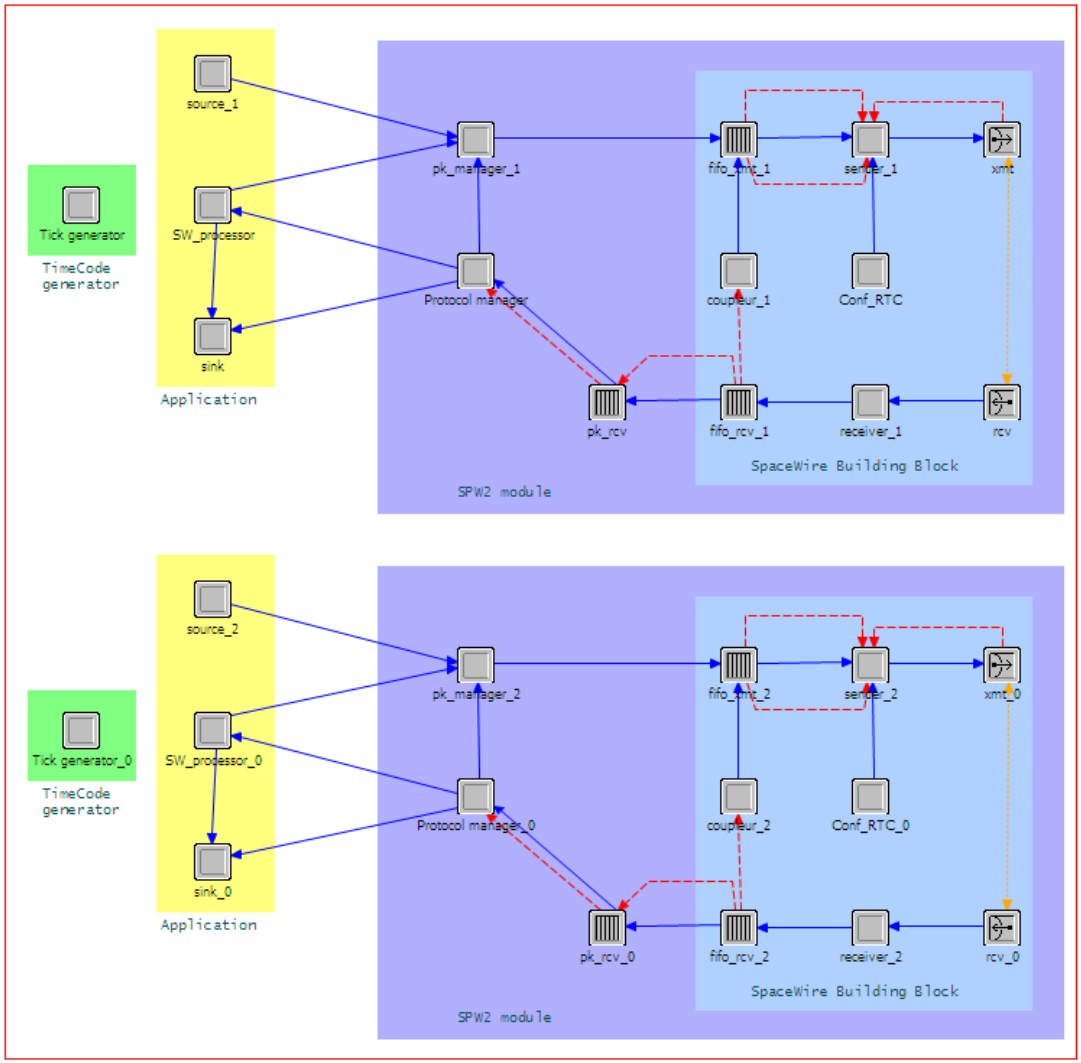


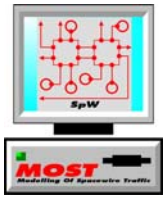


RTC - description

- **Sophisticated LEON2-based System-on-Chip**
- **Many interfaces, including CAN and SpaceWire, all are driven by the processor**
- **Current requirements focus on SpaceWire interfaces *only***
- **RTC has two SPW2 IP cores**
  - Each core has one SpaceWire ports
  - SPW2 is the focus of simulation
- **Cores support RMAP and "Virtual Channel Transfer Protocol"**
  - RMAP support is partially hardware, partially software
  - VCTP is hardware supported and routes packets to "virtual channels"
  - RTC supports one virtual channel and an RMAP channel per link
- **Functional behaviour of both protocols is important to network behaviour**
  - For example: VCTP operation can block the network
  - For example: latency of RMAP replies depends on type of operation

RTC - architecture





MOST validation & demonstration

2 study cases with the following main objectives:

- To demonstrate MOST performances:
  - Ability to test complex network topology
  - Availability of a large set of representative SpW component & BB
  - Possibility to test several configurations of a Network
  - Capacity to provide a large set of observables for detailed results analysis
- To test representative on-board SpW network topology
  - Analyze feasibility & performances (margins/bottlenecks/buffer status)
  - Adjustment of SpW BB setting & network configuration
- To test various SpW interfaces in one SpW network and make use of the RTC, the SMCS116SpW and the 10X router.

Main objectives

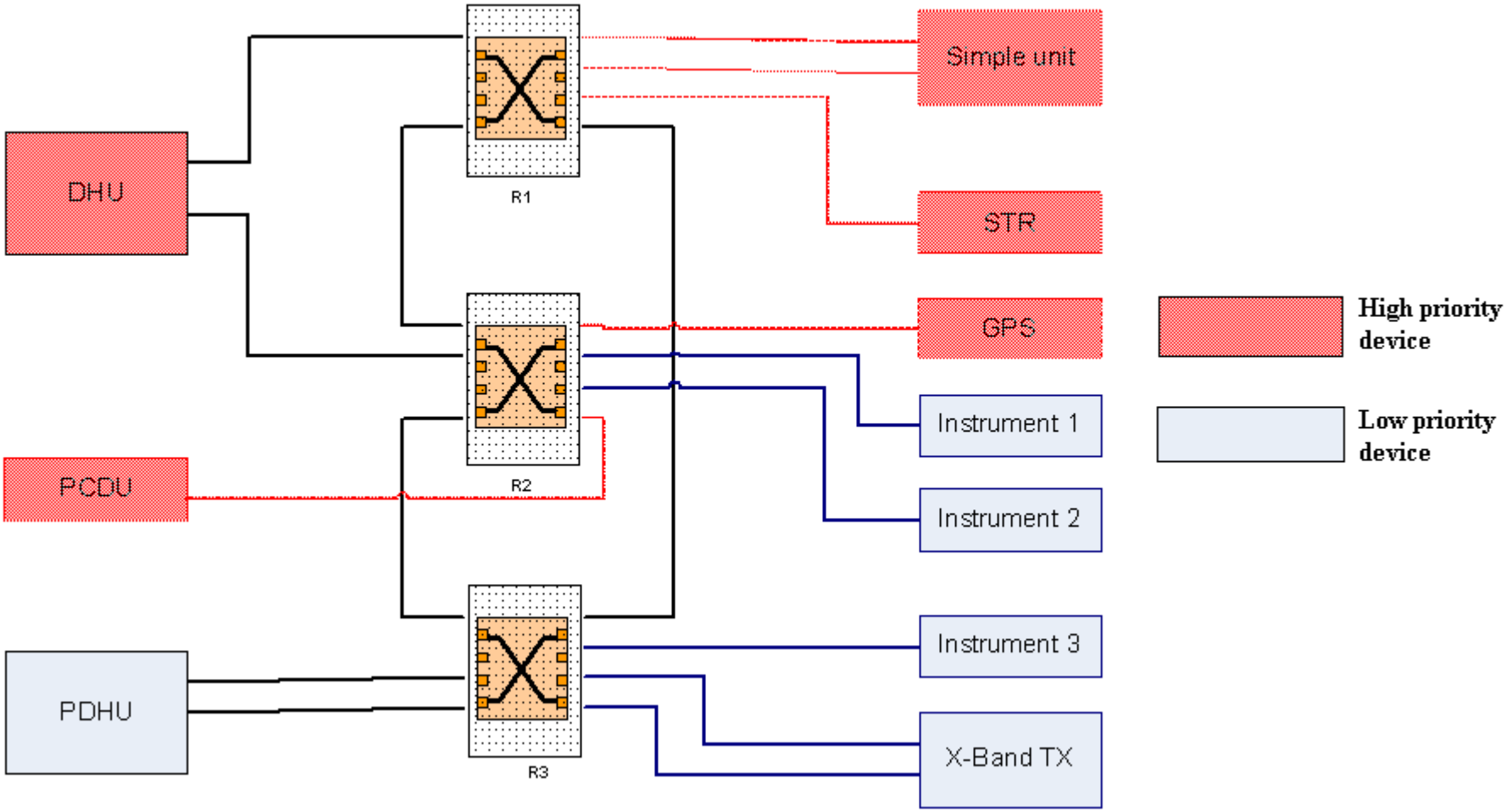
- To show a board SpW network including both PF units and instruments, which is current trend for new architecture R&D.

- To mix critical critical data items (TC) with large non critical data (Science data):
  - Platform side with Hard real-time constraints, matches DH cycle (~ 10 Hz)
  - Payload side with Lower real-time constraints: Low rate cyclic images, asynchronous missions, autonomous decisions (future)
  - Communication between DHU and PDHU
    - DHU controls PDHU (and instruments) activities (start / stop)
    - DHU store HK in PDHU
    - DHU reads/write files in PDHU, on request

Study Case 1: Network features

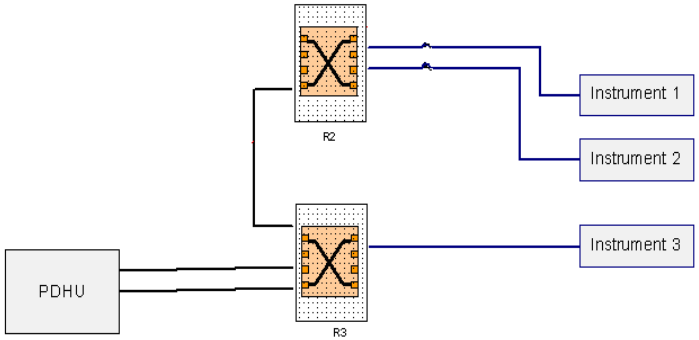
- **The network including both processing units and instruments:**
  - DHU : node which controls the spacecraft: sending commands, collecting HK
  - PDHU (MM): node which collects instrument data
  - “Intelligent” platform units (STR, GPS) are similar to host nodes: Consumes requests from DHU, answering data messages; Can deliver cyclic messages from its own
  - “Simple” platform units are controlled via simple I/O links: Analog, DR/DB, DS16, ML16, serial, HLC/LLC, specific I/O …
  - Instrument are similar to host nodes: Receives commands from DHU, deliver science to PDHU
  - PCDU

Study case 1: network topology

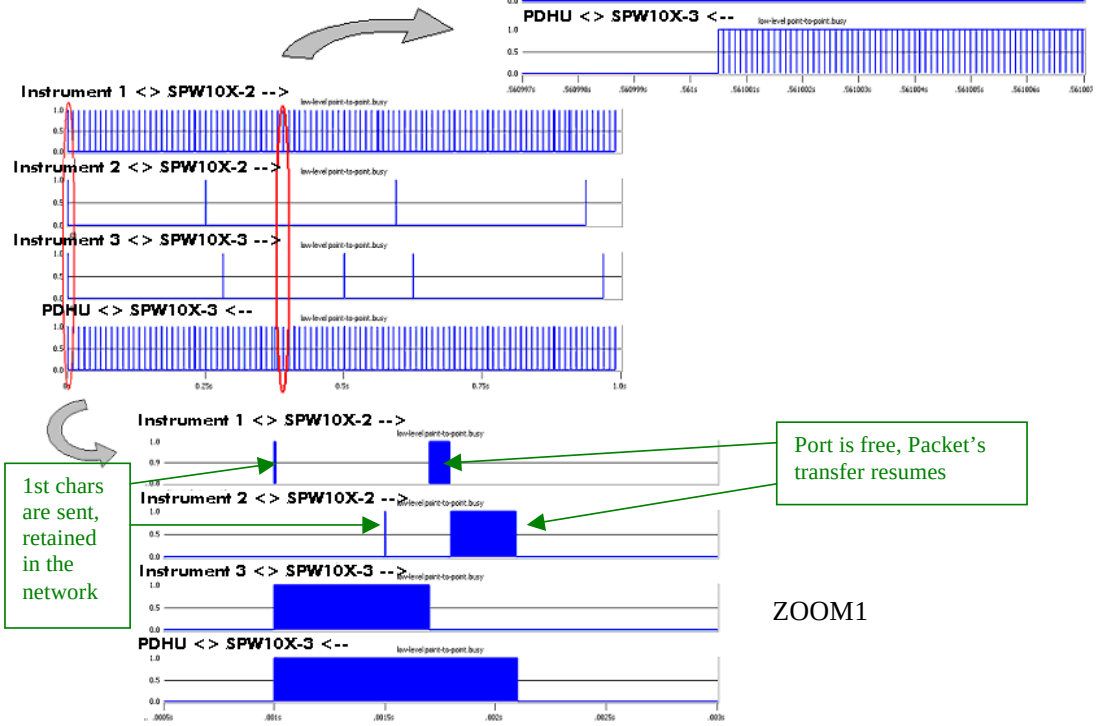


Tests – Low priority modules (1)

- Traffic between the PDHU and Instruments



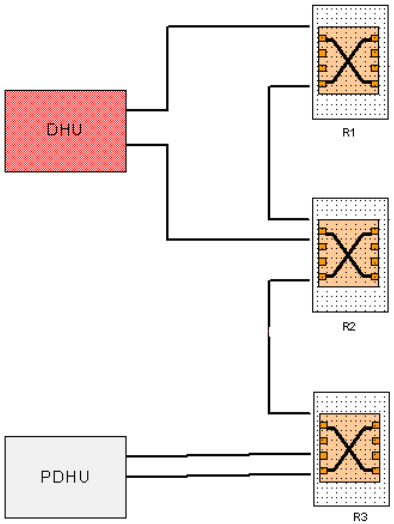
ZOOM2

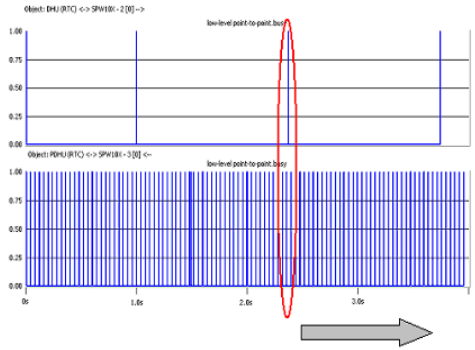


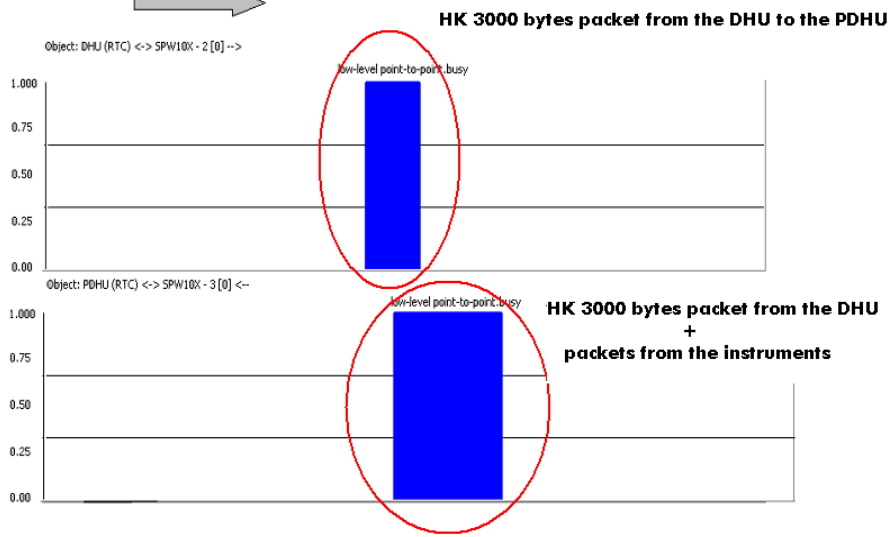
- Science packets are correctly delivered from the instruments to the PDHU.
- A packet is held if its destination port is blocked. It resumes when this gets free.

Tests – Low priority modules (2)

- Traffic between the DHU and PDHU



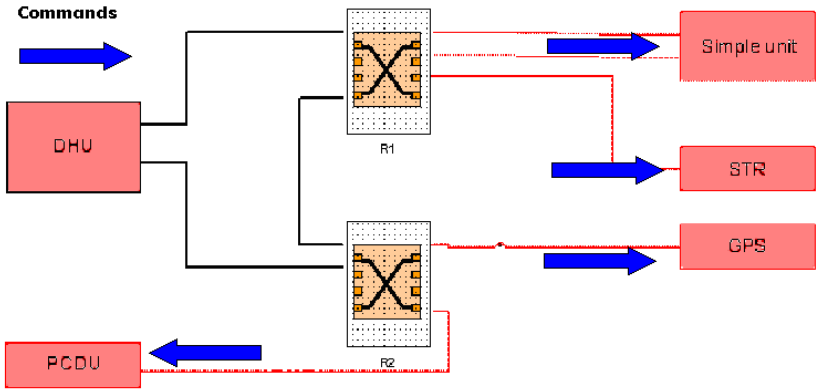


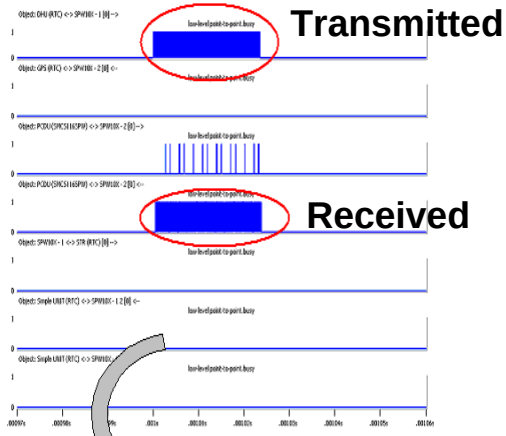


- Housekeeping packets correctly delivered from the DHU to the PDHU.
- These packets arrive to the PDHU at the same periods that the packets from the instruments.

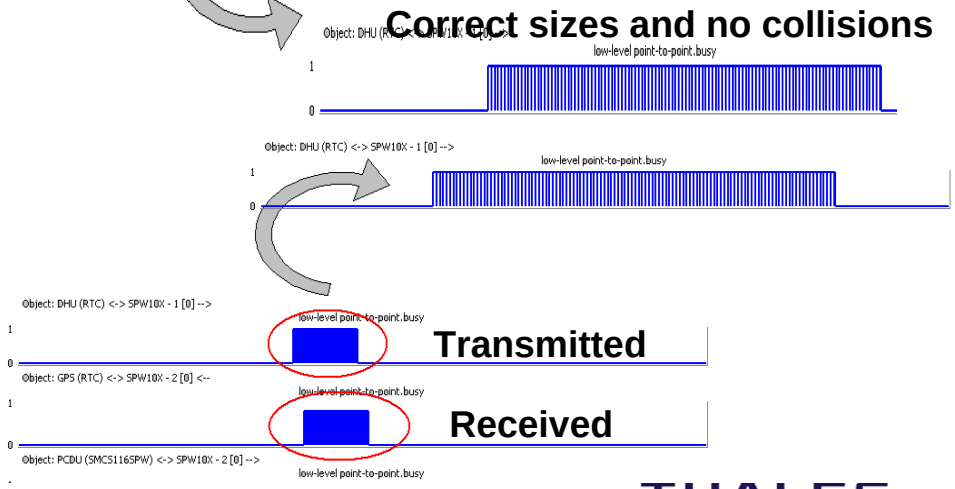
Tests – High priority modules (1)

- Traffic from the DHU to all nodes (commands)



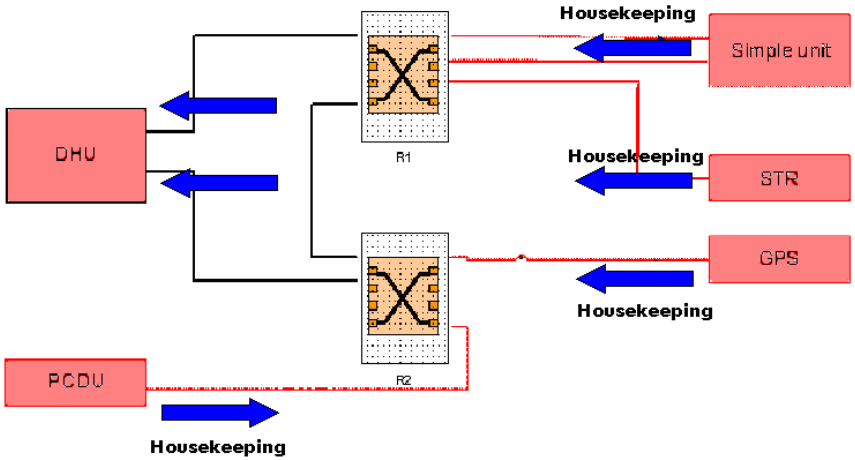


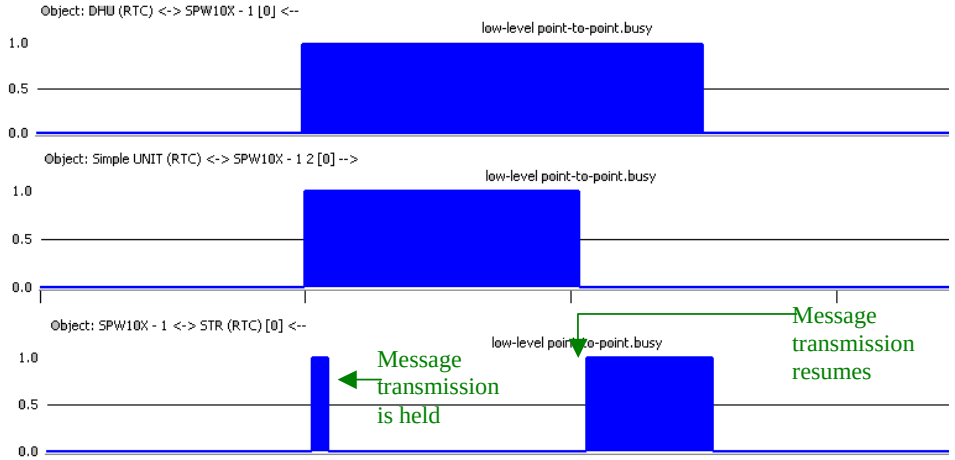
- DHU sends commands to ALL the nodes of the network in 32Hz cycles.
- Analysis states that all these packets are correctly delivered in the correct time periods.
- No collisions are observed, and channel are not fully occupied.



Tests – High priority modules (2)

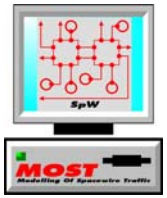
- Traffic from all nodes to the DHU (housekeeping)





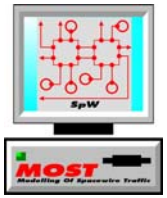
- All the nodes send housekeeping packets to the DHU in 10 Hz cycles.
- When several packets arrive at the same time to the DHU, they are held till the input port gets free.

GPS	PCDU	STR	Simple Unit
100 bytes	50 bytes	250 bytes	500 bytes
PVT message …	Cells Voltages, currents …	Attitude Quaternion …	I/O coupler multiple discrete channels



Study case 1 - Conclusion

- Different scenarios were analysed and conducted to the following conclusion
- The traffic analysis has been done per sections of the topology.
- The platform devices of the network (DHU, STR, GPS, PCDU and the Simple Unit) exchange commands and housekeeping data that are transferred with the shortest possible delay.
- The links remain free most of the time. The delay is mainly depending on the routers latencies and the packets sizes.
- The effects of the traffic burst created by the PDHU on the network mainly affects the global end-to-end delay of the packets

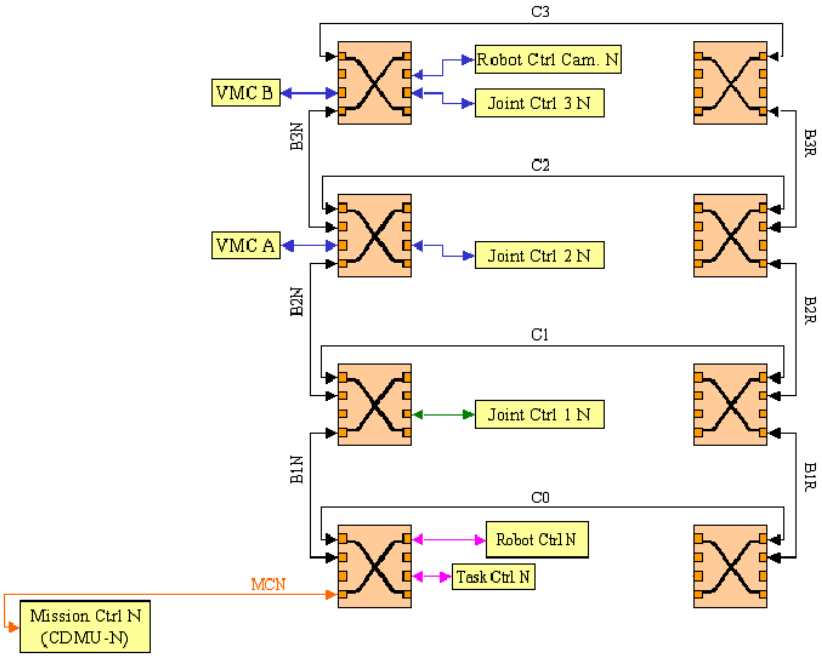


Study case 2 – Robotic mission

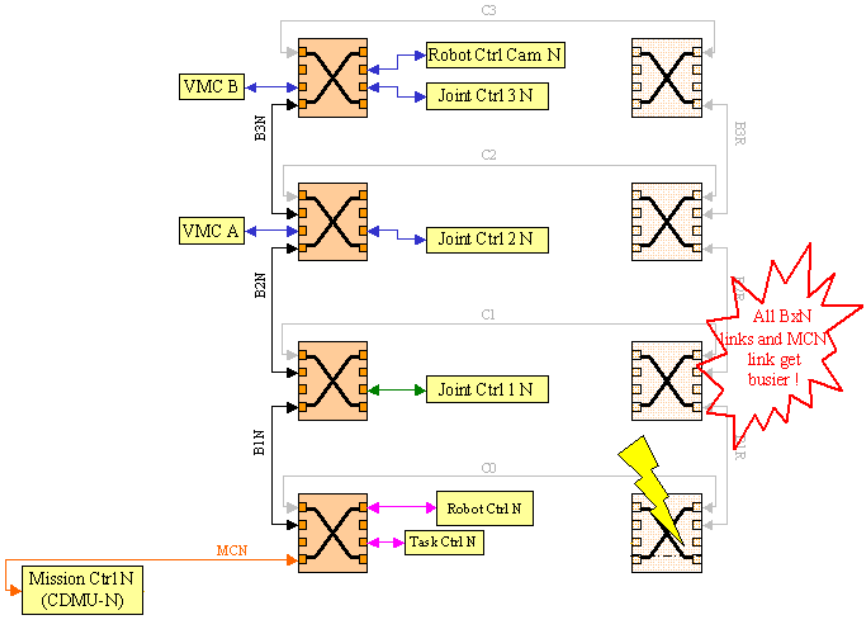
Main objectives:

- To show the capacity to synchronize device activities with Time-Code, scheduling activity of nodes and traffic load.
- To mix critical data items (high priority) with large non critical data (low priority) on the network
- To use both RTC and SMCS116 in one network topology
- To observe end-to-end delay per packet category (TC, TM, SC, CMD, Ack)

- **Full network**
  - Collisions
  - No collisions

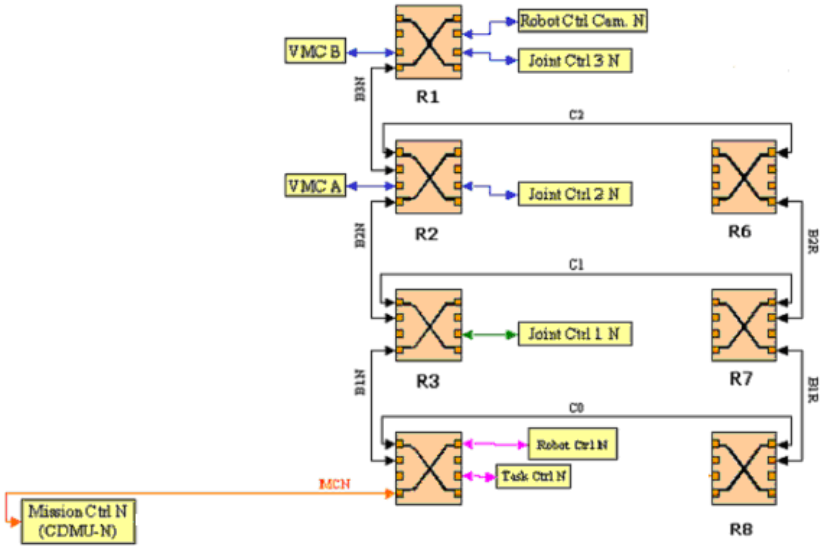


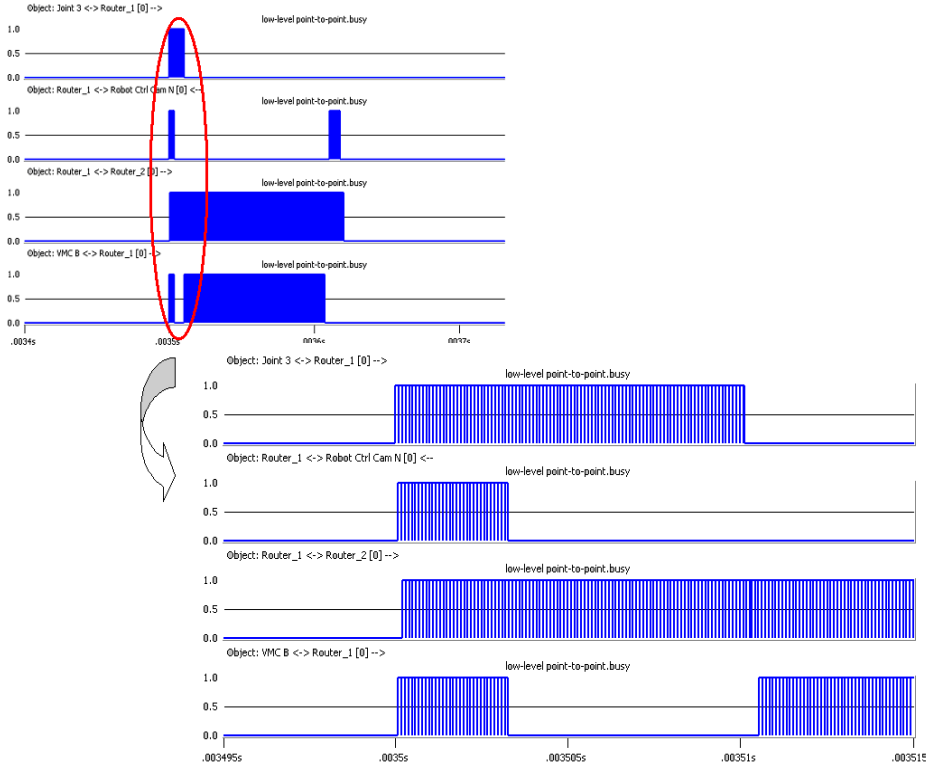
- **Network with degradation**
  - Collisions
  - No collisions



Network with degradation - Tests (1)

By **disabling Router #5**, the traffic from the VMC B camera, which is configured with a low priority, is held during long period of times. The effect on the system is the reception of delayed video images.

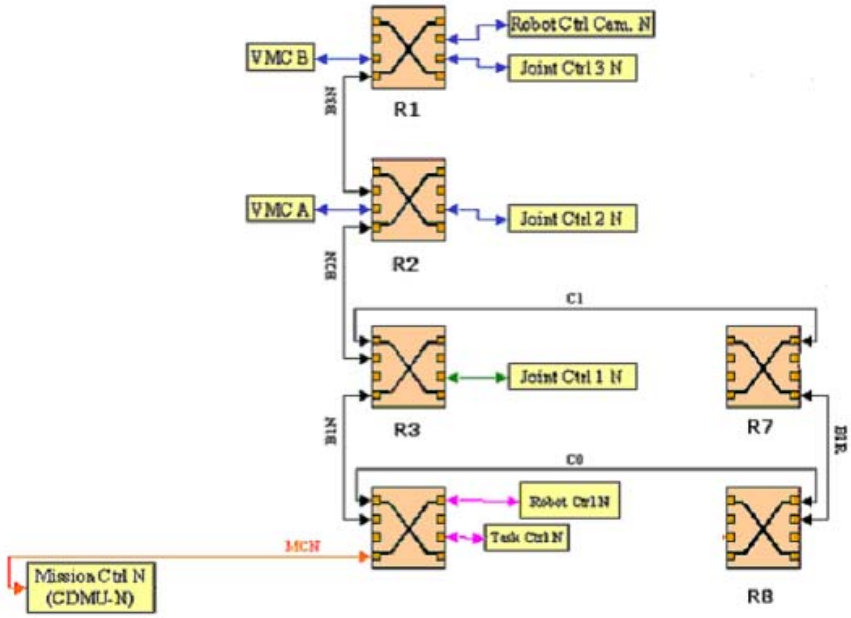


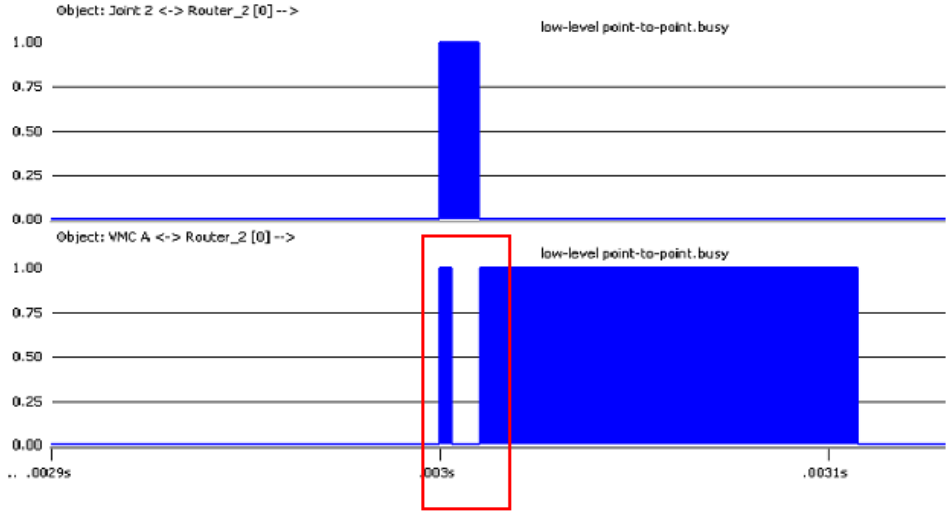


Network with degradation - Tests (2)

By **disabling Router #6,** some of the packets coming from the devices of the Router #1 are lost. Besides of this, the packets from the VMC A and the Joint Controller #2 are delayed in time.

- The **Robotic Arm** begins to not work electromechanically properly
- The **VMC B video** is delayed and sometimes is not refreshed
- The **VMC A video** is delayed.

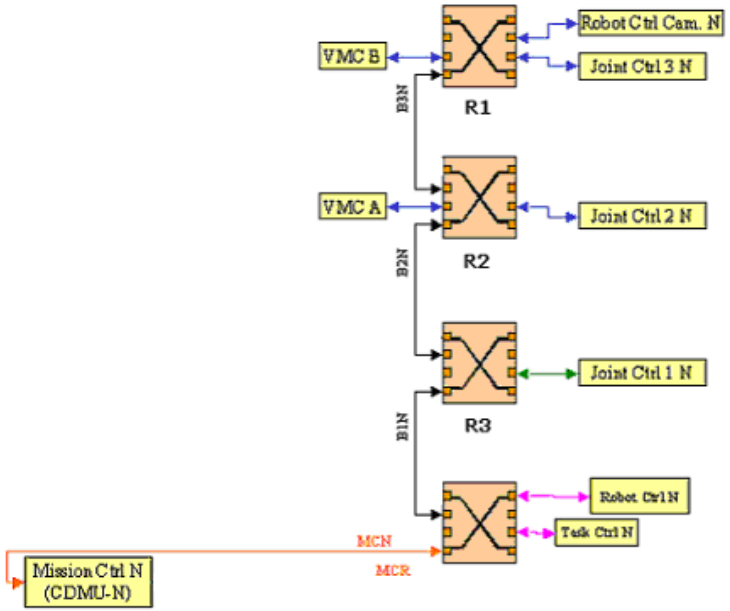


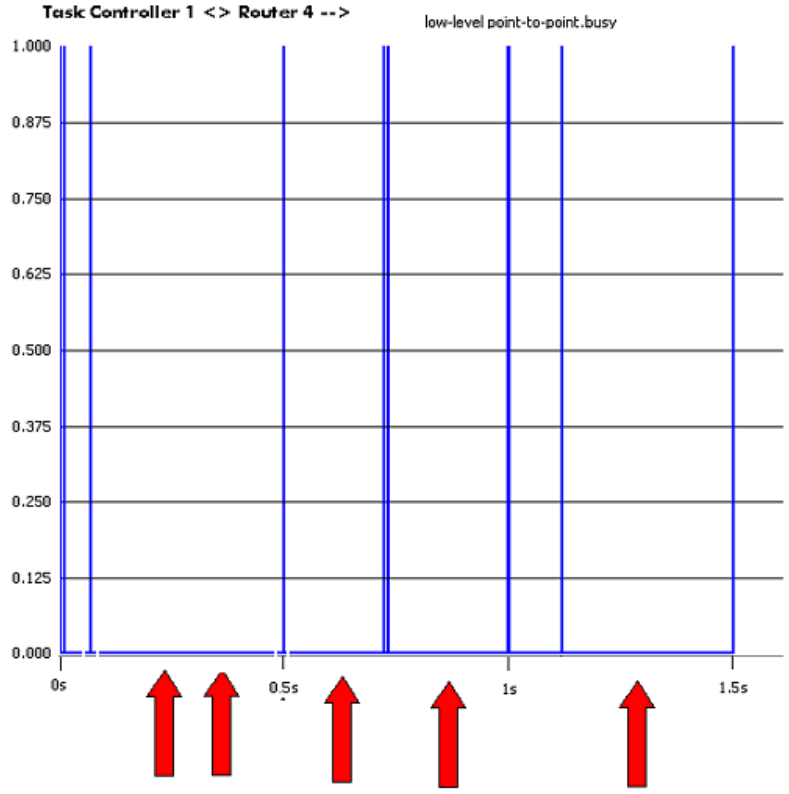


Delay in the VMC A reception

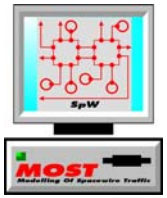
Network with degradation - Tests (3)

By **disabling all the redundant branch of routers**, a high quantity of packets are either delayed or lost. By this, neither the video images or the robotic arm are able to perform under the correct premises of the system.





Packets lost



Study case 2 - conclusion

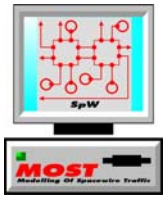
- Non-degradation and progressive degradation of the network have been considered. The analysis has been focused on the behaviour of the data flow in all these cases, dealing with collisions, busy ports and GAR functions in the routers.

- In full network scenarios, the GAR functions shares the traffic on nominal and redundant way, maintaining the load to reasonable level on links.

- Over the degradation scenarios, bottleneck occurs since packets have to share the same nominal channel to arrive to their destinations. When the network is more and more degraded, packets start being lost, the low priority components being more affected.

Documents delivered for the study

- Reference manual
- User manual
- Demonstration report
- HW cross validation test plan and test report
- Validation test report
- MOST compliance matrix wrt URD



Perspectives & Conclusion

- **MOST upgrade**
  - SpaceWire library extension with new components
  - Protocol BB upgrade with PUS standard (secondary header: type/subtype, calling for packet(s) generation)
  - Protocols evolution

- **MOST is a major contribution for the SpaceWire standard and associated technologies development.**

- **Demonstration based on representative study cases will show that the simulator can be used for current mission.**

- **MOST is a simulation tool which is representative enough to be used for current mission to support SpW network traffic analysis, predictions and validation.**

- **MOST offers SpW experts the possibility to test new design**

- **MOST concept provides a progressive tool, built with independent SpW building blocks which can be changed to test new SpW technology or even SpW standard evolutions, without waiting for HW development.**

ThalesAlenia Space
A Thales / Finmeccanica Company

Thank you for your attention

THALES