SIMULATION OF A SPACEWIRE NETWORK

Session: SpaceWire Networks and Protocols

Long Paper

Authors : Philippe Fourtier, Alain Girard, Antoine Provost-Grellier, François Sauvage

Thales Alenia Space,

Etablissement de Cannes,

100 bd du Midi, 06156 Cannes La Bocca, France

E-mail: *alain-felix.girard@thalesaleniaspace.com, antoine.provost-grellier@thalesaleniaspace, Francois.Sauvage@thalesaleniaspace.com, fourtier.philippe@thalesaleniaspace.com* (*phone : + 33492926104)*

Abstract :
Data traffic gets more and more heavy and complex, mainly due to performance enhancement of payloads. The SPACEWIRE standard usage covers the growing need of high speed board-to-board communication links. This requires new tools providing efficient help in conception, development and validation of flight network designs. This paper presents a simulator dedicated to embed SPACEWIRE networks, taking into account all communication layers, up to constraints induced by nodes' behavior.

1. INTRODUCTION:

The article covers the development and set-in-use of a first validated version of a SPACEWIRE traffic simulator, intended to support Space missions.
Use cases have been selected, covering the various domains: exploration, LEO missions, robotics, interplanetary for main ones.

2. DEVELOPMENT OF A REPRESENTATIVE SIMULATION TOOL:

2.1 Development history:

When SpaceWire standard started to be the privileged solution for high rate communications, it appeared that classical validation means as bus monitoring, line analyzer or any other SW based tools were not adapted to SpW based networks, multiple point-to point links requiring too many measurement points. Furthermore, the traffic becomes more complex, taking advantage of much higher data rates. This led to specify and start the development of a representative tool making and executing numerical models of SpaceWire networks, where each exchange can be observed.

A study simulator called MOST (Modeling Of SpaceWire Traffic) has been first developed by TAS-F in year 2006, based on OPNET toolkit dedicated to network modeling.
The present version of the simulator proposes a library including the ESA SpW router, generic nodes, generic application models (Producer/Consumer).
Many features of the standard are implemented, including all basic SpW protocol (flow control, communication opening sequence, time-code, logical and physical addressing, local

buffering …) and upper layers like are GAR mechanism and RMAP protocol (Remote Memory Access Protocol) as described in ECSS-E-50-51.

MOST has been developed with respect to requirement to be an opened and living tool. An incremental development has been applied, each step enriching the simulator by modeling new SpaceWire features and appending SpW component.
In addition, MOST provides user with ability to insert functional behavioral models of nodes, without impacting standard SpaceWire models. A specific documentation is available for users in the form of manuals, each dedicated to the role of developers (product maintenance, models developers and network designers).

2.2 Concept:

2.2.1 Objectives of the simulator:

SpaceWire networks are much more complex than data bus or single node-to-node serial links, such that hand-made analysis or classical traffic analyzers are no longer adapted. Hence, the main goal is to have a representative validation tool dedicated to SpaceWire Network, aimed at supporting system and design activities in all phases of a project, from phase-A (i.e. pre-developments, trade-off) up to operational mission life (maintenance, ground investigations), by providing mean:

- To create quickly a numerical model of networks topology,
- To decrease design risks and secure planning, by early verification,
- To Keep control on traffic load, globally and in all parts of the network,
- To identify weak parts of the network topology,
- To verify load margins and transfer performances,
- To evaluate/refine budgets of resources, meeting system specifications,
- To verify the network FDIR by injecting simulated failures,
- To replay anomaly cases observed as test scenario,

2.2.2 Description:

Built upon the OPNET toolkit, a SpW library is provided, with validated up-to-date models of SpW parts (routers and nodes). Components models are fully representative of the SpaceWire standard and of space qualified components of the market (i.e. ESA router, nodes), based on data sheets of suppliers.

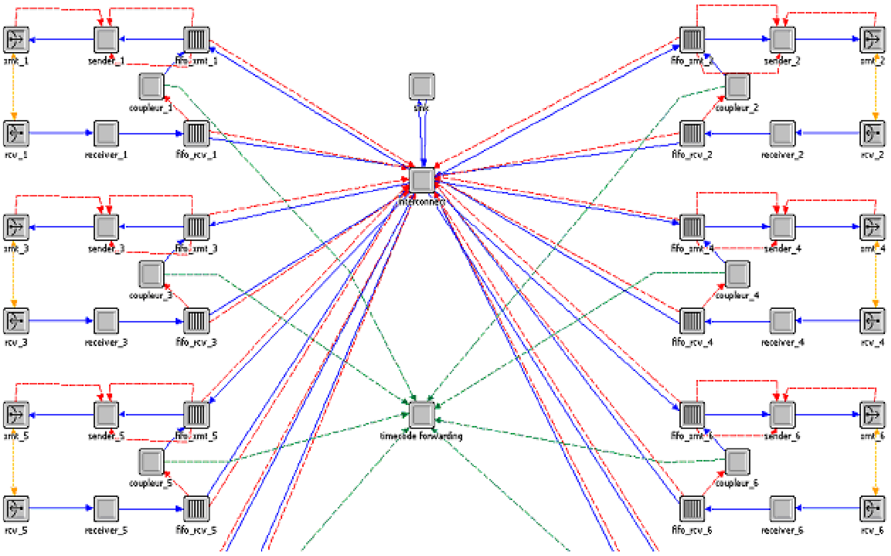


Figure 1: Standard model of router with time-code

The traffic simulation covers all layers of communication, starting with transfer of characters. It covers all standard features: flow control, time-out, time code, physical or logical addressing, GAR mechanism … up to application. Default applications are provided for nodes, based on simple data generation and consumption models.

MOST takes advantage of the OPNET Man Machine interface, to propose an easy and natural way to build custom networks. The nominal way to do it is to build components instances via a drag-and drop selection mechanism, then to link then together with wires as stated in mission design, and finally configure attributes of these links.

Once drawn and configured, scenarios are executed, recording standard or user-defined statistics, which can be drawn after execution in different ways. All or part of traffic can be observed, per wire and direction. Figure 3 represents the detailed traffic between 2 end users. The X axis drawing is time representative. Width of busy periods represents the duration of character transmission.

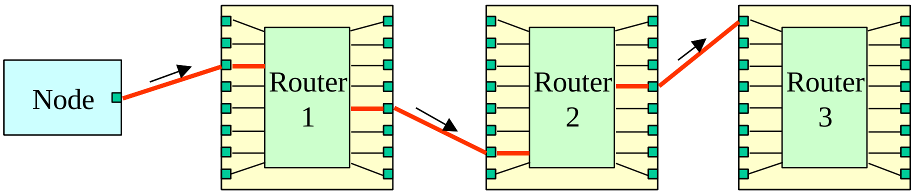


Figure 2: Network matching Traffic pattern of next figure

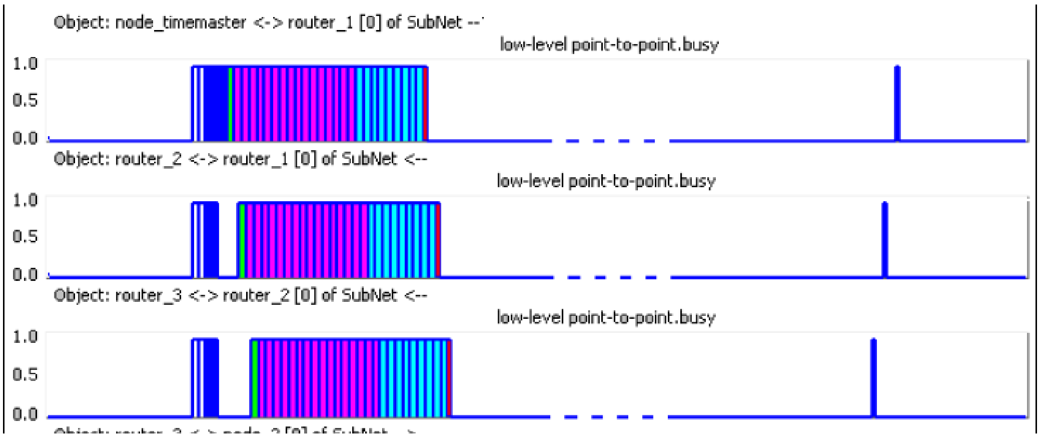


Figure 3: Graphical representation of traffic (node → router1→ router2 → router3**)**

The simulator is also required to cover anomaly cases, where numerical model is the unique way to check these unexpected events. To do so, pre-defined anomalies can be injected within scenarios simulating traffic loss, dumb node, erroneous characters, endless messages …

2.3 Two kinds of end users

2.3.1 Node Model Developers

These people are in charge of development of customized nodes. This work consists in replacing default applications by representative behavioral models of nodes. The standard part of models is not affected by these developments.

The developments are led under OPNET environment, using OPNET state machines and C language. A specific user manual is intended to model developers.

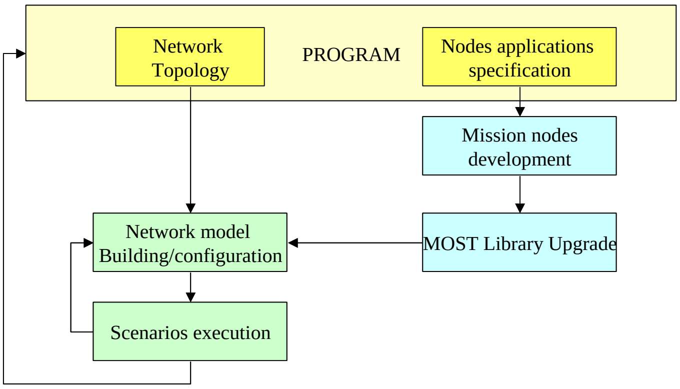

PROGRAM
Network Topology
Nodes applications specification
Mission nodes development
Network model Building/configuration
MOST Library Upgrade
Scenarios execution

Figure 4 : MOST process

2.3.2 *Network designers*

They are in charge of implementing network designs by using the graphical editor, picking building blocks in library, linking them by drag-and-drop interface.

Then, they affect the attributes of blocks with mission values. At that time, they are ready to run scenarios. A specific user manual is intended to network designers.

2.4 An open tool:

2.4.1 *Evolution (adding / enhancing components):*

Upgrade of the simulator can be done at several levels:

- Evolutions of the standard can be easily inserted, due to the layered and structured modeling approach (sender, transmitter, router function, time-code generator and others are clearly identified and separated bricks within models. Furthermore, transport layer is clearly separated from the protocol and upper application layers.
- New components can be added, using a building blocks approach (bricks re-use).
- New protocols can be easily added, by switching bricks on the dedicated layer.

This task is under responsibility of the development team in charge of maintaining the generic SpaceWire simulation tool.

2.4.2 *Customization to mission:*

Mission designers can build their network topology, configuring all links, but also:

- Nodes models let users customize application behaviors, introducing data generation and consumption models.
- Custom models can be enhanced anytime, such to follow evolutions of the embedded design (incremental approach, refining models).

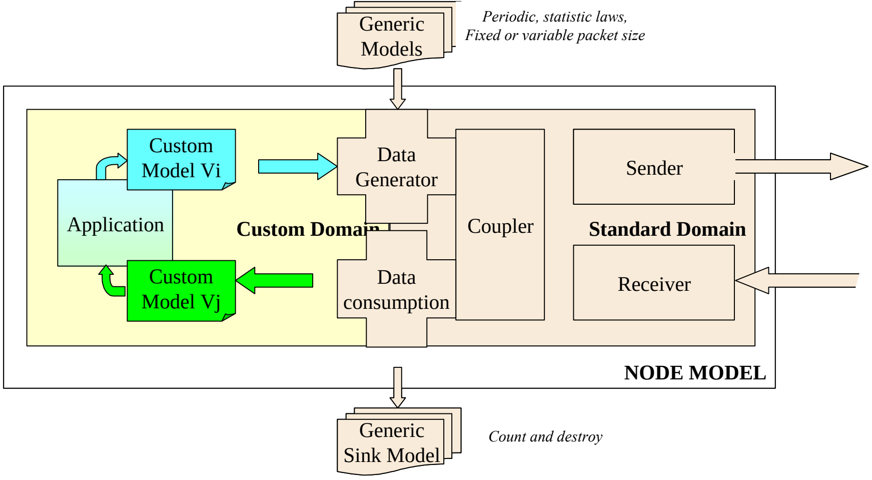


Figure 5: Node is composed of a standard part and a customized application

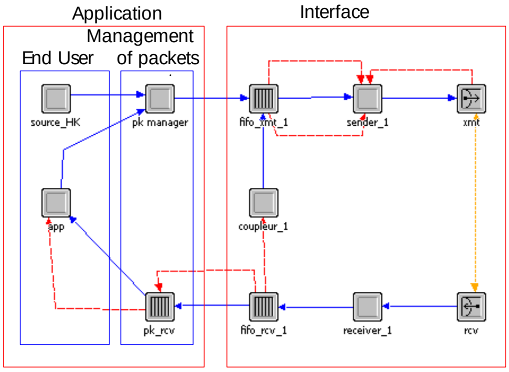


Figure 6: Architecture of node matching concept of Figure 5

Node models can be created such to represent a board function: computer, mass memory, instrument, or any other platform or payload unit. Models can be re-used in other missions (library is enriched each time). This task is realized in OPNET layered environment (see Figure 7), under responsibility of *Node Model developers*. Custom statistics can be defined, attached to these models to be selected at simulation time.

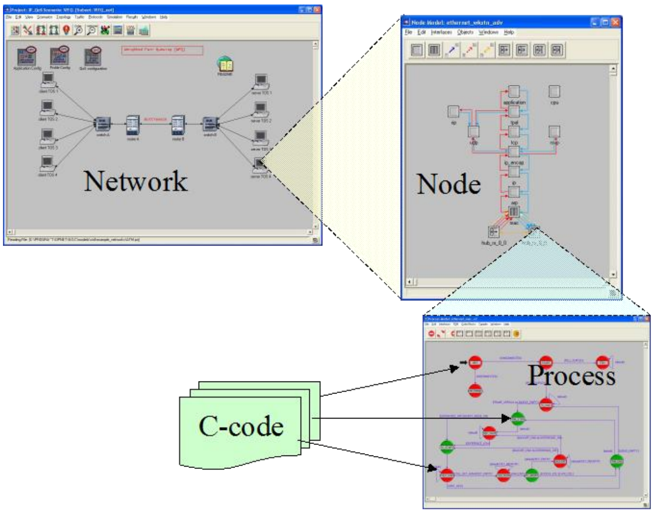
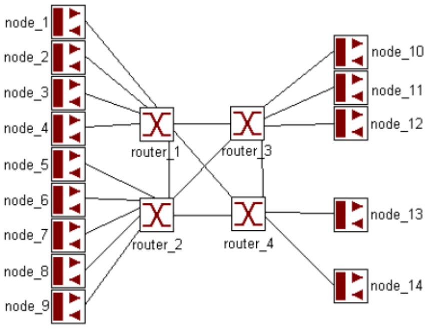


Figure 7: OPNET design layers

2.5 Simulator usage:



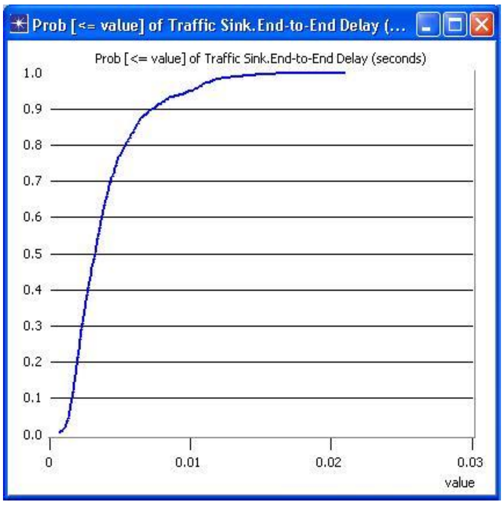
The objective of the simulator is to experiment and verify network designs, by executing mission representative scenarios of traffic load. When the network topology is specified, the designer builds the network model, picking the components in the SPACEWIRE library (including custom nodes) and connecting them by using the graphical editor.

Each network item (node, router and links) can be parameterized, setting values in a pre-defined set of attributes (data rates, routing table, buffer sizes etc …).

Figure 8: SpW network topology (example)

The network model can then be experimented within various scenarios, simulating activities of nodes as sources of traffic. This task is under responsibility of the *Network designer*.

2.6 Expected Results:



At execution time operators select which statistics will be logged.
Traffic analysis and resources consumptions are the main issues of simulation sessions, such to show margins, traffic jam or data loss, effects of anomalies, memory consumption or any other custom statistic attached to specific nodes.
The drawings, created from recorded results, either represents macroscopic statistics or accurate traffic.
Reporting helps designer in verifying the protocol efficiency, the design performance and margins, in refining budgets and identifying slowing sources.

Figure 9: statistics drawing example

An advantage of the simulator is its ability to restitute the traffic of any part of the network, even in contingency cases, which is impossible on actual HW.

Main Benefits are:
<u>At design level:</u>

- To Keep the complexity of SpaceWire traffic under control, by an accurate modeling,
- To get a quick return on design weakness or risks, identifying critical items,
- To verify margins and performances,
- To evaluate resources budgets (buffer sizes). This can be done by 2 ways:
  - infinite resources, showing the exact maximum need
  - sized resources, showing margin or raising error in case of overflow.

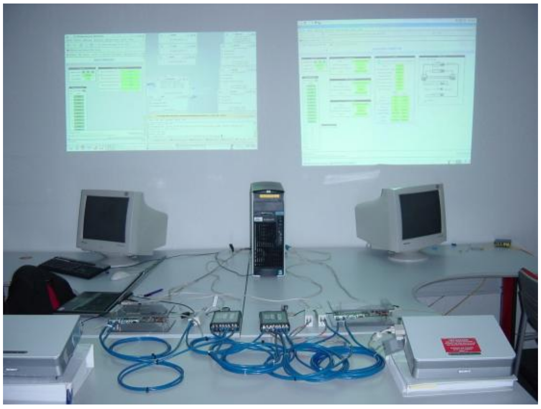
<u>At programmatic level:</u>

- To provide a continuous help in all phases of the development (from phase A to phase E), thru nodes models refinement.

<u>In validation:</u>

- To execute contingency cases or to replay tests found faulty on actual HW.

3. SIMULATOR'S VALIDATION



A particular effort has been put on an incremental test approach, starting with simple use cases of easy understanding, mainly verifying compliance to standard. Cross validation has been done by comparing HW provided results to simulator's.
Some use cases have been selected for functional validation and domain coverage, according to their representative features, concerning functions, architecture and network complexity.
A selection of these tests can be used as regression tests at each release of the simulator, in particular when adapting the simulator to new versions of OPNET environment.

Figure 10 : SW/HW Cross validation

4. APPLICATION DOMAIN

4.1 Expected coverage

The objective is to dispose of a tool able to cover all space applications.

- all families of orbits or trajectories, LEO, MEO, GEO, Lagrange point, inter-planetary, landers, rovers and robotics.
- All domains: observation, science, exploration, telecommunication, etc…

That’s why next work covers simulation of 4 typical and realistic use cases, respectively belonging to interplanetary, LEO observation, exploration and robotics domains.

4.2 First use case: interplanetary missions

This use case was proposed for an orbiter, saving science data from multiple instruments within a mass memory, and restituting these later on in X-band or Ka-band TM channels. The interesting point is that some auxiliary data have to be inserted within the science flow, which is done by the SSMM controller (Mass Memory) or PM (Processor Module). This kind of features is impacts fluidity of the traffic on the network.

Figure 11 is a limited scenario with science flow (in red) and C/C flow from PM (in blue).

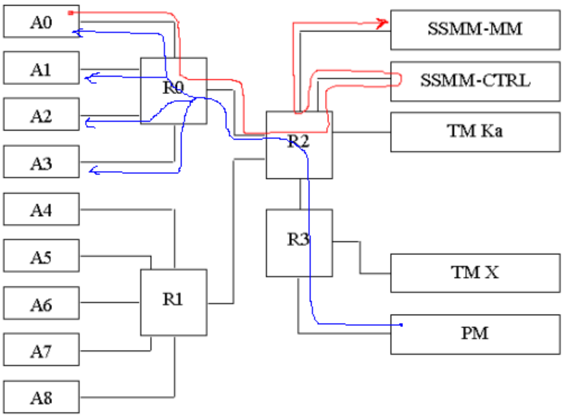
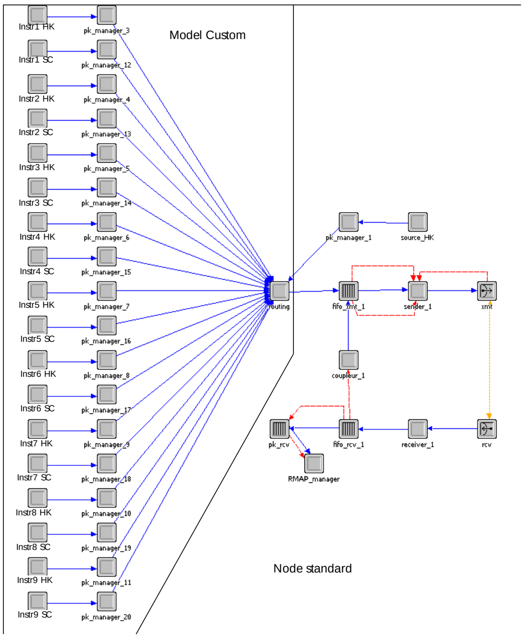


Figure 11 : Network for orbiter with 8 instruments A1 to A8



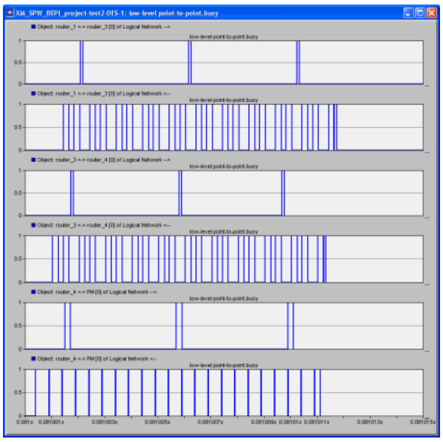

XM_SPW_BEPI_project-test2-DES-1: low-level point-to-point.busy
Object: router_1 <-> router_3 [0] of Logical Network -->
low-level point-to-point.busy
Object: router_1 <-> router_3 [0] of Logical Network <--
low-level point-to-point.busy
Object: router_3 <-> router_4 [0] of Logical Network -->
low-level point-to-point.busy
Object: router_3 <-> router_4 [0] of Logical Network <--
low-level point-to-point.busy
Object: router_4 <-> PM [0] of Logical Network -->
low-level point-to-point.busy
Object: router_4 <-> PM [0] of Logical Network <--
low-level point-to-point.busy
1
0.5
0

Figure 12 : Small extract of results obtained for scenario of Figure 11

Figure 13 : Custom model of the SSMM controller node

4.3 Second use case: LEO observation

This use case is Sentinel-3 application, involving 4 instruments, a PDHU as mass memory and the central computer of the platform. This is a classical configuration of embedded network.

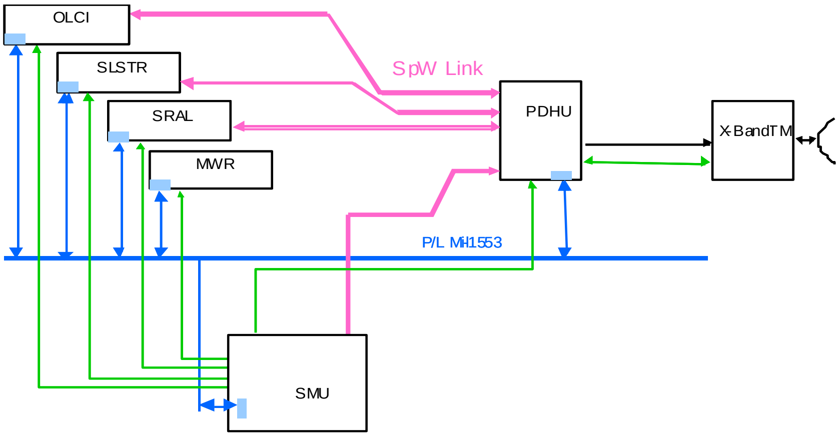


Figure 14: Sentinel 3 SpaceWire network

4.4 Third use case: exploration domain

The network of Figure 15 is issued from soft autonomous landing mission, requiring sensor data in a continuous way, at a high rate from specific complex sensors. The data load is far beyond the performance of classical data bus, as it requires imaging units (optical or LIDAR). The fail-op nature of the landing phase requires hot redundancy, making double complexity for the network.

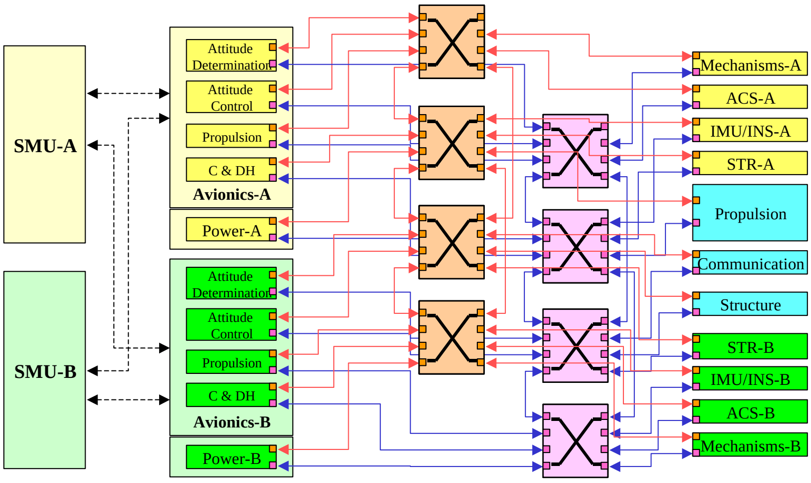


Figure 15: Typical exploration use case - Soft landing mission

4.5 Fourth use case: robotics

The network on Figure 16 is issued from robotics space application, requiring visual monitoring sensors and motors encoders as actuators.

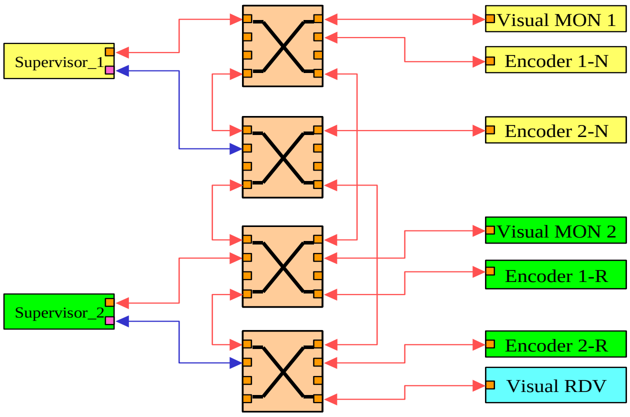


Figure 16 : Space embedded Robotics network

5. AND NEXT:

Taking advantage of new modules of last releases of OPNET environment, TAS is now on the point to investigate the possibility to connect the numerical simulator to HW sub-network, making a numerical / HW hybrid test configuration.
The main objectives are :

- to start early with verification of functional behavior of actual HW, even partly,
- to save planning in case of late delivery of design parts,
- to replace some part of the network by models, injecting easily traffic anomalies to verify contingency cases, or to check the behavior of actual HW in these cases,

Furthermore, TAS has been selected by ESA for a study, aimed at provide a SpaceWire simulator, which requirements cover widely the MOST TAS perimeter. SOW requires explicitly to study feasibility of hybrid simulation.

6. CONCLUSION:

TAS expressed the need for a SpW simulator in early 2006, with questions raised on validation of embedded network, considering such design more complex than classical data bus or single node to node links.
This led to the development of the first version of in-house simulator MOST, enriched later on with time-code and more complex features as GAR or RMAP. OPNET environment was chosen, because dedicated to network simulation, and because of its Man Machine interface.

The simulator has been experienced on several use cases, in the frame of nominal and contingency scenarios. Results have shown the usefulness of such a tool, and MOST could be the root for industrial simulation product.
This supposes to enhance models, to add new building blocks, to complete the formal validation with reference use cases, to complete documentation and finally ensure the maintenance, by in particular following main release of OPNET environment.

A door is opened towards the hybrid network test bench, involving HW sub-networks and numerical sub-networks. Such approach looks possible with modules of more recent OPNET versions, but requires additional feasibility analysis.

The growing implication of SpaceWire standard in board-to-board communications since first version of MOST has never been denied, even if slower than expected. The enhancement of performances of embedded avionics (more resources) lead to more ambitious performance requirements, including higher data rates, hence more complex communication networks and greater volumes of exchanged data.

Validation means shall follow this evolution to keep control and quality of designs of more complex embedded systems, communications being a critical issue. Numerical simulators are a part of the response, as HW means will be expensive and difficult to set in use early.