SpaceWire Network(s)
&
On-Board Realt-Time Aspects

Wahida Gasti
wahida.gasti@esa.int

AGENDA

- Real-time System
- On-Board Real-Time Systems: RT-CDMS
- RT-CDMS Pilars: OS and TDMA for RT-CDMS
- Classical RT-CDMS
- RT-CDMS based on SpW Network
- Requirements for RT-CDMS based on SpW Network
- RT-CDMS and Cluster SpW
- Conclusion: Present Status and evolution for SpW Network

Real-Time Systems require not only logical Functional correctness, but also *timing correctness*.

The timing constraints of a real-time system are often specified as *deadlines*.

It is desirable (required in many cases) for the system to respond within the *specified deadlines*.

If such a condition is violated, therefore this property can be regarded as a *non-functional property of Real-Time Systems.*

RT-S provides facilities which, if used properly, guarantee deadlines can be met generally *(soft real-time)* or deterministically *(hard real-time).*

2 Categories of *On-Board Real-Time Systems*:

- Platform Systems: *Real-Time Control and Data Management System RT-CDMS)*
  - Central Computer controlling the satellite, sending commands to the satellite units and collecting data from units and instruments
- Payload Systems: Instrument application driven

Scope of this presentation is to focus on : *RT-CDMS*

- Requirement *is to respond to commands and to control* the satellite within *specified deadlines*.
- If such a condition is violated, then it might cause a *satellite failure*.

Therefore, *Predictability* is a major characteristic for RT-CDMS.

RT-CDMS uses SW based on Real-Time Operating System (RTOS-CDMS) with specialized *scheduling algorithms* in order to produce *deterministic behavior* in the final system

Static schedulability analysis is used to predict whether the timing constraints of RT-CDMS is met or not

RTOS-CDMS is valued more for how quickly and predictably it can respond to a particular deadline than for the given amount of work it can perform over time.

Key factors in an RTOS-CDMS are:

- *minimal interrupt latency*
- *minimal thread switching latency*

There are two basic approaches for handling tasks in real-time applications:

- The Event-Driven approach (ED): task(s) initiated by particular event.
  - Event-driven design (*priority scheduling*) – Tasks switch only when an event of higher priority needs service: *pre-emptive priority*.

- The Time-Driven (TD) approach: task(s) initiated at predetermined points in time or during slots in time.
  - Time-Driven design (*Time sharing*) - Tasks switch on a clock interrupt, and on periodic events: *round robin*

Same duality is reflected at communication infrastructure level.
Communication activities can be activated:

- either *dynamically*, in response to an event,
- or *statically*, at predetermined moments in time.

Fundamental problem in RT-CDMS:
Sharing of communication resources between message/packet streams on different entities (e.g. sub-systems, units,modules..etc) such that *Hard-Real-Time respect of deadlines are satisfied and predictable.*

Systeme *predictability needs* to be assessed thus *tested*

Fundamental solution to the problem:
Using *Time-Division Multiple Access* (TDMA) communication protocols by assigning messages/packets to time slots such that *no two entities transmit at the same time and queuing delays of messages/packets are bounded.*

Typically these communication protocols operate on the basis of TDMA cycles, where a node is assigned one or many time slots. Usually each slot has a fixed length and the number of slots per cycle is fixed *(Isochronous slotting).*

TDMA cycle has a fix and known time duration, and *upper bounds on packets' queuing delays can be proved*.

System Communication Predictability is insured with TDMA approach

TDMA research work address the problem of finding appropriate schedules (*TDMA frames/templates*) for guaranteeing *timeliness* to real-time message streams

The flexibility in assigning time slots to an entity comes at a price:
an *unused slot is wasted* (i.e. cannot be used for any other hard real-time traffic)

A message/packet stream with *periodic messages/packets* may need a specific time slot in a TDMA cycle during only a few TDMA cycles.

In the other TDMA cycles this time slot is not used and hence wasted.

Ways to overcome this waste is:

- to have a large TDMA cycle serving several instances of a packet tream. In the extreme case may need to choose the length of a *TDMA cycle to be the least-common multiple of periods* to avoid wasted slots.
- to have a *large TDMA protocols with slot skipping (TDMA/SS):* a slot is skipped when it is not used, hence the next slot can start earlier and this reclaims time for hard real-time traffic.

Real-Time Computational model attributes :

- Components participating to the real-time behaviour are
  - *Active objects (sporadic and cyclic),*
  - *Protected objects,*
  - *Passive objects.*
- The scheduling type is *multi-threaded*,
- The scheduling model is *pre-emptive fixed priority* based,
- The analytical model is *Deadline Monotonic Scheduling*,
- The mean of communication between tasks and objects is message,

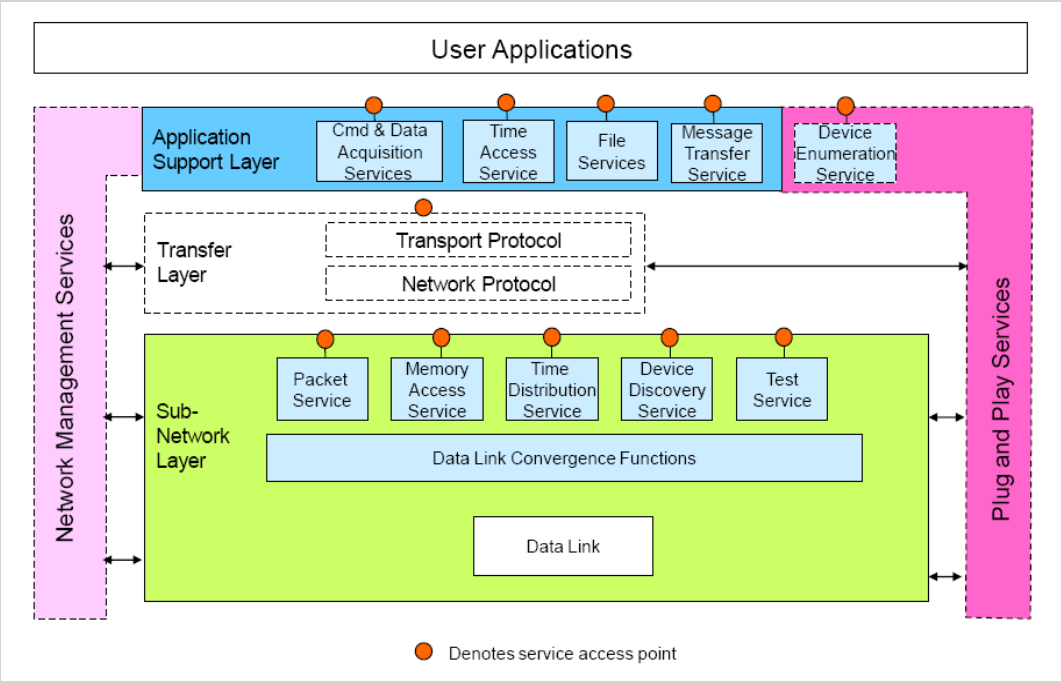
Communication ressource characteristics:

- Generally based on *MIL-1553B bus*
- Bus data rate ~ I Mbps

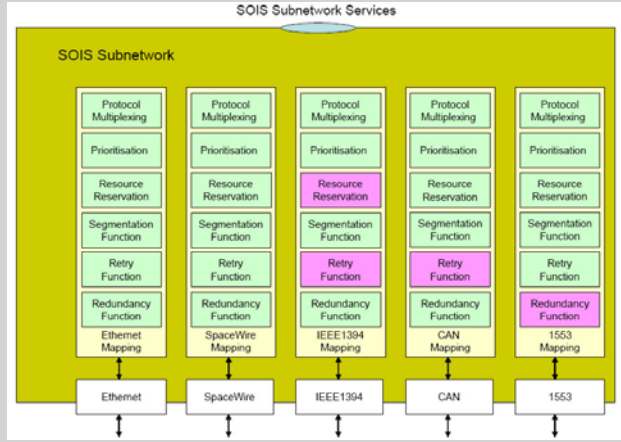
- Classical RT-CDMS: *RT-CDMS_**1553B***

Goal is to facilitate the systematic design of *dependable RT-CDMS* out of:

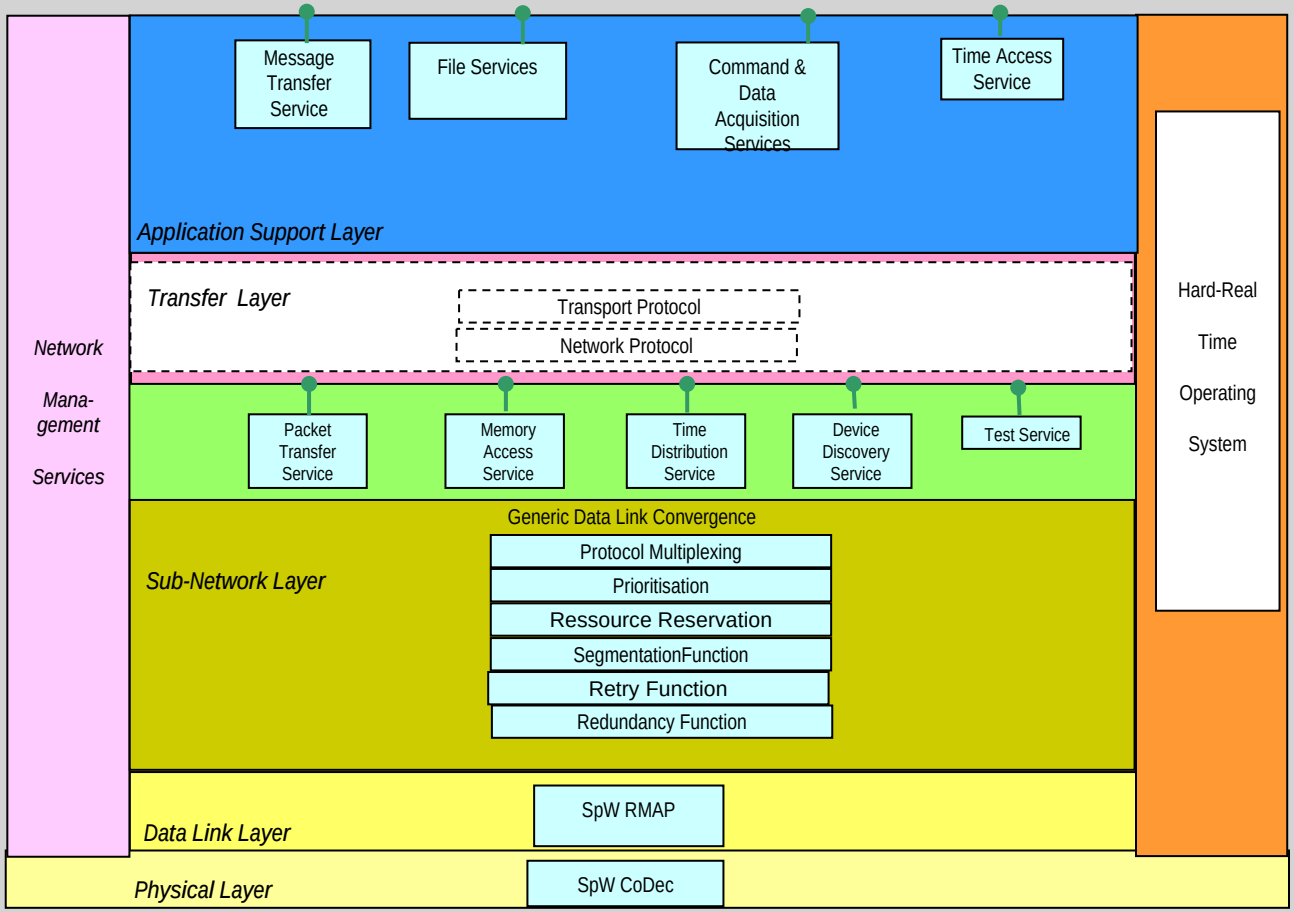
- HW Architecture: Modules (or nodes) connected via SpW network.
- SW Architecture: *SOIS architecture with SOIS QoS*



QoS (see SpW for Cmd&Ctrl presentation D.Jameux)



- **Application Support Layer (ASL) Heritage: *RT-CDMS_SpW* shall benefit from previous *useful and efficient* ASL SW development made for *RT-CDMS_1553B***



RT-CDMS_SpW

Requirements for *RT-CDMS_SpW* are:

- RT-CDMS_SpW is *Dependable* (RAMS: *Reliable, Available, Maintainable and Secure*)
- Heritage from Present RT-CDMS ⇒ Real-time communication exchange of *SpW packets* based on *TDMA protocol*
- RT-CDMS_SpW *scalable* and can be built *cost-effectively*

Composability: RT-CDMS_SpW must provide a framework for the systematic construction of operational SpW interfaces:

- a system out of subsystems and
- sub-systems out of modules.

Property Match for HW & SW Interfaces: SpW modules must comply with the system architectures (HW and SW) to avoid a *property mismatch* at the module interfaces.

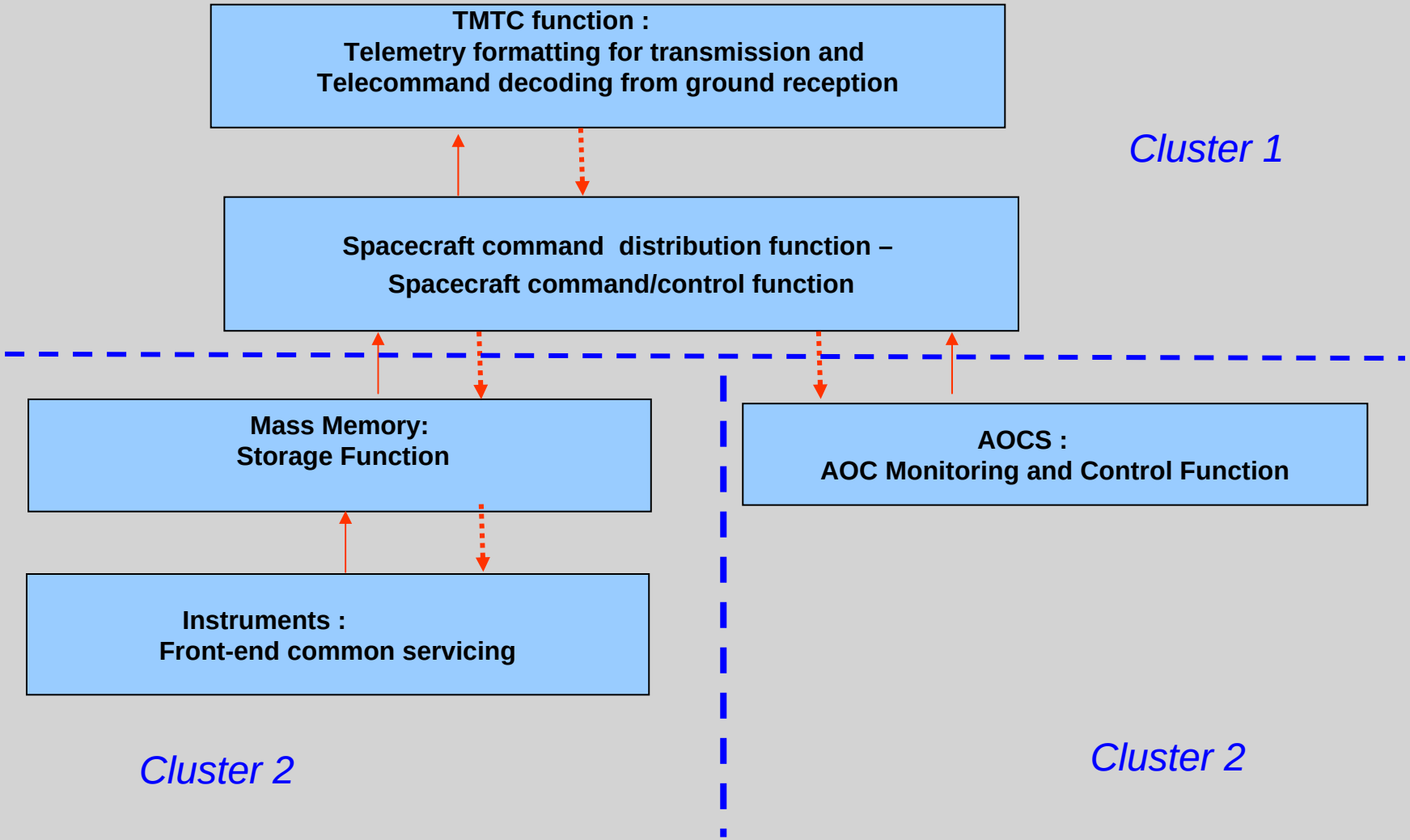
Elegance: RT-CDMS_SpW shall constrain the implementation in such a way that:

- The ensuing *system is RAMS compliant, scalable and can be built cost-effectively*
- The *system is elegant as it is representative in fit, form and functions.*

Composability requires:

- Investigate the *command flow* among modules
- investigate the *data flow* among the modules
- Decompose the design problem into SpW cluster networks based on SpW modules
- Allocate functions to modules:
  - *Standardization of building SpW modules* (make use of heritage, ease, maintainability, increase robustness)
  - *Reduction of modules for higher integration* (increase robustness and reliability, reduce integration time and cost)
- Analyse the implementation of redundancy ((reliability, robustness, maintainability) by:
  - *Interfaces cross coupling*
  - *Cascaded levels of redundancy*

Composability: Cluster in SpW Network 2/3



Cluster Concept in SpW Networks **must be introduced:**

Example: Central Computer, AOCS, MMS & PayLoad servicing

Cluster SpW network - Building cluster SpW network shall be possible out of :

- pre-validated newly developped modules
- and/or re-used modules with fully *operational interfaces* (i.e. in the temporal domain and in the data domain)

Composable Cluster must support:

- *Independent development of modules* related to the function
- *Stability of prior services* related to the modules and the system architectures
- *Performance of Real-Time communication System*
- *Deterministic Replication* to support fault tolerance implementation.
- *Testing and Diagnostics* It shall be possible to identify the packet errors and its source entity (i.e bubbling idiot). It shall be contained to its source entity if not possible to remove it.

Firewall concept must be introduced as: command flow and data flow shall be restricted to specific *Cluster SpW Network*

The firewall concept shall support the composability requirement.

Property Mismatches at Interfaces shall solve:

Syntactic	Big endian v.s little endian for data
Flow control	Data Flow push or data pull
Incoherence in naming	Same name for different entities (get/set) Different names for different entities (message/packet)
Data representation	Different styles for data representation Different formats for date
Temporal	Time distribution v.s time synchronisation, Inconsistent time-outs,
FDIR	Different failure mode, Different recovery status
Semantics	Differences in the meaning of the data

SpW packet is an atomic data structure that is formed for inter-module communication.

In a Cluster SpW Network: The endpoints of the communication are the module interfaces.

In the temporal domain, a SpW Packet can be characterized by

- The packet send instant: The instant when EOP leaves the sender.
- Or The packet receive instant: The instant when EOP of the packets arrives at the receiver

For the flow control, a SpW Packet can be characterized by

- Information Push Interface: SpW packet source write packet on destination
- Or Information Pull Interfaces: SpW packet destination read packets on source (packet producer)

What is better for RT-CDMS-SpW?

Valid question to be solve at cluster level and considering backplane issues

RT-CDMS-SpW System design considerations:

- RT-CDMS-SpW requires SpW Network Topology - *RT-CDMS-SpW backplanes* - SpW is not a multi-drop bus hence not oriented direct Cross strapping
- RT-CDMS-SpW Master for
  - Time generation and distribution (*SpW Time codes* or not) to SpW clusters and related modules
  - RT-CDMS-SpW Network configuration and re-configuration aspects
  - RT-CDMS-SpW Network monitoring -- Reliability
  - RT-CDMS-SpW SpW network Fault tolerance rules
  - RT-CDMS-SpW FDIR (HW and SW mechanisms for fault detection & isolation– HW mechanisms for recovery)

Conclusion

Present SpW Stds;

- ECSS-E-50-12A
- ECSS-E-50-11

Present SpW Standards lacks Real-Time mechanism for *RT-CDMS$_{-SpW}$* :
They do not offer any guaranty for SpW packet delivery with specified deadlines.

First but not last work to be achieved

Higher SpW Protocols need to incorporate TDMA protocol to insure Real-Time correct functioning, predictability and SW heritage requested by RT-CDMS.