Use Case Study for SpaceWire-RT

(Revised Version)

Takahiro Yamada (JAXA/ISAS)

20 February 2008

Tenth SpaceWire Working Group Meeting

ESTEC, Netherlands

Purpose of This Presentation

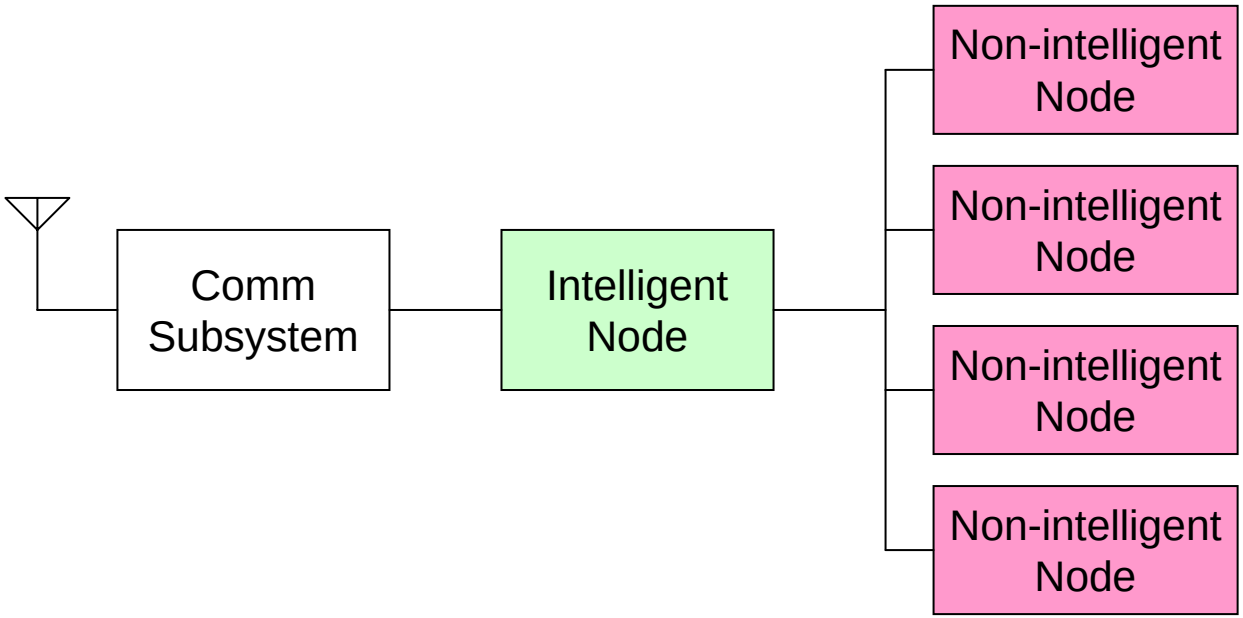
- This presentation shows possible use cases for SpaceWire-RT in the standard onboard data systems architecture that is being developed by JAXA.
- This presentation has two parts:
  1. Standard onboard data systems architecture of JAXA
  2. Use cases for SpaceWire-RT in this architecture

Part 1
Standard Onboard Data Systems Architecture of JAXA

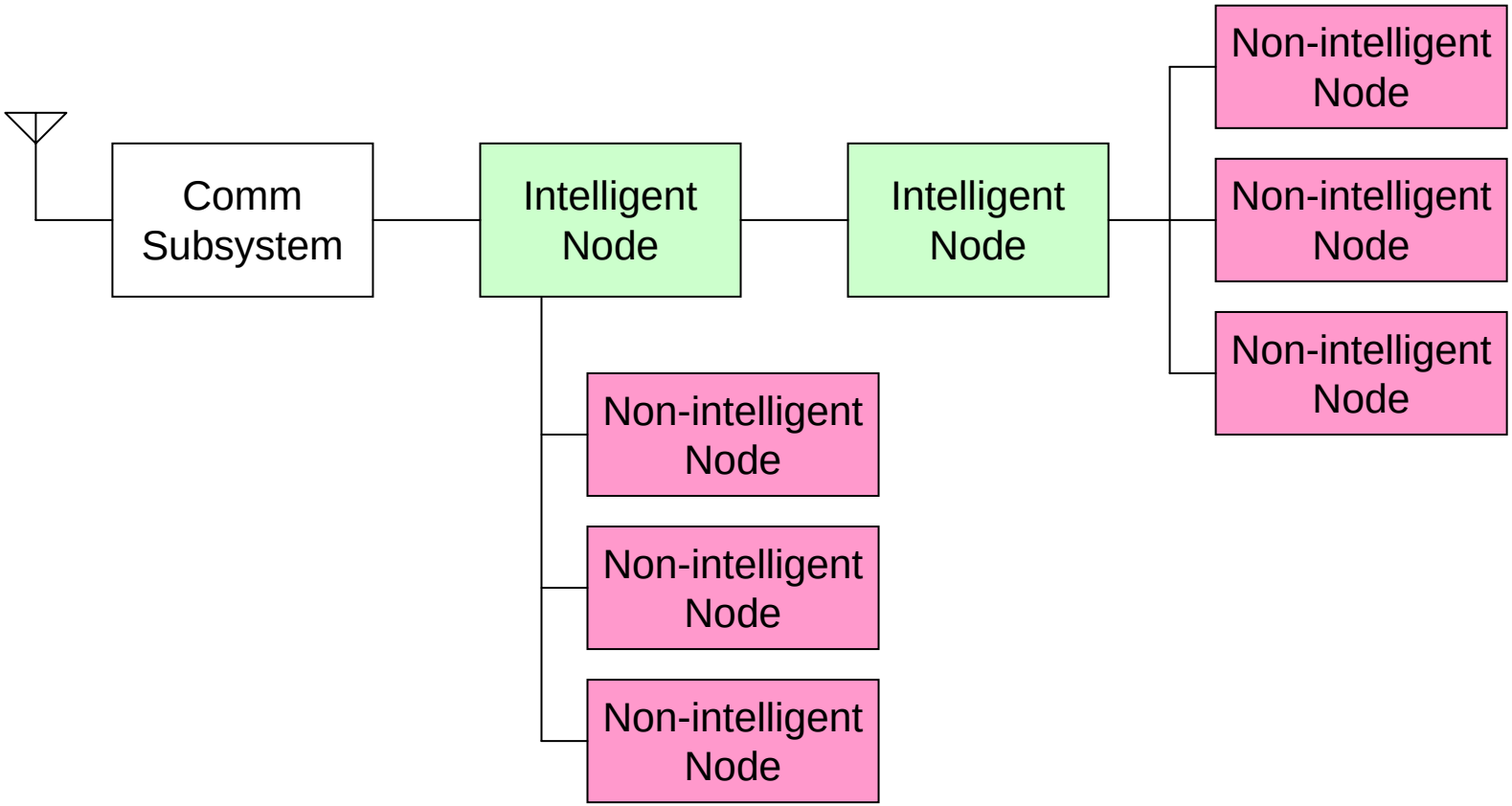
Onboard Data Systems Architecture of JAXA

- JAXA is developing an onboard data systems architecture that can be used for many different types of spacecraft and onboard instruments.
- Onboard data systems will consist of the following hardware elements (or Nodes):
  - Intelligent Nodes: Standard CPU box (SpaceCube) with SpaceWire interfaces
  - Non-intelligent Nodes: Sensors/Actuators + Standard I/O boards and/or ADC/DAC boards with SpaceWire interfaces
- The architecture supports the following standard protocols:
  - SpaceWire
  - Remote Memory Access Protocol (RMAP)
  - (Possibly) SpaceWire-RT
  - CCSDS Space Packet Protocol (SPP)
  - Spacecraft Monitor and Control Protocol (SMCP, JAXA standard)

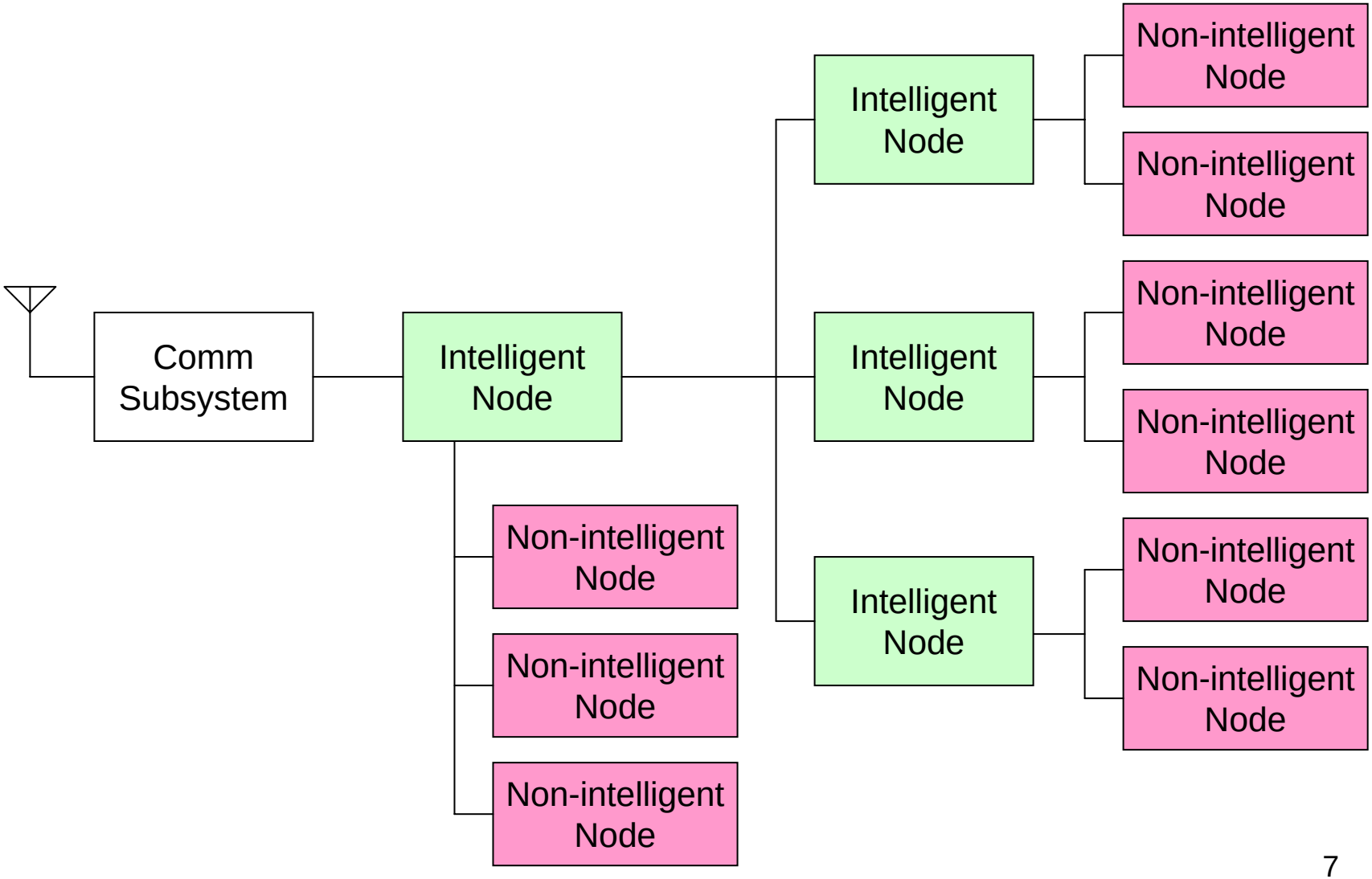
Typical System Configuration 1



Typical System Configuration 2



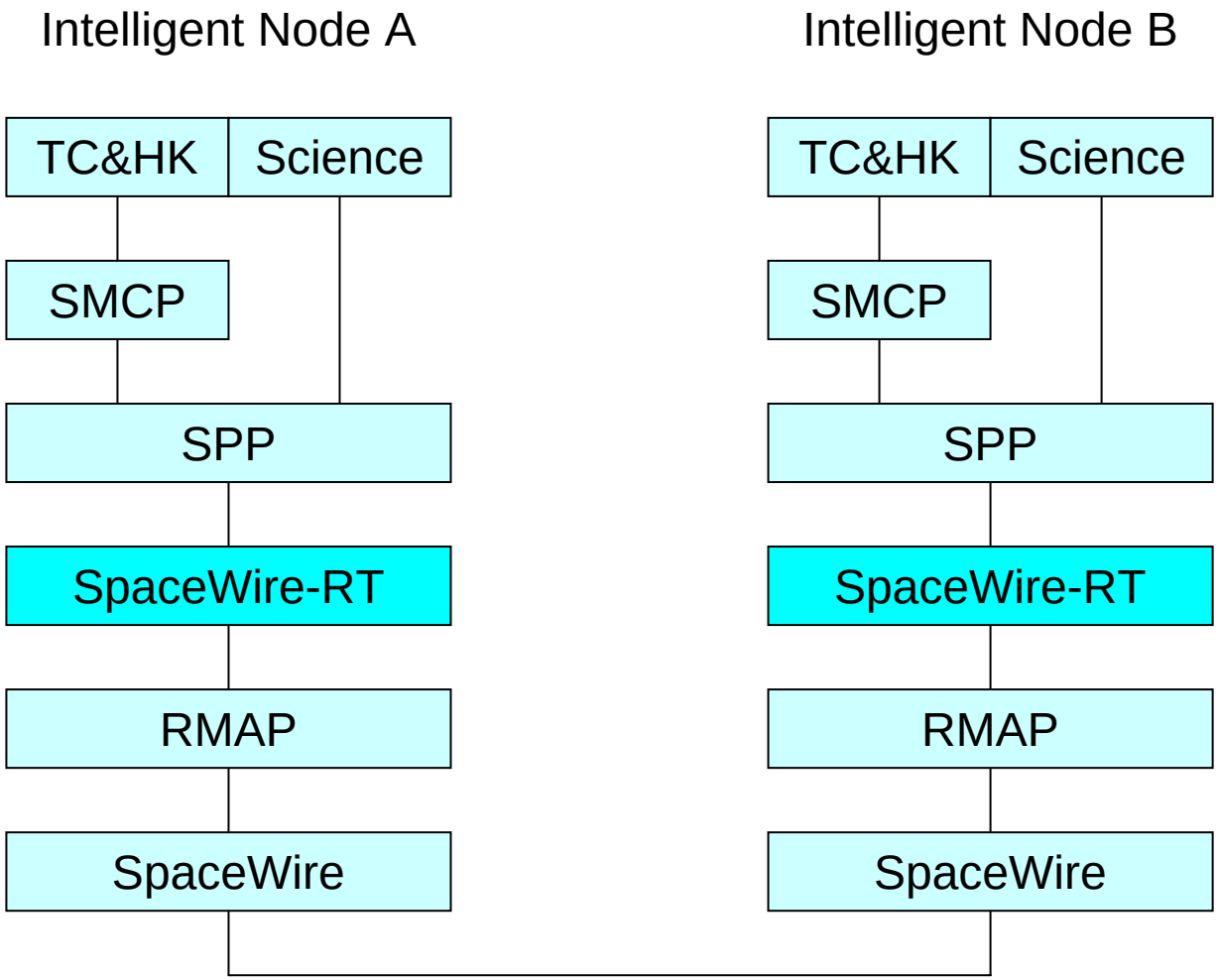
Typical System Configuration 3



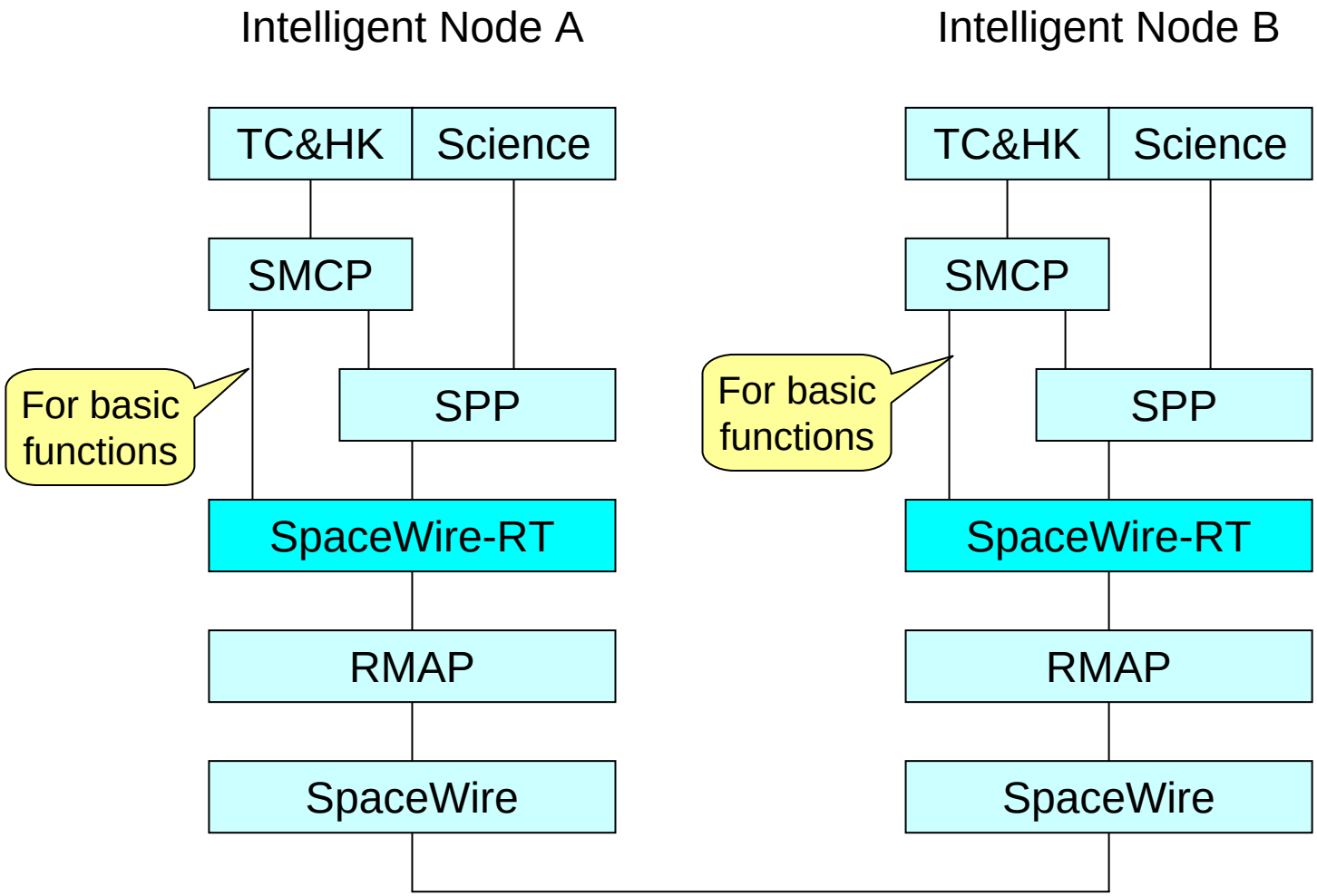
Protocol Configuration Principles

- SpaceWire is always used as the standard data link and physical layer protocol.
- RMAP is always used to standardize the interface between SpaceWire networks and Nodes.
- Between Intelligent Nodes, the Space Packet Protocol (SPP) is used as the standard network protocol and TM data is compressed or processed, but SPP may be bypassed for some basic functions performed by hardware.
- Between Intelligent and Non-intelligent Nodes, the Space Packet Protocol is bypassed and TM data is not compressed or processed.
- For commanding and housekeeping telemetry, the Spacecraft Monitor and Control Protocol (SMCP) is used as the standard application layer protocol.

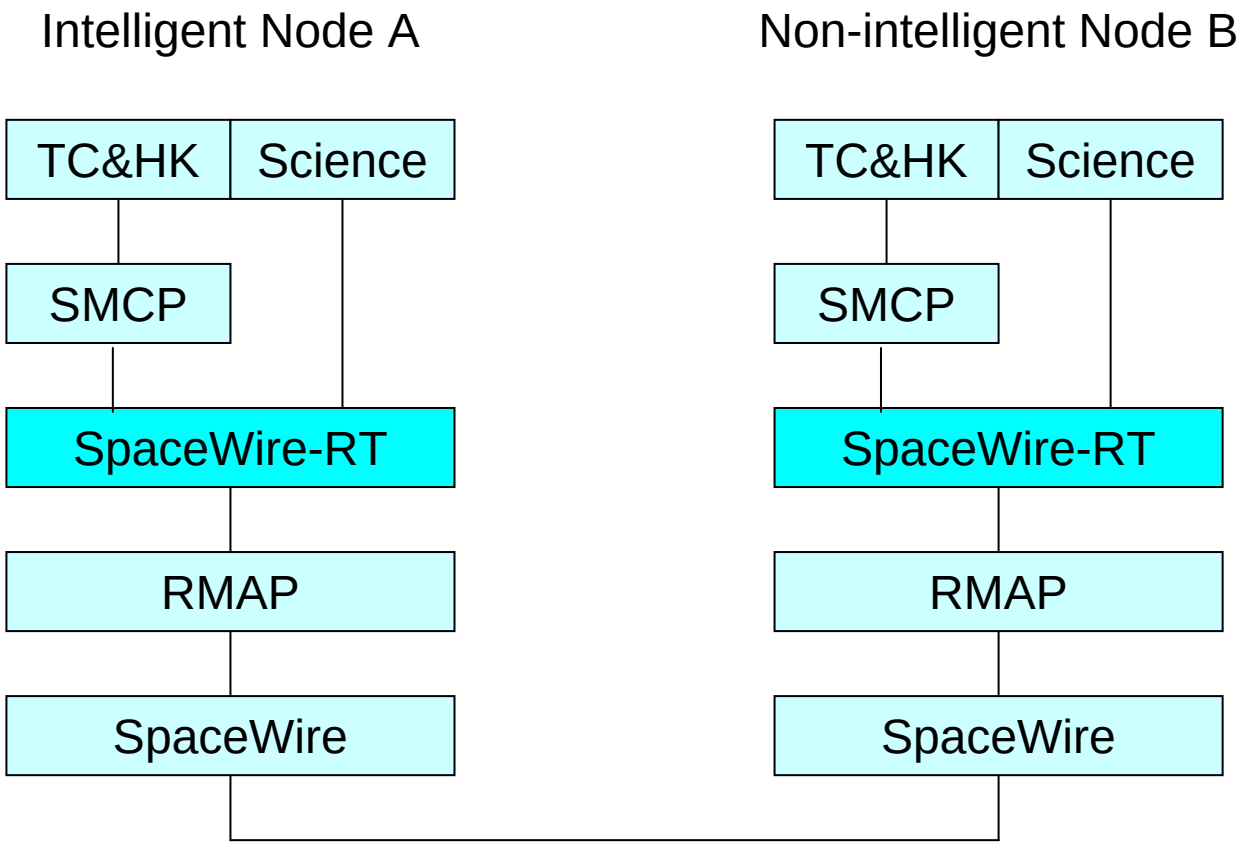
Typical Protocol Configuration 1



Typical Protocol Configuration 2



Typical Protocol Configuration 3



Clock Synchronization

- Clock synchronization will be done by sending, with RMAP-write from the central Node to the other Nodes, the value of the spacecraft master clock of the time at which a certain SpaceWire time code (e.g., with value 0) was (or will be) generated.

Part 2
Use Cases for SpaceWire-RT

QoS Requirements for Typical Data Types

Data Type	Latency Requirement	Reliability Requirement
Clock	High	Low
TC	High	High
HK TM (Raw)	High	Low
HK TM (Differencial)	High	High
Science TM (Raw)	Low	Low
Science TM (Compressed)	Low	High

Necessary Mechanisms

Data Type	Latency/Mux Mechanism	Reliability Mechanism
Clock	TDM or Priority	Best effort
TC	TDM or Priority	Assured
HK TM (Raw)	TDM or Priority	Best effort
HK TM (Differencial)	TDM or Priority	Assured
Science TM (Raw)	TDM or BW Reservation or Priority	Best effort
Science TM (Compressed)	TDM or BW Reservation or Priority	Assured

QoS Class Mapping

Data Type	QoS Class
Clock	Resource Reserved with TDM or Best Effort with Priority
TC	Guaranteed with TDM or Assured with Priority
HK TM (Raw)	Resource Reserved with TDM or Best Effort with Priority
HK TM (Differencial)	Guaranteed with TDM or Assured with Priority
Science TM (Raw)	Guaranteed with TDM or BW Rsrv. or Assured with Priority
Science TM (Compressed)	Guaranteed with TDM or BW Rsrv. or Assured with Priority

Considerations

- The SOIS concepts can be used as the framework for defining QoS requirements.
- We support the analysis behind the SPaceWire-RT requirements document edited by the UoD.
- There are multiple ways to realize latency control and multiplexing (i.e., priority, TDM and BW reservation). It seems like TDM is the simplest way for meeting different latency and multiplexing requirements.

Conclusions

- For most networks, we will use:
  - Resource-reserved service using TDM (possibly with priority)
  - Guaranteed service using TDM and retry (possibly with priority)
- For simple networks, we may use:
  - Best effort service (possibly with priority)
  - Assured service using retry (possibly with priority)