Plug-and-Play for SpaceWire: Feedback and User Needs Review

Peter Mendham

24 March 2011

Agenda

- User needs for a SpaceWire “plug-and-play” protocol
  - Collated from feedback
  - Offered by AFRL and GSFC
- Summary of feedback on the SpaceWire-PnP draft specification
- Use cases for SpaceWire-PnP

User Needs Sources

- Contributions made in this working group (2006 onwards)
- Contributions made at SpaceWire conferences (2007 onwards)
- Teleconferences organised by Glenn Rakow (GSFC) 2006-2007
- Feedback on the SpaceWire-PnP draft Protocol Specification

Why “Plug-and-Play”?

- “Why can’t we have a standard way to configure routing tables?”
- “Isn’t there a standard way to set the link speed?”
- “How can I find out what devices there are on a network?”
- “Can I check if my device is still there?”
- “I have my own standard for working with my devices, can I indicate that a device supports this in a standard way?”

scisys

User Needs – Goals

- A standard way to configure standard functions
- Detect all devices and their type
- Discover the topology of a network
- Obtain the status of a device
- Detect the services that a device provides
- Fulfil requirements for CCSDS SOIS DDS

User Needs – More Detail

- Must be consistent with the SpaceWire standard
- No topology restrictions
- No restrictions on standard-compliant devices
- Implementation method should not be restricted
  - e.g. software vs. hardware
- Everything except basic identification should be optional
- Must be simple to detect supported features
- Must be extensible for vendor-specific functions

GSFC / AFRL User Needs

scisys

SpaceWire-PnP Feedback

- Thanks to those who provided feedback
- Collated feedback is available as a support document to this WG
- ~23 comments made

- Lack of clarity – more examples needed
- Overly restrictive
- Non-standard features included
- No clear way to deal with optional functions
- No clear way to add vendor-specific functions

Lessons Learned: Dealing with Optionality

- Every function must be optional
  - Except for identifying the device
- Must be a clear way to offer a function
  - And to detect if that function is present
- If a function is not offered this must not require any implementation
- It must be easy to add vendor specific functions
- Vendor-specific functions must not be restricted by standard functions
  - Or vice-versa

Plug-and-Play and Standardisation

- Spectrum of possible approaches to devices:

No standardisation | Complete standardisation

- No device standardisation
  - Requires no changes to existing devices
  - Requires a device driver for every device
  - Device could be described in an electronic data sheet
  - Small amount of standardisation would permit device identification

- Complete standardisation
  - All devices must support a single standard interface
  - Device interface must cover all possible device features
  - Single standard device driver

SpaceWire-PnP and Standardisation

- Provide standard mechanisms for configuring functions identified in the SpaceWire standard
  - Supported by a standard device driver
- Configuration mechanism designed to be as generic as possible
  - Whilst considering implementations
- Everything other than basic identification is optional
- Mechanisms present for adding vendor-specific configuration functions
  - Requires a device driver

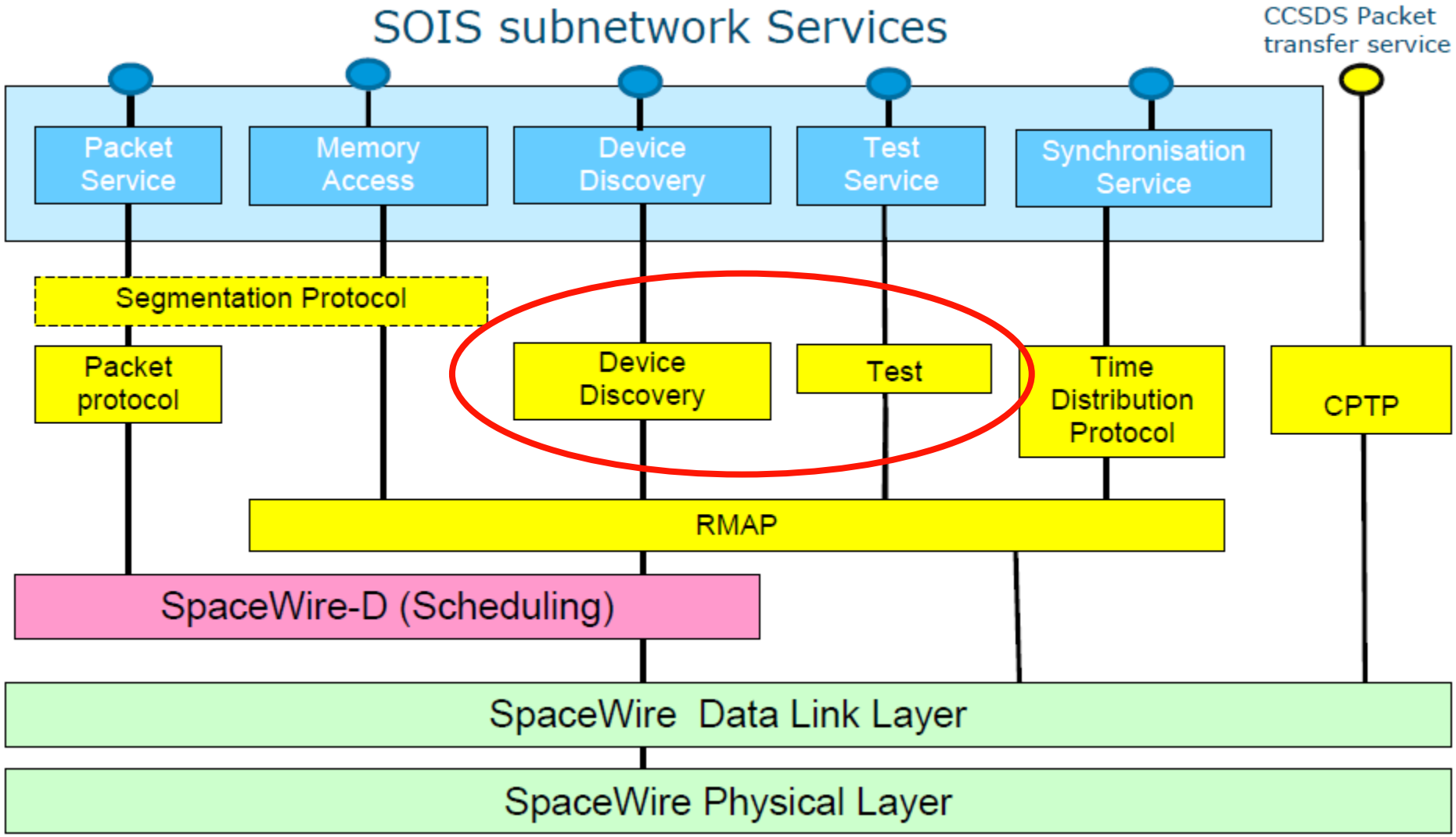
Lessons Learned

- Restrictions are a problem
  - Topology
  - Timing
- And largely unnecessary

- Non-standard features are a problem
- And largely unnecessary
  - Features to be included as part of SpaceWire Evolutions are OK(?)

SpaceWire-PnP and SOIS

- SOIS provides a reference architecture for SpaceWire protocols
- A plug-and-play protocol
  - Must satisfy the requirements for the Device Discovery Service
  - Should provide facilities for the management of SpaceWire network resources
- SpaceWire-PnP provides
  - All network management functions – but does not impose a network management policy
  - Implementation for Device Discovery Service and Test Service


SOIS subnetwork Services
CCSDS Packet transfer service
Packet Service
Memory Access
Device Discovery
Test Service
Synchronisation Service
Segmentation Protocol
Packet protocol
Device Discovery
Test
Time Distribution Protocol
CPTP
RMAP
SpaceWire-D (Scheduling)
SpaceWire Data Link Layer
SpaceWire Physical Layer

Using SpaceWire-PnP (1): Minimal

- Device information, network ID and link activity together permit device identification and network discovery
- Minimal implementation requirements:
  - 12 words of read-only constant registers
  - 1 read-only dynamic register
  - 1 read-write register
- Minimal set of primitives
  - 5 pairs (request/indication)

Using SpW-PnP (2): Datasheets

- Can use data source capability service to describe an RMAP region to read a datasheet from
  - E.g. direct interface to a PROM
- Data source type identifies format of datasheet
  - E.g. xTEDS
- Minimal implementation (in addition to previous)
  - 8 read-only words
  - 2 primitive pairs
- Uses the same RMAP core as for SpaceWire-PnP

Using SpW-PnP (3): RMAP Spaces

- Can use data source/sink capability services to describe an existing RMAP address space
  - E.g. JAXA standardised memory map
- Same resource requirements as datasheet example for read-only
  - Add 8 read-only words and 4 primitive pairs for read-write
  - This adds a data sink
- This is the suggested way to describe an existing memory space

Using SpW-PnP (4): Notification

- Ability for routers (or any device) to automatically inform a network manager when status changes
  - E.g. link connect/disconnect
- Uses a simple data source
- Additional requirements (from datasheet case):
  - 1 read-write field for a target source
  - 12 read-write fields for an initiator source
- Features to support multiple, uncoordinated network managers are documented

Conclusions

- A plug-and-play standard must fulfil two major goals:
  - Device discovery
  - Standard configuration
- Clarity, simplicity and extensibility are important
- Must not limit application of the SpaceWire standard