The SpaceWire-PnP Draft Standard

Peter Mendham

Stuart Mills, Steve Parkes, Martin Kelly, Stuart Fowell

Agenda

- The draft standard
  - Conceptual view of a network
  - SpaceWire Network Management
  - Architectural approach
- Details
  - Communications protocol
  - “Configuration space”
- An example device
- Conclusions

Network Discovery Protocols

- ESA TRP Activity investigating SpaceWire Network Management (Plug-and-Play)
- Project elements:
  - Requirements gathering
  - Protocol design
  - Protocol specification (draft ECSS standard)
  - Prototyping and validation
  - Demonstration
- An input was the previous draft protocol specification

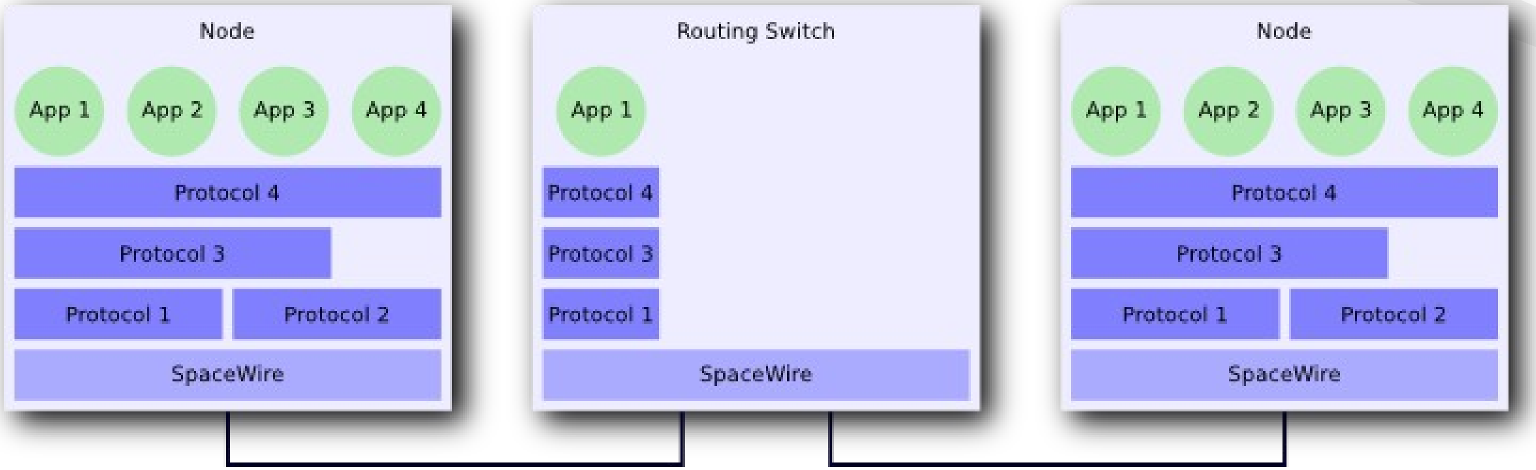
The SpW-PnP Draft Standard

Protocol Design

- Requirements for a plug-and-play protocol were gathered from various stakeholders
  - SCISYS
  - STAR-Dundee
  - TAS-F
- The protocol design was considered afresh based on requirements
- Inspiration was taken from the previous protocol specification where appropriate
- There are a number of key differences

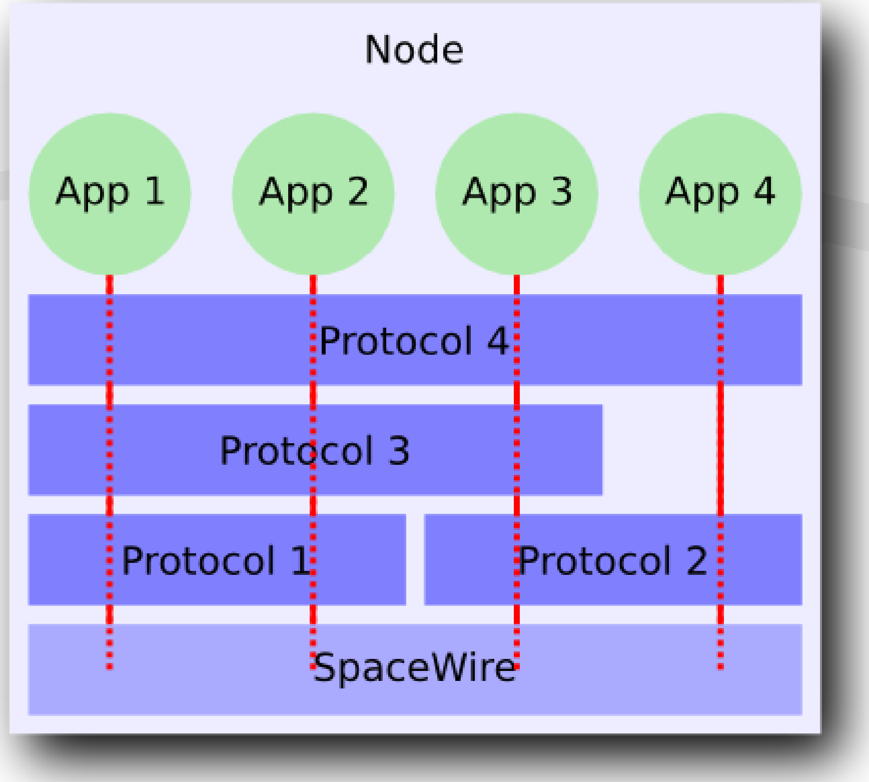
View of SpaceWire Network

- Consistent view of a SpaceWire network based on a protocol stack
- Scope is all SpaceWire protocols *not just SpaceWire*
- Network comprises **devices**
- Each device hosts **applications**
  - An application is the logical source/destination of packets from the perspective of the complete SpaceWire protocol stack
  - Each application uses one or more protocols (inc. SpaceWire)



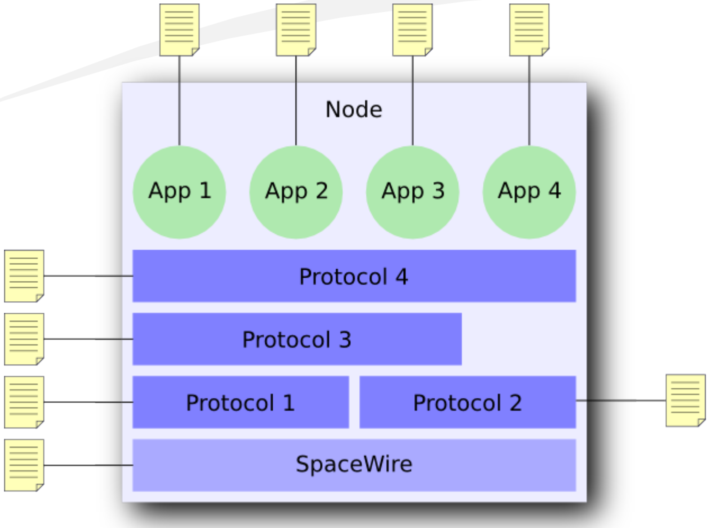
Protocol Stacks

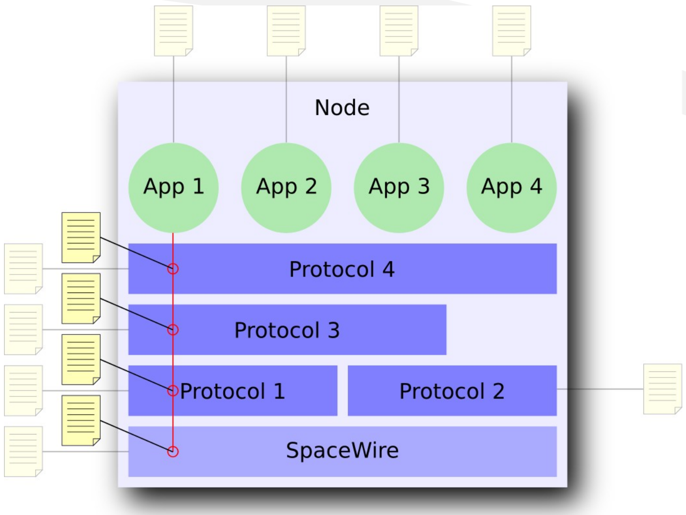
- An application uses a stack of protocols to communicate
- The route data takes through the stack is a channel



Network Management

- Each protocol and application have configuration or management parameters
  - Also the use of a protocol by a application has configuration parameters





Principles

- Using this view the detection and configuration of
  - Devices
  - Protocols
  - Applications
- …can be done in a uniform manner
- Once we can access the various configuration parameters we can
  - Perform discovery
  - Carry out complete SpaceWire network management
- These activities are independent of the underlying protocol used to access the device
  - Providing parameters are exposed through a defined service interface

Network Roles

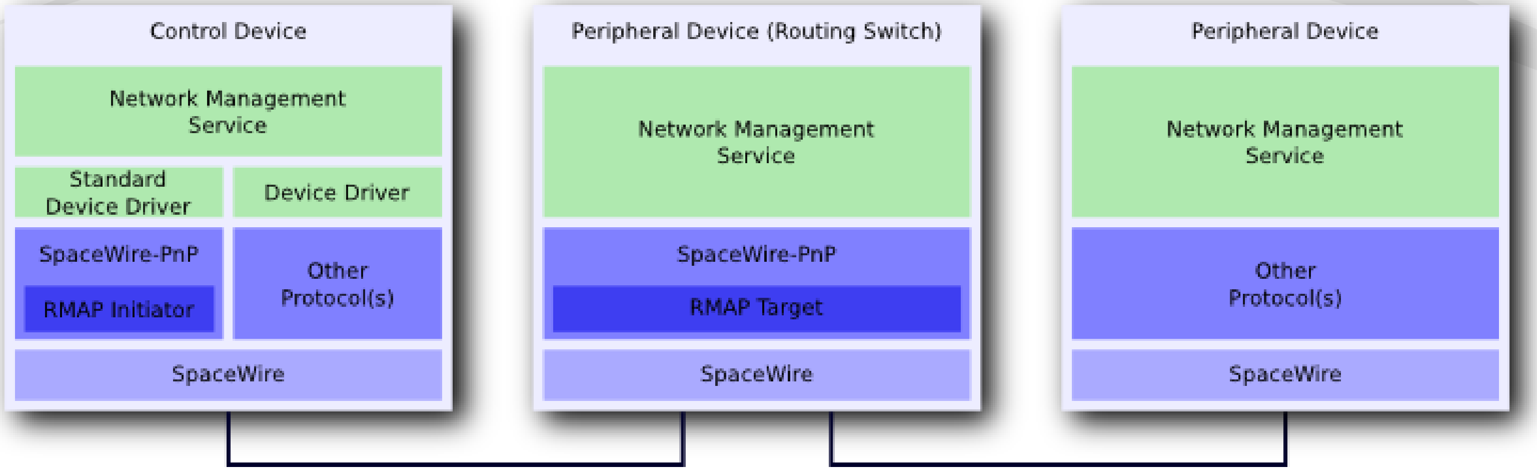
- Devices on the network are referred to as
  - **Control** devices
  - **Peripheral** devices
- A control device is also typically a peripheral device
- Nodes and routing switches are devices
  - The routing switch configuration endpoint is part of the device
- Devices can be grouped together arbitrarily as **units**
  - Independent of topology
- Two **assumptions** (restrictions) on nodes
  - Up to 31 links
  - Links must be equivalent

Network Management and Access Protocol

- Separate network management from the underlying communication protocol
- Provide support for many devices by permitting multiple protocols
  - Exposed using device drivers
  - Permits uniform support for existing devices
- Provide a standard protocol
  - Permits true interoperability
  - Necessary for truly open networks
  - Reduces implementation/validation complexities
  - Device driver for standard protocol is effectively null
  - Standard protocol based on RMAP

NMS and Device Drivers

- Example shows single control device with two peripheral devices
- One peripheral device uses the standard protocol
- The other uses a non-standard protocol
- Both supported by the network management service



NMS as an Application

- The network management service is itself an application
- It relies on a communication protocol
  - And SpaceWire, of course
- Quality of service can easily be added by adding protocols to the stack
  - Determinism
  - Reliability
  - Segmentation
  - Etc.
- Without modifying the network discovery service

Control and Peripheral Devices

- Design exploration indicated that complexity is concentrated in the NMS on the control device
- Previous protocol work focussed on making peripheral devices as simple as possible
- This increased the complexity of the control device
- This also necessitated a high level of standardisation in control device behaviour
- Take a different approach
  - Move some of the protection mechanisms to the peripheral device from the control device
  - Protection mechanism is now enabled by peripheral device functions
  - Not reliant on control device behaviour or specific algorithms
  - Removes the need for complete standardisation of control devices

Standardisation Approach

- Standardise the **minimum** amount to ensure that
  - Requirements are met
  - Interoperability will be guaranteed
  - Immediate needs of the community are met
- Current proposal standardises
  - A communication protocol
  - The network management service on a peripheral device
    - This is effectively the “configuration space”
- No standardisation of network management service on control device necessary
  - Reduces standardisation and validation effort
  - More useful for community

Current Draft Standard

- Includes overall architecture and rationale
- Simple protocol based on RMAP
- Protocol references RMAP but does not repeat it
  - This was not trivial to achieve due to structure of RMAP
- Protocol utilises the RMAP protocol “internally”
  - **Not** “layered” on top of RMAP from a logical perspective
  - May be layered in an implementation (not in scope)
  - Is layered from a standardisation perspective
- Peripheral device network management service
  - Meets all requirements
  - Is well defined, flexible and extensible (limited options)
  - Guarantees a minimum level of interoperability

Communication Protocol

- Operates on regular-sized **fields**
  - Each field is 32-bit
- Fields are grouped into **field sets**
  - Each field set contains 16,384 fields
- A block of 32 field sets is assigned to each management parameter set
  - i.e. a block for each protocol, application and application-protocol use
- A field is therefore accessed by specifying
  - <Application Index>, <Protocol Index>, <Field Set ID>, <Field ID>
- Available operations
  - Read
  - Write
  - Compare and swap (CAS)

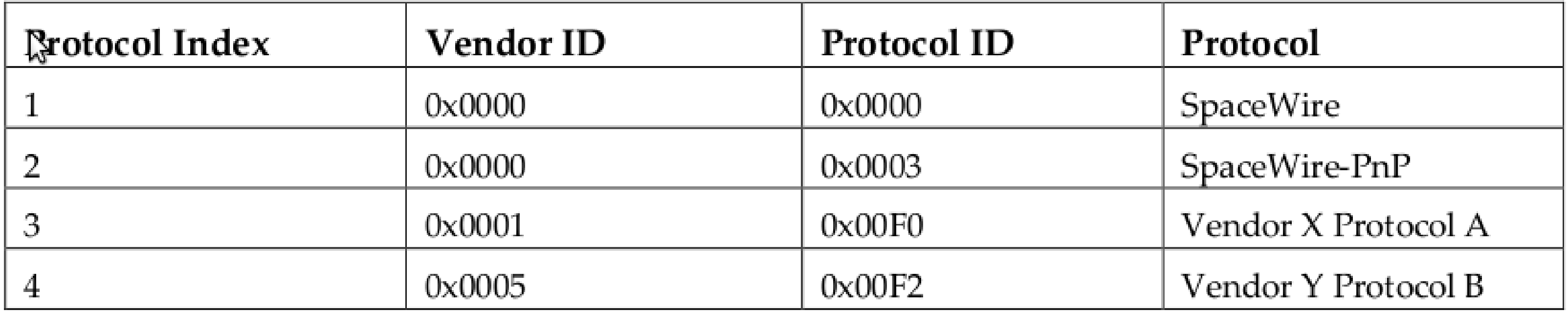
NMS: Device Information

- Specifying Protocol Index 0 and Application Index 0 provides access to root device information
- Vendor and product ID and identification strings
- Version
- List of protocols
- List of applications
- List of which protocol each application uses
- Status
- Network level view:
  - Available links and whether they are active
- Unit ID
- Device ID and owner

Protocol and Application Tables

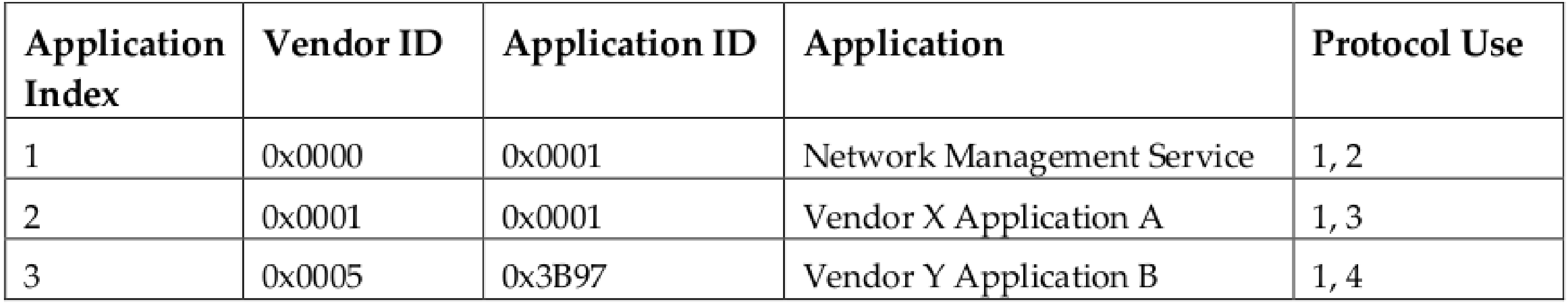
- Device can provide lists of supported protocols and applications
- Each identifier by a vendors ID (0=standard ECSS) and a protocol/application ID

Table 4-1: Example protocol support list



Protocol Index	Vendor ID	Protocol ID	Protocol
1	0x0000	0x0000	SpaceWire
2	0x0000	0x0003	SpaceWire-PnP
3	0x0001	0x00F0	Vendor X Protocol A
4	0x0005	0x00F2	Vendor Y Protocol B

Table 4-2: Example application support list



Application Index	Vendor ID	Application ID	Application	Protocol Use
1	0x0000	0x0001	Network Management Service	1, 2
2	0x0001	0x0001	Vendor X Application A	1, 3
3	0x0005	0x3B97	Vendor Y Application B	1, 4

Device ID and Ownership

- Device IDs are assigned by control devices
- The control device assigning the ID is the owner
- Device ID may **only** be assigned using CAS
- When the Device ID is 0 (as on reset) all other fields are read-only
- When Device ID is assigned, the device records the reply address from the communications (RMAP) packet
- This is the owner address
- Once a Device ID is assigned other fields may be written to
  - Providing that the reply address of the request matches the owner address
- Owner address can be read so that the current device owner can be located
  - Determine validity of current owner

SpaceWire Protocol

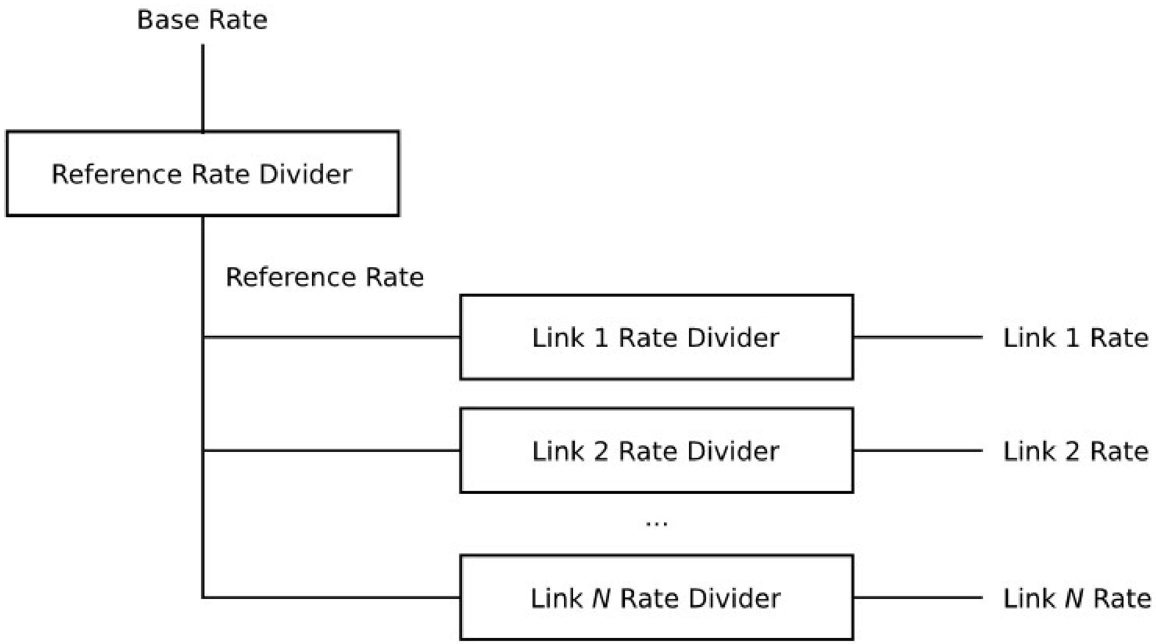
- Accessed using
  - Protocol index as specified in protocol support table
  - Application index 0
- Four field sets
  - Device configuration
  - Link configuration
  - Routing table (routing switches only?)
  - Time-code generation
- Fields are mandatory – functions are optional
- If representation does not match your existing implementation don't use it
  - Plenty of space for vendor-specifics

SpaceWire Management

- Time-code counting and propagation
- Link state, transmit and watchdog rates
- Link errors
- Debug information (FIFO states)
- Routing table (all addresses: 1-255)
  - Port association
  - Address control
- Fields are not always self-describing
  - e.g. link watchdog and transmit rates

Rate Scheme

- Link transmit and watchdog rates specified using a two-tier scheme
- Will not match all implementations
  - In which case don't use it
- *Does* match the majority of current applications



SpW-PnP Protocol and NMS

- SpaceWire-PnP Protocol
  - Identified using application index 0 and protocol index as per table
  - Protocol information specifies maximum supported read and write lengths
- NMS Application
  - Identified using protocol index 0 and application index as per table
  - Just reports status of NMS application

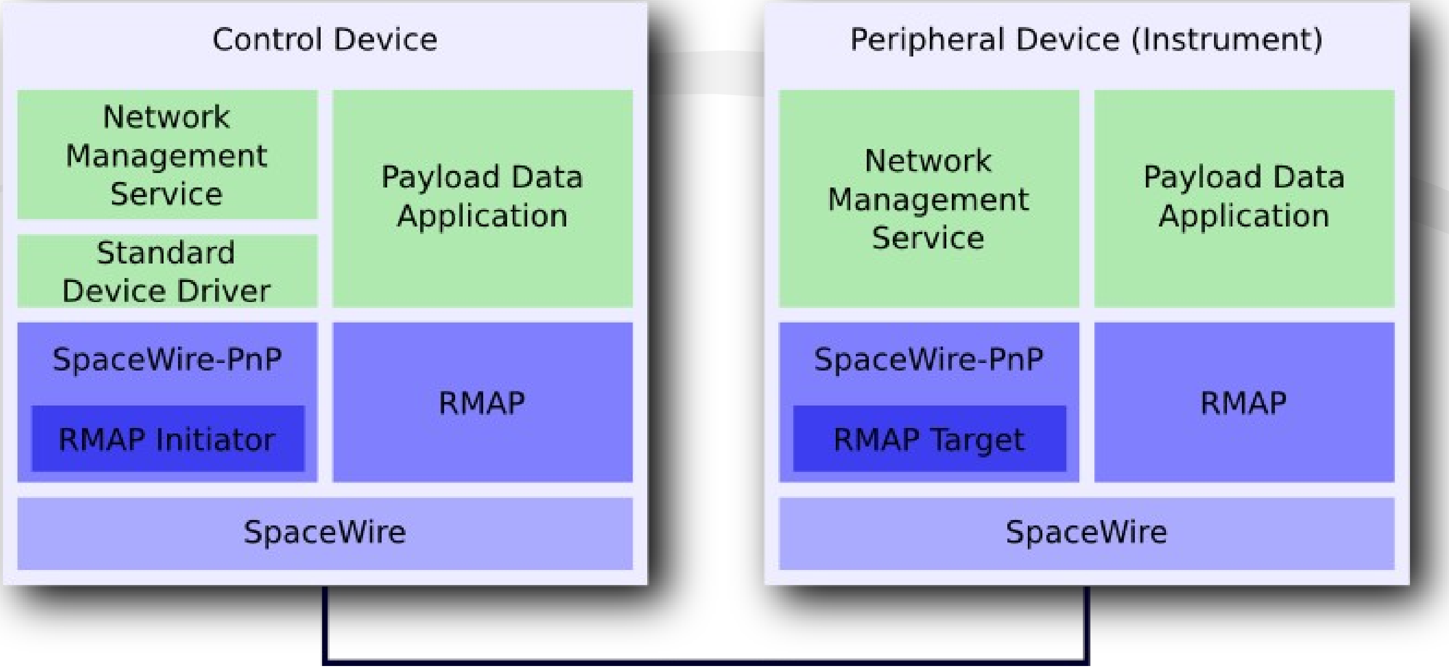
A Simple Example

An Example Device

- A hypothetical instrument
- A node
- Peripheral device only
- Two links
- RMAP interface to instrument data
- Instrument has various management parameters
  - Control parameters
  - Monitoring parameters

Example Device Protocols and Applications

- Instrument has two applications
- Uses two protocols
- Each application uses a single protocol in addition to SpaceWire



Minimal Implementation

- Communications protocol
    - Read and CAS only
- Network Management Service
    - Device information only
        - 2 constant read-only fields
        - 4 non-constant read-only fields
        - Device ID (modifiable using CAS only)
    - Unused fields must read as zero

Fuller Implementation

- Provide capability to manage
  - SpW-PnP
  - RMAP
  - NMS
  - Payload data application
  - The use of RMAP by the payload data application
- Payload data provided over RMAP
  - Target or initiator – you chose
- Payload monitoring and control using vendor-specific management space accessed using SpW-PnP

Conclusions

Summary

- SpW-PnP is a Network Management protocol
- Based on a clear concept of applications and protocols
- Network discovery is one application of SpW-PnP
- The current draft standard has evolved from previous proposed protocol
  - Re-assessed from first principles
  - Simplified
  - Reduced standardisation burden
- Split communications protocol from network management
- Extensible architecture which supports device drivers for non-standard devices

Applications

- Network discovery
  - Unknown or dynamic networks
  - Valuable for confirmation of known networks
- Management of SpaceWire protocol
- Management of vendor-specific
  - Applications
  - Protocols
- Exposure of management and monitoring parameter
  - FDIR

Conclusions

- SpW-PnP offers a network management solution for the complete SpaceWire stack
- Clear split between protocol and application
- Simple mandatory implementation
- Minimised standardisation
  - Achieves good interoperability
  - Low validation effort
- Builds on RMAP standard
  - Explicitly permits reuse of existing RMAP IP

Next Steps

- Change name – SpW-NM?
- Testing and demonstration (NDP)
- Feedback from working group
- Early implementations (e.g. A-G Router)
- Define management parameters
  - SpaceWire Protocol (including FDIR parameters?)
  - NMS
  - SpW-PnP
  - RMAP?
  - CPTP?
  - SpaceWire-D?
  - SpaceFibre?
- SpW-PnP should define management parameters for SpW-PnP, NMS and SpW
  - Others should be in their respective standards
- Process for Vendor (and Protocol) ID assignment

Feedback

- The draft standard is published as supporting material for this WG
- It will not be changing in the near future
- Feedback is very welcome
- The more the better!
  - peter@brightascension.com

Backup Slides

SpW Taxonomy

- The presented taxonomy makes it difficult to discuss a SpW network considering
  - The bigger picture (higher layers)
  - Network management
- From a SpW perspective the “higher layer” is always shared for a device
  - Node or router
- There are three levels at which the network makes sense
  - Devices (nodes/routing switches)
  - Endpoints (the boundary of SpW itself)
  - Applications (the logical source/destination of SpW protocol communications)

Notification

- The current standard does not include notification
- Design work did consider notification
- Proposal is a simple publish-subscribe model
- Underlying protocol provides
  - Subscribe
  - Unsubscribe
  - Publish
- You can then subscribe to a field just like reading or writing
  - Only a few fields would support subscription
- No QoS (e.g. retries) in protocol
  - To be added by additional layers (.e.g SpW-R)
- Still built on RMAP