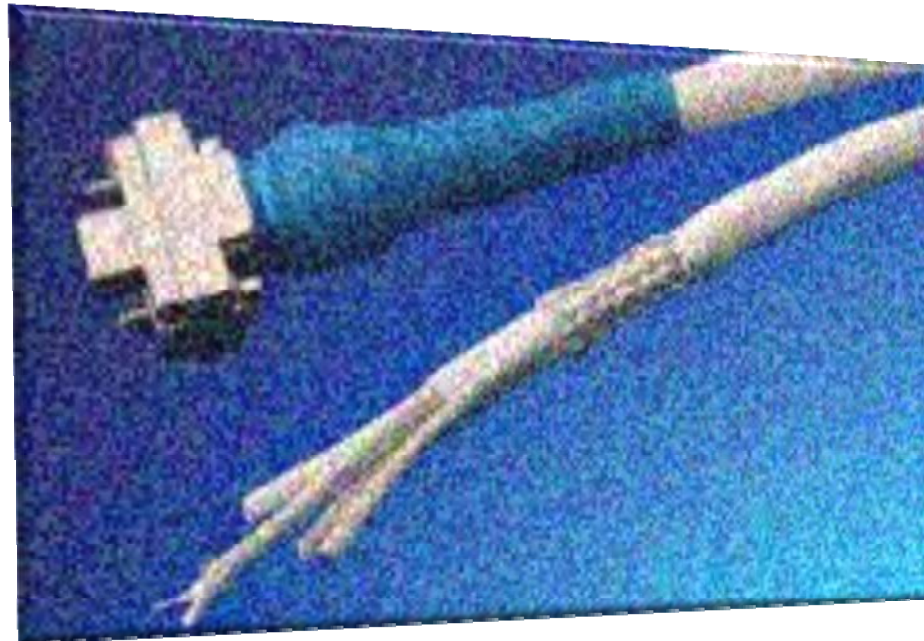


SpaceWire Evolutions



Accommodating Plug-and-Play in SpaceWire: Progress Report

*17th SpaceWire Working Group, 14th December 2011
Noordwijk, Netherlands*

Current Status:

- There is significant interest in generating one or more 'plug-and-play' standards for SpaceWire
 - There have been a number of proposals and the general techniques involved are well understood
 - Issues have been identified with the compatibility of these solutions with the current standard

The Problem:

- The adoption of one or more plug-and-play standards is hampered by the current definitions of terms
 - The definitions in the SpaceWire standard do not differentiate between the physical and logical/functional network
 - The definitions need to be clearer whilst respecting their current use and the way that similar terms are used to describe other communications technologies

The Proposed Solution:

- The modification of term definitions together with the introduction of a few new concepts
 - Differentiation between the physical and logical aspects of a SpaceWire network
 - Accommodation for the most critical aspects of plug-and-play standards without the imposition of any particular standard

Understanding the Problem (1/2)

- It should be possible to define a plug-and-play standard
 - Separate to SpaceWire (ECSS-E-ST-50-12C)
 - Compatible with SpaceWire (ECSS-E-ST-50-12C)
- Most proposed 'plug-and-play' techniques for SpaceWire make some use of a 'configuration port'
 - For both *nodes* and *routers*
- How a configuration port in a node is accommodated by the standard is not clear
 - The behaviour of configuration packets addressed to the configuration port of a SpaceWire-compliant node that does not implement a plug-and-play protocol is **undefined** and could be dangerous
- Before node configuration ports can be described in the standard, nodes and configuration ports must be **clearly** and **unambiguously** defined
- We mostly talk about SpaceWire networks as **physical** things but plug-and-play discovers the **logical** network
- Physical and logical networks might not be the same
 - e.g. a physical unit with multiple SpaceWire devices inside (such as multiple routing switches)

Understanding the Problem (2/2)

Node	source or destination of a packet, which can be a processor, memory unit, sensor, EGSE or some other unit connected to a SpaceWire network
Router	routing switch
Routing Switch	switch connecting several links that routes packets from one link to another where the destination address of each packet by the switch is used to determine which link a packet is sent out on

A node is a unit?

Physical?

Unit	box, board or subsystem, that can have one or more SpaceWire interfaces
Link	bidirectional connection of one unit to another unit for passing data and control information

Physical? Or Logical?

Link Interface	SpaceWire interface comprising a transmitter which takes data from a host system and transmits it across a SpaceWire link, and a receiver which accepts data from a SpaceWire link and passes it to the host system
-----------------------	---

Defines a link interface by its contents and suggests that a link interface always connects a link and a host system

What entities are discovered (and managed) by a plug-and-play protocol?

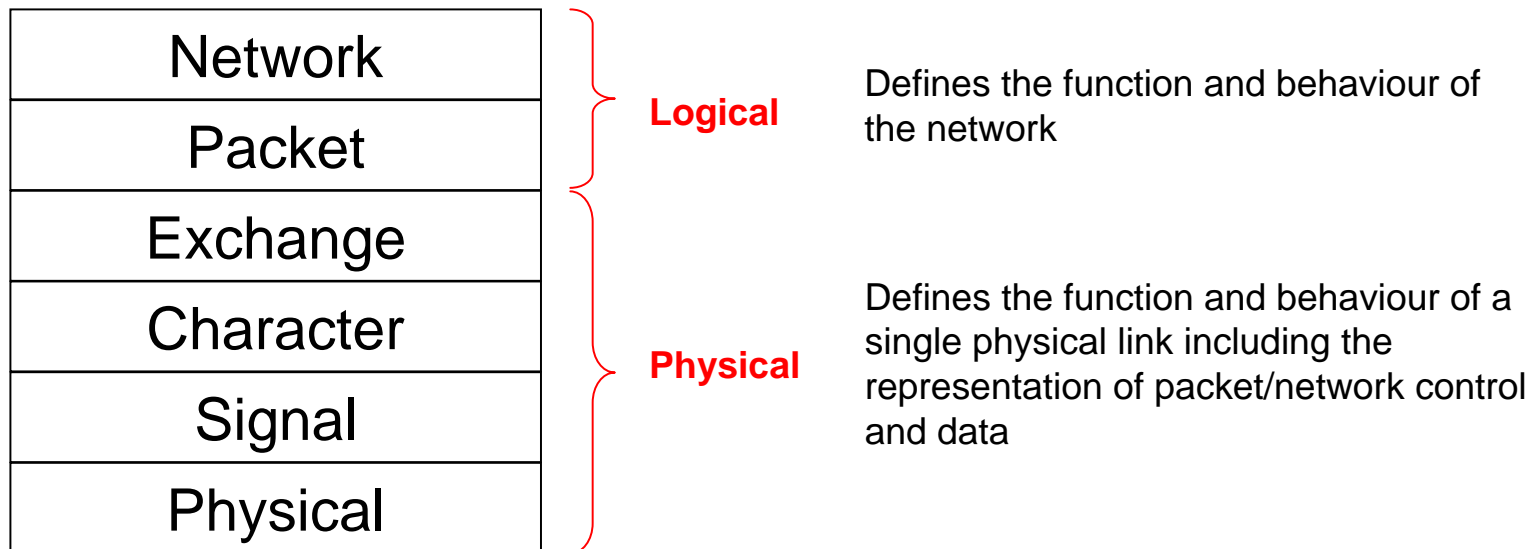
Requirements

- Update the terms in the SpaceWire standard, but only where necessary
- Terms must be self-consistent in content and concept
- Must be consistent with:
 - both physical and logical networks
 - other parts of SpaceWire evolutions (sideband signalling, low mass communications)
 - existing uses of terms by other major communications standards
 - existing uses of terms within the SpaceWire community (where possible)
- Propose a mechanism for accessing device identification and configuration information
 - Associate device identification and configuration information with a logical device (rather than physical)
- Be independent of network topology
- Be consistent with device discovery techniques currently under discussion e.g. network discovery ('walking' the network), router-based notification etc.

Physical and Logical Networks

- What is the physical network?
- What is the logical network?

SpaceWire Standard Levels

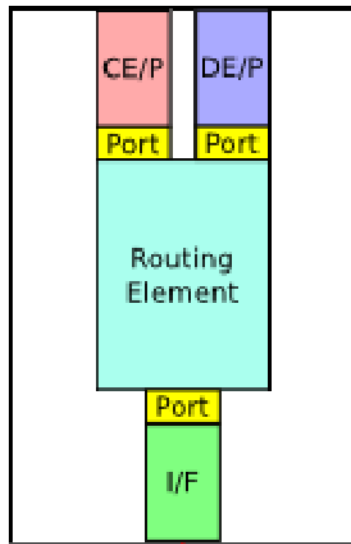


Concepts and Terms

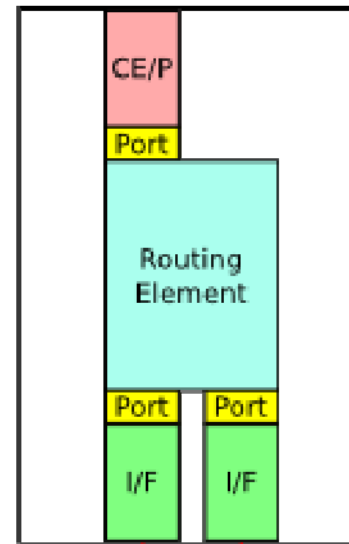
- We need to introduce a 'configuration port' to provide access to discovery and configuration information
- Nodes and routers need to have a configuration port
- This means that there are two logical packet sources/sinks in a node
- Define the source/destination of packets to be an **endpoint**
- Two types of endpoint:
 - **Configuration endpoint**: one (and only one) in every logical node or router
 - **Data endpoint**: one or more in every logical node, none in a router
- When packets arrive a node they need to be separated into configuration or data packets
 - Needs something which performs a routing function: call this a **routing element**
 - The function performed by a routing element might be very simple
- The routing element sends packets to a **port** which is
 - connected to either a link interface or an endpoint
 - associated with one or more destination identifiers

Nodes and Routers

- Simplest logical (a) node and (b) router



(a)



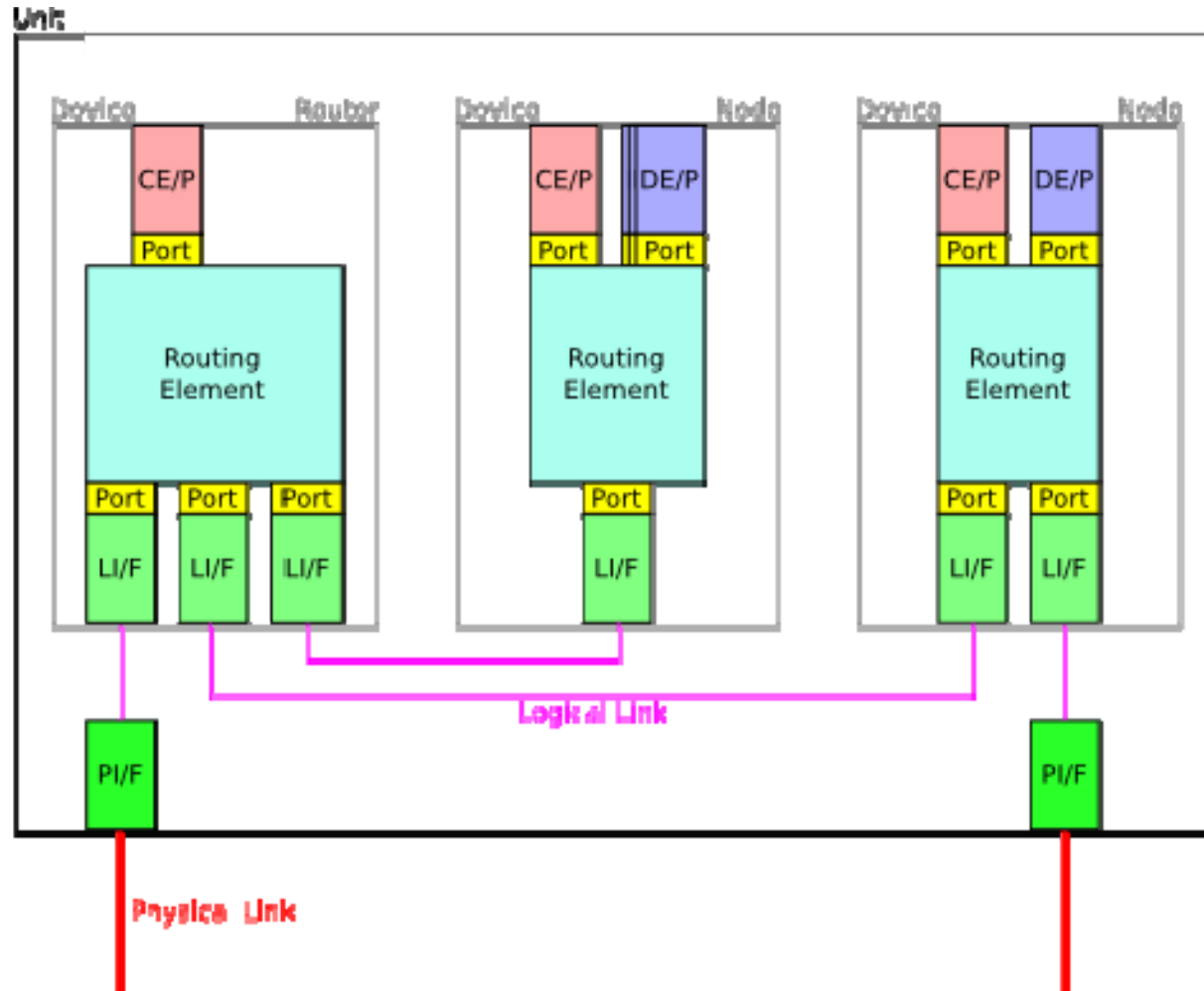
(b)

CE/P	Configuration Endpoint
E/P	Data Endpoint
I/F	Interface
RE	Routing Element

Logical and Physical Networks Revisited

- An entity with SpaceWire **physical interfaces** (physical, electrical and character levels) is a **unit**
- An entity with SpaceWire **logical interfaces** (exchange, packet, network levels) is a **device**
- There may be a one to one mapping between the unit and device
- A node and a routing switch are types of device
 - Obviously it is also possible to have, for example, a routing switch unit – this is a unit which contains a single routing switch device
- Every device has one and only one configuration endpoint
- Every node has one or more data endpoints
- Every routing switch has no data endpoints (by definition)
- A routing switch must perform a routing function
- A node *may* perform a routing function

Physical Unit with Multiple Devices



CE/P	Configuration Endpoint
DE/P	Data Endpoint
LI/F	Logical Interface
PI/F	Physical Interface
RE	Routing Element

Compatibility

- Tried to be compatible with the way that we use terms such as router and node
- Terms should not break any current uses of the SpaceWire standard
- Introduction of a mandatory configuration endpoint adds extra constraints to implementations
 - Minimal implementation is that configuration and data packets must be distinguished
 - Configuration packets should be spilt, no response necessary
- How to distinguish configuration and data packets?
 - Restrict a standard data cargo to begin with a non-null character.
 - *Easy to define but is overly restrictive and breaks the protocol ID standard (extended PIDs)*
 - Define that a **configuration packet** arriving at a destination device must begin with a null character and that a **data packet** arriving at a destination node must begin with a non-null character
 - *Works with protocol ID standard as the first character of a data packet arriving at a destination node should be a logical address which may not be zero.*
- There are issues for backwards compatibility here – this is necessary to move the standard forwards

Solution Summary

- Refine some basic terms in the SpaceWire standard to add clarity
 - e.g. node, routing switch, unit, link, link interface
- Define some new terms to ensure a consistent conceptual framework
 - e.g. device, endpoint, port, routing element
- Introduce a mandatory configuration endpoint for all devices
 - Can just spill packets
 - Operation of configuration endpoint is defined by a plug-and-play standard

- Permits use of plug-and-play protocols with SpaceWire in a standard and well-defined way
- No additional restrictions on plug-and-play protocol definition or implementation
- Limits issues with backwards incompatibility
 - The way that we use terms within the SpaceWire community
 - The way that we construct SpaceWire devices

Backup Slides

Layering

- Layering in SpaceWire is ill-defined as “levels” are not proper layers
- Making some of the levels better defined to permit layering will help more than just plug-and-play
 - Also essential for SpaceFibre definition
- Layering is not clean at exchange/packet levels