SpaceWire Network Definitions and Concepts

Peter Mendham

18th SpaceWire Working Group, 24th April 2012
Noordwijk, Netherlands

ESA Contract Number 4000104023, SpaceWire Evolutions

Overview

- Review of starting point and requirements
- Key outcomes from analysis
  - Separation of physical and logical/discoverable networks
  - Separation of link and network standardisation
- Accommodating network management (PnP)
- Simplified decision tree
  - Should network management be included in the SpaceWire standard?
  - Should nodes be permitted to include a routing function?
- Proposed solution

Starting Point

- Accommodating plug-and-play in the SpaceWire standard
- Plug-and-play discovers a network...
  - What entities does it discover?
  - How do these entities relate to the physical network?
  - How do these entities relate to the way the network behaves and is used?
- We need to have
  - Clear definitions of the entities that make up the SpaceWire network that is discovered
  - A clear a consistent concept of SpaceWire networks

- The network that is discovered is based on the way a network behaves and is used
  - This is the *logical network*
- The network that we work with when plugging cables between units is the *physical network*
  - These networks may or may not be different

Requirements

- SpaceWire Evolutions activity captured requirements
- It should be possible to define a plug-and-play (network management) standard
  - Separate to SpaceWire (ECSS-E-ST-50-12C)
  - Compatible with SpaceWire (ECSS-E-ST-50-12C)
- It should be possible to define multiple network management standards, if required
- There must be a uniform, standard mechanism to
  - Expose the identitfy of a device to permit discovery
  - Provide access to configuration and management parameters

- The current standard defines such a mechanism for routing switches only
  - The configuration port

- We need to understand
  - What entities make up the physical and logical networks
  - How a network management mechanism can be associated with each logical entity

The SpaceWire Evolutions Activity

- Intention was to produce a proposal for how to accommodate plug-and-play in ECSS-E-ST-50-12C
- Requirements were captured
  - And reviewed
- A thorough analysis of the problem was conducted

- The analysis was
  - Abstract
  - Theoretical
  - Conceptual
- But tried to converge on a pracrtical solution

- “Practical” means
  - Relates to current understanding by the community
  - Relates to the way the community currently uses SpaceWire
  - Does not restrict the applications of SpaceWire, present or future
  - Is easily understood
  - Fits into the existing standardisation framework

Activity Outcomes

- A deep understanding of the theoretical framework of SpaceWire networks
  - Hopefully no one will need to worry about this, in this much detail, ever again!
- The identification of key issues related to the accommodation of plug-and-play
  - Physical/logical networks
  - Separation of link/network standardisation
- A set of potential proposed solutions
  - Each solution has many implications
  - One preferred option

Physical and Logical Networks

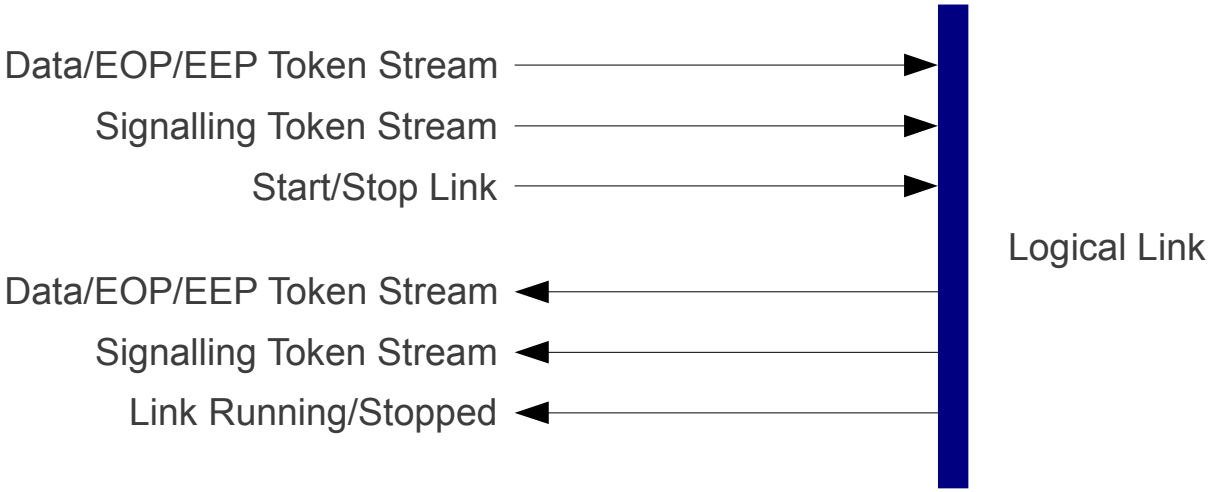
- When we discover or manage a network the entities we discover may be different than those physically apparent
- What we discover is the **logical network**
  - It consists of the logical entities that can be addressed, used and managed
  - The logical network is the level at which logical addresses are assigned
- What we physically construct is the **physical network**
  - An entity in the physical network may include one entity from the logical network...
  - ...or a portion of the logical network
- There is no single consistent definition for mapping these two networks
  - A single mapping is too restrictive
  - Must depend on application
- This is not a problem, just something to be aware of

- For the sake of describing these concepts in this presentation
  - A physical network consists of **units** connected by **physical links**
  - A logical network consists of **devices** connected by **logical links**
- These terms are not perfect but have been selected as a “least worst” option

Logical Links

- The physical implementation of a link in the logical network is not important
  - Only the behaviour is important
- A service interface to a logical link can be defined

- A SpaceWire link consists of two, independent, bi-directional channels
  - One carring a stream of data and EOP/EEP tokens
  - The other carrying a stream of signalling tokens
- “Signalling tokens” are time-codes and distributed interrupts
- Can also request that a links starts/stops and determine whether a link is running



Logical Link Implementations

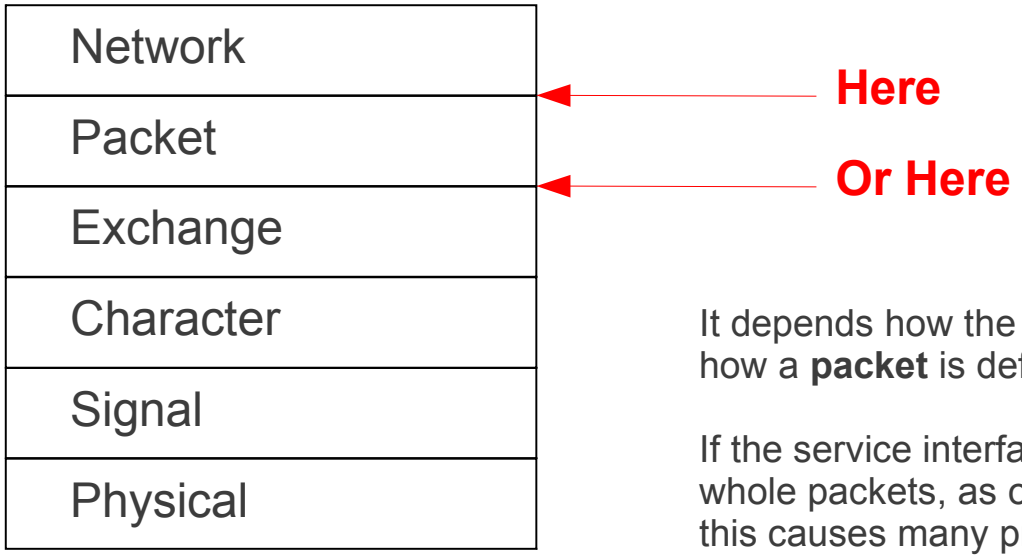
- There are multiple ways of impementing a logical link
- Two good examples in existence:
  - A standard LVDS, DS-encoded SpaceWire link
  - The external port(s) on the SpW-10X
    - 3.3V LVCMOS
    - 9-bit parallel bi-directional data/EOP/EEP token stream
    - 8-bit parallel bi-directional signalling token stream
    - As implemented, signalling token stream is actually shared between two links
    - Links always running (cannot start/stop so no connected status necessary)
- Many other examples exist
- SpaceFibre/SpaceWire-2?

- It is possible to implement only some levels of the SpaceWire standard
  - Providing that the service interface at the logical link-level is maintained
  - Can then choose to implement levels fromm the standard working downwards from the packet level

The SpaceWire Standard and Logical Links

- The structure of the SpaceWire standard should make a clear distinction between
  - The definition of the logical network, including logical links
  - The definition of a standard link implementation, split into levels with defined service interfaces
- This fits well with the planned updates to ECSS-E-ST-50-12C
- It's not clear where the logical link interface sits in relation to the current standard's levels

SpaceWire Standard Levels



It depends how the service interfaces are defined and how a **packet** is defined.

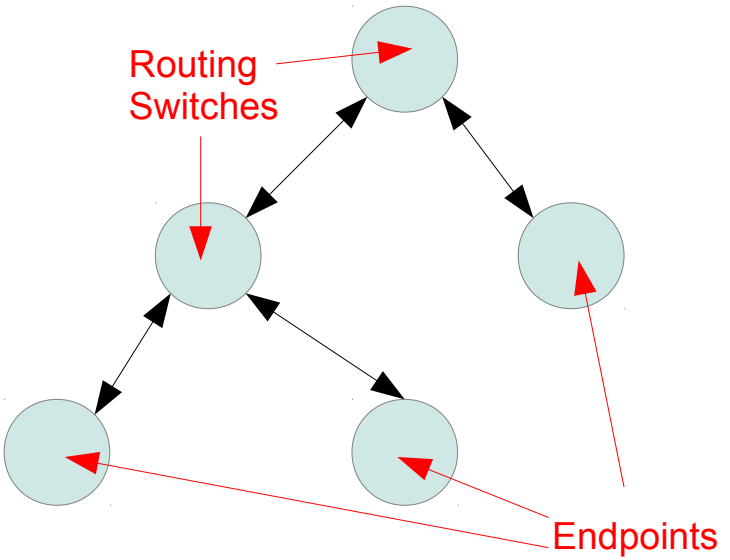
If the service interface to the packet level consists of whole packets, as opposed to a stream of tokens, then this causes many problems...

The Logical Network (1)

- So what are the entities that make up the logical network?
- We know that we need at least two
  - Packet sources/destinations, let's call these **endpoints**
  - Entities to direct the packets, these are **routing switches**
- These are connected by logical links

- As a graph
  - Endpoints are leaf vertices (and vice-versa)
  - Routing Switches are non-leaf vertices (and vice-versa)
  - Edges are logical links (and vice-versa)
- Graphs are either non-directed or directed and symmetric

- Assume that this graph represents the network in normal use
  - Data packet flow
  - Ignoring network management (for now)



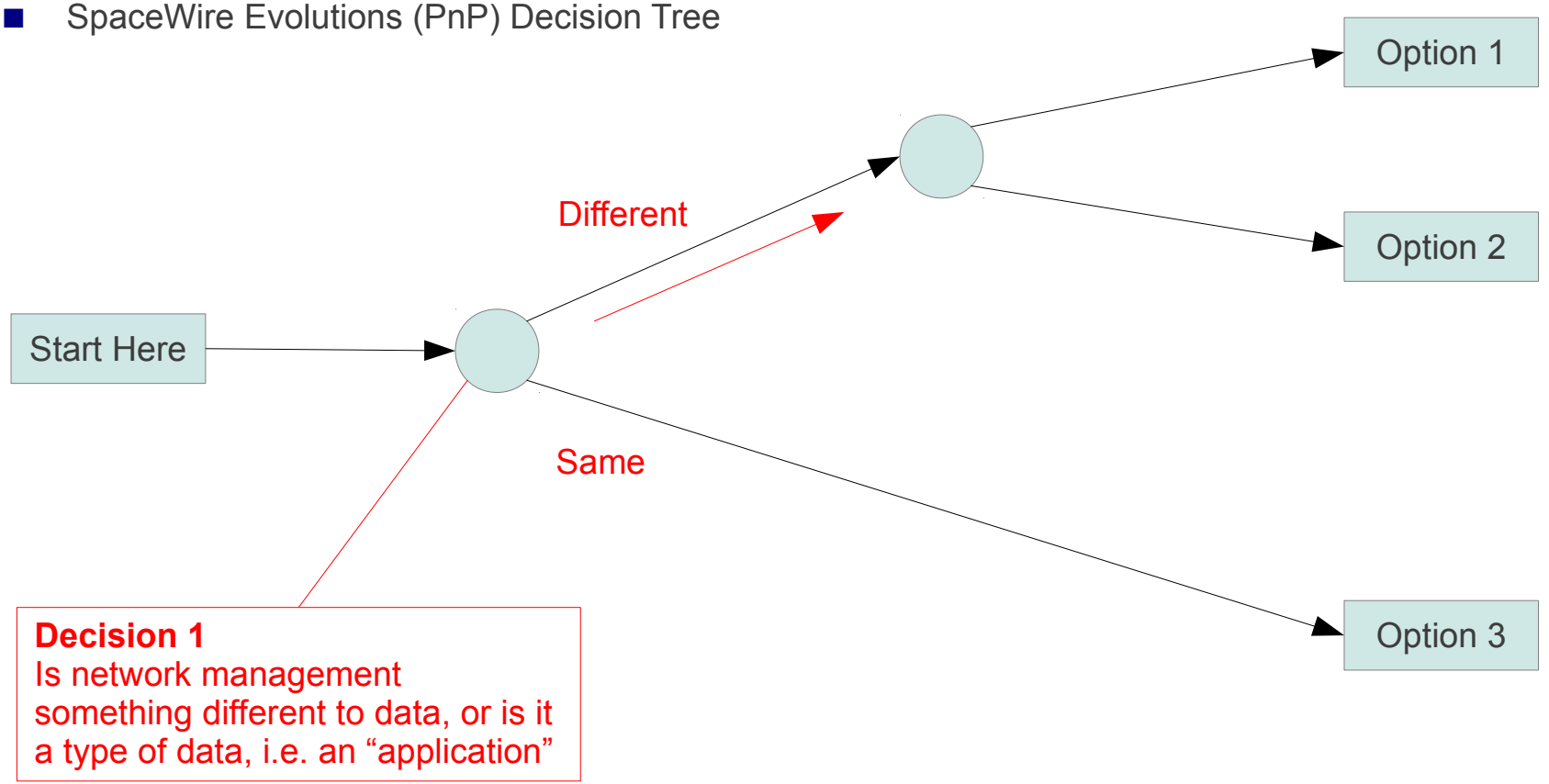
The Logical Network (2)

- How do endpoints and routing switches relate to
  - Node devices?
  - Router devices?
- We assume that a device is the unit of network management
  - i.e. it has a shared configuration space
- A node device must contain at least one endpoint
- A router device must contain a routing switch

- Key questions
  - How is network management represented by the graph?
  - Should we allow multiple endpoints in a node device?
  - Should we allow endpoints in a router device?
- And more

Choose Your Own Adventure (1)

- SpaceWire Evolutions (PnP) Decision Tree

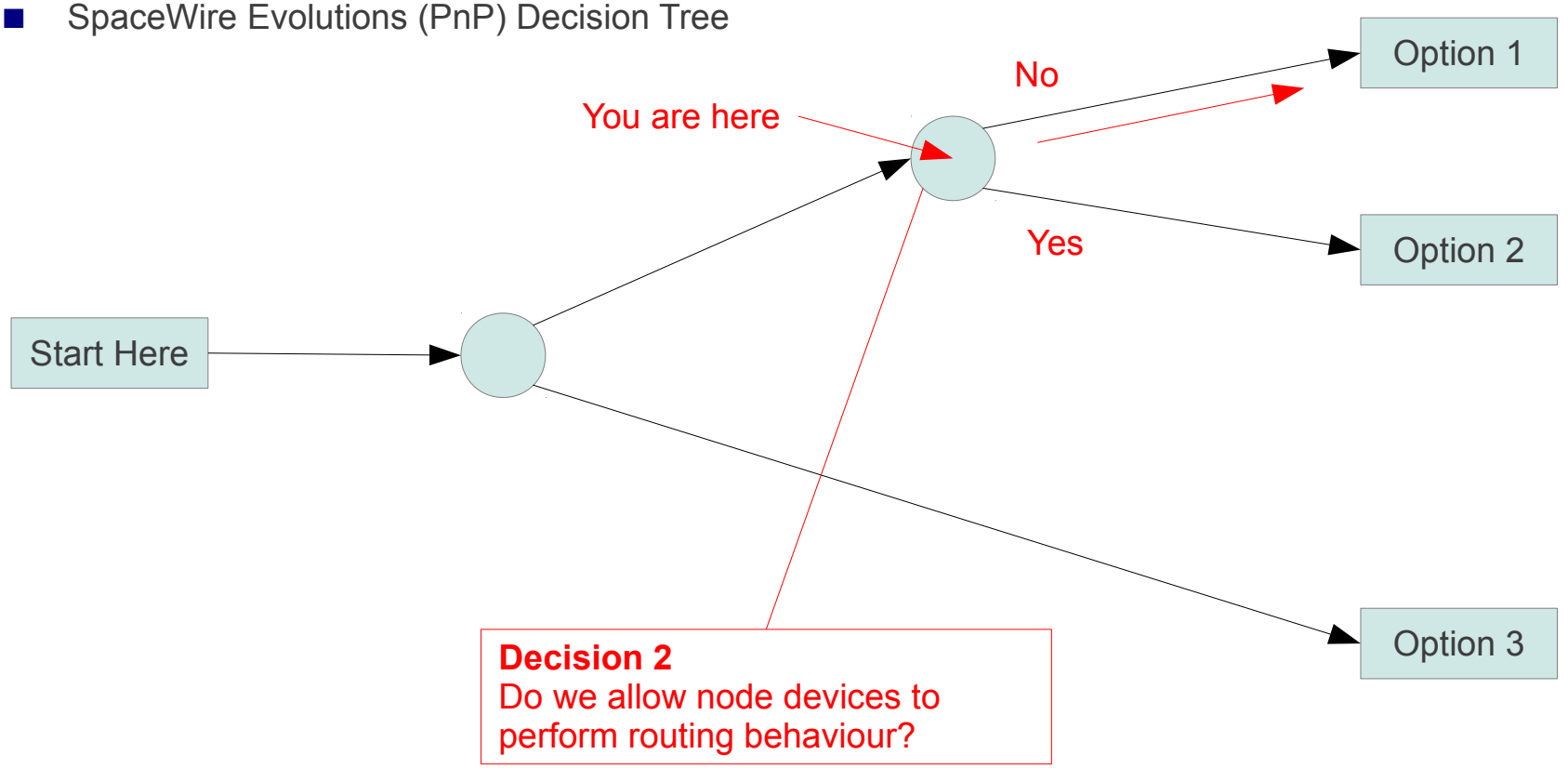


Network Management as First-Class Traffic

- We have decided to make network management “different” to data
- We can now have two different types of endpoint
  - Data endpoints
  - Network management (configuration) endpoints
- Following from this:
  - Each device has a single network management endpoint
  - A router device contains a routing switch and a configuration endpoint
  - A node device contains a configuration endpoint and one or more data endpoints
  - The graph now has three types of entity
    - Routing switches
    - Configuration endpoints
    - Routing switches
  - An endpoint is now something conceptual
    - Different from the host interfacein the current SpaceWire standard
    - Not necessarily associated with anything physical (e.g. a buffer or FIFO)
  - There must be something in each device which distinguishes between configuration and data packets and sends them to the appropriate endpoint

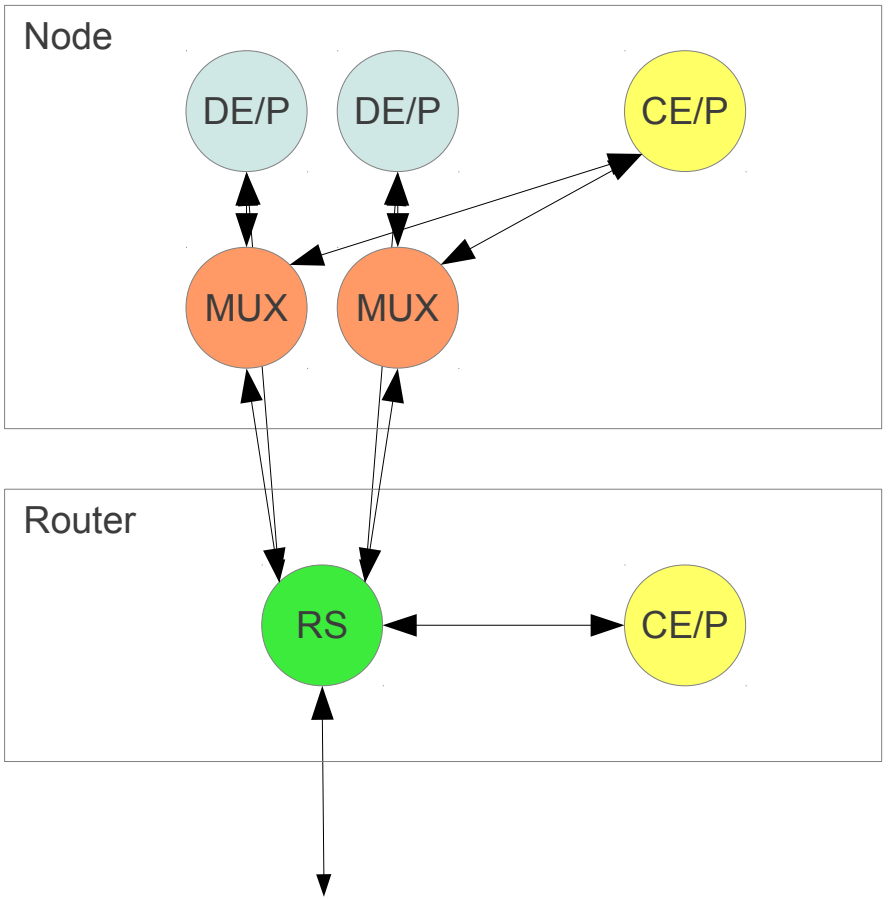
Choose Your Own Adventure (2)

- SpaceWire Evolutions (PnP) Decision Tree



Option 1: Nodes that do not Permit Routing

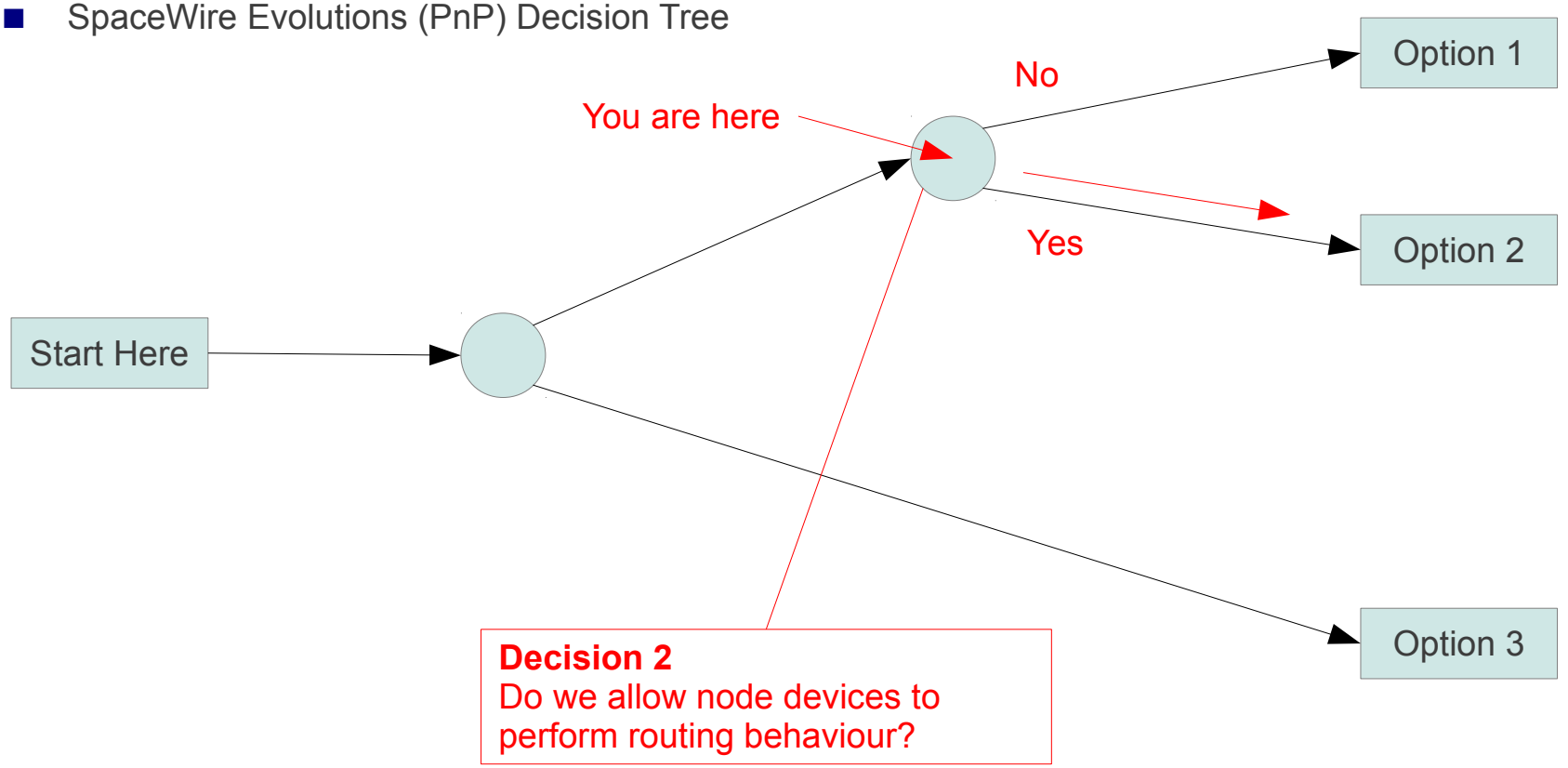
- We have chosen to prevent nodes from implementing routing behaviour
- We therefore have a clear distinction between router devices and node devices



- The graph now has four different vertices
  - Data endpoints
  - Network management endpoints
  - Routing switches
  - (De-)Multiplexers
- This is a fairly direct (literal) representation of a likely implementation

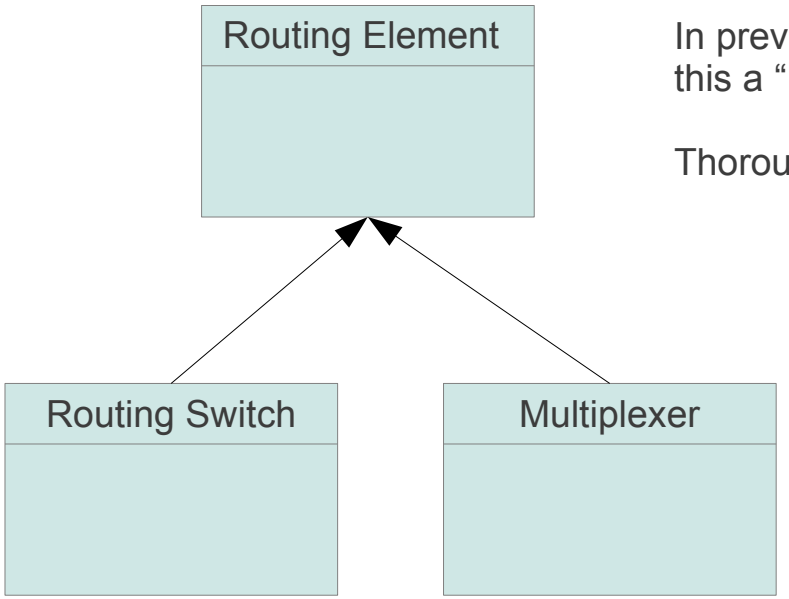
Choose Your Own Adventure (3)

- SpaceWire Evolutions (PnP) Decision Tree



Option 2: Nodes that Permit Routing (1)

- We have node devices which permit routing behaviour
- The packet selection in a node can now be
  - A (de-)mutliplexing-type behaviour (no routing)
  - Routing behaviour (full routing)
  - A bit of both... (limited routing)
- Define an entity which is more general than either a routing switch or a multiplexer
- In effect this is the mutual “superclass” or “parent class” of routing switches and multiplexers

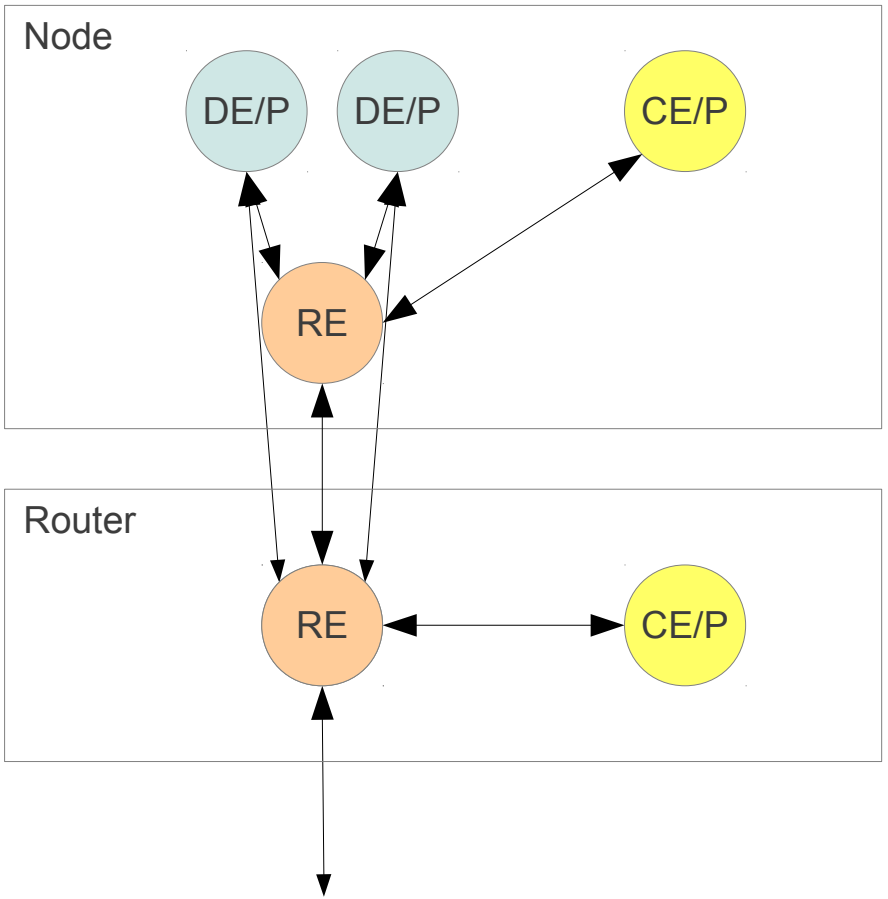


In previous presentations I called this a “Routing Element”

Thoroughly confusing, I'll admit...

Option 2: Nodes that Permit Routing (2)

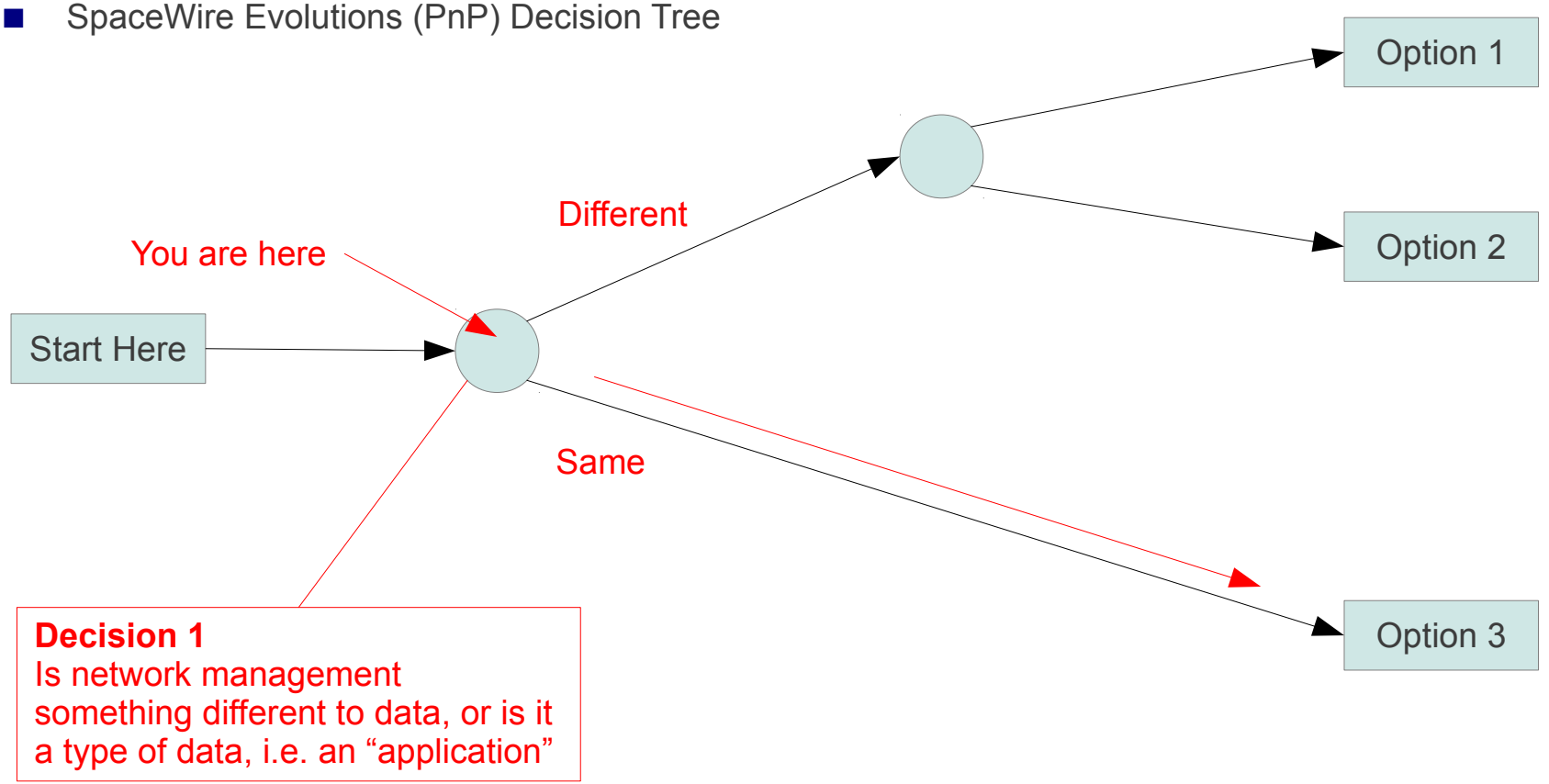
- The graph now has three different vertices
- Almost any network can be accommodated



- The logical network is now very conceptual
- The behaviour of a “Routing Element” is very abstract and potentially difficult to understand
- In order to cope with the behaviour of real devices further abstract concepts must be associated with the Routing Element
- More details on request...

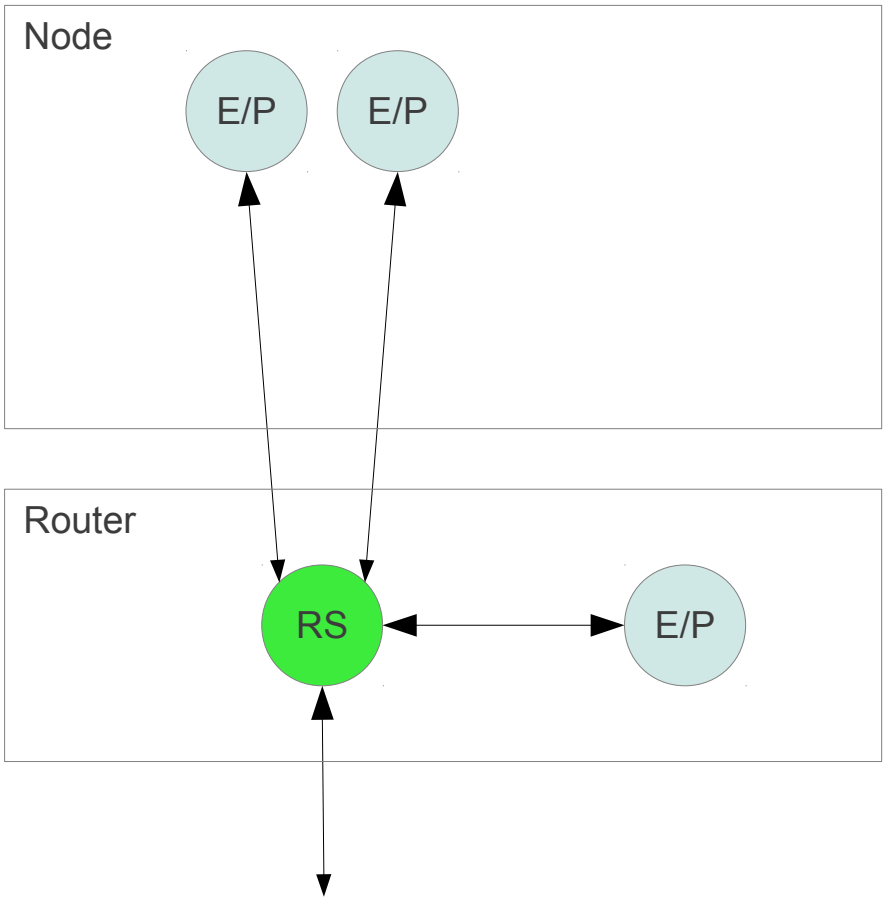
Choose Your Own Adventure (4)

- SpaceWire Evolutions (PnP) Decision Tree



Option 3: Network Management is Just Data (1)

- The contents of a packet is out of scope of the SpaceWire standard
- Mentions of “configuration” are removed or at least reduced from the standard



- The graph now has only two different vertices
  - Endpoints
  - Routing switches
- An endpoint corresponds closely to the host interface in the current SpaceWire standard
  - Also associated with a buffer or FIFO

Option 3: Network Management is Just Data (2)

- Distinguishing between network management and data packets must be done by a separate standard
- For consistency this must fit “in between” the SpaceWire standard and the Protocol ID standard
- The standard would be extremely simple
  - Identify packets based on the existence, or not, of a leading zero
- The behaviour of a routing switch as defined by the SpaceWire standard could be modified slightly
  - Port zero should be connected to a single internal endpoint
  - Header deletion should not be performed for address zero
- This presents a consistent packet format to the standard “above” SpaceWire
- No backwards compatability issues when new standard is issued

- Alternatively, a new “level” could be introduced within the SpaceWire standard
  - Above the network level
- There would be backwards compatability issues when the new standard is issued

Preferred Option

- The preferred option is **Option 3**

- Simplest
- No difficult concepts or behaviour
- Easy to relate concepts/definitions in standard to physical parts of implementation
- Very similar to current standard (limited changes)
- Backwards compatability issues can be prevented

- Nodes with routing behaviour are not permitted
  - To provide equivalent behaviour a unit must contain both a router device and a node device

Conclusions

- A thorough analysis of the problem of modelling SpaceWire has been conducted
- A set of recommendations has been made

- There should be a clear distinction between physical and logical networks
- Physical network entities (units) can contain an arbitrarily-sized logical subnetwork
- The standard should be split into links and networks with a clear service interface in between
- The logical network should make no distinction between data and configuration packets
- Distinguishing data/configuration should be handled by a separate standard

Your Opinion is Welcome – Please Discuss!