SPACEWIRE PLUG-AND-PLAY: FAULT-TOLERANT NETWORK MANAGEMENT FOR ARBITRARY NETWORK TOPOLOGIES.

Session: SpaceWire networks and protocols

Short Paper

Albert Ferrer Florit, Martin Suess

On-Board Payload Data Processing Section, ESA/ESTEC,

Noordwijk, The Netherlands

E-mail: albert.ferrer.florit@esa.int, martin.suess@esa.int

ABSTRACT

The SpaceWire Plug-and-Play protocol (SpW PnP) aims to provide a set of common features for SpaceWire devices to facilitate recognition and configuration of SpaceWire networks.

This paper presents a methodology for network discovery and configuration compatible with the SpW PnP protocol defined by the SpW PnP Working Group. It supports arbitrary network topology changes and provides fault tolerance features without requiring any manual configuration. Nodes and routers with identical hardware can be uniquely identified, and polling or active notification methods are used to register a new device in the network.

The proposed approach relies on the use of one or more intelligent nodes, called Network Node Managers, with the capability to independently configure any SpW PnP compliant device. Following suitable mechanisms to avoid race conditions, a network fully configured is only supervised by one Network Node Manager, with the others acting as hot backups.

Our methodology has been successfully prototyped with ordinary computers acting as network node managers, using RMAP protocol to emulate SpW PnP. Results proved that its use could be especially helpful for fast prototyping in the laboratory or in space manned missions.

1. INTRODUCTION

The SpaceWire Plug-and-Play protocol (SpW PnP) is a joint effort together with NASA and other partners in the frame of the SpW PnP Working Group [1]. It will enable the implementation of different network discovery and configuration methodologies depending upon a specific network management philosophy. The SOIS Plug-and-Play architecture is expected to be based on this protocol [2].

This paper presents a methodology for network discovery and network configuration without considering other typical Plug-and-Play services such as device drivers installation. The network management philosophy is based on the following assumptions:

A) SpaceWire network:

- A.1 The network topology is unknown and arbitrary.

- A.2 Arbitrary network topology changes can occur at any time due to failures or user intervention.

- A.3 Devices or subnets can be plugged and unplugged to/from any element of an existing network at any time. New devices plugged may not be in reset status.

- A.4 Multiple devices with the same hardware configuration may be present in the same network.

B) Network Discovery:

- B.1 It shall be executed by an intelligent node, called Network Node Manager.

- B.2 It shall detect plug/unplug events of any SpaceWire link, device or subnet.

- B.3 It shall uniquely identify all devices in the network.

- B.4 It shall support redundancy by using multiple Network Node Managers. Implementation shall avoid race conditions.

2. PROPOSED APPROACH

The proposed approach successfully deals with the previous assumptions and it is based on the use of intelligent nodes, called Network Node Managers (NNM).

Basic network discovery algorithm

A NNM interrogates routers about the status of their ports or links in order to discover new devices (nodes or other routers). Path addressing and configuration port is used for this purpose. Every device is configured with a different device identifier in order to avoid identifying multiple times the same device when there are loops in the network. This process ends when all devices in the network has been identified and programmed with a unique identifier.

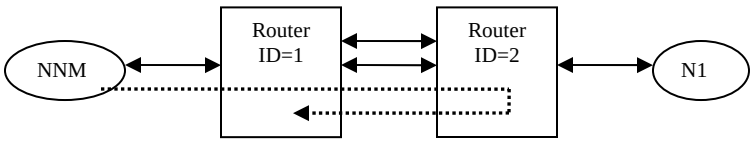

Figure 1: Simple network example. Dashed line shows how a NNM detects a loop.

Multiple Network Node Managers

Fault tolerance capability is implemented by using multiple NNMs. However, only one NNM, called Master NNM, should be active when the network is fully operative. This ensures that all devices have a unique identifier and all routing tables are consistent. Each NNM is configured with a different priority level used to determine who will continue mapping the network when another NNM is detected.

Race conditions

When more than one NNM is actively trying to discover the devices of a network, some mechanisms must be implemented in order to avoid conflicts when configuring devices, i.e. writing the device identifiers. In our approach these mechanisms are based on forcing all routers to be configured by only one NNM (that may be different for each router). This is achieved by programming Network Node Managers to write to a specific register, called GPN (Granted Port Number), the port number used to access to the router. If a NNM discovers a router with a valid GPN value that does not match the port number used to access it, the NNM will not immediately try to configure this router and will not look for other devices attached to this router.

Router configuration

As each router can be configured by only one NNM, some information must be hold by the router regarding its NNM. Our approach relies on the use of logical addressing to provide direct communication with the NMM of the router. Its logical address is stored in a specific register and the routing table is programmed to route incoming packets with this logical address to the GPN port. If the router configuration is consistent with the status of the network, the packets will be routed to other routers with the same NNM owner, until they reach the NNM itself.

Network Node Manager arbitration

The previous mechanism is used by a NNM to interrogate the NNM that is the owner of a router, regarding its status and priority level. If no reply is got, or the reply contains a priority level lower than the interrogation, the GPN will be overwritten with the value of the new NNM, and the router configuration will be updated accordingly. On the other hand, if the NNM receives back the message, it means that it already owns the router and a loop in the network has been found.

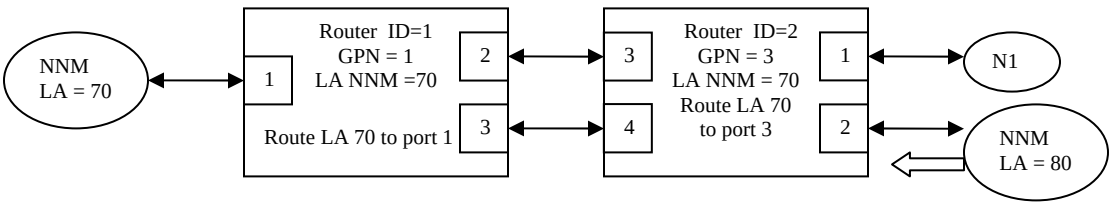

Figure 2: A new NNM is connected to an already configured network. The new NNM interrogates the NNM owner of the router to check its status and priority level.

The NNM interrogation message contains the identifier and the priority level of the NNM, the router identifier, and the return path from the router. The total return path is built by the NNM receiver. The reply message contains the identifier and the priority level of the NNM receiver and includes the interrogation message. The NNM receiver shall stop mapping the network if it receives a message with a priority level higher than their own.

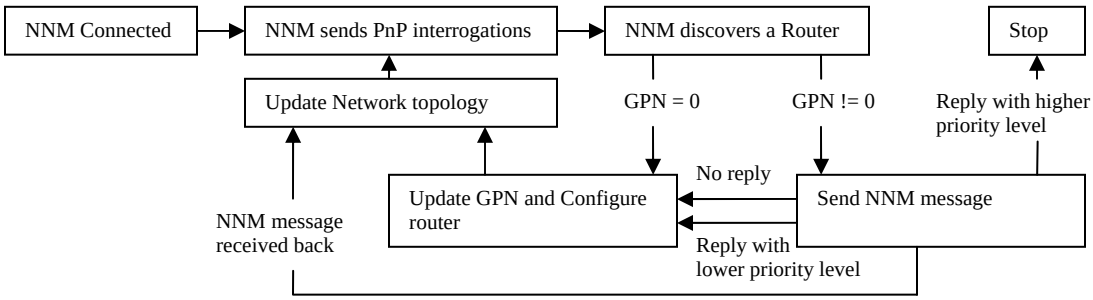

Figure 3: Basic flow of the Network Discovery procedure proposed.

Once a Master NNM is configured in an operative network, the other deactivated NNMs act as hot backups by periodically polling the Master NNM with NNM interrogation messages.

New device announcement

When a new intelligent device is connected to a router, it can notify its presence using the NNM logical address stored by the NNM Master. In case of passive devices, the Master NNM can detect them by polling mechanisms or by active notification from the router, as described in the PnP standard. If the new device is a router, network discovery procedure should be executed to map a possible new subnet.

CONCLUSIONS

A methodology for network discovery and network configuration has been presented using the SpaceWire Plug-and-Play protocol (SpW PnP) defined by the SpW PnP Working Group. The proposed approach supports any arbitrary change in the SpaceWire network topology and provides fault tolerant capabilities by using multiple Network Node Managers. It does not require previous manual configuration and it may become the first step towards the definition of a complete set of PnP services for SpaceWire.

REFERENCES

[1] SpW PnP Working Group, “SpaceWire Plug-and-Play Draft A”

[2] Proposed SOIS Plug-and-Play architecture and Resulting Requirements on SpaceWire Mapping, International SpaceWire Conference 2007.