Plug and Play for SpaceWire Networks: Centralized Algorithm

Ksenia Khramenkova
State University of Aerospace Instrumentation
Saint-Petersburg, Russia
ksu.khramenkova@gmail.com

Abstract

SpaceWire solves the problem of communication protocol to the existing on-board equipment. This standard was developed for the network nodes and switches interconnected via high-speed bidirectional serial connection. The communication network can be quite large, thus forming another problem - setting and monitoring of connections / disconnections in the network. Plug and Play - an algorithm for computer-aided learning of the network and setting up basic configuration of devices and connections. In other words, without a human operator the switches will be configured with routing tables and the nodes will be given logical address. There are two types of algorithms: the centralized and decentralized. In this paper we look at the centralized algorithm for computer-aided learning of the network and setting up the network settings. We show the results of simulating this algorithm and make some conclusions.

Index Terms: Embedded Networked Systems, SpaceWire, Plug and Play, RMAP.

I. Introduction

At present there is no standardized algorithm for Plug and Play for SpaceWire. Conventionally, all the existing algorithms can be divided into centralized and decentralized.

The centralized algorithm is based on the presence of only one network manager, which performs the configuration and the setup of the network. The disadvantage of this method is obvious: if the device fails, the network will have to configure again, the information will be lost.

The decentralized algorithm solves this problem. But on the other hand, it imposes additional requirements on the network and centers, which carry out the computer-aided learning of the network.

You must allow arbitrage between these centers in such a situation as: if two centers want to claim one node; if the switch is already set up the first center, then how do we tell this to the second center; who set up separate switches in the network during the network learning process.

It should also be noted that the Plug and Play algorithms can be different levels of complexity.

Some can be used for simple regular structures, some - for a network of arbitrary topology.

With the financial support of the Ministry of Education and Science of the Russian Federation

Some algorithms may be more complex and incorporate many features of the network and take into account the collected statistics during normal operation, etc.

Therefore, some algorithms can have a number of limitations, the most common of which are: the algorithm can’t be used for all network topologies; the solution which a Plug and Play algorithm provides may not be optimal. The equipment should be also taken into account, each piece of equipment may differ from another by a set of parameters and functions, this adds to the limitations or puts additional conditions to the application of the Plug and Play algorithm: the algorithm will stipulate a set of parameters and functions that are necessary for a full and proper functioning algorithm. The creation of an universal algorithm is extremely difficult [1].

One of variants of algorithm Plug and Play is algorithm offered by University of Dundee. The offered variant supports presence both one network manager, and several. All devices can be divided into two levels. First level devices support elementary network functions: discovering of devices, network management, router and channel configuration. At first level network can exist only one network manager. The second level devices have to support the wide set of functions and more complex algorithm. At second level network the several network managers can exist. Important feature of the given realization is that fact that a considerable quantity of parameters is involved in devices. For successful functioning the devices should support difficult algorithms (maintenance only RMAP protocol is not enough) [2].

The Plug and Play algorithm offered by NASA demands change existing devices software. During discovering of a network the part of functionality is transferred on routers which should find connections / disconnections devices or group of devices. For the each Plug and Play function in the protocol the format of a packet is provided [3].

II. THE SPACEWIRE STANDARD

SpaceWire standard was developed under the auspices of the European Space Agency. This is the standard for high-speed connections and networks for use onboard a spacecraft. The purpose of the standard is to facilitate the design of high-speed onboard data processing systems, cost reduction, promotion of interoperability between the equipment and data processing subsystems, re-use of equipment for different tasks. The equipment is connected to the SpaceWire network, which allows for the exchange of discrete messages, control and time information.

SpaceWire Plug and Play works on the network level. The basic concepts of packet routing networks are introduced. Devices in a network exchange packetsby means of communicational channels. Packets of data are the smallest elements of data that are handled at the network level. Packets are regarded as being indivisible by the networklevel and are transported as a whole across a network. SpaceWire packets have a simple structure: destination (the address of the destination node), cargo (the data to deliver) and end of packet marker (a special character which indicates the end of a packet).

Flow control is used to manage the movement of packets across a link connecting node or a router to another node or router. A node or router accepts data only when buffer space for that data is available in the receiving node or router. When the receive-buffer becomes full the receiver stops the transmitting node from sending any more data.

The wormhole routing is used in SpaceWire networks. Each packet contains a header which holds the destination address either as the route through the network or as the identity of the destination node. As soon as the header for a packet is received the switch determines the output port to route the packet to, by checking the destination address. If the requested output port is free then the packet is routed immediately to that port. Wormhole routing cuts down on the amount of buffering used within each switch, compared to a store and forward technique.

Also a header deletiontechnique is used in network. The first header data character (destination identifier) of a packet is used to specify the router output port address. When a packet is received at a routing switch its first destination identifier is checked to determine the output port to route the packet through. The first destination identifier of the header is then deleted and the packet passes through the switch without this first destination identifier. The second destination identifier of the original header (now the first destination identifier) is used as the first.

In the SpaceWire network several types of routing are used, but on this stage Plug and Play uses only two types: path and logical. With path addressing the destination address is specified as a sequence of router output port numbers, used to guide the packet across the network. Path addressing is simple. Its drawback is that the destination address can become relatively large;the length of the destination address can vary depending on where the destination is located on the network relative to the source. In logical addressing each destination has a unique number or logical address associated with it. When the source node transfers a message to a destination node, it simply addresses the packet with the logical address. To support logical addressing each routing switch is provided with a routing table. This tells the router the output port to transfer a packet to, for each possible logical address. For a reasonable sized network the routing table can become fairy large [2].

III. Remote memory access protocol

This protocol is used for accessing the remote program elements available in the device. Its primary purpose however is to configure a SpaceWire network, to control SpaceWire units and to gather data and status information from those units. RMAP may be used to configure SpaceWire routing switches, setting their operating parameters and routing table information. It may also be used to monitor the status of routing switches. RMAP may be used to configure and read the status of nodes in the SpaceWire network. For simple SpaceWire units (devices) without an embedded processor, RMAP may be used to set application configuration registers, to read status information and to read or write data into memory in the unit. For intelligent SpaceWire units RMAP can provide the basis for a wide range of communications services. Configuration, status gathering, and data transfer to and from memory can be supported [3].

IV. Centralized plug and play algorithm

SpaceWire Plug and Play includes: discovering, identification and configuration of devices. It doesn't require support of other standards, it only needs RMAP.

A. Defining the network and the devices

The philosophy of the SpaceWire Plug and Play technology is to provide maximum compatibility with legacy SpaceWire devices (nodes and routers); to support all forms of SpaceWire network addressing.

The network consists of nodes and routers (devices). Each device has several ports. In the centralized algorithm it is assumed that node has only one port. The router has at least one active port. A connection can be established for any set of ports. By SpaceWire standard the router can have up to 31 ports. The active ports can have non sequential numbers (for example, 3, 5 and 7). Devices connect to each other by links.

Routers carry out function of transfer of packets using path and logical addressing after correct setup the routing table. A node can't be sending device. Routers and nodes consider thata packethas arrived to its destination if header equals zero, it means that thepacket has to go to the configuration port of the device and has to be analyzed, for a response to be formed. It is necessary for the device to know from what port the packet has come. It is necessary to uniquely define the path for a response packet. In the case of node, this is not a problem, as at it only has one active port. In the case of a router this problem is solved as follows:the port number though witch the responding packet has to go though is written to the data of packet. So this way, it becomes known through what port the packet has been received.

In this versionof algorithm there is a work to the main registers of router, a minimum necessary for correct functioning of the device in a network: the routing table, the notification register for definition of output port for response to Network Manager. In the future it is possible to add other parameters, for example speeds and adaptive group registers.

The configurable node parameter is its logic address. Node also has to correctly receive, analyze the RMAP packets and form the response.

SpaceWire Plug and Play doesn't limit network topology and network size. However it is necessary take into account the response time on the packet. The timer is used for this. The time properties for processing, sending packet and forming response packet are known, so, it easy to calculate the time needed to get to destination device, forming response and receive it by Network Manager. In case of end of timer and not receiving the response the packet will be sent again. The response packet cannot be received by several reasons: the output channel one of routing devices is disconnect, the input channel one of devices is disconnect, the packet was corrupted during transmission and destination device deleted it. If network is rather big time, the waiting time will be rather big. And at this time in network some changes can occur. As a result of this the limitation is added to algorithm: devices can't connect or disconnect during first discovering.

B. SpaceWire Plug and Play organization

During network discover exists the initiator or Network Manager. All other devices are targets. The Network Manager use RMAP for coordination with other devices. At run time the Plug and Play algorithm Network Manager is always active. In case that contact with this device is lost the network configuration and administration are impossible. Therefore at failure of the initiator of requests it will be necessary to start the algorithm again.

As mentioned above Network Manager use RMAP, namely: read, response to read, write with acknowledge commands. The reply to write commend is needed to attest, that failure wasn't exist during writing data. When sending to the network the initiator saves packet in memory and calculate time requiring for packet get to destination and receive reply to it. In case if response packet hadn't received, the Network Manager would send packet again. So, if configuration or response packet was corrupted, the packet would be sent again. But as mentioned above, if network is rather big, the waiting time will be long. So configuring can take a lot of time. After receiving correct reply packet the relevant packet will be deleted from memory.

C. The Network Manager Algorithm

The Network Manager starts with sending on first connected device command read device identificator. If this is router, the reply will be 1, in other cases – node. Apparently that for network organization is need that the first device has to be router.

The initiator after determining that first device is router, send request to read its vendor identificator. It's necessary for define if this device had been defined or not. If this value isn't in list Network Manager, the router will be configured. In that case the configuration of routing table (Network Manager has information about router output port and has logical address (33)) and notification register is occurred. If before this router the nodes had been discovered and configured, The Network Manager would configure routing table with this nodes. After the initiator read the active ports register. By receiving value it define what port is active and sent there packet for read device identificator. If it does determine node, this device will be given a logical address. At the same time the configuration all discovered routers in network are occurred with new node.

The process of discover the simply network is demonstrate on two pictures below (Fig. 1 and Fig. 2).

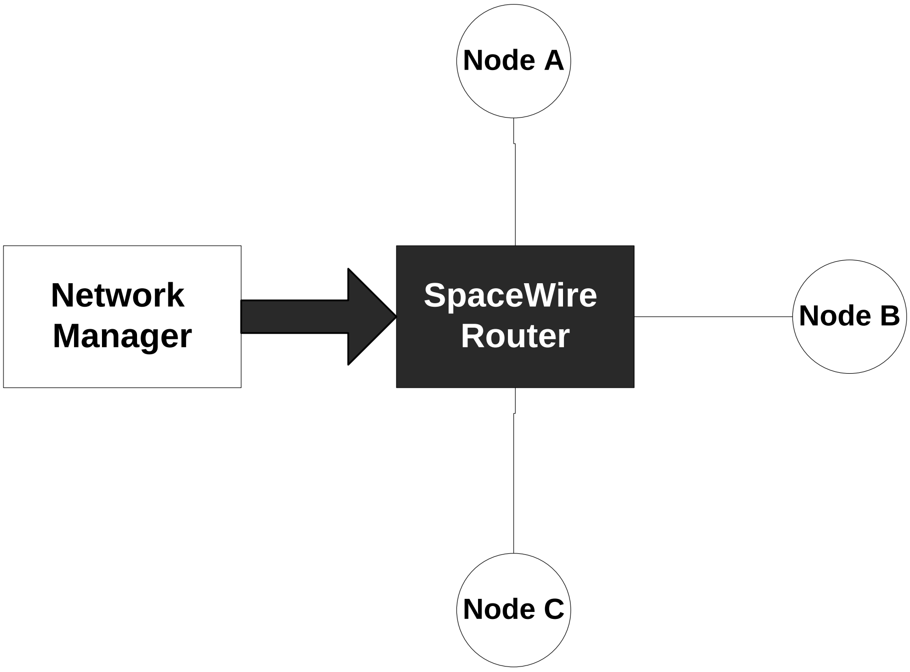

Fig. 1 Discover SpaceWire router

The Network Manager find out router, connected directly to it. After discover and configuration this device the register of active ports will be read. The Network Manager identify another three connections and sent there packet for determinate the type of devices. There are nodes.

The Network Manager can function in one time with several devices. If it detects another 3 router connections, it will work with them in one time.

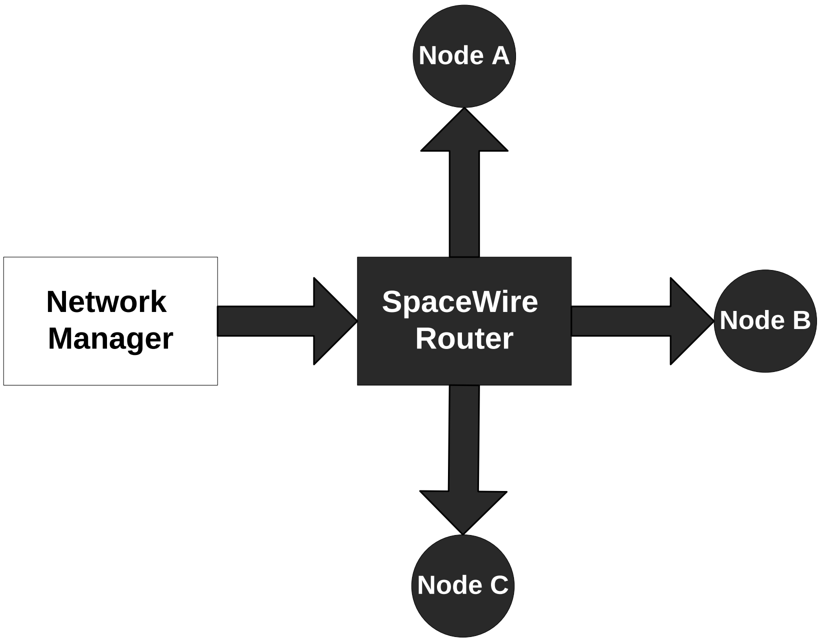

Fig. 2 Discover nodes

When new devices is discovering depend on its type (router or node) the network configuration is taken place. That is to say the configuration is occurred when network is discovered, but after discovering. For example, if in network had already discovered 3 routers and new device is node, the 3 routers would be configured with this node (routing table) after given node the logical address. If in network had already discovered 5 nodes and new device is router, the router will be configured with them. But it means that Network Manager is needed to have enough memory for store this information.

D. Modeling and results

Because of simulation software functioning of the distributed network it was necessary to write the main functions of router and node software and Network Manager: receive, analyze, sent and form configuration and response packets. The standard has already realized in used software, so it’s no need to describe all components of network and devices. They have the basics and standard parameters as SpaceWire devices. For creation noise in channel it needed just white the error bit rate.

By specified algorithm the modeling program was created using the simulation software described above. The two topologies were discovering: tree and lattice. Channel speeds are 10 Mb/sec. Apparently that running algorithm time is depend on network size. In average for first router discovering the 4 packets are required. For node – 2. It assuming that all packets were delivered to destinations, and were received the responses by Network Manager. With growing network size also is growing the time required for packet to get to destination device and back. If error bit rate isn’t zero the time will grow, and predict finish time is impossible as predict corrupted bit. So the time for each packet increases twice, and algorithm time grows.

Assuming that error bit rate is zero the modeling discovering time is 99230 ns and 130160 ns accordingly.

Also the channels with nonzero bit rate are researched: 10^{-12}, 10^{-16}, 10^{-17}. The modeling time was 300000000 ns.

The tree topology is shown on Fig. 3.

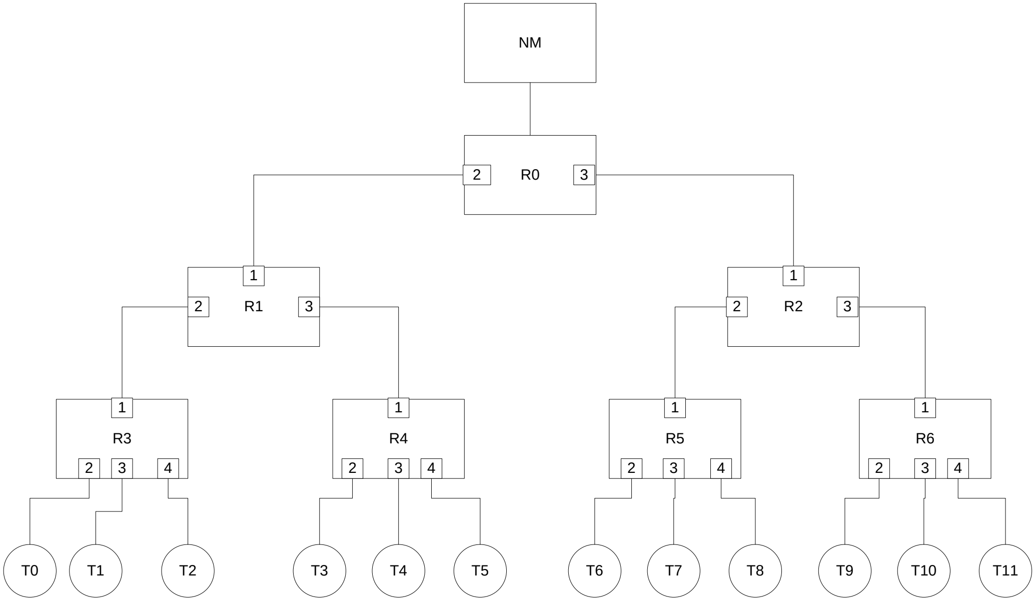

Fig. 3 Tree topology

With error bit rate 10^{-12} the network hadn't configured. With 10^{-16} the result was the same. With error bit rate 10^{-17} the network was successfully configured at 99230 ns.

For second topology, shown on Fig. 4, result is the same. Network was configured with error bit rate 10^{-17} at 130160 ns.

For network topology shown on Fig. 1 the configuration network time with error bit rate 10^{-17} is equal for time with zero error bit rate is 11730 ns. With standard error bit rate 10^{-12} the configuration time is 29207340 ns.

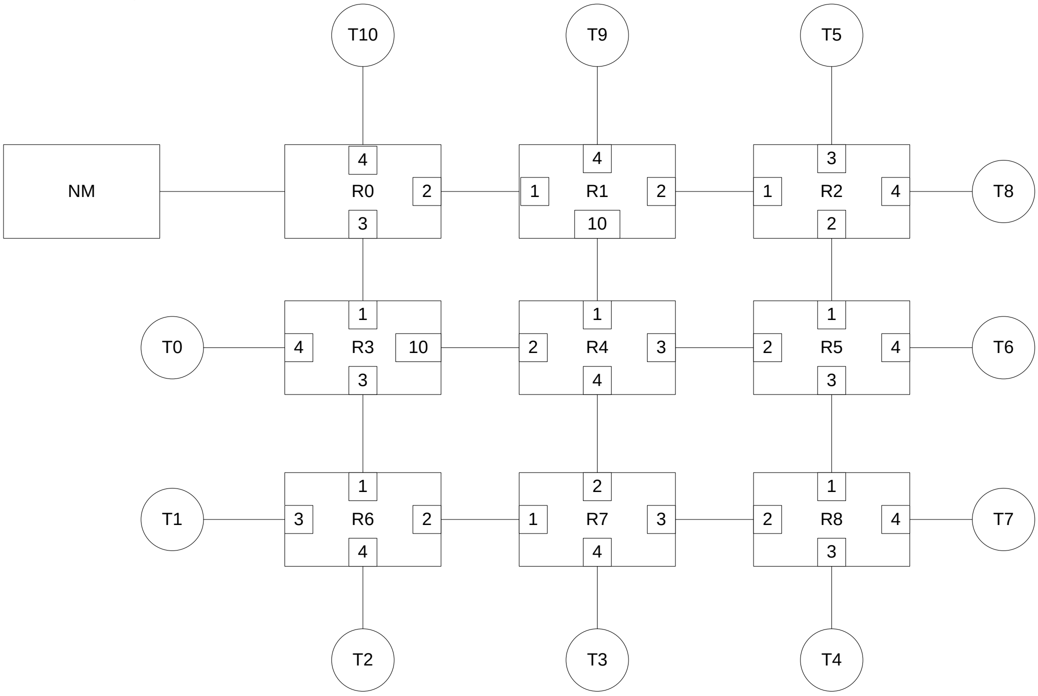

Fig. 4 Lattice topology

With error bit rate 10^{-12} the beaks in connection channels are occurred very often, so network configuration takes a long time. Network Manager will send repeated packets, but because channel is disconnected packets will be deleted by devices because it’s can’t send them.

Given active connection standard SpaceWire prescribes devices send to output channels the control codes all time. It means that channel is active. The same codes all time receives by device. If device receives any code what doesn’t expect on this stage of link work, the connection will immediately be broken. The neighbor stops receive codes and put an end sending codes. After 20000 ns devices will try to recover connection.

So, with error bit rate $10^{-12}$and 10^{-16}often the connection break occurs, therefore send packet to some devices fails, and network configuration isn’t succeed.

V. Conclusion

For simple network shown on Fig. 1 using simulation software functioning of the distributed network the results are given: with error bit rate $10^{-12}$the configuration time is 29207340 ns. With 10^{-17}– 11730 ns. The time with zero error bit rate is the same. For tree topology the results are: with error bit rate 10^{-17} the configuration time is 99230 ns. With standard error bit rate 10^{-12} it will take a much more time. For lattice topology the results are similar, with error bit rate 10^{-17} the network was configured at 130160 ns. With standard error bit rate as a tree topology the algorithm need to have a much more time for configure network. It is occurred as a result of frequent channel breaks.

It need to be mentioned that with standard error bit rate (10^{-12}) the network also will be configured but for a long time. Modeling 300000000 ns takes about 6 hours. So some limitations appear for testing algorithm with standard error bit rate.

References

[1] Koblyakova L., Khramenkova K., «Administration and configuration of communication SpaceWire network», 9th Conference of Open Innovations Framework Program FRUCT, Petrozavodsk, Russia, 2011.

[2] ESA (European Space Agency), standard ECSS-E-50-12A, “Space engineering. SpaceWire – Links, nodes, routers and networks. European cooperation for space standardization”, ESA Publications Division ESTEC, Noordwijk, The Netherlands, 2003.

[3] ESA, standard ECSS-E-50-11 Draft F, “Remote memory access protocol”.