Dynamic Network Reconfiguration and Transparent Failover in Fieldbus Protocol (PESNet)

Yoon-Soo Lee*, ***Regular Member***

ABSTRACT

The dynamic network reconfiguration feature for a network can be important to applications where nodes in a network may be removed or added. Here, a discussion about how a network can be dynamically reconfigured in a fieldbus protocol which operates in a counter rotating dual ring topology and behaves in a master-slave oriented fashion is made. The focus is on what the roles of the master node and the slave nodes are during network reconfiguration. Moreover, a method to allow failover between redundant nodes in the network is also mentioned in order to take advantage of the dynamic network reconfiguration feature. While the discussion is focused on modifications to be made on PESNet, the idea can be easily employed to any protocol that behaves in a master-slave fashion.

Key Words : Fieldbus Protocol, Dynamic Reconfiguration, Failover, Replication

요 약

동적으로 네트워크를 재구성할 수 있는 능력은 네트워크상에서 노드가 빠지거나 추가되어야 할 때 매우 중요한 요소이다. 여기에서는 마스터와 슬레이브 형태로 상호통신이 이루어지는 카운터 로테이팅 듀얼 링 토폴로지를 사용하는 필드버스 프로토콜에서 동적으로 네트워크를 어떻게 구성할 수 있는지에 대해서 논한다. 동적으로 네트워크를 재구성하는데 있어서 마스터와 슬레이브의 역할을 서술하는데 주안점을 두었다. 그리고 더 나아가 동적으로 네트워크를 재구성할 수 있는 네트워크에서 중요한 기능이라고 할 수 있는 노드가 죽었을 때 그 노드의 기능을 대체할 수 있는 노드가 작업을 이어서 할 수 있는 failover를 가능하게 하는 방법을 논한다. 기본적으로 여기의 내용은 PESNet이라는 기존의 프로토콜을 기반으로 모든 것을 설명하지만, 여기서 논의되는 방법은 마스터와 슬레이브로 동작하는 프로토콜에 적용될 수 있을 것이다.

Ⅰ. Introduction

Dynamic network reconfiguration is an essential feature in networks to provide robustness and flexibility. The feature can not only allow nodes to fail and recover and then rejoin the network, but also provide a method to swap nodes during operation known as hot-swap.Hence, the overall network becomes more robust to failure and flexible to maintain. Robustness and flexibility may be couple of the desirable qualities critical to the real-time distributed control network. However, in real-time distributed control networks, sometimes it is more important for the required nodes to be operating than simply keeping the network alive. It is important to make sure that certain set of nodes must be kept operating as each node consisting the network may play a critical role in the distributed controled system.

PESNet is a fieldbus protocol for Power Electronics System Network[1]. The development of PESNet is intended for designing power electronics systems in a distributed manner[2]. The idea is to construct the system by integrating modularized

* 삼성SDI(yoon-soo.lee@samsung.com)
논문번호 : 08031-0526, 접수일자 : 2008년 5월 26일

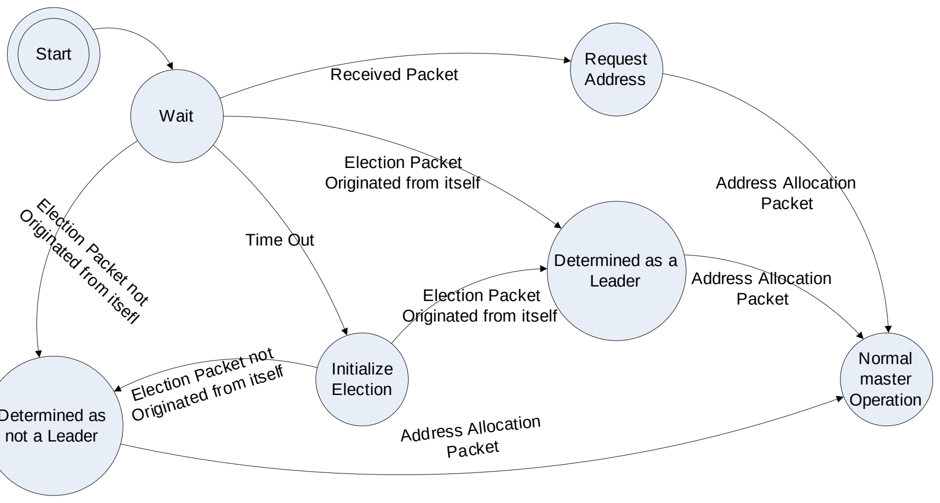
Start
Wait
Received Packet
Request Address
Election Packet Originated from itself
Address Allocation Packet
Election Packet not Originated from itsefl
Time Out
Determined as a Leader
Election Packet Originated from itself
Address Allocation Packet
Normal master Operation
Election Packet not Originated from itself
Initialize Election
Determined as not a Leader
Address Allocation Packet

Fig. 1. State diagram for the operation of a master node during network reconfiguration process

units called PEBB (Power Electronics Building Block) via counter rotating dual ring network. The role of a PEBB is either a master or a slave. While the master holds the control logic and sends out specific instructions to the slaves according to the control logic, the slaves are responsible for the actual control of the system as instructed from the master. It is possible to imagine a slave node to fail and recover or be replaced. In such cases, the network must be reconfigured due to the rejoining node. Also, there must be a replicated node of the failed node in order to continuously service the commands from the master. The previous version of PESNet (ver. 2.2) does not accommodate these kind of features.

Provided that real-time systems such as power electronics systems must be reliable to a certain extent, robustness becomes a critical issue. As mentioned from above, dynamic network reconfiguration will allow room for improving robustness to the network by making hot-swap and silently recovering from failure possible.

About providing dynamic network reconfiguration mechanism to fieldbus protocols, Especially for PESNet is discussed in the following order. First the limitation of PESNet 2.2 is examined, and the consideration points for dynamic reconfiguration are stated. Next will be the implementation.

Ⅱ. Limitations of PESNet 2.2

The specification of PESNet 2.2 can be found in Jerry Francis's work on PESNet[3]. The proposed idea for network configuration can be said to be somewhat static. The master which happens to carry the control logic is hard coded to have address 0x1. The node to have address 0x1 broadcasts an address allocation packet on the network. The packet contains the next available address which is to be assigned to the node receiving the packet. Once the address is assigned to the node, the next available address is incremented and passed on to the next node. This sequence continues until the master receives the address allocation packet. This operation only happens once after the system has started. The address allocation may be dynamic, but the address can never change for the nodes and new addresses cannot be allocated anymore. Therefore, the configuration of the network becomes static.

This configuration method can only work for optimistic situations when the failed node keeps its network address until it has recovered from the failure. Not being able to add new nodes is a drawback. Also, the need of having a predefined designated startup master with hard-coded address 0x1 must be modified as well as it limits flexibility and reliability. The system can encounter a problem as the predefined designated startup master fails.

Therefore, the implementation of the protocol must be changed so that network address assignment can be done whenever needed.

Ⅲ. Consideration Points

The following are the issues that are to be considered for augmenting the protocol to allow dynamic network reconfiguration. First of all, additional properties that are required for taking advantage of the dynamic network reconfiguration feature must be considered. Secondly, the role of the master and slave during the process must be identified. And finally, the steps for dynamic network reconfiguration according to the considerations made from above must be defined.

As stated in the introduction, dynamic network reconfiguration can make such features as hot-swap and failover to work. However, the feature itself is insufficient to handle hot-swap and failover. In order to add such features, replicated nodes are required for continuous normal system operation. When introducing replication, there must be some sort of group communication method in existence. The replicated nodes can keep consistency by total ordered multicast[4]. Since the network topology is in a ring form, it is safe to assume that total order is preserved as every packet traverses each node sequentially in the same order. Granted the total order by the topology, the only additional work required is introducing group address to the protocol.

The role of the master and slave does not need much change. However, the existence of replicated master or multiple masters may affect the operation of initializing the network reconfigure process. As stated in the section of describing the limitations of PESNet 2.2, one of the improvement points was to eliminate the need of predefined startup master. For this reason, a task to elect the master to initiate the network configuration at startup is necessary. Also, a slave node may have to request for an address as it is newly added to the network or has rejoined the network after recovering from a failure. In this scenario, the slave must request for an address to one of the existing masters.

According to the considerations made from above, the steps required for dynamic network reconfiguration can be described as follows.

1. Elect network configuration startup master on startup.
2. Allocate an address to each node without an address.
3. Allocate group address for replicated nodes.
4. Initialize node configuration according to the control logic.
5. Start normal operation.
6. OPTIONAL – Recovered or newly added slave node requests for an address to a master. And repeat step 2.

These required steps depicts the overall behavior of the network for dynamic network reconfiguration. The detailed implementation of the master and slave will be shown in the next section.

Ⅳ. Implementation

First we discuss on how the master and slave nodes must be implemented. Then we discuss on how to keep slave nodes consistent in order to provide the transparent failover functionality.

In order to illustrate the operation of master and slave nodes, state diagrams are used to assist the explanation. The circles indicate the state and the arcs indicate the state transition according to the packet they receive or some internal activity. Refer to figure 1 and 2 for the state diagram for master and slave node’s operation during network reconfiguration.

The method of keeping the replicated slave node’s consistency for failover operation will be outlined after the illustration of master and slave

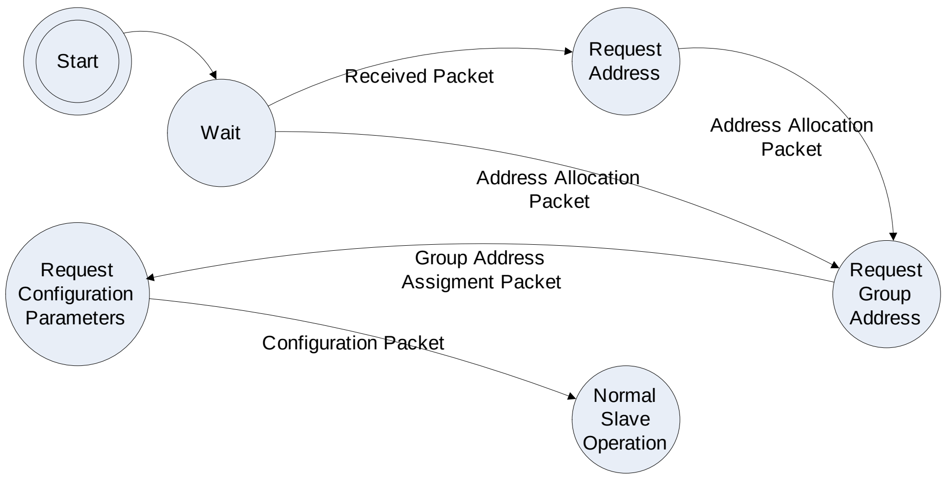
Start
Wait
Received Packet
Request Address
Address Allocation Packet
Address Allocation Packet
Request Configuration Parameters
Group Address Assigment Packet
Request Group Address
Configuration Packet
Normal Slave Operation

Fig. 2. State diagram for the operation of a slave node during network reconfiguration process

node's operation during network reconfiguration.

4.1 Master and Slave node operation

4.1.1 Master Implementation

As the network is powered up, a counter is triggered in every master. The counter is set to a value which will give the master sufficient time to determine whether the network has just started or is in normal operation mode. The master will wait for packets before the counter overflows. The master determines that it has been added to the network during normal operation if any packet arrives before the counter overflows. If this is the case, the master will request for an address to another master. Otherwise, it is an indication that the network has just been started up.If this is the case, the startup master election process takes into place. The election is based on a unique hardware property called a Slot ID which is a numeric value derived from a PEBB's physical location. This property can be replaced with any other property that is unique to a node. The master with the highest Slot ID is elected as the initial startup master. The LCR algorithm is used for the election process and is done in the following manner[5].

```
var statep
begin
    statep="unknown";
    value:=SlotID;
    send value to Nextp;
    while statep ≠ "leader" do
    begin
        receive a message v;
        if v = value
            then statep:="leader"
        else if v > value
            then value := v;
        end if
    end
end
```

Every master broadcasts its Slot ID. The master consumes the broadcast message if it discovers a message that originated from a master with a lower Slot ID. When a master receives a message with its own Slot ID, this indicates that it has been elected as the designated startup master. The network address assignment process starts as the election process is finished. The remaining process of assigning addresses to the nodes stays the same as proposed for PESNet 2.2. The network address allocation is finished when the network address allocation packet is received by the designated startup master. Then the network is prepared for the nodes to communicate normally using their network addresses.

4.1.2 Slave Implementation

Unlike the masters, the slaves are not involved in the startup master election process. Therefore, they remain idle until the election process finishes. Recall that the network enters normal operation mode when the startup master is elected among the masters and initiates an address allocation. When a slave is added to a network that is already operating or has recovered from a failure, it will observe packets containing regular network traffic. Therefore, the master knows that the network is already in normal operation mode and will not receive an address allocation packet unless it requests for one to a master. The slave must give one of the masters an indication of its existence. Notification of its existence is done by sending an address request packet to the masters' group address. As the packet is not destined to a specific master, the first master to detect the packet can then issue an address allocation packet on the network for the newly added node.

Since every packet will traverse the network in a certain sequence, the task of address allocation can be distributed among the masters depending on their position on the network.

4.2 Replication and Failover

As mentioned previously, there must be a mechanism to keep consistency between the replicated nodes in order to provide failover functionality. Such operation is called replication. There are two different ways in supporting this operation. Active replication and passive replication are commonly practiced replication methods for fault tolerant systems[6].

In genral, the failover process must occur quick enough so that it will not affect the system's operation. Therefore, active replication is chosen to be used as it is known to better suit real-time system's needs[7]. The consistency of the slave nodes in active replication are kept by receiving the same commands from the master node. This can be easily done by the group communication method by multicasting which was mentioned previously. Commands from the master node that are targeted to a certain set of replicated slave nodes can be sent to them via the group address which was assigned to those certain set of nodes.

It is usually required for all the slave nodes to respond to the command from the master node for active replication. However, general implementation of active replication in PESNet may result in saturation of packets in the network because of its unique network characteristic. Therefore, only one slave node which receives the multicast command first responds to it by piggybacking the response to the command packet. This method provides the network to perform silent or transparent failover as the failover process does not require any tasks. However, this decision has some drawbacks.

First of all, while the general implementation of active replication is capable of resolving Byzantine faults, also known as logical faults, suggested decision here cannot tolerate Byzantine faults as only one slave is responding to the master's command. Secondly, there is a chance for the multicast command with the response contained in it getting lost due to a node failure. The lost information may be recovered at some point if the network is capable if detecting the fault and handling it, but it will introduce delay for the master to receive the response. If Byzantine faults and packet loss is critical a general implementation of the active replication can be used.

Ⅴ. Conclusion

The discussion of on how to resolve dynamic network reconfiguration problem in PESNet shows how the master and slave should be implemented. Also, it showed that transparent failover functionality can be provided by using group addresses to the replicated set of nodes. Although PESNet is one type of fieldbus protocol, the idea may be applied to any network which behaves in a master and slave fashion. Some of the implementation details that inherit from the characteristics of PESNet were not specified here as the goal was to provide a general idea for implementing dynamic network reconfiguration and transparent failover among replicated nodes..

Through the discussion, not only have we formalized a way to reconfigure the network dynamically, but we also made the feature more useful by allowing replicated nodes in the network by introducing group addresses. As group address property is added to the protocol, active replication as a replication scheme becomes trivial by using multicast.

References

〔1〕 Milosavljevic, I., "Power Electronics System Communications," in Electrical Engineering. 1999, Virginia Tech: Blacksburg, Virginia.

〔2〕 Celanovic, I., et al. "A new distributed digital controller for the next generation of power electronics building blocks." 2000.

〔3〕 Jerry Francis, J.G., and Stephen H. Edwards, "Protocol Design of Dual Ring PESNet (DRPESNet)," in *CPES 2002 Power Electronics Seminar and NSF/Industry Annual Review.* April, 2002.

〔4〕 Guerraoui, R.; Schiper, A., "Software-based replication for fault tolerance," *Computer, vol.30, no.4, pp.68-74,* Apr., 1997.

〔5〕 Taolue, C., H. Tingting, and L. Jian. "Analysis of a leader election algorithm," in /spl mu/CRL. 2005.

〔6〕 Hengming, Z. and F. Jahanian. "Optimization of a real-time primary-backup replication service." 1998.

〔7〕 Budhiraja, N. and K. Marzullo. "*Trade offs in implementing primary-backupprotocols.*" 1995.

이 윤 수 (Yoon-Soo Lee) 정회원

2004년 5월 Virginia Tech Computer Science 졸업

2006년 7월 Virginia Tech Computer Science 석사

2006년 9월~현재 삼성 SDI PDP 개발팀 근무

<관심분야> Software Design/ Architecture, Software Platform, System Software, Embedded System