

Real Ethernet Protocol

Hong Seong Park, Myong Soon Jung, Weon Joon Kang

Dept. of Elec. and Comp. Eng, Kangwon Natl. Univ. Kangwon-Do, 200-701. Korea

(Tel: 82-33-251-6501 ; E-mail: hspark@kangwon.ac.kr, ([@control.kangwon.ac.kr">jms, tailhook](mailto:jms, tailhook))@control.kangwon.ac.kr)

Abstract

This paper suggests the methodology to guarantee the real-time service over Ethernet and TCP/IP and to solve the problems such as re-transmission due to collision and platform-transparency and independence. This paper proposes a new mechanism called Real-time Ethernet protocol combined the centralized medium control mechanism with the token passing mechanism. The centralized medium control mechanism is used to get the token, the right to transmit the data, and the token passing mechanism is used to return the token to the token controller and to transmit data within the specified time interval. The proposed Real-Time Ethernet Protocol is based on Java and Java RMI(Remote Method Invocation). The presented protocol is believed to work enough in real-time applications considering latency occurred due to the Java RMI, which is less small than data transmission time.

1. Introduction

Ethernet(or CSMA/CD) and TCP/IP are the protocols mostly used around the world and there are many applications based on them. Those protocols are frequently required to be used in and/or be applied to various industrial environments such as Factory Automation and Process Control. Applying Ethernet and TCP/IP to the industrial environments, there are benefits such as low cost for constructing network system and enabling integration of heterogeneous factory network into one network. As a result, we can manage factory circumstance effectively using Ethernet-TCP/IP. The factory/process control requires real-time constraints depending on the applications. But there is a barrier in application of Ethernet to real-time environments. That is, the barrier is the non-deterministic re-transmission mechanism of Ethernet when collisions occur. Moreover, those environments require the platform-transparent and -independent protocol working for heterogeneous platforms because the general factory system is the distributed system consisting of PC, DCS, PLC, and other dedicated devices.

To cope with collision problem, there have been some researches[1-3]. Rether(A real-time Ethernet Protocol) is a distributed token passing protocol to solve the collision problem at the data link layer[2]. But there were some limitations such as the maximum number of nodes and non platform-transparent and -independent protocol. Note that non platform-transparent and -independent protocol mean that the node for real-time applications should use the specialized data link layer designed for Rether. In [1], the combined protocol token passing with TDMA was suggested and the software, client-server based MAC protocol and the associated protocol was developed. But this research has still a limitation such as non platform-transparent and -independent protocol and increases the maximum number of node proposed in [2]. In [4], the real-time fast Ethernet switch called EtheReal was proposed, which provided bandwidth guarantees to real-time

applications on Ethernet without modification to H/W and OS on the host node. But this research requires the special switch, EtheReal so that there can be a barrier in extension of real-time systems. FL-net[3] is the protocol that guarantees real-time service with 50ms period for 32 nodes and supports periodic transmission, point-to-point and multicasting. But, all nodes should use the specialized FA link protocol layer over UDP to utilize FL-net. FL-net didn't solve the problem of the platform-transparency and -independence.

This paper suggests the methodology to guarantee the real-time service over Ethernet and TCP/IP and to solve at the same time the problems such as re-transmission due to collision and platform-transparency and -independence. In this paper, to solve the problems addressed above, a new Real-Time Ethernet Protocol based on Java, especially Java RMI(Remote Method Invocation)[5, 6], is suggested. Basically Java works in any platform which supports JavaOS or JavaVM. This paper provide the centralized medium control mechanism with the token passing mechanism similar to profibus[7]. That is, the centralized medium control mechanism is used to get the token, the right to transmit the data, and the token passing mechanism is used to return the token to the token controller and to transmit data within the specified time interval. Those mechanisms are implemented over TCP/IP because Java RMI is used so that the platform-transparency and independence are guaranteed. In addition, the Real-Time Ethernet Protocol uses remote objects and a timer component implemented with Java Beans. RMI is adopted to obtain the token between a token controller(server component) and nodes(client components). RMI enables user to access to remote object with the transparent location of the token controller on network.

Generally, it is disadvantage that the usage of Java and RMI results in more delays in latency time that the usage of TCP/IP itself. But it is believed that there is little effects on real-time applications because the Java RMI is used only in managing the token. And the presented protocol in this paper is believed to work enough in real-time applications considering those latency. In the presented mechanism, all systems can have the proposed Real-Time Ethernet Protocol running on the Ethernet. But it is noted that other researchers' Protocols are always able to work for real-time application only in the case that the station is implemented those protocols before the system is working.

In section 2, the state machines for the proposed Real-time Ethernet protocol are proposed and described both in the node side and in the token controller side. In section 3, RMI classes and user methods for the presented Java-based Real-time Ethernet protocol are introduced and the functions are presented. Finally, we give some conclusions in section 4.

2. The state machine of Java-based Real-Time Ethernet Protocol(JaRE)

The state machines in both a node and a token controller are shown in Fig. 1 and the interactions between the node and the token controller are also shown in Fig. 1. Token controller