

# 모듈 기반 퍼스널 로봇의 결함 허용 지원을 위한 네트워크 연결 유지 관리 기법

## Method of network connection management in module based personal robot for fault-tolerant

최 동 회\*, 박홍성\*\*  
(Donghee Choi, Hongseong Park)

**Abstract** - Middleware offers function that user application program can transmit data independently of network device. Connection management about network connection of module is important for normal service of module base personal robot. Unpredictable network disconnection is influenced to whole robot performance in module base personal robot. For this, Middleware must be offer two important function. The first is function of error detection and reporting about abnormal network disconnection. Therefore, middleware need method for network error detection and module management to consider special quality that each network device has. The second is the function recovering that makes the regular service possible. When the module closed from connection reconnects, as this service reports connection state of the corresponding module, the personal robot resumes the existing service. In this paper proposed method of network connection management for to support fault tolerant about network error of network module based personal robot.

**Key Words** : Module-base robot, fault-tolerant, connection management

### 1장 서론

퍼스널 로봇은 모듈이라 불리는 하드웨어 컴포넌트 단위로 구성되어 있으며, 모듈 간 통신을 통해서 작동하는 일종의 분산 시스템으로 사용되고 있다. 물리적으로 분리된 분산 모듈은 각각이 로봇의 특정 기능을 수행하는 하나의 집합적 단위이다. 이러한 모듈은 네트워크를 통하여 다른 모듈과 통신을 하며 데이터를 공유하게 된다. 모듈 간 데이터 통신을 통하여 모듈이 속한 전체 시스템의 기능을 수행하게 된다. 이러한 모듈로 구성된 퍼스널 로봇은 모듈 간 네트워크 인터페이스를 통해 연결되어 동작하는 일종의 분산 제어 시스템으로 볼 수 있다. 따라서 각 모듈에서 수행하는 로봇 기능을 정상적으로 수행할 수 있도록 하기 위하여 모듈과 모듈을 연결하는 네트워크의 기능이 중요하게 다뤄지고 있다.

퍼스널 로봇에서 모듈 간 인터페이스는 하나 이상 사용될 수 있기 때문에 언제든지 다른 종류의 인터페이스와 연결할 수 있어야 한다. 실제로 모듈기반 로봇 구조에서 사용될 수 있는 인터페이스에는 IEEE1394, Ethernet, USB, CAN, Bluetooth[9], Wireless LAN등 다양한 종류들이 있다. 그리고 이러한 인터페이스들은 필요에 따라 로봇 모듈 벤더에 의해 추가되거나 삭제될 수 있다. 따라서 모듈 기반 퍼스널 로봇에서 인터페이스의 종류에 상관없이 모든 인터페이스를 지원 및 수용할 수 있는 개방형 구조의 통합 소프트웨어 계층이 필요하다. 하나 이상의 인터페이스에 기반을 둔 모듈들을 하나의 가상 네트워크로 통합하는 데에는 고려해야 할 수많은 사항이 존재한다. 그 이유는 네트워크 종류마다 프로토콜, 사용방법, 네트워크 토폴로지 등이 모두 다르고 이를 유연하게 통합해야 하는 개방형 구조를 갖추어야 하기 때문이다. 종류

가 다른 자원이나 환경에서 여러 분산 요소들을 통합하는데 사용되는 것이 미들웨어이다. 미들웨어는 종류가 다른 여러 자원이나 환경 환경으로 발생하는 문제점을 해결하기 위해 필요한 서비스 및 프로토콜을 정의해 놓은 표준을 일컫는 용어이다.

본 논문에서는 모듈 기반 퍼스널 로봇에서 예측 불가능한 네트워크 오류에 대하여 이를 감지, 오류 보고, 네트워크 재 연결 등에 대한 Fault-tolerant를 지원하기 위한 네트워크 관리 기법을 제안한다.

본 논문은 다음과 같이 구성하였다. 2장에서는 모듈 기반 퍼스널 로봇의 미들웨어 구조를 설명하고, 3장에서는 미들웨어에서 수용하는 네트워크 장치들에서의 오류 감지에 대한 주요 메커니즘을 설명한다. 4장에서는 모듈 기반 퍼스널 로봇의 결함 허용 지원을 위한 네트워크 연결 유지 관리 기법 방안 및 세부 기능에 관하여 설명하고, 5장에서 결론을 맺는다.

### 2. 모듈 기반 퍼스널 로봇의 미들웨어 구조

본 논문에서 적용한 미들웨어[1]는 어플리케이션에 미들웨어의 서비스를 제공하는 스트리밍 계층(Streaming Layer, SL), 여러 이종 네트워크를 수용하는 네트워크 적응 계층(Network Adaptation Layer, NAL), 네트워크 의존적인 기능을 담당하는 네트워크 인터페이스 계층(Network Interface Layer, NIL)의 3 계층으로 구성되어 있다.

SL은 NAL을 이용하여 어플리케이션이 네트워크의 종류에 상관없이 원격 모듈에 존재하는 객체에 접근하기 위한 메커니즘을 제공하는 것이다.

NAL은 NIL에 추가된 여러 종류 네트워크 컴포넌트들을 통합하는 계층으로서 실질적인 미들웨어의 핵심 기능을 담당하는 계층이다. 이 계층은 메시지 라우팅, 이종 네트워크 간 모듈 어드레싱, 네이밍 서비스 등의 기능을 수행하며, 네트워크

저자 소개

\* 최동회 : 江原大學 통신멀티미디어學科 碩士課程

\*\* 박홍성 : 江原大學 통신멀티미디어學科 教授 · 工博

크 인터페이스 계층의 여러 네트워크 컴포넌트들을 통합한다.

NIL은 여러 종류의 네트워크 컴포넌트가 존재하는 계층으로서 네트워크 종류에 따라 하드웨어, 소프트웨어에 의존적인 부분을 수행하는 컴포넌트로 구성되어 있다. 이러한 네트워크 인터페이스 계층이 존재하는 이유는 미들웨어에서 새로운 네트워크가 추가 되거나 삭제될 때 일일이 미들웨어를 수정하는 일이 없도록 하기 위해서이다. 만약 미들웨어 구조가 정적으로 특정 네트워크들만을 지원하도록 되어 있다면 2장에서 언급한 미들웨어의 개방형 구조에 어긋나게 된다. 따라서 네트워크 인터페이스 계층이 반드시 필요하게 되며, 네트워크 인터페이스 계층은 각 네트워크의 의존적인 부분을 담당 및 관리할 수 있는 구조이다.

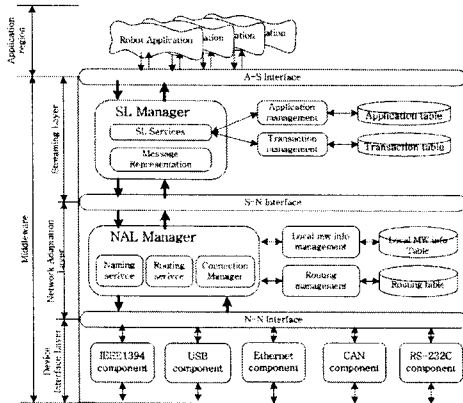


그림 1 미들웨어 구조

그림 1과 같은 미들웨어 구조에서 필요한 네트워크는 그 종류에 상관없이 언제나 미들웨어에 쉽게 추가할 수 있고, 모듈 기반 퍼스널 로봇에서 분산 어플리케이션 개발자는 네트워크상의 원격 또는 로컬에 존재하는 객체에 접근할 때 네트워크의 종류에 무관하게 미들웨어를 통해 손쉽게 어플리케이션을 설계 및 구현할 수 있다.

### 3장 네트워크 장치

네트워크 디바이스는 두 가지로 분류할 수 있다. 비정상적인 연결 오류에 대하여 상위 계층으로 이벤트를 발생하는 장치와 발생하지 않는 장치로 구분된다. 본 장에서는 전자에 해당하는 네트워크 장치의 오류 이벤트 발생 기법을 기술한다. 후자에 해당하는 장치는 4장의 네트워크 연결 관리자를 통하여 해결 방법을 설명한다.

#### 2.1절 IEEE1394

IEEE1394[6]는 물리 계층에 정의된 CONNECT\_TIMEOUT 상수가 유효 응답시간 이내에 연결 확인 응답이 발생하지 않으면, INTERRUPT를 발생한다. 이벤트 발생에 의한 연결 상태는 PHY register page0의 Address 10002의 6번째 비트(Connected 필드)에 저장된다. 해당 필드의 값이 "0"이면 disconnect 이고, "1"이면 connect 이다. 또한, 원격지의 Active port가 "Disable" 인 경우 PHY register의 page0의 Address 10002의 8번째 비트(Disabled 필드)가 "1"인 경우 Disable 상태를 나타낸다. 위에 기술된 레지스터상의 두 필드에 변화에

대한 이벤트를 사용자에게 알려준다.

#### 2.2절 USB

USB[7]는 물리적 연결이 해제되면 포트로부터 들어오는 전압 강하 또는 전압상승에 의해 감지된 값으로 판단한다. 결과 값이 '0'인 경우 USB 초기상태 또는 "Disconnected" 상태를 나타내고, 레지스터에 저장된 값에 의해 상위 계층에 이벤트를 발생한다.

#### 2.3절 CAN

CAN[8]은 ERROR HANDLING 기법 및 FAULT CONFINEMENT를 통해서 연결 오류를 감지할 수 있다. ERROR HANDLING에서 Transmitter가 ACL SLOT 동안 'dominant' bit를 모니터 할 수 없을 경우 ACKNOWLEDGEMENT ERROR를 발생시킨다. FAULT CONFINEMENT에서는 ("error active" 상태와 "error passive" 상태에서) 원격지로 데이터 전송이 안 되거나, 버스 상의 다른 노드들로부터 특정 원격지 노드에서 ERROR 발생에 따른 ACTIVE ERROR Flag 값을 받은 경우 "TRANSMIT ERROR COUNT"의 변화와 원격지로부터 "RECEIVE ERROR COUNT"의 값에 따라 원격지로부터 데이터 수신이 안 되었을 때 LLC로부터 상위계층으로 이벤트를 발생한다.

### 4장 네트워크 연결 관리자

그림2는 그림1의 미들웨어에서 NAL에서의 Connection Manager의 상태를 나타낸 것이다.

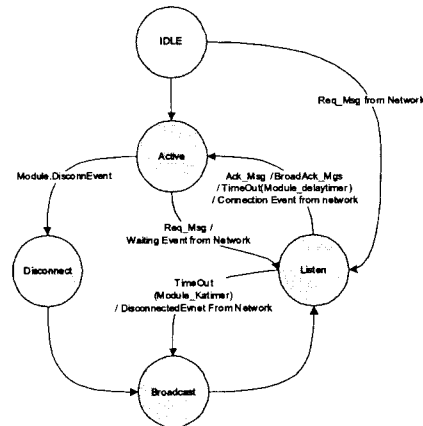


그림 2 네트워크 연결 관리자 상태도

NAL은 네트워크 인터페이스 계층에 추가된 여러 종류 네트워크 컴포넌트들을 통합하는 계층으로서 실질적인 미들웨어의 핵심 기능을 담당하는 계층이다. 이 계층은 메시지 라우팅, 이중 네트워크 간 Addressing, Naming 서비스 등의 기능을 수행하며, 네트워크 인터페이스 계층의 여러 네트워크 컴포넌트들을 통합한다. 또한, Network Connection Manager (NCM)를 통해 모듈 Fault-tolerant 지원을 한다.

첫 번째, Active 프로세스는 오류 검출 이벤트를 발생하지 않는 Ethernet 과 RS232C 네트워크의 경우 원격지의 모듈로 주기적인 Req\_Msg를 생성해서 전송한다. Req\_Msg에 대하여 Module.delaytime이 3회 이상 timeout이 발생하면 네트워크 연결 해지를 감지하고 Module.DisconnEvent를 발생시켜

Disconnect 프로세스로 상태전환을 한다.

오류 검출 이벤트를 발생시키는 네트워크 장치의 경우는 연결 이상 이벤트 수신시 Module\_DisconnEvent를 발생시켜 연결 모듈에 대한 연결 해지 및 원격지 모듈의 서비스 불가 이벤트를 미들웨어의 상위 계층으로 송신하고 Disconnect 프로세스 상태전환을 한다.

두 번째, Listen 프로세스는 크게 3가지 유형에 따라 나뉜다. 원격지로부터 메시지를 수신하는 경우, 네트워크 장치로부터 이벤트는 받아들이는 경우, 타이머 만료에 의한 타임아웃이 발생하는 경우이다. 먼저 외부로부터 메시지를 수신하는 경우는 4가지 유형의 메시지를 수신할 수 있다. ① Req\_Msg를 수신에 대한 Ack\_Msg를 송신 측으로 전송 후 다시 Listen 상태로 전환된다. ②Ack\_Msg를 수신시 로컬 모듈에서 Req\_Msg에 대한 응답 메시지로 Module.counter 값을 초기화하고 Active 상태로 상태전환을 한다. ③ BroadReq\_Msg를 수신시 원격지로부터 새로운 연결 또는 재연결된 것으로 로컬 정보를 원격지 모듈로 전송 후 Listen 상태로 상태전환을 한다. ④BroadAck\_Msg를 수신시 로컬 모듈에서 요청한 BroadReq\_Msg의 확인 응답으로 BroadAck\_Msg 수신 후 원격지로부터 모듈 정보를 수신한다. 수신한 원격지 정보를 통해 새로운 연결인 경우 NAL Manager의 Naming Service 와 Routing Service를 통해 원격지 모듈의 정보를 저장하게 된다. 원격지 모듈이 재연결인 경우 라우팅 테이블에서 해당 모듈의 정보를 Disconnected에서 Connected로 변경 후 Active 상태로 상태 전환한다. ⑤네트워크 장치로부터 DisconnectEvent를 수신시 해당 모듈 정보를 Connected에서 Disconnected로 변경 후 BroadCast 상태로 상태전환을 한다. ⑥Timer가 만료되는 경우는 2가지 타이머에 의하여 발생한다. Module\_Katimer는 원격지 모듈이 지속적으로 연결되어 있는지를 확인하기 위한 Req\_Msg를 송신 후 Ack\_Msg를 수신하기까지 걸리는 시간의 상한 제한 값이다. 만약 Req\_Msg 송신 후 Ack\_Msg를 수신하기 전에 Module\_Katimer가 만료되면, Req\_Msg 재전송을 위해 Active 상태로 상태전환을 하게 된다. Module\_delaytimer는 원격지로 연결되지 않는 네트워크를 통해 새로운 연결 또는 기존의 모듈이 재 연결되는 지를 지속적으로 확인하기 위한 BroadReq\_Msg를 송신 후 BroadAck\_Msg를 수신하기까지 걸리는 시간의 상한 제한 값이다. 만약 BroadReq\_Msg 송신 후 BroadAck\_Msg를 수신하기 전에 Module\_Katimer가 만료되면, BroadReq\_Msg 재전송을 위해 BroadCast 상태로 상태전환을 하게 된다.

세 번째, Disconnect 프로세스는 Active 상태에서 Disconnect\_Event를 발생으로 인하여 전환된 상태이며, 원격지 모듈에 대한 연결 불가 메시지를 사용자에게 알려주고, BroadCast 프로세스로 상태전환을 한다.

네 번째, BroadCast 프로세스는 해당 모듈이 네트워크 연결이 되지 않은 상태에 있을 때 새로운 연결 및 재 연결을 위하여 원격지의 모듈로 주기적 메시지를 보내 네트워크 연결시 원격지의 모듈에 대한 정보를 요구하는 메시지를 보내는 프로세스이다.

위 네 가지 프로세스에 의하여 미들웨어는 원격지 모듈이 비정상적인 연결 해지에 대하여 Fault를 감지하고, 원격지 모듈에 대한 정보를 사용자에게 이벤트를 발생하여 사용자는

모듈 에러로 인한 서비스 불가를 사용자에게 알린다. 연결 종료된 모듈은 서비스 가능한 모듈과 격리를 시켜 정상적인 서비스에 영향을 미치지 않도록 한다. 또한, BroadReq\_Msg 응답에 의해 수신된 원격지 모듈 정보를 NAL Routing table 정보와 비교하여 새로운 모듈이 추가되는 경우와, 기존의 모듈이 재 연결되는 경우에 대하여 NAL Manager의 Naming service와 Routing service 기능에서 Routing table을 갱신하여 새로운 모듈의 연결 또는 재 연결에 대한 모듈 연결 관리가 가능하다.

## 5장 결론

본 논문에서는 모듈 기반 퍼스널 로봇의 네트워크 오류에 관한 결함 허용 지원을 위한 연결 유지 관리 및 모듈 연결 관리 기법을 제안 하였다.

본 논문에서는 모듈 기반 퍼스널 로봇 전체 성능에 영향을 미치게 되는 비정상적인 네트워크 연결 종료에 대한 처리를 지원하기 위하여 두 가지 기법을 제안하였다. 첫째는 비정상적인 네트워크 연결 종료에 관한 오류 감지 및 보고 기능이다. 둘째는 연결 해지된 모듈이 재 연결시 해당 모듈의 재 연결 상태를 보고함으로써 기존의 서비스를 재개하여 정상적인 서비스를 가능하게 하는 서비스 회복 기능이다.

앞으로의 일은 미들웨어에 적용/구현하여 제안하는 방법의 성능을 검증하는 것이다.

## 참 고 문 헌

- [1] Gun Yoon, "Middleware Structure for Module-based Personal Robot", Journal of Control, Automation, and Systems Engineering, May 2004
- [2] Rondey A. Brooks, "A robust layered control system for a mobile robot," IEEE Journal of Robotics and Automation, RA-2(1):14-23, 1996
- [3] Ishiguro, H., Kanda, T., Kimoto, K, Ishida, T, "A robot architecture based on situated modules" Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on. Volume:3,199
- [4] Pardo-Castellote, G.; Schneider, S., "The network data delivery service: real-time data connectivity for distributed control applications " Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on 8-13 May 1994 Page(s):2870 - 2876 vol.4
- [5] Schwartz D., "cooperating heterogeneous system: A black-board based meta approach." Technical Report 93-112.Center for Automation and Intelligent systems Research. case Western reserve University, 1993
- [6] IEEE standard for a High Performance Serial Bus" IEEE std 1394-1995, IEEE1394 std 1394a-2000"
- [7] Universal Serial Bus Specification revision 1.1: September 23, 1998
- [8] CAN specification Part A and Part B
- [9] Bluetooth SIG groups, Specification of the Bluetooth System, Ver1.1 Draft Oct 2000.
- [10] P.A. Bernstein, "Middleware: A Model for Distributed System Services", Communications of the ACM, vol. 39, pp.86-98, 1996