

## 모듈 기반의 지능형 로봇을 위한 미들웨어 구조

이광국, 최선희, 김성훈, 최형섭, 박홍성  
강원대학교 전자통신공학과

### A Middleware Architecture for Module-based Intelligent Robot

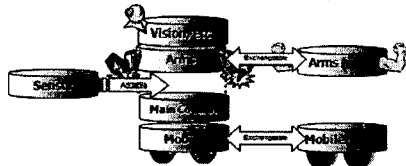
Kwang Koog Lee, Sun Hee Choe, Seong Hoon Kim, Hyeong Seob Choi, Hong Seong Park  
Kangwon National University, Dept. of Electronics and Telecommunication Engineering

**Abstract** - 현대 로봇 분야의 개발에 있어 지능형 서비스 로봇은 모듈 기반으로 구성될 수 있다. 여기서 모듈이란 로봇의 특정 태스크를 수행하기 위해 만들어진 자동화 하드웨어 단위를 의미한다. 이러한 모듈 기반의 로봇 시스템 내부는 네트워크의 이종성에 따라 다양한 네트워크들을 통해 연결될 수 있기 때문에 다중 모듈간 안정적인 통신을 하기 위해서는 상호운용을 위한 기술이 뒷받침 되어야 한다. 이를 위해 본 논문은 모듈 기반의 로봇을 위한 통신용 미들웨어를 제안한다. 제안된 미들웨어는 네트워크 인터페이스 계층과 네트워크 적응 계층으로 나누어진다. 네트워크 인터페이스 계층은 각 이종 인터페이스들을 논리적인 채널로 추상화하는 기능을 갖는다. 반면, 네트워크 적응 계층은 모듈간의 통신을 위한 메시지 형식을 정의하고 주소할당 및 이종 네트워크를 고려한 라우팅 기능들을 갖는다. 결국, 제안된 미들웨어는 두 계층을 통해 상위 로봇 어플리케이션 개발자들에게 네트워크의 투명성을 보장할 수 있으며 모듈 기반의 로봇내에서 모듈간의 안정적인 통신을 지원한다.

**Keywords** - 이종 인터페이스, 미들웨어, 모듈 기반의 로봇

## 1. 서 론

일반적으로 지능형 로봇은 인간처럼 걷고 생각하며 특정 상황에 따라 인간이 원하는 서비스를 제공하는 휴머노이드형 로봇에서부터 인간의 감정에 반응하면서 즐거움을 주기 위한 엔터테인먼트형 로봇까지 특정 목적에 따라 다양한 형태로 존재할 수 있다. 이러한 로봇들은 추구하고자 하는 목표에 보다 융통성을 제공하면서 필요한 기능들을 쉽게 확장할 수 있기 위해 모듈의 형태로 존재할 수 있다[1-2]. 여기서 모듈이란 특정한 기능들을 수행하는 목적으로 만들어진 하드웨어 단위를 말한다. 각 모듈은 해당하는 특정 목적을 수행하기 위해 USB[3], CAN[5], IEEE1394[4], RS232C, Ethernet과 같은 다양한 이종 인터페이스들을 수용할 수 있으며 로봇 내부에 존재하는 다른 모듈들과의 연결을 통해 분산된 통합 제어 구조를 이룰 수 있다. 아래의 그림 1은 모듈화된 로봇의 개념도를 보여준다.



〈그림 1〉 모듈 기반의 로봇 개념도

이종 네트워크들을 통해 연결된 분산 네트워크의 구조에서 각 모듈간에 상호인식 및 제어를 위해서는 미들웨어 기반의 상호운용 기술이 뒷받침 되어야 한다. 네트워크 프로토콜이 다른 환경에서 네트워크를 통합하는 방법은 크게 최상위 계층인 응용계층에서 연동을 필요로 하는 네트워크 인터페이스간 일대일 브릿지를 구성하는 방법과 새로운 미들웨어 계층을 통해 어플리케이션과 이종 인터페이스들 간에 독립성을 제공하는 방법으로 나눌 수 있다.

우선 이종 네트워크 프로토콜 간의 데이터 전달을 브릿지에서 처리하는 방법은 각 네트워크의 특징이 그대로 유지되므로 기존의 어플리케이션들을 그대로 사용할 수 있는 이점을 갖는다. 하지만 이 방법은 시스템이 여러 종류의 인터페이스들을 제공되는 경우 해당하는 인터페이스의 숫자만큼 브릿지가 필요하므로 비효율적이며 개발 시간이 길어지는 단점을 갖는다. 반면 새로운 미들웨어 계층을 사용하는 방법은 기존의 어플리케이션들을 그대로 사용할 수 없지만 각 이종 인터페이스들의 추상화를 통해 어플리케이션으로부터 의존성을 줄일 수 있고 모듈 기반으로 분산된 네트워크에서 네트워크 종류에 제한을 받지 않으면서 원하는 목적지 모듈까지 안정적으로 데이터를 전송할 수 있는 장점을 갖는다.

본 논문에서는 모듈 기반의 지능형 로봇 시스템에서 이종 네트워크 인터페이스 종류에 제한을 받지 않으면서 각 모듈간의 원활한 통신을 지원하기 위한 새로운 로봇용 통신 미들웨어 구조를 제안한다. 제안된 미들웨어를 통해 지능형 로봇은 이종 인터페이스들의 추상화를 통한 최상위 계층으로부터 독립성을 제공받으면서 로봇 시스템이 다양한 모듈을 통해 구성될 수 있는 확장성을 제공받을 수 있다. 또한 이종 인터페이스로 이루어진 환경에서 다양한 대역폭과 최대전송단위(MTU) 등을 라우팅 메트릭(metrics)으로

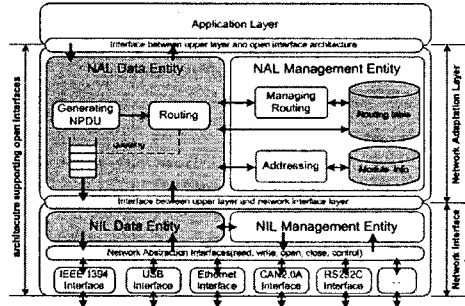
사용함으로써 이종 인터페이스로 이루어진 네트워크에 보다 효율적인 데이터 송수신을 수행할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 제안된 미들웨어의 구조와 기능들을 상세하게 설명한다. 3장에서는 제안된 미들웨어를 테스트하기 위한 실험이 소개되며 마지막 장에서 결론을 맺는다.

## 2. 이종 인터페이스를 수용하는 모듈 기반의 로봇을 위한 미들웨어

### 2.1 제안된 미들웨어 구조

제안된 미들웨어는 다양한 이종 인터페이스들을 추상화하는 네트워크 인터페이스 계층과 각 모듈과의 일반화된 통신을 수행하기 위한 네트워크 적응 계층으로 구성된다. 네트워크 인터페이스 계층의 경우 실제로 이종 인터페이스들의 제어 및 입출력 기능을 수행하기 위한 프로그래밍 인터페이스들을 논리적인 채널(Channel)을 통해 랩핑(Wrapping)함으로써 이종 네트워크를 추상화하는 기능을 제공한다. 반면, 네트워크 적응 계층의 경우 모듈 간의 원활한 통신을 수행하기 위해 일반화된 메시지 포맷을 정의하고 각 모듈의 식별을 위한 모듈의 주소할당 기능과 모듈간의 라우팅 기능을 제공한다.



〈그림 2〉 미들웨어의 구조

### 2.2.네트워크 인터페이스 계층(Network Interface Layer)

네트워크 인터페이스 계층은 여러 종류의 이종 네트워크 인터페이스들을 채널로 추상화하는 기능을 통해 최상위 계층으로부터의 독립성을 제공한다. 따라서 네트워크 인터페이스 계층은 실제로 다양한 이종 인터페이스들의 제어 및 입출력 기능을 수행하기 위한 open, close, control, write 및 read와 같은 프로그래밍 인터페이스들을 추상화된 채널의 인터페이스들로 랩핑(Wrapping)한다. 이처럼 논리적인 채널을 통해 이종 인터페이스들이 추상화되는 경우 응용 개발자들은 네트워크의 의존성이 없이 쉽게 로봇용 응용 프로그램에 적합한 상황을 따라서 동작이 가능하도록 주소는 Short Type(8비트)과 Long Type(16비트)으로 나누어 구성한다. Short Type에서 구분할 수 있는 주소는 약 240개이고 Long Type의 경우는 약 65,000개이다.

### 2.3 네트워크 적응 계층(Network Adaptation Layer)

#### 2.3.1 주소의 크기와 주소할당 방법

로봇을 위한 주소 포맷은 많은 크기를 차지하게 될 때 모듈 간 메시지 전송 시 오버헤드가 될 수 있고 크기가 작으면 확장성에서 문제가 될 수 있으므로 이를 적절히 고려한 주소할당 체계가 필요하다. 주소의 크기를 생각하면 8비트(바이트)를 갖는 경우 로봇 시스템 내부에 255개의 모듈 이상이 들어갈 수 있다는 의문에 답을 해야 한다. 그러나 16비트(2바이트)로 하게 되면 로봇 시스템에서 사용하기에는 매우 큰 주소가 된다. 따라서 로봇 시스템에 적합한 상황에 따라서 동작이 가능하도록 주소는 Short Type(8비트)과 Long Type(16비트)으로 나누어 구성한다. Short Type에서 구분할 수 있는 주소는 약 240개이고 Long Type의 경우는 약 65,000개이다.

이러한 주소의 크기를 바탕으로 다양한 이종 인터페이스로 구성된 모듈 기반의 지능형 로봇 시스템에서 각 모듈간 식별을 위한 주소할당이 이루어져야 한다. 지능형 로봇 시스템에서 모듈은 잘 알려진 모듈이 있을 수 있고 그렇지 않은 모듈이 있을 수 있다. 즉, 이미 알려진 모듈의 경우는 주소를 정적으로 지정할 필요가 있으며 그렇지 않은 경우에 대해서는 주소를 동적으로 지정할 필요가 있다.

우선 정적인 주소 할당에서 모듈들은 수행하는 특정 업무에 따라 미리 정의된 주소의 체계들을 통해 주소를 지정받을 수 있다. 하지만 지능형 로

봇을 구성하는 모듈의 주소를 이와 같은 식으로 정적으로 할당하면 각 기능에 맞게 수동적으로 주소를 지정해야 하는 번거로움을 겪어야 한다. 따라서 로봇 시스템은 각 모듈들에 대해서 주소를 자동적으로 설정할 수 있는 동적인 주소 할당 방식을 갖는다. 로봇 시스템을 구성하는 모듈들에게 전원이 인가된 이후 지정된 주소 사이즈에 따라 랜덤하게 주소를 생성하고 로봇 시스템 내부에 브로드캐스팅을 수행함으로써 해당 주소의 사용을 알리게 된다. 이때 만일 이미 로봇 시스템 내부에서 주소를 설정한 모듈이 같은 주소를 사용하고 있는 경우 중복된 주소를 갖는 모듈이 현재 주소를 사용하고 있음을 시스템 내부에 브로드캐스트 한다. 이를 수신받은 모듈은 다시 랜덤하게 주소를 생성하여 동일한 방식으로 브로드캐스팅을 수행한다. 상기 과정과 같은 주소 충돌 알고리즘을 수행하여 로봇 시스템을 구성하는 각 모듈들은 유일한 주소를 가질 수 있다.

### 2.3.2 모듈간 통신을 위한 메시지 포맷 정의

지능형 로봇 시스템이 이중 인터페이스들을 통해 다수 개의 모듈로 연결되는 경우 각 모듈간의 원활한 통신을 수행하기 위한 일반화된 메시지 포맷이 정의되어야 한다. 메시지 포맷은 표 1과 같이 헤더와 페이로드로 구성된다. 여기서 헤더는 모듈간의 통신을 위해 목적지 및 근원지의 주소가 포함되며 페이로드에는 미리 약속된 기능을 수행하기 위한 추가적인 명령 포맷이나 상위 계층으로부터 전달된 데이터가 위치한다. 제어필드는 1바이트로 구성되며 메시지 타입, 프로토콜 버전 및 목적지/근원지 모듈의 주소형에 대한 필드를 갖는다.

〈표 1〉 네트워크 적응 계층의 일반적인 메시지 포맷

1byte	1/2 bytes	1/2 bytes	1 byte	variable(가변)
제어필드	목적지 모듈주소	근원지 모듈주소	시퀀스 번호	
헤더				페이로드

상기 표 1에서 목적지 모듈의 주소는 주소형에 따른 목적지 모듈의 주소를 나타내며 근원지 모듈의 주소는 주소형에 따른 근원지 모듈의 주소를 나타낸다. 마지막으로 시퀀스 번호는 메시지에 해당하는 ID로 매 메시지 전송마다 1씩 증가하며 근원지 모듈 주소와 쌍을 이루어 메시지의 유일성을 보장하는데 쓰인다.

### 2.3.3 이중 인터페이스를 고려한 라우팅 기법

모듈이 구성되는 지능형 로봇에서 모듈간 통신을 수행하기 위해 제안된 미들웨어는 동적 라우팅 방식에서 사용되는 Distance-vector 방식을 사용한다. 따라서 각 모듈은 로봇 내부에 존재하는 이웃하는 모듈에 대한 비용(거리)을 갖는 벡터를 라우팅 테이블에 유지하고 이 정보를 추상화된 채널을 통해 이웃 모듈들에게 라우팅 테이블의 정보를 브로드캐스팅 한다. 이후 상기 메시지를 수신하는 이웃 모듈은 메시지 페이로드에 포함된 라우팅 정보를 바탕으로 자신의 라우팅 테이블을 갱신한다. 만일 테이블을 갱신하는 경우 데이터가 네트워크 인터페이스 계층의 임의의 채널로부터 수신되기 때문에 데이터가 어떤 채널로 도착했는지 알 수 있다. 따라서 라우팅 테이블에 해당 목적지 모듈의 정보와 포워딩을 하기 위한 채널 정보를 유지할 수 있다. 또한 메시지 내에 포함되어 있는 코스트 정보를 통해 해당 모듈까지의 코스트 정보를 유지할 수 있다. 라우팅 테이블을 갱신한 모듈들은 변경된 사항을 이웃 모듈에게 알리기 위해 다시 브로드캐스팅을 수행한다. 브로드캐스팅 이후 중복되는 메시지는 메시지의 유일성을 보장하는 시퀀스 번호와 근원지 모듈의 주소를 브로드캐스트 캐시로 이용하여 처리함으로써 메시지의 중복 수신을 막을 수 있도록 한다.

지능형 로봇 시스템에서는 복수 개의 이중 인터페이스들을 통해 각 모듈들이 연결될 수 있기 때문에 각 모듈간의 다중 경로가 형성될 수 있다. 따라서 라우팅을 효율적으로 수행하려면 인터페이스에 따른 대역폭과 전송 속도 등을 고려할 수 있어야 한다. 본 발명의 방법론에 따르면 라우팅 메트릭은 대역폭과 최대 전송 단위 등을 메트릭으로 사용할 수 있다. 여기서 대역폭은 연결된 링크의 수용할 수 있는 바이트 능력을 말하며 이는 현재 회선에서 사용되는 유효 대역폭을 고려할 수 있다. 다음으로 최대 전송 단위(Maximum Transmission Unit, MTU)는 해당 경로에서 보낼 수 있는 최대 바이트수를 말한다.

이렇게 고려된 라우팅 메트릭은 아래 수식 1을 통해 라우팅 코스트를 계산할 수 있다. 여기서 n은 목적지 모듈로 가기 위해 몇 개의 모듈을 통과해야하는가를 의미하며 이 정보는 라우팅 테이블의 홉 수 정보와 동일하다. 또한 라우팅 테이블은 해당 채널을 통해 거치는 인터페이스들의 모든 정보를 저장하고 있기 때문에 해당 수식을 통해 각 인터페이스에 대한 대역폭과 최대전송단위를 고려하여 라우팅 코스트를 계산할 수 있다. 여기서 Di는 수신된 메시지의 바이트 수이며 Mi는 현재 인터페이스가 갖는 최대전송단위이고 Bi는 현재 인터페이스가 갖는 대역폭을 의미한다. 마지막으로 ci는 비례계수로써 각 인터페이스의 메시지 지연에 따른 상대적인 차이를 의미한다.

$$\sum_{i=0}^n 10 \times c_i \times \{ D_i / M_i \} \times \log(10^8 / B_i)$$

〈수식 1〉 라우팅 코스트 계산 공식

결국 모듈간의 다중 경로가 형성된다 해도 각 인터페이스에 대한 라우팅 메트릭을 상기 수식 1을 통해 각 링크에 대한 코스트를 계산할 수 있기 때문에 형성된 라우팅 테이블을 통해 가장 빠르게 보낼 수 있는 라우팅 경로를 찾을 수 있게 된다.

## 3. 구현 및 실험

제안된 미들웨어는 Ethernet, CAN 2.0A, IEEE 1394, RS232C 디바이스들을 제어하기 위한 프로그래밍 인터페이스들과 함께 C++를 통해 구현되었다. 제안된 미들웨어를 테스트하기 위해 모듈간의 통신을 수행할 수 있는 간단한 어플리케이션을 제작하였다.

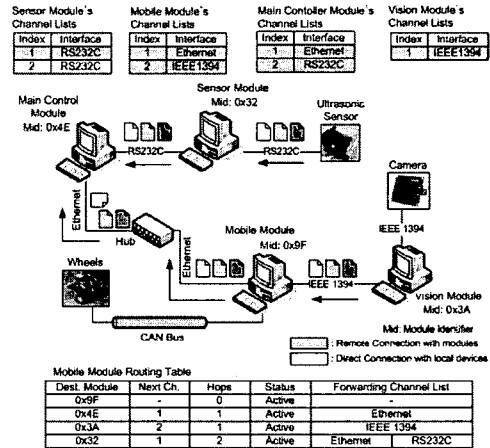
### 3.1 테스트 베드 구성

본 논문에서 제안된 미들웨어의 동작을 검증하기 위해 아래 그림 3과 같은 테스트 베드를 구성하였다. 아래 그림에서 보는 바와 같이 테스트베드 내에는 4개의 모듈들이 RS232C, IEEE1394, Ethernet, CAN 등의 인터페이스를 통해 직접 연결이 되어 있으며 SRF04 초음파 센서를 갖춘 임베디드 보드(AVR Atmega 128 MCU)가 RS232C 인터페이스를 통해 센서 모듈과 연결되어 있다. 해당 모듈들은 펜티엄-VI 1.6GHz, 512 MB RAM의 사양을 갖는 시스템의 윈도우 XP 환경에 구현이 되었다. 카메라와 휠에 관련된 어플리케이션들은 가상으로 존재한다고 가정하였다.

### 3.2 실험 시나리오

시나리오 상에서 모든 모듈들은 정적인 주소 방법을 통해 주소를 할당 받는다. 이중 인터페이스들로 연결된 모듈들간 통신을 테스트하기 위해 주소설정 이후 각 센서, 모바일, 비전 모듈은 주기적으로 센싱된 정보를 메인 컨트롤러 모듈에게 전송한다. 그림 3에서 보는 바와 같이 비전 모듈은 메인 컨트롤러 모듈과 직접 연결된 네트워크가 없기 때문에 모바일 모듈은 비전 모듈로부터 전송된 데이터를 메인 컨트롤러 모듈에게 전송하기 위해 라우터 역할을 수행해야 한다.

초음파 센서는 매 30초마다 센싱된 값을 채널 2번을 통해 센서 모듈에게 전송하였다. 이후 해당 센싱값을 전달 받은 센서 모듈은 센싱된 값이 열 번이 되는 경우 해당 정보를 채널 1번을 통해 메인 컨트롤러 모듈에게 전송하게 하였다. 모바일 모듈은 라우터로써 Distance-vector 기반의 라우팅을 통해 네트워크 내에 존재하는 모든 모듈의 정보를 그림 3과 같이 저장했다. 따라서 비전 모듈 또는 해당 모바일 모듈의 어플리케이션으로부터 내려온 데이터들을 메인 컨트롤러 모듈(0x4E)로 전송할 수 있었다. 또한 이 과정에서 라우팅 코스트를 3장에서 소개된 수식을 이용하여 유지할 수 있도록 하였다. 해당 실험을 통해 각 모듈간의 안정적인 통신을 확인할 수 있었다.



〈그림 3〉 테스트베드 및 실험 과정

## 4. 결론

본 논문에서는 모듈 기반의 지능형 로봇 시스템에서 이중 네트워크 인터페이스 종류에 제한을 받지 않으면서 각 모듈간의 원활한 통신을 지원하기 위한 새로운 로봇용 통신 미들웨어 구조를 제안했다. 제안된 미들웨어를 통해 지능형 로봇은 이중 인터페이스들의 추상화를 통한 최상위 계층으로부터 독립성을 제공받으면서 로봇 시스템이 다양한 모듈을 통해 구성될 수 있는 확장성을 제공할 수 있다. 또한 이중 인터페이스로 이루어진 환경에서 다양한 대역폭과 최대전송단위 등을 라우팅 메트릭(metrics)으로 사용함으로써 이중 인터페이스로 이루어진 네트워크에 보다 효율적인 데이터 송수신을 수행할 수 있다.

### 〈참고 문헌〉

- [1] T. Taira, N. Kamata, and N. Yamasaki, "Design and Implementation of Reconfigurable Modular Humanoid Robot Architecture", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, pp. 3566-3571, Aug. 2005.
- [2] G. Dubek and R. Sim, "RoboDaemon - A Device Independent, Network-oriented, Modular Mobile Robot Controller", IEEE International Conference on Robotics and Automation, pp. 3434-3440, Sep. 2003
- [3] Universal Serial Bus Specification revision 2.0: 27. April. 2000.
- [4] "IEEE standard for a High Performance Serial Bus", IEEE std 1394-1995, IEEE 1394 std 1394a-2000.
- [5] CAN specification Part A and Part B.