

사이버 물리 시스템을 위한 미들웨어 구조 및 제어 메시지 설계

이 형 수^o 김 성 조

중앙대학교 컴퓨터공학과

moriya@cs.cau.ac.kr, sjkim@cau.ac.kr

Middleware Structure and Control Message Design for Cyber Physical System

Hyungsoo Lee^o Sungjo Kim

School of Computer Science and Engineering, Chung-Ang University

요 약

현재 사이버 물리 시스템을 통해 IT와 교통, 항공, 의료, 화학 등 많은 분야들이 융합되어 Cyber Element와 Physical Element가 상호 작용하는 대규모 시스템 구축에 많은 연구가 이루어지고 있다. 하지만 정적인 제어 구조를 가진 기존 시스템들의 통합으로 인해 새롭게 구축되는 대규모 시스템은 복잡도가 크게 증가하여 제어 구조에 관한 명세가 새롭게 이루어져야 하는 문제가 발생하고 있으며 해결하기 위해 사이버 물리 시스템에서는 일시적인 동적 제어가 가능한 미들웨어에 대한 요구가 증가하고 있다. 이에 본 논문에서는 사이버 물리 시스템을 위한 장치 간 동적 제어를 제공할 수 있는 미들웨어를 개발하기 위하여 하위 계층에서는 Star형, 상위 계층에서는 Flat한 계층 구조를 가진 하이브리드 계층 구조를 제안한다. 또한 미들웨어 구성요소들에 대한 특성 및 기능을 정의하고, XML(eXtensible Markup Language)을 이용하여 다양한 상황과 각 구성요소들의 기능에 따라 확장성과 유연성 있는 제어 메시지를 설계한다.

1. 서론¹²

사이버 물리 시스템(Cyber Physical System)[1][2][3]은 새롭게 떠오르는 공학 분야 중 하나이다. 하지만 사이버 물리 시스템은 완전히 새로운 분야가 아닌 내장형 시스템(Embedded System)의 물리적 특성이 반영되고, 네트워크로 연결된 시스템 간의 제어의 중요성이 강조되는 시스템이다. IT와 교통, 항공, 의료, 화학 등 실제로 존재하는 많은 분야들과 활발히 융합(Convergence)되고 있는 추세에 있는 것도 사이버 물리 시스템의 등장과 같은 맥락으로 볼 수 있다.

이러한 배경으로 인하여 현재 걸음마 단계에 있는 사이버 물리 시스템에서는 서로 다른 분야의 각 물리장치들 간에 일시적인 동적 제어를 위하여 미들웨어를 개발할 필요성이 대두되고 있다. 따라서 본 논문에서는 물리 장치들 간에 일시적인 동적 제어를 제공할 수 있는

구조를 개발하기 위하여 하위 계층에서는 Star형, 상위 계층에서는 Flat한 계층구조를 가진 Controller간 서로 평등한 계층구조를 유지하는 하이브리드 계층구조를 제안한다. 그리고 C&A, Controller, Actuator, Sensor로 구성되어 있는 미들웨어 구성요소들에 대한 특성 및 기능을 정의하고, XML[4](eXtensible Markup Language)을 이용하여 다양한 상황과 각 구성요소들의 기능에 따라 요구되는 제어 메시지를 설계한다. 또한 대규모 네트워크 구성 시 현재 인터넷을 통해 가장 안정적이라고 검증이 된 TCP/IP를 사용하여 네트워크를 구축한다. 하지만 IPv4는 현재 IP 주소가 포화상태에 도달하여 향후 필연적으로 IPv6[5] 주소 체계가 사용될 것이므로, 본 논문에서는 대규모 네트워크를 가정하고 IPv6를 이용하여 네트워크를 구성한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로서 사이버 물리 시스템과, IPv6에 대해 간단히 살펴 본 후, 3장에서는 제안하는 동적 제어 미들웨어의 구조 및 미들웨어의 구성요소에 관하여 설명하고 4장에서는 장치간 제어를 위한 제어 메시지를 설명하며 마지막으로 5장에서 결론 및 향후 연구과제에 대해서 기술한다.

¹ 본 연구는 서울시 산학연 협력사업(CR070019) 지원으로 수행되었음

² 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터(홈네트워크연구센터) 육성, 지원사업의 연구결과로 수행되었음

2. 관련 연구

2.1. 사이버 물리 시스템

최근 떠오르고 있는 분야인 사이버 물리 시스템은, 소프트웨어의 도움으로 시스템 간의 통신 및 제어를 담당하는 Cyber Element 와 로봇, 차량, 항공기, 의료기기, 도로망 등의 실 세계의 Physical Element로 나뉘며, 대규모의 시스템을 구축하기 위해서는 Cyber Element와 Physical Element들의 상호 작용이 필요하다. 하지만 실 세계의 Physical Element들은 서로 다른 목적으로 설계되어 정적인 제어구조를 가지고 있기 때문에, 각각의 이질성이 존재하는 시스템들의 통합은 새롭게 구축되는 대규모 시스템의 복잡도가 크게 증가 하게 만드는 원인이 된다. 따라서 제어 구조에 관한 명세가 새롭게 이루어져야 하는 문제가 발생하고 있으며, 이를 해결하기 위해 사이버 물리 시스템에서는 시스템간의 일시적인 동적 제어가 가능한 미들웨어에 대한 요구가 증가하고 있다.

이에 따라 현재 사이버 물리 시스템은 연구 초기 단계에 있는 만큼 대규모 망에서 동적 제어 미들웨어에 대한 표준이 필요한 실정이다.

2.2. IPv6

현재 사용되고 있는 32bit의 IPv4의 무분별한 클래스 단위의 할당으로 인하여 주소공간 부족이 야기 되었으며 통계 자료[6]에 따르면 2011년 9월에 모든 주소가 소진 될 것으로 예상 된다. 이로 인해 지속적인 인터넷 발전에 문제가 예상되어 IPv4의 대안으로 IPv6 프로토콜을 제정하였다.

IPv6의 특징은 다음과 같다. 첫째, IP주소의 확장을 통해 128bit의 주소공간을 갖게 되었다. 둘째, 호스트 주소 자동 설정으로 인하여 네트워크에 접속하는 순간 자동적으로 네트워크 주소를 부여 받는다. 셋째, 패킷 크기의 제한을 없어서 대역폭이 넓은 네트워크를 더욱 효율적으로 사용할 수 있게 되었다. 넷째 네트워크의 물리적 위치에 제한받지 않고 같은 주소를 유지하면서 자유롭게 이동할 수 있는 이동성이 있다.

3. 동적 제어 미들웨어의 구조 및 구성요소

3.1. 동적 제어 미들웨어의 구조

본 논문에서 제안하는 동적 제어 미들웨어 구조는 (그림 2)와 같다. 동적 제어 미들웨어는 C&A(Controller & Administrator), Controller, Actuator, Sensor로 구성되어 있다. 본 논문에서 목적으로 하는 동적인 제어가 가능한 시스템을 구축하기 위해서는 모든 Controller가 서로의 정보를 공유해야 한다. 하지만 Controller들의 정보를 공유하게 될 경우 Controller의 개수가 증가함에 따라 저장해야 할 정보의 양은 산술급수적으로 증가하게 되며

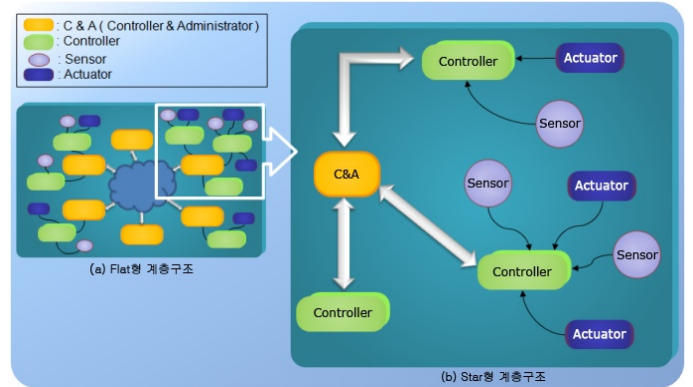


그림 1. 동적 제어 미들웨어 구조

현실적으로 모든 Controller들의 정보를 각각의 Controller들이 보유하는 것은 불가능해진다. 이에 정보를 분산시켜 저장하기 위한 계층 구조가 필수적으로 요구된다. 하지만 전체 Controller에 대해 계층 구조를 유지할 시 기존 통합 미들웨어와 같은 중앙 집중형 계층 구조를 따르면 중앙 Controller에 과부하가 예상되며 사이버 물리 시스템에 필요한 동적인 Controller의 추가 및 삭제가 용이하지 않다는 단점을 가진다. 따라서 본 논문에서 제안한 미들웨어가 적용된 사이버 물리 시스템의 구조는 하위계층에서는 Star형의 계층구조를 유지 하면서 상위계층에서는 Flat하며 Controller간 서로 평등한 계층구조를 유지하는 하이브리드 계층구조를 가진다.

3.1.1. Star형 계층구조

미들웨어 구조 중 하위에서 Star형(그림 2의 (b) 참조)의 계층구조를 갖게 되는 것은 Controller간 제어 요청 시 하위 계층에서 가장 활발하게 요청이 이뤄질 것이기 때문이다. 이에 따른 장점은 Star형 계층구조를 유지할 시 하위 계층에서 2 홉(Hop) 안에 모든 Controller간 통신이 가능하며 사이버 물리 시스템에서 요구되는 Controller의 동적인 추가 및 제거 과정이 단순하다. 또한 하위 Star형 구조에서 최상위 계층에 하위 Controller들의 정보를 보유하는 C&A를 위치시킴으로써 모든 Controller가 서로의 정보를 공유하는 오버헤드를 제거 하며 하위 Star형 계층구조에서 전체적인 메모리 효율을 높인다.

3.1.2. Flat형 계층구조

미들웨어 구조 중 상위에서는 각 하위 계층구조에서 최상위에 위치한 C&A들이 Flat하며 평등한 구조를 가진다. (그림 2의 (a) 참조) 하지만 상위에서 C&A간 정보를 공유하게 되면 모든 Controller의 정보를 각각의 C&A가 가지고 있어야 한다. 그러나 이는 현실적으로 불가능하므로 C&A는 하위 Controller의 정보만을 유지하며 C&A간 정보를 공유하지 않는다. 그렇기 때문에 같은 C&A

에 등록된 Controller간 또는 서로 다른 C&A에 등록된 Controller간 제어를 요청할 시에는 Controller IP, Actuator name, Function name, Sensor name을 검색 조건으로 검색 하여 하위 Star형 계층구조와 상위 Flat형 계층구조에서 제어해야 할 Controller를 확인하는 선작업을 수행함으로써 C&A간 정보를 공유해야 하는 오버헤드를 제거한다.

3.2. 동적 제어 미들웨어의 구성요소

본 논문에서 제안하는 미들웨어의 구성요소는 C&A, Controller, Actuator, Sensor로 구성된다.

Controller는 Actuator 그리고 Sensor와 연결되어 있으며, Actuator를 제어하고 Sensor로부터 데이터를 전달 받는다. 본 논문에서 제안하는 동적 제어 미들웨어 구성요소 중 Controller가 보유한 데이터는 [표 1]과 같다.

C&A(Controller & Administrator)는 Controller간 정보 공유의 오버헤드를 감소시키기 위하여 존재하고, 하위 네트워크에 속한 Controller를 관리하는 역할을 수행하며, C&A간은 서로 Flat하며 평등한 관계를 가진다. C&A는 Controller를 관리하기 위하여 자신의 하위 Controller들의 정보를 가진다. 또한, C&A는 일반 Controller와 동일한 장치이므로 어떠한 Controller들도 C&A가 될 수 있다. 따라서 하위 네트워크 상에서 새로운 Controller가 출현하였을 때, 기존의 C&A와 새로운 Controller와의 하드웨어 성능 비교를 통하여 새로운 Controller가 성능이 우수할 시 기존의 C&A와 교체될 수 있다. 본 논문에서 제안하는 동적 제어 미들웨어 구성요소 중 C&A가 보유한 데이터는 [표 2]와 같다.

표 1. Controller가 보유한 데이터

Data	설명
IP	IPv6기반의 Controller IP 주소
Parent C&A IP	자신을 관리하는 상위 C&A의 IP주소
H/W Information	하드웨어 정보
AID	보유하고 있는 Actuator의 식별 ID
* Actuator Name	Actuator의 이름
* Function Name	Actuator의 기능에 대한 이름
- Attribute Data	Actuator의 속성 데이터
- State Data	Actuator의 상태 데이터
* Actuator Usability	Actuator의 제어 가능 여부 식별 속성
SID	보유하고 있는 Sensor의 식별 ID
* Sensor Name	센싱하는 데이터 속성의 이름
* Sensing Data	Sensor로부터 전송 받는 센싱 데이터
Role of Controller	C&A, Controller 역할 구분

표 2. C&A가 보유한 데이터

Data	설명
Basic Information of controller	C&A도 Controller이므로 Controller의 기본정보들을 보유
Child Controller IP	하위 Controller들의 IP 주소
AID	하위 Controller의 Actuator 식별 ID
* Actuator Name	하위 Controller의 Actuator 이름
* Function Name	하위 Controller의 Actuator 기능

표 3. Actuator가 보유한 데이터

Data	설명
AID	Actuator의 식별 ID
Actuator Name	Actuator의 이름
Function Name	기능에 대한 이름
* Attribute Data	기능의 속성 데이터
* State Data	현재 상태 데이터

표 4. Sensor가 보유한 데이터

Data	설명
SID	Sensor의 식별 ID
Sensor Name	센싱하는 데이터 속성의 이름
Data	물리 세상을 센싱한 데이터

Actuator는 물리 세상에 실제로 물리력을 능동적으로 행사한다. Actuator의 특성에 따라 다양한 기능 및 속성을 가지며 Controller를 통해 제어된다. 본 논문에서 제안하는 동적 제어 미들웨어 구성요소 중 Actuator가 보유한 데이터는 [표 3]와 같다.

Sensor는 물리 세상의 정보를 센싱하는 역할을 담당한다. 센싱한 데이터를 Controller에게 전달함으로써 Controller는 물리 세상에 대한 정보를 얻을 수 있다. 본 논문에서 제안하는 동적 제어 미들웨어 구성요소 중 Sensor가 보유한 데이터는 [표 4]와 같다.

4. 장치 간 제어를 위한 제어 메시지 설계

본 논문에서 제안한 제어 메시지는 XML(eXtensible Markup Language)로 설계되었다. XML은 데이터 구조를 기술하는 유연한 기법이다. 이를 통해 이질적 미들웨어 간 동적 통합 기술 개발에서 서로 다른 형식으로 정의되어 있는 이질적 미들웨어의 제어 메시지 변환 엔진을 개발할 때 확장성과 유연성을 제공한다.

일반적으로 XML문서는 Element, Attribute, Entity, DTD 등 4개의 주요 구성 요소를 포함한다. Element는 태그 데이터를 표현하며, Attribute는 Element에 대한 정보를 구체화하기 위해 사용된다. Entity는 XML문서에서 참조할 수 있는 데이터의 실질적 값이다. DTD는 개개의 XML 문서의 구조를 정의하며 선택적인 부분이다.

표 5. 제어 메시지 구조

```

1 <Command>Command</Command>
2 <RequestIP>From_Controller_IP</RequestIP>
3 <Information>
4   <NController>Number_of_Controller</NController>
5   <Controller>
6     <IP>Controller_IP</IP>
7     <Name>Controller_Name</Name>
8     <NHardware>Number_of_Hardware</NHardware>
9     <Hardware>
10      <Name>Hardware_Name</Name>
11      <Value>Hardware_Value</Value>
12    </Hardware>
13    <NActuator>Number_of_Actuator</NActuator>
14    <Actuator>
15      <AID>Actuator_ID</AID>
16      <Name>Actuator_Name</Name>
17      <NFunction>Number_of_Function</NFunction>
18      <Function>
19        <Name>Actuator_Function_Name</Name>
20        <Type>Function_Attribute_Type</Type>
21        <Attribute>Function_Attribute_Value</Attribute>
22        <Selected>Attribute_Value_State</Selected>
23      </Function>
24    </Actuator>
25    <NSensor>Number_of_Sensor</NSensor>
26    <Sensor>
27      <SID>Sensor_ID</SID>
28      <Name>Sensor_Name</Name>
29      <Value>Sensor_Value</Value>
30    </Sensor>
31  </Controller>
32 </Information>
    
```

[표 5]에서는 제안한 제어 메시지의 구조가 3가지 Element(Command, Request IP, Information)로 구성됨을 보여주고 있다. Command 부분에서는 제어 메시지가 수행하는 명령을 구분한다. Request IP 부분에서는 제어 메시지를 보낸 Controller의 IP 주소를 추가하여 패킷에서 송신자의 IP 주소를 추출하는 과정을 생략함으로써 IP 주소를 추출하는 과정을 단순화하였다. 그리고 마지막 Information 부분에서는 실제적으로 제어에 활용되는 정보들을 삽입한다. 또한 설계된 제어 메시지의 Attribute를 통하여 Command, Request IP, Information 등 3개의 Element에 대한 정보를 구체화한다. Controller에서는 XML 파싱 작업을 통하여 분석된 태그의 의미와 구조에 기초하여 Element, Attribute, Entity, DTD 등 4개의 구성 요소를 구분하여 Information에서 정보들을 추출한 후 실제적으로 제어를

수행한다.

4.1. Controller의 동적 등록 및 C&A 교체

Controller가 새로운 네트워크에 진입한 후 등록을 할 때 우선 IPv6의 로컬 멀티캐스트를 이용하여 인접한 C&A를 찾는다. [C&A Search Request, 표 6의 1] 만약 C&A가 존재하지 않는다면, 자신 스스로가 C&A가 되어 Star형 구조를 형성하게 된다. C&A가 존재할 경우에는 그 Star형 구조에 등록 되는데, 이때 기존의 C&A와 중앙처리 장치 및 가용메모리 등을 기준으로 C&A 처리시간과 네트워크를 구성 가능한 Controller의 개수 비교를 통해 C&A를 교체[C&A Replace, 표 6의 2, 3] 또는 유지[Controller Insert, 표 6의 4, 5]한다. 교체가 이루어진 경우, 기존의 Star형 계층 구조의 Controller 정보들을 이전의 C&A에게 상속[C&A Replace & Information Report, 표 6의 6]받게 된다. 그리고 각 Controller들에게는 새로운 C&A의 IP를 전송[New C&A IP Report, 표 6의 7]하게 된다.

4.2. Controller의 일시적인 동적 제어

Controller들은 정적인 관계를 유지하지 않고 언제든지 상대방으로부터 정보를 얻어오거나, 제어를 하는 측(Controller)과 제어를 받는 측(Controllee)의 역할이 역전될 수 있어야 한다. 따라서, 제안하는 미들웨어의 Controller는 다른 장치를 제어할 때, 다른 모든 Controller의 정보를 가지지 않고 자신의 상위 C&A 정보만 가지면서, 제어할 Controller의 정보를 C&A의 멀티캐스팅 검색[Star Search, Flat Search, 표 6의 9, 11]을 통해 얻는다[Search Response, 표 6의 10]. 멀티캐스팅을 할 때 미리 정해진 멀티캐스트 그룹ID를 이용한다. 그 정보를 바탕으로 제어가능 여부[Actuator Control, 표 6의 12, 13]를 확인한 후, 제어가 가능하다면 Controllee를 일시적으로 제어[Controller Command, 표 6의 14, 15]한다. 이때, Controllee는 Actuator에게 동작을 요청하고, 이에 대한 응답을 받는다. [Actuator Command, 표 6의 24, 25]

4.3. 미들웨어 구성요소의 등록 및 해제

4.3.1. C&A의 해제

C&A의 해제에는 정상 종료로 통한 해제와 비정상 종료를 통한 해제가 있다. 먼저 정상 종료로 통한 해제의 경우 C&A는 Star형 구조 내의 Controller들에게 하드웨어 정보를 요구하여 하드웨어 정보들을 수집한 후 [Hardware Information, 표 6의 16, 17] 하드웨어의 성능을 비교하여 가장 고사양의 Controller에게 C&A의 권한을 주게 된다. [C&A Replace, 표 6의 2, 3] 교체가 이루어진 후, Star형 계층 구조의 Controller 정보들을 새로운

C&A에게 상속[C&A Replace & Information Report, 표 6의 6]한다. 그리고 각 Controller들에게는 새로운 C&A의 IP를 전송[New C&A IP Report, 표 6의 7]하게 된다. 비정상 종료로 통한 C&A 해제의 경우 가장 먼저 C&A의 해제를 발견한 Controller가 같은 Star형 계층에 있던 Controller들에게 하드웨어 정보를 요구하여 하드웨어 정보들을 수집한 후[Hardware Information, 표 6의 16, 17] 하드웨어의 성능을 비교하여 가장 고사양의 Controller에게 새로운 C&A의 권한을 준다. [C&A Replace, 표 6의 2, 3] 그런 후, Star형 계층구조 내의 Controller들에게 새로운 C&A의 IP를 전송[New C&A IP Report, 표 6의 7]하면서 각 Controller들의 정보를 얻기 위해 각 컨트롤러들에게 정보를 요청[Information Request, 표 8의 8]하는 메시지를 함께 보낸다.

4.3.2. Controller의 해제

Controller는 정상 종료 시 C&A에게 먼저 알린 후 종료를 하여 C&A가 정상적인 데이터를 가질 수 있도록 한다. [Controller Delete Request, 표 6의 18] 비정상 종료일 경우 C&A에는 값이 갱신되지 않아 다른 Controller가 잘못된 정보를 검색할 수도 있다. 그럴 경우 C&A에게 확인해 달라는 메시지를 보내 Controller가 정상적으로 존재하는지 확인하여 그 결과를 요청한 Controller에게 응답한다. [Controller Check, 표 6의 19, 20] 이때 C&A는 존재하지 않는 Controller의 정보를 삭제한다.

표 6. Command 리스트

No	Command	내용
1	C&A Search Request	Controller가 등록을 할 때 우선 IPv6의 로컬 멀티캐스트를 이용하여 인접한 C&A를 찾는다. 만약 C&A가 존재하지 않는다면, C&A를 찾은 Controller가 스스로 C&A가 되어 Star형 구조를 형성하게 된다. C&A가 존재할 경우에는 그 Star형 구조에 등록 되는데, 이때 기존의 C&A와 중앙 처리장치 및 가용메모리 등을 기준으로 C&A 처리시간과 네트워크를 구성 가능한 Controller의 개수 비교를 통해 C&A를 유지 또는 교체한다. 교체가 이루어진 경우, 기존의 Star형 계층 구조의 Controller 정보들을 이전의 C&A에게 상속받게 된다. 그리고 각 Controller들에게는 새로운 C&A의 IP를 전송하게 된다.
2	C&A Replace Request	
3	C&A Replace Response	
4	Controller Insert Request	
5	Controller Insert Response	
6	C&A Replace & Information Report	
7	New C&A IP Report	새로운 C&A에게 Controller들의 정보 전송을 요청한다. (C&A 비정상 종료 후 교체 시)
8	Information Request	
9	Star Search Request	
10	Search Response	하위 구조인 Star형 구조 내에서의 Search와 상위 구조인 Flat형 구조에서의 Search를 통해 해당되는 정보를 응답 해준다.
11	Flat Search Request	
12	Actuator Control Request	
13	Actuator Control Response	Controller A가 Controller B의 Actuator를 제어하기 위해 B에게 Actuator의 제어권을 요구한 후 A에게 제어 허용 여부를 알려준다. (제어 가능 시 Actuator의 상태 정보 전송)
14	Controller Command Request	
15	Controller Command Response	
16	Hardware Information Request	모든 하위 Controller들에게 하드웨어 정보를 요구한 후 하위 Controller들이 하드웨어 정보를 전송하는 응답한다.
17	Hardware Information Response	
18	Controller Delete Request	Controller가 상위 C&A에게 해제를 요구한다. (Controller의 정상 종료 시)
19	Controller Check Request	
20	Controller Check Response	
21	Actuator Insert Request	Actuator가 Controller에게 자신의 등록을 요청한 후 등록 요청에 대한 응답을 한다.
22	Actuator Insert Response	
23	Actuator Delete Request	Actuator 해제를 요청한다.
24	Actuator Command Request	
25	Actuator Command Response	
26	Sensor Insert Request	Sensor가 Controller에게 자신의 등록을 요청한 후 등록 요청에 대한 응답을 한다.
27	Sensor Insert Response	
28	Sensor State Request	Controller가 Sensor에게 상태 정보를 요구한 후 상태 정보를 전송 한다.
29	Sensor State Response	

4.3.3. Actuator의 등록 및 해제

Actuator는 Controller에게 자신의 등록을 요청을 한 후, Controller에서 등록 응답을 하면 등록이 된다. [Actuator Insert, 표 6의 21, 22] 해제는 Controller가 Actuator에게 메시지를 보내 해제 한다. [Actuator Delete Request, 표 6의 23]

4.3.4. Sensor의 등록

Sensor는 Controller에게 자신의 등록을 요청을 한 후, Controller에서 등록 응답을 하면 등록이 된다. [Sensor Insert, 표 6의 26, 27]

4.4. 제어 메시지의 확장

본 논문의 제어메시지는 XML로 설계 되었다. 따라서 XML의 장점인 확장성 및 유연성을 활용하여 위에서 언급한 29개의 Command 외에도 추가적으로 필요에 따라 확장이 용이하다.

5. 결론 및 향후 과제

본 논문에서 우리는 사이버 물리 시스템이 완전히 새로운 분야가 아닌, 내장형 시스템의 물리적 특성과, 네트워크로 연결된 시스템 간의 제어의 중요성이 강조되는 시스템이라는 것을 알게 되었다. 또한 현재 초기 단계에 있는 사이버 물리 시스템을 위한 동적 제어 미들웨어 기술 개발의 필요성을 알게 되었으며 이를 위해 동적인 제어를 지원하기 위한 새로운 미들웨어 구조를 제안하였다. 새로운 미들웨어 구조는 중앙 집중형 구조가 아닌 하위 계층에서 Star형 계층구조를 유지하고 상위계층에서는 Flat하며 Controller간 서로 평등한 하이브리드 계층 구조이며 이를 통해 Controller간의 정보를 공유하는 오버헤드를 제거함으로써 메모리 효율을 높일 수 있다. 또한, XML을 이용해 각 노드들 사이에서 전달되는 제어 메시지를 설계함으로써, 확장성 및 유연성을 제공함으로써, 기존의 메시지 구조에 대한 변경 없이 현재 설계된 메시지 구조가 사용될 수 있다는 장점을 갖는다. 그러므로 본 논문에서 제안된 미들웨어 구조를 통해 대규모 사이버 물리 시스템을 구축할 때, 각 노드들에 필요한 자원을 줄일 수 있으며, 기존의 시스템을 크게 변경하지 않고 통합 시스템을 구축하는 것이 가능하다.

향후 제안한 구조를 이용하여 차량간 통신, 홈 네트워크와 같은 이질적인 미들웨어간 제어 관계가 동적으로 변경될 수 있도록 미들웨어 통합 구조를 설계하고, 구성요소들 간의 관계가 동적 특성을 갖도록 각 구성요소의 역할과 동작 방식 등을 명세화 하며, 다른 미들웨어의 데이터 전달, 명령 수행 등의 제어 메시지를 개발된 미

들웨어의 제어 메시지로 변환하는 엔진의 개발에 대한 연구가 필요하다.

참고 문헌

- [1] E. A. Lee., "Cyber Physical Systems: Design Challenges", University of California, Berkeley Technical Report No. UCB/EECS-2008-8, 2008.
- [2] E. A. Lee., "Cyber-Physical Systems - Are Computing Foundations Adequate?", Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Vol. 1, pp. 1-9, 2006.
- [3] E. A. Lee., "Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report", Technical Report No. UCB/EECS-2007-72, 2007
- [4] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler., "Extensible markup language (XML) 1.0 (second edition) - W3C recommendation", 2000.
- [5] R. Hinden and S. Deering., "IP Version 6 Addressing Architecture." RFC 4291, 2006.
- [6] 한국인터넷진흥원 IPv6, "http://www.vsix.net/"
- [7] Deering, S., Fenner, W., Haberman, B., "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, 1999.
- [8] Hinden. R., Deering. S., "IPv6 Multicast Address Assignments", RFC 2375, 1998.