



조명 제어 네트워크에서 디바이스 관리를 위한 표준 프로토콜 기술 동향

이상헌 · 최상일 · 고석주 <경북대학교 컴퓨터학부>
김인수 · 임상규 · 강태규 <한국전자통신연구원 LED통신연구팀>

1 서론

최근 LED 조명산업의 활성화에 더불어 수 많은 조명 장치로 구성되는 조명 제어 네트워크를 효율적으로 구현하고 관리하기 위한 기술이 주목을 받고 있다 [1-2]. 하지만, 기존 연구들은 대부분 과거에 개발된 DMX512-A 네트워킹[3] 기술을 토대로 진행되고 있다.

한편 조명 관련 국제 표준기구인 Professional Lighting And Sound Association(PLASA)[4]에서는 조명 제어 네트워크에서 조명 디바이스(device) 관리를 위한 기술로서 Remote Device Management(RDM)[5] 및 Device Management Protocol(DMP)[6] 프로토콜을 개발하였다.

RDM 프로토콜의 기반이 되는 DMX512-A는 조명의 점등제어를 위해 개발된 네트워킹 기술이다. 하지만 조명 디바이스의 기능이 다양해지고 복잡해짐에 따라 PLASA에서는 조명 제어기(controller)를 통해 조명 디바이스의 점등 외 기능을 효율적으로 관리하기 위해 RDM 프로토콜을 개발하였다. RDM은 조명 제어 네트워크에 위

치한 디바이스들을 탐색하고 각 디바이스들을 구성, 감시, 관리하는 기능을 제공한다. 한편 최근에는 엔터테인먼트 산업에서 요구되는 다양한 영상/음향 정보들을 Internet Protocol(IP) 기반의 네트워크를 통해 전송하기 위해 Architecture for Control Networks(ACN)[6] 구조를 제안하였다. 아울러 ACN 구조상에서 동작하는 DMP 프로토콜을 개발하여 수 많은 조명 디바이스를 인터넷 기반으로 관리 및 제어하는 기능을 제공하려고 노력하고 있다.

이에 본 논문에서는 PLASA에서 개발되고 있는 RDM 및 DMP 표준 프로토콜 기술에 대하여 살펴보고 각각의 특징을 비교 분석함으로써 향후 관련 연구에 참고정보를 제공하고자 한다.

본 논문은 다음과 같이 구성된다. 먼저 2절에서는 RDM 프로토콜의 동작절차 및 메시지 형식에 대하여 알아보고, 3절에서는 최근에 개발된 IP 네트워크를 기반으로 하는 DMP 프로토콜에 대하여 기술한다. 4절에서는 두 가지 프로토콜의 특징을 간략하게 비교 정리한다. 끝으로 5절에서는 향후 연구 이슈와 함께 본 논문의 결론을 맺는다.

2. RDM 프로토콜

2.1 개요

RDM은 기존 DMX512-A 기반의 조명 제어 네트워크에서 디바이스들을 원격으로 관리하기 위해 개발된 프로토콜이다.

그림 1은 RDM 프로토콜의 동작 개요를 보여준다. 그림에서 DMX512-A를 사용하는 제어기와 디바이스간에 요청(request) 메시지와 응답(response) 메시지 교환을 통해 디바이스 관리 기능을 수행하고 있다. 이러한 메시지 교환을 통해 제어기는 네트워크에 위치한 디바이스의 존재 여부를 탐색(discovery)하고, 감시(monitoring) 및 관리 기능을 통해 각 디바이스가 사용하는 파라미터 정보를 조회(get)하거나 설정(set)할 수 있다.

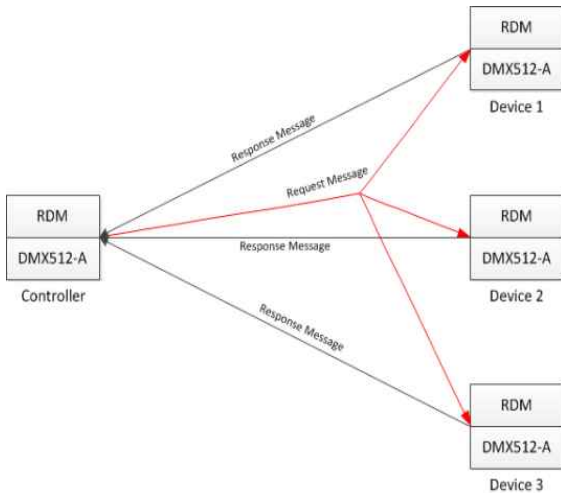


그림 1. RDM 프로토콜 동작 개요

표 1은 디바이스의 탐색 및 감시/관리를 위해 사용되는 RDM 메시지를 정리하고 있다. Discovery 메시지는 맨 처음에 제어기가 디바이스의 존재 여부를 탐색하기 위해 사용되는 메시지이고, Get과 Set 메시지는 제어기가 각 디바이스가 사용하는 조명 제

어 관련 파라미터 값을 조회하거나 설정하는 경우에 사용된다.

한편 3가지 종류의 Command 메시지는 제어기에서 디바이스로 전송하며, Command_Response 메시지는 디바이스에서 제어기로 전송한다. 메시지의 전송 방식을 살펴보면, Discovery_Command 메시지는 브로드캐스트 방식으로 전송되는 반면에, 나머지 5개의 메시지는 특정 디바이스 혹은 제어기에게만 전송된다.

표 1. RDM 메시지 종류

메시지 종류	From	To
Discovery_Command	제어기	디바이스
Discovery_Command_Response	디바이스	제어기
Get_Command	제어기	디바이스
Get_Command_Response	디바이스	제어기
Set_Command	제어기	디바이스
Set_Command_Response	디바이스	제어기

2.2 프로토콜 동작

RDM 프로토콜의 동작은 크게 탐색 과정과 감시/제어 동작으로 구분해 볼 수 있다.

A. 디바이스 탐색 절차

초기 단계에서 조명 제어기는 DMX512-A 네트워크에 있는 디바이스들을 탐색해야 한다. 이를 위해 다른 알고리즘을 사용할 수도 있지만, RDM 규격[3]에서는 이진 탐색 트리(Binary Search Tree) 기법을 사용한다.

디바이스의 탐색을 위해서는 Discovery_Command 메시지와 Discovery_Command_Response 메시지가 사용되는데, 이 중 제어기가 전송하는 Discovery_Command 메시지는 탐색 알고리즘 관

점에서 다음 3가지 유형으로 세분화 된다.

표 2. Discovery_Command 메시지 유형

메시지 유형	비고
Disc_Unique_Branch	이진 트리의 탐색 범위 지정
Disc_Mute	Command 메시지에 대한 응답
Disc_Un_Mute	여부 지정(Mute or Un_Mute)

Disc_Unique_Branch 메시지는 제어기가 디바이스들의 탐색 범위를 지정하는데 사용이 된다. 제어기는 브로드캐스트 방식으로 해당 메시지를 전송한다.

Disc_Mute 메시지와 Disc_Un_Mute 메시지는 디바이스 탐색 시 디바이스의 상태를 변경시켜 주는 메시지이다. 모든 디바이스는 초기에 Un-Mute 상태로 설정되어 있어서, 제어기가 보내는 Command 메시지에 응답을 해야 한다. 한편 제어기에 의해서 탐색이 완료된 디바이스는 더 이상 Discovery_Command 메시지에 응답을 하지 않도록 설정하기 위해, 제어기는 Disc_Mute 메시지를 해당 디바이스에 전송하여 디바이스의 상태를 Mute 상태로 변경시킨다.

상기한 메시지를 토대로 RDM 프로토콜에서 동작하는 디바이스 탐색 과정을 제어기 관점에서 정리하면 그림 2와 같다.

초기 상태에서 모든 디바이스들은 Un-Mute 상태에서 시작되며, 탐색이 완료된 디바이스는 Mute 상태로 변경하면서 탐색 과정을 진행한다. RDM 네트워크에서 각 디바이스는 고유한 Unique ID(UID)를 가지고 있으며, 이를 토대로 수행되는 탐색 과정은 다음과 같다.

시작 단계에서 탐색을 위한 디바이스의 UID 범위를 전체로 지정한다.

UID 탐색 범위를 lower와 upper로 지정한다.

lower와 upper가 같은지 확인한다. 같으면 해당 UID를 가지는 디바이스가 탐색되었음을 의미한다.

lower와 upper가 같지 않으면 해당 UID 범위로 Disc_Unique_Branch 메시지를 전송한다.

디바이스로부터 Disc_Unique_Branch 메시지에 대한 응답 메시지가 오는지 확인한다.

응답이 여러 디바이스로부터 오면 범위를 (lower, mid), (mid+1, upper)로 UID 범위를 세분화하여 재지정 하고 단계 2)로 간다. 이때 나누어진 범위를 대상으로 각각 탐색 절차가 수행된다.

해당 디바이스가 탐색된 경우, 제어기는 Disc_Mute 메시지를 해당 디바이스에게 보낸다.

해당 디바이스의 탐색이 종료된다.

제어기는 상기의 방식을 RDM 네트워크에 위치하고 있는 모든 디바이스가 탐색될 때까지 수행한다. 서버 디바이스 (sub-device) 혹은 프락시(proxy) 디바이스가 존재하는 경우에는 별도로 추가적인 절차가 요구된다[3].

B. 디바이스 감시 및 제어

각 디바이스에 대한 탐색을 통해 네트워크를 구성한 후 디바이스 파라미터를 조회 및 설정하는 기능을

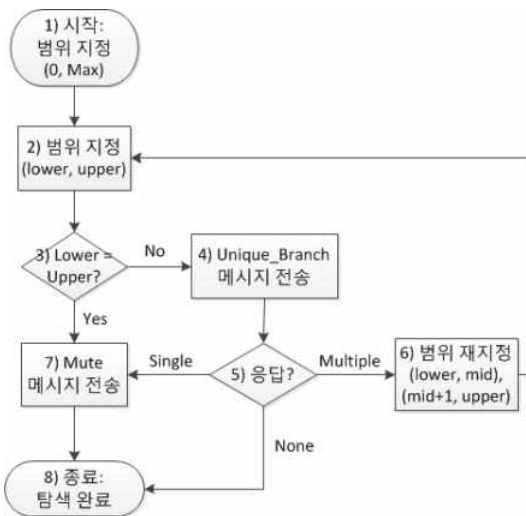


그림 2. RDM 디바이스 탐색 과정

수행한다.

먼저 제어기는 Get_Command 메시지를 디바이스에 전송하여 디바이스에서 사용 중인 파라미터 값을 조회할 수 있다. 또한, 필요한 경우 적절한 값으로 파라미터 값을 설정 혹은 변경하기 위해 Set_Command 메시지를 해당 디바이스에 전송할 수 있다.

예를 들어, Comm_Status 메시지는 네트워크의 구성과 통신 링크의 질을 결정해주기 위해서 사용된다. 이 메시지는 Get, Set 두 가지 형태로 모두 사용이 가능하다. Get_Command와 Set_Command 메시지에 대하여 해당 디바이스는 응답 메시지를 제어기에 전송한다.

그림 3은 디바이스 감시 및 제어를 위해 제어기와 디바이스간에 주고 받는 메시지 흐름을 나타낸다.

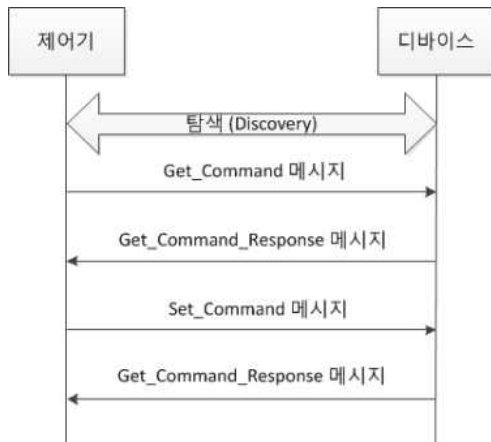


그림 3. RDM 디바이스 감시 및 제어

2.3 메시지

A. Unique ID(UID)

그림 4에서와 같이 RDM 네트워크에서 각 디바이스를 식별하기 위해 48-bit 크기의 UID가 사용된다. UID는 16-bit의 제조사 ID와 32-bit의 Device ID로 구성된다. 제조사는 두 개 이상의 디바이스가 같은 UID를 가지지 않도록 Device ID를

할당해야 한다.



그림 4. UID 형식

B. 메시지 형식

RDM 메시지는 DMX512-A 패킷에 포함되어 전송되며, 세부 적인 메시지 형식(format)은 그림 5와 같다.

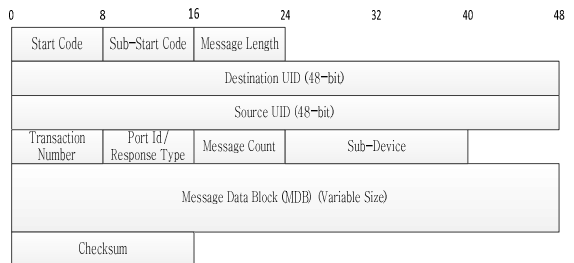


그림 5. RDM 메시지 형식

그림에서 Start Code는 RDM 패킷의 시작을 알리는 Code이며, 값은 0xCC를 사용한다. Sub-Start Code는 추후 확장될 버전을 위해 예약되어진 Code이며, 현재 ANSI E1.20표준에서는 0x01 값을 사용한다. Message Length는 메시지의 길이를 나타내기 위해 사용된다. Start Code부터 Checksum High까지의 Slot(8bit 단위) 수를 나타낸다. Destination 및 Source UID는 송수신 디바이스의 UID를 나타낸다.

Transaction Number는 패킷이 전송 될 때마다 그 값이 증가하게 된다. 초기값은 0으로 시작되며 최대 255까지 증가한다. Port ID/Response Type은 패킷이 Request, Response일 때 달리 사용된다. Request일 때 Port ID로 패킷이 전송되는 Port의 ID(1~255)를 사용한다. Response일 때

Response Type으로 Discovery, Get, Set 메시지에 따른 응답 유형(ACK, ACK_TIMER 등)을 나타낸다. Message Count는 디바이스가 제어기에 보낼 추가적인 데이터의 수를 표기하는데 사용한다. Sub-Device는 같은 속성 값으로 설정되어 있는 디바이스의 수를 나타낸다. Checksum은 패킷의 유효성을 알아내기 위하여 사용되며 패킷 전체 Slot의 값을 더하여 값을 산출한다.

Message Data Block(MDB)는 실제 RDM 제어 메시지를 포함하며 그림 6의 형식을 취한다. 그림에서 Command Class(8-bit)에 표 1에서 정의한 RDM 메시지의 종류가 표시된다. 한편 메시지에 포함되는 파라미터(parameter)를 나타내기 위해 16-bit의 Parameter ID, 8-bit의 Parameter Data Length 및 Parameter Data 필드가 사용된다[3].

Command Class (8-bit)	Parameter ID (16-bit)	Parameter Data Length (8-bit)	Parameter Data (Variable Length)
--------------------------	--------------------------	----------------------------------	-------------------------------------

그림 6. RDM 메시지의 MDB 형식

3. DMP 프로토콜

3.1 개요

PLASA 표준기구에서는 IP 기반의 조명 제어 네트워크 구조를 위해 Architecture for Control Networks(ACN)[5] 표준을 제정하였다. ACN은 구조 표준으로서 여러 프로토콜과 언어의 조합으로 구성되어 있으며, Session Data Transport(SDT) 및 Device Management Protocol(DMP) 등의 프로토콜과 Device Description Language(DDL)를 포함한다. SDT는 인터넷 기반의 조명 제어 네트워크 데이터 전송을 위해 개발되었으며 UDP/IP 위에서 동작하고, DDL는 디바이스를 표현하기 위해 사용되

는 언어이다.

그림 7은 DMP 프로토콜의 동작 개요를 나타낸다. 그림에서 DMP는 Ethernet, UDP/IP, SDT 위에 위치해 있으며 제어기와 디바이스 간에 요청(request) 메시지와 응답(response) 메시지 교환을 통하여 디바이스 관리를 수행하고 있다. 이러한 메시지 교환을 통해 제어기는 네트워크의 디바이스 감시 및 관리 기능을 통해 각 디바이스가 사용하는 속성값을 조회 하거나 설정할 수 있으며 비동기적인 오류 및 이벤트 탐지를 할 수 있다.

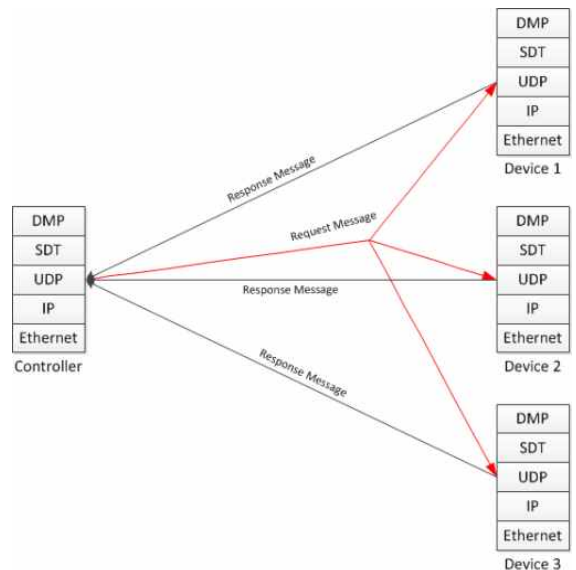


그림 7. DMP 프로토콜 및 메시지 흐름

3.2 메시지 종류

표 3에서 나타내는 DMP의 메시지의 종류는 제어기가 디바이스에게 전송하는 요청(Request) 메시지와 이에 대한 디바이스의 응답(Response) 메시지로 구분되며 총 9가지의 메시지가 사용된다. 표에서 *은 오류 복구를 통한 신뢰전송이 요구됨의 의미하고, X는 해당 메시지가 없는 경우를 의미한다.

표 3. DMP 메시지 종류

No.	요청 메시지	응답 메시지
1	Get_Property	Get_Property_Reply Get_Property_Fail
2	Set_Property	Set_Property_Fail
3	Allocate_Map*	Allocate_Map_Reply*
4	Deallocate_Map*	X
5	Map_Property*	Map_Property_Fail*
6	Unmap_Property*	X
7	Subscribe*	Subscribe_Accept* Subscribe_Reject*
8	Unsubscribe*	X
9	X	Event

A. Get/Set 메시지

표 3에서 유형 1~2번 메시지는 디바이스의 파라미터 정보를 조회하거나 설정하는 경우에 사용된다.

Get_Property 메시지는 제어기가 디바이스의 속성값을 조회하는 메시지이다. 이에 대한 응답 메시지는 Get_Property_Reply 혹은 Get_Property_Fail 메시지가 있다. Get_Property_Reply 메시지는 요청에 따른 속성값을 응답한다. Get_Property_Fail 메시지는 요청에 따른 속성값을 전송할 수 없을 때 전송된다.

Set_Property 메시지는 제어기가 디바이스의 속성값을 설정하는 메시지이다. 이 메시지에 대한 응답 메시지는 Set_Property_Fail이 있으며, 디바이스 속성값의 설정이 성공하지 못하였을 때 전송된다.

B. Map 메시지

표 3에서 유형 3~6번 메시지는 DMP 파라미터의 속성값을 매핑(mapping)하기 위해 사용되는 메시지이다.

DMP에서 각 디바이스의 속성값은 실제값(Actual

Property) 혹은 가상값(Virtual Property)으로 설정할 수 있다. 실제 값을 사용하는 경우 메모리 상의 오버헤드(overhead)가 큰 경우에는 가상 값을 사용할 수 있으며, 이 때 실제값-가상값 간에 매핑 정보에 대한 관리가 요구된다.

먼저, 제어기는 디바이스에게 Allocate_Map 메시지를 전송하여 실제값-가상값 간의 매핑 정보를 저장할 메모리 할당을 요구한다. 디바이스는 매핑 공간을 할당하고 제어기에게 Allocate_Map_Reply 메시지로 응답한다. Deallocate_Map 메시지는 더 이상 매핑 공간 사용이 필요없는 경우에 할당된 공간을 해제하기 위해 사용되며, 이에 대한 응답 메시지는 존재하지 않는다.

한편 실제값-가상값 간의 매핑 정보를 지정하기 위해 제어기는 디바이스에게 Map_Property 메시지를 전송한다. 이를 토대로 디바이스는 할당된 공간에 매핑 정보를 저장한다. 관련 작업이 실패한 경우 Map_Property_Fail 메시지로 제어기에 응답할 수 있다. Unmap_Property 메시지는 저장된 매핑값을 제거하기 위해 사용되며, 이에 대한 응답 메시지는 존재하지 않는다.

C. Subscription 메시지

표 3에서 유형 7~9번 메시지는 디바이스가 제어기에게 비정기적으로 오류 혹은 이벤트(event)를 보고하는 경우에 사용된다.

일반적으로는 Get/Set 메시지를 사용하여 제어기는 정기적으로 디바이스의 상태를 파악하고, 비정기적인 보고 기능을 위해서 '이벤트 보고' 기능을 사용한다. 예를 들어 디바이스에서 속성값의 변경이 일어났을 때 디바이스는 제어기에게 Event 메시지를 전송하여 속성값의 변화를 보고할 수 있다. 상기 보고(report 혹은 notification) 기능을 사용하기 위해서 각 디바이스는 미리 제어기에 가입(subscription)을

해야 한다.

이를 위해 먼저 제어기는 Subscribe 메시지를 디바이스에 전송하여 각 디바이스에게 가입을 요청한다. 이 때 각 디바이스는 Subscribe_Accept 혹은 Subscribe_Reject 메시지로 응답할 수 있다. 이후 디바이스는 수시로 Event 메시지를 통해 제어기에게 변경된 파라미터 속성값을 보고할 수 있다. 제어기는 더 이상 비정기적 보고를 받을 필요가 없는 경우에 Unsubscribe 메시지를 전송한다. 이에 대한 디바이스의 응답은 존재하지 않는다.

3.3 프로토콜 동작

DMP 프로토콜의 동작은 크게 네트워크에 위치한 디바이스의 탐색, 디바이스에 대한 파라미터 관리 및 오류/이벤트 감시 기능으로 구분해 볼 수 있다.

A. 디바이스 탐색

RDM 프로토콜과는 달리 DMP 프로토콜은 디바이스 탐색 기능을 자체적으로 제공하지 않는다. 다만 기존에 유사한 목적으로 사용되는 있는 표준인 Service Location Protocol(SLP)[7]을 사용하는 것으로 규정하고 있다. 즉 ACN 구조[6]에서는 이처럼 네트워크 상의 ACN 디바이스들을 탐색하기 위해 SLP 프로토콜을 사용한다.

B. 디바이스 관리 및 파라미터 매핑 관리

DMP 프로토콜에서 디바이스의 파라미터를 조회 및 설정하는 관리 기능은 RDM 프로토콜과 유사하다. 제어기는 표 3의 1, 2번에 기술된 Get_Property 메시지와 Set_Property 메시지를 디바이스에게 전송하여 각 디바이스의 파라미터 정보를 조회 혹은 설정할 수 있다.

한편 디바이스별로 매우 많은 파라미터를 사용하는

경우 각 파라미터들의 속성값들을 나타내는 정보의 양이 클 수 있다. 대개 실제(real) 속성값들은 4바이트 크기로 표현된다. 이러한 문제점을 해결하기 위해 4바이트의 파라미터 정보를 1바이트 크기의 가상(virtual) 속성값으로 매핑(mapping)하는 기능이 사용된다.

실제 속성값과 가상 속성값을 매핑시키기 위해서는 먼저 각 디바이스별로 실제값-가상값 간의 매핑 정보를 저장할 공간을 확보해야 한다. 이를 위해 제어기는 Allocate_Map 메시지를 보내서 디바이스에 가상 주소공간을 확보 시킨다. 이후 제어기는 Map_Property 메시지를 디바이스에 전송하여 매핑 정보를 관리한다.

그림 8은 DMP 프로토콜에서 실제값-가상값 매핑을 통한 파라미터 설정 및 관리 절차를 보여준다. 그림에서 먼저 디바이스의 가상주소 공간 확보를 위해 제어기는 Allocate_Map 메시지를 전송한다. 공간 확보 후 디바이스는 Allocate_Map_Reply 메시지로 응답한다. 이후 제어기는 Map_Property 메시지를 통해 파라미터 매핑 정보를 전송한다. 파라미터 매핑 정보의 해제를 원하는 경우 Unmap_Property 메시지를 디바이스에게 전송한다. 이후 필요한 작업이 종료되면 제어기는 할당된 공간을 제거하기 위해 Deallocate_Map 메시지를 전송한다.

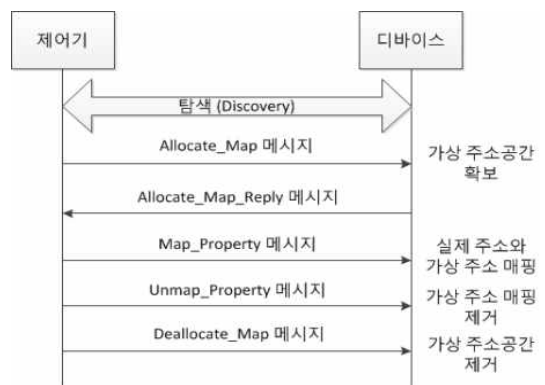


그림 8. 매핑 메시지의 흐름도

C. 비정기적인 이벤트 보고

DMP 프로토콜에서 각 디바이스의 제어기에게 비정기적으로(asynchronous) 이벤트 정보를 전송할 수 있다. 그림 9는 이에 대한 절차를 보여주고 있다.

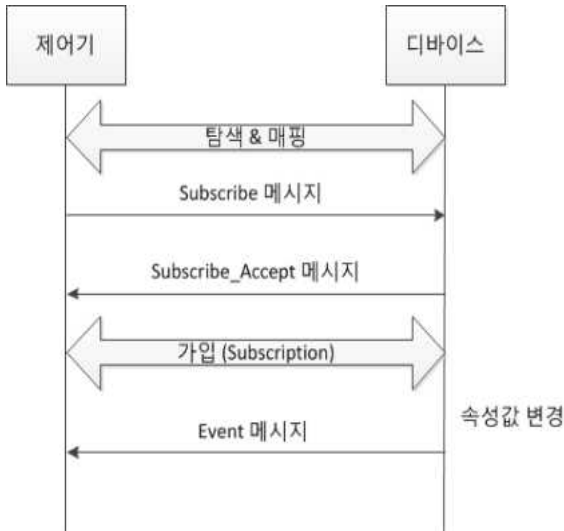


그림 9. 가입 메시지의 흐름도

비정기적인 보고 기능을 사용하기 위해서, 각 디바이스는 먼저 제어기의 해당 서비스에 가입을 해야 한다. 이를 위해 제어기는 디바이스에게 Subscribe 메시지를 전송하고, 디바이스는 Subscribe_Accept 메시지를 보내어 가입을 승낙한다. 이 후 디바이스는 파라미터 속성값이 바뀌는 등의 이벤트가 발생하면, 곧바로 제어기에게 관련 정보를 전송할 수 있다.

3.4 메시지 포맷

DMP의 메시지 포맷은 ACN 구조를[6] 따른다. 그림 10은 DMP 프로토콜의 메시지 형식을 보여준다.

그림에서 DMP 프로토콜의 정보는 Protocol Data Unit(PDU)에 포함된다. 그림에서 알 수 있듯이 여러 개의 PDU 정보가 Block 형태로 취합되어

전송될 수 있다.

PDU 내부에는 위의 그림과 같이 필드가 나뉘어 있다. Flag 필드는 Length, Vector, Header, Data 각각 1bit를 사용한다. Length는 PDU의 전체길이를 나타내고, Vector는 PDU가 지나온 경로를 나타낸다. Header는 PDU 형태에 따른 보충 정보를 나타내고, Data 필드는 DMP 프로토콜 동작에 따른 세부적인 내용을 포함한다. PDU는 여러 개의 PDU들을 하나로 묶어서 PDU Block 형태로 데이터 필드에 포함할 수도 있다.

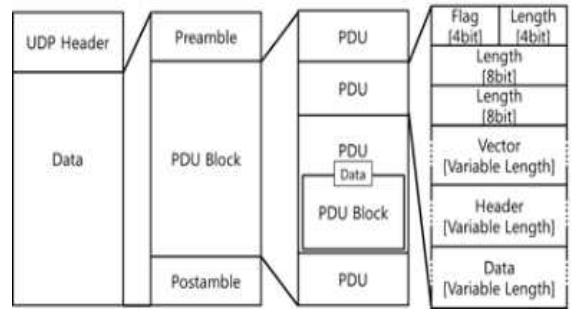


그림 10. DMP 패킷 포맷

이러한 PDU의 집합은 Preamble와 Postamble에 둘러싸여 진다. Preamble은 Transport Protocol에 의존적이면서 Protocol에 필요한 Data를 기록한다. 특별한 내용이 없는 경우는 비어 있을 수도 있다. Postamble 역시 Preamble과 유사하게 사용된다.

4. RDM과 DMP 프로토콜 비교

RDM과 DMP는 모두 조명 제어 네트워크에서 디바이스를 관리하기 위해 개발된 프로토콜이다. RDM은 조명 제어 네트워크 전용 기술인 DMX512-A 네트워크를 사용하는 반면에, DMP는 IP 네트워크에서 사용되는 기술이다.

표 4는 두 프로토콜의 주요 특징을 비교하고 있다.

표 4. RDM와 DMP의 비교

구분	RDM	DMP
Underlying Network	DMX512-A	IP Network
Discovery	O	X
Parameter Control (Get/Set)	O	O
Asynchronous Event (Properties change)	X	O

표에서 알 수 있듯이 RDM은 디바이스 탐색 기능을 제공하는 반면에, DMP에서는 탐색 기능을 자체적으로 제공하지 않고 기존의 SLP 프로토콜을 사용하도록 규정하고 있다. 두 가지 프로토콜 모두 디바이스 파라미터의 조회(get) 및 설정(set) 기능을 제공한다.

RDM 프로토콜에 비해서 DMP 프로토콜이 가지는 특징은 ‘비정기적인 이벤트 보고’ 기능이다. DMP 프로토콜에서는 디바이스의 파라미터 속성값이 바뀌었을 때, 이를 즉시 제어기에게 보고하는 기능을 지원하고 있다.

5. 결 론

지금까지 본 논문에서는 조명 제어 네트워크에서 디바이스를 관리하기 위해 개발된 RDM 프로토콜과 DMP 프로토콜에 대하여 살펴보았다.

RDM은 조명 전용 네트워크 기술인 DMX512-A를 토대로 동작하며, 디바이스 탐색 기능 및 파라미터 관리 기능을 제공한다. DMP는 IP 네트워크 기반의 조명 네트워크를 위한 ACN 구조를 토대로 개발되었으며, 파라미터 조회/설정 등의 관리 기능과 함께 실제값-가상값 매핑 관리 기능을 제공한다. 아울러 DMP는 디바이스의 비정기적인 이벤트 보고 기능을

제공한다.

최근 전 세계적으로 조명산업의 활성화와 함께 네트워크에서 조명 장치를 관리하기 위한 조명 네트워킹의 필요성이 증가하고 있다. 특히 엔터테인먼트 등의 타 산업과 연계하여 조명 네트워킹 기술의 효율적인 사용이 요구되고 있는 상황이다. 이에 수 많은 조명 장치를 효과적으로 관리하기 위한 기술 개발이 요구되며, 향후 국내에서도 이에 대한 연구개발 및 표준화에 적극적인 관심을 기울일 필요가 있다.

Acknowledgement

이 연구는 지식경제부의 지원을 받는 정보통신표준화 및 인증지원사업의 연구결과로 수행되었음.

참 고 문 헌

- [1] 유영수, 문철훈, “임베디드 리눅스를 이용한 원격 조명 네트워크 시스템 구현,” 대한전자공학회 추계학술대회, 2007.
- [2] 김유진, 김인수, 박성희, 강태규, 한동원, “DMX over IP의 LED 조명 제어 네트워크 분석,” 한국정보처리학회 추계학술 발표대회, 2009.
- [3] American National Standard Institute(ANSI), Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories, American National Standard E1.11, 2008.
- [4] Professional Lighting And Sound Association (PLASA), <http://tsp.plasa.org/>.
- [5] American National Standard Institute(ANSI), Remote Device Management Over DMX512 Networks, American National Standard E1.20, 2006.
- [6] American National Standard Institute(ANSI), Architecture for Control Networks, American National Standard E1.17, 2006.
- [7] Internet Engineering Task Force(IETF), Service Location Protocol: Version 2, IETF RFC 2608, 1999.

◇ 저 자 소 개 ◇



이상헌(李相憲)

1985년 4월 30일생. 2012년 금오공과대학교 컴퓨터공학과 졸업. 2012년~현재 경북대학교 컴퓨터학부 석사과정.
관심분야 : 컴퓨터통신, LED 조명 제어 네트워크



최상일(崔相一)

1987년 9월 1일생. 2010년 경북대학교 컴퓨터학부 졸업. 2012년 경북대학교 컴퓨터학부 졸업(석사). 2012년~현재 경북대학교 컴퓨터학부 박사과정.

관심분야 : 컴퓨터통신, LED 조명 제어 네트워크



고석주(高碩住)

1969년 9월 11일생. 1992년 KAIST 졸업. 1994년 KAIST 졸업(석사). 1998년 KAIST 졸업(박사). 1998~2004년 ETRI 표준연구센터 선임연구원.

2004년~현재 경북대학교 컴퓨터학부 교수.

관심분야 : 미래인터넷, LED 조명 제어, 멀티캐스트



김인수(金寅洙)

1974년 2월 3일생. 2000년 2월 건국대학교 졸업. 2002년 2월 건국대학교 졸업(석사). 2008년 2월 건국대학교 졸업(박사). 2009년 5월~현재 한국전자

통신연구원 선임연구원.

관심분야 : 조명 제어 네트워크, LED 가시광 무선통신, LED 감성 조명



임상규(林相圭)

1969년 9월 18일생. 1995년 2월 서강대학교 졸업. 1997년 2월 서강대학교 졸업(석사). 2001년 8월 서강대학교 졸업(박사). 2001년 7월~현재 한국전자

통신연구원 책임연구원.

관심분야 : LED 가시광 무선통신, 조명 제어 네트워크, LED 감성 조명, 초고속 광통신



강태규(姜泰奎)

1961년 3월 29일생. 1996년 전자계산 조직응용 정보처리기술사. 2002년 경기대학교 졸업(박사). 1989년 2월~현재 한국전자통신연구원 LED통신연구실

실장. 2007년 5월~현재 TTA 가시광 통신 서비스 실무반 의장. 2009년 3월~9월 IEEE 802.15.7 VLC Subcommittee 의장. 2012년 2월~현재 기술표준원 조명전문위원회(IEC TC34) 위원.

관심분야 : LED 가시광 무선통신, 조명 제어 네트워크, LED 감성 조명, VLC-LED 드라이빙, 쌍방향 IT 조명 네트워크, 시스템조명 등