

Fundamentals of IoT Network



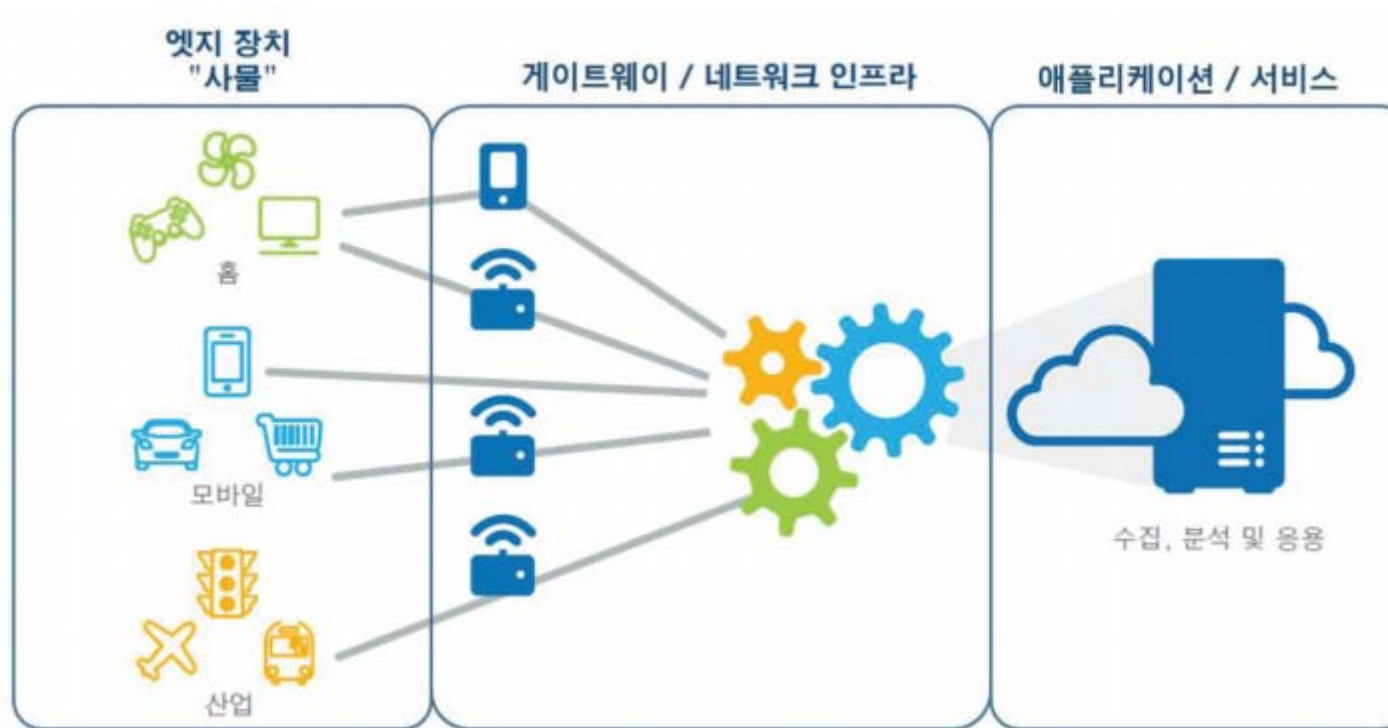
Kim, Eui-Jik

Contents

- IoT Network Structure
- Communication Protocols
- OSI 7 Layer
- Application Protocols for IoT Network
- Q&A

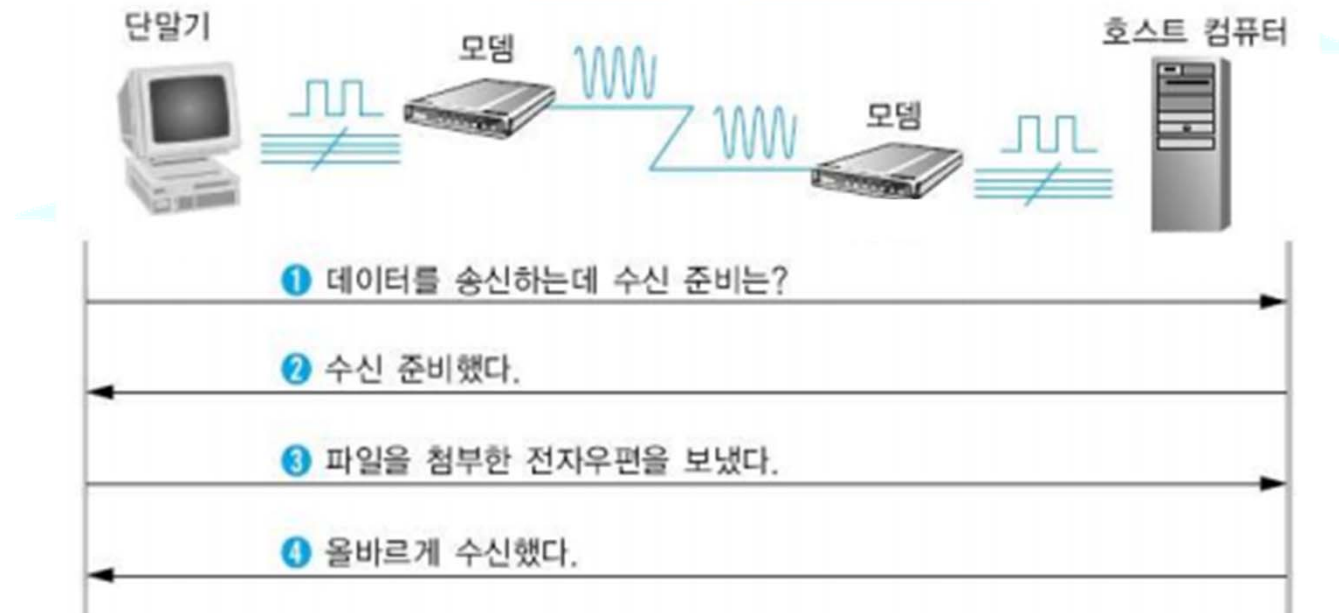
IoT Network Structure

■ IoT 네트워크 구조



Communication Protocols

- 통신 프로토콜의 정의
 - 통신을 원하는 서로 다른 시스템에 존재하는 두 기기들 간에 무엇을 어떻게 언제 통신할 것인가에 대해 서로 약속한 운영규범(혹은 절차)
 - 즉, 컴퓨터와 단말기, 또는 컴퓨터 등 IoT 장치들은 상호정보 교환을 위해 사전에 약속된 프로토콜을 사용함.
 - 장치간 전자우편(데이터) 송수신 예제



Communication Protocols

■ 통신 프로토콜의 기능

■ 흐름제어

- 통신망 내의 원활한 흐름을 위해 송·수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능
- 송신 측과 수신 측 간의 처리속도 또는 버퍼 크기의 차이에 의해 생길 수 있는 수신 측 버퍼의 오버플로(Overflow)를 방지하기 위한 기능을 말함

■ 에러제어

- 전송 중에 발생하는 오류를 검출하여 정정하여 데이터 또는 제어 정보의 파손에 대비하는 기능

■ 동기화

- 송·수신 측이 같은 상태를 유지하도록 타이밍을 맞추는 기능

■ 순서제어

- 전송되는 데이터 블록에 전송 순서를 부여하는 기능

■ 주소지정

- 데이터가 목적지까지 정확하게 전송될 수 있도록 목적지 이름, 주소, 경로를 부여하는 기능

Communication Protocols

- 다중화
 - 한 개의 통신 회선을 여러 가입자들이 동시에 사용하도록 하는 기능
- 경로제어
 - 송·수신 측 간의 송신 경로 중에서 최적의 패킷 교환경로를 설정하는 기능
- 단편화와 재결합
 - 단편화 : 송신 측에서 전송할 데이터를 전송에 알맞은 일정 크기의 작은 블록으로 자르는 기능
 - 재결합: 수신 측에 단편화된 블록을 원래의 데이터로 모으는 기능
- 캡슐화
 - 단편화된 데이터에 송·수신지 주소, 오류 검출 코드, 프로토콜 기능을 구현하기 위한 프로토콜 제어 정보 등의 정보를 추가하는 기능
- 우선순위제어
 - 메시지에 우선 순위를 부여해 우선 순위가 높은 메시지가 먼저 도착하도록 하는 기능

이 외에도 더 많은 기능들이 존재하며,
해당 기능들을 제공하기 위해 다양한 프로토콜들이 존재함.

OSI 7 Layer

■ OSI 7 Layer (1/3)

- 다른 시스템 간의 원활한 통신을 위해 국제 표준화 기구 ISO (International Organization for Standardization)에서 제안한 통신규약 임.
 - 장치간의 데이터 통신시 필요한 장비 및 처리 방법 등을 7 Layer (계층)으로 표준화하여 규정함.
 - 프로세스나 기술적인 면에서 명백히 다른 기능을 처리하도록 계층을 분리함.
 - 인접한 상·하위 계층간에는 계층간 상호작용을 위해 인터페이스를 둠.
 - 각 계층간 독립성 가지기 때문에, 각 계층의 유지보수에 용이함.

■ 각 계층 명칭

- 1계층: 물리 계층(Physical Layer)
 - 2계층: 데이터 링크 계층 (Data Link Layer)
 - 3계층: 네트워크 계층(Network Layer)
 - 4계층: 전송 계층(Transport Layer)
 - 5계층: 세션 계층(Session Layer)
 - 6계층: 표현 계층(Presentation Layer)
 - 7계층: 응용 계층(Application Layer)
- 하위계층
- 상위계층

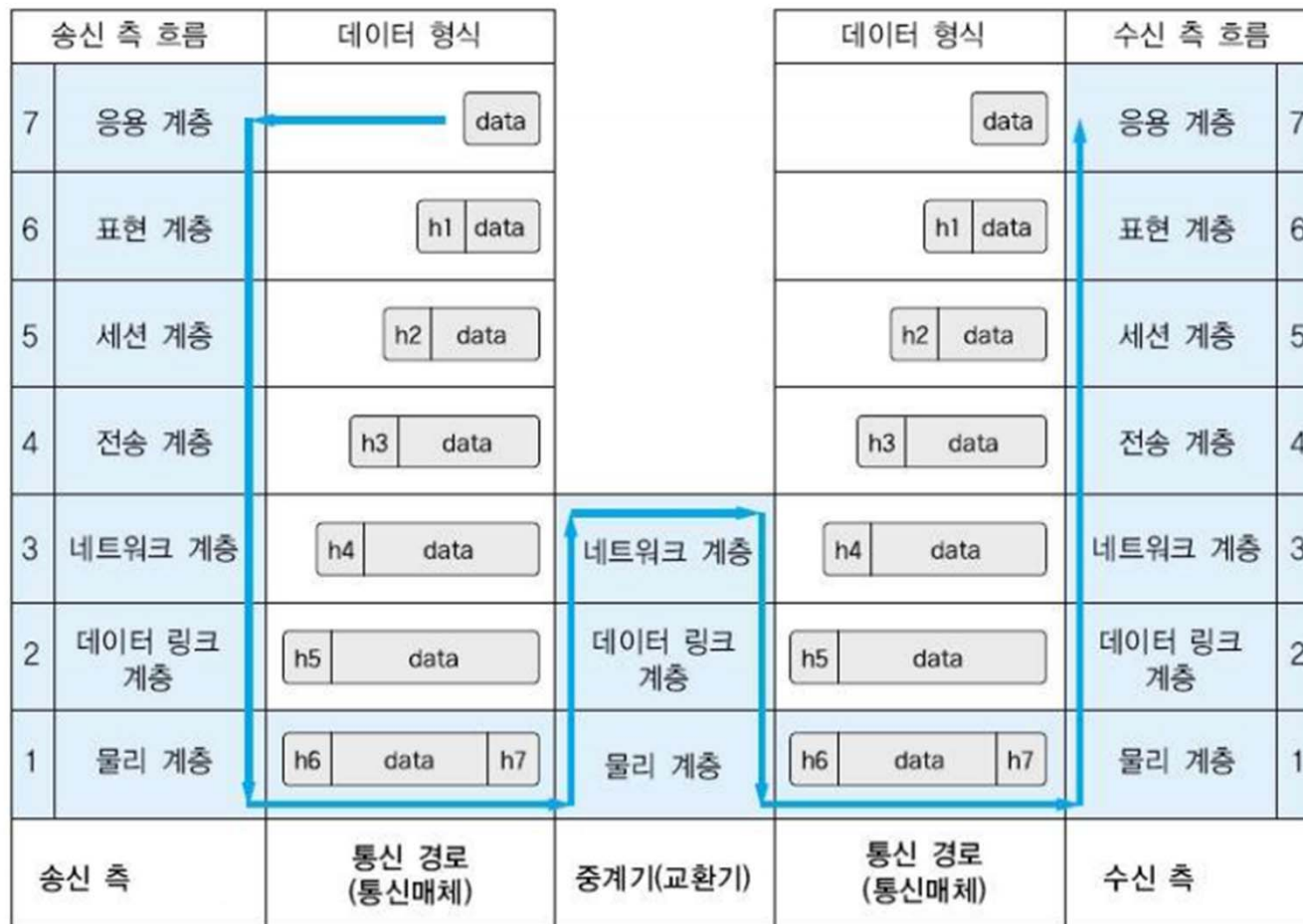
OSI 7 Layer

■ OSI 7 Layer (2/3)

계층	내용
Application Layer	사용자가 네트워크에 접근할 수 있도록 하는 사용자 인터페이스를 제공하고, 전자우편, 원격 파일 전송, DB 관리 등의 응용 서비스를 제공
Presentation Layer	데이터 표현양식(Format) 제공, 통신 및 사용자에게 맞는 변환 (인코딩, 압축, 암호화 등)을 담당
Session Layer	어플리케이션 사이의 연결을 설정, 관리, 해제 하는 기능을 담당
Transport Layer	종단간 (End-to-End) 데이터 전송 방식에 대한 계층으로, 프로토콜에 따라 신뢰성에 차이가 있을 수 있음
Network Layer	장치간 논리적 IP 주소로 식별하고, 패킷을 발신지로부터 여러 링크를 통해 목적지까지 전달하기 위한 경로설정을 담당
Data Link Layer	MAC 주소로 장치를 식별하고, 물리적으로 직접 연결된 노드 간 (Point-to-Point, 점대점) 연결을 보장
Physical Layer	두 장치간의 상호작용을 위해 물리적 매체를 통해 연결하는 역할과 장치가 송신하는 신호를 전기적 신호로 변환하는 역할

OSI 7 Layer

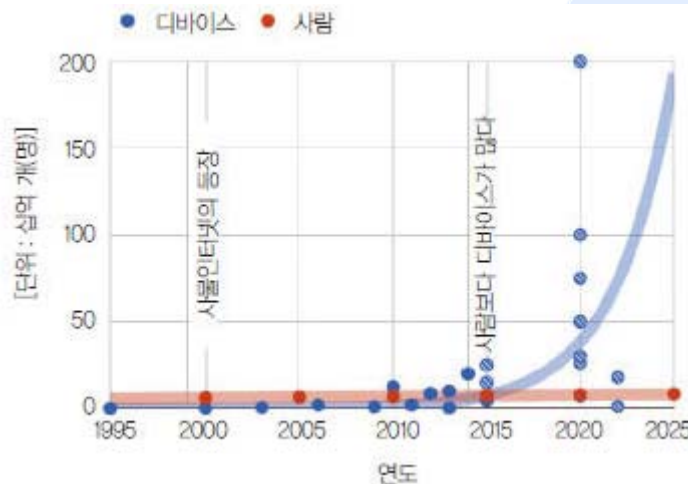
■ OSI 7 Layer (3/3)



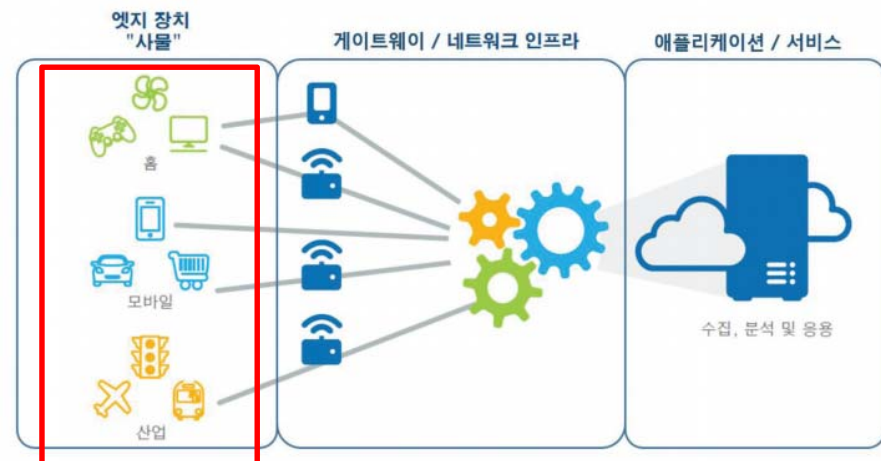
Application Protocols for IoT Network

■ IoT network Issues

- 2014년 이미 커넥티드 장치(IoT 장치)의 수가 세계 인구 수를 넘어섰다고 추정되며, 2020년에는 500억 개에 달하는 IoT 장치들이 서로 연결될 것이라 전망됨.
- '이렇게 연결된 수십억 개의 물리적 리소스(전력, 메모리, CPU 등)가 제한된 IoT 장치'가 어떻게 최종 노드와 클라우드, 그리고 서비스 제공업체들 사이에서 정보를 효율적으로 교환할 할 것인가가 중요한 문제임.



<출처> Philip N. Howard, George Washington Univ.



IoT 장치의 수가 기하급수적으로 늘어나고 있음

Application Protocols for IoT Network

- Application protocols for IoT network

- IoT 장치는 인터넷을 연결을 사용함. 따라서, HTTP(HyperText Transfer Protocol)와 같이 인터넷 연결을 위한 어플리케이션 프로토콜이 필요함.

- 그러나, 기존 HTTP 프로토콜은 수많은 디바이스가 상호작용하는 IoT 환경을 고려하지 않고 개발되었으며, 특히, 많은 전력과 큰 사이즈의 메모리 등 물리적 리소스가 풍부한 장치들 고려하여 개발되었음. 즉, 다양한 기능들을 제공하며, 이를 위해 사이즈가 큰 헤더를 사용함.

➡ HTTP는 IoT 네트워크 환경에 적합하지 않음.

따라서, 혼잡한 네트워크환경에서 동작하는 제한된 리소스를 가진 IoT 장치가 안정적으로 데이터를 전송하고 장시간 운용될 수 있는 효율적이고 가벼운 어플리케이션 프로토콜이 IoT 네트워크 환경을 위해 필요함.

Application Protocols for IoT Network

■ CoAP & MQTT

- CoAP 및 MQTT는 폭발적으로 성장하는 IoT 시장을 위한 주요 경량 어플리케이션 프로토콜(Light-weight application protocol)로서 빠르게 부상하고 있음.
 - CoAP: Constrained Application Protocol
 - MQTT: Message Queuing Telemetry Transport
- 즉, CoAP과 MQTT는 IoT 네트워크에서 제한된 리소스를 가진 IoT 장치간 통신을 지원하기 위해 사용되는 대표적인 경량 어플리케이션 프로토콜임.
 - 메시지 헤더의 사이즈가 HTTP 대비 작음.
 - 에러 보정을 위해 간단한 전송 재시도와 같은 가벼운 신뢰성 전략 등을 제공하나, 복잡한 연산이 필요한 경우 리소스가 더 풍부한 기기에 맡김.

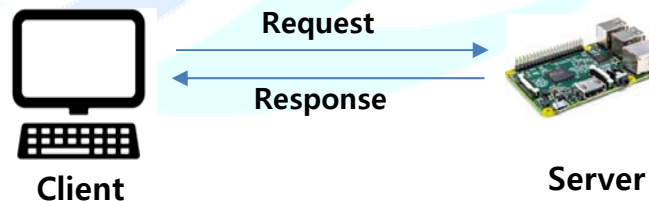
Application Protocols for IoT Network

■ CoAP 개요

- 제약이 있는(constrained) 장치들을 위한 어플리케이션 프로토콜로서 IETF의 RFC 7252 표준으로 채택되었음.
 - <https://tools.ietf.org/html/rfc7252>
- CoAP을 위한 다양한 확장 기능들이, 다른 표준으로 추가적으로 정의됨.
 - RFC 7252 표준 이외에 CoAP 확장 기능을 위한 많은 표준들이 존재함.

■ CoAP 통신 구조

- HTTP와 동일하게 Sever/Client 구조를 가짐.
- Client는 Server에게 데이터를 요청하고, Server는 Client가 요청한 데이터를 전송해줌.



Application Protocols for IoT Network

■ MQTT 개요

- 1999년 IBM에 의해 개발되었으며, 2013년 표준화 단체 OASIS (Organization for the Advancement of Structured Information Standards)에 의해 표준으로 채택됨.
- <http://mqtt.org/>

■ MQTT 통신 구조

- Publish/Subscribe (발행/구독) 구조를 가짐.
- 발행자는 자신의 데이터에 해당하는 토픽을 발행하여 브로커에게 전달하고, 브로커는 해당 토픽의 구독을 원하는 구독자에게 전달함.



Application Protocols for IoT Network

■ CoAP 및 MQTT 상세 통신 구조

