IoT 서비스 탐색 기술

IoT 환경에서 서비스는 어떻게 찾을까?

윤주상, 최영환*
동의대학교, 한국전자통신연구원*

Ⅰ. 서 론

본고에서는 사물인터넷 환경에서 사용 가능한 자원 및 서비스 탐색 프로토콜에 대해서 알아본다.

최근 사물인터넷을 위한 다양한 응용 서비스가 개발 중이다. 개발 중인 사물인터넷 서비스는 통신기능을 가진 센서 디바이스처럼 낮은 파워, 제약적 메모리, 낮은 프로세스로 구성된 자원 제약적 디바이스 (Constrained Device)를 통해 다양한 정보를 수집하고 이런 정보를 활용하는 응용 서비스가 집중적으로 개발 중이다. 특히 사물인터넷 응용 서비스는 서비스 실행 시 필요한 설정 및 관리 측면에서 인간의 개입을 최소화 하기를 원하며 자원 제약적 디바이스 내에 구현 가능하도록 오버헤드를 최소화 해야 한다. 그러나 기 개발된 서비스 및 자원 탐색 프로토콜은 사물인터넷을 구성하는 자원 제약적 디바이스인 "THING"내에 구현하기에 오버헤드가 크기 때문에 부적합하다[1]. 따라서 사물인터넷 환경에서는 기존 방법 보다 서비스 설정 및 탐색 과정에서 오버헤드가 작은 서비스 탐색 프로토콜이 필요하다. 이에 따라 최근 IETF CoRE WG에서는 CoAP 프로토콜 기반의 서비스 탐색 기법이 제안되었다. 제안된 CoAP 기반 서비스 탐색 기법은 분산형 CoAP 자원 탐색 기법과 자원디렉터리(Resource Directory) 기반의 중앙형CoAP 자원 탐색 기법 등이 제안되었다. 우선, 본고에서는 서비스 탐색 프로토콜의 기본적 원리와 기 개발된 탐색 프로토콜들을 살펴보고 자원 및 서비스 탐색을 위해 제안된 두 가지 CoAP 기반 자원 탐색 기법을 자세히 알아볼 것이다.

본고에서는 2장에서 서비스 탐색 프로토콜의 원리를 살펴보고 3장에서 지금까지 제안된 서비스 탐색 프로토콜에 대해서 알아본다. 또한 4장에서 CoAP 기반 서비스 탐색 프로토콜을 살펴보고 5장에서 결론을 내린다.

사물인터넷 서비스 활성화로 인해 CoAP 사용이 증가하고 있으며 IETF core WG에서는 사물인터넷 자원 및 서비스 탐색을 위한 CoAP Resource Directory(RD) 기반 사물인터넷 서비스 탐색 기술 개발이 활발히 진행 중이다.

Ⅱ. 서비스 탐색 프로토콜 원리

일반적으로 서비스 탐색 프로토콜(SDP: Service Discovery Protocol)의 역할은 네트워크 공간 내에 존재하는 서비스를 제공하는 디바이스를 탐색하여 디바이스 내에 서비스에 접근할 수 있는 방법을 제공해주는 것이다. 일반적으로 서비스 탐색 프로토콜은 중앙형 접근 방식(centralized approach) 과 분산형 접근 방식(distributed approach)으로 구분된다. 중앙형 접근 방식은 네트워크에 존재하는 서비스 디바이스의 정보를 수집하고 관리하는 한 개 이상의 서비스 디렉터리를 통해 서비스 탐색 과정이 이루어진다. 이에 반해 분산형 방식은 네트워크 내에 서비스 디렉토리가 존재하지 않으며 각 디바이스 간 서비스 탐색에 필요한 관계를 형성하고 탐색이 이루어지는 방식이다.

서비스 탐색 프로토콜의 주요 기능 및 절차는 출판(Publication), 등록 (Registration), 탐색

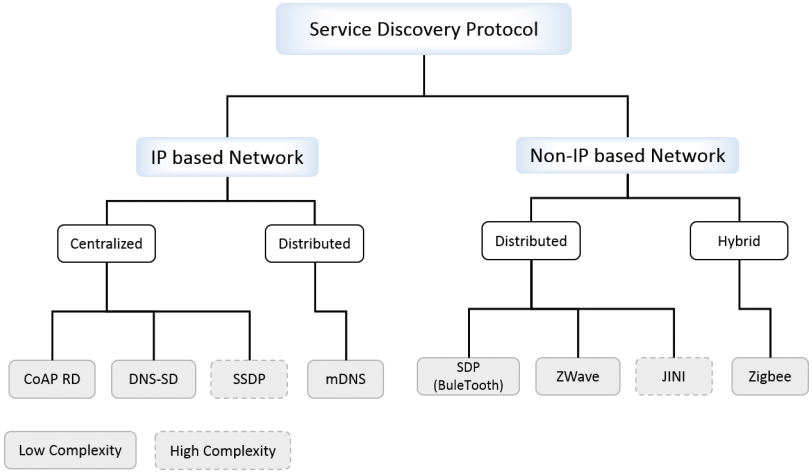

그림 1. 서비스 탐색 프로토콜

(Discovery & Resolution) 등으로 이루어져 있다. 여기서 등록의 경우 중앙형 접근 방식인 서비스 디렉터리가 사용되는 경우는 서비스 엔트리에 대한 디렉토리 탐색, 업데이트, 제거, 타당성 검토 등의 추가적인 기능이 요구된다. 우선, 출판 기능을 살펴 보면 다음과 같다. 출판은 분산형 방식에서 서비스 탐색 프로토콜에 의해서 사용된다. 따라서 서비스 정보를 저장하는 디렉토리를 사용하지 않는 환경에서 수행되는 기능이다. 출판과정에서 서비스 디바이스는 서비스와 관련된 정보들을 다른 디바이스에 알리기 위한 기능으로 정의된다. 여기서 서비스와 관련된 정보들은 서비스 타입, 서비스 접속 방식, 서비스 이름, 서비스 도메인, 서비스 특성 등이다. 서비스 타입은 예를 들어 온도를 수집하는 디바이스의 경우 서비스 타입을 온도수집으로 정의할 수 있으며 접속 방식 정의는 디바이스 IP 주소 및 포트 정보 또는 ID 등으로 정의된다. 서비스 이름은 디바이스 내에 할당된 자원을 서비스로 정의하고 도메인 정보는 디바이스의 위치 정보로 정의된다. 출판 기능은 위에 나열된 서비스 관련 정보들을 하나의 집합 정보로 형성하여 네트워크 내에 다른 디바이스와 정보를 교환한다.

서비스 탐색 프로토콜의 주요 기능 및 절차는 출판(Publication), 등록 (Registration), 탐색 (Discovery & Resolution) 등으로 이루어져 있다.

등록은 네트워크 내에 서비스 디렉터리와 같은 네트워크 내에 특정 요소에 자신의 서비스를 등록하는 과정으로 정의된다. 따라서 서비스를 가진 디바이스가 서비스 디렉터리 기반 등록 과정을 수행하기 위해서는 자신의 서비스를 등록할 디렉터리의 탐색 방법, 등록 업데이트, 등록 타당성 검증, 등록 제어 등의 기능이 요구된다. 탐색은 관련된 서비스를 가진 서버 및 디바이스의 IP 주소 또는 이름을 탐색하는 것으로 정의된다.

Ⅲ. 다양한 서비스 탐색 프로토콜

서비스 탐색 프로토콜은 네트워크 내에 존재하는 서비스 및 디바이스가 관리자 또는 사용자와의 상호작용 없이 자동적으로 서비스 및 서비스가 정의된 디바이스를 탐색할 수 있는 기능 제공을 목적으로 하고 있다. 또한 서비스가 정의된 네트워크 내의 IP 사용 유무, 서비스 제공 방법에서의 토폴로지, 복잡성 등에 따라서 특징지어 질 수 있다. 지금까지 개발된 서비스 탐색 프로토콜을 위에 기술된 특징으로 분류하면 〈그림 1〉과 같다. 서비스 탐색 프로토콜은 사용되는 네트워크 환경에 따라서 IP 네트워크 환경과 Non-IP 네트워크 환경으로 분류가 가능하며 프로토콜 동작의 복잡도 관점에서도 분류가 가능하다. 또한 서비스 및 디바이스 정보를 디렉터리 기반으로

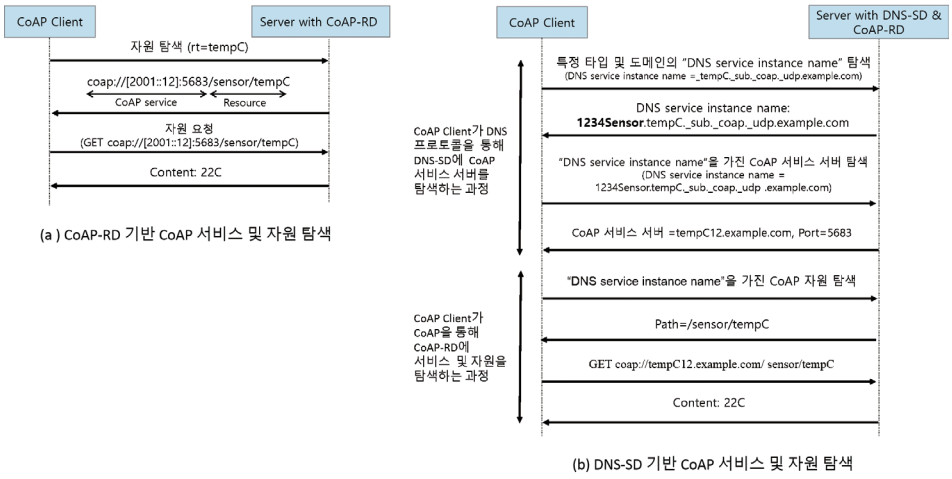

그림 2. CoAP 서비스 및 자원 탐색

관리하는 중앙형 프로토콜과 정보를 분산형으로 관리하는 분산형 프로토콜로 분류된다. 우선 IP 네트워크에서 사용되는 IP 기반 서비스 탐색 프로토콜 중에서 서비스 탐색에 필요한 정보를 중앙에서 관리하는 중앙형 프로토콜은 CoAP RD[2], DNS-SD(DNS-Based Service Discovery)[3], SSDP based UPnP등이 있다. Simple Service Discovery Protocol(SSDP)[4] 는 일반 거주지 및 소규모 사무환경에서 UPnP(Universal Plug and Play)를 위한 프로토콜로 널리 사용되고 있으며 마이크로 소프트, 인텔, 소니, 삼성과 같은 벤더 등이 오피스 환경에서 컴퓨터 및 디지털 응용 제품을 탐색할 때 널이 사용된다. SSDP는 Uniform Resource Identifier(URI)로 서비스를 정의하며 탐색과정에서 HTTP를 사용하며 HTTP 알림 기능을 통해 서비스를 네트워크 내에 알린다. 여기서 HTTP를 사용하기 때문에 복잡성이 높다 평가되고 있다. 따라서 IoT 로컬 네트워크 환경인 저전력 네트워크 환경에서 서비스 탐색 프로토콜로 사용하기에 비용이 많이 발생하는 것으로 평가되고 있다.

서비스 탐색 프로토콜은 IP 네트워크 환경과 Non-IP 네트워크 환경으로 분류 가능하며 IP 네트워크 환경에서 사용되는 대표적인 서비스 탐색 프로토콜로 CoAP RD 및 DNS-SD를 들수 있으며 Non-IP 네트워크 환경은 ZWave, Bluetooth-SDP, JINI 등을 들수 있다.

다른 대표적인 서비스 탐색 프로토콜은 애플이 구현한 Bonjour가 있다. Bonjour은 IETF Zeroconf(Zero Configuration Networking) 프로토콜을 구현한 것으로 오피스 환경에서 서비스 및 제품을 탐색할 때 사용된다. Bonjour은 DNS-SD기능을 가진 mDNS (multicast-DNS)[5]를 통해 기존 인터넷 인프라 기반에서 서비스 탐색이 이루어진다. mDNS는 복잡도가 낮기 때문에 자원 제약적 네트워크 환경에 사용 가능하다고 평가 받고 있다. 또한 CoAP RD(Resource Directory)의 경우 IETF CoRE WG에서는 CoAP 프로토콜 구현과 함께 개발 중인 서비스 및 자원 탐색 기법으로 자원 제약적 네트워크 환경에서 사용하기 위한 목적으로 개발 중이다.

이밖에 Non-IP 기반 서비스 탐색 프로토콜로는 ZWave[6], Bluetooth Service Discovery Protocol (SDP)[7], JINI[8] 등이 있다. 이 프로토콜들은 IP 기반 프로토콜과 다르게 IP 계층 아래에서 서비스 탐색과정이 이루어지는 프로토콜이다. 따라서 사물인터넷 로컬 네트워크 환경인 저전력 무선 네트워크 환경에서 자원 제약적 디바이스에 사용하기에 적합한 기술이다. 하지만 최근 사물인터넷 환경에서는 자원 제약적 디바이스의 IP 연결성에 대한 요구가 증가하고 있다. 따라서 복잡도가 낮은 IP 기반의 서비스 탐색 프로토콜이 제안되고 있다. 더불어 Non-IP 기반 자원 제약적 디바이스를 수용하기 위해 IP 게이트웨이를 활용하여 Non-IP 기반 자원 제약적 디바이스를 수용하기 위한 기술 개발이 제안되고 있다. 따라서 다음 장에서는 최근 개발 중인 IP 기반의 서비스 탐색 프로토콜인 CoAP 및 DNS 기반 서비스 탐색 기법에 대해서 자세히 살펴본다.

Ⅳ. CoAP 및 DNS 기반 서비스 탐색 기법

최근 CoAP-RD과 DNS-SD 기법은 사물인터넷 환경에서 서비스 탐색 방법으로 고려 중이며 관련 기술이 개발 중이다. 특히 DNS-SD는 서비스 탐색을 위한 방법으로 사용되며 CoAP의 서비스 탐색 시 상호 보완적 방법으로 활용이 가능하다. 여기서 DNS가 고려된 이유는 자원 제약적 디바이스의 메모리 자원 측면에서 적은 비용으로 사용이 가능하기 때문이다. 우선, CoAP은 CoAP 자원의 위치와 식별 방법으로 URI를 사용한다. CoAP URI 방법은 다음과 같은 형식으로 자원을 정의한다.

• CoAP URI format: "coap:""//"host [":"port] path-abempty ["?" query]

위 형식에서 호스트 부분은 IP 주소 또는 등록된 디바이스 이름이 위치한다. 만약 호스트가 등록된 디바이스 이름 또는 서비스 이름이라면 DNS를 통해 디바이스 IP주소를 얻기 위한 과정이 추가된다.

사물인터넷 서비스 탐색 기술은 CoAP-RD 기술과 DNS-SD 기술이 상호 보완적 형태로 개발이 진행 중이며 자원 탐색 방법도 분산형 방식과 중앙형 방식으로 개발 중이다.

사물인터넷 환경에서 서비스 탐색 과정은 서비스 탐색과 자원 탐색으로 구분된다. 또한 CoAP-RD와 DNS-SD 프로토콜 사용에 따라서 서비스 및 자원 정의를 다르게 사용하고 있다. 우선, CoAP을 사용할 경우, 서비스는 프로토콜, 호스트, 포트 등의 정보 집합으로 정의된다. 이는CoAP 관점에서 자원을 가진 CoAP서버를 정의하는 거와 같다. 따라서 서비스는 coap://[2001::11]:5683로 정의된다. 또한 자원 정의는 REST 기반의 상호작용을 위한 포인트로 정의되면 coap://[2001::11]:5683/sensor/temp로 정의된다. 더불어 CoAP 자원은 자원 타입(rt)으로 정의되며 temp를 자원 타입으로 보면 된다. CoAP관점과 다르게 DNS-SD를 이용하는 경우는 서비스를 서비스의 서브타입과 프로토콜 타입 등으로 정의된다. 따라서 서비스는 _tempC._sub._coap. _udp 와 같은 형태로 정의된다. 여기서 서비스의 서브타입은 Standards Development Organization(SDO)에서 정의된 형식을 따른다. 또한 서비스는 도메인과 함께 정의되며 _tempC._sub._coap._udp.example.com 형태로 정의된다. 전체 형식은 _1234Sensor._tempC._sub._coap._udp.example.com.이다. 〈그림 2(a)〉는 CoAP RD 기반의 CoAP 서비스 및 자원 탐색 과정을 도시하고 있으며 〈그림2(b)〉는 DNS-SD 기반의 CoAP 서버 및 자원 탐색을 도시하고 있다. 〈그림 2〉는 전체적인 서비스 및 자원 탐색 개념 모델을 도시하고 있다.

본고에서는 CoAP의 서비스 및 자원 탐색 기법을 자세히 알아본다. CoAP에서는 분산형 CoAP 자원 탐색 기법과 자원디렉터리 기반의 중앙형 CoAP 자원 탐색 기법 등이 제안 되어 있다. 분산형 CoAP 자원 탐색 기법은 자원디렉터리와 같은 외부 요소를 이용하지 않고 디바이스 스스로 자원 탐색을 수행하는 방법이며 이에 반해 자원디렉터리 기반 서비스 및 자원 탐색 기법은 모든 자원을 자원디렉터리 내에서 관리하고 자원에 대한 문의를 자원디렉터리 내에서 처리하는 방식인 중앙 관리 방식을 사용한다. 분산형 자원 탐색 기법에 대해서 자세히 살펴본다.

Distributed CoAP Resource Discovery: 이 기법은 자원디렉터리가 필요 없는 방법으로 다른 디바이스 내에 존재하는 자원을 탐색이 CoAP 디바이스 내에서 이루어진다. 또한 CoAP 클라이언트와 CoAP 서버 사이에 직접 자원 탐색을 위한 문의를 요청하는 방식이다. 따라서 클라이언트는 서버의 IP 또는 도메인 이름을 알고 있는 경우에 가능한 방법이다. 〈그림 3〉는 빛 조절 기능을 가진 스마트 스위치와 램프를 가진 CoAP 서버(센서)의 자원을 탐색하는 3가지 방법을 보여 주고 있다.

CoAP 클라이언트인 스마트 스위치는 CoAP 서버인 센서디바이스의 URI 정보인 "/.well-known/core" 정보를 알고 있다는 가정에서 자원 탐색이 이루어진다. 여기서 앞에서 정의된 서

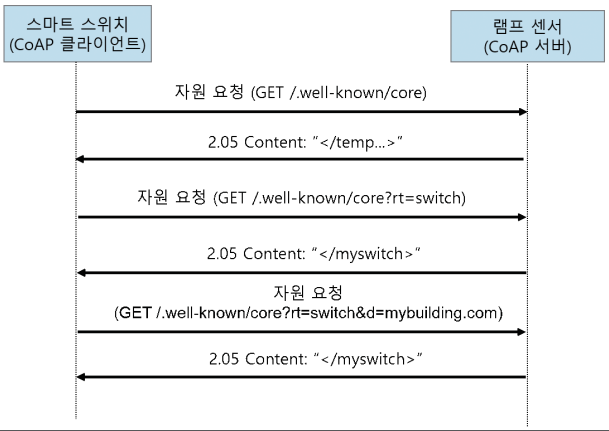

그림 3. 분산형 RD 탐색

비스 탐색의 경우 서버의 정보를 알고 있기 때문에 자원 탐색과정만 이루어 진다. CoAP GET 함수를 이용하여 CoAP 서버에 자원 정보를 요청한다. 또한 CoAP 클라이언트인 특별한 CoAP 자원에 대한 자원 탐색이 가능하다. 이는 "parameter=value"값을 URI 내에 표현하여 CoAP 서버에 요청하는 방식으로 자원 탐색을 수행한다. 마지막으로 특정 도메인 내에 존재하는 CoAP 자원을 탐색하기 위해 "URI/.well-known/core?rt=switch&d=mybuilding. Com" 형식으로 자원 탐색을 CoAP 서버에 요청하여 자원 탐색을 수행한다. 여기서 URI 내에 표현된 자원 탐색은 도메인 내에 존재하는 스위치 자원에 대한 탐색이다.

Centralized CoAP Resource Discovery: 이 기법은 네트워크 내에 자원 탐색을 위해 네트워크 내에 자원 정보를 관리하기 위해 자원디렉터리를 이용한다. 여기서 자원디렉터리는 CoAP 서버들의 자원을 저장하고 CoAP 클라이언트의 자원 탐색 문의에 응답하기 위한 기능을 수행하는 목적으로 생성된다. 따라서 자원을 가진 CoAP 서버는 자원디렉터리에 자원을 등록하는 과정이 필요하다. 또한 CoAP 클라이언트가 자원디렉터리를 통해 자원 탐색을 수행하기 위해 자원디렉터리 서버에 접속하기 위한 서버 정보 및 별도의 접속 방법을 가지고 있어야 한다. 따라서 사물인터넷 환경에서 자원디렉터리 서버는 경계 라우터 내에 위치 시킨다. 또한 router advertisement 메시지를 통해 자원디렉터리 서버 정보를 네트워크 내에 전달한다. 자원디렉터리 서버 정보를 알고 있는 CoAP 서버는 자원디렉터리 서버 탐색이 완료되고 이후 자신의 자원 정보를 등록하는 과정을 수행한다. 이때 CoAP 서버는 자원 등록을 위해 자원디렉터리 내에 정보를 저장할 위치의 디렉터리를 요청(?rt= core.rd)하며 자원디렉터리 서버는 자원을 저장할 디렉터리 정보(/rd)를 CoAP 서버에게 응답해준다. 그 이후 CoAP 서버는 POST 함수를 통해 자원 저장을 위한 디렉터

분산형 CoAP 자원 탐색 방법은 네트워크 내에 RD 서버 없이 디바이스 스스로 자원 탐색이 가능한 방식이며 중앙형 CoAP 자원 탐색 방법은 네트워크 내에 RD 기능을 가진 서버를 통해 서비스가 이루어진다.

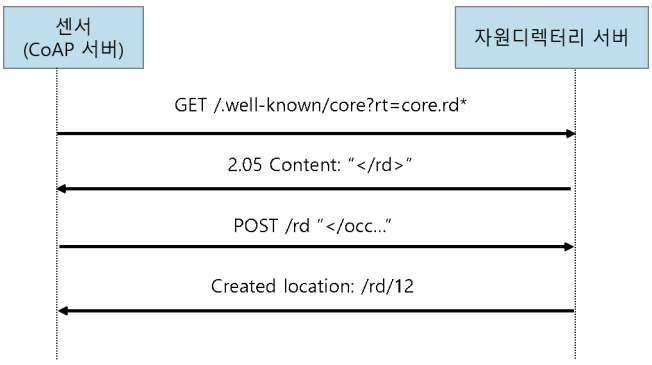

그림 4. CoAP RD 등록 과정

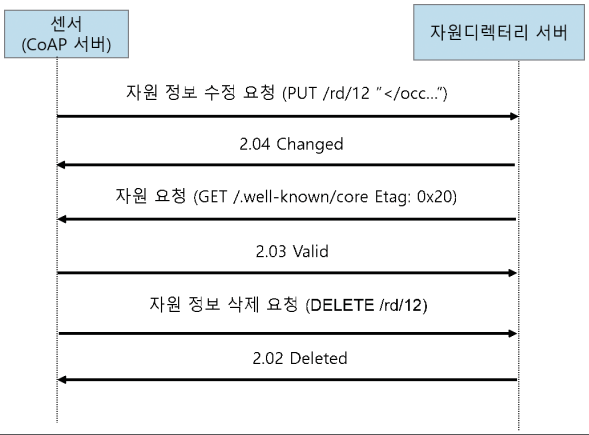

그림 5. CoAP RD 관리 인터페이스

리 생성을 요청하고 자원디렉터리는 자원 저장을 위해 생성된 디렉터리 위치(/rd/12)를 응답한다. 이 과정은 〈그림 4〉에 도시되어 있다.

자원디렉터리 서버와 CoAP 서버 사이에서 등록과정이 완료되면 자원 관리를 위해 필요한 Update, Validation, Removal 인터페이스를 생성한다. 〈그림 5〉는 자원디렉터리 서버와 CoAP 서버 사이에서 자원 관리를 위해 생성된 인터페이스를 통해 자원이 관리되는 예를 보여주고 있다. 필요 시 서버는 Update 인터페이스를 통해 CoAP PUT 함수로 자신의 자원 값을 업데이트 할 수 있으며 자원디렉터리는 PUT 함수가 요청되면 성공 유무를 서버에 응답한다. 또한 자원디렉터리는 자신이 가진 자원 값에 대한 검증을 서버에 요청할 수 있으며 이는 Validation 인터페이스를 통해 이루어 진다. 추가적으로 서버는 자신의 자원을 삭제하기 위해 Removal 인터페이스를 통해 CoAP DELETE 함수를 자원디렉터리에 요청하고 자원디렉터리 내에 관련 정보가 삭제되면 서버에 삭제 정보를 알려준다.

중앙형 CoAP 자원 탐색 기술은 자원디렉터리 서버와 CoAP 서버 사이에 자원 등록과정이 필요하며 등록 완료 후 CoAP 클라이언트는 등록된 자원을 탐색할 수 있다.

CoAP 서버는 자신의 정보가 자원디렉터리에 등록이 완료된 후 CoAP 클라이언트는 자원디렉터리를 통해 CoAP 서버의 자원을 탐색할 수 있다. CoAP 클라이언트가 CoAP 서버의 자원을 탐색할 시CoAP GET 함수를 통해 자신이 원하는 자원 타입(rt) 또는 서비스 정보(ep:endpoint) 등으로 값을 정의하고 자원 디렉터리에 문의한다. 〈그림 6〉에 도시된 예를 보면 CoAP 클라이언트는 자원 타입을 switch로 정의하여 문의를 하였고 이 경우는 자원디렉터리 서버에서 자원 정보로 정의된 경로 정보를 응답한다. 또한 서비스 정보를 문의할 경우 자원디렉터리 서버는 ep 타입으로 자원디렉터리 내 서비스 정보를 전달해 준다. 이런 ep 타입으로 정의된 문의는 서비스 탐색으로 정의될 수 있다.

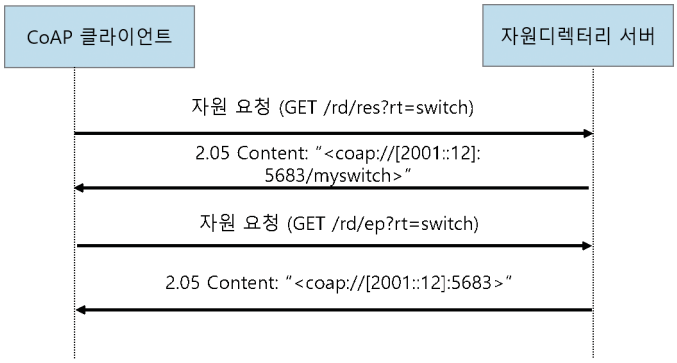

그림 6. RD 기반 자원 탐색 과정

V. 결 론

본고는 사물인터넷 환경에 적용 가능한 서비스 탐색 프로토콜에 대해서 알아보았다. 최근 사물인터넷 서비스 개발 확산으로 인해 서비스 및 자원을 탐색하기 위한 기 개발된 다양한 서비스 탐색 프로토콜이 검토 중에 있다. 하지만 기 개발된 서비스 탐색 프로토콜은 사물인터넷 로컬 네트워크 환경인 자원 제약적 네트워크 환경에 적용하기에 오버헤드가 크다고 평가 받고 있다. 또한 사물인터넷 환경에서 자원 제약적 디바이스의 IP 연결성에 대한 요구가 증가함에 따라 복잡도가 낮은 IP 기반의 서비스 탐색 프로토콜 개발이 요구 되고 있다. 이를 위해 CoAP RD 및 DNS-SD 방법이 제안되어 개발 중이다. CoAP RD 기술은 낮은 오버헤드로 인해 IP 기반의 자원 제약적 디바이스에 적용할 수 있는 장점을 가지고 있으며 DNS-SD는 기존 네트워크 환경에 쉽게 적용할 수 있는 장점을 가지고 있다. 따라서 사물인터넷 환경에 적합한 서비스 탐색 프로토콜로 CoAP RD와 DNS-SD 방법을 결합한 상호 보완적 형태로 개발이 진행 될 것으로 예상된다.

사물인터넷 환경에서의 서비스 탐색 기술은 CoAP RD와 DNS-SD가 결합된 형태로 기술 개발이 진행될 것이다. .

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술연구 진흥센터의 정보통신 · 방송 연구개발사업 [R0166-15-1008, 인터넷 기반 IoT 연동 기술 표준개발]과 한국연구재단의 지원을 받아 수행된 기초연구사업 (NRF-2010-0024523)의 일환으로 수행하였음.

참 고 문 헌

[1] B. C. Villaverde, De Paz Alberola, A. J. Jara, S. Fedor, S. K. Das, and D. Pesch, "Service Discovery Protocols for Constrained Machine-to-Machine Communications," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 41-60, first quarter, 2014.

[2] Z. Shelby, M. Koster, C. Bormann, and P. van der Stok, "CoRE Resource Directory," IETF Internet Draft, draft-ietf-core-resource-directory-04,"July 2015.

[3] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," IETF RFC 6763, February 2013.

[4] Y. Yaron, T.C. Goland, P. Leach, Y. Gu and S. Albright, "Simple Service Discovery Protocol/1.0 Operating without an Arbiter," IETF Internet Draft, draft-cai-ssdp-v1-03, April 2000.

[5] S. Cheshire and M. Krochmal, " Multicast DNS," IETF RFC 7228, February 2013.

[6] Network M. Galeev, "Catching the Z-Wave," In technical papers and application notes on embedded & system design, EE Times India, October, 2006.

[7] Bluetooth Service Discovery Application Profile (SDAP), October 2013, from https://www.bluetooth.org/Building/HowTechnologyWorks/ProfilesAnd Protocols/SDAP.htm.

[8] Jini Specification, October 2013, from http://river.apache.org/doc/spec-index.html.