규모 가변형 클라우드 IoT 플랫폼을 위한 더블헤더 캡슐화 로드밸런싱 서비스

손승철, 고석갑, 이병탁*
한국전자통신연구원

{sson, softgear, bytelee}@etri.re.kr

Double header encapsulation load balancing service for scalable cloud IoT platform

Seung-Chul Son, Seokkap Ko, Byung-Tak Lee
Electronics and Telecommunications Research Institute (ETRI).

요 약

클라우드 상에서 IoT 서비스 플랫폼을 운영하기 위해서는 장치의 규모에 따라 부하분산 정책을 적용할 수 있어야 한다. 본 논문에서는 클라우드 환경에서 확장 가능한 IoT 마이크로서비스 플랫폼을 위한 로드밸런서에 집중한다. 제안하는 DTLS 로드밸런서는 IP/UDP 더블헤더를 기반으로 UDP 터널링을 수행한다. 장치의 규모에 따라 확장된 동일한 기능의 서비스에게 네트워크 부하를 분산함은 물론, 서비스 장애에 대응하며 효율적인 DTLS 암호문 교환도 지원함으로써 궁극적으로 대부분의 클라우드를 지원한다.

Ⅰ. 서 론

IoT 의 궁극적인 목표는 네트워크로 연결된 사물이 서로 협력하여 인간에게 유용한 서비스를 제공하는 것이다. 따라서 IoT 관련 표준을 선점하기 위하여 국내 및 국제 표준화기관의 활동이 활발하며, 지속적인 시장 요구사항을 수용하여 표준화를 제고하고 있다. 특히 OMA 가 개발한 IoT 장치 관리용 LwM2M[1]표준은 장치의 정보 검색 및 제어를 위해 오버헤드가 낮은 CoAP[2] 통신을 채택하고 있다. 최근에는 국제적인 기업들이 표준 권고안에 기반하여 LwM2M 기반 IoT 제품을 출시하는가 동시에 그들을 관리하기 위한 단일 장치형 서버 플랫폼을 제공하고 있다. LwM2M IoT 플랫폼 시장은 5G 와 같은 네트워크 기술 및 인프라가 확대되고 빅데이터, 머신러닝, 증강현실 등 다양한 차세대 기술을 흡수하게 되면서 더욱 확대될 것으로 예상된다. 따라서 LwM2M IoT 플랫폼은 서비스의 규모에 따라 장치를 관리하고 사용자에게 서비스를 제공하기 위한 확장성을 제공해야 한다. IoT 플랫폼이 확장성을 제공하기 위해서는 기존의 모놀리식 플랫폼을 탈피하고 Microservice Architecture (MSA) 기반으로의 변화되어야 한다. 본 논문에서는 모놀리식 IoT 서버 플랫폼을 대체할 클라우드 LwM2M IoT 서비스의 확장성을 위한 DTLS LB(Load Balancing) 기술을 제안한다.

Ⅱ. 본론

제안 기법은 그림 1 과 같은 규모 가변적인 클라우드상 IoT 마이크로서비스 플랫폼에서 장치의 규모에 따라 확장된 ICS(IoT Core Service) 즉, LwM2M 서비스 복제본에게 장치로부터의 발생하는 네트워크 부하를 또 하나의 MS(Microservice)인 BLB(Backend Load Balancer)를 통하여 분산시킴과 동시에 DTLS 보안 통신을 제공하기 위하여 설계되었다. ICS 간 메시징용 공유메모리는 키-값 쌍의 데이터구조를 갖고 발행(PUB: publish)/구독(SUB: subscription), 메시지큐 기능을 포함하는 Redis 를 활용하였다.

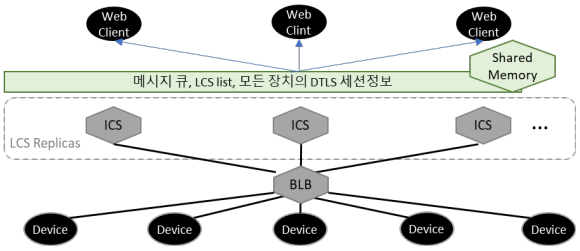

그림 1. LwM2M 마이크로서비스 플랫폼 개요

MS 가 오케스트레이션 된 플랫폼에서 기존에 사용되는 LB 로 사용되는 Nginx 는 프론트엔드 웹사용자의 HTTP 부하분산 및 백엔드 장치로부터의 UDP 부하분산을 지원한다. SDN(Software-Defined Networking)에서 서버통계를 기반으로 하는 프론트엔드 웹 부하분산 기법도 제안되었다[3]. 하지만 기존 기법들은 UDP 부하분산시 매번 새로운 세션을 만든다는 단점이 있다. 게다가 재전송 메시지 조차도 새로운 세션을 만들며 같은 클라이언트의 패킷이 다른 서버에 전달될 수도 있고 Server 상태를 확인하지 않고 전송을 시도함으로 불필요한 트래픽을 유발한다. 대안으로 L2 Direct 라우팅, NAT, IP Tunneling 등의 방법이 있다. 첫번째는 LB 와 Server 가 동일한 IP 사용하지만 서로 다른 MAC 의 헤더 변경으로 전달하는 방법으로, 적은 LB 부하, 장치 IP 를 보지 않으므로 장치에 따른 MS 할당 불가, 망 환경 즉. Martian filter 에 따라 적용이 불가하다는 단점이 있다. 두번째는 LB 가 IP, Port 번호를 변경하는 방법으로 IP, Port 테이블 유지와 패킷 헤더 변경 및 계산 필요하고 원래 IP/UDP header 변경되어 보안문제 가능성이 존재한다. 일부 클라우드에서는 사설 IP 사용을 차단하는 경우도 있으므로 클라우드 적용에 어려움이 있다. 마지막은 IP 헤더를 하나 더 붙여서 전달하는 방법으로 원시 IP 와

UDP 헤더를 변경하지 않는다는 장점이 있으나, 일부 클라우드에서는 IP/IP 를 허용하지 않는다.

본 논문에서는 그림 2 와 같이 BLB 가 IP 와 UDP 헤더를 하나 더 붙여서 ICS 전달하는 UDP 더블헤더 터널링 방법을 제시한다. 원시 IP 및 UDP 헤더를 변경하지 않기 때문에 대부분의 클라우드에서 LB 지원이 가능하다. 물론, 데이터 그램이 약간 커질 수 있지만 클러스터 내부의 MS 간의 통신 부분에만 해당하며 외부망에는 영향을 주지 않기 때문에 무시할 수 있다. 장치로부터 패킷이 도착하면, BLB 는 소스 IP 와 Port 를 추출하고 특별히 관리되는 장치-ICS 맵으로부터 검색된-검색 실패 시 라운드 로빈 방식 담당 ICS 매핑-담당 ICS 에게 더블헤더가 추가된 패킷을 전달한다.

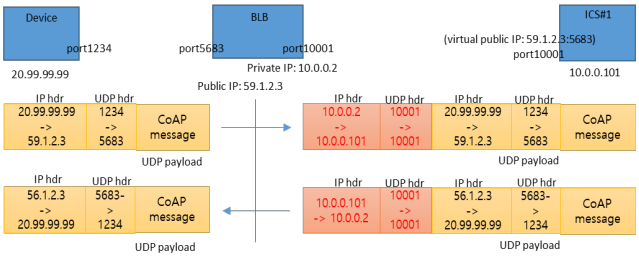

그림 2. UDP 터널링 예제

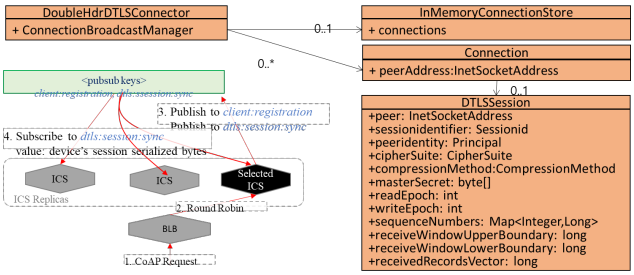

그림 3. DTLS 세션 동기화 과정

장치-ICS 맵이 변경되는 경우로는 오랫동안 장치로부터 패킷이 오지 않아 타이머가 만료되는 경우나 임의의 장치와 메시지를 교환중인 ICS 에게 장애가 발생할 경우 등이 있다. 이러한 상황이 발생하면 해당 장치는 새로운 ICS 와 매핑된다. 평문 통신을 할 경우 새로운 ICS 와 통신에 아무런 문제가 없지만 암호문 통신의 경우 새로운 DTLS 세션 핸드쉐이크 과정으로 인한 시스템 성능 저하 필연적이다. 장치-LB 간에는 암호문, LB-ICS 간에는 평문이 되도록 BLB 에 암복호 기능을 추가할 수도 있지만 이는 로드밸런서의 복잡도와 로드를 증가시킨다. 제안 기법에서는 성능 유지를 위해 모든 ICS 가 모든 장치의 DTLS 세션을 공유하므로써 핸드쉐이크 과정 없이 암호문 통신을 가능하게 하였다. 다만 DTLS 세션 자체가 만료될 경우 새로운 핸드쉐이크 과정은 피할 수 없다. 그림 3 은 ICS 간 공유하는 DTLS 세션 동기화 과정과 구현된 클래스 다이어그램을 보여준다. 장치 등록 메시지를 수신한 BLB 는 해당 장치를 담당할 ICS 를 매핑하고 CoAP 을 통해 등록 메시지를 전달한다. 담당 ICS 는 Redis 의 client:registration 키에 등록 메시지를 발행한다. 발행된 등록메시지는 client:registration 키를 구독한 응용이나 웹클라이언트에게 전송된다. 이와 동시에 ICS 는 장치와의 DTLS 세션 정보를 공유하여 다른 ICS 가 동기화하도록 한다. CoAP/DTLS 를 위해 Scandium[4]을 사용하였으며 그림 3 의 클래스 다이어그램에서 세션 동기화에 필요한 간략화 된 Scandium DTLS 세션 정보를 확인할 수 있다. 이러한 세션정보는 직렬화 되어 Redis 의 dtsl:session:sync 키를 통해 다른 ICS 들에게 공유된다.

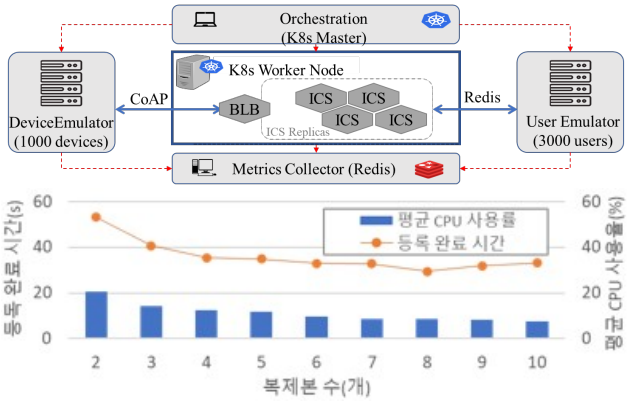

그림 4. 실험 구성 및 결과

실험은 그림 4 와 같이 구성하였다. 에뮬레이터를 통해 1000 개의 장치가 등록하고 3000 명의 사용자가 모든 장치의 등록 메시지를 수신한다. ICS 를 2~10 개까지 확장하면서 등록완료시간 즉, 모든 사용자가 모든 장치의 등록 메시지를 완전히 수신할 때까지의 경과시간과 모든 MS 복제본의 평균 CPU 사용율을 조사하였다. 복제본 수가 늘어날수록 각 복제본의 CPU 사용율이 감소함을 확인할 수 있는데, 등록 메시지가 모든 복제본에게 고르게 분산되고 있음을 의한다. 등록완료 시간도 복제본 수 증가에 따라 단축되어 8 개에서 최소치를 보이다 다시 증가함을 볼 수 있다. 따라서 부하분산이 고르게 된다고 할지라도 복제본 수를 계속 늘리는 것은 무의미 할 수 있으며 서비스의 규모에 따라야 할 것이다. 실험에서는 7~10 개까지는 CPU 사용율이 거의 같지만 등록완료시간이 최소인 8 개가 가장 효율적인 복제본 개수임을 알 수 있다.

Ⅲ. 결론

본 논문에서는 클라우드 LwM2M IoT 서비스의 확장성을 위한 DTLS 로드밸런싱 기술이 제안되었다. 제안 기술은 서버클러스터 뿐만 아니라 가상 컨테이너 복제본에게 효과적으로 LwM2M 트래픽 부하를 분산하고 평문 또는 DTLS 암호문 통신을 지원한다. IP/UDP 더블헤더를 통한 UDP 터널링 기법으로 클라우드 기반 IoT 솔루션의 성능 제고에 기여할 것으로 예상된다.

ACKNOWLEDGMENT

이 논문은 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신산업진흥원의 지원을 받아 수행된 에너지 AI 융합 연구개발 사업임(No. S0255-22-1001)

참 고 문 헌

[1] "Lightweight Machine to Machine Technical Specification", Open Mobile Alliance; OMA-TS-LightweightM2M-V1_0- 20160407-C, 2016.

[2] RFC, Jul. 2020 [Online], “The constrained application protocol (CoAP)”. Available at: https://datatracker.ietf.org/doc/html/rfc7252, vol. 7252.

[3] K. Soleimanzadeh, M. Ahmadi, and M. Nassiri, “SD-WLB: An SDN-aided mechanism for web load balancing based on server statistics,” ETRI Journal, vol. 41, no. 2. Wiley, pp. 197- 206, 21-Jan-2019.

[4] S. Jucker, “Securing the constrained application protocol,” Ph.D. dissertation, Master’s thesis, Department of Computer Science, ETH Zurich, Switzerland, 2012.