

클라우드 기반 네트워크 장비 관리 시스템 설계

정상우*, 신대민**, 서정현**, 이상현*, 이미진*, 김기일*

*충남대학교 컴퓨터공학과

** (주)큐버

e-mail: kikim@cnu.ac.kr

Design of Network Equipment Management System on Cloud

SangWoo Jung*, DanMin Shin**, JeongHyun Seo**, SangHyun Lee*,

MiJin Lee*, Ki-Il Kim*

*Dept of Computer Science and Engineering, Chungnam National University

**QUBER Co., Ltd.

요 약

기존의 Simple Network Management Protocol (SNMP) 기반의 장비관리를 위한 모니터링 시스템은 새로운 장비들의 추가 및 설정에 따른 복잡도가 증가한다. 이러한 문제점을 해결하기 위하여 최근에 Technical Report 069(TR-069) 기반의 시스템을 구축함으로써 여러 가지 접속 방법에 대한 자동 설정을 방안을 제공하고 있다. 하지만, 기존의 방법들 역시 확장성에 대한 문제점을 가지고 있으므로 본 논문에서는 클라우드 기반의 확장성 있는 장비관리 시스템 구조를 제안한다. 제안된 구조는 클라우드 기반의 Auto Configuration Servers(ACS) 서버를 구축함으로써 각 사업자별 Software as a Service(SaaS) 기반의 서비스를 제공한다.

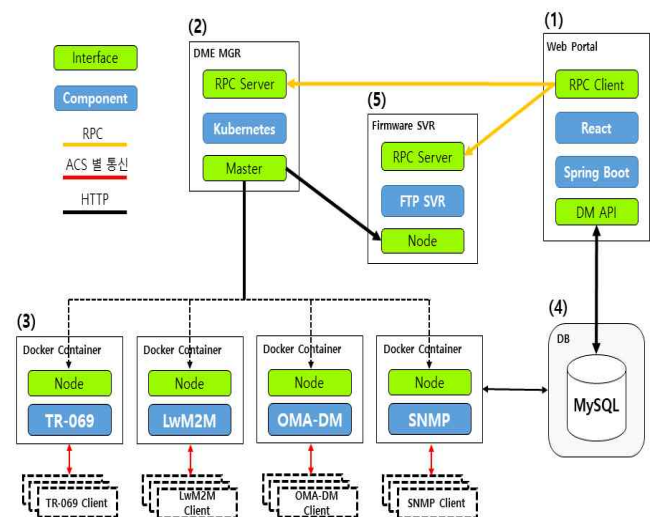
1. 서론

기존의 모니터링 시스템은 기본적으로 SNMP를 기반으로 구성되며 다양한 분야에서 활용되어 왔다. 대표적인 메커니즘으로 SNMP 기반의 전력설비 모니터링 시스템[1]이 제안되었다. 또한, [2]에서는 센서 네트워크를 위한 SNMP 모니터링 방법으로 게이트웨이 기반의 연동방안이 제시되었다. 하지만, 최근의 다양한 기기들의 출현에 따라 기존의 SNMP만을 통한 제어가 불가능하게 되었다. 이를 위하여 새롭게 제안된 것이 바로 TR-069기반의 모니터링 시스템[3-4]이다.

비록 TR-069가 기존의 SNMP의 문제점을 해결할 수는 있지만 해당 기술 또한 새로운 기기들의 추가와 이에 따른 관리를 위한 확장성 문제를 해결하여야 하지만 아직까지 이러한 기능의 제공은 제한적이다. 예를 들어 [3]에서 제안된 시스템의 경우 TR-069 프로토콜의 검증하기 위한 구조가 제안되었으며 [4]에서는 안드로이드 운영체제를 탑재한 디바이스만을 지원한다. 따라서, 확장성을 고려한 시스템 구조의 설계가 반드시 요구된다. 이를 위하여 본 논문에서는 클라우드 기반의 ACS 서버를 구축하는 동시에 각 사업자별 SaaS 기반의 서비스를 제공하기 위한 시스템을 제안함으로써 확장성 문제를 해결하고자 한다.

의 인터페이스로 구성된다.

제안하는 시스템에서 사용자가 언제 어디서든 웹을 통해 관련 장비를 모니터링 하고 관리를 할 수 있도록 가입된 장비의 데이터는 주기적으로 데이터베이스에 전송되고 저장된다. 또한, 관리자의 경우 가입된 장비의 가입량에 따라 ACS를 증설이 가능한 구조를 가지고 있다.



(그림 1) 전체 시스템 구성도

2. 전체 시스템 구조

본 논문에서 제안하는 네트워크 장비 관리 시스템 구조는 (그림 1)과 같으며 다수의 핵심 컴포넌트들과 이들 간

제안된 시스템은 (그림 1)과 같이 다음과 같은 컴포넌트로 구성된다. 각각의 컴포넌트는 1) 실제 사용자에게

관리 및 모니터링 서비스를 제공할 웹 포탈 (Web Portal) 컴포넌트, 2) 웹 포탈에서 내린 명령을 알맞은 도커 (Docker)에 명령하는 도커 관리(Docker Manager, Kubernetes) 컴포넌트, 3)도커관리부에서 내려온 명령을 실제로 수행하고 가입된 장비들의 데이터를 주기적으로 데이터베이스에 저장하는 도커 컨테이너(Docker Container, ACS) 컴포넌트, 4) 장비들의 데이터가 실제로 저장 되고 웹을 통해 값을 보게 해주는 데이터베이스 컴포넌트, 5)가입된 장비들의 펌웨어를 관리하고 배포하는 펌웨어 서버 컴포넌트이며 이들의 기능 설명과 개발을 위한 도구에 대한 설명은 다음과 같다.

2.1 컴포넌트 구성

2.1.1 웹 포탈

관리자 및 사용자는 웹앱을 통해 장비관리 서비스를 이용한다. 관리자는 웹을 통해 운용중인 ACS 컨테이너들을 관리하기 위하여 사용자에게 ACS 및 펌웨어 서버를 할당하는 동시에 최대 가입자 수를 제한 할 수 있다. 사용자는 장비를 가입시키고 데이터베이스에서 주기적으로 데이터를 수신하여 장비의 상태를 웹으로 모니터링 할 수 있다. 제안하는 시스템에서는 운영체제에 종속되지 않고 모니터링하기 위해 Spring Boot와 React를 사용한 반응형 웹앱을 이용한다.

2.1.2 도커 관리

웹앱에서 제공하는 제어, 관리 명령들은 모두 도커 관리부로 전송된다. 해당 명령들은 매니저에서 처리된 후 해당 ACS 컨테이너에게 전송된다. 도커 관리를 위하여 관리자는 ACS 컨테이너의 추가, 삭제, 서비스 정지, 서비스 시작 등을 수행하는 반면 사용자는 할당된 ACS에 장비의 가입, 해제, 펌웨어 배포, 재배포, 롤백 등의 기능을 요청한다. 제안하는 시스템에서 컨테이너를 관리하는 관리 도구는 Kubernetes로 이용하며 RPC 명령을 처리하는 함수들은 C 또는 C++로 구현한다.

2.1.3 도커 컨테이너

ACS 컨테이너는 관리자, 사용자로부터 온 명령들이 실질적으로 수행되는 곳으로 가입된 장비의 초기 셋업, 펌웨어 및 소프트웨어 관리, 원격 진단, 정보 수집 등의 기능이 수행되고 수집된 정보들을 데이터베이스에 저장한다. 각 프로토콜의 ACS 컨테이너들은 도커 이미지로 관리되며 Kubernetes는 이 이미지화일을 통해 컨테이너를 추가한다. ACS 컨테이너는 TR-069 표준안을 기준으로 Java로 구현되어 있는 FreeACS를 사용한다.

2.1.4 데이터베이스

데이터베이스는 ACS 컨테이너들로부터 장비 정보들을 주기적으로 저장하고 웹앱에게 저장된 데이터 값을 전달한다. 각 표준안들은 제공하는 기능 및 기종에 데이터베이스의 구성요소도 다르게 구성된다. 따라서, 웹 사용자 데이터베이스를 비롯한 모든 데이터베이스는 개별적으로 구

축된다.

2.1.5 펌웨어 서버

펌웨어 서버는 장비들의 펌웨어 업데이트 배포에 사용될 펌웨어 설치 파일들이 저장한다. 따라서, 사용자는 제안하는 시스템의 펌웨어 서버에 파일을 저장하는 동시에 개인 서버의 펌웨어 파일을 URL을 통해 장비들에게 배포한다. 제안하는 시스템의 펌웨어 서버 역시 Kubernetes를 통해 관리되는 도커 컨테이너로 구성하며 펌웨어 서버는 SFTP를 지원한다.

2.2 인터페이스 구성

2.2.1 Remote Procedure Call (RPC)

웹앱은 Kubernetes로 명령을 보내기 위해 RPC를 사용한다. Kubernetes는 RPC 서버의 기능을 담당하기 위하여 특정 기능을 수행하는 함수들을 RPC 서버에 구현하게 되면 RPC 클라이언트인 웹앱에서는 필요한 기능의 함수를 호출한다. 이를 위해서 Kubernetes는 해당 함수를 수행하고 웹앱에게 결과 값을 반환한다. 제안하는 시스템에서는 오픈소스 기반의 Apache Thrift API로 RPC 인터페이스를 구성한다.

2.2.2 ACS와 장비 간의 통신

ACS와 장비간의 통신은 각각의 표준안 별로 사용되는 프로토콜이 서로 다르므로 개발자의 필요에 맞게 변경이 가능하도록 시스템을 구성하는 것이 요구된다. TR-069의 경우 ACS와 장비는 RPC 또는 XMPP로 데이터의 송수신이 가능하며 LwM2M의 경우 CoAP 프로토콜을 이용한다. 따라서, 각각의 장비에 사용하는 표준안에 따라 해당 통신 모듈들을 모두 탑재하여야 한다. <표 1>은 각 표준안 별 통신 제원을 비교한다.

<표 1> 각 표준안 별 통신 메커니즘 비교

	TR-069	OMA-DM	LwM2M
전송 기술	XMPP/RPC	HTTP	CoAP
장치 유형	셋톱박스 게이트웨이	모바일단말기 게이트웨이	센서 노드
Payload	XML	XML	JSON
보안 방식	TLS	TLS	DTLS
API 방식	SOAP	REST	REST

4. 결론

본 논문에서는 각기 다른 관리 프로토콜을 사용하는 네트워크 장비들을 관리하기 위한 확장성 높은 시스템 구조를 설명하였다. 제안된 시스템에서 관리자는 각기 다른 프로토콜의 ACS 증설을 위하여 도커와 Kubernetes를 사용한다. 이에 따라 추가적으로 발생하는 비용을 줄여주는 동시에 웹앱을 통해 관리자 및 사용자의 이용 편의성을 높일 수 있을 것으로 기대된다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학사업의 연구결과로 수행되었음 (2015-0-00930)

참고문헌

- [1] 장경배, 김종민, “SNMP를 이용한 전력설비 모니터링 시스템 연구,” 한국조명·전기설비학회 춘계학술대회 논문집, 2017년 5월.
- [2] 조성훈, 이국형, 김영곤, “SNMP 기반의 무선 센서 네트워크 모니터링 시스템 설계,” 한국통신학회 추계종합학술발표회 논문집, 2017년 11월.
- [3] Norbert Nemet, Sasa Radovanovic, Rade Simikic, Istvan Papp, Teodora Novkovic, “Adaptive TR-069 system service for Android-based consumer electronic devices,” IEEE International Conference on Consumer Electronics, Sep. 2013.
- [4] Ivana Savic, Milan S. Savic, Gordana Velikic, “Implementation of TR-069 connection request mechanism,” International Symposium on Industrial Electronics, Nov. 2014.