

멀티코어 임베디드 스위치를 활용한 SDN 테스트베드 구축

김성환^{*}, 안남원, 임 혁

광주과학기술원 정보통신공학부

Construction of SDN Testbed using Multi-core Embedded Switches

Sunghwan Kim, Namwon An, Hyuk Lim

School of Information and Communications, Gwangju Institute of Science and Technology

요 약

Internet of things (IoT) 및 cyber-physical system (CPS) 환경에서 임베디드 장치는 인터넷과 연결되어 다양한 응용 및 서비스 제공을 위하여 사용되고 있다. 이러한 임베디드 장치는 기존의 전통적 임베디드 시스템의 활용범위를 벗어나 보다 많은 양의 데이터 처리뿐만 아니라, 다양한 동작 및 서비스를 동시에 수행할 수 있어야 한다. 본 논문에서는 software-defined networking (SDN) 을 위한 OpenFlow 를 지원하고, 하드웨어를 가상화시켜 다양한 제어/서비스를 virtual machine (VM) 에서 실행할 수 있는 멀티코어 임베디드 SDN 스위치를 개발하였다. 개발된 멀티코어 임베디드 SDN 스위치는 리눅스 기반의 멀티코어 임베디드 보드, 오픈소스 소프트웨어 스위치 Open vSwitch (OVS) 와 Kernel-based Virtual Machine (KVM) 하이퍼바이저로 구성되어 있다. 개발한 스위치들을 SDN 컨트롤러 OpenDaylight 과 연동하여 SDN 테스트베드 네트워크를 구축하고, 동시에 여러 VM 을 구동시킬 수 있다. 가상화 환경에서 CPU 부하에 따른 네트워크 처리 변동량을 측정함으로써 임베디드 스위치의 네트워크 성능 평가를 수행하였다.

1. 서 론

인간과 사물, 사물과 사물이 네트워킹을 이루는 사물인터넷 (IoT: Internet of things) 과 사이버물리시스템 (CPS: cyber-physical system) 은 인류사회문명 전반에 걸쳐서 새로운 정보 혁명을 만들어가고 있다. 미래 사회는 모든 사물이 네트워크에 연결되어 센서에서 수집한 센싱 정보를 클라우드 기반 데이터센터로 전송하고, 빅데이터 기술로 계산된 제어/서비스 데이터를 전송 받아 다른 사물들과 데이터를 교환하는 정보시스템으로 구현되어가고 있다. 이러한 IoT, CPS 환경에는 무수히 많은 종류의 사물들이 존재하고, 그만큼 방대한 양의 데이터를 네트워크 상에서 주고받아야 한다. 하지만 기존 네트워크 구조에서는 Command Line Interface (CLI) 를 통해 네트워크 정책들과 프로토콜 설정을 수행해야 하는 경직성, 하드웨어 의존성 및 네트워크에 연결된 장치들의 증가로 인한 라우팅 분산처리의 오버헤드 등의 한계가 있다.

IoT, CPS 환경에서 임베디드 장치로 이루어진 사물들로부터 수집된 데이터를 처리하기 위해서 네트워크를 직접 프로그래밍할 수 있고 수많은 네트워크 장비들을 중앙 집중화된 컨트롤러에서 관리 및 제어할 수 있는 소프트웨어 정의 네트워킹 (SDN: software defined networking) 기술이 적용되고 있다. SDN 은 제어평면을 중앙 집권화하고 데이터평면과 분리 시켜 네트워크 트래픽의 제어를 중앙의 SDN 컨트롤러에서 소프트웨어로 할 수 있게 하여, 하드

웨어의 제약 없이 전체 네트워크를 효율적으로 관리할 수 있게 해준다. 이러한 방식은 기존 벤더 중심의 네트워크 환경을 탈피하여 보다 동적인 환경에서 유연하게 네트워크 기반 서비스를 제공할 수 있도록 한다. 또한, IoT, CPS 환경에서 임베디드 장치들은 다양한 응용 및 서비스 제공을 위하여 기존의 전통적 임베디드 시스템을 벗어나 보다 많은 양의 데이터 처리뿐만 아니라, 다양한 동작 및 서비스를 동시에 수행할 수 있어야 할 필요성이 있다.

본 논문에서 멀티코어 임베디드 보드를 이용해 SDN 테스트베드 네트워크를 구성하는 것을 제안한다. 제안된 시스템을 활용하면, 임베디드 장치로 이루어진 사물의 센서로부터 수집된 데이터를 곧바로 원격지 데이터센터로 전송하지 않고, 필요에 따라 virtual machine (VM)을 이용하여 임베디드 장치에서 데이터 압축, 패킷 보안성 검사 등의 기능을 수행한 후 전송하는 것이 가능하게 된다. 즉, 서비스 및 자원의 필요에 따라 임베디드 스위치를 가상화시켜 VM 을 통해 다양한 동작을 SDN 네트워킹과 함께 수행하는 방식의 IoT 네트워킹을 구현하게 할 수 있다.

2. 멀티코어 임베디드 SDN 스위치

본 논문에서는 리눅스 기반의 멀티코어 임베디드 보드 컴퓨터 ODROID-XU3 [1] 와 오픈소스 프로젝트 Open vSwitch (OVS) [2] 와 하이퍼바이저 Kernel-based Virtual Machine (KVM) [3] 을 이용하여 스위치

하드웨어를 가상화 시켜 다양한 제어/서비스를 VM에서 실행할 수 있는 OpenFlow 지원 멀티코어 임베디드 SDN 스위치를 개발하였다. 그림 1은 멀티코어 임베디드 SDN 스위치의 구조이다. 8개의 코어를 가진 멀티코어 임베디드 보드인 ODROID-XU3에 OVS 2.0.2 버전을 설치하여 OpenFlow 지원 가능한 SDN 스위치로 만들고, 데이터 평면을 위한 네트워크 인터페이스로 USB3.0 허브 1개를 보드에 연결하고 각 허브에 4개의 USB3.0 to Gigabits Ethernet을 연결하였다. 또한, KVM을 이용하여 ODROID-XU3가 필요에 따라 다수의 VM을 생성할 수 있도록 하였다. ODROID-XU3의 host OS로는 Ubuntu 14.04 LTS 버전을 사용하였고 커널은 odroidxu3 - 3.10.82 버전을 사용하였다.

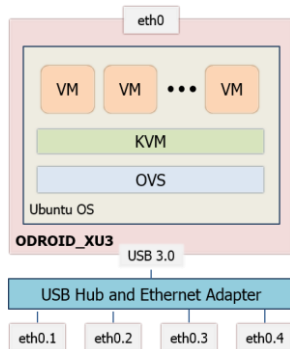


그림 1 멀티코어 임베디드 SDN 스위치 구조

3. SDN 테스트베드

3.1. 테스트베드 구성

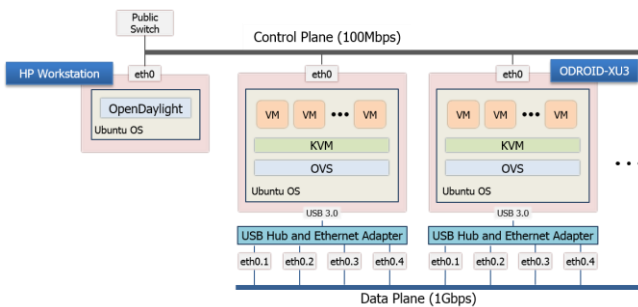


그림 2 SDN 테스트베드 네트워크 구성

개발한 멀티코어 임베디드 SDN 스위치들을 오픈 소스 SDN 컨트롤러인 OpenDaylight [4]과 연동하여 SDN 테스트베드를 구성하였다. OpenDaylight은 오픈 소스 기반의 SDN 컨트롤러 개발 프로젝트로써 기존 Floodlight, NOX, ONOS 등의 컨트롤러와 다르게 OpenFlow 1.3 버전외에도 BGP, SNMP, LISP, NETCONF 등의 다양한 southbound interface (SBI)를 지원한다. 또한 클라우드 컴퓨팅 솔루션 Openstack의 네트워킹 부분을 담당하는 Neutron과의 연동을 통해 네트워크 기능 가상화 (NFV: network function virtualization)를 지원한다. 그림 2는 멀티코어 임베디드 SDN 스위치를 활용하여 구축한 SDN 테스트베드의 구성이다. 멀티코어 임베디드 SDN 스위치의 인터페이스들과 KVM에 의해 생성된 VM의 가상

네트워크 인터페이스들은 OVS를 통해 만든 가상 브리지에 연결하여 데이터평면을 구성하였다. 보드에 내장된 10/100Mbps Ethernet 네트워크 인터페이스는 SDN 컨트롤러로 사용한 OpenDaylight “Helium” 버전과 연결하여 제어평면을 구성하였다.

3.2. 테스트베드 성능 평가

SDN 테스트베드의 성능 평가를 위해 각각의 멀티코어 임베디드 SDN 스위치의 CPU 부하에 따른, 네트워크 처리 변동량을 측정해 보았다. CPU 부하 및 모니터링을 위해서는 stress와 top 소프트웨어를 사용하였고, 네트워크 처리 변동량은 iperf 소프트웨어를 통해 트래픽을 생성하여 측정하였다. 그림 3은 CPU 부하에 따른 네트워크 처리 변동량을 측정하는 결과이다. 측정 결과 CPU 부하가 증가하여도, 제어평면과 데이터 평면 모두 네트워크 처리량이 저하되지 않음을 확인할 수 있다.

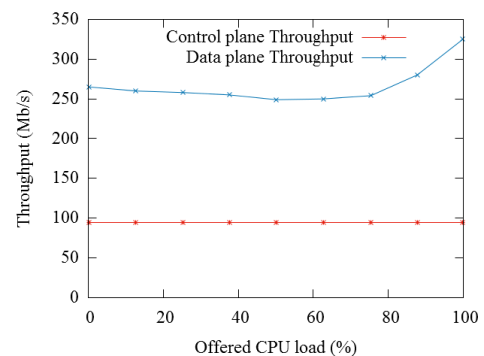


그림 3 CPU 부하에 따른 네트워크 처리 변동량

4. 결론

본 논문에서 제안한 멀티코어 임베디드 SDN 스위치를 활용한 SDN 테스트베드 네트워크는 Flow Table 생성 및 제거, 포트 상태 제어 등의 OpenFlow 기능들을 수행하여 SDN을 테스트해볼 수 있고, 임베디드 보드를 가상화하여 VM을 이용한 다양한 제어/서비스도 동시에 수행할 수 있는 장점이 있다. 또한, VM 부하에 따른 네트워크 성능 평가를 통해 CPU 부하에도 네트워크 처리량이 감소하지 않는 결과를 확인할 수 있었다.

참고문헌

- [1] ODROID-XU3 website, <http://www.hardkernel.com>
- [2] Open vSwitch website, <http://openvswitch.org>
- [3] KVM website, <http://www.linux-kvm.org>
- [4] OpenDaylight website, <https://www.opendaylight.org>

※ 본 연구는 미래창조과학부 재원으로 정보통신기술연구진흥센터 정보통신방송연구개발사업(B0101-15-0557, 고신뢰 CPS 연구센터) 및 한국연구재단 중견연구지원사업(핵심)의 지원을 받아 수행된 연구임 (2014R1A2A2A01006002).