

OpenFlow 기반 SDN 환경에서 다중 플로우 테이블을 이용한 향상된 패킷 처리 기법 성능 분석

김경식, 김철민, 박진호, 김근수, 고석주*

경북대학교 컴퓨터학부

kyungsik850@knu.ac.kr, cheolminkim@vanilet.pe.kr, jinhopark@knu.ac.kr,

kks36145919@gmail.com, *sjkoh@knu.ac.kr

A Study on the enhanced performance for packet processing using the multiple flow tables in the OpenFlow based SDN environments

Kyung-Sik Kim, Cheol-Min Kim, Jin-Ho Park, Keun-Soo Kim, Seok-Joo Koh*

School of Computer Science and Engineering, Kyungpook National University, Daegu, Korea

요 약

SDN (Software-Defined Networking)은 소프트웨어를 통해 데이터 전달과 네트워크 제어를 논리적으로 구현한 기술로, 기존 하드웨어 기반 네트워크 환경의 문제점이었던 확장성, 유연성, 비용 문제를 해결하기 위해 활발히 도입되고 있는 기술이다. SDN 장치 간 규격화 된 통신을 위해 ONF (Open Networking Foundation)는 OpenFlow 프로토콜을 정의하였으며 현재 널리 사용되고 있다. OpenFlow 프로토콜은 SDN 장치가 패킷을 처리하는 과정에서 플로우 테이블을 사용한다. 플로우 테이블은 패킷을 처리하기 위한 규칙이 정의되어 있으며, 전송되는 패킷은 OpenFlow 스위치에서 플로우 테이블의 규칙에 따라 처리된다. 이때, 여러 규칙을 하나의 플로우 테이블로 구성하는 경우 패킷 처리를 단일 경로로 수행하게 되어 성능이 떨어지는 문제점이 있다. 이를 해결하기 위해 OpenFlow 버전 1.3에서 파이프라인 프로세싱(pipeline processing)이 도입되어 규칙을 여러 플로우 테이블로 분산하여 패킷을 병렬 구조로 처리할 수 있게 되었다. 본 논문은 다중 플로우 테이블을 이용한 패킷 처리 성능과 단일 플로우 테이블을 이용한 패킷 처리 성능의 차이를 비교하기 위해 TCP 패킷 처리 실험을 진행하였으며, 다중 플로우 테이블을 이용한 환경이 단일 플로우 테이블보다 패킷 전송량 및 처리량이 약 8% 향상됨을 확인했다.

I. 서론

SDN (Software-Defined Networking)은 데이터 전송 역할을 하는 데이터 평면(data plane)과 제어 역할을 하는 제어 평면(control plane)이 결합된 하드웨어 장비 기반의 기존 네트워크 환경에서 소프트웨어를 통해 데이터 평면과 제어 평면을 논리적으로 분리하여 하드웨어와 독립적으로 관리할 수 있도록 새롭게 제안된 기술이다. SDN을 사용하지 않는 기존 환경은 새로운 네트워크 요구사항이나 급증하는 트래픽을 처리하기 위한 새로운 장비를 구입하고, 설치 및 구성하는 과정이 필요하다. 이로 인해 확장성, 유연성 및 비용 문제가 발생하며 신속한 대응이 어려운 문제가 발생한다. 이러한 문제를 해결하기 위해 SDN 연구가 시작되어, 네트워크 요구사항이 급격히 변하는 클라우드 컴퓨팅, 5G 등의 분야에서 사용되고 있다. SDN 환경에서 데이터 전달은 규격화된 범용 장비를 사용하여 수행되며, 트래픽 제어 및 네트워크 관리 등의 실제 네트워킹 기능은 소프트웨어 기반의 SDN 컨트롤러를 통해 제공된다 [1].

SDN 환경에서 제어 평면을 담당하는 네트워크 컨트롤러는 OpenFlow 스위치의 패킷 처리 방식 및 전달 경로를 정의하며, OpenFlow 프로토콜의 플로우 테이블(flow table)을 사용한다. 플로우 테이블에 정의되는 항목은 규칙(rule), 동작(action), 통계(stats)로 구성되어 있으며, OpenFlow 스위치가 송/수신하는 패킷은 자신의 플로우 테이블에 정의된 규칙에 따라 패킷을 처리한다 [2]. 이때, 전송되는 패킷을 다수의 규칙으로 이루어진 단일 플로우 테이블을 통해 처리하는 경우, 테이블 내의 모든 규칙을 확인하고 처리되므로 효율성이 떨어진다. 하지만, 규칙을 여러 테이블로 분산하는 경우, 패킷 처리가 여러 플로우 테이블을 통해 병렬로 수행되어 효율적인 패킷 처리가 가능하다 [3].

본 논문은 OpenFlow 스위치에서 단일 플로우 테이블을 사용한 환경과 다중 플로우 테이블을 사용한 환경을 설명하며, 실험을 통해 두 환경 간의 패킷 처리 속도를 비교 및 분석한다.

II. 본론

그림 1과 같이, SDN 컨트롤러에서 동작하는 소프트웨어인 네트워크 애플리케이션을 통해 플로우 테이블을 OpenFlow 스위치에 전달하여 네트워크를 제어 및 관리한다 [4].

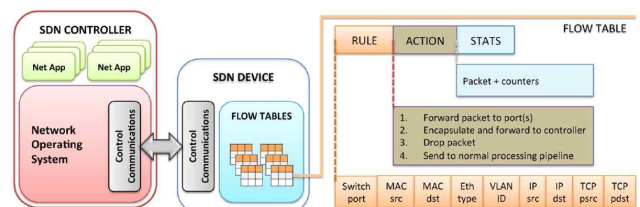


그림 1 OpenFlow 기반 SDN 네트워크 구성 및 Flow Table 구조. 플로우 테이블은 OpenFlow 스위치에 수신된 패킷을 처리하기 위해 플로우를 정의하는 패킷 헤더 정보, 패킷 처리 방법을 표시하는 동작 정보 및 해당 플로우의 통계 정보로 구성된다 [5].

단일 플로우 테이블과 다중 플로우 테이블을 이용한 패킷 처리 성능 분석을 위해, 그림 2와 같이 실험 환경을 구성하였다.

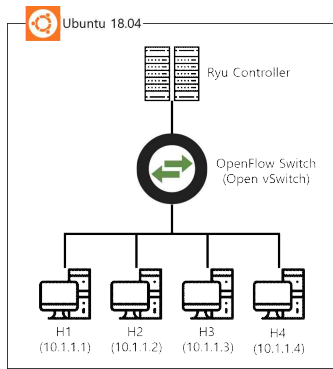


그림 2 실험 환경

실험은 Ubuntu 18.04가 설치된 PC에 네트워크 컨트롤러인 Ryu Controller와 SDN 네트워크 환경을 구축하기 위한 Mininet 에뮬레이터를 이용하였다. Mininet 환경에서 OpenFlow 프로토콜 1.3 버전을 지원하는 가상 스위치인 Open vSwitch를 이용하였으며, Mininet의 Host 엔티티를 사용하여 실험을 진행하였다. 네트워크 컨트롤러는 실험을 위한 단일 플로우 테이블과 다중 플로우 테이블 애플리케이션을 구현하였으며 이들의 구성은 그림 3 및 그림 4와 같다.

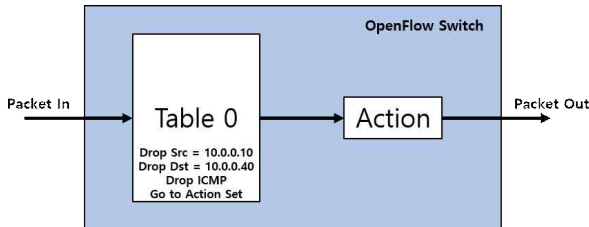


그림 3 단일 플로우 테이블 실험 구성

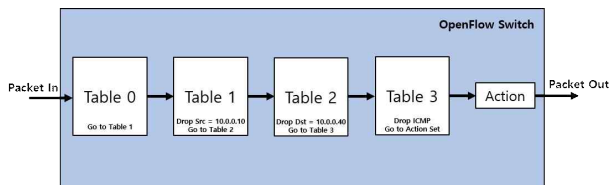


그림 4 다중 플로우 테이블 실험 구성

단일 플로우 테이블은 OpenFlow 스위치로 들어온 패킷의 출발지 IP가 10.0.0.10, 목적지 IP가 10.0.0.40, 상위 프로토콜이 ICMP인 경우 패킷을 버리는(drop) 규칙을 구성하였다. 다중 플로우 테이블은 단일 플로우 테이블의 조건과 동일한 규칙을 세 개의 테이블로 나눠서 구성하였다. 단일 플로우 테이블을 사용하는 통신과 다중 플로우 테이블을 사용하는 통신의 속도 차이를 비교하기 위해 호스트 1 (10.1.1.1)에서 호스트 4 (10.1.1.4)로 TCP 패킷을 10초 동안 전송하는 실험을 10회 진행하였으며, 그림 5 및 그림 6은 그 결과이다.

```
Client connecting to 10.1.1.4, TCP port 5001
TCP window size: 85,3 KByte (default)
-----
[ 17] local 10.1.1.1 port 49820 connected with 10.1.1.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 17] 0.0-10.0 sec    23.2 GBytes 19.9 Gbits/sec
```

그림 5 단일 플로우 테이블 환경의 성능 측정 결과

```
Client connecting to 10.1.1.4, TCP port 5001
TCP window size: 85,3 KByte (default)
-----
[ 17] local 10.1.1.1 port 47456 connected with 10.1.1.4 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 17] 0.0-10.0 sec    25.4 GBytes 21.9 Gbits/sec
```

그림 6 다중 플로우 테이블 환경의 성능 측정 결과

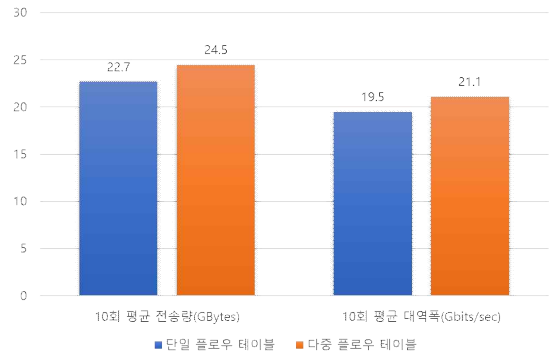


그림 7 단일, 다중 플로우 테이블 간 패킷 전송량 및 대역폭

그림 7은 10회 실험에서 환경별 전송량과 대역폭의 평균을 나타낸 그래프이다. 패킷 전송량은 단일 플로우 테이블을 사용했을 때 22.7GB, 다중 플로우 테이블을 사용했을 때 24.5GB로 약 8%(1.8GB)차이를 보였으며, 대역폭은 단일 플로우 테이블을 사용했을 때 19.5Gbps, 다중 플로우 테이블을 사용했을 때 21.1Gbps로 약 8%(1.6Gbps)의 차이를 보였다. 실험 결과, 다중 플로우 테이블을 사용하여 패킷을 처리하는 것이 더 우수한 패킷 처리량을 보여주는 것을 확인하였다.

III. 결론

본 논문은 OpenFlow 기반 SDN 환경에서 다중 플로우 테이블을 이용한 패킷 처리 성능 개선 방안을 제안하고 실험을 통해 성능 분석을 진행하였다. OpenFlow의 파이프라인 프로세싱을 이용한 다중 플로우 테이블 기능은 단일 플로우 테이블보다 패킷 평균 전송량 및 평균 대역폭이 약 8% 향상되는 것을 확인했다.

본 연구를 기반으로 향후 OpenFlow 스위치에서 플로우 테이블 내의 규칙 비교 및 동작 처리 속도를 개선하는 방안을 연구하여 본 연구에서 제안한 방법과 성능 비교를 할 예정이다.

ACKNOWLEDGMENT

이 연구는 2021년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임(20002214, 20009633).

참고 문헌

- [1] Yoo, J., Kim, W. & Youn, C. "A Technical Trend and Prospect of Software Defined Network and OpenFlow," KNOM Review, Vol. 15, No. 2, pp. 1-24, Dec. 2012
- [2] Kreutz D. et al., "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, Vol. 103, No. 1, pp. 14-76, Jan. 2015
- [3] ONF (Open Networking Foundation), "The Benefits of Multiple Flow Tables and TTPs," Feb. 2015
- [4] Lee, S., "SDN 관련 연구의 간략한 역사 및 다양한 연구 주제들," 정보과학회지, Vol. 31, No. 9, pp. 46-52, Sep. 2013
- [5] ONF (Open Networking Foundation), "OpenFlow Switch Specification, Version 1.5.1," Mar. 2015