

SDN 에서 확장성 있는 메시지 플로우 제어 메커니즘

염철원*, 김강석*, 예홍진*

*아주대학교 대학원 지식정보공학과

e-mail : {yeom, kangskim, hjyeh}@ajou.ac.kr

Scalable Message Flow Control Mechanism in SDN

Cheolwon Yeom*, Kangseok Kim*, Hongjin Yeh*

*Dept. of Knowledge Information Engineering, Graduate School of Ajou University

요 약

SDN(Software Defined Networking)은 프로그래밍을 통해 네트워크 경로 설정과 제어 및 기존에 복잡했던 운용관리를 처리할 수 있게 해주는 기술이다. 하지만 중앙 집중 식으로 관리하는 특성에 따라 병목현상이 발생할 수 있으며, 이는 컨트롤러에 트래픽이 집중되고 제어 명령을 지연시키는 문제가 있다. 본 논문은 이러한 문제점을 개선하고자 확장성 있는 메시지 플로우 제어 메커니즘을 제안한다. 이 방법은 기존 스위치와 다르게 목적지만을 저장하는 Table 을 별도로 관리하여 동일한 목적지로 전송되는 패킷에 대해 반복되는 제어 메시지를 감소시킨다. 이를 통해 컨트롤러의 역할을 스위치로 분산하여 Control Plane 트래픽에 대한 부하를 줄일 수 있고, 패킷 전송 Delay, Flow Setup 동안 발생하는 Latency 의 문제점을 효율적으로 개선 할 수 있다.

1. 서론

최근 인터넷 사용이 증가하게 되면서 기존 인터넷에 내재된 구조적인 문제점이 드러나게 되었다. 이를 해결하기 위해, 다양한 미래의 응용서비스들을 빠르고 안전하게 사용할 수 있도록 하는 미래 지향적 네트워크 및 인프라 구축이 요구되고 있으며, 이에 따라 네트워크의 개방화와 가상화를 위해 SDN 기술의 연구개발에 초점이 맞춰지고 있다 [1].

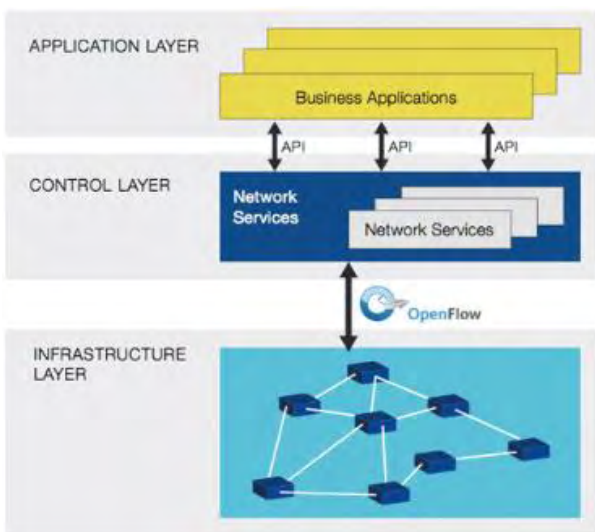
SDN(Software Defined Networking)은 네트워크 설정을 소프트웨어적으로 제어할 수 있도록 하는 기술이다. 기존의 네트워크 기술은 분산된 하드웨어에 제어 기능이 개별 관리되는 비유연성 구조를 기반으로 하는 반면 SDN 은 제어 기능을 별도로 분리하여 중앙 집중식으로 관리되어 유연한 네트워크를 제공한다[2]. 이를 통해 고품질의 서비스 제공이 가능해 지고 네트워크 자원 관리 및 통제를 가능하게 함으로써 변화하는 상황에 빠른 대응을 할 수 있게 된다.

즉, 자신의 요구 사항에 따라 통신망을 소프트웨어 기반으로 쉽게 제어 및 관리를 할 수 있고 토폴로지, 트래픽 전달 경로 등을 차별화하는 서비스를 개발하여 쉽게 적용할 수 있다[3].

OpenFlow 는 SDN 구조 네트워크의 표준 인터페이스이다. 구성 요소로는 컨트롤러와 스위치가 있고, 스위치 내부에는 Flow Table 이 있으며, OpenFlow 프로토콜에 의해 상호 연결되어 실시간으로 트래픽의 경로나 동작을 정의한다[4].

앞서 제시된 다양한 이점에도 불구하고, SDN 의 중앙 집중적인 구조는 스위치에서 발생하는 제어 트래픽이 컨트롤러에 집중되어 부하가 커지게 되며 병목현상을 야기한다[5][6][7][8]. 이러한 문제점은 전체적인 네트워크 성능에 영향을 미치게 되며, 컨트롤러를 통해 경로를 전달 받는 시간이 지연될 수 있다.

본 논문에서는 해당 문제점을 개선하고자 스위치에 Destination Table 을 두어 관리하고 스위치와 컨트롤러 간의 제어 메시지 발생을 감소시키는 방법에 대해 제안한다.



(그림 1) SDN 아키텍처

본 연구는 미래창조과학부 및 한국인터넷진흥원의 "고용계약형 지식정보보안 석사과정 지원사업"의 연구결과로 수행되었음. (과제번호 H2101-13-1001)

Destination Table 은 기존에 Flow Table 과는 다르게 목적지 정보만을 저장하게 되며 동일한 목적지를 갖는 패킷에 대해 스위치에서 컨트롤러에 집중되는 제어 메시지를 감소시키는 역할을 하게 된다.

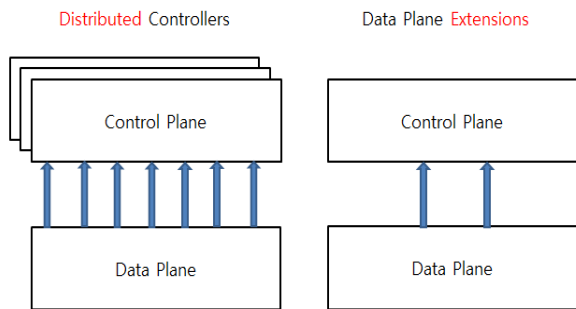
이를 통해 스위치와 컨트롤러간의 메시지 교환으로 인한 병목현상 완화에 효과를 줄 수 있으며, 컨트롤러의 역할을 스위치에서 대신 해줌으로써 확장성을 개선 할 수 있다.

논문의 구성은 다음과 같다. 2장은 관련연구를 소개하고 3장에서는 Destination Table을 활용한 제안방법에 대해 서술하며, 4장에서는 결론 및 향후 연구에 대해 검토한다.

2. 관련연구

OpenFlow 의 중앙집중식 방식은 각 Flow 의 첫 번째 패킷이 캡슐화되어 컨트롤러로 전송된 후 컨트롤러가 새로운 Flow Entry 를 생성해야 된다. 하지만 이와 같은 과정을 모든 패킷 마다 적용하게 되면 잦은 Flow Setup 으로 오버헤드가 발생하게 되며, 네트워크의 규모가 커질수록 확장성에 문제가 생긴다.

OpenFlow 의 확장성 문제를 해결하기 위한 방법으로 (그림 2)와 같은 두 가지 방법이 있다.



(그림 2) 확장성 이슈를 해결하기 위한 방법

2.1. Distributed Controllers

[5]는 OpenFlow 의 분산 컨트롤러를 활용하여 확장성 이슈를 해결하고자 한다. 다수의 컨트롤러는 물리적으로 분산된 구조를 갖지만 논리적으로는 중앙제어를 수행한다. 이러한 분산 컨트롤러 방법에서는 하나의 컨트롤러가 아닌 전체의 컨트롤러가 네트워크의 정보를 필요로 하기 때문에 빈번하게 컨트롤 트래픽이 발생하며, 동기화 문제로 인하여 실시간 대응에 어려움이 발생할 수 있다[5].

[6]의 경우 계층적 컨트롤러 구조로 되어 있다. 컨트롤러는 두 가지로 나뉘어 있는데 로컬 컨트롤러와 Routing 을 위해 전체 네트워크 정보가 필요한 루트 컨트롤러가 있다. 해당 계층으로 분류된 컨트롤러는 트래픽에 따라 처리하게 되는 방법이 다르게 적용되며, 컨트롤러의 동작을 기능적으로 분리함으로써 컨트롤 채널의 부하를 줄여주는 역할을 한다[6].

2.2. Data Plane Extensions

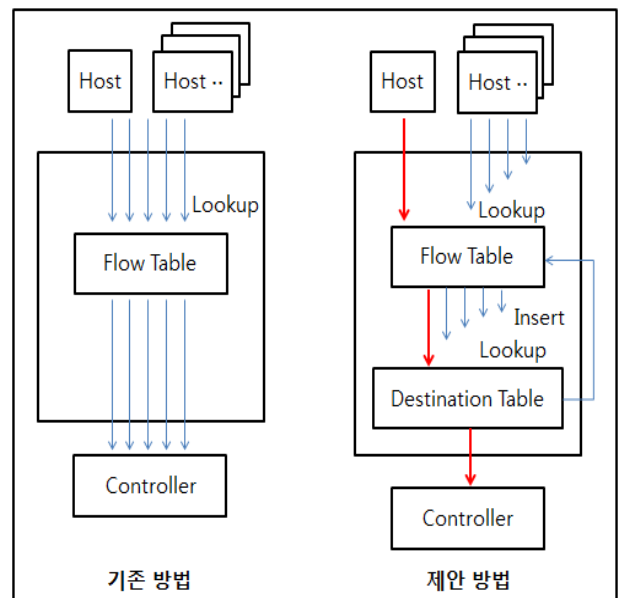
[7]은 스위치 기능과 위치에 따라 제어에 관한 rule

들을 컨트롤러를 통하지 않고도 처리 될 수 있도록 스위치에 따라 Ingress, egress, Authority 스위치로 기능을 나누며 컨트롤러는 전체 rule 을 Authority 스위치에 분산하여 관리한다. 이와 같이 컨트롤러의 역할을 스위치로 분산 시켜 직접 처리할 수 있으므로 컨트롤러로 전달 되는 트래픽을 효율적으로 줄일 수 있다[7].

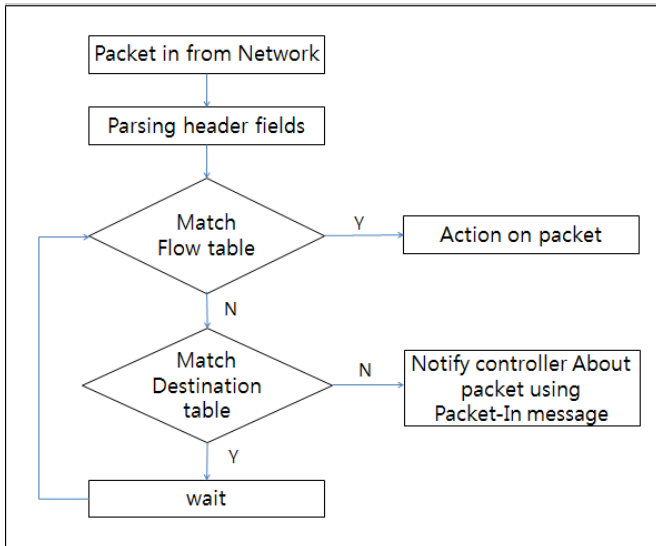
[8]은 컨트롤러에서 제어와 관련된 기능들을 스위치로 분산하여 컨트롤러로 전송되는 제어 메시지를 줄일 수 있다. Flow 의 처리 방법으로는 미리 임계점을 정해 둔 다음 그 임계점이 넘을때만 컨트롤러에 트리거 하여 처리 되는 방식이다. 이는 일정 크기 이상의 플로우만 컨트롤러에서 처리하고 그 외의 경우 Data Plane 내에서 처리하게 됨으로써 컨트롤러에 적은 메시지를 전달하게 된다[8].

3. 제안방법

본 논문에서 제안하는 OpenFlow 스위치의 자세한 동작 방식은 (그림 3)과 같다. 스위치에 패킷이 수신되면 해당 스위치는 패킷의 처리를 위해 ①Flow Table 을 검색하게 된다. Flow Table 에 수신 받은 패킷의 처리 방법이 없다면 스위치는 별도로 생성된 ②Destination Table 에 패킷 목적지를 등록한다. 그리고 ③스위치는 기존 방식과 동일하게 컨트롤러에 Packet-In Message 를 전송한다. 컨트롤러는 해당되는 요청을 확인 후 경로 설정을 위해 ④Packet Out Message 를 스위치에 전송하게 된다. 이때 스위치에 새로운 패킷이 들어오게 되면 앞서 설명한 동작과 같이 Flow Table 을 검색하게 되는데 매칭되는 Flow Entry 가 없을 경우 Destination Table 에 목적지가 일치하는지 확인한다. 목적지가 일치할 경우 이미 처리중인 ⑤Flow Entry 가 생성 될 때까지 대기 상태에 있으며 생성 된 후 설정된 경로에 따라 처리 된다.



(그림 3) 같은 목적지를 갖는 패킷 처리 방법



(그림 4) 제안하는 OpenFlow 스위치의 플로우 처리

제안방법을 적용하게 되면 같은 목적지로 다수의 사용자가 동시 접속 하는 경우에 반복되는 Packet-In Message 를 제어할 수 있다. 만약 동일한 경로의 트래픽이 높아 질 경우 전체 네트워크를 관리하는 컨트롤러는 해당 스위치에서 Flow Statistics 정보를 통해 상태를 확인 할 수 있고 Flow Entry 를 업데이트 하기 때문에 대체 경로를 통하여 목적지로 전송 할 수 있다. 이를 통해 패킷 전송의 Delay, Flow Setup 동안 발생하는 Latency 의 문제점을 효율적으로 개선 할 수 있다.

4. 결론 및 향후 연구

본 논문에서는 최근 활발하게 연구되고 있는 SDN 에 대해 소개하고, 관련 연구에 대해 알아 보았다. 또한 OpenFlow의 확장성 문제를 해결하기 위한 방법으로 기존 스위치에 Destination Table을 추가하여 컨트롤러의 역할을 스위치로 분산하는 방법을 제안하였다.

이는 동일한 목적지로 전송되는 패킷에 대해 반복되는 제어 메시지를 감소시키는 역할을 하며 컨트롤 채널 트래픽에 대한 부담을 줄일 수 있고, 패킷 전송 Delay, Flow Setup 동안 발생하는 Latency의 문제점을 효율적으로 개선 할 수 있다.

향후 연구로는 해당 이슈에 대해 병목현상 완화 및 오버헤드 감소 방법에 대해 연구를 해볼 계획이며, 더 나아가 시뮬레이션을 통해 제안하는 방안을 검증 할 것이고 네트워크에 적용하여 확장시키고자 한다.

참고문헌

[1] 이승익, 이종화, 신명기, 김형준, 손승원 “스마트 인터넷을 위한 SDN 및 NFV 표준기술 동향분석”, 전자통신동향분석 제 29 권 제 2 호, 2014.
 [2] “Software-Defined Networking: The New Norm for Networks,” ONF white paper, 2012.
 [3] 강세훈, 김영화, 양선희 “SDN 핵심 기술 및 진화 전망 분석”, 한국통신학회지 제 30 권 제 3 호, 2013.

[4] N.McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” SIGCOMM CCR, Vol.38, No.2, pp.69-74, 2008.
 [5] A. Tootoonchan and Y. Ganjali, “HyperFlow: a distributed control plane for OpenFlow,” in Proceedings of the 2010 internet network management conference on Research on enterprise networking, ser. INM/WREN’10. Berkeley, CA, USA: USENIX Association, 2010.
 [6] S. H. Yeganeh and Y. Ganjali, “kandoo: A Framework for Efficient and Scalable Offloading of Control Applications,” SIGCOMM HotSDN, pp.19-24, 2012.
 [7] M. YU, J. Rexford, M. J. Freedman, and J. Wang, “Scalable Flow-Based Networking with DIFANE,” SIGCOMM, 2010.
 [8] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, “DevoFlow: scaling flow management for high-performance networks,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 254-265, Aug. 2011.