

OpenFlow 기반 네트워크에서 링크 상태에 따른 동적 플로우 관리 메커니즘

문승일^o, 홍충선^{*}
경희대학교 컴퓨터공학과
{moons85, cshong}@khu.ac.kr

Dynamic Flow Management Mechanism in accordance with Link state in OpenFlow based Network

Seung Il Moon^o and ChoongSeonHong^{*}
Department. of Computer Engineering, Kyung Hee University

요 약

오늘날 요구되는 다양한 서비스의 요구조건을 만족하기에는 기존의 인터넷은 장비와 인프라의 폐쇄적인 구조라는 태생적인 문제로 인하여 그 한계가 있다. 이러한 문제를 해결하기 위해 네트워크를 제어 평면(control plane)과 데이터 평면(data plane)으로 나누어 관리하는 소프트웨어 정의 네트워크(SDN : Software Defined Network)라는 차세대 네트워킹 기술을 통해 해결하려는 연구가 지속적으로 진행되고 있는 추세이다. SDN을 통해 네트워크를 제어 평면(control plane)과 데이터 평면(data plane)으로 나누어 관리함으로써 네트워크 운영자로 하여금 여러 상황에 맞게 네트워크를 제어할 수 있는 환경을 제공할 수 있고 이를 통해 다양한 요구사항을 반영함으로써 보다 유연하게 네트워크를 제어할 수 있게 해준다. 본 논문에서는 이러한 SDN 기술중 하나인 OpenFlow를 이용하여 네트워크를 제어하는 환경에서 보다 효율적인 네트워크 자원 관리를 위하여 링크 상태에 따라 동적으로 Flow를 관리하는 메커니즘을 제안한다.

1. 서 론

오늘날 인터넷은 장비와 인프라의 폐쇄적인 구조라는 태생적인 문제를 갖고 있어 요구되는 다양한 서비스를 제공하기 위해 관리자로 하여금 다양한 상황에 맞게 네트워크를 제어하기에는 그 한계가 존재한다. 이러한 문제를 해결하기 위한 방안 중 하나로 OpenFlow[1][2]가 제안되었으며 이를 활용한 다양한 연구가 진행되고 있다.

OpenFlow는 소프트웨어 정의 네트워크[3] (SDN : Software Defined Network)의 기술 중 하나로 네트워크를 제어 평면(control plane)과 데이터 평면(data plane)으로 나누어 관리하고 OpenFlow 프로토콜[4]이라는 개방형 프로토콜을 제공함으로써 네트워크 운영자로 하여금 여러 상황에 맞게 네트워크를 보다 유연하게 제어할 수 있는 프로그래밍 가능한 네트워크 환경을 제공해준다.

본 논문에서는 OpenFlow 기반 네트워크에서 효율적인

Flow관리를 위하여 링크상태에 따라 동적으로 Flow를 관리하는 메커니즘을 제안하였다.

제안하는 Flow 관리 메커니즘은 다음과 같은 과정을 통해 수행된다. 먼저 관리자의 요청에 의하여 Flow를 할당 할 때 각 링크에 할당된 Flow의 요구대역폭에 따라 Flow를 할당한다. 그 후 지속적으로 링크상태를 모니터링 하다가 이전에 할당된 Flow를 통해 전송되는 트래픽의 양이 요구대역폭을 넘어서게 되어 QoS를 보장할 수 없게 된다면 해당 Flow보다 요구대역폭을 증가 시킨 새로운 Flow를 생성하여 할당하고 이전 Flow는 삭제함으로써 해당 서비스 Flow의 QoS를 보장하는 한편 요구 대역폭 이상의 트래픽을 전송하는 Flow 때문에 같은 링크 상에 할당되어 있던 다른 서비스 Flow의 QoS를 보장할 수 없게 되는 상황을 방지할 수 있게 된다.

2. OpenFlow

OpenFlow는 서로 다른 스위치와 라우터의 플로우 테이블을 프로그래밍 할 수 있는 오픈 프로토콜을 제공하는

“본 논문은 한국정보화진흥원(NIA) 지원으로 수행된 연구결과임” Dr. CS Hong is the corresponding author.

다. 아래의 그림 1은 OpenFlow 스위치의 개념도를 보여 주고 있다. 그림에서 알 수 있듯이 OpenFlow는 FlowTable, Secure Channel, OpenFlow Protocol의 세 개의 파트로 나눌 수 있다. FlowTable은 각 플로우 엔트리와 연관된 동작이 정의된 부분으로 이를 통해 스위치는 플로우에 대한 처리를 할 수 있으며, Secure Channel은 컨트롤러와 스위치 사이에서 명령어나 패킷의 안전성을 보장과 함께 전달 될 수 있도록 도와주는 역할을 한다. 마지막으로 OpenFlow Protocol은 공개되어있는 표준적인 방식을 제공함으로써 스위치와 Allocator가 서로 통신할 수 있도록 한다.

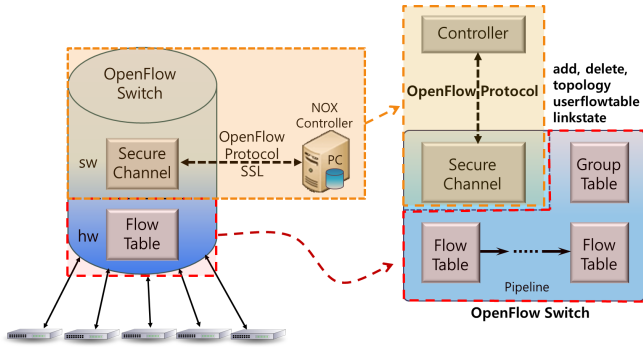


그림 1. OpenFlow 구조

2. 시스템 구조

본 논문에서 제안하는 OpenFlow 기반의 네트워크 관리를 위한 시스템의 구조는 그림 2에서 알 수 있듯이 Path Manager, VN Manager, NOX Controller[5][6]로 구성되어 있다.

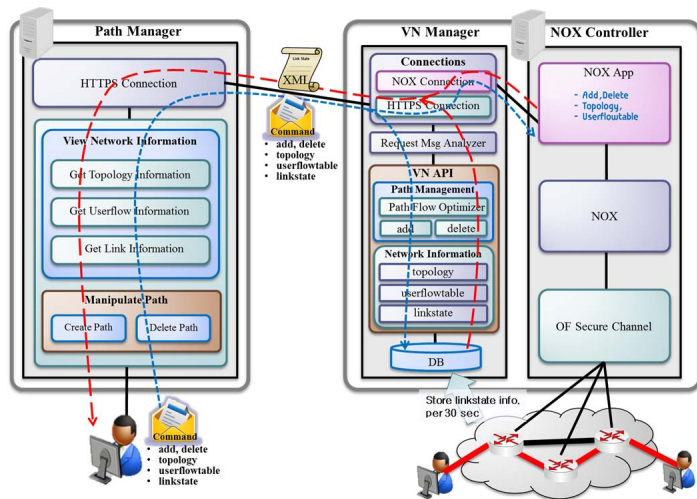


그림 2. 시스템 구조

Path Manager의 하위 모듈로는 View Network Information, Manipulate Path가 있고 HTTPS Connection을 통해 VN Manager와 통신을 한다.

VN Manager는 크게 Connections, Request Msg Analyzer, VN API로 나눌 수 있다. Connections는 Path Manager 및 NOX Controller와의 통신을 담당하고,

Request Msg Analyzer는 Path Manager의 Request Message를 분석하여 그에 해당하는 작업을 호출한다. VN API는 OpenFlow기반의 네트워크를 관리하기 위한 기능들을 제공한다.

NOX Controller는 네트워크 제어를 위한 Application이 동작하는 NOX App, 네트워크 운영체제인 NOX, NOX와 OpenFlow 스위치간의 통신을 담당하는 OF Secure Channel로 구성되어 있다.

관리자가 Path Manager를 통해 서비스를 요청한 경우, Path Manager는 Request Message를 생성하여 VN Manager에게 전달한다. Request Message를 전달 받은 VN Manager는 Request Msg Analyzer를 통해 Request Message를 분석하고 VN API를 통해 그에 맞는 API를 호출한다. 관리자의 요청이 수행이 되고 해당 결과값을 XML로 변환한 VN Manager는 이를 다시 Path Manager에게 전달하여 관리자에게 서비스 요청에 대한 결과를 알려준다.

3. 링크상태에 따른 동적 Flow 관리 메커니즘

링크상태에 따른 동적 Flow 관리 메커니즘은 아래와 같은 순서로 동작한다. 먼저 처음 Flow를 할당할 때 기존에 할당된 Flow의 요구 대역폭의 크기를 고려하여 Flow를 할당함으로써 1차적으로 최대한 각 서비스 Flow별 QoS를 보장할 수 있도록 하였고 각 링크는 주기적으로 linkstate를 통해 링크상태를 모니터링 된다.

알고리즘 1. 링크상태에 따른 동적 Flow 관리 알고리즘

WHILE

IF (link || Data packet loss) **THEN**

FOR number of service flow list

IF (required bw for flow < transmitted traffic) **THEN**

Set current flow Incomplete_flow

Set current flow's required bw Adjustment_bw

Set new service flow using Incomplete_flow,

Adjustment_bw Adjustment_flow

Delete current flow

END IF

END FOR

END IF

END WHILE

이때 링크에서 패킷 손실이 발생한다면 설정된 각각의 서비스 flow의 요구 대역폭과 해당 flow를 통해 전송되고 있는 트래픽의 양을 비교하여 QoS가 보장되고 있는지 확인한다. 만약 할당된 서비스 Flow가 자신의 요구 대역폭 이상의 트래픽을 전송하게 된다면 VN Manager의 PathFlow Optimizer는 해당 서비스 Flow의 QoS 보장을 위해 요구 대역폭을 증가시킨 새로운 Flow를 할당하고 이전의 Flow는 삭제하여 해당 Flow를 다시 할당 한다.

아래 알고리즘 1은 이러한 과정 중 할당 된 서비스 Flow에서 전송되는 트래픽이 요구 대역폭의 크기 이상일 때 이를 만족하는 새로운 flow를 할당하고 기존의 flow를 삭제하는 과정을 거침으로써 네트워크의 QoS를 보장하는 과정을 나타내고 있다.

4. 결론 및 향후 연구

본 논문에서는 링크상태에 따라서 동적으로 서비스 Flow를 관리하는 메커니즘에 대하여 제안하였다. 한번 서비스 Flow를 할당한 이후 해당 Flow의 필요한 대역폭이 증가하는 경우 이를 탐지하여 그에 맞는 새로운 Flow를 할당하고 기존의 Flow를 삭제함으로써 수행되는 서비스에 따라 동적으로 Flow의 요구 대역폭을 증가시켜 QoS를 지속적으로 보장함과 동시에 같은 링크를 사용하는 다른 서비스 Flow의 요구 대역폭을 보장할 수 있도록 하였다. 이를 통해 서비스의 QoS를 보장하고 네트워크 자원을 효율적으로 사용가능할 것으로 기대된다.

앞으로 향후 연구로 새로 생성된 서비스 Flow의 링크 사용률을 계산하여 사용률이 적은 Flow의 경우 요구 대역폭의 크기를 줄이고 이를 필요로 하는 다른 서비스 Flow의 요구 대역폭 크기를 증가시키는 등 한정된 네트워크 자원을 효율적으로 사용할 수 있도록 최적화이론 등과 같은 기법을 적용하여 링크의 활용률의 최적화 할 수 있는 방안에 대하여 연구할 것이다.

5. 참고 문헌

- [1] OpenFlow, <http://openflow.org>
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69-74, Mar. 2008.
- [3] ONF, "Software-Defined Networking: The New Norm for Networks", ONF White Paper, (2012. 4)
- [4] OpenFlow switch specification version 1.3.2, <https://www.opennetworking.org>
- [5] Nox Controller, <http://noxrepo.org>
- [6] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards and operating", system for networks. In ACM SIGCOMM Computer Communication Review, July 2008.