

# OpenFlow 기반 네트워크에서 효율적인 네트워크 자원 관리를 위한 동적 플로우 관리 기법

문승일<sup>○</sup>, 홍충선<sup>\*</sup>

경희대학교 컴퓨터공학과

moons85@khu.ac.kr, cshong@khu.ac.kr

## Dynamic Flow Management Mechanism for efficient network resource management in OpenFlow-based Network

Seung Il Moon<sup>○</sup>, Choong Seon Hong<sup>\*</sup>

Department. of Computer Engineering, Kyung Hee University

### 요 약

네트워크가 발전함에 따라 그에 대한 의존도가 높아지고 있는 만큼 기존의 네트워크는 폐쇄적이라는 그 태생의 문제로 인하여 다양한 사용자 요구사항을 만족시키기에는 한계가 있다. 이러한 문제를 해결하고자 최근 기존의 네트워크에서 제어 평면(Control Plane)과 데이터 평면(Data Plane)을 분리하여 네트워크의 제어 관리 측면에서의 구조적인 유연성을 제공하는 소프트웨어 정의 네트워크(SDN : Software Defined Network) 기술에 대한 연구가 활발히 진행되고 있다. 이러한 SDN은 소프트웨어 기반의 프로그래밍 가능한 네트워크 환경을 제공하기 때문에 네트워크 관리 관점에서 보다 효율적으로 네트워크를 운용할 수 있게 해준다. 본 논문에서는 SDN 기술 중 하나인 OpenFlow를 활용하여 OpenFlow 기반 네트워크에서의 네트워크 상태 모니터링을 통한 동적 플로우 관리 기법을 제안하고 이에 대한 검증을 수행한다.

### 1. 서 론

현재 네트워크의 구조는 장비와 인프라의 폐쇄적인 구조라는 태생적인 문제를 갖고 있어 다양한 사용자의 요구사항을 만족시키기가 어렵다. 이러한 문제를 해결하기 위한 방안으로 OpenFlow[1][2]가 제안되었으며 이를 활용한 다양한 연구가 진행되고 있다. OpenFlow는 제어 평면(Control Plane)과 데이터 평면(Data Plane)이 분리된 아키텍처인 SDN[3] 기술 중 하나로 분리된 제어 계층과 인프라스트럭처 계층 간의 통신을 위한 개방형 표준 프로토콜이다. 이러한 OpenFlow는 컨트롤러에 의해 관리를 통해 네트워크를 일괄적으로 제어하며, 소프트웨어 기반의 프로그래밍 가능한 네트워크를 통해 보다 유연하고 확장 가능한 네트워크 제어 환경을 제공해준다.

본 논문에서는 OpenFlow 기반 네트워크에서 주기적인 모니터링을 통해 네트워크의 상태를 수집하고 수집한 정보를 활용하여 네트워크 자원을 효율적으로 사용할 수 있는 동적 플로우 관리 기법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 통해 OpenFlow에 대하여 설명하고 3장에서는 SDN 기반 네트워크 제어 프레임워크에 대하여 설명한다. 4장에서 효율적인 네트워크 자원 관리를 위한 동적 플로우 관리 기법에 대하여 제안하고 5장에서 제안하는 기법에 대한 성능평가를 수행한다. 그 후 끝으로 6장에서 결론 및 향후 연구 방향에 대하여 논한다.

### 2. OpenFlow

OpenFlow는 제어 평면과 데이터 평면을 나누어 관리하며, 서로 다른 스위치와 라우터의 플로우 테이블을 프로그래밍 할 수 있는 오픈 프로토콜을 통하여 소프트웨어 기반의 프로그래밍 가능한 네트워크 환경을 제공해준다.

아래의 그림 1은 OpenFlow의 구조를 보여주고 있다. 그림에서 알 수 있듯이 OpenFlow는 FlowTable, Secure Channel, OpenFlow Protocol의 세 개의 파트로 나눌 수 있다.

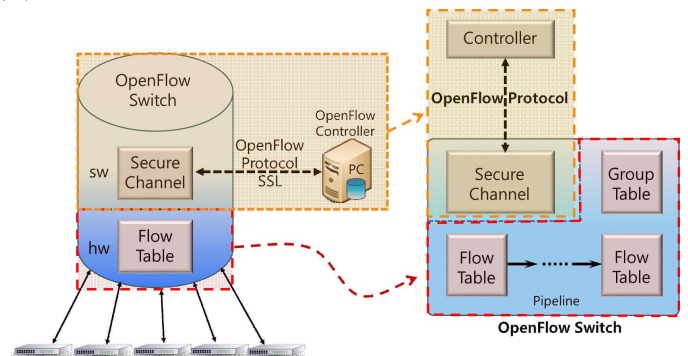


그림 1. OpenFlow 구조

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 전달될 수 있도록 도와주는 역할을 한다. 마지막으로 OpenFlow

본 연구는 미래창조과학부가 지원한 2014년 정보통신/방송(ICT) 연구개발사업의 연구결과로 수행되었음. \*Dr. CS Hong is the corresponding author.

Protocol은 공개되어있는 표준적인 방식을 제공함으로써 스위치와 컨트롤러가 서로 통신할 수 있도록 한다.[4]

### 3. 시스템 구조

본 논문에서 제안하는 OpenFlow 기반의 네트워크 관리를 위한 시스템의 구조는 그림 2에서 알 수 있듯이 Path Manager, VN Manager, SDN Controller로 구성되어 있다.

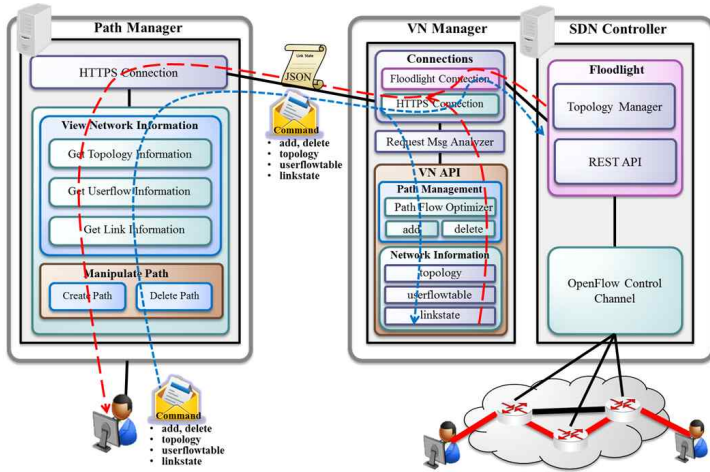


그림 2. 시스템 구조

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 SDN Controller인 Floodlight[5]와의 통신을 담당하고, Request Msg Analyzer는 Path Manager의 Request Message를 분석하여 그에 해당하는 작업을 호출한다. VN API는 OpenFlow기반의 네트워크를 관리하기 위한 기능들을 제공한다.

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 결과 값을 XML로 변환한 VN Manager는 이를 다시 Path Manager에게 전달하여 관리자에게 서비스 요청에 대한 결과를 알려준다.

### 4. 제안하는 동적 플로우 관리 기법

본 논문에서 제안하는 동적 플로우 관리 기법은 다음과 같다. 알고리즘 1에서와 같이 컨트롤러를 통해 새로운 플로우가 생성되면, 먼저 해당 플로우의 서비스 요구 대역폭값을 CurServ\_BW에 저장한다. CurServ\_BW는 QoS를 위한 파라미터로 이를 기준으로 대역폭 조절을 함으로써 QoS를 보장한다.

### 알고리즘 1 : Dynamic Flow Management Algorithm

Initialize for Add flow CurServ\_BW = Req\_BW

Network State Monitoring : For each flow<sub>i</sub> Per PERIOD

- 1: Measure flow<sub>i</sub>'s data rate until DURATION<sub>i</sub>
- 2: Calculate Average data rate of flow<sub>i</sub>
- 3: Update AvrDataRate<sub>i</sub> = Average data rate of flow<sub>i</sub>

Dynamic Flow Management

- 4: Calculate Utilization of flow<sub>i</sub>:
- 5: Utilization of flow<sub>i</sub> = AvrDataRate<sub>i</sub> / Req\_BW<sub>i</sub> \* 100 (1)
- 6: if Utilization of flow<sub>i</sub> less than Threshold<sub>DecBW</sub> then
- 7:     Decrease the CurServ\_BW<sub>i</sub> as 20%
- 8: end if
- 9: if AvrDataRate<sub>i</sub> greater than Req\_BW<sub>i</sub> then
- 10:     Calculate Total Req\_BW and Total Utilization
- 11:      $totalReq\_BW = \sum_{i=1}^n Req\_BW_i$  (2)
- 12:     AvrDataRate of Whole flow =  $\sum_{i=1}^n AvrDataRate_i$  (3)
- 13: Calculate Total Utilization
- 14: totalUtilization =
- 15: AvrDataRate of Whole flow / totalReq\_BW \* 100 (4)
- 16: if totalUtilization less than Threshold<sub>IncBW</sub> then
- 17:     Increase the CurServ\_BW<sub>i</sub> as 20%
- 18: end if
- 19: else
- 20:     Packet drop for QoS
- 21: end if

Req_BW	: 서비스 플로우의 요구 대역폭
CurServ_BW	: 서비스 플로우의 현재 허용 대역폭
flow <sub>i</sub>	: 서비스 플로우
PERIOD	: 모니터링 주기
DURATION	: 평균 전송 속도를 구하기 위한 측정 시간
AvrDataRate	: 평균 전송 속도
totalReq_BW	: 전체 플로우의 요구 대역폭 총합
totalUtilization	: 전체 플로우의 네트워크 자원 사용률
Threshold <sub>DecBW</sub>	: CurServ_BW를 감소시키기 위한 임계값
Threshold <sub>IncBW</sub>	: CurServ_BW를 증가시키기 위한 임계값

플로우가 추가된 후 네트워크 상태를 모니터링 하기 위해 지정된 PERIOD 만큼의 주기로 플로우의 평균 전송 속도인 AvrDataRate을 구한다.

네트워크 상태에 따른 동적 플로우 관리를 위한 제어 방법은 크게 두 가지로 예약된 자원 대비 사용률이 낮을 때와 예약된 자원 이상의 대역폭을 사용했을 때 이다.

예약된 자원 대비 사용률이 낮은 플로우의 경우 유휴 자원을 줄이기 위하여 플로우의 Req\_BW 대비 AvrDataRate의 비율인 사용률을 계산하여 만약 사용률이 Threshold<sub>DecBW</sub> 미만이라면 해당 플로우의 CurServ\_BW의 크기를 20% 감소시켜 할당 가능한 네트워크 자원을 확보한다.

반대로 특정 플로우의 AvrDataRate이 Req\_BW 이상인 경우 네트워크 자원의 상태를 고려하기 위하여 예약된 자원의 총 량을 뜻하는 전체 플로우의 Req\_BW 총합 대비 전체 플로우의 AvrDataRate 총합 비율을 계산하여 totalUtilization을 구한 후 그 값이 Threshold<sub>IncBW</sub> 미만인

면 예약된 네트워크의 자원의 양이 여유가 있기에 해당 플로우의 CurServ\_BW의 크기를 20% 증가시켜 증가하는 트래픽에 대응하여 QoS를 유지한다. 만약 totalUtilization의 값이 Threshold<sub>IncBW</sub> 이상이면 해당 플로우의 트래픽을 Drop시켜 QoS를 유지한다. 이와 같이 네트워크 상태를 모니터링 하여 그 결과에 따라 동적으로 플로우를 관리함으로써 네트워크 자원을 효율적으로 관리할 수 있다.

5. 성능평가

본 논문에 제안한 동적 플로우 관리 기법에 대한 성능평가를 위해 다음과 같이 시뮬레이션을 수행하였다.

표 1. 시뮬레이션을 위한 토폴로지 정보

Number of Nodes	12 nodes
Number of Links	14 links
Link Capacity	1 Gbps
Measurement Period for Utilization	every 10sec

표 1에서와 같이 12개의 노트와 14개의 링크를 갖는 네트워크 토폴로지서 무작위로 플로우를 생성하며 생성된 플로우는 무작위로 트래픽을 발생시킨다. 생성된 플로우에서 발생된 트래픽을 주기적으로 모니터링하며 매 10초마다 전체 네트워크의 할당된 모든 플로우에 대한 사용률을 측정하였다.

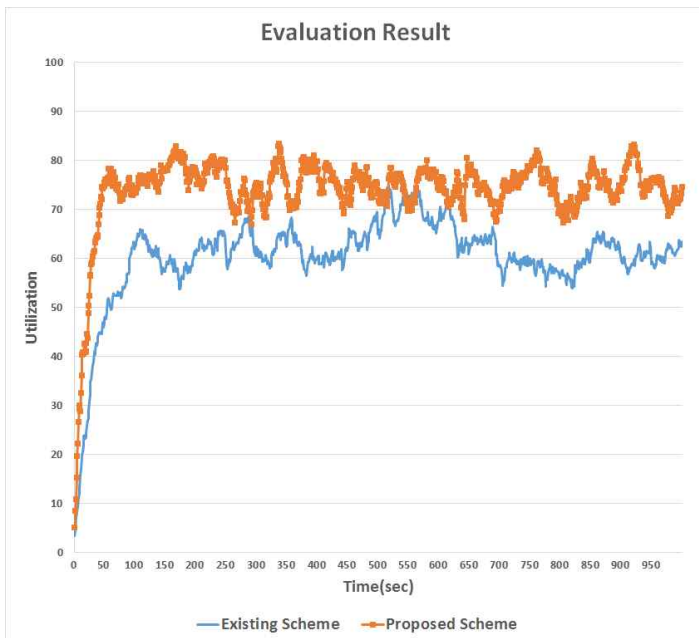


그림 3. 제안하는 동적 플로우 관리기법의 성능 평가

그림 3은 제안하는 기법에 대한 성능평가 결과 이다. 결과에서 알 수 있듯이 제안하는 동적 플로우 관리 기법을 사용했을 때가 그렇지 않을 때 보다 평균 네트워크 자원 사용률이 18.6% 증가하였음을 확인 할 수 있다.

6. 결론 및 향후 연구

본 논문에서는 효율적인 네트워크 자원 관리를 위한 동적 플로우 관리 기법에 대하여 제안하였다. 새로운 플로우가 할당된 후 QoS 제한을 위한 파라미터인 CurServ\_BW를 서비스 수행에 필요한 대역폭인 Req\_BW 값으로 초기화 한다. 이를 기반으로 플로우를 엔트리 단위로 모니터링 하고 예약된 자원 대비 모니터링을 통해 구한 실제 사용률을 기반으로 유휴자원 발생 감소 및 특정 플로우의 트래픽이 폭주하여 혼잡이 예상될 때 전체 네트워크의 예약된 자원 대비 사용률을 기반으로 하여 제한적으로 플로우의 CurServ\_BW를 증가시켜 폭주하는 트래픽에 대비하여 QoS를 유지할 수 있게 하였으며, 성능 평가를 통해 사용률에 대한 성능 향상을 확인 하였다.

이렇듯 네트워크 상태에 따른 동적 플로우 관리 기법을 통하여 네트워크 자원을 보다 효율적으로 관리할 수 있고 보다 다양한 서비스를 수행할 수 있을 것으로 기대된다.

앞으로의 연구로는 최적화 이론을 통해 Threshold<sub>DecBW</sub>, Threshold<sub>IncBW</sub>등 성능에 관련된 파라미터의 최적화 값을 구하는 연구와 네트워크 자원 사용 비용 관점에서 게임 이론을 적용한 최적의 이윤을 창출하는 방안에 대하여 연구할 것이다.

7. 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 1.3.4, <https://www.opennetworking.org>  
 [5] Floodlight Controller, <http://www.projectfloodlight.org/floodlight/>