

# SDN 환경에서 전송 지연 예측 기반 Flow 분류를 통한 부하 분산 기법 연구

김도현<sup>o</sup>, 홍충선\*  
 경희대학교 컴퓨터공학과  
 { doma<sup>o</sup>, cshong\* }@khu.ac.kr

## Delay Prediction-based Flow Classification Strategy for Load Balancing in Software-Defined Network

Do Hyeon Kim<sup>o</sup>, Choong Seon Hong\*  
 Department of Computer Science and Engineering, Kyung Hee University

### 요 약

Software-Defined Network는 중앙 집중 형태의 네트워크 구조로써 Controller가 Data plane에 있는 모든 Openflow enabled switch들을 관리하고, host간의 통신을 제어한다. 또한 하드웨어적인 네트워크 기능들을 소프트웨어적인 형태의 모듈로 Controller에 적용할 수 있기 때문에 다양한 네트워크 관리정책에 대한 연구개발이 진행되고 있고, 이러한 관리정책을 뒷받침 할 수 있는 네트워크 모니터링에 대한 기법들 또한 제시되어 왔다. 이에 본 논문에서는 Transmission Delay 예측 모듈을 통한 효율적인 Flow 분류 기법을 제안한다. 이는 다양한 대역폭을 갖는 여러 경로들로 이루어진 네트워크상에서 분류된 Flow를 전송하는 데 있어 효율적인 경로 분배 역할을 할 수 있고, 부하를 분산시킴으로써 보다 원활한 네트워크 환경을 제공할 수 있다.

### 1. 서 론

Software-Defined Network(SDN)는 새로운 패러다임으로써 기존 네트워크의 구조를 변화시키고 있다. SDN 환경에서는 네트워크 애플리케이션 연동을 통한 손쉬운 대응이 가능하고, 더 나아가 관리자가 원하는 운영정책을 소프트웨어 모듈로 직접 개발하여 적용할 수 있다. 이러한 장점에 힘입어 전 세계적으로 SDN 시장이 확대되고, 이에 따른 다양한 연구 개발이 진행되고 있는 추세이다[1]. 트래픽 분류는 네트워크 환경에 있어서 보안, Quality of Service(QoS)제공, 효율적인 트래픽 관리 및 제어를 위한 요소로 많은 역할을 해왔으며 전통적인 트래픽 분류의 기법으로는 전송 포트를 기반으로 한 분류와 패킷의 Payload 검출(예 : Deep Packet Inspection)을 통한 분류가 있다[2]. 최근 연구에서는 머신러닝 기반의 통계적인 분류 기법이 제시되고 있는 추세이며, 대표적인 알고리즘으로는 K-Means, Naive Bayesian, C4.5, C5.0, J48, Random Forest 가 있고[3], 최근 SDN 관련 연구에도 이러한 트래픽 분류를 통한 보안 및 효율적인 네트워크 운용 정책에 관한 연구가 꾸준히 진행되고 있다[4]. 또한 네트워크 관리 측면에 있어 모니터링에 관한 부분은 유연한 관리 정책을 적용하기 위한 기본적인 시스템으로써 관련 연구가 과거부터 현재까지 지속적으로 진행되어 오고 있다. [5]에서는 SDN 환경에서 Data Plane에 위치한 Openflow enabled switch의 포트의 Usage 및 여러 데이터를 통계적으로 측정하고 이를 라우팅에 반영하였고, [6]에서는 중단 host들 간의 연결 상태 즉, TCP Handshake와 같은 절차가 완료되었는지에 따라 각 Flow를 Useful과 Non-Useful로 나누어 이를 Forwarding 정책에 반영하였다. 이처럼 본 논문에서는 Flow 분류를 통한 네트워크의 부하 분산 기법 적용을 위해 기존에 제안했던 Machine Learning 기반 Naive Bayesian 분류기에 링크

의 대역폭 예측 모니터링 모듈을 적용하여 보다 효율적이고 동적인 네트워크 관리가 가능하도록 하는 기법을 제안한다.

### 2. 관련 연구

#### 2.1. Naive Bayesian 패킷 분류 알고리즘



그림 1 기 제안된 Packet 분류 과정[7]

그림 1은 기 제안한 Naive Bayesian 패킷 분류 모듈로써 OpenFlow Switch의 기본 Action을 ‘to Controller’로 하여 유입되는 패킷을 분류하고 결과 값에 따라 적절한 경로를 할당한 후, 전송하는 알고리즘이다[7]. 하지만 기 제안된 알고리즘의 문제점은 컨트롤러 상에 할당해야 할 경로들을 분류 클래스와 미리 측정 된 링크의 대역폭을 통하여 사전에 정의하기 때문에 대역폭의 가변적인 특성을 파악하지 못한다. 이러한 이유로 사전에 정의된 큰 대역폭을 갖고 있는 경로 일지라도 지속적인 링크 사용으로 오히려 가용 대역폭이 낮아져 QoS를 제공하지 못하는 결과를 초래하게 된다. 따라서 본 논문에서는 end host 간 전송경로를 추출 후, 포트의 대역폭 사용량을 파악하여 전송 지연을 예측하는 모듈을 추가함으로써 위와 같은 문제점을 해결하고자 하였다.

#### 2.2. SDN Flow Table의 Match Field 구성

Flow Table에는 Match field, Counter, Action등이 포함되어 있는 Flow Entry가 있다. 이러한 Flow Entry를 위해 사전에 Match field를 정의하여 이에 부합하는 요소들을 갖는 Packet 또는 Flow가 유입 될 경우, Controller는 Match field에 해당하는 Action을 취한다[8]. 하지만 시시각각 변화하는 네트워크 서비스 요구사항에 대하여 유연한 대응을 하기 위해

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 Grand ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2016-R6812-15-0001)

\*Dr. CS Hong is the corresponding author

서는 보다 동적인 Flow Forwarding 기법이 필요하고, 이를 위해 본 논문에서는 Machine Learning 알고리즘을 적용하여 Training Set을 통해 Flow를 Forwarding하는 동적인 기법을 제안한다.

### 3. 제안 사항

#### 3.1 Flow Classifier Module

표 1 Naive Bayesian 분류기 Training Set

Features			Class
Length per Packet	Protocol	Port Number	
Large, Medium, Small	TCP	22(SSH)	Weakness Flow
		23(Telnet)	
Large (800~1500 bytes)	TCP	Unspecified	Big Flow
	UDP		
Medium (300~799 bytes)	TCP		Normal Flow
	UDP		
Small (1~299 bytes)	TCP		Small Flow
	UDP		

표 1은 Naive Bayesian 알고리즘에 적용되는 확장된 Training Set으로 Port Number, SrcIP, DstIP가 같은 첫 10개의 패킷을 토대로 분류가 이루어진다. 이 때, Flow로 정해지는 10번째 패킷의 Length값에 따라 Length per packet 에 대한 구분이 Large, Medium, Small로 나뉜다. 또한 22번 포트와 23번 포트를 사용하는 Flow에 대해서는 또 다른 Secure 링크를 만들어 전송할 수 있도록 Training Set을 결정하였다. 하지만 본 논문에서는 해당 내용을 다루지는 않는다.

#### 3.2 Transmission Delay Prediction Module

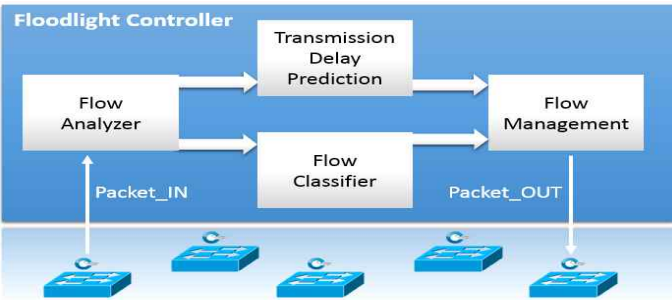


그림 4 Floodlight Controller Modules

그림 2는 제안하는 기법에 대한 Controller 모듈을 나타낸 그림이다. Flow Analyzer 모듈에서 Packet\_IN 메시지를 분석하여 다섯 가지의 요소(Packet Length, SrcIP, DstIP, Protocol, Port Number)를 추출한다. 또한 Port Number, SrcIP, DstIP가 같은 첫 10개의 패킷을 분석한 후, SrcIP, DstIP과 Length를 Transmission Delay Prediction 모듈로 전송하여 효율적인 전송경로 설정을 위해 end host 간 링크의 전송지연을 측정한다.

$$tBW(bps) = (tx_1 - tx_0) \times 8 / 5 \quad (1)$$

$$rBW(bps) = (rx_1 - rx_0) \times 8 / 5 \quad (2)$$

$$PortBW(bps) = tBW(bps) + rBW(bps) \quad (3)$$

$$Prediction\ Value\ of\ TD(sec) = \frac{Packet\ Length \times 8}{MaxBW - PortBW} \quad (4)$$

SrcIP, DstIP를 통해 end host 간 전송 가능 경로를 추출한 후, 해당 경로 상에 있는 OpenFlow 스위치의 사용 중인 포트 대역폭(Bandwidth) 값을 수식 (1), (2), (3)를 통해 측정하고 5초 간격으로 갱신함으로써 지속적으로 실시간 대역폭 사용량을 파악할 수 있다. 이 대역폭 값과 Flow Analyzer 모듈로부터 얻은 Length 값을 이용하여 수식 (4)를 통해 전송 지연에 대한 예측 값을 계산한다. Flow Management 모듈에서는 각 스위치 포트들의 전송 지연 값을 기반으로 Flow Forwarding 경로를 선택하고 각 OpenFlow 스위치에 Packet\_OUT메시지를 이용하여 Action Field를 전송한다.

#### 알고리즘 1 Transmission Delay Prediction Module

- 1: AL ← Average Length From FA Module
- 2: PBW<sub>ij</sub> ← Port j Bandwidth on Switch i
- 3: PVTD<sub>ij</sub> ← Port j prediction value of Transmission Delay on Switch i
- 4: **for** each Switch i on Forwarding Path
- 5:     **for** each Port j on Switch **except** in\_port
- 6:         Compute PBW<sub>ij</sub>
- 7:         Compute PVTD<sub>ij</sub>
- 8:     **end for**
- 9:     Forward PVTD<sub>ij</sub> set to Flow Management
- 10: **end for**

#### 3.3 Flow Management Module

$$PVTD\ Value\ of\ Path = \frac{\sum PVTD_{ij}\ on\ Path}{Number\ of\ Switches} \quad (5)$$

Flow Management 모듈에서는 Transmission Delay Prediction 모듈로부터 받은 각 경로의 스위치 포트 별 전송 지연 값을 이용하여 수식 (5)을 통해 경로의 평균 전송 지연 값을 결정한 후, 경로들의 전송 순위를 결정한다. 또한 Flow Classifier 모듈로부터 받은 Flow 분류의 결과 값에 따라 평균 전송 지연 값이 가장 작은 경로부터 차등적으로 Flow를 전송한다.

#### 알고리즘 2 Flow Management Module

- 1: PVTDs ← PVTD<sub>ij</sub> set From TDP Module
- 2: PTDp ← Average Transmission Delay Prediction Value of the Forwarding Path
- 3: C ← Flow Class from FC Module
- 4: **for** each PVTDs
- 5:     Compute PTDp
- 6: **end for**
- 7: **if** C is Big Flow **then**
- 8:     Select Smallest PTDp Path and Send Action Field to Switches on Forwarding Path
- 9: **else if** C is Normal Flow **then**

- 10: Select Second Smallest PTDP Path and Send Action Field to Switches on Forwarding Path
- 11: else
- 12: Select Other Path and Send Action Field to Switches on Forwarding Path
- 13: end if

4. 성능 평가

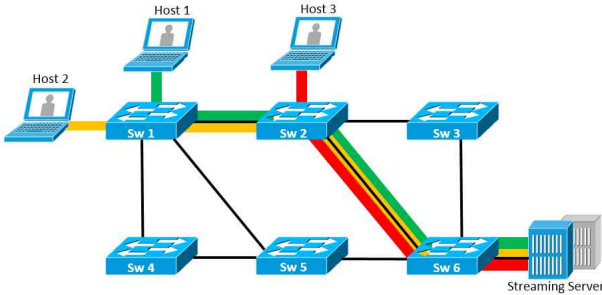


그림 3 Floodlight Controller의 기본 전송 경로

그림 3은 성능 평가를 위해 구축한 Network Topology로 Floodlight Controller를 사용하였으며 PC기반 OpenVswitch 6대와 3개의 host, 그리고 1개의 Streaming Server가 있다. Host 1은 0~20초 동안 서버에 비디오 스트리밍을 요청하고 Host 2는 10~30초 동안 서버에 오디오 스트리밍을 요청한다. 아울러 Host 3 또한 20~40초 동안 또 다른 오디오 스트리밍을 요청한다. 이를 통해 링크의 가용 대역폭 변화를 측정 한 후, 측정값을 바탕으로 전송 지연 값을 예측하여 Forwarding 경로의 변화 및 네트워크의 부하 분산 여부를 측정하고자 하였다. 기본 Floodlight Controller의 경우 Dijkstra 알고리즘을 바탕으로 Forwarding 경로를 결정하기 때문에, Sw2의 eth1포트의 병목현상이 발생하였다.

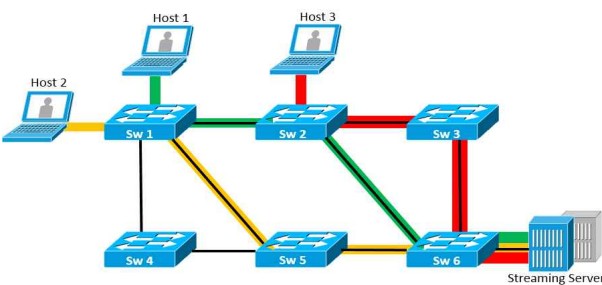


그림 4 제안하는 알고리즘이 적용된 Controller의 전송 경로

하지만 제안하는 알고리즘에 의한 Forwarding의 경우, 그림 4와 같이 Flow 분산을 통해 네트워크의 여러 자원을 적절히 활용하여 Flow를 전송하였다. 아울러 그림 5는 해당 Sw2의 eth1포트의 전송 패킷 량을 나타낸 그림으로, 해당 포트의 tx bytes 수를 측정하였다. 제안하는 알고리즘에 의해, 20초 이후 eth1포트로 전송되는 패킷이 없기 때문에 더 이상 tx bytes가 증가하지 않는다.

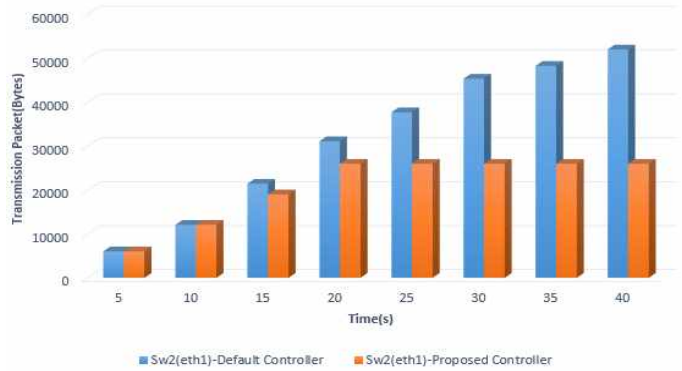


그림 5 Sw2-eth1 포트에서의 Packet 처리량

5. 결론 및 향후 연구

본 논문에서는 SDN Controller상에 기존에 제안했던 Naive Bayesian 알고리즘과 더불어 전송 지연 예측 모듈을 적용하여 전체 네트워크의 부하를 분산하는 정책을 제안하였다. Packet 분류를 위해 적용하였던 Naive Bayesian 알고리즘을 Flow 분류 기법에 적용하기 위해, 첫 10개의 Packet을 분석하고 그에 따른 분류 값 및 전송 지연 예측 값을 기반으로 Forwarding 한 결과, 네트워크 전반에 걸쳐 여러 자원을 이용할 수 있었다. 하지만 본 논문에서 제안하는 Transmission Delay Prediction 모듈의 경우, 전송 가능한 모든 경로를 추출하고 해당 경로 상에 있는 스위치의 포트들을 모두 점검하기 때문에 큰 규모의 네트워크에 적용하기에는 컨트롤러의 과부하 등의 문제가 존재한다. 이러한 이유로 향후 연구로서 네트워크의 확장성을 고려하여 Transmission Delay Prediction 모듈에서 보다 효율적인 전송 링크 선정 방법을 추가하여 그에 대한 전송 지연 예측 값을 측정할 계획이며 서비스 별 네트워크 포트를 기반으로 새롭게 Training Set을 정의하여 다양한 네트워크 Flow에 대한 효율적인 분류가 가능하도록 확장할 계획이다.

6. 참고 문헌

- [1] "IDC, 전세계 SDN 시장 연평균 53.9% 성장으로 2020년 125 억달러 전망", <http://www.kr.idc.asia/press/pressreleasearticle.aspx?prid=401>
- [2] Jun Zhang et al., "Network Traffic Classification Using Correlation Information", IEEE Transactions on Parallel and Distributed System, Vol.24, No.1, Jan.2013
- [3] "Traffic classification", [https://en.wikipedia.org/wiki/Traffic\\_classification](https://en.wikipedia.org/wiki/Traffic_classification)
- [4] Lu He et al., "VTC: Machine Learning Based Traffic Classification as a Virtual Network Function", 2016 ACM International Workshop on Security in Software-Defined Networks & Network Function Virtualization, 9-11 March, 2015
- [5] Davide Adami et al., "A Network Control Application enabling Software Defined Quality of Service", IEEE International Conference on Communications (ICC) 2015, 8-12 June 2015
- [6] Naman Grover et al., "liteFlow: Lightweight and Distributed Flow Monitoring Platform for SDN", 2015 1st IEEE Conference on Network Softwarization (Netsoft), 13-17 April 2015
- [7] 김도현, 강효성, 김남호, 이진원, 홍충선, "SDN환경에서 확률적인 패킷 분류를 통한 QoS 보장 기법", 제42회 한국정보과학회 정기총회 및 동계학술발표회, 17-19 Dec. 2015
- [8] "CPS 590: Software Defined Networking", Theophilus Benson, <http://slideplayer.com/slide/5722212/>