

SDN 환경에서 확률적인 패킷 분류를 통한 QoS 보장 기법

김도현⁰, 강효성, 김남호, 이진원, 홍충선*
 경희대학교 컴퓨터공학과

{ doma, kanghs, knm1471, notwonz, cshong }@khu.ac.kr

QoS Technique Using Probabilistic Packet Classification in Software-Defined Network Environment

Do Hyeon Kim⁰, Hyo Sung Kang, Nam Ho Kim, Jin Won Lee, Choong Seon Hong*
 Department of Computer Science and Engineering

요 약

최근 네트워크 환경이 하드웨어 중심에서 소프트웨어 중심으로 변화하고, 분산형 구조에서 중앙집중형 구조로 변화함에 따라 SDN Controller를 활용하는 네트워크 제어 기법이 제시되고 있으며, 다양한 관리정책을 보다 쉽게 적용할 수 있게 되었다. 본 논문에서는 소프트웨어 정의의 네트워크 환경에서 확률적인 기법을 통하여 패킷을 분류함으로써 각 패킷에 요구되는 즉, QoS를 보장하는 방안을 제안한다. 다양한 대역폭을 갖는 여러 경로들을 분류된 패킷과 매칭시킴으로써 관리자는 효율적으로 네트워크를 운용할 수 있고, 서비스 이용자는 보다 나은 네트워크 환경을 제공받을 수 있다.

1. 서 론

최근 미래 인터넷과 관련된 여러 기술 등장하고 있는 가운데, Programmable network 기술의 일종인 Software-Defined Network(SDN)는 전통적인 네트워크의 구조를 변화시키고 있다. SDN이 등장함에 따라 네트워크 환경은 하드웨어 기반에서 소프트웨어 기반으로 변화되고 있다. 기존의 하드웨어 기반 환경에서는 특정 기능을 네트워크에 적용시키려 할 때, 그 기능을 수행하는 하드웨어 장비를 설치해야 하기 때문에, 비용적인 측면에서 많은 부담이 된다. 하지만 소프트웨어 기반의 네트워크 환경에서는 특정 기능을 Controller에 소프트웨어 적으로 구현함으로써 추가 장비 없이 기능 적용이 가능하다 [1]. 즉, 각 하드웨어 장비가 갖고 있던 Control Plane과 Data Plane을 분리시킴으로써 모든 네트워크 제어는 Controller에서 수행하며 SDN 스위치에서는 패킷 전달만을 담당한다 [1], [2]. 이러한 중앙집중형 관리를 통해 SDN에서는 보다 다양한, 보다 유연한 네트워크 정책을 적용시킬 수 있다. 실제로 SDN Controller를 이용하여 QoS를 보장하는 다양한 네트워크 정책을 적용하는 연구가 진행되고 있다. [3]에서는 Controller 안에 Deep Packet Inspector(DPI)와 Machine Learning(ML) 기능을 구현하고 DPI와 ML이 갖는 한계점을 상호 보완하여 패킷을 분류함으로써 QoS를 보장하는 기법에 대하여 기술하였다. 이에 본 논문에서는 기존 SDN 환경에서 QoS

보장 기법 연구와 더불어 분류 알고리즘인 Naive Bayesian 알고리즘을 적용하여 SDN 환경에서 Video, Audio, Text 패킷을 분류함으로써 다양한 네트워크 자원을 활용하고 사용자는 서버로부터 보다 나은 서비스를 제공받을 수 있는 기법을 제안한다.

2. 관련 연구

2.1. SDN Controller에서의 Packet_IN 메시지 분석

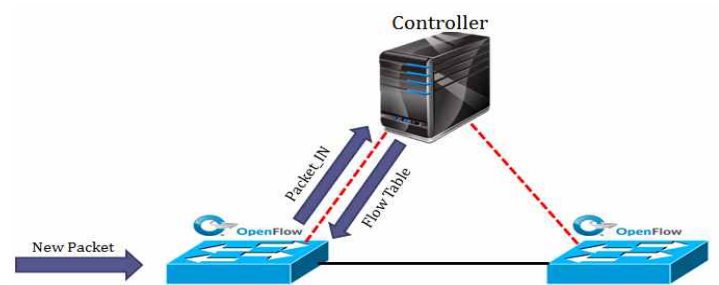


그림 1 SDN환경에서의 Flow Table 요청 절차

그림 1에서는 SDN환경에서 새로운 패킷이 OpenFlow 스위치로 들어왔을 때, OpenFlow스위치가 Controller에게 Flow Table을 요청하는 절차를 나타낸다. 이 패킷의 정보에 해당하는 Flow Table이 없다면 스위치는 패킷을 Controller로 전송하고 Controller에서는 패킷을 분석하여 네트워크 정책에 따라 OpenFlow 스위치에 Flow Table을 할당함으로써 패킷을 전송할 수 있게 한다.

본 연구는 한국정보화진흥원(NIA)의 미래네트워크연구시험망(KOREN) 사업 지원과제의 연구결과로 수행되었음(15-951-00-001).
 *Dr. CS Hong is the corresponding author

2.2. Naive Bayesian 알고리즘

머신러닝 알고리즘 중, 분류 알고리즘에 속하는 Naive Bayesian 알고리즘은 지도학습을 활용한다. 이전의 경험(이전에 알고 있던 정보)과 현재의 증거(새로 습득한 정보)를 토대로 어떤 사건의 확률을 추론한다[4]. 이는 분류의 기준이 되는 Training Set을 미리 학습시키고, 새로운 data가 입력이 되었을 때, 조건부 확률을 이용하여 data가 어떤 클래스에 속하는지를 확률적으로 분류하는 알고리즘이다.

기 때문에 본 논문에서는 전송 Protocol의 종류 또한 패킷의 특성으로 정하였다.

$$\begin{aligned}
 P(C|x) &\propto P(x|C)P(C) \\
 P(C|x) &= P(x_1, x_2 | C) \\
 &= P(x_1 | C) P(x_2 | C) P(C)
 \end{aligned}
 \tag{1}$$

식 (1)은 Naive Bayesian Classifier의 수학적 모델[4]로 C는 Class로 Video, Audio, Text로 분류되고 x는 Data의 특성으로 패킷 Length와 Protocol이다.

3. 제안 사항

3.1. Naive Bayesian Packet Classifier



그림 2 패킷의 경로 생성 과정

그림 2에서와 같이 OpenFlow스위치가 컨트롤러로 Packet In 메시지를 전송하면 Packet Analysis 모듈이 패킷의 Protocol 및 Length정보를 추출하여 Packet Classification 모듈로 전송한다.

$$P(x|C) = \frac{Count(x, c) + 1}{(\sum_{x \in F} Count(x, c)) + |F|}
 \tag{2}$$

$$PL_{norm} = \frac{PL_{Inbound}}{Max(PL_{TrainingSet})}
 \tag{3}$$

$$0 < PL \leq 1
 \tag{4}$$

식 (3)을 이용하여 패킷 Length의 특성을 정규화 한 후 Large, Medium, Small과 같이 세 가지 Level로 분류한다. 각 Class의 기준값, 예를 들어 Video Class인 경우 800 bytes와 1500 bytes가 각각 하한 및 상한 기준 값이 되므로 이를 이용하여 Large Level의 범위를 얻을 수 있다. 아래의 표 2는 각 Level별 값의 범위를 나타낸다.

표 1 Packet 분류 Training Set

Class	Features	
	Protocol	Packet Length
Video	TCP	Large
	UDP	
Audio	TCP	Medium
	UDP	
Text	TCP	Small
	UDP	

표 2 패킷 Length에 따른 Level 분류

Level	Range of Packet Length	Range of Value
Large	800 ~ 1500 bytes	1 ~ 0.54
Medium	300 ~ 799 bytes	0.53 ~ 0.2
Small	1 ~ 299 bytes	0.199 ~

Packet Classification 모듈에서는 패킷 분류의 기준이 되는 표 1의 Training Set을 이용하여 패킷의 Class를 지정한다. 패킷 Length의 범위는 WireShark를 이용하여 각 패킷이 전송 될 때의 Length를 측정 한 후 이를 Training Set에 반영하였다. 아울러 TCP와 UDP를 구분하는 이유는 같은 Packet Length를 갖는 패킷이라도 Protocol에 따라서 전송환경이 달라지기 때문이다. 예를 들어 TCP와 같이 Packet Loss를 허용하지 않는 경우 전송 경로 상에 혼잡이 발생하지 않아야 하고, UDP 같은 경우 이를 무시한 채 전송이 진행되

예를 들어 Controller가 UDP 프로토콜을 사용하며 950 bytes의 패킷 Length를 갖는 패킷을 받았다면, 식 (1)에서의 x는 UDP와 Large라는 값을 갖게 된다. 패킷 분석을 통하여 얻은 x값을 식 (1)에 대입하여 조건부 확률에 대한 결과 값이 가장 클 때의 Class를 선택한다. 아울러 패킷 Length값의 최대치를 1500 byte로 해놓은 이유는 링크의 MTU(Maximum Transmission Unit)를 1500 byte로 사용했기 때문이다. 또한 식 (2)를 통해 Training Set에 존재하지 않는 Feature에 대한 확률이 0이 됨을 방지 할 수 있다[4].

3.2. 분류 결과에 따른 Flow 지정

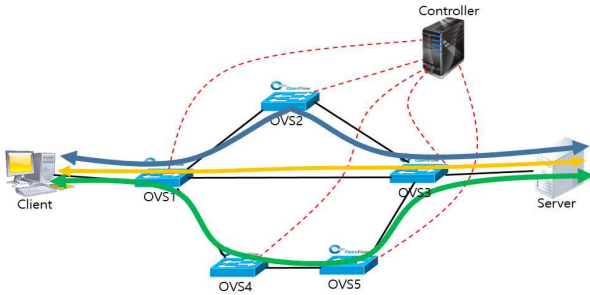


그림 3 시뮬레이션 Topology

그림 3에서 볼 수 있듯이 컨트롤러에서 지정할 수 있는 전송 경로는 세 가지이며 각 전송 경로는 서로 다른 대역폭을 갖는 경로로써 패킷 분류의 결과 값에 따라 세 경로 중 하나의 경로로 패킷을 전송한다.

표 3 Class별 전송 경로

Class	Forwarding Path	Link Bandwidth
Video	OVS1-OVS4-OVS5-OVS3	38.89Mbps
Audio	OVS1-OVS2-OVS3	22.61Mbps
Text	OVS1-OVS3	9.53Mbps

컨트롤러는 패킷 분류 결과 값을 기반으로 표 3과 같이 전송경로를 결정한다. 이러한 절차를 거침으로써 각 패킷에 대한 QoS를 만족시킬 수 있다.

4. 성능 평가

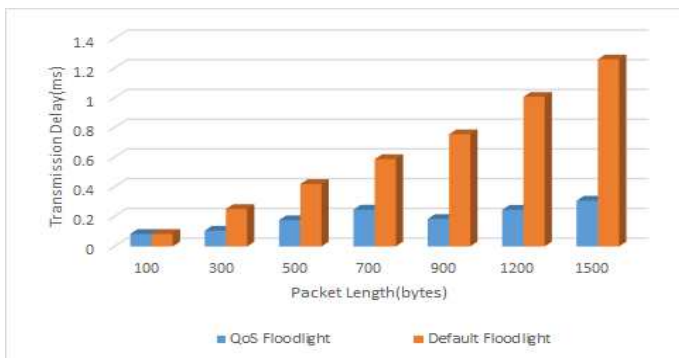


그림 4 패킷 Length의 증가에 따른 전송 딜레이

본 논문에서는 3장의 그림 3과 같은 Topology를 Mininet 과 Floodlight Controller를 이용하여 구성하였고, Client로부터 각각 다른 Length를 갖는 패킷을 서버에게 보냄으로써

Video, Audio, Text 패킷을 구분하였다. 평가 방법으로는 Client가 Server로 패킷을 전송할 때, 기본적인 Floodlight Controller와 본 논문에서 제안한 QoS모듈을 적용시켰을 때의 전송 딜레이를 비교하였고 그림 4와 같이 결과를 얻을 수 있었다. 기본 Floodlight Controller는 홑 수가 가장 적은 경로로 패킷을 전송하기 때문에 패킷 Length가 800 bytes 이상이 되더라도 전송 경로는 변하지 않으므로 전송 딜레이가 지속적으로 증가 한다. 반면 QoS Floodlight Controller는 패킷 Length가 800 bytes 이상일 때, 더 높은 Bandwidth를 갖는 경로로 패킷을 전송하기 때문에 전송 딜레이가 줄어든다[5].

5. 결론 및 향후 연구

본 논문에서는 SDN Controller 상에 Naive Bayesian 알고리즘을 이용하여 패킷을 분류하고 이에 따른 전송 경로를 할당하는 QoS 정책을 제안하였다. 제안된 정책을 가상 네트워크 환경인 Mininet에 적용한 결과, 콘텐츠를 전송할 때 발생하는 딜레이가 기존의 Floodlight Controller 환경에서 콘텐츠를 전송할 때 발생하는 딜레이보다 훨씬 적은 수치를 얻을 수 있었다. 특히, 패킷 Length가 900 bytes 이상인 경우에 Bandwidth가 큰 다른 경로로 패킷을 전송시킴으로써 훨씬 더 적은 딜레이가 소요되는 것을 확인할 수 있었다.

패킷 분류기에 적용된 현재 Training Set의 패킷 특징은 전송 Protocol과 패킷 Length로만 정의했기 때문에, 다양한 서비스가 요구되고 개발되고 있는 미래 네트워크에 적용하기에는 다소 무리가 있다. 이러한 이유로 향후 연구로는 Training Set에 다양한 패킷 특성을 정의하여 분류기의 성능 및 정확도를 향상시킬 것이다.

6. 참고 문헌

- [1] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, Haiyong Xie, "A Survey on Software-Defined Networking" IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL 17, NO. 1, FIRST QUARTER 2015
- [2] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig, "Software-Defined Networking: A Comprehensive Survey" Proceedings of the IEEE, Volume:103, Issue:1
- [3] Yunchun LI, Jingxuan LI, "MultiClassifier: A combination of DPI and ML for application-layer classification in SDN", ICSAI 2014, Nov.15-17 2014
- [4] Naive Bayes classifier, Online Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- [5] http://www.cs.colostate.edu/~massey/Teaching/cs457/Restricted_Access/Slides/457lecture02.pdf