

# SDN 환경에서 실시간 패킷 유형과 QoS 평가 기반한 효율적인 Load Balancing 기법

윤정현\* · 권태욱\*\*

Efficient Load Balancing Techniques Based on Packet Types and Real-Time QoS Evaluation in SDN

Jung-Hyun Yoon\* · Tae-Wook Kwon\*\*

## 요 약

4차 산업혁명 기술로 네트워크 트래픽은 공급의 증가, 수요의 증가, 트래픽 패턴의 복잡도 증가의 특징으로 유통되는 Data가 늘고 있다. 이런 방대한 트래픽을 효율적으로 관리하기 위해 H/W와 S/W가 분리된 개념인 SDN이 차세대 네트워크로 주목받고 있다. S/W Opensource로 구현된 SDN의 Control Layer, Infrastructure Layer 컨트롤러를 이용하여 경직된 Vendor 종속성의 문제를 탈피하여 유연한 정책을 적용할 수 있는 장점으로 많은 연구가 진행되고 있다. 이에 본 논문에서는 SDN을 이용하여 네트워크 계층의 패킷 구조를 통해 그룹화하고 Packet Delay, Transmission Rate 등을 분석하여 효율적인 부하분산 기법을 제안한다.

## ABSTRACT

With the technology of the 4th industrial revolution, network traffic is increasing due to an increase in supply, an increase in demand, and an increase in the complexity of traffic patterns. SDN, a concept in which H/W and S/W are separated in order to efficiently manage such massive traffic, is attracting attention as a next-generation network. A lot of research is being conducted on the merits of applying flexible policies by avoiding the problem of rigid vendor dependency by using the SDN controller implemented with S/W Opensource. Therefore, in this paper, we propose an efficient load balancing technique by grouping through the packet structure of the network layer using the control layer and infrastructure layer of SDN and analyzing the packet delay and reception rate.

## 키워드

SDN(Software Defined Networking), Load Balancing, Packet Priority  
소프트웨어 정의 네트워크, 부하 분산, 패킷 우선 순위

\* 국방대학교 컴퓨터공학과 (202026012@kndu.ad.kr)

\*\* 교신저자 : 국방대학교 컴퓨터공학과

• 접수일 : 2021. 07. 21

• 수정완료일 : 2021. 09. 03

• 게재확정일 : 2021. 10. 17

• Received : Jul. 21, 2021, Revised : Sep. 03, 2021, Accepted : Oct. 17, 2021

• Corresponding Author : Tae-Wook Kwon

Dept. of Computer Engineering, Korea National Defense University,

Email : kwontw@mnd.go.kr

## 1. 서론

컴퓨터, 통신의 발전의 발전으로 초연결의 시대인 4차 산업혁명 시대를 영위하고 있다. 사람과 사람의 통신뿐만 아니라 사람과 기계, 기계와 기계가 통신하는 흐름에서 전 세계 네트워크 트래픽은 공급의 증가, 수요의 증가, 트래픽 패턴의 복잡도의 특징으로 그림 1과 같이 연평균 53% 증가하고 있다. 경제능력을 떠나 사람들은 통신을 필수재로 생각하고 스마트폰이나 집에서 활용하는 PC에 한정하지 않고 IPTV(Internet Protocol Television), 웨어러블기기 등 하루가 멀다하고 네트워크에서 교환되는 데이터양이 급속도로 증가하고 있다. 또한 초기 네트워크장비의 기술적인 한계와 대역폭의 문제로 단순 Text파일의 전송에서 현재는 4G, 5G 기술을 이용하여 실시간 방송시청 송수신, Youtube 플랫폼으로 불과 몇 년 만에 네트워크층에서 교환되는 영상, 음성파일로 데이터의 크기가 커진 것을 알 수 있다.

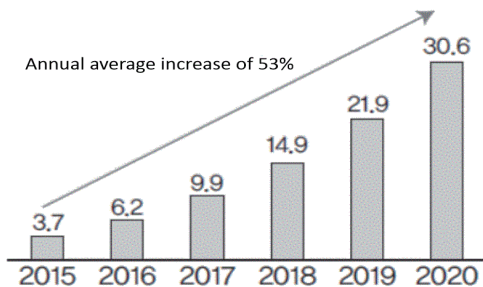


그림 1. 네트워크층에서 교환되는 트래픽 증가  
Fig. 1 Increasing traffic exchanged on the network layer

이런 방대한 트래픽 생산으로 기존 네트워크 Legacy 체계는 각 Vendor에 종속된 H/W(Hardware) 기능과 정책으로 고비용에 따른 확장제한, 상황과 규모에 맞는 구성이 제한되는 등 제한사항과 비효율성으로 SDN(Software Defined Network)가 등장하였다.[1] SDN은 그림 2처럼 기존의 Switch나 Router와 같이 S/W 기반의 Control Layer를 SDN Controller를 통해 중앙집중화하여 Infrastructure Layer의 장비를 제어하는 기술로 정의할 수 있다[2]. 현 시대의 트래픽 공급·수요의 증가, 복잡도 증가의 문제를 유연한 인프라 확장의 장점을 가진 SDN이 해결할 수 있는

대안으로 평가되면서 세계적 플랫폼 기업인 Google과 Facebook도 적극적으로 SDN을 활용하여 인프라를 구축하고 있다. 또한 S/W(Software) OpenFlow 기반으로 SDN 제어기능을 구축하여 사용자가 요구하는 정책과 방향으로 설계할 수 접근성으로 점차 SDN 기반으로 네트워크 인프라구축이 활성화되고 있다.

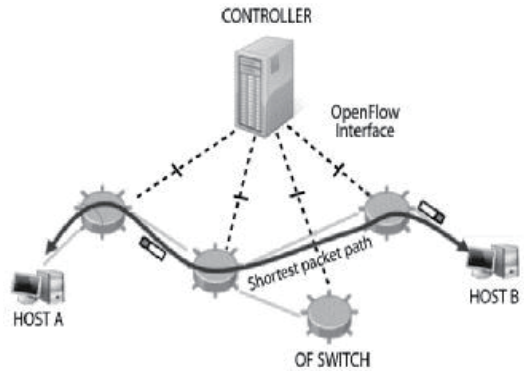


그림 2. SDN Controller에 의한 Switch 통제  
Fig. 2 Switch control by SDN Controller

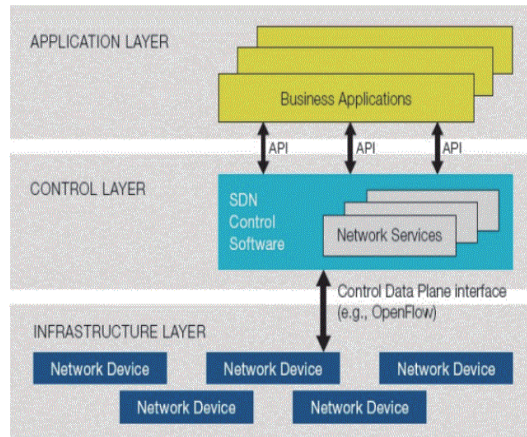


그림 3. SDN의 구조  
Fig. 3 Structure of SDN

과거와 달리 네트워크층에서 다루어지는 데이터의 트래픽 증가, 대용량의 패킷 Flow를 효율적이면서 유연한 정책적응으로 설계되는 SDN체계로 컨트롤하기 접근을 모색하고 있고, 기존 데이터 전송품질을 나타내는 QoS(Quality of Service)를 기준으로 활발한 연구가 진행되고 있다. 특히 제한된 능력을 지속적으로 최적의 상태에서 활용하거나 버거운 프로세스를 분할하여 최적의 능력을 낼 수 있는 범위로 분할처리하는

Load Balancing에 대한 연구가 심도있게 진행 중이다.

본 논문에서는 기존 연구범위 외에서 SDN의 S/W 기반 Controll Layer가 급변하는 트래픽 패턴의 복잡도를 실시간으로 확인하고 이를 효과적으로 컨트롤하여 네트워크층의 특정 패킷 Flow가 과도한 딜레이 시간이 생성되지 않으며 데이터의 신뢰성으로 평가되는 송·수신율을 향상시키는 방안을 제안한다.

## II. 관련연구

### 2.1 SDN Load Balancing 연구동향

#### 2.1.1 서버 응답시간 기반 부하분산

문중배(2006)는 주소캐싱매커니즘에 의해 부하불균형이 발생가능성이 높은 DNS(Domain Name System)의 동적 갱신과 라운드로빈 방법을 이용하여 새로운 부하분산방안을 제시하였다[3]. 서버의 부하량에 따라 그림 4와 같이 서버리스트를 추가, 삭제하는 방법으로 과부하의 서버를 관리하였다.

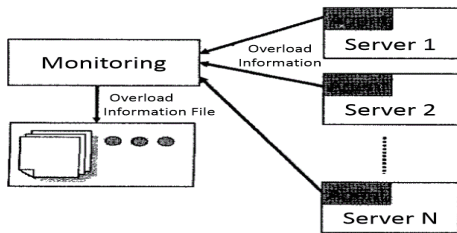


그림 4. 과부하정보 전송을 통한 서버관리  
Fig. 4 Server management through overload information

동적 갱신을 통한 과부하 서버관리를 통해 100개의 클라이언트 기준으로 대조군인 라운드로빈 방식보다 응답시간 60s에서 40s로 33% 감소하였다.

Hong Zhong(2018)은 Controll Layer와 Infrastructure Layer가 분리되어 있는 SDN의 구조적 특성을 이용하여 Controll Layer가 서버의 응답시간을 측정하고 측정된 응답시간을 기반으로 부하가 적은 서버를 선택하여 부하분산의 효과를 제시하였다[4].

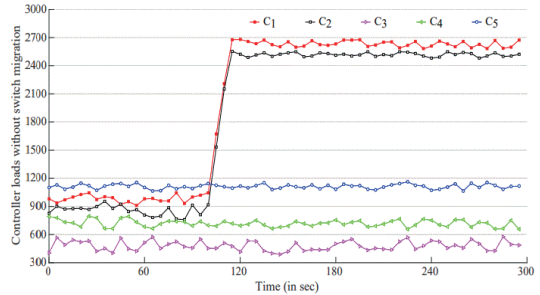


그림 5. 대조군 라운드로빈 방식의 서버 과부하  
Fig. 5 round-robin server overload

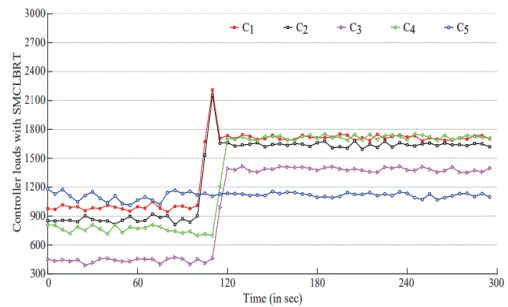


그림 6. SMCLBRT 방식의 서버 과부하  
Fig. 6 SMCLBRT server overload

Hong Zhong(2018)의 제안기법으로 그림 5, 6과 같이 C1, C2로 편중된 과부하가 100s 부근에서 C3, C4, C5로 분산되면서 네트워크 자원의 효율적인 사용목적을 달성하였다.

#### 2.1.2 Packet 형태를 고려한 부하분산

손재혁(2015)은 SDN 방식을 도입하여 목적지 IP주소를 기반으로 서비스별 우선순위를 부여하는 방식으로 패킷을 분류하는 방안을 제시하였다[5]. 패킷에 가장 적합한 네트워크 자원을 그림 7와 같이 Flow Manager가 판단하고 할당하여 Flow Table에 반영하는 방식으로 패킷 전송 시 대기시간 최소화에 집중하였다.

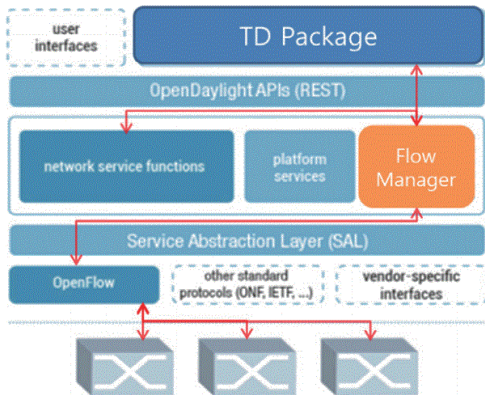


그림 7. Flow Manager를 이용한 Flow Table 구성  
Fig. 7 Flow Table Configuration with Flow Manager

실험경로는 100, 300, 600Mbps 총 3가지로 패킷의 크기에 따라 실험을 진행하였다. 실험결과 500byte 이하의 패킷은 큰 차이가 없었으나 600Byte 이상의 패킷에서 크기가 클수록 대용량의 대역폭의 경로로 우선순위로 선정되는 것이 가장 효율적인 전송대기 시간이 형성되는 것을 확인하였다.

윤지영(2019)은 인터넷 인프라의 발전으로 특정 정보에 노출될 경우가 많아졌다는 사실에 기반하여 특정한 데이터가 폭증되었을 때 SDN을 이용한 효율적인 부하분산 방안을 제시하였다[6]. 그림 8과 같이 SDN Controller 안에 위치한 Module을 이용하여 패킷을 분류하고 폭증하는 패킷에 우선순위를 부여하여 일반적으로 분류되는 패킷보다 25.2% 대기시간이 감소하는 부하분산의 효과를 얻었다.

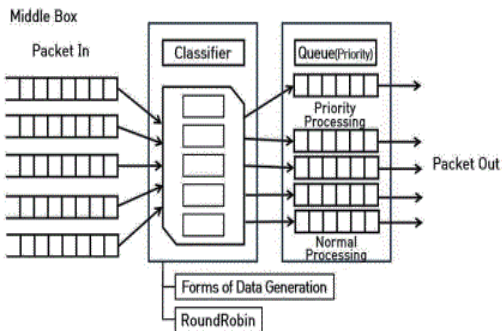


그림 8. Middle Box를 이용한 Packet 분류와 Queue  
Fig. 8 Packet Classification and Queue with Middle Box

2.1.3 Server & Packet 형태를 고려한 부하분산  
김종건(2020)은 네트워크 계층으로 유입되는 데이터를 효과적으로 SDN 기술로 유형별로 그림 9와 같이 분류하고, 분류한 데이터를 서버의 응답시간을 기준으로 처리하여 효율성을 향상 시켰다[7].

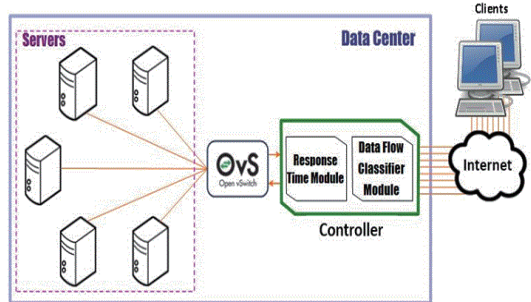


그림 9. Controller를 통한 Packet 분류, 서버 응답시간 측정  
Fig. 9 Packet classification, server response measurement

김종건(2020)은 폭증 유형의 패킷을 우선으로 처리하는 대조군 윤지영(2019)의 기법보다 지연시간을 평균 27.1% 감소시키는 효과를 얻었다.

### III. 제안사항

#### 3.1 제안하는 기법의 동작방법

제안하는 기법은 SDN Controller를 이용한 Packet Flow를 평가하고 QoS 변수인 특정 유형의 Packet Delay, Transmission Rate가 전체적인 수치와 일정범위 내에서 관리되도록 부하분산하는 기법을 제안한다.

이를 위해 SDN Controller는 ① 현재 Packet Flow를 평가 및 분석하고 ② 판단된 Delay, Transmission Rate 수치의 이격을 조정한다. Packet Flow 분류는 홍승표(2018), 홍순화(2001)의 실시간 패킷분석 시스템과 대용량 네트워크 트래픽 모니터링 분석도구 연구를 통해 확인된 현재 인터넷 환경에서의 패킷 Size를 기반으로 분류를 한다.[8] 해당 연구내용을 통해 인터넷 환경에서 측정되는 패킷의 평균 Size는 700Byte 수준으로 작은 Size는 128Byte이하, 큰 Size는 1500Byte가 넘는 수준으로 본 논문에서는 100Byte Size를 A Group, 1500Byte를 B Group으로 구분하였

다.

### 3.1.1 Evaluate the Current Flow Packet

Packet Flow를 SDN의 Classifier가 그림 10와 같이 2개 그룹으로 분류를 하고, 그림 11과 같이 SDN의 Infrastructure Layer의 Meter, Marker를 이용해 혼잡제어를 할 수 있도록 준비를 한다.

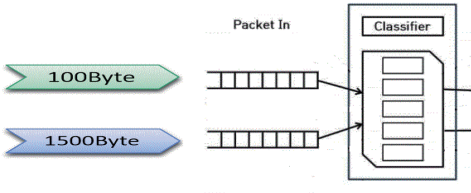


그림 10. 패킷 포워딩 및 Classifier의 역할  
Fig. 10 packet forwarding and Role of classifier

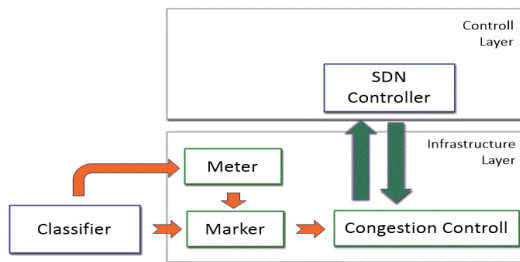


그림 11. Meter, Marker를 이용한 Flow 상태판단  
Fig. 11 Determination of Flow Status Using Meter, Marker

### 3.1.2 제안기법에서 SDN Controller의 역할

기존의 QoS보장을 위한 패킷 스케줄링 기법 중 가장 보편화된 스케줄링 기법은 라운드로빈이다. 그러나 인터넷 환경에서 생성되는 패킷을 일정한 능력을 갖춘 서버에 지속적으로 처리를 강요하면 과부하 방지를 위한 패킷 Drop이 발생하게 된다. 패킷 Drop은 결론적으로 네트워크 QoS에 부정적인 요소로 신뢰성을 바탕으로 하는 영상, 음성과 같은 데이터 품질에 악영향을 미치는 것으로 평가된다.

SN.John(2011) 패킷 Drop에 대한 연구에서도 서버 부하와 관계없이 포워딩이 진행될 경우 그림 12과 같이 패킷 손실이 지속적으로 증가하는 것을 확인할 수 있다[9]. 특히 연구된 내용을 바탕으로 Packet Size가 클수록 손실되는 비율이 커지는 것이 확인되었다.

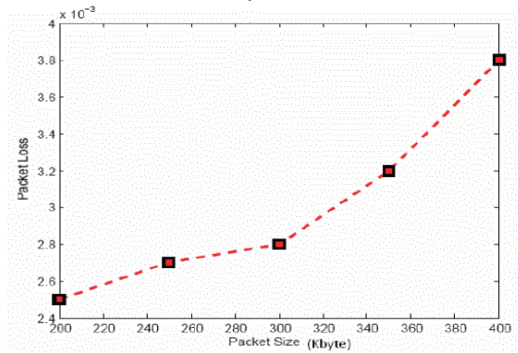


그림 12. Server 과부하 상태에서의 Packet Loss  
Fig. 12 Packet Loss under Server Overload

이런 상황에서 Size에 따른 패킷의 우선순위 부여 처리는 반대로 Small Size Packet의 Delay와 Loss에 부정적인 결과를 미치기 때문에 상황에 유동적으로 대처하는 부하분산이 필요하다.

실시간으로 동적으로 변화하는 Flow를 고려하면서 Load Balancing을 하기 위한 SDN Controller의 세부 기능으로 S/W구현된 Trace, Module Manager, Flow Cache, Packet Streamer, Routing 기능을 활용한다. Afaq Muhammad(2017)의 SDN에서 Flow의 실시간 분류, 시각화 및 QoS제어에 사용했던 SDN Controller Floodlight는 Packet Flow를 실시간으로 검출하고 QoS 변수에 대한 시각화가 가능하다는 것을 정리하였다[10]. 연구에서 사용한 Flow Delay 산출방법을 사용하여 Group A, B의 Delay, Transmission Rate를 산출하고 이를 일정수준으로 관리되도록 하는 부하분산 방안을 제안한다.

### 3.1.3 Equalize Load Balancing

제안기법의 Controll Layer에서 Packet Flow를 통해 Packet In Time과 Packet Out Time을 그림 13과 같이 산출한다.

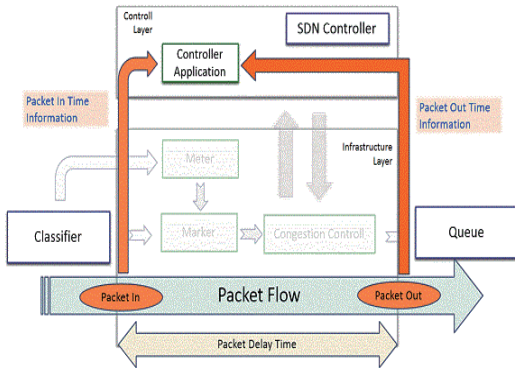


그림 13. Control Application을 이용한 Delay 측정  
Fig. 13 Delay measurement using Control Application

Packet Delay는 Flow의 진입과 Queue 진입 시간으로  $T_{delay} = T_{PO(packet\ out)} - T_{PI(packet\ in)}$ 으로 산출하고 Group A, B의 Delay 차이를 지속적으로 측정한다. Group A와 B의 Delay 피리울 DR(Disparate Rate)를

$$Disparate\ Rate(packet) = \frac{T_{avg.\nabla ay(A)} - T_{avg.\nabla ay(B)}}{T_{avg.\nabla ay}}$$

산출하여, -10 이하로 측정되면  $T_{avg.delay(B)}$ 가 큰 값을 의미하기 때문에 그림 14과 같이 SDN 컨트롤러는 Infrastructure Layer의 Marker를 이용하여 추후 진입하는 Packet의 ToS(Type of Service)필드의 DSCP(Differentiated Services Code Point)를 수정한다. 반대의 상황에서도 동일하게 처리하여 DR이 지속적으로 10 이내로 유지한다.

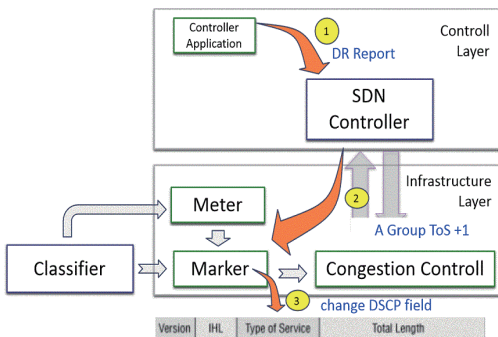


그림 14. SDN Controller를 이용한 혼잡제어과정  
Fig. 14 Congestion Control Process using SDN Controller

### 3.1.4 제안기법의 Algorithm

분석된 DR 수치에 따라 DSCP에 +1을 통해 우선 순위를 조정하는 알고리즘으로 생성된 Packet은 라운드 로빈과정에서 먼저 처리하게 되어 그 격차를 줄이게 된다.

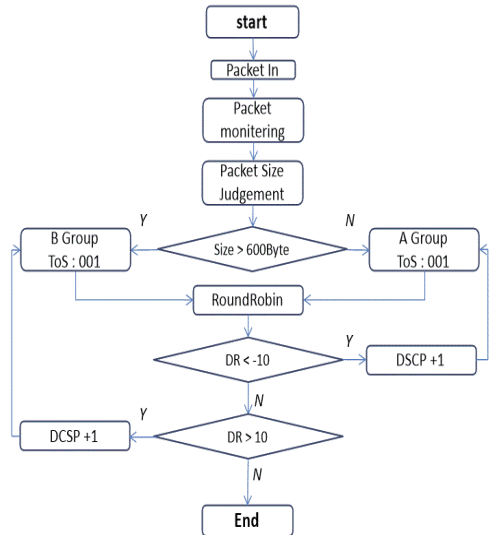


그림 15. 제안기법의 알고리즘  
Fig. 15 Proposal Algorithm

## IV. 실험 결과

### 4.1 시스템 구현

제안기법을 구현하기 위하여 그림 16과 같이Group A, B의 패킷을 생성하는 Source 1, 2를 구성하고 SDN Controller 역할을 하는 Node를 Riverbed Modeler로 토폴로지를 완성하였다. 또한 일반 라운드 로빈 방식의 대조군을 그림 17와 같이 구성하였다.

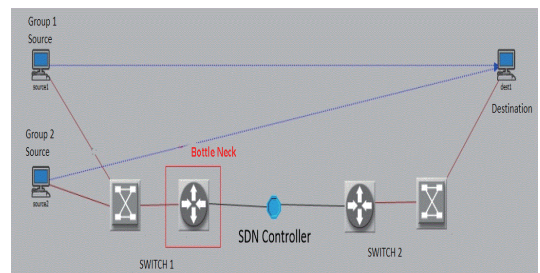


그림 16. 제안기법의 토폴로지  
Fig. 16 Proposal Topology

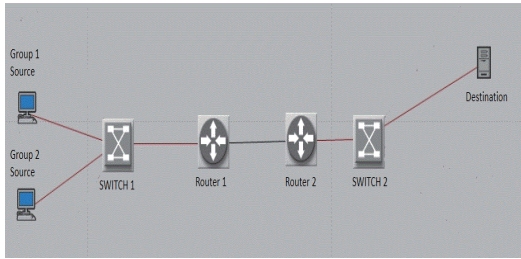


그림 17. 동일한 실험환경의 대조군  
토폴로지(Roundrobin)

Fig. 17 Comparative Topology in the Same  
Experimental Environment

### 4.2 실험 환경

제안하는 부하분산 기법은 Riverbed Modeler 를 이  
용하였으며, 실험 환경은 CPU Intel (R) Core(TM)  
i3-8100 @ 3.60GHz, OS Windows 10, RAM 8GB이다.

### 4.3 실험 방법

실험은 과부하현상을 만들어주는 Source 1, 2와  
Node를 연결하는 Router 1에 Bottleneck section을 설  
정하여 제안기법의 알고리즘이 동작하도록 설정하였  
다. 이후 동일한 환경의 그림 11과 그림 12를 실험시  
간 5분을 설정하여 Group A, B에서 생성된 Flow가  
Destination까지 도착하는 Delay 시간과 송신된  
Packet 대비 수신된 Packet으로 분석을 한다.

### 4.4 결과 분석

#### 4.4.1 Packet Delay

Delay 분석 결과, 그림 18의 라운드로빈과 그림 19  
의 Proposal 모두 병목현상은 40s부터 시작되었다. 그  
러나 라운드로빈은 Packet Size가 큰 Group B의 딜  
레이 진폭 차이가 Maximum 0.116s, Minimum 0.027s  
로 큰 차이를 보였지만 제안기법을 사용하였을 때는  
Maximum 0.036s, Minimum 0.009s로 감소하였고, 일  
정수준의 격차로 유지가 되는 것을 확인할 수 있었다

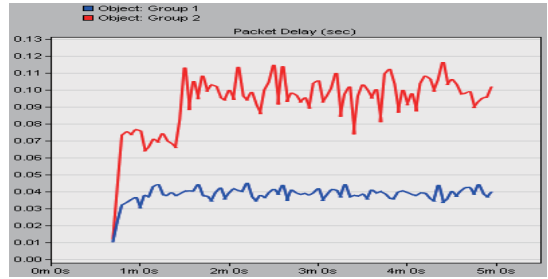


그림 18. Roundrobin 패킷 딜레이  
Fig. 18 Roundrobin Packet Delay

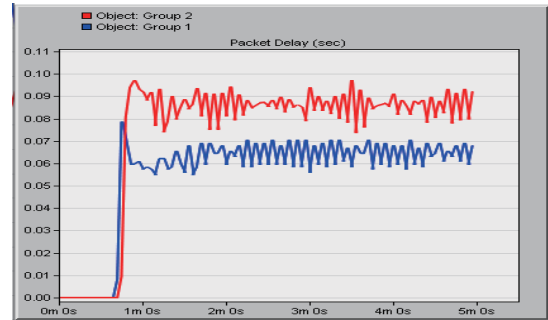


그림 19. Proposal 패킷 딜레이  
Fig. 19 Proposal Packet Delay

#### 4.4.2 Transmission Rate

인위적으로 생성한 Packet의 Drop여부를 확인할  
수 있는 Transmission Rate를 통해 라운드로빈과  
Proposal의 실험결과를 살펴보면 그림 20, 그림 21과  
같다. 그림 15의 라운드로빈의 경우 10,000,000개의 패  
킷이 도착하기 위해서는 10,900,000개의 패킷이 그림  
16의 Proposal의 경우 10,420,240개의 패킷이 필요하  
였다. 대략 5%의 479,760개의 패킷이 Proposal 기법  
을 사용했을 때 Drop을 예방할 수 있었다.

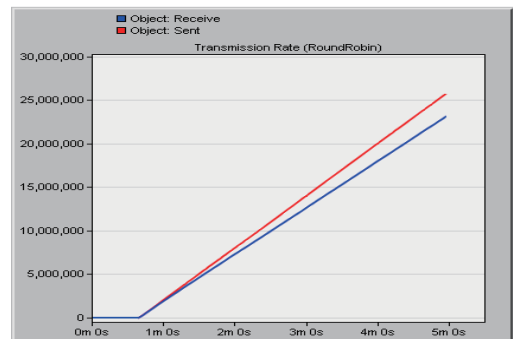


그림 20. Roundrobin 송수신율  
Fig. 20 Roundrobin Transmission Rate

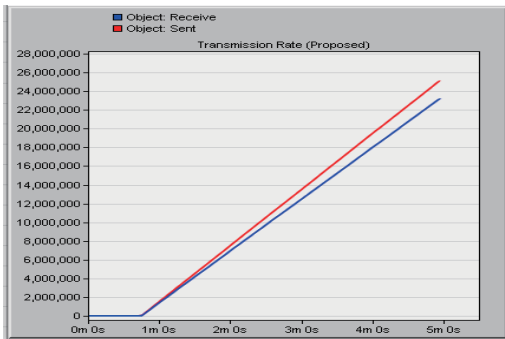


그림 21. Proposal 송수신율

Fig. 21 Proposal Transmission Rate

## V. 결론 및 향후 연구

본 논문에서는 공급의 증가, 수요의 증가, 트래픽 패턴의 복잡도 증가라는 현재 인터넷 환경을 바탕으로 SDN Controller를 이용하여 효율적인 부하분산하는 방안을 제시하였다. 실험결과와 같이 일반적인 라운드로빈 또는 우선순위 스케줄링의 경우, 최초 프로그래밍된 방향으로 일률적으로 진행되어 유동적인 상황에 대한 특정 그룹의 Packet Delay와 Transmission Rate를 해결할 수 없었다. 고질적인 기존 네트워크 장비의 한계로 평가되는 Vendor 종속성은 이런 유연한 대처가 제한되었지만, SDN Controller의 기능을 이용한 Packet Size 기반 제안기법으로 트래픽 관리에 유연하게 대처가 가능하였다.

향후 연구로, SDN Controller가 프로세스 기준으로 현재 제안기법보다 단순하면서 Packet Flow를 구별하고 Jitter나 Packet Drop 비율 등을 함께 분석하여 효율적인 부하분산이 되도록 SDN Controller 역할을 설계하여 효과를 확인하고자 한다.

## References

[1] H. Lim, "Feature Selection Method for the Classification of Traffic in SDN," *The J. of Korean Institute of Communications and Information Sciences*, vol. 44, no. 1, 2019, pp. 106-116.

[2] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig "Software-defined networking : A comprehensive

survey." *Proceedings of the IEEE*, vol. 103, no. 1, 2014, pp. 14-76.

[3] J. Moon, "DNS-Based Dynamic Load Balancing Method on a Distributed Web-Server System," *Korean Institute of Onformation Scientists ans Engineers: System and theory*, vol. 33, no. 1, 2006, pp. 193-204.

[4] H. Zhong, Q. Lin, J. Cui, R. Shi, and L. Liu, "An efficient SDN load balancing scheme based on variance analysis for massive mobile users," *Mobile Information Systems*, vol. 15, no. 1, 2015, pp. 1-9.

[5] J. Son and C. Hong, "SDN-Based Packet-Forwarding and Delay Minimization Algorithm for Efficient Utilization of Network Resources and Delay Minimization," *Korean Institute of Onformation Scientists ans Engineers*, vol. 21, no. 11, 2015, pp. 727-732.

[6] J. Yoon and T. Kwon, "An Efficient Load Balancing Technique Considering Forms of Data Generation in SDNs," *J. of Korea Multimedia Society*, vol. 22, no. 2, pp. 247-254.

[7] J. Kim and T. Kwon, "Efficient Load Balancing Technique Considering Data Generation Form and Server Response Time in SDN," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 15, no. 4, 2020, pp. 679-686.

[8] S. Hong, K. Kim, J. Kim, M. Woo, D. Oh, and H. Lee, "Real-time Packet Analysis System Development," *In Proc. KIIT (Korea Institute od Information Technology)*, Gwangju, Republic of Korea, 2018, pp. 265-266.

[9] O. Adegbenro, S. N. John, and B. Akinade, "The Contributory Effect of Latency on the Quality of Voice Transmitted over the Internet," *Master's Thesis, University of Lagos Graduate School of Electrical and Electronics engineering*, 2011.

[10] A. Muhammad, and W. Song, "Real-Time Classification, Visualization, and QoS Control of Elephant Flows in SDN," *The J. of Korean Institute of Communications and Information Sciences*, vol. 42, no. 3, 2017, pp. 612-622.



저자 소개



**윤정현(Jung-Hyun Yoon)**

2011년 육군사관학교 전자공학과  
졸업 (공학사)  
2020년 ~ 현재 국방대학교 대학  
원 컴퓨터공학과 석사과정

※ 관심분야 : 네트워크, SDN



**권태욱(Tae-Wook Kwon)**

1986년 육군사관학교 컴퓨터공학  
과 졸업 (공학사)  
1995년 美. 해군대학원 컴퓨터공  
학과 (공학석사)

2001년 연세대학교 컴퓨터공학과 (공학박사)

2007년 ~ 현재 국방대학교 컴퓨터공학과 교수

※ 관심분야 : 네트워크, Sensor Networking,  
CCN, SDN, NFV

