

# SDN 환경에서 Dynamic Flow Management에 의한 Load Balancing 기법

김택영\* · 권태욱\*\*

Load Balancing Technique by Dynamic Flow Management in SDN Environment

Taek-Young Kim\* · Tae-Wook Kwon\*\*

## 요 약

네트워크 장비의 하드웨어 영역과 소프트웨어 영역을 분리하고 오픈소스 기반의 소프트웨어를 사용하여 네트워크를 정의하는 차세대 네트워크 기술인 SDN의 등장으로 기존 네트워크 체계가 가지고 있던 복잡성과 확장성의 문제를 해결하고 저비용으로 사용자의 환경과 요구조건에 맞춤형 네트워크 구성이 가능해졌다. 하지만, 컨트롤러와 스위치 간에 발생하는 많은 제어 통신으로 인한 네트워크의 부하가 발생할 수 있다는 구조적 단점을 가지고 있어 이를 효과적으로 해결하기 위한 네트워크 부하분산에 대한 많은 연구가 선행되었다. 특히 플로우 테이블과 관련된 부하분산 기법의 기존 연구에서는 플로우 엔트리에 대한 고려 없이 진행된 연구가 많아 플로우 수가 많아지게 되면 패킷 처리속도가 떨어져 오히려 부하를 가중시키는 결과를 가져오기도 했는데, 본 논문에서는 이러한 문제점을 해결하기 위해 실시간으로 플로우를 모니터링하고 동적 플로우 관리 기법을 적용하여 플로우 수를 적정 수준으로 조절하면서도 높은 패킷 처리속도를 유지할 수 있는 새로운 네트워크 부하분산 기법을 제안한다.

## ABSTRACT

With the advent of SDN, a next-generation network technology that separates the hardware and software areas of network equipment and defines the network using open source-based software, it solves the problems of complexity and scalability of the existing network system. It is now possible to configure a custom network according to the requirements. However, it has a structural disadvantage that a load on the network may occur due to a lot of control communication occurring between the controller and the switch, and many studies on network load distribution to effectively solve this have been preceded. In particular, in previous studies of load balancing techniques related to flow tables, many studies were conducted without consideration of flow entries, and as the number of flows increased, the packet processing speed decreased and the load was increased. To this end, we propose a new network load balancing technique that monitors flows in real time and applies dynamic flow management techniques to control the number of flows to an appropriate level while maintaining high packet processing speed.

## 키워드

SDN(Software Defined Networking), Network Load Balancing, Flow Table

소프트웨어 정의 네트워킹, 네트워크 부하 분산, 플로우 테이블

\* 국방대학교 석사과정(kty2062@naver.com)

\*\* 교신저자 : 국방대학교 컴퓨터공학과 교수

• 접수일 : 2022. 10. 20

• 수정완료일 : 2022. 11. 15

• 게재확정일 : 2022. 12. 17

• Received : Oct. 20, 2022, Revised : Nov. 15, 2022, Accepted : Dec. 17, 2022

• Corresponding Author : Tae-Wook Kwon

Dept. Computer Science, Korea National Defense University

Email : kwontw9042@kndu.ac.kr

## 1. 서론

4차 산업혁명과 정보화시대로 접어들면서 네트워크에서 유통되는 데이터 양이 기하급수적으로 늘어나고 있는데 기존 LTE 기술에 비해 최대 20배 이상 빠른 속도로 데이터를 주고받을 수 있는 5G기술의 상용화로 인해 초고용량의 데이터를 저지연으로 주고받을 수 있게 되면서 인공지능, 머신러닝, 자율주행기술 등의 첨단기술들이 상용화되기 시작하였고 활용 분야가 갈수록 많아지고 있다. 거기에 더하여 최근 전 세계적으로 코로나 팬데믹을 겪으며 온라인 동영상 서비스(OTT)와 사회관계망서비스(SNS) 등 비대면 서비스 의존도가 더욱 높아지며 네트워크 트래픽이 갈수록 급증하고 있다. 이러한 변화에 대응하기 위해 네트워크 장비들에게 요구되는 성능은 갈수록 높아지고 보안 요구수준도 높아지면서 장비에 보다 더 많은 기능이 계속 탑재되어야 하는 상황에 직면했다.

기존의 하드웨어 중심의 네트워크 환경에서는 네트워크 장비를 생산하는 각 제조사 별로 각기 다른 HW와 SW로 제품을 생산하여 제공하다 보니 서로 간에 호환이 되지 않는 경우가 많고, 네트워크 규모는 점점 더 커지고 추가적으로 설치해야 하는 장비들은 점점 늘어나다 보니 네트워크가 매우 복잡해지고 유연하지 못해서 새로운 기능과 서비스를 추가하거나 변경하는 것이 매우 어려워졌다. 그리고 소비자들은 네트워크 장비를 구매할 때 필요한 기능만을 탑재한 장비를 구매하는 것이 불가능하고 제조사는 완제품으로 장비를 제공하기 때문에 불필요한 기능까지 포함되어 있는 장비를 구매하게 되어 구매비용이 상승하는 문제점을 가지고 있다. 이렇듯 기존 하드웨어 중심의 네트워크 구조가 가지고 있는 복잡성과 유연성의 한계를 극복하고 사용자 중심의 네트워크를 구현하기 위해서 나온 차세대 네트워크 체계가 SDN(Software Defined Networking)이다.

SDN은 소프트웨어 정의 네트워킹으로 소프트웨어로 네트워크를 정의하고 제어하겠다는 개념으로 데이터 영역과 제어 영역을 분리하여 중앙 집중적인 통제 시스템에서 네트워크 트래픽 처리 방식과 각종 기능을 관리할 수 있도록 설계, 구축된 프레임워크이다. 과거에는 네트워크 장비의 제어 기능이 각 하드웨어에 있었지만 SDN에서는 그림 1과 같이 제어 영역을

하드웨어에서 분리하여 SW를 통해 논리적 또는 가상적인 실체로서 네트워크를 관리 및 제어하는 것이 가능하여 신속하고 용이하게 네트워크 환경 구축이 가능하고, 통합 인터페이스로 네트워크 환경을 관리할 수 있다.

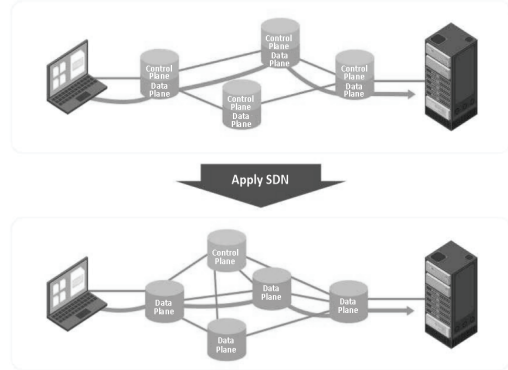


그림 1. SDN 네트워크 구조[1]  
Fig. 1 Network structure of SDN[1]

SDN 체계가 가지는 이러한 장점들을 활용하여 효율적인 네트워크 부하분산에 대한 여러 연구가 진행되고 있는데, 본 논문에서는 SDN 환경에서 플로우 모니터링을 통한 동적 플로우 관리기법을 적용하여 네트워크 장비의 부하를 분산하는 방법에 대하여 연구하고자 한다.

## II. 관련연구

### 2.1 SDN Flow Table

기존 네트워크에서 패킷을 전달할 때는 목적지 주소 기반의 라우팅 테이블을 사용하였는데 SDN 네트워크에서는 흐름 정보를 기반으로 하는 플로우 테이블을 활용하여 패킷을 전달하게 된다. 플로우 테이블의 기본 구조는 그림 2와 같이 규칙, 행동, 통계의 3개의 필드로 구성되어 있는데 규칙 필드에서는 패킷 헤더에 들어있는 정보를 읽어오고, 행동 필드에서는 스위치가 수행할 행동에 대해 정의하며, 통계 필드에서는 패킷에 대한 통계수치 값을 저장한다[2].

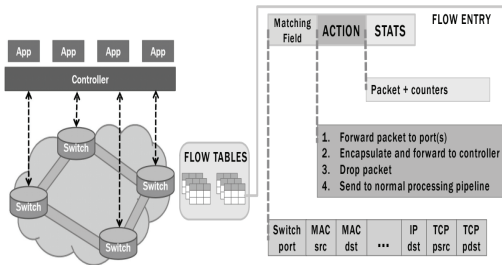


그림 2. Flow Table[3]  
Fig. 2 Flow Table[3]

기존의 네트워크 체계에서는 경로를 결정하는 방법에 따라 RIP, OSPF, BGP 방식으로 구분할 수는 있지만 근본적인 구조는 변경하여 사용할 수 없고 장비 제조사에서 설계한 대로 사용해야 하는데, SDN에서 사용하는 플로우 테이블은 오픈소스의 기반의 오픈플로우에서 제공하고 있기 때문에 사용자의 환경과 요구조건에 따라 플로우 테이블을 관리하는 기법을 다양하게 설계하여 사용할 수 있게 되었다.

플로우 테이블과 관련한 기존 연구 중 플로우 엔트리 교체 과정에서 플로우의 존재 시간을 예측하여 짧은 시간 동안 존재하는 플로우를 긴 시간 동안 존재하는 플로우보다 우선 적으로 삭제하는 SFF(Short Flow Fast) 교체 알고리즘을 제안하여 LRU(Least Recently Used) 알고리즘과의 비교실험을 진행하였고, 실험 결과 SFF 알고리즘이 LRU 알고리즘에 비해 전체 플로우 엔트리를 매칭하는 횟수가 증가하였고 스위치와 제어기 간 메시지 교환 횟수가 감소하였으며 제어기의 부하가 감소하였다는 연구결과가 있었다[4].

SDN에서 패킷을 처리할 때 기본적으로는 패킷을 각기 개별적으로 들어올 때마다 바로 처리하는 개별 플로우 방식으로 동작하지만 방화벽 구축, 침입 차단 시스템 구축, 서버 부하분산 등의 목적으로 다중 플로우 테이블을 활용하여 플로우를 그룹화 하는 것이 가능하다. 이에 대한 기존 연구 중 그룹 플로우 테이블을 활용하여 서버별 처리능력에 따라 가중치를 부여하여 근원지 IP 주소를 기준으로 플로우를 그룹화 하여 할당하는 방법에 대하여 제시하였고 이를 통해 서버 부하분산과 플로우 엔트리 수를 효과적으로 조절할 수 있음을 확인한 연구결과가 있었다[5].

## 2.2 SDN 부하분산 기법

SDN 네트워크에서는 패킷을 처리하는 과정이나 컨트롤러와 스위치 사이의 제어 메시지 교환 등 여러 과정에서 네트워크에 미치는 부하가 가중될 염려가 있는데 이를 효과적으로 줄이기 위해 네트워크 부하분산에 대한 연구가 다양하게 진행되고 있다. SDN 네트워크 부하분산에 대한 연구는 서버의 부하분산에 대한 연구와 스위치와 컨트롤러 등 네트워크 장치 부하분산에 대한 연구의 크게 2가지 분야로 나누어 연구가 진행되고 있다.

먼저 서버 부하분산에 대한 연구는 SDN 환경에서 데이터센터로 유입되는 데이터를 생성 형태에 따라 분류하고, 응답속도가 빠른 서버가 분류된 데이터를 처리하도록 함으로써 부하를 효율적으로 분산하는 방법에 대하여 제안하였고[6][7], SDN 환경에서 서버로부터의 부하 정보 보고를 기반으로 컨트롤러에서 서버 상태를 변경하고 이를 반영하여 부하분산을 수행하는 방법을 제시하였고 RR( Round-Robin) 방식과 비교를 통해 서버 간 부하 차이를 줄이는 데 보다 효과적인 방법임을 증명하였다[8].

다음으로 스위치와 컨트롤러 등 네트워크 장치에 대한 부하분산에 대한 연구는 오픈 플로우의 Pipe-line 프로세싱을 이용한 다중 플로우 테이블을 활용하여 개별 플로우 테이블을 사용했을 때보다 패킷 평균 전송량과 평균 대역폭이 향상되어 전체 네트워크의 부하가 분산되는 것을 확인하였고[9], 패킷에 대한 해시값을 주소 값으로 활용하여 플로우 테이블을 구성하여 패킷이 스위치로 들어왔을 때 해시 주소 값을 통해 플로우 테이블에 집적 접근하여 플로우 정보를 획득함으로써 기존에 선형검색을 통해 플로우 정보를 반환하던 기존 방식에 비해 빠른 처리속도를 보인 연구결과가 있었다[3]. 다만 해당 연구에서는 비교적 적은 플로우 수를 가진 네트워크에서는 높은 처리속도를 보인 반면에 플로우 수가 많아지게 되면 기존방법들과 처리속도가 비슷하게 떨어지는 한계점이 존재했다.

### III. 제안 기법

#### 3.1 DFM(: Dynamic Flow Management) 기법

제안하는 기법은 새로운 패킷이 스위치로 들어왔을 때 ① 컨트롤러는 각 스위치의 전체 플로우 수를 보고받아 ② 플로우 수가 미리 설정해 놓은 임계치와 비교하여 ③ 임계치를 초과하지 않은 경우는 기본 동작 상태인 개별 플로우 방식으로 처리하고 그렇지 않으면 그룹 플로우 방식으로 전환하여 처리하고 ④ Idle Time-out 시간이 지나서 전체 플로우 수가 임계치 이하로 내려가게 되면 다시 개별 플로우 방식으로 전환되는 개념이다. 이를 요약한 제안기법 Flow-chart는 그림 3과 같다.

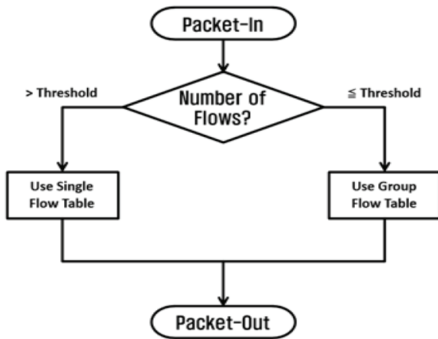


그림 3. 제안기법 Flow-chart  
Fig. 3 Suggestion technique Flow-chart

#### 3.2 Flow Table 구성

제안기법 구현을 위해 오픈 플로우에서 제공하는 다중 플로우 테이블을 활용하게 되는데 그림 4와 같이 개별 플로우 방식으로 동작할 때는 규칙 영역의 구성요소로 패킷이 들어온 스위치 포트와 근원지, 목적지 MAC MAC 주소를 모두 사용하는 개별 플로우 테이블을 활용하고, 임계치를 초과하여 그룹 플로우 방식으로 전환된 경우에는 포트번호와 목적지 주소가 같은 패킷을 그룹화 하여 플로우 정보를 짓아하는 그룹 플로우 테이블을 활용하도록 하였다. 플로우 수에 따라 테이블을 전환하는 방법은 Table\_Id를 구분하는 것으로 구현하였다.

Single Flow Table					
Cookie	Duration	Table_Id	n_packets	priority	in_port
dl_src	dl_dst	Idle_Timeout	Hard_Timeout	actions	

Group Flow Table					
Cookie	Duration	Table_Id	n_packets	priority	
in_port	dl_dst	Idle_Timeout	Hard_Timeout	actions	

그림 4. Flow Table 구성  
Fig. 4 Flow Table Construction

#### 3.3 플로우 모니터링 방법

패킷이 스위치로 들어오게 되면 스위치와 컨트롤러 사이에는 그림 5와 같은 메시지 교환과정이 일어나게 되는데 제안기법을 구현하기 위해서 플로우 정보 획득을 위해 스위치에서 컨트롤러로 Packet-In 메시지를 전송하게 되는데 이 메시지를 전송할 때 각 스위치의 전체 플로우 수를 컨트롤러로 전송하게 되면 컨트롤러가 스위치의 플로우 수를 판단하여 개별 플로우로 처리하도록 할지 그룹 플로우로 처리하도록 할지를 결정하여 스위치로 제어 명령을 내리게 된다[4].

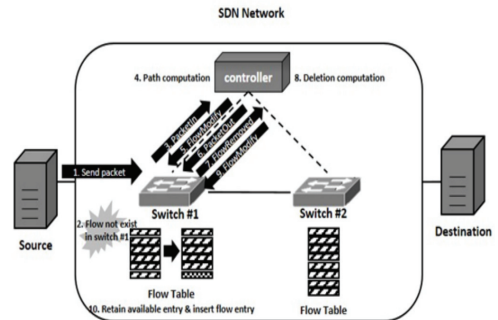


그림 5. 스위치와 컨트롤러 사이의 메시지 교환과정[4]  
Fig. 5 Message exchange process between switch and controller[4]

### IV. 실험 결과

#### 4.1 실험환경 구성

DFM 기법을 활용한 부하분산의 효과를 확인하기 위해서 그림 6과 같이 패킷 생성기의 역할을 하는 Host PC와 OpenFlow 1.3버전이 적용된 SDN 스위치 5대, 각 스위치를 중앙에서 통제하는 컨트롤러 1대로 구성하였고 컨트롤러의 운영체제는 오픈소스 기반의 Ryu-Controller를 활용하였다. 전체 플로우 수의 변화

에 따른 제안기법의 효과를 확인하기 위해서 Host PC 수를 변경해가며 실험을 진행하였다. 실험환경 시스템 구성은 표 1에 요약하였다.

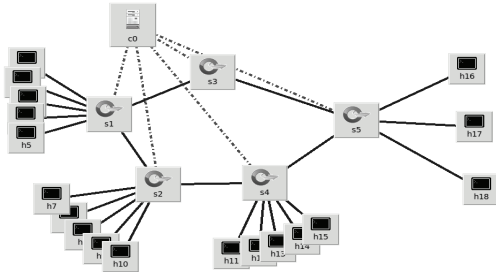


그림 6. 실험환경 Topology 구성

Fig. 6 Experimental environment topology configuration

표 1. 실험환경 시스템 구성  
Table 1. Experimental environment system configuration

Parameter	Value
CPU	Ryzen5 5600X
RAM	32GB
OS	Ubuntu 20.04(64bit)
Virtual Network	Mininet
Software SDN Switch	OpenFlow switch 1.3
SDN Controller	Ryu-controller

### 4.2 실험 방법

실험 간 비교 대상은 개별 플로우 방식으로만 처리하는 개별 플로우 단독 방식으로 선정하였고 제안 기법을 적용했을 때와 전체 플로우 수의 변화, 패킷 처리 속도의 2가지를 서로 비교하여 제안 기법의 성능을 측정한다. 플로우를 발생시키는 방법은 네트워크 내 연결되어 있는 Host PC 간에 ICMP 패킷을 상호 교환하는 방식으로 발생시키고 전체 네트워크의 Host PC의 수를 조절하는 방식으로 네트워크의 규모를 구분하여 묘사하였다.

처음으로 전체 플로우 수의 비교는 먼저 대조기법을 적용했을 때의 전체 플로우 수를 측정하여 비교 기준으로 설정하고, 동일 네트워크 환경에서 임계치 플로우 수를 50%에서 80%까지 10% 단위로 변경하여 설정한 뒤 제안기법을 적용했을 때 전체 플로우 수의 변화를 측정하여 상호 비교함으로써 제안 기법

을 적용했을 때 플로우 수를 얼마나 효과적으로 줄여서 제어할 수 있는지를 검증한다.

두 번째로 제안 기법과 대조기법 간 패킷 처리속도의 차이를 비교하기 위해서 동일 네트워크 환경에서 임의의 Host PC 2대를 선정하여 ICMP 패킷 50개를 송수신하여 평균 처리속도를 측정하여 비교하는 방식을 선택하였다. 여기서 스위치 별로 전체 플로우 수가 다르기 때문에 어떤 Host PC를 선정하는지에 따라 패킷 처리속도의 차이를 보일 수 있어서 같은 스위치에 연결되어 있는 Host PC 간 패킷교환에 대한 경우는 제외하였고 서로 다른 스위치에 연결되어 있는 Host PC 간 패킷교환에 대한 처리속도를 비교하였다.

### 4.3 전체 Flow 수 측정

첫 번째 실험에 관한 결과로써 제안기법과 대조기법을 적용 시 전체 플로우 수의 변화를 측정한 결과는 그림 7과 같다. 그래프의 5가지의 선 중에서 맨 위의 동그라미로 표시되어 있는 선이 기준이 되는 대조기법의 전체 플로우 수를 나타내고 있는데 임계치를 50%에서 80%로 변경하여 제안기법을 적용했을 때 모든 경우에서 대조기법에 비해 최소 10%에서 최대 30% 까지 적은 플로우 수로 전체 네트워크를 통제할 수 있음을 알 수 있었다.

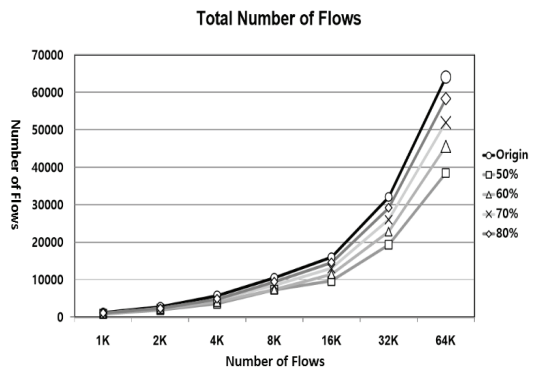


그림 7. 전체 Flow 수의 변화 비교

Fig. 7 Comparison of changes in the total number of flows

### 4.4 패킷 처리속도 비교

다음으로 제안기법과 대조기법 간 패킷 처리속도를 비교한 결과는 그림 8과 같다. 전체 플로우 수가 10K 미만의 소규모 네트워크 환경에서는 대조기법이 제안



기법 보다 평균 5~10% 정도 빠른 처리속도를 보였지만, 10K 이상의 대규모 네트워크 환경에서는 제안 기법의 처리속도가 대조기법의 처리속도를 앞서게 되어 제안기법의 패킷 처리속도가 더 좋아진다는 것을 알 수 있었다. 그리고 설정한 임계치에 따라 처리속도가 다르게 나타남을 확인하여 사용자의 네트워크 환경과 스위치의 처리능력을 고려해 적정 수준의 임계치를 설정해 제안기법을 적용했을 때 효율적으로 네트워크 부하분산을 달성할 수 있을 것으로 보인다.

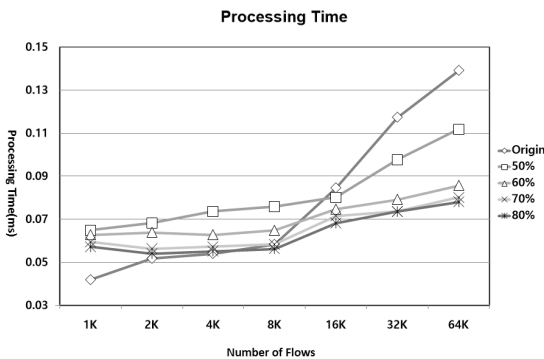


그림 8. 패킷 처리속도 비교  
Fig. 8 Packet processing speed comparison

## V. 결 론

본 논문에서는 SDN 환경에서 개별 플로우 방식으로 처리하는 기존기법의 플로우가 증가했을 때 패킷 처리성능이 저하되는 문제를 DFM 기법을 적용하여 패킷 처리성능이 떨어지지 않도록 플로우 수를 적절하게 유지시키고, 유통되는 플로우 수가 많은 대규모 네트워크 환경으로 갈수록 기존기법에 비해 높은 패킷 처리속도를 보인다는 것을 확인하였다. 실험 결과와 같이 제안 기법은 기존의 개별 플로우 단일 방식에 비해 적은 플로우 수를 가지고 네트워크를 통제할 수 있었고, 데이터 센터와 같은 대규모 네트워크 환경에서 기존 방식보다 빠른 처리속도로 패킷을 처리할 수 있음을 확인하였다.

이를 통해 제안기법은 사용자의 네트워크 환경과 설치되어 있는 스위치의 처리능력을 고려하여 적정 수준의 임계치를 설정했을 때 부하분산에 높은 효율성을 보일 것으로 기대된다. 비록 소규모 네트워크에

서는 제안기법을 적용하지 않았을 때가 오히려 높은 처리속도를 보이긴 하지만 스위치의 성능이 낮은 경우에는 전체 플로우 수 조절을 위해 제안 기법을 적용할 수도 있을 것으로 보이고, 대규모 네트워크에서는 제안기법이 기존기법보다 빠른 처리속도를 보이기 때문에 임계치를 적절하게 설정하여 사용한다면 큰 효과를 볼 수 있을 것으로 기대된다.

향후 연구로, 현재는 그룹 플로우 테이블 구성 간에 근원지 MAC 주소만을 제외하여 그룹화를 실시하여 기본 패킷 처리속도가 낮은 단점을 보완하기 위해 그룹화 하는 기준을 여러 가지로 다르게 적용하여 각 기준 별 패킷 처리속도를 비교하여 보다 나은 처리속도를 보이는 방법을 찾아 제안기법의 효과를 더욱 높여보고자 한다.

## References

- [1] S. Shin, *Evolution of SDN/NFV-based 5G communication network infrastructure*. Seoul: National Information Society Agency, 2019.
- [2] S. Sezer, S. Scott-hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, no. 7, July 2013, pp. 36-43.
- [3] N. Ha and N. Kim, "Efficient Flow Table Management Scheme in SDN-Based Cloud Computing Networks," *J. of Information Processing Systems*, vol. 14, no. 1, 2018, pp. 228-238.
- [4] D. Kim, M. Jo, and B. Lee, "An Efficient Flow Table Management Scheme in SDN," *J. of Korean Institute of Information Technology*, vol. 17, no. 9, 2019, pp. 65-74.
- [5] M. Qilin and S. Weikang, "A Load Balancing Method Based on SDN," *2015 Seventh International Conf. on Measuring Technology and Mechatronics Automation*, Nanchang, China, June 2015, pp. 18-21.
- [6] 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.
- [7] J. Yoon and T. Kwon, "An Efficient Load Balancing

- Technique Considering Forms of Data Generation in SDNs," *J. of Korea Multimedia Society*, vol. 23, no. 2, 2020, pp. 247-254.
- [8] J. Lee and T. Kwon, "Efficient Load Balancing Technique through Server Load Threshold Alert in SDN," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 5, 2021, pp. 817-824.
- [9] K Kim, C. Kim, J. Park, K. Kim, and S. Koh, "A Study on the enhanced performance for packet processing using the multiple flow tables in the OpenFlow based SDN environments," In *Proc. Symp. of the Korean Institute of communications and Information Sciences*, Jeju, Korea, June 2021, pp. 745-746.
- [10] C. Yeon and K. Seok, "A study on radio wave management regulations in the United States to improve the domestic radio wave management system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 15, no. 3, 2020, pp. 379-388.

## 저자 소개



### 김택영(Taek-Young Kim)

2018년 한양사이버대학교 컴퓨터공학과 졸업(공학사)

2021년~ 현재 국방대학교 대학원 컴퓨터공학과 석사과정

※ 관심분야 : 네트워크, SDN



### 권태욱(Tae-Wook Kwon)

1986년 육군사관학교 컴퓨터공학과 (공학사)

1995년 미국 해군대학원 컴퓨터공학 졸업(공학석사)

2001년 연세대학교 대학원 컴퓨터공학과 졸업(공학박사)  
2007년 ~ 현재 국방대학교 컴퓨터공학과 교수

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

