

# SDN 환경에서 효율적 부하분산 기법

이강욱, 권태욱\*

국방대학교

fogis1989@naver.com, \*kwontw9042@korea.kr

## An Efficient Load Balancing in SDN

Lee Kang Uk, Kwon Tae Wook\*

Korea National Defense Univ.

### 요약

본 논문은 SDN(Software-Defined Network) 환경에서 서버들 간의 효율적인 부하분산이 이루어질 수 있도록 제안하였다. 현대의 네트워크는 클라우드, 빅데이터, AI 등 다양하게 발전하여 수많은 데이터를 처리하기 위해 네트워크의 효율이 중요하게 되었다. 기존의 Legacy 환경에서의 네트워크에서는 장비별로 정책을 변경하고 적용하여 효율적인 데이터 처리에 제한이 있었지만 SDN 환경에서는 소프트웨어 중심으로 네트워크를 구성하는 것이 가능하여 효율적인 데이터 처리가 가능하게 되었다. SDN 환경에서는 중앙의 컨트롤러가 다량으로 유입되는 데이터(패킷)에 대하여 서버들의 상태를 확인하고 이를 효율적으로 분배하는 것이 가능한 것이다. 최초 서버별로 일정하게 분배하는 RR(Round Robin) 방식에서는 서버별 성능 차이를 고려하지 못해 과부하가 걸리는 서버가 발생하던 문제를 해결함으로써 데이터 처리의 효율을 올리기 위하여 서버의 성능을 고려하여 데이터를 분배하고 서버의 과부하가 된 시점을 고려하여 부하분산 하는 기법을 제안하였다.

## I. 서론

현대의 네트워크는 클라우드, 빅데이터, AI 등 다양하게 발전하여 수많은 데이터를 처리하는 것이 중요해졌다. 하지만 Legacy 환경에서의 네트워크에서는 이러한 수많은 데이터를 처리하기에는 제한점이 발생하였다. 하드웨어와 벤더에 종속된 Legacy 네트워크로는 일관된 정책을 적용하기 어려웠으며 패킷을 효율적으로 서버 간에 분배하는 것이 제한되었다. 이러한 제한점을 해결할 수 있는 다양한 연구가 진행되었으며 그중에서 SDN(Software Defined Network)[01] 기술이 주목받고 있다. 소프트웨어 중심으로 유연한 대응이 가능하여 많은 분야에서 적용연구[02]가 진행되고 있다.

SDN은 컨트롤러를 통해 소프트웨어 기반의 중앙집중적인 네트워크를 구성하여 벤더에 일관된 정책을 적용하는 것이 가능하다. 이러한 일관된 정책적용으로 효율적인 Load Balancing이 가능해져 특정 서버에 부하가 집중되는 것을 방지하고 패킷의 처리 속도를 증가시키는 연구가 진행되어왔다. SDN 환경에서 기존 서버들 간의 Load Balancing 연구들은 각각의 서버 부하량을 확인하여 특정 임계값에 도달 여부를 확인하여 임계값에 도달한 서버의 패킷 분배량을 줄이는 방식을 택하였다. 이러한 방식을 통해 특정 서버의 성능에 비해 과도한 패킷이 부여되는 것을 방지함으로써 패킷 처리의 효율성을 향상하였다. 하지만 이 방식은 관리자가 서버들의 상태를 파악하여 적절한 임계값을 설정하지 않으면 효율이 떨어지는 제한점이 발생하였다. 이에 본 논문에서는 서버의 임계값을 서버들의 부하량에 따라 유동적으로 변환하여 적용함으로써 서버들의 성능과 유입되는 패킷량을 고려하여 효율적인 Load Balancing이 되도록 하는 방식을 제안하고자 한다.

## II. 관련연구

### 2.1. SDN(Software Defined Network)

SDN의 주요 특징은 제어부(Control Plane)와 전송부(Data Plane)의 분리이다. SDN은 하나의 벤더에 종속되어 있던 제어기능과 전송기능을 분리하여 3개의 계층으로 분리하였다. 네트워크 제어 시에 사용되는 소프트웨어 애플리케이션으로 구성된 응용계층(Application Layer), 전체 네트워크 벤더들에 대한 중앙집중적 제어기능을 실시하는 제어계층(Control Plane), 제어계층의 지시에 따라 패킷을 전송하는 전송계층(Data Plane)으로 구성된다. 3개의 계층 간에는 아래의 그림 1과 같이 응용계층과 제어계층 간에는 Northbound API가 적용되어 서비스에 필요한 다양한 응용의 개발을 지원하고 제어계층과 전송계층 간에는 Southbound API가 적용된다. Southbound API의 대표적인 개방형 프로토콜로는 Openflow 등이 있다.

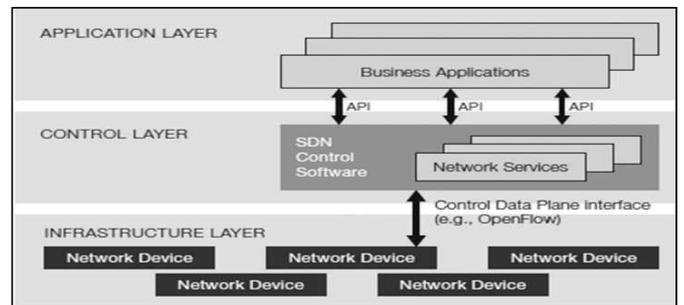


그림 1. SDN 개념 구조도

### 2.2. 부하분산 연구동향 분석

SDN 환경에서 적용되는 부하분산 기법[03]으로는 크게 세 가지로 분류되었다. 첫 번째는 서버들에게 무작위로 할당하는 방법이며 두 번째는

라운드로빈 방식으로 인덱스에 따라 순서대로 부여하는 기법, 마지막으로 세 번째는 서버의 부하 정도에 따라 부하가 가장 낮은 서버에 할당하는 방식이다.

서버의 CPU, 메모리, 연결 가능한 세션 수를 매겨변수로 하여 서버의 부하량에 따른 부하분산을 실시하는 연구[04]가 진행되었다. 이는 서버와 주기적인 통신을 통해 위 매겨변수에 따라 컨트롤러가 서버의 부하를 측정하여 서버 부하도에 따라 분류하여 부하분산 하는 기법이 제안되었다.

그 다음으로는 서버의 주기적인 통신에서 발생하는 비효율적인 트래픽 교환의 최소화를 위해 특정 임계값에 도달 했을 경우에만 컨트롤러가 서버의 부하를 측정하여 서버들의 부하량에 따른 부하분산을 실시하는 방식[05]이 제안되었다. 이때 임계값에 도달한 서버는 패킷의 분배에서 제외되어 다른 서버로만 분배하는 방식이다.

이후 연구된 기법으로는 임계값에 도달한 서버를 부하분산 시 제외하지 않고 전체 서버의 CPU와 메모리 사용률을 확인하여 서버들의 부하 정도에 따른 가중치를 주어 패킷을 분배하는 기법[06]과 임계값에 도달한 서버만 부하도의 가중치 값에서 절반만 패킷을 할당하여 이전 기법보다 빠르게 회복되도록 하는 기법을 제안하였다.

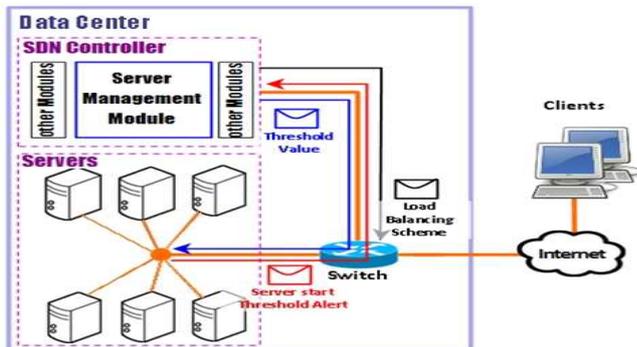
### III. 제안하는 효율적 부하분산 기법

#### 3.1. 기본개념

본 논문은 가변적인 임계값을 설정하여 임계값에 도달 시 서버들의 부하 상태를 확인하고 부하 상태에 따른 가중치를 부여하여 부하분산을 실시하는 새로운 기법을 제안한다. 서버의 CPU와 메모리 사용률을 계산하여 서버의 부하를 계산하여 임계값에 도달한 서버가 있는 경우 모든 서버의 부하를 확인하여 패킷 분배의 가중치를 부여한다. 임계값에 도달한 서버는 가중치를 부여한 값의 절반으로 패킷을 할당하여 부하도가 내려갈 수 있게 한다. 이때 서버들의 부하 값과 임계값의 차이를 계산하여 임계값을 조정함으로써 이후 임계값에 도달하는 경우가 적어지며 보다 효율적인 부하분산이 이루어지도록 한다. 이와 같은 과정을 반복하여 임계값을 재설정하여 최적의 부하분산이 이루어지는 임계값을 설정하게 된다.

#### 3.2. 세부수행절차

실험환경 구성으로는 아래의 그림 2와 같이 각기 다른 성능을 가진 서버 6개가 SDN 환경에서 유입되는 패킷들이 컨트롤러에 의해 부하분산을 받는 구조이다. 컨트롤러는 부하분산 방식을 스위치에 입력하고 스위치는 컨트롤에서 지시한 방식에 따라 유입된 패킷을 6개의 서버에게 나눠서 분배한다. 이때 임계값의 초기값은 60%를 설정한다.



Flow Chart는 아래의 그림 3과 같다. 최초에는 라운드로빈(Round Robin) 방식에 의해 부하분산이 이루어진다. 라운드로빈 방식은 패킷을 순서대로 돌아가며 서버들에게 배정하는 방식으로 서버 성능의 차이에 따

라 서버에 발생하는 부하 정도에 차이가 발생하게 된다. 패킷이 유입량을 지속적으로 증가시켜 임계값을 넘어서도록 환경을 구성한다. 그러면 서버의 CPU 및 메모리값을 계산하던 서버가 임계값에 도달하게 되어 이를 컨트롤러에 보고한다. 보고받은 컨트롤러는 모든 서버의 부하량을 보고 받게 되고 다음과 같이 서버의 패킷 할당량을 계산한다.

1) 임계값 도달 서버 : 현재 부하량 × 0.5

2) 다른 서버 : 현재의 부하량 × (1 + ((임계값 - 부하량) / 100))

다음으로 서버 부하율의 평균을 이용하여 임계값을 다시 설정한다. 이때 서버 부하율의 평균과 임계값의 차이가 클수록 임계값의 증가량이 늘어나고 서버 부하율의 평균과 임계값의 차이가 작을수록 임계값의 증가량이 줄어들므로써 서버의 상태를 고려한 임계값이 설정될 수 있도록 한다. 기존의 임계값에서 서버 부하율의 평균과의 차이의 절반을 더해준다.

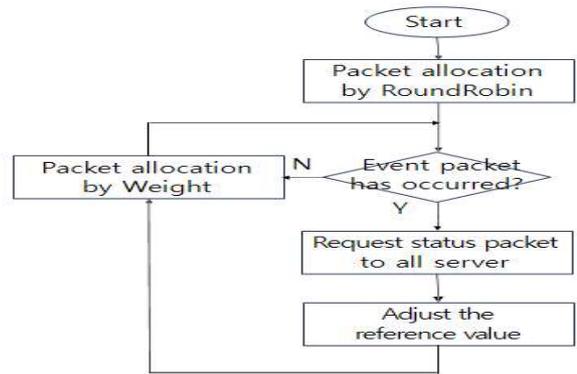


그림 3. 제안 기법의 Flow Chart

### IV. 결론

본 논문에서는 SDN 환경에서 서버들의 부하정도를 기반으로 가변적인 임계값을 적용하여 효율적인 부하분산 기법을 제시하였다. 이전 기법들과는 달리 가변적인 임계값을 적용하여 패킷이 유입될수록 최적의 임계값이 설정되어 임계값에 도달하는 서버가 적어지며 특정 서버에 부하가 발생하는 경우가 적어져 패킷의 처리 속도가 증가할 것으로 판단된다. 향후 실험 및 연구를 통해 제안한 기법의 효율성을 입증할 계획이다.

### 참고 문헌

[01] Open Networking Foundation, "Software Defined Networking : The Form for Networks", OFN White Paper, (2012)

[02] IDC, "Worldwide Data center Software-Defined Networking Forecast, 2019-2023," Nov, 2019

[03] Li, L., & Xu, Q. (2017, July). Load balancing researches in SDN: A survey. In 2017 7th IEEE ICEIEC (pp. 403-408). IEEE.

[04] Deepalakshmi, P. (2018). DServ LB: Dynamic server load balancing algorithm. International Journal of Communication Systems, 1(32)

[05] 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. 817-814.

[06] K. Lee and T. Kwon, "Server state-based weighted load balancing techniques in SDN environments" J. of The Korea Institute of Electronic Communication Sciences, vol. 17, no. 6, 2022, pp. 1039-1046.