

# 데이터 센터 네트워크의 특성을 고려한 네트워크 트래픽 매트릭스 생성 방법

한윤선<sup>01</sup>, 유재형<sup>2</sup>, 홍원기<sup>1,2</sup>

<sup>1</sup> 포항공과대학교 정보전자융합공학부

<sup>2</sup> 포항공과대학교 컴퓨터공학과

{seon054, styoo, jwkhong}@postech.ac.kr

## 요 약

데이터 센터 네트워크 트래픽은 인터넷에서 발생하는 트래픽과 다른 특성을 지니고 있으나, 대부분의 트래픽 생성기는 인터넷 트래픽의 특성만을 반영하고 있다. 데이터 센터에서 발생하는 네트워크 트래픽의 특성을 반영하여, 유사한 네트워크 트래픽을 가상으로 생성하는 방법을 제안한다. 제안하는 방법은 네트워크 구조 파악, 거시적 특성 주입, 플로우 생성, 미시적 특성 주입, 플로우 형태 성형, 트래픽 매트릭스 생성의 과정의 6 단계로 이루어진다. 제안하는 방법은 Python 프로그래밍 언어를 통해 작성되었으며, Mininet 과 iPerf 를 연동하여 구현되었다. 생성된 유사 트래픽은 실제 데이터 센터에서 수집된 트래픽과 유사한 특성을 가지고 있음을 비교하여 검증 하였다.

## 1. 서론

클라우드 컴퓨팅 기술의 발전과 더불어 대용량 멀티미디어 콘텐츠의 증가, 빅 데이터 분석 기술의 도입 등의 추세에 따라 기업, 정부 기관, 대학 등 다양한 기관 및 단체에서는 데이터 센터를 앞다투어 구축하고 있다. 더불어, 새로이 설치되는 데이터 센터는 기존에 비하여 규모가 커지고 있는 양상을 보이고 있다 [1]. 데이터 센터를 설치 함에 있어, 새로이 설치되는 데이터 센터 및 네트워크의 특성을 검증을 통해 요구사항들이 적절하게 반영되었는지를 판단하기 위한 방법이 요구된다.

이러한 상황에서 데이터 센터 네트워크의 성능을 파악하기 위한 효율적인 방법 중의 하나로서, 가상 네트워크 트래픽을 생성하고 이를 대상 데이터 센터 네트워크에 송출하여 그 양태를 관찰하는 방법이 있다. 하지만, 데이터 센터 네트워크에서 발생하는 네트워크 트래픽은 기존의 인터넷 백본 네트워크에서 발생하는 트래픽과 다른 특성들을 지니고 있지만, 기존에 개발된 대부분의 네트워크 트래픽 생성 도구 및 방법들은 인터넷 백본에서 발생하는 인터넷 트래픽 특성만을 반영하도록 설계되어 있어 데이터 센터 네트워크를 대상으로 하는 경우 적합하지 않다. 또한, 다수의 노드들로 구성되는 데이터 센터 네트워크의 특징을 반영하기 위해서는 일반적으로 단일의 노드 또는 장치로 구성되는 트래픽 생성 방법 및 장치는 데이터 센터 네트워크의 특성 검증 목적으로는 적합하지 않다.

가장 기초적인 네트워크 트래픽 생성 방법은 데

이터 센터 또는 네트워크를 구성하는 서버들로부터 실제 트래픽을 수집하여 원하는 형태 또는 특성을 반영하도록 가공하여 사용하는 것이지만, 이 방법의 경우 비효율적이며 생성하는 방법도 복잡하다. 더불어, 트래픽 수집을 위한 장치 및 방법이 필요하며, 정보 보안과 관련된 문제로 인해 실제 데이터를 수집하여 사용하는 것은 매우 어렵다. 이러한 이유로 인해 데이터 센터 네트워크 트래픽이 가지는 특성을 가지도록 임의로 데이터 센터 네트워크 트래픽을 생성하는 방법이 필요하다. 본 고에서는 실제 데이터 센터 내에서 발생하는 네트워크 트래픽과 유사한 특성들을 지니는 가상 트래픽 매트릭스의 생성 기법을 제안한다.

제안하는 트래픽 매트릭스 생성 방법은 여러 선행 연구 [1-3]에서 파악된 데이터 센터 네트워크 트래픽의 특성을 반영하여 유사 네트워크 트래픽을 생성하는 일련의 과정이며, 이때 네트워크 트래픽의 특성을 적용하는 기본 단위는 트래픽 플로우이다. 트래픽 플로우에 특성을 주입 함으로써 실제 네트워크에서 발생하는 것과 비슷한 특성을 지니도록 하는 것에 본 제안 방법의 핵심이다. 이러한 특성의 종류로는 플로우의 발생 빈도, 시간 별 플로우의 발생 빈도의 차이, 내부 랙에서 흐르는 플로우의 비율, 플로우의 길이, 플로우의 크기가 있다. 제안하는 방법은 일련의 과정을 통해 유사 네트워크 트래픽에 대한 전체 정보를 담고 있는 트래픽 매트릭스를 생성하게 된다. 생성된 트래픽 매트릭스를 이용하여 기존에 개발된 트래픽 생성기를 통해 실제 네트워크 패킷을 생성할 수 있다. 또한, 생성 과정을 특정 단위 시간 마다 주기적으로 반복 수행하여 시간에

흐름에 따른 네트워크 트래픽의 변화를 표현 할 수 있다. 제안된 방법을 통해 생성된 트래픽 매트릭스를 일련의 도구를 사용하여 트래픽을 생성하였을 경우, 생성된 네트워크 트래픽은 실제 데이터 센터에서 추출된 네트워크 트래픽과 유사한 특성을 가지고 있음을 비교를 통해 검증하였다.

## 2. 관련 연구

데이터 센터 (Data Center)는 일반적으로 다수의 서버 컴퓨터와 이들 사이의 연결을 위한 네트워크로 구성되는 시설을 지칭한다. 근래에 설치되는 데이터 센터는 수천 개 이상의 서버 컴퓨터로 구성되는 경우가 많으며, 이를 지원하기 위한 대용량 네트워크를 필수적으로 요구한다 [4]. 이러한 네트워크를 구성하기 위해 일반적으로 복수개의 최상위 노드로 구성된 트리 구조의 형태로 구성된다. 하지만, 이러한 네트워크 구조는 구성하고 관리함에 있어 제한적인 용량 및 확장성, 높은 설비투자비용 (CAPAX)과 운용비용 (OPEX)등과 같은 한계점을 지니고 있다. 이러한 한계를 극복하기 위해, 데이터 센터 네트워크 구성을 위한 새로운 형태의 토폴로지들이 제안되고 있다. 저렴한 스위치를 이용하여 다양한 경로를 생성하는 Clos 네트워크에 기반한 Fat-Tree [4], VL2 [5], PortLand [6]와 같은 토폴로지가 있다.

데이터 센터의 구축에 대한 인기에도 불구하고, 데이터 센터 네트워크에서 발생하는 트래픽의 특성에 대해서는 많이 알려져 있지 않다. Benson의 연구 [1, 2]에서는 대학 네트워크, 사내 기업 네트워크, 상용 클라우드 네트워크에서 발생하는 트래픽과 SNMP 로그를 수집하여 특성을 분석하였다. Kandula의 연구 [3]에서는 1500 여개의 서버 컴퓨터로 구성되는 데이터 센터를 대상으로 2 개월 동안 자체개발한 모니터링 프로그램을 통해 수집한 데이터를 바탕으로 데이터 센터 네트워크의 특성을 분석하였다.

현재 다양한 트래픽 생성 기법과 도구들이 연구되었으며, 그 중 몇몇의 제품화되어 있다. [7]에서는 네트워크 트래픽 생성 기법에 따라 다양한 트래픽 생성 방법들을 요약하여 몇 가지 범주로 분류하였다. iPerf [8]와 같은 최대 수용 생성(Maximum Throughput Generator)은 종단 사이의 성능(End to End Performance)를 측정하기 위해 주로 사용된다. 다음으로, 일반적인 트래픽의 통계적인 특성을 반영하기 위해 확률적 모델 (Stochastic Model)에 기반한 생성 기법 [9]이 있다. 마지막으로, 실제 네트워크에서 수집되는 트래픽과 통계적으로 유사한 특성을 지니도록 상위 수준의 모델과 자동화된 기법을 사용하는 방법들이 있다. HARPOON [10]은 다양한 플로우 수준의 특성들을 반영하기 위한 트래픽 생성 도구이다. 이외에도 다양한 생성도구 들이 존재한다. 하지만, 이러한 네트워크 트래픽 생성 기법 및 도구들은 인터넷 백본 네트워크에서 관찰되는 트래픽을 대상

으로 하여, 데이터 센터 네트워크에서 발생하는 트래픽의 특성을 반영하고 있지는 않다.

## 3. 네트워크 트래픽 생성 기법

제안하는 트래픽 생성 기법은 기존의 인터넷 백본 네트워크에서 발생하는 트래픽이 아닌 데이터 센터 네트워크에서 발생하는 트래픽의 특성들을 반영하여 유사한 트래픽을 생성하는 것을 목표로 한다. 데이터 센터 네트워크의 특성을 고려한 트래픽 매트릭스를 생성한 후, 생성된 매트릭스를 참조하여 iPerf [8]와 같은 도구에 명령을 내림으로써 실제 패킷을 생성하게 된다. 트래픽 매트릭스는 주어진 특정 시간에 데이터 센터 네트워크에 흐르고 있는 모든 트래픽의 크기를 가지고 있으며, (송신 서버, 수신 서버, 트래픽량)의 3 가지 요소로 이루어진 튜플 (Tuple)의 집합으로 표현된다. 본 과제에서 제안하는 방법을 수용하기 위한 시스템의 구조는 그림 1 에 도식화 되어 있다.

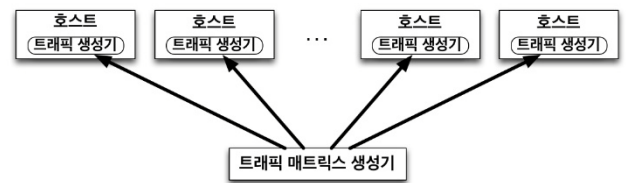


그림 1. 제안 시스템 구조

제안하는 방법은 트래픽 매트릭스를 생성하기 위해 6 단계를 거치게 된다. 각각의 단계는 네트워크 구조 파악, 거시적 특성 주입, 플로우 생성, 미시적 특성 주입, 플로우 형태 성형, 트래픽 매트릭스 생성의 과정을 거치게 된다. 제안하는 방법의 데이터 센터 네트워크의 특성을 반영하기 위한 가장 기본적인 단위는 플로우이며, (송신 서버 IP 주소, 수신 서버 IP 주소, 송신 포트 번호, 수신 포트 번호, 프로토콜)이 동일한 패킷의 집합을 지칭한다. 생성되어야 하는 플로우의 개수, 플로우의 지속시간, 플로우의 크기, 수신 서버와 송신 서버와 같은 특성들을 반영하게 된다. 다양한 데이터 센터 네트워크 특성을 반영하여 특정 시간 대의 트래픽 매트릭스를 생성하고, 이를 참조하여 네트워크 패킷을 생성한다. 이러한 과정을 반복하여 실제 데이터 센터 네트워크에서 발생하는 것과 유사한 트래픽을 생성하게 된다.

### 3.1 네트워크 구조 파악

트래픽 매트릭스를 생성하기 위한 첫 번째 단계는 네트워크 구조 파악 단계로써 생성되어야 하는 트래픽 매트릭스의 대상이 되는 네트워크의 토폴로지 정보를 분석하여 대상 네트워크의 특성을 파악하기 위한 단계이다. 네트워크 토폴로지 정보를 표현하기 위해서, 그래프 구조를 사용하며 이는  $G(V,$

E)로 표현된다. G에서 Vertex의 집합을 지칭하는 V는 호스트, 스위치, 라우터에 해당하며, Edge의 집합에 해당하는 E는 각 Vertex 사이의 링크를 의미한다. 이 단계에서는 그래프 G뿐만 아니라 각 장치 및 링크의 능력을 표현하기 위해 추가적인 메타 데이터를 입력으로 받아 들이는데, 이는 V와 E에 대해 장치의 종류, 링크 용량과 같은 정보들을 포함하게 된다.

입력을 통해 추출하게 되는 관계정보는 랙을 이루는 호스트 집합, 서브넷을 이루는 호스트 집합이 있다. 이러한 정보를 표현하기 위해, 출력으로 Hosts()와 HList(s, d) 함수를 반환한다. Hosts()함수는 대상 네트워크의 전체 호스트의 리스트를 반환하며, HList(s, d)는 주어진 호스트 s에서 d만큼의 홉 거리(Hop count)를 가지는 호스트의 리스트를 반환하는 함수이다. HList(s, d) 함수를 사용하여 동일한 논리적 특성을 가지는 호스트들을 파악할 수 있다. 예를 들어, 호스트 A와 동일한 랙에 위치한 호스트들은 HList(A, 1)을 통해 파악할 수 있다.

### 3.2 거시적 트래픽 특성 주입

이 단계는 생성되어야 하는 플로우의 개수를 결정하기 위한 단계이다. 일반적으로, 데이터 센터 네트워크는 시간대 별로 다른 사용량을 보여주는데, 이는 생성되어야 하는 플로우의 개수에 영향을 미치게 된다. 예를 들어, 주간 데이터 센터 사용량은 새벽 시간 보다 많은 패턴을 보여준다. 본 고에서는 시간대별 트래픽 변동 및 평일, 주말의 변동량을 고려하였다. 이러한 트래픽의 변동량을 고려하여 생성되어야 하는 플로우의 개수를 생성하기 위해,  $\lambda$ 와  $U(t)$ 를 입력으로 받아 들인다.  $\lambda$ 는 단위 시간(초) 동안 데이터 센터의 평균 플로우 생성 개수를 지칭하며,  $U(t)$ 는 특정 시간대 t에서 생성되는 플로우의 개수를  $\lambda$ 와 비교하였을 때 비율을 지칭한다. 본 단계의 출력은 주어진 시간 t에서 생성되어야 하는 플로우의 개수  $N_f$ 로 표현되며, 프아송 확률 분포(Poisson distribution)를 따르는 난수 생성을 통해 얻을 수 있다.

$$N_f \sim f(k; \lambda \cdot U(t)) = \Pr(X = k)$$

### 3.3. 플로우 생성

거시적 트래픽 특성 주입 단계를 통해, 주어진 시간 t에 생성되어야 하는 플로우의 개수를 결정하였다. 이 과정에서는 생성되어야 하는 각각의 플로우를 생성하는 과정으로써, 플로우를 결정하는 5가지 속성인 (송신 호스트, 수신 호스트, 송신 포트, 수신 포트, 프로토콜)을 결정하게 된다. 새롭게 생성되는 플로우의 집합은  $F_{new}$ , t-1 시간에 존재하는 플로우의 집합은  $F_{t-1}$ 으로 표현된다. 따라서, 주어진 시간 t에 존재하는 플로우의 집합  $F_t$ 은 다음과 같이 표현된다.

$$F_t = F_{t-1} \cup F_{new}$$

이 과정의 다른 중요한 입력은  $R_{int}$ 와  $R_{tcp}$ 이다.  $R_{int}$ 는 대상 트래픽 네트워크에서 전체 플로우 대비 랙 내부에서 흐르는 플로우의 비율을 지칭한다. [6,7]에서는 약 80%의 플로우가 랙 내부에서 흐르는 것을 관찰 하였다.  $R_{tcp}$ 는 전체 플로우에서 TCP 프로토콜을 사용하는 플로우의 비율을 지칭한다.  $R_{int}$ 와  $R_{tcp}$ 는 주어진 랙 내부에서 흐르는 플로우의 개수와 사용하는 프로토콜을 결정하기 위해 사용된다.

각 플로우를 생성하기 위한 첫 번째 과정은 송신 호스트와 수신 호스트를 결정하는 것이다. 송신 호스트 s는 전체 호스트들 중에서 하나를 임의로 선택한다. 송신 호스트 d는  $B(\infty, R_{int})$ 의 분포를 따르는 베르누이 시행(Bernoulli trial)을 통해 결정하게 된다. 이때 d가 같은 랙에 위치한다면, HList(s, 1)을 통해 반환되는 호스트 중 하나를 임의로 선택한다. d가 랙 외부에 위치하고 있다면, {Hosts() - Hlist(s, 1)} 중에서 하나의 호스트를 임의로 선택한다. 송신 포트와 수신포트는 각각 (1, 65535)의 범위에서 임의로 선택하게 된다. 마지막으로,  $B(\infty, R_{tcp})$ 를 따르는 베르누이 시행을 통해 해당 플로우가 TCP 또는 UDP를 사용하는 지 여부를 결정하게 된다. 이때, 플로우의 중복을 막기 위해서, 새롭게 속성을 결정하는 플로우  $f_i$ 는 다음과 같은 조건을 만족하여야 한다. 조건을 만족하지 못한다면 송신 호스트를 결정하는 과정부터 다시 반복함으로써 중복되지 않은 플로우 속성을 결정한다.

$$f_i \nexists f_j \in (F_{t-1} \cup F_{new})$$

### 3.4. 미시적 트래픽 특성 주입

본 단계에서는 각각의 플로우의 지속시간과 크기를 결정하는 것을 목적으로 한다. 하지만, 플로우의 지속시간과 크기는 서로 독립적인 관계에 있지 않다. 플로우의 지속시간과 크기 사이의 관계의 확률분포를 보여주는 결합분포함수(Joint distribution)를 모델링하는 것은 매우 어렵다. 이러한 문제를 해결하기 위해, 본고에서는 지속시간과 크기의 연관성을 표현하기 위해, 먼저 플로우의 지속시간에 대한 분포함수를 통해 지속 시간을 결정하였고, 이 결과를 인자로 받아 플로우의 크기를 결정하는 확률 분포함수를 이용하여 플로우의 크기를 결정하도록 하였다. 이러한 방법을 통해, 플로우의 지속시간과 크기 사이의 정확한 확률분포함수를 구하지 않더라도 유사한 결과를 반환할 수 있는 확률분포함수를 설계할 수 있다.

[1, 2]에서는 80%의 플로우가 11초 이하의 지속시간을 가지고 있고, 0.1%에 해당하는 플로우들은 200초 이상 지속된다는 것을 관찰 하였다. 이러한 현상을 가장 잘 표현하는 확률 분포함수는 파레토 확률 분포 (Pareto distribution)이다. 파레토 확률 분포 함수의 인자로 사용되는 형상 모수 (shape

parameter)  $a_p$ 와 척도 모수 (scale parameter)  $M_p$ 를 결정함으로써 플로우의 지속시간에 해당하는 확률 분포 함수를 결정할 수 있다. 플로우의 지속시간에 해당하는 확률 변수 (Random variable)  $D$ 는 다음과 같은 분포를 따르게 된다.

$$D \sim \bar{F}(x) = \Pr(X > x) = \begin{cases} \left(\frac{x}{M_p}\right)^{a_p} & \text{for } x \geq M_p \\ 1 & \text{for } x \leq M_p \end{cases}$$

이렇게 생성된 확률 분포 함수를 이용하여 플로우의 지속시간에 해당하는 난수를 생성할 수 있다. 이를 이용하여  $\text{InitTime}(f_i)$ 와  $\text{Duration}(f_i)$  함수를 반환한다.  $\text{InitTime}(f_i)$ 는 플로우  $f_i$ 의 생성 시간을 반환하며,  $\text{Duration}(f_i)$ 는  $f_i$ 의 지속시간을 반환하는 함수이다.

[1, 2]에서는 80%의 플로우가 10KB 이하의 크기를 가지고 있음을 관찰하였다. 더불어, 트래픽의 상위 10%에 해당하는 플로우들에 의해 발생함을 관찰하였다. 이러한 특성을 가장 잘 표현하는 확률 분포는 파레토 확률 분포이다. 하지만, 플로우의 지속시간과 크기는 서로 독립적이지 않으며, 비례관계 (Proportional relationship)에 있지만 정비례 하지는 않는다. 따라서, 플로우의 지속시간과 크기에 관한 결합분포 함수를 얻어야 한다. 하지만, 플로우의 지속시간과 크기의 결합분포 함수를 설계하는 것은 매우 어렵다. 이러한 문제를 해결하기 위해, 플로우의 크기를 결정함에 있어 웨이블 확률 분포 (Weibull distribution)를 사용한다. 이때 웨이블 확률 분포의 형상 모수와 척도 모수를 플로우의 지속시간에 따르는 함수가 되도록 설계하였다. 형상 모수  $k$ 를 반환하는 함수는  $W_a(\text{Duration}(f_i))$ 이며, 척도 모수  $a$ 를 반환하는 함수는  $W_M(\text{Duration}(f_i))$ 이다. 따라서, 플로우의 크기를 나타내는 확률 변수  $S$ 는 다음과 같이 표현된다.

$$S \sim f(x; a, k) = \frac{k}{a} \left(\frac{x}{a}\right)^{k-1} e^{-\left(\frac{x}{a}\right)^k}$$

확률 변수  $S$ 를 따르는 난수 생성을 통해 플로우의 크기를 결정할 수 있으며, 플로우  $f_i$ 의 크기를 반환하는 함수인  $\text{Size}(f_i)$ 를 생성할 수 있다.

### 3.5. 트래픽 형태 성형

각각의 플로우는 실행하는 응용에 따라 다양한 형태의 패턴을 보인다. 대표적으로 ON-OFF 패턴은 ON에 해당하는 기간 동안 많은 트래픽을 전송하며 OFF에 해당하는 기간 동안에는 적은 트래픽을 전송한다 [4]. 이러한 다양한 플로우의 행동적 특성을 반영하기 위해, 본 고에서는  $\text{shape}(x)$  함수를 도입하였다.  $\text{Shape}(x)$  함수의 특성은 (0,1) 범위에서 연속적인 함수이며,  $\int_0^1 \text{shape}(x) = 1$ 의 조건을 만족하여야 한다. 그림 2는 다양한 형태의  $\text{shape}(x)$  함수를 보여준다.  $\text{shape}(x)$  함수를 각 플로우의 지속시간  $[0,$

$\text{Duration}(f_i)$ ]에 투영(Projection) 함으로써, 각 시간 동안에 플로우가 전송 해야 하는 트래픽의 크기를 결정할 수 있다. 특정 시간 동안 전송되어야 하는 트래픽의 크기  $ts$ 는 다음과 같은 수식을 통해 결정된다. 여기서,  $t_a$ 는 플로우의 생성시간,  $t_b$ 는 종료시간,  $a = \frac{t_a - \text{InitTime}(f_i)}{\text{Duration}(f_i)}$ ,  $b = \frac{t_b - \text{InitTime}(f_i)}{\text{Duration}(f_i)}$  지칭한다.

$$ts = \int_a^b \text{shape}(x) dx$$

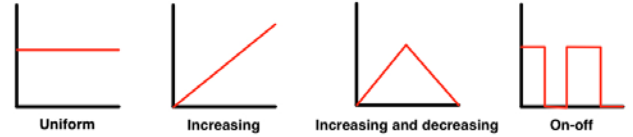


그림 2. 다양한 형태의 shape 함수

### 3.6. 트래픽 매트릭스 생성

트래픽 매트릭스는 특정 단위시간 동안 호스트 사이에 흐르는 트래픽의 크기를 표현하는 것으로서, 일반적으로 (송신 호스트, 수신 호스트, 크기)로 표현된다. 앞선 과정들을 거쳐, 특정 시간대에 존재하는 플로우들을 생성하였으며 이를 이용하여 플로우 매트릭스를 생성하게 된다. 플로우 생성정보에는 송신 호스트, 수신 호스트 정보와 단위 시간동안에 전송되어야 하는 트래픽의 크기가 명시되어 있다. 따라서 이를 종합하여 트래픽 매트릭스를 생성하게 된다.

## 4. 구현 및 검증

본고에서 제안하는 방법은 Python 프로그래밍 언어를 사용하여 구현되었으며, Numpy 와 Random 라이브러리를 사용하여 각 확률 분포에 해당하는 난수를 생성하도록 구현되었다. 표 1에는 각 확률 분포 함수에서 사용된 인자들이 정리되어 있다.

표 1. 각 단계 별 출력, 확률분포 모델과 인자 설정

출력	확률분포	인자
생성 플로우 개수	Poisson	$\lambda = 190$
내부 플로우 비율	Bernoulli	$R_{int} = 0.8$
TCP 플로우 비율	Bernoulli	$R_{tcp} = 0.85$
플로우 지속시간	Pareto	$a_p = 2.7, M_p = -1$
플로우 크기	Weibull	$W_a(d) = \begin{cases} 4 & \text{for } d < 8 \\ 0.25 & \text{for } d \geq 8 \end{cases}$ $W_M(d) = 750$

제안된 방법이 데이터 센터에서 발생하는 트래픽과 유사한 트래픽을 생성하는지 검증하기 위해, Mininet [11]과 iPerf [8]를 사용하였다. Mininet 은 스위치 및 호스트들을 가상으로 생성할 수 있도록 하

는 에뮬레이터이며, iPerf는 Mininet과 연동하여 트래픽을 생성할 수 있게 해주는 도구이다.

비교에 사용된 트래픽은 [4]에서 사용된 것과 동일한 것으로서, 미국에 위치한 모 대학의 데이터 센터에서 실제로 수집된 것이다. 수집된 트래픽은 500여대의 호스트들로 구성되어 있다. 수집된 트래픽은 38355개의 TCP와 UDP 플로우를 포함하고 있으며, 총 41,446,637KB 트래픽 크기를 보였다. 제안된 방법의 검증에 사용된 네트워크 토폴로지는 432대의 호스트들로 구성되는 데이터 센터 네트워크를 가정하였다. 생성된 트래픽은 38701개의 플로우를 포함하며, 45,353,574KB의 트래픽을 생성하였다. 그림 3은 수집된 트래픽과 생성된 트래픽의 플로우 지속시간 분포를 보여준다. 그림 4는 수집된 트래픽과 생성된 트래픽의 플로우 시간에 따른 플로우 크기의 분포를 보여준다. 이를 통해, 제안하는 방법이 실제 데이터 센터에서 수집된 트래픽과 유사한 특성을 가지고 있음을 확인 할 수 있다.

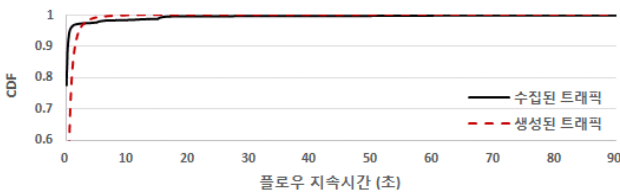


그림 3. 플로우 지속시간 비교

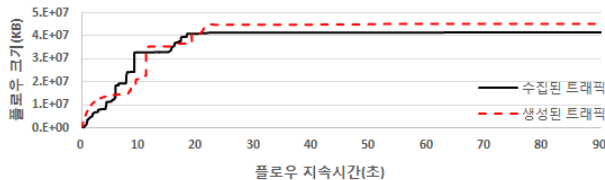


그림 4. 플로우 지속 시간에 따른 크기 비교

## 5. 결론 및 향후 계획

본 고에서는 트래픽 매트릭스를 이용하여 데이터 센터에서 발생하는 트래픽과 유사한 특성을 가지는 트래픽을 가상으로 생성하는 방법을 제안하였다. 제안하는 방법은 패킷을 바로 생성하는 것이 아니라, 플로우 수준에서 트래픽의 특성을 반영하고, 이를 종합하여 트래픽 매트릭스를 생성하여 패킷을 생성하는 생성도구를 이용하여 유사 트래픽을 생성하였다. 트래픽의 생성 과정은 총 6 단계로 이루어져 있으며, 생성된 트래픽은 Python, Mininet, iPerf의 도구를 이용하여 구현되었다. 생성된 트래픽은 실제 데이터 센터에서 수집된 트래픽과 유사한 특성을 가지고 있음을 확인 하였다.

향후 계획으로 각 확률 분포 모델의 인자들을 결정함에 있어 실험적 방법을 사용하였는데, 이를 체계화 하여 원하는 특성을 가지는 트래픽을 생성할 수 있도록 하는 수학적 기법의 도입에 대해 연구를 진행 중이다. 다음으로 현재 반영된 트래픽의

주된 특성은 플로우의 지속시간과 크기인데, 이것을 확장하여 플로우의 폭발성(Burstiness) 및 플로우 사이의 관계와 같은 특성들을 추가적으로 고려한 모델을 개발한 계획이다. 마지막으로, 트래픽 매트릭스 정보로부터 패킷을 생성하는 도구의 개발을 추진할 계획이다.

## 5. 참고문헌

- [1] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. New York, NY, USA: ACM, 2010, pp. 267–280.
- [2] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," SIGCOMM Comput. Commun. Rev., vol. 40, no. 1, pp. 92–99, Jan. 2010.
- [3] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements & analysis," in Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference. New York, NY, USA: ACM, 2009, pp. 202–208.
- [4] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication. New York, NY, USA: ACM, 2008, pp. 63–74.
- [5] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," SIGCOMM Comput. Commun. Rev., vol. 39, no. 4, pp. 51–62, Aug. 2009.
- [6] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: A scalable fault-tolerant layer 2 data center network fabric," SIGCOMM Comput. Commun. Rev., vol. 39, no. 4, pp. 39–50, Aug. 2009.
- [7] S. Molnar, P. Megyesi, and G. Szabo, "How to validate traffic generators?" in Communications Workshops (ICC), 2013 IEEE International Conference on, 2013, pp. 1340–1344.
- [8] M. Gates, A. Tirumala, J. Dugan, and K. Gibbs, Iperf version 2.0.0, Part of Iperf's source code distribution, NLANR applications support, University of Illinois at Urbana-Champaign, Urbana, IL, USA, May 2004. [Online]. Available: <http://iperf.sf.net>
- [9] "MGEN - The Multi-Generator Toolset," <http://manimac.itd.nrl.navy.mil/MGEN/>, Feb. 2004.
- [10] J. Sommers, H. Kim, and P. Barford, "Harpoon: A flow-level traffic generator for router and network tests," SIGMETRICS Perform. Eval. Rev., vol. 32, no. 1, pp. 392–392, Jun. 2004.
- [11] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. NY, USA: ACM, 2010, pp. 19:1–19:6.