http://dx.doi.org/10.7236/JIIBC.2013.13.2.53
JIIBC 2013-2-8

Fat-Tree에서의 새로운 패킷 단위 부하분산 방식

A New Packet-level Load-balancing Scheme for Fat-Trees

임찬숙*

Chansook Lim

요 약 본 데이터센터 네트워크를 위한 대표적인 토포로지들 중 하나인 Fat-Tree는 같은 출발지/목적지에 대해 다중 경로를 갖는다. 또한 같은 홉 수를 갖는 다중 경로의 지연시간은 주로 장비의 큐에서의 지연시간에 의해 좌우된다. 그러나 대부분의 기존 부하 분산 방식들은 이러한 특성을 이용하지 못하고 패킷의 순서 바뀜 현상을 막기 위해 플로우 단위로 부하분산을 한다. 드물기는 하지만 지금까지 제안된 패킷 단위의 부하분산 방식들은 고비용의 전송계층 프로토콜의 사용을 전제로 이루어진다. 본 논문에서는 Fat-Tree의 특성을 이용하여 패킷의 순서 바뀜을 최소화하면서도 패킷 단위로 부하를 분산하여 네트워크의 활용률을 높이는 새로운 부하분산 방식을 제안한다. 모의실험 결과는 제안된 방식이 플로우 단위의 무작위 Valiant 부하방식이 가장 좋은 성능을 보일 때만큼의 TCP 성능을 제공할 수 있음을 보여준다.

Abstract A Fat-Tree topology has multiple paths between any pair of hosts. The delay for the multiple paths with an equal number of hops depends mainly on the queuing delay. However, most of the existing load-balancing schemes do not sufficiently exploit the characteristics of Fat-Tree. In most schemes load-balancing is performed at a flow level. Packet-level load-balancing schemes usually require the availability of special transport layer protocols to address packet reordering. In this paper, we propose a new packet-level load-balancing scheme which can enhance network utilization while minimizing packet reordering in Fat-Trees. Simulation results show that the proposed scheme provides as high TCP throughput as a randomized flow-level Valiant load balancing scheme for a best case.

Key Words : data center network, fat-tree, TCP, packet reordering, load balancing

Ⅰ. 서 론

최근에 제안된 대표적인 데이터센터 네트워크 토포로지들은 공통적인 독특한 특성을 갖고 있는데 이러한 특성을 고려한 네트워크 프로토콜들에 대한 연구가 한창 진행 중이다. 종래의 데이터센터 네트워크가 계층적이었던 것과 달리 최근에 제안된 데이터센터 네트워크들은[1][2][3] 어떤 두 대의 서버에 대해서도 여러 개의 경로를 제공할 수 있는 구조를 갖는다. 이 다중 경로를 가장 잘 활용할 수 있는 방안은 패킷 차원에서 여러 경로로 트래픽을 분산하는 것이지만 대부분의 기존 방식들은 부하분산을 패킷단위로 하지 않고 플로우 단위로 한다. 그 이유

*정회원, 홍익대학교 컴퓨터정보통신공학과
접수일자 : 2013년 2월 15일, 수정완료 : 2013년 3월 15일
게재확정일자 : 2013년 4월 12일

Received: 15 February 2013 / Revised: 15 March 2013 / Accepted: 12 April 2013
*Corresponding Author: chansooklim@hongik.ac.kr
Dept. of Computer & Info. Communications Engineering, Hongik University, Korea

는 수신 호스트에 패킷들이 순서가 바뀌어 도달하는 현상이 빈번히 발생하면 TCP의 성능이 극히 저하되기 때문에 패킷의 순서 바뀜 현상을 막기 위해서이다.

플로우 단위의 부하분산을 하는 몇 가지 대표적인 예를 살펴보자. 우선 데이터센터 네트워크에서 기본적으로 제공되곤 하는 ECMP 방식이 실제로 사용될 때에는 비용이 같은 여러 경로를 통해 패킷 단위로 부하 분산을 하기보다 플로우 단위로 부하를 분산하는 방식을 취한다. Valiant 부하 분산 방식이 사용될 때에도 플로우 단위로 부하를 분산하도록 사용되었다[2]. 또한 트래픽 요구량에 관한 사전정보가 있다는 가정 하에 트래픽이 적은 경로를 활용하기 위한 플로우 스케쥴링 방안이 제안되기도 하였다[6].

대부분의 패킷 단위 부하분산 방식들은 패킷 순서 바뀜 현상을 처리하기 위한 TCP 버전을 필요로 하는데 다중경로를 위한 전송계층 프로토콜로서 최근에 가장 주목을 받은 프로토콜 중 하나가 MPTCP이다[4]. MPTCP는 한 TCP 플로우 내에 여러 부플로우(subflow)를 만들어 각 부플로우를 다중경로 중의 한 경로에 할당하여 독립적으로 혼잡제어를 하도록 한 TCP 버전이다. 그러나 실제 환경에 구현될 때에는 해결해야 할 문제점이 많다[5]. 또한 MPTCP가 다중경로를 활용할 수 있는 여러 상황에서 유용하지만 가장 효과적으로 동작할 수 있으려면 멀티호밍(multi-homing)이 제공되어야 한다[4].

본 논문에서는 Fat-Tree에서 패킷 단위로 부하 분산을 하면서도 패킷 순서 바뀜 현상을 최소화하여 MPTCP와 같은 고비용의 전송계층 프로토콜을 필요로 하지 않는 새로운 방식을 제안한다. 이 방식의 기본 아이디어는 Fat-Tree의 Core 스위치들을 겹치지 않는 그룹들로 나눠놓고 한 플로우가 통과할 Core스위치들을 선택할 때 그룹 하나를 선택하고 그 그룹에 속하는 Core스위치들로 패킷을 분산하는 것이다. ns-2를 이용한 간단한 모의실험 결과는 본 논문에서 제안하는 방식이 플로우 단위의 Valiant 부하 분산 방식이 가장 좋은 성능을 보일 때와 거의 같은 성능을 얻음을 보여준다.

본 논문의 구성은 다음과 같다. 2절에서는 기존 부하분산 방식의 문제점을 논의하고, 3절에서는 본 논문에서 제안하는 새로운 부하분산 방식을 설명한다. 4절에서는 제안한 방법의 효과를 보여주는 모의실험 결과에 관해 논하고, 5절에서는 결론을 맺는다.

II. 기존 부하 분산 방식의 문제점

지금까지 데이터센터 네트워크에서의 경로의 다양성을 활용하여 부하를 분산하기 위해 제안된 방식으로는 ECMP, VL2에서 사용한 Valiant Load Balancing[2], Hedera[6], Packet-level load balancing 방식 등이 있다. 이러한 방식들이 기본적으로 어떻게 네트워크의 활용률을 높이는지, 그리고 부하분산으로 인해 발생할 수 있는 패킷의 순서 바뀜 현상에 대해 어떻게 사전방지 혹은 사후처리를 하는지 우선 살펴본다.

잘 알려진 ECMP 방식은 데이터센터 네트워크에서 기본적으로 제공되곤 하는데 비용이 같은 여러 경로를 통해 정적인 방식으로 플로우 단위의 부하분산을 수행한다.

Hedera[6]는 동적 플로우 스케줄링 방식으로서 스위치들로부터 플로우에 관한 정보를 수집하여 플로우들이 가능한 한 서로 같은 링크를 사용하지 않고 한가한 링크들을 사용하도록 경로를 계산한 후 스위치들이 이 계산결과에 따라 플로우의 경로를 수정할 수 있도록 한다. 이렇게 함으로써 전체적인 네트워크 활용률을 최대화하고자 하는 것이다. 이 방식이 요구하는 비용은 라우팅을 전체적으로 볼 수 있어야 하고 트래픽 요구량(traffic demand)을 미리 알고 있어야 한다는 점이다.

Valiant 부하 분산 방식(VLB)은 Hedera와 같이 중앙집중식의 조정이나 트래픽 엔지니어링을 요구하지 않고 모든 가용 경로에 걸쳐 트래픽을 분산시킨다[2]. 원래 VLB방식은 (a) 네트워크에 hot-spot이 발생하지 않도록 하기 위해 작은 패킷 단위의 무작위 분사가 수행되게 하고 (b) 네트워크로 보내지는 트래픽은 호스(hose) 모델을 따라야 함을 필요로 한다. 그러나 VL2의 경우 VLB 방식을 채택하되 첫 번째 요구조건 (a)를 완화하여 각 패킷 단위가 아닌 각 플로우 단위로 경로를 선택하여 전송한다[2]. VLB에서 각 호스트는 독립적으로 각 플로우를 위한 중간 스위치(Fat-Tree에서는 Core스위치에 해당)를 무작위로 선택함으로써 경로를 선택하게 되는데 이 방식이 사용되면 경로의 길이를 늘임으로 인해 네트워크 용량을 추가적으로 소모하게 되지만 bisection 대역폭을 최대한 활용할 수 있다.

플로우 단위의 VLB 방식은 그림 1이 Fat-Tree에서의 예를 보여주는 바와 같이 플로우들이 가능한 한 서로 같은 링크를 선택하지 않는 최선의 경우를 기대한다. 그러나 스위치를 선택할 때의 무작위성 때문에 그림 2에서 보

여주는 바와 같이 어떤 링크들에는 hot-spot이 형성되는 반면 다른 링크들의 활용률은 극히 낮을 수 있다.

VL2에서와는 달리 Fat-Tree에서 패킷 단위의 VLB를 시험하고 플로우 단위의 VLB와 비교한 연구[7]가 있다. 해당 저자들은 플로우 단위의 VLB가 균일하지 않은 랜덤 트래픽에 대해서는 좋은 성능을 제공하지 못한다고 보고하고 있다. 이들의 연구에서는 TCP가 사용되었으나 패킷 순서 바뀜 현상이 TCP에 미치는 영향을 배제하기 위하여 수신된 패킷들이 TCP에 전달되기 전에 정렬하는 방식을 가정하였다.

이러한 연구 결과들을 종합해볼 때 패킷 단위의 VLB는 네트워크 활용률을 높일 수 있지만 순서가 바뀌는 패킷들을 너무 많이 발생시키므로 별도의 처리를 하지 않는다면 TCP의 시간당처리량에 심각한 영향을 줄 수 있다. 이에 반해 플로우 단위의 VLB는 패킷 순서 바뀜의 문제는 없지만 네트워크 활용률을 저하시키고 혼잡한 링크를 통과하는 플로우들의 지연시간, 시간당처리량 등의 성능을 낮출 수 있음을 알 수 있다.

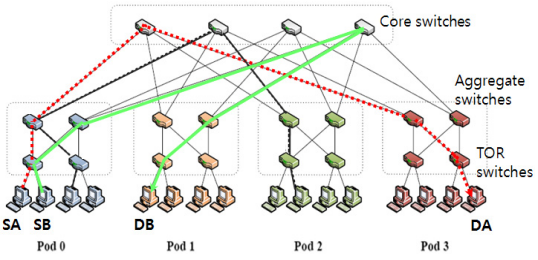

그림 1. 무작위 방식으로 플로우 단위의 부하부산을 하는 최선의 경우의 예

Fig. 1. Illustration of a best case for a randomized flow-level load balancing scheme

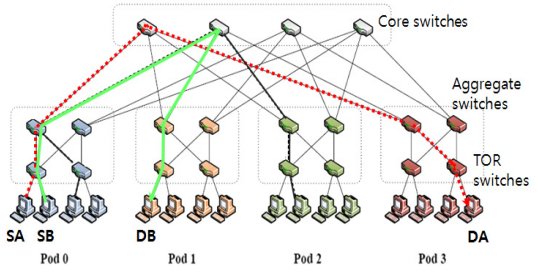

그림 2. 무작위방식으로 플로우 단위의 부하분산을 하는 최악의 경우

Fig. 2. Illustration of a worst case for a randomized flow-level load balancing scheme

Ⅲ. 제안하는 부하분산 방식

본 논문에서는 Fat-Tree를 위한 패킷 단위의 새로운 부하분산 방식을 제안한다. 이 방식은 플로우 단위의 부하분산을 할 때의 문제를 피하기 위해 패킷 단위의 부하분산을 하되 순서가 바뀌는 패킷이 적게 발생하게 하는 것을 목표로 한다. 물론 트래픽 요구량과 같은 사전 정보를 필요로 하지 않는다. 이 방식에서는 가용 다중 경로 중 일부를 사용하여 부하를 분산시키되 같은 플로우에 속하는 패킷들은 거의 같은 부하를 갖는 경로들의 집합을 통해 전달되도록 만들고자 한다. Fat-Tree에서는 주어진 출발지/목적지에 대한 경로들의 길이가 동일하므로 목적지에 도착하는 패킷의 순서가 바뀌게 하는 주요 요소는 큐에서의 지연시간이다. 따라서 한 플로우에 속하는 패킷들이 통과하는 경로들의 부하를 거의 비슷하게 유지하게 함으로써 지연시간의 차이를 최소화할 수 있는 것이다.

이 방식의 기본 아이디어는 여러 종류의 토포로지에 적용될 수 있을 것이나 본 논문에서는 특별히 Fat-Tree에 초점을 맞추고 있으므로 Fat-Tree 토포로지의 특성을 다음과 같이 간략히 설명한다. k-ary Fat-Tree에는 k개의 pod가 있고 각 pod는 각각 k/2개의 스위치를 포함하고 있는 두 개의 계층을 갖고 있다. pod내의 하위 계층에 있는 각 k-port 스위치(edge 계층 스위치 또는 Top-of-Rack(ToR) 스위치)는 k/2개의 호스트와 직접 연결되어 있고 나머지 k/2개의 port들은 상위계층 스위치들(Aggregation 스위치들)과 연결되어 있다. Aggregation 계층 스위치의 k개의 port중 k/2개의 port는 하위계층 스위치와 연결되어 있고 나머지 k/2개의 port는 Core 스위치들과 연결되어 있다. 한 Fat-Tree에는 $(k/2)^2$개의 Core 스위치들이 있다. 각 Core스위치의 k개의 port는 k개의 pod 각각에 한 port씩 연결되어 있다. 각 Core 스위치의 i번째 port는 i번째 pod로 연결되는데 Aggregation 계층 스위치들의 port들은 k/2개씩 차례로 Core 스위치들로 연결된다. 일반적으로 k-port 스위치들로 구성되는 fat-tree는 $k^3/4$개의 호스트를 지원한다.

새로 제안하는 부하분산 방식은 기본적으로 각 스위치들이 목적지까지의 다중경로를 모두 찾아놓는 것을 전제로 한다. Fat-Tree에서 각 호스트는 하나의 ToR 스위치와 연결되며 각 ToR 스위치는 k/2개의 Aggregation 스위치와 연결되어 있고 이를 통해 Core 스위치들에 도달할 수 있다. 따라서 ToR스위치에서는 k/2개의 Aggregation

스위치로 패킷들을 분산시킨다. Aggregation 스위치가 Core 스위치들로 패킷을 분산할 경우에는 모든 Core 스위치로 분산하지 않고 그 중 어느 Core 스위치들을 사용할 것인지 결정한다. 이를 위해 우선 Core 스위치들을 서로 겹치지 않는 그룹들(disjoint groups)로 미리 나누어 놓고 Core 스위치를 선택할 때에는 출발지/목적지 주소를 입력으로 받는 해쉬함수에 의해 그룹을 결정한 후 그 그룹에 속한 모든 Core 스위치들로 라운드 로빈 방식에 의해 패킷들을 분산시킨다. 다시 말해서 2개 이상의 그룹으로부터 Core 스위치들을 택하여 부하를 분산하지 않는다. 결과적으로 임의의 두 개의 플로우를 생각해볼 때 각 플로우에 속하는 패킷들은 같은 집합의 Core 스위치들을 통과하거나 또는 공유하는 Core 스위치가 전혀 없거나 둘 중 하나이다. 이 방식을 사용하면 같은 플로우에 속하는 패킷들이 통과하는 링크들은 모두 거의 같은 부하를 가지므로 지연시간의 차이가 극히 적어지게 되고 이로써 패킷 순서 바뀜 현상을 줄이면서도 패킷 단위의 부하분산을 구현할 수 있게 된다. 본 논문에서 제안하는 부하분산 방식을 알고리즘 1이 보여주는 바와 같이 간략히 정리할 수 있다.

Core 스위치들을 주어진 수의 서로 겹치지 않는 그룹으로 분할함.

각 Aggregation 스위치에서는:
1. 들어오는 패킷의 출발지/목적지 주소에 따라 그룹의 번호가 해쉬함수에 의해 결정됨.
2. 해당 그룹에 속하는 Core스위치들로 패킷을 라운드 로빈 방식으로 전달함.

각 ToR 스위치에서는:
1. 연결된 모든 Aggregation 스위치로 패킷을 라운드 로빈 방식으로 분산함.

알고리즘 1. 새로 제안하는 패킷 부하분산 방식
Algorithm 1. Proposed packet-level load-balancing scheme

이해를 돕기 위해 그림 3은 가장 간단한 Fat-Tree 토포로지인 4-ary Fat-Tree에서 어떻게 부하분산이 이루어지는지 보여주고 있다. 호스트 SA로부터 호스트 DA까지의 서로 다른 경로는 모두 4개 있는데 그 중 2개를 사용하여 패킷이 분산된다. 4-ary Fat-Tree는 매우 단순하므로 Aggregation 스위치에서 패킷을 받았을 때 출발지주소의 패리티에 따라 Core 스위치를 선택하도록 구현할 수도 있는데 그림 3이 그 예이다. 이 예에서 SA->DA 플로우에 속하는 패킷들이 통과하는 Core 스위치들은 SB->DB 플로우에 속하는 패킷들이 통과하는 Core 스위치들과 겹치지 않음을 볼 수 있다. 또한 한 플로우에 속하는 패킷들이 통과하는 경로들의 부하가 비슷할 것임을 알 수 있다.

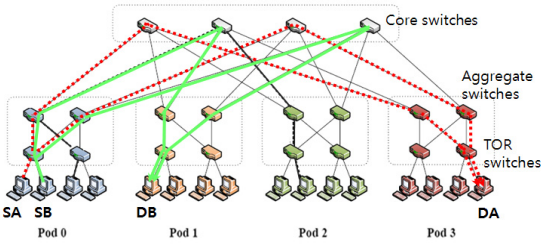

그림 3 본 논문이 제안하는 방식에 의한 부하분산의 예
Fig. 3. Illustration of how the proposed packet-level load balancing scheme works

본 논문과 관련성이 깊은 연구는 [9]에서 찾아볼 수 있다. 기존 연구에서는 모든 가용 다중 경로를 통해 패킷을 분산시키는 "완전 분산(Packet Scatter)" 방식을 시험한 반면 이번 연구에서는 가용 다중 경로 중 일부를 사용하면서도 같은 플로우의 패킷들의 순서 바뀜 현상을 제거하고자 한 것이다. 모든 Core스위치들을 사용하여 패킷을 분산시키는 방식은 사실 이 방식의 일종으로서 모든 Core 스위치가 한 그룹에 속하는 극단적인 경우에 해당한다. 가용 다중 경로를 모두 사용하는 방식은 규모가 큰 Fat-Tree 토포로지에서 링크 문제가 발생할 때 영향을 받을 가능성이 더 크다.

위에서 간단히 언급하였지만 새로 제안하는 방식이 원래 목표한 바대로 부하 분산이 잘 이루어지지 않을 가능성은 링크에 문제가 생겨 같은 플로우에 속하는 패킷들이 통과하는 링크들의 부하 상태가 달라지는 경우에 발생한다. 링크의 문제는 두 부류로 나눠 생각해볼 수 있는데 우선 Core스위치와 Aggregation 스위치를 연결하는 링크의 문제일 경우에는 문제의 링크를 포함하지 않는 다른 그룹을 선택하도록 조치하면 쉽게 해결될 수 있다. 그러나 Aggregation스위치와 ToR 스위치 간의 링크에 문제가 발생하면 같은 그룹에 속하는 경로 중 일부를 사용할 수 없어 남은 링크들에 더 많은 부하를 주게 된다. 이는 문제의 링크를 사용하는 플로우와 같은 경로를 공유하는 플로우에 속한 패킷들이 통과하는 경로 간에 부하의 차이를 초래한다. 따라서 균등하지 못한 부하가 지연시간의 차이를 만들고 패킷의 순서 바뀜 현상을 일으키는 것이다.

이러한 일이 얼마나 자주 발생할 것인가는 데이터 센터에서의 링크 신뢰성(link reliability)에 관한 연구 결과를 통해 추정해볼 수 있다. 최근의 연구 결과[8]에 의하면 데이터센터 네트워크의 장비들은 Top-of-Rack(ToR) 장비를 제외할 때 "four 9's"(즉, 99.99%) 이상의 높은 신뢰성을 보였다. 또한 ToR 장비의 경우에는 비교적 오랫동안 지속되는 문제로 인해 높은 "downtime"을 보이지만 어떤 문제라도 겪은 ToR 장비는 전체의 3.9%에 불과했다고 보고하고 있다. 따라서 링크 문제로 인해 겪게 될 성능 손실은 플로우 단위의 무작위 VLB로 인해 겪게 될 성능 손실보다 적을 것으로 추측할 수 있다.

Ⅳ. 모의실험

본 연구에서는 패킷 순서 바뀜 현상 등을 정확히 관찰하기 위해 ns-2를 사용하였으며 약 10초간의 전송을 시뮬레이션 하였다. 모의실험에 사용된 네트워크는 그림3에서 보여주는 4-ary Fat-Tree 토포로지를 갖고 있다. 각 링크의 대역폭은 모두 1Gbps, 각 링크의 큐 크기는 250패킷, 링크 별 전파 지연시간은 50μs로 설정하였다. 모의실험 지속시간은 약 10초이며 ns-2에서 제공하는 기본적인 TCP-Reno를 사용하였다. 사용된 라우팅 방식은 ns-2에서 기본적으로 제공하는 거리벡터 방식을 사용하였는데 이 방식은 가용 다중경로를 모두 찾아놓는 기능을 가지고 있다. Fat-Tree토포로지에서 ToR 스위치와 Core 스위치는 이미 ns-2에 구현되어 있는 그대로의 ECMP 기능, 즉, 거리(비용)가 같은 최단 경로들에 걸쳐 라운드 로빈 방식으로 패킷을 분산할 수 있는 ECMP 기능을 사용하도록 설정하였다. (ns-2의 ECMP는 데이터센터 네트워크에서의 ECMP와 달리 패킷 단위로 분산한다.) 그러나 Aggregation 스위치에서는 본 논문에서 제안하는 알고리즘대로 Core 스위치를 선택하도록 구현하였다. 앞서 언급한 바 있지만 특별히 이 모의실험의 Fat-Tree는 매우 단순하므로 Aggregation 스위치에서 패킷을 받았을 때 출발지주소의 패리티에 따라 Core 스위치를 선택하도록 구현하였다.

이 모의실험에서는 간단한 비교를 위하여 두 개의 플로우를 사용하여 새로 제안한 부하분산 방식의 성능을 확인하고자 하였으며 그림 1과 그림 2가 보여주는 바와 같이 플로우 단위의 무작위 VLB 방식이 최선의 성능 및 최악의 성능을 보이는 경우와 비교하였다. 그림 4는 모의실험 결과를 보여준다. 본 논문에서 제안하는 부하분산 방식이 플로우 단위의 무작위 VLB방식의 최선의 경우와 맞먹는 성능을 보임을 알 수 있다.

Fat-tree의 규모를 임의의 큰 k-ary Fat-tree로 확장하고 여러 가지 형태의 트래픽 유형을 사용한다 하더라도 제안된 부하분산 방식이 좋은 성능을 보일 것임을 쉽게 예상할 수 있다. 그 이유는 출발지 호스트 쪽의 ToR 스위치로부터 목적지 호스트 쪽의 ToR 스위치까지의 다중 경로 중에서 (k/2)개의 링크 분리 경로들(link-disjoint paths)을 이용하는데 임의의 두 플로우에 대해서 그 두 플로우가 완전히 같은 집합의 Aggretation스위치-Core 스위치 링크를 사용하거나 공유하는 Aggregation스위치-Core스위치 링크가 전혀 없거나 둘 중 하나이므로 각 플로우가 통과하는 다중 경로 간에 부하의 차이가 극히 적기 때문이다. 따라서 링크에 문제가 생기지 않는 한 플로우 단위의 무작위 VLB 방식이 최선의 경우에 보이는 성능에 근접하는 성능을 보일 수 있다.

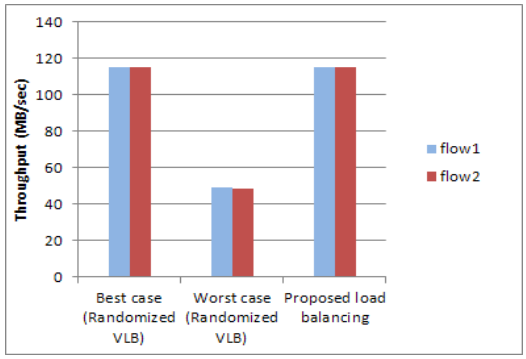

그림 4. 모의실험 결과
Fig. 4. Simulation results

Ⅴ. 결 론

본 논문에서는 데이터센터 네트워크를 위해 제안된 대표적인 토포로지인 Fat-Tree에서 패킷 단위로 부하분산을 하되 패킷 순서 바뀜 현상을 최소화할 수 있는 방식을 새로이 제안하였다. 이 방식은 Aggregation 스위치에서 Core스위치들을 선택하여 패킷을 전달할 때 패킷의 출발지 주소에 따라 Core 스위치의 그룹 중 하나를 택하여 그 그룹에 속한 Core 스위치로 패킷을 분산한다. 간단한 모의실험 결과는 새로 제안한 방식이 플로우 단위의 무작위 Valiant 부하방식이 가장 좋은 성능을 보일 때만

큼의 TCP 성능을 제공함을 보여준다.

데이터센터 네트워크 장비는 신뢰성이 높은 것으로 알려져 있지만 링크에 문제가 발생하면 본 논문에서 제안한 부하방식 하에서는 다중경로 간의 지연시간의 차이로 인해 패킷 순서 바뀜 현상이 일시적으로 발생하거나 TCP 성능이 저하될 수 있을 것이다. 향후에는 이러한 문제에 관련된 trade-off에 관한 분석을 진행할 예정이다.

참 고 문 헌

[1] Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat, “A Scalable, Commodity Data Center Network Architecture,” proceedings of SIGCOMM, 2008.

[2] Albert Greenberg, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, Dave Maltz, Parveen Patel, and Sudipta Sengupta, “VL2: A Scalable and Flexible Data Center Network,” proceedings of SIGCOMM, 2009.

[3] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu, "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," proceedings of SIGCOMM, 2009.

[4] Costin Raiciu, Sébastien Barré, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, Mark Handley, "Improving datacenter performance and robustness with multipath TCP," Proc. ACM SIGCOMM 2011, pp. 266-277.

[5] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, M. Handley. "How Hard Can It Be ? Designing and Implementing a Deployable Multipath TCP," USENIX NSDI'12. San Jose (CA). 2012.

[6] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, Amin Vahdat, “Hedera: Dynamic Flow Scheduling for Data Center Networks,” USENIX NSDI 2010.

[7] Santosh Mahapatra and Xin Yuan, "Load Balancing Mechanism in Data Center Networks," IEEE CEWIT 2010.

[8] Phillipa Gill, NavenDu Jain, Nachiappan Nagappan, "Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications," Proc. ACM SIGCOMM 2011.

[9] Chansook Lim, "Effects of Packet-Scatter on TCP Performance in Fat-Tree," JIWIT Vol. 12, No. 6, 2012.

[10] H. Jo, S-H. Kim, S. K. Lee, “A Strategic Design of Green Data Center : the Case of Data Center in the Domestic Public Sector,” Journal of Korean Institute of Information Technology, vol. 10, no. 4, pp. 143-152, Apr. 2012.

※ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임. (과제번호 2012R1A1A3013408)

저자 소개

임 찬 숙(정회원)

- University of Southern California (박사)
- 홍익대학교 과학기술대학 컴퓨터정보통신공학과 조교수

<주관심분야 : 라우팅, TCP, 네트워크 코딩, 인터넷 측정>