http://dx.doi.org/10.7236/JIWIT.2012.12.6.215
JIWIT 2012-6-27

Fat-Tree에서의 패킷분산이 TCP 성능에 미치는 영향

Effects of Packet-Scatter on TCP Performance in Fat-Tree

임찬숙*

Chansook Lim

요 약 데이터센터 네트워크에서의 병목현상 문제를 해결하기 위해 경로의 다양성을 제공하는 네트워크 구조들이 제안되고 있다. 이렇게 제공되는 다중 경로들을 활용함에 있어 TCP의 성능에 미치는 영향을 고려해야 하는데 이는 같은 플로우 내의 패킷들이 다중 경로를 통해 전송될 경우 패킷 순서 바뀜으로 인해 TCP성능이 저하될 수 있기 때문이다. 지금까지 제안된 대부분의 방식들은 패킷의 순서 바뀜을 막기 위해 사용가능한 경로들 중 하나를 선택하여 플로우를 할당함으로써 부하를 분산한다. 본 연구에서는 경로의 다양성을 제공하는 대표적인 토포로지인 Fat-Tree에서 패킷 단위의 분산방식으로 다중 경로를 이용할 때 패킷 순서 바뀜이 TCP성능에 큰 영향을 미칠 만큼 심하지 않음을 주장한다. 다양한 패턴의 트래픽을 이용한 모의실험 결과는 Fat-Tree와 같은 토포로지에서 큰 비용을 들이지 않고 TCP의 성능문제를 해결할 가능성을 암시한다.

Abstract To address the bottleneck problem in data center networks, there have been several proposals for network architectures providing high path-diversity. In devising new schemes to utilize multiple paths, one must consider the effects on TCP performance because packet reordering can make TCP perform poorly. Therefore most schemes prevent packet reordering by sending packets through one of multiple available paths. In this study we show that packet reordering does not occur severely enough to have a significant impact on TCP performance when scattering packets through all available paths between a pair of hosts in Fat-Tree. Simulation results imply that it is possible to find a low-cost solution to the TCP performance problem for Fat-Tree-like topologies.

Key Words : Data Center Network, Fat-Tree, load balancing, TCP, packet reordering

Ⅰ. 서 론

데이터센터의 규모가 갈수록 커지고 있는 요즈음 데이터센터 네트워크에서의 병목현상을 해결하기 위한 연구들이 한창 진행 중이다. 데이터센터 네트워크에서의 병목현상은 응용프로그램이 만들어내는 결과의 질에 영향을 미치고 기업의 매출에까지 영향을 줄 수 있으므로 해결책이 절실히 요구되는 문제로 인식되고 있다.

종래의 데이터센터 네트워크가 계층적이었던 것과 달리 최근 제안된 데이터센터 네트워크 구조들이[1][2][3] 높은 정도의 경로 다양성을 제공하는 이유도 상기의 병목현상을 완화하기 위함이다. 그러나 풍부한 경로의 다양

*정회원, 홍익대학교 컴퓨터정보통신공학과
접수일자 : 2012년 10월 19일, 수정완료 : 2012년 12월 18일
게재확정일자 : 2012년 12월 14일

Received: 19 October 2012 / Revised: 18 December 2012 / Accepted: 14 December 2012
**Corresponding Author: chansooklim@hongik.ac.kr
Dept. of Computer & Info. Communications Engineering, Hongik University, Korea

성이 제공되어도 이를 잘 활용하지 못하면 소용이 없을 것이다. 직관적으로 생각할 때 경로의 다양성을 가장 잘 활용할 수 있는 방안은 패킷 차원에서 여러 경로로 트래픽을 분산하는 것이지만 대부분의 방법들은 이러한 방식을 기피한다. 예를 들어 데이터센터 네트워크에서 기본적으로 제공되곤 하는 ECMP 방식은 비용이 같은 여러 경로를 통해 패킷 단위로 부하 분산을 하기보다는 플로우 단위로 부하를 분산하는 방식을 취한다. Valiant 부하 분산 방식이 사용될 때에도 플로우 단위로 부하를 분산하도록 사용되었다[2]. 또한 트래픽이 적은 경로를 활용하도록 하기 위한 플로우 스케쥴링 방안이 제안되기도 하였다.[6] 이렇게 부하분산을 패킷단위로 하지 않고 플로우 단위로 하는 이유는 수신 호스트에 패킷들이 순서가 바뀌어 도달하는 현상이 빈번히 발생하면 TCP의 성능이 극히 저하되기 때문에 원천적으로 패킷의 순서 바뀜 현상을 막기 위해서이다.

가장 바람직한 방법은 한 TCP 플로우에 속하는 패킷들이 반드시 같은 경로를 따라 전송되어야 하는 제약사항으로부터 자유로울 수 있으면서도 TCP의 성능이 저하되지 않도록 다중경로를 활용하는 것이다. 다중경로를 위한 전송계층 프로토콜로서 최근 들어 가장 주목을 받는 프로토콜은 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. MPTCP는 표준화 작업도 거쳤으므로 더욱 많은 주목을 받고 있지만 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 하므로 MPTCP를 위한 데이터센터 네트워크 구조가 새로이 제안되기도 하였다[4].

본 논문에서는 경로의 다양성을 제공하는 네트워크 구조 중 가장 대표적인 Fat-Tree에서 데이터센터 내부의 트래픽을 패킷분산(packet scatter)방식으로 모든 가능한 경로로 분산 시켰을 때 패킷 순서 바뀜 현상이 심하지 않음을 모의실험을 통해 보여준다. 이는 모든 트래픽이 패킷분산방식으로 전송될 때 같은 플로우에 속하는 패킷들이 어떠한 경로를 통과하든 Fat-Tree 구조의 특성상 거의 동일한 부하를 받고 있는 스위치들을 통과하게 되기 때문이다. 이러한 결과는 Fat-Tree와 같은 환경에서의 MPTCP의 사용이 지나치게 많은 비용을 지불하는 방법임을 암시하며 기존의 표준 TCP를 변형하여 신속한 재전송으로 인한 성능감소 현상을 완화하는 저비용의 방법으로도 좋은 성능을 얻을 수 있다는 가능성을 보여준다.

본 논문의 구성은 다음과 같다. 2절에서는 데이터센터 네트워크에서의 경로의 다양성에 관한 기존의 연구에 대해 논의하고, 3절에서는 Fat-Tree에서의 완전한 패킷분산방식의 사용을 제안하며, 4절에서는 모의실험 결과와 의미를 논의하고, 5절에서는 결론을 맺는다.

II. 데이터센터 네트워크에서의 경로의 다양성

본 연구의 배경이 된 데이터센터 네트워크에서의 경로의 다양성과 관련된 기존의 연구들을 살펴보고자 한다. 우선 네트워크 토포로지에 관한 연구들을 살펴보고 네트워크의 활용도와 지연시간 측면에서의 개선을 위한 부하분산방식과 플로우 스케줄링 방식, 그리고 다중경로의 활용을 목적으로 제안된 MPTCP에 대해 살펴본다.

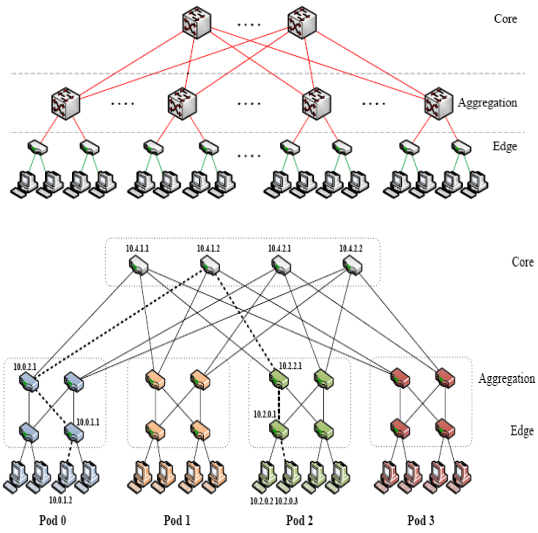

그림 1. 종래의 데이터센터 네트워크 토포로지(위)와 Fat-Tree 토포로지(아래). (본 그림은 참고문헌 [1]로부터 발췌된 것임.)

Fig 1. Conventional Data Center Network Topology(top) and Fat-Tree Topology (bottom)

1. 토포로지에 관한 연구들

종래의 데이터센터 네트워크는 그림1의 예시와 같이 대체로 계층적 구조를 가지고 있다. 데이터센터 트래픽의 추세를 볼 때 데이터센터 내부의 트래픽이 갈수록 더 많은 비중을 차지하고 있다. 이러한 추세는 계층 구조의 네트워크에서 트래픽이 매우 고르지 않게 분포하게 만들 것으로 예상되고 있어 이러한 문제를 해결하기 위한 여러 가지 토포로지들이 제시되었다. 우선 Clos 토포로지 부류에 속하는 Fat-Tree[1]는 데이터센터 네트워크 내의 어떠한 호스트 쌍에 대해서도 full 대역폭을 제공하기 위해 일정한 규칙을 따르는 토포로지를 가진다(그림1 참조). VL2[2]는 Fat-Tree와 유사한 토포로지를 가지지만 Fat-Tree에 비해 스위치들 간에 더 적은 수의 고속의 링크들을 사용하며 다중경로를 활용하여 hot-spot을 없애기 위해 Valiant 부하분산 방식(Valiant Load Balancing)을 플로우 단위로 적용한다. BCube[3]의 경우에는 계층구조를 버리고 트래픽을 릴레이하기 위해 호스트도 함께 사용하는 하이퍼큐브(hypercube) 형의 토포로지를 형성한다. 더 최근에는 정형화된 토포로지를 사용하던 기존 방식들과는 달리 Top-of-Rack 스위치들을 랜덤 그래프 토포로지 형태로 연결하는 방안[8]이 제안되었다.

2. 플로우 단위의 패킷 분산

데이터센터 네트워크 내에 형성된 다중경로를 활용하기 위한 방안은 패킷 순서 바뀜을 막아야 한다는 조건으로 인해 상당한 제약을 받는다. 오래된 방식인 ECMP는 같은 비용의 경로들에 걸쳐 트래픽을 분산시키지만 패킷 순서가 바뀌어 수신되는 것을 피하기 위해 플로우 단위로 경로를 결정하여 전송하는 방식이 더 많이 사용된다.

Valiant 부하 분산 방식(VLB)은 중앙집중식의 조정이나 트래픽 엔지니어링 없이 모든 사용가능한 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 각 패킷 단위가 아닌 각 플로우 단위로 경로를 선택하여 전송한다[2]. 패킷 순서 바뀜을 피하기 위해 VLB의 첫 번째 요구조건 (a)를 완화한 것이다. VLB에서는 bisection 대역폭의 최대화를 위해 경로의 길이가 늘어나는 비용을 감수한다. 즉 각 호스트는 독립적으로 각 플로우를 위한 중간 스위치(Fat-Tree에서는 Core스위치에 해당)를 무작위로 선택함으로써 경로를 선택하는데 이 방식이 사용되면 지연시간을 증가시키거나 경로의 길이를 늘임으로 인해 네트워크 용량을 추가적으로 소모하게 되지만 bisection 대역폭을 최대한 활용할 수 있다. VL2에서는 어떤 출발지 호스트로부터도 중간 스위치까지 경로의 길이가 동일하게 3홉이므로 ECMP를 함께 이용하였다.

3. MPTCP의 비용

MPTCP는 경로 다양성의 활용에 장애요인으로 작용하는 TCP문제를 해결하기 위해 제안된 프로토콜이다. MPTCP처럼 부플로우(subflow)의 개념을 사용하여 각 부플로우마다 따로따로 혼잡제어를 하는 제안들이 과거에도 있었다. 기존의 유사한 제안들과 비교할 때 MPTCP의 특징은 부플로우의 혼잡윈도우의 증가 시 윈도우 크기가 크면 더 빨리 증가하도록 하여 덜 혼잡한 경로로 트래픽을 이동시키는 효과를 얻도록 한다는 점, 그리고 RTT가 다른 경우에 대해 보상을 하며 같은 병목을 지나가는 부플로우들이 정규 TCP 플로우와 경쟁할 때의 공평성을 갖도록 한다는 점을 들 수 있다. 표준화 작업을 통해 더욱 널리 알려지게 된 MPTCP는 데이터센터 네트워크, multihomed 인터넷, 3G와 WiFi의 동시 사용이 가능한 스마트폰 등 다양한 환경에서 유용함이 주장되었다.

그러나 MPTCP는 여러 가지 비용을 요구한다. 우선 MPTCP에 가장 유리한 환경은 멀티호밍 환경이므로 이중 홈(Dual-Homed) Fat-Tree 구조[4]와 같은 새로운 구조가 갖추어졌을 때 최상의 성능을 보일 수 있다. 멀티호밍이 아닌 경우에는 어떤 방식으로든 활용 가능한 여러 개의 경로가 있음을 MPTCP가 파악을 하여 해당 개수만큼의 부플로우를 만들 수 있도록 해야 한다. 데이터센터 환경에서의 MPTCP의 활용방안을 제시한 [4]에서는 ECMP가 여러 개의 경로를 찾아 부플로우들을 임의로 여러 경로에 할당할 때 같은 IP주소에 여러 다른 포트를 열어서 각각의 포트에 부플로우들이 할당되도록 하였는데 이를 위해서는 사용가능한 경로 개수에 관한 정보가 필요하다.

데이터센터 환경이 아닌 일반 인터넷 환경을 위한 시스템에서 구현될 때 MPTCP는 같은 플로우내에 부플로우들을 가져야 하는 특성으로 인해 많은 어려운 문제를 해결해야 한다고 보고하고 있다[5]. 예를 들어 일반 TCP와는 달리 부플로우의 개념을 사용하므로 부플로우의

ID, 일련번호 등의 메타데이터를 어떻게 인코딩해야 할 것인가에 관한 문제를 해결해야 한다. 이는 오늘날의 미들박스(middlebox)들이 패킷들의 전송계층 헤더를 읽고 때로는 수정까지 하며 일반 TCP의 형태에서 벗어나는 패킷을 폐기하는 경우도 있기 때문이다. 또한 MPTCP가 구현되는 운영체제 안에서 부플로우를 위한 버퍼를 관리함에 있어 교착상태에 빠지기 쉬운 문제를 해결해야 한다. 이외에도 MPTCP를 설계함에 있어 고려해야 할 여러 가지 문제들이 보고되었는데[5] 이 중 몇 문제들은 데이터센터 네트워크에도 해당된다. 이러한 문제들을 잘 해결할 수 있다면 같은 플로우 내의 패킷들이 지나가는 여러 경로들의 특성, 즉 경로의 길이나 경로의 가용대역폭, 또는 신뢰성 등이 매우 이질적인 경우에 MPTCP는 특히 유용하다.

4. 패킷분산 방식의 가능성

그렇다면 TCP 성능을 감소시키지 않고 데이터센터 네트워크의 다중경로를 이용할 방법이 플로우 단위의 스케쥴링을 사용하거나 MPTCP를 사용하는 방안 외에는 없을까? 본 논문에서는 실제로 구현되려면 비용이 높은 MPTCP를 사용하지 않으면서도 패킷 차원에서의 분산을 통해 다중 경로를 활용할 수 있는 가능성에 대해 살펴본다. 우리는 특별히 Fat-Tree 환경에서의 패킷 분산에 초점을 맞추어 생각해본다.

Ⅲ. Fat-Tree에서의 패킷 분산과 TCP 성능

1. Fat-Tree의 특징

k-ary Fat-Tree의 구조는 그림1에서 보여주는 바와 같다. (그림1의 아래그림은 k=4일 때의 토포로지를 보여준다.) 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)는 k/2개의 호스트와 직접 연결되어 있고 나머지 k/2개의 port들은 상위계층 스위치들(aggregation 계층 스위치들)과 연결되어 있다. aggregation 계층 스위치의 k-port중 k/2 port는 하위계층 스위치와 연결되어 있고 나머지 k/2 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/2)개씩 차례로 core 스위치들로 연결된다. 일반적으로 k-port스위치들로 구성되는 fat-tree는 $k^3/4$개의 호스트를 지원한다.

Fat-Tree의 장점은 모든 스위칭 요소들이 동일하므로 모든 스위치를 위해 저렴한 상용 부품들을 사용할 수 있다는 점을 꼽을 수 있다. 또한 Fat-Tree는 “rearrangeably non-blocking”의 성질을 가진다. 즉, 임의의 통신 패턴에 대해서도 토포로지 내의 호스트들이 사용할 수 있는 모든 대역폭을 포화시킬 수 있는 경로들의 집합을 갖고 있다. 같은 에지 스위치에 연결된 호스트들은 그것들만의 서브넷을 형성한다. 그러므로 같은 하위 계층으로 연결된 호스트로 가는 트래픽은 스위칭 되며 그 외의 트래픽은 라우팅 된다.

2. FAT-Tree에서의 패킷 순서 바뀜 현상

Fat-Tree의 특성 상 트래픽을 완전히 분산시킬 경우 oversubscription 비율이 1:1이 되는 효과를 얻을 수 있다. oversubscription 비율이란 [호스트 간에 최악의 경우 도달할 수 있는 대역폭의 합]:[토폴로지의 총 bisection 대역폭]을 의미한다. 따라서 oversubscription비율이 1:1이라는 것은 각 호스트가 임의의 다른 호스트와 네트워크 인터페이스의 full 대역폭으로 통신하는 것이 가능함을 암시한다. 그러나 지금까지 제안된 거의 모든 트래픽 분산 방식은 플로우 내의 패킷 순서 바뀜 현상을 방지하기 위해 패킷별로 분산시키는 방법은 피하고 플로우 단위로 분산시키는 방법을 택한다. 패킷별 분산 방식에 비하면 플로우 단위의 분산방식은 트래픽을 모든 네트워크에 고루 분산하지 않을 확률이 크다.

그러나 우리가 Clos 토폴로지의 일종인 Fat -Tree나 VL2의 경우들에 초점을 맞춘다면 이 토포로지들이 어느 쌍의 호스트들에 대해서도 사용가능한 다중경로들의 길이가 같다는 특징을 가진다는 점을 이용할 수 있다. 우리는 이러한 토포로지에서 네트워크 내에 있는 모든 내부 트래픽에 대해서 완전한 패킷 분산 방식(또는 per-packet spreading)으로 전송하면 패킷 순서 바뀜 현상이 심하지 않을 수 있다는 점에 주목한다. 그 이유는 한 출발지/목적지 쌍에 대해 흩어지는 패킷들이 통과하는 스위치들의 혼잡정도가 모두 거의 동일하기 때문이다.

그렇다고 해서 패킷 순서 바뀜 현상이 전혀 일어나지 않는 것은 아닐 것이다. 같은 종류의 스위치나 호스트라 하더라도 처리속도의 차이가 있을 수도 있으며 이 외에도 여러 가지 요인이 패킷의 순서가 바뀌어 수신되도록 작용할 수 있다. 중요한 점은 그러한 요인들로 인해 혼잡윈도우가 심하게 교란되지 않는다면 적은 오버헤드로 네트워크의 활용도를 높일 수 있는 패킷별 분산방식을 사용할 수 있다는 점이다. 또한 패킷 순서 바뀜 현상이 발생하되 정도가 심하지 않을 경우에는 신속한 재전송 방식으로 인한 부작용을 완화한 TCP 버전의 사용만으로도 좋은 효과를 얻을 수 있을 것이다.

본 논문에서는 Fat-Tree에서 완전한 패킷별 분산방식을 사용했을 때의 패킷 순서 바뀜의 정도를 관찰한다. 관찰된 결과는 적은 오버헤드로 패킷 순서 바뀜 현상을 줄이면서 네트워크 활용도를 높일 수 있는 새로운 방안을 설계하는 데에 도움이 될 것이다.

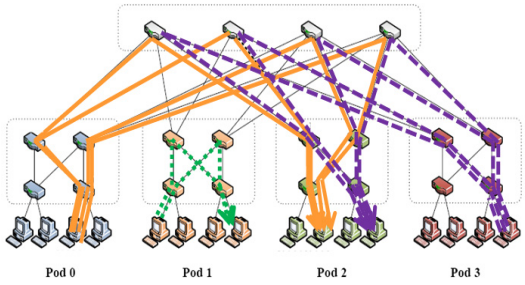

그림 2. Fat-Tree에서의 패킷 분산
Fig 2. Packet Scatter in Fat-Tree

Ⅳ. 모의실험

1. 모의실험 환경 설정

일반적으로 데이터센터 네트워크 관련 모의실험에서는 방대한 양의 트래픽이 발생하므로 ns-2와 같은 패킷 단위의 모의실험은 매우 긴 시간이 걸린다. 이 때문에 플로우 단위의 모의실험방식이 사용되기도 하지만 세세한 부분을 시뮬레이션 할 수 없을 수도 있다는 단점이 있다. 본 연구에서는 패킷 순서 바뀜 현상 등을 정확히 관찰하기 위해 ns-2를 사용하였으며 1초간의 전송만을 시뮬레이션 하였다. 모의실험에 사용된 네트워크는 그림2에서 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 1초이다. 모의실험에는 ns-2에서 제공하는 기본적인 TCP-Reno를 사용하였다.

이 모의실험의 목적은 Fat-Tree에서 패킷이 다중경로로 분산될 때 패킷 순서 바뀜의 정도를 관찰하는 데에 있다. 사용된 라우팅 방식은 ns-2에서 기본적으로 제공하는 거리벡터 방식이며 거리(비용)가 같은 최단 경로 간에 패킷 별로 분산할 수 있는 ECMP 기능을 사용하였다. 따라서 플로우의 출발지와 목적지에 따라 경로의 수가 1개, 2개, 혹은 4개까지 다를 수 있다. 따라서 VL2에서 사용된 VLB방식에서처럼 bisection 대역폭를 최대한 활용하지는 않는다.

모의실험에는 데이터센터 네트워크 관련 연구에서 많이 사용하는 트래픽 패턴[1]이 사용되었는데 크게 세 종류로 분류할 수 있다.

Random방식: 각 호스트는 네트워크 내의 uniform 확률을 가지고 무작위로 선택된 다른 호스트에 전송한다.

Stride(i)방식: 인덱스 x를 가진 호스트는 (x+i)mod 16 인덱스를 가진 호스트로 전송한다. 본 모의실험에서는 Stride(1), Stride(2), Stride(4), Stride(8) 방식을 사용하였다.

Staggered Probability (SubnetP, PodP) 방식: 각 호스트가 subnet내에 있는 다른 host에게 전송할 확률은 subnetP이며 같은 pod내에 있는 다른 호스트에게 전송할 확률은 podP이다. 그리고 1-subnetP-podP의 확률을 가지고 그 밖의 다른 호스트로 전송한다. 우리는 (0.5, 0.3)일 때와 (0.2, 0.3)일 때 두 경우에 대한 모의실험을 수행하였다.

본 모의실험에 사용된 네트워크에는 16개의 호스트가 있으므로 각 방식에서 16개의 데이터 흐름이 생성되며 16개의 플로우가 서로 다른 16개의 출발지를 가지거나 서로 다른 16개의 목적지를 가지도록 생성되었다. 본 모의실험에서는 각 트래픽 패턴에 대해 3번씩 실행하였다.

2. 모의실험 결과

주어진 큐 크기에서는 중간에 패킷이 손실되는 경우가 없었다. 따라서 TCP 시간당처리량은 RTT와 패킷 순서 바뀜 현상에 주로 의존한다.

그림 3은 단일경로 전송과 다중 경로를 통한 packet scatter방식간의 시간당 처리량을 보여준다. 모의실험에 사용한 TCP는 표준 TCP-Reno였음에도 불구하고 다중경로를 통한 패킷전송 방식이 단일경로 전송에 비해 시

간당처리량이 적은 경우는 거의 없다. 패킷의 순서가 바뀌는 경우가 매우 드물기 때문이다. 패킷 순서 바뀜 현상으로 인한 3회 중복승인 수신은 총 13번의 수행(run) 중 다섯 번의 수행에 대해서만 (random1, random3, stride4, stride8, stagger(0.2,0.3)2) 발생하였다. 또한 3회의 중복승인 수신이 가장 많았던 수행의 경우 총 6차례 발생하였으며 심한 순서 바뀜 현상이 발생한 수행은 없었다.

패킷 단위로 사용가능한 모든 경로에 걸쳐 분산시켜도 패킷 순서 바뀜 현상이 예상보다 심하지 않은 이유는 간단하다. 이 방식에서는 같은 플로우에 속하는 패킷들이 서로 다른 경로를 통과하게 되는데 어느 시점에도 각 경로가 부담하고 있는 트래픽의 양이 거의 동일하기 때문이다.

패킷 순서 바뀜 현상을 겪은 플로우들의 불필요한 재전송 및 전송률 감소를 방지하고 공평성을 보장하기 위해서는 신속한 재전송 방식으로 인한 부작용을 완화하는 TCP버전을 사용할 수 있다. 이러한 TCP들은 MPTCP와 달리 기존의 표준 TCP방식을 약간 수정함으로써 구현할 수 있다.

이 모의실험의 주목적은 패킷 순서 바뀜에 대한 관찰이지만 '왜 패킷별로 분산시켰을 때 성능이 더 좋아지지 않는가?' 하는 질문이 생길 수도 있다. 앞서 기술하였듯이 각 트래픽 패턴은 16개의 플로우를 사용하며 16개의 플로우가 서로 다른 16개의 출발지를 가지거나 서로 다른 16개의 목적지를 가지도록 생성되었다. 이러한 환경에서는 각 플로우가 단일 경로를 통과하든, 여러 경로로 분산되든 간에 시간당처리량을 결정하는 병목 링크가 에지(edge)스위치와 호스트를 연결하는 링크이므로 시간당처리량에는 별로 차이가 없다.

플로우 단위로 경로를 할당하는 방식에 비해 패킷 단위로 분산하는 방식이 더 좋은 성능을 보일 수 있는 상황들은 기존의 관련 연구에서 찾아볼 수 있다. VL2 저자들은 ECMP와 VLB가 플로우 단위의 스케줄링을 할 때의 한계로서 거대한 플로우(elephant flow)로 인해 어떤 링크들에는 혼잡현상이 지속되고 다른 링크들의 활용도는 저하되는 현상이 발생할 가능성을 언급한다. 그들이 데이터센터에서 측정할 당시에는 그러한 현상을 발견하지 못했지만 (당시의 측정결과에서 거대한 플로우가 발생하지 않은 이유는 응용프로그램에서 거대한 플로우가 생기지 않도록 분할하여 전송하였기 때문이다) 만일 발생한다면 TCP가 심한 혼잡현상을 겪을 경우 플로우를 다른 경로에 다시 할당하는 방식을 사용할 것을 제안하였다. 물론 추가의 비용을 필요로 하는 방법이다. 또한 Fat-Tree에서 플로우 차원의 VLB와 패킷 차원의 VLB를 비교한 별도의 연구[7]에서도 플로우 차원의 VLB가 균일하지 않은 랜덤 트래픽에 대해서는 좋은 성능을 제공하지 못한다고 보고하고 있다. 이들의 연구에서는 TCP가 사용되었으나 패킷 순서 바뀜 현상이 TCP에 미치는 영향을 배제하기 위해 수신된 패킷들이 TCP에 전달되기 전에 정렬하는 방식을 가정하였다. 이러한 연구들은 TCP의 성능을 최적화할 수 있는 패킷 단위의 부하분산 방식이 필요함을 보여주고 있으며 본 모의실험 결과는 그러한 방식의 설계의 방향을 보여준다고 할 수 있다.

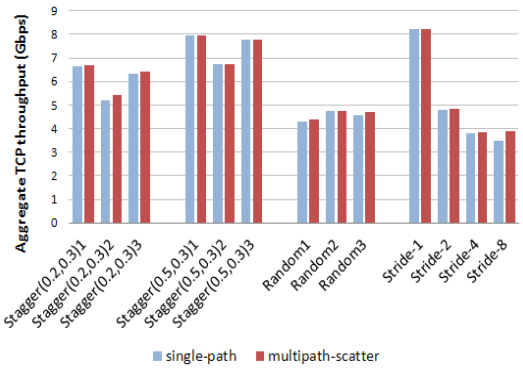

그림 3. Fat-Tree에서의 TCP 시간당 처리량
Fig 3. TCP throughput in Fat-Tree

V. 결 론

본 논문에서는 Fat-Tree에서의 패킷 분산이 어느 정도의 패킷 순서 바뀜 현상을 유발하는가를 모의실험을 통해 살펴보았다. 사용된 대표적인 트래픽 패턴에 대해 심한 순서 바뀜은 발생하지 않았으며 이러한 결과는 오버헤드가 적으면서도 TCP가 패킷 순서 바뀜 현상으로 인해 성능이 급감하는 문제를 해결할 수 있는 방안에 관한 새로운 관점을 제공해주어 이 방안에 관한 연구가 현재 진행 중이다.

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] Al-Fares, Mohammad and Radhakrishnan, Sivasankar and Raghavan, Barath and Huang, Nelson and Vahdat, Amin, “Hedera: dynamic flow scheduling for data center networks,” USENIX NSDI'10, San Jose (CA). 2010.

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

[8] Ankit, Singla, Chi-Yao Hong, Lucian Popa, P. Brighten Godfrey, "Jellyfish: Networking Data Centers Randomly," USENIX NSDI'12, 2012.

[9] 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>