

<http://dx.doi.org/10.7236/JIIBC.2013.13.4.9>

JIIBC 2013-4-2

Fat-Tree에서 링크 장애가 패킷분산방식의 성능에 미치는 영향

Effects of Link Failures on Performance of Packet Scatter Schemes in Fat-Trees

임찬숙*

Chansook Lim

요약 데이터센터 네트워크와 관련된 대부분의 최근 연구들은 Fat-Tree와 같은 대칭성 있는 다중루트트리 토폴로지를 가정한다. 이러한 네트워크에서는 패킷단위의 분산방식이 매우 효과적이지만 장애가 발생할 경우에는 토폴로지의 대칭성이 깨져 TCP의 성능이 저하된다고 알려져 있다. 본 논문에서는 모의실험을 통하여 Fat-Tree에서 링크 장애가 패킷분산방식의 성능에 주는 영향을 다시 조사한다. 모의실험결과는 큰 규모의 Fat-Tree에서는 링크 장애가 발생해도 TCP 성능이 저하될 만큼 패킷 순서 바뀔이 심하게 발생하지 않음을 보여주며 링크 장애에 대비한 복잡한 방식이 반드시 필요한 것은 아님을 암시한다.

Abstract Most of recent studies on data center networks are based on the assumption of symmetrical multi-rooted tree topologies such as a Fat-Tree. While packet scatter schemes are very effective for such a network topology, it is known that various failures can result in an asymmetric topology which degrades TCP performance. In this paper, we reexamine the effects of link failures on packet scatter schemes in Fat-Trees. Our simulation results show that in case of a single link failure in a large-scale Fat-Tree, packet reordering does not occur enough to degrade TCP performance. This implies that we do not necessarily need a complex scheme to make packet schemes robust to link failures.

Key Words : Fat-Tree, multi-path, packet scatter, TCP, link failure

1. 서론

데이터센터 네트워크에서의 TCP 성능에 관한 최근 연구 결과들을 살펴볼 때 Fat-Tree[1]와 같이 다중 루트를 가지며 대칭성(symmetry)이 있는 네트워크 토폴로지를 전제로 하는 연구가 많다. TCP 성능 문제를 해결하고

자 하되 TCP 자체의 변화는 최소화하고 하위 프로토콜 계층의 부하분산 방식을 바꾸려는 노력이 대부분이며 본격적인 실용화를 위해 단순성에 점점 더 초점이 맞추어지는 추세이다.

이러한 추세 가운데 데이터센터 네트워크에서의 패킷 단위의 부하 분산 방식에 대한 연구도 활발히 이루어지

*정회원, 홍익대학교 컴퓨터정보통신공학과
접수일자 : 2013년 7월 9일, 수정완료 : 2013년 8월 14일
게재확정일자 : 2013년 8월 16일

Received: 9 July, 2013 / Revised: 14 August, 2013

Accepted: 16 August, 2013

*Corresponding Author: chansooklim@hongik.ac.kr

Dept. of Computer & Info. Communications Engineering, Hongik University, Korea

고 있다. 최근의 연구결과들^[2-4]은 데이터센터 네트워크 토폴로지의 특징인 대칭성과 극히 짧은 전파지연시간이 다중 경로를 통한 패킷 분산 방식의 실제 사용을 가능하게 함을 보여준다. 이러한 연구들은 다중경로를 통해 패킷단위로 트래픽을 분산해도 TCP 성능에 심하게 영향을 줄 만큼 패킷 순서가 바뀌지 않는다는 것을 보여주면서 과거에 제안된 복잡한 방식을 대체할 새로운 단순한 방식들을 제안하고 있다.

그러나 링크 장애가 발생할 경우 패킷 분산 방식이 의존하던 네트워크 토폴로지의 대칭성(symmetry)이 깨짐으로 인해 같은 플로우에 속하는 패킷들이 통과하는 경로 간에 트래픽이 균일하지 않아 패킷 순서 바뀔 현상이 더 많이 발생할 수 있다. 이 문제에 대한 해결책으로 제안된 방식에는 SRED (selective RED)^[2]가 있다. SRED에서는 정상적인 경우에는 사용할 수 있던 다중경로의 일부를 링크 장애 등으로 인해 사용하지 못하는 플로우들에 대해서는 RED가 동작하는 반면 링크 장애와 상관없이 다중경로를 다 사용할 수 있는 플로우들에 대해서는 계속 droptail 방식이 동작한다. 이 방식은 문제의 링크가 속한 pod와 관련이 없는 플로우들의 성능이 보호된다는 점에서는 좋지만 문제의 pod에서 발생하는 플로우들은 차별받게 된다는 점에서 공평성(fairness)의 정의에 대한 질문을 던진다. 또한 이 방식은 패킷에 추가 정보를 실어 보내야 한다는 비용도 필요로 한다.

이 논문에서 우리는 Fat-Tree에서 단일 링크 장애가 발생할 때 TCP의 성능에 미치는 영향을 다시 조사한다. 특히 Fat-Tree의 대칭성이 깨질 때 트래픽의 편중과 패킷 순서 바뀔 현상이 미치는 영향에 대해 조사하되 세 쌍의 출발지/목적지에 속하는 플로우들에 대한 조사를 추가하였다. 또한 지금까지 간과되었던 큰 k값에 대한 k-ary Fat-Tree에서의 영향을 파악하기 위해 8-ary Fat-Tree에 대한 모의실험을 수행하였다.

가장 의미 있는 모의실험 결과는 4-ary Fat-Tree에서와는 달리 큰 k값의 k-ary Fat-Tree에서는 단일 링크 장애로 인한 패킷 순서 바뀔의 영향이 극히 작아진다는 것이다. 이러한 결과는 실제 Fat-Tree에서 패킷분산 방식을 사용할 때 단일 링크 장애에 대비하기 위하여 반드시 고비용의 방식을 사용할 필요는 없음을 보여준다.

본 논문의 구성은 다음과 같다. 2절에서는 Fat-Tree에서의 패킷분산방식과 링크 장애 문제를 기술하고, 3절에서는 모의실험 결과와 이 결과들이 주는 의미에 대해

분석하며, 4절에서는 결론을 맺는다.

II. Fat-Tree에서의 패킷분산방식과 링크 장애 문제

1. Fat-Tree 토폴로지

데이터센터 네트워크 토폴로지로서 고려되는 대표적인 토폴로지인 Fat-Tree의 구조는 다음과 같이 간략히 요약될 수 있다. 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(Top-of-Rack(ToR) 스위치 또는 edge 계층 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 Core스위치들이 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.

k값이 증가함에 따라 Fat-Tree내에 연결되는 스위치와 호스트, 그리고 링크의 수가 급격히 증가하므로 큰 k값에 대해서 모의실험에 소요되는 시간이 급격히 증가하고 실제 시스템 구현에 많은 비용이 필요하게 된다.

2. Fat-Tree에서의 패킷 분산 방식

인터넷과 같은 정형화되지 않은 네트워크에서와는 달리 Fat-Tree와 같은 네트워크 토폴로지에서는 패킷 분산 방식을 사용해도 TCP의 성능이 크게 저하되지 않는다는 사실이 여러 논문에서 언급되었고 구체적인 패킷 분산 방식이 제안되기도 하였다.

예를 들어 RPS(Random Packet Spraying) 방식^[2]은 각 플로우의 패킷들을 등가 클래스(equivalence class)라고 정의된 모든 등가 경로 상에 있는 모든 링크들 사이에 똑같이 분산한다. 그러면 주어진 플로우에 대해 서로 다른 경로를 지나가는 패킷들이 똑같은 큐잉 지연시간을

겪게 되므로 수신측에 순서가 바뀌어 도착하는 패킷은 소수에 불과하다.

[4]에서는 모든 경로를 다 사용하여 패킷을 분산하기 보다는 Core 스위치를 서로 겹치지 않는 그룹으로 나누어 한 플로우에 속하는 패킷들이 가용 다중 경로의 일부를 사용하여 분산되면서도 패킷의 순서가 바뀌지 않도록 하였다. 이 논문에서는 이를 “부분 패킷 분산(partial packet scatter)”이라고 명명하고 RPS와 같이 가용 다중 경로를 모두 이용하는 방식을 “완전 패킷 분산”이라고 명명한다.

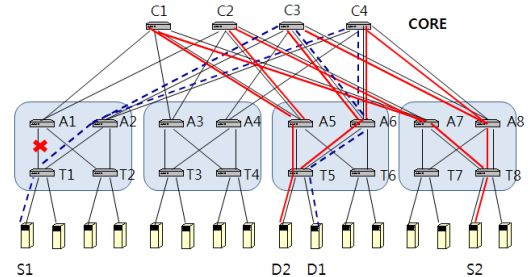
3. 장애가 주는 영향

앞서 언급되었듯이 Fat-Tree와 같은 토폴로지의 대칭성은 한 플로우에 속한 패킷들이 다중경로를 통해 패킷 단위로 분산된다 하더라도 패킷의 순서가 심하게 바뀌지 않도록 해준다. 그러나 데이터센터 네트워크의 스위치나 링크가 장애를 일으킬 경우에는 대칭성이 손상된다. 다시 말해서 한 플로우에 속한 패킷들이 여러 개의 서로 다른 경로를 지날 때 경험하는 트래픽의 혼잡정도가 다를 수 있는 것이다. 따라서 패킷 단위의 분산 방식이 실제로 사용가능하게 되기 위해서는 네트워크 장애에 관한 문제가 해결되어야 한다.

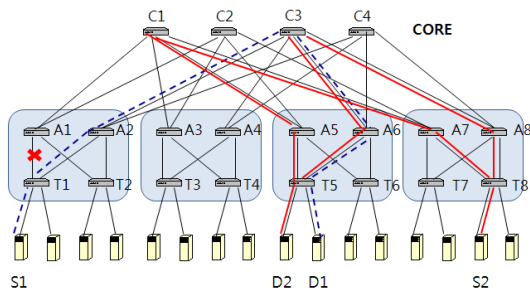
스위치 장애가 발생할 경우 일단 장애가 탐지되면 스위치와 연결된 모든 링크 또한 라우팅에 사용되지 않는다. 예를 들어 한 Core 스위치가 완전히 다운되었다고 하자. 이 상태에서 RPS와 같은 완전패킷분산 방식을 사용할 경우 모든 플로우는 장애가 발생한 Core스위치를 통하는 경로가 없다는 동일한 조건하에서 패킷전송이 이루어진다. 그러나 별도의 조치가 없으면 각 경로로 균일한 양의 트래픽을 분산하게 되지 않게 되므로 이로 인해 패킷 순서가 바뀔 수 있다. [4]에서 제안한 방식과 같이 Core 스위치들의 일부를 사용하여 패킷 분산을 할 경우에는 장애가 발생하지 않은 Core 스위치들로 구성된 집합을 선택하도록 조치할 수 있다.

링크에 장애가 발생할 경우 플로우에 따라서는 한 플로우 내에 속하는 패킷들이 통과하는 경로들의 부하가 서로 다를 수 있다. 링크 하나에 장애가 발생할 때 일어날 수 있는 현상을 보여주는 그림 1의 예시를 살펴보자. 이 예에서는 링크 A1-T1에 장애가 발생하였으며 TCP 플로우의 출발지 호스트인 S1은 문제의 링크가 있는 pod에 있다. 완전패킷분산방식을 사용할 경우 S1에서 D1으

로 가는 플로우들의 패킷이 분산될 수 있는 경로는 2개뿐이다. 그리고 이 플로우들은 S2-D2간의 플로우들의 패킷이 분산되고 있는 4개의 링크들, C1-A5, C2-A5, C3-A6, C4-A6간에 트래픽의 불균형을 초래한다. 즉, 링크 C1-A5, C2-A5에 비해 링크 C3-A6, C4-A6는 더 많은 트래픽을 겪게 되고 이로 인해 S2-D2간의 플로우들은 패킷의 순서가 바뀌어 수신되는 현상을 겪게 된다.



(a) 완전패킷분산(complete packet scatter)



(b) 부분패킷분산(partial packet scatter)

그림 1. 4-ary Fat-Tree 토폴로지에서 링크 장애가 발생 한 경우 두 쌍의 출발지-목적지에 대한 모의실험 설정

Fig. 1. Simulation setup for 2 source-destination pairs in the presence of a link failure in a 4-ary Fat-Tree topology

RPS를 제안한 연구자들은 링크 장애 문제에 대한 해결책으로서 SRED(Selective RED)를 제안하였다[2]. SRED에서는 링크 문제 등으로 인하여 모든 다중 경로를 사용하지 못하는 플로우들의 패킷에 대해서만 RED가 동작하고 모든 경로를 다 사용할 수 있는 플로우들에 속하는 패킷들에 대해서는 여전히 droptail 방식이 동작한다. 어떤 플로우가 다중경로를 모두 사용하지 못함으로 인해 스위치들 간에 큐 길이의 차이를 만들 경우 그 플로우에 속하는 패킷을 더 많이 없애 전송률을 줄이게 함으로써 다른 “정상적인” 플로우들에게 피해를 주지 않게 하기

위함이다. 그러나 이 방식은 링크 장애와는 상관없어야 할 플로우들의 시간당처리량에만 초점을 맞추고 있어 문제의 링크가 경로에 포함되도록 되어 있는 플로우들의 시간당처리량은 대폭 감소하는 형평성의 문제를 갖고 있다.

우리는 RPS와 같은 완전 패킷 분산과^[4]에서 제안한 부분 패킷 분산 방식이 링크 장애로 인해 받는 영향을 조사한다. 이 논문에서는 4-ary Fat-Tree에서 두 쌍의 출발지-목적지에 대해서만이 아니라 세 쌍의 출발지-목적지에 대해서도 링크 장애의 영향을 조사한다. 또한 지금까지는 거의 조사되지 않았던 8-ary Fat-Tree에 대해 링크 장애의 영향을 조사한다. 이 결과는 큰 k값의 k-ary Fat-Tree에 대한 이해를 도울 것이다.

III. 성능 평가 및 분석

1. 모의실험 설정

우리는 ns-2를 이용하여 링크 장애가 발생한 Fat-Tree에서의 TCP 성능을 측정하였다. 각 링크의 대역폭은 모두 1Gbps, 각 링크의 큐 크기는 250패킷, 링크별 전파지연시간은 50 μ s로 설정하였다. 모의실험 지속시간은 약 1초이며 ns-2에서 제공하는 기본적인 TCP-Reno를 사용하였다. 사용된 라우팅 프로토콜은 ns-2에서 기본적으로 제공하는 거리벡터 방식의 프로토콜로서 가용 다중경로를 모두 찾아놓는 기능을 가지고 있다. Fat-Tree토폴로지에서 ToR 스위치와 Core 스위치는 거리가 같은 최단 경로들에 걸쳐 라운드 로빈 방식으로 패킷을 분산할 수 있는 ECMP 기능을 사용하도록 설정하였다. Aggregation 스위치에서의 패킷분산방식은 완전패킷분산과 부분패킷분산 중 하나로 설정된다. 완전패킷분산의 경우 Core스위치나 ToR 스위치와 마찬가지로 기본 ECMP 방식에 의해 이루어지지만 부분패킷분산의 경우에는^[4]에서 제안된 방법대로 Core 스위치들을 겹치지 않는 그룹으로 나누고 출발지 호스트 주소에 따라 한 그룹을 선택하여 그 그룹에 속한 Core스위치들만을 사용하여 패킷을 분산한다.

먼저 4-ary Fat-Tree에서 두 쌍의 출발지-목적지에 대한 TCP 플로우들의 성능을 측정하였다. 출발지-목적지 호스트들은 그림 1이 보여주는 바와 같이 선택된다. 목적지 호스트들은 동일한 ToR 스위치에 연결되도록 하여 두 쌍의 목적지-출발지 간의 플로우들 간의 영향이

크도록 조건을 조성한다. 짧은 모의실험 시간 내에 적은 오차의 결과를 얻기 위하여 각 출발지-목적지 쌍에 대해 작은 범위 내에서 시작시각이 무작위로 설정된 10개의 TCP 플로우들을 생성하고 각 출발지-목적지 간의 플로우들에 대한 총 TCP 시간당처리량(throughput)을 측정하였다. 4-ary Fat-Tree에 대해서는 각 조건에 대해서 10번씩, 8-ary Fat-Tree에 대해서는 5번씩, 실험을 수행하였으며 링크 장애가 발생한 경우와 발생하지 않은 경우를 비교하였다.

세 쌍의 출발지-목적지에 대한 실험을 위해서는 그림 3이 보여주는 바와 같이 문제의 링크가 있는 pod 내의 호스트를 출발지로 하고 다른 정상적인 pod내에 있는 호스트를 목적지로 하는 TCP 플로우들을 추가하여 성능을 측정한다.

또한 우리는 8-ary Fat-Tree를 구성하여 Aggregation 스위치와 ToR스위치를 연결하는 한 개의 링크에 장애가 발생한 상태에서 두 쌍의 출발지-목적지에 대한 TCP 플로우들의 성능을 측정하였다. 8-ary Fat-Tree에는 최대 128개의 호스트가 연결되지만 4-ary Fat-Tree 모의실험에서와 비슷한 방법으로 출발지-목적지 쌍을 선택하였다. 즉 한 쌍의 출발지-목적지에 대해서는 문제의 링크에 연결된 ToR 스위치에 연결된 호스트를 출발지로 하고 다른 정상적인 pod에 있는 호스트를 목적지로 하여 플로우들을 생성한다. 그리고 두 번째 출발지-목적지 쌍에 대해서는 첫 번째 목적지 호스트와 같은 ToR스위치에 연결된 호스트들 중에 목적지 호스트를 선택하며 출발지 호스트는 다른 정상적인 pod에 있는 호스트들 중에 선택하였다.

2. 모의실험 결과 및 분석

우리는 링크 장애가 TCP 시간당처리량에 주는 영향을 조사하기 위해 혼잡원도우를 감소시킬 만큼 패킷 순서가 바뀌는 횟수에 주목한다. 혼잡원도우가 줄어드는 경우는 중복승인이 3번 이상 수신되거나 타임아웃이 발생하는 경우이다. 그러나 한 혼잡원도우내에서 여러 패킷이 손실되어도 혼잡원도우는 한 번만 감소할 수도 있기 때문에 패킷이 손실되는 횟수만큼 반드시 혼잡원도우가 감소하는 것은 아니다. (패킷 손실이 극심하여 타임아웃이 거듭 발생하고 혼잡원도우가 증가도 감소도 되지 않는 경우는 무시해도 된다고 가정한다.) 만일 혼잡원도우가 감소하는 횟수가 손실된 패킷의 수를 초과한다면

패킷순서가 바뀔므로 인해 혼잡원도우가 감소된 경우가 반드시 발생했다는 것을 의미한다. 이러한 경우의 발생 횟수는 아래와 같이 정의된 D 값으로 어느 정도 추정 가능하다.

$$D = [\text{네트워크에서 손실된 패킷의 수}] - [\text{혼잡원도우가 감소하는 횟수}]$$

패킷순서가 바뀌는 현상이 없으면 D 값은 음수가 될 수 없다. 그러나 패킷 손실에 의해서만이 아니라 패킷순서가 바뀔므로 인해 혼잡원도우가 감소하게 되면 D 값은 음수가 될 수 있다. 따라서 우리는 패킷손실이 아닌 이유로 인한 혼잡원도우 감소 횟수는 $D < 0$ 일 때 $|D|$ 보다 적지 않다고, 즉, 패킷 순서 바뀔므로 인해 혼잡원도우가 감소한 경우가 최소한 $|D|$ 만큼 발생한 것이라고 추정할 수 있다.

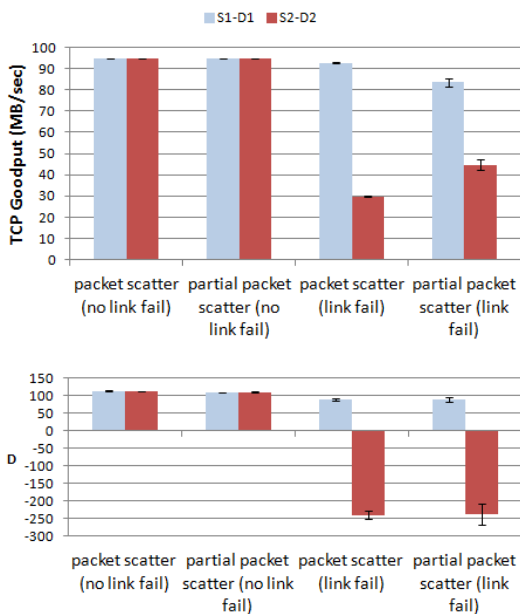


그림 2. 4-ary Fat-Tree에서의 두 쌍의 출발지-목적지에 대한 모의실험에서의 TCP 시간당처리량과 D 값
 Fig. 2. TCP throughput and D value for 2 source-destination pairs in 4-ary Fat-Tree

가. 4-ary Fat-Tree에서의 결과

그림 2는 두 쌍의 출발지-목적지에 대한 시간당처리량과 D 값을 보여준다. 이 결과로부터 관찰되는 사실들은 다음과 같이 요약될 수 있다.

링크 문제가 발생한 pod에서 출발한 플로우들에 비해

정상적인 pod간에 전송되는 플로우들의 시간당처리량이 더 많이 줄어든다. 정상적인 pod들에 있는 서버들인 S2-D2 간의 플로우들에 대해서는 D 값이 매우 작은 음수 값이다. 즉 손실된 패킷 수보다 혼잡원도우의 감소 횟수가 훨씬 많아 패킷 순서가 심하게 바뀌었음을 추측할 수 있다.

S2-D2간의 플로우들의 시간당처리량을 낮게 만드는 원인은 패킷의 순서 바뀔만이 아니다. 링크 장애가 없을 때에는 균일하게 분산되던 트래픽이 불균형하게 분산되면서 특정 링크에 집중되어 패킷손실이 더 많이 발생하는 것도 또 다른 원인이다.

완전패킷분산 방식에 비해 부분패킷분산 방식은 링크 장애가 발생할 때 두 출발지-목적지 쌍 모두에서 성능 감소를 보인다. D 값은 완전분산방식과 비슷하다. 따라서 두 방식 간의 차이는 트래픽의 편중으로 인한 패킷 손실의 정도에 기인한 것임을 알 수 있다.

그림 3은 세 쌍의 출발지-목적지에 대한 모의실험 설정을, 그림 4는 결과를 보여준다. 관찰되는 사실들은 다음과 같이 요약될 수 있다.

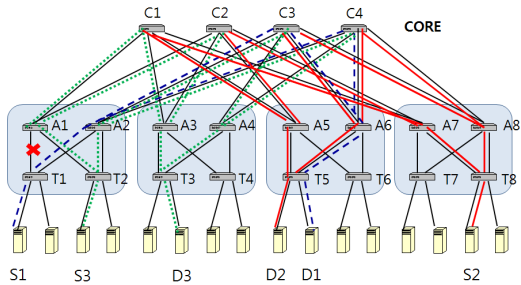
완전패킷분산 방식의 경우 S2-D2간의 플로우들의 시간당처리량은 대폭 감소하지만 S3-D3간의 시간당처리량은 별로 감소되지 않는다. S3-D3간의 플로우들의 D 값이 항상 양수 값이라는 점은 이 플로우들이 성능감소를 초래할 만큼 심한 패킷 순서 바뀔 현상을 겪지 않았다는 것을 의미한다. S3-D3간의 플로우들이 링크 A2-C3, A2-C4에서는 S1-D1간의 플로우들과 링크를 공유하지만 이로 인해 패킷들이 심하게 뒤바뀌지 않았음을 알 수 있다. 이는 가용다중경로가 많을 때에는 트래픽의 편중 정도가 낮아지고 패킷의 순서 바뀔의 정도도 혼잡원도우를 감소시킬 만큼 심하지 않게 됨을 암시한다.

부분패킷분산 방식의 경우 링크 장애가 발생했을 때 세 출발지-목적지 쌍 간에 성능감소가 분배되는 경향이 분명하다. 왜 모든 출발지-목적지 쌍에서 성능 감소가 있는지 살펴볼 때 우선 S1-D1간의 플로우들의 성능감소는 패킷 순서 바뀔 현상과는 전혀 관련이 없다. 이는 단일경로를 이용해야 하는 S1-D1 간의 플로우들이 다른 링크 A2-C3와 C3-A6에서 각각 S2-D2, S3-D3 간의 플로우들과 만나면서 패킷손실을 입기 때문에 발생하는 성능감소이다. S2-D2, S3-D3 간의 플로우들도 S1-D1 간의 플로우들을 만남으로 인해 패킷 손실과 패킷 순서 바뀔을 동시에 겪으면서 성능감소가 심하다. 이는 가용다중경로

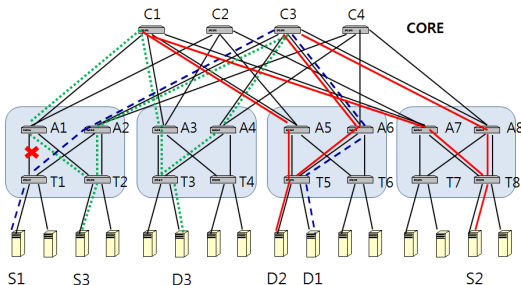
의 수가 적을 경우 링크 장애 발생 시 트래픽 편중의 정도가 더 심하고 TCP 성능에 더 많은 영향을 미침을 보여준다.

나. 8-ary Fat-Tree에서의 결과

그림 5는 4-ary Fat-Tree 모의실험에서와 같이 8-ary Fat-Tree에서도 Aggregation 스위치와 ToR 스위치를 연결하는 링크 하나를 단절시킨 상태에서 두 출발지-목적지 쌍에 대한 모의실험을 한 결과를 보여준다. 4-ary Fat-tree에서와는 달리 8-ary Fat-Tree에서는 최대 16개의 경로를 통해 패킷을 분사하는 완전패킷분산 방식과 단 4개의 경로로 분사하는 부분패킷분산 방식 모두 TCP 시간당처리량의 감소를 거의 겪지 않음을 알 수 있다. 손실된 패킷의 개수와 혼잡윈도우 감소 횟수의 차이도 링크 장애가 없을 경우와 거의 차이가 없다. 가용 다중경로의 수가 많을 때에는 링크 장애가 발생해도 트래픽의 편중 정도가 완화되고 성능감소로 이어지지 않는 것이다.



(a) 완전패킷분산(complete packet scatter)



(b) 부분패킷분산(partial packet scatter)

그림 3. 링크 장애가 발생한 경우 세 쌍의 출발지-목적지에 대한 모의실험 설정

Fig. 3. Simulation setup for 3 source-destination pairs in the presence of a link failure

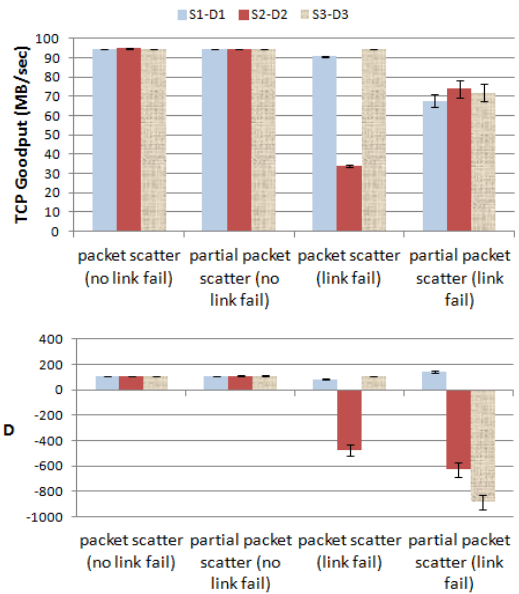


그림 4. 4-ary Fat-Tree에서의 세 쌍의 출발지-목적지에 대한 TCP 시간당처리량과 D값

Fig. 4. TCP throughput and D value for 3 source-destination pairs in 4-ary Fat-Tree

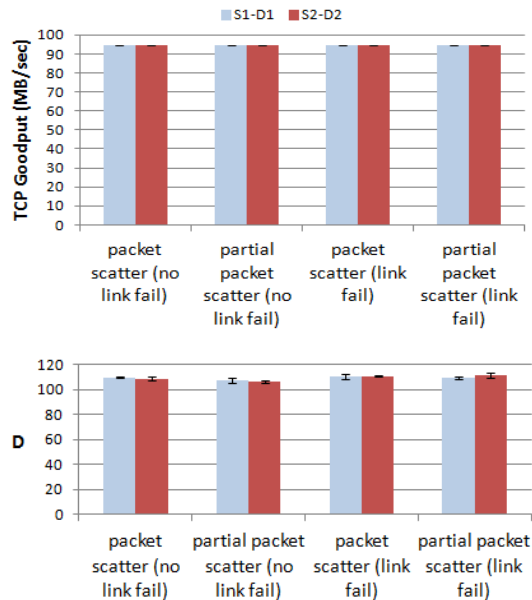


그림 5. 8-ary Fat-Tree에서의 두 쌍의 출발지-목적지에 대한 TCP 시간당처리량과 D값

Fig. 5. TCP throughput and D value for 2 source-destination pairs in 8-ary Fat-Tree

이러한 결과는 매우 중요한 의미를 갖는데 실제 데이터센터에서 사용될 큰 규모의 Fat-Tree 방식의 네트워크에서는 단일 링크 장애가 발생할 때 그로 인한 TCP의 성능 감소가 작을 것임을 암시하기 때문이다.

IV. 결론

우리는 Fat-Tree에서 단일 링크 장애가 발생할 경우 패킷분산방식이 TCP성능에 미치는 영향에 대해 조사하였다. 모의실험 결과가 보여주는 중요한 사실들 중 지금까지 발견되지 않았던 사실은 실제로 사용될 큰 규모의 Fat-Tree 토폴로지에서는 단일 링크 장애가 발생해도 패킷분산방식이 TCP 성능을 거의 저하시키지 않을 것이라는 점이다. 이러한 점에 착안하여 여러 종류의 장애를 견딜 수 있는 새로운 패킷분산방식에 관한 연구를 진행 중이다.

References

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

[2] Advait Dixit, Pawan Prakash, Y. Charlie Hu, and Ramana Rao Kompella, "On the Impact of Packet Spraying in Data Center Networks," IEEE Infocom 2013.

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

[4] Chansook Lim, "A New Packet-level Load-balancing Scheme for Fat-Trees," JIWIT Vol. 13, No. 2, 2013.

[5] 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.

[6] 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.

[7] 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.

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

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

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

[11] 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.

저자 소개

임 찬 숙(정회원)



- University of Southern California (박사)
- 홍익대학교 과학기술대학 컴퓨터정보통신공학과 조교수
- <주관심분야 : 라우팅, TCP, 네트워크 코딩, 인터넷 측정>

※ 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임. (No. NRF-2012R1A1A3013408)