

<https://doi.org/10.7236/JIIBC.2018.18.1.15>

JIIBC 2018-1-3

데이터센터 네트워크의 패킷단위 스케줄링에서의 DCTCP 성능

Performance of DCTCP with per-packet scheduling in data center networks

임찬숙*

Chansook Lim*

요약 데이터센터 네트워크의 자원을 효율적으로 활용하여 플로우들의 지연시간을 줄이기 위해서는 플로우 단위의 스케줄링보다는 패킷단위의 스케줄링이 적합하다. 따라서 최근에는 데이터센터 네트워크의 다중경로를 활용하여 패킷단위로 스케줄링을 수행하는 방식이 많이 제안되었다. 그러나 이로 인해 발생하는 순서 바뀐 패킷들이 표준 TCP 성능에 미치는 부정적인 영향을 줄이기 위해, 대부분의 방식들은 TCP 아래 계층에서 순서를 다시 맞추는 등의 보완방안을 필요로 한다. 본 논문에서는 데이터센터 네트워크의 대표적인 전송계층 프로토콜인 DCTCP가 다중경로상의 패킷단위 분산과 함께 사용될 때 어떤 성능을 보이는지 모의실험을 통해 조사한다. 모의실험 결과는 DCTCP가 큐의 길이를 짧게 유지할 수 있지만 혼잡신호에 따라 혼잡윈도우를 줄이는 방식 때문에 오히려 공정성 등의 측면에서는 기존 TCP보다도 낮은 성능을 가짐을 보여준다.

Abstract Per-packet scheduling is more suitable than per-flow scheduling to reduce the flow completion time by efficiently utilizing resources in data center networks. Recently, many per-packet scheduling schemes utilizing multiple paths have been proposed. However, to mitigate the negative effect of packet reordering on TCP performance, most of the schemes require supplemental measures such as putting packets in order at the lower layer. In this study, we investigate how well DCTCP, which is a representative TCP for data center networks, performs with per-packet scheduling through simulation. Simulation results show that DCTCP keeps the queue length short but that DCTCP shows low fairness due to the way of reducing the congestion window by ECN.

Key Words : data center network, per-packet scheduling, DCTCP, packet reordering

1. 서론

데이터센터 네트워크는 인터넷과 다른 특성을 가지고 있으므로 기존에 인터넷에서 사용하던 네트워크 프로토콜이 좋은 성능을 보이지 못하는 경우가 많다. TCP도 그

러한 대표적인 프로토콜이므로 데이터센터 네트워크의 특성에 맞춰 성능을 개선하기 위한 방안들이 많이 연구되어 왔다. 그 중 DCTCP(Data center TCP)^[1]는 가장 잘 알려져 있고 많이 사용되고 있는 대표적인 데이터센터용 TCP이다.

*정회원, 홍익대학교 컴퓨터정보통신공학과
접수일자: 2017년 12월 20일, 수정완료: 2018년 1월 20일
게재확정일자: 2018년 2월 9일

Received: 20 December, 2017 / Revised: 20 January, 2018

Accepted: 9 February, 2018

*Corresponding Author: chansooklim@hongik.ac.kr

Dept. of Computer & Information Communications Engineering,
Hongik University, Sejong, Korea

DCTCP는 기존의 TCP가 데이터센터 내부 네트워크에서 사용될 때 발생하는 문제들을 해결하기 위해 제안된 전송계층 프로토콜이다. 최근에는 데이터센터 네트워크에서 드물지 않게 혼잡이 발생하는 것으로 보고되고 있다. Google은 네트워크 활용률이 25%에 달할 때 혼잡으로 인한 패킷 손실이 발생한다고 보고하였다^[3]. DCTCP는 스위치의 큐의 길이가 짧게 유지되게 함으로써 패킷의 손실을 줄이고자 한다. DCTCP는 RED와 ECN 비트 표시를 사용하는데, ECN으로 혼잡의 신호를 받았을 때 혼잡원도우를 기존 혼잡원도우의 절반으로 줄이지 않고 혼잡의 정도, 즉 ECN 표시가 된 패킷들의 비율에 따라 혼잡원도우를 줄이는 정도를 조절한다는 점이 주요 차이점이다.

데이터센터 네트워크 스위치의 패킷 스케줄링도 많이 연구되어온 분야이다. 데이터센터 네트워크는 높은 대역폭, 낮은 지연시간, 다중경로가 많음, 토폴로지의 대칭성 등의 특징을 가지고 있지만, 기존에 인터넷에서 사용해오던 단일경로 라우팅과 패킷 전달방식으로는 이러한 특성의 데이터센터 네트워크를 충분히 활용할 수 없다. 따라서 인터넷에서는 실제로 활용되지 못하던 방식들도 심도 있게 연구되고 있는데 패킷단위의 분산방식도 그중 하나이다.

지금까지 다중 경로에 걸쳐 패킷단위로 분산하는 방식과 TCP는 양립하기 어려운 것으로 인식되어왔다. 패킷의 순서가 바뀌어 수신될 경우 발생하는 중복 ACK들을 TCP가 혼잡의 신호로 잘못 인식하여 불필요하게 전송률을 줄이기 때문이다. 그러나 데이터센터 네트워크의 대칭성과 낮은 지연시간 때문에 많은 연구자들은 패킷단위의 분산방식의 활용을 다시 고려하게 되었다. 예를 들어, Fat-Tree에서 패킷단위의 분산방식을 사용하면서 기존의 TCP를 사용할 때의 성능을 조사한 연구가 있다^[4]. 또한 Fastpass^[5]는 중앙제어장치를 통해 패킷단위의 스케줄링을 수행함으로써 스위치의 큐의 길이를 줄이고자 제안된 방법이다.

그러나 다중경로를 활용한 패킷단위의 분산방식을 DCTCP와 함께 사용할 때의 성능에 관한 연구는 거의 없으며 DCTCP에 관해 연구한 네트워크 환경은 주로 단일 경로를 통해 패킷을 전달하는 환경이다. 따라서 요즘 많이 연구되고 있는 패킷단위의 분산방식 환경에서 DCTCP가 어떻게 작동하는지에 관해서 조사할 필요가 있다. 구체적으로 우리의 질문을 요약하면 다음과 같다.

- 스위치에서 다중경로를 통해 패킷단위로 분산할 때 DCTCP는 기존의 TCP에 비해 어떤 차이를 보이는가? (모의실험 전에 우리는 DCTCP가 스위치 큐의 길이를 짧게 유지하려고 하므로 패킷순서 바뀔므로 인한 성능저하가 완화될 것이라고 예상하였다.)

- 만일 RED와 ECN표시만 사용할 때에는 어떠한 효과가 있는가?

- 대표적인 데이터센터 네트워크 토폴로지 중 하나인 Fat-Tree에서 링크 장애가 발생할 때 DCTCP는 기존의 TCP에 비해 어떤 성능을 보이는가?

우리는 ns-2를 이용한 모의실험을 통해 이 질문들에 대한 답을 찾아보고자 하였다. 우리는 4-ary Fat-Tree에서 가용한 모든 경로를 통해 패킷을 분산하는 방식을 사용하였고 링크에 장애가 발생하여 네트워크의 대칭성이 손상될 때 TCP버전들의 성능을 비교해보았다. 주요 실험 결과는 아래와 같이 요약할 수 있다.

- 링크 장애가 발생하여 대칭성이 깨질 경우에는 DCTCP가 표준 TCP에 비해 오히려 총 시간당 처리량이 낮아질 수 있다.

- 출발지-목적지로 구별되는 플로우 그룹간의 공평성을 평가해볼 때 링크 장애로 인해 대칭성이 깨질 경우, DCTCP가 표준 TCP에 비해 공평성이 더 좋지 않다.

모의실험 결과 중 우리가 특별히 주목하는 점은 장애로 인해 대칭성이 깨진 데이터센터 네트워크에서 패킷단위의 분산이 수행되어 패킷들의 순서가 유지되지 않을 때 DCTCP는 표준 TCP보다도 더 낮은 성능을 보인다는 점이다.

오늘날에는 데이터센터 네트워크의 방대한 자원을 효율적으로 사용하기 위해 패킷단위 분산의 실용화 방안이 그 어느 때보다도 활발하게 모색되고 있다. 지금까지는 TCP자체가 패킷 순서 바뀔을 처리하는 방식보다는 네트워크 프로토콜 스택의 하위 계층 어디에선가 패킷순서를 다시 맞추는 방식이 많이 연구되었다. 그러나 TCP자체에서 순서 바뀐 패킷들을 처리할 수 있다면 네트워크 자원을 보다 효율적으로 활용할 수 있을 것이다. 이러한 이유로 DCTCP의 효율성을 유지하면서 순서 바뀐 패킷들의 수신시 갖게 되는 취약점을 보완할 수 있는 새로운 TCP의 개발이 필요하다고 볼 수 있다.

II. 배경

1. DCTCP

인터넷에서 사용하는 표준TCP의 혼잡제어 방식은 데이터센터 네트워크의 성능 향상을 가로막는 장애로 일찍부터 지적되었다. 이를 해결하기 위해 많은 TCP변형들이 제안되었지만 DCTCP는 그중 가장 잘 알려진 TCP버전이다.

DCTCP는 TCP송신측의 전송률을 조절할 때 혼잡을 겪은 패킷의 비율에 기초하여 조절한다. DCTCP에서는 스위치들이 큐의 길이가 일정한 임계값을 넘을 때 패킷들을 ECN bit으로 표시하도록 되어있다. ECN bit으로 표시된 패킷의 비율에 비례하여 송신률을 조절함으로써 DCTCP는 큐의 길이를 낮게 유지하고 이로 인해 높은 시간당처리량(throughput)을 얻는 것이다.

DCTCP 알고리즘의 주요 요소는 다음과 같다.

- **스위치에서의 혼잡 표시.** DCTCP는 AQM(Active Queue Management)방식으로서 RED를 사용한다. 따라서 각 스위치에서는 큐의 길이가 임계값을 넘으면 들어오는 패킷들을 즉시 CE(Congestion Experienced) codepoint로 표시한다. 이렇게 하여 수신측과 송신측이 큐가 차오른 상태를 속히 알 수 있게 하는 것이다.
- **수신측에서의 ECN-echo.** DCTCP의 수신측은 수신된 패킷의 CE codepoint에 실려 온 정보를 송신측으로 그대로 보낸다. 이를 위해 DCTCP수신측은 모든 패킷에 대해 ACK을 보내되 CE codepoint로 표시되어 수신된 패킷들에 대해서는 ECN-Echo flag를 설정하여 보낸다..
- **송신측에서의 혼잡제어.** 송신측은 표시된 패킷들의 비율 추정치인 α 를 대략 매 RTT마다 다음과 같이 갱신한다.

$$\alpha \leftarrow (1 - g) \times \alpha + g \times F$$

여기서 F는 마지막 데이터윈도우에서 표시된 패킷들의 비율이다. $0 < g < 1$ 는 α 를 추정할 때 과거대비 신규 표본에 부여할 가중치를 의미한다. α 가 1에 가까워질수록 혼잡이 심함을 의미한다.

2. 데이터센터 네트워크의 패킷단위 분산방식

데이터센터 네트워크의 다중경로에 걸친 패킷단위의 분산방식이 활발히 연구되고 있다. 인터넷에서 거의 고

려되지 못했던 이 방식을 데이터센터 네트워크에서 사용하려는 이유는 높은 대역폭, 짧은 지연시간, 많은 다중경로 등의 특징을 갖는 데이터센터 네트워크의 방대한 자원을 효율적으로 이용하여 플로우들의 지연시간을 최대한 줄이기 위함이다. 특히 데이터센터 네트워크 토폴로지의 대칭성은 다중경로를 통한 패킷단위 분산방식을 사용해도 기존의 TCP가 패킷순서 바뀔 현상으로 인해 성능이 저하되는 정도가 덜 심할 수 있다는 기대를 갖게 했다. 실제로 [4]에서는 네트워크 장애가 발생하지 않은, 대칭적 네트워크에서 다중경로에 걸친 패킷단위로 분산해도 기존의 TCP가 성능저하를 겪지 않음을 보여주었다. 그러나 대칭성이 깨질 경우에 대비하여 새로운 AQM 방식을 제안하였다.

다중경로 상에서 패킷단위의 스케줄링을 하면서도 TCP의 성능을 유지하려는 최근의 연구들에는 중앙제어 방식, 수신측에서 패킷들의 순서를 맞추는 방안, flowlet을 이용하여 스케줄링하는 방식 등이 있다. Fastpass^[5]는 스위치 큐의 길이가 길어지지 않도록 패킷단위의 스케줄링을 중앙에서 수행하는 방식이다. JUGGLER^[7]는 프로토콜 스택 하단에서 패킷순서를 정리하여 올려보내는 방식이다. 또한 패킷 단위의 스케줄링을 하지는 않지만 패킷의 순서 바뀔을 막기 위해 플로우보다 더 작은 단위로 스케줄링하는 방식이 제안되었다. CONGA^[2]는 연이어 함께 전송되는 패킷들을 의미하는 “플로렛(flowlet)”단위로 스케줄링을 한다. 혼잡 피드백에 따라 스케줄링하는 CONGA는 스위치에 특수한 하드웨어를 필요로 한다. 이처럼 다양한 방식이 제안되었지만, 거의 대부분의 방식이 TCP 아래 계층에서 패킷 순서를 다시 맞추는 작업을 제안하거나 필요로 한다.

III. 모의실험

우리는 4-ary Fat-Tree상에서 3개의 출발지-목적지 그룹의 TCP 플로우들을 만들어 성능을 비교하기 위해 ns-2 모의실험을 수행하였다. DCTCP의 비교대상으로는 TCP-Reno를 사용하였다. 네트워크 토폴로지로는 그룹1이 보여주는 바와 같은 4-ary Fat-Tree를 사용하였으며 단일경로를 통한 패킷 전달방식과 다중경로를 통한 패킷 분산 방식을 비교하였다.

데이터센터 네트워크의 대표적인 토폴로지 중 하나인

Fat-Tree는 여러 개의 경로를 갖는 토폴로지이다^[8]. 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)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. 이렇게 구성되므로 한 호스트로부터 다른 호스트로 가는 서로 다른 경로가 여러 개 존재하게 된다.

출발지 호스트와 목적지 호스트는 그림1이 보여주는 바와 같이 선택되었으며 각 출발지-목적지 쌍에 대해 10개의 TCP 플로우를 생성하였다. 다중경로를 통한 패킷 분산방식에서는 각 스위치가 모든 가용경로를 통해 패킷을 분산한다. 그림1은 3쌍의 출발지-목적지 호스트 쌍에 대해 패킷들이 분산될 경로를 보여준다. 링크 장애 상황을 만들기 위해서는 그림1이 보여주듯이 A1-T1간의 링크를 단절하여 모의실험을 수행하였다.

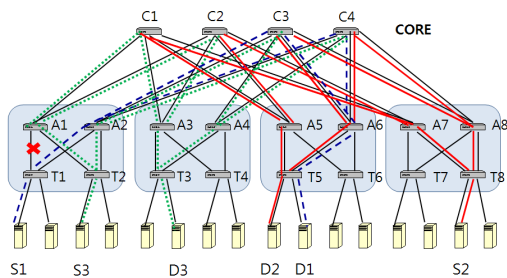


그림 1. 모의실험에 사용한 4-ary Fat-Tree
Fig. 1. 4-ary Fat-Tree used for the simulation

네트워크 파라미터 값은 아래와 같이 설정되었다. 최소 RTO값은 200ms와 20ms 두 가지로 설정하여 모의실험을 수행하였는데 그 이유는 데이터센터 네트워크 같은 고성능 네트워크에서는 TCP의 RTO가 성능에 지대한 영향을 미치지 때문이다.

표 1. 모의실험의 주요 파라미터
Table 1. Major parameters for simulations

파라미터	값
링크 대역폭	1Gbps
링크 지연시간	50 μ s
큐 크기	250
minimum RTO	200ms 또는 20ms
Receive Window	1256
RED 임계값 K	65

IV. 모의실험결과

단일경로 라우팅 환경에 대한 모의실험 결과를 그림 2가 보여준다. 그림2의 (a)는 링크장애가 없을 경우의 결과를, (b)는 링크 장애가 발생했을 경우의 결과를 보여준다. 링크장애가 발생하지 않은 경우보다 발생한 경우에 총 시간당처리량과 공평성이 더 좋다. 이러한 결과가 나온 이유는 링크 장애가 없는 경우에는 같은 링크를 선택 하던 플로우들이, 링크장애로 인해 네트워크 상황이 바뀔 때 다른 링크를 선택하게 됨에 따라 오히려 병목현상이 완화되었기 때문이다.

TCP-Reno + DropTail의 경우에는 최소 RTO의 영향을 많이 받는다. 그림2와 그림3은 최소 RTO가 짧을 때 더 좋은 결과를 얻음을 보여준다. 데이터센터 네트워크에서 최소 RTO를 줄임으로써 성능을 개선하는 방안은 이미^[6]에서 조사되었다. 반면에 DCTCP와 TCP-Reno + RED의 경우에는 스위치의 큐가 다 차기 전에 전송되는 ECN bit으로 인해 심한 혼잡을 피하게 되므로 재전송타임이 타임아웃되는 경우가 발생하지 않은 것이다. 결과적으로 DCTCP와 TCP-Reno + RED는 대체로 시간당처리량과 공평성에 있어 TCP-Reno + DropTail보다 좋은 성능을 보인다.

다중경로를 통해 패킷들을 분산할 때의 시간당처리량을 그림3이 보여준다. 이 경우 DCTCP는 TCP-Reno + RED보다 더 나은 결과를 보이지는 않는다. 링크에 장애가 발생하여 네트워크의 대칭성이 깨지면 DCTCP와 TCP-Reno 모두 시간당처리량이 저하된다. 특히 3개의 출발지-목적지 그룹들 중, S2에서 D2로 가는 플로우들의 시간당처리량이 많이 저하됨을 알 수 있다. 이는 S2-D2 패킷들이 대칭성이 깨진 다중경로를 따라 전송되므로 순서가 뒤바뀌어 도착하는 패킷이 생기기 되고 이로 인해 중복 ACK가 지속적으로 전송되기 때문이다.

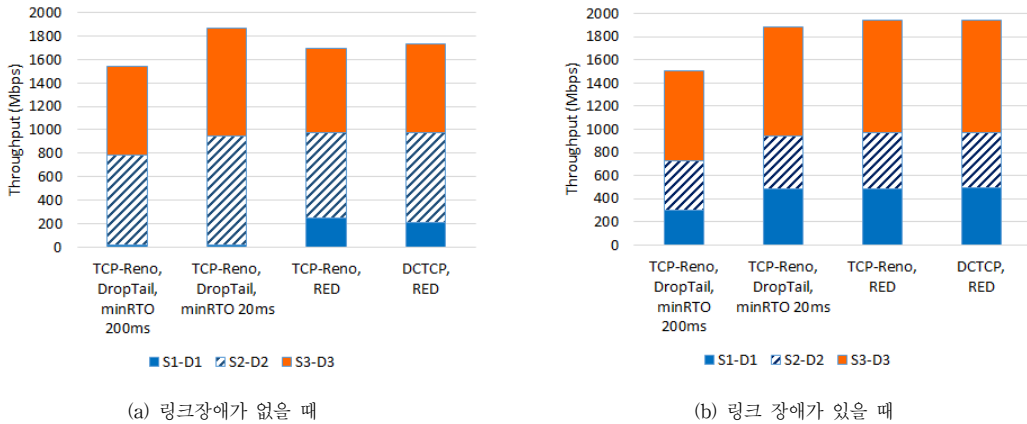


그림 2. 단일경로 환경에서의 시간당처리량
 Fig. 2. TCP Throughput in a single-path environment

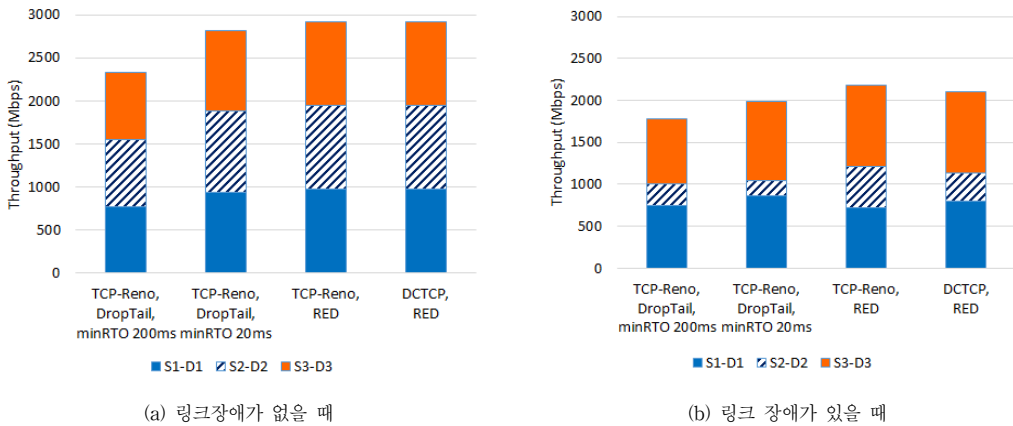
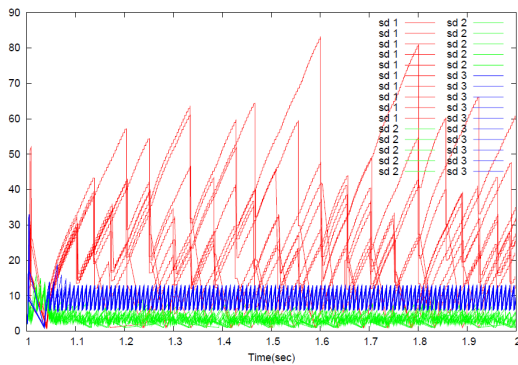


그림 3. 다중경로를 통한 패킷분산을 한 경우의 시간당처리량
 Fig. 3. Throughput of TCP with per-packet scheduling in a multipath environment

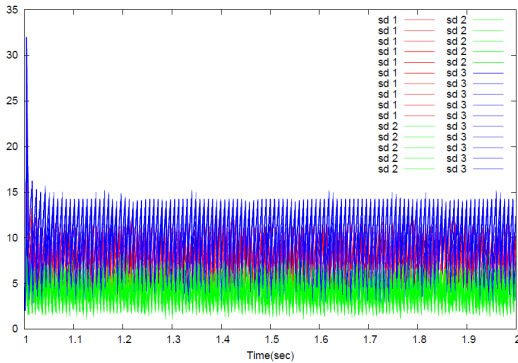
DCTCP는 ECN이 아닌 중복ACK을 받아 혼잡으로 판정하면 혼잡윈도우를 절반으로 줄이는데 이는 ECN에 의해 혼잡을 판정하여 혼잡윈도우를 줄일 때에 비해 훨씬 더 큰 폭으로 줄이는 셈이다.

TCP-Reno + RED는 DCTCP에 비해 총 시간당처리량과 공정성 면에서 더 나은 결과를 보이고 있다. 이는 TCP-Reno의 경우 ECN에 의한 혼잡 판정이든, 중복ACK에 의한 혼잡 판정이든, 혼잡윈도우를 절반으로 줄이게 되어있는 반면, DCTCP의 경우 중복ACK에 의한 혼잡 판정에 대해서만 혼잡윈도우를 절반으로 줄이게 되어있기 때문이다. 그러지 않아도 스위치의 큐 길이를 짧

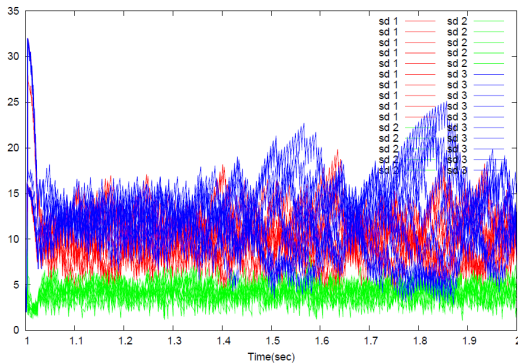
게 유지하도록 전송률을 줄이게 되어있는 DCTCP는 중복ACK들을 만나면 전송률을 더욱 줄이게 된다. 세 가지 다른 방식의 혼잡윈도우 추이를 보여주는 그림4는 DCTCP와 TCP-Reno가 중복ACK을 만나면 어떻게 작동하는지를 보여준다. 이 그림에서는 녹색 선이 패킷순서가 뒤바뀌어 수신되는 플로우들의 혼잡윈도우를 나타낸다. TCP-Reno + RED의 경우 ECN과 중복ACK 모두 혼잡윈도우를 반으로 줄이게 되어있어 세 그룹 플로우들 간의 차이가 작다. 반면 DCTCP의 경우 중복ACK를 수신하는 S2-D2 플로우들만 혼잡윈도우를 반으로 줄여야 하는 상황을 만나게 되므로 이 그룹의 플로우들의 혼잡



(a) TCP-Reno + DropTail (min RTO 20ms)



(b) TCP-Reno + RED



(c) DCTCP

그림 4. 3가지 방식 간의 혼잡윈도우 크기 비교
Fig. 4. Congestion window size of three schemes

윈도우가 특별히 작게 유지된다. DCTCP는 플로우 완료 시간(flow completion time)을 줄이기 위해 제안되었고 가장 알려진 데이터센터 네트워크용 TCP이다. 그러나 위 모의실험결과들이 보여주듯이 DCTCP도 최근에 연구되고 있는 패킷단위의 스케줄링 방식과 함께 사용하려면 패킷순서 바뀌는 현상이 발생할 경우에 대비한 개선책이

필요하다.

우리는 ECN 비트만을 혼잡의 신호로 사용하고 중복 ACK가 수신될 때 혼잡윈도우를 줄이지 않도록 수정하여 모의실험을 수행하였다. 그림 5는 그 결과를 보여준다. 같은 조건에서 원래의 DCTCP(그림 4(c) 참조)의 결과와 비교해 볼 때 총 시간당처리량에는 크게 변함이 없으나 S1-D1플로우들과 S2-D2 플로우들 간의 공평성은 좋아졌음을 알 수 있다.

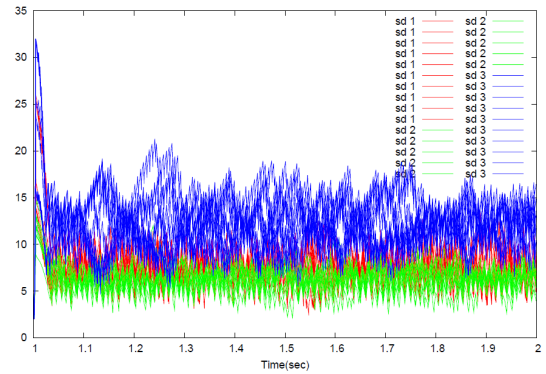


그림 5. 중복ACK에 반응하지 않도록 수정된 DCTCP의 혼잡윈도우 크기 변화

Fig. 5. Congestion window size of the modified DCTCP not reacting to duplicate ACKs

V. 결론

우리는 데이터센터 네트워크를 위한 대표적인 전송계층 프로토콜인 DCTCP가 다중경로상의 패킷단위 분산방식과 함께 사용될 때 어떤 성능을 보이는지 모의실험을 통해 조사해보았다. 모의실험결과는 TCP-Reno와 RED를 사용할 때에 비해 DCTCP의 성능이 더 좋지는 않음을 보여준다. 특히 네트워크의 대칭성이 성립되지 않아 패킷의 순서가 바뀔 때에는 DCTCP의 성능이 오히려 낮음을 볼 수 있다. 이러한 결과는 DCTCP가 패킷순서가 바뀌어 수신될 때의 문제를 스스로 해결할 수 있도록 개선되어야 할 필요를 보여준다.

References

[1] Mohammad Alizadeh, Albert Greenberg, David A.

- Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan, "Data Center TCP (DCTCP), ACM SIGCOMM 2010.
doi>10.1145/1851275.1851192
- [2] Mohammad Alizadeh, Tom Edsall, Sarang Dharmapurikar, Ramanan Vaidyanathan, Kevin Chu, Andy Fingerhut, Vinh The Lam, Francis Matus, Rong Pan, Navindra Yadav, George Varghese, "CONGA: distributed congestion-aware load balancing for datacenters," ACM SIGCOMM 2014.
doi>10.1145/2619239.2626316
- [3] Keqiann He, Eric Rozner, Kanak Agarwal, Yu (Jason) Gu, Wes Felter, John Carter, and Aditya Akella, "AC/DC TCP: Virtual Congestion Control Enforcement for Datacenter Networks," ACM SIGCOMM, 2016.
doi>10.1145/2934872.2934903
- [4] Advait Dixit, Pawan Prakash, Y. Charlie Hu, and Ramana Rao Kompella, "On the Impact of Packet Spraying in Data Center Networks," IEEE Infocom 2013.
DOI: 10.1109/INFCOM.2013.6567015
- [5] Jonathan Perry, Amy Ousterhout, Hari Balakrishnan, Devavrat Shah, Fans Fugal, "Fastpass: A Centralized "Zero-Queue" Datacenter Network," ACM SIGCOMM 2014.
doi>10.1145/2619239.2626309
- [6] Vijay Vasudevan, Amar Phanishayee, Hiral Sshah, Elie Krevat, David G. Andersen, Gregory R. Ganger, Garth A. Gibson, Brian Mueller, "Safe and Effective Fine-grained TCP Retransmissions for Datacenter Communication," ACM SIGCOMM 2009.
DOI: <http://doi.acm.org/10.1145/1594977.1592604>
- [7] Yilong Geng, Vimalkumar Jeyakumar, Abdul Kabbani, Mohammad ALizadeh, "JUGGLER: A Practical Reordering Resilient Network Stack for Datacenters," ACM Eurosys 2016.
doi>10.1145/2901318.2901334
- [8] Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat, "A scalable, commodity data center network architecture," ACM SIGCOMM 2008.
doi>10.1145/1402958.1402967
- [9] Chansook Lim, "Effects of Link Failures on Performance of Packet Scatter Schemes in Fat-Trees," The Journal of The Institute of Internet, Broadcasting and Communication, Vol.13, No. 4, pp. 9-15, August 2013.
DOI: <https://doi.org/10.7236/JIIBC.2005.5.2.56>.

저자 소개

임 찬 숙(정회원)



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