논문 2011-6-11

다중경로 환경의 네트워크 코딩에서의 TCP 성능개선 방안

TCP Performance Improvement in Network Coding over Multipath Environments

임찬숙*

Chansook Lim

요 약 네트워크 코딩위에서의 TCP 성능문제를 해결하기 위해 제안된 가장 잘 알려진 방식에서는 네트워크 코딩 계층이 혁신적인(innovative) 선형 결합을 수신하면 새로 디코드 된 패킷이 없다 하더라도 승인을 보낸다. 이 방식은 매우 효과적이지만 실제로 구현될 때에는 패킷 헤더 크기의 제한으로 인해 코딩 윈도우 크기를 제한해야 하므로 패킷 순서 바뀜 현상이 많이 발생할 때 성능이 저하될 수 있다. 본 연구에서는 네트워크 코딩 환경에서도 패킷 순서 바뀜 현상과 관련된 문제를 다루기 위해서는 중복승인을 사용하지 않고 타이머에 의존하는 TCP가 필요함을 주장한다. 또한 이러한 TCP를 위한 새로운 네트워크 코딩계층을 제안한다. 모의실험 결과는 두 개의 경로를 사용하는 라우팅 환경에서 패킷 순서가 바뀌어 수신되는 패턴에 따라 최대 19%까지 성능이 개선됨을 보여준다.

Abstract In one of the most impacting schemes proposed to address the TCP throughput problem over network coding, the network coding layer sends an acknowledgement if an innovative linear combination is received, even when a new packet is not decoded. Although this scheme is very effective, its implementation requires a limit on the coding window size. This limitation causes low TCP throughput in the presence of packet reordering. We argue that a TCP variant detecting a packet loss relying only on timers is effective in dealing with the packet reordering problem in network coding environments as well. Also we propose a new network coding layer to support such a TCP variant. Simulation results for a 2-path environment show that our proposed scheme improves TCP throughput by 19%.

Key Words : Network Coding, TCP, ARQ, Packet Reordering

Ⅰ. 서 론

네트워크 코딩방식의 실용화를 가로막는 중요한 문제 중 하나는 TCP와 함께 사용될 때 종종 좋은 성능을 얻지 못하는 문제이다. 이를 해결하기 위한 연구 결과들[1-7] 가운데 [1]에서 제안된 방식은 가장 영향력 있는 연구결과 중 하나로 꼽힌다. 기존에 나온 대부분의 방식에서는 수신측이 받은 패킷에 대한 승인을 보낼 때 네트워크 코딩 계층에서 디코드 한 패킷을 TCP계층이 받아 그에 대한 승인을 한다. 이에 반해 [1]에서 제안한 방법은 네트워크 계층이 TCP를 대신하여 승인을 하되 디코드 된 패킷에 대한 승인이 아닌 자유도(degree of freedom)에 대한 승인을 한다. 다시 말해서 디코드 하여 얻을 수 있는, 원래의 TCP 패킷에 대한 승인을 하는 것이 아니라 혁신적인(innovative) 선형결합이 도착할 때마다 승인을 하는 것이다. 이를 위해 실제로는 아직 보이지 않은 패킷 순서번호 중 가장 오래된 번호를 승인 패킷에 실어 보낸다. 이렇게 함으로써 블록 단위의 네트워크 코딩 방식과 TCP가 사용하는 슬라이딩 윈도우 방식과의 부조화를 해결하고자 한 것이다.

[1]의 저자들은 이 방식이 실제로 구현될 때에는 패킷

*정회원, 홍익대학교 컴퓨터정보통신공학과
접수일자 2011.10.12, 수정완료 2011.11.28
게재확정일자 2011.12.16

헤더 크기에 제한을 두는 것이 불가피하다고 설명한다. 따라서 이론상으로는 송신측 데이터 윈도우내의 모든 패킷을 인코딩할 수 있지만 헤더의 길이에 제한을 두어야 하므로 실제로는 송신 데이터 윈도우에 있는 패킷들 중 일정 수(W)만큼의 패킷만을 인코딩하여 보낼 수 있다는 것이다. 저자들은 실제로 W의 크기에 따라 TCP의 성능이 어떻게 영향을 받는지를 보여주었는데 패킷 손실률이 5%인 탠덤(tandem) 형태의 네트워크에서는 W=2일 때 가장 성능이 좋았다. 그보다 큰 W값을 사용하면 너무 많은 패킷 손실이 TCP에게 감춰져서 타임아웃이 발생한다고 분석하고 있다. 또한 W는 패킷손실률이 높을수록 커야 하며 RTT 등의 요소들에 의존한다고 설명하고 있다.

그런데 이러한 코딩 윈도우의 제한은 [1]에서 제안된 방식이 패킷 순서 바뀜 현상에도 많은 영향을 받도록 만든다. 네트워크 코딩 기술은 어느 정도의 순서 바뀜 현상을 잘 견디게 하지만 코딩 윈도우의 제한으로 인해 기존의 TCP가 순서 바뀜 현상에 부분적으로 노출되게 함으로써 결국 TCP 성능이 저하되는 결과가 초래된다. 패킷 손실과는 달리 패킷 순서 바뀜 현상은 TCP가 불필요하게 송신 윈도우를 줄이게 만든다.

본 연구에서는 이와 같은 기존 방식의 한계를 해결하기 위해 기존의 TCP 대신 순서 바뀜 현상에 강한 TCP의 사용이 효과적이고 필요함을 주장한다. (이러한 필요성이 [7]에서도 조사된 바 있으나, 이번 연구에서는 intra-flow 네트워크 코딩 환경에서 자유도 기반의 승인을 보낼 때의 TCP 성능에 대한 조사라는 점에서 차이가 있다.) 순서 바뀜 현상에 강한 TCP는 종종 중복 승인 방식 대신 타이머에 의존하는 방식을 사용한다. 이러한 TCP 버전들은 [1]에서 제안한 승인 방식을 그대로 사용할 수 없다. 가장 오래된 안 보인(oldest unseen) 패킷 번호 이상의 정보가 필요하기 때문이다. 우리는 새로 보인(newly seen) 패킷 번호를 가지고 승인하는 방식을 취하여 TCP-PR[9]과 함께 사용함으로써 패킷 순서 바뀜 현상이 지속적으로 발생할 때 기존의 방식보다 상당히 좋은 성능을 얻을 수 있음을 보여준다.

II. 자유도 기반의 승인 방식

네트워크 코딩은 일반적으로 블록 기반으로 이루어져 왔다. 블록의 크기는 고정되어 있고 각 블록은 서로 공유하는 패킷이 없으며 블록 단위의 선형결합을 전송하는 것이다. 그러나 이러한 방식은 몇 가지 단점을 갖고 있는데, 그 중 가장 심각한 문제는 슬라이딩 윈도우 방식의 프로토콜인 TCP와 함께 사용될 때에 TCP 윈도우의 진행에 문제가 생긴다는 점이다. 이러한 문제를 해결하기 위해 [1]에서는 블록 기반의 방식을 버리고 슬라이딩 윈도우 방식과 함께 자유도에 대한 승인 방식을 네트워크 코딩 계층에 적용하였다.

네트워크 코딩에서의 기존 승인 방식에서는 디코딩이 완료되어 원래의 패킷을 얻었을 때만 승인을 한다. 이와는 대조적으로 자유도에 대한 승인 방식에서는 어떤 패킷이 아직 디코드 되지 않았다 하더라도 송신된 순서 상 그 패킷 뒤의 패킷들이 디코드 될 수 있다면 이 패킷도 디코드 될 수 있을 때 승인을 한다. 기호와 정의를 사용하여 보다 정확하게 설명한다면 다음과 같다. p를 어떤 패킷이라고 하고 q는 송신자 측에서의 패킷 순서번호 상 p 다음의 패킷들로만 이루어진 선형결합이라고 가정한다. 만일 한 노드가 (p+q)형태의 선형결합을 계산하기에 충분한 정보를 갖고 있다면 그 노드는 패킷 p를 “보았다(see)”고 말한다. 디코딩이 이루어진 경우도 패킷을 ″본″ 것에 해당되는데 이 경우는 q=0이 되는 특별한 경우이다. 수신자들이 패킷 p를 보았다면 송신자 측에서는 p이후의 패킷들만 관리하면 된다. 이 패킷들을 디코드 하고 나면 수신자들은 q를 계산할 수 있으므로 p도 얻을 수 있다. 따라서 수신자들이 아직 p를 디코드 하지 않았다 하더라도 송신자 측에서 p를 버린다고 해서 잃게 될 정보는 없다. 따라서 기존의 ARQ는 패킷이 디코드 되었을 때 패킷을 승인하지만 이 방식에서는 패킷이 “보였을(seen) 때” 승인한다.

자유도에 대한 승인 방식은 여러 가지 유익을 제공하지만 특히 TCP와 함께 사용될 때의 가장 큰 장점은 TCP의 슬라이딩 윈도우가 앞으로 진행할 수 있도록 해준다는 점이다. TCP는 패킷 수신에 대한 승인을 위해 패킷의 순서번호를 사용한다. 그러나 패킷들이 함께 코딩될 때에는 승인을 위해 사용할 자유도의 순서가 분명하지 않다. 따라서 패킷의 순서번호와 일치하는 자유도의 순서를 정의하기 위해 앞서 언급한 “보이는” 패킷이라는 개념을 사용한 것이다. 수신 노드는 선형 결합으로 코딩된 패킷을 받자마자 새로 도착한 패킷 덕분에 새로이 보이는 패킷이 있는지 알아본다. 따라서 수신노드는 아직 디코드 할 수 없음에도 새로운 (원래의) 패킷을 이미 받은

체 하는 셈인데 이는 결국 모든 패킷이 보였을 때에는 디코딩도 될 수 있기 때문이다.

[1]에서는 슬라이딩 윈도우 방식의 온라인 코딩과 자유도에 대한 승인을 함께 사용하는 네트워크 코딩 계층을 프로토콜 스택의 전송계층과 네트워크 계층 사이에 끼워 넣었다. 이 네트워크 코딩 계층에서는 송신측이 패킷을 전송할 수 있을 때마다 혼잡 윈도우에 있는 모든 패킷들의 랜덤 선형 결합을 보낸다. 또한 수신자는 원래의 패킷에 대해서 승인하는 것이 아니라 자유도에 대해 승인한다. 송신측과 수신측 각각의 네트워크 코딩 계층의 동작 알고리즘을 요약하면 다음과 같다.

우선 송신측의 경우 TCP로부터 내려온 각 패킷에 대해 평균 R개의 선형 결합이 IP계층으로 내려 보내진다. 여기서 R은 중복 계수로서 채널에서의 손실률을 보상하고 TCP의 전송률과 데이터가 실제로 수신자에게 전달되는 수신률을 맞추기 위해 패킷을 중복해서 보내는 정도를 의미한다. 이상적으로는 R을 성공적인 수신 확률의 역수와 같도록 유지하는 것이 좋으므로 R값은 손실률을 추정하여 동적으로 조정한다. 표 1의 알고리즘은 송신측 네트워크 코딩 계층의 동작 방식을 보여주고 있는데 밑줄이 그어진 부분을 제외한 나머지 부분이 [1]에서 제안된 기존의 알고리즘이다. 밑줄 그어진 부분은 본 논문에서 제안하는 알고리즘에서 추가된 부분으로 IV절에서 설명하기로 한다.

수신 측 네트워크 코딩 계층의 동작 방식은 다음과 같다. 선형 결합을 받으면 수신 모듈은 우선 헤더에서 코딩 계수(coding coefficients)를 꺼내어 디코딩 행렬(basis matrix)에 추가한다. 그 다음, 가우스 소거법을 이용하여 어느 패킷이 새로 보이는가를 계산하고 혁신적인 선형결합이 수신되었으면 승인을 보낸다. 수신 모듈은 아직 디코드 되지 않은 패킷들의 선형 결합들을 보관하는 버퍼를 가지고 있다. 패킷들이 디코드 되면 TCP 수신측으로 그 패킷들을 전달한다. 표 2의 알고리즘은 수신측 네트워크 코딩 계층의 동작 방식을 보여준다. 기존의 알고리즘과 본 논문에서 제안하는 방식의 가장 큰 차이는 d)와 d′)에 있는데 기존의 알고리즘에서는 d)를 수행하는 반면, 새로 제안하는 알고리즘에서는 d′)을 수행한다. d′)에 대해서는 IV절에서 설명한다.

표 1. 송신측 네트워크 코딩 계층의 동작방식. 밑줄 그어진 부분을 제외한 부분들이 기존의 알고리즘이며 새로운 알고리즘에는 재전송 처리를 위한, 밑줄 그어진 부분이 추가되었다.

Table 1. Algorithm for the network coding layer at the sender. In the proposed algorithm, the underlined part for retransmission has been added to the existing algorithm.

송신측 네트워크 코딩 계층

1. Set NUM to 0.
2. Packet arrives from TCP sender:
   a) If packet is not already in the coding window, add it to the coding window.
   b) Set NUM = NUM + R (R=redundancy factor)
   c) Repeat the following $\lfloor NUM \rfloor$ times:
      i) Generate a random linear combination of the packets in the coding window.
      ii) Add the network coding header specifying the set of packets in the coding window and the coefficients used for the random linear combination.
      iii) Deliver the packet to the IP layer.
   d) Set NUM := fractional part of NUM.
3. ACK arrives from receiver: Remove the ACKed packet from the coding buffer and hand over the ACK to the TCP sender.
   a) If one of the following cases holds, retransmission is performed
      i) the ACKs with the same oldest decoded number are received more than 30 times.
      ii) the highest ACK number minus the oldest decoded number (notified by the receiver) is greater than 50.

III. 실제 구현 시의 한계점들

이론적으로는 송신 윈도우의 모든 패킷을 인코딩해서 보낼 수 있다. 그러나 인코딩 되는 패킷의 개수에 비례하여 헤더의 크기가 증가한다. 패킷 헤더가 무한히 커질 수는 없으므로 인코딩 할 패킷의 개수에 제한을 두어야 하는데 이를 최대 코딩 윈도우 크기(W)라고 한다. [1]에서는 네트워크 코딩이 실제 시스템으로 구현될 때의 이슈들도 기술하고 있다. 이들이 발견한 현상 중 하나는 단일 경로를 이용한 시뮬레이션 중에 시간당 처리량이 장시간 극심하게 낮은 경우가 발생한 것이었다. 저자들은 이유 중 하나로서 패킷이 “보인“ 순서가 뒤바뀜으로 인해 중

복승인이 발생했기 때문으로 추측하였다. 이러한 현상이 필드의 크기를 크게 하였을 때에는 감소하였다고 보고하고 있으나 필드 크기의 크고 작음이 어느 정도를 의미하는지에 대한 언급은 하지 않았다. 이러한 결과가 보여주는 바는 패킷이 "보인" 순서가 뒤바뀔 때 성능이 많이 감소할 수 있다는 사실이다.

표 2. 수신측 네트워크 코딩 계층의 동작방식. 밑줄이 그어진 d')은 새로운 알고리즘에서 수행되는 부분으로 기존 알고리즘의 d)를 대체한다.

Table 2. Algorithm for the network coding layer at the receiver. In the proposed alg orithm, the underlined d') substitutes for d) from the existing algorithm.

수신측 네트워크 코딩 계층

1. ACK arrives from TCP sink: If the ACK is a control packet for connection management, deliver it to the IP layer and return to the wait state; else, ignore the ACK.
2. Packet arrives from source side:
   a) Remove the network coding header and retrieve the coding vector.
   b) Add the coding vector as a new row to the existing coding coefficient matrix, and perform Gaussian elimination to update the set of seen packets.
   c) Add the payload to the decoding buffer. Perform the operations correspo nding to the Gaussian elimination, on the buffer contents. If any packet gets decoded in the process, deliver it to the TCP sink and remove it from the buffer.
   d) Generate a new TCP ACK with sequence number equal to that of the oldest unseen packet.
   d') Generate a new TCP ACK with sequence number equal to that of newly seen packet.

네트워크 코딩 계층의 주요 목표 중 하나는 패킷 손실과 순서 바뀜이 TCP에게까지 전달되지 않도록 가리는 것이다. 송신측 네트워크 코딩 계층에서 선형 결합하여 전송할 패킷들의 집합이 코딩 버퍼인데 승인되지 않은 패킷은 코딩 버퍼에 계속 남아 있다가 다른 패킷과 함께 코딩이 되어 전송되어야 한다. 앞서 언급한 바와 같이 코딩 윈도우의 크기(W)는 이론적으로는 제한이 없고 이 크기를 고정하지 않아도 된다. 패킷 손실을 가리기 위해서는 W가 클수록 유리하지만 실제 시스템에서 구현될 때에는 헤더의 크기가 무한정 클 수 없으므로 일정한 크기로 정해져야 한다. [1]에서는 이 W의 값을 시행착오를 통해 정하였는데 패킷 손실률이 5%인 조건에서 W=2일 때 가장 높은 성능을 보였다. W가 이보다 큰 값으로 설정될 때에는 오히려 시간당처리량이 급감하는데 그 이유는 TCP로 실제 네트워크 용량보다 더 큰 것으로 잘못 인식하게 하여 타임아웃이 발생하기 때문이라고 파악하였다. 그러나 W가 작으면 패킷 순서 바뀜 현상도 TCP에 전달될 가능성이 커진다.

Ⅳ. 새로운 알고리즘

[1]에서 제안한 자유도 기반의 승인 방식은 네트워크 코딩이 TCP와 잘 연동하게 만드는 획기적인 아이디어로 평가받는다. 그러나 패킷의 순서가 뒤바뀌는 현상이 지속되는 환경에서는 아직 보이지 않은 가장 오래된 패킷의 순서번호(oldest unseen sequence number)로 승인 하는 방식의 한계를 드러내게 된다. 즉 새로이 보이는 패킷의 순서번호가 순차적이 아닐 때에는 똑같은 번호로 승인을 보내게 되고 만일 이러한 중복된 번호를 실은 승인을 3개 이상 받게 되면 송신측 TCP는 패킷 손실이 발생했다고 판단하게 된다. 이는 기존 TCP의 중복승인에 의존하는 방식이 패킷 순서 바뀜에 취약하여 TCP 성능을 떨어뜨리는 것과 유사한 현상이다.

기존 TCP가 순서 바뀜 현상에 취약하다는 것은 오래된 문제이다. 이 문제를 해결하기 위해 제안된 TCP 버전들 중 한 부류는 기존 TCP의 중복승인을 사용하지 않고 타이머를 사용하여 패킷의 손실을 판단하는데 본 연구에서 사용된 TCP-PR[9]도 이러한 방식을 사용한다. TCP-PR의 경우 패킷의 순서 바뀜 현상이 지속적으로 발생할 때에는 성능 감소가 없고 패킷 순서 바뀜 현상이 없을 때에는 기존의 TCP와 비슷한 성능을 보인다.

중복승인을 사용하지 않고 타이머에 의존하여 패킷 손실을 판단하는 TCP 버전들은 [1]에서 제안한 대로 가장 오래된 안 보인(oldest unseen)패킷에 대해 승인하는 방식과 잘 맞지 않는다. 이를 해결하기 위해 본 논문에서 제안하는 새로운 알고리즘에서는 새로 보인(newly seen) 패킷의 순서번호를 가지고 승인을 한다. 표 2가 보여주는 수신측 네트워크 코딩 계층 알고리즘에서는 기존의 d)를 대체하는 d')이 이에 해당한다. 이렇게 함으로써, 코드화된 패킷의 순서가 심하게 바뀐 상태로 수신되어 보인

(seen) 패킷의 순서가 보낸 순서와는 크게 다를 때에도 송신측 네트워크 코딩 계층과 TCP는 어느 패킷이 무사히 도착했는지 파악할 수 있다.

타이머를 이용하여 패킷의 손실 여부를 판단하는 TCP는 패킷 손실이 발생할 경우 그 손실을 파악할 수 있을 때까지 기다리려면 그만큼 많은 버퍼가 필요하게 된다. 이러한 코딩 버퍼 문제를 완화하기 위해 네트워크 코딩 계층에서 패킷을 재전송한다. 표 1의 송신측 네트워크 코딩 계층 알고리즘에서는 ACK에 실려 온 가장 오래된 디코딩 일련번호(oldest decoded number)가 일정한 횟수 이상 똑같이 반복되거나 가장 최신의 ACK 번호와 가장 오래된 디코딩 일련번호의 차이가 일정 범위를 넘을 때 재전송을 하고 있다. 언제 재전송하는 것이 좋은가는 환경에 따라 달라질 것이므로 이에 관한 향후의 연구는 흥미로운 주제가 될 것으로 보인다.

V. 모의실험 결과

ns-2를 이용하여 수행된 모의실험에서는 기존의 자유도 승인 기반의 네트워크 코딩 계층위에 TCP-Reno를 사용할 때의 성능을, 새로 제안한 방식의 네트워크 코딩 계층 위에 TCP-PR을 사용할 때의 성능과 비교한다. 주요 성능 메트릭은 TCP의 시간당처리량(goodput)이다. 이 모의실험에서는 패킷순서 바뀜 현상을 지속적으로 발생시키기 위해 종단 간 송/수신 노드 간에 다중경로를 통해 패킷을 전송하였다. 라우팅 프로토콜로서는 AOMDV를 사용하였는데 원래의 AOMDV는 다중경로를 찾기는 하지만 주경로(primary path)가 끊어지지 않는 한 단일 경로로만 패킷을 전송하게 되어 있어 송신 노드에서 패킷을 분산시키기 위한 수정이 필요하였다. 실험을 단순화하기 위해 송신, 수신 노드 간에 지연시간이 차이가 나는 두 개의 경로만 생기도록 노드를 배치시켰다. 이를 위해 가장 가까운 노드 간의 거리는 150m 또는 210m가 되도록 하였다. 각 링크의 대역폭은 1Mbps로 설정하였으며 패킷 순서 바뀜 현상에 초점을 맞추기 위해 손실률은 0%로 설정하였다. 형성된 두 경로 중 한 경로는 4홉으로 구성되었고 다른 경로는 14홉으로 구성되었다.

W는 3으로 고정하였다. 사실 W는 네트워크 코딩 계층이 순서 바뀜 현상에 얼마나 잘 적응하도록 할 수 있는가에 영향을 미친다. W가 클수록 수신되는 패킷 순서가 바뀌어도 "보인" 패킷의 순서가 덜 바뀐다. [1]에서는 손실률이 5%이고 중복계수 R이 1.06일 때 W가 2인 경우 가장 좋은 성능을 얻었다. W가 작을수록 보인 패킷의 순서가 많이 바뀔 것이지만 W=2인 경우는 패킷 순서 바뀜에 취약한 극단적인 경우라고 판단하여 W=3에 대해 실험을 수행하였다. 또한 두 경로 간에 지연시간의 차이가 정해진 상황에서는 첫 번째 라우터에서 트래픽을 어떻게 분배하느냐가 코드화된 패킷의 도착순서, 그리고 패킷이 보이는 순서에 영향을 미친다.

우리는 지연시간이 긴 주경로와 지연시간이 짧은 두 번째 경로 사이의 트래픽 배분 방식을 여섯 가지(1:9, 6:14, 8:12, 12:8 14:6, 9:1)로 달리 하여 모의실험을 수행하였다. 여기서 x:y라 함은 첫 번째 경로로 x개의 패킷을 보낸 후 두 번째 경로로 y개의 패킷을 보냄을 의미한다. (패킷 순서 바뀜 현상이 언제 극단적으로 많이 발생하는가에 대해서는 더 연구가 필요할 것으로 보인다.)

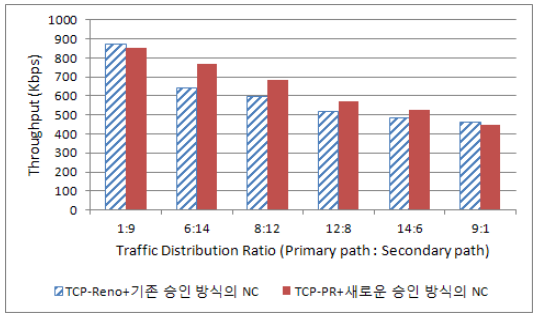

그림 1. 기존 자유도 승인 기반 네트워크 코딩과 TCP-Reno를 사용했을 때와 수정된 네트워크 코딩과 TCP-PR을 사용했을 때의 시간당 처리량

Fig. 1. TCP goodput comparison between TCP-Reno over the existing network coding scheme based on degree-of-freedom ACK and TCP-PR over our modified network coding scheme

그림 1은 모의실험 결과를 보여준다. 두 TCP 모두의 공통점은 트래픽 배분 비율이 긴 경로로 더 많은 패킷을 보내도록 설정될 때 시간당 처리량이 낮아진다는 것이다. 그러나 트래픽 배분율에 따라 두 TCP간에 시간당처리량의 차이가 달라짐을 관찰할 수 있다. 이 배분율은 코딩 윈도우 크기 W와 결합하여 성능에 영향을 미친다. 배분율이 1:9이거나 9:1일 때에는 두 TCP 간에 성능 차이가 많지 않을 뿐 아니라 TCP-Reno의 성능이 TCP-PR보다 조금 나은 것을 알 수 있다. 이는 10개 중 한 개의 패킷이

늦게 도착하거나 빨리 도착하게 될 때에는 W가 3이어도 패킷의 순서 바뀜을 충분히 가릴 수 있어 네트워크 코딩 계층에서 가장 오래된 안 보인 패킷에 대한 중복 승인을 3개 이상 발생시키지 않기 때문이다. 그러나 6:14와 같이 여러 개의 패킷이 한꺼번에 뒤늦게 도착하게 되는 경우 보인 패킷의 순서는 많이 바뀌게 되고 가장 오래된 안 보인 패킷의 번호가 동일한 승인들, 즉 중복 승인들이 많이 발생하게 된다. 이로 인해 TCP-Reno의 경우 성능이 낮아진다. 이에 반해 TCP-PR을 사용하는 버전에서는 TCP-Reno에 비해 19%정도 좋은 성능을 보이는 것을 볼 수 있다.

Ⅵ. 결 론

우리는 패킷 순서가 바뀌는 현상이 심한 환경에서 네트워크 코딩이 사용될 때 TCP의 성능이 개선될 수 있도록 하는 방안에 대해 연구하였다. 전송계층에서는 중복 승인을 사용하지 않고 타이머에 의해 패킷 손실을 판단하는 TCP-PR을 사용하고 네트워크 계층에서는 가장 오래된 안 보인 패킷으로 승인하는 대신 새로이 보인 패킷을 승인하는 방식을 사용하였다. 모의실험 결과는 지연시간이 다른 두 경로 간에 트래픽 배분 방식을 달리 할 때 최대 19%까지 TCP성능이 개선됨을 보여주었다.

패킷 순서 바뀜이 심한 환경에서 패킷손실률과 코딩 윈도우 크기, 중복계수 등의 영향을 함께 조사하는 작업을 현재 진행 중이다.

참 고 문 헌

[1] Jay Kumar Sundararajan, Devavrat Shah, Muriel M´edard, Michael Mitzenmache, Joao Barros, "Network coding meets TCP: Theory and Implementation," Proceedings of the IEEE 2011.

[2] MinJi Kim, Muriel Médard, João Barros, "Modeling Network Coded TCP Throughput: A Simple Model and its Validation", CoRR abs/1008.0420: (2010).

[3] Yong Huang, Majid Ghaderi, Don Towsley, and Weibo Gong, "TCP Performance in Coded Wireless Mesh Networks", Proc of IEEE SECON, pp. 179-187, 2008.

[4] Hulya Seferoglu, Athina Markopoulou "Network Coding-Aware Queue Management for Unicast Flows over Coded Wireless Networks", IEEE NetCod 2010.

[5] Sofiane Hassayoun, Patrick Maille, and David Ros, "On the impact of random losses on TCP performance in coded wireless mesh networks", IEEE Infocom 2010.

[6] S. Hassayoun, P. Maille, and D. Ross, "On the impact of random losses on TCP performance in coded wireless mesh networks", IEEE Infocom 2010.

[7] 임찬숙, "중복승인을 사용하지 않는 TCP의 코드화된 무선 메쉬 망에서의 효과", 한국인터넷방송통신학회 2010년 2월 논문지 제 11권 1호.

[8] Szymon Chachulski, Michael Jennings, Sachin Katti, Dina Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing", ACM SIGCOMM 2007.

[9] S. Bohacek, J. P. Hespanha, J. Lee, C. Lim, K. Obraczka, "A new TCP: TCP for Persistent Packet Reordering", ACM/IEEE Transactions on Networking, pp. 369-382, 2006.

저자 소개

임 찬 숙(정회원)

- 서울대학교 계산통계학과 학사, New York University 석사, University of Southern California 박사
- 홍익대학교 과학기술대학 컴퓨터정보통신공학과 조교수

<주관심분야 : 라우팅, TCP, 네트워크 코딩, 인터넷 측정>

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 연구임 (No. 2011-0003541).