SCTP의 멀티호밍 특성 및 재전송 정책에 대한 성능평가

송정화[O*] 이미정[*] 고석주[†]
[*]이화여자대학교 컴퓨터학과, [†] 한국전자통신연구원 표준연구센터 선임연구원
(jhsong[O], lmj)@ewha.ac.kr[*], sjkoh@etri.re.kr[†]

Performance Evaluation of SCTP multi-homing and retransmission policy

Jeong Hwa Song[O*], Mee Jeong Lee[*], Seok Joo Koh[†]
[*]Dept. of Computer Science and Engineering Ewha Womans
[†] Protocol Engineering Center, ETRI

요 약

SCTP는 TCP와 마찬가지로 연결 지향적이며 신뢰성 있는 데이터의 전송을 위한 전송 계층 프로토콜이다. SCTP는 오류 및 플로우 제어 등 많은 부분에 있어서 TCP의 방식을 그대로 따르며, 거기에 더하여 멀티스트리밍과 멀티호밍 특성을 가진다. 이 논문에서는 TCP와 다른 대표적인 특징들 중 멀티호밍이 성능에 미치는 영향을 살펴 보았다. 먼저 프로토콜간의 성능 비교를 통해서는 SCTP 프로토콜이 TCP와 다른 혼잡 제어 메커니즘을 사용함으로 인해 TCP Reno나 TCP SACK에 비해 향상된 성능을 보임을 확인할 수 있었다. 또한, 멀티호밍을 지원하는 경우 현재의 SCTP의 재전송 정책이 프라이머리 경로와 대체 경로간의 경로 특성의 따라 성능 저하를 가져올 수 있는 잠재적인 문제점을 가짐을 확인하였다.

1. 서 론

SCTP는 IP 네트워크에서의 텔레포니 시그널링 메시지를 전송하기 위해 IETF SIGTRAN 워킹 그룹에서 제안되었으며, 범용 전송 프로토콜로 2000년 10월에 SCTP 표준 문서 RFC 2960으로 제정되었다. SCTP는 연결 지향적인 프로토콜로써 통신을 위해서 두 엔드 포인트는 커넥션을 설립해야 하며, SCTP에서는 이 커넥션을 어소시에이션 (association)이라 부른다. SCTP는 TCP와 마찬가지로 신뢰성 있는 프로토콜이며, 오류 및 플로우 제어는 대부분 TCP의 방식을 그대로 따른다.

TCP와 다른 SCTP의 대표적인 특징으로는 멀티스트리밍과 멀티호밍이 있다. 멀티스트리밍은 데이터를 여러 개의 다른 스트림으로 나누는 것이 가능하도록 하며, 각 스트림은 그 특성에 따라 배달될 수 있고, 다른 스트림들과 독자적으로 다루어 질 수 있도록 한다. 또, 하나의 호스트가 여러 개의 IP 주소를 가지는 것을 허용하는 멀티호밍 지원을 통해 네트워크 결함에 좀 더 탄력적으로 대처할 수 있도록 하며, IP계층에서 보다 더 높은 수준의 신뢰성을 제공할 수 있도록 한다.

이 논문에서는 이 중 멀티호밍이 성능에 미치는 영향을 살펴보고자 한다. 본 논문의 구성은 다음과 같다. 1장의 서론에 이어서 2장에서는 SCTP의 새로운 특성들 중에서 우리가 다루고자 하는 특성인 멀티호밍에 대해서 소개하고, 3장에서는 비교하는 프로토콜들의 성능에 영향을 미치는 TCP의 혼잡 제어 메커니즘과 SCTP의 혼잡 제어 메커니즘을 설명한다. 4장에서는 시뮬레이션을 통해서 TCP Reno와 TCP Sack, 멀티홈드 호스트(multi-homed host)에서의 SCTP와 싱글홈드 호스트(single-homed host)에서의 SCTP의 성능을 비교, 분석한다. 마지막으로 5장에서는 결론과 함께 앞으로의 연구 방향을 기술한다.

2. SCTP 멀티호밍

앞 절에서 언급한 것처럼 SCTP와 TCP의 큰 차이점은 SCTP가 여러 개의 IP 주소들로 어소시에이션을 구성할 수 있다는 점이다. TCP의 경우는 두 엔드 포인트 간의 멀티호밍을 지원하지 않는다. 따라서 엔드 포인트의 IP 주소가 더 이상 액세스 가능하지 않게 되면, TCP 커넥션은 끊어지게 되고 이는 응용에 의해 복구되어야 한다. 이러한 복구를 위한 오버헤드와 지연은 응용에 따라서는 중요한 문제이다. 그러나, SCTP는 이러한 문제를 염두해 두고 물리적으로 다중의 인터페이스가 존재하는 경우 모든 인터페이스로의 경로에 문제가 생기지 않는 한 어소시에이션이 유지될 수 있도록 디자인되었다. 만약 다중의 인터페이스에 매핑된 IP 주소가 다른 경로를 통해 액세스되도록 라우팅 되었다면, 이러한 특성은 네트워크 계층의 중복성을 제공하게 된다. 이러한 네트워크 계층에서의 중복성은 IP 주소들 중 하나가 도달할 수 없게 되어도 대체 IP 주소들 중 하나로 패킷을 재라우팅함으로써 그 호스트를 액세스하는 것이 가능하도록 한다.

SCTP는 여러 개의 주소 중에서 프라이머리 주소라 불리는 하나의 주소를 정하고 이 주소를 통해서 상대편과 통신을 한다. 그 밖의 주소에 대해서는 세컨더리 주소(secondary) 혹은 대체 주소라 부르며, 프라이머리 주소로 보낸 데이터가 손실되는 경우에 재전송을 위한 경로로써 사용한다[1].

이러한 SCTP의 현재 재전송 정책은 모든 재전송을 대체 목적지 주소로 보냄으로써 재전송 패킷이 수신측에 제대로 도착할 수 있는 기회를 늘리자는 데에 목적을 둔다. 여기에는 기본적으로 패킷 손실의 원인이 해당 목적지 주소가 더 이상 도달 가능하지 않거나 해당 경로가 혼잡이라는 가정을 가진다. 그러나 경우에 따라서는 현재의 SCTP의 재전송 정책은 성능을 감소시킬 수도 있다. 이 논문에서는 시뮬레이션을 통해 이러한 현재 SCTP의 재전송 정책의 잠재적인 문제점을 보여주고자 한다.

3. 혼잡 제어 메커니즘

멀티스트리밍이나 멀티호밍 외에도 두 프로토콜간의 혼잡 제

어 메커니즘의 차이로 인해 두 프로토콜의 성능은 달라지게 된다. SCTP는 기본적으로 TCP의 혼잡 제어 메커니즘을 기반으로 하고 있으나, 부분적인 차이로 인해 두 프로토콜은 성능은 다르게 된다. 이를 살펴보기 위해 두 프로토콜의 혼잡 제어 메커니즘에 대해 간략히 설명하고자 한다.

3.1 TCP의 혼잡 제어 메커니즘

TCP에서의 혼잡 제어는 TCP가 패킷 손실을 네트워크의 혼잡이라고 판단함으로써 이루어진다. TCP는 타임아웃이나 수신자로부터의 중복된 ACK을 받게 됨으로 인해 패킷 손실을 감지하고, 이러한 패킷 손실을 네트워크의 혼잡이라고 생각하고 전송율을 줄이게 된다. 패킷 손실로 인해 타임 아웃이 발생하는 경우에는 슬로우 스타트(Slow start)가 시작되고, 중복된 ACK 수신으로 패킷 손실을 발견하는 경우에는 빠른 재전송(fast retransmit)을 하게 된다[2].

슬로우 스타트 과정은 cwnd가 슬로우 스타트 임계치에 도달하여 혼잡 회피(congestion avoidance) 과정에 들어가기 전까지를 말하며 이 과정에서는 매 RTT마다 cwnd를 두 배로 증가시킨다. 즉, 송신자의 패킷 전송율을 RTT마다 두배로 증가시키게 된다. cwnd가 슬로우 스타트 임계치에 도달하게 되면, 패킷 손실이 감지되거나 수신자의 버퍼가 모두 차게 될 때까지 cwnd를 RTT마다 1 MTU (Maximum Transmission Unit)씩 증가시킨다. 만약 송신자가 세 개의 연속적인 중복 ACK을 수신하게 되면, TCP는 이를 패킷 손실로 판단하고 타임 아웃이 되기 전에 미리 손실된 패킷을 재전송하는데, 이를 빠른 재전송이라고 한다.

3.2 SCTP의 혼잡 제어 메커니즘

SCTP는 TCP처럼 슬로우 스타트, 혼잡 회피, 빠른 재전송으로 이루어지는 윈도우 기반 혼잡 제어 메커니즘을 사용하여 신뢰성 있는 전송을 보장하고 손실된 패킷, 순서에 어긋나게 도착한 패킷, 중복된 패킷을 감지한다. 기본적인 혼잡 제어 메커니즘은 TCP와 유사하며, 다음과 같은 차이점을 가진다.

- TCP에서 SACK은 선택 가능한 옵션이지만, SCTP에서는 필수적인 사항이다. 일반적으로 수신자는 패킷을 받으면 그에 대한 ACK을 송신자에게 보내는데, 이 ACK은 누적된 ACK (cumulative ACK)으로써 손실된 가장 낮은 번호의 패킷에 관한 정보만을 제공한다. 따라서 RTT당 단지 하나의 손실된 패킷만을 복구할 수 있다. SACK은 이러한 문제를 해결하기 위해서 RTT내에서 잃어버린 모든 패킷을 알려 준다.
- TCP의 경우는 cwnd의 초기값이 TCP의 경우는 일반적으로 1 MTU이나, SCTP에서는 2*MTU이다. 이는 동일한 네트워크 상황에서 SCTP가 TCP에 비해 좀 더 빠른 시간 내에 많은 데이터를 전송할 수 있도록 한다.
- TCP는 cwnd가 슬로우 스타트 임계치보다 적은 경우에만 슬로우 스타트를 하고 cwnd가 슬로우 스타트와 같은 값이 되면 혼잡회피 과정으로 들어가는데 반해, SCTP에서는 cwnd가 슬로우 스타트 임계치와 같을 때까지 슬로우 스타트를 수행한다. 이는 작은 차이로 보이나 [3]에서의 성능 실험 결과 의미 있는 수준의 성능 차이를 보임을 알 수 있다.
- TCP와는 달리 SCTP는 명시적인 빠른 복구 알고리즘을 사용하지 않는다. TCP나 SCTP의 혼잡 제어 알고리즘은 패킷을 전송한 후 ack 받지 못한 패킷들의 수보다 cwnd가 큰 경우에만 새로운 데이터의 전송이 가능하다[1, 2]. SCTP는 중복 ACK을 받는 동안에도 SACK 사용으로 인해 새롭게 ack된 패킷을 파악할 수 있기 때문에 전송하고 ack받지 못한 패킷들의 크기를 감소시킬 수 있으므로 cwnd의 증가 없이도 새로운 데이터의 전송이 가능하게 된다. 따라서 SCTP는 명시적인 빠른 복구 알고리즘을 사용하지 않으면서, TCP의 빠른 복구 알고리즘과 같은 역할을 한다.

4. 성능 평가

먼저 일반적인 데이터 전송시에 SCTP와 TCP의 성능 차이를 알아보기 위해 SCTP와 TCP 에 대해 실험을 수행하였다. TCP는 현재 신뢰성 있는 전송 프로토콜로써 일반적으로 가장 많이 사용되고 있는 프로토콜인 TCP Reno와 함께 TCP SACK을 실험하였다. SCTP가 기본적으로 SACK을 사용하므로 TCP SACK과의 비교를 통해 SACK 사용으로 인한 두 프로토콜 간의 성능 차이를 배제하고, 멀티호밍과 함께 SACK이외의 SCTP 혼잡제어 방식으로 인해 발생하는 SCTP와 TCP의 성능 차이를 볼 수 있다. 또한, 멀티호밍을 지원하는 경우 프라이머리 경로와 대체 경로들간의 경로 특성에 따른 SCTP 성능을 살펴보기 위한 실험을 수행하였다. 시뮬레이션은 ns-2 네트워크 시뮬레이터를 이용하였으며, 리눅스 레드햇 7.3에서 수행되었다.

4.1 시뮬레이션 모델

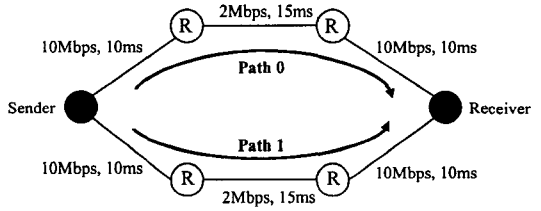

그림 1 시뮬레이션 네트워크 토폴로지

그림 1은 시뮬레이션을 위해 사용된 토폴로지로써 두 에지 노드(송신원과 수신원)는 동일한 대역폭 및 지연을 가지는 두 개의 경로 path0과 path1을 통해 연결되어 있다. 각 에지 노드에는 TCP 또는 SCTP 에이전트가 올려지며, TCP 송신측은 path0을 통해 TCP 수신측에 데이터를 전송한다고 가정하였고, SCTP 송신측은 멀티호밍 호스트를 실험하는 경우 path0을 프라이머리 경로로 사용하고 path1을 대체 경로로써 재전송을 위해 사용한다고 가정하였다. 싱글호밍 호스트인 경우의 SCTP 성능 측정을 위해서는 TCP와 마찬가지로 path0을 이용하여 데이터를 전송한다고 가정하였다.

4.2 시뮬레이션 결과 및 분석

4.1 절의 기본적인 가정하에 두 경로의 패킷 손실율 및 대체 경로인 path1의 대역폭과 지연을 다양하게 변화시켜 보면서 4MB 파일을 전송하는 데에 소요되는 시간을 측정하였다. 파일 전송 시간은 송신측 SCTP에서 4MB 파일을 전송하기 시작하는 시간으로부터 이 파일이 수신측에 모두 도착하기까지의 시간을 나타낸다. 무작위성을 가정하는 실험에 대해서는 모두 100번의 시뮬레이션을 수행하여 그 평균 값을 취한 결과를 보였다.

4.2.1 멀티호밍 특성에 따른 성능 평가

그림 2는 path0의 패킷 손실율을 0 ~ 16%까지 변화시키며 각 프로토콜의 파일 전송 시간을 측정한 결과를 보인 것이다. 이 때 기존의 SCTP 성능 연구들에서 가정한 바와 같이 재전송 경로인 path1은 패킷 손실이 전혀 없는 경로라고 가정하였다. 패킷 손실율은 어소시에이션이 유지되는 전체 시간 동안의 패킷 손실에 대

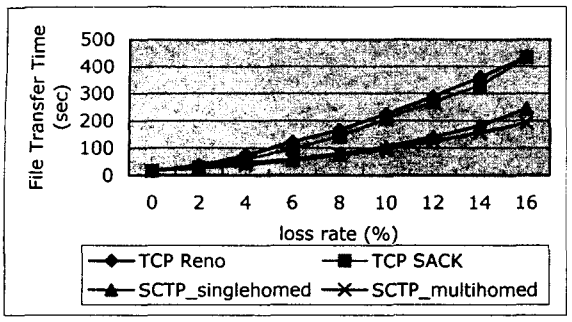

그림 2 4MB file 전송시 패킷 손실율에 따른 지연

한 것이며, 패킷 손실은 균일 에러 모델(uniform error model)에 따라 발생하도록 하였다. 패킷의 손실율이 높아질수록 멀티호밍을 사용하는지의 여부에 관계 없이 SCTP 스킴들이 TCP 스킴들에 비해 월등하게 좋은 성능을 보였다. TCP와 마찬가지로 path0만을 사용하는 SCTP도 TCP보다 좋은 성능을 보이므로 이것은 멀티호밍에 의한 성능 차이가 아닌 3장에서 설명한 SCTP와 TCP의 혼잡 제어 메커니즘의 차이에 따른 것이다. 또, 동일하게 SACK 메카니즘을 사용하는 TCP SACK과 싱글호밍 SCTP 간의 차이는 멀티호밍이나 SACK 사용 여부에 따른 성능 차이가 아닌 그 밖의 혼잡 제어 방식의 차이에 의한 것으로 볼 수 있다. 손실율이 4%를 넘어서게 되면 멀티호밍을 지원하는 경우의 SCTP가 path0만을 사용하는 SCTP보다 더 나은 성능을 보이는데, 이는 프라이머리 경로의 손실이 많은 경우에 재전송 경로를 사용하는 빈도수가 높아지게 되어 두 개의 경로를 사용하게 되는 시간이 길어짐에 따른 것이며, 또 이 실험에서 재전송 경로는 손실이 전혀 없는 경로이므로 빠른 손실 복구가 가능하기 때문이다. 이와 같은 실험 결과를 통해 종합적으로 볼 때, 패킷 손실에 대응하는 SCTP의 혼잡 제어 방식이 TCP보다 더 효율적이며, 완벽한 재전송 경로 가정 하에서는 멀티호밍을 사용하는 SCTP가 그렇지 않은 SCTP에 비해 더 높은 성능을 보임을 알 수 있다.

4.2.2 멀티호밍 호스트의 대체 경로 특성에 따른 성능 평가

이 절에서는 재전송을 위한 대체 경로의 경로 특성을 다양하게 설정해 보면서 SCTP의 성능을 살펴 보고자 한다. SCTP는 멀티호밍 호스트인 경우 프라이머리 경로로 패킷을 전송하고 프라이머리 경로에서 손실된 패킷을 재전송하는 경우에만 대체 경로를 이용한다. 인터넷 환경에서 두 경로는 다른 특성을 지니고 있다고 보는 것이 일반적이므로, 두 경로의 특성이 다른 경우 재전송 경로 선택에 따라 성능에 어떠한 영향을 주는지 살펴보고자 한다. 이를 위해 대체 경로로 재전송하는 현재의 SCTP 재전송 정책과 데이터가 손실되었던 이전 경로로 재전송하는 정책 두 가지를 서로 비교하였다.

그림 3은 프라이머리 경로의 패킷 손실율이 각각 0.2%, 2%, 4%인 경우 대체 경로의 패킷 손실율을 0 ~ 5%로 변화시켜 보면서 파일 전송 시간을 측정한 결과를 보인 것이다. 프라이머리 경로의 패킷 손실율이 4%일 때를 살펴보면 대체 경로의 손실율이 낮은 경우에는 현재의 재전송 정책을 사용하는 것이 상대적으로 짧은 파일 전송 시간을 가지나 대체 경로의 패킷 손실율이 약 2%를 넘어서면서부터는 처음 전송과 재전송이 같은 경로로 이루어지는 방식이 더 나은 성능을 보인다. 그리고 대체 경로의 손실율이 높아질수록 그 차이가 더 커지게 된다.

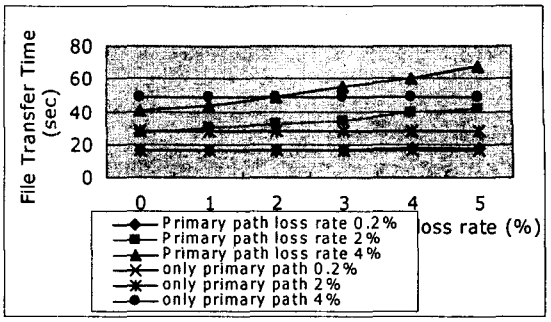

그림 3 경로의 패킷 손실율에 따른 지연

프라이머리 경로의 패킷 손실율이 2%인 경우도 유사한 결과를 보이며, 프라이머리 경로의 패킷 손실율이 0.2%인 경우는 프라이머리 경로에서의 패킷 손실율이 낮기 때문에 재전송 경로를 이용하는 빈도 자체가 적으므로 프라이머리 경로로만 재전송을 하는 방식과 전송 시간의 차이가 거의 없다. 위의 결과를 통하여 손실율 측면에서 볼 때 재전송을 위한 대체 경로가 데이터 전송을 위해 사용되고 있는 프라이머리 경로에 비해 신뢰성 있는 경로인 경우에만 두 경로를 동시에 사용하는 것이 성능 향상에 도움이 됨을 알 수 있다. 손실율 외에도 두 경로의 대역폭이나 전파 지연이 다른 경우에 실험에서도 이와 유사하게 대체 경로의 대역폭이나 전파 지연이 프라이머리 경로의 대역폭이나 전파 지연에 비해 일정 수준이상 좋은 경우에만 대체 경로로 재전송을 하는 것이 효과적임을 확인하였다.

SCTP 계층에서 대역폭이나 전파 지연은 RTT를 통해서만 파악이 가능하므로, 결론적으로 현재의 재전송 정책은 대체 경로가 프라이머리 경로에 비해 신뢰성 있으며, 짧은 RTT를 가질 때에만 두 개의 경로를 사용하는 것이 효과적임을 알 수 있다. 따라서 현재 SCTP의 재전송 정책은 이러한 조건을 만족하지 않는 경우에 오히려 성능 저하를 가져올 수 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 SCTP의 대표적인 특징 중 멀티호밍이 성능에 미치는 영향에 대해서 살펴보았다. 이를 통해 프라이머리 경로와 대체 경로의 손실율, RTT등의 차이가 프로토콜 성능에 큰 영향을 미치며, 대체 경로가 대역폭이나 전파 지연, 손실율 등에서 프라이머리 경로에 비해 나쁜 특성을 가질 때는 물론 비슷한 특성을 지니는 경우에도 대체 경로를 재전송 경로로써 사용하는 것이 오히려 성능 저하를 가져옴을 알 수 있었다. 따라서 대체 경로를 재전송을 위해 이용할 때에는 대체 경로들의 상태 파악이 중요한 요소가 됨을 알 수 있다. 따라서 대체 경로들의 상태를 파악하기 위한 방안과 파악한 정보를 바탕으로 적절한 재전송 경로를 선택하도록 하는 방안이 제안되어야 할 것이다.

참고문헌

[1] R. Stewart, Q. Xie, et al., "Stream Control Transmission Protocol," Proposed standard, RFC2960, October 2000.

[2] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, April 1999.

[3] A. Caro, K. Shah, J. Iyengar, P. Amer and R. Stewart, "SCTP and TCP Variants: Congestion Control Under Multiple Losses," Report TR2003-04 CIS Dept, U of Delaware, February 2003.