



WHITE PAPER

Aspera *fast*™ High Speed Transport

A Critical Technology Comparison



요즘 디지털 세상에서, 전세계에 걸친 대용량 디지털 데이터의 빠르고 신뢰성 있는 이동은 사실상 모든 산업에 걸쳐 사업 성공의 필수 사항이 되고 있다. 그러나, 전통적으로 이런 데이터 이동의 엔진 역할을 해 온 전송 제어 프로토콜(TCP: Transmission Control Protocol)은 성능 면에서, 특히 왕복 시간(RTT: Round-trip Time)이나 패킷 손실이 높은 망에서 병목 현상을 갖고 있으며(그림 1), 고 대역 망에서 가장 두드러진다. 이러한 내재된 "소프트" 병목 현상이 망의 가용 대역폭을 점진적으로 탐지하여, 패킷 손실이 감지될 때까지 전송률을 증가시키고 감지되면 전송률을 지수 감소시키는 TCP의 AIMD(Additive-Increase, Multiplicative-Decrease) 혼잡 회피 알고리즘으로부터 생긴다는 것은 잘 알려져 있다. 그러나 물리적 망 미디어에 의한 손실과 같은 또 다른 패킷 손실 원인들은 전송률을 동일하게 줄이는 망 혼잡과 연관이 없다는 것은 덜 알려져 있다. 사실, TCP AIMD 자체도 손실을 유발하며, 병목 현상에 똑같이 영향을 끼친다. 손실이 발생할 시점까지 전송률을 올리는 데 있어서, AIMD는 가용 대역폭을 선천적으로 증속하게 된다. 일부 경우에서, 이러한 자체-유도 손실은 실제로 다른 원인(예를 들어, 물리적 미디어 또는 트래픽 간 버스트 등)으로 발생하는 손실을 넘어서게 되어 무(無) 손실 통신 채널을 비 예측 손실률을 갖는 신뢰성이 떨어지는 채널로 바꾸게 한다.

TCP AIMD에서의 손실 기반 혼잡도 제어는 처리량에 치명적인 영향을 끼치게 한다: 모든 패킷 손실은 재 전송을 유발하며, 재 전송이 발생할 때까지 수신하는 애플리케이션에 데이터 배달을 멈추게 된다. 이는 모든 망 애플리케이션의 성능을 느리게 할 수 있으나 전송 순서(byte stream)가 필요하지 않는 파일 전송의 예와 같은, 대량의 자료 데이터의 신뢰성 있는 전송에 대해서는 기본적인 결점이라고 볼 수 있다.

TCP의 혼잡도 제어에 대한 신뢰도(재 전송)과의 결합은 광 대역 망에서의 FTP, HTTP, CIFS, 그리고 NFS와

같은 TCP 상에 구성된 전통적인 파일 전송 프로토콜의 취약한 성능으로 알 수 있듯이, 파일 전송에 대한 심각한 인위적 처리량 불이익을 만들어 낸다. 하드웨어 장비 혹은 TCP 대안을 기반으로 적용된 TCP 가속기와 같은 이러한 프로토콜에 대한 최적화는 왕복 시간과 패킷 손실률이 적당할 경우에는 어느 정도 전송률을 개선하지만, 그 이점은 광대역 거리에서는 상당히 감소한다. 게다가, 이 자료의 후반에서 보겠지만, 병렬 TCP 혹은 UDP 블래스팅(blasting) 기술은 분명히 보다 높은 처리량을 달성할 수 있는 대체 수단을 제공하지만 막대한 대역폭 비용을 초래한다. 이런 접근 방식은 이미 전송 중이거나 혹은 수신 데이터와 중복하여 상당한, 때론 엄청난 량의 불필요한 파일 데이터를 재 전송하며, 결국은 필요 보다 전송하는데 몇 배의 시간이 걸리며, 막대한 대역폭 비용을 야기한다. 특별히, 재 전송된 데이터 패킷을 제외한 유용한 비트 전송률 -즉, "유효 처리율(goodput)" -은 매우 낮아진다. 이러한 방법들은 대역폭을 쓰레기로 채움으로써 망 대역폭 활용도를 개선하는 것처럼 현혹하지만, 전송 시간은 아직 느리다!

TCP 최적화 혹은 단순한 블래스터들이 고속 유효 데이터 처리량을 달성하는 협소한 망 조건을 위해서, 망 중심 프로토콜과 마찬가지로, 그들은 저장 시스템으로 입출력하는 데이터 이동에서 보다 더 소프트 병목 현상에 부딪히게 된다.

자료 데이터를 최대 속도로 전송하는 것은 *전체 전송 경로* -즉 데이터 근원에서 목적지까지-에 따라 전송 중이거나 아직 수신되지 않은 데이터가 아닌 *유효 데이터 전송을 위한 가용 대역폭을 모두 사용하는* 종단간(end-to-end) 방식을 필요로 한다. 현재 상거래 인터넷 WAN 환경의 넓은 범위의 망 왕복 시간, 손실률 및 대역폭 수용 능력 특성에 걸쳐서 이런 목표를 달성하는 것은 자료 데이터 이동을 위한 새롭고 혁신적인 방식, 특히 신뢰도와 전송률 제어를 완전히 분리시키는 방식이 요구된다. 그런 방식의 신뢰도 체계에

서는, 100% 유효 데이터 처리량을 위해, 꼭 필요한 데이터만 재 전송해야 한다. 전송률 제어에 있어서는, 공유 인터넷 망 망에서의 범용 배치를 위해, 방식은 필요 시 최우선 전송을 위한 대역 폭 전용화 옵션을 제공하면서, 다른 전송 및 망 트래픽이 있을 경우, 대역 폭 공정성(fairness)과 혼잡 회피의 원칙을 유지해야 한다.

Aspera *fast*는 이러한 핵심 원칙 상에 구축된 공중 및 사설 IP 망 위에서 파일 전달을 위한 전통적 TCP 기반 전송 기술에 최적의 대안을 제공하기 위해 고안된 혁신적인 자료 데이터 전송 기술이다. 이는 표준 망 체계에 어떠한 변경도 가하지 않으면서, 중단간 애플리케이션 프로토콜로서 애플리케이션 층에 구현되었다. *fast*는 어떠한 IP 망 위에서도 -망의 지연 시간 및 패킷 손실에 무관하게- 자료 데이터 이동에 대한 궁극적인 고 성능 차세대 방식을 제공하면서 자료 데이터의 100% 대역 폭 효율 전송을 제공하도록 설계되었다.

이 자료에서는 상업적 그리고 학술적 측면 모두에서 대역폭 활용도, 망 효율, 전송 시간 측면에서 파일 기반의 전송을 가속하는 대안들을 설명하고, 그들의 성능과 실제 대역 폭 비용을 Aspera *fast*와 비교한다.

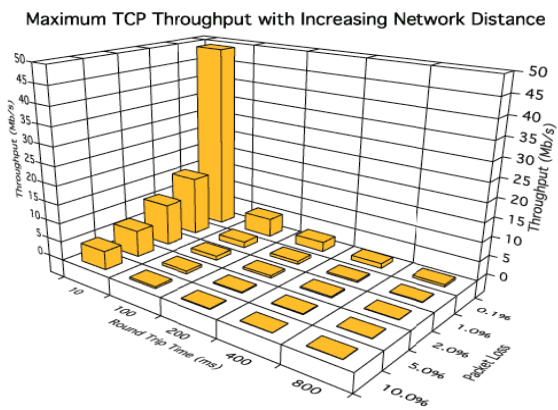


그림 1. 막대 그래프는 TCP를 사용하는 파일 전송 기술을 위한 OC-3(155 Mbps) 링크 상에서 다양한 패킷 손실과 망 대기 시간 조건 하에서 달성할 수 있는 최대 전송률을 나타낸다(주황색 부분). 처리량은 망 RTT 및 패킷 손실에만 의존하는 명백한 이론적 한계를 갖고 있다. 대역 폭을 추가하는 것이 유효 처리량을 변경하지 못한다는 것에 유의해야 한다. 파일 전송 속도가 개선되지 않으며 비싼 대역폭은 충분히 활용되지 않는다.

고속 TCP 개요

최근에, 이러한 변종들을 구현한 다수의 새로운 TCP 프로토콜의 초고속 버전과 TCP 가속 장치들이 개발되어 왔다. 초고속 TCP 프로토콜들은 AIMD의 기본적 결점들을 인지하고 자체 발생 인공적 병목 현상을 줄이고, 전체적 평균 처리량을 개선하기 위해 이런 윈도우 기반의 혼잡 제어 알고리즘을 개량했다. 이런 프로토콜들의 가장 진보된 버전들은 보통 손실 이벤트가 발생될 때까지 처리량 증가 보다는 망 큐잉 지연 시간과 같은 보다 많은 신호들을 측정함으로써 혼잡도 탐지 부분을 개선하는데 목표를 두고 있다. 이는 TCP 흐름이 패킷 손실을 만들지 않게 하여, 그래서 인위적으로 혼잡 회피에 들어가게 하고, 전체적인 처리량을 거의 무 손실 망 상태까지 개선할 수 있도록 도와 준다.

그러나, 이런 개선은 광 대역 망에서는 급격히 사라지는데, 여기서는 물리적 미디어 오류 혹은 트래픽 간 버스트(Cross Traffic Burst)에 따른 버퍼 오버플로우에 의한 패킷 손실을 무시할 수 없게 된다. 이러한 망에서는 단 하나의 패킷 손실도 TCP 송신 윈도우 크기를 심하게 줄어들게 하기 때문에, 다수의 손실은 데이터 처리량에 최악의 영향을 끼치게 될 것이다. 윈도우 당 하나 이상의 패킷 손실은 보통 전송 타임 아웃을 야기하며 그 결과로 송신 측과 수신 측간의 대역폭과 지연의 곱(bandwidth-delay-product) 파이프 라인은 비게 되고 데이터 처리량은 0으로 떨어진다. 필연적으로 송신 측은 데이터 전송의 느린 재 시작을 하게 된다.

이에 반해, Aspera *fast*에서는 전송률이 손실 이벤트와 연계되어 있지 않다. 손실 데이터는 중단간 요구 대역폭과 같은 율로 재 전송된다. 재 전송은 사실상 이상적 대역 폭 효율을 보인다 - 중복 전송되는 데이터가 없고 총 목표 성능은 충분히 활용된다.

그림 2에서 보는 바와 같이, 초 고속 TCP의 상업화된 버전 중에 하나(여기에는 CUBIC, H-TCP, BIC 등과 같은 변종이 포함된다)인 FAST TCP의 처리량은 1%의 패킷 손실이 있는 망 상에서는 낮은 지연 율을 갖는 망의 표준 TCP Reno를 상회하는 처리량을 개선하지만, 개선 효과는 국가 간 그리고 대륙 간 링크 전형인 높은 왕복 시간의 경우에는 급속히 떨어진다. 이에 반해 *fasp* 처리량은 망 지연 시간의 증가에 따른 성능 저하가 없으며 100% 효율적 전송과 대역 폭의 95%가 넘는 유효 파일 전송 처리량을 달성한다. 이와 유사하게, 패킷 손실이 증가(예를 들어, 5% 혹은 그 이상의 손실률)됨에 따라 *fasp* 처리량은 단지 동일한 양만큼만 감소한다. 보다 높은 손실률에서 가속화 TCP의 처리량은 Reno와 비슷하다.

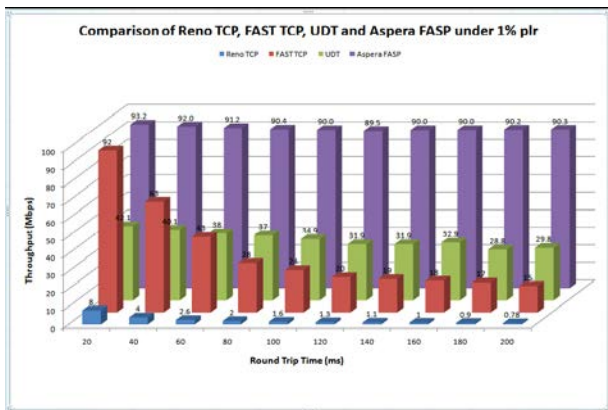


그림 2. 중급 패킷 손실(1%)인 링크에서 Reno TCP, 상업적으로 사용 가능한 고속 TCP, UDT, 그리고 Aspera *fasp*를 비교하는 1GB 파일에 대한 파일 전송 처리량. 가속화된 TCP가 낮은 지연 시간의 망에서 Reno 처리량을 개선하는 반면, 그 개선 효과가 국가 간 및 대륙 간 연결과 같은 높은 왕복 시간에서는 급격히 떨어짐에 유의한다. 이에 반하여 *fasp* 처리량은 망 지연 시간의 증가에 따른 성능 저하가 없다. 이와 비슷하게, 패킷 손실이 증가(예를 들어, 5%의 손실률)됨에 따라 *fasp* 처리량은 단지 동일한 양만큼만 감소하지만, 초 고속 TCP는 Reno와 비슷하다.

패킷 손실에 대한 표준 및 초고속 TCP의 반응은 송신 단이 불규칙한 전송 속도를 만들어 내는 송신 윈도우 크기를 줄이게 할 뿐만 아니라, TCP 프로토콜이 송신 윈도우 내에서 순서적 전달 보증을 유지하도록 재 전송된 패킷으로 새로운 패킷을 선점하게 한다. 동일한 TCP 송신 윈도우에서의 이런 새로운 또는 재전송

패킷의 전송은 전송 무결 성을 보장하는 TCP 신뢰성 제어를 갖는 TCP 혼잡도 제어의 성능 저하와 얽히게 만들고, 자료 데이터와 같이 순서적 배달이 요구되지 않는 애플리케이션을 위한 전송 처리량을 불필요하게 불리하게 한다.

TCP 신뢰 성은 유실 데이터가 없고(손실 패킷들은 수신 단에 의해 탐지되고 후에 송신 단에 의해 재전송될 것이다), 수신된 데이터는 순서적으로 애플리케이션에 전달된다는 것이다. 이러 두 가지 보장을 달성하기 위해서, TCP는 손실 패킷을 재 전송할 뿐만 아니라 먼저 도착한, 순서가 뒤바뀐 패킷(커널 메모리에 임시적으로 상주하고 있는)을 빠진 패킷이 도착하기 전까지 멈추게 해야 하고, 수신된 데이터는 애플리케이션 층에 순서적으로 전달될 수 있어야 한다. 수신 단은 인입된 패킷들을 RAM에 저장해야 한다는 요구 사항으로 볼 때, 재 전송은 시급하고 최 우선적 이어야 하며, 새로운 데이터를 전송하는 것은 보조를 맞추는 면에서 느려져야만 한다. 보다 상세하게, 매 패킷 손실 이벤트가 일어날 때 마다 수신 단에서 바이트 스트림에서의 "구멍"을 채우기 위한 재 전송 패킷들을 기다리면서 새로운 패킷들은 느려져야 한다 (보통 송신 윈도우는 손실된 패킷들이 수신 단에 재 전송되고 인지될 때까지 동결된다). 본질적으로 TCP에서 신뢰 성과 흐름 제어(혹은 혼잡 제어)들은 설계상, 완전히 연계되어 있다.

비록 이런 체제 유형은 TCP에게 많은 애플리케이션에 의해 요구되는 엄격한 순서적 바이트 스트림 전달을 제공하지만, 파일 전송과 같이, 선천적으로 엄격한 바이트 순서를 요구하지 않는 애플리케이션에는 큰 손실을 끼치게 되어, 결국은 이런 애플리케이션들에게 자신에 해당하는 데이터 처리량을 제한하는 감춰진 인위적 병목 현상을 유도하게 된다.

보다 명확하게 하기 위해서, 우리는 매 손실 마다 1/8씩의 윈도우 크기 축소를 하는 초고속 TCP에서 한

개의 혼잡도와 무관한 패킷 손실에 의한 처리량 손실을 계산하는 간단한 예를 하나 연구해 볼 수 있다. 1%의 패킷 손실률과 100ms의 왕복 시간 지연을 갖고 있는 1GB 망에서, 매 단일 패킷 손실은 8분의 1씩 율을 줄이게 하고(TCP Reno에서는 2분의 1) 송신 단에서 패킷 손실 이벤트 전까지 초기 전송 속도(1Gbps)를 회복 하는데 걸리는 시간은 $1\text{Gbps} \div 8(\text{bits/byte}) \div 1024(\text{bytes/packet}) \times 100\text{ms} \times 0.125$ (drop 율/손실) $\times 100\text{ms} \approx 152.5$ 초의 시간이 걸린다. 이런 복구 기간 동안, 초고속 TCP는 단 하나의 손실 발생 때문에 약 $152.6\text{s} \times 1\text{Gbps} \times 0.125/2 \approx 9.6\text{GB}$ 처리량의 손실을 가져온다! 현실의 광 대역 망에서, 실제 값은 더 클 수 있는데, 이는 망 큐잉, 물리적 미디어 접근, 스케줄링 및 복구 등의 이유로 RTT가 더 커질 수 있기 때문이다. 그러므로 보통 송신 단이 복구되는데 152.6초 이상 걸리게 된다. 다수의 연이은 패킷 손실은 재앙이 될 것이다. 인터넷 엔지니어링 태스크 포스(IETF: Internet Engineering Task Force)로부터의 인용 문구는 직설적으로 그 영향을 아래와 같은 식으로 표현한다:

“윈도우 크기를 LFN(long fat network)의 처리 능력에 맞추기 위해 확장하는 것은 윈도우 당 하나 이상의 패킷이 폐기(drop)된다는 가능성을 그 만큼 증가시키는 결과를 초래한다. 이는 LFN 상에서 TCP 처리량에 대단히 큰 손실 효과를 갖게 될 수 있다. 게다가, 무작위 폐기의 일부 형태를 기반으로 하는 혼잡 제어 체제인 RFC 1323 표준이 게이트웨이에 도입되어 왔고, 무작위 간격 패킷 폐기가 일반화된 이후로; 이는 윈도우 당 하나 이상의 패킷을 폐기할 확률을 증가시킨다.”

우리는 이러한 전송률 저하 혹은 처리량 손실이 때론 엄격한 순차적 전달이 필수 사항인 바이트-스트림 애플리케이션 들에게는 꼭 필요하다는 것을 주목해야 한다. 그렇지 않으면, 이전의 예 에서와 같이 적어도 하나의 RTT에 대한 각 TCP 연결 마다 한 개의 유실

패킷을 기다리는데 만 적어도 $1\text{Gbps} \times 100\text{ms} \times 0.125 \approx 1.5\text{MB}$ 의 추가 데이터를 RAM이 수용해야 한다. 그러나, 이런 저하는 비순차적 데이터의 경우 이러한 유실 패킷을 기다리지 않고 디스크에 쓰여질 수 있기 때문에 파일 전송 애플리케이션에는 불필요하게 되는데, 이는 필요 시기에 최신의 전송률 제어 체제에 의해 측정되는 망 내부의 가용 대역 폭과 동일한 속도로 재 전송될 수 있다는 것이다.

실제로 TCP 자체는 신뢰성과 혼잡 제어를 분리할 수 없을 것이며 그래서 TCP 목적 -신뢰성 있고, 바이트-스트림 형식의 전달 제공-이 IETF에 의해 재 정의되지 않는 한 이런 인위적인 병목 현상을 없애지 않을 것이다. 신뢰성 있는 스트리밍과 비 스트리밍 애플리케이션 모두에 대한 단일 전송 제어 프로토콜 상에서의 전형적 의존성이 이 두 영역 모두에 대해서 차선책이라는 것이 실제로 증명되었다.

UDP 기반 초 고속 솔루션

TCP에 의해 제공되는 신뢰성은 망 처리 능력을 저하시키고, 평균 지연 시간을 증가시키며 지연 변위(delay jitter)를 악화시킨다. 혼잡 회피로부터 신뢰성을 분리해내기 위한 노력이 수년 동안 이루어져 왔다. 하지만 TCP 자체의 변경이라는 복잡성 때문에, 최근 몇 년 동안 학계와 산업계의 방향성은 분리된 전송 율과 신뢰성 제어를 특징으로 하는 애플리케이션 -수준에서의 프로토콜을 추구하고 있다. 이런 방식은 TCP 대안으로 전송 층에서 UDP를 사용하고 애플리케이션 계층에서 신뢰성을 구현한다. 이와 같은 방법을 사용하는 대부분을 UDP 블래스터라고 하지만 - 이 방식은 유실 데이터의 재 전송을 위한 일부 방법을 채택해서 UDP로 신뢰성 있게 데이터를 전송 - 가용 대역폭에 대한 중대한 고려가 없으면 자신의 처리량은 말할 것도 없이 망의 붕괴의 위험이 있다. 그림 3은 상품화된 UDP 블래스터인 RocketStream

제품이 일반적 WAN 조건을 갖고 있는 300Mbps 링크 위에서 동작했을 경우의 처리 능력을 나타낸 것이다(RTT 및 패킷 손실률을 증가하면서).

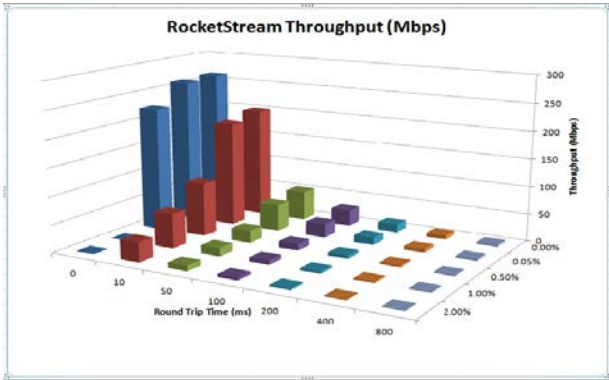


그림 3. 막대 그래프 그룹은 RocketStream 제품에 대해 300Mbps 상에서 여러 가지 패킷 손실과 망 지연도 조건(WAN) 하에서 달성된 처리량을 나타낸다. 높이가 0인 막대들은 전송 단과 수신 단 간의 연결 설정 실패는 표시하는데, 이는 RTT 혹은 패킷 손실률이 높은 경우 흔히 볼 수 있다.

오픈 소스 개발품인 Tsunami 및 UDT(Signant, File Catalyst 및 Sterling Commerce와 같은 제품에서 사용된)을 포함하는 최신 UDP 솔루션들은, 패킷 손실이 발생했을 때 전송 율을 줄이는 단순화한 알고리즘을 통하여 UDP 블래스터에서의 혼잡도 제어를 강화시키고 있다. 이 Back-Off 기법이 대역폭, 왕복 지연, 패킷 손실 및 Flow 수 간의 단일 조합을 의미하는 케이스 별 특정 망 선로에 대한 적절한 성능을 얻기 위해 튜닝 될 수는 있지만, 그 설계는 태생적으로 실제 인터넷 망에 있는 망 RTT 범위, 패킷 손실 조건 및 동시 Flow에 맞추어질 수 없다. 결과적으로, 이러한 접근 방식은 가용 대역폭을 모두 사용하지 못하거나 모두 사용하게 되지만, 망을 여분의 데이터 전송을 하도록 증속(増速)하는 과정에서 - 기존 망 상태 하에서 무려 50%에 달하는 여분의 데이터 - 처음에는 대역폭을 낭비하게 되고, 다음에는 유효 파일 전송 처리량("유효 처리율")을 붕괴시킨다. 궁극적으로, 이 과정에서 이러한 방식들은 그들의 증속(増速)이 다른 TCP 애플리케이션 때문에 패킷 손실을 발생시킴에 따라 다른 트래픽에 의해서 망을 사용하지 못하는 상태로 만들 수 있고 그들의 유효 처리량을

멈추게 된다.

우리는 이러한 문제를 증명하기 위해 가장 최신의 재전송(NACK-기반) UDP 전송 솔루션 중의 하나를 채택했으며, 이를 회사에 의해 재 포장된 UDT라 한다. 특별히 UDT는 다음과 같은 특성을 포함하고 있다:

○ 저급의 혼잡도 회피. UDT에 채용된 동적 AIMD 알고리즘(D-AIMD)는 AIMD 알고리즘과 유사하게 동작하지만 전송 율이 증가됨에 따라, 율 증가 속도를 줄이는 AI(additive increase) 변수를 줄이게 한다. 이 방식은 앞에서 언급한 TCP의 핵심 이슈들을 - 신뢰성 및 전송 율 조절의 복잡한 관계 - 인식하지 못하며, 대신에 하나의 변수를 튜닝 하면 AIMD 및 심지어 TCP의 저 성능 문제를 풀 수 있다는 가정을 한다. 실제로, 특별히 튜닝 된 D-AIMD는 처리 능력 면에서 하나의 시나리오에서 TCP를 능가하지만 다른 경우에는는 곧바로 TCP 보다 성능이 떨어진다. 그러므로, 대부분의 WAN 구성에서, UDT의 처리 능력은 실제로 TCP 보다 못하다.

○ UDT의 공격적 데이터 전송 체제는 자체적인 전송 성능을 약화시킬 뿐 아니라 타 트래픽을 위태롭게 하며, 전체 망 성능을 저하시키는 극적인 전송 율 진동 현상과 패킷 손실을 야기한다.

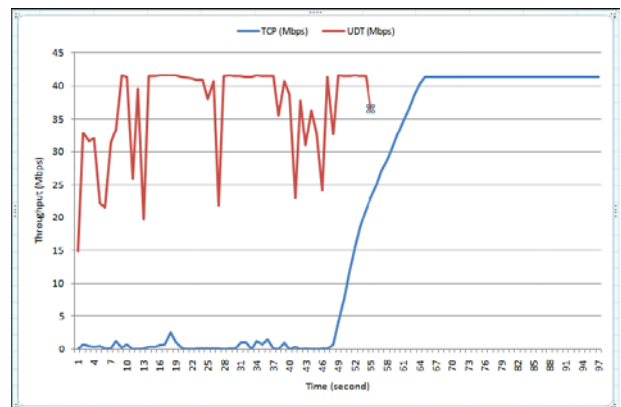


그림 4. 0% 패킷 손실률 및 50ms RTT를 갖는 전형적 T3 연결 상에서의 단일 UDT 전송의 파일 전송 처리량과 일반적 TCP 흐름 상에서의 UDT 전송 효과. TCP 흐름은 UDT 흐름이 종료될 때까지 대부분 기간 내내 "가시화"되지 않는다.

일반적 TCP 흐름(즉, 웹 클라이언트의 HTTP 세션)이

UDT 전송과 대역폭을 공유하는 전통적인 WAN 상황에서, TCP 흐름은 UDT 흐름의 공격성 때문에 잠재적으로 서비스 거부(DOS : denial of service)를 경험할 수 있다.(그림 4) 실제로, 극단적 TCP 비우호성의 가능성은 최초 UDT 논문에서 이론적으로 연구되었고 저자는 이러한 극단적 비우호성을 회피하기 위해 만족해야 되는 특정 조건을 제안했다. 실제로, 매우 전형적인 WAN 환경(즉, 100ms RTT 및 0.1% PLR 인 WAN)에서, 그 조건은 만족될 수 없었으며 그래서 극단적 TCP 비우호성은 피할 수 없었다. 이것은 UDT 기반의 데이터 전송 솔루션을 사용하기 위해서는, 단골 고객이 UDT가 전체 망 생태계(즉, 웹, 이 메일, 인터넷 전화, 망 관리)의 운영에 피해를 입히지 않을 것을 보장하는 일부 QoS 장비에도 시간과 돈을 투자해야 한다는 것을 의미한다.

○ UDT에서 공격적 전송과 잘못된 재전송은 값진 대역폭의 효율성을 저하시키고 고객이 불필요하게 보다 많은 대역폭을 구매하도록 강요한다.(실제로 사용하지 못하는 비싼 대역 폭을 보다 잘 활용하도록 의도된 그 솔루션!) 일부 실험(그림 6과 7)에서의 전송률, 수신률 및 유효 파일 전송률 간의 큰 차이는 UDT의 매우 공격적 데이터 주입 및 결함이 있는 재전송 체제로 인한 라우터 및 수신 단에서의 대량의 데이터 폐기를 나타낸다. **일부 전형적인 광 대역 망에서 측정 효율("유효 처리율")을 20% 이하로 떨어진다. 그것은 UDT에 의해 100% 완전 가용된 망은 중복된 데이터를 전송하거나 초과된 버퍼에 유용한 데이터를 전송함(UDT 자체에 의해 증속됨)으로써 대역 폭 성능의 80%를 사용한다는 의미가 된다.**

단골 고객에 대해 UDT를 사용하는 "이점"과 "비용"은 정확한 비교를 위해 정량화될 수 있다. "이점"이 빠른 전송 시간으로 직역되는 필요 데이터의 전송(유효 전송률)을 위한 대역폭의 효과적인 사용으로 측정될 수 있는 반면, "비용"은 얼마나 많은 중복 데이터

사본들이 상대 망 단에 있는 애플리케이션 계층으로 성공적으로 전달되는 하나의 필요 패킷을 얻기 위해서 전송되는가로 정의되는, 하나의 필요 데이터 패킷을 전송하는 노력으로 요약될 수 있다. 이러한 비용은 실제로 다른 전송자에 대한 유도 비용을 의미하기도 하는데, 이는 공정한 대역 폭 공유(그림 4)에 대한 그들의 손실과 이로 인한 저하된 처리량이 반영된다. 구체적으로 말하면, 그림 5에서 이미 일부 반영된 바와 같이, UDT는 넓은 범위의 WAN 조건에서 낮은 유효 전송 처리량(느린 전송으로 나타남)을 갖고 있고, 그래서 사용자에게 이득을 거의 제공하지 않는다. 그리고, 증속과 추가적인 재 전송에 따른 관련 대역 폭 비용은 다른 업무 흐름에 많은 영향을 끼친다.

그림 5는 서로 상이한 RTT와 패킷 손실률 하에서 T3 (45Mbps) 링크 상의 단일 UDT 전송에 의해 하나의 패킷을 전송하는 전체 비용을 나타낸 것이다. **대부분의 전형적인 광 대역 망에 대해서, 하나의 패킷 전송은 8에서 10개의 재 전송이 필요하다! 다시 말해서, 1기가바이트 파일을 전송하기 위해, UDT 전송 단은 최대 9에서 11기가바이트를 망에 복사하게 된다. 전송은 필요 보다 9 ~ 11배 더 걸리게 되고, 또한 다른 전송 흐름에 많은 패킷 손실을 야기할 수 있다.**

그림 5에서 비용은 UDT 전송 단에서의 매우 공격적인 주입율과 UDT 수신 단에 의해 폐기되는 중복된 재 전송에 의해 발생된다. 좀더 자세하게 말하면, 우리는 전송 단에 의해 매우 공격적인 주입에 따른 손실을 반영하기 위해 송신 비용을, 그리고 수신 단에서 폐기되는 중복 재 전송을 반영하기 위해 수신 비용을 정의할 수 있다. 정확하게는, 송신 비용은

$$\text{송신비용} = \frac{\text{전송 총 바이트 수} - \text{수신 실제 바이트 수}}{\text{수신 실제 바이트 수}}$$

그리고 수신 비용은

$$\text{수신 총 바이트 수} - \text{실제 유효 바이트 수}$$

수신비용 =

실제 유효 바이트 수

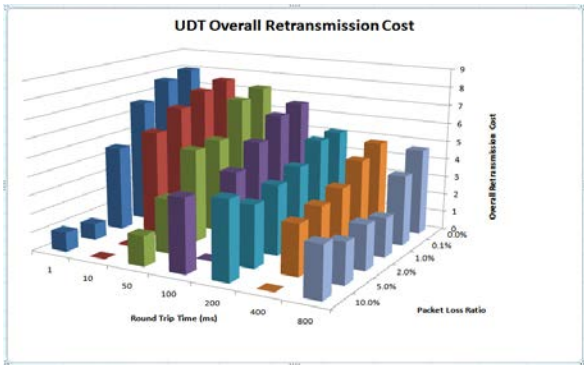
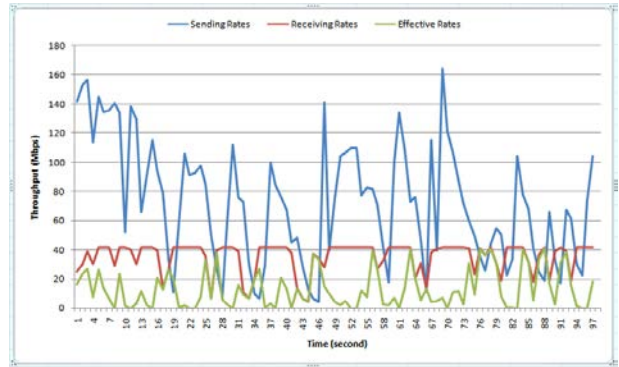
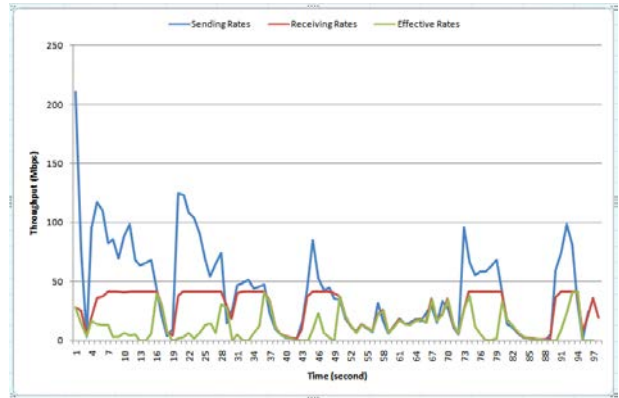


그림 5. 막대 그래프는 상이한 RTT와 패킷 손실을 하에서의 단일 UDT 재 전송을 나타낸다. 각 막대의 높이, “전송 비용”은 하나의 1 기가바이트 파일이 전송될 때 기가바이트 단위로 재 전송된 양이다. 높이가 0인 막대는 전송 단과 수신 단 간의 연결이 실패한 것을 나타내는데, 이는 RTT 혹은 패킷 손실이 클 경우 UDT에서 흔히 나타난다. 원본 파일 크기 보다 9배까지 전송되고 낭비적인 재 전송이 된다는 것을 유의한다!



(b) 1% plr 및 100ms RTT를 갖는 T3 상에서의 UDT 전송



(c) 5% plr 및 200ms RTT를 갖는 T3 상에서의 UDT 전송

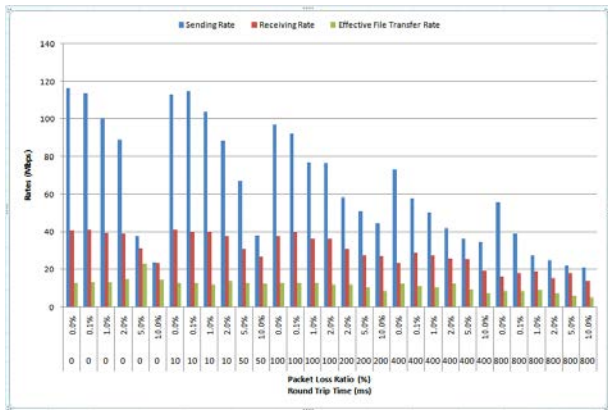
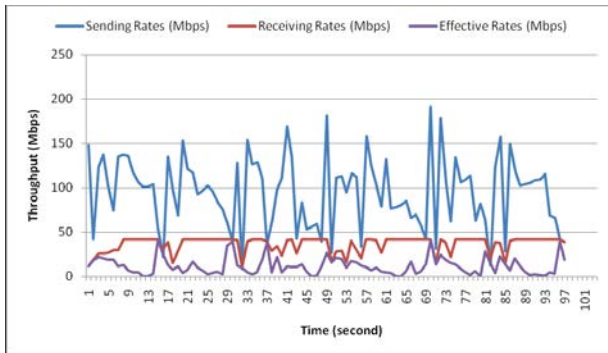


그림 6. 막대 그래프는 T3 링크 상에서 상이한 RTT 및 패킷 손실 을 하에서의 단일 UDT 전송의 송신 율, 수신 율, 그리고 유효 전송 율을 나타낸다. 송신 율과 수신 율 간의 큰 차이는 개입된 망 경로 상의 패킷 손실이 크다는 것을 의미하고, 수신 율과 유효 수신 율의 큰 차이는 중복적인 재 전송의 수가 크다는 것을 의미한다.



(a) 0% plr 및 100ms RTT를 갖는 T3 상에서의 UDT 전송

그림 7. 0%, 1%, 5% 패킷 손실 율(plr: packet loss ratio) 및 100ms, 100ms, 200ms RTT를 갖는 T3 링크 상에서의 단일 UDT 전송의 송신 율, 수신 율, 유효 수신 율. 송신과 수신 율 간의 격차는 라우터에서의 많은 데이터 손실을 의미하며, 수신과 유효 수신 율 간의 차이는 많은 수의 UDT 수신 단에서의 중복적인 재 전송의 폐기를 반영한다.

송신 비용이 높아 갈수록, 더 많은 패킷이 라우터에서 폐기되며, 수신 비용이 높아 갈수록 수신 단에서 보다 많은 패킷이 폐기된다는 것에 유의한다. 그림 7은 T3 링크에서 서로 상이한 RTT와 패킷 손실 율 하에서 단일 UDT 전송에 대한 송신 율, 수신 율, 그리고 유효 수신 율을 나타낸다. 송수신 율(수신 율에 대한 송신 율 및 유효 수신 율에 대한 수신 율)은 위의 정의된 비용들이 될 것이다. 우리는 송신 율이 수신 율보다 지속적으로 높고, 이는 다시 모든 망 구성에서 유효 수신 율 보다 도 지속적으로 높다는 것을 관찰할 수 있다. 이러한 비용은 망를 망 활용도 (대역폭 대비 처리율)가 1에 근접하도록 하지만 망 효율성(대역폭 대비 유효 처리율)은 기껏해야 15% 정도가 된다.

결과적으로, 어떠한 파일 전송도 실제보다 6배 느리게 된다!

명확하게 하기 위해서, 우리는 서로 상이한 광 대역 망 하에서 다음과 같은 성능 관련 질문에 대답함으로써 간단한 파일 전송을 통한 상기의 비용을 증명할 수 있다:

- 전송해야 하는 바이트 수는?
- 실제 전송된 바이트 수는?
- 실제 수신된 바이트 수는?
- 걸린 전송 시간은?
- 유효 파일 전송 속도는?

결과는 테이블 1에 정리되어 있으며, 동일 조건에 대해 Aspera *fasp*와 비교되어 있다.

표 1에서 보여진 UDT 파일 전송의 성능의 직접적인

결과는 유용한 데이터가 망을 전혀 통과할 수 없거나, 망 효율성(테이블 1의 8번째 칼럼에서 표시)을 희생해서 통과하는데, 이는 낮은 성능을 더욱 악화시킬 뿐만 아니라 대역폭을 포화 상태로 만들어 다른 트래픽에 대한 서비스 거부 현상을 야기한다. 일부 UDT 및 TCP 솔루션에서 채택된 것과 같이 보다 높은 송신율 및 망 활용도를 위해 병렬 전송을 하는 것은 대역폭 낭비를 악화시킬 뿐이며 고객으로 하여금 좀 더 많은 대역폭을 너무 일찍 투자하도록 강요하게 한다는 것에 주목해야 한다. 망 활용도 및 데이터 처리량의 결과적 개선은 거의 없으며 오히려 결과 비용(그림 8)은 급격하게 증가된다. 재전송은 두 개의 UDT 연결에서 추가적으로 40%나 증가되었다. 동일한 예에서(표 1), UDT는 1GB 미만의 파일을 성공적으로 전달하는데 망에 13GB에서 15GB 정도를 복사하게 된다. 병렬 TCP 및 UDT 전송 기법을 사용하는 솔루션들은 그림 8에서 나타난 것과 유사하거나 더 심각한 성능을 보인다.

테이블 1. 전형적 WAN 상에서의 UDT 파일 전송 - 높은 대역폭 비용 및 낮은 전송율

Bandwidth (Mbps)	RTT (ms)	plr (%)	How much to be sent (MB)	How much needs to be sent (actual data + inevitable loss by media, MB)	How much data actually sent (MB)	Sending Cost (Sender's Overhead, %)	How much data actually received? (MB)	Receiving Cost (Receiver's Overhead, %)	How long does it take? (s)	Effective file transfer speed (Mbps)	Observed Network Utilization	Network Efficiency (Effective Utilization, %)
45	0	0	953.7	953.7	9093.2	314.1 %	2195.8	130.2 %	625.0	12.8	66.2 %	28.4 %
45	100	1	953.7	963.2	5941.8	234.9 %	1774.0	86.0 %	618.0	12.9	54.1 %	28.8 %
45	400	5	953.7	1001.4	3764.1	150.6 %	1501.8	57.5 %	830.0	9.6	34.1 %	21.4 %
45	800	5	953.7	1001.4	3549.9	152.9 %	1403.9	47.2 %	1296.0	6.2	20.4 %	13.7 %
100	100	1	953.7	963.2	1413.0	14.0 %	1239.8	30.0 %	239.0	33.5	44.0 %	33.5 %
100	200	5	953.7	1001.4	2631.2	19.6 %	2200.1	130.7 %	571.8	14.0	32.6 %	14.0 %
300	100	1	953.7	963.2	1060.0	2.1 %	1038.4	8.9 %	232.0	34.5	12.6 %	11.5 %
300	200	1	953.7	963.2	1083.0	2.3 %	1059.1	11.1 %	273.0	29.3	11.0 %	9.8 %
500	200	1	953.7	963.2	1068.9	1.7 %	1051.5	10.3 %	252.0	31.8	7.1 %	6.4 %
500	200	5	953.7	1001.4	1660.9	5.3 %	1576.7	65.3 %	539.1	14.8	5.0 %	3.0 %

테이블 2. 전형적 WAN 상에서의 Aspera *fasp* 파일 전송 - 0에 근접한 대역폭 비용 및 빠른 전송율

- ASPERA FASP 고속 전송 - 핵심 기술 비교 -

Bandwidth (Mbps)	RTT (ms)	Plr (%)	How much data to be sent? (MB)	How much needs to be sent (actual data + inevitable loss by media, MB)	How much data actually sent? (MB)	Sending Cost (Sender's Overhead, %)	How much data actually received? (MB)	Receiving Cost (Receiver's Overhead, %)	How long does it take? (s)	Effective File Transfer Speed (Mbps)	Observed Network Utilization (by receiver)	Network Efficiency (Effective Utilization, %)
45	0	0	953.7	953.7	953.7	0.0 %	953.7	0.0 %	185.4	43.1	98.5	95.9 %
45	100	1	953.7	963.2	963.3	1.0 %	953.7	0.0 %	187.8	42.6	97.1	94.6 %
45	400	5	953.7	1001.4	1002.1	5.0 %	954.3	0.1 %	197.0	40.6	92.1	90.3 %
45	800	5	953.7	1001.4	1003.5	5.1 %	955.2	0.2 %	197.0	40.6	91.6	90.3 %
100	100	1	953.7	963.2	963.3	1.0 %	953.8	0.0 %	85.0	94.1	96.3	94.1 %
100	200	5	953.7	1001.4	1002.4	5.0 %	954.5	0.1 %	88.9	90.0	91.9	90.0 %
300	100	1	953.7	963.2	964.0	1.0 %	954.4	0.1 %	29.3	273.4	92.6	91.1 %
300	200	1	953.7	963.2	964.7	1.0 %	955.1	0.1 %	29.2	274.3	91.9	91.4 %
500	200	1	9536.7	9632.1	9635.0	1.0 %	9539.0	0.0 %	181.6	440.6	90.6	88.1 %
500	200	5	9536.7	10013.6	10018.5	5.0 %	9541.2	0.0 %	186.9	428.0	88.0	85.6 %

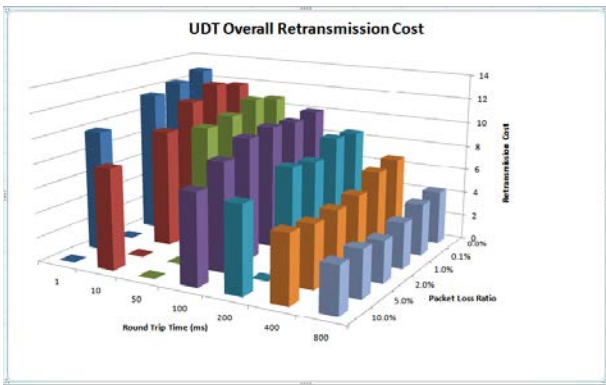


그림 8. 병렬 UDT 스트림의 대역폭 비용. 막대 그래프는 T3 망의 상이한 RTT 및 패킷 손실 을 하에서 하나의 1GB 파일 전송을 위한 두 개의 병렬 UDT 연결의 재전송 비용을 나타낸다. 전송 비용을 나타내는, 각 막대의 높이는 1GB 파일이 전송될 때 기가바이트 단위의 재 전송 량을 나타낸 것이다. 높이가 0인 막대는 송신 단과 수신 단의 연결 설정에 실패한 것을 의미하는데, 이는 RTT 혹은 패킷 손실 을 중에 하나라도 높을 때 UDT에 흔하게 나타난다. 그 과정에서 파일 크기의 14배인 거의 14GB 데이터가 재전송된다는 것에 유의한다.

ASPERA FASP 솔루션

Aspera *fasp*는 바이트-스트림 전송을 요구하지 않는 애플리케이션을 위한 신뢰성 있는 전송을 제공하고 신뢰성과 전송률 제어를 완전히 분리함으로써 TCP에 의해 남겨진 공백을 채운다. 이는 전송 계층에서는 표준 UDP를 사용하고 애플리케이션 층에서는 채널 상에서 정확하게 실제 패킷 손실 분만을 전송하는 이론적 최적 방법을 통해 분리된 혼잡도와 신뢰성 제어를 달성한다. 전송률 제어와 신뢰도의 분리로 인하여, 새로운 패킷들은 TCP 기반의 바이트-스트리밍 애플리케이션과 같이 유실 패킷을 재 전송하는데 속도 저하를 필요치 않는다. 전송 과정에서 유실된 데이터는 내부 중단간 경로의 허용 대역 폭과 동일한 율이나, 혹은 수신 비용을 없애기 위한 중복 재 전송 값이 0인, 어떤 설정 목표 율로 재전송된다.

내부 경로의 허용 대역폭은 0에 가까운 전송 비용을 위해서 지연-기반 전송률 제어 체제에 의해 알 수 있다. 특히, *fasp* 적응적 전송률 제어는 평가 큐잉 지연(measured queuing delay)을 망에서 작고, 안정된 량의 "큐잉"을 유지를 지향하는 망(혹은 디스크-기반) 혼잡도의 주 경보 표시로 사용한다; 전송 율은 평가 큐잉이 목표 치보다 낮아질 경우(일부 대역 폭이 사용되지 않은 상태이며 전송이 빨라질 필요가 있음을 표시)에는 상황 조정하고, 큐잉이 상기의 목표치 보다 증가될 경우(대역 폭이 모두 사용 중에 있고 혼잡이 높음을 표시)에는 하향 조정한다. 정기적으로 탐사 패킷(probing packet)을 망에 전송함으로써, *fasp*는 전송 경로에 걸친 큐잉 지연에 대한 정확하고 시기 적절한 측정치를 얻을 수 있다. 큐잉 지연이 증가된다고 감지되면, *fasp* 세션은 자신의 전송률을 목표 큐잉과 현재의 큐잉 간의 차이에 비례하여 줄이게 되고, 그래서 망 증속을 회피한다. 망 혼잡이 안정화되면, *fasp* 세션은 재빨리 목표 큐잉의 비율에 따라 전송률을 증가시키고 그래서 망 허용 처리량의

100%를 사용하기 위해 다시 출력을 상승시킨다.

TCP의 전송률 제어와는 다르게, *fasp* 적응적 전송률 제어는 몇 가지 주요 이점을 갖고 있다: 첫째로, 이는 망 큐잉 지연을 주 혼잡도 신호로 사용하고 패킷 손실을 이차적 신호로 사용하여, 결과적으로 미디어에 의한 패킷 손실을 갖는 망 상에 인위적으로 속도를 느리게 하지 않으면서도, 망 혼잡의 정확한 예측을 얻을 수 있다. 둘째로, 내장된 신속 반응 체제는 많은 동시 병행 전송들이 있을 때 안정되고 높은 처리량이 가능하도록 초 고속 파일 전송이 자동적으로 속도를 낮출 수 있게 하지만, 가장 효율적 전달 시간을 위해 미 사용 대역 폭을 모두, 그리고 효율적으로 사용하기 위해 자동적으로 출력을 상승시킨다. 셋째로, 선진 피드백 제어 체제는 *fasp* 세션 전송률이 혼잡한 라우터에서 큐로 들어가는 목표량의 비트 수가 버퍼로 주입되는 안정된 균형 전송률로 빠르게 수렴할 수 있게 한다. 안정된 전송 속도와 큐잉 지연은 추가적인 QoS(Quality of Service) 하드웨어 또는 소프트웨어를 투자하지 않고서도 QoS 경험을 최종 사용자에게 가져다 줄 수 있다. 데이터 전달 시간은 예측 가능하게 되고 데이터 이동은 동일 망을 공유하고 있는 다른 애플리케이션에 투명하게 된다. 넷째로, 대역 폭의 전체 사용은, NACK 기반의 UDP 블래스터와는 다르게, 망에 사실상 비용이 없도록 했으며 망 효율은 거의 100%에 가깝게 유지하게 한다.

Aspera FASP Throughput (Mbps)

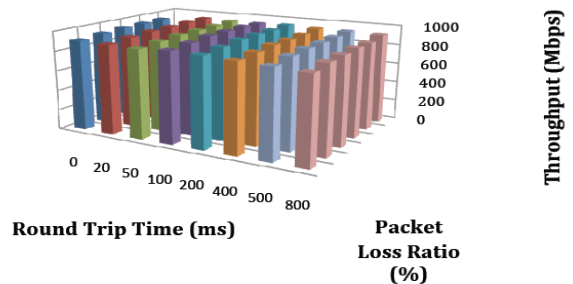


그림 9. 막대 그래프는 *fasp* 혁신적 전송 기술을 사용하는 파일 전송을 위한 1Gbps 링크에서 다양한 패킷 손실 및 망 지연 조건 하에서 달성된 처리량을 표시한 것이다. 대역 폭 효율은 망 지연 및 패킷 손실로 인해 저하되지 않는다.

가용 대역폭에 대한 효율적 활용에 더하여, *fasp* 적응적 전송률 제어의 지연-기반 특성은 애플리케이션들로 하여금 전송 서비스에서의 의도적인 우선 순위를 구축할 수 있게 한다. 망 큐잉에 대한 내장된 응답 체제는 각 개별 전송들이 동시 병행의 *fasp* 전송에 차별화된 대역 폭 우선 순위 제공과 같은, 애플리케이션 목표를 이를 수 있게 하는 우선순위화/우선순위해제가 될 수 있게 하는 가상적 "핸들"임을 증명한다.

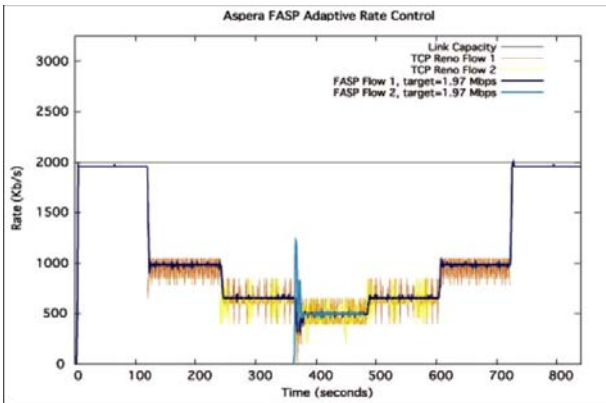


그림 10. *FASP* 는 타 *FASP* 및 표준 TCP 트래픽과 링크 처리량을 공유한다. 이는 프로토콜 내 및 프로토콜 간의 공정성을 달성하고 있다.

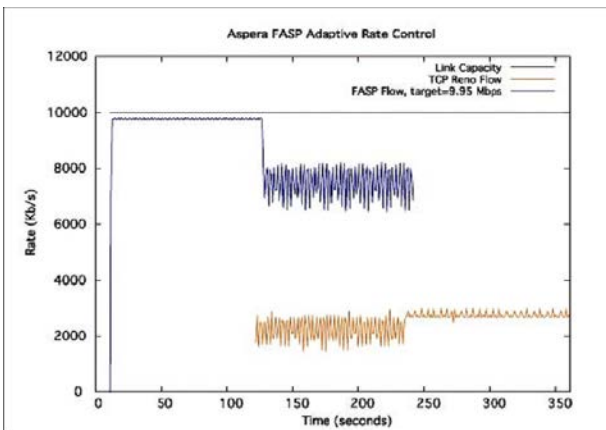


그림 11. *FASP* 은 망 조건에 의해 TCP가 제한적일 때에도 허용 대역폭을 사용하는데, 이는 *FASP* 흐름과 다른(TCP) 트래픽 간의 완전한 공정성을 달성하고 있다.

망 전송에서 인위적인 병목현상을 제거하고 최종 사용자에게 전체 대역 폭을 풀어줌으로써, *fasp* 전송은 때로 디스크 입출력, 파일 시스템, 그리고 CPU

스케줄링 등에서 새롭게 나타나는 병목 현상 포인트를 드러내는데, 이는 필연적으로 특히 수 기가비트 망에서 전체 망 속도로 전송률이 가해질 때 새로운 장애물을 만들어내게 된다. *fasp* 적응적 전송률 제어는 상대적으로 느린 저장 장치 경로로 고속 파일 전송 기록에서 유실 데이터를 피하기 위해 디스크 흐름 제어를 포함하도록 확장되어 왔다. 유사한 지연 기반 모델(특히 출원중)이 디스크 버퍼를 위해 개발되었다. 네트워크와 디스크의 상이한 시간 척도로 인해, 이중 시간 척도 설계가 대역 폭과 디스크 속도 변화를 수용하기 위해 도입되었다. 미시 단위, 즉 빠른 시간 척도, 에서는 로컬 피드백 체제가 예를 들어 운영체제 스케줄링으로 인한 주기적 디스크 속도 저하를 감당하기 위해 수신단에 도입되며, 반면에 거시 단위, 즉 느린 시간 척도, 에서는 통합 지연 기반 혼잡 회피가 대역 폭 제어와 디스크 제어 모두를 위해 구현되어, 이는 *fasp* 전송이 허용 대역 폭 뿐만 아니라 디스크 속도에도 동시에 적용될 수 있게 한다.

파일 시스템 병목 현상은 다양한 측면에서 나타난다. 실제로, 대부분의 고객들은 동일한 크기의 파일 하나를 전송하는 것과 비교해서 여러 개의 작은 파일을 전송할 때 속도가 급격하게 떨어지는 경험을 합니다. 새로운 파일 스트리밍 기술을 사용함으로써, *fasp* 은 파일 시스템에 의해 발생하는 인위적인 병목 현상을 제거하고 대량의 작은 파일들의 전송에 대해 동일한 이상적인 효율성을 달성합니다. 예를 들어, 1,000개의 2MB 파일들이 미국에서 뉴질랜드까지 OC-3(Optical Carrier Level 3) 전체를 채울수 있는 155Mbps의 유효 전송 속도로 전송될 수 있다.

그 결과로, *fasp* 은 FTP와 UDT와 같은 TCP 혹은 UDP 기반의 파일 전송 기술의 기본적인 병목 현상을 제거하며, 공중 및 사설 IP 망 상의 전송을 매우 빠르게 한다. *fasp* 은 불안정한 혼잡 제어 알고리즘과 패킷 손실(물리적인 미디어, 트래픽 간 버스트, 혹은 정교하지 못한 프로토콜 자체에 의해 생기는), 그리고

신뢰성과 혼잡 제어와의 연동에 의해 발생하는 인위적인 병목 현상을 제거한다. 이외에, *fasp* 혁신은 디스크 입출력, 파일 시스템, CPU 스케줄링, 등에서 생기는 최근의 병목 현상도 제거하고 가장 길고, 가장 빠른 광대역 망상에서도 전체 망 속도를 달성한다. 우리는 그 결론이 상용 망 상에서 대량의 파일 기반의 데이터의 전송을 위한 TCP에 의해 생기고 점점 커지는 틈을 채워주고, 그래서 전세계의 대량의 디지털 데이터의 일상적인 이동을 가능하게 하는 차세대 초고속 전송 프로토콜이라고 믿는다.