
자기유사성 트래픽 조건에서 다중 시간 간격을 이용한 혼잡제어

박동석* . 나상동*

Congestion Control with Multiple Time Scale under Self-Similar Traffic

Dong-Suk Park . Sang-Dong Ra
computer Engineering, Chosun University

이 논문은 2004년도 조선대학교 학술연구비의 지원을 받아 연구되었음.

요 약

컴퓨터 네트워크에서 전송제어 프로토콜 상에서 다중시간 간격으로 혼잡제어의 지연을 피드백 루프의 시간간격을 조정하여 명시적 예측을 실행해 제어의 단점을 개선하는데 연구한다. 자기 사성은 다양한 네트워크 환경에서 공통적으로 발생하는 현상인 것으로 정체 제어에서 자기 유사성 트래픽의 장거리 상관 구조를 활용할 가능성과 MTS 정체 제어의 프레임워크를 발전시켜, 이 프레임워크가 전송을 기반 피드백 제어의 성능을 향상시키는 데 시뮬레이션을 통해 트래픽 조건으로 나타낸다. 자기 유사성 트래픽 조건에서 3가지 단계로 구성한다. 먼저 TCP의 모듈 형 확장에서 Tahoe, Reno, Vegas 등 다양한 버전의 TCP에 적용되는 간단한 인터페이스를 통한 함수 호출을 정의하고, 이것이 성능을 크게 향상시킨다. 두 번째로, 광대역 WAN에서 지연-대역폭의 곱이 높을 경우 더욱 심각해지는 사후 제어의 불확실성 차이를 해소함으로써 MTS TCP가 기반 피드백 제어에 사전성을 부여한다. 세 번째는 트래픽 제어의 3가지 차원인, 즉 추적 능력, 연결 지속 기간, 공정성이 성능에 미치는 영향 등을 비교 분석하여 입증한다.

ABSTRACT

Measurement of network traffic have shown that the self-similarity is a ubiquitous phenomenon spanning across diverse network environments. In previous work, we have explored the feasibility of exploiting the long-range correlation structure in a self-similar traffic for the congestion control. We have advanced the framework of the multiple time scale congestion control and showed its effectiveness at enhancing performance for the rate-based feedback control.

Our contribution is threefold. First, we define a modular extension of the TCP-a function called with a simple interface-that applies to various flavours of the TCP-e.g., Tahoe, Reno, Vegas and show that it significantly improves performance. Second, we show that a multiple time scale TCP endows the underlying feedback control with proactivity by bridging the uncertainty gap associated with reactive controls which is exacerbated by the high delay-bandwidth product in broadband wide area networks. Third, we investigate the influence of the three traffic control dimensions-tracking ability, connection duration, and fairness-on performance.

Keywords

TCP-IP, ATM, MTS 트래픽 혼잡제어, 통신망, 컴퓨터 네트워크

1. 서론

컴퓨터 네트워크에서 사용되는 메시지 교환에 관한 규칙을 정의 하는 프로토콜들의 모음을 의미 하는 전송제어 프로토콜(TCP: Transmission Control Protocol)이 트랜스포트 계층 프로토콜에서 가장 광범위하게 사용한다. TCP의 가장 중요한 성분은 혼잡 제어와 복구를 수행하기 위해 사용된 알고리즘으로서 지속적으로 확장[1,2] 연구되어 왔다.

본 논문에서는 트래픽을 네트워크 경로를 통해 동일한 종단 노드 쌍 사이의 반대 방향에서 데이터를 전송하는 두 개 또는 그 이상의 TCP 연결로부터 생긴 트래픽 패턴을 사용하고 네트워크의 교환기와 라우터들을 공유한다. 승인 압축의 결과는 연결을 경쟁하면서 수신된 처리율에서 불공평을 표시하고 이러한 결과가 없는 예측과 비교해서 전체적인 처리율을 감소시킨다. 처리율에서 불공정성과 전체적인 감소를 야기 시키는 양방향 트래픽하에서 OSI을 기반으로 한 네트워크에서 결과[3]를 볼 수 있다.

TCP가 비율 제어된 채널을 통해 운용될 때 비동기 전송 모드 네트워크에서 가변 비트 율에 의해 네트워크 노드에서 볼 수 있는 트래픽의 버스트성은 비율 제어가 없는 채널을 통해 운용되는 것과 비교하고 네트워크 노드에 의해 도입된 양방향 TCP 트래픽의 효율성은 비율 제어된 네트워크 환경에서 중요하다. 그러나 연결 사이의 바람직하지 않은 상호 동작은 종단 시스템에서 발생하고 교환기나 라우터에서 승인 압축과 같은 동작을 이끌어낸다. TCP 승인 압축의 효과와 처리율 손실 결과는 세그먼트와 승인이 종단 시스템의 공통 큐를 공유함으로써 지속되고 TCP 연결의 데이터 세그먼트가 반대의 연결에 대한 승인을 가지고 있는 가상 채널을 공유하는 계층에 있다.

자기유사성 트래픽을 제어한다는 것은 트래픽 흐름의 관리 문제를 의미하며, 네트워크 성능을 최적화하기 위해 자기유사성(self-similarity) 및 LRD와 관련된 속성을 활용하게 될 것이다. 이는 트래픽 제어 용도로 활용 가능하고, 현재 어떤 트래픽 제어 알고리즘에서도 다루지 못하는 정보이다. 버퍼 용량 및 프로토콜 스택 또는 네트워크 구성의 세부 사항과 상관없이 연결 지속 기간 또는 전송되는 객체 크기가 대형이라면, 집합 트래픽의 LRD 및 자기유사성은 네트워크의 다중화 지점에서 존재함을 입증한다.

혼잡제어를 위한 자기유사성 네트워크 트래픽의 장거리 상관 구조의 활용[5]은 다중시간간격 혼잡 제어 프레임워크를 전송을 기반 피드백 제어 측면

에서 성능을 향상시키는 데 효과적이다. 긴 시간간격에서 일반 전송을 반 피드백 혼잡제어에 상관 구조를 포함시킴으로써, 성능을 크게 향상시킬 수 있다. 적응식 이중화 제어를 사용하여 실시간 멀티미디어 트래픽, 백그라운드 및 소스 트래픽 모두에 LTS 상관 구조를 활용함으로써 종단간 QoS가 크게 향상 된다. 실시간 트래픽 제어 프레임워크는 MTS(Multiple Time Scale) 이중화 제어이며 종단간 QoS 제어를 위한 패킷 레벨의 적응적 순방향 오류 정정에 대한 초기 작업이다

본 논문에서는 MTS 트래픽 제어 프레임워크를 TCP 기반의 신뢰할 수 있는 전송 및 윈도우 기반 혼잡제어로 확대 하고 TCP의 대역폭 소비 반응의 적극성을 LTS(large time scale) 네트워크 합수 형태, 즉 RTT(round-trip time)가 결정된 피드백 루프의 한계를 넘어서는 정보의 형태로 조정하는 LTS 모듈과 TCP를 연계시키는 방법으로 수행된다. TCP-MT는 피드백 루프 범위에 속하지 않는 LTS 정보를 활용한다. 자기유사성 버스트 현상, 즉 1차적인 성능 효과가 과 활용 미 활용 집중 기간의 형태로 2차적인 성능 효과를 유발하는 현상이 혼잡제어 연구의 대표적인 주제였던 피드백 트래픽 제어의 난제를 해결 한다. TCP-MT는 연결 지속 기간이 길 때 가장 큰 성능 이득을 제공되며 TCP-MT의 대역폭 공유 반응이 TCP의 반응과 유사하고, 잘 알려진 TCP의 공정성 속성에서 긍정적, 부정적 영향을 전혀 미치지 않는다는 것을 입증한다.

II. 트래픽 구조에서 TCP 연결 제어

2.1 TCP 비율 제어

종단 노드들은 전형적인 점대점 링크에 의해 서로 연결되고 각각의 노드는 가상채널을 포함하고 있다.

그림 2-1와 같이 노드 j 가 세그먼트 4'와 5'의 전송을 완료하고 반면에 세그먼트 6'과 7'이 노드 j 내에 큐잉 될 때 시간에 대한 네트워크 상태에서 연결 i 에 대한 세그먼트 12,13,14에 승인은 IP 큐속에 누적되고 $t+13$ 시간에 집단적으로 전송된다. 노드 i 가 세그먼트 4'와 7'사이에 어떠한 승인도 수신하지 못하므로 세그먼트 4'부터 7'에 대한 승인은 연결 i 의 세그먼트 사이에 끼이지 않고 노드 j 로 되돌아간다. 순차적으로 응답에 대한 노드 j 에서 발생된 새로운 4개의 세그먼트는 연결 i 의 어떠한 승인도 포함하지 않으므로 손실 없는 네트워크에서는 슬로우 스타트 단계 동안에 설정된 집단적인 효과가 연결의 전체적인 수명을 유지한다.

슬로우 스타트 윈도우 증가 단계에서 일어나는 승인 압축의 발생이 혼잡회피 단계에서 일어나는 윈도우 증가와 비슷한 결과를 가지고 있으므로 승인 압축의 결과에 대한 두 윈도우 증가 단계에서의 차이는 혼잡회피 단계와 비교하여 슬로우 스타트 단계에서 일괄적인 승인 압축의 크기가 좀 더 빠르게 증가한다는 것이다. 노드 j로부터 수신된 각각의 세그먼트에 대한 승인을 전송하는 것을 제외하고 주기 시간에 노드 j로부터 다음 승인 수신을 기다린다. 이것은 양방향 TCP 트래픽을 이용하여 처리율 손실을 발생시키는 근본적인 이유로 양방향 트래픽 배치 구조에서 처리율의 지속적인 손실이 발생한다.

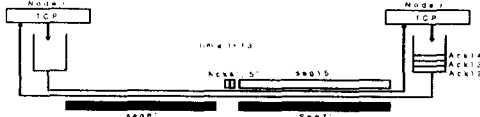


그림 2-1. 세그먼트에 인의 처리 과정
Fig. 2-1 Acknowledgement process for segments

종단 노드 i에서 j까지의 전송된 TCP 세그먼트에 의해 발생하는 지연을 D_{ij} 로 하고, 노드 j에서 i까지의 세그먼트에 의해 발생한 지연은 D_{ji} 로 한다. 효율적인 분석을 위해 전송된 TCP 세그먼트의 크기는 일정하다고 가정하며, 대칭적인 네트워크 배치 구조에서 각각의 방향에서 종단 노드의 전송율은 동일한 것으로 간주하며 전송율은 초당 네트워크 전송률을 통해 전송된 TCP의 세그먼트의 수로 정의하며 ρ 로 나타내고 승인의 전송 시간은 데이터 세그먼트의 전송 시간보다 상당히 적으므로 분석을 하기 위해서 승인의 전송시간을 0으로 설정한다. 양방향 연결의 윈도우 크기는 안정적이고 각각 노드 i와 j에서 발생하는 연결에 대한 윈도우 크기를 W_i 와 W_j 의 세그먼트로 나타낸다. 노드 i부터 j까지의 한쪽방향 링크를 채우기 위해 필요한 세그먼트의 수를 L_{ij} 로 나타내고 반대편 링크에서는 L_{ji} 로 하고 L_{ij} 는 ρ 와 D_{ij} 의 곱으로 L_{ji} 는 ρ 와 D_{ji} 의 곱으로 된다. $Q_i(t)$ 는 데이터 세그먼트만을 고려하고 승인에 의해 점유되는 공간을 무시한 채로 시간 t에서 노드 i 출력 IP 큐의 점유기간이 비슷한 방법으로 $Q_j(t)$ 는 노드 j에 대한 점유 기간이다. $\tau_{i,k}$ 는 노드 j에 도착한 연결 i의 k번째 번잡 기간 동안에 노드 i에 의해 전송된 첫 번째 세그먼트 시간이다. 마찬가지로 $\tau_{j,k}$ 는 노드 i에 도착한 연결 j의 k번째 번잡 기간 동안에 노드 j에 의해 전송된 첫 번째 세그먼트의 시간이다. 처리율은 라운드 전송 파이프의 대역폭-지연 곱을 초과하는 윈도우 크기의 합이며,

노드 i의 윈도우 크기가 노드 j의 윈도우 크기와 노드 i에서 노드 j로 링크를 채우기 위해 필요한 세그먼트의 수 L_{ij} 와 노드 j에서 노드 i로 링크를 채우기 위해 필요한 세그먼트의 수 L_{ji} 의 합보다 클 때 식(2.1)과 같다.

$$W_i > W_j + (L_{ij} + L_{ji}) \quad (2.1)$$

노드 j에서 연결 i의 승인의 최악의 경우 큐잉 지연은 W_j 이므로 노드 i는 노드 j의 윈도우를 다 비우지 못하게 되며, W_i 의 세그먼트로 구성된 연결 i의 각각의 번잡 주기는 집단화된 W_j 승인의 순서와 함께 동반하여 다음 번잡 주기에 의해 즉시 뒤따라온다.

연결의 번잡 주기는 주기적인 동작이 존재한 상태에서 윈도우 크기 W_i 와는 별도로 분리되어서 공간화 된다. 링크 용량에 비례하는 연결의 처리율을 연결 효율성이라고 하며, 각각의 연결은 W_i 세그먼트 전송시간 동안에 윈도우 세그먼트를 전송하므로 i노드와 j노드에서 연결의 효율성은 식(2.2)와 같다.

$$F_i = 1, F_j = W_j / W_i \quad (2.2)$$

연결의 k번째 번잡 주기의 첫 번째 세그먼트가 노드 j에 도착할 때 노드 j의 IP 큐의 점유 기간은 $Q_j(\tau_{i,k})$ 이므로 번잡 주기의 첫 번째 세그먼트에 대한 승인은 $\tau_{i,k} + Q_j(\tau_{i,k}) / \rho$ 시간에 노드 j에서 출발하고 시간 t1에서 노드 i로 도착하고, t1은 식(2.3)과 같다.

$$t_1 = \tau_{i,k} + Q_j(\tau_{i,k}) / \rho + D_{ji} \quad (2.3)$$

시간 t1에서 노드 i의 출력 IP 큐의 점유기간은 먼저 $\tau_{i,k} - D_{ij}$ 시간에 노드 i의 k번째 번잡 주기 전송을 시작함으로써 이루어지고 번잡 주기에 대한 첫 번째 승인 시간이 지난 후에 노드 i로 되돌아오고, 노드 i는 식(2.4)와 같은 세그먼트로 전송된다.

$$\rho(t_1 - (\tau_{i,k} - D_{ij})) = Q_i(\tau_{i,k}) + (L_{ij} + L_{ji}) \quad (2.4)$$

노드 i의 각각의 번잡 주기는 W_i 세그먼트로 구성되고 노드 i의 출력 큐는 k번째 번잡 주기의 첫 번째 승인이 t1 시간에 되돌아 올 때 식(2.5)와 같은 세그먼트를 갖는다.

$$Q_i(t_1) = W_i - (Q_j(\tau_{i,k}) + L_{ij} + L_{ji}) \quad (2.5)$$

$Q_j(\tau_{i,k}) \leq W_j$ 와 $W_i > W_j + (L_{ij} + L_{ji})$ 임으로 $Q_i(t_1)$ 은 0보다 크다. 그러므로 연결 j 의 모든 승인은 버스트로서 노드 j 로 되돌아가고 연결 i 는 전송을 위해서 출력 큐에서 이용 가능한 데이터를 항상 가지고 있다. 다음 번잡 주기는 $t_1 + Q_i(t_1) / \rho$ 시간에 노드 i 에서 시작하고 $t_1 + Q_i(t_1) / \rho + D_{ij}$ 시간에 노드 j 로 도착한다. 그러므로 $\tau_{i,k+1} = t_1 + Q_i(t_1) / \rho + D_{ij}$ 과 같다. 식(2.3)과 식(2.5)로부터 t_1 과 $Q_i(t_1)$ 을 각각 대입하면 식(2.6)과 같이 된다.

$$\tau_{j,k+1} = \tau_{i,k} + W_j / \rho \quad (2.6)$$

유사한 방법으로 $\tau_{j,k+1}$ 은 식(2.7)과 같이 구할 수 있다.

$$\tau_{j,k+1} = \tau_{j,k} + \frac{W}{\rho} \quad (2.7)$$

식(2.6)은 식(2.1)과 같이 $W_i > W_j + (L_{ij} + L_{ji})$ 인 경우에 적용할 수 있으며 $W_i < W_j - (L_{ij} + L_{ji})$ 인 경우에 대해서는 보완적이고 동일한 분석으로 취급한다. $\tau_{j,m}$ 은 연결 j 의 m 번째 번잡 주기에서 첫 번째 세그먼트가 노드 i 에 도착한 시간이며, $\tau_{i,k}$ 로부터 $\tau_{j,m}$ 을 다음과 같이 결정한다. $\tau_{i,k}$ 시간에 노드 j 는 m 번째 번잡 주기의 $W_i - Q_j(\tau_{i,k})$ 를 전송한다. 그러므로 $\tau_{i,k} - (W_i - Q_j(\tau_{i,k})) / \rho$ 시간에 m 번째 번잡 주기의 전송을 시작하고 이러한 번잡 주기의 첫 번째 세그먼트는 D_{ji} 의 지연 후에 노드 i 에 도착한다. t_1 을 노드 i 가 k 번째 번잡 주기의 전송을 완료했을 때 시간은 식(2.9)과 같다.

$$t_1 = \tau_{i,k} - D_{ij} + \frac{W}{\rho} \quad (2.8)$$

$(\tau_{j,m}, t_1)$ 간격 동안에 노드 i 는 연결 j 의 m 번째 번잡 주기로부터 세그먼트를 수신하나 노드 i 의 출력 큐는 비어있지 않은 상태로 남아 있으며, 노드 i 에서 노드 j 쪽에 의해 발생된 승인은 집단화된다. 이 번잡 주기의 첫 번째 세그먼트는 $\tau_{i,k+1} = \tau_{i,k} + W_j / \rho$ 시간에 노드 j 에 도착한다. $k+1$ 번째 번잡 주기가 연결 j 의 세그먼트 W_j 에 대해 압축된 승인 다음에 옴으로서 이러한 번잡 주기의 첫 번째 패킷은 노드 j 에서 $Q_j(\tau_{i,k+1}) = W_j$ 세그먼트 큐 크기 임을 알 수 있고 동일한 결과는 $Q_j(\tau_{i,k+1}) = \min((W_i + W_j) - (L_{ij} + L_{ji}) - Q_i(\tau_{i,k}), W_j)$ 와 $W_i > (L_{ij} + L_{ji}) + Q_j(\tau_{i,k})$ 를 조합함으로써 얻을 수 있다.

그러므로 $k+1$ 번째 번잡 주기에 대한 첫 번째 응답은 $\tau_{i,k+1} + W_j / \rho$ 시간에 출발하고 $t_2 = \tau_{i,k+1} + W_j / \rho + D_{ji}$ 시간에 노드 j 에 도착한다. $W_i \leq W_j + (L_{ij} + L_{ji})$ 임으로 $t_2 \geq t_1$ 임을 쉽게 계산할 수 있고, 노드 i 가 연결 i 의 $k+1$ 번째 번잡 주기의 전송을 완료한다. 그러므로 $k+2$ 번째 번잡 주기는 t_2 시간에 시작되고, D_{ij} 지연 후에 노드 j 에 도착한다. 그러므로 $\tau_{i,k+2}$ 는 식(2.9)와 같은 결론이 될 것이다.

$$\begin{aligned} \tau_{i,k+2} &= t_2 + D_{ij} \\ &= \tau_{i,k+1} + W_j / \rho + D_{ji} + D_{ij} \\ &= \tau_{i,k} + (W_i + W_j) / \rho + (D_{ij} + D_{ji}) \end{aligned} \quad (2.9)$$

$Q_i(t_1) = 0$ 인 경우로서 이 경우는 $W_i \leq W_j + (L_{ij} + L_{ji})$ 일 때만 발생한다. 이 경우에서 연결 i 의 $k+1$ 번째 번잡 주기는 t_1 시간에 노드 i 로 출발하고, 그것의 첫 번째 세그먼트는 $t_1 + D_{ij}$ 시간에 노드 j 에 도착한다. 그러므로 $\tau_{i,k+1}$ 은 식(2.10)과 같다.

$$\tau_{i,k+1} = t_1 + D_{ij} = \tau_{i,k} + \frac{Q_j(\tau_{i,k})}{\rho} + (D_{ij} + D_{ji}) \quad (2.10)$$

이러한 번잡 주기에 대한 첫 번째 응답은 시간 $\tau_{i,k+1} + Q_i(\tau_{i,k+1}) / \rho$ 시간에 노드를 출발하고 식(2.11)와 같은 시간에 노드 i 에 도착한다.

$$t_3 = \tau_{i,k+1} + \frac{Q_i(\tau_{i,k+1})}{\rho} + D_{ji} \quad (2.11)$$

노드 i 는 t_3 시간에 $k+1$ 번째 번잡 주기의 전체를 전송함으로써 t_3 에서 $k+2$ 번째 번잡 주기가 시작된다. 이러한 번잡 주기의 첫 번째 세그먼트는 $t_3 + D_{ij}$ 시간에 노드 j 에 도착한다. 따라서 $\tau_{i,k+2}$ 는 식(2.12)과 같다.

$$\begin{aligned} \tau_{i,k+2} &= t_3 + D_{ij} \\ &= \tau_{i,k} + \frac{W_i + W_j}{\rho} + (D_{ij} + D_{ji}) \end{aligned} \quad (2.12)$$

$W_j - (L_{ij} + L_{ji}) \leq W_i \leq W_j + (L_{ij} + L_{ji})$ 인 경우에 대해 TCP 연결의 주기적인 동작이다. 윈도우 크기는 $W_i = W_j = 4$ 개의 세그먼트로 가정하고 링크상에서 각각의 세그먼트 시간은 1초로 한다. 각기 방향에서 네트워크의 지연은 2초로서 L_{ij} 와 L_{ji} 는 2초이다. 그러므로 각기 연결에 허용하는 네 개의 세그먼트는 한쪽 방향 배치 구조에서 노드의 최대 처리율을 달성할 수 있다.

III. 트래픽에서 시간 간격 제어

3.1 피드백 트래픽 혼잡제어

피드백 혼잡제어 개선에서 $n > 1$ 보다 크다 해도 LTS 모듈이 비활성 화되면, 혼잡제어는 원래의 피드백 혼잡 제어 수준으로 떨어지게 된다.

피드백 혼잡제어의 트래픽 제어 반응을 조절하는 데 LTS 상관 구조를 두 가지로 구분하여 적용시킨다. 첫 번째 방법은 선택적 기울기제어(SSC; selective slope control)로서, 그림3-1와 같이 예측된 LTS 네트워크 상태를 선형 증가/지수 감소 혼잡제어의 선형 증가 단계에서 선형 증가 곡선의 기울기를 조정하는 것이다. 만약 네트워크 경쟁이 낮은 편이라면 기울기가 증가하고, 네트워크 경쟁이 높으면 그 반대가 된다.

SSC는 지속적인 네트워크 상태를 통해 기울기를 변경함으로써, 불안정성을 유발하지 않고서도 피드백 혼잡제어의 대역폭을 점유하는 적극적인 동작을 조절할 수 있다. RTT 또는 피드백 루프를 차수 크기 이상으로 초과하는 LTS에서도 기울기는 안정적으로 유지된다.

피드백 트래픽 제어에서 LTS 상관 구조를 활용하는 선택적 레벨 제어(SLC; selective level control)방법으로서 LTS 네트워크 상태 함수 형태로 출력 전송율을 부가적으로 조정한다. 즉 네트워크 경쟁이 낮은 편이면 'DC' 레벨을 높이고, 경쟁이 높아지면 그 반대로 한다. 이 방법은 그림 3-2와 같이 SLC는 혼잡 제어를 위한 방식으로 재전송이 불가능할 때 적응 패킷 레벨의 순방향 오류 정정이 적용된다. 실시간 멀티미디어 트래픽 제어에서 MPEG1 프레임의 도착 및 디코딩을 지원하기 위한 SLC를 사용한다.

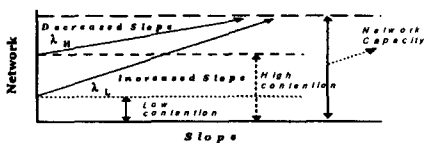


그림 3-1. 고 경합 및 저 경합의 선형증가에서의 선택적 기울기 조정

Fig. 3-1 Selective slope adjustment during linear increase phase for high and low contention.

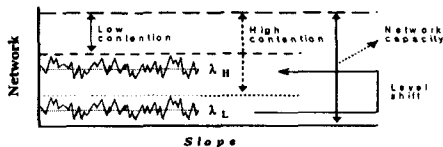


그림3-2. 고경합 및 저경합간의 선택적 'DC' 레벨조정
Fig. 3-2 Selective 'DC' level adjustment between high and low contention.

3.2 다중시간에서 예측스케줄

TCP-MT의 구조는 TCP의 특별한 특징인 피드백 제어 및 선택적 기울기 제어(SSC)를 구현하는 LTS 모듈로 이루어진다. LTS 모듈은 다시 LTS 상관 구조를 온라인상에서 추출되는 명시적 예측 모듈, TCP로 전달되는 기울기의 최종 단계를 결정하는 적극성 스케줄이 사용된 기울기 값의 범위를 조정하는 메타컨트롤로 구성된다.

SSC는 기반 피드백 제어에서 측정 가능한 입출력 동작은 전송된 TCP 세그먼트 개수, 수신 ACK의 개수 등을 기반으로 연산을 수행 한다. LTS 모듈은 수신자 쪽에서만 증가시킬 수 있으며, 발신자 쪽은 변함없다. TCP-MT의 전체적인 구조는 그림 3-3과 같이 SSC 모듈의 LTS T_L 에서 추출된 정보를 사용하여 피드백 루프 $T_L \gg T_S$ 의 시간 규모 T_S 에서의 TCP의 대역폭 점유 동작을 조절한다.

명시적 예측 모듈은 조건부 확률 밀도 $\Pr L_2 | L_1 = \ell, \ell \in \{1, h\}$ 를 온라인 측정하는 Bayesian 측정 방식을 사용한다.

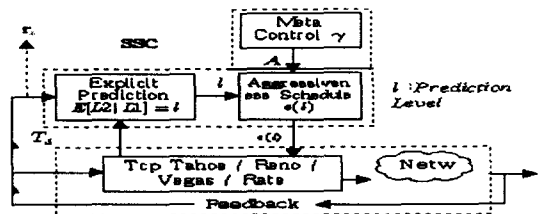


그림 3-3. TCP-MT의 구조

Fig. 3-3 Structure of TCP-MT

온라인 측정은 업데이트 간격 SSC의 시간 규모 T_L 마다 $O(1)$ 연산을 사용하는 방식으로 수행될 수 있다. 발신자 쪽에서는 SSC의 명시적 예측 모듈이 크기가 $h \times (h + 1)$ 인 2차원 배열 $CondProb[\cdot][\cdot]$ 를 관리하며, 각 $\ell \in [1, h]$ 마다 행이 하나씩 지정된다. 각 $\ell' \in [1, h]$ 마다 $CondProb[\ell][\ell']$ 은 개수 h_{ℓ} 을 유지하며 식 (3.1)과 같다.

$$\Pr L_2 = \ell' | L_1 = \ell = h_{\ell'} / h_{\ell} \quad (3.1)$$

식(3.1)에서 테이블 $CondProb$ 는 알고 있는 조건부 확률 밀도와 같다. T_L 에서 측정된 트래픽 값 $a > 1$ 일 때, 조건부 기대값 $\ell = E[L_2 | L_1 = x]$ 을 계산할 수 있고 이를 적극성 스케줄 인덱스에 사용한다.

적극성 스케줄 모듈에 대한 선택적 기울기 조정의 응용은 보편적으로 선택한 TCP 버전에 대해서

$a=1$ 이지만 TCP에서 혼잡 윈도우 크기의 변화는 적극성 상수 $a > 0$ 에 의해 매개변수로 표시되고, a 는 적극성 변수 ξ 로 대체되어 변수화 된다. $\xi' > \xi$ 라면, TCP(ξ')은 TCP(ξ)보다 더 적극적이다. LTS 모듈을 TCP와 결합하는 작업은 $\xi = \epsilon(\mathcal{L}_2)$ 로 설정함으로써 완료된다. \mathcal{L}_2 는 시간간격 T_L 에서의 예측 경쟁 레벨을 의미하며, 명시적 예측 모듈로 계산한 것이다. $\epsilon(\cdot)$ 는 적극성 스케줄이라 부르며 $\mathcal{L}_2 = E[L_2|L_1]$ 의 감소함수로 표시되며, 역 선형 스케줄은 식 (3.2)와 같다.

$$\epsilon(\mathcal{L}_2) = \frac{A-a}{h-1}(h-\mathcal{L}_2) + a, \mathcal{L}_2 \in [1, h] \quad (3.2)$$

식(3.2)에서 A 는 적극성 최대 레벨을 나타내며, $\mathcal{L}_2 = 1$ 이면 가장 큰 기율기이다. 따라서 가장 적극적인 활동에 대해 영향을 미치고, 반면 $\mathcal{L}_2 = h$ 는 가장 비적극적인 활동으로서 기본 기율기 $\xi = a$ 로 축소된다. 메타컨트롤은 최대 기율기 레벨 A 를 설정할 책임이 있고, A 는 역 선형 스케줄에서 나머지 값을 결정하게 되고 적극성 스케줄의 조건은 식 (3.3)과 같다.

$$l \leq l' \rightarrow \epsilon(l) \geq \epsilon(l') \quad (3.3)$$

여기서 각 값 $\epsilon(l)$ 은 별도로 계산되며, 다른 $\epsilon(\cdot)$ 값들과는 독립적으로 메타컨트롤에 의해 계산된다.

3.2.1 TCP 피드백 메타컨트롤

메타컨트롤에서 나타나는 최대 적극성 매개변수 A 는 고정된 순위 값으로 설정되며 보통 네트워크 상태의 함수로서 동적으로 조정된다. 정적인 네트워크 환경에서는 A 가 잘 정의되며, 정적과 비슷한 조건에서 A^* 로 수렴하는 대칭적 제어 규칙은 식 (3.5)와 같다.

$$\frac{dA}{dt} = \begin{cases} \nu, & dy_\ell/dA_\ell > 0, \ell \in [1, h] \\ -\nu, & dy_\ell/dA_\ell < 0, \ell \in [1, h] \end{cases} \quad (3.5)$$

식 (3.5)에서 $\nu > 0$ 은 조정 인수이다. 제어 동작은 현재 측정된 경쟁 레벨 $L_1 = l \in [1, h]$ 의 조건에 포함되며, dy_ℓ/dA_ℓ 은 동일한 레벨 $\ell, \ell \in [1, h]$ 로 분류되는 최근 시간 블록에서 계산할 수 있다. 네트워크 시스템이 단일모드 로드-처리율 곡선을 갖는다는 차원에서 혼잡에 민감한 경우, 안정성을 보장하기

위해서는 비대칭성이 필요하다. 그렇지 않은 경우 충분히 $\nu > 0$ 을 작게 함으로서 점차적으로 안정성을 실현하는 데 충분하다. SSC의 메타컨트롤을 관리하는 제어 규칙과 TCP 피드백 제어가 결합한 멀티레벨 피드백 제어 시스템이 안정 상태를 유지하는 이유는 T_L 과 T_S 간에 큰 시차가 존재하기 때문이다. TCP 혼잡제어가 활성화되는 동안 A 는 시간 T_L 의 간격에 대해 일정한 값을 유지함으로써 선형 증가/지수 감소 제어 및 $T_S \ll T_L$ 의 안정성에 의해 각 T_L 간격마다 안정성이 유지된다. 기율기 레벨 매개변수 A 는 시작할 때 기본적으로 $A(0) = a$ 로 설정된다. A 는 항상 양수 값을 가지며 a 보다 크므로 A 는 시간이 지나면서 증가하므로 $[a, A]$ 범위의 값을 갖는 역 선형 스케줄에 따라 개별적인 적극성 레벨이 설정된다.

3.3 TCP에서 버전과 대역폭

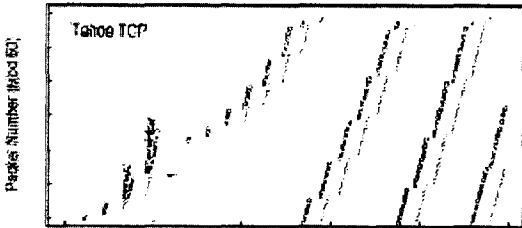
여러 가지 종류의 TCP 버전에서 전송을 기반 확장에 SSC를 결합하는 다양한 형태를 고려하며, 또한 광범위하게 적용하기 위해 ATM 전송을 기반 혼잡제어에 결합을 보였다. 그림 3-4, 3-5, 은 TCP 각 버전에 대한 패킷의 흐름 및 UDP에 따른 대역폭의 변화를 실험에 의해 나타내었다.

TCP Reno와 MTS 결합은 서로 다른 2가지 형태, 즉 혼잡 회피 구성 요소와 Slow-Start 구성 요소로 이루어지며, Slow-Start가 추가적인 최적화로 사용된다. 간단한 확장을 통해 이러한 동일한 결합을 TCP Tahoe에서도 적용될 수 있다. TCP Reno의 혼잡회피 단계에서는 적극성 상수 a 를 혼잡 윈도우 변경의 기율기라고 하면 즉 $a=1$ 일 때 $cwnd \leftarrow cwnd + (a/cwnd)$ 로 된다. 이러한 결합은 a 를 $\epsilon(\mathcal{L}_2)$ 으로 대체하고 선형 증가 단계의 기율기에 영향을 미침으로써 전반적인 네트워크 상태에 도움이 된다. Slow-Start : 타임아웃이 발생할 때마다 윈도우 크기 $cwnd$ 와 현재의 트래픽 레벨 L_1 을 연관시킨다. 즉 $cwnd = cwnd(L_1)$ 이 된다.

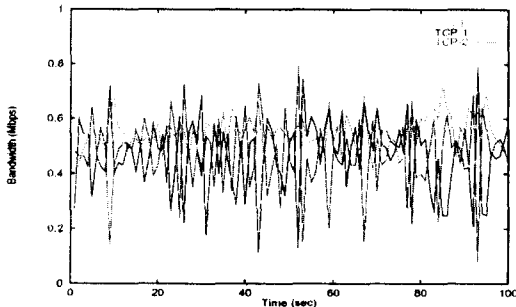
경험을 바탕으로 Slow-Start 임계값을 $ssthresh \leftarrow cwnd(\mathcal{L}_2)$ 로 설정하며 $cwnd$ 을 위해 예상 트래픽 레벨 \mathcal{L}_2 가 인덱스 역할을 한다. Reno의 Fast Recovery 메커니즘에 대해서도 비슷한 방식의 결합을 구현함으로써 Tahoe와 달리 보다 강력한 최적화가 가능하다. TCP Vegas는 사전에 지정된 목표 한계 $\alpha < Diff/\beta$ 에서 실제 전송율과 예상 전송율 간에 측정된 차이 $Diff$ 를 유지하면서 네트워크에 추가적인 데이터를 유지하려고 한다. 여기에는 타임아웃 및 재전송을 줄임으로써 보다 지속적인 호

를 보장하는 사전 준비성의 측정도 포함된다. TCP Vegas와의 결합은 수정된 혼잡 회피 메커니즘을 통해 구현되며 $Diff\alpha$ 일 때 선형 증가 기울기를 조절한다. 따라서 이벤트 트리거의 경우를 제외하고는 결합 형태는 Reno 및 Tahoe와 동일하다.

TCP Rate는 TCP Reno를 전송율 기반을 확장한 것으로 표 1에서와 같이 지연 변이를 기준으로 Reno의 혼잡 회피 절차를 수정한다.



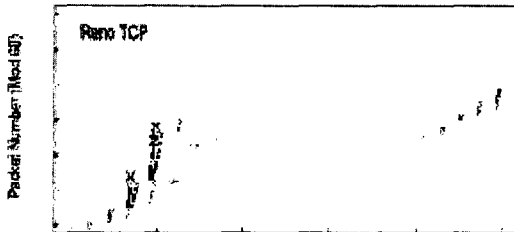
(a) TCP Tahoe 곡선



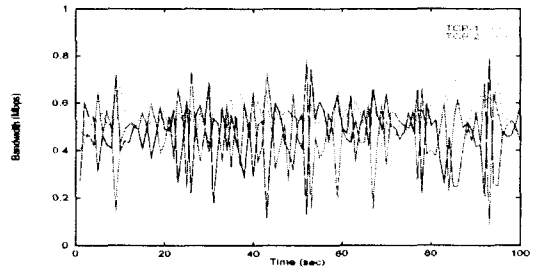
(b) TCPUDP-Tahoe 대역폭

그림 3-4. TCP Tahoe 곡선과 TCPUDP-Tahoe의 대역폭

Fig. 3-4 TCP Tahoe of curve and Bandwidth access of TCPUDP-Tahoe



(a) TCP Reno 곡선



(b) TCPUDP-Reno 대역폭

그림 3-5. TCP Reno 곡선과 TCPUDP-Reno의 대역폭
Fig. 3-5 TCP Reno of curve and Bandwidth access of TCPUDP-Reno

제어 규칙에서 $0 < a < b$ 이고 ΔRTT 는 연속된 두 RTT 값의 차이를 나타내며 τ 은 해당 ACK 패킷의 패킷 간격을 나타낸다. 결합은 증가 부분의 상수 a 를 $\epsilon(\mathcal{L}_2)$ 로 대체하고 네트워크 상태 측면에서 기반 피드백 혼잡제어의 추적 능력이 뛰어날수록 SSC 결합으로 얻는 성능 상의 이점이 커진다.

IV. 시뮬레이션 결과 및 분석

4.1 SSC의 기본 성능 특징

임계값에 의한 처리율을 곡선 지향할 때 최상의 방법으로서 네트워크 구성이 하나의 정적 처리율 상태에서 다른 상태로 전이하는 비정적인 상황에서 SSC의 적응성은 도움이 된다. 처리율 곡선의 굴곡 뿐 아니라 성능까지 증가하는데 이는 부분적으로 TCP Rate의 우수한 추적 능력 때문이다. 이 추적 능력은 SSC로 하여금 LTS 상관 구조를 보다 효과적으로 추출할 수 있게 하며 SSC를 Slow-Start, 혼잡 회피, 그리고 Slow-Start와 혼잡 회피 두 단계 모두에 적용한 경우에 대한 한계 효과를 나타낸다.

SSC의 효과에 영향을 미칠 수 있는 피드백 혼잡제어가 네트워크 상태 추적을 잘 할수록 LTS 상관 구조가 더 정확해지고 보다 효과적인 제어 활동이 수행된다. TCP에 대한 TCP-MT의 이러한 종속성은 TCP의 접속 단위마다 출력 반응을 사용하여 LTS의 네트워크 경쟁을 측정하는 SSC 때문이다. 오버헤드 관점에서 보면 네트워크에 검중 패킷을 전송하여 상태를 측정하거나 그렇지 않으면 네트워크 협력을 가정하는 별도의 상태를 측정하는 모듈을 만드는 것보다 이 방법이 더 효율적이다. 본 논문에서는 TCP Reno, Vegas, Ra-

te의 추적 능력을 측정하여 병목 링크 G_1, G_2 에서 집합 백그라운드 트래픽의 신뢰 처리율의 상관 계수를 각각 계산한다. 그림 4-1과 같이 효과적인 추적이 이루어질 경우에 가용 대역폭이 높아짐으로서 백그라운드 트래픽 레벨이 낮을 경우 신뢰 처리율은 높아지고 그 반대의 경우는 낮아짐을 알 수 있다. 따라서 완벽한 추적이 수행되는 경우 상관 계수는 -1로 계산되므로 Reno, Vegas, Rate의 계수 값이다. TCP Rate가 가장 우수한 추적 능력을 보이고 있으며, 그 다음으로 Vegas, Reno 순이다. Reno의 추적 능력 감소는 가용 대역폭의 빠르고 정확한 식별이 방해받는 Reno의 선형 증가 단계를 감안하기 때문이다.

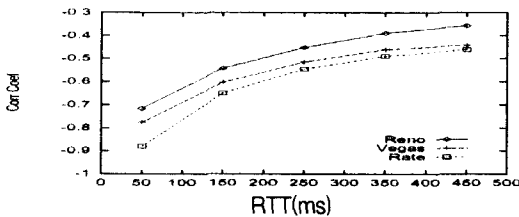


그림 4-1. TCP Reno, Vegas, Rate의 상관 계수 관점에서의 추적 능력

Fig. 4-1 Tracking ability in terms of correlation coefficient for TCP reno, Vegas, and Rate.

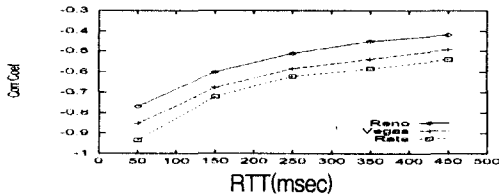


그림 4-2. TCP Reno, Vegas, Rate에 SSC가 적용된 경우 추적 능력이 향상되는 시너지 효과

Fig. 4-2 Synergy effect increasing tracking ability when SSC is applied to TCP Reno, Vegas, and Rate

시뮬레이션 결과로부터 얻을 수 있는 또 다른 특징은 왕복시간이 증가하는 경우에 추적 기능은 감소한다는 것으로서 사후 제어의 특징인 피드백 정보의 노후성 때문이다.

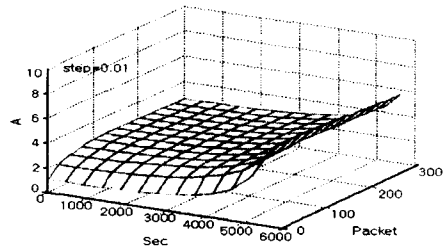
그림 4-2는 동일한 설정에서 SSC가 TCP Reno, Vegas, Rate와 각각 결합된 경우의 상관 계수를 나타내고 있으며, 모든 곡선이 -1을 향하고 있음을 알 수 있다. 이것은 조치의 시기적절성이 개선되어 TCP-MT가 TCP보다 추적 능력이 향상되는 결합의 시너지 효과를 나타낸다.

4.2 대칭적 메타컨트롤 및 공정성

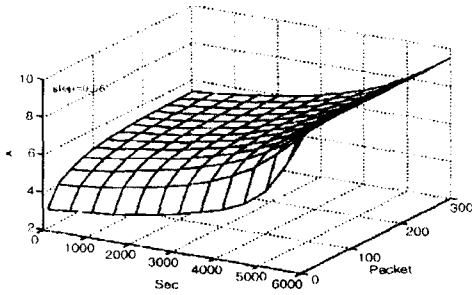
SSC 내에서 최대 기율기 레벨 A 를 동적으로 조정하는 메타컨트롤의 역할로서 대칭적 메타컨트롤의 안정성이 조정 인수 ν 에 따라 결정되며, ν 는 점근적 안정성을 가져올 만큼 작아야 하고, ν 값이 커지면 진동형 결과가 발생한다. 그림 4-3은 ν 값을 연속하여 5배씩 증가시키면서 대칭적 메타 컨트롤의 동적 특성을 나타내고 있다. ν 값이 커질수록 보다 두드러진 파장 형태의 결과를 얻는다. 더 중요한 점은 2가지 모든 경우에 대해 대칭적 메타컨트롤이 기율기 레벨 매개변수 A 의 값이 6을 향하여 나아가고 있으며 ν 는 A 주변의 진동 크기를 결정한다.

그림 4-4은 최대 적극성의 정적 설정 및 동적 설정에 따른 처리율 성능을 나타낸 것으로 단일 모드 곡선은 A 가 1-10 범위의 고정 priori 값으로 설정된 정적인 경우에 신뢰한 혼잡제어 처리율을 나타낸다. 동적 메타컨트롤의 처리율은 상위 점선으로 표시되며 이 값은 최적의 정적 최대 적극성 값인 $A=6$ 일 때의 성능과 거의 비슷한 결과를 나타내고 있다. 특정 네트워크 구성에서 A 가 어떤 priori 인지 파악하기가 어렵기 때문에 이 문제를 해결하기 위한 동적 메타컨트롤이 요구된다. 그림 4-5의 하위 점선은 참조용으로 TCP Rate의 처리율을 나타낸 것이다.

공정성 측면에서 보면 TCP-MT는 여러 개의 연결이 가용 자원을 향해 경쟁하는 공유 네트워크 환경에서 실행되도록 설계한 것으로서 여러 연결이 SSC와 관련될 경우에 대해 공정성 측면에서 TCP-MT의 반응을 살펴보고 TCP-MT 흐름의 대역폭 공유 반응을 다수의 TCP Reno 연결의 반응과 비교한 것이다. TCP에 SSC가 적용된 경우 대역폭 공유 반응 및 그 결을 나타내는 공정성 속성이 TCP와 질적인 면에서 동일하기 때문에 공정성이 아주 높았다.



(a) $\nu = 0.01$ 일 때 대칭적 메타컨트롤의 동적 특성



(b) $\nu=0.05$ 일 때 대칭적 메타컨트롤의 동적 특성
 그림 4-3. 조정 인수 ν 및 결과값 A의 함수로 표현한 대칭적 메타컨트롤의 동적 특성

(a): $\nu = 0.01$, (b): $\nu = 0.05$
 Fig. 4-3 Dynamic of symmetric control as a function of adjustment factor ν and the resultant evolution of A
 (a): $\nu = 0.01$, (b): $\nu = 0.05$

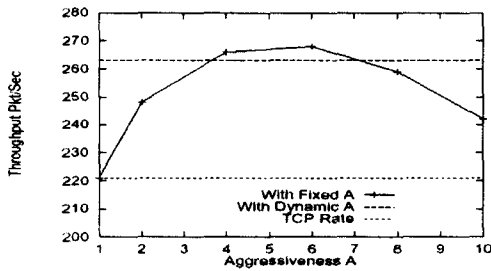


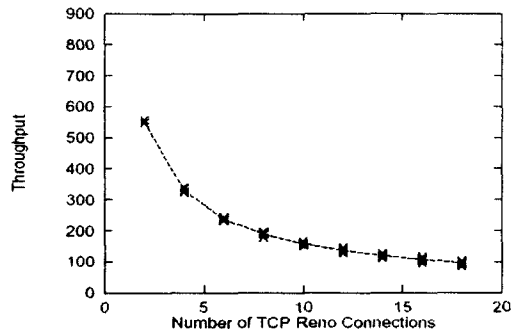
그림 4-4. 최대 기울기 레벨 A가 서로 다른 경우의 처리율 성능

Fig. 4-4 Throughput performance with different maximum slope levels A

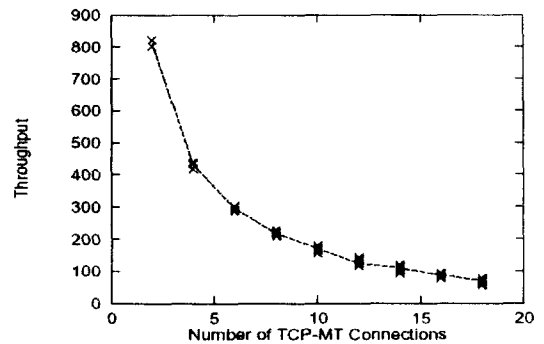
또한 길거나 짧은 대기 시간 연결, 패킷 크기, 윈도우 크기와 관련된 경우처럼 SSC도 TCP와 동일한 공정성 문제를 겪는다는 것을 알 수 있다. 이러한 결과는 그림 4-1의 설정을 기반으로 얻어진 것이며, 병목 링크 대역폭이 20Mbps로 증가하여 전체 50개의 TCP-MT 중 최대 18개만 수용하게 된 경우는 제외한다. 최초 32개 연결의 평균 트래픽 전송률, 즉 SSC 백그라운드 트래픽 소스는 5 Mbps에서 안정적으로 유지된다. 그림 4-3의 검은 점은 개별 처리율의 분포 현황이며 점선은 여러 연결의 평균 처리율을 나타낸다. 그림 4-3(b)는 TCP-MT 연결의 개수가 2개에서 18개로 증가하더라도, 대역폭은 최대-최소 측면에서 공정하게 공유하면서 성능도 향상되었다. 각 처리율의 분포는 좁은 범위에서 유지되며, TCP-MT 연결 개수가 늘어나면 개별 공유 폭은 줄어든다. 그림 4-5(a)는

TCP-MT를 TCP Reno로 대체했을 때 성능이 개선된 수치다.

표 1은 총 처리율 및 각 연결의 처리율 값 범위 형태로 자세한 정보를 제공한 것이다. 표의 첫 번째 행은 모든 연결에 걸쳐 실현된 총 처리율을 나타내며, 나머지 3개 행은 각 처리율의



(a) TCP Reno 연결시 대역폭 액세스



(b) TCP-MT 연결시 대역폭 액세스

그림 4-5 대역폭 액세스

(a) : 여러 TCP Reno 연결, (b) : 여러 TCP-MT 연결

Fig. 4-5 Bandwidth access

(a) Multiple TCP Reno connections, (b) Multiple TCP-MT connections

평균값 및 범위를 나타낸다. 첫 번째 행을 보면, 연결 개수가 최대 $n = 6$ 이 될 때까지 TCP-MT의 총 처리율은 증가하지만, 그 다음부터는 떨어지기 시작한다. 그러나 TCP-MT 연결 개수가 계속 증가하면, 전체 적극성의 증가는 가산 특성으로 인해 처리율에 부정적인 영향을 미치고 결국 순 감소를 유발하고 TCP Reno에서도 유사한 결과가 나타난다. 동시 피드백 혼잡제어 연결의 수가 증가하면서 전체적인 적극성도 증가되었다.

표 1. 여러 TCP-MT 연결
Table. 1 Multiple TCP-MT connections

SSC	n=2	n=6	n=10	n=14	n=18
Total Thr.	1621.1	1761.0	1691.0	1535.8	1253.0
Avg. Thr.	810.5	293.0	168.2	107.7	68.3
Max. Thr.	811.8	301.3	177.4	191.2	77.5
Min. Thr.	808.7	285.7	153.3	92.3	53.8

V. 결론

컴퓨터 네트워크에서 사용되는 메시지 교환에 관한 규칙을 정의 하는 프로토콜들의 모음을 의미 하는 전송제어 프로토콜을 트랜스포트 계층 프로토콜 상에서 다중시간 간격으로 혼잡제어를 연구 한다.

본 논문에서 LTS, SSC은 인터페이스를 통해 모듈화 될 수 있으며, 이 인터페이스를 통해 동일한 모듈을 Tahoe, Reno, Vegas, 전송을 기반 확장 버전 등 다양한 TCP에서 적용함으로써 네트워크 트래픽의 자기유사성 및 LRD를 나타냈다. TCP는 피드백 루프 범위에 속하지 않는 LTS 정보를 활용함으로써 하나의 유사 정적 처리를 상태에서 다른 상태로 이동하는 비정적인 상황에서도 SSC가 정확하게 적용되었으며, SSC로 하여금 LTS 상관구조를 효과적으로 추출할 수 있었다. 또한 다중 시간간격 TCP의 가장 중요한 속성은 피드백 루프와 관련된 왕복시간이 늘어나면 피드백이 전달하는 상태 정보가 노후 됨으로서 TCP가 취한 사후 혼잡 제어 조치의 효과가 떨어지는데 이와 같은 현상은 지연-대역폭의 좁은 지연 또는 대역폭과 비례하여 증가 되었다.

선택적 기율기 제어의 적극성 스케줄을 연결 지속 시간 및 개방 루프와 폐쇄 루프에 적용한 결과 선택적 기율기 제어가 연결 지속 시간이 늘어남에 따라 성능 이득도 향상되었다. 이러한 결과로부터 정확한 지속 기간 및 priori 정보의 유무에 따라 짧은 연결에서도 성능 이득을 향상 시킬 수 있음을 입증하였다. SSC 내에서 최대 기율기 레벨 A 를 동적으로 조정하는 메타컨트롤의 역할로써 최대 적극성의 정적 설정 및 동적 설정에 따른 처리율 성능이 A = 6일 경우에 최적의 정적 최대 적극성 값을 가지고 있다.

대기 시간 연결, 패킷 크기, 윈도우 크기와 관련

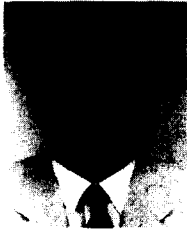
하여 선택적 기율기 제어도 TCP와 마찬가지로 동일한 공정성 문제에 있어서 TCP-MT 연결의 개수가 2개에서 18개로 증가하더라도 대역폭은 최대-최소 측면에서 공정하게 공유되었으며, 각 처리율의 분포는 좁은 범위 내에서 유지되고 TCP-MT 연결 개수가 늘어남에 따라 개별 공유 폭은 줄어들고 동시에 피드백 혼잡제어 연결의 수가 증가하면서 전체적으로 성능이 향상 되었다.

참고문헌

- [1] N. Poojary, S. V. Krishnamurthy, and S. Dao. Medium Access Control in a Network of Ad Hoc Mobile Nodes with Heterogeneous Power Capabilities. In IEEE International Conference on Communications (ICC 2001), Vol. 3, pp. 872-877, 2001.
- [2] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. In INFOCOM 2000, Vol. 2, pp. 404-413, March 2000.
- [3] C. Lee, J. Jang, E.K. Park, S. Makkic. A simulation study of TCP performance over IEEE 1394 home networks. Computer Communications Vol. 26, pp. 670-678, 2003.
- [4] T. Tuan and K. Park. Multiple time scale congestion control for self-similar network traffic. *Performance Evaluation*, Vol. 36, pp. 358-386, 1999.
- [5] T. Tuan and K. Park. Multiple time scale redundancy for QoS-sensitive transport of real-time traffic. To appear in Proc. IEEE INFOCOM '00, 2000.
- [6] K. Park, W. Wang. QoS-sensitive transport of real-time MPEG video using adaptive forward error correction. In Proc. IEEE Multimedia Systems '99, pp. 426-432, 1999.
- [7] K. Park and W. Willinger. Self-similar network traffic: An overview. In K. Park and W. Willinger, editors, *Self-similar Network Traffic and Performance Evaluation*. Wiley Interscience, 1999.
- [8] K. Park and W. Willinger, editors. *Self-similar Network Traffic and Performance Evaluation*. To be published by Wiley Interscience, 1999.
- [9] A. Feldmann, A. C. Gibert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of Internet

WAN traffic. In] '98, pp. 42-55, 1998.
 [10] A. C. Gilbert, W. Willinger, and A. Feldmann. Scaling analysis of conservative cascades. with application to network traffic. *IEEE Trans. Information Theory*, Vol. 45, No. 3, pp. 971-991, 1999.

저자 소개



나상동(Sang-dong Ra)

1968년 조선대학교 전기공학과 졸업(공학사)

1980년 건국대학교 대학원 졸업(공학석사)

1995년 원광대학교 대학원 졸업(공학박사)

1973년~현재 조선대학교 전자.정보대학 컴퓨터공학부 교수

1995년~1996년 Dept. of Electrical & Computer Eng. Univ. of California Irvine 연구교수.

1998년 조선대학교 정보전산원장 역임

2001년~2002년 Dept. of Electrical & Computer Eng. Univ. of California Irvine 연구교수

※관심 분야 : 실시간 통신, 디지털 통신 신호제어, 데이터 및 이동통신, 신경망, 적응 신호처리 등임.



박동석(Dong-Suk Park)

1995년 광주대학교 전자계산학과 졸업(공학사)

2000년 조선대학교 산업대학원 졸업(공학석사)

2005년 현재 조선대학교 대학원 박사과정

1998년~ 2003년 전남과학대학 겸임전임강사

2000년~ 현재 조선대학교 컴퓨터공학부 시간강사

※관심 분야 : 실시간 통신, 통신망, 이동통신.