TCP의 동적 적응을 위한 결정 메커니즘

황재현O 최진희 김세원 유혁
고려대학교 컴퓨터학과
{jhhwangO, jhchoi, swkim, hxy}@os.korea.ac.kr

Dynamic Adaptation of TCP

Jae-Hyun Hwang0 Jin-Hee Choi Se-Won Kim Hyuck Yoo
Dept. of Computer Science, Korea University

요 약

다양한 통신 매체의 등장과 애플리케이션의 요구사항이 늘어남에 따라 유연성과 확장성이 있는 네트워크 시스템에 대한 연구가 진행되어 왔다. 이러한 시스템들은 새로운 서비스나 프로토콜의 확장이 쉽게 가능하며, 사용자의 요구사항에 따라 프로토콜 스택을 구성해낼 수 있는 유연한 구조를 가진다는 특징이 있다. 그러나 사용자의 요구사항에 맞추어 구성된 프로토콜이라 할지라도 네트워크의 변화하는 환경을 반영할 수 없다면 시스템의 성능 저하 문제는 피할 수 없다. 이에 본 논문에서는 유연성과 확장성을 고려한 시스템에서 네트워크의 변화하는 환경을 추가적으로 고려하여 TCP를 대상으로 적절한 변종의 선택을 결정하여 주는 메커니즘을 제안한다. 제안된 기법에 대해 시뮬레이션을 통하여 그 적합성을 평가하였고 다양한 네트워크 상황이 주어졌을 때 표준 TCP와 비교하여 최대 80~194%의 성능 향상을 가져올 수 있음을 보였다.

1. 서 론

인터넷이 등장한 이래 현재까지 유선 망뿐 아니라 무선 망, 위성 망에 이르기까지 다양한 통신 환경들이 발전되어 왔다. 그에 따라 이동성이 부여된 소형 컴퓨팅 장치 형태의 단말들이 등장하였고 사용자들이 사용하는 애플리케이션의 종류도 급증하였다. 이러한 통신 환경의 변화는 여러 네트워크 자원의 결합을 통해 사용자에게 매우 다양한 선택을 가능하게 해줄 것으로 예상된다. 여기에 발 맞추어 단말들의 네트워크 시스템 또한 애플리케이션의 다양한 요구사항과 네트워크의 환경 변화에 적절히 대응 가능하도록 유연성과 확장성을 가져야 할 필요가 있다.

유연성과 확장성을 고려한 네트워크 시스템들은 이미 1990년대 초반부터 시작하여 TIP, ADAPTIVE, F-CSS 등의 프로젝트를 통하여 제안되었는데, 이러한 시스템들은 1) 새로운 서비스나 프로토콜이 추가되어야 할 때 쉽게 확장 가능한 구조를 가지며, 2) 사용자의 요구사항에 따라 프로토콜 스택을 동적으로 조합할 수 있는 유연한 구조를 가진다는 특징이 있다[1]. 여기서 주목해야 할 부분은 두 번째 특징, 즉 동적으로 프로토콜 스택을 구성하는데 있어서 중요한 기준이 되는 것이 바로 사용자(또는 애플리케이션)의 요구사항이라는 것이다. 이것은 언급한 시스템들의 공통된 특징으로, 애플리케이션들이 성능, 신뢰성, 시간제약, 보안 등과 같은 속성을 요구하면 시스템은 각 요구사항을 만족시키는 작은 단위의 모듈들을 조합하여 실행 시간에 동적으로 프로토콜 스택을 구성하는 방식을 취하고 있다. 그러나 네트워크의 변화하는 환경을 고려하지 않는다면 프로토콜의 성능은 저하될 수 밖에 없다. 특히, TCP의 경우는 처음 설계될 당시의 가정들이 오늘날 새로운 망에 적용되지 않는 사례가 등장하면서 이들을 해결하려는 연구들이 제안되어 왔기 때문에 망의 상황을 고려하는 것은 프로토콜의 적응에 있어서 중요한 문제라 할 수 있다.

이에 본 논문에서는 앞서 언급한 시스템을 바탕으로 TCP를 망의 환경에 적응 시키기 위해 프로토콜 모듈을 Slow Start, Congestion Avoidance, Error Recovery로 분류하고 네트워크의 상황에 근거하여 적절한 모듈의 선택을 가능하게 해주는 결정 메커니즘을 제안한다.

본 논문의 전체적인 구성 순서는 다음과 같다. 2장은 TCP의 구성에 있어서 각 모듈을 선택하는 방법에 대해 자세히 기술한다. 3장은 제안된 기법으로 인해 예상되는 성능의 향상 정도를 평가한다. 마지막으로 4장에서 결론을 내리면서 논문을 마무리한다.

2. 프로토콜 구성을 위한 모듈 결정 방법

본 논문에서 제안하는 결정 메커니즘의 목적은 네트워크의 환경을 추정한 뒤, 그 정보를 바탕으로 TCP를 적절하게 적응시키도록 하는 것이다. 네트워크의 상태를 표현하기 위해 본 논문에서는 지연시간(round-trip delay),

본 연구는 한국과학재단 특정기초연구(R01-2004-000-10588-0) 지원으로 수행되었음.

지연시간의 양방향 간 비 대칭률(asymmetric ratio), 손실률(loss rate), 손실 패턴(loss pattern)의 네 가지 파라미터를 사용하였다. 이 각각의 정보에 대한 측정 기법들은 이미 많은 연구가 이루어진 상태이기 때문에, 여기서는 각 기법들의 정확성에 대해 언급하기 보다는 각 정보들이 정확하게 측정 가능하다고 가정한다.

일단 네트워크의 환경을 네 가지의 파라미터를 통해 측정하였으면 현재 상황에 적합한 TCP 모듈을 선택해야 한다. 이를 위해 선행되어야 할 몇 가지 과정이 있다. 먼저 다양한 네트워크 환경에서 여러 TCP 변종들의 처리량(throughput) 정보를 측정하여 프로토콜의 성능 데이터베이스를 구축한다. 이것은 시뮬레이션이나 도는 실제 네트워크 망에서의 측정을 통해 이루어질 수 있다. 측정된 값은 다음과 같이 처리량의 기대 값으로 나타낸다.

$$Expected\ Throughput_{(SS\_i,\,CA\_j,\,ER\_k)}(RD, AR, LR, LP)$$

이것은 즉 i 번째 Slow Start 모듈, j 번째 Congestion Avoidance 모듈, k 번째 Error Recovery 모듈로 조합된 TCP가 사용되는 시스템에서 기대되는 처리량을 의미하며, 이 때 주어지는 인자는 각각 Round-trip Delay(RD), Asymmetric Ratio(AR), Loss Rate(LR), Loss Pattern (LP)을 의미한다.

두 번째 단계는 ANOVA[5]를 이용하여 주어진 처리량에 대한 각 메커니즘의 영향 정도를 판단한다. 현재 시스템에 SS, CA, ER 별로 두 가지의 서로 다른 모듈이 존재한다고 할 때, 다음과 같은 변수의 정의가 가능하다.

$$x_{SS} = \begin{cases} -1 & for\ SS\_1 \\ 1 & for\ SS\_2 \end{cases}$$

$$x_{CA} = \begin{cases} -1 & for\ CA\_1 \\ 1 & for\ CA\_2 \end{cases}$$

$$x_{ER} = \begin{cases} -1 & for\ ER\_1 \\ 1 & for\ ER\_2 \end{cases}$$

이 변수를 이용하여 처리량의 기대 값 y는 다음과 같이 표현될 수 있다. 여기서 q_0는 y의 평균 성능이 되며, 나머지는 각 모듈의 영향 정도를 나타낸다.

$$y = q_0 + q_{SS}x_{SS} + q_{CA}x_{CA} + q_{ER}x_{ER}$$

이제 세 변수의 값들과 위 식을 조합하여 아래의 네 새식을 도출할 수 있다.

$$thru\_1 = q_0 - q_{SS} - q_{CA} - q_{ER}$$
$$thru\_2 = q_0 + q_{SS} - q_{CA} - q_{ER}$$
$$thru\_3 = q_0 - q_{SS} + q_{CA} - q_{ER}$$
$$thru\_4 = q_0 - q_{SS} - q_{CA} + q_{ER}$$

이 식을 풀게 되면 결과적으로 다음과 같이 각 모듈의 영향력을 구해낼 수 있다.

$$x_{SS}\text{'s } impact = \frac{q_{SS}^2}{q_{SS}^2 + q_{CA}^2 + q_{ER}^2}$$

$$x_{CA}\text{'s } impact = \frac{q_{CA}^2}{q_{SS}^2 + q_{CA}^2 + q_{ER}^2}$$

$$x_{ER}\text{'s } impact = \frac{q_{ER}^2}{q_{SS}^2 + q_{CA}^2 + q_{ER}^2}$$

구해진 영향력은 프로토콜의 모듈 조합 시 성능에 높은 이득을 미치는 것으로 선택하기 위해 사용된다. 예를 들어 한 시스템이 [SS_1, CA_1, ER_1]의 조합을 가지는 TCP를 사용하며 각 모듈의 영향력이 현재 환경에서 각각 80%, 15%, 5%라고 한다면, 이것은 SS_1가 아닌 SS_2로 조합했을 때 기존보다 약 80%의 성능 향상이 기대된다는 의미가 된다.

3. 성능 평가

이번 절에서는 몇 가지 시나리오를 바탕으로 NS-2[6]를 이용한 시뮬레이션을 수행하여 본 논문에서 제안한 결정 메커니즘을 평가한다. 본 실험에서는 논문의 서론 부분에서 언급한 네트워크 시스템이 TCP의 부분 재조합을 지원해준다는 가정하에, TCP의 변종들만을 사용하여 같은 결과를 이끌어 내었다. 예를 들어, Reno에서 SS 모듈이 BW-Estimate로 바뀐 경우와 처리량 비교를 하기 위해서는 단순히 Westwood와 비교하면 된다. 이 관계를 쉽게 확인하기 위해 표 1에 시뮬레이션에 사용된 변종들과 그에 대응하는 모듈들을 정리하였다.

네트워크의 환경의 변화는 표 3에 정리하였으며, 하나의 시나리오에서 한 번에 100초씩 같은 실험을 100번 반복하여 평균 처리량을 얻어내었다. 그 후, 2절에 설명한 방법에 따라 각 환경에서 가장 높은 영향력을 가지는 모듈을 결정하였다.

표 1. 시뮬레이션에 사용된 TCP 변종

TCP Variant	Slow Start	Congestion Avoidance	Error Recovery
Reno	Heuristic	AIMD	Duplicate ACK
SACK[4]	Heuristic	AIMD	Duplicate ACK
BI[3]	Heuristic	Binary Search	Selective ACK
West-wood[2]	Bandwidth Estimate	AIMD	Duplicate ACK

표 2. 각 시나리오 별 최대 처리량 증가율 및 모듈 별 영향력 요약

	Maximum Throughput Improvement	Average Impact of SS	Average Impact of SS	Average Impact of SS	Main Reconfigured Instance
Scenario 1	192%	2.85	97.08	0.07	Binary Search
Scenario 2	194%	5.94	93.95	0.11	Binary Search
Scenario 3	80%	18.99	45.69	35.32	Binary Search
Scenario 4	102%	17.23	41.34	41.43	SACK

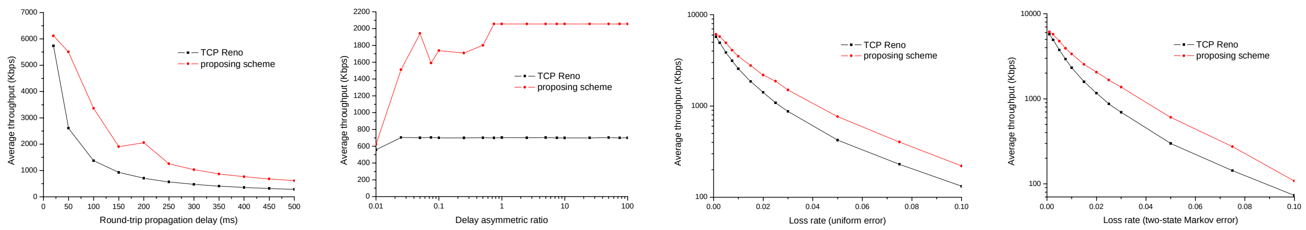

그림 1. 평균 처리량 그래프 : TCP Reno 와 제안된 메커니즘을 적용하여 특정 환경에 적응된 TCP 조합과의 비교

표 3. 각 시나리오 별 네트워크 파라미터

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Band-width	10 Mbps	10 Mbps	10 Mbps	10 Mbps
Round-trip delay	20~500 ms	20 ms	20 ms	20 ms
Asym-metric ratio	1.0	0.01~100	1.0	1.0
Loss rate	0.001	0.001	0.001~0.3	0.001~0.3
Error model	uniform	uniform	uniform	2 state Markov

실험 결과는 표 2와 그림 1에 정리하였다. 먼저 시나리오 1에서는 지연시간의 변화에 따른 TCP의 적응을 보여주고 있는데, CA에 의한 영향력이 가장 큰 것으로 나타나고 있다. 이는 실험에 사용된 Binary-search 혼잡 제어 기법이 높은 BDP(Bandwidth-Delay Product)망을 대상으로 제안된 것이기 때문이다. 시나리오 2는 비 대칭 환경에 적합한 변종이 실험에 사용되지 않았기 때문에 보유한 모듈 중 높은 것을 선택한 것이며, 시나리오 4에서는 예상대로 ER의 영향력이 가장 높게 나타나고 있다. 즉, 대부분의 환경에서 Reno를 대체할 수 있는 모듈이 존재하며, 이는 환경에 따른 TCP의 적응이 잘 이루어지고 있음을 보여주고 있다.

4. 결 론

본 논문에서는 다양한 네트워크 환경에 TCP를 적응시키기 위한 결정 메커니즘을 제안하였다. 기존의 유연성 있는 네트워크 시스템에서는 애플리케이션의 요구사항에 기반하여 프로토콜을 조합한 반면, 본 논문에서는 손실률 등과 같은 네트워크 파라미터에 기반하고 있다. 즉, 네트워크의 상황에 따라 적합한 TCP 전송 알고리즘이 다르다는 사실에 기인한 것이며, 시스템에서 보유하고 있는 모듈을 대상으로 성능에 대한 통계적 분석을 거쳐 결정의 신뢰성을 높였다. 또한 시뮬레이션 결과, 제안된 결정 메커니즘을 이용하여 TCP Reno에 비해 최대 80~194%의 성능 향상을 가져올 수 있음을 보였다.

참고 문헌

[1] S. Bocking, "Object-Oriented Network Protocols," In Proceedings of IEEE INFOCOM, 1997.
[2] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, S. Mascolo, "TCP Westwood: Congestion Window Control Using Bandwidth Estimation," In Proceedings of IEEE GLOBECOM, Nov. 2001.
[3] L. Xu, K. Harfoush and I. Rhee, "Binary increase congestion control for fast long-distance networks," IEEE INFOCOM, 2004.
[4] S. Floyd, M. Mahdavi, M. Mathis, and M. Podolsky, "An Extension to the Selective Acknowledgement(SACK) option for TCP," RFC 2883, IETF, 2000.
[5] George W. Cobb, "Introduction to Design and Analysis of Experiments," Springer, Mar. 1998.
[6] ns2 Network Simulator version 2.26, http://www.isi.edu/nsnam/ns, 2003.