

TCP Vegas의 성능 파라미터에 관한 연구

표지훈*, 정윤식, 임석구
백석대학교

pyojihun@gmail.com, zacard@naver.com, sklim@bu.ac.kr

A Study on Performance Parameter of TCP Vegas

Ji-Hun Pyo*, Yun-Sick Chung, Seog-Ku Lim
BaekSeok University

요 약

인터넷의 안정성에 가장 큰 영향을 미치는 요소는 종단간에 이루어지는 TCP 혼잡제어이다. 현재 TCP Vegas가 효율성과 안정성 측면에서 TCP Reno보다 우수함에도 불구하고, TCP Vegas와 TCP Reno를 혼용하여 사용하는 경우 충분한 성능을 발휘하지 못하는 경우와 또한 RTT가 서로 다른 TCP Vegas간에 존재하는 공정성 문제가 있다. 본 논문에서는 이러한 불공정성 문제에 대해 TCP Vegas의 알고리즘의 주요 파라미터인 α , β , γ 값의 변화에 따라 TCP Reno와의 공정성 향상과 RTT가 서로 다른 TCP Vegas끼리의 공정성 문제를 분석하였다.

1. 서 론

TCP(Transmission Control Protocol)의 혼잡제어 알고리즘(Congestion control algorithm)은 1988년의 TCP Tahoe 이래로 1990년 TCP Reno, 1995년 TCP SACK에 이르기 까지 다양하게 연구되어 왔다[1]. 이러한 TCP 혼잡제어 알고리즘의 주요 목적은 송신단의 전송률을 제어하여 혼잡상황으로 인해 발생하는 데이터의 무분별한 손실을 막기 위함이다.

현재 여러 TCP 알고리즘 중 TCP Reno가 가장 많이 사용되며, TCP Reno의 성능을 개선하기 위한 많은 제안이 이루어지고 있다. TCP Reno는 패킷이 손실되었다고 판단될 경우에만 혼잡 상황으로 보고 혼잡 윈도우(CWND : Congestion Window)를 일정 비율 줄이는 방법을 사용함으로써, 가용한 대역폭이 큰 폭으로 변할 경우 제대로 적응을 하지 못하는 단점을 가지고 있다[2][3].

TCP Vegas는 TCP Reno에 비해 전송량, 네트워크의 안정성, 패킷 손실률, 지연시간 등 많은 부분에서 더 우수한 혼잡제어 기법임이 증명되었다[4][5]. 그러나 실제 TCP Vegas와 TCP Reno를 동일한 네트워크 환경에서

경쟁시키면, TCP Reno의 전송률의 변화에 따른 영향으로 경쟁하는 TCP Vegas의 전송률이 현저하게 감소하는 현상이 나타난다. 이러한 문제를 불공정성이라 한다. 또한 TCP Vegas 플로우(Flow) 간의 경쟁에서도 두 가지의 문제점이 발생한다. 첫 번째는 RTT가 서로 다른 TCP Vegas 플로우의 경쟁에서 RTT가 작은 플로우의 RTT가 큰 플로우에 비해서 더 낮은 전송률을 갖게 되는 불공정성 문제이며 두 번째는 연결 시점이 다른 TCP Vegas 플로우 간의 경쟁에서 이전에 연결을 맺은 플로우가 이후에 연결을 맺은 플로우에 비해서 더 낮은 전송률을 갖게 되는 불공정성 문제이다.

본 논문에서는 TCP Vegas 알고리즘에서 사용되는 파라미터인 α , β , γ 값의 변화에 따른 TCP Vegas와 TCP Reno의 공정성 문제를 검토하고 이를 기반으로 향후 연구방향을 제시하였다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 TCP Vegas 알고리즘에 대한 설명하고, 3장에서는 α , β , γ 값의 변화에 따른 성능을 분석하기 위해 시뮬레이션을 수행하여 그 결과를 분석하며, 마지막으로 4장에서는 결론을 맺는다.

2. TCP Vegas 알고리즘

TCP Vegas는 패킷 손실에 대한 정보만으로 네트워크의 혼잡제어를 수행하는 TCP Reno와 달리 측정된 RTT를 기반으로 네트워크의 혼잡제어를 수행한다. 즉, RTT가 커지면 네트워크의 혼잡정도가 증가되었다고 판단하여 윈도우 크기를 감소시키고, 반대로 RTT가 작아지면 네트워크의 혼잡정도가 감소되었다고 판단하여 윈도우 크기를 증가시킨다. 이러한 TCP Vegas의 혼잡제어 알고리즘은 윈도우의 크기를 이상적인 상태로 관리하는데 효과적이며, 결과적으로 평형상태에서도 적절한 전송률을 유지할 수 있다.

이와 같은 TCP Vegas의 윈도우 크기 변화는 TCP Reno에 비해 진동폭이 작고 전체적으로 안정된 상태를 유지하는 특징이 있다. 이러한 TCP Vegas의 혼잡제어 알고리즘의 동작과정을 기능별로 서술하면 다음과 같다 [6][7].

혼잡이 없을 때의 RTT를 BaseRTT라고 정의하면, 측정된 RTT중 최소 RTT가 BaseRTT로 지정된다. 일반적으로 라우터의 큐가 길어지기 이전인 첫 번째 TCP 세그먼트에 대한 RTT 측정값이 BaseRTT가 된다.

BaseRTT 동안에 전송할 수 있는 윈도우 크기를 w 라고 하면, 이 플로우에 대해 기대전송률인 Expected를 식 (1)과 같이 구한다.

$$Expected = \frac{w}{BaseRTT} \quad (1)$$

측정된 RTT동안에 실제로 전송한 실제전송률인 Actual은 w 를 현재의 RTT로 나누어 식 (2)와 같이 구한다. $BaseRTT \leq RTT$ 이므로 $Actual \leq Expected$ 의 관계가 항상 성립한다.

$$Actual = \frac{w}{RTT} \quad (2)$$

Actual과 Expected를 비교하여 두 전송률 간의 차이인 Diff를 구한다. Diff는 네트워크의 상태를 정량적으로 표현한 값이다.

$$Diff = (Expected - Actual) \times BaseRTT \quad (3)$$

TCP Vegas는 Diff와 두 개의 임계값 α 와 β 를 이용하여 식 (4)와 같이 세 개의 구간을 정의하여 Diff에 따라 윈도우 크기를 조절한다.

$$w = \begin{cases} w - 1, & \text{if } Diff > \beta \\ w, & \text{if } \alpha \leq Diff \leq \beta \\ w + 1, & \text{if } \alpha > Diff \end{cases} \quad (4)$$

3. 시뮬레이션 및 결과 분석

3.1 시뮬레이션 모델

본 논문에서 사용한 시뮬레이션 모델은 그림 1과 같이 구성하였고 각 링크의 부하는 70%로 실험하였으며, 각 Source 노드의 수는 각 10개로 설정하였다. 각 Source 노드에서는 512 kbyte의 고정된 크기를 갖는 패킷을 지수분포에 따라 랜덤하게 발생시키며 라우터 R1에 도착한다. 라우터 R1은 Source 노드로부터 전송된 데이터 패킷을 1.5Mbps의 전송용량, 20ms의 전송 지연을 갖는 링크를 통해 라우터 R2로 전송한 후, 마지막으로 라우터 R2는 목적지 노드로 패킷을 전송한다. S1~S10까지의 노드는 5~90초 동안 패킷을 발생시키고, S11~S20까지의 노드는 10~90초 동안 패킷을 발생시켰다. 시뮬레이션은 LBNL(Lawrence Berkely National Laboratory)의 NS-2(Network Simulator)를 사용하여 수행하였다[8].

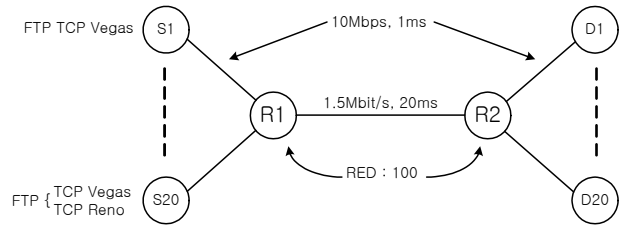


그림 1. 시뮬레이션 모델

3.2 α , β , γ 값의 변화에 따른 공정성

앞에서 살펴본 바와 같이 TCP Vegas의 공정성 문제를 해결하기 위하여 TCP Vegas 알고리즘에서 사용하는 상수 α , β , γ 의 값을 변화시키면서 공정성 변화에 미치는 영향을 분석하였다. 그림 2는 혼잡 상황에서 TCP Vegas와 TCP Reno가 경쟁하는 경우를 나타내었는데, α , $\beta(=\alpha+2)$ 값을 변화시켰을 경우 $\alpha=2$, $\beta=4$, $\gamma=1$ 인 경우가 공정성이 가장 좋을 것을 알 수 있다.

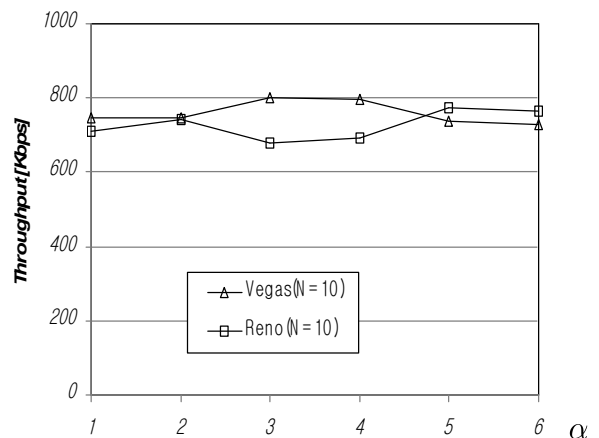


그림 2. Vegas vs Reno 경쟁($\gamma=1$, $\beta = \alpha + 2$)

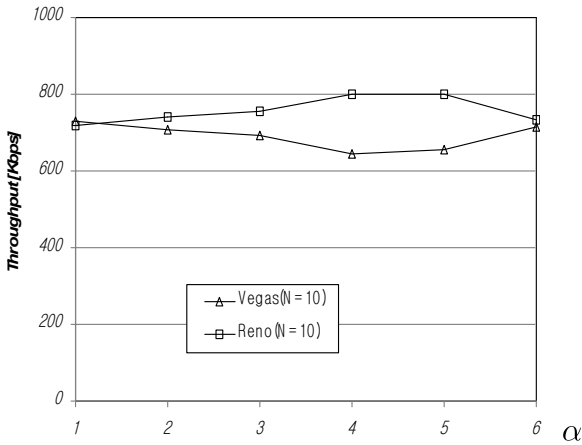


그림 3. Vegas vs Reno 경쟁($\gamma=\alpha, \beta = \alpha + 2$)

그림 3은 $\alpha=\gamma$ 인 경우를 나타내었는데, α 와 β 값을 증가시켜도 TCP Vegas와 TCP Reno간의 공정성에는 문제가 없음을 알 수 있다.

그림 4는 RTT가 서로 다른 TCP Vegas들이 경쟁하는 경우를 나타내었는데, $\alpha=3, \beta=5, \gamma=1$ 와 $\alpha=5, \beta=7, \gamma=1$ 에서 공정성이 가장 좋은 영향을 주지만, α 와 β 의 변화에 따른 각각의 Vegas의 전송률의 변화의 차이가 크기 때문에 α 와 β 의 변화는 RTT가 서로 다른 TCP Vegas의 공정성 문제에서는 큰 영향을 미치지 못함을 알 수 있다.

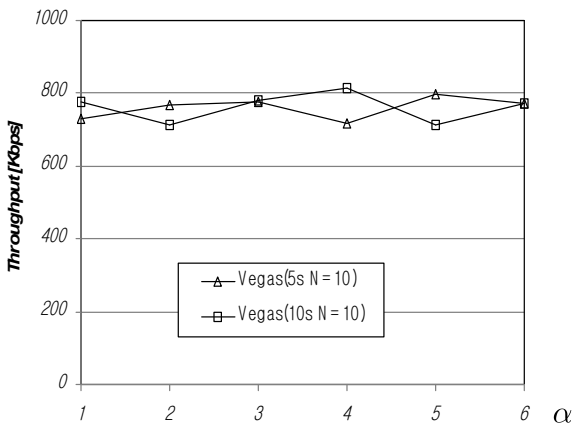


그림 4. RTT가 서로 다른 Vegas간 경쟁($\gamma=1, \beta = \alpha + 2$)

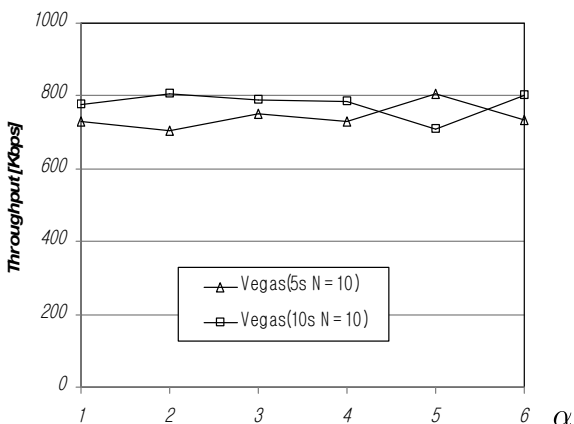


그림 5. RTT가 서로 다른 Vegas간 경쟁($\gamma=\alpha, \beta = \alpha + 2$)

마지막으로 그림 5는 $\alpha=\gamma$ 인 경우를 나타내었는데, α 와 β 의 변화는 TCP Vegas끼리의 경쟁에서는 공정성 향상에 영향을 미치지 못함을 알 수 있었다.

4. 결론

현재까지 송신단의 전송률을 제어하여 혼잡으로 인해 발생하는 데이터의 무분별한 손실을 막기 위한 많은 TCP 혼잡제어 알고리즘이 연구되어 왔다. 이러한 알고리즘 가운데 TCP Vegas는 효율성과 안정성 측면에서 우수한 혼잡제어 알고리즘으로 증명되었지만, 불공정성 문제로 인해 사용에 제약을 받아 왔다.

본 논문에서는 TCP Vegas의 상수인 α, β, γ 값의 변화에 따라 공정성 문제를 분석하였다. 분석 결과 지연이 높은 네트워크에서 α, β, γ 값의 변화는 TCP Vegas와 TCP Reno에서는 공정성에 큰 영향을 미치지 않지만, RTT가 서로 다른 TCP Vegas의 공정성 향상에 큰 영향을 미치지 못함을 알 수 있었다.

향후 연구 과제로는 RTT가 서로 다른 TCP Vegas의 공정성을 위한 방법과 여러 상황에서의 α, β, γ 값을 구하기 위한 지속적인 연구가 필요하다.

참고문헌

- [1] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", *Proceeding of ACM SIGCOMM '96*, pp.5.-21, July 1996.
- [2] V. Jacobson, "Congestion avoidance and control", Addison Wesley Publishing Company, 1994.
- [3] F. Kevin and S. Floyd, "Simulation based Comparisons of Tahoe, Reno, and SACK TCP", *Computer Communication Review*, October 1996.
- [4] J. Mo, R. La and J. Walrand, "Analysis and Comparison of TCP Reno and TCP Vegas", *Proceeding of IEEE INFOCOM '99*, pp.1556-1563, March 1999.
- [5] T. Bonald, "Comparison of TCP Reno and TCP Vegas: Efficiency and Fairness", *Proceeding of Elsevier Science Publisher*, pp.307-332, August 1999.
- [6] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm", *LBNL Technical Report*, April 1990.
- [7] L. Brakmo, S. O'Malley and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", *Proceeding of ACM SIGCOMM '94*, pp.24-35, August 1994.
- [8] The Network Simulator ns-2, <http://www.isi.edu/nanam/ns/>