

# 임베디드 환경을 위한 TCP의 새로운 버퍼관리 메커니즘

이승호<sup>0</sup>, 이선현, 최웅철, 이승형, 정광수

광운대학교 전자공학부

{shlee<sup>0</sup>, sunlee}@cclab.kw.ac.kr, {wchoi, shrhee, kchung}@daisy.kw.ac.kr

## A New Buffer Management Mechanism of TCP for Embedded Environment

Seungho Lee<sup>0</sup>, Sunhun Lee, WoongChul Choi, Seung Hyong Rhee, Kwangsue Chung

School of Electornics Engineering, Kwangwoon University

### 요 약

최근 종단 호스트에서 나타나는 데이터 전송의 병목현상이 두드러지면서 종단 호스트에 관한 많은 연구가 진행되고 있다. 기존의 연구를 통해 종단 호스트의 병목현상의 원인으로 TCP의 비효율적인 버퍼관리가 제기되었다. 특히 임베디드 환경에서는 시스템 자원이 제한적일 수밖에 없기 때문에 TCP의 버퍼관리의 효율성이 매우 중요하다. 본 논문에서는 새로운 TCP 버퍼할당 기법인 RTBA(Rate-Based TCP Buffer Allocation) 기법을 제안하였다. RTBA는 플로우의 RTT(Round Trip Time), RTO(Retransmit Time Out), 패킷 손실률 등을 고려한 TCP의 기대전송율을 기반으로 버퍼를 동적으로 할당함으로써 상대적으로 적은 버퍼를 사용하는 임베디드 환경에서도 다수의 TCP 플로우들이 높은 전송성능을 얻게 한다. 실험을 통해 제안한 RTBA 기법이 기존의 고정적인 버퍼할당 기법에 비해 우수한 성능을 보임을 확인할 수 있었다.

### 1. 서 론

이전의 전송계층에 관련된 네트워크 기술들은 증가된 트래픽으로 인해 발생하는 혼잡상황을 해결하기 위한 TCP(Transmission Control Protocol)의 혼잡제어 기법을 위주로 연구되어 왔다 [1]. 그러나 네트워크에서의 혼잡상황이 크게 개선됨에 따라 전송 성능이 상대적으로 송수신 호스트에 의해 제약 받는 현상이 나타났으며, 이를 해결하기 위해서 TCP 자체의 제어뿐만 아니라 종단 호스트의 제어에 대한 필요성이 제기되었다.

기존에 구현된 유닉스 환경의 TCP는 256KB, 리눅스 환경의 TCP는 64KB와 같이 전송링크의 용량에 관계없이 모든 플로우에 동일하게 버퍼를 할당함으로써 종단호스트에서 병목현상을 야기한다. 이러한 병목현상은 윈도우 기반의 전송메커니즘을 사용하는 TCP에 있어 부족한 버퍼공간으로 인해 전송윈도우가 충분히 커지지 못하는 현상으로 설명할 수 있다. 이는 웹서버와 같이 다수의 플로우를 관리해야 하는 인터넷 서버에서 주로 발생하며, 최근 들어 임베디드 시스템에서의 인터넷 서비스에 대한 관심이 높아지면서 안정적인 서비스를 위한 효율적인 버퍼할당 기법의 필요성이 대두되고 있다.

Tierney는 BDP(Bandwidth Delay Product)를 버퍼크기로 할당하는 기법을 소개하였다 [2]. 이는 연결이 맺어지기 이전에 별도의 유틸리티를 통해 BDP를 측정하여 시스템 콜을 통해 정적으로 버퍼를 할당하는 방법으로 네트워크의 상황에 따른 버퍼 활용도를 예측할 수 없는 단점이 있다. 이러한 단점을 보완하여 진보된 TCP 버퍼할당 기법으로 ATBT(Automatic TCP Buffer Tuning)가 제안되었다 [3]. ATBT는 TCP의 혼잡윈도우 크기에 따라 전송단의 소켓 버퍼를 동적으로 할당하는 기법으로 전송환경 변화에 따른 유연한 버퍼를 제공한다. 그러나 ATBT는 패킷손실로 인하여 혼잡윈도우 크기가 빈번히 발생하는 경우 버퍼의 크기 변화가 심하며 버퍼를 재분배하는 과정에

서 다수의 플로우 사이에 불공정성 문제가 발생한다.

본 논문에서는 종단 호스트에서 다수의 플로우에 대해 공정하고 높은 전송 성능을 보장하기 위한 새로운 TCP 버퍼할당 기법으로 RTBA를 제안하였다. RTBA에서는 RTT, RTO, 패킷 손실률 등을 고려한 기대전송율을 기반으로 네트워크 상황에 적응적이면서도 메모리 자원을 효율적으로 사용할 수 있는 최적의 버퍼크기를 할당한다. 또한 기대전송율에 비례하여 버퍼를 재분배하여 기존의 ATBT기법에서 나타나는 플로우간의 불공정성 문제를 개선하였다.

논문의 2장에서는 TCP의 윈도우기반 전송메커니즘과 ATBT의 버퍼할당 기법에 대한 소개와 함께 발생 가능한 문제점들에 대해 지적하였으며 3장에서는 제안한 RTBA 기법의 세부 알고리즘을 소개하였다. 4장에서는 실험을 통해 제안한 알고리즘의 성능을 평가하였으며 마지막으로 5장에서는 결론을 맺었다.

### 2. 관련 연구

#### 2.1 TCP의 윈도우 기반 전송 메커니즘

TCP에서 사용하는 윈도우는 ACK를 받지 않고 한번에 전송할 수 있는 데이터의 양을 의미하며 소켓 버퍼 영역에서 슬라이딩 윈도우 메커니즘을 사용하여 관리된다. 하지만 전송링크의 용량에 비해 운영체제로부터 할당받은 송신단의 버퍼크기가 너무 작은 경우, 송신단은 네트워크에서의 혼잡상황이나 수신단의 버퍼크기와 상관없이 버퍼크기에 의해 전송율에 제한을 받게 된다. 반대로 전송링크의 용량에 비해 버퍼크기가 너무 크게 할당된 경우, 버퍼공간에 비해 실제 전송에 사용되는 윈도우크기가 작으므로 버퍼가 낭비된다. 메모리 자원이 제한적인 임베디드 시스템에서 이러한 과도한 버퍼할당은 처리 가능한 플로우의 수에 제한을 주거나 자칫 메모리 고갈로 인한 시스템 불안정 등의 문제를 발생시킬 수 있다.

기존의 연구들을 통해, TCP에 할당되는 최적의 버퍼크기는 TCP의 성능에 영향을 주지 않는 최소의 버퍼크기라고 정의할 수 있다. 따라서 TCP의 성능을 최대한 보장하고, 시스템의 메모리 자원을 효율적으로 사용하기 위해서는 여러 가지 전송환경을 고려한 최적의 버퍼크기를 할당해야만 한다.

\* 본 논문은 유비쿼터스 컴퓨팅 프론티어 사업단의 지원에 의한 연구결과입니다.

### 2.2 ATBT 버퍼할당 기법

ATBT의 목적은 다수의 플로우를 처리하는 서버에서 플로우 간에 공정한 버퍼를 제공하는 것이다. ATBT 기법에서는 버퍼크기를 전송경로의 BDP를 나타내는 혼잡윈도우의 크기에 따라 결정한다. 즉 혼잡윈도우의 크기가 커짐에 따라 플로우에 더 큰 버퍼를 할당하고, 혼잡윈도우 크기가 줄어들면 버퍼할당량을 감소시킨다.

ATBT의 알고리즘은 크게 혼잡윈도우 크기에 따라 기대 버퍼크기를 계산하는 과정과 할당된 버퍼크기가 전체 버퍼크기를 초과하였을 경우에 수행되는 재분배 과정으로 나눌 수 있다. 혼잡윈도우의 변화는 전송링크의 용량과 관계되므로 ATBT에서는 혼잡윈도우 값을 기대 버퍼크기에 직접 반영한다. 그러나 기대 버퍼크기가 혼잡윈도우에 따라 급격하게 변화하는 것을 방지하기 위해 임계값을 두어 기대 버퍼크기를 관리하며 이로 인해 기대 버퍼크기는 혼잡윈도우의 2배에서 4배 정도의 범위 내에서 변화한다.

플로우들에 할당될 기대 버퍼크기가 제한된 크기의 메모리를 초과하는 경우, ATBT에서는 최대-최소 공정분배 원칙에 따라 할당된 버퍼크기를 공정하게 재분배한다. 이러한 재분배는 500ms마다 호출되는 tcp\_slowtimer() 함수 내에서 일어나는데, 이는 모든 플로우에 할당된 버퍼크기를 검사해야 하는 재분배 과정이 너무 자주 일어남으로써 발생할 수 있는 오버헤드를 줄이면서 동시에 만족할 만한 공정성을 유지하기 위함이다.

ATBT에서는 플로우에 버퍼를 오직 현재의 혼잡윈도우의 크기만을 고려하여 할당한다. 따라서 중복된 ACK나 Timeout과 같은 패킷손실이 발생하는 경우 할당되는 버퍼의 크기는 급격히 줄어들게 되고 심한 경우 실제로 그 플로우가 필요로 하는 크기보다 작게 할당될 수도 있다. 이로 인해 윈도우 크기의 변화폭이 커지기 때문에 할당되는 버퍼크기의 변화가 큰 폭으로 빈번히 발생하는 문제가 발생한다. 이러한 문제들은 버퍼의 재분배 주기를 짧게 하여 좀 더 빈번하게 공정한 버퍼크기를 계산함으로써 해결할 수도 있지만, 이는 버퍼를 할당하는데 많은 오버헤드를 유발하기 때문에 근본적인 해결책은 될 수 없다. ATBT의 또 다른 문제점은 플로우간의 불공정성 문제이다. ATBT의 최대-최소 공정 분배는 플로우들의 전체 버퍼요구량이 소켓 버퍼를 초과할 때 수행된다. 먼저 플로우들의 최소 요구량을 공급하고 나머지 버퍼공간을 각각의 플로우들에게 같은 크기로 재분배하므로 큰 버퍼를 요구하는 플로우들은 상대적으로 작은 버퍼를 할당받게 되는 불공정성 문제가 발생한다.

## 3. RTBA 알고리즘

### 3.1 RTBA 개요

본 논문에서는 ATBT와 같이 간단하면서도 네트워크의 상태를 충분히 고려하기 위해 플로우의 기대전송율을 기반으로 버퍼를 할당하는 RTBA 기법을 제안하였다. RTBA는 응용에서의 부가적인 지원을 필요로 하지 않을 뿐만 아니라 TCP에서 사용하는 대부분의 정보를 그대로 이용하여 BDP측정을 위한 오버헤드를 최소화 하였다. 또한 전송율 모델을 기반으로 가용 대역폭을 예측하는 RTBA는 혼잡윈도우를 버퍼크기에 직접 적용하는 ATBT에 비해 버퍼의 변화가 안정적이므로 플로우 사이의 불공정성 문제를 개선할 수 있다.

RTBA에서 플로우에 버퍼를 할당하는 과정은 ATBT와 유사한 두 단계를 통해 이루어진다. 즉 RTBA는 기대전송율을 기반으로 플로우가 요구하는 버퍼크기를 계산하는 과정과, 할당되는 버퍼크기가 전체 버퍼공간을 초과하는 경우에 재분배하는 과정으로 구성된다.

### 3.2 버퍼할당 및 재분배

Padhye는 평형상태에 있는 TCP의 전송율  $R$ 을 혼잡제어 매

커리즘의 수학적 분석을 통해 식 (1)과 같이 RTT와 RTO, 패킷손실률 등의 함수로 표현하였고, 실험을 통해 그 정확성을 입증하였다 [4].

$$R = \min \left[ \frac{W_m \cdot s}{t_{RTT}}, \frac{s}{t_{RTT} \sqrt{\frac{2bp}{3}} + t_{RTO} \min \left( 1, 3 \sqrt{\frac{3bp}{8}} \right) p(1 + 32p^2)} \right] \quad (1)$$

RTBA에서는 매 RTT마다 식 (1)의 전송율 모델을 통해 기대 전송율을 유추한다. RTT와 RTO 그리고 최대 윈도우 크기  $W_m$  등 사용되는 변수들은 TCP의 변수를 그대로 사용하므로 비교적 간단하게 기대전송율을 구할 수 있다.  $s$ 는 세크먼트의 크기를 나타내며,  $b$ 는 지연된 ACK를 고려한 변수로서 하나의 ACK에 의해 응답 확인된 패킷의 수를 나타낸다.

파라미터 값을 통해 얻어진 기대전송율은 할당되는 버퍼크기에 직접적인 영향을 미치므로 버퍼할당량의 빈번한 변화를 줄이기 위해 RTBA에서는 EWMA(Exponentially Weighted Moving Average) 기법을 통해 기대전송율을 정규화한다.  $R_i$ 를 플로우  $i$ 의 평균전송율이라고,  $R_{now}$ 를 현재 RTT에서의 전송율이라 하면, 플로우  $i$ 의 기대전송율은 식 (2)과 같이 결정된다.

$$R_i = (1 - w) \cdot R_i + w \cdot R_{now} \quad (2)$$

따라서 플로우  $i$ 가 필요로 하는 버퍼의 크기  $S_i$ 는 식 (2)에서 구해진 기대전송율과 전송경로상의 왕복지연시간의 곱으로 식 (3)과 같이 표현할 수 있다.

$$S_i = R_i \times RTT_i \quad (3)$$

또한 RTBA에서는 전송율 측정상의 오차로 인한 TCP의 저속 증가구간(Slow-Start)에서의 버퍼공간 부족 문제를 방지하기 위해 혼잡윈도우의 크기 변화를 동시에 고려하였다. 이 구간에서는 혼잡윈도우 크기의 두배 정도로 충분한 버퍼크기를 제공함으로써 버퍼부족으로 인한 전송율 감소문제를 최소화하였다. 이는 저속증가구간에서 윈도우 증가폭이 TCP의 전체 전송율에 큰 영향을 미치기 때문이다. 위와 같은 요구조건을 종합하여 플로우에 할당하는 초기 버퍼크기는 식 (4)와 같이 결정된다.

$$Buffer\ Size = MAX(S_i, 2 \times cwnd) \quad (4)$$

RTBA에서의 버퍼 재분배는 ATBT에서와 마찬가지로 할당되는 버퍼크기가 최대 버퍼크기를 초과하는 것을 방지하면서, 동시에 플로우들 간에 공정한 버퍼 분배를 위한 과정이다. 그러나 ATBT의 경우와 같이 최대-최소 공정 분배원칙에 따라 버퍼를 재분배할 경우 나타나는 플로우 사이의 불공정성 문제를 해결하기 위해 RTBA에서는 각 플로우의 기대전송율에 따라 비례적으로 재분배하였다. 그림 1과 같이 특정 플로우의 공정한 버퍼크기는 전체 할당된 버퍼크기 중 특정 플로우가 차지하는 비율에 따라 가용한 버퍼크기를 재분배함으로써 부족한 버퍼크기로 인해 특정 플로우가 전송율에 심각한 손해를 보는 불공정성 문제를 해결하였다.

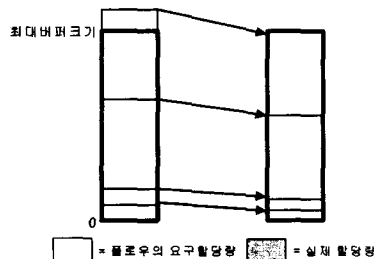


그림 1. RTBA에서 초과 버퍼크기의 공정한 재분배

4. 실험 및 성능 평가

4.1 실험 환경

제안한 RTBA의 성능 평가를 위해 lwTCP(LightWeight TCP)를 기반으로 알고리즘을 구현하였고 그림 2와 같은 간단한 실험망을 구성하여 실험하였다 [5]. 라우터 R1에서는 NIST Net을 사용하여 서버와 클라이언트 사이의 대역폭, 지연시간, 패킷손실율 등을 설정하였으며 라우터 R2에는 300KB의 Drop-tail 큐를 사용하였다 [6]. 테스트 서버로는 펜티엄4 PC에 RTBA를 기반으로 한 FTP서버를 구현하였으며, 별도의 변수를 두어 필요한 경우 할당 가능한 총 버퍼크기를 제한하였다.

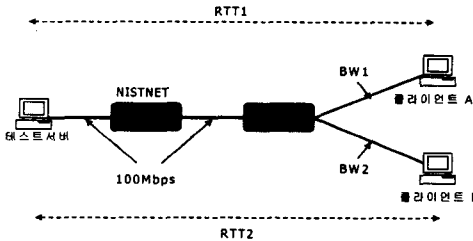


그림 2. 실험 환경

4.2 효율성 및 공정성 실험

그림 3은 RTBA의 동적인 버퍼할당의 효율성에 대한 실험 결과를 보여주고 있다. 실험을 위해 그림 2의 실험환경에서 BW1는 50Mbps, RTT1은 1ms로 설정하였고, RTBA를 기반으로 구현한 테스트 서버에서 클라이언트 A로 10MB의 파일을 전송하여 시간에 따른 버퍼할당량의 변화를 측정하였다. RTBA를 적용하였을 때의 전송율은 평균 49KB/s 정도로 나타났으며 이는 50Mbps, 1ms의 환경에서 얻을 수 있는 최대 전송율에 가까운 수치이다. 또한 RTBA가 할당한 버퍼의 크기는 평균 20KB정도로 최적의 버퍼크기에 가까운 버퍼를 할당하였으며, 버퍼크기 변화도 안정적으로 이루어지고 있음을 알 수 있다.

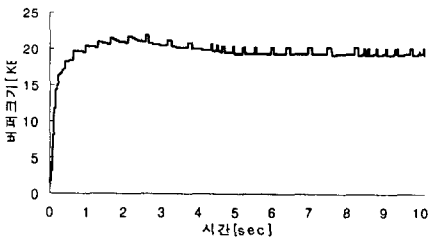
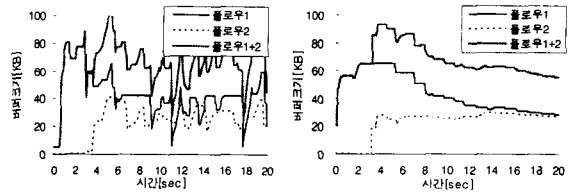


그림 3. 시간에 따른 버퍼할당량 변화

RTBA는 기존의 TCP에서 부족한 버퍼로 인해 나타나는 저조한 전송율을 최고 136% 향상시켰으며, 320KB로 과도한 버퍼를 할당한 경우에 비해 비슷한 전송율을 보이면서 버퍼할당량은 16% 감소하였다.

그림 4는 10ms, 20Mbps의 동일한 환경에서 두 개의 플로우 사이에 할당되는 버퍼크기 비교를 통해 RTBA의 공정성 실험을 하였다. ATBT는 혼잡원도우에 따라 버퍼의 크기를 결정하기 때문에 할당되는 버퍼의 변화 또한 혼잡원도우의 크기 변화와 유사하게 나타난다. 그러나 결과를 통해 두 플로우 모두 버퍼할당량이 심하게 변하는 것을 볼 수 있으며 이러한 변화는 버퍼공간이 부족한 경우 재분배 과정에서 발생하는 불공정성의 원인이 된다. 반면 RTBA는 ATBT에 비해 버퍼할당량의 변화가 매우 안정적으로 이루어지고 있음을 볼 수 있다. 플로우1에 할당되는 버퍼의 크기는 약 3초 후 클라이언트 B로 전송되는

플로우2에 의해 줄어든 대역폭에 반응하여 점차 줄어들어 약 12초 후에는 플로우2와 동일한 크기를 할당받게 된다. ATBT에 비해 다소 느리게 반응하는 이유는 트래픽의 증가로 인한 패킷손실의 영향이 혼잡원도우에 비해 기대전송율에서는 덜 민감하게 나타나기 때문이다.



(a) ATBT의 경우 (b) RTBA의 경우

그림 4. 플로우 사이의 공정성 실험

그림 3과 그림 4의 실험 결과를 통해 RTBA가 플로우에 최적의 버퍼크기를 할당하며, 동일한 조건의 플로우들이 경쟁하는 상황에서도 각각의 플로우들에 공정하게 버퍼를 할당하는 것을 확인할 수 있다.

5. 결론

현재 인터넷의 주요한 전송 프로토콜로 사용되고 있는 TCP는 전송링크의 대역폭이 충분함에도 불구하고 비효율적인 버퍼할당 기법으로 인한 전송율에 제한을 받는 현상이 발생한다. 이러한 문제를 해결하기 위해 네트워크 환경에 적응적인 버퍼할당 기법이 필요하게 되었으며, ATBT에서는 혼잡원도우의 크기에 따라 버퍼를 할당하는 기법을 사용하였다. 그러나 ATBT는 할당되는 버퍼크기의 변화가 심하고, 플로우간에 불공정한 버퍼가 할당되는 문제를 가지고 있다.

본 논문에서는 이러한 ATBT의 문제점을 개선하여 다수의 플로우에 대해 효율적이고 공정한 버퍼할당 기법인 RTBA를 제안하였다. ATBT가 단지 혼잡원도우 크기에만 의존하는데 반해 RTBA는 RTT, RTO, 패킷손실율과 같은 여러 가지 네트워크 환경을 고려한 기대전송율에 기반을 두고 있다. 따라서 RTBA는 혼잡원도우가 빈번히 변화하는 경우에도 안정적으로 버퍼를 할당할 수 있으며 전송율에 비례하여 버퍼를 재분배함으로써 특정 플로우의 전송율에 제약을 받는 불공정성 문제를 해결하였다. 실험을 통해 RTBA의 성능을 분석하였고, 그 결과로 효율적이고 공정한 버퍼할당이 이루어짐을 확인하였다.

향후 과제로는 실험 환경을 확장하여 다양한 환경에서 RTBA에 대한 성능 측정과 함께, 빠르게 최적의 버퍼크기를 찾아내는 방법에 대한 연구가 수행되어야 할 것이다.

참고 문헌

- [1] V. Jacobson and M. Karels, "Congestion Avoidance and Control," Proceedings of SIGCOMM, August 1988.
- [2] B. Tierney, "TCP Tuning Guide for Distributed Applications on Wide-Area Networks," USENIX and SAGE Login, February 2001.
- [3] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," Proceedings of SIGCOMM, October 1998.
- [4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," Proceedings of SIGCOMM, August 1998.
- [5] lwTCP (LightWeight TCP/IP stack), <http://www.sics.se/~adam/lwip/>
- [6] NIST Net, <http://snad.nsl.nist.gov/itg/nistnet/index.html>