

## 허용범위내 가용대역측정값을 가지는 새로운 무선 TCP-Westwood 알고리즘에 대한 연구

유창열\*, 김동회\*\*

### 요약

높은 오류율을 가지는 무선 TCP(Transmission Control Protocol) 환경을 위한 TCP-Westwood 방법에 대한 연구가 많이 진행되었다. 이러한 TCP-Westwood 방법은 가용대역폭측정(Available Bandwidth Estimation)에 의해서 혼잡윈도우를 조절하는 방법이기 때문에 오류율이 높아질수록 이 가용대역폭측정값의 정확도가 떨어지는 문제점이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 ACK의 도착간격이 주어진 시간 간격을 벗어나는 경우에 ABE값을 무시하는 기존의 패킷 패턴 기반 가용대역폭필터링 방법과 가용대역폭측정값이 허용 가능한 범위를 벗어나는 경우에 주어진 허용 가능한 가용대역폭측정값으로 할당하는 새로운 방법을 모두 사용하였다. 제안한 방법의 실험결과를 기존의 다른 알고리즘들보다 가용대역폭측정값이 설정한 시뮬레이션 대역폭에 가장 유사함을 확인하였다.

키워드 : 무선 TCP, TCP-Westwood 방법, 가용대역폭측정, 혼잡윈도우, 슬로우스타트역치

## The Study on New Wireless TCP-Westwood Algorithm having Available Bandwidth Estimation within Allowable Range

Chang-Yeol Yoo\*, Dong-Hoi Kim\*\*

### Abstract

There have been many researches for TCP-Westwood algorithms in wireless TCP environment with high packet loss rate. Because the TCP-Westwood algorithm adjusts the congestion window according to the ABE(Available Bandwidth Estimation), the algorithm has a problem which the accuracy of ABE decreases as the error rate increases. To solve such a problem, the proposed scheme in this paper adopts the existing packet pattern based algorithm that the ABE is ignored when the arriving interval time of ACK is longer than a given interval time and uses new algorithm that the ABE is reallocated to a given allowable ABE when the ABE is over the allowable range. The proposed scheme shows the simulation result that the ABE is closest to the setting bandwidth for simulation compared to the existing algorithms.

Keywords : Wireless TCP, TCP-Westwood Algorithm, Available Bandwidth Estimation, Congestion Window, Slow Start Threshold

### 1. 서론

현재 사용되는 인터넷은 TCP/IP라 불리는 인

터넷 프로토콜의 안정성에 기인한다. 오류율이  $10^{-7}$  정도로 낮은 유선망에서의 TCP는 인터넷에서 신뢰할 수 있는 성능을 보여주고 있다. 그러나 TCP는 유선망을 기반으로 만들어졌기 때문에 오류율이  $10^{-3}$  이상으로 유선망에 비해 상당히 높은 무선망에서는 유선망에서처럼 신뢰할 수 있는 성능을 보여주지 못하고 있다. 따라서 높은 오류율을 가지는 무선 환경에 적합한 TCP가 필요하게 되어 이러한 방법들로는 크게 세 가지가 있다. 첫 번째 방법으로는 Snoop TCP

※ 교신저자(Corresponding Author): Dong-Hoi Kim  
접수일:2013년 12월 05일, 수정일:2014년 02월 11일  
완료일:2014년 03월 08일

\* 강원대학교 IT대학 전자통신공학과  
Tel: +82-33-250-6349, Fax: +82-33-259-5676  
email: donghk@kangwon.ac.kr

\*\* 강원대학교 IT대학 전자통신공학과

또는 Indirect TCP와 같이 무선링크의 패킷손실을 중간 기지국에서 인지하여 재전송함으로써, 무선링크에서 발생하는 패킷손실을 TCP 송신자가 인지하지 못하도록 하는 방법이다[1][2][3]. 두 번째 방법으로는 패킷손실을 유선 네트워크 혼잡에 의한 패킷손실과 무선링크에 의한 패킷손실로 구분하여 그에 따라 대처하는 방법이 있다. 이 방법은 패킷간의 전송시간을 모니터링 하여 구분해낸다[4][5]. 세 번째 방법은 평균적인 네트워크 가용대역폭을 측정하여 패킷손실이 발생하였을 경우 혼잡윈도우(congestion window)의 크기를 측정된 대역폭으로 설정하여 주는 방법이다. 이 방법은 두 번째 방법이 패킷손실을 구분하는 정확도에 문제가 있어서 나온 방법이다. 이 방법은 패킷손실을 구분 할 필요가 없어 단순해지지만, 무선네트워크의 대역폭을 충분히 사용하려면 대역폭을 정확히 측정 할 수 있어야 한다는 점에 유의해야한다. 본 논문에서 다루는 내용은 위에서 설명한 세 번째 방법을 사용하여 유/무선상의 패킷손실에 관계없이 혼잡윈도우를 결정하는 TCP-Westwood 방식을 기반으로 한다[6][7][8]. TCP-Westwood 방식은 무선 링크의 특성중의 하나인 사용자의 이동성에 영향을 받지 않고, 네트워크의 도움을 필요로 하지 않는 점에서 기존의 다른 방법들에 비해 장점이 있다. 그러나 TCP-Westwood 방식은 1%미만의 낮은 패킷손실률에서는 기존의 유선기반 TCP들에 비해 나은 성능을 보이지만 1%이상 높은 패킷손실률에서는 성능의 향상이 미미하다. 이러한 문제점을 패킷들의 송수신 패턴 측면에서 대역폭의 잘못된 측정을 원인으로 보고, ACK의 도착 시간의 평균값 및 분산을 이용하여 패킷 도착 간격이 주어진 값( $T_{\text{thresh}}$ )를 초과하는 경우를 해당 B(bandwidth: 현재대역폭)을 ABE(available bandwidth estimation:가용대역폭측정값)에 반영하지 않는 패킷 패턴 기반 알고리즘이 제안되었다[9]. 그러나 이 방식도 아직 1%이상의 높은 패킷손실률에서는 정확한 성능을 보여주지 못하는 단점이 있다. 따라서 본 논문에서는 이러한 TCP-Westwood를 보다 더 효율적이고 정확하게 성능을 향상시키기 위해 기존의 패킷 패턴 기반 알고리즘에서의  $T_{\text{thresh}}$ 를 사용하는 방법과 허용범위를 벗어난 ABE를 허용 가능한 값으로 제한당하는 새로운 방법을 모두 사용하여 가용

대역폭을 최대한으로 정확하게 활용할 수 있는 새로운 방법을 모두 사용하는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 TCP-Westwood에 대해 설명하고, 이를 개선한 기존 패킷 패턴 기반 알고리즘을 기술한다. 또한 3장에서는 본 논문에서 제안하는 알고리즘에 대한 설명을 기술한다. 4장과 5장에서는 결과 확인을 위해 사용된 시뮬레이션환경과 시뮬레이션 결과를 기술하였고 6장에서는 결론을 말한다.

## 2. 기존 알고리즘 설명

### 2.1 기존 TCP-Westwood 알고리즘

패킷 손실 구분의 정확도가 떨어질 수 있다는 단점을 회피하면서 하부 네트워크의 도움 없이 TCP 자체만의 수정으로 무선 링크에서 성능을 향상시키기 위해 ABE를 측정하여 사용하는 방법이 제안되었다. TCP-Westwood는 ACK를 수신할 때마다 ABE를 측정 한다. 그리고 패킷 손실 발생시 혼잡윈도우와 슬로우스타트역치(slow start threshold)를 ABE으로 설정한다. TCP-Westwood의 ABE를 측정하는 알고리즘은 패킷 페어(packet pair) 알고리즘에 기반을 두고 있다[10]. 병목현상이 발생하는 라우터에 의해 두 패킷의 도착 시간에 차이가 발생하게 되고 이 시간을 이용하여 대역폭을 측정하는 것이 패킷 페어 알고리즘의 기본 아이디어이다. TCP-Westwood 는 ACK를 매번 수신 할 때마다 다음과 같은 방법으로 B와 ABE를 계산한다.

$$\begin{aligned} B &= ACK_i / (T_i - T_{i-1}) \\ ABE &= (1-a)ABE + a*B \end{aligned} \quad (1)$$

여기서 B는 현재 ABE값으로 ACK간의 시간차를 이용하여, 단위 시간당 얼마만큼의 데이터가 수신되었는지 측정하여 계산할 수 있다. 식 (1)에서  $ACK_i$ 는 i번째 ACK가 응답하는 데이터 바이트를 나타낸다.  $T_i$ 는 i번째 ACK가 도착한 시간을 나타낸다. 이렇게 측정된 B값은 i번째 ACK가 수신될 때 측정되는 현재의 가용대역폭이다. 또한 a는 0과 1사이의 값으로 이동 평균

(moving average)을 위한 파라미터 값을 나타낸다. 즉, ABE는 ACK를 수신 할 때마다 현재의 가용대역폭을 측정하여 이동 평균 방법을 이용하여 그 값들의 평균값을 계산한 평균대역폭을 측정하는 것이다. 위와 같이 측정된 ABE는 다음과 같이 사용한다.

패킷손실시 : 혼잡윈도우=ABE  
 슬로우스타트역치=ABE  
 타임아웃시 : 혼잡윈도우=1  
 슬로우스타트역치=ABE

위와 같이 패킷 손실이 일어났을 때는 혼잡윈도우와 슬로우스타트역치를 측정된 ABE값으로 설정한다. 기존의 유선망 기반 TCP는 TCP에서 무조건 혼잡윈도우를 기존 혼잡윈도우의 절반으로 감소시키지만, TCP-Westwood에서는 ABE값으로 혼잡윈도우를 설정해주기 때문에 훨씬 더 효율적이라 볼 수 있다. 타임아웃의 경우에 네트워크 상태가 매우 좋지 않아 일어나는 경우 이므로 기존의 TCP와 같이 혼잡윈도우 값을 1로 설정하고, 슬로우스타트역치는 ABE값으로 설정해준다.

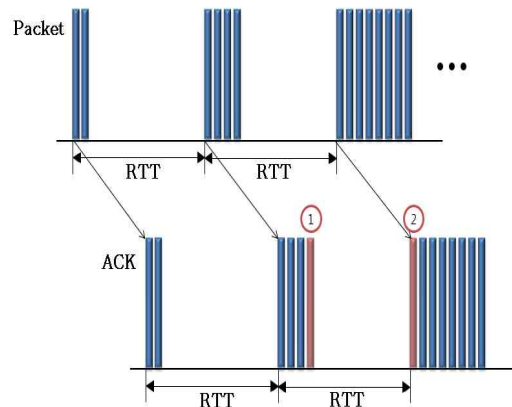
서론에서 말했듯이 TCP-Westwood가 무선네트워크의 대역폭을 충분히 사용하려면, ABE를 정확히 측정할 수 있어야 한다. 그러나 위에서 설명한 기존의 TCP-Westwood의 방법으로는 패킷손실률이 1% 이상이 될 경우에 ABE의 정확도가 심각하게 떨어지는 현상이 발생한다. 이러한 부정확한 값은 TCP-Westwood가 무선 링크에서 제공되는 대역폭을 충분히 사용하지 못하게 된다.

## 2.2 기존 TCP 패킷 패턴 기반 TCP-Westwood 알고리즘

기존 TCP-Westwood 방식에서의 패킷손실률이 높을 때 ABE값이 부정확하게 측정되는 문제를 해결하기 위한 한 가지 방법으로 TCP의 패킷 송수신의 패턴을 고려하여 향상된 대역폭 측정방법을 구하는 TCP 패킷 패턴 기반 알고리즘이 제안되었다. 이 알고리즘은 TCP의 패킷 송수신의 패턴을 분석하였을 때 ACK간의 시간차가 큰 경우에는 B값이 잘못된 값으로 측정될 수 있기 때문에 ACK간의 시간차가 어떤 기준값이상

인 경우에 무시하는 방법을 제안하였다. (그림 1)은 패킷손실이 발생하지 않을 때 TCP송신자가 어떠한 패턴으로 패킷을 전송하고, TCP수신자가 어떤 패턴으로 ACK를 수신하는지를 시간의 흐름에 따라 보여주고 있다.

(그림 1) TCP의 패킷 송신 패턴 및 ACK의 수신패턴



(Figure 1) The packet sending pattern and ACK receiving pattern in TCP operation

그림을 보면 패킷과 ACK들은 TCP의 슬로우스타트 알고리즘[10][11]에 따라 혼잡윈도우 크기 안에서 촘촘히 패킷들을 전송하고, 그에 따른 ACK들도 촘촘히 전송되고 있다. 하나의 윈도우사이클 주기와 주기사이에는 왕복시간(RTT : Round Trip Time)만큼의 시간간격이 있고, 혼잡윈도우는 슬로우스타트 알고리즘에 따라 지수적으로 증가하고 있다. 이러한 균집한 트래픽 특징은 B값 계산에 부정확한 요인까지 포함시킨다. 2.1절에서 언급했다시피 B값을 계산하는 부분에 사용되는 패킷 페어 알고리즘은 두 개의 패킷을 동시에 보내어 두 개의 패킷이 얼마만큼 떨어져서 도착하는 지를 측정한다. 그러므로 정확하게 현재의 가용대역폭인 B값이 계산되기 위해선 두 패킷이 동시에 오거나, 매우 인접하여 전송되어야 한다는 전제조건이 필요하다. 그러나 위에서 언급한 TCP의 균집한 트래픽 특징 때문에 문제가 발생하게 된다.

하나의 혼잡윈도우사이클 안에서의 패킷들은 촘촘히 보내지기 때문에 B값 계산에 문제가 없다. 그러나 그림에서 ①, ②로 표시된 ACK들처럼

하나의 주기와 주기사이의 경계에 위치하는 패킷들은 처음부터 거의 RTT 간격으로 송신된다. 이 경우에 B값 계산에 사용되어지는 ACK간의 시간차  $T_i - T_{i-1}$ 의 값이 거의 RTT의 값만큼 큰 값으로 측정이 되어 본래의 패킷 페어알고리즘의 전제조건을 만족시키지 못한다. 따라서 기존의 TCP 패킷 패턴 기반 알고리즘은 이러한 이유로 B값을 제대로 측정하기 위해  $T_i - T_{i-1}$  값이 일정한 값을 초과할 경우에 구한 B를 ABE 계산에 넣지 않는 알고리즘이다. 다음은 기존의 TCP 패킷 패턴 기반 알고리즘을 설명하는 의사 코드(pseudocode)를 보여주고 있다.

*i*번째 ACK가도착한 경우

$$T = T_i - T_{i-1} \quad (2)$$

$$T_{average} = (1-g) \cdot T_{average} + g \cdot T \quad (3)$$

$$T_d = (1-h) \cdot T_d + h \cdot |T_{average} - T| \quad (4)$$

$$T_{thresh} = T_{average} + 3T_d \quad (5)$$

if  $T > T_{thresh}$  return;

else

$$B = ACK_i / (T_i - T_{i-1})$$

$$ABE = (1-\alpha)ABE + \alpha B$$

이 알고리즘은 ACK간의 도착간격이 패킷의 전송간격과 유사하다는 점에 착안하여, ACK의 도착시간의 평균값 및 분산을 이용하여 왕복시간계산식과 동일하게 B값 필터링을 위한 문턱치값을 결정한다. 식 (2)은 *i*번째 ACK의 도착간격을 의미하며, 식 (3)은 이러한 도착간격의 평균값을 이동 평균 방식을 이용하여 계산한다. 식 (4)은 도착간격의 표준편차를 계산하고, 계산된 평균과 표준편차를 이용하여, 식 (5)를 계산, B값 필터링을 위한 기준값  $T_{thresh}$ 로 사용한다. ACK의 도착간격  $T$ 가  $T_{thresh}$ 보다 작을 경우에만 적절한 B값으로 추정하고 ABE 계산에 사용한다.

### 3. 제안하는 알고리즘

본 논문에서 제안하는 방법은 패킷 패턴 시간 기반 TCP-Westwood 알고리즘에서 제안한 방법을 반영하고 추가적으로 현재 측정된 ABE이 값이 허용 가능한 ABE값을 벗어나는 값인 경우

에 주어진 허용 가능한 ABE값으로 제한당하는 방법이다. 다음은 제안하는 방법을 설명하는 의사 코드(pseudocode)를 보여주고 있다.

*i* 번째 ACK가도착한 경우

$$B = ACK_i / (T_i - T_{i-1}) \quad (6)$$

$$\text{if}((B \neq 0) \&\&(ABE_{min} \neq 0)) \{ \quad (7)$$

$$\text{if}(B < ABE_{min}) \{ \quad (8)$$

$$B = ABE_{min};$$

}

$$\text{else if}(B > ABE_{max}) \{ \quad (9)$$

$$B = ABE_{max};$$

}

}

$$T = T_i - T_{i-1} \quad (10)$$

$$T_{average} = (1-g) \cdot T_{average} + g \cdot T$$

$$T_d = (1-h) \cdot T_d + h \cdot |T_{average} - T|$$

$$T_{thresh} = T_{average} + 3T_d$$

if  $T > T_{thresh}$  return;

else{

$$ABE_i = (1-\alpha)ABE_{i-1} + \alpha \cdot B \quad (11)$$

$$ABE_{last} = ABE_i; \quad (12)$$

}

$$ABE_{min} = ABE_{last} * C_{min}; \quad (13)$$

$$ABE_{max} = ABE_{last} * C_{max}; \quad (14)$$

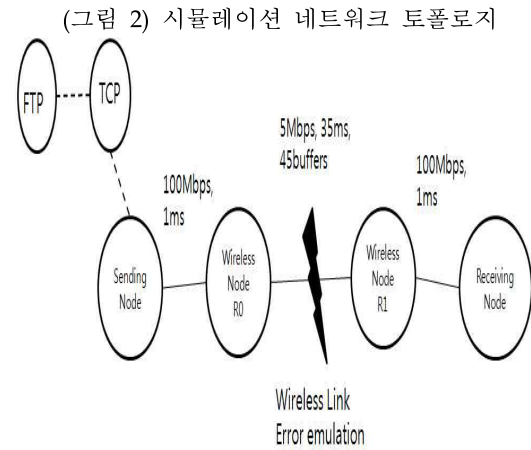
식 (6)에서 *i*번째 ACK가 도착했을 때, 기존의 TCP-Westwood의 방법과 같이 B값을 계산한다. 또한 식 (7)에선 B값이 0이 아니면서, 새로운 필터값이 된  $ABE_{min}$  또는  $ABE_{max}$ 가 0이 아닌 경우, 즉, TCP 통신이 맨 처음 새로 시작되는 경우를 제외한 경우에 if문의 조건에 성립되어 필터링을 시작하게 된다. 여기에서  $ABE_{min}$ 와  $ABE_{max}$ 는 Default값을 0으로 설정해 놓아서 TCP가 사용되지 않을 경우는 0의 값을 갖는다. 위와 같이 필터링을 하여 새로운 B값을 설정한 후 식 (10)을 수행한다. 식 (10)부터 else문 이전까지는 기존의 패킷 패턴 기반 알고리즘을 사용한다. 식 (11)에서는 필터링된 B값을 이용하여  $ABE_i$ 를 계산한다. 식 (12)에선 그렇게 계산된  $ABE_i$ 를  $ABE_{last}$ 에 저장한다. 그 후 식 (13)과 식 (14)에서  $ABE_{min}$ ,  $ABE_{max}$ 의 값을  $ABE_{last}$  값의 최소 허용치( $C_{min}$ ), 최대 허용치

( $C_{max}$ )를 곱한 값으로 설정해준다. 이렇게 설정된  $ABE_{min}$ ,  $ABE_{max}$ 는 식 (8)와 식 (9)에서 B값을 필터링을 위한 필터값이 된다. 이렇게 하여 항상 새로운 ACK가 도착할 때의 B값이 이전에 지금까지의 ABE값 즉,  $ABE_{i-1}$ 과 비교하여 심하게 낮아지거나 심하게 높게 측정되는 경우(즉,  $ABE_{min}$ 보다 낮거나  $ABE_{max}$ 보다 크게 되는 경우)에  $ABE_{min}$ 와  $ABE_{max}$ 값으로 재할당하여 필터링을 하게 된다. 여기에서 필터값  $ABE_{min}$ 와  $ABE_{max}$ 는 항상 현재 도착한 패킷의 바로 전 패킷 즉,  $ACK_{i-1}$ 까지의 ABE값을 기반으로 설정된다. 이러한 필터값은 불규칙적으로 발생하는 비정상적인 B를 무시함으로써 정확한 ABE를 설정할 수 있게 된다.

#### 4. 시뮬레이션 환경

(그림 2)는 본 논문에서 사용되는 네트워크 시뮬레이션 모델이다. 네트워크 시뮬레이터는 Ns-2를 사용하였다[12][13]. 시뮬레이션은 송신노드와 수신노드를 가지고 있으며, 각 노드는 무선노드와 100Mbps의 링크용량과 1ms의 지연시간으로 연결되어있다. 무선노드 R0과 R1은 5Mbps의 무선링크용량을 가지고 있고 35ms의 지연을 가진다. 버퍼크기는 45로 설정하였다. R0에서 확률 p(랜덤패킷손실률)로 패킷을 버림으로써 무선 링크 오류를 에뮬레이션하고 있다. TCP를 송신노드에 장착시키고, 기본적인 FTP를 TCP에 사용하였다. 확인을 위해 수신노드에 TCP를 설정을 하였다. 패킷크기는 1400byte로 설정하였고 패킷크기의 default값은 50이다. 100초 동안 시뮬레이션을 실행하였다. ABE를 구하기 위한  $\alpha$  값은 0.9를 사용하였다. 그 이유는 이동평균의 원리에서 찾을 수 있다.  $ABE=(1-\alpha)ABE+\alpha*B$  식에서  $\alpha$  값이 1에 가까울수록 현재 측정된 가용대역폭인 B값에 비중을 높게 두어 평균값을 계산하게 된다. 만일  $\alpha$  값을 0.1 같이 주어서 현재 B값이 아닌 지금까지 측정된 ABE값에 비중을 많이 두게 된다면, 결과적으로 시간이 지날수록 알고리즘에 따라 비슷한 값을 측정하게 되나,  $ABE_{last}$  값이 천천히 증가하게 되어, 처음 TCP를 이용 할 때 정상적인 ABE를 측정하는데 시간이 걸리게 된다. 그 이유는 평균 값

측정에서 맨 처음 값인 ABE의 Default 값이 0이기 때문이다.  $\alpha$  값을 1에 가까운 값인 0.9로 설정해줌으로써 빠른 시간 내로 정상적인 ABE값을 측정해낼 수 있다. 본 시뮬레이션에서는 이동평균값 g값과 h값은 각각  $g=0.25$ 와  $h=0.125$ 를 사용하였다. 제안한 알고리즘에서 사용된  $C_{min}$ 은 0.98이고  $C_{max}$ 는 1.02를 사용했다.



(Figure 2) Simulation network topology

<표 1> 시뮬레이션 파라메타

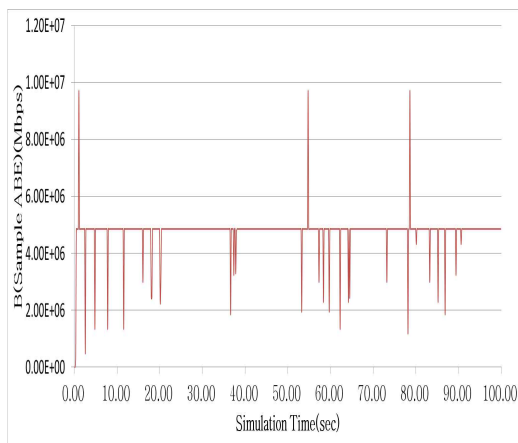
Parameter	Value
Link Capacity of Sending Node	100Mbps
Delay time of Sending Node	1ms
Link capacity of the Receiving Node	100Mbps
delay time of the Receiving Node	1ms
Link capacity between Wireless Nodes	5Mbps
Delay time between Wireless Nodes	35ms
Buffer Size	45
Simulation Time	100(sec)
Application Protocol	FTP
Packet Size	1400byte
$\alpha$	0.9
g	0.25
h	0.125
$C_{min}$	0.98
$C_{max}$	1.02

<Table 1> Simulation parameters

### 5. 시뮬레이션 결과

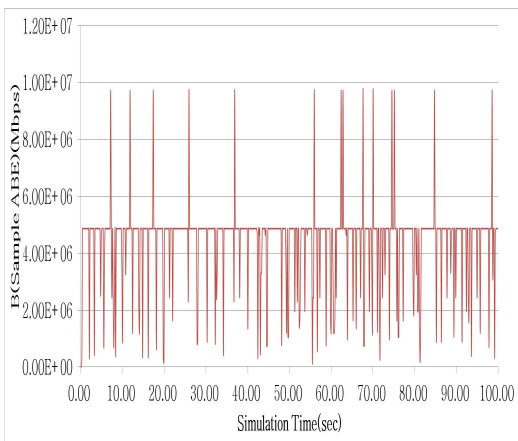
ABE는 매번 ACK가 도착 할 때마다 측정되어지는 현재 대역폭 B값의 평균값이다. 부정확한 ABE값이 측정되었다는 것은 이 B값이 부정확하게 계산되는 빈도수가 많다는 것을 의미한다. 그리고 오류율이 높을수록 이러한 부정확한 B값의 빈도수가 더욱더 많아진다. 먼저, 이러한 현상을 확인하기 위해 시뮬레이션을 통해 오류율이 높아지면 얼마나 많은 부정확한 B값이 측정되는지 알아보았다.

(그림 3) 오류율이 0.001일 때 B의 성능 그래프



(Figure 3) The graph for B performances when the error rate is 0.001

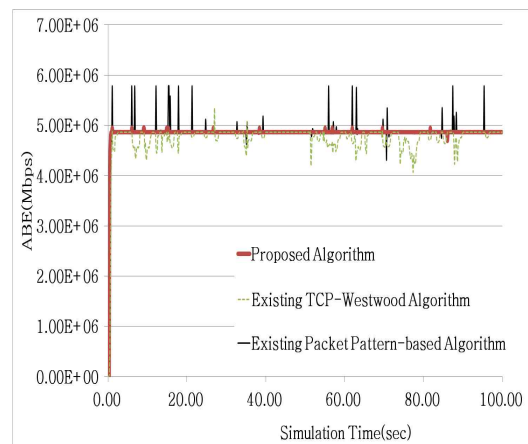
(그림 4) 오류율이 0.01일 때 B의 성능 그래프



(Figure 4) The graph for B performances when the error rate is 0.01

(그림 3)과 (그림 4)는 오류율에 따른 B값의 그래프를 나타낸다. 그림을 보면 알 수 있듯이, 오류율이 커지자 제대로 된 값을 측정하지 못하고, 0 근처로 측정되는 B값의 빈도수가 확연히 늘어남을 알 수 있다. 이로 인해 B값들의 평균인 ABE값이 실제 가용대역폭에 미치지 못한 값으로 측정되는 것이다. (그림 3)과 (그림 4)에서 그래프가 불규칙적으로 5Mbps보다 크게 발생하는 경우는 채널상황의 영향으로 B값 계산에 사용되는 ACK간의 시간차가 매우 작은 값으로 측정되어 실제 가용대역폭보다 심하게 높게 측정되는 경우로 이러한 경우가 자주 발생하면 잘못된 ABE값을 측정하게 된다. 제안한 알고리즘의 성능을 동일한 시뮬레이션환경에서 평가하였다. 성능평가의 지표는 ABE값의 정확성으로 각각의 알고리즘에 대해 패킷 손실률에 따른 ABE값을 측정하였다. 오류율 0.0001과 같이 낮은 오류율일 경우는 기존의 TCP-Westwood 알고리즘과 기존의 패킷 패턴 기반 알고리즘 그리고 제안한 알고리즘 모두에서 비교적 정확하게 5Mbps에 근접한 대역폭을 측정되었다.

(그림 5) 오류율이 0.001일 때 ABE의 성능 그래프

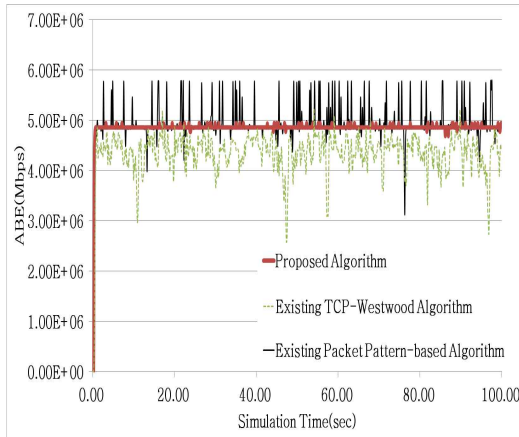


(Figure 5) The graph for ABE performances of three algorithms when the error rate is 0.001

(그림 5)는 오류율이 0.001인 경우에 기존 TCP-Westwood 알고리즘과 기존 TCP 패킷 패턴기반 알고리즘 그리고 제안한 알고리즘에 대한 ABE값의 나타내는 그래프이다. 기존

TCP-Westwood 알고리즘과 기존 TCP 패킷 패턴기반 알고리즘의 경우는 시뮬레이션에서 무선 대역폭으로 설정한 5Mbps를 벗어나는 ABE값이 측정되고 있지만 제안한 알고리즘의 경우에는 매우 정확하게 5Mbps에 근접한 ABE를 측정하는 것을 확인 할 수 있다.

(그림 6) 오류율이 0.01일 때 ABE의 성능 그래프



(Figure 6) The graph for ABE performances of three algorithms when the error rate is 0.01

(그림 6)은 오류율이 0.01일 때 3가지 알고리즘에서 측정된 ABE값의 그래프이다. 기존 TCP-Westwood 알고리즘의 경우에 심각하게 측정값이 가장 잘못측정 되는 것을 확인할 수 있다. 이 경우 무선망이 제공하는 대역폭을 제대로 활용하고 있다고 볼 수 없다. 또한 기존 TCP 패킷 패턴기반 알고리즘의 경우는 기존 TCP-Westwood 알고리즘보다는 더 안정적인 ABE값을 측정하기는 하지만 측정값이 5Mbps보다 크거나 작아지는 현상이 발생하는 것을 보여 준다. 첫 번째로 ABE값이 5Mbps보다 작아지는 경우(예를들어, 시뮬레이션 시간이 80초 근처에선 3Mbps까지 떨어지는 등 심하게 잘못 측정됨)가 발생한다. 그 이유는 오류율이 증가함에 따라 혼잡원도우가 충분히 커지지 못하고 정상적인 패킷들을 촘촘히 보내지 못해 균집한 패킷 전송주기가 자주 반복되어(그림 1 을 참조)  $T_{thresh}$  계산에 포함되지 말아야 할 ACK들간의 도착간격(균집 패킷 전송들 간의 간격인 RTT)

이 자주 포함된다는 것을 의미한다. 이때  $T_{thresh}$  값이 상대적으로 커져서 잘못된 B값이 ABE값에 자주 포함되어 ABE의 정확도가 떨어지게 된다. 이 경우 무선망에서 제공되는 대역폭을 충분히 활용하고 있다고 볼 수 없다.

두 번째로 ABE가 5Mbps보다 크게 측정되는 경우는 실제 가용대역폭보다 더 커져서 혼잡원도우 사이즈가 증가하여 과도한 데이터를 전송하게 되어 타임아웃이나 패킷전송실패의 현상이 많이 발생하게 한다. 그에 비해 제안한 알고리즘의 경우 매우 정확하게 5Mbps에 가까운 값을 측정하는 것을 확인할 수 있다. 이러한 결과들로 다른 알고리즘들에 비해 제안한 알고리즘이 가장 안정적이고 정확하게 무선네트워크에서의 ABE값을 측정하는 것을 알 수 있다.

또한 (그림 5)와 (그림 6)로부터 기존의 2가지 방법들에 대해 오류율이 0.001인 (그림 5)의 경우가 오류율이 0.01인 (그림 6)의 경우보다 이 5Mbps를 벗어나는 ABE값을 비교적 작은 빈도수로 측정하는 것을 볼 수 있다. 이는 패킷 손실률이 비교적 낮아 혼잡원도우가 충분히 커져서 균집한 패킷 전송주기가 자주 반복되지 않기 때문에 나타나는 현상이다.

## 6. 결론

본 논문에서는 무선 TCP 종류중 하나인 TCP-Westwood가 무선 링크상의 오류율 증가에 따라 성능이 저하되는 요인을 분석했다. 따라서 본 논문에서는 이러한 TCP-Westwood를 보다 더 효율적이고 정확하게 성능을 향상시키기 위해 기존의 패킷 패턴 기반 알고리즘에서의  $T_{thresh}$ 를 사용하고 허용범위를 벗어난 ABE를 허용 가능한 값으로 제한당하는 새로운 방법을 추가로 사용하여 ABE를 최대한으로 정확하게 활용 할 수 있는 방법을 제시한다. 그 결과 시뮬레이션 상에서 무선네트워크상의 ABE값을 보다 정확하게 측정할 수 있었다. TCP-Westwood 에서 정확한 ABE를 측정한다는 것은 유선에서 뿐만 아니라 무선에서도 네트워크의 대역폭을 충분히 활용하여 보다 안정적이고 효율적으로 TCP를 활용 할 수 있음을 의미한다.

References

[1] H. Balakrishnam, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP performance over wireless networks," In First ACM International Conference on Mobile Computing and Networking (MOBICOM), p p. 469-481. November, 1995

[2] A. Bakre and B. R. Badrinath, "Indirect TCP for Mobile Host," 15th International Conference on Distributed Computing Systems, 1995.

[3] Chang-Hee Kim, "A Study on Improving TCP Performance in Wireless Network," Journal of Digital Contents Society, Vol. 10, No. 2, pp. 279-289, June 2009.

[4] Mi-Young Park, Sang-Hwa Chung and Yun-Sung Lee, "A Sender-based Packet Loss Differentiation Algorithm based on Estimating the Queue Usage between a TCP sender/receiver", The Journal of Korea society of Computer Information, Vol. 16, No. 1, pp. 133-142, 2011.01.

[5] Nam-jin Cho and Sung-Chang Lee, "Performance Improvement of WTCP by Differentiated Handling of Congestion and Random Loss", The Journal of the Institute of Electronics Engineers of Korea, TCP Session, Vol. 34. No. 9, pp. 30-38, 2008.9.

[6] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," In Proceedings of ACM Mobicom 2001, pp. 287-297, Rome, Italy, July 16-21 2001.

[7] Min-Woo Park and Hyun-Seung Choo, "Analysis and Modified Algorithm Proposal of TCP Westwood in Next Generation Network", Conference proceeding of Korea Society for Internet Information, Vol. 8, No. 2, pp.27-30,2007.11.

[8] URL:http://c3lab.poliba.it/index.php/C3Lab TCP-Westwood

[9] Moonsoo Kang and Seokjoo Shin, "Enhanced Bandwidth Estimation for TCP Westwood", Journal of Kor

ea Society for Internet Information, Vol. 9, No 6, pp. 179-188, 2008.12.

[10] S. Keshav, "A Control-Theoretic Approach to Flow Control," Proc.ACM SIGCOMM, pp. 3-15, Sept. 1991.

[11] Behrouz A. Forouzan, "TCP/IP Protocol Suite," Second Edition, McGraw-Hill, pp.297-335, 2003.

[12] UCB/LBNL/VINT Network Simulator, URL:http://www.isi.edu/ns-nam/ns.

[13] The application of NS-2 Network Simulation, Young-Min Jang, Hongrung Publishing Company.



유 창 열

2008년 3월 - 현재 :강원대학교  
IT대학 전기전자공학부 전  
자통신학과 학사과정

관심분야 : 차세대 이동통신 및 무선 네트워크



김 동 회

2005년 5월 : 고려대학교 전과공학과 (공학박사)

1989년 1월 ~ 1997년 1월 : 삼성전자 전임연구원  
2000년 8월 ~ 2005년 8월 : 한국전자통신연구원 선임  
연구원  
2006년 3월 ~ 현재 : 강원대학교 IT대학 전기전자공  
학부 전자통신학과 교수  
관심분야 : 차세대 이동통신 및 무선 네트워크