

Yellow-Light TCP: 모바일 데이터 전송을 위한 에너지 절감형 프로토콜

최 원 준*, Ramneek*, 석 우 진^o

Yellow-Light TCP: Energy-Saving Protocol for Mobile Data Transmission

Won-jun Choi*, Ramneek*, Woo-jin Seok^o

요 약

정보화 시대에 도래함에 있어서 많은 양의 데이터가 인터넷 상에 쌓이게 되었다. 이러한 정보를 공유하기 위해 유무선 네트워크를 통해서 데이터를 전달하고자 하는 요구가 증대되었다. 특히, 배터리 기반으로 동작하는 모바일 장치를 사용하는 무선 네트워크에서는 많은 양의 데이터 업로드, 다운로드에 인해서 에너지 소모가 많이 발생하였다. 이와 같은 문제를 해결하기 위해서 본 논문에서는 대부분 네트워크 프로토콜 동작에서 사용되는 전송계층인 TCP 혼잡제어 방법을 수정하여 에너지를 절감해 보고자 한다.

Key Words : TCP, Wireless Network, Network Protocol, Energy, Mobile, Transmission

ABSTRACT

Abundant data has been generated over the internet as the arrival of information age. To share the information, Wired, wireless network are required to transmit the data. Especially, In the wireless network which is using mobile device based on battery, energy consumption is growing due to uploading, downloading the abundant data on mobile device. In order to solve the problem, This paper addresses the protocol of the modified TCP congestion control that is being used for the most network protocol to save energy.

I. 서 론

초고속 인터넷의 급속한 발전과 더불어 무선 네트워크 상에서 모바일 장치를 사용한 데이터의 전송 요구가 확대됨에 따라 한정된 에너지로 유지되는 모바일 장치에서 데이터 전송으로 인한 에너지 소비가 중요한 관심사가 되었다^[1]. 그중에서도 모바일 장치에서 에너지 소비를 많이 하는 부분은 CPU, RAM, 그

래픽, LCD, 백라이트 등이 있지만 네트워크에서의 다운로드와 업로드가 에너지 소비의 많은 부분을 차지하고 있다^[2,14]. 이러한 네트워크에서 빼놓을 수 없는 부분이 신뢰적인 전송 프로토콜인 TCP로써 유선에서 뿐만 아니라 무선에서도 많이 사용되고 있어 TCP를 이용한 연구가 진행되고 있다^[10,15,16]. 그 대표적인 예로 무선 네트워크 상에서 SACK, Newreno, Reno 의 에너지 효율을 휴면 상태 구간에서 소비되는 에너지

* 본 연구는 KISTI 연구과제(첨단연구망 기반 협업플랫폼 서비스:K-15-L01-C04-S03) 지원으로 수행되었습니다.

• First Author : Korea University of Science and Technology, cwj@ust.ac.kr, 학생회원

◦ Corresponding Author : Koera Institute of Science and Technology, wjseok@kisti.re.kr, 정회원

* Korea University of Science and Technology, ramneek@ust.ac.kr

논문번호 : KICS2014-10-436, Received October 29, 2014; Revised January 12, 2015; Accepted February 13, 2015

와 전체 에너지 면에서 비교한 연구¹¹⁾와 헤더 크기를 압축하여 전송하고 선택적 ACK 를 추가하여 버스트 에러에 대처하고 수정된 타이머를 사용하여 불필요한 재전송을 하지 않도록 설계된 E2TCP 프로토콜 연구¹²⁾가 있다. 또한, 송신자에서 계산되는 체크섬 비용 계산과 데이터 복사에 필요한 수행의 축소와 TO(TimeOut), TD(Triple Duplicate) 수행의 축소 그리고 MTU(Maximum Transmission Unit) 크기를 증가시켜 계산 비용을 감소시킴으로 에너지 소모를 감축시키고자 한 연구¹³⁾ 등이 있다. 하지만 이러한 논문들은 TCP에서 가지고 있는 혼잡제어 방법을 사용하여 패이징이나 간섭 등의 다양한 원인으로 발생하는 무선 망에서 패킷 손실이 발생하면 CWND (Congestion Window)를 감소시켜 전송률이 저하되고 전송 지연의 원인으로 핸드폰 배터리의 사용량이 증가하여 에너지의 효율적 사용에 제약을 주게 된다^{4,6)}.

따라서 본 논문에서는 여유 대역폭을 확보하여 오류 확률을 감소시키면서 재전송을 방지하고 혼잡 상태나 간섭으로 인한 패킷 손실이 발생할 때 전송을 보류하여 혼잡 구간을 피함으로써 효율을 높이는 Yellow-Light TCP를 제안한다. 전송 중 손실이 발생할 경우에는 여유 대역폭을 100% 사용하지 않고 여유 대역폭을 남겨둠으로써 혼잡이나 간섭 등 기타 원인으로 인한 패킷 손실률을 줄이면서 안정적인 CWND를 전송하도록 한다. 하지만 네트워크 상황이 일정시간이 지나도록 좋아지지 않으면 CWND의 양을 전송하지 않음으로써 전송을 보류하여 수면 모드¹³⁾로 전환하게 된다. 전송 모듈을 정지시키는 방법도 있지만 갑작스런 전송 모듈의 정지는 도중에 큐에 저장된 패킷들의 손실을 유발하게 되고 재전송을 증가시켜 에너지 소모를 더 불러올 수 있다.

Yellow-Light TCP는 실시간으로 대역폭을 예측하게 되는데 예측의 기준은 무선 망에서 성능이 좋은 프로토콜로 알려진 TCP Westwood^{13,7,9)}의 대역폭의 측정 방법을 수정한 방법이다. 다른 TCP도 많이 있지만 TCP Westwood와 구별되는 점은 중간 노드에서 TCP 패킷 분석을 요구하지 않고, 지속적인 ACK 모니터를 통해서 송신자측에서 연결 상태를 지속적으로 확인한다는 데에 있다. 이러한 특징은 수시로 네트워크 상황이 변하는 무선 네트워크 환경에서 적합한 프로토콜로 알려져 있다¹³⁾. TCP Westwood 의 대역폭 예측 방법은 송신자에서 수신자로 보내진 패킷에 대한 ACK의 도착 시간의 차이를 통해서 대역폭을 예측하게 되는데 네트워크 상황이 수시로 변하는 무선 망에서 지연된 ACK가 발생할 수도 있고¹⁵⁾ 패킷 손실이

발생했을 경우에 원인에 대한 판단이 불분명하여 손실이 발생할 때마다 전송률을 조절한다는 단점을 가지고 있다^{4,9)}. 제안된 방법은 대역폭 예측을 위하여 ACK 도착 시간을 통계적인 방법으로 신뢰성 있는 값을 사용하게 된다. 네트워크 대역폭 상황에 따라서 지연된 ACK가 발생할 수 있기 때문에 ACK가 도착할 때마다 대역폭을 변경하여 전송 혼잡 윈도우 값을 변경하는 방식은 패킷 손실이 발생하였을 때 원인 판단이 불분명하여 전송률을 조절한다는 단점을 조금이라도 해결해 보고자 도착한 ACK의 값의 신뢰성을 높이기 위해서 바로 예측 대역폭을 변경하는 것이 아니라 연속된 3개의 ACK 도착 시간 비교를 통해서 결정하게 된다. 예를 들어 A, B, C 라는 ACK 도착 시간이 있을 때, A와 B의 도착시간의 차이와 B와 C의 도착시간의 차이가 동시에 일정 값 sValue 이상일 때에만 혼잡 윈도우를 조정하여 전송률을 조절하는 것이다. sValue는 여러번의 실험을 거쳐 최적 값을 결정하였다. 이렇게 하는 이유는 ACK가 오면 바로 전송률을 조절하는 것이 아니라 현재 도착한 ACK의 도착 시간이 현재의 대역폭 상황에 맞는 값인지를 확인하기 위한 방법이라고 생각하였기 때문이다. 이렇게 함으로써, 전송 속도는 조금 더디지만 혼잡을 피하며 패킷 손실률을 줄여서 재전송을 감소하는 방법으로 에너지 소모량을 조절하는 방법을 사용한다. 연속적으로 3 Dup Acks(Triple Duplicate Acknowledgements)가 빈번하게 일어나는 경우와 RTO(Retransmission TimeOut)가 발생할 경우에는 CWND의 양을 전송하지 않음으로써 네트워크 상태는 살아 있지만 패킷을 전송하고 수신하는데 필요한 에너지 소모를 줄인다. 이렇게 함으로써 유휴 구간에 소비되는 에너지를 보내는 패킷의 양으로 조절하게 된다. 그리고 3 Dup Ack와 RTO가 빈번하게 발생하는 구간에 대해서는 네트워크 상황이 좋지 않음을 판단하여 여유 대역폭을 남겨두고 CWND를 전송하게 된다. 이와 같이 전송 보류 방식과 여유폭 확보의 방법을 사용하는 방식으로 돌아가는 Yellow-Light TCP는 에너지 효율이 많이 떨어지는 무선 네트워크 환경에서 효과적인 프로토콜로써 기여할 것으로 보인다.

본 논문에서는 다음과 같은 구조로 작성되었다. 2장에서는 Yellow-Light TCP의 원리와 구조, 그리고 동작 방법에 대해서 설명되었다. 3장에서는 제안하는 Yellow-Light TCP를 검증하기 위해서 시뮬레이션 모델을 정의하고 실험에서 도출된 결과를 토대로 토의해 본다. 마지막으로 4장에서는 본 논문의 결론을 설명한다.

II. Yellow-Light TCP

2.1 Yellow-Light TCP 정의

본 장에서는 무선 네트워크 상에서 많은 데이터 전송이 필요한 모바일 장치에서 에너지의 효율적 사용이 이슈화 되었고 이러한 문제를 해결해 보고자 유선과 무선에서 널리 사용되고 있는 TCP 전송 프로토콜을 수정하여 제안된 Yellow-Light TCP의 기본적인 원리에 대해서 설명하고자 한다.

무선 네트워크 상태에서는 대역폭의 상황이 수시로 변하기 때문에 그때마다 CWND를 적절하게 조절해 주되 너무 자주 전송량을 바꾸다 보면 처리량의 감소와 손실률의 증가를 초래할 수 있다. 그래서 Yellow-Light TCP는 전송 초기에 빈 윈도우로 대역폭을 미리 예측하여 CWND를 결정하게 된다. 빈 윈도우의 개념은 다음과 같다. TCP는 송신자에서 패킷을 보내면 수신자는 그에 대한 ACK를 전송하여 패킷 전송 확인 절차를 거치게 된다. 수신자 쪽에서 ACK를 만들어 패킷 header에 첨부하여 보내는 패킷도 하나의 데이터이지만 50bytes 이하의 용량을 가지고 있기 때문에 ACK로 인한 혼잡 상황은 드물게 일어난다는 사실이 hybrid-SlowStart 논문에서 언급한 바 있다. 따라서, 빈 패킷을 생성해서 패킷 header에 ACK만 첨부하여 보내는 수신자의 활동처럼 송신자에서 실제 데이터를 보내기 전에 SSTH같은 초기 셋팅 값을 결정하기 위해 전송 초기에 빈 윈도우를 전송한다는 메카니즘을 선택하게 되었다. TCP-Westwood 같은 경우에도 ACK의 도착 시간으로 대역폭을 예측하고 있기 때문에 이와 같은 메카니즘이 대역폭 상황을 정확히는 아니지만 전송 초기에만 사용할 수 있는 간단한 방법이다. 대역폭의 값을 계산할 경우 연속적으로 수신된 패킷의 시간의 차로 구하게 되는데 대역폭 예측 구간의 신뢰성을 높이기 위해 대역폭을 예측할 때마다 CWND를 바꾸지 않고 일정 구간에서의 대역폭의 변화량을 관찰하여 CWND의 값을 변경한다. 이렇게 하기 위해 연속된 ACK에 대한 도착 시간의 차이를 비교하여 변화가 있을 경우에는 감소를 수행하고 대역폭 상태가 다시 높아졌을 때에는 슬로우 스타트를 다시 수행하여 여유 대역폭까지 증가시킨다. 유휴 구간에서의 에너지 소모를 감소시키기 위해 3 Dup Acks와 RTO 상황 시에는 CWND의 양을 전송하지 않고 일정 구간 네트워크 상황을 관찰하여 대역폭 상황이 좋아지기 시작하면 다시 CWND를 늘리는 방식을 사용한다. TCP-westwood에서는 임계값을 결정할 때 현재 대역폭과 최소 RTT를 기준으로 정하게 된

다. 현재의 대역폭은 $(\text{Ack 개수}) \times (\text{세그먼트 크기}) / (\text{현재 ACK 도착시간} - \text{바로 전에 도착한 ACK 도착시간})$ 의 공식으로 구하게 되며, 임계값은 이렇게 구한 현재의 대역폭을 최소 RTT로 곱해서 결정하게 된다. 네트워크의 상황이 안좋아지면 DUP ACK의 개수가 증가하게 되므로 이러한 경우 손실된 패킷에 대한 재전송을 할 경우에 기존에는 혼잡 윈도우의 값을 임계치로 설정하여 재전송하였다면 제안된 방법은 혼잡 윈도우의 값을 임계치의 70% 정도의 값으로 설정하여 전송하게 된다. 이러한 방법이 여유 대역폭을 확보하여 전송하는 간접적인 방법이라고 생각한다.

그림 1은 제안된 방법의 대역폭 사용 구상도를 나타내고 있다. 전송 초기에는 Pre-estimation 단계로써 빈 윈도우를 보내서 대역폭을 예측하게 되고 대역폭 예측에 의해서 임계치가 결정이 되면 슬로우 스타트 단계부터 패킷을 전송한다. 일정 구간 네트워크 상황을 관찰하기 위해 다음과 같은 방법을 사용한다. 슬로우 스타트 구간과 혼잡 회피 구간에서의 대역폭 예측에 따른 윈도우의 양의 결정은 연속된 3개의 ACK의 시간 차이로 결정하게 되고 정해진 sValue 값을 넘게 되면 대역폭의 상황이 변경되었음을 인식해서 전송되는 윈도우의 양을 변경하게 된다. Stable Transmission 구간은 윈도우의 양을 일정하게 조정하면서 전송하다가 다시 네트워크 상황이 안좋아진 구간에서는 Robust Reduction 구간으로 윈도우의 양을 -1만큼 감소시킨다. 이러한 구간이 지속될 때에는 전송 상태를 수면모드로 변경하여 전송 윈도우의 양을 0으로 설정하여 전송하게 되고 다시 네트워크 상황이 호전되었을 경우에는 네트워크 전송 대역폭을 100% 사용하지 않고 alpha% 만을 사용하여 전송한다. 남겨진 여유 대역폭은 $(100 - \alpha)\%$ 이다. 이렇게 하는 이유는 정해진 bottleneck 대역폭에서 서로 다른 플로우 간에 간섭이 발생할 수 있고, 각 플로우가 대역폭을 100% 사용하게 되면 이로 인한 손실이 발생할 수 있다. 이와 같은 상황을 미리 예방하기 위해서 정해진 alpha% 만큼의 여유 대역폭 만을 사용한다.

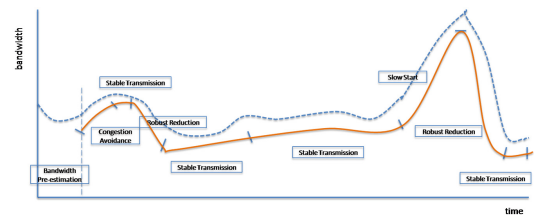


그림 1. Yellow-Light TCP의 대역폭 사용 구상도
Fig. 1. Yellow-Light TCP Layout of bandwidth usage

2.2 Yellow-Light TCP 구조

본 장에서는 제안된 Yellow-Light TCP와 무선 네트워크 상에서 성능이 좋은 프로토콜로 알려진 TCP-Westwood와의 차이점을 기술하면서 제안된 Yellow-Light TCP의 내부 구조를 기술한다.

제안된 방법은 연속된 ACK의 시간차이가 비슷할 경우에만 CWND를 올리는 방식인 수정된 대역폭 예측 방법을 사용하게 된다. 타임아웃이 발생하였을 경우에는 CWND를 전송하지 않음으로써 유휴 구간에서의 에너지 소비를 줄인다. 중복 ACK를 수신하게 되면 휴면 상태 모드로 전환하고 전송 중인 CWND의 양을 대역폭의 Alpha%만을 사용하여 전송하게 되며, 계속 네트워크 상황이 좋지 않으면 다시 CWND의 양을 전송하지 않음으로써 에너지를 절약한다. 주어진 ACK 도착 시간의 차이를 통해서 대역폭을 예측하는 방법은 신뢰성이 다소 떨어질 수 있다. 왜냐하면 수시로 변하는 무선 네트워크 환경에서 지연된 ACK 수신과 간섭으로 인한 패킷 손실률이 많이 발생하기 때문이다. 따라서 신뢰성을 높이기 위해서 일정 시간동안 수신되는 ACK 시간 차이의 변화량을 확인해서 CWND의 증가량을 결정하게 된다. 또한, 네트워크 상황이 혼잡이 발생하는 상황이라면 대역폭의 알파%만 사용하여 여유를 남겨둬으로써 패킷 손실률을 줄이고 다른 플로우와의 간섭을 줄여서 모바일 장치에서의 에너지 소모를 방지한다.

그림 2는 Yellow-Light TCP의 구조도를 설명한 것이다. 무선 네트워크에 좋은 성능을 보이고 있는 TCP Westwood의 기본 구조에서 몇 가지를 추가하거나 보완하였다. 첫 번째는 대역폭 예측 방법론으로써 기존의 방법이 현재 도착된 ACK의 시간과 이전에 도착한 ACK의 도착시간과의 차이를 통해서 대역폭을 결정했다면 제안된 방법은 첫 번째 도착한 ACK의 시간을 A, 두 번째 도착한 ACK의 시간을 B, 세 번째 도착한 ACK의 시간을 C라고 놓았을 때 A-B와 B-C의 절대값의 차이가 sValue값 이상이 되었을 경우에 무선 네

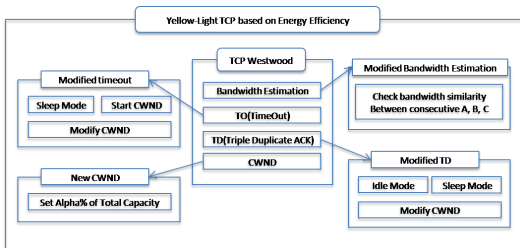


그림 2. Yellow-Light TCP의 구조
Fig. 2. The structure of Yellow-Light TCP

트워크 대역폭의 상황이 변경되었음을 확인하는 방법이다. 두 번째 방법은 수정된 타임아웃 방법이다. 타임아웃이 발생하게 되면 수면모드로 변경되게 되는데 이때 CWND의 양을 보내지 않음으로써 패킷 손실률과 재전송을 방지한다. TCP 전송 모듈을 정지하는 것도 하나의 방법이 될 수 있지만, 전송 도중에 전송 모듈을 정지하게 되면 중간 노드의 버퍼에 존재하는 패킷들의 손실이 발생하여 더 많은 손실률을 초래하게 되고 근본적으로 재전송을 증가시키게 된다. 따라서 TCP 전송 세션을 유지하면서 패킷을 전송하지 않음으로써 손실률을 억제하여 에너지 효율을 높이게 된다. 타임아웃을 마치고 일정 시간 후 재전송을 시도할 경우에는 alpha% 이론을 적용한다. 세 번째로 중복된 3개의 ACK가 발생하였을 경우에는 휴면모드로 전환되게 되는데 이때의 CWND는 주어진 bottleneck 대역폭의 alpha% 만큼의 CWND를 전송하게 되고 중복된 ACK가 연속적으로 발생하는 구간에서는 수면모드로 변경하여서 CWND의 양을 보내지 않음으로써 손실률을 방지하여 모바일 장치에서의 에너지 효율을 높이게 된다. 새로운 CWND를 전송할 때에는 (100-alpha)%만큼의 여유 대역폭을 남겨두고 전송함으로써 에너지 효율을 높인다.

2.3 Yellow-Light TCP 에너지 모델

본 장에서는 무선 네트워크 환경에서 모바일 장치를 사용하여 데이터를 전송할 때 수신자인 모바일 장치에서 데이터를 수신할 때 소비된 에너지를 측정하기 위해서 사용된 에너지 모델을 제시한다. 모바일 장치에서 소모되는 에너지의 양을 계산하기 위해 다음과 같은 공식을 사용하였다.

$$E=V*I*T \tag{1}$$

- E: 모바일 장치에서 사용된 전기 에너지
- V: 모바일 장치에서 사용된 전압
- I: IDLE, SLEEP, TX, RX에 따른 전류
- T: 하나의 패킷이 전송되는데 걸리는 지연 시간

$$Re = Ie - Ce \tag{2}$$

- Re: 노드의 남은 에너지
- Ie: 초기 에너지
- Ce: 네트워크 상태(Idle, Sleep, Tx, Rx)에 따른 소비 에너지(Ce)

네트워크 상태에 따른 에너지를 알아보기 위해 초기 에너지를 설정하고 전류 값을 다르게 주어 실험 노트에서 데이터 전송 후 남은 에너지를 계산한다.

2.4 Yellow-Light TCP 동작 방법

본 장에서는 제안된 Yellow-Light TCP의 시뮬레이션 알고리즘과 에너지 효율에 필요한 중요한 모듈을 설명하고 내부적으로 돌아가는 동작 방식을 기술한다.

무선 네트워크 상에서 에너지 효율을 높이기 위해 대역폭 예측 방법을 신뢰성을 높이는 방법을 사용한다. 대역폭 예측 알고리즘의 기존 방법은 도착한 ACK의 시간의 차이를 통해서 현재의 대역폭을 예측하고 임계치와 윈도우 값을 결정하게 된다. 이와 같은 메커니즘을 신뢰성을 높이기 위해 대역폭을 예측하고 난 후에 바로 윈도우 값을 올리지 않고 연속된 ACK 도착시간의 변화 경향을 확인하여 올리게 된다. 그리고 연속적으로 Dup ACK가 3번 일어날 경우 또는 RTO가 발생할 경우에는 CWND를 보내지 않고 빈 윈도우를 보내서 대역폭의 상황을 확인한다. 모바일 장치를 혼잡 또는 버스트한 패킷 손실이 발생할 때에 정지하게 되면 중간 노드의 버퍼에 쌓여진 패킷이 손실될 우려가 있기 때문이다. 또한, Dup ACK가 발생한 후에 CWND를 재전송 할 경우 주어진 대역폭의 100%를 사용하지 않고 Alpha%만 사용한다.

그림 3은 Yellow-Light TCP의 내부 알고리즘을 도식화 한 것이다. 송신자측에서 패킷을 생성하고 수신자 쪽으로 패킷을 보낸 후에 보낸 패킷에 대한 ACK를 기다릴 때에 다음과 같은 알고리즘을 사용한다. 우선 ACK가 왔는지 또는 타임아웃이 되었는지를 확인한다. 타임아웃이 되었으면, 현재의 CWND와 임계치를 저장하고 임시 휴면 모드로 전환되게 된다. 이때의

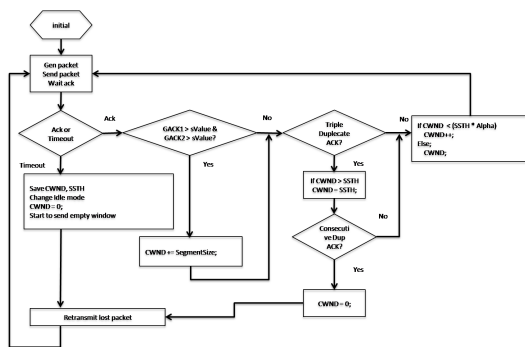


그림 3. Yellow-Light TCP 시뮬레이션 알고리즘
Fig. 3. Yellow-Light TCP Simulation algorithm

전송 윈도우는 0으로 설정한다. 그리고 현재의 대역폭 상황을 수시로 확인하기 위해 빈 윈도우 전송을 시작한다. ACK가 수신 되었을 때에는 현재 대역폭의 상황을 판단하기 위해 연속된 ACK의 도착 시간 변화율을 계산하여 $|A-B|=GACK1$, $|B-C|=GACK2$ 로 설정한다. GACK1과 GACK2가 sValue 값을 초과하였을 경우에는 대역폭의 상황이 변경되었음을 인지하고 전송되는 CWND의 양을 변경한다. 시뮬레이션에서 사용된 sValue값은 여러 번의 실험을 거쳐 도착된 시간의 변화율을 확인하여 결정하였다. 네트워크 상황이 호전되었을 경우에는 패킷의 세그먼트 양만큼을 더해 주게 되고 네트워크 상황이 안 좋아질 경우에는 주어진 양만큼 빼서 전송한다. 중복된 3개의 ACK가 발생하였을 경우에는 전송중인 CWND가 임계치를 넘어설 경우에 CWND를 임계치에 맞춰주고 지속적으로 중복 ACK가 발생할 경우에는 CWND를 0으로 설정한다. 연속적으로 중복 ACK가 발생하지 않을 경우에는 주어진 대역폭의 alpha%만큼으로 CWND를 설정하게 된다. 중간에 손실된 패킷에 대해서는 재전송 메커니즘을 적용하여 전송함으로써 손실된 패킷을 관리하고 혼잡으로 인한 간섭을 피함으로써 모바일 노트에서의 패킷 수신으로 인한 오버헤드를 줄여 에너지 효율을 높이게 된다.

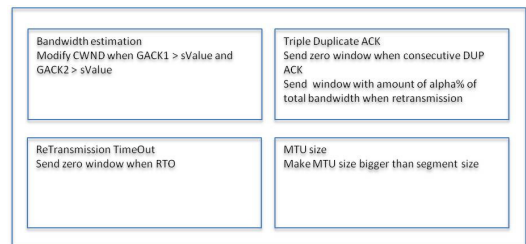


그림 4. Yellow-Light TCP의 설계도
Fig. 4. The design of Yellow-Light TCP

2.4.1 대역폭 예측

대역폭을 예측하기 위해 현재 도착한 ACK의 시간과 이전에 도착한 ACK의 차이를 저장하고 연속된 3개의 차이를 저장하기 위해 GACK 배열을 사용한다. 3개를 사용하는 이유는 수신된 ACK의 차이를 많이 보면 현재의 네트워크 대역폭 상황을 좀더 신뢰성 있게 추정할 수 있지만 복잡한 계산으로 인한 CPU사용률의 증가로 인하여 에너지가 더 많이 소모될 수 있으므로 최소 개수인 3개를 선택한다. 연속적인 ACK의 도착 시간의 차이가 일정 값(sValue)보다 크게 되면 CWND를 올리는 방식을 사용한다. ACK를 수신하게

되면 GACK의 배열에 순차적으로 ACK 차이값을 저장하고 CWND를 올릴 때 사용한다.

그림 5는 Yellow-Light TCP의 대역폭 예측 방법을 나타낸 것이다. 송신자 S에서 수신자 R로 데이터 패킷 D1, D2, ... 를 전송하였을 경우에 그에 해당하는 ACK1, ACK2, ... 에 대해서 GACK1=ACK2-ACK1; GACK2=ACK3-ACK2; 로 설정하고 GACK1과 GACK2의 비교를 통해서 대역폭을 예측한다.

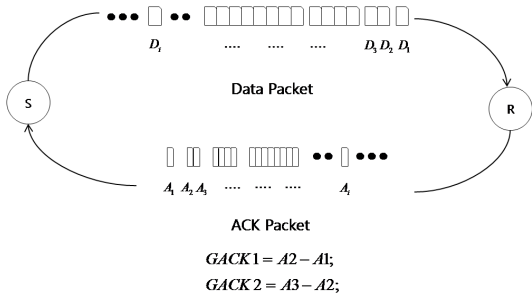


그림 5. Yellow-Light TCP의 대역폭 예측
Fig. 5. Bandwidth estimation of Yellow-Light TCP

2.4.2.3 DUP ACK

기본적으로 임계치를 결정할 때에는 전송하고 수신하는데 걸리는 전체 시간(RTT)을 사용한다. CWND가 SSTH보다 크게 되면, SSTH의 alpha = 70% 정도만 사용하여 여유 대역폭을 확보하게 된다. 70%를 결정하는 이유는 성능 평가에서 증명할 것이다. 연속적으로 3 Dup ACKs가 발생하는 구간에서는 다시 CWND를 전송하지 않음으로써 네트워크 상황을 모니터링하고 현재 네트워크에 여유 대역폭이 생길 때까지 빈 윈도우를 보내서 확인한다.

2.4.3 RTO

재전송 타임아웃이 발생하였을 때에는 임시 휴면모드로 전환하고 대역폭 상황이 여전히 좋아지지 않으면, CWND를 보내지 않고 여유 대역폭이 생길 때까지 빈 윈도우를 보내서 확인한다. 네트워크의 상황이 호전되게 되면 CWND를 주어진 대역폭의 alpha% 만큼 증가시켜 전송하게 된다.

2.4.4 MTU 크기

OS마다 다소 차이는 있지만 MTU 크기가 크면 클수록 네트워크 전송 프로토콜인 TCP에서 발생하는 에너지 효율을 높일 수 있다^[14]. 따라서, 이더넷 최대 크기인 1500 바이트를 시뮬레이션에 사용한다.

NewAck	
1	if (CWND < SSTH)
2	If (GACK1 - GACK2 > sValue And
3	GACK2 - GACK1 > sValue)
4	CWND += SegmentSize;

슬로우 스타트 구간에서 CWND가 주어진 임계치보다 작을 경우에 연속된 3개의 ACK에 대한 도착 시간의 차이가 sValue보다 크게 되면 CWND의 양을 전송되는 패킷의 세그먼트의 크기 만큼 증가시킨다.

Received Ack	
1	GACKTMP = (CurACK - LastACK);
2	GACK(0) = preLastACK - pre-preLastACK;
3	GACK(1) = LastACK - preLastACK;
4	GACK(2) = GACKTMP;
5	GACK1 = GACK(1) - GACK(0);
6	GACK2 = GACK(2) - GACK(1);

송신자에서 수신자로 보내어진 패킷에 대한 ACK가 송신자에서 수신되었을 경우에 도착된 ACK의 시간의 차이를 GACK 배열에 저장하여 GACK1과 GACK2를 계산한다. 이 값을 토대로 CWND의 양을 결정하게 되고 모바일 장치에서의 에너지 효율을 높이게 된다.

DupAck	
1	SSTH = CurBW * RTT;
2	if (CWND > SSTH)
3	CWND = SSTH * alpha;
4	if (ContinuousDupACK = 3)
5	CWND = 0;

중복된 ACK가 발생하였을 경우에는 임계치는 기존 방법과 같이 현재의 대역폭에 RTT를 곱해서 결정하게 되고, 현재 전송되고 있는 CWND가 계산된 임계치보다 크게 되면 CWND를 주어진 임계치의 alpha% 만큼 조절하여 전송한다. 연속적으로 중복 ACK가 발생하게 되면 CWND를 0으로 만들고 빈 윈도우를 전송함으로써 대역폭을 예측하며 손실률을 줄이고 혼잡을 피하면서 에너지 효율을 높이게 된다.

Retransmission TimeOut	
1	CWND = 0;

재전송 타임아웃이 발생하게 되면 임시 휴면 모드로 변경되었다가 타임아웃이 길어지게 되면 CWND를 0으로 만들어 네트워크 TCP 전송 세션을 유지하면서 네트워크 상황을 판단하여 다시 패킷을 전송함으로써 에너지 효율을 높이게 된다.

III. 성능 및 에너지 효율성 검증

3.1 시뮬레이션 모델

시뮬레이션 테스트를 위해서 NS-3^[18]를 사용하여 구성 토폴로지를 OS X10.9.5에서 구현하였으며, 패킷의 크기는 1040byte로 전송하고 MTU는 이더넷의 최대 크기인 1500byte를 적용하였다. 패킷의 개수는 1000개로 설정하였으며, 송신자에서 수신자로 보내는 패킷의 양이 1000개이고 각 패킷간의 간격은 1초로 설정한 크로스 트래픽을 만들어 n1에서 n7으로 UDP(User Datagram Protocol)를 전송하면서 메인 플로우의 값을 측정하였다. 모바일의 초기 에너지는 0.1mA이고, 전송에 필요한 전류 값은 0.0002mA, 수신에 필요한 전류 값은 0.0001mA, 유휴 구간에서 소비되는 전류 값은 0.00001mA로 설정하였다. 링크의 Data rate는 500Kbps이며 delay는 2ms로 설정하였다. 전체 시뮬레이션 시간은 100초로 설정하여 테스트 하였다. N0 - N4는 유선 네트워크이며 N4 - N7은 무선 네트워크이다. 무선 네트워크의 설정은 랜덤 워크 시뮬레이션을 사용하였다. 크로스 트래픽을 N1에서 N7으로 보내면서 메인 플로우가 N0에서 N5로 패킷을 전송할 때의 실험 결과를 비교해 보았다.

그림 6은 실험에 사용된 토폴로지를 나타낸 것이다. N4는 AP로써 무선 네트워크를 위한 것이고 N0에서 N4까지는 유선, N5에서 N7까지는 무선으로 연결되어 있다. 실험에 사용된 플로우는 메인 플로우로써 N0에서 모바일 노드인 N5로 패킷을 전송하게 된다.

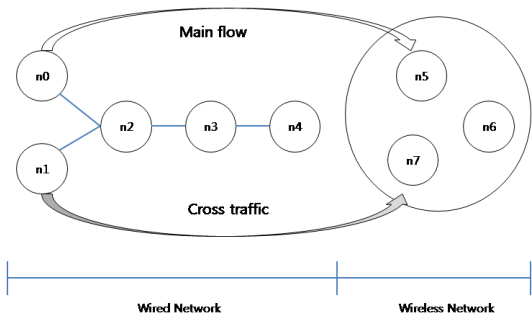


그림 6. 시뮬레이션 토폴로지
Fig. 6. Simulation topology

그와 동시에 크로스 트래픽은 N1에서 N7으로 전송한다. 메인 플로우는 신뢰적인 전송 프로토콜인 TCP를 사용하여 제안된 방법을 적용하게 되고 크로스 트래픽은 영상 정보의 전송에 유용한 UDP 프로토콜을 사용하게 된다.

크로스 트래픽의 패킷의 양은 각 플로우마다 1000개씩 설정하였으며, 패킷 사이의 구간은 1초로 정의하고 패킷 크기는 1024byte로 설정하였다. 대역폭이 500Kbps 인 상태에서 실험 플로우의 대역폭을 100Kbps로 설정하면 패킷을 전송하기 위한 최저 대역폭을 넘어서기 때문에 오류가 많이 발생하고 최대 로 설정하면 중간 노드에서 패킷이 손실될 확률이 있기 때문에 대역폭 사용률을 400Kbps 로 설정하고 크로스 트래픽이 없을 때에는 아래 표 1과 같았다.

표 1. 프로토콜의 기본 설정 결과
Table 1. The results of protocol basic setting

Variant	loss(%)	throughput(Mbps)	duration(s)
TCP-Westwood	0	0.391248	22.0828
Yellow-Light TCP	0	0.386824	22.3239

3.2 결과 토의

3.2.1 alpha%의 결정

위와 같은 기본 설정을 토대로 alpha%를 결정하기 위해 크로스 플로우가 16개, 18개, 20개 일때에 메인 플로우의 대역폭 사용률을 변화시켜 가면서 패킷 손실률과 처리량 그리고 지연시간을 확인하였다. 메인 플로우가 대역폭을 100Kbps를 사용하고 있을 때는 주어진 대역폭 500Kbps에서 크로스 플로우가 있는 상태에서 오류없이 패킷을 보낼 수 있는 대역폭임을 확인할 수 있다. 하지만, 메인 플로우가 현재 네트워크 상의 주어진 대역폭을 100Kbps 이하로 사용하게 되면 패킷을 보낼 수 있는 최저 대역폭을 벗어나기 때문에 버스트한 에러가 발생한다. 메인 플로우가 주어진 대역폭을 100Kbps로 사용하고 있을 경우 손실률은 없지만 전송시간은 증가하는 현상이 생긴다. 그 이유는 TCP는 혼잡제어를 사용하여 Dup Ack가 발생하면 재전송^[17]을 하기 때문에 대역폭 사용률이 낮을 때에는 전송 패킷에 대한 기다리는 시간이 증가하여 이와 같은 현상이 발생한다. 처리량은 여유 대역폭까지는 점차적으로 증가하다가 400Kbps에서는 패킷 손실률이 증가하기 때문에 감소한다. 트래픽의 빈도가 낮을 때에는 처리량이 조금 올라가는 것을 결과를 통해

표 2. 크로스 플로우가 16개 일 때 메인 플로우
Table 2. Main flow when cross flow is 16

main	loss	loss rate(%)	throughput (Mbps)	duration(s)	cross flow
100	0	0	0.095199	84	16
150	1	0.1	0.143044	56	16
200	1	0.1	0.190659	42	16
250	1	0.1	0.237336	34	16
300	1	0.1	0.285789	28	16
350	1	0.1	0.331063	24	16
400	6	0.6	0.334379	24	16

표 3. 크로스 플로우가 18개 일 때 메인 플로우
Table 3. Main flow when cross flow is 18

main	loss	loss rate(%)	throughput (Mbps)	duration(s)	cross flow
100	0	0	0.095204	84	18
150	1	0.1	0.14301	56	18
200	1	0.1	0.190659	42	18
250	1	0.1	0.237336	34	18
300	1	0.1	0.285454	28	18
350	1	0.1	0.31806	25	18
400	52	5.2	0.316911	24	18

표 4. 크로스 플로우가 20개 일 때 메인 플로우
Table 4. Main flow when cross flow is 20

main	loss	loss rate(%)	throughput (Mbps)	duration(s)	cross flow
100	0	0	0.095174	84	20
150	1	0.1	0.142923	56	20
200	1	0.1	0.190643	42	20
250	1	0.1	0.237336	34	20
300	1	0.1	0.285108	28	20
350	1	0.1	0.307456	26	20
400	77	7.7	0.306743	24	20

확인할 수 있다. 시뮬레이션에서 크로스 플로우가 존재하기 때문에 메인 플로우의 대역폭 사용률이 150Kbps ~ 350Kbps 일 때에는 손실률이 아주 작게 발생한다. 하지만, 400Kbps 부터는 다시 손실률이 증가하는 현상이 있다. 이와 같은 현상은 크로스 플로우의 개수를 다르게 주어도 비슷하게 나타난다. 따라서, 크로스 플로우의 영향을 덜 받고 여유 대역폭을 남겨

두면서 보낼 수 있는 메인 플로우의 최대 대역폭은 주어진 대역폭의 70%를 사용하게 된다.

3.2.2 MTU 크기의 결정

MTU의 크기를 결정하기 위해서 전송 패킷 크기를 600bytes로 주고 손실률이 어느 정도 발생할 수 있는 상황을 만들어 테스트해 보기로 했다. 크로스 플로우의 개수를 20개 정도로 설정하고 MTU의 크기에 따른 손실률을 측정한 결과는 다음과 같다. MTU의 크기가 전송되는 패킷의 크기와 가까울 경우에는 손실률에 변동이 있지만 전송되는 패킷 크기에 대하여 어느 정도 이상일 경우에는 손실률이 일정한 것을 확인할 수 있다. 이를 통해서 손실률이 일정한 MTU 크기를 정할 수 있다. 또한, 처리량도 MTU의 크기가 어느 정도 이상일 경우에는 일정한 값을 보이고 있다. 모바일 장치(n5)에서의 패킷을 전송하고 남은 에너지를 비교해 보면 MTU가 작을 경우에는 남은 에너지가 적지만 어느 정도 이상이 되면 남은 에너지가 높아지고 일정해 지는 것을 확인할 수 있다. 따라서, MTU의 크기가 어느 정도 크면 에너지의 효율을 높일 수 있기 때

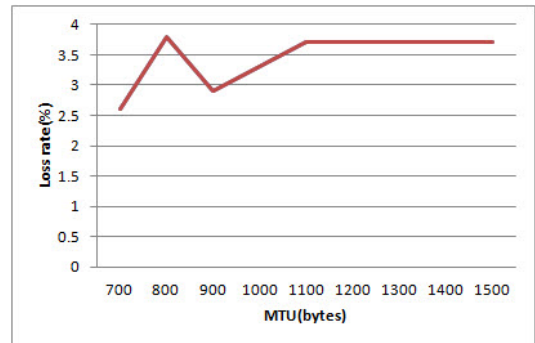


그림 7. MTU 크기에 따른 손실률 비교
Fig. 7. Loss rate comparison for MTU size

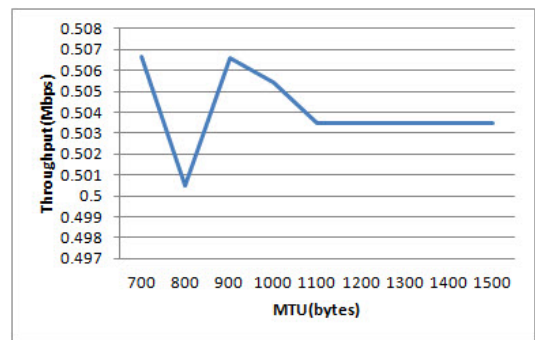


그림 8. MTU 크기에 따른 처리량 비교
Fig. 8. Throughput comparison for MTU size

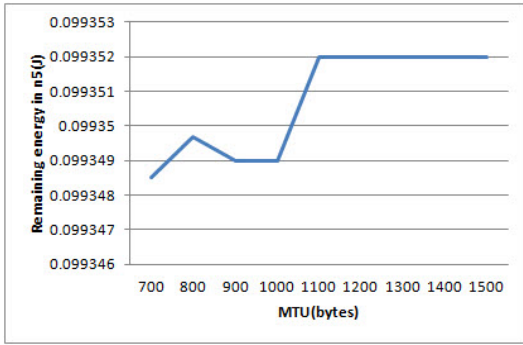


그림 9. MTU 크기에 따른 n5의 남은 에너지
Fig. 9. The remaining energy for MTU size in n5

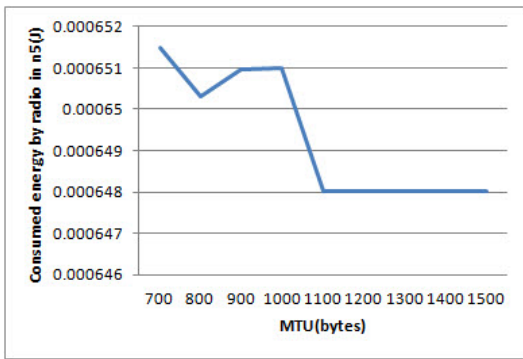


그림 10. MTU 크기에 따른 n5의 전파 소비 에너지
Fig. 10. The consumed energy by radio for MTU size in n5

문에 본 논문의 시뮬레이션에서는 1500byte를 사용했다. 전파에 의한 소비 에너지도 MTU가 작을 경우에는 에너지를 많이 소비하지만 높을 경우에는 적게 소비를 하게 되고 일정해 지는 결과를 도표를 통해서 확인할 수 있다.

3.2.3 모바일 장치에서의 패킷 도착 시간 비교

주어진 토폴로지에서 모바일 장치에서 수신된 패킷의 도착 시간을 시퀀스 번호순으로 비교한 그래프가 아래 그림 11과 같다. 실험을 수행할 때에 손실이 발생하지 않을 때 즉 크로스 플로우가 5개 정도 되었을 경우에는 도착시간이 거의 차이가 없었다. 하지만, 패킷 손실률이 20% 이상이 될 때 즉, 크로스 플로우가 25개 정도 되었을 때에는 처음에는 기존의 방법보다 조금은 빠르다가 나중에는 도착시간이 늦어지는 것을 확인할 수 있다. 수신된 시간이 크로스 플로우가 25개 일 때 더 빨리 수신된 것처럼 나타나는 이유는 나머지 패킷은 트래픽으로 인한 손실이 발생하였기 때문이다. 혼잡이 덜 발생하는 구간에서는 기존의 방법과 비슷

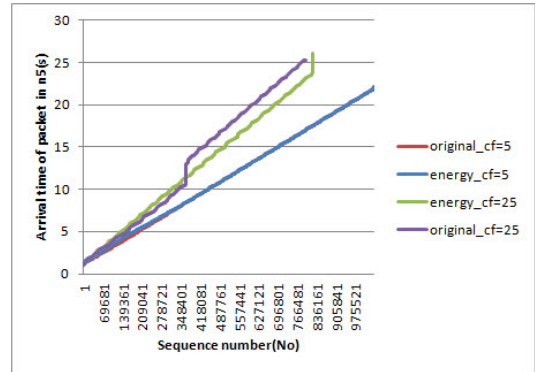


그림 11. n5에서의 패킷 도착 시간
Fig. 11. Arrival time of packet in n5

한 성능을 유지하다가 혼잡이 많이 발생하는 구간에서는 전송 속도를 늦춰 줌으로써 혼잡을 피하고 패킷을 전송하며 대역폭을 100% 사용하지 않고 안정적인 전송을 유지하게 된다.

3.2.4 Yellow-Light TCP의 성능 비교

본 논문에서는 3가지의 주요 원리를 통해서 에너지 효율성을 증가시키고자 하였다. 첫 번째는 기존의 대역폭 예측 방법을 통계적인 방법을 사용하여 대역폭 예측 신뢰도를 높이고자 하였으며, 두 번째 방법은 혼잡 상황이 발생할 경우에 여유대역폭을 확보하는 방법으로 DUP ACK 가 발생하였을 경우에 혼잡 윈도우의 값을 예측된 임계값의 70% 정도로 설정하여 전송하면서 혼잡 상황에 유연하게 대처하도록 하고 패킷 손실로 인한 재전송을 할 때와 RTO가 발생하였을 경우에 패킷 전송을 잠시 보류하는 방법으로 혼잡 윈도우의 값을 0으로 설정하여 전송함으로써 모바일 장치에서의 에너지 소비를 줄이고자 하였다.

새로운 TCP를 사용한 방법의 에너지 절감을 확인하기 위해서 재전송을 포함한 전송한 데이터의 총량을 비교해 보았다. 크로스 플로우가 15일 때에는 손실률이 거의 발생하지 않지만 크로스 플로우가 21일 경우에는 제안된 방법의 손실률이 좀 더 줄어는 것을 확인할 수 있다. 또한, 오버헤드를 포함한 같은 양의 데이터를 보냈을 때, 전송 횟수가 영향을 주는지 확인하기 위해서 크로스 플로우가 30 일 때 1000개의 패킷을 전송하여 아래와 같은 결과가 나타난 상태에서 DUP ACK의 개수를 세어 보았더니 기존의 방법이 52개, 제안된 방법이 51개의 DUP ACK가 발생하였다. DUP ACK가 많이 발생하면 재전송이 많이 발생하게 되고 손실률에 영향을 주게 되는데 차이가 없는 것으로 보아 재전송 횟수와는 관계가 없는 것으로 확

인되었다.

본 논문에서는 혼잡 상황이 발생할 경우에 즉, DUP ACK가 발생할 경우에 혼잡 윈도우의 값을 임계값의 70% 정도로 설정하여 전송률을 조절하게 된다. 그리고 혼잡 상황이 너무 가중된다면 재전송 할 경우에 CWND를 0 상태로 만들어서 전송하면서 잠시 기다렸다가 전송하는 메카니즘을 구현하였다. 만약, DUP ACK가 연속적으로 3번 이상 발생한 구간에 대해서는 네트워크 상황이 많이 안좋다고 판단하여 혼잡 윈도우의 값을 0으로 설정하여 전송하게 된다. 패킷 전송을 잠시 보류하는 구간은 DUP ACK가 많

표 5. 크로스 플로우에 따른 데이터의 총량 비교
Table 5. Comparison for total amount of data

Packet size: 1040byte, transmitted data: 1000, cross traffic=21			
	Received packet(ea)	Throughput (Mbps)	Loss rate(%)
Yellow-Light TCP	910	0.301322	9
TCP-Westwood	897	0.295315	10.4
Packet size: 1040byte, transmitted data: 1000, cross traffic=15			
Yellow-Light TCP	999	0.340404	0.1
TCP-Westwood	999	0.345485	0.1

표 6. 대역폭 예측 방법에 따른 에너지 소비 비교
Table 6. Energy consumption for bandwidth

Modified bandwidth estimation				
	Throughput(Mbps)	Loss rate(%)	Consumed energy(J)	Remained energy(J)
Yellow-Light TCP	0.301322	9	0.000924941	0.0990751
TCP-Westwood	0.295315	10.4	0.000957611	0.0990424

표 7. 여유 대역폭 확보와 전송 보류로 인한 에너지 비교
Table 7. Energy consumption for suggestion

Remaining bandwidth capacity, waiting transmit a packet				
	Throughput(Mbps)	Loss rate(%)	Consumed energy(J)	Remained energy(J)
Yellow-Light TCP	0.294923	10.6	0.000957429	0.0990426
TCP-Westwood	0.295315	10.4	0.000957611	0.0990424

이 생기는 혼잡 상황이 많이 발생하는 구간에서 설정하게 된다. 이러한 방법이 에너지 효율성에 얼마나 적합한지를 실험을 통하여 증명하였다.(표 6, 7)

제안된 대역폭 예측 방법으로 손실률이 약 1.4% 감소 되었으며 여유 대역폭 확보 방법과 전송 보류의 방법으로 손실률은 약 0.2% 늘어 났지만 남은 에너지를 절약한 것으로 에너지 효율성을 나타냈다.

기존의 방법과 Yellow-Light TCP의 크로스 트래픽의 변화율에 따른 손실률을 알아보기 위해 손실률이 0%인 크로스 트래픽 플로우의 개수 15 개부터 실험한 결과를 나타낸 그래프가 다음과 같다.

그림 12는 송신자인 n0에서 모바일 장치 n5로 1000개의 패킷을 전송한 후에 n5의 남은 에너지를 측정한 결과이다. 크로스 플로우를 15개부터 설정하여 테스트 한 이유는 손실률이 발생하기 시작할 때부터 측정한 것이다. 트래픽으로 인한 오류가 발생하기 시작할 때부터 남은 에너지가 기존의 방법보다 높음을 확인할 수 있고 기존의 방법은 트래픽이 일정 개수를 초과하게 되면 에너지가 많이 떨어지는 현상이 있음을 확인하였다. 사용자가 많은 네트워크 구간에서는 제안된 방법이 모바일 장치의 에너지 효율을 높이는 데 효과적일 수 있다.

그림 13은 크로스 트래픽에 따른 손실률을 나타낸 것이다. 기본적으로 네트워크 트래픽이 많이 발생하면 손실률이 그에 비례해서 증가하게 된다. 손실되는 패킷 수가 많아지게 되면 송신자측에서 재전송이 많아지게 되고 수신에 필요한 에너지도 증가하게 된다. 그림 12의 경우, 제안된 방법이 손실률이 거의 발생하지 않는 경우인 cross traffic이 15일 경우에는 에너지 효율성이 거의 차이가 발생하지 않는다. 하지만 cross traffic이 점차 많아지고 네트워크 상황의 혼잡, 간섭 등의 다양한 원인으로 인하여 패킷 손실률이 증가하게 되면 에너지 효율성이 기존의 방법보다 더 좋을

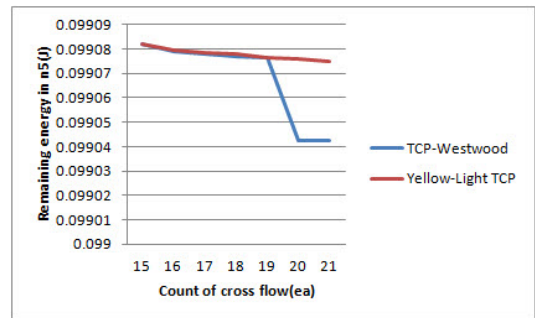


그림 12. 크로스 플로우에 따른 n5의 남은 에너지
Fig. 12. The remaining energy for cross flow in n5

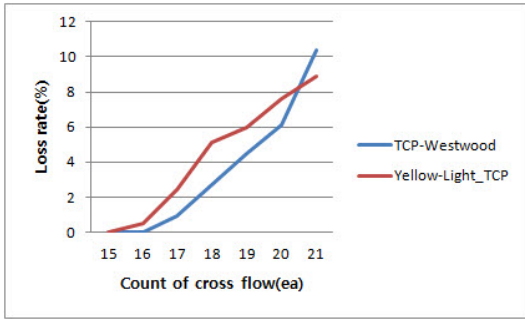


그림 13. 크로스 플로우에 따른 손실률 비교
Fig. 13. The comparison of loss rate for cross flow

나타내고 있다. 그림 13에서는 손실률이 거의 발생하지 않는 구간에서는 비슷한 경향을 보이다가 손실률이 증가하는 경우에 대해서는 제안된 방법이 손실률이 더 높게 나온 것을 확인할 수 있다. 본 논문에서는 손실률이 조금 발생하더라도 에너지 효율을 높이고자 하여 토폴로지를 구성하였고 흥미로운 점은 cross traffic이 많아지면 많아질수록 손실률도 증가할 것 같았는데 제안된 방법으로 인해 손실률이 역전되는 경향을 보이고 있음을 알 수 있다. 따라서, 네트워크 상황이 많이 않좋은 때 제안된 방법이 에너지 효율 면에서 좋은 성능을 발휘할 수 있을 것으로 기대한다.

그림 14는 크로스 트래픽에 따른 처리량을 나타낸 것이다. 모바일 장치에서 데이터 패킷을 많이 처리하게 되면 배터리 소모를 촉진 시켜 에너지 효율성이 저하된다. 주어진 시간에 수신된 패킷의 양이 클수록 처리량은 증가하게 되는데, 혼잡이 발생할 경우에 전송을 보류하거나 주어진 전체 대역폭을 사용하지 않는 방법을 사용하여 처리량은 기존보다 감소하게 된다. 하지만, 혼잡을 피하고 여유 대역폭을 남겨두어 플로우의 손실률을 방지하고 다른 플로우와의 간섭을 줄여 에너지를 줄일 수 있는 장점도 가지고 있다. 또한,

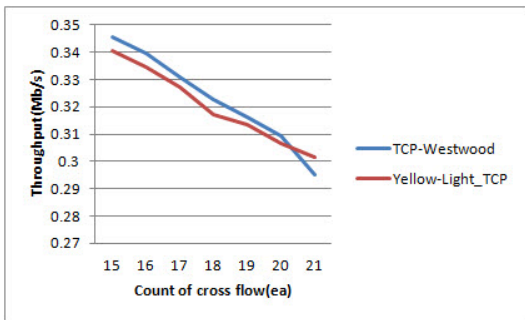


그림 14. 크로스 플로우에 따른 처리량 비교
Fig. 14. The comparison of throughput for cross flow

트래픽이 많이 발생하는 구간에서는 제안된 방법이 오히려 처리량이 좋게 나오는 현상을 확인할 수 있다. 이와 같은 현상은 혼잡이 많이 발생하여서 손실이 버스트하게 발생하게 되고 처리량의 감소와 에너지의 소비의 급격한 증가를 초래하는 구간에서 제안된 방법이 효율적임을 확인할 수 있다.

IV. 결 론

무선 네트워크 환경에서의 사용자가 갈수록 급증하고 있고 그에 따른 모바일 장치의 사용과 대용량 데이터 전송 및 수신 요구가 증대됨에 따라 에너지와 관련한 다양한 이슈와 해결책이 제시되고 있는 시점에서 본 논문에서는 네트워크 전송 프로토콜인 TCP의 메커니즘을 사용해서 에너지 효율을 증대시키고자 Yellow-Light TCP라는 새로운 프로토콜을 제안하였다. 논문의 기본적인 아이디어는 혼잡 상황에서 패킷을 전송하게 될 때 혼잡 상황을 피하여 패킷을 전송함으로써 모바일 장치에서의 에너지 사용량을 줄이고자 하는데 목적이 있다. 혼잡 상황에서 패킷 전송률을 조절하는 방법에 있어서 세가지 메커니즘을 제안하였고, 그 첫 번째가 ACK 도착시간을 사용한 통계적 대역폭 예측 방법이며, 두 번째는 DUP ACK가 발생하였을 경우에 재전송 패킷을 잠시 보류하는 방법이다. 세 번째는 재전송 패킷을 전송할 경우에 예측 대역폭의 100%를 사용하지 않고 약, 70% 정도를 사용하여 혼잡 윈도우를 조절하면서 전송하게 된다. 이러한 방법이 네트워크 환경에서 혼잡 상황이 많이 발생할 경우에 보다 효과적일 것으로 기대한다. 네트워크 상황이 수시로 변하는 무선 네트워크 환경에서 네트워크 트래픽이 많이 발생하는 구간에서는 패킷 전송을 잠시 보류하고 대역폭이 호전되었을 경우에는 대역폭을 100% 사용하지 않고 여유 대역폭을 남겨두고 전송함으로써 다른 사용자와의 간섭과 패킷 손실을 줄이면서 재전송 증가를 방지함으로써 모바일 장치의 에너지를 절약하고자 하였다. 실험에서 사용된 여유대역폭은 대략 30% 정도 남기고 전송하는 것이 효율적인 것을 확인했으며, MTU 크기는 1500byte로 설정되었다. 혼잡이 발생하지 않을 때에는 모바일 장치에서 소비된 에너지가 차이가 없지만 사용자가 많은 구간 즉, 트래픽이 많이 발생하는 구간에서는 제안된 방법이 0.0000336(J) 을 절약하는 것으로 검증되었다. 반면에 처리량은 크로스 플로우가 15일 때에는 0.005081(Mbps) 차이가 났으며, 20일 경우에는 0.002591(Mbps) 차이가 났다. 네트워크 환경에서 트

래픽이 많이 발생하는 경우에는 처리량의 차이가 많이 감소하는 경향을 보이고 있다. 따라서, 제안된 방법이 무선 네트워크 환경에서 사용자가 많을 경우에 에너지 효율을 어느 정도 높일 수 있을 것으로 예상된다. 본 논문은 전송 프로토콜에서 전송 메커니즘을 수정하여 에너지 효율성을 높이고자 하였다. 기본적으로 제안한 알고리즘을 증명하기 위해서 메인 플로우를 TCP로 설정하였고 크로스 트래픽은 UDP flow로 설정하였다. 추후에 flow에 대한 fairness에 대한 부분을 고려하여 더 발전된 연구를 할 계획이다.

References

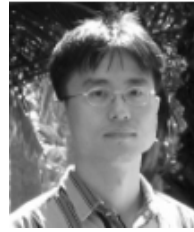
- [1] H. Singh and S. Singh, "Energy consumption of TCP reno, newreno, and SACK in multi-hop wireless networks," in *Proc. ACM SIGMETRICS Performance Evaluation Review Measurement and modeling of computer systems*, vol. 30, no. 1, pp. 206-216, Jun. 2002.
- [2] B. Wang and S. Singh, "Computational energy cost of TCP," in *Proc. IEEE INFOCOM*, pp. 785-795, Hong Kong, Mar. 2004.
- [3] Y. J. Jang and S. K. Lee, "An energy efficient transition algorithm based on TCP," in *Proc. KIECS*, vol. 5, no. 2, pp. 394-398, Nov. 2011.
- [4] S. Hwang, J. Lee, and K. Chung, "A new energy saving transport protocol in wireless environments," in *Proc. KIISE Computer Systems and Theory*, vol. 32, no. 12, pp. 654-662, Dec. 2005.
- [5] S. Hwang and S. Mo, "A study on efficient energy saving transport protocol in wireless environment," in *Proc. KIISE*, vol. 32, no. 2, pp. 586-588, Feb. 2004.
- [6] N. Cho and K. Chung, "Energy efficient congestion control scheme in ad-hoc networks," in *Proc. KIISE Inf. Netw.*, vol. 33, no. 5, pp. 369-379, Oct. 2006.
- [7] M. Gerla, R. Wang, and A. Zanella, "TCP westwood: Congestion window control using bandwidth estimation," in *Proc. IEEE Globecom*, vol. 3, pp. 1698-1702, USA, Nov. 2001.
- [8] G. Smit and P. Havinga, "Energy efficient TCP," in *Proc. Int. Conf. AMOC Electrical Eng., Math. and Comput. Sci.(EEMCS)*, pp. 18-28, Malaysia, May 2001.
- [9] M. Park and H. Choo, "Analysis and modified algorithm proposal of TCP westwood in next generation network," in *Proc. Korean Soc. Internet Inf.(KSII)*, vol. 8, no. 2, pp. 27-30, Aug. 2007.
- [10] M. Oulmahdi, C. Chassot, and E. Exposito, "An energy-aware TCP for multimedia streaming," in *Proc. Smart Commun. in Netw. Technol.(SaCoNeT)*, pp. 1-5, France, Jun. 2013.
- [11] M. Jada, et al., "Power efficiency model for mobile access network," in *Proc. Inst. Electrical and Electronics Engineers(IEEE)*, pp. 317-322, Istanbul, Sept. 2010.
- [12] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. The Advanced Computing Systems Association (USENIX)*, pp. 21-21, USA, Mar. 2010.
- [13] S. Mascolo and C. Casetti, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. Mobicom*, pp. 287-297, USA, Jul. 2001.
- [14] S. Agrawal and S. Singh, "An experimental study of TCP's energy consumption over a wireless link," in *Proc. Eur. Personal Mob. Commun. Conf. (EPMCC)*, pp. 13-18, Austria, Feb. 2001.
- [15] S. Lee, H. An, and M. Kim, "A method to resolve TCP packet out-of-order and retransmission problem at the traffic collection point," *J. KICS*, vol. 39B, no. 6, pp. 350-359, Jun, 2014.
- [16] W. Seok, M. Lee, and M. Lee, "Receiver-initiated slow start for improving TCP performance in vertical handoff," *J. KICS*, vol. 38B, no. 8, pp. 597-606, Aug. 2013.
- [17] S.-K. Lee, H.-M. An, and M.-S. Kim, "A method to resolve TCP packet out-of-order and retransmission problem at the traffic collection point," *J. KICS*, vol. 39B, no. 6, pp. 350-359, Jun. 2014.
- [18] NS-3 Simulator, <http://www.nsnam.org/>

최 원 준 (Won-jun Choi)



2006년 2월 : 원광대학교 수학
통계학과 학사
2013년 2월~현재 : 과학기술연
합대학원대학교 그리드 및
슈퍼컴퓨팅학과 석박통합과
정 재학중
<관심분야> 통신공학, 네트워
크 통신, TCP 성능 분석

석 우 진 (Woo-jin Seok)



1996년 2월 : 경북대학교 컴퓨터
공학과 학사
2003년 2월 : Univ. North
Carolina, Computer Science
석사
2008년 2월 : 충남대학교 컴퓨
터공학과 박사
<관심분야> 무선/이동 QoS, TCP 성능 분석

람 닉 (Ramneek)



2010년 2월 : Univ. Guru
Nanak Dev 컴퓨터 공학과
졸업
2013년 8월~현재 : 과학기술연
합대학원대학교 그리드 및
슈퍼컴퓨팅학과 박사과정 재
학중

<관심분야> 무선 통신, 모바일 통신