

# 무선 환경에서 TCP 슬로우 스타트의 성능 분석 및 개선 방안

구본욱, 김유경, 양시훈, 유준

가천대학교

{chris3565, kimy0316, tlgnsihun, joon.yoo}@gachon.ac.kr

## Performance Analysis and Improvement of TCP Slow Start in Wireless Environments

Koo Bon Wook, Kim Yu Kyung, Yang Si Hun, Yoo Joon  
Gachon Univ.

### 요약

본 논문은 모바일 네트워크 환경에서의 급격한 대역폭의 변동으로 인하여 SUSS (Speeding Up Slow-Start)의 Slow Start 가속화 방식이 가지는 성능 한계점을 분석한다. 또한 이를 개선하기 위하여 기존 SUSS 코드를 수정하여 다양한 조건에서 실험을 진행하고 성능을 분석한다. 실험 결과, 본 제안은 CUBIC 대비 평균 약 2%의 전송 시간 단축을 기록하였으며, 최대 6.1%의 성능 개선을 달성하였다.

### I. 서론

현대 모바일 네트워크 환경에서의 데이터 전송은 다중 지연 요소의 삽입과 급격한 대역폭의 변동으로 인해 높은 신뢰성과 효율성을 동시에 확보하는 데 큰 도전에 직면하고 있다. 특히, 4G 및 5G 와 같은 최신 이동통신 환경에서는 전송 지연(RTT)의 변동성(jitter)과 패킷 손실률이 유선 네트워크보다 높은 경향을 보인다.

이러한 특성은 전송 프로토콜의 성능 저하로 직결되며, 특히 전송 초기에 가용 대역폭을 효과적으로 활용하지 못하는 문제를 야기한다. 기존의 TCP 슬로우 스타트(slow-start) 메커니즘은 혼잡 윈도우(cwnd)를 점진적으로 증가시키며 대역폭을 탐색한다 [1]. 그러나 모바일 환경의 높은 변동성으로 인해 가용 대역폭을 충분히 활용하지 못하고, 이로 인해 전송 지연과 성능 저하 문제가 발생한다.

이러한 문제를 해결하기 위해 제안된 SUSS (Speeding Up Slow-Start)[2]는 슬로우 스타트 초기에 혼잡 윈도우를 보다 빠르게 증가시키기 위해 패킷 pacing 과 ACK clocking 을 조합하여 사용한다. SUSS는 네트워크 혼잡이 발생하지 않은 것으로 판단되는 조건에서 cwnd 의 지수 증가를 가속화함으로써, 특히 소규모 TCP 플로우에서 전송 속도를 크게 향상시킨다. 기존 연구에 따르면, SUSS는 다양한 유선 네트워크 환경에서 전통적인 CUBIC 대비 평균 20% 이상의 전송 지연 개선 효과를 보였다.

하지만, SUSS는 혼잡 예측을 위해 HyStart 메커니즘에 의존하므로[3] RTT 가 급변하거나 ACK 간격이 불안정한 모바일 환경에서는 예측 조건을 충족하지 못할 가능성이 높다. 그 결과 SUSS는 지나치게 보수적으로 작동하거나, 애초에 작동 조건을 충족하지 못하여 성능 개선 효과를 발휘하지 못하는 한계를 갖는다.

본 연구는 이러한 문제 인식을 바탕으로, SUSS의 모바일 네트워크 성능을 개선할 수 있는 가능성을 탐색하고자 한다.

### II. 본론

SUSS는 혼잡 예측 기반의 가속 슬로우 스타트 방식으로 유선 환경에서는 일정 수준의 성능 향상을 보이지만, 모바일 네트워크와 같이 높은 RTT 변동성과 불안정한 ACK 간격이 존재하는 환경에서는 그 효율성이 제한적일 수 있다.

본 연구는 이러한 SUSS의 한계를 해결하기 위해, 초기 혼잡 감지 및 pacing 로직이 모바일 환경에서도 효과적으로 작동할 수 있도록 파라미터 튜닝을 적용하는 방안을 제안한다. 특히 기존 SUSS 코드의 조건 판별 로직에 포함된 HyStart 기반 임계 값들을 조정함으로써, 보수적인 혼잡 예측을 완화하고 가용 대역폭을 더 능동적으로 활용하는 방향으로 설계를 진행하였다.

이하에서는 실험 환경, 설계 방법, 그리고 측정 결과에 대해 구체적으로 서술한다.

#### 1. 실험 환경

본 실험은 SUSS 기반 커널이 적용된 AWS EC2 인스턴스(Ubuntu 24.04)를 송신 서버로 구성하였으며 [4], 수신자는 Wi-Fi 에 연결된 MacBook 을 사용하여 모바일 네트워크 환경을 재현하였다.

전송 도구	Iperf3
분석 도구	tcpdump, Wireshark
커널 설정	/sys/module/tcp_cubic/parameters 내의 suss, suss_kmax, suss_max 제어

## 2. 실험 방법

세 가지 실험 조건(CUBIC, SUSS, SUSS\_TUNED)과 0.2MB, 2MB, 5MB 전송 크기로 실험을 설계하였다. 튜닝은 HyStart 기반 혼합 예측의 과도한 보수성을 완화하고 초기 페이싱을 적극적으로 조정하는데 초점을 맞추었다.

각 전송 크기당 실험은 50 회 반복 시행하였으며, 평균값 및 분산을 수집해 분석에 활용하였다. 실험 조건의 상세 내용은 다음과 같다.

실험 조건	설명
CUBIC	기본 리눅스 커널 TCP CUBIC
SUSS	커널 빌드 시 SUSS 적용 및 suss=1
SUSS_TUNED	suss_kmax=9, suss_max=0.6 로 튜닝

## 3. 실험 결과 및 분석

### 3-1 전송 시간 비교

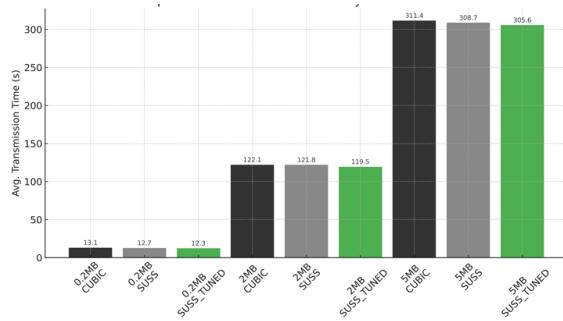


그림 1. 전송 시간 비교

위는 각 실험 조건에서 전송 크기별 전송 시간을 나타낸 그레프이다. 실험 결과, SUSS 는 무선 환경에서 CUBIC 대비 0.9% ~ 3%의 성능 개선을 보였다. 이는 SUSS 가 CUBIC 보다 일정 수준 이상의 성능 개선을 제공함을 보여준다. 특히 SUSS\_TUNED의 경우 CUBIC에 비하여 0.2MB 인 경우 6.1%, 2MB 인 경우 2.1%, 5MB 인 경우에서 1.9%의 성능 개선을 이루어 냈다. 이를 통해 데이터 전송량이 적을수록 성능 개선 폭이 더 크게 나타남을 확인할 수 있었다. 전반적으로 SUSS\_TUNED 는 모든 전송 구간에서 CUBIC 대비 우수한 성능을 보여주었다.

### 3-2 PPS(Packet Per Second) 비교

아래는 각 실험 조건에서 초당 패킷 수 (PPS)를 비교한 결과를 나타낸 그레프이다. 0.2MB 구간에서는 SUSS\_TUNED 가 가장 높은 PPS를 기록하며 작은 전송량에서도 보다 빠른 패킷 처리 성능을 보였다. 2MB 및 5MB 구간에서도 SUSS\_TUNED 가 CUBIC 및 기존 SUSS 대비 미세하게 높은 PPS를 유지하였다.

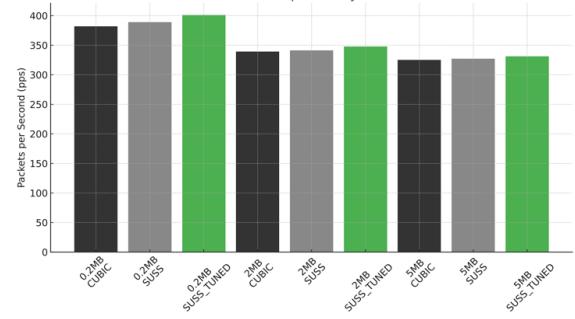


그림 2. PPS 비교

이는 파라미터 튜닝을 통해 SUSS 의 초기 pacing 을 보다 공격적으로 수행함으로써 패킷 처리 효율을 개선한 결과임을 알 수 있다.

## III. 결론

본 논문에서는 모바일 네트워크 환경에서 CUBIC 과 SUSS 의 성능을 비교하고, SUSS 의 파라미터를 튜닝하여 성능 개선의 가능성을 실험적으로 분석하였다. 실험은 전송량을 0.2MB, 2MB, 5MB 로 구분하고, 각 조건 (CUBIC, SUSS, SUSS\_TUNED)에서 전송 시간, PPS 를 평가하였다.

실험 결과, 튜닝 된 SUSS\_TUNED 는 모든 전송량에서 CUBIC 대비 평균 약 2%의 전송 시간 단축을 기록하였으며, 특히 전송량이 0.2MB 인 경우 6.1%의 성능 개선을 달성하였다. 또한, PPS 수치 역시 SUSS\_TUNED 가 다른 조건들에 비해 전반적으로 높아 패킷 처리 효율에서도 우위를 확보하였다.

본 연구는 모바일 환경에서 SUSS 의 성능 한계를 보완 할 수 있는 파라미터 튜닝의 실효성을 입증했으며, 향후 모바일 네트워크에서의 적용 가능성과 최적화 방안 마련에 기여할 수 있을 것으로 기대된다.

## ACKNOWLEDGMENT

본 연구는 2021 년도 과학기술정보통신부 이공분야기초사 업의 지원(NRF-2021R1F1A1063640) 및 2020 년도 과학 기술정보통신부 및 정보통신기획평가원의 SW 중심 대학사업 지원을 받아 수행하였음.

## 참 고 문 헌

- [1] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: a new TCP-friendly high-speed TCP variant. SIGOPS Oper. Syst. Rev. 42, 5 (July 2008), 64– 74.
- [2] Mahdi Arghavani, Haibo Zhang, David Eyers, and Abbas Arghavani. 2024. SUSS: Improving TCP Performance by Speeding Up Slow-Start. In Proceedings of the ACM SIGCOMM 2024 Conference (ACM SIGCOMM '24). Association for Computing Machinery, New York, NY, USA, 151– 165.
- [3] P. Balasubramanian, Y. Huang, and M. Olson. 2023. RFC 9406: HyStart++ : Modified Slow Start for TCP. RFC Editor, USA.
- [4] Canonical Ltd., “Ubuntu Documentation,” [Online]. Available: <https://help.ubuntu.com/lts/ubuntu-help/index.html> [Accessed: May 3, 2025].