

Programming Assignment

Improving the performance TCP Reno algorithm

Computer Networks_SWE3022_42

Due date : Dec. 12, 2025

학과: 소프트웨어학과

학번: 2022315902

이름: 곽태운

- **개요**

이번 과제의 목표는 리눅스 운영체제에 구현된 TCP Reno 알고리즘의 Congestion Control 방법의 한계를 분석하고, 이를 해결한 후 검증하는 것이다. 이를 위해 중간 보고서를 작성하는 현재까지는 Mininet 기반의 가상 네트워크 환경을 구현하고, Linux 커널 모듈의 TCP Reno 를 활용하여, TCP Reno 의 특징과 한계를 분석하였습니다.

- **과제 요구 환경 설정**

과제를 수행하기 위해 아래와 같이 Ubuntu 환경에서 리눅스 커널을 다운받아 TCP Reno 를 실험할 수 있는 환경을 구성했습니다.

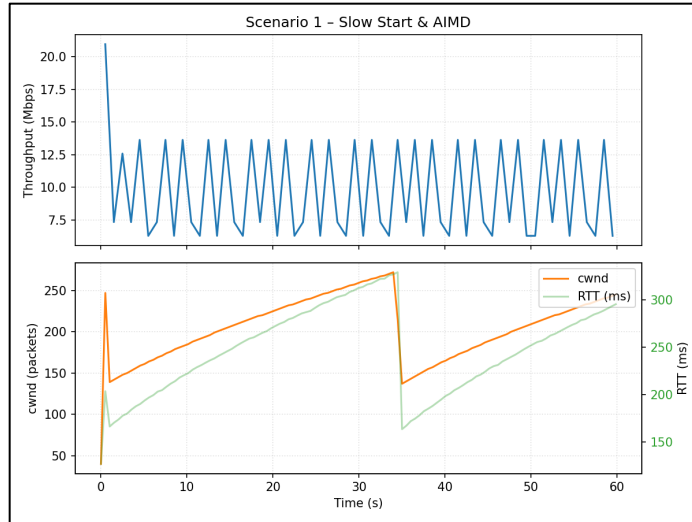
- 실습 환경: Ubuntu 22.04
- 개발 및 네트워크 도구: net-tools iproute2 ethtool
- 성능 측정 및 에뮬레이터: iperf3 mininet
- 커널 헤더/소스: linux-headers-6.14.0-33-generic

- **TCP Reno 실험**

실험 환경에서 아래의 실험들을 통해 TCP-Reno 의 기본적인 특징들을 파악할 수 있었습니다.

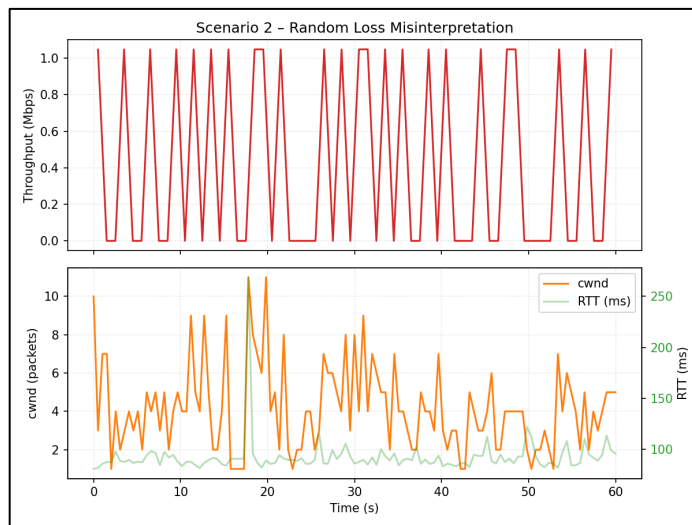
1. **Slow Start 와 AIMD**

- **환경 설명:** 표준 10 Mbps 링크, 30 ms 지연, 무손실 환경
- **관측 가능한 특징:** TCP Reno 는 선형 증가하다가 패킷 손실 발생 시 1/2 로 지수 감소 시킨다. (AIMD: Additive Increase Multiplicative Decrease)



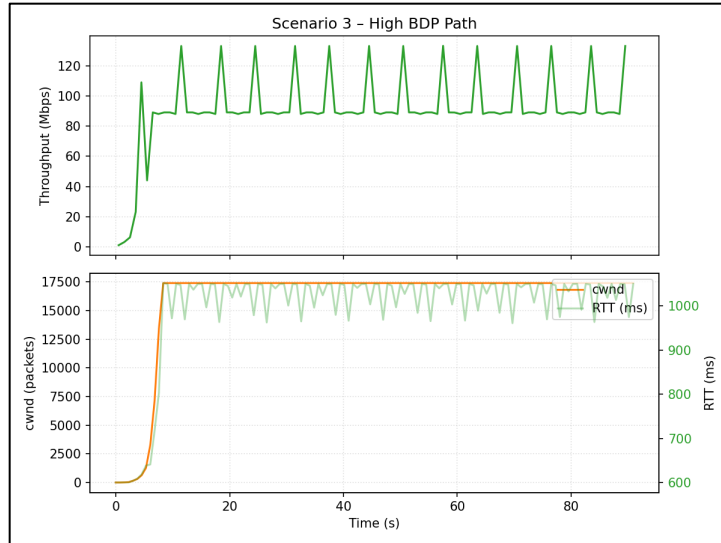
2. 패킷 손실을 네트워크 혼잡으로 판단

- **환경 설명:** 무작위 손실 발생(5%)
- **관측 가능한 특징:** Reno 는 패킷 손실을 네트워크 혼잡으로 판단하고 손실 발생 시 cwnd 를 절반으로 줄이는데, 이러한 특징이 5%의 무작위 손실 발생 환경에서 처리량 급감으로 이어짐.



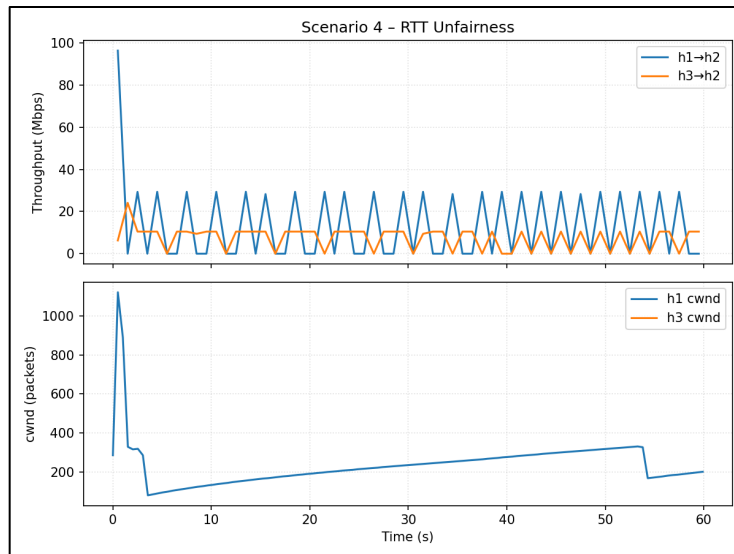
3. 지연(RTT)가 큰 환경에서 네트워크 폭 활용 비효율

- **환경 설명:** 150 ms 네트워크 지연 발생
- **관측 가능한 특징:** RTT 가 크면 전체 네트워크 대역폭 100mbps 를 온전하게 사용하지 못하고 90mbps 만 사용하는 모습 확인



4. 불공정성

- 환경 설명: RTT 가 긴 송신자와 RTT 가 짧은 송신자가 링크 공유
- 관측 가능한 특징: RTT 가 짧은 송신자가 네트워크 대역폭의 대부분을 점유



5. 네트워크 혼잡 민감도 & Fast Recovery

- 환경 설명: 큐가 과도하게 크면 RTT 가 급증하며 지터도 커짐
- 관측 가능한 특징: 큐가 과도하게 크면 RTT 가 급증, Fast Retransmit/Recovery 효과 확인

- 과제 계획

TCP-Reno 의 가장 큰 문제는 모든 패킷 손실을 네트워크 혼잡으로 인한 패킷 손실이라고 판단하여, 송신 속도를 절반으로 줄인다고 판단하였습니다. 이를 해결하기 위해 패킷 손실이 네트워크 혼잡 때문인지 아닌지 분류하는 모듈을 추가하여 그 성능을 확인해보도록 하겠습니다.