

# 고속철도 환경에서 MPTCP 성능 향상을 위한 서브플로우 차단 방안

박창훈\*, 김건환\*, 김정근\*, 엄원주\*, 서상진\*, 조효섭\*, 조유제\*

## Subflow Blocking Method to Improve Performance of MPTCP in High-Speed Train Environments

Chang-hoon Park\*, Geon-hwan Kim\*, Jeong-keun Kim\*, Won-ju Eom\*,  
Sang-jin Seo\*, Hyo-sup Cho\*, You-ze Cho\*

### 요약

MPTCP는 하나의 디바이스에서 다수의 인터페이스를 동시에 사용할 수 있으며, 단일 TCP에 비해 총 데이터 처리량을 향상시킬 수 있다는 장점이 있다. 기존의 MPTCP를 고속철도 환경에 적용할 경우 총 데이터 처리량이 단일 TCP에 비해 상승하였으나, LTE 신호 세기 감소와 핸드오버가 흔히 일어나기 때문에 Out-of-Order 패킷이 정지한 환경에서보다 자주 발생한다. 이로 인해 MPTCP 수신 버퍼에서의 HoL Blocking 문제가 자주 일어나며, 일시적으로 단일 TCP보다 성능이 감소하는 현상이 제시되었다. 이러한 문제점을 해결하기 위해, 본 논문에서는 핸드오버 시에 해당 서브플로우를 일시적으로 차단함으로써 MPTCP의 총 데이터 처리율을 향상시키는 방안을 제안한다. 제안한 서브플로우 차단 방안은 업링크 환경에서만 적용이 가능하다는 단점을 가지고 있으나, NS-3 업링크 시뮬레이션 실험에서 핸드오버 동안의 성능 감소를 완화시켜 기존 MPTCP보다 총 데이터 처리율이 최대 10% 상승하였으며, Out-of-Order 패킷이 약 10% 감소하였다.

**키워드** : 다중 경로 TCP, 고속철도 환경, 스케줄러, Head-of-Line 블로킹, Out-of-Order 패킷

**Key Words** : Multipath TCP, High-speed Train Environment, Scheduler, Head-of-Line Blocking, Out-of-Order Packet

### ABSTRACT

MPTCP uses multiple interfaces simultaneously in one device, and it improves the total throughput compared to single TCP. The total data throughput is similarly improved compared to single TCP when MPTCP is applied to the high-speed train environment, but Out-of-Order packets occur more frequently than stationary environment because of LTE signal strength reduction and handover. Therefore, HoL Blocking problems in the MPTCP receive buffer occur frequently, and a phenomenon that temporarily decreases performance compared to a single TCP has been resulted. To solve this problem, we propose an method to

※ This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1A6A1A03025109) and in part by supported by National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006249).

♦ First Author : School of Electronics Engineering, Kyungpook National University, pch4495@knu.ac.kr, 학생회원

° Corresponding Author : School of Electronics Engineering, Kyungpook National University, yzcho@ee.knu.ac.kr, 종신회원

\* School of Electronics Engineering, Kyungpook National University, kgh76@ee.knu.ac.kr, {kjg818, dnjsw9612, chil258, yg01175}@knu.ac.kr, 학생회원

논문번호 : 202107-183-B-RN, Received July 28, 2021; Revised October 12, 2021; Accepted October 20, 2021

improve the total data throughput of MPTCP by temporarily blocking the corresponding subflow during a handover. The proposed subflow blocking method has the disadvantage that it can be applied only in an uplink environment, but it mitigates the performance decrease during handover in NS-3 uplink simulation experiment and the total data throughput increased by up to 10% compared to the regular MPTCP, and Out-of-Order packets decreased by about 10%.

## 1. 서 론

MPTCP (Multipath TCP)는 TCP (Transmission Control Protocol)의 확장버전 전송계층 프로토콜로 2013년 IETF (Internet Engineering Task Force)의 RFC 6824에서 처음으로 표준화되었으며<sup>[1]</sup>, RFC 8684에서 새로 개정되었다<sup>[2]</sup>. 기존의 TCP는 디바이스에 여러 인터페이스가 존재하더라도 하나의 인터페이스만을 사용하여 데이터를 송수신하니<sup>[3]</sup>, MPTCP는 디바이스에 연결된 경로들을 탐색하여 동시에 사용할 수 있다.

LTE, 5G 등 이동통신 기술이 발달함에 따라 사용자들 또한 다양한 환경에서 높은 품질의 이동통신을 사용하기를 희망하고 있으며, 그 예로 고속철도 환경에서의 TCP 및 MPTCP에 대한 연구가 지속적으로 진행되고 있다<sup>[4,5]</sup>. 중국에서는 하나의 디바이스에 두 개의 SIM 슬롯을 가진 Dual-SIM이 사용되고 있으며, 이 Dual-SIM을 이용한 MPTCP의 연구가 진행되고 있다<sup>[5]</sup>. 국내에서는 하나의 디바이스에 여러 통신사를 동시에 연결하는 Dual-SIM 서비스를 지원하지 않기 때문에, 일반적인 사용자들에게 Dual-SIM을 이용한 MPTCP를 직접적으로 제공하는 것은 어렵다. 하지만 고속철도 객차 내에 WiFi 연결 서비스를 제공하는 모바

일 액세스 라우터에 MPTCP를 적용한다면, 사용자들에게 향상된 네트워크 성능을 제공할 수 있을 것으로 예상된다. <그림 1>은 본 논문에서 고려하는 고속철도 모바일 액세스 라우터에 적용된 MPTCP 예시이다. 고속철도 내의 각 객실 AP는 모바일 액세스 라우터에 유선으로 연결되어있으며, 해당 모바일 액세스 라우터에 MPTCP를 적용하여 두 개 이상의 ISP (Internet Service Provider)에 연결한다면, 두 개의 경로를 동시에 사용할 수 있다.

MPTCP는 이상적인 경우 각 경로 대역폭의 합만큼 데이터 처리율을 상승시킬 수 있다는 장점을 가지고 있다<sup>[6,7]</sup>. 하지만 MPTCP에 구성된 각 서브플로우의 특성이 크게 차이는 환경이라면, MPTCP는 비효율적으로 데이터를 전송하게 되면서 총 데이터 처리율의 감소 또한 발생할 수 있다<sup>[8,9]</sup>. 고속철도 환경에 MPTCP를 적용할 경우 일반적인 유, 무선 환경과 마찬가지로 단일 TCP에 비해 데이터 처리량이 상승할 수 있다. 하지만 LTE 신호 세기 감소와 핸드오버가 자주 일어나는 고속철도 환경에서는 각 서브플로우의 특성이 차이는 경우가 많고<sup>[4]</sup>, MPTCP는 일반적인 정지한 환경에서보다 총 데이터 처리율의 감소가 흔히 발생한다. 최근 여러 연구에 의하면, 고속철도 환경에서 MPTCP를 사용할 경우 단일 TCP보다도 오히려 성능이 떨어질 수 있다는 문제점이 제시되었다<sup>[5]</sup>.

또한, 이용자들이 이동 단말에서의 클라우드 서비스와 스트리밍 어플리케이션 및 SNS 등 다양한 목적으로 네트워크를 사용하게 되면서 업링크 트래픽이 꾸준히 증가하고 있다<sup>[10]</sup>. 이러한 이유로, 다운링크뿐만 아니라 업링크 또한 꾸준히 연구가 진행되는 추세이다<sup>[11]</sup>.

본 논문에서는 업링크 환경의 트래픽 증가, 그리고 고속철도 환경에서의 MPTCP의 문제점을 해결하기 위해 먼저 핸드오버가 흔히 발생하는 고속철도 환경에서의 단일 TCP와 기존 MPTCP의 업링크 성능을 시뮬레이션 실험을 통해 분석한다. 이후, 업링크에서 핸드오버 여부에 따라 서브플로우를 일시적으로 차단하는 스케줄링 알고리즘을 통해 MPTCP의 성능 감소를 완화하는 방안을 제안하고자 한다.

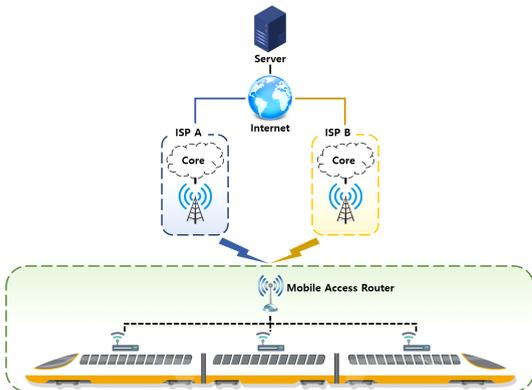


그림 1. 고속철도 모바일 액세스 라우터에 적용된 MPTCP 예시  
Fig. 1. MPTCP for mobile access router in high-speed train

## II. 연구 배경

### 2.1 MPTCP 스케줄러

MPTCP는 패스 매니저, 혼잡제어 알고리즘, 스케줄러로 크게 3가지의 중요한 요소로 구성된다<sup>[12]</sup>. 패스 매니저는 디바이스에 연결된 인터페이스들의 IP 주소 쌍을 활용하여 서브플로우를 생성한다. 혼잡제어 알고리즘은 네트워크가 패킷 손실 혹은 병목 링크의 발생 등 네트워크의 혼잡이 일어난 경우, 혼잡 윈도우를 조절하여 데이터를 얼마나 전송할지를 결정하는 역할을 한다<sup>[13]</sup>. 마지막으로, 스케줄러는 보낼 수 있는 데이터를 스케줄링 요소를 통해 사용 가능한 서브플로우들 중 어느 서브플로우를 통해 얼마나 보낼 것인지를 할당하는 역할을 하며, 전반적인 성능을 결정하는 중요한 요소 중 하나이다<sup>[14,15]</sup>.

스케줄러는 데이터 레벨 시퀀스 넘버를 사용하여 선택된 서브플로우를 통해 수신자가 신뢰할 수 있는 순서로 데이터를 전송한다. 이 데이터 레벨 시퀀스 넘버는 수신자가 각 서브플로우로부터 전송받은 데이터를 취합하여 올바른 순서대로 패킷을 정렬하는데 사용된다. 이때, 이중망 네트워크, 즉 각 서브플로우의 지연시간, 대역폭, 혹은 패킷 손실 등 특성이 다를 경우, 송신 측에서는 순서대로 전송한 패킷들이 송신 측에서 의도하지 않은 시간에 수신 측으로 도착하게 되면서 올바르게 받은 순서의 패킷 발생, 즉 Out-of-Order 패킷 문제가 발생한다<sup>[16]</sup>. Out-of-Order 패킷은 올바른 순서의 패킷이 오기 전까지 수신 버퍼에 머물러 있게 되고, 이러한 문제가 지속되어 수신 버퍼가 가득 차게 될 경우, 올바른 순서의 패킷이 수신되기 전까지 패킷을 일시적으로 차단하는 현상이 일어날 수 있다. 이를, MPTCP 수신 버퍼에서의 HoL Blocking (Head-of-Line Blocking) 문제라고 부른다<sup>[17]</sup>. 따라서 경로의 특성을 고려하지 않고 스케줄링을 하게 된다면 MPTCP의 성능 저하를 야기할 수 있으며, 이러한 문제점을 해결하기 위하여 다양한 스케줄러들이 제시되었다<sup>[18-20]</sup>.

### 2.2 고속철도 환경에서의 MPTCP

MPTCP는 최대 각 경로의 대역폭의 합만큼 데이터 처리율을 상승시킬 수 있으나, 특정 환경에서는 총 데이터 처리율이 감소할 수 있다. 이러한 문제점은 지연 시간 및 대역폭 등 회선의 특성이 상당히 차이는 환경에서 가장 명확하게 드러나며, 대표적으로 UE Node (User Equipment Node)가 빠른 속도로 이동하

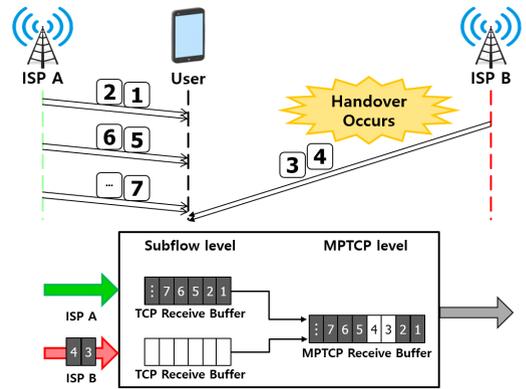


그림 2. 핸드오버 발생시 Dual-LTE MPTCP 환경에서의 데이터 흐름  
Fig. 2. Data flow from Dual-LTE MPTCP when handover occurs

는 고속철도 환경을 예시로 들 수 있다.

고속철도 환경에서 서로 다른 ISP의 기지국에 각 서브플로우를 연결하여 MPTCP를 구성할 경우, 고속철도 환경의 특성상 LTE 신호 세기의 감소와 핸드오버가 자주 발생한다. <그림 2>는 핸드오버 발생시 Dual-LTE MPTCP 환경에서의 데이터 흐름을 나타낸 것이다. <그림 2>에서 ISP B에서 핸드오버가 발생하면서 3, 4번 패킷의 전송이 중단되고, 그동안 ISP A를 통해 5, 6번 패킷을 추가로 전송한다. 이때 핸드오버로 인한 예기치 않은 지연시간의 증가 혹은 패킷 손실이 일어난다면 MPTCP 수신 측에서는 올바르게 받은 순서로 패킷이 도착함에 따라 Out-of-Order 패킷이 발생하고, 이러한 현상이 주기적으로 반복되는 고속철도 환경에서는 MPTCP 수신 버퍼에서의 HoL Blocking 문제가 더욱 자주 일어난다. 최근 여러 연구결과에 따르면, 핸드오버가 자주 발생함에 따라 단일 TCP보다 낮은 성능을 보이는 현상이 시뮬레이션 실험과 실제 네트워크 환경의 실험 모두에서 자주 나타난다고 보고되었다<sup>[4,5]</sup>.

### 2.3 고속철도 환경에서의 LTE 시뮬레이션 실험

고속철도 환경에서 MPTCP의 문제점을 확인 및 분석하기 위하여, NS-3 시뮬레이터<sup>[21]</sup>에 고속철도 환경을 구성하여 실험을 진행하였다.

<그림 3>은 LTE 핸드오버 시뮬레이션을 진행하기 위한 고속철도 실험환경을 나타내었으며, <표 1>은 실험에 사용된 시뮬레이션 파라미터들을 정리하였다. 고속철도 모바일 액세스 라우터 역할을 하는 UE Node는 서로 다른 ISP를 사용하여 MPTCP를 구성하였다. 각각의 eNodeB는 서로 2,000m의 간격을 두고

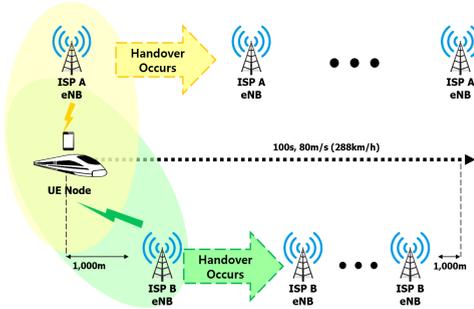


그림 3. 고속철도 환경에서의 LTE 핸드오버 시뮬레이션 실험 환경  
 Fig. 3. LTE handover simulation experiment on high-speed train environment

표 1. 시뮬레이션 파라미터  
 Table 1. Simulation Parameter

Parameter	Value
Bandwidth of ISP A, B	25Mbps
RTT of ISP A, B	5ms
UE speed	80m/s
The distance between each base stations	2,000m
The distance between base station and railway	500m
Experiment time	100s

배치되었으며, UE Node가 지나는 고속철도에서는 500m의 간격을 두고 <그림 3>과 같이 배치되어 있다. 또한, UE Node는 80m/s의 속도로 이동하였다. 각 ISP의 대역폭은 25Mbps, 지연시간은 5ms로 동일하게 설정하였다. 실험은 총 100초 동안 진행하였으며, UE Node는 실험시간 동안 업링크를 통해 대역폭을 가득 채워 데이터를 전송하였다. 해당 환경에서의 단일 TCP 및 MPTCP의 성능을 분석하기 위하여, UE Node를 ISP A의 eNodeB와 연결한 단일 TCP와 ISP A, B의 eNodeB를 함께 연결한 MPTCP의 데이터 처리율을 비교하였다. MPTCP 스케줄러는 minRTT 스케줄러를 사용하였다<sup>22)</sup>.

<그림 4>는 고속철도 환경에서 ISP A만을 사용하여 단일 TCP 환경을 구성하였을 때의 데이터 처리율 그래프이다. 약 12.5초 전후로, UE Node는 ISP A의 eNodeB들과 가장 멀리 떨어진 상태가 된다. 핸드오버가 일어나는 동안 단일 TCP의 데이터 처리율이 순간적으로 5Mbps 이하로 감소하는 것을 그래프에서 확인할 수 있다.

<그림 5>는 고속철도 환경에서 ISP A, ISP B를 함

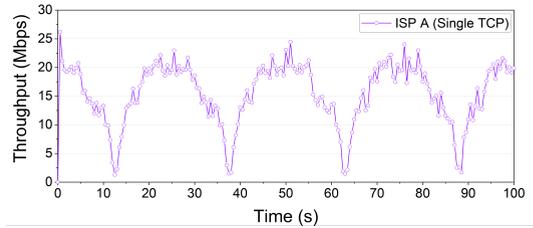


그림 4. 고속철도 환경에서 핸드오버가 발생할 시 단일 TCP의 데이터 처리율  
 Fig. 4. Throughput of single TCP when handover occurs on high-speed train environment

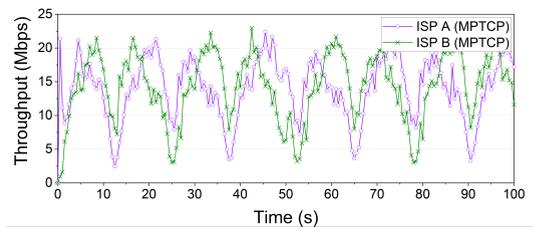


그림 5. 고속철도 환경에서 핸드오버가 발생할 시 MPTCP의 서브플로우 데이터 처리율  
 Fig. 5. Subflow throughput of MPTCP when handover occurs on high-speed train environment

게 사용하는 MPTCP 환경을 구성하였을 때의 데이터 처리율을 나타낸 그래프이다. 핸드오버가 발생하지 않을 때에는 기존의 MPTCP 만큼의 성능을 보인다. 하지만 ISP A를 사용하는 서브플로우에 핸드오버가 발생하면 각 서브플로우의 특성이 서로 크게 차이 나고, 수신 측에서는 패킷을 올바른 순서로 수신하지 못하는 Out-of-Order 패킷 문제가 발생한다. 이는 MPTCP 수신 버퍼의 HoL Blocking 문제로 이어지고, 따라서 MPTCP의 총 데이터 처리율이 감소한다. 결과적으로, MPTCP의 기본 스케줄러가 모든 서브플로우를 함께 사용하려는 특성으로 인해 발생하는 문제이다. 마찬가지로 25초 부근에서, ISP B를 사용하는 서브플로우에 핸드오버가 발생하였으나, ISP A의 데이터 처리율이 ISP B의 데이터 처리율과 함께 감소하면서 MPTCP의 총 데이터 처리율이 감소한다.

고속철도 환경의 경우 각 기지국 간의 거리가 대도시 환경보다 멀리 배치되어 있다. 그 구간을 UE Node가 빠르게 이동한다는 특징을 가지고 있으며, 따라서 정지 혹은 대도시에서 걸어나면서 이동하는 환경보다 핸드오버가 자주 발생하게 된다. 이로 인해 고속철도 환경에서 MPTCP를 구성 및 사용하게 될 경우 각 MPTCP의 총 데이터 처리율이 감소하는 경우가 대도시 환경보다 상대적으로 자주 발생한다.

### III. SB 알고리즘

현재 MPTCP의 기본 스케줄러로는 minRTT 스케줄러가 지정되어 있으나, 해당 스케줄러는 고속철도 환경에서 Out-of-Order 패킷 문제로 인한 HoL Blocking 문제, 총 데이터 처리율의 감소 등을 겪을 수 있다. 이러한 문제점을 해결하기 위하여, 핸드오버가 발생한 서브플로우를 일시적으로 차단함으로써 총 데이터 처리율을 향상시키는 SB (Subflow Blocking) 알고리즘을 제안한다. 해당 알고리즘은 minRTT 스케줄러를 기반으로 짜여졌으나, 핸드오버가 발생한 경우에만 서브플로우를 일시적으로 차단하는 구조이기 때문에 다양한 스케줄러에도 쉽게 적용할 수 있다. 또한, 제안한 알고리즘은 실제 적용될 환경에서는 크로스레이어 기법을 통해 핸드오버를 검출하는 것을 가정하였으며, 진행된 NS-3 시뮬레이터 실험에서는 Handover Message를 통해 알고리즘이 동작한다.

<알고리즘 1>은 제안 알고리즘의 의사 코드이다. 만약 어느 한쪽의 서브플로우에서 핸드오버가 발생하면, 해당 서브플로우의 혼잡 윈도우를 일시적으로 0으로 설정한다. 이때, 해당 서브플로우의 혼잡 윈도우가 0이기 때문에 해당 서브플로우로 패킷의 전송이 이루어지지 않고, 수신 측은 핸드오버로 인한 성능 감소를 겪는 서브플로우로부터 전송되는 패킷을 기다릴 필요가 없다. 따라서, Out-of-Order 패킷 현상과 MPTCP 수신 버퍼에서의 HoL Blocking 문제들을 방지할 수 있다. 또한, 핸드오버가 동시에 발생할 경우 양쪽 서브플로우가 모두 차단되는 것을 방지하기 위해, 다른

알고리즘 1. SB (Subflow Blocking) 알고리즘  
Algorithm 1. Subflow Blocking Algorithm

```

initialization
current_subflow ← NULL

1: for each subflow i do
2:   current_subflow = subflow_i
3:   detect handover
4:   if handover == true &&
      other_subflow_handover == false then
5:     current_subflow→cwnd = 0
6:   else
7:     minRTT_Scheduler()
      // Operate minRTT Scheduler.
8:   end if
9: end for
    
```

서브플로우가 이미 핸드오버가 발생하여 차단된 상태라면 스케줄러는 더 이상 추가로 서브플로우를 차단하지 않는다. 핸드오버가 발생하지 않는다면, 혼잡 윈도우가 0으로 설정되지 않은 서브플로우들 중에서 각 서브플로우들의 지연시간을 비교, minRTT 스케줄러와 동일하게 가장 짧은 지연시간을 가진 서브플로우를 우선적으로 사용한다. 즉, 핸드오버가 일어나지 않을 때에는 사용 가능한 서브플로우를 모두 사용하여 MPTCP의 성능을 보장하고, 핸드오버가 일어날 때에는 해당 서브플로우를 일시적으로 차단함으로써 MPTCP의 성능을 향상시킬 수 있다.

### IV. 시뮬레이션에서의 성능 평가

#### 4.1 Out-of-Order 패킷 비교

<표 2>는 고속철도가 정지한 환경, 고속으로 이동 중인 환경에서 각각 50초 동안 UE Node가 파일을 전송할 때 각각의 Out-of-Order 패킷 수를 측정하였다. 정지 환경에서는 SB 알고리즘이 기존의 minRTT 스케줄러와 유사한 결과를 보이며, 고속으로 이동하는 고속철도 환경에서는 SB 알고리즘의 Out-of-Order 패킷이 약 10% 감소하였다.

표 2. 고속이동 및 정지 환경에서의 Out-of-Order 패킷  
Table 2. Out-of-Order packets in a high-speed train and stationary environment

	MPTCP	MPTCP-SB
Stationary	892	897
High-speed train	2154	1839

#### 4.2 고속철도 환경에서의 성능 평가

<그림 6>은 SB 알고리즘을 사용하였을 때, 각 ISP에 연결한 서브플로우의 데이터 처리율을 측정한 그래프이다. 제안한 알고리즘의 경우 핸드오버가 일어나

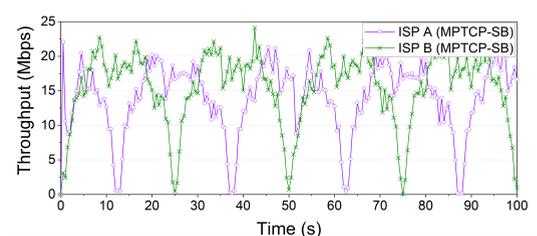
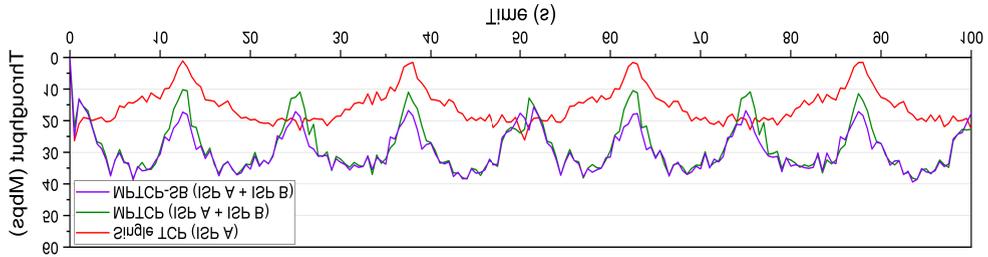
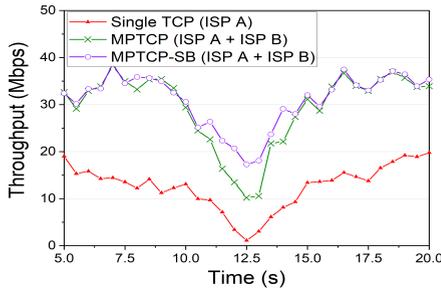


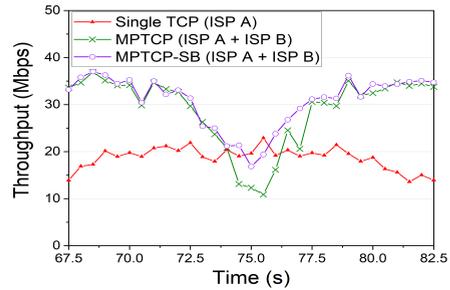
그림 6. 고속철도 환경에서 핸드오버가 발생할 시 SB 알고리즘의 각 서브플로우 데이터 처리율  
Fig. 6. Subflow throughput of SB algorithm when handover occurs on high-speed train environment



(a) 0초에서 100초까지



(b) 5초에서 20초까지



(c) 67.5초에서 82.5초까지

그림 7. 고속철도 환경에서 핸드오버가 발생할 시의 총 데이터 처리율 비교  
Fig. 7. Comparison of total throughput when handover occurs in high-speed train environment

는 서브플로우의 혼잡 윈도우를 0으로 설정하여 데이터를 일시적으로 전송하지 않는다. 이로 인해 핸드오버가 발생한 서브플로우로 전송되는 패킷을 기다릴 필요가 없어지고, 해당 서브플로우의 Out-of-Order 패킷으로 인한 HoL Blocking 현상이 줄어들게 된다. <그림 6>의 75초 지점에서 ISP B를 사용한 서브플로우가 핸드오버가 발생할 때, 해당 서브플로우의 데이터 처리율은 0에 가까워진다. 이는 핸드오버가 발생한 ISP B 서브플로우의 혼잡 윈도우를 일시적으로 0으로 설정하여 서브플로우를 차단하기 때문이다. 기존 MPTCP와는 다르게, 핸드오버가 일어나지 않은 서브플로우는 데이터 처리율의 감소 없이 정상적으로 전송한다. 즉, 핸드오버가 발생하는 시간 동안 핸드오버를 수행하지 않는 서브플로우의 성능 감소 폭을 완화시킨다.

<그림 7>은 ISP A만을 사용하여 구성한 단일 TCP, ISP A와 B 모두 사용하여 MPTCP 환경을 구성한 기존 MPTCP, 그리고 제안한 알고리즘을 적용한 MPTCP의 총 데이터 처리율을 비교한 그래프이다. <그림 7 (a)>의 5~20초, 67.5~82.5초 구간을 확대하여 <그림 7 (b), (c)>에 나타내었다. <그림 7 (b)>는 ISP A에서 핸드오버가 발생한 구간으로, 핸드오버로 인하여 기존 MPTCP의 데이터 처리율이 감소하는 것

을 확인할 수 있다. 하지만, SB 알고리즘은 그 감소 폭을 완화하여 성능을 향상시켜 기존 MPTCP보다 개선된 성능을 보인다. <그림 7 (c)>는 ISP B에서 핸드오버가 발생한 구간으로, 단일 TCP는 ISP A만을 사용하기 때문에 데이터 처리율이 감소하지 않는다. 하지만 기존의 MPTCP는 핸드오버가 일어나면서 Out-of-Order 패킷으로 인한 MPTCP 수신 버퍼에서의 HoL Blocking 문제가 발생하고, 총 데이터 처리율이 감소하여 일시적으로 단일 TCP보다도 성능이 떨어진다. 제안한 알고리즘의 경우 핸드오버가 발생한 서브플로우의 혼잡 윈도우를 0으로 만들기 때문에 Out-of-Order 패킷의 발생이 줄어든다. 이로 인해 기존의 MPTCP보다 더 나은 성능을 보이며, 단일 TCP와 유사한 데이터 처리율을 보인다.

다음은 제안한 알고리즘이 정지한 환경에서 기존 MPTCP와 유사한 성능을 보일 수 있는지 평가하였다. 해당 실험환경에서 UE Node는 50초 동안 이동한 후, 20초 동안 정지하였다. UE Node가 정지한 곳은 기차역을 가정하여 각 ISP의 eNodeB가 인접하여 배치되었다.

<그림 8 (a)>는 ISP A만을 사용하여 구성한 단일 TCP, ISP A와 B 모두 사용하여 MPTCP 환경을 구성한 기존 MPTCP, 그리고 제안한 알고리즘을 적용한

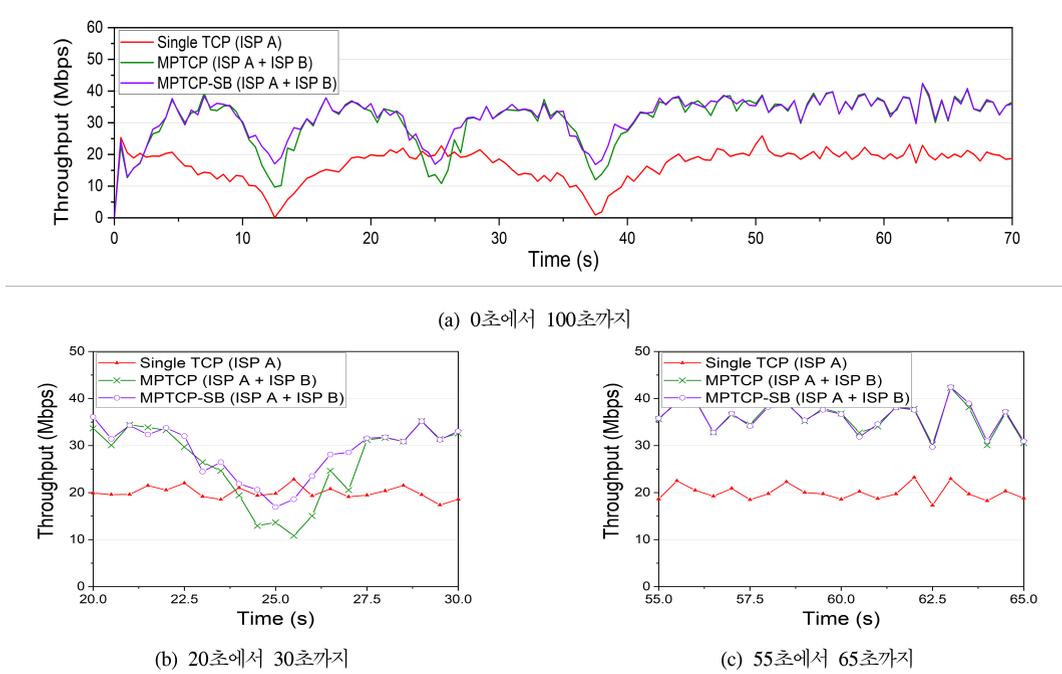


그림 8. 고속철도가 이동 후 정지한 환경에서의 총 데이터 처리율 비교  
 Fig. 8. Comparison of total throughput when high-speed train has been stopped after moving

MPTCP의 총 데이터 처리율을 비교한 그래프이다. <그림 8 (a)>의 5~20초, 55~65초 구간을 확대하여 <그림 8 (b), (c)>에 나타내었다. 먼저 <그림 8 (b)>에서, 이전의 실험과 동일하게 ISP B에서 핸드오버가 발생하였기 때문에 제한한 알고리즘이 기존의 MPTCP보다 더 나은 성능을 보인다. <그림 8 (c)>는 기차역에 기차가 정지된 상태를 가정한 구간이다. 이때, 제한한 알고리즘은 핸드오버가 발생하지 않을 경우 minRTT 스케줄링 알고리즘으로 동작함에 따라 기존의 MPTCP와 동일한 성능을 보인다. 즉, 제한한 알고리즘은 UE Node가 고속으로 이동하는 환경에서는 핸드오버 발생 여부에 따라 서브플로우의 혼잡 원도를 0으로 줄임으로써 MPTCP의 성능을 향상시키고, UE Node가 정지한 환경에서는 minRTT 스케줄러로 동작하기 때문에 성능이 감소하지 않고 기존의 성능과 동일한 성능을 보인다.

<그림 9>는 <그림 3>의 실험 환경에서 UE Node가 이동하면서 100MB, 500MB, 그리고 1GB의 파일을 전송하는데 소모된 시간을 측정하여 나타낸 그래프이다. 100MB 파일의 경우 기존의 MPTCP는 28.4초, 제한한 알고리즘은 27.2초로 약간의 성능 향상을 보였으며, 1GB 파일의 경우 각각 289초와 268초로 10% 정도의 성능 향상을 보임으로써 제한한 알고리즘이

기존의 MPTCP에 비해 성능이 개선되었음을 확인할 수 있다.

<그림 10>은 <그림 3>의 실험 환경에서 ISP B eNodeB를 ISP A eNodeB의 우측에 50m의 간격을 두고 배치함으로써 핸드오버가 함께 발생하는 환경을 구성하였을 때 총 데이터 처리율을 비교한 그래프이다. 제한한 알고리즘은 핸드오버가 동시에 발생할 경우, 양쪽 서브플로우를 모두 차단하는 현상을 방지하기 위하여 한쪽 서브플로우가 이미 차단되어있다면

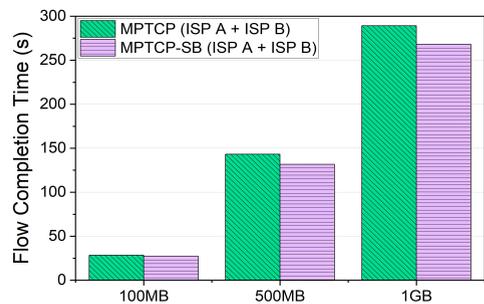


그림 9. 고속철도 환경에서 핸드오버가 발생할 시 전송 완료 시간 비교  
 Fig. 9. Comparison of flow completion time when handover occurs in high-speed train environment

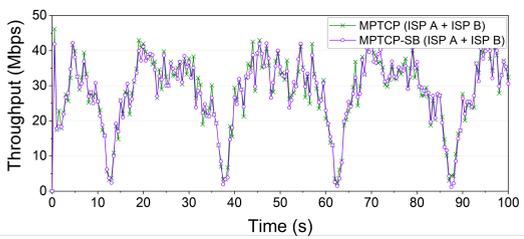


그림 10. 고속철도 환경에서 핸드오버가 동시에 발생할 시 총 데이터 처리율 비교  
 Fig. 10. Comparison of total throughput when handover occurs simultaneously on high-speed train environment

다른 서브플로우는 차단하지 않는다. 이로 인해, 핸드 오버가 함께 발생하는 환경에서도 기존의 MPTCP와 유사한 성능을 보이는 것을 확인할 수 있다.

### V. 결론

본 논문에서는 LTE 신호 세기의 감소와 핸드오버가 자주 발생하는 고속철도 환경에서는 일시적으로 MPTCP의 총 데이터 처리율이 단일 TCP의 데이터 처리율보다도 성능이 떨어지기도 함을 NS-3 시뮬레이션 실험을 통해 확인하였다.

이러한 문제점을 해결하기 위해, 본 논문에서는 핸드오버에 따라 서브플로우를 일시적으로 차단함으로써 MPTCP의 성능을 향상시키는 알고리즘을 제안하였다. 제안한 알고리즘은 서브플로우에서 핸드오버가 발생하면 일시적으로 해당 서브플로우를 차단함으로써 Out-of-Order 패킷의 발생을 감소시키며, 이로 인해 핸드오버가 일어나지 않은 서브플로우를 기존의 MPTCP보다 효율적으로 사용할 수 있다. 제안한 알고리즘은 핸드오버가 자주 발생하는 고속철도 환경에서 기존의 MPTCP보다 Out-of-Order 패킷이 10% 정도 감소하였으며, 데이터 처리량이 최대 10% 상승하였음을 확인하였다. 추후에는 제안 알고리즘을 업링크뿐만 아니라 다운링크에서도 적용할 수 있도록 개선할 예정이다.

### References

[1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," *RFC 6824*, IETF, 2013.  
 [2] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP extensions

for multipath operation with multiple addresses," *RFC 8684*, IETF, 2020.

[3] J. Postel, "Transmission control protocol," *RFC 793*, IETF, 1981.  
 [4] L. Li, K. Xu, T. Li, K. Zheng, C. Peng, D. Wang, X. Wang, M. Shen, and R. Mijumbi, "A measurement study on multi-path TCP with multiple cellular carriers on high speed rails," in *Proc. ACM Sigcomm*, pp. 161-175, Budapest, Hungary, Aug. 2018.  
 [5] N. Mohan, T. Shreedhar, A. Zavodovski, J. Kangasharju, and S. Kaul, "Is two greater than one?: Analyzing multipath TCP over Dual-LTE in the Wild," *arXiv:1909.02601*, Sep. 2019.  
 [6] P. Anilal, B. V. Sainandan, S. S. S. Sai, and P. Yellai, "Experimentation and analysis of Multipath TCP," in *IEEE Int. Conf. Advanced Netw. and Telecommun. Syst.*, pp. 1-3, Kolkata, India, Dec. 2015.  
 [7] S. C. Nguyen, X. Zhang, T. M. T. Nguyen, and G. Pujolle, "Evaluation of throughput optimization and load sharing of multipath TCP in heterogeneous networks," in *Int. Conf. Wireless and Optical Commun. Netw.*, pp. 1-5, Paris, France, May 2011.  
 [8] V. Adarsh, P. Schmitt, and E. Belding, "MPTCP Performance over Heterogenous Subpaths," in *Int. Conf. Computer Commun. and Netw.*, pp. 1-9, Valencia, Spain, Jul. 2019.  
 [9] J. Hashimoto and T. Shigeyasu, "Performance evaluation of MPTCP under heterogeneous channel characteristics," in *Int. Wkshps. Computational Intell. and Appl.*, pp. 137-141, Hiroshima, Japan, Apr. 2015.  
 [10] J. Oueis and E. C. Strinati, "Uplink traffic in future mobile networks: Pulling the alarm," in *Int. Conf. Cognitive Radio Oriented Wireless Netw.*, pp. 583-593, Grenoble, France, Nov. 2015.  
 [11] Y. Yu and C. Yin, "A cross-layer mechanism for TCP connection over wireless uplink in cellular networks," in *Veh. Technol. Conf.*, vol. 4, pp. 2250-2254, Stockholm, Sweden, May 2005.

[12] O. Bonaventure, M. Handley, and C. Raiciu, "An overview of multipath TCP," *USENIX Netw. Syst.*, vol. 37, no. 5, pp. 17-23, Oct. 2012.

[13] C. Raiciu, M. Handly, and D. Wischik, "Coupled congestion control for multipath transport protocols," RFC 6356, IETF, 2011.

[14] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? Designing and implementing a deployable multipath TCP," in *USENIX Symp. NSDI'12*, pp. 29, San Jose, USA, Apr. 2012.

[15] G. Kim, C. Park, J. Kim, W. Eom, Y. Song, W. Seo, and Y. Cho, "MPTCP scheduler research trends and problem analysis," *J. KICS*, vol. 45, no. 9, pp. 1570-1586, Sep. 2020.

[16] A. Alheid, D. Kaleshi, and A. Doufexi, "An analysis of the impact of Out-of-Order recovery algorithms on MPTCP throughput," in *Int. Conf. Advanced Inf. Netw. and Appl.*, pp. 156-163, Victoria, Canada, May 2014.

[17] M. Scharf and S. Kiesel, "Head-of-line blocking in TCP and SCTP: Analysis and measurements," in *IEEE GLOBECOM*, pp. 1-5, San Francisco, USA, Nov. 2006.

[18] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," in *IFIP Netw. Conf.*, pp. 431-439, Vienna, Austria, May 2016.

[19] Y. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *Int. Conf. Emerging Netw. EXperiments and Technol.*, pp. 147-159, Incheon, Republic of Korea, Dec. 2017.

[20] F. Yang, Q. Wang, and P. D. Amer "Out-of-Order transmission for in-order arrival scheduling for multipath TCP," in *Int. Conf. Advanced Inf. Netw. and Appl.*, pp. 749-752, Victoria, Canada, May 2014.

[21] *NS-3 Network Simulator*, Retrieved Oct. 27, 2021, from <https://www.nsnam.org>

[22] *MultiPath - Linux Kernel implementation*, Retrieved Oct. 27, 2021, from <https://www.multipath-tcp.org>

박 창 훈 (Chang-hoon Park)



2019년 2월: 경북대학교 전자공학부 졸업  
 2021년 8월: 경북대학교 전자전기공학부 석사  
 <관심분야> TCP, MPTCP, LTE

[ORCID:0000-0002-5066-2980]

김 건 환 (Geon-hwan Kim)



2015년 2월: 경북대학교 전자공학부 졸업  
 2017년 2월: 경북대학교 전자공학부 석사  
 2017년 3월~현재: 경북대학교 전자전기공학부 박사과정

<관심분야> 차세대 이동 네트워크, 드론 ICT 융합 기술, TCP 혼잡제어, MPTCP 혼잡제어  
 [ORCID:0000-0003-2739-8939]

김 정 근 (Jeong-keun Kim)



2020년 2월 : 대구대학교 전자  
제어공학전공 졸업  
2020년 3월~현재 : 경북대학교  
전자전기공학부 석사과정  
<관심분야> QUIC, MPTCP,  
차량 애드혹 네트워크

[ORCID:0000-0002-5577-6004]

조 효 섭 (Hyo-sup Cho)



2021년 2월 : 대구대학교 전자  
전기공학부 졸업  
2021년 3월~현재 : 경북대학교  
전자전기공학부 석사과정  
<관심분야> TCP, MPTCP,  
AQM  
[ORCID:0000-0001-6201-7551]

엄 원 주 (Won-ju Eom)



2020년 2월 : 경북대학교 전자  
공학부 졸업  
2020년 3월~현재 : 경북대학교  
전자전기공학부 석사과정  
<관심분야> SDN, 차세대 네트  
워크, TCP 혼잡제어

[ORCID:0000-0003-4322-0882]

조 유 제 (You-ze Cho)



1982년 : 서울대학교 전자공학  
과 졸업  
1983년 : 한국과학기술원 전기  
전자공학 석사  
1988년 : 한국과학기술원 전기  
전자공학 박사  
1989년~현재 : 경북대학교 전자  
공학부 교수

1992년~1994년 : Univ. of Toronto in Canada, 객원  
교수

2002년~2003년 : NIST(미국국립표준연구소) 객원연  
구원

<관심분야> 차세대 이동 네트워크, 무선 애드혹 네  
트워크, 이동성 관리 기술, 차세대 전송 계층 프  
로토콜

[ORCID:0000-0001-9427-4229]

서 상 진 (Sang-jin Seo)



2021년 2월 : 경북대학교 전자  
공학부 졸업  
2021년 3월~현재 : 경북대학교  
전자전기공학부 석사과정  
<관심분야> TCP 혼잡제어, 강  
화학습, 무선 전송 계층 프  
로토콜

[ORCID:0000-0002-5464-8131]