

고속열차에서 다중 셀룰러 망을 이용한 MPTCP를 위한 선제적 핸드오버 인지 패킷 스케줄링

김민기*, 김응협*, 강순주**, 조유제^o

Proactive Handover-Aware Packet Scheduling for MPTCP with Multiple Cellular Networks on High-Speed Trains

Min-Ki Kim*, Eung-Hyup Kim*, Soon Ju Kang**, You-Ze Cho^o

요약

고속열차 환경에서는 높은 이동성으로 인한 빈번한 셀 간 핸드오버에 의해 LTE (Long Term Evolution)의 성능이 크게 감소한다. 다수의 LTE를 통합한 MPTCP (Multi Path Transmission Control Protocol)는 빈번한 핸드오버로 인한 문제를 해결하기 위한 방안으로 주목받고 있다. 그러나 기존의 다중경로 스케줄러를 고속 열차에서 사용할 경우 핸드오버 발생 시 네트워크의 급격한 변화로 인해 정상적인 스케줄링을 수행하지 못하고 Out-of-Order 패킷이 발생한다. 고속 열차의 이동 경로는 선로를 따르므로 변하지 않으며 일정한 속도로 이동한다. 또한, 기지국의 위치도 고정되어 있다. 따라서 사전에 수집된 핸드오버 정보는 이후의 열차 운행에서 사용할 수 있다. 이 논문에서는 사전에 수집된 핸드오버 정보와 열차의 실시간 위치 정보를 이용하여 핸드오버 발생 시 해당 서브플로우를 차단하는 선제적 핸드오버 인지 스케줄러를 제안한다. 제안된 기법은 Network Simulator-3를 이용해 평가되었으며, 핸드오버 발생 시 순간 처리량을 개선할 수 있음을 확인하였다.

Key Words : MPTCP, Packet Scheduler, High-Speed Train, Handover

ABSTRACT

In the high-speed train environment, frequent handovers between cells due to high mobility significantly degrade the performance of LTE (Long Term Evolution). The integration of multiple LTE connections through MPTCP (Multi Path Transmission Control Protocol) has gained attention as a solution to address the challenges posed by frequent handovers. However, when using conventional multipath schedulers in high-speed trains, the network experiences abrupt changes during handovers, leading to the inability to perform normal scheduling and resulting in Out-of-Order packet occurrences. The movement path of high-speed trains follows railway tracks, remaining unchanged and moving at a constant speed. Additionally, the base station's location is fixed. Therefore, handover information collected in advance can be utilized for subsequent train operations. In this

※ This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by Ministry of Education (No. NRF2018R1A6A1A03025109) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C1003928).

• First Author : School of Electronic and Electrical Engineering, Kyungpook National University, gms03158@knu.ac.kr, 학생회원

^o Corresponding Author : School of Electronic Engineering, Kyungpook National University, yzcho@ee.knu.ac.kr, 종신회원

* School of Electronic Engineering, Kyungpook National University, ehkim@ee.knu.ac.kr, 학생회원

** Department of Electronics, Kyungpook National University, sjkang@ee.knu.ac.kr, 종신회원

논문번호 : 202401-021-C-RE, Received January 26, 2024; Revised March 5, 2024; Accepted March 25, 2024

paper, we proposed the proactive handover-aware scheduler, utilizing pre-collected handover information and real-time location data of the train to block the corresponding subflows when handovers occur. The proposed scheduler is evaluated using Network Simulator-3, confirming its capability to improve throughput at the moment of handover occurrences.

1. 서 론

전 세계적으로 고속 철도의 이용률은 꾸준한 증가를 보이고 있다. 따라서 기차 내에서 제공되는 무선 인터넷 서비스의 품질 개선이 요구되고 있다. 또한 온라인 게임 및 비디오 스트리밍과 같은 실시간 콘텐츠 소비가 계속 증가함에 따라 저지연, 연결 지속성 및 고속 통신에 대한 요구도 높아지고 있다. 이러한 요소들은 차세대 전송 매체의 성능을 평가할 때 중요한 고려사항이다^[1].

최근에는 고속 열차의 객실 내에 무선 인터넷 서비스를 제공하기 위해 두 개의 서로 다른 사업자의 셀룰러 네트워크를 활용하고 있다. 그러나 고속 이동 환경에서는 정지 환경과 비교하여 도플러 효과, 다중 경로 페이딩, 거리에 따른 신호 감쇠, 음영 지역 통과 및 핸드오버와 같은 문제로 인해 LTE(Long Term Evolution)의 성능이 크게 저하된다^[2].

현재 이러한 문제를 해결하기 위해 고속 열차에 MPTCP (MultiPath Transmission Control Protocol)를 적용하는 연구가 진행 중이다^[3]. MPTCP는 하나의 사용자 장치가 둘 이상의 인터페이스와 다수의 IP 주소를 가지고 있는 경우에 단일 연결 내에서 여러 TCP 경로를 사용할 수 있게 하는 TCP의 확장 버전이다. MPTCP에서는 하위 TCP 흐름을 서브플로우(Subflow)라고 한다.

MPTCP에서 여러 서브플로우를 활용하기 때문에 단일 TCP 플로우의 상위 계층에는 각 패킷을 연결된 서브플로우에 순서대로 분배하는 다중 경로 패킷 스케줄러가 필수적이다. 현재 MPTCP의 기본 스케줄러는 minRTT (Minimum Round-Trip Time)^[4]를 사용하고 있으며, minRTT의 단점을 개선하기 위해 여러 스케줄러가 개발되었다^{5-10]}.

그러나 대부분의 Linux용 MPTCP 스케줄러는 정적인 시나리오에서 평가되었으며, 배포 단계에서 극단적인 이동성을 고려하지 않고 설계되었다^[11]. 따라서 기존의 스케줄러를 고속 열차에 적용할 경우 OFO (Out-of-Order) 문제가 발생할 수 있다^[12]. MPTCP의 안정적인 데이터 전송을 보장하는 특성으로 인해 OFO 패킷은 HoLB (Head-of-Line Blocking)을 일으켜 프로토콜의 성능을 크게 저하시킨다. 실제 환경에서의 평가는 MPTCP의 서브플로우 중 하나에서 핸드오버가 발

생할 때, 핸드오버 없이 단일 TCP 연결을 사용하는 것보다 MPTCP의 처리량이 낮다는 것을 보여준다^[13].

고속이동 시나리오나 급격한 네트워크 조건 변화에 대응하여 패킷을 적응적으로 분배하기 위한 여러 스케줄러가 제안되었다. 이들의 접근 방식에는 OFO 대기열의 크기를 측정하는 것^[14], 패킷 손실률을 기반으로 임계값을 설정하여 임계값을 초과하는 손실률이 관측될 때 반응하는 것^[15], 또는 과거 관측과 보상에 기반하여 패킷을 예약하는 데 머신 러닝을 도입하는 것이 포함된다^[16,17]. 그러나 이미 발생한 손실과 OFO 패킷을 감지하거나 과거 패킷에서 학습하기 때문에 네트워크 장애에 대한 예측과 선제적인 대응에서는 그 정확도가 떨어질 수 있다.

우리는 고속열차에서 MPTCP의 문제를 개선하기 위해 선제적 핸드오버 인식 스케줄러를 제안하고, MPTCP의 기본 스케줄러에 적용하여 핸드오버로 인한 문제를 개선할 수 있음을 보였다^[18]. 본 논문에서는 제안 방안을 여러 스케줄러에 적용하여 성능을 분석하고, 제안 방안을 실제 환경에 적용할 새로운 방안을 제안한다. 본 논문은 다음의 세 가지 주요 기여를 가진다.

제안된 패킷 스케줄러를 구현하고 평가함으로써 고속 열차 운행 중 미리 수집한 핸드오버 정보를 활용하여 핸드오버 발생 중 순간적인 처리량 저하 문제를 완화할 수 있음을 검증했다.

기존에 배포된 스케줄러 중 고속열차에서 사용하기 가장 적합한 스케줄러가 무엇인지 확인했다.

마지막으로, 제안된 스케줄러를 실제 고속 열차에 적용하는 현실적인 방안을 제안한다.

논문의 나머지 부분은 다음과 같이 구성되어 있다. II 장에는 본 연구를 뒷받침하기 위한 관련 연구 및 연구 배경을 소개한다. III 장에서는 본 연구에서 제안하는 스케줄러의 구조와 동작을 설명한다. IV 장에서는 제안한 스케줄러의 성능평가 실험 결과를 제시하고 분석한다. 마지막으로, V 장에서 논문의 결과를 요약 및 결론을 도출하고 향후 연구에 대해 간략히 설명한다.

II. 관련 연구 및 연구 배경

2.1 관련 연구

N. Williams 등은 이동 환경에서 MPTCP의 잠재적 이점을 강조하고 있다¹⁹⁾. MPTCP는 여러 경로를 통한 전송을 수행함으로써 네트워크 대역폭을 더 효과적으로 활용할 수 있으며, 안정된 전송을 위해 서로 다른 서브플로우 간을 전환할 수 있다.

S. Rene 등은 Deep Packet Inspection (DPI)을 활용하여 이동 네트워크에서 MPTCP의 성능을 개선하는데 중점을 두고 있다²⁰⁾. 이를 통해 핸드오버 중에 원활한 통신을 지원하고 품질 경험을 적응적으로 향상시키며 이중 네트워크에서 발생하는 지연 차이로 인한 문제에 대응한다.

D. Zhu 등은 데이터가 송신자에서 수신자로 전송되는 데 걸리는 시간과 ACK (Acknowledgment)가 수신자에서 송신자로 도착하는 데 걸리는 시간을 측정한다²¹⁾. 이러한 시간을 비교함으로써 이동성을 감지하고 노드 간 거리를 측정한다. 측정된 거리를 사용하여 핸드오버를 감지하고 더 안정적인 경로를 우선시하는 핸드오버 메커니즘을 제안한다. 그러나 ACK가 도착하는 데 걸리는 시간은 노드 간 거리에만 의존하는 것이 아니라 다양한 요인에 영향을 받을 수 있다.

I. Khan 등은 핸드오버로 인한 MPTCP의 성능 저하, 이동성으로 인한 손실 및 지연 증가를 완화시키는 데 중점을 두고 있다¹⁴⁾. OFO 대기열이 특정 임계값을 초과할 때 기본 minRTT 스케줄러와 중복 스케줄러의 처리량을 비교한다. 중복 스케줄러가 높은 처리량을 달성한다고 예상되면 중복 스케줄링을 채택한다. 이 접근 방식은 잦은 핸드오버와 높은 손실률이 있는 상황에서 패킷 손실을 줄이고 처리량을 향상시킨다. 그러나 이 방법은 OFO 대기열이 쌓인 후에 스케줄링 방식을 전환하므로 반응속도가 느릴 수 있다.

C-H. Park 등은 고속 열차 환경에서 핸드오버로 인한 OFO 문제에 대응한다²²⁾. 두 사업자의 LTE를 활용하며, 하나의 LTE에서 핸드오버가 감지될 때 핸드오버가 발생한 경로의 혼잡 창을 즉시 0으로 설정하여 해당 서브플로우를 차단한다. 이 방식은 고속 열차에서 파일 다운로드 시간을 10% 줄였다. 그러나 이 방법을 사용할 경우 차단된 서브플로우가 다시 연결될 때 혼잡 창을 처음부터 채워야 해서 링크 복구가 느려질 수 있다.

2.2 MPTCP

MPTCP는 2013년에 표준화된 TCP 프로토콜의 다중경로 확장 프로토콜이다. MPTCP는 중단 호스트와

TCP 서버 간에 여러 TCP 경로를 설정하고, 이러한 다중 경로를 통해 동시에 데이터를 송수신할 수 있게 한다. MPTCP가 사용하는 각 TCP는 서로 다른 주소를 가지며, MPTCP의 패킷 스케줄러는 각 서브플로우에 패킷을 분배한다. MPTCP는 운영 체제의 수정 없이 TCP 기반 응용 프로그램에서 사용할 수 있으며, 응용 프로그램에게 일반 TCP와 동일한 서비스를 제공한다.

MPTCP 운영 모드: MPTCP는 다중 경로를 활용함에 있어 두 가지 다른 모드를 가진다. 첫 번째는 MPTCP full 모드이다. 이 모드에서 MPTCP는 가능한 모든 경로를 동시에 사용하여 데이터를 전송한다. Full 모드는 이론적으로 여러 경로의 대역폭을 병합하여 더 빠른 데이터 전송을 가능하게 하며, 연결의 안정성을 높인다. 그러나 서로 다른 특성을 가질 경우 효율적으로 트래픽을 분배하기 어려워진다. 두 번째는 MPTCP 예비 모드다. 예비 모드에서 여러 경로는 주 경로와 예비 경로로 분류된다. 데이터는 주로 주 경로를 통해 전송되며, 주 경로가 사용 불가능해지면 예비 경로로 데이터를 전송한다. 이 모드는 대역폭을 희생하는 대신 안정적이고 지속적인 데이터 전송을 가능하게 한다.

2.3 고속 열차의 네트워크 특성

모바일 시나리오에서 전송 프로토콜 및 LTE 성능 저하의 주요 요인 중 하나는 이동성 예측의 어려움이다. 특히 차량 주행 환경에서는 여러 장애물의 간섭으로 인해 이동 경로의 예측이 어려워진다. 반면 고속 열차의 경우, 경로가 미리 정의되어 있고 각 열차가 고정된 선로를 따른다. 또한, 열차의 최대 속도 및 구간별 속도는 운행마다 일정하다. 기지국 또한 고정되어 있어 채널의 물리적 특성이 운행 시간 및 경로 전방에서 어떻게 변할지 예측할 수 있다. 관련 연구에 따르면 고속 열차에서의 RSRP (Reference Signal Received Power)는 동일한 위치에서 측정되었을 때 시간이 다르더라도 유사한 크기와 변동성을 보인다²⁾. 또한, 음역 지역을 통과하는 데 걸리는 시간도 거의 일정하다. 이러한 특성은 네트워크 변동성을 사전에 예측할 수 있도록 한다. 예측에는 시간을 활용한 방법과 위치 정보를 활용한 방법이 있다. 고속열차는 때때로 수 분 가량 지연되기도 하므로 시간을 사용한 방법은 정확도가 떨어질 수 있다. 때문에 고속열차 환경에서는 위치정보를 기반으로 핸드오버를 예측하는 것이 바람직하다.

고속 열차에서의 MPTCP는 두 LTE가 동시에 핸드오버 되지 않는다는 특징을 가진다^{13,23)}. 이는 각 LTE 인터페이스는 하나의 사업자를 통해서만 서버에 데이터를 전송할 수 있으며, 하나의 인터페이스 내에서 사업

자 간의 핸드오버가 발생하지 않기 때문이다. 또한 각 LTE 사업자는 독립적으로 기지국을 소유하며, 각 기지국들은 서로 다른 위치에 설치되어 있기 때문이다. 이러한 특성은 고속 열차에서 MPTCP를 사용할 때 핸드오버 발생 시에도 정상 서브플로우를 통해 데이터를 계속 전송할 수 있게 한다. 따라서 채널 변동 정보의 사전 수집 및 이상적인 패킷 스케줄링을 통해 핸드오버 중에 적어도 하나의 LTE와 동등한 처리량을 보장할 수 있다.

III. 선제적 핸드오버 인지 스케줄러

본 장에서는 고속 열차의 경로 특성을 활용하는 선제적 핸드오버 인식(Proactive Handover-Aware, PHA) 스케줄러를 제안한다.

그림 1은 고속 열차에 MPTCP를 적용한 시나리오에서의 네트워크 구성을 보여준다. 일반적으로 휴대폰에는 LTE 인터페이스가 1개 밖에 없으므로 고속열차의 승객은 MPTCP를 사용하기 위해 고속열차의 백홀 네트워크의 도움을 받아야 한다. 이를 위해 승객은 우선 객실 내에 제공되는 Wi-Fi를 이용하여 백홀 네트워크에 데이터를 전송한다. 이때 승객은 regular TCP를 사용한다. 백홀 네트워크에는 MPTCP를 사용하여 외부와 통신이 가능한 MPTCP 라우터가 있다. 라우터에는 고속열차에서 사용되는 각 사업자의 LTE망에 해당하는 LTE 인터페이스가 각각 설치되어 있으며, 각 인터페이스는 독립적인 IP 주소와 MAC 주소를 가진다. Wi-Fi를 통해 백홀 네트워크로 전달된 TCP 패킷이 MPTCP 패킷으로 전환되기 위해 백홀 네트워크 내에 있는 MPTCP 라우터는 TCP 패킷을 MPTCP 패킷으로 변환하는 프록시 역할을 수행해야 한다. 그러나 MPTCP 프록시를 구현하는 것은 본 연구의 범위를 벗어나므로 본 논문에서는 MPTCP 라우터가 MPTCP 프

록시 기능을 포함한다고 가정하고 네트워크를 구성한다. MPTCP 라우터는 Wi-Fi를 통해 TCP 패킷을 수신한 후 이를 MPTCP 패킷으로 변환하고 라우터에 설치된 두 LTE 인터페이스를 통해 외부 서버로 전송한다. 이때 서버는 수정되지 않은 native MPTCP를 사용하며, 두 개 이상의 독립적인 네트워크 인터페이스를 가진다. 반대로, MPTCP 라우터가 서버로부터 수신한 MPTCP 패킷은 TCP 패킷으로 변환되고 Wi-Fi와 regular TCP를 통해 사용자 장치로 전달된다. 본 논문에서 제안하는 스케줄러를 사용하기 위해 MPTCP 라우터는 열차의 GPS(Global Positioning System) 장치로부터 열차의 실시간 위치 정보를 얻으며, 미리 저장된 핸드오버 정보를 불러와 가지고 있게 된다.

PHA 스케줄러는 사전에 수집된 핸드오버 정보와 열차의 위치 정보를 이용한다. 열차의 실시간 위치 및 핸드오버가 발생하는 위치는 GPS를 이용하여 수집한다. PHA 스케줄러의 동작은 II 장에서 언급한 고속 열차의 두 특징을 활용하여 다음과 같이 설명할 수 있다. 핸드오버가 발생하는 위치와 해당 ISP (Internet service provider)에 대한 정보를 선제적으로 수집하고, 이 정보를 운행 중에 수집한 열차의 실시간 위치 데이터와 비교하여 핸드오버가 언제 발생할지 예측한다. 핸드오버가 예측되면 해당 서브플로우를 미리 차단하고 나머지 서브플로우로 패킷을 전송한다. 이렇게 함으로써 잘못된 스케줄링으로 인한 OFO 패킷의 수를 줄이고, 적어도 단일 TCP 연결과 동등한 성능을 보장할 수 있다.

알고리즘 1은 PHA 스케줄러의 의사 코드를 나타낸

알고리즘 1. 선제적 핸드오버 인지 스케줄러의 알고리즘
Algorithm 1. Algorithm of proactive handover-aware scheduler.

Algorithm 1 Proactive Handover-Aware Scheduler

Input:

- 1) *HO_table*: Handover table
- 2) *location*: Real-time location of train

```

1: while true do
2:   available_sf = getAvailableSubflow();
3:   for do
4:     handover_Occurrence =
       predict(HO_table, location);
5:   if handover_Occurrence == true then
6:     return available_sf - i
7:   else then
8:     return operateExistingScheduler();
       // minRTT, BLEST, ECF, Peekaboo
9:   end if
10: end for
11: end while
    
```

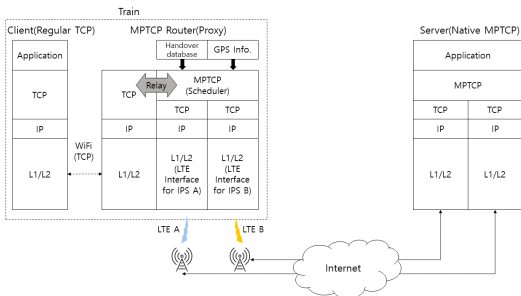


그림 1. 고속 열차에서 MPTCP를 사용하는 시나리오의 네트워크 구성
Fig. 1. Network architecture of scenarios using MPTCP on high-speed trains.

다. 먼저, 미리 수집된 핸드오버 테이블과 열차의 실시간 위치 정보를 입력으로 받는다. 핸드오버 테이블은 각 사업자의 LTE에서 핸드오버가 발생하는 위치를 저장하고 있다. 운행이 시작되면 스케줄러는 사용 가능한 서브플로우를 검색한다. 이 경우 서브플로우는 각 사업자의 LTE를 나타낸다. 그런 다음 각 서브플로우에 대해 핸드오버 테이블과 열차의 실시간 위치 정보를 비교하여 언제 핸드오버가 발생할지 예측한다. 열차가 핸드오버가 발생하지 않는 위치에 있는 경우, 기존 스케줄러를 사용하여 동작한다. 열차가 핸드오버가 발생하는 위치에 있는 경우, 핸드오버 프로세스가 완료될 때까지 정상 서브플로우를 통해서만 패킷을 전송한다.

그림 2는 열차의 실시간 위치와 미리 수집된 핸드오버 테이블의 예시를 나타낸다. GPS 오차를 고려하여 핸드오버 발생 위치를 범위로 표시한다. 이 영역의 크기는 핸드오버 기법과 기지국의 사양을 기반으로 결정할 수 있다. 핸드오버 테이블의 데이터는 물리적 환경의 변화에 적응하고 예측의 정확성을 향상시키기 위해 각 운행마다 누적된다. 열차의 실시간 위치는 핸드오버에 빠르게 대응하기 위해 매 10 ms마다 측정된다. 그림 2에 나타난 예시에서는 핸드오버가 (x1, y1) 및 (x2, y2) 사이에서 발생한다. 핸드오버 테이블은 사전 수집된 핸드오버가 발생하는 모든 영역을 나열하며, 이 중에는 (x1, y1) 및 (x2, y2)도 포함되어 있다. 또한 각 영역에 대해 해당하는 LTE 사업자를 구분한다. PHA 스케줄러는 열차 위치가 업데이트될 때마다 핸드오버 테이블을 확인한다. 열차가 핸드오버 테이블에 나타난 영역에 진입하면 서브플로우 차단 절차를 시작한다.

PHA 스케줄러를 다운링크에 적용하기 위해서는 MPTCP 서버가 열차에서 수집한 정보를 알아야 한다. 이를 달성하기 위해서는 서버의 운영 체제를 수정하고 연결된 호스트가 고속 열차인지 아닌지를 구별해야 한다. 이는 비용과 시간이 많이 소요되는 문제이며, 정보

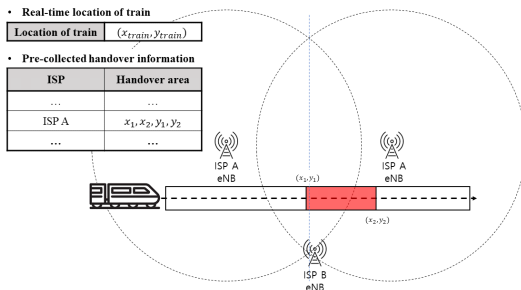


그림 2. 열차의 실시간 위치와 핸드오버 테이블의 예시
Fig. 2. An example of real-time trains location and handover table.

의 실시간 교환으로 인해 추가적인 오버헤드가 발생할 수 있다. 이러한 문제를 해결하기 위해 열차 측에서 모든 핸드오버를 예측하고 반응할 수 있어야 한다.

II 장에서 언급한 대로, MPTCP에는 여러 서브플로우를 동시에 연결하는 두 가지 방법이 있다. 호스트는 MPTCP 헤더의 subtype 필드를 사용하여 연결이 이미 설정된 후에도 연결 모드를 변경할 수 있다. 그림 3은 MPTCP의 헤더의 옵션부를 나타낸다. 수신자가 subtype 필드를 MP_PRIO로 설정하고 B 플래그를 0으로 설정된 패킷을 송신자에게 보내면 해당 서브플로우는 주 경로가 된다. B 플래그가 1로 설정되면 서브플로우는 예비 경로가 되며, 주 경로가 하나라도 연결되어 있다면 예비 경로로 데이터를 전송하지 않는다. MP_PRIO를 통한 서브플로우 우선순위 변경은 수신자에 의해서만 요청될 수 있으며, 수신자는 예비 경로를 통해 패킷을 계속 전송할 수 있다. 따라서 수신자는 필요에 따라 예비 경로를 주 경로로 자유롭게 전환할 수 있다.

우리는 이러한 MPTCP 특성을 활용하여 핸드오버가 예상되는 서브플로우를 미리 예비 경로로 전환하여 핸드오버로 인한 중단을 방지하는 다운링크용 선제적 핸드오버 인지 스케줄러 (PHA-Downlink, PHA-D)를 제안한다. PHA-D는 서버 측 운영 체제를 변경하지 않고 동작하며 비용 효율적이고 쉽게 배포할 수 있다.

그림 4는 고속열차 환경에서 핸드오버가 발생할 때 PHA-D의 메시지 흐름을 나타낸다. 데이터 전송이 가능한 서브플로우는 실선으로 나타내고, 예비 경로는 점선으로 표시한다. 기본적으로 서버는 모든 서브플로우

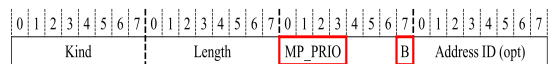


그림 3. 서브플로우 우선순위 변경 옵션 (MP_PRIO)
Fig. 3. Change subflow priority option (MP_PRIO).

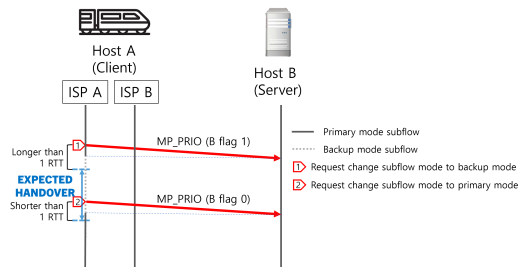


그림 4. 핸드오버가 예측될 때 PHA-D의 MP_PRIO 메시지 흐름
Fig. 4. MP_PRIO message flow of PHA-D when a handover is expected.

를 사용하는 Full 모드로 운영된다. 핸드오버가 예상되면 열차는 B 플래그가 1로 설정된 MP_PRIO 패킷을 전송하여 해당 서브플로우를 예비 경로로 변경하도록 서버에 요청한다. MP_PRIO 이전에 전송된 패킷이 문제없이 수신되기 위해 예상되는 핸드오버 시작 시간보다 적어도 1 Round-Trip Time (RTT) 전에 MP_PRIO 패킷을 보내야 한다. 핸드오버가 완료될 것으로 예상되면 열차는 B 플래그가 0으로 설정된 MP_PRIO 패킷을 전송하여 해당 서브플로우를 주 경로로 변경하도록 요청한다. 이 경우 주 경로로의 전환과 동시에 정상적으로 패킷을 수신하기 위해서 예상되는 핸드오버 완료 시간보다 최대 1 RTT 전에 MP_PRIO 패킷을 보내야 한다.

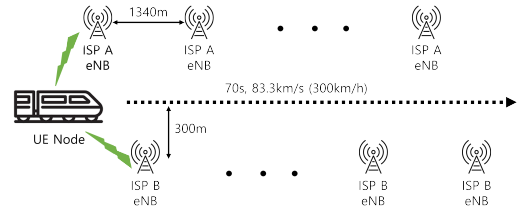


그림 5. 고속열차에서 다중 셀룰러 네트워크를 이용한 MPTCP 시나리오
Fig. 5. MPTCP scenarios using multiple cellular networks on high-speed train.

IV. 성능평가

4.1 실험 환경 구성

제안한 스케줄러를 평가하기 위해 MPTCP 용으로 확장된 NS-3 (Network Simulator 3)^[24]을 사용했다. 성능 지표는 처리량을 사용하였다. 고속열차에서 MPTCP의 성능을 자세하고 광범위하게 평가한 L. Li 등의 연구에 따르면 시속 300km로 달리는 고속열차에서 하나의 LTE는 분당 평균 3회의 핸드오버를 경험하며, 두 개의 LTE로 이루어진 MPTCP를 사용할 경우 분당 평균 6회의 핸드오버를 경험한다^[13, 23]. 따라서 본 실험에서도 분당 6회의 핸드오버를 시뮬레이션 하며, 실험은 총 70초 동안 진행하였다. 그림 5와 같이 실험 환경을 구성했으며, 자세한 설정은 표 1과 같다. 핸드오버 시나리오에서는 열차가 여러 eNodeB (Evolved Node B) 사이에 놓인 선로를 따라 직선으로 이동한다. 각 eNodeB는 교외 지역의 기지국 위치를 고려하여 1340m 간격으로 배치되었다^[25]. 선로와 기지국 사이의 거리는 현실 조건과 일치하도록 300m 간격으로 배치되었다.

핸드오버 기법은 A3 이벤트를 기반으로 한 X2 핸드오버를 사용하였다^[26]. 각 LTE 인터페이스는 25 Mbps의 전송속도를 가진다. 혼잡 제어 알고리즘은 LIA (Linked Increase Algorithm)^[27]를 사용했으며, 비교를

위한 스케줄러는 minRTT^[4], ECF (Earliest Completion First)^[7], BLEST (BLocking ESTimation-based)^[8], Peekaboo^[28]를 사용하였다. 핸드오버 테이블을 작성하기 위해 총 50회의 예비 시뮬레이션을 수행했다.

4.2 실험 결과

먼저, MPTCP의 각 서브플로우가 이 환경에서 어떻게 동작하는지 살펴본다. 그림 6(a)는 ISP A 및 ISP B를 사용하는 각각의 단일 TCP에 대한 시간에 따른 처리량을 나타낸다. 두 TCP의 평균 처리량은 2.4 Mbps로, 두 경로가 동등한 성능을 가짐을 볼 수 있다. 또한 그래프는 각 경로에서 동시에 핸드오버가 발생하지 않음을 보여준다. 핸드오버가 발생하면 각 TCP 연결의 순간 처리량이 약 20% 정도 감소하는 것을 관찰할 수 있다.

그림 6(b)는 동일한 실험 환경에서 minRTT 스케줄러를 사용하는 MPTCP의 각 서브플로우의 처리량을 나타낸다. 핸드오버가 발생하면 해당 서브플로우로 전송한 패킷이 예상보다 늦게 도착한다. 이로 인해 MPTCP 레벨에서 OFO 패킷이 발생한다. 결과적으로 HoLB 현상이 발생하며, 정상 경로의 패킷이 수신 버퍼에 갇히게 된다. 따라서 정상 서브플로우의 처리량도 감소하며, 이로 인해 전체적인 MPTCP 처리량이 감소한다.

그림 6(c)은 동일한 실험 환경에서 PHA 기반 minRTT 스케줄러를 사용하는 MPTCP의 각 서브플로우의 처리량을 나타낸다. PHA 기반 minRTT는 예상되는 핸드오버 위치에 접근할 때 핸드오버가 예상되는 서브플로우의 패킷 전송을 미리 차단한다. 이는 OFO 패킷에 의한 처리량 감소를 완화시킬 수 있다. 결과적으로 정상 서브플로우의 처리량 감소폭이 감소했음을 관찰할 수 있다.

그림 6(d)는 기본 minRTT 스케줄러와 PHA 기반 minRTT의 종합 처리량을 나타낸다. 핸드오버가 발생하지 않을 때는 두 스케줄러가 동일한 동작을 하므로

표 1. 실험 구성
Table 1. Experimental setup.

Simulation tool	Network Simulator 3
Experiment time	70 Seconds
Data rate of each ISP	25 Mbps
Congestion control algorithm	LIA
Packet schedulers	minRTT, ECF, BLEST, Peekaboo

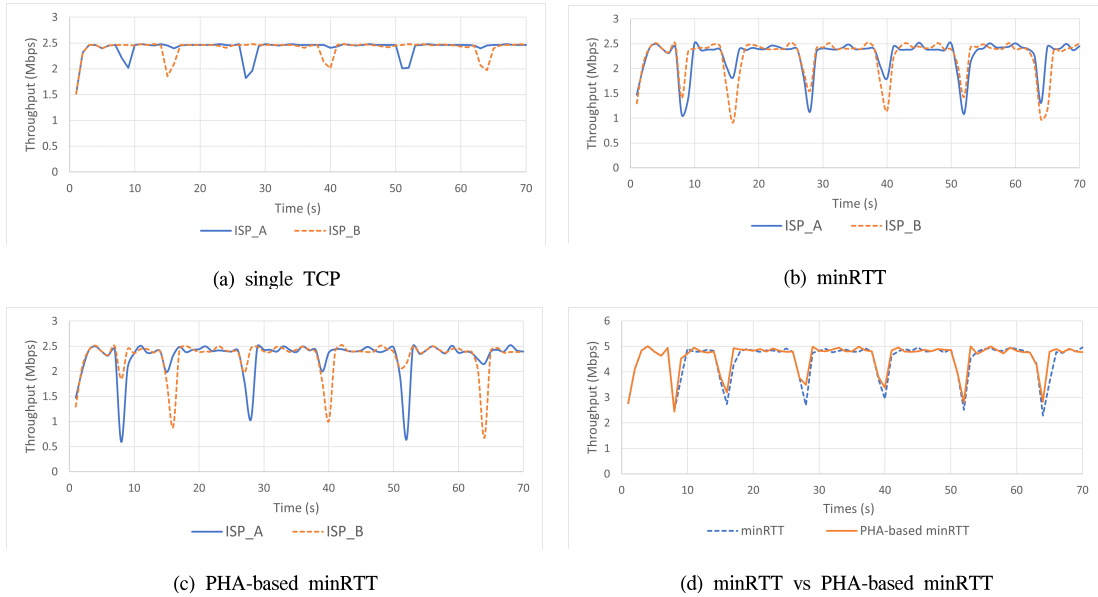


그림 6. minRTT와 PHA-based minRTT 간 처리량 비교
 Fig. 6. Comparison of throughput between minRTT and PHA-based minRTT.

동일한 성능을 보인다. 그러나 핸드오버가 발생할 경우 PHA 기반 minRTT는 기본 minRTT에 비해 평균 순간 처리량이 20.8% 향상되었다. 그러나 전체 실험에서 PHA 기반 minRTT는 1.95%의 평균 처리량 향상만을 보인다. 핸드오버 처리 시간은 일반적으로 수 밀리초 정도로 짧기 때문에 서버플로우 차단에서 얻는 이점은 단기간에만 해당된다.

이제 PHA 기반 스케줄러들의 전반적인 성능을 평가한다. 이 평가는 고속 열차 환경에 가장 적합한 스케줄러를 결정하는 데 도움이 될 것이다. 그림 7은 각 스케줄러의 평균 처리량을 나타낸다. PHA 스케줄러가 적용되지 않은 경우 처리량은 minRTT, ECF, BLEST, Peekaboo 순으로 높았다. 핸드오버는 물리적인 연결 중단 및 재연결 과정이므로 핸드오버로 인한 RTT 변화 및 패킷 재전송은 실제 네트워크 상태를 정확히 반영하지 않는다. 따라서 네트워크 상태에 따라 패킷을 스케줄링하는 Peekaboo와 같은 스케줄러는 일시적으로 네트워크 상태를 잘못 판단하고 불리한 결정을 내린다. PHA 스케줄러가 적용된 경우 모든 스케줄러에 대해 평균 처리량이 약 1.9% 향상된다. 각 스케줄러의 평균 처리량 향상 비율에 대한 자세한 내용은 표 2에 요약되어 있다. 최대 평균 처리량 향상을 보인 스케줄러는 ECF로, 실험 전체에서 평균 처리량이 2.2% 증가했다.

그러나 앞서 언급한 대로 핸드오버가 발생할 때 스케줄러 간에는 유의미한 성능 차이가 없다. 실험에서

사용된 모든 스케줄러는 minRTT 스케줄링을 기반으로 하고 있으며, 고속 열차의 MPTCP에서 사용하는 네트워크는 두 개의 LTE로 구성되어 있다. 동종경로로 구성된 환경에서는 minRTT 기반의 스케줄러들이 유사한

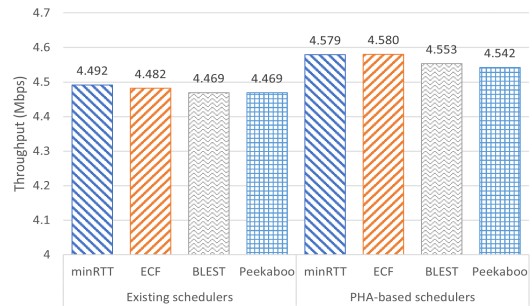


그림 7. 기존 스케줄러와 PHA 기반 스케줄러 간의 처리량 비교
 Fig. 7. Comparison of throughput between existing schedulers and PHA-based schedulers.

표 2. PHA 기반 스케줄러의 처리량 향상률
 Table 2. Throughput improvement ratio of PHA-based schedulers.

	PHA minRTT	PHA ECF	PHA BLEST	PHA Peekaboo
Throughput improvement ratio	1.95%	2.20%	1.87%	1.64%

동작을 나타내며, 스케줄러의 추가 메커니즘이 성능에 미치는 영향이 미미하다. 가장 큰 처리량 차이는 minRTT와 Peekaboo 간에 나타났으며, 약 0.5%의 차이를 보인다.

그림 8은 각 스케줄러에서 핸드오버가 발생하는 순간의 처리량을 보여준다. PHA를 사용하지 않을 때 모든 스케줄러의 평균 순간 처리량은 2.37 Mbps이다. 반면에 PHA 기반 스케줄러의 평균 순간 처리량은 2.84 Mbps로 측정되었다. 네 가지 스케줄러 전반에서 순간 처리량의 평균 향상률은 20.1%로, PHA 스케줄러가 핸드오버로 인한 순간 처리량 저하를 효과적으로 완화할 수 있음을 보여준다. 각 스케줄러의 순간 처리량 향상 비율에 대한 자세한 내용은 표 3에서 확인할 수 있으며 ECF가 가장 높은 향상을 보인다. MPTCP에서 핸드오버로 인한 순간 처리량 저하 문제는 동영상 스트리밍이나 온라인 게임 등 실시간 콘텐츠에 대한 사용자 경험을 저하시키는 주요 원인이다. PHA 스케줄러는 핸드오버 발생 시 적어도 하나의 LTE 만큼의 처리량을 유지할 수 있으며, 이러한 이점은 실시간 콘텐츠의 지연을 줄이고 사용자 경험을 향상시킬 수 있다.

표 4는 실험 중의 패킷 재전송 비율을 나타낸다. 고속 열차 환경에서 기존 스케줄러는 전체 패킷 중 평균

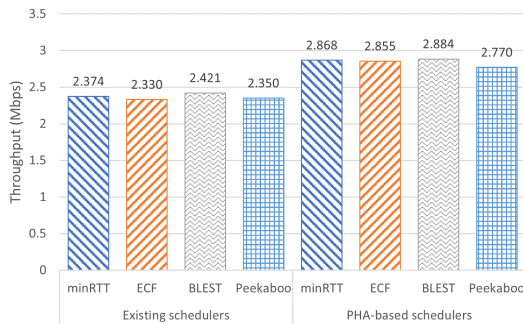


그림 8. 기존 스케줄러와 PHA 기반 스케줄러 간의 순간 처리량 비교
Fig. 8. Comparison of instantaneous throughput between existing schedulers and PHA-based schedulers.

표 3. PHA 기반 스케줄러의 순간 처리량 향상률
Table 3. Instantaneous throughput improvement ratio of PHA-based schedulers.

	PHA minRTT	PHA ECF	PHA BLEST	PHA Peekaboo
Instantaneous throughput improvement ratio	20.785%	22.536%	19.118%	17.879%

표 4. 패킷 재전송률 비교
Table 4. Packet retransmission ratio comparison.

Retransmission ratio	minRTT	ECF	BLEST	Peekaboo
Existing	0.301%	0.267%	0.269%	0.279%
PHA-based	0.243%	0.223%	0.239%	0.243%

0.279%의 패킷을 재전송했다. 반면 PHA 기반 스케줄러는 전체 실험에서 평균 0.237%의 패킷을 재전송했다. 이는 핸드오버 중 발생하는 지연 및 OFO 패킷이 효과적으로 감소되었음을 나타내며, PHA를 적용할 때 장기적으로 패킷 재전송으로 인한 오버헤드가 감소할 수 있음을 보여준다.

V. 결론

본 논문에서는 고속 열차 환경에서 다중 셀룰러 네트워크로 구성된 MPTCP를 사용할 때 발생할 수 있는 전송 프로토콜의 성능 저하 문제를 개선하기 위해 선제적 핸드오버 인지 스케줄러를 제안하였다.

실험을 통해 PHA 기반 스케줄러는 핸드오버 발생 지점에서 순간 처리량을 기존 스케줄러 대비 20.1% 향상시켜 단일 TCP 연결과 동등한 성능 수준을 제공할 수 있음을 보였다. 더 나아가 제안된 방법이 Linux 커널에 포함된 다양한 MPTCP 스케줄러에 유연하게 적용될 수 있음을 확인하였다. 각 스케줄러의 평가 결과에서 ECF가 가장 우수한 성능을 보였다. 다만, 각 스케줄러 간에 성능 차이는 미미하며, 커널 수정에 필요한 시간과 비용 등을 고려하면 기본 스케줄러인 minRTT를 사용하는 것이 바람직하다.

본 연구에서는 서버가 핸드오버 테이블 및 실시간 열차 위치 정보를 알고 있다고 가정하고 PHA 스케줄러를 사용하여 실험을 진행하였다. 그러나 PHA-D와 PHA는 모두 핸드오버를 유발할 것으로 예상되는 서브플로우를 차단하는 공통 특성을 가지고 있다. 따라서 다운로드에서 PHA-D를 활용하여 성능을 향상시킬 가능성을 확인할 수 있다.

PHA 스케줄러는 핸드오버 중의 일시적인 성능 저하 문제를 해결했지만, 전반적인 성능 향상은 제한적이었다. 이는 핸드오버 시간 자체가 매우 짧기 때문에 PHA 스케줄러를 통해 이점을 얻을 수 있는 절대적인 시간이 짧기 때문이다. 추후 연구를 통해 고속 열차 환경에서 MPTCP의 성능을 전반적으로 향상시키기 위한 스케줄러에 대한 연구를 진행할 예정이다.

References

- [1] Z. Li, et al., "5G URLLC: Design challenges and system concepts," *2018 15th ISWCS*, pp. 1-6, Lisbon, Portugal, 2018. (<https://doi.org/10.1109/ISWCS.2018.8491078>)
- [2] Q. Xiao, et al., "TCP performance over mobile networks in high-speed mobility scenarios," *2014 IEEE 22nd ICNP*, pp. 281-286, Raleigh, NC, USA, 2014. (<https://doi.org/10.1109/ICNP.2014.49>)
- [3] A. Ford, et al., "TCP extensions for multipath operation with multiple addresses," RFC 6824, 2013. [online]. Available: <https://datatracker.ietf.org/doc/html/rfc6824>
- [4] J. Hwang and J. Yoo, "Packet scheduling for multipath TCP," *2015 7th ICUFN*, pp. 177-179, Sapporo, Japan, 2015. (<https://doi.org/10.1109/ICUFN.2015.7182529>)
- [5] C. Paasch, et al., "Experimental evaluation of multipath TCP schedulers," in *Proc. ACM SIGCOMM Wkshp. CSWS '14*, pp. 27-32, New York, NY, USA, 2014. (<https://doi.org/10.1145/2630088.2631977>)
- [6] A. Frommgen, et al., "ReMP TCP: Low latency multipath TCP," *2016 IEEE ICC*, pp. 1-7, Kuala Lumpur, Malaysia, 2016. (<https://doi.org/10.1109/ICC.2016.7510787>)
- [7] Y. Lim, et al., "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *Proc. 13th CoNEXT*, pp. 147-159, New York, NY, USA, 2017. (<https://doi.org/10.1145/3143361.3143376>)
- [8] S. Ferlin, et al., "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," *2016 IFIP Netw.*, pp. 431-439, Vienna, Austria, 2016. (<https://doi.org/10.1109/IFIPNetworking.2016.7497206>)
- [9] N. Kuhn, et al., "DAPS: Intelligent delay-aware packet scheduling for multipath transport," *2014 IEEE ICC*, pp. 1222-1227, Sydney, NSW, Australia, 2014. (<https://doi.org/10.1109/ICC.2014.6883488>)
- [10] Y. E. Guo, et al., "Accelerating multipath transport through balanced subflow completion," in *Proc. 23rd MobiCom*, pp. 141-153, New York, NY, USA, 2017. (<https://doi.org/10.1145/3117811.3117829>)
- [11] L. Chao, et al., "A brief review of multipath TCP for vehicular networks," *Sensors* (Basel, Switzerland), vol. 21, no. 8, p. 2793, 2021. (<https://doi.org/10.3390/s21082793>)
- [12] A. Alheid, et al., "An analysis of the impact of out-of-order recovery algorithms on MPTCP throughput," *2014 IEEE 28th AINA*, pp. 156-163, Victoria, BC, Canada, 2014. (<https://doi.org/10.1109/AINA.2014.50>)
- [13] L. Li, et al., "A measurement study on multi-path TCP with multiple cellular carriers on high speed rails," in *Proc. ACM SIGCOMM 2018*, pp. 161-175, New York, USA, 2018, (<https://doi.org/10.1145/3230543.3230556>)
- [14] I. Khan and K. Chen, "EBA: efficient bandwidth aggregation for connected vehicles with MPTCP," in *IEEE Internet of Things J.*, vol. 9, no. 8, pp. 5812-5823, Apr. 2022. (<https://doi.org/10.1109/JIOT.2021.3065911>)
- [15] E. Dong, et al., "LAMPS: A loss aware scheduler for multipath TCP over highly lossy networks," *IEEE 42nd LCN*, pp. 1-9, Singapore, 2017. (<https://doi.org/10.1109/LCN.2017.69>)
- [16] Y. Xing, et al., "An online learning assisted packet scheduler for MPTCP in mobile networks," in *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2297-2312, Oct. 2023. (<https://doi.org/10.1109/TNET.2023.3246168>)
- [17] H. Zhang, et al., "ReLeS: A neural adaptive multipath scheduler based on deep reinforcement learning," *IEEE INFOCOM 2019*, pp. 1648-1656, Paris, France, 2019. (<https://doi.org/10.1109/INFOCOM.2019.8737649>)
- [18] M.-K. Kim and Y.-Z. Cho, "Enhancing MPTCP performance on high-speed trains with predictive handover-aware packet scheduling," *14th ICTC*, pp. 1224-1226, Jeju Island, Korea, 2023.

(<https://doi.org/10.1109/ICTC58733.2023.10393606>)

[19] N. Williams, et al., "Multipath TCP in vehicular to infrastructure communications," *Grenville Armitage Centre for Advanced Internet Architectures, Technical Report A, 140828*, 2014.

[20] S. Rene, et al., "Multipath TCP architecture for infotainment multimedia applications in vehicular networks," *2015 IEEE 81st VTC Spring*, pp. 1-5, Glasgow, UK, 2015. (<https://doi.org/10.1109/VTCSpring.2015.7145942>)

[21] D. Zhu, et al., "Mobility-aware multimedia data transfer using multipath TCP in vehicular network," *2017 13th IWCMC*, pp. 1041-1046, Valencia, Spain, 2017. (<https://doi.org/10.1109/IWCMC.2017.7986429>)

[22] C.-H. Park, et al., "Subflow blocking method to improve performance of MPTCP in high-speed train environments," *J. KICS*, vol. 47, no. 1, pp. 48-57, 2022. (<https://doi.org/10.7840/kics.2022.47.1.48>)

[23] T. Li, et al., "An in-depth analysis of subflow degradation for multi-path TCP on high speed rails," *2022 IEEE 23rd WoWMoM*, pp. 231-240, Belfast, United Kingdom, 2022. (<https://doi.org/10.1109/WoWMoM54355.2022.00042>)

[24] ns-3 Network Simulator, Retrieved Jan., 20, 2024, [online] Available: <https://www.nsnam.org/>

[25] Spectrummap, Retrieved Jan., 20, 2024, [online] Available: https://spectrummap.kr/gis/mobile_service.do?menuNo=300480.

[26] C. Lin, et al., "Handover algorithm with joint processing in LTE-advanced," *2012 9th ECTI-CON*, pp. 1-4, Phetchaburi, Thailand, 2012. (<https://doi.org/10.1109/ECTICon.2012.6254240>)

[27] C. Raiciu, et al., "Practical congestion control for multipath transport protocols," *Univ. College London*, London, U.K., Tech. Rep., 2009.

[28] H. Wu, et al., "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous Environments," in *IEEE JSAC*, vol. 38, no. 10, pp. 2295-2310, Oct. 2020. (<https://doi.org/10.1109/JSAC.2020.3000365>)

김민기 (Min-Ki Kim)



2022년: 금오공과대학교 전자공학부 학사
 2024년: 경북대학교 전자전기공학부 석사
 <관심분야> MPTCP, QUIC, 무선통신공학
 [ORCID:0009-0002-5048-7614]

김응협 (Eung-Hyup Kim)



2024년: 경북대학교 전자공학부 박사
 2017년~2018년: 한국정보통신기술협회 선임연구원
 2019년~2022년: 경북IT융합산업기술원 연구원

2024년 3월~현재: 경북대학교 자율근집소프트웨어 연구센터 연구원
 <관심분야> MPTCP 혼잡제어, SDN/NFV, 무선 에드혹 네트워크, 클라우드 컴퓨팅
 [ORCID:0000-0003-0448-1094]

강 순 주 (Soon Ju Kang)



1996년~현재: 경북대학교 IT 대학 전자공학부 교수
2011년~2027년: 자율군집소프트웨어 연구센터장
1983년~1996년: 한국원자력연구원(KAERI) 선임연구원 및 전산정보실장

2000년, 2007년: University of Pennsylvania, 방문 연구교수 정보과학회, 통신학회, 전자공학회 중신회원

<관심분야> 옛지컴퓨팅, 옛지 AI, 실시간 임베디드 시스템

[ORCID:000-0002-8171-195X]

조 유 제 (You-Ze Cho)



1982년: 서울대학교 전자공학과 학사
1983년: KAIST 전기전자공학 석사
1988년: KAIST 전기전자공학 박사
1989년~2023년: 경북대학교 전자

공학부 교수

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

2002년~2003년: NIST(미국국립표준연구소) 객원연구원

2017년: 한국통신학회 회장

<관심분야> 차세대 이동 네트워크, 무선 애드혹 네트워크, 이동성 관리 기술, 차세대 전송 계층 프로토콜

[ORCID:0000-0001-9427-4229]