

Multipath TCP (MPTCP) 표준화 및 기술개발 동향

고한얼, 이재욱, 백상현, 황재현*

고려대학교, Bell Lab Seoul*

요약

인터넷의 핵심 프로토콜인 TCP의 성능을 획기적으로 개선시키기 위해 다수의 경로/인터페이스를 동시에 사용하는 Multipath TCP (MPTCP)가 표준화되었다. 본고에서는 MPTCP에 대한 최신 표준화 동향과 연구개발 동향을 살펴보고 이를 통해 MPTCP와 관련된 향후 연구 이슈를 파악해 보도록 한다.

I. 서론

인터넷 환경이 복잡해짐에 따라 경로의 다양성(path diversity)이 증대되어 왔다. 많은 서버들 또는 자율 시스템(autonomous system)들이 안정된 연결성 유지를 위해 하나 이상의 인터넷 서비스 제공자(Internet service provider)와 연계하여 멀티호밍(multi-homing)을 지원하고 있고, 데이터센터와 같은 환경에서도 서버 간에 대부분 하나 이상의 경로가 존재한다. 이러한 다중 경로를 잘 활용하기 위해서 전통적으로 다중 경로 라우팅(multipath routing)이 연구되어 왔으나, 라우팅 프로토콜만으로 경로 다양성으로부터 최대의 성능을 끌어내기에는 부족함이 있다. 즉, 데이터 전송 시 가능한 모든 경로를 동시에 활용할 수 있어야 하며 이를 위해서는 전송계층 프로토콜의 도움이 요구된다.

MPTCP (Multipath TCP)는 이러한 요구사항을 만족시키기 위하여 설계된 TCP의 확장 버전 프로토콜이다. 기존의 TCP는 통신을 수행하는 두 노드 사이에 다중 경로가 존재하여도 오직 하나의 경로만으로 데이터를 주고 받는데 반해, MPTCP는 전송할 데이터를 둘 이상의 경로로 동시에 나누어 전송하는 것이 가능하다.

〈그림 1〉은 MPTCP를 사용할 때 응용 계층과 TCP 계층의 논리적 연결 구조를 보여준다. 응용 계층에서는 기존의 TCP와 같이 하나의 논리적 연결을 가지나, TCP 계층에서 이를 노드 간 연결 가능한 경로의 수만큼 서브플로우(subflow)를 생성하고,

응용으로부터 넘겨 받은 데이터를 생성된 서브플로우로 나누어 전송하는 구조를 가진다. 이때 각각의 서브플로우는 단일 경로(single path) TCP와 비슷하게 하나의 독립적인 연결처럼 다루어지기 때문에, 수신 측에서 각 서브플로우로 전달된 데이터를 잘 조합하여 전체 데이터의 신뢰성을 보장할 수 있어야 한다. 따라서 서브플로우의 순서 번호를 부여하는 방법과 이로부터 전체 데이터를 재순서화 하기 위한 방법 등이 설계 시에 고려되어야 하며, 흐름 제어 및 혼잡 제어에 대한 방법도 별도의 설계 과정이 필요하다. 그 외 다중 경로의 탐색 및 새로운 서브플로우의 생성 방법, 서브플로우 간의 패킷 스케줄링, 단일 경로 TCP와의 공평성 등과 같은 이슈가 존재한다. 또한 기존 인터넷 프로토콜들과의 호환성도 중요한 설계 요소 중에 하나이다.

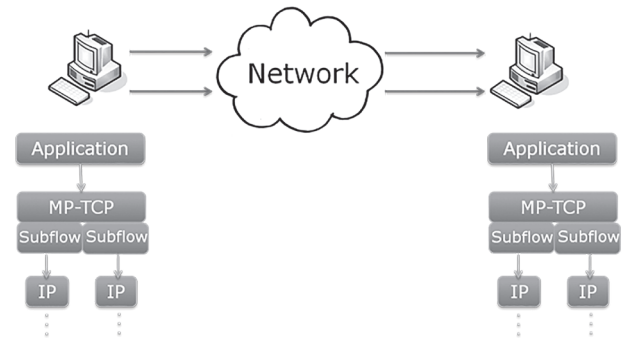


그림 1. MPTCP의 논리적 연결 구조

본 고에서는 MPTCP의 기본적인 구조와 동작과정 그리고 IETF MPTCP WG의 표준화 동향과 관련 연구 이슈 및 구현 현황 등을 살펴본다. 본 고의 구성은 다음과 같다. II장에서는 MPTCP의 기본적인 구조와 동작과정에 대해 설명한다. III장에서는 현재 IETF에서 진행되고 있는 MPTCP 표준화 동향에 대해서 살펴본다. 그 후, IV장에서는 MPTCP관련 이슈들과 해당 이슈를 해결하고 있는 연구들을 소개한다. V장에서는 MPTCP 구현 동향을 살펴보고 VI장에서 본 논문의 결론을 맺는다.

II. MPTCP 개요

MPTCP의 기본 구조는 <그림 2>와 같이 구성된다[1]. MPTCP는 기본적으로 2개의 계층으로 구성되는데 MPTCP 계층이 존재하고 그 밑에 다수개의 서브플로우 계층이 존재한다. MPTCP 계층에서는 연결관리와 어플리케이션 단으로 보내지는 패킷의 순서를 재조정하는 역할을 맡고, 각 서브플로우 계층은 신뢰적인 패킷 전송과 네트워크 혼잡제어를 맡는다.

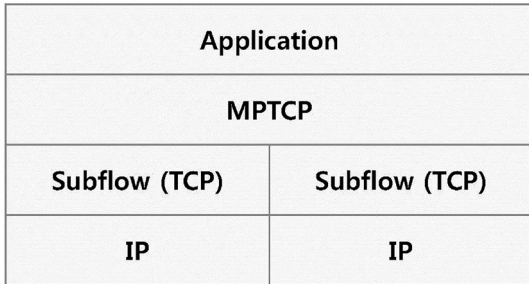


그림 2. MPTCP layer

기본적으로 MPTCP의 모든 동작 과정은 TCP header의 option field를 사용해 동작한다. MPTCP에 사용하는 옵션은 <표 1>에 정리되어 있다.

표 1. MPTCP option

Symbol	Name	Value
MP_CAPABLE	Multipath capable	0x0
MP_JOIN	Join connection	0x1
DSS	Data sequence signal	0x2
ADD_ADDR	Add address	0x3
REMOVE_ADDR	Remove address	0x4
MP_PRIO	Change sub-flow priority	0x5
MP_FAIL	Fallback	0x6

MPTCP의 초기 연결 설정은 <그림 3>과 같이 이루어 진다. MPTCP 초기 연결 설정은TCP에서와 마찬가지로 하나의 경로 상에서 SYN, SYN/ACK, SYN 패킷을 주고 받아 (즉, 3-way handshaking) 초기 연결 설정이 이루어 지는데, 각 패킷은 MP_CAPALBE (multipath capable) option을 포함하고 있다. MP_CAPALBE option에는 64-bits key가 포함되어 있는데, 이 키는 서브플로우를 추가할 때 사용된다.

MPTCP의 서브플로우 추가과정은 <그림 4>와 같다. 서브플로우 추가는 MP_JOIN (multipath join) option을 사용하여 진

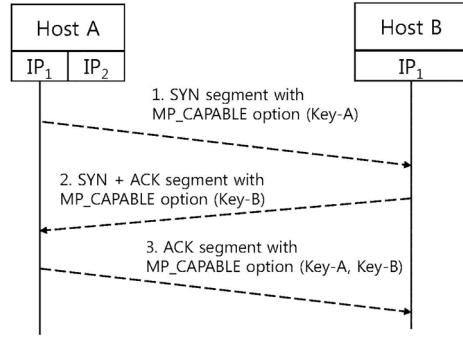


그림 3. MPTCP 초기 연결 설정 과정

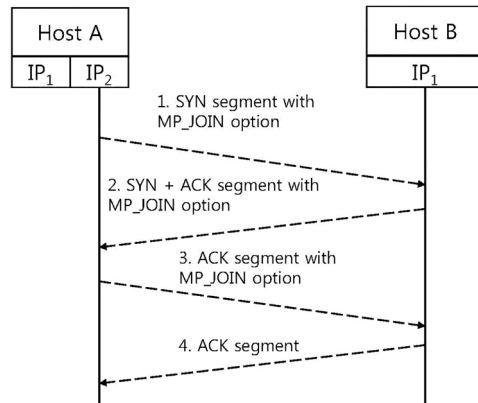


그림 4. MPTCP 서브플로우 추가 과정

행된다. 서브플로우 추가 시에 초기 연결설정에서 주고 받았던 key와 해쉬 함수를 이용해 다른 사용자의 침입을 방지한다. 초기 연결 설정과 유사하게 3-way handshaking 과정을 거치는데 이 패킷들은 MP_JOIN option을 포함한다 (Steps 1-3). 그 후, ACK 패킷을 받는 것으로 (Step 4) 서브플로우 추가 과정이 마무리 된다. 여기서 마지막 ACK 패킷이 필요한 이유는 Step 3의 MP_JOIN option에 포함된 key를 제대로 받았는지를 확인하기 위함이다.

MPTCP의 데이터 전송은 두 종류의 sequence 번호를 사용한다. 즉, 서브플로우 level에서의 sequence 번호와 connection level에서의 sequence 번호를 사용한다. 데이터 전송은 DSS (data sequence signal) option을 사용하여 전송하는데 이는 <그림 5>에 나와있다. 서브플로우 level sequence 번호는 각 서브플로우의 sequence 번호를 의미한다. 이 sequence 번호는

Kind	Length	Subtype	(reserved)	F	m	M	a	A
Data ACK (4 or 8 octets)								
Data sequence number (4 or 8 octets)								
Sub-flow sequence number (4 octets)								
Data-level length (2 octets)				Checksum (2 octets)				

그림 5. Sequence 번호

각 서브플로우의 flow control에 사용된다. 한편, connection level sequence 번호는 data sequence number field에 적혀 보내지는데 이는 어플리케이션 단으로 순차적인 데이터 전송을 위해 사용된다.

MPTCP 연결 종료 과정은 TCP에서와 마찬가지로 FIN 패킷을 통해 이루어진다. 단일 경로 TCP와의 차이점은 단일 경로 TCP에서는 하나의 FIN 패킷으로 해당 TCP 연결이 종료되지만, MPTCP에서는 각 서브플로우에서의 독립적인 FIN 패킷이 해당 서브플로우 연결만을 종료시킨다는 것이다. 즉, 두 개의 서브플로우가 연결되어 있는 MPTCP 세션에서 연결을 종료하기 위해서는 두 개의 서브플로우 모두에서 FIN 패킷을 전송해야 한다.

III. MPTCP 표준화 동향

IETF MPTCP WG은 2009년 7월 IETF-75 미팅에서 가진 BoF (birds of feather) 회의를 시작으로 2009년 10월에 처음 개설되었고, 같은 해 11월 IETF-76 미팅에서 첫 공식 WG 회의를 개최하였다. IETF MPTCP WG은 Transport Area 산하에 구성되었고, 의장으로는 Philip Eardley (BT)와 Yoshifumi Nishida (Keio university)가 선임되었다.

MPTCP WG에서 추구하는 MPTCP의 핵심 목표는 기존 인터넷 인프라의 큰 변화 없이도 MPTCP를 실제 환경에서 유용하게 쓸 수 있도록 만드는 것이며, 이는 응용 프로그램의 수정 없이도 사용 가능해야 한다는 것을 포함한다. 또한 NAT (network address translation) 등 다양한 종류의 인터넷 기능과도 안정적으로 동작해야 한다. MPTCP WG의 초기 차터(charter)는 전체 구현 구조와 보안 등에 초점을 두고 다음과 같이 정의되었다.

- 혼잡 종속적(congestion-dependent)인 다중 경로 전송 프로토콜을 위한 구조적 프레임워크 제안
- MPTCP를 위한 보안 위협 분석
- 결합된(coupled) 다중 경로 혼잡제어 알고리즘 개발
- 다수 개의 주소를 사용하는 MPTCP를 지원하기 위한 TCP 확장
- 응용 인터페이스에 대한 고려

이와 같은 차터의 방향에 따라 다중 경로 운용을 위한 TCP 확장에 대한 위협 분석 (RFC 6181), MPTCP 개발을 위한 구조적 가이드라인 (RFC 6182), MPTCP 혼잡 제어 알고리즘 (RFC 6356), 다중 경로 운용을 위한 TCP 확장 (RFC 6824), MPTCP API 고려 (RFC 6897) 등의 5개 RFC가 출판되었으며, 현재

MPTCP의 내재된 위협 및 수정 방안에 대한 분석에 대한 I-D가 IESG에 제출된 상황이다.

MPTCP WG은 최근 들어 다양한 유즈 케이스(use-cases) 및 운용적 경험(operational experiences)을 새로운 차터 아이템으로 정의하였고, 현재 관련 결과들을 수집하고 있는 중이다. 대표적인 유즈 케이스로는 3G에서 WiFi로의 트래픽 오프로딩(off-loading)과 데이터센터 내에서 트래픽 관리 등을 꼽을 수 있다. 또 다른 차터 아이템으로는 MPTCP 프록시(proxy)가 있다. 이것은 네트워크 상에 MPTCP용 미들박스(middlebox)를 두고, 두 단말 중 한쪽 단말만 MPTCP를 사용할 때 미들박스로 하여금 다른 단말을 위한 프록시 역할을 수행하도록 하는 것이다. 이는 MPTCP의 단계적 사용 확산에 도움을 줄 수 있다. MPTCP WG에서는 MPTCP용 미들박스가 해결할 수 있는 실제적인 문제들이 무엇인지에 대해 검토 중이며, MPTCP 구조에 어떠한 영향을 줄 것인가를 분석하고 있다.

2014년 7월 캐나다 토론토에서 열린 IETF-90에서 MPTCP 활용 경험에 관한 안건으로 다양한 내용들이 논의되었다. 먼저 MPTCP가 NAT, 방화벽, 세그먼트 분리(splitting) 등의 잘 알려진 미들박스 기능 상에서 잘 동작하는지에 대한 실험 내용이 보고되었다. 유즈 케이스로 선택된 데이터센터와 모바일 디바이스에 대한 내용에서는 MPTCP가 RFC6824에서 기술된 한쪽 또는 양쪽 모두 멀티호밍, 멀티어드레스 지원이 되어야 한다는 가정을 깨고, 싱글호밍(single-homing) 데이터센터 서버들 간에 ECMP (Equal Cost Multipath Routing)를 기반으로 트래픽 부하 분산을 위해 사용될 수 있음을 보여주었다. 모바일 디바이스의 경우 iOS7의 시리(Siri) 응용에서 MPTCP가 사용됨이 알려졌지만, 안타깝게도 MPTCP 커뮤니티에서는 이에 대한 아무런 피드백을 받을 수 없었으며, 현재는 안드로이드 스마트폰으로 포팅된 버전을 통해 스마트폰 상에서의 MPTCP 실험 결과를 수집하는 중이다. 특히 모바일 유즈 케이스에서는 혼잡하거나 채널 상태가 매우 안 좋은 링크로 데이터를 전송할 시에 MPTCP의 성능에 악영향을 줄 수 있는 문제점을 지적하였고, 이를 해결하기 위해 MPTCP 호스트로 하여금 적절한 시점에 RST 패킷을 통해 해당 서브플로우를 종료할 수 있도록 하는 방안을 모색하고 있다. 그 외에 다양한 실험적, 운용적 경험들을 통해 RFC6824가 매우 견고한 내용임을 밝혔고, 패킷 스케줄링과 경로 관리(path/subflow management) 등을 통한 성능 향상의 여지가 남아 있음을 확인하였다.

MPTCP 프록시 역시 하나의 중요한 유즈 케이스로써 논의되고 있다. 스마트폰과 같은 사용자 디바이스들은 3G/LTE와 WiFi를 동시에 지원하는데 반해 대부분의 인터넷 서버들은 아직까지 일반 TCP 서버를 사용하고 있으며, 이 경우 MPTCP로

부터 얻을 수 있는 이득이 매우 제한된다. MPTCP프록시의 주요 목표는 이와 같은 상황에서 그림6과 같이MPTCP 연결과 일반 TCP 연결 사이에서 데이터 매핑 기능을 수행하는 것이다. 즉, MPTCP 프록시를 통해서 사용자 단말은 상대방의 MPTCP 지원 여부와 상관없이 여러 네트워크 인터페이스를 동시에 활용 할 수 있다.

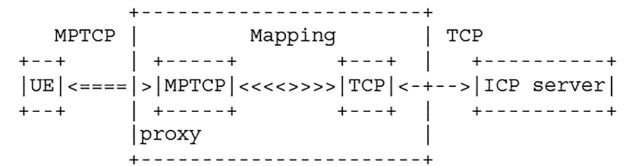


그림 6. MPTCP 프록시의 데이터 매핑 기능

일반적으로 MPTCP 프록시 메커니즘은 프록시의 위치에 따라 on-path와 off-path의 두 가지 모델로 나뉜다. on-path 프록시 모델은 단말에서 접근 가능한 서로 다른 액세스 네트워크들이 서버로 가는 라우팅 경로를 공유하는 환경에 적용 가능하다. 예를 들어 그림 7에서 단말이 3GPP 셀룰러 네트워크와 무선랜 네트워크를 통해 ICP 서버로 연결을 시도하고 있고, 이때 두 네트워크가 같은 네트워크 오퍼레이터(network operator)에 의해 관리가 된다면 오퍼레이터는 프록시의 위치를 두 네트워크 트래픽이 공유하는 경로 상에 둘 수 있다. 즉, 프록시가 단말에서 서버로 가는 모든 서버플로우들을 감지하여 단말과는 MPTCP 연결을, 서버와는 그에 대응되는 TCP 연결을 맺어 통신을 유지시킨다. 이 경우 단말은 프록시의 존재에 대해 알 필요가 없다.

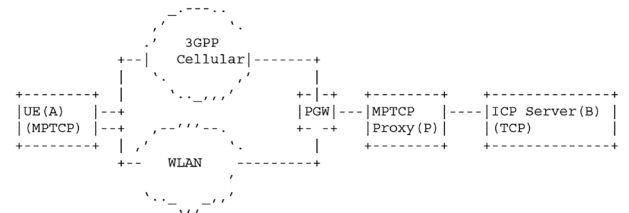


그림 7. On-path 프록시 시나리오

반면 off-path 프록시 모델은 그림 8과 같이여러 액세스 네트워크로부터 발생하는 트래픽 중 어느 한쪽 트래픽만이 프록시를 경유하게 되는 환경에 적용된다. 이 모델에서는 프록시를 지나는 서버플로우를 통해 MPTCP 사용자가 있음을 감지하게 되면, 명시적으로 단말에게 프록시의 존재를 알리고 프록시와 직접적으로 MPTCP 연결을 맺도록 한다. 이때 다른 서버플로우들 역시 연결 대상을 ICP 서버가 아닌 프록시로 변경시킨다. 이 두 가지 모델을 기반으로 현재 MPTCP WG에서는 MPTCP

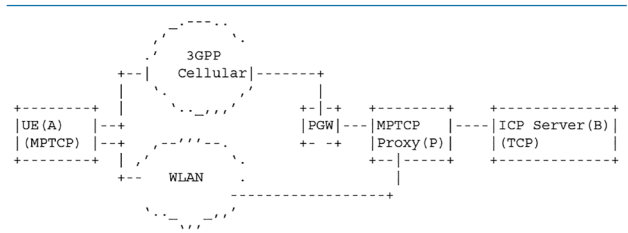


그림 8. Off-path 프록시 시나리오

프록시를 지원하기 위한 다양한 요구사항들을 분석하고 있다.

IV. MPTCP 연구 동향

본 장에서는 MPTCP 관련 연구 이슈들을 살펴보고 이를 해결하는 방법들에 대해 알아본다.

1. 공평성

MPTCP를 사용하게 되면 단일 경로 TCP와의 공평성 문제가 발생할 수 있다 [2-6]. 즉, 단일 경로의 TCP 연결보다 다중 경로의 TCP 연결이 네트워크 자원을 더 많이 사용하게 되는 문제점이 발생할 수 있다. 논문 [2]에서는 AIMD (adaptive-increase and multiplicative-decrease) 기반의 혼잡제어 알고리즘을 가정하여 공평성 문제를 해결하는 방법을 제시하였다. 즉, 혼잡 제어 알고리즘의 파라미터 (즉, α , β)를 조절하여 공평성 문제를 해결하고 있다. AIMD기반의 혼잡 제어 알고리즘은 네트워크 혼잡도가 크지 않다고 생각되어 congestion window size를 증가시킬 때는 α 만큼 window size를 증가시키고, 네트워크 혼잡도가 클 경우에는 현재 window size에 β 를 곱한 값을 사용하는 방식이다. 이 때, 다중 경로의 TCP연결이 단일 경로의 파라미터 (즉, α , β) 값과 같은 값을 사용하게 되면 단일 경로의 TCP가 대역폭 할당에 불리함을 겪게 된다. 이를 해결하기 위해 논문 [2]에서는 수학적 모델링을 통해 α 와 β 값을 설정하는 방법을 제시하였다. 논문 [3]에는 공평성 문제를 해결하기 위해서 몇 개의 혼잡 제어 알고리즘을 제시하고 이를 다양한 시나리오에 적용하여 그 타당성을 검증하였다. 이와 유사하게 논문 [4][5]에서는 기존의 혼잡 제어 알고리즘들을 소개하고, 공평성 문제 해결뿐 아니라 처리율 향상을 동시에 이루기 위해 기존 알고리즘의 단점을 개선한 새로운 혼잡제어 알고리즘을 제시했다. 그 후, QualNet 시뮬레이터를 이용하여 여러 가지 시나리오를 만들고 이에 따른 성능 평가를 진행하였다. 한편, 논문 [6]에서는 현재 표준 MPTCP 알고리즘인 LIA (linked-increases algorithm)가 파레토 최적이지 아니라는 것을 증명하였

다. 즉, 단일 경로 TCP 단말을 MPTCP를 사용할 수 있도록 업그레이드하더라도 해당 단말의 처리율을 올릴 수 없을 뿐 아니라 다른 유저의 처리율을 감소시킬 수 있는 것을 증명하였다. 이를 해결하기 위해, OLIA (opportunistic linked-increase algorithm)를 제시하고 이 알고리즘이 파레토 최적인 것을 보여주었다.

2. 프록시 서버

가까운 미래에는 MPTCP를 지원하는 서버가 많지 않을 것이라 예상된다. 또한, 다중 경로로 연결을 맺기 위해서는 단말이 사용 가능한 인터페이스로 각각 컨트롤 메시지를 교환해야 하기 때문에 긴 종단간 딜레이로 인해 해당 인터페이스의 성능을 충분히 사용할 수 없는 경우가 발생할 수 있다. 이를 해결하기 위해 프록시 서버를 이용해 MPTCP를 지원하는 방안이 제시되었다 [7]. <그림 9>는 논문 [7]에서 제시한 프록시 서버 역할을 하는 MA (multipath anchor)를 두어 MPTCP를 지원하는 구조를 보여준다. 이와 같은 구조를 사용하게 되면 단말과 MA 사이의 딜레이가 짧기 때문에 빠른 인터페이스 사용이 가능하고 MPTCP를 사용하지 않는 서버와 연결 시에도 단말의 인터페이스를 모두 사용할 수 있다는 장점이 있다.

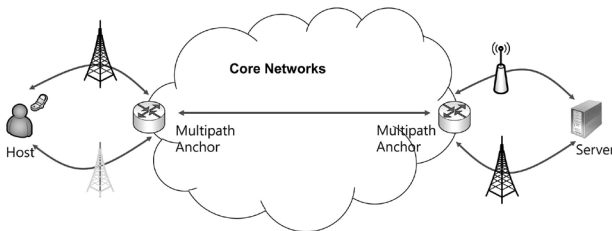


그림 9. Multipath anchor 사용 MPTCP 지원 방안

한편, 논문 [8]에서는 MPTCP가 사용될 수 있는 여러 가지 구조를 소개하였는데, 이 중 프록시 서버를 이용한 구조도 제시하였다. 이 구조는 단말이 MPTCP stack을 가지고 있지 않더라도 프록시 서버를 통해 여러 경로를 이용하여 데이터를 전송할 수 있기 때문에 트래픽을 분산시킬 수 있다.

3. 이동성에 의한 무선자원의 변화

MPTCP를 사용하는 단말이 이동하게 되면 사용할 수 있는 무선자원이 변화하게 된다. 예를 들어, WiFi 같은 경우 전송 범위가 작기 때문에 단말이 이동하면서 MPTCP를 사용할 경우 간헐적으로 WiFi 자원을 사용할 수 있다. 이를 고려하지 않고 WiFi 인터페이스를 사용한다면 긴 재전송 타이머로 인해 해당 인터페이스를 제대로 사용할 수 없다. <그림 10>은 30초에서 한 인

터페이스를 사용하지 못하게 되고, 60초에 다시 그 인터페이스가 사용 가능하게 됐을 경우에 TCP sequence 번호의 변화량을 나타내는 그래프이다 (즉, 이 그래프의 기울기는 MPTCP의 처리율을 나타냄)[9]. 60초에서 인터페이스가 사용 가능하게 됐음에도 기울기가 변하는 시점은 75초가 지나면서부터인 것을 확인할 수 있다. 이는 긴 재전송 타이머로 인해 해당 인터페이스를 제대로 사용할 수 없음을 나타낸다.

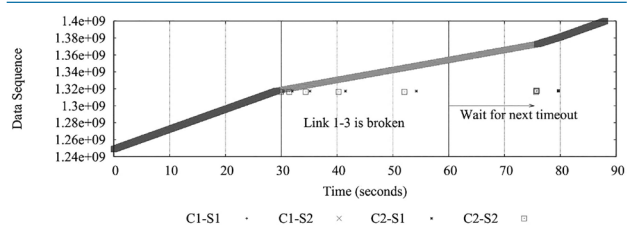


그림 10. 재전송 타이머에 의한 처리율 감소 예제

이를 해결하기 위해 인터페이스의 사용 여부를 결정하는 MPTCP-MA (MPTCP with MAC-layer awareness)가 제안되었다 [9]. 시그널 강도를 통해 인터페이스의 사용 여부를 결정하게 되면 noise, fading 과 같은 주위 환경에 따라 성능이 저하 될 수 있다. 이를 해결하기 위해 실험을 통한 시그널 강도와 재전송 횟수의 상관 관계를 보여주고 재전송 횟수를 통해 인터페이스의 사용 여부를 결정하는 알고리즘이 제시되었다. 한편, 논문 [10]에서는 MPTCP를 사용하는 단말의 이동성 관리 방법을 proxy mobile IP (PMIP)에 기반하여 제시하였다.

4. 처리율 향상

각 인터페이스의 처리율을 고려하지 않고 여러 개의 네트워크 인터페이스를 동시에 사용하게 되면 패킷을 순차적으로 어플리케이션 단으로 올려야 하기 때문에 가장 느린 인터페이스 처리율에 따라 전체 처리율이 결정되게 되는 문제가 발생한다. 이를 해결하기 위해 여러 연구들이 진행되었다[11-14]. 논문 [11]에서는 각 서브플로우의 종단간 딜레이 차이로 전체 처리율이 결정이 되는 것을 수학적으로 증명한 뒤에 딜레이 차이가 최대한 발생하지 않도록 새로운 MPTCP의 혼잡 제어 알고리즘을 제시하였다. 이 알고리즘은 각 서브플로우의 심한 처리율 감소를 방지하기 위해 최대 감소 횟수를 정해두고, 가장 긴 딜레이를 가지는 경로의 윈도우 크기를 줄이는 것으로 각 서브플로우의 종단간 딜레이 차이를 줄인다. MPTCP의 처리율을 향상시키기 위해 network coding을 사용하는 연구들도 진행되었다 [12][13]. NC-MPTCP (Network coding based MPTCP)에서는 각 서브플로우를 보통 서브플로우와 네트워크 코딩 서브플로우로 구분 하여, 보통 서브플로우는 일반적인 데이터를 보

내게 되고 네트워크 코딩 서브플로우로는 네트워크 코딩을 사용하여 보통 서브플로우로 보낸 데이터의 선형 결합을 보내 처리율을 향상시키는 방법을 제안하였다[12]. 이와 유사하게 논문 [13]에서는 무선 메시 네트워크 상황에서 네트워크 코딩을 사용하여 처리율을 향상시키는 방법을 제시하였다. 한편, 논문 [14]에서는 각 인터페이스의 상태에 따른 패킷 스케줄링 방법과 수신단 기반의 패킷 손실 복구 방법이 소개되었다. 이 방식은 패킷이 각 전송 경로를 통해 전송될 경우에 수신단에 도착할 예상 시간을 구한 뒤에, 수신단에 빨리 도착하는 순서로 패킷을 정렬한 뒤에 이를 기반으로 각 경로에 각 경로에 패킷을 할당해 주는 방식이다.

V. MP-TCP 구현 동향

이동 단말이 여러 개의 네트워크와의 연결이 가능해지면서 MT-TCP에 대한 연구와 구현이 지속적으로 진행되고 있다.

벨기에 Louvaion대학교에서 MPTCP를 Linux kernel을 통해 구현하였다. 구현된 MPTCP 구조는 <그림 11>과 같다. 구현된 MPTCP는 3가지 요소로 구성되어 있다. 첫 번째 구성 요소는 어플리케이션과 TCP Kernel 사이 통신 인터페이스를 제공하는 master subsocket이다. 두 번째 구성 요소는 multiple flow를 관리하는 MPCB (multi path control block)이고, 마지막 구성 요소는 어플리케이션에서는 감지할 수 없지만 master subsock과 함께 서브플로우를 형성하는 slave subsock이다 [15]. 이렇게 구현된 MPTCP는 보안 소스코드 및 공동개발 업체인 Github를 통해 배포되고 있다[16].

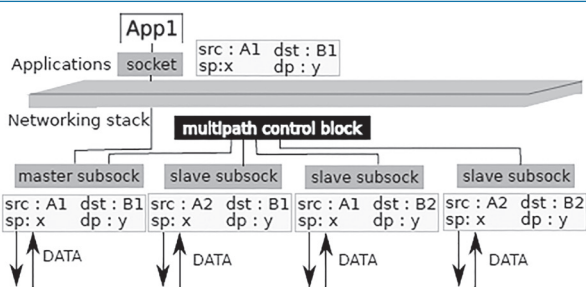


그림 11. Linux kernel을 이용한 MPTCP 구현

또한 이동 단말 생산기업인 애플에서는 iOS7에 MPTCP를 구현하였다. 구현된 MPTCP는 iOS7를 사용하는 애플의 이동 단말 (예, iPhone, iPad등)에 두 가지 유형 (즉, WiFi를 통한 기본 TCP 연결, 셀룰러 네트워크를 통한 백업 연결)을 제공한다. 이를 위해 이동 단말은 먼저 WiFi를 통한 MPTCP연결을 시도

하고, 성공시 셀룰러 네트워크 연결을 통한 백업 TCP 연결을 설정한다. 하지만 WiFi가 사용할 수 없거나 신뢰할 수 없으면 이동 단말은 MPTCP를 셀룰러 네트워크 통한 표준 TCP연결을 시도한다. 즉, iOS7에 구현된 MPTCP는 셀룰러 네트워크와 WiFi의 동시전송(simultaneous transmission)을 사용하는 것이 아니라 빠른 네트워크 인터페이스 전환을 통한 끊김 없는 연결(seamless connection) 지원에 무게를 두고 있다. 예를 들어 iOS7의 대표적인 어플리케이션인 시리의 경우 사용자가 질문을 하면 우선적으로 WiFi를 통한 MPTCP 연결을 시도한다. 만약 WiFi를 사용할 수 없거나 신뢰할 수 없으면 3G나 LTE네트워크를 통한 표준 TCP연결을 하여 데이터를 전송한다. 그렇지 않고 WiFi를 통한 MPTCP연결이 성공하면 3G나 LTE네트워크를 통한 백업 연결을 한다. 이러한 백업 연결이 WiFi와 3G, LTE네트워크간의 빠른 전환을 제공해 주면서 끊김 없는 연결서비스를 제공해준다[17].

SK 텔레콤에서도 MPTCP를 구현하고 있다. SK 텔레콤에서 구현한 MPTCP는 '네트워크 전송률 독립 조절' 기술이 적용돼 한 네트워크의 성능이 저하돼도 나머지 네트워크들은 영향을 받지 않는 장점을 가지고 있다. 구현된 MPTCP는 우선적으로 LTE네트워크와 WiFi를 묶는데 사용할 예정이다. LTE네트워크와 WiFi를 결합하면서 사용자들은 서비스 중인 광대역 LTE-A (최대 속도 225Mbps)와 기가WiFi (최대 속도 866.7Mbps) 두 네트워크를 결합하여 이론적으로 최대 1Gbps가 넘는 속도로 서비스를 제공할 수 있다. 또한 SK텔레콤은 기술적용이 가능한 협력사의 스마트기기 및 MPTCP 최적화 서버의 개발 계획에 맞춰 지속적으로 구현해갈 예정이다[18].

그 밖에도 호주의 Swinburne 대학교에서는 FreeBSD를 통해 MP-TCP를 구현하였다. 또한 어플리케이션 트래픽 관리 솔루션 제공업체인 F5네트웍스에서는 네트워크 단일 통합 솔루션인 BIG-IP LTM을 통해 MPTCP를 하고, 가상 컴퓨팅 솔루션 기업인 Citrix system은 Netscaler를 통해 MPTCP를 지원하고 있다[19].

한편, 기존 TCP연결에 적용된 많은 상업용 네트워크 장비들은 보안의 목적으로 알려지지 않은 TCP 옵션필드를 변형시킨다. 이러한 과정은 MPTCP연결을 불가하게 만든다. 또한 일반 홈 네트워크에서의 네트워크 장비들도 MPTCP를 지원하지 않고 있다. 이러한 네트워크에서 MPTCP지원의 문제점을 해결하기 위해 네트워크 기기 회사인 Cisco에서는 MPTCP의 구현을 도와줄 수 있는 네트워크 기기들의 목록을 제공하고 있다 [20].

VI. 결론

유무선 네트워크에서 다수의 경로/인터페이스를 동시에 사용하여 처리율을 향상시키고자 하는 연구는 지속적으로 진행되어 왔으며 이러한 환경에 최적화된 전송 계층 프로토콜이 MPTCP이다. 특히, 무선 네트워크에서의 제한된 대역폭 문제를 극복하기 위해서 다수의 모바일 기기 개발 업체 및 이동통신사업자가 MPTCP를 활발히 채택하고 있다. 향후에는 모바일 환경에 대한 고려와 사용자 체감 성능 향상 측면에서 보다 많은 연구 개발이 진행될 것으로 예상된다.

참고 문헌

- [1] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," IETF RFC 6824, January 2013.
- [2] D. Zhou, W. Song, and Y. Cheng "A Study of Fair Bandwidth Sharing with AIMD-Based Multipath Congestion Control," IEEE Communications Letters, vol. 2, no. 3, June 2013.
- [3] M. Becke, T. Dreiholz, H. Adhari, and E. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment," in Procs. IEEE International Conference on Communications (ICC) 2012, June 2012.
- [4] A. Singh, M. Xiang, and Y. Zaki, "Enhancing Fairness and Congestion Control in Multipath TCP," in Procs. IFIP Wireless and Mobile Networking Conference (WMNC) (ICC) 2013, April 2013.
- [5] A. Singh, M. Xiang, A. Konsgen, and C. Goerg "Performance and Fairness Comparison of Extensions to Dynamic Window Coupling for Multipath TCP," in Procs. IEEE Wireless Communications and Mobile Computing Conference (IWCMC) 2013, July 2013.
- [6] R. Khalili, N. Gast, M. Popovic, and J. Boudec, "MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution," IEEE/ACM Transactions on Networking, vol. 21, no. 5, pp. 1651-1665, October 2013.
- [7] T. You, C. Park, H. Jung, T. Kwon, and Y. Choi, "Multipath Transmission Architecture for Heterogeneous Wireless Networks," in Procs. International Conference on ICT Convergence (ICTC) 2011, September 2011.
- [8] T. Leva, H. Warma, A. Ford, A. Kostopoulos, B. Heinrich, R. Widera, and P. Earcley, "Business Aspects of Multipath TCP Adoption," Future Internet Assembly (FIA) 2010, April 2010.
- [9] Y. Lim, Y. Chen, E. Nahum, D. Towsley, and K. Lee, "Cross-Layer Path Management in Multi-path Transport Protocol for Mobile Devices," in Procs. IEEE International conference on Computer Communications (INFOCOM) 2014, April/May 2014.
- [10] L. Tuan, K. Kim, J. Choe, and S. Ro, "A Study on Multi-Path TCP Mobility Management Protocol," Journal of Korean Institute of Information Technology (KIIT), vol. 12, no. 6, pp. 109-117, June 2014.
- [11] D. Zhou, W. Song, and M. Shi "Goodput Improvement for Multipath TCP by Congestion Window Adaption in Multi-Radio Devices," in Procs. IEEE Consumer Communications and Networking Conference (CCNC) 2013, January 2013.
- [12] M. Li, A. Lukyanenko, and Y. Cui, "Network Coding Based Multipath TCP," in Proc. IEEE Computer Communications Workshop (INFOCOM Workshop) 2013, April 2013.
- [13] X. Zhuoqun, C. Zhigang, and Y. Hui, Z. Ming, "An Improved MPTCP in Coded Wireless Mesh Networks," in Procs. IEEE International Conference on Broadband Network & Multimedia Technology (IC-BNMT) 2009, October 2009.
- [14] B. Oh, H. Kim, and J. Lee, "Packet Scheduling Scheme and Receiver-Based Recovery Scheme for MPTCP in Heterogeneous Networks," Journal of Korea Information and Communications Society (JKICS), vol. 37B, no. 11, pp. 975-983, Journal of Vol.37B No.11, November 2012.
- [15] Sebastien Barre, Christoph Paasch, and Olivier Bonaventure, "MultiPath TCP: From Theory to Practice," in Networking, vol.6640, 2011.
- [16] Multipath TCP-Linux Kernel Implementation.

[Online]Available : multipath-tcp.org/pmwiki.php/Users/Android

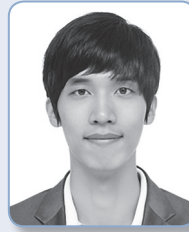
[17] How Apple Could Boost Speeds 20 Times on The Next iPhone. [Online]Available : <http://www.technologyreview.com/news/519646/how-apple-could-boost-speeds-20-times-on-the-next-iphone/>

[18] SKT, 광대역 LTE-A보다 4배 빠른 주파수 기술 개발. [Online]Available: <http://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=105&oid=421&aid=0000942021>

[19] Multipath TCP. [Online]Available: http://en.wikipedia.org/wiki/Multipath_TCP

[20] MPTCP and Product Support Overview. [Online]Available: <http://www.cisco.com/c/en/us/support/docs/ip/transmission-control-protocol-tcp/116519-technote-mptcp-00.html>.

약 력



고 한 열

2011년 고려대학교 전기전자전파공학부 학사
2011년~현재 고려대학교 전기전자공학과 석박사
과정
관심분야: 이동성/핸드오버 관리, 미래 인터넷



이 재 욱

2014년 고려대학교 학사
2014년~현재 고려대학교 전기전자전파공학부
석박사 과정
관심분야: IP 이동성/핸드오버, MPTCP,
미래인터넷구조



백 상 현

2000년 서울대학교 컴퓨터공학부 학사
2005년 서울대학교 전기컴퓨터공학부 박사
2007년~현재 고려대학교 전기전자전파공학과
부교수
관심분야: 미래 인터넷, 무선 이동 네트워크



황 재 현

2003년 가톨릭대학교 공학사
2005년 고려대학교 이학석사
2010년 고려대학교 이학박사
2010년~현재 서울 벨 연구소 책임연구원
관심분야: MPTCP, data center networks,
HTTP adaptive streaming