

논문 2006-43TC-3-15

FHMIPv6 핸드오버에서 TCP 성능 향상을 위한 재정렬 알고리즘 설계 및 성능 분석

(Design and Performance Evaluation of Resequencing Algorithm for
TCP Performance Enhancement in FHMIPv6 Handover)

황 안 규*, 이 재 용**, 김 병 철**, 이 재 훈***

(An Kyu Hwang, Jae Yong Lee, Byung Chul Kim, and Jae Hoon Lee)

요 약

FHMIPv6에서 이동단말은 핸드오버시에 지연시간과 자원소비를 줄이기 위한 HMIPv6 프로토콜과 패킷 손실을 줄이기 위한 fast 핸드오버 알고리즘의 장점을 동시에 갖는다. Fast 핸드오버 알고리즘은 핸드오버시에 이전 라우터로 전송되는 패킷을 터널링을 통해 새로운 라우터로 전송해 줌으로써 패킷 손실을 줄일 수 있다. 반면 송신측에서 이전 라우터를 경유하여 새로운 라우터로 터널링 되어 전송되는 패킷과 새로운 라우터로 직접 전송되는 패킷의 순서가 뒤바뀌어 혼잡제어를 발생시킴으로써 TCP 성능을 저하시키는 문제점을 가지고 있다. 본 논문에서는 데이터 세그먼트의 순서 어긋남 현상을 해결하기 위해서 홀딩 타이머를 이용하는 방법과 snoop 프로토콜을 탑재한 라우터에 새로운 알고리즘을 추가하는 방법의 두 가지 재정렬 알고리즘을 제안하였다. 또한 시뮬레이션을 통해 제안된 재정렬 알고리즘과 기존의 FHMIPv6 프로토콜과 성능을 비교 분석하였고, 제안된 재정렬 알고리즘을 적용한 fast 핸드오버 알고리즘이 데이터 세그먼트의 순서 어긋남 현상을 해결하여 혼잡제어를 방지함으로써 TCP 성능을 향상시킴을 보였다.

Abstract

Mobile nodes in FHMIPv6 has both advantages of HMIPv6 protocol which reduces signaling delay time and resource consumption during a handover and fast handover algorithm which reduces packet loss. Fast handover algorithm can reduce packet loss by 'tunneling' method ; that transmits a packet from old access router to new access router in case of handover. However, the fast handover algorithm can cause a reordering problem in a receiver between packets tunneled from the previous access router and packets transmitted directly to the new access router, which could degrade the TCP performance due to congestion control. In this paper, we propose two algorithms to solve the reordering problem in fast handover. The first one uses a holding timer for tunneling, the other adds a new algorithm to routers that adopt snoop protocol. We compare the performance of the proposed resequencing algorithms with that of the existing FHMIPv6 protocol by simulation. The simulation results show that the proposed algorithms solve the reordering problems and enhance TCP performance by preventing TCP sender entering congestion control.

Keywords : Mobile IPv6, Fast handover, resequencing algorithm, TCP performance

I. 서 론

무선 이동 인터넷 서비스에 대한 연구가 활발해지면 서 가입자의 액세스 망 이동에 상관없이 연속적인 데이터 서비스가 가능하도록 하는 mobile IP 규약이 IETF를 중심으로 제안되었고 이에 대한 구현 및 서비스가 몇몇 선진국을 중심으로 활발히 이루어지고 있다^[1]. 이러한 이동 인터넷 서비스는 기존 서비스의 범위를 크게 확장하였고 따라서 향후 인터넷 시장의 주 패러다임

* 학생회원, 충남대학교 정보통신공학과
(Department of Information and Communications Engineering, Chungnam National University)

** 중신회원, 충남대학교 전기정보통신공학부
(Electronics, Radio, Information communications Engineering, Chungnam National University)

*** 중신회원, 동국대학교 정보통신공학과
(Department of Information and Communications Engineering, Dongguk University)

※ 본 논문은 한국과학재단의 특정기초연구(R01-2003-000-10628-0) 지원으로 수행되었음.

접수일자: 2005년8월23일, 수정완료일: 2006년3월14일

로 등장하게 될 것이다. 그러나 이동성 처리를 위해 제안된 mobile IP 규약의 경우 초기에 이동 가입자를 충분히 고려하지 못하여 가입자의 이동 여부 발견, 새로운 임시 주소 획득 및 신규 위치 등록 등에 시간이 많이 소요되고 전체적으로 망에 많은 부하를 주는 형태로 만들어졌다. 또한 고속으로 이동하는 가입자의 경우 Home Agent(HA)와 현재 이동 단말이 방문 중인 외부 망과의 거리가 멀다면 등록을 위해 소요되는 시간이 오래 걸려 등록 전까지 이전 망으로 전달되는 데이터의 유실이 일어날 가능성이 높으며 제어 메시지가 등록을 위해 사용하는 자원이 많아지는 단점이 있다.

이러한 문제점을 해결하기 위해 IETF에서 제안한 가장 기본적인 해결 방안은 hierarchical mobile IPv6^[2]와 fast handover^[3]이다. 이 두 방안은 mipshop (MIPv6 signaling and handoff optimization) 이라는 별도의 작업 그룹에서 활발히 논의되고 있는 분야이다. 기존의 Mobile IPv6에서는 이동 단말이 위치 변경할 때마다 자신으로부터 먼 곳에 위치할 가능성이 많은 HA에 등록을 수행하여야 함으로써 긴 핸드오버 시간을 유발한다. Hierarchical mobile IPv6(HMIPv6)의 경우 Mobile Anchor Point(MAP)이라는 지역적 이동성 관리 노드를 두며 각 이동단말에 대해서 Regional Care-of-Address(RCoA)와 On-link Care-of-Address(LCoA) 두개의 CoA를 관리한다. 이동단말의 RCoA는 MAP의 서브넷을 따르며 LCoA는 라우터의 서브넷을 따른다. 외부 Corresponding Node(CN)이 이동단말과 통신을 하기 위해서 RCoA를 사용해서 데이터를 전송하면 MAP은 LCoA로 캡슐화하여 이동단말에게 전송하며, 내부 CN이 이동단말과 통신하기 위해서 LCoA를 사용하여 직접 전송하게 된다. 따라서 이동단말이 동일 MAP내에서 라우터간을 이동할 경우에 MAP에만 새로운 LCoA를 등록함으로써 이동 등록에 다른 지연 시간 및 자원 소비를 줄이는 장점을 가진다. 그러나 HMIPv6를 사용하더라도 네트워크 액세스 망의 이동이 일어난 뒤 이를 등록하는데 링크 계층의 이동성 처리 및 임시 주소 할당에 많은 시간이 소요된다는 것을 알 수 있고, 또한 이런 핸드오버 지연 시간은 불가피한 것이라 이 시간 동안 유실되는 데이터가 존재하며 이로 인해 무선 TCP의 성능을 저하시키는 결과를 가져온다.

이러한 문제점을 해결하기 위해서 fast 핸드오버(FMIPv6) 기술이 제안되었다. FMIPv6는 네트워크 2계층에서 핸드오버에 대한 트리거(trigger) 신호를 받게 되면 이를 이용하여 미리 새로운 서브넷에서 사용될

care-of-address(CoA)를 구성한다. 따라서 mobile IPv6에서 2계층 핸드오버가 끝난 이후에 새로운 CoA를 구성하여 바인딩 갱신을 함으로써 생기는 핸드오버에서의 지연을 줄일 수 있다. 또한 새로운 CoA를 구성하여 바인딩 갱신이 완료되기 이전까지 데이터 손실을 막기 위하여 이전 라우터는 새로운 라우터와 임시 터널을 생성하게 되고 이를 통해 Previous Access Router(PAR)로 오는 데이터를 New Access Router(NAR)로 전달하게 된다. 이렇게 할 경우 PAR을 떠난 이동 가입자에게 오는 데이터가 유실되지 않게 되어 성능 향상을 가져올 수 있다. 그러나 터널링을 통해서 패킷 순서가 뒤바뀌는 문제가 발생하며 이로 인해 네트워크의 상태와 관련 없이 TCP 성능이 저하되는 원인이 된다.

논문에서는 라우터에 간단한 snoop 프로토콜을 탑재한 알고리즘을 추가하거나 PAR에서 NAR을 통해 전송되는 패킷이 모두 전송될 때까지 CN에서 NAR로 직접 전송된 패킷을 일정시간 버퍼링한 후 전송함으로써 fast 핸드오버에서 발생할 수 있는 데이터 세그먼트의 순서 어긋남 현상을 방지할 수 있는 프로토콜을 제안한다. 이는 이동단말이 다른 액세스 망으로 이동할 경우 새로운 라우터는 데이터 순서 어긋남이 발생할 수 있는 시간을 산출하게 되고 이 시간을 이용함으로써 TCP 성능 향상 시킬 수 있는 알고리즘이다.

본 논문의 구성은 다음과 같다. 제 II장에서는 fast 핸드오버에서의 문제점과 snoop 알고리즘 및 기존의 재정렬 알고리즘에 대해서 알아보고, 제 III장에서는 제안된 변형 snoop 프로토콜과 타이머를 이용한 재정렬 알고리즘에 대해서 설명한다. 제 IV장에서는 시뮬레이션을 통해 분석하였으며 제 V장에서 결론을 맺는다.

II. 기존 관련 연구

본 장에서는 기존의 fast 핸드오버 알고리즘과 문제점을 알아보고, 무선망에서의 TCP 성능을 높이기 위해 제안된 snoop 프로토콜과 fast 핸드오버에서 발생하는 패킷 순서의 뒤바뀐 문제를 해결하고자 제안된 Delay Duplicated Ack(DDA)^[4] 프로토콜에 대해서 알아본다.

1. Fast 핸드오버 알고리즘 및 문제점

Fast 핸드오버 알고리즘은 L2 트리거를 이용하여 Mobile Node(MN)는 핸드오버가 이루어질 것을 미리 감지할 수 있다. 이로 인해 MN은 새롭게 연결될 NAR에서 사용될 mobile IP를 미리 얻고 핸드오버 시에 받

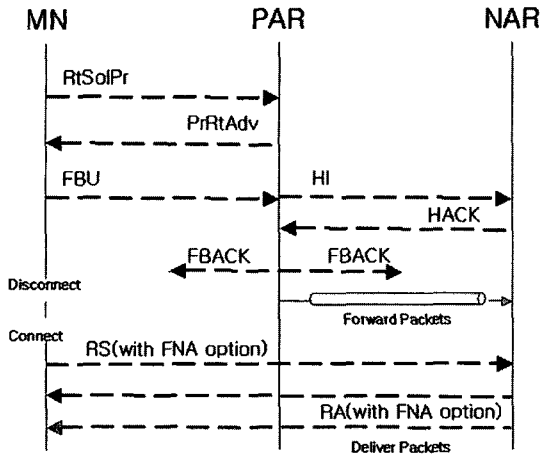


그림 1. FHMIPv6에서 핸드오버 절차
 Fig. 1. Handover procedure in FHMIPv6.

아야 할 데이터를 PAR에서 NAR로 터널링을 통해 미리 전송해 줌으로써 핸드오버 시간을 줄일 수 있고, 패킷 손실을 줄일 수 있다.

Fast 핸드오버의 처리 절차는 그림 1과 같다. L2 트리거가 발생하면 MN은 PAR에게 Router Solicitation for Proxy(RtSolPr) 메시지를 보낸다. PAR은 RtSolPr에 대한 응답으로 새로운 Care of Address(CoA) 또는 새로운 CoA를 위한 network prefix가 포함된 Proxy Router Advertisement(PrRtAdv) 메시지를 보낸다. PrRtAdv를 받은 MN은 CoA를 생성하여 Fast Binding Update(F-BU) 메시지를 보낸다. F-BU를 받은 PAR은 MN의 PCoA, 링크계층 주소, 새로운 CoA에 대한 정보를 포함한 Handover Initiate(HI)메시지를 NAR에게 전송한다. 이후에 MN에게 전송되는 패킷은 PAR에 버퍼링된다. HI는 MN이 생성한 새로운 CoA가 유효한지 NAR에게 검증하는 절차이며, NAR과 PAR사이에 터널링이 형성하기 위한 메시지이다. NAR은 HI에 대한 확인으로 Handover ACK(HACK) 메시지를 PAR에게 보내면 PAR과 NAR간의 터널링이 형성된다. HACK을 수신한 PAR은 F-BU에 대해 응답으로 F-ACK메시지를 NAR과 MN에게 전송한다. 또한 PAR은 MN에게 전송하기 위해 버퍼링된 패킷은 터널링을 통해 NAR로 보내고 MN에게 전송되는 패킷이 수신되면 터널링을 통해 전송한다. NAR에 전송된 MN의 패킷은 MN이 NAR에 접속이 이루어질 때까지 버퍼링된다. L2 트리거를 마친 후 MN은 Fast Neighbor Advertisement(F-NA) option을 포함한 Router Solicitation(RS) 메시지를 NAR에게 보낸다. NAR은 이에 대한 응답으로 Neighbor Advertisement ACK(NAACK) option을 포함

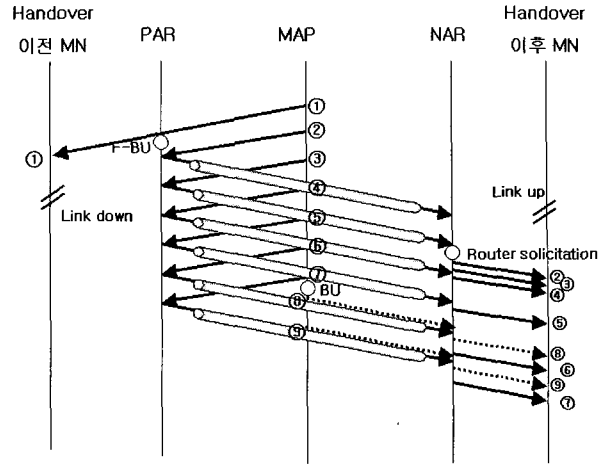


그림 2. FHMIPv6에서 패킷 뒤바뀔 문제
 Fig. 2. Packet reordering problem in FHMIPv6.

한 Router Advertisement (RA)을 보내며 NAR에서 버퍼링된 MN의 패킷을 전송한다.

FHMIPv6 환경에서 Fast 핸드오버에 의해서 발생하는 패킷 순서뒤바뀔 문제점은 그림 2와 같다. MN이 핸드오버 이전에 CN에서 전송되는 패킷은 1번 패킷과 같이 MAP을 통해 PAR을 경유하여 MN에게 전송된다. PAR이 MN으로부터 F-BU를 수신한 시점부터 MAP이 MN으로부터 핸드오버 이후에 BU를 수신하기까지 MAP에 전송된 패킷은 PAR을 경유하여 터널링을 통해 NAR에게 전송된다. NAR은 MN에게 전송되는 패킷을 버퍼링 한 후에 MN이 정상적인 핸드오버 이후에 RS 메시지를 송신하게 되면 저장된 패킷을 MN에게 전송하게 된다. 2번부터 7번까지의 패킷은 MAP이 BU를 수신하기 이전에 MAP에 도착한 패킷으로 PAR을 거쳐 NAR로 터널링을 통해 전송된 패킷을 나타낸다. MAP이 지역 BU를 수신하게 되면 그 이후에 도착한 패킷 8, 9번은 NAR로 직접 전송하게 된다. 따라서 터널링에 의한 패킷 지연시간으로 인해 MN에서 수신되는 패킷의 순서가 2-3-4-5-8-6-9-7로 뒤바뀌어 전송되는 것을 알 수 있다. MN의 TCP 계층은 순서가 어긋난 데이터를 수신하게 되고 이로 인해 중복 ACK(duplicated acknowledgement)을 전송하게 된다. 중복 ACK을 수신한 CN의 TCP 계층은 데이터가 유실된 것으로 판단하고 송신 윈도우를 줄이게 되어 망의 성능을 저하 된다.

2. Snoop 알고리즘

Snoop 프로토콜은 무선 망에 TCP가 적용될 경우 무선 망에서의 페이딩 등으로 인한 데이터 유실을 송신자

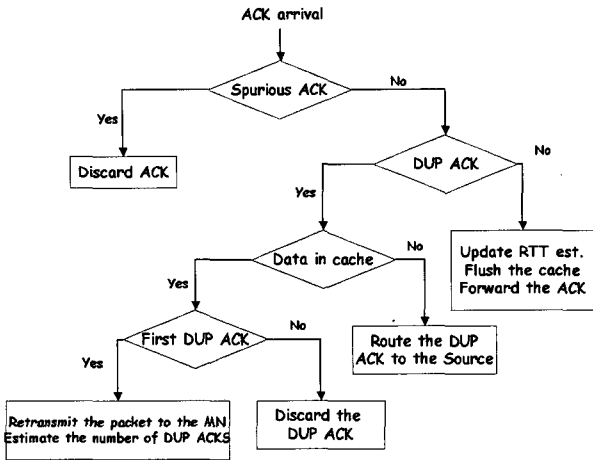


그림 3. Snoop 에이전트에서 snoop_ack() 알고리즘
Fig. 3. Snoop_ack() algorithm in snoop agent.

가 혼잡으로 판단하여 혼잡 제어 절차 수행하는 것을 방지하기 위해 무선 망과 통신을 담당하는 곳에 snoop 에이전트를 추가한다. Snoop 에이전트는 고정된 단말과 이동 단말 사이에 삽입되어 트래픽을 모니터하고 패킷의 재전송과 중복 ACK을 무시하는 역할을 한다. Snoop은 종단간의 데이터 전송을 유지하는 방법이며, 고정 단말에서 이동 단말로의 전송시에는 각 종단 단말에서의 변화가 필요 없고 이동 단말에서 고정 단말로 전송시에는 이동 단말에서 최소한의 변화가 요구된다. 모든 TCP 세그먼트는 무선 디바이스로부터 ACK이 올 때까지 기지국에 저장되어 있기 때문에 무선 링크에서 손실된 TCP 세그먼트는 기지국에 의해 재전송 될 수 있다. 또한, 중복 ACK은 송신측에 전달되어 불필요하게 혼잡제어 알고리즘을 실행하지 않도록 하기 위해 기지국에서 억제시킨다.

Snoop의 처리절차는 크게 Snoop_data()와 Snoop_ack() 두 가지로 나눌 수 있다. Snoop_data() 처리절차는 BS가 고정 단말로부터 수신한 데이터 패킷을 저장하는 절차이다. 저장된 패킷은 수신단으로부터 ACK이 수신되면 지운다. Snoop_data()에는 세 가지 경우가 있다. 순서대로 들어오는 경우와 패킷이 순차적이지 않고 패킷을 저장하는 경우 그리고 패킷이 순차적이지 않고 패킷을 저장하지 않는 경우이다. 그림 3은 snoop 에이전트가 ACK 세그먼트를 수신한 경우에 수행하는 snoop_ack() 알고리즘을 나타낸다.

3. 기존 재정렬 알고리즘

기존의 타이머를 이용하여 Fast 핸드오버에서 발생하는 패킷 순서의 뒤바뀜 문제를 해결하고자 Delay

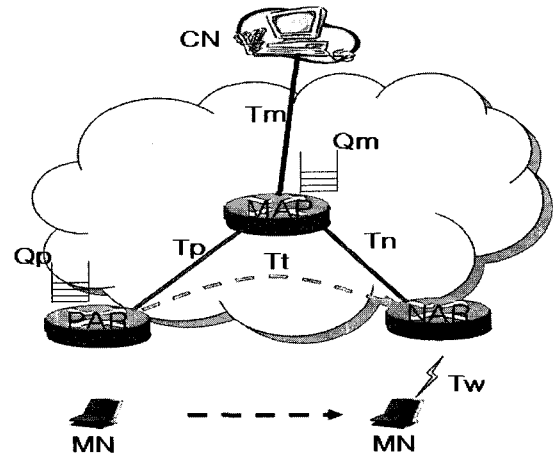


그림 4. FHMPv6에서 패킷 지연 시간
Fig. 4. Packets delay time in FHMPv6.

Duplicated Ack(DDA)^{[4][5]}가 제안되었다. DDA는 이동 가입자의 TCP 계층과 IP 계층 사이에 별도의 계층을 두어 unstable time period(UTP) 시간동안 송신자에게 ACK을 보내지 않고 저장한 후에 타이머가 끝나면 저장된 ACK을 정리하여 보내는 알고리즘이다. UTP는 패킷의 세그먼트가 어긋나게 도착할 수 있는 시간간격을 나타내며, 이 기간동안 MN은 기존의 경로를 통해 PAR에서 NAR까지 터널링된 패킷과 새로운 경로를 통해 CN에서 바로 전달된 패킷을 동시에 수신하게 된다. 따라서 UTP 동안 ACK을 보내지 않고 정리한 후에 보내게 되면 중복 ACK을 피할 수 있다.

그림 4는 FHMPv6에서 CN에서 전송된 패킷이 MN에 전송되기까지 패킷 지연시간을 나타낸다. CN에서 MAP까지의 패킷 지연 시간을 T_m , MAP에서 PAR과 NAR까지의 패킷 지연시간을 각각 T_p, T_n 이라고 하고, PAR에서 NAR까지 터널링을 통해 전달되는 지연 시간을 T_t 라 한다. 또한 MAP과 PAR에서 각각의 큐잉 지연시간을 Q_m 과 Q_p 라 하고 무선 환경에서 단말기와 라우터간의 패킷 지연시간을 동일한 T_w 라 가정하였을 때 UTP는 식 (1)과 같이 계산할 수 있다.

$$UTP = \max(0, (T_p, T_n) + Q_p + T_t) \quad (1)$$

DDA의 경우 이동단말에서 여러 CN과의 모든 연결에 대해서 UTP를 계산하는 알고리즘은 이동 가입자 단말의 프로세싱에 대한 오버헤드를 가져오며, 복잡도를 높이고 기존 단말이 모두 업그레이드되어야 하는 어려운 단점이 있다.

III. TCP 성능 향상을 위한 재정렬 알고리즘

Mobile IPv6에서 MN이 이동시 핸드오버에 의해서 패킷이 손실되는 단점을 보완하기 위해서 fast 핸드오버 알고리즘이 제안되었으며 패킷 손실을 줄임으로써 TCP 성능을 향상시킬 수 있었다. 하지만 TCP 성능에 영향을 주는 요소로 패킷 손실뿐만 아니라 패킷 지연과 수신자에서 수신된 패킷의 순서도 중요하다. 본 장에서 fast 핸드오버 알고리즘에서 발생하는 TCP 재정렬 문제를 해결하기 위해서 홀딩 타이머를 이용한 알고리즘과 변형된 snoop 알고리즘을 제안한다.

1. 홀딩 타이머를 이용한 알고리즘

DDA 프로토콜의 경우 모든 이동 가입자의 단말기가 upgrade 되어야 하며 단말기의 처리가 복잡한 단점을 가지게 된다. 본 논문에서는 NAR에서 UTP 시간 동안 새로운 경로에 의해 CN에서 직접 수신되는 패킷을 저장시키는 알고리즘을 제안한다. 자세한 알고리즘은 다음과 같다.

Fast 핸드오버시 MN가 NAR에 접속하고 RS메시지를 송신하면 NAR에서는 이동 MN의 PAR에서 터널링 되어 저장된 패킷을 MN에게 전송하게 된다. 이후에 PAR에서 캡슐화를 통해 NAR에 전송된 패킷은 분해해가 되며 저장되지 않고 MN에게 바로 전송하게 된다. 이러한 터널링은 CN이 MN에게 BU메시지를 수신하기 전까지 계속되며 BU가 수신되면 CN은 NAR에게 바로 패킷을 전송하게 된다. NAR은 MN가 자신에게 접속한 후에 CN에서 직접 전송된 첫번째 패킷을 수신하게 되면 타이머를 UTP 동안 설정한다. 이 기간에 CN에서 직접 전송되는 패킷은 저장되며 PAR에서 NAR로 터널링 되어 전송되는 패킷은 MN에게 전송되게 된다. 타이머가 종료되면 저장된 패킷을 이동 MN에게 전송함으로써 패킷 순서 뒤바뀔 문제점을 해결 할 수 있다.

NAR에서 UTP 시간은 관리자에 의한 설정과 PAR과 NAR간의 컨트롤 메시지(HI, HACK, FBACK) 전송 시 메시지에 time stamp option 필드를 사용하여 계산할 수 있다.

2. 변형된 snoop 알고리즘

기존 snoop 프로토콜의 가장 중요한 알고리즘은 데이터 세그먼트가 수신되었을 때 처리하는 snoop_data()와 단말로부터 ACK이 수신되었을 때 처리하는

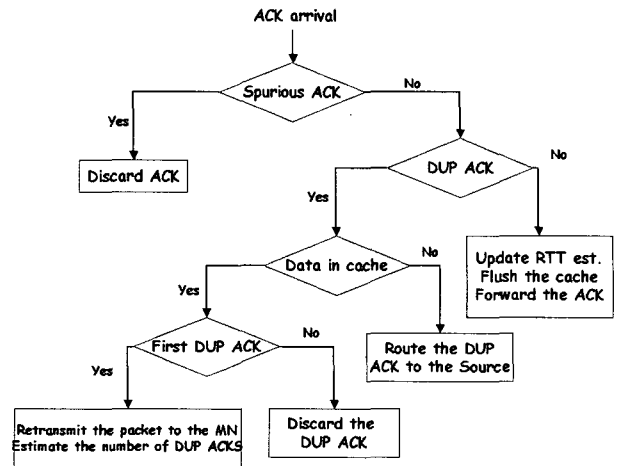


그림 5. Snoop 에이전트에서 snoop_ack_new() 알고리즘
Fig. 5. Snoop_ack_new() algorithm in snoop agent.

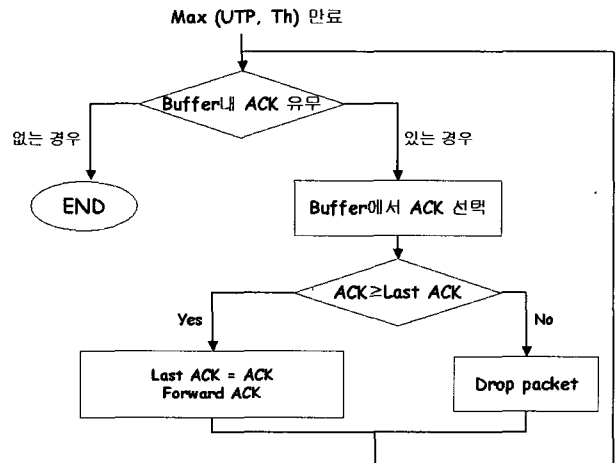


그림 6. 타이머 만료 시 snoop 동작 절차
Fig. 6. Snoop operation procedure at timer expire.

snoop_ack() 루틴이다. 제안하는 방안은 snoop_data() 알고리즘은 변경하지 않고 snoop_ack() 알고리즘만을 수정하도록 한다. 향후 논문에서 수정한 알고리즘을 snoop_ack_new()라고 부르기로 하며 전체적인 ACK 처리 알고리즘은 그림 5와 같다.

이동단말이 핸드오버를 수행할 때 PAR은 snoop 프로토콜 관련 상태정보를 NAR에게 전송하게 된다. 이를 수신한 NAR은 이동 단말에 대한 새로운 snoop 상태 정보를 생성하게 되고 이동 가입자가 자신의 망으로 접속하게 되면 데이터 세그먼트 순서가 바뀔 수 있는 일정 시간 동안 ACK 세그먼트 처리를 유보하는 snoop_ack_new() 알고리즘을 수행한다. 이 때 ACK 세그먼트 처리가 유보되는 시간을 T_a 라고 하면 이 시간은 식 (2) 와 같다.

$$T_a = \max(T_h, UTP) \tag{2}$$

위의 식에서 T_h 는 PAR에서 snoop 정보를 NAR로 전송할 때까지의 지연 시간이고 UTP 는 데이터 세그먼트가 순서에 어긋나게 도착할 가능성이 있는 기간이다.

따라서 이 시간 동안 순서가 어긋나서 올 수 있는 ACK를 처리하지 않고 있다가 이 시간 후에 snoop 계층에서 그 동안 받은 ACK를 모두 고려하여 송신 단말로 ACK를 보내면 이동으로 인한 중복 ACK 문제를 해결할 수 있다. NAR이 UTP 와 T_h 시간 중 최대 시간을 기다린 후 타이머가 만료되면 저장하고 있던 ACK 패킷을 순서에 맞게 정리하여 송신자에게 보내게 되며 이에 대한 상세 알고리즘은 그림 6과 같다.

IV. 시뮬레이션 및 성능 평가

본 장에서는 기존의 FHMPv6와 재정렬 알고리즘을 적용한 FHMPv6에 대해서 network simulator(ns-2)^[6]를 통해 비교 분석하였다. 시뮬레이션을 위해 ns-2.1b6을 확장한 모토롤라의 MobiWan^[7]을 이용해서 구현하였으며, Ns-2.1b6 에서의 TCP 관련된 모듈이 일부 구현되어 있지 않은 TCP 모듈을 ns-2.27 버전으로 업데이트하여 시뮬레이션을 수행하였다.

첫 번째로 변형된 snoop 알고리즘을 적용하였을 경우와 기존의 FHMPv6에 대한 시뮬레이션을 수행하였다. 시뮬레이션 파라미터로 PAR, NAR에서 MAP까지의 지연시간을 30ms, MAP에서 CN까지의 지연시간을 200ms, MN의 이동 속도를 50m/s로 설정하였다. TCP reno를 사용하였으며 FTP 트래픽을 CN에서 MN으로 데이터를 전송하였다.

그림 7은 기존의 FHMPv6에서 각 패킷에 대해서 CN에서 전송한 시간과 MN에서 수신한 시간을 나타내며, 핸드오버 시간을 확대한 그림이다. 그림 7에서 AR간 핸드오버시에 패킷 손실은 발생하지 않았지만 PAR을 거쳐 NAR에게 전송되는 패킷과 직접 NAR로 전송되는 패킷에 의해 패킷의 순서 어긋남이 발생하는 것을 볼 수 있다. 이러한 순서 어긋남으로 인해 중복 ACK이 발생하며 송신자는 윈도우를 줄이고 패킷 재전송을 하므로 TCP성능이 저하된다. 그림 9는 TCP CWND를 나타낸 그래프로써 기존의 FHMPv6의 경우 AR간 핸드오버시에 중복 ACK으로 인해 송신측 CNWD의 값이 반으로 줄어드는 것을 볼 수 있다.

그림 8은 본 논문에서 제안한 snoop를 적용한

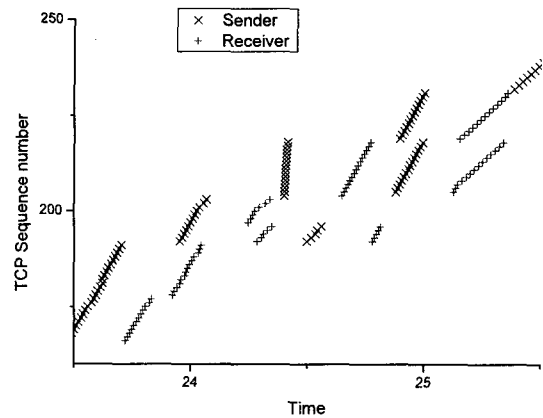


그림 7. FHMPv6에서 TCP sequence number
Fig. 7. TCP sequence number in FHMPv6.

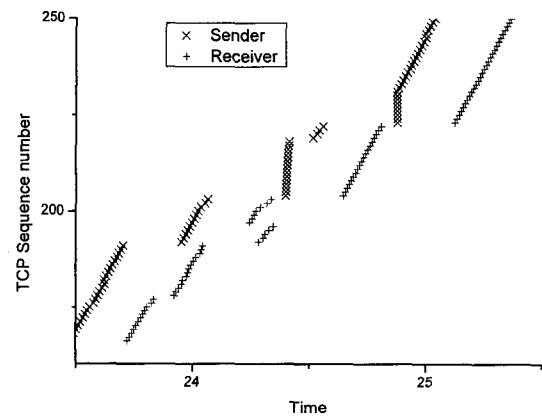


그림 8. 수정된 snoop 재정렬 알고리즘을 사용한 FHMPv6에서의 TCP sequence number
Fig. 8. TCP sequence number in FHMPv6 with modified snoop resequencing algorithm.

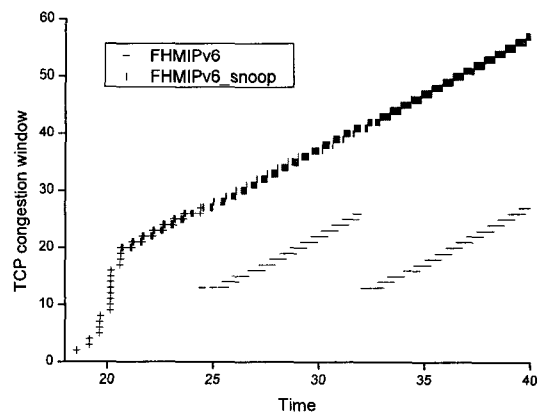


그림 9. TCP 혼잡 윈도우
Fig. 9. TCP congestion windows.

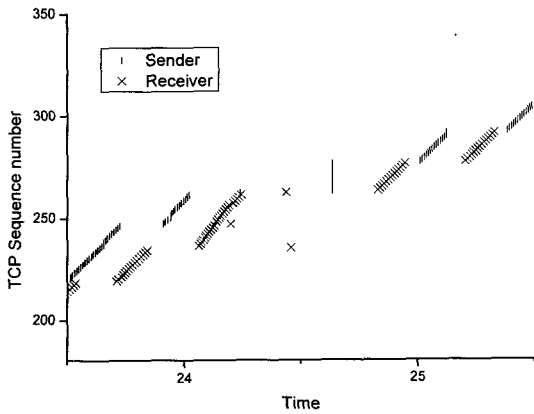


그림 10. FHMIPv6에서 TCP sequence number
Fig. 10. TCP sequence number in FHMIPv6.

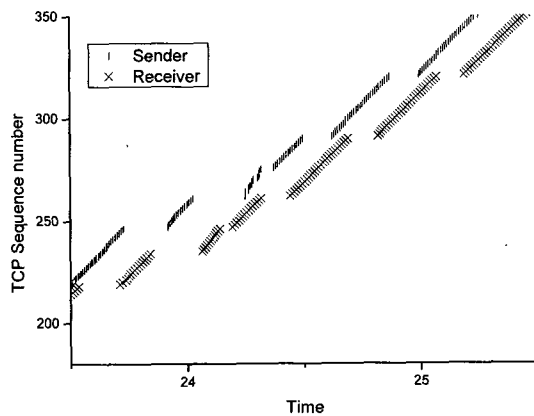


그림 11. Holding timer 재정렬 알고리즘을 적용한 FHMIPv6에서 TCP sequence number
Fig. 11. TCP sequence number in FHMIPv6 with holding timer resequencing algorithm.

FHMIPv6 환경에서의 그림을 보여준다. 기존의 FHMIPv6와 동일하게 수신측에서는 패킷 순서가 뒤바뀌어 전송되는 것을 볼 수 있다. 하지만 그림 9에서 TCP CWND의 값은 반으로 감소하지 않고 계속 증가하는 것을 볼 수 있다. 이는 snoop 에이전트 기능을 수행하는 NAR이 논문에서 제안한 변형된 snoop 알고리즘을 적용함으로써 새로운 MN의 핸드오버시 *UTP* 시간동안 뒤바뀌는 TCP ACK 메시지를 그림 5와 같은 알고리즘을 적용하여 처리하기 때문이다. 따라서 MN에서는 중복 ACK을 송신하지만 CN에서의 중복 ACK이 발생하지 않음으로서 TCP 성능이 향상되는 것을 볼 수 있다.

두 번째로 기존 FHMIPv6와 홀딩 타이머를 이용한 알고리즘에 대한 시뮬레이션을 수행하였다. 시뮬레이션

파라미터로 PAR, NAR에서 MAP 까지의 지연시간을 20ms, MAP에서 CN까지의 지연시간을 160ms, MN의 이동 속도를 50m/s로 설정하였다. 트래픽은 FTP를 이용하여 CN 에서 MN으로 데이터를 전송하였다.

그림 10은 기존의 FHMIPv6에 대한 그래프로써 MN가 핸드오버 시에 패킷의 순서가 뒤바뀌어 전송되는 것을 볼 수 있고, 이로 인해 송신자의 CWND의 값이 반으로 줄어드는 것을 볼 수 있다.

그림 11은 본 논문에서 제안한 NAR에서 타이머를 이용한 FHMIPv6 환경에서의 시뮬레이션 결과 그래프를 나타낸다. MN가 NAR로 핸드오버 이후 CN에서 MN에게 직접 전송되는 최초의 패킷이 NAR에 도착하면 *UTP* 시간동안 저장한 후에 보내게 된다. 따라서 새로운 경로가 설정된 후 *UTP* 시간 안에 도착한 패킷에 대해서 최대 *UTP* 시간만큼 지연은 발생하지만 패킷의 순서가 뒤바뀜 없이 정상적으로 도착하는 것을 확인할 수 있다. 결과적으로 핸드오버 이후 CWND가 반으로 감소하지 않고 계속적으로 증가하여 TCP 성능이 향상되는 것을 시뮬레이션을 통해 확인할 수 있었다.

V. 결 론

Fast 핸드오버 알고리즘은 이동 단말이 이동 간에 통신을 할 경우 발생하는 패킷손실의 단점을 보완하기 위해서 제안되었고 패킷 손실을 많이 줄일 수 있음을 시뮬레이션을 통해 확인할 수 있었다. 또한 fast 핸드오버에서 패킷 손실이 줄어든 이유는 PAR에서 NAR로 패킷을 터널링을 통해서 전송해 주기 때문이며, 이로 인해 BU 메시지가 수신되는 시점에서 패킷의 순서가 뒤바뀌게 되는 것을 시뮬레이션을 통해 확인할 수 있었다. 결과적으로 패킷 손실에 의해서 TCP 성능이 크게 떨어지는 것에 비해서 향상되는 결과를 가져올 수 있지만 패킷 순서가 바뀌는 문제로 인해서 TCP 성능이 저하되는 결과를 가져오게 된다.

본 논문에서는 snoop을 변형한 알고리즘과 홀딩 타이머를 이용하는 알고리즘을 기존의 FHMIPv6에 적용하여 핸드오버 시에 패킷 순서 뒤바뀌는 문제를 해결함으로써 정상적인 네트워크 상황에서 TCP 성능이 저하되는 문제점을 해결하였다. 또한 제안된 알고리즘이 기존 알고리즘에 비해 TCP의 성능향상이 되었음을 시뮬레이션을 통해 확인할 수 있었다.

참 고 문 헌

[1] C. Perkins, "IP mobility support in IPv4", draft-ietf-moblieip-ipv12.txt, Internet Engineering Task Force, Mar. 2002.
 [2] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical mobile IPv6 mobility management (HMIPv6)", draft-ietf-mipshop-hmipv6-04.txt, 2004. 12.
 [3] R. Koodli (Ed), "Fast handovers for mobile IPv6", draft-ietf-mipshop-fast-mipv6-03.txt, 2004, 10.
 [4] Dongwook Lee, Jong Won Kim, "Delayed-duplicated ACK(DDA) algorithm for TCP performance enhancement to overcome packet

sequence disruption in fast-handoff of mobile IPv6", K-JIST, 24th International Conference on Distributed Computing Systems Workshops - W1: MNSA, (ICDCSW'04) Hachioji, Tokyo, Japan, pp. 98-103, March 23 - 24, 2004.
 [5] D. Tandjaoui, N. Badache, H. Bettahar, A. Bouabdallah and H. Seba, "Performance enhancement of smooth handoff in mobile IP by reducing packet disorder", Pro. of the 8th IEEE International Symposium on Computers and Communications(ISCC '03), 2003.
 [6] ns-2, <http://www.isi.edu/nsnam/ns/>
 [7] MobiWan, <http://www.inrialpes.fr/planete/mobiwan/>

저 자 소 개



황 안 규(학생회원)
 1999년 한남대학교 물리학과 학사
 2004년 충남대학교 정보통신공학과 석사
 2005년~현재 충남대학교 정보통신공학과 박사과정
 <주관심분야 : 이동통신 네트워크, 데이터 통신, 초고속 통신>



이 재 용(중신회원)-교신저자
 1988년 서울대학교 전자공학과 학사
 1990년 한국과학기술원 전기 및 전자공학과 석사
 1995년 한국과학기술원 전기 및 전자공학과 박사
 1990년~1995년 디지콤 정보통신연구소 선임연구원
 1995년~현재 충남대학교 정보통신공학부 부교수
 <주관심분야 : 초고속통신, 인터넷, 네트워크 성능분석>



김 병 철(중신회원)
 1988년 서울대학교 전자공학과 학사
 1990년 한국과학기술원 전기 및 전자공학과 석사
 1996년 한국과학기술원 전기 및 전자공학과 박사
 1993년~1999년 삼성전자 CDMA 개발팀
 1999년~현재 충남대학교 정보통신공학부 부교수
 <주관심 분야 : 이동인터넷, 이동통신 네트워크, 데이터통신>



이 재 훈(중신회원)
 1985년 한양대학교 전자공학과 학사
 1987년 한국과학기술원 전기 및 전자공학과 석사
 1995년 한국과학기술원 전기 및 전자공학과 박사
 1987년 3월~1990년 4월 데이콤 연구원
 1990년 9월~1999년 2월 삼성전자 정보통신부문 선임 연구원
 2000년 3월~2000년 12월 삼성전자 자문교수
 2000년 5월~현재 한국이더넷포럼 운영위원
 1999년 3월~현재 동국대학교 정보통신공학과 부교수
 <주관심분야 : 초고속통신, 다중 액세스 프로토콜, 인터넷 프로토콜, 메트로 이더넷>