

버티컬핸드오버에서 TCP Flow의 성능향상을 위한 연구동향

추현동, 석승준

경남대학교 컴퓨터 공학과

e-mail: {bogushd, sjseok}@net.kyungnam.ac.kr

A Survey on Improvement TCP Flow Performance for Vertical Handover

Hyun-Dong Chu, Seung-Joon Seok
Kyungnam University

요 약

최근에 다양한 네트워크의 기술들이 존재하고 개발되어지고 있다. 이러한 네트워크 기술들을 효과적으로 사용하기 위해서 버티컬 핸드오버가 다양한 측면에서 연구되어지고 있다. 버티컬 핸드오버는 단말기가 이종망에서 더 나은 성능을 가진 네트워크로 끊임없는 서비스를 제공하는 핸드오버 기술이다. 버티컬 핸드오버를 수행하는 동안에 라우터간 시그널링의 지연과 더불어 인증과 등록절차에 대한 시간지연으로 인하여 혼잡한 상황이 아님에도 불구하고 TCP에서는 혼잡한 상황으로 인식하여 성능이 낮아지는 문제가 발생한다. 또한, 버티컬 핸드오버를 수행하는 중에 링크의 특성이 다르고 대역폭이 달라서 패킷의 순서번호가 일치하지 않아서 잘못된 빠른 재전송(false fast retransmission:FFR)발생하고 이러한 현상들은 TCP 처리량의 저하를 가져온다. 논문에서는 이러한 문제를 해결하기 위한 방법들에 대한 동향을 조사해보았다.

키워드 : Vertical Handover, TCP, Reordering, false fast retransmission

1. 서론

최근에 무선 인터넷 서비스의 기술이 발전됨에 따라 심리스(Seamless)한 서비스를 제공하기 위한 연구가 활발히 진행 중이다. 모바일 노드(Mobile Node)가 네트워크간 이동을 하면서 심리스한 통신을 하는것을 핸드오버(Handover)라고 정의한다. 또한 동일한 네트워크내에서 모바일 노드의 이동을 수평적 핸드오버(Horizontal Handover)라고 하며, 서로다른 이종망의 네트워크들을 이동하면서 심리스한 통신을 하는것을 버티컬 핸드오버(Vertical Handover)라고 한다. 수평적 핸드오버의 경우 동일한 네트워크의 이동성을 지원하기 때문에 물리 매체에 대한 고려를 하지 않고 그림 1.에서와 같이 RSS(Received Signal Strength)의 크기만으로

거리를 식별할 수 있다. 따라서 이동에 따른 핸드오버를 비교적 쉽게 할 수 있다. 하지만 버티컬 핸드오버는 이종간의 핸드오버를 수행을 하는것 이므로 각각 물리적인 특성이 다르다. TCP는 유선통신에 맞게 설계되었기 때문에 무선환경에서의 에러상황은 혼잡에 의한 상황으로 가정을 한다. 따라서 에러 혹은 ACK를 송신측에서 수신하지 못한다면 곧 혼잡한 상황으로 인식하여 처리율의 감소로 이어진다.

또한 버티컬 핸드오버에서의 핸드오버에 따른 대역폭의 급격한 변화와 RTT(Round Trip Time)시간의 변화에 따른 유동적인 상황을 수용하지 못하는 문제점과 핸드오버에서 망의 변경을 하는중에 인증과 권한부여에 따른 지연시간과 핸드오버 시그널링에 따른 지연에 의해서 재전송 타이머의 동작으로 혼잡하지 않는데 혼잡한 상황으로 인식되어지는 문제가 존재한다. 그래서 최근 연구에서 버티컬 핸드오버를 할 때 TCP

의 처리량을 높이기 위해 많은 연구가 진행되어지고 있다.

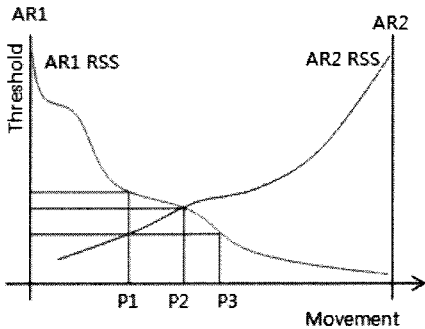


그림 1. 이동에 따른 RSS의 변화

II. 시스템 모델

기존의 연구들에서 Mobile IPv6를 이용하여 버티컬 핸드오버를 수행했을 경우 매번 HA(Home Agent)에게 BU(Binding Update)를 전송하게 된다. 모바일 노드와 라우터, 라우터와 HA간에 CoA(Care-of-Address)를 주고 받고, 관리를 해줘야 하고 인증절차도 있어야 함으로 핸드오버 지연시간은 발생요소는 더욱 많아지게 된다. 따라서 초기화 시간과 처리시간 등, 핸드오버의 지연시간 길어져서 전송량의 감소를 가져오게 된다.

이러한 지연시간을 감소시키기 위해 Hierarchical MIPv6가 제안되었다. 핸드오버를 할 때, 중간에 앵커 액세스라우터(AnchorAR)를 삽입하여 일정지역 내에서의 핸드오버를 수행한다면, 모바일 노드가 HA와 직접적인 핸드오버를 수행하지 않고 AnchorAR와 핸드오버의 절차만 거쳐서 핸드오버 지연시간이 줄어들게 된다. 그럼에도 불구하고 핸드오버의 지연 크게 나아지지 않았다.

이 문제를 해결하기위해 3계층의 핸드오버가 수행되기 이전에 2계층의 핸드오버를 우선 수행하는 Fast MIPv6가 제안 되었으며 2계층의 트리가 발생하게 되면 미리 RtSolPr 메시지를 PAR(Previous Access Router)에게 전송을 해서 인접한 라우터들의 정보를 요청하게 되고 PAR은 인접한 라우터의 IP 주소, 2계층의 주소와 서브넷 Prefix정보를 PrRtAck메시지와 함께 모바일 노드에게 전송한다. 모바일 노드는 PrRtAdv정보를 바탕으로 NAR(New Access Router)에서 사용할 NewCoA를 생성한다. 모바일 노드는 NewCoA가 맞는지 확인하기 위해서 Fast-Binding Update(F-BU)를 PAR에게 전송하고 새로운 라우터인 NAR에게 Handover Initiation(HI)신호를 전송하게 된다. NAR은 PAR로부터 수신받은 NewCoA가 유효한지 확인하게 되며 만일 잘못되었다더라도 HAck메시지에 NewCoA를 삽입하여 전송하게 된다. PAR이 HAck를 수신 받는다면 핸드오버의 2계층의 핸드오버

절차는 완료가 되어진 것으로 판단하고 Fast Binding Acknowledgement를 NAR과 모바일 노드에게 전송한다. 이때 모바일 노드는 PAR과는 Disconnect상태가 되고 CN으로 받은 데이터를 모바일 노드에게 전달하기위해 NAR과 터널링을 한다. 이때의 NAR과 모바일 노드는 Connect된 상태가 된다. 모바일 노드가 CN에게 Fast-Neighborhood Advertisement(F-NA)를 전송하면 수신받은 CN은 이 때 이후로 모바일 노드에게 오는 패킷을 NAR을 통해 모바일 노드에게 전달하게 된다. 마지막으로 PAR으로 전달되던 패킷들을 손실 시키지 않기 위해서 버퍼링후 NAR에게 전달하게 되고 전달받은 패킷을 NAR은 모바일 노드에게 전달하게 된다.

III. TCP Flow 성능 향상 방법

버티컬 핸드오버를 수행할 때 TCP성능에 영향을 미치는 요소들은 첫째로 패킷의 손실을 들 수가 있다.

핸드오버를 수행하기에 앞서서 인증과 등록 과정등의 복잡한 절차를 거쳐야 하는 상황에 대한 핸드오버 지연이 발생하게 되고 이는 송신측에서 타임아웃(Timeout)시간내에 도착하지 않는다면 TCP의 특성상 혼잡한 상황으로 인식하여 처리량을 반으로 떨어뜨리게 된다. 하지만 이러한 상황은 혼잡에 의한 상황이 아니므로 네트워크를 효과적으로 사용하지 못하는 결과를 초래한다.

두 번째로는 갑작스러운 RTT의 변화에 따른 수신측으로부터의 Ack의 수신이 재전송 타이머가 경과된 이후에 도착하는 문제가 발생한다.

3.1. 타임아웃을 줄이는 방법

타임아웃(TimeOut)은 핸드오버의 처리과정에서 발생하고 재전송으로 인하여 처리량이 감소하는 문제가 발생하는데 TCP 성능을 높이기 위해서 2가지의 구조적인 방법이 존재한다.

3.1.1. Hierarchical MIPv6

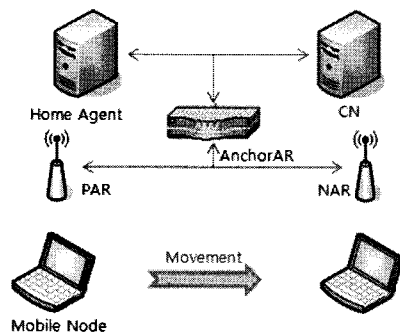


그림 2. Hierarchical MIPv6

그림 2.에서와 같이 버티컬 핸드오버의 HA와 모바일 노드간 바인딩 업데이트의 빈도수를 작게하여 지연을 줄이고 자 제안되어진 방법이다. 불필요하게 많은 핸드오버의 요청을 하기보다는 중간에 AnchorAR를 설치하여 구조적인 방법으로 핸드오버를 관리하는데 일정 지역내에서의 핸드오버에 대해서는 AnchorAR이 관리를 하게 되고 AnchorAR을 벗어나는 핸드오버에 대해서만 HA에게 핸드오버를 알리고 BU를 수행하는 방식으로 핸드오버의 지연이 대폭 줄어들고 재전송 타이머의 발생을 적게 한다.

3.1.2. Fast MIPv6

그림 3.에서와 같이 모바일 노드가 CN에게 Fast-Neighborhood Advertigement(F-NA)를 전송하면 수신 받은 CN은 이 때 이후로 모바일 노드에게 오는 패킷을 NAR을 통해 모바일 노드에게 전달하게 된다. 마지막으로 PAR으로 전달되던 패킷들을 손실 시키지 않기 위해서 버퍼링후 NAR에게 전달하게 되는데 이때 전달된 패킷은 이미 PAR을 거쳐서 들어오기 때문에 순서번호가 CN에서 NAR을 통해 들어온 패킷보다 낮다. 즉 터널링을 통한 지연이 발생한것을 의미하고 TCP계층순서 번호가 어긋난 데이터를 수신하게 된다.[1] 중복된 Ack(Duplicate Acknowledgement)를 전송하게 된다. 중복 Ack를 수신한 CN은 패킷이 유실 혹은 손실된 것으로 판별하고 다시 재전송을 수행한다. 이때 경쟁 윈도우의 크기를 줄이게 되고 이때 불필요한 혼잡제어로 인한 처리량의 감소가 생긴다.

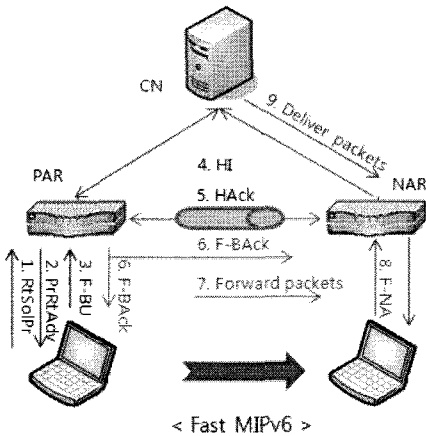


그림 3. Fast MIPv6 Signaling

3.2. Reordering문제

서로 다른 링크특성을 갖는 두 매체의 특성에 의해서 하나의 링크가 다른 링크보다 더욱 큰 대역폭을 가지고 있는 3G to WLAN과 같은 상황에서 3G로 전송되던 패킷이 WLAN으로 핸드오버될 때, 급격하게 증가된 대역폭으로 인해서 3G로 전송되던 패킷보다 WLAN으로 패킷이 먼저 수신측에 도

착을 함으로써 재전송 타이머가 동작하는 현상이 발생한다. 결과적으로 혼잡한 상황이 아닌데도 혼잡한 상황으로 인식하여 혼잡윈도우의 크기를 1로 줄이는 문제 발생(3), 중복Ack가 발생할 수 있다.

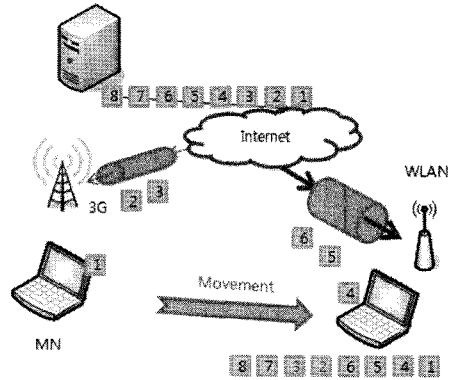


그림 4. 3G To WLAN 버티컬 핸드오버

3.2.1. 상·하향 버티컬 핸드오버

1. 네트워크간 전송속도비를 계산하는 방법[4]

버티컬 핸드오버를 수행하기전에 다른 망으로부터 수신신호가 들어온다면 핸드오버의 가능성이 존재 하는것을 의미하며 현재 물리적 링크의 대역폭과 핸드오버를 하고자하는 링크의 대역폭의 비율을 계산해서 상향인지 하향인지 를 결정할 수 있다.

$$RTT_{correction} = pkt\ size * (1/BW_{next} - 1/BW_{current}) + (Delay_{next} - Delay_{current}) \dots(1)$$

$RTT_{correction}$: RTT보정 값, pkt size : 패킷의 사이즈, BW_{next} : 핸드오버 할 다음 네트워크, $BW_{current}$: 현재 네트워크, $Delay_{next}$: 핸드오버 할 다음 네트워크의 지연시간, $Delay_{current}$: 현재 네트워크의 지연시간

$$RTT_{new} = RTT_{next} + 2(RTT_{correction}) \dots(2)$$

보정된 값을 상대노드에게 전달하면 모바일 노드는 식(1)을 식(2)에 대입하여 계산 후 새로운 망의 RTT값을 계산한다.

2. 홀딩 타이머를 사용하는 방법

Fast MIPv6의 환경에서 모바일 노드가 F-BU를 NAR에게 전달하면 NAR은 UTP(Unstable Time Period) 타이머를 Start하고 CN으로 F-BU를 전달한다. 이는 정상적인 버티컬 핸드오버의 수행이 완료된 상황이며 CN은 패킷을 NAR로 전송을 하게 된다. 이때 NAR에서는 TCP의 순서번

호의 중복을 막기위해 CN으로부터의 패킷을 저장하고 있다. CN으로부터 먼저 송신된 패킷은 PAR에 저장되어 있으며 NAR사이의 터널을 통해 먼저 모바일 노드에게 전달되고 UTP 타이머는 종료한다. 다음으로 NAR에 저장되어있는 패킷을 모바일 노드에게 전달함으로써 순서번호의 중복을 막아 Reordering문제를 해결할 수 있다.

3. Snoop알고리즘을 이용한 방법[5]

TCP가 적용된 네트워크에서 무선상의 페이딩 현상에 대한 데이터의 유실을 송신측은 네트워크가 혼잡한 상황으로 인식, 처리량을 낮추게 됨으로써 네트워크의 효율성을 떨어뜨린다. 이를 방지하기 위한 방법으로써 무선망과 통신하는 즉, NAR 부분에 삽입을 하여 감시를 수행하는 트래픽의 모니터 기능, 에러상황에 대한 패킷의 재전송, 혼잡상황이 아닌 것에 대해서 처리량을 줄이지 않는 기능, 패킷이 여러경로를 통해서 들어오거나 뒤늦게 들어오는 중복 Ack를 무시하는 역할을 담당한다.

3.2.2. 상향 버티컬 핸드오버

RTT가 짧은 WLAN망에서 RTT가 긴 3G망으로의 버티컬 핸드오버를 하는 동안 RTT의 차이로 인한 짧은 재전송 타이머로 여러번 재전송이 발생하는것을 방지하기 위함이다.

1. Probe Packet 전송방법

RTT를 측정하기 위해서 패킷을 전송할 때 추가적인 Probe Packet을 전송하여 새로운 망의 RTT시간을 계산한다. 이는 불필요한 TCP에서의 재전송을 막기위한 방법이며 중간에 Probe Packet이 손실된다면 버티컬 핸드오버의 성능이 떨어지게 된다.

3.2.3. 하향 버티컬 핸드오버

1. ERTT와 HEnd를 사용하는 방법[6]

새로운 네트워크에 버티컬 핸드오버를 시작하기 전에 RTT시간을 측정하여 ERTT(Expected Round-Trip Time) 값으로 저장 하여 핸드오버 수행시 정상적인 동작인지 아닌지 판별하기위해 사용하고 핸드오버가 수행된 직 후에 혼잡원도우의 마지막 바이트번호를 HEnd에 저장한다. 이것은 핸드오버를 시작하는 메시지를 받을때의 혼잡원도우의 크기이고 이 마지막 바이트번호 이상 수신된 후의 패킷이 새로운 네트워크로부터 전송되는 패킷이라는것을 알려준다.

IV. 결론

본 논문에서는 버티컬 핸드오버 상황에서 시그널링과 인증과 등록과정에 의한 지연과 패킷이 송신측에서 수신측으로

전달될 때 패킷의 순서번호가 서로 맞지 않아 TCP에서의 혼잡한 상황으로 인식되어져서 처리량이 급감되는 현상을 보았다. 이런 문제들에 대한 해결 방법들은 시그널링의 지연을 줄여주는 방법으로써 Hierarchical MIPv6와 Fast MIPv6가 있으며 핸드오버의 과정에서 급격한 링크의 특성 변동으로인한 RTT변동에 대하여 TCP는 능동적으로 대처하지 못하는 문제점을 가지고 있고 이 문제를 해결하기위해서 상·하향 버티컬 핸드오버에 대한 다양한 해결점들에 대해서 조사해 보았다. 현재는 동향에 대한 조사였으므로 다음 연구에서 네트워크의 대역폭과 RTT시간을 계산하고 상·하향 핸드오버를 수행할 때 TCP성능 변화에 대한 조사를 수행하고 좀더 효과적인 망의 탐색방법에 대한 연구를 진행할 예정이다.

참고 문헌

- [1] E. Blanton and M. Allman, "On Making TCP More Robust to PacketReordering", ACM Computer Communication Review, Vol. 32, No.1, pp. 20-30, Jan. 2002.
- [2] C. Ma and K. Leung, "Improving TCP Robustness under Reordering Network Environment", Proc. IEEE LOBECOM'04, Vol. 2, pp. 828-832, Nov. 2004
- [3] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, "Freeze-TCP: A True End-to-End TCP Enhancement Mechanism for Mobile Environments," in Proc. of INFOCOM 2000, June 2000.
- [4] 서기남, 임재성 "버티컬 핸드오버 환경에서 종단간 이동성 관리 및 TCP 흐름 제어기법" 한국통신학회논문지 05-6 Vol.33, No.6B
- [5] J.C. Moon, B. G. Le, "Rate-adaptive Snoop: A TCP Enhancement Scheme over Rate-controlled Lossy Links," IEEE/ACM Transactions on Volume 14, Issue 3, June 2006.
- [6] 최여민, 송주석 "하향 수직 핸드오버 상황에서 송신자에 기반을 둔 TCP 혼잡제어 기법" 한국통신학회논문지 08-06 vol.33, No. 6