

# 이동 IP의 성능향상을 위한 Labeled-COA 기법과 평가

윤형구<sup>o</sup>, 최성민, 강필용, 신용태  
승실대학교 컴퓨터학과

Labeled-COA method and Appraisal for improved Mobile IP

Hyunggu Yoon<sup>o</sup>, Seongmin Choi, Pilyoung Kang, Yongtae Shin  
Dept. of Computer, Soongsil University

## 요 약

현재의 인터넷은 IPv4를 기반으로 하며, CIDR(Classless InterDomain Routing)을 이용하여 IP의 고갈을 막으려고 노력하고 있다. 그러나, 이는 근본적인 대책이 아니다. 그 때문에 IP는 점차 고갈되어 미래에는 IPv6로 옮겨갈 것이다. 이렇게 IPv6로 발전함에 따라 IP 주소는 매우 늘어날 수 있으나, 출발지 주소와 목적지 주소의 헤더가 커짐에 따라 헤더의 IP 주소를 모두 읽어보는 데에 많은 비용(cost)이 들게 된다. 따라서, 각각의 ISP(Internet Service Provider)들은 MPLS(Multiprotocol Label Switching)를 사용하여 이 문제를 해결하려 노력할 수 있다. 또한, MPLS는 기존의 IP 주소를 이용한 라우팅은 경로를 찾기 위해 필요이상으로 많은 정보를 요구한다고 보고, 레이블 개념을 도입하여 스위칭해주는 역할을 하기 때문에 라우팅을 위해 IPv6로 인하여 더욱 길어진 IP 주소를 요구하지 않고 단지 레이블을 붙임으로 속도를 향상시킬 수 있다. 이러한 환경에서의 이동 IP는 COA를 얻게되면 레이블을 붙여서 목적지까지 전달되게 되고, 이렇게 전달되는 이동 IP에서 COA와 레이블을 합쳐서 하나로 보내는 방식으로 이동 IP의 효율을 높일 수 있다.

## 1. 서 론

인터넷 사용자의 증가에 비례하여, IP의 고갈도 빨라지고 있다. 인터넷의 참여 조직이 그다지 많지 않던 시대에는 별 문제가 없었다. 그러나, 요즘 들어 인터넷의 발전에 따라 IP 주소의 부족 현상이 발생하고 있다[7,8].

현재에는 CIDR기술로 IP 주소 문제를 해결하려고 노력하고 있다. CIDR은 기본적으로 class A는 224(1천 600만개)의 host를, class B는 216(6만 5000개)의 host를, class C는 255개의 host를 가지고 있다. 그런데 실제로 대부분의 organization들에게 class B가 너무 크고 class C가 너무 작아서 쓸 수 없는 organization들은 class B를 요구할 수밖에 없기 때문에 class B는 고갈될 수밖에 없다. CIDR에서의 기본적인 구성은 인터넷 주소 space를 쪼개고, 라우팅 정보의 집합들을 위한 mechanism을 제공한다는 것이다. 따라서, 기본적인 Idea는 일련의 class C를 할당하는 것이다. 이렇게 하여 IP의 고갈을 막아 보고자 노력하지만, 결국 근본적으로 부족한 IP를 해결할 수 있는 방법은 되지 못한다.

이런 상황을 극복하고자 하려는 노력이 IPv6이다. 이것은 장기적이고 근본적인 해결 방법이다. 이로써 주소 공간의 절대수가 부족한 문제를 해결할 수 있다.

이런 환경에서는 IP 주소를 모두 확인하지 않고 라우팅을 할 수

있는 방법을 사용하는 것이 보다 효율적이다. MPLS는 이러한 방법의 하나로 네트워크 자원 이용률과 traffic performance를 최적화시킬 수 있는 효율적이고 신뢰성 있는 네트워크를 만들 수 있다. 본 논문에서는 이렇게 IPv6를 쓰고, MPLS를 사용하는 ISP에서 이동 IP가 COA를 받게 되면 그 COA에 해당하는 레이블을 할당 받게 된다. 또한, 성능을 높이기 위해서 레이블과 COA를 binding할 수 있다. 2장에서는 IPv6와 MPLS, 그리고 이동 IP를 간략하게 살펴보고, 3장에서는 이동 IP가 MPLS환경에서 어떻게 동작해야 하는지를 설명하고, 4장에서 제안된 기법을 평가한다. 마지막으로 5장에서는 결론 및 향후 과제를 제시하고자 한다.

## 2. 관련 연구

### 2.1. IPv6

IPv4는 현재 인터넷에서 사용되는 프로토콜이며, 32비트 주소를 사용한다. 그러나, 조만간 IP 주소가 고갈될 것이라는 예측 때문에 IPng(IP next generation)에서의 차세대 IP 프로토콜 선정작업에 오른 SIPP(Simple Internet Protocol Plus)가 채택되었고, 이 프로토콜이 IPv6라는 명칭을 갖게 되었다[8].

IPv6는 128비트의 주소 field를 가지므로, IPv4에서의 주소 할당

공간 부족 문제를 많이 해소하였고, 헤더 포맷을 간략화 시키고, 확장 헤더를 사용하여 불필요하게 헤더에 있는 부분을 없애고, 필요한 부분은 확장 헤더로 옮김으로써, 헤더의 낭비를 막았다. 그로 인해, 보안이나 인증 등은 확장 헤더를 이용해 인터넷 계층에서 보안 기능을 제공하게 되었다.

아래의 [그림 1]은 IPv4와 IPv6의 헤더를 보여주고 있다.

ver	L	TOS	total length	
identification		F	frag. offset	
TTL	protocol		header checksum	
source address				
destination address				

ver = version  
L = header length  
F = flags  
32bit address field  
< IPv4의 header >

ver	traffic class	flowlabel	
payload length		next header	hop limit
source address			
destination address			

128bit address field  
< IPv6의 header >

[그림 1] IPv4와 IPv6의 헤더비교

## 2.2. MPLS

앞으로 인터넷 사용의 확산이 보다 고속화되어 IP고갈 문제를 고려해 보면, IPv6로 옮겨 갈 것이다. IPv6에서는 source와 destination의 주소가 IPv4보다 길어져서 일일이 IP계층까지 살펴보고 라우팅하는 것은 낭비이다[1]. 또한, 현재 Internet은 사용자의 급증으로 인한 대역폭문제, 서버의 증가로 인한 라우팅문제가 있고, 사용자들은 QoS를 보장받기를 원한다[5].

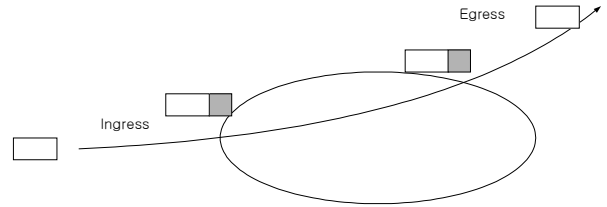
이를 해결하는 방안으로는 고성능 라우터, 3계층 스위칭 방식 등 다양한 해결책이 제시되고 있으며, 이중 최근 들어 관심을 모으는 기술이 MPLS이다. MPLS는 백본 네트워크의 2계층과 3계층을 연결함으로써 네트워크 서비스를 통합하면서도 동시에 기타 IP 또는 ATM 연결 옵션에 비해 비용을 줄일 수 있고, 복잡하지도 않다[4].

이것은 레이블 Swapping을 이용한 Data Link forwarding paradigm과 네트워크 레이어 라우팅의 통합이라고 할 수 있다. 이는 네트워크를 통해 움직이는 각각의 패킷에 대해 레이블을 부여하고 스위치 장비에서는 이 레이블만을 읽어서 패킷을 전송함으로써 속도를 향상시키는 기술을 의미한다. MPLS에서의 가장 큰 특징은 레이블 스위칭 기법으로 기존 라우터가 IP 헤더에 있는 모든 정보를 읽어 데이터를 전송했던데 반해 레이블 스위칭은 데이터와 헤더에 데이터의 정보를 나타내는 짧은 레이블을 첨부하여 그를 통해 최적의 전달 경로를 찾는다 따라서, 기존의 방식에서 데이터가 대량으로 라우터에 접속될 경우 라우터는 정체현상을 일으키고 주소만을 파악하기 때문에 QoS나 VPN(Virtual Private Networks)과 같은 부가서비스를 제공할 수 없었으나, 레이블 스위칭 방식은 데이터 양에 관계없이 고속의 전송속도를 유지할 뿐만 아니라 다양한 부가서비스도 제공할 수 있다[3].

MPLS의 핵심요소는 레이블 swapping, 레이블 binding, 레이블 distribution으로, 레이블 swapping은 입력 레이블을 출력 레이블로 변경하는 것을 의미하고 레이블 binding은 레이블을 특정 데

이터 흐름에 대응하게 하고, 레이블 distribution은 레이블 binding을 인접노드에 전파시킨다.

[그림 2]에서와 같이 MPLS는 Ingress에서는 패킷의 레이블을 첨가하고, Intermediate 라우터에서는 레이블에 따라 패킷을 처리한다(레이블로 LIB를 검색). 또, Egress에서는 패킷에서 레이블을 삭제한다.

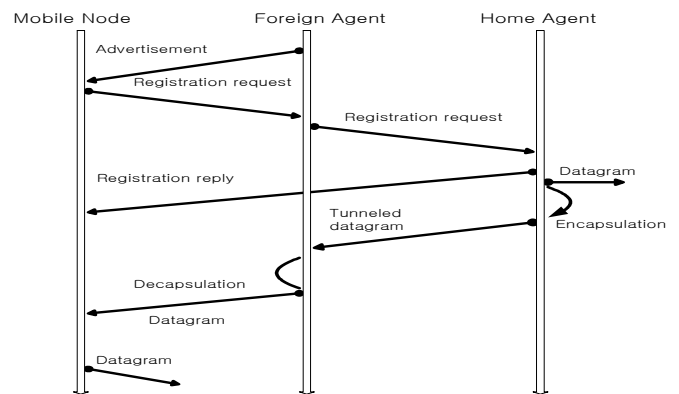


[그림 2] MPLS의 동작

## 2.3. 이동 IP

이동 IP는 IP가 이동성을 가진다는 의미이다. 이동 노드들이 인터넷에서 쓰여질 경우 현재 인터넷 네트워크 프로토콜 버전에서는 충분하지가 않다. 예를 들어, 노트북을 가진 사용자가 열차를 타고 출장을 간다고 할 경우에 사용자는 계속하여 이동하게 된다. 이동하는 모든 장소에 관계없이 항상 네트워크에 자신의 컴퓨터를 연결하여 수시로 필요한 정보를 얻을 수 있도록 하는 것이 이동 Internet 프로토콜의 기본 Idea이다.

즉, 자신의 IP 서브넷으로부터 다른 IP 서브넷으로 연결의 링크-계층 접속지점을 변화시켰음에도 불구하고 자신의 IP를 변경시키지 않고 다른 노드들과 통신하는 것을 말한다. 이것은 이동 노드들에게 COA(Care-Of Address)를 할당함으로써 가능하다. 이동 노드가 속해 있던 Home Agent가 이동 노드의 Home 주소로 전송 요청된 패킷을 COA를 사용하여 이동 노드에게 재전송하는 Tunneling기법을 사용하는데, 터널링기법은 새로운 IP 헤더를 원래의 패킷 헤더 앞부분에 더해서 Encapsulate된 패킷을 COA로 전송하고, 이를 수신하면 앞부분의 헤더를 제거하여 Decapsulate된 원래의 패킷을 받는 방법이다. 따라서 이동 IP에서는 [그림3]과 같이 전송이 이루어지게 된다[8].



[그림 3] 이동 IP에서의 Data전송

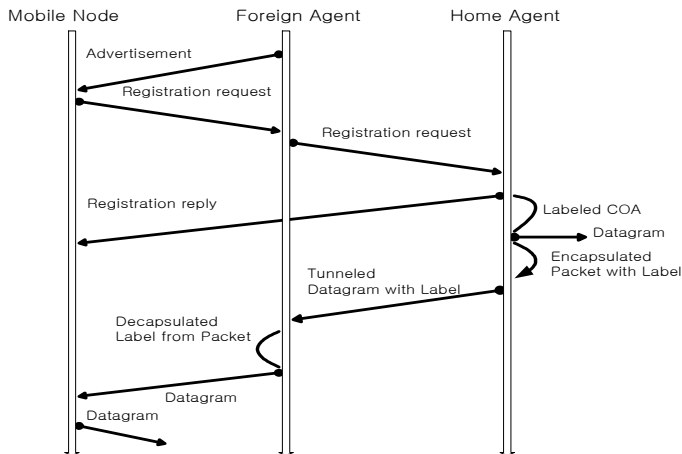
### 3. MPLS환경에서의 이동 IP 기법

기존의 이동 IP기법을 MPLS환경에 적용시키면 이동 노드는 Foreign Agent에 등록을 하고, Foreign Agent는 Home Agent에 등록 요청을 하여 이동 노드가 등록에 대한 응답을 받는다. 그 후, Encapsulate된 패킷이 Correspondent 노드와 통신을 하게 된다. 이때, 이동 노드는 COA를 부여받는다. 그 후, Correspondent 노드는 이동노드가 부여받은 COA와 통신을 이루게 된다.

또한, MPLS는 Layer 3의 라우팅을 Layer 2의 스위칭으로 대체함으로써, 효율을 높이고자 한다.

따라서, 두 기술을 접목시키면, 이동 노드가 이동하게 되면 Foreign Agent로부터 COA를 부여받아, Correspondent 노드는 COA에 해당하는 곳으로 스위칭을 하기 위해, COA 헤더앞에 레이블을 붙인다. 이렇게 하여, Correspondent 노드는 레이블을 보고 이동노드와 통신을 하게된다.

그렇게 되면, Correspondent 노드와 이동 노드가 통신을 하기 위해서 필요한 부분은 COA가 아니라, 레이블이 된다. 즉, COA도 이동 노드의 위치를 가리키고, 레이블도 이동 노드의 위치를 가리키므로 COA와 레이블을 binding해 놓고, COA는 없애고, 레이블만으로 이동 노드와 통신을 하여 효율을 높일 수 있다.



[그림 4] MPLS기법을 사용한 이동 IP에서의 Data 전송

### 4. IPv6 기반, MPLS환경에서의 이동 IP 기법 분석

이 논문에서는 IPv6와 MPLS를 기반으로 하고 있다. 이유는 앞으로 IP의 고갈은 IPv6으로 해결을 할 수 있고, 이렇게 IPv6로 옮겨가게 되면 주소 공간을 많이 차지하게 된다. 그러면, 각각의 라우터는 패킷들의 라우팅을 결정하기 위해 IP까지 살펴볼게 되는데 이는 매우 비효율적이다. 왜냐하면, IPv4에서의 주소 길이에 비해, IPv6에서는 주소의 길이가 매우 길어졌기 때문이다. 이 때문에 스위칭기법을 사용하여 레이블을 붙임으로써, 주소전체를 살펴볼 필요가 없고, 라우팅은 속도가 향상되어 많은 패킷이 한꺼번에 오더라도 처리할 수 있는 MPLS를 사용하게 될 것이다. 이렇게 되면, 처리해야 하는 일이 줄어들기 때문에 다른 서비스도 첨가할 수 있게 된다.

또한, 이동 IP를 지원하기 위하여 COA에 대한 레이블을 붙임에 있어서, COA를 레이블과 Binding하여 COA를 생략하고, 레이블만을 붙임으로써, 헤더의 길이를 줄일 수 있다. 이것은 작은 네트워크에서는 큰 성능을 발휘하지 못하지만, COA만큼 줄어든 헤더는 네트워크가 크면 클수록 점점 더 이로움을 가져다 줄 것이다.

즉, 아래의 [그림6]와 같이 요약될 수 있다. 이 그림에서는 라우터의 개수를 1~1000개까지 걸쳐간다고 가정하고, 각각 라우터의 스피드를 10~100Mbps, IPv6에서 라우팅을 위해 드는 비트수를 32~96 비트, MPLS에서 라우팅을 위해 드는 비트수를 20비트로 하였을 때의 시뮬레이션을 보여주고 있다. 이 그림을 보면 IPv6에서 라우팅을 위해 32, 64, 96비트를 사용한 것을 볼 수 있는데, 이유는 IPv4에서와 같이 라우팅을 위해 모든 바이트를 읽을 필요가 없기 때문에 IPv6에서도 128비트를 모두 읽지 않고 일부만을 읽었을 경우를 생각해 주었다. 즉, IPv4에서 라우팅을 위해 전체의 32비트를 읽지 않는다. 대신, netid만을 읽어서 그 방향으로 보내줄 수 있다. 이런 방법이 IPv6에서 사용되었을 경우, 128비트를 모두 읽는 것은 낭비이고, 그중 32~96비트로 라우팅 방향을 알 수 있기 때문에 이러한 방식을 택한다.

또한, 이 시뮬레이션에서는 다른 요소들은 빼고, 오로지 라우팅을 위해 라우터에서 비교하는 주소 부분만을 고려한다. 그 이외의 요소들은 제외하고, 오로지 주소를 비교하는데 저속인 10Mbps의 속도를 가진 라우터에서부터 빠른 이더넷과 같은 고속의 라우터를 고려하여 10Mbps~100Mbps의 라우터까지를 고려하였다.

이러한 요소로 시뮬레이션을 한 결과의 C코드의 일부를 아래에서 보여준다.

```
for (router_su=1;router_su<=1000;router_su++)
for (ipv6_addr=32;ipv6_addr<=96;ipv6_addr+=32)
f                                o                                r
(router_speed=10;router_speed<=100;router_speed+=10)
{
ipv6_time=(ipv6_addr / router_speed)*router_su;
mpls_time=(MPLS_Switch / router_speed)*router_su;
difference=ipv6_time - mpls_time;
fprintf(out, " %f, %f, %f, %d, %f, %f, %f",router_su,router_speed,ipv6_addr,MPLS_Switch,ipv6_time,mpls_time,difference);
fprintf(out, "WnWn");
}
```

[그림 5] 제안된 기법의 성능 시뮬레이션 의사-코드

아래의 그림은 이 시뮬레이션 코드의 결과 중 일부를 발췌한 것으로 각각의 성능비를 알 수 있다.

A	B	C	D	E	F	G
1	10	32	20	3.2	2.0	1.2
1	20	32	20	1.6	1.0	0.6
1	30	32	20	1.0667	0.6667	0.4
1	40	32	20	0.8	0.5	0.3
1	50	32	20	0.64	0.4	0.24
1	60	32	20	0.5333	0.3333	0.2
1	70	32	20	0.4571	0.2857	0.1714
1	80	32	20	0.4	0.25	0.15
1	90	32	20	0.3555	0.2222	0.1333
1	100	32	20	0.32	0.2	0.12
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	10	32	20	6.4	4.0	2.4
2	20	32	20	3.2	2.0	1.2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	100	96	20	1.92	0.4	1.52
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	10	96	20	9600	2000	7600
1000	20	96	20	4800	1000	3800
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	100	96	20	960	200	760

[그림 6] IPv6와 MPLS의 라우팅 속도 차이

[그림6]에서 A는 라우터의 개수, B는 라우터의 스피드, C는 IPv6에서 라우팅을 위해 드는 비트 수, D는 MPLS에서 라우팅을 위해 드는 비트 수, E는 IPv6를 이용했을 때 걸린 마이크로 초, F는 MPLS를 이용했을 때 걸린 마이크로 초, G는 이 둘의 차이를 나타낸다. 이 식을 토대로 시뮬레이션을 할 수 있다. 그 결과, 위의 그림에서 보는 바와 같이, 라우터를 지날수록 둘(IPv6에서 라우팅을 할 때와 MPLS에서 라우팅을 할 때)의 차이는 커진다. 물론, 라우터의 속도가 빨라질수록 그 차이는 줄어들게 되지만, 라우터를 1000개를 지나고 10Mbps를 사용하는 라우터에서는 그 차이가 7600마이크로 초가 되어 주소를 비교하는 시간에만 7600마이크로 초의 성능 향상을 가져올 수 있다.

## 5. 결론 및 향후과제

이 논문은 기존의 라우팅기법에 MPLS의 스위칭기법을 도입함으로써 많은 변화를 일으킨다. 우선, MPLS를 사용함으로써 라우팅이 간소화되고, 그로 인해 라우팅을 위해 많은 일을 하지 않으므로 다른 서비스가 가능하다. 이것은 인터넷에 큰 변화를 가져올 것으로 기대된다. 보내는 패킷마다 헤더의 길이가 줄어서 전송되므로 하드웨어의 부하는 물론이고 라우팅을 하는 데에도 성능 향상을 기대할 수 있다. 또한 실시간 서비스를 고려해 볼 때, 실시간 서비스는 최대한 압축을 하여 빨리 보내는 것을 중요시하기 때문에 [그림 4]에서와 같이, 이 기법을 사용하여 보내는 것은 더욱 빠른 서비스를 할 수 있는 요건이 된다. 이를 이용하면 기존의 이동 노드에게 제공하던 실시간 서비스보다 향상된 서비스를 할 수 있고, 더 많은 라우터를 경유할 때마다 그 효능을 더욱 더 커질 수 있다.

또한, 향후 이러한 기반 위에 QoS를 향상시키는 방법 등을 첨가시킨다면 보다 품질 좋은 실시간 서비스를 이동 노드에게 제공할 수 있어서, QoS를 제공하는 방법 등이 과제로 떠오를 수 있다.

## [참고문헌]

- [1] R. Callon, P. Doolan, N. Feldman, A. Fredette, G. Swallow, A. Viswanathan, "A Framework for Multiprotocol Label Switching", draft-ietf-mpls-framework-05.txt, <internet-draft>September 1999.
- [2] Charles E. Perkins, "Mobile IP, Design Principles and Practices"
- [3] "MPLS를 통한 New World VPN 구축", Cisco Systems Inc, White Paper
- [4] Arup Acharya, Frédéric Griffoul, Furquan Ansari, "IP Multicast Support in MPLS", Proceeding of the IEEE ATM Workshop '99, May 1999.
- [5] Eric C. Rosen, Arun Viswanathan, Ross Callon, "Multiprotocol Label Switching Architecture", draft-ietf-mpls-arch-06.txt, <internet-draft>August 1999.
- [6] Eric C. Rosen, Daniel Tappan, Guy Fedorkow, Tony Li, "MPLS Label Stack Encoding", draft-ietf-mpls-label-encaps-07.txt, <internet-draft>September 1999.
- [7] Girish Chiruvolu, Anshul Agrawal, Marc Vandenhoute, "Mobility and QoS support for IPv6-based Real-time Wireless Internet Traffic", Proceedings of the 1999 IEEE International Conference on Communications, June 1999.
- [8] Claude Castelluccia, "A Hierarchical Mobility Management Scheme IPv6", Proceedings of the Third IEEE Symposium on Computers and Communications, June 1998.