Mobile IP를 이용한
단말기 이동성 통신 예제

문서 정보

구 분	소 속	성 명	비 고
제 목	Mobile IP를 이용한 단말기 이동성 통신 예제		
작성자 및 검토자	한국전자기술연구원	정한균	
	한국전자기술연구원	성동규	
문서 버전	0.2		
상 태	작성중		
문서 소유	한국전자기술연구원 모빌리티플랫폼연구센터		

개정 이력

수정 일자	수정자	문서 버전	내 용
2023년 8월 21일	성동규	0.1	- 초안 작성 시작
2023년 8월 24일	성동규	0.2	- 내용 보완
2023 8월 31일	성동규	0.3	- 내용 보완
			-
			-
			-
			-

차 례

1. 개요 ····· 5
1.1. 용어 ····· 5
1.2. Mobile IP ····· 5
1.3. 터널링 ····· 6
1.3.1. IPIP(IP in IP) ····· 6
2. 라우팅 예제 ····· 7
2.1. 예제1. MN HA 직접 접속 ····· 8
2.1.1. 노드 장치 별 라우팅 테이블 설정 방법 ····· 10
2.1.2. 예제1 시험 결과 ····· 12
2.2. 예제2. MN가 FA를 통해 HA에 접속 ····· 14
2.2.1. HA와 FA의 Tunneling을 위한 설정(IP in IP) ····· 16
2.2.2. 노드 장치 별 라우팅 테이블 설정 방법 ····· 17
2.2.3. 터널링을 통해 전달하는 IPIP방식으로 캡슐화된 패킷 확인 ····· 19
2.2.4. 예제 2. 시험 결과 ····· 21

1. 개요

본 문서에는 Mobile IP 표준을 이용해 이동하는 장치가 자신과 같은 서브넷에서 다른 서브넷을 이동할 때, 고정된 IP Address로 네트워크와 접속을 유지하는 방법을 기술한다.

1.1. 용어

표 1 용어

항목	약자	설명	비고
Mobile Node	MN	하나의 네트워크 또는 서브넷에서 다른 네트워크 또는 서브넷으로 연결점을 변경하는 장비	
Home Agent	HA	MN의 Home Address로 서비스를 제공하는 라우터로 MN이 Home 네트워크에서 멀어졌을 때, 터널링을 통해 데이터그램은 전달	
Foreign Agent	FA	MN의 CoA로 서비스를 제공하는 라우터로 MN이 접속해있는 동안 HA로부터 터널링된 데이터그램을 디터널링해 MN으로 전달	
Care-of Address	CoA	HA가 MN으로 터널링을 통해 데이터그램을 전달할 때, 디터널링을 하는 터널의 종료지점으로 아래 두가지 유형의 Address를 사용할 수 있음 - FA CoA는 MN이 등록된 FA의 주소 - co-located CoA는 MN의 네트워크 장치 중 하나와 연관된 외부에서 획득한 Local address	
Correspondent Node	CN	MN과 통신하는 Peer로 본 문서에서는 Station(서버?)	
Foreign Network	-	MN의 Home 네트워크와 다른 네트워크	
Home Address	-	MN가 접속한 네트워크와 관계없이 유지되는 MN의 IP Address	
Home Network	HN	MN의 Home Address와 같은 서브넷을 갖는 네트워크 혹은 가상 네트워크 표준 IP 라우팅을 통해 MN Home Address로 향하는 데이터그램을 MN의 HN으로 전달	
Tunnel	-	캡슐화된 데이터그램의 경로로 이 데이터그램은 캡슐을 해제 할 수 있는 노드까지 라우팅이 되고 해당 노드는 최종목적지로 데이터그램은 전달	

1.2. Mobile IP

Mobile IP는 IETF의 표준 통신 프로토콜로 MN(Mobile Node)들이 고정된 IP 주소를 유지하며 네트워크 간 이동을 할 수 있도록 고안되었다. RFC 5944(IP Mobility Support for IPV4)[1]에 기술되어 있으며 아래 기능을 포함한다.

- 각 MN는 현재 연결된 인터넷과 관계없이 항상 같은 IP address(Home Address)로 식별
- MN는 HA(Home Agent) 외 다른 CoA(Care-of-Address)에 연결되면, 현재 접속된 인터넷 연결점의 정보를 전달한다. HA는 이 정보를 이용해 해당 CoA를 등록
- HA는 MN으로 데이터그램을 보낼 때, HA에서 터널링(IP in IP encapsulation[2], Minimal encapsulation, GRE encapsulation)을 통해 MN과 연결된 CoA로 데이터그램을 전달
- 터널의 종료 지점인 MN과 연결된 CoA는 HA로부터 전달받은 데이터그램을 MN으로 전달

또한, Mobile IPv6의 기술은 RFC 6275에 되어있다.

1.3. 터널링

터널링의 HA가 FA로 데이터그램을 전달할 때 사용하며 RFC 5944에서 제안하는 캡슐화 방법(Encapsulation Types)은 'IP in IP'(RFC 2003 "IP Encapsulation within IP")이며, MN에서 요청하거나 HA의 재량에 따라 Minimal, GRE 터널링 방식도 가능하다.

본 문서에서는 IP in IP를 캡슐화 방법으로 사용한다.

1.3.1. IPIP(IP in IP)

TBA

표 2. Mobile IP에서 IPIP IP 헤더 구성

항목	Source IP Address	Destination IP Address	비고
Outer IP Header	HA Address	FA CoA	
		MN co-located CoA	
IP Header	CN IP Address	MN IP Address	

2. 라우팅 예제

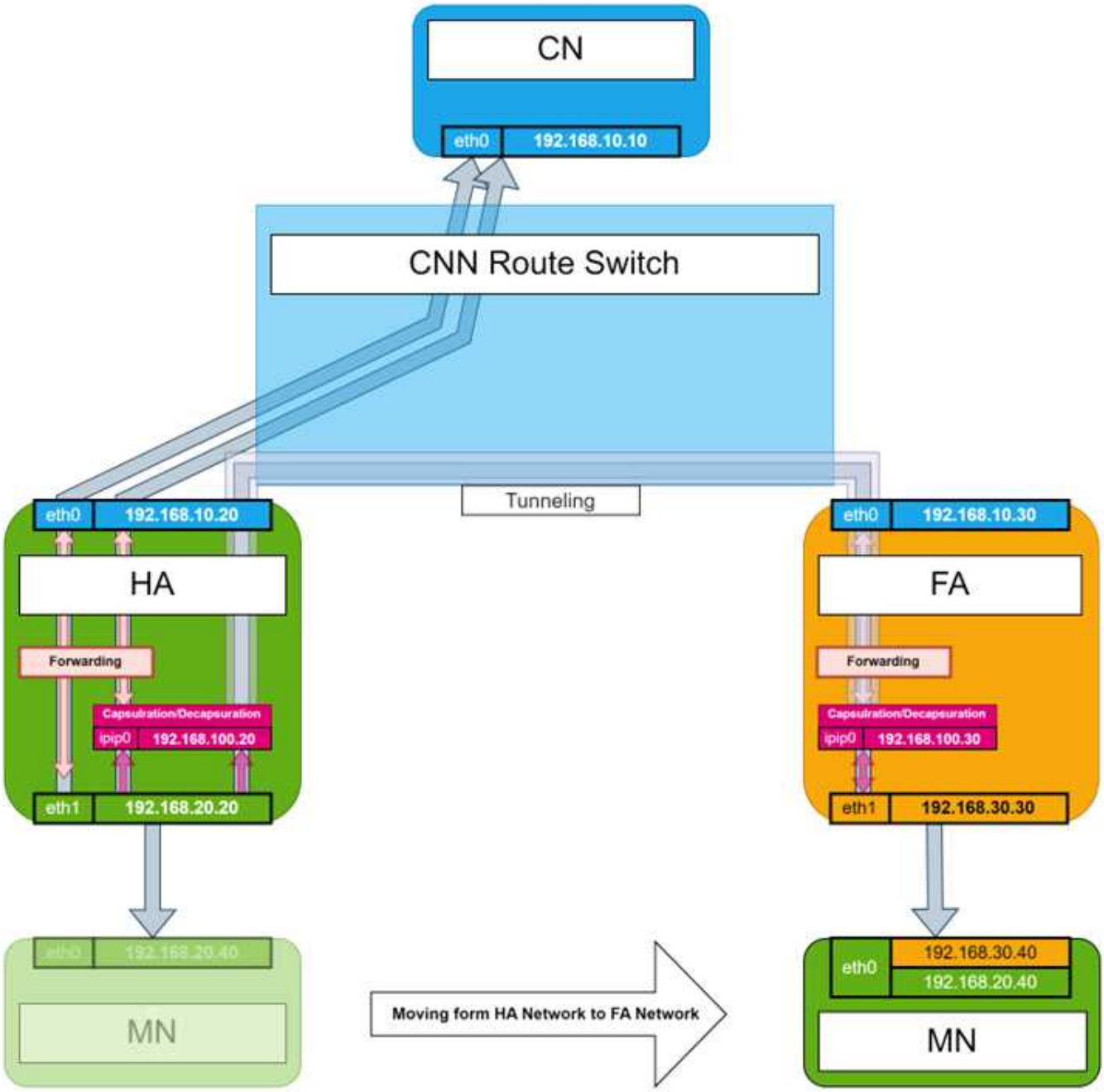

그림 3. Mobile IP를 이용한 단말기 이동성 통신 전체 시험 구성도(MN이 HA Network, HA Network에서 FA Network로 이동했을 때 통신 예제)

2.1. 예제1. MN HA 직접 접속

MN가 HA의 네트워크 안에 있고 HA와 직접 연결되었을 때, Ping 통신을 통해 MN과 CN의 연결을 확인한다. 아래 표 3은 노드의 장치별 IP Address, 표 4는 Routing table이다.

표 3. 예제1. 노드의 장치별 IP Address

노드	장치	IP Address	네트워크 영역	운영체제	비고
CN	eth0	192.168.10.10	CN 네트워크	Windows	
HA	eth0	192.168.10.20	CN 네트워크	Linux	
	eth1	192.168.20.20	HA 네트워크		
MN	eth0	192.168.20.40	CN 네트워크	Linux	

표 4. 예제1. 노드의 Routing table

운영체제	노드	Route Table								비고
Windows		Destination	Netmask	Gateway	Interface	Metric				
	CN	192.168.20.0	255.255.255.0	192.168.10.20	192.168.10.10	45				
Linux		Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface	
	HA	0.0.0.0	0.0.0.0	255.255.255.0	U	100	0	0	eth1	
		192.168.10.0	0.0.0.0	255.255.255.0	U	50	0	0	eth0	
	MN	0.0.0.0	192.168.20.20	255.255.255.0	U	100	0	0	eth0	

표 5. 예제1. Metric 값 설정 기준(수치는 예제에서 임의로 정함)

Destination	Gateway	Local Interface	Gateway	Metric	비고
Static or Dynamic	Static	YES	-	50	
Any	0.0.0.0	YES	-	100	

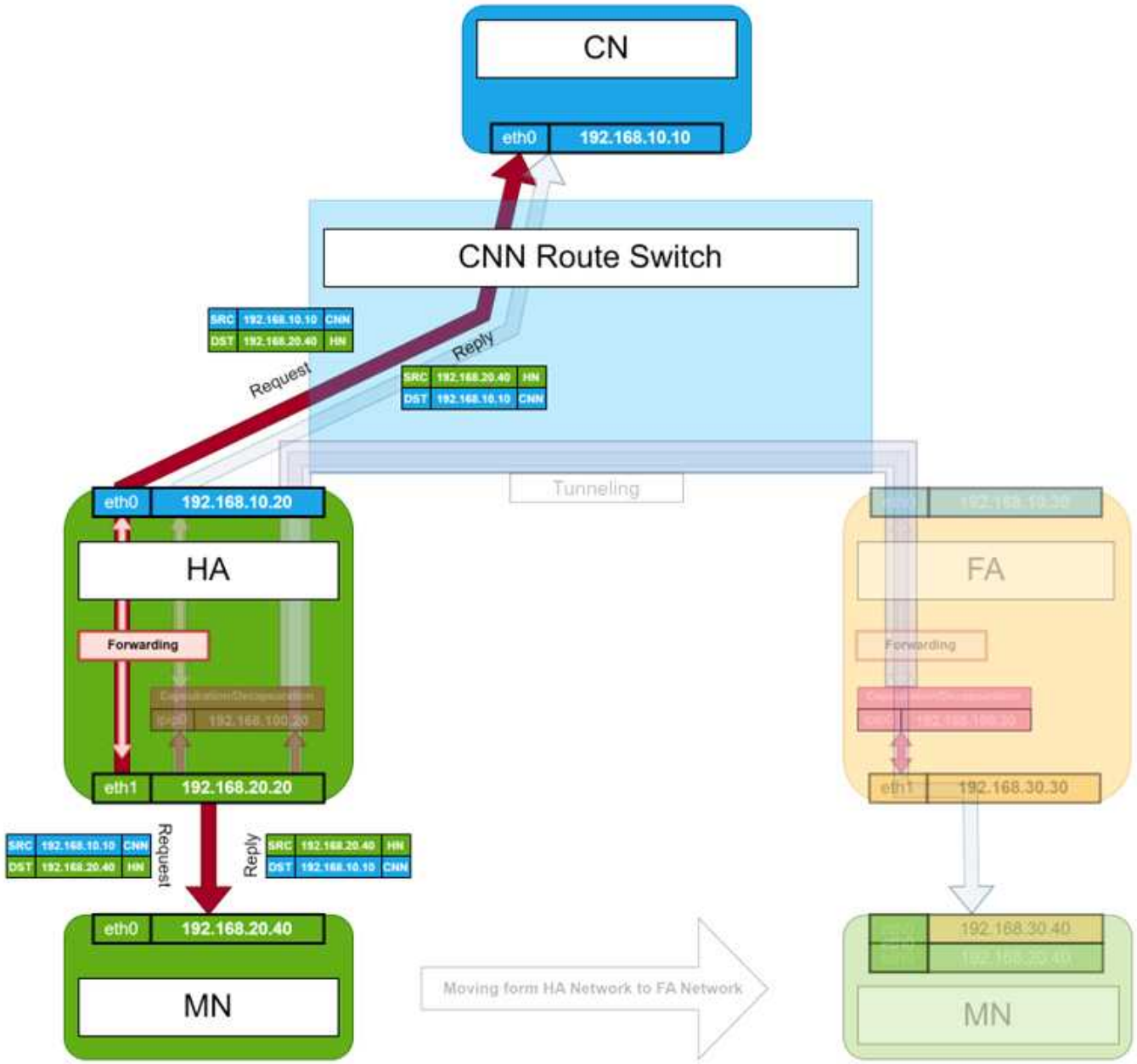

그림 4. 예제1 데이터그램 흐름도(CNN:CN Network)

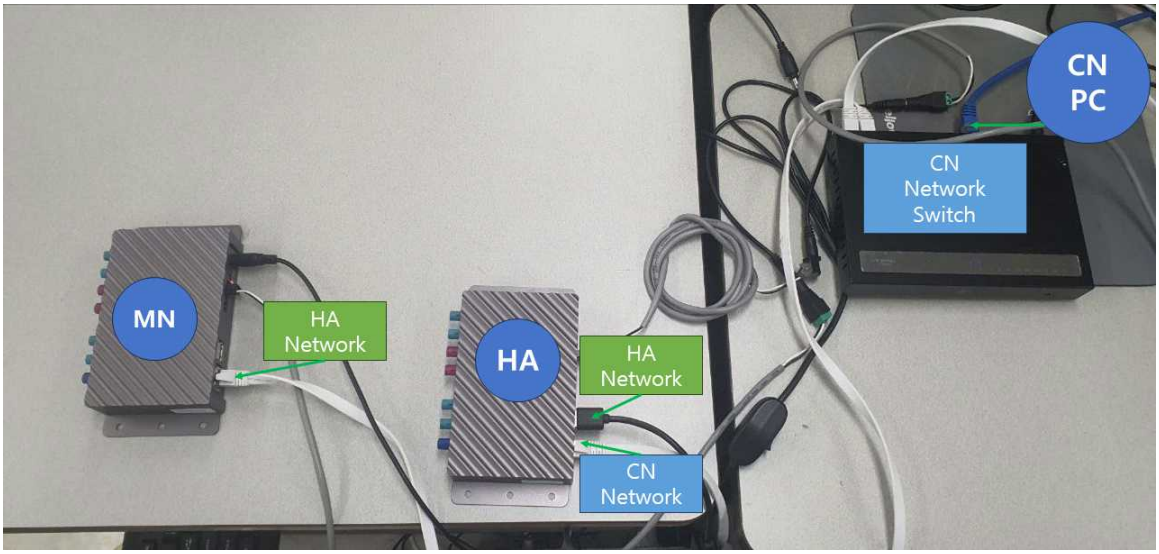

그림 5. 예제1 시험 구성 예시

2.1.1. 노드 장치 별 라우팅 테이블 설정 방법

2.1.1.1. CN(Windonw)

관리자모드로 터미널(PowerShell, CMD)을 켬

줄	명령어
1	route add 0.0.0.0 mask 0.0.0.0 192.168.10.20 metric 20 if 21

줄	명령어 설명
1	모든 트래픽을 주소가 192.168.10.10인 로컬 인터페이스(21)로 게이트웨이 192.168.10.20를 통해 전달하는 내용을 라우팅 테이블에 추가

***if 값은 장치의 IP Address가 IPv4 일 때, route print -4를 실행해 'Interface List'를 확인하면 찾을 수 있음**

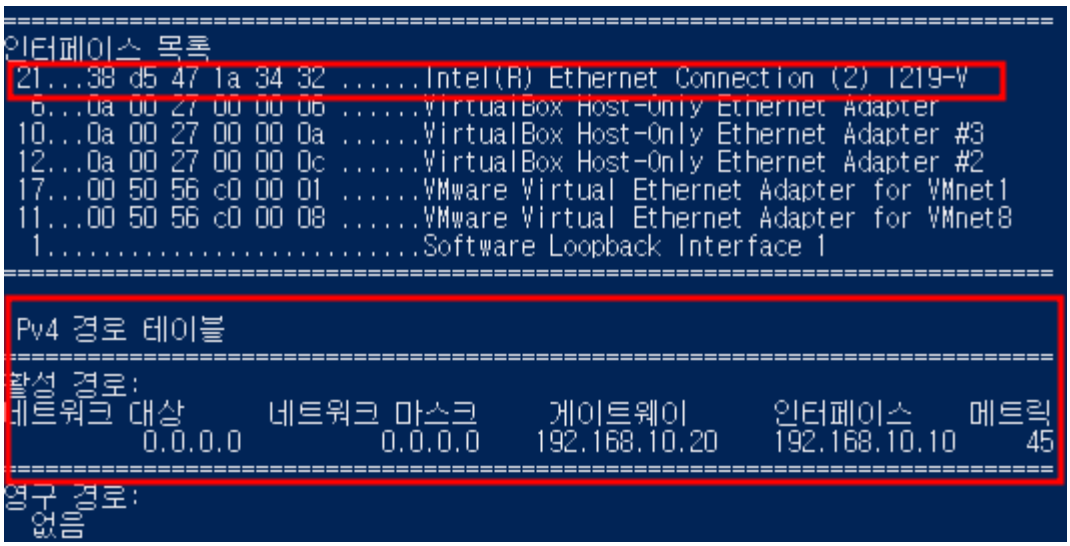

그림 6. CN Route table 설정 예시

2.1.1.2. HA(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip route add 192.168.10.0/24 dev eth0 metric 50
2	ip route add default dev eth1 metric 100

줄	명령어 설명
1	192.168.10.0~192.168.10.255이 Destination인 트래픽을 로컬 인터페이스 eth0를 통해 외부로 전달하는 내용을 라우팅 테이블에 추가
2	모든 트래픽을 로컬 인터페이스 eth1를 통해 외부로 전달하는 내용을 라우팅 테이블에 추가

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         0.0.0.0         0.0.0.0         U     100    0        0 eth1
192.168.10.0    0.0.0.0         255.255.255.0   U     50     0        0 eth0
```

그림 7. HA Routing table 설정 예시

2.1.1.3. MN(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip route add default via 192.168.20.20 dev eth0 metric 20 onlink

줄	명령어 설명
1	모든 트래픽을 로컬 인터페이스 eth0으로 게이트웨이 192.168.20.20를 통해 전달하는 내용을 라우팅 테이블에 추가

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.20.20   0.0.0.0         UG    20     0        0 eth0
```

그림 8. MN Routing table 설정 예시

2.1.2. 예제1 시험 결과

2.1.2.1. CN이 패킷을 전송하지 않을 때

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Fri Jan 29 03:00:54 2021

Chain INPUT (policy ACCEPT 160 packets, 12160 bytes)
 pkts bytes target     prot opt in     out     source               destination
  166 11124 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  eth0   *       192.168.30.30        192.168.20.20

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0            icmp --  eth0   eth1    192.168.10.10        192.168.20.40
    0     0            icmp --  eth1   eth0    192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 162 packets, 12368 bytes)
 pkts bytes target     prot opt in     out     source               destination
  141 18948 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  *      eth0    192.168.20.20        192.168.30.30
```

HA

eth0에서 eth1로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 증가하지 않음

eth1에서 eth0으로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 증가하지 않음

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Fri Jan 29 03:00:12 2021

Chain INPUT (policy ACCEPT 162 packets, 12368 bytes)
 pkts bytes target     prot opt in     out     source               destination
   29  1816 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  eth0   *       192.168.10.10        192.168.20.40

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 162 packets, 12312 bytes)
 pkts bytes target     prot opt in     out     source               destination
   21  3184 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      eth0    192.168.20.40        192.168.10.10
```

MN

eth0으로 수신하는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 증가하지 않음

eth0에서 송신하는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 증가하지 않음

*iptables 어플리케이션을 이용한 ICMP 패킷 흐름 확인

2.1.2.2. CN이 패킷을 전송할 때

```
PS C:\Users[redacted] ping 192.168.20.40 -n 4

Pinging 192.168.20.40 with 32 bytes of data:
Reply from 192.168.20.40: bytes=32 time=3ms TTL=62
Reply from 192.168.20.40: bytes=32 time=1ms TTL=62
Reply from 192.168.20.40: bytes=32 time=2ms TTL=62
Reply from 192.168.20.40: bytes=32 time=1ms TTL=62

Ping statistics for 192.168.20.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms
```

CN(192.168.10.10)

192.168.20.40으로 icmp(ping) 패킷 4개를 송수신 성공

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Fri Jan 29 03:13:53 2021

Chain INPUT (policy ACCEPT 4328 packets, 329K bytes)
 pkts bytes target     prot opt in     out     source               destination
 2298  139K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  eth0   *       192.168.30.30        192.168.20.20

Chain FORWARD (policy ACCEPT 8 packets, 480 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   240            icmp --  eth0   eth1    192.168.10.10        192.168.20.40
    4   240            icmp --  eth1   eth0    192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 4370 packets, 334K bytes)
 pkts bytes target     prot opt in     out     source               destination
 2273  231K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  *      eth0    192.168.20.20        192.168.30.30
```

HA

eth0에서 eth1로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 4개 증가

eth1에서 eth0으로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 4개 증가

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Fri Jan 29 03:12:29 2021

Chain INPUT (policy ACCEPT 4047 packets, 309K bytes)
 pkts bytes target     prot opt in     out     source               destination
  713 37516 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   240 ACCEPT     icmp --  eth0   *       192.168.10.10        192.168.20.40

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 4047 packets, 308K bytes)
 pkts bytes target     prot opt in     out     source               destination
  705 70828 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   240 ACCEPT     icmp --  *      eth0    192.168.20.40        192.168.10.10
```

MN

eth0으로 수신하는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 4개 증가

eth0에서 송신하는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 4개 증가

*iptables 어플리케이션을 이용한 ICMP 패킷 흐름 확인

2.2. 예제2. MN가 FA를 통해 HA에 접속

MN가 HA의 네트워크 안에 없고 FA CoA 네트워크에 접속했을 때, Ping 통신을 통해 MN과 CN의 연결을 확인한다. 아래 표 5은 노드의 장치별 IP Address, 표 6는 Routing table이다. HA와 FA간 터널링의 캡슐화 방법은 'IP in IP'를 사용한다.

*본 예제2에서 패킷의 흐름은 Mobile IP 표준의 데이터그램 흐름 같이 CN->HA->FA->MN->FA->CN이 아닌 CN->HA->FA-MN->FA->HA->CN이다.

표 14. 예제2. 노드의 장치별 IP Address

노드	장치	IP Address	네트워크 영역	운영체제	비고
CN	eth0	192.168.10.10	CN 네트워크	Windows	
HA	eth0	192.168.10.20	CN 네트워크	Linux	
	eth1	192.168.20.20	HA 네트워크		
	ipip0	192.168.100.20	tunnel 네트워크		
FA	eth0	192.168.10.30	CN 네트워크	Linux	
	eth1	192.168.30.30	FA CoA 네트워크		
	ipip0	192.168.100.30	tunnel 네트워크		
MN	eth0	192.168.20.40	CN 네트워크	Linux	
		192.168.30.40	FA CoA 네트워크		

표 15. 예제2. 노드의 Routing table

운영체제	노드	Route Table								비고
Windows		Destination	Netmask	Gateway	Interface	Metric				
	CN	192.168.20.0	255.255.255.0	192.168.10.20	192.168.10.10	45				
Linux		Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface	
	HA	0.0.0.0	0.0.0.0	0.0.0.0	U	100	0	0	eth1	
		192.168.10.0	0.0.0.0	255.255.255.0	U	50	0	0	eth0	
		192.168.20.0	0.0.0.0	255.255.255.0	U	50	0	0	ipip0	다운스트림
		192.168.30.30	192.168.30.30	255.255.255.255	UGH	20	0	0	eth0	다운스트림
	FA	0.0.0.0	0.0.0.0	0.0.0.0	U	100	0	0	eth1	
		192.168.10.0	0.0.0.0	255.255.255.0	U	50	0	0	eth0	
		0.0.0.0	192.168.100.30	0.0.0.0	UG	50	0	0	ipip0	업스트림
		192.168.20.20	192.168.20.20	255.255.255.255	UGH	20	0	0	eth0	업스트림
	MN	0.0.0.0	192.168.30.30	0.0.0.0	UG	20	0	0	eth0	

표 16. 예제2. Metric 값 설정 기준(수치는 예제에서 임의로 정함)

Destination	Gateway	Local Interface	Gateway	Metric	비고
Static or Dynamic	Static	-	YES	20	
Static or Dynamic	Static	YES	-	50	
Any	0.0.0.0	YES	-	100	

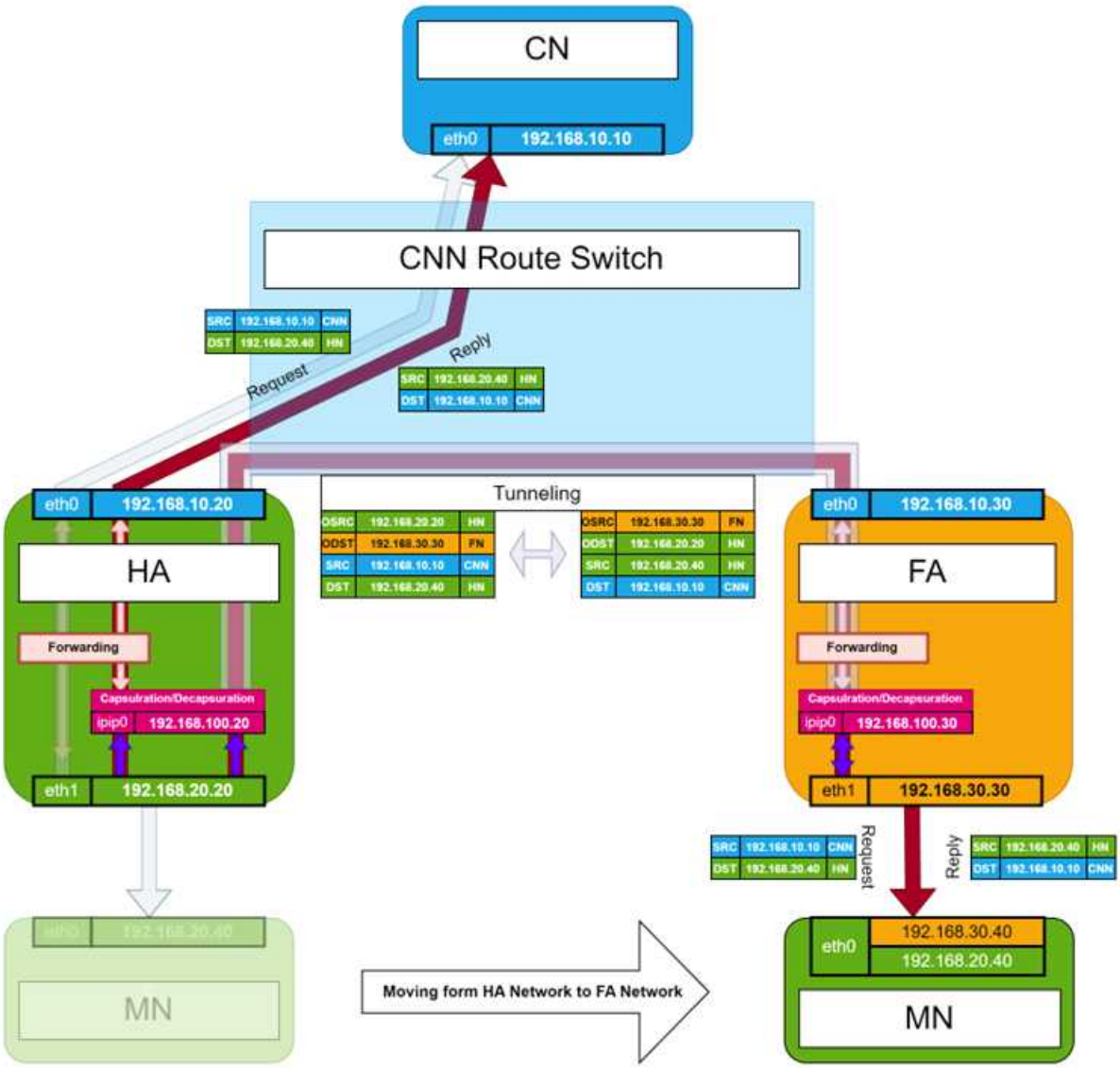

그림 14. 예제2 데이터그램 흐름도(CNN:CN Network, OSRC:Outer SRC, ODST: Outer DST)

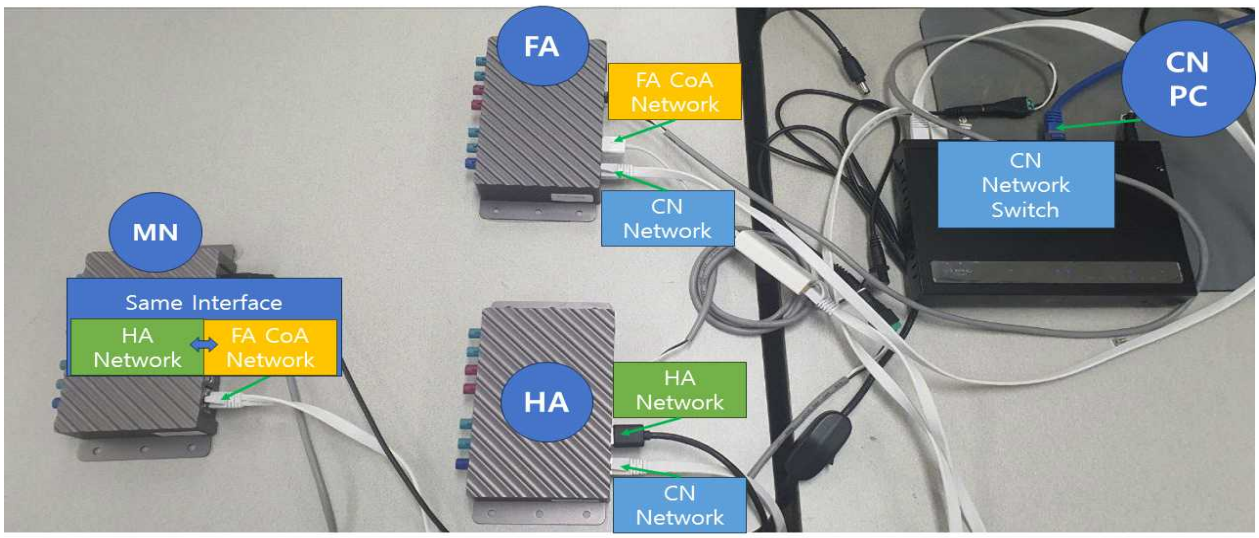

그림 15. 예제2 시험 구성 예시

2.2.1. HA와 FA의 Tunneling을 위한 설정(IP in IP)

2.2.1.1. HA(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip link add name ipip0 type ipip local 192.168.20.20 remote 192.168.30.30
2	ip address add ipip0 192.168.100.20
3	ip link set ipip0 up

줄	명령어 설명
1	‘IP in IP’ 캡슐화 타입의 터널링용 인터페이스를 ipip0라는 이름으로 추가 이때, 캡슐화 Outer Source는 192.168.20.20, Outer Destination은 192.168.30.30
2	터널링용 인터페이스 ipip0의 IP Address를 192.168.100.20으로 설정
3	터널링용 인터페이스 ipip0를 활성화

2.2.1.2. FA(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip link add name ipip0 type ipip local 192.168.30.30 remote 192.168.20.20
2	ip address add ipip0 192.168.100.30
3	ip link set ipip0 up

줄	명령어 설명
1	‘IP in IP’ 캡슐화 타입의 터널링용 인터페이스를 ipip0라는 이름으로 추가 이때, 캡슐화 Outer Source는 192.168.30.30 Outer Destination은 192.168.20.20 *FA의 ipip0 인터페이스가 HA에서 캡슐화된 데이터그램(패킷)을 수신해 캡슐을 제거하기 위해서 HA ipip0의 다운스트림에 업스트림 할 수 있는 Outer Source, Outer Destination IP Address 값을 설정해야 함 ** Outer Source, Outer Destination IP Address가 Any인 인터페이스도 캡슐화된 데이터그램(패킷)을 수신해 캡슐을 해제할 수 있음
2	터널링용 인터페이스 ipip0의 IP Address를 192.168.100.30으로 설정
3	터널링용 인터페이스 ipip0를 활성화

2.2.2. 노드 장치 별 라우팅 테이블 설정 방법

2.2.2.1. HA(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip route add 192.168.20.0/24 dev ipip0 metric 50
2	ip route add 192.168.30.30/32 via 192.168.30.30 dev eth0 metric 20 onlink

줄	명령어 설명
1	192.168.20.0~192.168.20.255이 Destination인 트래픽을 로컬 인터페이스 ipip0를 통해 외부로 전달하는 내용을 라우팅 테이블에 추가
2	192.168.30.30이 Destination인 트래픽을 로컬 인터페이스 eth0으로 192.168.30.30 게이트웨이를 통해 전달하는 내용을 라우팅 테이블에 추가

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         0.0.0.0         0.0.0.0         U     100    0        0 eth1
192.168.10.0    0.0.0.0         255.255.255.0   U     50     0        0 eth0
192.168.20.0    0.0.0.0         255.255.255.0   U     50     0        0 ipip0
192.168.30.30   192.168.30.30   255.255.255.255 UGH   20     0        0 eth0
```

그림 16. HA Routing table 설정 예시

2.2.2.2. MN(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip address add 192.168.30.40 dev eth0
2	ip route change default via 192.168.30.30 dev eth0 metric 20 onlink

줄	명령어 설명
1	인터페이스 eth0에 192.168.30.40인 IP Address를 추가
2	설정된 default 라우팅 테이블 설정을 모든 트래픽을 로컬 인터페이스 eth0으로 192.168.30.30 게이트웨이를 통해 전달하는 내용으로 변경

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.30.30   0.0.0.0         UG    20     0        0 eth0
```

그림 17. MN Routing table 설정 예시

2.2.2.3. FA(Linux(kernel 4.9.11))

ip 어플리케이션을 이용

줄	명령어
1	ip route add default dev eth1 metric 100
2	ip route add 192.168.10.0/24 dev eth0 metric 50
3	ip route add 192.168.20.20/32 via 192.168.20.20 dev eth0 metric 20 onlink
4	ip route add default via 192.168.100.30 dev ipip0 metric 50 table 20
5	ip rule add from 192.168.20.40/32 iif eth1 table 20

줄	명령어 설명
1	모든 트래픽을 로컬 인터페이스 eth1를 통해 외부로 전달하는 내용을 라우팅 테이블에 추가
2	192.168.10.0~192.168.10.255이 Destination인 트래픽을 로컬 인터페이스 eth0를 통해 외부로 전달하는 내용을 라우팅 테이블에 추가
3	192.168.20.20이 Destination인 트래픽을 로컬 인터페이스 eth0으로 192.168.20.20 게이트웨이를 통해 전달하는 내용을 라우팅 테이블에 추가
4	모든 트래픽을 192.168.100.30인 로컬 인터페이스 ipip0를 통해 외부로 전달 이때, table 20의 규칙을 따름
5	Destination이 192.168.20.40인 트래픽이 eth1으로 전달되었을 때만 table 20의 라우팅 테이블이 동작하는 내용을 라우트 테이블 필터(rule)에 추가

*192.168.10.10과 ssh 접속을 위해서 ip-rule을 사용

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         0.0.0.0         0.0.0.0         U     100    0        0 eth1
192.168.10.0    0.0.0.0         255.255.255.0   U     50     0        0 eth0
192.168.20.20   192.168.20.20   255.255.255.255 UGH   20     0        0 eth0
root@CONDOR5:~# ip route show table 20
default via 192.168.100.30 dev ipip0 metric 50
```

그림 18. FA Routing table 설정 예시(table 20의 Routing table 포함)

2.2.3. 터널링을 통해 전달하는 IPIP방식으로 캡슐화된 패킷 확인

HA의 인터페이스 eth0을 통하는 트래픽 데이터 덤프를 Wireshark 프로그램으로 구문분석해 IP in IP 캡슐화된 패킷을 확인

```
Frame 22: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Jan 31, 2021 12:42:33.947187000 대한민국 표준시
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1612064553.947187000 seconds
    [Time delta from previous captured frame: 0.041719000 seconds]
    [Time delta from previous displayed frame: 0.041719000 seconds]
    [Time since reference or first frame: 2.385527000 seconds]
    Frame Number: 22
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:icmp:data]
    [Coloring Rule Name: ICMP]
    [Coloring Rule String: icmp || icmpv6]
> Ethernet II, Src: ASUSTekC_1a:34:32 (38:d5:47:1a:34:32), Dst: 06:33:4e:d0:b4:c0 (06:33:4e:d0:b4:c0)
v Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.20.40
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xd80a (55306)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0xc333 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.10.10
    Destination Address: 192.168.20.40
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xafab [correct]
    [Checksum Status: Good]
    Identifier (BE): 3 (0x0003)
    Identifier (LE): 768 (0x0300)
    Sequence Number (BE): 40365 (0x9dad)
    Sequence Number (LE): 44445 (0xad9d)
  > [No response seen]
  v Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761626364656667686 9
      [Length: 32]
```

IP in IP 캡슐화 전

ICMP(ping) 패킷에 ICMP 패킷의 Source, Destination을 확인

수신시간(장치시간 기준) : 2021.01.31. 12:42:33.94718000

Control Message Protocol의 Sequence Number : 40365/4445

```
Frame 23: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
  Encapsulation type: Ethernet (1)
  Arrival Time: Jan 31, 2021 12:42:33.947241000 대한민국 표준시
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1612064553.947241000 seconds
  [Time delta from previous captured frame: 0.000054000 seconds]
  [Time delta from previous displayed frame: 0.000054000 seconds]
  [Time since reference or first frame: 2.385581000 seconds]
  Frame Number: 23
  Frame Length: 94 bytes (752 bits)
  Capture Length: 94 bytes (752 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:ip:icmp:data]
  [Coloring Rule Name: ICMP]
  [Coloring Rule String: icmp || icmpv6]
Ethernet II, Src: 06:33:4e:d0:b4:c0 (06:33:4e:d0:b4:c0), Dst: 6e:64:ae:3a:5b:79 (6e:64:ae:3a:5b:79)
Internet Protocol Version 4, Src: 192.168.20.20, Dst: 192.168.30.30
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0xa335 (41781)
> 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 127
  Protocol: IPIP (4)
  Header Checksum: 0xa4f1 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.20.20
  Destination Address: 192.168.30.30
Internet Protocol Version 4, Src: 192.168.10.10, Dst: 192.168.20.40
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 60
  Identification: 0xd80a (55306)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 127
  Protocol: ICMP (1)
  Header Checksum: 0xc433 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.10.10
  Destination Address: 192.168.20.40
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0xafab [correct]
  [Checksum Status: Good]
  Identifier (BE): 3 (0x0003)
  Identifier (LE): 768 (0x0300)
  Sequence Number (BE): 40365 (0x9dad)
  Sequence Number (LE): 44445 (0xad9d)
  [Response frame: 24]
v Data (32 bytes)
    Data: 6162636465666768696a6b6c6d6e6f707172737475767761626364656667686 9
    [Length: 32]
```

IP in IP 캡슐화 후

ICMP(ping) 패킷에 추가된 IPIP 프로토콜의 Source, Destination, 원래 ICMP 패킷의 Source, Destination을 확인

수신시간(장치시간 기준) : 2021.01.31. 12:42:33.94741000

Control Message Protocol의 Sequence Number : 40365/4445

2.2.4. 예제 2. 시험 결과

2.2.4.1. CN이 패킷을 전송하지 않을 때

```
Every 2.0s: iptables -L -nv                          CONDOR5: Sun Jan 31 03:33:27 2021

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   63  5464 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  eth0   *       192.168.30.30        192.168.20.20

Chain FORWARD (policy ACCEPT 5 packets, 200 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0            icmp --  eth0   ipip0   192.168.10.10        192.168.20.40
    0     0            icmp --  ipip0  eth0    192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 1 packets, 96 bytes)
 pkts bytes target     prot opt in     out     source               destination
   41 10968 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  *      eth0    192.168.20.20        192.168.30.30
```

HA

eth0에서 ipip0로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 증가하지 않음
eth0에서 송신하는 source 192.168.20.20, destination 192.168.30.30의 icmp(ping) 패킷 수가 증가하지 않음
eth0으로 수신하는 source 192.168.30.30, destination 192.168.20.30의 icmp(ping) 패킷 수가 증가하지 않음
ipip0에서 eth0로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 증가하지 않음

```
Every 2.0s: iptables -L -nv                          CONDOR5: Mon Jan 25 02:57:48 2021

Chain INPUT (policy ACCEPT 23508 packets, 1785K bytes)
 pkts bytes target     prot opt in     out     source               destination
 7761  530K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  eth0   *       192.168.20.20        192.168.30.30

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0            icmp --  ipip0  eth1    192.168.10.10        192.168.20.40
    0     0            icmp --  eth1   ipip0   192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 23633 packets, 1804K bytes)
 pkts bytes target     prot opt in     out     source               destination
 7712  691K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  *      eth0    192.168.30.30        192.168.20.20
```

FA

eth0으로 수신하는 source 192.168.20.20, destination 192.168.30.30의 icmp(ping) 패킷 수가 증가하지 않음
ipip0에서 eth1로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 증가하지 않음
eth1에서 ipip0로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 증가하지 않음
eth0에서 송신하는 source 192.168.30.30, destination 192.168.20.30의 icmp(ping) 패킷 수가 증가하지 않음

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Sat Jan 30 08:07:44 2021

Chain INPUT (policy ACCEPT 17228 packets, 1308K bytes)
 pkts bytes target     prot opt in     out     source               destination
 2925  120K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  eth0   *       192.168.30.30        192.168.20.20

Chain FORWARD (policy ACCEPT 3012 packets, 124K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0            icmp --  eth0   ipip0   192.168.10.10        192.168.20.40
    0     0            icmp --  ipip0  eth0    192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 17206 packets, 1310K bytes)
 pkts bytes target     prot opt in     out     source               destination
 2904  387K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     4    --  *      eth0    192.168.20.20        192.168.30.30
```

MN

eth0으로 수신하는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 증가하지 않음

eth0으로 송신되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 증가하지 않음

*iptables 어플리케이션을 이용한 ICMP 패킷 흐름 확인

2.2.4.2. CN이 패킷을 전송할 때

```
PS C:\Users[redacted] ping 192.168.20.40 -n 4

Pinging 192.168.20.40 with 32 bytes of data:
Reply from 192.168.20.40: bytes=32 time=3ms TTL=62
Reply from 192.168.20.40: bytes=32 time=1ms TTL=62
Reply from 192.168.20.40: bytes=32 time=2ms TTL=62
Reply from 192.168.20.40: bytes=32 time=1ms TTL=62

Ping statistics for 192.168.20.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 1ms
```

CN(192.168.10.10)

192.168.20.40으로 icmp(ping) 패킷 4개를 송수신 성공

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Sun Jan 31 03:32:28 2021

Chain INPUT (policy ACCEPT 29436 packets, 2235K bytes)
 pkts bytes target     prot opt in     out     source               destination
 9858  403K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   320 ACCEPT     4    --  eth0   *       192.168.30.30        192.168.20.20

Chain FORWARD (policy ACCEPT 10087 packets, 418K bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   240            icmp --  eth0   ipip0   192.168.10.10        192.168.20.40
    4   240            icmp --  ipip0  eth0    192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 29377 packets, 2234K bytes)
 pkts bytes target     prot opt in     out     source               destination
 9824 1260K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   320 ACCEPT     4    --  *      eth0    192.168.20.20        192.168.30.30
```

HA

eth0에서 ipip0로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 4개 증가

eth0에서 송신하는 source 192.168.20.20, destination 192.168.30.30의 icmp(ping) 패킷 수가 4개 증가

eth0으로 수신하는 source 192.168.30.30, destination 192.168.20.30의 icmp(ping) 패킷 수가 4개 증가

ipip0에서 eth0로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 4개 증가

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Mon Jan 25 22:01:09 2021

Chain INPUT (policy ACCEPT 28635 packets, 2175K bytes)
 pkts bytes target     prot opt in     out     source               destination
14555  876K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   320 ACCEPT     4    --  eth0   *       192.168.20.20        192.168.30.30

Chain FORWARD (policy ACCEPT 8 packets, 480 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   240            icmp --  ipip0  eth1    192.168.10.10        192.168.20.40
    4   240            icmp --  eth1   ipip0   192.168.20.40        192.168.10.10

Chain OUTPUT (policy ACCEPT 28814 packets, 2199K bytes)
 pkts bytes target     prot opt in     out     source               destination
14511 1429K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   320 ACCEPT     4    --  *      eth0    192.168.30.30        192.168.20.20
```

FA

eth0으로 수신하는 source 192.168.20.20, destination 192.168.30.30의 icmp(ping) 패킷 수가 4개 증가

ipip0에서 eth1로 전달되는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 4개 증가

eth1에서 ipip0로 전달되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 4개 증가
eth0에서 송신하는 source 192.168.30.30, destination 192.168.20.30의 icmp(ping) 패킷 수가 4개 증가

```
Every 2.0s: iptables -L -nv                                  CONDOR5: Sun Jan 31 03:37:43 2021

Chain INPUT (policy ACCEPT 31402 packets, 2397K bytes)
 pkts bytes target     prot opt in     out     source               destination
 5311  278K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   240 ACCEPT     icmp --  eth0   *       192.168.10.10        192.168.20.40

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 31402 packets, 2387K bytes)
 pkts bytes target     prot opt in     out     source               destination
 5293  519K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
    4   240 ACCEPT     icmp --  *      eth0    192.168.20.40        192.168.10.10
```

MN

eth0으로 수신하는 source 192.168.10.10, destination 192.168.20.40의 icmp(ping) 패킷 수가 4개 증가
eth0으로 송신되는 source 192.168.20.40, destination 192.168.10.10의 icmp(ping) 패킷 수가 4개 증가

*iptables 어플리케이션을 이용한 ICMP 패킷 흐름 확인