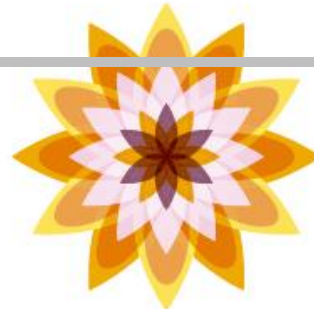


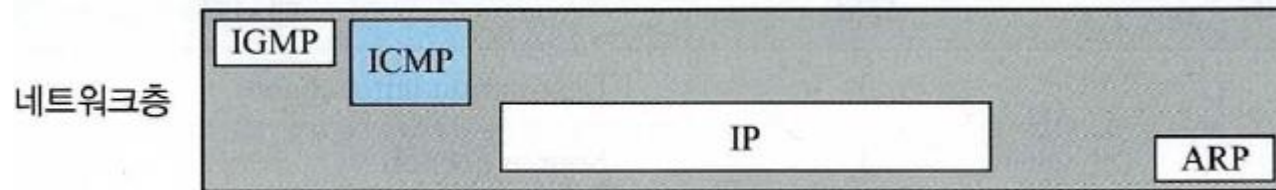
Chapter 09

인터넷 제어 메시지 프로토콜(ICMP)



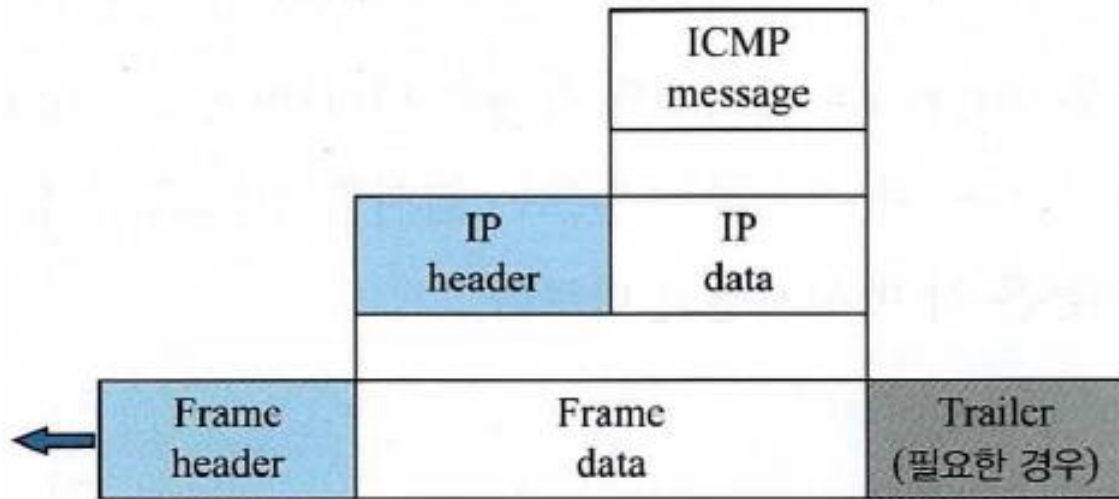
1. 개요

- IP 프로토콜은 오류 보고와 오류 수정 기능이 없다.
- IP 프로토콜은 호스트와 관리 질의(또는 조회)를 위한 메커니즘도 없다.
- 인터넷 제어 메시지 프로토콜(ICMP: Internet Control Message Protocol)은 위의 두 가지 단점을 보완하기 위해서 설계되었다.
- ICMP는 IP 프로토콜과 함께 동작하는 프로토콜이다.
- 그림은 네트워크 층에서 IP와 다른 프로토콜에 대한 ICMP의 위치를 보여준다.



1. 개요

- ICMP는 네트워크층 프로토콜이다.
- 그러나 이 프로토콜의 메시지는 예상과는 달리 직접 데이터 링크층으로 전달되지 않는다.
- 그 대신에 메시지는 데이터 링크층으로 가기 전에 IP 데이터그램 내에 캡슐화된다.



2. 메시지 유형

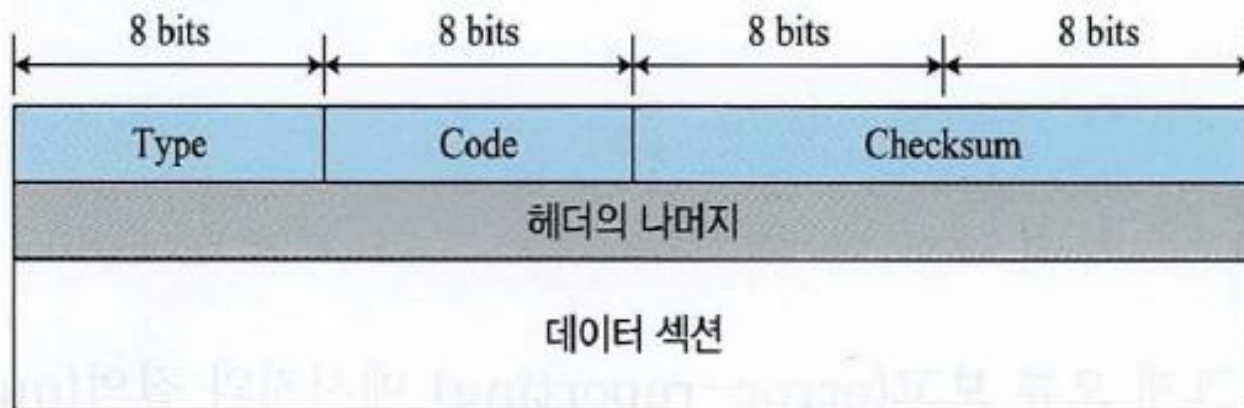
- ICMP 메시지는 크게 오류 보고(error-reporting) 메시지와 질의 (query) 메시지로 나눌 수 있다.
- 오류 보고 메시지는 라우터나 (목적지) 호스트가 IP 패킷을 처리하는 도중 발견되는 문제를 보고한다.
- 질의 메시지는 쌍으로 생성되는데 호스트나 네트워크 관리자가 라우터나 다른 호스트로부터 특정 정보를 획득하기 위해 사용한다.
- 표는 각 범주의 ICMP 메시지 유형을 정리한 것이다.

범주	유형	메시지
오류-보고 메시지	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
질의 메시지	8 또는 0	Echo request 또는 reply
	13 또는 14	Timestamp request 또는 reply

2. 메시지 유형

■ 메시지 형식

- ICMP 메시지는 8바이트의 헤더와 가변 길이의 데이터 부분으로 되어있다.
- 헤더의 일반 형식은 각 메시지 별로 다르지만 처음 4바이트는 전부 공통이다.
- 그림에서 보듯이 첫 번째 필드인 ICMP 유형 (type)은 메시지의 유형을 나타낸다.
- 코드(code) 필드는 특정 메시지 유형의 이유를 지정한다.



- 오류 메시지의 데이터 부분은 오류를 발생시킨 원래 패킷을 찾기 위한 정보를 전달한다.
- 질의 메시지에서 데이터 부분은 질의의 유형에 기초한 추가 정보를 전달한다.

2. 메시지 유형

■ 오류 보고 메시지

- ICMP의 주 임무 중 하나는 오류를 보고하는 것이다.
- IP는 신뢰성이 없는 프로토콜이다.
- ICMP는 이러한 단점을 보완하기 위하여 설계되었다.
- 그러나 ICMP는 오류를 수정하는 것이 아니고 단지 보고를 할 뿐이다.
- 오류 수정은 상위 계층 프로토콜에 맡긴다.
- ICMP는 발신지 IP 주소를 사용하여 오류 메시지를 데이터그램의 발신지로 보낸다.
- 다섯가지의 오류유형이 처리되는데, 이는 각각 목적지 도달 불가능, 발신지 억제, 시간 경과, 매개변수 문제 및 재지정이다.



2. 메시지 유형

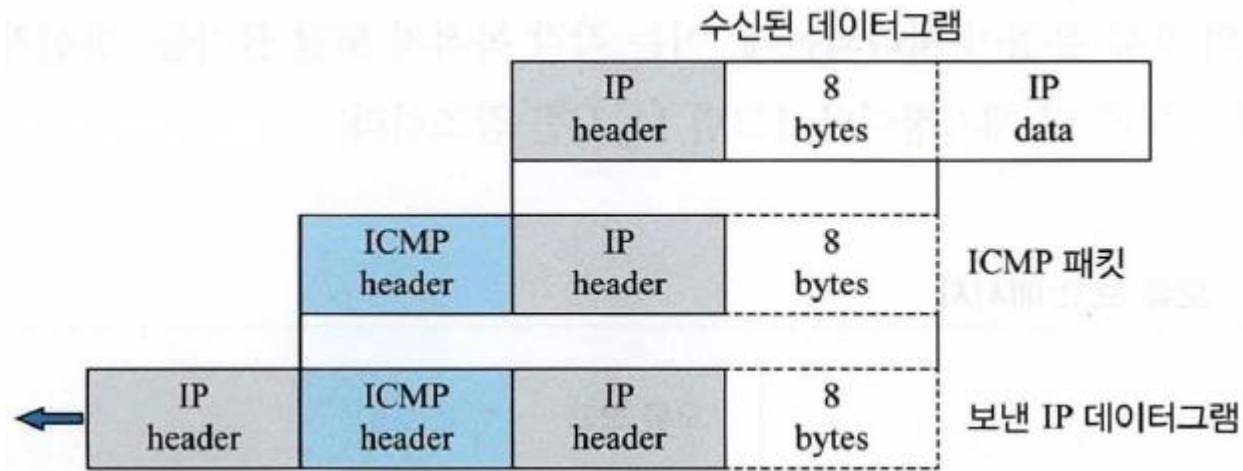
■ 오류 보고 메시지

- 다음은 ICMP 오류 메시지에 대한 중요한 사항들이다 .
 - ICMP 오류 메시지를 전달하는 데이터그램에 대해서는 ICMP 오류 메시지가 생성되지 않는다.
 - 처음 단편이 아닌 단편화된 데이터그램에 대해서는 ICMP 오류 메시지가 생성되지 않는다.
 - 멀티캐스트 주소를 가진 데이터그램에 대해서는 ICMP 오류 메시지가 생성되지 않는다.
 - 127.0.0.0 이나 0.0.0.0과 같은 특별한 주소를 가진 데이터그램에 대해서는 오류 메시지가 생성되지 않는다.
- 모든 오류 메시지의 데이터 부분에는 원래 데이터그램의 IP 헤더와 이 데이터그램의 데이터 중 처음 8바이트가 포함되어 있다.

2. 메시지 유형

■ 오류 보고 메시지

- ICMP는 오류 패킷을 생성하고 이것은 IP 데이터그램 내에 캡슐화된다.



2. 메시지 유형

■ 목적지 도달 불가능

- 라우터가 데이터그램을 전달할 수 없거나 호스트가 데이터그램을 전달할 수 없을 때 데이터그램은 폐기되고 라우터나 호스트는 데이터그램을 보냈던 발신지 호스트에게 목적지 도달 불가능(destination unreachable) 메시지를 보낸다.
- 그림은 목적지 도달 불가능 메시지의 형식을 보여준다.

Type: 3	Code: 0부터 15까지	Checksum
미사용(모두 0)		
IP 헤더와 데이터그램 데이터의 처음 8바이트가 포함된 수신된 IP 데이터그램 부분		

- 코드 0. 하드웨어 고장 등의 이유로 네트워크에 도달할 수 없다.
- 코드 1. 호스트에 도달할 수 없다.
- 코드 2. 프로토콜에 도달할 수 없다.
- 코드 3. 포트에 도달할 수 없다.
- 코드 4. 단편화가 필요하나 데이터그램의 DF(do not fragment) 필드가 설정되어 있다.
- 코드 5. 발신지 라우팅이 수행될 수 없다.
- 코드 6. 목적지 네트워크가 알려져 있지 않다.

2. 메시지 유형

■ 목적지 도달 불가능

- 코드 7. 목적지 호스트가 알려져 있지 않다.
 - 코드 8. 발신지 호스트가 고립되어 있다.
 - 코드 9. 목적지 네트워크와 통신이 관리상의 이유로 금지되어 있다.
 - 코드 10. 목적지 호스트와 통신이 관리상의 이유로 금지되어 있다.
 - 코드 11. 지정된 서비스 유형에 대해 네트워크에 도달할 수 없다.
 - 코드 12. 지정된 서비스 유형에 대해 호스트에 도달할 수 없다.
 - 코드 13. 관리자가 필터를 설치하여 호스트에 도달할 수 없다.
 - 코드 14. 호스트 우선순위가 위반되었기 때문에 호스트에 도달할 수 없다.
 - 코드 15. 우선순위가 상대적으로 높지 않아서 호스트에 도달할 수 없다.
-
- 코드 2와 3 메시지는 목적지 호스트에 의해서만 생성될 수 있다.
 - 나머지 코드들은 라우터에 의해서만 생성될 수 있다.

2. 메시지 유형

■ 발신지 억제

- ICMP의 발신지 억제 (source quench) 메시지는 IP에 흐름 제어와 혼잡 제어 기능과 유사한 기능을 추가하기 위하여 설계되었다.
- 혼잡으로 인해 데이터그램을 폐기하면, 라우터나 호스트는 데이터그램의 송신자에게 발신지 억제 메시지를 보낸다.
- 이 메시지는 두 가지 목적을 가지고 있다.
- 첫 번째로 데이터그램이 폐기되었음을 발신지에게 알린다.
- 두 번째로 경로상에 혼잡이 일어났고 발신지는 송신 과정을 천천히(또는 억제) 해야 한다는 것을 발신지에게 경고한다.
- 발신지 억제 메시지 형식은 그림과 같다.

Type: 4	Code: 0	Checksum
미사용(모두 0)		
IP 헤더와 데이터그램 데이터의 처음 8바이트가 포함된 수신된 IP 데이터그램 부분		

■ 시간 경과

- 시간 경과(time exceeded) 메시지는 다음의 두 경우에 생성된다.
 - 패킷을 수신할 다음 홉(다음 라우터)을 찾기 위해 라우터는 라우팅 테이블을 사용한다. 한 개 또는 그 이상의 라우팅 테이블에 오류가 있다면 패킷은 루프 또는 사이클을 지날 수 있다. 각 데이터그램은 이러한 상황에 대처하여 수명 (time to live) 필드를 가지고 있다. 그러나 데이터그램이 폐기될 때 라우터는 시간 경과 메시지를 원래의 발신지에 송신하여야 한다.
 - 둘째로 한 개의 메시지에 속하는 단편들이 정해진 시간 내에 목적지 호스트에 전부 도착하지 않은 경우에도 시간 경과 메시지가 생성된다. 첫 번째 단편이 도착하면 목적지 호스트는 타이머를 시작한다. 만약 타이머가 만료되었음에도 아직 모든 단편이 도착하지 않았다면 원래의 발신지에게 시간 경과 메시지를 보낸다.

2. 메시지 유형

■ 시간 경과

- 그림은 시간 경과 메시지 형식을 나타낸다.
- 코드 0은 수명 필드의 값이 0이 되어 데이터그램을 폐기하는 경우 사용한다.
- 코드 1은 단편의 일부가 정해진 시간 내에 도착하지 않아서 이미 도착한 단편을 폐기하는 경우 사용된다.

Type: 11	Code: 0 또는 1	Checksum
미사용(모두 0)		
IP 헤더와 데이터그램 데이터의 처음 8바이트가 포함된 수신된 IP 데이터그램 부분		

2. 메시지 유형

■ 매개변수 문제

- 만약 라우터나 목적지 호스트가 데이터그램의 필드에서 불명확하거나 빠진 값을 발견하게 되면 데이터그램을 폐기하고 매개변수 문제(parameter problem) 메시지를 발신자에게 보낸다.
- 그림은 매개변수 문제 메시지의 형식을 보여주고 있다.

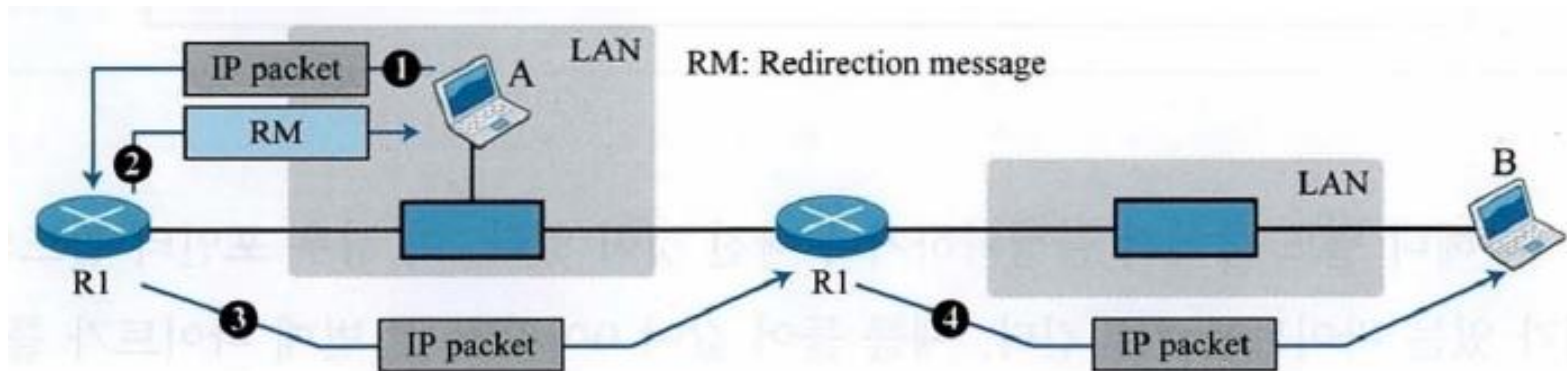
Type: 12	Code: 0 또는 1	Checksum
미사용(모두 0)		
IP 헤더와 데이터그램 데이터의 처음 8비트가 포함된 수신된 IP 데이터그램 부분		

- 코드 0. 헤더 필드 중에서 불명확하거나 빠진 것이 있다. 이 경우 포인터 필드의 값은 문제가 있는 바이트를 가리킨다.
- 코드 1. 옵션의 요구되는 부분이 빠졌다는 것이다. 이 경우 포인터는 사용되지 않는다.

2. 메시지 유형

■ 재지정

- 다른 네트워크로 가는 데이터그램을 보낼 때 호스트는 틀린 라우터에게 보낼 수 있다.
- 이 경우에 데이터그램을 받은 라우터는 데이터그램을 올바른 라우터에게 전송한다.
- 이때 호스트의 라우팅 테이블을 갱신하기 위해 호스트에게 재지정 (redirection) 메시지를 보낸다.
- 재지정 메시지의 개념은 그림에 나타나 있다.



2. 메시지 유형

■ 재지정

- 재지정 메시지 형식은 그림과 같다.

Type: 5	Code: 0부터 3까지	Checksum
타겟 라우터의 IP 주소		
IP 헤더와 데이터그램 데이터의 처음 8바이트가 포함된 수신된 IP 데이터그램 부분		

- 재지정 메시지의 코드 필드는 재지정을 제한한다.
 - 코드 0. 네트워크 지정 경로를 위한 재지정
 - 코드 1. 호스트지정 경로를 위한 재지정
 - 코드 2. 특정한 서비스 유형에 기초한 네트워크 지정 경로를 위한 재지정
 - 코드 3. 특정한 서비스 유형에 기초한 호스트 지정 경로를 위한 재지정

2. 메시지 유형

■ 질의 메시지

- 오류 보고 외에 ICMP는 네트워크 문제를 진단할 수도 있다.
- 이러한 기능은 질의(또는 조회) 메시지를 통하여 수행될 수 있다.
- 이 목적을 위해 다섯 쌍의 메시지가 설계되었지만 현재 이 중 세 쌍의 메시지는 사용되지 않고 있다.
- 현재 에코 요청과 응답, 타임스탬프 요청과 응답 메시지만 사용되고 있다.

■ 에코 요청과 응답

- 에코 요청과 에코 응답(echo request and reply) 메시지는 고장 진단의 목적으로 설계되었다.
- 에코 요청과 에코 응답 메시지는 IP 계층에서 통신이 되는지 결정하기 위하여 사용될 수 있다.
- ICMP 메시지는 IP 데이터그램에 의해 캡슐화되므로 에코 요청이 보내진 장치로부터 에코 응답이 왔다는 사실은 송신자와 수신자의 IP 프로토콜이 IP 데이터그램을 사용하여 서로 통신하고 있다는 것을 증명한다.

2. 메시지 유형

■ 에코 요청과 응답

- 에코 요청과 에코 응답 메시지를 사용하면 호스트가 다른 호스트에 도달할 수 있는지도 점검할 수 있다.
- 에코 요청과 에코 응답은 노드가 정상적으로 동작하고 있는지 검사할 수 있다.
- 검사될 호스트에 에코 요청 메시지를 보낸다.
- 에코 요청과 에코 응답은 노드가 정상적으로 동작하고 있는지 검사할 수 있다.
- 검사될 호스트에 에코 요청 메시지를 보낸다.

Type 8: Echo request

Type 0: Echo reply

Type: 8 또는 0	Code: 0	Checksum
Identifier		Sequence number
요청 메시지에 의해 보낸 옵션 데이터; 재전송 메시지에 의해 반복됨		

2. 메시지 유형

■ 타임스탬프 요청과 응답

- 두 시스템(호스트나 라우터)은 타임스탬프 요청과 응답(timestamp request and reply) 메시지를 사용하여 IP 데이터그램이 이 둘 사이를 지나가는 데 필요한 왕복 시간(round-trip time)을 결정할 수 있다.
- 이 메시지는 두 장치의 시계를 동기화하기 위해서도 사용될 수 있다.
- 이 두 메시지의 형식은 그림과 같다.

Type 13: request

Type 14: reply

Type: 13 또는 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

- 3 개의 타임스탬프 필드들은 각각 32 비트 길이를 가진다.
- 각 필드에는 그리니치 표준시(Greenwich Mean Time)라 불렸던 세계 표준시 (Universal Time) 자정부터의 시간을 밀리 초 단위로 표현한 값이 저장된다.

2. 메시지 유형

■ 타임스탬프 요청과 응답

- 발신지는 타임스탬프 요청 메시지를 생성한다.
- 발신지는 출발시각에서의 자신의 시계 값을 세계 표준시로 표현한 값을 원래의 타임스탬프(original timestamp) 필드에 삽입한다.
- 다른 두 타임스탬프 필드는 0으로 채워져 있다.
- 목적지는 타임스탬프 응답 메시지를 생성한다.
- 목적지는 요청 메시지에 있는 원래 타임스탬프 값을 응답 메시지의 같은 필드에 복사한다.
- 그런 다음 수신 타임스탬프(receive timestamp) 필드에는 요청이 수신된 시점에서의 자신의 시계 값을 세계 표준시로 표현하여 삽입한다.
- 마지막으로 전달 타임스탬프(transmit timestamp) 필드에는 응답 메시지가 출발하는 시점의 시계 값을 세계 표준시로 표현하여 삽입한다.

2. 메시지 유형

■ 타임스탬프 요청과 응답

- 타임스탬프 요청과 응답 메시지는 데이터그램이 발신지에서 목적지로 가서 다시 돌아오는 동안 걸리는 편도 또는 왕복 시간을 계산하는 데 사용될 수 있다.
- 공식은 다음과 같다.

송신 시간 = 수신 타임스탬프 - 원래의 타임스탬프

수신 시간 = 패킷이 돌아온 시간 - 전달 타임스탬프

왕복 시간 = 송신 시간 + 수신 시간

- 예를 들어 다음과 같은 정보가 주어졌다고 가정하자.

원래의 타임스탬프 값: 46

전달 타임스탬프 값: 60

수신 타임스탬프 값: 59

패킷 도착 시간: 67

- 왕복 시간은 다음과 같이 20ms로 계산될 수 있다.

송신 시간 = $59 - 46 = 13\text{ms}$

수신 시간 = $67 - 60 = 7\text{ms}$

왕복 시간 = $13 + 7 = 20\text{ms}$

2. 메시지 유형

■ 타임스탬프 요청과 응답

- 실제 편도 시간이 주어지면 다음 공식을 사용하고 타임스탬프 요청과 타임스탬프 응답 메시지를 사용하여 두 시계를 동기화시킬 수 있다.

$$\text{시간 차} = \text{수신 타임스탬프} - (\text{원래의 타임스탬프 필드} + \text{편도 시간 구간})$$

- 편도 시간 구간은 (만약 송신 시간과 수신 시간이 같다고 할 수 있으면) 왕복 시간을 둘로 나누어 구할 수도 있고 또는 다른 방법으로 구할 수 있다.
- 예를 들어 바로 앞의 예에서 두 시계는 3ms 차이가 있다는 것을 알 수 있다.

$$\text{시간 차} = 59 - (46 + 10) = 3$$

2. 메시지 유형

■ 검사합

- ICMP에서 검사합은 헤더와 데이터를 포함하는 전체 메시지에 대해 계산된다.

■ 검사합 계산

- 송신자는 1의 보수 연산을 사용하여 다음의 단계를 수행한다.
 - 검사합 필드를 0으로 만든다.
 - 헤더와 데이터에 대해 16비트 단어의 합을 구한다.
 - 합의 보수를 취하여 검사합을 구한다.
 - 검사합을 검사합 필드에 저장한다.

■ 검사합 검사

- 수신자는 1의 보수 연산을 사용하여 다음의 단계를 수행한다.
 - 헤더와 데이터에 대해 16비트 단어의 합을 구한다.
 - 합의 보수를 구한다.
 - 전 단계의 결과가 16개의 0이면 메시지는 받아들여지고 그렇지 않으면 거절한다.

2. 메시지 유형

■ 검사합 계산

- 예제. 그림은 간단한 에코 요청 메시지에 대해 검사합을 계산하는 예를 보여주고 있다.
- 임의로 식별자는 1이고 순서 번호는 9라고 정한다.
- 메시지는 16 비트(2바이트)의 단어들로 분할된다.
- 이 16비트 단어를 전부 더하고 그 합의 보수를 구한다.
- 송신자는 이 값을 검사합 필드에 넣을 수 있다.

8	0	0	
1	9		
TEST			
8 & 0	→	00001000	00000000
0	→	00000000	00000000
1	→	00000000	00000001
9	→	00000000	00001001
T & E	→	01010100	01000101
S & T	→	01010011	01010100
Sum	→	10101111	10100011
Checksum	→	01010000	01011100

3. 디버깅 도구

■ ping

- ping 프로그램을 사용하여 호스트가 정상적으로 동작하고 있으며 응답하고 있는지 점검할 수 있다.
- 발신지 호스트는 ICMP 에코 요청 메시지(유형 : 8, 코드: 0)를 보내고 정상적으로 동작하는 목적지는 ICMP 에코 응답 메시지로 응답한다.
- ping 프로그램은 에코 요청과 응답 메시지의 식별자 필드 값을 설정하고 순서번호는 0부터 시작한다.
- ping은 왕복 시간(round-trip time)도 계산할 수 있다.
- ping 프로그램을 사용하여 amazon.com 서버를 테스트한 결과는 다음과 같다.

```
C:\Users\jinu>ping -t amazon.com

Ping amazon.com [54.239.28.85] 32바이트 데이터 사용:
54.239.28.85의 응답: 바이트=32 시간=190ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=190ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=193ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=190ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=190ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=199ms TTL=229
54.239.28.85의 응답: 바이트=32 시간=217ms TTL=229

54.239.28.85에 대한 Ping 통계:
    패킷: 보냄 = 7, 받음 = 7, 손실 = 0 (0% 손실),
    왕복 시간(밀리초):
        최소 = 190ms, 최대 = 217ms, 평균 = 195ms
Control-C
^C
C:\Users\jinu>
```

3. 디버깅 도구

■ ping

- ping 프로그램의 두 번째 예에서는 mail.google.com이라는 메일 서버가 정상적으로 동작하고 있는지 확인하고자 하며 결과는 다음과 같다.

```
C:\Windows\System32>ping -t mail.google.com

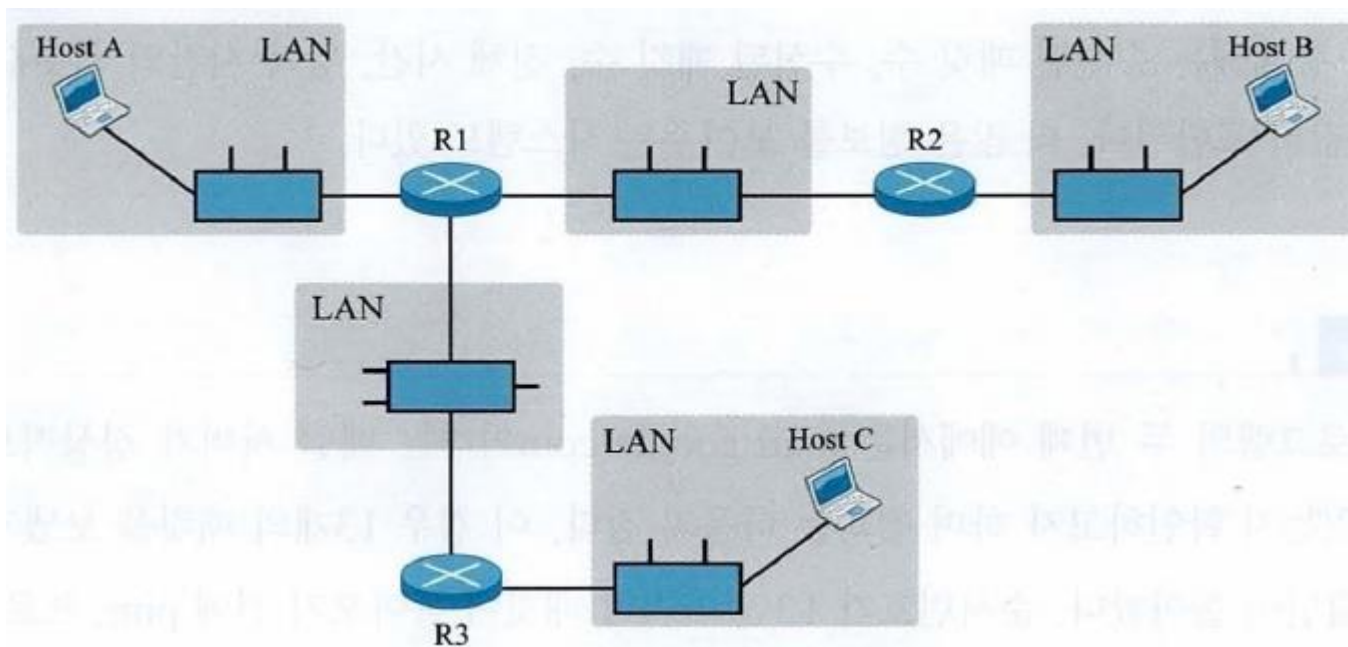
Ping mail.google.com [172.217.161.197] 32바이트 데이터 사용:
172.217.161.197의 응답: 바이트=32 시간=34ms TTL=57
172.217.161.197의 응답: 바이트=32 시간=37ms TTL=57
172.217.161.197의 응답: 바이트=32 시간=37ms TTL=57
172.217.161.197의 응답: 바이트=32 시간=35ms TTL=57
172.217.161.197의 응답: 바이트=32 시간=35ms TTL=57
172.217.161.197의 응답: 바이트=32 시간=69ms TTL=57

172.217.161.197에 대한 Ping 통계:
    패킷: 보냄 = 6, 받음 = 6, 손실 = 0 (0% 손실),
왕복 시간(밀리초):
    최소 = 34ms, 최대 = 69ms, 평균 = 41ms
Control-C
^C
C:\Windows\System32>
```

3. 디버깅 도구

■ tracert(UNIX의 경우 traceroute)

- 윈도우즈의 tracert나 UNIX의 traceroute 프로그램을 사용하여 패킷이 발신지에서 목적지까지 전달되는 경로를 추적할 수 있다.
- tracert 프로그램의 개념을 그림을 사용하여 설명한다.



■ **tracert(UNIX의 경우 traceroute)**

- 이 프로그램은 시간 경과 메시지와 목적지 도달 불가능 메시지를 사용한다.
- **tracert** 프로그램은 ICMP 메시지와 IP 패킷 내의 TTL 필드를 사용하여 이 경로를 찾는다.
 - tracert 프로그램은 라우터 R1의 주소와 호스트 A와 라우터 R1사이의 왕복 시간을 찾는다.
 - tracert 프로그램은 동일한 방법으로 라우터 R2의 주소와 호스트 A와 라우터 R2 사이의 왕복 시간을 구한다.
 - 그러나 이번에는 TTL 값이 2로 설정되어 라우터 R1은 패킷을 전달하지만 라우터 R2는 패킷을 폐기하고 시간 경과 ICMP 메시지를 보낸다.
 - tracert 프로그램은 앞의 과정을 반복함으로써 호스트 B의 주소와 호스트 A와 호스트 B 사이의 왕복 시간을 구한다.
 - 호스트 B가 패킷을 받으면 TTL을 감소한다.
 - 그러나 패킷이 최종 목적지에 도달하였으므로 패킷은 폐기되지 않는다.
 - 어떻게 호스트 A에게 ICMP 메시지가 보내질 수 있는가?
 - 여기에서 tracert 프로그램은 다른 방법을 사용한다.
 - UDP 패킷 내의 포트 번호는 UDP 프로토콜이 지원하지 않는 번호로 설정된다.

■ **tracert(UNIX의 경우 traceroute)**

- **tracert 프로그램은 ICMP 메시지와 IP 패킷 내의 TTL 필드를 사용하여 이 경로를 찾는다**
 - 호스트 B가 패킷을 받으면 이 패킷을 받을 응용 프로그램을 찾을 수 없으므로 패킷을 폐기하고 유형값 3과 코드값 3의 목적지 도달 불가 ICMP 메시지를 호스트 A에게 보낸다.
 - 라우터에서 UDP 헤더를 점검하지 않으므로 라우터 R1과 R2에서는 이 상황이 발생하지 않는다.
 - tracert 프로그램은 도착한 IP 데이터그램의 발신지 주소를 기록하고 왕복 시간을 계산한다.
 - 코드 3의 목적지 도달 불가능 메시지가 수신되었다면 경로가 발견되었고 더 이상 패킷을 보낼 필요가 없음을 알 수 있다.

3. 디버깅 도구

■ tracert(UNIX의 경우 traceroute)

- tracert 프로그램을 사용하여 현재 사용 중인 컴퓨터와 www.naver.com 서버 사이의 경로를 찾은 결과는 다음과 같다.
- 이 경우 전체 경로를 찾는데 시간이 많이 걸린다.
- 중간에 tracert가 5초 이내에 응답을 받지 못하면 문제가 있음을 알리기 위하여 "*"를 출력하고 다음 홑을 시도한다.

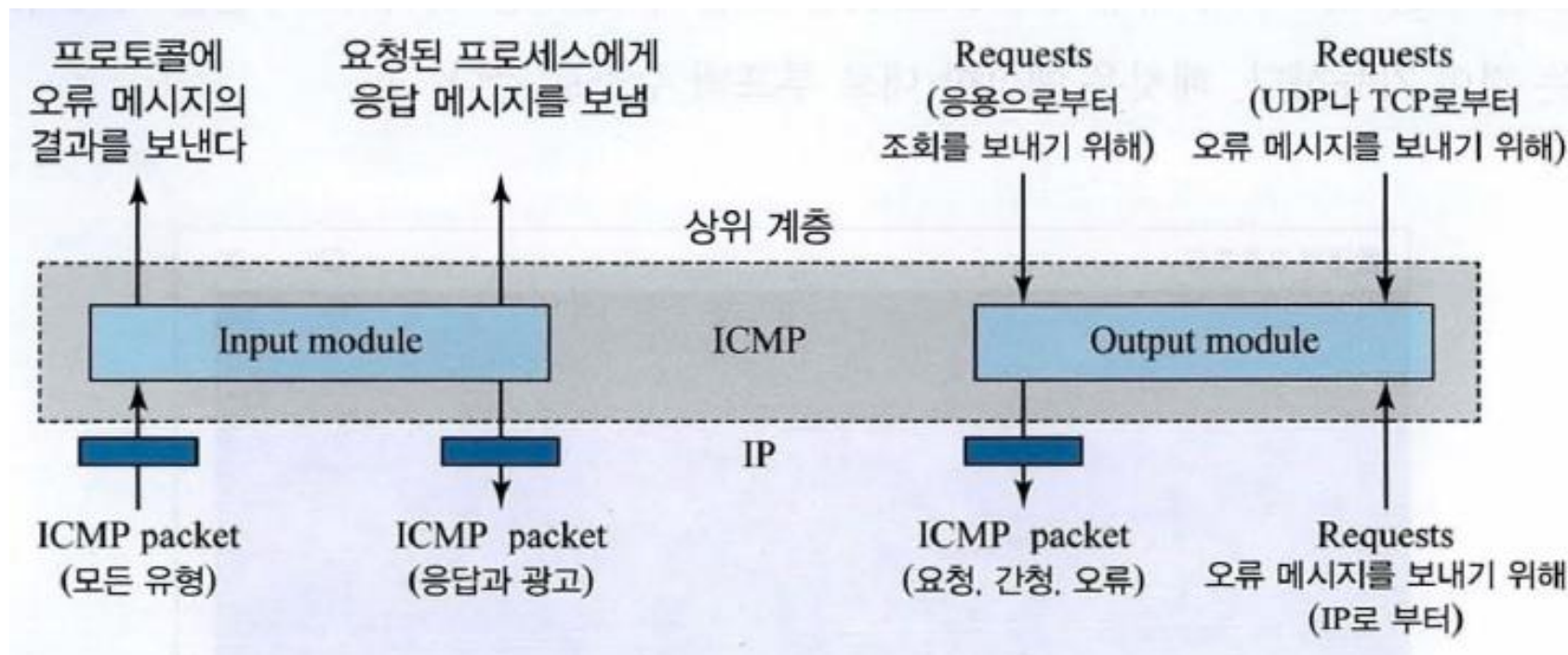
```
C:\Windows\System32>tracert www.naver.com

최대 30홑 이상
www.naver.com.nheos.com [223.130.200.107](으)로 가는 경로 추적:

 1      *          *          *      요청 시간이 만료되었습니다.
 2      18 ms      2 ms      8 ms     112.190.107.121
 3       2 ms      2 ms      2 ms     112.190.97.233
 4      *          *          *      요청 시간이 만료되었습니다.
 5      *          *          *      요청 시간이 만료되었습니다.
 6      *          *          *      요청 시간이 만료되었습니다.
 7      *          *          *      요청 시간이 만료되었습니다.
 8      *          *          *      요청 시간이 만료되었습니다.
 9      *          *          *      요청 시간이 만료되었습니다.
10     *          *          *      요청 시간이 만료되었습니다.
11     *          *          *      요청 시간이 만료되었습니다.
12     *          *          *      요청 시간이 만료되었습니다.
13     *          *          *      요청 시간이 만료되었습니다.
14     *          *          *      요청 시간이 만료되었습니다.
15     *          *          ^C
```

4. ICMP 패키지

- 그림은 ICMP 패키지의 입력 모듈과 출력 모듈을 보여주고 있다.



4. ICMP 패키지

■ 입력 모듈

- 입력 모듈(input module)은 수신된 모든 ICMP를 처리한다.
- ICMP 패킷이 IP 계층으로부터 전달되면 수행된다.
- 의사코드는 표와 같다.

1	ICMP_input_module (ICMP_Packet)
2	{
3	If (the type is a request)
4	{
5	Create a reply
6	Send the reply
7	}
8	If (the type defines a redirection)
9	{
10	Modify the routing table
11	}

4. ICMP 패키지

■ 입력 모듈

- 입력 모듈(input module)은 수신된 모든 ICMP를 처리한다.
- ICMP 패킷이 IP 계층으로부터 전달되면 수행된다.
- 의사코드는 표와 같다.

12	If (the type defines other、 error messages)
13	{
14	Inform the appropriate source protocol
15	}
16	Return
17	}

■ 출력 모듈

- 출력 모듈(output module)은 상위 계층 또는 IP 프로토콜에 의해 요청된 요청(request), 간청(solicitation), 오류 메시지를 생성한다.
- 이 모듈은 IP, UDP, 또는 TCP로부터 ICMP 오류 메시지 중 하나를 보내달라는 요구를 받는다.
- 다음의 네 경우(ICMP 메시지를 배달하는 IP 패킷, 단편화된 IP 패킷, 멀티캐스트 IP 패킷, 0.0.0.0이나 127.X.Y.Z와 같은 IP 주소를 가지는 IP 패킷)에는 ICMP 메시지가 생성될 수 없다는 것을 기억해야 한다.
- 출력 모듈은 응용 프로그램으로부터 ICMP 요청 메시지 중 하나를 보내달라는 요청을 받을 수 있다.

4. ICMP 패키지

■ 출력 모듈

- 의사코드는 표와 같다.

1	ICMP_Output_Module (demand)
2	{
3	If (the demand defines an error message)
4	{
5	(demand comes from IP AND is forbidden)
6	{
7	Return
8	}
9	If (demand is a valid r、edirection message)
10	{
11	Return
12	}

4. ICMP 패키지

■ 출력 모듈

- 의사코드는 표와 같다.

13	Create an error message
14	If (demand defines a request)
15	{
16	Create a request message
17	}
18	Send the message
19	Return
20	}

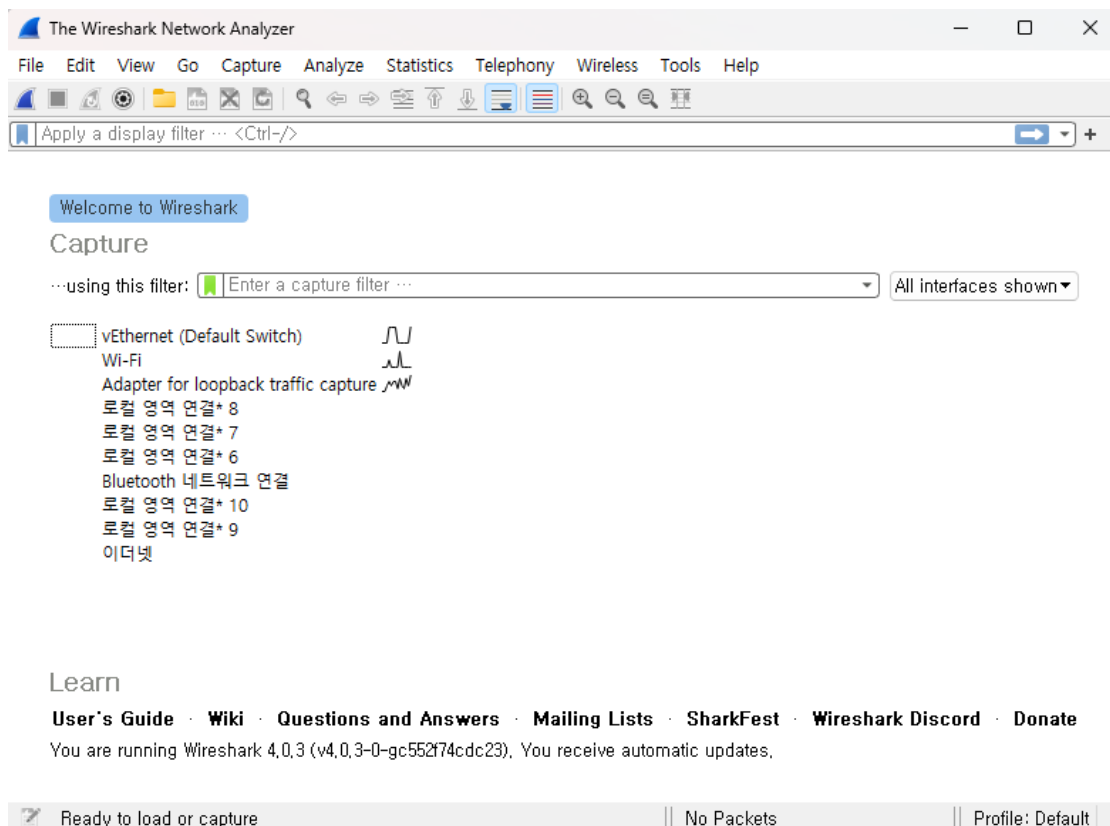
5. ping 명령어를 이용한 ICMP 덤프 분석

- 네트워크에 트러블슈팅 (trouble shooting, 문제점 해결)을 위해 가장 많이 사용하고 있는 명령어가 ping이다.
- 여기서 트러블슈팅의 기본 명령어인 ping에 대해서 그 동작과 프로토콜의 상세한 내용을 실제로 패킷을 캡처하면서 확인해 보도록 한다.
- 확인 순서는 다음과 같다.
 - 와이어샤크를 실행하여 패킷을 캡처한다.
 - 명령 프롬프트에서 ping 명령을 실행하고 그 결과를 확인한다.
 - 패킷 캡처를 정지하고 내용을 확인한다.

5. ping 명령어를 이용한 ICMP 덤프 분석

■ 패킷 캡처 개시와 ping 명령어 실행

- ping 명령어를 실행하기 전에 먼저 패킷을 캡처한다.
- 와이어샤크를 실행시켜 시작 화면의 [Interface List]에서 사용하고 있는 LAN 카드의 링크를 클릭하여 패킷 캡처를 시작한다.



5. ping 명령어를 이용한 ICMP 덤프 분석

■ 패킷 캡처 개시와 ping 명령어 실행

- 여기서 ping 명령어를 실행하여 그 결과를 확인해 보자.
- 명령 프롬프트에서 `ping -t www.amazon.com` 을 실행시킨다.

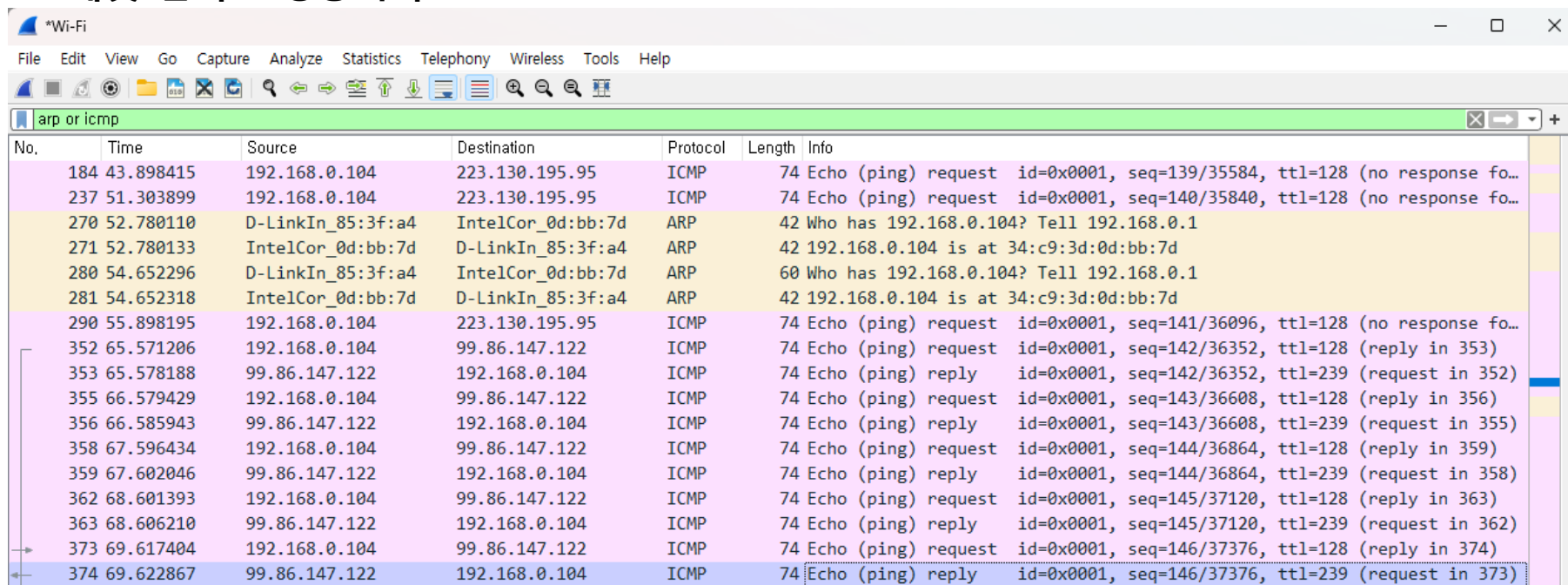
```
C:\Windows\System32>ping -t www.amazon.com

Ping d3ag4hukkh62yn.cloudfront.net [99.86.147.122] 32바이트 데이터 사용:
99.86.147.122의 응답: 바이트=32 시간=7ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=6ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=4ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=8ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=7ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=7ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=5ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=7ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=7ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=6ms TTL=239
99.86.147.122의 응답: 바이트=32 시간=14ms TTL=239
```

5. ping 명령어를 이용한 ICMP 덤프 분석

■ 패킷 캡처 정지와 내용 확인

- 메인 화면에 있는 툴바의 [Stop] 버튼을 클릭하여 패킷 캡처를 정지한다.
- 이때 화면에는 ping 패킷뿐만 아니라 같은 시간에 인터페이스를 통해 송/수신한 다른 패킷도 나타난다.
- 따라서 필터 디스플레이 기능을 사용하여 ping에 이용되는 ICMP 패킷만으로 제한한다.
- 필터 툴바의 텍스트 상자에 [arp or icmp]라고 입력하고 엔터키를 누른다.
- 패킷 목록 정보의 [protocol] 열에 맨 먼저 ARP 패킷이 나타난 다음 ICMP가 이어지면 패킷 캡처는 성공이다.



No.	Time	Source	Destination	Protocol	Length	Info
184	43.898415	192.168.0.104	223.130.195.95	ICMP	74	Echo (ping) request id=0x0001, seq=139/35584, ttl=128 (no response fo...
237	51.303899	192.168.0.104	223.130.195.95	ICMP	74	Echo (ping) request id=0x0001, seq=140/35840, ttl=128 (no response fo...
270	52.780110	D-LinkIn_85:3f:a4	IntelCor_0d:bb:7d	ARP	42	Who has 192.168.0.104? Tell 192.168.0.1
271	52.780133	IntelCor_0d:bb:7d	D-LinkIn_85:3f:a4	ARP	42	192.168.0.104 is at 34:c9:3d:0d:bb:7d
280	54.652296	D-LinkIn_85:3f:a4	IntelCor_0d:bb:7d	ARP	60	Who has 192.168.0.104? Tell 192.168.0.1
281	54.652318	IntelCor_0d:bb:7d	D-LinkIn_85:3f:a4	ARP	42	192.168.0.104 is at 34:c9:3d:0d:bb:7d
290	55.898195	192.168.0.104	223.130.195.95	ICMP	74	Echo (ping) request id=0x0001, seq=141/36096, ttl=128 (no response fo...
352	65.571206	192.168.0.104	99.86.147.122	ICMP	74	Echo (ping) request id=0x0001, seq=142/36352, ttl=128 (reply in 353)
353	65.578188	99.86.147.122	192.168.0.104	ICMP	74	Echo (ping) reply id=0x0001, seq=142/36352, ttl=239 (request in 352)
355	66.579429	192.168.0.104	99.86.147.122	ICMP	74	Echo (ping) request id=0x0001, seq=143/36608, ttl=128 (reply in 356)
356	66.585943	99.86.147.122	192.168.0.104	ICMP	74	Echo (ping) reply id=0x0001, seq=143/36608, ttl=239 (request in 355)
358	67.596434	192.168.0.104	99.86.147.122	ICMP	74	Echo (ping) request id=0x0001, seq=144/36864, ttl=128 (reply in 359)
359	67.602046	99.86.147.122	192.168.0.104	ICMP	74	Echo (ping) reply id=0x0001, seq=144/36864, ttl=239 (request in 358)
362	68.601393	192.168.0.104	99.86.147.122	ICMP	74	Echo (ping) request id=0x0001, seq=145/37120, ttl=128 (reply in 363)
363	68.606210	99.86.147.122	192.168.0.104	ICMP	74	Echo (ping) reply id=0x0001, seq=145/37120, ttl=239 (request in 362)
373	69.617404	192.168.0.104	99.86.147.122	ICMP	74	Echo (ping) request id=0x0001, seq=146/37376, ttl=128 (reply in 374)
374	69.622867	99.86.147.122	192.168.0.104	ICMP	74	Echo (ping) reply id=0x0001, seq=146/37376, ttl=239 (request in 373)

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- ICMP(Internet Control Message Protocol)는 IP의 접속 시험을 하기 위한 프로토콜이다.
- ICMP에는 에코 요청, 에코 응답 등의 메시지 종류가 정의되어 있으며, ping을 비롯하여 네트워크 상태 확인을 위해 이용된다.
- 여기서는 앞서 캡처한 패킷 가운데 패킷 목록 정보의 [Info] 열에서 [Echo(ping) request], [Echo (ping) reply]라고 표시된 2개의 패킷에 대하여 덤프 분석해 보자.
- 먼저 ICMP 에코 (ping) 요청 패킷을 전개하면 그림과 같다.

```
> Frame 352: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF{...}
> Ethernet II, Src: IntelCor_0d:bb:7d (34:c9:3d:0d:bb:7d), Dst: D-LinkIn_85:3f:a4 (c4:00:11:85:3f:a4)
> Internet Protocol Version 4, Src: 192.168.0.104, Dst: 99.86.147.122
▼ Internet Control Message Protocol
  ① Type: 8 (Echo (ping) request)
  ② Code: 0
  ③ Checksum: 0x4ccd [correct]
  ④ [Checksum Status: Good]
  ⑤ Identifier (BE): 1 (0x0001)
  ⑥ Identifier (LE): 256 (0x0100)
  ⑦ Sequence Number (BE): 142 (0x008e)
  ⑧ Sequence Number (LE): 36352 (0x8e00)
  ⑨ \[Response frame: 353\]
    > Data (32 bytes)
```

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- ICMP 에코 (ping) 응답 패킷을 전개하면 그림과 같다.

```
> Frame 353: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \
> Ethernet II, Src: D-LinkIn_85:3f:a4 (c4:a8:1d:85:3f:a4), Dst: IntelCor_0d:bb:7d (34
> Internet Protocol Version 4, Src: 99.86.147.122, Dst: 192.168.0.104
v Internet Control Message Protocol
  ① Type: 0 (Echo (ping) reply)
  ② Code: 0
  ③ Checksum: 0x54cd [correct]
  ④ [Checksum Status: Good]
  ⑤ Identifier (BE): 1 (0x0001)
  ⑥ Identifier (LE): 256 (0x0100)
  ⑦ Sequence Number (BE): 142 (0x008e)
  ⑧ Sequence Number (LE): 36352 (0x8e00)
  ⑨ [Request frame: 352]
  [Response time: 6.982 ms]
  > Data (32 bytes)
```

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- 맨 위에 나타난 ① [Type] 필드는 ICMP 패킷의 유형을 나타낸다.
- 여기서 352 번 프레임의 ICMP 에코 (ping) 요청 패킷을 확인해 보자.
- 이번에는 에코 요청이므로 “8”이 들어가 있다.
- 또한, 다음 353 번 프레임의 ICMP 에코 (ping) 응답 패킷은 에코 응답이므로 “0”이 들어가 있다.
- 에코(echo)란 메아리와 같은 뜻으로 이야기한 내용이 그대로 되돌아오므로 에코 테스트를 말한다.
- 에코 요청 에코 응답과 유형 값의 관계는 표와 같다.

메시지	유형 값
에코 요청	8
에코 응답	0

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- 다음 ② [Code] 필드는 이번에는 에코 테스트이므로 352번 프레임, 353번 프레임 모두 코드 "0"이 들어간다.
- ping은 에코 요청과 에코 응답 2개의 패킷을 교환함으로써 접속 가능성 확인을 수행한다
- 또한 이어지는 ③ [Checksum] 필드에는 검사합이 2바이트 들어간다.
- 이것은 ICMP 메시지가 도중에 변경되지 않았는지 데이터 내용을 확인하기 위해 준비된 필드이다.
- 이 필드는 에코 요구와 에코 응답 모두 [correct]로 되어 있다.
- 또한 이 뒤에 2바이트의 ⑤, ⑥ [Identifier]가 들어간다.
- 이것은 ICMP 메시지가 여러 개 있을 경우에 각각을 구별하기 위한 식별자 필드이다.
- 이어서 2바이트의 ⑦, ⑧ [Sequence number] 필드가 들어간다.
- 이 필드에는 연속하여 ping을 송신할 때에 각각의 ping을 구별하기 위한 순서번호가 들어간다.
- 이번에는 에코 요구와 에코 응답으로 [Sequence number]가 일치하고 있는 점에 주의하기 바란다.

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- 또한 [Identifier]와 [Sequence number] 필드는 비트 열의 상위 자리에서 하위 자리 순으로 표현하는 BE(Big Endian)와 하위 자리에서 상위 자리 순으로 비트를 표현하는 LE(Little Endian) 방식으로 나타낸다.
- 보통 TCP/IP 통신에서는 BE 방식이 이용된다.
- 이어서 와이어샤크가 추가한 헤더로서 ⑨ [Request Frame : 번호]에 대응하는 ICMP 에코 요청과 ICMP 에코 응답 패킷 번호의 링크를 제공함과 동시에 ICMP 응답의 경우에는 [Response Time]으로서 ping의 패킷 왕복 지연 시간(RTT)을 나타낸다.
- 그리고 마지막 [Data] 필드에 ICMP 에코 데이터가 들어간다.
- 일반적으로 ping에서 이용하는 메시지의 초기 길이는 윈도우즈에서는 32바이트 UNIX/Linux 등에서는 64바이트이다.
- 또한 옵션을 지정하여 메시지 길이를 바꿀 수도 있다.

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- 또, ping과 마찬가지로 네트워크층에서 접속을 확인하는 명령어로 tracert가 있다.
- tracert 명령어는 일반적으로 [tracert]라고 하며 Linux와 UNIX에서는 traceroute 명령어로 되어 있다.
- 또 라우터나 계층3 스위치 등에도 있는 명령어이다.
- tracert 명령어는 타겟 서버까지 중계하는 라우터의 IP 주소나 이름과 그 응답 시간을 순차적으로 기록해 나감으로써 타겟 서버까지 라우터 목록을 보여준다.
- 이 명령어를 이용함으로써 어떤 라우터를 거쳐 도착했는가 하는 네트워크의 경로를 확인할 수 있다.
- 또 각 경유지의 지연 시간을 비교함으로써 어느 라우터나 통신 회선에서 시간이 걸렸는가 하는 이른바 병목현상을 발견할 수도 있다.

5. ping 명령어를 이용한 ICMP 덤프 분석

■ ICMP 덤프 분석

- 그림을 보면 클라이언트 PC 부터 목표로 삼은 www.amazon.com (23.207.177.171)까지는 9 대의 라우터(또는 계층3 스위치 등)를 경유하여 도착한 것임을 알 수 있다.

```
C:\Windows\System32>tracert www.amazon.com
```

최대 30홉 이상의

e15316.dsca.akamaiedge.net [23.207.177.171](으)로 가는 경로 추적:

1	3 ms	*	*	210.179.239.254
2	2 ms	3 ms	2 ms	112.190.107.121
3	3 ms	2 ms	2 ms	112.190.110.209
4	*	*	*	요청 시간이 만료되었습니다.
5	*	*	*	요청 시간이 만료되었습니다.
6	4 ms	4 ms	3 ms	112.191.127.19
7	3 ms	3 ms	3 ms	112.191.127.201
8	10 ms	7 ms	9 ms	221.168.64.170
9	25 ms	48 ms	2 ms	a23-207-177-171.deploy.static.akamaitechnologies.com [23.207.177.171]

추적을 완료했습니다.



Thank You
