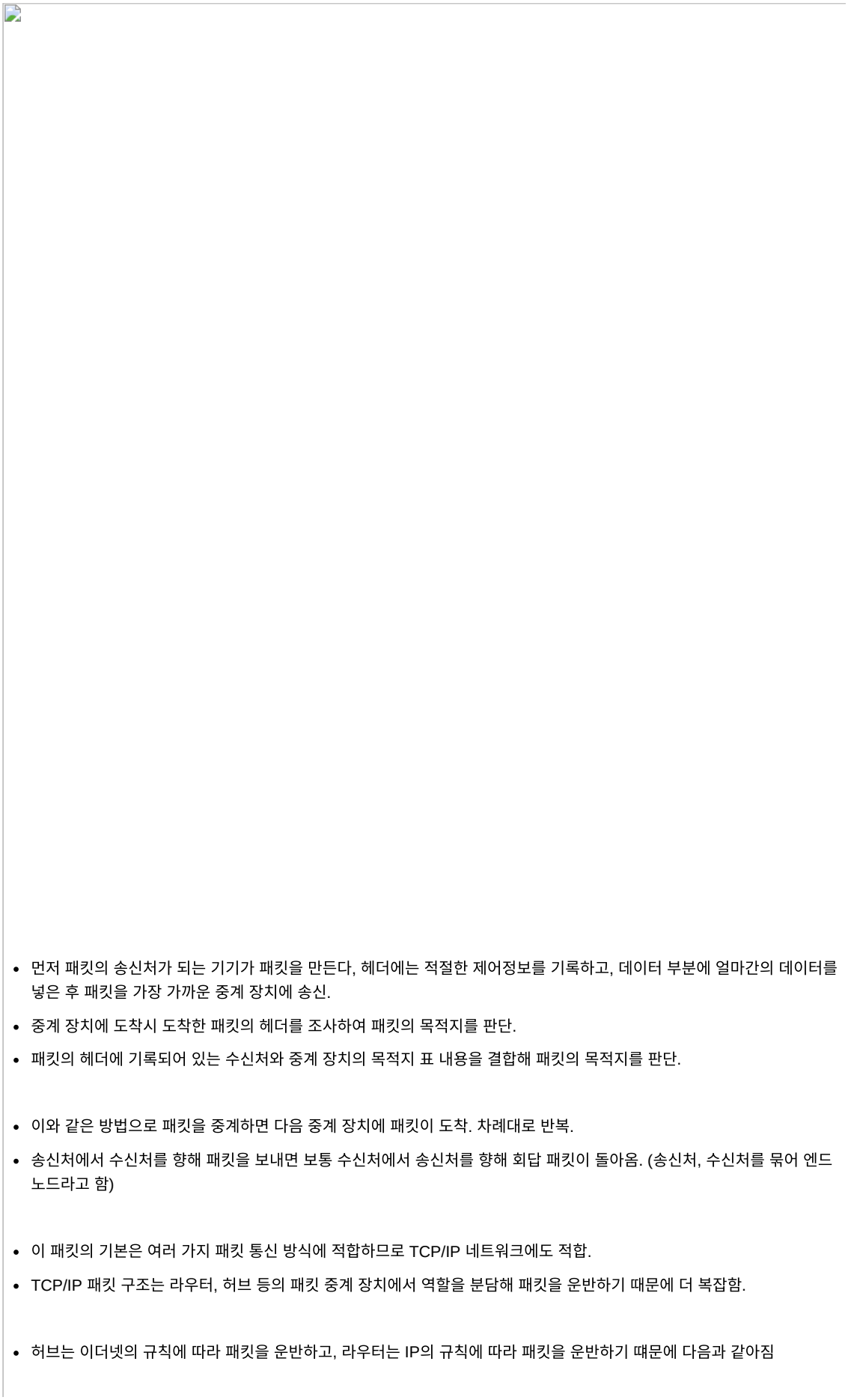
2023-08-27_2 (Chapter 02)

▼ Story.05 IP와 이더넷의 패킷 송,수신 동작

- TCP 프로토콜의 메세지 송,수신 동작의 다음은 실제로 패킷을 송수신 하는 동작.
- 프로토콜 스택과 LAN 어댑터가 연대하여 작은 조각으로 분할한 데이터를 패킷 형태로 묶고 그것을 전기나 빛의 신호로 변환하여 케이블에 송출하는 모습을 설명.

▼ 1) 패킷의 기본

- 의뢰를 받은 IP 담당 부분이 어떻게 패킷을 상대에게 송신하는지 살펴보자.
- 패킷은 헤더, 데이터 두 부분으로 구성.
- 헤더에는 수신처를 나타내는 주소 등의 제어 정보가 들어있음.
- 데이터는 내용물.



- 먼저 패킷의 송신처가 되는 기기가 패킷을 만든다, 헤더에는 적절한 제어정보를 기록하고, 데이터 부분에 얼마간의 데이터를 넣은 후 패킷을 가장 가까운 중계 장치에 송신.
- 중계 장치에 도착시 도착한 패킷의 헤더를 조사하여 패킷의 목적지를 판단.
- 패킷의 헤더에 기록되어 있는 수신처와 중계 장치의 목적지 표 내용을 결합해 패킷의 목적지를 판단.

- 이와 같은 방법으로 패킷을 중계하면 다음 중계 장치에 패킷이 도착. 차례대로 반복.
- 송신처에서 수신처를 향해 패킷을 보내면 보통 수신처에서 송신처를 향해 회답 패킷이 돌아옴. (송신처, 수신처를 묶어 엔드 노드라고 함)

- 이 패킷의 기본은 여러 가지 패킷 통신 방식에 적합하므로 TCP/IP 네트워크에도 적합.
- TCP/IP 패킷 구조는 라우터, 허브 등의 패킷 중계 장치에서 역할을 분담해 패킷을 운반하기 때문에 더 복잡함.

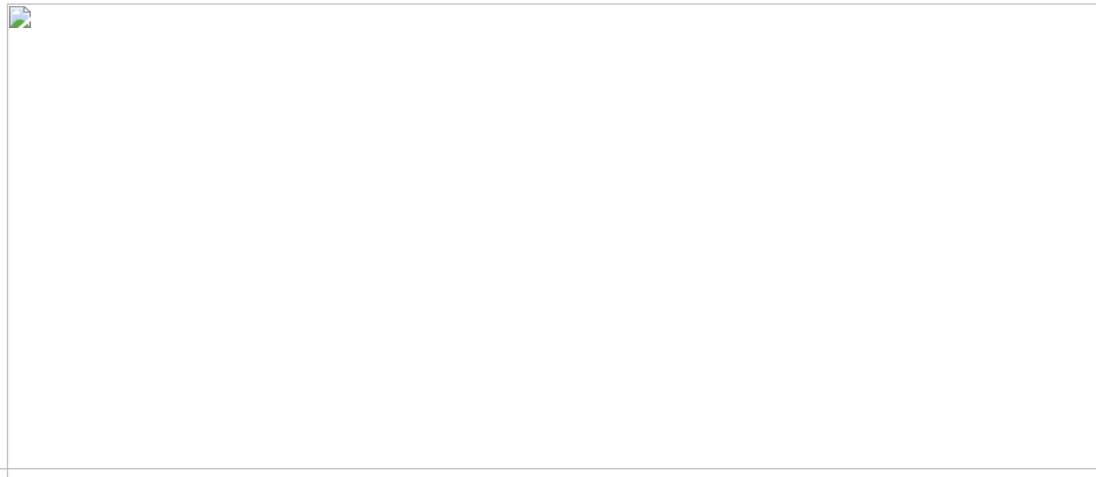
- 허브는 이더넷의 규칙에 따라 패킷을 운반하고, 라우터는 IP의 규칙에 따라 패킷을 운반하기 떄문에 다음과 같아짐

1. 라우터가 목적지를 확인하여 다음 라우터를 나타냄.
2. 허브가 서브넷 안에서 패킷을 운반하여 다음 라우터에 도착.

→

1. IP가 목적지를 확인하여 다음 IP의 중계 장치를 나타냄
2. 서브넷 안에 있는 이더넷이 중계 장치까지 패킷을 운반.

- TCP/IP 패킷에는 MAC 헤더(이더넷용 헤더), IP 헤더(IP용 헤더) 2개 헤더가 붙어있다.
- 송신처에서 패킷의 목적지가 되는 액세스 대상 서버의 IP 주소를 IP 헤더의 수신처에 기록.
- IP는 이 수신처가 어느 방향에 있는지를 조사하고, 그 방향에 있는 다음 라우터를 조사.



- 위 그림의 경우 R1 라우터가 첫 다음 라우터. 거기에 패킷을 보내도록 의뢰.
- 이때 다음 라우터에 할당된 이더넷 주소(MAC 주소)를 조사하고, 그것을 MAC 헤더에 기록.

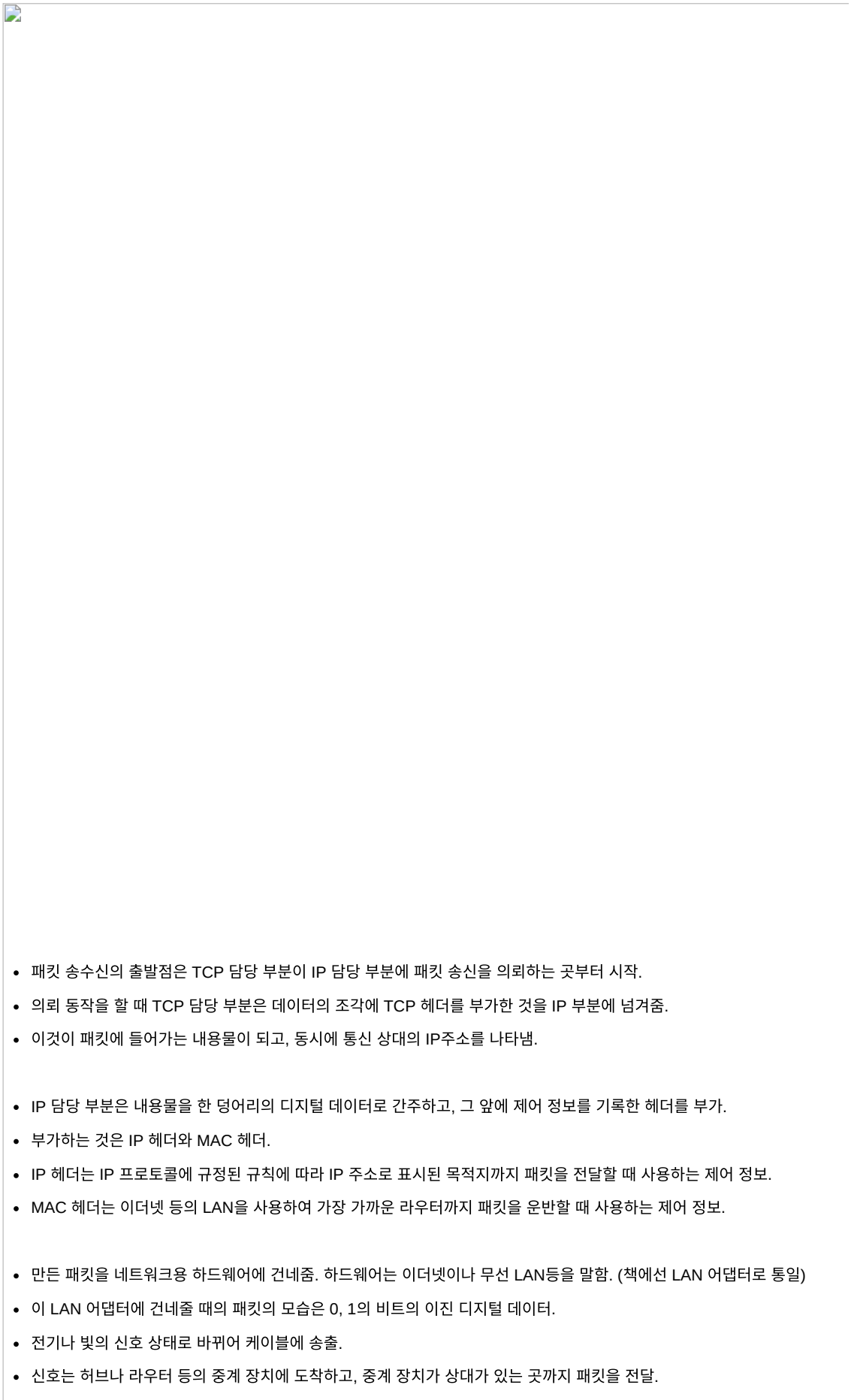
- 이렇게 패킷을 송신하면 이더넷의 원리에 따라 움직이는 허브에 도착.
- 허브에는 패킷의 목적지를 판단하기 위한 표(이더넷용 표)와 같은 것이 있어 이더넷의 헤더의 수신처 정보와 표를 결합해 패킷의 목적지를 판단하여 중계.
- 허브가 복수이면 허브를 순차적으로 경유하여 패킷이 진행됨.

- 그러면 패킷은 다음 라우터에 도착.
- 라우터에는 IP용 표가 있으므로 이것과 IP 헤더의 수신처를 결합해 다음에 어느 라우터에 패킷을 중계하면 좋을지 결정됨.
- 그리고 다음 라우터에 패킷을 건네주기 위해 라우터의 MAC 주소를 조사하고, 이것을 MAC 헤더에 기록. 다음 라우터에 송신.
- 허브가 있다면 허브를 경유해 다음 라우터가 되는 R2에 패킷이 도착.
- 이것을 반복하면 패킷은 목적지에 도착해 수신시 전달 동작 완료.

- 조금 복잡하지만 역할을 분담하는 이유가 있다.
- 이더넷 부분은 다른 것으로 대체할 수 있다.
  (무선 LAN, ADSL, FTTH 등 IP의 의뢰를 받아 패킷을 운반할 수 있는 것이면 무엇이든..)
- 이더넷과 같은 거대한 네트워크를 구축하려면 유연성이 필요한데, 이것이 역할을 분담하는 이유.

▼ 2) 패킷 송,수신 동작의 개요

- 프로토콜 스택의 IP 담당 부분의 패킷 송신 동작에 대해 알아보자.
- IP 담당 부분은 패킷을 상대에게 송출만 하기 만 함.
- 그 뒤에 상대가 있는 곳까지 패킷을 운반하는 것은 허브나 라우터 같은 네트워크 기기의 역할.



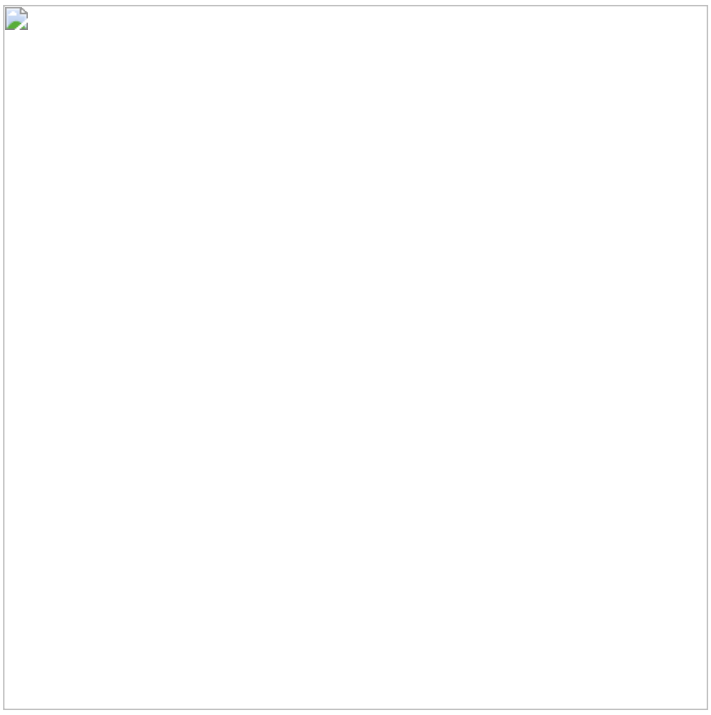
- 패킷 송수신의 출발점은 TCP 담당 부분이 IP 담당 부분에 패킷 송신을 의뢰하는 곳부터 시작.
- 의뢰 동작을 할 때 TCP 담당 부분은 데이터의 조각에 TCP 헤더를 부가한 것을 IP 부분에 넘겨줌.
- 이것이 패킷에 들어가는 내용물이 되고, 동시에 통신 상대의 IP주소를 나타냄.

- IP 담당 부분은 내용물을 한 덩어리의 디지털 데이터로 간주하고, 그 앞에 제어 정보를 기록한 헤더를 부가.
- 부가하는 것은 IP 헤더와 MAC 헤더.
- IP 헤더는 IP 프로토콜에 규정된 규칙에 따라 IP 주소로 표시된 목적지까지 패킷을 전달할 때 사용하는 제어 정보.
- MAC 헤더는 이더넷 등의 LAN을 사용하여 가장 가까운 라우터까지 패킷을 운반할 때 사용하는 제어 정보.

- 만든 패킷을 네트워크용 하드웨어에 건네줌. 하드웨어는 이더넷이나 무선 LAN등을 말함. (책에선 LAN 어댑터로 통일)
- 이 LAN 어댑터에 건네줄 때의 패킷의 모습은 0, 1의 비트의 이진 디지털 데이터.
- 전기나 빛의 신호 상태로 바뀌어 케이블에 송출.
- 신호는 허브나 라우터 등의 중계 장치에 도착하고, 중계 장치가 상대가 있는 곳까지 패킷을 전달.

- TCP 담당 부분의 데이터 송수신 동작에는 몇 개의 단계가 있다.
- 각 단계에서 다양한 역할을 가진 패킷이 등장.
- IP 패킷 송수신 동작은 패킷의 역할에 관계없이 모두 같다.
- IP 담당은 TCP 헤더와 데이터 조각을 한덩어리의 바이너리 데이터로 간주하여 내용을 보지않고 송수신 동작을 실행.
- TCP 동작 단계도 신경쓰지 않음.

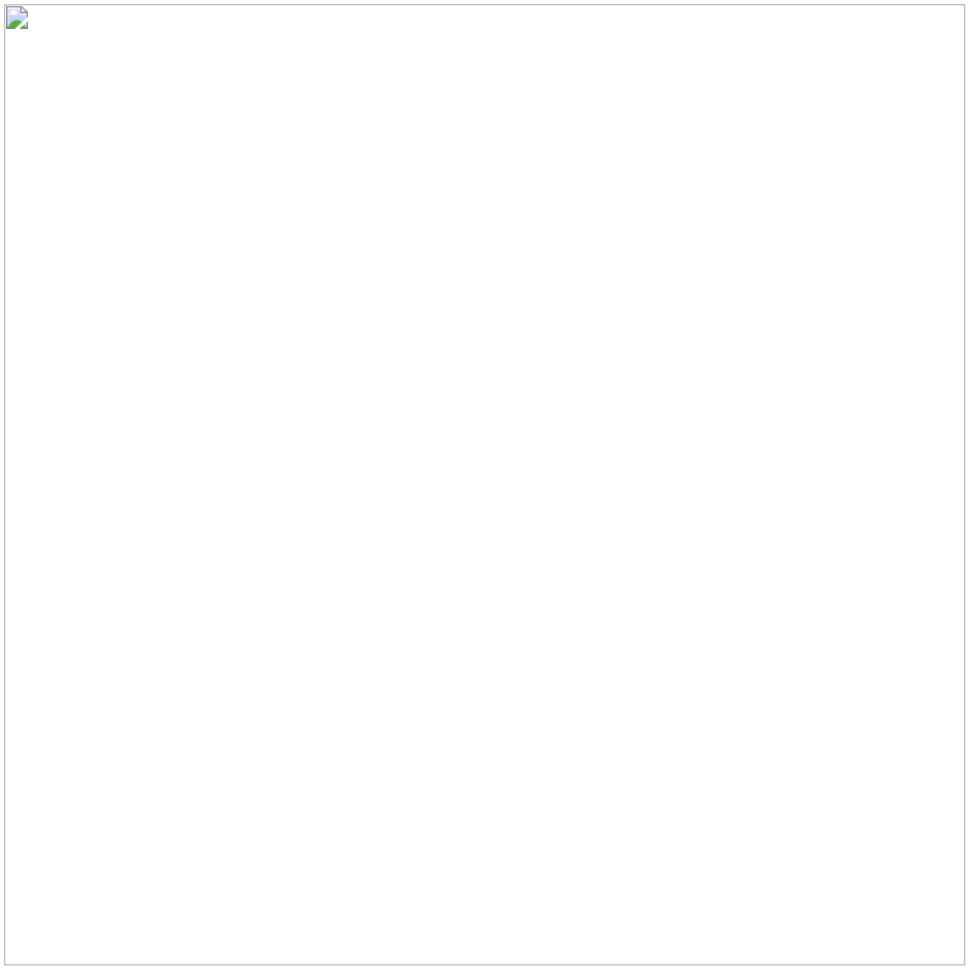
▼ 3) 수신처 IP 주소를 기록한 IP 헤더를 만든다.



- IP 헤더표
- 가장 중요한 것은 수신처 IP 주소.
  (여기에 TCP 담당 부분에서 통지된 통신 상대의 IP주소를 설정)
- IP는 스스로 수신처를 판단하지 않고 애플리케이션이 지정한 상대에게 패킷을 송신할 뿐 애플리케이션이 IP주소를 잘못 지정해도 그 IP주소를 그대로 IP 헤더에 설정.

- 송신처 IP주소도 설정.
- 이 컴퓨터에 할당된 IP주소를 설정한다고 생각하면 됨.
  (보통 PC, LAN 어댑터가 하나인 경우).
- but IP 주소는 사실 컴퓨터가 아니라 LAN 어댑터에 할당되므로, 여러 개의 LAN 어댑터가 장착되있으면 각 LAN 어댑터에 서로 다른 IP주소가 할당되 일반적이진 않음.

- 서버 기계 등에서 복수의 LAN 어댑터를 장착할 수 있는데, 이 경우에는 한 대의 컴퓨터에 할당된 IP 주소가 여러 개가 되므로 어느 IP주소를 설정해야할지 판단해야 한다.
- 패킷을 건네주는 상대의 라우터를 결정하는 것과도 같다.
- 패킷을 건네줄 상대를 판단하는 방법은 라우터가 IP용 표를 사용해 다음 라우터를 결정하는 동작과 같다.



- IP용 표를 경로표라고 부르는데, 사용법은 3장에서 설명.
- 경로표는 route print 명령으로 내용 표시 가능.
- 소켓에 기록되어 있는 수신처 IP 주소를 경로표의 왼쪽 끝에 있는 Network Destination 항목과 비교해 어느 행에 해당하는지 찾아냄.
- ex)
  - TCP 담당 부분에서 통지해온 수신처 IP주소가 192.168.1.21이면 192.168.1 이라는 부분이 일치하기 때문에 6행에 해당.
  - 수신처 IP 주소가 10.10.1.166 이라면 10.10.1 이라는 부분이 일치하므로 세 번째 행에 해당.
- 이렇게 해당하는 행을 찾아낸 다음 오른쪽부터 두 번째와 세 번째 항목을 조사.
- 오른쪽에서 두 번째에 있는 Interface 항목은 LAN 어댑터 등의 네트워크용 인터페이스를 나타내고, 인터페이스에서 패킷을 송신하면 상대에 패킷을 전해줄 수 있다는 의미.
- Gateway 항목은 다음 라우터의 IP주소를 기록하게 되있어서 IP 주소를 가진 라우터에 패킷을 건네주면 라우터가 목적지에 패킷을 중계해 준다는 것을 나타냄.

- 경로표의 맨 위에는 목적지와 넷마스크가 0.0.0.0으로 등록되 있다.
- 소위 기본 게이트위이를 나타내며 다른 곳에 일치하는 곳이 없으면 이 행이 해당하는 것으로 간주.
- 이렇게 해서 어느 LAN 어댑터에서 패킷을 송신해야 하는지 알고 나서 LAN 어댑터에 할당되어 있는 IP 주소를 IP 헤더의 송신처 IP 주소로 설정.

- 프로토콜 번호라는 필드에도 값을 설정.
- 패킷에 들어간 내용물이 어디에서 의뢰받은 것인지를 나타내는 값을 설정.

- TCP에서 의뢰받은 내용물이라면 06 (16진수), UDP에서 의뢰받은 것이면 17(16진수).
- 값은 규칙에 결정되어 있다.
- 지금은 브라우저의 HTTP 리퀘스트 메세지를 TCP에서 운반하기로 했으므로 여기에는 TCp를 나타내는 06 이라는 값을 설정해야 함.

▼ 4) 이더넷용 MAC 헤더를 만든다.

- IP 헤더를 만들었으면 MAC 헤더를 붙인다.
- 이더넷에는 TCP/IP 개념이 통용되지 않음.
- 이더넷의 수신처 판단 구조로 사용하는 것이 MAC 헤더.

- MAC 헤더의 맨 앞에 있는 수신처 MAC 주소와 그 다음의 송신처 MAC 주소는 각각 패킷을 전달하는 상대와 패킷을 송신한 송신처의 MAC 주소를 나타냄.
- IP 헤더에 있는 수신처 IP 주소 및 송신처 IP 주소와 같은 역할.

- IP 주소는 32비트 이지만 MAC 주소는 48비트.
- IP 주소는 그룹화 개념이지만, MAC주소는 48비트 한 개의 값으로 생성.

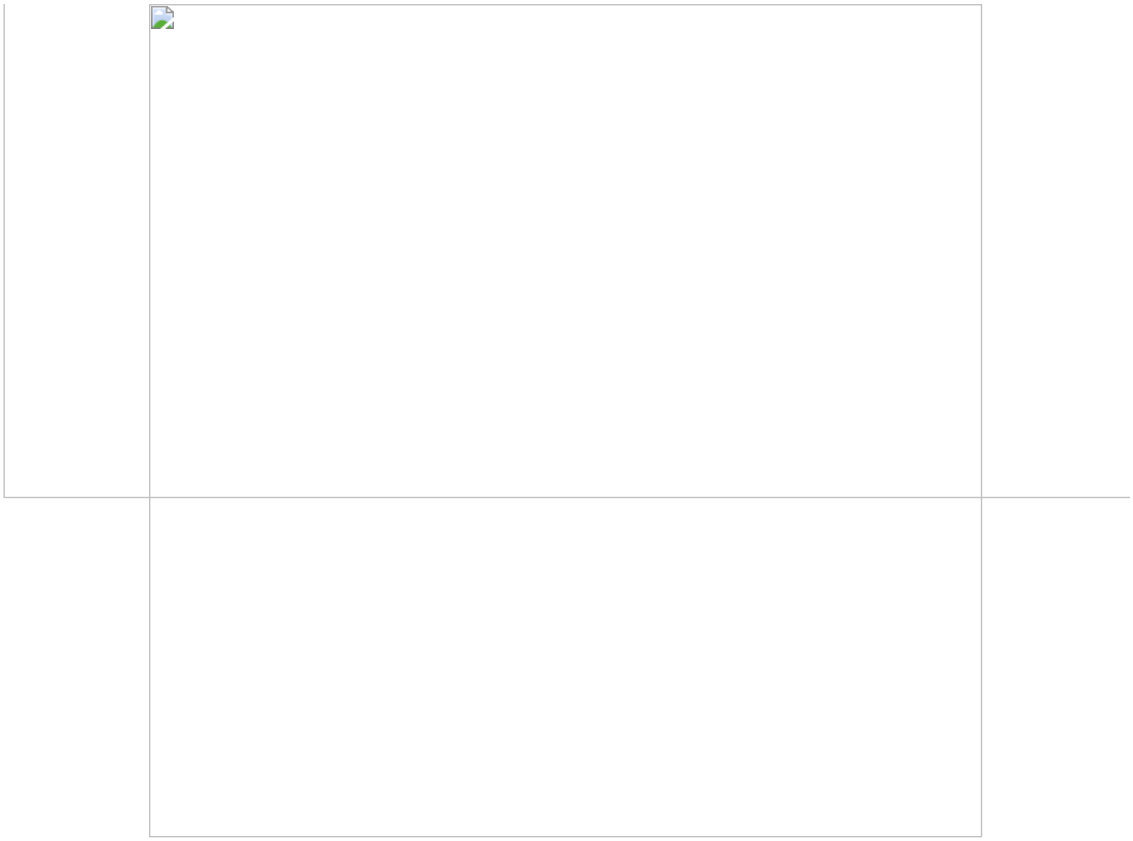
- 3개의 이더 타입(EtherType) 항목은 IP 헤더의 프로토콜과 비슷.
- IP의 경우 IP 헤더 뒤에 이어지는 거이 패킷의 내용물. 의뢰 포맷 주체는 프로토콜 번호.
- 이더넷의 경우 이더 타입까지가 MAC 헤더, 그 뒤에 이어지는 것이 패킷의 내용물.
- 내용물의 종류를 이더 타입으로 나타낸다.
- 이더넷의 내용물은 IP, ARP라는 프로토콜의 소켓.

- MAC 헤더를 만들 때는 세가지 항목에 값을 설정하기만 함.
- 이더 타입 필드는IP 프로토콜을 나타냄 (0800 이라는 값 설정) 16진수.
- 다음은 송신처 MAC 주소, 여기에 자체의 LAN 어댑터의 MAC 주소를 설정한다.
- MAC 주소는 LAN 어댑터를 제조할 때 그 안에 있는 ROM에 기록되므로 여기에 기록되어 있는 값을 읽어와서 MAC 헤더로 설정.
- 여러 개의 LAN 어댑터가 장착되어 있는 경우 송신처 IP 주소를 설정할때 어느 LAN 어댑터에서 송신할지를 판단하고 나서 LAN 어댑터에 할당된 MAC 주소를 설정.

- 수신처 MAC 주소는 다소 복잡.
- 여기에 패킷을 건네주는 상대의 MAC 주소를 설정하여 이더넷에 의뢰한 후 상대에게 패킷이 전달되므로 상대의 MAC 주소를 기록해야 함.
- 첫 시점에서는 누구에게 패킷을 건네주어야 할지 모르기 떄문에, 우선 패킷을 줄 상대가 누구인지 조사하는데, 이것은 경로표에 기록되어 있다.
- 경로표에서 일치하는 행의 GateWay항목에 기록되어 있는 IP주소의 기기가 패킷을 건네줄 상대가 된다.

- 패킷을 건네줄 상대를 알았으면 상대의 MAC 주소를 수신처 MAC 주소의 필드에 설정하면 되지만, 상대의 MAC 주소는 지금까지 어디에서도 없기 때문에 IP 주소에서 MAC 주소를 조사흐는 동작을 실행함.

▼ 5) ARP로 수신처 라우터의 MAC 주소를 조사한다.



- ARP 개념은 이더넷에 브로드캐스트로 MAC주소를 조회.
- 상대가 자신과 같ㅌ은 네트워크에 존재하면 이것으로 MAC 주소를 알 수 있다.
- 그러면 MAC 주소를 MAC 헤더에 설정하여 MAC 헤더를 만든다.

- 패킷을 보낼 때마다 이 동작을 하면 ARP의 패킷이 불어나기 때문에 한 번 조사한 결과는 ARP 캐시 메모리 영역에 보존하여 다시 이용.
- 패킷을 송신할 때 우선 ARP 캐시를 조사 거기에 상대 MAC 주소가 저장되 있으면 ARP를 조회하지 않고 없다면 ARP를 조회.

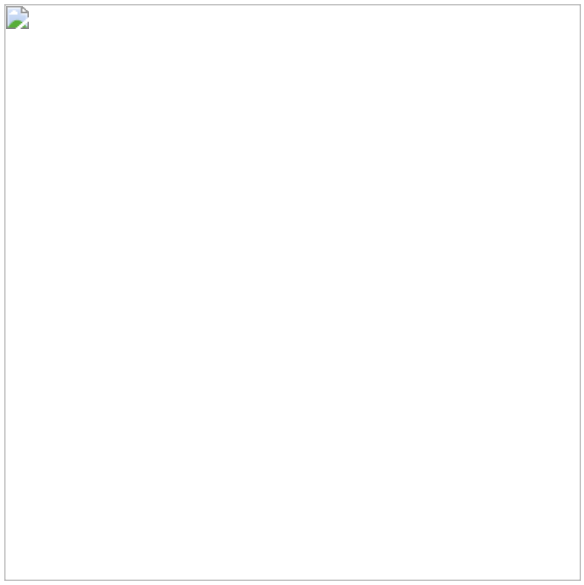
- ARP 캐시를 사용해 패킷을 줄일 수 있지만, 캐시에 저장된 MAC 주소를 언제까지나 계속 사용하면 현실의 내용과 캐시의 내용이 일치하지 않을 수 있어 문제가 될 수 있다.
- 이것을 막기 위해 ARP 캐시에 저장된 값은 시간이 되면 삭제하게 되어 있다 (보통 몇 분)
- 캐시 설정후 IP 주소를 다시 설정하면 통신 오류 발생할 수 있다.

- MAC 헤더를 IP 헤더의 앞에 붙이면 패킷 완성. 패킷 만들기 까지가 IP 담당 부분의 역할.
- MAC헤더는 IP 범위는 아니지만 IP 에서 담당한쪽이 더 좋은 방법이라 포함된다.
- LAN 어댑터는 완성된 패킷만 송신하면 됨.

- IP 이외의 패킷인 경우도 마찬가지로 LAN 어댑터에 건네주기 전에 패킷을 완성하면 LAN 어댑터는 IP와 같이 완성된 패킷만 송신하면 된다.
- IP 이외의 특수 패킷도 한개의 LAN 어댑터로 대응 가능.

▼ 6) 이더넷의 기본

- 이더넷은 다수의 컴퓨터가 여러 상대와 자유럽게 적은 비용으로 통신하기 위해 고안된 통신 기술.
- 네트워크의 실체는 케이블만 있다.
- 트랜시버(transceiver)라는 작은 기기도 있지만, 이것을 연결한 케이블 사이에 신호를 흘리는 역할만 함.
- 컴퓨터 한 대가 신호를 보내면 전원에게 신호가 도착한다. 이런 동작만으로는 도착한 신호가 누구에게 갈 지 판단할 수 없으므로 신호의 맨 앞부분에 누구에게 갈 것인지를 나타내는 정보를 쓴다.
- 목적지를 알기 때문에 패킷을 수신하지 않는 다른 기기는 패킷을 폐기.



- 이 동작을 제어하기 위해 MAC 헤더를 사용.
- 수신처 MAC 주소에 따라 패킷의 송신자, 수신자를 알며, 이더 타입에 의해 내용물이 무엇인지 아는 것.

- 더 발전된 모습으로는 트렁크 케이블이 리피터 허브로 바뀌고, 트랜시버 케이블이 트위스트 페어 케이블로 바뀌었다고 생각하면 된다.

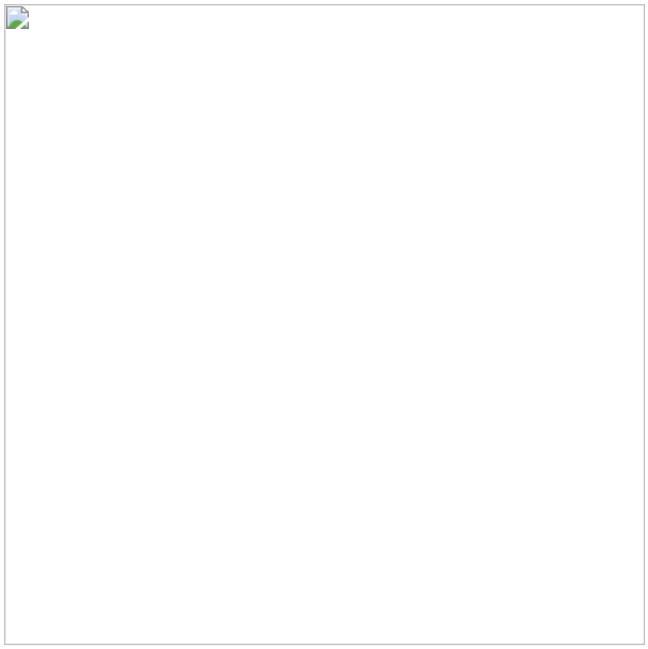
- 그 후 스위칭 허브를 사용한 형태가 보급되었는데, 이게 현재의 이더넷 형태.
- 비슷해 보이지만, 전원에게 신호가 전달된다는 성질이 변했으며, 수신처 MAC 주소로 나타내는 원하는 기기가 존재하는 부분에만 신호가 흐르고, 다른 곳에는 신호가 흐르지 않게 된 것.

- 이더넷에 접속된 기기는 이더넷 하나의 사양에 기초하여 동작하기 때문에 클라이언트 PC뿐 아니라 서버와 라우터를 포함한 모든 기기에 공통저긍로 적용됨.
- 이더넷도 IP와 마찬가지로 패킷의 내용물을 보지 않으므로 송수신 동작은 TCP 동작 단계에 상관 없이 모든 것에 공통.

▼ 7) IP 패킷을 전기나 빛의 신호로 변환하여 송신한다.

- IP 패킷은 메모리에 기억된 디지털 데이터이므로 그대로 상대에게 보낼 수 없다.
- 디지털 데이터를 전기나 빛의 신호로 변환하여 네트워크의 케이블에 송출하는데, 이것이 송수신의 본질.

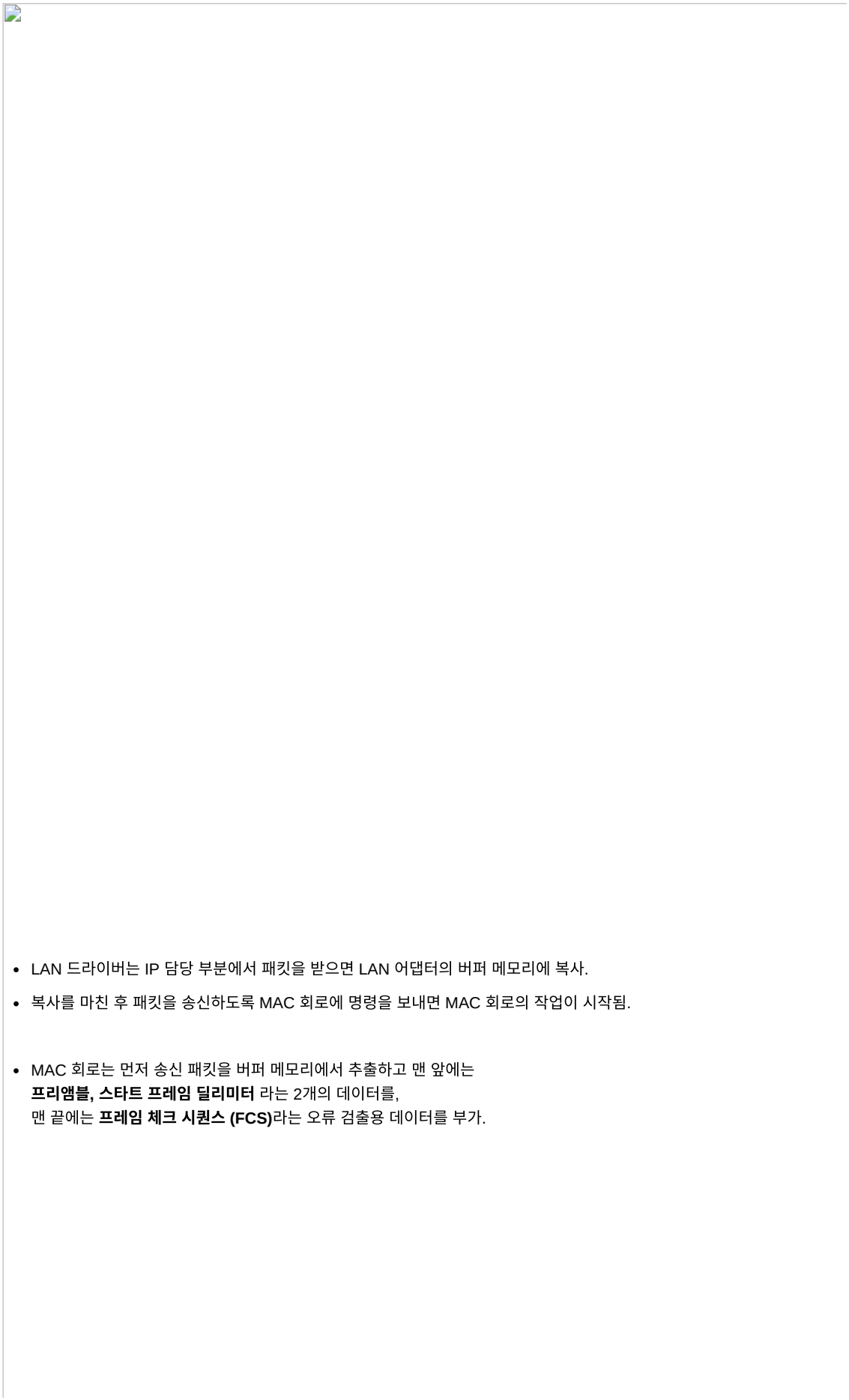
- 이 동작을 실행하는 것이 LAN 어댑터인데, LAN 어댑터는 단독으로는 동작하지 않는다.
- LAN 어댑터를 제어하려면 LAN 드라이버 소프트웨어가 필요.
- 이것은 LAN 어댑터에 한정되지 않고, 키보드, 마우스.. 등 모든 하드웨어에 공통이다.



- LAN 어댑터의 개념도.
- LAN 어댑터는 전원을 공급하면 즉시 사용하는 것이 아니라 초기화 작업이 필요함.
- 전원을 공급하여 OS를 시동할 때 LAN 드라이버가 하드웨어의 초기화 작업을 수행해야 사용 가능한 상태가 된다.
- 초기화에선 하드웨어 이상 검사, 초기 설정등과 이더넷 특유의 MAC 주소 설정도 함.

- LAN 어댑터의 ROM에는 전 세계에서 중복되지 않도록 일원화한 MAC 주소를 제조할 때 기록한다.
- 이것을 읽어와서 MAC 회로에 설정.
- 명령이나 설정 파일에 MAC 주소를 받아 설정하는 특수한 사용법도 있다.
  - 이때 ROM에 기록된 것 무시.
- 결국 ROM의 MAC 주소가 아니라 초기화 작업시 설정된 MAC 주소가 유효하게 되는 것.
- OS를 기동할 때 초기화를 끝낸 후 IP에서 의뢰하기를 기다린다.

▼ 8) 패킷에 3개의 제어용 데이터를 추가한다.



- LAN 드라이버는 IP 담당 부분에서 패킷을 받으면 LAN 어댑터의 버퍼 메모리에 복사.
- 복사를 마친 후 패킷을 송신하도록 MAC 회로에 명령을 보내면 MAC 회로의 작업이 시작됨.

- MAC 회로는 먼저 송신 패킷을 버퍼 메모리에서 추출하고 맨 앞에는
  프리앰블, 스타트 프레임 딜리미터 라는 2개의 데이터를,
  맨 끝에는 **프레임 체크 시퀀스 (FCS)**라는 오류 검출용 데이터를 부가.

- 프리앰블은 송신하는 패킷을 읽을 때의 타이밍을 잡기 위한 것, ‘10101010….’과 같이 1, 0이 번갓아 나타나는 비트열이 56 비트 이어진 것.
- 이 1010이라는 비트 패턴을 신호로 바꾸면 파형이 일정한 모습이 된다.
- 수신측은 신호를 수신할 때 이 파형에서 타이밍을 판단.

- 디지털 데이터를 전기 신호로 나타낼 때는 0,1 비트 값을 전압이나 전류의 값에 대응.
- 신호에서 데이터를 읽을 때의 동작은 이 대응을 반대로 실행하면 된다.
- 실제 신호에는 각 비트의 구분을 나타내는 보조선이 있으므로 각 비트의 구분선이 어디까지 인지 판단하면서 전압이나 전류의 값을 읽어야 한다.
- 1, 0의 값이 연속되면 신호의 변화가 없어져서 비트 구분을 판단할 수 없는 문제가 생긴다.
- 이 문제를 해결하기 위해 데이터를 나타내는 신호와는 별도로 비트 구분을 나타내는 클록이라는 신호를 보내는 방법.
- 클록 신호가 아래에서 위로 변할때 데이터 신호의 전압이나 전류의 값을 읽고 0, 1로 대응시키면 된다.
- 거리가 멀어져 케이블이 길어지면 신호선의 길이가 달라져서 데이터 신호와 클록 신호가 전달되는 시간에 차이가 생기기 때문에 클록이 틀어져 버리는 문제도 있음.

- 데이터 신호와 클록 신호를 합성하여 한 개의 신호로 만들면 해결된다.
- 클록 신호는 일정 주기로 결정된 모습으로 변화하는 신호.

- 따라서 변화의 타이밍까지 알고 있으면 수신한 신호에서 클록 신호를 추출하고 클록 신호를 추출하면, 수신 신호와 클록 신호의 두 신호에서 원래 데이터 신호를 추출할 수 있다.

- 클록 신호로 타이밍을 잡으면 데이터 신호에서 비트 값을 읽을 수 있다.
- 이렇게 전압, 전류의 값에서 0, 1의 비트 값으로 되돌릴 수 있다.

- 클록 신호의 타이밍을 판단하는 것이 중요.
- 10메가비트/초, 100메가비트/초 라는 식으로 클록이 변화하는 주기는 결정되어 있으므로 잠시 신호의 변화를 볼 수 있으면 타이밍을 파악할 수 있다.
- 갑자기 패킷의 신호를 흘리는 것이 아니라 클록 신호의 타이밍을 잡기 위한 특별한 신호를 패킷 앞에 부가하면 된느데, 이것이 프리앰블의 역할.

- 이더넷에는 속도나 케이블의 종류에 따라 파생 방식이 있으며, 이 방식에 따라 신호의 모습이 달라지므로 단순히 0,1을 전압이나 전류로 나타내는 것으로 한정되지는 않는다.
- 101010.. 이라는 디지털 값을 전기 신호로 바꾼 프리앰블의 파형이 반드시 위 그림처럼 되는게 아니라 방식에 따라 달라짐.
- but 프리앰블의 역할과 기본적인 개념은 달라지지 않음.

- 스타트 프레임 딜리미터도 있지만 마지막 비트 패턴이 조금 다르다.
- 수신측은 이것을 표시하여 신호에서 데이터를 추출하기 시작함.
- 스타트 프레임 딜리미터가 패킷의 시작을 나타내는 표시가 됨.

- 끝에 부가하는 FCS는 패킷을 운반하는 도중에 잡음 등의 영향으로 파형이 흐트러져 데이터가 변한 경우 이것을 검출하기 위해 사용.
- 32비트의 비트열이며, 패킷의 맨 앞부분에서 맨 끝까지의 내용을 어떤 계산식에 기초하여 계산한 것.
- 구체적인 계산식은 생략하지만 CRC(Cyclic Redundancy Check)라는 디스크 장치 등에 사용하는 오류 검사 코드와 같은 종류이므로 계산의 바탕이 된 데이터 값이 1비트라도 변화하면 계산한 결과도 달라진 값을 취하도록 고안.
- 패킷을 운반하는 도중 잡음 등의 영향으로 내용의 데이터가 변하면 수신측에서 계산한 FCS가 송신할 때 계산한 것과 다른 값이 됨.
- 이런 불일치를 통해 데이터가 변화한 시실을 검출.

▼ 9) 허브를 향해 패킷을 송신한다.

- 프리앰블, 스타트 프레임 딜리미터, FCS 세가지를 부여하면 케이블에 송출하는 패킷이 완성.
- 신호 송출은 리피터 허브를 사용했을 때 반이중 모드, 스위칭 허브를 사용한 전이중 모드 2가지가 있다. (여기선 반이중 모드 설명)

- 반이중 모드 동작은 신호의 충돌을 피하기 위해 동작
- 먼저 케이블에 다른 기기가 송신한 신호가 흐르고 있는지 조사, 흐르고 있으면 끝날 때까지 기다림.
- 신호가 흐르고 있을 때 송신 동작을 시작하면 신호가 충돌하기 때문.
- 신호가 정지했거나 흐르지 않는다면 송신 동작을 시작.
- 송신 동작은 MAC 회로가 프리앰블의 맨 앞부터 1비트씩 차례로 디지털 데이터를 전기 신호로 변환하고, 이것을 PHY, MAU라는 송수신 신호 부분에 보냄.
- 디지털 데이터를 신호로 변환하는 속도가 전송 속도.
- 1초 동안 10메가 비트 분량의 디지털 데이터를 신호로 변환하면 10메가비트/초 라는 전송률

- PHY(MAU) 회로는 이 신호를 케이블에 송출하는 형식으로 변환하여 송신.
- 이더넷은 케이블의 종류나 전송 속도에 따라 몇 가지 신호 형식이 규정되어 있지만, MAC회로는 형식 차이를 신경쓰지 않고 어느 형식으로도 변환할 수 있는 공통 형식의 신호를 PHY(MAU) 회로에 보낸다.
- PHY(MAU) 회로는 MAC 회로가 송신한 신호의 형식을 변환하기 위한 변환회로라고 생각하면 됨.

- 이더넷에는 다수의 파생 형식이 있으며 케이블에 송출하는 신호의 형식에는 많은 변형이 있다.
- LAN 어댑터의 MAC 회로가 공통 형식의 신호를 만들고 PHY(MAU) 회로가 케이블에 송출하는 형식으로 변환하여 케이블에 송신.

- PHY(MAU) 회로가 MAC 회로에서 받은 신호를 케이블에 송신할 때 단지 송신 동작만 실행하는 것이 아니라 수신 신호선에서 신호가 들어오는지 감시.
- 송신 개시 전에 신호가 흐르지 않는 것을 확인한 후 송신 동작에 들어가 송신을 시작했을 때는 수신 신호선에 신호가 흐르지 않을 것.
- 이더넷이라는 통신 방식은 송신한 신호가 상대에게 완전하게 도착했는지 확인하지 않음.
- 이더넷은 사양에서 기기와 기기 사이를 연결하는 케이블 길이를 100m 이내여서 오류가 좀처럼 발생하지 않는다.
- 오류가 발생해도 프로토콜 스택위 TCP가 검출하므로 신호를 송신할 때 오류를 확인할 필요가 없다.

- 신호를 송신하고 있는 사이에 수신 신호가 흘러오지 않으면 되는데, 흘러들어 올수도 있다. (드물지만 동시에 복수의의 기기가 송신 동작에 들어갈 가능성이 있기 때문)
- 만약 동시에 송신 동작에 들어간 기기가 있으면 기기가 보낸 신호가 수신 신호선으로 흘러온다.
- 리피터 허브를 사용한 반이중 모드의 경우 서로의 신호가 뒤섞여서 분간할 수 없는 상태가 되는데, 이것이 바로 충돌이라는 현상.
- 이렇게 되면 이상 송신을 계속해도 의미가 없으므로 송신 동작을 중지.
- 충돌이 일어난 사실을 다른 기기에 알리기 위해 재밍 신호라는 특수한 신호를 잠시 흘리고 송신 동작을 멈추고 잠시 기다렸다 다시 송신 동작을 시도.
- 이때 충돌을 일으킨 기기의 대기 시간이 동일하다면 다시 충돌이 일어나므로 대기 시간이 중복되지 않도록 고안됨.
  (MAC 주소를 바탕으로 난수를 생성하고 여기에서부터 대기 시간을 계산)
- 이더넷이 혼잡해지면 충돌의 가능성이 높아지므로 다시 보낼 때 다른 기기와 송신 동작이 겹쳐서 다시 충돌할 수 있다.
- 이러면 대기 시간을 2배로 늘려서 다시 보냄.
- 열 번째 까지 다시 보냈는데 해결되지 않는 다면 오류로 판단한다.

▼ 10) 돌아온 패킷을 받는다.

- LAN 어댑터에서 패킷을 전기 신호로 변환하여 송출하는 동작은 이것으로 끝
- 리피터 허브를 이용한 반이중 동작의 이더넷에서는 1대가 송신한 신호가 리피터 허브에 접속된 케이블 전부에 흘러감.
- 자신뿐 아니라 누군가 신호를 보내면 그것이 전부 수신 신호선에서 흘러온다.
- 그러므로 수신 동작은 이러한 신호를 전부 받아들이는 것부터 시작.

- 신호의 맨 앞에는 프리앰블이 있으므로 파형에서 타이밍을 계산하여 스타트 프레임 딜리미터가 나오면, 다음 비트부터 디지털 데이터로 변환하여 동작을 개시.
- 송신할 떄와 반대로 PHY(MAU) 회로에서 MAC 회로쪽으로 진행.
- PHY(MAU) 회로에서 신호를 공통 형식으로 변환하여 MAC 회로에 보내고, MAC 회로에서 신호를 앞부터 차례대로 디지털 데이터로 변환하여 버퍼 메모리 저장.
- 신호의 마지막에 FCS를 검사.
  (패킷의 앞부터 계산을 적용해 FCS 값을 계산하고 마지막 FCS값과 비교 정상히면 둘이 일치, 잡음 등의 영향으로 파형이 흐트러지면 오류 패킷으로 간주하여 폐기)

- FCS에 문제가 없으면 MAC 헤더의 수신처 MAC 주소를 조사하여 LAN 어댑터를 초기화할 떄 설정한 자체의 MAC 주소와 비교한 후 자신에게 오는 것인지 판단.
- 다른 곳에 갈 패킷은 수신할 필요가 없으므로 폐기.
- 수신처 MAC 주소가 자신에게 오는 것인 경우에만 패킷을 버퍼 메모리에 저장.
- MAC 회로가 할 일이 끝나면 패킷을 수신한 사실을 컴퓨터 본체에 통지.
- 이 통지는 인터럽트 구조를 사용
- LAN 어댑터가 패킷 송수신 동작을 실행하고 있는 사이에 컴퓨터 본체는 LAN 어댑터의 움직임을 감시하는 것이 아니라 다른 작업을 실행하고 있다.
- 그러므로 LAN 어댑터측에서 알려주지 않으면 컴퓨터 본체는 패킷의 도착을 알아차리지 못한다.
- LAN 드라이버도 컴퓨터 본체측에서 움직이는 프로그램이라 패킷의 도착을 알아차리지 못함.
- 이런 상태일 때 컴퓨터 본체가 실행하고 있는 작업에 끼어들어 LAN 어댑터쪽에 주의시키는 것이 인터럽트.

- LAN 어댑터가 확장 버스 슬롯 부분에 있는 인터럽트용 신호선에 신호를 보냄.
- 이 신호선은 컴퓨터 본체측의 인터럽트 컨트롤러를 통해 CPU에 연결되어 있으며, 신호가 흘러오면 CPU는 실행하고 있던 작업을 일시적으로 보류하고 OS 내부의 인터럽트 처리용 프로그램쪽으로 전환.

- LAN 드라이버가 호출되어 LAN 어댑터를 제어하면서 송수신 동작을 실행.

- 인터렙트에는 번호가 할당되어 있어서 LAN 어댑터를 설치할 떄 번호를 하드웨어로 설정.
- 인터럽트 처리용 프로그램쪽은 하드웨어의 인터럽트 번호에 대응하도록 드라이버 소프트웨어를 등록하게 되어 있다.
- LAN 어댑터에 11번이라는 인터럽트 번호를 설정하면 11번에 대응하여 LAN 드라이버가 호출되도록 인터럽트 처리용 프로그램에 등록하여 LAN 어댑터가 인터럽트를 걸면 LAN 드라이버가 호출됨.
- 현재는 PnP 사양에 따라 번호를 자동으로 설정하므로 인터럽트 번호를 걱정할 필요가 없다.
- 수동으로 했을때는 문제가 발생.

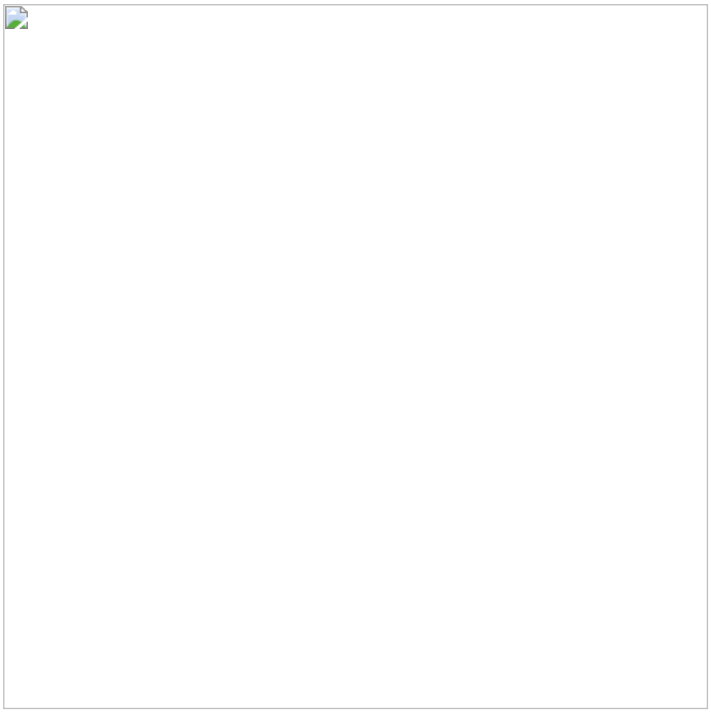
- 인터럽트에 의해 LAN 드라이버가 동작하고 LAN 어댑터의 버퍼 메모리에서 수신한 패킷을 추출하면, LAN 드라이버는 MAC 헤더의 타입 필드의 값으로부터 프로토콜을 판별.
- TCP/IP 이외의 프로토콜을 사용하는 예가 적지만 프로토콜은 TCP/IP 이외에도 NetWare에 사용하던 IPX/SPX, AppleTalk 등이 있는데, 이런 프로토콜을 타입 필드의 값으로 판별.
- 타입의 값이 0800(16진수)면 IP 프로토콜 데이터 이므로 TCP/IP의 프로토콜 스택에 건네주고 809B이면 AppleTalk이므로 AppleTalk의 프로토콜 스택에 건네준다.

- 웹 서버에 패킷을 보낸 뒤 도착한 패킷은 웹 서버에서 돌아온 패킷이라고 생각하기 쉽지만, 실제로는 컴퓨터 내부에서 복수의 프로그램이 동시에 작동해 복수의 통신동작이 함께 진행되고 있으므로 수신 패킷은 다른 애플리케이션의 것일 수도 있다. but 노 상관.
- LAN 드라이버는 이런 것에 신경쓰지 않고 타입 필드의 값에 대응하는 프로토콜 스택에 패킷을 건네주기만 한다.
- 프로토콜 스택이 어느 애플리케이션에 대응하는 패킷인지 판단하여 적절한 조치를 취한다.

▼ 11) 서버의 응답 패킷을 IP에서 TCP로 넘긴다.

- 웹 서버에서 패킷이 돌아온 것으로 간주하고 다음 프로토콜 스택의 동작을 추적.
- 서버에서 반송된 패킷의 타입은 0800이므로 LAN 드라이버는 TCP/IP의 프로토콜 스택에 패킷을 건넬 것.
- IP 담당 부분은 IP 헤더 부분부터 조사하여 포맷에 문제가 없는지 확인하고 수신처 IP 주소를 조사한다.
- 패킷을 수신한 기기가 윈도우의 클라이언트 PC였으면 서버에서 회신된 패킷의 수신처 IP 주소는 수신한 LAN 어댑터에 할당된 주소와 일치할 것이므로 이것을 확인하고 수신.

- 수신처 IP 주소가 자신의 주소와 다르면 오류가 있는 것.
- 클라이언트 PC의 경우 패킷을 중계하지 않으므로 자신에게 올 것 이외의 패킷이 도착하는 일이 없기 떄문.
- 이런 오류가 발생했을 때는 IP 담당 부분이 ICMP라는 메세지를 사용해 통신 상대에게 오류를 통지하게 되어 있다.
- ICMP에는 여러 타입의 메세지가 정의되어 있으며 Destination unreachable이라는 메세지를 통지.



- 수신처 IP 주소가 올바르면 수신하지만, 한 가지 일이 더 있다.
- IP 프로토콜에는 조각 나누기라는 기능이 있는데, 패킷을 운반하는 도중에 통신 회선이나 LAN 중에는 짧은 패킷만 다룰 수 있는 것이 있다.
- 따라서 패킷을 짧게 하기 위해 하나의 패킷을 여러 개로 분할 하는 경우가 있다.
- 수신한 패킷이 분할된 것이면 IP 담당 부분은 그것을 원래 패킷으로 되돌린다.
- 분할된 패킷은 IP 헤더에 있는 플래그라는 항목을 보면 알 수 있으므로 수신 패킷이 분할된 것이면 IP 담당 부분 내부의 메모리에 일시적으로 보관한다.
- 그리고 IP 헤더에 있는 ID 정보에 같은 값을 가진 패킷이 도착하기를 기다리고, 분할된 패킷은 ID 정보의 값이 모두 같은 값인 패킷이므로 이것을 참조함.
- 프래그먼트 오프셋이라는 항목에는 패킷이 원래 패킷의 어느 위치에 있었는지를 나타내는 정보가 들어 있다.
- 이런 정보를 바탕으로 분리된 패킷이 전부 도착하기를 기다렸다가 패킷을 원래의 모습으로 되돌리는 동작을 리어셈블링이라고 함.
- IP 담당 부분의 역할은 끝나므로 리어셈블링이 끄나면 패킷을 TCP 담당 부분에 건네줌.
- 그러면 TCP 담당 부분은 IP 헤더에 기록된 수신처 IP 주소와 송신처 IP 주소, TCP 헤더에 기록된 수신처 포트 번호 및 송신처 포트 번호의 네 가지 항목을 조사하여 해당하는 소켓을 찾음.
- 소켓을 찾으면 통신의 진행 상태가 기록되어 있으므로 적절한 동작을 실행한다.
- 애플리케이션의 데이터를 넣을 패킷이 있다면 수신 확인 패킷을 반송한 후 데이터를 수신 버퍼에 저장하고 애플리케이션이 가지러 오기를 기다림.
- 접속, 연결 끊기 단계라면 응답의 제어용 패킷을 반송하거나 접속 및 연결 끊기 동작의 상황을 애플리케이션에 통지.