작성자(학번 이름) : 2019180052 한창우

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 2주차 | **기간** | 2024.7. 6 ~ 8 | **지도교수** |  |
| 이번주 한일 요약 | 인프런에서 네트워크 이론 강의 수강 | | | | |

<상세 수행내용>

* 네트워크 역사

프로토콜이란 약속을 뜻한다. 그리고 네트워크에서 TCP/IP는 컴퓨터끼리 통신하기 위한 프로토콜 이다. 발신자와 수신자 사이에서 데이터를 어떻게 정리하고 처리하는지 약속한 것.  
네트워크는 단순히 소프트웨어로만 이루어지는 것이 아니라 하드웨어까지 연결되는 굉장히 복잡한 시스템이며, 그래서 네트워크를 여러계층으로 나누었다.

여러 컴퓨터가 서로 데이터를 공유하면 더 효과적으로 작업할 수 있겠다는 아이디어를 생각하면서 네트워크 라는 기술이 탄생 되었다. 처음에 사용한 프로토콜은 ARPANET에서 사용된 NCP인데, 이는 단점들이 있었고, 이러한 단점을 해결하기 위해 개발된 프로토콜이 현재의 TCP/IP이다.

1985년에는 유닉스 운영체제에 TCP/IP가 기본적으로 내장되었고 이로써 TCP/IP는 현재 인터넷을 이루는 주류가 된 것이다.

TCP/IP는 복잡한 네트워크를 다양한 제조사와 사람들이 쉽게 사용할 수 있도록 네트워크의 기능을 크게 나누었으며, TCP/IP는 5계층을 나눌 수 있다.



* 물리 계층 : 데이터를 전기신호로 변환하고 전기신호를 다시 데이터로 복원 (데이터를 물리적으로 전송) (랜카드, 케이블)
* 데이터 링크 계층 : 여러개의 기기가 연결돼 있더라도 MAC 주소라는 고유한 주소로 기기를 분류
* 네트워크 계층 : 복잡하게 연결된 네트워크에서 IP주소를 이용해 출발지부터 목적지까지 정확하게 데이터를 보냄
* 트랜스포트 계층 : 두 컴퓨터 사이에 신뢰성있는 전송을 보장하고, 포트를 이용해 사용자의 특정 애플리케이션으로 정확하게 전달
* 애플리케이션 계층 : 사용자(프로그래머)가 다루는 애플리케이션

이렇게 계층화 되어있기 때문에 프로그래머는 네트워크를 몰라도 본인의 개발에만 집중할 수 있다. 하지만 서버 프로그래머는 알아야한다.

* 서버와 클라이언트

서버 : 제공하다

클라이언트 : 의뢰인

여기서 사용되는 개념 또한 똑같다. 클라이언트 요청에 파일을 저장하거나 제공해주는 역할을 서버가 하는 것이다. 결국 **“누가 요청을 하고 누가 응답을 하는가”** 이다.

우리가 구글을 검색하면 구글 서버로 데이터를 요청하고, 다시 요청한 데이터를 받는 일련의 과정은 다음과 같다.

1. 컴퓨터에 연결된 LAN선을 통해 유무선 공유기로 전달

2. 공유기 → 통신사(ISP)에서 설치한 모뎀과 연결

3. 통신사에 설치한 모뎀은 중간배선반으로 연결 된다.

4. 중간배선반은 주배선반으로 연결된다. (주배선반에는 아파트의 모든 케이블들이 연결되어 있다.)

5. 주배선반은 ISP(Internet Service Provider), 즉 통신사들과 연결된다.

6. 한국 ISP는 미국 ISP와 해저 광케이블을 설치해서 연결된다.

7. 한국 ISP에서 보내진 데이터는 미국 ISP에 연결된 구글의 웹 서버로 전달된다.

8. 구글의 웹 서버는 요청에 응답해 웹 페이지 데이터를 요청한 컴퓨터로 다시 전송한다.

모뎀 : 홈네트워크를 인터넷에 연결하는 장치

라우터 : 여러 장비를 연결해주는 장치 (중앙 허브 역할)

* 네트워크 연결 방식

네트워크를 구성하는 장비와 ISP가 어떻게 연결되느냐에 따라서 형태는 다양하다.

네트워크의 다양한 형태를 “토폴로지” 라고 한다.

1. 버스 토폴로지 : 하나의 중앙 통신 회선에 여러 대의 노드를 연결한 형태

1. 장점 : 노드간에 영향을 주지 않는다.

2. 단점 : 중앙 통신 회선을 공유하기 때문에 트래픽이 증가하면 충돌이 발생해 성능 하락

2. 스타 토폴로지 : 중앙 노드를 중심으로 다른 모든 노드가 연결되는 형태

1. 장점 : 문제가 생기면 발견하기 쉽고 관리가 편함

2. 단점 : 주 노드에 문제가 발생하면 전체 네트워크가 다운 되는 것이다.

3. 링 토폴로지 : 좌우 인접한 노드와 연결되어 원형을 이루는 형태

1. 장점 : 하나의 노드에서 다른 하나의 노드로만 데이터를 전송할 수 있으므로 충돌할 위험이 거의 없다

2. 단점 : 1. 하나의 노드에 문제가 발생하면 전체 네트워크에 문제 발생

2. 노드를 제거하거나 새로운 노드를 추가하기 힘들다.

4. 트리 토폴로지 : 트리 자료구조의 형태

1. 장점 : 노드를 제거하거나 추가하는 것이 쉬움

2. 단점 : 1. 스타 토폴로지와 마찬가지로 루트 노드에 문제가 생기면 전체 네트워크가 다운됨.

2. 특정 노드에 트래픽이 집중되면 성능 저하

5. 매쉬 토폴로지 : 모든 노드가 직접 연결되는 형태

1. 장점 : 특정 노드에 문제가 발생하더라도 전체 네트워크엔 이상이 없다.

2. 단점 : 모든 노드를 연결해야하므로 비용이 많이 든다.

- **네트워크 통신 방식**

유니캐스트, 브로드캐스트, 멀티캐스트, 애니캐스트

1. 유니캐스트 : 노드끼리 1:1 통신 방식

2. 브로드캐스트 : 출발지에서 연결된 모든 노드에게 데이터를 전송하는 방식

3. 멀티캐스트 : 1:N으로 데이터를 전송하는 방식 (유니캐스트와 브로드캐스트의 단점을 보완한 방식), 유니캐스트처럼 데이터를 여러 번 보낼 필요도 없고 브로드캐스트처럼 억울하게 CPU를 사용하게 만드는 일도 없다. 하지만 장비가 멀티캐스트를 지원해야 한다는 단점이 있다.

4. 애니캐스트 : 네트워크에 연결된 노드 중 가장 가까운 노드에만 전송하는 방식

* **물리 계층에 해당 되는 것**

**케이블**

**리피터**

유선으로 데이터를 전송할 때, 거리가 멀어지면 신호가 감쇠되고 왜곡될 수 있습니다. 리피터를 사용하면 이러한 신호 감쇠와 왜곡을 방지할 수 있으며, 데이터를 더 먼 거리로 전송할 수 있습니다.

**허브**

컴퓨터를 여러 대 연결해야한다고 할 때, 연결 할 때 마다 랜카드를 장착하는건 말이 안된다.

그래서 등장한 장치가 허브 이다.

허브는 여러 대의 컴퓨터를 연결하는 중심이 되는 장치 이다. 컴퓨터는 서로 연결하기 위해서 서로 연결하는 것이 아닌 중간에 있는 허브에 케이블로 연결하는 것이다.

그리고 허브에서는 전달 되는 신호를 다시 증폭까지하여 리피터 역할도 하게 된다.

허브는 충돌이 발생하지 않도록 자신으로 전송되는 데이터가 없을 때까지 기다렸다가 허브로 데이터를 보내는데 이를 CSMA/CD 라고 한다. 이러한 충돌이 발생할 수 있는 영역을 “콜리전 도메인” 이라고 한다. 아무튼 이러한 한계 때문에 스위치와 브릿지가 탄생하게 되었다.

* **데이터 계층에 해당 되는 것**

**랜카드** ( Network Interface Card)는 하드웨어로 데이터링크 계층에 해당하는 장치이다.

랜카드는 컴퓨터와 네트워크 사이에서 디지털 데이터를 전송하는 장치입니다. 데이터는 전송할 때 디지털 신호로 유지되며, 랜카드는 이를 전기적으로 처리하여 네트워크를 통해 전송합니다.

랜카드는 고유의 주소를 나타내는 MAC 주소를 가지고 있다. 데이터를 받으면 MAC 주소를 확인하여 자신에게 온 것인지 확인하고 맞다면, 디지털 신호로 변환하고 CPU에게 건네준다.

**브리지**

동축케이블을 이용해 버스 토폴로지 방식으로 통신하면 CSMA/CD 방식으로 반이중통신만 가능하다는 단점이 있습니다. UTP 케이블과 허브를 이용해 스타 토폴로지를 구성해도 마찬가지이다.

그래서 생각한 방안은 다음과 같다.

허브는 들어온 데이터를 브로드캐스트 하는 기능만 있다는 것이 문제이다.

그래서 허브에 메모리와 이를 처리하는 프로세서(CPU)를 탑재하여 브리지 라는 것이 탄생했다.

브리지는 허브의 기능에서 MAC 주소를 이용해 해당 포트로만 전송할 수 있게 되었다.

**스위치**

스위치는 다음과 같은 동작으로 주소를 저장하고 데이터를 이동시킨다.

* Learning
* Flooding
* Forwarding
* Filtering
* Aging

**스위치와 브리지의 차이**

- 스위치는 브리지보다 더 많은 포트를 가지고 있다.

- ASIC 이라는 하드웨어로 스위칭 작업을 수행하므로 소프트웨어적으로 처리하는 브리지보다 속도가 더 빠르다.

- 스위치는 오류를 체크하는 기능이 있지만 브리지는 없다.

- 스위치는 버퍼를 가지고 있지만, 브리지는 없는 경우도 있다.

- 각 포트마다 별도의 충돌 도메인을 제공합니다. 이는 각 포트에 연결된 장치들이 독립적으로 데이터를 전송할 수 있음을 의미합니다.

**스위치의 한계**

스위치가 브로드캐스트를 전혀 안하는 것은 아니다. 그 이유는 아직 모르는 주소는 Learing 또는 Flooding을 통해 MAC 주소를 테이블에 저장하기 때문이다. 따라서 스위치와 연결된 영역을 브로드캐스트 도메인이라고 부르며, 또한 브로드캐스트 도메인은 근거리 네트워크, 즉 LAN(Local Area Network) 영역이라고도 부른다.

만약 스위치를 계속 연결해 수십만 대의 PC가 브로드캐스트 영역에 있게 되면, 쓸모없는 트래픽이 증가하고 성능 저하가 생긴다.

* **네트워크 계층에 해당 되는 것**

**라우터**

허브와 스위치로 연결되어 하나의 브로드캐스트 도메인에 있는 네트워크에선 MAC 주소만으로 통신할 수 있지만 다른 영역의 도메인으로 보낼 수 없다. 그래서 라우터라는 것이 나오게 되었다.

라우터는 브로드캐스트 도메인을 나눈다.

기존에 사용하던 출발지, 목적지 MAC 주소만으로는 다른 LAN 영역에 있는 컴퓨터에 전송하기가 힘들다. 그래서 논리적인 주소를 하나 더 추가해야겠다는 결론이 나왔고 그것이 “IP 주소” 이다.

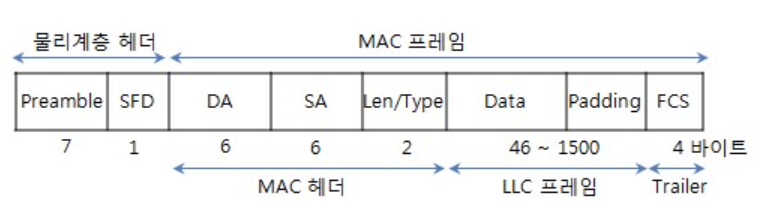
그리고 IP주소를 활용해 LAN 영역 사이를 문제없이 통신하도록 하는 장치가 라우터 이다.

라우터를 넘어가지 않는 영역을 LAN 영역이라고 부르고 라우터를 넘어가는 영역을 WAN(인터넷) 여역이라고 부른다.

* **이더넷과 이더넷 헤더 (데이터링크 계층)**

데이터링크 계층에서 현재 많이 사용되는 프로토콜은 “이더넷” 이다.

이더넷은 구조가 간단하고 장비가 저렴해 데이터링크 계층을 대표한다.



* Data 필드는 네트워크 계층에서 전달된 데이터, 즉 패킷을 말한다.
* DA(Destination Address) 와 Len/Type 필드까지를 이더넷 헤더라고 부른다.
* FCS는 “이더넷 트레일러” 라고 부른다. FCS는 데이터가 손상되었는지 체크하기 위한 필드 이다.

그리고 이더넷 헤더와 트레일러가 붙은 전체의 데이터를 프레임이라고 부른다.

* 프리앰블과 SDF(Start Frame Delimiter) 라는 필드는 랜카드에게 이더넷 프레임이 시작되는 것을 알리는 역할을 한다.
* DA가 만약 모두 1(FF-FF-FF-FF-FF-FF)로 되어있다면 이 데이터를 브로드캐스트 하라는 뜻이다.
* Len/Type 필드에는 상위계층의 프로토콜 타입이나 현재 프레임의 데이터 길이가 저장되는데, 만약 이 값이 1500 이하라면 데이터의 길이를, 1500 보다 크다면 상위 프로토콜의 타입을 나타낸다.

이더넷 프레임의 크기는 Data 필드에 따라서 64 ~ 1518 바이트가 된다.

* **IP 클래스와 서브넷 마스크**

IP주소는 대략 43억 개가 될 수 있다. 그래서 이 주소를 효율적으로 관리하기 위해서 클래스가 등장했다. 클래스는 A, B, C, D 총 4가지가 있다.

* 클래스 A : 가장 앞에 있는 비트가 0으로 시작하는 클래스이다. 이 비트를 네트워크 주소로 사용하고 나머지 24비트는 호스트 주소로 사용한다. 네트워크 주소를 나타내는 8비트에서 가장 첫 번째 비트는 0으로 고정한다. 클래스 A는 1600만 개의 호스트를 지정할 수 있는 네트워크가 128개 있다.
* 클래스 B : 32비트 중에서 가장 앞에 있는 2비트가 10으로 시작하는 크래스 이다.
* 클래스 C : 32비트 중에서 가장 앞에 있는 3비트가 110으로 시작하는 클래스 이다.
* 클래스 D : 32비트 중에서 가장 앞에 있는 4비트가 1110으로 시작하는 클래스 이다. (멀티캐스트 용도로 사용된다) 그래서 일반 사용자의 IP 주소로 할당되는 일은 없다.
* **네트워크 주소와 호스트 주소를 나눌 때 표기 방법?**

예) 클래스 A : 10 . 1 . 2 . 3

가장 앞에 10이 네트워크 주소이다. 네트워크 주소를 나타내는 8비트를 IP주소 뒤에 슬래시(/)를 적어서 표현한다. “10.1.2.3/8” 이렇게.

예) 클래스 C : 192.168.0.1

→ 192.168.0.1/24

* **IP 주소 클래스의 한계?**

사내 통신을 하기 위해선 네트워크를 구축하고 각 호스트에 IP를 할당해야 한다.

회사에서 필요한 IP 주소가 300개라고 가정한다면 클래스 C는 사용할 수 없다.

따라서 회사에서는 국제인터넷주소관리기구에 클래스 B를 요청한다.

근데 고작 300개의 IP를 할당하고자 클래스 B를 주면 6만개가 넘는 주소가 놀게 된다.

그래서 이러한 낭비를 막기 위해서 **“서브넷 마스크”** 라는 개념이 나왔다.

즉, IP를 관리하기 쉽게 클래스로 나눴지만 부족한 IP를 아껴쓰기 위해서 서브넷 마스크가 등장하게 된 것이다.

**서브넷 마스크 : 클래스로 나뉘어서 낭비되는 공간을 다시 한번 나누는 기술**

* **NAT과 PAT**

부족한 주소를 절약하기 위해서 IP 대역의 일정 부분은 Private IP를 사용한다.

즉, Private IP는 로컬 네트워크(LAN)에서만 사용하기로 약속함. 따라서 다른 LAN 끼리는 Private IP가 똑같을 수도 있다.

집 내부에서는 Private IP를 사용하고, 외부와 통신하는 곳에 부여하는 고유한 IP를 Public IP라고 한다. 그래서 Private IP를 Public IP로 변환하는 것을 NAT 이라고 부른다.

NAT은 Static NAT, Dynamic NAT, Static PAT, Dynamic PAT 이렇게 4개로 나뉜다.

* Static NAT : 관리자가 라우터에 직접 변환 Private IP 주소에 Public IP를 매핑하는 방식 (주로 보안 목적으로 사용된다)
* Dynamic NAT : 관리자가 여러 개의 Public IP를 준비해두고 요청한 호스트에게 여분의 Public IP를 빌려주는 방식. 만약 여분의 IP가 없다면 통신하고 싶은 노드가 있어도 못할 수도 있다는 단점이 있어서 요즘은 사용되지 않는 방식이다.
* Static PAT : Private IP 주소와 포트 번호를 고정된 공용 IP 주소와 포트 번호에 매핑하는 NAT 방식이다. 하나의 Public IP 주소에 여러 Private IP 주소를 매핑할 수 있다는 장점이 있다. 하지만 관리자가 일일이 IP주소와 포트를 지정해줘야하는 번거로움이 있다.
* Dynamic PAT : Static PAT과 비슷하지만, 포트를 자동으로 할당해준다. 주소와 포트를 할당할 때, 라이프타임도 같이 할당하여 일정 시간이 지나면 매핑테이블에서 소멸된다. (다른 컴퓨터에게도 주소와 포트를 할당해주기 위함)

PAT(Port Address Translation) : IP뿐만 아니라 트랜스포트 계층의 포트까지 이용해 매핑테이블을 작성하는 방법

* **TCP**

문제가 발생하여 패킷이 왜곡되거나 분실되는 상황이 발생할 수 있기 때문에 나온것이 TCP 프로토콜이다. TCP는 신뢰성을 보장하는 프로토콜이다.

TCP는 상대와 통신하기 위해서 3 way handshake로 커넥션을 만든다.

3 way handshake의 과정은 다음과 같다.

A와 B가 있다.

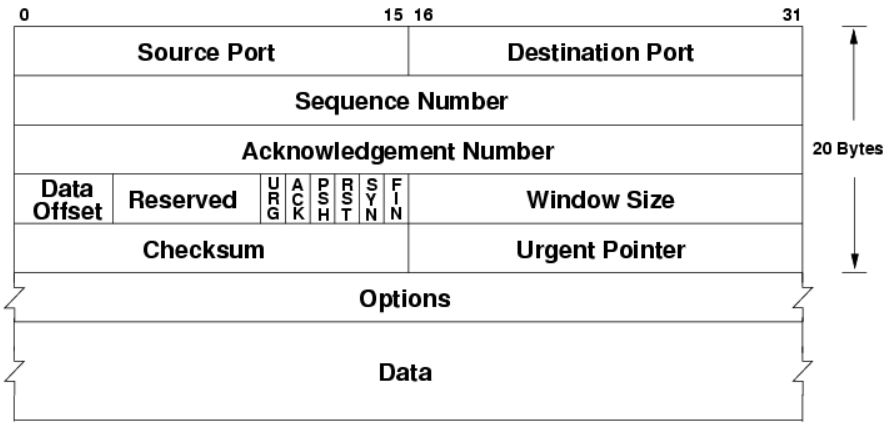
1. A가 SYN(Synchronization) 패킷을 보낸다.

2. B는 ACK(acknowledgement) 패킷을 보낸다.

3. A가 다시 ACK 패킷을 B에게 보낸다.

위와 같이 데이터가 3번 송수신되어서 3 way handshake라고 부르는 것이다.

* **TCP 헤더**



* Source Port : 송신측의 포트
* Destination Port : 수신측의 포트
* Sequence Number : 데이터의 일련 번호. 시퀀스 넘버가 있어서 데이터의 순서가 뒤죽박죽으로 도착해도 원상태로 복원할 수 있고, 누락된 데이터도 알 수 있다. 시퀀스 넘버는 처음 보낼 때 난수로 결정 되고 그다음부터는 현재 시퀀스 넘버와 데이터의 크기에 1을 더한다.
* Acknowledgement Number : 자신이 다음에 받아야 할 상대의 시퀀스 넘버가 된다.
* 데이터 오프셋 : 현재 TCP 데이터가 TCP Segment의 가장 앞에서 얼마나 떨어져 있는지를 4바이트 단위로 나타낸다.
* Window Size : 이 필드는 TCP의 흐름제어인 슬라이딩 윈도우를 사용할 때 사용된다.
* Checksum : 데이터의 변조나 에러를 알아내기 위한 용도로 사용된다.
* Urgent Pointer : 애플리케이션에서 이 포인터가 가리키고 있는 데이터를 우선 처리 할 수도 있고 하지 않을 수도 있다. 결정하는 애플리케이션 개발자 마음이다.
* Options : 이 필드는 TCP의 기능을 확장할 때 사용하는데 크기가 정해져 있지 않고 가변적이다.

진행 상황 유튜브 링크:

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** |  | | |
| **해결방안** |  | | |
| **다음주차** | 3주차 | **다음기간** | 7. 8 ~ 7. 14 |
| **다음주 할일** | 3주차에 계획되어있는 내용들 | | |
| **지도 교수**  **Comment** |  | | |