

Network Layer

백설기 2차 스터디

AI6/1반 조해정

지난 주 발표 자료: OSI 7 Layer

OSI 7 Layer

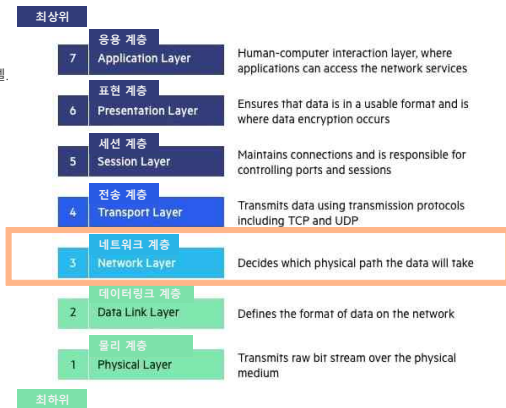
통신이 일어나는 과정을 7단계로 구분해서 정의한 모델.
모든 계층은 독립적, 상하구조를 가집니다.

◆ 상위층

- 응용 계층(Application Layer)
- 표현 계층(Presentation Layer)
- 세션 계층(Session Layer)

◆ 하위층

- 전송 계층(Transport Layer)
 - 네트워크 계층(Network Layer)
 - 데이터링크 계층(Data Link Layer)
 - 물리 계층(Physical Layer)
- 아파서 티내다, 피나다



글, 그림 참조:

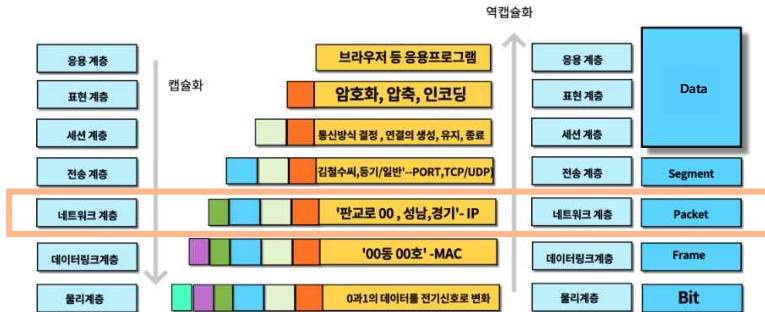
https://ko.wikipedia.org/wiki/OSI_%EB%AA%A8%ED%98%95

<https://www.imperva.com/learn/application-security/osi-model/>

CONTENTS

- Network Layer
- Forwarding & Routing
- Delay
- IPv4
- Subnet
- CIDR

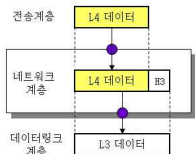
Network Layer



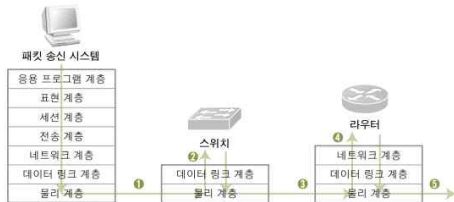
Network Layer

Network Layer

네트워크 계층은 다양한 길이의 패킷을 네트워크들을 통해 전달하고, 그 과정에서 전송 계층이 요구하는 서비스 품질(QoS)을 위한 수단을 제공하는 계층이다. 라우팅, 패킷 포워딩, 인터 네트워킹 등을 수행한다.



- 패킷을 네트워크를 통해 발신지에서 목적지까지 전달하기 위해, 라우팅 프로토콜을 사용하여 **최적의 경로** 선택한다.
- 데이터를 전송할 데이터의 주소 확인 후 전송 계층으로 전달한다.
- **프로토콜** : IP, ARP, RARP, ICMP, IGMP, 라우팅 프로토콜
- **장비** : 라우터(Router), L3 스위치



PC에서 네트워크 계층까지의 데이터의 통신 흐름

포워딩(Fowarding)과 라우팅(Routing)

Fowarding & Routing

for + ward : 앞으로 나아가게 하다.

route : 물품이나 정보 등이 전해지는 경로.

라우팅(Routing) : 루트(이동 경로)를 찾는 것을 말한다. 포워딩 테이블을 만들고 갱신한다.

- 네트워크 계층은 패킷이 발신지에서 목적지까지 갈 수 있도록 경로를 라우팅해야 한다.
- 물리적인 네트워크는 네트워크(LAN과 WAN)와 네트워크를 연결하는 라우터의 조합.
- 발신지에서 목적지까지 적어도 하나 이상의 라우터가 있다는 의미.
- 네트워크 전반에 걸쳐서 일어난다.

포워딩(Fowarding) : 포워딩 테이블을 찾아보고, 그에 맞춰 패킷을 라우터의 입력 포트에서 출력 포트로 이동시키는 작업이다.

- 각 라우터의 의사결정 테이블을 만들기 위해 라우팅에 규칙을 적용하고 라우팅 프로토콜을 실행하는 작업.
- 라우터 상의 하나의 인터페이스로 패킷이 도착했을 때 라우터가 취하는 행동으로 정의.
- 라우터가 하나의 네트워크로부터 패킷을 수신하면 해당 패킷을 다른 하나의 네트워크로 포워딩하거나, 여러 네트워크로 포워딩 함
- 이런 결정을 위해 라우터는 패킷 헤드에 있는 목적지 주소나 레이블 정보를 사용하여 포워딩 테이블에서 상응하는 출력 인터페이스 번호를 찾는다.
- 매우 짧은 시간동안 하드웨어에서 실행.

글 참조:

Oxford Languages

<https://ish0301.tistory.com/entry/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EC%A0%95%EB%A6%AC-5-%ED%8C%A8%ED%82%B7%ED%99%94-%EB%9D%BC%EC%9A%B0%ED%8C%85-%ED%8F%AC%EC%9B%8C%EB%94%A9-%EC%A0%95%EB%A6%AC?category=1002841>

포워딩(Fowarding)과 라우팅(Routing)

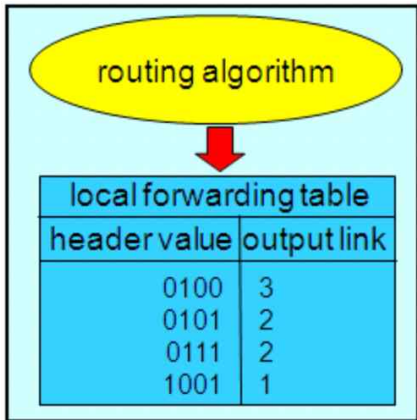
라우팅 테이블 & 포워딩 테이블

라우팅 테이블(Routing Table) :

- 라우팅 알고리즘을 통해 만들어지는 테이블.
- 라우팅 알고리즘: RIP나 OSPF 같은 라우팅을 만드는 방법(protocol).
- 라우팅이란 라우팅 알고리즘을 사용하여 포워딩 테이블을 만드는 작업이다.
- 라우팅 테이블을 만드는 과정이 출발지부터 목적지까지 경로를 찾는 일이 된다.

포워딩 테이블(Fowarding Table) :

- 라우팅 테이블을 참조하여 만들어지는 테이블이다.
- 많은 경로 중 비용이 작은 경로를 선택하는 라우팅 알고리즘을 통해 선택된 경로를 저장해 둔 것이 포워딩 테이블이다.



포워딩(Fowarding)과 라우팅(Routing)

Fowarding & Routing

for + ward : 앞으로 나아가게 하다.

route : 물품이나 정보 등이 전해지는 경로.

라우팅(Routing) : 루트(이동 경로)를 찾는 것을 말한다. 포워딩 테이블을 만들고 갱신한다.

- 네트워크 계층은 패킷이 발신지에서 목적지까지 갈 수 있도록 경로를 라우팅해야 한다.
- 물리적인 네트워크는 네트워크(LAN과 WAN)와 네트워크를 연결하는 라우터의 조합.
- 발신지에서 목적지까지 적어도 하나 이상의 라우터가 있다는 의미.
- 네트워크 전반에 걸쳐서 일어난다.

포워딩(Fowarding) : 포워딩 테이블을 찾아보고, 그에 맞춰 패킷을 라우터의 입력 포트에서 출력 포트로 이동시키는 작업이다.

- 각 라우터의 의사결정 테이블을 만들기 위해 라우팅에 규칙을 적용하고 라우팅 프로토콜을 실행하는 작업.
- 라우터 상의 하나의 인터페이스로 패킷이 도착했을 때 라우터가 취하는 행동으로 정의.
- 라우터가 하나의 네트워크로부터 패킷을 수신하면 해당 패킷을 다른 하나의 네트워크로 포워딩하거나, 여러 네트워크로 포워딩 함
- 이런 결정을 위해 라우터는 패킷 헤드에 있는 목적지 주소나 레이블 정보를 사용하여 포워딩 테이블에서 상응하는 출력 인터페이스 번호를 찾는다.
- 매우 짧은 시간동안 하드웨어에서 실행.

글 참조:

Oxford Languages

<https://ish0301.tistory.com/entry/%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC-%EC%A0%95%EB%A6%AC-5-%ED%8C%A8%ED%82%B7%ED%99%94-%EB%9D%BC%EC%9A%B0%ED%8C%85-%ED%8F%AC%EC%9B%8C%EB%94%A9-%EC%A0%95%EB%A6%AC?category=1002841>

네트워크 성능 측정

네트워크 계층을 사용하는 상위 계층 프로토콜은 이상적인 서비스를 원하지만
네트워크 계층은 항상 완벽하지 않다.

네트워크의 성능은 **지연(Delay)**, **처리량(Throughput)**, **패킷 손실률(Packet Loss)**
로 측정 가능하다.

지연(Delay) : 전송이 지연되는 것을 말한다.

- 전송 지연 (Transmission Delay)
- 전파 지연 (Propagation Delay)
- 처리 지연 (Processing Delay)
- 큐 내부의 지연 (Queuing Delay)
- 전체 지연 (Total Delay)

전송 지연(Transmission Delay)

- 발신지 호스트나 라우터는 패킷을 즉시 보낼 수 없다.
- 송신자는 전송해야 할 패킷에 하나하나 비트 정보들을 추가한다.
- $\text{Delay}(tr) = (\text{패킷 길이 Packet Length}) / (\text{전송 비율 Transmission Rate})$

전파 지연(Propagation Delay)

- 전송 매체를 통해 A지점에서 B지점까지 1비트가 전달되는데 걸리는 시간을 말한다.
- $\text{Delay}(pg) = (\text{거리 Distance}) / (\text{전파 속도 Propagation Speed})$

처리 지연(Processing Delay)

- 라우터나 목적지 호스트가 입력 포트로 패킷을 받고, 헤더를 제거하고, 오류 탐지를 수행한 뒤, 출력 포트로 패킷을 보내거나 상위 계층 프로토콜로 패킷을 전달하는데 걸리는 시간
- $\text{Delay}(pr) = \text{라우터 혹은 목적지 호스트의 패킷을 처리하는데 필요한 필수 시간 (Time required to process a packet in a router or a destination host)}$

큐 내부의 지연(Queuing Delay)

- 일반적으로 큐 내부의 지연은 라우터에서 발생
- 라우터는 각 입력 포트에 처리할 패킷을 보관할 큐와, 출력 포트에 전송할 패킷을 보관할 큐를
- 각각 보유하고 있다.
- 큐 내부의 지연은 라우터의 입력 큐와 출력 큐에서 패킷이 대기하는 시간을 측정한다.
- **Delay(qu) = 라우터 안에 있는 입력, 출력 큐의 대기시간 (The time a packet waits in input and output queues in a router)**

전체 지연(Total Delay)

- 전체 경로상의 라우터 개수 N 을 알고, 송신자, 라우터, 수신측의 각각 지연을 가지고 있다고 가정할 때, 패킷에 발생할 전체 지연은 아래와 같다.

$$\text{Total Delay} = (N+1)(\text{Delay(tr)} + \text{Delay(pg)} + \text{Delay(pr)}) + (N)(\text{Delay(qu)})$$

- 여기서 N 개의 라우터가 있다면 $(N+1)$ 개의 링크가 존재한다.
- N 개의 라우터와 발신지와 관련하여 $(N+1)$ 개의 전송 지연이 발생하고, $(N+1)$ 개의 링크와 관련하여 $(N+1)$ 개의 전파 지연이 발생하며, N 개의 라우터와 목적지와 연관하여 $(N+1)$ 개의 처리 지연이 발생한다.
- N 개의 라우터에서 N 개의 큐 내부의 지연이 발생한다.

IP 주소

인터넷 상에서 통신하기 위해 각각의 컴퓨터 및 통신장비에 부여하는 고유한 주소.

- IPv4
- IPv6

IPv4 주소



IPv4주소는 **32비트 길이**의 식별자로 0.0.0.0~255.255.255.255까지의 숫자의 조합으로 이루어진다.

총 **네구간**으로 나뉘져있으며 최대 12자리의 번호로 이루어져 있다.

A, B, C 등 클래스 단위의 비순차적 할당.

IPv4를 통해 최대 약 43억개의 서로 다른 주소를 부여할 수 있다. 하지만 전세계 공용으로 사용되며 인터넷 사용자수가 급증하면서 IPv4 주소가 고갈될 문제에 처해있다. 이러한 고갈 문제를 해결하기 위해 등장한 주소가 바로 IPv6(128비트)이다.

A, B, C.. 클래스

IP 주소 공간을 5개 클래스로 구분

몇 비트로 주소의 클래스 확인 가능

Class	주소 개수	점유율	주소 범위
A	(2,147,483,648)	50%	0.0.0.0 ~ 127.255.255.255
B	(1,073,741,824)	25%	128.0.0.0 ~ 191.255.255.255
C	(536,870,912)	12.5%	192.0.0.0 ~ 223.255.255.255
D	(268,435,456)	6.25%	224.0.0.0 ~ 239.255.255.255
E	(268,435,456)	6.25%	240.0.0.0 ~ 255.255.255.255

10진수 대신 비트로 치환해서 본다면:

- A클래스는 00000000.00000000.00000000.00000000
- B클래스는 10000000.00000000.00000000.00000000 (128)
- C클래스는 11000000.00000000.00000000.00000000 (192)
- D클래스는 11100000.00000000.00000000.00000000 (224)
- E클래스는 11110000.00000000.00000000.00000000 (240)

A, B, C.. 클래스 문제점

몇 비트로 주소의 클래스 확인 가능

Class	주소 개수	점유율	주소 범위
A	(2,147,483,648)	50%	0.0.0.0 ~ 127.255.255.255
B	(1,073,741,824)	25%	128.0.0.0 ~ 191.255.255.255
C	(536,870,912)	12.5%	192.0.0.0 ~ 223.255.255.255
D	(268,435,456)	6.25%	224.0.0.0 ~ 239.255.255.255
E	(268,435,456)	6.25%	240.0.0.0 ~ 255.255.255.255

- 클래스 A 네트워크는 총 128개가 있고, A클래스 하나 마다 사용 할 수 있는 아이피의 개수는 16,777,216개이다. 초기에 미국의 대형 기업들이 선점해갔다.
- 클래스 B 네트워크는 16,384개가 있다. B클래스 하나당 사용 할 수 있는 아이피는 65,536개이다.
- C클래스는 사용할 수 있는 IP가 256개 밖에 안된다.
- C클래스를 받기엔 너무 모자라고, B클래스를 받기엔 너무 많이 남는 식의 문제가 발생하였다. 그래서 이 클래스에 따른 IP분배는 1993년까지만 쓰이고 지금은 쓰이지 않는다. 지금은 서브넷을 이용한 분배를 사용한다.

서브넷(Subnet)

- 서브넷은 하나의 네트워크가 분할되어 나뉜 작은 네트워크이다.
- 서브넷을 만들기 위해 네트워크를 분할하는 것을 **서브네팅(Subnetting)**이라고 한다.
- 서브네팅을 하면 IP 할당 범위를 더 작은 단위로 쪼갤 수 있게 된다.
그래서 만일 ip가 100개만 필요하다면 C클래스(256개)를 더 쪼개서 줄 수 있다.
- 그리고 이 서브네팅을 **서브넷 마스크(Subnet Mask)**를 통하여 계산되어 수행된다.

서브넷 마스크(Subnet Mask)

- 서브넷 마스크는 ip주소에서 네트워크 ID와 호스트 ID로 구분하기 위한 목적으로 만들어 졌다. 보다 쉽게 구분하기 위해서 고유의 표기법으로 구분한다.
- 서브넷 마스크는 IP주소와 똑같은 **32비트** 2진수로 표현된다.
- 아이피와 표현이 다른 점은 서브넷 마스크는 **연속된 1과 연속된 0**으로 구성되어있다.

255.255.255.0

글 참조:

[https://inpa.tistory.com/entry/WEB-IP-%ED%81%B4%EB%9E%98%EC%8A%A4-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%88%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC-%EC%84%9C%EB%B8%8C%EB%84%B7_%EC%84%9C%EB%B8%8C%EB%84%B7_%EB%A7%88%EC%8A%A4%ED%81%AC/_%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC](https://inpa.tistory.com/entry/WEB-IP-%ED%81%B4%EB%9E%98%EC%8A%A4-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%88%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC-%EC%84%9C%EB%B8%8C%EB%84%B7_%EC%84%9C%EB%B8%8C%EB%84%B7_%EB%A7%88%EC%8A%A4%ED%81%AC/_%EC%84%9C%EB%B8%8C%EB%84%A4%ED%8C%85_%EC%B4%9D%EC%A0%95%EB%A6%AC)

서브넷(Subnet)

- 서브넷은 하나의 네트워크가 분할되어 나뉜 작은 네트워크이다.
- 서브넷을 만들기 위해 네트워크를 분할하는 것을 **서브네팅(Subnetting)**이라고 한다.
- 서브네팅을 하면 IP 할당 범위를 더 작은 단위로 쪼갤 수 있게 된다.
- 그래서 만일 ip가 100개만 필요하다면 C클래스(256개)를 더 쪼개서 줄 수 있다.
- 그리고 이 서브네팅을 **서브넷 마스크(Subnet Mask)**

서브넷 마스크(Subnet Mask)

- 서브넷 마스크는 ip주소에서 네트워크 ID를 구하기 위해서 고유의 표기법으로 구분한다.
- 서브넷 마스크는 IP주소와 똑같은 32비트이다.
- 아이피와 표현이 다른 점은 서브넷 마스크

◆ A Class

IP주소 : 10 . 10 . 10 . 10

S/M : 255 . 0 . 0 . 0



◆ B Class

IP주소 : 172 . 16 . 1 . 10

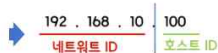
S/M : 255 . 255 . 0 . 0



◆ C Class

IP주소 : 192 . 168 . 10 . 100

S/M : 255 . 255 . 255 . 0

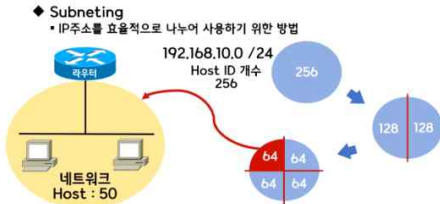


글 참조:

[https://inpa.tistory.com/entry/WEB-IP-%ED%81%B4%EB%9E%98%EC%8A%A4-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%B8%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC%EC%84%9C%EB%B8%8C%EB%84%B7-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%B8%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC](https://inpa.tistory.com/entry/WEB-IP-%ED%81%B4%EB%9E%98%EC%8A%A4-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%B8%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC%EC%84%9C%EB%B8%8C%EB%84%B7-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%B8%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%A4%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC)

서브네팅(Subnetting)

- IP주소를 효율적으로 나누어 사용하기 위한 방법
- 네트워크 성능 보장, 자원을 효율적으로 분배하기 위해 네트워크 영역과 호스트 영역을 **조개는** 작업을 행한다.
- 서브네팅을 하면 IP 할당 범위를 더 작은 단위로 나눌 수 있게 된다. (자신의 네트워크 주소를 더 작은 서브 네트워크로 **2의 배수로** 나누는 과정)



글, 그림 참조:

https://inpa.tistory.com/entry/WEB-IP-%ED%81%B4%EB%9E%98%EC%8A%A4-%EC%84%9C%EB%B8%8C%EB%84%B7-%EB%A7%88%EC%8A%A4%ED%81%AC-%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85-%EC%B4%9D%EC%A0%95%EB%A6%AC%EC%84%9C%EB%B8%8C%EB%84%B7_%EC%84%9C%EB%B8%8C%EB%84%B7_%EB%A7%88%EC%8A%A4%ED%81%AC/_%EC%84%9C%EB%B8%8C%EB%84%A4%ED%8C%85_%EC%B4%9D%EC%A0%95%EB%A6%AC

CIDR(Classless Inter-Domain Routing)

* 사이더

- CIDR은 클래스 없는 도메인간 라우팅 기법이라는 뜻이다.
- 즉, 도메인간의 라우팅에 사용되는 인터넷 주소를 원래 IP주소 클래스 체계를 쓰는 것보다 더욱 능동적으로 할 수 있도록 할당하여 지정하는 방식중 하나이다.
- 서브네팅, 슈퍼네팅 이러한 IP를 나누고 합치는 기법을 모두 CIDR라고 이해하면 된다.
 - 서브네팅: 네트워크 분할
 - 슈퍼네팅: 네트워크 합침

CIDR 표기법 : A.B.C.D/N

- CIDR는 네트워크 정보를 여러개로 나누어진 Sub-Network들을 모두 나타낼 수 있는 하나의 Network로 통합해서 보여주는 방법이다. 서브넷 마스크를 짧게 적은 prefix.
- 예) 192.168.10.70/26 ('/26'은 서브넷 마스크를 표현한 것: 앞에서부터 차례대로 1의 개수가 26개)
- 단 한줄만으로 네트워크 범위를 추측 또는 측정 할 수 있다.
- CIDR 계산기: <https://cidr.xyz/>

글, 그림 참조:

<https://inpa.tistory.com/entry/WEB-%F0%9F%8C%90-CIDR-%EC%9D%B4-%EB%AC%B4%EC%96%BC-%EB%A7%90%ED%95%98%EB%8A%94%EA%B1%B0%EC%95%BC-%E2%87%9B-%EA%B0%9C%EB%85%90-%EC%A0%95%EB%A6%AC-%EA%B3%84%EC%82%B0%EB%B2%95>

감사합니다