

HTTP



Contents

1

HTTP

2

RESOURCE & MIME

3

URI

4

TRANSACTION

5

MESSAGE & TCP/IP

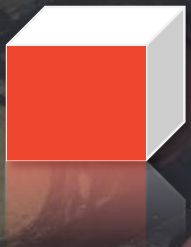


HTTP

웹 서버

하이퍼텍스트 문서

통신규약



HTTP?

HyperText Transfer Protocol

인터넷에서 웹 서버와 사용자의 인터넷 브라우저
사이에 하이퍼텍스트 문서를 교환하기 위하여 사용
되는 통신규약이다.



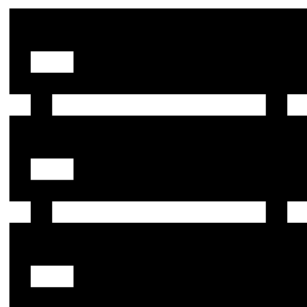
HTTP?

HyperText Transfer Protocol

웹 서버



웹 서버?

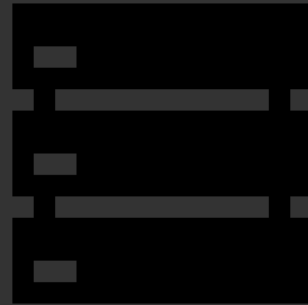


웹 서버!

사용자에게 웹(Web)을 제공하기 위한 서버로,
사용자가 웹으로 서비스를 요청하는 경우
네트워크를 통해 HTML로 구성된 웹 페이지를 제공한다.

하드웨어 측면

웹 사이트의 **컴포넌트 파일**들을 저장하는 컴퓨터
(**컴포넌트 파일**에는 HTML 문서, images, CSS stylesheets 그리고 JavaScript files 등)
이 파일들을 최종 소비자의 디바이스에 전달, 웹 서버는 인터넷에 연결되어 있고, URL 통해 접속가능



소프트웨어 측면

앞으로 설명하는 부분!

웹 서버!

사용자에게 **웹(Web)**을 제공하기 위한 서버로,
사용자가 **웹으로 서비스를 요청**하는 경우
네트워크를 통해 HTML로 구성된 웹 페이지를 제공한다.

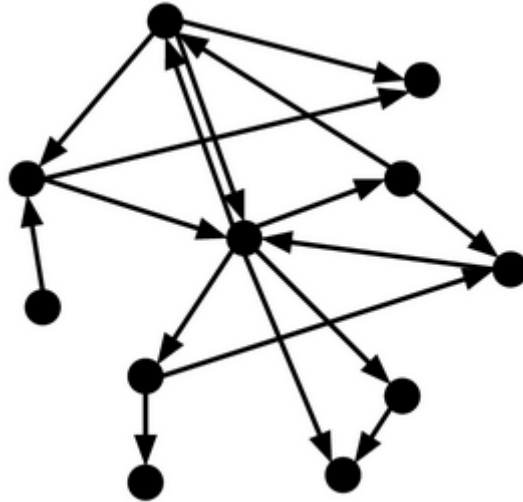


웹?



Web (=World Wide Web)

[= 세상의 크기만한 거미줄]

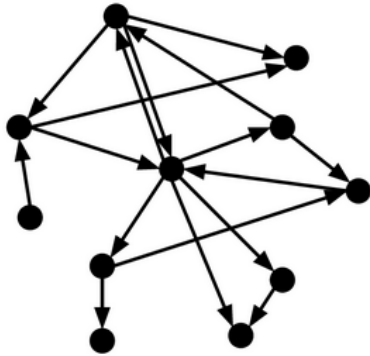


인터넷에 연결된 컴퓨터들을 통해 사람들이 정보를 공유할 수 있는 전 세계적인 정보 공간을 말한다. 이를 간단히 웹이라 부르는 경우가 많다.

웹은 전자 메일과 같이 인터넷 상에서 동작하는 하나의 서비스일 뿐이다.
그러나 1993년 이래로 웹은 인터넷 구조의 절대적 위치를 차지하고 있다.



Web ? HTTP ?



HTTP의 활용도에 있어 **Web**이 차지하는 비중이 워낙 높다 보니 두 단어를 같은 의미로 사용하는 사람이 많다.



Web ? HTTP ?

Web = URI + HTML + HTTP

Web 은 다음의 세가지 기능으로 포함시킨 서비스 개념



Web ? HTTP ?

Web = URI + HTML + HTTP

웹 자원의 위치를 식별하기 위한 방법 URL,



Web ? HTTP ?

Web = URI + HTML + HTTP

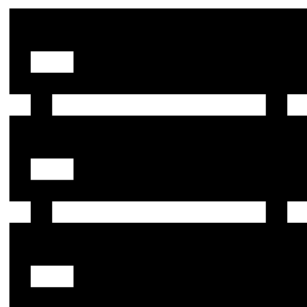
페이지가 어떻게 구조화되어 있는지
브라우저로 하여금 알 수 있도록 해주는 마크업 언어 HTML



Web ? HTTP ?

Web = URI + HTML + HTTP

웹의 자원에 접근하는 것에 관련된 통신규약 HTTP



웹 서버!

웹 서버는 유닉스 운영체제를 기반으로 개발되었지만
현재에는 윈도우, 리눅스 등 다양한 운영체제에도 설치되었다.
WAS를 구축해야하며 대표적으로 **아파치 (Apache)** 가 있다.






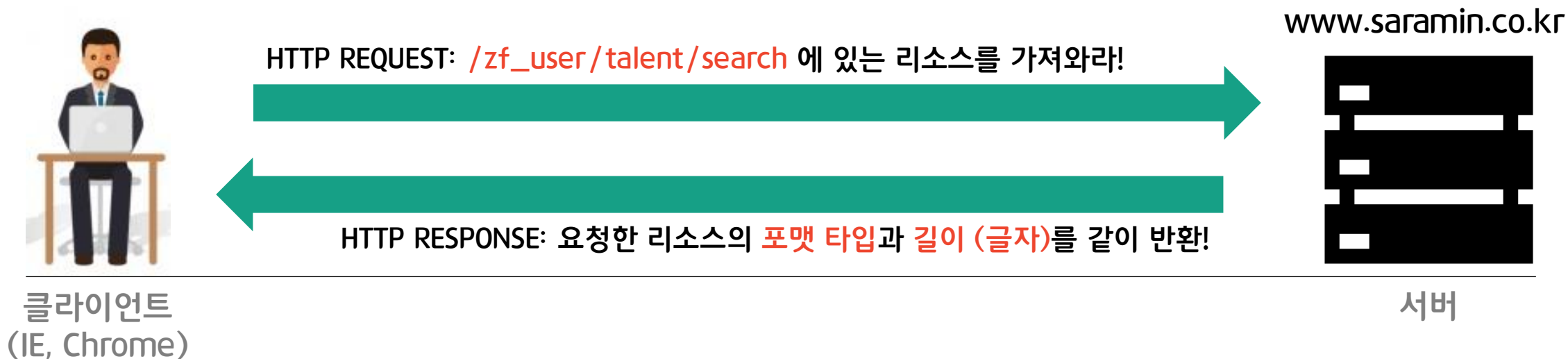
WAS?

웹 서버!

웹 서버는 유닉스 운영체제를 기반으로 개발되었지만
현재에는 윈도우, 리눅스 등 다양한 운영체제에도 설치되었다.
WAS를 구축해야하며 대표적으로 **아파치 (Apache)** 가 있다.



http: // www.saramin.co.kr / zf_user / talent / search



요청은 어떻게?

HTTP 클라이언트가! (Internet Explorer, Chrome...)



HTTP?

HyperText Transfer Protocol

하이퍼텍스트 문서



하이퍼텍스트 문서?



하이퍼텍스트 문서!

책이나 일반 워드에서 사용하는 순서를 가진 문서와 다르게
문서 안에 링크를 통해 연결되어 있기 때문에 쉽게 다른 문서로
이동할 수 있는 것이 하이퍼 텍스트다.



HTTP?

HyperText Transfer Protocol

통신규약



통신규약!

정보를 원활하게 교환할 수 있게 하기 위해 필요한 규칙의 집합

통신규약은 상호 간에 이해할 수 있는 의미 내용을 표현하는 형식,
즉 정보 교환 형식과 정보의 송수신 방법 등을 규정하는 규칙으로 구성



통신규약!

같은 통신 규약을 사용하면 **기종과 모델이 달라도 컴퓨터 간에 통신**할 수 있게 되고, 각각의 컴퓨터상에서 다른 프로그램을 사용하고 있더라도 **컴퓨터 사이에서 데이터의 형식을 일치시켜 프로그램을 동작**시킬 수 있게 된다.

인터넷용으로 개발되어 기본으로 사용되는 통신 규약은  **TCP/IP**이다.



RESOURCE & MIME

RESOURCE


MIME TYPE



RESOURCE

text
html
word
jpeg
avi
pdf
...

문서 파일, 음악 파일,
그림 파일, 파워포인트 파일
카메라 라이브 영상,
주식거래 데이터,
부동산 데이터베이스 데이터,
온라인 쇼핑몰...



RESOURCE

text

html

어떤 종류의 콘텐츠 소스도 리소스가 될 수 있다.

jpeg

avi

pdf

...

문서 파일, 음악 파일,
프로그램 파일, 인포그래픽 파일,
카메라 라이브 영상,
주식거래 데이터,
부동산 데이터베이스 데이터,
온라인 쇼핑몰...



RESOURCE

인터넷은 수천가지 데이터 타입을 다루기 때문에
웹 서버는 전송되는 객체 각각에 신중하게
MIME TYPE 이라는 데이터 포맷 라벨을 붙인다.

부동산 데이터베이스 데이터,
온라인 쇼핑몰...



MIME (다목적 인터넷 메일 확장)

MIME은 이메일에서 잘 동작했기 때문에, HTTP에서도
멀티미디어 콘텐츠를 기술하고 라벨을 붙이기 위해 채택

text/html

text/plain

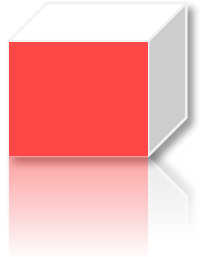
image/jpeg

image/gif

video/quicktime

application/vnd.ms-powerpoint

..



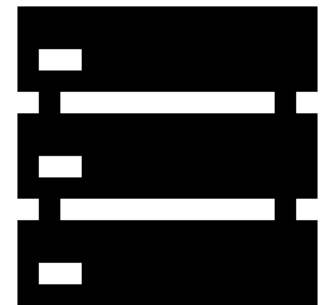
MIME (다목적 인터넷 메일 확장)



클라이언트



서버로부터 리소스를 돌려받을 때, 다룰 수 있는 리소스인지 MIME 타입을 통해 확인



서버

MIME (다목적 인터넷 메일 확장)

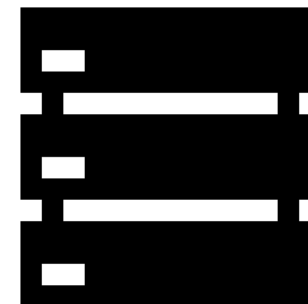


Content-type: image/jpeg
Content-length: 12984

서버로부터 리소스를 돌려받을 때, 다룰 수 있는 리소스인지 MIME 타입을 통해 확인



클라이언트



서버

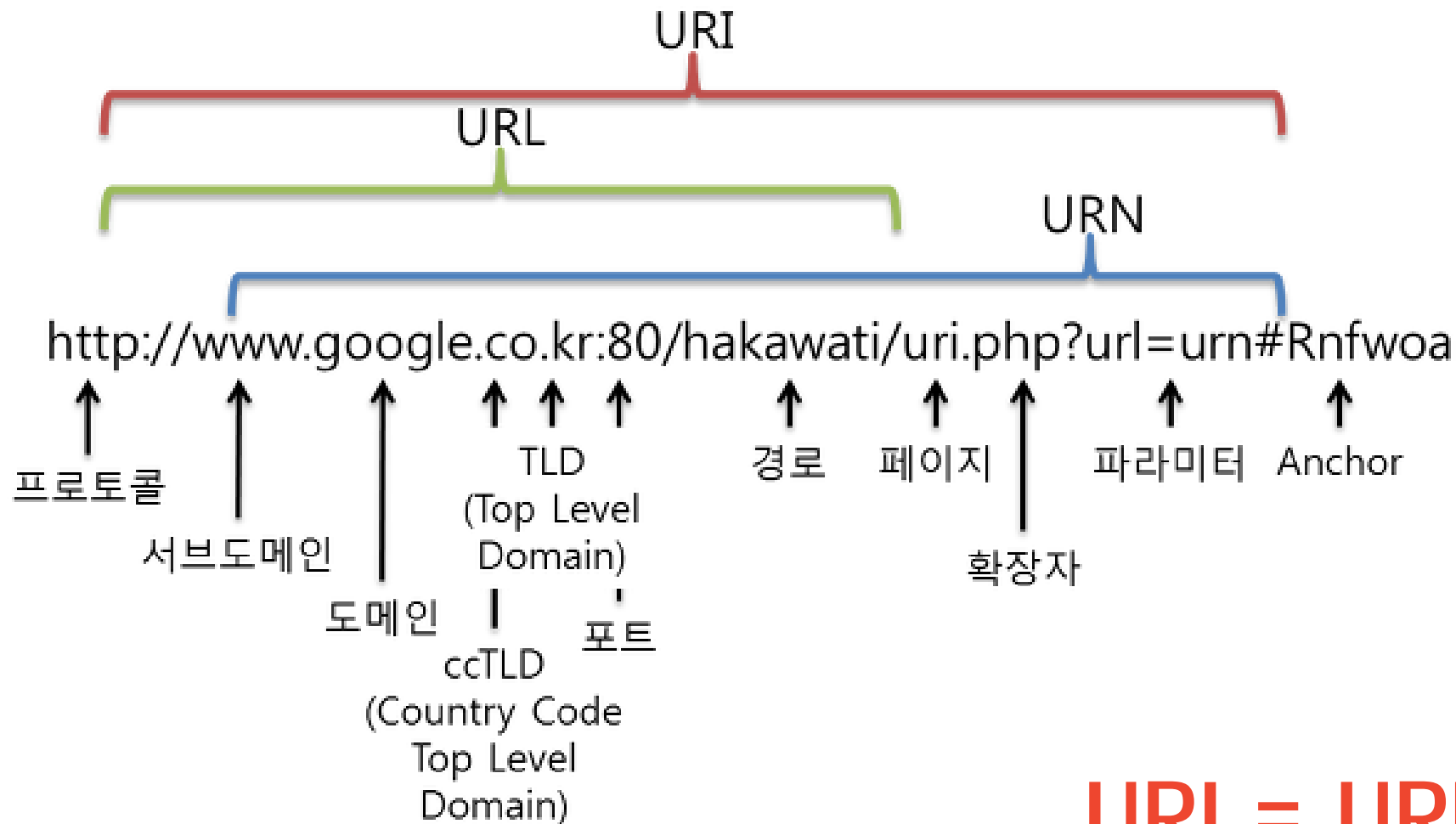


URI

URI

URL & URN

Uniform Resource Identifier :URI



$$\text{URI} = \text{URL} + \text{URN}$$

Uniform Resource Locator :URL

`http: / / www.saramin.co.kr / zf_user / seminar.jpg`

Uniform Resource Locator :URL

`http://www.saramin.co.kr/zf_user/seminar.jpg`



HTTP 프로토콜을 사용하라
(=스킴 Scheme)



Uniform Resource Locator :URL

`http: // www.saramin.co.kr / zf_user / seminar.jpg`

↓
HTTP? HTTPS?

HTTP 프로토콜을 사용하라
(=스킴 Scheme)

Uniform Resource Locator :URL

`http: // www.saramin.co.kr /zf_user /seminar.jpg`



`www.saramin.co.kr` 으로 이동하라

Uniform Resource Locator :URL

`http: / / www.saramin.co.kr / zf_user / seminar.jpg`



`/zf_user/seminar.jpg` 라고 불리는 리소스를 가져와라

Uniform Resource Name :URN

urn : isbn : 9788982814471

URL이 http와 같은 프로토콜로 시작한다면, URN의 경우 urn으로 시작하여, 콜론(:)으로 구분하여 표현

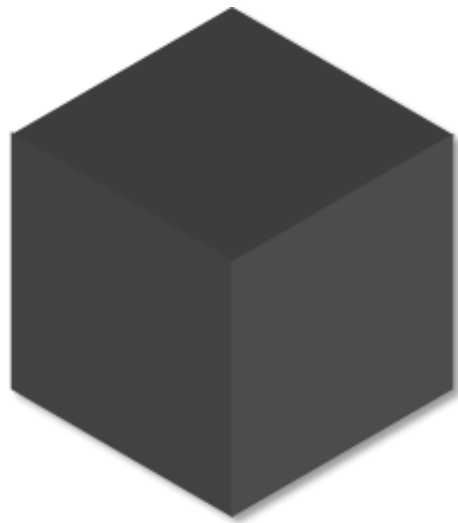
위 예제에서 ISBN 이란 국제표준도서번호를 말하고, 9788982814471 은 연금술사란 도서의 ISBN이다.

즉, 위 URN주소는 연금술사 도서를 가리키는 URN이다.



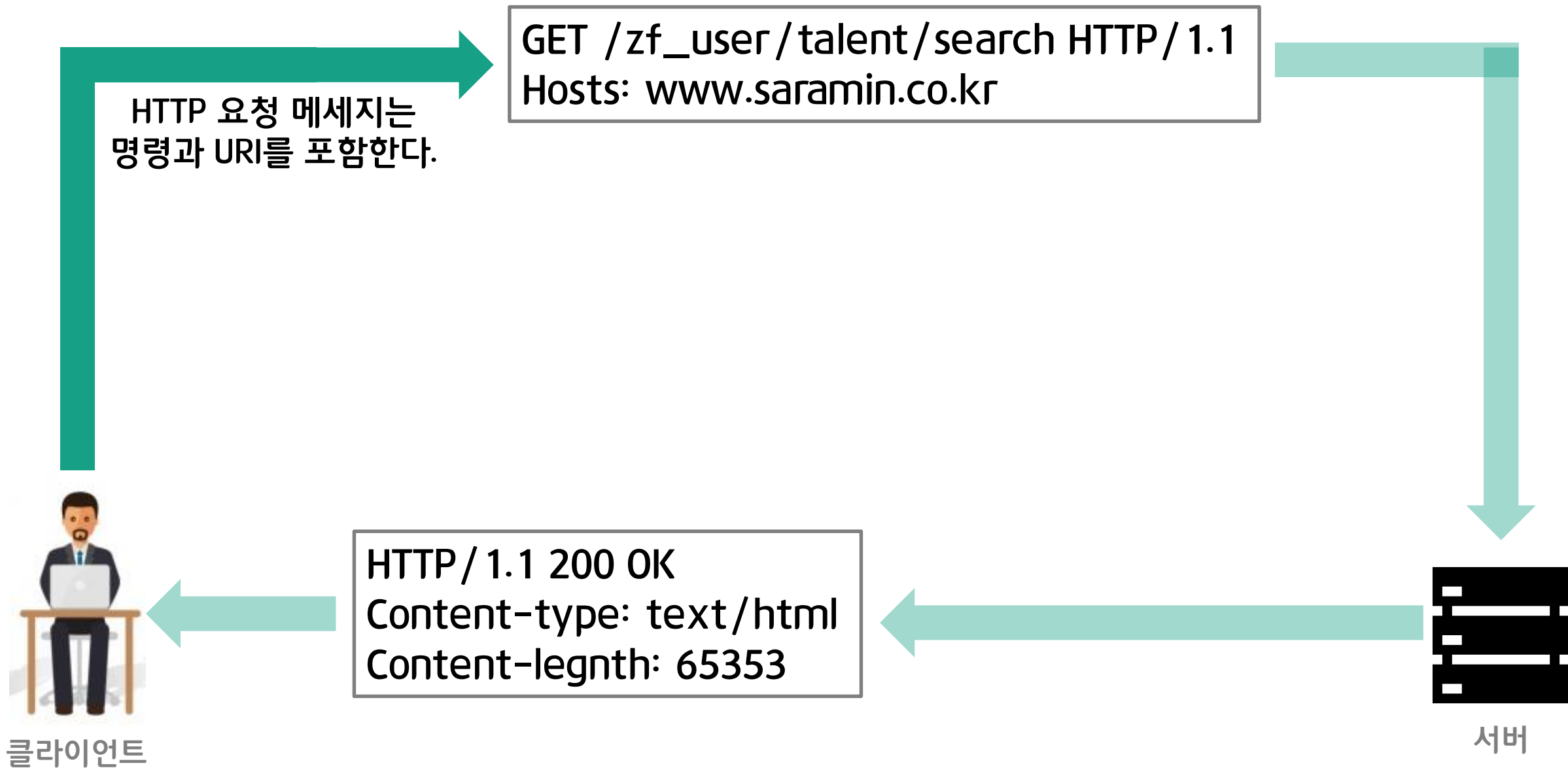
TRANSACTION

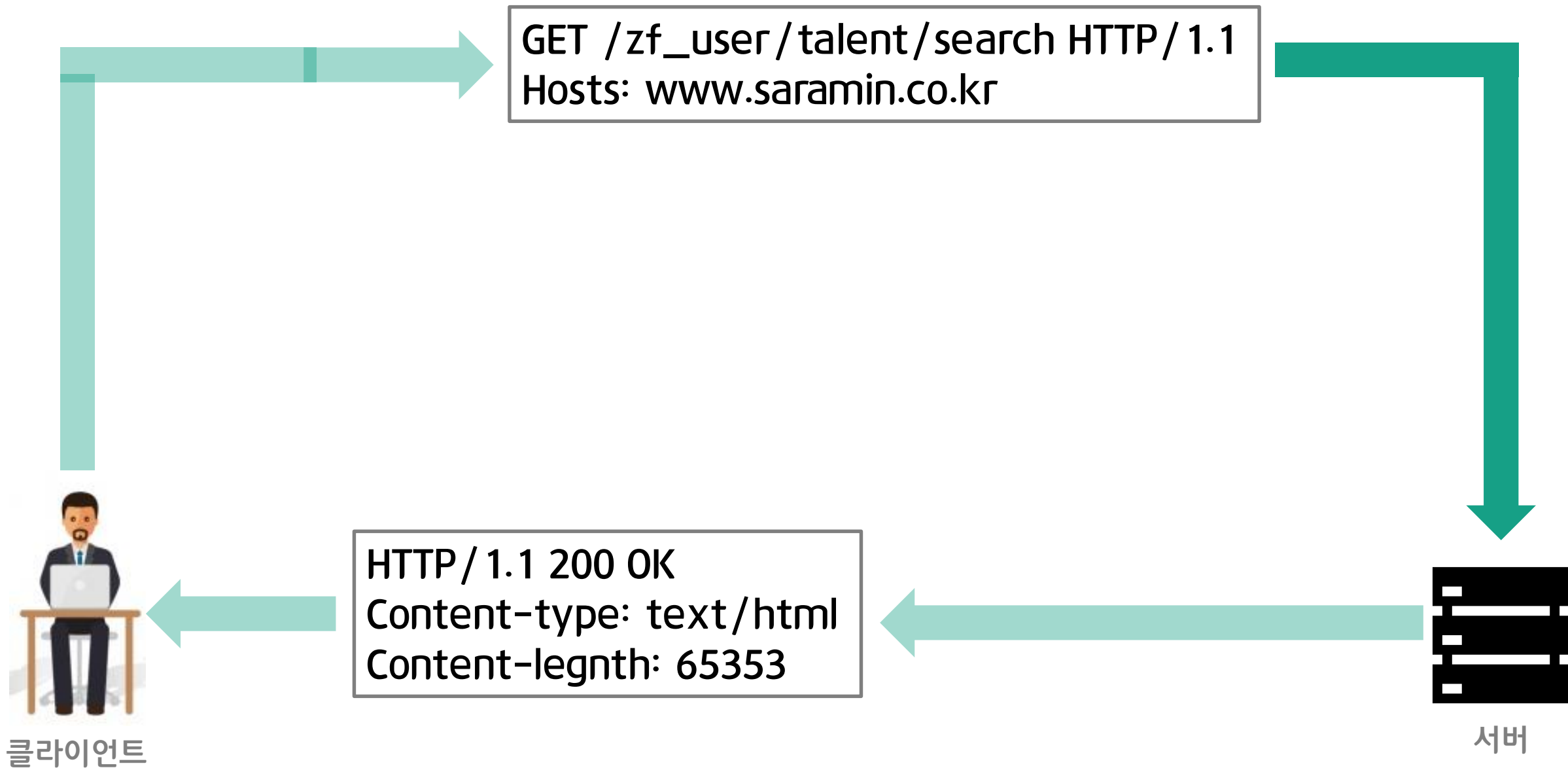
TRANSACTION

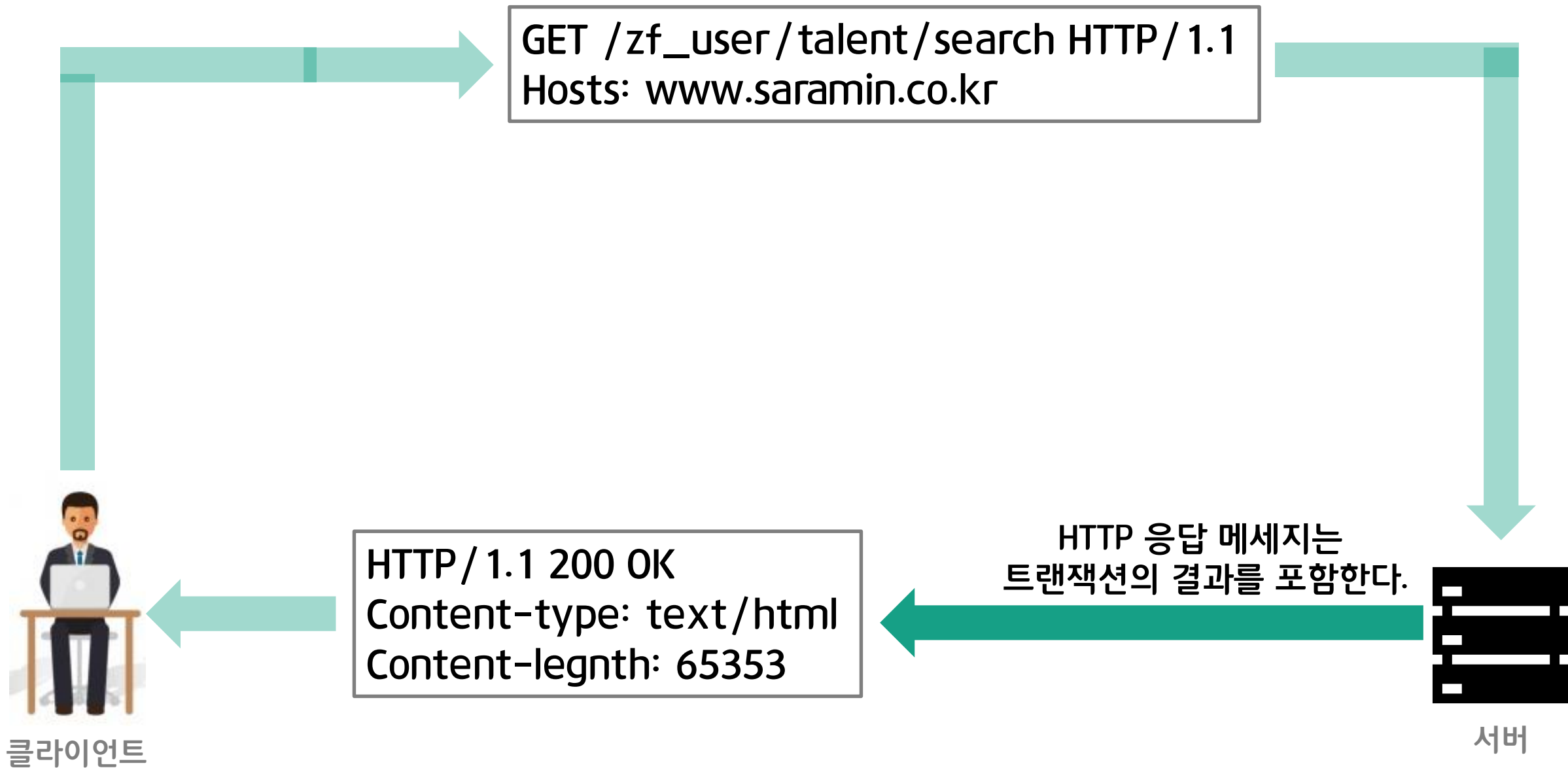


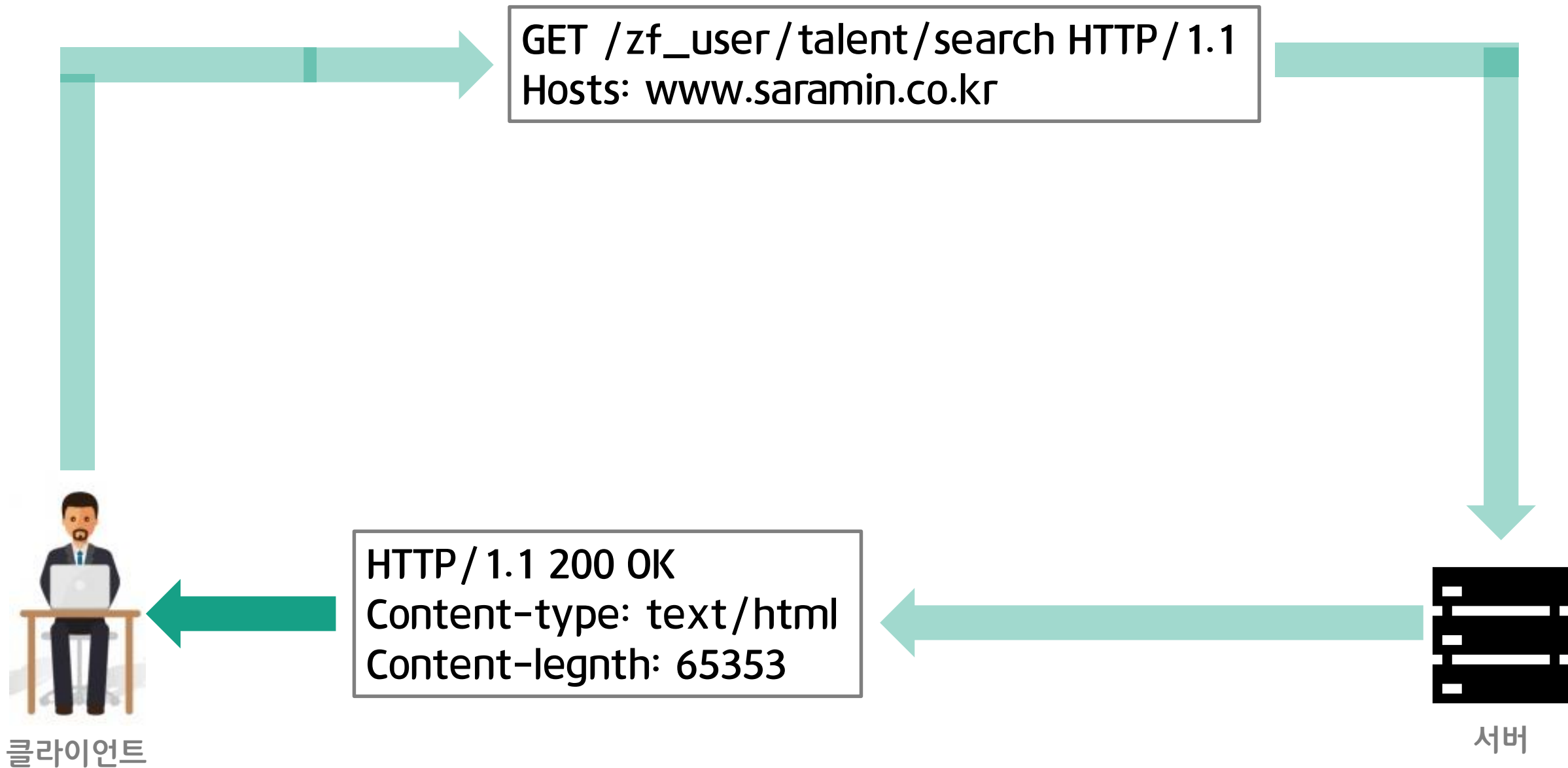
HTTP TRANSACTION

**요청 명령(클라이언트에서 서버로 보내는)과
응답 결과(서버가 클라이언트에게)로 구성되어 있다.**









HTTP METHOD

GET : 서버에서 클라이언트로 지정된 리소스를 보내라는 의미

```
GET /zf_user/talent/search HTTP/1.1
Host: www.saramin.co.kr
```

POST : 클라이언트가 리소스의 위치를 지정하지 않았을 때 리소스를 생성하기 위해 사용하는 메서드 (서버가 위치를 지정)

ex) 공고 테이블 > 공고 생성 > 마지막 공고의 번호 + 1 로 생성

PUT : POST와 유사한 전송 구조를 가지고 있지만, 리소스 위치를 명확히 지정하고 요청을 보내는 메서드 (사용자가 위치를 지정)

ex) 2712번 공고를 수정 또는 생성하도록

DELETE : 지정된 리소스를 서버에서 삭제하라

```
HTTP/1.1 200 OK
Content-type: image/gif
Content-length: 15352
```

HEAD : GET과 유사한 방식이나 웹 서버에서 헤더 정보 이외에는 어떤 데이터도 보내지 않는다. 웹 서버의 다운 여부 점검이나 웹 서버 정보등을 얻기 위해 사용



클라이언트



서버

HTTP STATE CODE



```
GET /zf_user / talent / search HTTP / 1.1
Hosts: www.saramin.co.kr
```

1xx : 처리중... (조건부 응답)

2xx : 성공

3xx : 리다이렉션

4xx : 클라이언트 오류

5xx : 서버 오류

```
HTTP / 1.1 200 OK
Content-type: image / gif
Content-legnth: 65353
```



클라이언트



서버

HTTP STATE CODE



```
graph LR; Client[클라이언트] -- "GET /zf_user / talent / search HTTP / 1.1  
Hosts: www.saramin.co.kr" --> Server[서버]; Server -- "HTTP / 1.1 200 OK  
Content-type: image / gif  
Content-legnth: 65353" --> Client;
```

GET /zf_user / talent / search HTTP / 1.1
Hosts: www.saramin.co.kr

100 (계속): 요청자는 요청을 계속해야 한다. 서버는 이 코드를 제공하여 요청의 첫 번째 부분을 받았으며 나머지를 기다리고 있음을 나타낸다.

101 (프로토콜 전환): 요청자가 서버에 프로토콜 전환을 요청했으며 서버는 이를 승인하는 중이다.

102 (처리)

HTTP / 1.1 200 OK
Content-type: image / gif
Content-legnth: 65353



클라이언트



서버

HTTP STATE CODE



```
graph LR; Client[클라이언트] -- "GET /zf_user / talent / search HTTP / 1.1  
Hosts: www.saramin.co.kr" --> Server[서버]; Server -- "HTTP / 1.1 200 OK  
Content-type: image / gif  
Content-legnth: 65353" --> Client;
```

GET /zf_user / talent / search HTTP / 1.1
Hosts: www.saramin.co.kr

200 (성공): 서버가 요청을 제대로 처리했다는 뜻이다. 이는 주로 서버가 요청한 페이지를 제공했다는 의미로 쓰인다.

202 (허용됨): 서버가 요청을 접수했지만 아직 처리하지 않았다.

204 (콘텐츠 없음): 서버가 요청을 성공적으로 처리했지만 콘텐츠를 제공하지 않는다.



클라이언트

HTTP / 1.1 200 OK
Content-type: image / gif
Content-legnth: 65353



서버

HTTP STATE CODE

```
GET /zf_user/talent/search HTTP/1.1
Host: www.saramin.co.kr
```

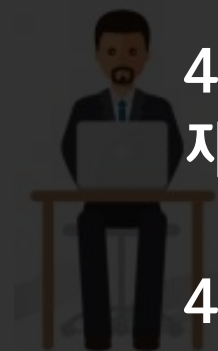
400 (잘못된 요청): 서버가 요청의 구문을 인식하지 못했다.

401 (권한 없음): 이 요청은 인증이 필요하다. 서버는 로그인에 필요한 페이지에 대해 이 요청을 제공할 수 있다.

403 (금지됨): 서버가 요청을 거부하고 있다. 예를 들자면, 사용자가 리소스에 대한 필요 권한을 갖고 있지 않다.

404 (찾을 수 없음): 서버가 요청한 페이지를 찾을 수 없다. 예를 들어 서버에 존재하지 않는 페이지에 대한 요청이 있을 경우 서버는 이 코드를 제공한다.

408 (요청 시간초과): 서버의 요청 대기 시간이 시간을 초과하였다.



클라이언트



서버

HTTP STATE CODE

GET /zf_user / talent / search HTTP / 1.1
Hosts: www.saramin.co.kr

500 (내부 서버 오류): 서버에 오류가 발생하여 요청을 수행할 수 없다.

505 (HTTP 버전이 지원되지 않음): 서버가 요청에 사용된 HTTP 프로토콜 버전을 지원하지 않는다.

598 (네트워크 읽기 시간초과 오류, 알 수 없음)

599 (네트워크 연결 시간초과 오류, 알 수 없음)


HTTP / 1.1 200 OK
Content-type: image/gif
Content-length: 65353



클라이언트



서버



MESSAGE & TCP/IP

MESSAGE

TCP/IP



MESSAGE

요청 메시지

응답 메시지

웹 클라이언트에서 웹 서버로 보낸 HTTP 메시지를 **요청 메시지**라고 부른다.
서버에서 클라이언트로 가는 메시지는 **응답 메시지**라고 부른다.



MESSAGE


요청 메시지

```
GET /zf_user/talent/search HTTP/1.1  
User-agent: Mozilla/4.75 [en] (WinXp; U)  
Host: www.saramin.co.kr  
Accept: text/html, image/gif, image/jpg  
Accept-Language: en
```





MESSAGE



```
HTTP/1.1 200 OK
Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.1 Bsafe-SSL (Unix)
Last-modified: Tue, 04 Jul 2000 09:46:21
Content-length: 403
Content-type: text/html

<HTML>
<HEAD><TITLE>Hello World!</TITLE></HEAD>

<BODY>
<H1>Tools Page</H1>
...
```



응답 메시지



MESSAGE

요청 메시지

응답 메시지

시작줄

GET /zf_user/talent/search HTTP/1.1

HTTP/1.1 200 OK

헤더

User-agent: Mozilla/4.75 [en] (WinXp; U)
Host: www.saramin.co.kr
Accept: text/html, image/gif, image/jpg
Accept-Language: en

Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.1 Bsafe-SSL (Unix)
Last-modified: Tue, 04 Jul 2000 09:46:21
Content-length: 403
Content-type: text/html

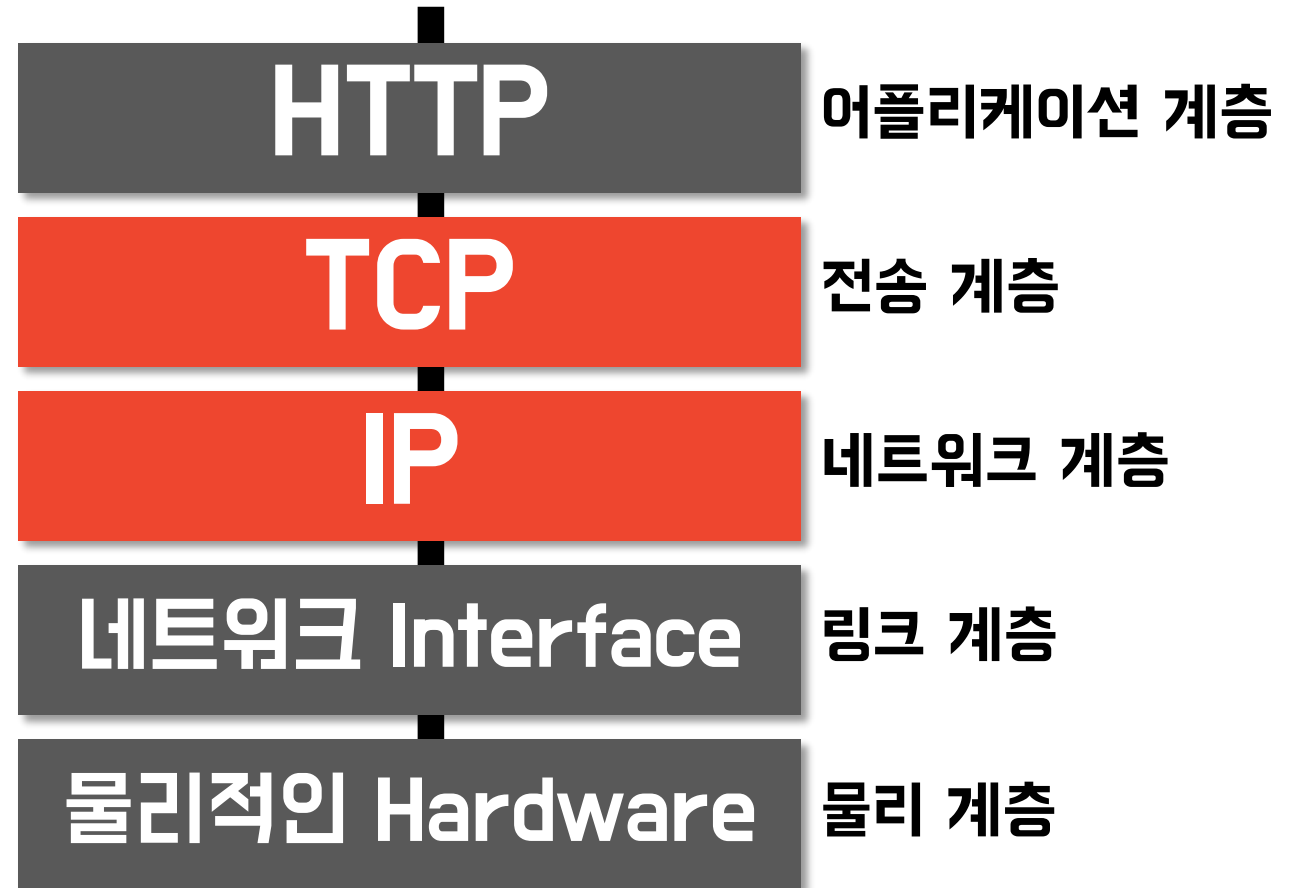
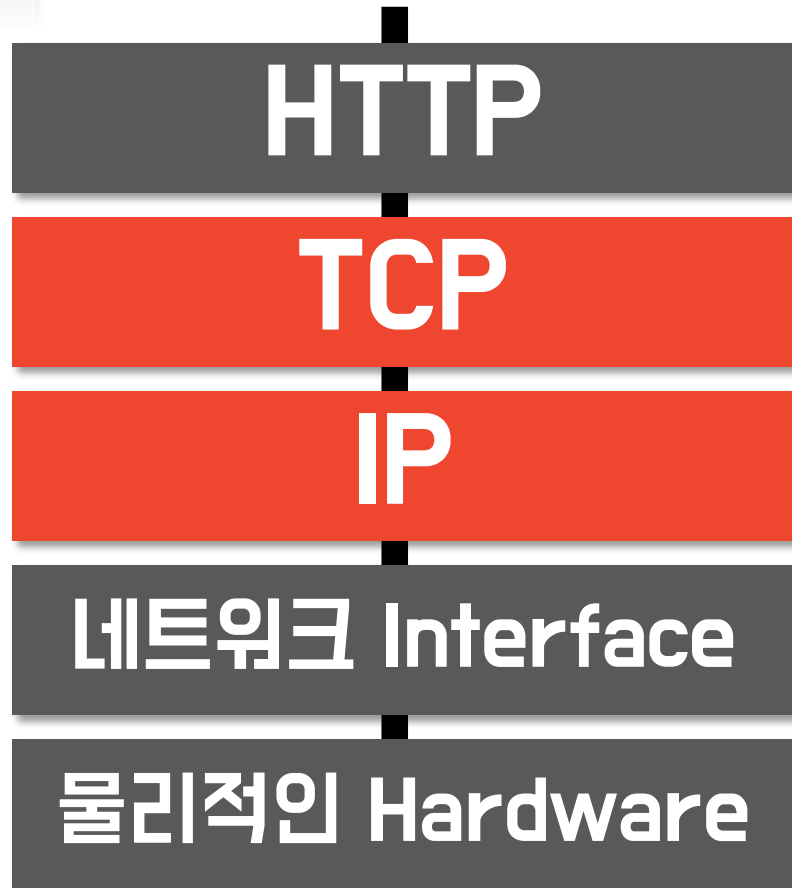
본문

<HTML>
<HEAD><TITLE>Hello World!</TITLE></HEAD>

<BODY>
<H1>Tools Page</H1>
...



TCP/IP



1. www.naver.com 입력

TCP/IP

HTTP

TCP

IP

네트워크 Interface

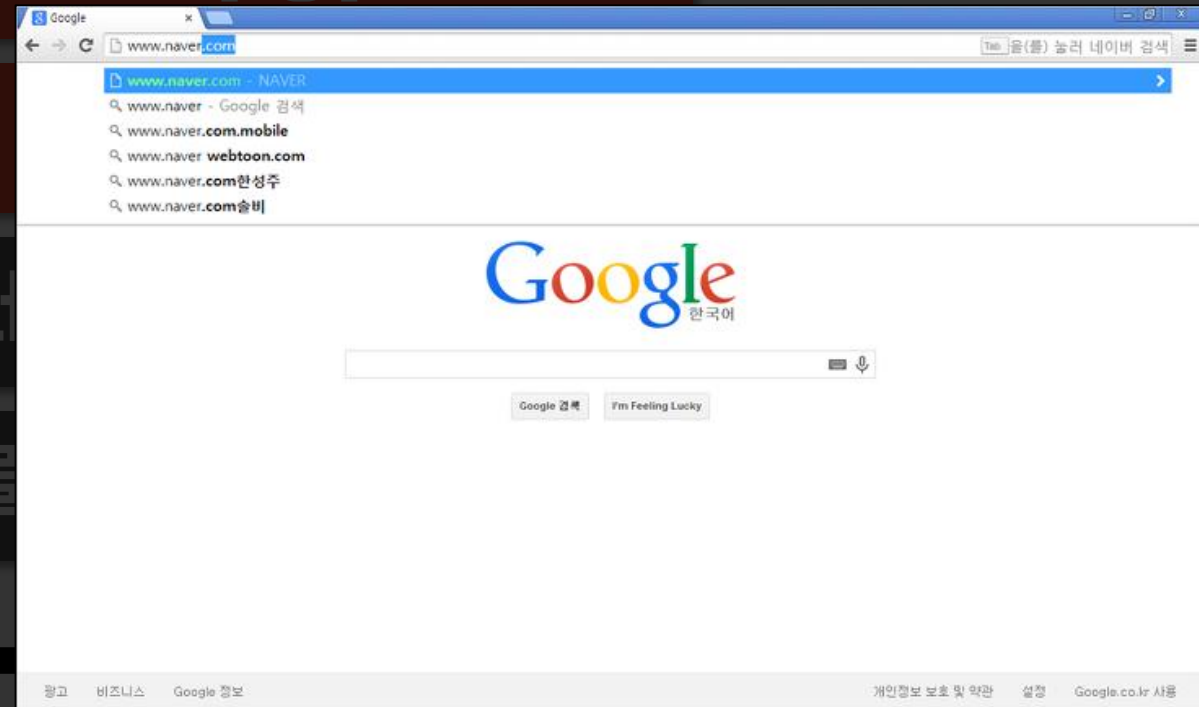
물리적인 Hardware

HTTP

TCP

어플리케이션 계층

전송 계층



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

TCP Header																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
0	Source port																Destination port																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
32																	Sequence number																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
64																	Acknowledgment number																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
96	Data offset				Reserved				C	E	U	A	P	R	S	F	W																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												</

HTTP

TCP

IP

네트워크 Interface

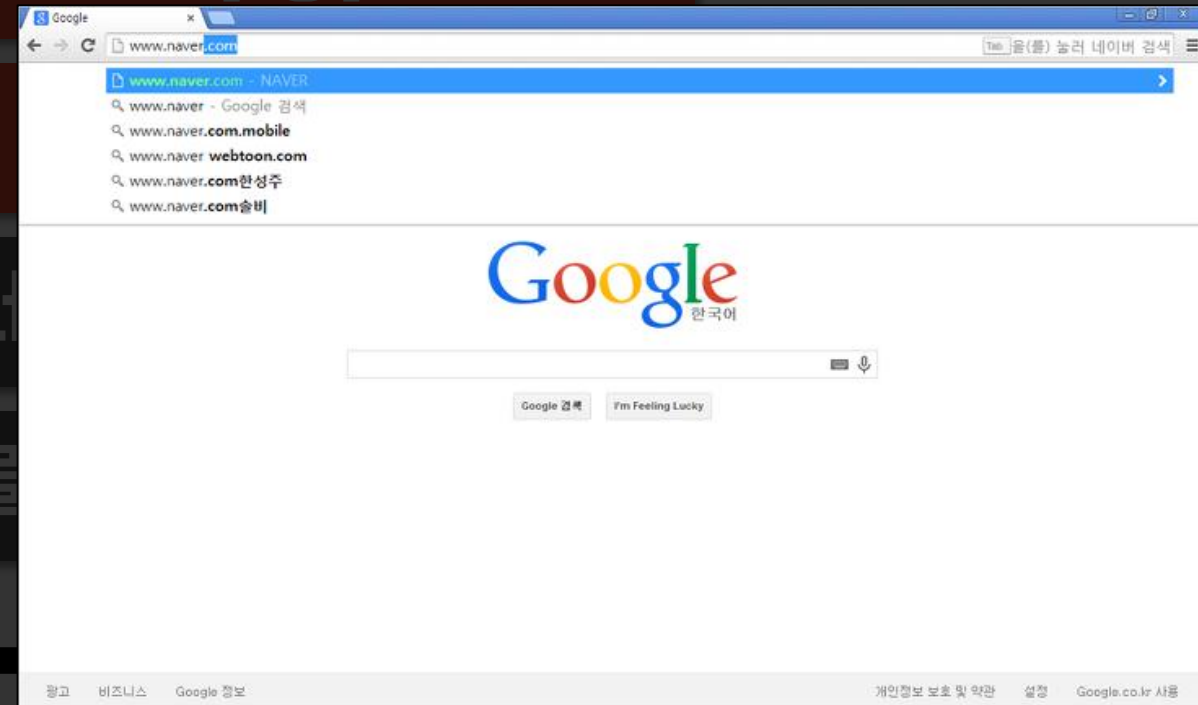
물리적인 Hardware

HTTP

TCP

어플리케이션 계층

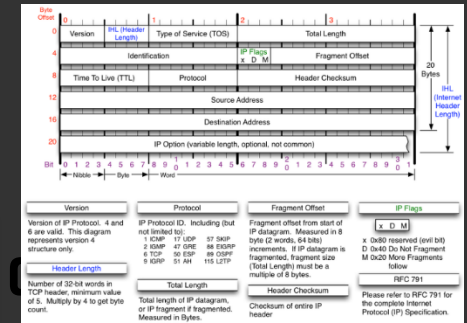
전송 계층



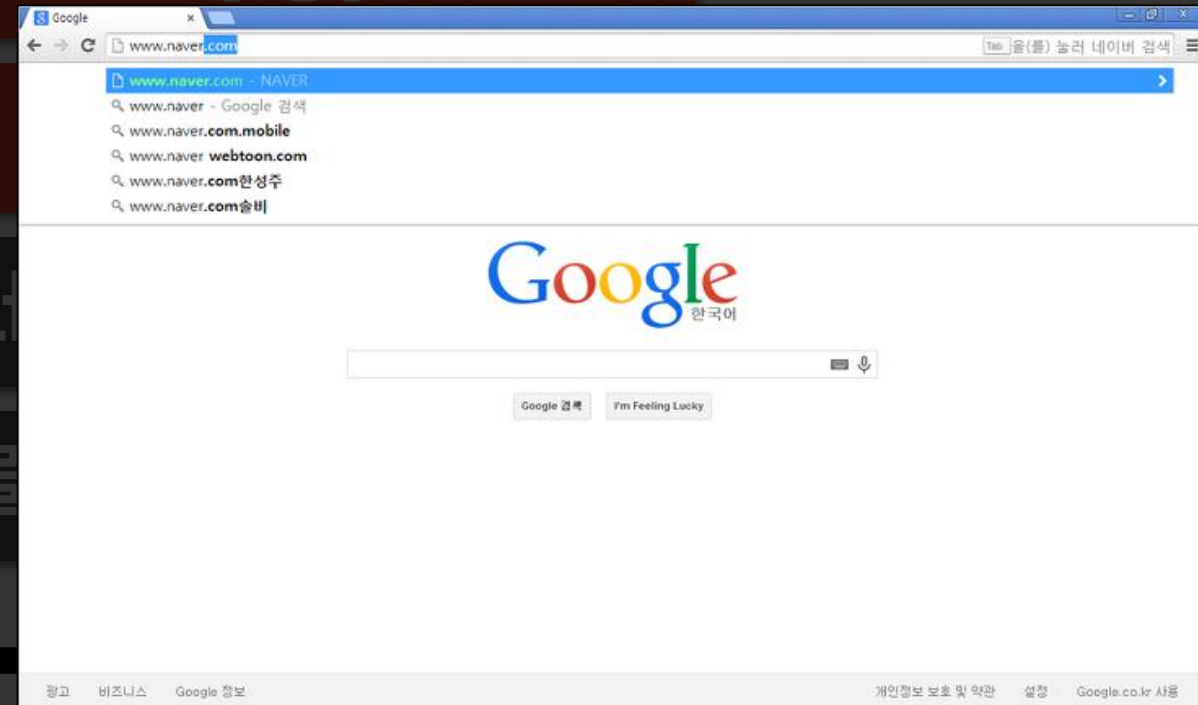
1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)



전송 계층



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)

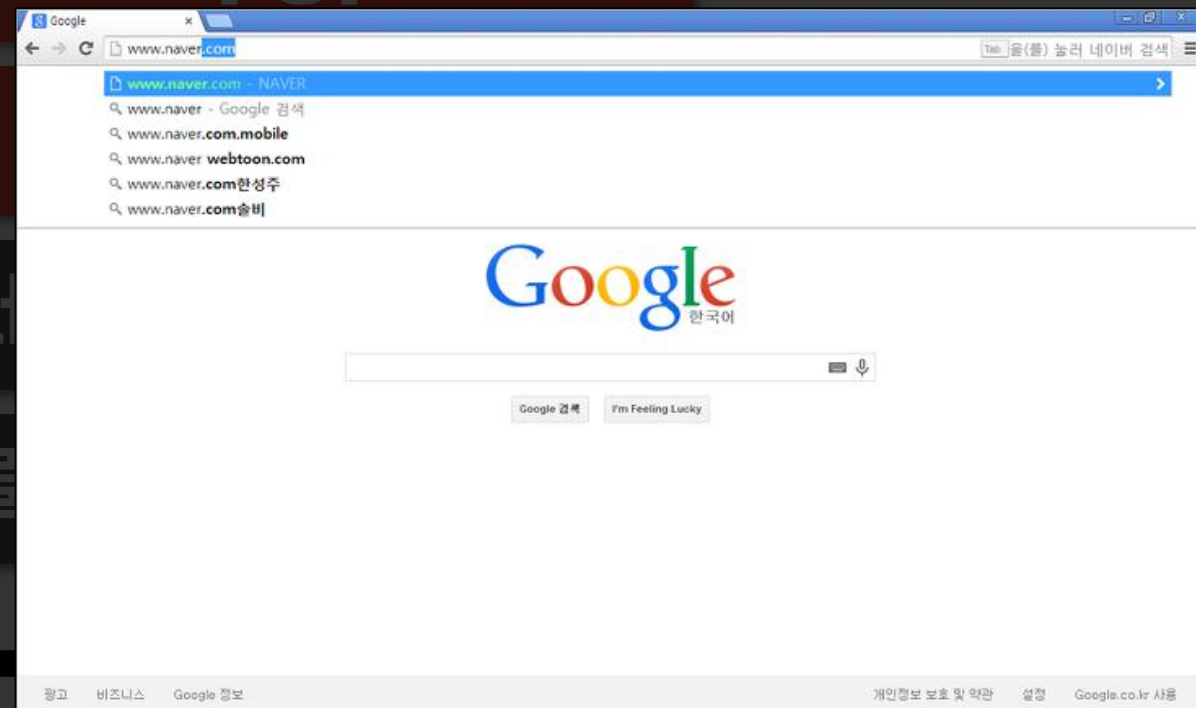
4. 이것을 Ethernet 카드로 보내 Internet 으로 나가게 된다.

어플리케이션 계층

전송 계층

네트워크 Interface

물리적인 Hardware



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)

4. 이것을 Ethernet 카드로 보내 Internet 으로 나가게 된다.

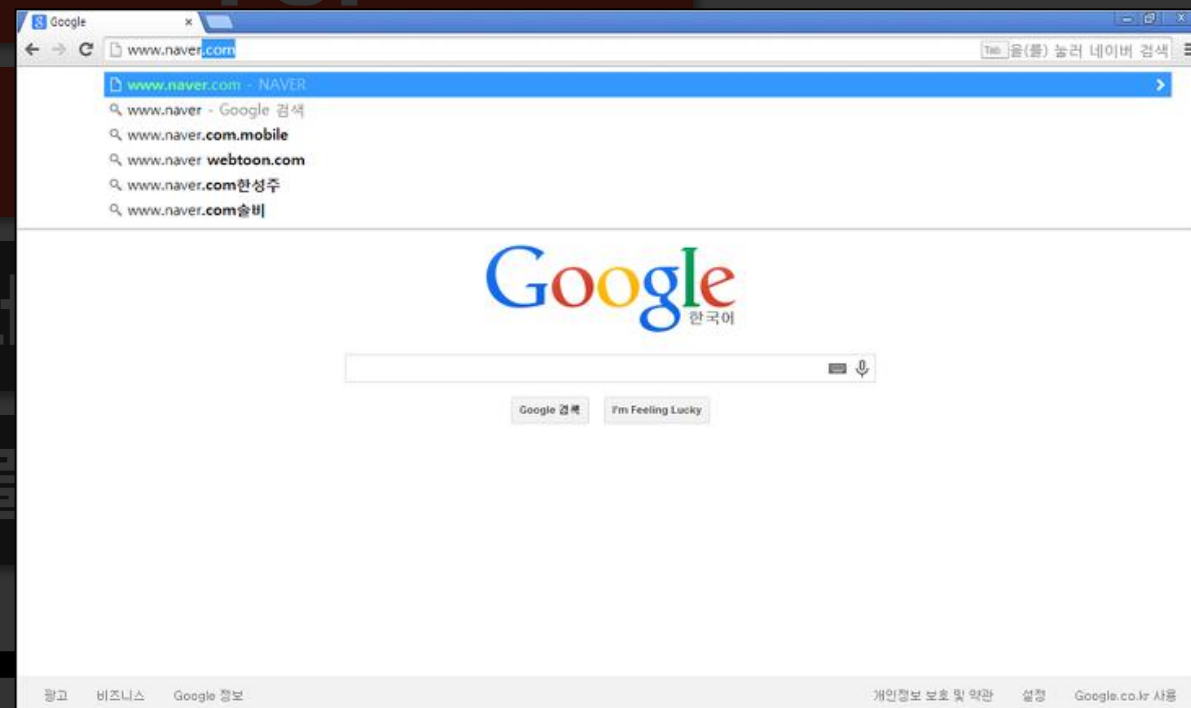
5. 네이버의 Ethernet 카드로 TCP/IP 패킷을 전달

어플리케이션 계층

전송 계층

네트워크 Interface

물리적인 Hardware



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)

4. 이것을 Ethernet 카드로 보내 Internet 으로 나가게 된다.

5. 네이버의 Ethernet 카드로 TCP/IP 패킷을 전달

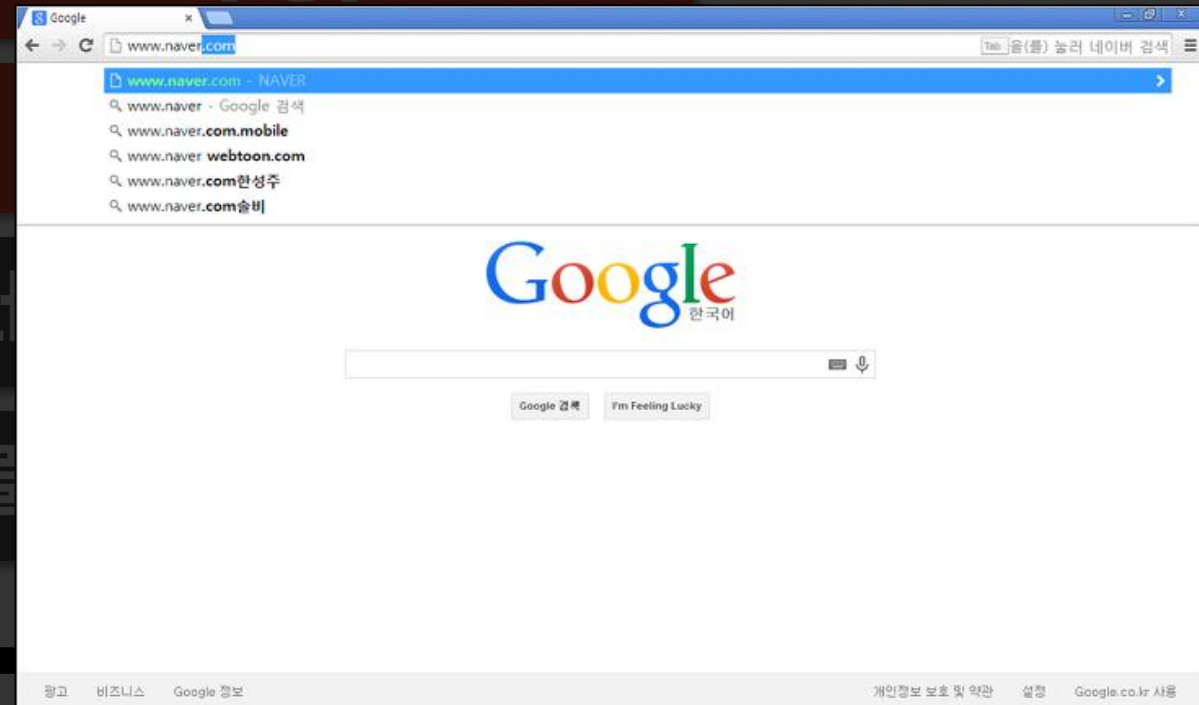
6. IP 패킷을 분석해서 어디서 왔는지, 목적지가 어딘지 분석

어플리케이션 계층

전송 계층

네트워크 Interface

물리적인 Hardware



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)

4. 이것을 Ethernet 카드로 보내 Internet 으로 나가게 된다.

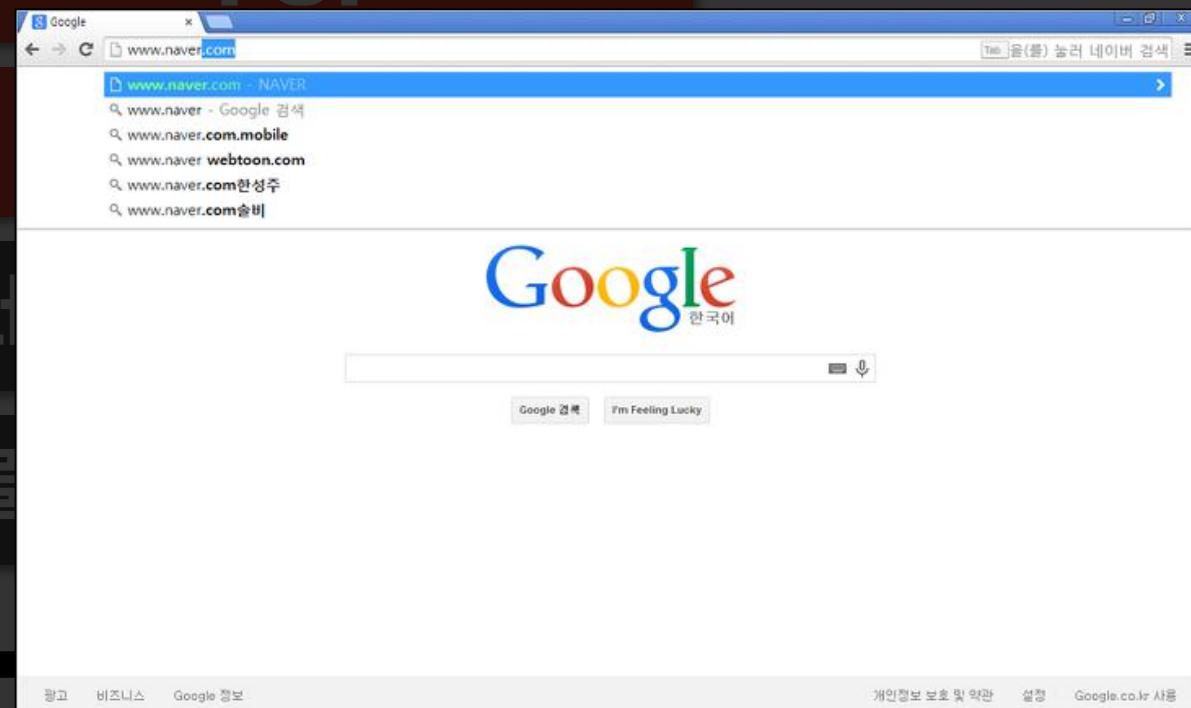
5. 네이버의 Ethernet 카드로 TCP/IP 패킷을 전달

6. IP 패킷을 분석해서 어디서 왔는지, 목적지가 어딘지 분석

7. TCP 계층에서 요청받은 메시지에 대한 검증

어플리케이션 계층

전송 계층



1. www.naver.com 입력

2. 사용자의 요청을 TCP 패킷으로 만들어줌

3. IP 패킷으로 다시 만들어줌 (자신의 주소와 서버의 주소가 담겨있다.)

4. 이것을 Ethernet 카드로 보내 Internet 으로 나가게 된다.

5. 네이버의 Ethernet 카드로 TCP/IP 패킷을 전달

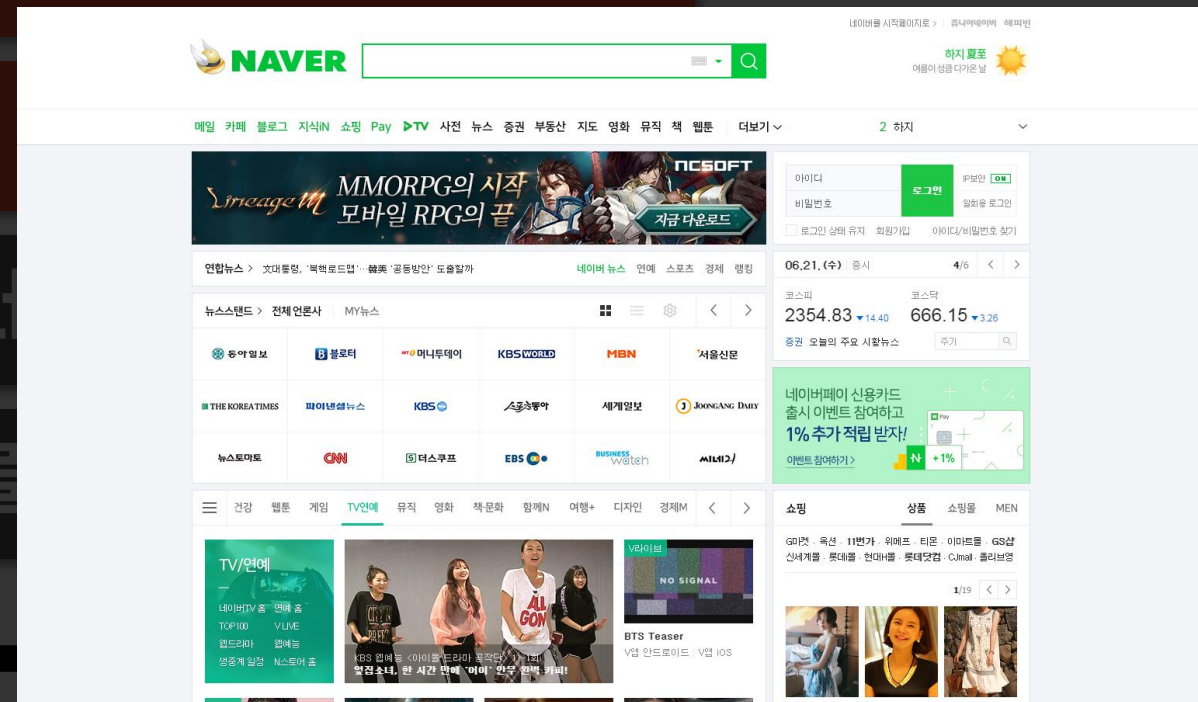
6. IP 패킷을 분석해서 어디서 왔는지, 목적지가 어딘지 분석

7. TCP 계층에서 요청받은 메시지에 대한 검증

8. 브라우저는 HTTP 프로토콜에 준하여 렌더링 작업

어플리케이션 계층

전송 계층





TCP/IP

TCP

전송 계층

IP

네트워크 계층

- 오류 없는 데이터를 전송
- 순서에 맞는 전달 (데이터는 언제나 보낸 순서대로 도착한다)
- 조각나지 않는 데이터 스트림 (언제든 어떤 크기로든 보낼 수 있다)



TCP/IP



TCP/IP 커넥션

전화번호를 누르면 연결이 되고,
연결된 동안에는 대화를 나눌 수 있으며,
대화가 끝나서 전화를 끊으면 연결이 끊기는 것!



TCP/IP

`http://www.saramin.co.kr`

- 1. 웹브라우저는 서버의 URL에서 호스트명을 추출한다.**



TCP/IP

<http://www.saramin.co.kr>

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.



TCP/IP

`http://182.135.75.25`

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.
2. 웹브라우저는 서버의 호스트명을 **IP**로 변환한다.



TCP/IP

`http://182.135.75.25(:80)`

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.
2. 웹브라우저는 서버의 호스트명을 **IP**로 변환한다.
3. 웹브라우저는 URL에서 **포트번호(있다면)**를 추출한다



TCP/IP

`http://182.135.75.25(:80)`

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.
2. 웹브라우저는 서버의 호스트명을 **IP**로 변환한다.
3. 웹브라우저는 URL에서 **포트번호(있다면)**를 추출한다
4. 웹브라우저는 웹 서버와 **TCP 커넥션**을 맺는다.





TCP/IP

`http://182.135.75.25(:80)`

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.
2. 웹브라우저는 서버의 호스트명을 **IP**로 변환한다.
3. 웹브라우저는 URL에서 **포트번호(있다면)**를 추출한다
4. 웹브라우저는 웹 서버와 **TCP 커넥션**을 맺는다.
5. 웹브라우저는 서버에 **HTTP 요청**을 보낸다.
6. 서버는 웹브라우저에 **HTTP 응답**을 돌려준다.



TCP/IP

`http://182.135.75.25(:80)`

1. 웹브라우저는 서버의 URL에서 **호스트명**을 추출한다.
2. 웹브라우저는 서버의 호스트명을 **IP**로 변환한다.
3. 웹브라우저는 URL에서 **포트번호(있다면)**를 추출한다
4. 웹브라우저는 웹 서버와 **TCP 커넥션**을 맺는다.
5. 웹브라우저는 서버에 **HTTP 요청**을 보낸다.
6. 서버는 웹브라우저에 **HTTP 응답**을 돌려준다.
7. 커넥션이 닫히면, 웹브라우저는 문서를 보여준다.



감사합니다.

서비스개발 2팀 김기진
92.kr.dev@Saramin.co.kr

