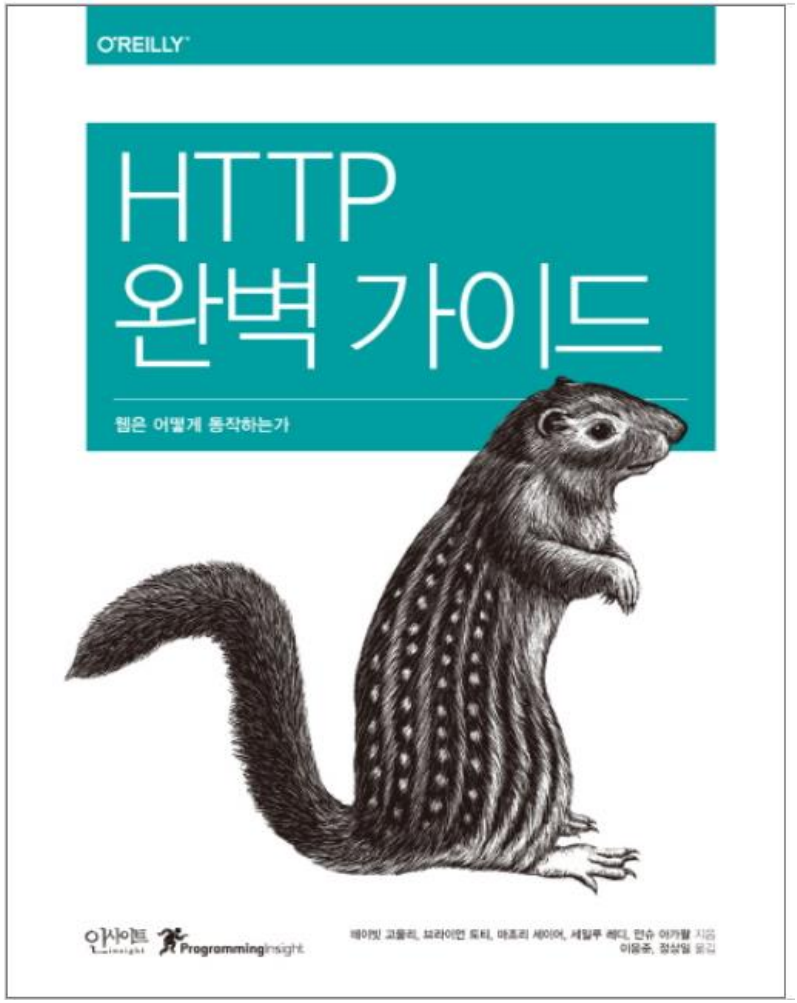
Cache

주 Reference


O'REILLY
HTTP
완벽 가이드
웹은 어떻게 동작하는가
Programming Insight


IT COOKBOOK
238
쉽게
배우는
운영체제
Introduction to
Operating Systems
조성호 지음
한빛아카데미

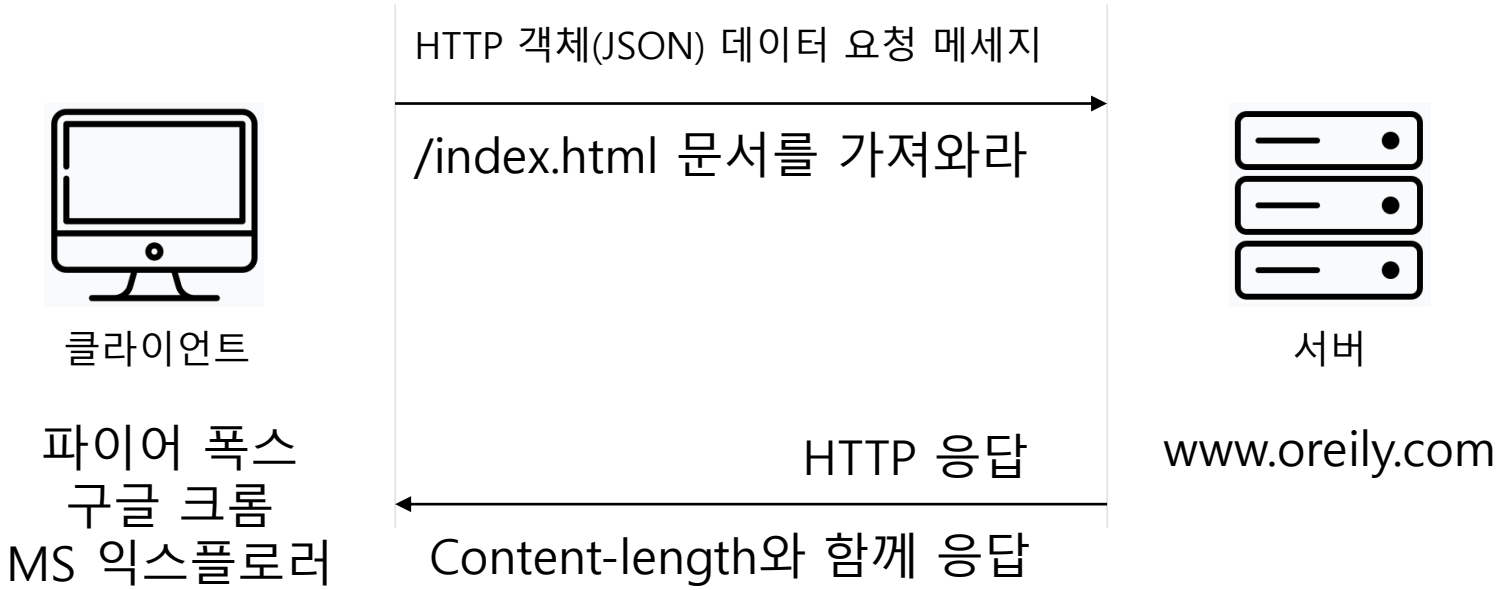
HTTP 개관

HTTP란?
- 브라우저와 서버가 통신하기 위한 프로토콜
- 팀 버너스 리가 고안 (WWW, URL, HTML도 마찬가지)
- 1991년 최초로 문서화 HTTP/0.9
- 1996년 HTTP/1.0
- 1999년 HTTP/1.1
- 2015년 HTTP/2

- HTTP 버전을 매기는 방식이 존재
- HTTP의 버전 형식은 HTTP/<메이저>.<마이너>, 메이저, 마이너는 모두 정수

HTTP 개관

웹 서버와 웹 클라이언트 통신
주소창에 http://www.oreily.com/index.html 입력



HTTP 개관

리소스
- 웹 서버는 웹 리소스(렌더링에 필요한 모든 데이터) 관리 및 제공
- 가장 간단한 웹 리소스는 웹 서버 파일 시스템에 있는 정적 파일(HTML, CSS, JAVASCRIPT, 폰트, 이미지)

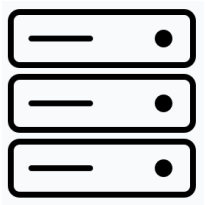
미디어 타입(MIME)
- 웹 서버는 모든 HTTP 객체 데이터에 MIME(Multipurpose Internet Mail Extension) 타입을 붙임
- 웹 브라우저는 MIME 타입을 통해서 다룰 수 있는 객체인지 확인

클라이언트

파이어 폭스
구글 크롬
MS 익스플로러

HTTP 객체 데이터 요청 메세지

/index.html 문서를 가져와라



서버

www.oreily.com

HTTP 응답

Content-type: image/html 라벨(이외에도 image/jpeg, image/gif등)
Content-length: 12984
MIME: image/html
Primary Object Type: image, **Specific subtype**: jpeg
Header: Content-type, Content-length

HTTP 개관

URI(Uniform Resource Identifier)
웹 서버의 웹 리소스는 이름(URI)을 갖는다. → 클라이언트가 지목 가능
이후 HTTP에 의해서 URI가 해석

해석
1. http 프로토콜을 이용하라
2. www.조하드웨어.com으로 이동하라
3. speicals/saw-blade.gif 리소스를 가져와라

HTTP 개관

URI는 URL과 URN으로 구성

(1) URL(Uniform Resource **Locator**)
- 리소스에 대한 구체적인 **위치**를 서술

예)
http://www.oreily.com/index.html
오라일리 출판사 홈페이지의 URL

http://www.yahoo.com/images/logo.gif
야후! 웹 사이트 로고의 URL

http://www.joes-hardware.com/inventory-check.cgi?item=12731
물품 #12731의 재고가 있는지 확인하는 프로그램에 대한 URL

HTTP 개관

1. scheme으로, 리소스에 접근하기 위해 사용하는 프로토콜
2. 서버의 주소(www.joes-hardware.com)
3. 웹 서버의 리소스(예: /specials/saw-blade.gif)

1과 2로 구성된 요청, 즉 http://www.조하드웨어.com/을 **루트 요청**이라 함
이 경우 암묵적으로 http://www.조하드웨어.com/index.html을 응답

URL 구성에 대해서 더 알아보기

HTTP 개관

(2) URN(Uniform Resource **Name**)
URL은 리소스에 대한 구체적인 **위치**를 서술

URN은 리소스 위치에 영향 받지 않는 **유일무이한 이름** 서술
리소스가 이름을 바꾸지 않는 한, 여러 종류의 네트워크 접속 프로토콜로 접근해도 문제 없음

URN과 URL 구분
URN은 위치(주소)나 접근법에 대한 명시 없이, 리소스에 대해 이야기할 때 사용
예를 들면, ISBN 시스템에서 0-486-27557-4는 셰익스피어의 작품 로미오와 줄리엣의 특정 에디션을 지칭
이를 URN으로 나타내면, urn:isbn:0-486-27557-4로 표기
이 표기법은 **리소스에 어떻게 접근할 것인지를 명시하지 않으며**, **리소스 자체를 특정하는 것**을 목표

※ ISBN(International Standard Book Number)
전 세계적으로 쏟아져 나오는 방대한 양의 서적을 체계적으로 분류하기 위해 국제적으로 정한 **도서표준 고유코드 번호**. 세계 어디서나 통용될 수 있으며, 번호만으로 어느 나라 어느 출판사에서 나온 책인지 알 수 있음

HTTP 개관

HTTP 트랜잭션
- 요청 명령과 응답 결과로 구성
- 이 과정에서 정형화된 데이터인 **HTTP 메시지**를 이용

메서드
- 서버에게 어떤 동작을 취해야 하는지 알려줌
- 모든 HTTP 요청 메시지는 한 개의 메서드를 가짐

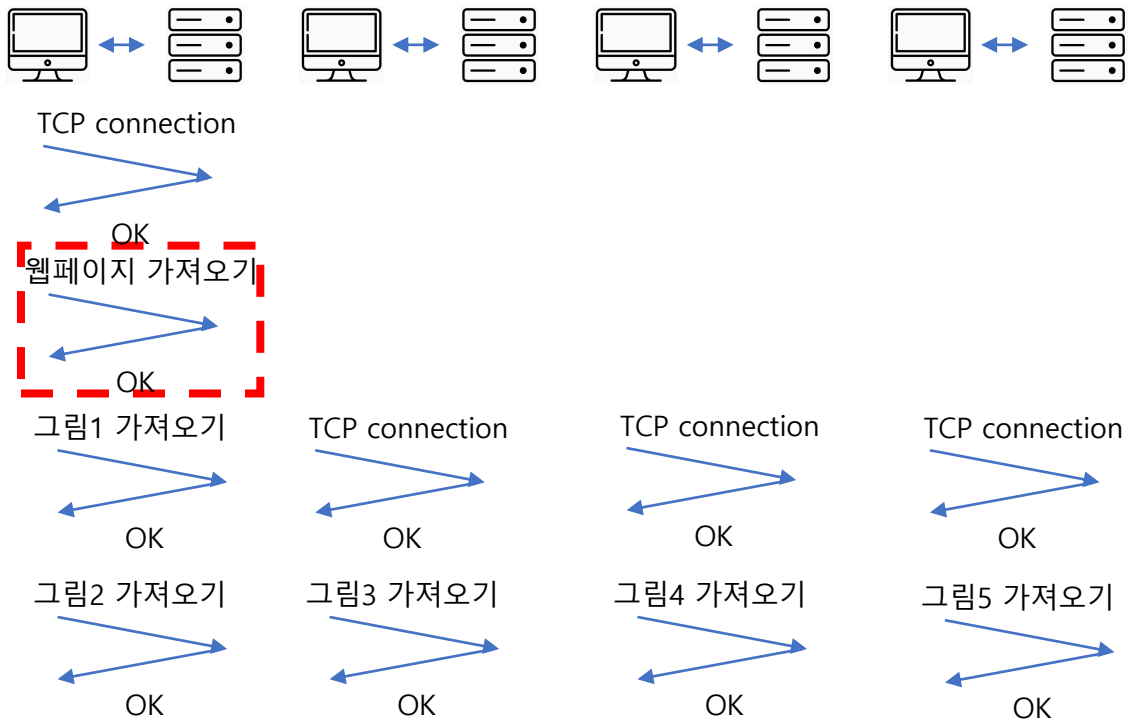
(1) GET: 서버에서 클라이언트로 지정한 리소스를 보내라
(2) PUT: 클라이어트에서 서버로 보낸 데이터를 지정한 이름의 리소스로 저장하라
(3) DELETE: 지정한 리소스를 서버에서 삭제해라
(4) POST: 클라이언트 데이터를 서버 게이트웨이 애플리케이션으로 보내라
(5) HEAD: 지정한 리소스에 대한 응답에서 HTTP 헤더 부분만 보내라

상태코드
- 모든 HTTP 응답 메시지는 상태 코드와 함께 반환
- 요청이 성공했는지, 조치가 필요한지 등을 알려주는 세 자리 숫자
예) 200(문서가 바르게 반환), 302(다른 곳에 가서 리소스를 가져가라, Redirection), 404(리소스 찾을 수 없음)
※ 302 코드는 리소스의 위치가 바뀌어 사용자를 새로운 URL로 이동시킴
- 사유구절(reason phrase)과 함께 반환.
예) 200 OK, 200 Success, 200 All's cool, dude

HTTP 개관

HTTP는 여러 객체로 이루어질 수 있다
- 브라우저는 시각적으로 풍부한 웹페이지를 가져올 때 대량의 HTTP 트랜잭션을 수행
- 페이지 레이아웃을 서술하는 HTML '뼈대'를 한 번의 트랜잭션으로 가져온 뒤, 첨부된 이미지, 그래픽 조각, 자바 애플릿 등을 가져오기 위해 추가로 HTTP 트랜잭션 수행.
- 리소스들은 서로 다른 서버에 위치할 수 있음

※ 자바 애플릿: 자바 바이트코드 형태로 배포되는 애플릿
※ 애플릿: 플러그인의 하나로, 큰 프로그램 범위 내에서 실행되는 특정한 작업을 수행하는 조그마한 응용 프로그램



HTTP 개관

HTTP 메시지
- 줄 단위의 문자열로, 일반 텍스트이기 때문에 사람이 읽고 쓰기 쉬움
- HTTP 요청 메시지, HTTP 응답 메시지로 구성

1. 요청줄: HTTP메서드, 대상(URL), HTTP버전
2. 응답줄: HTTP버전, 상태코드
3. 헤더: 요청 또는 응답에 대한 부가적인 정보, 헤더나 엔터티 본문이 있든 없든 **항상 빈 줄(CRLF)로 끝남!**
헤더의 종류로는 공통헤더, 요청헤더, 응답헤더, 엔터티헤더 존재
4. 엔터티 본문: 가공되지 않은 데이터로, 엔터티 헤더는 데이터의 의미를 설명함
엔터티 헤더로는 Content-type, Content-Length 등

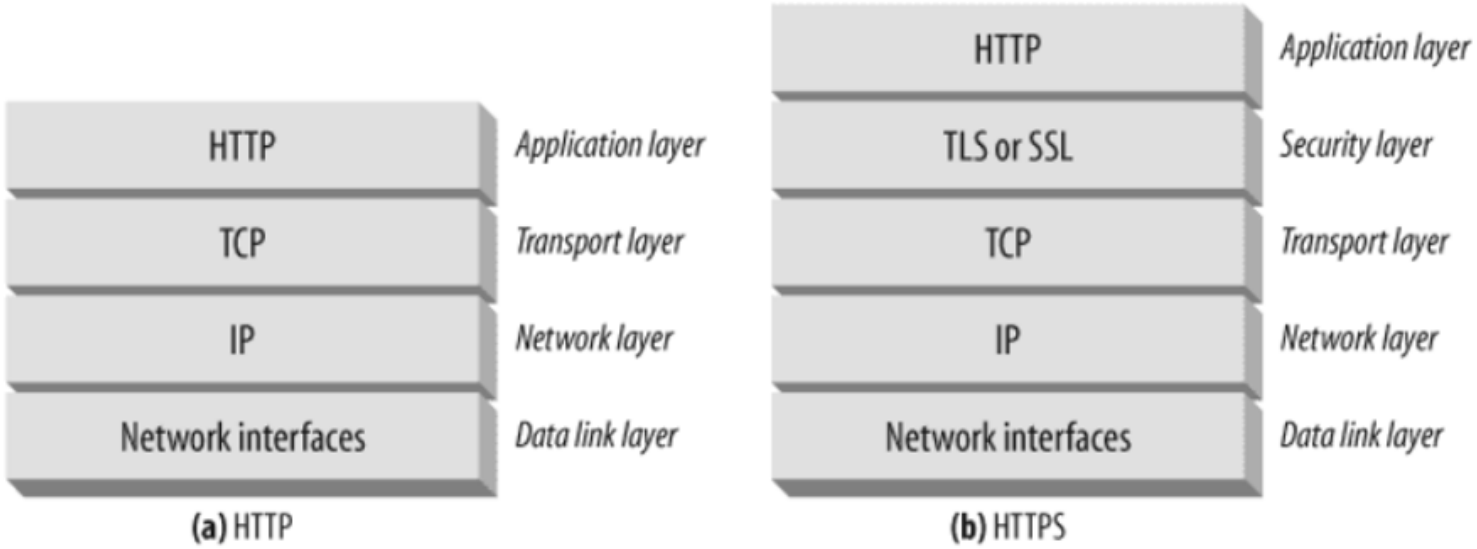
※ HTTP/0.9 버전은 헤더가 없었음. Content-type 기재가 불가능하므로 HTML 문서만 전달이 가능했음.
※ CRLF(Carriage Return Line Feed)는 줄바꿈을 의미. line break, EOL(End Of Line)과 통용

	(a) 요청 메시지	**(b) 응답 메시지**
시작줄	GET /test/hi-there.txt HTTP/1.0	HTTP/1.0 200 OK
헤더	Accept: text/* Accept-Language : en, fr	Content-type: text/plain Content-length: 19
엔터티 본문		Hi! I'm a message!

HTTP 개관

TCP 커넥션
- HTTP는 애플리케이션 계층 프로토콜
- HTTP는 네트워크 통신의 세부사항에 대해서 신경쓰지 않고, TCP/IP 프로토콜에게 맡김

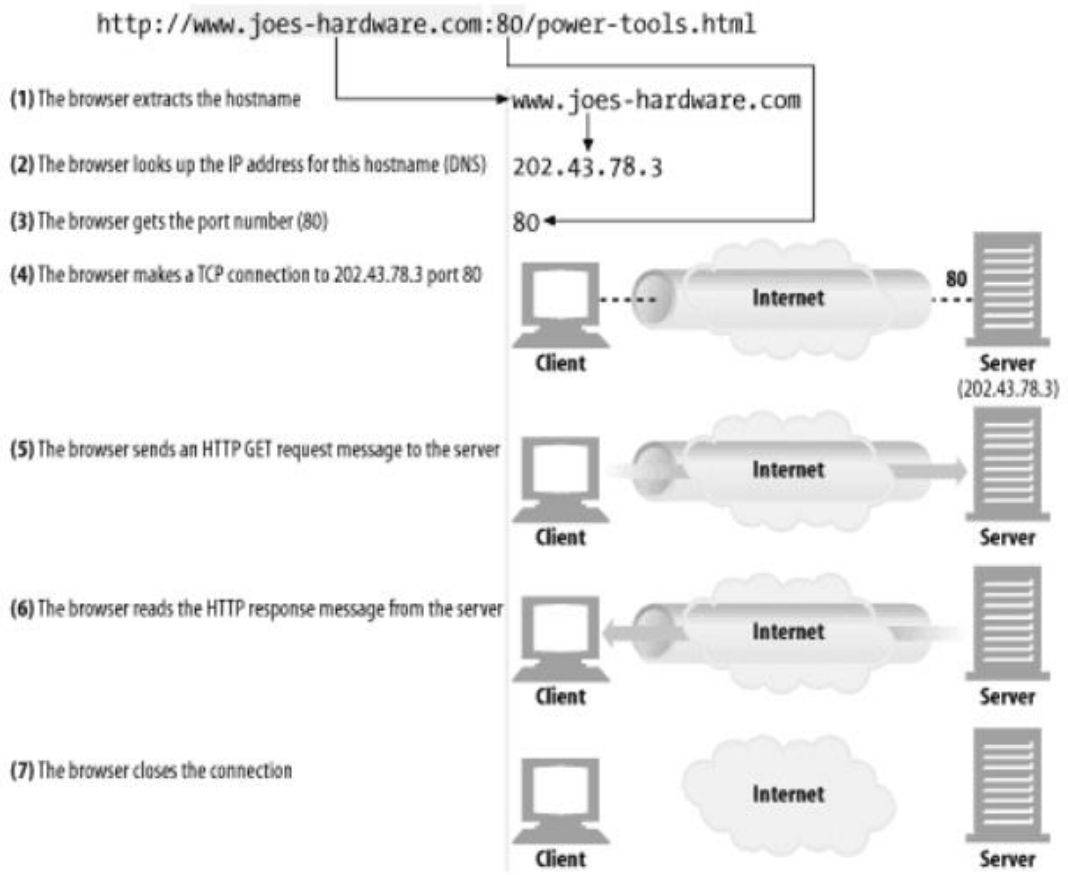
TCP/IP 프로토콜의 기능
- 오류 없는 데이터 전송
- 순서에 맞는 데이터 전달(보낸 순서대로 전달)
- 조각나지 않는 데이터 스트림(어떤 크기로든 보낼 수 있음)



HTTP 개관

전 세계 모든 HTTP 통신은 TCP/IP를 통해 이루어짐
인터넷 주소창에 http://www.joes-hardware.com:80/power-tools.html을 입력하면 브라우저는 다음을 수행

(1) 브라우저가 www.joes-hardware.com 호스트 호출
(2) 브라우저가 이 호스트 명에 대한 IP주소를 찾음
(3) 브라우저가 포트 번호(80)을 얻음
(4) 브라우저가 202.43.78.3의 80포트로 **TCP 커넥션** 생성
(5) 브라우저가 서버로 HTTP GET 요청 메시지 보냄
(6) 브라우저가 서버에서 온 HTTP 응답 메시지 읽음
(7) 브라우저가 커넥션을 끊음



URL의 예
http://207.200.83.29:80/index.html
http://www.netscape.com:80/index.html
http://www.netscape.com/index.html
(포트 번호가 빠진 경우, 기본값 80)

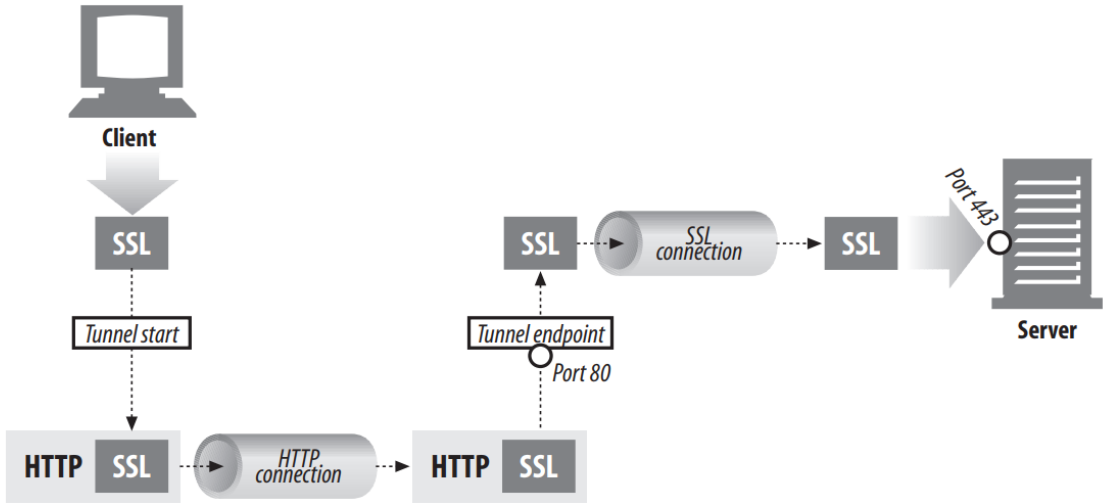
HTTP 개관

웹의 구성요소
인터넷과 상호작용할 수 있는 웹 애플리케이션은 많다.
※웹 애플리케이션: 인터넷을 통해, 브라우저에서 이용할 수 있는 소프트웨어(웹 클라이언트, 웹 서버 etc..)
※애플리케이션: 운영체제에서 실행되는 모든 소프트웨어, 사용자와 상호작용이 가능한 프로그램
※웹(World Wide Web, WWW, W3): 인터넷에 연결된 컴퓨터를 통해 정보를 공유할 수 있는 정보 공간
1. 프록시(Proxy)
- 보안, 성능 최적화를 위해 사용
(바이러스 검출, 성인 콘텐츠 차단, 프록시 캐시 등)

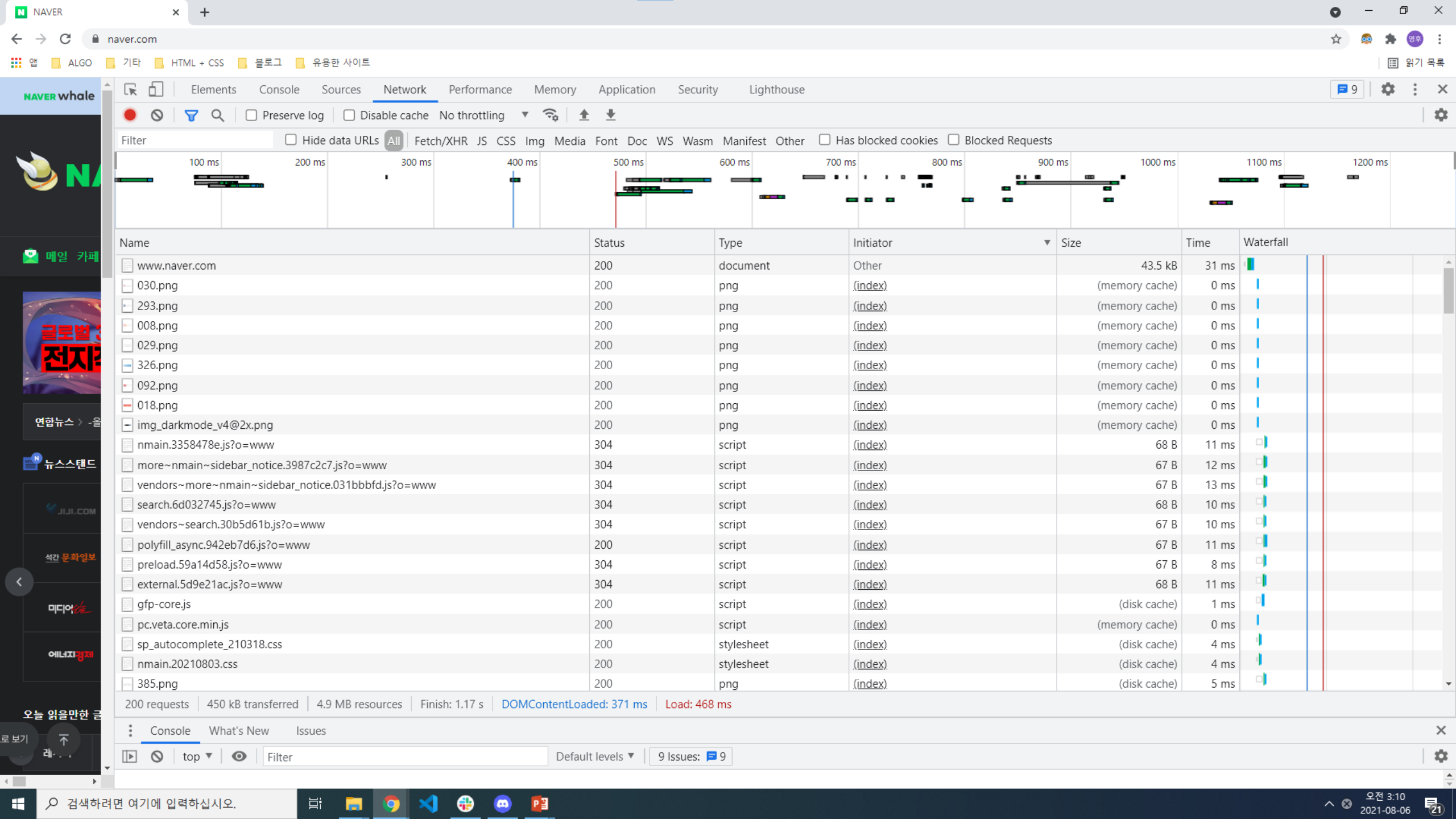
2. 캐시
3. 게이트웨이
- 다른 서버들의 중개자로 동작하는 서버
- 진짜 서버인 것처럼 요청을 다루므로, 클라이언트는 게이트웨이와 통신하고 있음을 알아채지 못함
- FTP URI에 대한 HTTP 요청을, FTP 프로토콜을 이용해 문서를 가져와서 HTTP메시지에 담아 응답

HTTP 개관

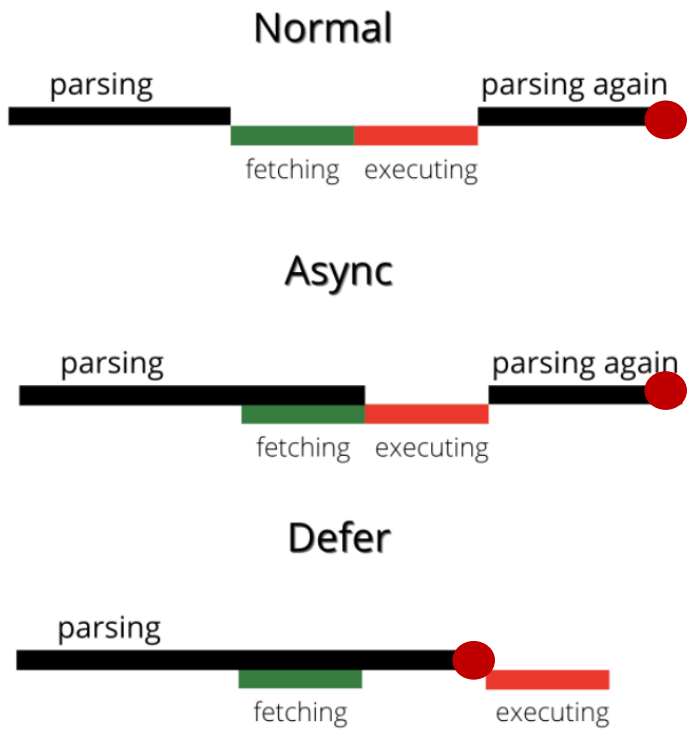
웹의 구성요소
터널
HTTP 커넥션 안에, HTTP가 아닌 트래픽을 올릴 수 있음



에이전트(사용자 에이전트)
- 사용자를 위해 HTTP 요청을 만들어주는 프로그램
- 대표적인 예로 웹 브라우저가 있음
- 이외에도 사람의 통제 없이 스스로 웹을 돌아다니며 HTTP 트랜젝션을 일으키고 콘텐츠를 받아오는 자동화된 사용자 에이전트 존재

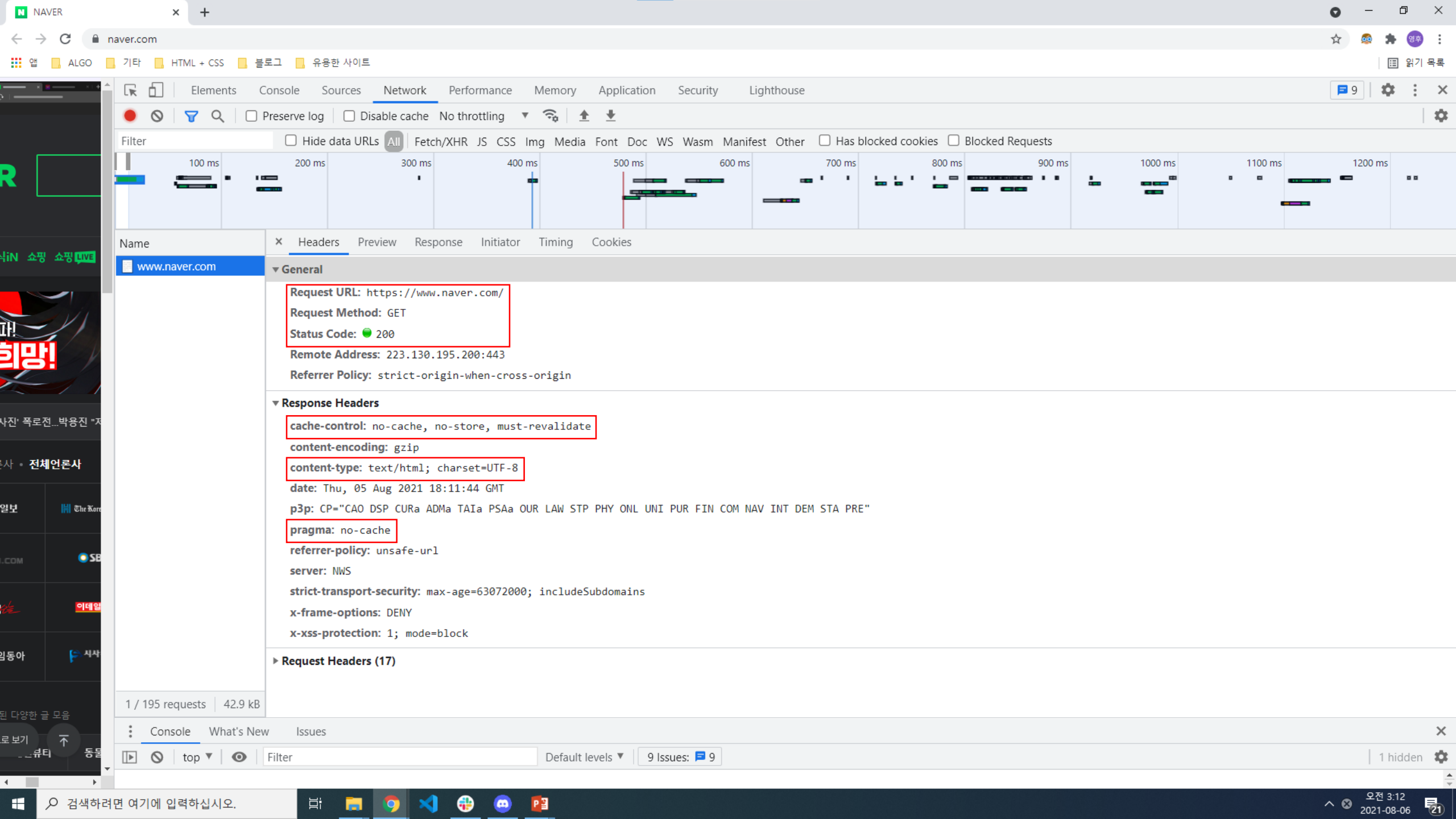

NAVER
naver.com
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
Fetch/XHR
JS
CSS
Img
Media
Font
Doc
WS
Wasm
Manifest
Other
Has blocked cookies
Blocked Requests
100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms
800 ms
900 ms
1000 ms
1100 ms
1200 ms
Name
Status
Type
Initiator
Size
Time
Waterfall
www.naver.com 200 document Other 43.5 kB 31 ms
030.png 200 png (index) (memory cache) 0 ms
293.png 200 png (index) (memory cache) 0 ms
008.png 200 png (index) (memory cache) 0 ms
029.png 200 png (index) (memory cache) 0 ms
326.png 200 png (index) (memory cache) 0 ms
092.png 200 png (index) (memory cache) 0 ms
018.png 200 png (index) (memory cache) 0 ms
img_darkmode_v4@2x.png 200 png (index) (memory cache) 0 ms
nmain.3358478e.js?o=www 304 script (index) 68 B 11 ms
more~nmain~sidebar_notice.3987c2c7.js?o=www 304 script (index) 67 B 12 ms
vendors~more~nmain~sidebar_notice.031bbbfd.js?o=www 304 script (index) 67 B 13 ms
search.6d032745.js?o=www 304 script (index) 68 B 10 ms
vendors~search.30b5d61b.js?o=www 304 script (index) 67 B 10 ms
polyfill_async.942eb7d6.js?o=www 200 script (index) 67 B 11 ms
preload.59a14d58.js?o=www 304 script (index) 67 B 8 ms
external.5d9e21ac.js?o=www 304 script (index) 68 B 11 ms
gfp-core.js 200 script (index) (disk cache) 1 ms
pc.veta.core.min.js 200 script (index) (memory cache) 0 ms
sp_autocomplete_210318.css 200 stylesheet (index) (disk cache) 4 ms
nmain.20210803.css 200 stylesheet (index) (disk cache) 4 ms
385.png 200 png (index) (disk cache) 5 ms
200 requests
450 kB transferred
4.9 MB resources
Finish: 1.17 s
DOMContentLoaded: 371 ms
Load: 468 ms
Console
What's New
Issues
top
Filter
Default levels
9 Issues: 9
검색하려면 여기에 입력하십시오.
오전 3:10
2021-08-06

적색 원에서 DOMContentLoaded 이벤트 발생



이미지출처

HTML 파싱 후(DOM 트리생성 후), DOMContentLoaded 이벤트 발생 ➔ 371ms
이미지까지 화면에 로드되는 시간 ➔ 468ms


NAVER
naver.com
앱 ALGO 기타 HTML + CSS 블로그 유용한 사이트
읽기 목록
Elements Console Sources Network Performance Memory Application Security Lighthouse
Preserve log Disable cache No throttling
Filter Hide data URLs All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other Has blocked cookies Blocked Requests
100 ms 200 ms 300 ms 400 ms 500 ms 600 ms 700 ms 800 ms 900 ms 1000 ms 1100 ms 1200 ms
Name
www.naver.com
Headers Preview Response Initiator Timing Cookies
General
Request URL: https://www.naver.com/
Request Method: GET
Status Code: 200
Remote Address: 223.130.195.200:443
Referrer Policy: strict-origin-when-cross-origin
Response Headers
cache-control: no-cache, no-store, must-revalidate
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Thu, 05 Aug 2021 18:11:44 GMT
p3p: CP="CAO DSP CURa ADMa TAIa PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"
pragma: no-cache
referrer-policy: unsafe-url
server: NWS
strict-transport-security: max-age=63072000; includeSubdomains
x-frame-options: DENY
x-xss-protection: 1; mode=block
Request Headers (17)
1 / 195 requests 42.9 kB
Console What's New Issues
top Filter Default levels 9 Issues: 9 1 hidden
검색하려면 여기에 입력하십시오.
오전 3:12
2021-08-06

URL과 리소스

URL과 리소스

친구와 complete-catalog.xls 파일을 공유하려고 했다면...

URL에 있기 전
"ftp.joes-hardware.com에 FTP로 접속해. 익명 사용자로 로그인한 다음 비밀번호로 네 이름을 입력해. pub 디렉터리로 이동한 다음, 바이너리 형식으로 전환해. 이제 complete-catalog.xls란 이름의 파일을 너의 로컬 파일 시스템에 내려 받은 다음 보면 될거야"

URL 등장 후
"브라우저에서 ftp://ftp.lots-o-books.com/pub/complete-catalog.xls를 열어봐"

- 애플리케이션이 리소스에 접근할 수 있는 영리한 방법 제공
- HTTP프로토콜이 아닌, 다른 프로토콜을 사용할 수 있음

1. 이메일 주소: mailto:president@whitehouse.gov
2. FTP(File Transfer Protocol): ftp://ftp.lots-o-books.com/pub/complete-price-list.xls
3. RTSP(Real Time Streaming Protocol): rtsp://www.joes-hardware.com:554/interview/cto_video

URL과 리소스

URL문법
- 스킴에 의존적
- 일반적으로 9개로 나뉘고, 중요한 것은 스킴, 호스트, 경로
  <스킴>://<사용자 이름>:<비밀번호>@<호스트>:<포트>/<경로>;<파라미터>?<질의>#<프래그먼트>

1. 스킴
어떤 프로토콜로 리소스를 요청하는가?

2. 호스트와 포트
- **호스트**는 장비/**포트**는 장비 내에서 리소스를 갖고 있는 서버
- TCP프로토콜을 사용하는 HTTP는 기본 포트가 80. SSL(Secure Socket Layer)프로토콜은 443
- 호스트 컴포넌트에는 호스트명 혹은 IP주소 기재
- 아래 두개의 URL은 동일한 리소스를 가리킴

http://www.joes-hardware.com:80/index.html
http://161.58.228.45:80/index.html

URL과 리소스

3. 사용자 이름과 비밀번호
- 데이터에 접근을 허용하기 전에 사용자 이름과 비밀번호를 요구
(1) 스킴, 호스트, 경로 ftp://ftp.prep.ai.mit.edu/pub/gnu
(2) 스킴, 사용자 이름, 호스트, 경로 ftp://anonymous@ftp.prep.ai.mit.edu/pub/gnu
(3) 스킴, 사용자 이름, 비밀번호, 호스트, 경로 ftp://anonymous:my_passwd@ftp.prep.ai.mit.edu/pub/gnu

(1)과 같이, 사용자 이름과 비밀번호가 삽입되있지 않을 경우, 기본 사용자 이름과 비밀번호 값을 넣어놓는다.

4. 경로
- 서버의 어디에 리소스가 있는지
- '/'문자를 기준으로 경로조각으로 나눔
- 각 경로조각은 자체 파라미터를 가질 수 있음
- 아래 URL의 경로 조각은 두개
http://www.joes-hardware.com:80/seasonal/index-fall.html

URL과 리소스

5. 파라미터
- 경로 정보만으로는 리소스를 찾지 못함. **보다 정확한 요청을 하기 위해 기술**
- 각 경로조각은 자체 파라미터를 가질 수 있음
- **';'**로 구분한 후, **이름:값** 쌍의 리스트로 기술

http://www.joes-hardware.com/hammers;sale=false/index.html;graphics=true

6. 질의 문자열(쿼리문, 쿼리스트링)
- 데이터베이스 같은 서비스들은 **리소스 형식의 범위를 좁히기 위해서** 질문을 받을 수 있음
- **이름:값** 쌍의 질의 문자열을 기술
- 여러 쌍을 전달하는 경우, **'&'**로 구분
- 아래 code=192150과 item=12731은 질의 컴포넌트

네이버 영화 모가디슈에 대한 주소 https://movie.naver.com/movie/bi/mi/basic.naver?code=192150
죠 컴퓨터 가게의 특정 물품 재고 조회 https://www.joes-hardware.com/inventory-check.cgi?item=12731

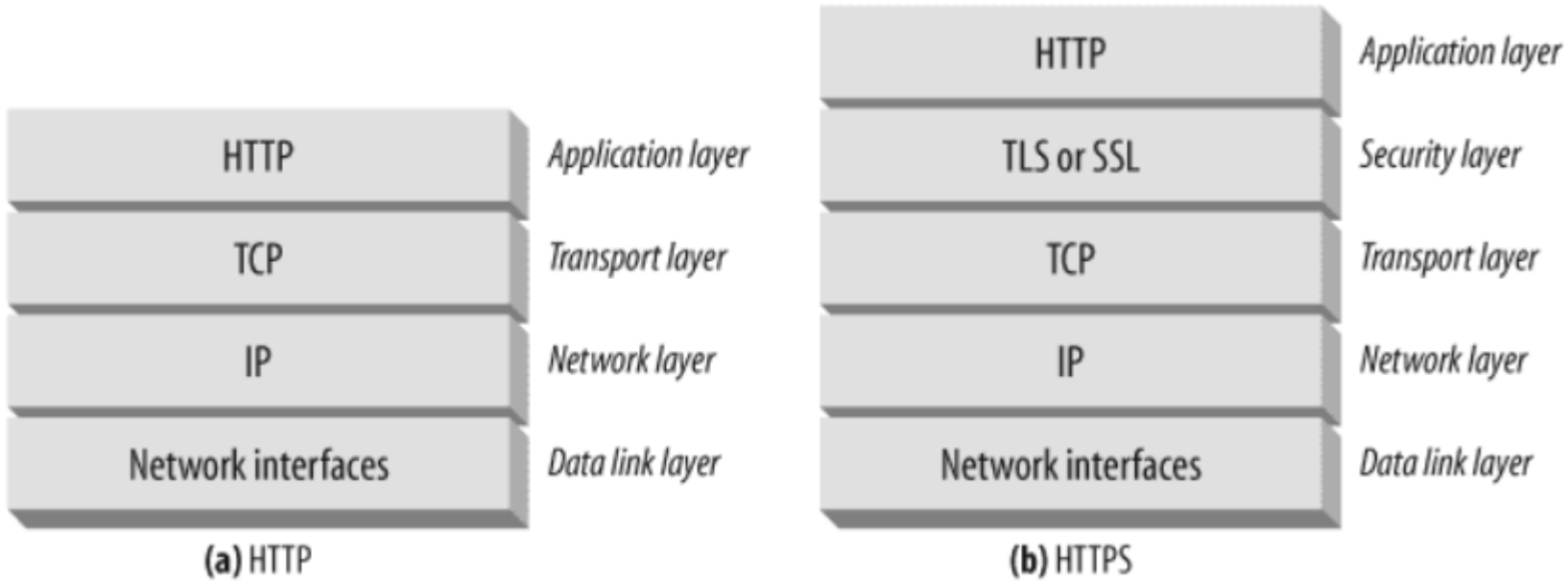
4장_커넥션 관리

계층별 데이터 단위

OSI 7계층 모델의 계층별 데이터 단위를 보면,

데이터 링크 계층에서는 DPDU(Datalink Protocol Data Unit)이며, 보통 **프레임**이라 부름
네트워크 계층에서는 NPDU(Network -)이며, 보통 **패킷**이라 부름
전송 계층에서는 TPDU(Transport -)이며, TCP에서는 **세그먼트**, UDP에서는 데이터그램이라 부름

TCP와 관계된 **프로토콜 스택**은 아래와 같음



이미지출처

TCP/IP 4계층 VS OSI 7계층

OSI 7계층 모델의 계층별 데이터 단위를 보면,

데이터 링크 계층에서는 DPDU(Datalink Protocol Data Unit)이며, 보통 **프레임**이라 부름
네트워크 계층에서는 NPDU(Network -)이며, 보통 **패킷**이라 부름
전송 계층에서는 TPDU(Transport -)이며, TCP에서는 **세그먼트**, UDP에서는 데이터그램이라 부름

TCP와 관계된 **프로토콜 스택**은 아래와 같음

이미지출처

TCP 커넥션

전 세계 모든 HTTP 통신은 TCP/IP를 통해 이루어짐
인터넷 주소창에 http://www.joes-hardware.com:80/power-tools.html을 입력하면 브라우저는 다음을 수행

(1) 브라우저가 www.joes-hardware.com 호스트 호출

(2) 브라우저가 이 호스트 명에 대한 IP주소를 찾음

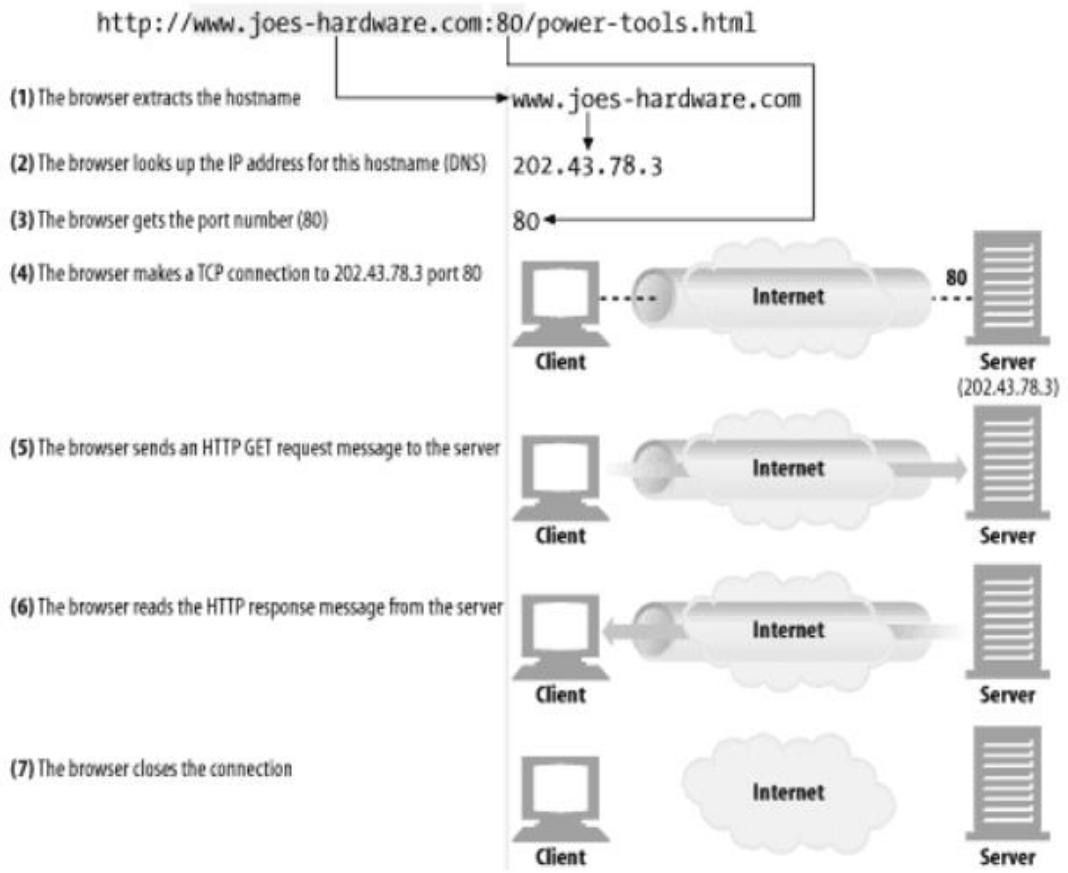
(3) 브라우저가 포트 번호(80)을 얻음

(4) 브라우저가 202.43.78.3의 80포트로 **TCP 커넥션** 생성

(5) 브라우저가 서버로 HTTP GET 요청 메시지 보냄

(6) 브라우저가 서버에서 온 HTTP 응답 메시지 읽음

(7) 브라우저가 커넥션을 끊음



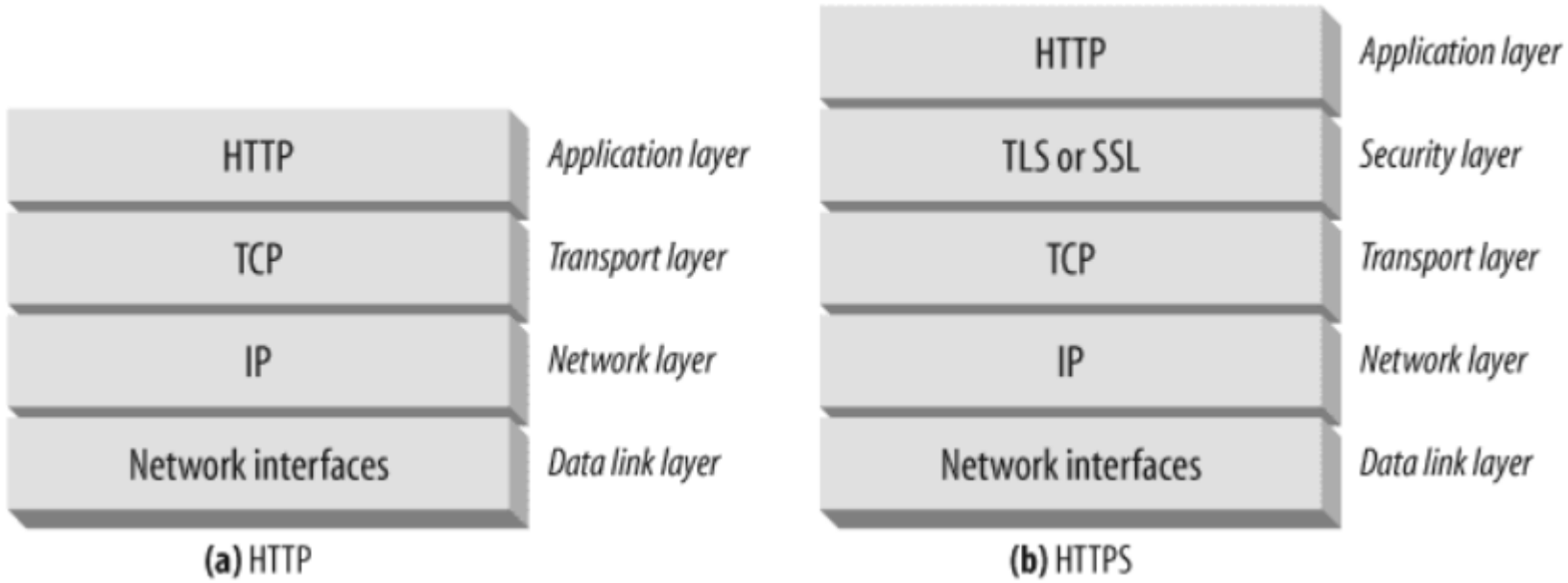
이미지출처

TCP 커넥션

OSI 7계층 모델의 계층별 데이터 단위를 보면,

데이터 링크 계층에서는 DPDU(Datalink Protocol Data Unit)이며, 보통 **프레임**이라 부름
네트워크 계층에서는 NPDU(Network -)이며, 보통 **패킷**이라 부름
전송 계층에서는 TPDU(Transport -)이며, TCP에서는 **세그먼트**, UDP에서는 데이터그램이라 부름

TCP와 관계된 **프로토콜 스택**은 아래와 같음

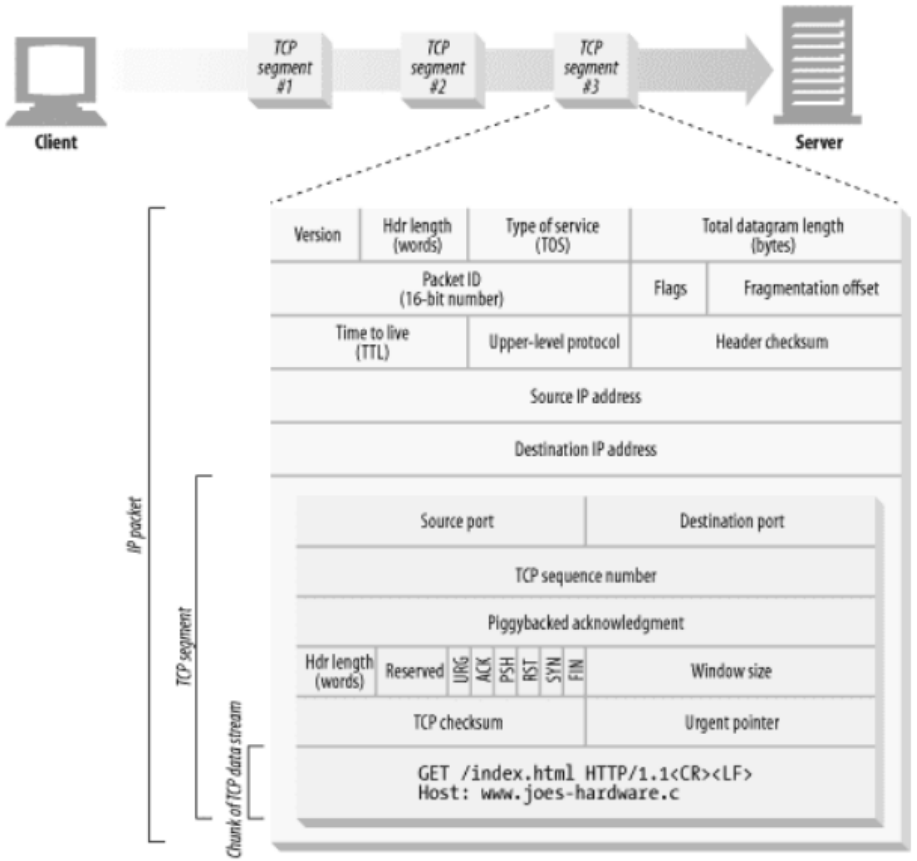


이미지출처

TCP 커넥션

TCP는 세그먼트 단위로 데이터 스트림을 자른 후, IP 패킷에 담아 인터넷을 통해 데이터를 전달
이 과정은 TCP/IP 소프트웨어에 의해서 처리

IP 패킷 헤더는 발신지와 목적지 IP주소, 크기, 기타 플래그를 가짐
TCP 세그먼트 헤더는 TCP 포트 번호, TCP 제어 플래그, 데이터의 순서와 무결성 검사를 위한 숫자 포함

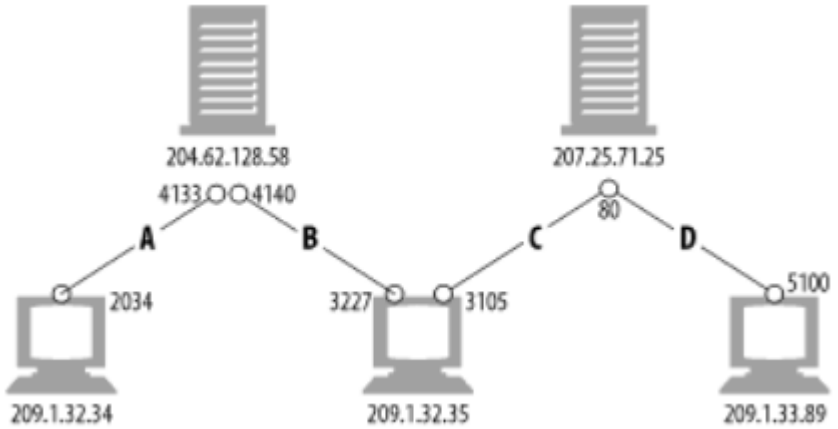


[이미지출처]

TCP 커넥션

TPC 커넥션 유지
- 컴퓨터는 항상 TCP 커넥션을 여러 개를 가지고 있음
- TCP는 포트 번호를 통해서 여러 개의 커넥션 유지
- IP 주소는 컴퓨터에 연결되고, 포트 번호는 애플리케이션에 연결(무슨 말인지 모르겠음. 찾아봐야 할듯)

TCP 커넥션은 네 가지 값으로 구성 <발신지 IP주소, 발신지 포트, 수신지 IP주소, 수신지 포트>
이 네 가지 값으로 **유일한** TCP 커넥션 생성. 네 값 모두 동일한 값을 가리키는 커넥션은 존재하지 않음



같은 포트를 가리키는 커넥션이 존재할 수 있음(C, D)
같은 발신지 IP 주소를 가리키는 커넥션이 존재할 수 있음(B와 C)
같은 IP 주소를 가리키는 커넥션이 존재할 수 있음(A, B, C, D). 하지

이미지출처

TCP 소켓 프로그래밍

4.3 HTTP 커넥션 관리

HTTP는 TCP 바로 위에 있는 계층이기 때문에, HTTP 트랜잭션의 성능은 TCP 성능에 영향을 받음

이미지출처

7장_캐시

cache의 정의와 이점

웹 캐시는 자주 쓰이는 문서의 사본을 자동으로 보관

장점

1. 불필요한 데이터 전송

복수의 클라이언트가 한 페이지에 접근할때, 서버는 같은 문서를 클라이언트들에게 각각 한 번씩 전송하게 됨
이로 인해, (1) 대역폭을 잡아먹음 (2) 전송을 느리게 만듬 (3) 웹서버의 부하 증가 등의 단점이 발생.

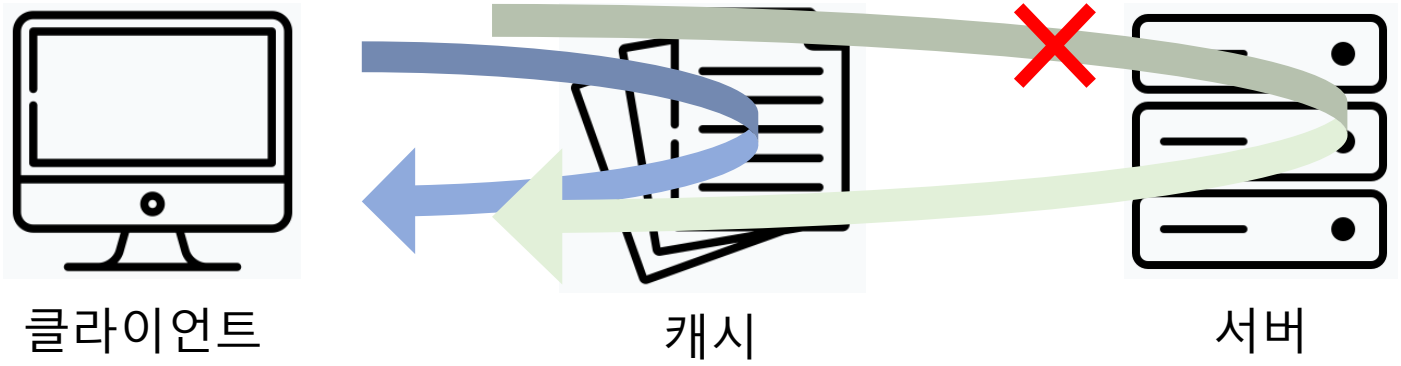
캐시된 사본이 뒤 이은 요청들에 대한 응답으로 사용될 수 있기 때문에, 원 서버가 중복해서 트래픽을 주고받는 낭비가 줄어들게 됨.

※ Traffic: 웹 사이트 방문자에 의해서 받거나 보내지는 데이터의 양

참고 트래픽과 대역폭 계산하기

1) 트래픽 계산
용량 * 사용자 수 * 개수 = 트래픽
예) 4GB 영화 * 10명 * 10개 = 400GB

2) 네트워크 대역폭 계산
(용량 * 사용자수 * 8(비트계산)) / 8 = bps



cache의 정의와 이점

웹 캐시는 자주 쓰이는 문서의 사본을 자동으로 보관

장점

1. 불필요한 데이터 전송

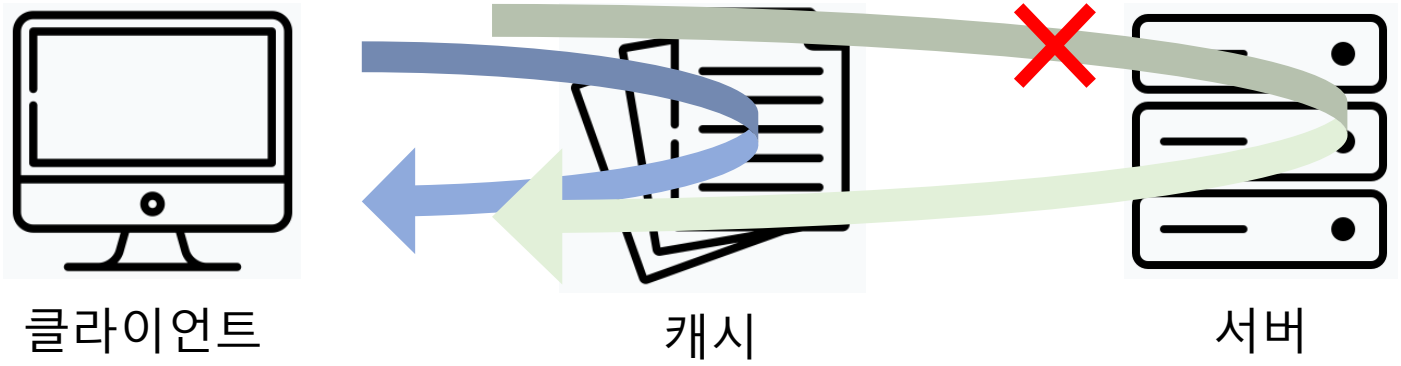
복수의 클라이언트가 한 페이지에 접근할때, 서버는 같은 문서를 클라이언트들에게 각각 한 번씩 전송하게 됨
이로 인해, (1) 대역폭을 잡아먹음 (2) 전송을 느리게 만듬 (3) 웹서버의 부하 증가 등의 단점이 발생.

캐시된 사본이 뒤 이은 요청들에 대한 응답으로 사용될 수 있기 때문에, 원 서버가 중복해서 트래픽을 주고받는 낭비가 줄어들게 됨.

※ Traffic: 웹 사이트 방문자에 의해서 받거나, 보내지는 데이터의 양

참고 트래픽과 대역폭 계산하기
1) 트래픽 계산
용량 * 사용자 수 * 개수 = 트래픽
예) 4GB 영화 * 10명 * 10개 = 400GB

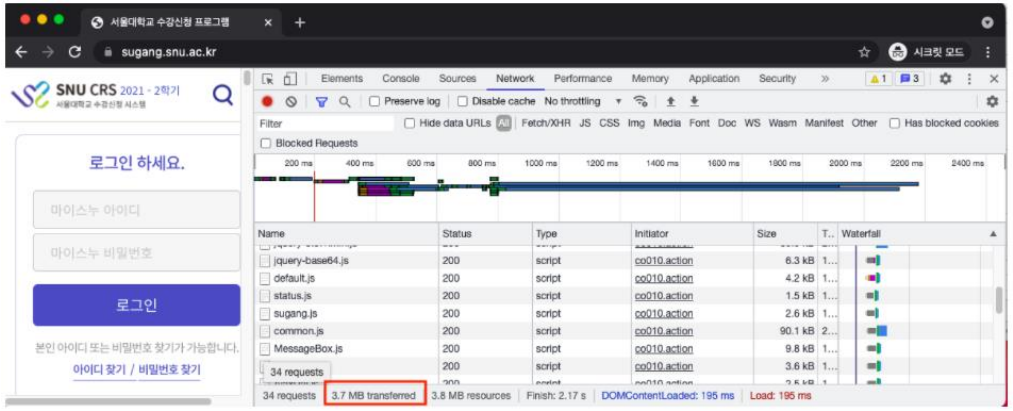
2) 네트워크 대역폭 계산
(용량 * 사용자수 * 8(비트계산)) / 8 = bps



cache의 정의와 이점

참고 AWS EC2 스펙 기준으로, 서울대 수강신청 기간에 얼만큼의 대역폭이 필요할까?

서울대 재적학생 수 2만명 및 서울대 수강신청 메인 페이지 기준 3.7MB(4MB) 네트워크 전송



재학생들이 한 번씩 접속 한다는 가정 하에 트래픽량 계산
20,000명 * 4MB = 80,000MB = 80GB 트래픽 발생

10분 이내에 수강신청이 완료된다는 가정 하에 대역폭 계산
(20,000명 * 4MB * 8) / 10 * 60s = 1066Mbps = 1.066Gbps

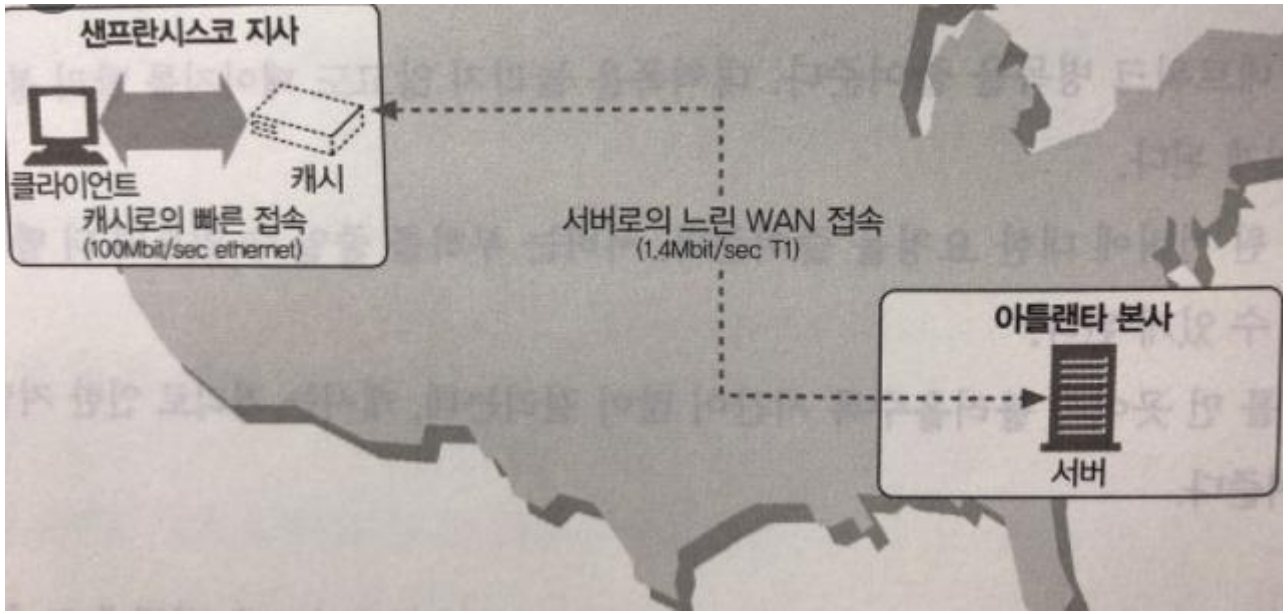
인스턴스	vCPU	메모리(GiB)	인스턴스 스토리지(GiB)	네트워킹 성능(Gbps)	EBS 대역폭(Mbps)
r5.large	2	16	EBS 전용	최대 10	최대 4,750

참조

cache의 정의와 이점

2. 대역폭 병목

- 네트워크는 원격 서버보다 로컬 네트워크 클라이언트에 더 넓은 대역폭 제공
- 클라이언트들이 서버에 접근할 때의 속도는, 그 경로에 있는 가장 느린 네트워크의 속도와 같음



cache의 정의와 이점

3. 갑작스러운 요청 쇄도(Flash Crowds)
캐시는 갑작스러운 요청 쇄도(뉴스 속보, 스팸 메일 등으로 인한)에 대처하기 위해 중요
불필요한 트래픽 급증은, 네트워크와 웹 서버의 심각한 장애를 야기

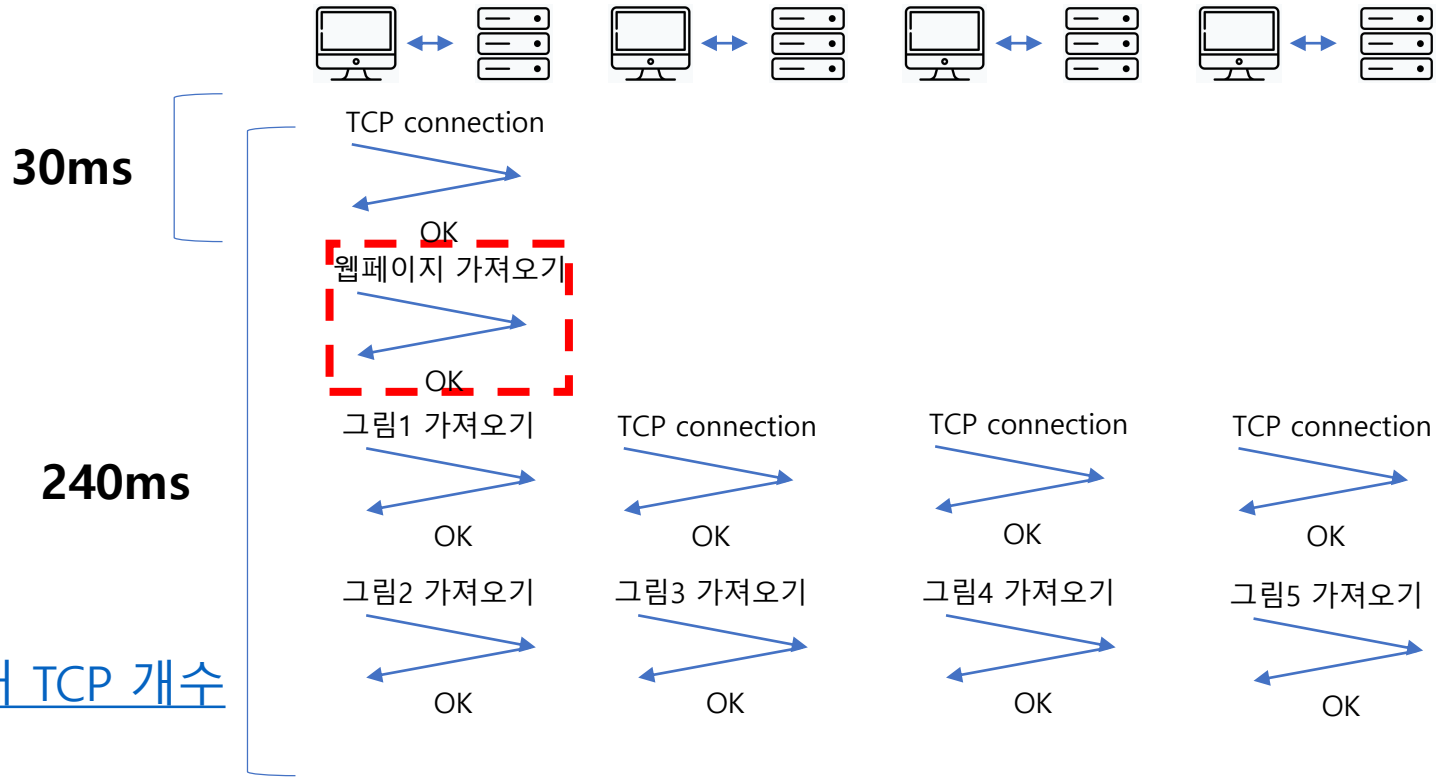
예)
1998년 9월 11일, 케네스 스타가 클린턴 미 대통령에 대한 조사 내용이 담긴 '스타 보고서'가 인터넷에 공개된 날, 미 하원 웹 서버는 한 시간 동안 3백만 건이 넘는 요청을 받았고, 이는 평소에 50배.

CNN.com은 웹 서버가 평균 매초 50,000건이 넘는 요청을 받음.

cache의 정의와 이점

4. 거리로 인한 지연
- 빛의 속도(실제 신호는 빛 보다 느리다) 그 자체가 유의미한 지연을 유발

예)
(1)보스턴과 샌프란시스코 사이의 거리가 약 4,400km 빛의 속도(300,000km/s)로 왕복하기 위해서 30ms 소요.
(2)샌프란시스코 서버에 이미지 20개가 있을 때, 보스턴에 있는 클라이언트가 서버로 동시에 네 개의 커넥션을 열고, 이미지를 받아온다면, 빛의 속도로 인한 지연은 240ms



참고 브라우저가 도메인마다 연결 가능한 최대 TCP 개수

cache의 특징

적중과 부적중
- cache에서 원하는 데이터를 찾으면 cache hit
- cache에서 원하는 데이터가 없으면 cache miss
- hit / (hit + miss) = cache hit ratio

(1) cache hit

(2) cache miss

(3) cache revalidation hit

추가 Reference

https://web-km.tistory.com/17

HTTP와 HTTPS 차이

https://web-km.tistory.com/17