01 HTTP 개관

김수빈(kimziou77@naver.com)

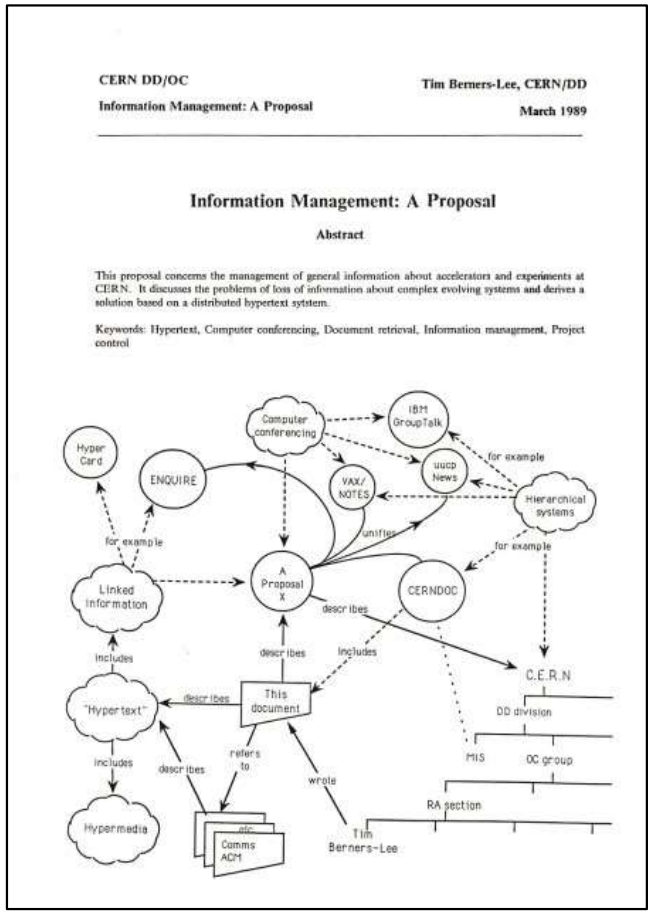
2022.02.07.

HTTP

WWW에서 통신하는데 사용하는 프로토콜 프로그램

WWW

팀 버너스리 아저씨



CERN DD/OC

Tim Berners-Lee, CERN/DD

Information Management: A Proposal

March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

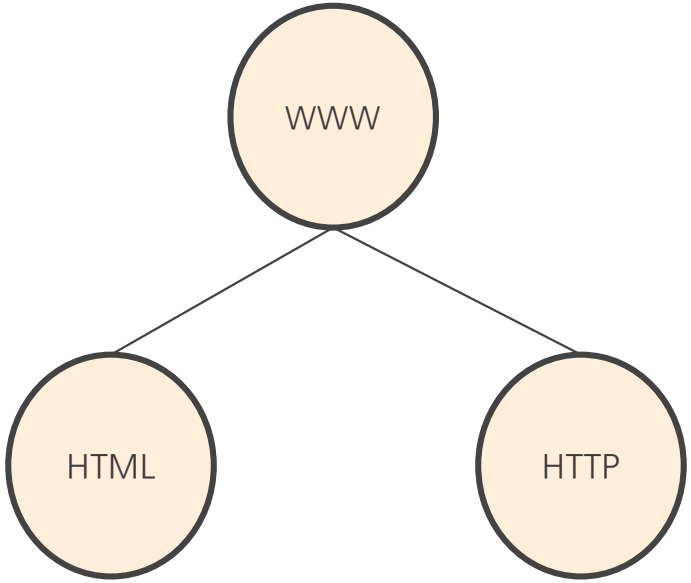
Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

https://cds.cern.ch/record/369245/files/dd-89-001.pdf

"누구나 조건 없이 동등하게 웹에 접근할 수 있고 이용할 수 있어야 한다"

WWW

하이퍼텍스트 개념을 도입함으로써
전세계 네트워크에 흩어져 있는 다양한 형태의 자료들을 연결(link)할 수 있게 됐다.



HyperText Markup Language

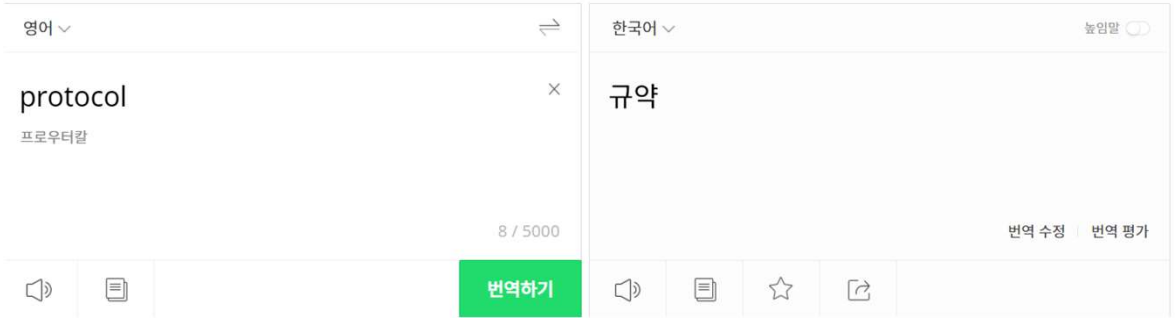
HyperText Transfer Protocol

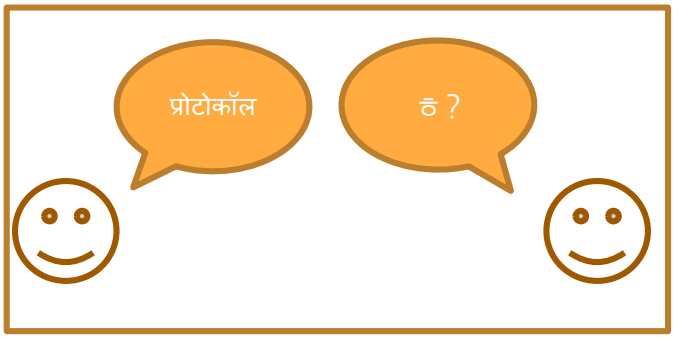
자원 하나하나를 개별적으로 찾아갈 필요 없이
일정한 규칙(HTML)에 따라 모여서 일정한
형식을 갖춰 컴퓨터에 나타나도록 했다.

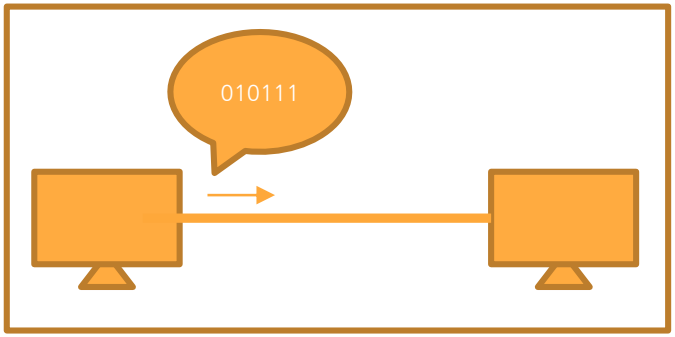
HTTP는 웹의 구성요소들이 서로
통신할때 사용하는 프로토콜

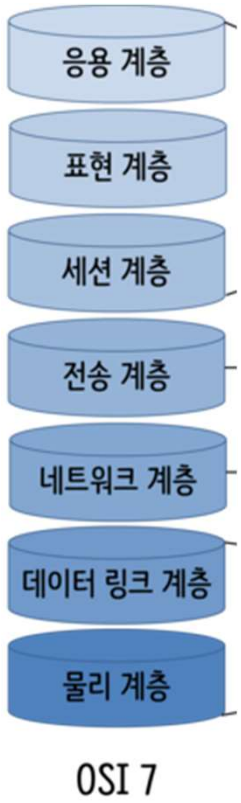
HTTP

WWW에서 **통신**하는데 사용하는 **프로토콜** 프로그램





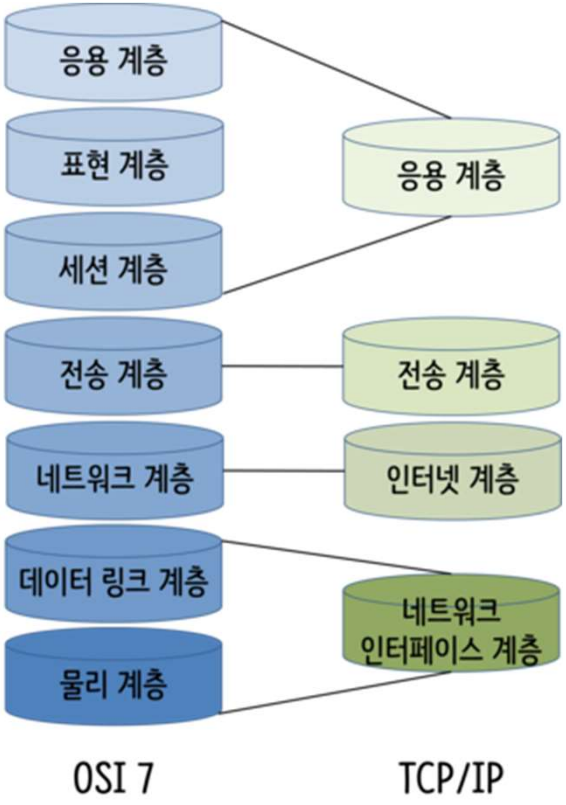


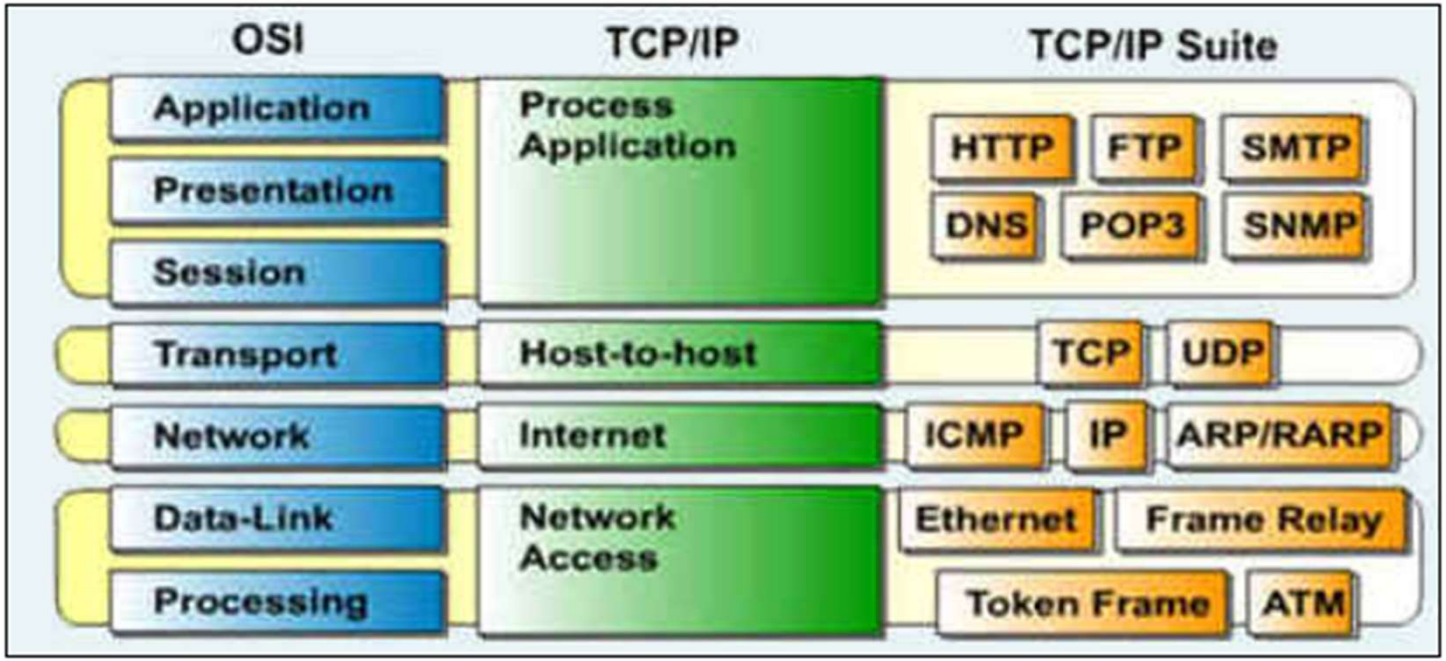


OSI 7

→ 프로토콜(Protocol)이 필요하다

OSI 7계층 vs TCP/IP





네트워크 통신의(OSI, Open Systems Interconnection)모델

왜 HTTP를 이해하는 것이 중요한가?

✓ 선호하는 언어가 바뀌고

✓ 대세인 프레임워크가 바뀌더라도

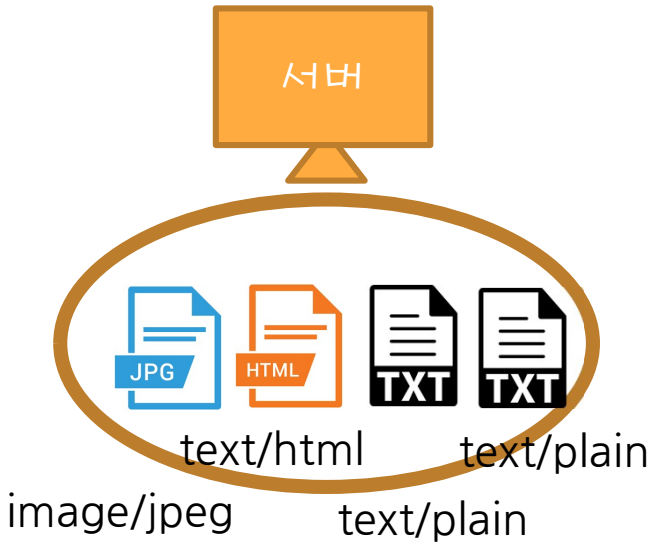
✓ 웹 애플리케이션이 HTTP를 사용한다는 사실은 변하지 않는다.

✓ HTTP를 이해한다는 것은 웹이 어떻게 동작하는지 이해한다는 것

목차

1.03 리소스(데이터)

- ✓ 서버는 모든 HTTP 객체에 MIME타입을 붙인다.
- ✓ MIME : Multipurpose Internet Mail Extensions (미디어 타입)
- ✓ URI



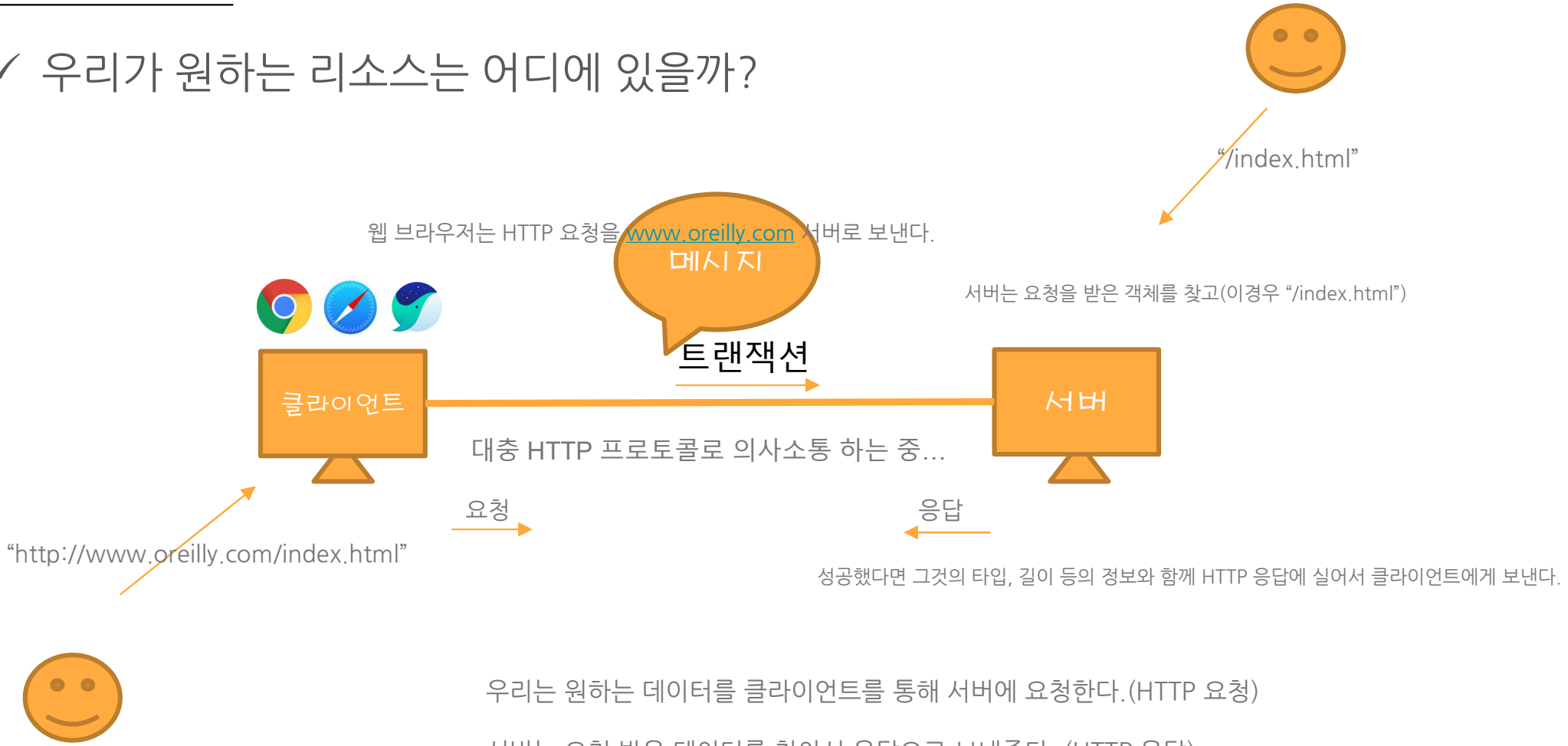
MIME
주타입/서브타입
매개변수 존재할 경우 세미콜론으로 시작

Content-Type: video/quicktime
Content-Type: text/html; charset="iso-8859-6"
Accept: image/gif

1.02 웹 클라이언트와 서버

클라이언트와 서버가 리소스를 주고받기 위해 HTTP를 어떻게 사용하는지 자세히 알아보자

✓ 우리가 원하는 리소스는 어디에 있을까?



우리는 원하는 데이터를 클라이언트를 통해 서버에 요청한다.(HTTP 요청)

서버는 요청 받은 데이터를 찾아서 응답으로 보내준다. (HTTP 응답)

1.05 트랜잭션

- ✓ HTTP 트랜잭션은 **요청명령/응답결과**로 구성되어있음
- ✓ HTTP 메시지 라고 불리는 정형화된 데이터 덩어리를 통해 이루어짐
- ✓ 일반 텍스트 구조 (이진데이터 x) - 사람들이 읽고 쓰기 쉬움

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

1.04 (1) 요청 메시지

- ✓ HTTP 메서드라고 불리는 여러가지 종류의 요청 명령을 지원한다.
- ✓ 메서드는 서버에게 어떤 동작이 취해져야 하는지 말해준다.
- ✓ 모든 HTTP 요청 메시지는 한 개의 메서드를 갖는다.

요청 메시지

시작줄 GET /test/hi-there.txt HTTP/1.0
메서드 / 리소스 / 버전번호

헤더
Accept : text/*
Accept-Language: en.fr

HTTP 메서드	설명
GET	서버에서 클라이언트로 지정한 리소스를 보내라
PUT	클라이언트에서 서버로 보낸 데이터를 지정한 이름의 리소스로 정해라
DELETE	지정한 리소스를 서버에서 삭제해라
POST	클라이언트 데이터를 서버 게이트웨이 애플리케이션으로 보내라
HEAD	지정한 리소스에 대한 응답에서 헤더부분만 보내라

1.04 (2) 응답 메시지

- ✓ 모든 HTTP 응답 메시지는 상태코드와 함께 반환된다.
- ✓ 각 숫자 상태코드에 텍스트로 된 “사유구절”도 함께 보낸다.
- ✓ 실제 응답처리에는 숫자로 된 코드가, 구문은 설명만을 위해서 포함되었다.

응답 메시지

버전번호 / 상태코드 / 사유구절

시작줄	HTTP/1.0 200 OK
헤더	Content-type: text/plain Content-length: 19
본문	Hi! I'm a message!

HTTP 상태코드	설명
200	정상반환, ㅇㅋ
302	여기 아니고 다른곳에서 리소스 찾아
404	리소스 없음

1.01 HTTP: 인터넷의 멀티미디어 배달부

HTTP는 **신뢰성있는 데이터 전송 프로토콜**을 사용하기 때문에,
데이터가 지구 반대편에서 오더라도 전송중 손상되거나 꼬이지 않음을 보장한다.(4p)

…

HTTP는 애플리케이션 계층 프로토콜이다.
따라서 네트워크 통신의 핵심적인 세부사항에 대해서 신경쓰지 않는데,
대신 **TCP/IP**에게 맡긴다.

1.06 TCP (Transmission Control Protocol) 커넥션 (1)

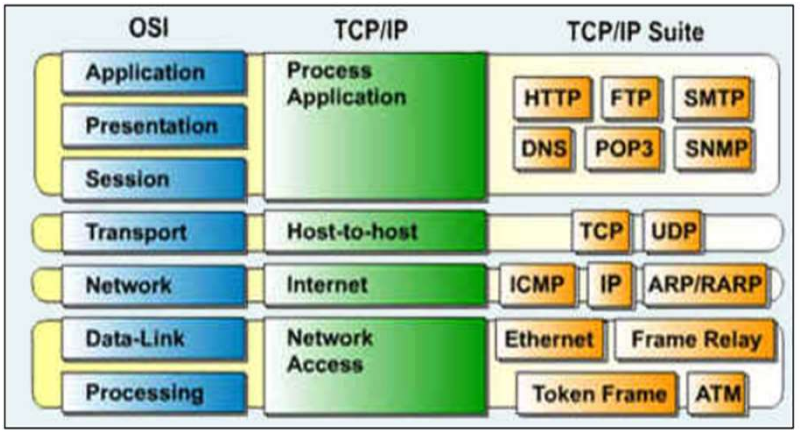
- ✓ 각 네트워크와 하드웨어의 특성을 숨기고
- ✓ 어떤 종류의 컴퓨터나 네트워크든 서로 **신뢰성 있는 의사소통**을 하게 해준다.
- ✓ **UDP(User Datagram Protocol)** 는 신뢰성 있는 데이터 전송을 **보장하지 않는다**.

- ✓ 회선교환(Circuit Switching) 방식

  경로를 정해놓는 방식. 해당 경로가 안되면 **통신이 바로 끊긴다**.

- ✓ 패킷교환(Packet Switching) 방식

  경로가 정해져있지 않고, 여러곳을 우회해서 갈 수 있음.
  단, 네트워크 환경의 안정성 떨어짐. 중간에 데이터 유실, 늦은 전송 등
  보낸순서와 불일치하게 도착할 수 있음



1.06 TCP 커넥션 (2) - 신뢰성의 정체

✓ **확인응답**을 받아 송수신시 계속 데이터가 잘 갔는지를 확인

✓ TCP 프로토콜의 작동은 크게 세가지 흐름으로 구분
- 1. 연결생성 (3-way handshaking과정을 통해 연결설정)
- 2. 자료전송 (순서제어, 흐름제어, 혼잡제어, 오류제어)
- 3. 연결종료 (4-way handshaking과정을 통해 연결해제)

✓ 3-way handshaking
- 서로 연결되어 있다는 것을 보장한다.

✓ 4-way handshaking
- 연결을 중단할때 한 번 더 확인하고 중단함.

이러한 제어가 있어 신뢰성 있는 데이터 전송이 가능하다

1.07 프로토콜 버전(1)

- ✓ TCP/IP 위에서 동작하는 ASCII 프로토콜
- ✓ 클라이언트-서버 에서의 요청-응답 프로토콜

  - ✓ **HTTP 0.9 (1991)**

```
GET /mypage.html
```

```
<HTML>
A very simple HTML page
</HTML>
```

  - ✓ **HTTP 1.0 (1991~1996)**

```
GET /mypage.html HTTP/1.0
User-Agent: NCSA_Mosaic/2.0 (Windows 3.1)
```

```
200 OK
Date: Tue, 15 Nov 1994 08:12:31 GMT
Server: CERN/3.0 libwww/2.17
Content-Type: text/html
<HTML>
A page with an image
  <IMG SRC="/myimage.gif">
</HTML>
```

```
200 OK
Date: Tue, 15 Nov 1994 08:12:32 GMT
Server: CERN/3.0 libwww/2.17
Content-Type: text/gif
(image content)
```

hypertext transfer → hypermedia transport ?

1.07 프로토콜 버전(2)

- ✓ **HTTP 1.0+**
  `keep-alive 커넥션` `가상호스팅` `프락시 연결` 등을 지원
- ✓ **HTTP 1.1 (1997.1.)**

```
GET /en-US/docs/Glossary/Simple_header HTTP/1.1
Host: developer.mozilla.org
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://developer.mozilla.org/en-US/docs/Glossary/Simple_header
```

```
200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 20 Jul 2016 10:55:30 GMT
Etag: "547fa7e369ef56031dd3bff2ace9fc0832eb251a"
Keep-Alive: timeout=5, max=1000
Last-Modified: Tue, 19 Jul 2016 00:59:33 GMT
Server: Apache
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding

(content)
```

- ✓ **HTTP 2.0 (2015.5. ~)**
  전송성능을 개선하고, 낮은 지연속도와 높은 처리량을 가능하게 만드는 것

- ✓ **HTTP 3.0 (Internal Draft)**
  UDP 기반인 QUIC 프로토콜 위에서 동작한다

https://quicwg.org/base-drafts/draft-ietf-quic-http.html

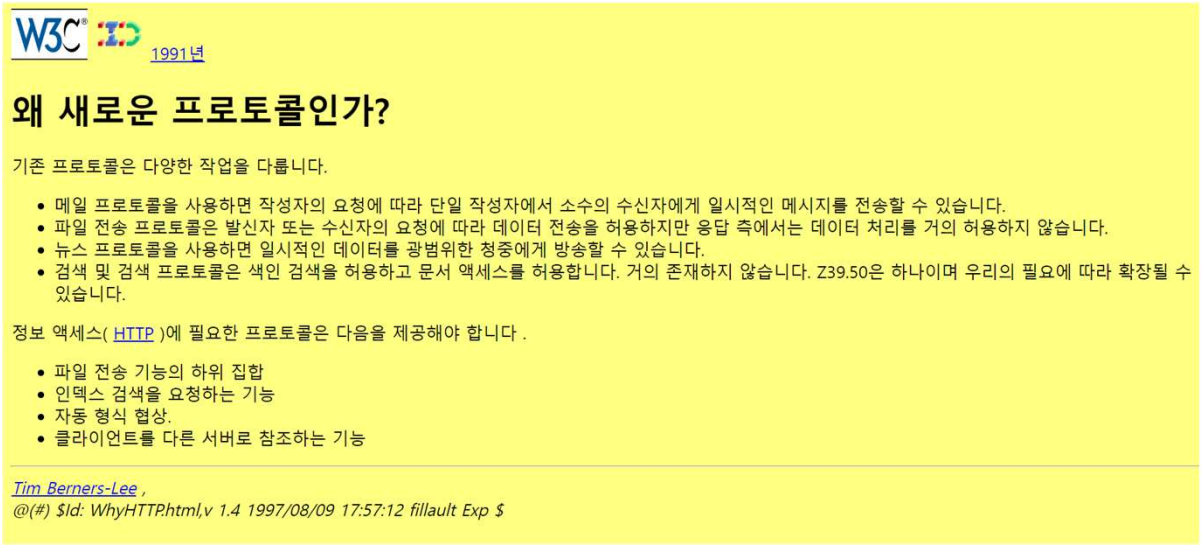
1.08 웹의 구성요소

인터넷과 상호작용할 수 있는 웹 애플리케이션은 많다.

- ✓ 캐시
  - 많이 찾는 웹페이지를 클라이언트 가까이에 보관하는 HTTP 창고
- ✓ 프락시
  - 클라이언트와 서버 사이에 위치한 HTTP 중개자 (HTTP-HTTP)
- ✓ 게이트웨이
  - 다른 애플리케이션과 연결된 특별한 웹서버 (HTTP-비HTTP)
- ✓ 터널
  - 날(raw) 데이터를 열어보지 않고 그대로 전달해주는 HTTP 애플리케이션 (비HTTP-HTTP)
- ✓ 에이전트
  - 자동화된 HTTP 요청을 만드는 준지능적 웹클라이언트

1.10 추가정보

- ✓ http://www.w3.org/
- ✓ http://www.w3.org/Protocols/
- ✓ http://www.w3.org/Protocols/WhyHTTP.html
- ✓ http://www.w3.org/History.html



1991년

왜 새로운 프로토콜인가?

기존 프로토콜은 다양한 작업을 다룹니다.

- 메일 프로토콜을 사용하면 작성자의 요청에 따라 단일 작성자에서 소수의 수신자에게 일시적인 메시지를 전송할 수 있습니다.
- 파일 전송 프로토콜은 발신자 또는 수신자의 요청에 따라 데이터 전송을 허용하지만 응답 측에서는 데이터 처리를 거의 허용하지 않습니다.
- 뉴스 프로토콜을 사용하면 일시적인 데이터를 광범위한 청중에게 방송할 수 있습니다.
- 검색 및 검색 프로토콜은 색인 검색을 허용하고 문서 액세스를 허용합니다. 거의 존재하지 않습니다. Z39.50은 하나이며 우리의 필요에 따라 확장될 수 있습니다.

정보 액세스(HTTP)에 필요한 프로토콜은 다음을 제공해야 합니다 .

- 파일 전송 기능의 하위 집합
- 인덱스 검색을 요청하는 기능
- 자동 형식 협상.
- 클라이언트를 다른 서버로 참조하는 기능

Tim Berners-Lee ,
@(#) $Id: WhyHTTP.html,v 1.4 1997/08/09 17:57:12 fillault Exp $

QnA