



HTTP와 웹 아키텍처 (ver1.0.1)

목 차

1. 웹 개요
2. HTTP 둘러보기
3. HTTP 이해
4. 웹 서버와 웹 아키텍처
5. 웹 애플리케이션 아키텍처
6. 웹 애플리케이션 트렌드



1. 웹(Web) 소개

- 1.1 웹이란?
- 1.2 웹의 등장
- 1.3 사용자 관점에서 보는 웹
- 1.4 요약

1.1 웹이란?

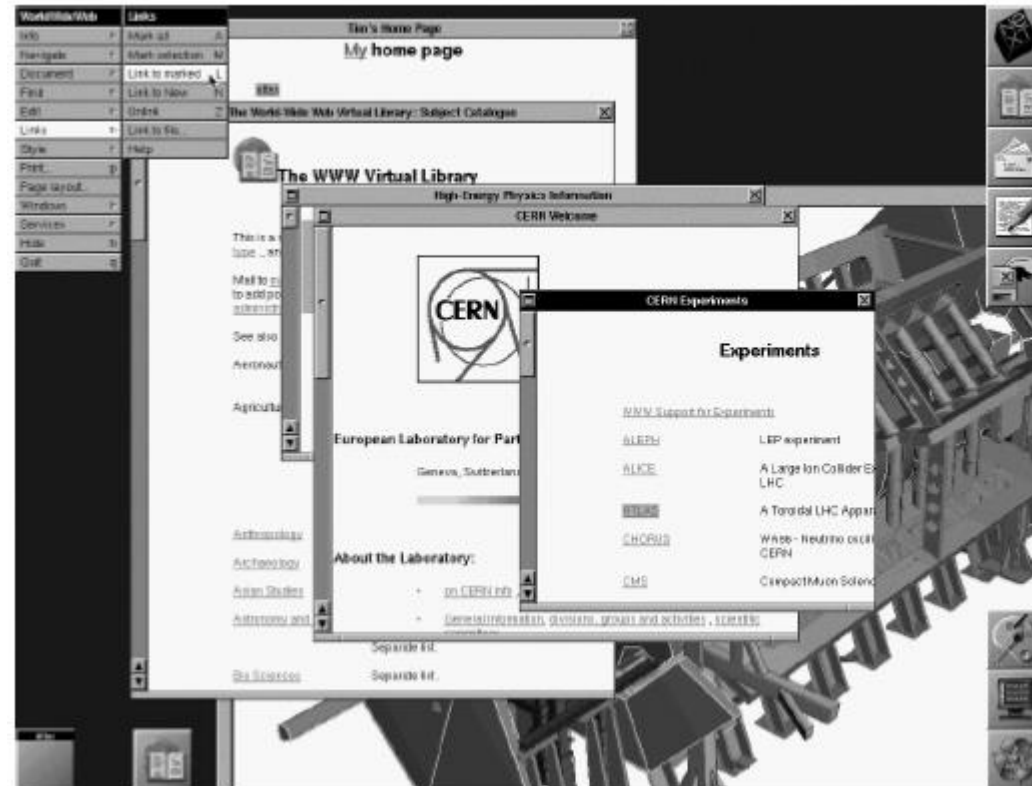
- ✓ 웹은 월드와이드웹(World Wide Web)의 줄임 말로, 전 세계의 정보를 공유할 수 있는 시스템 구조입니다.
- ✓ 인터넷에 연결되어 있는 컴퓨터라면 웹을 통해 손쉽게 전 세계의 지식과 자원을 주고 받을 수 있습니다.
- ✓ 근본적인 웹의 구조를 이해하는 것은 웹 애플리케이션을 설계하고 개발하는데 있어 탄탄한 밑거름이 될 것입니다.



출처 : <http://mobilemarketingmagazine.com/wp-content/uploads/2015/08/Connected-Globe-Internet.jpg>

1.2 웹의 등장

- ✓ 웹의 개념은 1989년 3월 CERN(유럽 입자 물리학 연구소)에서 처음 등장했습니다.
- ✓ 팀 버너스는 멀리 떨어져 있는 연구원들과의 지식 공유를 위해 웹을 고안하게 되었습니다.
- ✓ 월드와이드웹은 최초의 웹 브라우저 이름이며 후에 웹 시스템과 구분하기 위해 넥서스(Nexus)로 이름을 변경합니다.
- ✓ 대표적인 웹의 구성 요소로서 HTML, HTTP 그리고 URL이 세 가지가 제안되었습니다.

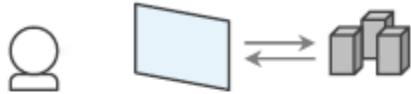


https://www.w3.org/History/1994/WWW/Journals/CACM/screensnap2_24c.gif

1.3 사용자 관점에서 보는 웹

- ✓ 사용자는 웹 브라우저를 실행하고 웹주소를 입력 합니다.
- ✓ 그 후 사용자는 다양한 메뉴와 링크를 통해 원하는 정보를 확인합니다.
- ✓ 이 같은 간단한 웹의 접근을 통해 웹을 구성하는 요소를 파악할 수 있습니다.
- ✓ 웹의 구성은 다음과 같습니다.

1 웹 브라우저



사용자는 웹 브라우저를 통해 웹을 경험합니다.
사용자의 이벤트에 따라 서버에 요청을 보내고, 응답을 받아 다시 사용자에게 보여 줍니다.


3 하이퍼텍스트



웹 페이지는 순서가 없습니다. 사용자의 선택에 따라 변경되고, 다른 문서로 연결됩니다. 이러한 문서 작성 기법을 하이퍼텍스트(Hypertext) 라고 하고, 이 문서를 작성하기 위한 언어를 HTML이라고 부릅니다.



2 URL

 `http://www.nextree.co.kr/p11927/`

정보에 찾아가려면 주소를 입력해야 합니다.
이를 URL(Uniform Resource Locator)라고 합니다.
즉, 웹에 표시되는 모든 자원은 주소를 가지고 있습니다.

4 HTTP



URL의 가장 앞 부분에 HTTP (Hyper Text Transfer Protocol) 가 등장합니다. 웹은 HTTP라는 이름의 하이퍼텍스트 문서 전송 규칙들을 이용해 통신합니다.

1.4 요약

- ✓ 웹은 월드와이드웹(World Wide Web)의 줄임 말로서, 전 세계의 정보를 공유할 수 있는 시스템 구조입니다.
- ✓ 팀 버너스 리 박사가 제안한 웹은 HTML, URL, HTTP로 구성됩니다.
- ✓ 사용자는 웹 브라우저를 통해 웹 시스템을 사용하며, 브라우저는 HTML 문서 등을 이용해 웹 페이지를 표현합니다.
- ✓ HTTP는 요청과 응답에 대한 데이터를 전송하는 규칙을 의미하며, URL을 이용해 요청한 자원을 쉽게 탐색할 수 있습니다.

1 웹 브라우저




사용자는 웹 브라우저를 통해 웹을 경험합니다.
사용자의 이벤트에 따라 서버에 요청을 보내고, 응답을 받아 다시 사용자에게 보여 줍니다.

3 하이퍼텍스트



웹 페이지는 순서가 없습니다. 사용자의 선택에 따라 변경되고, 다른 문서로 연결됩니다. 이러한 문서 작성 기법을 하이퍼텍스트(Hypertext)라고 하고, 이 문서를 작성하기 위한 언어를 HTML이라고 부릅니다.

2 URL



`http://www.nextree.co.kr/p11927/`

정보에 찾아가려면 주소를 입력해야 합니다.
이를 URL(Uniform Resource Locator)라고 합니다.
즉, 웹에 표시되는 모든 자원은 주소를 가지고 있습니다.

4 HTTP



URL의 가장 앞 부분에 HTTP (Hyper Text Transfer Protocol)가 등장합니다. 웹은 HTTP라는 이름의 하이퍼텍스트 문서 전송 규칙들을 이용해 통신합니다.



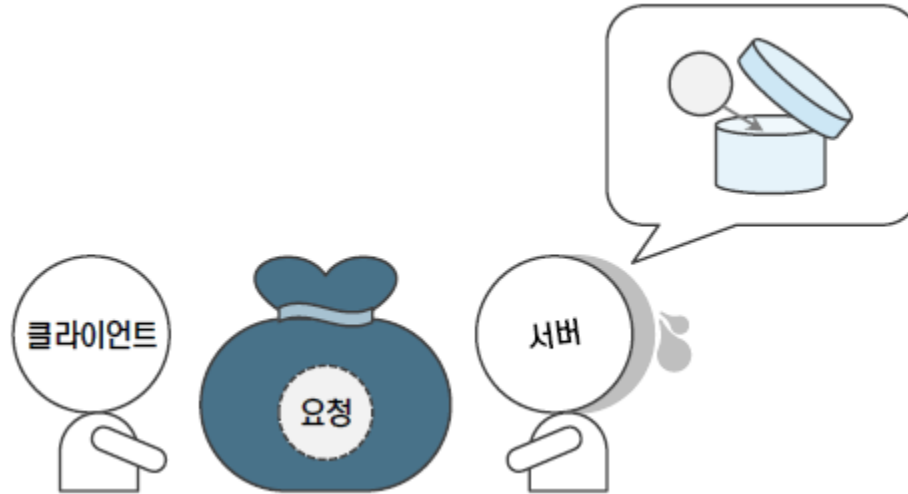


2. HTTP 둘러보기

- 2.1 HTTP 소개
- 2.2 요청에서 응답까지
- 2.3 Stateless HTTP
- 2.4 TCP/IP와 HTTP
- 2.5 요약

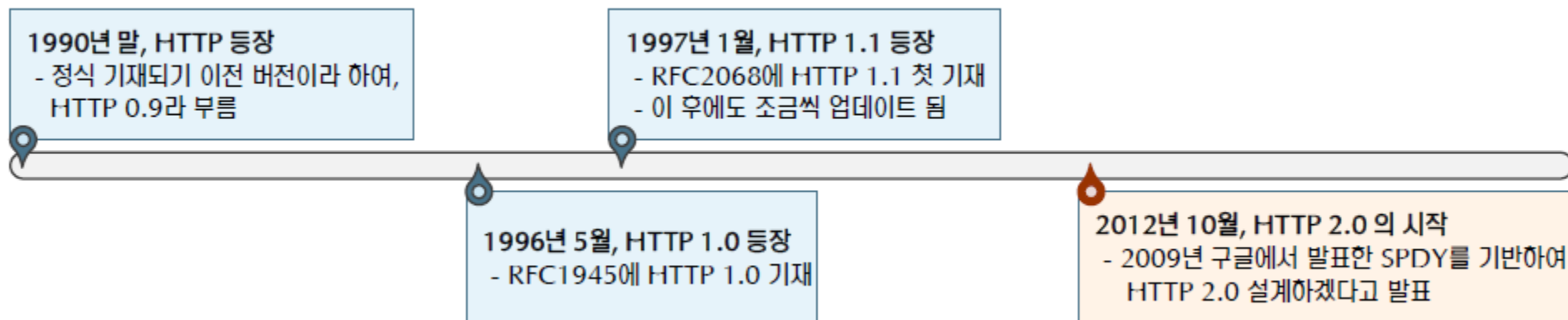
2.1 HTTP 소개(1/2) – 프로토콜

- ✓ HTTP(Hyper-Text Transfer Protocol)은 하이퍼텍스트 문서를 전송하기 위한 통신 규약입니다.
- ✓ 프로토콜(Protocol)이란 클라이언트와 서버간의 통신을 위해서 서로 협의해야 하는 규칙을 의미합니다.
- ✓ 데이터를 송신하는 쪽과 수신하는 쪽 모두 데이터를 주고 받는 방식에 대한 약속을 지켜야 합니다.
- ✓ 웹을 통한 통신에는 다양한 프로토콜이 존재하며, HTTP는 웹의 기반이 되는 프로토콜입니다.



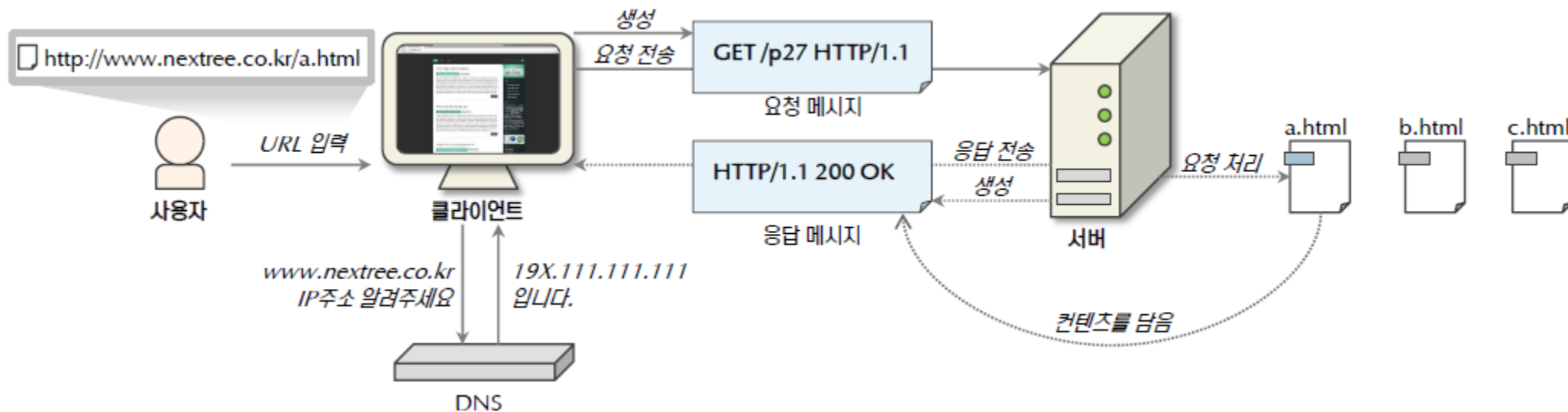
2.2 HTTP 소개(2/2) – HTTP의 역사

- ✓ 1995년 5월, 인터넷 표준 문서인 RFC(Request For Comments)에 HTTP/1.0이 정식으로 기재되었습니다.
- ✓ HTTP는 요청과 응답의 연속입니다. 따라서 이전 요청에 대한 응답이 도착하지 않으면 새로운 요청을 할 수 없습니다.
- ✓ 이와 같은 회전 지연(latency)으로 인해 성능이 저하되는 문제점을 해결하기 위해 HTTP/2.0이 등장 했습니다.
- ✓ HTTP/2.0은 클라이언트가 매번 요청하지 않아도 서버가 필요한 리소스들을 보내는 서버 푸시 방식을 도입했습니다.
- ✓ 하지만 현재에도 HTTP/1.1이 가장 널리 사용되고 있습니다.



2.2 요청에서 응답까지

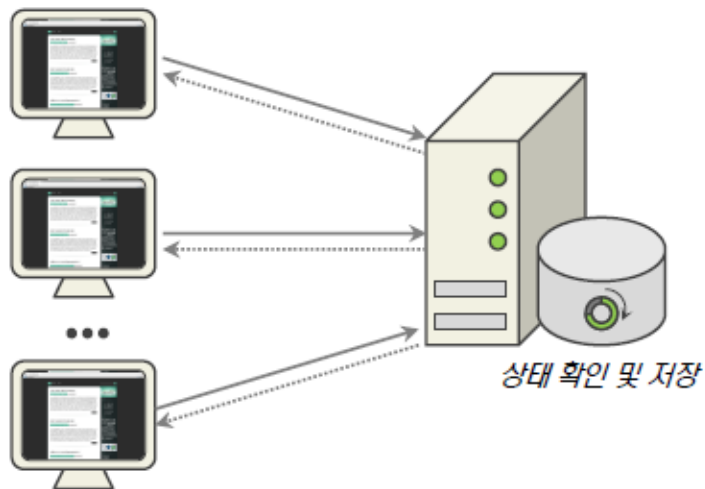
- ✓ HTTP는 클라이언트와 서버간의 커뮤니케이션을 가능하게 합니다.
- ✓ 클라이언트는 사용자가 입력한 주소(URL)를 기반으로 웹 서버에 연결 합니다.
- ✓ HTTP 요청 메시지 규격에 맞게 자원의 주소와 요청 방식 들을 메시지에 담아 서버에 요청을 보냅니다.
- ✓ 서버는 요청에 대한 자원과 처리 결과 코드를 응답 메시지에 담아 반환하고, 커넥션을 닫습니다.



2.3 Stateless HTTP

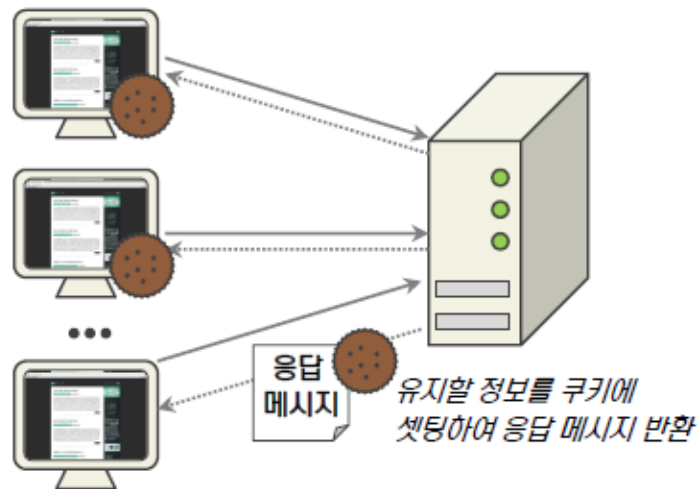
- ✓ HTTP는 기본적으로 상태를 유지하지 않는 Stateless 프로토콜입니다.
- ✓ Stateless 프로토콜은 과거 상태를 신경 쓰지 않기 때문에, 많은 양의 요청을 일관된 방법으로 빠르게 처리합니다.
- ✓ 반면 Stateful 프로토콜은 서버가 많은 클라이언트의 상태를 기억해야 하기 때문에 리소스에 부담을 줍니다.
- ✓ 또한 웹 애플리케이션에 쿠키(Cookie) 기술을 도입하면서, 사용자 식별 및 상태 유지 등 Stateless의 한계를 극복합니다.

상태를 유지하는(Stateful) 프로토콜



Stateful 프로토콜은 매번 요청이 일어날 때마다, 서버가 수많은 클라이언트의 상태를 유지하기 위해 상태 정보를 동기화하고 저장 하는 등의 단계를 거쳐야 합니다. 따라서 상태 유지는 서버의 CPU나 메모리에 부담을 주게 됩니다.

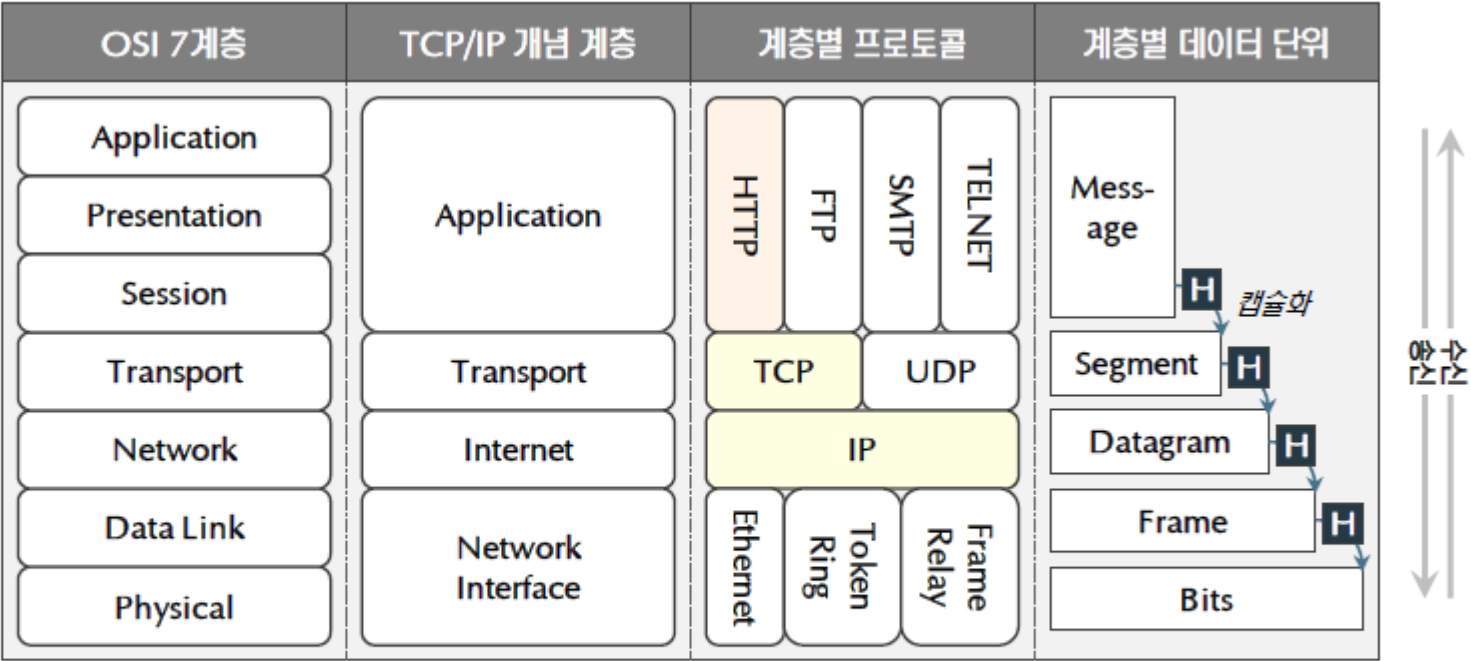
상태를 유지하지 않는(Stateless) 프로토콜



Stateless 프로토콜은 상태 유지를 위한 과정을 생략하기 때문에, 빠르게 요청을 처리합니다. 하지만 클라이언트를 식별하거나 과거 상태가 필요한 경우가 있습니다. 이 때 쿠키를 활용합니다. 서버가 기억할 정보를 쿠키에 담아 보내면 클라이언트는 이 쿠키를 보존합니다. 이 후 요청은 쿠키를 함께 보내기 때문에, 이를 통해 서버는 클라이언트의 상태를 알 수 있습니다.

2.4 TCP/IP와 HTTP (1/2)

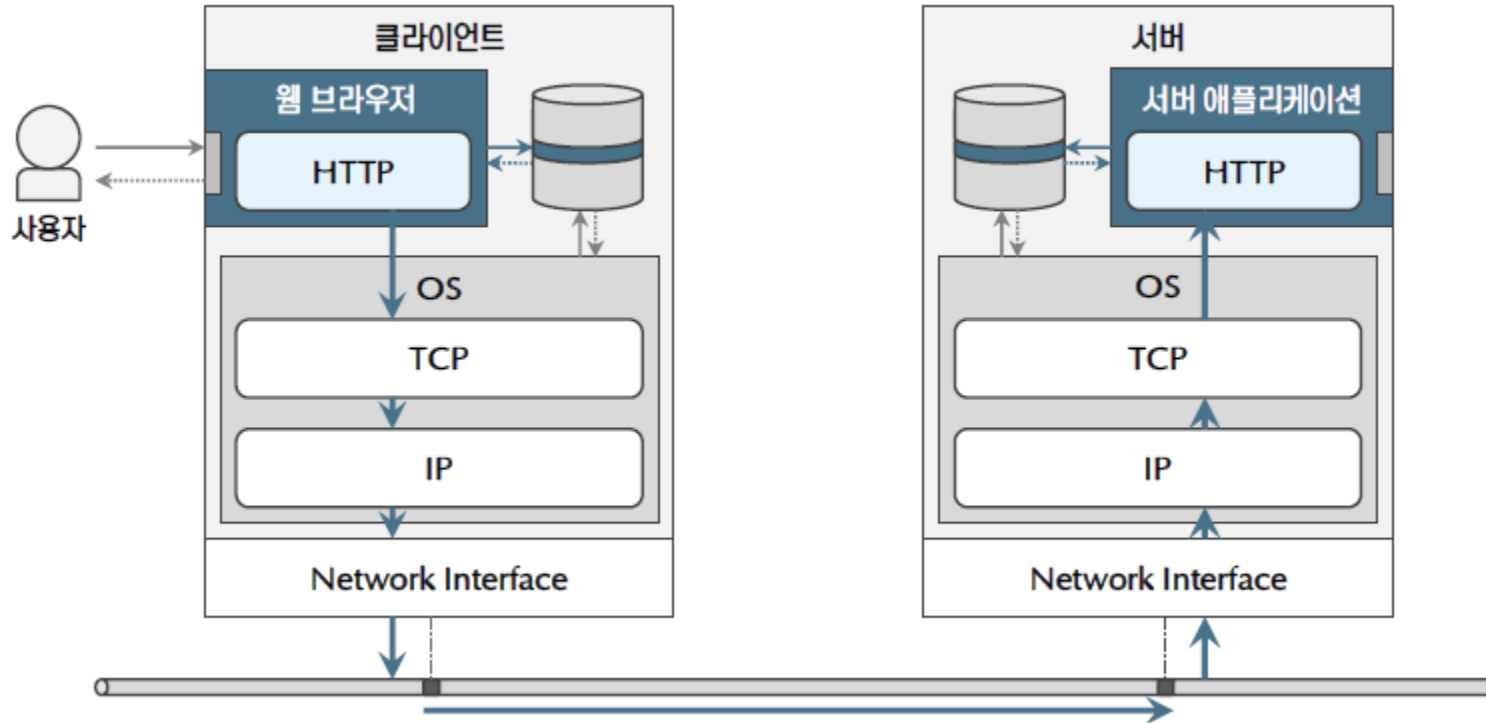
- ✓ TCP/IP(Transmission Control Protocol/Internetworking Protocol)는 컴퓨터의 네트워크 통신 규약입니다.
- ✓ TCP/IP는 총 다섯 계층으로 구성되며 각 계층별 네트워크마다 사용되는 프로토콜과 데이터 단위가 다릅니다.
- ✓ 각 계층에서 데이터를 전송할 때는 헤더로 감싸야 하며, 이 과정을 캡슐화(Encapsulation)이라고 합니다.
- ✓ TCP는 신뢰를 기반합니다. 데이터를 보내기 전 후로 상대의 상태를 확인하고 만약 실패했다면 재전송합니다.
- ✓ HTTP는 애플리케이션 계층에서 담당하는 부분만 고려하면 정확하게 메시지를 전송할 수 있습니다.



- OSI(Open System Interconnection, 개방형 시스템 상호접속)는 ISO(국제표준화기구)가 작성한 컴퓨터 통신절차에 관한 국제표준규격입니다.
- TCP/IP의 네트워크 연결 계층에서는 특정 프로토콜을 지정하기 보다는 모든 표준과 기술에 정의된 프로토콜을 지원합니다. 이더넷과 토큰링은 LAN, 프레임 릴레이는 WAN에 기반합니다.
- 애플리케이션 계층에는 FTP(File Transfer Protocol), SMTP(Simple Mail Transfer Protocol), POP(Post Office Protocol), Telnet(Tele Network) 등의 프로토콜이 있습니다.

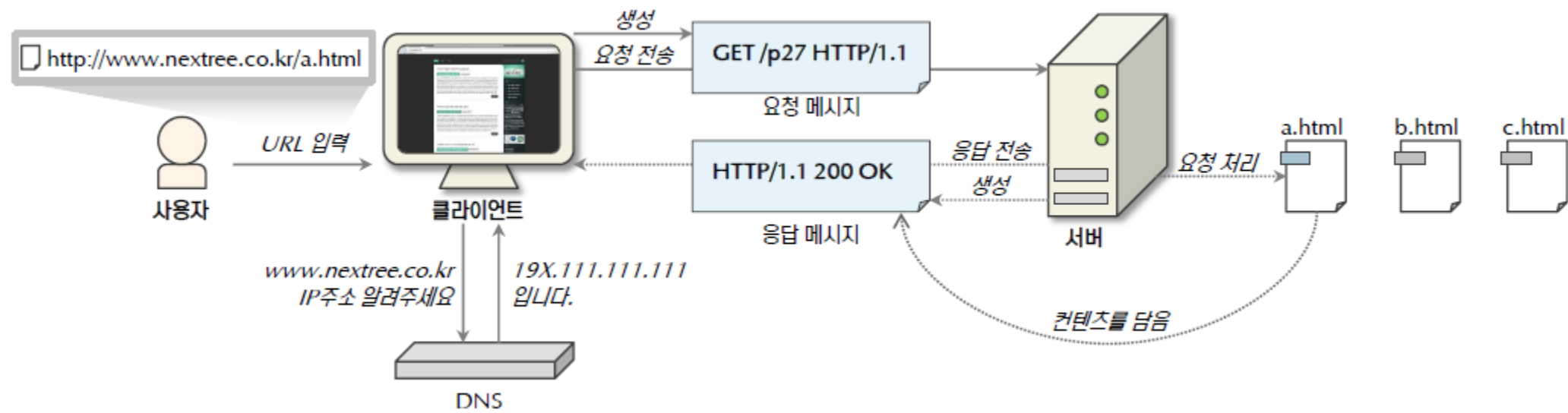
2.4 TCP/IP와 HTTP (2/2)

- ✓ 만약, TCP/IP가 계층 구조가 아니라면 어떤 네트워크 기술의 사양이 변경 되더라도 계층 전체가 변경되어야 합니다.
- ✓ 계층화하면 각 계층별로 독립적인 진화가 가능하며, 자신이 담당하는 기능만 수행하면 됩니다.
- ✓ HTTP는 애플리케이션에서 구현합니다. 클라이언트 역할을 하는 웹 브라우저와 서버 프로그램이 HTTP를 구현하고 있습니다.
- ✓ 애플리케이션은 OS에서 구현한 TCP/IP 서비스를 이용하여 통신하며, OS가 관리하는 저장소에서 자원을 관리합니다.



2.8 요약

- ✓ HTTP(Hyper-Text Transfer Protocol)은 하이퍼텍스트 문서를 전송하기 위한 통신 규약입니다.
- ✓ 클라이언트와 서버간의 통신을 위해 URL, 요청 메소드 등을 담은 요청 메시지와 상태코드를 담은 응답메시지를 구성합니다.
- ✓ HTTP는 기본적으로 상태를 유지하지 않는 Stateless 프로토콜입니다. 쿠키를 이용하면 상태를 유지할 수 있습니다.
- ✓ HTTP는 애플리케이션 계층 프로토콜로서 TCP와 통신합니다. TCP 덕분에 정확하게 데이터를 전송할 수 있습니다.



토론

- ✓ 질문과 대답
- ✓ 토론

