2026/10/16, 2026 보안 기초 세미나

TCP/IP 프로토콜

- 22장 월드 와이드 웹과 HTTP, 23장 SMTP, POP, IMAP, MIME 개요 -

이 성 현(seonghyeon@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

- 보충
- 월드 와이드 웹(WWW, World Wide Web)
- HTTP (Hypertext Transfer Protocol)
- 전자우편

목 차

- 보충
- 월드 와이드 웹(WWW, World Wide Web)
- HTTP (Hypertext Transfer Protocol)
- 전자우편

보충

- 월드 와이드 웹 (1/2)
  - 웹 서버 (1/2)
    - 정의
      - 웹 클라이언트의 요청을 받아 웹 문서나 자원을 제공하는 프로그램
    - 특징
      - 웹 페이지를 저장하며, 요청에 따라 클라이언트에게 문서를 응답으로 반환함
      - 빠른 응답을 위해 요청된 문서를 메모리의 캐시에 저장하고 필요할 때 캐시의 문서를 응답에 활용할 수 있음
      - 다중 스레드, 다중 프로세스 등을 이용하여 한 번에 하나 이상의 요청에 응답할 수 있음

보충

- 월드 와이드 웹 (2/2)
  - 웹 서버 (2/2)
    - 예시
      - Apache
        - 웹 서버를 구현해주는 프로그램이며, 모듈을 통해 높은 확장성을 가짐
        - 동적 문서는 주로 모듈이나 외부 프로그램 연동을 통해 생성함
      - Nginx
        - 이벤트 기반 구조를 사용하는 웹 서버 프로그램이며, 다수의 동시 연결 처리가 용이함
        - 다수의 사용자 요청을 처리하고, 요청을 다른 서버로 전달하는 데에 적합함
      - Node.js
        - JavaScript를 서버에서 실행할 수 있는 런타임 환경이며, HTTP 서버를 직접 구현할 수 있음
        - 높은 개발 자유도로 동적 문서 생성이 용이함

목 차

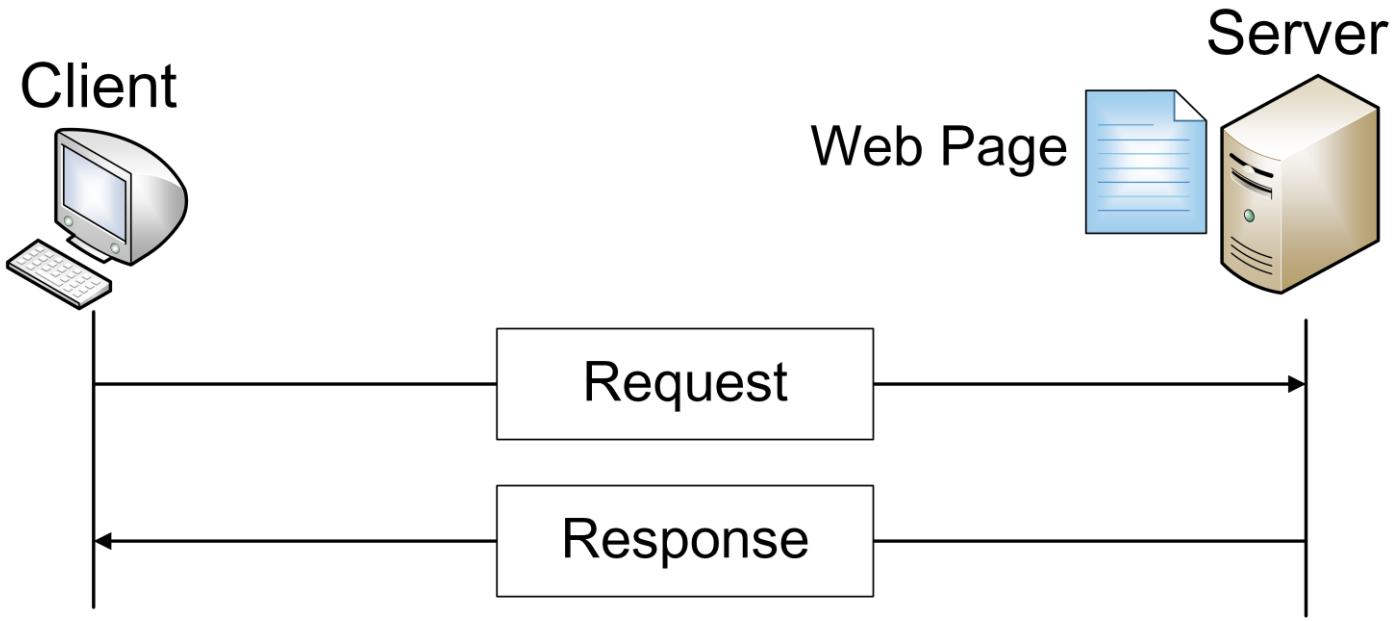
- 보충
- 월드 와이드 웹(WWW, World Wide Web)
- HTTP (Hypertext Transfer Protocol)
- 전자우편

월드 와이드 웹

- 정의
  - 웹 클라이언트와 웹 서버가 인터넷을 통해 웹 문서와 자원을 주고받는 시스템
- 개요
  - 오늘날 WWW는 분산 클라이언트-서버 서비스로서 동작하며, 클라이언트는 서버를 통해 서비스를 제공받음
  - 제공되는 서비스는 여러 사이트에 분산되어 있으며, 각 사이트는 웹 페이지(Web Page)라고 불리는 하나 이상의 문서들을 유지함
  - 하이퍼텍스트(Hypertext)는 텍스트 링크를 통해 다른 문서를 참조할 수 있도록 함
  - 하이퍼미디어(Hypermedia)는 이미지, 음성, 영상 등 다양한 미디어에 링크를 적용함

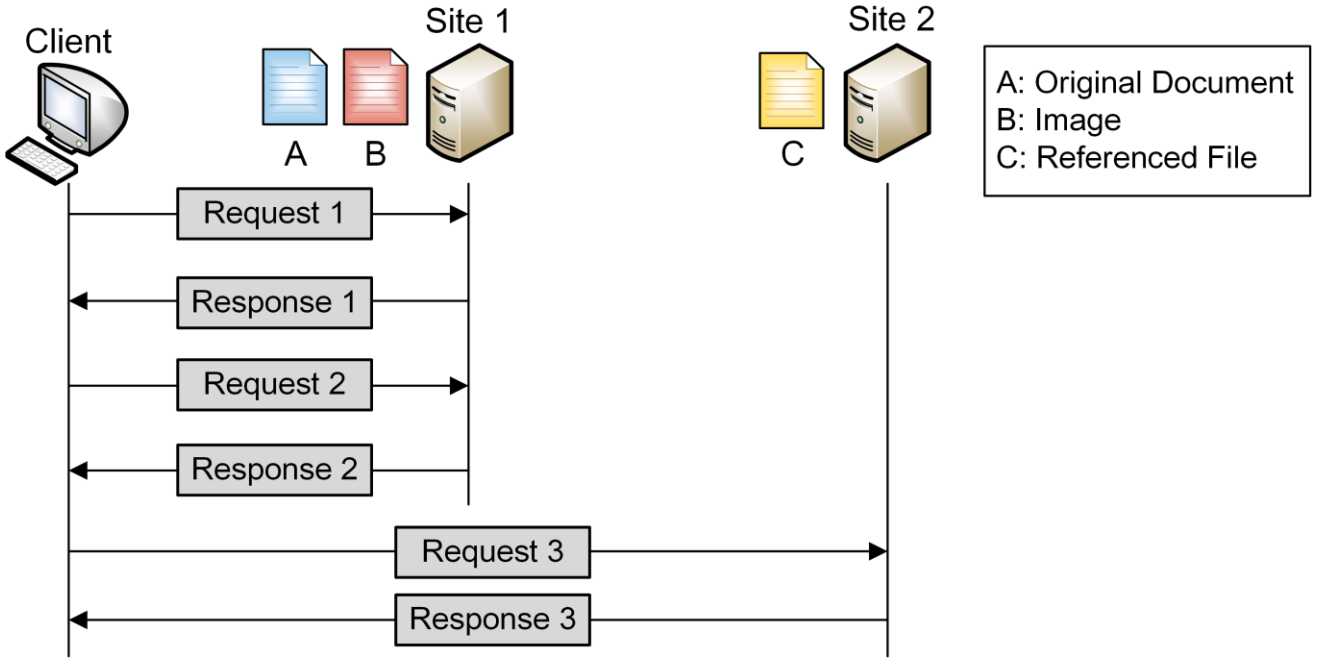
월드 와이드 웹

- 단일 웹 페이지
  - 사진 등이 분리된 파일로 저장되지 않은 페이지의 경우, 한 번의 요청과 응답으로 문서를 제공받을 수 있음



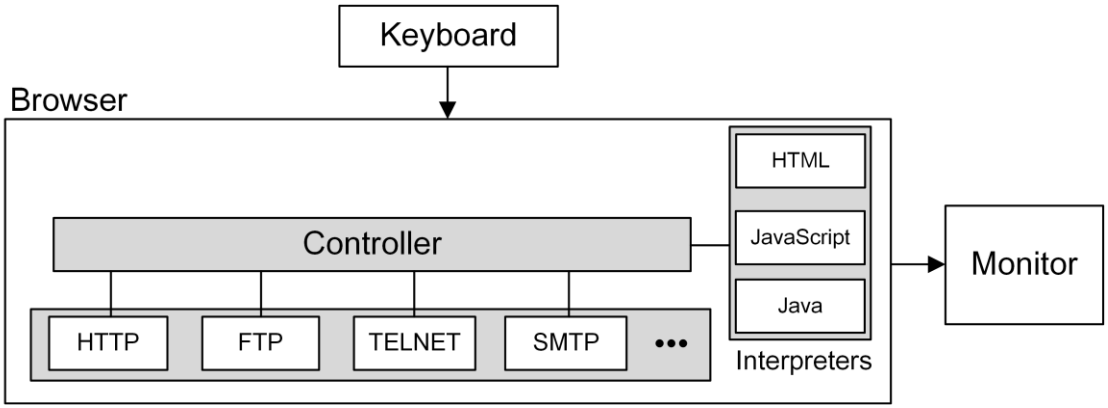
월드 와이드 웹

- 복합 웹 페이지
  - 웹 페이지에 다른 문서에 대한 참조가 있는 경우, 전체 문서를 확인하기 위해서는 참조된 문서에 대한 요청과 응답이 필요함
  - 필요에 따라 각 웹 페이지를 독립적으로 요청하고 응답 받을 수 있음



월드 와이드 웹

- 웹 클라이언트(브라우저) (1/2)
  - 정의
    - 웹 서버에 HTTP 요청을 보내고 응답을 받는 프로그램
  - 구성
    - 제어기(Controller)
      - 키보드, 마우스로부터 입력을 받아 문서에 접근함
    - 클라이언트 프로토콜
      - 서버와 통신을 할 때 사용함
    - 해석기(Interpreter)
      - 문서나 데이터를 실제로 해석하여 사용자에게 표시함



월드 와이드 웹

- 웹 클라이언트(브라우저) (2/2)
  - 예시
    - Google Chrome
      - Chromium 기반으로 높은 웹 표준 지원과 확장 기능을 제공하는 브라우저
    - Mozilla Firefox
      - 오픈소스 기반의 독립적인 웹 브라우저
    - Microsoft Edge
      - Windows와의 통합성이 높은 Chromium 기반 브라우저
    - Safari
      - Apple 생태계에 최적화된 웹 브라우저

월드 와이드 웹

- 웹 서버 (1/2)
  - 정의
    - 웹 클라이언트의 요청을 받아 웹 문서나 자원을 제공하는 프로그램
  - 특징
    - 웹 페이지를 저장하며, 요청에 따라 클라이언트에게 문서를 응답으로 반환함
    - 빠른 응답을 위해 요청된 문서를 메모리의 캐시에 저장하고 필요할 때 캐시의 문서를 응답에 활용할 수 있음
    - 다중 스레드, 다중 프로세스 등을 이용하여 한 번에 하나 이상의 요청에 응답할 수 있음

월드 와이드 웹

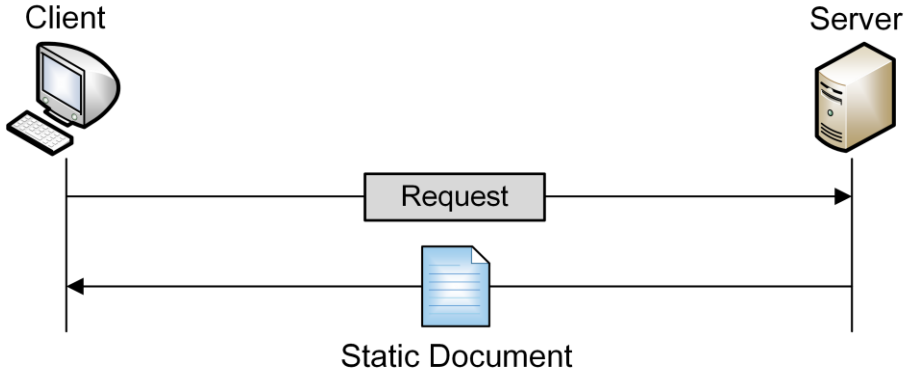
- 웹 서버 (2/2)
  - 예시
    - Apache
      - 웹 서버를 구현해주는 프로그램이며, 모듈을 통해 높은 확장성을 가짐
      - 동적 문서는 주로 모듈이나 외부 프로그램 연동을 통해 생성함
    - Nginx
      - 이벤트 기반 구조를 사용하는 웹 서버 프로그램이며, 다수의 동시 연결 처리가 용이함
      - 다수의 사용자 요청을 처리하고, 요청을 다른 서버로 전달하는 데에 적합함
    - Node.js
      - JavaScript를 서버에서 실행할 수 있는 런타임 환경이며, HTTP 서버를 직접 구현할 수 있음
      - 높은 개발 자유도로 동적 문서 생성이 용이함

월드 와이드 웹

- URL (Uniform Resource Locator)
  - 개요
    - 클라이언트는 웹 페이지에 접근하기 위해 해당 페이지의 주소를 필요로 하며, 이를 “Protocol://Host:Port/Path”와 같은 형태로 표현할 수 있음
  - Protocol
    - HTTP, FTP, TELNET 등 문서를 불러올 때 사용되는 프로토콜
  - Host
    - 해당 문서가 위치하는 컴퓨터의 도메인 네임
  - Port
    - 서버가 사용하는 포트 번호이며, 생략 가능함
  - Path
    - 서버 내에서 해당 문서가 위치하는 경로

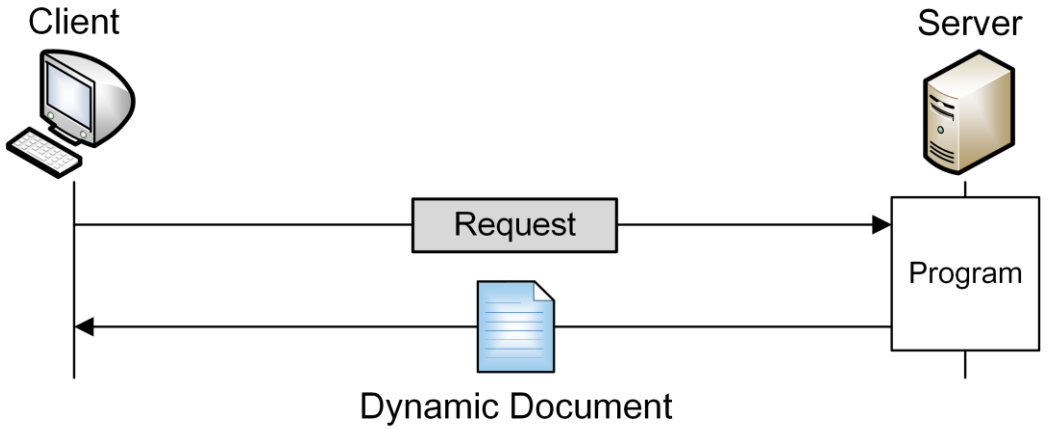
월드 와이드 웹

- 웹 문서 (1/6)
  - 개요
    - WWW에서 문서는 문서의 내용이 결정되는 시점을 기준으로 정적(Static), 동적(Dynamic), 액티브(Active) 문서로 구분됨
  - 정적 문서(Static Document)
    - 서버에서 생성되어 저장되며, 문서의 내용은 생성될 때 결정됨
    - 클라이언트가 문서에 접근하는 경우, 문서의 복사본이 전송됨
    - HTML (Hypertext Markup Language), XML (Extensible Markup Language) 등을 통해 작성됨



월드 와이드 웹

- 웹 문서 (2/6)
  - 동적 문서(Dynamic Document) (1/3)
    - CGI (Common Gateway Interface) (1/2)
      - 정의
        - 웹 서버와 외부 프로그램 사이의 데이터 전달 및 실행 방식을 정의한 표준 인터페이스
      - 특징
        - CGI를 따라 C, C++, Perl 등의 언어로 동적 문서를 생성하는 프로그램을 작성하고, 해당 프로그램으로 요청에 따른 문서를 생성할 수 있음

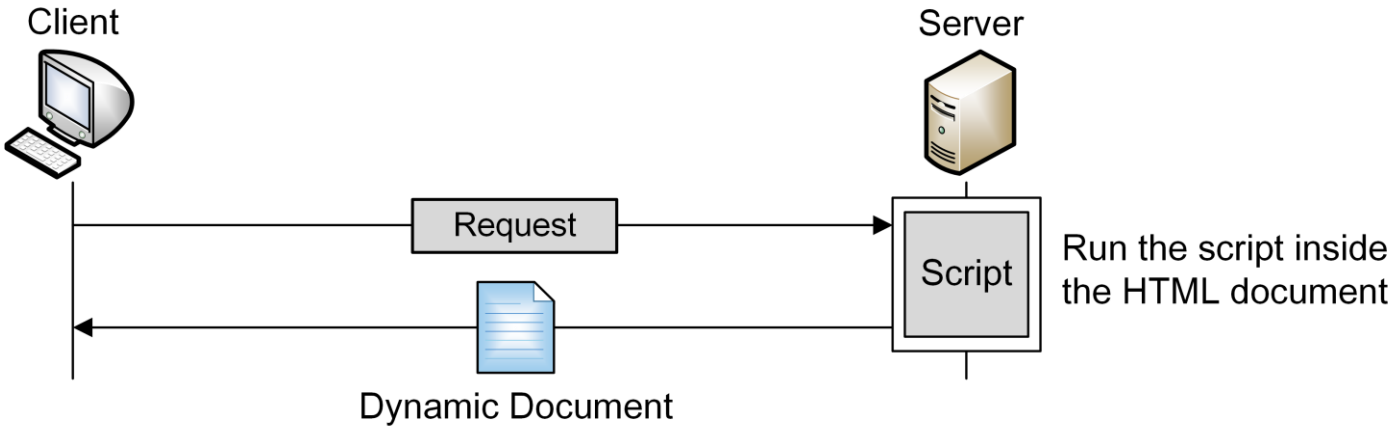


월드 와이드 웹

- 웹 문서 (3/6)
  - 동적 문서(Dynamic Document) (2/3)
    - CGI (Common Gateway Interface) (2/2)
      - 입력
        - 문서 생성에 간단한 정보가 입력으로서 필요한 경우, URL에 '?'를 포함하여 입력 값을 함께 전송할 수 있음
          - e.g., "http://www.example/cgi-bin/prog.pl?23"를 통해 23을 전송할 수 있음
        - 입력 값이 많은 경우에는 서버에게 양식(Form)을 전송하겠다는 요청을 한 후, 입력 데이터로 양식을 채워 전송하면 문서 생성에 활용될 수 있음
      - 출력
        - CGI 프로그램에 의해 HTML, 텍스트, 그래픽, 이진 데이터, 상태 코드, 캐시 제어 정보, 문서 이동 정보 등의 결과가 출력됨
        - 출력 문서 종류를 CGI 헤더에 작성하고 빈 줄로 구분하여 출력 결과와 함께 클라이언트에게 전송하면 브라우저는 헤더를 통해 문서를 해석함

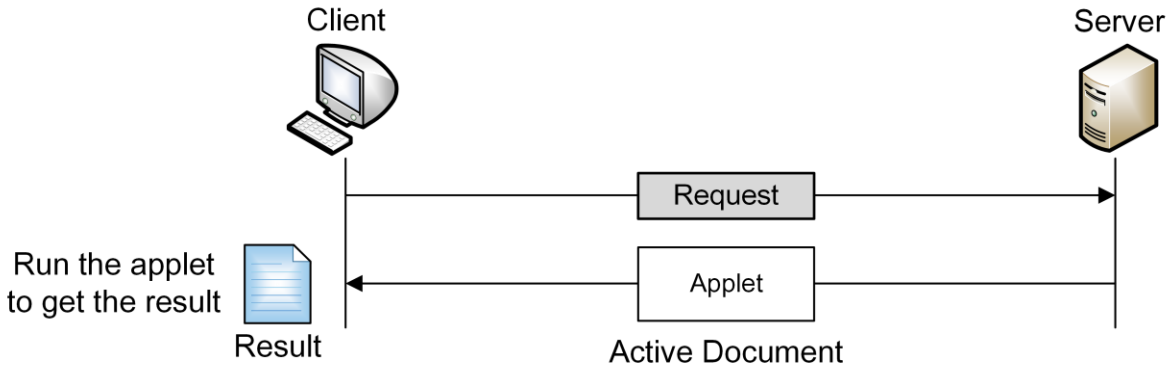
월드 와이드 웹

- 웹 문서 (4/6)
  - 동적 문서(Dynamic Document) (3/3)
    - 스크립트(Script)
      - 정의
        - 웹 페이지나 서버에서 특정 동작을 수행하거나 동적 내용을 생성하기 위해 작성하는 프로그램 코드
      - 특징
        - PHP (Hypertext Preprocessor), JSP (Java Server Pages) 등을 통해 생성한 스크립트를 HTML 문서 내에 포함하여 일부만 동적으로 변경되는 문서를 생성하는 데에 사용할 수 있음



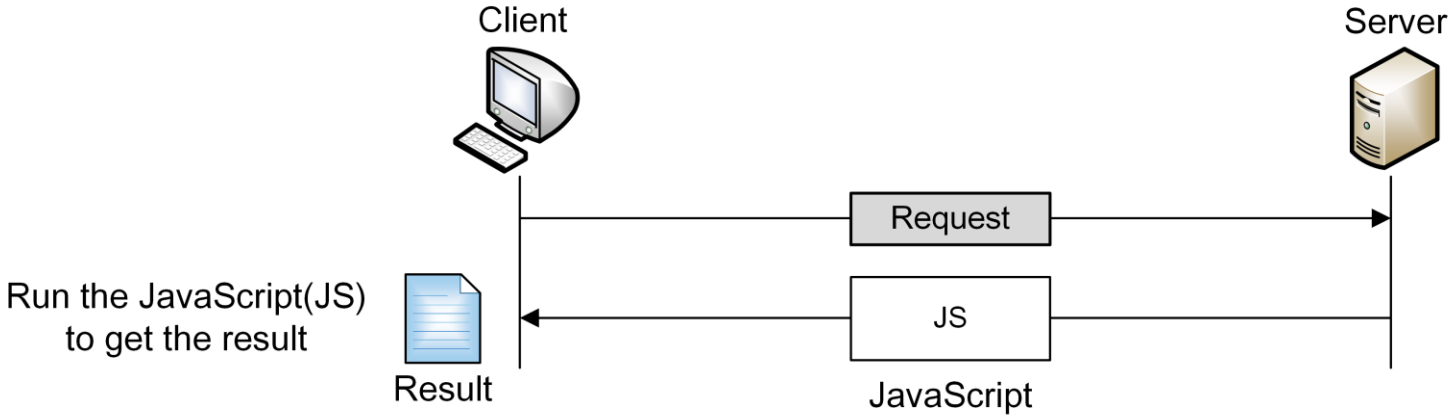
월드 와이드 웹

- 웹 문서 (5/6)
  - 액티브 문서(Active Document) (1/2)
    - 자바 애플릿(Java Applet)
      - 정의
        - 객체지향 프로그래밍 언어인 Java로 작성되어 웹 페이지 안에서 실행되는 소형 프로그램
      - 특징
        - 문서를 2진수 형태의 바이트코드(Bytecode)로 전송하면 브라우저가 해당 애플릿의 인스턴스를 생성하고 수행함
        - URL을 통해 직접 요청하거나 HTML에 포함된 주소에 의해 요청됨



월드 와이드 웹

- 웹 문서 (6/6)
  - 액티브 문서(Active Document) (2/2)
    - 자바스크립트(JavaScript)
      - 정의
        - 웹 문서에 포함되어 클라이언트의 브라우저에서 실행되며, 문서의 내용이나 동작을 동적으로 변경하는 스크립트 언어
      - 특징
        - 문서를 텍스트 형태의 소스 코드(Source code)로 전송하면 브라우저가 해당 소스 코드를 해석하고 실행함
        - HTML 내 코드로서 작성되었거나 HTML에 포함된 주소에 의해 서버에 요청됨



목 차

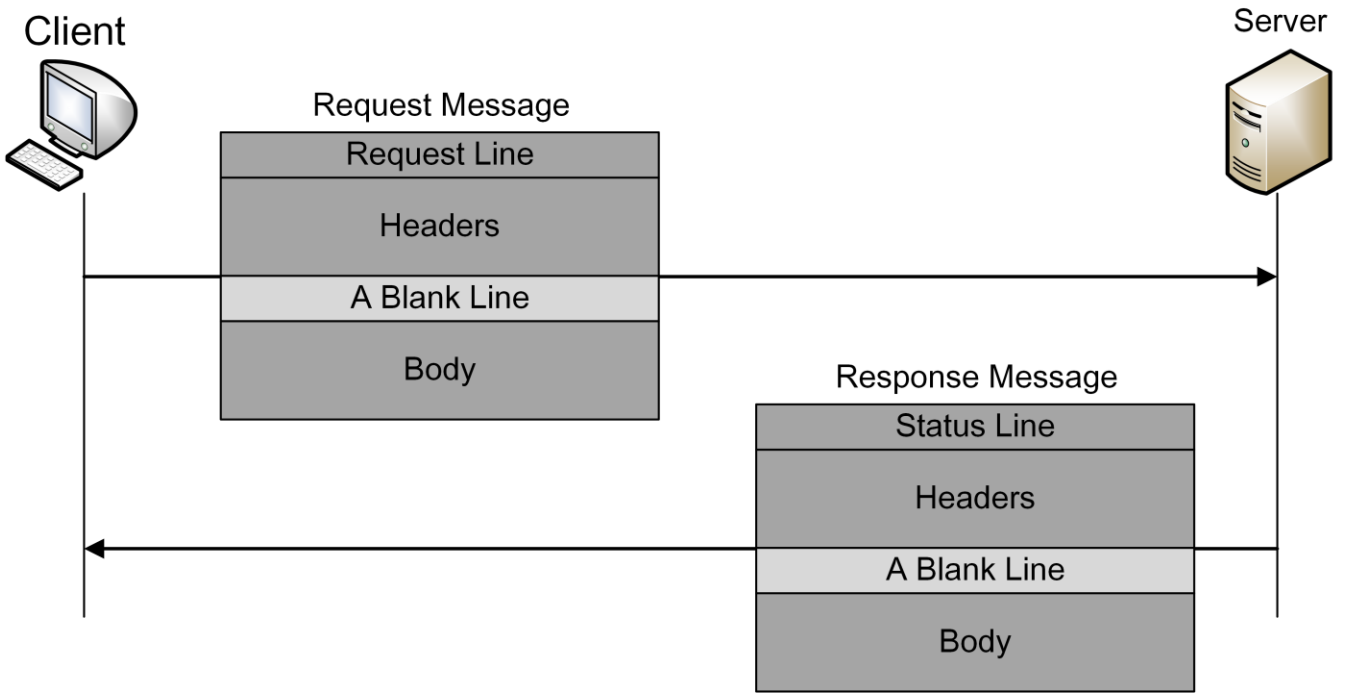
- 월드 와이드 웹(WWW, World Wide Web)
- HTTP (Hypertext Transfer Protocol)
- 전자우편

HTTP

- 정의
  - 월드 와이드 웹에서 클라이언트가 서버에 자원을 요청하고, 서버가 그 결과를 응답하도록 하는 요청-응답 기반 프로토콜
- 특징
  - TCP (Transmission Control Protocol)를 이용하여 파일을 전송하며, 단일 TCP 연결을 사용함
  - HTTP 메시지는 HTTP 서버와 HTTP 클라이언트에 의해 읽히고 해석됨
  - 서버로의 명령은 요청 메시지에, 요청에 따른 파일 내용 및 정보는 응답 메시지를 통해 전송됨
  - Well-known 포트로 80을 이용하여 TCP 서비스를 이용함

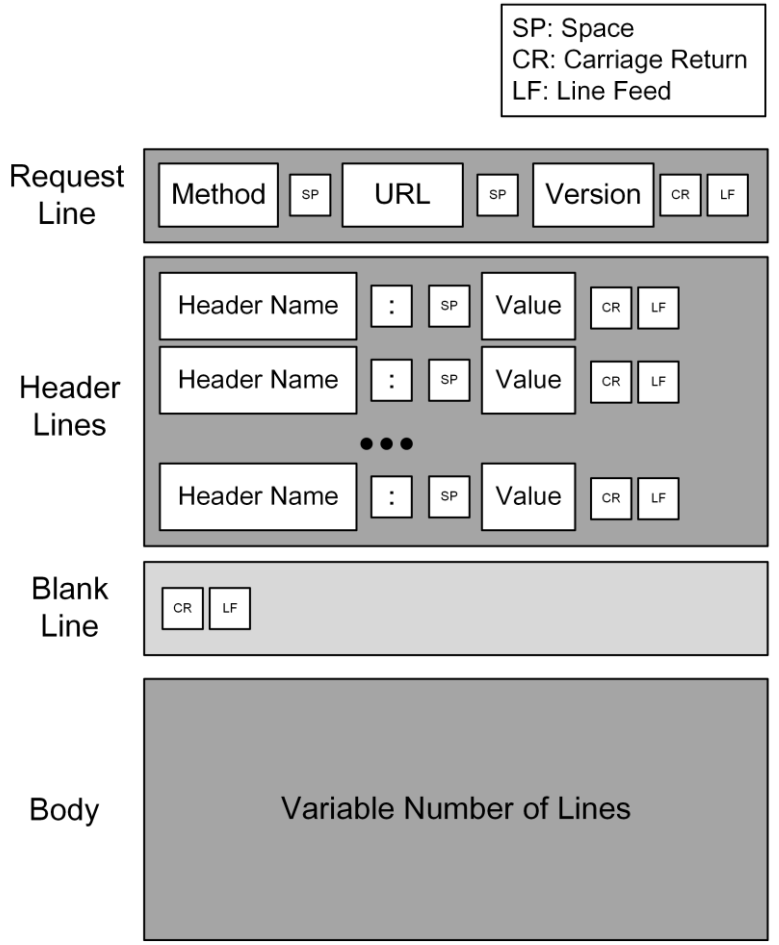
HTTP

- 트랜잭션 (1/11)
  - HTTP 자체는 상태가 존재하지 않는(Stateless) 프로토콜임
  - 요청 메시지를 통해 트랜잭션을 시작하고, 응답 메시지를 통해 요청에 대한 응답을 전송함



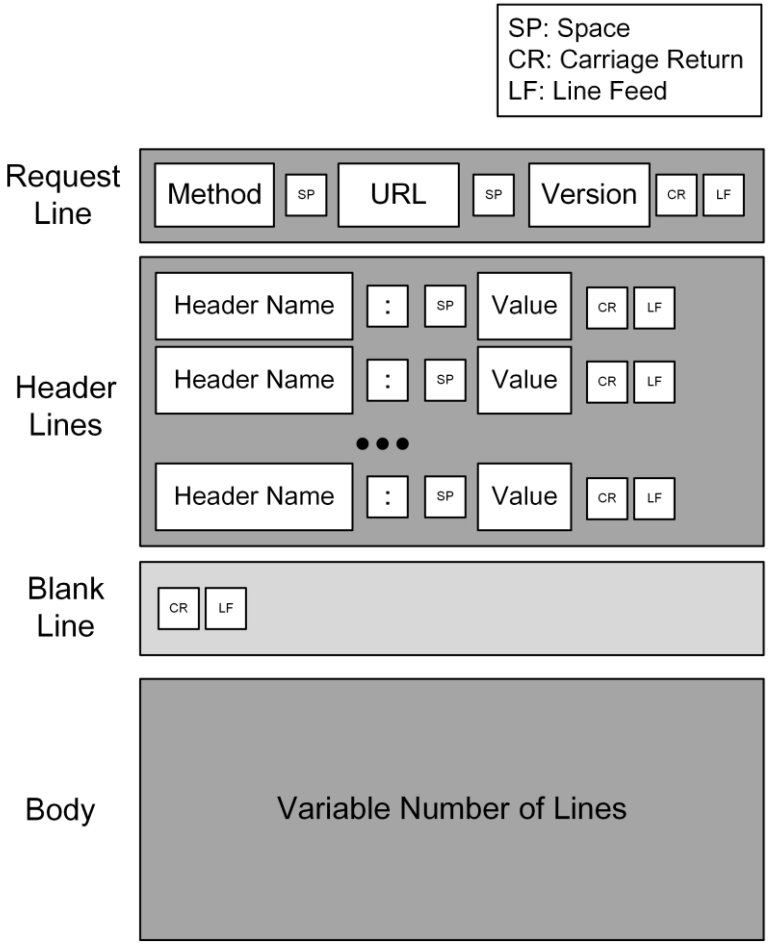
- 트랜잭션 (2/11)
  - 요청 (1/3)
    - 요청 라인(Request Line)
      - 문자 구분 기호에 의해 Method, URL, Version 필드로 구분됨
      - CR (Carriage Return), LF (Line Feed) 문자를 통해 라인을 종료함
      - Method는 요청 유형, URL은 문서 주소, Version은 HTTP 버전을 나타냄

Method	Action
GET	서버에 문서를 요청함
HEAD	문서 자체는 제외하고, 문서에 대한 정보만 요청함
POST	클라이언트에서 서버로 일부 정보를 전송함
PUT	클라이언트가 전송한 데이터로 서버의 자원을 생성하거나 대체함
TRACE	수신된 요청을 그대로 되돌려 보냄
CONNECT	대상 서버와의 터널 연결을 설정함
DELETE	대상 자원 삭제를 요청함
OPTIONS	사용 가능한 옵션을 조회함

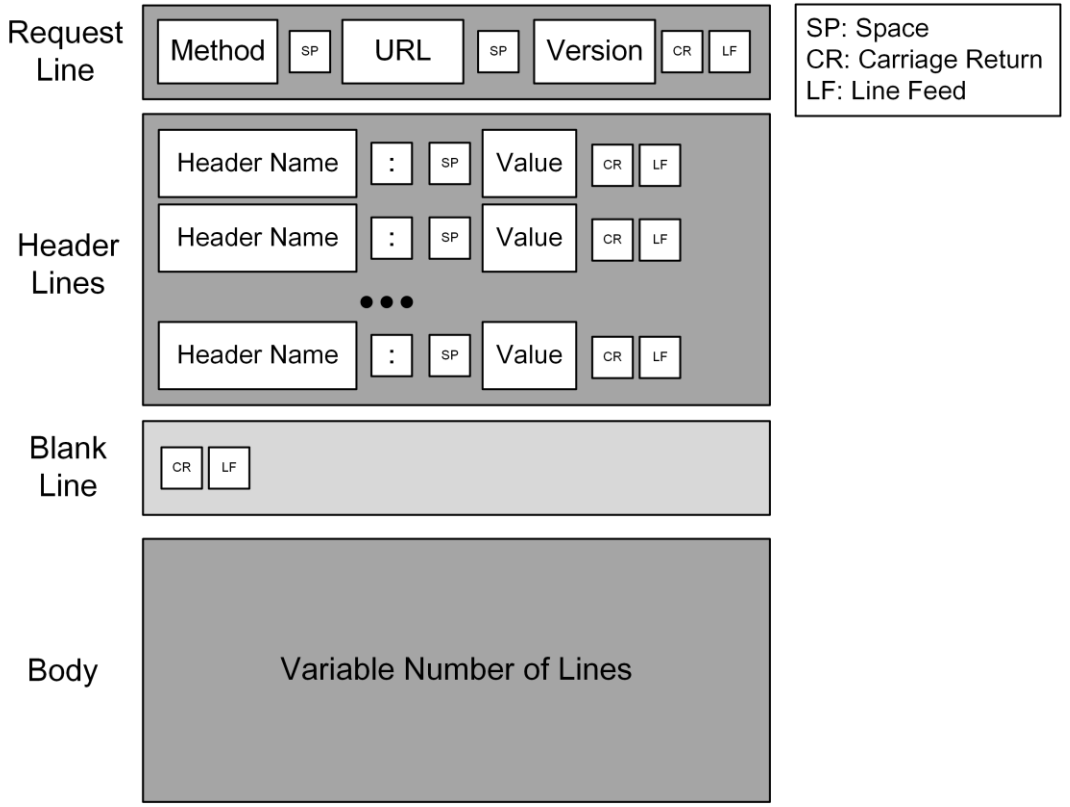


- 트랜잭션 (3/11)
  - 요청 (2/3)
    - 헤더 라인(Header Line)
      - 클라이언트로부터 서버로의 추가 정보를 전송함
      - 헤더 이름, 콜론, 공백, 헤더 값으로 구성됨

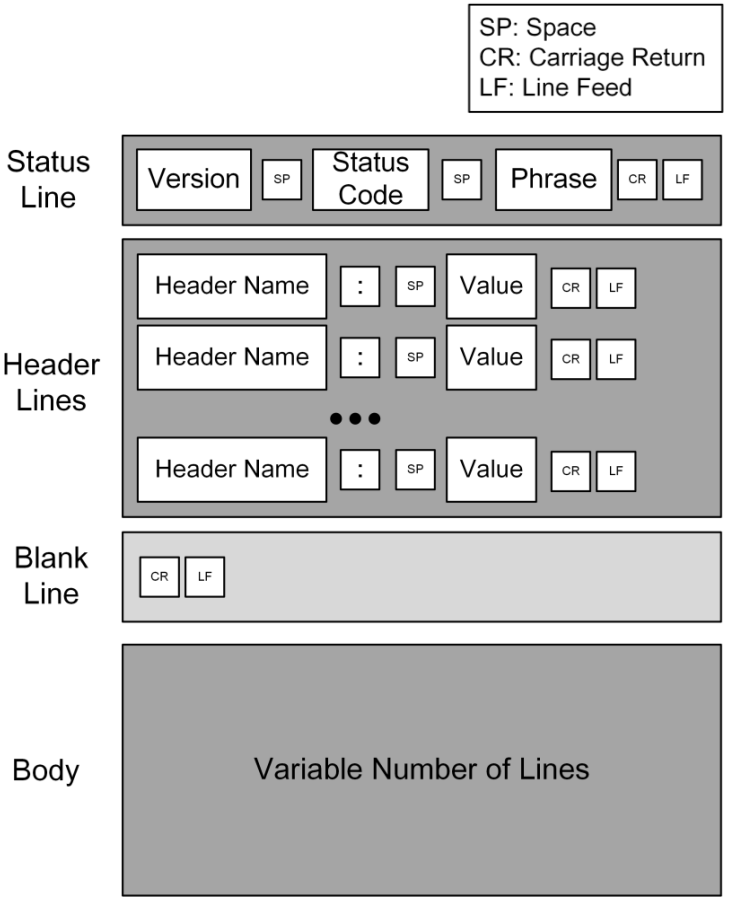
Header	Description
User-agent	클라이언트 프로그램을 식별함
Accept	클라이언트가 받아들일 수 있는 미디어 형식을 나타냄
Accept-charset	클라이언트가 처리할 수 있는 문자 집합을 나타냄
Accept-encoding	클라이언트가 처리할 수 있는 인코딩 방식을 나타냄
Accept-language	클라이언트가 받아들일 수 있는 언어를 나타냄
Authorization	클라이언트가 가진 권한을 나타냄
Host	요청 대상 서버의 호스트 이름과 포트 번호를 나타냄
Date	현재 날짜를 나타냄
Upgrade	선호하는 통신 프로토콜을 지정함
Cookie	쿠키를 서버로 반환함
If-Modified-Since	지정한 시각 이후 자원이 수정된 경우에만 해당 자원을 보내도록 요청함



- 트랜잭션 (4/11)
  - 요청 (3/3)
    - 몸체(Body)
      - 입력 값, 업로드 데이터 등 요청에 필요한 정보가 포함됨



- 트랜잭션 (5/11)
  - 응답 (1/4)
    - 상태 라인(Status Line) (1/2)
      - 문자 구분 기호에 의해 Version, Status Code, Phrase 필드로 구분됨
      - CR (Carriage Return), LF (Line Feed) 문자를 통해 라인을 종료함
      - Version은 HTTP 버전, Status Code는 요청 상태, Phrase는 상태에 대한 설명을 나타냄



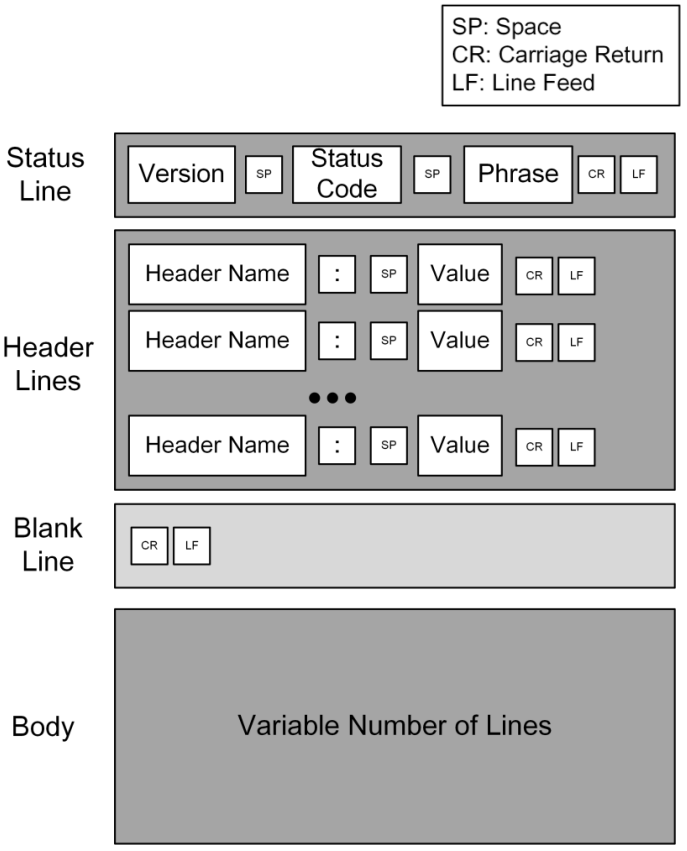
분류	Status Code	Status Phrase	Description
Informational	100	Continue	요청의 초기 부분을 수신했으며, 계속 진행하라는 의미
	101	Switching	서버가 프로토콜 전환 요청을 수락하고 전환 중임을 의미
Success	200	OK	요청이 성공적으로 처리됨
	201	Created	새로운 URL/자원이 생성됨
	202	Accepted	요청이 수락되었지만 즉시 처리되지는 않았음
	204	No content	응답 본문에 내용이 없음
	206	Partial content	요청한 자원의 일부만 성공적으로 반환함

• 트랜잭션 (6/11)

• 응답 (2/4)

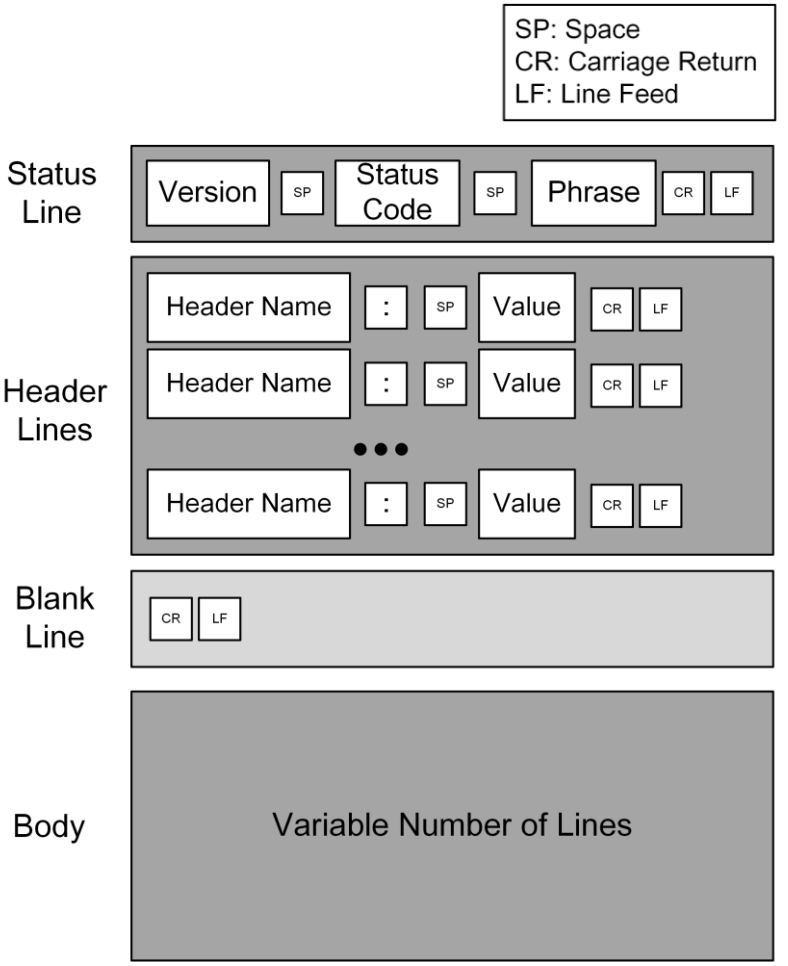
• 상태 라인(Status Line) (2/2)

분류	Status Code	Status Phrase	Description
Redirection	301	Moved permanently	요청한 URL이 더 이상 기존 위치에서 사용되지 않으며 영구적으로 이동됨
	302	Moved temporarily	요청한 URL이 일시적으로 다른 위치로 이동됨
	304	Not modified	문서가 수정되지 않았음
Client Error	400	Bad request	요청에 문법 오류가 있음
	401	Unauthorized	요청에 적절한 인증 정보가 없음
	403	Forbidden	서비스 또는 자원 접근이 거부됨
	404	Not found	요청한 문서를 찾을 수 없음
	405	Method not allowed	해당 URL에서 요청한 메서드를 지원하지 않음
	406	Not acceptable	요청한 형식으로 응답할 수 없음
	408	Request timeout	클라이언트가 제한 시간 내에 요청을 완료하지 못함
Server Error	500	Internal server error	서버 내부에 오류가 발생함
	501	Not implemented	요청한 기능을 서버가 수행할 수 없음
	502	Bad gateway	게이트웨이나 프록시 서버가 상위 서버로부터 잘못된 응답을 받음
	503	Service unavailable	서비스를 일시적으로 사용할 수 없음
	504	Gateway timeout	게이트웨이나 프록시 서버가 상위 서버의 응답을 제한 시간 내에 받지 못함

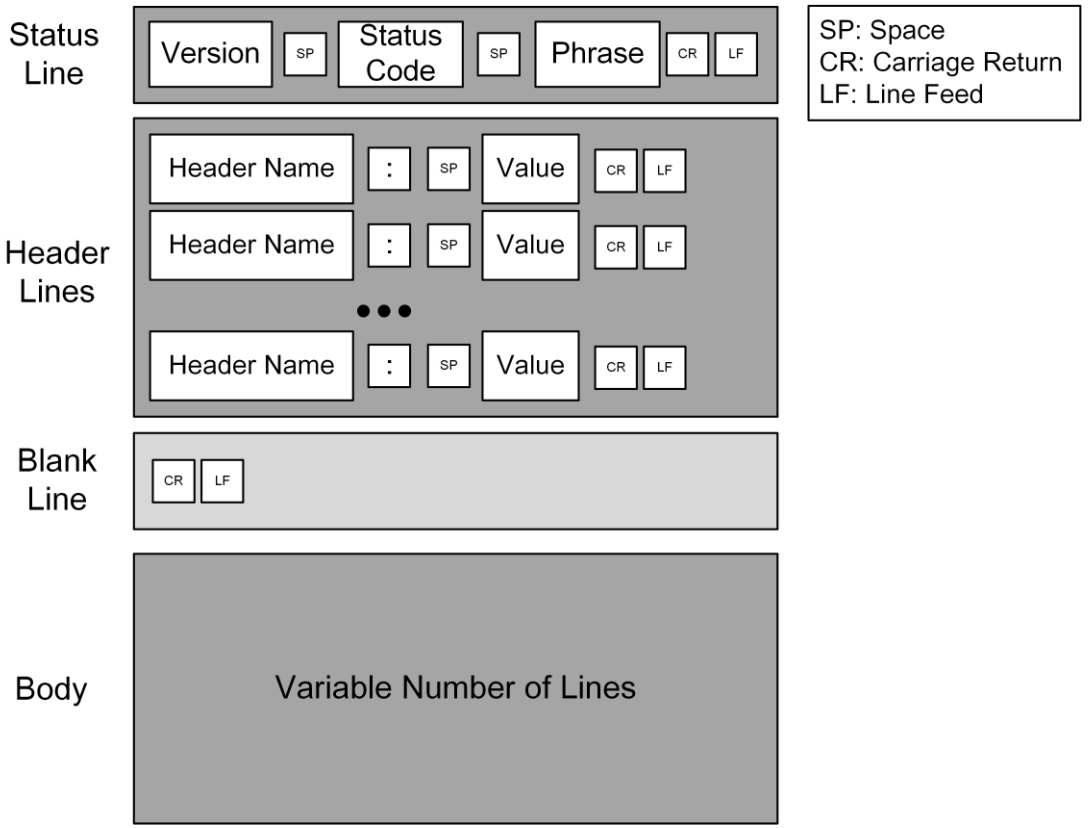


- 트랜잭션 (7/11)
  - 응답 (3/4)
    - 헤더 라인(Header Line)
      - 서버로부터 클라이언트로의 추가 정보를 전송함
      - 헤더 이름, 콜론, 공백, 헤더 값으로 구성됨

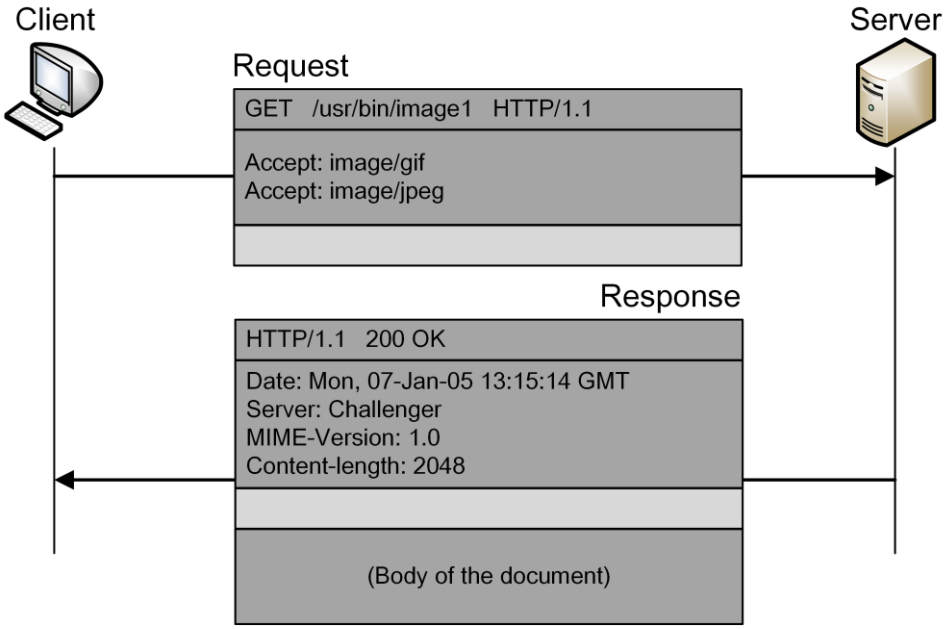
Header	Description
Date	현재 날짜와 시간을 나타냄
Upgrade	선호하는 통신 프로토콜을 지정함
Server	서버에 대한 정보를 제공함
Set-Cookie	서버가 클라이언트에게 쿠키를 저장하도록 요청함
Content-Encoding	콘텐츠에 적용된 인코딩 방식을 지정함
Content-Language	콘텐츠의 언어를 지정함
Content-Length	문서 또는 응답 본문의 길이를 나타냄
Content-Type	콘텐츠의 미디어 유형을 지정함
Location	클라이언트가 요청을 다른 위치로 보내도록 지정함
Accept-Ranges	서버가 요청된 바이트 범위 전송을 지원함을 나타냄
Last-Modified	문서가 마지막으로 변경된 날짜와 시간을 나타냄



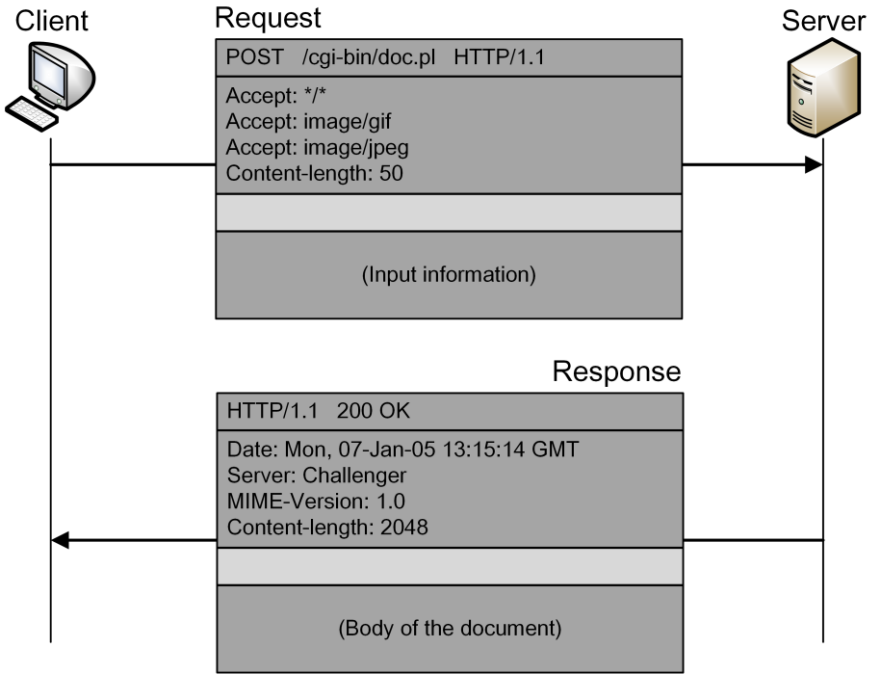
- 트랜잭션 (8/11)
  - 응답 (4/4)
    - 몸체(Body)
      - HTML 문서, 이미지, 파일 등 응답에 필요한 정보가 포함됨



- 트랜잭션 (9/11)
  - 파일 요청 예시
    - 클라이언트가 이미지를 요청하기 위해 GET Method를 사용하고, Accept 헤더 라인을 통해 gif와 jpeg을 받을 수 있음을 알림
    - 서버는 요청을 성공적으로 처리하고 응답 메시지를 전송함



- 트랜잭션 (10/11)
  - 데이터 송신 요청 예시
    - 클라이언트가 데이터 송신을 위해 POST Method를 사용하고, Body에 전송할 데이터를 추가함
    - 서버는 요청을 성공적으로 처리하고 응답 메시지를 전송함



HTTP

- 트랜잭션 (11/11)
  - TELNET을 통한 요청 송신 및 응답 수신 예시
    - TELNET을 통해 웹 서버의 TCP 80번 포트로 접속하여 HTTP 요청을 입력함
    - 서버로부터 요청에 의한 응답을 직접 수신받을 수 있음

```
$telnet www.mhhe.com 80
Trying 198.45.24.104...
Connected to www.mhhe.com (198.45.24.104).
Escape character is '^]'.
GET /engcs/compsci/forouzan HTTP/1.1
From: forouzanbehrouz@fhda.edu
```

```
HTTP/1.1 200 OK
Date: Thu, 28 Oct 2004 16:27:46 GMT
Server: Apache/1.3.9 (Unix) ApacheJServ/1.1.2
PHP/4.1.2 PHP/3.0.18
MIME-version:1.0
Content-Type: text/html
Last-modified: Friday, 15-Oct-04 02:11:31 GMT
Content-length: 14230

Connection closed by foreign host.
```

HTTP

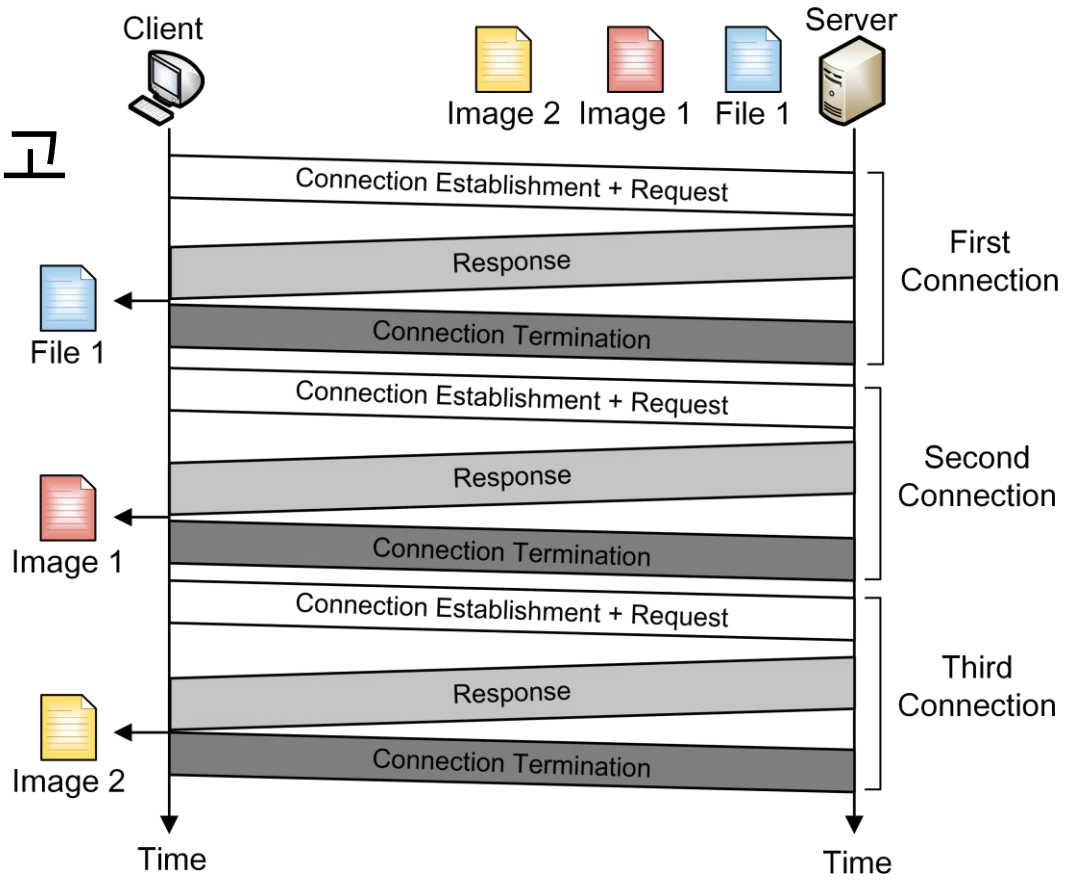
- 조건부 요청
  - If-Modified-Since 등의 헤더 라인을 이용하면 서버에게 특정 조건을 만족한 경우에만 응답하라고 요청할 수 있음
  - 해당 조건이 만족되면 서버는 요청한 웹 페이지를 응답하고, 만족되지 않으면 해당 내용을 통보함

```
GET http://www.commonServer.com/information/file1 HTTP/1.1
If-Modified-Since: Thu, Sept 04 00:00:00 GMT
```

```
HTTP/1.1 304 Not Modified
Date: Sat, Sept 06 08 16:22:46 GMT
Server: commonServer.com

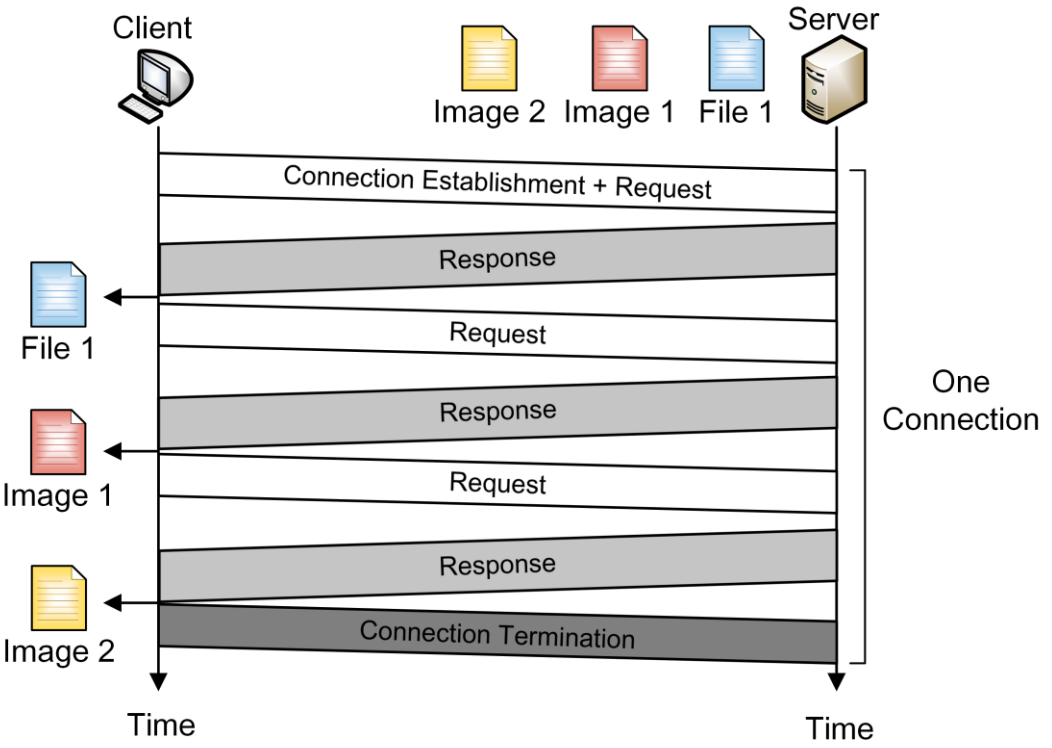
(Empty Body)
```

- 영속성 (1/2)
  - 비영속적 연결
    - 각 요청/응답에 하나의 TCP 연결이 수립됨
    - 연결마다 버퍼를 할당하고 느린 시작(Slow Start) 과정을 거쳐야 하므로 오버헤드가 큼



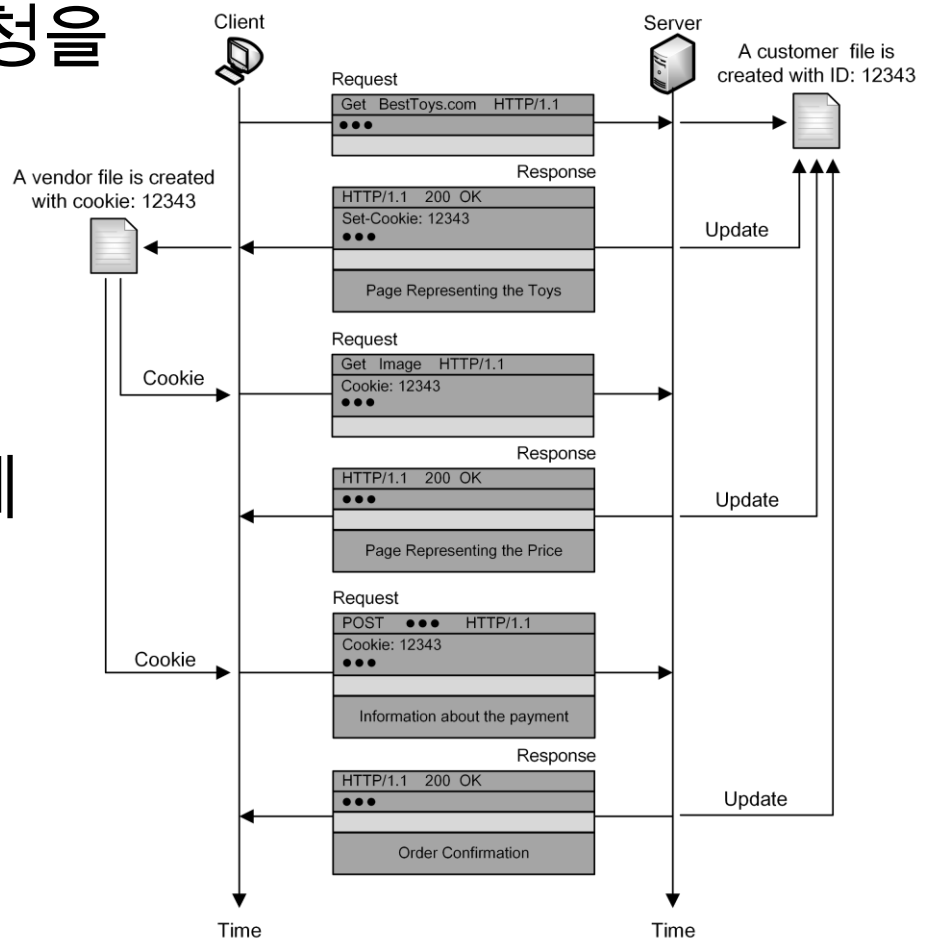
HTTP

- 영속성 (2/2)
  - 영속적 연결
    - 클라이언트가 종료를 요청하거나 서버가 타임아웃 되기 전까지 연결을 유지함
    - 각 응답은 해당 응답의 끝을 알리기 위한 데이터 길이를 포함함
    - 서버가 데이터 길이를 알기 어려운 경우, 이를 클라이언트에게 알리고 응답 후 연결을 종료함

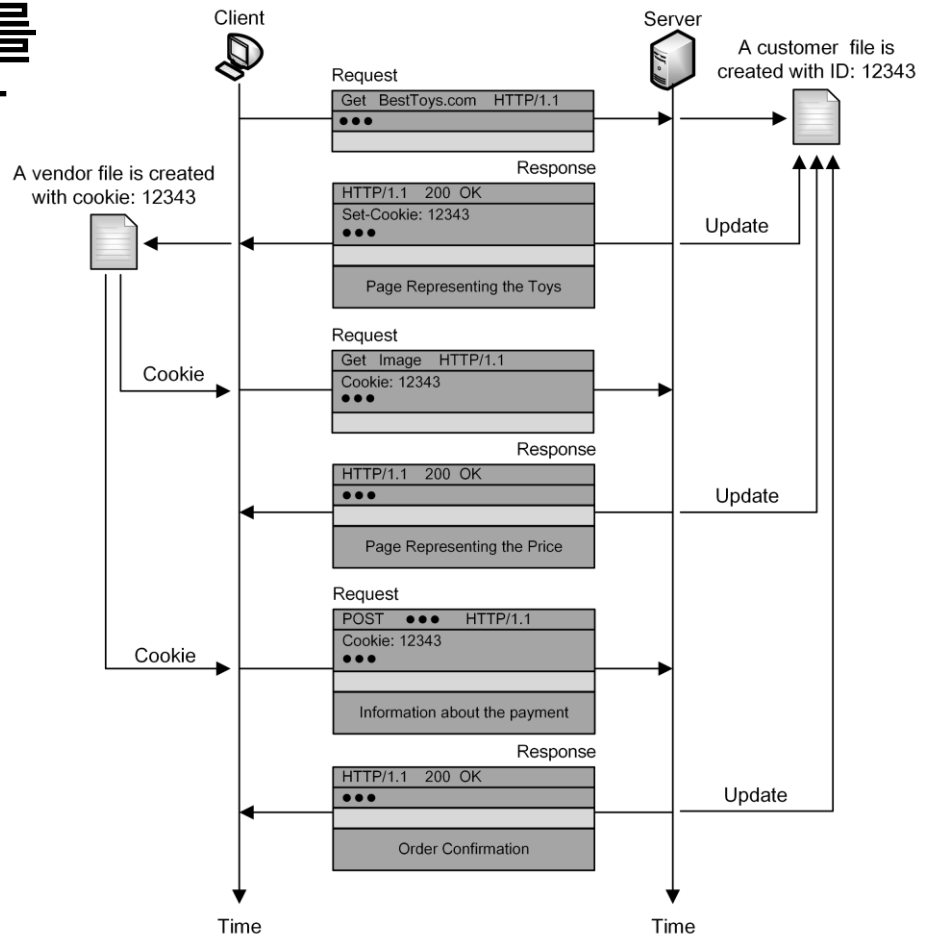


- 쿠키 (1/3)
  - 개요
    - WWW는 Stateless로 설계되었으나, 웹 사이트가 온라인 쇼핑, 접근 통제, 포털, 광고 등을 목적으로 사용하면서 상태의 유지가 필요해짐
  - 생성 및 저장
    1. 서버가 클라이언트로부터 요청을 받을 때, 클라이언트와 관련된 정보를 파일이나 문자열로 저장함
    2. 서버가 클라이언트에게 쿠키를 포함한 응답을 전송함
    3. 클라이언트의 브라우저가 응답 내 쿠키를 쿠키 디렉토리에 저장함

- 쿠키 (2/3)
  - 사용 예시 (1/2)
    1. 클라이언트가 서버에게 요청을 전송함
    2. 서버는 클라이언트에 대해 ID를 부여하고 Set-Cookie 헤더 라인을 포함한 응답을 전송함
    3. 클라이언트는 Set-Cookie에 따라 쿠키를 생성하고 이후 요청부터 쿠키를 포함함



- 쿠키 (3/3)
  - 사용 예시 (2/2)
    4. 서버는 클라이언트를 쿠키를 통해 구분하고, 요청에 대한 정보를 쿠키에 최신화 함
    5. 클라이언트는 입력 값을 포함하는 요청에 대해서도 쿠키를 포함함
    6. 서버는 클라이언트의 입력 값 또한 쿠키에 최신화 함



HTTP

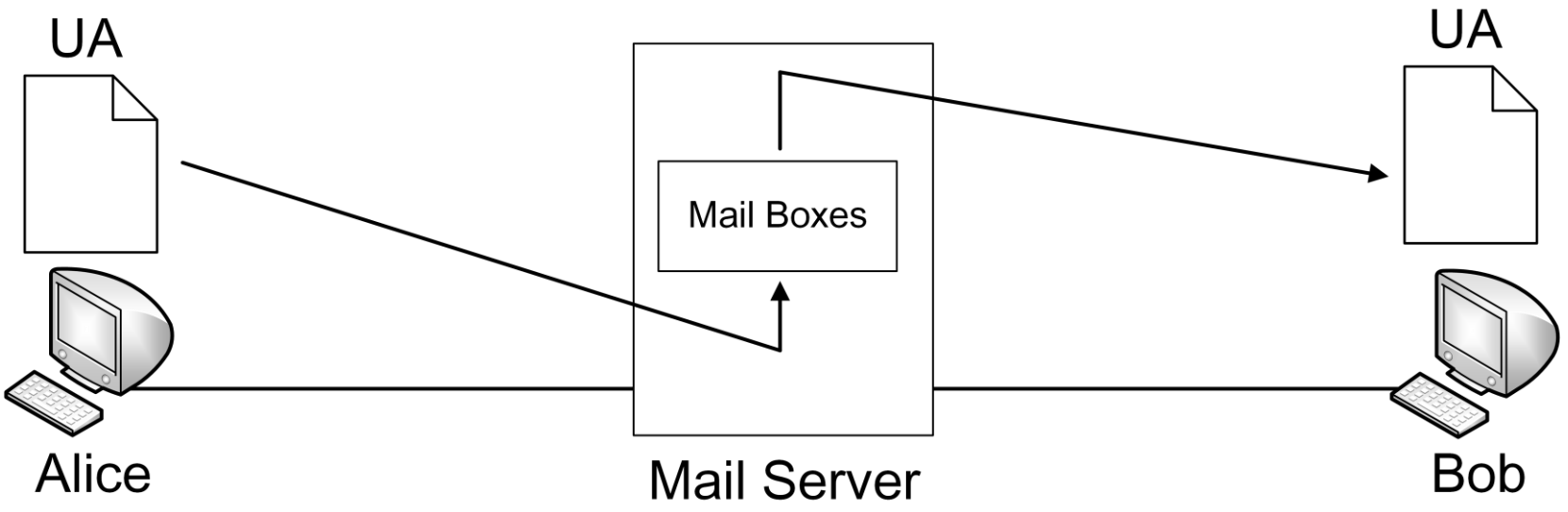
- 프록시 서버(Proxy Server)
  - HTTP는 프록시 서버를 지원하므로, 클라이언트가 허용하면 서버 부하 감소를 위해 클라이언트 측에 프록시 서버를 두고 자주 사용하는 문서를 저장하여 웹 캐싱을 제공할 수 있음
  - 클라이언트 요청에 대한 응답이 캐싱되어 있는 경우, 프록시 서버는 클라이언트에게 서버로서 동작함
  - 클라이언트 요청에 대한 응답이 캐싱되지 않은 경우, 프록시 서버는 서버에게 클라이언트로서 동작함
- 보안
  - HTTP 자체는 보안을 제공하지 않음
  - 보안이 필요한 경우, TLS (Transport Layer Security) 위에 적용하여 HTTPS (Hypertext Transfer Protocol Secure)를 구현함

목 차

- 월드 와이드 웹(WWW, World Wide Web)
- HTTP (Hypertext Transfer Protocol)
- 전자우편

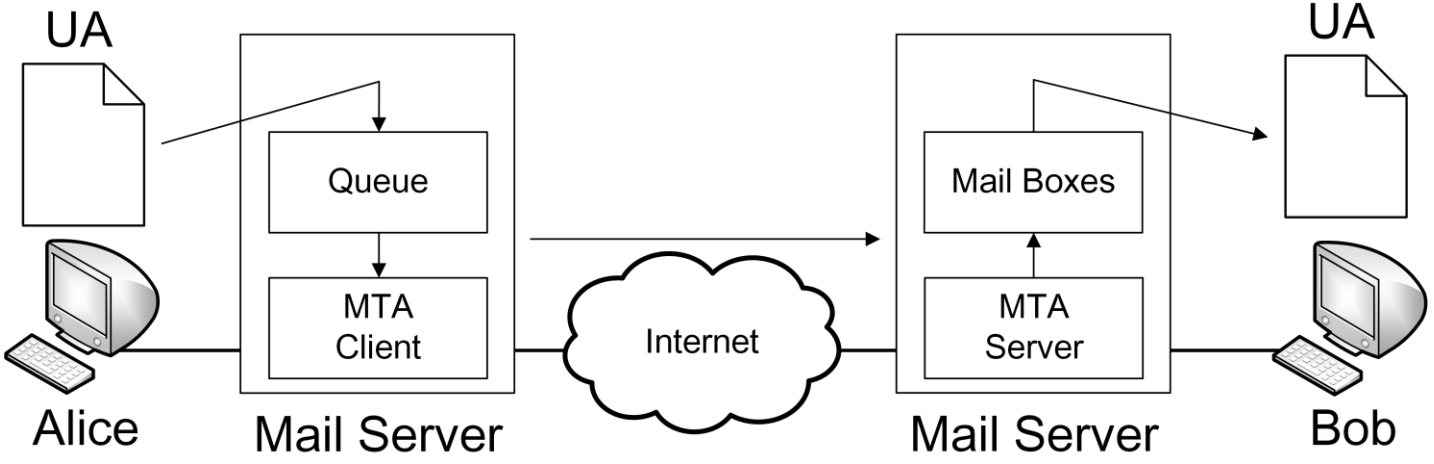
전자우편

- 구조 (1/6)
  - 동일 시스템 환경
    - 동일한 시스템에 있는 사용자가 공유된 메일 서버에 직접 연결된 경우, 메일 서버에는 사용자마다 메일박스가 생성됨
    - 송신자는 UA (User Agent)를 통해 수신자의 메일박스에 전자우편을 저장함
    - 수신자는 필요할 때 UA를 통해 전자우편을 확인할 수 있음



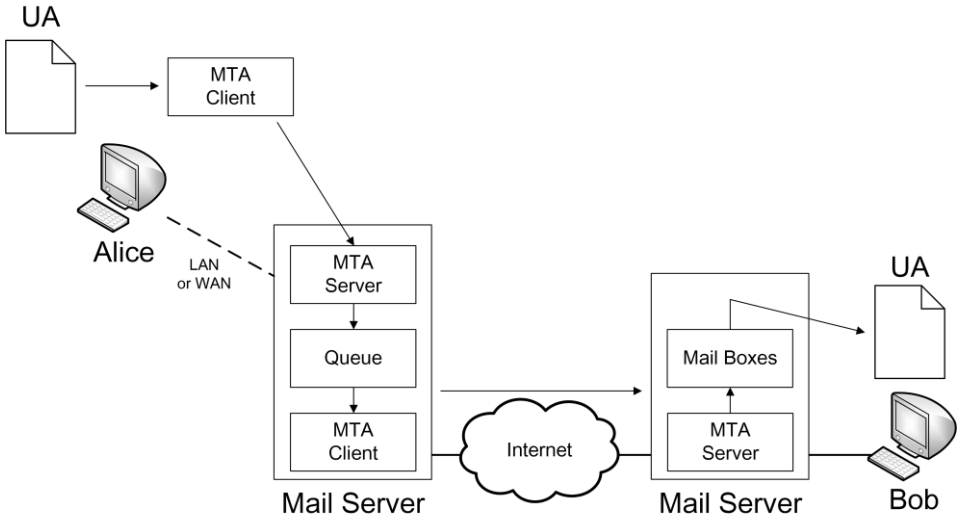
전자우편

- 구조 (2/6)
  - 분리 시스템 환경
    - 사용자가 서로 다른 시스템 환경에 존재하는 경우, 전자우편은 인터넷을 통해 전송되어야 함
    - 송신자가 UA로 전자우편을 큐에 저장하면 MTA (Message Transfer Agent) 클라이언트가 인터넷을 통해 수신자의 MTA 서버로 전자우편을 전송함
    - 수신자는 필요할 때 UA를 통해 전자우편을 확인할 수 있음



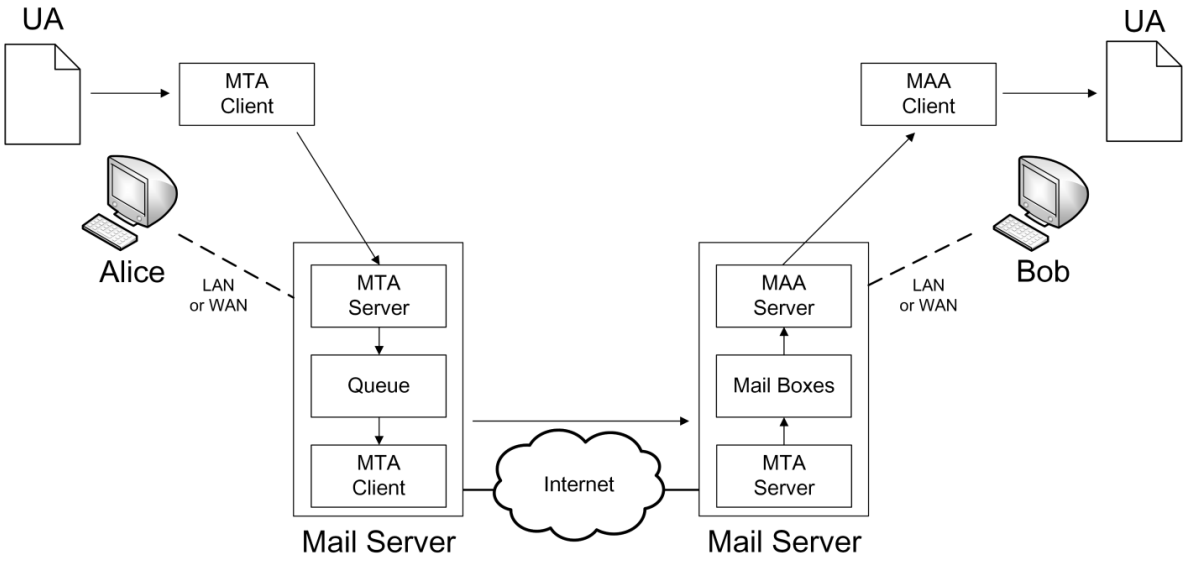
전자우편

- 구조 (3/6)
  - 메일 원격 전송 환경
    - 송신자가 메일 서버와 떨어져 있는 경우, 전자우편은 송신자의 메일 서버를 거쳐 전송되어야 함
    - 송신자가 UA로 MTA 클라이언트에 전자우편을 전송하면, 메일서버의 MTA 서버가 큐를 거쳐 MTA 클라이언트로 수신자의 MTA 서버로 전자우편을 전송함
    - 수신자는 필요할 때 UA를 통해 전자우편을 확인할 수 있음



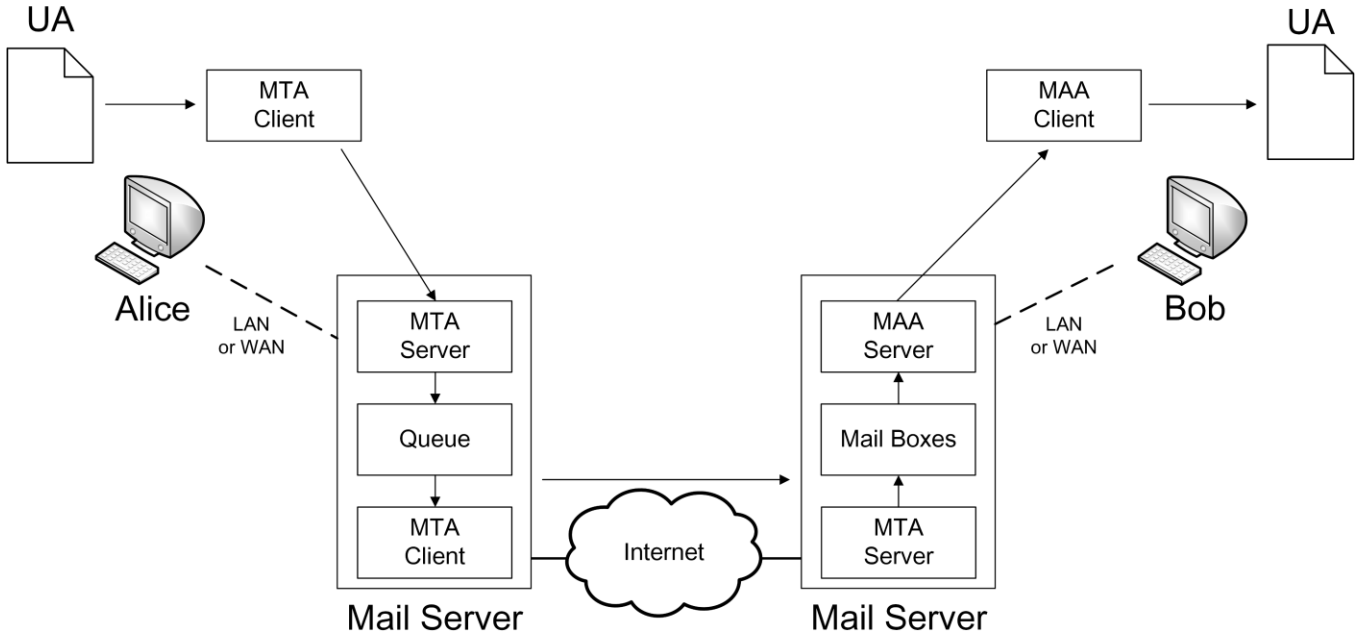
전자우편

- 구조 (4/6)
  - 메일 원격 접근 환경 (1/2)
    - 수신자가 메일 서버와 떨어져 있는 경우, 수신자는 MAA (Message Access Agent)를 통해 전자우편을 가져와야 함
    - 송신자가 UA로 MTA 클라이언트에 전자우편을 전송하면, 메일서버의 MTA 서버가 큐를 거쳐 MTA 클라이언트로 수신자의 MTA 서버로 전자우편을 전송함



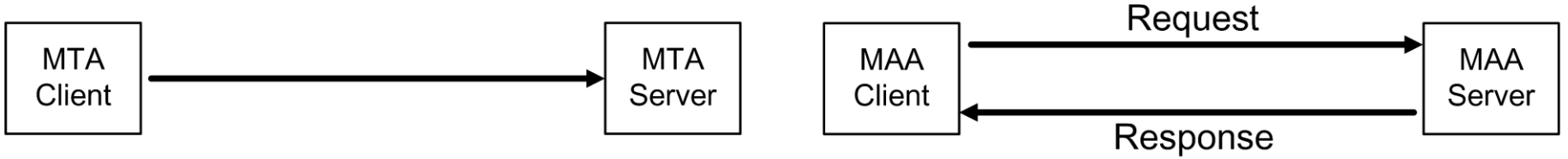
전자우편

- 구조 (5/6)
  - 메일 원격 접근 환경 (2/2)
    - 수신자는 필요할 때 MAA 클라이언트를 통해 MAA 서버에게 전자우편을 요청하고, MAA 클라이언트가 UA에게 전자우편을 전달하면 UA를 통해 이를 확인할 수 있음



전자우편

- 구조 (6/6)
  - PULL과 PUSH
    - MTA 클라이언트는 MTA 서버에게 일방적으로 전송하는 PUSH 방식으로 전자우편을 전송함
    - MAA 클라이언트는 MAA 서버에게 요청을 통해 가져오는 PULL 방식으로 전자우편을 수신함
    - 메일 서버는 상시 메일 수신 대기를 해야 하므로, MAA 서버는 수신자의 메일 서버에 존재하면서 MAA 클라이언트의 PULL 요청에 의해 메일을 전송해야 함



a. Client pushes messages

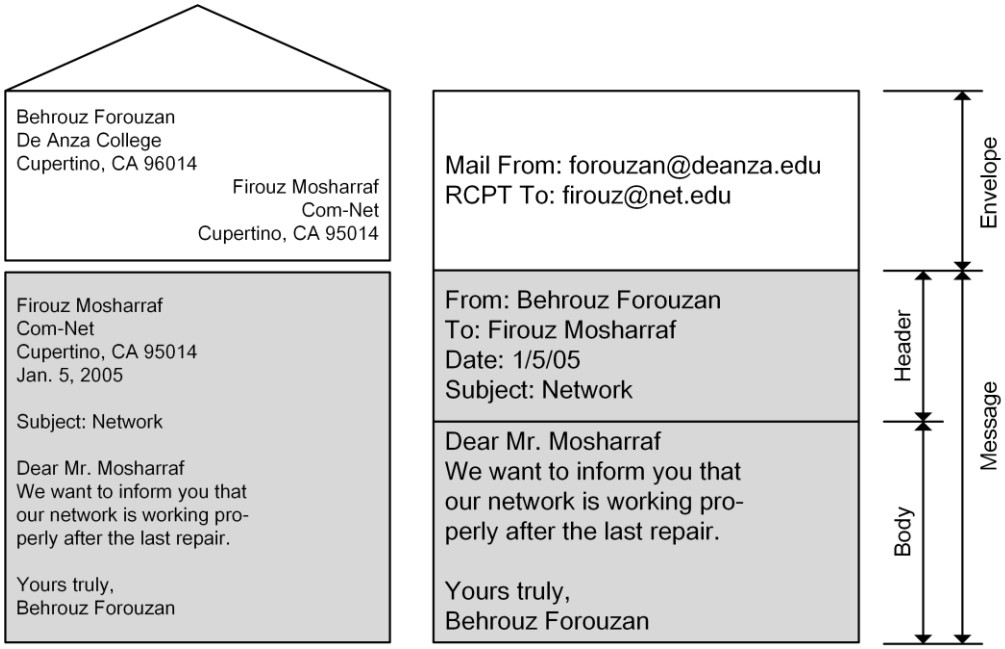
b. Client pulls messages

전자우편

- UA (User Agent) (1/4)
  - 개요
    - UA는 메시지에 대해 구성, 읽기, 응답 등의 기능을 수행하는 프로그램이며, 사용자 컴퓨터에서 로컬 메일박스를 처리함
  - 유형
    - 명령 수행형(Command Driven) UA
      - 명령어를 입력하여 사용하는 UA
        - e.g., Mail, Pine, Elm 등
    - GUI 기반형(Graphical User Interface Based) UA
      - 그래픽 인터페이스를 이용하는 UA
        - e.g., Eudora, Outlook, Netscape 등
  - 전자우편 수신
    - 사용자나 타이머 등에 의해 메일박스에 전자우편이 있음이 확인되면, UA는 사용자에게 전자우편이 왔음을 통보함
    - 사용자 요청에 따라 전자우편의 요약 정보나 메시지를 보여줌

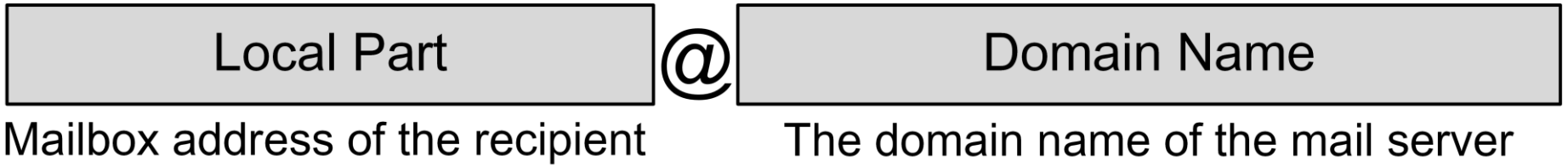
전자우편

- UA (User Agent) (2/4)
  - 전자우편 송신
    - 개요
      - 사용자는 종이 우편을 보내듯이 UA를 통해 봉투(Envelope)와 메시지(Message)로 구분된 전자우편을 생성함
    - 봉투
      - 송신자 주소, 수신자 주소 등의 정보들로 구성됨
    - 메시지
      - 송신자, 수신자, 제목 등의 정보로 이루어진 헤더와 수신자가 실제 읽을 몸체로 구성됨



전자우편

- UA (User Agent) (3/4)
  - 주소 (1/2)
    - 개요
      - 전자우편 처리 시스템은 유일한 주소를 갖는 주소 체계를 사용해야 하며, 인터넷에서 주소 체계는 @ 기호에 의해 로컬 부분과 도메인 이름으로 구성됨
    - 로컬 부분(Local Part)
      - 특정 사용자의 메일함을 식별하는 부분이며, 해당 메일 서버 안에서 어느 사용자의 메일함인지 나타냄
    - 도메인 네임(Domain Name)
      - 메일을 처리할 메일 서버 측 도메인을 식별하는 부분이며, DNS를 통해 해당 조직의 메일 서버를 찾는 데 사용됨

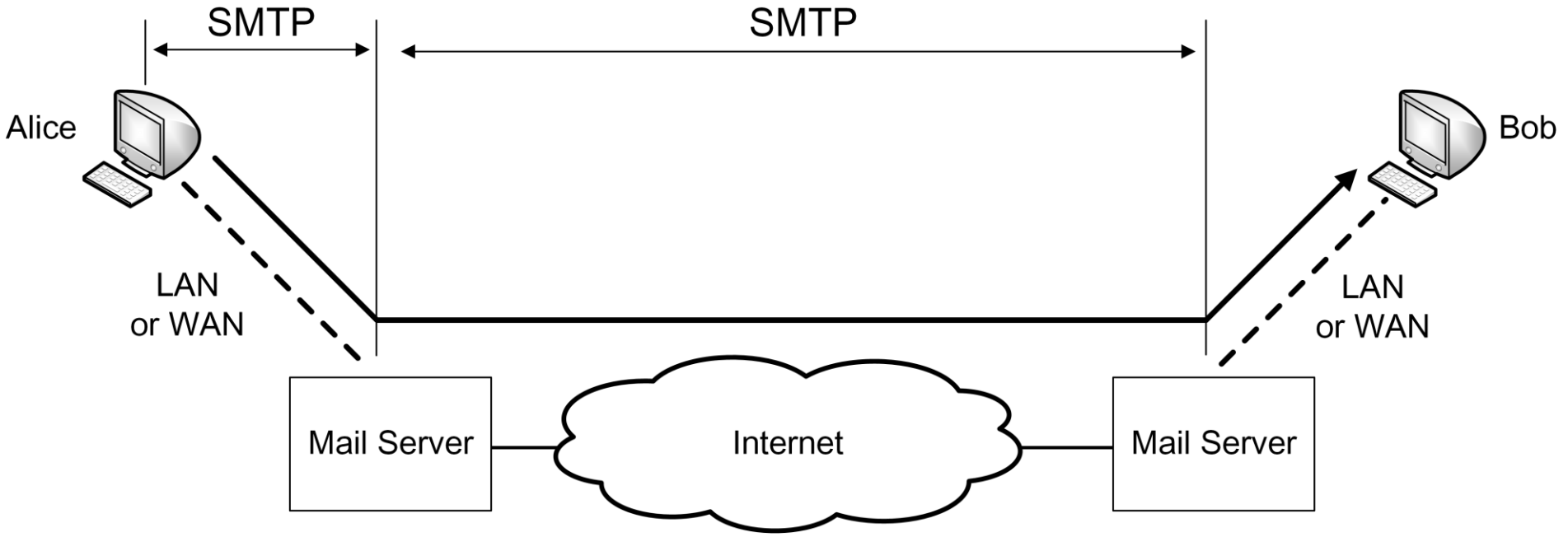


전자우편

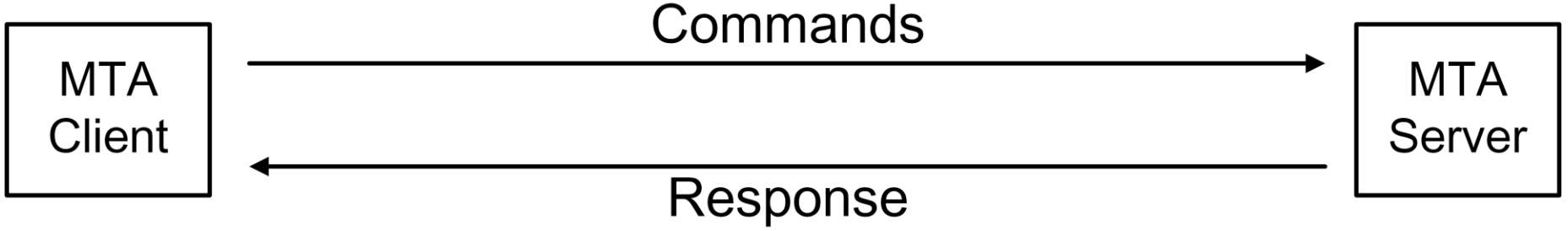
- UA (User Agent) (4/4)
  - 주소 (2/2)
    - 주소 목록(Address List)과 별칭(Alias)
      - 하나의 이름으로 여러 전자우편 주소를 묶어 관리하는 목록을 주소 목록이라고 하며, 이를 나타내는 대표 이름을 별칭이라고 함
      - 별칭을 대상으로 메일을 수신하는 경우, 해당 별칭이 가리키는 주소 목록 내 주소에게 메일을 복제하여 전송함
        - e.g., members-all@pel.sejong.ac.kr로 메일을 보내는 경우, 해당 주소 목록에 포함된 seonghyeon@pel.sejong.ac.kr, yunhong@pel.sejong.ac.kr, minhee@pel.sejong.ac.kr 등에게 모두 메일이 전송됨

전자우편

- MTA 에이전트
  - 전자우편을 송수신하기 위해서는 MTA 클라이언트와 MTA 서버가 필요함
  - SMTP를 통해 MTA 클라이언트와 서버의 송수신 규칙을 규정함



- SMTP (Simple Mail Transfer Protocol) (1/11)
  - 명령과 응답 (1/6)
    - 개요
      - SMTP는 명령(Command)과 응답(Response)을 사용하여 MTA 클라이언트와 MTA 서버 사이의 메시지를 전송함
      - 각 명령과 응답은 CR과 LF로 이루어진 EOL (End Of Line)으로 종료를 알림



전자우편

- SMTP (Simple Mail Transfer Protocol) (2/11)
  - 명령과 응답 (2/6)
    - 명령 (1/4)
      - 개요
        - 명령은 클라이언트에서 서버로 전송되며, “Keyword: Argument(s)”의 형태로 구성됨
        - 각 키워드는 0개 이상의 인수(Argument)를 사용함

Keyword	Argument(s)	Keyword	Argument(s)
HELO	송신자의 호스트 이름	NOOP	-
MAIL FROM	메시지 송신자	TURN	-
RCPT TO	의도된 수신자	EXPN	주소 목록
DATA	전자우편 본문	HELP	명령 이름
QUIT	-	SEND FROM	송신자 경로/주소
RSET	-	SOML FROM	송신자 경로/주소
VRFY	수신자의 이름	SAML FROM	송신자 경로/주소

전자우편

- SMTP (Simple Mail Transfer Protocol) (3/11)
  - 명령과 응답 (3/6)
    - 명령 (2/4)
      - HELO
        - “HELO: 도메인 네임” 형태로 구성되며, 클라이언트를 알림
      - MAIL FROM
        - “MAIL FROM: 메일 주소” 형태로 구성되며, 전자우편 송신자를 알림
      - RCPT TO
        - “RCPT TO: 메일 주소” 형태로 구성되며, 메시지의 의도된 수신자를 알림
      - DATA
        - “DATA” 형태로 구성되며, 전자우편 전송을 시작할 것임을 알림
          - 서버로부터 메일 입력 시작 응답을 받으면 각 줄을 EOL로 구분하여 메시지를 한 줄씩 전송함
          - 메시지 전송이 종료된 경우, 마침표만을 전송하여 메시지 전송이 종료되었음을 알림
      - QUIT
        - “QUIT” 형태로 구성되며, 연결을 종료함

전자우편

- SMTP (Simple Mail Transfer Protocol) (4/11)
  - 명령과 응답 (4/6)
    - 명령 (3/4)
      - RSET
        - “RSET” 형태로 구성되며, 현재의 전자우편 트랜잭션을 중지시켜 연결을 유지한 상태에서 송수신자에 대해 저장된 정보가 삭제되고 초기화 됨
      - VRFY
        - “VRFY: 메일 주소” 형태로 구성되며, 인수의 수신자 주소가 정당한 수신자인지 확인하라는 요청을 함
      - NOOP
        - “NOOP” 형태로 구성되며, 수신자의 상태를 확인하기 위해 응답을 요청함
      - TURN
        - “TURN” 형태로 구성되며, 현재 연결에서 송수신자의 위치를 교환함
      - EXPN
        - “EXPN: 주소 목록” 형태로 구성되며, 주소 목록 내에 있는 주소들을 반환하도록 요청함

전자우편

- SMTP (Simple Mail Transfer Protocol) (5/11)
  - 명령과 응답 (5/6)
    - 명령 (4/4)
      - HELP
        - “HELP: 명령” 형태로 구성되며, 인수의 명령에 대한 정보를 송신하도록 요청함
      - SEND FROM
        - “SEND FROM: 메일 주소” 형태로 구성되며, 전자우편을 수신자 단말로 직접 전송하도록 요청함
      - SOML FROM
        - “SOML FROM: 메일 주소” 형태로 구성되며, 수신자가 접속 중인 경우에는 수신자 단말로, 그렇지 않은 경우에는 메일박스로 전자우편이 전송되도록 요청함
      - SAML FROM
        - “SAML FROM: 메일 주소” 형태로 구성되며, 수신자가 접속 중인 경우에는 수신자 단말과 메일 박스로, 그렇지 않은 경우에는 메일박스에만 전자우편이 전송되도록 요청함

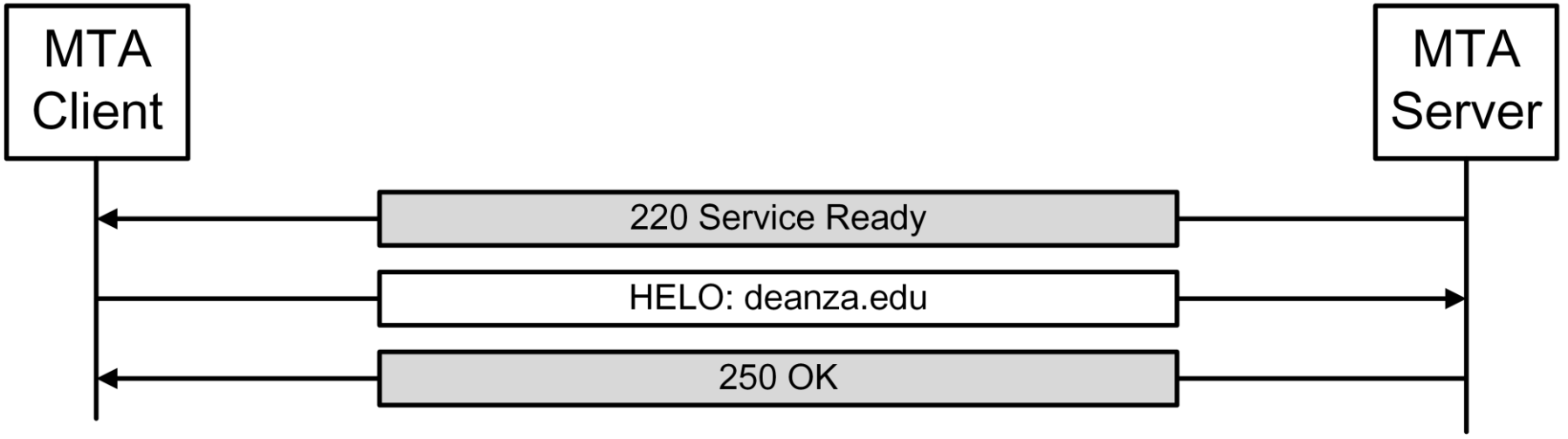
- SMTP (Simple Mail Transfer Protocol) (6/11)
  - 명령과 응답 (6/6)
    - 응답
      - 세 자리 숫자 코드를 통해 명령에 대한 처리 결과를 나타냄

분류	Code	Description
Positive Completion Reply	211	시스템 상태 또는 도움말 응답
	214	도움말 메시지
	220	서비스 준비 완료
	221	서비스가 전송 채널을 종료함
	250	요청한 명령이 완료됨
	251	사용자가 로컬에 없으나 메시지가 전달됨
Positive Intermediate Reply	354	메일 입력 시작
Transient Negative Completion Reply	421	서비스를 사용할 수 없음
	450	메일함을 사용할 수 없음
	451	로컬 오류로 인해 명령이 중단됨
	452	저장 공간 부족으로 인해 명령이 중단됨

분류	Code	Description
Permanent Negative Completion Reply	500	인식할 수 없는 명령에 의한 구문 오류
	501	인수의 구문 오류
	502	명령이 구현되지 않음
	503	명령 순서가 올바르지 않음
	504	명령이 일시적으로 구현되지 않음
	550	메일함을 사용할 수 없어 명령이 실행되지 않음
	551	사용자가 로컬에 없음
	552	저장 공간 초과로 인해 요청한 작업이 중단됨
	553	허용되지 않는 메일함 이름으로 인해 요청한 작업이 수행되지 않음
	554	트랜잭션 실패

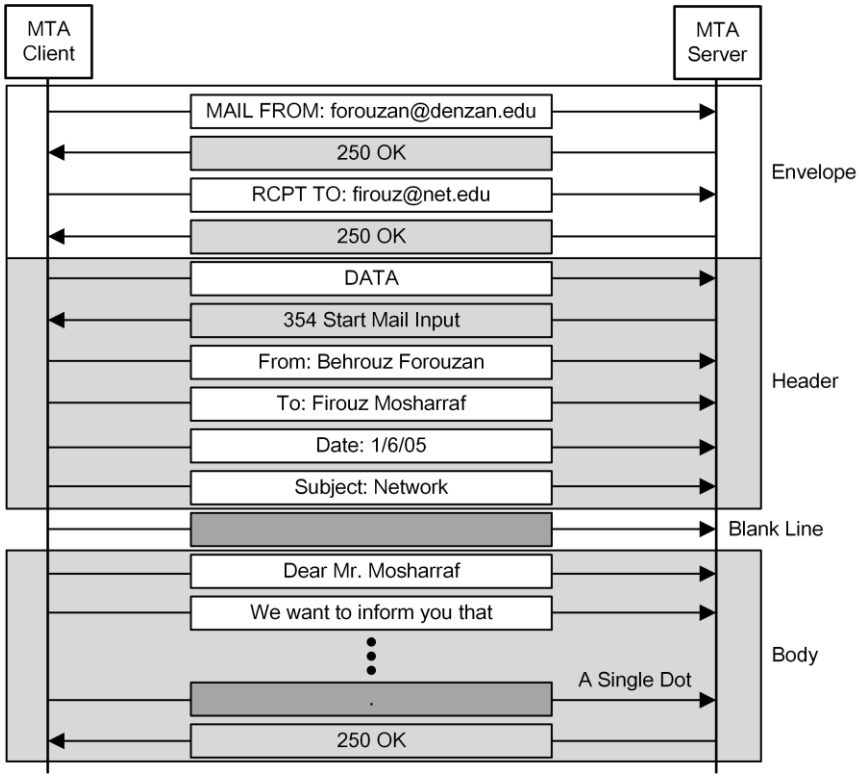
전자우편

- SMTP (Simple Mail Transfer Protocol) (7/11)
  - 전자우편 전송 단계 (1/4)
    - 연결 수립
      - TCP의 Well-known 포트 25번을 이용하여 연결을 수립한 후, 다음 과정을 통해 SMTP 연결을 수립함
        1. 서버가 220 코드를 전송하여 전자우편을 받을 준비가 되었음을 알림
        2. 클라이언트가 HELO 명령을 통해 자신의 도메인 네임을 알림
        3. 서버가 250 코드를 전송하여 연결이 성공했음을 알림



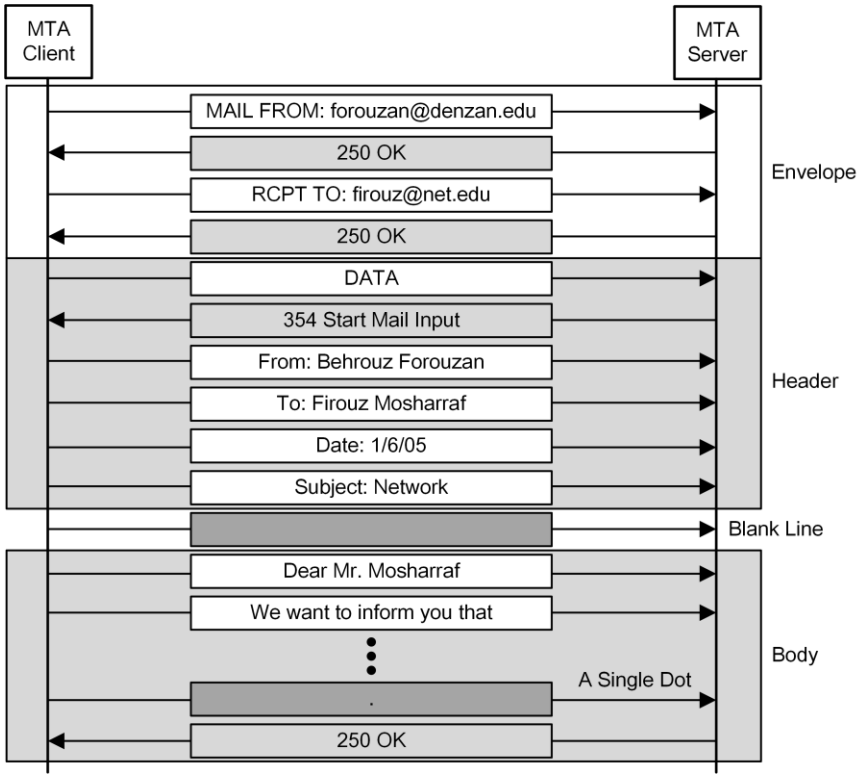
전자우편

- SMTP (Simple Mail Transfer Protocol) (8/11)
  - 전자우편 전송 단계 (2/4)
    - 메시지 전송 (1/2)
      1. 클라이언트가 MAIL FROM 명령을 통해 메시지의 송신자를 소개함
      2. 서버는 250 코드를 전송하여 명령을 수신했음을 알림
      3. 클라이언트가 RCPT TO 명령을 통해 메시지 수신자를 알림
      4. 서버는 250 코드를 전송하여 명령을 수신했음을 알림
      5. 클라이언트는 DATA 명령을 통해 메시지 시작을 알림
      6. 서버는 354 코드를 전송하여 메일 입력 시작을 알림



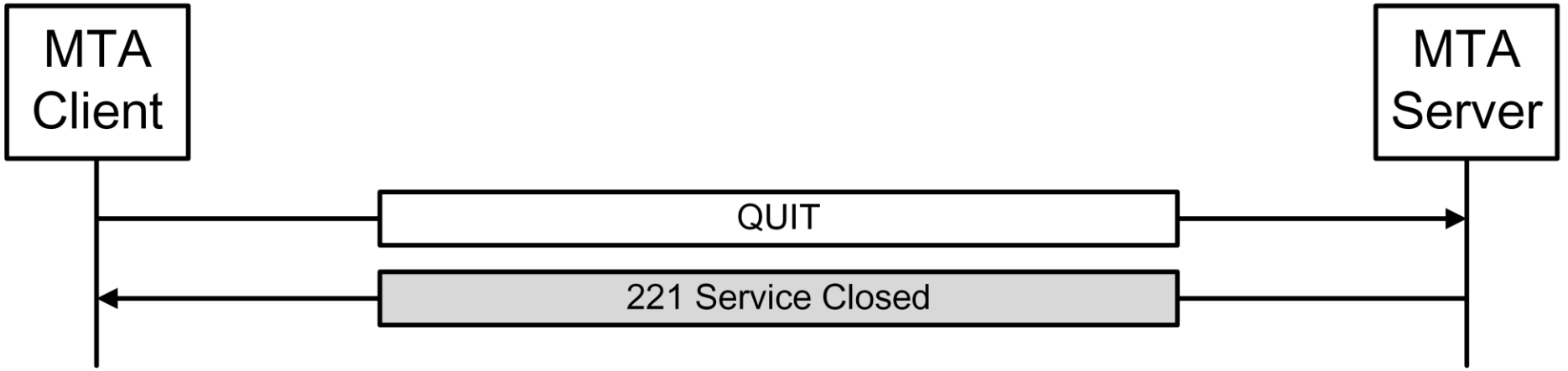
전자우편

- SMTP (Simple Mail Transfer Protocol) (9/11)
  - 전자우편 전송 단계 (3/4)
    - 메시지 전송 (2/2)
      7. 클라이언트는 EOL로 구분하여 메시지를 한 줄씩 전송함
      8. 클라이언트는 메시지 전송이 종료되면 마침표만을 보내어 메시지 전송이 종료됨을 알림
      9. 서버는 250 코드를 전송하여 메시지 전송 종료를 확인했음을 알림



전자우편

- SMTP (Simple Mail Transfer Protocol) (10/11)
  - 전자우편 전송 단계 (4/4)
    - 연결 종료
      - 메시지 전송이 종료되면 다음 과정을 통해 SMTP 연결을 종료한 후, TCP 연결을 종료함
        1. 클라이언트가 QUIT 명령을 통해 연결을 종료할 것임을 알림
        2. 서버가 221 코드를 전송하여 연결을 종료했음을 알림



전자우편

- SMTP (Simple Mail Transfer Protocol) (11/11)
  - TELNET을 통한 명령 및 응답 수신 예시
    - TELNET을 통해 포트 25로 연결함
    - 서버로부터 대기중 응답을 수신하고 SMTP 연결을 수립함
    - 봉투에 해당하는 부분을 전송함
    - DATA 명령을 보낸 후, 메일 입력 시작 응답을 받아 메시지의 헤더를 전송함
    - 빈줄로 구분하고 몸체 내용을 전송함
    - 메시지 전송이 종료된 후, 메시지 수신 응답을 받아 연결 종료를 진행함

```
$ telnet mail.adelphia.net 25
Trying 68.168.78.100...
Connected to mail.adelphia.net (68.168.78.100).

220 mta13.adelphia.net SMTP server ready Fri. 6 Aug 2004 . . .
HELO mail.adelphia.net
250 mta13.adelphia.net

MAIL FROM: forouzanb@adelphia.net
250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
250 Recipient <forouzanb@adelphia.net> Ok

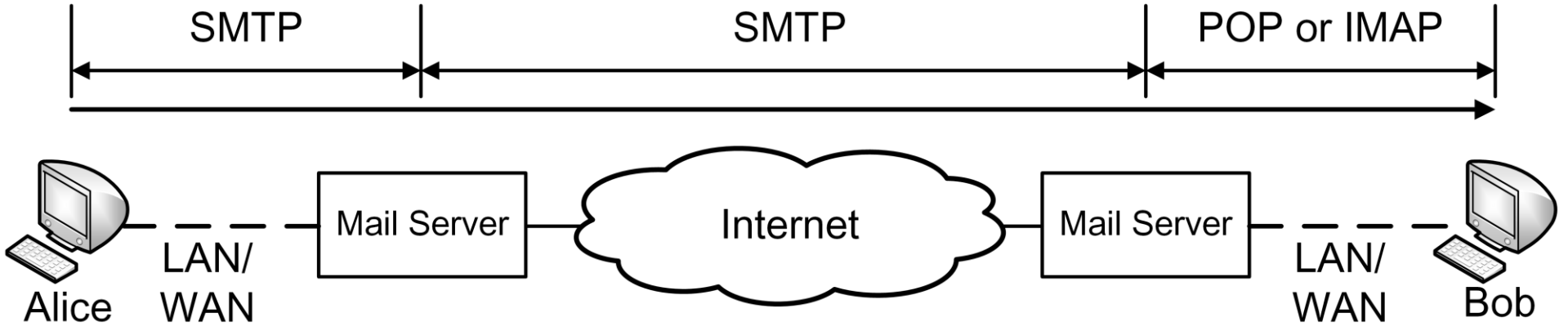
DATA
354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
To: Forouzan

This is a test message
to show SMTP in action.
.

250 Message received: adelphia.net@mail.adelphia.net
QUIT
221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

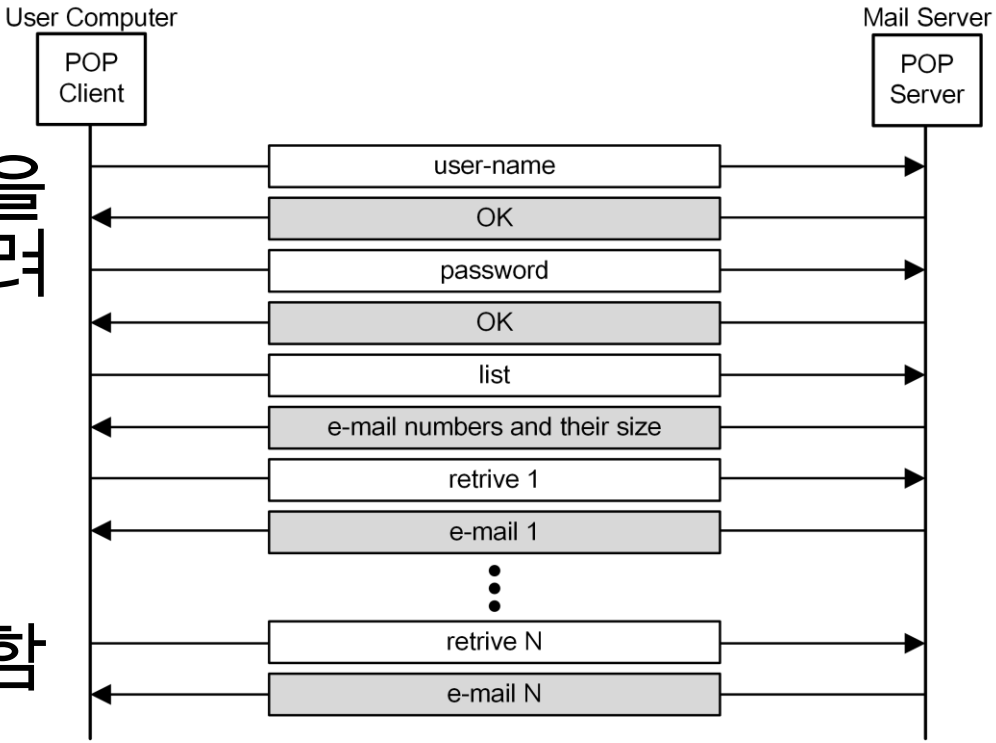
전자우편

- MAA 에이전트
  - SMTP는 PUSH 프로토콜이나, 클라이언트가 서버로부터 전자우편을 가져오기 위해서는 PULL 프로토콜이 필요함
  - POP (Post Office Protocol)이나 IMAP (Internet Mail Access Protocol)을 통해 서버로부터 전자우편을 가져옴



전자우편

- POP (Post Office Protocol)
  - 사용자가 전자우편을 내려받고자 하는 경우, TCP 포트 110을 사용하여 서버에 연결함
  - 사용자 이름과 비밀번호를 통해 계정을 인증해야 함
  - 인증 후에는 전자우편 목록을 보고 필요한 전자우편을 내려 받을 수 있음
  - 내려받은 후 메일박스에서 삭제하는 삭제 모드와 내려받아도 메일박스에서 유지되는 유지 모드를 지원함



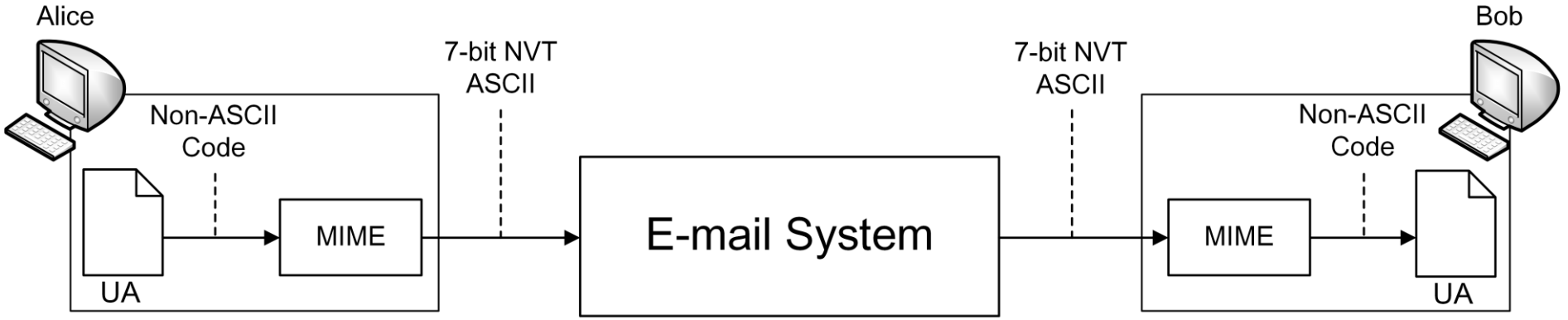
전자우편

- IMAP (Internet Mail Access Protocol)
  - 사용자가 서버에서 전자우편을 체계적으로 정리할 수 있는 추가 기능들을 지원함
    - 전자우편을 내려받기 전 헤더 검사가 가능함
    - 전자우편을 내려받기 전 특정 문자열을 통한 색인이 가능함
    - 전자우편을 부분적으로 내려받을 수 있음
    - 메일 서버에서 메일박스 생성, 삭제, 이름 변경 등의 관리가 가능함
- MAA 프로토콜의 비교
  - POP는 하나의 기기에서 전자우편을 내려받아 로컬 중심으로 관리하거나 오프라인 사용이 중요한 경우에 적합함
  - IMAP은 여러 기기에서 동일한 메일박스를 사용하며 폴더, 전자우편, 읽음 상태 등을 동기화해야 하는 경우에 적합함

전자우편

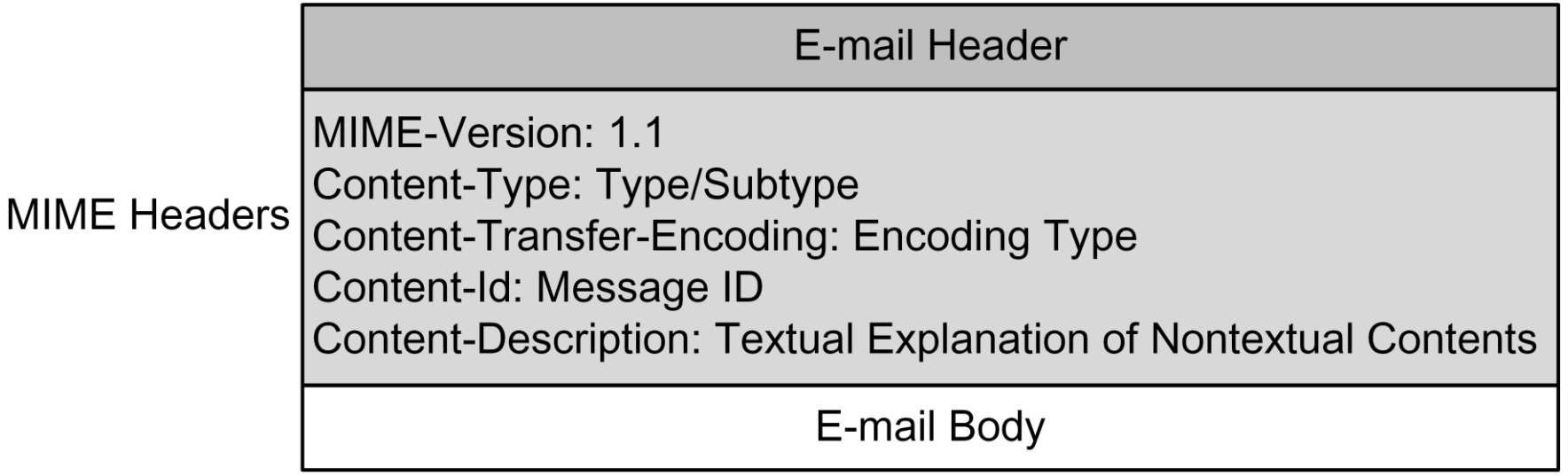
- MIME (Multipurpose Internet Mail Extensions) (1/9)
  - 개요
    - 전자우편 시스템은 7-bit NVT ASCII 형식의 메시지만 전송할 수 있음
    - MIME를 이용하여 ASCII가 아닌 데이터를 전자우편으로 전송할 수 있음

NVT ASCII (Network Virtual Terminal ASCII):
서로 다른 시스템 간에 텍스트를 일관된 형식으로 전송하기 위해 정의된 표준 ASCII 기반 네트워크 문자 표현 방식



전자우편

- MIME (Multipurpose Internet Mail Extensions) (2/9)
  - MIME 헤더 (1/8)
    - 개요
      - MIME는 기존 전자우편 헤더에 변환 인수를 정의하기 위한 MIME 헤더를 추가할 수 있음
    - MIME-Version
      - 사용중인 MIME 버전을 나타냄



- MIME (Multipurpose Internet Mail Extensions) (3/9)
  - MIME 헤더 (2/8)
    - Content-Type (1/4)
      - 개요
        - 메시지 몸체에 사용되는 데이터 종류를 나타냄

Type	Subtype	Description
Text	Plain	서식이 없는 텍스트
	HTML	HTML 형식
Multipart	Mixed	본문이 서로 다른 데이터 유형의 부분들을 순서대로 포함함
	Parallel	Mixed와 동일하지만 부분 간 순서가 없음
	Digest	Mixed와 유사하지만 기본 유형이 message/RFC822임
	Alternative	각 부분이 동일한 메시지의 서로 다른 버전임
Message	RFC822	본문이 캡슐화된 메시지임
	Partial	본문이 더 큰 메시지의 일부임
	External-Body	본문이 다른 메시지에 대한 참조임
Image	JPEG	JPEG 형식의 이미지
	GIF	GIF 형식의 이미지
Video	MPEG	MPEG 형식의 비디오
Audio	Basic	8 kHz의 단일 채널 음성 인코딩
Application	PostScript	Adobe PostScript
	Octet-stream	일반적인 이진 데이터(8비트 바이트)

전자우편

- MIME (Multipurpose Internet Mail Extensions) (4/9)
  - MIME 헤더 (3/8)
    - Content-Type (2/4)
      - Text
        - 원래의 메시지가 7-bit ASCII 형식으로 되어 있어 MIME에 의한 변환이 불필요함
      - Multipart
        - 몸체가 여러 개의 독립적인 부분으로 이루어진 경우에 사용함
        - 인수를 통해 각 부분의 경계를 나타냄
        - Mixed Subtype은 부분들이 수신자에게 순서대로 보여야 함을 나타냄
        - Parallel Subtype은 부분들이 수신자에게 순서대로 보일 필요 없음을 나타냄
        - Digest Subtype은 부분들이 Message/RFC822의 Type/Subtype을 가짐을 나타냄
        - Alternative Subtype은 같은 부분을 서로 다른 형식으로 작성하였음을 나타냄

RFC822:
전자우편 메시지의 헤더와 본문 형식을 정의한 초기 인터넷 메시지 표준

```
Content-Type: multipart/mixed; boundary=xxxx

--xxxx
Content-Type: text/plain;
...
--xxxx
Content-Type: image/gif;
...
--xxxx--
```

전자우편

- MIME (Multipurpose Internet Mail Extensions) (5/9)
  - MIME 헤더 (4/8)
    - Content-Type (3/4)
      - Message
        - RFC822 Subtype은 메시지 몸체가 전자우편 자체임을 나타냄
        - Partial Subtype은 메시지가 단편화 되었으며, 해당 몸체가 단편 중 하나임을 나타냄
          - id 인수는 메시지를 구별하기 위한 식별자를 나타냄
          - number 인수는 해당 메시지가 전체 단편 중 몇 번째 단편인지를 나타냄
          - total 인수는 전체 단편의 개수를 나타냄

```
Content-Type: message/partial;
id="forouzan@challenger.atc.fhda.edu";
number=1;
total=3;
...
```

        - External-body Subtype은 메시지 몸체가 다른 메시지에 대한 참조임을 나타냄

```
Content-Type: message/external-body;
name="report.txt";
site="fhda.edu";
access-type="ftp";
...
```

전자우편

- MIME (Multipurpose Internet Mail Extensions) (6/9)
  - MIME 헤더 (5/8)
    - Content-Type (4/4)
      - Image
        - 몸체가 이미지 파일로 구성됨
      - Video
        - 몸체가 비디오 파일로 구성됨
      - Audio
        - 몸체가 음성 파일로 구성됨
      - Application
        - 몸체가 앞선 Content-Type으로 정의할 수 없는 데이터로 구성됨

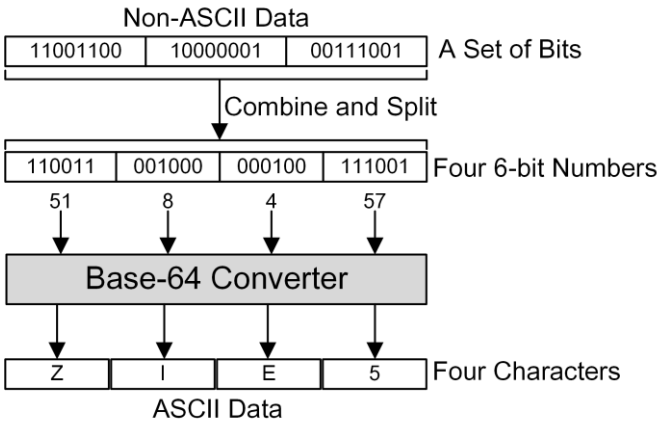
전자우편

- MIME (Multipurpose Internet Mail Extensions) (7/9)
  - MIME 헤더 (6/8)
    - Content-Transfer-Encoding (1/3)
      - 개요
        - 메시지를 인코딩하는 방법을 정의함
      - 7bit
        - 7-bit NVT ASCII 부호화를 사용하며, 한 줄의 최대 길이는 1,000 글자를 초과하지 못함
      - 8bit
        - 8-bit 부호화를 사용하며, 한 줄의 최대 길이는 1,000 글자를 초과하지 못함
        - ASCII가 아닌 문자를 전송할 수 있으나, 하위 SMTP 프로토콜이 8 Bits Non-ASCII 전송이 가능해야 함

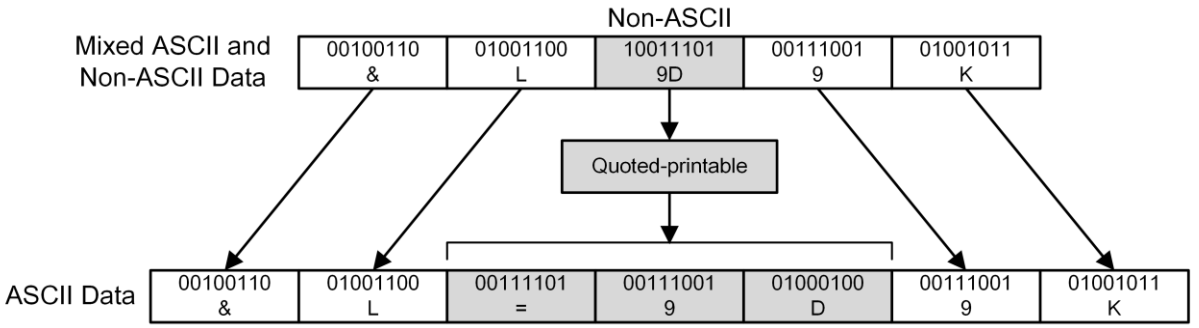
Type	Description
7bit	NVT ASCII 문자와 짧은 행
8bit	비ASCII 문자와 짧은 행
Binary	비ASCII 문자와 길이 제한이 없는 행
Base64	6비트 데이터 블록을 8비트 ASCII 문자로 인코딩
Quoted-printable	비ASCII 문자를 등호(=)와 ASCII 코드로 부호화

전자우편

- MIME (Multipurpose Internet Mail Extensions) (8/9)
  - MIME 헤더 (7/8)
    - Content-Transfer-Encoding (2/3)
      - Binary
        - 8-bit 부호화를 사용하며, 한 줄의 최대 길이에 제한이 없음
        - ASCII가 아닌 문자를 전송할 수 있으나, 하위 SMTP 프로토콜이 2진 데이터를 전송할 수 있어야 함
      - Base64
        - Non-ASCII 데이터를 ASCII 데이터로 변환함
        - 2진 데이터를 24-bit 블록으로 분할하고, 해당 블록을 다시 4개의 6-bit 단락으로 분할함
        - 각 6-bit 단락은 Base-64 변환기에 의해 하나의 ASCII 문자로 변환됨



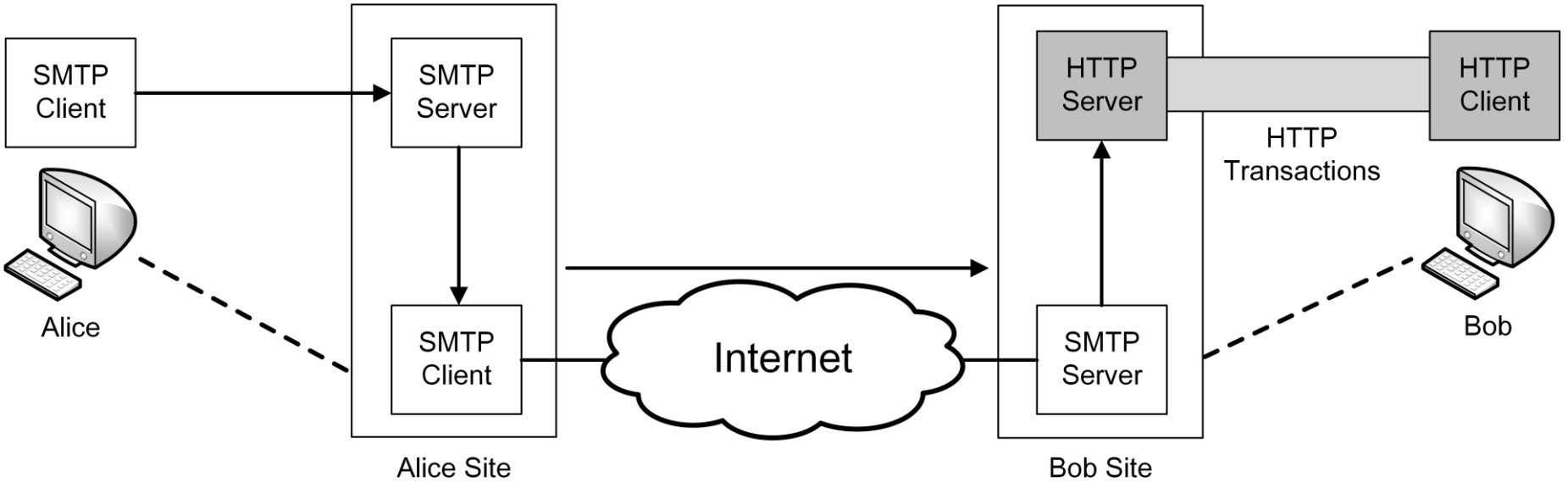
- MIME (Multipurpose Internet Mail Extensions) (9/9)
  - MIME 헤더 (8/8)
    - Content-Transfer-Encoding (3/3)
      - Quoted-printable
        - 일부만 Non-ASCII 문자인 경우에 활용할 수 있음
        - Non-ASCII 문자의 바이트를 16진수로 나타내고 이를 등호와 함께 ASCII 코드로 변환함



    - Content-Id
      - 여러 개의 메시지가 있는 경우에 메시지를 유일하게 식별함
    - Content-Description
      - 몸체 내용에 대해 사람이 읽을 수 있는 설명을 나타냄

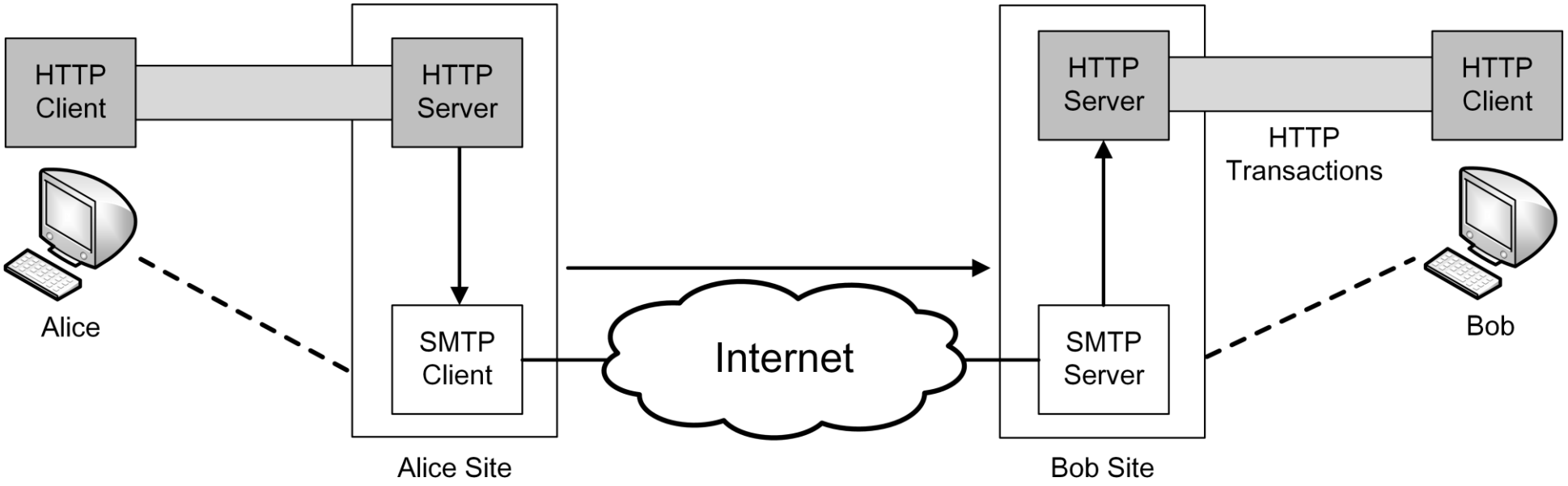
전자우편

- 웹 기반 전자우편 (1/2)
  - 웹 메일 수신자 구조
    - 수신자가 웹 메일을 사용하는 경우, POP, IMAP 대신 HTTP가 사용될 수 있음
    - 수신자가 전자우편을 확인하고자 하는 경우, 웹 메일 서버에 HTTP로 계정정보를 포함한 요청 메시지를 보내면 웹 메일 서버로부터 HTML 형식의 전자우편 목록을 받을 수 있음



전자우편

- 웹 기반 전자우편 (2/2)
  - 전체 웹 메일 사용자 구조
    - 송수신자 모두 웹 메일을 사용하는 경우, SMTP, POP, IMAP 대신 HTTP가 사용될 수 있음
    - 송신자가 송신자의 메일 서버로 메일을 전송할 때에는 HTTP가 사용되지만, 메일 서버 간의 전자우편 전송에는 SMTP가 사용됨



전자우편

- 보안
  - SMTP, MIME는 보안과 관련된 기능을 제공하지 않음
  - POP, IMAP은 사용자 인증과 관련된 기능만을 제공함
  - PGP (Pretty Good Privacy), S/MIME (Secure/Multipurpose Internet Mail Extensions) 등을 통해 메시지 자체에 대한 보안을 제공할 수 있음
  - TLS (Transport Layer Security) 등을 통해 전자우편 송신을 위한 통신 연결 및 전송 구간 보안을 제공할 수 있음

Thanks!

이 성 현(seonghyeon@pel.sejong.ac.kr)

부록 #1

- Base64 인코딩 표

Value	Code	Value	Code	Value	Code	Value	Code	Value	Code	Value	Code
0	A	11	L	22	W	33	h	44	s	55	3
1	B	12	M	23	X	34	i	45	t	56	4
2	C	13	N	24	Y	35	j	46	u	57	5
3	D	14	O	25	Z	36	k	47	v	58	6
4	E	15	P	26	a	37	l	48	w	59	7
5	F	16	Q	27	b	38	m	49	x	60	8
6	G	17	R	28	c	39	n	50	y	61	9
7	H	18	S	29	d	40	o	51	z	62	+
8	I	19	T	30	e	41	p	52	0	63	/
9	J	20	U	31	f	42	q	53	1		
10	K	21	V	32	g	43	r	54	2		