네트워크 HW4



소프트웨어전공 20171596 김수진

**1. HTTP/2 의 기본 개념**

HTTP/2는 응답 다중화를 지원하고 성능을 높여 기존의 HTTP/1.0과 HTTP/1.1의 문제점을 해결한 새로운 프로토콜이다. (M.Belshe, BitGo, PeonR. 2015)

**2. 등장 배경, 역사, 장점**

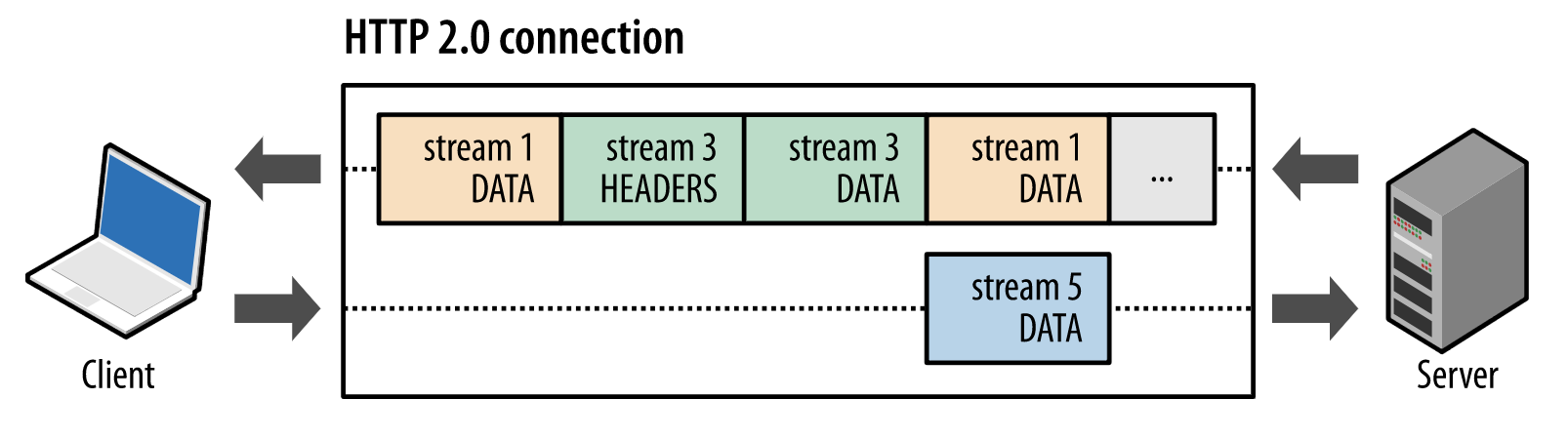
**등장배경과 역사**

Hugues de Saxce는 HTTP가 브라우저와 웹 서버 간 데이터 통신 기능을 수행하기 위한 프로토콜이라고 한다. (Saxce, Oprescu and Chen n.d.) HTTP는 HTTP/0.9로 처음 문서화 되었으며 이후 HTTP/1.0과 HTTP/1.1도 나오게 되었다. M. Belshe에 따르면 HTTP/1.0은 TCP 연결에서 한 개의 요청만 처리할 수 있다고 한다. HTTP/1.1이 파이프라이닝을 추가하였으나, 다중 요청 문제가 완벽히 해결되지 않았으며 Head Of Line (HOL) blocking의 문제도 있었다. 따라서 HTTP/1.0과 HTTP/1.1의 클라이언트는 TCP 연결을 동시에 설정하여 응답 시간을 줄여야 했다. 그러나 새로운 TCP 연결에서 여러 요청이 이루어지면 과도한 대기 시간이 발생할 수 있다. 게다가 HTTP의 헤더 필드가 반복적이고 장황해서 불필요한 네트워크 트래픽을 유발할 가능성이 컸다. (M.Belshe, BitGo, PeonR. 2015) 그러자 2009년 Google에서 실험용 프로토콜인 SPDY를 개발했다. SPDY는 HTTP/1.1의 문제점인 head of line blocking 문제를 해결하여 웹 페이지의 대기 시간을 줄였다. SPDY 프로토콜은 Google, Twitter, Facebook과 같은 대형 사이트들이 SPDY를 자사 인프라 내에 배포하며 급격히 성장하였고 이를 주시하던 HTTP Working Group(HTTP-WG)에서 SPDY를 사용하여 개선된 프로토콜인 HTTP/2를 만들고자 하였다. 그 후 몇 년 동안 SPDY와 HTTP/2가 함께 발전했으며 다중화, 헤더 압축 등의 기능을 포함한 효율적인 프로토콜이 되었다. (GrigorikIlya, Surma 2019)

**장점**

HTTP/2의 첫 번째 장점은 새로운 바이너리 프레이밍 계층을 사용하여 효율적으로 메시지를 처리한다는 것이다. HTTP/1.1은 텍스트 기반의 프로토콜이기 때문에 공백 라인을 일일이 찾아 프레임을 구분해야 하는 어려움이 있었다. 그러나 HTTP/2에서는 이진 데이터로 이루어진 프레임을 사용하기 때문에 구현이 간단해졌다.

두 번째 장점은 요청과 응답의 다중화이다. 앞서 말했듯이 HTTP/1.1은 다중 요청, 응답의 문제를 해결하지 못했다. 반면, HTTP/2의 새 바이너리 프레이밍 계층은 요청과 응답의 다중화를 지원한다. 구글의 Ilya Grigorik은 HTTP 메시지를 독립된 프레임으로 세분화하고, 이 프레임을 상호배치하여, 다른 쪽에서 다시 조립하도록 한다고 말한다.

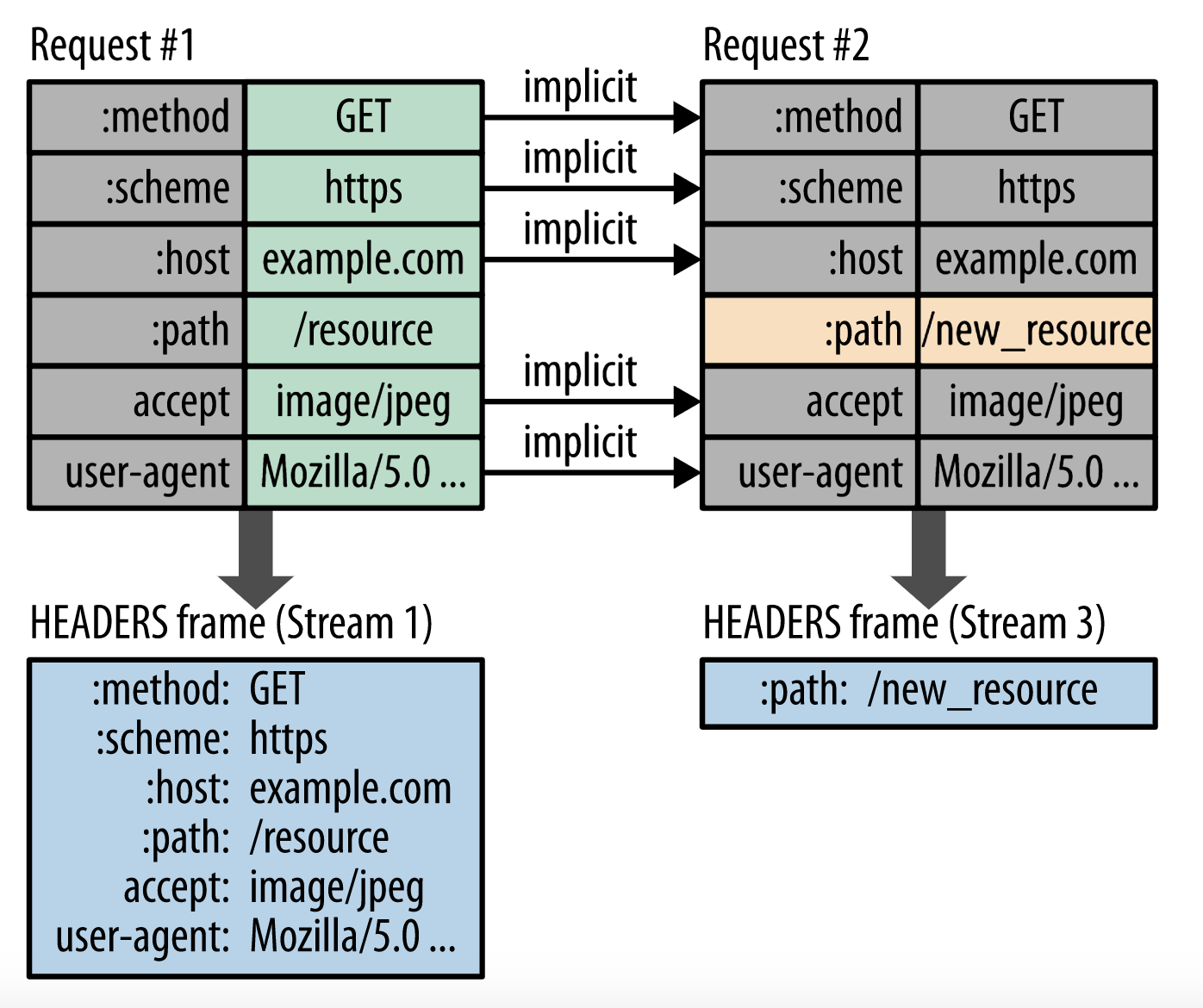


[그림1] 동일한 연결 내의 여러 스트림을 캡처한 것

[그림1]과 같이 HTTP 메시지를 세분화하여 클라이언트와 서버가 각각 스트림을 전송한다. 서버는 스트림1과 스트림3을, 서버는 스트림 5를 전송하기 때문에 3개의 병렬 스트림이 존재하게 된다. 여기서 스트림이란 구성된 연결 내에서 전달되는 바이트의 양방향 흐름으로, 하나 이상의 메시지가 전달될 수 있다. (GrigorikIlya, Surma 2019)

위와 같은 방법을 이용하여, HTTP/2는 여러 개의 요청과 응답 처리를 동시에 수행하여 애플리케이션의 속도를 높이고 배포 비용을 절감하였다.

세 번째 장점은 헤더를 압축했다는 것이다. 기존의 HTTP 헤더는 종종 같은 값을 전달했기 때문에 대기 시간이 늘어나고 오버헤드가 발생하는 문제가 있었다. HTTP/2에서는 HPACK 알고리즘을 사용하여 이러한 문제점을 해결했다. HPACK은 [그림2]와 같이 이전에 전송되었던 헤더 목록을 참고하여 중복을 검사한 후, 인덱스만 전송하여 효율적으로 헤더 정보를 재구성 할 수 있다.



[그림2] HPACK의 압축 방식

또한, HTTP/2는 헤더 필드를 정적 Huffman 코드로 인코딩한다. Huffman 코드란 문자들의 빈도를 사용하여 암호화하는 기법으로, 빈도가 낮을수록 긴 문자를 쓰고, 빈도가 높으면 짧은 문자를 사용한다. 이 방법을 사용함으로써, 개별 전송 크기를 줄일 수 있다.

**3. HTTP/2의 메시지 종류와 포맷**

HTTP/2의 종류는 크게 두 가지로, “요청 메시지”와 “응답 메시지”가 있다. HTTP/1.1과 요청 및 응답 헤더 필드의 정의는 변함이 없지만 [그림 2]에서 알 수 있듯이, 모든 헤더 필드 이름은 소문자이며 요청 라인이 :method, :scheme,

:authority, 그리고 :path와 같이 유사 필드로 대체된다.

M. Belshe의 논문에 따르면 HTTP/2의 메시지 포맷은 다음과 같다.

+-----------------------------------------------+

| Length (24) |

+---------------+---------------+---------------+

| Type (8) | Flags (8) |

+-+-------------+---------------+-------------------------------+

|R| Stream Identifier (31) |

+=+=============================================================+

| Frame Payload (0...) ...

+---------------------------------------------------------------+

[도형1] HTTP/2메시지의 일반 포맷

* Length : unsigned int 24-bit integer로 표현되는 frame payload의 길이이다.
* Type : 프레임의 유형으로 프레임의 형식과 의미를 결정하는 데 사용된다.

지정되어 있지 않은 유형을 가진 프레임이 오면 반드시 무시해야 한다.

* Flags : 지정된 프레임인지 아닌지 판단하기 위해 사용하는 Boolean

플래그이다. 지정되지 않은 프레임은 False 값(0x0)을 가진다.

* R : 이 비트의 의미는 정의되어 있지 않으며, 전송 시에 0x0 값을 가져야

한다. 수신할 때는 이 비트를 무시한다.

* Stream Identifier : 스트림 식별자이다. 값 0x0은 개별 스트림과 다르게 전체적으로 연결과 연관된 프레임에 예약되어 있다.

(M.Belshe, BitGo, PeonR. 2015)

**4.각 브라우저별로 HTTP/2의 지원 실태**

HTTP/2는 현재 다양한 브라우저에서 지원한다. 크롬, iOS용 크롬, 파이어폭스, 인터넷 익스플로러 11 (윈도 10에 한정), 마이크로소프트 엣지, 오페라, 사파리 9 등이 그 예이다.

**5. HTTP/2를 사용중인 웹사이트**

HTTP/2를 사용중인 대표적인 웹사이트는 다음과 같다.

1. youtube.com

2. facebook.com

3. duckduckgo.com

4. stackoverflow.com

5. twitter.com

6. reddit.com

7. en.wikipedia.org

8. instagram.com

9. vk.com

10. amazon.com

**6. 향후 전망**

HTTP/2는 고속 네트워킹 서비스를 제공하는 효율적인 프로토콜로서 발전 가능성이 아주 크다. 대기 시간도 단축하고 오버헤드를 줄이는 데 쉽기 때문에 웹은 물론 모바일 시장에서도 더욱 사용 범위가 넓어질 것으로 예상한다.

# 참고 문헌

GrigorikIlya, Surma. 2019. “Web Fundamentals | Google Developers.” 5월 29일. 액세스: 2019년 6월 4일. https://developers.google.com/web/fundamentals/performance/http2/?hl=en.

M.Belshe, BitGo, PeonR. 2015. “Internet Engineering Task Force (IETF).” 5. https://tools.ietf.org/html/rfc7540.

SaxcedeHugues, OprescuIuniana, ChenYiping. 날짜 정보 없음. “Is HTTP/2 Really Faster Than HTTP/1.1?” “18th IEEE Global Internet Symposium.” Issy-les-Moulineaux, France. 293.

날짜 정보 없음. “Wappalyzer.” 액세스: 2019년 6월 4일. https://www.wappalyzer.com/technologies/http-2.