HTTP 2.0
: The New Web Standard and Issue

허태성
saturn.twinfish@gmail.com

목차

I. HTTP 2.0의 등장 배경

HTTP 2.0 등장 배경

- 15년간 웹 전송기술 표준 유지

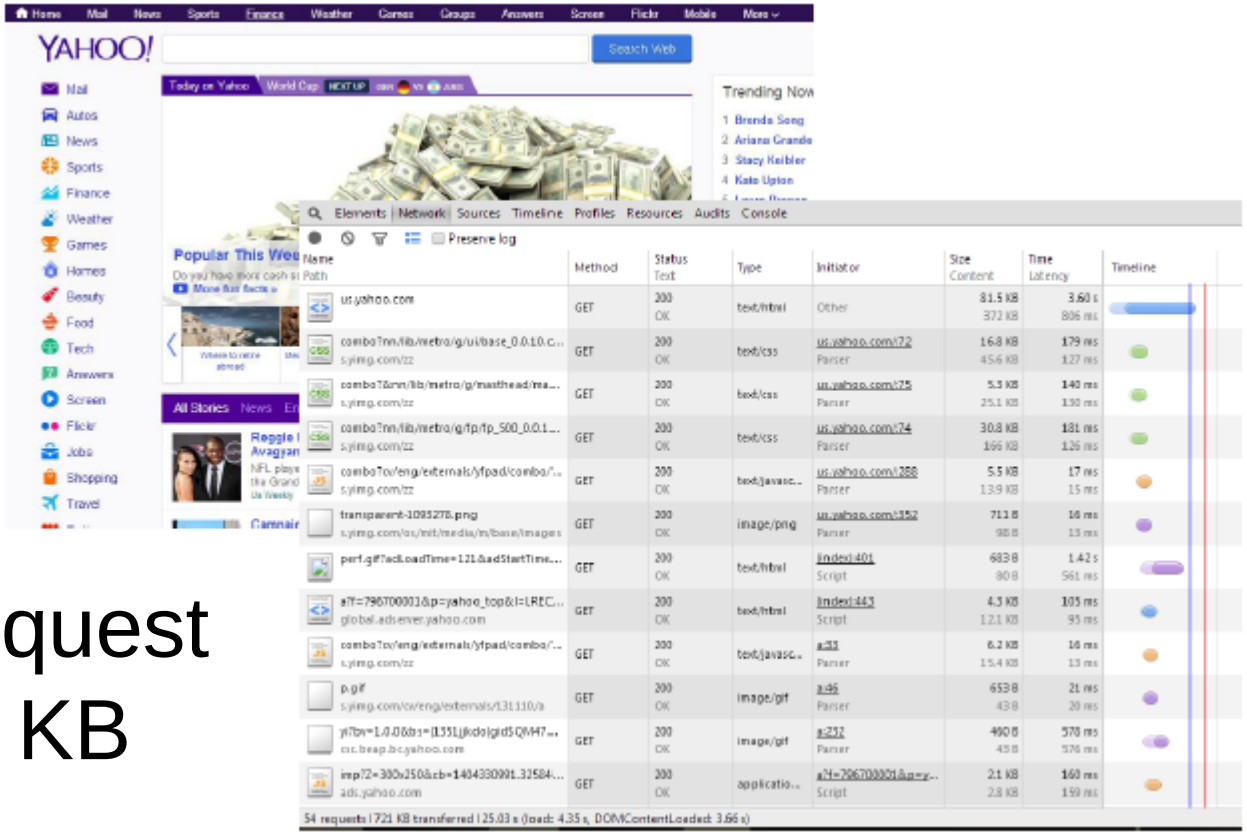
HTTP 2.0 등장 배경

- 웹페이지 @ 1996년

3 Request
34 KB

HTTP 2.0 등장 배경

- 웹페이지 @ Today



54 Request
721 KB

HTTP 2.0 등장 배경

- 웹페이지 & 전송속도의 변화

1996		2015
3 Request 34 KB	Resource **20X**	54 Request 721 KB
56 Kbps	Speed **100X**	4 Mbps

변화에 따른 새로운 전송 기술이 필요

HTTP 2.0 등장 배경

Google
"Let's make a web faster"

IETF

SPDY Protocol 발표
(draft #1, 2009.11)

SPDY 표준화 시작
(draft #3, 2012.2)

SPDY 개정
(draft #3.1, 2013.9)

SPDY 보완
(draft #4 alpha2, 2013.11)

httpbis WG 생성
"기존 문제점 보완"
(2007.10)

"HTTP 2.0 개발" 천명
(2012.6)

HTTP 2.0 drfat-00
(2012.11~)

HTTP 2.0 작업 완료
(2014.11)

HTTP 2.0 표준 승인
(2015.2)

HTTP 2.0 표준 발표
(RFC7540, 2015.5)

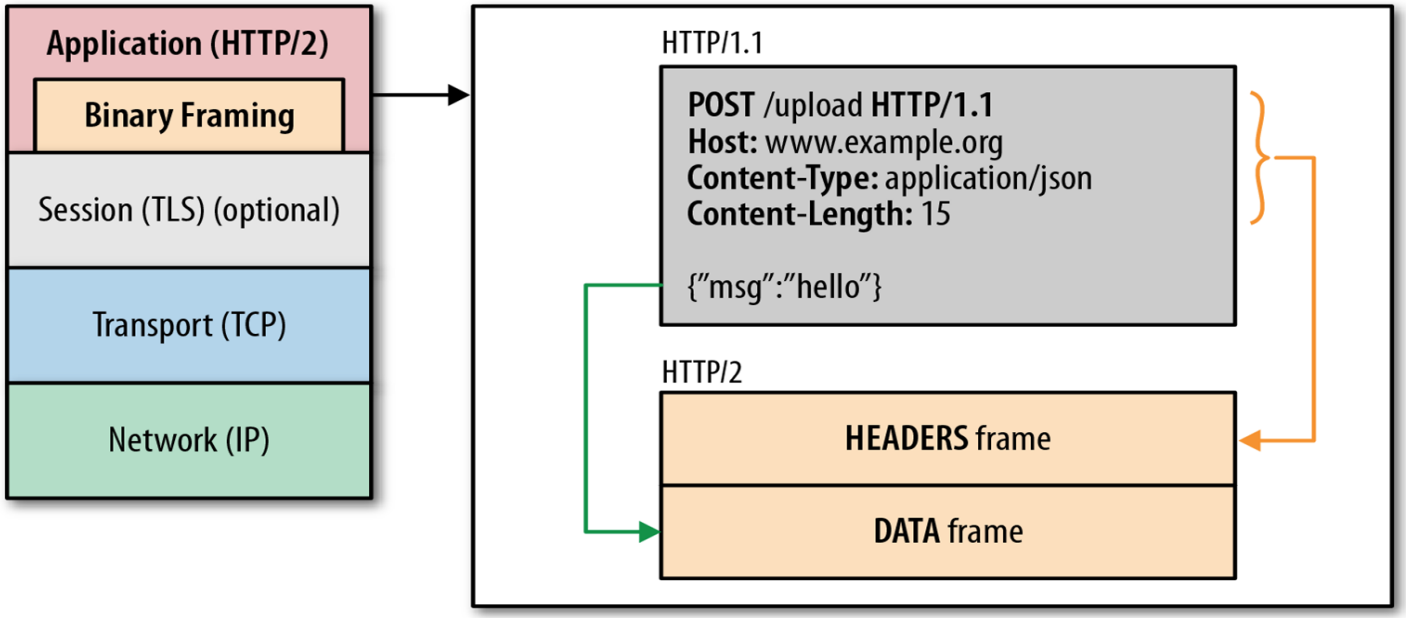
II. HTTP 2.0 특징

HTTP 2.0 특징

바이너리 프로토콜	– 텍스트가 아닌 Binary 프레임으로 구성 → 파싱이 더 빠르고, 오류 발생 가능성이 낮음
Multiplexing	– 하나의 TCP Connection내에서 다수의 Stream을 생성 – 하나의 요청이 지연되면 나머지 응답이 늦어지는 HTTP Pipelining과는 달리 각각의 요청/응답을 독립적으로 처리 → 다수의 요청/응답을 동시에 처리가능
헤더 압축	– 반복적으로 사용되는 헤더를 헤더 테이블내의 인덱스로 표기 → 헤더 크기를 80% 정도 줄임
우선순위 설정	– Stream별로 우선 순위를 지정 → 중요한 리소스의 처리 지연을 방지
Server Push	– 클라이언트가 요청하지 않아도 필요가 예상되는 리소스를 서버가 미리 전송

HTTP 2.0 특징

- 바이너리 프로토콜



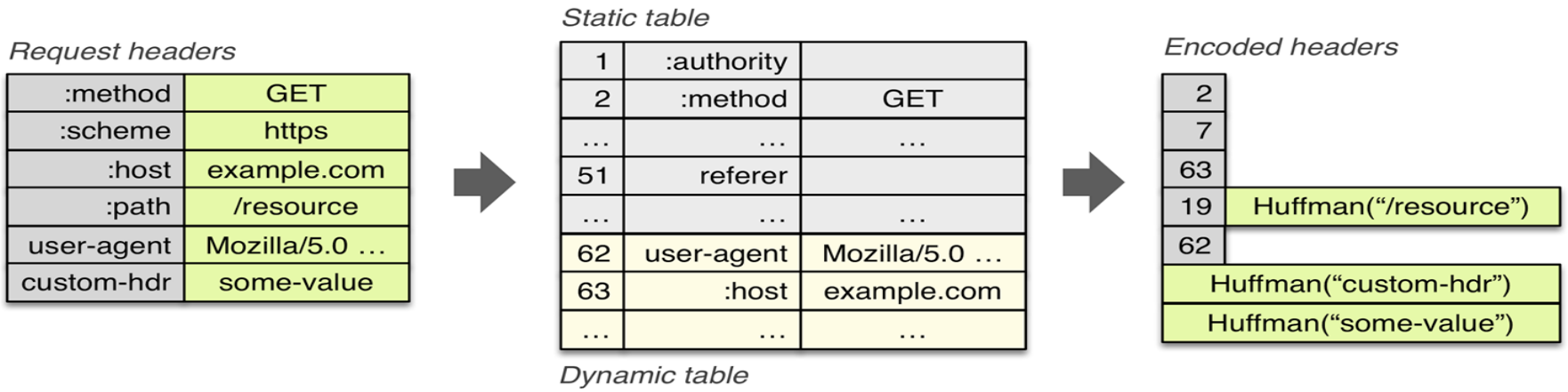
Bit	+0..7	+8..15	+16..23	+24..31
0	Length		Type	Flags
32	R	Stream Identifier		
...	*Frame Payload*			

HTTP 2.0 특징

- Multiplexing
  - 프레임마다 관련된 요청 순서 번호(Stream #)를 기재
  - 요청 순서에 상관없이 데이터 전송 가능
  - 하나의 TCP 세션 사용

HTTP 2.0 특징

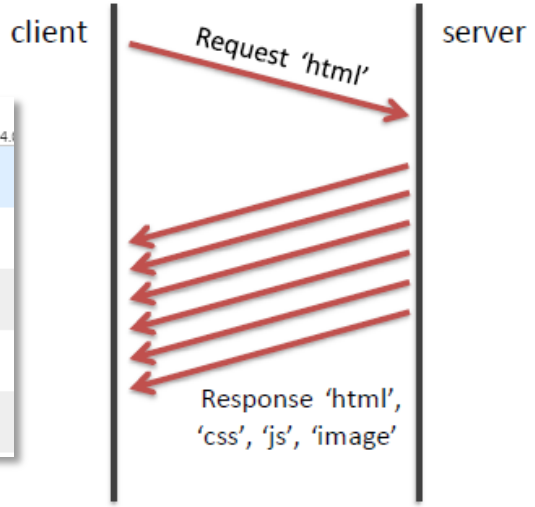
- Header Compression
  - 이전 헤더 내용을 테이블에 저장
  - 다음 request에서 동일한 헤더를 사용할 경우 테이블의 Index 번호만을 사용



HTTP 2.0 특징

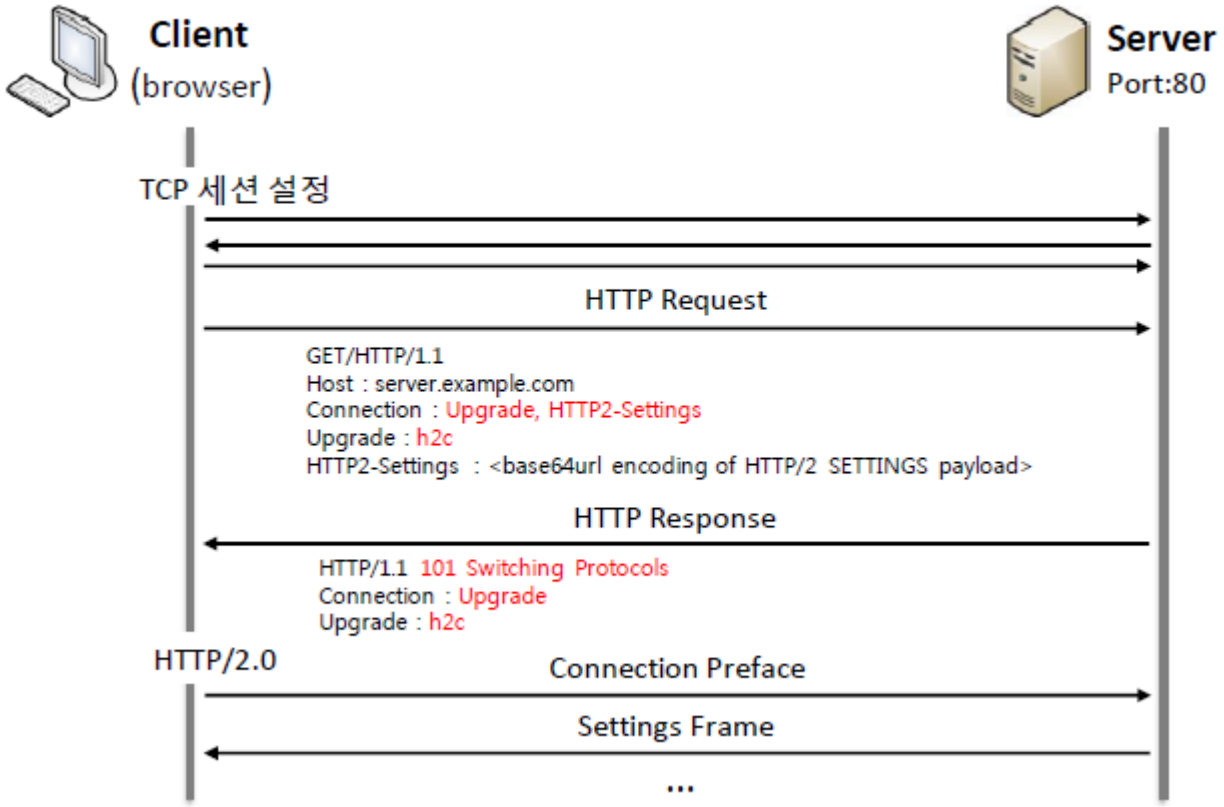
- Server Push
  - Client가 요청하기 전에 필요가 예상되는 리소스를 Server가 먼저 전송
  - Server 쪽에서의 추가적인 프로그래밍이 필요

Name / Path	Method	Status / Text	Domain	Type	Size / Content	Time / Latency	Timeline
www.naver.com	GET	200 OK	www.naver.com	document	19.7 KB 80.5 KB	43 ms 36 ms	
main_v20150603.css s.pm.naver.net/css	GET	200 OK	s.pm.naver.net	stylesheet	20.1 KB 114 KB	1.62 s 1.62 s	
api_atcmp_0319.css sstatic.naver.net/search/css/2015	GET	200 OK	sstatic.naver.net	stylesheet	4.8 KB 23.7 KB	1.62 s 1.62 s	
nlog_20140205.min.js s.pm.naver.net/js/c	GET	200 OK	s.pm.naver.net	script	5.6 KB 14.8 KB	1.94 s 1.94 s	
nmms_224940510.gif img.naver.net/static/www/u/2013/0731	GET	200 OK	img.naver.net	gif	4.6 KB 4.3 KB	1.94 s 1.94 s	



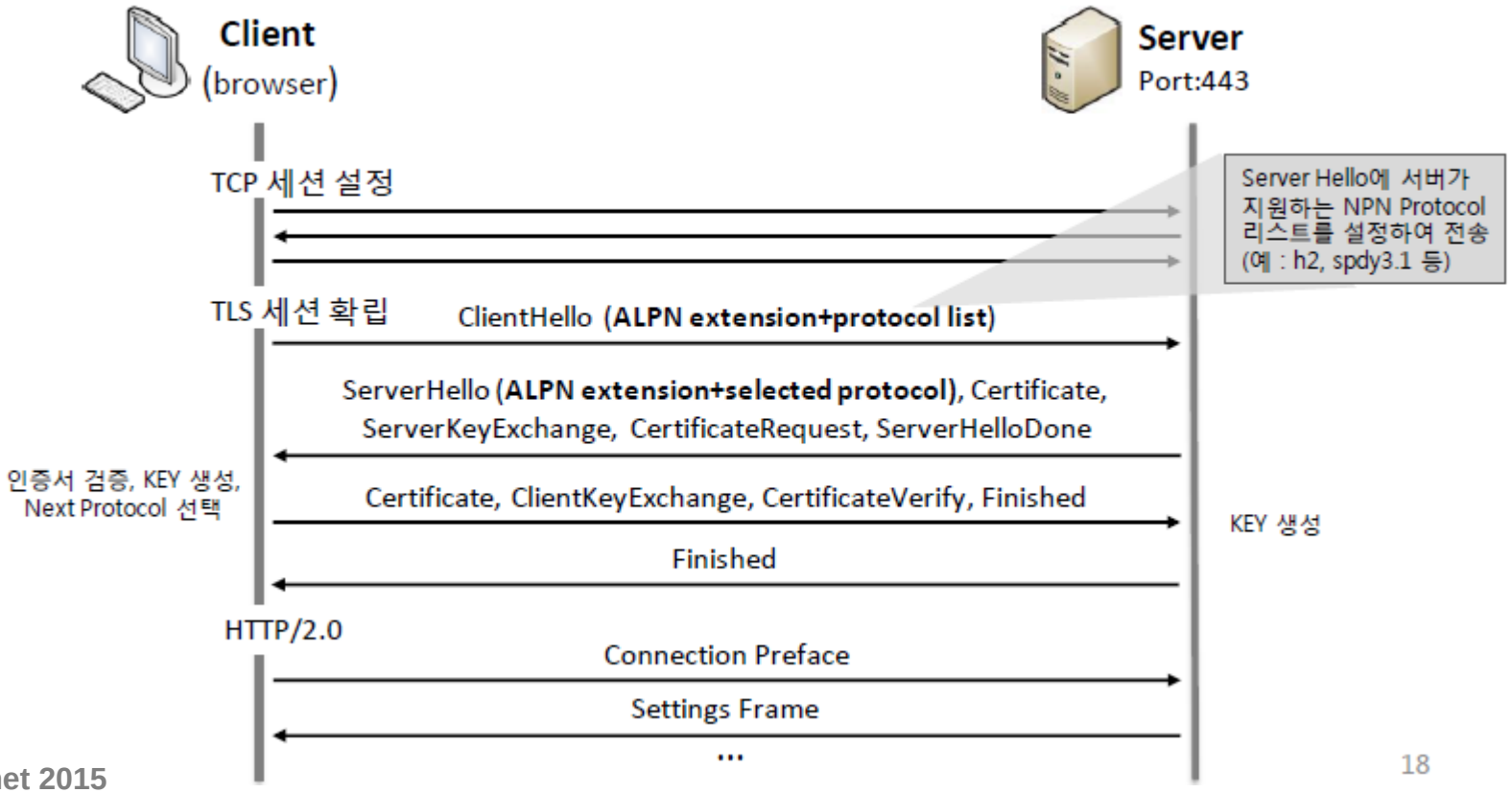
HTTP 2.0 특징

- Client ~ Server 연결 (http)
  - HTTP Request시, Upgrade 헤더 필드에 “h2c”를 명시
  - HTTP/2.0 지원시, “101 Switching Protocols” 응답



HTTP 2.0 특징

- Client ~ Server 연결 (https)
  - TLS-ALPN(Application Layer Protocol Negotiation) 사용
  - HTTP 2.0 지원시 프로토콜 리스트에 “h2”를 표기

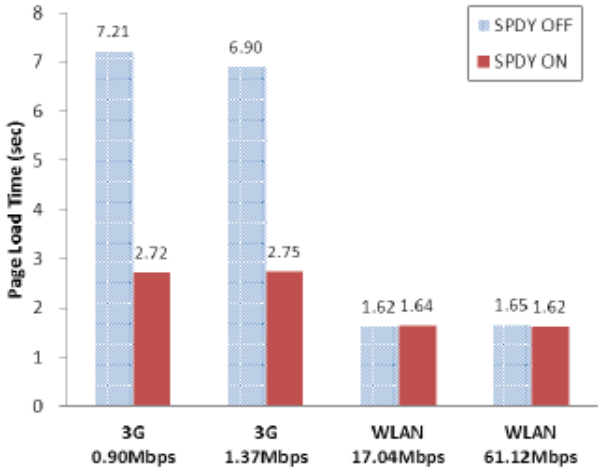


III. HTTP 2.0 기대 효과

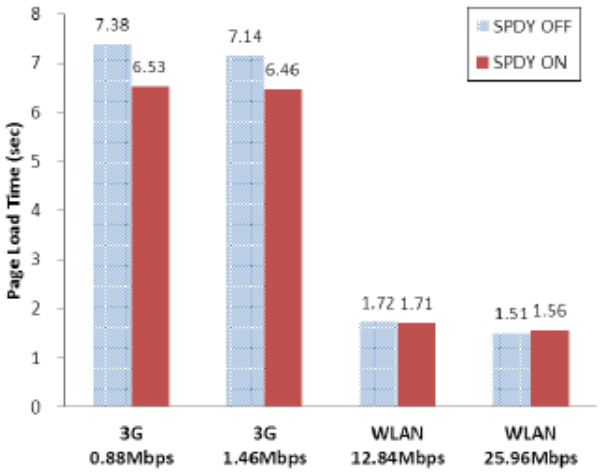
HTTP 2.0 기대 효과

- 3G / 무선랜 환경 (https = h2)
  - 3G에서는 10~60%의 로딩시간 단축 예상
  - 무선랜 환경에서는 효과 미미

✓ Google play 사이트에 대한 성능 측정 결과 (평균 RTT 33ms)
✓ 웹 페이지의 크기는 1.5MB이며 100번 이상 반복 측정을 실시
✓ 크롬의 경우, 3G 환경에서는 60% 정도의 성능향상을 보였으나 무선랜 환경에서는 차이가 미미함
✓ 파이어폭스의 경우, 3G 환경에서는 10%정도 성능향상을 보이지만 무선랜은 그 차이가 미미하거나 성능이 오히려 더 나빠짐



< Chrome 브라우저 >

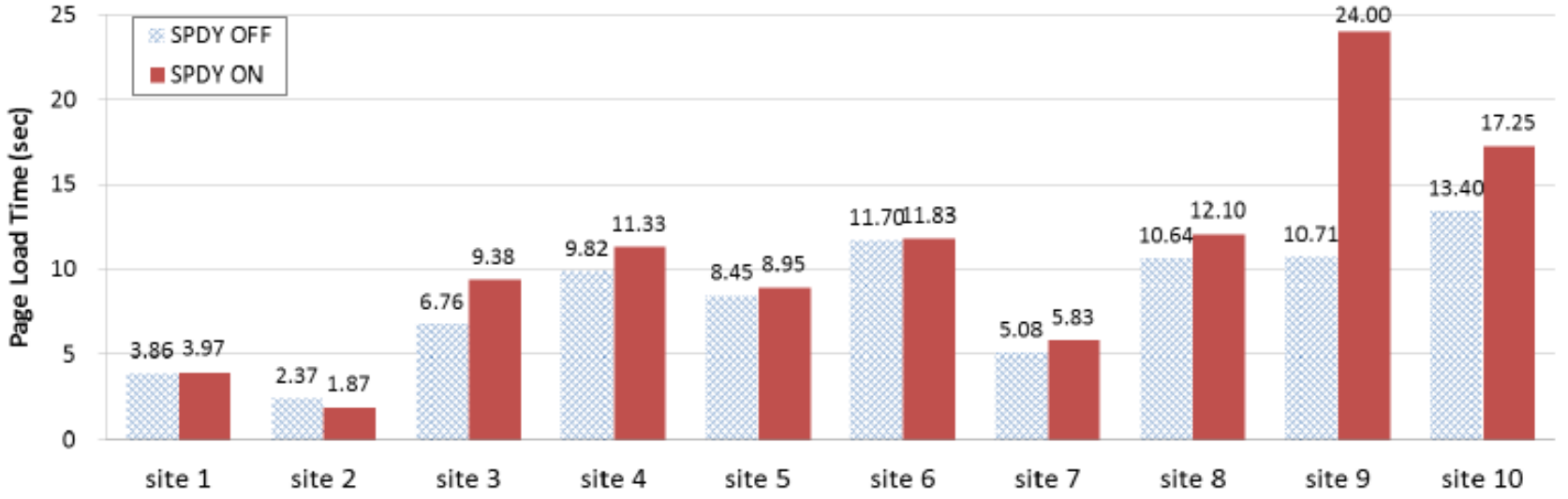


< Firefox 브라우저 >

HTTP 2.0 기대 효과

- 3G 환경 (https = h2)
  - 로딩시간 다소 증가 (최대 2배)

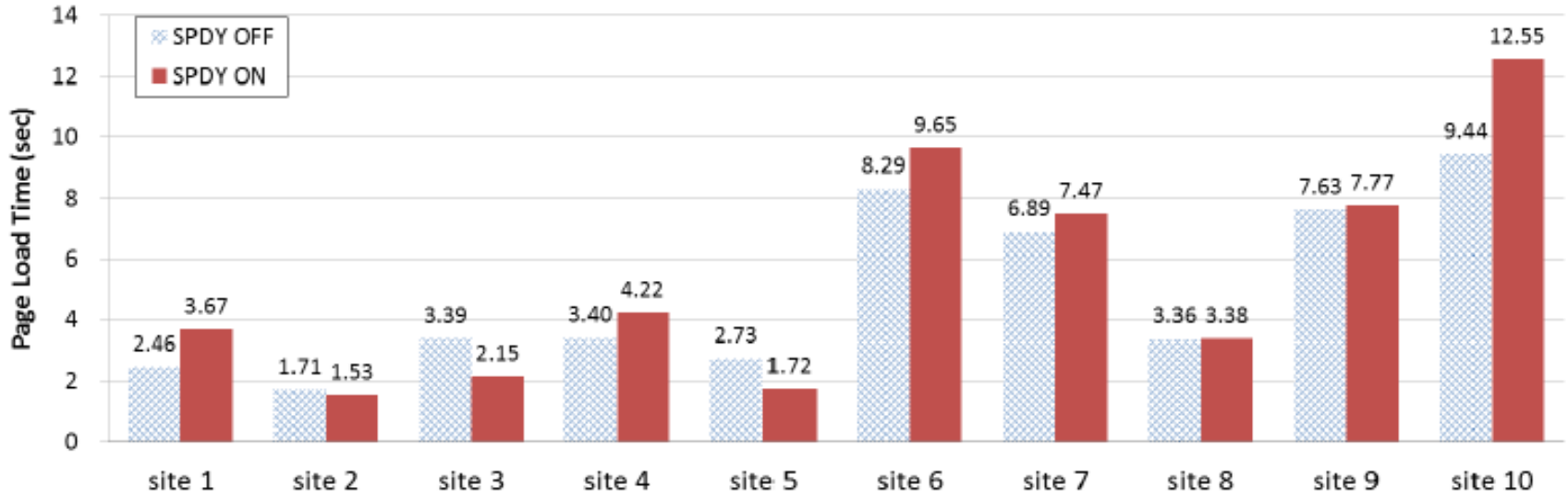
✓ 3G 환경에서 측정한 10개 사이트의 초기 접속 시간 (Chrome 브라우저에서 측정)
✓ 41KB ~ 600KB 까지 비교적 페이지의 크기가 작은 10개 사이트에 대해 성능 측정을 진행
✓ 두 번째 사이트를 제외하고 모든 경우 SPDY 사용 시 많게는 두 배 이상 페이지 로드 타임이 증가



HTTP 2.0 기대 효과

- 무선랜 환경 (https = h2)
  - 일부 사이트에서는 로딩시간이 다소 증가 (1~49%)

✓ WLAN 환경에서 측정한 10개 사이트의 초기 접속 시간 (Chrome 브라우저에서 측정)
✓ 700KB ~ 5MB 까지 비교적 페이지의 크기가 큰 10개 사이트에 대해 측정 진행
✓ 두 번째, 세 번째, 다섯 번째를 제외한 일곱개의 사이트에서는 SPDY 사용 시 페이지 로드타임이 1~49%까지 증가함



HTTP 2.0 기대 효과

- LTE 환경 (https = h2)
  - 전송 데이터량은 감소하나 로딩 시간은 오히려 증가 (약 50%)

✓ 단말의 Chrome 브라우저에서 LTE 환경에 대한 SPDY 성능 측정을 진행
✓ SPDY 사용 시 데이터 압축으로 인해 전송 데이터의 크기는 약 50% 감소
✓ SSL세션 확립 절차, RTT 증가 등으로 인해 전체 페이지 로드 타임은 오히려 증가함

		다음 www.daum.net	네이트 www.nate.com	네이버 www.naver.com
SPDY OFF	페이지로드 타임(sec)	1.49	3.08	0.95
	전체 시간(sec)	3.00	4.17	5.19
	Request 수	55	78	57
	웹페이지 크기(KB)	1331.20	830.33	1331.20
SPDY ON	페이지로드 타임(sec)	2.25	4.01	1.50
	전체 시간(sec)	4.96	5.26	5.89
	Request 수	55	77	58
	웹페이지 크기(KB)	609.33	437.33	701.00
ON/OFF 시 비교	페이지로드 타임(sec)	50%	30%	58%
	웹페이지 크기(KB)	-54%	-47%	-47%

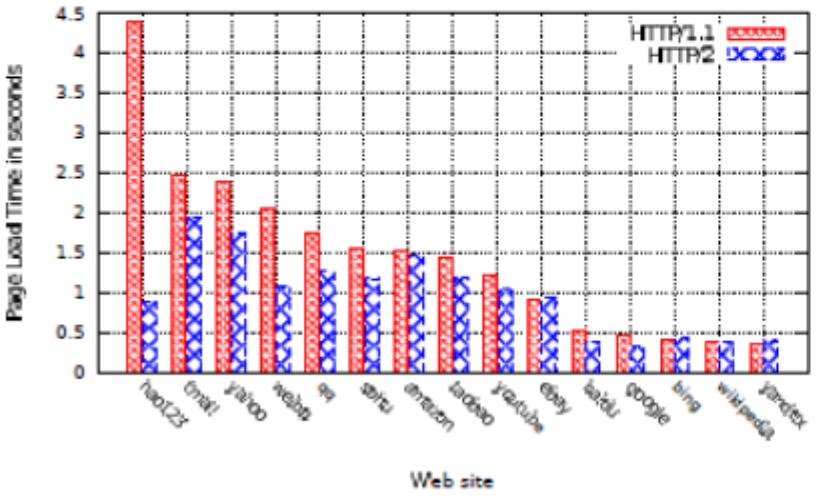
* SPDY 사용 시 증가 SPDY 사용 시 감소

HTTP 2.0 기대 효과

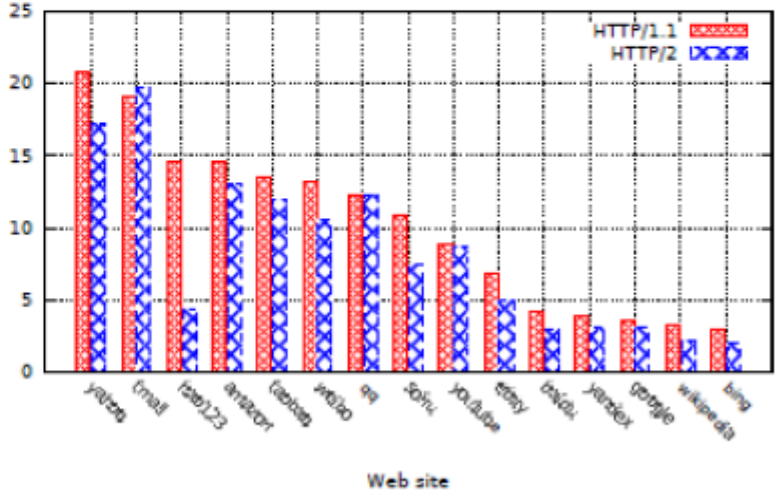
- ADSL / 3G 환경 (http = h2c)
  - 로딩시간 단축 (20~48%)
  - 웹페이지 크기가 작을 경우 효과 미미

✓ ADSL 환경
- 간단한 웹 페이지의 경우, HTTP/2.0 사용 시의 성능 개선이 미미함
- 웹 페이지의 크기가 클 경우, 20~48% 까지 페이지 로드타임의 감소를 보임

✓ 3G 네트워크 환경
- HTTP/2.0 사용 시, 페이지 로드 타임은 평균 20%감소



<Page load time with an ADSL Livebox, 50ms latency>



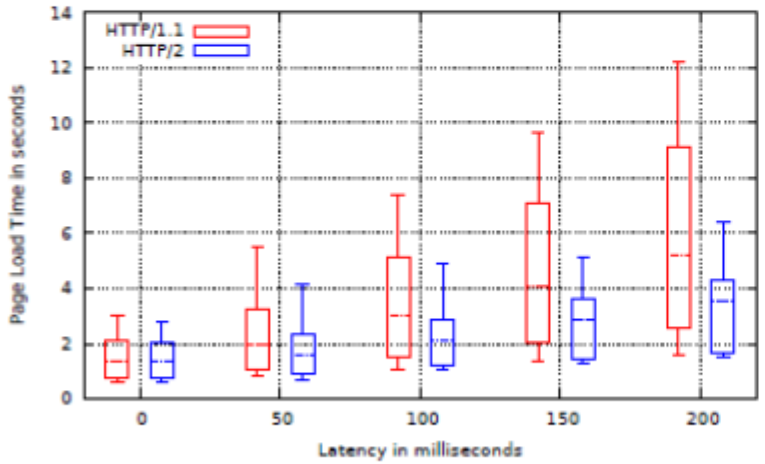
<Page load time with a 3G modem, 400ms latency>

HTTP 2.0 기대 효과

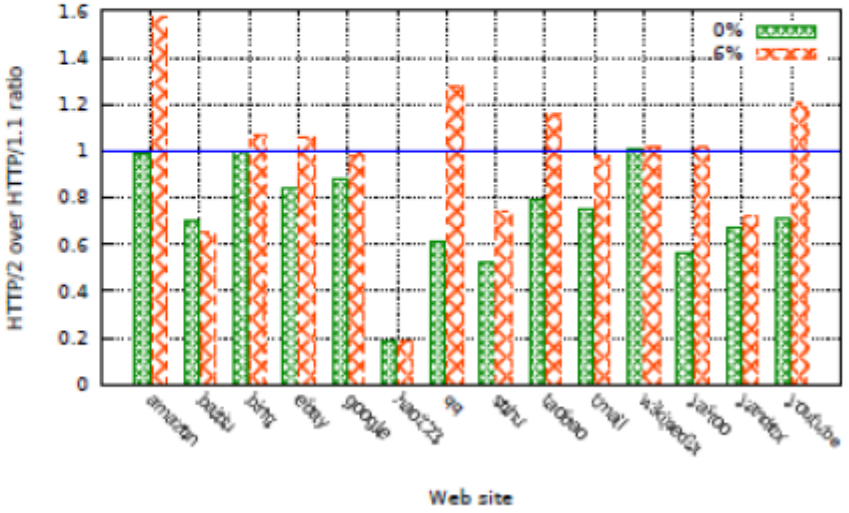
- ADSL / 3G 환경 (http = h2c)
  - Latency가 큰 네트워크에서 성능 개선 효과
  - Packet Loss 발생시 성능 저하

✓ Impact of Latency
- Latency 발생 시, HTTP/2.0 을 사용할 때의 성능이 더 좋음

✓ Impact of Packet Loss
- HTTP/2.0은 하나의 TCP 커넥션에서 다수의 Stream을 처리하므로, Packet Loss 발생 시 성능이 저하 됨



<Impact of latency, 0% loss>



<Impact of packet loss, 100ms latency>

Ⅳ. 현황 및 이슈

현황 및 이슈

- HTTP 2.0 지원 현황 (Client / Server)

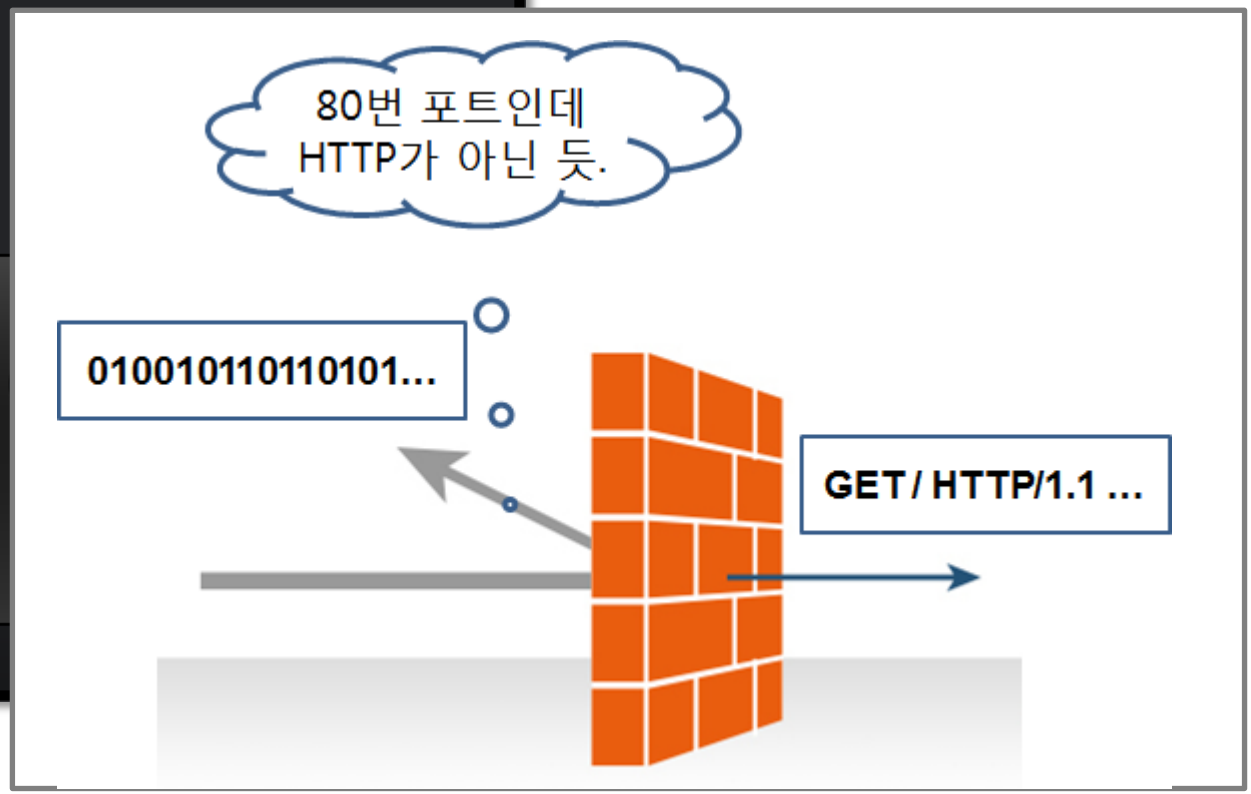
Client (Browser)			Server		
제품	제품 버전	HTTP 2.0 버전	제품	제품 버전	HTTP 2.0 버전
IE	11(Win10) Edge	h2-14	~~Apache~~		
Chrome	41+	h2-14 ~ h2	Nginx	2015년말	
Firefox	34+	h2-14 ~ h2	IIS	10 (Win10)	h2-14
~~Safari~~			~~Varnish~~		

(위 정보에는 오류가 있을 수 있습니다. 정확한 현황은 해당 제조사를 통해 확인하세요.)

https://github.com/http2/http2-spec/wiki/Implementations http://caniuse.com/#feat=http2

현황 및 이슈

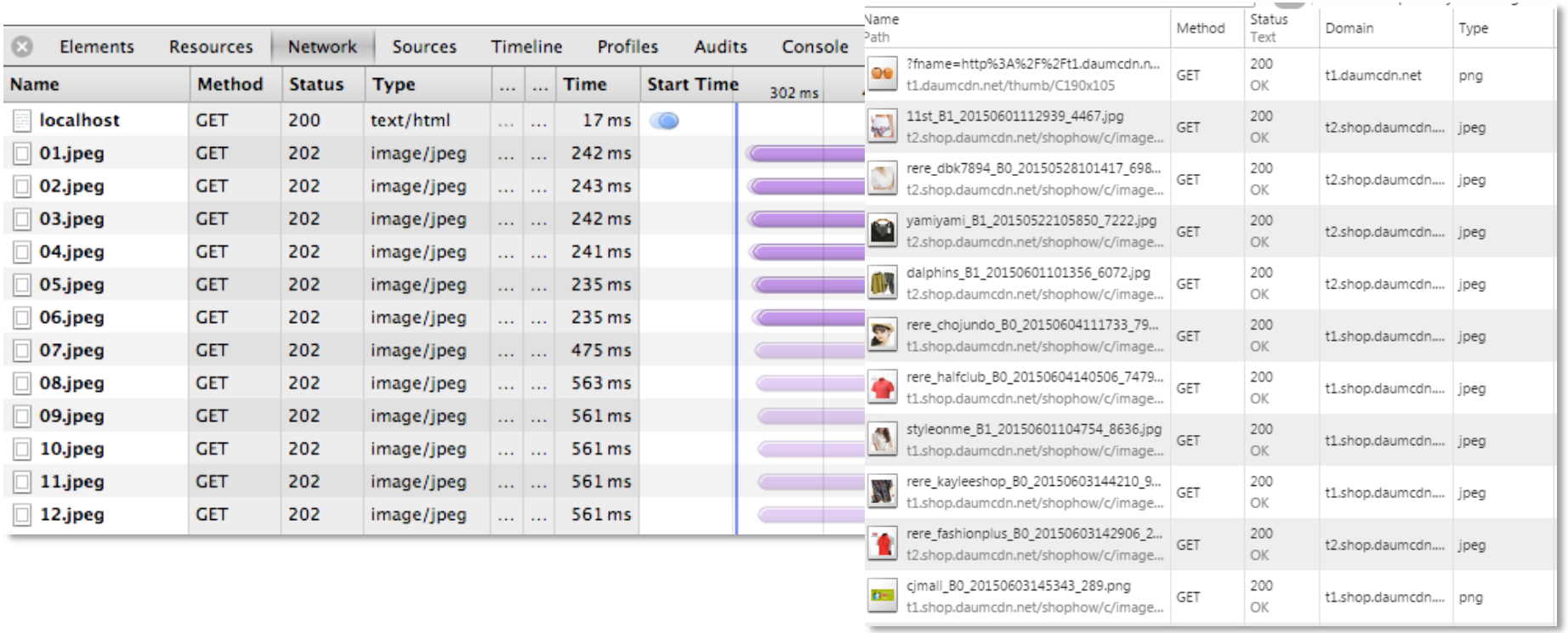
- Middle Box
  - Firewall, Proxy, CDN 등 Middle Box의 HTTP 2.0 트래픽 처리
  - 암호화된 h2 보다 암호화 되지 않은 h2c에서 오류 가능성 ↑



현황 및 이슈

- 기존 웹 전송 최적화 기법 재고(再考)
  - ① Domain Sharding
    - 서버당 최대 TCP 연결 개수 제한 : IE11(13개)을 제외한 대부분 6개

    → 다른 FQDN을 가지는 서버 여러 개로 분산 처리



Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console

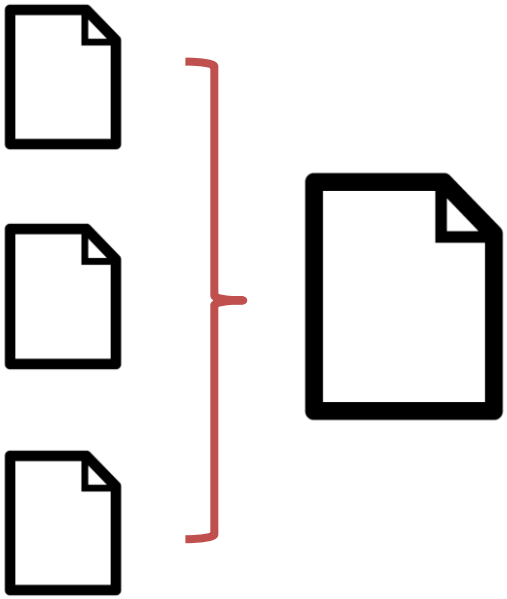
Name	Method	Status	Type	...	...	Time	Start Time (302 ms)
localhost	GET	200	text/html	...	...	17 ms	
01.jpeg	GET	202	image/jpeg	...	...	242 ms	
02.jpeg	GET	202	image/jpeg	...	...	243 ms	
03.jpeg	GET	202	image/jpeg	...	...	242 ms	
04.jpeg	GET	202	image/jpeg	...	...	241 ms	
05.jpeg	GET	202	image/jpeg	...	...	235 ms	
06.jpeg	GET	202	image/jpeg	...	...	235 ms	
07.jpeg	GET	202	image/jpeg	...	...	475 ms	
08.jpeg	GET	202	image/jpeg	...	...	563 ms	
09.jpeg	GET	202	image/jpeg	...	...	561 ms	
10.jpeg	GET	202	image/jpeg	...	...	561 ms	
11.jpeg	GET	202	image/jpeg	...	...	561 ms	
12.jpeg	GET	202	image/jpeg	...	...	561 ms	

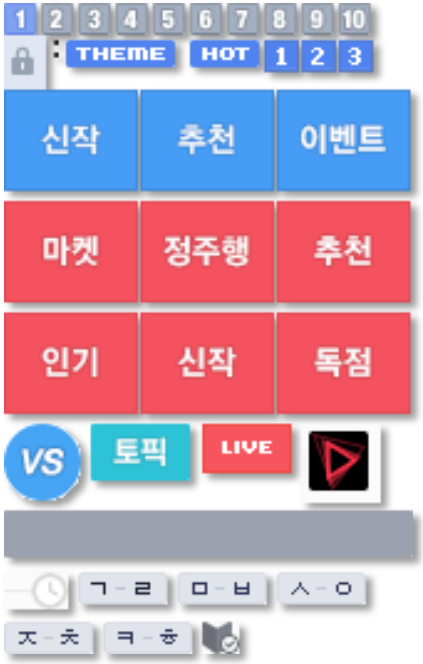
Name / Path	Method	Status Text	Domain	Type
?fname=http%3A%2F%2Ft1.daumcdn.n... t1.daumcdn.net/thumb/C190x105	GET	200 OK	t1.daumcdn.net	png
11st_B1_20150601112939_4467.jpg t2.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t2.shop.daumcdn....	jpeg
rere_dbk7894_B0_20150528101417_698... t2.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t2.shop.daumcdn....	jpeg
yamiyami_B1_20150522105850_7222.jpg t2.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t2.shop.daumcdn....	jpeg
dalphins_B1_20150601101356_6072.jpg t2.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t2.shop.daumcdn....	jpeg
rere_chojundo_B0_20150604111733_79... t1.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t1.shop.daumcdn....	jpeg
rere_halfclub_B0_20150604140506_7479... t1.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t1.shop.daumcdn....	jpeg
styleonme_B1_20150601104754_8636.jpg t1.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t1.shop.daumcdn....	jpeg
rere_kayleeshop_B0_20150603144210_9... t1.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t1.shop.daumcdn....	jpeg
rere_fashionplus_B0_20150603142906_2... t2.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t2.shop.daumcdn....	jpeg
cjmall_B0_20150603145343_289.png t1.shop.daumcdn.net/shophow/c/image...	GET	200 OK	t1.shop.daumcdn....	png

"HTTP 2.0은 하나의 TCP 연결에서 Multiplexing이 가능하므로 불필요"

현황 및 이슈

- 기존 웹 전송 최적화 기법 재고(再考)
  - ② Concatenated Assets / Inline Image
    - Request 오버헤드를 줄이기 위해 작은 리소스를 하나로 합침





"HTTP 2.0은 하나의 TCP 연결에서 Multiplexing이 가능하여 Request가 많아도 오버헤드가 없으므로 불필요"

현황 및 이슈

- https (h2) !!!
  - 주요 브라우저는 HTTP 2.0에서 http(h2c) 미지원
  - HTTP2.0 기반 서비스를 위해서는 https only 서비스로 전환 필요

Internet Architecture Board

Home About Activities Documents Liaisons Appeals

← Reappointment of Lars Eggert as IRTF Chair

IAB Seeks Feedback on ICANN

IAB Statement on Internet Confidentiality

Posted on November 14, 2014 by Cindy Morgan

In 1996, the IAB and IESG recognized that the growth of the Internet depended on users having confiden protect their private information. RFC 1984 documented this need. Since that time, we have seen eviden activities of attackers are greater and more pervasive than previously known. The IAB now believes it is i designers, developers, and operators to make encryption the norm for Internet traffic. Encryption should b possible, but even protocols providing confidentiality without authentication are useful in the face of pervas described in RFC 7258.

W3C

Securing the Web

W3C TAG Finding 22 January 2015

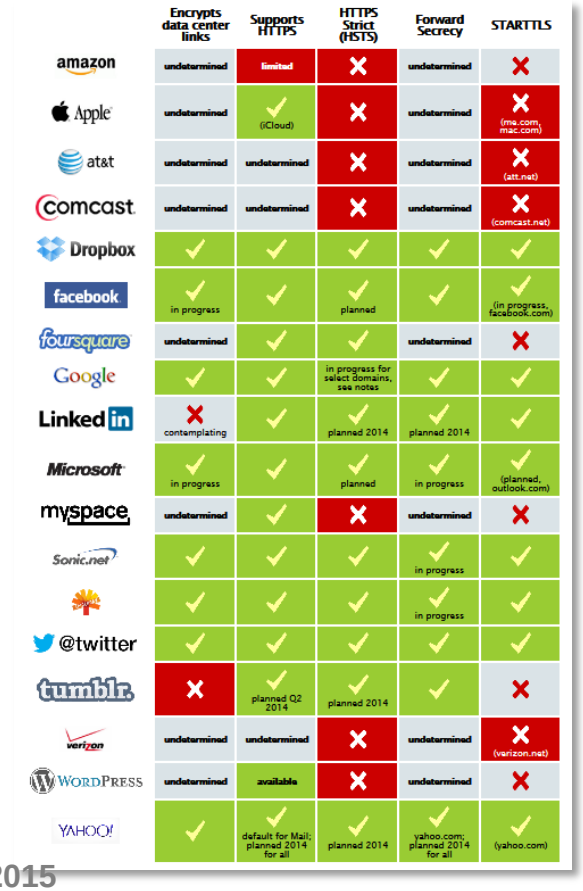
Latest editor's draft:
https://w3ctag.github.io/web-https/
Editor:
Mark Nottingham
Participate:
File a bug.
Commit history.
Mailing list.

Abstract

This finding documents the TAG's position on securing the Web through the use of cryptography, aid in its use.

현황 및 이슈

- Ready to encrypt ?
  - 인증서, AJAX Mesh-up, 광고…
  - 모든 요소들이 동시에 https ready 상태가 되어야 함



	Encrypts data center links	Supports HTTPS	HTTPS Strict (HSTS)	Forward Secrecy	STARTTLS
amazon	undetermined	limited	✗	undetermined	✗
Apple	undetermined	✓ (iCloud)	✗	undetermined	✗ (me.com, mac.com)
at&t	undetermined	undetermined	✗	undetermined	✗ (att.net)
comcast	undetermined	undetermined	✗	undetermined	✗ (comcast.net)
Dropbox	✓	✓	✓	✓	✓
facebook	✓ in progress	✓	✓ planned	✓	✓ (in progress, facebook.com)
foursquare	undetermined	✓	✓	undetermined	✗
Google	✓	✓	in progress for select domains, see notes	✓	✓
LinkedIn	✗ contemplating	✓	✓ planned 2014	✓ planned 2014	✓
Microsoft	✓ in progress	✓	✓ planned	✓ in progress	✓ (planned, outlook.com)
myspace	undetermined	✓	✗	undetermined	✗
Sonic.net	✓	✓	✓	✓ in progress	✓
	✓	✓	✓	✓ in progress	✓
@twitter	✓	✓	✓	✓	✓
tumblr.	✗	✓ planned Q2 2014	✓ planned 2014	✓	✗
verizon	undetermined	undetermined	✗	undetermined	✗ (varizon.net)
WordPress	undetermined	available	✗	undetermined	✗
YAHOO!	✓	✓ default for Mail; planned 2014 for all	✓ planned 2014	✓ yahoo.com; planned 2014 for all	✓ (yahoo.com)

VS

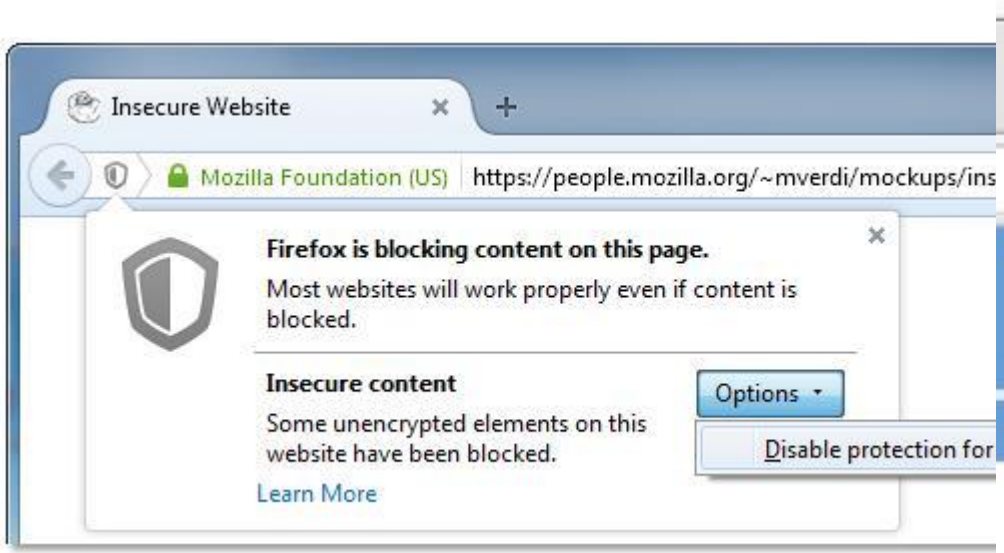
NAVER DAUM NATE TISTORY

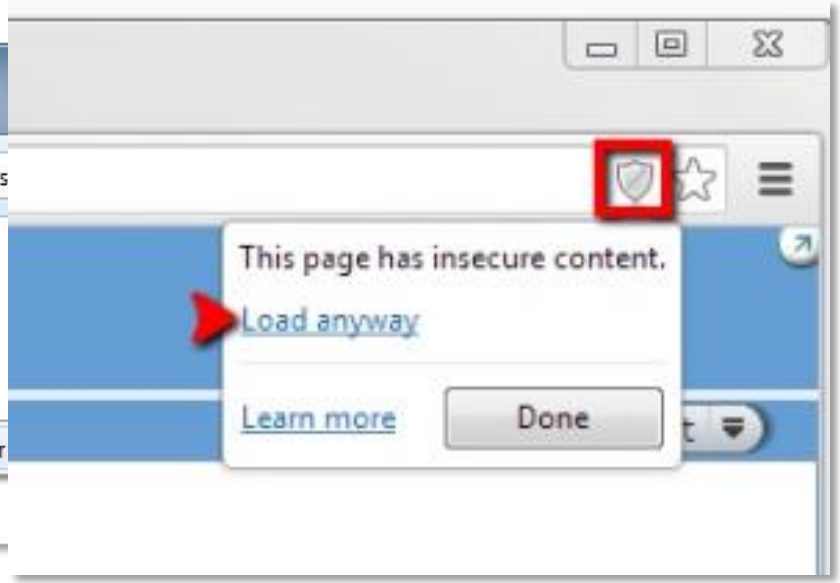
UPGRADE YOUR LIFE 11ST 11번가

?

현황 및 이슈

- 주요 브라우저가 h2 처리시 엄격한 정책을 적용
  - Mixed Content Blocking
    - h2(https) 기반 페이지 로딩시 페이지내에 http 기반 리소스가 포함 될 경우 해당 리소스를 로딩하지 않음

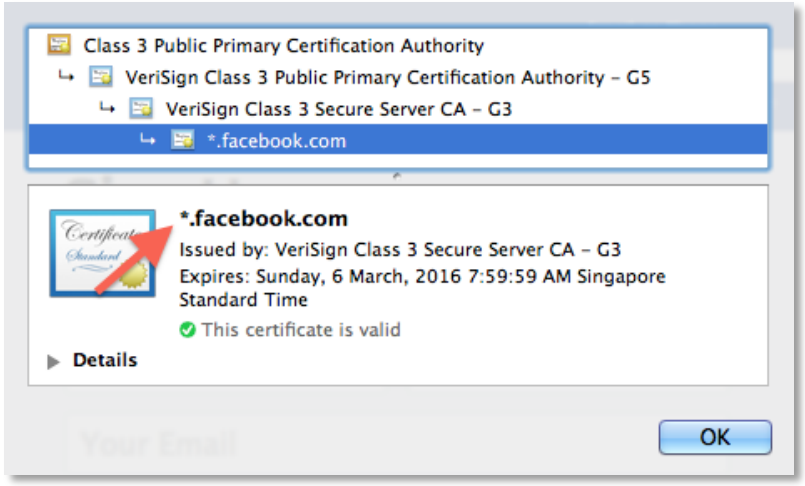




현황 및 이슈

- 인증서 비용 및 관리
  - 페이지내에 포함된 리소스 관련 모든 호스트가 인증서를 필요
  - 페이지당 평균 17개 도메인

1) 17개의 인증서
2) n개의 멀티도메인 **개당 약 1.7배 가격**
3) n개의 와일드카드 인증서 **약 7배 가격**

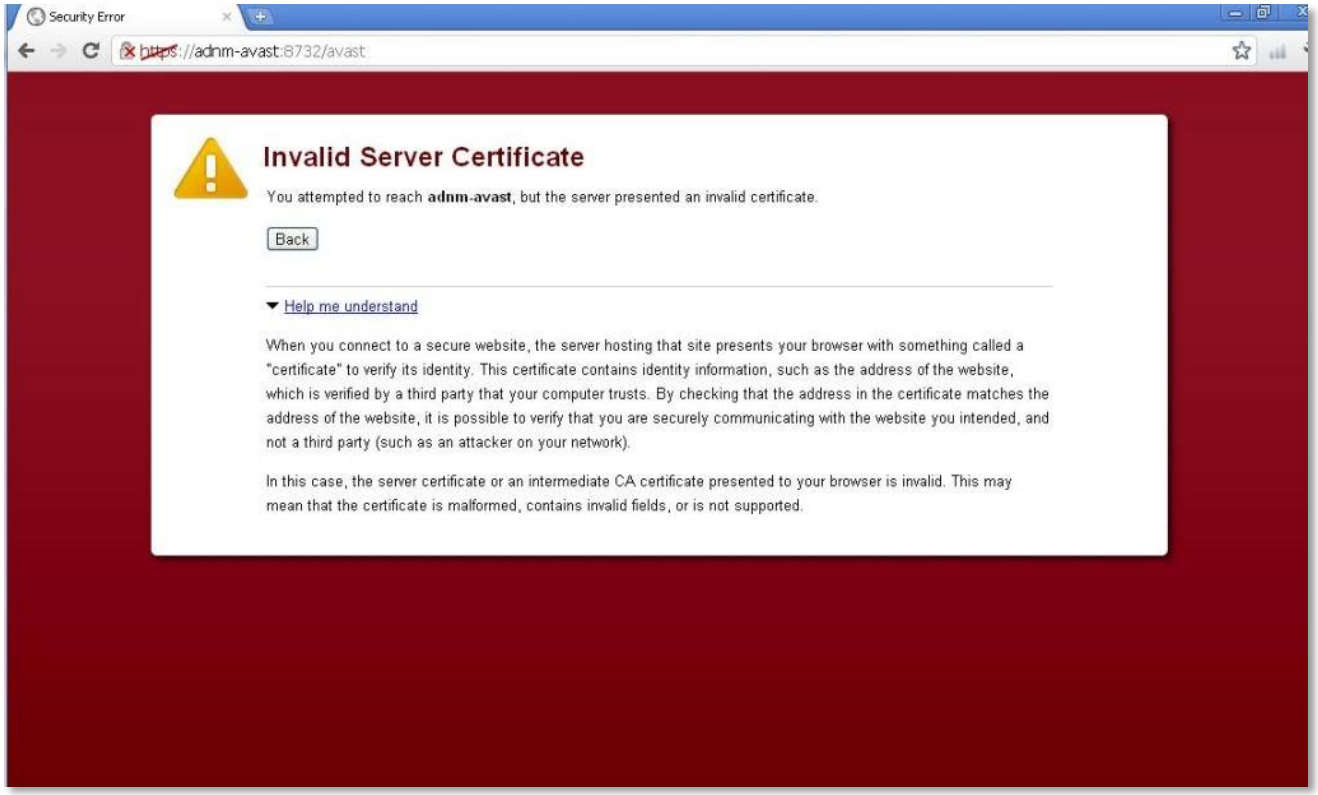


"NO Domain Sharding !!!"

현황 및 이슈

- 사설 인증서 사용?
  - 주요 브라우저가 h2 처리시 엄격한 정책을 적용
  - 사설 인증서 유효성 확인이 안될 경우 접속 차단

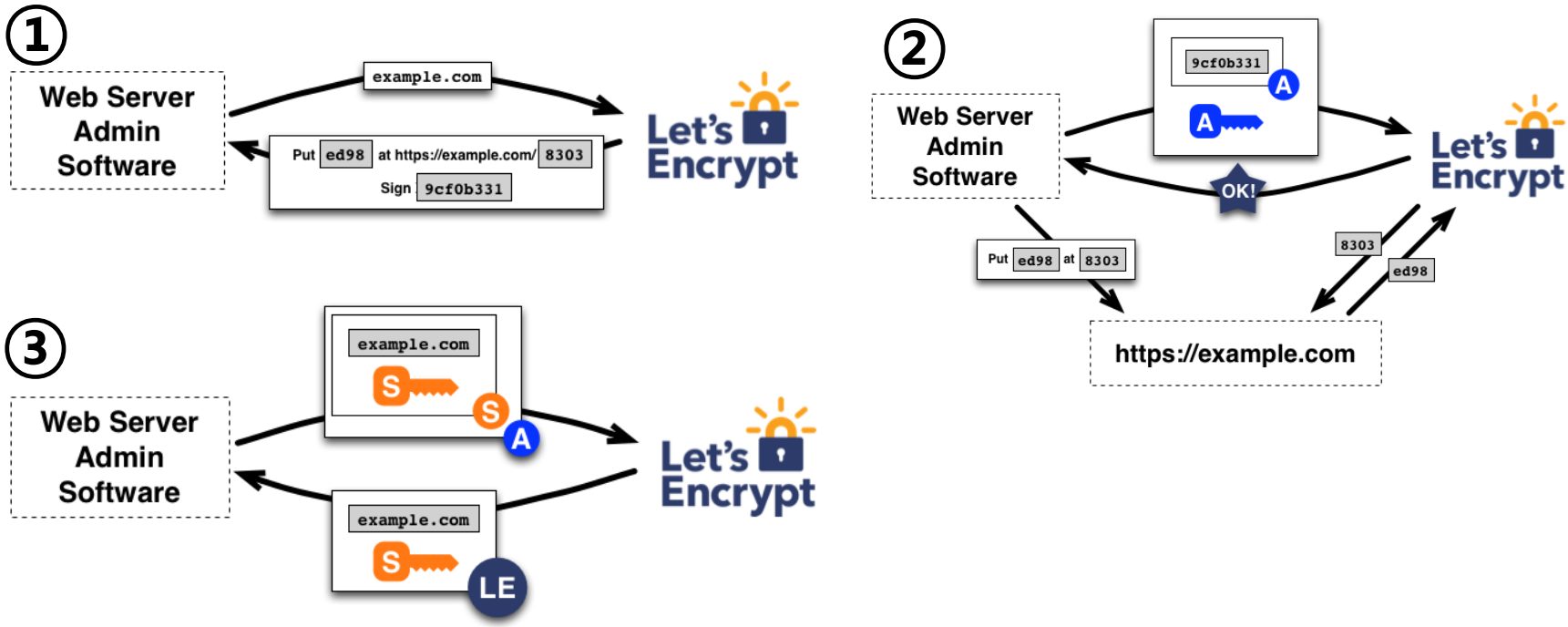
  → 사설 Root 인증서를 클라이언트 마다 설치해야 함



현황 및 이슈

- Let's Encrypt (letsencrypt.org)
  - **Free**, **Automated**, and Open **Certificate** Authority

```
$ sudo apt-get install lets-encrypt
$ lets-encrypt example.com
```



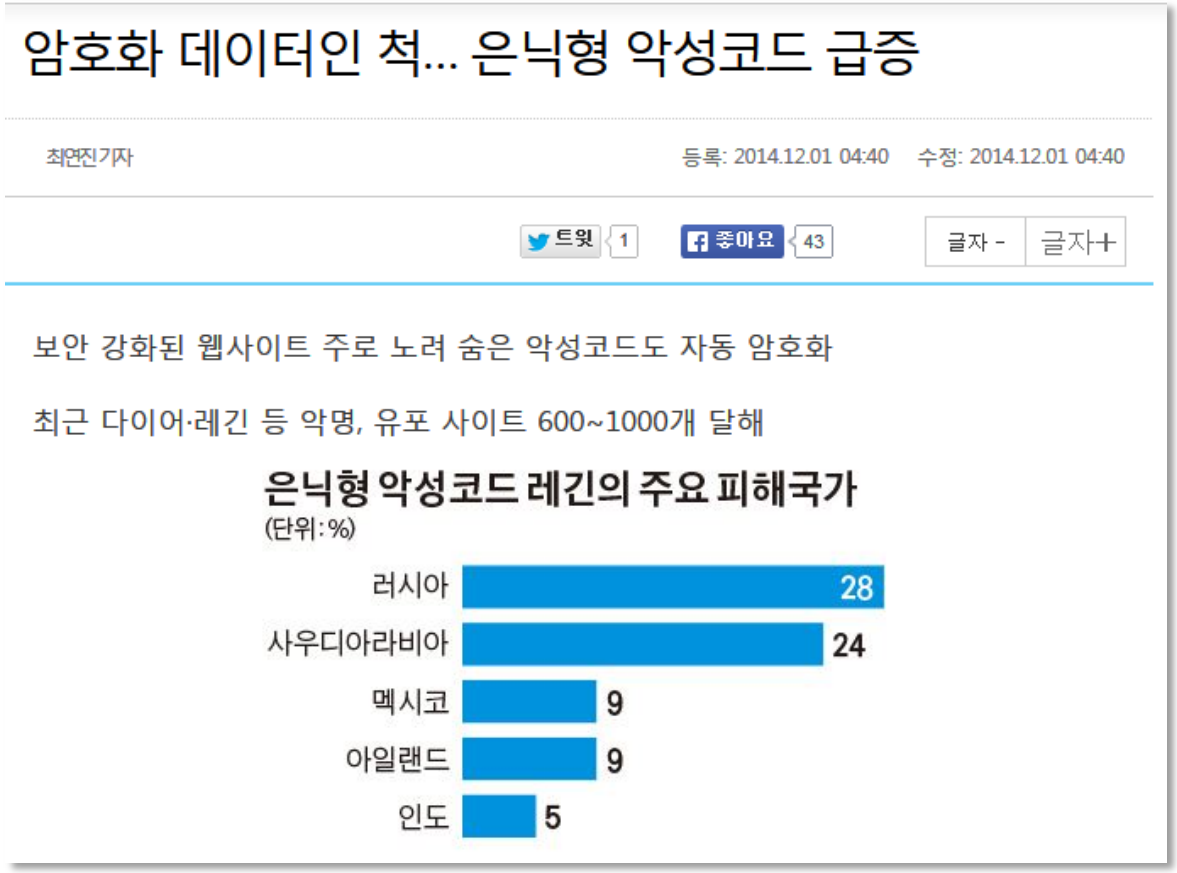
현황 및 이슈

- 웹페이지 수정 (http → https)
  - Find & Replace ('http://' → 'https://')
    - grep -rli 'http:' ./ | xargs sed -i 's/http:/https:/g'
    - 자바스크립트내 동적 생성 URL 처리
  - Protocol Relative URL
    - src="**//**example.com/image.jpg'
  - CMS(Content management system)
  - HSTS (HTTP Strict Transport Security)
    - http, https 모두 제공되지만 https로만 접속하도록 브라우저에 통보
    - 웹서버의 응답에서 HSTS 관련 헤더를 포함하여 수신하면 다음부터는 https로만 접속

***Strict-Transport-Security**: max-age=16070400; includeSubDomains*

현황 및 이슈

- 암호화 트래픽 모니터링
  - 멀웨어, 해킹, 기밀 유출을 어떻게 모니터링 할 것인가?



요약

- 변화된 웹서비스 및 네트워크 환경에 맞게 15년여만에 HTTP 표준 개정
- HTTP 2.0을 통해 웹서비스 성능 향상을 기대 가능
- 환경에 따라서는 오히려 성능 저하가 올 수 있음
- 웹서비스 최적화 기법의 재검토가 필요하며
- 주요 브라우저의 h2(https) only 정책으로 인해
- 기존 웹서비스에 대한 수정/보완 작업이 필요