

HTTP/2 Pros and Cons

Nick Shadrin / NGINX
nick@nginx.com
@shadrin

About me

- Nick Shadrin
- Technical Solutions Architect with NGINX
- Based in San Francisco
- 15 years experience with web tech
- nick@nginx.com / @shadrin



Links

All links in one page:

<https://shadrin.org/talks/>



Agenda

- Protocol overview
- HTTP/1 and HTTP/2 optimizations
- Troubleshooting
- Benchmarks
- Use of HTTP/2 with NGINX
- Conclusions



HTTP history

Year	Version	RFC#
1991	0.9	
1996	1.0	1945
1999	1.1	2616 and 7230 .. 7235
2015	2	7540

see Wikipedia



SCALE
14x

The Fourteenth Annual
Southern California Linux Expo

HTTP/1.1 example

```
GET /test HTTP/1.1
Host: example.com
User-Agent: Mozilla
X-Forwarded-For:
192.168.10.1
Accept: image/gif, image/
jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip,
deflate
```

```
HTTP/1.1 301 Moved
Permanently
Server: nginx/1.9.9
Date: Tue, 19 Jan 2016
00:19:07 GMT
Content-Type: text/html
Content-Length: 184
Connection: close
Location: https://
example.com/test
```



SPDY

- Announced in 2009 by Google
- Since then implemented in all major browsers
- Major goal: reduce page load time
- Major performance enhancements:
 - Compressed headers
 - Flow control
 - Server Push

NGINX



HTTP/2 overview

- Introduced in 2015 as a standard
- Based on SPDY
- Includes major changes compared to HTTP/1:
 - Binary headers with HPACK
 - Multiple streams
 - Prioritization
 - Server Push

Encryption

- Is encryption required?
 - Spec says "No"
 - Browser vendors say "Yes"

Protocol negotiation

- Upgrade header
- NPN
- ALPN

NPN

- Next protocol negotiation
- Server lists the protocols
- Client picks one

ALPN / RFC7301

- Application level protocol negotiation
- Client lists the protocols
- Server picks one
- Results in fewer round trips
- Available in openssl 1.0.2

Revise your optimizations

- Domain sharding
- Image sprites
- Concatenating code files

Domain Sharding

- Browser opens 6 connections to the host
- Distribute your resources through multiple domains
- Does it help when you use HTTP/2? - **No.**

Image Sprites

- Aggregate multiple images in a single file
- Separate images on the client side
- Does it help when you use HTTP/2? - **Somewhat.**

Concatenating code files

- Combine JS and CSS into larger files
- Does it help when you use HTTP/2? - **Not significantly.**

Revise your optimizations

- Domain sharding
- Image sprites
- Concatenating code files
- **All these optimizations add to the management overhead.**

HTTP/2 stats today

- Is it time already to use HTTP/2? Let's look at the statistics

HTTP/2 protocol - OTHER

Networking protocol for low-latency transport of content over the web. Originally started out from the SPDY protocol, now standardized as HTTP version 2.

Current aligned Usage relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8								4.3	
9			² 45					4.4	
10	² 12	² 42	² 46			8.4		4.4.4	
^{1 2} 11	² 13	² 43	² 47	^{2 3} 9	² 34	² 9.2	8	46	² 47
	² 14	² 44	² 48	^{2 3} 9.1	² 35	² 9.3			
		² 45	² 49		² 36				
		² 46	² 50						

Notes Known issues (0) Resources (5) Feedback



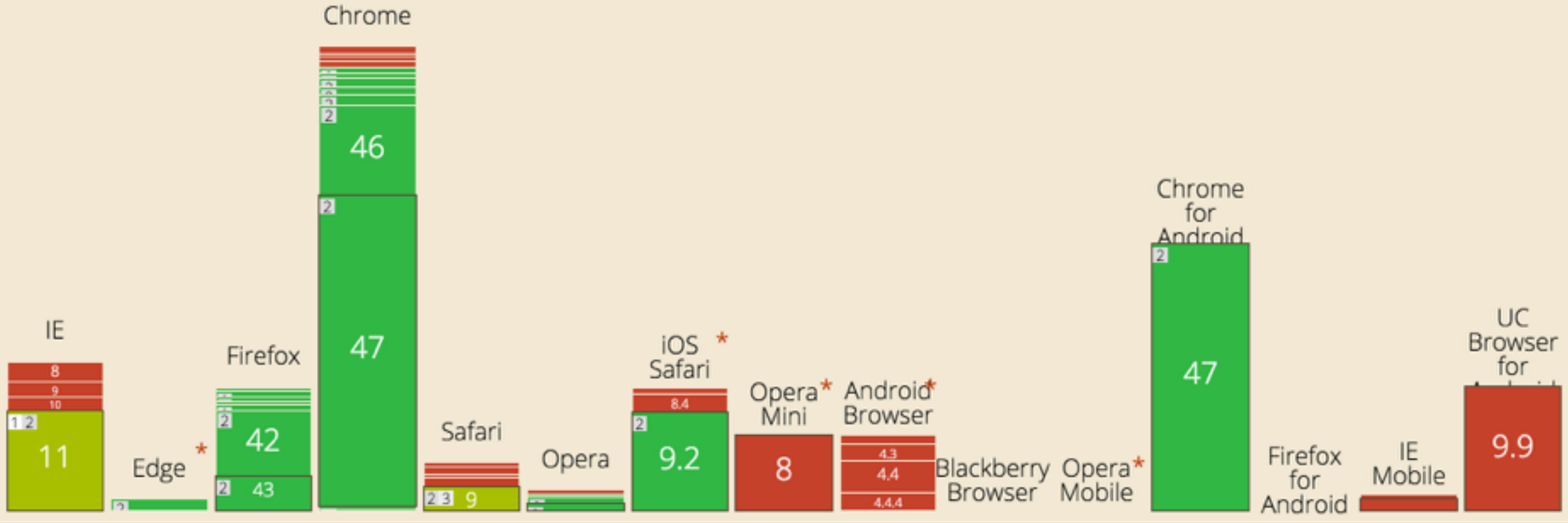
HTTP/2 usage (caniuse.com)



HTTP/2 protocol - OTHER

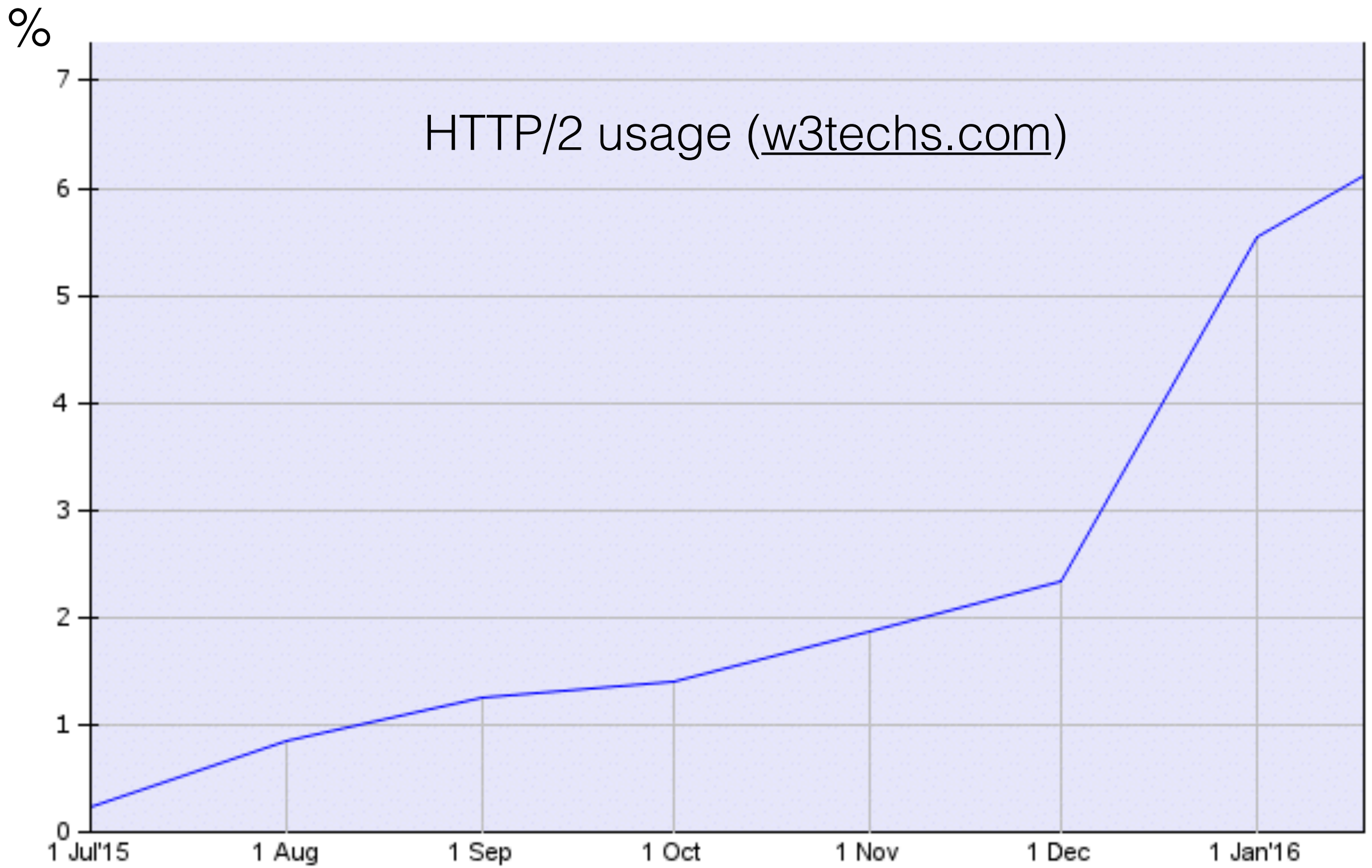
Networking protocol for low-latency transport of content over the web. Originally started out from the SPDY protocol, now standardized as HTTP version 2.

Current aligned Usage relative Showing all



HTTP/2 usage (caniuse.com)





Usage of HTTP/2 for websites, 18 Jan 2016, W3Techs.com



Downsides

- Greater overhead for the single connection.
- You might not need SSL.
- HTTP/1.x optimizations hurt.
- Big downloads don't benefit.
- Your customers may not care.

Wireshark · Follow TCP Stream (tcp.stream eq 8) · wireshark_pcapng_en0_2016011815...

```
.....L.)v0.WR.....rb.U-....h...8B.u.W.....+./.....
...9.    ...3...5./..
...{.....shadrin.org.....#...
.....3t.....http/1.1.spdy/
3.1.h2uP.....
.....Q...M....g4@..>...SC4<!..}.._..... ..4<.e.../..
%.....#..3t...h2.http/1.1... ..
...0...0.....q.H.....In
..0
.      *.H..
.....0J1.0..U....US1..0...U.
.
Let's Encrypt1#0!..U....Let's Encrypt Authority X10..
151204200300Z.
16030320030070.1.0...U....shadrin.org0.."0
.      *.H..
.....0..
```

Packet 261. 8 client pkt(s), 8 server pkt(s), 7 turns. Click to select.

Entire conversation (5372 bytes) Show data as ASCII Stream 8

Find: Find Next

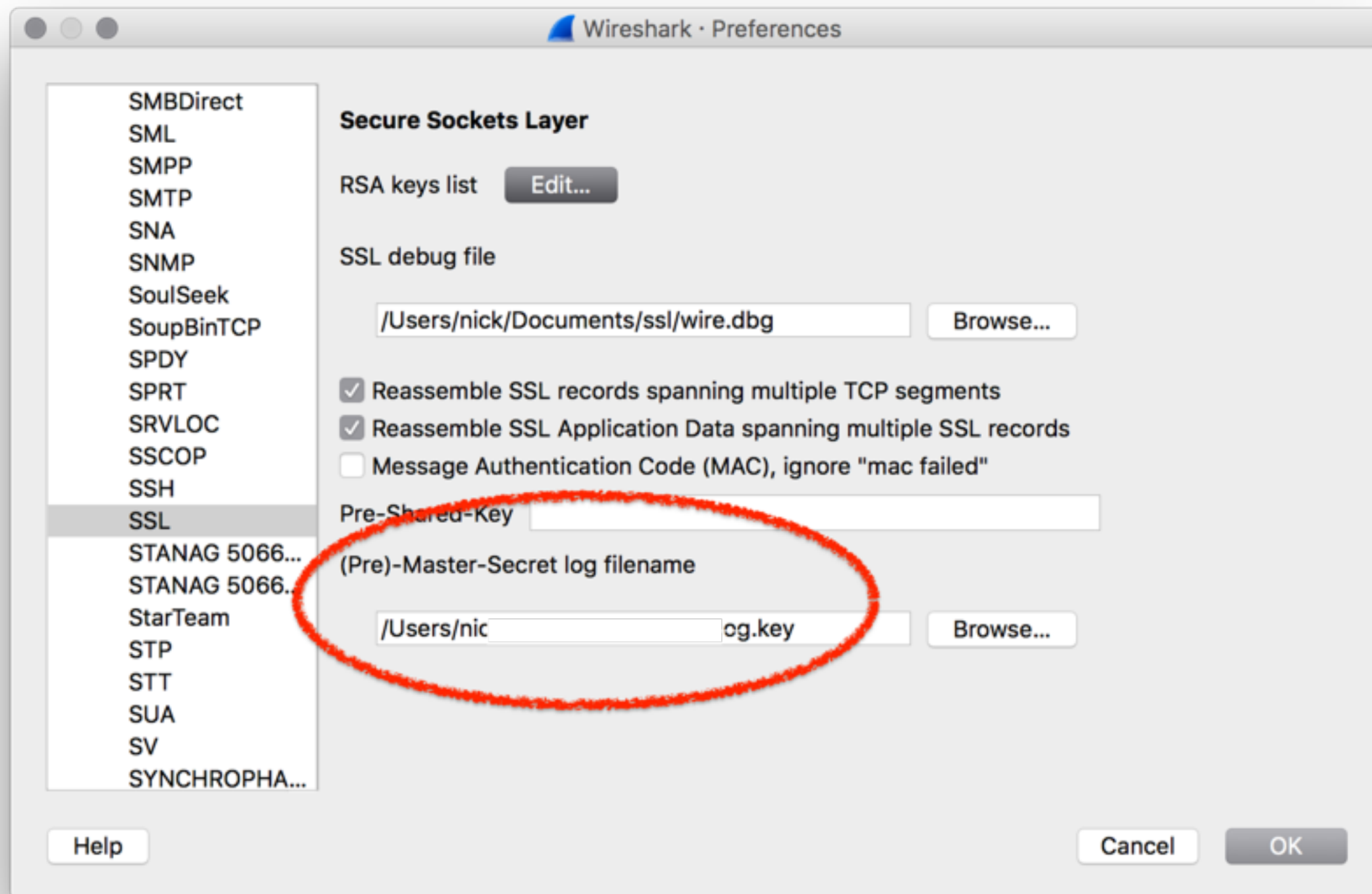
Help Hide this stream Print Save as... Close

HTTP/2 is unreadable



Troubleshooting HTTP/2 with Wireshark

1. Set the key log file ENV variable:
`export SSLKEYLOGFILE=/Users/path/file.key`
2. Open your browser:
`open -a Google\ Chrome`
3. Set the key file in Wireshark



Wi-Fi: en0

http2

No.	Time	Source	Destination	Protocol	Length	Info
32	0...	10.10.20.105	216.58.193.78	HTTP2	119	Magic
33	0...	10.10.20.105	216.58.193.78	HTTP2	116	SETTINGS
34	0...	10.10.20.105	216.58.193.78	HTTP2	108	WINDOW_UPDATE
35	0...	10.10.20.105	216.58.193.78	HTTP2	503	HEADERS
37	0...	216.58.193.78	10.10.20.105	HTTP2	122	SETTINGS
38	0...	216.58.193.78	10.10.20.105	HTTP2	108	WINDOW_UPDATE
42	0...	10.10.20.105	216.58.193.78	HTTP2	104	SETTINGS
44	0...	216.58.193.78	10.10.20.105	HTTP2	104	SETTINGS
46	0...	216.58.193.78	10.10.20.105	HTTP2	473	HEADERS
47	0...	216.58.193.78	10.10.20.105	HTTP2	104	DATA
48	0...	216.58.193.78	10.10.20.105	HTTP2	112	PING
52	0...	10.10.20.105	216.58.193.78	HTTP2	112	PING

[Weight real: 183]
Header Block Fragment: 0085b9495339e4074f5054494f4e53418bae81fa5e639e6a...
[Header Length: 665]

- ▶ Header: :method: OPTIONS
- ▶ Header: :authority: play.google.com
- ▶ Header: :scheme: https
- ▶ Header: :path: /log?format=json
- ▶ Header: access-control-request-method: POST
- ▶ Header: origin: https://www.google.com
- ▶ Header: user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrom...
- ▶ Header: access-control-request-headers: accept, authorization, content-type, x-goog-authuser
- ▶ Header: accept: /*/*
- ▶ Header: x-client-data: CKW2yQEIXLbJAQj9lcoB
- ▶ Header: dnt: 1
- ▶ Header: referer: https://www.google.com/_/chrome/newtab-serviceworker.js
- ▶ Header: accept-encoding: gzip, deflate, sdch
- ▶ Header: accept-language: en-US,en;q=0.8,es;q=0.6,fr;q=0.4,pt;q=0.2,sk;q=0.2

```

0000  00 90 7f a8 65 f8 a0 99 9b 11 3c df 08 00 45 00  ....e... ..<...E.
0010  01 e9 1c 93 40 00 40 06 64 80 0a 0a 14 69 d8 3a  ....@.@. d....i.:
0020  c1 4e d0 97 01 bb 3c 83 d1 da 2d 62 ff c2 80 18  .N....<. ..-b....
0030  10 00 27 e3 00 00 01 01 08 0a 49 78 31 f5 d0 6c  ..'..... ..Ix1..l
0040  1d 42 17 03 03 01 b0 00 00 00 00 00 00 04 ce  .B..... ..

```

Frame (503 bytes) | Decrypted SSL data (408 bytes) | Decompressed Header (665 bytes)

Packets: 948 · Displayed: 12 (1.3%) · Dropped: 0 (0.0%) Profile: Default

▼ HyperText Transfer Protocol 2

▼ Stream: HEADERS, Stream ID: 3, Length 131

Length: 131

Type: HEADERS (1)

▼ Flags: 0x04

.... 0 = End Stream: False

.... 1 = End Headers: True

.... 0 = Padded: False

..0. = Priority: False

00.0 ..0. = Unused: 0x00

0... .. = Reserved: 0x00000000

.000 0000 0000 0000 0000 0000 0000 0011 = Stream Identifier: 3

[Pad Length: 0]

Header Block Fragment: 88760b6e67696e782f312e392e34611d4d6f6e2c20313820...

[Header Length: 217]

▶ Header: :status: 200

▶ Header: server: nginx/1.9.4

▶ Header: date: Mon, 18 Jan 2016 23:51:37 GMT

▶ Header: content-type: text/css

▶ Header: last-modified: Sun, 19 Jul 2015 18:00:27 GMT

▶ Header: etag: W/"55abe5bb-df"

▶ Header: content-encoding: gzip

Padding: <MISSING>

Benchmarks

Benchmarks from nginx.conf 2015

Test Environment

Hardware: Intel Core i7-4770S, 16Gb of RAM,
no disk I/O was involved

Kernel: Linux 4.0.9-gentoo

Network: loopback, 1400 MTU, netem

Server: nginx 1.9.5

Client: Chromium 45.0.2454.85 (64-bit)
via Selenium WebDriver

The results were analysed using `ministat`:

<http://www.freebsd.org/cgi/man.cgi?query=ministat>

**Please note, that absolute numbers are irrelevant.
Look at the trend.**

nginx.conf

```
events { }

http {
    include conf/mime.types;
    root ../http2rulez.com/public;

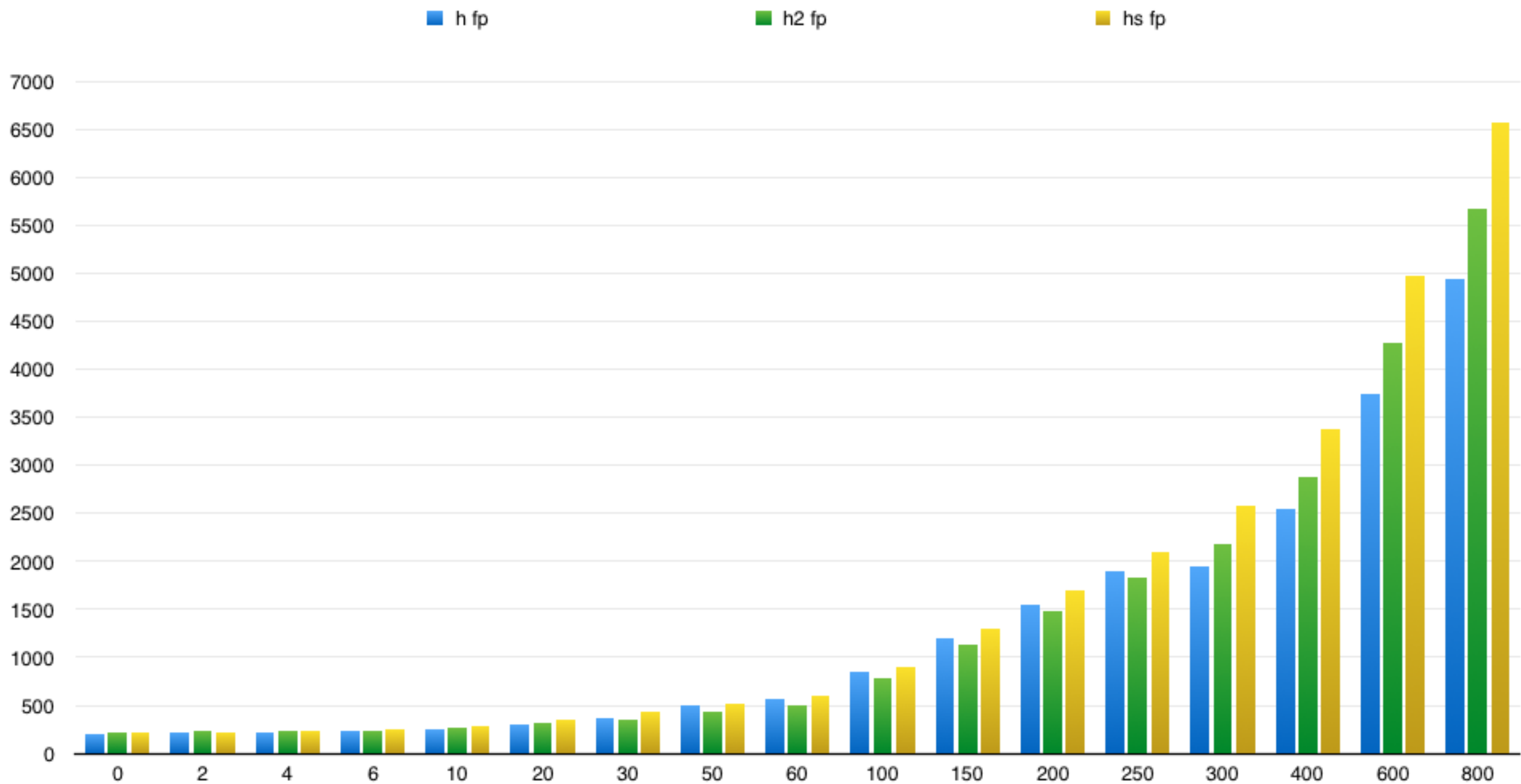
    ssl_certificate ../server-ca.crt;
    ssl_certificate_key ../server.key;

    ssl_buffer_size 4k;
    output_buffers 2 1m;

    gzip on;
    gzip_types text/css application/javascript;

    server {
        listen 127.0.0.2:4433 ssl http2;
        location / { }
    }
    server {
        listen 127.0.0.2:4432 ssl;
        location / { }
    }
    server {
        listen 127.0.0.2:8080;
        location / { }
    }
}

#nginx #nginxconf
```

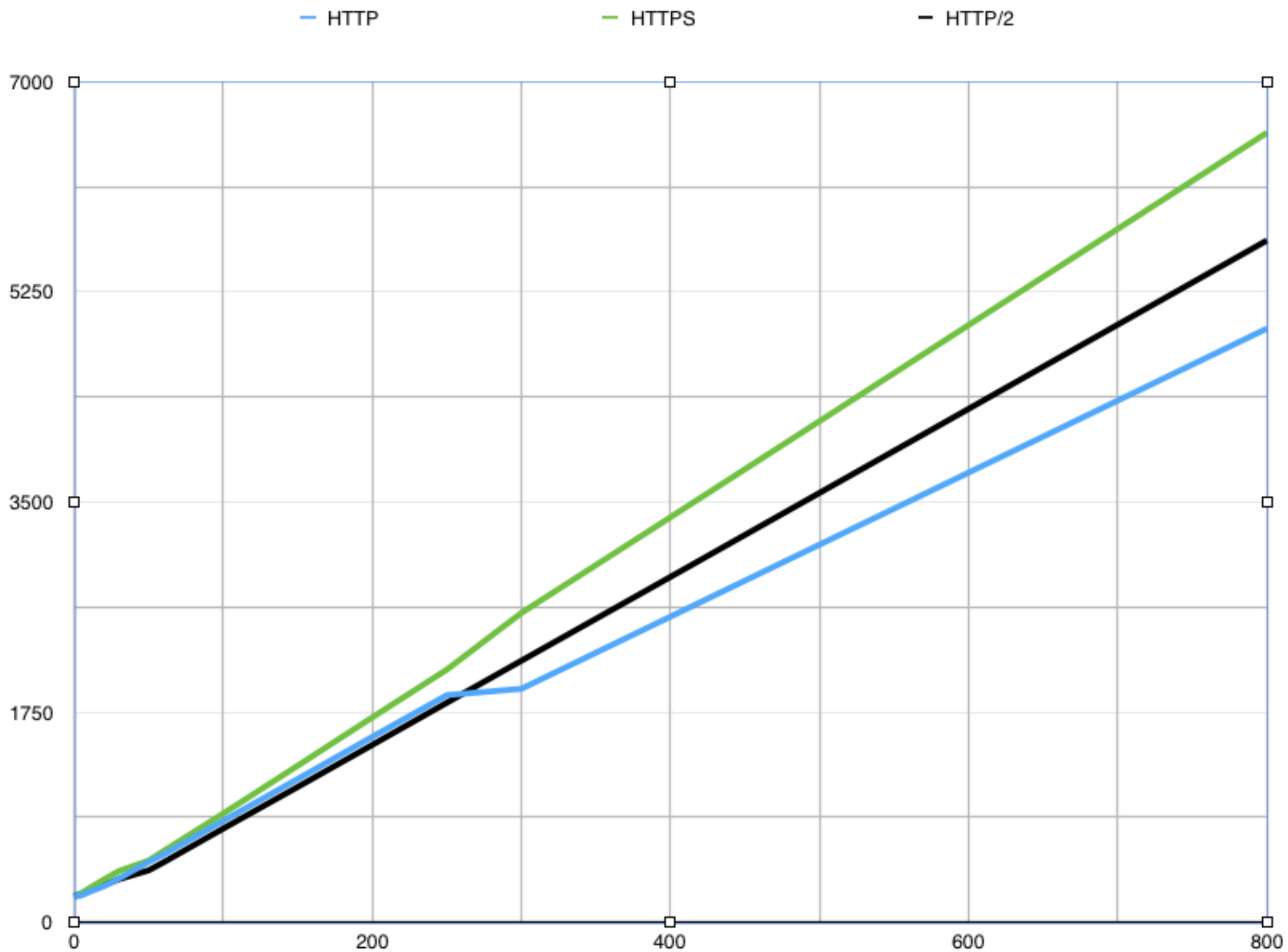


NGINX



SCALE
14x

The Fourteenth Annual
Southern California Linux Expo



Benchmark setup

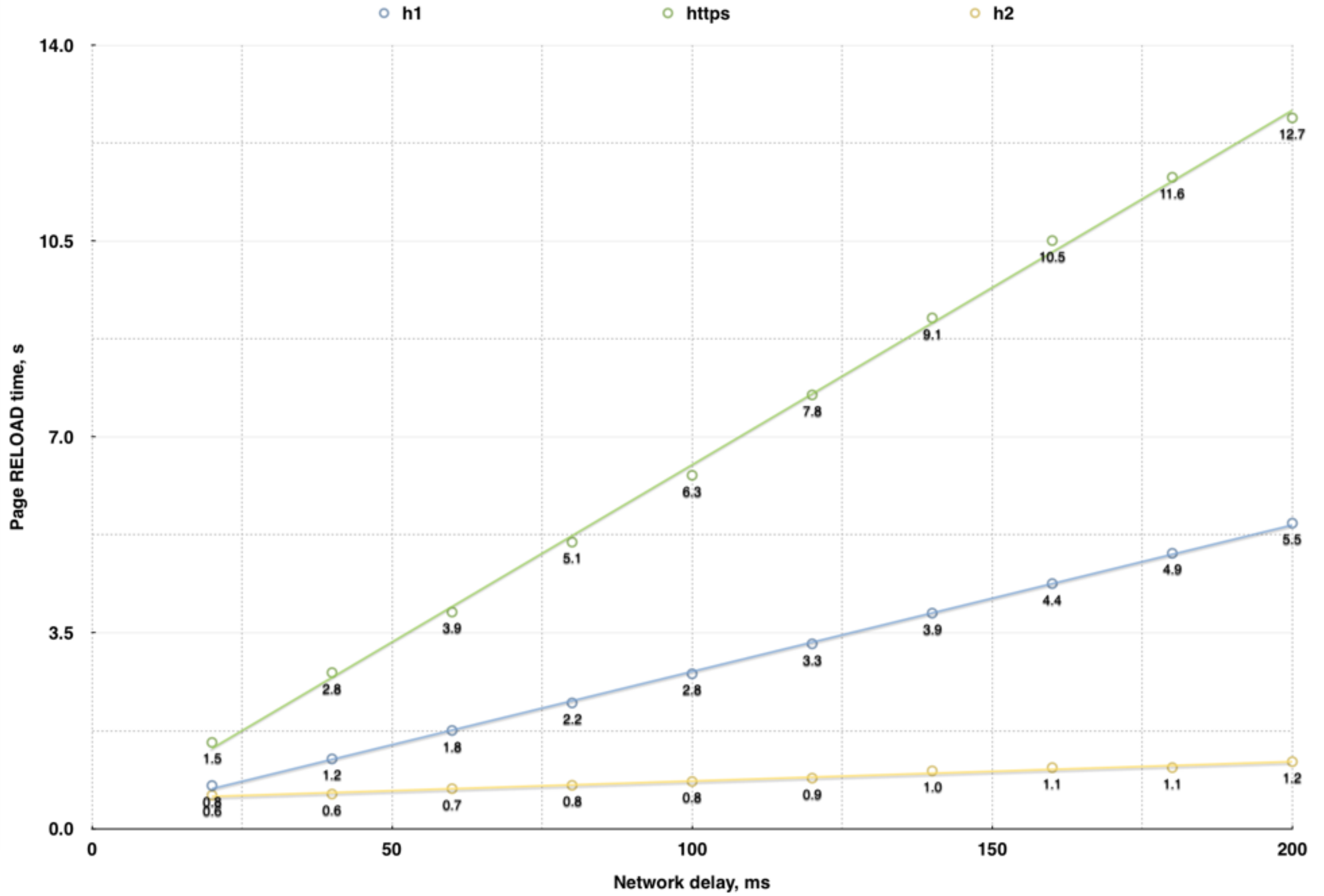
- NGINX 1.9.9
- Ubuntu 15.10
- Openssl 1.0.2
- Chrome
- Measuring full page reload

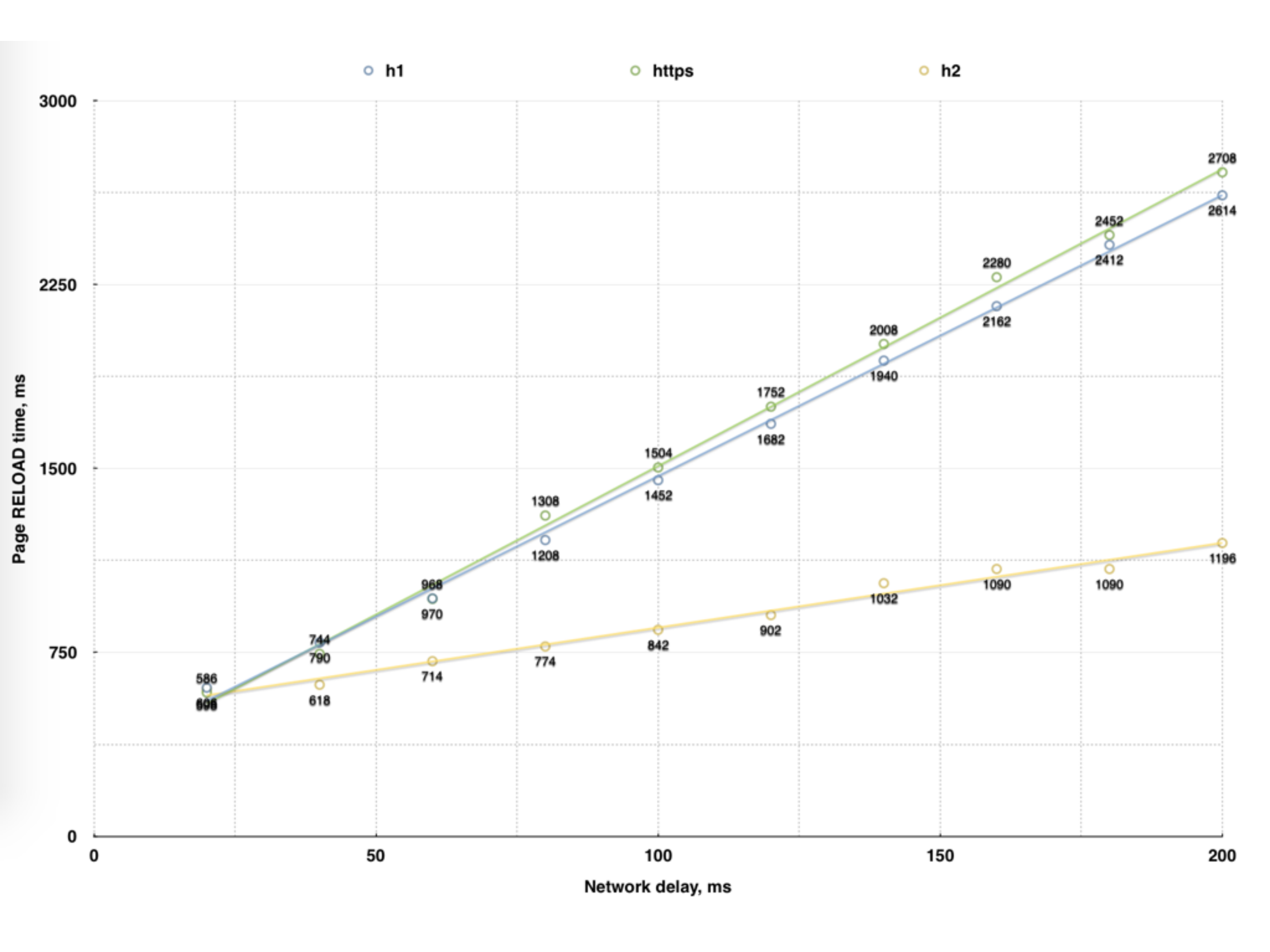


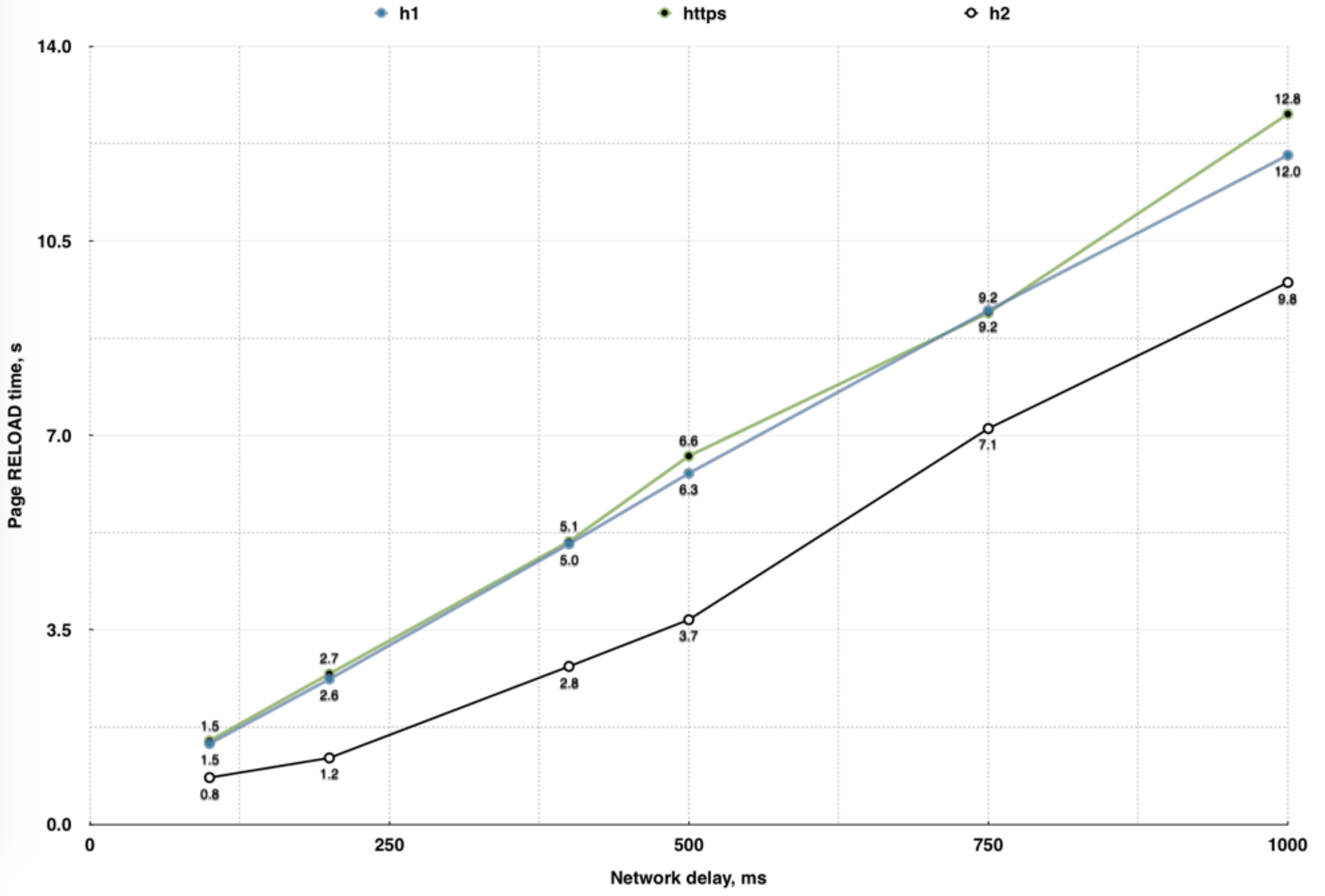
New benchmark page

- Free template BSCORE Admin:
<http://binarytheme.com/bootstrap-free-admin-dashboard-template/>
- Added more images: 54 objects total

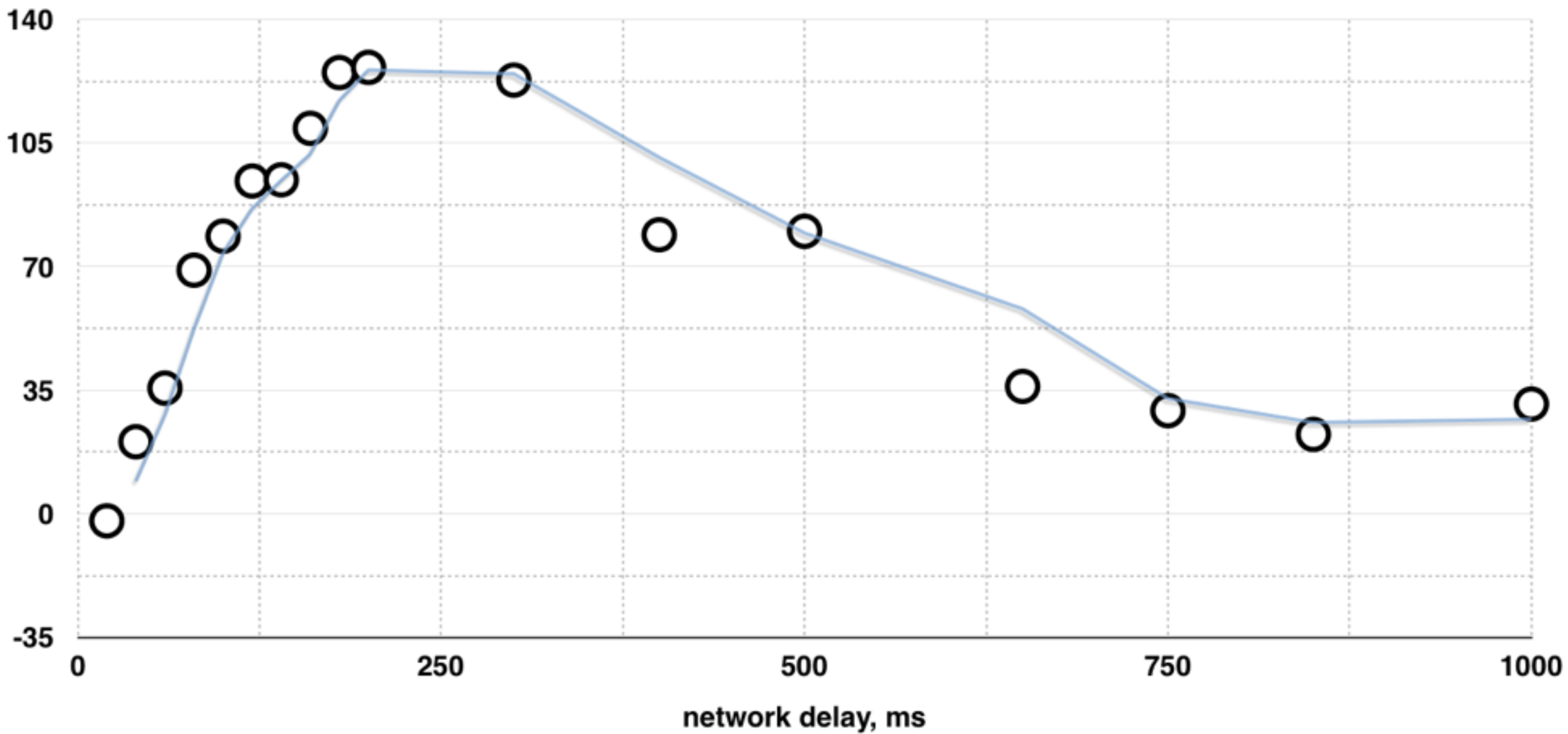
The screenshot displays the BSCORE Admin dashboard. At the top left, the logo 'BSCORE Admin' is shown in green. The top right corner features notification icons for 2 messages, 5 alerts, 8 chat messages, and a user profile. The main header area includes a user profile for 'Joe Romlin' (Online) and the title 'Admin Dashboard'. Below the header, a row of six widget cards shows: Products (2), Messages (456), Profit (+25), value (3.14159265), tasks (107), and Tickets (20). The left sidebar contains a menu with items: Dashboard, UI Elements (10), Forms (5), Pages (6), Charts (4), 3 Level Menu, 4 Level Menu, and Error Pages (5). The right sidebar contains a statistics box with 'Visitor : 23,000', 'Users : 53,000', and 'Registrations : 3,000', and a vertical list of buttons: Help, Tickets, New, Users, Profit, and Sales. The main content area is divided into two sections: 'Real Time Traffic' with a line graph showing values between 80 and 100, and a 'Chat' window with a message from 'Jack Sparrow' (50x50 profile picture) received 12 minutes ago, containing placeholder text.







○ % benefit from using http2



NGINX support for h2

```
user@server$ ./configure --with-http_v2_module --  
with-http_ssl_module
```

```
[...]
```

```
user@server$ nginx -V
```

```
nginx version: nginx/1.9.9
```

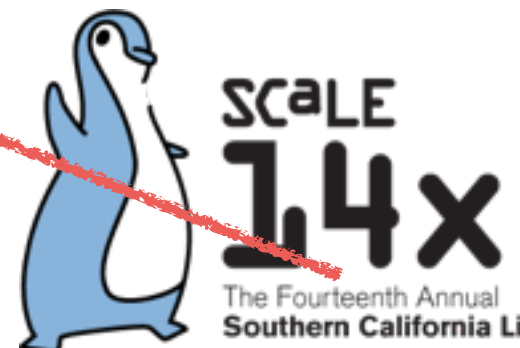
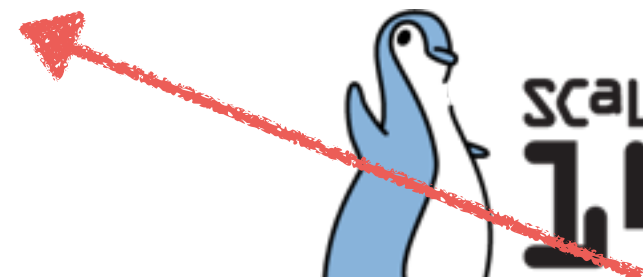
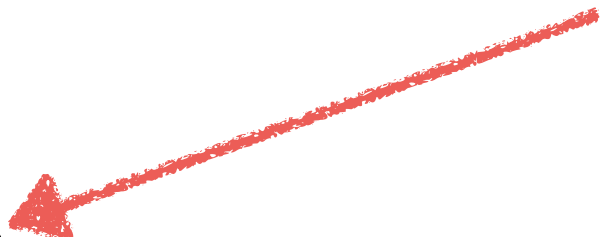
```
built by gcc 4.8.4 (Ubuntu 4.8.4-2ubuntu1~14.04)
```

```
built with OpenSSL 1.0.1f 6 Jan 2014
```

```
TLS SNI support enabled
```

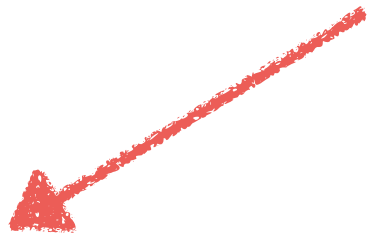
```
configure arguments: --with-http_v2_module --
```

```
with-http_ssl_module
```



NGINX config for h2

```
server {  
    listen 443 ssl http2;  
    server_name .example.com;  
    ssl_certificate /etc/nginx/ssl/example.com.crt;  
    ssl_certificate_key /etc/nginx/ssl/  
example.com.key;  
    ssl_protocols TLSv1.2;  
    root /data/example.com;  
    location / {  
        proxy_pass http://backend.example.com/;  
    }  
}
```

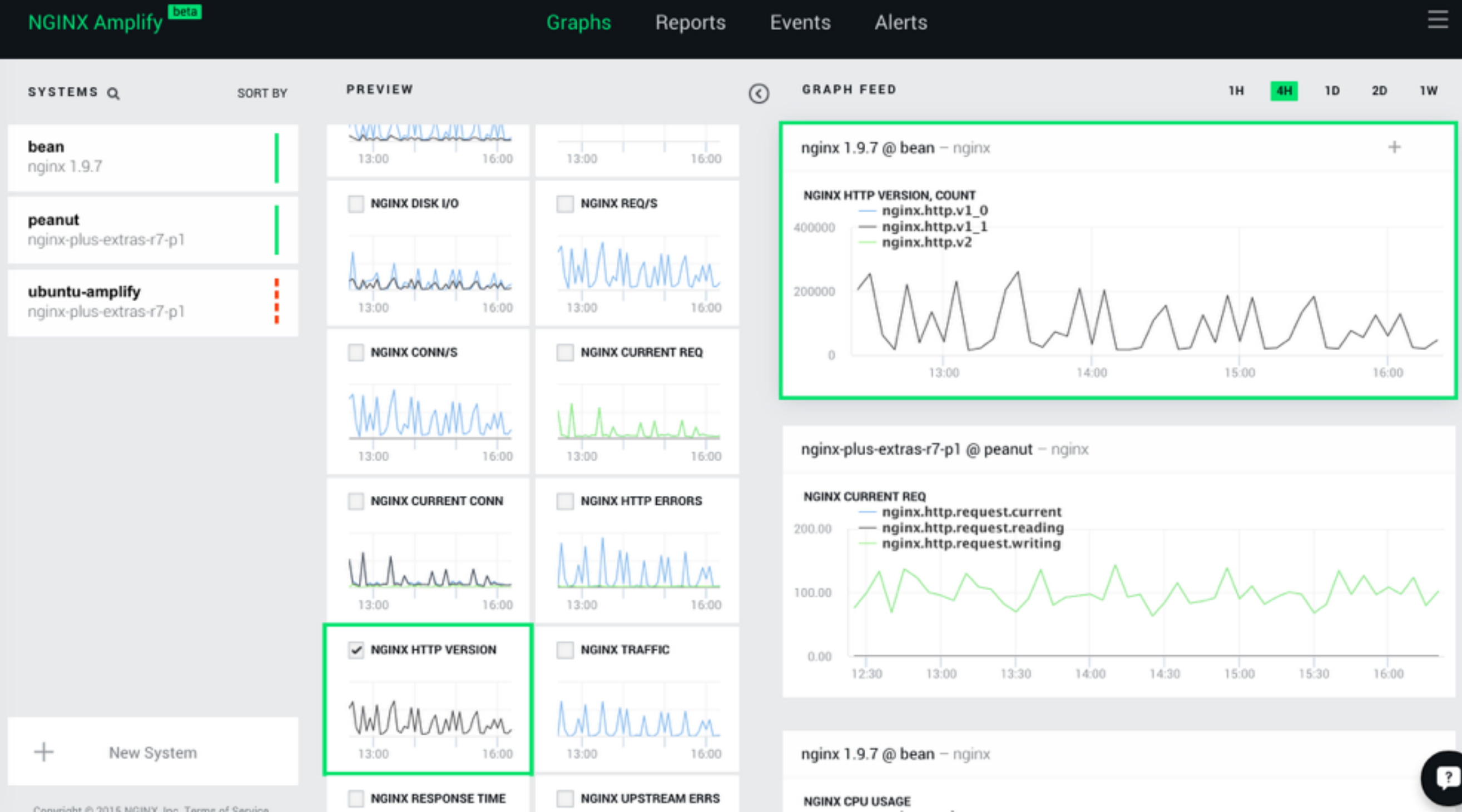


NGINX logs for h2

- \$request = GET /url HTTP/2.0
- 10.2.2.2 - - [18/Jan/2016:16:51:40 -0800] "**GET / HTTP/2.0**" 200 3470 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36"



NGINX Amplify for monitoring



Useful tools

caniuse.com

letsencrypt.com

webpagetest.org

Links

All links in one page:

<https://shadrin.org/talks/>

