

# HTTP/2 Out Of The Box

Can you get it with “stable” Linux?



**HTTP/2**

# HTTP/2 was published at May 14, 2015

HTTP 1.1 was standardized in 1997 - it's 18 years old!

HTTP/2 was published at May 14, 2015 as [RFC 7540](#).

According to [W3Techs](#), as of April 2016 7.1% of the top 10 million websites supported HTTP/2.

On July 28, 2016 we have 8.9% of all websites using HTTP/2

On July 30, 2016 we have 9% of all websites using HTTP/2

What is so exciting in HTTP/2?

# HTTP/2 Benefits

Compatible with HTTP 1.1

Focused on performance:

- Data compression of HTTP headers
- HTTP/2 Server Push
- Pipelining of requests
- Fixing the head-of-line blocking problem in HTTP 1.x
- Multiplexing multiple requests over a single TCP connection

# HTTP/2 Antipatterns

Domain sharding

Concatenated assets

Inline resources

# If you want to know more:

<https://http2.github.io/>

<https://en.wikipedia.org/wiki/HTTP/2>

<https://tools.ietf.org/html/rfc7540>

<https://http2.akamai.com/>

<https://www.smashingmagazine.com/2016/02/getting-ready-for-http2/>

<https://www.cloudflare.com/http2/>

# Exciting, isn't it?

You work over new project.

You have plenty of small files: JS, CSS and images.

You can throw away a lot of routine like JS/CSS concatenation and **sprite generation**.





Can you adopt HTTP/2?

# First try with NGINX

On August 5, 2015 NGINX announced an Early Patch for “mainline”.

So the first experiment was:

- try to compile NGINX with Patch
- try to run it with self-signed certificates
- see how it works

The NGINX logo is displayed in a bold, green, sans-serif font. The letters are thick and blocky, with a distinctive white horizontal bar cutting through the middle of the 'G' and 'M' characters. The 'X' at the end is formed by two thick diagonal strokes meeting at a central point.

# It compiles and HTTP/2 works!



# But you can't apply it...

- NGINX “mainline” is not so “stable” as “stable”
- You need to patch source
- You need to compile by yourself

Too much work and responsibility - you must support your custom version until desired feature will become “stable”.

Harder to implement automated setup with Puppet.

You don't want to use unstable and  
untested software

# “Stable” Linux only!

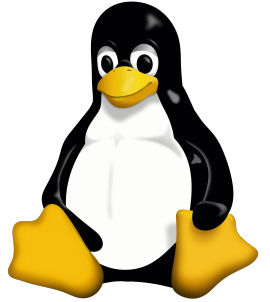
You want stability of your infrastructure.

You may be limited in using only “stable” Linux.

You don’t want to compile manually too much software.

You don’t want to deal with dozen software repositories you don’t trust on 100%.

You even might be limited to some Linux distribution only.



So we moved to SPDY with  
SSL at that moment.



Page load speed increase was huge!

And we were happy!!!1!



# Second try with NGINX

On April 26, 2016 NGINX released “stable” version 1.10.0 with HTTP/2 support.

Upgrade was seamless:

1. Install new NGINX version from “stable” repository
2. Change “spdy” to “http2” in NGINX config
3. Restart NGINX

Voilà - everything works!



Well, almost...

# One day everything went wrong...

It was just an ordinary day in May, 2016

Front-End Developers with:

- Microsoft Internet Explorer 11
- Microsoft Edge
- Google Chrome Canary

reported that HTTP/2 is not working.

But it was working on: Chromium, Google Chrome and Firefox.

And you are in situation like...





And you start searching for the  
issue...

And you missed notes about  
OpenSSL version!

And this is mentioned everywhere!



# World is Changing

Transition from HTTP 1.1 or SPDY to HTTP/2 is in progress.

Browsers are constantly updated.

HTTP Servers are constantly updated.

Linux software is constantly updated.

New Linux versions are released.

# Notable change in browsers

Dropping SPDY support.

Dropping HTTP/2 with NPN (Next Protocol Negotiation).

Implementing HTTP/2 with ALPN (Application-Layer Protocol Negotiation).

ALPN is Required by HTTP/2  
[RFC 7301](#)

# Our Issue

We use HTTP/2 over TLS.

We use Debian 7.

We have OpenSSL version 1.0.1 which supports only NPN (Next Protocol Negotiation). Some older browsers might still work with this protocol.

Newer browsers require ALPN (Application-Layer Protocol Negotiation) for HTTP/2 and it is available only in OpenSSL 1.0.2.



Some day every browser will move to  
ALPN \*\*\*

This slide was added at May 15, 2016

**\*\*\* Browser already moved to ALPN**

This slide was added at August 2, 2016

So we are a bit doomed.



Let look on other Linux distributions and  
OpenSSL versions

# OpenSSL on “stable” Linux

## Version 1.0.1

Debian 7, 8

Ubuntu 14.04

Fedora 22

CentOS 7

openSUSE 13.2

Linux Mint 17.3

## Version 1.0.2

Debian 8 “backports”

Ubuntu 15.10, 16.04

Fedora 23, 24

--

--

Linux Mint 18

# Solutions if you have OpenSSL 1.0.1

Find trusted repository with OpenSSL 1.0.2.

Download source of OpenSSL 1.0.2 and compile manually.

Change your distribution to which has OpenSSL 1.0.2.

HTTP/2 and SSL termination on host with different Linux distribution that has OpenSSL 1.0.2.

Make your choice

# Discussion



Sergej Kurakin

Work @mail: [sergej.kurakin@nfq.it](mailto:sergej.kurakin@nfq.it)

Personal @mail: [sergej@kurakin.info](mailto:sergej@kurakin.info)

<https://www.linkedin.com/in/sergejkurakin>