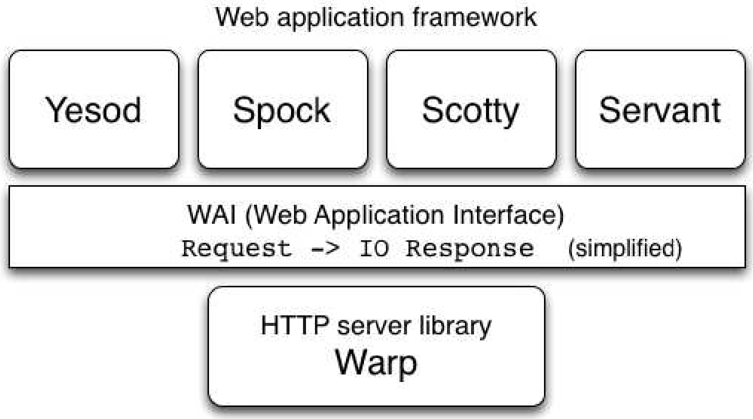
HTTP/2 and TLS in Warp

Internet Initiative Japan Inc.
Kazu Yamamoto

What's Warp?

- High performance HTTP server library in Haskell

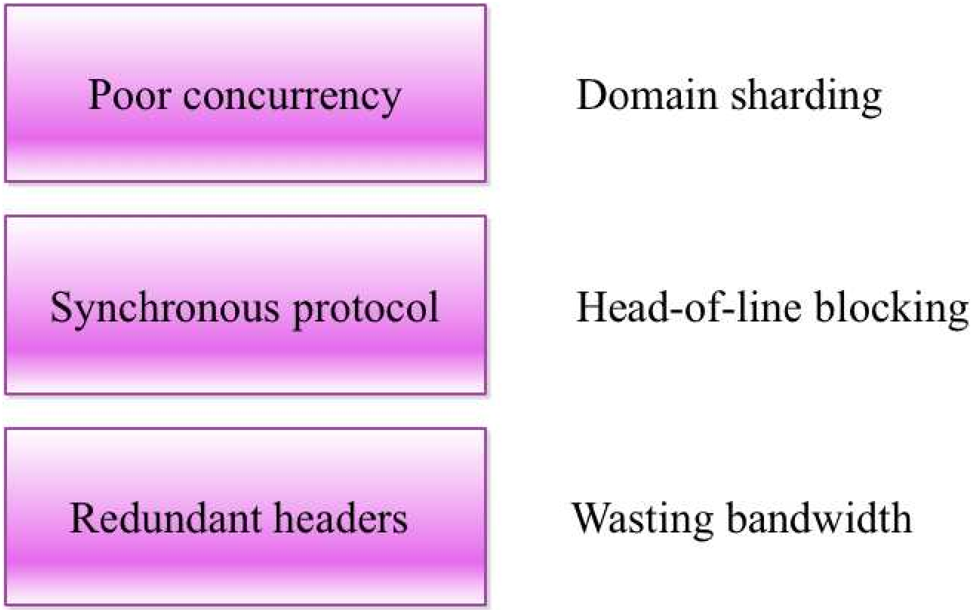

Web application framework
Yesod
Spock
Scotty
Servant
WAI (Web Application Interface)
Request -> IO Response (simplified)
HTTP server library
Warp

Warp now supports HTTP/2

Demonstration

HTTP/1.1 vs HTTP/2

Why HTTP/1.1 is slow?



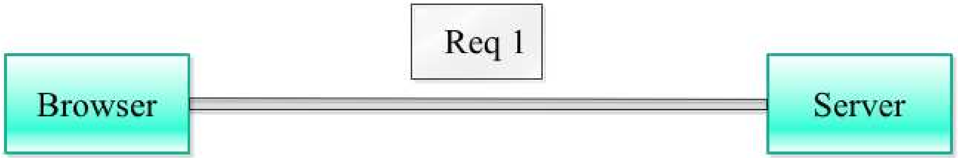
Poor concurrency — Domain sharding

Synchronous protocol — Head-of-line blocking

Redundant headers — Wasting bandwidth

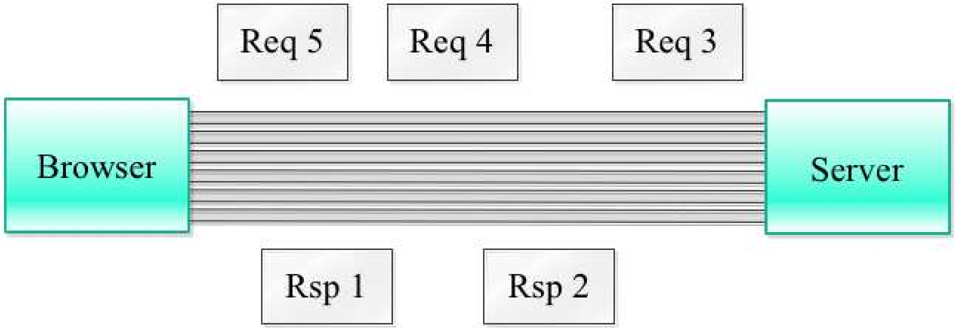
Poor concurrency

- HTTP/1.1 can host one job per connection at most
  - One request, one response or nothing



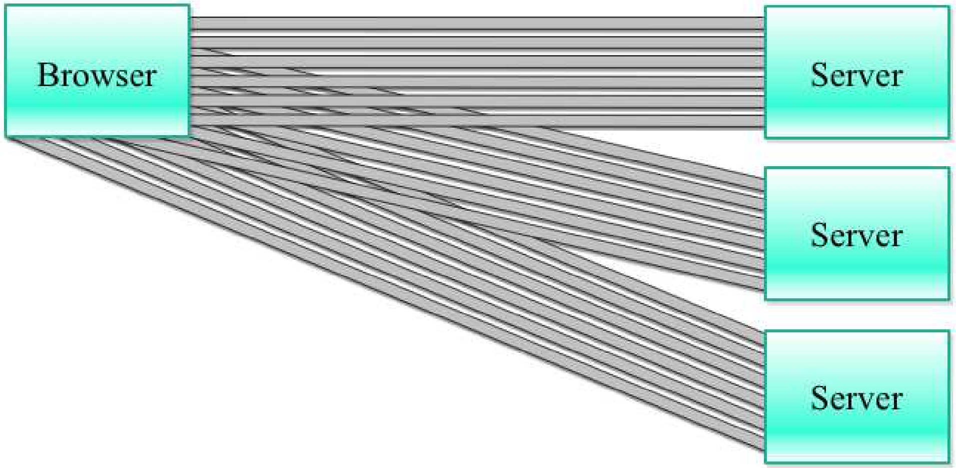
Workaround

- Major browsers make multiple connections
  - Up to 6 - 8 connections per origin



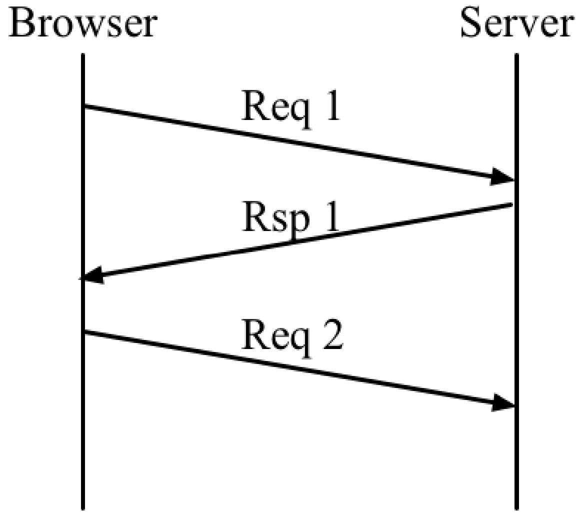
Domain sharding

- Increasing origins to increase concurrency
  - Distributed content make content management harder



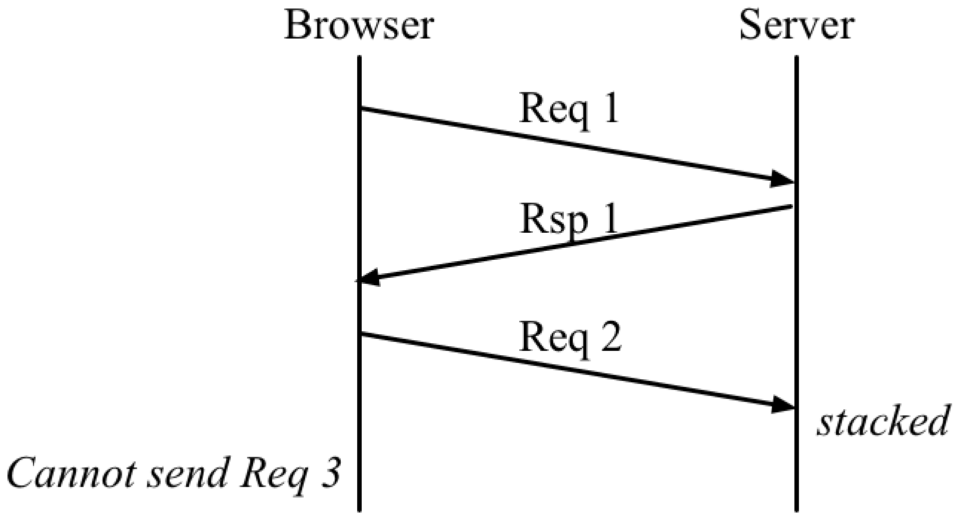
Synchronous protocol

- HTTP/1.1 is synchronous
  - Server: processes a request then sends a response
  - Browser: receives a response then sends the next request



Head-of-line blocking

- Requests are stacked
  - Browser cannot send requests if a response generation is stacked

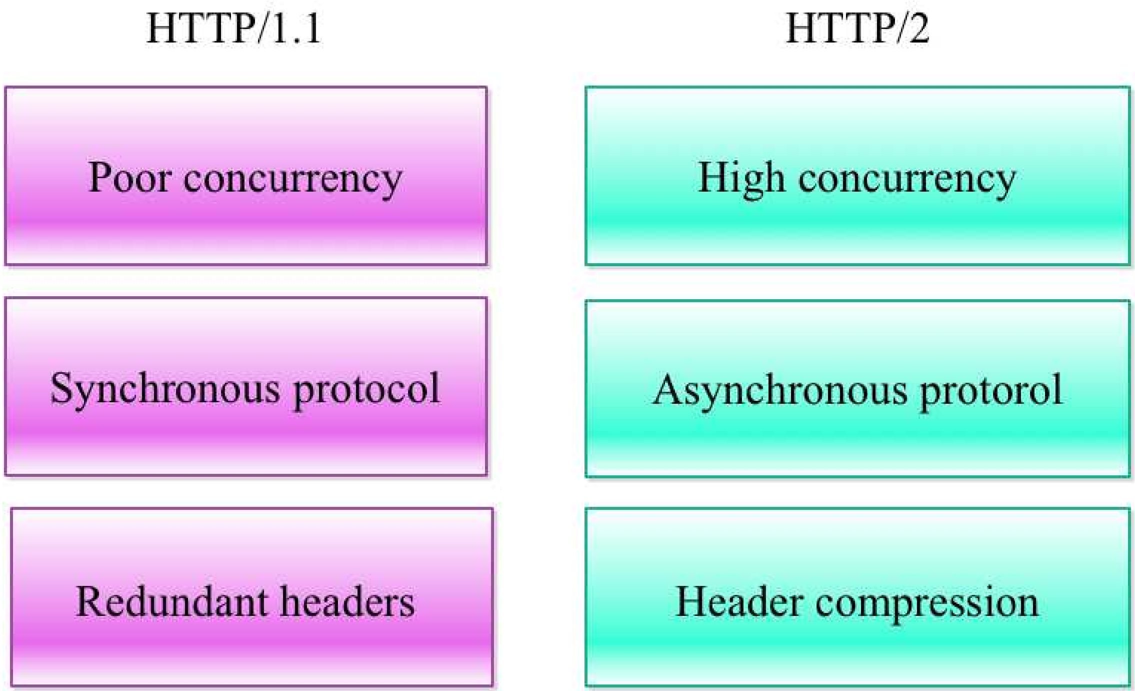


Redundant headers

- Wasting bandwidth
  - Average size of request headers is about 800 bytes
  - Almost the same header is sent in every request

```
GET /roversync/ HTTP/1.1
Host: rover.ebay.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8;
 rv:16.0) Gecko/20100101 Firefox/16.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.ebay.com/
Cookie: ebay=%5Esbf%3D%23%5E; dp1=bpbf/%2380000000000055
 276504d^u1p/QEBfX0BAX19AQA**5276504d^; cssg=c67883f113a
 0a56964e646c6ffaa1abe; s=CgAD4ACBQlm5NYzY3ODgzZjExM2EwY
 TU2OTY0ZTY0NmM2ZmZhYTFhYmUBSgAYUJZuTTUwOTUxY2NkLjAuMS4z
 LjE1MS4zLjAuMeN+7JE*; nonsession=CgAFMABhSdlBNNTA5NTFjY
 2QuMC4xLjEuMTQ5LjMuMC4xAMoAIFn7Hk1jNjc4ODNmMTEzYTBhNTY5
 NjRlNjQ2YzZmZmFhMWFjMQDLAAFQlSPVMX8u5Z8*
```

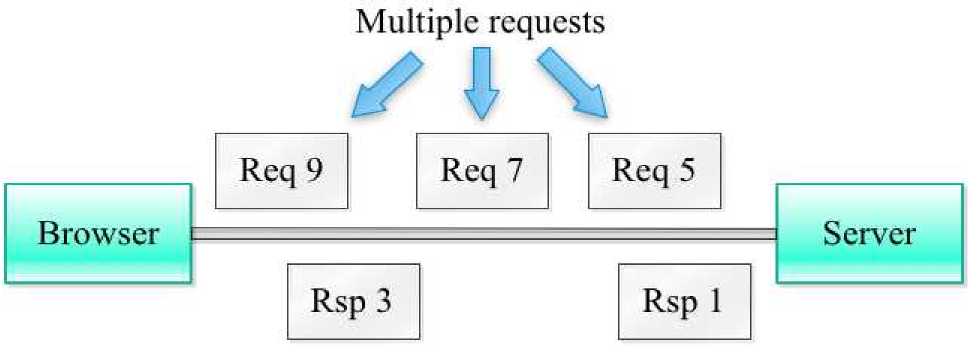
HTTP/2 solutions



HTTP/1.1	HTTP/2
Poor concurrency	High concurrency
Synchronous protocol	Asynchronous protorol
Redundant headers	Header compression

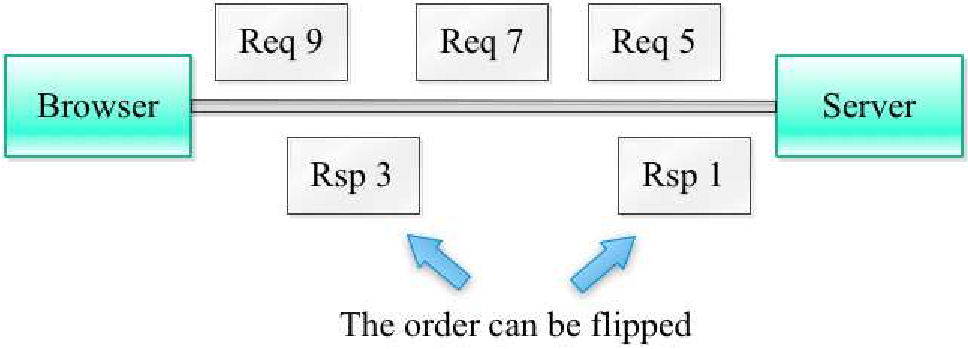
High concurrency

- Only one TCP connection
- Multiplexing frames up to 100 by default



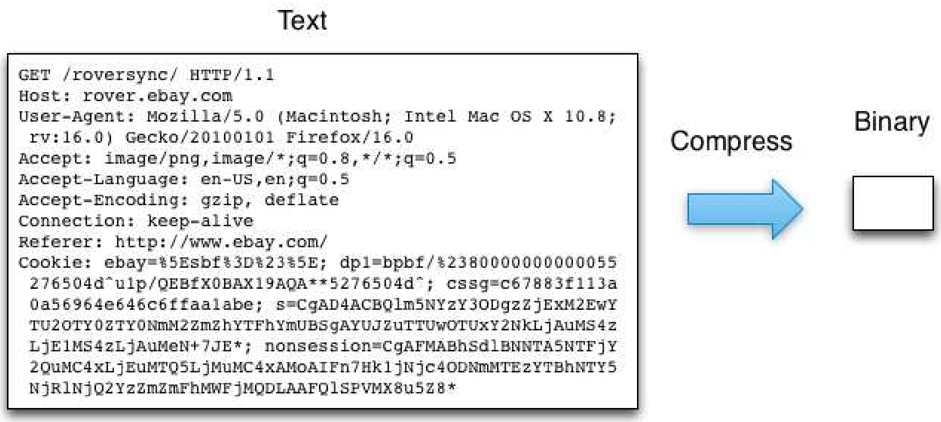
Asynchronous protocol

- Server can send responses in any order
  - Associating a response with a request by ID
  - Solving HTTP/1.1 head-of-line blocking



Header compression

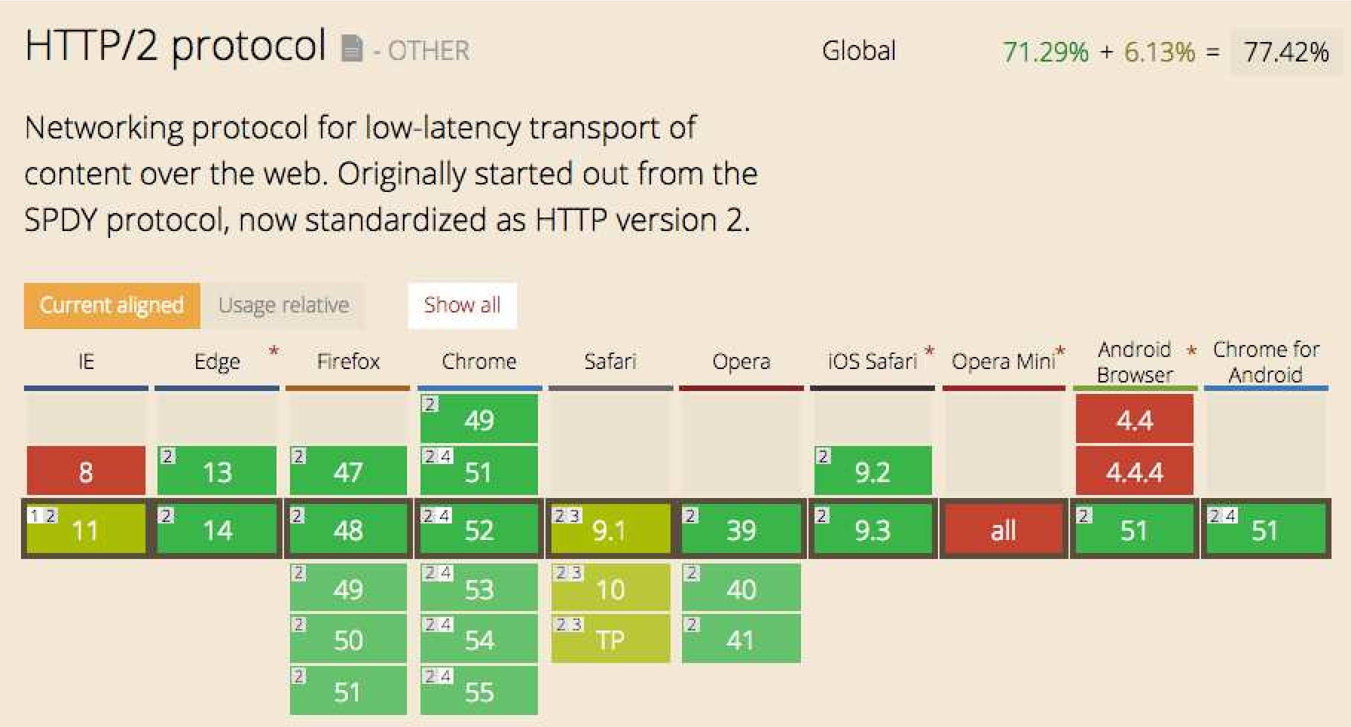
- Reducing about 70%



```
GET /roversync/ HTTP/1.1
Host: rover.ebay.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8;
 rv:16.0) Gecko/20100101 Firefox/16.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.ebay.com/
Cookie: ebay=%5Esbf%3D%23%5E; dp1=bpbf/%2380000000000055
 276504d^u1p/QEBfX0BAX19AQA**5276504d^; cssg=c67883f113a
 0a56964e646c6ffaa1abe; s=CgAD4ACBQlm5NYzY3ODgzZjExM2EwY
 TU2OTY0ZTY0NmM2ZmZhYTFhYmUBSgAYUJZuTTUwOTUxY2NkLjAuMS4z
 LjE1MS4zLjAuMeN+7JE*; nonsession=CgAFMABhSdlBNNTA5NTFjY
 2QuMC4xLjEuMTQ5LjMuMC4xAMoAIFn7Hk1jNjc4ODNmMTEzYTBhNTY5
 NjRlNjQ2YzZmZmFhMWFjMQDLAAFQlSPVMX8u5Z8*
```

Browser Status

- You are using HTTP/2



HTTP/2 protocol - OTHER

Global 71.29% + 6.13% = 77.42%

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

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			[2] 49					4.4	
8	[2] 13	[2] 47	[2] [4] 51			[2] 9.2		4.4.4	
[1] [2] 11	[2] 14	[2] 48	[2] [4] 52	[2] [3] 9.1	[2] 39	[2] 9.3	all	[2] 51	[2] [4] 51
		[2] 49	[2] [4] 53	[2] [3] 10	[2] 40				
		[2] 50	[2] [4] 54	[2] [3] TP	[2] 41				
		[2] 51	[2] [4] 55						

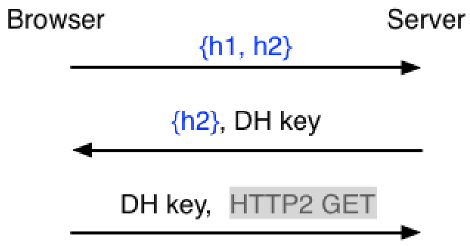
http://caniuse.com/#feat=http2

What I have done for HTTP/2

- Releasing HTTP/2 library in Haskell
  - Framing, header compression, priority
- Enhancing Warp
  - With HTTP/2 library
  - High performance on par with nginx
- Enhancing TLS library in Haskell

TLS is mandate for HTTP/2

- Major browsers requires TLS for HTTP/2
  - Resisting pervasive monitoring such as PRISM
  - Using ALPN (Application Layer Negotiation Protocol)



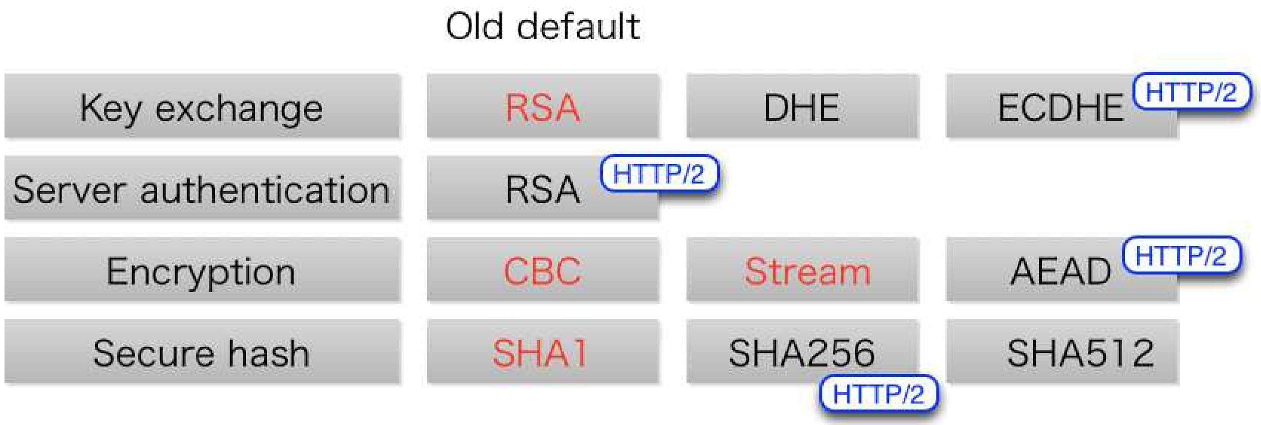
- HTTP/2 requires TLS 1.2 with modern technologies

TLS versions

	Defiend	Attack	Usage
SSL 2.0	1995	DROWN	Prohibited by RFC 6176
SSL 3.0	1996	POODLE	Prohibited by RFC 7568
TLS 1.0	1999	BEAST	No AEAD support
TLS 1.1	2006		No AEAD support
TLS 1.2	2008		AEAD support
TLS 1.3	Coming		AEAD support

- AEAD
  - Authenticated Encryption with Associated Data
  - Only secure encryption mode

TLS 1.2



	Old default		
Key exchange	RSA	DHE	ECDHE HTTP/2
Server authentication	RSA HTTP/2		
Encryption	CBC	Stream	AEAD HTTP/2
Secure hash	SHA1	SHA256 HTTP/2	SHA512

- Forward secrecy
  - DHE (Diffie Hellman, ephemeral)
  - ECDHE (Elliptic curve Diffie Hellman, ephemeral)

What I have done for TLS

- Enhancing TLS library in Haskell
  - ALPN (Application Layer Protocol Negotiation)
  - ECDHE(Elliptic curve Diffie Hellman, ephemeral)
  - AEAD (Authenticated Encryption with Associated Data)
    - AES GCM(Galois/Counter Mode)

Let’s Encrypt

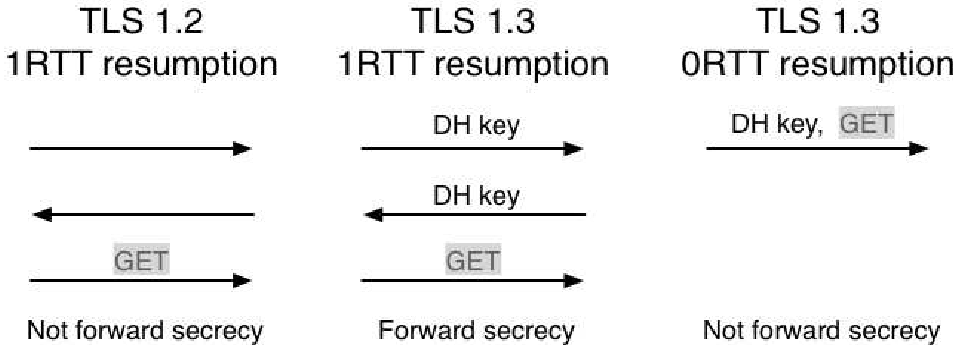
- You might hesitate to introduce HTTP/2
  - Misunderstanding: TLS certificates are costly
- TLS certificates are now free
  - You can get DV (Domain Validation) certificates
  - Not OV (Organization Validation)
  - Not EV (Extended Validation)

Conclusion

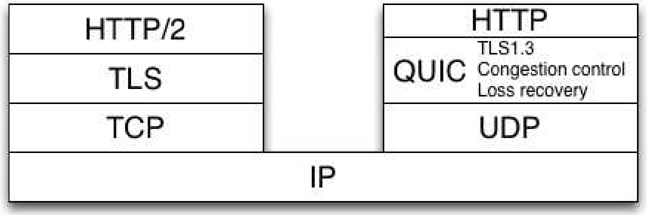
Use the latest Warp with TLS,
and your Web applications are HTTP/2 ready

What’s next?

- TLS 1.3 (would be renamed to TLS 2.0)
  - Integrating many extensions and simplified them



- QUIC
  - Fixing TCP’s head-of-line blocking



Future Reading

- Supporting HTTP/2
  - http://www.yesodweb.com/blog/2015/07/http2
- Getting Rating A from the SSL Server Test
  - http://www.yesodweb.com/blog/2015/08/ssl-server-test
- Implementing HTTP/2 server push
  - http://www.yesodweb.com/blog/2016/07/http2-server-push
- Experience Report: Developing High Performance HTTP/2 Server in Haskell
  - Haskell Symposium 2016