Upgrade your web to HTTP/3, or, better, don't?

Nick Shadrin
NGINX

OSCon, Portland
2019-07-18

Agenda

Intro

History

Stack

Benefits

Challenges

Conclusions

About the speaker

Nick Shadrin

Manager at F5 / NGINX

Previously: Product for NGINX Unit,
SE at NGINX, Zscaler, Citrix

4th time OSCon speaker

nick@nginx.com

Twitter: @shadrin

All links are here:

shadrin.org/talks

Basics

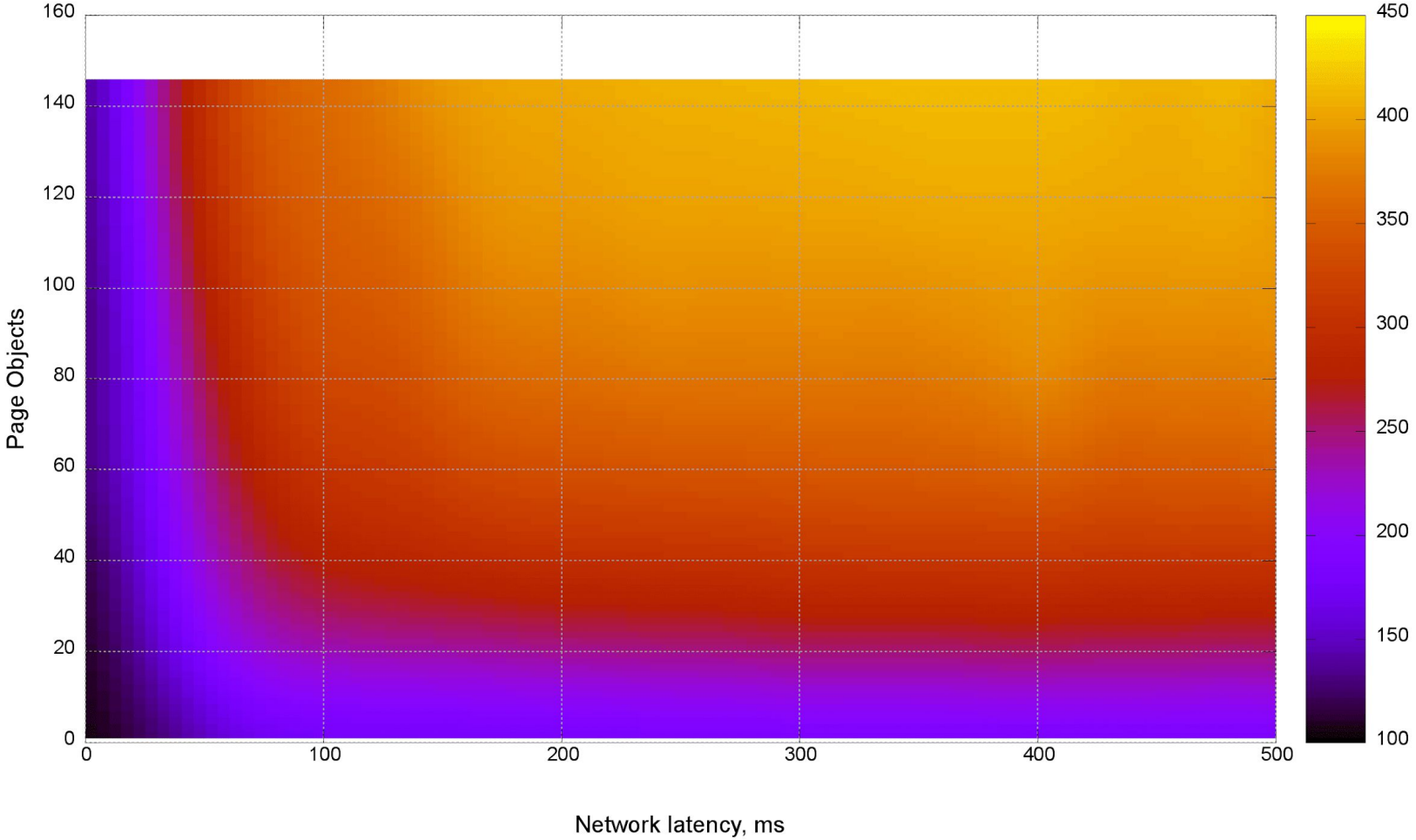
```
GET /test HTTP/1.1
Host: example.com
User-Agent: Mozilla
X-Forwarded-For: 192.168.10.1
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
```

```
HTTP/1.1 301 Moved Permanently
Server: unit/1.9
Date: Thu, 18 Jul 2019 21:19:07 GMT
Content-Type: text/html
Content-Length: 184
Connection: close
Location: https://example.com/test
```

History

HTTP Versions

```
HTTP/0.9: year 1991; one method

HTTP/1.0: year 1996; headers, caching, languages

HTTP/1.1: year 1999; keepalives, upgrades, ranges, hosts

SPDY:     year 2009; one tcp conn, gzip headers, flow control

HTTP/2:   year 2015; HPACK, priorities, server push
```


160
140
120
100
80
60
40
20
0
Page Objects
0
100
200
300
400
500
Network latency, ms
450
400
350
300
250
200
150
100

HTTP/2 usage, 2016

HTTP/2 usage, 2019

HTTP/2 facts

HTTP/1 is still available

Bots use HTTP/1

Search engines use HTTP/1

Internal connections are done via HTTP/1

HTTP/2 facts

HTTP/1 is still available

Bots use HTTP/1

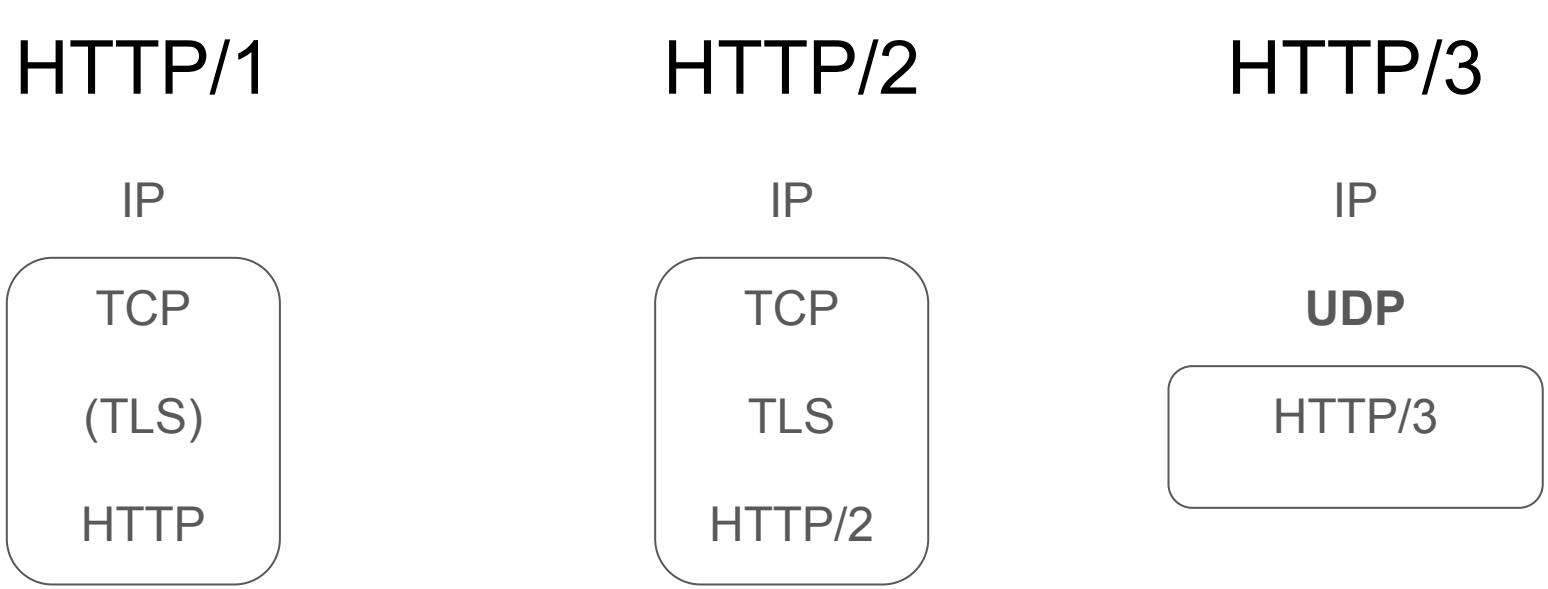
Search engines use HTTP/1

Internal connections are done via HTTP/1

Don't turn off HTTP/1!

Stack

HTTP/1	HTTP/2	HTTP/3
IP	IP	IP
TCP	TCP	**UDP**
(TLS)	TLS	HTTP/3
HTTP	HTTP/2	


HTTP/1
IP
TCP
(TLS)
HTTP
HTTP/2
IP
TCP
TLS
HTTP/2
HTTP/3
IP
UDP
HTTP/3

Benefits

Benefits

Less reliance on kernel

Built-in encryption

Connection ID: migrate connections

Faster negotiation*

Negotiation

Negotiation

HTTP to HTTPS: 3xx redirect, Meta, JavaScript

HTTP to HTTPS: HSTS headers

HTTP(s)/1 to Websocket: Upgrade header

HTTP/1 to HTTP/2: ~~Upgrade header, NPN,~~ ALPN (TLS negotiation)

HTTP/{1,2} to HTTP/3: Alt-Svc header

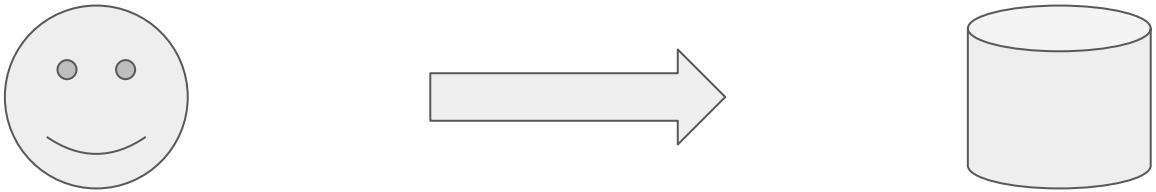
Alt-Svc header

```
Alt-Svc: h2="new.example.com:443"; ma=86400;

Alt-Svc: h3="newest.example.com:50781"; ma=86400;

Alt-Svc: h3=":50781"; ma=86400;
```

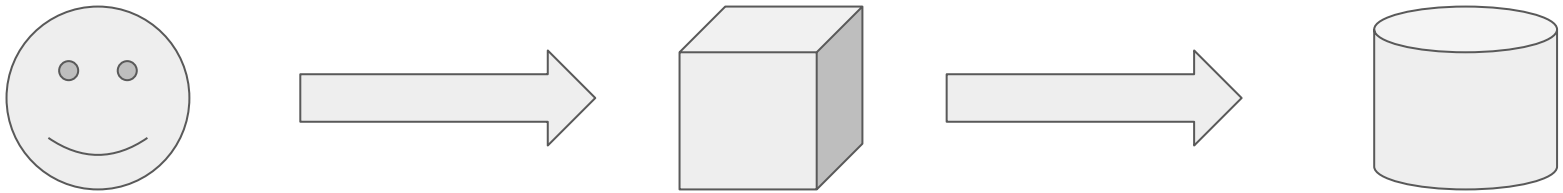
RFC draft: Servers MAY serve HTTP/3 on any UDP port, since an alternative always includes an explicit port.

Challenges

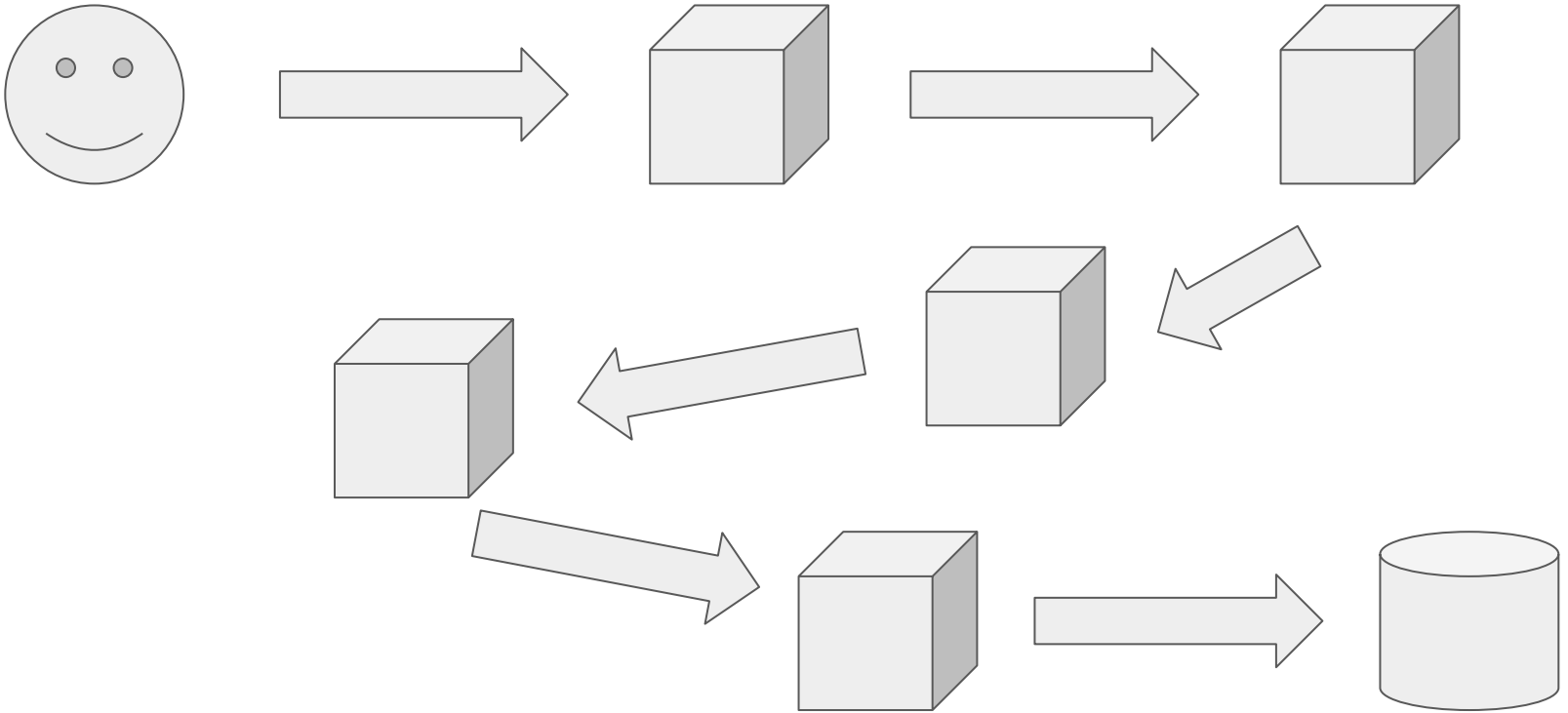
Infrastructure Challenges



Infrastructure Challenges



Infrastructure Challenges



Infrastructure Challenges

Hardware is tuned for old protocols

Slow upgrade cycles

Boxes are not yours

Requires significant effort between major Internet entities

Server engineer challenges

UDP stack is not optimized

Need to reimplement some features of TCP

Complicated multiprocessing

Tooling challenges

No plaintext version

Minimal debug tools

No visibility / monitoring

Security challenges

UDP is not trusted due to lots of recent "misuse"

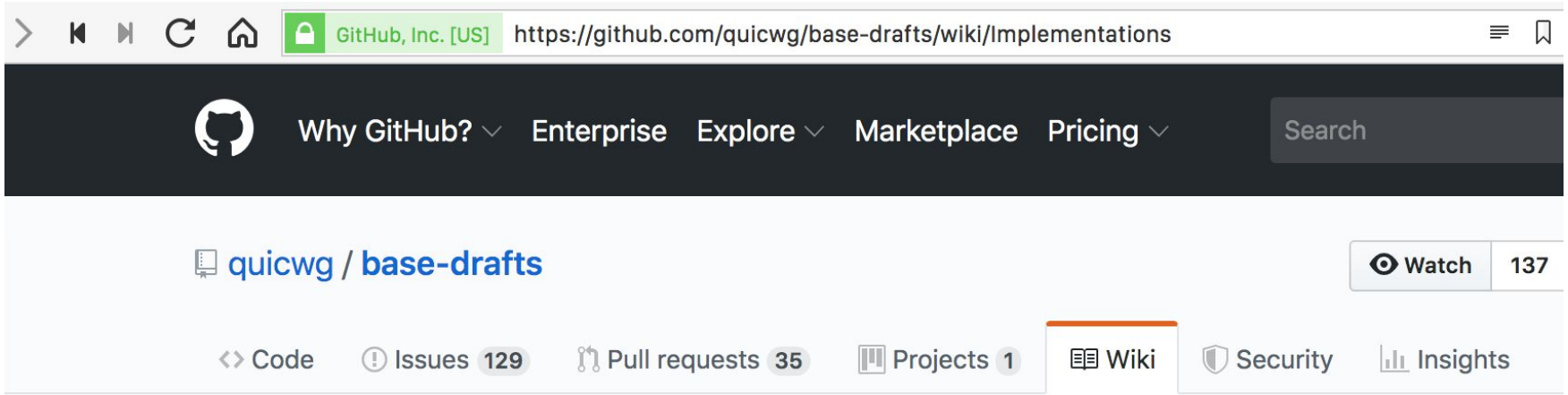
0-RTT replay and misconfiguration

Need to design new security devices

Conspiracy theories: Google owns both ends of HTTP/3

Agility of the protocol

Implementations



Links

shadrin.org/talks

quicwg.org/base-drafts/draft-ietf-quic-http.html

daniel.haxx.se/http3-explained/

w3techs.com/technologies/details/ce-http2/all/all

NGINX Conf 2019

Seattle

September 09-12, 2019

nginx.com/nginxconf