

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: December 1, 2014

M. Nottingham  
Akamai Technologies  
May 30, 2014

HTTP/2: Operational Considerations for Servers  
draft-nottingham-http2-ops-00

## Abstract

This document gives advice regarding performance and operability to servers deploying HTTP/2.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 1, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Notational Conventions . . . . .	<a href="#">3</a>
<a href="#">2.</a>	TCP Configuration . . . . .	<a href="#">3</a>
<a href="#">3.</a>	TLS Configuration . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Load Balancing and Failover . . . . .	<a href="#">4</a>
<a href="#">5.</a>	Use of HPACK . . . . .	<a href="#">4</a>
<a href="#">6.</a>	Use of Flow Control . . . . .	<a href="#">5</a>
<a href="#">7.</a>	Use of Prioritisation . . . . .	<a href="#">5</a>
<a href="#">8.</a>	Use of Server Push . . . . .	<a href="#">5</a>
<a href="#">9.</a>	Security Considerations . . . . .	<a href="#">5</a>
<a href="#">10.</a>	Acknowledgements . . . . .	<a href="#">5</a>
<a href="#">11.</a>	References . . . . .	<a href="#">5</a>
<a href="#">11.1.</a>	Normative References . . . . .	<a href="#">5</a>
<a href="#">11.2.</a>	Informative References . . . . .	<a href="#">5</a>
	Author's Address . . . . .	<a href="#">6</a>

[1.](#) Introduction

HTTP/2 [[I-D.ietf-httpbis-http2](#)] does not change the semantics of HTTP [[I-D.ietf-httpbis-p2-semantics](#)], but it does substantially change how they are mapped to the underlying protocols.

While some of these changes can enhance performance and/or operability on their own, getting the full benefit of the new protocol requires changes beyond the scope of just the Web server.

Likewise, HTTP/2 offers new in-protocol mechanisms like header compression, flow control, prioritisation and server push. Used effectively, they can improve the performance characteristics of the protocol, but they can also cause operability issues if used incorrectly.

This document gives advice about both cases; how to configure lower-layer protocols, as well as how to use HTTP/2's in-built mechanisms effectively.

It is primarily focused on the needs of origin servers, since there are generally many more instances of origin servers than there are unique client deployments. It is also primarily focused on the Web browsing use case; however, much of the advice here is applicable to non-browsing uses as well.

Note that the advice here is specific to when it was written; changes in underlying protocols, deployment practices, and HTTP itself may obsolete it at any time. As such, it is not intended to be long-lived, but instead to aid initial deployment of the new protocol.

### [1.1.](#) Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

## [2.](#) TCP Configuration

HTTP/2 has been designed to use a single TCP connection, whereas current practice for HTTP/1 is to use multiple connections to achieve parallelism (generally, between four and eight).

The change has a number of benefits. Using fewer TCP connections to load a Web page consumes less server-side resources, and it also reduces the chance of a congestion event caused by a large number of connections simultaneously starting (overcoming TCP Slow Start), and returns HTTP to being a "fair" protocol. Using a single connection also enables better efficiency with header compression.

However, using a single connection can also lead to unfavorable performance, as compared with HTTP/1's use of multiple connections, primarily due to side effects of TCP congestion control.

In particular, a single HTTP/2 connection with an `initcwnd` of 3 can only have three unacknowledged packets during the startup phase of the connection, whereas (for example) six HTTP/1 connections would have as many as 18 packets outstanding. This places HTTP/2 at a significant disadvantage compared to HTTP/1, but can be mitigated by adopting an `initcwnd` of 10 for HTTP/2 connections, as outlined in [[RFC6928](#)].

Similarly, a congestion event on a HTTP/2 connection can cause disproportionate havoc, as compared to HTTP/1, in those cases where the event only affects a subset of open connections (such as random packet loss). TBD: mitigation

Key recommendations: \* HTTP/2 servers SHOULD adopt an initcwnd of 10, as per [\[RFC6928\]](#).

### [3.](#) TLS Configuration

Beyond the typical performance and operational considerations of deploying TLS [\[RFC5246\]](#), a concern specific to HTTP/2 is the TLS record size; because HTTP/2 is a multiplexed protocol, a large record size can cause packet loss to affect a disproportionate number of streams, due to an individual record not being available until it is complete.

Nottingham

Expires December 1, 2014

[Page 3]

---

Internet-Draft

HTTP/2 Server Ops

May 2014

Therefore, small record sizes are preferred for HTTP/2; if a record is sent within a single packet, the chances of blocking are minimized. That said, records ought not be much smaller, since this will increase processing overhead, and in some circumstances (e.g., non-interactive applications, downloads), it may be reasonable to have larger record sizes.

Key recommendations: \* HTTP/2 servers SHOULD use a small TLS record size; ideally, small enough that a record fits completely in a single packet.

### [4.](#) Load Balancing and Failover

It's common to use multiple servers to server a single HTTP origin, in order to provide a scalable and reliable service. DNS is also commonly used to direct clients to the best (by some metric) server available.

In HTTP/1, the transition from one server to another in these scenarios is often done between connections; because HTTP/1 connections are generally short-lived, it's possible to load balance clients as they re-connect.

HTTP/2, however, is designed to have fewer, longer-lived connections, and it's anticipated that clients will be keeping them open much more aggressively. This provides fewer opportunities for servers to shift traffic. If a server breaks connections pre-emptively in order to load balance or failover, it can also have a greater negative effect, since more than one request can be "in flight" simultaneously.

The new protocol accommodates these situations in a few ways, improving operability along the way.

Firstly, the GOAWAY frame allows servers to announce that they will not serve additional requests on a connection, while still completing those that precede the GOAWAY. This allows a connection to be shut down in an orderly fashion, and its use is required in HTTP/2.

Additionally, the ALTSVC frame allows a server to redirect traffic to another location, without changing the resource's URL. This can be used for load balancing (both local and global), as well as controlled failover of services.

## [5.](#) Use of HPACK

TBD

Nottingham

Expires December 1, 2014

[Page 4]

---

Internet-Draft

HTTP/2 Server Ops

May 2014

## [6.](#) Use of Flow Control

TBD

## [7.](#) Use of Prioritisation

TBD

## [8.](#) Use of Server Push

TBD

## [9.](#) Security Considerations

This document introduces no unique security considerations beyond those discussed in HTTP/2 itself.

## [10.](#) Acknowledgements

TBD

## [11.](#) References

### 11.1. Normative References

- [I-D.ietf-httpbis-http2]  
Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", [draft-ietf-httpbis-http2-12](#) (work in progress), April 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC6928] Chu, J., Dukkupati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", [RFC 6928](#), April 2013.

### 11.2. Informative References

- [I-D.ietf-httpbis-p2- semantics]  
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [draft-ietf-httpbis-p2-semantics-26](#) (work in progress), February 2014.

#### Author's Address

Mark Nottingham  
Akamai Technologies

Email: [mnot@mnot.net](mailto:mnot@mnot.net)

URI: <http://www.mnot.net/>

