

- 
- [High Performance Browser Networking](#)
- [Comments Off](#)
- [Chapters](#)

#### Table of Contents

- [Foreword](#)
- [Preface](#)
  - [About This Book](#)
  - [Conventions Used in This Book](#)
  - [Safari® Books Online](#)
  - [How to Contact Us](#)
  - [Content Updates](#)
    - [May 23, 2014](#)
- [I. Networking 101](#)
  - [1. Primer on Latency and Bandwidth](#)
    - [Speed Is a Feature](#)
    - [The Many Components of Latency](#)
    - [Speed of Light and Propagation Latency](#)
    - [Last-Mile Latency](#)
    - [Bandwidth in Core Networks](#)
    - [Bandwidth at the Network Edge](#)
    - [Delivering Higher Bandwidth and Lower Latencies](#)
  - [2. Building Blocks of TCP](#)
    - [Three-Way Handshake](#)
    - [Congestion Avoidance and Control](#)
      - [Flow Control](#)
      - [Slow-Start](#)
      - [Congestion Avoidance](#)
    - [Bandwidth-Delay Product](#)
    - [Head-of-Line Blocking](#)
    - [Optimizing for TCP](#)
      - [Tuning Server Configuration](#)
      - [Tuning Application Behavior](#)
      - [Performance Checklist](#)
  - [3. Building Blocks of UDP](#)
    - [Null Protocol Services](#)
    - [UDP and Network Address Translators](#)
      - [Connection-State Timeouts](#)
      - [NAT Traversal](#)
      - [STUN, TURN, and ICE](#)
    - [Optimizing for UDP](#)
  - [4. Transport Layer Security \(TLS\)](#)
    - [Encryption, Authentication, and Integrity](#)
    - [TLS Handshake](#)
      - [RSA, Diffie-Hellman and Forward Secrecy](#)
      - [Application Layer Protocol Negotiation \(ALPN\)](#)
      - [Server Name Indication \(SNI\)](#)
    - [TLS Session Resumption](#)
      - [Session Identifiers](#)

Buy on oreilly.com

- [Session Tickets](#)
- [Chain of Trust and Certificate Authorities](#)
- [Certificate Revocation](#)
  - [Certificate Revocation List \(CRL\)](#)
  - [Online Certificate Status Protocol \(OCSP\)](#)
- [TLS Record Protocol](#)
- [Optimizing for TLS](#)
  - [Computational Costs](#)
  - [Early Termination](#)
  - [Session Caching and Stateless Resumption](#)
  - [TLS False Start](#)
  - [TLS Record Size](#)
  - [TLS Compression](#)
  - [Certificate-Chain Length](#)
  - [OCSP Stapling](#)
  - [HTTP Strict Transport Security \(HSTS\)](#)
- [Performance Checklist](#)
- [Testing and Verification](#)
- [II. Performance of Wireless Networks](#)
  - [5. Introduction to Wireless Networks](#)
    - [Ubiquitous Connectivity](#)
    - [Types of Wireless Networks](#)
    - [Performance Fundamentals of Wireless Networks](#)
      - [Bandwidth](#)
      - [Signal Power](#)
      - [Modulation](#)
    - [Measuring Real-World Wireless Performance](#)
  - [6. WiFi](#)
    - [From Ethernet to a Wireless LAN](#)
    - [WiFi Standards and Features](#)
    - [Measuring and Optimizing WiFi Performance](#)
      - [Packet Loss in WiFi Networks](#)
    - [Optimizing for WiFi Networks](#)
      - [Leverage Unmetered Bandwidth](#)
      - [Adapt to Variable Bandwidth](#)
      - [Adapt to Variable Latency](#)
  - [7. Mobile Networks](#)
    - [Brief History of the G's](#)
      - [First Data Services with 2G](#)
      - [3GPP and 3GPP2 Partnerships](#)
      - [Evolution of 3G Technologies](#)
      - [IMT-Advanced 4G Requirements](#)
      - [Long Term Evolution \(LTE\)](#)
      - [HSPA+ is Leading Worldwide 4G Adoption](#)
      - [Building for the Multigeneration Future](#)
    - [Device Features and Capabilities](#)
      - [User Equipment Category](#)
    - [Radio Resource Controller \(RRC\)](#)
      - [3G, 4G, and WiFi Power Requirements](#)
      - [LTE RRC State Machine](#)
      - [HSPA and HSPA+ \(UMTS\) RRC State Machine](#)

- [EV-DO \(CDMA\) RRC State Machine](#)
    - [Inefficiency of Periodic Transfers](#)
    - [End-to-End Carrier Architecture](#)
      - [Radio Access Network \(RAN\)](#)
      - [Core Network \(CN\)](#)
      - [Backhaul Capacity and Latency](#)
    - [Packet Flow in a Mobile Network](#)
      - [Initiating a Request](#)
      - [Inbound Data Flow](#)
    - [Heterogeneous Networks \(HetNets\)](#)
    - [Real-World 3G, 4G, and WiFi Performance](#)
  - [8. Optimizing for Mobile Networks](#)
    - [Preserve Battery Power](#)
    - [Eliminate Periodic and Inefficient Data Transfers](#)
      - [Eliminate Unnecessary Application Keepalives](#)
    - [Anticipate Network Latency Overhead](#)
      - [Account for RRC State Transitions](#)
      - [Decouple User Interactions from Network Communication](#)
    - [Design for Variable Network Interface Availability](#)
    - [Burst Your Data and Return to Idle](#)
    - [Offload to WiFi Networks](#)
    - [Apply Protocol and Application Best Practices](#)
- [III. HTTP](#)
  - [9. Brief History of HTTP](#)
    - [HTTP 0.9: The One-Line Protocol](#)
    - [HTTP/1.0: Rapid Growth and Informational RFC](#)
    - [HTTP/1.1: Internet Standard](#)
    - [HTTP/2: Improving Transport Performance](#)
  - [10. Primer on Web Performance](#)
    - [Hypertext, Web Pages, and Web Applications](#)
    - [Anatomy of a Modern Web Application](#)
      - [Speed, Performance, and Human Perception](#)
      - [Analyzing the Resource Waterfall](#)
    - [Performance Pillars: Computing, Rendering, Networking](#)
      - [More Bandwidth Doesn't Matter \(Much\)](#)
      - [Latency as a Performance Bottleneck](#)
    - [Synthetic and Real-User Performance Measurement](#)
    - [Browser Optimization](#)
  - [11. HTTP/1.X](#)
    - [Benefits of Keepalive Connections](#)
    - [HTTP Pipelining](#)
    - [Using Multiple TCP Connections](#)
    - [Domain Sharding](#)
    - [Measuring and Controlling Protocol Overhead](#)
    - [Concatenation and Spriting](#)
    - [Resource Inlining](#)
  - [12. HTTP/2](#)
    - [Brief History of SPDY and HTTP/2](#)
    - [Design and Technical Goals](#)
      - [Binary Framing Layer](#)
      - [Streams, Messages, and Frames](#)

- [Request and Response Multiplexing](#)
    - [Stream Prioritization](#)
    - [One Connection Per Origin](#)
    - [Flow Control](#)
    - [Server Push](#)
    - [Header Compression](#)
    - [Upgrading to HTTP/2](#)
    - [Brief Introduction to Binary Framing](#)
      - [Initiating a New Stream](#)
      - [Sending Application Data](#)
      - [Analyzing HTTP/2 Frame Data Flow](#)
  - [13. Optimizing Application Delivery](#)
    - [Optimizing Physical and Transport Layers](#)
    - [Evergreen Performance Best Practices](#)
      - [Cache Resources on the Client](#)
      - [Compress Transferred Data](#)
      - [Eliminate Unnecessary Request Bytes](#)
      - [Parallelize Request and Response Processing](#)
    - [Optimizing for HTTP/1.x](#)
    - [Optimizing for HTTP/2](#)
      - [Eliminate Domain Sharding](#)
      - [Minimize Concatenation and Image Spriting](#)
      - [Eliminate Roundtrips with Server Push](#)
      - [Test HTTP/2 Server Quality](#)
- [IV. Browser APIs and Protocols](#)
  - [14. Primer on Browser Networking](#)
    - [Connection Management and Optimization](#)
    - [Network Security and Sandboxing](#)
    - [Resource and Client State Caching](#)
    - [Application APIs and Protocols](#)
  - [15. XMLHttpRequest](#)
    - [Brief History of XHR](#)
    - [Cross-Origin Resource Sharing \(CORS\)](#)
    - [Downloading Data with XHR](#)
    - [Uploading Data with XHR](#)
    - [Monitoring Download and Upload Progress](#)
    - [Streaming Data with XHR](#)
    - [Real-Time Notifications and Delivery](#)
      - [Polling with XHR](#)
      - [Long-Polling with XHR](#)
    - [XHR Use Cases and Performance](#)
  - [16. Server-Sent Events \(SSE\)](#)
    - [EventSource API](#)
    - [Event Stream Protocol](#)
    - [SSE Use Cases and Performance](#)
  - [17. WebSocket](#)
    - [WebSocket API](#)
      - [WS and WSS URL Schemes](#)
      - [Receiving Text and Binary Data](#)
      - [Sending Text and Binary Data](#)
      - [Subprotocol Negotiation](#)

- [WebSocket Protocol](#)
  - [Binary Framing Layer](#)
  - [Protocol Extensions](#)
  - [HTTP Upgrade Negotiation](#)
- [WebSocket Use Cases and Performance](#)
  - [Request and Response Streaming](#)
  - [Message Overhead](#)
  - [Data Efficiency and Compression](#)
  - [Custom Application Protocols](#)
  - [Deploying WebSocket Infrastructure](#)
- [Performance Checklist](#)
- [18. WebRTC](#)
  - [Standards and Development of WebRTC](#)
  - [Audio and Video Engines](#)
    - [Acquiring Audio and Video with getUserMedia](#)
  - [Real-Time Network Transports](#)
    - [Brief Introduction to RTCPeerConnection API](#)
  - [Establishing a Peer-to-Peer Connection](#)
    - [Signaling and Session Negotiation](#)
    - [Session Description Protocol \(SDP\)](#)
    - [Interactive Connectivity Establishment \(ICE\)](#)
    - [Incremental Provisioning \(Trickle ICE\)](#)
    - [Tracking ICE Gathering and Connectivity Status](#)
    - [Putting It All Together](#)
  - [Delivering Media and Application Data](#)
    - [Secure Communication with DTLS](#)
    - [Delivering Media with SRTP and SRTCP](#)
    - [Delivering application data with SCTP](#)
  - [DataChannel](#)
    - [Setup and Negotiation](#)
    - [Configuring Message Order and Reliability](#)
    - [Partially Reliable Delivery and Message Size](#)
  - [WebRTC Use Cases and Performance](#)
    - [Audio, Video, and Data Streaming](#)
    - [Multiparty Architectures](#)
    - [Infrastructure and Capacity Planning](#)
    - [Data Efficiency and Compression](#)
  - [Performance Checklist](#)

◦ [Index](#)

• [Log In / Sign Up](#)

•

**Velocity**

Enjoy this free online version of *High Performance Browser Networking*.

Purchase and download the DRM-free ebook on [oreilly.com](#).

Learn more about the O'Reilly [Ebook Advantage](#).

Buy the Ebook

[Buy the print or ebook on oreilly.com](#)

Preface

[Prev](#)

[Next](#)

# Preface

The web browser is the most widespread deployment platform available to developers today: it is installed on every smartphone, tablet, laptop, desktop, and every other form factor in between. In fact, current cumulative industry growth projections put us on track for 20 billion connected devices by 2020—each with a browser, and at the very least, WiFi or a cellular connection. The type of platform, manufacturer of the device, or the version of the operating system do not matter—each and every device will have a web browser, which by itself is getting more feature rich each day.

The browser of yesterday looks nothing like what we now have access to, thanks to all the recent innovations: HTML and CSS form the presentation layer, JavaScript is the new assembly language of the Web, and new HTML5 APIs are continuing to improve and expose new platform capabilities for delivering engaging, high-performance applications. There is simply no other technology, or platform, that has ever had the reach or the distribution that is made available to us today when we develop for the browser. And where there is big opportunity, innovation always follows.

In fact, there is no better example of the rapid progress and innovation than the networking infrastructure within the browser. Historically, we have been restricted to simple HTTP request-response interactions, and today we have mechanisms for efficient streaming, bidirectional and real-time communication, ability to deliver custom application protocols, and even peer-to-peer videoconferencing and data delivery directly between the peers—all with a few dozen lines of JavaScript.

The net result? Billions of connected devices, a swelling userbase for existing and new online services, and high demand for high-performance web applications. Speed is a feature, and in fact, for some applications it is *the feature*, and delivering a high-performance web application requires a solid foundation in how the browser and the network interact. That is the subject of this book.

## About This Book

Our goal is to cover what every developer should know about the network: what protocols are being used and their inherent limitations, how to best optimize your applications for the underlying network, and what networking capabilities the browser offers and when to use them.

In the process, we will look at the internals of TCP, UDP, and TLS protocols, and how to optimize our applications and infrastructure for each one. Then we'll take a deep dive into how the wireless and mobile networks work under the hood—this radio thing, it's very different—and discuss its implications for how we design and architect our applications. Finally, we will dissect how the HTTP protocol works under the hood and investigate the many new and exciting networking capabilities in the browser:

- Upcoming HTTP/2 improvements
- New XHR features and capabilities
- Data streaming with Server-Sent Events
- Bidirectional communication with WebSocket
- Peer-to-peer video and audio communication with WebRTC
- Peer-to-peer data exchange with DataChannel

Understanding how the individual bits are delivered, and the properties of each transport and protocol in use are essential knowledge for delivering high-performance applications. After all, if our applications are blocked waiting on the network, then no amount of rendering, JavaScript, or any other form of optimization will help! Our goal is to eliminate this wait time by getting the best possible performance

from the network.

*High-Performance Browser Networking* will be of interest to anyone interested in optimizing the delivery and performance of her applications, and more generally, curious minds that are not satisfied with a simple checklist but want to know how the browser and the underlying protocols actually work under the hood. The "how" and the "why" go hand in hand: we'll cover practical advice about configuration and architecture, and we'll also explore the trade-offs and the underlying reasons for each optimization.

Our primary focus is on the protocols and their properties with respect to applications running in the browser. However, all the discussions on TCP, UDP, TLS, HTTP, and just about every other protocol we will cover are also directly applicable to native applications, regardless of the platform.

## Conventions Used in This Book

The following typographical conventions are used in this book:

### *Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

### `Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

### **Constant width bold**

Shows commands or other text that should be typed literally by the user.

### *Constant width italic*

Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.

This icon indicates a warning or caution.

## Safari® Books Online

[Safari Books Online](#) is an on-demand digital library that delivers expert [content](#) in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of [product mixes](#) and pricing programs for [organizations](#), [government agencies](#), and [individuals](#). Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens [more](#). For more information about Safari Books Online, please visit us [online](#).

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472  
800-998-9938 (in the United States or Canada)  
707-829-0515 (international or local)  
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://oreil.ly/high-performance-browser>.

To comment or ask technical questions about this book, send email to [<bookquestions@oreilly.com>](mailto:bookquestions@oreilly.com).

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

## Content Updates

**May 23, 2014**

1. Added new section on using TLS False Start to optimize TLS handshake.
2. Added new section on benefits of TLS Forward Secrecy.
3. Updated TLS record size optimization with recommendation to use dynamic record sizing.
4. Updated WebRTC code examples to use latest authentication and callback syntax.
5. Updated SPDY roadmap reference to 2014 timelines.
6. Fixed odd/even stream ID references in [Chapter 12](#).
7. Fixed spelling mistakes in text and diagrams.

---

[Prev](#)

Foreword

[Home](#)

[Next](#)

Part I. Networking 101

© 2013, O'Reilly Media, Inc.

- [Terms of Service](#)
- [Privacy Policy](#)
- Interested in [sponsoring content?](#)