High Performance WebSocket

@wesleyhales
Real Time Enablers

- HTTP/1.1: 1999
- REST: 2000
- AJAX: 2005
- SSE: 2006
- WebSocket: 2008
- SPDY: 2009
- HTTP/2: 2015
In 2015, bidirectional client <-> server communication can only be achieved with AJAX and/or WebSocket
Client Server Connectivity

Figure 17-2. Communication flow of XHR, SSE, and WebSocket

High Performance Browser Networking by @igvita
When you make an AJAX request from the browser you don’t have to think about TCP.

The underlying complexity is abstracted away.
WebSocket

When you send a message over a WebSocket connection, you have to think about everything.

Subprotocols, Fallbacks, Network
You might not need a WebSocket

Posted by justin on June 24th, 2014 — Filed under Protocols, WebSockets (Permalink)

Before I begin, I want to say that WebSockets are great. I’ve even implemented RFC 6455 myself in Zurl and Pushpin, which are used by the Fanout.io service. Fanout.io also supports WebSockets via its XMPP-FTW interface using Primus.

However, after spending quite some time working on large distributed applications and gaining a greater appreciation of REST and messaging patterns, I feel that much of what typical web applications want to accomplish with WebSockets (or with socket-like abstractions) is perhaps better solved by other means.

HTTP streaming and Server Sent Events

WebSockets aren't the only game in town when it comes to efficiently pushing data to browsers. Consuming an HTTP response of indefinite length with

http://blog.fanout.io/2014/06/24/you-might-not-need-a-websocket/
Real Time?

- Rapid client <-> server communication
- Real time execution of events in browser
- Binary data delivery
WebSocket
The Promise Idea

“facilitating live content and the creation of real-time games”
So let’s give that a try…

connect now onskyde.com/go/619
High Performance WS

Open Connection

Punch Through Proxies

Process and/or Respond

Live Demo

@wesleyhales
High Performance Hurdles

• Server Side Connections
• Networks and Proxies
• Subprotocols
• Time
• Services vs. Frameworks
Server Side Connections

- Up to 1M concurrent connections
- Low Overhead
- Bidirectional
Networks and Proxies
Networks & Proxies

TL;DR Use TLS
99.9% Guaranteed Connection
Networks

Figure 7-4. Radio Resource Controller

High Performance Browser Networking by @igvita
Networks

Figure 7-14. Heterogeneous network infographic (Ericsson)
## Networks

<table>
<thead>
<tr>
<th>Service Provider</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>tw telecom holdings inc.</td>
<td>282</td>
</tr>
<tr>
<td>service provider corporation</td>
<td>52</td>
</tr>
<tr>
<td>(not set)</td>
<td>37</td>
</tr>
<tr>
<td>t-mobile usa inc.</td>
<td>13</td>
</tr>
<tr>
<td>cellco partnership dba verizon wireless</td>
<td>6</td>
</tr>
<tr>
<td>softlayer technologies inc.</td>
<td>5</td>
</tr>
<tr>
<td>sprint nextel corporation</td>
<td>5</td>
</tr>
<tr>
<td>comcast cable communications holdings inc</td>
<td>4</td>
</tr>
<tr>
<td>agile-inap</td>
<td>3</td>
</tr>
<tr>
<td>vodafone limited</td>
<td>2</td>
</tr>
</tbody>
</table>

@wesleyhales
Subprotocols
Because one isn’t enough
Subprotocols

Meta data for the client <-> server transmission

```javascript
var ws = new WebSocket('wss://example.com/socket',
    ['appProtocol', 'appProtocol-v2']);

ws.onopen = function () {
    if (ws.protocol == 'appProtocol-v2') {
        ...
    } else {
        ...
    }
}

*No effect on the core WebSocket API*
Messaging Protocol

WHAM!
LAST CHRISTMAS

@wesleyhales
Subprotocols

You have to use an envelope e.g. JSON
Time

Why should you care?
Time

Client Side Synchronization

• Headers vs. Content rewrite
• Resource Timing on AJAX
Get the data from an AJAX request in 4 easy steps

```javascript
var serverTimeMillisUTC;
var oReq = new XMLHttpRequest();
oReq.onload = function() {
    var dateStr = oReq.getResponseHeader('Date');
    serverTimeMillisUTC = new Date(Date.parse(dateStr)).getTime();
};
oReq.open("HEAD", "/?foo123", false);
oReq.send();

function getServerTime() {
    for (tvalue in window.performance.getEntriesByType("resource")) {
        tresource = window.performance.getEntriesByType("resource")[tvalue];
        if (tresource.name.indexOf('foo123') > 0) {
            return serverTimeMillisUTC + (tresource.responseEnd - tresource.responseStart);
        }
    }
}
```
Time

Why would I do this?

What is response time?
Time

1. Get the date header from the server

```javascript
var oReq = new XMLHttpRequest();
oReq.onload = function(){
    var dateStr = oReq.getResponseHeader('Date');
    serverTimeMillisUTC = new Date(Date.parse(dateStr)).getTime();
};
oReq.open("HEAD", "/?foo123", false);
oReq.send();
```

2. User random query param
3. Filter on random query param

```javascript
if(tresource.name.indexOf('foo123') > 0){
    return serverTimeMillisUTC +
    (tresource.responseEnd -
    tresource.responseStart);
}
```

4. Calculate Server Time
What Time Is It?

Edge 4 (Data)
Edge 2 (Seek)
Services and Frameworks
Service or Homegrown?

Services
- PubNub
- KAAZING
- PusHer
- Akamai

Frameworks
- socket.io

Servers
- NGINX
- Apache
Tips
Head of Line Blocking

Large messages block frame delivery

Latency Sensitive Data? Split up the large messages

Screenshots are sent as binary data over wss:// in onskyde’s dashboard

client side code on github

@wesleyhales
Compression

Extensions are managed by the browser
(spec says it’s a “framework”)

wss:// connection headers

```
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Key: /LqLDugB0+EK9Bnp6d3gqw==
Sec-WebSocket-Version: 13
```

1) Chrome has implicit support (no config on client)
2) Compresses any message over 10 bytes
Make Sure You Need It

Do you need WebSocket?

HTTP2, SPDY, SSE, and WebSocket are all different. They could be combined.

Are you willing to polyfill your WebSocket Implementation?
Thanks!
Appendix
Browser Event Firing

All Events Fire onmessage

```javascript
onmessage: function (m) {
  var event = (m.data);
  event = (new Function("return " + event))();
  event.onslydeEvent.fire();
}
```

Remote Control listens

```javascript
window.addEventListener('remoteMarkup', function(e) {
  var markup = JSON.parse(e.markup);
  document.getElementById('from-slide').innerHTML =
   decodeURIComponent(markup.remoteMarkup);
}, false);
```
HTTP/2

Demo 1
Demo 2