HTML5 WebSocket

Sang Shin
Founder and Chief Instructor
JPassion.com
"Learn with Passion!"

Topics

- What is and Why WebSocket?
- WebSocket protocol
- WebSocket Client API
- Server implementations
- WebSocket subprotocols

What is and Why WebSocket?

Characteristics of Modern Web Apps

- Collaborative
  - Many participants access the same app
- Participatory content generation
  - Participants post contents
- Real-time
  - Contents are published and distributed real-time
- Multi-device & Mobile-friendly
  - Apps are accessed through many client types: Desktop browser, Tablet, Mobile phone, TV, etc

Modern Web Apps

- Online gaming
- Distance learning
- Collaborative authoring
- Auctions
- Shared WebDAV filesystem
- Blogging and reader comments
- SIP-coordinated mobile applications
- Hybrid chat/email/discussion forums
- Customer assistance on sales/support pages
- Multi-step business process made collaborative
- Shared trip planner or restaurant selector with maps
- Shared calendar, “to do” list, project plan

Requirements of Modern Web Apps

- Modern web apps demand the following features
  - Reliable communication
  - Real-time (event-driven) communication
  - Bi-directional communication
  - Minimum latency
  - Minimum overhead

Problems of HTTP

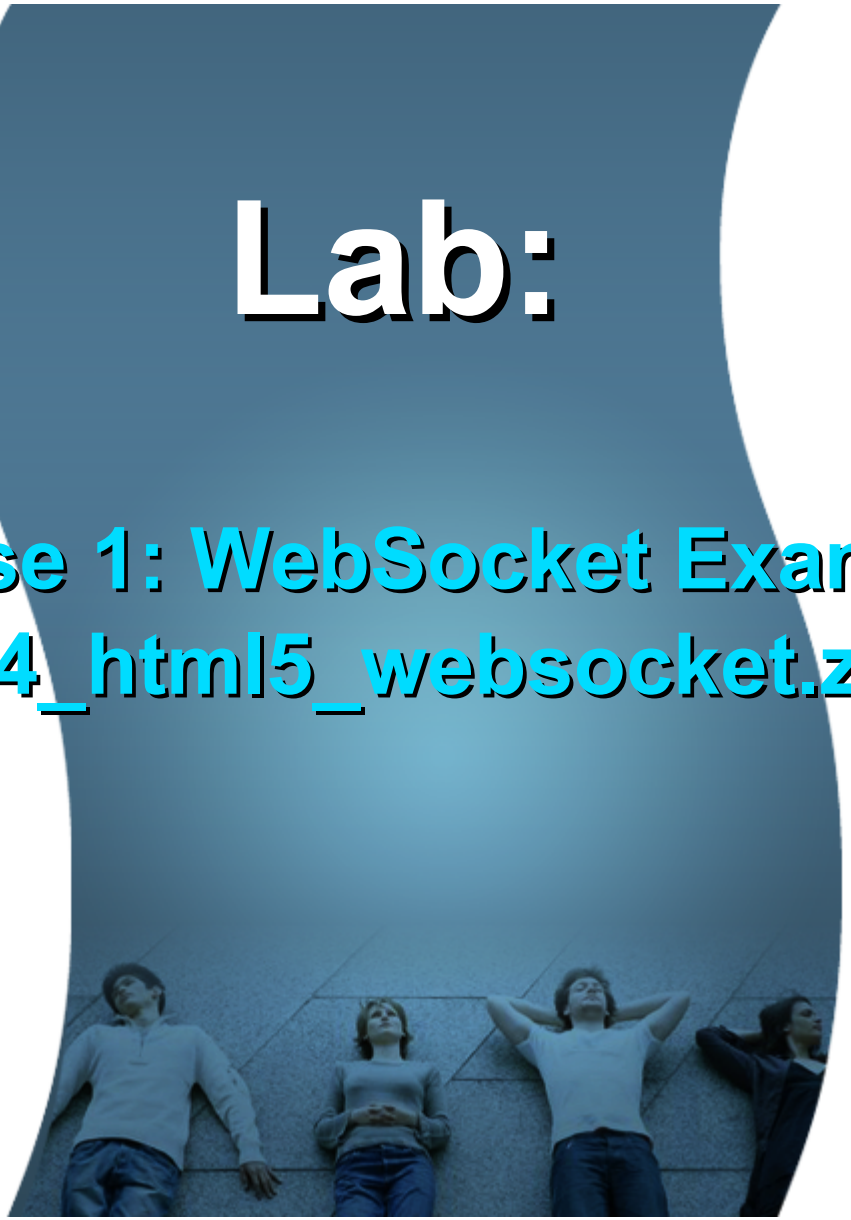
- It is half-duplex
  - Traffic flows in only one direction at a time
- It has high overhead
  - Header data is sent with each HTTP request and response
- It is slow
  - Higher overhead results in slow communication
- It requires complex hacks to achieve real-time & bi-directional communication
  - Without a hack like Comet, clients have to poll the server, which results long latency

Complex Hacks to Achieve Real-time

- Polling (Ajax)
  - Poll server for updates
  - Client has to wait
- Long polling (Comet)
  - Uses two connections
  - Adds complexity to both client and server

Lab:

Exercise 1: WebSocket Examples
1234_html5_websocket.zip

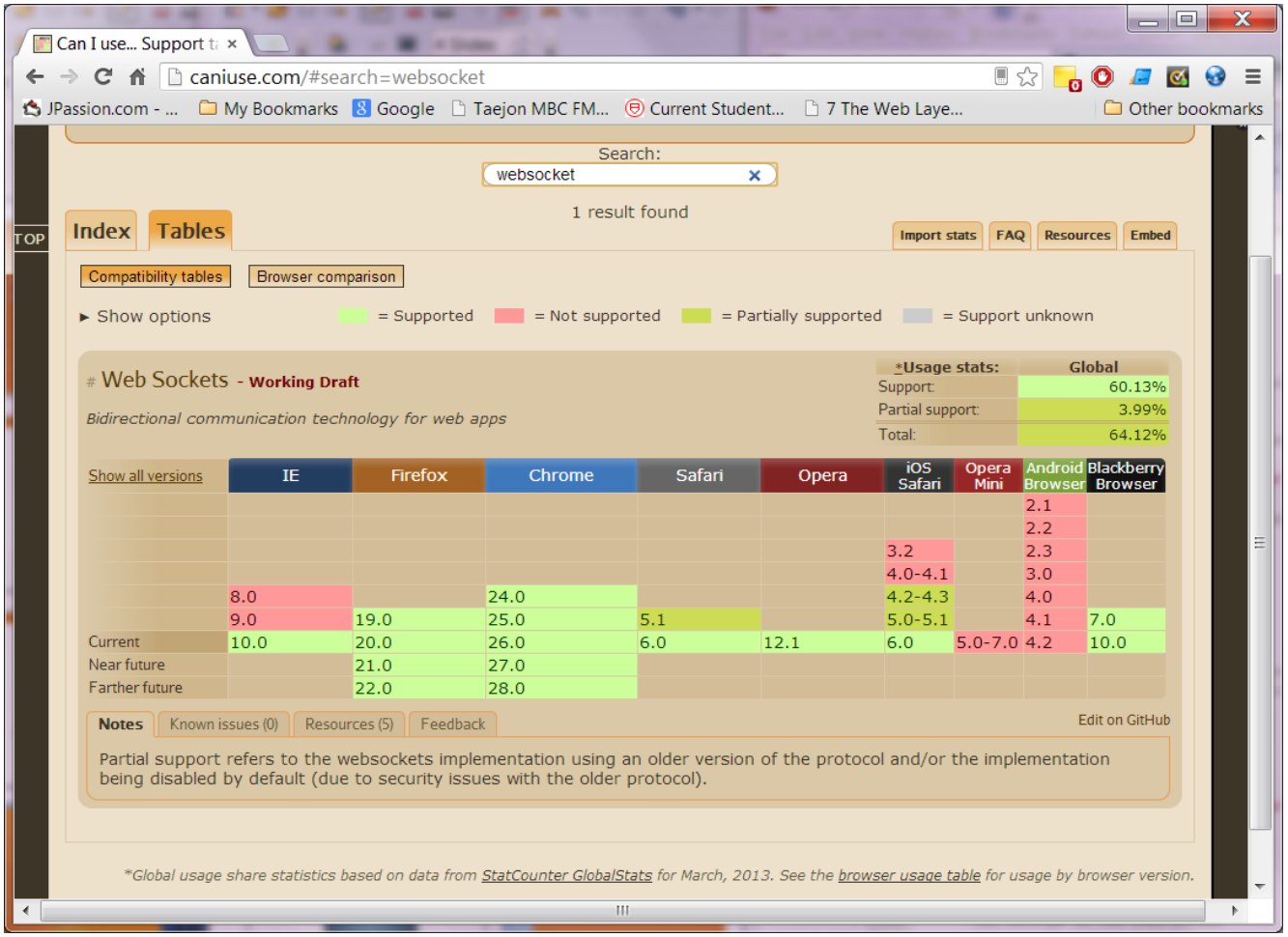


What is and Why WebSocket?

- Full-duplex bidirectional communication channel over a single socket
  - Both client and server can send data at any time at the same time
- Much lower overhead
  - Data is sent without the overhead of HTTP headers
- Works with firewalls/proxies
  - HTTP-compatible handshake
  - Integrates with Cookie based authentication
- Replaces Comet
  - No more complexity needed for server sending data to client
- Cross domain capable through CORS
  - Secure

WebSocket Support (May 2013)

- http://caniuse.com/#search=websocket



WebSocket Protocol

Connection Establishment Scheme

- Connection gets established by “upgrading” from HTTP to WebSocket protocol
  - Using the same TCP connection
- WebSockets and Secure WebSockets
  - ws://
  - wss://

WebSocket has Low Overhead

- Each message (“frame”) has only 2 bytes of overhead (for messages whose length is 126 bytes and under)
  - Reducing kilobytes of overhead to 2 bytes
- No latency in establishing new TCP connections for each HTTP message
  - Reducing 150ms to 50ms
- No polling overhead
  - Client and server sends messages only when there is something to send

Upgrade Client/Server Handshake

- Request Headers

```
Cache-Control:no-cache
Connection:Upgrade
Cookie:__utma=9925811.1883938102.1360068195.1365797484.1366590814.7;
__utmb=9925811.2.10.1366590814; __utmc=9925811;
__utmz=9925811.1366590814.7.4.utmcsr=websocket.org|utmccn=(referral)|
utmcmd=referral|utmcct=/
Host:echo.websocket.org
Origin:http://www.websocket.org
Pragma:no-cache
Sec-WebSocket-Extensions:x-webkit-deflate-frame
Sec-WebSocket-Key:1Su2ADJUEnTFfyDDb8JEHQ==
Sec-WebSocket-Version:13
Upgrade:websocket
```

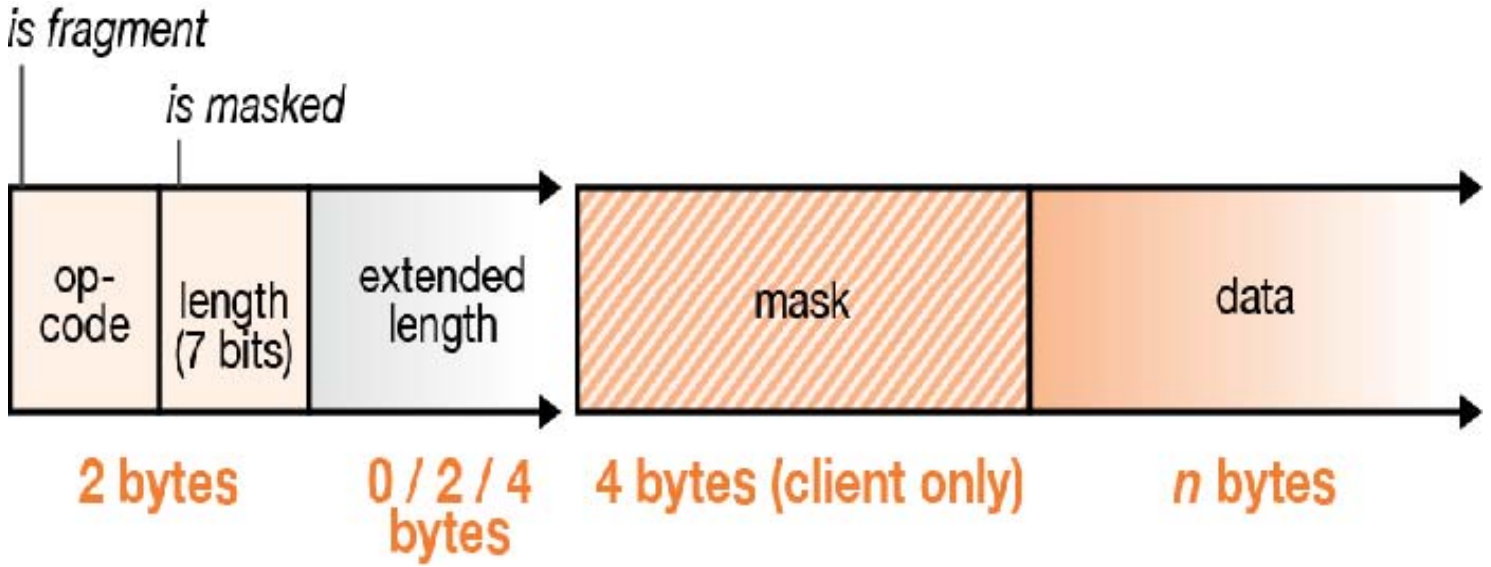
Upgrade Client/Server Handshake

- Response Headers

```
Access-Control-Allow-Credentials:true
Access-Control-Allow-Headers:content-type
Access-Control-Allow-Origin:http://www.websocket.org
Connection:Upgrade
Date:Mon, 22 Apr 2013 00:32:17 GMT
Sec-WebSocket-Accept:98Icoamxq4bM7ZdeSogKsFAm4xc=
Server:Kaazing Gateway
Upgrade:WebSocket
```

Data Frame Structure

- “is fragment” bit indicates if this fragment is final fragment
- op-code (4 bit)
  - 1 (text), 2 (binary), 8 (close), 9 (ping), 10 (pong)
- “is masked” bit indicates if this frame is masked



Lab:

Exercise 2: WebSocket Handshake
1234_html5_websocket.zip

WebSocket Client API

WebSocket API: Socket Creation

```
// Create a WebSocket
var wsocket = new WebSocket('ws://websocket.jpassion.com/echo');

// Create a WebSocket with optional higher-layer protocol
var wsocket = new WebSocket('ws://websocket.jpassion.com/echo',
                            'stomp');

// Create a WebSocket with optional higher-layer protocols
var wsocket = new WebSocket('ws://websocket.jpassion.com/echo',
                            [ 'stomp', 'myprotocol']);

// Create a WebSocket with SSL
var wsocket = new WebSocket('wss://websocket.jpassion.com/echo');
```

WebSocket API: Event Handlers

```
// Event handler for successful open
wsocket.onopen = function(event) {
   wsocket.send('Hello, WebSocket');
};

// Event handler for received message
wsocket.onmessage = function(event) {
   console.log(event.data);
}

// Event handler for closing the connection
wsocket.onclose = function(event) {
   console.log('closed');
}

wsocket.onerror = function(error) {
   console.log('error:' + error);
}
```

WebSocket API: Methods

```
// Send text message
wsocket.send("Hello world!");

// Send a binary message
var blob = new Blob("blob contents");
wsocket.send(blob);

// Close a connection
wsocket.close();
```

Lab:

Exercise 3: WebSocket API – echo client
1234_html5_websocket.zip

Server Implementations

Server Implementations

- Java
  - JavaEE 7
  - JWebSocket
  - Jetty WebSocketServlet
- PHPWebSockets
- Node.js with WebSocket plugin
  - https://github.com/Worlize/WebSocket-Node
- ...

Lab:

Exercise 4: WebSocket Server (Node.js)
Exercise 5: Chat application
1234_html5_websocket.zip

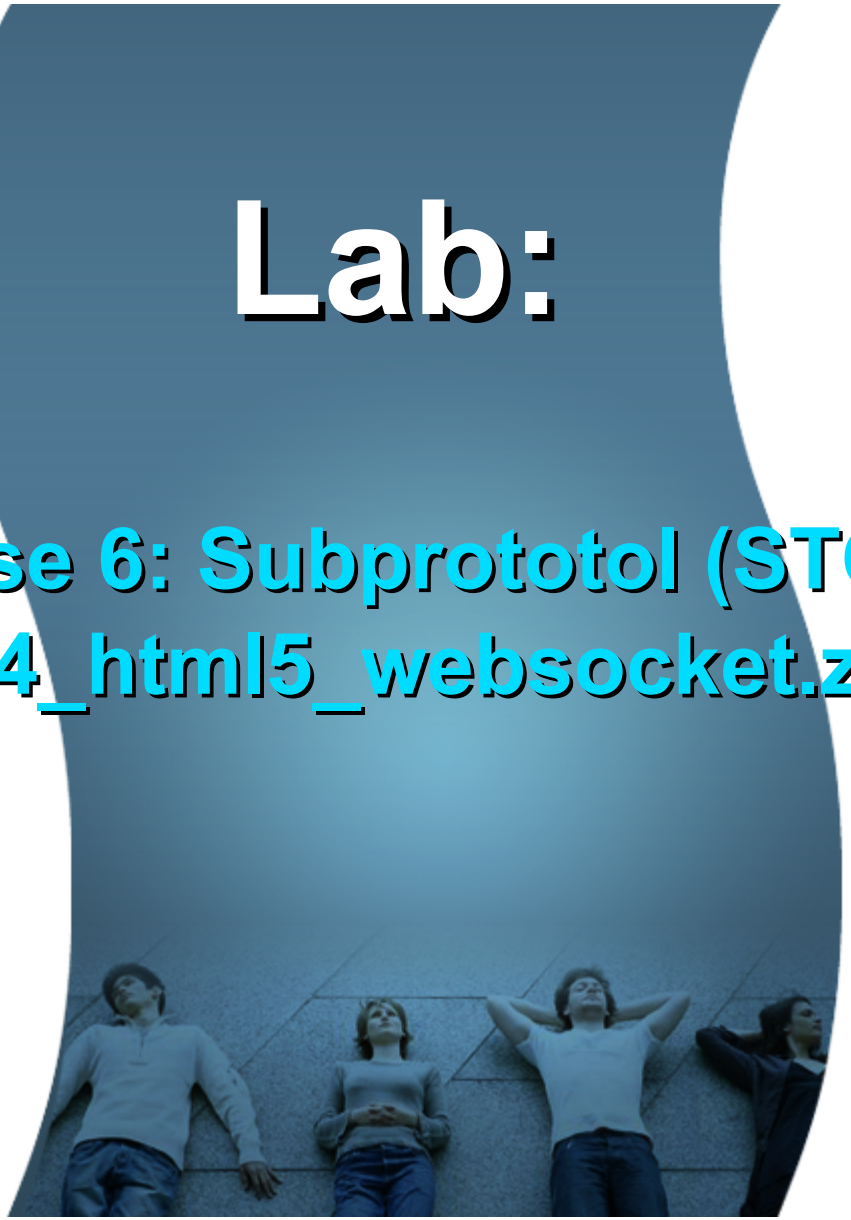
WebSocket Subprotocols

Examples of Subprotocols

- XMPP (Instant messaging)
  - Jabber
- Pub/Sub (Messaging)
  - Stomp
  - AMQP
- Custom subprotocols

Lab:

Exercise 6: Subprototol (STOMP)
1234_html5_websocket.zip



Learn with Passion!
JPassion.com