Le protocole applicatif WebSocket

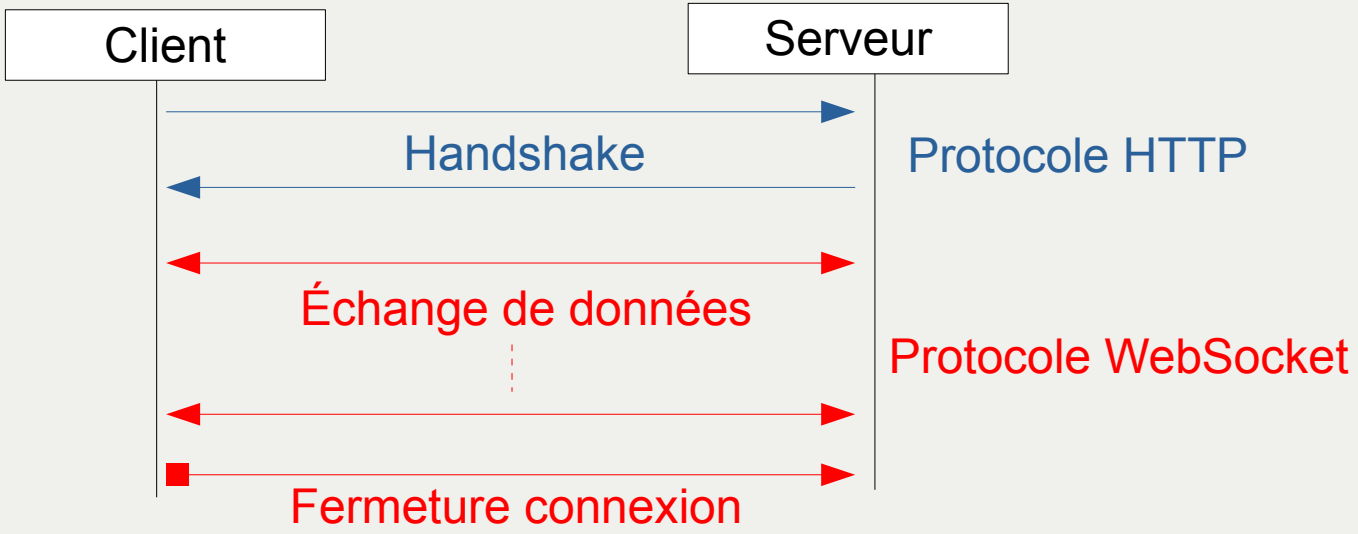
Nicolas Le Sommer
Nicolas.Le-Sommer@univ-ubs.fr
Université Bretagne Sud, IUT de Vannes, Département Informatique

Présentation du protocole WebSocket

- Le protocole WebSocket est normalisé par l'IETF dans la RFC 6455 en 2011
- L'interface de programmation est standardisée par le W3C
- Protocole de niveau application assurant une communication bidirectionnelle et full duplex entre le client et le serveur
  - Permet au serveur de notifier le client d'un changement de son état
  - Permet au serveur de pousser de l'information vers le client
- Repose sur une connexion TCP
- Pas d'en-têtes dans les messages
- Permet de développer des applications quasi temps réel pour émettre et recevoir des données

Fonctionnement du protocole WebSocket

- 2 phases dans le protocole
  - requête/réponse HTTP avec l'option upgrade pour changer de protocole (handshake)
  - Échange de données au format texte ou binaire (format d'échange libre)



Fonctionnement du protocole WebSocket

- C’est toujours le client qui initie la demande d’ouverture d’une WebSocket
- Une WebSocket est identifiée par une URI particulière

```
ws(s)://host[:port]/path[?param]
```

Exemple de requête

```
GET ws://echo.websocket.org/ HTTP/1.1
Host: echo.websocket.org
Connection: Upgrade
Pragma: no-cache
Cache-Control: no-cache
Upgrade: websocket
Accept-Encoding: gzip, deflate, sdch
Accept-Language: fr,en-US;q=0.8,en;q=0.6
Sec-WebSocket-Key: mWVGwS9Z6VXWMAsDXSGH7g==
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
```

Exemple de réponse

```
HTTP/1.1 101 Web Socket Protocol Handshake
Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: content-type, x-websocket-version, x-websocket-protocol
Connection: Upgrade
Date: Mon, 21 Nov 2016 20:42:32 GMT
Sec-WebSocket-Accept: +Uy0ajDrCqbMUyxNnCxY0J3vlvM=
Server: Kaazing Gateway
Upgrade: websocket
```

API Javascript : Exemple de création d'une WebSocket

```
// Create a new WebSocket.
var socket = new WebSocket('ws://echo.websocket.org');

// Show a connected message when the WebSocket is opened.
socket.onopen = function(event) { console.log('WebSocket opened') ; };

socket.onclose = function(event) { console.log('WebSocket closed') ; };

// Handle any errors that occur.
socket.onerror = function(error) { console.log('WebSocket Error: ' + error); };

// Handle messages sent by the server.
socket.onmessage = function(event) {
  var message = event.data;
  alert(message) ;
};
// send a message
socket.send('My WebSocket');

// close the socket
socket.close() ;
```

Socket.io : alternative à l'API WebSocket

- Framework Javascript client et serveur (https://socket.io/)
- Protocole s'appuyant sur des WebSockets

```
var io = require('socket.io').listen(80);

io.sockets.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

Serveur

Client

```
<script src="/socket.io/socket.io.js"></script>
<script>
  var socket = io.connect('http://localhost');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('hello', { my: 'data' });
  });
</script>
```

Bibliographie

- RFC
  - https://tools.ietf.org/html/rfc6455
  - http://www.w3.org/TR/websockets/
  - https://developer.mozilla.org/fr/docs/Web/API/WebSockets_API