



Websocket vs REST



[Mise à jour le 23/2/2020]

Ressources

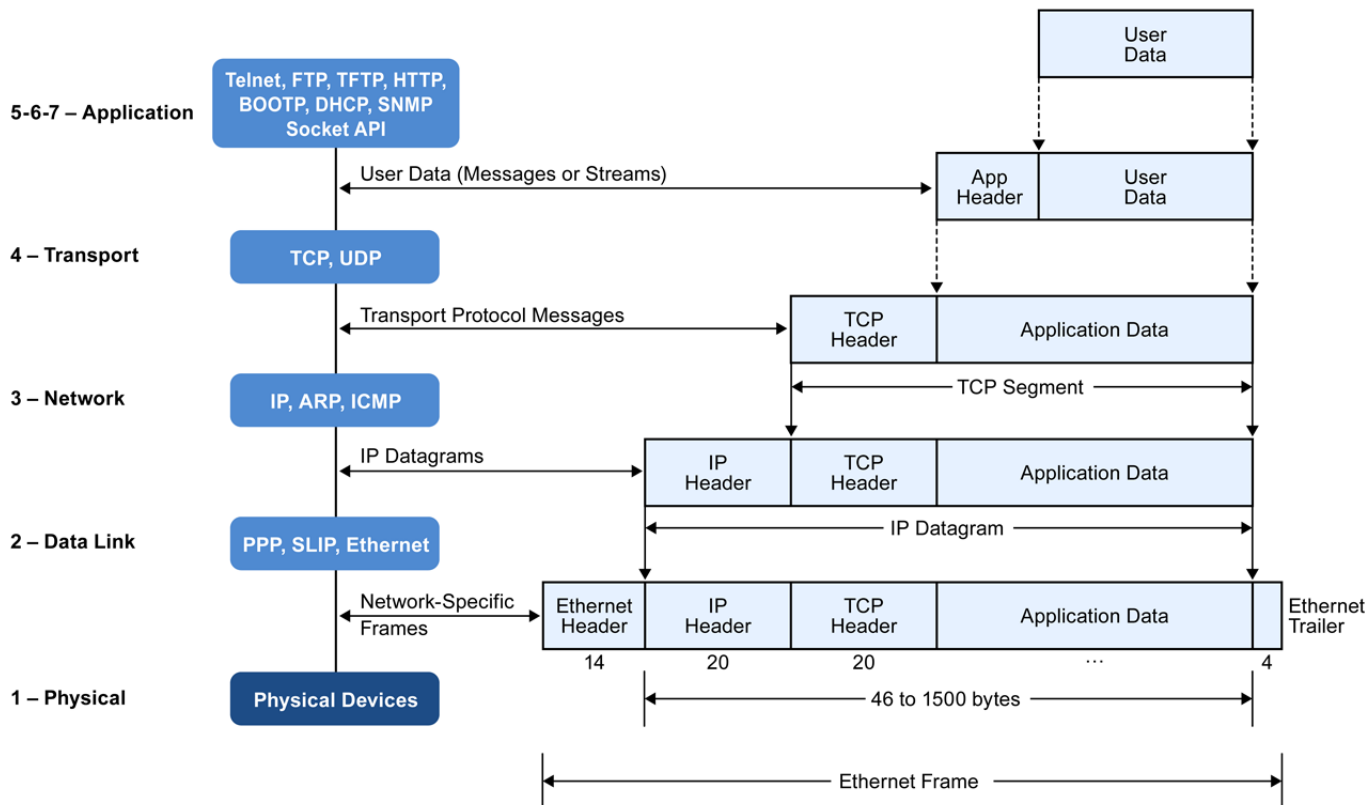
- Référence - [WebSockets](#) sur MDN web docs
- JDN - [Passez à l'échange de données en temps réel avec WebSockets](#) (JavaScript, PHP).
- Tutoriel "[Communication WebSocket avec un ESP8266 ou Arduino en Python](#)".

Lecture connexe

- [Utilisation de MQTT sur WebSockets avec Mosquitto](#)
- [Utilisation du client JavaScript MQTT avec des Websockets](#)

Définition

WebSocket est un standard du Web désignant un protocole réseau de la **couche application** et une interface de programmation du World Wide Web visant à créer des canaux de communication **full-duplex** par-dessus une connexion TCP pour les navigateurs web. [Wikipédia](#)



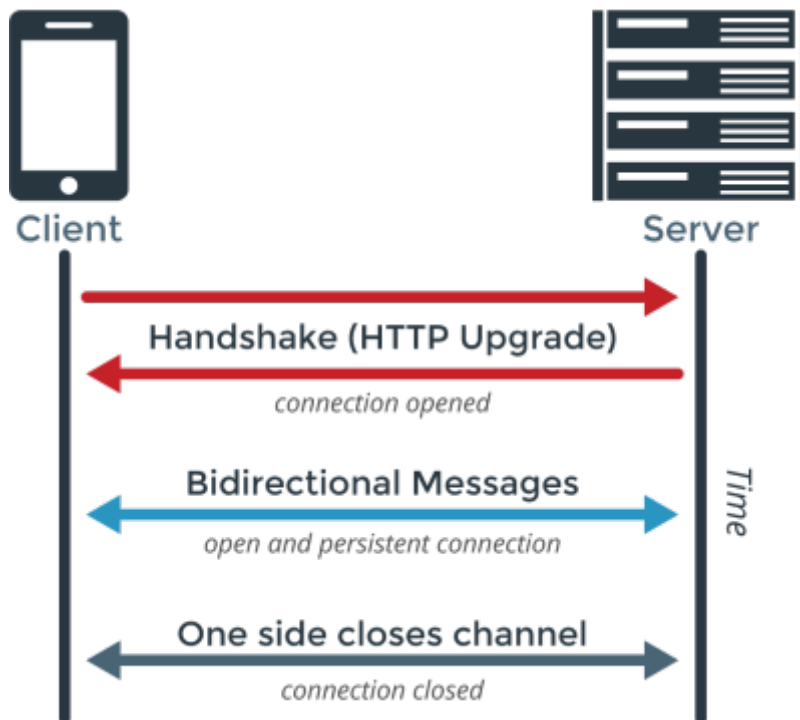
Avantages du WebSocket par rapport à l'API REST HTTP classique

Le protocole WebSocket a été élaboré pour les applications qui nécessitent des réponses rapides ou interactives. Le HTTP a été élaboré au début du Web (par le CERN de Genève). Le protocole HTTP est employé pour faire fonctionner les sites Internet mais également les applications mobiles (par exemple). Les API REST sont également basées sur l'HTTP. **L'HTTP n'est pas adapté aux applications qui nécessitent des réponses rapides ou interactives.** En effet, à chaque fois que le client fait une requête au serveur, on doit ouvrir une connexion, attendre la réponse du serveur puis refermer la connexion ce qui est consommateur de ressources et prend du temps de traitement.

On ne tient pas compte ici de la technologie Ajax qui permet d'actualiser le contenu d'une page web de manière asynchrone.

Le protocole **WebSocket vise à résoudre ces problèmes.** Il ouvre un tunnel de communication entre deux appareils. Ce tunnel reste ouvert jusqu'à ce que le client se déconnecte. À n'importe quel moment le client peut envoyer des messages (JSON, binaire, texte...) et vis versa.

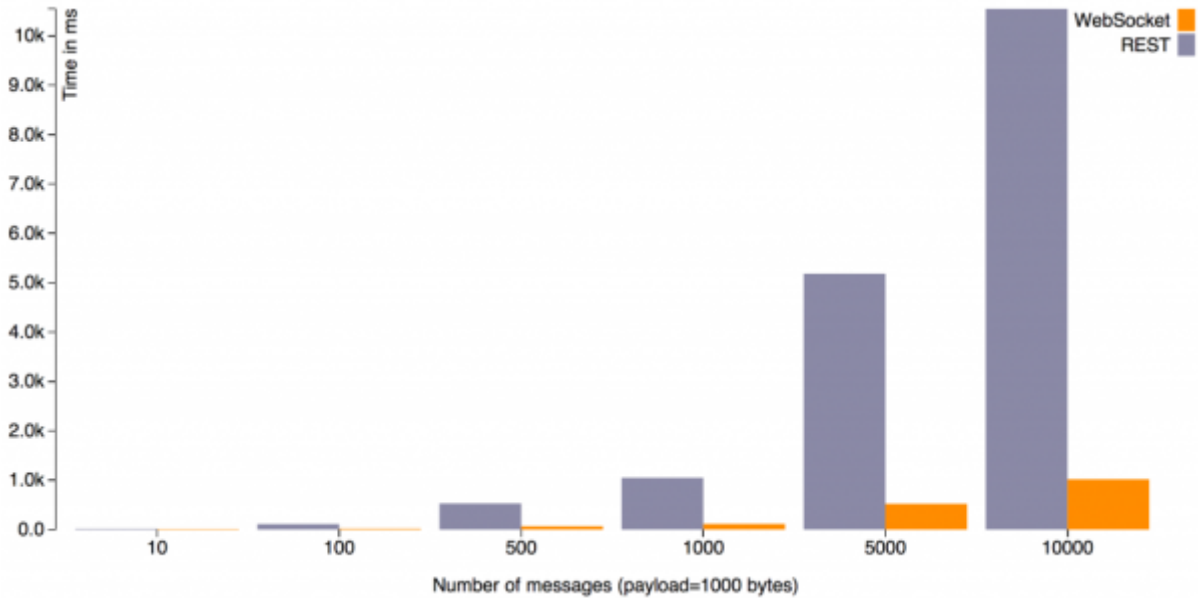
Pour établir une connexion WebSocket, une liaison HTTP spécifique est établie entre le client et le serveur. En cas de réussite, le protocole de la couche Application est « mis à niveau » de HTTP vers WebSockets, à l'aide de la connexion TCP établie. Une fois que cela s'est produit, le protocole HTTP disparaît du paysage. Les données peuvent être envoyées ou reçues aux deux points de terminaison via le protocole WebSocket, jusqu'à la fermeture de la connexion WebSocket.



Pour résumer, le WebSocket présente les avantages suivants :

- **Bidirectionnel** : le protocole HTTP est unidirectionnel, c'est-à-dire que le client envoie une requête à laquelle le serveur répond ensuite. Le client consomme ensuite la réponse et ainsi de suite. WebSocket est un protocole bidirectionnel dans lequel il n'y a pas de modèles de message prédéfinis tels que demande / réponse. Le client ou le serveur peut envoyer un message à l'autre partie.
- **Full-duplex** : serveur et client peuvent s'envoyer des messages à n'importe quel moment indépendamment des traitements en cours.
- **Connexion TCP unique** : Généralement, une nouvelle connexion TCP est lancée pour une requête HTTP et se termine après la réception de la réponse. Une nouvelle connexion TCP doit être établie pour une autre requête / réponse HTTP. Avec le WebSocket, le client et le serveur communiquent sur la même connexion TCP jusqu'à ce que le client ou le serveur ferme la connexion.
- **Léger** : le WebSocket se concentre sur l'essentiel contrairement à l'HTTP qui embarque de nombreuses informations à chaque question / réponse

En termes de performance, le WebSocket est beaucoup plus rapide comme le montre cette étude réalisée par le développeur Arun Gupta en 2014. Arun a mesuré le temps nécessaire pour envoyer des paquets de messages. Chaque message pèse 1000 bytes.



La différence peut sembler insignifiante (30% plus rapide) pour un nombre très faible de messages. Pour piloter un bras robotique ou un éclairage à LED depuis une application mobile, la différence est vraiment significative !

Constant payload, increasing number of messages			
Messages	REST (in ms)	WebSocket (in ms)	x times
10	17	13	1.31
100	112	20	5.60
500	529	68	7.78
1000	1050	115	9.13
5000	5183	522	9.93
10000	10547	1019	10.35

From: <https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link: <https://webge.fr/dokuwiki/doku.php?id=web:websockets&rev=1628666367>

Last update: **2021/08/11 09:19**

