Les protocoles applicatifs de l'IoT

Sommaire :

I - Présentation

Un **protocole applicatif** est un ensemble de règles définissant le mode de communication entre deux applications informatiques. Il se base sur les protocoles de transport (**TCP/UDP**) pour établir dans un premier temps des routes et échanger les données.

L'infrastructure du web classique n'est pas adaptée à la majorité des applications IoT. En effet, certains objets connectés, dits **contraints**, sont limités par de petits microcontrôleurs avec de petites quantités de mémoire (flash et RAM). Par conséquent, pour palier à ces contraintes il faut des protocoles moins verbeux avec un nombre limité de messages et de petites tailles.

On peut identifier trois familles de protocoles applicatifs :

- **Protocoles de messagerie** : **MQTT**, **XMPP** et **AMQ** ;
- **Protocoles de transfert web** : **CoAP**, **API ReST** ;
- **Protocole réseau** : **Websocket**.

Les protocoles applicatifs les plus adaptés aux objets connectés contraints (limités en ressources d'énergie et de calcul) sont **MQTT** et **CoAP**.

II - Les protocoles de messagerie

Les **protocoles de messagerie** s'appuient sur un mécanisme de publication et d'abonnement, où les transferts de données se font de manière asynchrone. On trouve principalement les protocoles :

- **MQTT** (Message Queuing Telemetry Transport) : Standard depuis 2015 ;
- **XMPP** (eXtensible Messaging and Presence Protocol) ;
- **AMQP** (Advanced Message Queuing Protocol).

III - Les protocoles de transfert Web

Les **protocoles de transfert Web** s'appuient sur un mécanisme basé sur une architecture client/serveur et sur le protocole **HTTP**. On trouve principalement les protocoles :

- **CoAP** (Constrained Application Protocol) ;
- **ReST** (Representational State Transfer).

IV - Le protocole Websocket

Le **protocole Websocket** permet l'établissement d'un canal de communication full-duplex en une seule connexion **TCP** entre un client et un serveur.

- la phase de connexion appelée «Handshake» initié par le client ;
- la phase d'échange bidirectionnel de messages ;
- la phase de clôture du canal initiée par l'une des deux parties.

V - Résumé des caractéristiques des principaux protocoles applicatifs

Protocole	MQTT	XMPP	AMQP	CoAP	WebSocket	API REST
Type de protocole	Messagerie	Messagerie	Messagerie	Transfert web	Réseau	Transfert web (HTTP)
Modèle de communication	Publish/Subscribe	Publish/Subscribe Request/Response	Producer/Consumer	Request/Response	bidirectionel	Request/Response
Transport	TCP/IP	TCP/IP	TCP/IP	UDP/IPv6 /6LowPAN	TCP/IP	TCP/IP
Sécurité	TLS/SSL	TLS/SSL	TLS/SSL	DTLS	TLS/SSL	TLS/SSL
Format	Binaire, Text (json, xml, csv)	XML	Binaire, Text	Binaire, Text	Binaire, Text	Binaire, Text
Contraintes sur les objets connectées	Fortes	Faibles/Moyennes	Moyennes	Fortes	Faibles	Faibles
Principaux framework	Emqtt, HiveMQ, Mosquitto, Eclipse Paho	Jabber, XMPPFramework	RabbitMQ, StormMQ	Eclipse Californium, nCoAP	jetty websocket, Apache Tomcat	Django REST, Apache Tomcat, Node.js, Ruby on Rails