

Critères	MQTT	HTTP
Protocole	Publish/Subscribe (pub/sub)	Requêtes client-serveur (REST)
Cas d'utilisation idéal	Transmission de données de capteurs IoT régulières et légères	Envoi de données ponctuelles, idéal pour requêtes simples
Simplicité de mise en place	Relativement simple, nécessite un broker (Mosquitto)	Très simple, nécessite seulement un serveur HTTP
Transmission des données	Asynchrone, les données sont publiées sur un broker MQTT	Synchrone, les données sont envoyées via des requêtes POST
Latence	Faible, conçu pour la faible bande passante	Moyenne, dépend de la requête et de la réponse
Fréquence de transmission	Adapté pour des transmissions fréquentes et continues	Adapté pour des envois ponctuels ou moins fréquents
Consommation de bande passante	Très faible (léger), conçu pour minimiser l'utilisation de la bande passante	Moyenne, chaque requête nécessite un en-tête complet
Sécurité	Supporte SSL/TLS pour sécuriser les messages	Supporte HTTPS pour sécuriser les transmissions
Fiabilité	Fiabilité configurable (QoS 0, 1 ou 2)	Bonne, mais chaque requête doit être gérée indépendamment
Cas d'utilisation dans mon projet	Collecte et envoi réguliers de données de capteurs	Envoi de résultats ponctuels (par exemple, à la fin d'un cycle de mesure)
Outils requis	Broker MQTT (Mosquitto), client MQTT (Paho)	Serveur HTTP (Flask, Django, etc.), client HTTP (requests en Python)

Conclusion : pour envoyer des fichiers importants

	<u>HTTP</u>	<u>MQTT</u>
Transfert de gros fichiers	Peut envoyer des fichiers de plusieurs mégaoctets en une seule requête sans fragmentation spécifique	Nécessite de fragmenter les fichiers en plusieurs petits messages
Gestion de la Fragmentation (si découpage de fichier (photo, vidéo))	<u>Pas besoin</u>	Fragmentation manuelle requise au niveau de l'application. Chaque fragment doit être envoyé et confirmé individuellement
Réassemblage des Paquets	<u>Automatique</u>	Risque de perte d'ordre ou de paquets
Temps de Transfert	Pas de fragmentation donc rapide même pour les fichiers volumineux	Plus lent en raison de l'envoi, de la confirmation et du ré assemblage de chaque fragment
Simplicité d'Implémentation	Facile à implémenter : une seule requête POST pour envoyer un fichier.	Complexe à implémenter : nécessite un système de gestion des fragments et de ré

		assemblage.
<u>Conclusion</u>	<u>Adapté</u> pour les transferts de gros fichiers (images, vidéos) sur une connexion LTE	Adapté uniquement pour de petites transmissions de données ou de petits fichiers par exemple uniquement pour les capteurs