



Université Abdelhamid Mehri – Constantine 2
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique Fondamentale et ses Applications

1

Artificial Intelligence of Things (AIoT)

2^{EME} ANNÉE MASTER
SDIA
S1

DR ILHAM KITOUNI

24-25

Chapitre 2- IoT Foundations

[AloT-syllabus-ang24-25.docx](#)

References

- González García, C., Núñez Valdéz, E. R., García Díaz, V., Pelayo García-Bustelo, B.C., & Cueva Lovelle, J. M. (2019). A review of artificial intelligence in the internet of things. International Journal Of Interactive Multimedia And Artificial Intelligence, 5.
- Ghosh, A., Chakraborty, D., & Law, A. (2018). Artificial intelligence in Internet of things. CAAI Transactions on Intelligence Technology, 3(4), 208-218. Andrew S. Tanenbaum, Maarten van Steen. Distributed Systems: Principles and Paradigms. 3rd edition. Pearson Education, 2017.

1. Principes de l'IoT : Capteurs, Réseaux, et Dispositifs Connectés

3

Capteurs IoT : Les capteurs sont au cœur de l'IoT. Ils permettent de mesurer divers paramètres comme la température, l'humidité, la luminosité, le mouvement, etc. Les capteurs collectent des données du monde réel qui sont ensuite transmises via des réseaux.

- **Exemples** : Les capteurs de qualité de l'air utilisés dans les villes intelligentes pour surveiller la pollution atmosphérique, ou les capteurs de mouvement dans les maisons intelligentes qui détectent les intrus.
- Le projet **Nest** de Google

1. Principes de l'IoT : Capteurs, Réseaux, et Dispositifs Connectés

4

- **Réseaux dans l'IoT** : Les capteurs IoT sont connectés à un réseau, généralement sans fil. Les réseaux utilisés dans l'IoT incluent :
 - **LPWAN** (Low Power Wide Area Network) : (LoRaWAN : Long Range WAN)
 - **Wi-Fi et Bluetooth** : Pour des dispositifs à courte portée.
 - **5G** : Besoin d'une faible latence et d'une haute bande passante (Voitures autonomes).
 - **Exemple** : La technologie **LoRaWAN** est utilisée dans les **villes intelligentes**.

1. Principes de l'IoT : Capteurs, Réseaux, et Dispositifs Connectés

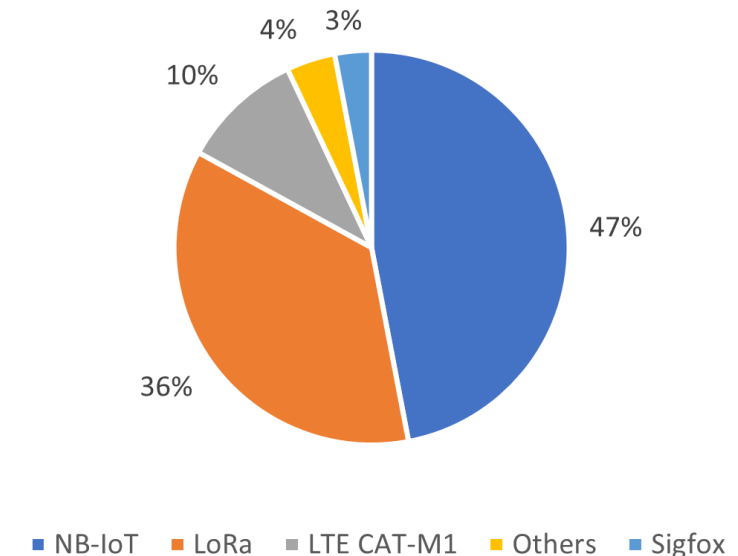
Comparaison des technologies LPWAN :

5

Les principaux facteurs de décision: la bande passante disponible, la vitesse de transmission/débit, la portée, la consommation d'énergie et le coût.

NB-IoT (Narrowband IoT) : NB-IoT est une technologie cellulaire optimisée pour les applications IoT à faible consommation d'énergie et à large portée.
LTE CAT-M1 : LTE CAT-M1 est une technologie cellulaire qui fonctionne au sein des réseaux LTE existants, offrant une couverture améliorée et une meilleure pénétration par rapport aux réseaux cellulaires traditionnels.

	Cellular		Non-Cellular	
	NB-IoT	LTE-CAT M1	Sigfox	LoRaWAN
Spectrum	Licensed		Unlicensed	
Bandwidth	200kHz	1.4MHz	0.1kHz	125-500kHz
Throughput	200kbps	1mbps	0.1kbps	50kbps
Range	Up to 10km	Up to 10km	Up to 50km	Up to 20km
Power Consumption	Low-to-Moderate	Moderate	Extremely Low	Ultra-Low

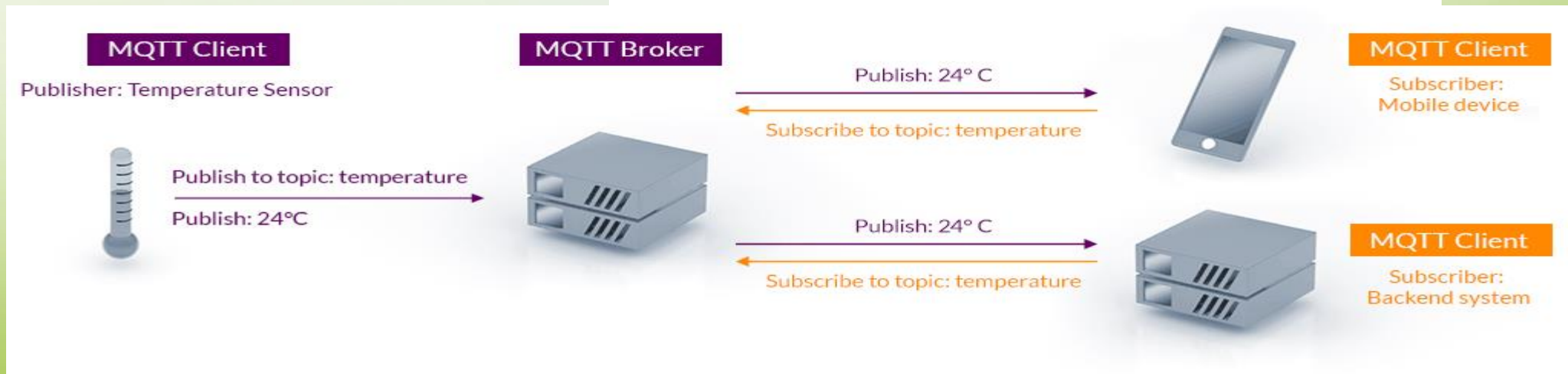


2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

6

- **MQTT** (Message Queuing Telemetry Transport) : C'est un protocole de communication léger utilisé pour les appareils à faible bande passante. Il est idéal pour les systèmes qui ont besoin de transmettre des données par petites quantités, en temps réel.

MQTT Publish / Subscribe Architecture



- **Référence** : OASIS MQTT Version 5.0 fournit une spécification de ce protocole, largement utilisé dans les dispositifs IoT pour le contrôle à distance et la collecte de données.

2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

CoAP (Constrained Application Protocol) : Conçu pour les appareils contraints en termes de ressources, comme ceux avec peu de puissance ou de mémoire, il est basé sur UDP et est particulièrement utilisé dans les applications industrielles. [RFC 7252](#)

Constrained Application Protocol,

Caractéristique	Description
Modèle	Basé sur le modèle REST, comme HTTP (méthodes GET, PUT, POST, DELETE)
Similitude avec HTTP	Très similaire à HTTP, facilitant l'intégration et le transfert de compétences pour les développeurs
Protocole de transport	Utilise UDP sur IP avec des en-têtes compacts et un codage efficace, minimisant la consommation de ressources
Appareils ciblés	Conçu pour des microcontrôleurs et appareils avec peu de mémoire, prenant en charge des milliards de nœuds dans l'IoT
Flexibilité des formats	Supporte différents formats de données : XML, JSON, CBOR
Découverte des ressources	Intégré avec des fonctionnalités de découverte de ressources (URL)
Fonctionnement stateless	Peut fonctionner de manière stateless (sans état) sur de nombreux serveurs
Standardisation	Développé comme un standard internet (RFC 7252)
Problématiques complexes	Aborde des problèmes comme le contrôle de congestion (renvoi aléatoire)
Sécurité	Utilise DTLS avec des paramètres robustes (équivalent à des clés RSA 3072 bits), offrant une sécurité solide même sur des dispositifs limités en ressources

2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

8

Format	Structure	Usage principal	Taille	Lisibilité humaine	Flexibilité
XML	Hiérarchique	Documents structurés, services web (SOAP)	Grand	Oui	Très flexible
JSON	Paires clé-valeur	API REST, échanges de données web	Moyenne	Oui	Flexible
CBOR(Concise Binary Object Representation)	Binaire (compact)	IoT, systèmes à faibles ressources	Très petite	Non	Flexible

- **Exemple réel : Smart agriculture** utilisant CoAP pour envoyer des données environnementales des capteurs de sol à un serveur central.

2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

9

- **HTTP** (Hypertext Transfer Protocol) : Le protocole classique du web est également utilisé pour les applications IoT, bien que son utilisation soit plus lourde par rapport à MQTT et CoAP. Il est principalement utilisé pour des communications qui ne nécessitent pas une latence faible.
- **Exemple : Amazon Alexa** utilise HTTP pour échanger des informations vocales avec les serveurs cloud.

2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

Caractéristique	MQTT (Message Queuing Telemetry Transport)	CoAP (Constrained Application Protocol)	HTTP (Hypertext Transfer Protocol)
Type de protocole	Protocole de messagerie publié/souscrit (Publish-Subscribe)	Protocole basé sur REST	Protocole basé sur REST
Modèle de communication	Asynchrone (pub-sub avec broker)	Synchrone (client-serveur)	Synchrone (client-serveur)
Protocole de transport	TCP/IP (connexion permanente)	UDP/IP (connexion légère, sans état)	TCP/IP
Léger/Conçu pour l'IoT	Oui, conçu pour des environnements à faible bande passante et des ressources limitées	Oui, spécialement conçu pour les appareils contraints	Non, plutôt conçu pour des systèmes puissants
Modes de message	Publication, abonnement, notifications	GET, POST, PUT, DELETE	GET, POST, PUT, DELETE
Consommation d'énergie	Faible (connexion persistante, messages légers)	Très faible (UDP, messages compacts)	Plus élevée (due à la surcharge de TCP)
Complexité du protocole	Simple	Simple à modérer	Relativement complexe
Support QoS (Quality of Service)	Oui (QoS 0, 1, 2)	Non, mais utilise des messages confirmables	Non

2. Protocole de Communication dans l'IoT (MQTT, CoAP, HTTP)

Caractéristique	MQTT (Message Queuing Telemetry Transport)	CoAP (Constrained Application Protocol)	HTTP (Hypertext Transfer Protocol)
Fiabilité	Fiabilité assurée par TCP et QoS	Pas de garantie de livraison (UDP), implémentable avec des messages confirmables	Fiabilité assurée par TCP
Découverte des ressources	Non	Oui (grâce au CoAP Resource Directory)	Non
Sécurité	TLS pour la sécurité du transport	DTLS (basé sur UDP)	TLS/SSL (sécurité basée sur TCP)
Cas d'utilisation typiques	Surveillance à distance, capteurs IoT, envoi de données en continu	IoT, capteurs, actionneurs, communications M2M	Services web, applications web
Scalabilité	Très scalable (pub/sub permet une gestion massive de clients)	Scalabilité élevée dans les réseaux contraints	Scalabilité modérée (connexion client-serveur lourde)
Latence	Très faible	Très faible	Plus élevée, à cause de la surcharge TCP
Taille des messages	Petite (quelques octets)	Très petite (4-byte header, options compactes)	Messages relativement volumineux

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

12

Introduction aux plateformes IoT

Définition : Infrastructure connectant les dispositifs IoT pour la collecte et l'analyse des données.

Rôle clé : Facilite la communication entre les objets physiques et le cloud/applications.

Objectifs :

- Surveillance des dispositifs.
- Optimisation des opérations.
- Analyse des données pour prise de décision.

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

13

Rôle et importance des plateformes IoT

- 1. Connectivité** : Communication entre appareils via divers protocoles (MQTT, HTTP, etc.).
- 2. Gestion des dispositifs** : Configuration, supervision et mise à jour d'appareils.
- 3. Sécurité** : Chiffrement, authentification, gestion des identités des appareils.
- 4. Analyse des données** : Transformation des données en informations exploitables.
- 5. Intégration** : Connexion avec d'autres systèmes (ERP, CRM, etc.).

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

14

Type, avantages et limites des plateformes

Selon leur mode de déploiement et leurs fonctionnalités spécifiques

Cloud-based :

- Scalabilité, coûts réduits.
- Dépendance à l'Internet, risques de confidentialité.

On-premise :

- Contrôle total, sécurité accrue.
- Coût initial élevé, maintenance complexe.

Hybrid :

- Flexibilité, optimisation des coûts.
- Complexité de gestion.

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

15

Exemples de plateformes populaires

AWS IoT : Amazon Web Services (AWS)

- Gestion à grande échelle, sécurité avancée, intégration avec les services AWS.

Google Cloud IoT :

- Analyse en temps réel, machine learning avec TensorFlow.

Microsoft Azure IoT :

- Edge computing, conformité sécurité.

ThingSpeak :

- Open source, visualisation simple, API REST.

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

16

Caractéristiques	AWS IoT	Google Cloud IoT	Microsoft Azure IoT	ThingSpeak
Avantages	Gestion d'appareils à grande échelle, outils avancés d'analyse	Fortes capacités d'analyse, machine learning intégré	Bonne gestion des appareils, intégration facile avec services Microsoft	Simplicité, open-source, idéal pour prototypes
Limites	Coûts élevés pour les grandes applications	Dépendance au cloud, coûts potentiels élevés pour le traitement des données	Complexité d'intégration pour les petits projets	Fonctionnalités limitées pour les grandes échelles

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

Caractéristiques	AWS IoT	Google Cloud IoT	Microsoft Azure IoT	ThingSpeak
Type	Cloud-based	Cloud-based	Cloud-based avec support Edge	Open-source
Scalabilité	Très élevée, gestion de millions d'appareils	Très élevée, intégration avec BigQuery pour l'analyse des grandes données	Très élevée, gestion des appareils et prise en charge de millions d'unités	Limité, convient aux petits projets
Edge Computing	Oui, avec AWS Greengrass	Oui, avec IoT Edge	Oui, avec Azure IoT Edge	Non
Analyse des données	Intégration avec AWS services comme Lambda, Kinesis, SageMaker	Google Cloud Dataflow, BigQuery, AutoML, TensorFlow	Intégration avec Power BI, Stream Analytics, AI et ML via Azure Machine Learning	Simple visualisation des données, pas d'analyse complexe
Sécurité	Très avancée (IAM, certificats, chiffrement)	Sécurité robuste via Google Security et IAM	Sécurité avancée avec Azure Security Center, conformité réglementaire (HIPAA, GDPR)	Basique, pas de chiffrement par défaut

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

18

Caractéristiques	AWS IoT	Google Cloud IoT	Microsoft Azure IoT	ThingSpeak
Gestion des appareils	AWS IoT Device Management, gestion à grande échelle	Google IoT Core, gestion efficace des appareils	Azure IoT Hub, gestion centralisée des appareils	Limité, principalement pour petits projets éducatifs
Machine Learning	SageMaker pour la modélisation ML	Intégration native avec TensorFlow et AutoML	Azure ML, intégration avec d'autres outils AI	Non intégré
Intégration avec services tiers	Large, avec tous les services AWS	Forte, avec les services Google et API REST	Forte, avec les services Microsoft et autres API	API REST simple
Flexibilité de déploiement	Cloud et hybride	Cloud principalement	Cloud et Edge computing	On-premise (serveurs personnels)

3. Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

19

Comparaison des fonctionnalités ML dans les plateformes IoT

Plateforme	Outils ML intégrés	ML en périphérie (Edge)	Cas d'usage courants
AWS IoT	Amazon SageMaker, AWS IoT Analytics	Oui, AWS IoT Greengrass	Maintenance prédictive, détection d'anomalies
Google Cloud IoT	TensorFlow, AutoML, Google Cloud AI	Oui, avec Edge TPU	Analyse en temps réel, optimisation, reconnaissance d'image
Microsoft Azure IoT	Azure Machine Learning, Cognitive Services	Oui, Azure IoT Edge	Optimisation industrielle, détection d'anomalies, ML temps réel
ThingSpeak	MATLAB Analytics	Non	Détection de tendances basiques, prototypes

3. Conclusion Les Plateformes IoT (AWS IoT, Google IoT Core, etc.)

20

Plateformes IoT : Pilier central de l'écosystème IoT.

Importance : Connectivité, gestion des appareils, sécurité, analyse des données.

Tendance : Scalabilité, intégration AI/ML, sécurité renforcée.

Défis Techniques : Sécurité, Interopérabilité, Gestion des Données