Optimisation de l'architecture LoRaWAN pour la Mobilité des Appareils IoT

Arnol Lemogue, Ivan Marino Martinez Bolivar, Laurent Toutain, Ahmed Bouabdallah

Optimisation de l'architecture LoRaWAN pour la Mobilité des Appareils IoT

Arnol Lemogue[1] et Ivan Martinez[1] et Laurent Toutain[1] et Ahmed Bouabdallah[1]

[1]*IRISA, UMR CNRS 6074, F-35700, IMT Atlantique, Rennes, France.*

LoRaWAN est une technologie de réseau ouvert conçue pour des équipements IoT qui permet une transmission de données sans fil sur une longue distance contrairement à d'autres technologies sans fil, comme Wi-Fi ou Bluetooth. Cependant, les appareils IoT qui utilisent ce réseau lors de leur mobilité doivent passer par des accords de roaming qui restent très centralisées aujourd'hui. Afin d'apporter des solutions plus flexibles à ce problème, nous proposons dans cet article, une nouvelle architecture sécurisée et facilement utilisable basée sur la technologie DNS over HTTPS (DoH). Des expériences réelles ont démontré que cette nouvelle architecture est à la fois évolutive et sécurisée pour la mobilité des appareils IoT entre les réseaux LoRaWAN.

Mots-clefs : IoT, LoRaWAN, Mobilité, DoH, PKI

1 Introduction

La notion de mobilité des appareils encore appelée "roaming" a fait son apparition pour la première fois dans le domaine des télécommunications avec la montée en popularité des téléphones mobiles, ce qui a incité les opérateurs de réseau à étendre leurs services pour la prendre en compte. Selon l'association GSM[BR04], il s'agit de la "capacité pour les clients d'utiliser leur téléphone mobile ou autre appareil mobile en dehors de leur zone de couverture". En d'autres termes, le roaming se produit lorsqu'un appareil est connecté à un réseau sans fil et sort de la zone de couverture de ce réseau pour entrer dans une autre.

Dans la termologie LoRaWAN, le réseau d'origine est appelé le "*home/forwarding Network Server (hNS/fNS)*" et le réseau visité est appelé le *"visited/serving Network Server(vNS/sNS)"*. Le passage d'un appareil d'un réseau à l'autre est soumis à des accords de roaming entre les opérateurs de réseau. Ces accords définissent les conditions du roaming, telles que les services disponibles pour l'appareil en mode roaming, les frais et autres détails pratiques. Si aucun accord de roaming n'existe entre les opérateurs du réseau d'origine et du réseau visité, le roaming ne peut pas être effectué avec l'appareil en question. Dans cet article, nous proposons une extension de l'architecture de roaming actuelle proposée par la LoRa Alliance.

Notre approche repose sur l'introduction d'une nouvelle entité appelée *"DNS Broker"* améliore l'évolutivité et la sécurité de l'architecture de roaming en supprimant la nécessité pour un propriétaire d'appareils IoT(PA) de gérer des identifiant complexes, ce qui permet de réduire le nombre d'échanges nécessaire à la localisation d'un hNS en service.

Le reste de cet article est structuré comme suit : Section 2 donne un aperçu de quelques travaux sur des architectures de Roaming LoRaWAN existantes. Section 3 décrit notre architecture proposée et son implémentation. Section 4 analyse les performances de notre solution et la compare avec IoTRoam. Enfin, Section 5 conclut l'article.

2 Etat de l'art et évolution de LoRaWAN dans le contexte de Roaming

Dans[BBMA21], les auteurs proposent une architecture de roaming LoRaWAN appelée "IoTRoam" qui permet aux PA de connecter leurs appareils à d'autres sNS à l'échelle mondiale de manière transparente en utilisant les technologies, telles que le PKI (Public Infrastructure Key) et DNS. Bien que leur approche présentée à la figure 1 est basée sur le DNS, elle repose sur l'utilisation du JoinEUI. Cet identifiant qui

permet aux appareils IoT finaux de joindre le Join Server (JS) pendant le processus de roaming presente des problèmes suivants : i) Contrairement au `DevEUI` qui est statique, attribué lors de la fabrication de l'appareil et qui reste le même tout au long de son cycle de vie, le `JoinEUI` doit être modifié lorsqu'un appareil est acheté ou vendu à un autre client. iii) La zone et les enregistrement DNS sont gérés par la LoRa Alliance pour assurer la correspondance entre `JoinEUI` et JS ou `NetID` et NS. Cela implique une procédure d'authentification. iv) Si le JS est interne au hNS, seul le NS y accédera pour authentifier les appareils en utilisant le `NetID` du hNS ([étape 5-6] de la figure 1). Cela peut poser des problèmes de sécurité, car la JS peut ne pas être en mesure de prévoir quel fNS enverra une requête et exposera un élément sensible à l'internet.

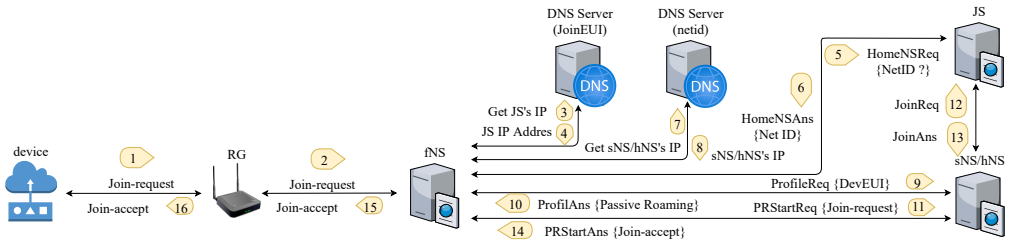

FIGURE 1 – Procédure de roaming dans le réseau LoRaWAN.

3 Architecture de Roaming LoRaWAN proposée

Nous proposons dans cette section, une extension de l'architecture IoTRoam en introduisant une nouvelle entité appelée DNS Broker(détails dans la section 3.1)
La figure 3 donne le principe de roaming avec notre approche. Elle fonctionne de la manière suivante :

i) L'appareil IoT envoie un message de demande d'adhésion (JR). La passerelle LoRa transmet le message JR à son fNS [étapes 1-2]. ii) Le fNS détermine s'il existe un accord d'itinérance avec hNs qu'il souhaite joindre en effectuant une requête DNS comportant le `DevEUI` afin d'obtenir le `NetID` et l'adresse IP du hNS/sNS. [étapes $3' - 4'$ et $7' - 8'$]. Les étapes 5-6 de la figure 1 qui ne sont plus nécessaires, elles sont supprimées. iii) Le fNS envoie un message `ProfileReq` au hNS/sNS portant le `DevEUI` [étape 9]. iv) Le hNS/sNS envoie un message `ProfileAns` indiquant le type de profil d'itinérance. v) Le fNS envoie un message `PRStartReq` transportant le message `Join-request`. Ensuite, le hNS/sNS transmet le message `JoinReq` à son JS [étapes 11-12]. vi) Le JS répond par un message `JoinAns` contenant le message de Join-accept [étape 13]. vii) Le hNS/sNS envoie un message `PRStartAns` contenant le `Join-accept`. Enfin, le fNS envoie un message `Join-accept` à l'appareil [étapes 14-16].

3.1 Intégration du DNS-Broker

Comme le montre la figure 3, notre architecture de roaming introduit un nouvel élément de réseau appelé DNS Broker. Par conséquent, chaque fois qu'un message de demande d'adhésion(JoinRequest) à un réseau LoRaWAN envoyé par un appareil IoT portant un DevEUI inconnu arrive au fNS, le fNS effectue une requête DNS contenant le DevEUI pour obtenir le NetID correspondant et l'adresse IP du hNS. Ces deux valeurs sont utilisées ultérieurement pour établir une connexion IP entre les deux NS.
Cette résolution DNS est divisée en deux parties : une partie publique pour obtenir l'IP d'un DNS Broker et une partie privée où le DevEUI est résolu en un NetID/une adresse IP.Cette partie privée est gérée par le Broker.
La Figure 2 représente schématiquement l'architecture globale avec le Broker. Le PA ajoute dans une base de données le DevEUI des appareils qui doivent être pris en compte. Cette base de données est synchronisée avec celle du Broker. Lorsqu'un nouveau appareil apparaît chez un hNS et envoie un message Join-request, au lieu d'utiliser le JoinEUI pour localiser son sNS, le fNS contacte le Broker pour obtenir l'adresse du sNS et poursuit la procédure d'adhésion.

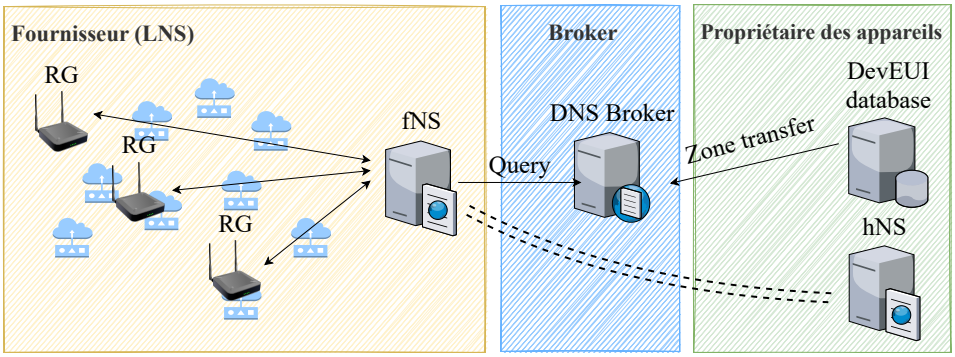

FIGURE 2 – Architecture DNS-Broker

3.2 *Résolution DNS privée utilisant DoH et certificats d'authentification*

En utisant DoH, notre approche fonctionne de la manière suivante : le nom de domaine est connu par le demandeur pour être divisé en différents éléments. Par exemple, pour contacter le sNS, le nom de domaine est composé d'une partie publique et d'une partie privée (soulignée). On a donc : `DevEUI{EUI.64}.deveui.iot-roam.net`, et la résolution se fait comme suit : i) Partie publique : `deveui.iot-roam.net` est résolu en utilisant le serveur DNS local en amont, la réponse DNS correspond à l'IP du Broker. ii) Partie privée : `DevEUI{EUI.64}` est résolu en utilisant le DNS Broker comme serveur en amont. Un certificat côté client est nécessaire pour obtenir une réponse DNS correcte. Les certificats clients sont délivrés par le Broker, qui fait également office de CA. En ce qui concerne la connexion entre NS, contrairement à IoTRoam, nous proposons qu'elle se fasse selon les politiques de hNS/sNS sans que les autorités centrales n'émettent de certificats.

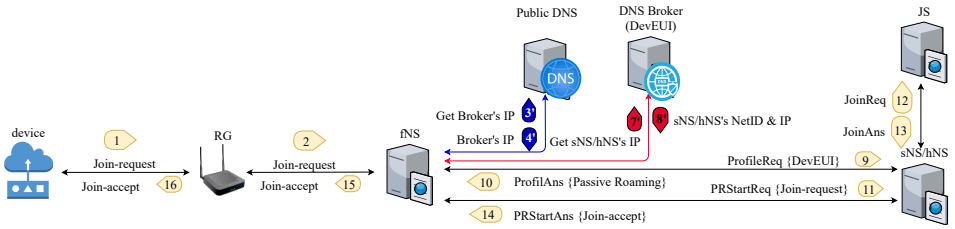

FIGURE 3 – Procédure d'adhésion dans notre architecture d'itinérance proposée

L'architecture d'itinérance a été testée et mise en œuvre comme suit † : 1)hNS/fNS est une version modifiée de Chirpstack avec un client DoH dnsproxy supportant l'authentification du client ; 2) **DNS Broker** est un serveur avec le domaine : `broker.iot-roam.net`.

4 Évaluation des Performances

Nous avons évalué la performance de notre architecture en analysant le trafic dans le fNS lorsqu'un appareil IoT connecté à une passerelle LoRa, qui est relié à la plateforme Chirpstack, est en situation de roaming. Nous avons déterminé le volume de trafic IP, DNS et UDP pour obtenir l'adresse IP des différents serveurs et mesuré le volume de trafic IP, TCP, TLS et HTTPS pour les échanges entre les serveurs.
Les résultats présentés sur la figure 4 compare le trafic généré par IoTRoam à celui généré par notre proposition dans les trois catégories suivantes : (i) DNS, correspondant au trafic nécessaire pour obtenir l'adresse IP des différents serveurs (JS, NS, Broker), (ii) HTTPS, correspondant au trafic généré par l'échange de messages entre les serveurs, et (iii) la granularité TLS, qui correspond au volume de trafic nécessaire pour

†. Le code source, ainsi qu'un tutoriel sur la mise en œuvre de notre plate-forme d'itinérance, sont disponibles ici : `https://github.com/MarinMtz/LoRaWAN-Roaming-tutorial`.

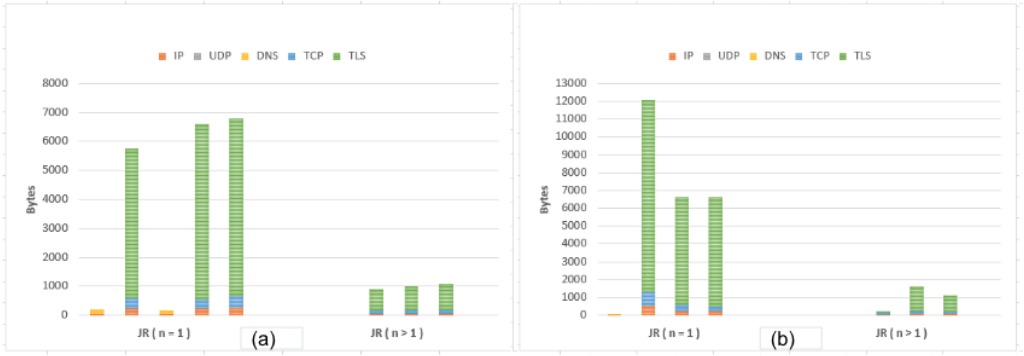

FIGURE 4 – Flux de l'architecture existante(a) et de celle proposée(b)

la poignée de main et les données.

Lors de cette évaluation, nous avons effectué plusieurs demandes de connexion (JR). (i) La première demande (n=1) se produit la première fois qu'un appareil essaie de se connecter à un réseau. Les demandes ultérieures (n>1) proviennent du même appareil. Nous avons observé que le nombre d'échanges de DNS a diminué passant de 8 à 4, mais le trafic HTTPS a augmenté, passant de 15 à 24 paquets en raison de l'introduction de certificats pour l'authentification mutuelle avec le Broker-DNS. Cela se reflète dans la charge totale de trafic TLS qui a largement augmenté. (ii) Pour les JRs (n > 1), aucun changement significatif n'a été constaté entre les deux architectures. Le nombre d'échanges DNS reste identique pour les deux (0). En ce qui concerne HTTPS, la charge TLS est presque identique et le nombre de paquets TLS est égal à 6.

5 Conclusions et perspectives

Nous avons développé une architecture de roaming souple, évolutive et sécurisée pour faciliter la mobilité des appareils IoT dans les réseaux LoRaWAN nécessitant peu de modifications de l'architecture LoRaWAN existante. Le processus de localisation du sNS peut être simplifié en utilisant un nouvel élément appelé DNS Broker, en plus de DNS, permettant une résolution privée basée sur DoH. ce qui représenté la clé de la scalabilité et de la sécurité apportée par notre approche. Cela confirme également le nouveau rôle que DoH pourrait jouer pour tirer parti de sa robustesse et fiabilité. Le DNS Broker mérite cependant une étude dédiée à l'avenir. Les mesures de performance sont légèrement pénalisées en raison de l'utilisation de HTTPS. Cependant, d'autres approches plus légères telles que DNS over CoAP(DoC)[LAG+21] seront considérées comme une tentative pertinente pour alléger la charge du protocole tout en conservant la même fonctionnalité.

6 Remerciements

Ce travail a été partiellement financé par le projet DiNS de l'ANR.

Références

[BBMA21] Sandoche Balakrichenan, Antoine Bernard, Michel Marot, and Benoit Ampeau. IoTRoam : design and implementation of a federated IoT roaming infrastructure using LoRaWAN. 2021.

[BR04] Harishankar M.V. Borgaonkar Ravishankar. Roaming Issues in 3GPP Security Architecture and Solution Using UMM Architecture. IEEE Wireless Communications, 2004.

[Hof18] P. Hoffman. RFC8484 : DNS Queries over HTTPS (DoH). October 2018.

[LAG+21] M.S. Lenders, C. Amsüss, C. Gündoğan, T.C. Schmidt, and M. Wählisch. DNS queries over CoAP (DoC) - draft-lenders-dns-over-coap-02. *IETF*, oct. 2021.

[lor17] LoRaWAN 1.1 specification. technical standard. Technical report, LoRa Alliance, 2017.