Lab 7

Mise en ouvre d'un réseau de type LoRaWAN simplifié - LoRaTS

7.1. Lien LoRa simple et la couche d'accès

Un réseau étendu basé (Long Range) sur le protocole **LoRaWAN** permet la communication à bas débit, par radio, d'objets communiquant selon la technologie **LoRa** et connectés à l'Internet via des pesserelles, passerelles, participant ainsi à l'Internet des objets (IoT) , les réseaux machine-à-machine M2M, etc. ..

La technologie de modulation liée à LoRaWAN est LoRa.

Le protocole LoRaWAN sur la couche physique LoRa permet de connecter des capteurs ou des objets nécessitant une longue autonomie de batterie (compter en années), dans un volume (taille d'une boite d'allumettes ou d'un paquet de cigarette) et un coût réduits.
LoRaWAN est l'acronyme de *Long Range Wide-area network* que l'on peut traduire par «réseau étendu à longue portée».

La portée d'une communication LoRa est déterminée par sa bande passante, la puissance de sortie du signal ainsi que par le facteur d'étalement utilisé – Spreading Factor (SF).

L'étalement du signal augmente sa portée, au détriment du débit car il est transmis sur une plus longue période. Au détriment également de l'autonomie de l'équipement car la communication radio est énergivore !

Par conséquent **une communication plus longue implique une consommation d'énergie plus importante**. Je vous propose ci-dessous un schéma illustrant l'impact de l'étalement du signal sur la portée et sur le débit d'une communication LoRa (pour la bande 868 MHz).

Un réseau LoRaWAN, s'appuyant une modulation LoRa dans la bande de fréquence 868 MHz, supporte 6 facteurs d'étalement (SF7, SF8, SF9, SF10, SF11, SF12) comme le montre le schéma ci-dessus.
L'orthogonalité des SF permet la réception de plusieurs signaux en parallèle sur le même canal. Par défaut, le réseau doit au minimum supporter les trois canaux suivants (de 125 KHz de bande passante chacun) : 868,10 ; 868,30 et 868,50 MHz.

D'autres mécanismes, comme la variation de la fréquence à chaque émission, contribuent également à la robustesse d'une communication LoRa.

Deux équipements LoRa peuvent communiquer en P2P sans nécessairement passer par un réseau.
Cependant, pour que ces équipements communiquent à travers un réseau, l'implémentation de la seule couche physique n'est plus suffisante. C'est ainsi que LoRaWAN définit le protocole réseau (LoRa MAC), pour une communication d'équipements LoRa à travers un réseau.

En s'appuyant sur les performances de la modulation LoRA, le protocole LoRaWAN assure des communications bidirectionnelles et définit ainsi trois classes d'équipements : Classe A, B et C.

La différence entre ces classes réside essentiellement dans le nombre de fenêtres allouées par l'équipement (passerelles) pour la réception des messages envoyés par le réseau.

Chaque équipement LoRaWAN est au minimum compatible avec la classe A ; le support des autres classes est optionnel.
Afin d'optimiser davantage la capacité du réseau, le protocole LoRAWAN prévoit un débit adaptatif – ADR – compris entre 0,3 et 50 kbits par seconde. En fait, le réseau est en mesure d'ajuster le débit des communications et la puissance de sortie de chaque équipement en fonction de sa couverture radio, assurant de ce fait une gestion optimisée des ressources et de la capacité du réseau.

7.2 Architecture d'un réseau LoRaWAN

La topologie d'un réseau LoRaWan est en étoile : les équipements – appelés *end-devices* – communiquent en LoRa avec des concentrateurs – appelés *gateways*. Ces concentrateurs centralisent les messages pour les transmettre au serveur de gestion du réseau. La liaison entre les concentrateurs et le serveur de gestion du réseau s'appuie sur des technologies très haut débit (Ethernet, 4G,…).

Toute l'intelligence, à savoir la gestion du débit adaptatif, de la sécurité des données ou encore de la redondance des données reçues, est assurée par le serveur du réseau. Enfin, ce dernier communique avec un ou plusieurs serveurs applicatifs au travers desquels les fournisseurs d'applications exploitent les données de leur(s) équipement(s).

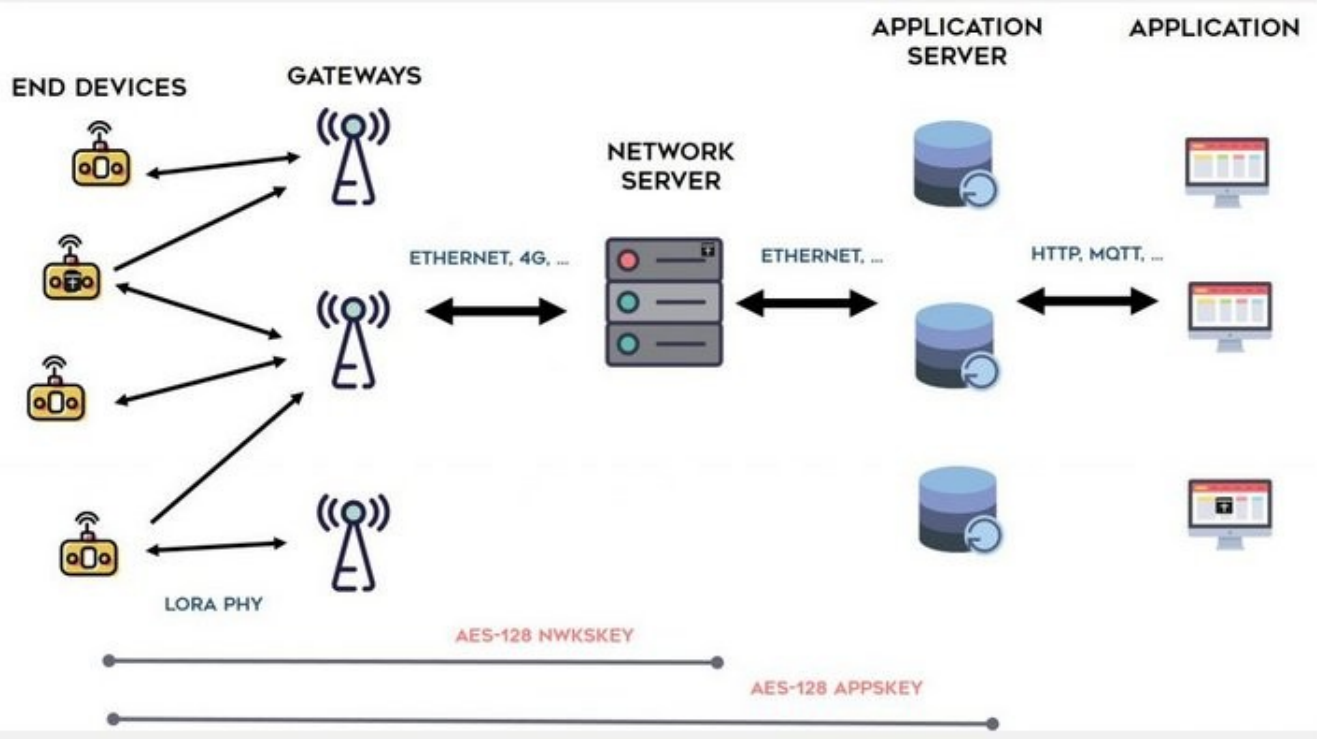

Figure 7.1 Architecture globale d'un réseau LoRaWAN

Car en effet, une des particularités d'un réseau LoRaWAN, est qu'un équipement ne communique pas exclusivement à travers un concentrateur. Tous les concentrateurs couvrant l'équipement peuvent recevoir les données transmises par ce dernier.

Cela facilite grandement la communication avec les équipements en mobilité en dispensant le réseau de mécanismes de hand-over (passage d'un concentrateur à un autre) qui auraient pour effet de complexifier sa gestion et très probablement de réduire ses performances.

Par contre, lorsque le serveur envoie un message à destination d'un équipement, c'est par le biais d'un seul concentrateur.
C'est le cas en particulier des messages requérant un acquittement par le serveur.

7.3 La sécurité d'un réseau LoRaWAN

Maintenant que nous connaissons l'architecture d'un réseau LoRaWAN, une question essentielle reste à traiter : celle de la sécurité. Que ce soit pour assurer la sécurité du réseau ou pour garantir la confidentialité et la sécurité des données, la question de la sécurité est extrêmement importante. Une question qui, soit dit en passant, concerne l'Internet des Objets dans son ensemble.

Pour assurer la sécurité du réseau et des données, le réseau LoRaWAN a recours à deux chiffrements AES-128. Le premier via la clé de session réseau – *Network Session Key* (NwkSKey) – assure l'authenticité des équipements sur le réseau ; quant au deuxième chiffrement, il assure via la clé de session applicative – Application Session Key (AppSKey) – la sécurité et la confidentialité des données transmises à travers le réseau.

En d'autre termes, la clé réseau permet à l'opérateur de sécuriser son réseau alors que la clé applicative permet au fournisseur de l'application de sécuriser les données qui transitent à travers le réseau.

Les données utiles que souhaite transmettre l'équipement sont tout d'abord chiffrées via la clé de session applicative. Un en-tête, contenant entre autres l'adresse de l'équipement, est ensuite ajouté aux données chiffrées. À partir de cette concaténation, le MIC – Message Integrity Code – est calculé via la clé de session réseau. Le MIC permet au réseau de vérifier l'intégrité des données et de l'équipement sur le réseau. Enfin, le MIC est ajouté au message contenant l'en-tête et les données chiffrées avant transmission.
À réception du message par le serveur de gestion du réseau, ce dernier pourra vérifier l'intégrité des données grâce au MIC tout en préservant la confidentialité des données (chiffrées par la clé de session applicative).
Pour illustrer cela, je vous ai préparé un petit schéma :

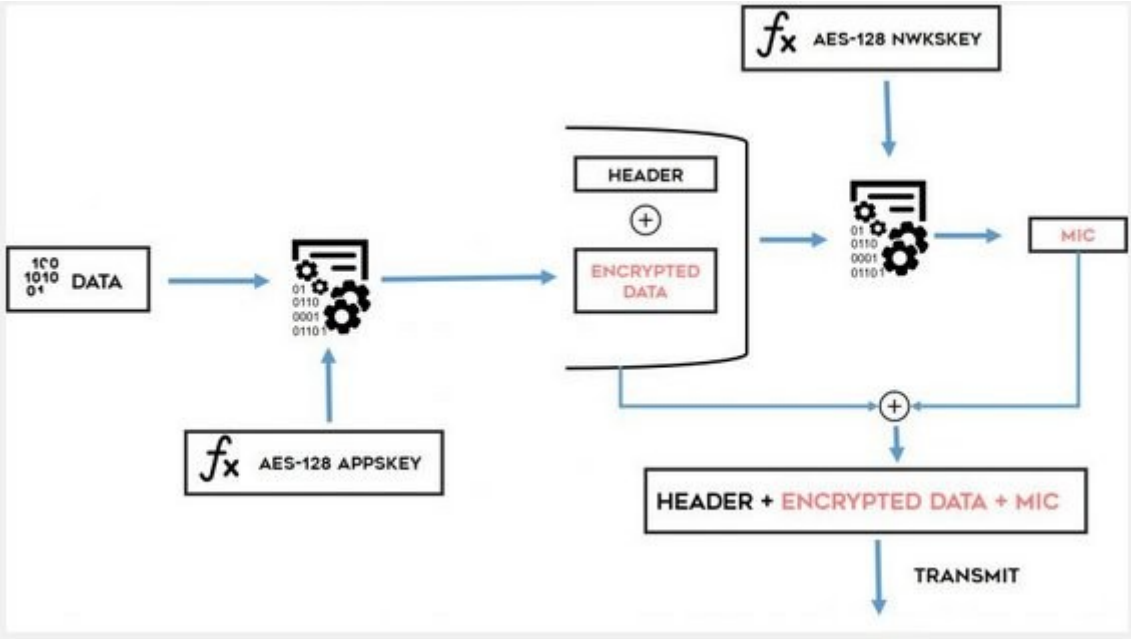

Figure 7.2 Shéma de gestion de la sécurité avec code AES-128

Logiquement, la clé applicative n'est connue que du fournisseur de l'application en question, évitant ainsi qu'un tiers – l'opérateur compris – puisse consulter les données. La clé réseau, pour sa part, est communiquée par l'opérateur du réseau aux fournisseurs d'applications autorisés.
Ces clés permettent de sécuriser les données transmises par les équipements à travers le réseau mais elles sont également indispensables à leur activation sur le réseau.

7.4 Activation d'un équipement LoRaWAN

Avant toute communication à travers un réseau LoRaWAN, les équipements doivent obtenir les clés de session, en suivant une procédure d'activation au choix parmi deux méthodes : Over-The-Air Activation (OTAA) ou Activation By Personalization (APB).

7.4.1 Activation par la méthode OTAA

Pour activer un équipement sur le réseau par la méthode OTAA, l'équipement doit transmettre au réseau une demande d'accès : *join request*. Pour ce faire, celui-ci doit être en possession de trois paramètres :

- Le DevEUI, identifiant unique (de type EUI64) de l'équipement (fourni par l'équipementier).
- AppEUI, identifiant du fournisseur de l'application (EUI 64).
- AppKey, clé AES 128 déterminée par le fournisseur de l'application.

L'équipement envoie, à travers le réseau, la requête *join request,* contenant DevEUI, AppEUI ainsi qu'un MIC calculé via la clé AppKey. Cette requête est transmise au serveur d'enregistrement qui vérifie le MIC via la clé AppKey (qui lui a été communiquée au préalable). Si l'équipement est autorisé par le serveur d'enregistrement, la requête *join accept* est transmise en réponse à l'équipement.
Cette réponse contient des données à partir desquelles l'équipement va pouvoir calculer les clés de session (réseau et applicative). Parmi les données contenues dans cette réponse, se trouve également l'adresse – Device Adress (*DevAddr)* sur 32 bits – qu'utilisera l'équipement pour communiquer sur le réseau.

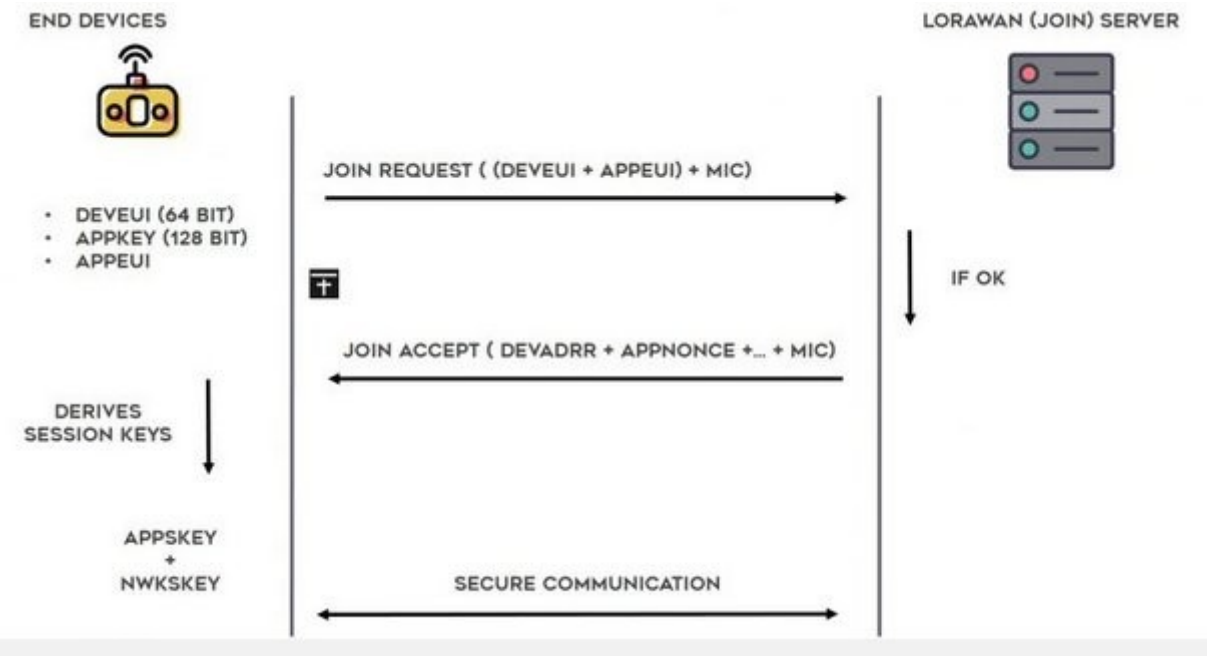

Figure 7.3 Shéma d'activation selon la méthode OTAA

A chaque nouvelle session, les clés de session sont renouvelées.
Je vous rappelle que tous ces mécanismes sont gérés par les serveurs de gestion et d'enregistrement. Les concentrateurs, pour leur part, relaient toutes les données transmises par les équipements présents dans leur zone de couverture, qu'ils soient activés ou non.

7.4.2 Activation par la méthode APB

Pour la seconde méthode, APB, les clés de session NwkSKey et AppSKey ainsi que l'adresse de l'équipement (DevAddr) sont directement inscrits dans l'équipement LoRaWAN. Ainsi, l'équipement n'a plus besoin d'envoyer de requête avant de communiquer sur le réseau. L'utilisation de cette méthode implique que les équipements communiquent avec un réseau spécifique car les clés de session sont connues par avance. Et contrairement à la méthode OTAA, les **clés de session sont statiques**.
En résumé, la méthode OTAA est plus complexe à implémenter que la méthode APB mais offre un niveau de sécurité supérieur. Dans le cas d'un prototypage ou pour une utilisation sur un réseau connu, la méthode APB suffit largement. Par contre, lorsqu'un déploiement à plus grande échelle est envisagé, il est conseillé d'utiliser la méthode OTAA, plus sécurisée et plus agile.

7.5 Réseaux publics VS réseaux privés

Vous pouvez, moyennant le paiement d'un abonnement, connecter vos équipements à un réseau LoRaWAN dit public ; traditionnellement proposé par les opérateurs de téléphonie mobile. À ce jour en France, Orange et Bouygues Telecom proposent tous deux une offre pour l'utilisation de leur réseau LoRaWAN avec l'accès à une plateforme pour la récupération des données.

Même si des opérateurs télécom déploient des réseaux LoRaWAN, il est tout à fait possible pour un particulier, une entreprise ou même une ville, de déployer son propre réseau LoRaWAN. Et détrompez vous, cela n'a rien d'insurmontable.

Tout d'abord, contrairement aux réseaux mobiles, les bandes de fréquences sont libres. Toutefois, elles sont régulées : vous devez notamment respecter un certain temps d'occupation des canaux spectraux (duty cycle). Autrement dit, vous ne pouvez pas communiquer sans interruption ou à très haute fréquence des données.
Ensuite le coût d'infrastructure est plutôt abordable. Bien sûr, il ne s'agit pas de couvrir tout le pays avec son réseau LoRaWAN mais deux concentrateurs peuvent largement suffire à couvrir quelques kilomètres. Le prix d'un concentrateur peut varier de 200 à 2000$ selon les options et les fonctionnalités qu'il propose.
Enfin, en ce qui concerne le serveur applicatif, vous pouvez développer le vôtre ou utiliser celui d'un tiers. Bien entendu, déployer son propre réseau pour y connecter une dizaine d'équipements ne sera pas une opération rentable…

7.6 Les applications LoRaWAN

Compte tenu de ses performances en termes de portée et d'autonomie des équipements, le réseau LoRaWAN permet le déploiement et la remontée de données de capteurs en tout genre (qualité de l'air, remplissage des poubelles, détecteur de présence, …). L'exploitation de ces données ouvre des perspectives considérables dans de nombreux domaines que ce soit pour la gestion des ressources, pour l'optimisation des déplacements ou encore pour l'augmentation de la productivité.

Par exemple, un simple détecteur de présence, qui remonterait l'information d'occupation d'une salle permettrait aux gestionnaires de bâtiments qu'ils soient privés (ruche d'entreprises) ou publics (équipements sportifs) d'optimiser l'utilisation de leurs ressources. Autre exemple, avoir l'information du remplissage des bennes à ordures permettrait le développement d'applicatif d'optimisation des tournées de récolte avec la possibilité d'adapter les véhicules selon les itinéraires de récolte.

On peut encore vous énoncer de nombreux exemples d'applications potentielles pour la maintenance prédictive, la mobilité intelligente, etc. pour lesquelles les réseaux LoRaWAN conviennent parfaitement (à la condition que celles-ci ne nécessitent pas une communication haut débit ou trop fréquente).

7.7 LoRaWAN et LoRaTS

Les spécifications du réseaux type LoRaWAN ont été établie pour répondre aux plusieurs objectives et différentes types de terminaux LoRa.
LoRaTS est un réseau relativement beaucoup plus simple mais également plus spécifique.

LoRaTS est une solution qui s'appuie à la fois sur les caractéristiques spécifiques de contrôleurs intégrés dans les terminaux et dans les modems radio LoRa, tels que la possibilité de fonctionner en mode **deep_sleep**, ne nécessitant que tres peut d'énergie (<1mA) et sur les caractéristiques des services offertes par les serveurs de type ThingSpeak. L'appélation LoRaTS est une contraction de LoRa et ThingSpeak.

Le protocôle LoRaTS permet de communiquer avec plusieurs terminaux (max 254 par une passerelle – ou canal LoRa). La synchronisation, ou accès au canal est contrôlé par la passerelle qui maintient un « agenda » et qui alloue dynamiquement les tranches de temps (**slots**) pour recevoir les données provenant des terminaux. Selon les besoins et les ressources énergétiques disponibles les terminaux peuvent fonctionner en mode **deep_sleep** ou en mode active vis à vis de ses capteurs/actuateurs.
Bien sur le mode deep_sleep est un fonctionnement de préférence.

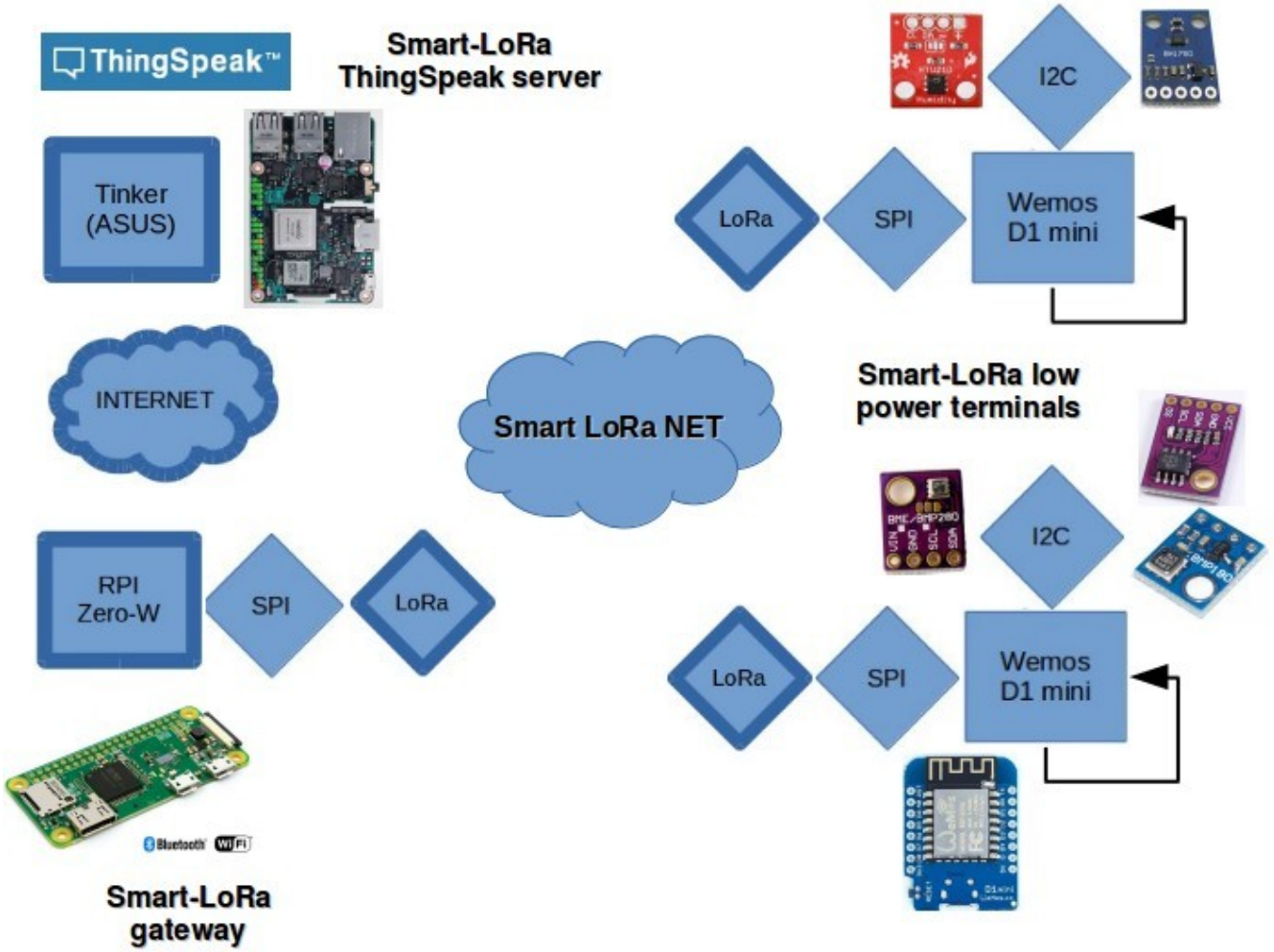

Figure 7.4 Une architecture minimale de LoRaTS avec un serveur, une passerelle, et deux terminaux

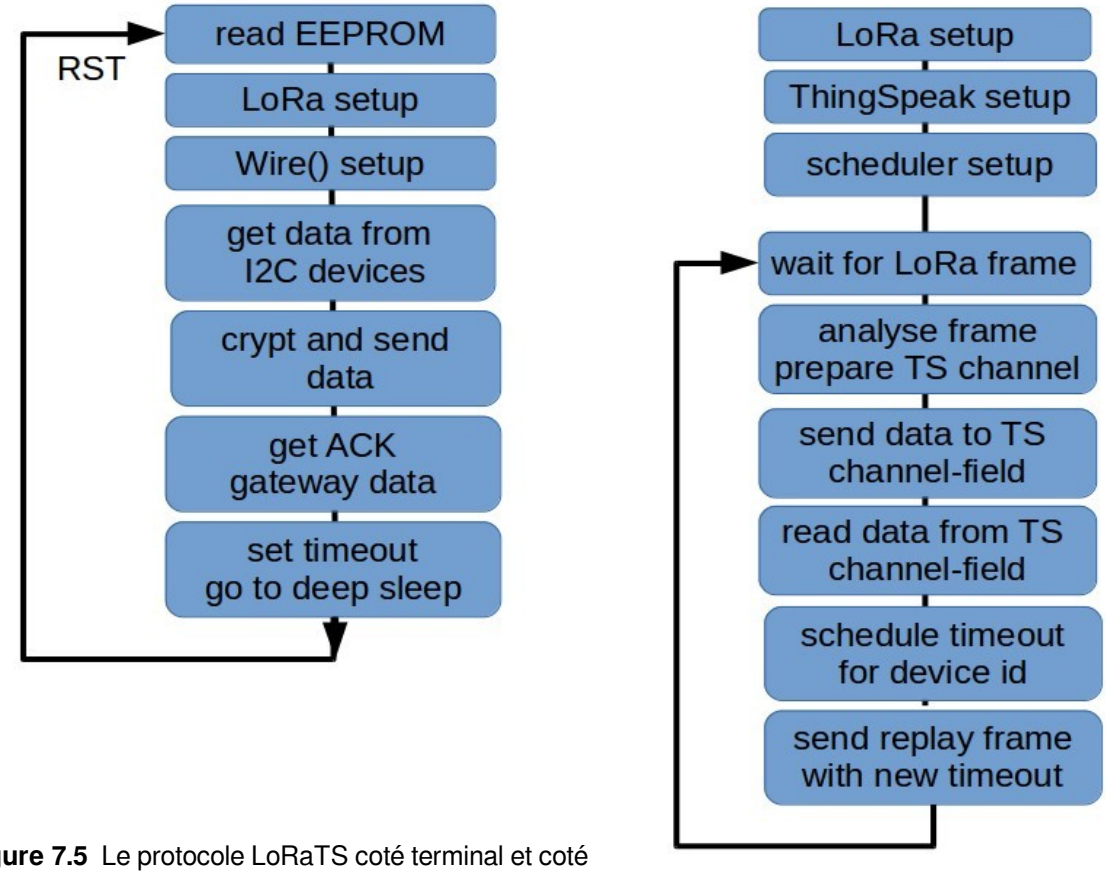

Figure 7.5 Le protocole LoRaTS coté terminal et coté passerelle (ordonnanceur)

Dans le lab 6 nous avons exploré plusieurs facettes du protocole LoRaTS. Notamment nous avons vu la trame LoRaTS, et nous avons expérimenté avec le cryptage/décryptage avec une clé symétrique et un code CRC.

7.7.1 L'ordonnanceur LoRaTS (passerelle)

Ci dessous on vous présente le schéma de base de l'ordonnancer implémenté dans le réseau LoRaTS.
Le fonctionnement de cet ordonnancer est basé sur deux paramètres principaux :

- - nombre de terminaux dans le réseau
- - longueur du cycle d'ordonnancement

Ces paramètres peuvent être initialisés comme suit :

```
int confScheduler()
{
printf("Give the number of devices to be scheduled: \n");
scanf("%u",&devnumber);
printf("Give the length of scheduler cycle (sec): \n");
scanf("%u",&schedcycle);
}
```

Un exemple de valeurs à utiliser : 8 nombre de terminaux, 200 secondes cycle d'ordonnancement.

La fonction principale de l'ordonnanceur calcule le time-out a envoyer vers le terminal actuellement en communication avec la passerelle.
Cette valeur sera utilisée pour calibrer la durée de l'état deep_sleep (ou d'attente) du terminal concerné.

Par défaut chaque terminal utilise la valeur de time-out égale à 30 secondes. On y ajoutant le temps d'initialisation et du traitement la durée d'un cycle par défaut est estimé entre 33 et 34 secondes.

Voici le code C de la fonction d'ordonnancement. L'équation essentielle pour cette fonction est :

timeout= cycle + (did*cycle/dnum - ctime);

ou :

- **cycle** – la durée du cycle d'ordonnancement
- **dnum** – nombre de terminaux (devices) à gérer
- **did** – numéro du terminal (device ID)

La variable **now** contient le nombre de secondes du temps UNIX (nombre de secondes à partir du 1 janvier 1970. La variable **ctime** contient le nombre de secondes dans le cycle d'ordonnancement. Finallement le **timeout** est la valeur a envoyer vers le terminal en communication (device ID).

```
int gettimeout(uint32_t cycle, uint32_t dnum, uint8_t did)
{
uint32_t ctime,now;
int timeout;
struct timeval tv;
gettimeofday(&tv,NULL);
now = (uint32_t)tv.tv_sec;
printf("now: %u\n",now);
ctime = now%cycle;
printf("ctime: %u\n",ctime);
timeout= cycle + (did*cycle/dnum - ctime);
return timeout;
}
```

Les informations affichées par la passerelle

```
frame ok!, id=1                  // trame correcte et identificateur valable
got reply RSSI= -35              // puissance du signal reçu
1=24.96&2=56.39&3=118.00&4=8.98&t=30  // données de la trame LoRaTS
field1=24.96&field2=56.39&field3=118.00&field4=8.98
Sent to TS!  // les données pour ThingSpeak bien envoyées
now: 1506298773  // temps réel de la passerelle
ctime: 33        // temps dans le cycle (cycle de 60 secondes)
Send to LoRa!
frame ok!, id=1  // trame envoyée vers le terminal id=1
got reply RSSI= -36 // cycle suivant
1=24.91&2=56.61&3=117.00&4=8.98&t=30
field1=24.91&field2=56.61&field3=117.00&field4=8.98
Sent to TS!
now: 1506298833
ctime: 33
Send to LoRa!
```

7.7.2 Les capacités et la gestion d'un réseau LoRaTS

Un réseau LoRaTS peut contenir maximum 254 terminaux.
L'identificateur 0 est réservé pour la passerelle, l'identificateur 255 pour la diffusion (**`BROADCAST`**).

Chaque terminal peut communiquer (via la passerelle) avec un canal de **ThingSpeak**, mais plusieurs terminaux peuvent utiliser le même canal.

L'organisation des terminaux vis-a-vis d'un serveur **ThingSpeak** dépend de l'application. Par exemple dans un canal on peut intégrer les information sur la température à partir de 8 terminaux.

La gestion de cette association terminal-canal TS est effectuée par la passerelle et mémorisée dans son **fichier de gestion**.

Initialement tous les terminaux du même réseau utilisent le même code de cryptage. Cette caractéristique peut être modifiée selon les besoins de l'application.

Projet final (évaluation sur 20 pts)

On vous propose de spécifier votre **mini-projet**

- (application) , choix de capteurs, de terminaux, etc.
- d'adapter les caractéristiques du réseau LoRaTS à vos besoins
- mettre en œuvre l'ensemble de cette application.

Remarque :

Un autre moyen de communication radio peut être également envisagée.
HC-12, Wifi, Bluetooth, ..