

Rapport final

Rapport de la semaine du 17 au 23 juin 2019

Table des matières

Introduction.....	1
But du projet.....	1
Solution proposée.....	2
Dernières précisions.....	2
Les installations complètes.....	2
Sur le serveur.....	2
LAMP depuis les dépôts.....	3
Configurer le root de MySQL.....	3
PHPMyAdmin.....	3
Sur le Raspberry Pi.....	4
Connexion SSH par paire de clés.....	4
Utilisation de l'antenne avec le RTL-SDR.....	4
La pensée.....	4
Outil.....	5
Processus principal.....	5
Formats des données et traitement de ces dernières.....	6
Données renvoyées par dump1090.....	6
Table présente dans la base de donnée.....	7
Scripts pour le flux des coordonnées.....	8
Script coté Raspberry Pi.....	8
Script coté serveur.....	8
Exploitation de la base de donnée par la page Web.....	9
Page Web coté client.....	9
Script PHP coté serveur.....	9
Conclusion.....	9
Première version terminée.....	9
Améliorations éventuelles.....	9
Annexes.....	9

Introduction

Le but de ce rapport est de résumer toutes les avancées que nous avons effectué au cours de l'année ainsi que des éventuelles améliorations que nous pourrions effectuer après.

Le travail que nous avons effectué cette semaine y est bien évidemment inclus.

But du projet

Nous rappelons l'énoncé : proposer une page Web positionnant les avions de lignes survolants l'IUT sur une carte.



Depuis la disparition de l'avion de ligne appartenant à la compagnie Malaysia Airline, une nouvelle norme aéronautique a été imposée pour les avions de ligne, qui est d'envoyer leur coordonnées toutes les secondes à d'éventuels récepteurs. Ce trafic n'étant pas chiffré, nous pouvons donc nous aussi capter les coordonnées GPS de ces avions.

Solution proposée

Nous avons donc proposer d'utiliser une antenne que l'on peut relier à un ordinateur qui communique avec un serveur. L'ordinateur est chargé de relever les coordonnées captées par l'antenne à laquelle il est relié et de les envoyer vers un serveur. Le serveur en question sera chargé d'envoyer les coordonnées reçues dans une base de donnée qu'il hébergera ainsi que d'héberger la page Web demandée. Nous avons choisi d'utiliser un Raspberry Pi relier à l'antenne pour sa portabilité.

Dernières précisions

Ce rapport traite de programmes que nous avons développé. Il sont normalement joints à ce rapport. Nous ne ferons que parfois résumer brièvement leur fonctionnement, les détails sont présents dans les fichiers avec les commentaires adéquats.

Les installations complètes

Cette partie traite des installations à effectuer sur le serveur et sur le Raspberry Pi. Elle a déjà été traitée dans le précédent rapport mais cette fois-ci des explications l'accompagne afin de mieux comprendre pourquoi est-ce que l'on fais ces modifications. De plus, il y a des précisions supplémentaires que nous avons pu élaborer lors de nos test.

Cette partie ne traitera pas du programme permettant de communiquer avec l'antenne, elle décrit les besoins pour le circulation des coordonnées vers la base de donnée ainsi que pour la page Web. Cette partie ne traite donc que des installations logicielles que nous avons besoin sur nos machines et non sur le matériel qui sera traité dans la partie suivante.

Le serveur et le Raspberry utiliseront tous les deux l'OS GNU/Linux. Une petite précision pour la suite :

```
$ commande en tant qu'utilisateur  
# commande en tant que root
```

Sur le serveur

Sur le serveur va donc être installé LAMP, pour Linux Apache MySQL PHP. Il y aura donc tous les fichiers nécessaires afin de faire fonctionner la base de donnée que nous avons besoin ainsi que la page Web.

Apache servira donc pour la page Web, afin que notre serveur réponde aux navigateur en tant que serveur Web en affichant notre carte.

MySQL servira à héberger la base de donnée qui contiendra les coordonnées GPS des avions ainsi que d'autres coordonnées éventuelles.

PHP servira à l'envoi automatique des coordonnées vers la base de donnée mais aussi pour l'exploitation des données de cette dernière pour la page Web, soit pour communiquer avec la base de donnée.

LAMP depuis les dépôts

Mise à jour des dépendances :

```
# apt-get update
```

Installation d'Apache, MySQL et PHP avec les librairies nécessaires :

```
# apt-get install apache2 php libapache2-mod-php mysql-server php-mysql
```

Configurer le root de MySQL

Se connecter en root sans mot de passe à MySQL :

```
# mysql -u root
```

Saisir un mot de passe pour le root (ici « root ») :

```
mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root';
```

Quitter mysql avec exit :

```
mysql > exit
```

PHPMyAdmin

Nous décidons d'installer en plus PHPMyAdmin afin d'avoir une interface graphique pour l'utilisation de notre base de donnée, plus rapide et plus simple selon nous.

```
# apt install phpmyadmin
```

Lorsqu'ils demandent de saisir le mot de passe, saisir « root ».

Certaines erreurs dans les fichiers installés sont présentes par rapport à la version trop récente de PHP. Il faut donc les corriger :

Afin de pouvoir uploader les coordonnées par fichier vers la BDD, ouvrir le fichier :

```
# nano /usr/share/phpmyadmin/libraries/plugin_interface.lib.php
```

Trouver la ligne contenant :

```
if ($options != null && count($options) > 0) {
```

La remplacer par :

```
if ($options != null && count((array)$options) > 0) {
```

Afin de pouvoir afficher les résultats des requêtes SQL correctement, ouvrir le fichier :

```
# nano /usr/share/phpmyadmin/libraries/sql.lib.php
```

Trouver la ligne contenant :

```
(count($analyzed_sql_results['select_expr'] == 1)
```

La remplacer par :

```
((count($analyzed_sql_results['select_expr']) == 1)
```

Afin de pouvoir voir combien d'avions nous avons captés depuis le temps que l'on souhaite, on effectue l'affichage de la table en regroupant les résultats par numéro d'avion. Il faut donc changer une variable système :

Aller à l'accueil > Variables > Sélectionner la variable « sql mode », la modifier en retirant la partie « ONLY_FULL_GROUP_BY, ».

Sur le Raspberry Pi

Connexion SSH par paire de clés

Le Raspberry va récupérer en permanence le retour console d'un programme qui communiquera directement avec l'antenne. Comme il s'agit de l'envoi d'un seul fichier, il est inutile de créer un quelconque serveur FTP. Nous décidons d'utiliser la commande « scp » en permanence afin d'envoyer le résultat au serveur.

Comme il s'agit donc une connexion ssh en permanence, il ne faut surtout pas qu'un mot de passe soit demandé à chaque envoi, sinon rien ne sera automatique. La solution est donc d'utiliser une paire de clé SSH, ce qui fait que l'authentification avec le serveur se fera par clé et non plus par mot de passe, ce qui sera donc totalement automatique.

Sur le Raspberry Pi, en tant qu'utilisateur pi :

Dans le dossier privé SSH de pi :

```
$ cd ~/.ssh
```

Création de la clé SSH :

```
$ ssh-keygen -t rsa
```

Laisser le nom par défaut (id_rsa) et ne pas saisir de passphrase.

Envoi de la clé publique au serveur :

```
$ ssh-copy-id user@ipDuServeur
```

Utilisation de l'antenne avec le RTL-SDR

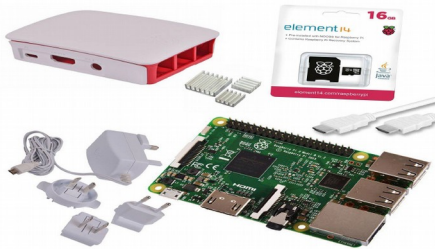
La pensée

L'ADS-B est une surveillance automatique de corrélation de type diffusion, c'est-à-dire que, sans intervention manuelle, elle peut automatiquement obtenir des paramètres de l'équipement embarqué approprié pour diffuser la position, l'altitude, la vitesse, le cap, le numéro d'identification et d'autres informations de l'aéronef à d'autres aéronefs ou stations au sol.

Le contrôleur est surveillé par le contrôleur. La fréquence de communication de l'aéronef civil est de 1090 MHz et la barre de télévision (RTL2832U + R820T) reçoit une plage de fréquences de 24 ~ 1766 MHz. Et parce que les normes aéronautiques sont ouvertes (emplacement, magnitude relative, etc.). De plus, en transmettant des informations en texte clair, le logiciel de démodulation (SDR) peut être utilisé pour démoduler correctement le signal.

Outil

Système Linux (Raspberry Pi)



RTL-SDR / Antenne



Voici à quoi cela ressemble une fois l'installation terminée :



Processus principal

Nous devons d'abord installer un logiciel: dump1090

Ici, nous devons expliquer ce qu'est dump1090

C'est un simple décodeur en mode S qui peut être utilisé avec SDR pour accepter les informations textuelles des avions.

Terminal ouvert

Développer une bonne habitude pour construire un dossier dédié

```
mkdir sdr
```

Cloner le source sur la machine virtuelle sur github et compiler

```
cd sdr
git clone https://github.com/antirez/dump1090.git
cd dump1090/
make
```

Ensuite, utilisez la commande pour vérifier

```
lsusb
```

Avec cette commande, nous pouvons trouver le port auquel rtl-sdr est connecté.

Après la découverte, vous pouvez exécuter la commande pour exécuter le fichier.

```
cd sdr
cd dump1090
./dump1090
```

Les commandes d'exécution peuvent également ajouter des paramètres

```
./dump1090 --interactive
```

Cela montrera un meilleur mode d'interaction, nous avons des informations de vol:

Hex	Mode	Sqwk	Flight	Alt	Spd	Hdg	Lat	Long	Sig	Msgs	Ti/
440486	S	1113	EJU85FQ	37000					5	4	1
4D00F5	S								3	3	20
710124	S								4	2	25
34508A	S	1000							5	3	11
345043	S	5567	VLG8154	34000	473	002	43.125	1.266	6	46	23
342090	S	5303		36025					4	25	31
44CE4F	S	7134		39000	418	177	44.305	2.727	6	36	26
4841DB	S	5341		28700					4	11	28
491306	S	1051		30000					4	22	51
440A8F	S	6767	EJU28FT	28300	365	232	42.338	4.054	6	243	1
406C80	S	5346	EXS128A	38000	473	354	42.104	1.566	6	259	0
4853D3	S	5373	TRA57R	27350	398	321	42.498	2.144	6	491	0
0A0020	S	1000	DAH1052	35975	413	305	42.451	4.042	10	447	0
344158	S	1132	VLG18WZ	35600	431	179	43.633	2.854	8	241	0
406267	S	2163	CLF574	41000					5	453	33
4CA84E	S	1000	RYR1LW	19150	355	104	43.618	3.770	5	2206	0
3B777D	S	1000	FAF4110	11000			43.342	3.646	6	1920	4
4CA34B	S	1042	RYR9LY	38000	455	353	42.451	1.515	7	221	0
345646	S	0736	VLG5WX	30375	416	211	42.356	3.906	4	1036	51
44003E	S	1000	EJU29QN	36025	522	039	44.282	2.780	5	196	21
392AEF	S	7661	AFR22JR	13250	417	191	42.107	2.750	4	4532	44
3B77DD	S	4361	FAF9025	32000					6	1099	3

Enfin, nous avons confirmé l'exactitude des informations sur l'aéronef en entrant le numéro de l'aéronef recherché sur FlightAware



La commande permettant donc de récupérer le flux console suivant dans un fichier texte est donc la suivante :

```
./dump1090 --interactive -net 1> data.txt
```

Formats des données et traitement de ces dernières

Données renvoyées par dump1090

Comme nous avons pu le voir précédemment, les données captées par l'antenne sont bien plus que ce à quoi nous nous attendions. Avec la commande de base, nous avons accès en tout aux coordonnées suivantes :

Hex	Numéro de série de l'avion, ce qui sera utilisé pour différencier les avions sur la carte
Mode	Mode de capture (par défaut S)
Sqwk	Code que saisi le commandant de bord afin de renseigner l'état de l'avion
Flight	Numéro de vol
Alt	Altitude par rapport au niveau de la mer

Spd	Vitesse
Hdg	Heading : direction de l'avion
Lat	Latitude
Long	Longitude
Sig	Force du signal
Msgs	Nombre de messages reçus pour un même avion depuis le démarrage du programme
Ti	Temps depuis lequel un message a été reçu d'un avion (en seconde)

Cependant, en retour texte dans un fichier, on ne peut avoir accès au mode ainsi qu'au Sqwk, ce qui peut s'expliquer par la méthode avec laquelle dump1090 écrit dans un fichier texte.

Le reste des coordonnées seront donc envoyées en base de donnée.

Table présente dans la base de donnée

Toutes les coordonnées que nous pouvons avoir seront toutes stockées dans une même table, bien que la plus part seront à titre informatives. Cependant une information cruciale pour la page Web manque, la date à laquelle on reçoit les coordonnées, afin d'afficher sur la page Web les avions en fonction de la récence de leur coordonnées.

La table aura donc les champs suivants :

Nom	Type	Interclassement	Valeur par défaut
prim	int(11)		Aucune
hex	varchar(6)	utf8_general_ci	NULL
mode	varchar(1)	utf8_general_ci	NULL
sqwk	int(11)		NULL
flight	varchar(7)	utf8_general_ci	NULL
alt	int(11)		NULL
spd	int(11)		NULL
hdg	int(11)		NULL
lat	double		NULL
lng	double		NULL
sig	int(11)		NULL
msgs	int(11)		NULL

Mise à part le premier champ qui est la clé primaire, nous laissons les autres champs avec comme valeur par défaut « NULL », ce qui permettra de ne pas avoir à saisir de valeur pour les champs donc nous n'avons pas de coordonnée.

Scripts pour le flux des coordonnées

Les scripts à faire fonctionner sont des script codé pour être exécuté en permanence, seul un arrêt brutal peut les faire cesser de fonctionner. Nous sommes bien conscients qu'il s'agit d'une méthode peut recommandable, mais une autre méthode comme l'utilisation des CRON TABLE ne marchait tout simplement pas.

Script coté Raspberry Pi

Le script sur le Raspberry sera placé dans le répertoire /home/pi/sdr/dump1090/. En permanence, il lancera en démon le programme afin de continuer en parallèle l'exécution du script. Après un certain nombre de secondes que nous aurons fixé, le script « kill » le processus correspondant au programme dump1090 puis envoi le contenu du fichier dans lequel il a écrit vers le serveur.

Script coté serveur

Le script coté serveur va vérifier en permanence, à une fréquence bien plus élevée que l'envoi du Raspberry, si le fichier envoyé existe bel et bien. Si oui, il va donc en prendre le contenu et le remettre en forme afin de le mettre dans un autre fichier qui lui sera prêt à être envoyé vers la BDD.

La remise en forme du contenu se fait en supprimant d'abord les lignes inutiles puis en remplaçant certains groupes de caractères par d'autres. La commande permettant de remettre en forme le contenu se base sur la redirection de flux de sortie de commande successives avec le caractère « | ».

En voici l'explication en détail sous forme de tableau :

cat data.txt	Affichage du contenu du fichier reçu
grep -v 2J	Enlèvement des lignes « headers »
grep -v -	Enlèvement des lignes tirets, faisant la séparation entre les « headers » et les données
grep -v 0.0	Enlèvement des lignes avec une latitude ou longitude de 0
sed 's/ sec//g'	Enlèvement des caractères « sec »
grep -v " "	Enlèvement des lignes n'ayant pas de numéro de vol
grep " "	Enlèvement des lignes non complètes
sed 's/ / /g' (x3)	Remplacement de tous les groupes d'espaces par un seul
sed 's/ /,/g'	Remplacement de tous les espaces par une virgule (notation CSV)
> coordonnees.txt	Envoi du flux final vers le fichier qui sera importé en BDD

Après avoir remis en forme les données dans un autre fichier, ce dernier sert à envoyer vers la BDD les données par requête SQL utilisant un fichier texte, ce qui est bien plus pratique et bien plus rapide. Le script se charge donc à la fin d'exécuter avec l'interpréteur PHP le script PHP permettant de faire cette requête.

Exploitation de la base de donnée par la page Web

Cette partie a déjà été brièvement expliquée dans le rapport N°19. Le consulter pour plus de détails.

Page Web coté client

La page Web coté client va donc être la représentation de l'exploitation de la base de donnée. Elle contient donc la carte faite avec l'API d'Open Street Map avec les boutons de navigations mais aussi le code JavaScript qui sera exécuté sur le client.

Script PHP coté serveur

Le script PHP coté serveur sera chargé de vérifier si de nouvelles coordonnées sont arrivées avec le fichier qu'aura écrit le script PHP à « 1 ». Si le contenu de ce fichier est « 1 », il va donc chercher en BDD les coordonnées les plus récentes selon un critère de récence que nous aurons fixé et remettre le contenu du fichier à « 0 ».

La page Web coté client se contentera de faire recharger la carte seulement si les données renvoyées par le serveur sont des coordonnées.

Conclusion

Première version terminée

Nous avons donc réussi à remplir l'objectif principal de notre projet. Nous pourrions vous montrer en direct le fonctionnement de notre système ou simplement vous montrer les coordonnées que nous avons pu enregistrer dans notre base de donnée.

Nous considérons que cette version est présentable en tant que concrétisation du projet pour une première version.

Améliorations éventuelles

Les améliorations éventuelles que nous pourrions faire serait matérielles et logicielles :

- Essayer d'avoir ou de concevoir une antenne plus précise afin de capter plus d'avions, plus vite
- Améliorer la page Web en exploitant les autres coordonnées que nous n'utilisons pas ou encore améliorer l'apparence afin de la faire ressembler plus à une page Web de l'IUT

Annexes

Source ayant lancé notre projet :

<https://www.kaspersky.fr/blog/suivi-des-avions-comment-flightradar24-fonctionne/4457/>

Sources pour l'installation du serveur :

<https://doc.ubuntu-fr.org/lamp>

<https://doc.ubuntu-fr.org/mysql>

<https://www.hostinger.fr/tutoriels/generer-cle-ssh/>

<https://devanswers.co/problem-php-7-2-phpmyadmin-warning-in-librariessql-count/>

<https://stackoverflow.com/questions/23921117/disable-only-full-group-by>

Sources pour la page Web :

<https://nouvelle-techno.fr/actualites/2018/05/11/pas-a-pas-inserer-une-carte-openstreetmap-sur-votre-site>

<https://www.borelly.net/cb/>

Ainsi que bien évidemment tous les cours et autres conseils qui nous ont été fournis.