

Réalisation d'une API de configuration automatique de réseau WiFi maillé sous Linux

BUGNOT Thibaut

22 février 2018

Table des matières

Introduction	2
1 Objectifs, définitions, contraintes	3
1.1 Introduction aux réseaux wifi	3
1.2 La norme 802.11s	4
1.3 Adressage et routage	6
2 Réseaux sans fils sous Linux	7
2.1 Gestion du réseau sous Linux	7
2.2 Création de réseaux 802.11s	7
2.3 Détection de réseaux existants et sélection de canal	7
3 Adressage et routage	8
3.1 Adressage dans un réseau maillé	8
3.2 Routage	8
4 Implémentation et architecture logicielle	9

Introduction

Chapitre 1

Objectifs, définitions, contraintes

1.1 Introduction aux réseaux wifi

Le wifi, abréviation de wireless fidelity, est un ensemble de protocoles permettant la communication sans fil entre deux appareils en utilisant des ondes radios. Ces protocoles se situent au niveau de la couche d'accès du modèle tcp/ip. La standardisation de cette norme a été initiée l'IEEE¹ en 1990. Cela a abouti, en 1997, au standard IEEE 802.11 définissant les réseaux locaux sans fils [1]. La norme d'origine prévoyait l'utilisation d'ondes radios dans la bande de fréquences libre entre 2401 et 2495 MHz[2], couramment appelée bande à 2,4 GHz, ou d'infra rouges. Cependant, pour suivre l'évolution des technologies, le standard IEEE 802.11 s'est enrichi afin d'augmenter le débit et d'utiliser la bande de fréquences libre entre 5170 et 5710 MHz. Les standards IEEE 802.11a et IEEE 802.11b ont donc été définis en 1999, le standard 802.11g en 2003 et le standard 802.11n en 2009.

Depuis sa création, la norme IEEE 802.11 définit 14 canaux dans la bande 2,4 GHz. Chaque canal a une largeur de 22 MHz et l'écart entre les centres de deux canaux successifs est de 5 MHz². Il en résulte donc un fort recouvrement entre les différents canaux comme le montre la figure 1.1.

Un réseau wifi est un réseau local découpé en "cellules" appelée BSS³. Deux appareils doivent se trouver dans le même BSS pour communiquer entre eux.

1. Institute of Electrical and Electronics Engineers

2. Sauf les centres des canaux 13 et 14 qui sont espacés de 12 MHz

3. Basic Service Set

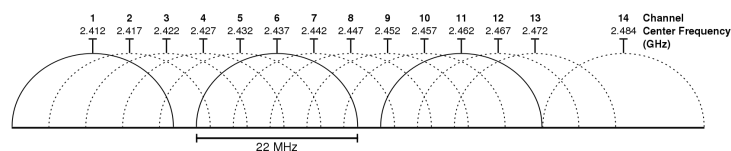


FIGURE 1.1 – Répartition des canaux dans la bande 2,4 GHz

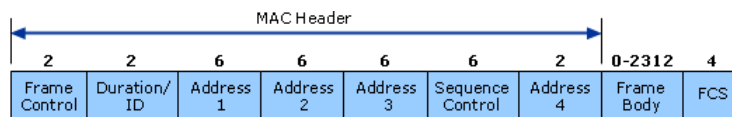


FIGURE 1.2 – Format des trames 802.11

Il existe deux modes de BSS : Le mode Infrastructure et le mode ad-hoc[3]. La plupart des réseaux wifi de particuliers ou d'entreprises sont des réseaux en mode Infrastructure.

Le mode infrastructure est une topologie centralisé. Il se caractérise par le fait que chaque BSS possède une station de base, appelé aussi point d'accès, et que toutes les communications passent nécessairement par le point d'accès de la BSS, et ce même si l'émetteur et le récepteur du message se trouvent dans le même BSS. Un point d'accès peut être relié par un réseau cablé à un ou plusieurs autres points d'accès, étendant ainsi le LAN⁴ [4], ou à un routeur pour accéder à un réseau WAN⁵. Le mode ad-hoc, au contraire, est un mode "d'égal à égal". Deux entités au sein du même BSS peuvent communiquer directement.

Comme le montre la figure 1.2[5], le premier champ de l'en tête wifi est le FCF⁶, permettant d'identifier les trames en fonction de leur rôle. Ainsi, les trames peuvent être de trois types, identifiées par les deux bits en position 3 et 4 du FCF : Management, Contrôle ou Données.[6]. Les 4 bits suivants identifient le sous type, et les 8 derniers bits sont des flags. Les trames de données sont utilisées pour transporter des données de plus haut niveau. Les trames de contrôles sont utilisées pour les acquittements et les réservations, et les trames de management servent à organiser et maintenir le réseau[7].

Les Beacon frames sont des trames de management particulières qui permettent à un point d'accès de déclarer sa présence aux appareils à proximité. Ils transportent différentes informations comme le SSID⁷ du réseau, qui est une chaîne de 2 à 32 caractères, un timestamp permettant de se synchroniser, le canal sur lequel il émet, et d'autres informations.[3].

1.2 La norme 802.11s

Comme dit précédemment, le mode infrastructure est actuellement le plus utilisé. Cependant, il possède des limites du fait que, dans certaines situations, il n'est pas toujours possible de connecter un point d'accès à un switch[8]. En effet, la longueur des câbles ethernet est limitée, ce qui rend difficile le déploiement de points d'accès dans des environnements ouverts.

C'est ce qui fait la force du mode ad-hoc. Chaque appareil peut communiquer avec tous les autres appareils qui sont à portée. De plus, chaque appareil peut relayer le message si le destinataire final n'est pas à portée. Ainsi, si l'on prends l'exemple la figure 1.3, chaque nœud peut communiquer avec n'importe quel autre, à la condition qu'un algorithme de routage s'exécute sur le réseau et que chaque nœud sache quel est le suivant pour atteindre la destination. Ce genre

4. Local Area Network, ou Réseau local

5. Wide Area Network, ou Réseau étendu

6. Frame Control Field, ou Champ de contrôle de trame

7. Service Set Identifier

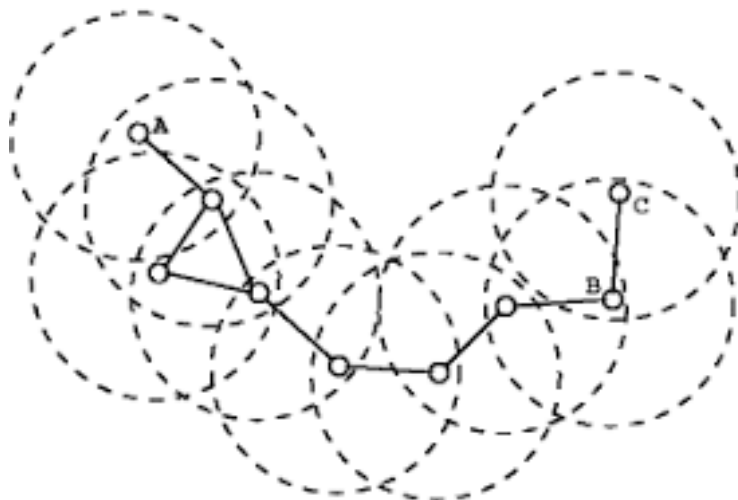


FIGURE 1.3 – Exemple de réseau ad-hoc

de réseau est appelé réseau maillé⁸. Le gros avantage de ces réseaux est qu'ils sont très flexibles. On peut les étendre sans avoir à tirer de nouveaux câbles ou à ajouter de nouveaux équipements intermédiaires[8]. À l'inverse, le retrait d'un petit nombre de nœuds ne doit pas empêcher le réseau de fonctionner si il est possible de trouver des routes alternatives pour les trames.

Le standard 802.11s est un amendement de la norme 802.11, définissant la manière dont les appareils disposant d'une carte réseau sans fil peuvent s'interconnecter pour former un réseau sans fil maillé. L'IEEE a commencé à travailler sur ce standard en 2003 et celui-ci a été adopté en 2006. Pour faciliter l'interopérabilité, un réseau 802.11s est vu de l'extérieur comme un unique segment ethernet. Pour permettre la retransmission des informations d'un nœud à l'autre, la norme 802.11s étend l'en-tête 802.11 classique avec un en-tête mesh comme montré dans la figure 1.4[7].

Les 4 champs d'adresses de l'en-tête 802.11 sont utilisées, puisqu'il faut à chaque transmission du message donner l'adresse du nœud qui a effectué la transmission, du prochain nœud, du destinataire final et de l'expéditeur originel. Dans certains cas plus complexes, par exemple si l'émetteur ou le destinataire, ou les deux, ne se trouvent pas dans le réseau mesh, mais que la trame va traverser un réseau mesh, il faut ajouter des adresses supplémentaires, d'où le fait que l'en-tête mesh comporte un champ optionnel d'extension d'adresse. Parmi les autres valeurs ajoutées, le TTL⁹ et le Mesh sequence number¹⁰ permettent d'éviter les boucles infinies qui risqueraient de saturer le réseau.

8. Mesh Network en anglais

9. Nombre de fois maximal que peut être relayé une trame avant d'être abandonnée, cette valeur est décrémentée à chaque saut

10. nombre identifiant de manière unique un paquet

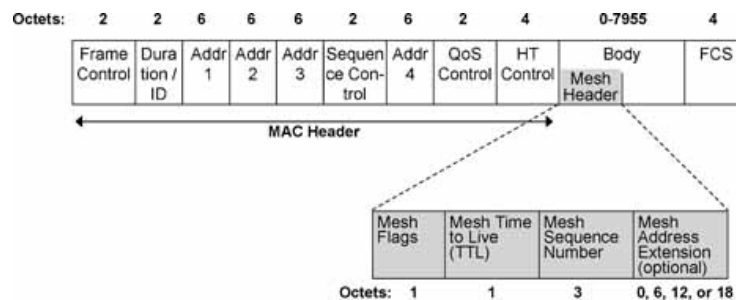


FIGURE 1.4 – Format des trames 802.11s

1.3 Adressage et routage

Dans un réseau TCP/IP, chaque noeud doit disposer de deux adresses. Chacune permet de l'identifier, en théorie, de manière unique. La première est l'adresse MAC, une adresse sur 48 bits, utilisé pour identifier les noeuds dans les protocoles de la couche d'accès du modèle TCP/IP. Cette adresse permet à une trame de voyager sur un LAN jusqu'à sa destination, mais sera changé à chaque fois que le paquet passe par un routeur. La deuxième est l'adresse IP, une adresse sur 32 bits qui est utilisé par le protocole IP, qui est un protocole de la couche réseau du modèle TCP/IP. Cette adresse est inchangé d'un bout de la transmission à l'autre ¹¹.

L'adresse MAC est attribué à une carte réseau par le constructeur. Ainsi, nous avons la garantie que chaque appareil possède une adresse MAC unique. L'adresse IP doit également être unique mais, contrairement à l'adresse MAC, elle n'est pas enregistré dans la carte réseau par le constructeur car toutes les adresse IP identifiant les appareils d'un même LAN doivent avoir le même préfixe. Il existe des protocoles permettant d'affecter automatiquement des adresses IP à des appareils sans avoir besoin de recourir à une intervention humaine. Le protocole majoritairement utilisé est DHCP ¹². Ce protocole nécessite qu'un serveur dispose d'une liste d'adresses IP disponible qu'il va affecter à chaque noeuds du réseau sur demande de ces derniers[9]. Néanmoins, le recours à un serveur central d'adresse IP amoindrit les avantages à l'utilisation d'une infrastructure décentralisé tel qu'un un réseau maillé.

Dans un réseau maillée, il est aussi nécessaire de prévoir le routage des trames. La norme 802.11s définit également le protocole HWMP ¹³ comme protocole de routage pour les réseaux wifi maillées. Contrairement à la majorité des protocoles de routages, HWMP ne se base pas sur les adresses IP, mais sur les adresses MAC, puisque le but est d'aiguiller les trames au sein d'un même LAN. Il s'agit d'un protocole de routage à vecteur distance puisque les noeuds n'ont pas connaissance de l'intégralité de la topologie du réseau mais uniquement des noeuds qui le constituent et de la "distance" de chacun d'eux [10].

11. en l'absence de mécanismes de traduction d'adresse (NAT)

12. Dynamic Host Configuration Protocol

13. Hybrid Wireless Mesh Protocol

Chapitre 2

Réseaux sans fils sous Linux

2.1 Gestion du réseau sous Linux

Une grosse partie du projet consiste à pouvoir communiquer avec les interfaces sans fils dont dispose les appareils. En effet, il est nécessaire d'une part, de récupérer des informations à propos de ces dernières et d'autre part, de les configurer pour arriver à les utiliser de la manière que nous le souhaitons. Historiquement, sous linux, la communication avec les interfaces se faisait avec des appels systèmes `ioctl`¹. Des outils permettant de manipuler les interfaces en utilisant cette méthode sont depuis longtemps fournis avec les distribution linux. C'est le cas par exemple du package `net-tools`, incluant le programme `ifconfig`, et permettant de manipuler les interfaces (état, informations d'adressages) ou de `wireless-tools`, incluant le programme `iwconfig`, permettant de manipuler plus précisément les interfaces sans fils.

Cependant, depuis 2007, il existe un autre moyen de manipuler les interfaces. En effet, se développe **netlink**, une famille de socket ayant pour but de faire communiquer les processus entre eux. Cela permet, entre autre, de faire communiquer un processus utilisateur avec un processus du noyau linux. La librairie `libnl` implémente les pré-requis fondamentaux pour utiliser le protocole `netlink`. Cependant, celle ci se veut minimaliste. C'est pourquoi elle est complétée par 3 API : `libnl-route`, `libnl-genl` et `libnl-nf`[11]. Des outils de configuration d'interfaces utilisent ces nouveaux moyens. C'est le cas de la suite d'outil `iproute` pour contrôler les interfaces et de `iw` pour contrôler plus précisément les interfaces sans fils. C'est également sur cette librairie que sera basé notre code.

2.2 Création de réseaux 802.11s

2.3 Détection de réseaux existants et sélection de canal

1. Abréviation de Input Output ConTroL, il s'agit d'une fonction permettant de manipuler des fichiers spéciaux

Chapitre 3

Adressage et routage

3.1 Adressage dans un réseau maillé

3.2 Routage

Plus précisément, le protocole HWMP dispose de deux modes de fonctionnement : “on demand” et “proactive tree building”. Le second nécessite qu’un noeud soit désigné comme noeud racine[10].

Avec le mode “on demand”, chaque fois qu’un noeud a besoin de connaître le chemin vers un autre noeud, il envoie un paquet “route request (RREQ)” en broadcast, en identifiant le noeud de destination. Le paquet RREQ contient aussi un champ métrique initialisé à 0. Chaque noeud intermédiaire va recevoir le paquet RREQ, éventuellement en plusieurs exemplaires. Si le paquet RREQ a une métrique plus faible que celle déjà connue, le noeud intermédiaire met à jour sa table de routage et le retransmet après avoir augmenté la métrique. Lorsque le paquet atteint le noeud de destination, ce dernier répond avec un paquet “Route Reply (RREP)” en unicast vers la source. Ainsi, tous les noeuds entre la destination et la source connaissent une route vers ces deux points [10].

Chapitre 4

Implémentation et architecture logicielle

Bibliographie

- [1] Michel Terré. Wifi, mars 2007.
<http://easytp.cnam.fr/terre/images/WiFi.pdf>.
- [2] Canaux et fréquences wifi 2.4 ghz et 5 ghz, janvier 2015.
<https://infos-reseau.com/canaux-et-frequences-wifi-2-4-ghz-et-5-ghz/>.
- [3] F.Nolot. Les réseaux sans-fil : Ieee 802.11.
<http://www.nolot.eu/Download/Cours/reseaux/m1info/ProtoAv-Cours7-Wifi.pdf>.
- [4] James F. Kurose ; Keith W. Ross. Ieee 802.11 lans, 1999.
https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/05.07.htm.
- [5] How 802.11 wireless works, Mars 2003.
[https://technet.microsoft.com/en-us/library/cc757419\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc757419(v=ws.10).aspx).
- [6] Nicolas Darchis. 802.11 frames : A starter guide to learn wireless sniffer traces, octobre 2010.
<http://www.nolot.eu/Download/Cours/reseaux/m1info/ProtoAv-Cours7-Wifi.pdf>.
- [7] Guido Hiertz ; Dee Denteneer ; Sebastian Max ; Rakesh Taori ; Javier Cardona ; Lars Berlemann ; Bernhard Walke. Ieee 802.11s : The wlan mesh standard, fevrier 2010.
<http://ieeexplore.ieee.org/document/5416357/>.
- [8] Jerome Henry. 802.11s mesh networking, novembre 2011.
https://www.cwnp.com/uploads/802-11s_mesh_networking_v1-0.pdf.
- [9] R. Droms. Dynamic host configuration protocol, mars 1997.
<https://tools.ietf.org/html/rfc2131>.
- [10] Avinash Joshi ; Hrishikesh Gossain ; Jorjeta Jetcheva ; Malik Audeh ; Michael Bahr ; Jan Kruys ; Azman-Osman Lim ; Shah Rahman ; Joseph Kim ; Steve Conner ; Guenael Strutt ; Hang Liu ; Susan Hares. Hwmp protocol specification, novembre 2006.
doc. : IEEE 802.11-06/1778r1.
- [11] Netlink protocol library suite (libnl).
<http://www.infradead.org/~tgr/libnl/>.