REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE

DEPARTEMENT INFORMATIQUE

Mémoire
en vue de l'Obtention du
Diplôme de Master en Informatique

*Cycle LMD*

Thème

# Réalisation d'un Hotspot wi-fi sous une Raspberry-Pi

**Proposé et dirigé par :**

**Promoteur: Mr M.DAOUI**

**Présenté par :**

**TIDAF Juba**

**HAFID Salim**

**Année : 2014/ 2015**

# Remerciements

On tien à témoigner notre reconnaissance à dieu tout puissant, qui nous a aidé et béni par sa volonté durant toute cette période.

On tien à remercier notre promoteur

Mr M.DAOUI  et le Co-promoteur Mr S.DJIOUA pour leurs orientation et leurs disponibilité constante tout au long de notre travail.

On remercie également les membres de jury qui ont accepté d'évaluer notre travail.

On remercie tous ceux qui ont contribué de prés ou de loin à la réalisation de ce projet.

*Juba*
*Salim*

# *DEDICACES*

*Je dédie ce modeste travail à :*

*A mes parents .Aucun hommage ne pourrait être à la hauteur de l'amour Dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.*

*A mon très chaire grand frère Ghani. Que dieu l'accueil dans son vaste paradis. Il restera toujours ma source d'inspiration et un modèle que je dois continuer suivre toute au long de ma vie.*

*A mes deux sœurs Sabrina et Nassima ainsi qu'à son mari Rabah et leurs deux filles, Dacine et Liliane, sans oublier le petit Yanal.*

*A toute ma famille, et mes amis,*

*A mon binôme Juba et toute la famille TIDAF.*

*Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

***Salim***

# *DEDICACES*

- *L'eternel, Source d'Intelligence et de Sagesse Infinie,*
- *Premièrement à la mémoire de ma très cher grand-mère « Taous » et « nana kaissa ».*
- *Ceux qui ont éclairé ma vie, ma très chère maman et mon père.*
- *Mes très chers frères Nordine lamia et son marie Ahcene et Nordine a qui je souhait la réussite.*
- *ma grande mère « Mekiossa » .*
- *Dada amrouche et sa famille*
- *khwali, mes tentes et leur famille.*
- *Hassane, hakim ,makho, et le petit Amine, jigo, kaissa, achour, louenes, ghania a qui je souhait la guérissant.*
- *A mon binôme Salim et toute la famille HAFID.*
- *Toutes mes connaissances et ami(e)s.*
- *Tous ceux qui m'ont aidé de près et de loin.*
- *Sans oublier mes camarades de la promotion 2014/2015*

*Juba*

# Table des Matièrs

# Table des matières

**CHAPITRE II : Présentation de la carte électronique Raspberry-PI.**

# Table des Figures

# Table des figures :

# Listes des Tableaux

# Liste des tableaux :

# Introduction générale

# INTRODUCTION GENERALE

Depuis quelque temps des bornes sans fil placées dans des endroits publics donnent un accès gratuit ou payant à Internet. Ces bornes sans fil ‘’Wifi’’, ou Hotspot, dont le but commercial est d’attirer une nouvelle clientèle « nomade » doivent être à la fois simple d’accès et sécurisées.

Les Hotspot se sont rapidement développés à l'échelle mondiale mais ce n’est pas le cas de l’Algérie. Ces Hotspot permettant ainsi à des utilisateurs nomades disposant d'équipements adaptés (ordinateurs ou téléphones portables compatibles, PDA et autres) de se connecter à Internet de partout avec beaucoup de simplicité.

Le système que nous allons développer vise à exploiter les nouvelles possibilités de Hotspot pour offrir aux différents acteurs commerciaux un moyen de faire de la publicité de leurs produits. Ceci est réalisé à travers un Hotspot configuré de manière à fournir gratuitement l’accès internet au public après avoir regardé une publicité pour une période donnée.

Malgré la simplicité apparente de cette application, elle fait intervenir en réalités des mécanismes lourds de gestion des accès réseaux comme le portail captif ou le serveur Radius. Ce travail devant être réalisé sur une carte embarquer Raspberry pi, nous avons construit un système autonome sur cette carte contenant ces différentes technologies (portail captif : coovachilli, serveur RADIUS, BDD MySQL).

Pour mener à bien notre projet nous avons d’abord procédé, dans le premier chapitre à une brève présentation des réseaux sans fil, puis nous sommes passés à l’étude du portail captif et enfin nous avons fait une étude approfondie sur le serveur d’authentification RADIUS

Dans le deuxième chapitre nous avons fait une brève présentation sur les systèmes embarqués, puis nous avons fait une étude sur la carte électronique «Raspberrry-Pi » que ce soit sur le plan matériel que logiciel. Dans la partie matérielle nous avons spécifiés les capacités de la carte (Processeur, Ram, les E/S, …etc) puis nous somme passés à la partie logicielle dans laquelle nous avons spécifiés les types de systèmes d’exploitations qui sont installables via une cartes mémoire bootable.

Dans le troisième chapitre nous avons fait une modélisation de notre système avec UML. Dans la première partie nous avons fait une analyse, en commençant par l’étude des cas d’utilisations et

de leurs scénarios ainsi que les besoins fonctionnels du système, puis dans la deuxième partie nous avons présenté notre conception.

Dans le quatrième chapitre nous avons présenté les étapes de configuration pour la mise en place de notre Hotspot. Nous avons commencé par l'identification des composants matériels et la topologie utilisée. Ensuite, nous avons présentés les différentes étapes de configurations des logiciels utilisés en commençant par le système d'exploitation «RASPBIAN » puis le portail captif Coovachilli en finissant par le serveur d'authentification freeradius et sa base de données Mysql. Cette dernière hébergera les utilisateurs. Enfin dans la derrière partie, nous avons présenté les résultats du test de fonctionnement de notre Hotspot.

# CHAPITRE N°1

## Introduction:

Selon les statistiques *d'Internet World Stats*, en 2008, l'Algérie compte 3 500 000 d'internautes. Ce chiffre a quasiment triplé en 2014 [1]. Cet engouement à l'utilisation des TIC impose une augmentation de l'offre des services Internet. En effet, bon nombre de cette population dispose aujourd'hui d'un appareil mobile (portable, PDA, Smartphone, ...) et souhaite pouvoir accéder à Internet dans la majorité des lieux qu'ils fréquentent. Dans cette optique, l'expansion très rapide des points d'accès sans-fil permet la connexion des appareils nomades. Néanmoins chaque réseau possède sa politique d'accès et ne souhaite pas laisser n'importe qui accéder aux ressources réseaux et plus particulièrement les ressources Internet qui sont très limitées.

Ainsi, il est nécessaire de mettre en place des systèmes d'authentification sur ces réseaux qui doivent cumuler de multiples avantages. Ces avantages sont entre autres : une compatibilité avec la majorité des appareils mobiles du marché, une sécurité des échanges entre les clients et le reste du réseau, une plus grande transparence aussi bien lors de la phase d'authentification que lors de l'utilisation du réseau, une réduction de l'impact au niveau des ressources matérielles et de la bande passante, etc.

Face à ces enjeux, le portail captif s'est imposé comme une solution fréquente pour les points d'accès payants ou non. Il peut se généraliser à tous les modes d'accès (sans-fil ou filaire) nécessitant un contrôle.

Les réseaux de ce genre disposent aujourd'hui d'un réseau informatique dont la gestion se complique avec la diversité et le nombre croissant des utilisateurs d'où la nécessité de mettre en place un portail captif avec un serveur d'authentification externe.

## I- Réseau sans fil :

### 1-Définition d'un réseau sans fil :

Comme son nom l'indique, c'est un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu, c'est la raison pour laquelle on entend parfois parler de mobilité.

Les réseaux sans fil sont basés sur une liaison utilisant des ondes radioélectriques à la place des câbles habituels. Il existe plusieurs technologies se distinguant d'une part par la fréquence d'émission utilisée et d'autre part, par le débit et la portée des transmissions.

Les réseaux sans fil permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres. De plus l'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes comme c'est le cas avec les réseaux filaires (creusement de tranchées pour acheminer les câbles, équipements des bâtiments en câblage, goulottes et connecteurs), ce qui a valu un développement rapide de ce type de technologies. **[G.P secEYR04]**

## 2- Classification des réseaux sans fils :

La figure suivante récapitule les déférents réseaux sans fil selon l'étendue :

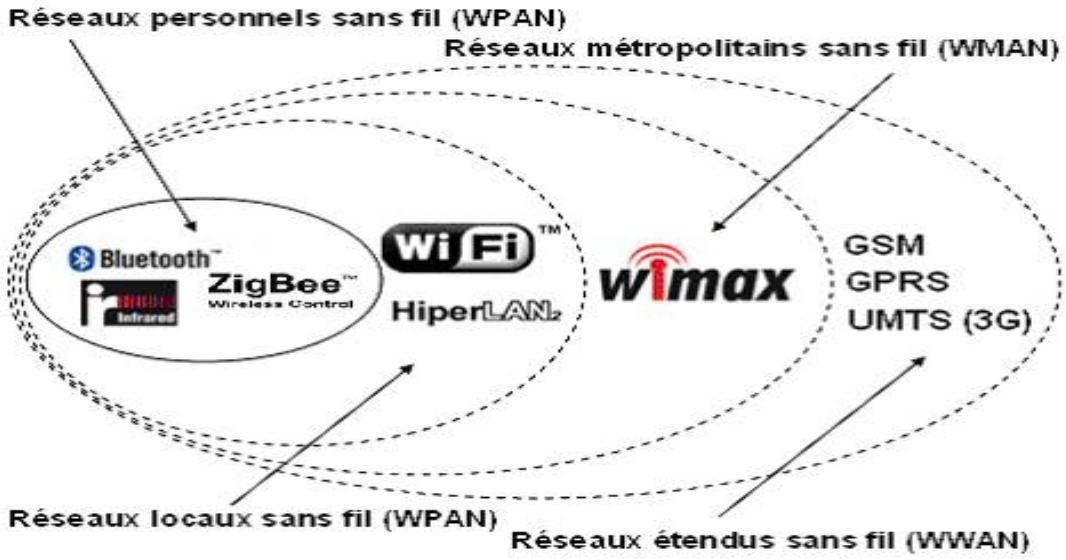



**Figure I.1:** Classification des réseaux sans fils selon l'étendue géographique **[B.M EIVD02]**

### 2-1 Réseaux personnels sans fils (WPAN) :

Le réseau personnel sans fils (appelé également réseau individuel sans fils ou réseau domotique sans fils et noté WPAN pour Wireless Personal Area Network) concerne les réseaux sans fils d'une faible portée : de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fils entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPAN :

- La principale technologie WPAN est la technologie Bluetooth, lancée par Ericsson en 1994, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Bluetooth, connue aussi sous le nom IEEE 802.15.1, possède l'avantage d'être très peu gourmand en énergie, ce qui le rend particulièrement adapté à une utilisation au sein de petits périphériques. La version 1.2 réduit notamment les interférences avec les réseaux Wi-Fi.
- HomeRF (Home Radio Frequency), lancée en 1998 par le HomeRF Working Group (formé notamment par les constructeurs Compaq, HP, Intel, Siemens, Motorola et Microsoft) propose un débit théorique de 10 Mbps avec une portée d'environ 50 à 100 mètres sans amplificateur. La norme HomeRF soutenue notamment par Intel, a été abandonnée en Janvier 2003, notamment car les fondeurs de processeurs misent désormais sur les technologies Wi-Fi embarquée (via la technologie Centrino, embarquant au sein d'un même composant un microprocesseur et un adaptateur Wi-Fi).
- La technologie ZigBee (aussi connue sous le nom IEEE 802.15.4) permet d'obtenir des liaisons sans fil à très bas prix et avec une très faible consommation d'énergie, ce qui la rend particulièrement adaptée pour être directement intégré dans de petits appareils électroniques (appareils électroménagers, hifi, jouets, ...).
- Enfin les liaisons infrarouges permettent de créer des liaisons sans fils de quelques mètres avec des débits pouvant monter à quelques mégabits par seconde. Cette technologie est

largement utilisée pour la domotique (télécommandes) mais souffre toutefois des perturbations dues aux interférences lumineuses.

| Technologie | Norme | Débit théorique | Portée(m) | Bande de Fréquence (GHz) | Observation |
|---|---|---|---|---|---|
| Bluetooth | IEEE 802.15.1 | 1 Mbits/s | Une trentaine | 2,4 -2,4835 | - Bas prix<br>- L'émission de puissance dépend de la réglementation |
| HomeRF | Consortium (Intel, HP, Siemens, Motorola et Compaq) | 10 Mbits/s | 50 | 2,4 – 2,4835 | - Permet de relier des PC portables, fixes et d'autres terminaux. |
| Zeegbie | IEEE 802.15.4 | 20 – 250 kbits/s | 100 | 2,4 – 2,4835 | - Très bas prix,<br>- Très faible consommation d'énergie. |
| Infrarouge | IrDA 1.1 | 1 Mb/s | 1-5 | | - Bas prix,<br>- Faible consommation d'énergie. |

**Tableau I.1:** Technologie des réseaux WPAN.

### 2-2- Réseaux locaux sans fils (WLAN) :

Le réseau local sans fils (WLAN pour Wireless Local Area Network) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre-eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes :

**Le WiFi** (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.

- **HiperLAN2** (High Performance Radio LAN 2.0), norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute), permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5150 et 5300 MHz.

- **DECT** (Digital Enhanced Cordless Telecommunication), norme des téléphones sans fils domestiques. Alcatel et Ascom développent pour les environnements industriels, telles les centrales nucléaires, une solution basée sur cette norme qui limite les interférences. Les points d'accès résistent à la poussière et à l'eau. Ils peuvent surveiller les systèmes de sécurité 24/24h et se connecter directement au réseau téléphonique pour avertir le responsable en cas de problème.

| **Technologie** | **Norme** | **Débit théorique** | **Portée(m)** | **Bande de Fréquence (GHz)** | **Observation** |
|---|---|---|---|---|---|
| WiFi | IEEE 802.11 | 2 - 54 | 35 -50 (indoor) des centaines (outdoor) | 2,4 – 2,48355 | Elle comporte plusieurs déclinaisons IEEE 802.11 a/b/g/n…etc |
| HiperLAN 1 | ETSI | 19 - 20 | 50 | 5 | - La vitesse de déplacement de l'utilisateur ne peut excéder 10 m/s<br>- Permet d'accéder aux réseaux ATM |
| HiperLAN 2 | | 25 | 200 | | |
| HiperLink | | 155 | 150 - 200 | 17,2 – 17,3 | Permet des liaisons fixes entre 2 points |
| DECT | | 2 | 300 | 1880 – 1900 MHz | Technique d'accès TDMA |

**Tableau 1.2:** Technologie des réseaux WLAN.

### 2-3- Réseaux métropolitains sans fils (WMAN) :

Le réseau métropolitain sans fils (WMAN pour Wireless Metropolitan Area Network) est connu sous le nom de Boucle Locale Radio (BLR). Les WMAN sont basés sur la norme IEEE 802.16. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication. Et ce sont des réseaux qui couvrent partiellement ou totalement la superficie d'une ville.

| Technologie | Norme | Débit théorique | Portée(m) | Bande de Fréquence (GHz) | Observation |
|---|---|---|---|---|---|
| WiMax | IEEE 802.16 | 70 | 50 | 1-66 | - Permet le raccordement des hotspots WiFi pour l'accès à Internet<br>- Techniques d'accès TDMA comporte plusieurs déclinaisons |
| HiperAccess | ETSI | 25 | 5 | 5 | - Permet d'accéder aux réseaux ATM |

**Tableau I.3:** Technologie des réseaux WMAN.

Le Wimax (standard de réseau sans fils poussé par Intel avec Nokia, Fujitsu et Prowim) est basé sur une bande de fréquence de 2 à 11 GHz, offrant un débit maximum de 70 Mbits/s sur 50km de portée, certains le placent en concurrent de l'UMTS, même si ce dernier est d'avantage destiné aux utilisateurs itinérants.

**2-4- Réseaux étendus sans fils (WWAN) :**

Le réseau étendu sans fils (WWAN pour Wireless Wide Area Network) est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fils les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fils. Les principales technologies sont les suivantes :

- **GSM** (Global System for Mobile Communication ou Groupe Spécial Mobile)
- **GPRS** (General Packet Radio Service)
- **UMTS** (Universal Mobile Telecommunication System)

| Technologie | Norme | Débit théorique | Portée(m) | Bande de Fréquence (GHz) | Observation |
|---|---|---|---|---|---|
| GSM | Européenne | 9.6 Kbits/s | 0.3 – 30 | [890-915] MHz [935-960] MHz [1710-1785] MHz [1805-1880] MHz | - Utilise une commutation de circuits Système très sécurisé |
| GPRS | Européenne | ≤ 12kbits/s | 0.3 – 30 | [890-915] MHz [935-960] MHz [1710-1785]MHz [1805 :1880]MHz | - Utilise une commutation de paquets<br>- Prise en charge des applications de données à moyens débits<br>- Utilise le protocole IP pour le formatage des données |
| UMTS | Européenne (ETSI) | ≤ 2 Mbits/s | 0.3 – 30 | 2 GHz | - Offre un accès à Internet et à ses serveurs web<br>- Supporte des applications audio et vidéo basse définition<br>- Fonctionne en mode paquet et mode circuit |
| CDMA 2000 | Américaine (TIA) | ≤ 2 Mbits/s | | 2 GHz | -Utilise la technique d'étalement de bande |
| EDGE | Européenne | 59.2 kbits/s | 0.3 – 30 | 2 GHz | - Utilise la commutation de circuit |
| IS 95 | Américaine | 1,2288 Mchips/s | | 800-900 MHz 1800-1900 MHz | - Utilise la technologie CDMA |

**Tableau I.4:** Technologie des réseaux WWAN.

### 3- Standard IEEE 802.11 :

L'IEEE a développé la norme 802.11 sous plusieurs versions regroupant ainsi les normes physiques suivies des normes d'amélioration. Elles offrent chacune des caractéristiques différentes en termes de fréquence, de débit ou de portée du signal.

### 3-1- Les normes physiques :

La première version normalisée par l'IEEE fût la 802.11. Elle utilisait la modulation DSSS sur la bande 2.4 GHz. Cette norme n'était pas compatible entre constructeurs. De plus, elle offrait un débit très faible (2 Mbps), comparés aux débits que proposait la norme Ethernet filaire. L'IEEE développa de nouvelles générations de réseaux sans fil : la 802.11b, la 802.11a et la 802.11g.

#### a. La 802.11b ou Wi-Fi 2 :

C'est la première norme Wi-Fi interopérable. Avec un débit de 11 Mbps, elle permet une portée de 300 mètres dans un environnement dégagé. Elle utilise la bande des 2.4GHz avec 3 canaux radios disponibles. Cette norme Wi-Fi a connu beaucoup d'extensions et chacune d'entre elles, visant à apporter une amélioration soit au niveau du débit, soit au niveau de la bande passante ou même de la sécurité, de la qualité de service ou de la capacité du canal etc. [14]

#### b. La 802.11 a :

Encore appelé Wi-Fi 5, cette norme permet d'obtenir du haut débit (54 Mbit/s) tout en spécifiant 8 canaux. Mais elle n'est pas compatible avec la 802.11b. Elle utilise la technique de modulation OFDM.

#### c. La 802.11g :

La 802.11a offre un débit assez élevé mais la portée est plus faible et son usage en extérieur est souvent interdit. Pour répondre à ces problèmes, l'IEEE développe la nouvelle norme 802.11g, offrant le même débit que le Wi-Fi 5, tout en restant compatible avec le Wi-Fi 2 (bande de fréquences de 2.4 GHz) .Cette norme vise aussi à remplacer Wi-Fi 2 sur la bande 2.4 GHz mais avec un débit plus élevé pouvant atteindre les 54 Mbits/s. Elle utilise la technique de modulation OFDM.

| Norme | Normalisation | Bande Ghz | Débit Théorique (Mbits/s) | Débit Réel (Mbits/s) | Portée Théorique | Observations |
|---|---|---|---|---|---|---|
| 802.1 | 1997 | 2.4 | 2 | <1 | 100 m | Utilisateurs particulier |
| 802.11a | 1999 | 5 | 54 | 2-24 | 20 m | Usage extérieur interdit-en France |
| 802.11b | 1999 | 2.4 | 11 | 4-6 | 60 m | Compatible 802.11 |
| 802.11g | 2003 | 2.4 | 54 | 20-28 | 20 m | Compatible 802.11b |
| 802.11n | 2009 | 2.4/5 | 450 | 200 | 50/125 m | Compatible 802.11a/b/g |

**Tableau I.5:** Comparaison des principales normes 802.11.

## 4- Communication entre équipements :

L'architecture d'un réseau Wi-Fi est basée sur un système cellulaire. Il existe deux principaux modes de fonctionnement.

### 4-1- Le mode ad hoc :

En mode ad hoc, il n'y a aucune administration centralisée. Il n'existe pas de point d'accès. Les stations terminales communiquent directement entre elles selon des liaisons point à point ou point multi point. Ces stations forment une cellule appelée IBSS (Idependant Basic Service Set). [13]

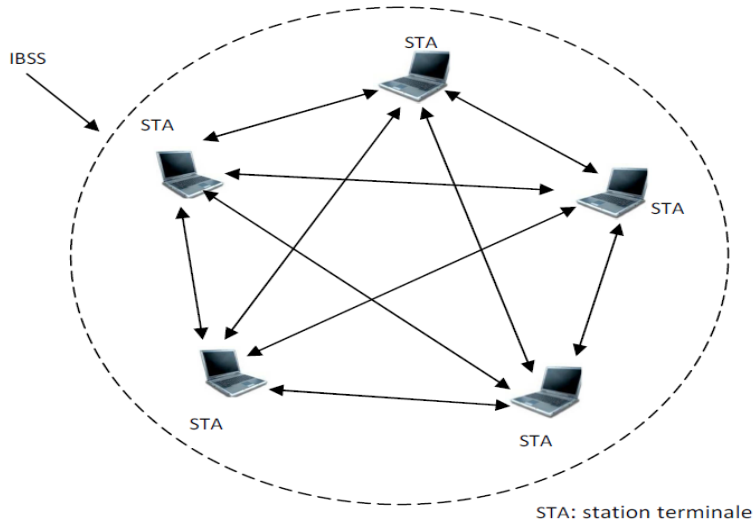



**Figure I.2:** Mode ad hoc.

### 4-2- Le mode infrastructure :

Dans ce mode, une station de base appelée Access Point (point d'accès) gère toutes les stations terminales à portée radio. Il permet aux stations terminales de communiquer entre elles et avec des stations d'un réseau filaire existant. L'ensemble constitué par le point d'accès et les stations sous son contrôle forme un BSS (Basic Service Set/Ensemble de services de base), la zone ainsi couverte est appelée BSA (Base Set Area). [13]

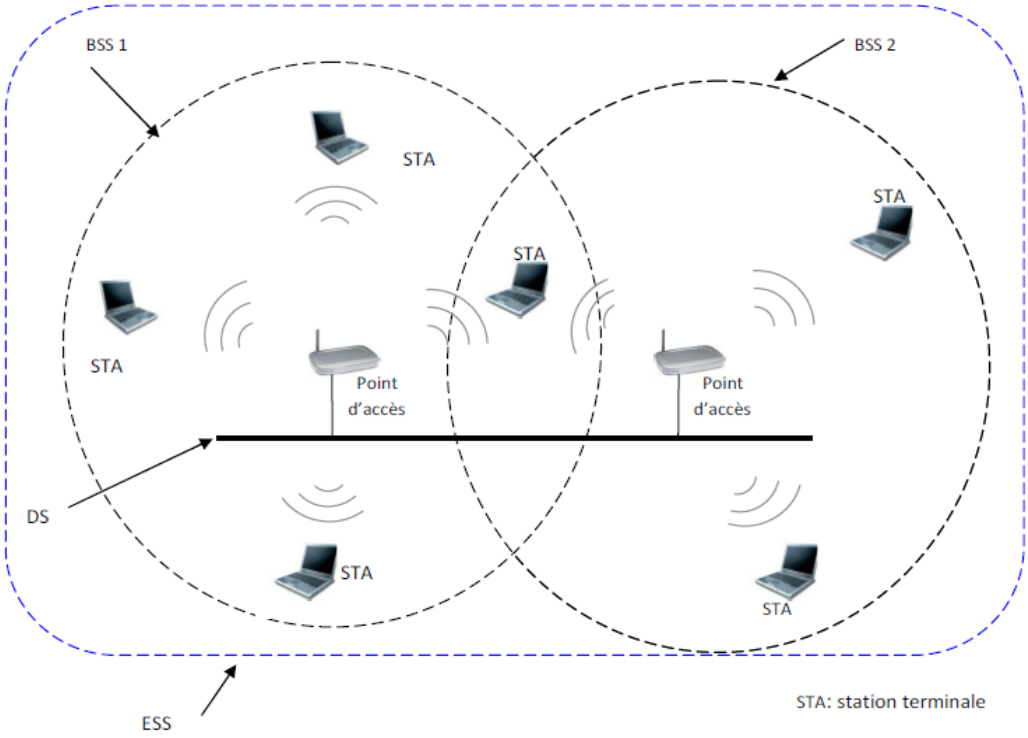



**Figure I.3:** Mode infrastructure.

Le BSS est identifié par un BSSID qui est généralement l'adresse MAC du point d'accès. Un ensemble de BSS forme un ESS (Extended Service Set). Les BSS (plus précisément leurs points d'accès) sont interconnectés via un DS (distribution system/système de distribution). Le système de distribution ou backbone est implémenté indépendamment de la partie sans fil, c'est généralement un réseau Ethernet, mais il peut aussi être un réseau Token Ring, FDDI ou un autre réseau local sans fil. Cette architecture permet aussi d'offrir aux usagers mobiles l'accès à d'autres ressources (serveurs de fichier, imprimante, etc.) ou d'autres réseaux (Internet).L'ESS est identifié par un ESSID communément appelé SSID qui constitue le nom du réseau. Le SSID est un premier niveau de sécurité, vu que la station doit connaître ce SSID pour pouvoir se connecter au réseau.

## II-Portail captif :

### 1-Définition d'un portail captif :

Un portail captif est une application qui permet de gérer l'authentification des utilisateurs d'un réseau local qui souhaitent accéder à un réseau externe (généralement Internet)[2]. Il oblige les utilisateurs du réseau local à s'authentifier avant d'accéder au réseau externe. Lorsqu'un utilisateur cherche à accéder à Internet pour la première fois, le portail capte sa demande de connexion grâce à un routage interne et lui propose de s'identifier afin de pouvoir recevoir son accès. Cette demande

d'authentification se fait via une page web stockée localement sur le portail captif grâce au serveur HTTP. Ceci permet à tout ordinateur équipé d'un navigateur web et d'un accès Wifi de se voir proposer un accès à Internet. Au-delà de l'authentification, les portails captifs permettent d'offrir différentes classes de services et tarifications associées pour l'accès Internet (Par exemple: Wifi gratuit, filaire payant, 1 heure gratuite,...). Cela est obtenu en interceptant tous les paquets quelles que soient leurs destinations jusqu'à ce que l'utilisateur ouvre son navigateur web et essaie d'accéder à Internet. Lors de l'établissement de la connexion, aucune sécurité n'est activée. Cette sécurité ne sera active que lorsque l'ordinateur connecté tentera d'accéder à Internet avec son navigateur web. Le portail captif va, dès la première requête HTTP, rediriger le navigateur web afin d'authentifier l'utilisateur, sans quoi aucune demande ne passera au-delà du serveur captif. Une fois l'utilisateur authentifié, les règles du firewall le concernant sont modifiées et celui-ci se voit autorisé à utiliser son accès Internet pour une durée fixée par l'administrateur. A la fin de la durée fixée, l'utilisateur se verra redemander ses identifiants de connexions afin d'ouvrir une nouvelle session.

Ce système offre donc une sécurité du réseau mis à disposition, il permet de respecter la politique de filtrage web de l'entreprise grâce à un module proxy et permet aussi grâce à un firewall intégré d'interdire l'accès aux protocoles souhaités.

## 2-Fonctionnement général d'un portail captif :

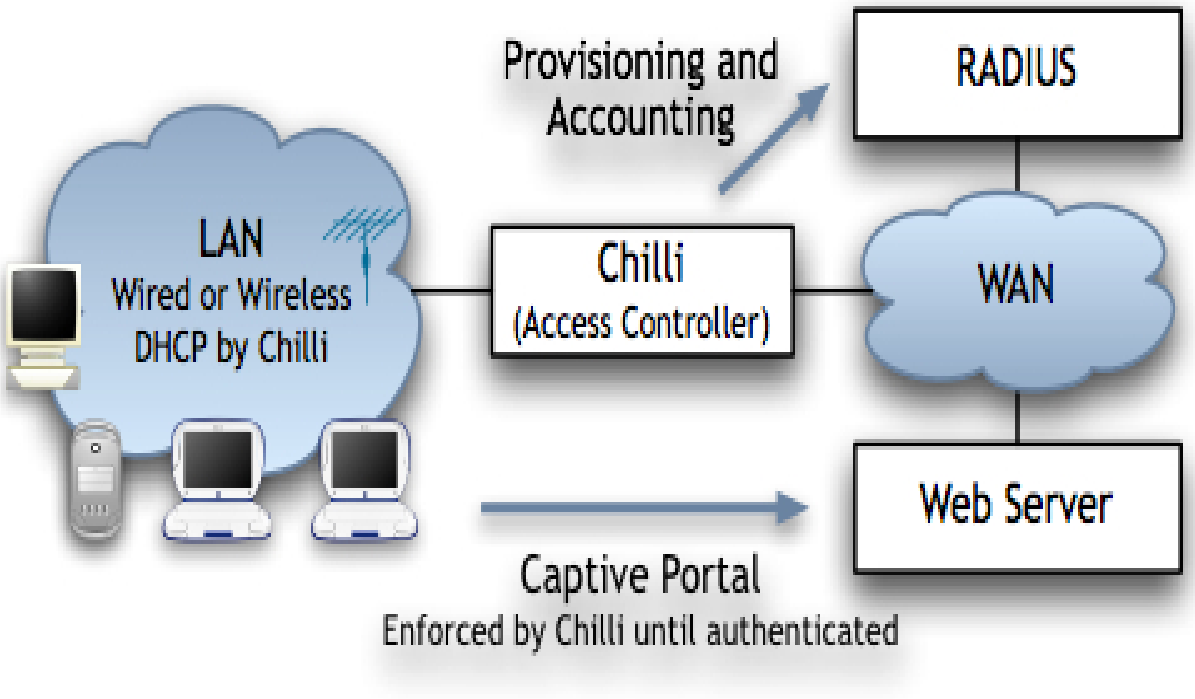



**Figure I.4:** Fonctionnement général d'un portail captif. [4]

La figure I.4 donne le schéma d'architecture d'un portail captif. Le client se connecte au réseau par l'intermédiaire d'une connexion filaire ou d'un point d'accès wifi. Ensuite un serveur DHCP lui fournit une adresse IP ainsi que les paramètres de la configuration du réseau. A ce moment-là, le client a juste accès au réseau entre lui et la passerelle. Cette dernière lui interdisant, pour l'instant, l'accès au reste du réseau. Lorsque le client va effectuer sa première requête de type web en HTTP ou HTTPS, la passerelle le redirige vers une page web d'authentification qui lui permet de s'authentifier grâce à un login et un mot de passe. Cette page est cryptée à l'aide du protocole SSL pour sécuriser le transfert du login et du mot de passe. Le système d'authentification va alors contacter une base de données contenant la liste des utilisateurs autorisés à accéder au réseau. Enfin le système d'authentification indique, plus ou moins directement selon les portails captifs, à la passerelle que le couple MAC/IP du client est authentifié sur le réseau.

Finalement le client est redirigé vers la page Web qu'il a demandé initialement, le réseau derrière la passerelle lui est dorénavant accessible. Le portail captif, grâce à divers mécanismes comme une fenêtre pop-up rafraîchie à intervalles réguliers ou des requêtes ping vers le client, est en mesure de savoir si l'utilisateur est toujours connecté au réseau. Au bout d'un délai d'absence sur le réseau, le portail captif va couper l'accès à cet utilisateur.

## 3- Aperçu des principaux portails captifs :

Toutes les solutions que nous avons étudiées sont des solutions libres et gratuites.

### 3-1 Coova-chilli :

Coovachilli est un applicatif open-source dédié à la gestion de l'authentification sur les réseaux, FORK du projet très populaire ChilliSpot (mais aujourd'hui disparue). Coova-chilli est très riche en fonctionnalités portail captif/walled garden serveur proxy et DHCP...etc. Cet applicatif s'installe via un package applicatif, les configurations se font via une interface de gestion sécurisée (HTTPS) ou bien en ligne de commande directement.

Coova-chilli a trois principales interfaces : une interface de liaison descendante pour accepter des clients, une interface de radius pour l'authentification des clients et une interface réseau en liaison montante pour transmettre le trafic vers d'autres. Une documentation très complète est disponible sur Internet, ainsi qu'une communauté très active. Coova-chilli assure une compatibilité multi-plates-formes, une personnalisation complète des pages accessibles aux utilisateurs ainsi qu'une simplicité d'utilisation grâce à une page de connexion très simple.

### 3.2 PFsense :

PFsense est une distribution FreeBSD développée en 2004. L'objectif de départ est d'assurer les fonctions de pare-feu et de routeur mais l'engouement généré par cet applicatif lui a permis d'étendre ses fonctionnalités et présente maintenant les fonctions de portail captif, serveur proxy, DHCP ...

Son installation se fait facilement via une distribution dédiée et toutes les configurations peuvent se faire soit en ligne de commande (SSH) ou via l'interface web (HTTPS). La sauvegarde et la restauration de configuration est disponible à travers l'interface web et permet de générer un simple fichier d'une taille raisonnable. Le portail assure une évolution constante grâce à des mises à jour régulières. L'installation est gérée automatiquement dans une partie du panneau d'administration. Cette solution permet une authentification sécurisée via le protocole HTTPS et un couple utilisateur / mot de passe. Une documentation très complète est disponible sur Internet, un support commercial est désormais présent en cas de gros incident. PFsense dispose aussi d'une communauté très active.

PFsense assure une compatibilité multi-plates-formes, une personnalisation complète des pages accessibles aux utilisateurs ainsi qu'une simplicité d'utilisation grâce à une page de connexion succincte où on ne retrouve que deux champs (utilisateur / mot de passe).

### 3.3 ALCASAR :

ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau) est un projet français essentiellement dédié aux fonctions de portail captif.

Cet applicatif s'installe via un script supporté par la distribution Linux Mandriva, les configurations se font via une interface de gestion sécurisée (HTTPS) ou bien en ligne de commande directement sur le Serveur Mandriva. Une sauvegarde de la configuration est prise en charge via la création d'un ghost système (fichier système) dans le panneau d'administration, ce qui engendre tout de même un fichier d'une certaine taille. Les mises à jour régulières assurent la pérennité de la solution.

L'authentification au portail est sécurisée par HTTPS et un couple utilisateur / mot de passe. Une documentation assez complète est disponible pour l'installation et la. Tout comme Coova-chilli et PFsense, ALCASAR est compatible avec de nombreuses plates-formes, la personnalisation des pages utilisateurs et la simplicité d'utilisation sont présentes.

### 3.4 ZeroShell :

ZeroShell est une distribution Linux conçue pour mettre en place une sécurité globale au sein d'un réseau (Pare-feu, VPN, portail captif...). Son installation est simple via une distribution dédiée. Elle présente une interface de gestion web simple d'utilisation qui permet entre autres de sauvegarder la configuration du portail captif ou encore de personnaliser les pages de connexion et déconnexion dans un éditeur HTML intégré.

Comme les trois autres solutions la page d'authentification est sécurisée et la connexion se fait via un couple utilisateur / mot de passe. On retrouve assez peu de documentation pour la gestion du système mais la communauté est bien présente. Son utilisation reste identique aux autres solutions présentées.

## 4-Comparaison des portails captifs :

Dans l'étude comparative des solutions nous avons mis en évidence plusieurs critères importants que doivent prendre en compte les différentes solutions:

- **Sécurité des échanges lors de l'authentification:** pour éviter la récupération de mot de passe sur le réseau.
- **Présence d'une documentation complète:** pour assurer la rapidité de mise en place de la solution.
- **Simplicité d'administration:** pour permettre à différentes personnes d'administrer le logiciel.
- **Simplicité d'utilisation:** pour permettre à tous les visiteurs (expérimentés ou non) de se connecter au réseau Wi-Fi ou filaire.
- **Compatibilité multiplate-formes** : pour permettre la connexion depuis des Smartphones, des navigateurs web et des différents systèmes d'exploitation.
- **Présence de sauvegarde et restauration de configuration** : pour permettre un redémarrage du système très rapidement en cas de problèmes.
- **Pérennité de la solution** : pour pallier les failles de sécurité et augmenter les fonctionnalités de la solution via des mises à jour.
- **Possibilité de personnaliser la page de connexion** : pour adapter le logiciel à la charte graphique de l'entreprise et ainsi le rendre plus convivial.

Le tableau suivant fait un récapitulatif des critères de comparaison :

| Critères | Solutions | | | |
|---|---|---|---|---|
| | **Coova-chilli** | **PFsense** | **ALCASAR** | **ZeroShell** |
| **Sécurité Authentification** | HTTPS | HTTPS | HTTPS | HTTPS |
| **Documentation** | ● | ● | ❖ | ■ |
| **Plates-formes Clientes Supportées** | Toutes | Toutes | Toutes | Toutes |
| **Personnalisation** | ● | ● | ❖ | ■ |
| **Facilité d'administration** | Installation via un paquet | Installation via une distribution dédiée | Installation via un script automatisé | Installation via une distribution dédiée |
| **Facilité d'Utilisation** | ● | ● | ❖ | ■ |
| **Sauvegarde/Restauration et Configuration** | ● | ● | ● | ■ |
| **Pérennité de la solution** | ● | ● | ● | ■ |

**Légende:**

- ● Disponibilité élevée.
- ❖ Moyennement disponible.
- ■ Moins disponible.

**Tableau I.6:** Comparaison entre les portails captifs.

## 5- Choix d'une solution de portail captif :

Bien que nous n'ayons pas mis en pratique toutes ces solutions pour les comparer, l'étude théorique permet de retenir les deux premières solutions à savoir Coova-chilli et PFsense car elles répondent toutes deux à nos besoins: solutions libres, peuvent s'installer sur un serveur comme sur un poste de travail, authentification des utilisateurs par login et mot de passe, contrôle de la bande passante, facilité d'administration, d'installation et de configuration, facilité d'utilisation, documentation très détaillée et disponible, disponibilité de mises à jour, etc.

Les deux solutions répondent tout à fait au cas étudié mais PFsense s'installe uniquement via une distribution FreeBSD. Par contre Coova-chilli s'installe via une méthode très simple en ligne de commande, ce qui rend impératif le choix de Coova-chilli. De plus Coova-chilli présente un fichier

de configuration très simple où l'on retrouve toutes les informations essentielles et que l'on peut modifier en fonction des besoins. Ce produit présente aussi une plus grande assurance car la communauté des utilisateurs est très active.

## 6. Présentation détaillée de coova-chilli

C'est un applicatif qui fait office de routeur/firewall open-source. Il est une reprise du projet ChilliSpot auquel il rajoute ses propres fonctionnalités. Coova est très réputé pour sa fiabilité. C'est un soft qui peut-être installé sur un simple poste de travail, un serveur ou même sur un boîtier en version embarqué. En plus, Coova s'adapte à tous les systèmes d'exploitation open-source.

Ce qui séduit chez Coova est sa facilité d'installation et de configuration des outils d'administration réseau. En effet, après une installation en mode console, il s'administre facilement depuis un fichier et gère nativement les VLAN (820.1q).

La distribution Coova met aussi à la disposition de l'administrateur réseau une multitude d'outils open-source et des services permettant d'optimiser ses tâches. Parmi ces services, figure **Captive Portal (**Portail Captif**)**.

### 6.1 Aperçu sur les fonctionnalités et services de Coova-chilli :

En fonction de la version du logiciel, le nombre de services et/ou de fonctionnalités peut varier. Pour notre projet, la version 1.3.0 est utilisée. Ainsi cette version dispose entre autres de :

- Portail captif.
- Support des VLAN tagués, c'est-à-dire qu'elle permet de créer et gérer nativement les VLAN.
- Routage IVP4 et IPV6.
- NAT (Network Adress Translation)
- Filtrage du trafic entrant et sortant pout tout type de trafic (ICMP, UDP, TCP) faire office de pare-feu.
- limitation des connexions pour empêcher un utilisateur de se connecter plusieurs fois avec son seul compte
- « Load Blancing » pour la répartition de charge en cas de surcharge.
- « Failover» pour le balancement d'une ligne à une autre si l'on possède par exemple plusieurs abonnements à Internet.
- Proxy transparent qui joue le rôle de serveur mandataire.
- DNS dynamique pour la gestion dynamique des noms de domaines.
- Serveur DHCP.
- Contrôle d'accès par adresses MAC ou authentification RADIUS.
- Serveur ou relay DHCP / DNS qui est relais du serveur DHCP / DNS.
- etc.

## 7-Les versions du logiciel : [4]

Depuis sa mise en route en 2006, le projet Coova-chilli ne cesse d'évoluer et différentes versions du logiciel se sont succédées. Pour chaque version, il en existe pour les architectures i386 (32bits) et amd64 (64-bits). De même elles sont disponibles pour les plateformes embarquées. Ainsi on a :

- CoovaChilli-v1.3.0
- CoovaChilli-v1.2.9
- CoovaChilli-v1.2.6
- CoovaChilli-1.0.1

## III- Serveur d'authentification RADIUS :

Afin de réaliser notre Hotspot, nous avons choisi une authentification par serveur. En effet ce type d'authentification est le plus sécurisé et permet une gestion simple des accès. L'authentification par serveur se fait via le protocole RADIUS.


**Figure I.5:** Logo RADIUS. [10]

### 1- RADIUS (Remote Authentification Dial-ln User Service) :

RADIUS est un acronyme pour (Remote Authentication Dial-In User Service), il permet la gestion des accès modems pour un grand nombre d'utilisateurs. Puisque les accès modem sont par définition un lien vers le monde extérieur, ils exigent une attention particulière à la sécurité, à l'autorisation et à la comptabilité.

Ceci peut mieux être réalisé en contrôlant une base de données contenant les détails des comptes utilisateurs (login, mot de passe, ….) aussi bien que l'information de configuration détaillant le type de service à délivrer à l'utilisateur (par exemple, Slip, PPP, telnet, rlogin…).

Radius permet le respect des trois A : « Authentification, Authorization and Accounting » (AAA) ou Authentification, Autorisation et Comptabilisation.

- Authentification : l'authentification consiste à vérifier qu'une personne/équipement est bien celle qu'elle prétend être. Ceci est généralement réalisé en utilisant un secret partagé entre l'utilisateur et le serveur mère AAAH ou à l'aide de certificats.
- Autorisation : l'autorisation consiste à permettre l'accès à certains services ou ressources. Un utilisateur peut par exemple demander à avoir une certaine bande passante. Le serveur AAA lui autorisera ou non cette demande.
- Comptabilisation : le serveur AAA a la possibilité de collecter des informations sur l'utilisation des ressources. Ceci permet à un opérateur de facturer un utilisateur suivant sa consommation.

Le protocole Radius a été développé à l'origine par Livingston, entreprise de serveur d'accès réseau (Network Access Server) PortMaster (Serveurs de modems). Radius est aujourd'hui une norme de l'IETF (Internet Engineering Task Force) qui est suivie par les principaux fournisseurs d'équipements réseau comme Cisco ou Lucent.

## 2- Les types de paquets RADIUS :

Le protocole RADIUS utilise 4 types de paquets différents pour assurer l'authentification :

- Le paquet Acces-Request qui est émis par le NAS vers le serveur RADIUS pour initialiser la conversation. Il contient l'attribut User-Name et d'autres attributs comme Nas-Identifier

- Le paquet Access-Accept qui est envoyé par le serveur RADIUS au NAS lorsque sa demande d'authentification a été validée.

- Le paquet Access-Reject est envoyé par le serveur RADIUS au NAS lorsque l'authentification échoue.

- Le paquet Access-Challenge qui est envoyé par le serveur après réception d'un Access-Request venant du NAS. Il a pour but de demander d'autres informations au NAS et de provoquer l'envoi d'un nouveau Access-Request. Il est toujours utilisé avec le protocole 802.1X EAP pour demander le mot de passe ou le certificat.

### 2-1 Format des paquets :

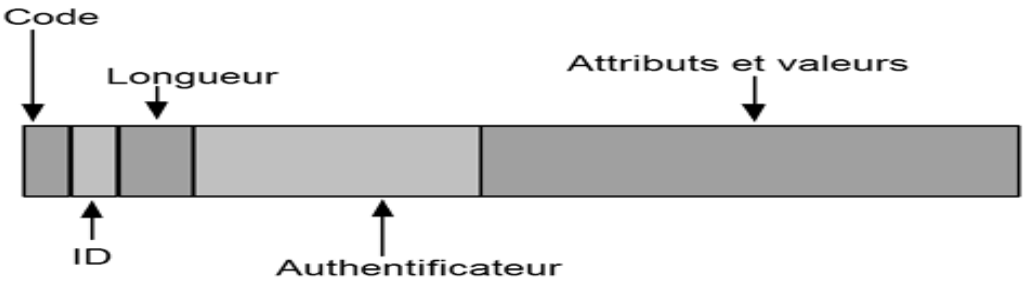



**Figure I-6:** Format des paquets Radius. **[radEYR07]**

**Code :**

Ce champ d'un seul octet contient une valeur qui identifie le type du paquet :

- ✓ Access-Request (code=1).
- ✓ Access-Accept (code=2).
- ✓ Access-Reject (code=3).
- ✓ Access-Challenge (code=11).

**ID :**

Ce champ, d'un seul octet, contient une valeur permettant au client Radius d'associer les requêtes et les réponses.

**Longueur :**

Champ de seize, octets contenant la longueur totale du paquet.

**Authentificateur :**

Lorsque le client NAS envoi un paquet access-request, il inclut un authentificateur appelé **request-authenticator** qui est une séquence aléatoire. Le serveur répond par un paquet access-accept ou access reject ou accept-challenge avec une **response-authenticator** composé avec les informations contenues dans le paquet access-request, le request authenticator et un secret partagé avec le NAS et le tout crypté en MD5. Le NAS est alors en mesure de vérifier que le serveur qui répond est bien celui qu'il a contacté.

**Attributs et valeurs :**

Ce champ du paquet est de longueur variable et contient la charge utile du protocole, c'est-à-dire les attributs et leur valeur qui seront envoyés soit par le NAS en requête, soit par le serveur en réponse.

**Les attributs :**

Les transactions RADIUS ont pour but de véhiculer des attributs et leur valeur entre le client NAS et le serveur. Ces attributs et leur valeur sont appelés paires attribut-valeur (AVP= attribut-value pair) Ces attributs permettent au client de communiquer des informations au serveur (password, MAC adresse…) et au serveur de communiquer les paramètres des autorisations qu'il délivre (vlan…) ou bien demander des informations complémentaires. La valeur d'un attribut peut correspondre à l'un des types suivants :

- ✓ adresse IP (4 octets).
- ✓ date (4 octets).
- ✓ chaîne de caractères (jusqu'à 255 octets).
- ✓ entier (4 octets).
- ✓ valeur binaire (1 bit).
- ✓ valeur parmi une liste de valeurs (4 octets).

Le champ Attributs et valeurs peut contenir plusieurs couples attribut-valeur suivant le format de la figure suivante :

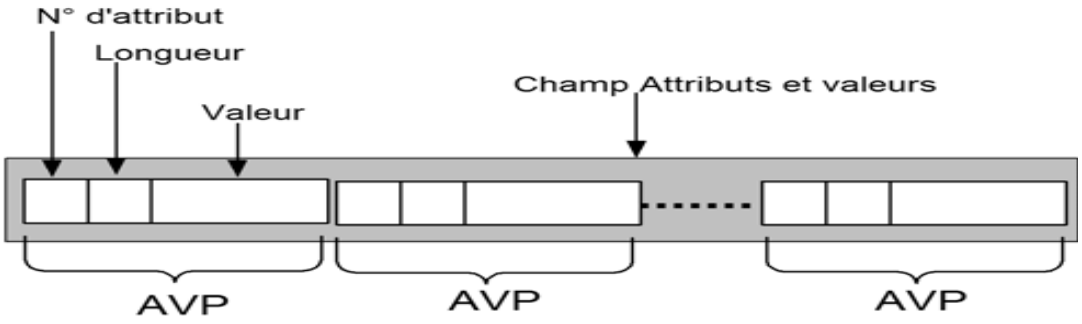



**Figure I.7:** Format du champ Attributs et valeurs (AVP). **[radEYR07]**

Le nom de l'attribut n'est jamais présent dans les paquets. Seul son numéro apparaît. La correspondance entre un numéro d'attribut et son nom sera faite grâce à un dictionnaire.

Il y a beaucoup d'attributs standards mais peu sont utilisables dans le cas d'une utilisation avec 802.1x. Par exemple l'attribut CALLBACK-NUMBER contient le numéro de téléphone sur lequel il faut rappeler le client. Ce qui est inutile dans notre cas...

On décrits ici uniquement les attributs intéressants pour l'authentification 802.1X et Radius Mac.

- ✓ User-Name envoyé par le NAS, il contient l'identifiant qui va servir de point d'entrée dans la base du serveur d'authentification
- ✓ User-Password, mot de passe associé au User-Name et envoyé par le NAS
- ✓ Nas-IP-Address qui contient l'adresse IP du NAS qui communique avec le serveur.
- ✓ Nas-Port qui est le numéro de port du NAS sur lequel le poste utilisateur est connecté.
- ✓ Called-Station-Id qui est l'adresse MAC du NAS. Celui-ci l'envoie au serveur afin d'authentifier chaque station utilisateur en fonction du matériel auquel il est connecté.
- ✓ Calling-Station-Id qui est l'adresse MAC de la station utilisateur qui se connecte au réseau.
- ✓ Les attributs « vendors » sont des fonctionnalités supplémentaires qui permettent de gérer les fonctions supplémentaires qui ne font pas partie des standards de RADIUS.

**Les attributs « vendor » :**

Les fabricants de matériel réseau (NAS) ont parfois intégré à leurs équipements des attributs spécifiques en plus des attributs standards définis dans le RFC. Ces attributs sont encapsulés dans l'attribut standard vendor-specific qui a pour numero 26. Ils sont appelés **VSA = Vendor Specific Attribut.**

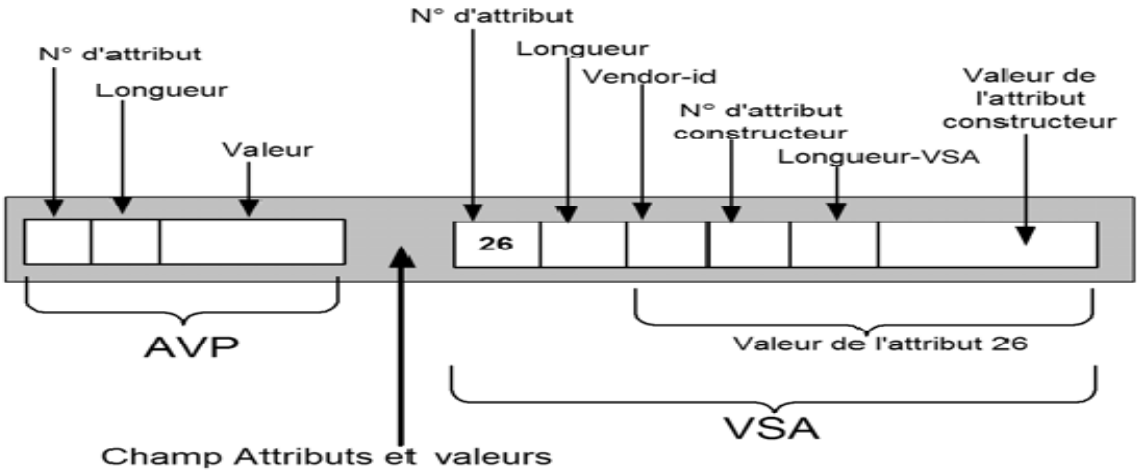



**Figure I.8:** Format des attributs « vendor ». **[radEYR07]**

**N° d'attribut :** il s'agit toujours de 26 pour indiquer qu'il s'agit d'un VSA.
**Longueur :** la longueur totale de l'attribut.

**Vendor-id :** le code international du constructeur tel que défini dans la RFC 1700 (Assigned Numbers).
**N° d'attribut constructeur :** il s'agit d'un numéro d'attribut défini par le constructeur.
**Longueur-VSA :** c'est la longueur de la valeur du champ VSA.
**Valeur de l'attribut spécifique :** c'est la valeur de l'attribut. Bien entendu, le sens de cette valeur n'est compréhensible que par le matériel du constructeur.
Comme pour les attributs standards, il devra exister un dictionnaire par constructeur.
**Dictionnaires d'attributs :**

Chaque attribut possède un numéro d'identification. Seul ce numéro est transmis dans les paquets. La correspondance entre le nom de l'attribut, son numéro et son type sont réalisés dans un dictionnaire.

## 3- Protocoles d'authentification Radius :

Deux possibilités :

- ✓Avec une authentification grâce à l'adresse MAC du poste de travail (Radius-MAC).
- ✓Avec les protocoles IEEE 802.1X et EAP (de l'anglais Extensible Authentication Protocol) qui permettront de pousser plus loin les possibilités et la sécurité des méthodes d'authentification.

### 3-1 Principe de l'authentification Radius-MAC :

L'authentification par adresse MAC (Figure I.9), appelée Radius-MAC (ou MAC-based), est la plus simple à mettre en œuvre. En revanche, c'est la moins sûre.

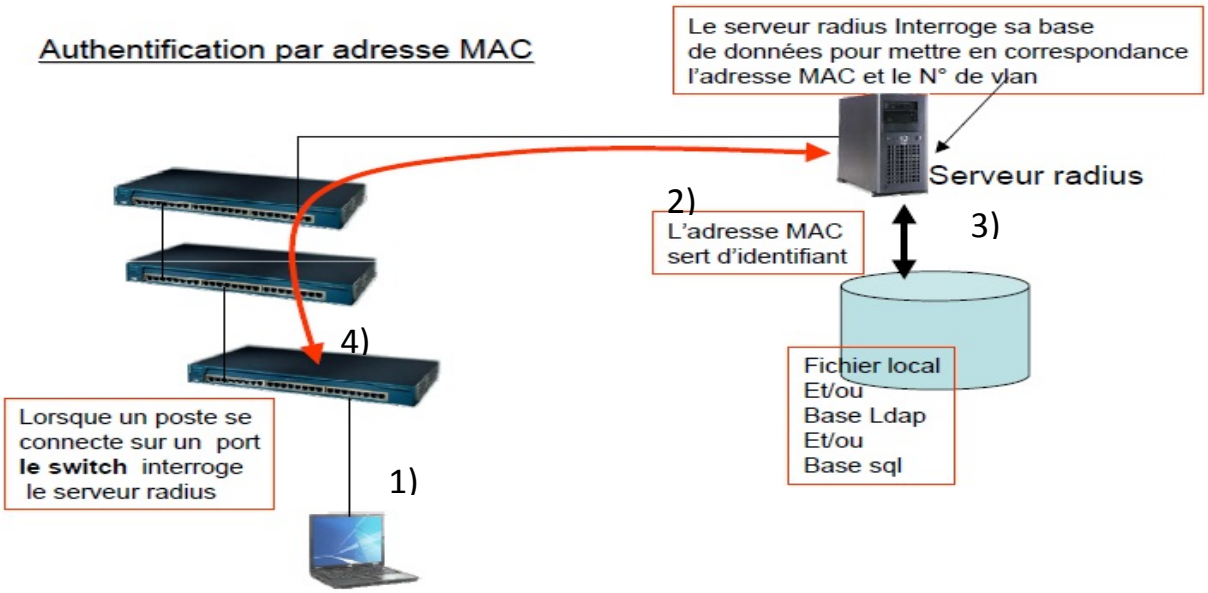



**Figure I.9:** Principe de l'authentification Radius-MAC **[PROT S.B]**

**1)-** Le poste de travail se branche sur un des ports du commutateur.

**2)-** Le commutateur détecte cette connexion et envoie une requête d'authentification (Access-Request) au serveur Radius. Dans cette requête, l'adresse MAC du poste de travail fait office d'identifiant.
**3)-** Le serveur reçoit ce paquet et utilise l'adresse MAC comme point d'entrée dans sa base de données d'où il récupère le VLAN associé si l'adresse MAC est connue.
**4)** Le serveur envoie sa réponse au commutateur. Si elle est négative (Access-Reject), le port du commutateur reste fermé et le poste n'est pas connecté au réseau. Si la réponse est positive (Access-Accept), elle contient le numéro de VLAN autorisé. Le commutateur ouvre alors le port sur ce VLAN et le poste peut commencer à travailler. Donc, dans ce type d'authentification, il n'y a pas de communication entre le poste de travail et le serveur Radius. Tous les échanges interviennent entre le commutateur et le serveur.

### 3-2 Principe de l'authentification 802.1X (EAP) :

Si le schéma général de l'authentification 802.1X ressemble à celui de Radius-MAC, les deux méthodes sont, en réalité, très différentes. L'authentification 802.1X est plus compliquée et délicate à mettre en œuvre. Tout d'abord, la différence la plus importante est que, cette fois, un logiciel particulier sera indispensable sur le poste de travail. Ce logiciel est appelé **supplicant**. C'est lui qui va envoyer (1) (Figure I.10) vers le serveur Radius les éléments d'authentification (certificat, identifiant, mot de passe…). Cependant, il ne communique pas directement avec le serveur et d'ailleurs, il ne le connaît pas. C'est le commutateur qui va servir d'intermédiaire (2) (Figure I.10), car il connaît l'adresse du serveur.

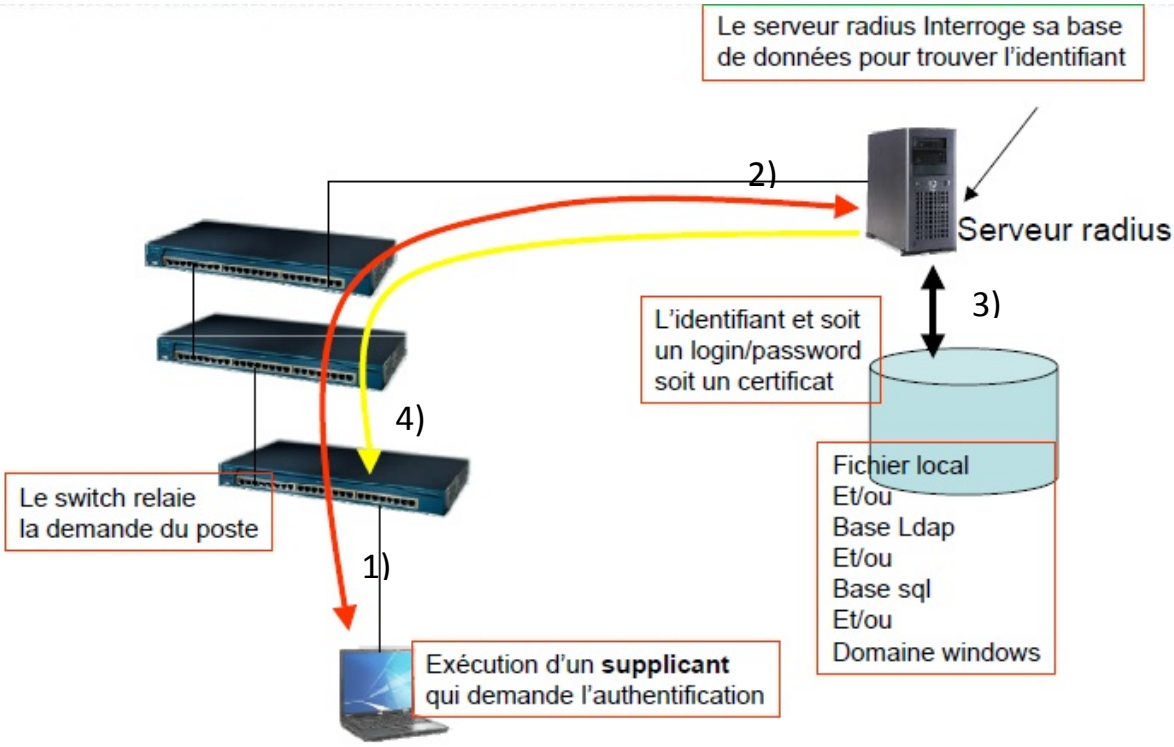



**Figure I.10:** Principes de l'authentification 802.1X. **[PROT S.B]**

Pour interroger sa base de données (3), le serveur Radius a besoin d'un identifiant qu'il utilise comme point d'entrée. Bien sûr, dans ce cas, il ne s'agira pas de l'adresse MAC. L'identifiant sera configuré et envoyé par le supplicant.
Le serveur accepte ou refuse l'authentification et renvoie sa réponse au commutateur (4). Et celui-ci ouvre le port sur le VLAN commandé par le serveur. Mais l'opération est différente du cas authentification Radius-MAC.

Avec Radius-MAC, l'authentification est réalisée sans aucune communication entre le poste de travail et le serveur. En 802.1X, dans la mesure où c'est le supplicant qui envoie les éléments d'authentification, il y a bien une communication. Or, comment peut-il y avoir une communication, et donc un trafic réseau, puisque le port du commutateur n'est pas ouvert et qu'il ne le sera que lorsque le poste aura été authentifié ?

C'est justement là que tient tout le protocole 802.1X. Les ports du commutateur seront configurés d'une façon particulière. Avant d'être complètement ouverts, ils ne laisseront passer qu'un seul type de protocole : EAP. D'ailleurs, l'autre nom de 802.1X est « Port-Based Network Access Control » qui, traduit littéralement, signifie « Accès au réseau basé sur le contrôle de port ».

Tout se passe comme si chaque port était coupé en deux. Une moitié est appelée **port contrôlé** et, au départ, elle est maintenue fermée par le commutateur. L'autre moitié est appelée **port non contrôlé**. Par cette voie, le commutateur n'accepte que le protocole EAP.

Comme l'indique la figure suivante, le supplicant du poste de travail envoie (1) ses informations vers le commutateur dans des paquets EAP. Celui-ci les reçoit par le port non contrôlé et les retransmet (2) encapsulés dans des paquets Radius vers le serveur.

Après interrogation de sa base et, éventuellement après plusieurs échanges avec le commutateur, le serveur lui renvoie (3) l'ordre d'ouvrir complètement le port sur un VLAN donné. C'est ce que fait le commutateur (4) : le poste peut alors utiliser pleinement le réseau.

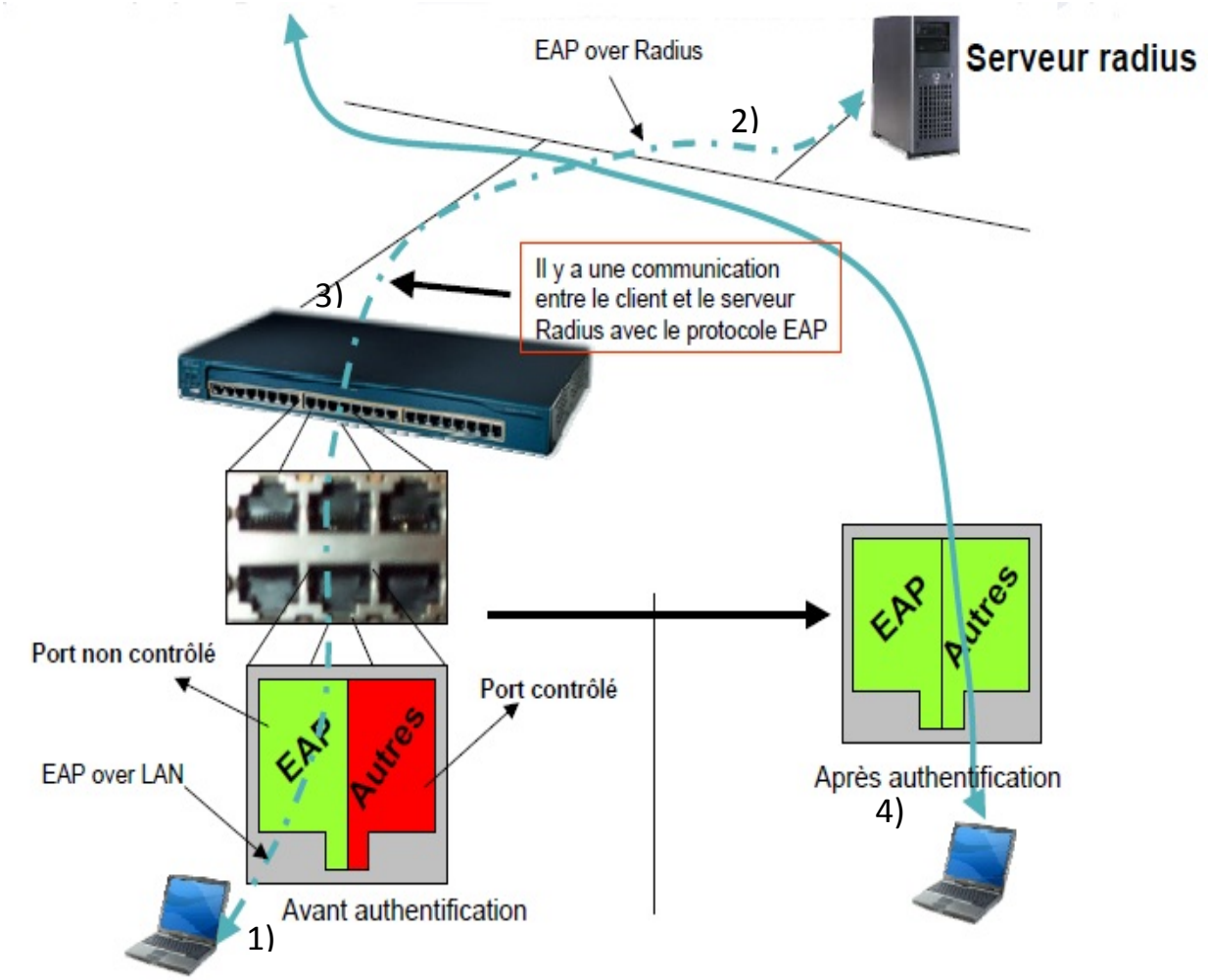



**Figure I.11:** Principe des ports contrôlés et non contrôlés. **[PROT S.B]**

EAP n'est pas un protocole d'authentification mais un protocole de transport du protocole d'authentification. L'intérêt de l'architecture d'EAP est de pouvoir utiliser divers mécanismes d'authentification sans que l'équipement réseau (NAS) n'ait besoin de les connaître. Dans ce cas il agit comme un tunnel transparent vers un serveur qui lui implémente les mécanismes souhaités. Par exemple: Mot de passe, certificats, carte à puce...

## 4- Les extensions du protocole Radius:

### 4-1 Le support des VLANs :

✓Le support des VLANs est réalisé par le biais des attributs de tunnel.

✓ Le support des attributs de tunnel est une extension du protocole de base de RADIUS dont le but initial est de créer des tunnels avec des clients distants.

✓ Ces extensions sont décrites dans le RFC 2868.

**Les attributs concernés sont :**

✓ **Tunnel-Type :** la valeur est VLAN ou 13.

✓ **Tunnel-Medium-Type :** la valeur est 802 pour indiquer qu'il s'applique à un réseau de type IEEE 802 (Ethernet, Token Ring, Wi-Fi).
✓ **Tunnel-Private-Group-Id :** la valeur est le numéro de VLAN qui doit être affecté au port sur lequel est connecté le poste de travail.

### 4-2 Le support de IEEE 802.1X et EAP :

Radius a été étendu pour supporter le protocole EAP et donc l'authentification 802.1x. Pour cela 2 attributs ont été ajoutés: Message-Authenticator et EAP-Message.

- ✓ Les paquets EAP sont encapsulés dans l'attribut EAP-Message.
- ✓ Message-Authenticator est un attribut qui permet de signer les requêtes qui contiennent des attributs EAP-Message. (calcul MD5 sur le contenu d'un access-request + secret partagé)

**Les couches EAP :**

Quatre types de paquets sont utilisés pour le protocole EAP :

- ✓ Request.
- ✓ Response.
- ✓ Success.
- ✓ Failure.

Ces paquets traversent trois couches comme l'indique la figure suivante :

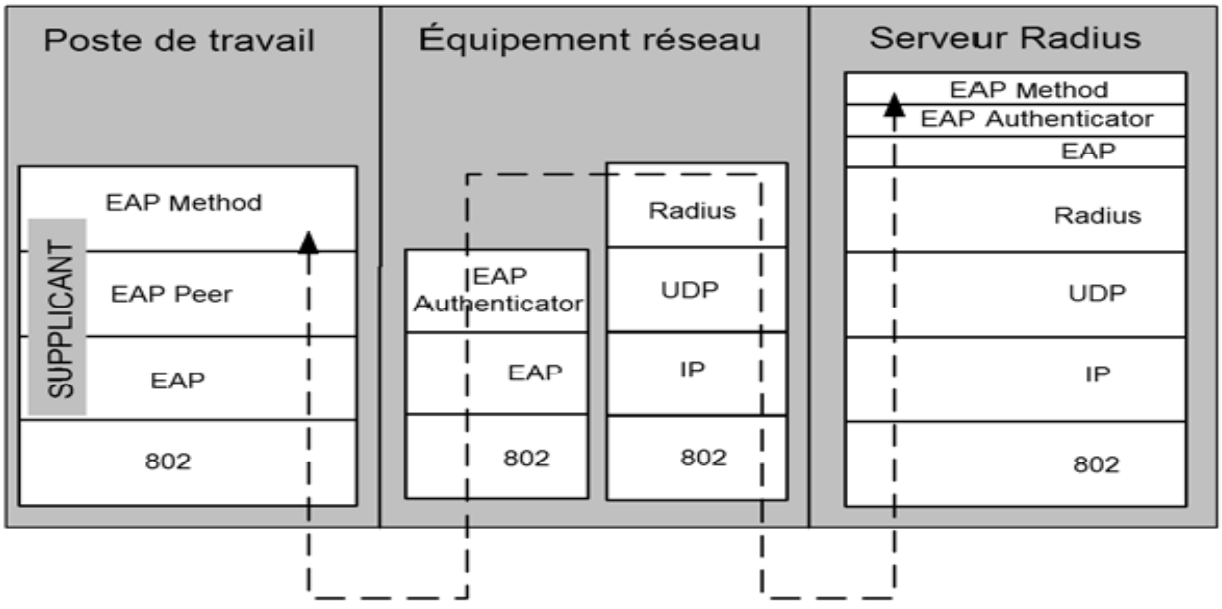



**Figure I.12:** Les couches EAP. **[radEYR07]**

**La couche EAP :**

Elle reçoit et envoie les paquets vers la couche basse (802) et transmet les paquets de type Request, Success et Failure à la couche EAP Peer. Les paquets Response sont transmis à la couche EAP Authenticator ».

**Les couches EAP Peer et EAP Authenticator :**

La couche EAP Peer est implémentée sur le poste de travail, tandis que la couche EAP Authenticator est implémentée sur le NAS et sur le serveur Radius.

Ces couches ont pour rôle d'interpréter le type de paquet Request ou Response et de les diriger vers la couche EAP Method correspondant au protocole d'authentification utilisé (par exemple, TLS).

**La couche EAP Method :**

C'est dans cette couche que se tient le code logiciel du protocole d'authentification utilisé.
On peut découper le protocole EAP en quatre étapes:

- ❖Identité externe.
- ❖Négociation de protocole.
- ❖Protocole transporté.
- ❖Gestion des clés de chiffrement.

**Étape « Identité externe » :**

Cette étape intervient entre le poste de travail, ou plus précisément le supplicant, et le NAS.

**1** - Le supplicant et le NAS négocient l'usage d'EAP.

**2**- Le NAS envoie un paquet EAP de type EAP-Request/Identity, c'est-à-dire qu'il demande au supplicant son identité.

**3**- Le supplicant répond par un EAP-Response/Identity, c'est-à-dire l'identité qui lui est demandée.

**4**- Le NAS fabrique un paquet Access-Request dans lequel à l'intérieur de champ attributs et valeurs, il écrit l'attribut EAP-Message dans lequel il encapsule le paquet EAP venant du supplicant. Il écrira également un attribut User-Name dans lequel il copiera l'identité (celle envoyée dans l'EAP-Response/ identity). Le serveur Radius utilisera le contenu d'User-Name comme point d'entrée dans sa base de données.

**5**- Le NAS envoie le paquet Access-Request au serveur. Le NAS écrit d'autres attributs dans l'Access-Request, parmi lesquels Calling-Station-Id qui permettra au serveur Radius de disposer de l'adresse MAC du poste de travail en plus de l'authentification envoyée par le supplicant.

**Étape « Négociation de protocole » :**

Cette étape correspond à la réception du paquet Access-Request par le serveur et à sa réponse vers le supplicant afin de proposer une méthode d'authentification.

**1**-Le serveur reçoit le paquet Access-Request.

**2**- Il construit un paquet Access-Challenge dans lequel il écrit un attribut EAPMessage formé d'un paquet EAP-Request qui contient une proposition de protocole d'authentification. Par exemple, il propose PEAP.

**3**-Le NAS décapsule le paquet EAP contenu dans EAP-Message et le transfère sur la couche EAP vers le supplicant. Celui-ci répond par un paquet EAP-Response. S'il connaît le protocole proposé et qu'il est configuré, il l'acceptera. Dans le cas contraire, il proposera un protocole pour lequel il est configuré, par exemple TLS.

**4**-La réponse du supplicant est encapsulée, comme dans la première phase, dans un nouveau paquet Access-Request. Si le serveur accepte ce protocole alors on passe à la troisième phase, c'est-à-dire l'exécution du protocole d'authentification. Dans le cas contraire, il envoie un Access-Reject au NAS.

**Étape « Protocole transporté » :**

Cette étape correspond à l'exécution du protocole d'authentification transporté. Le principe est le même que pour les deux premières étapes, c'est-à-dire un échange de paquets Radius Access-Request/Access-Challenge encapsulant des paquets EAPRequest ou EAP-Response. La quantité et le contenu de ces échanges dépend du protocole « Le protocole EAP/TLS », « Le protocole PEAP » et « Le protocole EAP/TTLS »).

**Étape « Gestion des clés de chiffrement » :**

Cette étape n'a de sens que dans le cas du Wi-Fi. Elle permet la gestion dynamique des clés de chiffrement.

**4-3 Support Authentification avec certificat (TLS) :**

La figure suivante décrit le scenario d'authentification avec certificat ;

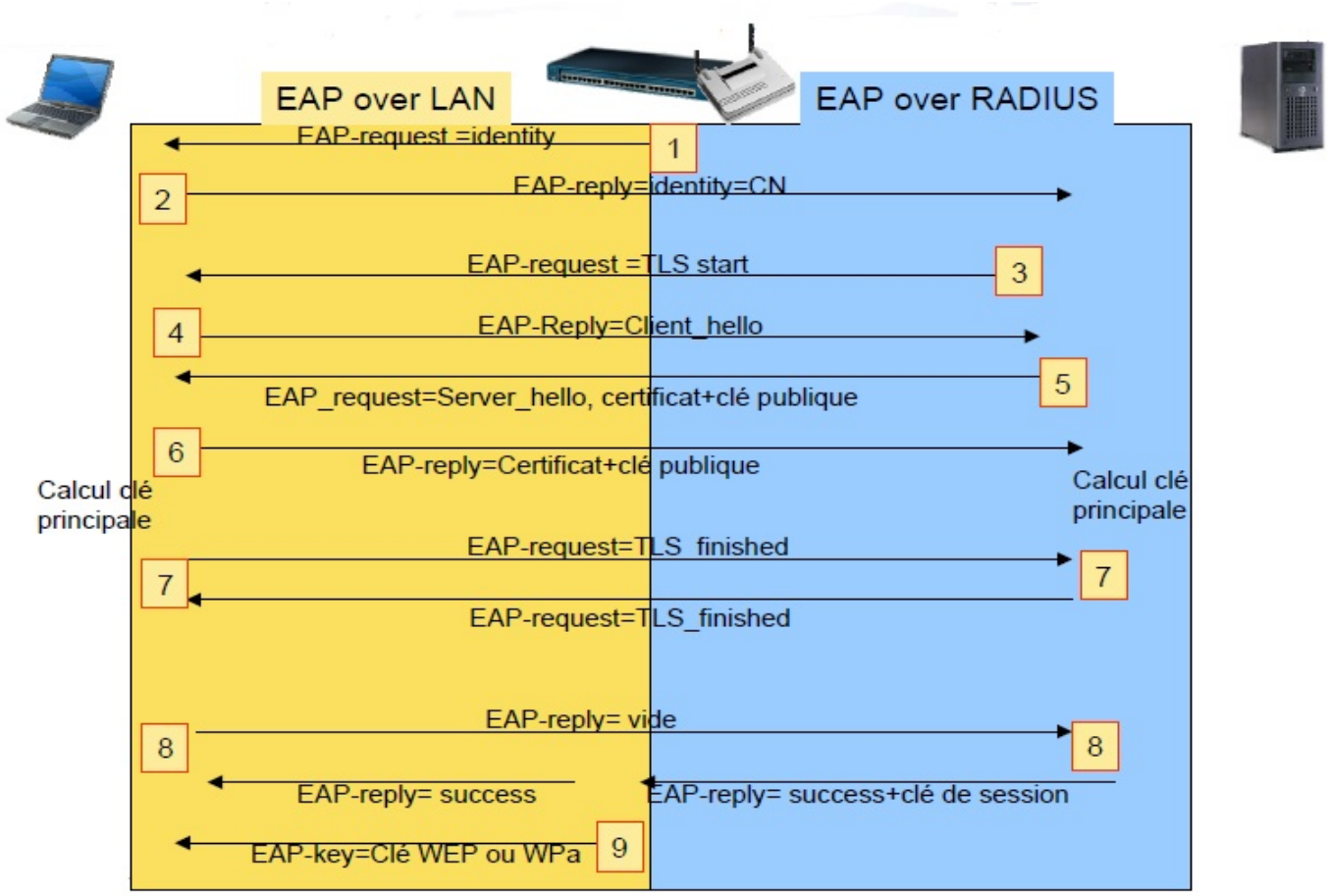



**Figure I.13:** Le protocole EAP/TLS. **[P-ATH S.B]**

1- Le NAS envoi au client une requète EAP lui demandant son identité
2- Le client répond avec le CN comme identité
3- Le serveur démarre la séquence TLS par l'envoi du message TLS_start
4- Le client répond par un message client_hello :
   La version de TLS
   Un challenge (nombre aléatoire)
   Un identifiant de session

La liste des algorithmes de chiffrement supportés par le client

5- Le serveur répond par un message server hello

Son certificat et sa clé publique

Demande au client d'envoyer son certificat

Un challenge

Un identifiant de session calculé à partir de celui du client.

Choisit un algorithme de chiffrement en fonction de ceux connus par le client

6- Le client vérifie le certificat du serveur et envoi le sien et sa clé publique

7- Le client et le serveur calculent une clé de chiffrement pour la session principale (à partir des challenges échangés).

8- Le client renvoi une réponse EAP vide et le serveur répond par un message EAP_success avec une clé de session pour la borne wifi.

9- A partir de cette clé de session, la borne calcule une clé WEP ou WPA et l'envoi au client.

Dans le cas d'authentification EAP/TLS la clé de session principale n'est pas utilisée. Seul l'échange de validation mutuelle des certificats est utile.

**4-4 Authentification avec login/password (PEAP) :**

La figure suivante décrit le scenario d'authentification PEAP;

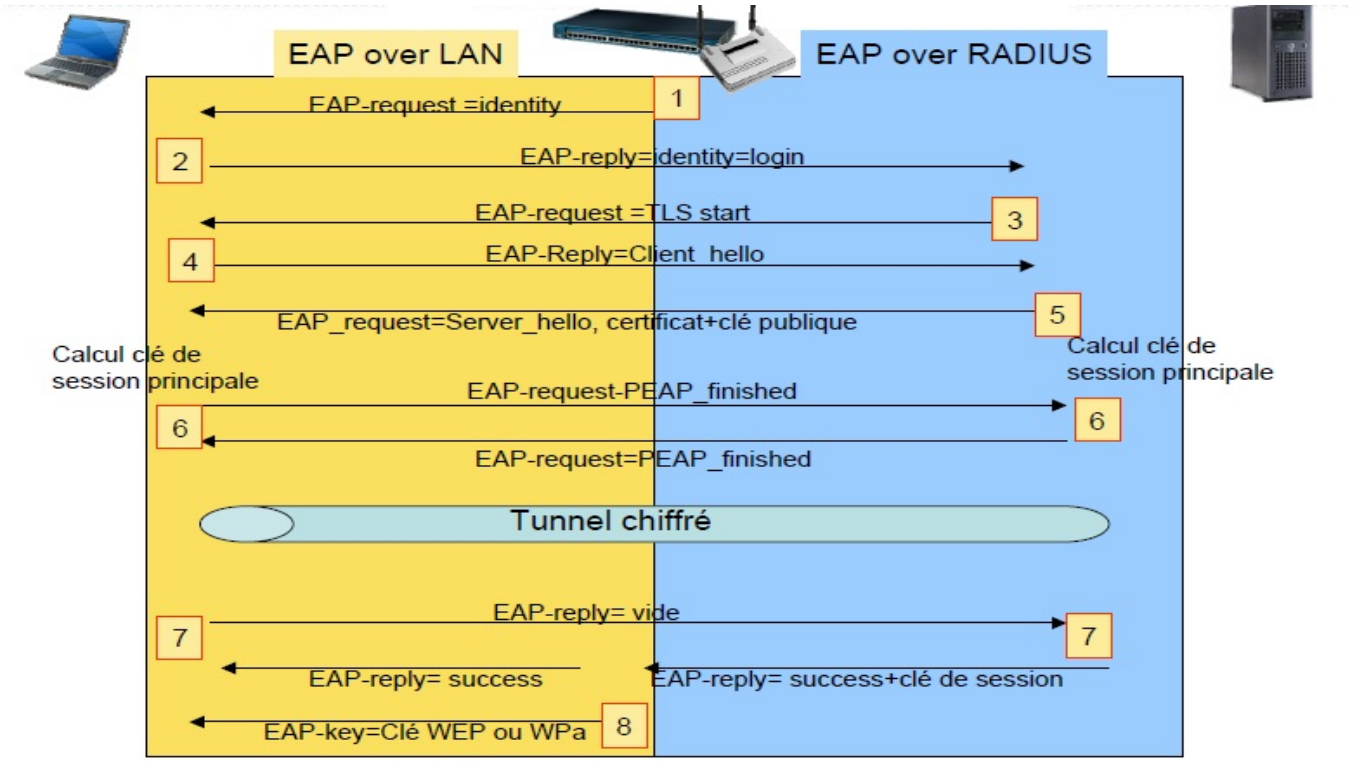



**Figure I.14:** Le protocole EAP/PEAP. **[P-ATH S.B]**

1 à 5) Les échanges sont presque similaires à EAP-TLS. Le client authentifie le serveur par l'intermédiaire d'un certificat (étape 5).

6) Cette étape diffère légèrement d'EAP-TLS car le client n'a pas besoin de fournir de certificat, la clé qui sert à chiffrer la session peut donc être créée directement. À la fin de cette étape, le TLS handshake est terminé, les échanges suivants seront donc chiffrés par la clé de session.

7) En effet, l'établissement d'un tunnel TLS permet de chiffrer les échanges, le client fournit donc ses identifiants (login/mot de passe) au serveur en utilisant par exemple MS-CHAPv2.

8 et 9) Similaires à EAP-TLS EAP-TTLS et EAP-PEAP sont des méthodes très proches et l'utilisation d'un tunnel TLS chiffré leur confère un bon niveau de confidentialité. EAP-PEAP présente l'avantage d'être supporté nativement par Windows XP et 2000. EAP-TTLS permet une meilleure interopérabilité avec les serveurs Radius ne supportant pas EAP.

## 5- Processus d'authentification et autorisation :

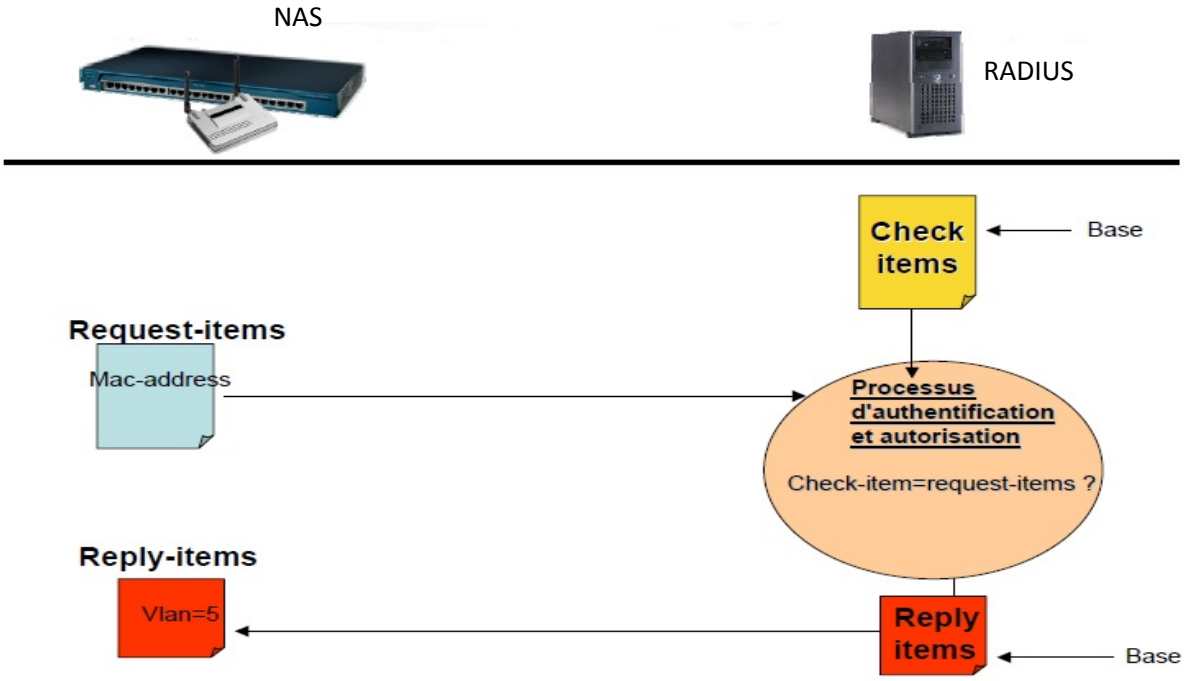



**Figure I.15:** Processus d'authentification et autorisation. **[P-ATH S.B]**

- L'équipement NAS envoi un **access-request** contenant une liste d'attributs appelés **request_items.** Par exemple, l'adresse MAC est un request-item.
- Le serveur Radius dispose dans sa base d'authentification/autorisation d'une liste de **check-items** associée à chaque utilisateur/machine connu.
- Radius interroge sa base pour trouver une entrée qui correspond au username envoyé et dont les check-items matches les request-item.
- Si aucune correspondance n'est trouvée un **access-reject** est envoyé.
- Si une correspondance est trouvée, une reply-list est formée à partir des **reply-items** contenu dans la base. Par exemple, le numéro de vlan est un reply-item.
- Le username envoyé est authentifié (mot de passe, certificats…..)
- Si l'authentification est ok, la reply-list est envoyé à l'équipement NAS avec un access-accept.

### 5-1 Eléments constitutifs de la base de données :

- ✓ **Identifiant :** Un username fournis par le client radius (switch) ou par le supplicant.
- ✓ **La méthode d'authentification:** Elle peut être indiquée explicitement dans la base de données ou bien être déduite implicitement en fonction du protocole utilisé par le NAS ou le supplicant. ( Local, EAP, LDAP)

- ✓ **Les check-items: Exemples:** Mot de passe : Fourni par le NAS dans le cas d'une authentification par adresse MAC. Dans ce cas le mot de passe est l'adresse MAC.
- ✓ **Les reply-items:** Exemple: le SSID ou le N° de vlan.

**Conclusion :**

Après avoir détaillé le principe de fonctionnement des HotSpot Wi-Fi où nous avons présenté sa structure globale qui est composée d'un Portail Captif et d'un serveur d'Authentification. Ces deux serveurs ont pour rôles d'assurer le bon fonctionnement de notre Hotspot. Dans le deuxième chapitre nous allons présenter la carte électronique sur la qu'elle sera installé notre HotSpot (portail captif, serveur d'authentification radius…etc). Cette carte est dite Raspberry PI.

# CHAPITRE N°2

**Introduction :**

Les systèmes embarqués prennent une place de plus en plus importante dans notre société, ils servent à contrôler, réguler des dispositifs électroniques grâce à des capteurs embarqués dans des robots, des véhicules spatiaux, …etc. Ces systèmes embarqués sont souvent utilisés par le public dans la vie de tous les jours sans même qu'on ne s'en rende compte, par exemple dans les systèmes de freinage d'une voiture, le pilot automatique d'un avion,... dans ce chapitre, notre but consiste à faire une brève présentation sur les systèmes embarqués, puis on étudiera la carte électronique ‘’Raspberry-Pi’’ du point de vue matériel et logiciel.

## I. Généralités sur les systèmes embarqués :

Un système embarqué peut être défini comme un système électronique et informatique autonome dédié à une tâche bien précise et répondant souvent à des contraintes temps réel. Ses ressources disponibles sont généralement limitées, elles sont d'ordre spatial (taille limitée) et énergétique (consommation restreinte).

La majorité de ces systèmes, centralisés ou distribués sont critiques pour la sécurité, soit parce qu'ils sont au cœur du comportement du système, soit parce qu’ils interagissent avec l'être humain dans des situations critiques. Assurer leur fiabilité et leur sûreté de fonctionnement est un défi majeur.

La complexité croissante des systèmes embarqués nécessite des méthodes de conception globale qui tiennent compte des fonctionnalités et des constituants (capteurs, actionneurs, contrôleurs, réseaux), des perturbations et des fortes contraintes sur coût et l’environnement (retards, incertitudes,...).

### 1. Historique :

Les premiers systèmes embarqués sont apparus en 1971 avec l'apparition de l’Intel 4004, développé en 1971. Ce premier microprocesseur, était le premier circuit intégré incorporant. Tous les éléments d'un ordinateur dans un seul boîtier: unité de calcul, mémoire, contrôle des entrées / sorties. Alors qu'il fallait auparavant plusieurs circuits intégrés différents, chacun dédié à une tâche particulière, avec ce type de microprocesseur un seul pouvait assurer autant de travaux différents que possible.

Ce sont alors les débuts de l'informatique embarquée. [5]

### 2. Contraintes :

Les systèmes embarqués exécutent des tâches prédéfinies et ont un cahier des charges contraignant à remplir, qui peut être:

- **De coût :** Le prix de revient doit être le plus faible possible surtout s'il est produit en grande série.
- **D'espace mémoire:** ayant un espace mémoire limité de l'ordre de quelques Go maximum (bien que la taille vienne à être de moins en moins limitée grâce à la miniaturisation des éléments). Il convient de concevoir des systèmes embarqués qui répondent aux besoins au plus juste pour éviter un surcoût.

- **De puissance de calcul :** Il convient d'avoir la puissance de calcul juste nécessaire pour répondre aux besoins et aux contraintes temporelles de la tâche prédéfinie. Ceci en vue d'éviter un surcoût de l'appareil et une consommation excédentaire d'énergie (courant électrique).
- **D'autonomie :** La consommation énergétique doit être la plus faible possible, due à l'utilisation de batteries et/ou, de panneaux solaires voire de pile à combustible pour certains prototypes.
- **Temporel :** dont les temps d'exécution et l'échéance temporelle d'une tâche sont déterminés (les délais sont connus ou bornés *a priori*). Cette dernière contrainte fait que généralement de tels systèmes ont des propriétés temps réel.
- **De sûreté de fonctionnement** : Car s'il arrive que certains de ces systèmes embarqués subissent une défaillance, ils mettent des vies humaines en danger ou mettent en périls des investissements importants. Ils sont alors dits « critiques » et ne doivent jamais faillir. Il faut comprendre toujours donner des résultats justes, pertinents et ce dans les délais attendus par les utilisateurs.
- **De sécurité :** Ces systèmes peuvent se révéler être porteurs d'informations confidentielles pour leur(s) utilisateur(s), qu'il convient de conserver et de protéger.

## 3. Architecture :

Les systèmes embarqués utilisent généralement des microprocesseurs à basse consommation d'énergie ou des microcontrôleurs, dont la partie logicielle est en partie ou entièrement programmée dans le matériel, généralement en mémoire dans une mémoire morte (ROM), EPROM, EEPROM, FLASH, (on parle alors de firmware) [S.EMB B.B 08].

## 4. Caractéristiques :

- Plutôt que des systèmes universels effectuant plusieurs tâches, les systèmes embarqués sont étudiés pour effectuer des tâches précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de fiabilité et de rentabilité. D'autres ayant peu de contraintes au niveau performances permettent de simplifier le système et de réduire les couts de fabrication.
- Les systèmes embarqués ne sont pas toujours des modules indépendants. Le plus souvent ils sont intégrés dans le dispositif qu'ils contrôlent.
- Le logiciel créé pour les systèmes embarqués est appelé *firmware*. Il est stocké dans la mémoire en lecture seule ou dans la mémoire flash plutôt que dans un disque dur. Il fonctionne le plus souvent avec des ressources matérielles limitées : un petit écran et peu de mémoire …etc.

## 5. Interface utilisateur :

Certains systèmes embarqués peuvent ne pas avoir d'interface utilisateur (ils sont alors spécialisés dans une seule tâche). Mais cette interface peut également être similaire à celle d'un système d'exploitation d'ordinateur (par exemple un PDA).

- Les systèmes les plus simples comportent uniquement des boutons, des LED.
- Les systèmes les plus complexes peuvent avoir un écran tactile ou encore un écran comportant des boutons de façon à minimiser l'espace. La signification des boutons

change selon l'écran et la sélection se fait naturellement en pointant la fonction désirée.

- Les ordinateurs de poche possèdent en général un bouton au style de « joystick » pour la navigation.
- Avec l'explosion du web, les fabricants de systèmes embarqués ont proposé une nouvelle option : une interface au style d'une page web sur une connexion au réseau. Cela permet d'éviter le coût d'un système sophistiqué tout en conservant une interface complète sur un autre ordinateur, quand cela est nécessaire. Interface couronnée de succès pour les installations permanentes à distance, les routeurs en particulier.

## 6. Fiabilité :

Les systèmes embarqués sont la plupart du temps dans des machines qui doivent fonctionner en continu pendant de nombreuses années, sans erreurs et, dans certains cas, réparer eux-mêmes les erreurs quand elles arrivent. C'est pourquoi les logiciels sont toujours développés et testés avec plus d'attention que ceux pour les PC. Les pièces mobiles non fiables (par exemple les lecteurs de disques, boutons ou commutateurs) sont proscrites.

La question de la fiabilité peut inclure :

- Le système ne peut pas être éteint pour des réparations ou ce sont des réparations inaccessibles.

La solution peut être des pièces détachées supplémentaires ou un "mode mou" du logiciel qui fournit un fonctionnement partiel.

- Le système doit rester en marche pour des raisons de sécurité. Souvent, les sauvegardes sont effectuées par un opérateur.

Dans ce cas, le « mode mou » est toléré.

- Un arrêt du système peut provoquer des pertes monétaires énormes s'il s'éteint.

Par exemple : les systèmes de ponts ou d'ascenseurs, les transferts de fond, les salles de bourse, les ventes ou services automatiques…

## 7. Domaines d'applications :

- Astronautique : fusée, satellite artificiel, sonde spatiale, etc.
- Automate programmable industriel, contrôle-commande
- Electroménager : télévision, four à micro-ondes
- Environnement
- Équipement médical
- Guichet automatique bancaire (GAB)
- impression : imprimante multifonctions, photocopieur, etc.
- Informatique : disque dur, Lecteur de disquette, etc.
- Métrologie
- Militaire : missile
- Multimédia : console de jeux vidéo, assistant personnel
- Télécommunication : Set-top box, téléphonie, routeur, pare-feu, serveur de temps, Téléphone portable, etc.

- Transport : Automobile, Aéronautique (avionique), Ferroviaire, etc. [5]

## II. Présentation du Raspberry Pi :

Le Raspberry Pi est un ordinateur dont les particularités sont la très petite taille (la taille d'une carte de crédit). Il a été créé par l'anglais David Braben, dans le cadre de sa fondation Raspberry Pi, dans le but d'encourager l'apprentissage de la programmation informatique. [15]

Pour la petite histoire, *raspberry* signifie *framboise* en anglais.

Il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, alimentation, sans boîtier, sans clavier, sans souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Son prix de vente était estimé à 25 $, soit 19,09 €, début mai 2011. Les premiers exemplaires ont été mis en vente le 29 février 2012 pour environ 25 €. Début 2015, plus de cinq millions de Raspberry Pi ont été vendus. [8]

### 1. Historique :

#### 1-1- Conception

En 2006, les premiers prototypes du Raspberry Pi sont développés sur des microcontrôleurs Atmel *ATmega 644*. Le schéma et le plan du circuit imprimé sont rendus publics. Cet ordinateur s'inspire *du BBC Micro* d'*Acorn Computer* (1981) et est destiné à encourager la jeunesse à la programmation. Le premier prototype ARM est intégré dans un boitier de la même taille qu'une clé USB avec un port USB d'un côté et un port HDMI de l'autre.

#### 1-2- Prototype

En août 2011, 50 cartes version Alpha sont construites, ces cartes étant identiques du point de vue fonctionnel au modèle B prévu mais elles sont plus grandes pour faciliter le débogage. Une démonstration montre la carte exécutant une distribution Debian avec un bureau LXDE, *Quake 3* en 1080p et une vidéo en Full HD MPEG-4 par HDMI.[8]

En octobre 2011, une version de *RISC OS 5* tournant sur la carte est présentée. Après une année de développement la version grand public sera terminée en novembre 2012. En décembre 2011, 25 cartes modèle B ont été construites et testées. [14] Le design des cartes version Beta est le même que les cartes grand public. Une seule erreur a été découverte dans le design, certaines broches du CPU ne fonctionnaient pas correctement, l'erreur a été corrigée avant la première production.

La première semaine de l'année 2012, 10 premières cartes sont mises aux enchères sur eBay. L'une est achetée anonymement et donnée au Centre « for Computing History », dans le *Suffolk* en Angleterre. Les 10 cartes qui représentaient un prix de 220 £ ont été vendues pour un total de 16 000 £. La carte possédant le numéro de série 01 est achetée pour 3 500 £. [8]

### 1-3- Lancement

La première série de 10 000 cartes est produite en Taïwan et en Chine. Les livraisons de la première série sont annoncées pour mars 2012 en raison de l'installation d'un mauvais connecteur Ethernet, mais la fondation annonce qu'elle s'attend à augmenter la production des futures séries sans difficulté.

Les ventes débutent le 29 février 2012 à 06:00 UTC. Au même moment est annoncé un modèle A à 256 MB de RAM au lieu des 128 MB prévus. Le site web de la fondation affiche : *« Six ans après le début du projet, nous sommes presque à la fin de la première session de développement - cependant ce n'est que le début de l'histoire de Raspberry Pi.* ». [14]

En septembre 2012, 500 000 cartes ont été vendues. Le 6 avril 2012, la fondation annonce que le Raspberry Pi a obtenu la certification CE, demandée par les distributeurs pour pouvoir lancer la distribution auprès des premiers acheteurs. [8]

Au 22 mai 2012, 20 000 cartes ont été envoyées. En juillet 2012, 4 000 unités sont produites chaque jour.

En septembre 2012, la Fondation Raspberry Pi annonce une deuxième révision du modèle B. De plus, les futures séries seront fabriquées au Royaume-Uni, dans les usines Sony de *Pencoed*, au *Pays de Galles*. Il est estimé que 30 000 unités seraient produites par mois, créant 30 emplois. la fréquence du processeur est passé de 700 à 1 000 MHz.

En octobre 2013, c'est un million de Raspberry Pi qui a été produits au Royaume-Uni. Le deux millionième kit est envoyé entre le 24 et le 31 octobre.

En avril 2014, une nouvelle version est annoncée, elle divise la carte en deux parties : une partie calcul et une partie interface d'entrées-sorties. La partie calcul *Compute Module* comporte 512 Mio de mémoire vive et 4 Gio de mémoire flash. La dimension de la carte est réduite au format SO-DIMM (environ 68 × 30 mm). La partie interface d'entrées-sorties *Compute Module IO Board* comporte des connecteurs HDMI et USB. Au mois de juin, trois millions de Raspberry Pi ont été vendus.[8]

## 2- Architecture matérielle :

Le Raspberry Pi possède un processeur ARM11 à 700 MHz. Il inclut 1, 2 ou 4 ports USB, un port RJ45 et 256 Mo de mémoire vive pour le modèle d'origine (512 Mo sur les dernières versions). Son circuit graphique BMC Videocore 4 en particulier permet de décoder des flux Blu-Ray full HD (1080p 30 images par seconde), d'émuler d'anciennes consoles et d'exécuter des jeux vidéo relativement récents.

## 3- Spécifications :

**Modèle A :**

- Processeur : *ARM1176JZF-S* (ARMv6) 700 MHz *Broadcom 2835* (dispose d'un décodeur Broadcam VideoCore IV, permettant le décodage *H.264 FullHD 1080P* et le calcul des opérations à virgule).

- RAM : 256 Mo.
- 2 Sorties vidéo : Composite et *HDMI*.
- 1 Sortie audio stéréo Jack 3,5 mm.
- Unité de lecture-écriture de carte mémoire : SDHC / MMC / SDIO.
- 1 Port USB 2.0.
- Prise pour alimentation Micro-USB (consommation : 400 mA + périphériques).
- Des entrées / sorties supplémentaires sont accessibles directement sur la carte mère *via* des pins (broches) 3v3: GPIO, S2C, I2C, SPI.

**Figure II.1:** Raspberry PI Model A. [9]

**Modèle A+ :**

Différences avec le A :

- Plus petit : 65 mm de long contrairement a 86 mm
- Lecteur de carte microSD en lieu du lecteur SD
- GPIO 40 broches
- Nouveau chipset audio
- Consommation électrique moindre
- Prix réduit à 20 $


**Figure II.2:** Raspberry PI Model A+. [9]

**Modèle B :**

**Figure II.3:** Raspberry PI Model B. [9]

**Modèle B Rev1 :**

Différences :

- 2 ports USB 2.0 au lieu de l'unique port du modèle A.
- 1 port réseau Fast Ethernet (10/100 Mbits/s) *via* le même composant SMSC.

**Modèle B Rev2 :**

Différences :

- Implantation du *reset*.
- Support JTAG.
- Support I2C.
- Suppression de quatre signaux GPIO utilisés pour l'identification de version, et réaffectation à d'autres rôles.
- Deux trous de fixation.
- Correction du marquage des LED sur la platine.

**Modèle B** 512 Mo :

- Prise pour alimentation micro-USB (consommation : 700 mA).

Différences :

- La RAM passe à 512 Mo (au lieu de 256 Mo sur les modèles précédents).

**Modèle B+ :**

Ce modèle est sorti en fin en juillet 2014.

Différences par rapport au modèle initial :

- GPIO 40 broches.
- 4 ports USB 2.0 et meilleur comportement en cas de surcharge.
- micro SD.
- réduction de consommation de 3,5 W à 3 W.
- meilleur circuit audio.

**Figure II.4:** Raspberry PI Model B+. [9]

**Modèle Pi 2 :**

Le 2 février 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 2, plus puissant, il est équipé d'un processeur Broadcom BCM2836, quatre cœurs ARMv7 à 900 MHz, accompagné de 1 Go de RAM.

Il possède les mêmes dimensions et connectiques que le modèle B+.

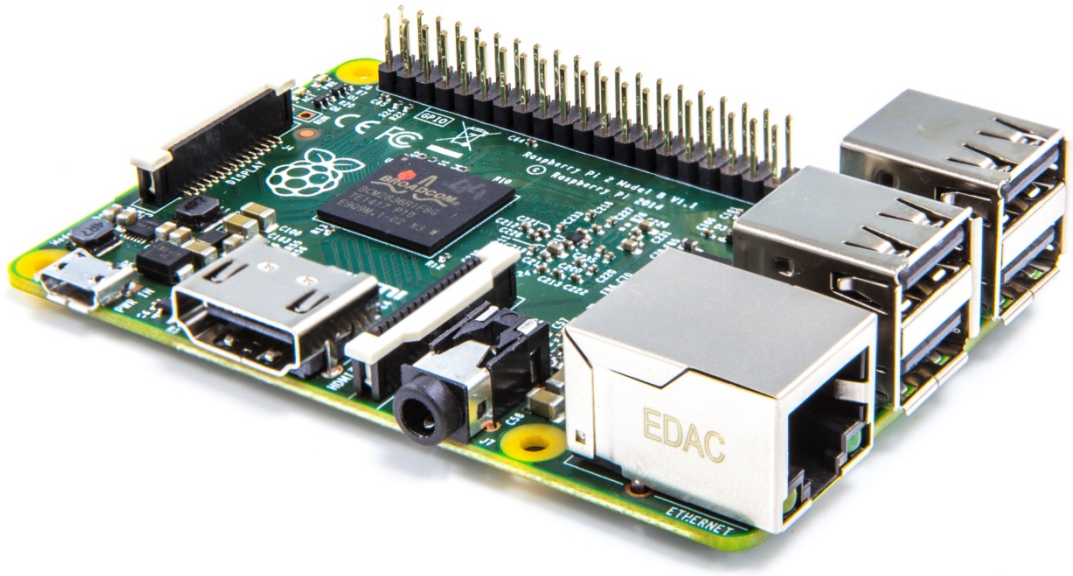



**Figure II.5:** Raspberry PI Model B2. [9]

**4- Tableau comparatif :**

Le tableau suivant résume la déférence entre les déférents modèles de Raspberry Pi.

| Modèle | A | A+ | B | B+ | B2 |
|---|---|---|---|---|---|
| CPU | Monocœur ARM 700 MHz | | | | Quadricœur ARM 900 MHz |
| GPU | Décodeur vidéo Broadcam VideoCore IV | | | | |
| RAM | 256 MO | | 512 MO | | 1 GO |
| USB | 1 * USB2.0 | | 2 * USB2.0 | 4 * USB2.0 | |
| Audio/vidéo | Jack 3.5, composite et HDMI | HDMI et jack audio/vidéo | Jack 3.5, composite et HDMI | HDMI et jack audio/vidéo | |
| Ethernet | non | | Port Ethernet | | |
| Entrées/sorties | GPIO 26 pts | GPIO 40 pts | GPIO 26 pts | GPIO 40 pts | |
| Stockage | SD | Micro SD | SD | Micro SD | |
| Dimensions | 86*54 | 65*54 | 86*54 | | |

| | | | | |
|---|---|---|---|---|
| **Poids** | 45g | 23g | 45g | |
| **Consommation** | 1.5W | 1W | 3.5W | 3W |

**Tableau II.1:** Tableau comparatifs pour les déférents types de Raspberry-Pi. [8]

## 5- Équipement supplémentaire :

De base, la Raspberry-Pi est fourni sans boîtier, alimentation, clavier, souris ni écran. Ceci dans le but de minimiser les coûts et de pouvoir recycler d'autres matériels. Des boîtiers adaptés à la carte mère (dont certains originaux) sont disponibles sur la plupart des sites commercialisant la Raspberry-Pi. Un chargeur de Smartphone (micro-USB vers USB) convient parfaitement pour alimenter la carte. En utilisant la prise USB, on peut connecter de nombreux périphériques : disques durs externes, lecteur Blue-ray, clef wifi, webcam, imprimante...

Il est possible de connecter la Raspberry-Pi avec :

- une carte réseau Ethernet ou Wi-Fi, pour établir une connexion en réseau, notamment avec Internet pour le modèle A. Le modèle B introduit durant l'été 2011 comporte une carte Ethernet intégrée.
- Des unités de stockage externe, clés USB ou disques durs alimentés séparément.
- Tout autre périphérique USB disposant d'un pilote informatique compatible : clé Bluetooth, clé Wi-Fi, mémoire flash, imprimante, webcam…

La fondation Raspberry Pi proposera à la vente une palette de périphériques validés compatibles. D'autres périphériques seront possibles, mais la disponibilité d'un pilote informatique adéquat, compatible avec le processeur, le système d'exploitation, le circuit d'interface, et le périphérique lui-même, devra être scrupuleusement vérifiée.

## 6- Les systèmes d'exploitation compatibles avec la Raspberry-Pi:

Les systèmes d'exploitation compatibles Raspberry Pi / ARM incluent : [15]

- Diverses distributions Linux - La plateforme par défaut du Raspberry Pi ;
  - ✓ **Debian** « wheezy » est recommandée par la fondation Raspberry Pi avec sa version dédiée ***Raspbian*** (Raspberry & Debian). L'image est disponible sur le site officiel de la fondation.
  - ✓ **Fedora** « Raspberry Pi Fedora remix ». L'image est disponible sur le site officiel de la fondation.
  - ✓ **Arch Linux** est fonctionnel avec sa version ARM pour Raspberry Pi. L'image est disponible sur le site officiel de la fondation.
  - ✓ **Gentoo** est utilisable classiquement.
  - ✓ **Slackware** est également utilisable classiquement.
  - ✓ **Suse** est fonctionnelle avec sa version ARM pour Raspberry Pi. L'image est disponible sur le site officiel dédié à cette carte.
  - ✓ **Kali Linux**, un système dédié au pentest (Évolution de BackTrack).

- **Firefox OS** (anciennement Boot to Gecko), le système d'exploitation mobile développé par Mozilla a été annoncé comme fonctionnel.
- **RISC OS** .
- **NetBSD** (aucun support de OpenBSD pour le Pi n'est à l'ordre du jour.)
- **Windows 10** : une version dédier est disponible. [14]

## Conclusion :

Après avoir présenté la carte électronique Raspberry-Pi où nous avons commencé par la présentation des différents modèles, puis nous avons réalisé une comparaison dans laquelle nous avons spécifié les capacités matérielles et logiciels de chaque carte, et enfin nous avons cité l'ensemble des systèmes d'exploitations qui sont installable sur ce type de carte électronique.

Dans le chapitre suivant nous allons faire l'analyse et la conception de notre Hotspot, en se basant sur UML.

# CHAPITRE N°3

## Introduction :

Dans les deux chapitres précédents nous avons défini les différentes technologies qui participent au fonctionnement de notre Hotspot.

La conception d'une solution logicielle doit être prise avec précision et détail, précédé d'une démarche méthodologique, car elle est le reflet du futur système avant même sa concrétisation. On présente dans ce chapitre la démarche de conception de notre système.

# I. Analyse :

Cette activité commence par l'étude des cas d'utilisations et de leurs scénarios ainsi que les besoins fonctionnels du système (ce que le système doit faire en réponse à une requête d'un utilisateur c.-à-d. entre les déférents acteurs qui mettent le système en fonction).

Ce modèle d`analyse est constitué par des classes et de collaboration des classes qui traduisent les comportements dynamiques et détaillés dans les cas d`utilisation et les besoins. Elle se concentre sur les besoins fonctionnels sans prendre compte des contraintes d`architectures de la machine, pourvue que les besoins fonctionnels exprimés par les cas d`utilisations soient réalisables dans le système.

## 1- Spécification des besoins :

### 1-2 Identification des acteurs :

Un acteur représente un rôle que peut jouer l'utilisateur avec le système. Cet acteur peut être humain ou automate. Dans notre cas, nous avons identifié trois acteurs :

- Utilisateur : toute personne qui se connecte au point d'accès.
- portail captifs : avant que l'utilisateur n'ait l'accès à internet, il doit passer par le portail Ce dernier activera la page de publicité puis exposera, à la fin, les termes et les conditions de connexion via le point d'accès.
- RADIUS : a pour but d'ajouter les nouveaux utilisateurs qui se connectent au point d'accès ou/et de leurs couper l'accès internet dés que le temps de connexion autorisé est atteint.

### 1-3 Spécification des tâches :

Pour chaque acteur, nous avons spécifié les tâches qu'il assure. Le tableau suivant résume ces tâches :

| Acteur | Tâches |
|---|---|
| utilisateur | **T01-** Sélectionne le SSID.<br>**T02-** Consulte une page publicitaire.<br>**T03-** Accède à la page d'authentification.<br>**T04-** Consulte la page des termes de connexion.<br>**T05-** Accepte les termes. |
| Portail captif | **T06-** Autorise l'accès à internet pour les nouveaux utilisateurs.<br>**T07** – Empêche l'accès à internet aux utilisateurs auquel leur délai de connexion autorisé est atteint. |
| RADIUS | **T08-** Insert les utilisateurs dans la base de données MySQL.<br>**T09-** Récupère les informations des utilisateurs de la base de données MySQL. |

**Tableau III.1:** Tableau spécification des tâches.

### 1-4 Spécification des scénarios :

Chaque tâche effectuée par un ou plusieurs acteurs sera décrite par un ensemble de scénarios.

| Acteur | Tâches | Scénarios |
|---|---|---|
| Utilisateur | **T01 –** sélectionne le SSID.<br>**T02-** consulte une page publicitaire.<br>**T03-** accède à la page d'authentification.<br>**T04-** consulte la page des termes de connexion.<br>**T05-** accepte les termes. | **S0-** se connecter au wifi.<br>**S01-**Ouvrir un navigateur web.<br>**S02-** Taper un URL.<br>**S03-** Accéder à la page publicitaire puis a la page d'authentification.<br>**S04-** Sélectionner le lien termes et conditions.<br>**S05-** Accéder à internet. |
| Portail captif (CoovaChilli) | **T06-** autorise l'accès à internet aux nouveaux utilisateurs.<br>**T07** – empêche l'accès à internet aux utilisateurs auquel leur délai de connexion autorisé est atteint. | **S06-** Communiquer avec RADIUS.<br>**S07-** Récupérer les informations de connexions de chaque utilisateur. |

| | | |
|---|---|---|
| RADIUS | **T08-** insert les utilisateurs dans la base de données MySQL.<br>**T09-** Récupère les informations des utilisateurs la base de données MySQl. | **S08-** Insérer les informations d'authentification et de comptabilisation des utilisateurs dans la table radcheck de la base de données MySQL.<br>**S09-**Récupérer les informations d'authentification et de comptabilisation chaque utilisateur à partir des tables radcheck, radacct et radreply de la base de données MySQL. |

**Tableau III.2:** Tableau spécification des scénarios.

**1-5 Les cas d'utilisation :**

Les cas d'utilisation que nous pouvons recenser sont :

- **Les cas d'utilisation d'un utilisateur :**
  - Sélectionne le SSID.
  - Consulte une page publicitaire.
  - Accède à la page d'authentification.
  - Consulte la page des termes de connexion.
  - Consulte la page des termes de connexion.

- **Les cas d'utilisation du Portail captif:**
  - Autorise l'accès à internet pour les nouveaux utilisateurs.
  - Empêche l'accès à internet aux utilisateurs auquel leur délai de connexion est terminé.

- **Les cas d'utilisation RADIUS:**
  - Insert les informations d'authentifications et comptabilisations des utilisateurs dans la base de données.
  - Consulte la base de données.

**1-5-1 Spécification des cas d'utilisation :**

Les figures suivantes présentent une description de l'ensemble des cas d'utilisation de notre système.

**a. utilisateur :**

**Use case:** sélectionner le SSID.
**Scenario:** S0.
**Rôle:** utilisateur.
**Description:**
**1.** L'utilisateur détecte le point d'accès.
**2.** il ouvre le gestionnaire de point d'accès.
**3.** il sélectionne le SSID.

**Figure III.1:** Spécification de cas d'utilisation «sélectionner le SSID».

**Use case:** consulter la pub et accéder a la page d'authentification.
**Scenario:** S01, S02, S03.
**Rôle:** utilisateur.
**Description:**
**1.** L'utilisateur ouvre un navigateur web.
**2.** Apparition de la fiche publicitaire.
**3.** Apparition de la page d'authentification une la fois la pub est terminée.

**Figure III.2:** Spécification de cas d'utilisation «consulté la pub et accéder a la page d'authentification».

**Use case:** consulter la page termes et conditions.
**Scenario:** S01, S02, S04.
**Rôle:** utilisateur.
**Description:**
**1.** sélectionner le lien termes et conditions.
**2.** Apparition de la page qui contient l'ensemble des termes et conditions d'utilisations

**Figure III.3:** Spécification de cas d'utilisation «accéder à la page terme et condition».

**Use case:** accepter les termes.
**Scenario:** S01, S02, S03, S05.
**Rôle:** utilisateur.
**Description:**
**1.** La page d'authentification apparait à l'utilisateur.
**2.** Il clique sur accepter les termes et les conditions.
**3.** L'utilisateur accède à internet.

**Figure III.4:** Spécification de cas d'utilisation «accepté les termes».

**b. portail captif :**

**Use case:** autoriser l'accès à internet pour les nouveaux utilisateurs.
**Scenario:** S06, S07.
**Rôle:** portail captif.
**Description:**
**1.** Dés qu'un un nouveau utilisateur se connecte, le portail captif récupère les informations d'authentification.
**2.** Le portail captif envoie les informations d'authentification et de comptabilisation au serveur Radius.

**Figure III.5:** Spécification de cas d'utilisation «autoriser l'accès à internet pour les nouveaux utilisateurs».

**Use case:** empêcher l'accès à internet aux utilisateurs auxquels leur délai de connexion autorisé est terminé.
**Scenario:** S06, S07.
**Rôle:** portail captif.
**Description:**
**1.** le portail captif envoie les informations d'authentification au serveur radius.
**2.** le portail
**3.** le portail captif coupe la connexion dès que le temps de connexion autorisé est atteint.

**Figure III.6 :** Spécification de cas d'utilisation «empêche l'accès à internet aux utilisateurs auquel leur délai de connexion est terminé».

**c. RADIUS :**

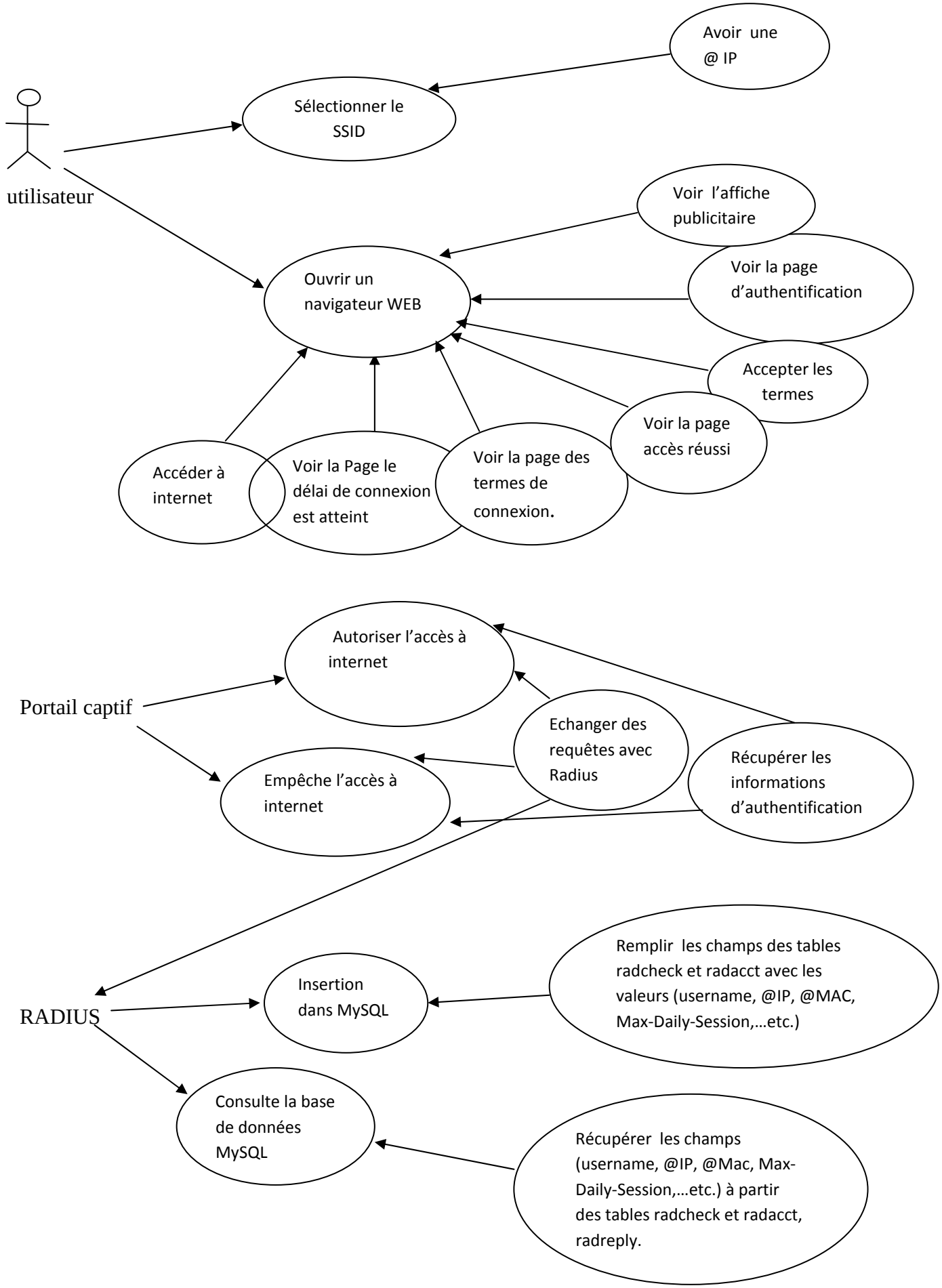
**Use case:** insérer les utilisateurs dans la base de données.
**Scenario:** S08.
**Rôle:** RADIUS.
**Description:**
**1.** Radius reçoit les informations d'authentification et de comptabilité (username, @IP, @MAC, Max-Daily-Session,…etc.) d'un utilisateur.
**2.** Radius insert les champs (username, @IP, @MAC, Max-Daily-Session,…etc.) dans la table radcheck de la base de données.

**Figure III.7:** Spécification de cas d'utilisation «insertion des utilisateurs dans la base de données».

**Use case:** Récupérer les informations des utilisateurs la base de données MySQL.
**Scenario:** S09.
**Rôle:** RADIUS.
**Description:**
**1.** Radius récupère les informations de connexion de l'utilisateur à partir des tables radacct et radreply (exp : username, temps de connexion autorisé, @Mac, …etc).

**Figure III.8:** Spécification de cas d'utilisation «Récupérer les informations des utilisateurs la base de données MySQL».

### 1-5-2- Diagramme des cas d'utilisation général :



Nous allons dans ce qui suit présenter les diagrammes de séquence de quelques cas d'utilisation à savoir :

Cas d'utilisation : se connecté à internet.
Cas d'utilisation : le délai de connexion est atteint.

**a. utilisateur :**

Se connecté à internet**:**

Ce cas d'utilisation contient les objets suivants :

- **Objet interface :**
  - Fenêtre gestionnaire de point d'accès.
  - Page publicitaire.
  - Page d'authentification.
  - Page de confirmation de l'accès.
  - Page accès réussi.
  - Page web demandée (internet).
- **Objet entités :**
  - Serveur MySQL.
  - Serveur web.
- **Objet contrôle :**
  - Portail captif.
  - Radius.

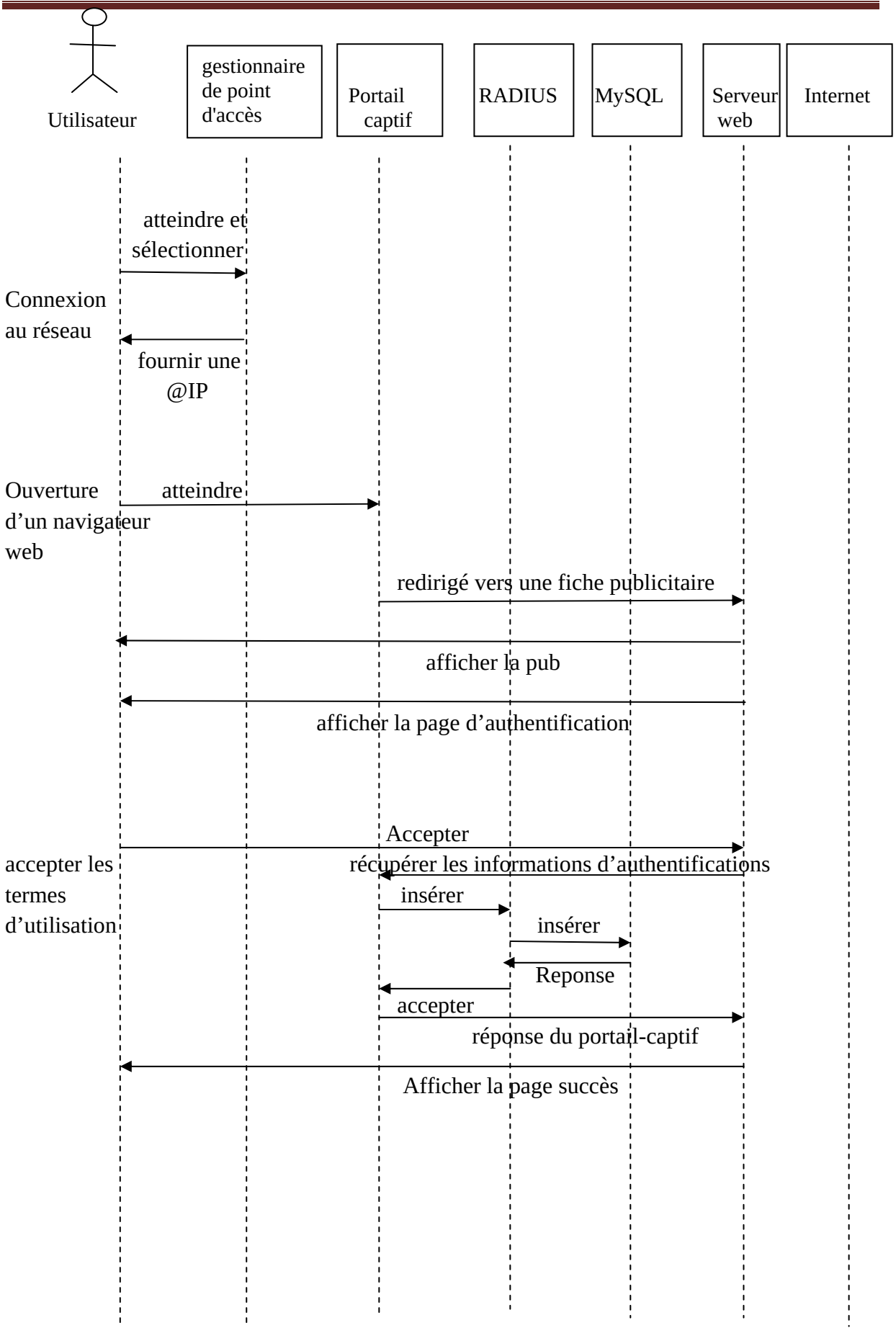



**Figure III.9:** Diagramme de séquence du cas d'utilisation « accès à internet».

**Cas d'utilisation «** le délai de connexion est atteint**» :**
Ce cas d'utilisation contient les objets suivants :

- **Objet interface :**
  - Page publicitaire.
  - Page d'authentification.
  - Page « le temps permet est atteint »
  - Page web demandée (internet).
- **Objet entités :**
  - Serveur MySQL.
  - Serveur web.
- **Objet contrôle :**
  - Portail captif.
  - Radius.

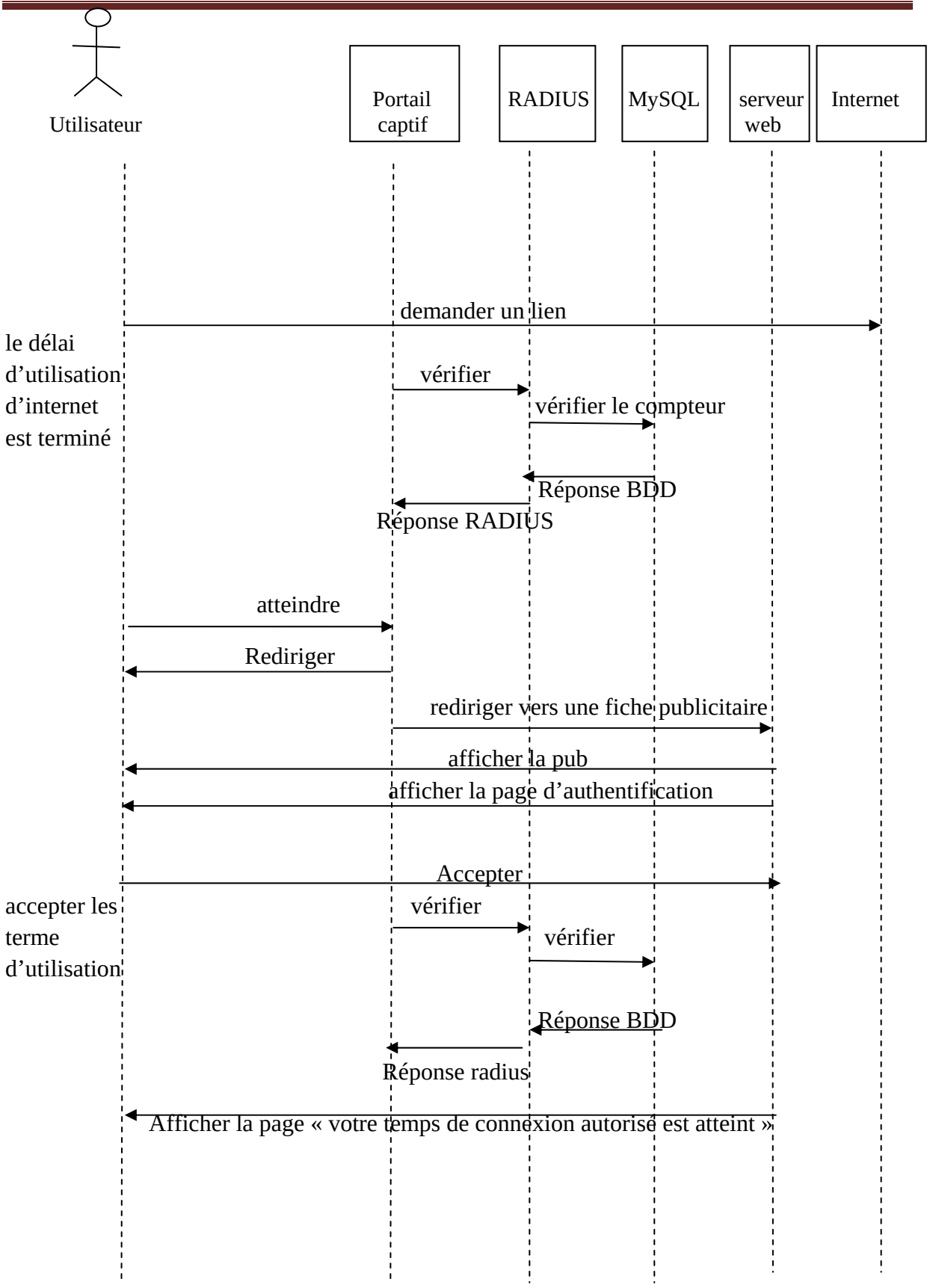
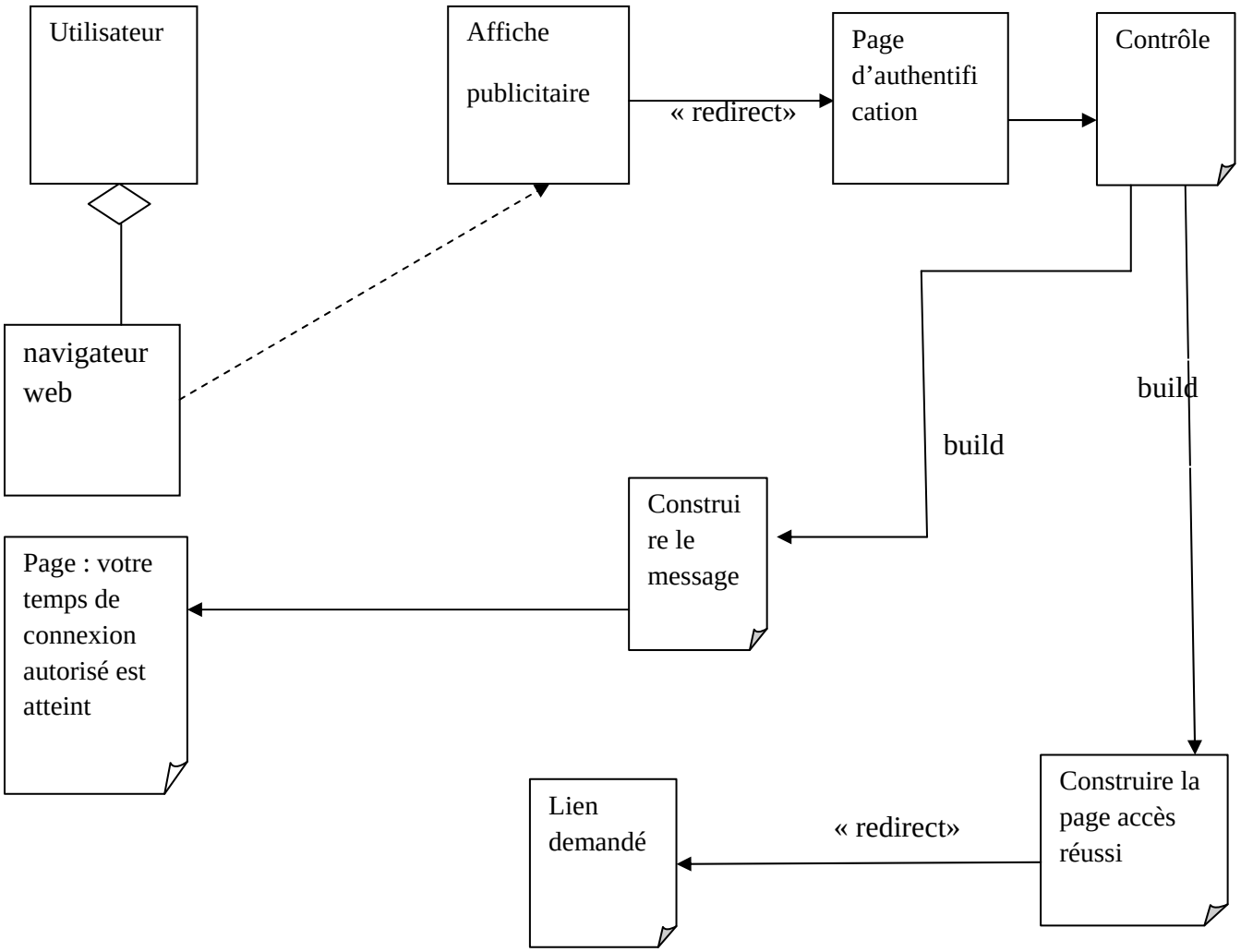



**Figure III.10 :** Diagramme de séquence du cas d'utilisation «le délai de connexion est atteint ».

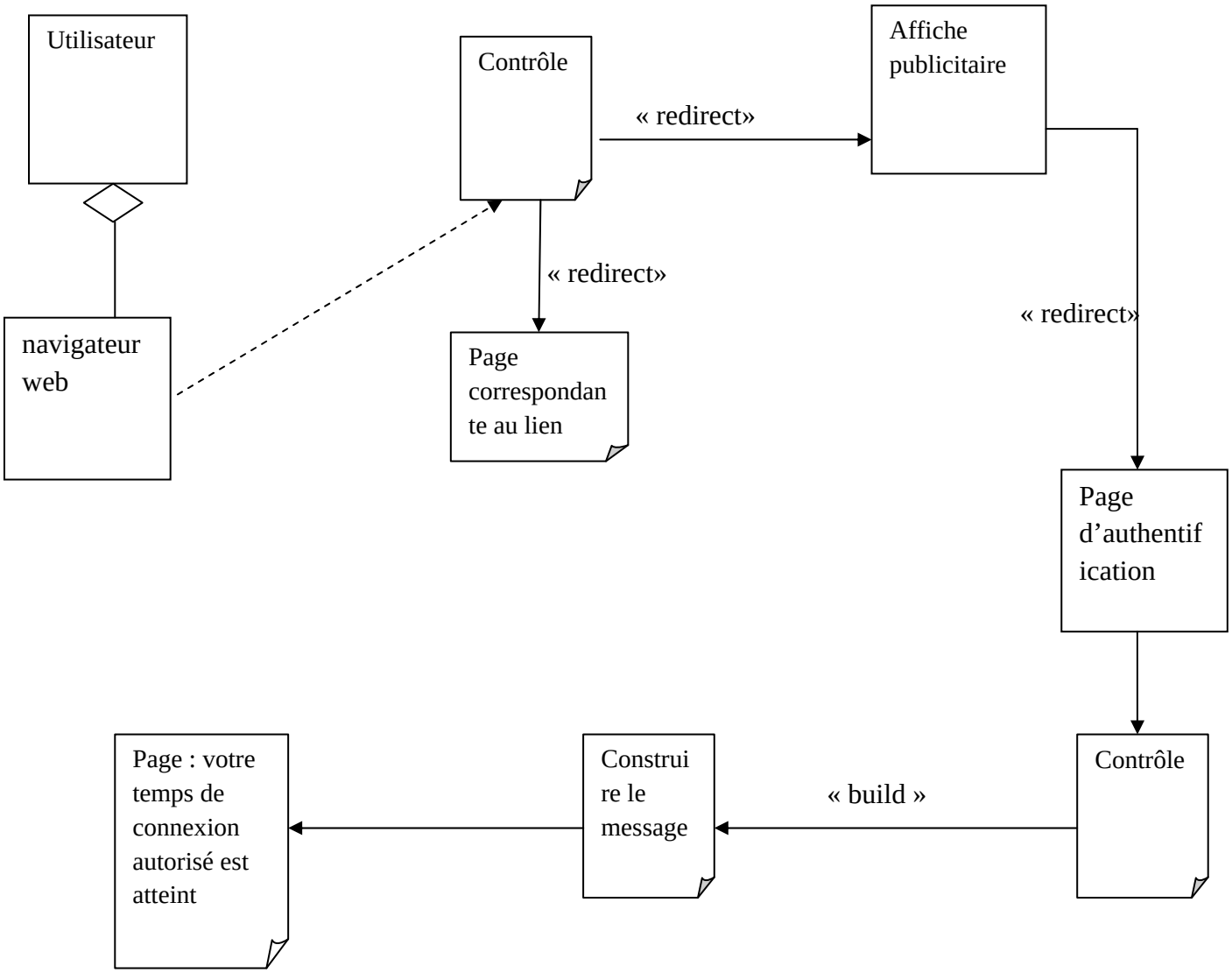
**II- Conception :**

Après avoir élaboré les différents diagrammes de séquence de cas d'utilisation, nous allons passer à l'étape de conception qui est consacrée essentiellement à la réalisation des cas d'utilisation à travers les diagrammes de classe des cas d'utilisation étudier.

**A- Diagramme de classe cas « accès internet » :**



**B- Diagramme de classe cas « le délai de connexion est atteint » :**



### III- La base de données : radius

Nous pouvons voir sur la figure ci-dessous la représentation des différentes tables de Radius. Les principales tables utilisées lors de l'authentification sont "radcheck" et "radacct". Nous verrons donc en détail leurs spécificités.

```
mysql> show tables;
+------------------+
| Tables_in_radius |
+------------------+
| nas              |
| radacct          |
| radcheck         |
| radgroupcheck    |
| radgroupreply    |
| radpostauth      |
| radreply         |
| radusergroup     |
+------------------+
8 rows in set (0.00 sec)

mysql>
```

**Figure III.11:** Tables Radius.

La table "radcheck" décrite ci-dessous montre comment les identifiants de session sont stockés.

La colonne "username" indique le nom d'utilisateur stocké dans la base, "attribute" spécifie un attribut de contrôle tandis que "value" correspondant à la valeur de l'attribut : dans l'exemple situé Figure **, nous avons l'attribut de contrôle "Crypt-Password" qui correspond à l'empreinte du mot de passe de l'utilisateur. Le mot de passe est hashé en utilisant l'algorithme MD5(salted).

```
mysql> desc radcheck;
+-----------+------------------+------+-----+---------+----------------+
| Field     | Type             | Null | Key | Default | Extra          |
+-----------+------------------+------+-----+---------+----------------+
| id        | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| username  | varchar(64)      | NO   | MUL |         |                |
| attribute | varchar(64)      | NO   |     |         |                |
| op        | char(2)          | NO   |     | ==      |                |
| value     | varchar(253)     | NO   |     |         |                |
+-----------+------------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

**Figure III.12:** Table des attributs de contrôle Radius.

La table "radacct" regroupe les données de comptabilité des utilisateurs. On y retrouve par exemple l'adresse IP de l'utilisateur, l'heure de début et de fin de chacune de ces sessions sur le portail captif, les données consommées en envoi et en réception.

```
mysql> desc radacct;
+----------------------+-------------+------+-----+---------+----------------+
| Field                | Type        | Null | Key | Default | Extra          |
+----------------------+-------------+------+-----+---------+----------------+
| radacctid            | bigint(21)  | NO   | PRI | NULL    | auto_increment |
| acctsessionid        | varchar(64) | NO   | MUL |         |                |
| acctuniqueid         | varchar(32) | NO   | MUL |         |                |
| username             | varchar(64) | NO   | MUL |         |                |
| groupname            | varchar(64) | NO   |     |         |                |
| realm                | varchar(64) | YES  |     |         |                |
| nasipaddress         | varchar(15) | NO   | MUL |         |                |
| nasportid            | varchar(15) | YES  |     | NULL    |                |
| nasporttype          | varchar(32) | YES  |     | NULL    |                |
| acctstarttime        | datetime    | YES  | MUL | NULL    |                |
| acctstoptime         | datetime    | YES  | MUL | NULL    |                |
| acctsessiontime      | int(12)     | YES  | MUL | NULL    |                |
| acctauthentic        | varchar(32) | YES  |     | NULL    |                |
| connectinfo_start    | varchar(50) | YES  |     | NULL    |                |
| connectinfo_stop     | varchar(50) | YES  |     | NULL    |                |
| acctinputoctets      | bigint(20)  | YES  |     | NULL    |                |
| acctoutputoctets     | bigint(20)  | YES  |     | NULL    |                |
| calledstationid      | varchar(50) | NO   |     |         |                |
| callingstationid     | varchar(50) | NO   |     |         |                |
| acctterminatecause   | varchar(32) | NO   |     |         |                |
| servicetype          | varchar(32) | YES  |     | NULL    |                |
| framedprotocol       | varchar(32) | YES  |     | NULL    |                |
| framedipaddress      | varchar(15) | NO   | MUL |         |                |
| acctstartdelay       | int(12)     | YES  |     | NULL    |                |
| acctstopdelay        | int(12)     | YES  |     | NULL    |                |
| xascendsessionsvrkey | varchar(10) | YES  |     | NULL    |                |
+----------------------+-------------+------+-----+---------+----------------+
26 rows in set (0.00 sec)
```

**Figure III.13:** Table des données de journalisation Radius.

## Conclusion :

Nous avons réalisé dans ce chapitre la conception de notre hotspot aidés par le modèle UML. Cette dernière est nécessaire et très importante pour pouvoir créer notre Hotspot. Dans le chapitre qui suit, nous détaillerons les étapes de mise en œuvre de notre Hotspot par la présentation des différentes installations et configurations des outils utilisées (Raspberry-pi, Portail captif, Serveur d'authentification, …etc.).

Dans le chapitre suivant nous allons présenter les différents logiciels utilisés, leurs installations et leurs configurations afin de mettre de notre Hotspot en service.

# CHAPITRE N°4

## Introduction :

Ce chapitre vise à fournir un guide de déploiement de notre HotSpot, en tenant compte de l'étude faite dans les chapitres précédents.

Nous détaillerons dans ce chapitre, les interventions nécessaires au niveau des nœuds réseau (point d'accès, portail captif, serveurs, …). Ces interventions sont :

- Le Choix des équipements.
- La Topographie utilisées.
- Configuration de la Raspberry-Pi.
- Configuration du portail captif et du serveur web apache2.
- Configuration du serveur d'authentification Freeradius et du serveur Mysql.
- Configuration du point d'accès.
- Personnalisation des pages web HTML.

## 1- Identification des composants matériels :

### 1-1- Présentation de la carte électronique RASPBERRY-PI :

Le **Raspberry-Pi** est un nano-ordinateur monocarte à processeur ARM conçu par le créateur de jeux vidéo David Braben, dans le cadre de sa fondation Raspberry Pi.

Cet ordinateur, qui a la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique, il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux et des logiciels compatibles. Il est fourni nu (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

- **Caractéristiques du Raspberry-Pi :**

  - ✓ Processeur ARM11 avec une fréquence 700MHz.
  - ✓ RAM 512Mo.
  - ✓ Carte mémoire (Micro SD) d'une capacité de 16Go.
  - ✓ Cartes réseau (Ethernet et USB).

### 1-2- Point D'accès sans-fil :

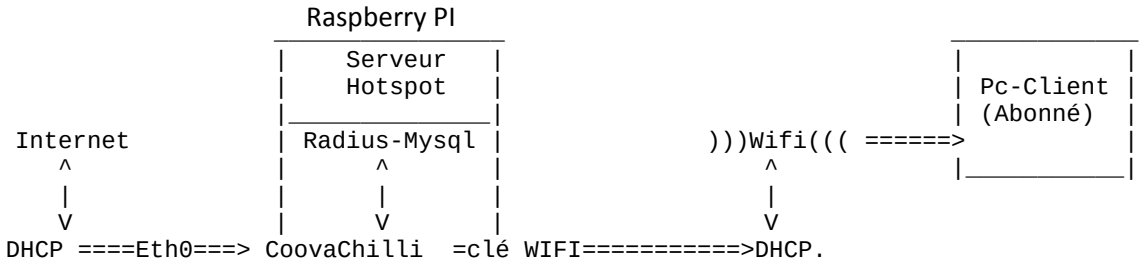
Une clé (dongle) modele TP-LINK TL-WN722N ayant les caractéristiques suivantes :

- Standard 802.11b/g, 2.4GHz, (150Mbps).
- Compatible avec les équipements 802.11b/g sans fil.
- Supporte le mode master (point d'accès).
- Port USB 2.0.
- Encodage WEP, WPA/WPA2, WPA/PSK/WPA2-PSk.
- Technologie N procurant une vitesse jusqu'à 1500 Mbps pour la transmission sans fil


**Figure IV.1:** TP-LINK TL-WN722N.

## 3- Topologie utilisée :

Le schéma suivant représente l'architecture de notre réseau :




## 4- Configuration matériels:

Avant de décrire les étapes de configuration, nous donnons dans les deux sections suivantes un aperçu des logiciels utilisées pour la configuration

### 4-1- Configuration de la Raspberry-Pi :

#### 4-1-1- Définition de Win32DiskImager :

Ceci est un programme Windows pour sauvegarder et restaurer des images de disques amovibles (clés USB, les cartes mémoire SD, etc.). Il peut être utilisé pour écrire les images de démarrage

(ubuntu-12.04-préinstallé-desktop-armhf + omap4.img ) dans un dispositif Flash SD ou un périphérique USB à mémoire flash et le rendre amorçable.

### 4-1-2- Définition du système d'exploitation RASPBIAN :

Raspbian est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry Pi. Raspbian fournit plus qu'un simple système d'exploitation: il est livré avec plus de 35.000 paquets, des logiciels pré-compilés livré dans un format adapté pour une installation facile sur votre Raspberry Pi.

Dans un premier temps nous avons besoin d'une Raspberry-Pi, de son alimentation, d'une carte mémoire avec Raspbian gravé dessus. Nous avons également besoin, pour des fins de configuration, d'un clavier, d'une souris, et enfin d'un écran.

### 4 1-3- Premiers démarrages de la Raspberry-Pi:

En premier lieu, Nous avons branché la Raspberry-Pi à l'écran, au clavier et à une sourie, sans l'alimenter, et nous avons installé la carte SD. Nous avons ensuite branché l'alimentation de la Raspberry-Pi. Le premier démarrage peut être long, car la Raspberry-Pi va installer le système Raspbian. Au cours de ce démarrage (et des suivants), des commendes spécifiques sont exécutées par le système afin de bien démarrer.

Si la led en vert de la Raspberry-Pi clignote au début du démarrage, nous concluons que le système est bien installé sur la carte et qu'elle a bien démarré sur l'écran. Comme tout système Linux, un login et un mot de passe sont demandés. Par défaut ; le login est « «pi », et le password est « raspberry ».

### 4-1-4- Configuration de Raspbian :

Lors du premier démarrage du système Raspbian une interface permettant de faire des choix de configuration apparait. Cette interface se présente sous forme d'un menu navigable au clavier.

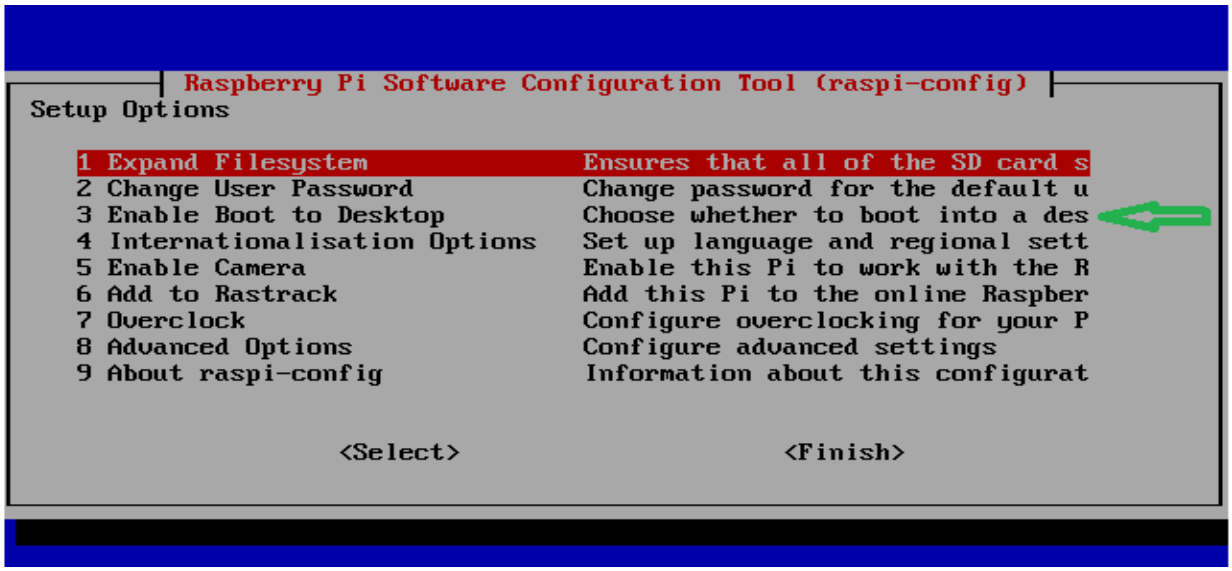



**Figure IV.2:** Le Menu de configuration da la Raspberry PI.

Nous donnons dans ce qui suit les choix que nous avons adopté pour configurer notre carte

- « **Expand Filesystem** » pour utiliser la totalité de la carte
- « **Change User Password** »**:** Pour changer le mot de passe de la Raspberry.
- « **Internationalisation options** » **:** Pour mettre Raspbian en français et le clavier en AZERTY**.**
- **Enable boot to desktop :** Pour passer de la console à l'interface graphique.

### 4-2- Configuration des interfaces réseaux et activation du routage sous linux :

#### 4-2-1- Configuration des interfaces réseau :

Pour configurer les deux interfaces réseau nous avons édité le fichier « **interfaces** » avec la commande **:**

```
# nano /etc/network/interfaces
```

Puis nous l'avons rempli avec :

```
allow-hotplug eth0
auto eth0                          } Interface eth0 qui est reliée au modem (WAN)
iface eth0 inet dhcp

allow-hotplug wlan0
 iface wlan0 inet static
     address 10.0.2.1              } Interface wlan0 avec l'@ Ip 10.0.2.1 qui sera l'@du NAS (LAN)
     netmask 255.255.255.0
     network 10.0.2.1
     broadcast 10.0.2.255
```

**Figure IV.3:** Fichier de configuration « interfaces ».

#### 4-2-2- activation du forward :

Pour activer le routage dans les systèmes d'exploitation Linux, il suffit d'éditer le fichier /etc/sysct et de changer la valeur de la ligne suivante :

```
net.ipv4.ip_forward = 1 (la valeur par défaut est 0)
```

### 4-3- Compilation et configuration du Portail Captif (Coova-chilli 1.3.0):

➢ **Rappel sur la fonctionnalité principale de Coova-chilli :** Il permet de rediriger tous les clients HTTP d'un réseau vers une page web qui peut demander une authentification et/ou un paiement ou tout simplement demander d'accepter les conditions d'utilisation avant d'accéder à Internet. Cette technique est souvent employée pour les accès Wi-Fi et peut être utilisée aussi pour l'accès à des réseaux filaires (ex. hôtels, campus etc.).

### 4-3-1- Les outils nécessaires pour la compilation de Coovachilli 1.3.0 sur une architecture ARM-11 :

- **Debhelper :** Ensemble de programmes que l'on peut utiliser dans un fichier debian/rules pour automatiser des tâches intervenant fréquemment dans la création de paquets Debian. Des programmes sont fournis pour installer divers fichiers dans notre paquet, compresser des fichiers, corriger les droits d'accès, intégrer le paquet au système de menu Debian, debconf, doc-base, etc. La plupart des paquets Debian utilisent Debhelper au cours de leur création.

- **Libssl-dev :** Ce paquet fournit les bibliothèques de développement pour libssl et libcrypto, ce qui inclut les fichiers d'en-têtes ainsi que les pages de manuel. Il fait partie de l'implémentation OpenSSL de SSL.

- **libcurl4-gnutls-dev :** Libcurl est une bibliothèque de transfert d'URL du côté client facile à utiliser, supportant les protocoles DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET and TFTP. Il supporte aussi SSL certificats, HTTP POST, HTTP PUT, FTP uploading, HTTP form upload, proxies, cookies, user + password Authentication (Basic, Digest, NTLM, Negotiate, Kerberos), http proxy tunneling …etc. C'est un package gratuit, très riche en fonctionnalités, compatible avec l'IPv6, bien soutenu, bien documenté. il est utilisé par de nombreuses grandes entreprises.

Pour installer ces outils il faut taper la commande suivante

```
$ sudo apt-get install debhelper libssl-dev libcurl4-gnutls-dev
```

Une fois le terrain est préparé pour la compilation de coovachilli -1.3.0. Nous avons suivi les étapes suivantes :

✓ Nous nous sommes déplacé au répertoire **/usr /src**/ et télécharger le fichier coova-1.3.0, on tapant les commandes

```
# cd /usr/src

# wget http://ap.coova.org/chilli/coova-chilli-1.3.0.tar.gz »
```

✓ Nous avons réalisé un test pour assurer que le paquet a était complètement télécharger (i.e pas de fichiers manquants). Le test a était fait on tapant la commande

```
$ sha256sum coova-chilli-1.3.0.tar.gz
```

Puis le résultat a était comparé à la valeur suivante :
**ca24ac274340c65a8e7ff704e6866a04380a87c444f261ba84097f0bd1c162e8**,

Après avoir obtenu le même hash, nous avons confirmé que le paquet a était bien télécharger et nous nous sommes rendu au répertoire /usr/src/ via la commande :

```
# cd /usr/src
```

Dans le but de décompresser le paquet, on tape la commande :

```
/usr/src # tar xzf coova-chilli-1.3.0.tar.gz
```

Cela nous a crée un répertoire nommé « coova-chilli-1.3.0 », nous nous somme rendu a l'intérieur via la commande :

```
/usr/src # cd /coova-chilli-1.3.0
```

Nous nous plaçons dans le répertoire covachilli en étant dans le mode utilisateur

```
$ cd /usr/src/coova-chilli-1.3.0
```

Une fois sur le répertoire de Coova nous avons configuré les fichiers source via la commande :

```
./configure  --prefix=/usr
```

Après nous avons modifié le fichier /usr/src/coova-chilli-1.3.0/debian/rules, en l'éditant via la commande :

```
$ sudo nano /usr/src/coova-chilli-1.3.0/debian/rules
```

Puis nous nous sommes déplacé à la ligne 54 et nous avons modifié le contenu de la ligne qui a était :

```
$(MAKE) DESTDIR=$(CURDIR)/debian/tmp install
```

Par :

```
$(MAKE) DESTDIR=/ install
```

Cette commande nous a assuré que les fichiers nécessaires sont mis dans le répertoire /etc/chilli/ et pas dans le répertoire indiqué à la ligne originale pour éviter des erreurs.

Acette étape nous avons compilé le code source via la commande :

```
/usr/src/coova-chilli-1.3.0 $ sudo dpkg-buildpackage –us –uc
```

Le résultat de compilation est un fichier.deb placé dans le répertoire **/usr/src .** Il est appelé : **coova-chilli_1.3.0_armhf.**

Ssur le répertoire **/usr/src/,** nous avons tapé la commande suivante pour installer coova-chilli_1.3.0_armhf sur la Raspi.

```
/usr/src $ sudo dpkg –i coova-chilli_1.3.0_armhf.deb
```

**4-3-2- Configuration de Coova-chilli :**

Après avoir compilé et installé coova-chilli sur la Raspberry-Pi nous somme passé à sa configuration réelle, en passant par ces étapes :

- **Etape-1 :** Nous avons édité le fichier /etc/defaut/chilli et modifié la première ligne « **START-CHILLI »** Nous avons mis a valeur à 1 afin de lancer coova-chilli sans rebooter. Nous avons tapé la commande suivant

Pour démarrer le service covachilli, on tape :

```
# /etc/init.d/chilli start
```

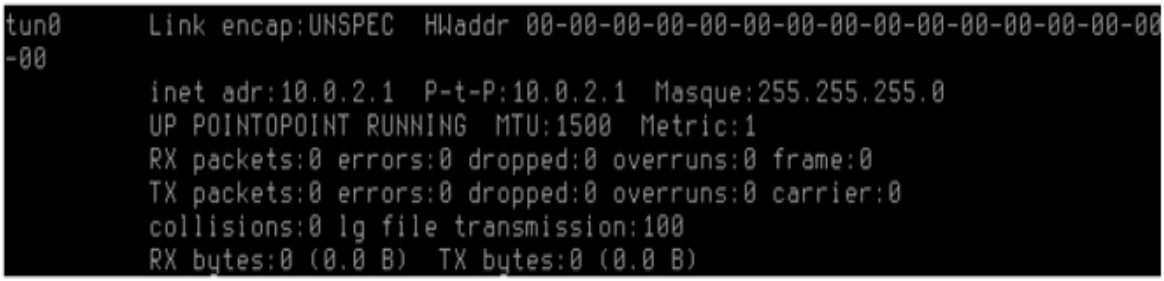



**Figure IV.4:** L'interface tun0-00.

- **Etape-2 :** Nous avons copié le fichier /etc/chilli/default à /etc/chilli/config via la commande :

```
cp /etc/chilli/default /etc/chilli/config
```

Comme « config » est le fichier principal du portail captif, nous avons étudié la signification de chaque ligne afin de pouvoir mené à bien notre configuration.

```
HS_WANIF=eth4            # WAN Interface toward the Internet
HS_LANIF=eth5              # Subscriber Interface for client devices
HS_NETWORK=10.0.2.0        # HotSpot Network (must include HS_UAMLISTEN)
HS_NETMASK=255.255.255.0   # HotSpot Network Netmask
HS_UAMLISTEN=10.0.2.1      # HotSpot IP Address (on subscriber network)
HS_UAMPORT=3990            # HotSpot UAM Port (on subscriber network)
HS_UAMUIPORT=4990          # HotSpot UAM "UI" Port (on subscriber network, for S
# OpenDNS Servers
HS_DNS1=10.0.2.1
HS_DNS2=10.0.2.1
HS_NASID=nas01
HS_RADIUS=127.0.0.1
HS_RADIUS2=127.0.0.1
HS_UAMALLOW=10.0.2.0/24
HS_RADSECRET=radiussecret    # Set to be your RADIUS shared secret
HS_UAMSECRET=uamsecret     # Set to be your UAM secret
HS_UAMALIASNAME=chilli
#   The server to be used in combination with HS_UAMFORMAT to
#   create the final chilli 'uamserver' url configuration.
HS_UAMSERVER=$HS_UAMLISTEN

#   Use HS_UAMFORMAT to define the actual captive portal url.
#   Shell variable replacement takes place when evaluated, so here
#   HS_UAMSERVER is escaped and later replaced by the pre-defined
#   HS_UAMSERVER to form the actual "--uamserver" option in chilli.
HS_UAMFORMAT=http://\$HS_UAMLISTEN:\$HS_UAMUIPORT/www/login.chi

#   Same principal goes for HS_UAMHOMEPAGE.
HS_UAMHOMEPAGE=http://\$HS_UAMLISTEN:\$HS_UAMPORT/www/coova.html
#   This option will be configured to be the WISPr LoginURL as well
#   as provide "uamService" to the ChilliController. The UAM Service is
#   described in: http://www.coova.org/CoovaChilli/UAMService
#
HS_UAMSERVICE=https://10.0.2.1/cgi-bin/hotspotlogin.cgi
HS_TCP_PORTS="81"

HS_LOC_NAME="My Hotspot"          # WISPr Location Name and used in portal
```

**Figure IV.5:** Le fichier de configuration de Coova.

Explication du fichier de configuration de Coova-chilli:

**HS_WANIF=eth1:** Interface menant à Internet
**HS_LANIF=wlan0 :** Interface reliée aux clients
**HS_NETWORK=10.0.2.0 :** L'adresse réseau du Hotspot
**HS_NETMASK=255.255.255.0 :** Le masque réseau du Hotspot
**HS_UAMLISTEN=10.0.2.1 :** L'adresse IP du Hotspot
**HS_UAMPORT=3990 :** C'est le port TCP qui est lié à l'authentification des clients, si un client non authentifié tente de se rendre sur internet, il sera redirigé sur ce port-là, à l'adresse IP spécifiée plus haut.
**HS_UAMUIPORT=4990 :** Port permettant de contacter le portail captif

**HS_DNS1=10.0.2.1 :** L'adresse IP du DNS
**HS_DNS2=10.0.2.1 :** Puisqu'on utilise un seul DNS, on met le même ici
**HS_NASID=nas01 :** L'identifiant du serveur d'accès réseau, par défaut nas0
**HS_RADIUS=127.0.0.1 :** On marque ici l'adresse IP du serveur d'authentification
**HS_RADIUS2=127.0.0.1 :** On réécrit la même chose
**HS_UAMALLOW=10.0.2.0/24 :** On reprend ici l'adresse réseau et le masque
**HS_RADSECRET=salim :** On précise le mot de passe qu'on avait choisi précédemment
**HS_UAMSECRET=uamsecret :** Secret partagé en le serveur UAM et Chilli.
**HS_UAMALIASNAME=chilli :** Alias pour les services UAM
**HS_UAMSERVER=$HS_UAMLISTEN :** URL du serveur WEB à utiliser pour authentifier les clients (ici10.0.2.1)
**HS_UAMFORMAT=http://\$HS_UAMLISTEN:\$HS_UAMUIPORT/www/login.chi :** Définit l'url du portail captif
**HS_UAMHOMEPAGE=http://\$HS_UAMLISTEN:\$HS_UAMPORT/www/coova.html :** L'url de la page d'accueil pour les utilisateurs non authentifiés.
**HS_UAMSERVICE=https://10.0.2.1/cgi-bin/hotspotlogin.cgi :** Permet d'aller rechercher chez un utilisateur ses informations de connexion et donc lui donner la possibilité de passer d'un AP à un autre sans avoir besoin d'avoir à se connecter au portail captif.
**HS_TCP_PORTS= »81″ :** On ouvre un port, par exemple le 81 si on veut que les utilisateurs puissent s'inscrire eux-mêmes.
**HS_LOC_NAME= »My Hotspot » :** Permet de mettre un petit nom au Coova-Chilli.

- **Etape-3 :** Dans cette étape nous avons configuré le Firewall (pare-feu) en utilisant iptables. Coova-chilli est préconfiguré mais nous avons ajouté dans le fichier /etc/chilli/up.sh via la commande :

```
$ sudo nano /etc/chilli/up.sh
```

la ligne suivante à la fin du fichier:

```
iptables –I POSTROUTING –t nat –o $HS_WANIF –j MASQUERADE
```

### 4-4- Installation et configuration du serveur web apache2 :

L'installation d'apache2 est très simple, nous l'avons réalisé via la commande suivante :

```
# apt-get install apache2
```

### 4-4-1- Les outils nécessaires pour configurer le serveur web apache2 :

- **un serveur web apache2 :**

Le logiciel libre **Apache HTTP Server** (**Apache**) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache. Apache fonctionne principalement sur les systèmes d'exploitation UNIX (Linux, Mac OS X, Solaris, BSD et UNIX) et Windows. La version Windows n'est considérée comme stable que depuis la version 1.2 d'Apache. Apache est utilisé par de nombreux produits, dont WebSphere d'IBM, ainsi que par Oracle Corporation.

➢ **Universal Access Method UAM (méthode d'accès universelle) :**

La méthode d'accès universelle (UAM) est fréquemment utilisée par les opérateurs WiFi pour permettre l'accès à un réseau sans fil ou l'accès à un autre réseau en itinérance. L'abonné itinérant utilise un navigateur Web pour accéder régulièrement une page de connexion sur le portail captif où il peut remplir ses qualifications (généralement son nom d'utilisateur et mot de passe) pour obtenir l'accès au réseau.

➢ **Définition du Commun Gateway interface (CGI) :**

Commun Gateway interface est une méthode standard utilisée pour générer et les applications Web. CGI, lorsqu'il est mis en œuvre sur un serveur Web, fournit une interface entre le serveur et les programmes qui génèrent le contenu Web. Ces programmes sont connus comme les « scripts CGI » ou tout simplement CGI, ils sont souvent écrits dans un langage de script, mais ils peuvent être écrits dans n'importe quel langage de programmation.

➢ **Définition du module Haserl :**

Haserl est un petit programme qui utilise des scripts shell pour créer des scripts web CGI. Il est destiné aux environnements où PHP ou Ruby sont trop gros. Il a été écrit pour Linux, mais il est connu pour son fonctionnent sur FreeBSD. Son utilisation permet d'exécuter des scripts CGI dans un environnement embarqué.

➢ **Définition du module SSL (Secure Sockets Layers) :**

SSL (Secure Sockets Layers, que l'on pourrait traduire par couche de sockets sécurisée) est un procédé de sécurisation des transactions effectuées via Internet. Le standard SSL a été mis au point par Netscape, en collaboration avec Mastercard, Bank of America, MCI et Silicon Graphics. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification.

Le système SSL est indépendant du protocole utilisé, ce qui signifie qu'il peut aussi bien sécuriser des transactions faites sur le Web par le protocole HTTP que des connexions via le protocole FTP,POP ou IMAP. En effet, SSL agit telle une couche supplémentaire, permettant d'assurer la sécurité des données, située entre la couche application et la couche transport (protocole TCP par exemple).

➢ **Définition du package libapache2-mod-auth-mysql :**

Est un module pour le serveur Web Apache2 qui permet l'authentification HTTP en se référant aux informations stockées dans une base de données MySQL.

Pour mettre en place Apache2, nous avons mis en place certains fichiers en tapant une série de commandes, voici un aperçu :

```
# cp /etc/chilli/defaults /etc/chilli/config

# mkdir /var/www/hotspot

# cd /var/www/hotspot

# cp /etc/chilli/www/* /var/www/hotspot

# mkdir /var/www/hotspot/images

# cp /var/www/hotspot/coova.jpg /var/www/hotspot/images/

# mkdir /var/www/hotspot/uam

# cd /var/www/hotspot/uam

# wget http://ap.coova.org/uam/

# wget http://ap.coova.org/js/chilli.js
```

En suite, nous avons placé le module CGI au bon endroit et nous avons rendu le script exécutable via les commandes :

```
# mkdir –p /var/www/hotspot/cgi-bin

# zcat –c /usr/share/doc/coova-chilli/hotspotlogin.cgi.gz | tee /var/www/hotspot/cgi-bin/hotspotlogin.cgi

# chmod a+x /var/www/hotspot/cgi-bin/hotspotlogin.cgi
```

Nous l'avons édité et modifié la ligne **$uamsecret= "ht2eb8ej6s4et3rg1ulp" (mot de passe échangé entre Coova et uam) via** la commande :

```
# nano /var/www/hotspot/cgi-bin/hotspotlogin.cgi
```

En suit, nous avons passé a l'installation du package libapache2-mod-auth-mysql, en tapant la commande :

```
# apt-get install libapache2-mod-auth-mysql
```

Puis nous avons créé le dossier qui contiendra le certificat via la commande :

```
# mkdir /etc/apache2/ssl
```

Note : Nous avons vérifié le nom de la machine avant de créer le certificat **# hostname –f** et nous avons eu « raspberry » comme nom de la machine.

Nous avons crée le certificat via la commande :

```
# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
```

Lorsque qu'il nous a demandé de taper le nom de la machine nous avons mis ce que nous avons eu par hostname (raspberry). Et nous avons ajouté le mode SLL à apache et le restart via les commandes :

```
# a2enmod ssl

# /etc/init.d/apache2 restart
```

Puis nous avons crée un site qu'on a nommé hotSpot via la commande :

```
# nano /etc/apache2/sites-available/hotspot
```

Et nous l'avons rempli comme suite :

```
NameVirtualHost 10.0.2.1:443

<VirtualHost 10.0.2.1:443>
ServerAdmin webmaster@domain.org
DocumentRoot "/var/www/hotspot"
ServerName "10.0.2.1"

<Directory "/var/www/hotspot/">
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

ScriptAlias /cgi-bin/ /var/www/hotspot/cgi-bin/

<Directory "/var/www/hotspot/cgi-bin/">
AllowOverride None
Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>

ErrorLog /var/log/apache2/hotspot-error.log
LogLevel warn
CustomLog /var/log/apache2/hotspot-access.log combined
ServerSignature On
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.pem
</VirtualHost>
```

**Figure IV.6:** Le contenue du fichier hotspot.

En suite nous l'avons activé par les commandes :

```
# a2ensite hotspot

# /etc/init.d/apache2 reload
```

Nous avons ensuite modifié le fichier **ports.conf** :

```
# nano /etc/apache2/ports.conf
```

Pour qu'il ne contiendra que ceci :

```
Listen *:443
Listen *:80
Listen *:81
```

En suite nous avons édité le fichier :

```
# nano /etc/apache2/sites-available/default
```

Et nous avons ajouté en première ligne :

```
NameVirtualHost *:80
```

Puis nous avons édité le fichier **apache2.conf** et nous avons ajouté à la fin la ligne suivante :

```
ServerName 10.0.2.1
```

Par la suite, nous avons édité le fichier **hosts** pour ajouter à la suite des deux premières lignes la ligne suivante :

```
10.0.2.1        Raspberry
```

A la fin de la configuration du serveur apache, nous avons redémarré apache et la machine via les commandes suivantes :

```
# /etc/init.d/apache2 restart

# reboot
```

Après avoir configuré le portail-captif Coova-chilli et le serveur apache nous sommes passé à l'installation puis la configurations du serveur d'authentification freeradius (radius).

### 4.5- Installation et configuration du serveur d'authentification Freeradius et sa base de données Mysql :

#### 4.5.1- Outils nécessaires pour la configuration Freeradius :

- **Définition du serveur d'authentification freeradius :**

Rappelons que FreeRADIUS est un serveur RADIUS libre, hautement riche en modules et en fonctionnalités. Il est considéré comme le serveur RADIUS le plus utilisé dans le monde,

compatible à la fois avec des systèmes embarqués et des systèmes multi utilisateurs. Il intègre depuis 2001, les modules pour les bases de données externes LDAP et MySQL.

FreeRADIUS est très utilisé par les fournisseurs d'accès internet pour authentifier leurs clients et leur offrir des adresses IP.

- **Définition de Mysql server :**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

Son nom vient du prénom de la fille du co-créateur Michael Widenius, My. SQL fait allusion au Structured Query Language, le langage de requête utilisé.

- **Définition de PHP :**

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet comme C++.

PHP a permis de créer un grand nombre de sites web célèbres, comme **Facebook, YouTube, Wikipedia**, etc. Est aujourd'hui considéré comme la base de la création des sites Internet dits dynamiques.

### 4.5.2- Installation du serveur d'authentification :

Pour installer le serveur d'authentification freeradius et sa base données mysql nous avons tapé dans un terminal le commande suivante :

```
# apt-get install freeradius freeradius-mysql mysql-server php5
```

**Note1 :** le paquet freeradius-mysql et un intermédiaire entre freeradius et mysql, il est utilisé pour authentifier et comptabiliser les utilisateurs.

**Note2 :** Lors de l'installation un mot de passe mysql nous a était demandé, nous avons tapé salim0409*. (Mot de passe pour accéder au serveur mysql).

### 4.5.3- Configuration de Freeradius :

Dans un premier temps nous avons crée une base de données qu'on nommée « radius » via les commandes suivantes :

```
# mysql –u root –p
Enter password : mysqladminsecret
mysql > CREATE DATABASE radius;
mysql > quit
```

Après ça, nous avons importé les tables crées par les fabricants de Freeradius à la base de données « radius » via les commandes suivantes :

```
# mysql –u root –psalim0409*  radius </etc/freeradius/sql/mysql/schema.sql

# mysql –u root –psalim0409*  radius < /etc/freeradius/sql/mysql/nas.sql
```

Puis nous avons ajouté un utilisateur et nous avons lui donné les privilèges sur la base de données « radius ».  Cet utilisateur est appelé : « radius ».

```
# mysql –u root –p

Enter password : mysqladminsecret

mysql > GRANT ALL PRIVILEGES ON radius.* TO 'radius'@'localhost' IDENTIFIED BY'mysqlsecret';

mysql > FLUSH PRIVILEGES;

mysql > quit
```

En suite, pour configurer Freeradius nous avons fait des modifications dans les 4 fichiers suivants : **radiusd.conf, sql.conf, et default**. Les deux premiers se trouvent tous dans **/etc/freeradius**, le dernier dans **/etc/freeradius/sites-available/default**


**Figure IV.7:** L'emplacement des fichiers de configuration de Freeradius.

Dans un premier temps, on s'est rendu dans le fichier sql.conf via la commande :

```
# nano /etc/freeradius/sql.conf
```

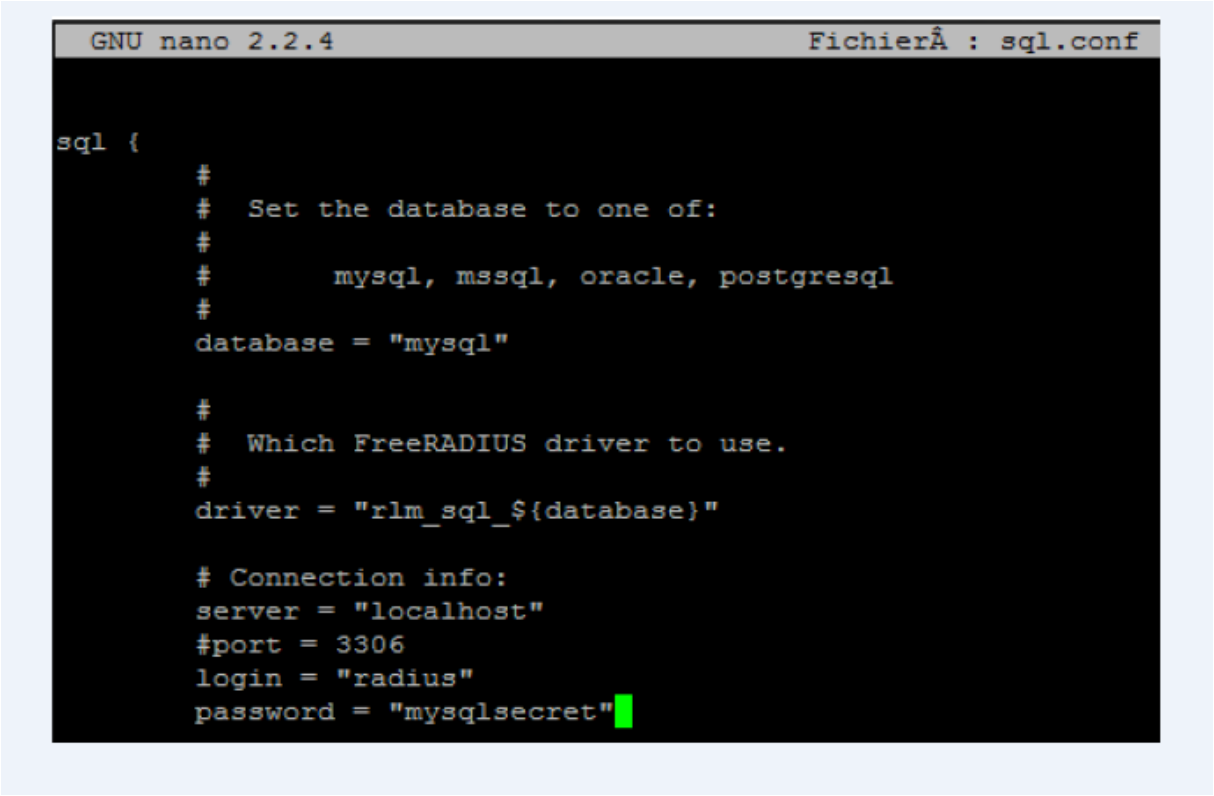



**Figure IV.8:** représente le contenu du fichier sql.conf.

Nous avons modifié la ligne « **password** », puisqu'on a choisi « salim0409* »comme mot de passe pour radius. Et nous avons laissé les autres lignes comme elles sont :

**database = « mysql »** Définit le type de base de données (mysql, mssql, oracle, postgresql) **server = « localhost »** Définit l‟ip du serveur sql (ici localhost puisque nous sommes sur le même serveur).

**login = « radius »** Définit le login permettant de se connecter au serveur sql Et on décommente la ligne **readclient = yes.**

Puis nous avons édité le fichier **radiusd.conf** via la commande suivante :

```
# nano /etc/freeradius/radiusd.conf
```

Et nous avons decommenté les lignes suivantes :

**$INCLUDE sql.conf**

**$INCLUDE sql/mysql/counter.conf (**module nécessaire pour réaliser le comptage **)**

Et en fin, on s'est rendu au fichier **default** via la commande **:**

```
# nano /etc/freeradius/sites-available/default
```

Dans ce fichier, on a retrouvé de nombreux paragraphes. Ceux qui nous intéressent sont les suivants : « **authorize** », « **accounting** » et « **session** ». Il nous a fallu decommenté dans ces paragraphes les lignes « **sql** » (afin que freeradius sache qu''on va utiliser sql). Dans la section

authorize nous avons ajouté au dessous de la ligne « **sql** » les lignes **noresetcounter, dailycounter** et **monthlycounter** et dans la section « **accounting** » nous avons decommenté la ligne « **Daily** » afin d'activé le comptage de Freeradius.

Une fois Freeradius est configuré, nous avons ajouté un utilisateur à la table radcheck(table dans laquelle sont enregistrés les utilisateurs ) via la commande suivante :

```
# echo "INSERT INTO radcheck(UserName, Attribute, Value) VALUES ("usercheck", « "Password", "passcheck"); " | mysql –u radius –psalim0409* radius
```

Puis nous avons redémarré le service et utilisé la commande « **radtest** » pour faire un teste en local afin d'assurer que Freeradius est bien configuré.

```
# /etc/init.d/freeradius restart
# radtest usercheck passcheck 127.0.0.1 0 radiussecret
```

La figure suivante nous montre que le résultat est positif et Freeradius est bien configuré.

```
debian:~# radtest usercheck passcheck 127.0.0.1 0 radiussecret
Sending Access-Request of id 135 to 127.0.0.1 port 1812
        User-Name = "usercheck"
        User-Password = "passcheck"
        NAS-IP-Address = 10.0.2.1
        NAS-Port = 0
rad recv: Access-Accept packet from host 127.0.0.1 port 1812, id=135, length=20
```

**Figure IV.9:** Le résultat du teste de Freeradius.

### 4.6- Configuration du point d'accès NAS (Tp-Link WN722N) :

#### 4.6.1- Quelques définitions :

- **Définition du NAS (Network Access Server) :**

Un Network Access Server (NAS) est un système qui permet d'accéder à un réseau. Dans certains cas, aussi connu comme un Terminal Server ou à distance Access Server (RAS).

Le NAS est censé agir comme une passerelle pour surveiller l'accès à une ressource protégée. Cela peut être quelque chose d'un téléphone réseau , d' imprimantes. Le client se connecte au serveur NAS. Le NAS se connecte alors à une autre ressource demandent si les informations d'identification fournies par le client sont valables. Basé sur cette réponse le NAS autorise ou interdit l'accès à la ressource protégée.

- **Définition du hostapd :**

Hostapd s'appuie sur les protocoles IEEE 802.11 AP et IEEE 802.1X/WPA/WPA2/EAP/RADIUS authentificateur.

Hostapd permet la création d'un point d'accès Wi-Fi, technologie sans fil utilisée pour se connecter à un réseau informatique. Dans les réseaux informatiques, un point d'accès sans fil (spot ou AP) est un dispositif qui relie les appareils de communication sans fil pour former un réseau sans fil. Le spot Wi-Fi se connecte généralement à un réseau câblé, et peut transmettre des données entre les appareils sans fil et les périphériques câblés. Plusieurs spots peuvent être liés ensemble pour former un réseau plus large qui permet le "**roaming**" (l'itinérance). Pour rappel, en revanche, un réseau où les machines clientes gèrent elles-mêmes - sans avoir besoin de point d'accès - devient un réseau ad-hoc.

Pour permettre aux utilisateurs d'accéder à notre Hotspot nous avons configuré le dongle wi-fi (**Tp-Link 722N**) en mode infrastructure en suivant les étapes suivantes :

Au début nous avons installé le logiciel Hostapd via la commande suivant :

```
# sudo apt-get  install hostapd
```

Puis nous avons édité le fichier **hostapd.conf** pour configurer notre point d'accès via la commande :

```
# nano /etc/hostapd/hostapd.conf
```

Et nous avons apportées les modifications suivantes :

```
# interface wlan du Wi-Fi
interface=wlan0

# nl80211 avec tous les drivers Linux mac80211
driver=nl80211

# Nom du spot Wi-Fi
ssid=Mon_Test

# mode Wi-Fi (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g)
hw_mode=g

# canal de fréquence Wi-Fi (1-14)
channel=6

# Wi-Fi ouvert, pas d'authentification !
auth_algs=0
```

**Figure IV.10:** Le contenue du fichier hostapd.conf.

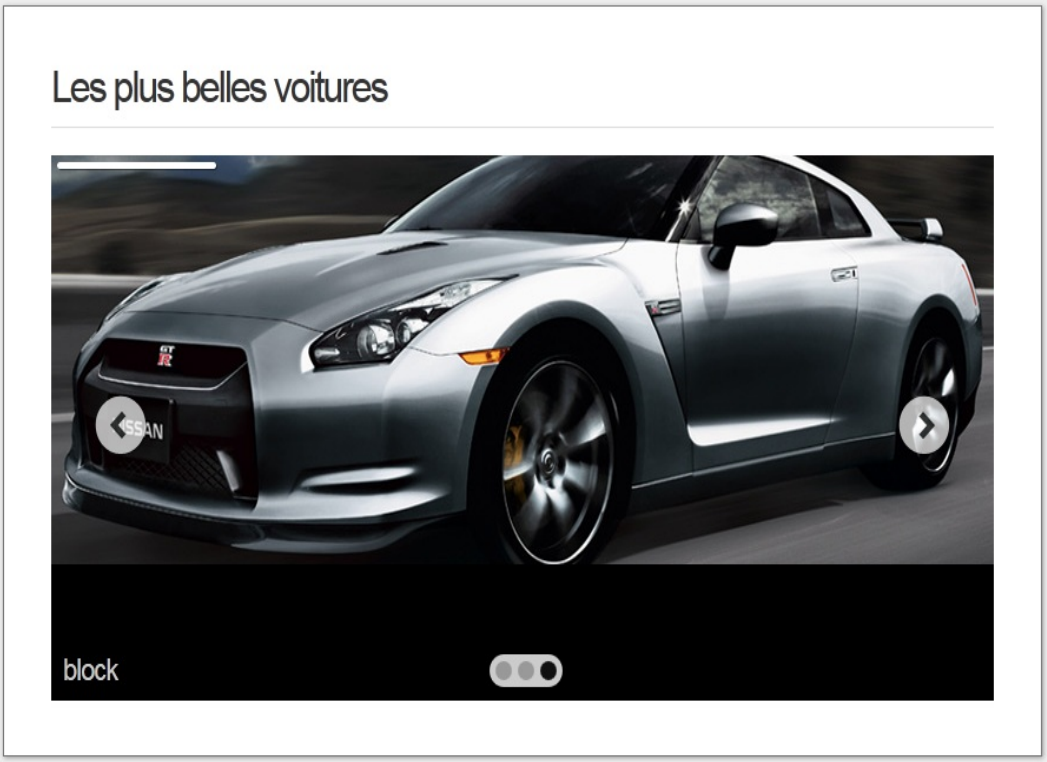
Puis nous avons redémarré hostapd pour qu'il prenne en considération les changements apportés via la commande :

```
# /etc/init.d/hostapd restart
```

## 5- Les interfaces de l'application :

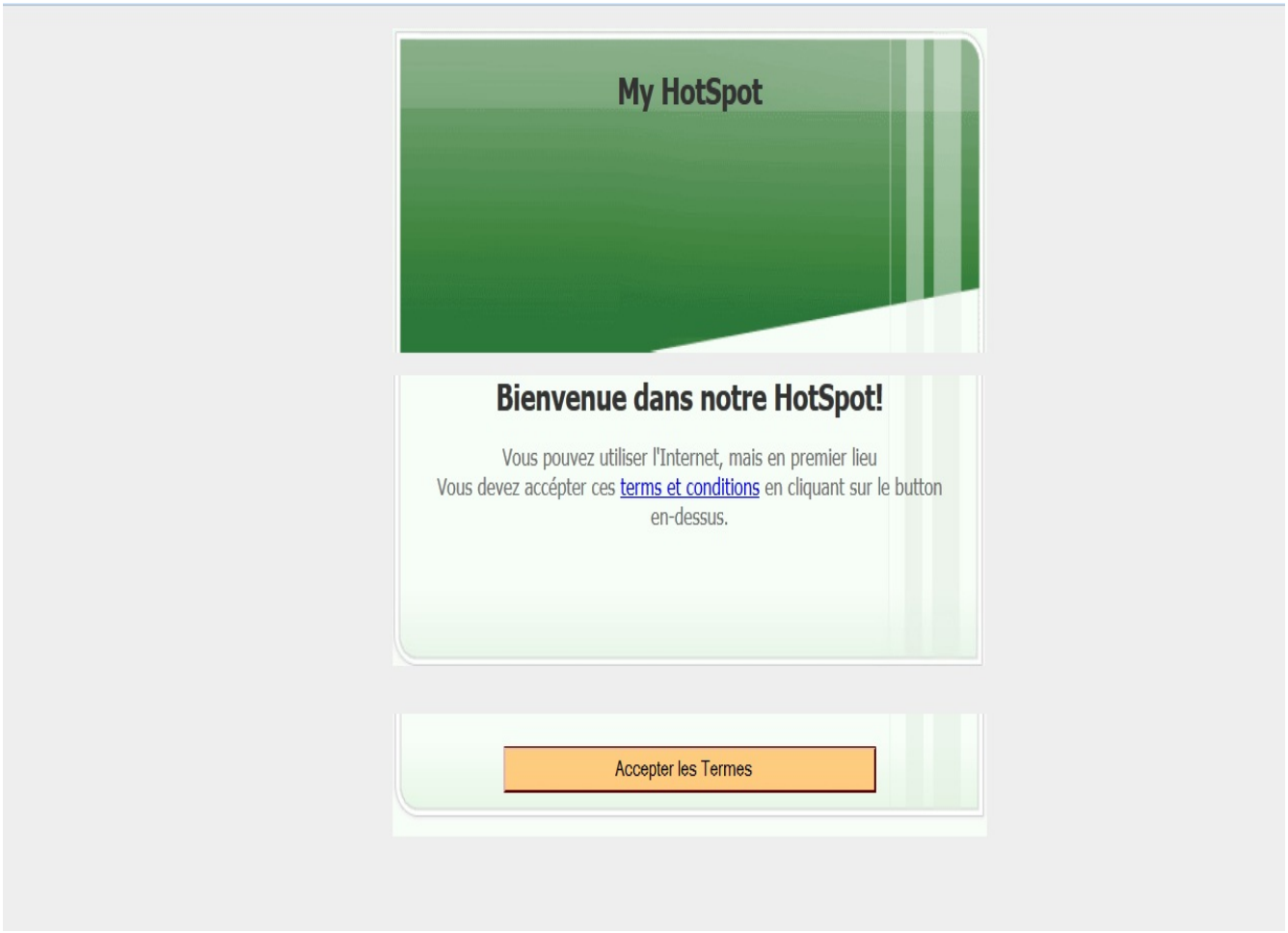
### 5-1- la page « publicitaire » :

C'est la première page qui s'affiche dés qu'un utilisateur ouvre son navigateur web, il regarde une publicité pendant un période de temps déterminé dans notre cas 20s.



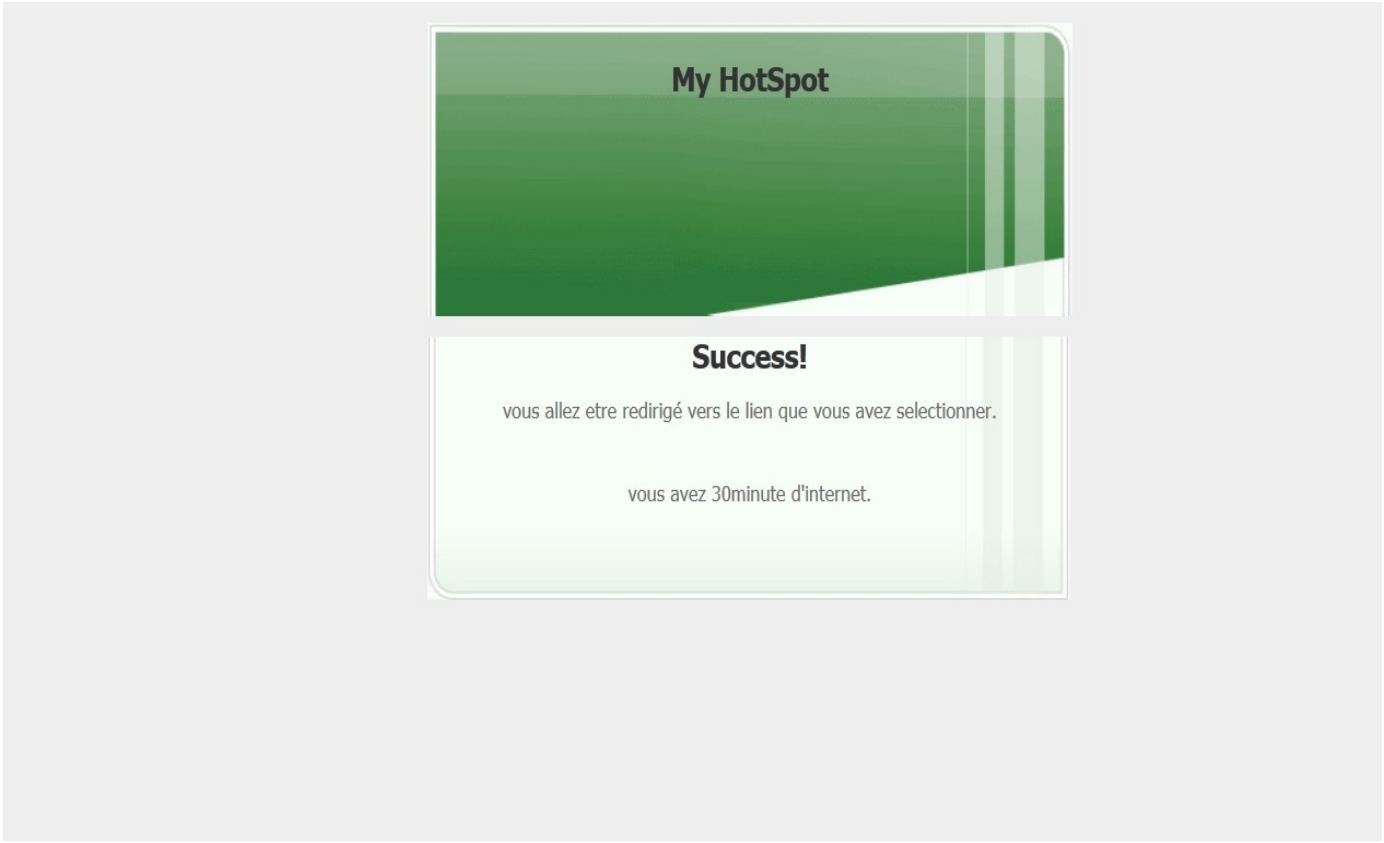
**5-2- la page « d'authentification » :**

Dés que l'affiche publicitaire est terminée, le système affiche automatiquement la page d'authentification suivante :



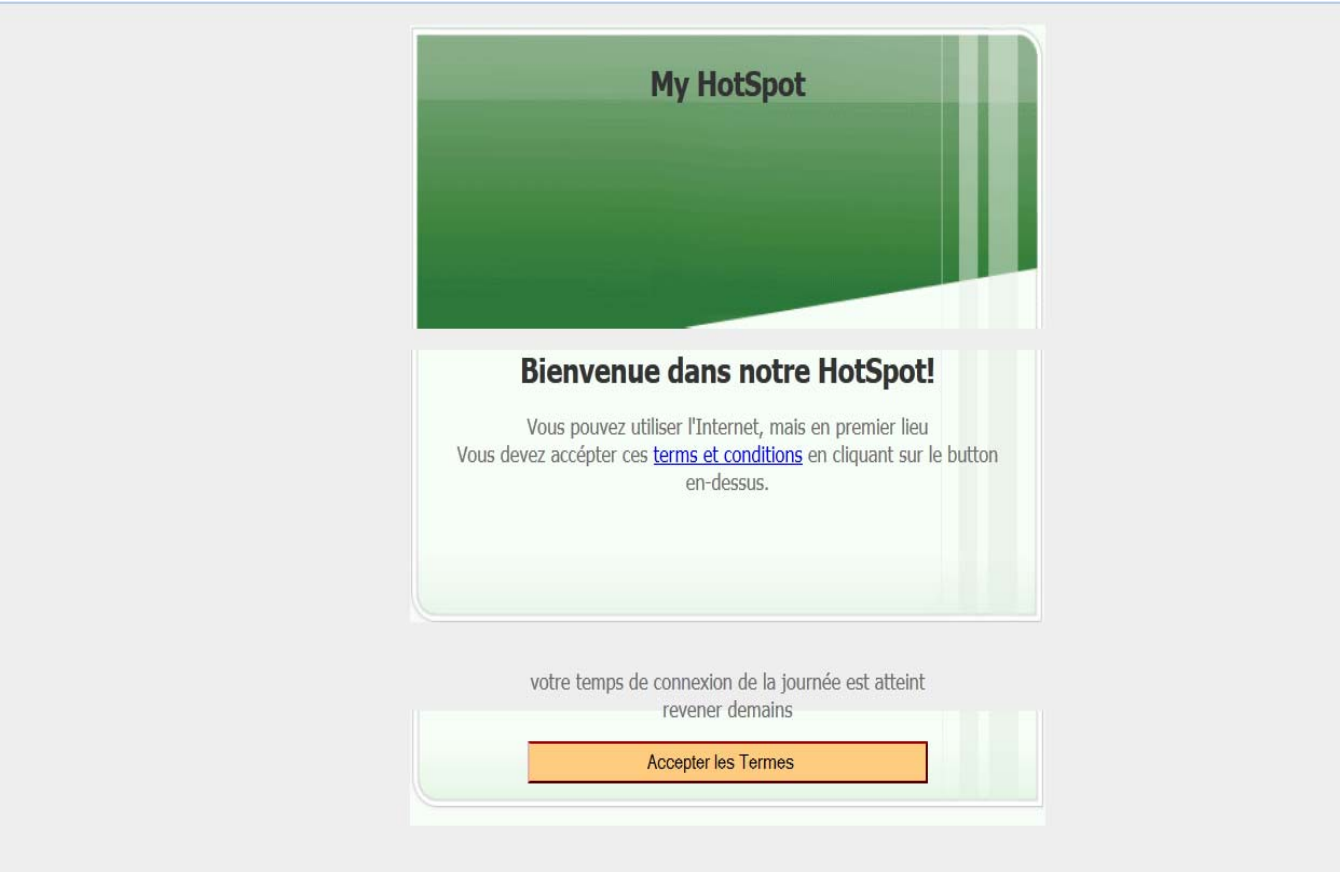
**5-3- la page « succès » :**

Lorsque l'utilisateur accepte les termes, le système lui affiche la page succès.



**5-4- la page « le temps de connexion est atteint » :**

Si le temps de connexion d'un utilisateur est atteint, le système lui indique que le temps de connexion de la journée est terminé.




## Conclusion :

On a vu dans ce chapitre que la mise en place d'un HotSpot Wi-Fi nécessite une longue configuration avant qu'il soit mis en service. D'après les première testes notre Hotspot est fonctionnel. Donc le projet est prêt d'être installé en dans un espace public, cafétéria, …etc.

# Conclusion générale

# CONCLUSION GENERALE:

Depuis leur apparition, les réseaux informatiques ont connu un franc succès, beaucoup de travaux ont traité ce sujet. Par conte, les réseaux sans fil restent un domaine vaste et encore fertile pour les chercheurs et les développeurs.

Dans ce travail de Hotspot, on a passé en revue le fonctionnement général du réseau sans fil en particulier le Wi-Fi. Ensuite, nous nous sommes familiarisés avec la carte électronique Raspberry-Pi, puis nous avons étudié le fonctionnement des systèmes de portails captifs, des serveurs d'authentification et de leurs protocoles mis au point pour assurer la sécurité des utilisateurs.

Enfin, l'élaboration de ce travail nous a permet d'une part d'approfondir les connaissances et le savoir faire acquis durant les années de notre formation au sein de l'université de Mouloud MAMMERI et plus précisément au département informatique, et d'autre part, de préparer notre intégration à la vie professionnelle et de nous nous situer sur le marché des télécommunications (réseaux, système de communication, services,…).

On espère que ce projet aidera les prochaines promotions qui voudront approfondir leurs connaissances sur les HotSpot Wi-Fi et de continuer notre travail en ce qui concerne le déploiement optimal des points d'accès et la configuration du Roaming entre les HotSpot.

# Bibiographie

# Bibliographie

**[radEYR07]** Serge Bordères, Authentification réseau avec Radius, Édition Eyrolles ,2007.

**[P-ATH S.B]** Protocoles d'authentification réseau Sans-fil et filaire, S. Bordères.

**[S-ATH A.M 09]** Services d'Authentication et Annuaires ; Abdelghani MAZOUZI ; UFR Informatique UCB Lyon1 ; 14 décembre 2009.

**[FIR A.J & A.M 04]** Les firewalls par Alban Jacquemin et Adrien Mercier; 15 février 2004.

**[G.P secEYR04]** Guy Pujolle : « sécurité wifi » octobre 2004. Edition Eyrolles.

**[M.D 04-05]** Analyse et simulation du déploiement d'un réseau sans fil à l'ULB. Mémoire de fin d'études présenté par Michel Duchateau en vue de l'obtention du grade d'Ingénieur Civil Electricien, spécialisé en Télécommunications. Année académique 2004-2005.

**[B.M EIVD02]** BALLESTEROS.M. (Les technologies sans fil) .EIVD, juin 2002.

**[J.E & W.A.A 04]** Jon Edney and William A. Arbaugh, Real 802.11 Security, Wi-Fi Protected Access and 802.11i; septembre 2004.

**[RAD T.M]** Rapport Servidor de RADIUS, HOTSPOT, *Tiago Mai.*

**[F-RAD A.D 09-10]** Rapport chillispot freeradius, Ahmet_DEMIR, Université de Reims, Master2 Administration et Sécurité des Réseaux, année 2009-2010.

**[M.M 08-09]** MOSTEFA MERIEM, PLACEMENT DES TACHES REPETITIVES SUR UNE ARCHITECTURE REGULIERE EMBARQUEE, Pour l'Obtention du Diplôme d'Ingénieur d'Etat en Informatique, Université d'Oran, promotion 2008-2009.

**[ALCAS H.O]** HOUSSENBAY Olivier, FreeRadius, un serveur d'authentification forte pour ALCASAR *FreeRadius, Mémoire de fin d'études,* ECOLE D'INGENIEUR DU MONDE NUMERIQUE.

**[FREE D.V.D.W 11]** Dirk van der Walt, FreeRADIUS Beginner's Guide, Edition Packt Publishing, 2011.

**[S.EMB B.B 08]** Systèmes Embarqués et Grandes Infrastructures –Bertrand BRAUNSCHWEIG Edition 2008.

# Webliographie

# Webliographie


[1] http://www.statistique-mondiale.com

[2] http://wifihotspot888.com/?page_id=181.wifi Hostspot888

[3] http://fr.wikipedia.org/wiki/coova principe

[4] http://translate.google.dz/translate?hl=fr&sl=en&u=http://www.coova.org/node/3535&prev=search

[5] http://fr.wikipedia.org/wiki/ Systèmes Embarqués

[6] http://www.tldp.org/HOWTO/8021X-HOWTO/freeradius.html

[7] http://freeradius.org/http://en.wikipedia.org/wiki/FreeRADIUS

[8] https://fr.wikipedia.org/wiki/Raspberry_Pi

[9] http://www.giiks.com/geek/raspberry-pi-infographie-44411/

[10] http://en.wikipedia.org/wiki/RADIUS

[11] http://en.wikipedia.org/wiki/IEEE_802.1X

[12] http://sourceforge.net/p/hotcakes/wiki/yfi_setup_FreeRADIUS/

[13] http://sourceforge.net/p/hotcakes/wiki/yfi_setup_nas_coova/

[14] http://www.raspberrypi.org,2015

[15] http://lea-linux.org/documentations/Pr%C3%A9sentationduRaspberryPi

[16] http://freeradius.org/

[17] http://deployingradius.com/

[18] http://doc.ubuntu-fr.org/coovachilli

[19] http://www.pobot.org/-La-carte-Raspberry-PI-.html

[20] http://doc.ubuntu-fr.org/tutoriel/comment_configurer_son_reseau_loca

[21] http://doc.ubuntu-fr.org/partage_de_connexion_internet

[22]http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-fr-4/s1-firewall iptfwd.html

[23]http://blog.inforeseau.com/2010/10/firewall-iptables-rediriger-un-port versun-autre

[24] https://www.yanx.eu/coova-chilli/

[25] http://antoine-schellenberger.com/linux/2014/12/14/hostapdiptable.html

[26] http://doc.ubuntu-fr.org/iptables

[27] http://wiki.vpslink.com/HOWTO:_Quick_n'_Dirty_IPTables-Based_Firewall

[28] https://help.ubuntu.com/community/WifiDocs/MasterMode

[29] http://www.armetiz.info/raspberry-pi-point-dacces-wifi/

[30] http://hardware-libre.fr/2014/02/raspberry-pi-creer-un-point-dacces-wifi/

[31] http://bac.sen.avp.st-gab.over-blog.com/article-configurer-un-netbook-ou un-pc-en-point-d-acces-wifi-71645913.html

[32] http://wiki.backtrack-fr.net/index.php/Configurer_son_interface

[33]http://blog.trifork.com/2013/01/15/building-a-captive-portal-controlling access-to-the-internet-from-your-network/

[34]https://help.ubuntu.com/community/WifiDocs/ChillispotHotspot

[35]http://subgroup-ash.blogspot.com/2014/02/modifying-coova-chilli-to allow.html

[36] http://www.geekmag.fr/raspberry-pi-creer-un-hotspot-wifi-avec-une-cle-edimax/

[37] http://deusyss.developpez.com/tutoriels/RaspberryPi/PythonEtLeGpio/