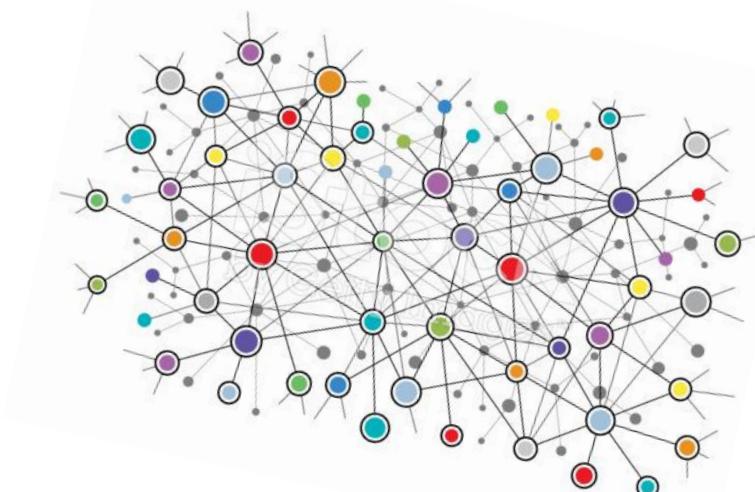


Conception d'un système communicant sécurisé pour le transport urbain

Rapport du stage de fin d'études



Etudiant

Xuan Thong **DANG**

Tuteur

Arnaud **FEVRIER**

Superviser

Traian **MUNTEAN**

(Systèmes embarqués communicants, réseaux sans fil et mobiles, C/C++, Linux/Androïde, Raspberry Pi)

Marseille, le 05/09/2016

Résumé

Dans le cadre d'une formation supérieure d'informatique à Aix-Marseille Université, j'ai effectué un stage de fin d'études dans le groupe de recherche eRISCS¹ en collaboration avec l'équipe technique de la RTM², sur un sujet de conception d'un système communicant sécurisé pour le transport urbain, sous la direction de Messieurs A. Février et T. Muntean, du Groupe ERISCS.

Mon travail fut, en premier temps, d'étudier l'état de l'art et la problématique de communication d'un système de transport urbain, plus précisément, la communication pour un réseau des bus et le(s) serveur(s) de surveillance et monitoring. C'est un système de communication spécifique, embarqué et mobile, soumis à des contraintes techniques et d'usages multi-réseaux de communication.

On distingue deux types de communication dans le système mentionné : l'une est la communication interne entre les dispositifs du véhicule et le conducteur, l'autre est la communication externe entre les bus et le serveur, et éventuellement plus tard entre les bus eux-mêmes. Mon travail dans ce stage de six mois concerne les deux dernières.

La communication externe est basée sur un réseau hétérogène, sur lequel s'imposent les contraintes liées au problème de basculement de connexions entre plusieurs réseaux. Ma première tâche était de pouvoir maîtriser le comportement et le contrôle d'un tel basculement, et de mesurer et donner une solution pour un basculement efficace.

Sur ces réseaux disponibles s'établit une connexion de bout-en-bout qui sera utilisée pour transmettre des données de manière sécurisée entre les bus et le serveur. Toutes ces données seront donc chiffrées pour protéger la vie privée des passagers et le fonctionnement correct du système contre tous types d'attaques. Un protocole de communication sécurisé spécifique est donc requis. Après une analyse des solutions existantes, tant parmi les standards et en les comparant à une proposition récente de l'équipe de travail, j'ai eu de la chance de concevoir une solution que je pense la plus adaptée au besoin industriel analysé, après des améliorations appropriées.

Mon travail dans l'équipe eRISCS est encore en cours pour finaliser et valider les phases d'implémentation et de mesure de performance de la solution proposée.

¹ Groupe d'Études et Recherche en Informatique des Systèmes Communicants Sécurisés

² Régie des Transports Marseillais

Table des matières

I.	INTRODUCTION	6
1.	LE CHOIX DE SUJET DE STAGE.....	6
a.	« <i>Tout ce qui concerne les transports m'intéresse !</i> »	6
b.	<i>Le sujet et la motivation</i>	6
c.	<i>Le groupe de recherche eRISCS et la collaboration avec la RTM</i>	7
2.	LA NATURE D'UN TRAVAIL D'INGENIEUR AU SEIN D'UNE EQUIPE DE RECHERCHE.....	8
II.	MISSIONS ENGAGEES.....	9
1.	MAITRISER L'ETAT DE L'ART.....	9
2.	LE TRAVAIL AVEC LES RESEAUX SANS FILS	10
a.	« <i>Understanding 802.11 connection process</i> »	10
b.	<i>Des mesures pour vérifier une hypothèse</i>	11
3.	LA CONCEPTION D'UN PROTOCOLE DE COMMUNICATION SECURISE	15
a.	<i>La problématique</i>	15
b.	<i>Analyse</i>	15
c.	<i>L'amélioration du SVC</i>	18
d.	<i>Une nouvelle approche d'architecture</i>	19
e.	<i>Évaluation de la solution</i>	21
III.	BILAN.....	22

Table des figures

Figure 1 – Une modèle de gestion de code source	8
Figure 2 – L'architecture actuelle de la communication du bus	9
Figure 3 – La machine d'état d'une connexion 802.11.....	10
Figure 4 – Les frames échanges dans une connexion de 802.11	11
Figure 5 – Les interfaces utilisées dans mon expérimentation.....	12
Figure 6 – Les réseaux Wi-Fi disponibles.....	12
Figure 7 – Le flux du programme de mesure	13
Figure 8 – Le résultat de la connexion normale versus celle de fréquence fixée	14
Figure 9 – La version originale du SVC	18
Figure 10 – Echange de messages dans la version modifiée du SVC.....	19
Figure 11 – L'architecture service-application du SVC.....	20

INTRODUCTION

1. Le choix de sujet de stage

a. « Tout ce qui concerne les transports m'intéresse ! »

Je suis né dans un pays où la sécurité routière devient le problème le plus majeur du gouvernement et des résidents. Selon une enquête, en 2014, il y avait plus de 25 miles accidents de la circulation, qui ont causé 9 miles morts et 24 miles blessés. En moyenne, chaque matin, 25 personnes sortent de leurs maisons et ne retournent jamais ; elles ont évidemment une famille qui les attendait le soir.

La qualité du réseau de transport est une des raisons qui sont à l'origine de ce fait. Je crois que, si on augmente la qualité du service de transport public, cela encourage les habitants de l'utiliser au lieu de conduire en motos, le nombre d'accidents baîssera significativement.

En étant un ingénieur en informatique dans le domaine SICA³, les problèmes des systèmes de transport intelligents m'intéressent vraiment. J'apprécie la qualité du service de transport de la France et des pays Européens, et voudrais bien apprendre ces modèles et techniques pour les appliquer au Vietnam. J'espère pourvoir contribuer au développement de mon pays, et réduire toutes les peines et pertes que causent les accidents de la circulation.

b. Le sujet et la motivation

Le sujet que m'a proposé le groupe de recherche eRISCS est définitivement tout ce que je cherche. Il concerne un système communicant sécurisé d'un réseau de transport urbain. Ce système est certainement la veine de communication d'un réseau de transport intelligent.

La problématique se produit quand on a de plus en plus des données à transmettre depuis le véhicule vers le serveur, notamment les données de la vidéo-surveillance en temps réel. L'utilisation d'un réseau mobile 4G/3G devient insuffisante sur quelques points d'accès, et on doit penser à des solutions complémentaires comme Wi-Fi/Wi-Max. Et pourtant, le basculement de connexion entre les réseaux reste à mesurer et évaluer.

Je considère ce stage comme une opportunité de pratiquer les connaissances apprises dans les matières Système d'Exploitation, Cryptographie, Linux Temps Réel, Architecture des Systèmes Embarqués, et de me former un style de travail d'ingénieur concepteur de systèmes complexes.

³ Systèmes d'Information Critiques et Applications

c. Le groupe de recherche eRISCS et la collaboration avec la RTM

eRISCS est le premier groupe de recherche interdisciplinaire « mathématique – informatique » créé par des coopérations scientifiques et projets communs menés par des chercheurs et enseignants-chercheurs d'Aix-Marseille Université. La thématique générale du groupe est la construction correcte et les applications des « Systèmes Communicants Critiques et Sécurisés ».

L'objectif général du groupe, en collaboration avec des acteurs industriels et dans le contexte d'une gamme d'applications naissantes pour les systèmes communicants, est donc de proposer une approche de la sécurité des systèmes communicants qui adresse les défis les plus actuels et contribue à l'évolution des standards.

Depuis des années, le groupe de recherche eRISCS et la RTM collaborent sur un projet pour l'amélioration du service et le monitoring des bus en temps-réel. J'ai eu vraiment de la chance d'effectuer mon stage chez eRISCS dans cette période, car la RTM est en train d'adopter une nouvelle norme du système de transport par bus, EBSF⁴, proposée par UITP⁵. C'est la première fois que je participe dans un grand projet de niveau européen, et cela me rend beaucoup d'expérience et des contacts avec les sociétés dans le domaine.

⁴ European Bus System of the Future

⁵ International Association for Public Transport

2. La nature d'un travail d'ingénieur au sein d'une équipe de recherche

Mon premier stage était un travail dans l'environnement d'entreprise. Cette fois c'est une opportunité pour moi de vécu au sein d'une équipe de recherche, ce qui va certainement me donner des expériences différentes.

Contrairement à un travail d'entreprise, où on est forcé de faire ce dont on est demandé et respecter le « deadline », on a un peu plus de temps dans le travail de recherche pour penser à une « meilleure » solution. La R&D demande beaucoup de lecture pour avoir la vision globale du problème, ce qui prend la plupart de temps de travail. J'ai reçu aussi beaucoup de renseignements et orientations depuis mon tuteur, car dans la recherche on peut facilement se perdre.

Le sujet de la recherche est à définir de l'étape par l'étape pendant tout le travail. Mais avant de commencer le stage, j'ai essayé de fixer un cadre de travail et des outils nécessaires. Ci-dessous est un modèle que j'ai appliqué pour la gestion de code source en utilisant le logiciel « git ».

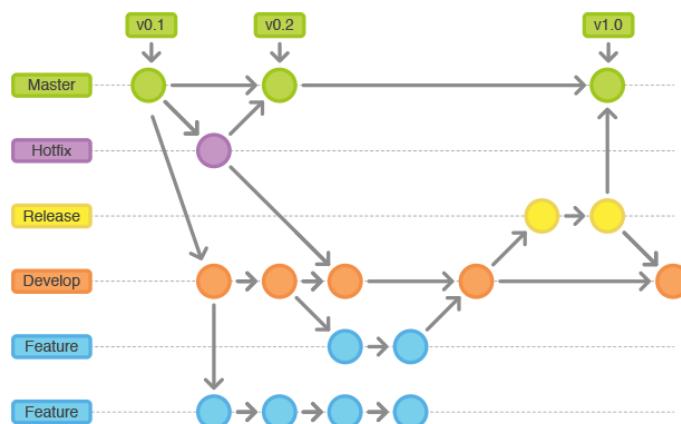


Figure 1 – Une modèle de gestion de code source

Après tout, je pense que, une aptitude de travail d'ingénieur, la capacité d'analyse et de synthèse, la capacité de poser et résoudre de problèmes, l'esprit de travail d'équipe et autonomie, sont tous les éléments qui déterminent le succès de mon stage.

MISSIONS ENGAGEES

1. Maitriser l'état de l'art

Il y avait beaucoup de documents à lire. J'ai commencé par l'architecture « hardware » de bus pour avoir une vision générale.

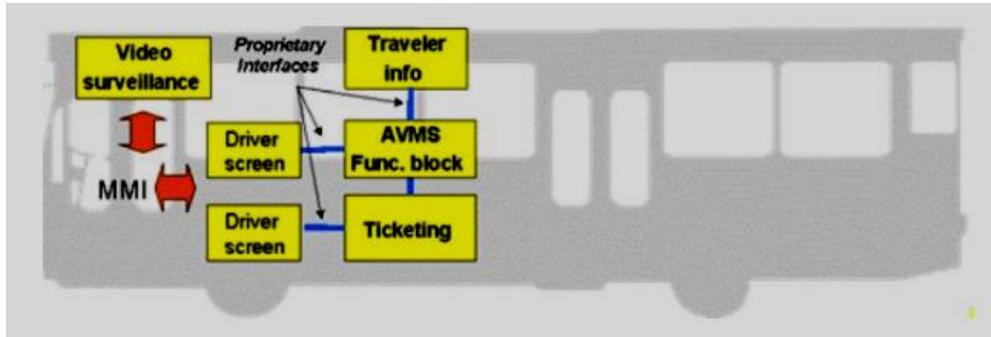


Figure 2 – L'architecture actuelle de la communication du bus

J'ai aussi eu la chance d'assister aux présentations des sociétés partenaires comme Visionetics, TCL, Digimobee concernant leurs solutions, dans lesquelles on a constaté des points forts et des points faibles.

Parmi les documents regardés, je me suis concentré sur des problématiques :

- Les problèmes de sécurité dans la communication :
 - o “A Critical Analysis on the Security Concerns of Internet of Things” [1]
 - o “Security in the Internet of Things: A Review” [2]
 - o “Vulnerabilities of LTE and LTE-Advanced Communication” [3]
- Les réseaux de communication:
 - o Réseau TETRAPOL: <https://en.wikipedia.org/wiki/TETRAPOL>
 - o Réseaux Wi-Max : <https://en.wikipedia.org/wiki/WiMAX>
 - o Etc...

Après des lectures, j'ai eu suffisamment de l'information et connaissance pour lancer la première attaque du sujet.

2. Le travail avec les réseaux sans fils

a. « Understanding 802.11 connection process »

Mon premier problème a été de mesurer et évaluer la connexion Wi-Fi dans des réseaux multiples. Ces réseaux varient en termes de types d'authentification et chiffrement, force de signal, bande passante, etc...

J'ai fait encore des lectures comme :

- Fast secure roaming [4]
- “A study of the discovery process in 802.11 networks” [5]

Sur lesquels j'ai trouvé un modèle d'authentification et d'association de 802.11 :

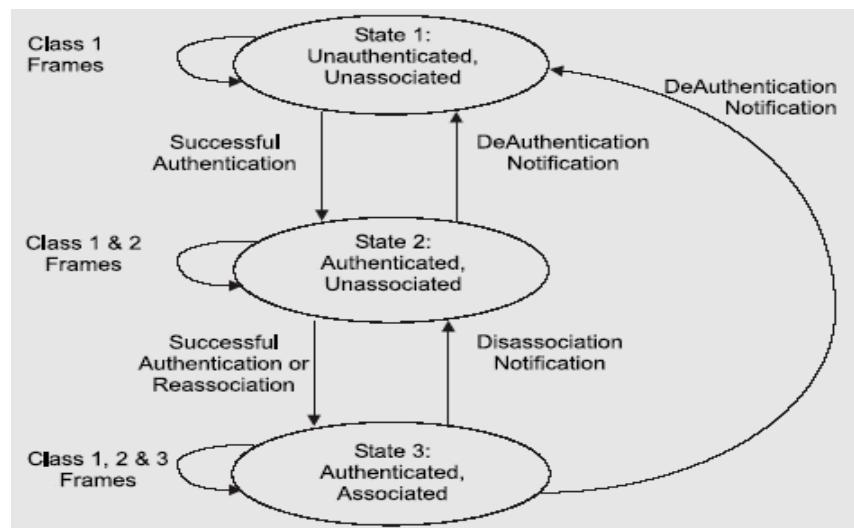


Figure 3 – La machine d'état d'une connexion 802.11

Avec une pratique sur le matériel « Raspberry » et l'utilisation du « wireshark », j'arrive à sortir mon premier résumé du processus de connexion Wi-Fi, et donner des suggestions d'amélioration de performance :

« Le comportement par défaut de Linux tente à effectuer un scan de réseau complet à chaque fois qu'il reconnaît un changement (une déconnexion, une reconnexion, une nouvelle balise d'un point d'accès), qui ralentira le processus de connexion. Ce défaut peut être simplement remplacé par un scan de fréquence fixe.

Dans le cas où un scan complet du réseau est nécessaire, un scan d'horloge adaptative peut être effectué.

CCKM doit également être pris en considération pour un basculement rapide de connexion.»

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	802.11	2462	Authentication, SN=2465, FN=0, Flags=-
2	0.000781	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	802.11	2462	Authentication, SN=275, FN=0, Flags=-
3	0.002579	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	802.11	2462	Association Request, SN=2466, FN=0, Flags=-
4	0.007765	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	802.11	2462	Association Response, SN=276, FN=0, Flags=-
5	0.012140	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAPOL	2462	Request, Identity
6	0.032606	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAPOL	2462	Start
7	0.055297	cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAPOL	2462	Request, Identity
8	0.061197	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAPOL	2462	Response, Identity
9	0.081408	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAPOL	2462	Request, Protected EAP (EAP-PEAP)
10	0.117423	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Client Hello
11	0.145299	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.167145	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.182327	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.196221	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAP	2462	Response, Protected EAP (EAP-PEAP)
15	0.201527	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	0.210076	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Certificate, Client Key Exchange, Change C
17	0.220032	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	0.222784	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAP	2462	Response, Protected EAP (EAP-PEAP)
19	0.227233	cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
20	0.291268	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Application Data, Application Data
21	0.291869	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Application Data, Application Data
22	0.295816	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	0.297768	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Application Data, Application Data
24	0.304669	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	0.313817	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
26	0.315992	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Application Data, Application Data
27	0.321376	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Request, Protected EAP (EAP-PEAP)
28	0.323863	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	TLSV1	2462	Application Data, Application Data
29	0.328766	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAP	2462	Success
30	0.330360	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAPOL	2462	Key (Message 1 of 4)
31	0.334225	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAPOL	2462	Key (Message 2 of 4)
32	0.338645	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:08	EAPOL	2462	Key (Message 3 of 4)
33	0.341932	Aironet_b7:ab:5c Cisco_f0:68:d8		84:78:ac:f0:68:08	EAPOL	2462	Key (Message 4 of 4)

Figure 4 – Les frames échangés dans une connexion de 802.11

Le résultat dans cette expérimentation me permet d'établir une hypothèse qui sera vérifiée par des mesures dans le travail suivant.

b. Des mesures pour vérifier une hypothèse

Mon hypothèse affirme que le temps du scan peut être réduit si on fixe la fréquence du scan et utilise un type d'authentification approprié. Pour pouvoir prouver cette hypothèse, des mesures et évaluations doivent être effectuées.

Avec une limite de dispositifs, j'ai essayé d'utiliser les ressources disponibles. Ici wlan0 est la carte Wifi de mon PC et wlan1 est celui qui est fourni par le groupe de recherche.

J'ai profité aussi les réseaux Wi-Fi de l'école pour avoir le plus possible des réseaux différents.

Logical interface name	wlan0	wlan1
Manufacturer	Intel	Realtek
Model	AC 3160	802.11n WLAN Adapter
Hardware interface	PCI	USB
Dual band	yes	no
Driver	iwlwifi	rtl8192cu
Additional capacities		
Memory	8 KB	-

Figure 5 – Les interfaces utilisées dans mon expérimentation

SSID	eRISCS	eduroam
Operating bands	2.4 Ghz and 5 Ghz	2.4 Ghz and 5 GHz
Authentication method	EAP-WPA	EPA-TTLS with phase2 MSCHAP
Signal strength	Good	Good
Internet connection	No	Yes

Figure 6 – Les réseaux Wi-Fi disponibles

J'ai écrit un programme C/C++ et compilé en g++ 4.9.xxx pour faire les mesures, en utilisant des bibliothèques utilitaires :

- wpa_ctrl.h: cet entête est une partie de « wpa_supplicant », un programme de gestion de réseaux sans fils. wpa_supplicant nous permet d'appliquer des configurations différentes aux dispositifs en définissant des paramètres nécessaires.
- pcap.h: cet entête vient de « pcaplib », ce qui sera utilisée pour capturer les trafics du réseau.
- nl80211.h: nl80211 est le nouvel entête public pour l'interface du type « netlink ». Avec cfg80211, nl80211 est tenté à remplacer WE [6], qui est actuellement l'interface de gestion de réseau sans fils.

Une conception de ce programme est réalisée avant tout acte de programmation. Cette conception permet une implémentation rapide et correcte.

Des scénarios de tests sont aussi prévus. Tout est prêt en attendant un programme de test.

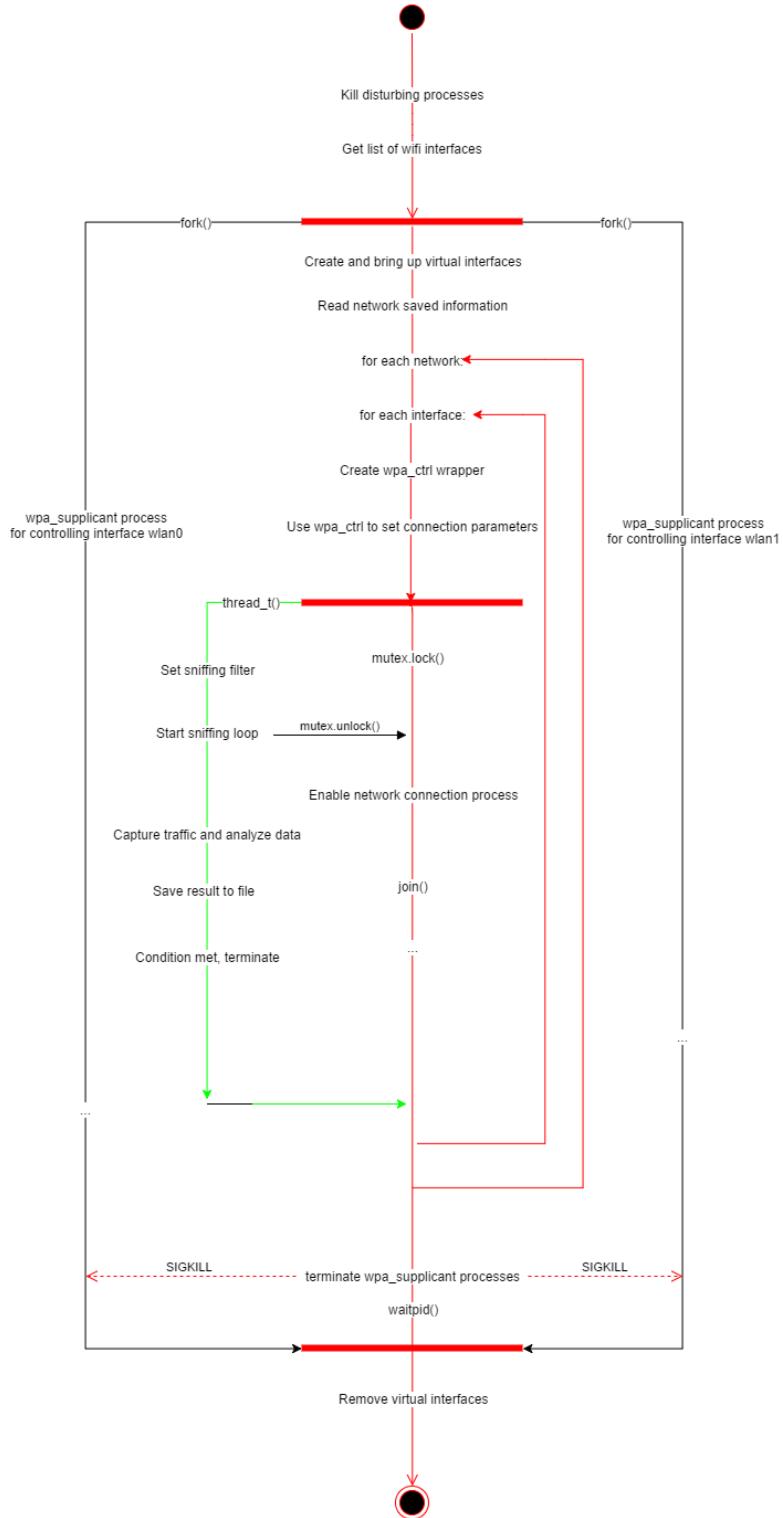


Figure 7 – Le flux du programme de mesure

Les données mesurées ont non seulement affirmé l'hypothèse posée, mais aussi m'ont inspiré un résultat hors attente. Parmi les options disponibles, on a montré l'option la plus efficace au problème de basculement de connexion Wi-Fi.

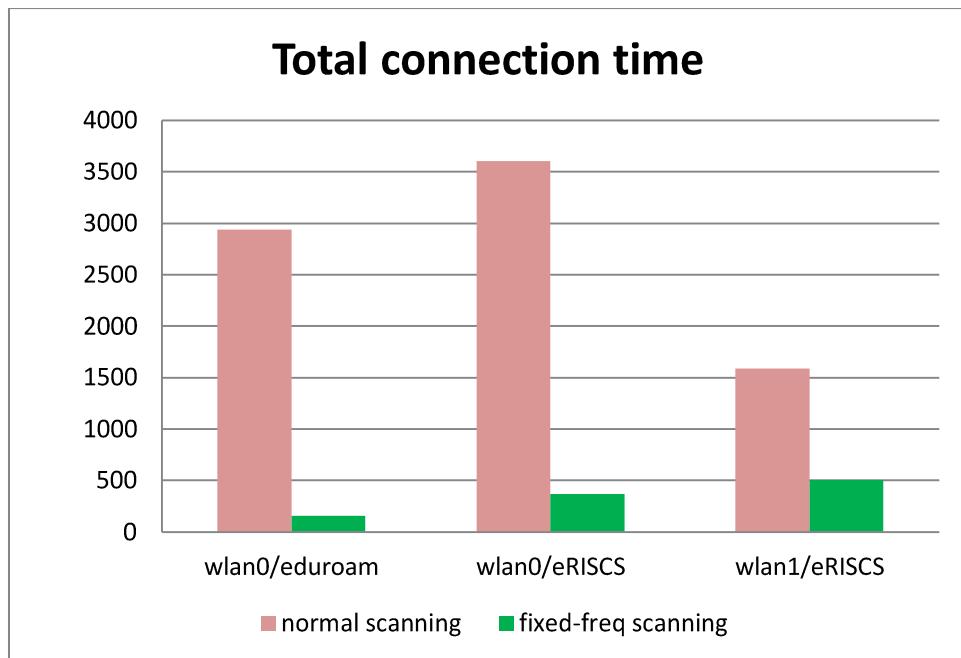


Figure 8 – Le résultat de la connexion normale versus celle de fréquence fixée

3. La conception d'un protocole de communication sécurisé

a. La problématique

Pour supporter la communication entre les bus et les serveurs, des réseaux hétérogènes seront utilisés (wifi, 3G, 4G), sur lesquels s'appliquent un protocole de communication sécurisé. De tels protocoles existent sur le marché, mais rien n'a été conçu pour ce problème spécifique.

Ma mission est donc d'analyser la nature de la communication, les avantages et désavantages des solutions existantes et de donner la solution la plus adaptée à ce besoin.

b. Analyse

i. La nature de la communication entre les bus et le serveur

La communication dans le réseau de transport est un type de communication spécifique. Elle subit à des contraintes et obligations issues de la nature du service.

- **Basculement rapide de connexion** : en raison du mouvement des bus, la connexion vers le serveur est souvent interrompue quand le bus sort d'une zone de couverture d'une antenne vers celle d'une autre, ou empêchée par de grands obstacles physiques. Le changement du type de réseaux peut aussi se produire au cas où le support du réseau courant ne peut pas desservir toutes les stations qui sont autour d'un point d'accès.
- **Gros volume de données** : les données récoltées pendant les trajets du bus, concernant le statut de tous les équipements, la position cartographique du véhicule et les points d'accès, et notamment le vidéo-surveillance, sont montées et envoyées vers le serveur.
- **Diversité de types et de priorités de données** : les données montées viennent de sources différentes, certaines sont capables de tolérer des erreurs (par exemple quelques trames de vidéo-surveillance), autres requièrent strictement d'être livrées sans erreurs. On parle aussi des priorités d'envoi, sur lesquelles les données sont traitées et envoyées dans le cas des congestions.
- **Temps réel** : toutes les données de surveillance doivent être transmises au serveur dès qu'elles sont disponibles, pour que les contrôleurs aient des réactions à temps en cas d'incidents.
- **Sécurité** : les informations collectées dans le bus, y compris les images des passagers sont liées à la vie privée, doivent être transmises et exploitées

de façon sécurisée. Une méthode de chiffrement est indispensable dans l'ensemble de la solution proposée pour les connexions.

ii. Les exigences d'un protocole de communication spécifique

Suite aux natures de la communication citées ci-dessus, un protocole qui s'adapte à notre besoin doit posséder des propriétés/technologies suivantes:

- Sans connexion : une communication avec un basculement rapide de connexion ne devrait pas utiliser un protocole de transport en mode connecté comme TCP, ce qui augmenterait la latence de connexion à cause de sa phase de négociation et des contrôles inutiles. Pour le moment, UDP restera comme le choix privilégié, mais un protocole « hybride » a été considéré comme remplaçant, avec des contrôles simples pour la garantie de livraison de données.
- Reprise de connexion rapide : dans le même but de baisser le temps de connexion/reconnexion, on utilisera une d'identité pour distinguer toutes les connexions vers/depuis des hôtes. Une identité de session nous permet d'identifier l'hôte, quand l'autre détermine le bout de communication (une instance de l'application).
- Confidentialité persistante parfaite : pour bien protéger les identités des clients et les données des services, le protocole sécurisé doit supporter la **confidentialité persistante parfaite** (« perfect forward secrecy » en anglais). Comme ça, il garantit que la découverte par un adversaire de la clé privée d'un correspondant (secret à long terme) ne compromet pas la confidentialité des communications passées.
- Cryptographie elliptique : les courbes elliptiques seront utilisées pour réduire la taille des clés de chiffrement en gardant le même niveau de sécurité et économiser la bande passante.
- Indépendance de l'environnement : le protocole devra être utilisé sur plusieurs plateformes et appareils, il faudrait donc ne pas être dépendant des versions du système d'exploitation et des bibliothèques externes, mais doit fournir ses propres implémentations des algorithmes et services.

iii. Les solutions existantes et la motivation pour un nouveau protocole

Les protocoles sécurisés qu'on rencontre de nos jours sont basées essentiellement sur TCP : TLS, SSH, OpenVPN, etc...

Mais, le TCP n'est pas un choix idéal pour la solution. Il ne satisfait pas la première caractéristique de la connexion, car l'établissement et la reprise de la connexion TCP prend toujours du temps considérable. De plus, la garantie de livraison de paquet n'est pas nécessaire en permanence, car la plupart de temps, les données sont tolérantes aux pertes. Le compromis entre la fiabilité et la rapidité du TCP augmente la latence de la connexion, ce qui la rend moins efficace pour l'utiliser dans un contexte temps réel.

On peut par contre trouver quelques versions UDP des protocoles ci-dessus (comme DTLS), ou ces protocoles offrent une option UDP eux-mêmes. Pour garder la même opérabilité qu'en TCP, des modifications supplémentaires sont introduites. Dans l'optique de minimiser la complexité, ce n'est pas non plus une bonne décision.

D'ailleurs, comme indiqué dans un travail de l'équipe ERISCS publié récemment [1], les protocoles mentionnés s'imposent encore des limites contre la protection d'identité du client ou l'exposition des informations sensibles.

Dans cette vision née l'idée de SVC (Secure Virtual Connectors), un nouveau type de protocoles sécurisés qui surmonte ces limites en proposant une négociation simple et un découplage entre l'authentification et le chiffrement. SVC implémente les algorithmes de chiffrement les plus récents sans utiliser des sources externes, cela facilitera le processus d'administration et maintenance.

c. L'amélioration du SVC

J'ai trouvé beaucoup d'idées innovantes dans la conception du SVC. Parmi lesquelles, j'ai choisi à développer l'idée de découplage de l'authentification et de l'autorisation, et l'idée de protection d'identité du client. Et pourtant, SVC présente encore dans la conception des points à améliorer.

Dans la version originale proposée par ses auteurs, le protocole commence par une requête depuis le serveur vers le client (après un succès de connexion). Cette approche est justifiée comme une façon de réduire la durée de la négociation. Je ne suis pas d'accord avec cette explication car pour la phase d'initialisation de connexion SVC consomme 4 exchanges de message.

Le problème le plus grave c'est le fait d'avoir l'étape d'échange de clé commencée par le serveur, ce qui rend le protocole sensible aux attaques par rejet. La seule façon pour l'éviter, c'est de rajouter du timestamps, qui ne sera pas une bonne idée.

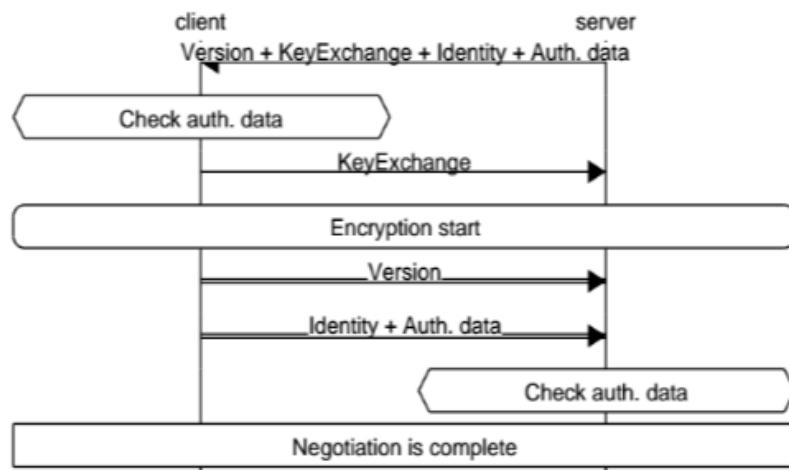


Figure 9 – La version originale du SVC

Avec seulement 3 échanges, j'ai proposé une nouvelle approche basé sur le principe du « serrement en 3 phases » qui garde encore les idées basiques du SVC.

Dans la vérification de l'authenticité, l'approche « challenge – proof » est introduite pour éviter tout type d'attaque par rejet. Le DH-STS est forcément imperméable, seulement vulnérable à quelques attaques du type « unknown key-share » [7].

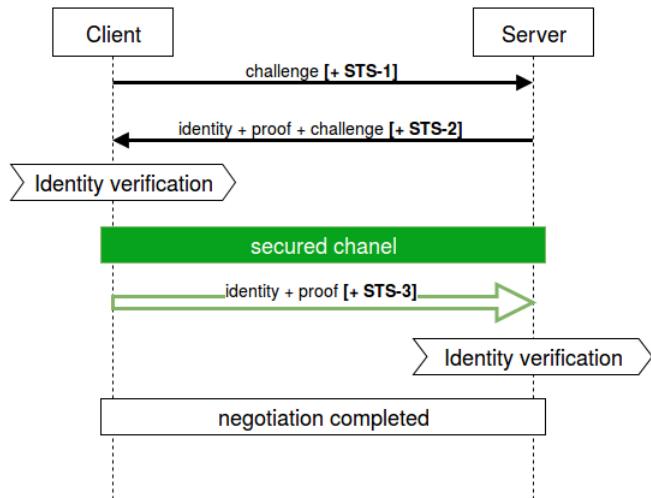


Figure 10 – Echange de messages dans la version modifiée du SVC

À comparer avec la version originale, la nouvelle montre des avantages supplémentaires:

- Moins d'échanges (3 contre 4)
- Résistant à l'attaque par rejeu

d. Une nouvelle approche d'architecture

À l'origine, SVC vise à servir une communication interne, c'est-à-dire une communication entre les instances d'une même application. Dans l'environnement multitâche, cette conception montre des faiblesses:

- chaque connexion doit manager ses propres (mais avec une même politique) paramètres de sécurité, chiffrement. Si on a plusieurs connexions vers une même hôte, toutes les étapes de négociation se refont.
- La mise à jour des protocoles de chiffrement du SVC impose la recompilation de toutes les applications qui l'utilisent.

L'idée est de découper la couche « application », qui est en charge de l'authentification et la communication de données, avec une seule instance de la couche « daemon », qui s'occupe des services de chiffrement et de négociation. Comme ça, on remédié tout de suite les problèmes posés :

- Les services gèrent toutes les connexions depuis et vers des hôtes. Quand une nouvelle connexion vers une même hôte est détectée, on utilise le service correspondant pour profiter d'un canal déjà sécurisé, laisse passer les échanges de clés.
- La mise à jour se fait en simplement replaçant et redémarrant l'instance du daemon. Toutes les applications restent intouchées.

Dans le diagramme ci-dessous se trouvera l'architecture de la solution :

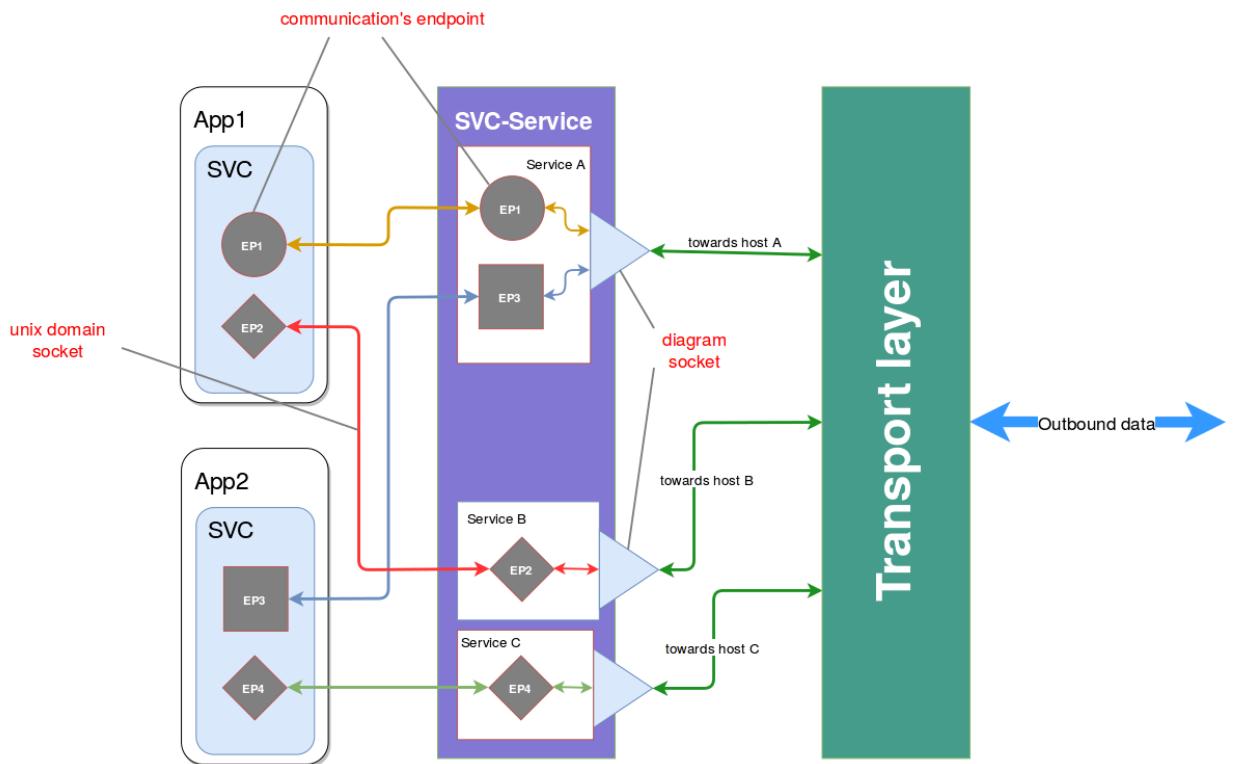


Figure 11 – L'architecture service-application du SVC

Le détail technique et l'implémentation de ce protocole se trouve dans l'annexe de ce rapport.

e. Évaluation de la solution

La phase de conception de cette solution est finie. Pendant la phase d'implémentation, des petites modifications peuvent être introduites, mais toujours en respectant les principes déterminés.

Une partie de l'implémentation est déjà en cours d'être réalisée. Dès qu'une version complète est disponible, on va tester et comparer ses performances avec les protocoles existants pour pouvoir l'améliorer. Une conception du HTP (Hybrid Transmission Protocol) et son implémentation est aussi un de nos travaux en cours pour supporter le SVC.

BILAN

J'ai été confronté à beaucoup de difficultés pendant toute la durée du stage. Une désorientation au début m'a pris des semaines avant que je puisse m'en sortir. Le contact avec le tuteur n'a pas toujours des réponses instantes et claires.

Le domaine sur lequel je travaille n'a pas beaucoup de référence et de documentation, ou ils ne sont pas accessibles au public. Les contenus de ces documents sont eux-mêmes difficile à comprendre, car ils parlent de façon trop technique ou générique.

La dépendance des dispositifs et les problèmes administratifs nous ont empêchés de pouvoir donner la solution complète le plutôt possible.

Sauf des problèmes mentionnés, ce stage de fin d'études est vraiment une bonne opportunité pour moi de rafraîchir les connaissances apprises et les appliquer à un des domaines les plus essentiels de la vie humaine.

Avec un esprit de toujours apprendre, j'ai eu la chance dans ce stage de découvrir le style d'un travail de recherche et d'avoir de bons contacts dans le domaine qui seront très précieux dans ma carrière future.

Références

- [1] M.U. Farooq, Muhammad Waseem, Anjum Khairi, Sadia Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)", dans *International Journal of Computer Applications*, Volume 111, No. 7, 2015
- [2] Hui Suo, Jiafu Wan, Caifeng Zou, Jianqi Liu, "Security in the Internet of Things: A Review", dans *International Conference on Computer Science and Electronics Engineering*, 2012
- [3] Rohde & Schwarz, "Vulnerabilities of LTE and LTE-Advanced Communication", https://www.rohde-schwarz.com/in/file/1MA245_2e_LTE_Vulnerabilities.pdf
- [4] 802.11 WLAN Roaming and Fast-Secure Roaming on CUWN, <http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/116493-technote-technology-00.html>
- [5] German Castignani, Andres Emilio Arcia Moret, Nicolas Montavont. "A study of the discovery process in 802.11 networks". ACM Sigmobile - *Mobile computing and communications review*, 2011, 15 (1), pp.25-36. <10.1145/1978622.1978626>. <hal-00609309>
- [6] About nl80211, <https://wireless.wiki.kernel.org/en/developers/documentation/nl80211>
- [7] Blake-Wilson, S.; Menezes, A. (1999), "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", *Public Key Cryptography, Lecture Notes in Computer Science*, 1560, Springer, pp. 154–170

Annexe

[La version complète de « Understanding 802.11 connection process » en anglais](#)

A basic configuration of a BSS (Basic Service Set) operating in infrastructure mode consists of a station (STA) and an access point (AP) attached to a backbone network.

I. Wi-Fi network discovering

There are two ways that a STA detects the presence of an AP:

- **Passive scan** - using beacon frame: the APs broadcast periodically the frames that identify themselves as well as provide information about the networks behind. We call this passive scan, because the Wireless NIC (WNIC) receives these frames without sending any data. Default beacon broadcasting interval is set to 100TU (1TU = 1024 µs).

FIGURE 4.5 Beacon frame structure

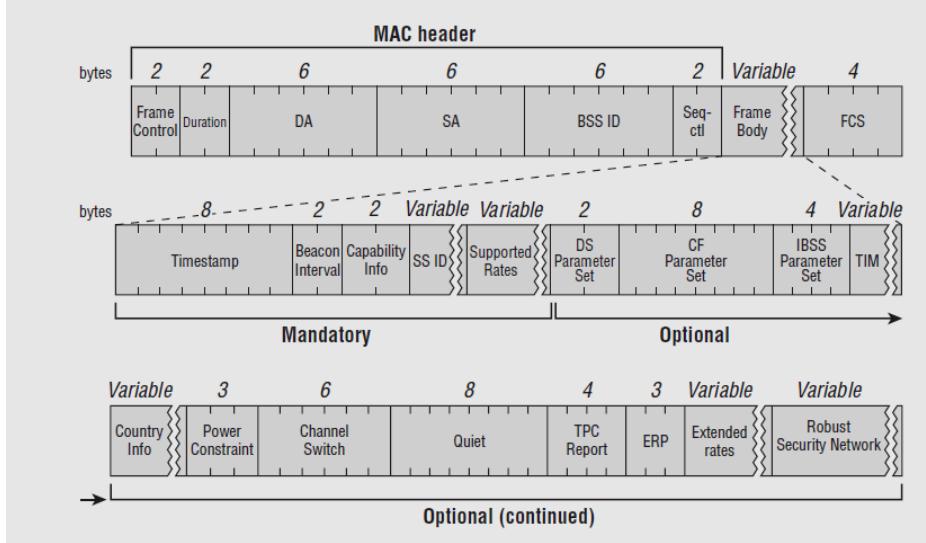


Figure 1.1 – Structure of a beacon frame

- **Active scan** - using probe request/response: the STA actively scans for APs by

sending probe request on each channel. Probe request is a broadcast Ethernet packet containing STA information and capabilities (supported data rates, channels). If an AP realizes that it has the **same capabilities**, an unicast packet is responded to the STA. MinCT is the maximum waiting time of a probe response, and MaxCT is the additional waiting time for other responses if there were any probe response received during MinCT.

A full network scan is an active scan that the probe is sent on every channel, trying to discover as many of access points as possible. This is obviously useful in a desktop environment, but unnecessarily being used in a real-time context.

Instead, the STA can perform a scan only on non-overlapping channels (ex. 1-6-11), on which it can be sure that there must be some access points operating, which are predefined by system architects.

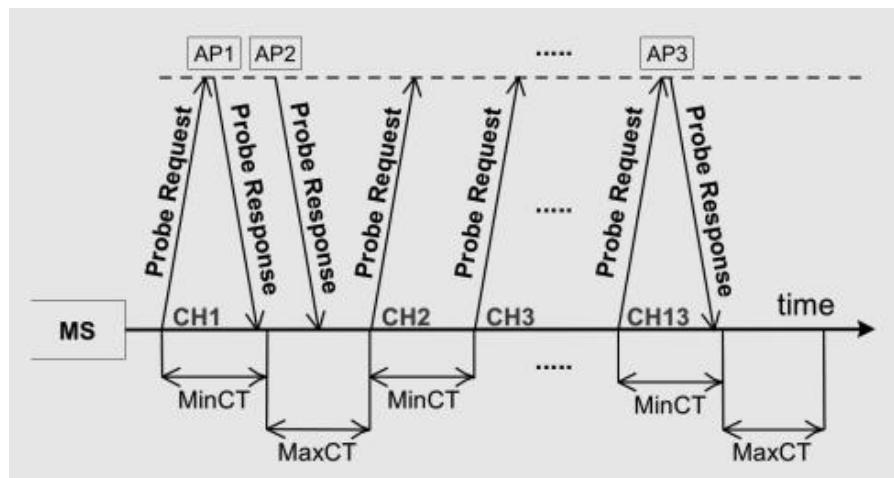


Figure 1.2 – Active scan process

In addition, MinCT and MaxCT can also be reconfigured, to be adapt with the current scan result, which reduces scanning time even better^[2].

I. Wi-Fi network authenticating and associating

1. Two basic phase of a connection

After an active scanning process, the STA has a list of “connectable” access points.

Suppose that our STA wants to connect to an access point AP. This process of two phases then has to be happened:

- **Authenticating phase:** this phase is used to authenticate a STA to an AP. In the past, there were two methods of authenticating:
 - o Open System: this method performs *no client verification*. At the birth of 802.11, this method allows any STA to connect with the AP without providing any password, creating an open network.*
 - o Pre-shared key: in this method, the STA and the AP agree on a passphrase that will be used to challenge the STA each time it wants to connect with the AP. This method was then deprecated because of its security vulnerabilities.

In this phase, the STA sends an *authentication request*, and the AP responds with an *authentication response* indicating the success of authentication process.

After authenticating itself to the AP, the STA goes into “*authenticated, unassociated state*”. Both the STA and the AP save this information.

*Nowadays, modern Wi-Fi standards use *Open System* as their default authentication method, along with an Extended Authentication Protocol (EAP) that happens after the association phase.

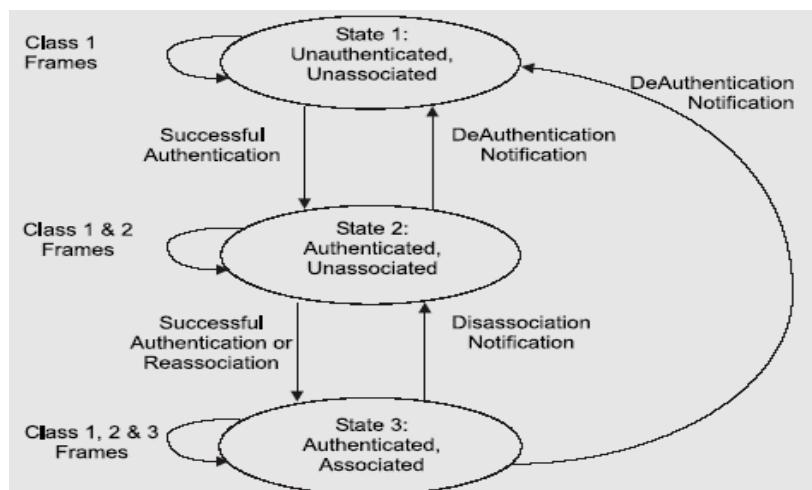


Figure 2.1 – 802.11 connection state machine

- **Association phase:** this phase happens when a STA has already been authenticated (aka being in “authenticated, unassociated” state) and decides to

connect to the desired AP. This is when the AP and the STA agree on the common parameters that will be used for the communication later on.

In the same way, the STA sends an *association request* to the AP it wants to connect, and the AP responds with an association response. Depend on the configuration, the AP can either accept or refuse the association request (ex. overloading, weak security policy...).

After successfully associated, the STA is in “authenticated, associated” state. From now on, it is connected to the Wi-Fi network.

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462 Authentication, SN=2443, FN=0	
2	0.000784	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462 Authentication, SN=2771, FN=0	
3	0.002428	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462 Association Request, SN=2444,	
4	0.007122	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462 Association Response, SN=2772	
5	0.995428 0.0.0.0		255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462 DHCP Discover - Transaction I	
6	2.996191 1.1.1.1		172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462 DHCP Offer - Transaction I	
7	2.998532 0.0.0.0		255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462 DHCP Request - Transaction I	
8	3.005016 1.1.1.1		172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462 DHCP ACK - Transaction I	

Figure 2.2 – Authentication and association frame exchanging

2. High-level authentication protocols

This is what happens after a STA has successfully been connected to a Wi-Fi network (authenticated, associated). If the security configuration is set, before that any data frame could be forwarded between the STA and the network, another “high-level” authentication phase will take place.

Again, it is pre-shared key with 4-way hand shake or key exchanging protocol through *EAPoL* (Extended Authentication Protocol over LAN).

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462 Authentication, SN=2465, FN=0, Flags=...,	
2	0.000783	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462 Authentication, SN=275, FN=0, Flags=...,	
3	0.002579	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462 Association Request, SN=2466, FN=0, Flags=,	
4	0.007765	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462 Association Response, SN=276, FN=0, Flags=,	
5	0.012140	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Identity	
6	0.052605	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAPOL	2462 Start	
7	0.055257	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Identity	
8	0.061197	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAP	2462 Response, Identity	
9	0.081403	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
10	0.117423	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Client Hello	
11	0.145293	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
12	0.167145	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAP	2462 Response, Protected EAP (EAP-PEAP)	
13	0.183267	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
14	0.196221	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAP	2462 Response, Protected EAP (EAP-PEAP)	
15	0.201527	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
16	0.210076	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Certificate, Client Key Exchange - Change C1	
17	0.220032	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
18	0.222784	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAP	2462 Response, Protected EAP (EAP-PEAP)	
19	0.227233	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
20	0.291267	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Application Data, Application Data	
21	0.291661	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Application Data, Application Data	
22	0.295816	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
23	0.297769	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Application Data, Application Data	
24	0.304661	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
25	0.313817	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
26	0.315947	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Application Data, Application Data	
27	0.321376	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Request, Protected EAP (EAP-PEAP)	
28	0.323863	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	TLSv1	2462 Application Data, Application Data	
29	0.328766	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAP	2462 Success	
30	0.330360	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAPOL	2462 Key (Message 1 of 4)	
31	0.334225	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAPOL	2462 Key (Message 2 of 4)	
32	0.338645	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	EAPOL	2462 Key (Message 3 of 4)	
33	0.341932	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	EAPOL	2462 Key (Message 4 of 4)	

Figure 2.3 – WPA-EAP key exchanging after successful association

II. Dis-authentication, Roaming and Fast Secure Re-association

1. Dis-authentication

- **Active leaving:** when a STA wants to leave an AP, or an AP decides to stop serving the STA, a dis-authentication frame will be sent with info about the reason. After being notified, the status of the STA is set back to “unauthenticated, unassociated”.
- **Passive leaving:** in case of brutal corruption of AP, the STA must wait for a certain time before notice that the signal has been lost. How? Remember of beacon frame. The beacon frame is the way the AP confirms its presence to the STA. Missing AP beacon frame, the STA will try to scan all the network to find the AP again. If the AP is not found in the result, the connection is considered to be lost.

3. Re-association

When a lost-signal STA tries to reconnect itself to the network, it must completely perform those phases that described above, including the key exchanging protocol to obtain a new encryption key. The only difference is that, if the new AP is in the same network with the old one, a re-association frame is sent instead.

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:92	84:78:ac:f0:2a:92	802.11	2437	Authentication, SN=2698,
2	0.001419	Cisco_f0:2a:92	Aironet_b7:ab:5c	84:78:ac:f0:2a:92	802.11	2437	Authentication, SN=3898,
3	0.003446	Aironet_b7:ab:5c	Cisco_f0:2a:92	84:78:ac:f0:2a:92	802.11	2437	Reassociation Request, S
4	0.009580	Cisco_f0:2a:92	Aironet_b7:ab:5c	84:78:ac:f0:2a:92	802.11	2437	Reassociation Response,
5	0.015767	Cisco_f0:2a:92	Aironet_b7:ab:5c	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 1 of 4)
6	0.030933	Aironet_b7:ab:5c	Cisco_f0:2a:92	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 2 of 4)
7	0.037448	Cisco_f0:2a:92	Aironet_b7:ab:5c	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 3 of 4)
8	0.052108	Aironet_b7:ab:5c	Cisco_f0:2a:92	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 4 of 4)

Figure 3.1 – Re-association frame exchanging with PSK

Note that a STA can only associate with one AP at a time. The re-association request notifies the new AP that the STA is roaming within the network from another AP. This allows the network to reallocate the necessary resources for handling the new STA (ex. remove the information of the STA in the previous AP).

4. Fast-Secure Roaming

Fast-Secure Roaming attempts to reduce the frames exchanged during re-association phase, which reduces the overall roaming time. This technique is based on caching necessary

information at first-time-authentication, then reuse it later. These information are embedded inside the re-association request and response,

There are several methods to achieve Fast-Secure Roaming. Cisco invented its own algorithm named Cisco Centralized Key Management (CCKM) which is only supported on some Cisco devices. Nevertheless, CCKM is one of the-best-out-there for fast secure roaming, thanks to these following advantages:

- CCKM is the fastest fast-secure roaming method mostly deployed on enterprise WLANs. Clients do not need to go over a key management handshake in order to derive new keys when a move between APs takes place, and are never again required to perform a full 802.1X/EAP authentication with new APs during the client lifetime on this WLAN.
- CCKM supports all of the encryption methods available within the 802.11 standard (WEP, TKIP, and AES), in addition to some legacy Cisco proprietary methods still used on legacy clients.

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:93	84:78:ac:f0:2a:93	802.11	2437	Authentication, SN=2714, FN=
2	0.002658	Cisco_f0:2a:93	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	Authentication, SN=2723, FN=
3	0.004702	Aironet_b7:ab:5c	Cisco_f0:2a:93	84:78:ac:f0:2a:93	802.11	2437	Reassociation Request, SN=2
4	0.010575	Cisco_f0:2a:93	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	Reassociation Response, SN=
5	0.843240	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:93	802.11	2437	QoS Data, SN=2717, FN=0, Flg=
6	0.849798	Cisco_f5:4a:40	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	QoS Data, SN=66, FN=0, Flg=

Figure 3.2 – Re-association with CCKM enabled

III. Improvement suggestions

Linux's default behavior tends to perform a full network scan each time it recognizes any change (a disconnection, reconnection, beacon of new APs), which slowdowns the connection process. This can be simply replaced by a fix frequency scan.

In case a full network scan is required, an adaptive timer scanning can be performed.

CCKM should be also taken into consideration for purpose of faster roaming.

REFERENCES

[1] Fast secure roaming

<https://supportforums.cisco.com/document/9879826/80211-wlan-roaming-and-fast-secure-roaming-cuwn>

[2] **German Castignani, Andres Emilio Arcia Moret, Nicolas Montavont.** A study of the discovery process in 802.11 networks. ACM Sigmobile - Mobile computing and communications review, 2011, 15 (1), pp.25-36. <10.1145/1978622.1978626>. <hal-00609309>



La solution du SVC modifié

I. Solution

1. La version modifiée du SVC

Avec seulement 3 échanges, on propose une nouvelle approche basé sur le principe du « serrement en 3 phases » qui garde encore les idées basiques du SVC.

Dans le diagramme suivant, on distingue les données « dans les crochets », avec celles qui sont en dehors. Les dernières sont des données de la négociation DH-STS, qui assurent l'autorisation du service, deviennent optionnelles en raison du découplage de service (à expliquer dans la section suivante). Les premières sont des données d'authentification, dépendantes de l'application qui utilise SVC.

Dans la vérification de l'authenticité, l'approche « challenge – proof » est introduite pour éviter tout type d'attaque par rejet. Le DH-STS est forcément imperméable, seulement vulnérable à quelques attaques du type « unknown key-share »⁶.

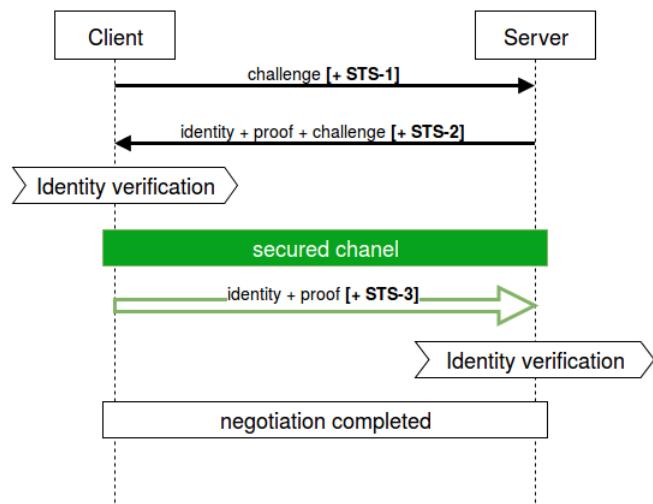


Figure 3.2 – Echange de messages dans la version modifiée du SVC

À comparer avec la version originale, la nouvelle montre des avantages supplémentaires:

- Moins de nombre des échanges (3 contre 4)

⁶ Blake-Wilson, S.; Menezes, A. (1999), "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", *Public Key Cryptography*, Lecture Notes in Computer Science, **1560**, Springer, pp. 154–170

- Résistant à l'attaque par rejetu

1. L'architecture service-applications

À la naissance, SVC est vise à servir une communication interne, c'est-à-dire une communication entre les instances d'une même application. Dans l'environnement multitâche, cette conception montre des défauts:

- chaque connexion doit manager ses propres (mais avec une même politique) paramètres de sécurité, chiffrement. Si on a plusieurs connexions vers une même hôte, toutes les étapes de négociation se refont.
- La mise à jour des protocoles de chiffrement du SVC impose la recompilation de toutes les applications qui l'utilisent.

L'idée est de découper la couche « application », qui est en charge de l'authentification et la communication de données, avec une seule instance de la couche « daemon », qui s'occupe des services de chiffrement et de négociation. Comme ça, on remédié tout de suite les problèmes posés :

- Les services managent toutes les connexions depuis et vers des hôtes. Quand une nouvelle connexion vers une même hôte est détectée, on utilise le service correspondant pour profite d'un canal déjà sécurisé, laisse passer les échanges de clés.
- La mise à jour se fait en simplement replaçant et redémarrant l'instance du daemon. Toutes les applications restent intouchées.

Dans le diagramme ci-dessous se trouvera l'architecture de la solution :

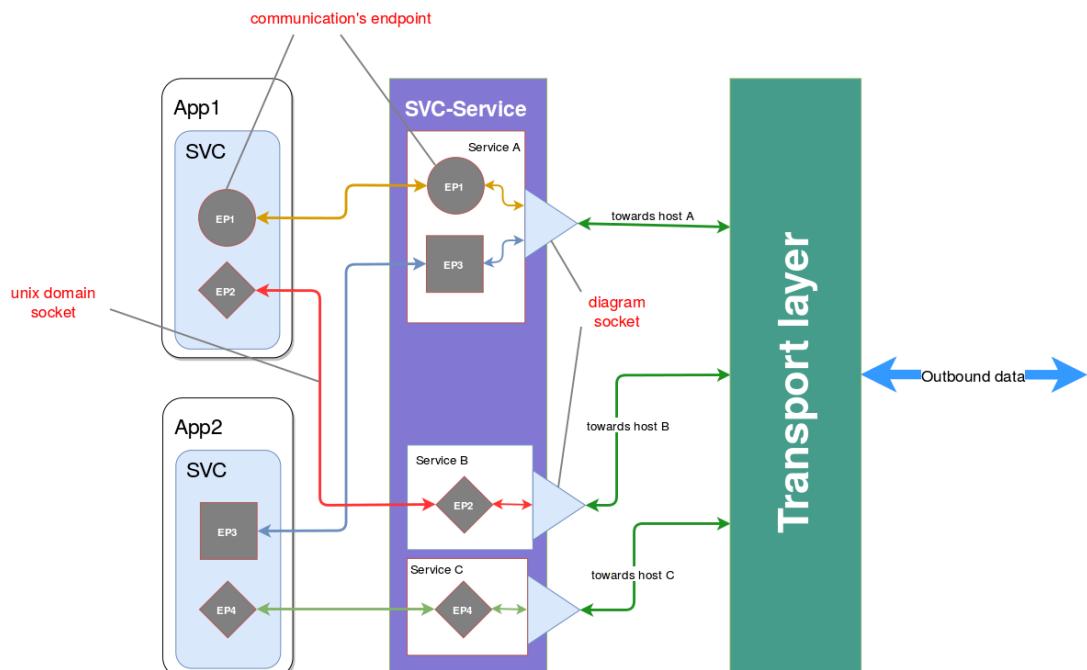


Figure 3.3 – L'architecture service-applications du SVC

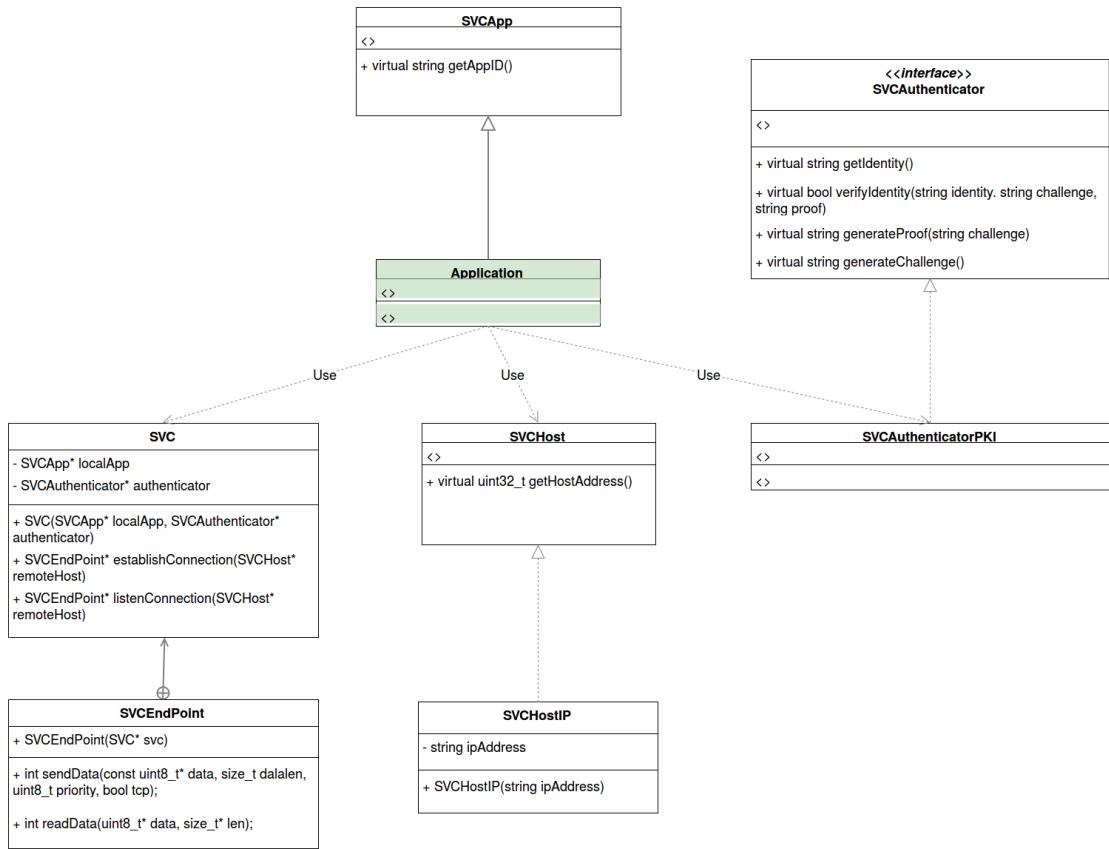


Figure 3.4 – Le diagramme de classe côté application

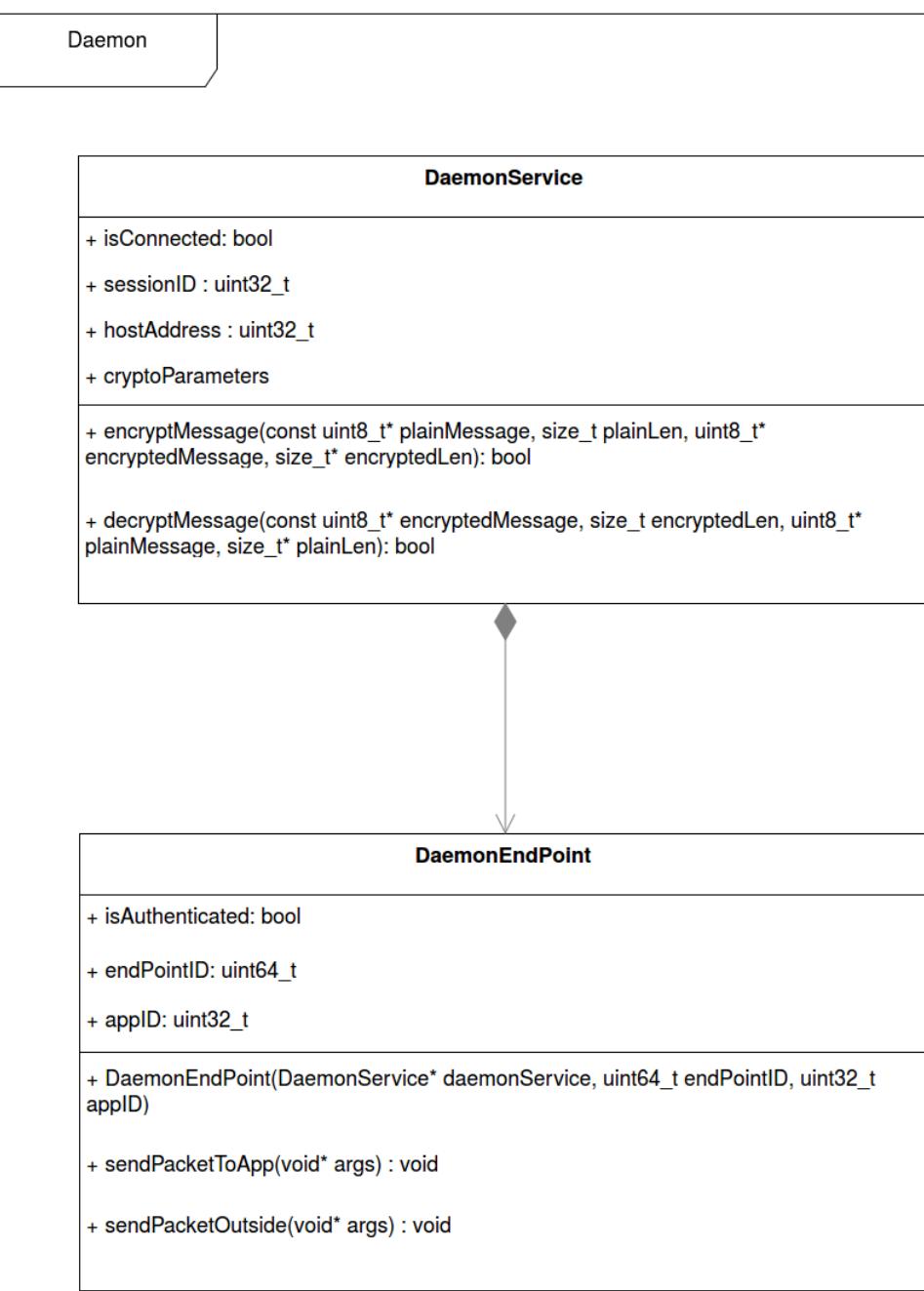


Figure 3.5 – Le diagramme classe côté service

2. La structure des trames SVC

Une trame SVC est encapsulée dans la partie de données d'une trame UDP. Il y a deux types de trames SVC : les trames de commandes servent à établir et gérer la connexion, les trames de données quant à elles contiennent les données chiffrées des applications. Une trame du type « données » est toujours chiffrée. Le format de données dans ce type de trame est géré par l'application.

Quel que soit le type de trame, toutes les trames SVC commencent par :

- **Identité de session** – 4 octets: identité unique de la session. Une session est une connexion établie entre deux services (ou bien deux hôtes), qui partage les mêmes paramètres de chiffrement avec tous les chemins de connexion.
- **Identité d'extrémité** – 8 octets : identité d'une extrémité de connexion. Un service SVC utilise cette identité pour distinguer les paquets viennent d'une même hôte. Un chemin de connexion est ce qui connecte 2 extrémités de communication, et ces extrémités partagent une identité identique.

L'utilisation de l'identité de session et d'extrémité permet la reprise rapide de connexion en cas de reconnexion.

Après l'identité de session et d'extrémité, succède un octet qui détermine le type de la trame et contient des paramètres pour la gestion de données.

- Données/commande: 7th bit, égale 1 si la trame est du type commande, 0 si la trame contient des données
- Réponse depuis daemon: 6th bit, égale 1 si la trame reçue est une notification du daemon local
- Pas d'usage pour le moment : 5th et 4th bits
- Chiffré: 3th bit, égale 1 si la partie de données de la trame est chiffrée
- TCP utilisé: 2nd bit, égale 1 si cette trame requiert une garantie de livraison, à lire et assurer par la couche de transport
- Priorité : 1^{er} et 0th bits, la priorité est parmi URGENT, HIGH, NORMAL, LOW, qui sera traitée par la couche transport

Une visualisation d'une trame SVC est démontré comme ci-dessous :

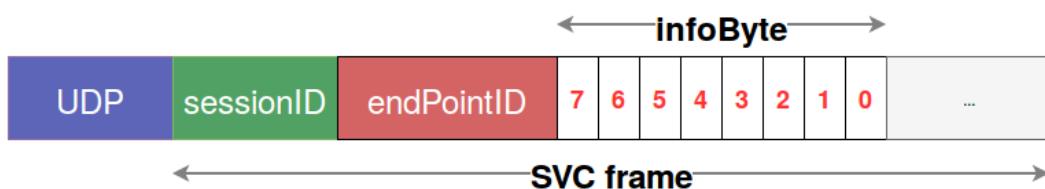


Figure 3.4 – La structure générale d'une trame SVC

Une trame de données du SVC contient, après l'entête, 4 octets de la longueur de la charge utile de données, suivie par la charge elle-même et le HMAC (l'algorithme et la longueur du HMAC dépend de la version du SVC). Une trame de données doit être toujours chiffrée.



Figure 3.5 – Une trame de données du SVC

Une trame de commande du SVC commence par l'identité de commande (CID), suivie par le nombre de paramètres (PRC). Le reste est la partie des paramètres, dans laquelle chaque paramètre est précédé par 2 octets de longueur. Sauf les commandes d'initialisation de connexion qui ne sont pas chiffrées, toutes les commandes restant sont chiffrées, et se terminent par un HMAC.

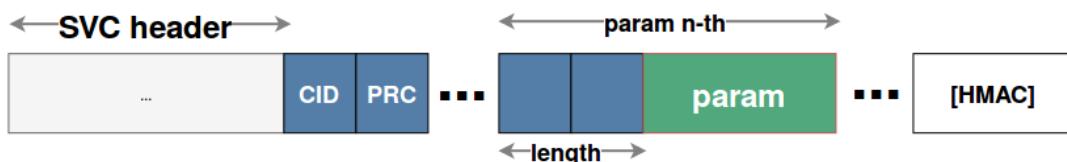


Figure 3.6 – Une trame de commande du SVC

3. Le détail d'un processus d'initiation de connexion

Une application crée une instance de SVC et demande la connexion. Dans la demande contient l'adresse de l'hôte à distance, qui sera utilisée par le « daemon » pour décider s'il doit créer un nouveau service (chaque service est en charge de connexion à un hôte unique).

À noter qu'il y a une seule instance du « daemon » qui s'exécute dans le background et qui est en charge de toutes instances du service. Pour pouvoir distinguer les demandes de connexion des applications différentes, le STEP_1 doit lui communiquer une identité de l'application (appID). Cette identité est liée à l'application et n'est pas au client sur lequel l'application s'exécute.

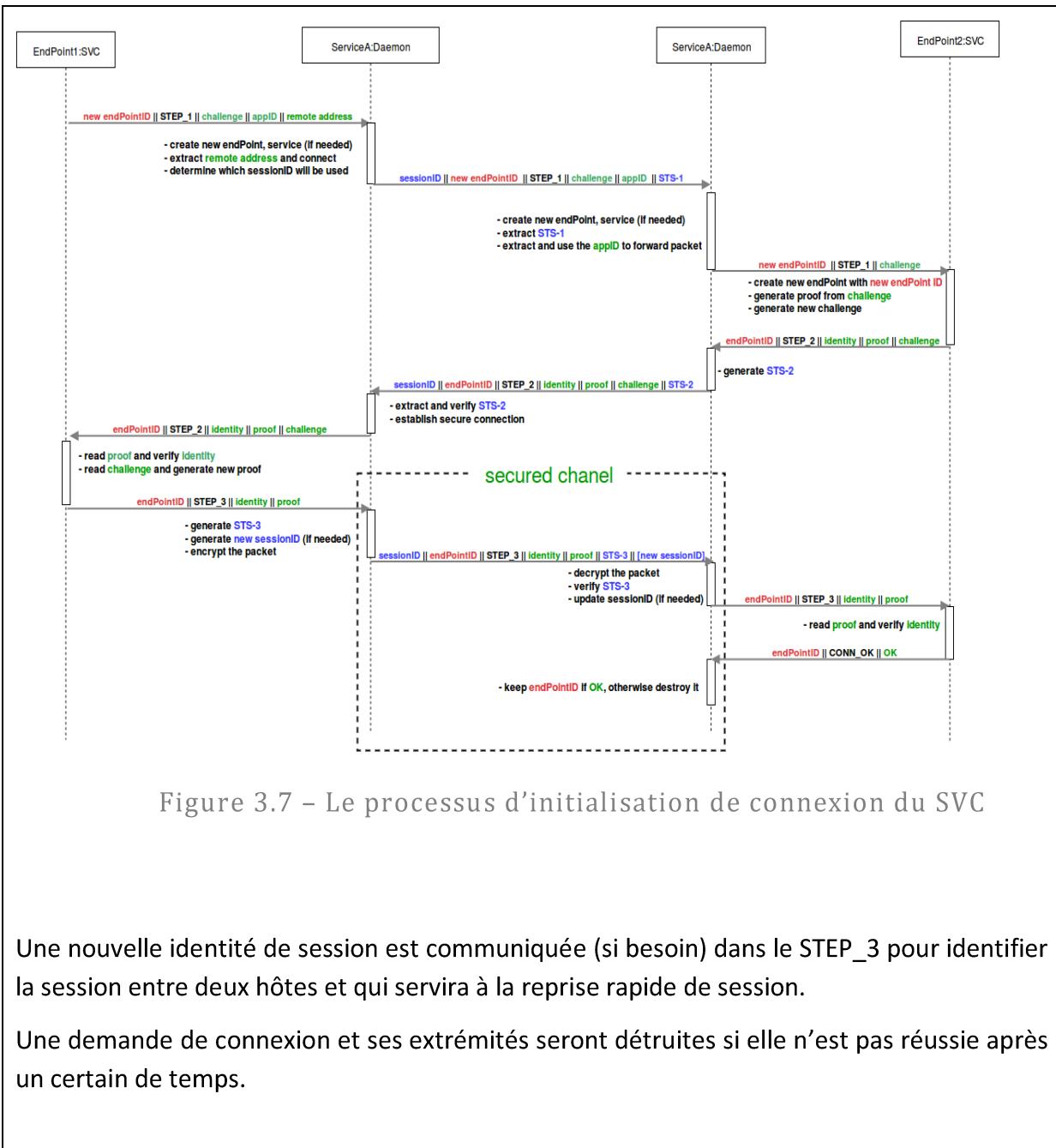


Figure 3.7 – Le processus d'initialisation de connexion du SVC

Une nouvelle identité de session est communiquée (si besoin) dans le STEP_3 pour identifier la session entre deux hôtes et qui servira à la reprise rapide de session.

Une demande de connexion et ses extrémités seront détruites si elle n'est pas réussie après un certain de temps.