

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLICUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة عمّار ثليجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT



كلية العلوم
FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatiques

Option : Réseaux, Systèmes et Applications Réparties

Par :

DJERADI Sarah
ABDELHAFIDI Fatima

THEME

**OPTIMISATION DE LA CONSOMMATION DES SERVICES DANS LE
CLOUD VEHICULAIRE**

Soutenu publiquement devant le jury composé de:

Mr T.BENDOUMA

M^{elle} F.BOUSBAA

Mr I.AMEURE

Mr N.LAGRAA

Mr B.BRIK

M.C.(B)

M.A.(A)

M.A.(B)

Professeur

M.A.(B)

Président

Examinatrice

Examineur

Encadreur

Co-encadreur

Année Universitaire 2015/2016

Résumé

Le Cloud Véhiculaire (CV) est un nouveau paradigme apparu pour permettre d'exploiter les des ressources Cloud dans les réseaux véhiculaires. Ce modèle combinant le Cloud Computing et les réseaux VANets a donné la possibilité aux véhicules intelligents d'offrir et de consommer des services Cloud selon des protocoles spécifiés.

Dans ce travail, nous avons étudié quelques protocoles d'offre et de consommation des services dans le Cloud véhiculaire et nous avons proposé aussi une amélioration au protocole DCCS-VC (Discovering and Consuming Cloud Services in Vehicular Cloud). Cette amélioration consiste en la sélection du meilleur service Cloud satisfaisant la demande du consommateur en utilisant une technique basée sur la logique floue. Enfin, nous avons comparé les résultats de simulation obtenus par la technique de la logique floue avec ceux obtenus grâce à la méthode paramétrable SAW (Simple Additive Weighting). Ces résultats ont montré l'efficacité de cette technique en termes de sélection des meilleures offres pour le consommateur.

Mots-clé:

VANet, Cloud Computing, Selection, DCCS-VS, logique floue.

Abstract

Vehicular Cloud (VC) is a new emerged paradigm allows exploiting Cloud resources in vehicular networks. Thus, the model combining Cloud Computing and VANETs gives the opportunity for intelligent vehicles to offer and consume cloud services according to specified protocols.

In this work, we studied some protocols of offer and services consumption in the vehicular Cloud and we proposed an improvement to the DCCS-VC protocol (Discovering and Consuming Cloud Services in Vehicular Cloud). This improvement consists in the selection of the best Cloud service satisfying consumer requirements using a technique based on fuzzy logic. Finally, we compared the simulation results obtained by the technique of fuzzy logic with those obtained by the configurable method SAW (Simple Additive Weighting). These results have shown the effectiveness of this technique in terms of selection of the best offers for the consumer.

Keywords:

VANet, Cloud Computing, Selection, DCCS-VS, Fuzzy logic.

Remerciements

Toute notre gratitude et nos remerciements au bon DIEU qui nous a donné la force, le courage et la volonté d'élaborer ce travail.

Nous tenons tout d'abord à exprimer nos sincères remerciements à nos encadreurs Prof. Nasreddine LAGRAA et Mr. Bouziane BRIK , pour leurs aides sans limite, leurs disponibilités et leurs précieux conseils et orientations sans lesquels ce travail n'aurait pas vu le jour. Ils nous ont ainsi offert l'opportunité de faire nos premiers pas sur un travail de recherche passionnant et prometteur : le monde des réseaux véhiculaires.

Nous remercions les membres de jury qui ont accepté de juger ce travail :

- Mr BENDOUMA Tahar
- Melle BOUSBAA Fatima
- Mr AMEURE Ilyass

Nous remercions également tous ceux qui de près ou de loin, nous ont accompagné et soutenu pour mener à bien ce travail.

Nous remercions aussi nos professeurs, nos collègues, nos familles et toutes les personnes qui nous ont aidé durant nos études universitaires.

Table des matières

| | |
|---|----------|
| Introduction générale | 1 |
| 1 Introduction au Cloud Computing | 3 |
| 1.1 Introduction | 3 |
| 1.2 Historique du Cloud Computing | 3 |
| 1.3 Définition | 5 |
| 1.4 Caractéristiques du Cloud Computing | 5 |
| 1.5 Acteurs du Cloud Computing | 5 |
| 1.5.1 Prestataires des services (fournisseurs) | 6 |
| 1.5.2 Ressources hébergées | 6 |
| 1.5.3 Consommateurs | 6 |
| 1.6 Architecture du Cloud Computing | 7 |
| 1.7 Services du Cloud Computing | 7 |
| 1.7.1 IaaS (Infrastructure as a Service / Infrastructure en tant que service) | 7 |
| 1.7.2 PaaS (Platform as a Service / Plate-forme en tant que service) | 7 |
| 1.7.3 SaaS (Software as a Service /Logiciel en tant que service) | 8 |
| 1.7.4 Types du Cloud Computing | 8 |
| 1.8 Cloud Computing Mobile | 9 |
| 1.8.1 Types du Cloud Computing Mobile | 9 |
| 1.9 Défis du Cloud Computing | 10 |
| 1.9.1 Problème de qualité de service | 10 |
| 1.9.2 Problème de sécurité | 10 |

| | | |
|----------|---|-----------|
| 1.9.3 | Sélection des services | 11 |
| 1.10 | Approches de sélection de services Cloud | 11 |
| 1.10.1 | Approches basées sur les méthodes MCDM (Multi-Criteria Decision-Making) | 11 |
| 1.10.2 | Approches basées sur les méthodes d'optimisation | 13 |
| 1.10.3 | Approches basées sur la logique | 13 |
| 1.11 | Conclusion | 13 |
| 2 | Cloud Computing Véhiculaire | 15 |
| 2.1 | Introduction | 15 |
| 2.2 | Réseaux véhiculaire (VANet) | 15 |
| 2.2.1 | Définition d'un réseau véhiculaire | 16 |
| 2.2.2 | Caractéristiques des réseaux VANet | 17 |
| 2.2.3 | Applications des réseaux véhiculaire | 18 |
| 2.3 | Cloud Computing Véhiculaire | 18 |
| 2.3.1 | Définition du Cloud Computing Véhiculaire | 19 |
| 2.3.2 | Services du Cloud Computing Véhiculaire | 19 |
| 2.3.3 | Architecture du Cloud Computing Véhiculaire | 20 |
| 2.3.4 | Protocoles du Cloud véhiculaire | 21 |
| 2.3.5 | Défis du Cloud Computing Véhiculaire | 31 |
| 2.4 | Conclusion | 33 |
| 3 | Amélioration du protocole DCCS-VC | 34 |
| 3.1 | Introduction | 34 |
| 3.2 | Partie 1 : Logique floue | 34 |
| 3.2.1 | Historique de la logique floue | 35 |
| 3.2.2 | Concept de la logique floue | 35 |
| 3.2.3 | Fonctions d'appartenance | 36 |
| 3.2.4 | Opérateurs flous | 37 |
| 3.2.5 | Règles floues | 38 |
| 3.2.6 | Fuzzification | 39 |

| | | |
|-------|--|-----------|
| 3.2.7 | Déffuzification | 39 |
| 3.2.8 | Inférence | 39 |
| 3.2.9 | Applications de la logique floue | 40 |
| 3.3 | Partie 2 : Notre approche : Sélection de services par la logique floue | 41 |
| 3.3.1 | Construction de la base des règles | 41 |
| 3.3.2 | Environnement de simulation | 43 |
| 3.3.3 | Paramètres de simulation et étude comparative | 45 |
| 3.3.4 | Conclusion | 49 |
| | Conclusion générale et perspectives | 50 |
| | Bibliography | 53 |

Liste des figures

| | | |
|----------|--|-----------|
| 1 | Introduction au Cloud Computing | 3 |
| 1.1 | Historique du Cloud Computing | 4 |
| 1.2 | Architecture du Cloud Computing | 6 |
| 1.3 | Les types du Cloud Computing | 9 |
| | | |
| 2 | Cloud Computing Véhiculaire | 15 |
| 2.1 | Un réseau VANet. | 16 |
| 2.2 | Éléments constituant le véhicule intelligent | 17 |
| 2.3 | Cloud véhiculaire | 20 |
| 2.4 | Véhicule utilisant le Cloud | 21 |
| 2.5 | Cloud hybride | 21 |
| 2.6 | Vue générale sur CROWN | 22 |
| 2.7 | Déroulement du protocole CROWN | 23 |
| 2.8 | Vue générale sur RCS-VC | 25 |
| 2.9 | Déroulement du protocole RCS-VC | 26 |
| 2.10 | Vue générale sur DCCS-VC | 27 |
| 2.11 | Déroulement du protocole DCCS-VC | 29 |
| 2.12 | Cloud véhiculaire basé sur un cluster | 30 |
| | | |
| 3 | Amélioration du protocole DCCS-VC | 34 |
| 3.1 | Fonction d'appartenance avec trois ensembles pour la variable « température ». | 36 |

| | | |
|-----|---|----|
| 3.2 | Veins reliant OMNET++ et SUMO. | 45 |
| 3.3 | Écart de SAW et logique floue | 49 |

Liste des tableaux

| | | |
|----------|--|-----------|
| 1 | Introduction au Cloud Computing | 3 |
| 1.1 | Exemple de la méthode SAW. | 12 |
| 2 | Cloud Computing Véhiculaire | 15 |
| 2.1 | Attributs des services offerts par le STAR | 22 |
| 2.2 | Attributs spécifié par un véhicule fournisseur dans RCS-VC. | 25 |
| 2.3 | Attributs requis par un véhicule consommateur dans RCS-VC. | 26 |
| 2.4 | Attributs spécifié par un véhicule fournisseur dans DCCS-VC. | 28 |
| 2.5 | Attributs requis par un véhicule consommateur(DCCS-VC). | 28 |
| 2.6 | Comparaison entre les protocoles de Cloud véhiculaire | 31 |
| 3 | Amélioration du protocole DCCS-VC | 34 |
| 3.1 | Différentes formes d'une fonction d'appartenance | 37 |
| 3.2 | Ensemble des règles. | 42 |
| 3.3 | Ensemble de règles optimisées. | 43 |
| 3.4 | Paramètres de simulation. | 46 |
| 3.5 | Résultats de simulation | 48 |

Introduction générale

Les réseaux véhiculaires **VANets**, sont des réseaux de communication entre véhicules intelligents équipés de processeurs, de périphériques réseau et de différents types de capteurs. Ils sont basés sur la communication et l'échange des informations entre les véhicules, et entre les véhicules et des éléments de la route (Exemple: les panneaux de signalisations, les feux tricolores...) ou des éléments de réseaux externes (satellites, antennes, Internet) [1].

Parmi les applications offertes par les réseaux **VANets**, il existe celles liées à la sécurité routière (alerte accident, alerte ralentissement, alerte déviation... etc.), les applications liées au confort tel que le guidage par **GPS**, la possibilité de jouer en réseau entre les passagers des voitures [1]... etc. En plus, ces dernières années, les réseaux **VANets** ont été combinés avec le concept du Cloud Computing pour donner naissance au Cloud Véhiculaire qui est une application récente des **VANets**.

Le Cloud Computing est un concept majeur faisant référence à l'utilisation de la mémoire et des capacités de calcul des ordinateurs et des serveurs répartis dans le monde entier liés par un réseau, tel Internet. Il imprègne tant notre quotidien que certains l'utilisent sans s'en rendre compte. Pour beaucoup, la vie sans le Cloud ne serait plus envisageable : sans lui, plus de Facebook, de Twitter ni de Gmail, mais le Cloud a également révolutionné le monde de l'entreprise. Aujourd'hui, des millions de sociétés du monde entier s'appuient sur des services Cloud pour presque tout : création et sauvegarde de documents, comptabilité ... etc [2].

Dans ces dernières années, le Cloud computing est devenu un domaine de recherche qui a attiré beaucoup de chercheurs, ce qui a permis l'apparition de nouveaux types de Cloud tel que le Cloud Mobile et le Cloud véhiculaire qui était le résultat de la fusion entre le Cloud Computing et les **VANets**.

Dans le Cloud véhiculaire, les véhicules intelligents peuvent être à la fois des utilisateurs du Cloud et des fournisseurs de services en permettant aux véhicules qui possèdent des ressources d'agir comme serveurs Cloud mobiles et l'offrir à d'autres véhicules.

Cette exploitation des ressources dans les réseaux véhiculaires a donné naissance au Cloud Véhiculaire [1].

Pour traiter l'offre et la consommation des services dans le Cloud plusieurs protocoles ont été proposés, tels que le protocole **CROWN**(disCoveRing and cOnsuming services **WithiN** vehicular Cloud) [3] le protocole **RCS-VC** (**R**enting out and **C**onsuming **S**ervices in **V**ehicular **C**louds based on **LTE-A**)[4], le protocole **DCCS-VS** (**D**iscovering and **C**onsuming **C**loud **S**ervices in **V**ehicular **C**louds) [5]... etc.

Cependant, la sélection des meilleurs services dans le Cloud véhiculaire selon les critères et les contraintes des consommateurs est un problème qui n'a pas été traité. Dans ce travail, nous avons proposé une technique de sélection de services non paramétrable à base de la logique floue [6] et de l'intégrer au protocole d'offre et de consommation DCCS-VS, Ce qui a permis de rendre meilleure la sélection des services Cloud demandés par le consommateur. Afin de tester la validité de cette nouvelle technique, nous avons comparé ses performances avec celles obtenues par la méthode **SAW** (**S**imple **A**dditive **W**eighting)[7].

Ce mémoire est composé de trois chapitres organisés comme suit :

- Le premier chapitre présentera une vue générale sur le concept du Cloud Computing.
- Le deuxième chapitre décrit le concept du Cloud véhiculaire, d'abord nous commençons par présenter les réseaux véhiculaires VANet, ensuite, nous décrivons le Cloud véhiculaire, VC, et les principaux concepts qui le caractérisent tel que son architecture, ses applications et les protocoles proposés.
- Le troisième chapitre est consacré à la présentation de notre travail, nous décrivons en premier lieu le concept de la logique floue, Au deuxième lieu, nous présentons avec détails le principe de l'idée proposée pour l'amélioration du protocole DCCS-VC ainsi qu'une étude comparative par simulation entre la technique de la logique floue proposée et la technique SAW.

Chapitre 1

Introduction au Cloud Computing

1.1 Introduction

Indéniablement, la technologie de l'Internet se développe de manière exponentielle depuis sa création. Actuellement, une nouvelle « tendance » à fait son apparition dans le monde des **TICs** (**T**echnologies de l'**I**nformation et de la **C**ommunication), il s'agit du Cloud Computing. Cette technologie offre des occasions aux sociétés de réduire les coûts d'exploitation des ressources matériels et logiciels par leurs utilisations directement à travers un réseau qui est généralement le réseau d'Internet. Divers fournisseurs comme **Google**, **Amazon**, **IBM** offrent une vaste gamme de services de Cloud Computing.

Dans ce chapitre, nous allons présenter les notions de base du Cloud Computing, son historique, sa définition, ses caractéristiques, ses acteurs, son architecture globale, les différents services et les types du Cloud. Ensuite, nous allons définir le Mobile Cloud Computing et les défis qui rencontrent le Cloud Computing. Enfin, nous allons donner une vue générale sur les approches de sélection des services dans le Cloud computing.

1.2 Historique du Cloud Computing

Après l'apparition des mainframes dans les années 60, l'architecture client serveur dans les années 80, le Windows dans les années 90, la révolution Web dans les années 2000, nous avons entré dans une décennie du Cloud Computing [8](voir figure 1.1):

- **Les mainframes** : en 1960, la machine n'était qu'une simple interface de visualisation de l'information et tout son traitement était internalisé. Les grandes entreprises (ex : banques, compagnies d'assurances, compagnies aériennes. . . etc.)

utilisaient les mainframes qui étaient capables de répondre aux besoins de leurs utilisateurs.

- **L'architecture client/serveur** 1980 – 1990 : Les ordinateurs personnels font leur apparition dans les années 80, ce qui a permis la création de l'architecture Client/serveur qui est sous forme d'un ensemble d'ordinateurs personnels clients liés à un ordinateur serveur.
- **L'apparition du World Wide Web** : Le world wide web est né en 1991, et en 1993, le navigateur Internet qui rend le « www » célèbre a été créé par les américains. C'est un système hypertexte public fonctionnant sur Internet et qui permet de consulter, avec un navigateur, des pages mises en ligne dans des sites.
- **L'architecture distribuée** : Elle est apparue dans les années 90, c'est une architecture où toutes les ressources ne se trouvent pas au même endroit ou sur la même machine, contrairement à l'architecture centralisée. Un exemple de l'architecture distribuée est le Peer-to-Peer, c'est un modèle où chaque ordinateur est à la fois un serveur de données et client des autres [9].
- **Le Cloud Computing** : C'est un réseau d'ordinateurs qui donne la possibilité aux gens et aux entreprises d'exploiter des ressources matériels et logiciels à travers le réseau. Nous présentons la notion du Cloud Computing avec plus de détails dans les sections suivantes.

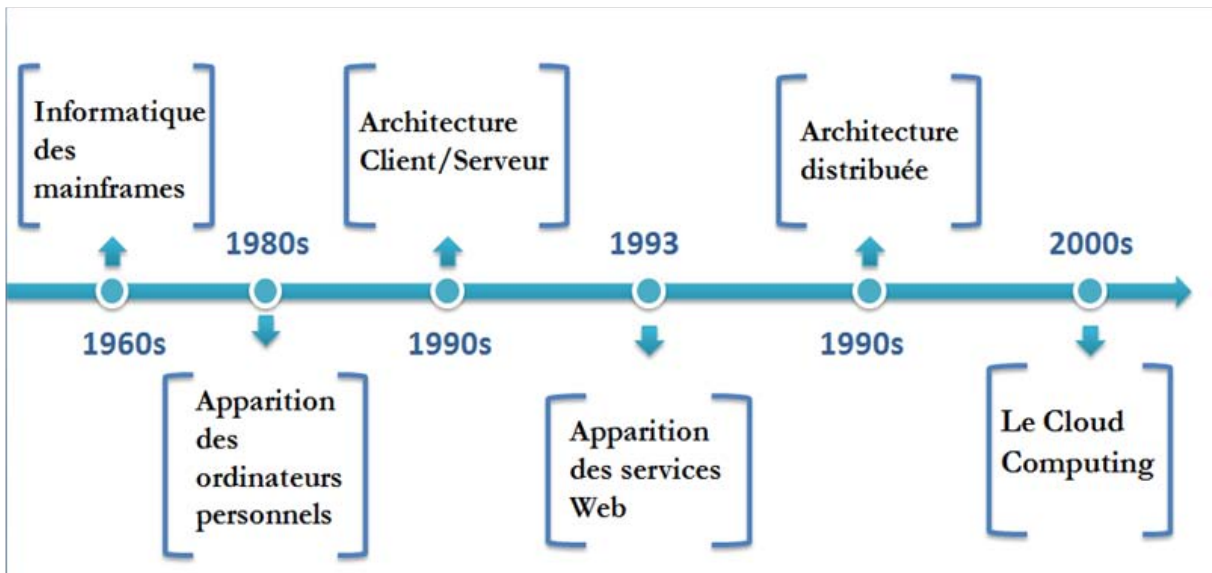


Figure 1.1: Historique du Cloud Computing

1.3 Définition

C'est un ensemble de ressources matériels et logiciels hébergés sur Internet, ces ressources sont stockées dans des serveurs et peuvent être accédées de manière transparente depuis des appareils (ordinateurs, téléphones) connectés à Internet [10].

1.4 Caractéristiques du Cloud Computing

Le Cloud Computing se caractérise par plusieurs propriétés dont on peut citer [11]:

- **Accès aux services par l'utilisateur à la demande** : L'utilisateur peut accéder aux ressources informatiques - physiques et virtuelles - à tout moment et selon ses besoins et ce, sans intervention « humaine » de la part du prestataire du service.
- **Accès au réseau large** : Les ressources informatiques sont accessibles n'importe où et n'importe quand par Internet ou autre réseau de télécommunication.
- **Une mise en commun des ressources** : Les ressources informatiques d'un fournisseur (ressources comprennent le stockage, le traitement et la bande passante réseau.) sont mises en commun pour servir plusieurs consommateurs sur un modèle multi-locataire, Ces ressources peuvent ainsi être affectées et réaffectées aux utilisateurs en fonction de la demande. L'utilisateur n'a habituellement ni connaissance ni contrôle du lieu exact des ressources qui lui sont fournies; il peut toutefois préciser le lieu à un plus haut niveau d'abstraction (par exemple, le lieu géographique ou le centre de données).
- **Élasticité rapide**: en fonction des besoins des clients, la mise en ligne des ressources et les capacités peuvent rapidement augmenter ou diminuer.
- **Facturation à l'usage** : La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Ainsi, une unité de traitement stoppée n'est pas facturée.

1.5 Acteurs du Cloud Computing

C'est l'ensemble des entités qui composent le Cloud Computing :

1.5.1 Prestataires des services (fournisseurs)

Un prestataire de services est toute entité physique qui offre des services tels que:

- **Amazon EC2** : Amazon Elastic Compute Cloud est un service Web qui fournit une capacité de calcul redimensionnable dans le Cloud. Destiné aux développeurs, il est conçu pour faciliter l'accès aux ressources de Cloud Computing à l'échelle du Web.
- **Google AppEngine** : C'est une plateforme de conception et d'hébergement d'applications web basée sur les serveurs de Google.

1.5.2 Ressources hébergées

C'est l'ensemble des ressources physiques et logiciels comme les serveurs, les moyens de stockages (les bases de données), les systèmes d'exploitation et les applications ... etc, offerts par les prestataires.

1.5.3 Consommateurs

C'est l'entité qui demande l'accès à des services du Cloud afin de les consommer.

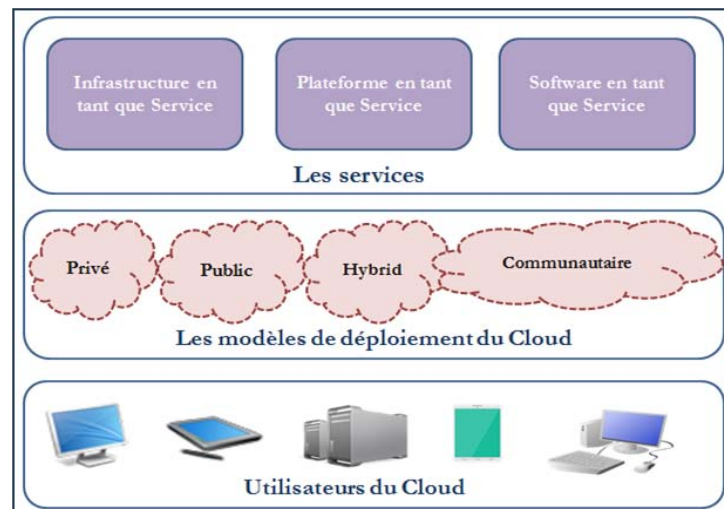


Figure 1.2: Architecture du Cloud Computing

1.6 Architecture du Cloud Computing

Les différentes couches du Cloud Computing d'après l'institut national des technologies et des standards (**NIST**) sont montrées dans la figure 1.2 [10]

1.7 Services du Cloud Computing

Généralement, on distingue trois classes de services qui peuvent être offerts en Cloud Computing [11] :

1.7.1 IaaS (Infrastructure as a Service / Infrastructure en tant que service)

Ce service offre des ressources informatiques comme les ressources de stockage et de calcul. Les utilisateurs d'une **IaaS** peuvent donc utiliser à la demande des serveurs virtuels situés dans des Datacenter pour déployer et exécuter des applications ou même des systèmes d'exploitation sans avoir à gérer les machines physiques (coûts de gestion, remplacement de matériel, climatisation, électricité. . .).

L'exemple emblématique de ce type de service est *Amazon Web Services* qui fournit du calcul (**EC2**), du stockage (**S3**, **EBS**) et des bases de données en ligne (**SimpleDB**).

1.7.2 PaaS (Platform as a Service / Plate-forme en tant que service)

Ce service offre à l'utilisateur une plateforme sur laquelle il peut développer, tester et exécuter ses applications.

La plate-forme supporte les langages de programmation, les outils et les bibliothèques qui peuvent être utilisés auprès des consommateurs pour générer et exécuter leurs propres applications. Comme exemple de **PaaS**, on peut citer **Google Apps Engine (GAE)** qui consiste en une plate-forme de conception et d'hébergement d'applications web basée sur les serveurs de **Google**.

1.7.3 SaaS (Software as a Service /Logiciel en tant que service)

Les applications sont fournies sous forme de services auxquels les utilisateurs se connectent via des logiciels dédiés ou un navigateur Internet. Les prestataires de solutions **SaaS** les plus connus sont :

- **Google avec Gmail**: Cette infrastructure fournit le service de messagerie à des centaines de millions d'utilisateurs et à des dizaines de millions d'entreprises.
- **Les réseaux sociaux *Facebook* et *Twitter*** qui permettent à leurs utilisateurs de publier du contenu, d'échanger des messages et d'utiliser une variété d'applications.

1.7.4 Types du Cloud Computing

Le Cloud Computing peut être subdivisé en quatre modèle de déploiement [10] (voir figure 1.3):

1.7.4.1 Cloud Public

C'est le Cloud qui est accessible par tout le monde et ouvert à tous, tels que ceux d'Amazon et de Microsoft, etc. Ce Cloud étant une infrastructure de serveurs virtuels hautement disponible et multi-Datacenter.

1.7.4.2 Cloud Privé

C'est le Cloud déployé pour une utilisation dédiée par un seul client. On distingue deux types de Clouds privés, Cloud interne, géré en interne par une entreprise pour ses besoins et le Cloud externe, dédié aux besoins propres d'une seule entreprise, mais dont la gestion est gérée par un prestataire externe.

1.7.4.3 Cloud Hybride

C'est le mode de déploiement faisant appel à la fois à des serveurs des deux types de Cloud public et privé.

1.7.4.4 Cloud Communautaire

L'infrastructure Cloud est partagée par plusieurs organisations pour les besoins d'une communauté qui souhaite mettre en commun des moyens (sécurité, conformité, etc..). Elle peut être gérée par les organisations ou par une tierce partie et peut être placée dans les locaux ou à l'extérieur.

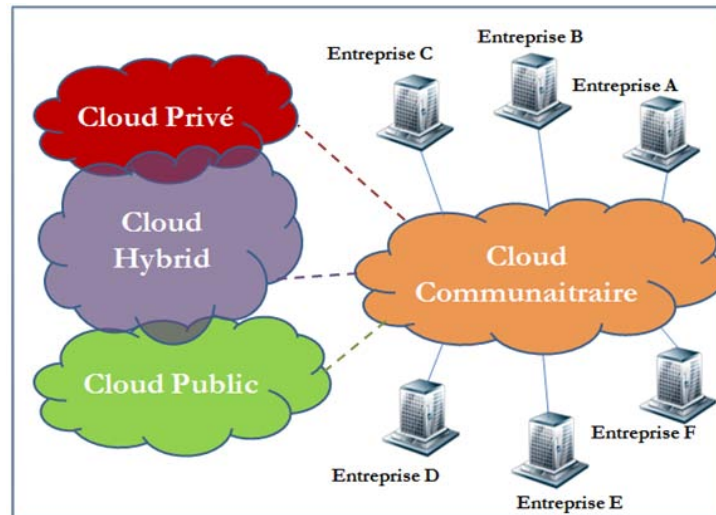


Figure 1.3: Les types du Cloud Computing.

1.8 Cloud Computing Mobile

Avec l'émergence des applications mobiles, le **Cloud Computing Mobile (CCM)** est introduit comme étant une intégration du Cloud Computing dans l'environnement mobile et qui peut être défini par une infrastructure où le stockage et le traitement de données se produisent en dehors de l'équipement mobile. Ce type de Cloud permet d'accéder à des services qui sont fournis par l'environnement du Cloud Computing à tout moment et n'importe où à partir des appareils mobiles [12].

1.8.1 Types du Cloud Computing Mobile

Parmi les types du **CCM**, on trouve [13] :

1.8.1.1 Mobile Vehicular Cloud

Le Cloud Véhiculaire offre des services tels que l'accès à l'internet, stockage... etc. Nous avons consacré le chapitre 2 pour bien le détailler.

1.8.1.2 Mobile Personal Cloud

Destiné aux Smartphones et les tablettes, le Personal Cloud diffère de celui des véhicules, car il offre un ensemble d'applications différentes et plus diversifiée comme par exemple, les applications des réseaux sociaux, applications de santé,... etc.

1.9 Défis du Cloud Computing

L'adoption du modèle du Cloud Computing soulève un certain nombre de défis [8]:

1.9.1 Problème de qualité de service

La qualité des services, notamment la performance et la disponibilité, est une des principales occupations des clients Cloud. A fin d'assurer un certain niveau de qualité à leurs clients, les fournisseurs Cloud disposent d'un système de contrôle pour enregistrer les **QoS** (**Q**uality of **S**ervice) et mettre à jour des Machines virtuelles (**MV**) du système, à chaque fois que la qualité de ces derniers est détériorée.

1.9.2 Problème de sécurité

Le problème de la sécurité dans le Cloud est un problème de confiance, les clients Cloud ne peuvent pas localiser où leurs données sont stockées ce qui augmente l'inquiétude des clients vis-à-vis des lois de préservation de confidentialité (selon le pays où sont stockées ses ressources). Parmi les principales menaces d'ordre sécuritaire sur le Cloud, on trouve:

- **Confiance entre fournisseur et client** : le fournisseur doit assurer au client la confidentialité des données stockées, mais le problème se définit dans le fait que les employés d'un fournisseur Cloud peuvent accéder aux données stockées du client et les vendre à des sociétés de marketing ou des entreprises concurrentes par exemple.
- **Client malveillant** : le partage de ressource entre les clients est une caractéristique sur lequel se base le Cloud et qui permet à plusieurs clients d'accéder

à la même ressource, cela augmente le risque qu'un client (pirate ou hacker) tente d'infiltrer les instances des services loués à d'autres clients pour voler ou inculquer des virus d'espionnage ou de destruction.

1.9.3 Sélection des services

Les prestataires de services fournissent aux utilisateurs des services à la demande grâce à Internet. Vu la disponibilité de plusieurs fournisseurs de services, un utilisateur de Cloud doit être en mesure de sélectionner les services appropriés en fonction de ses besoins. Plusieurs approches ont été proposées pour résoudre le problème de la sélection des services dans le Cloud Computing.

1.10 Approches de sélection de services Cloud

Le but du modèle de Cloud Computing est d'accroître les possibilités pour l'utilisateur du Cloud pour accéder à des applications, d'infrastructure et à des logiciels loués n'importe où et à tout moment. Vu la grande diversité des services Cloud disponibles, il est difficile pour le client de choisir les meilleurs services offerts ou même de déterminer une base valable pour leurs sélection.

Une des problématiques traitées dans le Cloud Computing est la sélection du service le plus adéquat tout en respectant les préférences de l'utilisateur. Dans cette partie, nous allons introduire les approches de sélection de services de Cloud.

1.10.1 Approches basées sur les méthodes MCDM (Multi-Criteria Decision-Making)

MCDM est un ensemble de techniques qui sont utiles pour l'aide à la décision multicritères. C'est une collection de méthodes permettant de choisir la meilleure solution ou la solution optimale parmi tout un ensemble de solutions [7].

Parmi les méthodes MCDM utilisées pour la sélection de services Cloud, on trouve:

1.10.1.1 Simple Additive Weighting (SAW)

SAW est une méthode de **MCDM** qui consiste en le calcul d'une simple somme (S_i) pondérée des critères de chaque alternative (A_i). La pondération est représentée par un vecteur de poids affectés par le consommateur.

Les résultats sont exprimés sous forme des scores : l'alternative qui a le score le plus élevé sera sélectionnée et préférée par le décideur. Les alternatives sont évaluées en utilisant l'équation suivante:

$$S_i = \sum_{j=1}^M w_j x_{ij} \quad i = 1..N$$

Où M : le nombre de critères. N : le nombre d'alternatives.

Et S_i est le score global qui correspond à l'alternative A_i , w_j est le poids du j^{me} critère et x_{ij} est la valeur du j^{ime} critère de l' i^{me} alternative A_i .

Exemple de la méthode SAW:

- **Objectif:** Sélection d'une automobile.
- **Critères:** Style, Fiabilité, Consommation du carburant
- **Alternatives:** Civic Coupe, Saturn Coupe, Ford Escort, Mazda Miata.

Le Tableau 1.1 représente l'évaluation de ces alternatives.

S_1 qui correspond au score global de l'alternative 1 (Civic Coupe) est calculé comme suit:

$$S_1 = \sum_{j=1}^3 w_j x_{1j}$$

$$S_1 = (0.3 * 7) + (0.4 * 8) + (0.3 * 9) = 8.4$$

Les autres Alternatives A_2 , A_3 et A_4 sont évaluées de la même façon.

Tableau 1.1: Exemple de la méthode SAW.

| Weight (Poids) | 0.3 | 0.4 | 0.3 | Score global |
|----------------|-------|-----------|---------------------------|--------------|
| Critères | Style | Fiabilité | Consommation du carburant | |
| Civic Coupe | 7 | 9 | 9 | 8.4 |
| Saturn Coup | 8 | 7 | 8 | 7.6 |
| Ford | 9 | 6 | 8 | 7.5 |
| Mazda | 6 | 7 | 8 | 7.0 |

D'après le tableau 1.1: $S_4 < S_3 < S_2 < S_1$ donc l'alternative A_1 du score S_1 (Civic coupe) est choisie.

1.10.2 Approches basées sur les méthodes d'optimisation

La sélection de service basée sur l'optimisation est définie comme (Trouver le service le plus adopté pour le client ou bien le fournisseur en maximisant ou minimisant un ou plusieurs critères, tout en respectant les contraintes) [7].

Plusieurs méthodes d'optimisation ont été appliquées pour la sélection de service comme la programmation dynamique¹, la programmation linéaire², les algorithmes gloutons³ [14]... etc.

1.10.3 Approches basées sur la logique

Les méthodes basées sur la logique sont parmi les méthodes utilisées pour la sélection de service dans le Cloud comme la logique du premier ordre [15], la logique descriptive (LDs)[16] et la logique floue [6].

- **La logique du premier ordre:** est un formalisme de représentation de connaissance dans laquelle chaque phrase se décompose en un objet et un prédicat. Le prédicat modifie ou définit les propriétés de l'objet.
- **La logique descriptive:** est un sous ensemble de la logique du premier ordre qui est utilisé pour décrire les concepts d'un domaine donné.
- **La logique floue:** Une méthode de modélisation des imperfections des données, c'est une extension de la logique booléenne qui se rapproche du raisonnement humain, plus de détails sont présentés dans le chapitre 3.

1.11 Conclusion

A travers ce chapitre, nous avons présenté quelques définitions sur le concept du Cloud Computing qui fait référence à la dématérialisation des ressources informatiques. Il permet de réduire les dépenses sur les infrastructures technologiques et de maintenir un

¹Une méthode algorithmique pour résoudre des problèmes d'optimisation. Le concept a été introduit au début des années 1950 par Richard Bellman.

²Une méthode dans laquelle on est amené à maximiser(ou minimiser) une application linéaire, appelée fonction d'objectif ou fonction économique, sur un ensemble d'équations et/ou d'inéquations linéaires, dites contraintes.

³Un algorithme qui suit le principe de faire, étape par étape, un choix optimum local.

accès facile aux renseignements de l'utilisateur avec des dépenses minimales et contrôlées. Par ces connaissances de base sur le Cloud Computing, nous donnons des détails approfondis sur un type du Cloud Computing qui est le Cloud véhiculaire dans le chapitre suivant.

Chapitre 2

Cloud Computing Véhiculaire

2.1 Introduction

Les réseaux **VANets** ne sont qu'une application des réseaux Ad hoc mobiles. Ils constituent le noyau d'un Système de Transport Intelligent(**STI**) ayant comme objectif principal l'amélioration de la sécurité routière et permettront aussi d'offrir de nouveaux services aux usagers de la route. En effet, grâce aux ressources et aux **OBUs** (**O**n **B**ord **U**nits) embarqués dans les véhicules, une nouvelle technologie est apparue. Cette technologie exploite l'idée du Cloud Computing dans les **VANet** qui est nommé Cloud Computing véhiculaire ou Cloud véhiculaire.

Ce chapitre est divisé en deux parties, dans la première partie nous exposons les réseaux véhiculaires, leurs caractéristiques et leurs applications. Puis, dans la deuxième partie, nous détaillons les concepts liés au Cloud Computing Véhiculaire, les protocoles proposés dans ce contexte, les défis du Cloud véhiculaires et la sélection de services dans le Cloud véhiculaire qui représente l'objectif de notre travail.

2.2 Réseaux véhiculaire (VANet)

La grande évolution des systèmes de communications sans fils a changé la vie humaine en offrant la facilité et la souplesse dans l'utilisation des services Internet et divers applications. Cette évolution a fait naissance des réseaux Ad hoc véhiculaire (**VANets**, pour **V**ehicular **A**d hoc **N**etworks), dans lesquels les véhicules, équipés de dispositifs de communication sans fil, système de positionnement et des cartes numériques, agissent comme des machines intelligentes qui communiquent pour des fins de sécurité et de confort.

2.2.1 Définition d'un réseau véhiculaire

Les réseaux **VANets** (Vehicular Ad hoc NETworks) (voir Figure 2.1) constituent une nouvelle forme de réseaux Ad hoc mobiles, dans laquelle les véhicules peuvent communiquer entre eux (**V2V**) ainsi qu'avec l'infrastructure (**V2I**) installée aux bords de routes (**RSU**, pour **Road Side Unit**). Par rapport à un réseau Ad hoc classique, les réseaux **VANets** sont caractérisés par une forte mobilité des nœuds rendant la topologie du réseau fortement dynamique [1] .

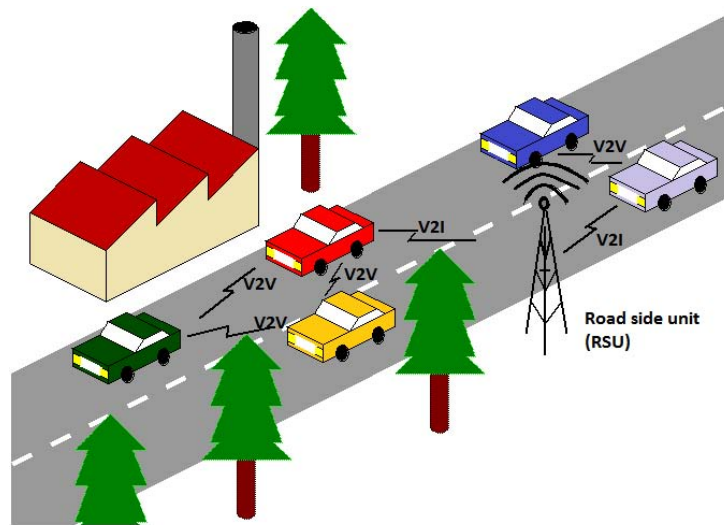


Figure 2.1: Un réseau VANet.

Actuellement, les nouveaux véhicules dits "*intelligents*" sont dotés de grandes capacités de stockage et de calcul, un équipement de communication sans fil, de dispositifs de perception de l'environnement (radar avant, radar arrière, caméras) et un système de localisation **GPS** (voir Figure 2.2). Ainsi, l'idée d'exploiter et d'offrir les ressources de véhicules a fait naissance du concept du Cloud Véhiculaire qu'on va le détailler dans les sections suivantes [17] .

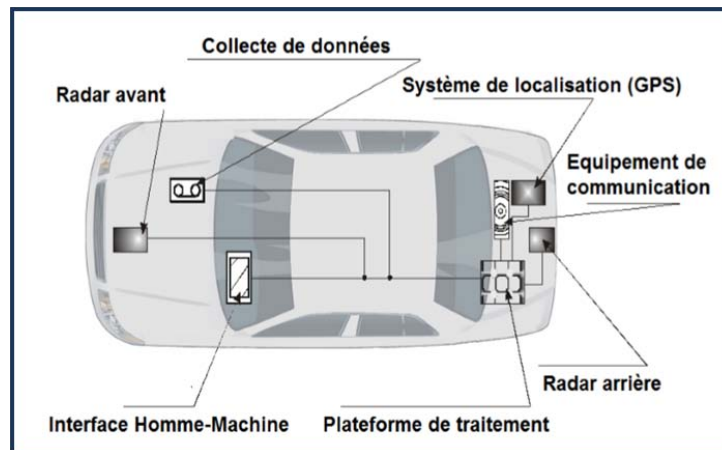


Figure 2.2: Éléments constituant le véhicule intelligent

2.2.2 Caractéristiques des réseaux VANet

Les réseaux véhiculaires ont des caractéristiques spécifiques qui les distinguent de réseaux Ad hoc mobiles. Parmi ces caractéristiques, on trouve [18]

2.2.2.1 Capacité de stockage et d'énergie

Contrairement au contexte des réseaux Ad hoc mobiles où la contrainte d'énergie représente un défi, les véhicules du réseau **VANet** disposent de suffisamment d'énergie, qui peut alimenter de façon continue ou presque tous ses différents équipements électroniques.

2.2.2.2 Topologie et la connectivité

Les réseaux **VANets** sont caractérisés par une connectivité sporadique, car un véhicule peut rejoindre ou quitter un groupe de véhicules en un temps très court, ce qui rend la topologie de ces réseaux très dynamique.

2.2.2.3 Modèle de mobilité

La mobilité des véhicules dans **VANet** est limitée par l'infrastructure routière (Par exemple, route, autoroute, les panneaux de signalisation, limites de vitesses, . . .). De plus, cette mobilité est liée directement au comportement des conducteurs et leurs réactions face à des obstacles ou des situations différentes et complexes rencontrées (accident, les heures d'embouteillage, . . . etc.).

2.2.2.4 Sécurité et l'anonymat

L'importance des informations échangées via les communications véhiculaires rend l'opération de sécurisation de ces réseaux cruciale, ce qui constitue un pré-requis au déploiement des VANets.

2.2.3 Applications des réseaux véhiculaire

Les principales applications des réseaux véhiculaires peuvent être classées en trois catégories [1]:

2.2.3.1 Applications pour la sécurité routière

Elles concernent les applications ayant un impact direct sur la sécurité des usagers de la route et des biens. Ces applications comprennent la diffusion des messages d'alerte sur l'état du trafic, des routes, de la météo, les accidents et les travaux sur la route, les limites de vitesse, la distance de sécurité, . . . etc.

2.2.3.2 Applications pour les systèmes d'aide à la conduite

Pour faciliter la conduite autonome et apporter un support au conducteur dans des situations particulières telles que: l'aide aux dépassements de véhicules, prévention des sorties de voies en ligne ou en virage, . . . etc.

2.2.3.3 Application de confort du conducteur et des passagers

Les réseaux VANet ne se contenteront pas seulement à offrir des services liés à la sécurité des usagers de la route, mais permettront aussi d'assurer leurs confort en offrant des services de communications et d'informations tels que, l'accès mobile à Internet, la messagerie, le chat inter-véhicules, les jeux en réseaux, . . . etc.

2.3 Cloud Computing Véhiculaire

Récemment, l'idée d'exploiter et d'offrir les ressources embarquées dans les véhicules a fait naissance au concept de Cloud véhiculaire. En effet, ce concept constitue une sorte

de fusion entre les deux domaines *Cloud Computing* (voir chapitre 1) et les **VANets** (voir Section 2.2.1).

2.3.1 Définition du Cloud Computing Véhiculaire

C'est un paradigme qui combine le concept du *Cloud Computing* et les réseaux véhiculaire, ce concept permet à chaque véhicule, qui se dispose de ressources suffisantes (stockage, calcul, . . .), d'offrir et de louer ces ressources en tant que service à des différents utilisateurs, et donc d'agir comme étant un serveur du Cloud mobile [13].

2.3.2 Services du Cloud Computing Véhiculaire

Généralement, Il existe trois types de services qui peuvent être offerts dans le Cloud Computing Véhiculaire. Nous décrivons ici ces services tout en soulignant qu'il peut y avoir d'autres services qui pourraient être ajoutés ultérieurement en fonction des besoins des consommateurs:

2.3.2.1 Network as a service ou NaaS (Accès à Internet)

Certains véhicules intelligents ont une connexion permanente à Internet (par exemple, via un réseau cellulaire), mais d'autres ne l'ont pas. Cela ouvre la porte pour les véhicules intelligents, possédant un accès à l'Internet, pour offrir gratuitement ou non leur bande passante à d'autres véhicules avec [19].

2.3.2.2 Storage as a service ou STaaS (Virtual network hard-disk)

Les véhicules qui ont une grande capacité de stockage peuvent fournir ce service à d'autres véhicules, qui peuvent avoir besoin de stockage supplémentaire pour faire fonctionner leurs applications [20].

2.3.2.3 Information as a Service (INaaS) et ENTertainment as a Service (ENaaS)

Les conducteurs peuvent avoir besoin d'informations spécifiques tels que, des informations de trafic, les avertissements de la congestion et les accidents, la météo ou les conditions routières . . . etc. Au lieu d'accéder à Internet pour obtenir ces informations,

ils peuvent les demander directement depuis les autres véhicules sur la route, qui ont déjà collecté ces informations [21].

2.3.3 Architecture du Cloud Computing Véhiculaire

Dans [22], Les auteurs ont proposé la classification suivante des architectures du Cloud Computing Véhiculaire:

2.3.3.1 Cloud Véhiculaire (Vehicular Cloud, VC)

La figure 2.3 montre l'architecture de **VC** dans laquelle un véhicule offre ces ressources en tant que service (INaaS, STaaS, NaaS . . . etc.) à d'autres véhicules consommateur.

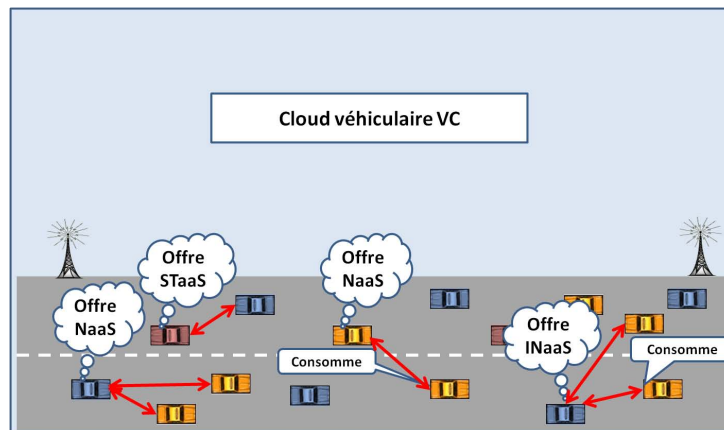


Figure 2.3: Cloud véhiculaire

2.3.3.2 Véhicules utilisant le Cloud Computing (Vehicles using Cloud, VuC)

La figure 2.4 représente l'architecture de **VuC** lorsque les véhicules utilisent les services de Cloud classique. Dans cette architecture, les **RSUs** agissent comme des passerelles pour les véhicules aux services de *Cloud Computing* traditionnel.

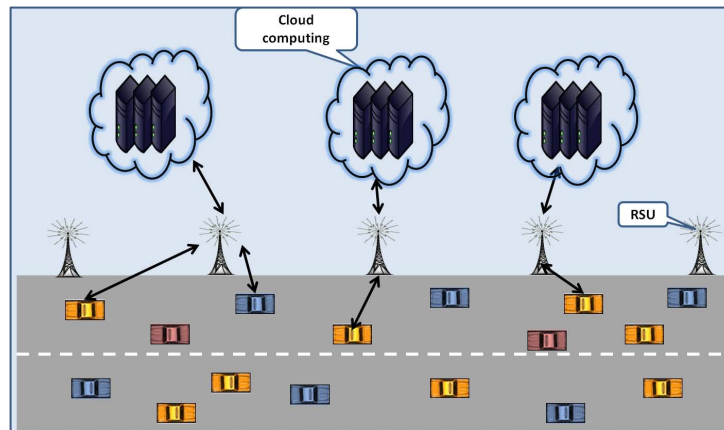


Figure 2.4: Véhicule utilisant le Cloud

2.3.3.3 Cloud hybride (Hybrid Cloud, HC)

La figure 2.5 illustre l'architecture de **HC** qui est la combinaison entre **VC** et **VuC**. Dans cette architecture, les véhicules, qui se déplacent sur la route, peuvent consommer des services soit depuis d'autres véhicules ou bien du *Cloud Computing* classique à travers les **RSUs**.

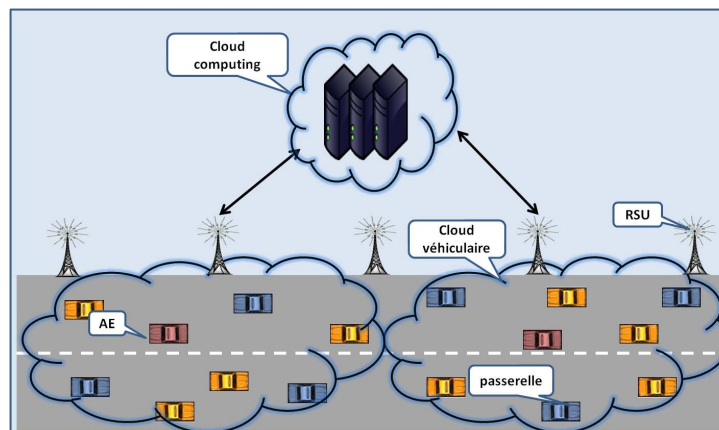


Figure 2.5: Cloud hybride

2.3.4 Protocoles du Cloud véhiculaire

Dans cette section, nous allons présenter quelques protocoles proposés pour l'offre et la consommation de services dans le Cloud véhiculaire.

2.3.4.1 Protocole CROWN (disCoveRing and cOnsuming services WithiN vehicular Cloud)

Dans [3], les auteurs ont proposé un protocole d'offre et de consommation des services dans le Cloud véhiculaire appelé **CROWN**. Ce protocole est basé sur l'utilisation des points d'accès **RSUs** (Road Side Unit) comme un répertoire de **Cloud** dans lequel chaque véhicule fournisseur nommé **STAR** peut sauvegarder ses offres. Ces **RSUs** permettent aux véhicules consommateurs de découvrir les offres de **STAR** et les consommer (voir figure 2.6). Le déroulement de ce protocole se fait en trois grandes phases : l'enregistrement des services, la découverte et la consommation des services.

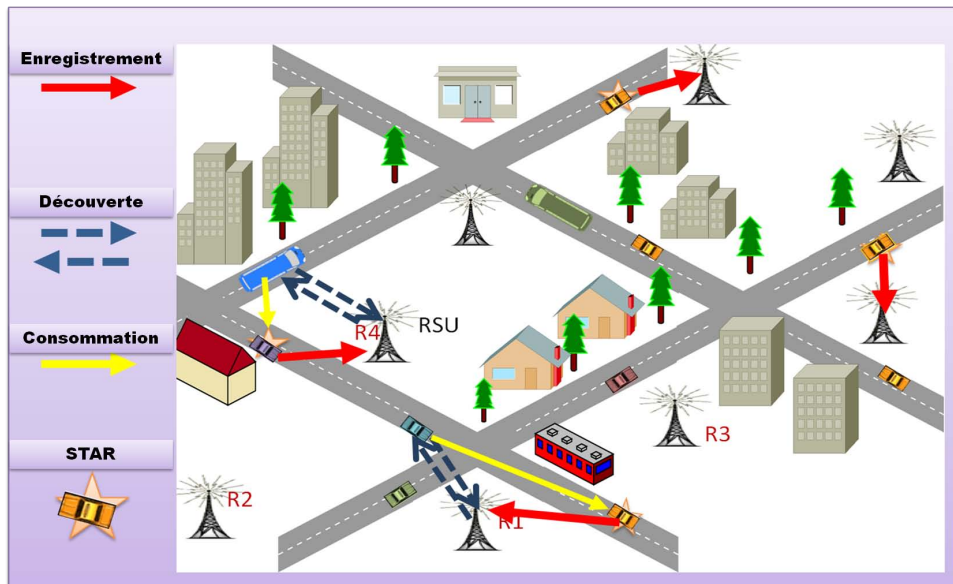


Figure 2.6: Vue générale sur CROWN

Tableau 2.1: Attributs des services offerts par le STAR

| Service | Attributs | | | |
|----------------|--|---------------------------------------|------------------------------|--------------------------|
| NaaS | Bande passante | Délais | Coût par heure | Type de service Internet |
| STaaS | Capacité max | Temps de stockage max par utilisateur | Coût par unité de stockage | Type de données stockées |
| INaaS et ENaaS | Capacité d'information max par utilisateur | | Coût par unité d'information | Type d'information |

A Enregistrement des services

Chaque véhicule qui décide de participer dans le Cloud Computing Véhiculaire comme **STAR**, formule un paquet d'enregistrement qui contient son identifiant, les services offerts et leurs attributs (voir tableau 2.1). Ensuite, il envoie ce paquet vers le plus proche **RSU** (étape 1, figure 2.7).

Quand le **RSU** reçoit le paquet du **STAR**, il sauvegarde ses offres, puis il définit une région où le **STAR** peut offrir efficacement ses services au consommateur. Ensuite, il partage avec les **RSUs** qui existent dans cette région les informations qu'il possède sur le **STAR** (étape 2, figure 2.7).

Le **STAR** envoie des messages périodiques au plus proche **RSU** pour la mise à jour de sa localisation (étape 3, figure 2.7).

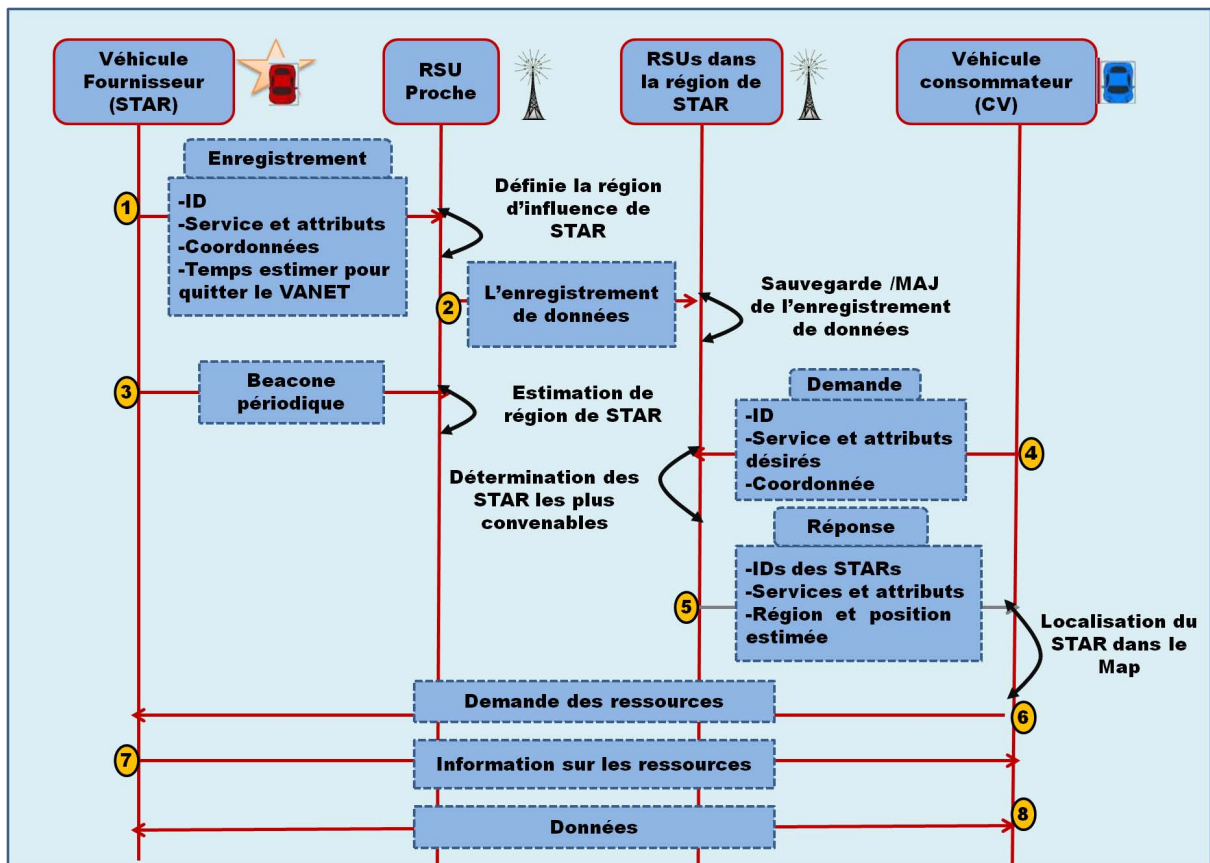


Figure 2.7: Déroulement de protocole CROWN

B Demande de service

Si un véhicule consommateur a besoin de ressources supplémentaires, il envoie une requête vers le plus proche **RSU** (étape 4, figure 2.7). Cette requête contient les ressources

demandées, les attributs de ces ressources et les coordonnées géographiques de ce véhicule.

Après la réception d'une requête, le RSU cherche dans son cache les STARs qui peuvent satisfaire les besoins de consommateur. Ensuite, il choisit les véhicules STARs les plus adéquats et répond par un paquet qui contient (étape 5, figure 2.7) :

- ID du STAR,
- La dernière localisation de STAR,
- Les ressources offertes par le STAR et leurs attributs

C Consommation des services

Le consommateur demande les ressources désirées en envoyant une requête directement vers le véhicule STAR qui contient des informations d'identification (étape 6, figure 2.7).

Si le véhicule STAR ne peut pas satisfaire la requête du consommateur, il refuse la demande du consommateur en envoyant un acquittement négatif. Dans le cas contraire, il répond par l'envoi des informations nécessaires avant de débiter la consommation de service, telles que la sécurité des données échangées et le mode de paiement (étape 7, figure 2.7).

Enfin, le STAR et le consommateur peuvent échanger les paquets de données qui correspondent aux ressources que l'utilisateur veut les consommer (étape 8, figure 2.7).

2.3.4.2 Protocole RCS-VC (Renting out and Consuming Services in Vehicular Clouds based on LTE-A)

Dans [4], les auteurs proposent un autre protocole d'offre et de consommation des services dans le Cloud Computing Véhiculaire, appelé **RCS-VC**. Ce protocole exploite les points d'accès, eNodeBs, du réseau cellulaire LTE[23][24], comme un répertoire de Cloud en ajoutant un nouveau serveur nommé **VSD** (Vehicular Services Directory), dans lequel chaque véhicule fournisseur stocke ses services offerts, et à partir duquel, un véhicule consommateur peut découvrir les services offerts. La figure 2.8 donne une vue générale sur ce protocole.

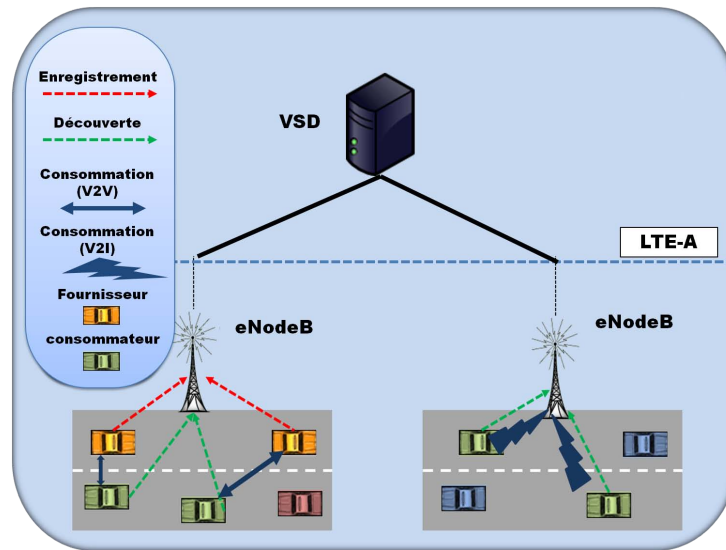


Figure 2.8: Vue générale sur RCS-VC

A Enregistrement de service

Après avoir défini les services à offrir et leurs attributs, un véhicule fournisseur envoie un paquet **SRP** (Service Registration Packet) au serveur **VSD** à travers un **eNodeB**, (étape 1 et 2, figure 2.9). Ce paquet contient l'identifiant du véhicule fournisseur, les services offerts et les attributs de ses services (voir Tableau 2.2). Une fois que le **VSD** a reçu le paquet, il vérifie l'identité du véhicule fournisseur. Ensuite, il stocke les services offerts dans son cache (étape 3, figure 2.9).

On note que le véhicule fournisseur envoie des paquets de mise à jour, nommé **SRUP** (Services Registration Update Packet), à chaque fois que les attributs des services sont modifiés (étape 4, figure 2.9). De plus, le **VSD** peut aussi mettre à jour les services selon leurs disponibilités.

Tableau 2.2: Attributs spécifiés par un véhicule fournisseur dans RCS-VC.

| Service | Attributs spécifiques | | Attributs communs | | |
|----------------|--|---------------------------------------|-------------------------|----------------|--------------------------|
| | | | Coût | Disponibilité | Type |
| NaaS | Bande passante | Délais | Par heure | Par heure | Type de service Internet |
| STaaS | Capacité max | Temps de stockage max par utilisateur | Par unité de stockage | /jour /semaine | Type de données stockées |
| INaaS et ENaaS | Capacité d'information max par utilisateur | | Par unité d'information | | Type d'information |

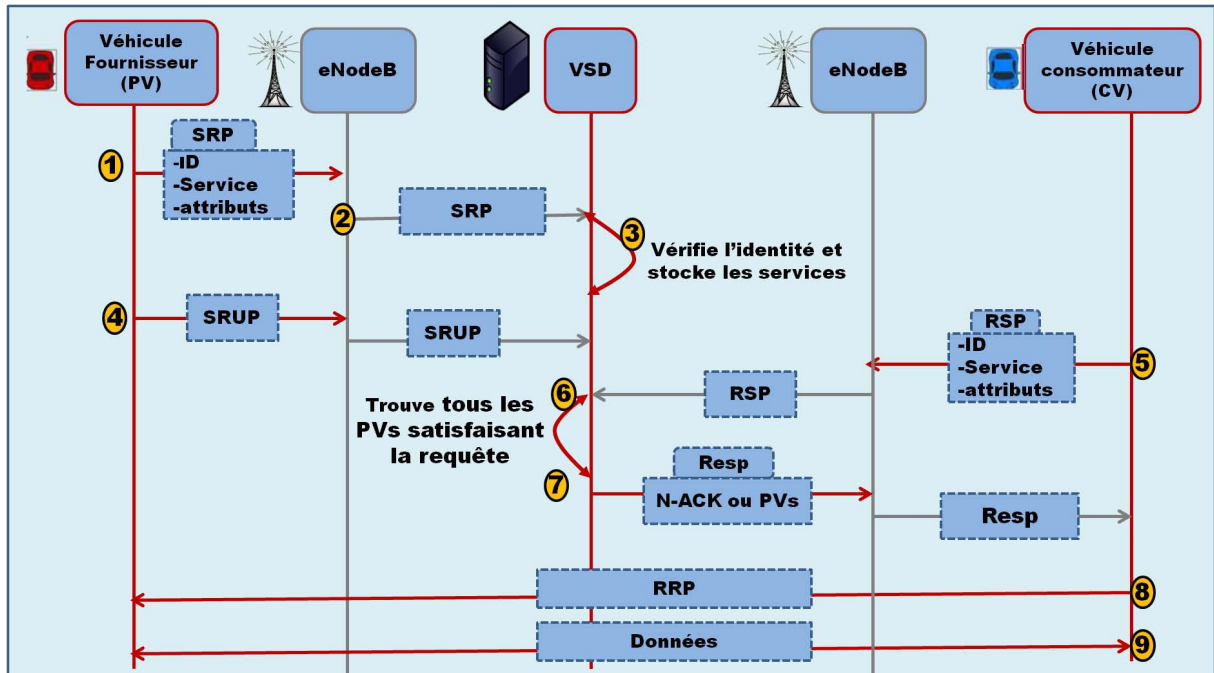


Figure 2.9: Déroulement de protocole RCS-VC

B Demande de service

Après que le véhicule consommateur spécifie les services requis et leurs attributs (voir Tableau 2.3), il envoie une requête **RSP** (Request Service Packet) au **VSD** via **eNodeB** (étape 5, figure 2.9). Ensuite, le **VSD** sélectionne les véhicules satisfaisant la requête, et il répond soit (étape 6 et 7, figure 2.9):

- Par un paquet **RESp** (**RES**ponse packet) contient les véhicules fournisseurs, leurs offres et la zone de chaque fournisseur.
- Par un acquittement négatif dans le cas où il ne trouve pas un véhicule fournisseur.

Tableau 2.3: Attributs requis par un véhicule consommateur dans RCS-VC.

| Service | Attributs spécifiques | | Attributs communs | | |
|-------------------|--|---------------------------------------|-------------------------|-------------------|--------------------------|
| | | | Coût | Disponibilité | Type |
| NaaS | Bande passante | Délais | Par heure | Par heure | Type de service Internet |
| STaaS | Capacité max | Temps de stockage max par utilisateur | Par unité de stockage | /jour /semaine | Type de données stockées |
| INaaS et ENaaS | Capacité d'information max par utilisateur | Freshness d'information | Par unité d'information | | Type d'information |

C Consommation de service

Quand le véhicule consommateur reçoit le paquet RESp, il choisit le véhicule fournisseur le plus adéquat selon ses besoins. Ensuite, il envoie le paquet **RRP** (**R**equest for **R**essources **P**acket) (étape 8, figure 2.9) directement vers le véhicule fournisseur pour demander les services(étape 9, figure 2.9).

On note que la communication entre le fournisseur et le consommateur se fait en mode V2V s'ils sont dans la même cellule ou à travers le LTE-A dans le cas contraire.

2.3.4.3 Protocole DCCS-VC

DCCS-VC(Discovering and Consuming Cloud Services in Vehicular Clouds) est un protocole d'offre et de consommation des services dans le Cloud Computing Véhiculaire [5]. DCCS-VC exploite la présence des bus publics dans l'environnement urbain et les utilise comme un répertoire du Cloud, dans lequel les véhicules fournisseurs stockent leurs offres et à partir duquel les véhicules consommateurs découvrent les services offerts (voir figure 2.10).

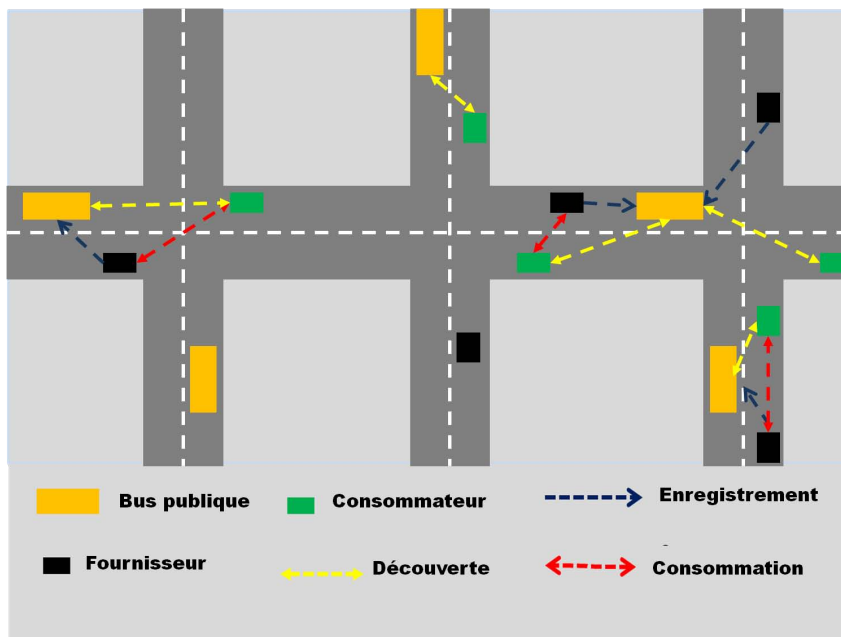


Figure 2.10: Vue générale sur DCCS-VC

A Enregistrement des services de PV

Une fois que le véhicule fournisseur définit ses offres et la qualité de service requis

de chaque offre, il envoie un paquet de registration **SR** (Service Registration) au plus proche bus public (étape 1, figure 2.11). Ce paquet comprend essentiellement :

- L'identifiant du véhicule fournisseur.
- Les services offerts.
- Les attributs de chaque service (voir Tableau 2.4).

Tableau 2.4: Attributs spécifiés par un véhicule fournisseur dans DCCS-VC.

| Service | Attributs spécifiques | | Attributs communs | | |
|----------------|--|---------------------------------------|-------------------------|-------------------|--------------------------|
| | | | Coût | Disponibilité | Type |
| NaaS | Bande passante | Délais | Par heure | Par heure | Type de service Internet |
| STaaS | Capacité max | Temps de stockage max par utilisateur | Par unité de stockage | /jour /semaine | Type de données stockées |
| INaaS et ENaaS | Capacité d'information max par utilisateur | | Par unité d'information | | Type d'information |

Puis, le bus envoie un paquet d'acceptation d'enregistrement des services au véhicule fournisseur, il envoie aussi un paquet **TBP**(**T**racking **B**us **P**ath) afin d'informer le véhicule fournisseur par le chemin qu'il couvre. Si le véhicule fournisseur peut se localiser en dehors de ce chemin (étape 4, figure 2.11) doit réenregistrer ses services auprès du plus proche bus. Enfin, le bus informe les autres bus dans son groupe de l'enregistrement du véhicule fournisseur (étape 3, figure 2.11). De plus, ces bus mettront à jour les services offerts selon la disponibilité de chaque service.

B Demande des services du véhicule fournisseur

Quand un véhicule consommateur cherche à consommer des services supplémentaires, il peut demander ses services depuis le plus proche bus par l'envoi d'un paquet **RS** (**R**equest for **S**ervice) (étape 7, figure 2.11). Le paquet **RS** contient les services désirés et leurs attributs requis (Voir tableau 2.5).

Tableau 2.5: Attributs requis par un véhicule consommateur(DCCS-VC).

| Service | Attributs spécifiques | | Attributs communs | | |
|----------------|--|---------------------------------------|-------------------------|-------------------|--------------------------|
| | | | Coût | Disponibilité | Type |
| NaaS | Bande passante | Délais | Par heure | Par heure | Type de service Internet |
| STaaS | Capacité max | Temps de stockage max par utilisateur | Par unité de stockage | /jour /semaine | Type de données stockées |
| INaaS et ENaaS | Capacité d'information max par utilisateur | Freshness d'information | Par unité d'information | | Type d'information |

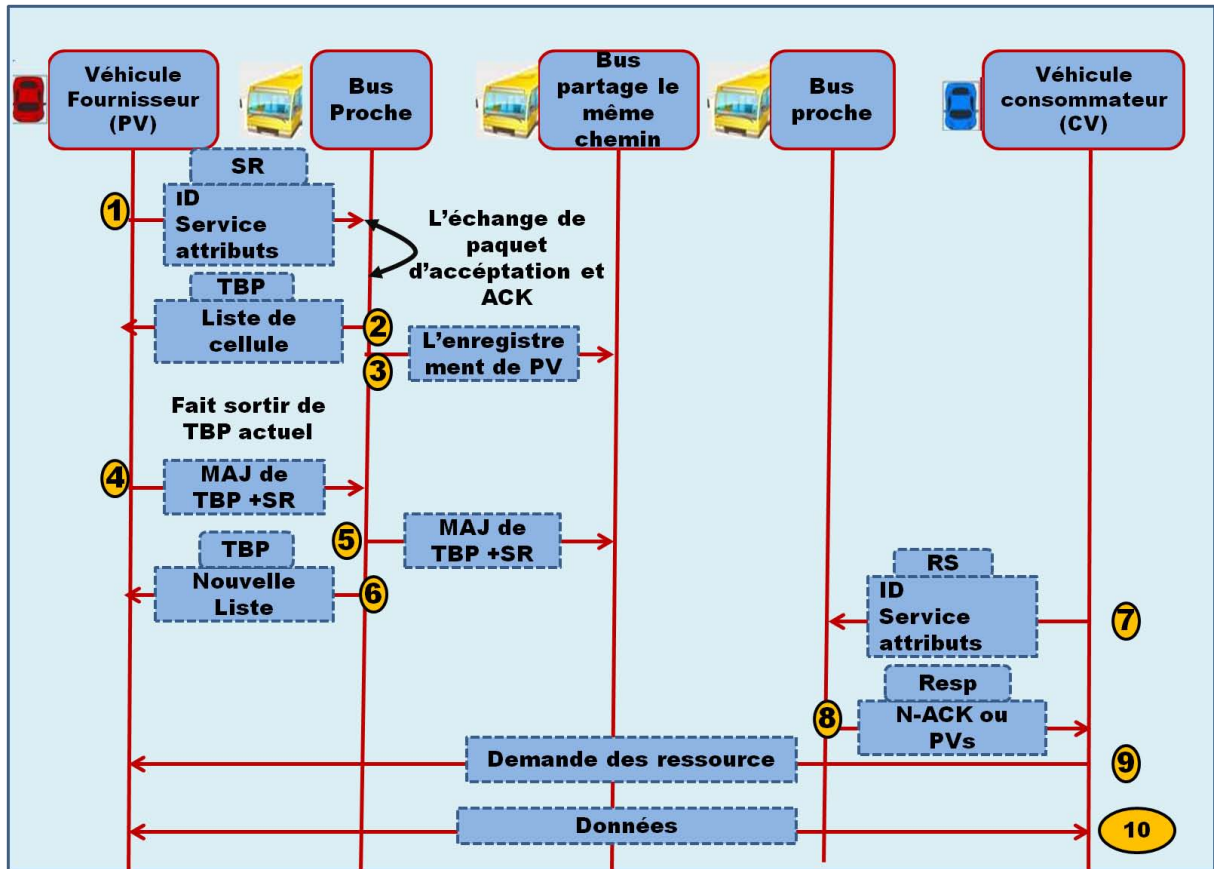


Figure 2.11: Déroulement de protocole DCCS-VC

Si le bus reçoit le paquet **RS**, il cherche tous les véhicules fournisseurs satisfaisant la requête. Dans le cas où il ne trouve pas des véhicules fournisseurs, il répond par un acquittement négatif via le paquet **RESp** (RESponse packet), dans le cas contraire, il envoie la liste de tous les véhicules fournisseurs ainsi que leurs services offerts (étape 8, figure 2.11).

C Consommation des services

A la réception d'une réponse positive, le véhicule consommateur choisit le véhicule fournisseur le plus adéquat à ses besoins et envoie une requête **RR** (Request for Ressource) au véhicule fournisseur choisi, qui comprend toutes les informations nécessaires avant le début de la consommation (étape 9 et 10, figure 2.11).

2.3.4.4 Protocole FcVcA

Le protocole **FcVcA** (A **F**uzzy **C**lustering-based **V**ehicular **C**loud **A**rchitecture) [25] forme un Cloud véhiculaire en organisant les véhicules du réseau en clusters (voir figure 2.12) selon la position du véhicule, le sens de déplacement et leurs vitesses. Dans chaque cluster, un véhicule est sélectionné comme étant un Cluster Head (CH).

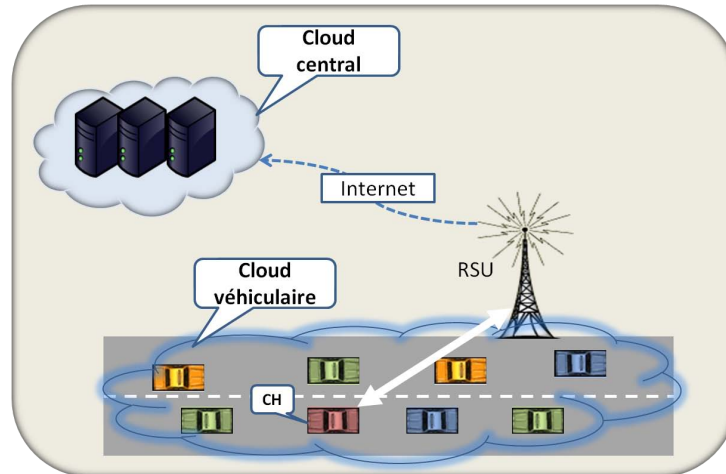


Figure 2.12: Cloud véhiculaire basé sur un cluster

Dans ce protocole, le **CH** (**C**luster **H**ead) est le contrôleur du Cloud qui a la responsabilité de la création, la maintenance, et la suppression du Cloud véhiculaire. Tous les autres véhicules virtualise ses ressources et les sauvegarde dans le CH. Ensuite, si un véhicule a besoin des ressources, il peut demander ses besoins depuis le **CH**.

2.3.4.5 Etude comparative

Les protocoles des Cloud véhiculaire sont classifiés selon plusieurs critères. On trouve des protocoles qui sont basés sur la communication véhicule à véhicule et d'autre sur la communication véhicule à infrastructure, des protocoles qui sont appliqués dans un environnement urbain et freeway, alors que d'autres sont appliqué seulement dans un enivrement urbain.

Le tableau 2.6 présente une étude comparative entre les protocoles du Cloud Computing Véhiculaire selon les critères de classification cités au dessus :

Tableau 2.6: Comparaison entre les protocoles de Cloud véhiculaire

| | CROWN | RCS-VC | DCCS-VC | FeVvA |
|-------------------------|-------------------------|-----------------------|-----------------------|-------------------------------|
| Domaine d'application | Cloud véhiculaire | Cloud véhiculaire | Cloud véhiculaire | Cloud véhiculaire |
| Répertoire de Cloud | RSU | LTE-A | Bus public | Véhicule CH (Cluster Head) |
| Type de communication | V2I | V2I/V2V | V2V | V2V |
| Application | STaaS, NaaS et INaaS | NaaS, StaaS, INaaS | NaaS, StaaS, INaaS | CaaS ¹ |
| Environnement | Urbain, Freeway | Urbain, Freeway | Urbain | Freeway |
| Architecture en cluster | Non | Non | Non | Oui |

2.3.5 Défis du Cloud Computing Véhiculaire

Dans cette section, nous présentons les principaux défis rencontrés dans le domaine du Cloud Computing Véhiculaire cités dans [21].

2.3.5.1 La vie privée et la sécurité de VC

La sécurité et la vie privée sont des aspects très importants pour l'établissement et le maintien de confiance de l'utilisateur de VC. Les mesures de la vie privée sont requises pour assurer les informations et la communication de VC dans un environnement fiable et isolé, tandis que les procédures de sécurité sont nécessaires pour la protection contre les menaces réseau.

Les principaux défis de la sécurité de VC comprennent :

- Vérifier l'authentification des utilisateurs et de l'intégrité des messages à cause de la forte mobilité des véhicules.
- Assurance de la confidentialité des messages sensibles en utilisant les algorithmes de cryptographie.
- Assurance de la sécurité des emplacements des véhicules, car la plupart des applications dans les systèmes véhiculaires reposent sur des informations de localisation.
- Fournir l'isolation des données pour assurer la sécurité des données stockées sur le Cloud.
- Accéder de façon sécurisée aux données stockées sur le Cloud.

¹CaaS (Computing as a service): c'est l'offre des capacités de calcul en tant que service.

2.3.5.2 La sélection de service dans le Cloud véhiculaire

La sélection de service dans le Cloud véhiculaire est un problème récent dans la recherche. Ce problème se base sur le choix d'un meilleur service offert dans le Cloud véhiculaire selon des critères de qualité, et dont le but est de satisfaire les besoins des clients.

Parmi les protocoles proposés dans le Cloud véhiculaire, seul le protocole CROWN (voir section 2.3.4.1) qui fait la sélection de service ayant le coût minimum.

A Les critères de qualité pour les services de Cloud véhiculaire

Généralement, on distingue cinq critères de qualité pour les services de Cloud Computing Véhiculaire :

- Débit ($q_{Thruput}$): $q_{Thruput}$ du véhicule fournisseur (**PV**) est défini comme étant le nombre des requêtes servie dans une période de temps donnée.
- Réputation (q_{Rep}): dépend de l'expérience d'utilisation de service du **PV** par le **CV**. q_{Rep} est calculée en fonction des opinions des **CVs** sur l'utilisation des services du **PV** c.à.d,

$$q_{Rep} = \frac{\sum_{i=1}^n R_i}{n}, R_i \in [0, 1]$$

Où R_i est l'opinion du **CV** sur le service de **PV** et n c'est le nombre des **CVs**.

- Taux d'exécution avec succès (q_{SER}): c'est la probabilité qu'un **PV** répond correctement à la demande du **CV** dans les échéances. q_{SER} est calculée en utilisant l'expression:

$$q_{SER} = \frac{N(s)}{K}$$

Où $N(s)$ est le nombre de fois qu'un service du **PV** a été consommé avec succès, et K est le nombre total d'invocations.

- Durée d'exécution (q_{ED}): définie comme étant le retard prévu en secondes entre le moment d'envoi de la demande et le moment de la réception des résultats. q_{ED} est calculée en utilisant l'expression:

$$q_{ED} = T_{process} + T_{Trans}$$

Où $T_{process}$ est temps de traitement, et T_{Trans} est le temps de transmission.

- Coût d'exécution (q_{EC}): Le montant pour une seule exécution de service de **PV**.

2.4 Conclusion

Dans ce chapitre, nous avons donné une vue générale sur les réseaux véhiculaire. Puis, nous avons présenté des points généraux sur le concept du Cloud Computing Véhiculaire, ses services, et son architecture. Ensuite, nous avons détaillé quelques protocoles d'offre et de consommation de services ainsi que les défis du Cloud Computing Véhiculaire. Enfin, nous avons présenté la sélection de service dans le Cloud Computing Véhiculaire qui représente l'objectif principale de notre travail.

Chapitre 3

Amélioration du protocole DCCS-VC

3.1 Introduction

Ce chapitre est scindé en deux parties. Dans la première partie, nous allons présenter les concepts de base de la logique floue. Tout d'abord nous allons donner une définition de la logique floue et son historique en suite, nous présentons un exemple introductif Afin de mieux appréhender la problématique envisagée, la définition d'une fonction d'appartenance, les formes d'une fonction d'appartenance, la conception des règles floues, nous allons parler aussi de la fuzzification et la défuzzification et en fin les applications de la logique floue. La majorité de cette partie a été prise d'un cours d'une université tunisienne.

La deuxième partie décrira notre travail réalisé, nous allons détailler le principe de l'idée proposée pour l'amélioration du protocole **DCCS-VC**, la construction de la base des règles et l'optimisation des règles. En suite nous décrivons notre environnement de simulation, les paramètres et le scénario de la simulation et en fin nous présentons les résultats de simulation de la technique proposée basée sur la logique floue et la technique **SAW**.

3.2 Partie 1 : Logique floue

La logique floue est une extension de la logique booléenne basée sur la théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue

permet d'accommoder le concept de vérité partielle: des valeurs entre complètement vrai et complètement faux sont admises. Son importance provient du fait que le raisonnement humain est approximatif .

3.2.1 Historique de la logique floue

- Le concept de la logique floue a été introduit en 1965 lorsque le professeur **Lotfi A. Zadeh**, de l'université de Berkeley aux **USA**, publie un article intitulé « *Ensembles flous* » (Fuzzy sets). Il a réalisé depuis de nombreuses avancées théoriques majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques [26].
- À partir des années 70, certains chercheurs se sont penchés parallèlement sur la résolution par logique floue de problèmes réputés difficiles, les premières applications sont apparues dans les domaines : systèmes expert, Aide à la décision en médecine, commerce... [27].
- En 1974, Le professeur **M. Mamdani** expérimentait la théorie énoncée par **Zadeh** sur une chaudière à vapeur, matériel dont on connaît la complexité, introduisant ainsi la commande floue dans la régulation d'un processus industriel. Plusieurs applications ont alors vu le jour en Europe, pour des systèmes parfois très complexes, telle la régulation de fours de cimenterie réalisée par la société **F. L. Smidt-Fuller** en 1978 [26].
- En 1985, c'est au Japon, où la recherche n'est pas seulement théorique mais également très applicative, que la logique floue connaît son véritable essor. Les produits grand public, machines à laver, appareils photographiques et autres caméscopes est ampillés « *fuzzy logic* » ne se comptent plus. Dans l'industrie, le traitement des eaux, les grues portuaires, les métros, les systèmes de ventilation et de climatisation sont touchés. Enfin, des applications existent dans des domaines très différents tels que la finance ou le diagnostic médical[28].
- A partir de 1990, et en Allemagne que des applications apparaissent en grand nombre ainsi qu'à une moindre échelle aux **USA**. Enfin en France, la logique floue devient aujourd'hui une réalité [26].

3.2.2 Concept de la logique floue

La logique floue permet de considérer des classes d'objets dont les frontières ne sont pas clairement déterminées, par l'introduction d'une fonction caractéristique (fonc-

tion d'appartenance) prenant des valeurs courantes entre 0 et 1., contrairement aux ensembles « booléens », dont la fonction caractéristique ne prend que deux valeurs possibles 0 et 1.

Ces ensembles flous ont le grand avantage de constituer une représentation mathématique de labels linguistiques largement utilisés dans l'expression de connaissances expertes, qualitatives et manipulées dans le raisonnement approché qui sera fait à partir de cette connaissance.

Ils apparaissent donc comme un moyen de réaliser l'interface entre l'information numérique (quantitative) et l'information symbolique (linguistique, qualitative) [29].

3.2.3 Fonctions d'appartenance

L'attribution d'un coefficient de confiance à l'affirmation « x appartient à un ensemble A » se présente facilement par une fonction dite d'appartenance $\mu_A(x)$ à valeurs dans $[0, 1]$, nous prenons comme exemple une personne désirant compléter le niveau d'un réservoir contenant de l'eau à une température donnée de façon à remplir ce réservoir d'eau à une température souhaitée T à l'aide d'un mitigeur. Dans le cas le plus simple, on peut distinguer trois valeurs « froid », « tiède », et « chaud » de la variable linguistique « température T », la fonction d'appartenance représentant la variable linguistique « température » peut être représenté par la figure 3.1 :

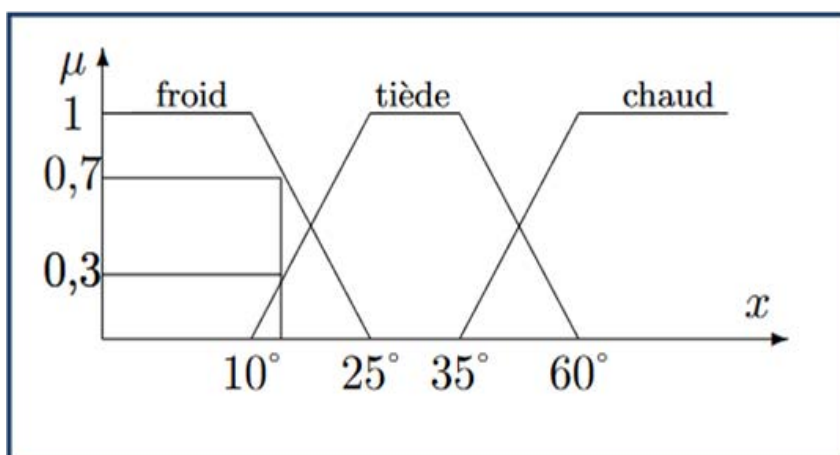


Figure 3.1: Fonction d'appartenance avec trois ensembles pour la variable « température ».

Ainsi, une température de 13° aura un facteur d'appartenance $\mu = 0.3$ à l'ensemble « tiède » et $\mu = 0.7$ à l'ensemble « froid ».

3.2.3.1 Différentes formes d'une fonction d'appartenance

Les fonctions d'appartenance peuvent prendre plusieurs formes, souvent, on utilise pour les formes les plus simples qui sont les formes trapézoïdales ou triangulaires composées par morceaux de droites.

. Le tableaux (3.1) donne quelques fonctions d'appartenances :

Tableau 3.1: Différentes formes d'une fonction d'appartenance

| Nom | Forme | Fonction |
|----------|-------|---|
| Triangle | | $\mu(x) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(m-a)} & x \in [a, m[\\ \frac{(b-x)}{(b-m)} & x \in [m, b] \\ 0 & x > b \end{cases}$ |
| Trapèze | | $\mu(x) = \begin{cases} 0 & x < a \\ \frac{(x-a)}{(m-a)} & x \in [a, m[\\ 1 & x \in [m, n[\\ \frac{(b-x)}{(b-n)} & x \in [n, b] \\ 0 & x > b \end{cases}$ |
| Cloche | | $\mu(x) = \frac{1}{1 + \frac{xv-x_0}{a}} \quad -\infty < x < \infty$ |

3.2.4 Opérateurs flous

Les variables linguistiques sont reliés entre elles par des opérateurs qui interviennent sur les fonctions d'appartenance représentant les variables linguistiques, comme dans l'exemple « l'air est froid et le vent est fort » ou dans « si l'air est froid ou si le vent est fort » il faut fermer la porte.

Il apparaît deux types de composition ET et OU auxquels il faut ajouter la négation. Notons x et y les variables linguistiques caractérisant la température de l'air et la force du vent et $\mu A(x)$, $\mu B(y)$, $\mu E(z)$, $\mu O z$, $\mu C z$ avec $z = x, y$, les fonctions d'appartenance associées aux propriétés respectives « l'air est froid », « le vent est fort », « l'air est froid et le vent est fort », « l'air est froid ou le vent est fort », « l'air n'est pas froid ».

3.2.4.1 Opérateur NON

La propriété « l'air n'est pas froid » peut être caractérisée de façon évidente par la fonction d'appartenance :

$$\mu C(z) = 1 - \mu A(x) \quad (3.1)$$

L'opérateur NON est appelé aussi « complément », « négation » ou « inverse ».

3.2.4.2 Opérateur ET

L'opérateur ET appelé aussi « intersection », est un opérateur qui caractérise la satisfaction simultanée de deux propriétés, il est réalisé par la formation du minimum appliquée aux fonctions d'appartenance $\mu A(x)$ et $\mu B(y)$ des deux ensembles A et B .

$$\mu E(z) = \mu A \text{ et } B(z) = \min(\mu A(x), \mu B(y)) \quad (3.2)$$

3.2.4.3 Opérateur OU

Dans la logique floue, l'opérateur OU est réalisé par la formation du maximum appliquée aux fonctions d'appartenance $\mu A(x)$ et $\mu B(y)$ des deux ensembles A et B :

$$\mu O(z) = \mu A \text{ ou } B(z) = \max(\mu A(x), \mu B(y)) \quad (3.3)$$

3.2.5 Règles floues

En générale, plusieurs valeurs de variables linguistiques, convenablement définies par des fonctions d'appartenance, sont liées entre elles par des règles.

Une règle floue est une déclaration de la forme suivante :

- SI x est A ALORS y est B

Où x et y sont des variables linguistiques, et A et B sont des valeurs linguistiques, déterminées par les ensembles flous sur les ensembles x et y .

Un exemple de règle floue est :

- Si température élevée ALORS ventilation forte.

3.2.6 Fuzzification

La fuzzification est l'étape qui consiste en la quantification floue des valeurs réelles d'une variable. Les systèmes à logique floue traitent des variables d'entrées floues et fournissent des résultats sur des variables elles même floues.

La fuzzification est l'étape qui consiste en la définition des fonctions d'appartenance pour les variables du système (les variables d'entrée et les variables de sortie) et la réalisation du passage des grandeurs physiques en variables linguistiques qui peuvent être traitées par des inférences (expliquée dans la section suivante) [29].

3.2.7 Déffuzification

La déffuzification est le processus de convertir une valeur floue en valeur nette [30]. Il existe plusieurs méthodes de déffuzification mais celle de centre de gravité est presque la méthode qui est toujours utilisée :

La déffuzification par centre de gravité : elle consiste en la détermination du centre de gravité de la fonction d'appartenance résultante $\mu_{res}(x_R)$. Dans ce contexte, il suffit de calculer l'abscisse x_R^* qui peut être déterminée à l'aide de la relation générale (3.4) :

$$x_R^* = \frac{\int_{-1}^1 x_R \mu_{res}(x_R).dx}{\int_{-1}^1 \mu_{res}(x_R).dx} \quad (3.4)$$

Pour simplifier, on utilise la somme au lieu de l'intégrale :

$$x_R^* = \frac{\sum_{x=a}^b \mu_{res}(x).x_R}{\sum_{x=a}^b \mu_{res}(x)} \quad (3.5)$$

3.2.8 Inférence

Cette étape fait relier les variables d'entrées, qui sont transformées en variables linguistiques pendant l'étape de fuzzification; à la variable de sortie sous sa forme linguistique, par des règles mentales traduisant une action ou une décision linguistique sur la commande floue.

Il existe plusieurs méthodes d'inférence, on donne comme exemple :

* Inférence de Mamdani

L'inférence de *Mamdani* procède selon quatre étapes :

- 1) fuzzification des variables d'entrée.
- 2) Evaluation des règles.
- 3) Agrégation des sorties des règles.
- 4) Déffuzification.

3.2.9 Applications de la logique floue

3.2.9.1 Commande Floue

Parmi les nombreuses applications de la logique floue, une commande ou un contrôleur flou est un système qui se réduit en générale à des problèmes de régulation robuste. La partie de la logique floue qui sert de base théorique se limite à l'inférence et à la représentation sous forme de variables linguistiques, ce qui permet d'introduire facilement les connaissances des opérateurs dans le régulateur.

La commande floue s'avère être le champ d'application le plus actif à travers le monde, la première application de la logique floue dans le domaine de la commande a été réalisée par Mamdani en 1974 [31].

3.2.9.2 Classification et reconnaissance de forme

La reconnaissance de forme est un ensemble des méthodes qui ont pour but d'identifier des motifs informatiques afin de donner une décision dépendant de la catégorie attribuée à ce motif. À fin d'améliorer les décisions classiques, la logique floue a été introduite dans les méthodes de la reconnaissance de forme [32].

C'est une activité qui intervient dans des nombreux domaines d'applications tels que : la reconnaissance vocale, l'analyse d'images (médical, radar, télédétection), la reconnaissance des cibles (domaine militaire), horaires de train, base de données, recherche d'information, modélisation de systèmes...

3.2.9.3 Agrégation multicritères et optimisation

Les méthodes d'agrégation multicritères peuvent s'appliquer à des problèmes de choix, de rangement, de classification, de décision. Elles font appel à des méthodes de comparaison et peuvent conduire à des schémas de modélisation des préférences d'agents.

L'optimisation est une branche des mathématiques et de l'informatique qui vise à modéliser, à analyser et à résoudre les problèmes qui consistent à déterminer quelles sont les solutions satisfaisants un objectif quantitatif tout en respectant d'éventuelles

contraintes. Les méthodes de l'optimisation dans le Cloud classique incorporant la logique floue ont été utilisées dans la sélection des services Cloud [28].

3.3 Partie 2 : Notre approche : Sélection de services par la logique floue

Le protocole Discovering and Consuming Cloud Services in Vehicular Clouds (DCCS-VC) repose sur trois phases détaillées dans le chapitre 2, notre travail touche la 3^{ème} phase, qui est la consommation des services offerts dans le Cloud, comme nous avons déjà montré, la phase de consommation des services dans ce protocole consiste en l'envoi d'une liste de tout les **PVs** enregistrés dans le bus et qui offrent le service demandé par le consommateur, ce dernier choisit un **PV** et lui envoie une demande de ressource ...

Cette partie du chapitre présente notre travail qui vise une optimisation de la sélection des services dans le Cloud véhiculaire, c.-à-d. au lieu que le bus envoie une liste des **PVs**, il envoie le **PV** le plus adéquat selon des critères de sélection présentés dans la section (2.3.5.2) qui sont: le débit, la réputation, le taux d'exécution avec succès, la durée d'exécution et le coût d'exécution.

3.3.1 Construction de la base des règles

Pour pouvoir considérer un service comme « meilleur », nous devons évaluer ses critères en utilisant le concept de la logique floue exposé dans la section précédente, chacune de ces critères représente un ensemble flou qui est composé de plusieurs sous-ensembles flous définis comme suit :

- **Débit** : repose sur trois sous-ensembles flous : Faible - Moyen – Élevé.
- **Réputation** : constitué aussi de trois sous-ensembles flous: Mauvaise Moyenne- Excellente.
- **Taux d'exécution avec succès** : contient les sous-ensembles : Mauvais - Moyen- Excellent.
- **Durée d'exécution** : prend les sous ensembles : Long - Moyenne- Courte.
- **Coût d'exécution** : ses sous-ensembles sont : Pas cher – Médium – Cher.

Nous auront comme résultat le « service » qui prendra les sous-ensembles : Mauvais - Moyen - Excellent.

Veins, dans ce qui suit nous allons définir cet environnement de simulation :

3.3.2.1 Simulateur OMNET++

OMNET++ (Objective Modular Network Testbed in C++) est un environnement de simulation open source dont le développement a commencé en 1992 par **Andras Vargas** à l'université de Budapest. C'est un outil de simulation à événements discrets orientés objets, basé sur **C++**, il a été conçu pour simuler les systèmes réseaux de communication, les systèmes multi processeurs, et d'autres systèmes distribués.

OMNET++ est connu par la facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés [33].

3.3.2.2 Simulateur de mobilité SUMO

SUMO (Simulator Urban MObility) est un logiciel open source de simulation du trafic routier, conçu pour gérer de grands réseaux routiers, il a été mis en œuvre en 2001 avec une première version open source en 2002. **SUMO** peut simuler les mouvements des véhicules sans collision, intégrer différents types de véhicules, des rues à voies multiples avec la possibilité de changements de voie et la jonction à base des règles de passage [34].

3.3.2.3 Module Veins

Veins est un Framework open source qui permet la simulation des réseaux véhiculaires. Il est basé sur deux simulateurs: **OMNeT++**, un simulateur de réseau basé sur les événements et un simulateur de trafic routier **SUMO** (définis dans les sections précédentes). Il étend ces deux simulateurs pour offrir une gamme complète de modèles pour la simulation **IVC** (Inter-Vehicular-Communication) [35].

Le module **Veins**, qui fonctionne sur **OMNeT++** (voir Figure 3.2), est responsable de la synchronisation entre le simulateur réseau **OMNeT++** et le simulateur du trafic routier **SUMO**. À des intervalles réguliers :

Ce module envoie toutes les commandes mises en mémoire tampon à **SUMO** et déclenche une étape de simulation de trafic routier.

À la fin de l'étape de simulation, **SUMO** envoie une série de commandes et la position de tous les véhicules instanciés au module **OMNeT++**.

Après le traitement de toutes les commandes reçues et la mise à jour de tous les nœuds, **OMNeT++** avance la simulation à l'étape suivante, et ainsi de suite. Avec cette approche, **OMNeT++** est toujours au courant à chaque fois qu'un véhicule est

créé ou supprimé et sa position est mise à jour dans **SUMO**. **OMNET ++** peut également envoyer des commandes qui s'influent sur le comportement de mobilité de **SUMO**, comme l'arrêt d'un véhicule ou changement d'itinéraire [36].

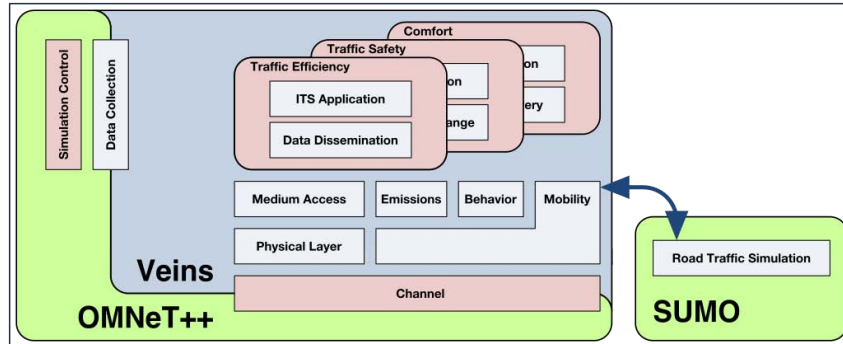


Figure 3.2: Veins reliant OMNET++ et SUMO.

3.3.3 Paramètres de simulation et étude comparative

Nous avons simulé le protocole **DCCS-VC** dans une zone urbaine ($9*9Km$). Pour valider notre technique de sélection, nous avons sélectionné un bus aléatoirement dans lequel on varie le nombre des offres des véhicules fournisseurs (**PVs**), et on applique notre technique de sélection à la réception d'une requête de la part d'un véhicule consommateur. Le tableau 3.4 représente les paramètres de notre simulation.

Afin d'évaluer notre sélection de service en utilisant la technique de la logique floue, nous avons établie une étude comparative entre la sélection de service à l'aide de la technique de la logique floue et la technique **SAW** (voir section 1.10.1.1). On note que la technique **SAW** se base sur des poids que le consommateur affecte à chaque critère du service qu'il veut obtenir.

Pour cela, on s'est concentré sur trois critères qui sont la réputation, le coût et la durée d'exécution. Dans chaque scénario, on favorise un critère par apport aux deux autres, c.à.d. on lui affecte le poids le plus fort par apport aux deux autres critères.

- La réputation est une évaluation (prend une valeur entre 0 et 10) effectuée par le consommateur concernant un service offert par un **PV** et qu'il l'avait consommé.
- Le coût est le montant pour une seule exécution de service de **PV** (unité monétaire)
- La durée d'exécution est le retard prévu entre le moment d'envoi de la demande et le moment de réception des résultats. (unité de temps)

Tableau 3.4: Paramètres de simulation.

| Paramètres | Valeurs |
|--|--------------------|
| Le temps de simulation | 100s |
| La surface de la simulation | 9 * 9 Km |
| Portée des nœuds | 500 m |
| Densité des véhicules | [50-200] véhicules |
| Nombre des bus | 40 |
| Densité des véhicules fournisseurs PVs | 5, 10, 15, 20 |
| La taille des paquets de l'enregistrement, TBP, demande et réponse | 128 octets |
| Densité des véhicules consommateurs | 1 |
| Nombres maximum de services offerts par PV | 3 services |
| Nombre maximum de services demandés par CV | 1 service |

3.3.3.1 Scénarios de simulation

Pour notre étude, nous avons simulé trois scénarios :

- **Scénario 1** : le véhicule consommateur favorise le coût par rapport aux deux autres critères (réputation et durée d'exécution). Autrement dit, le véhicule consommateur affecte un poids fort au coût et un poids faible à la réputation et la durée d'exécution.
- **Scénario 2** : Dans lequel on favorise la réputation par rapport aux deux autres critères (coût et durée d'exécution).
- **Scénario 3** : Le poids le plus fort dans ce scénario est associé à la durée.

Les résultats de simulations sont représentés dans le tableau et les figures au dessous.

3.3.3.2 Analyse des résultats

Dans la figure 3.3, les graphes (A, D, G) représentent l'écart entre la valeur de la réputation max offerte et la réputation du **PV** sélectionné, (B, E, H) représentent l'écart entre la valeur de la durée du **PV** sélectionné et la durée min offerte et (C, F, G) représentent l'écart entre la valeur de coût du **PV** sélectionné et le coût max offert, En variant à chaque fois la densité des **PVs**.

- **Scénario 1**: les graphes (A, B, C) représentant les résultats du scénario 1 montrent qu'en affectant le poids le plus fort au coût, et pour une densité de 5 et 10 **PVs**, la technique de la logique floue et la technique **SAW** possèdent

le même écart ce qui signifie que les deux techniques sélectionnent les mêmes **PVs** (voir tableau 3.5). Pour une densité de 15 et 20 **PVs**, l'écart de **SAW** est meilleur en terme de coût et l'écart de la logique floue est meilleure en terme de réputation et durée.

- **Scénario 2:** pour n'importe quelle densité de **PVs**, l'affectation du poids le plus fort à la réputation a produit le même écart de réputation, de coût et de durée pour les deux techniques **SAW** et logique floue, ces résultats sont présentés dans les graphes (D, E, F).
- **Scénario 3:** En affectant le poids le plus fort à la durée d'exécution (voir G, H, I), nous observons que pour une densité de 5 et 15 **PVs**, les techniques **SAW** et la logique floue sélectionnent les mêmes **PVs** (voir tableau 3.5). Cependant, pour une densité de 10 et 20 **PVs**, l'écart de **SAW** est meilleur en terme de durée et l'écart de la logique floue est meilleure en terme de réputation et de coût.

Ces résultats sont dû au concept de la logique floue qui est basé sur la sélection du **PV** ayant les meilleurs critères, par contre, la technique **SAW** favorise le **PV** selon le critère possédant le poids le plus fort.

Tableau 3.5: Résultats de simulation

| Scénario | Méthode | Densité de PV | L'ID de PV sélectionné | Réputation | Durée | Coût |
|-------------------|---------------|---------------|------------------------|------------|-------|------|
| Scénario 1 | Logique floue | 5 | 6 | 10 | 23 | 14 |
| | SAW | | 6 | 10 | 23 | 14 |
| | Logique floue | 10 | 10 | 7 | 27 | 10 |
| | SAW | | 10 | 7 | 27 | 10 |
| | Logique floue | 15 | 16 | 10 | 24 | 12 |
| | SAW | | 10 | 7 | 27 | 10 |
| Logique floue | 20 | 16 | 10 | 24 | 12 | |
| SAW | | 10 | 7 | 27 | 10 | |
| Scénario 2 | Logique floue | 5 | 6 | 10 | 23 | 14 |
| | SAW | | 6 | 10 | 23 | 14 |
| | Logique floue | 10 | 10 | 7 | 27 | 10 |
| | SAW | | 10 | 7 | 27 | 10 |
| | Logique floue | 15 | 16 | 10 | 24 | 12 |
| | SAW | | 16 | 10 | 24 | 12 |
| Logique floue | 20 | 16 | 10 | 24 | 12 | |
| SAW | | 16 | 10 | 24 | 12 | |
| Scénario 3 | Logique floue | 5 | 6 | 10 | 23 | 14 |
| | SAW | | 6 | 10 | 23 | 14 |
| | Logique floue | 10 | 10 | 7 | 27 | 10 |
| | SAW | | 11 | 4 | 21 | 17 |
| | Logique floue | 15 | 16 | 10 | 24 | 12 |
| | SAW | | 16 | 10 | 24 | 12 |
| Logique floue | 20 | 16 | 10 | 24 | 12 | |
| SAW | | 20 | 7 | 21 | 19 | |

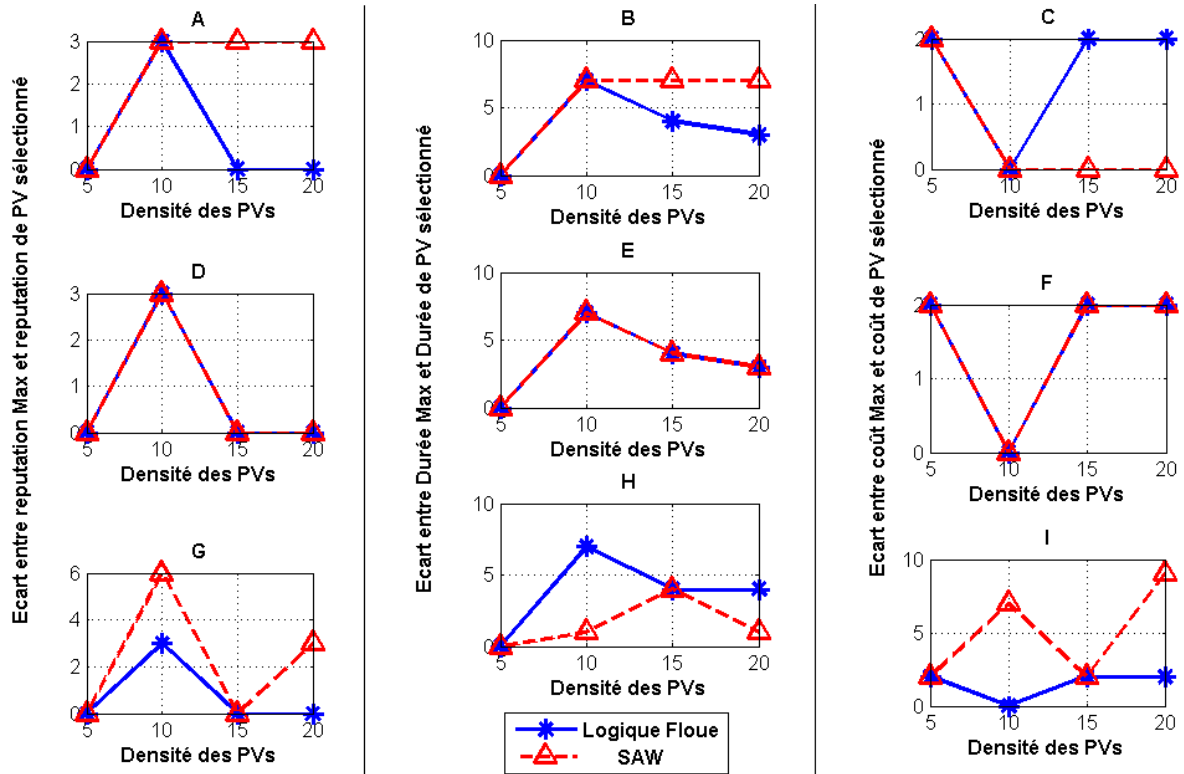


Figure 3.3: Écart de SAW et logique floue

3.3.4 Conclusion

Dans ce chapitre, nous avons consacré la première partie pour présenter une vue générale sur la logique floue qui est un concept sur lequel on s'est basé pour améliorer le protocole DCCS-VC.

Dans la deuxième partie du chapitre, nous avons décrit le principe de l'idée proposée pour l'amélioration, la construction de la base des règles, l'optimisation des règles et l'environnement de simulation, enfin nous avons présenté une comparaison par simulation entre la technique de la logique floue proposée et la technique SAW.

Conclusion générale et perspectives

La fusion des deux concepts du Cloud Computing et les réseaux véhiculaire VANets a donné naissance au Cloud véhiculaire, un concept dans lequel les services du Cloud Computing sont hébergés par des véhicules qui possèdent des ressources suffisantes pour agir en tant que serveurs Cloud mobiles.

Dans ce mémoire, nous avons présenté les protocoles proposés pour l'offre et la consommation des services dans le Cloud véhiculaire, ainsi que les méthodes de sélection des services dans le Cloud classique telles que les méthodes **MCDM**, méthodes basées sur la logique, les méthodes d'optimisation... etc. Puis, nous avons proposé une amélioration au protocole **DCCS-VS** (**D**iscovering and **C**onsuming **C**loud **S**ervices in **V**ehicular **C**loud) au niveau de la sélection des services en se basant sur le concept de la logique floue. En fin, nous avons comparé par simulation les performances de la technique de la logique floue proposée et la technique **SAW** (**S**imple **A**dditive **W**eighting).

Les résultats de la simulation ont montré que l'amélioration permet au consommateur d'atteindre le meilleur service dans le Cloud selon ses demandes et des critères de qualités.

Comme extensions futures à notre travail nous proposons:

- De résoudre le problème de la sélection avec d'autres techniques, par exemple: les méthodes **MCDM**, les méthodes d'optimisations... etc.
- Faire des études comparatives entre la technique de la logique floue et d'autres techniques de sélection de services dans le Cloud.

Références

- [1] M. Watfa, *Advances in Vehicular Ad-Hoc Networks: Developments and Challenges: Developments and Challenges*. IGI Global, 2010.
- [2] “Cloud Computing.” <http://www.salesforce.com/fr/cloudcomputing/>. [Online; consulte le 28-avril-2016].
- [3] K. Merhad and H. Artail, “Finding a star in a vehicular cloud,” *Intelligent Transportation Systems Magazine, IEEE*, vol. 5, no. 2, pp. 55–68, 2013.
- [4] B. Brik, N. Lagraa, A. Lakas, and Y. Ghamri-Doudane, “Rcs-vc: Renting out and consuming services in vehicular clouds based on lte-a,” in *Global Information Infrastructure and Networking Symposium (GIIS), 2015*, pp. 1–6, IEEE, 2015.
- [5] B. Brik, N. Lagraa, A. Lakas, and Y. Ghamri-Doudane, “Finding a public bus to rent out services in vehicular clouds,” in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*, pp. 1–7, IEEE, 2015.
- [6] F. Deroncourt, “Introduction à logique floue.” <http://franck-deroncourt.developpez.com/tutoriels/algo/introduction-logique-floue/>. [Online; consulte le 9-avril-2016].
- [7] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, “Cloud service selection: State-of-the-art and future research directions,” *Journal of Network and Computer Applications*, vol. 45, pp. 134–150, 2014.
- [8] H. Saouli, *Découverte de services web via le cloud computing à base d’agent mobile*. PhD thesis, Université Mohamed Khider Biskra, 2015.
- [9] “Architecture distribuée, centreon.” <https://documentation-fr.centreon.com/docs/centreon/en/2.5.x/architecture/03b.html>. [Online; consulte le 1-May-2016].

- [10] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *Journal of internet services and applications*, vol. 1, no. 1, pp. 7–18, 2010.
- [11] H. Jin, S. Ibrahim, T. Bell, W. Gao, D. Huang, and S. Wu, “Cloud types and services,” in *Handbook of Cloud Computing*, pp. 335–355, Springer, 2010.
- [12] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [13] M. Gerla, “Vehicular cloud computing,” in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pp. 152–155, IEEE, 2012.
- [14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, P. Chrétienne, and X. Cazin, *Introduction à l’algorithmique*. Dunod, 2001.
- [15] “First order logic, whatis .” <http://whatis.techtarget.com/definition/first-order-logic..> [Online; consulte le 28-avril-2016].
- [16] “ Descriptions Logic.” https://en.wikipedia.org/wiki/Description_logic, 2016. [Online; consulte le 28-avril-2016].
- [17] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, “Vehicular ad hoc networks (vanets): status, results, and challenges,” *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.
- [18] senouci, “ Communications véhicule à véhicule: applications et perspectives.”
- [19] H. Mousannif, I. Khalil, and S. Olariu, “Cooperation as a service in vanet: Implementation and simulation results,” *Mobile Information Systems*, vol. 8, no. 2, pp. 153–172, 2012.
- [20] S. Olariu, T. Hristov, and G. Yan, “The next paradigm shift: from vehicular networks to vehicular clouds,” *Mobile Ad Hoc Networking: Cutting Edge Directions, Second Edition*, pp. 645–700, 2013.
- [21] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, “A survey on vehicular cloud computing,” *Journal of Network and Computer Applications*, vol. 40, pp. 325–344, 2014.

- [22] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, “Rethinking vehicular communications: Merging vanet with cloud computing,” in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pp. 606–609, IEEE, 2012.
- [23] “3gpp specification: evolved universal terrestrial radio access (eutra) and evolved universal terrestrial radio access network (eutran)..” <http://www.3gpp.org/DynaReport/36300.htm>. [Online; consulte le 10-Mars-2016].
- [24] B. T. Scheme, “Lte: the evolution of mobile broadband,” *IEEE Communications magazine*, p. 45, 2009.
- [25] H. Arkian, R. Atani, and S. Kamali, “Fcvca: A fuzzy clustering-based vehicular cloud architecture,” in *Communication Technologies for Vehicles (Nets4Cars-Fall), 2014 7th International Workshop on*, pp. 24–28, IEEE, 2014.
- [26] C. Rosenta, *Histoire de la logique oue. Une approche sociologique des pratiques de démonstration*. 2006.
- [27] F. G. Francois CHEVRIE, *Cahier technique n 191, La logique floue*. 1998.
- [28] B. Meunier, “Logique floue, principes, aide à la décision,” *Hermès*, 2003.
- [29] G. Cabodevila, “Commande et identification floue.”
- [30] G. Cormier, *La logique floue*. 2011.
- [31] J. Aguilar-Martin, “La logique floue et ses applications industrielles,” *Quaderni*, vol. 25, no. 1, pp. 75–86, 1995.
- [32] N. Ragot, *Reconnaissance de formes par modélisation mixte intrinsèque/discriminante à base de systèmes d’inférence floue hiérarchisés*. PhD thesis, Rennes 1, 2003.
- [33] “Simulateur OMNET++.” <http://docplayer.fr/13487930-Simulateur-omnet.html>. [Online; consulte le 1-May-2016].
- [34] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, “Sumo—simulation of urban mobility,” in *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, 2011.
- [35] “Veins.” <http://veins.car2x.org/>, 2016. [Online; consulte le 9-May-2016].
- [36] R. J. Fernandes, “Large-scale simulation of vehicular ad hoc networks,” 2014.