



**HAL**  
open science

# ITS-based decision making mechanism for opportunistic networking in heterogeneous network environment

Rodrigo Silva

## ► To cite this version:

Rodrigo Silva. ITS-based decision making mechanism for opportunistic networking in heterogeneous network environment. Networking and Internet Architecture [cs.NI]. Ecole nationale supérieure Mines-Télécom Atlantique, 2020. English. NNT : 2020IMTA0174 . tel-02941004v2

**HAL Id: tel-02941004**

**<https://theses.hal.science/tel-02941004v2>**

Submitted on 18 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THESE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPERIEURE MINES-TELECOM ATLANTIQUE  
BRETAGNE PAYS DE LA LOIRE - IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 601  
*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*  
Spécialité : *Informatique*

Par

**Rodrigo SILVA**

## **ITS-based Decision Making Mechanism for Opportunistic Networking in Heterogeneous Network Environment**

Thèse présentée et soutenue à Rennes, le 27 janvier de 2020

Unité de recherche : **Systemes, reseaux, cybersécurité et droit du numérique (SRCD) / IRISA**

Thèse N° : **2020IMTA0174**

### **Rapporteurs avant soutenance :**

Yacine Ghamri-Doudane  
Farouk Kamoun

Professeur à l'Université de La Rochelle  
Professeur émérite à ENSI Tunis / Sesame

### **Composition du Jury :**

Président : Houda Labiod  
Examineurs : Farouk Kamoun  
Yacine Ghamri-Doudane  
Thierry Ernst  
Dir. de thèse : Jean-Marie Bonnin

Professeure à Telecom Paris  
Professeur émérite à ENSI Tunis / Sesame  
Professeur à l'Université de La Rochelle  
Docteur, PDG  
Professeur à l'IMT Atlantique

# Acknowledgements

I would like to thank my family. Although most of them are thousands of miles away, I am deeply thankful to them for their support, understanding, endless patience, and encouragement. Special thanks go to my wife Mary Ane, my little daughter Anne-Louise, my mom Marta, my father Celson (in memorian), my sister Elisama and my nephews Maria Eduarda and Benjamin. Thanks also to my friend Emanuel.

I am thankful to Jean-Marie Bonnin for accepting me as a Ph.D student. And, also, for his continuous advice and encouragement during my thesis. I particularly appreciated his support and understanding in difficult situations that I lived outside of the thesis context.

I would like to thank the Professors Frédéric Weis and Xavier Lagrange for accepting to be members of my CSID committee and also for their sound advices during my thesis.

I am happy with the partnership we had in this thesis with the Yogoko company. I would like to thank Thierry Ernst for his support.

Finally, I would like to thank all my colleagues. Particular thanks go to Christophe Couturier, Alessandro Laudin Molina, Renzo Navas, Indra Ngurah and Mahdi Ezzaouia.

# Résumé en français

Le nombre de véhicules connectés augmentent et pour une utilisation plus intelligente des systèmes de transport, les véhicules ont besoin d'accroître leur connaissance de l'environnement. Cela peut être atteint en permettant aux véhicules de communiquer avec leur environnement.

Pour une connectivité omniprésente, il sera nécessaire d'utiliser diverses technologies sans fil, telles que le WiFi véhiculaire (ITS-G5 et Dedicated Short Range Communications (DSRC)), le WiFi urbain (par exemple, 802.11 ac, g, n), 802.15.4, et cellulaire (3G, 4G et 5G en préparation). Dans un tel environnement de réseau d'accès hétérogène, il est nécessaire fournir aux applications des mécanismes décisionnels transparents pour gérer l'affectation des flux de données sur les réseaux disponibles.

Dans cette thèse, nous proposons le Ant-based Decision Maker for Opportunistic Networking (AD4ON), un Decision Maker (DM) mécanisme modulaire capable de choisir le meilleur profil de communication disponible pour chaque flux de données dans un environnement de réseau hétérogène et dynamique. Le mécanisme proposé gère les exigences et préférences de différents acteurs (applications, utilisateurs, etc.), ainsi que les information réseaux dans un futur proche, afin de prendre des décisions plus intelligentes. C'est-à-dire, afin d'augmenter la satisfaction de chaque acteur impliqué dans le processus de communication, ainsi que d'augmenter la stabilité de la décision (par exemple, en réduisant l'effet "ping-pong").

## 1.1. Introduction

Le numéro de véhicules est en croissance, avec une prévision d'arriver à 2 million de véhicules vendu en 2035. Cette croissance a un impact sur la qualité de vie, comme l'augmentation du trafic routier, l'augmentation des embouteillages, et du nombre d'accidents routière. Le nombre de morts et de blessés sur les routes sont élevés dans le monde. Selon le National Highway Traffic Safety Administration (NHTSA), il y a plus de 1,700 morts et 840,000 blessés chaque année dans les routes nord-américaines [1].

Malgré la grande variété de contre-mesures appliquées, telles que les différentes lois (en limitant la vitesse sur les autoroutes ou l'utilisation obligatoire de la ceinture de sécurité), et le développement de systèmes pour réduire l'intervention humaine sur le processus de conduite, le trafic reste chaotique et le nombre de morts et de blessés sur les routes restent élevé. Pour améliorer le système de transport, les véhicules doivent augmenter leur perception environnementale plus que les quelques mètres obtenus avec les systèmes actuels. Ceci pourrait être réalisé en permettant aux véhicules de communiquer.

Une fois les véhicules connectés et coopératifs, un écosystème de services peut être développé autour d'eux. Cette connexion peut être locale entre des appareils à proximité ou globale, c'est-à-dire une connexion sur Internet. Les véhicules peuvent se connecter localement pour améliorer la sécurité et pour l'assistance au conducteur. Par exemple, un véhicule peut communiquer avec d'autres véhicules se trouvant à proximité pour informer sur les embouteillages locaux, les accidents ou pour alerter sur une arrêt d'urgence. De même, les véhicules peuvent se connecter globalement pour améliorer l'expérience à bord, par exemple, en améliorant le service de navigation, ou en offrant un accès Internet à bord.

Ce Cooperative Intelligent Transportation Systems (C-ITS) environnement se caractérise par son hétérogénéité et dynamisme. Il y a une grande variété de applications, chacune avec un ou plusieurs flux de données ayant des exigences de communication spécifiques. Utilisateurs et administrateurs avec ces préférences et règles. De plus, en raison de la vitesse des véhicules, le contexte de leur connectivité peut changer fréquemment. La disponibilité des réseaux et ces conditions peuvent varier rapidement. Par conséquent de cette hétérogénéité et dynamisme, une seule technologie ne peut pas prendre en charge tous ces spécificités. Il est nécessaire utiliser toutes les technologies sans fil existantes, telles que le WiFi véhiculaire (ITS-G5 en Europe, ou DSRC en Amérique du Nord), WiFi urbain (par exemple, 802.11 g / n / ac), 802.15.4 ou cellulaire (3G, 4G et 5G sous préparation).

Cela conduit à un environnement radio complexe où une grande variété de réseaux d'accès pourrait être disponible. Dans un tel contexte, les applications ne peuvent prendre en compte toutes les particularités technologiques, sauf s'ils en ont explicitement besoin. Au lieu de cela, un DM est nécessaire pour gérer simultanément tous les réseaux d'accès disponibles, essayant de choisir pour chaque flux de données, les réseaux d'accès qui correspondent le mieux à leurs exigences.

La prise de décision dans un environnement C-ITS présente quelques défis. Tout d'abord, comme déjà mentionné, il y a une énorme variété d'acteurs pouvant présenter ces exigences de communication (par exemple les applications, les utilisateurs et les administrateurs). De plus, l'environnement réseau est composé par une grande variété de technologies. En outre, les véhicules doivent être capable de communiquer avec différents dispositifs. En somme, due la

grande mobilité des véhicules, l'environnement réseau peut changer fréquemment. Les véhicules peuvent se déplacer à grande vitesse, faisant apparaître et disparaître rapidement les réseaux.

Basé sur nos recherches et sur les normes International Organization for Standardization (ISO) et European Telecommunications Standards Institute (ETSI), nous avons identifié certaines propriétés importantes pour un mécanisme de prise de décision dans cet environnement C-ITS. Ensuite, nous proposons le AD4ON mécanisme, qui consiste d'une architecture et un algorithme capable de gérer les exigences et préférences de différents acteurs (applications, utilisateurs, etc.). L'AD4ON est développée en conformité avec les normes ISO Intelligent Transportation Systems (ITS). De plus, pour aborder le dynamisme des véhicules, nous avons conçu et intégré la prévision des environnements de communication (c'est dire, l'information du futur proche) dans le processus décisionnel. Ainsi, l'AD4ON est capable d'anticiper ses décisions.

## 1.2. Propriétés attendues d'un DM

Basé sur nos recherches et sur l'architecture des ITS proposé par ISO, nous avons identifié certaines propriétés que nous considérons importantes pour prendre des décisions dans un environnement C-ITS:

- Architecture modulaire - une architecture modulaire semble plus appropriée pour mieux assurer la coopération entre des dispositifs hétérogènes. De plus, une telle architecture peut être divisée en entités séparées dans le réseau véhicule;
- Gestion d'attributs et d'objectifs multiples - dans la communication C-ITS, différents acteurs peuvent présenter leurs besoins, leurs préférences, leurs contraintes et politiques dans le processus de prise de décision, comme illustré à la Figure 1.1.

Le DM doit prendre en compte tous ces attributs, afin de choisir la réseau qui satisfait le mieux aux besoins de chaque acteur.

- Gestion de réseau hétérogène - Comme il n'existe pas une seule technologie de communication capable de répondre à toute les besoins des services, il est nécessaire d'utiliser les différents technologies sans fil existantes, telles que WiFi véhiculaire (ITS-G5 en Europe ou DSRC en Amérique du Nord), WiFi urbain (par exemple, 802.11 g / n / ac / ax), 802.15.4 ou cellulaire (3G, 4G et 5G en préparation). Dans ce contexte, un DM est nécessaire pour gérer simultanément tous les réseaux d'accès disponibles, en essayant de choisir pour chaque flux de données, les réseaux d'accès qui correspondent le mieux à leurs besoins;

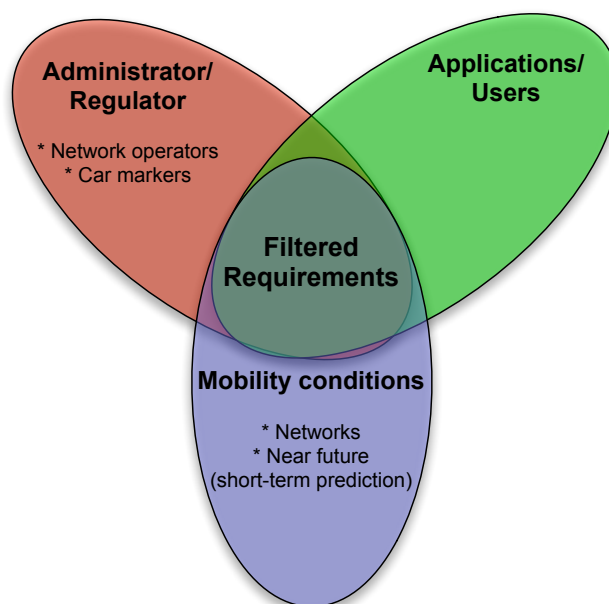


Figure 1.1: Les attributs multiples des différentes acteurs impliqués dans le processus de la prise de décision

- Connaissance de l'environnement véhicule - Le véhicule doit connaître son environnement, c'est à dire, l'environnement où il est inséré. Le DM doit surveiller les réseaux autour du véhicule ainsi que les flux en communication, afin qu'il puisse mieux gérer la communication de tous les flux;
- Gestion des flux par flux - il est possible de gérer la communication en différents couches de communication (par exemple en considérant paquet par paquet à la couche transport, ou en gérant interface réseau par interface réseau). Nous avons choisi de gérer la communication flux par flux. Comme chaque flux peut avoir des exigences spécifiques, une gestion flux par flux permet au DM de prendre des décisions en choisissant pour chacun des flux le réseau d'accès qui répond mieux à leurs exigences. De plus, une telle gestion permet au DM de mieux utiliser les ressources du réseau, par exemple en répartissant les flux entre les différents réseaux d'accès disponibles;
- Anticiper les décisions - dans l'environnement C-ITS, les véhicules peuvent bouger à haute vitesse et, donc changer fréquemment son environnement réseau. Dans ce cas, les réseaux peuvent apparaître et disparaître rapidement. Dans un scénario dynamique comme cela, le DM devrait disposer d'informations sur le future proche des réseaux. Si le véhicule est au courant d'environnement réseau dans lequel il sera bientôt inséré, le DM est capable d'anticiper les décisions pour améliorer la communication des flux.

## 1.3. AD4ON

Pour atteindre ces propriétés identifiées précédemment, nous avons conçu l'architecture modulaire AD4ON. Cette architecture est basée sur l'architecture proposée par ISO [2]. Figure 1.2 illustre les principaux modules qui composent chaque partie de l'architecture AD4ON.

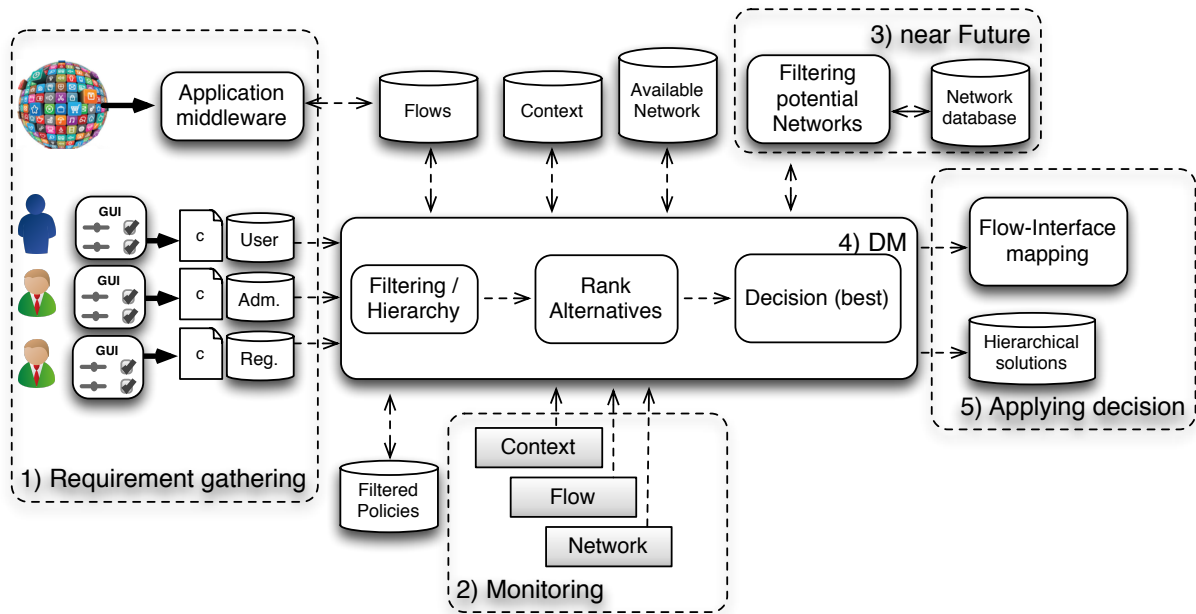


Figure 1.2: L'architecture AD4ON

Pour une meilleure compréhension, nous avons divisé l'architecture AD4ON en cinq parties principales, décrites ci-dessous.

### 1.3.1. Obtention des besoins de communication

Dans l'architecture AD4ON, nous considérons quatre acteurs en mesure de soumettre leurs exigences dans le processus de décision: les applications, les utilisateurs, administrateurs et organismes de réglementation. Besoins, préférences et les politiques de tous les acteurs sont stockées dans des bases de données et utilisées par le DM pour choisir le réseau d'accès qui correspond mieux aux besoins de chaque acteur.

### 1.3.2. Modules de surveillance

Pour prendre des décisions dans l'environnement C-ITS, il est nécessaire de surveiller diverses informations telles que la disponibilité et les performances des réseaux, l'état des flux et des



informations de contexte du véhicule.

Nous avons défini trois modules de surveillance (monitoring): le module de surveillance de réseau qui surveille les réseaux sans fil disponibles et leurs performances; le module de surveillance de contexte, responsable de surveiller des informations telles que la localisation des véhicules voisins, les embouteillages, vitesse du véhicule; et le module de surveillance de flux. Ce dernier module doit indiquer si un flux est actif ou non et évaluer les performances des flux, comme la bande passante utilisée et la latence.

#### 1.3.3. Future proche

Cette partie est responsable pour prendre en compte les information du future proche. Information comme disponibilité et performances des réseau sont prédits ici et ensuite envoyés vers le DM.

Lorsque le DM reçoit des informations d'un futur environnement, il n'est pas garanti que le véhicule trouvera les mêmes performances prévues lorsqu'il atteindra cet environnement. Les réseaux peuvent avoir changé entre temps, ou plusieurs dispositifs peuvent avoir connecté. Par conséquent, plus proche le véhicule est du environnement prévu, plus probable qu'il pourra rencontrer l'environnement dans les mêmes conditions prévues.

Nous classons les futurs environnements réseau en fonction de sa distance, comme illustré par la Figure 1.3.

- Très court terme (environ 10 secondes, par exemple): environnements proches de la position actuelle du véhicule;
- Court terme (quelques minutes): sont des environnements à quelques minutes de la position actuel du véhicule;
- À moyen terme (environ 10 minutes, par exemple): sont les environnements qu'un véhicule peut atteindre dans quelques dizaines de minutes;
- Long terme: sont des environnements qui peuvent être atteints en une heure ou plus.

Une fois que le DM prédits le future proche, il doit prendre en compte la précision de sa predictions en tenant en compte les différents classes de future décrites précédemment.

Afin de mettre en évidence l'utilité du future proche dans le processus de prise de décision, nous avons effectué simulation avec un simple scénario. La simulation démontre qu'en connaissant le future proche, l'AD4ON est capable d'adapter son comportement afin de trouver des solutions plus satisfaisantes au flux de communication.

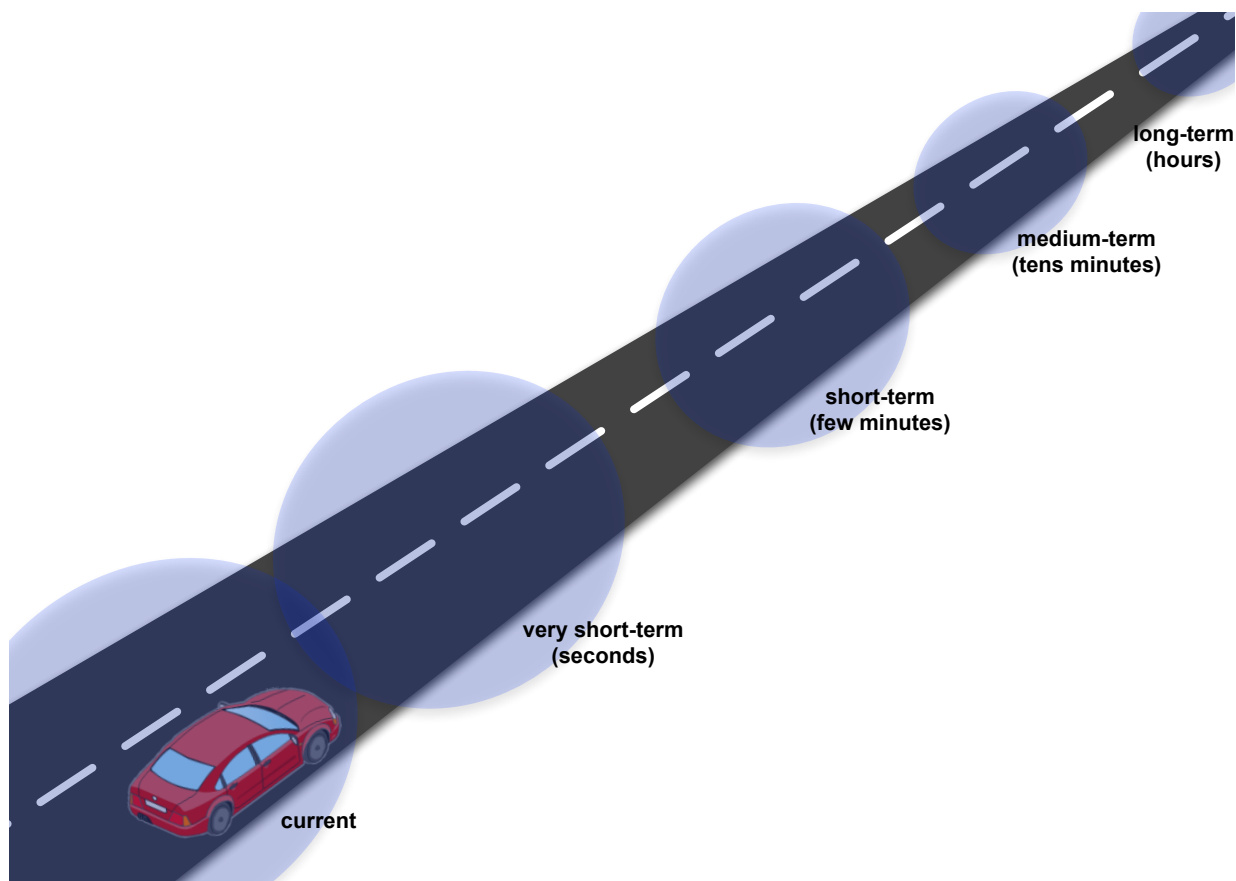


Figure 1.3: Classification des future proche

#### 1.3.4. Prise de décision (DM)

Le processus de décision est chargé de prendre en compte les exigences des flux, des utilisateurs, des administrateurs, ainsi que les différentes informations des réseaux, afin de gérer la communication des flux.

Afin de prendre en compte les différents types d'entrées, nous avons divisé le processus de prise de décision en trois modules: "Hiérarchie / Filtrage", "Liste d'alternatives" et "l'Algorithme de décision".

Le "Hiérarchie / Filtrage" module est chargé de recevoir et de gérer les exigences, les préférences et les politiques des différents acteurs. Comme les acteurs peuvent avoir leurs propres préférences et exigences, nous avons besoin "filtrer" (au sens informatique) les différentes valeurs définies pour un même paramètre.

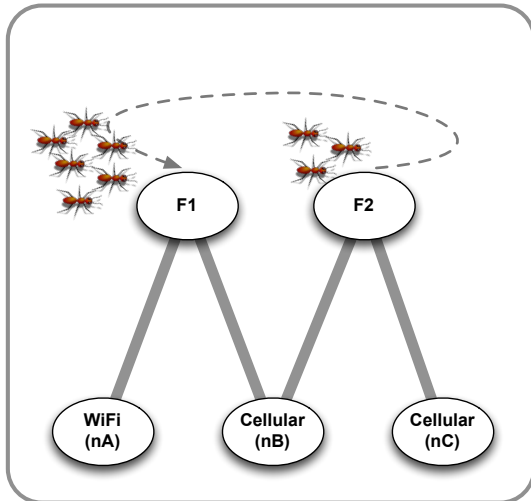
Le module "Liste d'alternatives" est responsable de la recherche de toutes les alternatives pour la communication de chaque flux. Ce module est aussi un filtre pour les réseaux interdits par un administrateur ou ceux qui ne correspondent pas aux besoins des flux.

L'algorithme de décision reçoit la liste de toutes les solutions potentielles, précédemment

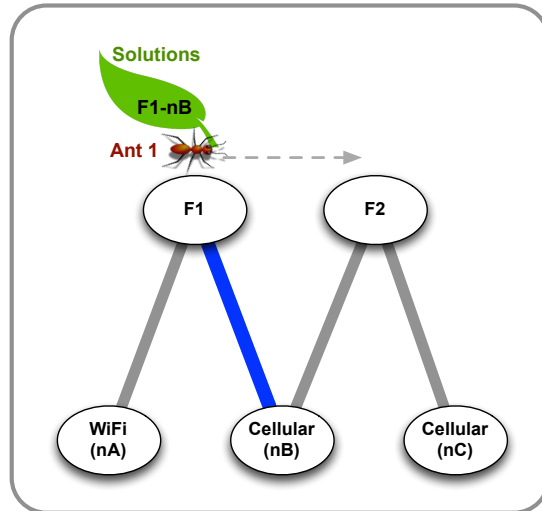
créées dans le module “Liste d’alternatives”, et il évalue les réseaux afin de trouver celui qui correspond le mieux les exigences de communication de chaque flux.

Nous avons développé l’algorithme de décision AD4ON basé sur la classe d’algorithmes appelé Ant Colony Optimization (ACO), parce que ils ont des propriétés qui peuvent être explorés pour répondre aux besoins de communication en environnement C-ITS, telles que la stabilité de décision, pour être adaptable en temps d’exécution, ou permettre la gestion de plusieurs flux simultanément.

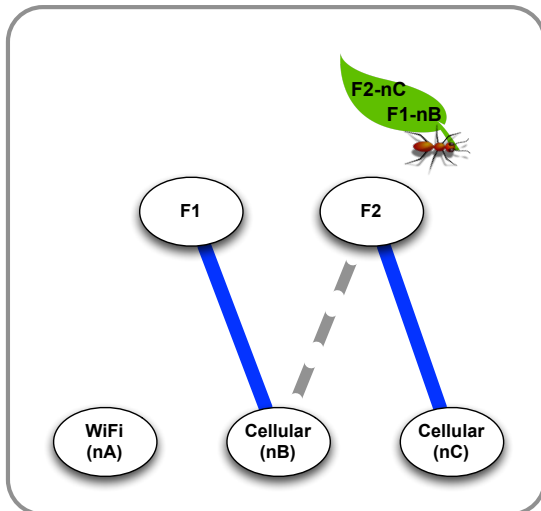
La Figure 1.4 illustre le processus de décision de l’AD4ON. Dans ce exemple, deux flux (F1 et F2) sont en concurrence pour communiquer à travers les trois réseaux d’accès (nA, nB et nC). Nous modélisons notre problème de sélection de réseau sous forme d’un graphe bipartite, comme illustré par la Figure 1.4a. Les flux sont considérés comme le nid de la colonie et les réseaux comme la source de nourriture. Les fourmis localisées dans un flux donné (nid), cherchent le meilleur réseau d’accès (source de nourriture) parmi tous les solutions potentiels. Par exemple, en regardant la Figure 1.4a, les fourmis du flux 1 peuvent choisir entre le réseau A et le réseau B, tandis que les fourmis du flux 2 peuvent choisir entre le réseau B et le réseau C.



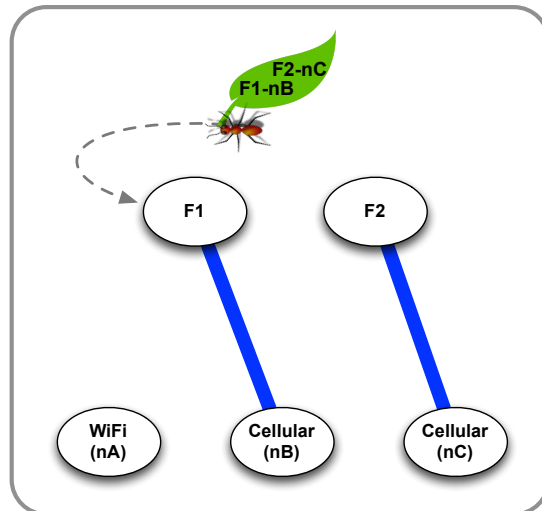
(a) Distribution des fourmis parmi les flux



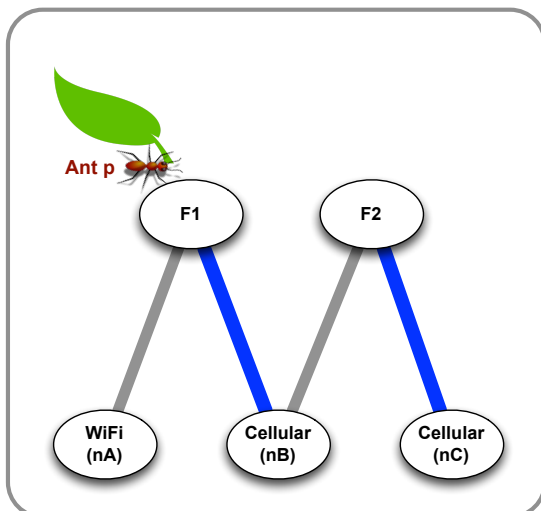
(b) Fourmi 1 commence par le flux F1



(c) Tour de la première fourmi (en flux F2)



(d) Retour de la formi au flux de départ



(e) La visite des prochaines fourmis considèrent la phéromone déposée par les fourmis précédentes

Figure 1.4: Exemple du processus de décision de l'AD4ON

Au début, l'algorithme AD4ON distribue les fourmis de manière aléatoire parmi les flux, comme indiqué par la Figure 1.4a. En supposant que la fourmi 1 choisit le réseau B pour fournir le Data Rate (DR) requis par le flux F1, elle stocke cette solution dans son mémoire et elle se dirige vers le flux suivant (Figure 1.4b). Nous supposons aussi que le réseau B a la capacité de recevoir un seule flux, donc une fois arrivé au flux F2, le fourmi 1 sait que le réseau B n'a plus la capacité de recevoir d'autre flux (comme illustré à la Figure 1.4c). Dans ce cas, la seul solution pour le flux F2 est le réseau C. La fourmi 1 choisit le réseau C pour le flux F2 et retourne à son point de départ (Figure 1.4d).

Les prochaines tours de l'algorithme AD4ON, les fourmis prennent en compte la quantité de phéromone déposée par les fourmis précédentes (Figure 1.4e). Au final, la solution pour chaque flux sera celui avec la plus forte concentration de phéromone.

### 1.3.5. Appliquer la décision

Dans le processus de l'application de la décision, les réglés produites par le DM sont appliqués dans le système. Dans ce processus, le DM peut interagir avec des entités contrôlées dans les différents couches de l'architecture de communication afin de bien acheminé le flux vers la réseau choisi.

Si des changements inattendus arrivent, comme par exemple la chute d'une réseau, l'AD4ON peut changer la communication pour une autre réseau. Pour cela, l'AD4ON maintient une base de données de solutions hiérarchisé avec toutes les solutions sous-optimales pour chaque flux. Cette base de données de solutions hiérarchisé est utilisée en cas de d'urgence, c'est-à-dire lorsque la meilleure solution réseau tombe inopinément et jusqu'à ce que le DM trouve un autre meilleure solution.

## 1.4. Conclusion

Ce résumé met l'accent sur les motivations de cette thèse, ainsi que illustre les travaux réalisé pour la définition de l'AD4ON, un mécanisme modulaire pour le processus de prise de décision dans un environnement de réseaux hétérogènes et changeantes.

Le manuscrit de thèse qui suit donne plus de détails de ce mécanisme et le compare aux autre solutions présentes dans la littérature. Nous proposons aussi l'utilisation du future proche pour améliorer la qualité de la prise de décisions.

# Abstract

Demand from different actors for extended connectivity where vehicles can exchange data with other vehicles, roadside infrastructure and traffic control centers have pushed vehicle manufacturers to invest in embedded solutions, which paves the way towards Cooperative Intelligent Transportation Systems (C-ITS). Cooperative vehicles enable the development of an ecosystem of services around them. Due to the heterogeneousness of such services and their specific requirements, as well as the need for network resources optimization, for ubiquitous connectivity it is necessary to combine existing wireless technologies, providing applications with a communication architecture that hides such underlying access technologies specificities. Moreover, due to vehicles' high velocity, their connectivity context can change frequently. In such a scenario, it is needed a Decision Maker (DM) mechanism capable to consider requirements from all actors, as well as take into account the short-term prevision about network environment in order to better manage all flow communications.

Based on the Intelligent Transportation Systems (ITS) architecture proposed by International Organization for Standardization (ISO), we proposed the Ant-based Decision Maker for Opportunistic Networking (AD4ON), a modular decision maker mechanism capable to choose the best available access network for each data flow in an heterogeneous and dynamic network environment. The proposed mechanism manages requirements and preferences from different actors (e.g., applications, users, administrators and regulators), and it takes into account the short-term prevision about the network environment in order to better satisfy the actors requirements.

Simulations have demonstrate that the AD4ON outperforms current benchmark algorithms like the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), by increasing decision's stability, reducing the "ping-pong" effect and maximizing flow's satisfaction. Moreover, we demonstrate by simulation that taking into account the short-term prevision, the AD4ON can optimize the algorithm reaction time, enabling flows to take better advantage of new networks as soon as they become available.

---

## Contents

<b>Acknowledgements</b> .....	<b>i</b>
<b>Abstract in French</b> .....	<b>ii</b>
<b>1. Introduction</b> .....	<b>1</b>
<b>1.1. Challenges</b> .....	<b>6</b>
1.1.1. Actors heterogeneities.....	6
1.1.2. Network heterogeneities .....	6
1.1.3. Devices heterogeneities .....	7
1.1.4. Highly dynamic mobility .....	7
<b>1.2. Thesis contribution</b> .....	<b>8</b>
<b>1.3. Thesis outline</b> .....	<b>9</b>
<b>2. Technical Context</b> .....	<b>11</b>
<b>2.1. Standardization efforts</b> .....	<b>13</b>
2.1.1. SILO approach.....	13
2.1.2. WAVE architecture .....	14
2.1.3. C2C-CC architecture .....	15
2.1.4. ISO ITS reference architecture .....	16
<b>2.2. Enforcement and Mobility management adopted by ISO</b> .....	<b>24</b>
<b>2.3. Example of ISO ITS architecture implementation</b> .....	<b>26</b>
<b>2.4. Conclusion</b> .....	<b>27</b>
<b>3. Literature review</b> .....	<b>29</b>
<b>3.1. Research works related with DM architecture</b> .....	<b>32</b>
3.1.1. Decision Making (DM) .....	32
3.1.2. Requirements gathering.....	35

3.1.3. Monitoring process.....	37
3.1.4. Applying decision .....	38
3.1.5. How works have addressed the DM architecture .....	42
<b>3.2. Research works related with DM algorithms .....</b>	<b>44</b>
3.2.1. MADM.....	45
3.2.2. ACO .....	49
3.2.3. How works have addressed the DM algorithms .....	52
<b>3.3. Attributes .....</b>	<b>54</b>
<b>3.4. Conclusion .....</b>	<b>56</b>
<b>4. AD4ON architecture.....</b>	<b>58</b>
<b>4.1. Expected properties .....</b>	<b>59</b>
4.1.1. Modular architecture .....	59
4.1.2. Multiple attributes and objectives management .....	60
4.1.3. Management of heterogeneous network.....	61
4.1.4. Awareness of the vehicle surroundings.....	62
4.1.5. Flow per flow management.....	63
4.1.6. Anticipate decisions.....	63
<b>4.2. Architecture design.....</b>	<b>64</b>
4.2.1. Main communication interfaces with AD4ON .....	64
4.2.2. AD4ON architecture .....	68
<b>4.3. Security and Privacy .....</b>	<b>78</b>
<b>4.4. Integration of the AD4ON in the ITS-S communication archi- tecture.....</b>	<b>79</b>
<b>4.5. Conclusion .....</b>	<b>80</b>



---

<b>5. AD4ON algorithm</b> .....	<b>81</b>
<b>5.1. Expected properties</b> .....	<b>82</b>
5.1.1. Decision stability .....	83
5.1.2. Allow partial recalculation .....	84
5.1.3. Take into account the near future .....	84
<b>5.2. The AD4ON Algorithm</b> .....	<b>84</b>
5.2.1. Filtering/Hierarchy .....	87
5.2.2. Rank Alternatives .....	87
5.2.3. Decision Algorithm .....	89
5.2.4. Why ACO-based algorithm.....	89
5.2.5. Problem representation .....	90
5.2.6. Utility Functions.....	91
5.2.7. General description of the AD4ON algorithm.....	94
<b>5.3. Stop condition of the AD4ON algorithm</b> .....	<b>99</b>
<b>5.4. AD4ON's parameter setting</b> .....	<b>101</b>
5.4.1. Properties achieved by the AD4ON algorithm.....	102
<b>6. Evaluation of the AD4ON algorithm</b> .....	<b>104</b>
<b>6.1. The evaluation process</b> .....	<b>105</b>
6.1.1. Evaluation overview.....	105
6.1.2. Input data.....	107
6.1.3. Making decision .....	114
6.1.4. Results.....	116
6.1.5. Comparison .....	116
<b>6.2. Description of simulation scenarios and results</b> .....	<b>118</b>
6.2.1. Scenario A .....	119
6.2.2. Scenario B .....	124

6.2.3. Scenario C .....	128
<b>6.3. Conclusion .....</b>	<b>133</b>
<b>7. Near Future .....</b>	<b>135</b>
7.1. Recent research work .....	136
7.2. How to acquire the near future .....	137
7.2.1. Near future about flows .....	138
7.2.2. Near future about networks environments.....	138
7.3. Decision Making using near future .....	142
7.3.1. Near future classification .....	142
7.3.2. How to apply the near future .....	144
7.4. Integrating prediction in the AD4ON .....	146
7.5. AD4ON-NF evaluation.....	149
7.5.1. Scenario A with Near Future .....	149
7.6. Conclusion .....	152
<b>8. Conclusions .....</b>	<b>154</b>
8.1. Future work .....	156
<b>Appendices.....</b>	<b>158</b>
<b>Annexe 1 – Key performance indicator .....</b>	<b>159</b>
<b>Annexe 2 – Number of networks for Scenario C.....</b>	<b>164</b>
<b>List of Acronyms.....</b>	<b>165</b>
<b>List of Publications .....</b>	<b>171</b>
<b>Bibliography.....</b>	<b>172</b>

## List of Figures

1.1. Les attributs multiples des différentes acteurs impliqués dans le processus de la prise de décision . . . . .	v
1.2. L'architecture AD4ON . . . . .	vi
1.3. Classification des future proche . . . . .	viii
1.4. Exemple du processus de décision de l'AD4ON . . . . .	x
1.1. Example of services for connected and cooperative vehicles . . . . .	4
2.1. Example of silo architecture . . . . .	14
2.2. WAVE architecture . . . . .	15
2.3. ISO ITS-S reference architecture . . . . .	16
2.4. Example of communication between MN and CN. . . . .	20
2.5. Flow assignment. . . . .	21
2.6. Example of NEMO + MCoA . . . . .	25
2.7. Example of ISO ITS architecture implementation in a single onboard device . . .	26
2.8. Example of ISO ITS architecture implementation in various onboard devices . . .	27
3.1. Decision Making process . . . . .	30
3.2. Decision Making classification based on decision location . . . . .	32
3.3. Requirements gathering. . . . .	36
3.4. Decisions at different layers. . . . .	39
3.5. Decision matrix template . . . . .	45
3.6. Process of ants searching for food . . . . .	50
3.7. Percentage of attributes per actors . . . . .	55
3.8. Flows requirements . . . . .	55
3.9. Network constraints . . . . .	56
4.1. DM mechanism Inputs/Output . . . . .	59
4.2. Multiple attributes/actors . . . . .	60
4.3. Heterogeneous access networks . . . . .	62
4.4. Communication towards MA-SAP and MN-SAP . . . . .	65
4.5. The structure of the <i>MA-Request.request</i> service. . . . .	66
4.6. The structure of the <i>MA-Request.confirm</i> service. . . . .	67
4.7. AD4ON architecture outline . . . . .	69
4.8. Proposed AD4ON Architecture . . . . .	70

4.9. Integration of AD4ON Architecture in the ITS-S communication architecture (centralized implementation). . . . .	79
5.1. Overview of AD4ON DM part . . . . .	85
5.2. An example of vehicle scenario used to illustrate how the AD4ON algorithm works	85
5.3. Flows competing to communicate through access networks . . . . .	86
5.4. Graph of feasible solutions . . . . .	91
5.5. Potential solutions . . . . .	95
5.6. Example of the AD4ON decision making process . . . . .	96
5.7. An ant in flow $i$ chooses the access network as a function of the pheromone values $\tau$ and the heuristic values $\eta$ on the arcs connecting flow $i$ to the access network $j$ .	101
6.1. AD4ON implementation . . . . .	106
6.2. Simulation overview . . . . .	107
6.3. Example of ITS-G5 network environment created from real measures . . . . .	109
6.4. Deployed ITS network scenario. . . . .	110
6.5. Equipment used. . . . .	111
6.6. Circuit of tests with ITS-G5 and Cellular coverage. . . . .	112
6.7. Measures from testbed . . . . .	113
6.8. Input of network parameters for scenario A . . . . .	121
6.9. Results for scenario A. . . . .	122
6.10. Key performance for streaming flow. . . . .	126
6.11. Total utility . . . . .	128
6.12. Example of DR required by the 20 flows in the sub-scenario C1. . . . .	130
6.13. Example of network DR availability for the sub-scenario 1 composed by 7 access networks. . . . .	131
6.14. Results for scenario C - Streaming. . . . .	132
7.1. Classification of network prediction . . . . .	139
7.2. Example of prediction confidence . . . . .	143
7.3. Classification of near future environments . . . . .	144
7.4. Example of AD4ON-NF approach. . . . .	148
7.5. Input of flow and network parameters for scenario A. . . . .	150
7.6. Comparison between AD4ON without near future and AD4ON considering near future information . . . . .	151

**List of Tables**

*LIST OF TABLES*

---

3.1. Main research works related with DM architecture . . . . .	43
3.2. Main research works related with DM algorithm . . . . .	53
4.1. Parameters of the MA-Request-request service . . . . .	66
4.2. Parameters of the MA-Request-confirm service . . . . .	67
4.3. Parameters of the MA-Command-request service . . . . .	67
4.4. Parameters of the MN-Request-request service . . . . .	68
4.5. Parameters of the MN-Command-request service . . . . .	68
4.6. Communication requirements between Application and DM . . . . .	71
4.7. Communication requirements between User and DM . . . . .	72
4.8. Communication requirements between Administrators and DM . . . . .	73
4.9. Example of output from the Hierarchy/Filtering module . . . . .	75
4.10. Example of output from the Ranking Alternative module . . . . .	76
5.1. Network parameters . . . . .	86
5.2. Flow requirements . . . . .	87
5.3. Network parameters . . . . .	87
5.4. Output from the Hierarchy/Filtering module . . . . .	88
5.5. Output from the Ranking Alternative module . . . . .	88
5.6. Output of the AD4ON algorithm . . . . .	99
5.7. AD4ON parameters . . . . .	102
6.1. Range of flow requirements . . . . .	108
6.2. Components used in the testbed . . . . .	111
6.3. Simulation scenarios . . . . .	120
6.4. Appl requirements . . . . .	120
6.5. Key performance results for scenario 1. . . . .	124
6.6. Range of Network Parameters . . . . .	125
6.7. Example of flows of sub-scenario C1 . . . . .	129
7.1. Key performance results for AD4ON and AD4ON-NF . . . . .	152

# 1

## Introduction

The number of vehicles on the planet is growing rapidly. According Ward's research, in 2010 they were more than 1 billion in operation worldwide, and total new vehicles sales suggests that there could be up to 2 billion vehicles by 2035 [3]. Moreover, more people live in urban areas than in rural areas. And cities are expected to continue growing. The United Nations estimates that in 2050 about 66% of the world's population would live in urban areas [4]. Such growth has a great impact on the quality of human life. Space is becoming insufficient to accommodate all vehicles. The road traffic is increasing, as well as traffic jams and the number of traffic accidents.

The number of deaths and injuries on roadways is high over the world. According to National Highway Traffic Safety Administration (NHTSA) of U.S Transportation Department, each year there are more than 1,700 fatalities and 840,000 injuries due to vehicle crashes on U.S public highways. Car crashes are the number one killer of children in the United States [1]. In Europe, more than 30,000 people died on the roads in 2011 [5].

Furthermore, the motor vehicles crashes are costly for economy. The cost components include several factors as medical costs, rehabilitation costs, congestion costs, police and fire services, productivity losses, and others.

The governments over the world have applied a variety of countermeasures in order to reduce road traffic accidents, as laws to regulate road traffic, or automotive systems to help driver in the driving process. In the following, we highlight the Europe Commission initiatives and the French

---

situation, but similar analysis could have been made in other countries. Europe Commission launched in 2011, the “Road Safety Programme” that aims to cut road deaths in Europe in half until 2020. The program comprises a mix of initiatives improving vehicle safety, improving safety of infrastructure and change road users’ behavior [6]. Since 1970, France has progressively implemented solutions as road laws limiting velocity on the highways, setting limits to the blood alcohol concentration (BAC) levels for drivers, introduction of points system for driver’s license, mandatory use of seat belt and others. The result of these countermeasures was the reduction of deaths on roadways. The number of death in France roads reduced from more than 18,000 in 1972 to less than 3,500 in 2014 [7].

Despite the wide variety of countermeasures applied by governments over the world, the transportation system still needs improvements. The traffic remains chaotic and the number of deaths and injuries on roadways remains high. Once main cause of accidents and crashes are due to human errors, it is necessary to reduce the human intervention on driving process.

Then, it is needed automobile systems to assist drivers for safety and better driving. Automobile manufacturers have worked to develop these systems. Carmakers have developed a wide variety of vehicle-related safety measures that can be divided in passive and active features, according to [8].

- Passive features – among the passive components, we can highlight the safety improvement of car body structures, implantation of seat belts, airbags and head restraints;
- Active features – as active components we can enumerate Electronic Stability Control (ESC), Anti-Lock Braking System (ABS) and so-called Advanced Driver Assistance Systems (ADAS), like Adaptive Light Control (ALC), legal speed limit assistance, Forward Collision Avoidance (FCA), Autonomous Emergency Braking (AEB) and others.

Car manufactures are adopting and developing a variety of ADAS sensors that allow vehicles to sense their environment, like cameras, radar and Lidar. However, despite the increasing number of in-vehicle sensors, the environmental perception of the vehicle remains limited to few meters. Only with these previous listed safety measures it is difficult to reduce more the number of death. To improve these systems, the vehicles need to increase their environment awareness. This could be achieved enabling vehicles to communicate locally between themselves and with the infrastructure (Vehicle-to-Everything (V2X)). After large pilot deployments, the European Commission is preparing a Delegated Act to bootstrap mass deployment, whereas some vehicle manufacturers are already starting to equip new series of vehicles with communication capabilities, e. g., Toyota, General Motors, and Volkswagen.

In this context, Intelligent Transportation Systems (ITS) becomes as a solution to a smarter use of transportation networks. In a first stage, this communication could be non-cooperative, i.e., vehicles receive information from the environment (e.g., alert of emergence breaking from

---

the vehicle in front) and make their own decisions (e.g., alert the driver or automatically reduce the speed). In the future, the goal is to achieve a truly cooperative communication. Connectivity and V2X communications enable vehicles to communicate with a wide variety of devices. This paves the way towards Cooperative Intelligent Transportation Systems (C-ITS), where vehicles, the roadside infrastructure, the urban infrastructure and control centers exchange information for a smarter and more efficient use of the road.

Besides the requirements for smarter use of transportation systems, other actors have also pushed the need for better connectivity. Original Equipment Manufacturers (OEM)s have requested for over-the-air (OTA) updates, enabling securely managing all in-vehicle software components (including firmware, applications, and configurations) anywhere and at any time. Demand for navigation services improvements, e.g., improving maps quality by using high definition maps. New demand for infotainment services, for example, saving driver profile on the cloud and applying it to any vehicle he/she drives, i.e., personalized infotainment pre-sets like ambient temperature, seat and mirror positioning, and favorite radio channels. Besides making traveler journeys more pleasant, some infotainment services have addressed more advanced issues, such as multimodal transportation planning. Combining information from different sources like traveler information provider (i.e., systems providing real-time public transport timetable information), vehicle's information, and traffic jam information, can facilitate to propose a wide range of mobility solutions.

Connectivity demands from consumers, business as well as government legislation have pushed vehicle manufacturers to invest in embedded connectivity solutions. As a result, the number of connected cars grows continuously. According to Gartner research company, by 2020 up to 80% of new vehicles will be connected to digital services, and connected cars would be a major element of the Internet of Things (IoT) [9]. Thanks to the miniaturization of mechanical, optical and electronic products and devices, nowadays it is possible to embed communication systems and sensors in many objects and places in a city. Such objects have the capability to acquire and exchange data with others, enabling the development of smart cities, in which vehicles will play an essential role.

Once vehicles become connected and cooperative, an ecosystem of services can be developed around them, as shown on Figure 1.1. Such connection can be local between nearby devices or global, i.e., connection over the Internet. Vehicles can connect locally in order to improve safety and driver assistance. For example, a vehicle can connect with other vehicles in its vicinity to inform about local traffic jams, accidents or to alert about emergency breaking. Similarly, vehicles can connect globally to enhance driver and passenger experience, e.g., improving the navigation service, offering on-board Internet access, or connecting with car dealers or car repair shops to maintain a regular maintenance schedule and technical repairs.





Figure 1.1: Example of services for connected and cooperative vehicles

In this context, users, devices and vehicles need to be connected anywhere, anytime with anything. Such environment is characterized by its heterogeneity. There is a wide variety of applications, each one with one or more data flows that have specific communication requirements. For example, a safety-based service (e.g., emergency breaking information) is highly sensitive to packet loss and latency, whereas a video streaming service is less sensitive to latency and bandwidth changes.

Due to the heterogeneousness of such services and their specific requirements, a single access technology cannot support all their connectivity needs. Moreover, due to the high monetary cost associated with network deployment, it is impractical to rely solely on a dedicated access technology for specific services. Therefore, for ubiquitous connectivity it is necessary to use all existing wireless technologies, such as vehicular WiFi (ITS-G5 in Europe, or Dedicated Short Range Communications (DSRC) in North America), urban WiFi (e.g., 802.11 g/n/ac/), 802.15.4 or cellular (3G, 4G, and 5G under preparation).

Vehicles equipped with multiple communication capabilities and running heterogeneous applications can use all such technologies at once in order to offer better Quality of Experience (QoE) for users, while making smarter use of available access networks, e.g., maximizing communication bandwidth, performing traffic load balancing among available networks, managing handovers to offer seamless communication, as well as choosing the access network that better matches communication requirements.

Due to the high velocity of vehicles, their connectivity context can change frequently, i.e.,

---

network availability and network conditions can vary rapidly. In such heterogeneous and dynamic network environment, applications cannot take into account all technology particularities, unless they explicitly need it. Instead, it is preferable to provide applications with a communication architecture that hides the heterogeneity of underlying access technologies, providing seamless communications. Moreover, possibility of having multiple applications simultaneously competing for communication resources requires a controlled access to these resources. In this context, a Decision Maker (DM) is needed to manage all available access networks simultaneously, attempting to choose for each data flow, the access networks that better match with their requirements. Therefore, applications are not limited to a single access technology, but they can take advantage of all available technologies.

Attempting to enable interoperability between such heterogeneous technologies and enabling cooperation between different services, some standardized bodies have worked through an harmonized communication architecture, e. g., the ITS Station (ITS-S) reference communication architecture proposed by International Organization for Standardization (ISO). Such ITS architecture combines multiple access technologies and common data management services.

These standards give some guidelines for ITS services developments and communication management. However, based on such guidelines developers have the freedom to develop their own mechanisms, which can be a competitive factor between stakeholders. An example of solutions based on ITS standards are the ones offered by the Yogoko<sup>1</sup> company, with whom we realized the present thesis in partnership. Yogoko is a communication solutions provider for connected, cooperative and autonomous vehicles. Yogoko has developed an unified communication platform enabling the communication of various types of applications through heterogeneous communication technologies (e. g., WiFi, Cellular, and LoRa). The Yogoko communication system is based in three pillar:

- Y-BOX - it is composed by a communication box (hardware) equipped with heterogeneous access technologies;
- Y-SMART - it is composed by a communication and data management platform (software), which enables a variety of applications to communicate through different access networks;
- Y-CLOUD - through the Y-CLOUD platform, Yogoko enables the communication and service management through an offboard platform.

## 1.1. Challenges

---

<sup>1</sup>[www.yogoko.com](http://www.yogoko.com)

The environment of connected and cooperative vehicles presents some challenges that are described below.

### 1.1.1. Actors heterogeneities

In a connected vehicle different actors (like applications, users, network administrators and regulators) should be able to present their requirements, preferences, constraints and policies. There is a wide variety of applications, each one with one or more data flows that have specific communication requirements. Flows can request a specific Data Rate (DR), latency, security level and more. For example, a safety-based service (e.g., emergency breaking information) is highly sensitive to packet loss and latency, whereas a video streaming service is less sensitive to latency and bandwidth changes. Users can present their preferences, e.g., defining a priority or security level for a given message. Industrial and mobility service providers (i.e., operators) can present their policies, such as network constraints and particular billing procedures.

Such wide variety of objectives can be contradictory. A network operator can aim at improving network load balancing without regard to monetary cost, while an user can prefer a cheaper network. Moreover, the requirements and preferences from different actors can change over time. An user can accept an expensive network if he or she has an high budget and reject them when the budget is below a threshold.

### 1.1.2. Network heterogeneities

Initial works related to ITS have assumed a universal communication technology being used for all ITS usages. However, due the high monetary cost associated with network deployment as well as the complex regulation issues to fully cover all countries, it is impractical to rely solely on a dedicated access technology for specific services.

For ubiquitous connectivity and to better fulfill requirements from the heterogeneous applications, it is necessary to use all existing wireless technologies, such as vehicular WiFi (ITS-G5 in Europe, or DSRC in North America), urban WiFi (e.g., 802.11 g/n/ac/), 802.15.4 or cellular (3G, 4G, and 5G under preparation).

In addition, the constant evolution of wireless access technologies leads to a complex radio environment where a wide variety of access networks could be available. Each of these networks has specific characteristics in terms of bandwidth, data rate, security and others. Due to this networks heterogeneity and its complementary characteristics, more connectivity opportunities are available.

Mobile devices equipped with multiple communication capabilities could take advantage of

these heterogeneous technologies. They can use multiple access technologies at once in order to maximize communication bandwidth, or choose the one that best matches communication requirements of a given application, perform traffic load balancing among available networks, manage handovers, and offer seamless communication, and others.

### 1.1.3. Devices heterogeneities

In the C-ITS environment, besides the communication between vehicles (Vehicle-to-Vehicle (V2V)) and between vehicles and infrastructure (Vehicle-to-Infrastructure (V2I)), vehicles should be able to communicate with a wide variety of connected devices. The connection can be local between neighbors' devices (e.g. connection with a sensor in the roadside) or global, i.e. connection over the Internet.

Moreover, embarked devices have different characteristics in terms of memory, CPU and communication capabilities. Such devices cannot embark multiple communication interfaces, and/or in some cases they do not have communication management capabilities. Therefore, it is needed a dedicated device in charge of managing communication between such on board devices and the outside. This dedicated device is called Mobile Router (MR) by the ISO/European Telecommunications Standards Institute (ETSI) standards.

### 1.1.4. Highly dynamic mobility

The possibility of having multiple applications simultaneously competing for communication resources requires a controlled access to these resources. In this context, a DM algorithm is needed to manage all available access networks simultaneously, attempting to choose for each data flow, the access networks that better match with their requirements.

A large number of research studies have concentrated on the development of decision algorithms for network selection. They aim at combining requirements, preferences and policies in the decision making process. However, the existing decision making algorithms do not meet the communication needs in the vehicular environment.

Due the highly mobility of vehicles, network environment can change frequently. Vehicles can move at high speed causing networks to appear and disappear rapidly. Moreover, inside the coverage of a given network, vehicle can experience different network conditions, in terms of bandwidth availability, latency and Packet Delivery Ratio (PDR).

In such an environment, the DM algorithm should present some properties. It should be capable of handling routing flow per flow. This allows the DM to select the most appropriate path for each flow as well as to manage flow priorities. It should be capable of handling multiple

objectives from different actors (applications, users, network administrators, and regulators), as well as multiple access technologies simultaneously. It should be capable to find high-quality solutions in a reasonable time. Due the vehicles mobility, recalculate a complete solution every time a network parameter changes can be an NP-complete problem. Therefore, the DM algorithm should prevent full recalculation when network parameters have only slight changes and an adaptation of the previous solution remains satisfactory. The DM algorithm should be run-time adaptable, i.e., it should adapt to the network conditions and vehicle context.

Moreover, in such highly dynamic mobility scenarios, besides the current vehicular environment, it is needed to take into account the short-term prevision about the vehicle context, i.e., predict its surroundings and the network environment in which the vehicle will be soon inserted. If the DM is aware about the near future of the vehicle environment it can anticipate decisions. For example, it can decide to increase the data buffer for a given video streaming, if the vehicle is going to cross a wireless dead zone; anticipate a network switching, by preparing the handover process in advance; or, delay a data transmission if it knows that a better network will soon be available.

## 1.2. Thesis contribution

Despite all works in decision making process, there is no viable solution available on the market that meets communication needs in the vehicular environment.

Based on our research, on the ITS architecture proposed by ISO and a survey of the literature, we identified the good properties such a decision mechanism should have. And then, we propose here the Ant-based Decision Maker for Opportunistic Networking (AD4ON), a DM mechanism that meet such identified properties. The AD4ON mechanism is composed by an architecture (AD4ON architecture) and an algorithm (AD4ON algorithm).

The main contribution of this thesis are summarized as follows:

- A literature review about recent works in multi-criteria decision making architecture and algorithms was performed;
- Design of the modular AD4ON architecture based on the ISO ITS-S architecture. Such DM architecture is capable to manage requirements and preferences from different actors (applications, users, administrators and regulators), it handles multiple access technologies simultaneously, and it takes into account the short-term prevision about the network environment in order to make smarter decisions;
- Based on ISO standards, we defined some information to be exchanged between the DM and other ITS-S architecture layers;

- Design, implementation and tests of the AD4ON algorithm based on the well-known Ant Colony Optimization (ACO) algorithms. The AD4ON algorithm is able to manage multiple flows and multiple access networks simultaneously, attempting to choose the best access network for each data flow while increasing decision stability, reducing the “ping-pong” effect and managing decisions flow by flow to maximize flow requirements satisfaction;
- Design and integration of the communication environment predictions (i.e., the near future information) in the decision making process, in order to perform proactive decisions.

## 1.3. Thesis outline

The rest of such manuscript is organized as follow:

Chapter 2 overviews some technical elements and architecture that have been addressed by different entities (i.e., standardization bodies, vehicles manufacturers, equipment suppliers and research institutions) toward a convergent ITS architecture.

Chapter 3 identifies how works have addressed networking management, analyzing some of the most significant work related to decision making process. We divided the literature review in three phases. The first phase describes how works have addressed the DM architecture. The second phase identifies the main algorithms that have been used in decision making process. The third and last phase is a continuation of the previous one, in which we identify the most frequent attributes used by the DM algorithms in the decision making process.

Based on our research and on the ITS architecture proposed by ISO, in the Chapter 4 we design the AD4ON architecture, a modular architecture for opportunistic networking in heterogeneous access network environment.

Chapter 5 identifies the main properties for decision making in vehicular environments that are not addressed by existing DM algorithms. Then, we develop the AD4ON algorithm, an ACO-based algorithm capable to manage multiple access networks simultaneously, attempting to choose the best access network for each data flow.

In Chapter 6, we evaluate the AD4ON mechanism. In this chapter, we demonstrate that the AD4ON is capable to address the expected properties identified in the previous chapter. With a mix of real and simulated input data, we performed simulation and comparison between four DM: AD4ON, Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), modified TOPSIS (mTOPSIS) and Commercial DM (CM).

Due the high mobility of vehicles, it is useful to predict the network environment in which the vehicle will be soon inserted. In Chapter 7, we discussed about the use of near future information

in the decision making process and we developed a version of the AD4ON algorithm that takes into account such near future information, in order to improve its decisions.

Finally, Chapter 8 concludes the thesis, reviewing the main findings of our work and suggesting some topics for further research.

# 2

## Technical Context

As discussed on the Introduction part, in order to improve ITS systems, the environment awareness of vehicles must be increased. Therefore, vehicles should communicate between themselves and with connected devices in their surrounding. Connectivity and V2X communications enable vehicles to communicate with a wide variety of devices. This paves the way towards C-ITS, where vehicles, the roadside infrastructure, the urban infrastructure and control centers make decisions together for a smarter and more efficient use of the road.

In favor of making vehicles connected, the automotive industry is investing to include connectivity technologies into their vehicles. However, one of the main automakers' issues is which access technology they should adopt. Current discussions have polarized around two options. At the one hand, some car manufacturers like Volkswagen and Renault push short-range technology (i.e., the ITS-G5) to be used. They argue WiFi is widely used technology that can be implemented quickly. At the other hand, other major players like Daimler, BMW, Ford, PSA Group, Deutsche Telekom, Ericsson, Huawei and chip makers Intel, Samsung and Qualcomm push the long-range technology (5G). They claim 5G has wider application possibilities. For example, reducing drastically the communication latency, which will be crucial for the autonomous car.

A consensus is that local communication is needed for safety applications, while global communication (e.g., through the Internet) can be used for Internet-based services. Despite current discussions about connected vehicles is mainly centered around ITS-G5 and cellular technologies



---

(e.g., 4G and 5G), there are other existing access technologies that can participate to the C-ITS environment, like urban WiFi (e.g., 802.11 g/n/ac/), ZigBee and LoRa. Such wide variety of access technologies leads to a complex radio environment.

Once vehicles become connected, an ecosystem of services can be developed around them. We notice two main trends in vehicular applications:

1. Embedded applications, i.e., applications that are integrated in-vehicle hardware (e.g., maps and navigation software installed in instrument cluster). In this case, the vehicle is part of the connected system;
2. Smartphone content to the dashboard. In this case, applications like iOS CarPlay developed by Apple and Android Auto developed by Google enable mirror features from smartphone device to a vehicle's entertainment head unit. In such a solution the smartphone is the one inserted in the connected ecosystem instead of the vehicle.

In the industry, services have been developed independently and in an isolated way, following a silo approach. They are developed to solve specific problems and usually over proprietary systems. These kind of silo applications are normally implemented in dedicated devices with dedicated servers and database. Each application is mapped over specific and dedicated communication interface and use proprietary protocols. For example, Electronic Fee Collection (EFC) used in tolls road. Such system consists of dedicated roadside antennas, roadside units, in-vehicle onboard unit, and integrated circuit cards. Due to independent and isolated way of current connected services are developed, the cooperation between them is impractical or even impossible. In some case, two different services cannot cohabit. It is the case of the aforementioned EFC (DSRC) and applications communicating through ITS-G5. They use frequencies that are close to each other, which may cause interference.

The environment of C-ITS is characterized by its heterogeneity. There is a wide variety of applications, each one with specific communication requirements, simultaneously competing for communication resources. Different actors (e.g., applications, users, administrators and regulators) can participate in the decision making process by presenting their requirements, preferences, constraints and policies to the DM. While moving vehicles can cross a wide variety of network environments, each one composed by heterogeneous access technologies with specific performance. And, there are a wide variety of devices (inside the vehicles and/or devices on the road infrastructure), each one with different communication capabilities.

In such heterogeneous network environment, applications cannot take into account all technology particularities, unless they explicitly need it. Instead, it is preferable to provide applications with a communication architecture that hides the heterogeneity of underlying access technologies, providing seamless communications. Many stakeholders are involved in ITS-related services. To

guarantee cooperation between them and a well-balanced market development, standardization bodies have worked on the definition of a communication architecture.

Besides all these heterogeneity, vehicles can move at high speed frequently changing their network environment. In this context of heterogeneity and highly dynamic mobility, a decision maker mechanism is needed to manage all available access networks simultaneously, attempting to choose for each application, the access networks that better match with the communication requirements from the different actors (application requirements, users preferences, and administrator policies). Due to such highly dynamic mobility, it is desired a DM capable to predict future network conditions, in order to anticipate decisions.

In the following, we overview some technical elements and architecture that have been addressed by different entities (i.e., standardization bodies, vehicles manufacturers, equipment suppliers and research institutions) toward a convergent ITS architecture.

## 2.1. Standardization efforts

Standardization bodies, research institutions and many stakeholders have worked toward a convergent ITS architecture. In the following we describe some of such efforts, as well as we overview the most relevant ITS-related standards for this thesis.

### 2.1.1. SILO approach

In the absence of a standardized communication architecture, services and applications have been developed based on silo approach, as illustrated on Figure 2.1. In this type of architecture, applications are developed independently and isolated. They are developed to a specific problem and over a proprietary system. This kind of application is normally implemented in dedicated devices with dedicated servers and database. Each application is mapped over specific and dedicated communication interface and use proprietary protocols [10].

However, this approach has many drawbacks. It does not readily support change or innovation. Once a new technology or even a new service arrives, sometimes it is necessary to add new equipment and dedicated systems to embed it. This usually requires a complex and expensive process to change the infrastructure or to adapt them. Due to independent and isolated way of applications' development, it is complex to integrate these multi services in a single Human-Machine Interface (HMI). Moreover, it is difficult and expensive to both maintain and re-configure the system. It is necessary to manage particularities of each system independently.

As a result of silo application approach, several heterogeneous standards and technologies are

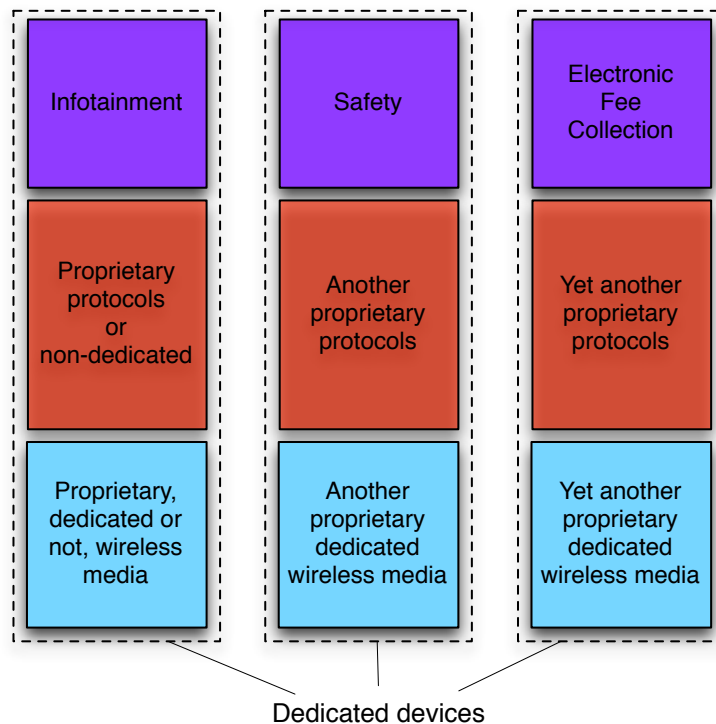


Figure 2.1: Example of silo architecture

available. However, such an applications do not cooperate among themselves.

### 2.1.2. WAVE architecture

Motivated to provide interoperability among vehicles and infrastructure, to achieve high level of security, comfort, and efficiency, the Institute of Electrical and Electronics Engineers (IEEE) developed an amendment to the 802.11 standards [11] to include vehicular environments. Such amendment is known as IEEE 802.11p and defines physical and MAC layers operating in the 5.9 GHz band with the goal of supporting DSRC for ITS.

While the IEEE 802.11p defines low layers, other family of IEEE standards ( IEEE 1609) define the upper layers. The set of both protocols stack composes the Wireless Access in the Vehicular Environment (WAVE) architecture [12]. Such architecture is mainly devoted to V2V and V2I wireless communications and it is shown in Figure 2.2.

The WAVE architecture presents a management plan and the capability to manage multiple radio channels (Control Channel (CCH) and Service Channel (SCH)). The CCH is mainly used for service announcement and safety application communication, while SCH are mainly used for Internet Protocol (IP) data transmission.

In order to accommodate different communication requirements, the WAVE architecture

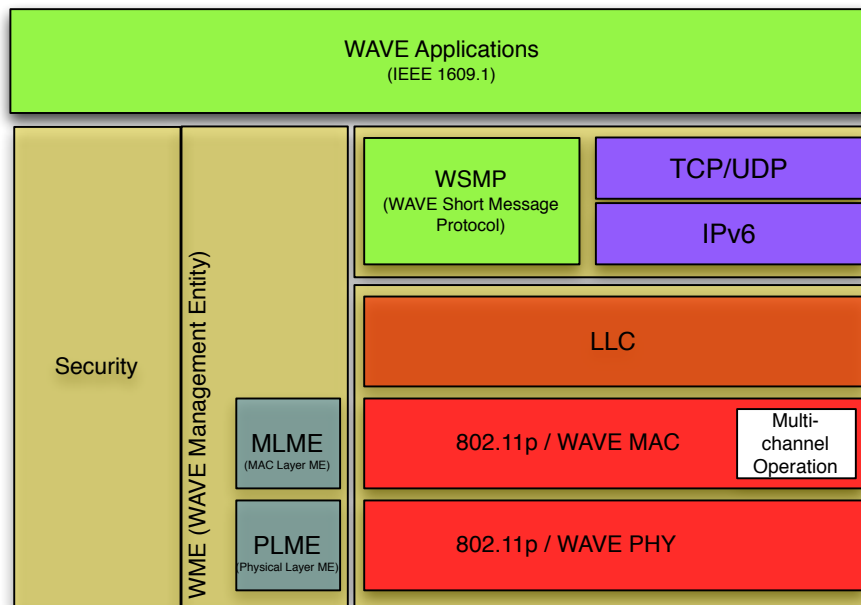


Figure 2.2: WAVE architecture

supports two protocol stacks. The WAVE Short Message Protocol (WSMP) is used for local communication (V2V and V2I) that are characterized by high-priority messages and time-sensitive communications. The Internet Protocol version 6 (IPv6) protocol is more dedicated for global communication over the Internet.

The WAVE architecture defines various functionalities related to applications, networking, management, communication and security. IPv6 has been chosen to allow Internet connectivity. Despite the capability to manage multiple radio channels, such architecture is not able to manage heterogeneous wireless access technologies.

### 2.1.3. C2C-CC architecture

In Europe, the Car-2-Car Communication Consortium (C2C-CC) [13] worked to standardize interfaces and protocols of wireless communication between vehicles and their environment. The C2C-CC was initiated by vehicle manufacturers and partners (e.g., BMW, Renault, Volkswagen, and Fiat) with the objective to increase road traffic safety and communication efficiency between vehicles.

The architecture proposed by C2C-CC is an evolution and an adaptation of the WAVE architecture. It supports the allocation of a frequency band (ITS-G5, based on IEEE 802.11p [14]) for safety critical messages. And allows the use of urban WiFi (e.g., IEEE 802.11 a/b/g) for Internet communication using IPv6.

### 2.1.4. ISO ITS reference architecture

ETSI and ISO have proposed convergent architectures. ETSI defined an ITS architecture that have been used in several testbed. However, in such architecture remains some vestiges of silo approach. ISO defined the ITS reference architecture that is layer oriented and, in which layers are independent of each others.

Both, the ETSI and ISO architectures have points in common and the tendency is that ETSI will converge towards the ISO architecture. In this thesis, we choose to work based on the ISO architecture, since it is the most conceptually advanced and better meets the thesis needs, i.e., it is an architecture capable to manage multiple and heterogeneous access networks, enabling the communication management of multiple and heterogeneous applications. And this is exactly what we intend to manage.

The technical committee ISO/TC 204 – ITS subcommittee prepared the ISO 21217 ITS standard, which describes the communications reference architecture of nodes called ITS-S [2]. This standard is the basis for several standards within ISO and beyond (ETSI and European Committee for Standardization (CEN)). Figure 2.3 shows the general ITS-S reference architecture, including interfaces between the various blocks with informative details.

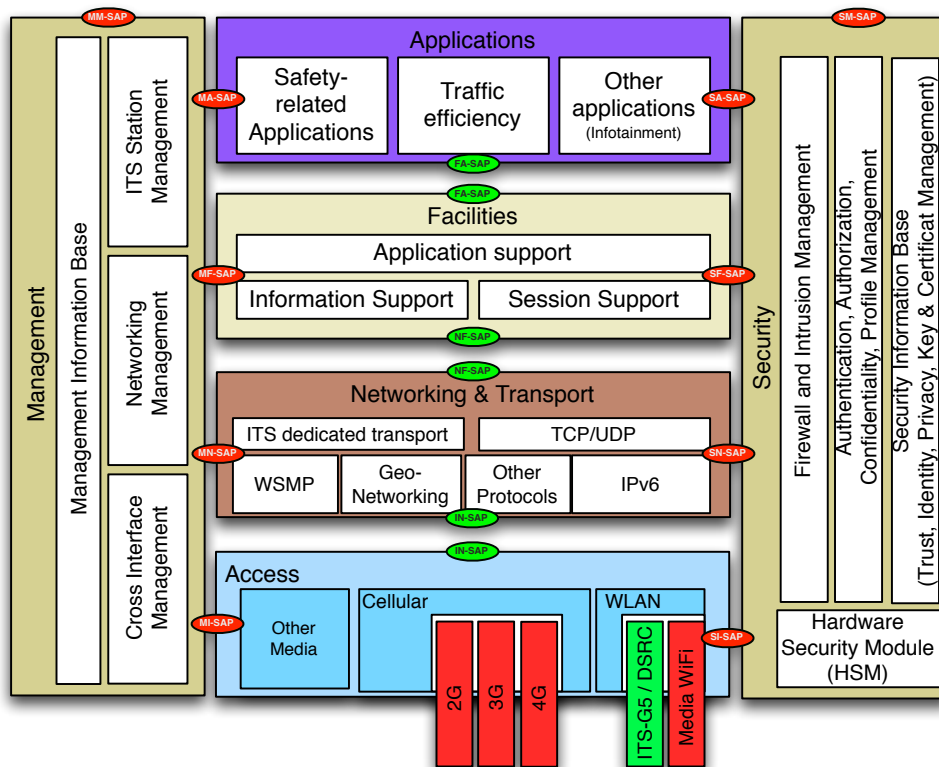


Figure 2.3: ISO ITS-S reference architecture

The ITS-S architecture is composed by four horizontal layers and two cross layers, which we

describe below:

- “Access” layer that provides means for ITS-S communication through different interfaces as vehicular WiFi (ITS-G5 in Europe, or DSRC in North America), urban WiFi (e.g., 802.11 g/n/ac/), cellular (3G, 4G, and 5G under preparation), etc. It is responsible for media access control and provides data transmission over physical links;
- “Networking & Transport” layer is responsible to execute operations like packet routing, path establishment, path monitoring and IP mobility;
- “Facilities” layer provides application, information and communication supports. As examples of these supports we can highlight the encode/decode messages support, common message distribution (e.g., Decentralized Environmental Notification Message (DENM), and Cooperative Awareness Message (CAM)), checking of received information, repetitive transmission of messages and more;
- “Application”, a horizontal entity that heberge applications. For example, such layer can provide HMI [2];
- “Management”, a cross entity that containing station management functionality. This entity communicates with all aforementioned layers in order to manage different process in the ITS-S, as communication profile selection, application management and communication interfaces management;
- “Security”, a cross entity that provides security and privacy services. This includes a set of procedures to establish a trust communication and to enable communication privacy, e.g., cryptography, authentication management, firewall and intrusion management.

The security in the ITS architecture is an important subject for connected mobile devices. Onboard Unit (OBU) should be reliable in order to ensure onboard systems work as intended and to mitigate safety risks. The communicating systems must have a strong cybersecurity environment to protect vehicles against cyberattacks. Since communicating systems like vehicles react to messages, it must ensure the authenticity of communicating devices. It must protect the data traffic against any unauthorized person. Critical vehicle systems, i.e., systems that govern safety (like braking systems) must be protected from unauthorized access and/or harmful attacks. Moreover, the communicating system must ensure users’ privacy, for example preventing an unauthorized person to acquire the geolocation of a mobile device. Despite we highlight the security importance in connected vehicles, in this present work we do not focus on that subject.

In the ITS-S concept, applications are abstracted from both the access technologies and the networks that transport the information between communicating nodes. Then applications are not limited to a single access technology, they could take advantage from all available technologies. However, this abstraction does not prevent application processes from requesting a specific communication profile to be considered in the communication profile selection process. For example, on the Networking & Transport layer there are two classes of network protocols identified. The Fast Geocasting protocols (e.g., FNTTP and WSMP) that are designed for application process with severe time constraints and low latency requirements, e.g., collisions risk warning, stationary vehicle information, roadwork warning, etc. The other class of network protocols is the Internet protocols (e.g., IPv6). This second class provides communication for global Internet services, e.g., infotainment, insurance and financial services, fleet management, etc. Therefore, an application that want to carefully manage the communication profile, could request a specific protocol among them.

The ITS reference architecture enables the management of communication flow by flow. ISO defines flow as an identifiable sequence of packets [15]. And packets are dependent of applied protocols, link and nodes characteristics. For example, packets sent over different paths to the same destination node experiment different network conditions/performances. This is a consequence of the protocols used by different layers, as well as the different characteristics of the path used (e.g., delay, throughput, and security level). According to such definition, an application is composed by one or more flows (i.e., data flows). For example, the Skype application has at least two types of flows: 1) real-time media, which is highly latency sensitive; and 2) signaling flows that is less sensitive to latency. Since flows have different communication requirements, managing communication flow by flow allows each flow requirement to be better met.

Besides ISO 21217, a large number of other standards compose the set of international standards for Communications Access for Land Mobiles (CALM). They specify functionalities required for all relevant layers and entities of the ITS-S architecture. Among these standards we can highlight:

- ISO 17423 [16] – such standard describes the ITS-S communication profile selection process. It defines the main functional modules (building blocks) needed to manage application communication. The process of selecting an ITS-S communication profile uses applications' requirements and objectives, set of rules from administrators and/or regulators, as well as communication protocols' status, in order to select the most appropriate communication profile for each application, i.e., to select suitable communication protocol stacks for each flow. A set of communication requirements is referred to as a Flow Type.

This determination enables the appropriate use of resources at the sending ITS-S, and it is necessary for interoperability with the recipient as the same protocols must be supported by

both communication end points.

How policies and regulations are made available in the management plan, as well as the mechanism used to select the most appropriate communication profile is outside the standardization scope. It is implementation dependent and can be a competitive factor between stakeholders. This is exactly where the work done here could be useful.

- ISO 24102-6 [15] – this standard specifies the procedures for the management of data flows communication. Once the Flow Type is chose (as defined by ISO 17423), it is necessary to choose the access network and communication interface that better meets flows communication requirements. ISO 24102-6 describes the requirements for different steps in flow communication management, e.g., requirements for flow registration and requirements for ITS-S capabilities management. The process of flow communication management is detailed in Section 2.1.4.1.

Like the communication profile process described in ISO 17423, the mechanism used to choose the better communication interface and access network for each flow is implementation dependent and outside the standardization scope.

- ISO 17429 [17] – defines useful procedures and generic mechanisms to designers and developers of ITS applications, enabling the exchange of data between ITS-S. It defines the following ITS-S facilities layer functionalities:
  - Communication Profile Handler (CPH) – entity responsible to check if there is a valid communication profile corresponding to flow requirements;
  - Content Subscription Handler (CSH) – is used to share message among multiple applications by means of publish/subscribe process;
  - Facilities Services Handler (FSH) – is used to apply different services to the data flow. For example, encryption, authentication, and compression.
- ISO 21210 – specifies IPv6 networking protocol functionalities between ITS-S communicating over the global Internet communication network. This standard defines modules like:
  - IPv6 forwarding module – as the name says, this module is responsible to forward IPv6 packets between layer above (i.e., Facilities layer) and the layer below (i.e., Access layer) through the “IPv6 LAN interface” or “External IPv6 interface”;
  - IPv6 LAN interface – responsible for transmitting IPv6 packets between the “IPv6 forwarding” module and stations belonging the same Local Area Network (LAN);



- External IPv6 interface – responsible for transmitting IPv6 packets between the “IPv6 forwarding” module and stations outside of the LAN.

**2.1.4.1. Example of communication between two ITS-S**

An ITS-S, like connected vehicles, may have more than one application running at the same time, e.g., safety related services, traffic monitoring, video streaming and onboard Internet accesses. Each of such applications is referred as “ITS-S application process” by ISO. And each one is identified by a unique reference number *ITS-AID* (of ASN.1 type), as specified by ISO 17419 [18]. In the follow, we depicts an example of application communication between two mobile devices, as defined by ISO standards. In this example, we show the main steps followed by a flow in the ITS-S sender (known as Mobile Node (MN)) before leaving to reach the ITS-S receiver (Correspondent Node (CN)). We divide such communication process in 6 steps, which are shown on Figure 2.4 and depicted in the follow:

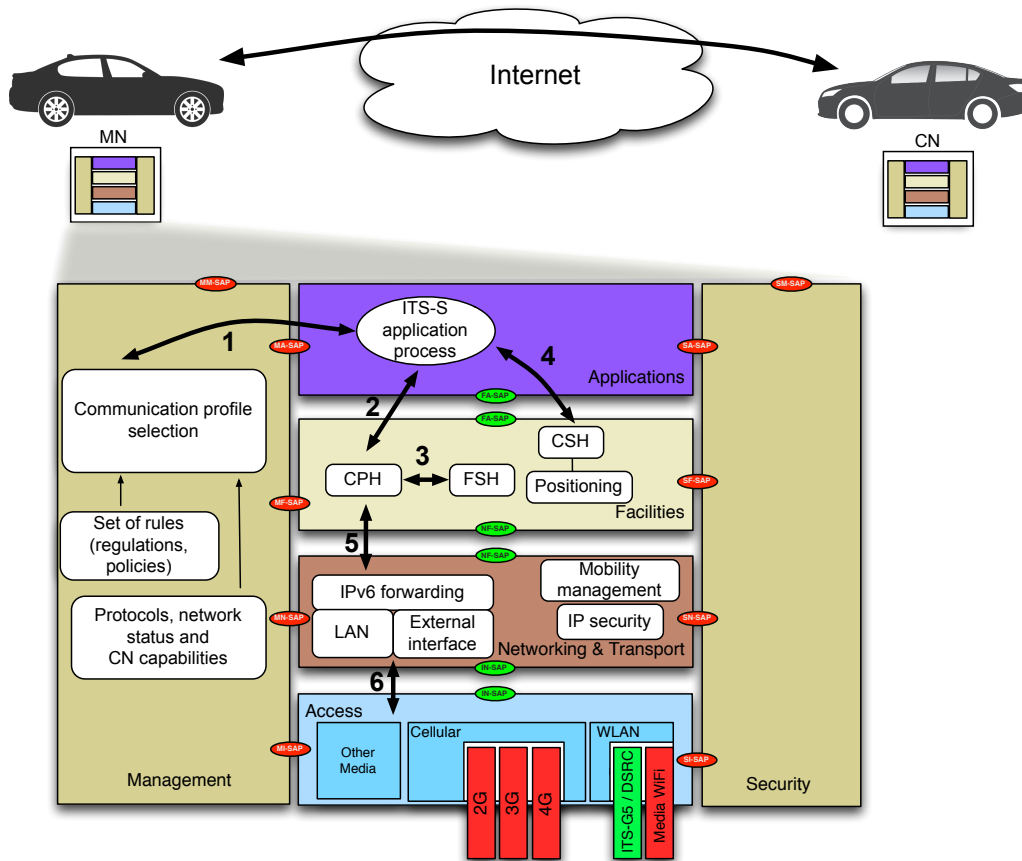


Figure 2.4: Example of communication between MN and CN.

**1) Communication profile selection and Flow assignment**

When an ITS-S needs to communicate with a CN, first the application running on the sender device proceeds the flow assignment operation, as defined by the ISO 24102-6 standard. This process aims to select the most suitable communication profile for each flow, i.e., to select a collection of facilities protocols, transport protocols, network protocols, access technologies and communication channels that are used for a given data flow [16].

The flow assignment is depicted in Figure 2.5. In the communication profile selection process, the flow presents its communication requirements to the management plan, which is responsible to choose the communication profile for each flow. If the ITS-S has the necessary capabilities to support flows with these requirements, a flow type identifier (*FlowTypeID*) is attributed for each set of communication requirements and objectives. In this way, two flows with the same communication requirements can have the same *FlowTypeID*. This *FlowTypeID* may be well-known registered identifiers or may be dynamically assigned [18].

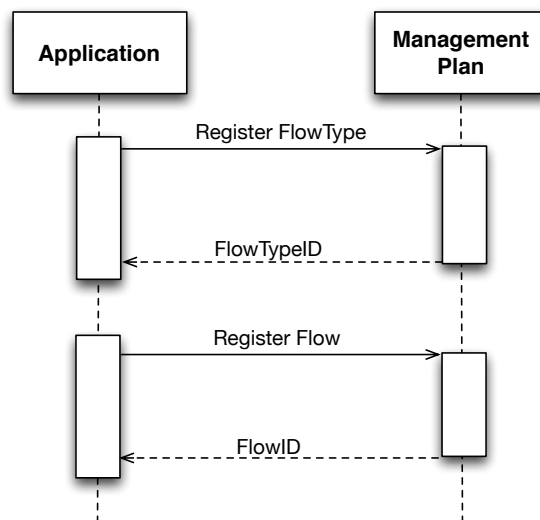


Figure 2.5: Flow assignment.

When the flow wants to send messages, it registers the destination with previously assigned *FlowTypeID* to the management plan. In this process, requirements and objectives presented by flow, user preferences, set of rules (e.g., regulations, network operator policies, etc.) and communication protocols' status are used by a DM in the management plan, in order to select the best suited communication profile per communication source. If this process fails to identify and select an appropriate communication profile (e.g., if no communication profile is currently available), the DM reports the status to application. In this case, either a best effort approach to enable communications or a communication refusal applies.

Once the selection process successfully completed, the DM replies an identifier of the mapped communication flow to the application (*FlowID*). Then, management plan shall notify the relevant ITS-S managed service entities from each layer. The *FlowID* points to

the information necessary for each layer to properly form its data for transmission.

### **Path selection**

Besides the selection of communication profile, it is necessary to select the most suitable path for routing a given flow. According to ISO 24102-6 [15], the determination of the path implies the selection of the communication interface, the logical node in the access network to which the ITS station is locally attached (ingress anchor node) and the node in the Internet which reaches destination node (egress anchor node).

Similarly to the communication profile selection, the management plan needs to receive some information to select a path. The DM needs to receive flows requirements, the capabilities of all layers (e.g., protocols supported, available communication interfaces, technologies and its functionalities, etc.), capabilities of the attached access network, capabilities of the CN (to know if CN is able to support the same communication profiles), and regulations and policies.

Several paths could be available, each one with different network conditions and performances. Then, the DM should map each flow to a path that best meets the communication requirements of each flow. Moreover, several flows with different priorities could take place (e.g., real-time media with high priority and mail services that have low priority). The DM should be capable to manage these variety of priority requirements. The methods to determine the most appropriate path and to perform flow-interface mapping is implementation specific. And it could be a competitive factor between stakeholders. In Chapter 5 we propose an algorithm to perform such mapping.

Once a path is selected, the management plan request the Networking & Transport layer to enforce them. In the same way, if a path is no longer available and must be removed or if a path must be updated, the Networking & Transport layer is requested to execute the appropriate operation. Similarly, whenever the Networking & Transport layer detects any change in a path status, it notify the management plan.

### **2) Send Application Data Unit**

To transmits messages (i.e., Application Data Unit (ADU)), the ITS application process send their ADU and the correspondent *FlowID* to the CPH. The CPH is an entity of ITS Facilities layer with management capability. The CPH checks whether there is a valid communication profile corresponding to the *FlowID* associated with the ADU. Then, it performs the appropriate actions to manage each ITS-S flow according to its specificities, e.g., send the ADU to the FSH entity if some service is needed (like encryption), otherwise, send it directly to the ITS-S network & transport layer.

### 3) Services Handler

The FSH is used to apply ITS-S facilities services to the ADU, as services related to security (encryption, privacy, authentication), time stamping, compression, etc. The list of services that must be applied onto the ADU is negotiated between the ITS application process and the management plan when the ITS flow is registered, i.e., when it receives the *FlowID*.

When an ADU is transmitted by an flow, the CPH sends this data unit to the FSH if at least one service is pointed by the *FlowID*. In this case, the FSH constructs the ITS-S Facility layer Protocol Data Unit (ITS-FPDU) by appending an header in front of the ADU. This header contains information about necessary operations to be performed on the ITS-FPDU in the reception side, e.g., decompressing when data is compressed or decrypting when the data is encrypted.

### 4) Publish/subscribe message

The CSH is used to share a specific message among multiple applications by means of the publish/subscribe process. Each application could subscribe to receive a specific type of message. For example, application could subscribes to receive positioning data from satellite networks such as Global Positioning System (GPS) or GALILEO. When a given message arrives, the CSH checks what are the subscribed applications for this message and delivery them to all subscribed ones.

### 5) Network & Transport management

The ITS-S network & transport layer presents a variety of protocols (e.g., Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Geo-Routing, IPv6 and their mobility extensions, and others). These protocols could be designed to meet specific flow requirements. However, to meet the needs of the majority application and services, IPv6 have been ideally suited. In this work we will consider this network protocol. The ISO 21210 [19] specifies networking protocol functionalities related to IPv6 networking in a global communication (Internet) between two or more ITS-S.

Whenever the ITS-S network & transport layer receives a data unit from Facilities layer, i.e., an ITS-FPDU, it should be forwarded to the appropriate communication interface. A module named “IPv6 forwarding” receives the data and sends them to “IPv6 LAN” or “External IPv6” interface modules, according to data destination and the policies sent by management plan. These modules provide mechanism for transmitting IPv6 packets to stations belonging the same local area network or stations outside of the local area network, respectively.

Similarly, the “IPv6 forwarding” module could receives IPv6 packets from its “External IPv6” interface or its “IPv6 LAN” interface. In this case, if packets are intended for itself, it

shall perform IPv6 address resolution and forward the packet to layer above. Otherwise, it shall perform IP next hop determination and forward the packet to the appropriate interface as specified in its forwarding table.

The ISO 21210 standard specifies two more modules: the "IPv6 security" and "IPv6 mobility management". The "IPv6 mobility management" module implements mobility support functions for Internet reachability and session continuity. While the "IPv6 security" module specifies security features and functions like encryption for IPv6 packets transmitted over the "External IPv6" interface, authentication for communication between stations deployed in different sub-systems, and location privacy to prevent an unauthorized entity to access the mobile location. IPsec is an example of used protocol to provide security support. But as mentioned before, in this work we do not enter in details about security.

### 6) Send/Receive packets

Once the Access layer receives a data unit, it instantiates a communication interface and forward the data over physical link. This process follow the management plan policies, i.e., taking into account the flow-interface mapping and selected path performed by the DM.

In the same way, packets received by interfaces are send to "IPv6 forwarding" in order to perform IPv6 address resolution or IP next hop determination, according to packet destination.

## 2.2. Enforcement and Mobility management adopted by ISO

Due the highly mobility of vehicles, network environment can change frequently. Vehicles can move at high speed causing networks to appear and disappear rapidly. In such an highly dynamic mobility environment, it is necessary a mobility management capable to offer seamless connection for communicating applications while vehicle is moving.

In order to enable the management of heterogeneous access technologies, as well as applications to be abstracted from communication mechanism, ISO has addressed decisions at network layer, using standardized protocols like Network Mobility Basic Support Protocol (NEMO) and Multiple Care of Addresses Registration (MCoA)/Flow Binding.

NEMO protocol is an extension of Mobile IPv6 and allows transparent session continuity for nodes in a mobile networks. NEMO mainly introduces the MR. The MR is in charge to manage the IP connectivity and mobility for all mobile networks devices attached to it. Then, even devices without capability for network mobility management (e.g., a vehicle's onboard sensor) can be

reachable in the Internet through the MR. Moreover, MR can take into account specific needs from each application and choose the interface that better match to the flow requirements. It enables devices be always best connected.

In the NEMO model, the mobile network gets a Mobile Network Prefix (MNP) depending on the Home Agent (HA). Based on this prefix, MR assigns unchangeable addresses to the Mobile Network Nodes (MNN). When a new network is available, the MR generates a new auto configured IP address within the new visited network and notifies them to HA. This address is called Care-of-address (CoA). Only MR and HA are aware of the network change, since MNNs continue connected with MR through their unchangeable IP address. Therefore, instead of manage several handovers (one for each mobile device), only the MR handover is performed.

MRs can be provided with multiple network interfaces. To manage these multi interfaces simultaneously, MCoA is used [20]. Figure 2.6 shows an example of communication using MCoA. The MCoA enables the registration of several CoA for a single HA. In this case, MR could establish multiple tunnels between their multiple interfaces and the HA. MR and HA exchange flow routing policies in order to determine which tunnel a given flow should use.

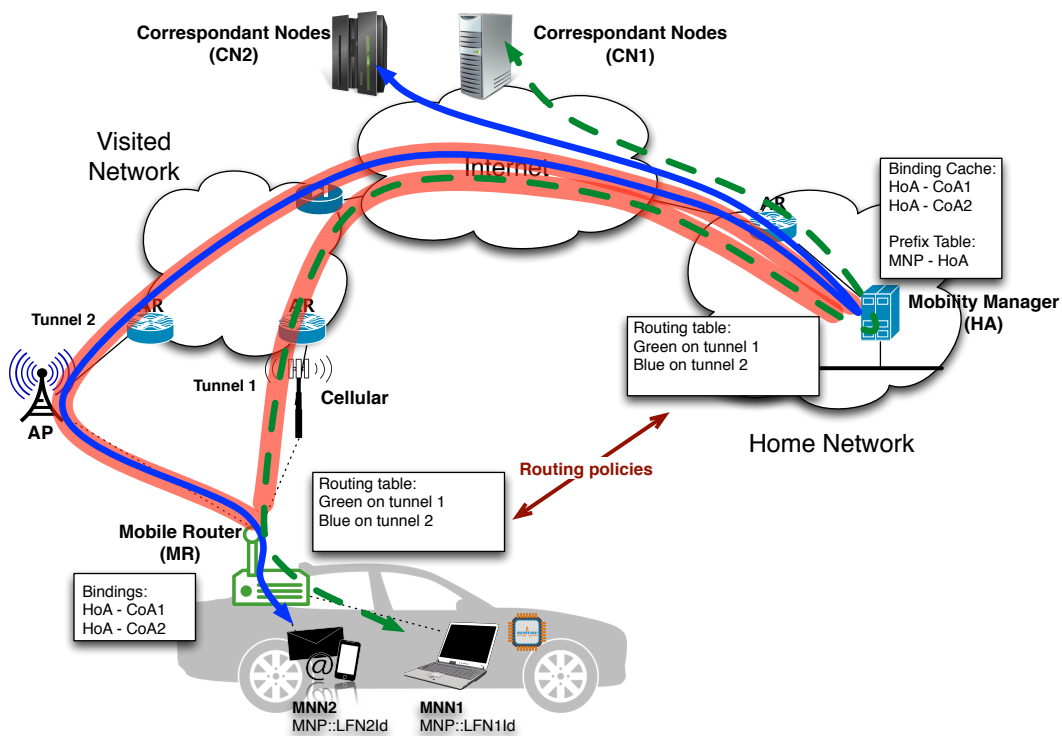


Figure 2.6: Example of NEMO + MCoA

Communication between any MNN and any correspondent nodes, i.e., nodes outside the MNNs network are performed through the HA. When a CN communicates with a MNN, it sends the packets towards the MNN home network. Then, HA redirects these packets to the current CoA

of MR, through the correspondent tunnel. Finally, MR distributes them to the MNN. In the same way, packets sent from MNN to a CN are routed by MR towards the HA through the tunnel. Then, HA forwards these packets to the CN.

### 2.3. Example of ISO ITS architecture implementation

Due to the modularity of the ISO ITS architecture, it can be implemented in different ways. The ISO ITS reference architecture can be implemented in a single Electronic Control Unit (ECU) or it can be split in distributed entities in the on-board network.

Figure 2.7 shows an implementation, in which all architecture modules are implemented in a single ECU. Such an approach can bring hardware and software complexity for the module. Communication management, facilities services (e.g., encryption/decryption, authentication), security services and applications all running in the same control unit are CPU and memory resources demanding for that module.

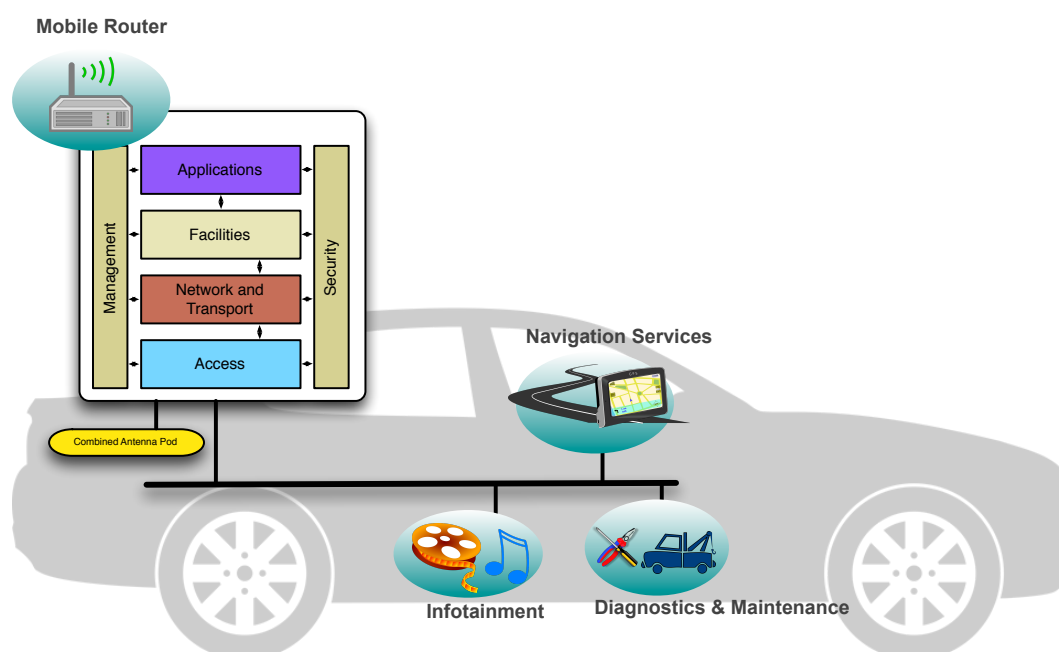


Figure 2.7: Example of ISO ITS architecture implementation in a single onboard device

Figure 2.8 shows an advanced implementation of the ISO ITS architecture in a vehicle. Such implementation is divide in various onboard devices.

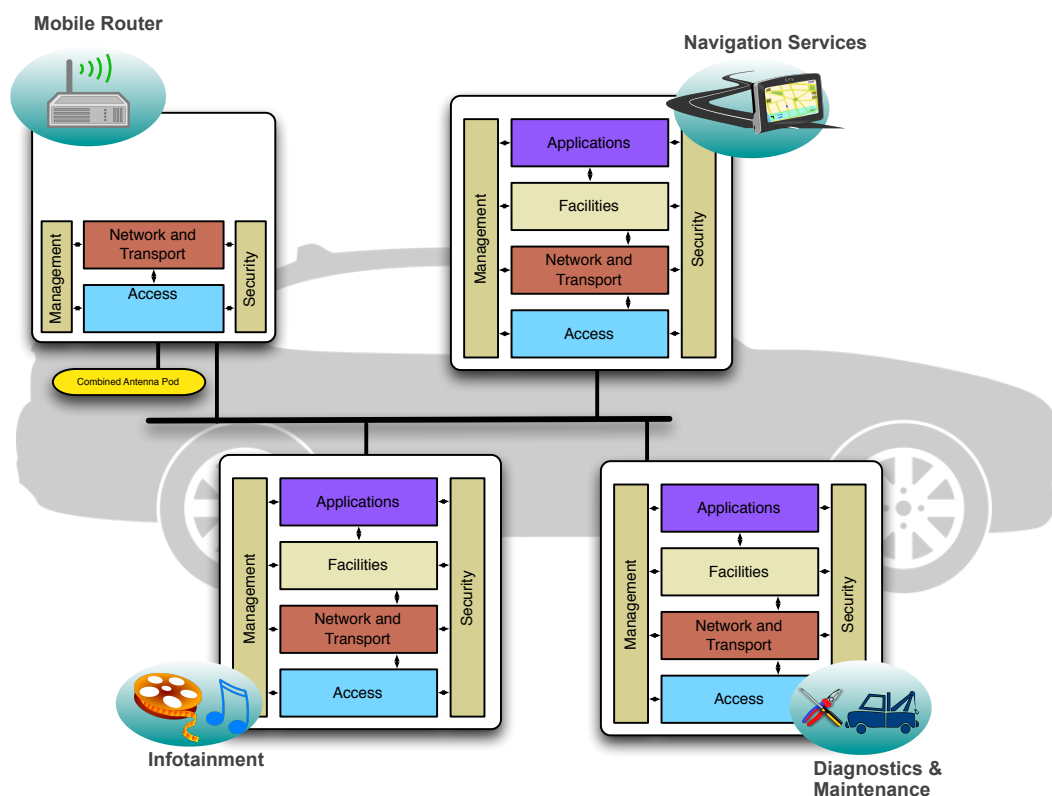


Figure 2.8: Example of ISO ITS architecture implementation in various onboard devices

As shown on Figure 2.8, the MR is designed in a dedicated module while applications are developed on other physical modules, both connected to the vehicular onboard network. The MR can be equipped with multiple communication interfaces and it is responsible to manage simultaneously all communication between onboard devices and outside devices.

Since all communication is handled by the MR, communication management like network selection and/or mobility management becomes transparent for onboard application. Therefore, other onboard devices like navigation system and backseat screen do not need to care about communication part, which are managed by the MR. Then their hardware and software can be simplified, e.g., they do not need antennas and software for communication management. Instead, they can concentrate in the application part.

Moreover, such an architecture allows onboard equipment without network mobility capabilities (e.g., in-vehicle sensor or camera) to be reachable from the outside through the MR.

## 2.4. Conclusion

Analyzing the standardization works we observe they are organized by layers. All architectures design a common management and security plan with cross-layer operation, while deal with



multiple interface management.

Among the ITS architectures, the one proposed by ISO is the most conceptually advanced. Such architecture is capable to manage multiple access technologies simultaneously, while the iteration with horizontal layers enables the management plan to acquire important information, in order to better manage the applications communication. A mobility management at network layer is designed, enabling seamless communication.

Due to the high velocity of vehicles, their connectivity context can change frequently, i.e., network availability and network conditions can vary rapidly. In such highly dynamic mobility scenarios, besides the current vehicular context, we consider that is needed to predict the network environment in which the vehicle will be soon inserted. Such near future network prediction enables the DM to anticipate decisions. However, such capability of taking into account short-term predictions about network environment is not yet integrated in the ITS-S communication architecture, and will require a proof of concept and performance study to be pushed in standardization.

The ISO ITS reference architecture designs functional modules in different layers. These functional modules are a kind of “black box”, i.e., an empty module, in which implementation are not specified. Standards give some guidelines, but developers have the freedom to develop their own mechanisms. Therefore, the implementation of such modules can be a competitive factor between stakeholders.

Among such “black box” we chose to work on the management plan, more precisely with a DM mechanism for management of flows communication. For that, we should to verify what is developed in terms of DM until now, i.e., we should to analyze the state of the art. Such analyze is described on the Chapter 3.

# 3

## Literature review

As observed on the Chapter 2, the standards give some guidelines to applications' developers. But they are not implementation guide with specificities for each implementation. The standards give some room for freedom in the way to implement. Thus, developers have the freedom to develop their own mechanisms, which can be a competitive factor between stakeholders, allowing a well-balanced market development.

We based our work on the ISO ITS architecture, due its capability to deal with heterogeneous access networks and multiple application communication, simultaneously. The architecture proposed by ISO designs functional modules, leaving some freedom for its implementation. Among these functionalities, the one we are convinced be able to provide added value is the decision making mechanism. We aim to design a DM mechanism capable to manage multiple flow communication simultaneously in a heterogeneous access network environment. The communication management described by the ISO ITS standards is composed by two main steps:

1. **Communication profile management** – this step is responsible to verify if the current ITS-S, i.e., the one that initiates the communication process, and the CN are capable to support the flow requirements. The management plan interrogates each layer to know if necessary protocol stacks are available. Moreover, the management plan acquires information about the CN capabilities, i.e., it certifies that the CN has all needed protocol stack and if it is capable to receive and process the given flow. We do not consider the CN capabilities in our

---

work, we just assume legacy IPv6 node running somewhere on the Internet;

2. **Networking management** – once the communication profile is verified and a *FlowTypeID* is assigned for each type of flow, the networking process starts. This process aims to find, among all available access networks, the one that better matches with communication requirements. Such process is responsible to find the access technology that better satisfy each flow requirements, users' preferences, administrators and regulators rules.

Despite the communication management is main conducted by the management plan, other actors from different layers are involved in the decision making process. These actors can have informative role, e.g., informing the DM about a current situation (protocols availability, access networks availability, and network conditions); or they can participate actively in the decision process, i.e., sending their requirements, preferences and rules for the DM.

Based on the ISO ITS architecture, we identified the main actors that should participate in the decision making process. Figure 3.1 shows an abstraction of the ISO ITS architecture, in order to highlight such actors and their interactions, which are briefly described below:

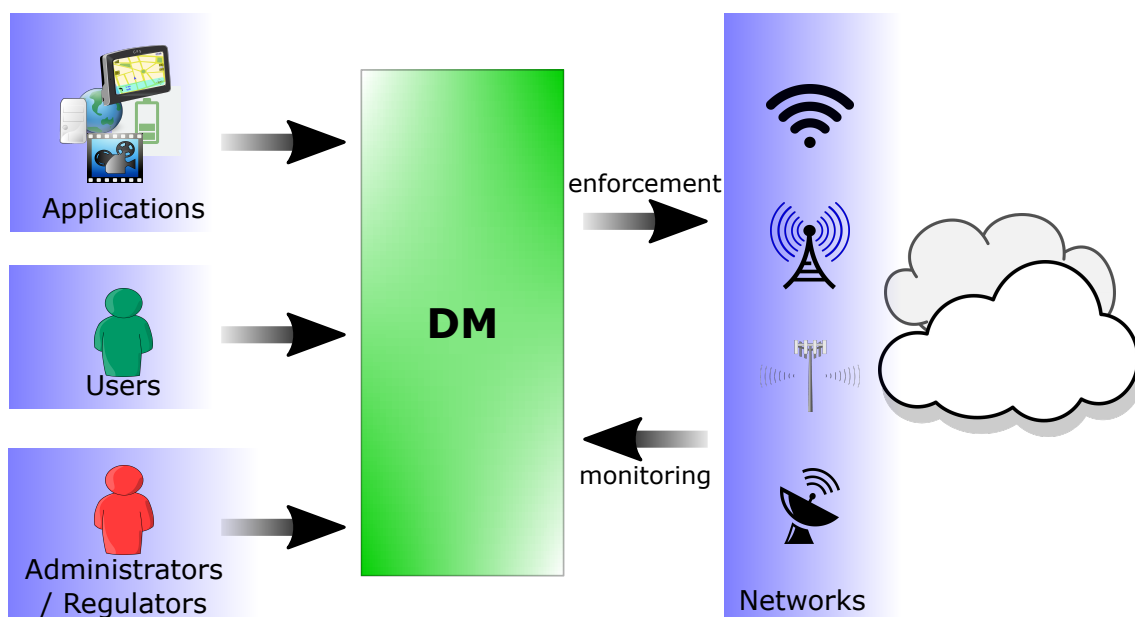


Figure 3.1: Decision Making process

- **DM** – the DM is the entity responsible to acquire information, rules and preferences in order to make decisions, i.e., to choose the access network that better meets each communication requirement;
- **Applications** – such entity represents the set of flows that send their communication requirements for the DM. The DM communicates with applications in order to inform them

---

about the decision status, e.g., to inform if communication is not possible due to absence of feasible access networks or to inform applications about the prediction of a better access network coming on the road;

- **Administrators and regulators** – network administrators as well as regulator entities can present their rules for the DM. Countries can have specific rules for some kind of flow communication. Network operator can have specific network constraints and particular billing procedures. Therefore, the DM should be capable to take all these rules into account;
- **Users** – in the decision making process, users can present their preferences, e.g., defining the amount of money he/she is willing to pay for a given type of flow, or defining priorities for different types of flows;
- **Access networks** – the DM should interact with the access networks for at least two reasons. First, it needs to know access networks availability and its performance. Second, the DM should enforce the decision, i.e., it should forward each flow through the chose access network.

Despite being a well-studied area, the decision making process should be improved to meet C-ITS needs. Research works still working in DM architectures and algorithms to improve network selection process and handover management.

Therefore, we need to verify in the literature what have been done in terms of DM mechanisms. The objective of this chapter is to identify how works related with networking management have addressed the different actors identified on Figure 3.1. We aim to identify which actors are developed, what kind of information are exchanged between them, and how the DM takes into account such information, i.e., which DM algorithms have been developed, as well as, which attributes DM algorithms have used to make decisions.

The following sections present some of the most significant work related to decision making process. We consider that a DM mechanism should be a complete solution for the decision making process inside an ITS-S, i.e., it should be composed by an architecture describing all modules and their interaction; and by a DM algorithm responsible for the network selection, and solution enforcement processes. In this way, we divided the literature review in three phases. The first phase, described on Section 3.1, is more related with DM architecture. Such phase focuses in how works have addressed the different actors identified on Figure 3.1, i.e., if they have developed all actors or only some of them. Section 3.2 describes the second phase, which is related with DM algorithms. Such second phase identifies the main algorithms that have been used in decision making process. The third and last phase is described on Section 3.3. Such phase is a continuation of the previous one, in which we identify the most frequent attributes used by the DM algorithms in the decision making process. Finally, Section 3.4 concludes the chapter.

### 3.1. Research works related with DM architecture

Analyzing the literature, we observe that research works have been interested by different parts in the decision making process. Some recent works have been carried out in an ISO/ETSI standard compliant way, while others have developed their own mechanisms regardless of current standardization efforts. However, even the ones not developed in a standardization compliant way can be compared with our abstraction of the ISO/ETSI ITS architecture showed on Figure 3.1.

In this section, we verify in which part of the ITS architecture researches have concentrate their efforts. For that, we present research works related with each actor identified on Figure 3.1.

#### 3.1.1. Decision Making (DM)

The DM module represents the network selection process. DM takes into account application's requirements, user's profiles, administrative rules (regulation and policies) as well as information from access networks and ongoing flows in order to manage flows communication. The management of flows communication is performed to achieve Always Best Connected (ABC) [21] concept, i.e., to choose always the best available access network for each flow. The "best" is a subjective judgment and very dependent of the flow type and the context in which the vehicle is inserted. For example, for some types of flow the best access network can be the one with more available bandwidth, while for others the best can be the more secure access network.

Whenever a decision is performed by DM, it enforces them in the respective managed entity. For example, it can prioritize a set of flows, select new access networks, or activate/deactivate network interfaces.

A large number of research works have been concentrate in development of DM modules, more precisely in the development of decision making algorithms to network selection. According to authors of paper [22] the decision making process can be classified in three categories: network-centric, terminal-centric or collaborative. In Figure 3.2 we give an overview of such definition. And each one of such three categories are described below.

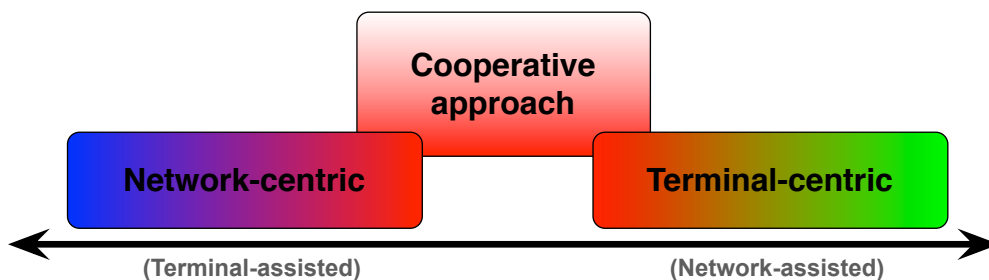


Figure 3.2: Decision Making classification based on decision location

- **network-centric** – in the network-centric approach, decisions are made at network side with or without assistance from the user terminal. Usually such decisions are controlled by the network operators. Core networks are supposed to have the knowledge of all access network, then a network monitoring process can be easily implemented, while decisions can be performed in order to guarantee efficient use of the infrastructure resources.

User terminals can assist the network, sending their acquired information. However, in such an approach the management of individual flow requirements and user preferences can be very difficult. Even harder if we consider dynamic requirements and preferences, i.e., if we consider that flows and users can constantly change their requirements and preferences, respectively.

Since the DM in the network side should have updated knowledge of the different actors (e.g., access networks, applications requirements, and users preferences), network-centric approach overloads the network and suppose constant connection between terminals and the network. In case of connection loss, it is necessary to adopt a palliative solution, e.g., leave the terminal to perform its own decision or adopt predefined decision making rules.

Another challenge in the network-centric decisions is the interoperability between networks managed by different operators. Routing flows between networks from different operators implies a tight collaboration agreements. Therefore, in such an approach, usually flows are restricted to only one network operator, i.e., they are routed through different access networks belonging to the same operator.

An example of network-centric decision is the cellular networks that manage the mobility of user devices while they move around. In the domain of cellular networks, some efforts have been done in order to facilitate the cooperation between different Mobile Network Operator (MNO). An example of this cooperation is the European project *5G Cross Border Control (5G Croco)* [23]. The main goal of this project is to test 5G technologies in the cross-border corridor along France, Germany and Luxembourg, enabling cooperation between some MNOs from these countries (i.e., Orange in France, POST in Luxembourg and DTAG in Germany).

- **terminal-centric** – in the terminal-centric approach decisions are made at user terminal. In this approach, no changes are required in the network infrastructure. The terminal is supposed to sense its environment in order to get information about available access networks and its performances. Since applications are running on it, the terminal is capable to take into account each flow requirement. In the same way, as the terminal is close to the end user the DM running on it can consider users preferences.

Unlike the network-centric, the terminal-centric approach can easily manage the access

network heterogeneity. Once the mobile terminal has the capability to connect with multiple access technologies, the DM can take advantages of available access networks, independently of the network operator.

Despite this flexibility to route flows through different networks, a terminal-centric decision does not guarantee efficient use of infrastructure resources. It has the knowledge of network conditions that is limited by what it observes. Consequently, such local perception of network condition prevents event anticipation. In some cases, the network can assist terminal decisions, by enriching it with its global perception of the network. However, such an assistance is limited to networks that support this.

- **collaborative** – in the collaborative approach both network and terminal participate in the decision making process. An example of the collaborative approach is the standard IEEE 802.21 Media Independent Handover (MIH), where mobile device and specific network entities exchange information in order to assist the handover between heterogeneous networks.

Collaborative approach assumes that both terminal and network support specific functionalities needed for cooperation between them. For example, network can provide information about all access network conditions, while terminal can manage users and applications requirements.

However, a DM designed strictly to work in a collaborative approach way can face limitations when the mobile terminal or network is not able to provide the information expected by the DM.

In the present thesis, we aim a DM mechanism capable to make opportunistic networking in a heterogeneous network environment. Therefore, since in such a heterogeneous network is not possible to assume that all networks are able to cooperate, we consider terminal-centric approach with network assistance when possible, i.e., when the network support cooperation.

Some studies on decision making process for network selection consider only one interface at a time [24, 25, 26, 27]. In this way, DM algorithms select only one access network at a time, i.e., the network that meets the requirements of most flows, and all other flows should communicate through such access network. In such an approach, some flows can have their requirements partially attended to the benefit of other flows. An example of commercial DM that follows such approach is the one used in most of smartphones. In this case, the DM chooses the access network to be used (e.g., urban WiFi or celular (3G/4G)), and all ascending and descending flows should communicate through such access network.

However, the number of devices capable to connect with heterogeneous networks simultaneously is increasing. This growth have motivated DM developers to explore such devices capability. As

results, some works have considered the use of multiple interface simultaneously and routing flow per flow, i.e., spreading flows among different network interfaces [28, 29, 30].

Such a works usually propose their own DM architecture, designed for specific uses cases. This is the case of the paper [30] that proposes an architecture composed by a shim layer between Network and MAC layers, which is responsible to receive and classify data packets from upper layers. Such DM deal only with ascending flows, i.e., flows leaving the user terminal. Packets are classified in five predefined categories based on their communication requirements and each category is assigned to a specific queue. The five categories of queue compete for the available access networks. Once the DM chooses access networks for the queues, packets waiting in such queues are transmitted.

Several works have developed DM for handover management [26, 27, 31, 32, 33]. In cellular telecommunication, the term handover is frequently used to describe the process of migrating communicating flows from one cell to another one, i.e., from one access point (antenna) to another one. This process is most commonly triggered when the mobile device moves out of the coverage area of a current cell. In this case, all ongoing flows should be transferred to the new cell.

In the present work, we are interested in map flows through access networks that better satisfy communication requirements. This process does not exactly match the notion of handover just described before. We will not necessarily transfer all flows through a new access network, instead we migrates only flows whose requirements are not 100% met and on the condition that a best access network is found.

Recent works that deal with DM module are usually interested only in the development of decision making algorithms to network selection, regardless to any architecture. Therefore, these papers using different techniques of DM algorithms are described on section 3.2.

#### 3.1.2. Requirements gathering

In the requirements gathering process, works have considered different actors presenting their requirements, preferences, constraints and policies to the DM, as shown on Figure 3.3. Flows from different applications can have specific requirements, e.g., requesting specific bandwidth, latency, and security level [34, 24]. Users can present their preferences, e.g., defining priority or security level for a given message [35]. Network administrators and regulators can present their policies, as network constraints and particular billing procedures [32].



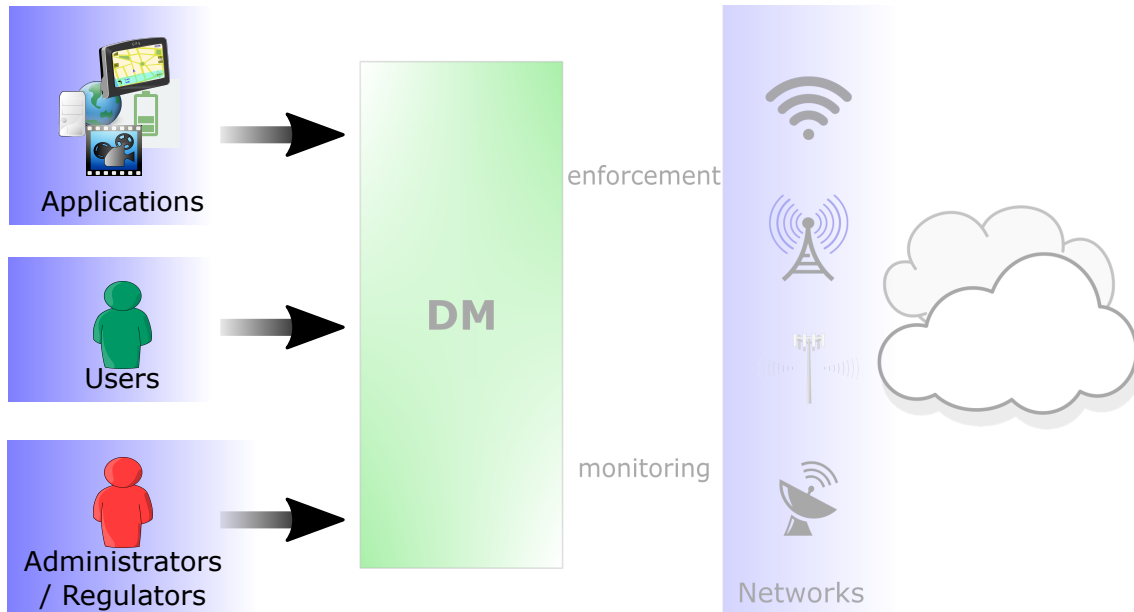


Figure 3.3: Requirements gathering.

The ISO ITS architecture defines interfaces between the different architecture layers, which enable the exchange of information between actors and the DM. Despite the standardization of such interfaces, the way how actors acquire requirements, preferences and rules are implementation dependent. DM developers can define how users should present their preferences (e.g., by using a configuration file, or a HMI), they can decide if such an information should be static, i.e., assigned once in the system, or if information can be assigned dynamically, varying with the context in which the terminal is inserted.

Few works deal with how these parameters are acquired. In paper [34] authors propose a module responsible for collecting and combining stakeholders policies. In such module, network operators, administrators and users can present their communication requirements as high level policies, i.e., in a human language way. An intermediary module is responsible of converting these high level policies into system level rules and send it to the DM. Authors of paper [36] propose a middleware which deals with profile management. Such profile management process has been patented [37]. They consider profiles as being files stored in a Profile Databases (PDB) that summarizes key information like users' preferences, administrators' rules, terminal resources, networks' conditions and applications' needs. Authors propose three kinds of profiles within the PDB: *generic* that describes a pattern of profiles (e.g., application class and forbidden access network); *specific* that are specific profiles corresponding to specific cases (e.g., minimum/maximum throughput and preferred network); since *specific* profiles from different applications can be contradictory, authors created the third profile named *active*. The *active* profile is obtained by filtering different specific profiles of the same type.

The majority of works do not care about the way requirements are acquired. They usually

assume that all necessary information is previously available, i.e., they have flow requirements, network information, user preferences, etc., without describing how these parameters are acquired. Such a works assume that these parameters are acquired either dynamically from some non detailed monitoring modules or assigned statically, i. e., parameters are configured previously in the decision making process [24, 38, 39, 40, 41].

#### 3.1.3. Monitoring process

Besides application requirements, user's preferences and regulators policies, as described on *Requirements gathering* part, the DM can be fed with other information like available access networks, networks performance, the context in which the DM is inserted, as well as updated information about communicating flows, e.g., currently used bandwidth or currently latency [42, 26, 43]. In this case, monitoring modules are responsible to acquire and provide such an information, which enable the DM to perform fine-grained decisions.

Despite these wide variety of useful information, most works limit to monitor information related with the access networks, i.e., discovering and evaluating the performance of the networks [44, 45, 46]. Analyzing these works, we observe they address the network monitoring process in different ways. In the follow, we list some of such approaches:

- In the literature, we find papers that do not develop the monitoring module. Despite they emphasize the need to monitor some network parameters (like DR and latency), such a papers usually perform simulation based on network parameters that are manually assigned [24, 30, 29];
- Some works consider simple monitoring process that evaluates only the Received Signal Strength Indication (RSSI) of access networks. In this case, the monitoring module measures the signal level in each active network interface. If the RSSI is above a threshold, the access network is considered available, and consequently considered in the decision process [27, 47, 42]. This kind of monitoring module can lead to a misinterpretation of the real network condition. A network can have a good RSSI level, but with low or no available bandwidth;
- Other works have proposed their own network monitoring modules. In the follow, we highlights two of them: First, paper [44] that proposes a monitoring module named "User and Context Management Module" (UCMM). The UCMM is responsible to gather information originated from various sources like network interface cards, Quality of Service (QoS)/QoE measurement tools, and bandwidth estimation tools. Such information are shared with the applications, which are responsible to make their own decision. In this paper approach, the

application developer should take care about the communication part. Second, paper [45] proposes a monitoring module called *scout*. This module is responsible for discovering and evaluating the performance of available access networks at any given time. It periodically attempts to establish network connections, and measures the throughput and latency of the connection by sending periodic beacon message. Besides being power consuming, such an approach can overloads the network.

- A large number of works consider cooperation with networks, usually based on the standard IEEE 802.21 [35, 28, 25, 26, 32, 31]. MIH is a framework aimed at assisting the handover between heterogeneous networks. As the name indicates (Media Independent Handover), it makes the communication of MIH users independent from the media (i.e., independent from the access technologies). This standard defines the Media Independent Handover Function (MIHF), a logical entity located above the media-dependent interfaces (link-layer), which provides a single media independent interface to upper-layers (e.g. mobility protocols). The MIHF should be developed in all communicating sides, i.e., in the user devices and in the network side. Such logical entities are responsible to detect changes in the link layer and report appropriate events from both local and remote interfaces. Moreover, it should provide a set of commands for both local and remote MIH users to control link state. Therefore, DM using such an approach should assume that all communicating devices and available networks support the specific functionalities defined by MIHF.

#### 3.1.4. Applying decision

In the applying decision process the policies and information produced by the decision maker are applied in the system. Decisions can be applied at different layers of computing system, i.e., it can be applied at application, transport, network or link layer. However, each of such layers presents specificities, in terms of level of technology dependency, knowledge of application requirements and time to adapt to a new decision policy, as shown on Figure 3.4.

Figure 3.4 shows examples of knowledge and possible actions in each layer. In the center of the figure, we exhibit the different layers and two arrows showing the technology dependence and adaptation time, i.e., the time necessary to effectively enforce a decision. As we can see, the technology dependence increases from application layer to the link layer, while the adaptation time decreases. For example, a decision that takes place at the “Access” layer (L2 from Open Systems Interconnection (OSI) model) is independent from upper layers and can be enforced rapidly. However, it is very dependent of the network interfaces and the access technologies. While a decision in the “Application” layer is less technological dependent but take longer to be applied, due the different processes applied by protocols in each layer.

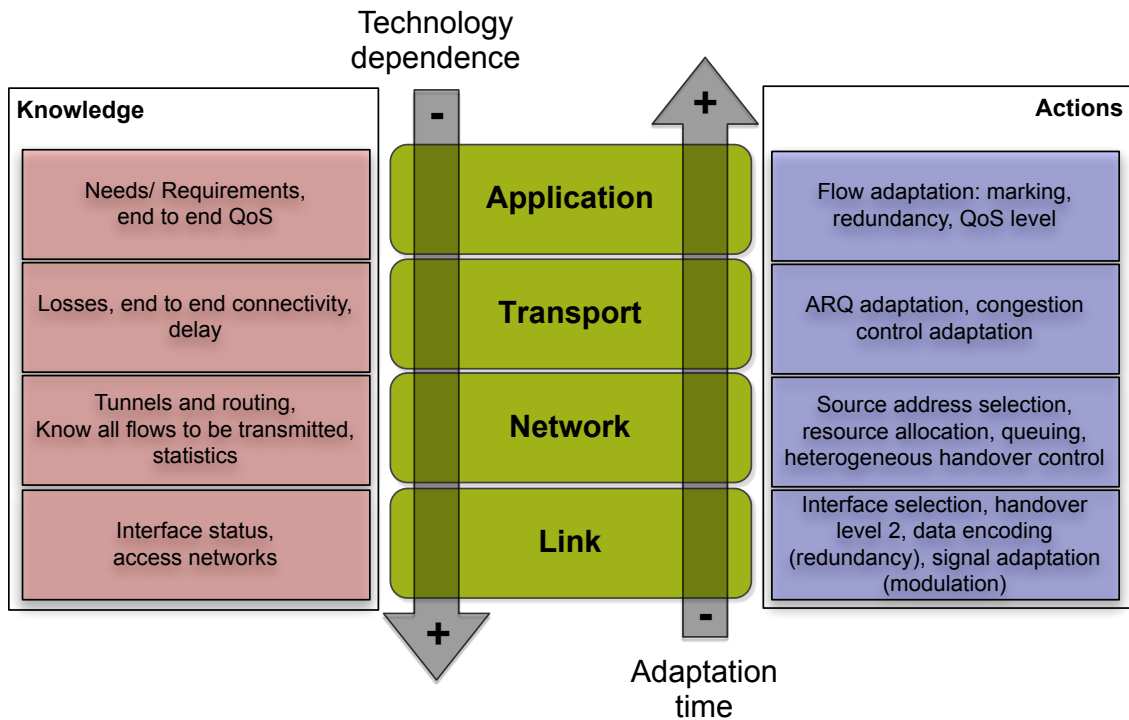


Figure 3.4: Decisions at different layers.

The left rectangle titled “knowledge”, in the Figure 3.4, list some information acquired by the different layers, while the right rectangle titled “actions” shows some possible actions enforced by the layers. Analyzing each layer individually, we observe that some information are more directly accessed by one layer than by others, enabling different enforcement, for example:

- **Link** – since the link layer has information about access networks (e.g., RSSI) and interfaces status (e.g., it knows if the interfaces are enabled or disabled), it can perform actions like interface selection, management of handovers between network interfaces, signal modulation, and data encoding (e.g., adding redundancy to a source in order to make it robust during transmission over a noisy channel);
- **Network** – thanks to its tunneling and IP routing capabilities, network layer can perform heterogeneous handover control. Such layer knows the flows being transmitted and the ones to be send, enabling better network resources management;
- **Transport** – transport layer provides end to end connectivity, i.e., it provides logical communication between application running on different hosts. Therefore, it has information about communication latency and messages losses, enabling congestion control adaptation. Such layer is also responsible to provide quality and reliability to the end user. For that, it implements protocols like Automatic Repeat reQuest (ARQ) to make more reliable data transfers;

- Application – application layer knows each flow requirement and user needs in terms of end to end QoS. Therefore, such layer can perform actions in order to increase QoS level, e.g., adapting flows according to real communication conditions.

Since each layer have direct access for different information, we can see the importance of an cross layer entity capable to acquire and consolidate specific information from such different layers. Such a cross layer enables a better knowledge of all layers and faster enforcement of decisions.

The following subsections, give more details about the enforcement at different layers and how works have addressed the enforcement of decision making process in such layers.

#### 3.1.4.1. Application layer

Since “Application” layer has full knowledge about requirements of running applications, it is prone to make optimizations that improve application’s QoS [48]. Applications can make fine-grained decisions that are aligned with their specific needs. Moreover, it can interact with other actors, like users and network operators, in order to take theirs preferences and policies into account.

As evoked by [49], most videoconference applications have end-to-end adaptation mechanisms based on the Round-Trip Time (RTT) and packet loss. Therefore, the DM can inform the application as soon as a significant event occurs, leaving applications to adapt their video quality.

However, decisions at “Application” layer present some issues. Due the underlying processes of lower layers, these decisions are supposed to take more time to be enforced. Moreover, in this case applications should be aware of communication specificities like supported protocols of underlying layers, available access networks and its characteristics. Therefore, such embedded functions can increase applications complexity and can also difficult the interoperability with existing applications.

#### 3.1.4.2. Transport layer

Some works address its decision at “Transport” layer. Using protocols like TCP and Stream Control Transmission Protocol (SCTP), they create multiple transport layer connections to transmit data over heterogeneous channels [50, 51]. The SCTP protocol allows the transport layer to associate a port to multiple IP addresses, enabling an application be joined by multiple IP addresses. Such an approach enables some useful functionalities. Applications can send the same message through two different network interfaces, in order to increase the communication reliability. Or, applications can send different messages through multiple network interfaces to increase the communication data rate.

Multipath TCP Protocol (MPTCP) is another implementation at the Transport layer. It establishes multiple TCP connections, one per available link/IP address and spread the traffic over them. The objective is to maximize resource usage and increase redundancy. However, such an operation mode requires pre-configuration and does not allow flexibility of switching between the established TCP connections. MPTCP should still be improved to manage interoperability between applications [52, 53].

Moreover, communication management at transport layer tend to take longer than communication managed in the lower layers. This is due to different factors as the need to unpack the corresponding header in each layer, as well as some internal transport layer process like flow control and adaptive compression.

Transport layer is responsible to ensure end-to-end communication, providing quality and reliability to the end user. Therefore, some transport layer protocols like TCP use flow control mechanism, which ensures that a sender is not overloading a receiver by sending more packets than it can consume. The main objective of the flow control is to avoid data congestion.

Such a protocols use flow control algorithms like the “stop and wait” and “sliding window”, which increase significantly the data transmission time. The “stop and wait” algorithm forces the sender to stop and wait until the acknowledgement of the data sent is received; while in the “sliding window” algorithm (e.g., the one used by the TCP protocol) both sender and receiver agree on the number of data to be transmitted before wait for the acknowledgement. Such a flow control algorithms contribute to increase the adaptation time.

In order to better adapt data transmission, transport layer implements adaptive data compression. In such process, a compression mechanism compress data according to the network conditions. For example, if networks do not have enough available bandwidth, the mechanism compress data to reduce transmitted bytes over network links. Otherwise, data are transmitted without compression. Once again, such a process contributes to increase the adaptation time.

#### 3.1.4.3. Network layer

Decisions at “Network” layer are very studied in the literature. This is the lowest layer that deals with end-to-end transmission. It enables the distribution of data packet over multiple network interfaces by using standardized protocols like IP. Due to the IP flexibility, heterogeneous access networks can be achieved across different domains and infrastructure. Therefore, it is possible to manage network traffic over multiple sites and through the internet [54, 25, 55, 32, 56].

As seen before, some transport layer protocols (e.g., SCTP, and MPTCP) enable communication through different access networks simultaneously. However, the migration of a flow from an access network to another (e.g., in case where the targeted network is more performer than the

current one) implies connection interruption.

Unlike the transport layer, in the network layer messages can be seamless transferred from one network interface to another. Thanks to the communication flexibility provided by standardized network protocols like Mobile IP (MIP) and NEMO, flows of data can be seamless switched among heterogeneous access technologies. Some works have addressed their decisions at network layer, considering such IP-session continuity protocols [28, 33].

#### 3.1.4.4. Link layer

Since decisions at data link layer are independent from upper layers, they can be applied very fast. Moreover, the knowledge about access networks and interfaces status enables a set of actions like performing handover, splitting data over multiple channels for better use of network capacity, as well as performing signal adaptation, i.e., adapting signal and protocol parameters according to the radio conditions [57, 58, 59].

However, decisions applied at data link layer are very dependent on the technology. Link layer protocols operate only on the link that the network interface is physically connected to. Therefore, such technological dependency inhibits interoperability between heterogeneous access technologies. Decisions applied at data link layer are limited to tight-coupled networks, usually networks belonging to the same operator.

#### 3.1.5. How works have addressed the DM architecture

As observed on the previous sections, works have addressed different parts of the ITS-S architecture. Some works have concentrated on the DM module and its interactions with the network monitoring modules, while other works consider interactions with users, applications, and/or network administrators.

In order to see how works have addressed the different actors identified on Figure 3.1, i.e., in which part of the ITS architecture they have put their efforts, we analyzed the main research works related with the DM architecture. The results of such analysis is showed on Table 3.1.

### 3.1. RESEARCH WORKS RELATED WITH DM ARCHITECTURE

Table 3.1: Main research works related with DM architecture

N	DM	APP	USER	ADM	Context	Network monitoring	Monitoring solution	Paper
1	v	v	v	v	v	v		[34]
2	v	v	x	x	v	v		[24]
3	v	v	v	x	x	v	MIH	[35]
4	v	v	v	x	x	v	MIH	[28]
5	v	v	x	x	x	x		[40]
6	v	x	x	x	x	v	MIH	[25]
7	v	v	x	x	v	x	Fixed values	[42]
8	v	v	x	x	x	v	MIH	[55]
9	v	v	v	x	v	v	MIH	[26]
10	v	v	v	v	x	v	MIH	[32]
11	v	v	v	x	x	v	MIH	[31]
12	v	v	x	x	x	v	Scout	[45]
13	v	v	x	x	x	x		[41]
14	v	x	x	x	v	v	MIH	[33]
15	v	v	x	v	x	v		[29]
16	v	x	x	x	x	v		[56]
17	v	v	x	x	x	v	UCMM	[44]
18	v	v	x	x	v	v		[27]
19	v	v	v	v	v	v		[54]
20	v	v	v	x	v	x		[30]
21	v	v	v	x	v	x		[39]
22	v	v	v	v	x	x		[38]
23	v	x	x	x	x	v		[60]
24	v	v	v	x	x	v		[61]
25	v	v	x	x	x	v		[62]
26	v	v	x	x	x	v		[63]
27	v	v	x	x	x	v		[64]
28	v	x	x	x	x	v		[65]
29	v	v	x	x	x	v		[66]
30	v	v	x	x	v	v		[67]
31	v	v	v	x	x	v		[68]
32	v	v	v	x	x	v		[69]



Table 3.1 is organized as follow:

- N – the first column N is a sequential number showing that we listed 32 works in this table. The objective is not to have an exhaustive list of all analyzed papers, but a sample how works efforts have been distributed;
- DM – it represents the management plan responsible to make decisions based on information received from different actors;
- APP, USER, and ADM – represent the actors previously described on Figure 3.1, i.e., Application, Users, and Administrators, respectively;
- Context – this column represent the contextual information, i.e., it shows if the decision making solution proposed by the paper takes into account context information like speed, monetary cost or battery charge;
- Network monitoring – it indicates if the DM proposed by the paper takes into account information from the network monitoring modules. It does not means that the paper develops a monitoring module. In some cases, the paper uses predefined values for simulations without take care how such values are acquired;
- Monitoring solution – it shows what monitoring solution was used by the paper. In most cases the monitoring solution is not mentioned.
- Paper – indicates the paper reference.

Each row of Table 3.1 shows a paper. We mark in green ( **V** ) the cell corresponding to a part addressed by this paper, and we mark in red ( **X** ) the part not considered by the paper. For example, the last row ( $N = 32$ ) shows that paper [69] addresses the DM, Application, Users and Network monitoring parts, while the Administrators and Context information are not addressed.

By analyzing Table 3.1 , we can observe that most papers consider network information in their decision making process. Some of them take into account applications requirements. However, the majority of papers do not consider users preferences, administrator rules or context information. Moreover, most papers use MIH as monitoring network solution.

## 3.2. Research works related with DM algorithms

Similarly to section 3.1, in which we analyzed the ITS architecture parts addressed by research works, in this section we analyze which kind of DM algorithms have been developed by these works.

Different techniques of decision are used in the network selection process. They aim to combine requirements, preferences and policies in the decision making process[70]. Among these techniques we can highlight the ones based on fuzzy logic [42, 32], the ones based on the game theory [29, 71], the ones based on Multi-Objective Optimization (MOO) [49] and the algorithm that use Multi-Attribute Decision Making (MADM) techniques [24, 26, 47, 33, 72].

Despite the wide variety of DM techniques, we observe that algorithms based on the MADM method still being the most used in the literature. Therefore, in the following we review such category of DM algorithm.

### 3.2.1. MADM

The MADM problem is formulated as a matrix (named decision matrix), in which the lines correspond to potential solutions (alternatives) and the columns correspond to the set of attributes describing such alternatives. Figure 3.5 shows a template of such a decision matrix composed by  $m$  potential solutions,  $S_i(i = 1, 2, \dots, m)$ , and  $n$  attributes ( $A_1, A_2, \dots, A_n$ ). Each potential solution is described in terms of its attributes ( $a_{mn}$ ).

		Attributes		
		$A_1$	$A_2$	$A_n$
Potential Solutions	$S_1$	$a_{11}$	$a_{12}$	$a_{1n}$
	$S_2$	$a_{21}$	$a_{22}$	$a_{2n}$
	$S_m$	$a_{m1}$	$a_{m2}$	$a_{mn}$

Figure 3.5: Decision matrix template

Therefore, such potential solutions are analyzed and compared to each other, in order to come out with a single optimal alternative. For that purpose, the attributes describing each alternative are normalized and combined in an objective function, while a weight vector is defined in order to express the relative importance for each of these attributes.

In the network selection process, the lines of the MADM decision matrix correspond to available access networks, while the columns correspond to the set of attributes describing such access networks, e.g., DR, latency, and PDR.

Different methods may be applied to combine these normalized attributes [73, 74]. Among such MADM methods, we can highlight the Simple Additive Weighting (SAW) algorithm [67, 39],

Multiplicative Exponential Weighting (MEW) [68], Weighted Product Method (WPM) [61], and TOPSIS [26, 66, 64, 60].

In the follow, we explain two of these MADM methods, i.e., MADM SAW that is the simplest MADM method, and MADM TOPSIS, which is the most used in the literature.

#### 3.2.1.1. MADM SAW

SAW is the simplest MADM method. The basic principle of SAW is to obtain a weighted sum of the performance ratings of each alternative under all attributes. Suppose we have a set of  $m$  potential solutions,  $S_i$  ( $i = 1, 2, \dots, m$ ), and  $n$  attributes ( $A_1, A_2, A_3, \dots, A_n$ ) that describe the utility of each solution. Furthermore, we assume that the weights of attributes supplied by decision makers are represented by a weighting vector  $W_j = (W_1, W_2, W_3, \dots, W_n)$  where  $\sum_{j=1}^n W_j = 1$ . The best solution is considered as being the one with the higher score  $V(S)$ , such that the score of each potential solution  $V(S_i)$  is described by Equation 3.1.

$$V(S_i) = \sum_{j=1}^n W_j \cdot A_j \quad (3.1)$$

Some research works have used SAW algorithms in the decision making process. In [67], authors develop a fuzzy MADM methodology to combine application QoS requirements with context components (e.g., monetary cost or network power consumption), in order to make context-aware network selection for each application. First they use Fuzzy Logic Controllers (FLC) that considers network parameters and application QoS requirements to determine the QoS suitability level of each network. Then, they use MADM SAW algorithm to combine the previous calculated QoS suitability level with context components. The network alternatives are ranked by their context suitability level. Finally, they choose the best network for each application, i.e., the network that maximizes the context suitability level. The entire process is repeated if a new access network becomes available or if current access networks change their attributes.

Paper [39] describes each criteria in terms of utility functions. In this way, it normalizes different parameters (usually represented by different units) to a comparable numerical representation. Like paper [67], this paper considers predefined weights for each criteria. First, it creates the set of potential solutions by analyzing the application requirements, i.e., if an access network cannot guarantee the requirements specified by an application, such access network is not considered in the decision matrix. Since the decision matrix is created, SAW algorithm is applied. The entire process (i.e., generation of the decision matrix and SAW algorithm execution) is performed for each application and repeated every time a change happens in the decision making process, i.e., application requirements and/or network conditions changes.

#### 3.2.1.2. MADM TOPSIS

Proposed in 1981 by Hwang and Yoon [75], TOPSIS has been the most used MADM method in the literature. The main idea of the TOPSIS is to choose the best alternative based on the concepts of the compromise solution, i.e., choosing the solution with the shortest Euclidean distance from a positive ideal solution and the farthest Euclidean distance from a negative ideal solution.

Once the decision matrix is created, as illustrated on Figure 3.5, the TOPSIS algorithm can be applied. For better understanding, we divided the TOPSIS process in three main steps, as following:

- Normalized and weighted matrix – first of all it is necessary to normalize the decision matrix. Paper [76] showed that normalization techniques have impact in the MADM results. Moreover, paper [77] tested different normalization techniques and concluded that vector normalization is the best one for traditional TOPSIS algorithms. Once normalized, it is necessary to weight the decision matrix. This is achieved multiplying each attribute by a weight that represent the importance of such attribute in the decision process;
- Ideal solutions – it is necessary to find the ideal positive and ideal negative solutions. In the context of network selection, the ideal positive solution is a hypothetical access network composed by the best attributes present in the decision matrix, i.e., maximum value for attributes that we want to maximize (e.g., DR, PDR) and the minimum value for attributes that we want to minimize (e.g., latency, monetary cost). Similarly, the ideal negative solution is composed by the worst attributes in the decision matrix;
- Select solution – once the hypothetical ideal solutions are identified, the algorithm chooses as solution the access network having the shortest Euclidean distance from the positive ideal solution and the farthest Euclidean distance from the negative ideal solution.

Several works have developed TOPSIS-based algorithms to network selection [26, 60, 64]. Analyzing such a TOPSIS-based algorithms, we observe some limitation for its use in the ITS domain.

TOPSIS algorithm is designed to choose solution for only one problem at a time. When applied for problems of network selection, it can find solution (i.e., the best access network) for only one application at a time. In case of multiple applications simultaneously competing for communication resources, the TOPSIS algorithm should be adapted accordingly. For example, the algorithm can be executed as many times as there are applications. In such an approach, the order in which applications are served is important. For example, if a bandwidth-hungry application is served first and without any bandwidth limitation policy, it can occupy the entire available

bandwidth, preventing other applications to communicate. While if the system enforces other applications first, more applications can communicate simultaneously.

Despite the MADM techniques to present relative low computation complexity, this approach has some issues. Comparisons between some MADM algorithms (SAW, MEW, and TOPSIS) showed that each algorithm has different sensitivity to weights [78]. Since the results are dependent of weight definition, to choose the best set of weights is a not trivial task. Moreover, MADM algorithms can present ranking abnormality, i.e., a small change in one parameter of the objective function implies full algorithm recalculation, which can determine a different best solution [79]. Therefore, since vehicles can move at high speed and frequently changing network environment, MADM techniques can be counterproductive. Besides being system process consuming (due to the need for constant full recalculations), MADM algorithms are susceptible to unstable decisions. Such decision instability can cause the well-known “Ping-Pong” effect, i.e., when mobile device switches from one access network to another but is quickly handed back to the previous one. This usually happens when the mobile approaches the limit of range of a given network (where the network can appear or disappear between two steps of calculations).

In order to reduce the number of recalculation performed by the traditional TOPSIS when network parameters change, authors of paper [60] define a new TOPSIS algorithm called Dynamic TOPSIS (D-TOPSIS). Such algorithm consider four attributes that are classified into two groups: static attributes (e. g., the network capacity previously reserved to the terminal, the monetary cost paid by the user to use a given network and the power consumption of the terminal) and dynamic attributes (e. g., the RSSI that is measured by the user terminal). The Euclidean distance to the ideal solution is performed only for dynamic attributes, reducing the necessary computational task when compared with traditional TOPSIS. In other words, the D-TOPSIS algorithm only limits the number of dynamic attributes, i.e., the number of attributes that can change during the network selection process.

TOPSIS approach selects the network with highest score regardless of the application satisfaction level. (i.e. the higher is the utility value of a decision, the better the solution). However, this evaluation does not reflect the actual applications needs. In some cases, flows can be solicited to change of access network even if the current one totally satisfy their communication requirements. For example, considering a flow with 300 Kbps of maximum required DR and that communicates through a WiFi network (called WiFi-1) with 1 Mbps of available DR; if another WiFi (WiFi-2) offering 2 Mbps appears and if we do not consider other parameters more than DR, the decision maker based only on the objective function is supposed to move the flow over WiFi-2. However, both WiFi networks satisfy 100% of flow requirement, i.e., 300 Kbps, and it would be better to maintain the flow through the WiFi-1, in order to avoid packet loss or increased latency due to the new network association.

Authors of [66] propose a new approach for the TOPSIS algorithm, which claims to remedy both, the so called TOPSIS ranking abnormality and the fact that original TOPSIS selects the network with highest score regardless of the application satisfaction level. To avoid the rank reversal, they propose to normalize attribute values individually. Therefore, changes in one attribute will have no effect on the normalization of others. In order to improve solution quality, i.e., to choose the network that better match the application needs, they propose to replace the classical quadratic normalization by a new approach based on utility functions. In this way, applications can present a specific utility function for each attribute. Simulations showed that the new approach eliminated the rank reversal and increased the ranking quality by fulfilling the application requirements. However, this approach eliminates rank reversal only for the specific cases when a mobile approaches the limit of range of a given network (where the network can appear or disappear between two steps of calculations), presenting ping-pong effect in other use cases. Moreover, this approach considers all flows over only one access network at time. We evaluated a version of this approach on chapter 6, which we named mTOPSIS.

Based on all these aforementioned TOPSIS limitations, we was motivated to verify other classes of algorithms capable to overcome them. ACO algorithm class has good properties that can be explored in order to overcome these TOPSIS limitations. In the following, we briefly describe the ACO principle, highlighting its main properties, and we show some areas in which ACO-based algorithms have been used.

#### 3.2.2. ACO

Swarm Intelligence (SI) is an important category of optimization methods that is based on behaviors and self-organizing interaction among agents, such as ants, bees, fish, and so on. The tasks performed by the collective and cooperative behaviors of agents cause global patterns capable of solving complex problems [80].

ACO is a SI metaheuristic optimization method based on the foraging behavior of real ants. Ants are insects that, in spite of their simplicity, have a large capacity to organize themselves and perform complex tasks. Observations on the ants behavior, such as the search for food, division of labor and cooperative transport, have inspired studies on the field of computation. Thus, models have been created in order to replicate such behaviors, allowing artificial ants to solve complex computational problems.

It is known that ants do not have an accurate eyesight, some species being totally blind. According to Dorigo [81], most of the communication among ants, or between ants and the environment is based on the use of pheromones, i.e., chemicals produced and deposited on the ground by the ants while walking. Therefore, while moving ants tend to choose the path with the

highest pheromone concentration.

Figure 3.6 demonstrates the process of ants searching for food, where the colony nest is located at point “N” and the food source at point “F”. When ants start the process from their nest there is no pheromone on the ground. Hence, the first ants leaving the nest do not have a preference and they select a path randomly.

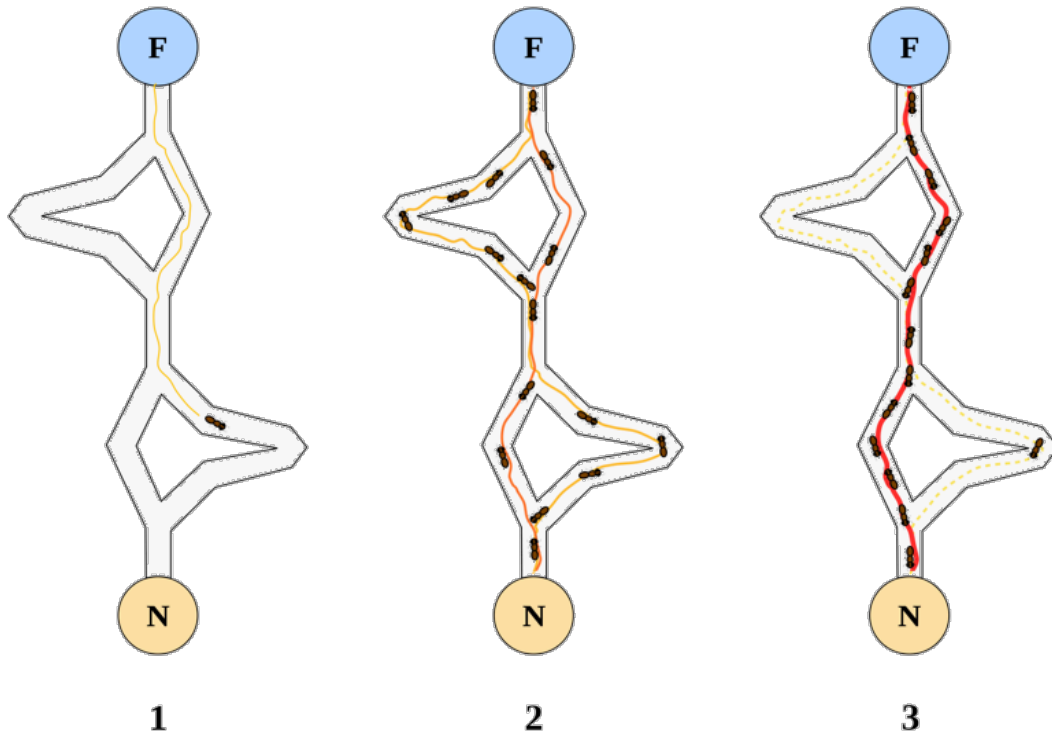


Figure 3.6: Process of ants searching for food

Ants choosing shorter paths are the first to reach the food and to start their return to the nest, always depositing pheromone where they pass (Figure 3.6-1). Therefore, pheromone starts to accumulate faster on the shortest path, while paths with less traffic of ants tend to extinguish the pheromone by evaporation (Figure 3.6-2). As shown by Dorigo [81], in the course of time the ants tend to navigate through the shortest path between the nest and the source of food (Figure 3.6-3).

The ACO algorithm simulates such behavior of ant colonies by creating artificial ants. This algorithm is designed to solve problems that can be represented by a graph, where a possible solution is a specific path in such graph. Despite real ants construct solution based on the Euclidean distance between nest and the source of food, the ACO metaheuristic can be applied to any combinatorial optimization problem for which an objective function (i.e., utility or cost function) value can be defined to each candidate solution.

ACO algorithms can solve both static and dynamic combinatorial optimization problems. We mean by static problems, those in which the characteristics of the problem are given once in the problem definition and do not change while the problem is being solved (e.g., the well

know Traveling Salesman Problem (TSP)). On the other hand, in dynamic problems the problem instance can change at run time. In this case, the optimization algorithm must be capable of adapting online to the changing environment. As examples of this kind of problem we can highlight the data routing in Vehicle Ad hoc Network (VANET), in which the data traffic and the network topology can vary in time [82]; as well as the choice of access network for a given data flow communication, i.e., one of the problems addressed in this thesis.

ACO algorithms have been used in different kind of optimization problems. For data mining: authors of paper [83], propose an ACO-based algorithm for data mining called Ant-Miner. The goal of such algorithm is to extract classification rules (like `<IF term1 AND term2 AND ... THEN class>`), from data sets. In this approach, each term is composed by a triple (attribute, operator, value), e.g., (Gender = female). In the Ant-Miner, each ant visits the data set and chooses one term at a time to its current partial rule. Such partial rule corresponds to the partial path followed by that ant, while the choice of terms depends on both an heuristic function and on the amount of pheromone associated with each term. Such discovery process is performed iteratively until the number of uncovered training cases is less than a user-specified threshold.

For cloud task scheduling: paper [84] proposes an ACO-based algorithm to cloud computing. The ACO algorithm is used to allocate incoming jobs to virtual machines, while it minimizes the makespan of a task set, i.e., the total time from the beginning to the end of a task set. The problem is represented as a graph  $G = (N, E)$ , where the nodes  $N$  are composed by tasks and virtual machines (VMs), and the edges  $E$  represents the connections between tasks and VMs. At the begin, ants are randomly placed on VMs. During the process to find solutions for the cloud task scheduling problem, ants move from one Virtual Machine (VM) to another, while choose the task to be allocated for each VM. The process stops when all tasks have been allocated.

And also for network selection: In paper [34] authors describe a DM framework for network management, in which a combination of MADM SAW and ACO-based algorithms is used to select access networks for each flow communication. The main goal of this work is to select the access network that better meets what they call system satisfaction, i.e., meets the requirements of each flow, reducing the network power consumption and network load. First, the SAW algorithm calculates an utility score for each feasible flow-network solution. Such score indicates the matching degree between flow requirements and network characteristics. In a second step, the ACO-based algorithm adds the network costs (i.e., power consumption and network load) to the previous utility function in order to find solutions that increases the whole system satisfaction.

In the ACO process designed by this paper, ants start the searching process from a given flow and visit all flows in a predefined order. In each visited flow, ants choose an access network that better meets the flow requirements. Once having visited all flows, ants compare their solutions, choosing the better one, i.e., the solution with higher utility score.



Paper [34] modeled the ACO problem in a way that the order in which flows are addressed is important, i.e., changing the order in which the ants traverse the graph can generate different solutions. Such an approach can be difficult to manage in a dynamic environment as the ITS one, in which flows and access networks can appear and disappear frequently. This paper addresses the decision stability by changing the weight of network costs so that currently enforced solutions are privileged. However, choosing the best weight is not a trivial task.

#### 3.2.3. How works have addressed the DM algorithms

In order to show which kind of decision making algorithms have been chosen by recent research works, we considered the same works listed on Table 3.1 and we added two more information, as shown on Table 3.2. The two added columns are:

- Algorithm – it indicates the algorithm name used in the decision making process;
- Single (S) / Multiple (M) interface – indicates if the DM manages single or multiple interfaces, i.e., it indicates if the algorithm is capable to send data flows through multiple access networks simultaneously (Multiple (M)), or the algorithm considers all flows through only one access network at a time (Single (S)).

### 3.2. RESEARCH WORKS RELATED WITH DM ALGORITHMS

Table 3.2: Main research works related with DM algorithm

N	DM	APP	USER	ADM	Context	Network monitoring	Monitoring solution	Single (S) Multiple (M) Interfaces	Algorithm	Paper
1	v	v	v	v	v	v		M	ACO	[34]
2	v	v	x	x	v	v		S	MADM GRA	[24]
3	v	v	v	x	x	v	MIH	S	MADM AHP	[35]
4	v	v	v	x	x	v	MIH	M	static decision	[28]
5	v	v	x	x	x	x			MADM AHP	[40]
6	v	x	x	x	x	v	MIH	S		[25]
7	v	v	x	x	v	x	Fixed values	S	Q-Learning	[42]
8	v	v	x	x	x	v	MIH	M		[55]
9	v	v	v	x	v	v	MIH	S	MADM TOPSIS	[26]
10	v	v	v	v	x	v	MIH	S	MADM AHP	[32]
11	v	v	v	x	x	v	MIH	S	WMC	[31]
12	v	v	x	x	x	v	Scout	M	static decision	[45]
13	v	v	x	x	x	x		S		[41]
14	v	x	x	x	v	v	MIH	S	MADM AHP	[33]
15	v	v	x	v	x	v		M	Game theory	[29]
16	v	x	x	x	x	v		M	Greedy	[56]
17	v	v	x	x	x	v	UCMM	M	static decision	[44]
18	v	v	x	x	v	v		S	static decision	[27]
19	v	v	v	v	v	v		M		[54]
20	v	v	v	x	v	x		M	MADM GRA	[30]
21	v	v	v	x	v	x		M	MADM SAW	[39]
22	v	v	v	v	x	x		M	MCAS	[38]
23	v	x	x	x	x	v		S	MADM D-TOPSIS	[60]
24	v	v	v	x	x	v		S	MADM WPM	[61]
25	v	v	x	x	x	v			Fuzzy	[62]
26	v	v	x	x	x	v			PSO and GA	[63]
27	v	v	x	x	x	v		S	MADM TOPSIS	[64]
28	v	x	x	x	x	v		S		[65]
29	v	v	x	x	x	v		S	MADM TOPSIS	[66]
30	v	v	x	x	v	v		S	MADM SAW	[67]
31	v	v	v	x	x	v		S	MADM MEW	[68]
32	v	v	v	x	x	v		M	Tabu search	[69]

As can be observed on Table 3.2, most of DM algorithms are based on the well-known MADM algorithm class. As mentioned before, despite such an algorithms have relative low computation complexity, they present some issues like high sensitivity to extreme values which leads to unstable decisions, high sensitivity to attributes' weight, and the necessity to make full recalculation even if only a given network parameter change.

### **3.3. Attributes**

Once we verified how works have addressed the actors in the decision making process, and the algorithms used by them, we need to identify which kind of attributes works have considered in their decision making process.

Most of previously described DM algorithms make their decisions based on an objective function (cost function or utility function), which is dependent of the attributes acquired from different actors. In this process, a DM mechanism defines the attributes to be analyzed (e.g., the available network bandwidth and the communication latency), their correspondent objectives (e.g., maximize the communication bandwidth while reduce the communication latency), and based on that defines an objective function. Therefore, the decision making process consists of finding the access network that better meets such objectives, i.e., the access network that present a better score for the objective function.

As already described on Figure 3.1 different actors are able to present theirs requirements, preferences, constraints and policies to the DM. In the follow, we aim to identify the attributes the most commonly used in the literature. Such analysis is based on more than 50 papers that tackles decision making process for network selection.

First, we identified the attributes used by each paper. Then, we identify which actor have provided such attributes. Figure 3.7 gives an overview of attributes distribution among actors. As we can observe, the two actors more frequently considered by papers are application and network, i.e., most papers consider application requirements and network constraints in their decision making process.

Therefore, in the follow we analyze in more details the attributes considered from both applications and networks. For that, we listed the application and networks attributes considered for all analyzed papers. Figure 3.8 highlights the main application/flow requirements. We observe that the most frequent attribute is the DR representing 27% of all flows requirements, i.e., 27% of analyzed papers consider the DR as one of the attributes for the decision making process. The latency occupies the second place among the most commonly used attributes, being used by 18% of the papers we analyzed. Packet or bit error rate represents 14%, while jitter represents 8% of all flows requirements. The other 33% are occasional attributes like energy consumption, vehicle speed and others, which are used by few works and usually for specific problems.

Attributes per actor

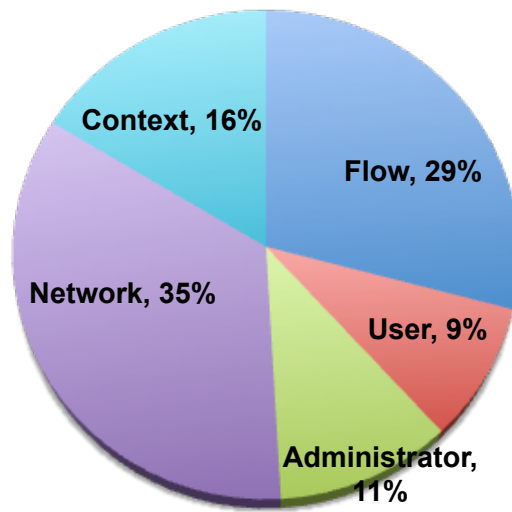


Figure 3.7: Percentage of attributes per actors

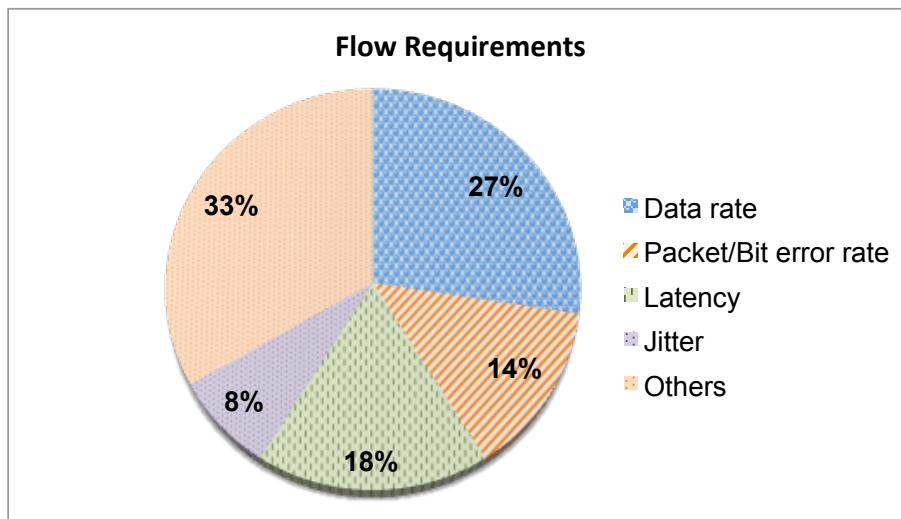


Figure 3.8: Flows requirements

We performed the same analysis for the attributes related with network conditions. The results are shown on Figure 3.9, which highlights the main network constraints. DR represents the most frequently used constraint. Around 19% of papers consider the access network DR as an attribute in their decision making process. The communication latency observed in each access network is considered by 16% of analyzed papers. While packet or bit error rate represent 14% of all network constraints. Like for the flow requirements, the “others” category (29%) are occasional network constraint, like probability of network availability used by few works.

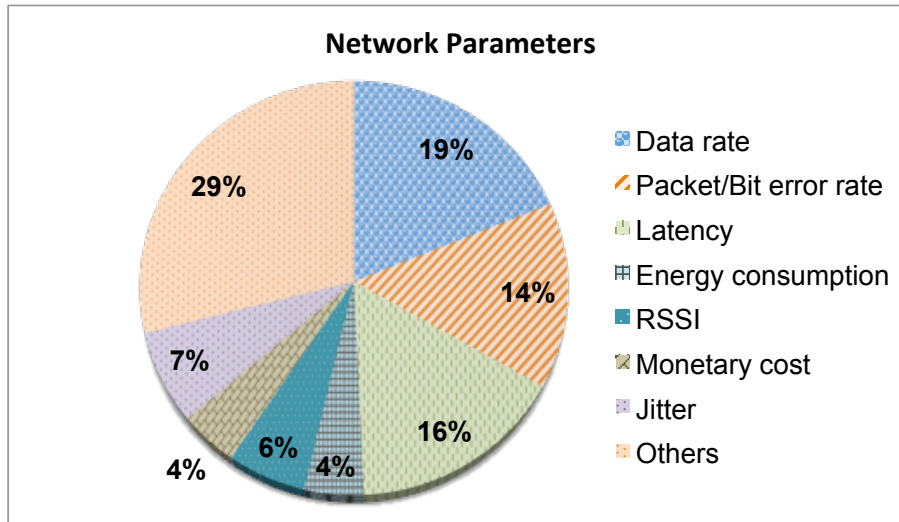


Figure 3.9: Network constraints

### 3.4. Conclusion

Research works have been interested for different parts of the ISO ITS architecture. Despite different actors are addressed (application requirements, user preferences, administrator rules and network information), most works have concentrated on the application and network information. Current DM usually considers static application requirements while looks for the best access network that satisfy such requirements.

Therefore, once the solution is found, the DM enforces them in the system. As previously discussed, decisions can be applied at different layers of computing system (i.e., application, transport, network or link layer). However, each of such layers presents specificities in terms of level of technology dependency, knowledge of application requirements and time to adapt to a new decision policy. Due the flexibility of network layer protocols, and the possibility to manage end-to-end communication flow by flow, in this thesis we consider to apply DM decisions at network layer.

Despite being a well-studied area, the decision making process does not have a complete solution that meets the C-ITS needs, like being implemented in a standard compliant way, or addressing the high dynamism of the vehicular environment. Most of current works have focused on the development of algorithms for handover process and network selection. The TOPSIS has been the most used MADM method in the literature. Despite such algorithm to present relative low computation complexity, this approach has some issues like high sensitivity to extreme values, which leads to unstable decisions; high sensitivity to attributes' weight; and the necessity to make

full recalculation even if only a given network parameter change.

Analyzing ACO-based algorithms, we observe that they have some properties that can be explored to address ITS communication needs. For example, ACO algorithms are capable to find high-quality solutions for combinatorial optimization problems in a reasonable time. Solutions are created smoothly over time, increasing decisions stability and preventing “ping-pong” effect. Such algorithms are running-time adaptable, i.e., if characteristics of a solution change (e.g., an access network become more expensive), ants are capable to find alternative solutions at running-time. Moreover, ACO are memory-based algorithm, i.e., current solutions take into account the previous status of the system. This behavior can be used to a better stability of decisions.

Current DM algorithms are reactive, i.e., they find new solutions by reacting to the network changes. However, due to the high velocity of vehicles, their connectivity context can change frequently. In such highly dynamic mobility scenarios, it is needed a DM capable to anticipate decisions based on forecasted information.

Based on these findings, we worked on the AD4ON, a DM mechanism for opportunistic networking in heterogeneous network environment. Based on such literature review and on the ISO ITS standards, we developed the AD4ON architecture, which is detailed in the next chapter (Chapter 4). Similarly, based on ACO we develop the AD4ON algorithm, which is described in the Chapter 5.

# 4

## AD4ON architecture

As mentioned in Chapter 3, works have developed their own DM architecture, usually developed for specific problems and scenarios in a silo approach. However, the C-ITS environment presents some communication challenges that such a silo approach is not capable to meet. For example, the need to manage multiple attributes and multiple devices, the need to deal with heterogeneous networks simultaneously, and the necessity of interoperability between applications and between connected devices.

Unlike the majority of the reviewed works, we do not want to create an isolated DM architecture. Instead, we aim to design an architecture that can be inserted in the existing ITS communication ecosystem, i.e., the ones described by current standardization bodies, as described in Chapter 2. Such a design approach, enables cooperation between applications and between devices, what is not addressed by works based on silo approach.

In order to enable C-ITS communication, we base our architecture in the ISO ITS-S reference architecture. We chose such architecture due its good properties for the C-ITS environment. As described before, the ISO ITS-S reference architecture is a framework that provides the DM mechanism with all needed inputs and interfaces, as well as the means to enforce the decisions. In this way, the DM mechanism is development specific and it can be seen as a “black box” that receives the inputs from different actors, make the decision, and apply it in the system, as shown on Figure 4.1.

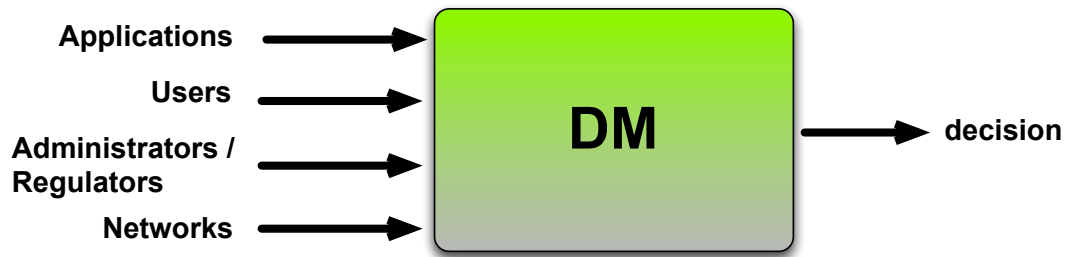


Figure 4.1: DM mechanism Inputs/Output

Therefore, our main objective is to design such “black box”, i.e., to design our DM mechanism. In this chapter, we design the AD4ON architecture, a modular architecture for opportunistic networking in heterogeneous access network environment.

## 4.1. Expected properties

Before start the AD4ON architecture design, it is necessary to define which are the properties we aim to meet with our DM mechanism. The C-ITS communication environment has specific characteristics, like the growing number of connected devices and the high velocity of vehicles, which require some properties from the DM. Based on our research and on the ITS architecture proposed by ISO, we identified some properties we consider important for such a decision mechanism. In the following, we describe such main properties.

### 4.1.1. Modular architecture

Because standards describe minimum sets of essential features for interoperability, we have studied a number of design choices. A main choice is how to integrate a wide variety of conceptual elements in the standards into actual implementation. A simple solution is to make a single self-contained software component, while the other way is to actually separate each entity.

In general, a single self-contained software is usually superior in terms of performance: if we couple all conceptual entities into a single component, the interface between the entities are much simpler. However, this solution lacks extensibility and is difficult to maintain, specifically if it is developed by multiple stakeholders. While a modular solution is efficient in terms of interoperability.

Since the main characteristic of the C-ITS systems is the cooperation between heterogeneous devices, a modular architecture seems most suitable. Besides to allow interoperability, such an architecture can be split in distributed entities in-vehicle onboard network.



Moreover, the communication architecture should be scalable, i.e., it should be able to manage a growing number of vehicles as well as a variety of flows and users.

#### 4.1.2. Multiple attributes and objectives management

In the C-ITS communication, different actors are able to present their requirements, preferences, constraints and policies in the decision making process, as shown on Figure 4.2. For example, applications can request a specific bandwidth, data rate or security level. Users can present their preferences, e.g., defining a priority or security level for a given message. Industrial and mobility service providers (i.e., operators) can present their policies, such as network constraints and particular billing procedures. These attributes are usually expressed in different scales and units. The DM should be capable to manage this heterogeneity of attributes.

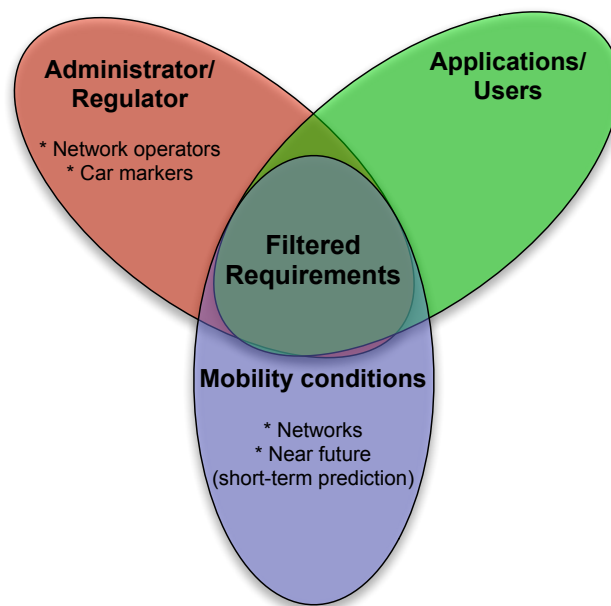


Figure 4.2: Multiple attributes/actors

Actors can have different objectives, which can be contradictory. A network operator can aim to improve network load balancing without regard to monetary cost, while an user can prefer a cheaper network. Few works have dealt with these kind of contradictory requirements [85]. However, the DM architecture should be capable of managing these multiple objectives simultaneously.

Moreover, the requirements and preferences from different actors can change over time. An user can accept a temporary QoS degradation if connected to a free access network, and require a better QoS when billed for this. Or accept a expensive network if he or she has a high budget, and reject them when the budget is below a threshold. In this context, the DM should be always

updated with latest network information, administrator's rules, user's preferences and applications' requirements.

Most of recent works in decision making process already consider multiple attributes [26, 31, 45]. However, such attributes come from a reduced number of actors, usually only from applications and networks, as shown in Table 3.1. Moreover, most of such works consider static and predefined attributes from applications, i. e., attributes are assigned at the begin of the decision process and remains always constant [32, 24].

#### 4.1.3. Management of heterogeneous network

Initial works related to ITS have assumed a universal communication technology being used for all ITS usages, e.g., IEEE 802.11p. However, as the studies evolved it became evident that such a deployment is impractical or even impossible. There are a wide variety of services, each one with specific requirements in terms of DR, latency, monetary cost and others. Due the high monetary cost associated with network deployments, as well as the complex regulation issues to fully cover all countries, it is impractical to rely solely on a dedicated access technology for specific services.

Nowadays, there is no a single communication technology that fulfills all the heterogeneity of services requirements. For ubiquitous connectivity and to better fulfill requirements from the heterogeneous applications, it is necessary to use all existing wireless technologies, such as vehicular WiFi (ITS-G5 in Europe, or DSRC in North America), urban WiFi (e.g., 802.11 g/n/ac/ax), 802.15.4 or cellular (3G, 4G, and 5G under preparation). Such a scenario, exposes the vehicle to a complex radio environment where a wide variety of access technologies can be available simultaneously, as shown on Figure 4.3.

Therefore, vehicles equipped with multiple communication capabilities can take advantage of these heterogeneous technologies, enabling better use of each access network. For example, devices can use multiple access technologies at once in order to maximize communication bandwidth, to choose the network that better matches with application requirements, and/or to make a better handover management.

In such heterogeneous network environment, applications cannot take into account all technology particularities, unless they explicitly need it. Instead, it is necessary a communication architecture that hides the heterogeneity of underlying access technologies, providing seamless communications. The possibility of having multiple applications simultaneously competing for communication resources requires a controlled access to these resources. In this context, a decision maker is needed to manage all available access networks simultaneously, attempting to choose for each data flow, the access networks that better match with their requirements. Therefore, applications are not limited to a single access technology, but they can take advantage of all

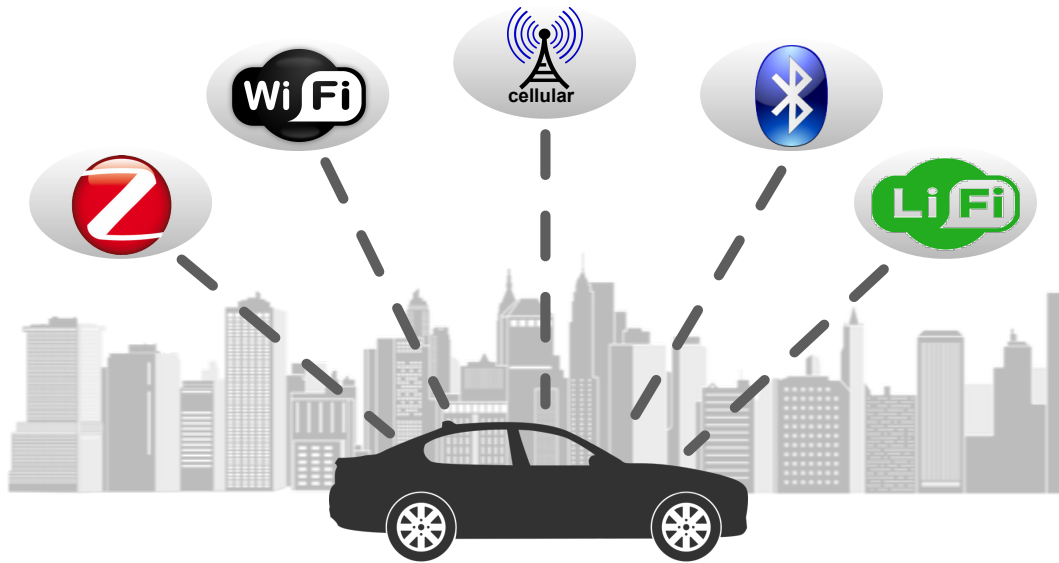


Figure 4.3: Heterogeneous access networks

available technologies.

#### 4.1.4. Awareness of the vehicle surroundings

Make smart decision in the C-ITS environment is not effortless. It requires awareness of the vehicle surroundings, i.e., it is necessary to monitor a variety of information, in order to make smart decisions. The most evident information to be monitored is the available networks and their characteristics. In the literature, most of works consider to acquire such network information by cooperation with the networks. This cooperation is usually based on the MIH standard (IEEE 802.21), in which specific functionalities are supposed to be implemented and supported on the network side. However, since we cannot guarantee that all networks support such a standard, monitoring modules should be able to monitor network information even no specific monitoring functionality is implemented on the network side.

Besides networks availability and performance, other information should be monitored. Since applications can start or stop their communication at different instants, as well as change their requirements over time, it is necessary to constantly monitor them. The DM should be informed whether an application flow is alive or not, as well as constantly monitor flows requirements.

In addition, to conceive an intelligent decision making process, context information can be considered. DM should be able to take information from their environment, as geographical position (e.g., GPS) or vehicle's speed in order to adjust the decision's strategies. For example, deactivate an interface when the vehicle arrives in a zone, in which the network assigned to such interface is forbidden, e.g., deactivate the ITS-G5 when the vehicles arrives near to a toll road that

uses DSRC as access technology.

### 4.1.5. Flow per flow management

As discussed in Chapter 3, we could manage the communication by considering package by package. However, this approach implies to work at the transport layer, and thus taking into account the CN conditions. Other solution could be to manage network interface by network interface, but this approach present less flexibility, e.g., flows with different communication requirements can be enforced through the same interface. In this thesis we make a very strong choice to work by flow, i.e., to manage communication flow by flow.

In C-ITS communication, there is a wide variety of applications, each one with one or more data flows. A data flow is defined by ISO as an identifiable sequence of packets [15]. And in this thesis we use both terms *data flow* and *flow* as synonyms. Each flow can have specific communication requirements in terms of DR, latency, PDR, and others. For example, a safety-based service (e.g., emergency breaking information) is highly sensitive to packet loss and latency, whereas a video streaming service is less sensitive to latency and bandwidth changes.

A flow per flow management enables the DM to make fine-grained decisions, choosing for each flow the access network that better meets their requirements. Moreover, such a management approach allows the DM to make better use of network resources, e.g., spreading flows among different access networks at a time.

Since some applications does not deal well with out of order packets, we try to reduce or even eliminate the interference in the packets order, e.g., avoiding routing packets of the same flow through different interfaces.

### 4.1.6. Anticipate decisions

Besides the heterogeneity of network technologies, applications and devices, the C-ITS communication environment is characterized by its highly dynamic mobility. Vehicles can move at high speed and frequently changing network environment, i.e., causing networks to appear and disappear rapidly. This causes frequently changes in the vehicle's point of attachment to the network. Moreover, moving inside the coverage of a given network, vehicle can experience different network conditions, in terms of DR, availability, latency and PDR.

In such a dynamic scenario, DM can be susceptible to unstable decisions, which cause the well-known “ping-pong” effect. This usually happens when the vehicle approaches the limit of range of a given network. Such constant changes in the access network bring some communication issues like PDR degradation, increased latency and consequent QoS reduction.

A way to mitigate such issues is by predicting the vehicle context, i.e., to predict vehicles' surroundings and the network environment in which it will be soon inserted. With the short-term prevision about the vehicle context, the DM responsible to manage all data communication services is capable to anticipate decisions or perform proactive decisions. For example, it can decide to increase the data buffer for a given video streaming, if the vehicle is going to cross a wireless dead zone; anticipate a network handover, by preparing the handover process in advance; or, delay a data transmission, if it knows that a better network will soon be available.

In the prediction process, besides the forecasted information (e.g., new access network availability, better DR for a given network, or the presence of a wireless dead zone) vehicles should be aware about how far out is such prediction, i.e., it should be informed if a given predicted context is a few seconds or some hours away from its current position. The prediction process is detailed in Chapter 7.

## 4.2. Architecture design

To achieve the expected properties, we designed the modular AD4ON architecture based on the ISO standards. The ISO ITS-S reference architecture is a framework defining the different layers and their related functionalities, as well as the main communication interfaces between them. In this way, such reference architecture gives some freedom for development of the DM mechanisms. In other words, once the interfaces between the DM and the distributed modules are identified and their contents defined (i.e., once the DM inputs and outputs are defined), the DM mechanism becomes independent. It can be seen as a module (or “black box” as mentioned in the introduction of this chapter) that receives inputs, make decision and present this decision in the output. Therefore, the AD4ON architecture is the development of such “black box”.

### 4.2.1. Main communication interfaces with AD4ON

The AD4ON can interact with controlled entities in all layers of the ITS-S communication architecture. Such communication is performed towards standardized communication interfaces between the different layers. In the following, we describe the two main interfaces for this thesis:

- MA-Service Access Point (MA-SAP) – interface between the ITS-S application layer and the ITS-S management layer;
- MN-Service Access Point (MN-SAP) – interface between the ITS-S management layer and the ITS-S network & transport layer.

ISO 24102-3 [86] classifies Service Access Point (SAP) in two types, according to who initiate the service. Services initiated by the ITS-S management layer are known as “*Commands*” while the ones initiated by the ITS-S application layer or ITS-S network & transport layer are known as “*Request*”. Furthermore, each one of such classification has two service primitives: one to trigger an action (i.e., “*request*”) and another one to report the results of the performed action (i.e., “*confirm*”). Figure 4.4 depict such classification.

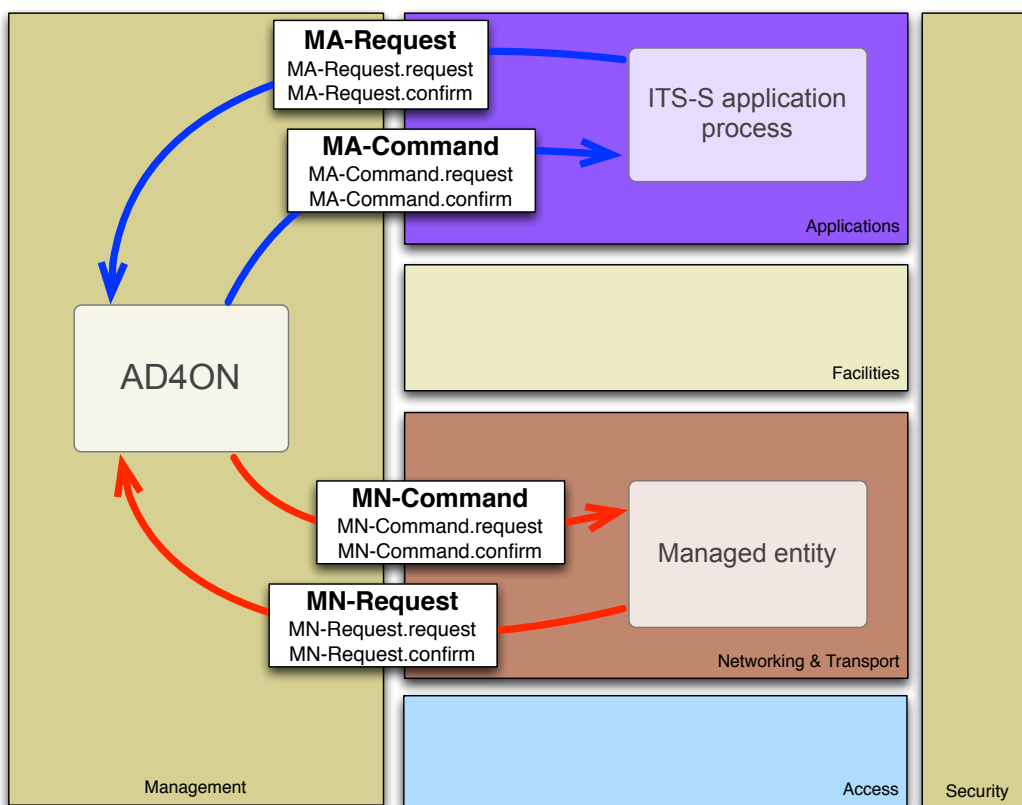


Figure 4.4: Communication towards MA-SAP and MN-SAP

The service primitives defined by ISO for the MA-SAP and MN-SAP are detailed below.

### MA-SAP

This service access point is used for communication between ITS-S application layer and ITS-S management layer. As shown on Figure 4.4, the MA-SAP has four service primitives: *MA-Request.request*, *MA-Request.confirm*, *MA-Command.request*, and *MA-Command.confirm*. Since the primitives follow the same framework, in the following we show the primitive structure only for *MA-Request.request* and *MA-Request.confirm*. The others are supposed to use similar structure.

When an ITS application process needs to trigger an action in the AD4ON, it sends the *MA-Request.request* service primitive. For example, an application uses such primitive to present

its communication requirements to the AD4ON. The structure of such primitive is showed on Figure 4.5, and the arguments used by the *MA-Request-request* service are described on Table 4.1.

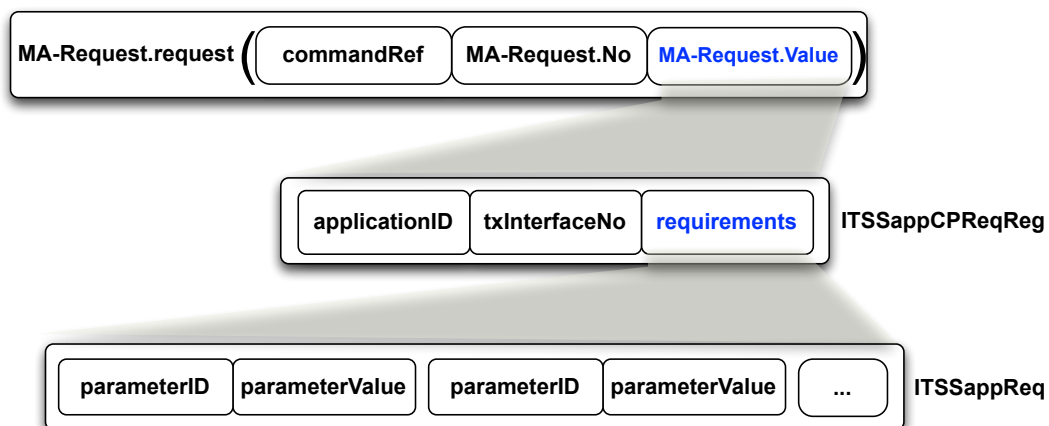


Figure 4.5: The structure of the *MA-Request.request* service.

Table 4.1: Parameters of the *MA-Request-request* service

Name	Description
commandRef	Unique cyclic reference number of command
MA-Request.No	Reference number of the request
applicationID	Identifier of an ITS-S application process. Specified in ISO 24102-1 [87]
txInterfaceNo	Sink or source of an ITS-S application process. Specified in ISO 17419 [18]
parameterID	Integer values predefined for each parameter. E.g., 15 indicates minimum throughput, 17 indicates maximum acceptable latency, and 29 indicates priority flow parameters. Specified in ISO 17423 [16]
parameterValue	Values assigned for each parameter

Once the action requested by the application is performed by the AD4ON, it replies the application with the *MA-Request.confirm* service primitive. The structure of such service primitive is showed on Figure 4.6, and its specifics arguments are described on Table 4.2.

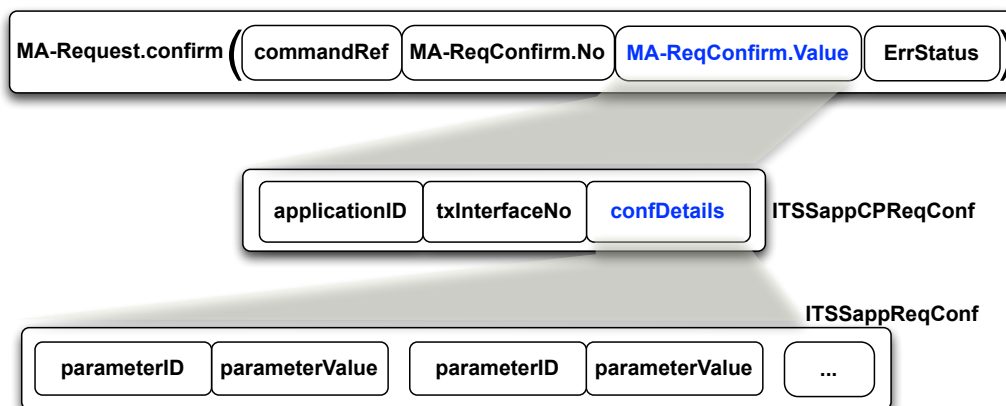


Figure 4.6: The structure of the *MA-Request.confirm* service.

Table 4.2: Parameters of the *MA-Request-confirm* service

Name	Description
MA-ReqConfirm.No	Reference number of the request. Same value as MA-Request.No in related <i>MA-Request.request</i> .
ErrStatus	Values predefined in ISO 24102-3 [86]. E.g., (0) success, (3) invalid parameter value, and (10) value not available.

Following the same reasoning, the *MA-Command.request* service primitive allows the ITS-S management entity to trigger an action at the ITS-S application layer. For example, such primitive enables the AD4ON to alert adaptive application about network conditions. The arguments used by the *MA-Command-request* service are described on Table 4.3.

Table 4.3: Parameters of the *MA-Command-request* service

Name	Description
commandRef	Unique cyclic reference number of command
MA-Command.No	Reference number of command.
MA-Command.Value	Value of command.

Once the action is performed by the application, it replies the AD4ON with the *MA-Command.confirm* service primitive.

## MN-SAP

This service access point is used for communication between ITS-S network & transport layer and ITS-S management layer. Similarly the MA-SAP, the MN-SAP has four service primitives: *MN-Request.request*, *MN-Request.confirm*, *MN-Command.request*, and *MN-Command.confirm*.



When modules in the ITS-S network & transport layer needs to trigger actions in the AD4ON, it uses the *MN-Request.request* service primitive. For example, network monitoring module located in the ITS-S network & transport layer uses such primitive to send information about network performance to the AD4ON in the ITS-S management entity.

The arguments used by the *MN-Request-request* service are described on Table 4.4.

Table 4.4: Parameters of the MN-Request-request service

Name	Description
commandRef	Unique cyclic reference number of command
MN-Request.No	Reference number of the request
MA-Request.Value	Value of the request

Once the action is performed by the ITS-S management entity, it replies with the *MN-Request.confirm* service primitive.

The management service primitive *MN-Command.request* allows the ITS-S management entity to trigger an action at the ITS-S network & transport layer. For example, such primitive enables the AD4ON to enforce a decision in the network layer. The arguments used by the *MN-Command-request* service are described on Table 4.5.

Table 4.5: Parameters of the MN-Command-request service

Name	Description
commandRef	Unique cyclic reference number of command
MN-Command.No	Reference number of the command.
MN-Command.Value	Value of the command

Once the action is performed by the ITS-S network & transport layer, it replies the AD4ON with the *MN-Command.confirm* service primitive.

### 4.2.2. AD4ON architecture

In this section, we define the different functional modules that compose the AD4ON architecture, as well as the main interfaces between them, i.e., we specify the type of information circulating through each interface. Based on some previous research works like the thesis of Lucian Suciu [88] and Rayene Rayana [49], we split the AD4ON architecture in five main parts, according to their functions:

1. Requirement gathering – responsible to acquire input from different actors;
2. Monitoring – accommodates different monitoring modules, e.g., the one responsible to monitor network conditions or the one responsible to monitor flow status;

## 4.2. ARCHITECTURE DESIGN

3. Near future – responsible to acquire and manage prediction about the vehicles context, i.e., forecast the context in which the vehicle will be soon inserted;
4. DM – is the core of the AD4ON mechanism, which accommodates the DM algorithm and it is responsible to perform the decision making process;
5. Applying decision – responsible to apply decision in the system.

Based on the ISO standards for ITS (e.g., [2, 16, 18, 15]), we can integrate the five parts of the AD4ON architecture in the ISO ITS-S reference architecture as shown on Figure 4.7.

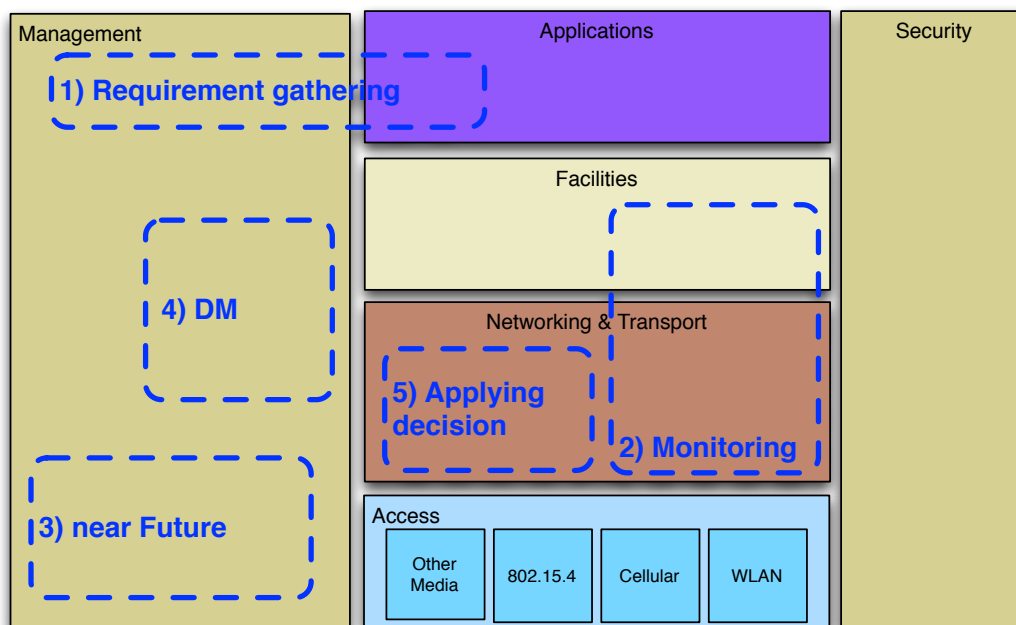


Figure 4.7: AD4ON architecture outline

In the following, we designed the main modules that compose each part of the AD4ON architecture. An complete view of the AD4ON architecture is shown on Figure 4.8. And each one of these five parts are detailed below.

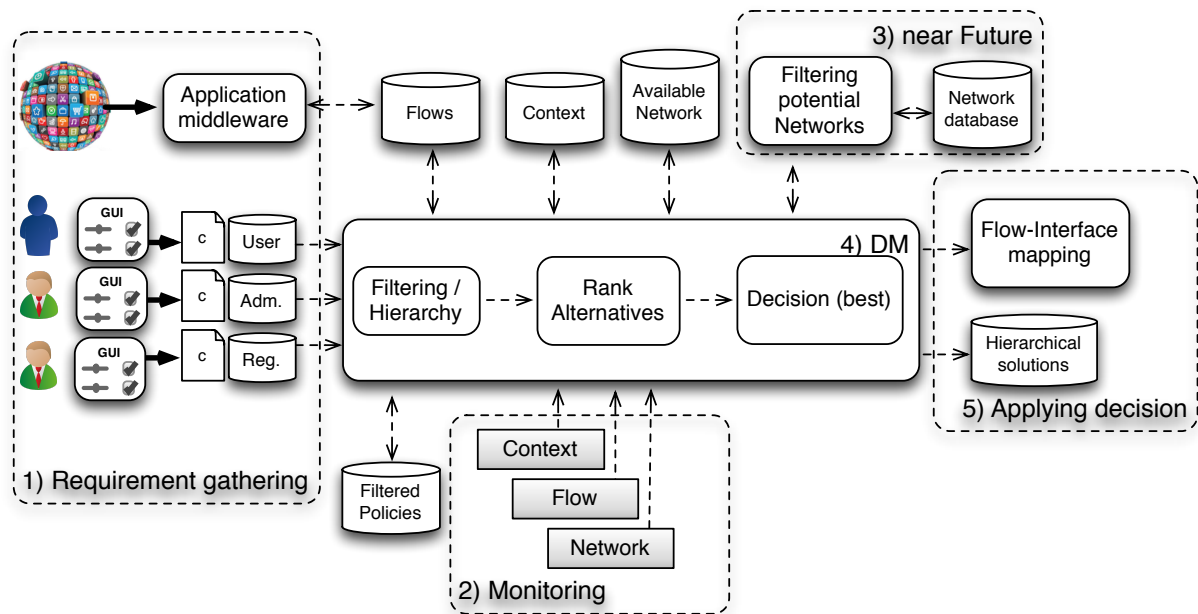


Figure 4.8: Proposed AD4ON Architecture

#### 4.2.2.1. Requirement gathering

As mentioned before, different actors are able to present their requirements in the decision making process. In the AD4ON architecture we consider four main actors: applications, users, administrators and regulator bodies, which are described below. Requirements, preferences and policies from all actors are stored in decision maker's databases and used by the DM to choose the communication interface that better matches the actors requirements.

#### Applications

As mentioned in Section 2.1.4, when an ITS station need to communicate with an CN, first the application running on the sender device proceeds the flow assignment operation, as defined by the ISO 24102-6 standard [15]. This process aims to select the most suitable communication profile for each flow, i.e., to select a collection of facilities protocols, transport protocols, network protocols, access technologies and communication channels that better meet the flow requirements [16].

As defined by [89], applications can be divided in four different traffic classes: conversational, streaming, interactive and background classes. Each one of these classes has specific requirements in terms of QoS. For example, conversational class groups real-time services like video and VoIP calls, which are very delay sensitive. While background class represents services like background downloads or e-mails, which are more delay tolerant. Services from interactive class, e.g., an online end-user requesting data from a remote server, usually have higher priority in scheduling

than services from background class.

Therefore, flows can have specific requirements in terms of QoS. First of all, it is necessary to define the flow requirements. Based on the literature and the communication requirements presented by ISO [16], we defined some key parameters that each flow should present to the DM, e.g., the maximum supported end-to-end delay, the sensibility for information loss, the minimum and maximum required throughput, the flow priority, and the security level (e.g., if flow carries sensitive information, it must be protected from unauthorized access). A list of such requirements are showed on Table 4.6.

Table 4.6: Communication requirements between Application and DM

Requirements	Unit	Description	FlowTypeID or FlowID
Minimum DR	bps	It indicates the minimum data rate required by a given application	FlowTypeID
Maximum DR	bps	It indicates the maximum data rate until which flow can improve its quality. Beyond such upper limit, we consider there are no quality improvements for the flow	FlowTypeID
Minimum latency	second	It indicates the minimum latency value, below which flow does not present quality improvements	FlowTypeID
Maximum allowed latency	second	It indicates the maximum latency supported by a given flow	FlowTypeID
Flow identifier	integer	Identifier of the ITS-S flow	FlowID
FlowType	integer	Identifier of the corresponding ITS-S flow type, i.e., it refers to the list of communication requirements	FlowID
Maximum Priority	string	It presents maximum allowed priority for a given flow (high, medium or low)	FlowID
Destination type	integer	It indicates type of transmission (e.g., single receiver, group of receivers, geographic area)	FlowID
Destination domain	integer	It indicates domain of communication (e.g., communication internal of the ITS-S, local communication, global communication (Internet))	FlowID

We divided such parameters in two groups. The first group is used to acquire the *FlowTypeID*, i.e., to verify if the ITS station has the necessary capabilities to support flows with a given flow requirement. This process is performed when DM does not know the flow. The second group is used to acquire the *flowID*, an identifier to a given communication profile. When application wants

to send messages, it registers the destination and the previously assigned *FlowTypeID* to the DM. In this process, a middleware enables different applications to send their requirements to the DM, and these requirements are used by the DM to select the best suited communication profile per flow (identified by *FlowID*). Table 4.6 shows such parameters exchanged between applications and DM, while specifies if they should be presented to obtain the *FlowTypeID* or *FlowID*. In this thesis, we do not address the development of such middleware, instead we consider to use existing solutions, as the one developed in the Lucian Suciuc's thesis [88].

## Users

Users can have specific needs. In the requirements gathering process, they can present their preferences to the DM, defining service priorities, security level for a given message or the amount of money they are willing to pay for a given service. Table 4.7 shows the user's preferences that are considered in the present work.

Table 4.7: Communication requirements between User and DM

Users' preferences	Unit	Description
Flow priority	string	It indicates the flows priority from user point of view. User presents such attribute in terms of linguistic terms (e.g., LOW, MEDIUM and HIGH)
Monetary cost	float	It indicates the maximum monetary value user is willing to pay for a given communication. (Euros per Mbps)
Security level preference	string	It indicates the expected security level for a given communication. User presents such attribute in terms of linguistic terms (e.g., LOW, MEDIUM and HIGH)

Such users preferences can be sent to the DM using system-level definition or using high-level (human friendly) definition. In the first case, the list of preferences can be directly interpreted by the decision making process without any pre-processing. However, such an approach is not easily readable by the end user. In the high-level definition (as addressed by Rayene Rayana's thesis [49]), users can specify generic goals through a Graphical User Interface (GUI) and the preferences are converted into system-level by using techniques like fuzzy logic.

Despite we design the high-level approach in the requirement gathering process, in this thesis we do not implement the translation between the GUI and the system database. For the sake of tests and simulation, we consider a list of preferences already converted into system-level definition.

## Administrators

Administrators, i.e., industrial and mobility service providers can also present their policies. For example, a network administrator can configure a particular billing procedure, present a network constraint, or forbid some types of flow on some networks. In this thesis, we consider two types of policies from the administrators, which are described on Table 4.8.

Table 4.8: Communication requirements between Administrators and DM

Policies	Unit	Description
Network restrictions	string	Telecom operator can forbid some type of flows for a given access network
Billing procedures	string	It is the way telecom operator is charging their customers (e.g., rental charge or usage charge)

Like in user preferences, for the administrator policies the translation between the GUI and the system database is not addressed. Instead, a list of already converted policies is considered.

### Regulator bodies

Regulator bodies can also express their policies. Countries or regions can define some specific rules, such as the prohibition of certain frequency ranges in certain areas, e.g., forbid the cohabitation of the ITS-G5 and the DSRC, whose cohabitation can cause harmful interference.

In this thesis, we consider only one type of information from the regulator bodies: the information of forbidden networks. We assume that such information is given as a list of prohibition by access network in certain areas and for certain flows. Such information are stored in the *Filtered Policies* database.

Like in user preferences and administrator policies, for the regulators rules the translation between the GUI and the system database is not developed.

#### 4.2.2.2. Monitoring modules

Making smart decisions in the C-ITS environment requires awareness of the vehicle surroundings, i.e., it is necessary to monitor a variety of information like network availability and performance, flows status, and vehicle context information (e.g., vehicle speed, geographical position).

We defined three monitoring modules.

- *Network monitoring module* – in this process, the network monitoring module listens to the wireless interfaces and informs the DM about the available wireless networks and their performances. Such monitoring module should be able to monitor network information

even if no specific monitoring functionality, such as IEEE 802.21 [90], is implemented on the network side.

- *Context monitoring module* – this module is responsible for vehicle surrounding monitoring. It is responsible to monitor information like location of the neighboring vehicles, traffic jam, vehicle's speed, and others. These information are part of the Local Dynamic Map (LDM) functionalities, i.e., the conceptual data store located within an ITS-S as outlined in [91]. Therefore, we aim to rely this monitoring module on such conceptual data store.
- *Flow monitoring module* – this module should inform whether a flow is alive or not and evaluate flows' performance, like the currently used bandwidth, the currently latency, etc.

### 4.2.2.3. Near Future

Due the high vehicle mobility, a connected vehicle changes its network environment constantly. A vehicle running in high speed can cross short-range network (e.g., urban WiFi) rapidly. Therefore, an available access network can be soon unavailable, or a vehicle can rapidly reach new access technologies coverage. In a such dynamic environment, if the DM is capable to anticipate networks conditions, it can perform a more fine-grained decision, as well as, offer a seamless communication. For example, if the DM knows that a network connection will be soon unavailable, it can decide in advance to reroute flows to another access network. Therefore, in dynamic environment, it is desirable a proactive DM mechanism. Such a mechanism should be capable to make decisions based on the near future about the network environment, in which the vehicle will be soon inserted. This part is detailed in Chapter 7.

### 4.2.2.4. Decision making process

The decision making process is responsible to take into account the application's requirements, user profiles, administrative rules (regulation and policies) as well as different monitored information, in order to manage flows communication.

We split the decision making process in three modules: "Hierarchy/Filtering", "Rank Alternative", and "Decision Algorithm", as shown on Figure 4.8. Such a modularity is already an algorithmic choice, in order to take into account the different type of inputs. And it is based on other previous works that have started similar approach, e.g., as [92] and [93]. We describe each one of the three DM modules below:

#### **Hierarchy/Filtering**

This module is responsible to receive and manage requirements, preferences, and policies from different actors. Since actors may have their own specific preferences and requirements, we need to “filter” (in Computer Science acceptance) the various values defined for the same parameter.

Moreover, actors can have contradictory objectives, e.g., an administrator can set an access network as forbidden for a given user, while such user sets the same access network to preferred. In this case, it is necessary to define who has the priority. Therefore, this “Hierarchy/Filtering” module is also responsible to define a priority order between actors, in order to manage contradictory objectives. To address such a contradictory objectives, different hierarchy mechanism can be developed, among which we can highlight the one designed and patented by France Telecom [94]. In this thesis, we do not implement a hierarchy mechanism, instead we consider no contradictory objectives between actors.

Therefore, the output of this module is a list of requirements with their respective values, as outlined on Table 4.9. The column “Requirements” presents the already filtered and hierarchized requirements from all actors, while the column “Values” represents the respective requirement’s value.

Table 4.9: Example of output from the Hierarchy/Filtering module

Requirements	Values
Flow’s required DR	<value>
Flow priority	<value>
Network latency	<value>
Forbidden networks	<value>
Others	<value>

### Rank Alternatives

The *Rank Alternatives* module is responsible to find all alternatives for flow communication. This module is a filter to avoid forbidden networks or networks that do not match with flows’ requirements.

It receives the coherent list of requirements from *Hierarchy/Filtering* module, a set of available access networks and their characteristics (i.e., networks performance) from the *network monitoring* module, and context information, in order to find all potential solutions for each flow, i.e., to find the access networks that meet at least the minimum flow requirements. The output of this module is a list of all potential solutions for each flow, as shown on Table 4.10.



Table 4.10: Example of output from the Ranking Alternative module

Flows	Possible solutions (i.e., access networks)
FlowID 1	NetworkID 1, NetworkID 2, ...
FlowID 2	NetworkID 1, NetworkID 3, NetworkID 4, ...
FlowID 3	NetworkID 2, NetworkID 4, ...
others	others

### Decision Algorithm

This module receives the list of all potential solutions previously created in the *Rank Alternatives* module and apply decision making algorithm in order to evaluate the matching degree of communication requirements with networks characteristics.

As described in 3.2, several decision making algorithms have been used in the network selection process. For example, the ones based on the game theory, the ones based on MOO, and algorithms that uses MADM techniques. The most used are the MADM methods (e.g., SAW, TOPSIS and Analytic Hierarchy Process (AHP)). Despite the MADM techniques present advantage as relative low computation complexity, this approach has some issues. For example, it is very difficult to choose the best weight for each attribute. Moreover, MADM algorithms could present ranking abnormality, i.e., change in one of the parameters of the objective function could determine a very different best solution.

To meet the needs for communication in C-ITS environment, we aim a DM algorithm that present the following properties:

1. management flow by flow – we aim an algorithm capable to manage decisions flow by flow. In this way, the DM can choose the access network that better match the communication requirements for each individual flow;
2. Multiple attributes management – the DM algorithm should manage multiple attributes from different actors (e.g., application requirements, user preferences, administrators and regulators rules);
3. Multiple objective management – the DM algorithm should be capable to manage multiple objectives simultaneously. For example, increase the communication QoS (data rate, latency) while reduce the overall monetary cost;
4. Increase stability – we aim an algorithm that increases the decision stability, avoiding “ping-pong” effect;

5. Avoid full recalculation – the DM algorithm should avoid full recalculation when only few network parameters change;
6. Near future – finally, we aim a DM algorithm capable to take into account forecast information about network environment and vehicle context, in order to make smarter decisions. If the DM algorithm is aware about the context in which the vehicle will be soon inserted, it can, for example, anticipate decisions in order to make flows always best connected.

The existing decision making algorithms do not meet such C-ITS needs. Therefore, we developed a new decision making algorithm (AD4ON algorithm) that is capable to take advantage of the entire proposed architecture. The AD4ON algorithm is described in Chapter 5.

### 4.2.2.5. Applying decision

In the applying decision process, the policies and information produced by the decision making process are applied in the system. In this process, the decision maker can interact with controlled entities in all layers of the ITS-S communication architecture. For example, applications can be informed about network conditions, interfaces can be activated or deactivated according to a given decision, and communication policies can be enforced on different layer in order to manage mobility.

As described in Chapter 3, works have addressed the enforcement of decision making process at different layers of computing system, i.e., it has been applied at application, transport, network or link layer. In order to use simultaneously multiple access technologies and transparently select the most appropriate communication profile to the applications, in this thesis we consider to apply decisions at network layer. We consider to use standardized protocols like NEMO/*FlowBinding* and MCoA. Such protocols have been chosen by several standardization bodies for IP-based mobility management, including ISO and ETSI [2].

As mentioned before, such a protocols create communication tunnels between MR network interfaces and the HA. Therefore, the decision process becomes to choose which tunnel for each flow communication. Using such an approach present some advantages. For example, data are encrypted before be sent over such communication tunnels, increasing the flow communication security. The tunnel approach enable the monitoring process see beyond the network interface. Therefore, it is possible to have information about the entire tunnel performance, i.e., the entire path between the MR and the HA. We can manage queues per tunnel. Moreover, it can be easily implemented in Unix-based systems. Tunnels can be seen as virtual interfaces, making the enforcement process a choice of which network interface to use.

Once the access network that better match the communication requirements is selected, the DM request the *Flow-Interface mapping* module to enforce the flow routing decision. In an ITS-S

(e.g., a vehicle), the DM should be capable to manage both *descending flows*, i.e., flows arriving to vehicle (coming from “Access” layer to upper layers); and *ascending flows*, i.e., flows leaving the vehicle (coming from upper layers to “Access” layer). The present work concentrates its attention for *ascending flows*, while it assumes that simple actions/decisions will be applied for *descending flows*. For example, we assume that replies flows (i.e., replies to descending flows) will be enforced through the same access network and interface used by the descending flow. However, if for any reason the reply needs to be sent through another network interface, it is necessary to inform the HA about this change. In this case, the MR and HA should communicate to determine the tunnel such flow should use.

Since the decision making process takes into account the short-term prevision about the network environment, proactive decisions are enforced in order to maintain flows always best connected. However, unexpected changes can occur in a wireless environment (e.g., a given access network can drops). In order to adapt to the network conditions in real time, the AD4ON maintain an hierarchical solution database with all sub-optimal solutions for each flow, as shown on Figure 4.8. This *hierarchical solutions* database is used by the “Flow-Interface mapping” module in case of emergency, i.e., when the best network solution drops unexpectedly and until the DM finds another better solution.

### 4.3. Security and Privacy

An important subject for connected vehicles is the security and privacy management. The communicating systems must be protected against cyberattacks. Since communicating systems like vehicles react to messages, it must ensure the authenticity of communicating devices, i.e., to ensure that the sender is who is expected to be. It must protect the data traffic, preventing unauthorized person to access the data traffic content. Moreover, it must prevent an unauthorized person to acquire the geolocation of a mobile device.

We consider to use existing security solutions, e.g., using Internet Protocol Security (IPsec) and the data encryption provided by NEMO/MCoA tunnels. Therefore, the security of a given network interface/tunnel is considered as an input for the DM algorithm. In this way, the DM can filter network interfaces that not correspond to a flow requirement. In case where an application requiring security sees its flows rejected, such application can encrypt its flows by itself or reduce the level of its security requirement.

### 4.4. Integration of the AD4ON in the ITS-S communication

## architecture

The ITS-S communication architecture functionalities can be implemented into a single physical unit or distributed into several physical units. On paper [95] we developed a real implementation of an ISO-based architecture into a single physical unit. However, once applied to vehicles, these functionalities can be performed by different modules in the vehicle’s electric/electronic architecture.

Since the AD4ON architecture is designed in an ISO standard compliant way, the different AD4ON functional modules can be implemented in a single OBU or they can be divided in different in-vehicle ECUs. Figure 4.9 shows one way how we can integrate the AD4ON architecture in the ITS-S communication architecture. This integration suggests a centralized implementation, i.e., a single OBU hosting all AD4ON architecture modules.

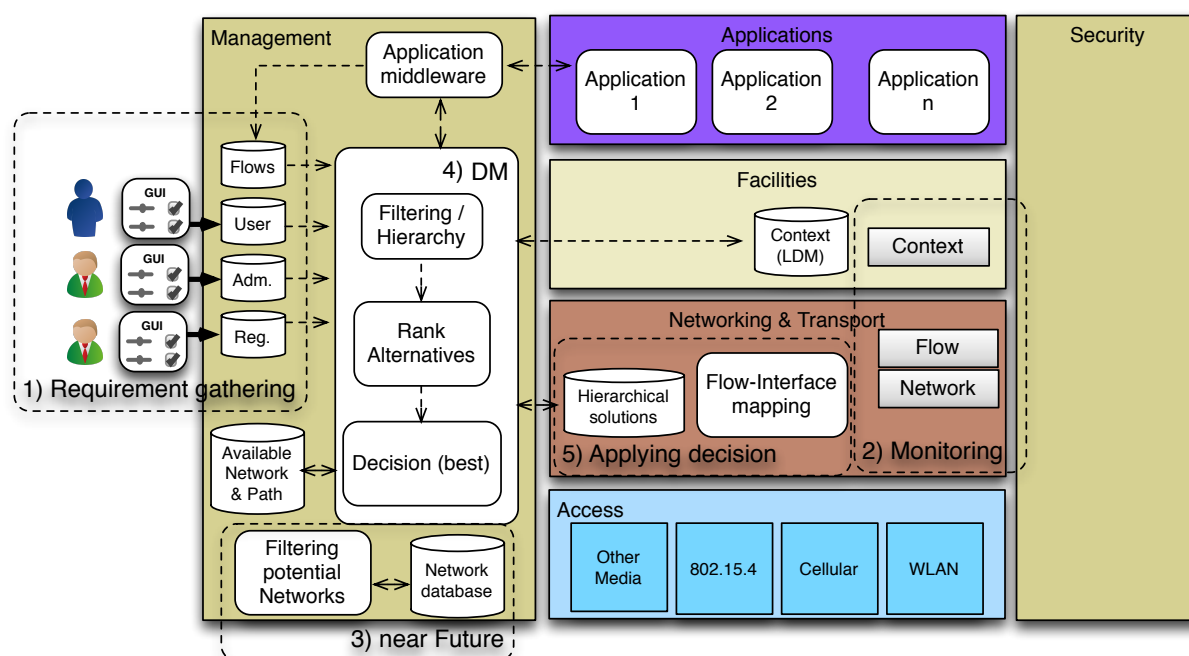


Figure 4.9: Integration of AD4ON Architecture in the ITS-S communication architecture (centralized implementation).

However, the standards only give some guidelines to the developers, leaving some room in the way to implement the ITS-S communication architecture. The NEMO standard mainly separates the applications and communications into two groups:

- MNN – group of nodes in the same mobility network;
- MR – on board device responsible to ensure session continuity for all the nodes in the *Mobile Network*, i.e., for all MNNs.

Therefore, the five functions described in Section 4.2.2 can also be separated into such nodes. For example, the requirement gathering can be implemented in the MNN, the monitoring modules can be implemented in both MR and MNN, while the near future, the decision making process and applying decision are functions of the MR.

## 4.5. Conclusion

The context of C-ITS is characterized by a large heterogeneity of applications, connected devices, network technologies, users and administrators, all interacting with each other. This context presents some communication challenges that are not addressed by actual DM.

Based on different modules and architecture observed in the literature, we identified the main properties for a DM in the C-ITS environment. Therefore, we designed the AD4ON, a modular architecture for opportunistic networking in heterogeneous access network environment.

The AD4ON architecture was developed in an ISO architecture compliant way. And it meets the identified properties for DM in the C-ITS communication. The AD4ON is a modular architecture that can be distributed into several physical units. Therefore, their modules can be implemented in different modules in-vehicle network. It is capable to manage requirements and preferences from different actors (e.g., applications, users, administrators and regulators), it takes into account the short-term prevision about the network environment. Thanks to different monitoring modules, the decision making process can have updated information about networks and the vehicular context (e.g., vehicle speed, battery level). The AD4ON architecture is capable to manage multiple access technologies simultaneously, finding for each flow the access network that better meets communication requirements.

Moreover, the AD4ON architecture is capable to consider forecasted information in order to anticipate decisions. It is capable to receive near future information, i.e., information about the context in which the vehicle will be soon inserted, and based on that make smarter decisions. The use of near future information is addressed in Chapter 7, a dedicated chapter for that.

Once the AD4ON architecture is defined, we aim to develop the AD4ON algorithm, i.e., a DM algorithm capable to take advantage of such entire architecture while makes smart decisions for better C-ITS communication. The AD4ON algorithm is detailed in Chapter 5.

# 5

## AD4ON algorithm

Once the input/output of the AD4ON mechanism, as well as the AD4ON internal architecture are defined, we need to design an algorithm that is capable to take advantage from such architecture, while meet the properties required in a C-ITS environment.

The heterogeneousness of C-ITS communication and the possibility of having multiple applications simultaneously competing for communication resources requires a controlled access to these resources. In this context, a DM algorithm is needed to manage all available access networks simultaneously, attempting to choose for each data flow the access networks that better match with their requirements.

In fact, as seen in Chapter 3, decision making process to choose the best network for data communication have been extensively studied. In the literature, different techniques of decision have been used. They aim to combine requirements, preferences and policies in the decision making process. A large number of such research studies have concentrated on the development of DM algorithms based on MADM methods. Among these algorithms, we can highlight the ones based on TOPSIS, which is the most used in the literature.

Despite the MADM methods present advantages, such an algorithms presents some behaviors that are not suitable for C-ITS communication. For example, they suffer from decision instability, i.e., solutions are very sensitive to small changes of inputs. Moreover, they usually trigger full recalculation at each input change. Therefore, despite such variety of DM algorithms, it lacks an

algorithm capable to cope with the C-ITS communication needs.

We consider we can take advantage of the freedom offered by ISO ITS-S reference architecture, in order to propose a smarter DM algorithm capable to address such a C-ITS needs. Using the inputs, outputs, and the means of enforcement offered by such architecture, we can create an algorithm to opportunistically choose the access network that better meets the communication requirements.

In the present chapter, we identify the main properties for decision making in vehicular environments, which are not addressed by existing DM algorithms. And in order to meet such properties, we develop the AD4ON algorithm, an ACO-based algorithm capable to manage multiple access networks simultaneously, attempting to choose the best access network for each data flow. In other words, we implement the part “4) DM” from the AD4ON architecture described on Chapter 4.

### 5.1. Expected properties

Besides the heterogeneity of applications, communicating devices and access networks, the C-ITS is also characterized by highly dynamic mobility. Vehicles can move at high speed, changing frequently their connectivity context, i.e., network availability and network conditions can vary rapidly.

Such a scenario requires some properties from the DM algorithm. The properties identified in the Chapter 4 for the AD4ON architecture remains valid here for the algorithm part:

- Multiple attributes management – the DM algorithm should be capable to manage multiple attributes simultaneously, which can be presented by different actors (e. g., applications, users, administrators, regulators and networks);
- Management of heterogeneous network – for ubiquitous connectivity, it is necessary to consider all existing wireless technologies, such as vehicular WiFi (ITS-G5 in Europe, or DSRC in North America), urban WiFi (e.g., 802.11 g/n/ac/ax), 802.15.4 and cellular (3G, 4G, and 5G under preparation). Therefore, the DM algorithm should deal with heterogeneous network environment;
- Flow per flow management – in order to make fine-grained decisions, we choose a management flow by flow. Managing flow by flow, allows the DM algorithm to choose the access network that better satisfies flows requirements. Moreover, it enables a better use of each technology, spreading flows among different communication interfaces.

Besides such properties, a DM algorithm in the C-ITS environment requires other properties. When vehicle is moving, it can experience different network conditions. Due its high speed, the connectivity context can change frequently, e. g., an available access network can be soon unavailable, or the vehicle can rapidly reach new access technologies coverage. In such a dynamic scenario, the DM algorithm should deal with the instability of the network environment.

To face such a network environment instability, we consider the DM algorithm should also take into account three main points: increase the decision stability, allow partial recalculation, and take into account the near future prediction about the network conditions. In the following, we describe such three main properties.

### 5.1.1. Decision stability

The C-ITS environment is characterized by high mobility. In such a dynamic scenario, DM algorithms (like TOPSIS) that are based only on objective function (e.g., the higher is the utility value of a decision, the better the solution), tend to make unstable decisions, i. e., small changes in one network parameter can result in a new solution. This high sensitivity to changes of parameter values leads to frequent network switching, especially when a vehicle approaches the limit of range of a given network (where the network can appear or disappear between two steps of calculations). Such a frequent changes of access network can increase the packet loss and the communication latency.

Therefore, once a flow is communicating through a given access network, we aim to privilege such connection as far as the access network is satisfying the communication requirements. However, in some cases flow requirements are not 100 % satisfied and a better access network becomes available, i. e., the vehicle becomes covered by an access network that better satisfy such flow requirements. In this case, we aim an algorithm capable to quickly adapt to this better network condition.

Such a decision is not easy to be performed. The DM algorithm should find the best compromise between a reduced number of network switching and an increased satisfaction of flow requirements.

Moreover, the decision stability, i. e., the decision to remains connected to an access network or switch the flow communication to another one should consider the data queue status. Some networks (as the cellular) develops buffers of data along the transmission chain. In this case, switch quickly to a new network can cause a high number of packet loss.

### 5.1.2. Allow partial recalculation



Due to the vehicle's high speed, the networks availability can change rapidly. Then, the DM algorithm is expected to be frequently called. If the DM algorithm needs to recalculate complete solution for all flows every time a network parameter or flow requirement change, the problem can become NP-complete.

However, only a small subset of the inputs may change between two consecutive runs. In this case, it can be interesting to cache previous results to avoid unnecessary recomputing. Therefore, the DM algorithm should be developed in order to prevent full recalculation when only few network parameters and/or flow requirements change.

### 5.1.3. Take into account the near future

Since the vehicle network environment changes rapidly, we are interested to know the next network conditions as soon as possible. If the DM receives information about the network context in which the vehicle will be soon inserted, it can make smarter decisions, e. g., reducing in advance the data buffers for a given communication that will be soon switched to another network, or delaying a data transmission if it knows that a better network will soon be available. This property is better described in Chapter 7.

## 5.2. The AD4ON Algorithm

In order to address the C-ITS properties described before, we developed the AD4ON algorithm. As already mentioned in Chapter 4, we split the DM process in three functional modules: "Filtering/Hierarchy", "Rank Alternatives", and the "Decision Algorithm". Such division was made to better manage the heterogeneity of inputs. Figure 5.1 shows an overview of the part "4) DM" from the AD4ON architecture described on Chapter 4.

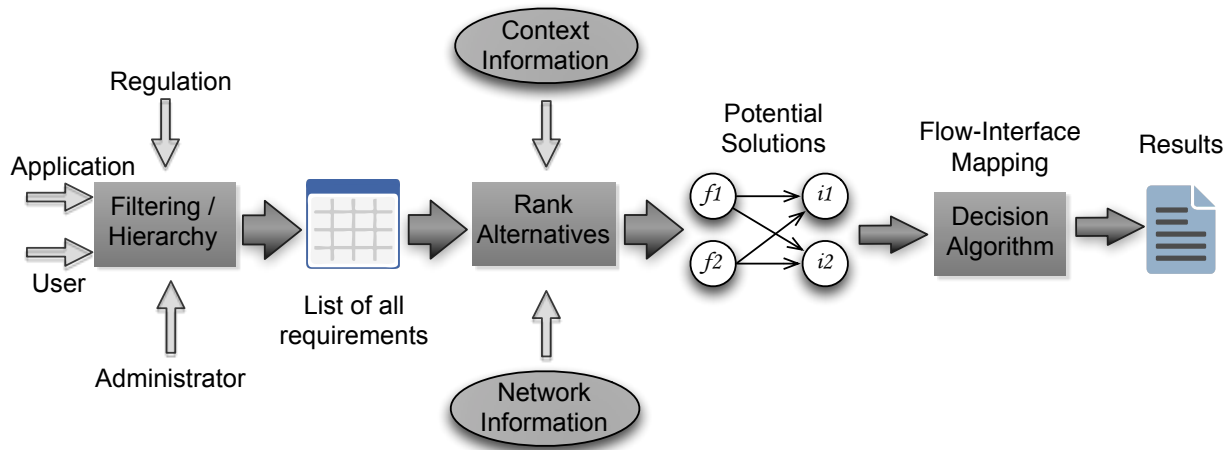


Figure 5.1: Overview of AD4ON DM part

In the following, we describe how we implemented each one of these three functional modules. To a better understanding how the AD4ON algorithm works, we consider a hypothetical scenario as illustrated on Figure 5.2, in which a vehicle moves in a zone covered by three networks (network A, network B and network C).

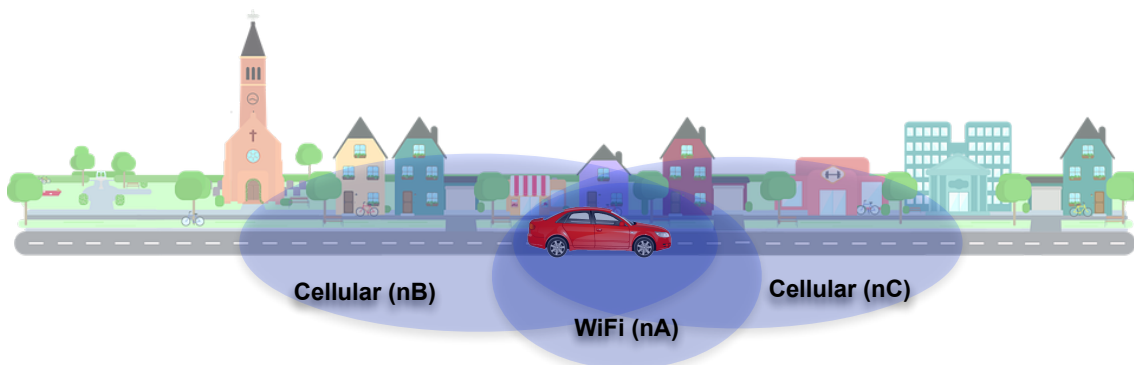


Figure 5.2: An example of vehicle scenario used to illustrate how the AD4ON algorithm works

The performance of each one of these networks is shown on Table 5.1. Such information is supposed to be obtained from the network monitoring module. For example, network A (which is a WiFi vehicular network, i. e., ITS-G5 in Europe or DSRC in North America) has 5 Mbps of available DR, communication through such network experiences a latency of 20 ms, and the packet delivery ratio is 99%. Moreover, the vehicle receives a signal of -45 dBm from the network A.

While the vehicle moves, two onboard applications (each one composed by one flow, i. e., Flow 1

Table 5.1: Network parameters

Network name	Network type	available DR (Mbps)	Latency (ms)	PDR (%)	RSSI (dBm)	Cost (euros/MB)
network A (nA)	WiFi vehicular	5	20	99	-45	0.0
network B (nB)	cellular	10	30	99	-75	0.3
network C (nC)	cellular	15	40	99	-75	0.3

(F1) and Flow 2 (F2), respectively) want to communicate with offboard CN. Therefore, we have two flows competing to communicate through three access networks, as shown in Figure 5.3.

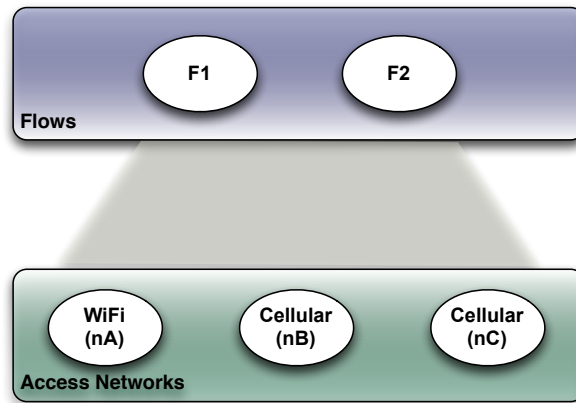


Figure 5.3: Flows competing to communicate through access networks

In this scenario, the AD4ON should choose which network each flow can use, in order to better satisfy the communication requirements, i.e., satisfy the flows requirements, user preferences and administrators rules. The flows requirements are shown on Table 5.2, in which  $minDR$  represents the minimum data rate required to start the flow communication,  $maxDR$  is the maximum value of DR for which the flow can improve its quality,  $minLatency$  is the minimum latency sensibility for a given flow, i. e., is the minimum latency value below which flow does not present quality improvements,  $maxLatency$  represents the maximum latency supported by a flow, finally  $minPDR$  represents the minimum required PDR to start the communication. For example, Flow 1 requires a minimum DR of 3 Mbps to start to communicate (i. e., to provide a minimum of QoS) and it can consume until 10 Mbps to provide the maximum QoS. In terms of latency, Flow 1 can support maximum 35 ms of delay, while the ideal delay is less or equal to 10 ms. Flow 1 is packet loss tolerant, requiring a minimum of 90% of PDR.

For this example, we consider the only input from the user is the amount of money he or she is willing to pay to communicate. Therefore, we assume user is willing to pay maximum 0.4 euros per Mbytes. Similar to the user input, we consider only one input from administrator (or regulator): the list of forbidden networks. As described in the AD4ON architecture chapter (Chapter 4), forbidden networks is given by flows. In the current development, such a restriction is

Table 5.2: Flow requirements

Flow name	minDR (Mbps)	maxDR (Mbps)	minLatency (ms)	maxLatency (ms)	minPDR (%)
Flow 1	3	10	10	35	90
Flow 2	2	12	20	1000	80

presented by network type, i. e., by type of technology. In this way, instead of list all network access point in which a given flow is forbidden, the administrator configure the interdiction by type of technology, as shown on Table 5.3. In this example, we assume the Flow 2 is forbidden to communicate through WiFi access technologies, while flow 1 has no network restriction.

Table 5.3: Network parameters

Flow	forbidden networks
Flow 2	type: WiFi vehicular

### 5.2.1. Filtering/Hierarchy

As already mentioned before, we assume that all ITS-S have communication capabilities for all type of flows running on it. Therefore, the AD4ON mechanism does not perform the communication profile management. Instead, we consider predefined *FlowTypeID*. Therefore, once a flow starts the communication process, first of all it sends its communication requirements (as described on Table 5.2) to the AD4ON mechanism. The “Filtering/Hierarchy” module receives such inputs as the ones from user and administrator, in order to create a coherent list of inputs. For the sake of simplicity, in this example the inputs are not contradictory. Therefore, this module only concatenates the requirements from the different actors and sends them to the *Rank Alternatives* module. However, in case two actors present contradictory requirements, such a inputs should be filtered/hierarchized. In this case, we can base on work developed by [88] in order to perform this filtering.

Table 5.4 shows the output of the “Filtering/Hierarchy” module for our example.

### 5.2.2. Rank Alternatives

The "Rank alternatives" module is responsible to find possible solutions for each flow. With the list of communication requirements from the *Filtering/Hierarchy* module, and other information like network performance (e.g., DR, latency, PDR) and vehicle context (e.g., vehicle speed, battery level), the *Rank Alternatives* module finds all potential solutions, i.e., it finds for each flow, the

Table 5.4: Output from the Hierarchy/Filtering module

Requirements	Flow 1	Flow 2
minDR (Mbps)	3	2
maxDR (Mbps)	10	12
minLatency (ms)	10	20
maxLatency (ms)	35	1000
minPDR (%)	90	80
cost (euros/MB)	0.4	0.4
forbidden networks		WiFi vehicular

access networks that meet at least the minimum flow requirements. Access network that does not meet the minimum requirement for one communication parameter (e.g., does not have enough DR or present a latency not supported by a flow) and/or the access networks forbidden by the administrators are filtered, i.e., they are not listed as potential solution.

In our example, the “Rank Alternatives” receives as input:

- Communication requirements - such information is the list of coherent communication requirements created by the “Filtering/Hierarchy” module, i.e., the information listed in Table 5.4;
- Network information - monitoring modules are in charge to acquire and send information about the network performance to the “Rank Alternatives” module. Such information are the ones showed in Table 5.1;
- Context information - in this example, we do not consider context information. However, this can be easily implemented in this module. For example, not considering short-range networks when the vehicle is moving at high speed.

Based on these information, the “Rank Alternatives” module creates a list of all potential solution for each flow, as shown on Table 5.5. As we can observe, the latency performed in network C (40 ms) does not meet the maximum latency tolerated by Flow 1 (35 ms). Therefore, network C is not considered as a potential solution for Flow 1. Moreover, according to the administrator rules, Flow 2 is forbidden on network A, therefore such network is not considered as solution for Flow 2. The output form "Rank alternatives" module is the list of potential solution shown on Table5.5.

Table 5.5: Output from the Ranking Alternative module

Flows	Possible solutions (i.e., access networks)
Flow 1	Network A, Network B
Flow 2	Network B, Network C

### 5.2.3. Decision Algorithm

The *Decision Algorithm* module receives the “Rank Alternatives” output and finds solution for each flow, i. e., it chooses the access network that better satisfy the communication requirements.

In the following sections, we describe the AD4ON algorithm responsible to make such a decisions. The AD4ON algorithm is based on the ACO, a swarm intelligence class of algorithms based on the collective and cooperative behavior of ants. This class of algorithm is capable to find high-quality solutions for complex combinatorial optimization problems in a reasonable time.

### 5.2.4. Why ACO-based algorithm

As observed in the state of the art, there are different DM algorithms that can be used for network selection. We can highlight the MADM techniques (like SAW and TOPSIS), the Artificial Intelligence (AI) algorithms like machine learning, and the ACO-based algorithms.

The MADM techniques do not meet the requirements we identified for C-ITS environment. They are not designed to naturally work with multiple flows simultaneously. To overcome this issue, current TOPSIS-based DM algorithms run separated instances for each flow. In this way, the order the algorithm considers flow can influence the final result. MADM algorithms perform full recalculation every time the communication system change, i. e., every time the flow requirements or network parameters changes. Moreover, these algorithms suffer of rank abnormality, i. e., few changes in the parameter of a potential solution (e. g., network parameter) can generate a new solution totally different from the previous one.

AI algorithms like machine learning algorithms relies on patterns from set of sample data (i. e., training data), in order to make decisions. For better solutions for such an algorithms, it is usually necessary a huge amount of training data with enough variety of scenarios, covering different use case.

Analyzing the ACO algorithm, we identified properties in this class of algorithm that can be explored to meet the identified C-ITS needs. Therefore, we choose this class of algorithm to develop the AD4ON. Below, we highlight these ACO properties.

- Allow management of multiple nodes simultaneously: This class of algorithms is designed to solve problems that can be represented by a graph, where a possible solution is a specific path in the graph. In this way, the C-ITS communication problem can be modeled as a graph, where flows and access networks correspond to the graph nodes and the edges are potential solutions for flows communication. Such a representation enables the AD4ON algorithm to consider multiple flows and multiple access networks simultaneously, as well as to manage communication flow per flow. This graph representation is described on Section 5.2.5.

- **Decision stability:** Ants deposit pheromone based on the solution qualities, i.e., depositing more pheromone on better solutions, while bad solutions tend to extinguish their pheromone by evaporation. Since decisions are driven by pheromone concentration, ACO solutions are created smoothly over time. For example, once a better access network become available, decision is not immediately switched through the new solution. Instead, ants tend to migrate from the previous solution through the new one over time. The AD4ON algorithm use such behavior to filter transient effects and thus offers a better decision stability.
- **Memory-based decisions:** ACO are memory-based algorithms. A new solution takes into account previous status of the graph, by means of the amount of pheromone in each potential solution, i.e., pheromone concentration in each specific path in the graph. Based on such ACO property, the AD4ON algorithm can take into account previous status of the network environment, in order to smoothly switch from one access network for another.
- **Run-time adaptable:** Once the graph condition change, e. g., a node is no more present or some inputs change their values, ants tend to adapt to this new condition by finding new solutions. Such property to react efficiently to graph change is useful in a highly dynamic scenario like the C-ITS one.

### 5.2.5. Problem representation

An important step in the process of problem solution is the correct understanding of what the problem is. Once the problem is defined, it is possible to apply the best method to solve them. In our access network selection problem, we define it by a graphical representation, which is stated as follow.

Given a set  $F$  of  $n$  flows ( $F = \{f_1, f_2, \dots, f_n\}$ ) simultaneously competing for communication resources, and a set  $N$  of  $m$  available access networks ( $N = \{N_1, N_2, \dots, N_m\}$ ), we aim to choose the access network that better match the communication requirements for each flow.

We modeled our *flow to network assignment* problem as a bipartite graph  $G(F, N, E)$  (as shown on Figure 5.4), where  $F$  correspond to data flows,  $N$  correspond to available networks and  $E$  is the union between the sets  $F$  and  $N$ , if flows in  $F$  can be assigned to networks in  $N$ , i.e.,  $E = \{function : i \mapsto j \mid i \in F, j \in N\}$ .

The AD4ON takes into account requirements and preferences from different actors (e.g., applications, users, administrators and regulators), as well as information about access networks conditions (e.g., data rate, latency) in order to construct this graph.

### 5.2.6. Utility Functions

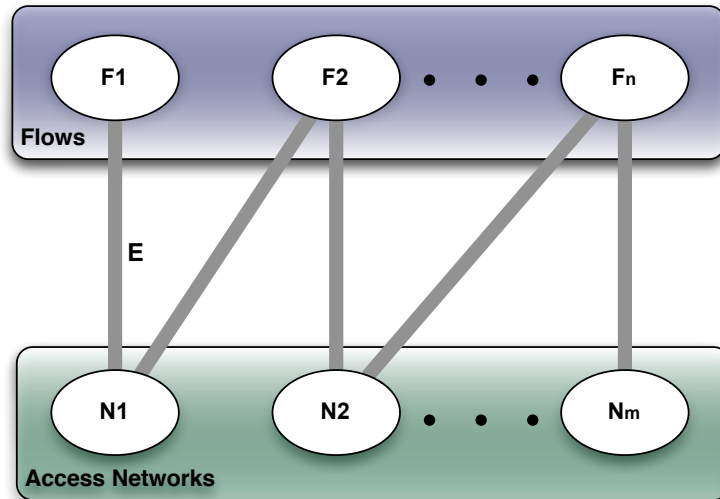


Figure 5.4: Graph of feasible solutions

As stated before, we aim to choose the best access network for each flow. However, it is necessary to define what “the best” means, i.e., which attributes characterizing an access network we consider as criteria to judge if a network is good or not.

Based on the literature review, we choose the five attributes most used in recent researches works, in order to perform our decision making process. In this thesis, we consider the following five attributes to be optimized:

- DR – data rate required by each flow;
- Latency – the delay supported by each flow;
- PDR – the packet delivery ratio required by flows;
- RSSI – signal level presented by the access networks;
- monetary cost – is the maximum amount of money an user is willing to pay, in order to send data.

Therefore, the best access network is the one that better meet flow requirements (i. e., DR, Latency and PDR), present a good RSSI and reduce the monetary cost to send data. It is an optimization problem, in which we aim to maximize DR, PDR, RSSI and minimize the latency and the monetary cost.

Since such attributes have different units, it is necessary to normalize them before start the optimization process. As described by [96], data normalization is essential for decision making problems. Since attributes are collected from different sources (i. e., different actors) and from heterogeneous data measurements, it is necessary a normalization process that convert all the criteria values into non-dimensional form.



According to authors of papers [79, 66, 97], one of the causes of rank abnormality in MADM techniques, more specifically in TOPSIS algorithm, is related to the normalization technique such algorithm uses (i. e., vector normalization).

Therefore, the choice of the normalization process is important. We aim a normalization technique that takes into account the attribute type, maintaining the normalized value truly representing the source data, i. e., we want to maintain the nature of the attribute when normalizing it. For example, if a given attribute is constant, we need a constant normalized value independently if other attributes changes.

Based on [66], we tackled the normalization of attributes values using utility functions. In this case, we select a suitable utility function for each attribute, i. e., we select functions that better describe the nature of the attribute. Besides allowing to choose function that better match with attributes behavior, using separated utility functions for each attribute prevent that changes in one attribute affects the normalization of others.

In networks, as the resource rate of some attributes increases, depending on attribute its utility can be seen differently by flows. For attributes like DR, PDR, and RSSI, as such network resource rate increases, the flow requirement is better satisfied, and consequently the utility of these attributes increases. For example, the higher the DR the better for the flow. However, flows usually use a finite threshold. They need a minimum network resource to start its communication and a maximum resource from which no more improvement are observed in the communication performance. Therefore, if the rate continues to rise, at some point the flow requirement will be totally satisfied, and anything above such limit will simply be inaccessible by the flow. At that point, the attribute reaches the maximum total utility.

In the other hand, for attributes like latency, flows utility decreases as such attributes' value increases. Flows usually have different sensibilities to latency. They usually have a minimum latency value that is perceived by the application, and any value below that will not increase the communication performance. If latency continues to rise, the utility of this attribute for the flow starts to decrease. And at some point the utility for that attribute may fall to zero.

In order to better meet flow requirements, we defined utility function for each attribute. In the following, we describe the utility function for each one of the five attributes, i. e., the utility function for DR, PDR, RSSI, latency, and the monetary utility.

### **Data rate utility function (uDR)**

In terms of DR requirement, we assume that all flows will send two data rate threshold to the AD4ON. The minimum data rate required to start the flow communication (*minDR*), and the maximum data rate (*maxDR*), i.e., the upper limit until which flow can improve its quality. Beyond

such upper limit, we consider there are no quality improvements for the flow. Such an information come from the flows through the MA-SAP interfaces, as described in section 4.2. The data rate utility function is defined by Equation 5.1.

$$u_{DR} = \begin{cases} 0 & \text{if } x < \text{minDR} \\ \left( \frac{0.9(x-\text{maxDR})}{\text{maxDR}-\text{minDR}} + 1.0 \right) & \text{if } \text{minDR} \leq x < \text{maxDR} \\ 1 & \text{if } x \geq \text{maxDR} \end{cases} \quad (5.1)$$

### Latency utility function (uL)

As the DR, we consider that all flows inform two latency threshold. The maximum acceptable latency (*maxLatency*), i.e., the upper limit beyond which flows cannot communicate properly; and the minimum perceptible latency (*minLatency*), i.e., the minimum latency value below which flow does not present quality improvements.

The latency utility function is defined by Equation 5.2.

$$uL = \begin{cases} 1 & \text{if } x \leq \text{minLatency} \\ \left( \frac{-0.9(x-\text{minLatency})}{\text{maxLatency}-\text{minLatency}} + 1.0 \right) & \text{if } \text{minLatency} < x \leq \text{maxLatency} \\ 0 & \text{if } x > \text{maxLatency} \end{cases} \quad (5.2)$$

### PDR utility function (uPDR)

PDR is the ratio between the number of successfully received packets to the total number of packets sent by sender. The higher is the PDR, better is the flow QoS, reaching the maximum utility when there is no packet loss, i.e., all sent packets are received. Flows have different level of tolerance for data loss. Therefore, since the maximum PDR (*maxPDR*) is 100%, we consider that flows indicate only the minimum required PDR (*minPDR*) to start the communication.

The PDR utility function is defined by Equation 5.1.

### Monetary utility function (uMC)

Users can define the maximum monetary cost they are willing to pay for a data communication. We defined the monetary utility function as inversely proportional to the monetary cost. Therefore, less is the monetary cost to transmit data in a given network, better is the monetary utility. The

maximum monetary utility is reached for free access network, i.e., when the monetary cost is zero. Similarly, the monetary utility reaches the minimum value (i.e., zero) when the monetary cost is greater than the maximum user defined cost (*maxCost*).

The monetary utility function is defined by Equation 5.3.

$$u_{MC} = \begin{cases} \left( \frac{-0.9x}{maxCost} + 1.0 \right) & \text{if } 0 < \text{current monetary cost} \leq maxCost \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

### RSS utility function (uRSS)

Communicating nodes should receive enough power (RSSI) from access network antennas, in order to enable data transmission. Usually, higher is the RSSI, better is the communication performance (e.g., in terms of data rate, latency and PDR). Therefore, we defined the Received Signal Strength (RSS) utility function as described in Equation 5.1.

However, access technologies do not work in the same RSSI range. For example, cellular antennas (e.g., 3G, 4G) can transmit data with lower signal strength than urban WiFi (e.g., 802.11ac,n). For this reason, the AD4ON has stored in its network database, the minimum required RSSI (*minRSS*), as well as the maximum RSSI (*maxRSS*) defined for each access technology. Based on such values and based on the current RSSI received from the network monitoring module, the AD4ON calculates the RSSI utility value from Equation 5.1.

### 5.2.7. General description of the AD4ON algorithm

The AD4ON algorithm receives all potential solutions from the “Rank Alternatives” module, as showed in Table 5.5. Such information represent a graph, like the one shown on Figure 5.5, in which the edges (in gray) represent feasible solutions and the dashed red line represents the forbidden access network for the Flow 2.

Since the AD4ON is an ACO-based algorithm, it simulates the process of ants searching for food, as described in section 3.2.2. Figure 5.6 illustrates all the AD4ON decision making process for our example of two flows (F1 and F2) competing to communicate through the three access networks (nA, nB, and nC).

Flows are seen as the colony nest and the networks are seen as the source of food. Ants located in a given flow (nest) try to find the best access network (source of food) among all potential solution. For example, in Figure 5.6a ants in Flow 1 can choose between Network A and Network B, while ants in Flow 2 can choose between Network B and Network C.

However, we can not find isolated solutions for each Flow. Otherwise, we risk to find unfeasible

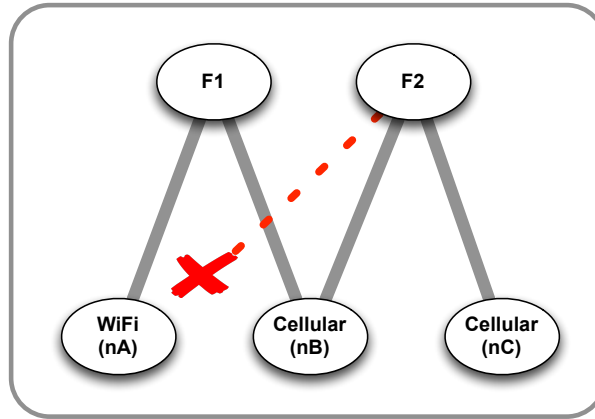
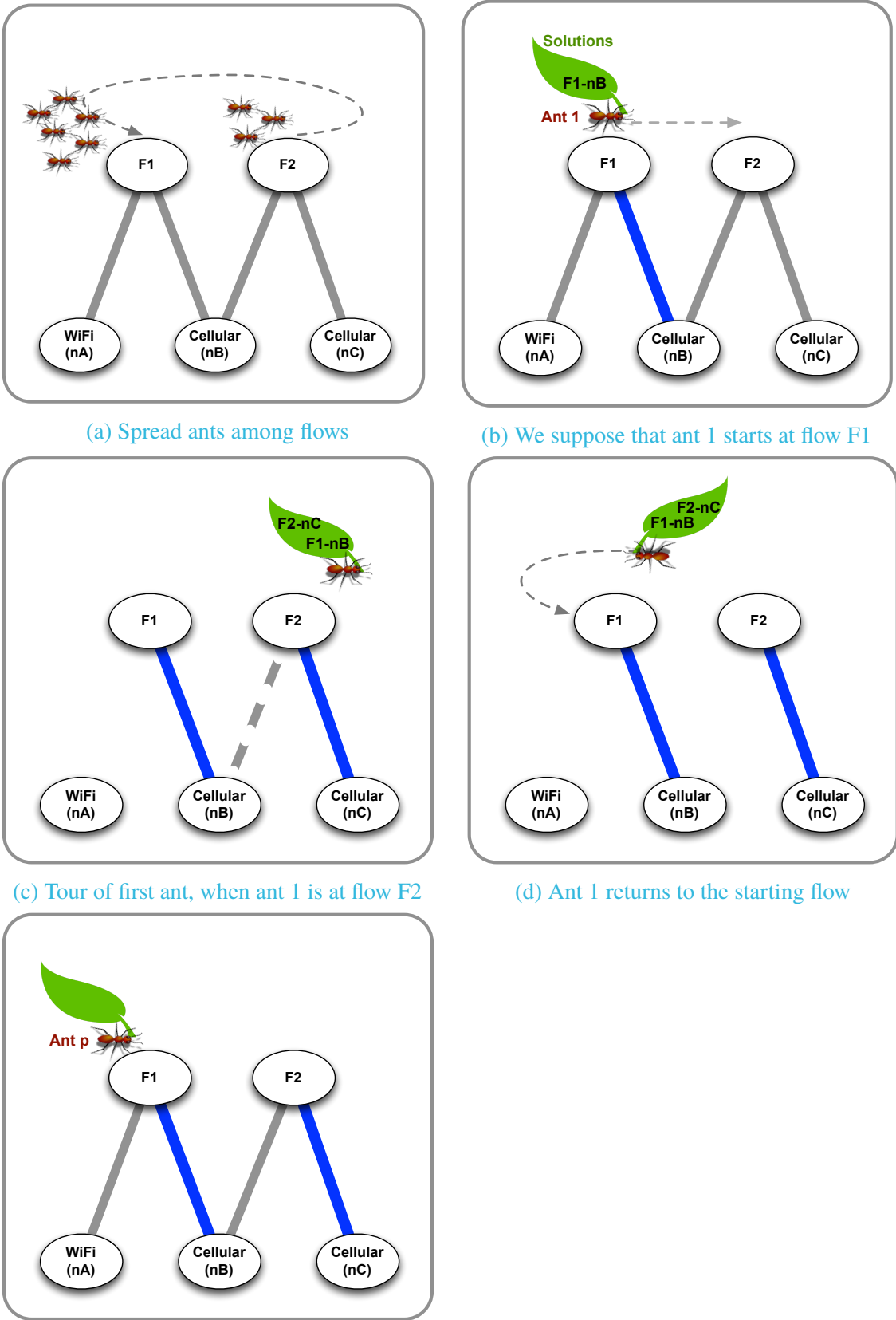


Figure 5.5: Potential solutions

solutions, e. g., Flow 1 and Flow 2 each one using 10 Mbps of DR, and trying to communicate through Network B. To prevent such an issue, each ant is supposed to find a complete solution for the graph, i. e., each ant visits all flows, while try to find the best access network for each flow. Once an access network is chosen by an ant, the capacity of such access network is updated from the point of view of the current ant.

A complete description of the AD4ON algorithm is given in Algorithm 1 and it is discussed below.

- a) First, we set the values of parameters  $\alpha$ ,  $\beta$  and  $\rho$  that respectively determine the relative influence of the pheromone trail, the heuristic information and the evaporation coefficient of pheromone. We also initialize the “PS” and “FL” variables that will respectively store the found solutions and the list of flows (lines 1 - 4).
- b) The DM receives the graph  $G(F, N, E)$  that represents all possible solutions (line 5). Since ants use pheromone and heuristic values to probabilistically construct solutions, we should initialize pheromone trails on the graph. We defined the initial pheromone ( $\tau_0$ ) between a flow  $i$  and network  $j$  as the amount of pheromone deposited by one ant when choosing network  $j$  for the flow  $i$ .
- c) At the begin, the AD4ON algorithm spreads the ants randomly through all flows, as shown in Figure 5.6a.
- d) Once the ants are distributed, they start to construct solutions by exploring the graph. For each visited flow, ant chooses one network among all possible networks, i.e., a path in the graph  $G$  between current flow and potential networks (line 11). In our example, we suppose that *ant 1* starts at flow F1, as depicted in Figure 5.6b. Assuming that *ant 1* chooses the network B to supply the 10 Mbps of DR required by the flow F1, it stores this solution in its memory and it walks to the next flow, i. e., F2 in this example. At the flow F2, *ant 1* is



(a) Spread ants among flows

(b) We suppose that ant 1 starts at flow F1

(c) Tour of first ant, when ant 1 is at flow F2

(d) Ant 1 returns to the starting flow

(e) The next ants tours consider the pheromone deposited by previous ants

Figure 5.6: Example of the AD4ON decision making process

aware that Network B has no capacity anymore (like showed in Figure 5.6c). And the only solution for the flow F2 is the network C. *Ant 1* chooses network C for flow F2 and returns to its starting point, i. e., flow F1 (Figure 5.6d).

The probability ( $P_{i,j}$ ) for an ant to choose the path  $i, j$ , i.e., the path between flow  $i$  and network  $j$  is given by Equation 5.4.

$$P_{i,j} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in V_i} [\tau_{ik}]^\alpha [\eta_{ik}]^\beta} \quad (5.4)$$

where  $\tau_{ij}$  is the amount of pheromone present between flow  $i$  and network  $j$ ,  $V_i$  is the set of available networks for the flow  $i$ ,  $\eta_{ij}$  is the heuristic information and it is given by Equation 5.5.

$$\eta_{ij} = \sum_{n=1}^N (w_n * uP_{n(i,j)}) \quad (5.5)$$

---

**Algorithm 1:** ACO algorithm
 

---

```

1 Set values of ACO parameters (e.g.,  $\alpha$ ,  $\beta$  and  $\rho$ )
2 PS = null ; // Initialize Pareto Set (PS) as empty
3 SS = null ; // Initialize Secondary Solutions (SS) as empty
4 FL  $\leftarrow$  list of flows
5  $G(F, N, E) \leftarrow$  "Rank Alternatives" module
6 while stop condition do
7   for  $k = 1 \rightarrow$  NumberOfAnts do
8     /* Construct a solution */
9     Sort the flow list FL
10    while remains not visited flow in FL do
11      for each possible network for such flow do
12        calculate the probability of choosing that network according to Equation 5.4
13      end
14      choose the network to be mapped
15    end
16    /* Evaporation */
17    apply the pheromone updating according to Equation 5.6
18    /* Evaluation */
19    Calculate the value of objective function for each solution in current ant population
20    (Equation 5.7)
21    Update the Pareto set solutions (PS)
22  end
23 Return the Pareto Set PS

```

---

where  $uP_{n(i,j)}$  is the utility function of a given parameter  $n$  (e.g., data rate, latency, monetary cost) between the flow  $i$  and the network  $j$ ,  $N$  is the number of decision parameters, and  $w_n$  is the weight of each parameter, such that  $\sum w_n = 1$ .

- e) In order to better meet flow requirements, a suitable utility function was defined for each parameter (i.e.,  $uP_n$ ). Such utility functions are the ones previously described on section 5.2.6.
- f) The process of solution construction realized by each ant terminates once the ant has visited all flows on the graph.
- g) Once an ant found a complete solution, i.e., an access network for each flow from the graph, such ant update the pheromone table (line 16). Besides the pheromone deposition, it is necessary to apply a pheromone evaporation rule. Such pheromone evaporation prevents the convergence of the ACO algorithm to a locally optimum solution while enables ants to “forget” low quality solutions. The pheromone update mechanism is given by Equation 5.6.

$$\tau_{i,j}(t+1) = (1 - \rho)\tau_{i,j}(t) + \sum_{k=1}^m \Delta\tau_{i,j}^k \quad (5.6)$$

where  $\rho$  is the pheromone evaporation rate ( $0 < \rho < 1$ ),  $m$  is the number of ants and  $\Delta\tau_{i,j}^k$  is the amount of pheromone deposited by ant  $k$  on the edge  $i, j$ .

- h) After all ants have visited the graph  $G$ , we should evaluate the found solutions. In decision making, utility refers to the satisfaction that a solution provides to the decision maker. Therefore, we propose an utility function that calculates a score representing the matching degree of each solution in the current ant colony (line 17). The utility function is defined by Equation 5.7.

$$U = \sum_{f=1}^F \eta_f \quad (5.7)$$

where  $\eta_f$  is the heuristic information between a given flow and its network solution.  $F$  is the number of flows in the graph  $G$ .

- i) Each solution not dominated by both other solutions in the current colony and the non-dominated solutions already in the Pareto set PS, should be added to PS. And all solutions dominated by the added one should be transferred from PS to SS, i. e., set of secondary solutions (line 18). Then all decision process restart from the line 6. In the next tours of the AD4ON algorithm, ants take into account the amount of pheromone deposited during the

previous tours. Figure 5.6e illustrates an *ant p* starting its tour from flow F1. This ant is influenced by the pheromones deposited in the edges F1-nB and F2-nC. The AD4ON stop condition is described below, in section 5.3.

As described before, the output of the AD4ON algorithm is a list of solutions containing both the primary solution and the secondary solutions (if it exists) for each flow. Such solutions are arranged in descending order of utility. The primary solution is the better solution found by ants in the graph, i. e., the access network with higher utility. The secondary solutions are access networks that leave communication requirements less satisfied.

Considering the hypothetical scenario illustrated on Figure 5.2, the output of the AD4ON algorithm is the one shown in Table 5.6, i. e., the better solution for Flow 1 is the network B, while network A is considered as secondary solution. Similar, the better solution for Flow 2 is the network C, while the network B is a sub-optimal solution.

Table 5.6: Output of the AD4ON algorithm

Flows	Solutions (i.e., access networks)
Flow 1	Network B, Network A
Flow 2	Network C, Network B

### 5.3. Stop condition of the AD4ON algorithm

Due the nature of the ACO-based algorithms, at the begin of the AD4ON execution ants tend to explore different solutions. This exploratory behavior can result in feasible solutions that are not necessarily the most optimized for the problem. Over the time, ants tend to converge to the optimized solution, i. e., they tend to converge through the best solution. Although the ant's choice is probabilistic (as shown by Equation 5.4), over the time ants tend to choose the solution with higher pheromone concentration. Therefore, ants will always reinforce the pheromone in the optimized solution, such that after the convergence step few improvements can be performed by ants, in terms of solution optimization.

In such context, it is necessary to define if the AD4ON will run continuously or it will stop after ants are converged through the optimized solution. We defined two AD4ON operating modes related with the stop condition:

- Stop execution – in this mode, we first discretize the vehicle movement and we execute the AD4ON for each discretized instant. We can discretize the vehicle movement based on



elapsed time or travelled distance. For example, we can define to run the AD4ON algorithm at every 1 second, or every time the vehicle has a displacement of 50 meters.

For each discretized instant we determine a stop condition, i. e., a maximum number of tours each ant will perform. This mode requires knowledge about the scenario, in order to define the good discretization step. For example, a scenario with high network changes requires more discretized instances than a scenario with low network changes. For this reason, this mode is suitable for simulation process, where we know in advance the scenario to be simulated. This mode will be used in our simulations described in the next chapter.

- Continuous execution – Due the high mobility of vehicles, their network environment can change frequently, changing the inputs for the AD4ON. For example, an access network can change its parameter values, the graph of potential solution can change (an access network can disconnect or a new access network becomes available), new flows can start to communicate, or users can change their preferences. In such a scenario, we cannot preset a number of tour each ant should perform. Otherwise, we risk to miss important environment changes between two AD4ON executions. To address such a dynamism, we designed the continuous execution mode. In this mode, the AD4ON algorithm runs permanently with two different calculation modes: normal mode or reduced mode. In the normal mode, all ants participate in the decision process. This mode is executed, for example, when the AD4ON starts the decision process for a given graph. In the other hand, if input does not change for some time, the ants tend to converge through the best solution in the graph. Consequently, few improvements can be performed by the ants. In this case, the AD4ON is switched to the reduced mode. The reduced mode reduces the MR usage (e. g., CPU and memory usage) by limiting the number of ants participating in the decision process and slowing down the calculation speed. Such a slowing down process can be easily implemented, e. g., setting up a time sleep in the AD4ON process.

Since the AD4ON runs permanently, we designed a mechanism of client-server that enables solutions be requested anytime of the decision process. However, due the nature of the ACO-based algorithms, at the begin ants tend to explore different solutions, and over the time ants tend to converge through the best solution. Therefore, if we request a solution at the beginning (e. g., the first solution found by the first ant), such solution can be far from the best solution. And as the AD4ON runs, solutions become closer to the best solution. Therefore, besides the solution, it is necessary to inform the level of confidence of such solution. We defined an Index of Convergence (IC), i. e., an index showing the level of ants convergence. Then, the AD4ON gives the solution followed by such IC.

We defined the IC as the dispersion of cost encountered by ants in relation to the mean. This

dispersion is measured by the Coefficient Variation ( $CV$ ), which is often expressed in percentage and defined by literature as the ratio of standard deviation ( $\sigma$ ) to the mean ( $\mu$ ). Therefore, the  $IC$  is given by Equation 5.8,

$$IC = \frac{\sigma}{\mu} * 100 \quad (5.8)$$

From Equation 5.8, we can conclude that as the ants converge, i. e., the standard deviation of different ants ( $\sigma$ ) tend to zero, the  $IC$  tend to zero.

## 5.4. AD4ON's parameter setting

In ACO-based algorithms, the problem is represented by a graph and ants concurrently build solutions by exploring such graph. In the AD4ON algorithm, the network selection problem is represented by a graph bipartite, as shown on Figure 5.4. At each construction step, ant  $k$  (in a given flow) applies a probabilistic action choice rule, called *random proportional* rule, to decide which network to visit, i.e., which access network to select.

As showed by Equation 5.4, the probability of the ant  $k$  currently at flow  $i$ , chooses the network  $j$  is dependent of both an heuristic part and an historic-based part, i.e., based on the pheromone trail, as shown on Figure 5.7.

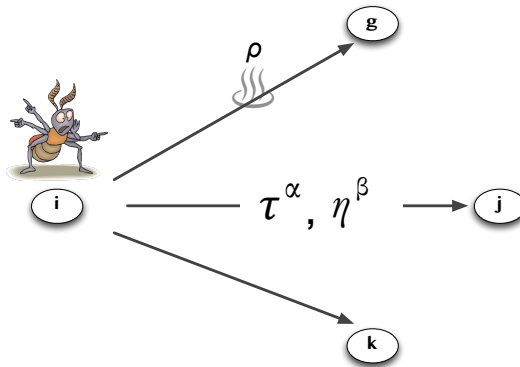


Figure 5.7: An ant in flow  $i$  chooses the access network as a function of the pheromone values  $\tau$  and the heuristic values  $\eta$  on the arcs connecting flow  $i$  to the access network  $j$ .

The influence of these two parts (i. e., heuristic and historic parts) are controlled by  $\alpha$  that determines the relative influence of the pheromone trail, and  $\beta$  that determines the relative influence of the heuristic information. Moreover, an evaporation rule prevents the convergence of the AD4ON algorithm to a locally optimum solution, while enables ants to “forget” low quality solutions. The amount of pheromone evaporation is controlled by a coefficient  $\rho$ .

Such coefficients play an important role in the solution quality. If  $\alpha = 0$ , i.e., without pheromone influence, the network with the higher utility value is more likely to be selected. This corresponds to a classic MADM algorithm like the TOPSIS. If  $\beta = 0$ , only pheromone is used without any heuristic bias. This generally leads to rather poor results, i. e., to rapid stagnation situation, in which all the ants follow the same path in the graph and construct the same solution.

Therefore, in order to define the value of such coefficients, we performed simulation of the AD4ON algorithm with different coefficient values ( $\alpha$ ,  $\beta$ , etc), and we chose the values that gave better results for the simulated scenarios. Selected parameter values are showed on Table 5.7.

Table 5.7: AD4ON parameters

Parameter	Values	Description
$\alpha$	2.0	influence of the pheromone trail
$\beta$	3.0	influence of the heuristic information
$\rho$	0.3	coefficient of pheromone evaporation
ants	10	number of ants in the colony
iterations	50	stop condition in the Algorithm 1

### 5.4.1. Properties achieved by the AD4ON algorithm

As described in Section 5.1, a dynamic and heterogeneous environment like the C-ITS, requires some properties from the DM algorithm. In the following, we describe how the AD4ON algorithm address such expected properties:

- Multiple attributes management – the AD4ON algorithm is capable to manage multiple attributes simultaneously. Through the “Filtering/Hierarchy” module, it is capable to take into account heterogeneous attributes from different actors (e. g., applications, users, administrators, regulators and networks);
- Management of heterogeneous network – the AD4ON algorithm deal with heterogeneous network environment. Thanks to the “Rank Alternatives” module, the AD4ON algorithm can combine the communication requirements with the available access networks in a heterogeneous network environment;
- Flow per flow management – in order to make fine-grained decisions, we choose a management flow by flow. The way in which the optimization problem was represented, enables the AD4ON algorithm to choose the access network that better satisfies flows requirements. In Chapter 6 we can see the AD4ON algorithm increasing the flows satisfaction;

- Decision stability – since ants find solutions based on the amount of pheromone concentration, solutions are created smoothly over time. If a new and better access network becomes available, ants start to migrate towards the new access network. This smooth process increases decisions stability and prevents “ping-pong” effect. Such a decision stability can be confirmed by the evaluations performed in the next chapter (Chapter 6);
- Allow partial recalculation – in the AD4ON algorithm, the normalization of each attributes is performed by independent utility functions. Therefore, the normalization of a given attribute does not interfere in others already normalized. In this way, if a given attribute value changes, it is not necessary to recalculate all normalization process, like MADM TOPSIS does. We can normalize only the new value, thus reducing the calculation process. Moreover, due the possibility of request solution to the AD4ON at anytime (like described on Section 5.3), it is possible to reduce the AD4ON execution time, i. e., the time needed to make solution.
- Take into account the near future – this property is addressed in the Chapter 7.

# 6

## Evaluation of the AD4ON algorithm

Once the AD4ON algorithm is designed, we need to evaluate it. In this chapter, we aim to demonstrate that the AD4ON is capable to address the expected properties identified in the previous chapter. For example, we aim to demonstrate that the AD4ON is capable do make decisions in a heterogeneous network environment while takes into account multiple attributes; that it is capable to choose access network that better satisfy the requirements from the different actors (e. g., increasing flows and user satisfactions); and that its decisions are stable, preventing issues like “ping-pong” effects.

For that purpose, we implemented the functional modules of the AD4ON architecture and we test it by simulation. In order to test the performance of the AD4ON algorithm, we compare it with other three DM algorithms:

- MADM TOPSIS – as verified in Chapter 3, the *MADM TOPSIS* has been one of the most used DM in the literature. Therefore, we judge relevant to compare the AD4ON with such algorithm;
- Modified version of TOPSIS (mTOPSIS) – as previously described, MADM techniques present some issues to be applied in dynamic environment as the C-ITS one (e. g., rank abnormality). Such an issues have motivated several works to propose modified TOPSIS versions to be applied in different domains. In the network selection domain, we can highlight the work developed by Senociu [66] that proposes modifications in some TOPSIS

processes. Such work claims to reduce the rank abnormality and to increase the satisfaction of a communicating flow. We implemented the modified TOPSIS algorithm based on the Senouciu work, which we named *mTOPSIS*, and we compare it with the AD4ON;

- Commercial DM (CM) – besides the traditional TOPSIS and the *mTOPSIS* algorithms, we compare the AD4ON with a *commercial DM* used in most of smartphones, in which decisions are based on predefined network priorities (e.g., connect to WiFi if available or to 3G/4G otherwise).

With a mix of real and simulated input data, we performed simulation with the four DM (i. e., AD4ON, TOPSIS, *mTOPSIS* and CM) and we compared the results from such algorithms.

In the following we first describe the AD4ON evaluation process. Then we explain how the testbed was implemented and the real data acquired. Finally, we describe the different simulated scenarios, as well as the results of the simulations process.

## 6.1. The evaluation process

Before to define the scenarios and start to evaluate the AD4ON, we need to define the evaluation process. In the follow, we show an overview of such process, and then we describe each step of such evaluation process.

### 6.1.1. Evaluation overview

In order to evaluate the AD4ON mechanism, we implemented the AD4ON architecture as shown on Figure 6.1.

In such figure, we depict the different approaches we used to implement the AD4ON architecture for the present simulations. The different implementation approaches are identified by the following color code:

- Blue squares – the functional modules marked with the squares filled in blue were simulated, i. e., they are models (not necessarily based on real data) created for study and analysis of a given scenario. An example is the flow requirements, which are modeled in order to simulate the different DM algorithms. To implement such functional modules we could use well-known simulators as Network Simulator 3 (NS-3) or SUMO. However, in the present simulations we do not need all capabilities of such a simulators. In fact, we need a piece of code capable to take inputs, process it and give results. Therefore, we implemented our own simulator, which was implemented using Python version 2.7 as the programming language;

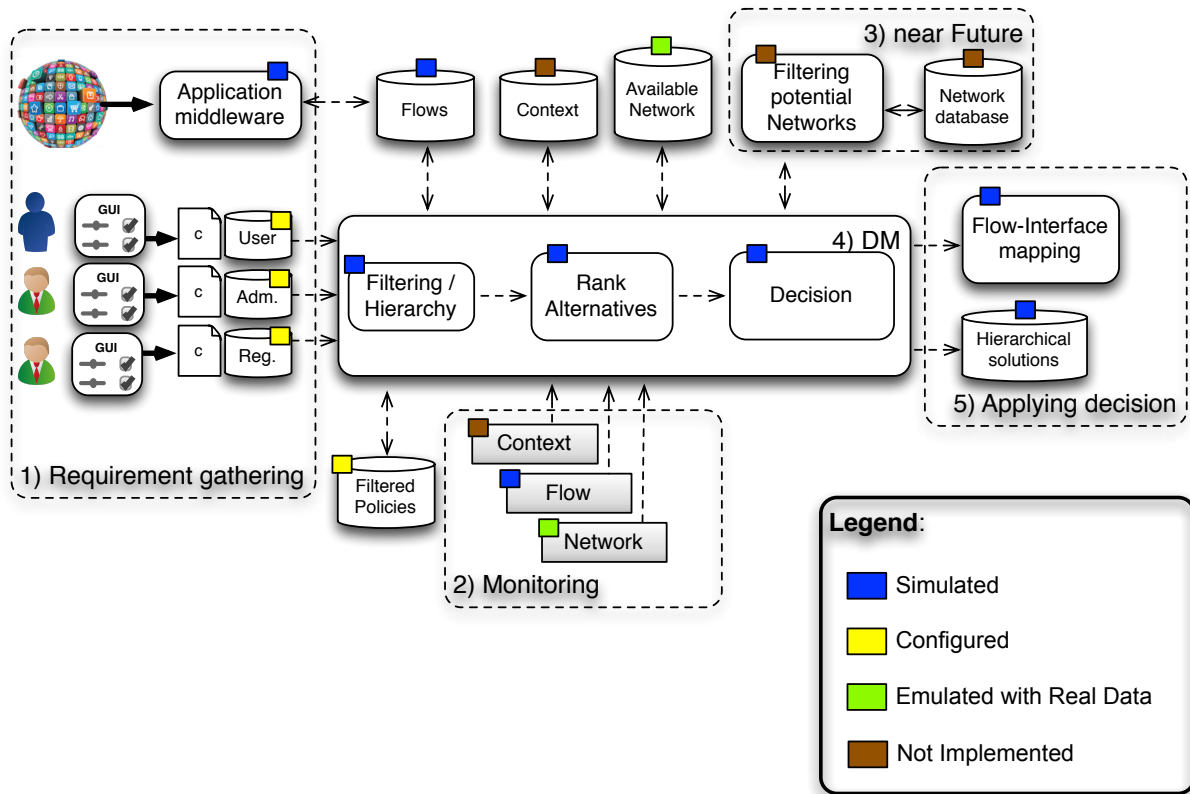


Figure 6.1: AD4ON implementation

- Green squares – modules marked with squares filled in green were emulated with real data, i. e., it mimics real data. Such a values are based on real data. For example, the networks performance are based on real measures acquired from a testbed realized in the campus of the University of Murcia in Spain.
- Yellow squares – for the sake of simplicity, we considered some modules with constant requirements. For that modules, values are configured once at the begin of the simulation and it remains constant during all simulation process. For example, in the simulation process we previously configure the amount of money an user is willing to pay for data transmission, which is constant during the entire simulation. Such a modules are marked with squares filled in yellow;
- Brown squares – the modules marked with squares filled in brown represent the functional modules related with the prediction of the near future about the vehicle environment. These modules were not implemented in the simulation performed in the present chapter. Instead, they are addressed in the Chapter 7.

We split the simulation process in four parts. Three parts representing the AD4ON architecture implementation (i. e., *Input data*, *Making decision* and *Results*), and one part representing the

comparison of the different algorithms based on some Key Performance Indicator (KPI) (i. e., *Comparison*). An overview of such a simulation process is depicted in the Figure 6.2 and described in the following sections.

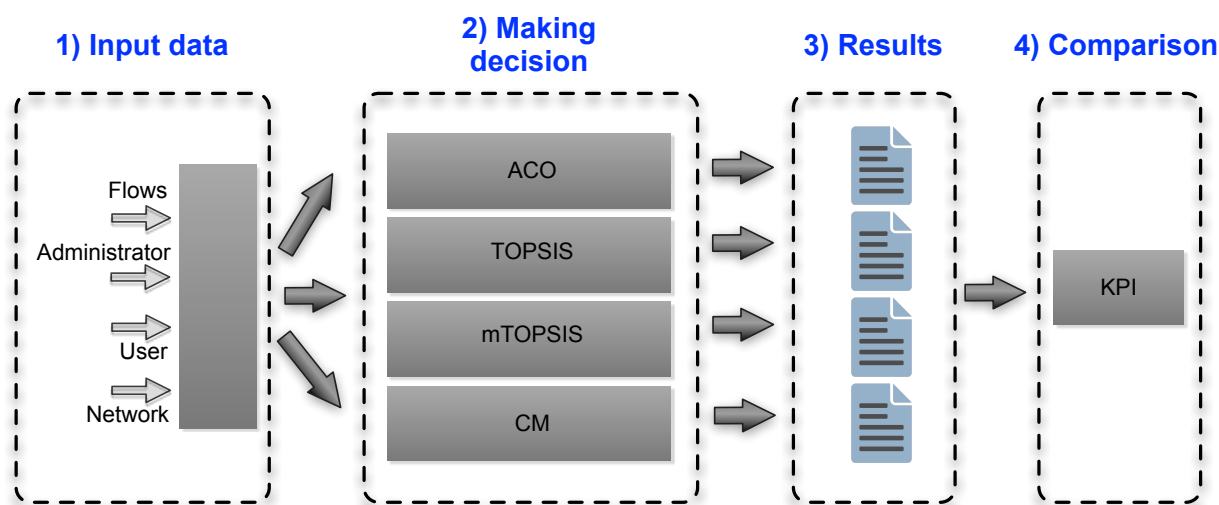


Figure 6.2: Simulation overview

### 6.1.2. Input data

This module implements the 1) Requirement gathering and 2) Network Monitoring modules from the AD4ON architecture. Therefore, it is responsible to acquire data from all actors involved in the communication decisions (i.e., flows, administrators, users and networks).

In the current version of the AD4ON algorithm, we does not consider contradictory requirements. Therefore, this module only concatenates the requirements from the different actors and sends them to the *Rank Alternatives* module. In the following, we describe the information that can be presented by each actor.

#### 6.1.2.1. Flows

As vehicle become connected and cooperative, a wide variety of on-board devices and applications are likely to communicate. According to ETSI, applications are divided in four main classes: conversational, streaming, interactive and background classes [89]. And each of such classes has specific requirements in terms of DR, latency, PDR and others.

Besides such applications, connected and cooperative vehicles can offer other ITS services. In the present simulations we consider one more service class that we called *safety*. Such class groups critical applications, i.e., applications that disseminate and/or exchange critical information related with safety of life.



Based on standard QoS requirements defined by [89], on the values more frequently used in the literature, as well as on reference values required by most of popular applications like Skype, YouTube and Netflix [98, 99], we defined some range values for each flow requirement. Table 6.1 list such requirements for the five classes.

Table 6.1: Range of flow requirements

Flow Name	Data rate (Kbps)	Latency (ms)	Packet Loss (%)
Conversational	[100 .. 500]	[30 .. 400]	[5 .. 15]
Streaming	[500 .. 1,933]	[500 .. 10,000]	[5 .. 20]
Interactive	[4 .. 500]	[500 .. 4,000]	[5 .. 30]
Background	[4 .. 1,933]	[500 .. 10,000]	[5 .. 35]
Safety	[2 .. 500]	[0 .. 100]	[5 .. 10]

#### 6.1.2.2. Administrators/Rulers

We assume that administrators or rulers can authorize or prohibit the use of a given access network, and/or prohibit the traffic of certain flows type through some access networks. Such policies are transmitted to the AD4ON by a configuration file.

#### 6.1.2.3. Users

According to the AD4ON architecture, users can express its preferences in the decision making process. In the simulation, we consider that user inform the amount of money he or she is willing to pay for a given communication (e.g., \$ 0.2/MB). Like the administrators policies, users preference is sent to the AD4ON by a configuration file.

#### 6.1.2.4. Networks

Network characteristics like DR, latency, and availability are key information in the network selection process. Few works deal with how these information are acquired. Some of research works consider to acquire such information by cooperation with the network, e.g., using specific functionalities from the IEEE 802.21 MIH standard. The majority of current works just assume that all necessary information is previously available, without describing how these parameters are acquired. They assume these information are acquired either dynamically from some non detailed monitoring modules or assigned statically, i.e., network information are configured previously in the decision making process.

The ISO ITS-S reference architecture defines the interfaces and their primitives between the “ITS-S management” cross layer and the other horizontal layers. In this way, the DM at the management plan is supposed to communicate with other layers (e. g., ITS-S access layer or ITS-S network layer) for the monitoring process. However, the ISO ITS-S reference architecture does not define a framework for such a functional monitoring modules. Therefore, each developer can implement its own monitoring module. In our simulations, we consider two ways to acquire network information. The network monitoring module can assign random values based on predefined threshold. Such an approach is used in most of researches in the literature. The second way we acquire network information is emulating access network based on real data measured on the field. The testbed used to acquire such a network information is described below.

#### 6.1.2.5. Testbed

Besides random input data for network conditions, in our simulations we aim to compare the different DM algorithms using input data near to real network environment. For that purpose, we aim to observe the behavior of the different technologies, i. e., how their connectivity evolve in the real world. Based on such observation, we can create different network environments to perform the simulations. Figure 6.3 shows an example of four created vehicular WiFi environment based on real measures of a given Roadside Unit (RSU). The *ITS-G5 1*, *ITS-G5 2*, *ITS-G5 3*, and *ITS-G5 4* are emulated access networks. They was created by varying the signal amplitude and/or the position of the access network from real measures.

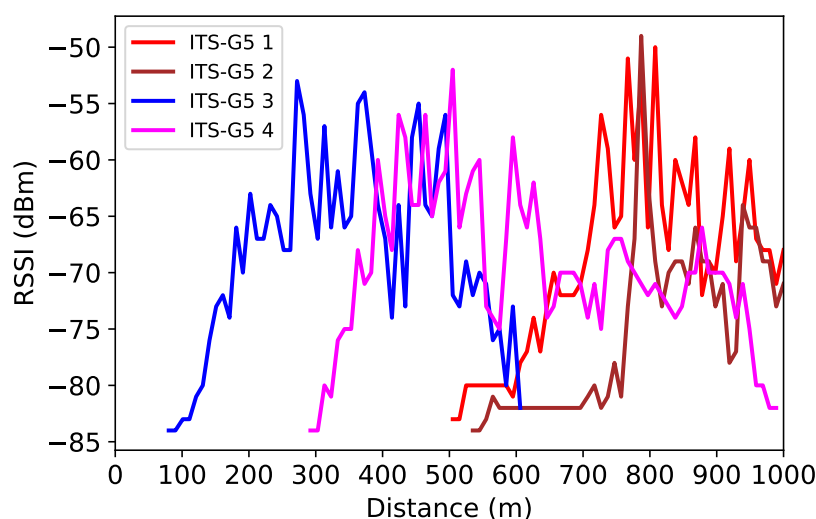


Figure 6.3: Example of ITS-G5 network environment created from real measures

In order to acquire real data from access networks, the AD4ON architecture has been deployed in a testbed installed in a real driving area. The testbed setup and results are presented below.

**Testbed scenario**

We deployed the testbed depicted in Figure 6.4. This testbed has been deployed in the Espinardo campus at the University of Murcia (UMU) in Spain. The three types of ITS-S presented by the ITS-S communication architecture (i.e., Vehicle ITS-S, Roadside ITS-S and Central ITS-S) are included. An vehicle equipped with a mobile router is capable to communicate with a CN by means of two available communication routes: the one provided by ITS-G5 and the one provided by cellular operator. We used IPv6 addressing scheme. Since cellular network does not support IPv6 connectivity, an OpenVPN [100] tunnel over IPv4 has been used. On the road-side, an ITS-S Access Router is placed inside the building of the Faculty of Computer Engineering and it is wired connected to the local network where is located the CN.

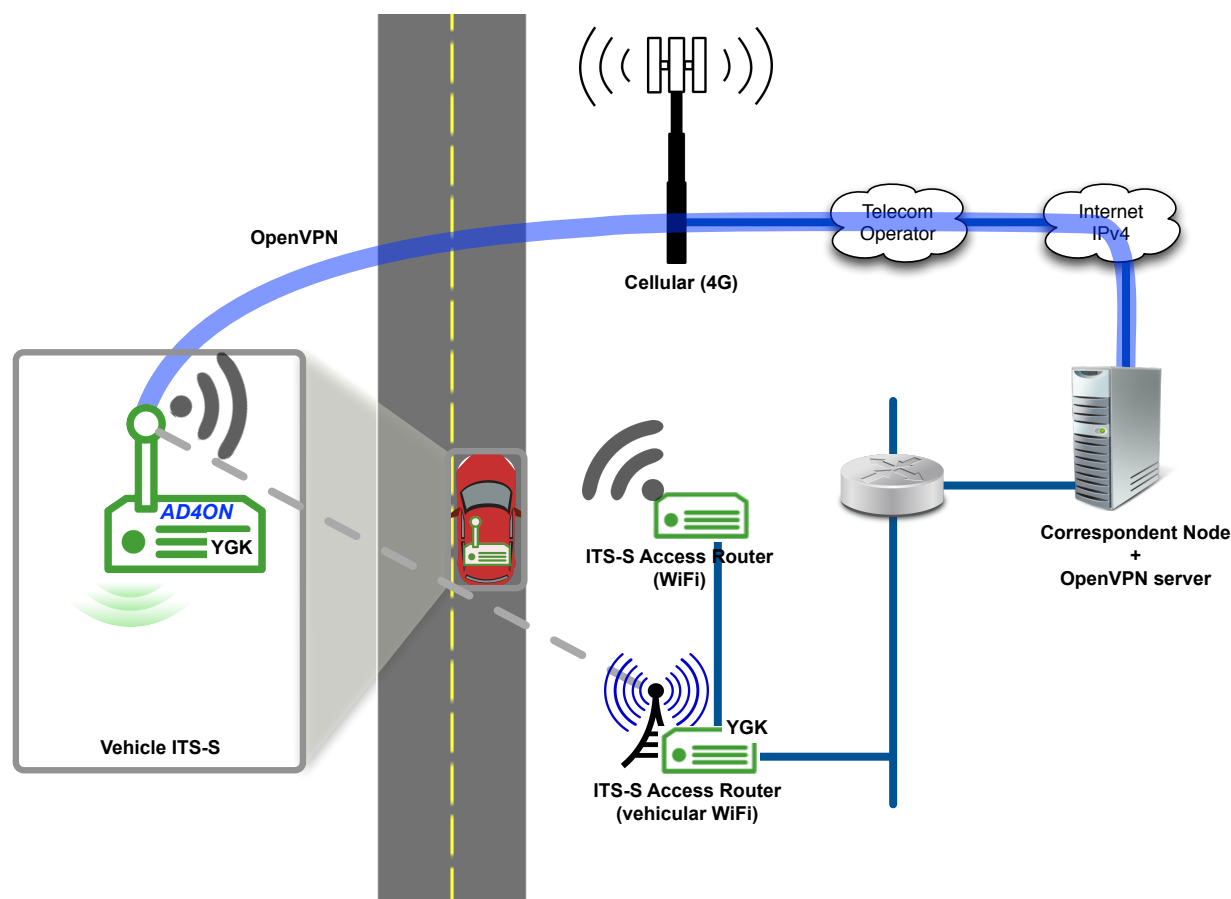


Figure 6.4: Deployed ITS network scenario.

The equipment used in the testbed are shown in Figure 6.5. A vehicle from the UMU fleet is used to mount the vehicle ITS-S (Figure 6.5a). It is equipped with a roof antenna and a Yogoko's mobile router [101], as shown on Figure 6.5b and Figure 6.5c, respectively. As showed on Figure 6.5d, the Roadside ITS-S antenna was placed in a window of the Faculty of Computer Engineering.

## 6.1. THE EVALUATION PROCESS



(a) Vehicle from the UMU fleet.



(b) Vehicle antenna.



(c) Mobile router.



(d) Roadside ITS-S antenna.

Figure 6.5: Equipment used.

For the sake of clarity, the components used in the testbed are listed in Table 6.2. As can be seen, the same base hardware is used for both ITS-S border router, i.e., the MR and the Access Router (AR), but the last one is located in the Roadside ITS-S.

Table 6.2: Components used in the testbed

Component	Hardware / Model	Software
MR	ARM iMX6	Debian 8
AR	ARM iMX6	Debian 8
CN	PC Intel i5, 3.1Ghz, 3GB	Ubuntu 10.4
Vehicle antenna	Omni 3G/11p/GPS 7dBi	-
Roadside antenna	Omni 12dBi	-
Cellular card	SIM 3G Vodafone	-

The purpose of this test is to measure the RSSI, Latency and the PDR for vehicular WiFi (ITS-G5) and Cellular (3G) networks, while vehicle moves in a predefined circuit of 1000 meters

inside the Espinardo Campus at UMU. Figure 6.6 shows such traced circuit as well as the network coverage along the path. The circuit is completely covered by 3G, while the ITS-G5 coverage is limited to a part of the circuit next to the ITS-G5 antenna.

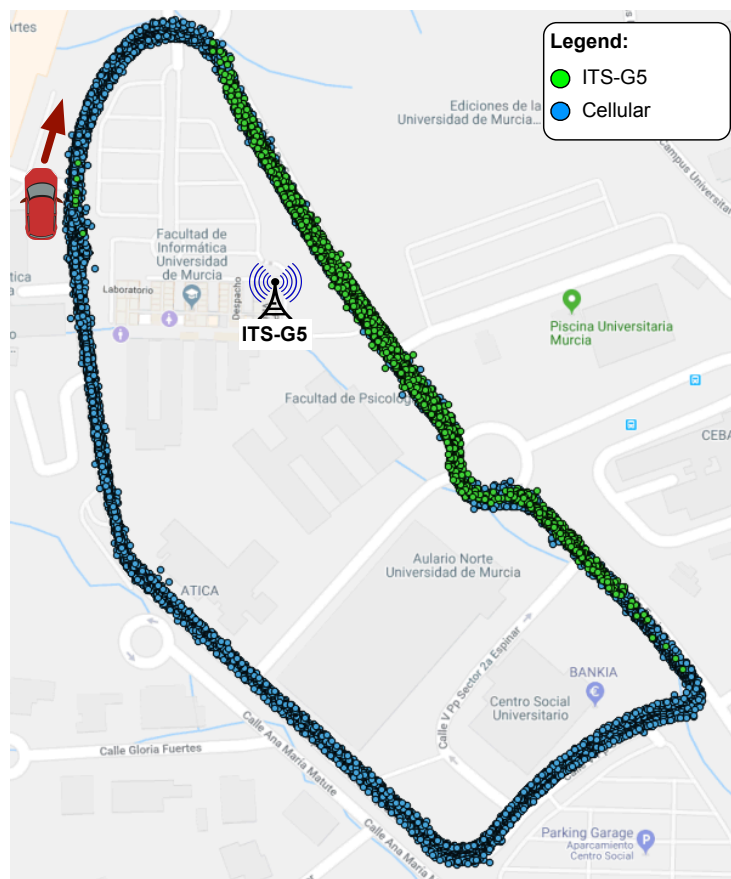


Figure 6.6: Circuit of tests with ITS-G5 and Cellular coverage.

In each trial, the vehicle starts at the same location, as indicated in Figure 6.6. And it performs a whole lap to the circuit, while measures data from both technologies. The measures was performed using different tools:

- PDR: it was measured using `iperf2` with UDP protocol. The `iperf` client and server functionalities was configured on the MNN and CN, respectively. UDP datagram is sent every second with different bandwidth (500 Kbps, 1 Mbps and 2 Mbps);
- Latency was measured by using `ping6` command line: a ping is performed every second from MNN to CN, sending 56 bytes of data in each packet. We assumed symmetrical delay between MNN and the CN. Therefore, latency was equal to half of RTT, i.e., half of the time it takes for a packet to go from the sending endpoint (MNN) to the receiving endpoint (CN) and back;
- RSSI was acquired by querying the embedded cards (cellular and WiFi) in the mobile router.

All measures have been performed at the same vehicle speed: 20 Km/h. With the aim of providing statistical confidence to the results, each particular configuration has been tested five times. Moreover, the GPS position was stored for each measure.

### Results from the testbed

The results from the testbed are shown on Figure 6.7. Each figure shows the mean of measured values as well as the standard deviations.

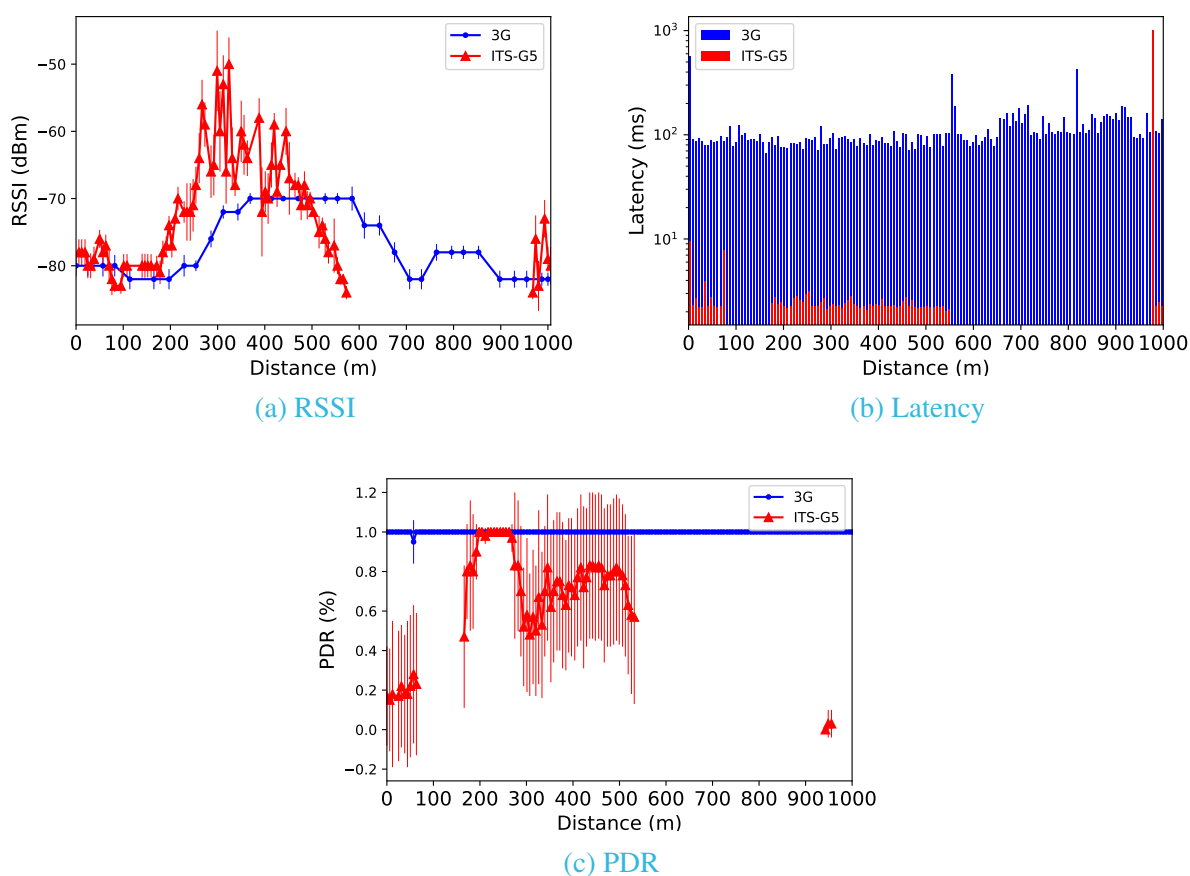


Figure 6.7: Measures from testbed

Figure 6.7a shows the RSSI for both, ITS-G5 and 3G network. As expected, the ITS-G5 signal is higher in front of the antenna (i.e., around position 300m). We observe a reduction of the signal just before position 400m. This is the instant the vehicle cross the roundabout, and the presence of buildings reduce the signal received by the vehicle. The 3G signal is low when vehicle is closer to the base station and high when vehicle is far from the base station. This can be explained by the 3G network capability to adapt the power according to the distance between the mobile node and the base station.

Figure 6.7b shows the latency average measured between the vehicle and the CN along the

trip. Due the signal reflection in the nearby buildings, the vehicle receives ITS-G5 signal at the start point of each lap. Therefore, we can observe some packet transmission at the start point. After that, as the vehicle is not covered by ITS-G5 no latency is measured until it enter in the ITS-G5 coverage zone. The latency in ITS-G5 is quite regular, i.e., it presents a low variation. As the cellular coverage is always present, we can get latency in the entire circuit. Since the communication by cellular use the tunnel VPN and go through the Internet, its latency is higher than the local communication through ITS-G5. Such difference of latency values between both technology is not a problem in the present work. Instead, it is an use case composed by one local communication and another global (i.e., through Internet) communication.

Figure 6.7c shows the PDR measured for both technologies. The cellular presents a PDR near to 100% all along the way, with low variation. Unlike the cellular network, the ITS-G5 presents high variation in the PDR measurements. Such a high variation indicates that in real implementation instantaneous PDR values for the vehicular WiFi should be used with care, i.e., the DM should take into account such a variations in the decision making process. Therefore, in the present simulations we considered average values for the PDR.

### 6.1.3. Making decision

The making decision module implements the part 4) DM from the AD4ON architecture. Since the current version of the AD4ON algorithm does not consider contradictory requirements, the *Filtering/Hierarchy* module reads the requirements from the different actors and sends them to the *Rank Alternatives* module.

Therefore, based on such requirements and network conditions, the *Rank Alternatives* module creates the graph of possible solution. And the different DM are executed. Besides the AD4ON, we implemented three other DM algorithms: a traditional implementation of TOPSIS, a modified version of TOPSIS, and a version of current commercial DM. In the following, we briefly describe the implementation of such three algorithms.

#### 6.1.3.1. TOPSIS

The MADM TOPSIS algorithm is designed to find the best solution for a given problem among a list of potential solutions. It is not designed to find solutions for multiple problems simultaneously, like the network selection problem. In the case addressed in this thesis, we have multiples flows competing to communicate through multiple and heterogeneous networks. Such a scenario requires the DM algorithm to find a solution for each flow (if a solution exist).

We implemented the traditional TOPSIS algorithm following the main steps described in

section 3.2.1.2. And to address such a network selection process, we randomly choose a flow as a starting point, and we visit all flows. For each visited flow we execute the TOPSIS algorithm as follow:

- Decision matrix creation – we create the decision matrix, which is composed by all potential solutions for that flow (i. e. all access networks that satisfy the minimum communication requirements);
- Decision matrix normalization – then we normalize such decision matrix using vector normalization technique;
- Weighted matrix – after normalization, the decision matrix is weighted. Since we consider all attributes have the same importance in the decision process, we use the same weight for all attributes, i. e.,  $weight = 0.2$ ;
- Ideal solutions – we find the ideal positive and ideal negative solutions, i. e., among all attributes in the decision matrix, we find the hypothetical access networks composed by the best and worst attributes values, respectively;
- Select solution – we compare all potential access network present in the decision matrix with the ideal solutions, and we select the best solution, i. e., the one with shortest Euclidean distance from the positive ideal solution and the farthest Euclidean distance from the negative ideal solution.

### 6.1.3.2. mTOPSIS

In TOPSIS, decision calculations are based on a matrix where lines represent available networks and decision attributes are set on columns. A quadratic vector normalisation is applied on columns in order to homogenize the weight of each attribute. Then, it chooses the solution with the shortest Euclidean distance from a positive ideal solution and the farthest Euclidean distance from a negative ideal solution. In this approach, the algorithm selects the network with highest score regardless of the application satisfaction level.

Moreover, the so-called ideal solutions are based on the maximum and minimum attribute values in the matrix. This results in a high sensitivity to extreme values which leads to unstable decisions when a mobile approaches the limit of range of a given network (where the network can appear or disappear between two steps of calculations).

To mitigate these issues, we implemented the modified version of the TOPSIS algorithm proposed by authors of paper [66], which we called mTOPSIS. This algorithm is created to overcome the two main issues of the traditional TOPSIS algorithm: 1) the fact that TOPSIS does



not take care about flow satisfaction, i.e., it selects the network with highest score regardless of the application satisfaction level; and 2) to avoid the rank reversal behavior, in which changes in one network parameter can results in a new solution.

The mTOPSIS algorithm replaces the classical quadratic normalization by a new approach based on utility functions. We applied the same utility functions used by the AD4ON algorithm. In this way, it stabilizes the normalized process by normalizing attribute values individually, while applications can present a specific utility function for each attribute. Therefore, changes in one attribute will have no effect on the normalization of others, and it reduces the high sensitivity to skewed data. Moreover, it sets the ideal positive and negative solutions as being “1” and “0”, respectively. Therefore, ideal solutions remains independent of available networks. As described by [66], this approach claims to eliminate the rank reversal and increases the ranking quality by fulfilling the application requirements.

### 6.1.3.3. CM

In order to compare the AD4ON DM algorithm with current commercial solution, we implemented the CM algorithm. The CM is a version of DM used in most of smartphones, in which decisions are based on predefined networks priority.

Before to start the decision process, we define a list of fixed decision rules, such that flows are connected to WiFi if available, or cellular networks otherwise. Therefore, for each flow the CM chooses the access network based on such predefined network priorities.

### 6.1.4. Results

Once each DM algorithm found their solutions, we can apply them in the system. The enforcement module implements the part 5) *Applying decisions* from the AD4ON architecture. It is responsible to receive decisions performed by DMs and enforce them in the system.

However, instead to apply the decisions, in the present simulations we aim to compare them. Therefore, the enforcement module has the set of access networks that better satisfy each flow. For the sake of comparisons, we store the DM results for each algorithm.

### 6.1.5. Comparison

The *Comparison* module is not part of the AD4ON architecture. It is a module used to evaluate decisions from the different DM algorithms.

As we have seen, the network selection addressed in this thesis consists in choose for each flow,

the access network that better satisfy the communication requirements, which come from different actors. According to the literature, performances of the DM algorithms are usually evaluated using an objective function (utility or cost function) regardless of whether or not it satisfies the actors needs. In this case, the higher is the utility value of a decision, the better the solution. However, such evaluation approach does not necessarily reflect the actual actors needs. In fact, it is necessary to analyze how the DM algorithms meet the requirements from all actors participating in the decision process. In other words, it is necessary to know the level of actors satisfaction.

In the AD4ON mechanism, four actors express their communication requirements in the decision making process (*flows*, *users*, *administrators* and *regulators*), and one actor (*access networks*) gives information about its availability and performance capability. For these actors expressing communication requirements, we are interested to evaluate how the simulated DM algorithms satisfy them. In the current AD4ON implementation, we assumed that administrators and regulators only express rules about forbidden networks. In this case, the “Rank Alternative” module ensure that only feasible solution are sent to the DM algorithm. Therefore, we will not define specific performance analysis for administrators and regulators. In terms of the other two actors (i. e., flows and users), we defined specific KPI to evaluate the flows and users satisfaction, respectively.

However, only actors satisfaction is not enough to evaluate DM algorithms in dynamic environment, as the C-ITS one. In such an environment, network performance and availability can change frequently. And changes in the access networks during the flow communication can results in QoS degradation, e. g., packet loss, out of order packet delivery, abrupt variation of latency due to the existing queue of data and/or the difference of performance between the access networks. Therefore, DM algorithms should perform stable decisions, reducing the network switching and consequently the “ping-pong” effect. In such a dynamic scenario, the DM should find a compromise between rapidly switch the flow communication in order to benefits of better access networks, and the decision stability to prevent QoS degradation. For that reasons, besides the flows and users satisfaction we defined another KPI related to the stability of the decisions. The three KPIs are defined as follow:

- The *flow satisfaction*: it is the percentage of meeting flow requirements. We consider that a given flow is completely satisfied if its all requirements are 100% satisfied by the chosen network. For example, a flow that requires a maximum data rate  $DR_{flow}$  and a minimum latency sensibility  $L_{min}$  is 100% satisfied by a network  $N$ , if such network is capable to supply the flow with a data rate  $DR_{net}$ , such that  $DR_{net} \geq DR_{flow}$ , and with a latency  $L_{net}$ , such that  $L_{net} \leq L_{min}$ . If the chosen network satisfy only the minimum value for all parameters required by a given flow, such a flow satisfaction will be the minimum one, i.e., 10% as considered in this work. The *flow satisfaction* is zero if the minimum value of at

least one flow requirement is not filled.

- The *stability of decision*: frequent changes of network can increase the packet loss and the communication latency. Therefore, we aim to reduce the number of network switching. To calculate this indicator, we first calculate the decision instability that is the ratio between the number of network switching performed by each DM algorithm and the maximum possibility of network switching in a given scenario. Then, the decision stability is given by  $(1 - instability)$ .
- *Monetary utility*: this KPI represents the user satisfaction. We aim at finding solutions that offer the lowest monetary cost for users (i.e., higher monetary utility). We assume that the user informs the DM algorithm of the maximum price he or she is willing to pay for data communication. Based on this information, the DM can calculate a monetary utility as  $(1 - cost)$ , where *cost* is the ratio between the monetary cost billed by a network operator for data communication and the maximum price the user is willing to pay.

In order to compare the AD4ON algorithm with others, we used the three previously defined KPIs. Therefore, the best DM algorithm will be the one that maximizes the flow satisfaction, the decision stability, and the monetary utility.

For that purpose, we defined a total utility function that is the average of the three KPI, as described by Equation 6.1. In this way, the algorithm with the best performance is the one that finds solutions with highest *Total Utility (TU)*.

$$TU = \frac{\sum_{i=1}^N KPI_{(i)}}{N} \quad (6.1)$$

where  $N$  is the number of KPI (i.e.,  $N = 3$ ).

In this thesis, we consider the three KPIs having the same importance in the decision making process. Therefore, they have the same weight in Equation 6.1. However, in scenarios where the DM should privilege a KPI over another, different weights can be assigned.

## 6.2. Description of simulation scenarios and results

In order to simulate the AD4ON algorithm, we consider the following context: a vehicle equipped with a MR is being driven in a zone covered by heterogeneous access networks. As the vehicle moves, it experiences different network conditions. The MR is equipped with several wireless access technologies such as, vehicular WiFi (ITS-G5 in Europe and DSRC in North America),

urban WiFi (IEEE 802.11n or 11ac), cellular (3G, 4G, and 5G under preparation) and GPS. And it is capable to connect with such multiple access networks simultaneously.

Therefore, the DM algorithms inside the MR should to manage a variety of on board applications such as, video conference, video streaming, emails and navigation system, while it moves across areas with different network conditions (i.e., network availability, coverage areas, and network performance in terms of data rate, latency and PDR).

Considering such context of connected and cooperative vehicles, we defined the simulation scenarios. We consider that *Administrator/Regulators* policies and *Users* preferences are presented to the AD4ON algorithm at the begin of simulation and remains constant during all simulations. Therefore, each simulated scenario is described in terms of access networks conditions and flows requirements.

For the sake of statistical analysis, in all scenarios the AD4ON algorithm was executed 5 times. Indeed, due to the stochastic property of ACO algorithms, the results of AD4ON may vary between two executions of the same scenario.

Table 6.3 overviews all simulated scenarios, given a brief description of each scenario. These scenarios are described below.

### 6.2.1. Scenario A

The scenario A is a simple scenario composed by one application flow and four access networks. The objective of this scenario is to show the output and the key performances for each DM algorithm separately, while the vehicle moves along a defined route of 1000 meters ( $m$ ).

#### 6.2.1.1. Flow inputs

We consider a hypothetical flow named *App1*, whose flow requirements are showed on Table 6.4. Such flow presents its requirements to the AD4ON algorithm at the begin of communication, which remain constant during all simulation.

Table 6.4 shows the minimum DR required by *App1* (min DR), below which the flow cannot communicate. The maximum DR (max DR) indicates the value of maximum utility for *App1*, i.e., value for which flow communication performance cannot be improved, even if more DR is available. The *min Latency* and *max Latency* indicate the minimum latency perceived by the flow and the maximum delay accepted by the flow, respectively. Access networks with latency greater than this *max Latency* is not a feasible solution. Finally, the *max Packet Loss* indicates the acceptable amount of data packets loss.

Table 6.3: Simulation scenarios

Scenario	Description
Scenario A	<ul style="list-style-type: none"> <li>• Flow: one hypothetical flow (called <i>App1</i>) with constant requirements;</li> <li>• Networks: four access networks, which are based on real measurements on the field;</li> <li>• Objective: Analyze and compare the output and KPI for each DM algorithm, while the vehicle moves along a defined route of 1000 <i>m</i>.</li> </ul>
Scenario B	<ul style="list-style-type: none"> <li>• Flows: four flows (Streaming, Conversational, Interactive, and Safety) with variable requirements. Requirements values randomly chose from a predefined range of values;</li> <li>• Networks: four access networks. The network parameters are randomly generated using a predefined range of values;</li> <li>• Objective: Compare the performance of the AD4ON algorithm with other 3 DM algorithms by using scenarios commonly used by most of literature works, i.e., with input values randomly chosen from a predefined range of values.</li> </ul>
Scenario C	<ul style="list-style-type: none"> <li>• Flows: We choose 20 flows that can start at different time. The time of flows arrival follows a Poisson distribution;</li> <li>• Networks: between 1 and 20 access networks. The number of access networks is randomly chose, while the network parameters (e.g., DR, latency, PDR, and RSSI) are derived from real measurements on the field;</li> <li>• Objective: Testing the AD4ON algorithm in a more complex scenario.</li> </ul>

Table 6.4: App1 requirements

Flow Name	min DR (kbps)	max DR (kbps)	min Latency (ms)	max Latency (ms)	max Packet Loss (%)
App1	200	2,500	2	200	30

### 6.2.1.2. Networks inputs

The network environment is composed by four access networks: one vehicular WiFi (ITS-G5), one Cellular (3G), and two urban WiFi (WiFi 1 and WiFi 2, respectively). The network parameters are based on real measurements on the field. The ITS-G5 and 3G values are based on the testbed realized in Murcia, as described on Section 6.1.2.5. While the urban WiFi values are based on measures realized inside the campus of the IMT Atlantique university at Rennes in France.

We combined such measures in order to create the network environment showed on Figure 6.8. For that, we take the real data and we randomize a little the values of the network attributes. Then, we overlap the different access network in order to create the network environment. We assume the vehicle moving from the start position 0 m until the final position 1000 m. The network parameter values observed by the vehicle along such route are the ones depicted on Figure 6.8. The RSSI is shown on Figure 6.8a. Figure 6.8b shows the available DR for each network. The latency values are shown on Figure 6.8c. And Figure 6.8d show the average of PDR.

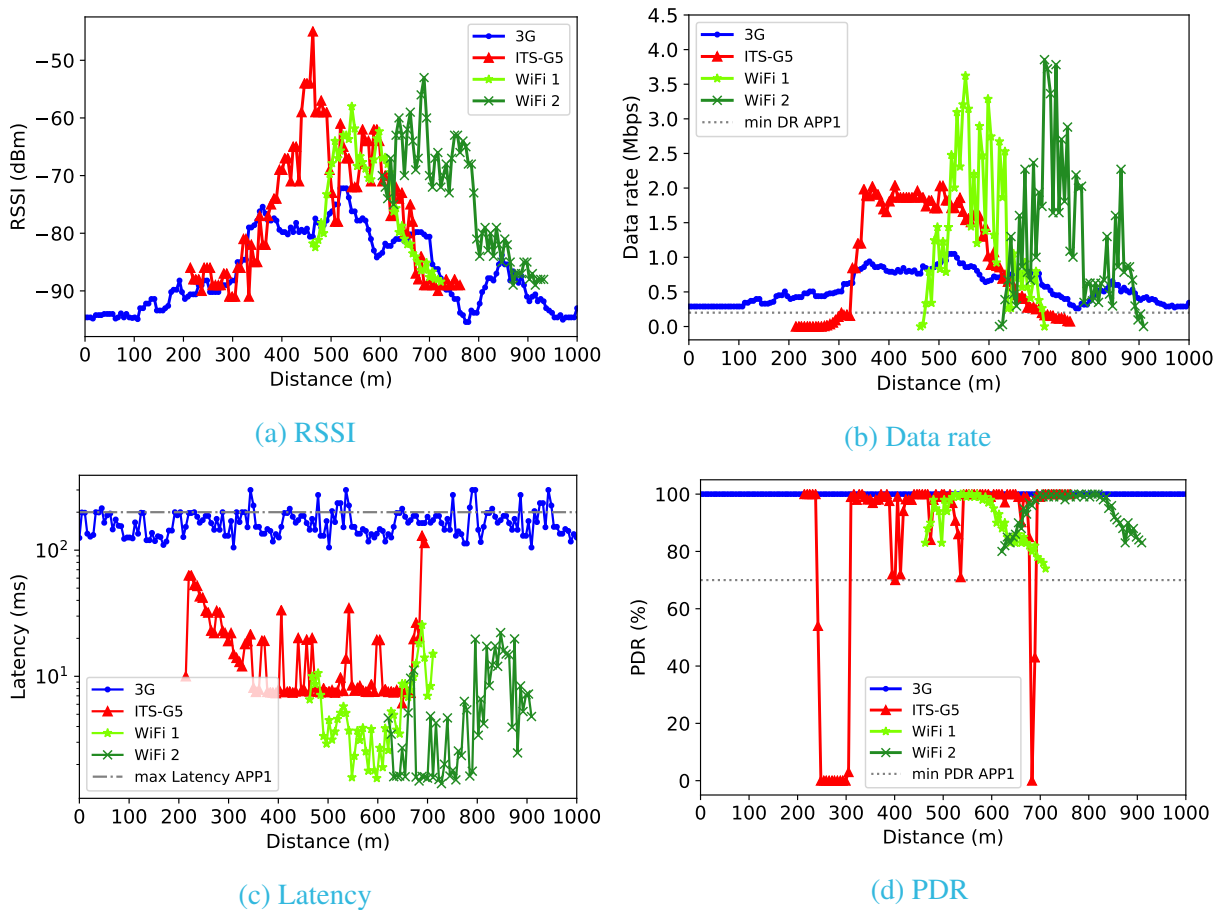


Figure 6.8: Input of network parameters for scenario A

### 6.2.1.3. Simulation results of scenario A

While vehicle moves along the defined route, we chose solutions using each one of the DM algorithms (i.e., AD4ON, TOPSIS, mTOPSIS and CM). Simulation results of the four algorithms on the scenario A are shown on Figure 6.9.

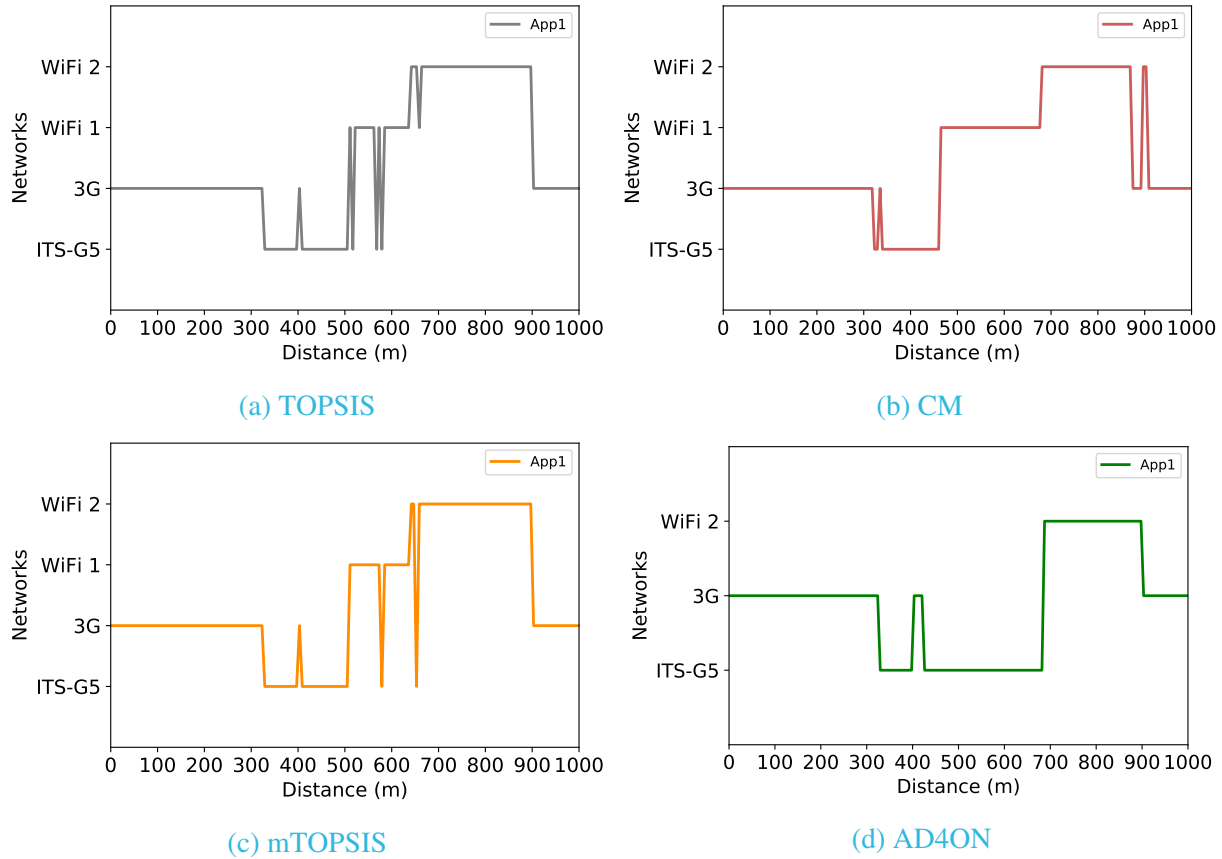


Figure 6.9: Results for scenario A.

Figure 6.9d shows the access networks chosen by the AD4ON algorithm for “App1”, while the vehicle moves along the route. As observed on Figure 6.8, at the begin (position 0 m) only 3G network is available. Therefore, the *App1* starts to communicate through 3G. When the vehicle enter in the ITS-G5 coverage such access network is considered in the decision making process. At that point, the vehicle is covered by both 3G and vehicular WiFi.

Around the position 320 m the AD4ON algorithm, as well as the other DM algorithms choose the ITS-G5 as a new solution to the *App1* communication. We observe that the CM algorithm presents a communication instability at that instant, as shown on Figure 6.9b. It switches the communication from 3G to ITS-G5, but quickly switches back to the 3G before to switch again to the ITS-G5 network, i.e., it presents the “ping-pong” effect. Since the CM decision is based on the RSSI, these network switching is caused by the variation of RSSI observed on Figure 6.8a.

At the position 400 m, the PDR offered by the ITS-G5 does not meet the minimum PDR required by the *App1*. Therefore, at that point such access network is not a feasible communication

solution for the flow *App1*. Since the vehicle remains covered only by ITS-G5 and 3G, all DM algorithms choose the 3G network as a new solution. Except the CM algorithm that does not take into account the flow's requirements. Such an algorithm are prone to choose bad quality solutions or even unfeasible communication solutions.

Unlike TOPSIS (Figure 6.9a) and mTOPSIS (Figure 6.9c), the AD4ON algorithm does not switch back to the ITS-G5 immediately after such network becomes a potential solution again. Instead, it maintains the *App1* communication through the ITS-G5 for a few period of time, in order to certify the quality of the new potential solution (in this case the ITS-G5), and to certify the vehicle is really covered by such new access network, i. e., if it is not only a transient coverage.

Such approach does not allow immediately take advantage of newcomer networks. However, it increases the decision stability. The reaction time can be reduced if the AD4ON is aware about the near future of the network conditions. In this way, it can perform proactive decisions and anticipate the network switching. We address the AD4ON proactive solutions on Chapter 7.

Analyzing Figure 6.9d we can see that the WiFi 1 network is not used. This is due the fact that both access network ITS-G5 and WiFi 1 have quite the same utility to the flow, i.e., the WiFi 1 utility is slightly higher than ITS-G5. In the AD4ON algorithm, decisions are driven by pheromone concentration based on the quality of solution. Therefore, due the slight difference between these two solutions, ants tend to reinforce quite the same amount of pheromone in both solutions. In this way, the pheromone evaporation process that is responsible to “forget” low quality solutions, takes longer to evaporate the pheromone concentration from the ITS-G5 solution. When the ITS-G5 pheromone concentration becomes lower than WiFi 1, a new better solution (WiFi 2) is already available. Then, the AD4ON algorithm chooses the WiFi 2 as new solution to *App1* communication.

In general, as expected the traditional TOPSIS present more “ping-pong” effect than the others, as we can observe on Figure 6.9a. Despite the adaptations implemented on the mTOPSIS algorithm, its decisions still presenting instabilities when network parameters changes.

The AD4ON algorithm outperforms the others in terms of decision stability. Thanks to the ACO property, in which decisions are driven by pheromone concentration, it is possible to create solutions smoothly over time. Such property enables to increase decision stability, avoiding the “ping-pong” effect.

Table 6.5 corroborates what we have observed so far. It shows the KPI (flow satisfaction, decision's stability and monetary utility) for each DM algorithm.

This specific scenario favors the CM algorithm. The RSSI distribution along the route, as showed on Figure 6.8a, enables a smooth network switching, reducing the “ping-pong” effect. Moreover, the network parameters follow the RSSI distribution, i.e., in most of times good RSSI levels coincide with good values of DR, Latency and PDR. Therefore, in this specific scenario, the



Table 6.5: Key performance results for scenario 1.

Algorithm	Flow satisfaction (%)	Monetary utility (%)	Decision stability (%)	Total utility (%)
TOPSIS	63.79	74.41	6.67	48.29
mTOPSIS	63.78	74.41	33.34	57.17
CM	59.79	70.48	46.67	58.98
AD4ON	62.22	73.76	66.67	67.55

CM presents not so bad performances.

The solutions found by the AD4ON present a flow satisfaction of 62.22% and a monetary utility of 73.76%. This means that, in average, the solution satisfied 62.22% of “App1” requirements and that for 73.76% of the time, the algorithm selected the access network with the lowest monetary cost.

The TOPSIS algorithm tries to find always the best utility value regardless the decision stability, i.e., without take care about the number of network switching. While the AD4ON algorithm can maintain a flow communicating through a less performing network until it certifies the quality of the new solution (as we discussed above about the AD4ON decision at the position 400 *m*). Therefore, for these two KPI (Flow satisfaction and Monetary utility) the AD4ON offers slightly lower performances than TOPSIS (around 1% difference). However, such slight underperformance is compensated by AD4ON stability, i.e., avoiding “ping-pong” effects.

Concerning decision stability, the AD4ON is better than the others. It performs 65% less network switching than TOPSIS and 50% less than mTOPSIS. This outperformance can be easily seen on Figure 6.9.

Analyzing the total utility (calculated by Equation 6.1), we observe that the AD4ON outperforms the other algorithms. Considering the three KPIs with the same importance (same weight), the AD4ON presents a total utility of 67.55%, i. e., the satisfaction of the entire communication system (actors satisfactions and decision stability) is 67.55%. Compared with TOPSIS, we observe that the AD4ON allows a slight reduction of the flow and user satisfactions in benefit of a more stable decision. The total utility results show that solutions found by the AD4ON algorithm offer the best compromise between flow satisfaction, decision stability and reduced monetary cost.

### 6.2.2. Scenario B

Several works perform simulations using scenarios in which flows and network values are randomly chosen from some predefined range of values [66, 47]. In this step, we aim to compare the DM algorithms using such a scenario approach. Then, input values from both flows and

networks are randomly chosen from a predefined range of values.

Scenario B is composed by 50 sub-scenarios. Each one involving four data flows and four access networks. Each data flow represents a given application with specific requirements in terms of DR, latency and PDR.

In each sub-scenario the vehicle moves along a route of 1000  $m$  long, while it experiences different network conditions and flow demands, as described below.

### 6.2.2.1. Flows inputs

In each sub-scenario, we considered four data flows: Streaming, Conversational, Interactive, and Safety. We assume that flows change their requirements once per second. And in each change the values of each requirement is randomly generated using the range of values given in Table 6.1.

### 6.2.2.2. Networks inputs

Like in scenario A, the sub-scenarios are composed by four access network: one vehicular WiFi (ITS-G5), one Cellular (3G), and two urban WiFi (WiFi 1 and WiFi 2, respectively). However, the values are not based on measured data. Instead, the network parameters (i.e., DR, latency, PDR, and RSSI) are randomly generated using the range of values given in Table 6.6. The access networks change their conditions once per second.

Table 6.6: Range of Network Parameters

Network Name	Data rate (Kbps)	Latency (ms)	PDR (%)	RSSI (dBm)	Cost (\$/MB)
Cellular	0 .. 14,000	0 .. 250	90 .. 100	-120 .. -65	0.1 .. 0.4
ITS-G5	0 .. 22,000	0 .. 200	80 .. 100	-110 .. -45	0
WiFi 1	0 .. 22,000	0 .. 200	40 .. 100	-110 .. -45	0
WiFi 2	0 .. 22,000	0 .. 200	40 .. 100	-110 .. -45	0

The network priorities defined for the CM were based on the monetary cost of networks, i.e., free networks were privileged. Therefore, we defined the following descending priority's order: urban WiFi (WiFi 1 and WiFi 2), ITS-G5 and Cellular. The CM algorithm chooses urban WiFi if such network is present. Otherwise, it looks for the ITS-G5. The cellular networks have lower priority, then they are chosen if vehicle is not covered by urban WiFi or ITS-G5.

### 6.2.2.3. Simulation results of scenario B

The results for the simulations of scenario B are shown below. We show each key performance results only for the streaming flow. For the other flows we only show the total utility. However, the key performance results for all four flows can be seen in Annex 1.

### Results for the streaming flow

Figure 6.10 shows the key performance of the four algorithm, i.e., AD4ON, TOPSIS, mTOPSIS and CM for the streaming flow. Each of such key performance results is explained in the following.

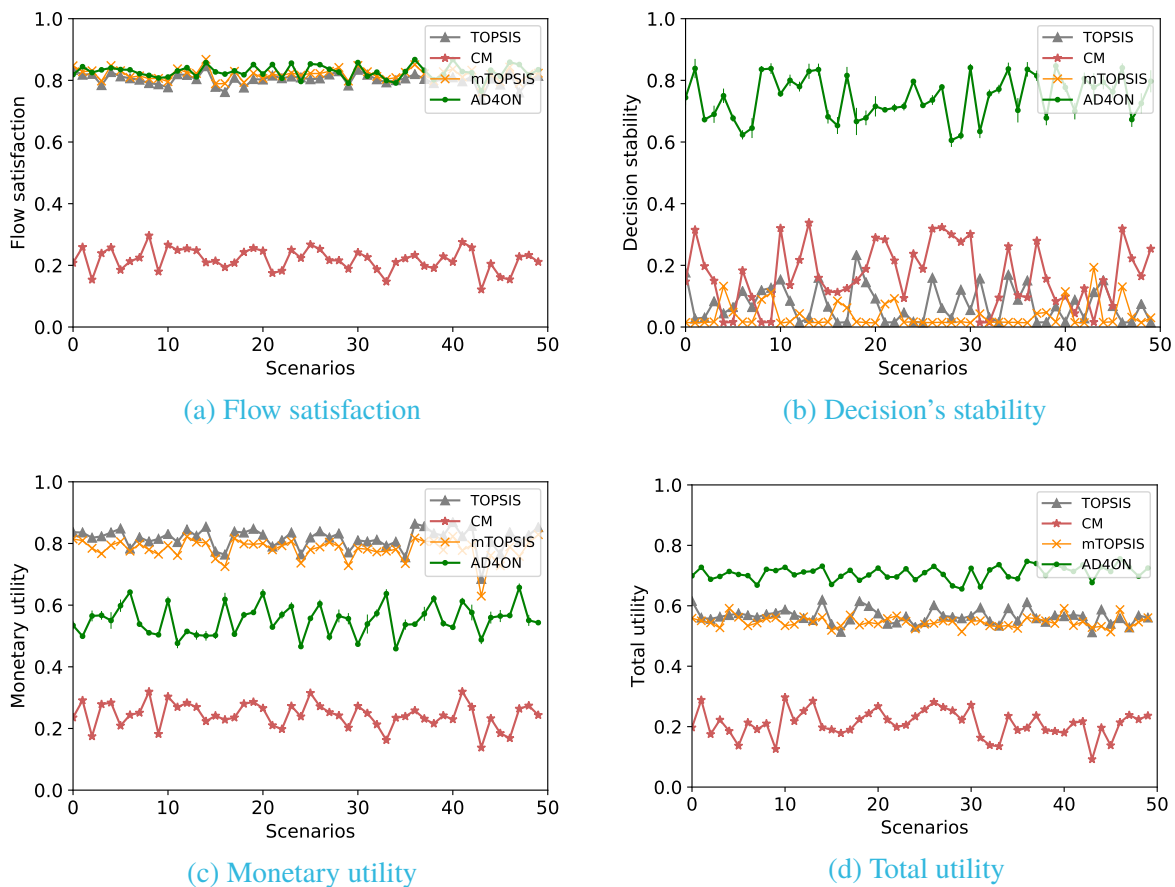


Figure 6.10: Key performance for streaming flow.

Figure 6.10a shows the streaming flow satisfaction with the solutions found by each algorithm. This figure represent an utility function, then higher value is better. We can observe that solutions found by the AD4ON generally meet more than 80% of the maximum streaming flow requirements for each sub-scenario, outperforming other algorithms. In the worst case, the AD4ON presents the same quality of TOPSIS solutions. Since the CM does not take into account the flow requirements, it presents the worst solutions in terms of flow satisfaction, meeting around 20% of the streaming flow requirements. Flows using such an algorithm, usually suffer constant disconnection and present poor QoS.

Figure 6.10b shows the decision stability of each DM algorithm. As we can observe, the AD4ON algorithm is more stable than the others, performing less network switching. In average, the AD4ON algorithm performs around 50 times less network switching than TOPSIS. We observe that the mTOPSIS is less performing than traditional TOPSIS. And this corroborates what we mentioned before on Section 3.2 about the paper [66], i.e., like TOPSIS, the mTOPSIS does not prevent network switching when few network parameters change. It is designed to eliminate rank reversal only for specific cases where the network can appear or disappear between two steps of calculations.

We aim to find solutions that minimize the monetary expenditure for users. Figure 6.10c shows the monetary utility of solutions found by each DM algorithm. In this criteria, TOPSIS outperforms the AD4ON algorithm. Each time the network condition change, the TOPSIS algorithm will recalculate a complete solution for the system, i.e., it chooses an access network for each flow. If the new solution presents a high utility value than the current one, the TOPSIS switches the communication through such new solution (even the flow is already 100% satisfied in the current access network). Unlike AD4ON, the TOPSIS algorithm takes advantage of cheaper access networks immediately after such networks become available, thus increasing the monetary utility. However, the cost of such immediate network switching is the risk of “ping-pong” effect and QoS degradation.

However, analyzing the total cost showed on Figure 6.10d we observe that, the AD4ON algorithm outperforms the other algorithms, i.e., when we consider the three key performance together. The AD4ON found solutions with lowest cost for all simulated scenarios.

### **Results for all simulated flows**

Figure 6.11 shows the total utility of solutions found by each DM algorithm and for all simulated flows, including the streaming (which was previously described). The individual KPI results for all four flows (Streaming, Conversational, Interactive, and Safety) can be seen in Annex 1.

Analyzing Figure 6.11, we observe that the CM has the worst performances. Since its solutions are based only on the RSSI, it is prone to find not feasible solution, i.e., it can choose access networks that have good RSSI, but does not meet the minimum flow requirements (e.g., in terms of PDR, DR or Latency). This behavior impacts negatively the key performances, and consequently increases the total cost.

The AD4ON algorithm outperforms the other algorithms, finding solutions with the highest total utility for all simulated sub-scenarios. The AD4ON found better solutions for streaming, conversational and interactive flows in all simulated sub-scenarios, as shown on Figures 6.11a, 6.11b, and 6.11c, respectively. For the safety flow (Figure 6.11d), the AD4ON outperforms the others in

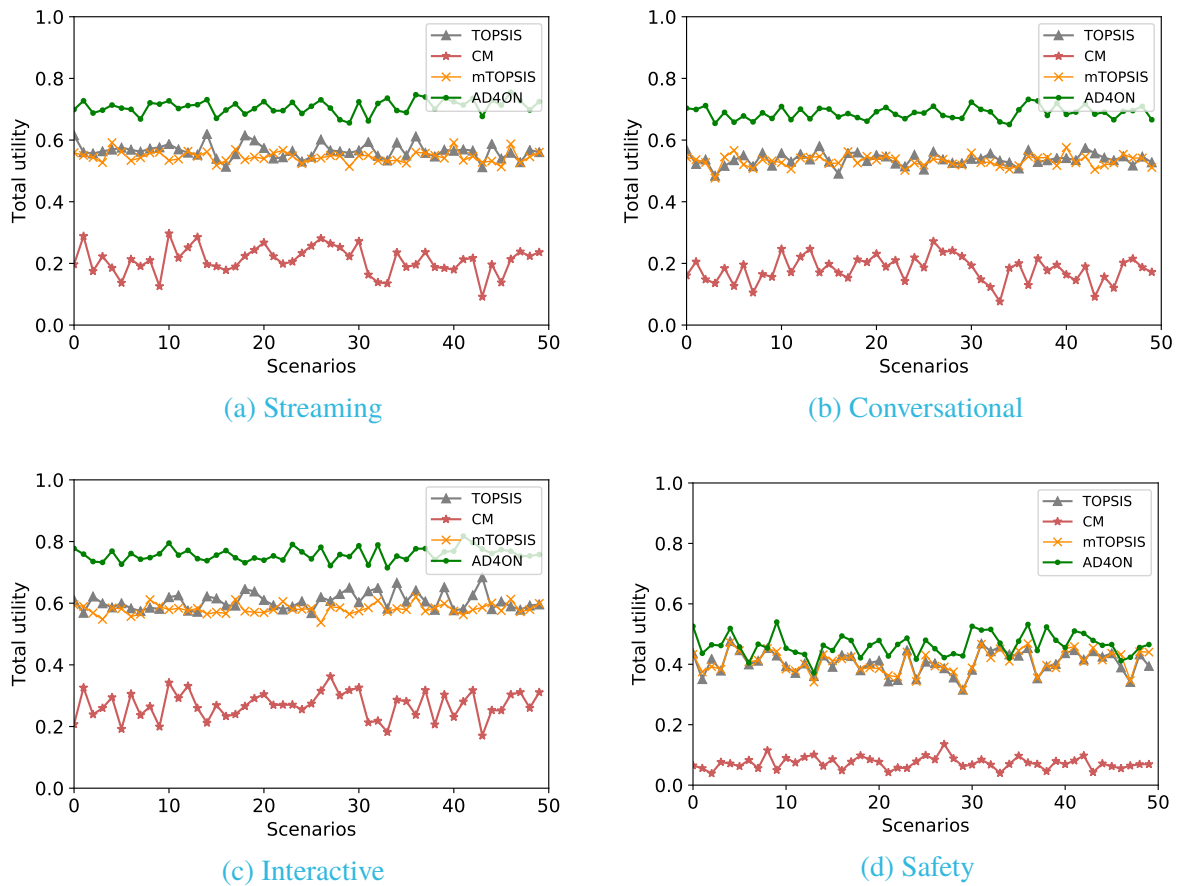


Figure 6.11: Total utility

most of sub-scenarios, and in the worst case, presents the same quality as TOPSIS or mTOPSIS solutions.

### 6.2.3. Scenario C

In order to test the AD4ON algorithm in a more complex scenario, i.e., in a scenario composed by more flows and more connection option, we defined the scenario C, which is composed by 50 sub-scenarios. In each sub-scenario, we consider a vehicle capable to connect up to twenty access networks simultaneously. In such vehicle, there are 20 flows that can start at different time.

#### 6.2.3.1. Flows inputs

We designed each sub-scenario with a fixed number of 20 flows, being four streaming, three conversational, three interactive, five background, and five safety. The flows start to communicate at different times while vehicle is moving. Such start time follows a Poisson distribution with

$$\lambda_{(f)} = 1/20.$$

At the begin of communication, the values of the flow requirements are randomly chosen from the range of values described on Table 6.1. Once a given flow starts to communicate, its requirements remains constant during the rest of simulation time.

For better understanding, we show below an example of flows for one of the 50 sub-scenarios. For the sake of simplicity, we show only the minimum DR required by each flow. However, the other flow requirements (i. e., latency and PDR) are also considered in the simulations. Table 6.7 shows the instant each flow starts the communication process, as well as the minimum DR required by each one in the *sub-scenario C1*.

Table 6.7: Example of flows of sub-scenario C1

Flow	scnFlow 1 start position (m)	DR (Kbps)
Background 5	10	399.68
Streaming 3	20	531.30
Streaming 4	20	770.84
Interactive 3	30	72.74
Background 2	40	892.43
Interactive 2	40	194.03
Safety 4	50	60.21
Background 3	60	859.62
Interactive 1	60	227.80
Streaming 1	60	204.13
Conversational 2	90	230.81
Safety 5	90	144.51
Safety 1	150	227.11
Conversational 1	170	128.30
Safety 3	200	26.66
Safety 2	240	75.03
Background 4	290	100.76
Streaming 2	370	734.37
Background 1	440	678.35
Conversational 3	800	120.26

As we can conclude by the table, the vehicle starts at position  $0m$  without communicating flows. When the vehicle is at position  $10m$ , the flow named *Background 5* requests the DM to find an access network capable to offer a minimum of 399.68 Kbps. The vehicle keeps moving and when it arrives at position  $20m$ , the flow *Background 5* remains communicating and two new flows (*Streaming 3* and *Streaming 4*) request access networks that are capable to offer a DR minimum of 531.30 Kbps and 770.84 Kbps, respectively. The vehicle is supposed to arrive at the destination point (position  $1000m$ ) with all 20 flows communicating or requesting communication.

Figure 6.12 overviews such flows request described in Table 6.7.

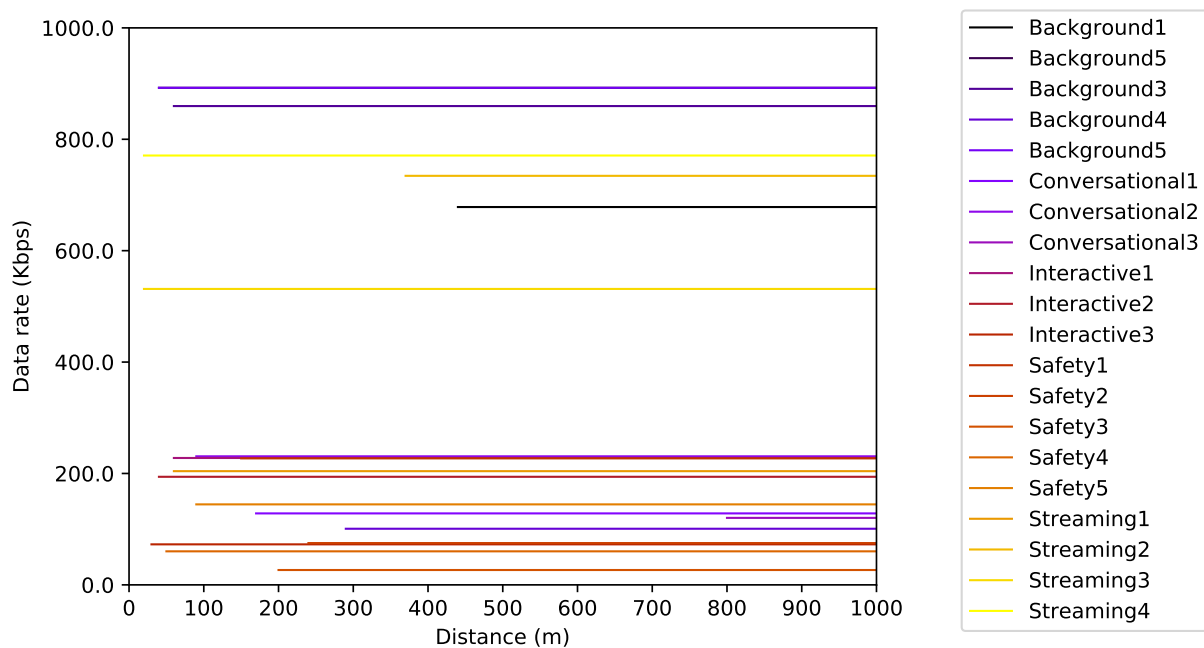


Figure 6.12: Example of DR required by the 20 flows in the sub-scenario C1.

### 6.2.3.2. Networks inputs

The number of access networks present in each sub-scenario is randomly chose. It can vary from 1 to 20 access networks. And like scenario A, the network parameters are based on real measurements on the field.

The network environment is generated as follow:

1. First, based on the measures realized on Murcia and Rennes testbed, we created ten network frameworks for each access technology, i.e., ten framework for vehicular WiFi (ITS-G5), ten for urban WiFi (802.11n) and ten for Cellular (3G). Each framework has information about the DR, latency, PDR, and RSSI along all access network coverage;
2. We randomly choose the number of access networks (per technology) should be present in each sub-scenario. Then, for each access technology, we randomly choose network frameworks among the predefined ones;
3. Finally, we place each network framework randomly distributed in the simulation area.

In Figure 6.13 we show the network environment (DR availability) for the *sub-scenario C1*, which is composed by a total of seven access networks (4 urban WiFi, 2 ITS-G5, and 1 cellular network).

The vehicle starts to move (at position  $0m$ ) under coverage of three access networks, i. e., one cellular network, one urban WiFi, and one ITS-G5. As the vehicle moves, the available DR from the different access networks vary as shown in Figure 6.13. The Annex 2 shows the number of access networks for each one of the 50 sub-scenario.

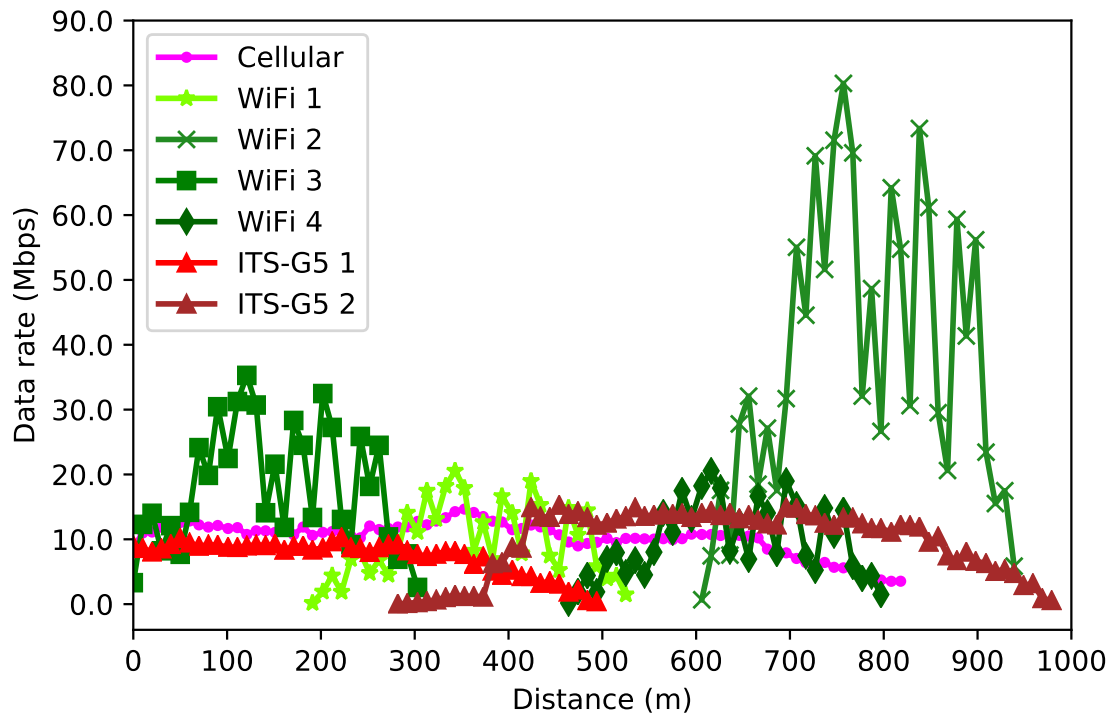


Figure 6.13: Example of network DR availability for the sub-scenario 1 composed by 7 access networks.

### 6.2.3.3. Simulation results of scenario C

The Figure 6.14 shows the different KPIs and the Total Utility encountered by each DM algorithm for the *streaming-1* flow.

From the Figure 6.14d we observe that AD4ON outperforms the other DM algorithms in terms of total utility. The AD4ON solution presents better or equal flow satisfaction than TOPSIS, but with less network switching, i. e., with a better decision stability than TOPSIS.

As for the other previous simulated scenarios (scenario A and B), the AD4ON presents a light underperformance related to the monetary utility, i. e., solutions found by the AD4ON maintain flows using costly networks (like cellular networks) for more time than TOPSIS. This is due to the AD4ON property in use pheromone concentration for a smooth network switch. Thus, avoiding “ping-pong” effect.

As we can observe in Figure 6.14b, the mTOPSIS algorithm presents the worst stability in most of simulated scenarios, underperforming even the traditional TOPSIS. As mentioned before, the



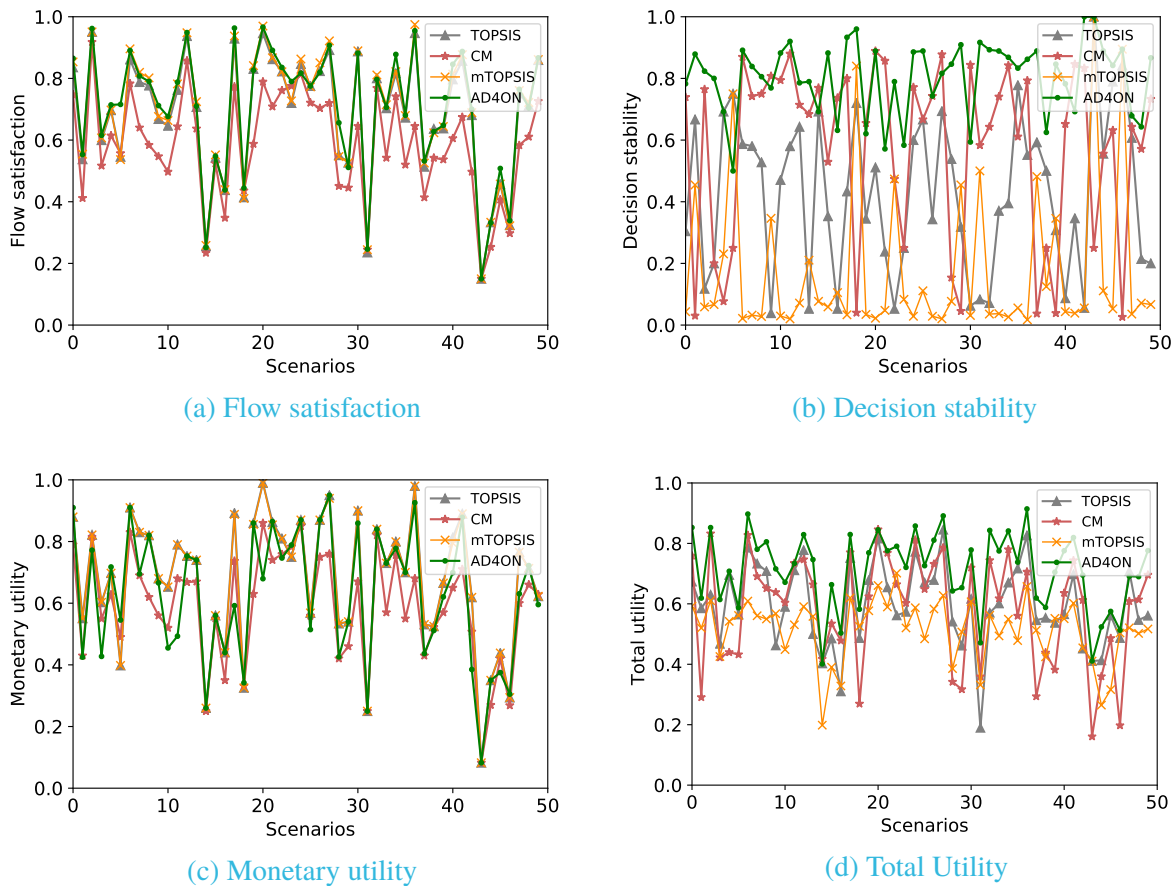


Figure 6.14: Results for scenario C - Streaming.

mTOPSIS normalizes the different requirements using utility functions that describe the intention of each actor. For example, the DR of an access network will be normalized by an utility function that describes the flow requirements. In this way, the normalized values will be 0 (zero) if it is lower than the DR required by such flow, reaching the maximum utility value (1) when the available DR is equal or greater than the maximum DR required by the flow.

In such simulation, the access networks capacities (in terms of DR, PDR and Latency) often overtakes the actors requirements. In other words, all access networks meet 100 % of flow requirements. Therefore, the choice of the access network for such algorithm becomes random and implementation dependent. Such a scenarios, makes the mTOPSIS unstable, and increases the number of network switching, as shown in Figure 6.14b. In the current mTOPSIS implementation, in cases where all potential solutions (i. e., access networks) have the same utility, the algorithm chooses the first access network present in the decision matrix.

### 6.3. Conclusion

In this chapter, we performed simulations to compare the AD4ON algorithm with three other DM algorithms: the well-known *TOPSIS*, a modified version of *TOPSIS* (*mTOPSIS*) and a *commercial DM (CM)* used in most of smartphones, in which decisions are based on predefined network priorities.

Simulation results demonstrate that AD4ON algorithm is capable to select access networks that better match with flow's requirements, while managing multiple attributes and objectives simultaneously. Moreover, it outperforms the other algorithms by increasing the total flow satisfaction, increasing the decision stability and consequently reducing the "ping-pong" effect.

As observed in the simulated scenarios, the AD4ON is capable to make decisions considering multiple attributes from different actors. In the simulations performed in this chapter, the AD4ON considered attributes like DR, Latency and PDR from flows requirements, maximum monetary cost users are willing to pay for data transmission, as well as DR, Latency, PDR, and RSSI about available access networks. The AD4ON algorithm combines such an information, in order to choose optimized solutions for the actors involved in the communication process.

AD4ON is designed to work with multiple flows and heterogeneous access technologies simultaneously. In such a heterogeneous scenarios, it is capable to find for each flow the access networks that better meets flow requirements.

Unlike *TOPSIS*-based algorithms, the AD4ON makes decisions independently of the flows order. In *TOPSIS*-based algorithms, the decision process starts in a given flow (in this thesis we randomly chose such starting flow) and it is executed sequentially until every flows are visited. In this way, depending on the starting flow the subsequent flows can have different solutions, which are not necessarily the most optimized one. On the other hand, in the AD4ON process, ants are scattered among the different flows and start the decision process in parallel. Therefore, different possibilities of solutions are explored before the AD4ON algorithm converges to an optimized solution. Such an AD4ON approach allows to choose access networks that better satisfy each actor.

In the AD4ON algorithm, decisions are driven by pheromone concentration based on the quality of solution. This approach increases the AD4ON decision stability. Unlike *TOPSIS* algorithms, the AD4ON does not immediately select a new solution if a given access network momentarily changes an attribute. This AD4ON property prevent unnecessary network switching, avoiding "ping-pong" effect.

However, this property does not allow the flows immediately take advantage of newcomer networks. When a new access network becomes available, the AD4ON algorithm maintains the flow communication through the current access network for a few period of time until the ants converge to the new access network. This reaction time can be reduced if the AD4ON is aware about the near future of the network conditions. Knowing the near future, the AD4ON can better

react to the network conditions.

Moreover, due to the high velocity of vehicles, their connectivity context can change frequently. In such highly dynamic mobility scenarios, besides the current vehicular context, it is needed to take into account the short-term prevision about the network environment in which the vehicle will be soon inserted. With this information, the AD4ON responsible to manage all data communication services is capable to anticipate decisions, allowing communicating flows take advantage of new networks as soon as they appear. We address such a short-term prediction in the next chapter (Chapter 7).

# 7

## Near Future

As vehicles move, they experience different network coverage. As mentioned before, due to the high velocity of vehicles, their connectivity context can change frequently. For example, while moving through different network environments, a vehicle experiences different network conditions (e.g., RSSI, DR, latency). A vehicle running in high speed can rapidly cross low-range networks (e.g., urban WiFi). In this way, an available access network can be soon unavailable, or the vehicle can rapidly reach new access technologies coverage.

Such highly dynamic mobility scenarios challenge current DM algorithms during the network selection process. DM algorithms like the TOPSIS, which try to find always the best access network for each flow, suffer of frequent network switching. In some case, presenting certain instability in decisions, i. e., presenting “ping-pong” effect. These frequent network switching can reduce the communication QoS, by increasing the latency and the number of packet loss. As we showed on the previous chapter (Chapter 6), the AD4ON algorithm performs more stable decisions. Despite such a stability, in highly dynamic scenarios the AD4ON takes longer to switch from an access network to another one. This delay prevents flows to immediately take advantage of new access networks.

For a better communication management, the DM should increase its knowledge about the upcoming network environment, i. e., it needs to take into account the short-term prevision about the network environment in which the vehicle will be soon inserted. With these information, the

AD4ON is capable to anticipate decisions and/or to prepare for the newcomers access networks.

If an onboard DM is aware about the network environment the vehicle is going to cross, such DM can better manage packets and flows to increase their communication utility. For example, the DM can increase the data buffer for a given video streaming, if the vehicle is going to cross a wireless dead zone; anticipate a network handover, by preparing the handover process in advance; or, delay a data transmission if it knows that a better network will soon be available.

Moreover, such a DM can inform applications about upcoming network environments. With these predictions, adaptive application, i.e., application capable to adapt its behavior at runtime according to communication conditions, can take different decisions. For example, an application can decide to delay a data transmission if it knows that a better network will soon be available. A mail client can delay attachment downloads until cheaper networks become available. In order to avoid service disruptions while vehicle is moving, a critical health monitoring application may prefer 4G to WiFi. Or a streaming application can react to connectivity degradation by reducing its video quality.

Netflix is an example of adaptive application that uses Dynamic Adaptive Streaming over HTTP (DASH) to transmit video streaming. DASH is a technique that enables adaptive bitrate streaming over Hypertext Transfer Protocol (HTTP) web servers. Therefore, if the DM is capable to anticipate information about future network environments, an onboard Netflix client can use such information in order to improve the QoE of the users.

The main challenges is the definition of how to acquire and how to integrate the near future prediction in the decision making process. And this is the objective of the present chapter. In this chapter we outline some possibilities in how to acquire near future information, and we demonstrate how to integrate such a future prediction in the AD4ON mechanism. Finally, we demonstrate the feasibility of such an integration, as well as the usefulness of the near future by simulating a simple scenario.

## 7.1. Recent research work

Based on the literature review, few works deal with the near future about network environments. Among them, some are based on the IEEE 802.21 standard as a cross layer interface to monitor and interact with heterogeneous technologies [102, 28, 33]. In this process, vehicle exchanges information with specific network entities to acquire information about future access networks. With such information, the DM can for instance, assist proactive authentication, enhancing actual availability of targeted networks. However, as mentioned before, the IEEE 802.21 standard supposes the implementation of specific MIH functionalities in the access technologies and

networks.

Ericsson performed a research indicating that almost 70% of all mobile data traffic will be from video in the coming years [103]. Based on this Ericsson research, Volvo and Ericsson proposed an intelligent media streaming for cars. This system aims to deliver high quality uninterrupted viewing experience in Volvo cars. The intelligent streaming system consists in predicting the route and look ahead at network conditions. Based on such information customized media list is created, enabling users to select content tailored to the duration of their trip.

Another work related to short-term prediction is the “predictive connectivity manager” announced by Continental in 2018 [104]. This service predicts network availability and reception quality along the vehicle route. Each vehicle equipped with the continental module collects data on the availability and quality of the communication channels, and sends it with its current position (GPS) to the Continental cloud. These centralized data are processed by a back-end service on the cloud, which creates a map containing a set of information (like availability of different networks, signal strength, bandwidth and latency) by position. Based on such information and on the knowledge of driver’s itinerary, the vehicle can predict the networks performance along its route. However, such solution does not define how to use these predictions in the decision making process.

Based on recent researches, it is possible to highlight the current interest in knowing more about the near future of network conditions, and how to use such an information to improve the vehicles’ communication.

## 7.2. How to acquire the near future

A wide variety of useful information can be predicted for improve decisions in ITS communications. Recent works have addressed prediction in different ways. Some addressed predictions about the future service requirements. Others have addressed the prediction about networks condition, e.g., prevision about latency, PDR and available bandwidth. While others have been interested in mobile motion prediction.

Motion prediction methods usually use information like vehicle speed, direction of the mobility, traffic in a given area, whether and road infrastructure, in order to predict a future vehicle’s position. For example, based on vehicle speed and direction it is possible to predict the vehicle position in a certain instant. Or based on context information like whether (e.g., if it is raining or not) and road infrastructure, the vehicle can predict the traffic in a given region and consequently, estimate the time necessary to cross them to arrive at its destination. The vehicle’s route can also obviously be obtained from the navigation system.

Despite these wide varieties of useful information, in this thesis we talk about flows and networks prediction. In fact, we talk quickly about prediction of flows requirements, and we focus on the near future about network environments.

### 7.2.1. Near future about flows

Vehicles can run a variety of applications simultaneously. These applications can have one or more data flows, each one with specific communication requirements. As mentioned before et addressed in our previous work [105], for a better communication management, the DM should manage communication flow by flow and choose the network that better match with flows requirements.

Therefore, for proactive decisions it is necessary to predict both future flows arrival, i.e., flows that are not currently communicating but have high probability to start in the near future, and future flows requirements, i.e., previsions about new requirements from current communicating flows.

These predictions about flows can be achieved by different ways. Administrators can define it in a static way. It can be achieved by applying data traffic models, in which functions describing different service are used to estimate future service requirements. It can also be achieved by defining user usage patterns, i.e., based on the historic of users, it is possible to predict the instant a given application is likely to be used. Most works in the literature, define flows requirements at the begin of simulation and consider flows constant during all simulation process. Some others define service patterns based on recorded historic of mobiles [106, 107, 108].

With these predicted information, the DM can make decisions in order to better satisfy the whole communication system, i. e., it can make a better load balancing by distributing flows among available networks. The DM can better allocate resources (protocols, and access network) for coming flows.

### 7.2.2. Near future about networks environments

To enable the DM anticipate decisions in order to improve the overall communication, it is necessary that the DM have, in advance, information about the network environment in which the vehicle will be soon inserted. It is important to highlight that we mean by network environment prediction the average values of each parameter describing the access network in a targeted area. The network environment is composed by one or more access technology covering a given area. Foresee such a network environment means predict the average value of each parameter like the DR, latency, PDR, and RSSI for each access network in that area.

The short-term prevision about the network environments, for which the vehicle is going

to cross, can be obtained in different ways. We classified the network acquirement process as illustrated in Figure 7.1. The vehicle can predict future network environments based on its own perception or by collaboration with its environment. In the direct perception, vehicles can learn with their previous experience and use this learning to predict the future. While in the collaborative mode, the vehicle use the communication with other connected devices (vehicles, infrastructure and network) to exchange information about the upcoming network environments.

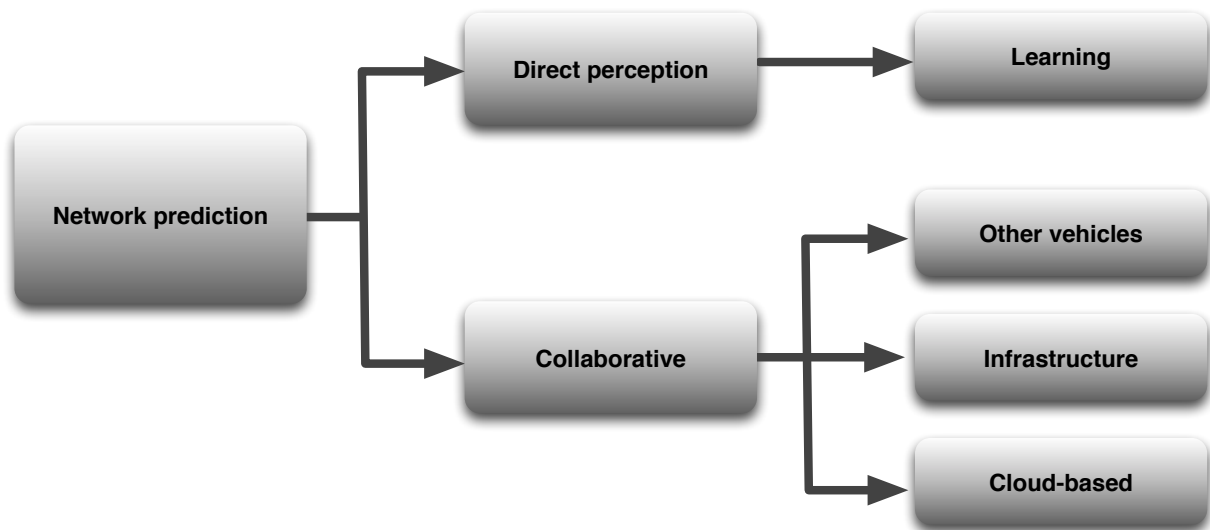


Figure 7.1: Classification of network prediction

Below, we describe these classes of perception, and we illustrate how research works have addressed the prediction of network environments.

### 7.2.2.1. Direct perception

In C-ITS, network monitoring modules from embedded DM mechanism are supposed to be constantly monitoring available access network while vehicle is moving. In this case, vehicles can store locally all acquired network information. The DM mechanism can use these historic information in order to make prediction. For example, an user who uses the same route every day, the database could stores information about network conditions in such route and uses it to predict networks conditions in the next time the user pass by this route. Therefore, each time the user pass by this route the DM can predict networks conditions based on the historic data, i. e., information stored from previous passages.

### 7.2.2.2. Collaborative



In C-ITS environment vehicles have the capacity to communicate with other connected objects. For example, vehicles can communicate with other vehicles (V2V), and the infrastructure (V2I). Therefore, vehicles can use such a cooperation to exchange information about the upcoming network environments. We split the collaborative-based prediction in three modes, according to who the vehicle communicates.

### **Cooperation with local vehicles**

Vehicles can use access technologies like ITS-G5 or 5G-PC5 in order to communicate directly with other vehicles (V2V communication). Such a communication with neighboring vehicles, enables information exchanges about network environment. For example, two vehicles in opposite directions can exchange information about network access points and/or networks condition they have experienced. For this purpose, a vehicle stores the position and performance of each access point in its past route, and give them to vehicles coming in the reverse direction. In this way, the vehicles are aware about network environment in their upcoming route.

In ITS architectures, LDM is a key feature to facilitate access to relevant commonly used C-ITS information. Such conceptual data storage entity covers information on nearby moving vehicles, and it can be used to store static and dynamic parameters. It can store information about available radio accesses, traffic road signs, road maps, information about running applications (QoE, QoS, user preferences, timing restrictions, etc.). Based on LDM information, cooperative vehicles can forecast the environment it is going to cross.

Similar to other connected devices, an important issue in the vehicular collaborative communication is the security and privacy. Nowadays, vehicles have hundreds of embedded computer controls and millions line of code. Such a system can be hacked and exploited to cause physical injury or death. Therefore, for secure vehicular communication, it is necessary a strong cybersecurity environment. Critical vehicle systems, i. e., systems that govern safety (like braking systems, and engine systems) must be protected from unauthorized access and/or harmful attacks.

Despite the V2X communication brings positive benefits and features, the privacy is a concern. Efforts have been done to ensure privacy into the data collection for V2X, e. g., rotating certificates, and forbidden vehicles to transmit any personally identifiable information within the basic safety messages (e.g., names, license numbers, VINs).

As already mentioned in Section 4.3 we do not propose a new security module, instead, we consider to use existing security solutions like IPsec and the data encryption provided by NEMO/MCoA tunnels.

### **Cooperation with infrastructure**

In the literature, cooperation with network side is the most used method to acquire near future information about network conditions [25, 28, 35]. Specific entities in the network side provide mobile devices with important information from the radio environment. With information like change in the state of a given interface or a predictive event indicating future change in the connectivity, the mobile device can perform accurate and proactive decisions.

For this purpose, the IEEE 802.21 standard proposes a media independent handover framework developed to assist handover between heterogeneous networks. The standard defines specific functionalities that are supposed to be implemented and supported on the mobile device and in the network side. For example, a service provides information about dynamic changes in link-layer characteristics, another service provides network information within a geographical area, and a last one enables upper layers to control the physical and data link-layers for optimal performance.

### **Cooperation with a cloud-based predictor**

Another way to predict network conditions can be achieved by cooperation with a centralized database. Vehicles equipped with standardized modules can collect data like bandwidth, latency, signal strength and network availability all along their route. This data are combined with GPS coordinates and sent to a centralized database in the cloud. In this way, all data from different cars can be stored in the cloud and shared with other traffic participants.

Such a prediction mode, benefits from a large amount of information, i. e., information from a large number of vehicles, which enables the use of algorithms like AI algorithm to predict network environments.

In real implementation, each car manufacturer can implement its own cloud, monitoring its vehicle fleet. Or we can imagine a centralized system capable to receive information from vehicles of different brands. However, the real implementation of a centralized network environment predictor is challenging in terms of infrastructure (e. g., the database size, the amount of DR such a centralized system should support) and it is challenging also in terms of administration, i. e., it is necessary to define who will be in charge of such a centralized system (e. g., the government, or a third-party operator).

An example of cloud-based prediction is the “predictive connectivity manager” realized by Continental [104], as previously described on Section 7.1. In this example, the cloud is private and operated by Continental.

## **7.3. Decision Making using near future**

Once the DM has all predicted information, it can use that to improve its decisions. In this chapter, we consider flows requirements constant during all decision making process and we take into account only the near future information about network environment.

The decision making process considering the near future is not trivial. Besides the heterogeneous network environment covering the vehicle in a given instant, the DM should take into account future network environments. However, the definition of future network environments is not clear, i. e., there is not a definition about what can be considered as relevant future network environments. In the following, a near future classification is given.

#### 7.3.1. Near future classification

First of all, it is necessary to define the notion of “future network environment”. This chapter considers as current network environment of a given vehicle, the set of heterogeneous access technologies covering such vehicle. Therefore, it can be deduced that future network environments are set of heterogeneous access technologies, which are supposed to cover the vehicle in future times.

The predictions of future network environments are dependent of vehicle speed. For example, two vehicles moving at different speeds toward the same future network environment will reach such environment at different time. Consequently, they are supposed to experience distinct network conditions. In this context, these two vehicles should have different predictions, i.e., they will predict the same network environment but at different instants.

Moreover, the confidence of predicted values are dependent on the distance the vehicle is from such future. For a better understanding, we consider the same example given before about two vehicles exchanging their network experiences. We also consider that one of these two vehicles stored network information in two points of its trajectory: a point “B” it crossed few hours ago, and a point “A” it just passed few minutes ago, as illustrated in Figure 7.2. The vehicle receiving these two measures cannot consider both information with the same weight in the decision making process. Values from the network environment at the point “A” tend to be closest to the current networks condition than the values acquired at point “B”. Therefore, such information from “A” are more pertinent in the decision making process. In other words, the probability that the vehicle finds, at the point “A”, a network environment close to the one announced by the other vehicle, is greater than the probability to arrive at point “B” and find the same network environment measured few hours ago.

This chapter classifies future network environments based on the position between the current network environment and the future one, as shown in Figure 7.3.

These environments are classified in four levels, following a time scale, i.e., according to the

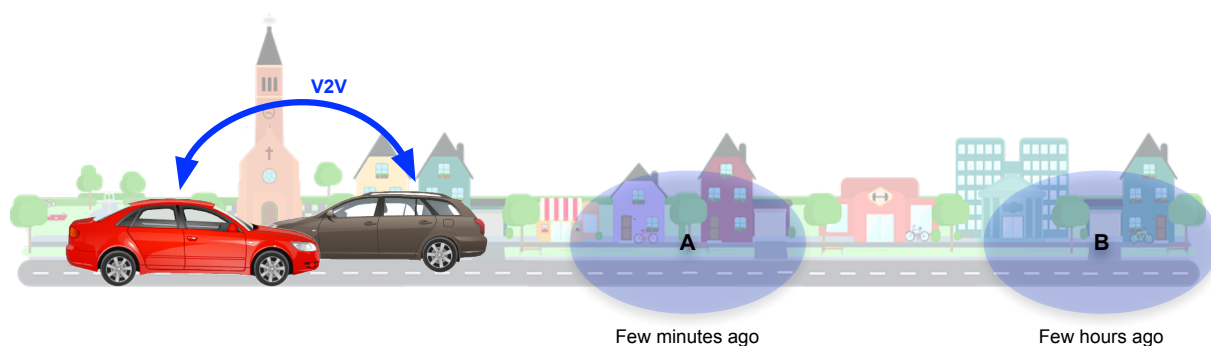


Figure 7.2: Example of prediction confidence

vehicle position in a time interval [109]:

- Very short-term (e.g., around 10 seconds): are environments near to current vehicle position. The vehicle can reach these environments in few seconds, e.g. networks around 10 seconds from the current vehicle position;
- Short-term (some minutes away): are environments minutes away from the current vehicle position;
- Medium-term (e.g., around 10 minutes away): are environments that vehicle can reach in dozens of minutes moving;
- Long-term (e.g., a given date): are environments that can be reached in one hour or more moving. For example, if a vehicle will arrive in its home at given date/time, it is possible to delay a system update, or the download of a mail attachment.

Based on the previous classification, the DM should take into account the uncertainty of predicted data. Predictions of previous defined environment classes present different levels of uncertainties, which are related to a variety of reasons. Besides the time in which information were predicted, predictions depend on the information source, i. e., the source reliability. Moreover, they depend on the methodologies used to predict the networks information.

When the DM receives information from a future environment, it is not guaranteed that the vehicle will find the same predicted performance when eventually reaching such environment. And this can happen for different reasons, networks load may have changed. Or access networks

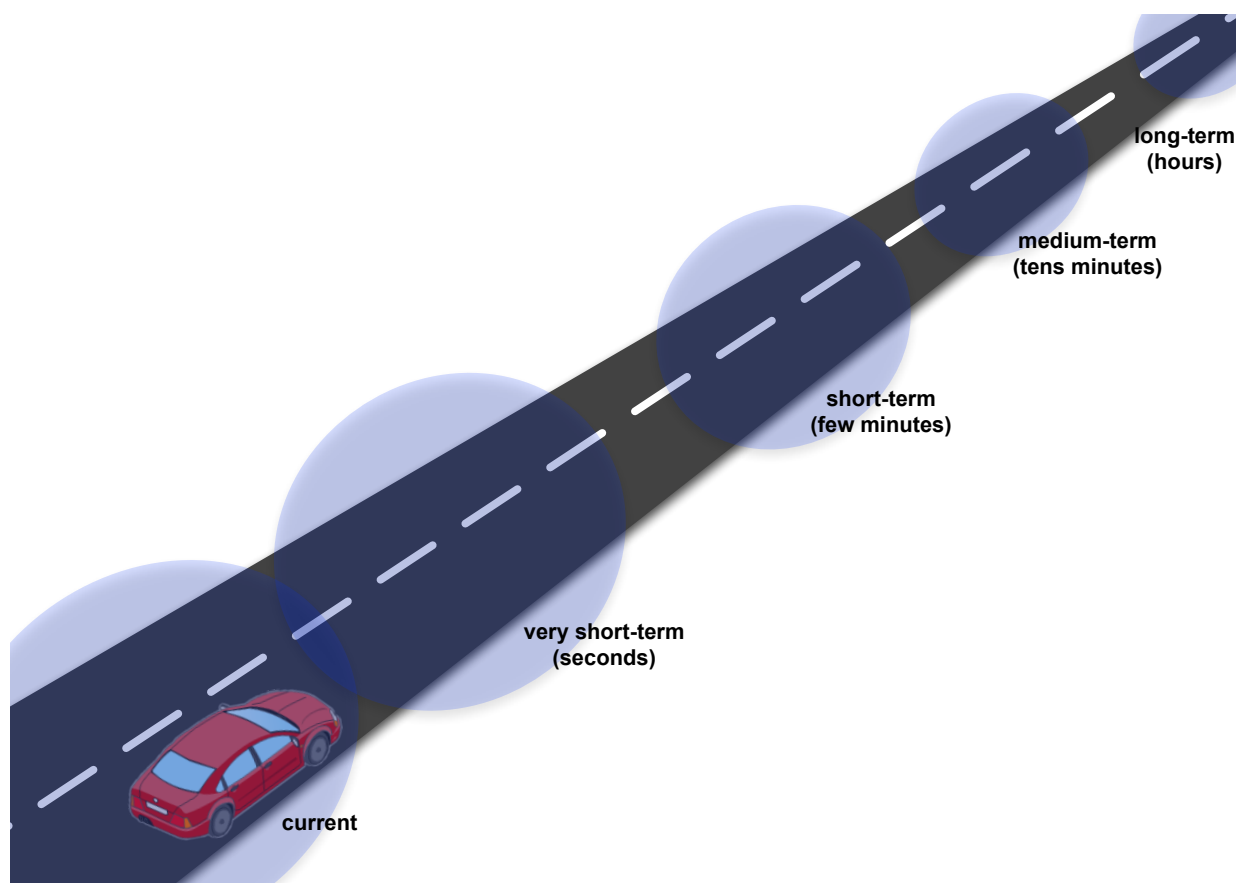


Figure 7.3: Classification of near future environments

may have been disconnected. Therefore, recent previsions tend to be more accurate than older ones. In this way, information from “very short-term” environments tends to be closer to real values than information from “long-term” environments.

#### 7.3.2. How to apply the near future

As we saw in the previous section, there are different types of future, each one with different level of confidence. Once the DM has the predicted network conditions, it is necessary to know how DM can apply its decision in the communication system. As described before, the decisions can be applied at different layers of computing system (Application, Transport, Network and Link layers). Below we highlight some enforcement the DM can perform.

##### 7.3.2.1. Management of adaptive applications

Adaptive applications are applications capable of changing their behavior at runtime, as a response to some changing conditions [110, 111]. Therefore, making adaptive applications aware about the

network environment can contribute to smarter network management.

In this process of adaptation at application level, DM plays the role of information source. It informs applications as soon as a significant mobility-related event occurs, while decisions are performed by each application. This communication between DM and Application layer is performed towards the standardized interface “MA-SAP”. The framework of this interface was described before in Section 4.2.

However, in some cases applications are not adaptive, or do not have communication management capabilities, or just do not want to be aware about communication processes. Therefore, the DM can make the decision and applies it, on behalf of such applications. For that, it interacts with controlled entities in all layers of the ITS-S communication architecture, in order to select the collection of facilities, transport and network protocols as well as access technologies that better match with flow requirements.

#### **7.3.2.2. Management of interfaces**

In the communication management, enforcement modules controlled by the DM are responsible for the activation/deactivation of the network interfaces and/or reconnecting such interfaces to other networks. Such interface management can be performed by different reasons. For example, an interface can be reconnected to apply a new DM solution, i.e., when the DM chooses a new best solution for a given flow. An unused interface can be deactivated to save energy. Or an interface can be activated to meet the demand of a new important flow.

If the DM knows about the network environment it will be inserted in a near future, it can proactively manage the network interfaces. For example, if the DM predicts the instant a given frequency will be available, it can inform controlled modules at layer 2 (from the OSI model), in order to start the interface configuration process as soon as such network becomes available.

#### **7.3.2.3. Management of flow routing and scheduling**

To apply the decision at network layer, the DM communicates with controlled entities at the “Network & Transport” layer from the ITS-S communication architecture. Such communication is performed towards the standardized interface “MN-SAP”. Such an enforcement enables the DM to use multiple access technologies simultaneously, select the most appropriate communication profile transparently to the application, as well as better management of network mobility. The DM can trigger different actions for ascending flows, for example:

- Drop packets: In case of loss tolerant applications, the DM can drop few packets in order to reduce bandwidth usage and accommodate more flows;

- Delay flows: It can delay data packets for a short period of time, if it knows that a better network will be available before the maximum latency supported by application. A flow can be delayed for different reasons. It can be delayed to improve its own satisfaction, i.e., to wait for a network that better meet its requirements. It can be delayed due to lack of networks capacity. Or, to improve the overall system performance.

Usually, flows are not aware about such communication management, unless they explicitly need it. Therefore, in case of delaying a flow, the DM should manage all processes, including buffering data packets.

### 7.3.2.4. Flow-Interface mapping

Once the vehicle is equipped with multiple network interfaces. The DM can distribute flows through such interfaces. Nowadays, there are tools that enable to monitor network interfaces, e.g., `ifstat` that is found by default in some linux distribution like Ubuntu, Debian and Fedora. This tool reports the network bandwidth for network interfaces. However, such a monitoring tools can provide DM only with current network interface conditions. In this way, even if a network interface will have a low performance in few time ahead, the DM is not aware and it risks to send flows through such network interface.

In such a scenario, knowledge about the performance of future network environment can help the DM to better decide which network interface it should map a given flow. For example, if the DM receives the information that a given access network will not have a good performance in the near future, it prevents to use the network interface connected to such access network.

## 7.4. Integrating prediction in the AD4ON

With all aforementioned information about current and future network environments, a DM is capable to make smarter decisions. It is capable to anticipate undesirable events, making flows connected in an ABC approach.

To integrate the near future in the AD4ON mechanism, we proceed as follow. Since there are multiple near future classes, we defined multiple instances of AD4ON algorithm, each one managing a class of near future environment. In this way, we have an AD4ON instance responsible to manage decisions at the current network environment, while other instances make decisions considering future network environments, i. e., one AD4ON instance for the very short-term environment, other for the short-term, and so on. All these AD4ON instances run continuously and in parallel.

In vehicles, as the onboard computer units are usually limited in terms of Central Processing Unit (CPU) and memory, we run such parallel AD4ON instances with resources scaling, i. e., we allocate more resources for the AD4ON instance making decisions for the current network environment. While for the instances considering future network environments, we reduce the resource consumption by reducing the number of ants participating in the decision process.

The AD4ON instance responsible to manage decisions at the current network environment takes into account both current measured network conditions and the decisions performed by the other AD4ON instances. Therefore, the instances executed with predicted values serves as input to the AD4ON instance being executed for the current network environment.

As described before, near future classes have different level of confidence. Therefore, the AD4ON of current network environment should take this confidence into account, i. e., decisions performed considering short-term environment are considered more confident than decisions performed with long-term environment.

This approach of parallel AD4ON instances is useful to increase the overall communication performance. The AD4ON mechanism can use decisions from different instances to anticipate actions. For example, it can use decisions performed with long-term environment to inform adaptive applications, as described by [49]. In this way, adaptive applications can adapt their operation.

In the following, we discuss how we integrate the near future information in the AD4ON algorithm.

#### **AD4ON-NF**

In order to integrate the near future information in the AD4ON mechanism, we create the AD4ON with Near Future (AD4ON-NF) algorithm. The AD4ON-NF is an evolution of the AD4ON algorithm that takes into account the near future about network condition. In this first algorithm version, we do not consider future information about flows requirements. Instead, flows are defined at the begin of simulation and remains constant during all simulation process.

For the sake of simplicity, we opted to develop only two AD4ON algorithm instances. Therefore, the AD4ON-NF is composed by two instance of AD4ON that run in parallel. A first instance (AD4ON-NF1) finds solution based on the current network conditions and apply an adaptive evaporation rule based on the solutions found by the second instance, i.e., the AD4ON-NF2. While the AD4ON-NF2 finds solution for predicted values of very short-term network environment.

The current implementation aims to improve the algorithm reaction time when a better access network becomes available. First of all, it is necessary to define the time window for the near future, i.e., how much time ahead should be considered by the AD4ON-NF. Once defined the time



window, the two instances start to run. The AD4ON-NF2 is not executed for a specific position in the future, instead it uses the average of each predicted network parameters inside the time window. Figure 7.4 shows an example of such approach.

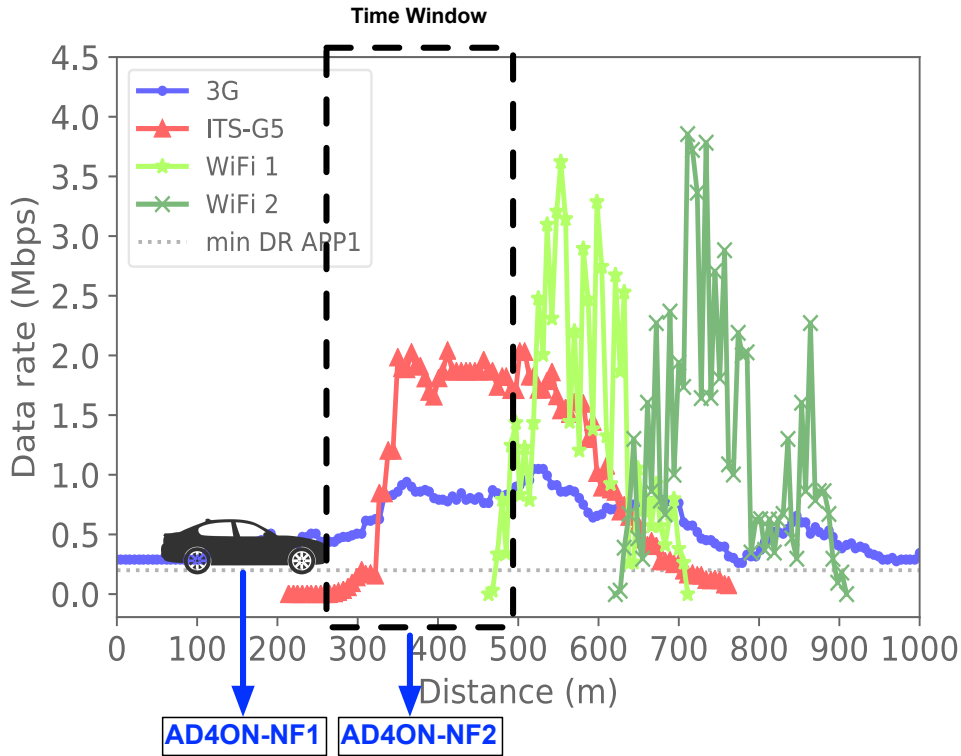


Figure 7.4: Example of AD4ON-NF approach.

At the begin both AD4ON-NF1 and AD4ON-NF2 instances start at the same time. And while the AD4ON-NF2 does not find a solution, the AD4ON-NF1 applies the same evaporation rules defined on Chapter 5, i.e., evaporate a predefined percentage of the pheromone concentration.

When the AD4ON-NF2 found a future solution, it sends the results to the AD4ON-NF1 that uses such results to adapt its evaporation rule. If the current solution (the one found by the AD4ON-NF1) is the same solution found by the AD4ON-NF2, i.e., the current access network still being the better solution in the future, we apply the evaporation rule defined on Chapter 5. Otherwise, we apply a more severe evaporation rule. If the current access network will not be a good solution in the future (e.g., it will not be available in the future), the AD4ON-NF1 uses a new coefficient of evaporation ( $\rho_{new}$ ) as defined by Equation 7.1.

$$\rho_{new} = \rho + \left( \frac{(1 - \rho)FS'}{2} \right) \quad (7.1)$$

where  $FS'$  is the future flow satisfaction calculated by the AD4ON-NF2 ( $0 \leq FS' \leq 1$ ). And  $\rho$  is the pheromone evaporation rate predefined in the AD4ON algorithm ( $0 < \rho < 1$ ).

From Equation 7.1, we observe that more the flow is supposed to be satisfied in the future, more

the evaporation is severe for access network that are forecasted as sub-optimal. The maximum evaporation rate is reached when the future flow is 100 % satisfied ( $FS' = 1$ ). In this case, the new coefficient of evaporation will be equal to  $\rho$  plus half of the distance between  $\rho$  and the maximum evaporation rate (i. e., 1.0). In the other hand,  $FS' = 0$  means the flow will not be satisfied by any access network in the future. Therefore, the AD4ON-NF1 applies the evaporation rules defined by the AD4ON algorithm in Chapter 5.

We implemented such algorithm version in order to evaluate the benefits of considering the near future in decision making process. Simulations with the AD4ON-NF algorithm demonstrate that, taking into account the short-term prevision about the adjacent network environment, the AD4ON-NF can optimize the algorithm reaction time. The simulations results are described on Section 7.5.

The capability of taking into account short-term predictions has to be integrated in the ITS standards. However, this is not straightforward and will require a proof of concept and performance study to be pushed in standardization.

## 7.5. AD4ON-NF evaluation

In order to evaluate the performance of the AD4ON algorithm when taking into account the near future, we implemented the previous described AD4ON-NF algorithm. And we performed simulations with a simple scenario. For comparisons purpose, we used one of the scenarios previously simulated with the AD4ON algorithm on Chapter 6, i.e., the scenario A.

The objective of this simulation is demonstrate that knowing the near future, the AD4ON is capable to adapt its behavior in order to find solutions that better satisfy the flow communication.

### 7.5.1. Scenario A with Near Future

The AD4ON-NF simulation is performed using the same scenario A described on Chapter 6, i.e., a scenario composed by one application flow (*App1*) and four access networks (ITS-G5, 3G, WiFi 1, and WiFi 2).

For the sake of clarity and ease of reading, we show again the flow and network environment on Figure 7.5. This is the same figure already described on Chapter 6 (i.e., Figure 6.8).

Using this simple scenario, we aim to compare the AD4ON and the AD4ON-NF algorithms. Therefore, while vehicle moves along the defined route, we choose solutions using both algorithms. For the AD4ON-NF we use a time window of 10 seconds, representing the very short-term predicted area. Such value of time window represents a displacement of around 55 meters for a

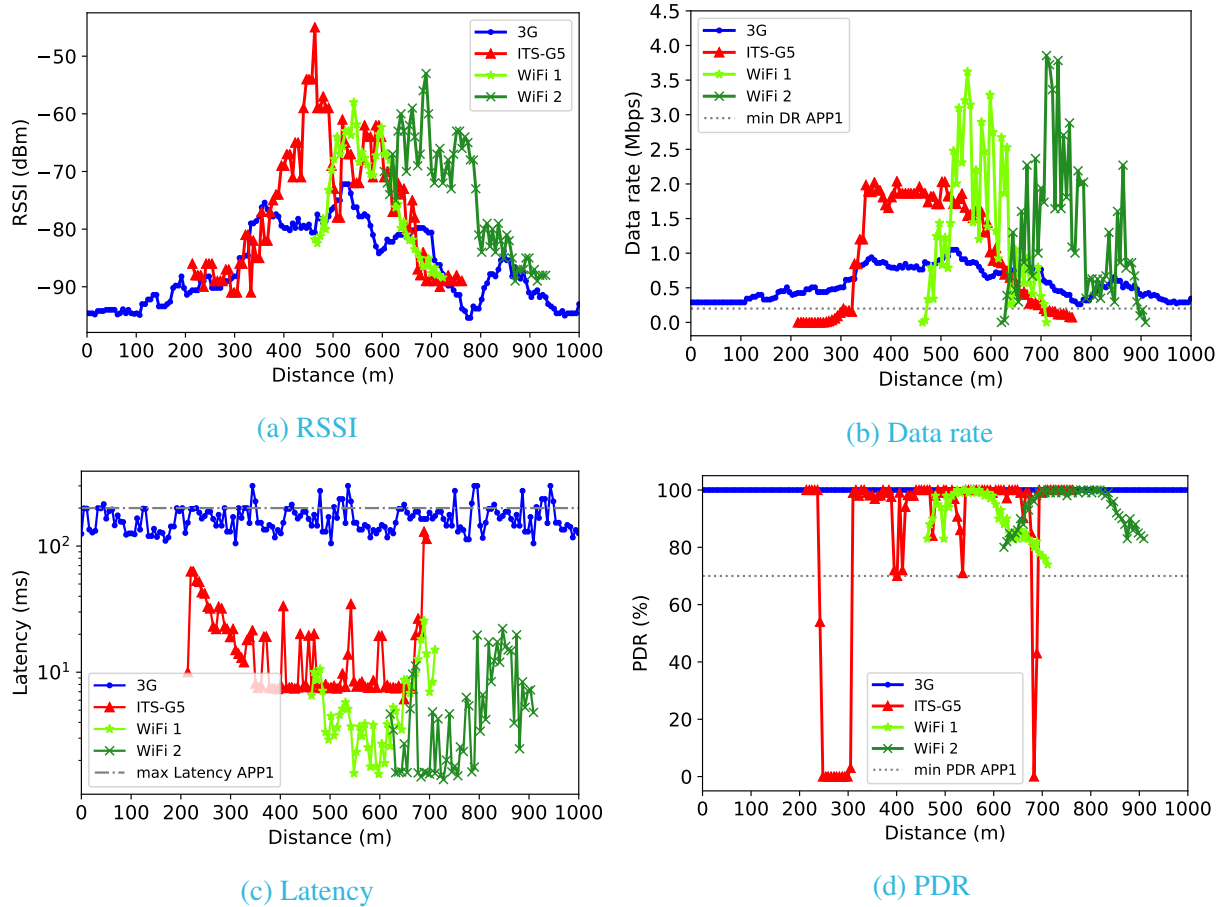


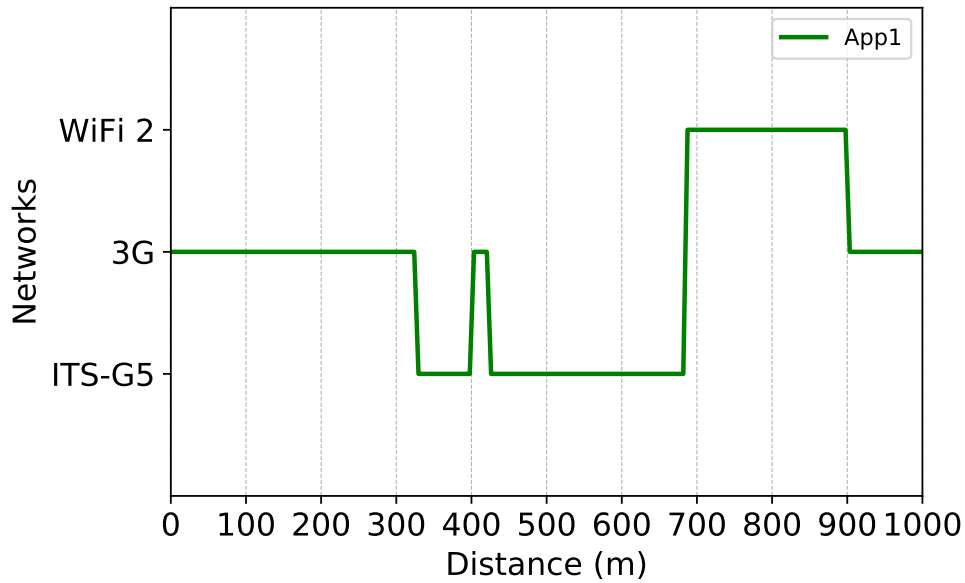
Figure 7.5: Input of flow and network parameters for scenario A.

vehicle moving at the speed of 20 Km/h or around 83 meters for a vehicle at 30 Km/h. In such a vehicle's movement, we can observe important variation of network conditions.

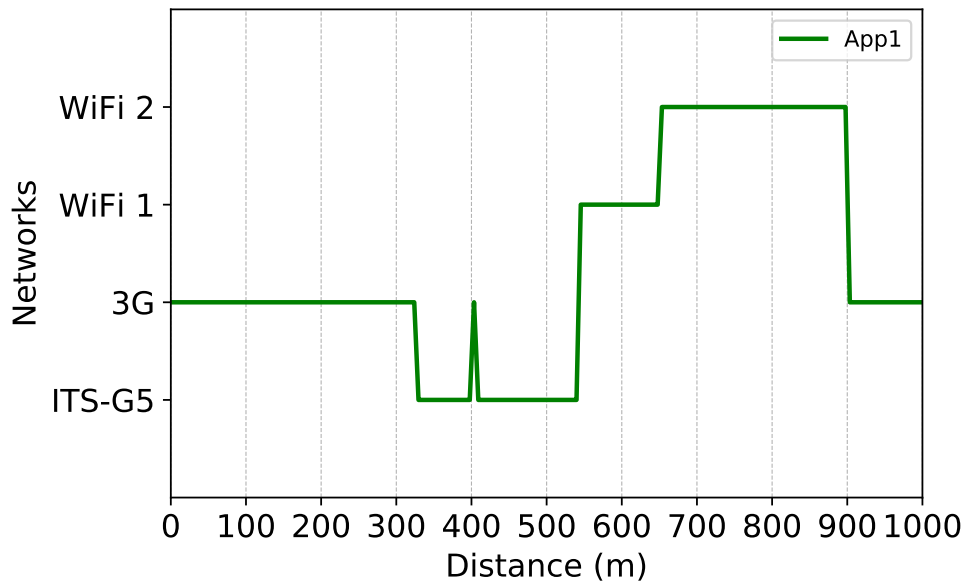
Figure 7.6 shows both the result of AD4ON algorithm (Figure 7.6a), i.e., without knowledge about the near future; and the result of AD4ON-NF algorithm (Figure 7.6b), i.e., taking into account the near future information about network conditions.

Comparing Figure 7.6a and Figure 7.6b, we observe that unlike the AD4ON, the AD4ON-NF solution includes the WiFi 1. As mentioned on Chapter 6, due the AD4ON reaction time, the AD4ON does not switch to WiFi 1. Since the AD4ON-NF is aware about the networks condition in the near future, it better manages the pheromone concentration. It anticipates the pheromone evaporation based on the short-term prediction about the network condition. Therefore, knowing the WiFi 1 condition in advance, enables the AD4ON-NF to adapt the pheromone concentration in order to switch faster for such a new solution.

We observe the same behavior at the position of 400 m. At that position, the only network that satisfy the flow requirement is the cellular one. The ITS-G5 presents a PDR lower than the minimum required by "App1". Therefore, both DM algorithms switch to 3G. However, due its adaptive evaporation rule, we observe that the AD4ON-NF stay less time in the cellular network



(a) AD4ON



(b) AD4ON-NF

Figure 7.6: Comparison between AD4ON without near future and AD4ON considering near future information

than the AD4ON. In this way, the flow can take more advantage of newcomer networks.

Table 7.1 shows the KPI for both AD4ON and AD4ON-NF. We use the same KPI used in Chapter 6. We can observe that, by taking into account the near future about network conditions, enables the AD4ON-NF to choose solution that increase the flow satisfaction and the user's preference (i.e., reduce the monetary cost, as required by the user). The decision stability is based on the number of network switching. Since, the AD4ON-NF uses one more network than the

AD4ON, i.e., it makes one more network switching, we observe that for this scenario its decision stability indicator is slightly lower than the AD4ON.

Table 7.1: Key performance results for AD4ON and AD4ON-NF

Algorithm	Flow satisfaction (%)	Monetary utility (%)	Decision stability (%)	Total utility (%)
AD4ON	62.22	73.76	66.67	67.55
AD4ON-NF	63.36	74.52	60.00	65.96

## 7.6. Conclusion

Due the highly mobility of vehicles and the frequent change in the network environment, DM algorithms that does not consider the near future in their decisions, present some issues. For example, TOPSIS suffers of decision instability and should perform full recalculation every time a network parameter change.

As described on Chapter 6, the AD4ON algorithm presents a reaction time that does not allow the flows immediately take advantage of newcomer networks. When a new access network becomes available, the AD4ON algorithm maintains the flow communication through the current access network for a few period of time.

Therefore, in such highly dynamic mobility scenarios, besides the current vehicular network environment it is useful to predict the network environment in which the vehicle will be soon inserted. In this chapter, we discussed about the use of near future information in the decision making process, in order to perform smarter decisions. And we developed a version of the AD4ON algorithm that takes into account such near future information (AD4ON-NF).

Simulations with the AD4ON-NF algorithm demonstrate that taking into account the short-term prevision, the AD4ON-NF can optimize the algorithm reaction time. Knowing the future network environment in advance, enables the AD4ON-NF to adapt the pheromone concentration, in order to switch faster for a new solution that better satisfy the communication requirements.

The simulation performed in this chapter used a very simple scenario. And the objective was to have a proof of concept of feasibility to consider the near future, as well as a demonstration of better decisions when the near future prediction is considered. More simulations should be performed in order to better analyze the AD4ON-NF algorithm.

The AD4ON-NF does not consider future information about flows requirements. Since flows can start and stop communication any time during the vehicle journey, as well as change they

requirements, as future works we aim to simulate the AD4ON algorithm considering both near future about flows requirements and network environments.

It is important to simulate the AD4ON-NF with different algorithm instances running simultaneously, i. e., executing an instance of AD4ON-NF for short-term predicted class, another instance for the medium-term class, as well as an algorithm instance for the long-term class. Then, the AD4ON-NF that makes decision based on the current vehicle network environment, can take the results of all these other algorithm instances, in order to improve its decisions. For example, it can use the results from the short-term and medium-term environment to better decide about a network switching or to improve the data queue management. Or it can use result from the long-term environment to inform adaptive applications about the predicted network environment, enabling such applications to anticipate and better adapt their behavior when the vehicle arrives in that area.

In the present chapter, we simulated using a time window of 10 seconds. Since in the algorithm we consider the average values inside the time window, large time window tend to hide the real network performance, i. e., the average can hide a instantaneous low performance. On the other hand, shorter time window will highlight specific behavior of each access network, but we cannot over reduce the time window to the point of consider only instantaneous parameter values. Otherwise, we risk to take into account a peak value that does not represent the reality. Therefore, as future work we aim to simulate the AD4ON-NF algorithm considering different size of time window (e. g., 0.5s, 1s, 2s, 3s and 5s). In this way, we can find a better compromise between the size of the time window and a good sample of the network behavior.

# 8

## Conclusions

The vehicle becomes more connected and capable to cooperate with its environment. With such a connection capabilities, an ecosystem of connected services can be developed around the vehicles, leading to a complex communication environment in which different actors can have their communication requirements. Application can have specific communication requirements in terms of DR, latency and PDR. Users can present their preferences, e. g., the amount of money user is willing to pay for data communication. Administrators and regulators present their rules like interdiction to use certain frequencies in determined zones.

With these wide variety of communication requirements, only one network technology is not capable to satisfy the specific requirements from actors. Instead, we consider that is necessary to take advantage of existing network technologies like urban WiFi, vehicular WiFi, and Cellular. In this way, while vehicles move they can experience different network environments composed by heterogeneous access technologies.

In such a heterogeneous environment, in which a wide variety of applications compete for a set of heterogeneous access networks, it is necessary a DM mechanism capable to manage these communication. In the literature, works have addressed such a problem in different ways. Some works have proposed architectures and algorithms for network selection, usually considering few number of actors (e. g., only application requirements). While others have proposed only DM algorithms for handover process. Most works have concentrated on the application and network

---

information. Current DM usually consider static application requirements while look for the best access network that satisfy such requirements. And then, once the solution is found, the DM enforces them in the system.

Despite being a well-studied area, the decision making process does not have a complete solution that meet the C-ITS needs, e. g., they do not address the high dynamism of the vehicular environment, or do not consider heterogeneous requirements from different actors.

Based on the literature, the TOPSIS has been the most used MADM method for DM algorithm. Despite such algorithm to present relative low computation complexity, this approach has some issues like high sensitivity to extreme values, which leads to unstable decisions; high sensitivity to attributes' weight; and the necessity to make full recalculation even if only a given network parameter change. In dynamic environment, MADM algorithms are prone to switch flow communication from one access network to another every time a network parameter changes. In some cases, the MADM algorithm can switch forward and switch back a given flow, leading to the “ping-pong” effect. This instability affects the flow communication by decreasing the QoS, e. g., decreasing the PDR and increasing the communication latency.

Moreover, current DM algorithms find new solutions by reacting to the network changes. However, due to the high velocity of vehicles, their connectivity context can change frequently. In such highly dynamic mobility scenarios, it is needed a DM capable to anticipate decisions based on forecasted information.

Based on these findings, we worked on the AD4ON, a DM mechanism for opportunistic networking in heterogeneous network environment. The AD4ON mechanism is composed by an architecture and a DM algorithm.

The AD4ON architecture was developed in an ISO architecture compliant way. And it meets the identified properties for DM in the C-ITS communication. The AD4ON is a modular architecture that can be distributed into several physical units. Therefore, their modules can be implemented in different modules in-vehicle network.

Moreover, the AD4ON is capable to manage requirements and preferences from different actors (e.g., applications, users, administrators and regulators), as well as take into account the short-term prevision about the network environment. Thanks to different monitoring modules, the decision making process can have updated information about network environments. The AD4ON architecture is capable to manage multiple access technologies simultaneously, finding for each flow the access network that better meets communication requirements.

We performed simulations to compare the AD4ON algorithm with three other DM algorithms: the well-known *TOPSIS*, a modified version of TOPSIS (*mTOPSIS*) and a *commercial DM (CM)* used in most of smartphones, in which decisions are based on predefined network priorities. Simulation results demonstrate that AD4ON algorithm is capable to select access networks



that better match with flow's requirements, while managing multiple attributes and objectives simultaneously. Moreover, it outperforms the other algorithms by increasing the total flow satisfaction, increasing the decision stability and consequently reducing the "ping-pong" effect.

Unlike TOPSIS-based algorithms, the AD4ON makes decisions independently of the flows order. It spreads ants among the different flows and start the decision process in parallel, exploiting different possibilities of solutions. Such AD4ON approach allows to choose access networks that better satisfy each actor. Thanks to the decisions based on the pheromone concentration, the AD4ON presents more stable decisions, i. e., the migration from one access network to another is not abrupt but it occurs smoothly over time.

However, this property does not allow the flows immediately take advantage of newcomer networks. When a new access network becomes available, the AD4ON algorithm maintains the flow communication through the current access network for a few period of time until the ants converge to the new access network. This reaction time can be reduced if the AD4ON is aware about the near future of the network conditions.

For a better communication management, the AD4ON should increase its knowledge about the upcoming network environment, i. e., it needs to take into account the short-term prevision about the network environment in which the vehicle will be soon inserted. With these information, the AD4ON is capable to anticipate decisions and/or to prepare for the newcomers access networks.

Therefore, we developed a version of the AD4ON algorithm that takes into account such near future information about network environments (AD4ON-NF). And we simulated the algorithm with a simple scenario. Simulations demonstrate that taking into account the short-term prevision, the AD4ON-NF can optimize the algorithm reaction time.

## 8.1. Future work

In the present work, we considered the same weight for all parameters to be optimized (DR, PDR, Latency, RSSI, and monetary cost), i. e., we considered that all parameters have the same importance level in the decision process. However, this assumption is not always true. In some case, one parameter can be more relevant than other. For example, a user can prioritize to reduce the monetary cost to the detriment of DR. In such a case, the monetary cost should have a higher weight than the DR, in the objective function. Therefore, in future works we consider important to simulate parameters with different proportion of weights.

Moreover, communication actors (e. g., flows, users or administrators) can have variable requirements. For example, instead to define a fixed level of importance for the monetary cost, an user can define this importance based on the available budget. He/she can be more tolerant

to costly networks if the budget is high (then its weight is low), and as the budget decreases the monetary cost becomes more important (i. e., high weight value). In this context, we aim to simulate with variable weight values.

As mentioned before, the simulations we performed considering the near future does not consider prediction about flows requirements. Therefore, as future works we aim to simulate the AD4ON algorithm considering both near future about flows requirements and network environments.

Since we aim the AD4ON mechanism performing stable decisions, it is important to consider also the stability of future networks. Therefore, besides the average of predicted network parameters (e. g., DR and PDR) it is important to consider also, how stable a given network is expected to be. We aim to consider such a network stability prediction in the next simulations.

Finally, we consider important to implement the AD4ON in a real testbed, i. e., implement the AD4ON, including the short-term prediction modules, in an embedded MR and perform real tests.

## Appendices

## Appendix 1 – Key performance indicator

This annex shows the key performance results for each flow from scenario B.

### Streaming

Figure 1 shows the key performance of the four algorithm, i.e., AD4ON, TOPSIS, mTOPSIS and CM for the streaming flow.

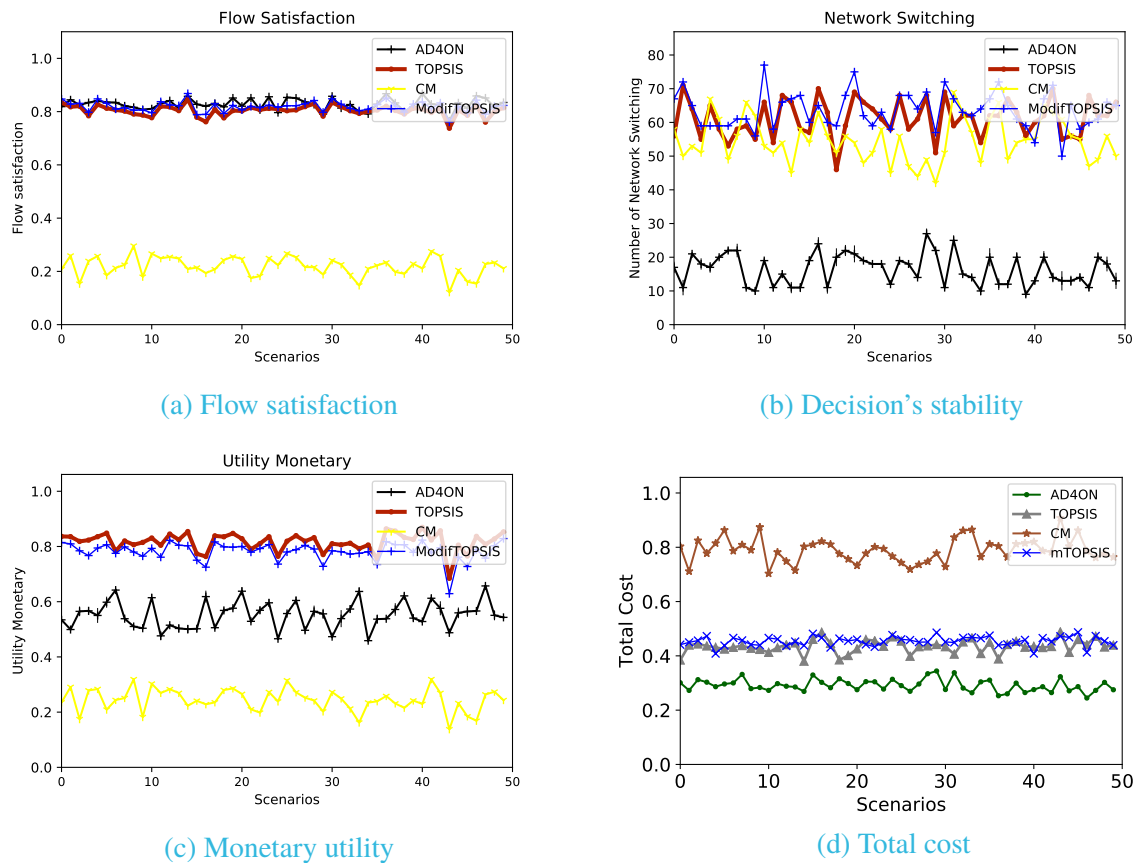
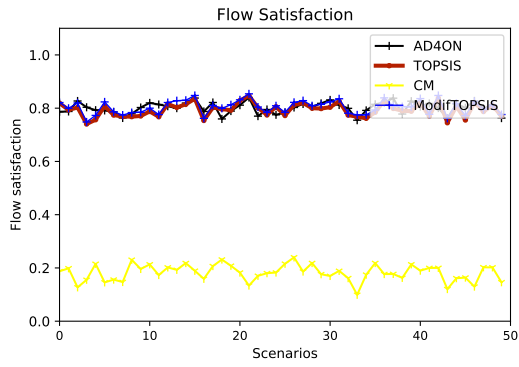


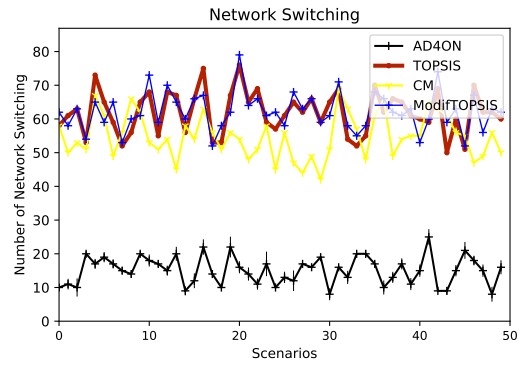
Figure 1: Key performance for streaming flow.

### Conversational

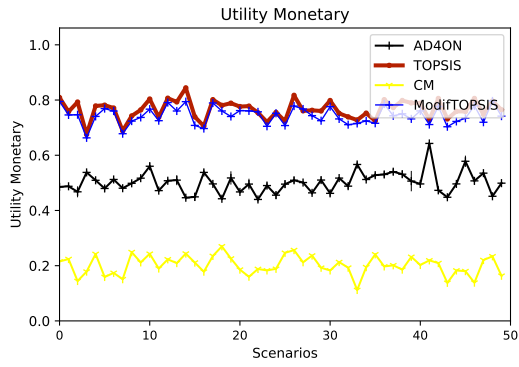
Figure 2 shows the key performance of the four algorithm for the conversational flow.



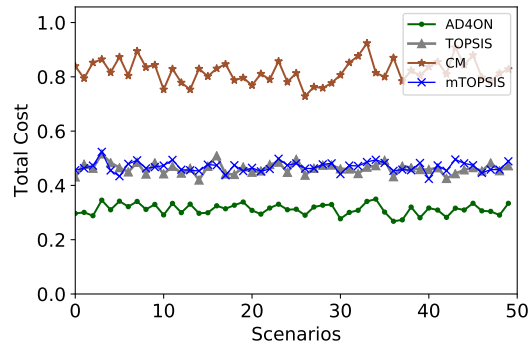
(a) Flow satisfaction



(b) Decision's stability



(c) Monetary utility

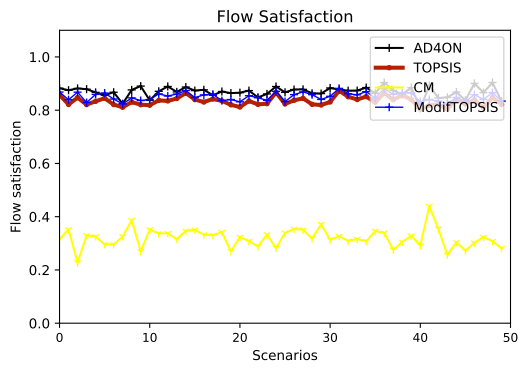


(d) Total cost

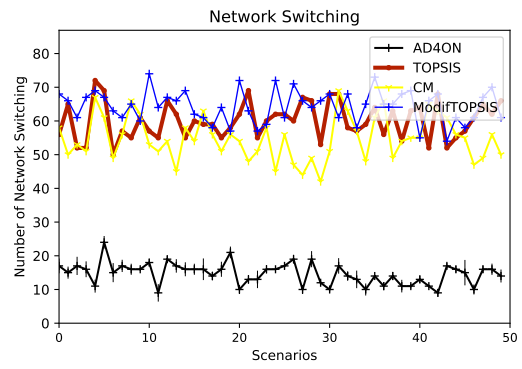
Figure 2: Key performance for conversational flow.

## Background

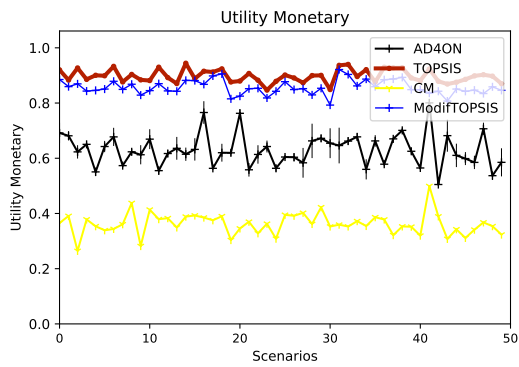
Figure 3 shows the key performance of the four algorithm for the background flow.



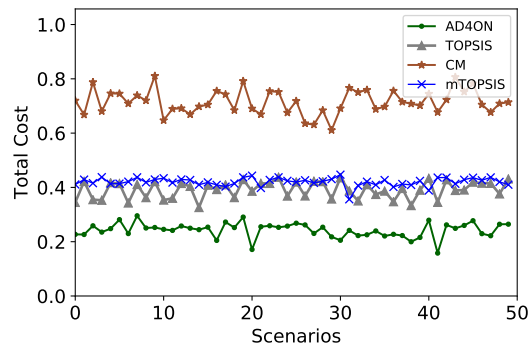
(a) Flow satisfaction



(b) Decision's stability



(c) Monetary utility

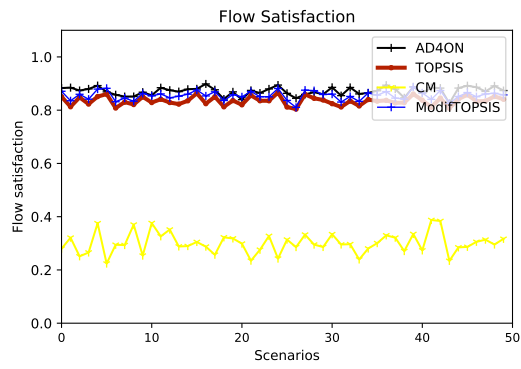


(d) Total cost

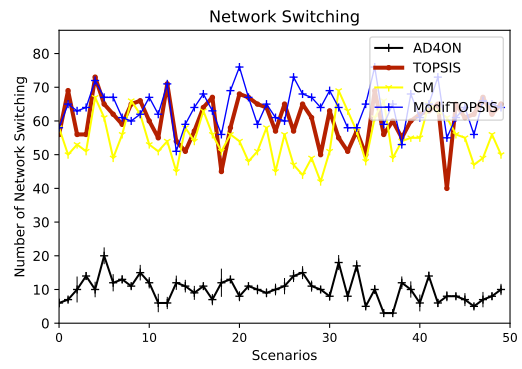
Figure 3: Key performance for conversational flow.

## Interactive

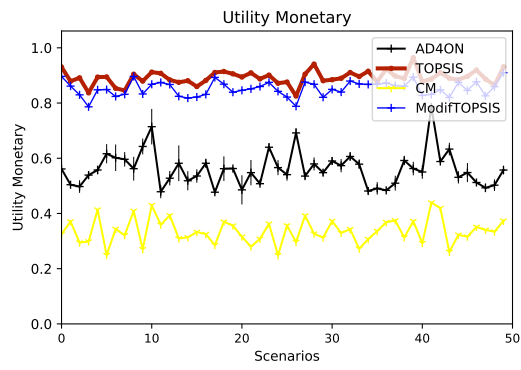
Figure 4 shows the key performance of the four algorithm for the interactive flow.



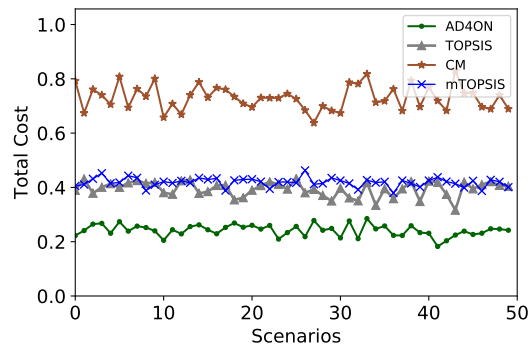
(a) Flow satisfaction



(b) Decision's stability



(c) Monetary utility



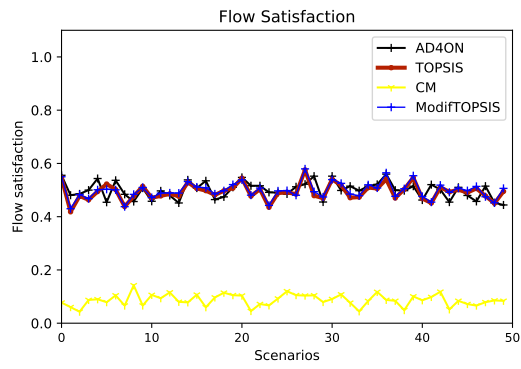
(d) Total cost

Figure 4: Key performance for interactive flow.

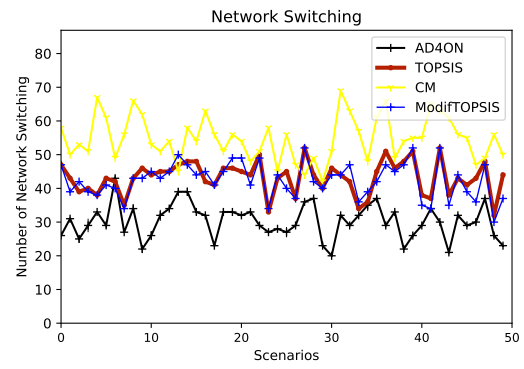
## Safety

Figure 5 shows the key performance of the four algorithm for the safety flow.

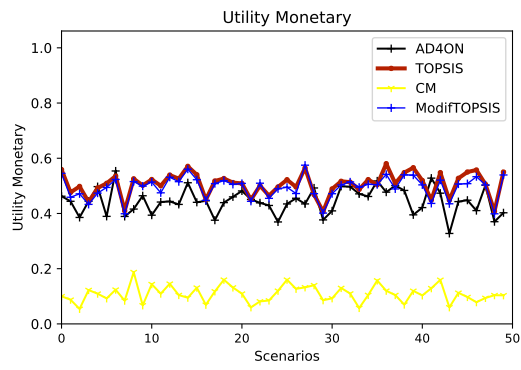
## 1. KEY PERFORMANCE INDICATOR



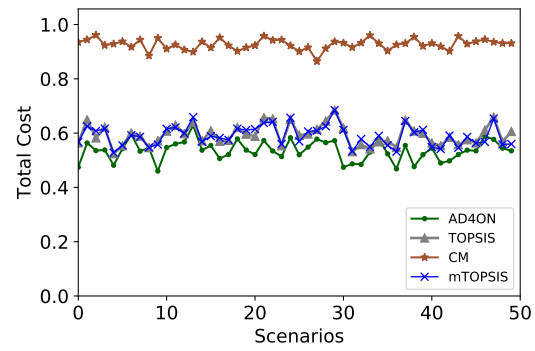
(a) Flow satisfaction



(b) Decision's stability



(c) Monetary utility



(d) Total cost

Figure 5: Key performance for safety flow.



## Appendix 2 – Number of networks for Scenario C

Subscenario	WiFi	ITS-G5	Cellular	TOTAL	Subscenario	WiFi	ITS-G5	Cellular	TOTAL
1	4	2	1	7	26	1	2	1	4
2	4	1	1	6	27	7	5	1	13
3	5	1	1	7	28	6	7	1	14
4	3	1	1	5	29	6	2	1	9
5	4	1	1	6	30	7	6	1	14
6	1	1	1	3	31	3	6	1	10
7	9	9	1	19	32	6	7	1	14
8	6	6	1	13	33	6	3	1	10
9	9	3	1	13	34	8	6	1	15
10	4	5	1	10	35	1	4	1	6
11	3	6	1	10	36	8	1	1	10
12	7	6	1	14	37	5	7	1	13
13	1	3	1	5	38	1	5	1	7
14	9	3	1	13	39	2	2	1	5
15	6	8	1	15	40	8	2	1	11
16	7	7	1	15	41	7	3	1	11
17	1	7	1	9	42	2	5	1	8
18	7	1	1	9	43	2	2	1	5
19	3	3	1	7	44	1	1	1	3
20	7	4	1	12	45	6	4	1	11
21	5	5	1	11	46	9	2	1	12
22	4	5	1	10	47	3	5	1	9
23	6	3	1	10	48	3	6	1	10
24	5	3	1	9	49	6	2	1	9
25	4	5	1	10	50	5	1	1	7

# List of Acronyms

ABC	Always Best Connected
ABS	Anti-Lock Braking System
ACO	Ant Colony Optimization
AD4ON	Ant-based Decision Maker for Opportunistic Networking
AD4ON-NF	AD4ON with Near Future
ADAS	Advanced Driver Assistance Systems
ADU	Application Data Unit
AEB	Autonomous Emergency Braking
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ALC	Adaptive Light Control
AR	Access Router
ARQ	Automatic Repeat reQuest
C-ITS	Cooperative Intelligent Transportation Systems
C2C-CC	Car-2-Car Communication Consortium
CALM	Communications Access for Land Mobiles
CAM	Cooperative Awareness Message
CCH	Control Channel

CEN	European Committee for Standardization
CM	Commercial DM
CN	Correspondent Node
CoA	Care-of-address
CPH	Communication Profile Handler
CPU	Central Processing Unit
CSH	Content Subscription Handler
DASH	Dynamic Adaptive Streaming over HTTP
DENM	Decentralized Environmental Notification Message
DM	Decision Maker
DR	Data Rate
DSRC	Dedicated Short Range Communications
ECU	Electronic Control Unit
EFC	Electronic Fee Collection
ESC	Electronic Stability Control
ETSI	European Telecommunications Standards Institute
FCA	Forward Collision Avoidance
FSH	Facilities Services Handler
GPS	Global Positioning System
GUI	Graphical User Interface
HA	Home Agent
HMI	Human-Machine Interface
HTTP	Hypertext Transfer Protocol

IC	Index of Convergence
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
ITS	Intelligent Transportation Systems
ITS-FPDU	ITS-S Facility layer Protocol Data Unit
ITS-S	ITS Station
KPI	Key Performance Indicator
LAN	Local Area Network
LDM	Local Dynamic Map
MA-SAP	MA-Service Access Point
MADM	Multi-Attribute Decision Making
MCoA	Multiple Care of Addresses Registration
MEW	Multiplicative Exponential Weighting
MIH	Media Independent Handover
MIHF	Media Independent Handover Function
MIP	Mobile IP
MN	Mobile Node
MN-SAP	MN-Service Access Point
MNN	Mobile Network Nodes
MNO	Mobile Network Operator

MNP	Mobile Network Prefix
MOO	Multi-Objective Optimization
MPTCP	Multipath TCP Protocol
MR	Mobile Router
mTOPSIS	modified TOPSIS
NEMO	Network Mobility Basic Support Protocol
NS-3	Network Simulator 3
OBU	Onboard Unit
OEM	Original Equipment Manufacturers
OSI	Open Systems Interconnection
PDR	Packet Delivery Ratio
QoE	Quality of Experience
QoS	Quality of Service
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
RSU	Roadside Unit
RTT	Round-Trip Time
SAP	Service Access Point
SAW	Simple Additive Weighting
SCH	Service Channel
SCTP	Stream Control Transmission Protocol
SI	Swarm Intelligence
TCP	Transmission Control Protocol

TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TSP	Traveling Salesman Problem
UDP	User Datagram Protocol
UMU	University of Murcia
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicle Ad hoc Network
VM	Virtual Machine
WAVE	Wireless Access in the Vehicular Environment
WPM	Weighted Product Method
WSMP	WAVE Short Message Protocol

# List of Publications

## Poster: Summer School (Germany)

- BMW – TUM – Eurecom Summer School. On the “Car as a service – creating tomorrow’s smart mobility service platform”, from July 18-23rd, 2016.

## Technical Reports

- R. Silva and J. M. Bonnin, “SIMHet - Smart Interface Management in Heterogeneous Networks - State of the Art”, Technical report D1.0, April. 2016
- R. Silva and J. M. Bonnin, “SIMHet - Smart Interface Management in Heterogeneous Networks - DM first solution”, Technical report D2.0, August. 2018

## Book chapter

- R. Silva, J. M. Bonnin and T. Ernst. “Opportunistic Networking for ITS”. In Intelligent Transport Systems (ITS): Past, Present and Future Directions. Nova Science, 2017.
- R. Silva, C. Couturier, T. Ernst and J. M. Bonnin. “Proactive Decision Making for ITS Communication”. In Global Advancements in Connected and Intelligent Mobility. IGI Global, 2019.

## International Conference

- R. Silva, J. M. Bonnin, and T. Ernst, “An ITS-based Architecture for Opportunistic Networking,” in VEHICULAR 2017: The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications, 2017, pp. 91–96.

- R. Silva, C. Couturier, J. M. Bonnin, and T. Ernst, “A Heuristic Decision Maker Algorithm for Opportunistic Networking in C-ITS” in VEHITS 2019: The Fifth International Conference on Vehicle Technology and Intelligent Transport Systems, 2019.

## **Journal**

- R. Silva, J. M. Bonnin, and T. Ernst, “AD4ON: An ITS-based Decision Making Architecture for Opportunistic Networking,” International Journal on Advances in Networks and Services (IJANS), vol. 11, pp. 11–21, 2018.



# Bibliography

- [1] W. G. Najm, J. Koopmann, J. D. Smith, and J. Brewer, “Frequency of Target Crashes for IntelliDrive Safety Systems,” U.S. Department of Transportation - National Highway Traffic Safety Administration, Tech. Rep., Oct. 2010.
- [2] “ISO 21217 Intelligent Transport Systems — Communications Access for Land Mobiles (CALM) — Architecture,” Tech. Rep., 2014.
- [3] J. Sousanis, “World Vehicle Population Tops 1 Billion Units,” <http://wardsauto.com/news-analysis/world-vehicle-population-tops-1-billion-units>, 2016.
- [4] United Nations, Department of Economic and Social Affairs, and Population Division, *World Urbanization Prospects: The 2014 Revision : Highlights*, 2014.
- [5] “Accidentalite Routiere en France et en Europe,” European Commission - Observatoire National Interministériel de la Securite Routiere, Tech. Rep., May 2015.
- [6] M. de l’Intérieur, “Securite Routiere : 26 mesures pour une nouvelle mobilisation,” <http://www.interieur.gouv.fr/Actualites/Communiqués/Securite-routiere-26-mesures-pour-une-nouvelle-mobilisation>, 2015.
- [7] “Bilan definitif de l’accidentalite routiere 2014,” <http://www.securite-routiere.gouv.fr/medias/espace-presse/publications-presse/bilan-definitif-de-l-accidentalite-routiere-2014>, 2014.
- [8] M. Lu, K. Wevers, and R. Van Der Heijden, “Technical Feasibility of Advanced Driver Assistance Systems (ADAS) for Road Traffic Safety,” *Transportation Planning and Technology*, vol. 28, no. 3, pp. 167–187, Jun. 2005.
- [9] R. van der Meulen and J. Rivera, “Gartner Says By 2020, a Quarter Billion Connected Vehicles Will Enable New In-Vehicle Services and Automated Driving Capabilities,” <http://www.gartner.com/newsroom/id/2970017>, [Accessed: September 2018].

- [10] J. W. Ross, “Creating a Strategic IT Architecture Competency: Learning in Stages,” Social Science Research Network, Rochester, NY, SSRN Scholarly Paper ID 416180, Apr. 2003.
- [11] “Standard for Information technology–Telecommunications and information exchange between systems local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11*, pp. 1–5229, 2012.
- [12] “IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture,” *IEEE Std 1609.0*, pp. 1–78, 2013.
- [13] “CAR 2 CAR Communication Consortium Manifesto,” Tech. Rep., Aug. 2007.
- [14] “Information technology – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments,” *IEEE Std 802.11p*, pp. 1–51, 2010.
- [15] “ISO/DIS 24102-6 Intelligent Transport Systems — Communications Access for Land Mobiles (CALM) — ITS station management — Part 6: Path and flow management,” Tech. Rep., 2015.
- [16] “ISO/TS 17423 Intelligent Transport Systems — Cooperative systems — Application requirements for selection of communication profiles,” Tech. Rep., 2014.
- [17] “ISO/DTS 17429 Intelligent Transport Systems — Cooperative systems — ITS station facilities for the transfer of information between ITS stations,” Tech. Rep., 2015.
- [18] “ISO/TS 17419 Intelligent transport systems — Cooperative systems — Classification and management of ITS applications in a global context,” Tech. Rep., 2014.
- [19] “ISO 21210 Intelligent transport systems — Communications access for land mobiles (CALM) — IPv6 Networking,” Tech. Rep., 2012.
- [20] H. Soliman, G. Tsirtsis, G. Giaretta, K. Kuladinithi, and N. Montavont, “Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support,” RFC 6089, 2011.
- [21] E. Gustafsson and A. Jonsson, “Always best connected,” *IEEE Wireless Communications*, vol. 10, no. 1, pp. 49–55, Feb. 2003.
- [22] K. Piamrat, A. Ksentini, J. M. Bonnin, and C. Viho, “Radio resource management in emerging heterogeneous wireless networks,” *Computer Communications*, vol. 34, no. 9, pp. 1066–1076, Jun. 2011.

- [23] E. Perraud and K. Eckert, “5G Croco - Test Case Definition and Test Site Description Part 1,” Tech. Rep. D2.1, Jun. 2019.
- [24] M. Khan, C. Jung, P. Uzoh, C. Zhenbo, J. Kim, Y. Yoon, A. Nadeem, and K. Han, “Enabling vertical handover management based on decision making in heterogeneous wireless networks,” in *Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug. 2015, pp. 952–957.
- [25] N. Omheni, F. Zarai, M. Obaidat, and L. Kamoun, “Optimized MIH-assisted P-NEMO design for vertical handover over heterogeneous network mobility,” in *IEEE International Conference on Communications (ICC)*, Jun. 2014, pp. 2879–2884.
- [26] N. Omheni, F. Zarai, M. Obaidat, and K.-F. Hsiao, “A novel vertical handoff decision making algorithm across Heterogeneous Wireless Networks,” in *International Conference on Computer, Information and Telecommunication Systems (CITS)*, Jul. 2014, pp. 1–6.
- [27] M. Fotuhi, V. Zolfaghari, M. Shokrnezhad, and S. Khorsandi, “Adaptive Joint Radio Resource Management algorithm for heterogeneous wireless networks,” in *2014 6th Conference on Information and Knowledge Technology (IKT)*, May 2014, pp. 44–49.
- [28] J. Santa, F. Pereniguez-Garcia, F. Bernal, P. Fernandez, R. Marin-Lopez, and A. Skarmeta, “A Framework for Supporting Network Continuity in Vehicular IPv6 Communications,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 1, pp. 17–34.
- [29] D. Tosh and S. Sengupta, “Heterogeneous access network(s) selection in multi-interface radio devices,” in *IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Mar. 2015, pp. 117–122.
- [30] C. Roman, P. Ball, and S. Ou, “Performance Evaluation of Dynamic Scheduler in Multiple Queue Shim Layer for Heterogeneous Vehicular Communications,” in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.
- [31] Y. Wang, P. Zhang, Y. Zhou, J. Yuan, F. Liu, and G. Li, “Handover Management in Enhanced MIH Framework for Heterogeneous Wireless Networks Environment,” *Wireless Personal Communications*, vol. 52, no. 3, pp. 615–636, Feb. 2010.
- [32] O. Khattab and O. Alani, “Algorithm for seamless vertical handover in heterogeneous mobile networks,” in *Science and Information Conference (SAI)*, Aug. 2014, pp. 652–659.
- [33] N. Varga, L. Bokor, and H.-J. Fischer, “LDM-based dynamic network discovery and selection for IPv6 mobility management optimization in C-ITS environments,” in *International*

*Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, Jun. 2015, pp. 483–490.

- [34] R. B. Rayana and J. M. Bonnin, “A Smart Management Framework for Multihomed Mobile Nodes and Mobile Routers,” in *IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2009, pp. 2881–2885.
- [35] M. Thaalbi and N. Tabbane, “Improving Vertical Handover over Heterogeneous Technologies Using a Cross Layer Framework,” in *IEEE Intl Conf on High Performance Computing and Communications, IEEE 6th Intl Symp on Cyberspace Safety and Security, IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICSS)*, Aug. 2014, pp. 1170–1176.
- [36] L. Suciu, J.-M. Bonnin, K. Guillouard, and B. Stévant, “Achieving “Always Best Connected” Through Extensive Profile Management,” in *Personal Wireless Communications*, ser. Lecture Notes in Computer Science, I. Niemegeers and S. H. de Groot, Eds. Springer Berlin Heidelberg, 2004, pp. 421–430.
- [37] S. Atheo, J. M. Bonnin, K. Guillouard, B. Stevant, and L. GHEORGHE SUCIU, “Procédé de génération automatique de profils actifs pour terminal de télécommunications,” France Patent.
- [38] D. Jiang, L. Huo, Z. Lv, H. Song, and W. Qin, “A Joint Multi-Criteria Utility-Based Network Selection Approach for Vehicle-to-Infrastructure Networking,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2018.
- [39] C. Roman, P. Ball, and S. Ou, “A shim layer for heterogeneous wireless communications scheduling in intelligent transport systems,” in *2015 IEEE Symposium on Computers and Communication (ISCC)*, Jul. 2015, pp. 174–179.
- [40] H. Tahir, A. Al-Ghushami, and Z. Yahya, “Selection of access network using cost function method in heterogeneous wireless network,” in *International Conference on Multimedia Computing and Systems (ICMCS)*, Apr. 2014, pp. 789–793.
- [41] K. Piamrat, A. Ksentini, J. Bonnin, and C. Viho, “Q-DRAM: QoE-Based Dynamic Rate Adaptation Mechanism for Multicast in Wireless Networks,” in *IEEE Global Telecommunications Conference.*, Nov. 2009, pp. 1–6.
- [42] Y. Xu, L. Li, B.-H. Soong, and C. Li, “Fuzzy Q-learning based vertical handoff control for vehicular heterogeneous wireless network,” in *IEEE International Conference on Communications (ICC)*, Jun. 2014, pp. 5653–5658.

- [43] F. Mirani, N. Boukhatem, and P.-N. Tran, “On terminal utility for multiple flow/interface association in mobile terminals,” in *Wireless Days (WD), IFIP*, Oct. 2011, pp. 1–6.
- [44] T. Sutinen and H. Rivas, “Cross-Layer Assisted Network Interface Selection for Multi-Interface Video Streaming,” in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, Jul. 2011, pp. 1–6.
- [45] B. D. Higgins, “Intentional Networking: Opportunistic Exploitation of Mobile Network Diversity,” *MobiCom*, 2010.
- [46] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, “Improved access point selection,” in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services*. ACM, 2006, pp. 233–245.
- [47] S. Bi, C. Chen, R. Du, and X. Guan, “Proper Handover between VANET and Cellular Network Improves Internet Access,” in *IEEE 80th Vehicular Technology Conference (VTC Fall)*., Sep. 2014, pp. 1–5.
- [48] Lei Weimin and Yang Yang, “A path selection decision-making model at the application layer for Multipath Transmission,” in *2012 International Conference on Information Management, Innovation Management and Industrial Engineering*. Sanya, China: IEEE, Oct. 2012, pp. 281–284.
- [49] R. Ben Rayana, “A Smart Management Framework for Multihomed Mobile Nodes & Mobile Routers,” Doctoral Dissertation, Jan. 2010.
- [50] C. W. Chang, W. T. Lee, and H. W. Wei, “FHR-SCTP: The Research of a Quality Aware SCTP Congestion Control Mechanism to Improve Handover Efficiency,” in *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Kitakyushu, Japan: IEEE, Aug. 2014, pp. 654–657.
- [51] I. Joe and E. Huang, “SCTP throughput improvement through automatic path switching,” in *2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012)*, vol. 2, Jun. 2012, pp. 243–246.
- [52] Jonghwan Chung, D. Han, J. Kim, and Chong-kwon Kim, “Machine learning based path management for mobile devices over MPTCP,” in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*. Jeju Island, South Korea: IEEE, Feb. 2017, pp. 206–209.

- [53] J. Kim, B.-H. Oh, and J. Lee, "Receive Buffer based Path Management for MPTCP in heterogeneous networks," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. Lisbon, Portugal: IEEE, May 2017, pp. 648–651.
- [54] M. Kalyanaraman, S. Seetharaman, and S. Srikanth, "Integrated approach for proxy-mobile-IPv6 (PMIPv6) based IP Flow Mobility and offloading," in *Twenty First National Conference on Communications (NCC)*, Feb. 2015, pp. 1–6.
- [55] C. Cho, J.-d. Cho, and J. Jeong, "Cost-Effective Proxy-LMA Mobility Management Scheme in Mobile Networks with Global Mobility Support," in *Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, Jul. 2014, pp. 121–127.
- [56] P. Ward, K. Naik, and J. Schmidtke, "Efficient Hashing for Dynamic Per-Flow Network-Interface Selection," in *IEEE 28th International Conference on Advanced Information Networking and Applications (AINA)*, May 2014, pp. 441–448.
- [57] G. P. Koudouridis, H. Lundqvist, H. R. Karimi, and G. Karlsson, "A quantitative analysis of the throughput gains and the energy efficiency of multi-radio transmission diversity in dense access networks," *Telecommun Syst*, vol. 59, no. 1, pp. 145–168, May 2015.
- [58] A. Yaver and G. P. Koudouridis, "Performance Evaluation of Multi-Radio Transmission Diversity: QoS Support for Delay Sensitive Services," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. Barcelona, Spain: IEEE, Apr. 2009, pp. 1–5.
- [59] G. P. Koudouridis, A. Yaver, and M. U. Khattak, "Performance Evaluation of Multi-Radio Transmission Diversity for TCP Flows," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. Barcelona, Spain: IEEE, Apr. 2009, pp. 1–5.
- [60] I. Bisio, C. Braccini, S. Delucchi, F. Lavagetto, and M. Marchese, "Dynamic multi-attribute Network Selection algorithm for Vertical Handover procedures over mobile ad hoc networks," in *2014 IEEE International Conference on Communications (ICC)*, Jun. 2014, pp. 342–347.
- [61] H. Ben Elhadj, J. Elias, L. Chaari, and L. Kamoun, "Multi-Attribute Decision Making Handover Algorithm for Wireless Body Area Networks," *Computer Communications*, vol. 81, pp. 97–108, May 2016.
- [62] M. Hu, W. Yang, K. Liu, and S. Cong, "A CRI Based Vertical Handoff Algorithm in Heterogeneous Networks," in *2014 IEEE International Conference on Computer and Information Technology (CIT)*, Sep. 2014, pp. 258–262.

- [63] N. U. Hasan, W. Ejaz, N. Ejaz, H. S. Kim, A. Anpalagan, and M. Jo, "Network Selection and Channel Allocation for Spectrum Sharing in 5G Heterogeneous Networks," *IEEE Access*, vol. 4, pp. 980–992, 2016.
- [64] A. Habbal, S. I. Goudar, and S. Hassan, "Context-Aware Radio Access Technology Selection in 5G Ultra Dense Networks," *IEEE Access*, vol. 5, pp. 6636–6648, 2017.
- [65] M. H. Cheung, F. Hou, J. Huang, and R. Southwell, "Congestion-Aware DNS for Integrated Cellular and Wi-Fi Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1269–1281, Jun. 2017.
- [66] M. A. Senouci, S. Hoceini, and A. Mellouk, "Utility function-based TOPSIS for network interface selection in Heterogeneous Wireless Networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.
- [67] F. Bouali, K. Moessner, and M. Fitch, "A Context-Aware User-Driven Framework for Network Selection in 5G Multi-RAT Environments," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sep. 2016, pp. 1–7.
- [68] N. Zarin and A. Agarwal, "A hybrid network selection scheme for heterogeneous wireless access network," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017, pp. 1–6.
- [69] M. A. Senouci, H. Senouci, S. Hoceini, and A. Mellouk, "Flow/Interface Association for Multihomed Mobile Terminals in Heterogeneous Wireless Networks," in *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, Oct. 2018, pp. 171–176.
- [70] L. Wang and G.-S. Kuo, "Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks #x2014; A Tutorial," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 271–292, First 2013.
- [71] V. Fux, P. Maillé, J. M. Bonnin, and N. KACI, "Efficiency or fairness: Managing applications with different delay sensitivities in heterogeneous wireless networks," in *WoWMoM : IEEE14th International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Madrid, Spain, Jun. 2013.
- [72] A. Ben Nacef and N. Montavont, "A generic end-host mechanism for path selection and flow distribution," in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications.*, Sep. 2008, pp. 1–5.

- [73] G. Castignani, “Exploiting Network Diversity,” Doctoral Dissertation, Télécom Bretagne, Université de Rennes 1, Nov. 2012.
- [74] M. Drissi and M. Oumsis, “Performance evaluation of multi-criteria vertical handover for heterogeneous wireless networks,” in *Intelligent Systems and Computer Vision (ISCV), 2015*, Mar. 2015, pp. 1–5.
- [75] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*. Springer Science & Business Media, 1981.
- [76] D. M. Pavlicic, “Normalisation affects the results of MADM methods,” *Yugoslav journal of operations research*, vol. 11, no. 2, pp. 251–265, 2001.
- [77] N. Vafaei, R. Almeida Ribeiro, and L. Camarinha-matos, “Importance of Data Normalization in Decision Making: Case study with TOPSIS method ON DECISION SUPPORT SYSTEMS TECHNOLOGIES AN EWG-DSS CONFERENCE. THEME: BIG DATA ANALYTICS FOR DECISION-MAKING,” in *INT. CONFERENCE ON DECISION SUPPORT SYSTEMS TECHNOLOGIES - ICDSST*, May 2015.
- [78] E. Stevens-Navarro and V. W. S. Wong, “Comparison between Vertical Handoff Decision Algorithms for Heterogeneous Wireless Networks,” in *2006 IEEE 63rd Vehicular Technology Conference*, vol. 2, May 2006, pp. 947–951.
- [79] E. Triantaphyllou, “More Cases of Ranking Abnormalities when some MCDM Methods are used,” in *Multi-Criteria Decision Making Methods: A Comparative Study*, ser. Applied Optimization. Springer US, 2000, no. 44, pp. 213–233.
- [80] M. Mavrovouniotis, C. Li, and S. Yang, “A survey of swarm intelligence for dynamic optimization: Algorithms and applications,” *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, Apr. 2017.
- [81] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, Mass: MIT Press, 2004.
- [82] R. Silva, H. S. Lopes, and W. Godoy, “A Heuristic Algorithm Based on Ant Colony Optimization for Multi-objective Routing in Vehicle Ad Hoc Networks,” in *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, Sep. 2013, pp. 435–440.
- [83] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, “Data mining with an ant colony optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, Aug. 2002.



- [84] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," in *2013 8th International Conference on Computer Engineering Systems (ICCES)*, Nov. 2013, pp. 64–69.
- [85] R. B. Rayana and J. M. Bonnin, "Intelligent Middle-Ware Architecture for Mobile Networks," in *MobileWireless Middleware, Operating Systems, and Applications*. Springer Berlin Heidelberg, Apr. 2009, no. 7, pp. 43–57.
- [86] "ISO 24102-3 Intelligent Transport Systems — Communications Access for Land Mobiles (CALM) — ITS station management — Part 3: Service access points," Tech. Rep., 2013.
- [87] "ISO/DIS 24102-1 Intelligent Transport Systems — Communications Access for Land Mobiles (CALM) — ITS station management — Part 1: Local management," Tech. Rep., 2013.
- [88] L. GHEORGHE SUCIU, "Gestion d'interfaces multiples et mécanismes de sélection automatique de l'interface réseaux," Ph.D. dissertation, ENST de Bretagne, 2005.
- [89] "ETSI TS 122 105 V14.0.0 Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Services and service capabilities," Tech. Rep., 2017.
- [90] "IEEE 802.21 – Standard for Local and metropolitan area networks – Part 21: Media Independent Handover Services," Tech. Rep., 2008.
- [91] "ETSI EN 302 895 Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)," Tech. Rep., 2014.
- [92] L. Suci, J. M. Bonnin, K. Guillouard, and B. Stevant, "Towards a highly adaptable user-centric terminal architecture," in *The 7th Intl. Symposium on Wireless Personal Multimedia Communications (WPMC)*, Italy, 2004.
- [93] J. M. Bonnin and R. Ben Rayana, "Heterogeneous networks and mobile management in intelligent transportation systems," in *Wireless Technologies in Intelligent Transportation Systems*. Nova Science, 2010.
- [94] S. Atheo, K. Guillouard, J. M. Bonnin, L. Suci, and B. Stevant, "Procédé de génération automatique de profils actifs pour terminal de télécommunications," Patent EP 1 569 489 A1.

- [95] R. Silva, S. Noguchi, T. Ernst, A. De La Fortelle, and W. Godoy Junior, “Standards for Cooperative Intelligent Transportation Systems: A Proof of Concept,” in *The Tenth Advanced International Conference on Telecommunications (AICT)*, Jul. 2014.
- [96] A. Jahan and K. L. Edwards, “A state-of-the-art survey on the influence of normalization techniques in ranking: Improving the materials selection process in engineering design,” *Materials & Design (1980-2015)*, vol. 65, pp. 335–342, Jan. 2015.
- [97] M. S. García-Cascales and M. T. Lamata, “On rank reversal and TOPSIS method,” *Mathematical and Computer Modelling*, vol. 56, no. 5-6, pp. 123–132, Sep. 2012.
- [98] Y. Chen, T. Farley, and N. Ye, “QoS Requirements of Network Applications on the Internet,” *Information Knowledge Systems Management*, vol. 4, no. 1, pp. 55–76, Jan. 2004.
- [99] “Connect Requirements Guide,” Skype, Tech. Rep., 2011.
- [100] “OpenVPN | The World’s Most Trusted Virtual Private Network,” <https://openvpn.net/>.
- [101] “Yogoko,” <https://www.yogoko.com/en/main.html>, [Accessed: August 2018].
- [102] P. J. Fernández, J. Santa, F. Pereñíguez, and A. F. Skarmeta, “Towards seamless inter-technology handovers in vehicular IPv6 communications,” *Computer Standards & Interfaces*, vol. 52, pp. 85–96, May 2017.
- [103] M.-P. Newswire, “Volvo Cars and Ericsson developing intelligent media streaming for self-driving cars,” <http://www.multivu.com/players/uk/7717751-volvo-ericsson-intelligent-media-streaming/>, 2016.
- [104] “A Glimpse into the Future: Continental demonstrates Predictive Connectivity Manager,” <https://www.continental-corporation.com/en/press/press-releases/2018-02-26-predictive-connectivity-122554>, [Accessed: December 2018].
- [105] R. Silva, J. M. Bonnin, and T. Ernst, “Opportunistic Networking for ITS,” in *Intelligent Transport Systems: Past, Present and Future Directions*. Nova publisher, 2017.
- [106] T. Couronne, V. Kirzner, K. Korenblat, E. Ravve, and Z. Volkovich, “Modelling Behavior Patterns in Cellular Networks,” in *ICCGI 2016 : The Eleventh International Multi-Conference on Computing in the Global Information Technology*. IARIA, 2016.
- [107] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, “Human Mobility Patterns in Cellular Networks,” *IEEE Communications Letters*, vol. 17, no. 10, pp. 1877–1880, Oct. 2013.

- [108] H. G. Othmer and L. E. Scriven, “Instability and dynamic pattern in cellular networks,” *Journal of Theoretical Biology*, vol. 32, no. 3, pp. 507–537, Sep. 1971.
- [109] R. Silva, C. Couturier, T. Ernst, and J.-M. Bonnin, “Proactive Decision Making for ITS Communication,” *Global Advancements in Connected and Intelligent Mobility: Emerging Research and Opportunities*, pp. 197–226, 2020.
- [110] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker, “Agile Application-Aware Adaptation for Mobility,” p. 12, 1997.
- [111] R. B. Rayana and J. M. Bonnin, “Mobility aware application manager for mobile networks,” in *8th International Conference on ITS Telecommunications.*, Oct. 2008, pp. 337–342.

---

**Titre :** Mécanisme de décision multicritères basé sur les ITS pour le placement de flux en environnement réseau hétérogène

**Mots clés :** ISO TC 204; ETSI TC ITS; architecture ITS; C-ITS; mécanisme de décision

**Résumé :** Le nombre de véhicules connectés augmentent et pour une utilisation plus intelligente des systèmes de transport, les véhicules ont besoin d'accroître leur connaissance de l'environnement. Cela peut être atteint en permettant aux véhicules de communiquer avec leur environnement.

Pour une connectivité omniprésente, il sera nécessaire d'utiliser diverses technologies sans fil, existantes. Dans un tel environnement de réseau d'accès hétérogène, il est nécessaire fournir aux applications des mécanismes décisionnels transparents pour gérer l'affectation des flux de données sur les réseaux disponibles.

Dans cette thèse, nous proposons le Ant-based Decision Maker for Opportunistic Networking (AD4ON), un Decision Maker (DM) mécanisme modulaire capable de choisir le meilleur profil de communication disponible pour chaque flux de données dans un environnement de réseau hétérogène et dynamique. Le mécanisme proposé gère les exigences et préférences de différents acteurs (applications, utilisateurs, etc.), ainsi que les informations réseaux dans un futur proche, afin de prendre des décisions plus intelligentes. C'est-à-dire, afin d'augmenter la satisfaction de chaque acteur impliqué dans le processus de communication, ainsi que d'augmenter la stabilité de la décision (par exemple, en réduisant l'effet "ping-pong").

---

**Title :** ITS-based Decision Making Mechanism for Opportunistic Networking in Heterogeneous Network Environment

**Keywords :** ISO TC 204; ETSI TC ITS; ITS-S communication architecture; C-ITS; decision making

**Abstract :** Demand from different actors for extended connectivity where vehicles can exchange data with other devices have pushed vehicle manufacturers to invest in embedded solutions, which paves the way towards Cooperative Intelligent Transportation Systems (C-ITS). Cooperative vehicles enable the development of an ecosystem of services around them. Due to the heterogeneity of such services and their specific requirements, for ubiquitous connectivity it is necessary to combine existing wireless technologies, providing applications with a communication architecture that hides such underlying access technologies specificities. Moreover, due to vehicles' high velocity it is needed a Decision Maker (DM) mechanism capable to take into account the short-term prevision about network environment in order to better manage all flow communications.

Based on the Intelligent Transportation Systems (ITS) architecture proposed by International Organization for Standardization (ISO), we proposed the Ant-based Decision Maker for Opportunistic Networking (AD4ON), a modular decision maker mechanism capable to choose the best available access network for each data flow in an heterogeneous and dynamic network environment. The proposed mechanism manages requirements and preferences from different actors, taking into account the short-term prevision about the network environment.

Simulations have demonstrated that the AD4ON outperforms current benchmark algorithms, by increasing decision's stability, reducing the "ping-pong" effect and maximizing flow's satisfaction. Moreover, we demonstrate that taking into account the short-term prevision, the AD4ON can optimize the algorithm reaction time.