

Rapport de stage :

Optimisation du placement de calcul pour l'application des usagers vulnérables de la route (VRU) sous les canaux Rician Fading, dans un contexte de véhicules connectés.

Élaboré par : Rhayoute Abdelmalek
Encadré par : Zoghلامي Chaima

2ème année Département Sciences du Numérique - Option Télécommunications
2021-2022



Institut de Recherche
en Informatique de Toulouse

Contents

1	Introduction :	
2	Mots clés :	
3	Modèle et formulation du problème :	
3.1	Modèle de calcul :	
3.1.1	Calcul local :	
3.1.2	Calcul dans le serveur MEC :	
3.1.3	Calcul dans le serveur EDGE cloud :	
3.2	Conclusion :	
3.2.1	Latence :	
3.2.2	Énergie :	
4	Le problème :	
5	Algorithme de résolution :	
6	Simulation :	
7	Conclusion :	
8	Références :	

List of Figures

1	A propos de C-ITS	1
2	Divers scénarios V2X dans les systèmes IoT [3]	2
3	Une architecture générale de système pour un système V2VRU basé sur le MBB* 4G.	2
4	Architecture du MEC	2

1 Introduction :

Vulnerable road users (VRUs), tels que les piétons, les cyclistes et les motocyclistes, sont les plus exposés aux risques liés à la circulation routière. À l'échelle mondiale, plus de la moitié des décès sur la route concernent des usagers vulnérables. Bien que des efforts considérables soient déployés pour améliorer la sécurité des VRUs, depuis les solutions techniques jusqu'à l'application de la loi, le nombre de décès de VRU continue d'augmenter [1]. La technologie émergente, **Cooperative Intelligent Transportation System (C-ITS)**, a le potentiel avéré d'améliorer la sécurité routière en permettant la communication sans fil pour échanger des informations entre les usagers de la route. Ces informations échangées sont utilisées pour créer une conscience de la situation et détecter à l'avance toute collision potentielle afin de prendre les mesures nécessaires pour éviter toute victime de la route. Les solutions actuelles de C-ITS pour la sécurité des VRU se limitent toutefois à une communication unidirectionnelle, les VRU n'ayant pour mission que d'avertir les conducteurs de leur présence dans le but d'éviter les collisions. Cette interaction unidirectionnelle limite considérablement l'énorme potentiel de la C-ITS qui pourrait être utilisé pour concevoir une solution plus efficace pour la sécurité des VRU, où les VRU peuvent être équipés d'une communication bidirectionnelle avec toutes les fonctionnalités de la C-ITS[4].

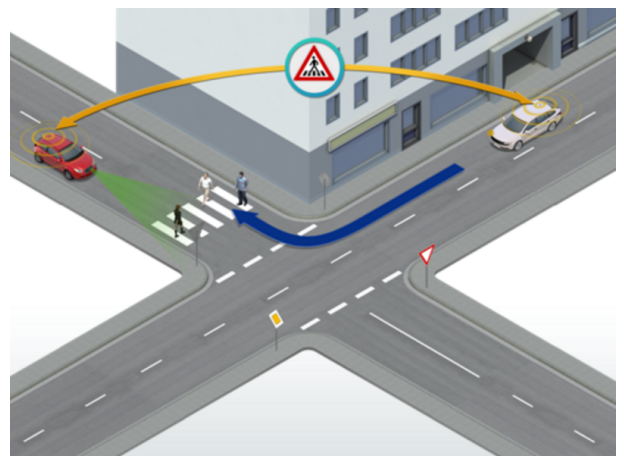


Figure 1: A propos de C-ITS

Les C-ITS permettent l'interconnexion entre les utilisateurs dans l'environnement de trafic en leur permettant de communiquer entre eux. Dans les C-ITS, les usagers de la route jouent un rôle im-

portant en termes de perception de l'environnement et de diffusion de l'information grâce à l'amélioration de l'interconnectivité, les approches C-ITS peuvent s'attaquer aux causes profondes d'une collision et, par conséquent, minimiser les accidents liés aux VRU... Équipés de capteurs et de technologies de communication, les usagers de la route s'échangent des données d'état actualisées (emplacement, vitesse, etc.) afin de créer et de maintenir une conscience coopérative. Véhicules à proximité (**Vehicle-to-Vehicle, V2V**), infrastructure (**Vehicle-to-Infrastructure, V2I**), piétons (**Vehicle-to-Pedestrian, V2P**) ou d'autres stations (**Vehicle-to-Everything, V2X**) peuvent être les canaux de communication de ce processus. Des progrès significatifs sont réalisés dans le domaine des C-ITS pour faire des communications V2V une réalité afin d'assurer la sécurité et le confort du conducteur. Par conséquent, les systèmes de communication V2X ont été principalement développés et testés pour les voitures et les camions, ce qui montre que l'on se préoccupe moins des usagers vulnérables de la route et de l'intégration des URV dans les communications V2X n'a été que très peu explorée [2]

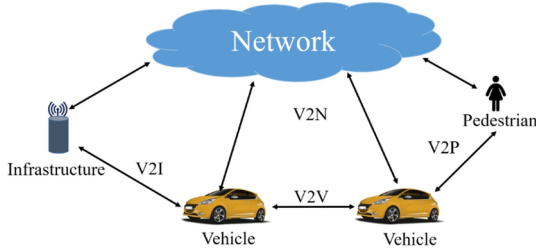


Figure 2: Divers scénarios V2X dans les systèmes IoT [3]

Dans un système **Vehicle-to-Vulnerable Road User (V2VRU)** cellulaire, les calculs de risque de collision et les algorithmes de génération d'avertissements de sécurité s'exécutent sur les dispositifs terminaux de l'utilisateur afin d'augmenter les performances du système en réduisant les frais généraux du serveur. En fait, lorsque l'extrémité mobile effectue des calculs à la place d'un serveur central, il devrait être capable de traiter les calculs efficacement sans aucun retard et ne devrait pas affecter lourdement la consommation d'énergie du dispositif. Néanmoins, les anciennes versions des smartphones peuvent ne pas avoir la capacité de traitement d'algorithmes complexes en raison de la puissance de calcul de ces appareils et de technologies dépréciées. Par conséquent, pour éviter les problèmes de pénétration du marché avec le système V2VRU, **Mobile Edge computing (MEC)** peut être envisagée pour les demandes de calcul. Le MEC est devenu une technologie évolutive qui étend la capacité de cloud computing centralisée conventionnelle à l'Edge, plus près des dispositifs de l'utilisateur final[4].

*: Wireless Internet access via mobile networks connu sous le nom Mobile broadband (MBB).

2 Mots clés :

C-ITS, V2X, Vulnerable Road User, V2VRU, MEC, Edge...

3 Modèle et formulation du problème :

Nous considérons un serveur Edge cloud unique ayant des coordonnées cartésiennes $s \triangleq (x, y)$, I utilisateurs finals (Véhicules, passagers, cyclistes ...) de coordonnées $u_i \triangleq (x_i, y_i)$ $i \in \mathbb{I} = \{1, 2, \dots, I\}$ et J MEC serveurs de coordonnées $u_j \triangleq (x_j, y_j)$ $j \in \mathbb{J} = \{1, 2, \dots, J\}$ [5].

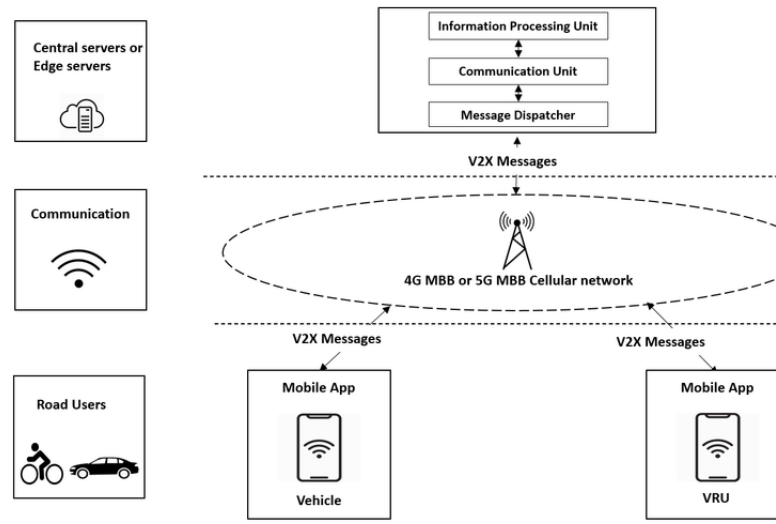


Figure 3: Une architecture générale de système pour un système V2VRU basé sur le MBB* 4G.

Chaque dispositif local est connecté au serveur MEC par une liaison sans fil, et chaque serveur MEC peut être connecté au serveur EDGE cloud par une liaison filaire. On suppose qu'il existe T time slots et que chaque intervalle de time slot est τ , et que chaque dispositif génère une tâche de calcul de taille fixe (bits) dans chaque tranche de temps $t \in \mathbb{T} = \{1, 2, \dots, T\}$

Au t -ième time slot, on peut choisir de traiter la tâche localement, par l'intermédiaire du MEC ou des serveurs Edge Cloud, ou même de ne pas la traiter. Dans ce système, la tâche de calcul ne peut choisir qu'un seul des modes de traitement. C'est-à-dire :

$$\begin{cases} X_{i,l}^t + X_{i,m}^t + X_{i,c}^t + X_{i,d}^t = 1, i \in \mathbb{I}, t \in \mathbb{T} \\ X_{i,l}^t, X_{i,m}^t, X_{i,c}^t, X_{i,d}^t = \{0, 1\}, i \in \mathbb{I}, t \in \mathbb{T} \end{cases} \quad (1)$$

[6]

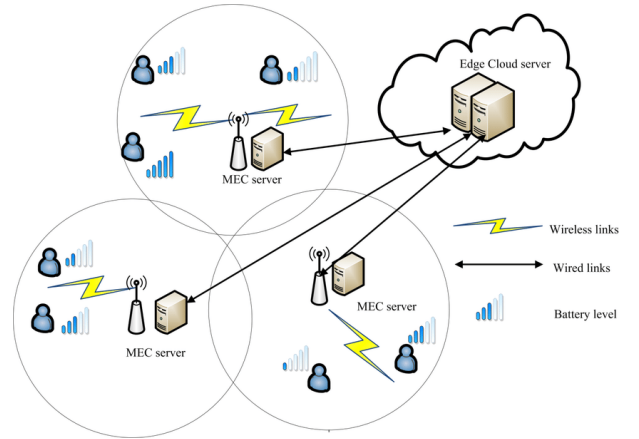


Figure 4: Architecture du MEC

3.1 Modèle de calcul :

Ici, on confond les véhicules, les passagers et les cyclistes. On suppose également que ces utilisateurs finaux sont conscients de l'environnement qui les entoure. Cela signifie qu'ils disposent des données nécessaires pour effectuer l'évaluation des risques et le traitement de génération d'alertes. Une distinction est faite entre les véhicules et les smartphones utilisés par les passagers et les cyclistes à des fins énergétiques, car l'énergie des voitures est très suffisante.

On définit, donc, \mathbb{V} : l'ensemble des véhicules et \mathbb{S} : l'ensemble des smartphones.

$$\mathbb{I} = \mathbb{V} \cup \mathbb{S}$$

3.1.1 Calcul local :

La consommation en temps :

Le premier mode de traitement est que la tâche est sélectionnée pour être traitée localement. Le retard généré est uniquement le temps de traitement du dispositif local.

$$\begin{cases} T_{i,l}^t = \frac{D_i^t C}{f_i^t}, i \in \mathbb{I}, t \in \mathbb{T} \\ 0 \leq f_i^t \leq f_l^{\max} \end{cases} \quad (2)$$

D_i^t : La taille de la tâche générée par le end user i.

C : Nombre de cycles CPU (Processeur) nécessaires pour traiter un bit de données.

f_i^t : fréquence de CPU de i.

f_l^{\max} : maximum des fréquences de \mathbb{I} .

La consommation en energie:

$$E_{i,l}^t = \begin{cases} 0 & \text{si } i \in \mathbb{V} \\ k D_i^t C (f_i^t)^2, i \in \mathbb{I}, t \in \mathbb{T} & \text{si } i \in \mathbb{S}. \end{cases} \quad (3)$$

[6]

k : le coefficient de capacité effective.

3.1.2 Calcul dans le serveur MEC :

Soit $\lambda_{i,j} \in \{0,1\}$ dénote la variable d'association entre MEC serveur j et le end user i où $\lambda_{i,j} = 1$ indique que le end user i est desservie par MEC serveur j et $\lambda_{i,j} = 0$ indique le contraire.

Donc :

$$\sum_{j=1}^{\mathbb{J}} \lambda_{i,j} = 1, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \quad (4)$$

La tâche générée localement est transférée au serveur MEC par le biais de la liaison sans fil. En supposant que les largeurs de bande des canaux sans fil sont distribuées de manière égale et qu'elles n'interfèrent pas entre eux, on obtient le taux maximum possible de la transmission par canal ($r_{i,j}$: capacité) selon la formule de Shannon Hartley :

$$\begin{cases} r_{i,j}^t = B \log_2 \left(1 + \frac{h_{i,j}^t p_i^t}{\sigma^2} \right), i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \\ 0 \leq p_i^t \leq p_{\max}, i \in \mathbb{I}, t \in \mathbb{T} \end{cases} \quad (5)$$

B : channel bandwidth

p_i^t : transmit power

σ^2 : noise power

$h_{i,j}^t$: channel gain

$$h_{i,j}^t = \gamma_{i,j}^t g_0 (d_0/d_{i,j}^t)^\alpha$$

$\gamma_{i,j}^t$: fading power gain

g_0 : loss constant

d_0 : relative distance

$d_{i,j}^t = \|u_i^t - v_j^t\|$: La distance entre i et j.

α : loss index

La consommation en temps :

$$T_{i,j}^t = D_i^t / r_{i,j}^t + D_i^t C / f_{\text{mec}}, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \quad (6)$$

f_{mec} : La fréquence du serveur MEC.

C'est le temps coûté lors de la transmission de la tâche au serveur MEC, et le temps de traitement du le serveur MEC

La consommation en énergie :

Puisque le déchargement des tâches vers le serveur MEC via la liaison sans fil nécessite l'énergie du dispositif mobile local, la consommation d'énergie ne prend en compte que la puissance d'émission du dispositif local, et la consommation énergétique du deuxième scénario de traitement est exprimée comme suit .

$$E_{i,j}^t = \begin{cases} 0 & \text{si } i \in \mathbb{V} \\ p_i^t \frac{D_i^t}{r_{i,j}^t}, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T}, t \in \mathbb{T} & \text{si } i \in \mathbb{S}. \end{cases} \quad (7)$$

3.1.3 Calcul dans le serveur EDGE cloud :

La consommation en temps :

$$T_{i,c}^t = D_i^t / r_{i,j}^t + H D_i^t / r_{mc} + \frac{D_i^t C}{f_c}, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \quad (8)$$

H : Nombre de liaisons filaires entre le serveur MEC et le serveur cloud.

r_{mc} : Channel transmission rate entre le serveur MEC et l'EDGE cloud.

La consommation en énergie :

Comme pour le mode de traitement MEC, lorsque $X_{i,c}^t = 1$, la tâche doit décider par quel serveur MEC télécharger les données vers le serveur EDGE cloud, définissant ainsi $g_{i,j}^t = \{0,1\}$. En outre, $g_{i,j}^t$ doit satisfaire aux critères suivants :

$$\begin{cases} \sum_{j=1}^{\mathbb{J}} g_{i,j}^t = 1, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \\ \sum_{i=1}^{\mathbb{I}} g_{i,j}^t \leq H, j \in \mathbb{J}, t \in \mathbb{T} \end{cases} \quad (9)$$

$$E_{i,c}^t = E_{i,j}^t = \begin{cases} 0 & \text{si } i \in \mathbb{V} \\ p_i^t \frac{D_i^t}{r_{i,j}^t}, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} & \text{si } i \in \mathbb{S}. \end{cases} \quad (10)$$

3.2 Conclusion :

3.2.1 Latence :

$$\begin{cases} T_{i,\text{all}}^t = X_{i,l}^t T_{i,l}^t + \sum_{j=1}^{\mathbb{J}} \lambda_{i,j}^t T_{i,j}^t + X_{i,c}^t T_{i,c}^t, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \\ 0 \leq T_{i,\text{all}}^t \leq \tau \end{cases} \quad (11)$$

$$\begin{cases} U_j^t \triangleq \sum_{i=1}^{\mathbb{I}} \frac{D_i^t \lambda_{i,j}^t C}{f_{\text{mec}} \tau}, j \in \mathbb{J}, t \in \mathbb{T} \\ U_j^t \leq 1, j \in \mathbb{J}, t \in \mathbb{T} \end{cases} \quad (12)$$

τ : Temps de réponse maximum possible.

U_j^t : l'utilisation de CPU du serveur MEC j.

On définit les variables suivantes :

$$A_i^t \triangleq [X_i^t, f_i^t, p_i^t, \lambda_i^t, g_i^t], i \in \mathbb{I}, t \in \mathbb{T}$$

$$\mathbf{X}_i^t \triangleq [X_{i,l}^t, X_{i,m}^t, X_{i,c}^t, X_{i,d}^t], i \in \mathbb{I}, t \in \mathbb{T}$$

$$\lambda_i^t \triangleq [\lambda_{i,1}^t, \lambda_{i,2}^t, \dots, \lambda_{i,j}^t], i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T}$$

$$g_i^t \triangleq [g_{i,1}^t, g_{i,2}^t, \dots, g_{i,j}^t], i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T}$$

Le problème est donc:

$$P1 : \begin{cases} \min_{A^t} \sum_{i=1}^{\mathbb{I}} T_{i,\text{all}}^t + \Phi \{X_{i,d}^t = 1\}, t \in \mathbb{T} \\ \text{st : (1)(2)(4)(5)(9)(11)(12)} \end{cases}$$

Φ : fonction de pénalisation.

Exemple : Fonction indicatrice

Si l'on est familier avec les fonctions pouvant prendre la valeur $+\infty$, on comprendra aisément qu'une fonction de pénalisation naturelle pour $X_{i,d}^t$ est :

$$\Phi : \mathbb{R} \rightarrow \cup\{+\infty\} : x \mapsto \Phi(x) = \begin{cases} 0 & \text{si } x = 0 \\ +\infty & \text{si } x = 1. \end{cases}$$

On adapte cette fonction à notre contexte. Pour nous, l'infini est τ .

$$\Phi : \mathbb{R} \rightarrow : x \mapsto \Phi(x) = \begin{cases} 0 & \text{si } x = 0 \\ \tau & \text{si } x = 1. \end{cases}$$

3.2.2 Énergie :

$$\left\{ \begin{array}{l} E_{i, \text{all}}^t = X_{i,l}^t E_{i,l}^t + \sum_{j=1}^{\mathbb{J}} \lambda_{i,j}^t E_{i,j}^t + X_{i,c}^t E_{i,c}^t, i \in \mathbb{I}, j \in \mathbb{J}, t \in \mathbb{T} \\ E_{i, \text{all}}^t \leq b_i^t \leq E_{\max} \\ b_i^{t+1} = b_i^t - E_{i, \text{all}}^t + e_i^t \\ 0 \leq e_i^t \leq E_{i,H}^t \end{array} \right. \quad (13)$$

b_i^t : Niveau de batterie le pendant t-ème time slot de l'appareil local i.

E_{\max} : Maximum d'énergie d'un appareil local i.

e_i^t : L'énergie chargée pendant le t-ème time slot de l'appareil local i.

$E_{i,H}^t$: Maximum d'énergie chargée pendant le t-ème time slot de l'appareil local i.

4 Le problème :

Dans cette partie, on doit tenir en compte l'énergie des appareils locaux (téléphones) dans le problème de minimisation globale. L'idée est de faire une combinaison homogène entre l'énergie et la latence, donc le problème de minimisation devient :

$$P2 : \left\{ \begin{array}{l} \min_{A^t} \sum_{i=1}^{\mathbb{I}} \frac{T_{i, \text{all}}^t + \Phi\{X_{i,d}^t = 1\}}{\tau} + \gamma \frac{E_{i, \text{all}}}{E_{\max}}, t \in \mathbb{T} \\ st : (1)(2)(4)(5)(9)(11)(12)(13) \end{array} \right.$$

γ : Un paramètre d'étalement à définir.

Si $\gamma = 0$, on ne prend pas en compte l'énergie.

Sinon $\gamma > 0$, l'importance de l'énergie dépend de γ .

Cette approche globale permet une bonne répartition des ressources, contrairement à une approche partielle qui ne tient pas compte des autres usagers de la route.

5 Algorithme de résolution :

6 Simulation :

Pas de données disponibles.

7 Conclusion :

Avec l'émergence de tâches de plus en plus gourmandes en ressources informatiques et sensibles aux délais, le traitement des tâches informatiques sur des serveurs cloud ne peut plus répondre aux besoins actuels. L'émergence de la technologie MEC (Mobile Edge Computing) et la popularité des applications peuvent résoudre ces demandes. Le déchargement des tâches sur le serveur MEC réduit la consommation d'énergie des appareils locaux et présente également une latence plus faible que le déchargement sur le serveur cloud.

8 Références :

[1]: Principaux faits, Accidents de la route, Organisation mondiale de la santé (OMS).

[2] : C-ITS: Cooperative Intelligent Transport Systems and Services.

[3] : EURASIP Journal on Wireless Communications and Networking.

[4] : Improving Vulnerable Road User Safety: Existing Practices and Consideration for Using Mobile Devices for V2X Connections.

[5] : Latency Optimization of UAV-Enabled MEC System for Virtual Reality Applications Under Rician Fading Channels.

[6] : Resource Allocation Strategy of Edge Systems Based on Task Priority and an Optimal Integer Linear Programming Algorithm.

[7] : Context-aware task offloading with QoS-provisioning for MEC multi-RAT vehicular networks.