



INGÉNIERIE DES RÉSEAUX

U.E SCIENCE DES RÉSEAUX ET APPRENTISSAGE

**Rapport sur le projet : Multi-armed bandit (MAB) pour
une sélection de communication directe ou via
infrastructure dans le cadre des communications V2X**

AUTEURS :

*HUGO LE CLAINCHE, LUCAS THIETART, MATHIS SIGIER, THOMAS
GRUGET*

Table des matières

1	Introduction	1
2	Etat de l'art et compréhension du problème	1
2.1	Généralités/Compréhension du sujet	1
2.2	Discussion sur les QoS	2
3	Modélisation du problème et simulations	3
3.1	NS3 - section annexe	3
3.2	Modélisation V2V	3
3.3	Modélisation V2I	5
4	Algorithmes MAB et présentation des résultats	5
4.1	Présentation des algorithmes MAB	5
4.1.1	L'algorithme Epsilon Greedy	5
4.1.2	L'algorithme UCB	7
4.2	Analyse des performances et comparaison des cas obtenus	8
4.2.1	Première simulation - variations de distance :	8
4.2.2	Seconde simulation - Variation de la charge sur le réseau 5g	9
4.2.3	Troisième simulation - Adaptabilité des aglorithmes	10
5	Difficultés rencontrées et axes d'amélioration	11
6	Conclusion	11
7	Bibliographie :	11

1 Introduction

L'amélioration de la sécurité des usagers vulnérables de la route (VRU) tels que les piétons, les cyclistes et les motocyclistes est un enjeu majeur dans le domaine des transports intelligents et de la sécurité routière. Pour répondre à cette problématique, l'utilisation d'applications de protection des VRU qui s'appuient sur les données de localisation GPS pour évaluer les situations de danger entre les VRU et les véhicules est une approche prometteuse. Ces applications peuvent générer des alertes et les transmettre à différents acteurs, tels que les systèmes embarqués dans les véhicules, les VRU eux-mêmes ou les infrastructures routières.

Toutefois, il est important de souligner que toutes les routes ne disposent pas d'infrastructures de communication dédiées pour assurer l'échange d'informations entre les différents acteurs. Ainsi, il est essentiel de prendre des décisions éclairées concernant la manière d'établir la communication entre les VRU et les véhicules, en prenant en compte divers facteurs tels que la qualité du signal, la proximité, la charge du réseau, etc.

Dans ce contexte, l'utilisation d'algorithmes Multi-Armed Bandit (MAB) s'avère particulièrement pertinente pour la gestion de la communication entre les VRU et les véhicules. Les algorithmes MAB permettent de prendre des décisions optimales en explorant et en exploitant différentes options disponibles, en se basant sur des critères prédéfinis. Ces critères peuvent inclure la proximité des VRU, la latence entre l'envoi et la réception du message, la charge du réseau, et d'autres facteurs pertinents. L'objectif principal de ces algorithmes est d'obtenir les meilleurs résultats en termes de qualité de service (QoS) considérés, en choisissant entre deux branches de communication : la communication Véhicule-à-Infrastructure (V2I) ou la communication Véhicule-à-Véhicule (V2V).

Dans le cadre de ce projet, nous avons tout d'abord effectué une analyse détaillée de l'état de l'art sur le sujet. Nous avons également réfléchi à l'aspect QoS du problème avant de nous lancer dans la réalisation de simulations et l'implémentation des algorithmes MAB UCB et Epsilon Greedy. Nous avons analysé et comparé les performances de ces algorithmes en fonction de différents cas de fonctionnement. Ce rapport présente les différentes étapes de notre projet et les résultats obtenus.

2 Etat de l'art et compréhension du problème

2.1 Généralités/Compréhension du sujet

Une infrastructure 5G routière est un réseau de communication sans fil qui utilise la technologie 5G pour fournir une connectivité à haut débit et à faible latence aux véhicules en mouvement. L'objectif principal de ce type d'infrastructure est d'améliorer la sécurité routière, la gestion du trafic et la mobilité en général.

Dans une infrastructure 5G routière, les véhicules sont équipés de dispositifs de communication sans fil (tels que des modems 5G), qui leur permettent de se connecter au réseau 5G. Les véhicules peuvent également être équipés de capteurs, tels que des caméras, des radars et des lidars, qui leur permettent de détecter leur environnement et de communiquer avec d'autres véhicules directement.

Les véhicules peuvent avoir connaissance de la position et de la vitesse des véhicules autour d'eux grâce aux technologies de communication V2V et V2I. La communication V2V permet aux véhicules de communiquer entre eux et d'échanger des informations sur leur position, leur vitesse et leur direction. La communication V2I permet aux véhicules de communiquer avec les infrastructures routières, telles que les feux de circulation et les panneaux de signalisation, pour obtenir des informations sur la circulation et les conditions routières.

Lorsqu'un utilisateur (véhicule) souhaite transmettre un message d'urgence autour de lui, il peut utiliser la technologie de communication V2V pour envoyer le message directement aux autres véhicules dans la

zone. Le message peut également être transmis aux infrastructures routières via la communication V2I, qui peuvent à leur tour diffuser le message à un plus grand nombre de véhicules.

Dans le cadre de notre projet, il est essentiel de comprendre le contexte et les enjeux liés à l'utilisation des algorithmes Multi-Armed Bandit (MAB) pour la gestion de la communication entre les usagers vulnérables de la route (VRU) et les véhicules.

Les algorithmes MAB sont fondés sur un principe de prise de décision séquentielle, qui consiste à explorer et à exploiter différentes options disponibles, en se basant sur des critères prédéfinis, afin d'obtenir les meilleurs résultats possibles. Dans notre cas, l'objectif est d'optimiser la qualité de service (QoS) en choisissant la meilleure branche de communication entre la V2I (Véhicule-à-Infrastructure) et la V2V (Véhicule-à-Véhicule).

Pour mettre en œuvre et évaluer les performances de nos algorithmes MAB, il est nécessaire de réaliser des simulations qui permettent de calculer, à chaque itération, la récompense associée à la décision prise. Cette récompense est un élément clé de l'apprentissage et de l'optimisation de l'algorithme.

À ce stade, une question essentielle se pose : de quoi va dépendre notre récompense ? Autrement dit, quelles sont les métriques que nous souhaitons récupérer et optimiser, et pourquoi ? Pour répondre à cette question, il est important d'analyser les différents facteurs et indicateurs de performance qui influencent la QoS dans le contexte de la communication entre les VRU et les véhicules. La sous-section suivante sera consacrée au choix des QoS retenus pour notre projet.

2.2 Discussion sur les QoS

Dans la continuité de notre réflexion sur les métriques à prendre en compte pour la récompense de nos algorithmes MAB, il est nécessaire d'analyser en détail le problème et les enjeux liés à la communication entre les usagers vulnérables de la route (VRU) et les véhicules.

Pour cela, nous avons mené une recherche documentaire approfondie sur les différents types de messages échangés dans ce contexte, les protocoles de communication utilisés en pratique, et les exigences en termes de qualité de service (QoS). Cette étape nous a permis de mieux cerner les besoins et les contraintes auxquels nous devons répondre dans notre projet.

En nous appuyant sur les résultats de cette analyse, nous avons commencé à justifier les choix que nous avons faits dans nos simulations, et en particulier le choix de la métrique principale que nous avons retenue pour la récompense de nos algorithmes MAB : le délai de transmission.

En effet, dans le contexte de la sécurité routière et de la protection des VRU, le délai de transmission des messages d'alerte ou d'information est un facteur critique, qui peut avoir un impact direct sur la sécurité des usagers de la route. Réduire ce délai au minimum est donc une priorité absolue, qui doit guider nos efforts d'optimisation de la communication entre les VRU et les véhicules.

D'autres métriques, telles que la fiabilité de la transmission, le taux de perte de paquets, ou encore la consommation énergétique, peuvent également être pertinentes dans ce contexte. Toutefois, après avoir pesé le pour et le contre, nous avons choisi de nous concentrer sur le délai de transmission comme métrique principale pour notre projet, afin de nous assurer de répondre au mieux aux enjeux de sécurité routière qui sont au cœur de notre préoccupation.

Nous reviendrons plus en détail dans les sections suivantes sur les choix que nous avons faits pour nos simulations, et sur les résultats que nous avons obtenus en termes d'optimisation de la communication entre les VRU et les véhicules à l'aide de nos algorithmes MAB.

3 Modélisation du problème et simulations

L'objectif principal de nos simulations est de permettre l'évaluation des performances et l'efficacité de nos algorithmes Multi-Armed Bandit (MAB) dans la gestion de la communication entre les usagers vulnérables de la route (VRU) et les véhicules. En particulier, nous cherchons à optimiser la qualité de service (QoS) en choisissant la meilleure branche de communication entre la V2I et la V2V, par rapport à la latence qu'elle implique.

Pour cela, nous avons développé un modèle de simulation qui prend en compte les différents paramètres et contraintes du système de communication tels que la mobilité des véhicules, les caractéristiques des messages échangés, les protocoles de communication utilisés, les temps de propagation et de traitement, mais aussi la congestion et la charge associées aux trafics V2V et V2I.

Dans les sections suivantes, nous présenterons en détail notre modèle de simulation, les scénarios que nous avons considérés, les résultats que nous avons obtenus, et les enseignements que nous avons tirés de cette étude.

3.1 NS3 - section annexe

Bien que nous ayons initialement envisagé d'utiliser le simulateur réseau NS-3 pour nos simulations, nous avons finalement décidé de nous concentrer sur des simulations Python personnalisées. Cette décision a été prise après avoir passé un certain temps à explorer les fonctionnalités de NS-3 et à évaluer si elles répondaient à nos besoins spécifiques.

Nous avons constaté que, bien que NS-3 soit un outil puissant et largement utilisé pour la simulation de réseaux, il présentait certains inconvénients pour notre projet. Tout d'abord, la courbe d'apprentissage de NS-3 est assez raide, et nous avons estimé que le temps nécessaire pour nous familiariser avec l'outil et le personnaliser pour notre projet serait trop long pour des bénéfices limités.

Dans l'ensemble, nous avons utilisé un modèle de simulation général pour nos deux scénarios, en nous concentrant sur la modélisation des véhicules, des messages et des protocoles de communication pertinents en fonction du mode de connexion emprunté.

3.2 Modélisation V2V

La modélisation de la communication V2V pour les simulations repose sur les réseaux ad hoc véhiculaires (VANETs). La qualité de service (QoS) dépend de la distance entre l'émetteur et le récepteur ainsi que de la densité du trafic dans la zone concernée.

Le code de modélisation V2V définit plusieurs paramètres, tels que la distance entre les véhicules, la densité de trafic, la taille des paquets d'alerte, le débit de communication, et des constantes comme la vitesse de propagation et la longueur moyenne d'un véhicule.

Dans les réseaux VANETs, la norme WiFi (IEEE 802.11p) est utilisée pour transmettre les messages de sécurité de base (BSM), d'une taille de 800 bits. Nous simulerons la perte de puissance du signal en fonction de la distance avec l'équation suivante : L'équation de perte de puissance en fonction de la distance est donnée par :

$$P_{\text{perte}} = P_0 - 10 \cdot n \cdot \log_{10}(d)$$

où P_{perte} est la perte de puissance, P_0 est la puissance initiale, n est l'exposant de perte de signal, et d est la distance entre l'émetteur et le récepteur. De plus, on fixe une distance limite de transmission de 1 km sera fixée. Plus le signal est faible, plus la probabilité de perte de signal augmente.

Le message d'urgence sera envoyé en mode broadcast pour maximiser les chances de transmission. Afin de tester tous les chemins possibles, nous utiliserons une méthode récursive qui ne retournera que le chemin avec le délai le plus court.

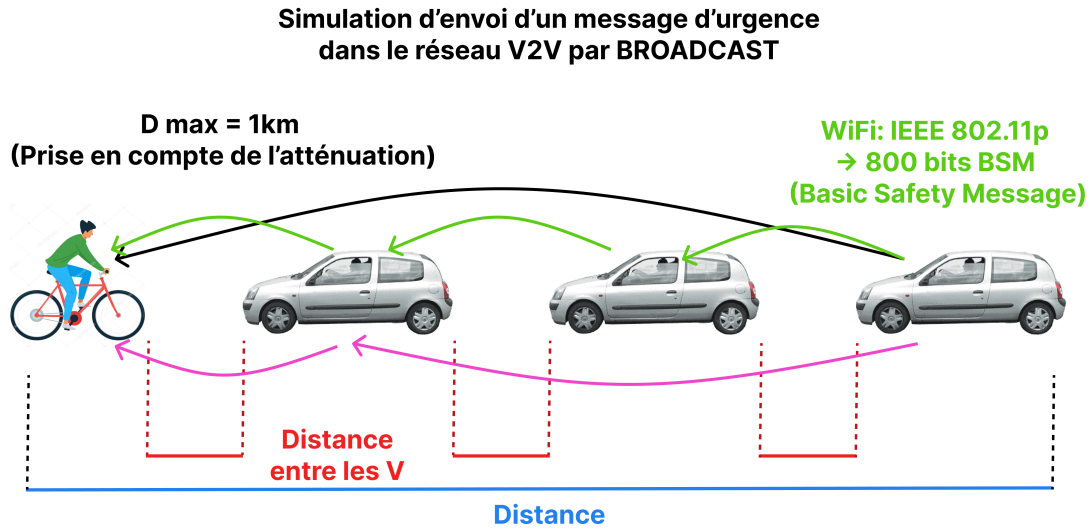


FIGURE 1 – Schéma de la simulation V2V

3.3 Modélisation V2I

La modélisation de la communication V2I dans notre simulation est basée sur les réseaux cellulaires. La qualité de service (QoS) dépend de deux facteurs principaux : la distance entre l'émetteur et le récepteur, et la charge du réseau.

Pour la distance, nous avons choisi d'utiliser une valeur moyenne comprise entre 100 et 400 mètres entre un véhicule et l'infrastructure en zone urbaine en France. Cette valeur peut varier en fonction de l'emplacement et du type d'infrastructure, nous le ferons varier dans nos simulations.

Pour la charge du réseau, nous avons utilisé un pourcentage de charge réseau comme entrée pour notre modèle. Cela nous permet de simuler différents niveaux de charge réseau et d'évaluer leur impact sur la QoS, le niveau de charge va allonger proportionnellement durée d'envoi.

Nous avons également utilisé un facteur d'atténuation de 2 pour la communication, ce qui signifie qu'il n'y a pas d'obstacles entre l'émetteur et le récepteur. Cela simplifie notre modèle et nous permet de nous concentrer sur les facteurs de distance et de charge réseau.

En ce qui concerne la taille du paquet d'alerte, nous avons choisi d'utiliser une taille de 120 Octets pour notre simulation. Cette taille est basée sur le format de message GeoMIP (GeoNetworking Mobility Information Protocol), un protocole de communication utilisé dans les réseaux de transport intelligents (ITS) pour échanger des informations sur la mobilité des véhicules. Il s'agit d'un protocole d'échange de données GPS via IP au travers de l'infrastructure routière 5g.

Enfin, nous avons utilisé des débits de 50 Mbit/s pour le traitement des données radio et de 20 Mbit/s pour l'émission des données radio dans notre modèle. Ces débits sont basés sur les capacités des réseaux cellulaires actuels et nous permettent de simuler des scénarios réalistes.

Dans l'ensemble, nous avons fait des choix d'approximation raisonnables pour notre modèle de communication V2I, en nous concentrant sur les facteurs clés de la distance et de la charge réseau. Ce modèle de simulation nous renvoie la latence associée à l'envoi d'un message de signalisation d'un VRU sur l'infrastructure 5g en fonction de la congestion sur l'infrastructure et de la distance avec le véhicule ciblé.

4 Algorithmes MAB et présentation des résultats

4.1 Présentation des algorithmes MAB

Explication des algo MAB

4.1.1 L'algorithme Epsilon Greedy

L'algorithme epsilon greedy fonctionne en équilibrant l'exploration et l'exploitation. Il maintient une estimation de la récompense moyenne pour chaque bras de la machine, et utilise cette estimation pour prendre des décisions. À chaque étape, l'algorithme choisit de manière aléatoire entre deux options : explorer ou exploiter.

Lorsque l'algorithme décide d'explorer, il tire un bras de la machine de manière aléatoire, indépendamment de ses estimations de récompense. Cela lui permet de découvrir de nouvelles informations sur l'environnement et d'améliorer ses estimations de récompense.

Lorsque l'algorithme décide d'exploiter, il tire le bras de la machine avec la récompense moyenne estimée la plus élevée. Cela lui permet de maximiser son gain à court terme en utilisant les informations qu'il a déjà recueillies.

Le paramètre epsilon contrôle la probabilité d'exploration. Au début de l'algorithme, la valeur de epsilon est élevée, ce qui signifie que l'algorithme explore plus souvent. Au fil du temps, la valeur de epsilon diminue, ce qui signifie que l'algorithme exploite plus souvent. Cela permet à l'algorithme de se concentrer sur les bras de la machine les plus rentables à mesure qu'il accumule des informations sur l'environnement.

Dans le code fourni (V2V.py), l'algorithme epsilon greedy est implémenté pour un environnement avec $k=2$ bras.

```

for t in range(T):
    # On définit le paramètre epsilon pour l'exploration
    with np.errstate(divide='ignore'):
        epsilon = np.power(t, -1/3) * np.power(k * np.log(t), 1/3)

    # Exploration-Exploitation Strategy
    if np.random.rand() < epsilon:          # Lancé de la pièce pour choisir entre exploration et exploitation
        # EXPLORATION
        arm = np.random.randint(k)
    else:
        # EXPLOITATION
        arm = np.argmax(est_means)          # Choisir le bras avec le PLUS PETIT TEMPS DE TRANSMISSION

    # On calcule la récompense pour chaque bras de la machine
    rewards_iteration = env.get_reward()

    reward = rewards_iteration[arm]          # On récupère le temps de transmission pour le bras choisi
    n[arm] += 1                             # On incrémente le nombre de fois que le bras choisi a été tiré
    rewards[arm] += reward                  # On ajoute la récompense observée aux récompenses cumulées pour le bras choisi
    est_means[arm] = rewards[arm] / n[arm]  # On met à jour la récompense moyenne estimée pour le bras choisi

```

FIGURE 2 – Code de la partie centrale de l'algorithme Epsilon Greedy - Python

À chaque étape, l'algorithme calcule la valeur de epsilon en utilisant une formule spécifique (décroissante avec le temps), puis décide d'explorer ou d'exploiter en fonction de cette valeur. Il met à jour ses estimations de récompense en fonction de la récompense observée, et calcule le regret de cette étape en comparant la récompense observée à la récompense optimale. Le regret cumulé est utilisé pour mesurer la performance de l'algorithme.

4.1.2 L'algorithme UCB

L'algorithme UCB fonctionne en équilibrant l'exploration et l'exploitation en utilisant une valeur appelée "UCB" pour chaque bras de la machine. La valeur UCB est calculée en ajoutant une incertitude à la récompense moyenne estimée pour chaque bras. Au début de l'algorithme, l'incertitude est élevée pour tous les bras, ce qui signifie que l'algorithme explore plus souvent. Au fil du temps, l'incertitude diminue pour les bras les plus rentables, ce qui signifie que l'algorithme exploite plus souvent.

```

for t in range(T):
    if t < k:
        # Jouer chaque bras k fois pour initialiser les estimations et les valeurs UCB
        reward_iteration = env.get_reward()
        reward = reward_iteration[t]
        n[t] += 1
        rewards[t] += reward
        est_means[t] = rewards[t] / n[t]
        regrets.append(0)
    else:
        reward_iteration = env.get_reward() # Obtenir la récompense pour chaque bras de la machine

        # Choisir le bras avec la plus grande valeur UCB
        ucb_values = [est_means[i] + np.sqrt(2*np.log(t) / n[i]) for i in range(k)] # Calculer les valeurs UCB pour chaque bras
        arm = np.argmax(ucb_values) # Sélectionner le bras avec la PLUS PETITE VALEUR UCB

        reward = reward_iteration[arm] # Obtenir la récompense pour le bras choisi

        n[arm] += 1 # Incrémente le nombre de fois que le bras choisi a été tiré
        rewards[arm] += reward # Ajouter la récompense observée aux récompenses cumulées pour le bras choisi
        est_means[arm] = rewards[arm] / n[arm] # Mettre à jour la récompense moyenne estimée pour le bras choisi

```

FIGURE 3 – Code de la partie centrale de l'algorithme Epsilon Greedy - Python

Dans le code fourni, l'algorithme UCB est aussi implémenté pour un environnement avec $k=2$ bras. Au début de l'algorithme, chaque bras est tiré $k=2$ fois pour initialiser les estimations de récompense et les valeurs UCB. À chaque étape suivante, l'algorithme calcule les valeurs UCB pour chaque bras et sélectionne le bras avec la plus petite valeur UCB (car dans ce cas, l'objectif est de minimiser le temps de transmission). Il met à jour ses estimations de récompense en fonction de la récompense observée, et calcule le regret de cette étape en comparant la récompense observée à la récompense optimale. Le regret cumulé est utilisé pour mesurer la performance de l'algorithme.

4.2 Analyse des performances et comparaison des cas obtenus

Pour étudier et comparer en détails les algorithmes UCB et MAB nous avons identifié différents cas de fonctionnement que nous avons simulé. Nous avons ensuite testé les algorithmes sur ces cas simulés.

4.2.1 Première simulation - variations de distance :

Dans cette première simulation nous considérons une variation uniforme de la distance entre 1100m et 1400m, sur 500 itérations (envoies de paquet entre le VRU et un véhicule).

```
V2V_simulation: [True, 0.0010006993333333335, 0, [0, 5,
11]]
final latency: 0.0010006993333333335
-----
V2I_simulation:
final latency 0.0008512506765411443
-----
V2I has best latency for distance 1200 m
-----
Distance: 1400
Traffic density: 0.05
V2V_simulation: [True, 0.0010006899999999998, 0, [0, 4,
13]]
final latency: 0.0010006899999999998
-----
V2I_simulation:
final latency 0.0008748188209564401
-----
V2I has best latency for distance 1400 m
```

FIGURE 4 – Comparaison des latences des deux branches autour de 1200M

Comme l'illustre la figure ci dessus, entre 1100m et 1400m, la voie V2I est la branche qu'il est préférable d'utiliser en terme de latence. On s'attend donc à ce que nos algorithmes utilisent cette voie. Observons maintenant les résultats des deux algorithmes. Nous allons tracer les courbes des cumules des regrets en fonction du temps (nombre d'itérations) pour les deux algorithmes. Le cumule des regrets c'est la somme des temps perdus par rapport au choix idéal sur chaque itération, ainsi, lorsqu'un algorithme fait le bon choix on ajoute 0s à son cumule des regrets, s'il se trompe on ajoute la différence entre la latence du choix de l'algorithme et celle du choix optimal.

Nous pouvons faire plusieurs analyses sur ce premier tracé. On constate d'abord qu'UCB est très performant, ce qui est cohérent puisqu'il sélectionne la branche V2I comme branche à utiliser et les performances (rewards) de la branche restent toujours supérieures à celles de V2V, ce qui ne le pousse pas à essayer la branche V2V. Dans notre présentation nous avons aussi illustré le cas d'une variation uniforme à courte distance (200m-800m), et UCB offrait les mêmes performances. UCB est donc logiquement (au vu de sa constitution) très performant dans les cas où la branche optimale ne varie pas.

Epsilon-greedy offre, lui, des performances moins séduisantes mais toujours cohérente avec son fonctionnement intrinsèque. En effet le paramètre d'exploration associé à l'algorithme le pousse à essayer la branche V2V et donc, dans le cas de cette simulation, à choisir une branche qui offre moins de rewards. D'où la croissance de son cumule des regrets. On peut d'ailleurs faire le lien entre l'allure de la courbe associée au paramètre epsilon et la courbe de cumule des regrets obtenus :

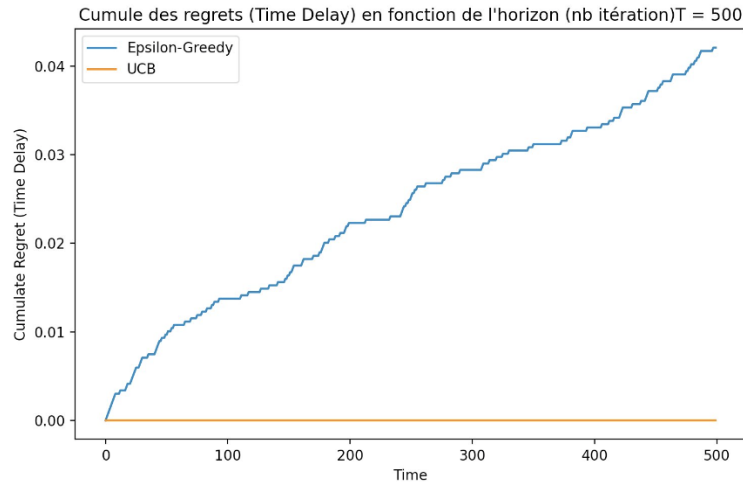


FIGURE 5 – Cumule des regrets sur des distances variant entre 1100m et 1400m

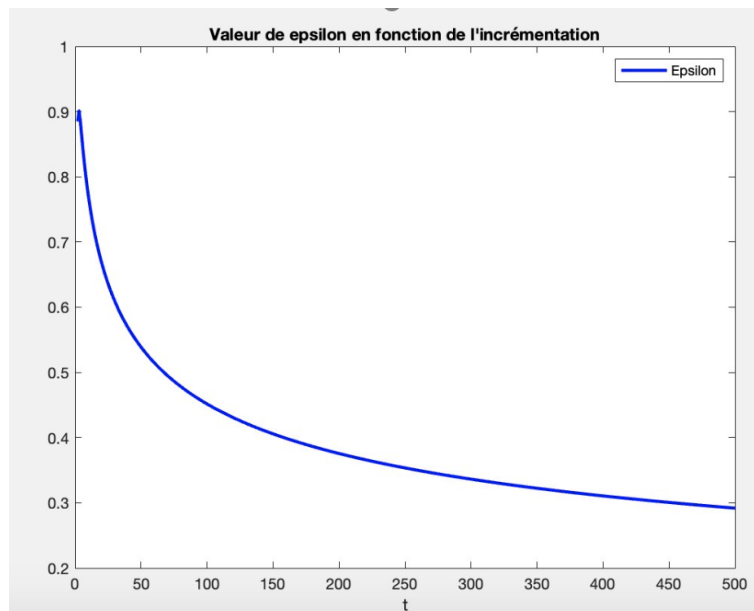


FIGURE 6 – Évolution du paramètre epsilon en fonction des itérations

4.2.2 Seconde simulation - Variation de la charge sur le réseau 5g

Dans cette simulation nous gardons le contexte d'une variation uniforme de distance entre 1100m et 1400m. Nous allons cette fois augmenter considérablement la charge sur le réseau 5g, ainsi la voie V2I offrira une latence plus importante que la voie V2V, voyons si nos algorithmes adaptent leurs choix :

On constate, logiquement, qu'à forte charge sur le réseau 5g (infrastructure congestionnée) les deux algorithmes préféreront la branche V2V pour envoyer leurs paquets.

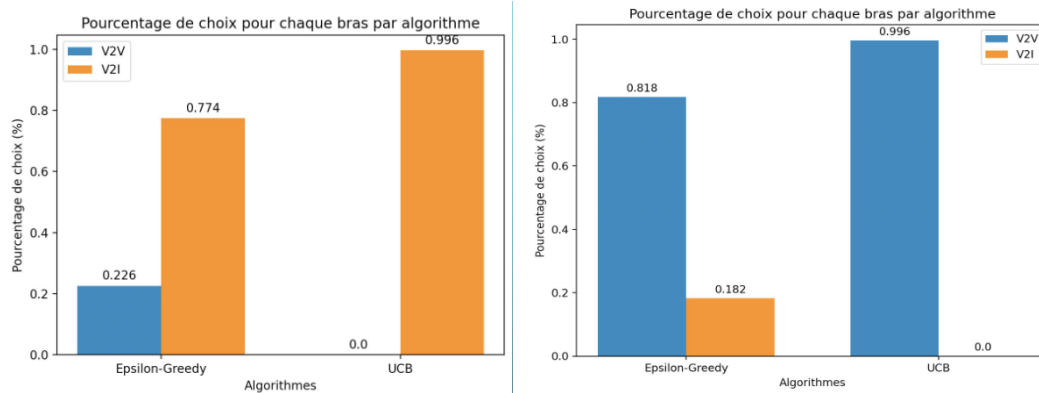


FIGURE 7 – À gauche : répartition des choix à faible charge, À droite : répartition des choix à forte charge

4.2.3 Troisième simulation - Adaptabilité des algorithmes

L'intérêt principal des algorithmes MAB est leur capacité d'adaptation lors de variations importantes du contexte. Pour mesurer la pertinence des algorithmes il est aussi très important de mesurer leur vitesse d'adaptation.

Dans cette simulation nous allons donc modéliser une situation où l'algorithme est face à 100 itérations à courtes distances (200m - 800m) puis passe à 400 itérations longues distance (1100m - 1400m). Voici ci dessous les courbes associées aux cumules des regrets :

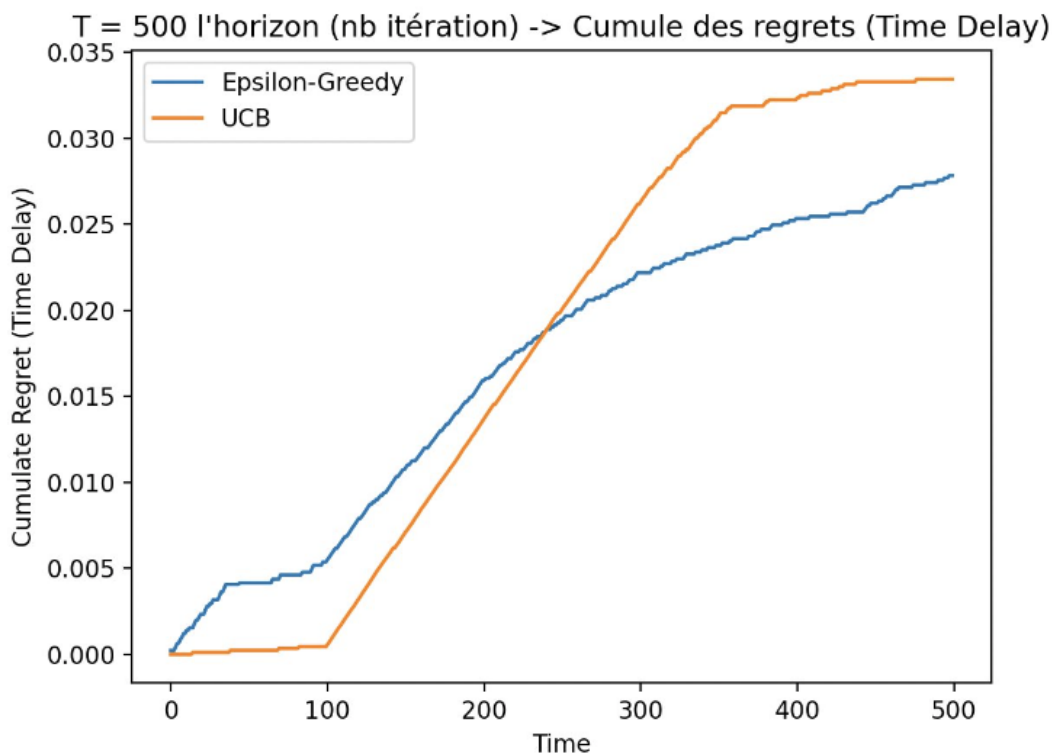


FIGURE 8 – Cumule des regrets pour les deux algorithmes dans une cas nécessitant l'adaptation

On constate que UCB possède une régime transitoire bien plus long qu'epsilon greedy, cependant, une fois la situation maîtrisée de nouveau par l'algorithme il retrouve des performances élevées.

5 Difficultés rencontrées et axes d'amélioration

Lors de notre étude sur l'utilisation d'algorithmes MAB dans les choix d'utilisation V2I ou V2V pour la sécurité routière, nous avons rencontré des difficultés à adapter nos simulations aux attentes des deux algorithmes. En effet, notre objectif principal était de minimiser la latence de transmission pour garantir une communication rapide et efficace entre les véhicules et les infrastructures routières. Cependant, les algorithmes MAB sont à l'origine conçus pour maximiser les récompenses et non pour minimiser les coûts. Nous avons donc dû ajuster nos simulations pour tenir compte de cette différence et trouver un équilibre entre la minimisation de la latence et la maximisation des récompenses.

D'autre part, pour préciser notre étude, nous aurions dû ajouter d'autres métriques de qualité de service (QoS) telles que le taux de perte de paquets, directement dans les algorithmes MAB. Cela aurait permis de prendre en compte plusieurs facteurs importants pour la sécurité routière (et l'état du réseau 5g...) et d'améliorer la performance globale des algorithmes. De plus, nous aurions dû approfondir notre étude sur la rapidité d'adaptation des algorithmes MAB. En effet, dans un cas réel, il est nécessaire d'avoir un algorithme qui réagit en quelques itérations pour garantir la sécurité de l'utilisateur. Une étude plus poussée sur ce sujet aurait permis de mieux comprendre les limites et les capacités des algorithmes MAB dans ce contexte.

6 Conclusion

L'utilisation d'algorithmes MAB dans les choix d'utilisation V2I ou V2V peut améliorer la sécurité routière en optimisant la qualité de service des communications entre les véhicules et les infrastructures routières. En utilisant des métriques telles que le délai de transmission, la charge des réseaux ou encore le taux de perte des paquets, les algorithmes MAB peuvent sélectionner la méthode de communication la plus appropriée pour chaque situation, réduisant ainsi les temps de réponse et les risques d'accidents. Cependant, des recherches supplémentaires sont nécessaires et en cours pour affiner ces algorithmes et les intégrer dans les systèmes de transport intelligents.

7 Bibliographie :

Contrôle dynamique des communications dans un environnement v2v et v2i - Thiwiza Bellache-Sayah

https://theses.hal.science/tel-01806340v1/file/70869_BELLACHE2018_archivage.pdf

Amélioration des systèmes de communication V2X pour la perception coopérative : sécurité des usagers de la route. Zoghلامي, Chaima.

<http://thesesups.ups-tlse.fr/5828/>