Ordonnancement et ACM conjoint sur canal aléatoire basé sur un transformer entrainé par apprentissage profond par renforcement

Sylvain NÉRONDAT[1,2] Xavier LETURC[2] Christophe LE MARTRET[2] Philippe CIBLAT[1]

[1]LTCI, Telecom Paris, Institut Polytechnique de Paris

[2]Thales, 4 avenue des Louvresses, 92230 Gennevilliers , France

Résumé – Nous proposons une solution conjointe d'ordonnancement et de sélection du schéma de modulation et codage pour un système de communications sans fil sur un canal aléatoire. Les trafics de chaque lien sont "contraints en délai" ou "best effort". Les paquets sont stockés dans des files d'attente de taille finie, et les paquets arrivant alors que la file est pleine sont supprimés. Les pertes de paquets peuvent être aussi dues au canal ou à un dépassement de délai. L'allocation des ressources se fait par bloc sur une trame, chaque bloc étant dédié à un type de trafic pour un lien donné. La solution repose sur un réseau de neurones profond de type "transformer", combiné à une ramification d'action et entraîné par apprentissage par renforcement afin de minimiser la perte de paquets.

Abstract – We propose a joint scheduler and modulation and coding scheme selection for a wireless communication system in random fading channel. Each link carries both delay-constraint and best-effort flows, with packets stored in a finite length buffer. Packet loss can occur due to random channel errors, delay violation, or buffer overflow. The scheduling is performed on a frame with several resource blocks (RB), each RB carrying one type of traffic for one link. The proposed scheduler leverages an encoder-only transformer combined with action branching, trained with deep reinforcement learning, to minimize packet loss.

1 Introduction

On considère une station de base (SB) réalisant l'ordonnancement des communications de plusieurs liens sans fil sur une trame composée de plusieurs ressources élémentaires appelées "resource blocks" (RB). La SB alloue au début de chaque trame un ou plusieurs RB à chaque lien et sélectionne un schéma de modulation et de codage (SMC) pour la transmission. Chaque lien subit des évanouissements aléatoires, indépendant entre chaque RB, caractérisés par un rapport signal à bruit (RSB) moyen supposé connu de la SB. Chaque réalisation du canal induit une probabilité d'erreur paquet qui dépend du RSB instantané, différente selon le SMC utilisé. Chaque lien supporte deux types de flux : un flux contraint en délai (CD) et un flux "best effort" (BE). Les paquets des différents flux sont stockés dans des files "first in first out" (FIFO) dédiées de tailles finies, et les paquets arrivant dans une file d'attente pleine sont supprimés, conduisant à des pertes par "buffer overflow" (BO). Aussi, les paquets des flux CD non transmis dans un temps imparti sont supprimés, conduisant à des pertes par violation de délai (VD). Des pertes de paquets peuvent ainsi survenir à cause du canal, d'un BO ou d'une VD.

L'ordonnancement est traditionnellement réalisé à partir d'heuristiques telles que le round-robin (RR) ou l'algorithme modified largest weighted delay first (MLWDF) [1], qui réalisent uniquement l'allocation des RB sans la sélection du SMC, qui doit donc être réalisée par une autre méthode.

Plus récemment, des approches utilisant de l'apprentissage profond par renforcement (deep reinforcement learning (DRL) en anglais) ont été proposées pour l'ordonnancement. Le DRL permet, contrairement aux heuristiques, de formuler et de résoudre des problèmes d'optimisation avec des critères choisis par le concepteur du système. De plus, le problème d'ordonnancement pouvant se modéliser sous la forme d'un processus de Markov décisionnel, il peut être résolu de manière optimale dans le cas où les dimensions sont petites (en utilisant par exemple l'algorithme *Value Iteration*), et de manière approchée par le DRL pour les plus grandes dimensions. Un des avantages de ces approches est aussi leur flexibilité pour intégrer facilement toute information utile en entrée du réseau de neurones, et de concevoir des récompenses adaptées au problème. Nous avons identifié dans [3] qu'une bonne architecture neuronale pour le problème de l'ordonnancement doit vérifier trois propriétés : i) être équivariante par permutation (EP), ii) être invariante au nombre de liens (INL), et iii) effectuer une gestion globale des files (GGF).

Les travaux antérieurs utilisant du DRL les plus pertinents que nous avons identifiés sont les suivants. Les auteurs de [4] proposent d'allouer des RB en fréquence à chaque lien en se basant sur les caractéristiques des files et du canal. L'architecture proposée est EP et INL, mais pas GGF, et ne s'étend pas facilement à l'allocation de SMC. Une architecture utilisant de la ramification d'action (RA) est proposée dans [5] pour allouer des SMC à différents utilisateurs sur différents RB. Les auteurs ne prennent pas en compte l'état des files. De plus, l'architecture est GGF, mais pas INL ni EP. Nous avons proposé dans [2] et [3] des architectures EP, INL et GGF, mais uniquement pour l'ordonnancement et donc sans allocation de SMC, et en considérant un canal de propagation sans erreur.

Notre contribution propose une solution conjointe d'ordonnancement et de sélection du SMC associée prenant à la fois en compte les pertes liées au canal de propagation et aux files d'attentes (pertes par BO et VD). La solution utilise un réseau de neurones de type transformer ayant les propriétés EP, GGF et INL, associé à de la RA et entraîné par DRL.

Le papier s'organise comme suit. La section 2 introduit le modèle du système considéré et le problème traité. La section 3

détaille la solution proposée. La section 4 fournit des résultats de simulations. Enfin, la section 5 conclut l'article.

2 Modèle du système et problème

On considère une SB servant n_L liens de communications, chaque lien supportant deux flux de données : un flux CD et un flux BE. Les paquets de chaque flux de chaque lien sont stockés dans une file FIFO de taille B finie, résultant en un nombre total de $n_Q = 2n_L$ files. On suppose pour simplifier l'exposé que tous les paquets ont le même nombre de bits. La bande totale W est divisée en N_f RB, qui sont alloués par la SB à chaque trame. Chaque lien est en outre sujet à un canal de propagation à évanouissements aléatoires, dont le modèle sera détaillé plus loin dans la section. Notons $\ell \in \{0, \ldots, n_L - 1\}$ l'indice d'un lien, et $t \in \{0, 1\}$ l'indice de chaque flux, où $t = 0$ et $t = 1$ correspondent respectivement au flux CD et au flux BE. Une file associée au $t$ème flux du ℓème lien est numérotée $i = 2\ell + t$. Par conséquent, $\ell_i = \lfloor i/2 \rfloor$ et $t_i = i \mod 2$ où ℓ_i et t_i sont le lien et le flux associés à la file i. Nous utiliserons dans la suite de manière équivalente l'indice i ou la paire (ℓ_i, t_i) pour la file i.

Chaque paquet arrivant dans une file a un temps d'attente (TA) fixé à 0 qui est incrémenté de 1 à la fin de chaque trame. Soit $d_{u,i} \in \{-1, 0, \ldots, D\}$ le TA du $u$ème paquet dans la $i$ème file avec $u \in \{0, \ldots, B-1\}$ et où, par convention, -1 représente une entrée vide. D est finie pour les files CD tandis qu'on fixe $D = +\infty$ pour les files BE.

On note $n_{i,k,j}$ le nombre de paquets dans la file i de la trame k, avant l'allocation du $j$ème RB. Ainsi, au début de chaque trame, la file i contient $n_{i,k,1}$ paquets. La SB sélectionne pour chacun des N_f RB une file à servir ainsi qu'un SMC associé parmi $\mathcal{M}$ disponibles. Chaque SMC permet d'extraire un nombre entier de paquets distincts, avec une capacité de protection contre les erreurs dues au canal qui varie d'un SMC à l'autre. Notons i_k^j, $j = 1, \ldots, N_f$, la file sélectionnée pour le $j$ème RB de la trame k, et $m_k^j \in \{1, \ldots, \mathcal{M}\}$ le SMC associé. Il est possible de servir plusieurs fois la même file au sein d'une même trame, et de sélectionner différents SMC pour une file sélectionnée plusieurs fois durant une même trame. Notons p_k^j le nombre de paquets que le SMC m_k^j permet d'extraire de la file sélectionnée pour le $j$ème RB. Les RB sont remplis de manière séquentielle et donc le nombre de paquets qu'il est possible d'extraire d'une file i donnée dépend i) de $n_{i,k,1}$, et ii) du nombre de paquets extrait de cette file durant les précédents RB de la trame. Ainsi, les $n_{i,k,j}^t := \min(p_k^j, n_{i,k,j})$ paquets les plus anciens sont extraits de la file, et le nombre de paquets restant est mis à jour à chaque RB via la relation $n_{i,k,j+1} = n_{i,k,j} - n_{i,k,j}^t$. Les bits des $n_{i,k,j}^t$ paquets extraits constituent un paquet d'information (PI). Les bits du PI sont entrelacés et encodés conjointement, modulés puis transmis sur le canal. Le taux de codage et la modulation utilisée dépendent du SMC. Le canal est susceptible de créer des erreurs dans le PI après décodage. Si le PI décodé contient des erreurs, le nombre de paquets perdus à cause du canal, noté $n_{i,k,j}^c$, est égal à $n_{i,k,j}^t$, et 0 sinon. Notons $n_{2\ell,k}^d$ le nombre de paquets CD du lien ℓ dont le TA est égal à D en fin de trame. Une VD se produit pour ces paquets, entraînant leur perte. Comme $D = +\infty$ pour les files BE, il résulte que $n_{2\ell+1,k}^d = 0$. Le nombre d'entrées libres dans la file i à la fin de la trame est alors égal à $\kappa_{i,k} = B - n_{i,k,N_f+1} + n_{i,k}^d$, où n_{i,k,N_f+1} désigne le nombre de paquets restants dans la file une fois les N_f RB servis.

Le TA de chaque paquet restant est ensuite incrémenté de 1, et $n_{i,k}^r$ nouveaux paquets arrivent dans la file i, où $n_{i,k}^r$ suit une distribution de Poisson de paramètre $\lambda_i \geq 0$. Les $n_{i,k}^o = \max(0, n_{i,k}^r - \kappa_{i,k})$ éventuels paquets surnuméraires sont supprimés pour cause de BO. Ce processus est répété à chaque trame, et le nombre de paquets dans la file i au début de la trame $k+1$ est de $n_{i,k+1,1} = n_{i,k,N_f+1} - n_{i,k}^d + n_{i,k}^r - n_{i,k}^o$.

On suppose que le PI transmis sur chaque lien est sujet à un canal aléatoire de Rayleigh à évanouissement plat en fréquence et variant indépendamment entre chaque RB, et qu'il est corrompu par un bruit blanc gaussien centré de variance $2\sigma_n^2$ en réception. Ainsi, le canal du lien ℓ lors du $j$ème RB de la trame k est modélisé par un coefficient complexe $h_\ell(k,j) \sim \mathcal{CN}(0, 2\sigma_\ell^2)$, où $\mathcal{CN}(0, 2\sigma_\ell^2)$ représente une distribution gaussienne complexe centrée de variance $2\sigma_\ell^2$. Le RSB instantané associé s'écrit $\Gamma_\ell(k,j) = |h_\ell(k,j)|^2/(2\sigma_n^2)$ et le RSB moyen du lien ℓ est donné par $\bar{\Gamma}_\ell := \sigma_\ell^2/\sigma_n^2$.

Notons $q_m(\Gamma_\ell(k,j))$ la probabilité d'erreur du PI lorsque le SMC m est utilisé et que le RSB instantané est de $\Gamma_\ell(k,j)$. On peut déterminer $q_m(\Gamma_\ell(k,j))$ si $\Gamma_\ell(k,j)$ est connu, ce qui correspond à une voie de retour plus rapide que le temps de cohérence du canal. On suppose dans cet article que $\Gamma_\ell(k,j)$ n'est pas connu, mais que le RSB moyen de chaque lien est disponible au niveau de la SB, ce qui permet de déterminer la probabilité d'erreur moyenne associée au SMC m comme :

$$\bar{q}_m(\bar{\Gamma}_\ell) = \int_0^{+\infty} q_m(x) f_{\Gamma_\ell}(x) dx \qquad (1)$$

où $f_{\Gamma_\ell}(x)$ est la densité de probabilité du RSB du lien ℓ dont la formule analytique est connue dans le cas du canal de Rayleigh considéré, et $q_m(x)$ est la probabilité d'erreur évaluée au RSB x. On suppose que $\bar{q}_m(\bar{\Gamma}_\ell)$ est connue au niveau de la SB.

Dans le système décrit précédemment, trois types de pertes de paquets peuvent survenir : i) des pertes liées au canal de propagation, ii) des pertes liées à des VD pour les flux CD, et iii) des pertes causées par des BO. On propose dans la prochaine section une solution d'allocation de RB et de SMC minimisant le nombre total de paquets perdus.

3 Solution proposée

La solution se base sur un réseau de neurones profond entrainé par DRL. Nous proposons d'abord une formulation pour les trois ingrédients requis par les approches DRL : un espace d'états (EE), un espace d'actions (EA) et une récompense. Nous proposons ensuite une architecture neuronale adaptée.

3.1 Espace d'états

Nous considérons trois EE différents qui se distinguent par les caractéristiques associées aux flux CD : HoL, xHoL et APD (les détails sont fournis dans [3] et omis ici par manque de place). On note $\mathbf{f}_{\ell,k}^{\text{CD-}x}$ le vecteur comportant les informations spécifiques au trafic CD pour l'EE de type $x \in \{\text{HoL}, \text{xHoL}, \text{APD}\}$ à la trame k. L'EE relatif au lien ℓ pour la trame k s'écrit sous la forme générale $\mathbf{f}_{\ell,k} := [\mathbf{f}_{\ell,k}^{\text{CD-}x}, n_{(2\ell+1),k,1}, \bar{q}_1(\bar{\Gamma}_\ell), \ldots, \bar{q}_{\mathcal{M}}(\bar{\Gamma}_\ell)]^T$, où $n_{(2\ell+1),k,1}$ correspond au nombre de paquets BE du lien ℓ au début de la

trame k. La différence par rapport aux EE proposés dans [3] provient de l'ajout des probabilité d'erreur moyenne $\bar{q}_m(\bar{\Gamma}_\ell)$ de chaque SMC m. On note n_{fe} le nombre d'entrées de $\mathbf{f}_{\ell,k}$. On définit alors l'EE global incluant les informations de toutes les files par la matrice $n_{\text{fe}} \times n_L$ suivante : $\mathbf{s}_k = [\mathbf{f}_{1,k}, \dots, \mathbf{f}_{n_L,k}]$.

3.2 Espace d'actions

Pour chaque RB de la trame k, l'ordonnanceur choisit conjointement la file à servir et le SMC associé, qui est noté (i_k^j, m_k^j). Ainsi, l'action s'écrit $\boldsymbol{a}_k := \{(i_k^1, m_k^1), \dots, (i_k^{N_f}, m_k^{N_f})\}$. L'EA est ainsi de dimension $(2n_L\mathcal{M})^{N_f}$. Par exemple, pour $n_L = 6$, $\mathcal{M} = 3$ et $N_f = 5$, l'EA est de dimension 60.466.176, ce qui dépasse les capacités des architectures neuronales conventionnelles. Pour cette raison, nous incorporons dans l'architecture proposée en section 3.4 de la RA.

3.3 Récompense

Notre objectif étant de minimiser le nombre de paquets perdus, nous définissons la récompense globale r_k par :

$$r_k = e^{\omega(\alpha_1 g_{d,k} + \alpha_2 g_{o,k} + \alpha_3 g_{c,k})} \tag{2}$$

où $g_{d,k} := -\sum_{\ell=0}^{n_L-1} n_{2\ell,k}^d$, $g_{o,k} := -\sum_{i=0}^{n_Q-1} n_{i,k}^o$ et $g_{c,k} := -\sum_{j=1}^{N_f}\sum_{i=0}^{n_Q-1} n_{i,k,j}^c$ représentent respectivement l'opposé du nombre de paquets perdus par VD, par BO, et du fait du canal, pour la trame k, $\omega > 0$ est un hyperparamètre contrôlant le comportement de l'exponentiel, et $\alpha_1, \alpha_2, \alpha_3 > 0$ sont fixés tels que $\sum_{j=1}^{3} \alpha_j = 1$ et contrôlent l'importance relative des différentes pertes. Le choix de l'exponentielle assure que la récompense est à valeurs dans $]0, 1]$.

3.4 Architecture

Nous utilisons une version adaptée de l'architecture transformer encodeur uniquement (TEU)-RA proposée dans [2] et illustrée en figure 1, où les blocs en gris correspondent aux paramètres appris au cours de l'entrainement. Une projection affine de l'état $\mathbf{s}_k$ est d'abord réalisée à l'aide d'une matrice $\boldsymbol{W}_e$ de dimensions $d_e \times n_{\text{fe}}$ et d'un vecteur de biais $\boldsymbol{b}_e$ de dimension $d_e \times 1$ où d_e est une dimension à choisir par l'utilisateur. La matrice résultante s'écrit $\tilde{\mathbf{s}}_k := \boldsymbol{W}_e\mathbf{s}_k + \boldsymbol{b}_e\mathbf{1}_{n_L}$, où $\mathbf{1}_k$ désigne le vecteur $1 \times k$ dont chaque entrée vaut 1. Cette matrice de dimension $d_e \times n_L$ est ensuite traitée par le TEU, qui dispose des propriétés EP, INL et GGF. La sortie est une matrice $\tilde{\boldsymbol{o}}_E$ de dimensions $d_e \times n_L$. Une projection affine de $\tilde{\boldsymbol{o}}_E$ est réalisée sur chaque branche j à l'aide d'une matrice $2\mathcal{M} \times d_e$ notée $\boldsymbol{W}_u^j$ et d'un vecteur de biais $\boldsymbol{b}_u^j$ de dimension $2\mathcal{M} \times 1$. La matrice résultante $\mathbf{A}_j := \boldsymbol{W}_u^j\tilde{\boldsymbol{o}}_E + \boldsymbol{b}_u^j\mathbf{1}_{n_L}$ est de dimension $2\mathcal{M} \times n_L$. Sa ℓème colonne correspond à la fonction avantage associée au choix de chaque SMC pour les deux types de flux du lien ℓ. Une approche de dueling est aussi utilisée, qui consiste à estimer la fonction valeur associée à chaque lien via une autre projection affine de $\tilde{\boldsymbol{o}}_E$: $\boldsymbol{v} := \boldsymbol{g}\tilde{\boldsymbol{o}}_E + b\mathbf{1}_{d_e}$, où $\boldsymbol{g}$ est un vecteur $1 \times d_e$ et b est un scalaire.

Le vecteur $\boldsymbol{v}$ de dimension $1 \times n_L$ est ensuite additionné à chaque ligne de la matrice $\mathbf{A}_j$ sur chaque branche j, conduisant à une matrice contenant les estimations des Q-valeurs associées au choix de chaque SMC pour chaque lien. Cette matrice est mise sous la forme d'un vecteur de dimension $2\mathcal{M}n_L \times 1$, où l'entrée a_k^j correspond à une unique combinaison de file et de SMC, où la file est $i_k^j = \lfloor a_k^j/\mathcal{M} \rfloor$ et le SMC est $m_k^j = a_k^j \bmod \mathcal{M}$. L'action est sélectionnée sur chaque branche j via un argmax, et correspond à l'action à effectuer pour le $j$ème RB. Un masque adaptatif dont le fonctionnement est détaillé dans [2] est utilisé pour éviter de sélectionner des files vides. L'architecture est entrainée avec l'algorithme bien connu du deep Q-Learning.

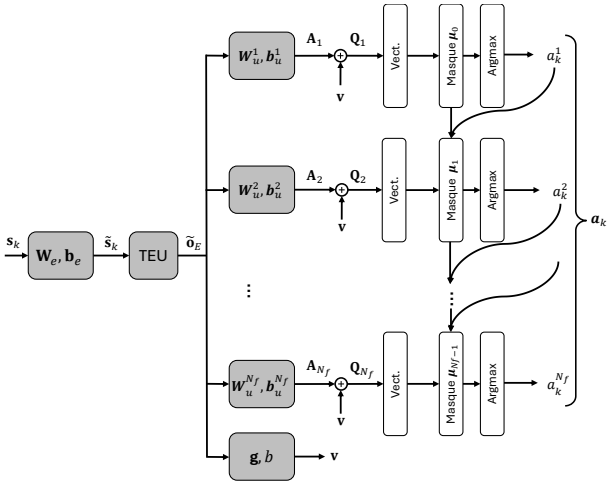

FIGURE 1 : Architecture proposée.

4 Résultats de simulations

Nous comparons dans cette section les performances de la solution TEU-RA proposée avec deux heuristiques de la littérature opérant à SMC fixe : le RR et le MLWDF [1]. Nous avons implémenté pour ces deux heuristiques une règle statique qui consiste à choisir le SMC permettant d'extraire le maximum de paquets tout en assurant une probabilité d'erreur moyenne liée au canal inférieure à un seuil fixé arbitrairement à 10^{-2}.

On fixe le nombre de liens à $n_L = 6$. La taille des files est fixée à $B = 40$, le délai maximum des flux CD à $D = 20$, et le nombre de RB par trame à $N_f = 5$. On suppose que les paquets ont tous une taille de 1000 bits, et les caractéristiques des $\mathcal{M} = 3$ SMC implémentés sont détaillés dans la table 1, où nous utilisons un code correcteur low density parity check (LDPC) quasi-cyclique.

TABLE 1 : SMC disponibles.

Modulation	QPSK	16QAM	64QAM
Taux de codage	1/2	1/2	2/3
Nb de bits encodés	1000	2000	3000
Nb de paquets extraits	1	2	3

On suppose que toutes les files ont un taux d'arrivée identique λ, et on définit le *taux d'arrivée global* comme $\Lambda := \lambda n_Q/N_f$. Nous avons fixé pour l'entrainement $\Lambda = 1{,}6$ (représenté par une barre verticale sur les figures). Les architectures neuronales sont entraînées sur 4000 épisodes de 7000 trames. Les RSB moyens de chaque lien sont tirés aléatoirement au début de chaque épisode suivant une loi uniforme entre 20 et 40 dB. Nous avons fixé $\omega = 4{,}6$, $\alpha_1 = \alpha_2 = 0{,}1$ et $\alpha_3 = 0{,}8$ dans (2).

Après entrainement, nous évaluons la capacité de généralisation des différentes méthodes en fonction de Λ. L'évaluation pour chaque valeur de Λ est réalisée sur 25 épisodes de

10^6 trames. Comme pour l'entrainement, les RSB moyens de chaque lien sont tirés aléatoirement au début de chaque épisode suivant une loi uniforme entre 20 et 40 dB. Dans les figures, la solution proposée correspond aux courbes "HoL", "xHoL" et "APD", qui correspondent aux trois EE évoqués en section 3.1.

La figure 2 représente le TPP total (TPPT), défini comme le rapport entre le nombre total de paquets arrivant dans les files et l'ensemble des épisodes et le nombre total de paquets perdus, quelle que soit la cause de la perte, en fonction de Λ. On observe pour $\Lambda \geq 1$ que les TPPT des trois EE de la solution proposée sont plus faibles que ceux des heuristiques. Par exemple on observe pour $\Lambda = 1{,}5$ un gain d'un ordre de grandeur des solutions proposées par rapport aux heuristiques. On peut observer que l'EE APD conduit au plus faible TPPT, suivi par HoL, et enfin xHoL. Quand $\Lambda < 1$, les TPPT de "APD" et "xHoL" sont plus élevés que ceux des heuristiques, qui ont approximativement le même TPPT. Le TPPT dans ce régime de Λ est principalement dû aux erreurs liées au canal, comme nous le verrons dans la suite.

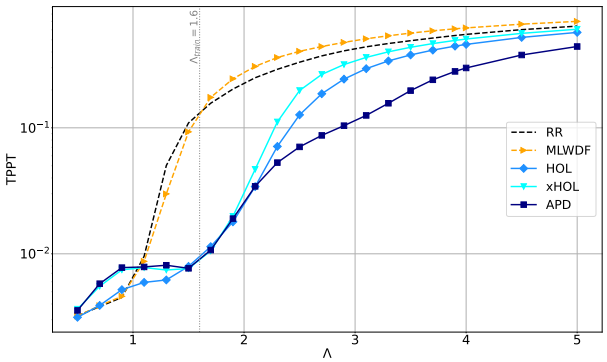

FIGURE 2 : TPPT vs. Λ.

La figure 3 représente le TPP canal (TPPC), défini comme le rapport entre le nombre total de paquets arrivant dans les files et le nombre total de pertes liées au canal, en fonction de Λ. On constate que le TPPC de la solution proposée est plus élevé que celui des heuristiques. Pour $\Lambda \geq 1{,}2$, le TPPC des heuristiques est constant avec une valeur inférieure au seuil de 10^{-2} cible. Le TPPC de la solution proposée augmente avec Λ, ce qui s'explique par le fait qu'en régime saturé, elle choisit des SMC moins robustes pour vider plus efficacement les files, conduisant à un TPPC plus élevé, ce qui est la contrepartie pour obtenir un TPPT plus faible comme vu en figure 2. On peut également voir que la relation d'ordre entre les EE est inversée par rapport au TPPT.

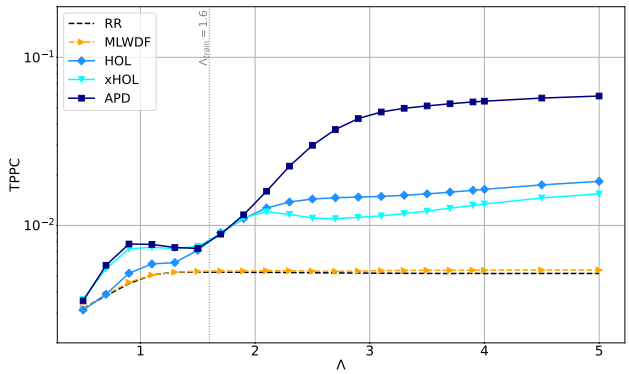

FIGURE 3 : TPPC vs. Λ.

Enfin, la figure 4 représente le throughput en fonction de Λ, défini comme le nombre de paquets transmis correctement par trame. On peut observer que toutes les méthodes obtiennent un throughput équivalent pour $\Lambda \leq 1{,}2$. Pour $1{,}2 \leq \Lambda \leq 2{,}2$, les performances des différents EE sont équivalentes et offrent un meilleur throughput que les heuristiques. Pour $\Lambda > 2{,}2$, les performances des EE se distinguent : le xHoL est le moins bon et sature à 10 paquets par trame, suivi par le HoL qui sature à 11 paquets par trame et enfin l'APD qui sature à 14 paquets par trame à partir de $\Lambda = 3{,}2$.

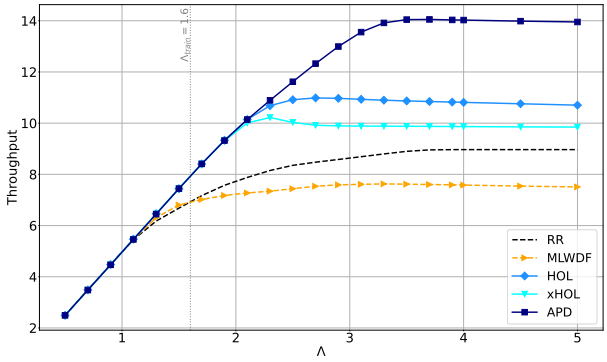

FIGURE 4 : Throughput vs. Λ.

5 Conclusion

Nous avons proposé une solution d'ordonnancement pour un système de communications sans fil lorsque les liens sont sujets à des canaux aléatoires. Nous avons proposé une architecture TEU-RA pour la sélection conjointe de file et de SMC pour l'ensemble RB de la trame. Nous avons montré par simulations que la solution proposée surpasse les heuristiques de l'état de l'art en termes de taux d'erreur et de throughput.

Références

[1] F. CAPOZZI, G. PIRO, L.A. GRIECO, G. BOGGIA et P. CAMARDA : Downlink packet scheduling in LTE cellular networks : Key design issues and a survey. *IEEE Commun. Surveys Tuts.*, 15(2):678–700, 2013.

[2] Sylvain NÉRONDAT, Xavier LETURC, Philipe CIBLAT et Christophe LE MARTRET : Efficient 5G resource block scheduling using action branching and transformer networks. *In IEEE Inter. Conference on Machine Learning for Communications and Networking (ICMLCN)*, 2025.

[3] Sylvain NÉRONDAT, Xavier LETURC, Christophe J. LE MARTRET et Philippe CIBLAT : Transformer-based packet scheduling under strict delay and buffer constraints. *In IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2025.

[4] Aurora PAZ-PÉREZ, Anxo TATO, J. Joaquín ESCUDERO-GARZÁS et Felipe GÓMEZ-CUBA : Flexible reinforcement learning scheduler for 5G networks. *In IEEE ICMLCN*, pages 566–572, 2024.

[5] Xiaowen YE et Liqun FU : Joint MCS Adaptation and RB Allocation in Cellular Networks Based on Deep Reinforcement Learning With Stable Matching. *IEEE Trans. on Mobile Computing*, 23(1):549–565, 2024.