Ibrahim CAMARA

Théo JULIEN

Christie ESSIANE-TSOGO

M. Rida KATHOUN

OPTIMISATION DE LA QoS SUR LES RESEAUX 5G EN UTILISANT L'IA

Compte rendu des travaux

1. Choix, dimensionnement et modèle

- Un individu avec son smartphone envoie pour simplifier, un paquet sur le réseau modélisé ici par une liste.
- La population modélisée par une matrice (liste de liste) est un ensemble de paquet envoyé sur le réseau.
- Un paquet est une ligne de la matrice (1er bit = 1 élément de la liste = 1, 2, 3 respectivement priorité haute à priorité basse, 2eme bit = élément de la liste représentant un slice = 1, 2ou 3, et 4 bits représentant le message à traiter. 1 bit à ajouter à la fin = une réponse de traitement = (1, requête traitée)

Deux uses cases :

- Le réseau est chargé à 30% ou moins
    - < Ou = 30% des paquets sur le réseau
- Le réseau est chargé à plus de 30%
    - Plus de 30% de la population donc des paquets arrive sur le réseau

Trois algorithmes d'ordonnancement pour traiter la file d'attente à tester les uns à la suite des autres

- **FIFO**
    - Fonction modélisant FIFO
    - Fonction de traitement de chaque paquets (algorithme de trie ou calcul de la somme des éléments de la liste)
    - Ajout du bit de réponse du traitement et le mettre à (1, requête traitée)
    - On stocke dans une liste le temps de traitement de chaque paquet pour ensuite faire la moyenne de traitement et avoir des statistiques en fonction de chaque use case
- **PQ**
    - Fonction modélisant PQ
    - On exécute les mêmes actions que sur FIFO sur les ordres de priorité

- **WFQ**

    - Fonction modélisant PQ
    - On exécute les mêmes actions que sur FIFO sur les ordres de priorité

On fait la comparaison de chaque algo dans un tableau récapitulatif avec temps moyen de traitement de chaque paquet et population

LES EXIGENCES :

- Faible latence (modélisation en cours de réflexion)
- Traitement de massif de paquets (nombre important de paquet),
- Haut débit (modélisé par le faible temps de traitement)

Un paquet est une liste (ex : [1 bit de priorité, 1 bit de slice, 4 bits de message] et 1 bit de traitement réalisé ajouté après le traitement du paquet). Par manque de Dataset (non trouvé), ces paquets ont été généré aléatoirement pour réaliser une matrice de 100 millions de paquets représentant la population de paquet à traiter.

Les données modélisant le nombre de paquet traité par jour sur des dates et des heures antérieures ont aussi été généré aléatoirement.

Choix du langage :

Dans un souci de maitrise et modélisation rapide, nous avons choisi d'utiliser le **langage python** parce qu'il donne un accès à des librairies assez varié et permettant de facilement faire de l'IA.

Récap :

- Le Dataset pour les tests d'ordonnancement est une base de données SQL comportant 100 millions de lignes représentant chaque liste explicitée ci-dessus
- Le Dataset pour les prédictions du nombre de paquet traité sur les réseaux à un instant donnée est une base de données comportant (le nombre de paquet traité, le jour de la semaine, l'heure)

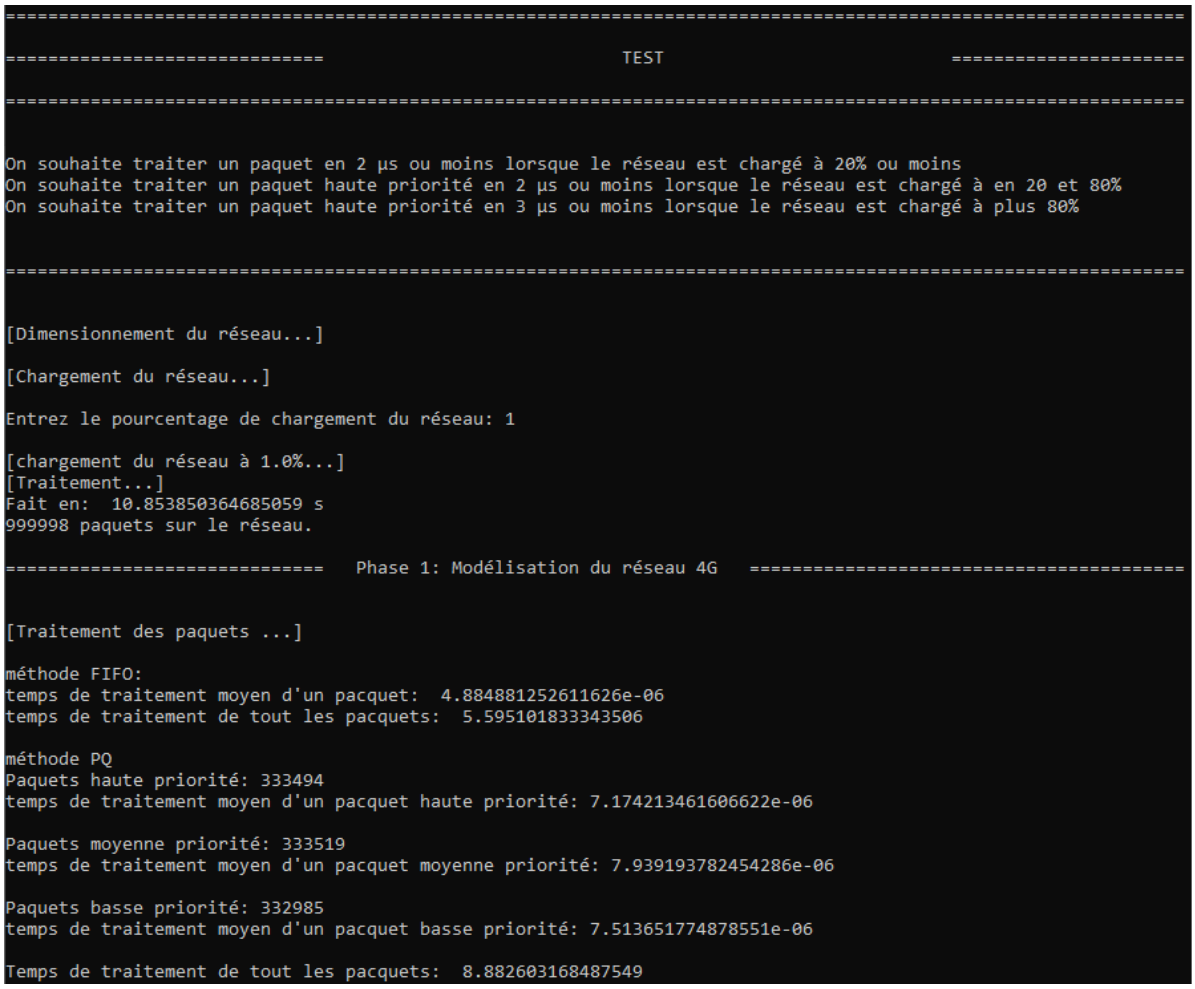
Réseau dans la phase 1 sera constituée comme suit :

- Des nœuds
    - Une fonction de chargement des paquets
    - Une fonction de gestion des paquets
    - Une fonction de traitement du paquet
- Des liens modélisés par le traitement subit par les paquets après chaque nœud

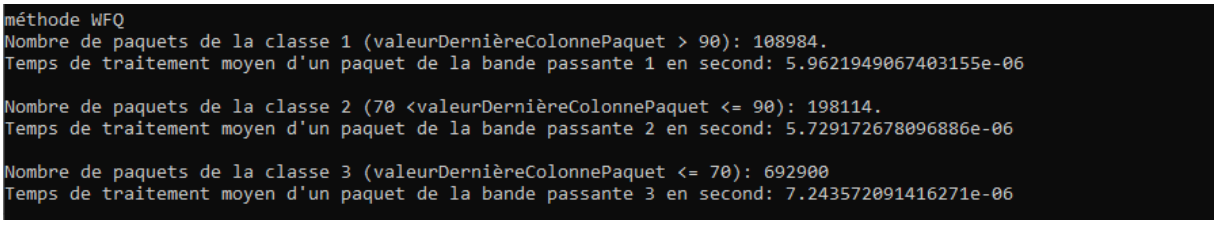
2. Phase 1 : Modélisation de la 4G

- On charge le réseau avec un pourcentage de paquet demandé en input à l'utilisateur
- On exécute chaque algorithme d'ordonnancement et de traitement des paquets
- On affiche les résultats

Ci-après les résultats des tests :



-



Réseau dans la phase 2 sera constituée comme suit :

- Des nœuds
  - Une fonction de chargement des paquets
  - Une fonction de répartition par slice
  - Une fonction de gestion des paquets
  - Une fonction de traitement du paquet
- Des liens modélisés par le traitement subit par les paquets après chaque nœud

3. Phase 2 : Modélisation de la 5G

On applique ensuite l'algorithme de la phase 1 à chaque slice en fonction de leur population respective :

```
[Phase 2: Modélisation du réseau 5G...]

[recupération des paramètres du dimensionnement précédent...]

[chargement du réseau à 1.0%...]
999998 paquets sur le réseau.

Fait en:  0.015620231628417969 s

[Traitement des paquets ...]

=============================================================================================================


traitement slice service 1: 333773 utilisateurs

méthode FIFO:
temps de traitement moyen d'un pacquet:  5.569470665448743e-06
temps de traitement de tout les pacquets:  2.1099305152893066

méthode PQ
Paquets haute priorité: 111343
temps de traitement moyen d'un pacquet haute priorité: 8.065793270636199e-06

Paquets moyenne priorité: 111085
temps de traitement moyen d'un pacquet moyenne priorité: 7.032490647997592e-06

Paquets basse priorité: 111345
temps de traitement moyen d'un pacquet basse priorité: 9.688847558211133e-06

Temps de traitement de tout les pacquets:  3.3913791179656982

méthode WFQ
Nombre de paquets de la classe 1 (valeurDernièreColonnePaquet > 90): 36667.
Temps de traitement moyen d'un paquet de la bande passante 1 en second: 6.006227424050007e-06

Nombre de paquets de la classe 2 (70 <valeurDernièreColonnePaquet <= 90): 66404.
Temps de traitement moyen d'un paquet de la bande passante 2 en second: 4.941053070524257e-06

Nombre de paquets de la classe 3 (valeurDernièreColonnePaquet <= 70): 230702
Temps de traitement moyen d'un paquet de la bande passante 3 en second: 5.426515868473397e-06

Temps de traitement de tout les pacquets:  2.40229868888855

*************************************************************************************************************
```

```
traitement slice service 2: 332992 utilisateurs

méthode FIFO:
temps de traitement moyen d'un pacquet:  5.87392719392705e-06
temps de traitement de tout les pacquets:  2.2029058933258057

méthode PQ
Paquets haute priorité: 111321
temps de traitement moyen d'un pacquet haute priorité: 1.2425432871618257e-05

Paquets moyenne priorité: 110998
temps de traitement moyen d'un pacquet moyenne priorité: 8.143784366243991e-06

Paquets basse priorité: 110673
temps de traitement moyen d'un pacquet basse priorité: 1.2110021409418557e-05

Temps de traitement de tout les pacquets:  4.124434947967529

méthode WFQ
Nombre de paquets de la classe 1 (valeurDernièreColonnePaquet > 90): 36156.
Temps de traitement moyen d'un paquet de la bande passante 1 en second: 7.384032297878221e-06

Nombre de paquets de la classe 2 (70 <valeurDernièreColonnePaquet <= 90): 66000.
Temps de traitement moyen d'un paquet de la bande passante 2 en second: 5.902424003138687e-06

Nombre de paquets de la classe 3 (valeurDernièreColonnePaquet <= 70): 230836
Temps de traitement moyen d'un paquet de la bande passante 3 en second: 7.417233022613703e-06

Temps de traitement de tout les pacquets:  2.8864693641662598
```

```
traitement slice service 3: 333233 utilisateurs

méthode FIFO:
temps de traitement moyen d'un pacquet:  7.490042226251361e-06
temps de traitement de tout les pacquets:  2.6716127395629883

méthode PQ
Paquets haute priorité: 110830
temps de traitement moyen d'un pacquet haute priorité: 1.181669208554491e-05

Paquets moyenne priorité: 111436
temps de traitement moyen d'un pacquet moyenne priorité: 6.888563076093091e-06

Paquets basse priorité: 110967
temps de traitement moyen d'un pacquet basse priorité: 1.196069439040585e-05

Temps de traitement de tout les pacquets:  3.7771499156951904

méthode WFQ
Nombre de paquets de la classe 1 (valeurDernièreColonnePaquet > 90): 36161.
Temps de traitement moyen d'un paquet de la bande passante 1 en second: 7.938776279701694e-06

Nombre de paquets de la classe 2 (70 <valeurDernièreColonnePaquet <= 90): 65710.
Temps de traitement moyen d'un paquet de la bande passante 2 en second: 5.759726265869698e-06

Nombre de paquets de la classe 3 (valeurDernièreColonnePaquet <= 70): 231362
Temps de traitement moyen d'un paquet de la bande passante 3 en second: 7.699139812895961e-06

Temps de traitement de tout les pacquets:  2.800132989883423

**********


==========


==========                    FIN DES TESTS                    ==========


==========
```

4. Analyse des résultats

```
['Analyse des résultats...']
Les tests on été réalisés pour avec un nombre de paquets de 10, 20, et 30, 40, 50, 60, 70, et 80 millions.

[Analyse 4G...]

Pour un nombre de paquet inférieur ou égale à 30% des paquet: l'algorithme FIFO prédomine au niveau du temps de
 traitement d'un paquet et celui du traitement de tous les paquets.
Pour les tests sur 10, 20, 30 millions de paquets:
Synthèse méthode FIFO:
Temps de traitement moyen d'un paquet: 2.4 µs
Temps de traitement total moyen des paquets: 52.4 s

Si on souhaite faire de la segmentation l'algorithme PQ est le plus adapté.
Synthèse méthode PQ:
Temps de traitement moyen d'un paquet haute priorité: 1.4 µs
Temps de traitement moyen d'un paquet moyenne priorité: 3.2 µs
Temps de traitement moyen d'un paquet basse priorité: 5.1 µs
Temps de traitement total des paquets: 238.5 s
Pour un nombre de paquet supérieur à 30% même si son temps de traitement total est largement supérieur à celui
de FIFO, le temps de traitement de chaque paquet avec PQ est 10 fois inférieur a celui de FIFO
C'est le bon compromis entre segmentation et traitement rapide de chaque paquet
Pour les tests sur 40, 50, 60, 70 et 80 millions de paquets:

Synthèse méthode PQ:
Temps de traitement moyen d'un paquet haute priorité: 0.18 µs
Temps de traitement moyen d'un paquet moyenne priorité: 0.23 µs
Temps de traitement moyen d'un paquet basse priorité: 1.5 µs
Temps de traitement total des paquets: 1260 s

Si vous souhaitez traiter tous les paquets le plus rapidement possible, l'algorithme FIFO est le plus adapté.
Synthèse méthode FIFO:
Temps de traitement moyen d'un paquet: 3.1 µs
Temps de traitement total moyen des paquets: 197 s

[Analyse 5G...]

Les résultats sont pareils que sur la 4G. Ils dependent de la population.
Seulement ici le temps de traitement de tous les paquets est ultra court du faite du principe de slicing.

C:\Users\admin-14697\Documents\FILE ROUGE>
```

5. <u>Phase 3 :</u> Prédiction du nombre de paquet sur le réseau par l'IA et sélection de l'algorithme approprié à appliquer à un instant donné

Il s'agit ici de faire une prédiction du nombre d'individus voir par slice en fonction des jours de la semaine, des heures, des nombres d'utilisateur aux mêmes dates antérieures. Ensuite en fonction de nos analyses sélectionner l'algo adapté.

Pour se faire à partir de notre dataset, nous transformé les jours de la semaine en valeurs continues comprise en 1 et 7. Ensuite pour que les prédictions soit interprétable et affichable nous avons normalisé les nombres d'utilisateurs qui estimés de base en millions ont été ramené à des valeurs comprises en -1 et 1. Enfin nous avons fait une régression linéaire multiple pour pouvoir prédire le nombre d'utilisateurs. Pour avoir un résultat se rapprochant un peu plus de la réalité, il faudra trouver un moyen d'inclure des données modélisant l'actualité.

Ci-après les résultats de notre algorithme :

```
Estimation du nombre de paquets sur le réseau par notre IA...

Entrez le jour de la semaine :LUNDI
Entrez maintenant l'heure d'affluence compris entre 0 et 24h (ex: 13h): 11h

Vous souhaitez estimer le nombre d'utilisateurs sur le réseau un lundi à 11h.

[Analyse en cours...]

Je prédis approximativement 37025392 sur le réseau un lundi à 11h avec une précision de 93%.

Fait en 18.92911982536316 seconds

[Analyse des résultats...]


Les tests on été réalisés pour avec un nombre de paquets de 40, 50, 60 et 70 millions.
Les résultats si après sont les valeurs moyennes des résultats de ces quatre jeux de données.

[Analyse 4G...]

37025392 utilisateurs prévus sur le réseau.
D'après nos analyses l'algorithme PQ serait le plus adapté pour traiter les files d'attentes

Synthèse méthode PQ:
Temps de traitement moyen d'un paquet haute priorité: 0.18 µs
Temps de traitement moyen d'un paquet moyenne priorité: 0.23 µs
Temps de traitement moyen d'un paquet basse priorité: 1.5 µs
Temps de traitement total des paquets: 1260 s

Si vous souhaitez traiter tous les paquets le plus rapidement possible, l'algorithme FIFO est le plus adapté.
Synthèse méthode FIFO:
Temps de traitement moyen d'un paquet: 3.1 µs
Temps de traitement total moyen des paquets: 197 s

[Analyse 5G...]

D'après nos analyses l'algorithme FIFO serait le plus adapté pour traiter les files d'attentes.

Synthèse méthode FIFO:
Temps de traitement moyen d'un paquet: 2.5 µs
Temps de traitement total moyen des paquets: 14 s

Ci après les tests avec les méthodes FIFO et PQ:

[Traitement...]
méthode FIFO:
temps de traitement moyen d'un pacquet:  6.531406405585011e-06
temps de traitement de tout les pacquets:  285.2736430168152
[Traitement...]
```

La suite des résultats ci-après :

```
méthode PQ
Paquets haute priorité: 12345152
temps de traitement moyen d'un pacquet haute priorité: 3.548132110084398e-05

Paquets moyenne priorité: 12346340
temps de traitement moyen d'un pacquet moyenne priorité: 3.7365881563708126e-05

Paquets basse priorité: 12333898
temps de traitement moyen d'un pacquet basse priorité: 1.0160481402974793e-05

Temps de traitement de tout les pacquets:  1182.5712957382202

Quel est donc votre choix?
Entrez 'FIFO' ou 'PQ': fifo
Application de de l'algorithme FIFO
Fait !
```

6. Conclusion partielle

Quand on regarde les résultats de du nombre de paquets prédit avec les simulations de charge du réseau associé à ce nombre de paquet on voit que les interprétations effectuées à la phase de test concordent bien.

Pour répondre aux exigences, nous voyons que sur la modélisation de la 5G les paquets repartis par slices sont traité dans leur globalité 3 fois plus vite que sur la modélisation de la 4G du fait de leur répartition par slices avant leurs traitements.

Les paquets sont traités dans l'ordre du µs et nous remarquons que plus le réseau est chargé plus les paquets sont traité rapidement sur PQ seulement les temps de traitement total des tous les paquets est très grand devant celui de FIFO. Pour un pool moyen de 55 millions on a un temps de traitement de 1260 seconds avec PQ contre 197 seconds sur avec FIFO mais un paquet est cependant traité 10 fois plus vite sur PQ que sur FIFO.

Conformément à l'exigence haut débit, nous voyons que PQ est le bon algorithme car il permet de traiter ultra vite un paquet et faire de la segmentation en fonction des QoS désirées par les utilisateurs d'un opérateur.

Quant à l'exigence 2 sur le massive IoT, nous avons essayé de la modéliser par le grand nombre de paquet qui est de 100 millions et pouvant être traité par le réseau dans la limite du processeur (il prend 30mn voir 45mn pour être exécuter et quelquefois il ne s'exécute pas en entier par manque de mémoire).

La modélisation de la faible latence est en cour de réflexion.

7. Phase 4 : Encore de l'IA pour faire de l'optimisation

La prochaine phase serait de pouvoir fixer un nombre de paquet et créer de façon automatique, des files d'attente imaginaire au fur et à mesure que le nombre de paquet augmente pour traiter les paquets de sorte à respecter un temps de traitement définit.