



Présentation orale

Khadija Akkar ; Salma Elkhatri ; Taha Missouri

Janvier 2023

Professeur encadrant : Meryem Ouederni

Table des matières

1	Introduction	2
2	Objectifs du projet	2
3	Problématique	2
4	Solution proposée	2
5	Algorithmes utilisés	2
6	Résultats obtenus	6
7	Conclusion	6

1 Introduction

Dans notre présentation nous allons vous montrer comment nos programmes facilitent la gestion et l'analyse des tables de routage. Les réseaux informatiques sont de plus en plus complexes, et il est devenu de plus en plus difficile pour les administrateurs réseau de gérer et d'analyser les tables de routage de manière efficace. Pour résoudre ce problème, nous avons développé un programme qui permet aux administrateurs réseau d'importer, et d'analyser les tables de routage de manière simple et efficace. Nous allons vous montrer comment cela fonctionne en détail dans la suite de notre présentation.

2 Objectifs du projet

L'objectif de ce projet est d'implanter, évaluer et comparer plusieurs manières de stocker et d'exploiter la table de routage d'un routeur.

3 Problématique

Comment améliorer les performances des réseaux informatiques en mettant en place un système de stockage et d'exploitation de tables de routage basé sur le cache, tout en garantissant la fiabilité et la sécurité des données?

4 Solution proposée

Notre solution consiste à mettre en place un système de stockage et d'exploitation de tables de routage basé sur le cache en utilisant une technologie de mémoire vive (RAM) pour stocker les tables de routage les plus fréquemment utilisées. Les données sont mises en cache dans la mémoire vive, permettant ainsi des accès rapides et réduisant les temps de latence pour les requêtes. En cas de besoin, les données peuvent également être récupérées à partir du stockage principal. Cette solution propose une utilisation de la mémoire vive pour stocker les tables de routage les plus fréquemment utilisées, permettant ainsi un accès rapide aux données.

5 Algorithmes utilisés

Algorithm 1 Définition de type pour une table de routage

```
1: TYPE T_TABLE EST POINTEUR SUR T_ROUTE_TABLE
2: TYPE T_ROUTE_TABLE EST ENREGISTREMENT
3:   Destination : T_ADRESSE_IP
4:   Masque : T_ADRESSE_IP
5:   Interface : CHAÎNE DE CARACTÈRES
6:   Suivant : T_TABLE
7: FIN ENREGISTREMENT
```

Algorithm 2 Remplissage de table de routage

```
1: repeat
2:   récupérer l'Adresse IP de cette ligne           IP : T_ADRESSE_IP
3:   Table: Destination ← IP
4:   récupérer le masque M de cette ligne           M : T_ADRESSE_IP
5:   Table: Masque ← M
6:   récupérer l'interface I de cette ligne I : CHAÎNE DE CARACTÈRE
7:   Table: Interface ← I
8:   Table: Suivant ← NULL
9:   Skip_Line;
10: until End_Of_File (Fichier)
```

Algorithm 3 Recherche d'interface pour une adresse IP donnée

```
1: curseur  $\leftarrow$  T
2: I_Trouvé  $\leftarrow$  Faux
3: while curseur  $\neq$  NULL ET non I_Trouvé do
4:   if (IP  $\wedge$  curseur.Masque) = curseur.Destination then
5:     I  $\leftarrow$  curseur.Interface
6:     I_Trouvé  $\leftarrow$  Vrai
7:   else
8:     Rien
9:   end if
10:  curseur  $\leftarrow$  curseur.Suivant
11: end while
```

Algorithm 4 Définition de type pour une table de cache de routage

```
1: TYPE T_ROUTE_CACHE EST ENREGISTREMENT
2:   Destination : T_ADRESSE_IP
3:   Masque : T_ADRESSE_IP
4:   Interface : CHAÎNE DE CARACTÈRES
5:   Suivant : T_CACHE
6:   Freq : ENTIER
7:   Time : DURATION
8: FIN ENREGISTREMENT
```

Algorithm 5 Chercher l'interface convenable

```
1: Créer un fichier résultat du nom "Résultats"          résultat : out
   File.Type
2: Ouvrir le fichier résultat du nom "Résultats" en mode écriture
3: curseur  $\leftarrow$  T
4: Ouvrir le fichier Paquet du nom "Paquet" en mode lecture   Paquet : in
   File.Type
5: if Taille = 0 then
6:   repeat
7:     Récupérer l'adresse IP de cette ligne                IP : in
       T_ADRESSE_IP
8:     Trouver l'interface de sortie dans la table T          T : in T_TABLE
       I : out chaîne de caractère
9:     Ajouter l'interface trouvé et l'adresse IP au fichier Résultats
10:  until Enf.Of.File (paquet)
11:  fermer le fichier Résultat
12:  fermer le fichier paquet
13: else
14:   Selon Politique Dans
15:     FIFO  $\Rightarrow$ 
16:       Donner le fichier Résultat en utilisant le cache avec la poli-
       tique FIFO
17:     LRU  $\Rightarrow$ 
18:       Donner le fichier Résultat en utilisant le cache avec la poli-
       tique LFU
19:     LFU  $\Rightarrow$ 
20:       Donner le fichier Résultat en utilisant le cache avec la poli-
       tique LRU
21:   Fin Selon
22:   Enregistrer le Cache dans un fichier text                cache.txt : out File_Type
23: end if
```

Algorithm 6 Définition de type pour une table de cache de routage

```
1: TYPE T_CACHE_Ptr EST POINTEUR SUR T_CACHE
2: TYPE T_CACHE EST ENREGISTREMENT
3:   DESTINATION : T_ADRESSE_IP
4:   MASQUE : T_ADRESSE_IP
5:   INTERFACE : CHAÎNE DE CARACTÈRES
6:   last_time_used : DURATION
7:   RIGHT : T_CACHE_Ptr
8:   LEFT : T_CACHE_Ptr
9: FIN ENREGISTREMENT
```

6 Résultats obtenus

-Présentation des résultats obtenus sur des exemples-

7 Conclusion

En conclusion, notre projet de stockage et d'exploitation de tables de routage basé sur le cache a permis d'améliorer les performances des réseaux informatiques en réduisant les temps de latence pour les requêtes et en accélérant les accès aux données. En utilisant une technologie de mémoire vive pour stocker les tables de routage les plus fréquemment utilisées . En somme, notre solution de stockage et d'exploitation de tables de routage basée sur le cache est une solution efficace pour améliorer les performances des réseaux informatiques modernes tout en garantissant la fiabilité et la sécurité des données.