

Raffinages

Types

T_AdresseIP is array [1..4] of integer;

T_Masque is array[1..4] of integer;

T_routes is record { Destination : **T_AdresseIP**; Masque : **T_Masque** ; Interface : **String** };

T_elt is record { route : **T_route** ; Suivant : **T_table** };

T_Table is access **T_elt**;

T_cache_LL is **T_Table**;

T_Noeud is record { Fg : **T_cache_LA** ; route : **T_route** ; date: **integer** ; Fd : **T_cache_LA** };

T_cache_La is access **T_Noeud**;

T_compteur is array [1..*Borne*] of integer;

Variables

Ligne de commande

nb_argument : *Integer*
Borne : *Integer*
politique : *String*
statistique : *booléen*
nom_fichier_routes : *string*
nom_fichier_paquets : *string*
nom_fichier_resultats : *string*

Structures

table : *T_table*
cache_LL : *T_cache_LL*
cache_LA : *T_cache_LA*
route : *T_routes*

Contrôle

fin : *string ou boolean*
date : *integer* -- date l'apparition d'une route dans cache_LA LRU
min_date : *Integer*
compteur : *T_compteur* -- compteur de fréquence pour les routes de cache_LL LFU
t_masque : *Integer* -- taille du masque, sert à choisir la route de masque le plus long

Statistique

dc : *integer* -- défaut de cache
nb_r : *integer* -- nombre de demande de route
-- politique est aussi une variable de statistique

Raffinage programme globale

R0 : Implémentation d'un routeur

R1 : Comment "implémenter un routeur"?

Récupérer les informations de la ligne de commande

nb_argument : **out**
Borne : **out**
politique : **out**
statistique : **out**
nom_fichier_routes : **out**
nom_fichier_paquets : **out**
nom_fichier_resultats : **out**

Créer une table de routage

table : **out**

Créer un cache

cache_L[LA] : **out**

Traiter les paquets

nom_fichier_routes : **in**
nom_fichier_paquets : **in**
nom_fichier_resultats : **in**
Borne : **in**
politique : **in**
statistique : **in**

R2: Comment "Récupérer les informations de la ligne de commande"?

nb_argument : *Integer* ← Argument_Count
Borne ← 0
politique ← FIFO
statistique ← True
nom_fichier_routes ← table.txt
nom_fichier_paquets ← paquets.txt
nom_fichier_resultats ← resultats.txt

Pour VarBoucle De 1 à nb_arg **Faire**
 Selon Argument(VarBoucle) **Dans**
 "-c" => Borne ← Argument(VarBoucle +1)

nb_arg : **in**

```

“-P” => politique ← Argument(VarBoucle +1)
“-S” => statistique ← False
“-t” => nom_fichier_routes ← Argument(VarBoucle +1)
“-p” => nom_fichier_paquets ← Argument(VarBoucle +1)
“-r” => nom_fichier_resultats ← Argument(VarBoucle +1)
Sinon => Rien

```

FinSelon

FinPour

R2: Comment “Créer une table de routage”?

Initialiser une table

table : **out**

Ajouter les éléments d’un fichier txt à la table

table : **out**

nom_fichier_routes : **in**

R2: Comment “Créer un cache” ?

cache : T_cache_L[LA]

Voir raffinage spécifique

cache : **out** T_cache_L[LA]

R2: Comment “traiter les paquets ?”

d:=0

Répéter

Extraire une ligne de paquet.txt dans un variable paquet

paquet : **out** String

Si est_commande(paquet) **alors**

exécuter la commande

paquet : **in**

SinonSi est_adresse_IP(paquet) **alors**

Trouver la route de sortie

paquet : **in**

table : **in**

cache : **in**

Mettre à jour le cache

cache : **in/out**

Ajouter l’interface de sortie au fichier resultat.txt

route : **in**

resultat.txt : **out**

d:= d+1

Sinon

Lever une exception

Tant que (non fin) et (non End_of_files(paquet.txt))

R3: Comment “Initialiser une table de routage” ?

Initialiser(table) -- fonction du module lca_table

table : **out**

R3: Comment “ajouter les éléments de nom_fichier_routes.txt à la table de routage”?

Extraire les lignes élément par élément de nom_fichier_route.txt dans table

nom_fichier_routes.txt : **in**
table : **out**

R3: Comment “Exécuter la commande”?

Selon paquet **Dans**

fin => fin := true

stat => afficher toutes les statistiques relatives au cache et à sa politique

cache : **in**

cache => afficher toutes les routes du cache

cache : **in**

table => afficher toutes les routes de la table de routage

table_de_routage : **in**

Fin du selon

R3: Comment “Trouver la route de sortie ”?

route : T_routes := Null;

nb_r := nb_r+1

chercher dans le cache

destination (-- anciennement paquet) : **in**

route : **out**

table : **in**

Si route=Null **Alors**

chercher dans la table;

destination : **in**

route : **out**

cache : **in**

dc := dc +1

Sinon

Null;

FinSi

R3: Comment “Mettre à jour le cache” ?

Voir raffinage spécifique au type de cache

cache : **in/out** ;

compteur : **in/out**

borne : **in**

date : **in**

min_date : **in**

route : **in/out**

politique : **in**

R3: Comment “Ajouter l’interface de sortie et la destination à resultat.txt” ?

Convertir destination en string

destination: **in/out**

Afficher(resultat, interface & destination)

R4 : Comment “chercher dans le cache” ?

Voir raffinage spécifique au type de cache

cache : **in**
destination : **in**
route : **out**

R4 : Comment “chercher dans la table” ?

```
t_masque := 0
route := null
Si est_vide(table) Alors
    Null
SinonSi (comp(Destination, table.all.route) et taille(table.all.route.masque)) >= t_masque Alors
    t_masque := taille(cache.all.route.masque)
    route := cache.all.route
    route_valide(Destiantion, table.all.suivant, compteur, res)
Sinon
    Null
FinSi
```

R4 : Comment “Afficher les routes de la table” ?

```
i : entier ← 0
Element : T_Element
Tant que (table_de_routage(i) <> Null) Faire
    Element ← table_de_routage(i)
    Afficher element
    i ← i+1
Fin Tant que
```

element : **in**

R4 : Comment “Afficher toutes les routes du cache” ?

Voir raffinage spécifique

R4 : Comment “Afficher les statistiques ” ?

```
Afficher(dc)
Afficher(dc/nb_r)
Afficher(politique)
```

Raffinage cache LL

R2 : Comment “creer un cache_LL”?

cache : **T_cache_LL**

Initialiser un cache

cache : **out**

R3 : Comment “Initialiser un cache_LL”?

Initialiser(cache) --fonction du module lca_table

R4 : Comment “rechercher une route dans un cache LL” ?

Si est_vide(cache) **Alors**

Null

SinonSi comp(Destination,cache.all.route) **et** taille(cache.all.route.masque)≥compteur **Alors**

compteur := taille(cache.all.route.masque)

res := cache.all.route

route_valide(Destiantion, cache.all.suivant, compteur,res)

Sinon

Null

FinSi

R4 : Comment “mettre à jour un cache LL” ?

Déterminer la bonne_route à ajouter au cache à partir de la route trouvée

route : **in/out**

Si appartient(cache_LL,route) **Alors**

Selon pol **dans**

“FIFO” => null

“LRU” => supprimer(cache_LL,route);

cache : **in/out**

route : **in**

enregistrer(cache_LL, route)

cache : **in/out**

route : **in**

“LFU”=> Déterminer la position de route dans cache_LL;

cache : **in**

route : **in**

i : **out**

compteur(i)=compteur(i)+1

compteur : **in/out**

i : **in**

FinSelon

compteur(i)=compteur(i)+1

SinonSi taille(cache_LL)<borne **Alors**

Selon pol **dans**

“FIFO” => enregistrer(cache_LL,route)

cache : **in/out**

route : **in**

“LRU” => enregistrer(cache_LL, route)

cache : **in/out**

<p>“LFU”=> enregistrer(cache_LL, route);</p> <p>Determiner la position i de route dans cache_LL;</p> <p>compteur(i)=compteur(i)+1</p> <p>FinSelon</p> <p>Sinon</p> <p>Selon pol dans</p> <p>“FIFO” => supprimer(cache_LL.all),</p> <p>enregistrer(cache_LL,route)</p> <p>“LRU” => supprimer(cache_LL.,route)</p> <p>enregistrer(cache_LL, route)</p> <p>“LFU”=> déterminer l’indice k du min de compteur;</p> <p>Remplacer le kième element de cache_LL par route;</p> <p>Compteur(k)=1</p> <p>Fin Selon</p> <p>FinSi</p>	<p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>i : out</p> <p>compteur : in/out</p> <p>i : in</p> <p>cache : in/out</p> <p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>cache : in/out</p> <p>route : in</p> <p>k : out</p> <p>cache : in/out</p> <p>route : in</p> <p>k : in</p> <p>cache : in/out</p> <p>compteur : in/out</p>
---	--

R4 : Comment “Afficher les routes d’un cache LL” ?

<p>i : entier ← 0</p> <p>route : T_route</p> <p>Si (cache <> Null) Alors</p> <p>route← cache.route</p> <p>Afficher route</p> <p>Afficher les routes de cache.all.suivant</p> <p>Sinon</p> <p>Null</p> <p>FinSi</p>	<p>route : in</p>
--	--------------------------

R5 : Comment “déterminer la bonne route à ajouter au cache à partir de la route trouvée” ?

```
    bon_masque : masque := route.masque
    Si table.all.route.adresseip masqué par route.masque = route.Adresseip et
    taille(table.all.route.masque)>taille route.masque Alors
        bon_masque := table.all.route.masque
    Sinon
        Null
    FinSi
    (-- faire de même avec table.all.suivant et la nouvelle valeur de bon_masque)
    bonne_route := {route.adresseip masqué par bon_masque ; bon_masque; route.interface }
```

R5 : Comment “Déterminer la position d’une route dans un cache_LL” ?

Modification simple de la fonction appartient (-- on ajoute un compteur) cache : **in**

R5 : Comment “Déterminer l’indice du min d’un tableau”?

```
Fonction Ind_min_tab(compteur : T_compteur) renvoie integer est
    k:integer :=0
    val:integer := compteur(borne)
    Pour i 1..(borne-1) faire
        Si compteur(i)<val Alors
            k:=i
            val:=compteur(i)
        Sinon
            Null
        FinSi
    Fin Pour
    Renvoie k
```

R5 : Comment “remplacer le kième élément de cache_LL par route”?

```
    Si k=0 Alors
        cache.all.route := route
    Sinon
        Remplacer(cache.all.suivant, k-1, route)
    FinSi
```


Raffinage cache LA

R2 : Comment “créer un cache_LA” ?

Initialiser un cache_LA

cache : **out**

R3 : Comment “initialiser un cache_LA” ?

Initialiser_Abr (-- procédure du module arbre_prefixe)

cache : **out**

R4 : Comment “rechercher dans un cache LA” ?

convertir destination en un entier écrit en base 2

destination : **in/out**

Si cache=NULL **alors**

Nul

SinonSi cache.all.Fg=NULL **et** cache.all.Fd=NULL **alors**

Renvoyer cache.all.route

SinonSi (destination **and** $2^{31} < 0$) **alors**

Rechercher destination*2 mod 2^{32} dans cache.all.Fd

Sinon

Rechercher destination*2 mod 2^{32} dans cache.all.Fg

FinSi

R4 : Comment “mettre à jour un cache LA” ?

Déterminer la bonne route à ajouter au cache

route : **in/out**

Si appartient(route, cache) **alors**

mettre à jour la date de route

route : **in/out**

cache : **in/out**

SinonSi taille(cache_LA) < Borne **alors**

Ajouter(cache_LA,route)

route : **in**

cache : **in/out**

Sinon

Trouver et supprimer la plus ancienne route

cache : **in/out**

Ajouter(cache_LA,route)

route : **in**

cache : **in/out**

FinSi

R4 : Comment “Afficher les routes d’un cache LA” ?

Arbre : T_arbre

Arbre ← cache.Arbre

Afficher les éléments des feuilles de l’arbre Arbre

Arbre : **in**

R5 : Comment “afficher les éléments des feuilles d’un arbre” ?

Procédure routes_cache_LA(cache : in T_cache_LA) **est**

Si (cache.All.Fd=NULL && cache.all.Fg=NULL) **Alors**

Afficher cache.all.route

Sinon

```

        routes_cache_LA(cache.all.Fg)
        routes_cache_LA(cache.all.Fd)
    FinSi

```

R5 : Comment “déterminer la bonne route à ajouter au cache à partir de la route trouvée” ?

```

    Si appartient(cache, route) alors
        cache : in
        route : in
        Null
    SinonSi table.all.route.adresselp masqué par route.masque = route.Adresselp et
    taille(table.all.route.masque)>taille route.masque Alors
        bon_masque := table.all.route.masque
    Sinon
        Null
    FinSi
    (-- faire de même avec table.all.suivant et la nouvelle valeur de bon_masque)

    route := {route.adresseip masqué par bon_masque ; bon_masque; route.interface }

```

R5 : Comment “déterminer la taille d’un cache_LA” ?

```

    Si cache_LA=Null alors
        Retourner 0
    SinonSi cache_LA.all.Fg=null et cache_LA.all.Fd=null Alors
        Retourner 1
    Sinon
        Retourner taille(cache_LA.all.Fg) + taille(cache_LA.all.Fd)
    FinSi

```

R5 : Comment “mettre à jour le compteur” ?

```

    Convertir route.Adressselp en un entier écrit en base 2
    Si cache.all.Fg=Null && cache.all.Fd=Null alors
        cache.all.date := d
    SinonSi route.Adressselp et (2**31 <>0) Alors
        Mettre à jour route.Adressselp*2 dans cache.all.Fd
    Sinon
        Mettre à jour route.Adressselp*2 dans cache.all.Fg
    FinSi

```

R5 : Comment “ajouter route à cache_LA” ?

```

    Si cache_LA=null alors
        cache ← new T_cache_LA
        cache.all.route := route; cache.all.date :=d ; cache.all.Fg=null ;cache.all.Fd=null
    SinonSi cache.all.Fg=null && cache.all.Fd=null alors
        Si route.adresselp and 2*31 = 0 alors
            cache.all.Fg ← new T_cache_LA

```

```
cache.all.Fg.all.Fg := null
cache.all.Fg.all.Fd := null
cache.all.Fg.all.route := route
cache.all.Fg.all.date := d
```

```
cache.all.Fd ← new T_cache_LA
cache.all.Fd.all.Fg := null
cache.all.Fd.all.Fd := null
cache.all.Fd.all.route := cache.all.route
cache.all.Fd.all.date := cache.all.date
```

```
cache.all.route := null ; cache.all.date := null;
```

Sinon

```
cache.all.Fd ← new T_cache_LA
cache.all.Fd.all.Fg := null
cache.all.Fd.all.Fd := null
cache.all.Fd.all.route := route
cache.all.Fd.all.date := d
```

```
cache.all.Fg ← new T_cache_LA
cache.all.Fg.all.Fg := null
cache.all.Fg.all.Fd := null
cache.all.Fg.all.route := cache.all.route
cache.all.Fg.all.date := cache.all.date
```

```
cache.all.route := null
cache.all.date := null;
```

FinSi

SinonSi (route.Adressselp **and** 2**31 <>0) **Alors**

Ajoute route.Adressselp*2 dans cache.all.Fd

Sinon

Ajoute route.Adressselp*2 dans cache.all.Fg

FinSi

R5 : Comment “trouver et supprimer la plus ancienne route” ?

Trouver la plus ancienne route

cache : **in**

route : **out**

Supprimer une route

cache : **in/out**

route : **in**

R6 : Comment “trouver la plus ancienne route” ?

ancienne_route : **T_route**

ancienne_date := date

Si cache =null **alors**

Nul

SinonSi cache.all.Fg=null **et** cache.all.Fd=null **et** cache.all.date < ancienne_date **Alors**

```
    ancienne_date := cache.all.date
    ancienne_route := cache.all.route
Si cache.all.Fg=null et cache.all.Fd=null et (non cache.all.date < ancienne_date) Alors
    Null
Sinon
    trouver la plus ancienne route dans cache.all.Fd
    trouver la plus ancienne route dans cache.all.Fg
FinSi
```

R6 : Comment “supprimer une route d’un cache_LA”

```
    Si (cache.all.Fg.all.Fg = null et cache.all.Fg.all.Fd = null) ou (cache.all.Fd.all.Fg = null et
cache.all.Fd.all.Fd = null) Alors
        Si (ancienn_route.Adressselp and 2**31 <>0) Alors
            cache.all.route := cache.all.Fg.all.route
            cache.all.date := cache.all.Fg.all.date
            cache.all.Fg := null
            cache.all.Fd := null
        Sinon
            cache.all.route := cache.all.Fd.all.route
            cache.all.date := cache.all.Fd.all.date
            cache.all.Fg := null
            cache.all.Fd := null
        FinSi
    Sinon
        Si (ancienn_route.Adressselp and 2**31 <>0) alors
            supprimer(cache.all.Fd,ancienne_route.adresse *2)
            Si (cache.all.Fg = null et cache.all.Fd.all.Fg=null et cache.all.Fd.all.Fd=null)
alors
                cache.all.route := cache.all.Fd.all.route
                cache.all.date := cache.all.Fd.all.date
                cache.all.Fd := null
            Sinon
                Null
            FinSi

        Sinon
            supprimer(cache.all.Fg,ancienne_route.adresse *2)
            Si (cache.all.Fd = null et cache.all.Fg.all.Fg=null et cache.all.Fg.all.Fd=null )
alors
                cache.all.route := cache.all.Fg.all.route
                cache.all.date := cache.all.Fg.all.date
                cache.all.Fg := null
            Sinon
                null
            FinSi
    FinSi
```