*IUT ANNECY- Département Informatique*

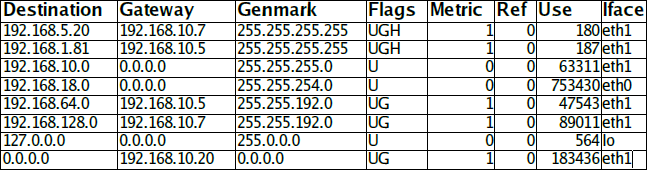
# Année 2018-2019

M3102 - Réseaux

TD 2 : Interconnexions

1. **Table de routage**

# On donne la table de routage relevées dans un routeur.



* 1. Combien de ports externes ce routeur possède-t-il ?

2 ports externes : 192.168.10.0/24, 192.168.18.0/23.

*Si une entrée de la table de routage ne comporte que le drapeau U, cela signifie que la destination est directement connectée à cette interface et qu'il ne s'agit pas d'un hôte (c'est un réseau).*

# *Une entrée avec des flags UG signifie que la destination est une adresse de réseau (pas une adresse d'hôte) et qu'elle n'est PAS directement connectée à cette interface.*

# *La présence de drapeaux UGH dans une entrée signifie que la destination est une adresse IP hôte et qu'elle n'est PAS directement connectée à cette interface.*

# *Le drapeau UH signifie que la destination est une adresse IP hôte, directement connectée à cette interface.*

# Expliquer la signification des flags

# U : Up = fonctionnelle

# G : Gateway

# H : Host

* 1. Dessiner un schéma montrant la configuration approximative du réseau interconnecté.

Flemme de dessiner le schéma (aller voir à la fin):

R1 => R2 (192.168.10.7) => machine (192.168.5.20)

R1 => R2 (192.168.10.7) => réseau (192.168.128.0)

R1 => R3 (192.168.10.5) => machine (192.168.1.81)

R1 => R3 (192.168.10.5) => réseau (192.168.64.0)

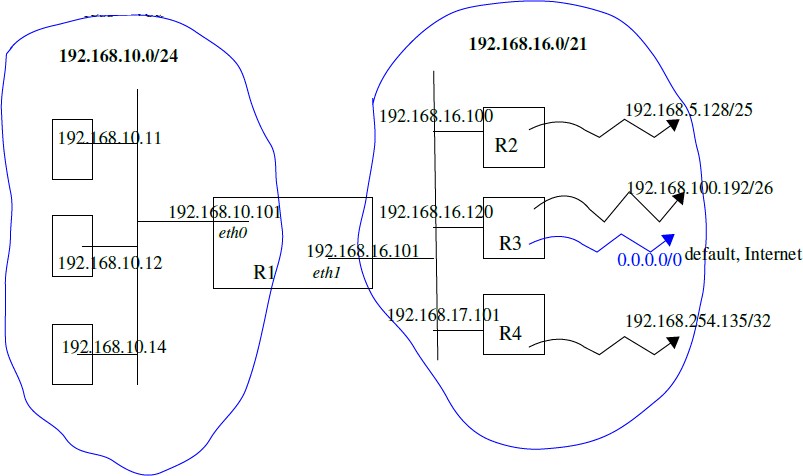
R1 => R4 (192.168.10.20) => Internet : route par default.

*Tout d'abord, remarquez que l'interface eth0 est sur le réseau 192.168.18.0/23 et que l'interface ethl est connectée au réseau 192.168.10.0/24.*

*Gardez à l'esprit que R sait seulement que pour envoyer des paquets de R au réseau 192.168.64.0/18 et à l'hôte 192.168.1.81, il doit envoyer des paquets à l'hôte 192.168.10.5 à travers l'interface eth1. Nous ne savons PAS comment 192.168.10.5 enverra des paquets à 192.168.64.0/18 ou 192.168.1.81.*

*d. L'adresse b'cast de l'interface eth0 est celle où les 9 derniers bits (bits d'hôte) de 192.168.18.0 sont mis à 1, ce qui donne 192.168.19.255. Pour l'interface eth1, l'adresse b'cast est celle où les 8 derniers bits de 192.168.10.0 sont à 1, soit 192.168.10.255.*

# Donner la table de routage simplifiée du routeur R1 ci-dessous en ne considérant que lles ports externes, et les colonnes : Destination | Gateway | Mask | Iface.



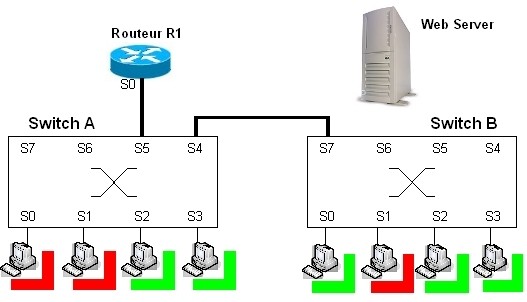
|  |  |  |  |
| --- | --- | --- | --- |
| **Destination** | **Gateway (passerelle)** | **Genmark (Masque)** | **(Iface) Interface** |
| 192.168.10.0 | 0.0.0.0 | 255.255.255.0 | Eth0 |
| 192.168.16.0 | 0.0.0.0 | 255.255.248.0 | Eth1 |
| 192.168.5.128 | 192.168.16.100 (R2) | 255.255.255.128 | Eth1 |
| 192.168.100.192 | 192.168.16.120  (R3) | 255.255.255.128 | Eth1 |
| 192.168.254.135 | 192.168.107.101  (R4) | 255.255.255.255 | Eth1 |
| 0.0.0.0 | 192.168.16.120 (R3) | 0.0.0.0 | Eth1 |

1. **VLans**

# On considère un réseau formé de 2 segments comportant chacun 4 machines, les segments étant reliés par 2 commutateurs A et B. On dispose en outre d'un routeur R1 relié au switch A. Pour l'adressage réseau des VLANs, on utilisera les adresses 192.168.<ID>.0 / 24 dans lesquelles <ID> représente l'identifiant de VLAN (2 pour les stations vertes et 4 pour les rouges).

* 1. Configurer les VLANs sur les deux commutateurs (sous forme de tables) et le routeur pour que les hôtes appartenant au VLAN rouge ainsi que ceux appartenant au VLAN vert soient directement connectés. Donner le plan d'adressage réseau correspondant à la configuration des VLANs.

# Donner deux configurations de VLAN possibles pour que le serveur Web soit accessible aux machines des VLANs vert et rouge.



2.1

Switch A

|  |  |  |
| --- | --- | --- |
| ***Interface*** | ***Mode*** | ***VLAN ID*** |
| S0 | Access | 1 |
| S1 | Access | 1 |
| S2 | Access | 2 |
| S3 | Access | 2 |
| S4 | Trunk | 1 & 2 |
| S5 | Trunk | 1 & 2 |

Switch B

|  |  |  |
| --- | --- | --- |
| ***Interface*** | ***Mode*** | ***VLAN ID*** |
| S0 | Access | 2 |
| S1 | Access | 1 |
| S2 | Access | 2 |
| S3 | Access | 2 |
| S7 | Trunk | 1 & 2 |

## R1

|  |  |  |
| --- | --- | --- |
| ***Interface*** | ***Mode*** | ***VLAN ID*** |
| S0 | Trunk | 1 & 2 |

*Plan d'adressage :*

*Commutateurs :*

VLAN 1 : adresse réseau 192.168.2.0 / 24

VLAN 2 : adresse réseau 192.168.4.0 / 24

*Routeur :*

Interface VLAN 1 ip address 192.168.2.17 / 24

Interface VLAN 2 ip address 192.168.4.17 / 24 (toute autre valeur que 17 est acceptable jusqu'à 254)

2.2.

*Solution A* : le serveur est connecté sur un port (e.g., S6) en utilisant une carte réseau (NIC).

|  |  |  |
| --- | --- | --- |
| ***Interface*** | ***Mode*** | ***VLAN ID*** |
| S0 | Access | 4 |
| S1 | Access | 2 |
| S2 | Access | 4 |
| S3 | Access | 4 |
| S6 | Trunk | 2, 4 |
| S7 | Trunk | 2, 4 |

Sur le serveur, on autorise le mode "trunk" et on configure deux adresses réseau sur la carte réseau : Interface VLAN 2 ip address 192.168.2.54 / 24 et Interface VLAN 4 ip address 192.168.4.54 / 24

*Solution B* : le serveur est connecté sur 2 ports (e.g., S5 et S6) en utilisant deux carte réseau.

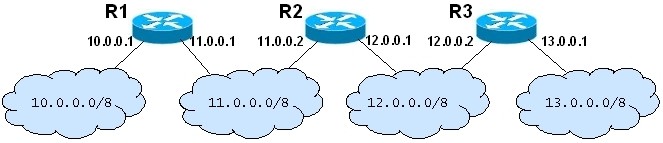
|  |  |  |
| --- | --- | --- |
| ***Interface*** | ***Mode*** | ***VLAN ID*** |
| S0 | Access | 4 |
| S1 | Access | 2 |
| S2 | Access | 4 |
| S3 | Access | 4 |
| S5 | Access | 2 |
| S6 | Access | 4 |
| S7 | Trunk | 2, 4 |

Sur le serveur, on configure deux cartes réseau, chacune sur un VLAN :

Carte S1 reliée à S5-switch B : ip address 192.168.2.54 / 24

Carte S2 reliée à S6-switch B : ip address 192.168.4.54 / 24.

## Routage à vecteur distance



On considère quatre réseaux reliés par trois routeurs qui utilisent un routage à vecteur distance (voir figure ci-dessus). On se place à l'état de convergence.

* 1. A quoi correspond la notion d'état de convergence.

**Etat de convergence** : une fois que chaque routeur a reçu toutes les tables de ses voisins (état stable)

* 1. Donner les trois tables de routage à l'état de convergence au format <destination>

<Interface><métrique>.

R1

|  |  |  |  |
| --- | --- | --- | --- |
| ***Destination*** | ***Gateway*** | ***Interface*** | ***Métrique*** |
| 10.0.0.0 | 0.0.0.0 |  | 0 |
| 11.0.0.0 | 0.0.0.0 |  | 0 |
| 12.0.0.0 |  | 11.0.0.2 | 1 |
| 13.0.0.0 |  | 11.0.0.2 | 2 |

R2

|  |  |  |  |
| --- | --- | --- | --- |
| ***Destination*** | ***Gateway*** | ***Interface*** | ***Métrique*** |
| 11.0.0.0 | 0.0.0.0 |  | 0 |
| 12.0.0.0 | 0.0.0.0 |  | 0 |
| 10.0.0.0 |  | 11.0.0.1 | 1 |
| 13.0.0.0 |  | 12.0.0.0 | 1 |

R3

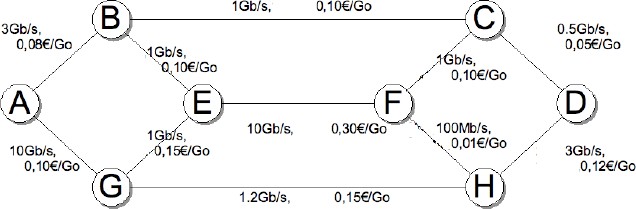
|  |  |  |  |
| --- | --- | --- | --- |
| ***Destination*** | ***Gateway*** | ***Interface*** | ***Métrique*** |
| 12.0.0.0 | 0.0.0.0 |  | 0 |
| 13.0.0.0 | 0.0.0.0 |  | 0 |
| 11.0.0.0 |  | 12.0.0.1 | 1 |
| 10.0.0.0 |  | 12.0.0.1 | 2 |

* 1. Supposons que l'interface 13.0.0.1 tombe en panne. Que se passe-t-il dans la table du routeur R3 ? Expliquez ce qui va se passer pour les tables de routage aux mises à jour suivantes. Quel est le problème ? Donnez au moins deux solutions possibles pour ce problème.
* R3 enlève la ligne correspondant à 13.0.0.0
* R3 reçoit la table de R2 et met à jour sa table avec la ligne 13.0.0.0 12.0.0.2 et incrémente la métrique qui passe à 2.
* R2 reçoit la table de R1 et met à jour sa table en incrémentant la métrique qui passe à 3.
* R1 reçoit la table de R2 et met à jour sa table en incrémentant la métrique qui passe à 4... On assiste à un processus de bouclage à l'infini.

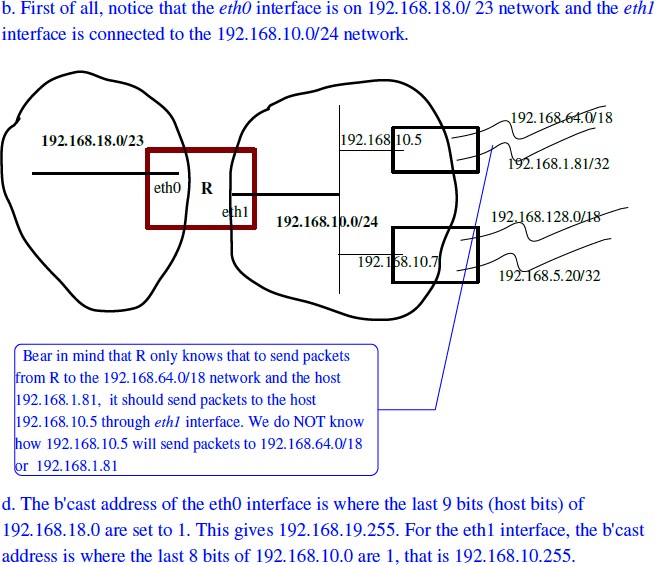
Exemple de solutions classiques : métrique maximum (limite du nombre de sauts, e.g., 16 pour RIP), le split horizon (filtrage de la direction d'apprentissage) et poison reverse (une route non valide est marquée avec un coût infini).

## Routage à état de lien

Utilisez la méthode qu’utilise OSPF (algorithme de Dijkstra) pour trouver les chemins les plus courts entre A et chaque nœud dans la structure ci-dessous en prenant successivement comme critère a) le débit b) le coût. Décrivez toutes les étapes. Quelle est dans chaque cas la distance entre A et D ?



1.3



4.

L'algorithme de Dijkstra est un algorithme de type glouton : à chaque nouvelle étape, on traite un nouveau sommet. Reste à définir le choix du sommet à traiter, et le traitement à lui infliger.

# Tout au long du calcul, on va donc maintenir deux ensembles :

* + *S*

# L’ensemble des sommets qui restent à visiter

* + *S*, l'ensemble des sommets pour lesquels on connaît déjà leur plus petite distance à la source ; au départ,

# L'algorithme se termine bien évidemment lorsque *S* est vide. Cas 1 : débit garanti → prendre le max

B

1

C

3

1

1

0.5

A

E

10

F

D

10

1

0.1

3

G

1.2

H

Initialisation :

*S*̄

# ={B,C,D,E,F,G,H} S={A}

Step 1:

(A)=0, (B)=3, (C)=+, (G)=10 ,(D)=(E)=(F)=(H)=+.

j=G

# *S*̄ ={B,C,D,E,F,H} S={A,G}

Step 2:

Step 3:

Step 4:

(H)=1.2, (E)=1,

(A)=0, (B)=3, (C)=+, (G)=10 ,(D)= +, (E)=1, (F)=+ (H)=1.2

j=B

# *S* ={C,D,E,F,H} S={A,G,B,}

(C)=1, (E)=1,

(A)=0, (B)=3, (C)=1, (G)=10 ,(D)= +, (E)=1, (F)=+ (H)=1.2

j=H

# *S* ={C,D,E,F} S={A,G,B,H}

(F)=0.1, (D)=1.2

(A)=0, (B)=3, (C)=1, (G)=10 ,(D)= 1.2, (E)=1, (F)=0.1 (H)=1.2

j=D

Step 5:

Step 6:

Step 7:

# *S* ={C,E,F} S={A,G,B,H,D}

(C)=0.5

(A)=0, (B)=3, (C)=1, (G)=10 ,(D)= 1.2, (E)=1, (F)=0.1 (H)=1.2

j=C

# *S* ={E,F} S={A,G,B,H,D,C}

(F)=1, (D)=0.5

(A)=0, (B)=3, (C)=1, (G)=10 ,(D)= 1.2, (E)=1, (F)=1 (H)=1.2

j=E

# *S* ={F} S={A,G,B,H,D,C,E}

(H)=5.1,

(A)=0, (B)=3, (C)=1, (G)=10 ,(D)= 1.2, (E)=1, (F)=1 (H)=1.2

j=F

*S* ={} Fin

d'où la table des débits maxi entre A et les autres points ainsi que le chemin utilisé. d(A,B) = 3

d(A,C) = 1 (par B) d(A,G) = 10

d(A,D) = 1.2 (par G, H)

d(A,E) = 1 (par B)

d(A,F) = 1 (par G, E)

d(A,H) = 1.2 (par G)

# Cas 1 : coût en euros → prendre le min

B 0.013

C

0.03

0.013

0.013

0.003

A

E

0.375

F

D

0.125

0.019

0.0001

0.045

G

0.023

H

Initialisation :

# *S* ={B,C,D,E,F,G,H} S={A}

(A)=0, (B)=0.03, (C)=+, (G)=0.125 ,(D)=(E)=(F)=(H)=+.

Step 1:

j=B

# *S* ={C,D,E,F,G,H} S={A,B}

Step 2:

(C)=0.043, (E)=0.043,

(A)=0, (B)=0.03, (C)=0.043, (G)=0.125 ,(D)=+ (E)=0.043, (F)=(H)=+

j=C

# *S* ={D,E,F,G,H} S={A,B,C}

Step 3:

(F)=0.056, (D)=0.046,

A)=0 )=0.03, C)=0.043, G)=0.125 D)==0.046 E)=0.043, F)=0.056, (H)=

j=D

# *S* ={E,F,G,H} S={A,B,C, D}

Step 4:

(H)=0.091,

A)=0 )=0.03, C)=0.043, G)=0.125 D)==0.046 E)=0.043, F)=0.056, (H)=

j=E

# *S* ={F,G,H} S={A,B,C,D,E}

Step 5:

(F)=0.418, (G)=0.062,

A)=0 )=0.03, C)=0.043, G)=0.062 D)==0.046 E)=0.043, F)=0.056, (H)=

j=F

# *S* ={G,H} S={A,B,C,D,E,F}

Step 6:

(E)=0.793, (H)=0.042,

A)=0 )=0.03, C)=0.043, G)=0.062 D)= E)=0.043, F) (H)=

j=H

# *S* ={G} S={A,B,C,D,E,F,H}

Step 7:

(D)=0.087, (G)=0.065,

A)=0 )=0.03, C)=0.043, G)=0.062 D)= E)=0.043, F) (H)=

j=G

*S* ={} S={A,B,C,D,E,F,H,G}

A)=0 )=0.03, C)=0.043, G)=0.062 D)= E)=0.043, F) (H)=

d'où la table des coûts mini entre A et les autres points ainsi que le chemin utilisé. d(A,B) = 0.03

d(A,C) = 0.043 (par B)

d(A,G) = 0.062 (par B, E)

d(A,D) = 0.046 (par B, C)

d(A,E) = 0.043 (par B)

d(A,F) = 0.056 (par B, C)

d(A,H) = 0.056 (par B, C, F)