

Répétitions du Cours d'Introduction aux Réseaux Informatiques

Couche Réseau - Première partie

François Cantin

Département Montefiore
Research Unit in Networking
Université de Liège

Année académique 2008–2009

Contact

- Bureau : Institut Montefiore (B28), 177a.
- Email : cantin@run.montefiore.ulg.ac.be.
- Téléphone : 04 366 26 94.

Table des matières

- 1 Adresses IP et acheminement des paquets IP
 - Attribution des adresses IP
 - Subnetting
 - Acheminement des paquets IP

Services rendus par la couche réseau

- Service principal = transfert de segments de la couche transport d'un hôte source à un hôte destinataire.
- Chaque hôte et chaque routeur est muni d'une couche réseau, constituée de plusieurs protocoles.
- A la source : encapsulation des segments de la couche transport dans des datagrammes.
- A la destination : décapsulation et passage des segments à la couche transport.
- Au cœur du réseau : chaque routeur examine l'en-tête de chaque datagramme reçu et détermine son interface de sortie.

Note : La construction des tables d'acheminement (*forwarding tables*) est une autre tâche de la couche réseau.

Internet (IP)

Le protocole de couche réseau communément utilisé dans l'Internet est IP. Ses principales caractéristiques sont :

- Service *best-effort* : aucune garantie de qualité de service.
- Pas d'établissement préalable de connexion.
- La décision de forwarding se base sur l'adresse destination uniquement.

Table des matières

- 1 Adresses IP et acheminement des paquets IP
 - Attribution des adresses IP
 - Subnetting
 - Acheminement des paquets IP

Classless InterDomain Routing (CIDR)

Principe général de CIDR

- Une adresse IP par interface réseau, pour chaque hôte et chaque routeur.
- Les adresses IP ont une longueur de 32 bits (\Rightarrow potentiel de 2^{32} hôtes)
- Problème : une table d'acheminement ne peut pas contenir 2^{32} entrées.
- Solution : allocation de blocs d'adresses successives et agrégation dans les tables (acheminement suivant la règle du *longest-prefix match*).

Classless InterDomain Routing (CIDR)

Blocs d'adresses

Les adresses IP ont une longueur de 32 bits. Elles sont attribuées par blocs aux ISPs et administrateurs de réseaux. La capacité d'un bloc d'adresses est déterminée par le nombre de bits qui peuvent être librement attribués.

La notation **w.x.y.z/s** signifie que

- Les s premiers bits sont imposés.
- Les $32 - s$ derniers bits peuvent être librement attribués par l'ISP ou l'administrateur.

Un bloc d'adresse peut lui-même être partitionné en sous-blocs.

Exercice 1 (Tanenbaum 4^{ème} édition, exercice 40)

Allocation de blocs d'adresses

Un grand nombre d'adresses IP consécutives sont disponibles à partir de 198.16.0.0. Supposons que quatre organismes A, B, C et D demandent respectivement 4000, 2000, 4000 et 8000 adresses (les demandes arrivent dans cet ordre). Pour chacune de ces demandes, donner

- la première adresse IP attribuée,
- la dernière adresse IP attribuée et
- le bloc d'adresses attribuées à l'aide de la notation $w.x.y.z/s$.

Résolution exercice 1

Organisme A

Comme $\log_2 4000 = \ln 4000 / \ln 2 = 11,97$, il faut 12 bits pour pouvoir encoder au moins 4000 adresses différentes.

La première adresse allouable est 198.16.0.0. On peut allouer le bloc d'adresses 198.16.0.0/20 (les 12 derniers bits sont libres).

Les adresses de ce bloc vont de 198.16.0.0 à 198.16.15.255. En effet, seuls les 4 derniers bits du troisième octet de l'adresse sont libres. La plus grande valeur qui peut être prise par cet octet est $00001111 = 2^4 - 1 = 15$.

- Bloc d'adresses alloué : 198.16.0.0/20
- Première adresse réservée : 198.16.0.0
- Dernière adresse réservée : 198.16.15.255

Résolution exercice 1 (2)

Organisme B

Comme $\log_2 2000 = \ln 2000 / \ln 2 = 10,97$, il faut 11 bits pour pouvoir encoder au moins 2000 adresses différentes.

La première adresse allouable est 198.16.16.0. On peut allouer le bloc 198.16.16.0/21

Les adresses de ce bloc vont de 198.16.16.0 à 198.16.23.255. En effet, le troisième octet de ces adresses est toujours de la forme 00010xxx (les 3 derniers bits sont libres). La plus grande valeur qui peut être prise par cet octet est $00010111 = 16 + (2^3 - 1) = 23$.

- Bloc d'adresses alloué : 198.16.16.0/21
- Première adresse réservée : 198.16.16.0
- Dernière adresse réservée : 198.16.23.255

Résolution exercice 1 (3)

Organisme C

On sait qu'il faut 12 bits pour pouvoir encoder au moins 4000 adresses différentes.

La première adresse allouable étant 198.16.24.0, on pourrait envisager d'allouer le bloc 198.16.24.0/20.

Non : ce groupe d'adresses contient des adresses déjà attribuées.

Le problème est que le masque de sous réseau correspondant à ce groupe est 255.255.240.0 ($240 = 11110000 \Rightarrow$ les 12 derniers bits sont libres) et que $24 = 00011000$: le 2^{ème} bit 1 de 24 est ignoré par le masque.

Par exemple, la première adresse du groupe 198.16.24.0/20 est $198.16.24.0 \text{ AND } 255.255.240.0 = 198.16.16.0$, une adresse déjà allouée à l'organisme B.

Résolution exercice 1 (4)

Donc :

- A a obtenu les adresses dont le 3^{ème} octet est 0000xxxx.
- B a obtenu les adresses dont le 3^{ème} octet est 00010xxx.

Le bloc d'adresses suivant suffisamment grand pour C (4 bits du 3^{ème} octet libres) est celui où le 3^{ème} octet est de la forme 0010xxxx. Les adresses de ce bloc vont de 198.16.32.0 à 198.16.47.255 ($32 + (2^4 - 1) = 47$).

- Bloc d'adresses alloué : 198.16.32.0/20
- Première adresse réservée : 198.16.32.0
- Dernière adresse réservée : 198.16.47.255

Il reste un 'trou' d'adresses non allouées : de 198.16.24.0 à 198.16.31.255.

Résolution exercice 1 (5)

Organisme D

Comme $\log_2 8000 = \ln 8000 / \ln 2 = 12,97$, il faut 13 bits pour pouvoir encoder au moins 8000 adresses différentes.

La première adresse allouable est 198.16.48.0. Cependant, comme $48 = 00110000$ et que les 5 derniers bits du 3^{ème} octet doivent rester libres, on ne peut pas définir un groupe à partir de cette adresse (même problème que pour C).

Le bloc d'adresses suivant suffisamment grand pour D est celui où le 3^{ème} octet est de la forme $010xxxxx$. Les adresses de ce bloc vont de 198.16.64.0 à 198.16.95.255 ($64 + (2^5 - 1) = 95$).

- Bloc d'adresses alloué : 198.16.64.0/19
- Première adresse réservée : 198.16.64.0
- Dernière adresse réservée : 198.16.95.255

Résolution exercice 1 (6)

Remarques

Si un organisme E demande un bloc pour 2000 adresses, on peut lui attribuer 198.16.24.0/21 (et ainsi "remplir" le trou laissé par l'allocation pour C).

Si un organisme F demande un bloc pour 4000 adresses, on peut lui attribuer 198.16.48.0/20.

Exercice 2 (Tanenbaum 4^{ème} édition, exercice 41)

Agrégation dans les tables d'acheminement

Un routeur vient juste de prendre connaissance des adresses IP suivantes : 57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/21 et 57.6.120.0/21.

Si elles utilisent la même ligne de sortie, peuvent-elles être agrégées ? Si oui, quel est le résultat de l'agrégation ? Sinon, pourquoi ?

Résolution exercice 2

Analyse du premier bloc d'adresses

Une adresse de ce bloc a les 21 premiers bits identiques aux 21 premiers bits de la séquence binaire qui représente l'adresse 57.6.96.0, tandis que ses 11 derniers bits sont libres. Ce bloc comporte 2^{11} adresses différentes.

Le dernier octet est donc totalement libre, ainsi que les 3 derniers bits de l'avant-dernier octet. La dernière adresse de ce bloc est donc 57.6.103.255 (car $96 + (2^3 - 1) = 103$).

L'analyse des autres blocs d'adresses peut être réalisé de la même manière.

Résolution exercice 2 (2)

Résultat de l'analyse de chaque bloc

57.6.96.0/21 représente les adresses 57.6.96.0 à 57.6.103.255

57.6.104.0/21 représente les adresses 57.6.104.0 à 57.6.111.255

57.6.112.0/21 représente les adresses 57.6.112.0 à 57.6.119.255

57.6.120.0/21 représente les adresses 57.6.120.0 à 57.6.127.255

Résolution exercice 2 (3)

On peut agréger ces quatre blocs en un seul car :

- toutes ces adresses sont consécutives et
- au total, il y en a $2^{11} \cdot 4 = 2^{13}$ (2^n , avec n un nombre entier).

⇒ **Bloc agrégé** : 57.6.96.0/19

Vérification (calcul de la dernière adresse de ce bloc) :

- Nombre de bits libres : $32 - 19 = 13$
- Répartition sur les derniers octets : $13 = 5 + 8$
- Nombre d'adresses différentes allouables sur le troisième octet : $2^5 = 32$
- Valeur du troisième octet de la première adresse du bloc suivant : $96 + 32 = 128$
- Dernière adresse de ce bloc : 57.6.127.255 ⇒ ok.

Exercice 3 (Tanenbaum, 4^{ème} éd., ex. 42, énoncé corrigé)

Agrégation dans les tables d'acheminement

L'ensemble des adresses IP allant de 29.18.0.0 à 29.18.127.255 a été agrégé à 29.18.0.0/17 et il y a un trou de 1024 adresses non assignées entre 29.18.60.0 et 29.18.63.255. Supposons que ces 1024 adresses soient maintenant assignées à un hôte accessible via une autre ligne de sortie.

Est-il nécessaire de diviser l'agrégat en plusieurs blocs, d'ajouter le nouveau bloc dans la table d'acheminement puis de voir si d'autres agrégations sont possibles ? Sinon, qu'est-il possible de faire ?

Résolution exercice 3

Les constituants du bloc initial sont :

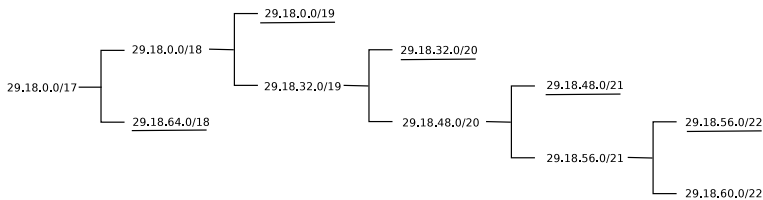
- 1 De 29.18.0.0 à 29.18.59.255
- 2 De 29.18.60.0 à 29.18.63.255 (= 29.18.60.0/22)
- 3 De 29.18.64.0 à 29.18.127.255 (= 29.18.64.0/18)

Le premier sous-bloc couvre $60 \cdot 2^8$ adresses. Or 60 n'est pas un exposant entier de 2. On ne peut donc pas regrouper toutes ces adresses dans un seul bloc avec la notation CIDR (w.x.y.z/s).

- 29.18.0.0/18 couvre toutes les adresses de 29.18.0.0 à 29.18.63.255, ce qui est trop grand,
- 29.18.0.0/19 couvre seulement les adresses de 29.18.0.0 à 29.18.31.255, ce qui est trop petit.

Résolution exercice 3 (2)

Pour pouvoir exprimer tous les constituants du bloc de départ avec la notation CIDR, il faut poursuivre la décomposition du premier sous-bloc :



Les sous-blocs soulignés sont ceux que l'on devrait faire apparaître dans la table de routage, sans l'existence de la règle de *longest prefix match*.

Résolution exercice 3 (3)

Grâce à la règle de *longest-prefix match*, la table de routage suivante, composée de deux lignes uniquement, ne présente aucune ambiguïté

Address/mask	Next hop
29.18.0.0/17	Interface 0
29.18.60.0/22	Interface 1

Par exemple un paquet adressé à 19.18.62.9 rend les deux entrées valides, mais c'est l'interface de sortie 1 qui sera choisie parce que le masque associé à la deuxième entrée de la table de routage est plus long. Par contre, tout paquet dont l'adresse de destination n'appartient pas au bloc de 29.18.60.0 à 29.18.63.255 ne rendra que la première entrée valide et sera dirigé vers l'interface 0.

Résolution exercice 3 (4)

Il est donc inutile de diviser le bloc initial en tous ses constituants et d'ajouter chacun d'eux dans la table de routage.

Remarque : Cette diminution de taille de la table d'acheminement n'est possible que grâce à la règle du longest-prefix match. Ce n'était pas le cas dans l'exercice précédent : il s'agissait juste d'une simplification d'écriture ne nécessitant pas la règle du longest-prefix match pour fonctionner.

Table des matières

- 1 Adresses IP et acheminement des paquets IP
 - Attribution des adresses IP
 - **Subnetting**
 - Acheminement des paquets IP

Subnetting

Motivation

Le *subnetting* permet à un administrateur de scinder son réseau en plusieurs sous-réseaux indépendants au niveau interne, sans que cette modification ne soit visible de l'extérieur.

Il est pratiqué lorsqu'un administrateur dispose de suffisamment d'adresses, mais ne désire pas installer tous les hôtes sur le même réseau local, pour des raisons physiques (canal partagé, distance) ou de sécurité.

Subnetting

Technique (principe)

Considérons un réseau ayant reçu un bloc d'adresses muni d'un masque de longueur 16.

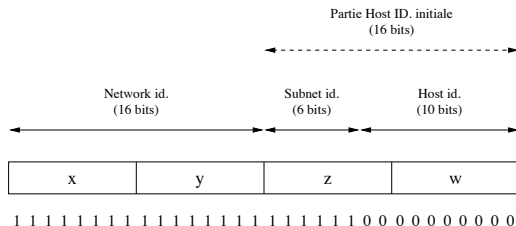
Soit un hôte ayant reçu une adresse $x.y.z.w$ de ce bloc. La partie $x.y$ est appelée son identificateur de réseau et la partie $z.w$ son identificateur d'hôte.

La technique du subnetting consiste à consacrer une partie de l'identificateur d'hôte pour distinguer les différents réseaux locaux.

Subnetting

Technique (exemple)

Voici comment subdiviser ce réseau en 64 (2^6) sous-réseaux.



Règle pratique : pas d'identificateur d'hôte rempli uniquement de '0'¹ ou de '1'².

Chaque sous-réseau de cet exemple peut comporter 1022 ($2^{10} - 2$) hôtes. Notez la perte de 126 adresses car $64 \times 1022 = (2^{16} - 2) - 126$.

¹Il s'agit de "l'adresse du réseau".

²Il s'agit de l'adresse broadcast pour le réseau.

Exercice 5

Un réseau /16 est subdivisé en sous-réseaux /20.

- 1 Quel est le nombre maximum de stations par sous-réseau ?
- 2 Donnez le nombre d'adresses disponibles pour adresser individuellement chaque équipement.
- 3 Si l'adresse IP d'une station est égale à 192.128.130.10, déterminez son identificateur local sur le sous-réseau.

Résolution exercice 5

- ① Le masque de sous-réseau sous la forme binaire est :

11111111.11111111.11110000.00000000

Le host ID est composé 12 bits. Comme on réserve les deux identificateurs constitués d'une suite de 0 ou d'une suite de 1, on dispose de 4094 adresses sur chaque sous-réseau.

- ② Comme le subnet ID est composé de 4 bits, on dispose de $2^4 = 16$ sous-réseaux. Le nombre total de machines qu'il est possible d'adresser est donc $16 \times 4094 = 65504$.
- ③ L'adresse IP de la station sous forme binaire est :

11000000.10000000.10000010.00001010

Comme on ne doit tenir compte que des 12 bits de poids faible de l'adresse IP, l'identificateur local sur le sous-réseau est : $512 + 8 + 2 = 522$.

Table des matières

- 1 Adresses IP et acheminement des paquets IP
 - Attribution des adresses IP
 - Subnetting
 - Acheminement des paquets IP

Traitement d'un paquet par un routeur (technique)

En interne, le routage entre les différents sous-réseaux est possible si et seulement si les tables d'acheminement comportent les **masques de sous-réseau**.

A la réception d'un paquet, un routeur :

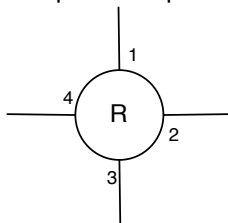
- 1 Détermine l'ensemble des entrées i de la table d'acheminement telles que

$$packet.IP_dest \text{ AND } table[i].netmask = table[i].IP_dest$$

- 2 S'il y a plusieurs entrées qui vérifient cette règle, il sélectionne celle qui a le masque le plus long (position du dernier '1').

Traitement d'un paquet par un routeur (exemple)

Voici par exemple un extrait de table d'acheminement :



Destination IP	Netmask	Interface
139.165.0.0	255.255.0.0	1
139.165.8.0	255.255.255.0	2
139.165.16.0	255.255.255.0	3
139.165.223.0	255.255.255.0	4
default (0.0.0.0)	0.0.0.0	1

Supposons qu'un paquet arrive par l'interface 4, voici l'interface de sortie pour différentes valeurs d'adresse destination :

- $5.1.1.1 \text{ AND } 0.0.0.0 = 0.0.0.0 \Rightarrow$ interface **1**
- $139.165.8.1 \text{ AND } 255.255.255.0 = 139.165.8.0 \Rightarrow$ interface **2**
(meilleure que 1)
- $139.165.10.1 \text{ AND } 255.255.0.0 = 139.165.0.0 \Rightarrow$ interface **1**

Exercice 6 (Tanenbaum, 4^{ème} édition, exercice 43)

Application de la règle du longest prefix match

Un routeur a les entrées suivantes dans sa table d'acheminement :

Address/mask	Next hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Interface 2
<i>default</i>	Interface 3

Pour chacune des adresses IP suivantes, que fera le routeur si un paquet comportant cette adresse arrive ?

- 1 135.46.63.10
- 2 135.46.57.14
- 3 135.46.52.2
- 4 192.53.40.7
- 5 192.53.56.7

Résolution exercice 6

Le masque /22 sous la forme binaire est :

$$11111111.11111111.11111100.00000000 = 255.255.252.0$$

Le masque /23 sous la forme binaire est :

$$11111111.11111111.11111110.00000000 = 255.255.254.0$$

La table d'acheminement peut être réécrite :

Destination IP	Netmask	Interface
135.46.56.0	255.255.252.0	0
135.46.60.0	255.255.252.0	1
192.53.40.0	255.255.254.0	2
default (0.0.0.0)	0.0.0.0	3

Résolution exercice 6 (2)

135.46.63.10

Nous avons :

$$IP_{dest} = 10000111.00101110.00111111.00001010$$

$$Mask_1 = 11111111.11111111.11111100.00000000$$

$$\begin{aligned} IP_{dest} \text{ AND } Mask_1 &= 10000111.00101110.00111100.00000000 \\ &= 135.46.60.0 \end{aligned}$$

et

$$IP_{dest} = 10000111.00101110.00111111.00001010$$

$$Mask_2 = 11111111.11111111.11111110.00000000$$

$$\begin{aligned} IP_{dest} \text{ AND } Mask_2 &= 10000111.00101110.00111110.00000000 \\ &= 135.46.62.0 \end{aligned}$$

Résolution exercice 6 (3)

packet.IP_dest AND table.mask	table.IP_dest	Entrée valide	Longueur du masque	Iface
135.46.60.0	135.46.56.0	non	22 bits	0
135.46.60.0	135.46.60.0	oui	22 bits	1
135.46.62.0	192.53.40.0	non	23 bits	2
0.0.0.0	0.0.0.0	oui	0 bit	3

Le paquet doit être envoyé sur l'interface 1 (longest prefix match).

135.46.57.14

packet.IP_dest AND table.mask	table.IP_dest	Entrée valide	Longueur du masque	Iface
135.46.56.0	135.46.56.0	oui	22 bits	0
135.46.56.0	135.46.60.0	non	22 bits	1
135.46.56.0	192.53.40.0	non	23 bits	2
0.0.0.0	0.0.0.0	oui	0 bit	3

Le paquet doit être envoyé sur l'interface 0 (longest prefix match).

Résolution exercice 6 (4)

135.46.52.2

packet.IP_dest AND table.mask	table.IP_dest	Entrée valide	Longueur du masque	Iface
135.46.52.0	135.46.56.0	non	22 bits	0
135.46.52.0	135.46.60.0	non	22 bits	1
135.46.52.0	192.53.40.0	non	23 bits	2
0.0.0.0	0.0.0.0	oui	0 bit	3

Le paquet doit être envoyé sur l'interface 3.

192.53.40.7

packet.IP_dest AND table.mask	table.IP_dest	Entrée valide	Longueur du masque	Iface
192.53.40.0	135.46.56.0	non	22 bits	0
192.53.40.0	135.46.60.0	non	22 bits	1
192.53.40.0	192.53.40.0	oui	23 bits	2
0.0.0.0	0.0.0.0	oui	0 bit	3

Le paquet doit être envoyé sur l'interface 2 (longest prefix match).

Résolution exercice 6 (5)

192.53.56.7

packet.IP_dest AND table.mask	table.IP_dest	Entrée valide	Longueur du masque	Iface
192.53.56.0	135.46.56.0	non	22 bits	0
192.53.56.0	135.46.60.0	non	22 bits	1
192.53.56.0	192.53.40.0	non	23 bits	2
0.0.0.0	0.0.0.0	oui	0 bit	3

Le paquet doit être envoyé sur l'interface 3.

Contenu de la table d'acheminement (1)

Lorsqu'un hôte (*host* ou *end system*) veut envoyer un datagramme, deux situations peuvent se présenter :

- 1 La destination est sur le même LAN : le datagramme est directement envoyé à la destination (service rendu par la couche 2, étudiée ultérieurement)
- 2 La destination est distante : le datagramme est envoyé à un routeur connecté au LAN (*idem*), qui dirige le datagramme vers la destination ou vers un routeur plus proche de la destination (service de couche 3, étudié ici)

Contenu de la table d'acheminement (2)

Pour parvenir à ce comportement, chaque entrée de la table d'acheminement contient les champs suivants :

Adresse IP de destination Il peut s'agir de l'adresse d'un hôte ou d'un réseau. La distinction se fait grâce aux *flags*.

Adresse IP du next-hop routeur Le routeur permettant de se rapprocher d'un hop de la destination finale.

Masque de (sous-)réseau de la destination

Flags Décrits au transparent suivant.

Interface réseau Le nom de l'interface qui doit être utilisée pour transmettre le datagramme.

Les champs "Next hop" et "Interface" ont un rôle complémentaire. Le next hop est nécessaire si la destination se trouve sur un réseau distant tandis que l'interface réseau à employer est nécessaire pour un hôte situé sur un LAN auquel le routeur est attaché.

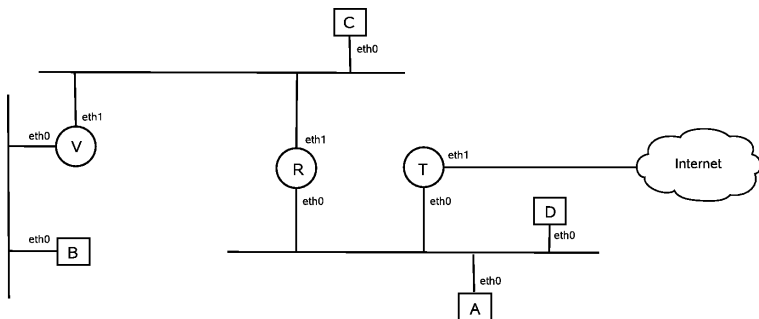
Contenu de la table d'acheminement (3)

- Flags :
 - U La route est en service (up).
 - H La route mène à un hôte (host) (càd la destination est l'adresse d'un hôte). L'absence du flag signifie que la route mène à un réseau.
 - G La route mène à un routeur (gateway). L'absence du flag signifie que la destination est directement connectée.
- Exemple (obtenu avec la commande *route* sous Linux) :

Destination	Gateway	Genmask	Flags	Iface
139.165.223.0	*	255.255.255.0	U	eth0
default	139.165.223.1	0.0.0.0	UG	eth0

Exercice 7

On considère le réseau IP suivant :



Topologie pour l'exercice 7

Exercice 7 (2)

- 1 Donnez l'adresse IP de *D* sachant que son host ID est 513.
- 2 Donnez les tables d'acheminement de *R*, *V*, *A* et *B*.
- 3 Considérons un transfert unidirectionnel de données entre *A* et *B*. Donnez les adresses IP qui sont utilisées dans les datagrammes qui circulent sur le réseau.

Données du réseau IP (routeurs)		
Routeur	Interface	Host ID
<i>R</i>	eth0	2
	eth1	2
<i>T</i>	eth0	3
	eth1	3
<i>V</i>	eth0	4
	eth1	4

Données du réseau IP (stations)			
Station	Interface	IP	Netmask
<i>A</i>	eth0	140.140.140.1	0xffff0000
<i>B</i>	eth0	150.150.150.1	0xfffff00
<i>C</i>	eth0	160.160.160.1	0xfffff00
<i>D</i>	eth0	?	?

Résolution exercice 7

1 Pour déterminer l'adresse IP de D , il nous manque son netmask et son network ID. Ces valeurs sont identiques pour toutes les adresses IP se trouvant sur le même sous-réseau que D . Nous pourrions donc utiliser $IP_{R_{eth0}}$, $IP_{T_{eth0}}$ ou IP_A . Nous avons :

$$IP_A = 10001100.10001100.10001100.00000001$$

$$Netmask = \underline{11111111.11111111.00000000.00000000}$$

$$IP_A \text{ AND } Netmask = 10001100.10001100.00000000.00000000$$

L'adresse IP de D est de la forme

$$10001100.10001100.xxxxxxxx.xxxxxxxx$$

où la partie non définie est complétée avec le hostID. Le hostID de D est 513. L'encodage de $513 = 512 + 1 = 2^9 + 2^0$ sur 16 bits est 0000001000000001. L'adresse du terminal D est donc :

$$\begin{aligned} IP_D &= 10001100.10001100.00000010.00000001 \\ &= 140.140.2.1 \end{aligned}$$

Résolution exercice 7 (2)

2 La table d'acheminement la plus probable pour R est :

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
140.140.0.0	*	255.255.0.0	U	eth0
150.150.150.0	160.160.160.4	255.255.255.0	UG	eth1
160.160.160.0	*	255.255.255.0	U	eth1
default	140.140.0.3	0.0.0.0	UG	eth0

Procédure à suivre :

- 1 Déterminer les identifiants et les masques des sous-réseaux.
- 2 Déterminer les adresses IP des interfaces des routeurs.
- 3 Créer une entrée dans la table par sous-réseau + une entrée localhost + une entrée default (si nécessaire).
- 4 Simplifier la table (si possible).

Résolution exercice 7 (3)

Une table d'acheminement possible pour V est :

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
150.150.150.0	*	255.255.255.0	U	eth0
160.160.160.0	*	255.255.255.0	U	eth1
140.140.0.0	160.160.160.2	255.255.0.0	UG	eth1
<i>default</i>	160.160.160.2	0.0.0.0	UG	eth1

Le comportement du routeur pour les entrées 140.140.0.0 et *default* est le même (envoi du paquet à R_{eth1}) et on peut regrouper ces entrées. La table d'acheminement la plus probable pour V est :

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
150.150.150.0	*	255.255.255.0	U	eth0
160.160.160.0	*	255.255.255.0	U	eth1
<i>default</i>	160.160.160.2	0.0.0.0	UG	eth1

Résolution exercice 7 (4)

La table d'acheminement la plus probable pour A est :

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
140.140.0.0	*	255.255.0.0	U	eth0
160.160.160.0	140.140.0.2	255.255.255.0	UG	eth0
150.150.150.0	140.140.0.2	255.255.255.0	UG	eth0
<i>default</i>	140.140.0.3	0.0.0.0	UG	eth0

Le comportement du routeur pour les entrées 160.160.160.0 et 150.150.150.0 est le même mais regrouper ces entrées est impossible. Le plus petit groupe contenant toutes les adresses de ces sous-réseaux est 128.0.0.0/2 et la table de A ne peut pas contenir l'entrée

128.0.0.0	140.140.0.2	192.0.0.0	UG	eth0
-----------	-------------	-----------	----	------

Prenons, par exemple, l'adresse 128.1.1.1. Les entrées valides pour cette adresse seraient 128.0.0.0 et *default*. Comme le masque de 128.0.0.0 est plus long, c'est cette entrée qui sera choisie \Rightarrow Erreur : cette adresse devrait être routée vers "Internet".

Résolution exercice 7 (5)

Une table d'acheminement possible pour B est

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
150.150.150.0	*	255.255.255.0	U	eth0
160.160.160.0	150.150.150.4	255.255.255.0	UG	eth0
140.140.0.0	150.150.150.4	255.255.0.0	UG	eth0
<i>default</i>	150.150.150.4	0.0.0.0	UG	eth0

Le comportement du routeur pour les entrées 160.160.160.0, 140.140.0.0 et *default* est le même et on peut regrouper ces entrées. La table d'acheminement la plus probable pour B est :

Destination	Gateway	Genmask	Flags	Iface
localhost	*	255.255.255.255	UH	lo0
150.150.150.0	*	255.255.255.0	U	eth0
<i>default</i>	150.150.150.4	0.0.0.0	UG	eth0

Résolution exercice 7 (5)

3 Le paquet doit transiter par R . Les adresses IP source et destination du paquet sont respectivement 140.140.140.1 et 150.150.150.1.

Important : Lors de l'envoi d'un paquet IP de A vers B l'adresse source du paquet sera IP_A et l'adresse de destination du paquet sera IP_B depuis l'émission du paquet par A jusqu'à la réception de celui-ci par B : **on ne modifie pas les adresses source et destination d'un paquet IP en cours de route.**

Exercice 8

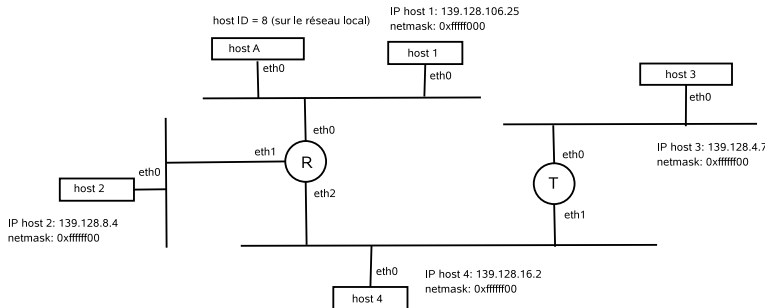
Considérons le réseau, sur le transparent suivant, interconnectant un ensemble de segments Ethernet.

- 1 Donnez l'adresse IP du terminal *A*.
- 2 Donnez les champs d'adresses source et destination dans les en-têtes IP quand on considère un transfert de données unidirectionnel *host 3* \rightarrow *host 1*.
- 3 Quelle est la table d'acheminement la plus probable de *R*, de *T*, de *host 1* et de *host 4* ?

Les informations relatives aux deux routeurs sont :

Interface	IP	Netmask
<i>routeur R</i>		
eth0	139.128.106.26 (R_0)	0xfffff000
eth1	139.128.8.5 (R_1)	0xfffff000
eth2	139.128.16.3 (R_2)	0xfffff000
<i>routeur T</i>		
eth0	139.128.4.8 (T_0)	0xfffff000
eth1	139.128.16.4 (T_1)	0xfffff000

Exercice 8 (2)



Topologie pour l'exercice 8

Résolution exercice 8

A réaliser comme exercice personnel. Il sera corrigé au début de la deuxième séance d'exercices sur IP.