
M1101

Initiation aux Réseaux

Travaux dirigés

Sami Evangelista
IUT de Villetaneuse
Département Réseaux et Télécommunications
2017–2018

<http://www.lipn.univ-paris13.fr/~evangelista/cours/M1101>

Table des matières

TD 1 — Codage de l'information	2
TD 2 — Internet Protocol	7
TD 3 — Services DNS et DHCP	13

TD 1 — Codage de l'information

Exercice 1 — Changement de base

Q 1.1 Convertir en nombres décimaux les nombres binaires suivants : 11, 1101, 100101110.

Correction

- $11_2 = 2^1 + 2^0 = 3$
- $1101_2 = 2^3 + 2^2 + 2^0 = 13$
- $100101110_2 = 2^8 + 2^5 + 2^3 + 2^2 + 2^1 = 302$

Q 1.2 Convertir en nombres binaires les nombres décimaux suivants : 7, 51, 128, 131, 234.

Correction

Pour les 4 premiers nombres on va directement décomposer en puissance de 2 :

- $7 = 4 + 2 + 1 = 2^2 + 2^1 + 2^0 = 111_2$
- $51 = 32 + 16 + 2 + 1 = 2^5 + 2^4 + 2^1 + 2^0 = 11\ 0011_2$
- $128 = 2^7 = 1000\ 0000_2$
- $131 = 2^7 + 2^1 + 2^0 = 1000\ 0011_2$

Pour 234 on va appliquer la méthode donnée dans le cours qui repose sur des divisions successives :

- $234 = 2 \times 117 + 0$
- $117 = 2 \times 58 + 1$
- $58 = 2 \times 29 + 0$
- $29 = 2 \times 14 + 1$
- $14 = 2 \times 7 + 0$
- $7 = 2 \times 3 + 1$
- $3 = 2 \times 1 + 1$
- $1 = 2 \times 0 + 1$

Ce sont les restes des divisions qui forment les bits du résultat. Le premier reste obtenu donne le bit de poids faible, le second donne celui de poids 1 et ainsi de suite. Donc $234 = 11101010_2$.

Q 1.3 Convertir en nombres binaires puis en nombres décimaux les nombres hexadécimaux suivants : 12, DADA et 5F3.

Correction

Pour la conversion hexadécimal → binaire :

- $12_{16} = 0001\ 0010_2$
- $DADA_{16} = 1101\ 1010\ 1101\ 1010_2$
- $5F3_{16} = 0101\ 1111\ 0011_2$

Pour la conversion hexadécimal → décimal :

- $12_{16} = 1 \times 16^1 + 2 \times 16^0 = 18$
- $DADA_{16} = 13 \times 16^3 + 10 \times 16^2 + 13 \times 16^1 + 10 \times 16^0 = 56\ 026$
- $5F3_{16} = 5 \times 16^2 + 15 \times 16^1 + 3 \times 16^0 = 1\ 523$

Exercice 2 — Opérations binaires

On travaille dans cet exercice sur des entiers naturels codés sur un octet.

Q 2.1 Quels sont les entiers naturels que l'on peut représenter sur un octet ?

Correction

Avec un octet (8 bits) on peut coder 2^8 valeurs : tous les entiers naturels de 0 (codé 0000 0000 en binaire) à $2^8 - 1 = 255$ (codé 1111 1111 en binaire).

Q 2.2 Donner le résultat des opérations suivantes en binaire puis en décimal.

- (a) $0001\ 0101_2 + 1011\ 0111_2$
- (b) $0100\ 0111_2 + 1101\ 1001_2$
- (c) 134_{10} **ET** 244_{10}
- (d) 17_{10} **OU** 123_{10}
- (e) **NON** 27_{10}
- (f) 44_{10} **XOR** 157_{10}

Correction

- (a) Calcul de
- $0001\ 0101_2 + 1011\ 0111_2$

$$\begin{array}{r}
 \\
 \text{ retenues} \\
 + \\
 \hline
 \\
 \\
 \hline
 = 2^7 + 2^6 + 2^3 + 2^2 = 204_{10}
 \end{array}$$

- (b) Calcul de
- $0100\ 0111_2 + 1101\ 1001_2$

$$\begin{array}{r}
 \text{ retenues} \\
 \\
 + \\
 \hline
 \\
 \\
 \hline
 = 2^5 = 32_{10}
 \end{array}$$

On observe ici un dépassement de capacité : le résultat est codé sur 9 bits alors qu'on ne dispose que de 8 bits. Le résultat de l'opération est donc erroné.

- (c) Calcul de
- 134_{10}
- ET
- 244_{10}

$$134_{10} = 1000\ 0110_2 \text{ et } 244_{10} = 1111\ 0100_2$$

$$\begin{array}{r}
 \\
 \text{ET} \\
 \\
 \hline
 \\
 \\
 \hline
 = 2^7 + 2^2 = 132_{10}
 \end{array}$$

- (d) Calcul de
- 17_{10}
- OU
- 123_{10}

$$17_{10} = 0001\ 0001_2 \text{ et } 123_{10} = 0111\ 1011_2$$

$$\begin{array}{r}
 \\
 \text{OU} \\
 \\
 \hline
 \\
 \\
 \hline
 = 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = 123_{10}
 \end{array}$$

- (e) Calcul de
- NON**
- 27_{10}

$$17_{10} = 0001\ 1011_2$$

$$\begin{array}{r}
 \text{NON} \\
 \\
 \hline
 \\
 \\
 \hline
 = 2^7 + 2^6 + 2^5 + 2^2 = 228_{10}
 \end{array}$$

- (f) Calcul de
- 44_{10}
- XOR
- 157_{10}

$$44_{10} = 0010\ 1100_2 \text{ et } 157_{10} = 1001\ 1101_2$$

$$\begin{array}{r}
 \\
 \text{XOR} \\
 \\
 \hline
 \\
 \\
 \hline
 = 2^7 + 2^5 + 2^4 + 2^0 = 177_{10}
 \end{array}$$

Q 2.3 Soit x un octet quelconque. Que vaut le résultat (en binaire ou en décimal) des opérations suivantes :

- (a)
- x
- OU
- 255_{10}

- (b)
- x
- ET
- 255_{10}

- (c) NON
- x

Correction

- (a) $255_{10} = 1111\ 1111_2$. Or comme 1 OU $x = 1$ quel que soit x , on aura toujours x OU $255_{10} = x$ OU $1111\ 1111_2 = 255$.

- (b) Comme 1 ET $x = x$ quel que soit x , on aura toujours x ET $255_{10} = x$ ET $1111\ 1111_2 = x$.

- (c) $255 - x$

Exercice 3 — Décalage binaire

Coder en binaire les nombres 26 et 52. Que remarque-t-on ? En déduire une méthode rapide pour multiplier ou diviser par 2^k un nombre binaire. Généraliser à une base B quelconque.

Correction

Le codage donne

$$— 26_{10} = 11010_2$$

$$— 52_{10} = 110100_2$$

On remarque que la représentation binaire de 52 est obtenue à partir de celle de 26 en rajoutant un bit de poids faible à 0. On appelle cette opération un décalage à gauche.

De manière générale, en rajoutant k bits de poids faible à 0 on multiplie le nombre par 2^k . Avant le décalage, la puissance de 2 associée au bit de poids p est 2^p . Après le décalage, cette puissance devient 2^{k+p} . On a donc multiplié par 2^k chacune des puissances associées de même que la somme des puissances.

En suivant le même raisonnement, on déduit que, dans une base B quelconque, décaler à gauche de k chiffres revient à multiplier par B^k . Par exemple, en hexadécimal, $D3_{16} = 211$ et $D300_{16} = 211 \times 16^2 = 54\,016$. ♦

Exercice 4 — Jeu de cartes

On s'intéresse à des jeux de 52 cartes réparties en 4 couleurs (pique, cœur, carreau et trèfle) de 13 cartes désignées par leurs rangs (As, 2, 3, ..., 10, Valet, Dame et Roi).

Q 4.1 Proposer un schéma de codage binaire des cartes du jeu.

Correction

Il y a quatre couleurs. Pour coder ces quatre couleurs on a besoin de 2 bits (car $2^2 \geq 4$). On peut ensuite utiliser le codage suivant : 00 → pique, 01 → cœur, 10 → carreau et 11 → trèfle.

Pour coder les 13 rangs on a besoin de 4 bits (3 bits ne sont pas suffisants car $2^3 = 8$ mais avec 4 bits on peut en coder $2^4 = 16 \geq 13$). On peut ensuite utiliser le codage suivant : 0000 → As, 0001 → 2, 0010 → 3, ..., 1100 → Roi.

Une carte peut donc être codée avec 6 bits $c_1, c_0, r_3, r_2, r_1, r_0$, les bits c_1, c_0 indiquant la couleur et r_3, r_2, r_1, r_0 le rang. ♦

Q 4.2 Donner, dans ce schéma, la représentation binaire du valet de trèfle.

Correction

Avec la représentation choisie, on a $c_1 c_0 = 11$ pour la couleur trèfle et $r_3 r_2 r_1 r_0 = 1010$ pour le valet. Donc 111010 est la représentation binaire du valet de trèfle. ♦

Q 4.3 On considère deux cartes dont les représentations binaires sont C_1 et C_2 . Sous quelles conditions, ces deux cartes ont-elle la même couleur ? le même rang ? On utilisera l'opérateur binaire ET pour déterminer ces conditions.

Correction

(a) Pour obtenir la couleur d'une carte à partir de sa représentation binaire, on peut effacer (ou mettre à 0) les bits de poids faible $r_3 r_2 r_1 r_0$ qui servent à coder le rang de la carte.

On peut utiliser le fait que, quel que soit x , $x \text{ ET } 0 = 0$ pour mettre à 0 les bits du rang et que $x \text{ ET } 1 = x$ pour garder uniquement les bits de la couleur.

La couleur d'une carte dont la représentation binaire est C peut donc être obtenue avec l'opération suivante : $C \text{ ET } 110000$.

Les deux cartes ont donc la même couleur si $C_1 \text{ ET } 110000 = C_2 \text{ ET } 110000$.

(b) En utilisant le même raisonnement, on trouve que les deux cartes ont le même rang si $C_1 \text{ ET } 001111 = C_2 \text{ ET } 001111$. ♦

Exercice 5 — Robot photographe (Examen 2014–2015)

On imagine dans cet exercice un robot utilisé pour explorer et photographier une surface plane quelconque qu'on représente par une grille. Il est piloté à distance par un utilisateur humain qui peut lui envoyer différentes instructions :

- AV = avancer d'une case
- TG = se tourner de 90 degrés vers la gauche
- TD = se tourner de 90 degrés vers la droite
- LT = lever la tête
- BT = baisser la tête
- PP = prendre une photo

Q 5.1 Quel est le nombre minimal de bits nécessaires pour coder les instructions ?

Correction

Il y a 6 instructions à coder. 2 bits ne sont pas suffisants mais 3 suffisent : $4 = 2^2 < 6 \leq 2^3 = 8$. ♦

Q 5.2 Proposer un schéma de codage binaire des instructions.

Correction

- AV → 000
- TG → 001
- TD → 010
- LT → 011
- BT → 100
- PP → 101

On note que deux combinaisons binaires sont inutilisées : 110 et 111. ♦

Q 5.3 Donner dans ce schéma, la séquence binaire correspondant à la séquence d'instructions suivante : AV (4 ×), TG,

LT, PP.

Correction

000 000 000 000 001 011 101

Q 5.4 Ce schéma n'est pas adapté pour les cas où le robot doit parcourir de longues distances en ligne droite. Pourquoi ?

Correction

Car dans ce cas, l'utilisateur humain devra envoyer de longues séquences de 000.

Q 5.5 Proposer un nouveau schéma de codage binaire qui résolve le problème. On pourra utiliser un nombre variable de bits selon l'instruction codée.

Correction

On peut utiliser la combinaison binaire 110 qui n'était pas utilisée dans notre schéma de codage pour coder l'instruction :

— AP = avancer de plusieurs cases

Comme cette instruction ne précise pas le nombre de cases sur lequel le robot doit avancer, on fait suivre sa représentation binaire d'un certain nombre de bits, par exemple 4, qui indique le nombre de cases sur lequel on souhaite faire avancer le robot. L'instruction AP est donc codée sur 7 bits : les 3 premiers bits valent 110 et les 4 suivants indiquent le nombre de cases sur lequel on fait avancer le robot.

Q 5.6 Donner dans ce nouveau schéma la séquence binaire correspondant à la séquence d'instructions donnée à la question Q 5.3.

Correction

La séquence d'instructions est maintenant codée par la séquence binaire suivante :

110 0100 001 011 101

Les sept premiers bits indiquent que l'on fait avancer le robot sur 4 cases : 110 pour l'instruction AP et 0100 pour demander 4 cases. Le reste de la séquence est inchangé.

Exercice 6 — Chiffrement

Lorsque l'on envoie un message sur Internet et qu'on ne veut pas que ce message puisse être intercepté et compris par une personne autre que le destinataire on doit utiliser une méthode (ou un algorithme) de *chiffrement*. Le principe général est d'altérer le message selon une méthode connue de l'émetteur et du récepteur et d'envoyer sur le réseau le message ainsi modifié. Bien entendu, seul le récepteur doit être en mesure de déchiffrer le message reçu pour retrouver le message initial, c'est-à-dire avant sa transformation par l'opération de chiffrement.

Certains algorithmes de chiffrement reposent sur l'utilisation d'une *clé* qui est une séquence de bits quelconque connue uniquement de l'émetteur et du destinataire du message. Le message chiffré envoyé sur le réseau est obtenu en transformant le message initial à l'aide de cette clé (selon une méthode connue de l'émetteur et du récepteur) de même que le déchiffrement par le destinataire est obtenu en transformant le message chiffré à l'aide de cette clé.

Un algorithme de chiffrement très simple consiste à chiffrer le message en utilisant une clé c de 8 bits et en transformant chaque octet m du message par l'octet $m \mathbf{XOR} c$. Le destinataire effectue la même opération : il transforme chaque octet m' qu'il a reçu par l'octet $m' \mathbf{XOR} c$ pour retrouver l'octet du message initial.

Cette méthode peut naturellement être généralisée pour une longueur de clé quelconque, et pas uniquement de 8 bits. Dans cet exercice, on s'intéresse au chiffrement d'un texte codé en ASCII (donc avec des caractères codés sur 7 bits) à l'aide d'une clé de 7 bits.

Q 6.1 On veut envoyer le mot `yop`. Donner, à l'aide de la table ASCII donnée dans le cours, la séquence binaire correspondant à ce mot.

Correction

$$\begin{array}{rcl} y & = & \underbrace{111}_7 \underbrace{1001}_9 \\ o & = & \underbrace{110}_6 \underbrace{1111}_F \\ p & = & \underbrace{111}_7 \underbrace{0000}_0 \end{array}$$

La séquence binaire est donc 1111001 1101111 1110000.

Q 6.2 On utilise la clé 55. Quel sera alors le message envoyé après chiffrement (en binaire et en texte) ?

Correction

La clé en binaire vaut 0110111. Les trois mots de 7 bits obtenus après chiffrement sont :

- $1111001 \text{ XOR } 0110111 = 1001110$
- $1101111 \text{ XOR } 0110111 = 1011000$
- $1110000 \text{ XOR } 0110111 = 1000111$

Le message chiffré est donc 1001110 1011000 1000111 en binaire.

Pour obtenir le message chiffré en texte :

- $1001110_2 = 4E_{16}$ soit le caractère ASCII 'N'
- $1011000_2 = 58_{16}$ soit le caractère ASCII 'X'
- $1000111_2 = 47_{16}$ soit le caractère ASCII 'G'

Le texte chiffré est donc NXG.

Q 6.3 Vérifier qu'après déchiffrement, le destinataire retrouve bien le message initial.

Correction

Le déchiffrement donne :

- $1001110 \text{ XOR } 0110111 = 1111001$
- $1011000 \text{ XOR } 0110111 = 1101111$
- $1000111 \text{ XOR } 0110111 = 1110000$

On retrouve bien la séquence binaire initiale.

Q 6.4 Prouver à l'aide d'une table de vérité que le destinataire peut toujours retrouver le message initial à partir de la clé et du message chiffré.

Correction

Soit m un mot de 7 bits du message non chiffré et c la clé. Le mot chiffré est $m \text{ XOR } c$ et celui obtenu après déchiffrement est $(m \text{ XOR } c) \text{ XOR } c$. La méthode est donc correcte si $m = (m \text{ XOR } c) \text{ XOR } c$. La table de vérité ci-dessous montre que c'est toujours le cas.

m	c	$m \text{ XOR } c$	$(m \text{ XOR } c) \text{ XOR } c$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Intuitivement, un 1 dans la clé signifie qu'on inverse le bit dans le mot à chiffrer et un 0 signifie qu'on ne change pas ce bit. Dans les deux cas, appliquer deux fois l'opérateur XOR permet donc de retrouver le bit initial.

Q 6.5 En pratique, une longueur de clé de 7 bits est insuffisante. On utilise des clés d'au moins 32 ou 64 bits. Pourquoi ?

Correction

Avec une longueur de clé de 7 bits, il n'y a que $2^7 = 128$ clés possibles, ce qui fait que 128 tentatives sont suffisantes pour déchiffrer le message chiffré. Un programme de déchiffrement qui tente de déchiffrer le message en tentant avec toutes les clés possibles peut générer ces 128 solutions possibles en quelques milli-secondes. (Il faut bien entendu qu'il puisse analyser ensuite si le message déchiffré est bien un message correct.)

Avec une clé de 2^{32} bits, on monte à environ 4 milliards de clés possibles. C'est mieux mais avec un ordinateur relativement puissant, on peut espérer décoder le message en quelques heures.

TD 2 — Internet Protocol

Exercice 1 — Adresses et masques

Q 1.1 Parmi les propositions ci-dessous lesquelles correspondent à des adresses IP valides :

- (a) 10.0.0
- (b) 10.0.0.1.2
- (c) 10.0.0.1
- (d) 0.1.2.3
- (e) 192.600.1.1

Correction

Pour qu'une adresse IP notée avec la notation décimale pointée soit correcte, il faut qu'elle soit notée $X_1.X_2.X_3.X_4$ avec, pour tout i , X_i un octet sous forme décimale, donc dans l'intervalle $[0, \dots, 2^8 - 1] = [0, \dots, 255]$.

Les adresses correctes sont donc : 10.0.0.1 et 0.1.2.3. ♦

Q 1.2 Parmi les propositions ci-dessous lesquelles correspondent à des masques valides :

- (a) 255.0.0
- (b) 0.255.0.0
- (c) 255.0.0.255
- (d) 255.255.0.0
- (e) 255.240.0.0
- (f) 255.241.0.0

Correction

Un masque doit avoir la même forme qu'une adresse IP et, de plus, il doit être constitué (en binaire) d'une série de 1 à gauche suivie d'une série de 0 à droite. On rappelle que $255_{10} = 1111\ 1111$.

255.0.0.0 et 255.0.0.255 ne sont donc pas des masques corrects. Le premier n'est composé que de 3 octets. Le second est une suite de 8 bits à 0 suivis de 8 bits à 1 suivis de 16 bits à 0. On observe un problème similaire avec la troisième proposition. 255.255.0.0 est un masque correct composé de 16 bits à 1 suivis de 16 bits à 0.

Pour les deux dernières propositions il faut traduire leurs deuxièmes octets en binaire.

— $240_{10} = 1111\ 0000 \Rightarrow 255.240.0.0$ est bien un masque correct.

— $241_{10} = 1111\ 0001 \Rightarrow 255.241.0.0$ n'est pas un masque correct. ♦

Q 1.3 Donner pour les adresses IP ci-dessous le masque correspondant en notation décimale pointée.

- (a) 10.0.1.1/8
- (b) 192.168.1.1/16
- (c) 192.168.144.1/20

Correction

Peu importe l'adresse il suffit de regarder le nombre de bits à 1 apparaissant après le '/'.
 — /8 $\Rightarrow$ masque avec 8 bits à 1 suivis de 24 bits à 0 = 255.0.0.0
 — /16 $\Rightarrow$ masque avec 16 bits à 1 suivis de 16 bits à 0 = 255.255.0.0
 — /20 $\Rightarrow$ masque avec 20 bits à 1 suivis de 12 bits à 0. On a donc 2 octets codés 1111 1111 puis un octet codé 1111 0000 = 240_{10} , ce qui fait en notation décimale pointée 255.255.240.0. ♦

Q 1.4 Les adresses de réseau ci-dessous sont-elles cohérentes avec les masques indiqués ?

- (a) 20.0.0.0/16
- (b) 10.10.0.0/8
- (c) 192.168.0.0/10
- (d) 192.168.0.0/13

Correction

Pour qu'une adresse de réseau soit cohérente avec un masque il faut que pour tout bit à 1 dans cette adresse on trouve à la même position un bit à 1 dans le masque. Dans le cas contraire, cela signifie qu'un bit à 1 de l'adresse de réseau se trouve dans la partie host-id (les bits qui identifient les machines à l'intérieur du réseau) ce qui est impossible.

— 10.0.0.0/16 $\Rightarrow$ Seul le premier octet de l'adresse contient des 1. Elle est donc cohérente avec le masque qui est composé de deux octets à 1.

— 10.10.0.0/8 $\Rightarrow$ Les deux premiers octets de l'adresse contiennent des 1. Elle ne respecte donc pas le masque qui n'est composé que d'un octet à 1 suivi de trois octets à 0.

— 192.168.0.0/10 $\Rightarrow$ Le masque n'étant pas un multiple de 8 on doit traduire l'adresse et le masque en binaire :

	Décimal	Binaire
Adresse de réseau	192.168.0.0	11000000 10101000 00000000 00000000
Masque	/10	11111111 11000000 00000000 00000000

Pour les deux bits à 1 soulignés dans l'adresse de réseau, le bit à la même position dans le masque est à 0. L'adresse ne respecte donc pas le masque.

— 192.168.0.0/13 $\Rightarrow$ En traduisant en binaire on trouve :

	Décimal	Binaire
Adresse de réseau	192.168.0.0	11000000 10101000 00000000 00000000
Masque	/13	11111111 11111000 00000000 00000000

L'adresse respecte donc bien le masque. ♦

Exercice 2 — Adresses de réseau et de diffusion

Soient 4 machines M1 à M4 disposant d'interfaces réseau dont les adresses IP sont données dans le tableau ci-dessous :

Machine	Adresse IP
M1	11.12.13.14/8
M2	200.1.23.45/24
M3	140.43.205.1/20
M4	17.93.0.1/10

Q 2.1 Donner, pour chaque machine, l'adresse du réseau sur lequel elle se trouve.

Correction

Pour trouver l'adresse du réseau d'une machine on rappelle qu'il faut effectuer un ET logique entre le masque et l'adresse IP de la machine.

Machine	Adresse IP	Masque	Adresse de réseau
M1	11.12.13.14	255.0.0.0	$\begin{array}{r} 11 \ . \ 12 \ . \ 13 \ . \ 14 \\ \text{ET } 255 \ . \ 0 \ . \ 0 \ . \ 0 \\ \hline 11 \ . \ 0 \ . \ 0 \ . \ 0 \end{array}$
M2	200.1.23.45	255.255.255.0	$\begin{array}{r} 200 \ . \ 1 \ . \ 23 \ . \ 45 \\ \text{ET } 255 \ . \ 255 \ . \ 255 \ . \ 0 \\ \hline 200 \ . \ 1 \ . \ 23 \ . \ 0 \end{array}$
M3	140.43.205.1	255.255.240.0	$\begin{array}{r} 140 \ . \ 43 \ . \ 205 \ . \ 1 \\ \text{ET } 255 \ . \ 255 \ . \ 240 \ . \ 0 \\ \hline 140 \ . \ 43 \ . \ 192 \ . \ 0 \end{array}$
M4	17.93.0.1	255.192.0.0	$\begin{array}{r} 17 \ . \ 93 \ . \ 0 \ . \ 1 \\ \text{ET } 255 \ . \ 192 \ . \ 0 \ . \ 0 \\ \hline 17 \ . \ 64 \ . \ 0 \ . \ 0 \end{array}$

Pour M3 il faut convertir son troisième octet en binaire car les 4 premiers bits de cet octet forment la fin du net-id et les 4 suivants forment le début du host-id. Le troisième octet est $205 = \underline{1100} 1101_2$ (avec les bits du net-id soulignés). Dans l'adresse de réseau, le troisième octet vaut donc $\underline{1100} 0000_2 = 192$. Pour M4, il faut effectuer la même opération avec le deuxième octet. ♦

Q 2.2 Donner, pour chaque machine, l'adresse de diffusion du réseau sur lequel elle se trouve.

Correction

L'adresse de diffusion est obtenue en gardant les bits du net-id et en mettant ceux du host-id à 1.

Machine	Adresse de réseau	Adresse de diffusion
M1	11.0.0.0/8	11.255.255.255
M2	200.1.23.45/24	200.1.23.255
M3	140.43.205.1/20	140.43.207.255
M4	17.93.0.1/10	17.127.255.255

Par exemple, pour M1, les 8 premiers bits forment le net-id et les 24 suivants forment le host-id. Pour obtenir son adresse de diffusion il suffit donc de garder le premier octet puis de mettre les 3 derniers octets à 255. Le raisonnement est identique pour M2.

Pour M3 il faut convertir son troisième octet en binaire car les 4 premiers bits de cet octet forment la fin du net-id et les 4 suivants forment le début du host-id. Le troisième octet est $205 = \underline{1100} 1101_2$ (avec les bits du net-id soulignés). Dans l'adresse de diffusion, le troisième octet vaut donc $\underline{1100} 1111_2 = 207$. Pour M4, il faut effectuer la même opération avec le deuxième octet. ♦

Q 2.3 Donner, pour chaque machine, le nombre de machines que l'on peut avoir sur son réseau.

Correction

Le nombre de machines que l'on peut placer sur un réseau dépend du nombre de bits qui composent le host-id : c'est le nombre de séquences binaires que l'on peut former avec les bits du host-id auquel il faut retrancher les deux adresses réservées pour le réseau (adresse de réseau et adresse de diffusion). Ce qui donne :

Machine	Nombre de bits dans le host-id	Nombre de machines adressables
M1	$32 - 8 = 24$	$2^{24} - 2 = 16\,777\,214$
M2	$32 - 24 = 8$	$2^8 - 2 = 254$
M3	$32 - 20 = 12$	$2^{12} - 2 = 4094$
M4	$32 - 10 = 22$	$2^{22} - 2 = 4\,194\,302$

Q 2.4 L'administrateur du réseau de la machine M4 installe un nouvel ordinateur M5 sur ce réseau et attribue à son interface réseau l'adresse 17.147.255.255/10. Cette attribution est-elle correcte ?

Correction

Il faut que les deux machines aient le même masque (ce qui est le cas : /10) et que l'adresse du réseau de M5 obtenue à partir du masque soit la même que celle de M4 :

	17	.	147	.	255	.	255
ET	255	.	192	.	0	.	0
	17	.	128	.	0	.	0

Les adresses de réseau de M4 (17.64.0.0) et M5 (17.128.0.0) sont différentes $\Rightarrow$ l'attribution est incorrecte.

Q 2.5 Donner, pour le réseau de la machine M4, la plage des adresses IP que l'on peut attribuer aux machines de ce réseau.

Correction

La première adresse que l'on peut attribuer est, en binaire, 00010001 01000000 00000000 00000001 (avec les bits du net-id soulignés). Ce qui fait 17.64.0.1 en notation décimale pointée, soit l'adresse de réseau + 1.

La dernière adresse que l'on peut attribuer est, en binaire, 00010001 01111111 11111111 11111110. Ce qui fait 17.127.255.254 en notation décimale pointée, soit l'adresse de diffusion - 1.

Toute adresse dans cette plage peut-être attribuée à une machine du réseau de M4.

Exercice 3 — Lecture de tables de routage

Q 3.1 Soit la table de routage ci-dessous obtenue avec la commande route que l'on exécute sur une machine M.

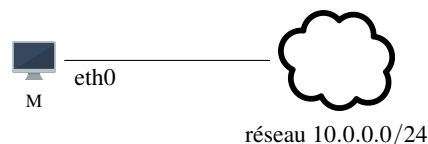
1	\$ route -n							
2	Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
3	10.0.0.0	*	255.255.255.0	U	0	0	0	eth0

Les colonnes qui nous intéressent dans cette exercice sont : Destination, Passerelle (appelée Routeur dans le cours), Genmask (Masque) et Iface (Interface).

La colonne Indic donnent l'état de la ligne. Un U (pour *up*) dans cette colonne signifie que la route est active. Un G (voir questions suivantes) signifie que la ligne correspond à une route en remise indirecte et donc que la colonne Passerelle n'est pas vide et contient l'adresse IP d'un routeur (G = Gateway qui se traduit par passerelle).

Les autres colonnes (Metric, Ref et Use) fournissent des statistiques qui peuvent être utilisées par des protocoles de routage dynamique.

(a) Dessiner la topologie du réseau connue de M en indiquant les différents réseaux et routeurs connus de M.

Correction

(b) Quel sera le résultat de la consultation de cette table si M souhaite envoyer un paquet à l'adresse 10.0.0.1 ?

Correction

L'unique ligne de la table indique comment envoyer des paquets au réseau 10.0.0.0/24. La machine d'adresse 10.0.0.1 est bien sur ce réseau (car 10.0.0.1 **ET** 255.255.255.0 = 10.0.0.0). La colonne Passerelle de cette ligne est vide, ce qui signifie que M est directement connectée au réseau 10.0.0.0/24 (il n'y a pas de routeur entre M et ce réseau). La colonne Iface indique l'interface réseau à utiliser pour envoyer des paquets sur le réseau 10.0.0.0/24 : eth0. La consultation donne donc la réponse suivante : *remise directe en utilisant l'interface eth0*.

(c) Quel sera le résultat de la consultation de cette table si M souhaite envoyer un paquet à l'adresse 1.2.3.4 ?

Correction

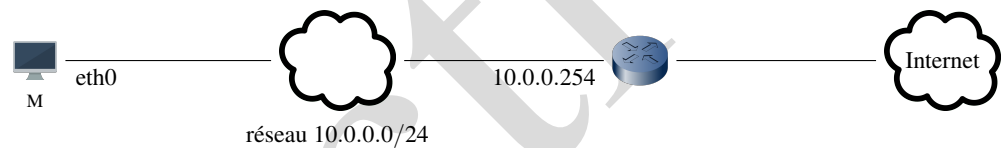
L'unique ligne de la table indique comment envoyer des paquets au réseau 10.0.0.0/24. La machine d'adresse 1.2.3.4 n'étant pas sur ce réseau (car 1.2.3.4 ET 255.255.255.0 = 1.2.3.0), la consultation donnera une réponse négative : *hôte inaccessible*.

Q 3.2 Soit la table de routage ci-dessous obtenue avec la commande route que l'on exécute sur une machine M.

1	\$ route -n							
2	Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
3	10.0.0.0	*	255.255.255.0	U	0	0	0	eth0
4	0.0.0.0	10.0.0.254	0.0.0.0	UG	1024	0	0	eth0

(a) Dessiner la topologie du réseau connue de M en indiquant les différents réseaux et routeurs connus de M.

Correction



(b) Quel sera le résultat de la consultation de cette table si M souhaite envoyer un paquet à l'adresse 1.2.3.4 ?

Correction

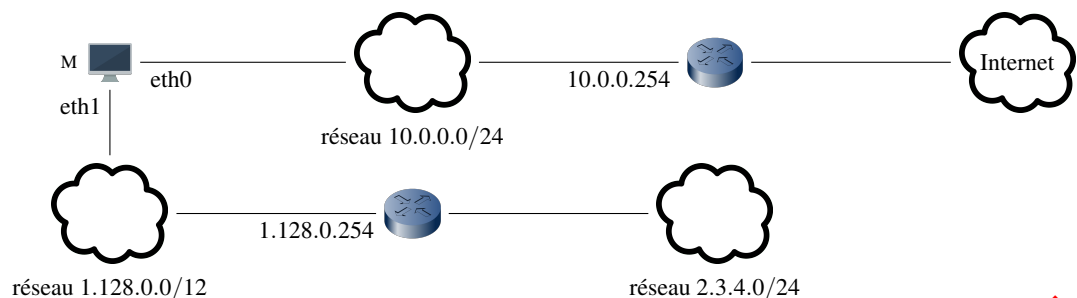
Comme dans le cas précédent la machine d'adresse 1.2.3.4 n'est pas sur le réseau 10.0.0.0/24. Par contre, la table a une route par défaut qui indique comment joindre tout réseau autre que 10.0.0.0/24. La colonne Passerelle de cette ligne contient l'adresse du routeur qui permet de joindre ces réseaux : 10.0.0.254 ; et la colonne Iface indique l'interface sur laquelle le paquet doit être envoyé pour joindre 10.0.0.254 : eth0. La consultation donne donc la réponse suivante : *remise indirecte à 10.0.0.254 en utilisant l'interface eth0*.

Q 3.3 Soit la table de routage ci-dessous obtenue avec la commande route que l'on exécute sur une machine M.

1	\$ route -n							
2	Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
3	0.0.0.0	10.0.0.254	0.0.0.0	UG	1024	0	0	eth0
4	10.0.0.0	*	255.255.255.0	U	0	0	0	eth0
5	1.128.0.0	*	255.240.0.0	U	0	0	0	eth1
6	2.3.4.0	1.128.0.254	255.255.255.0	UG	1024	0	0	eth1

(a) Dessiner la topologie du réseau connue de M en indiquant les différents réseaux et routeurs connus de M.

Correction



(b) Quel sera le résultat de la consultation de cette table si M souhaite envoyer un paquet à l'adresse 2.3.4.5 ?

Correction

La table contient une ligne avec le réseau de destination 2.3.4.0/24. La machine d'adresse 2.3.4.5 est bien sur ce réseau (car 2.3.4.5 ET 255.255.255.0 = 2.3.4.0). La colonne Passerelle de cette ligne contient l'adresse du routeur qui permet de joindre le réseau 2.3.4.0/24 : 1.128.0.254 ; et la colonne Iface indique l'interface sur laquelle le paquet doit être envoyé pour joindre 1.128.0.254 : eth1. La consultation donne donc la réponse suivante : *remise indirecte à 1.128.0.254 en utilisant l'interface eth1*.

- (c) L'administrateur du réseau rajoute la ligne (D=5.6.7.0, M=255.255.255.0, R=1.130.0.254, I=eth1). Cette nouvelle ligne est-elle cohérente avec la table ?

Correction

L'interface eth1 est connectée au réseau 1.128.0.0/12. Pour que cette nouvelle ligne soit cohérente avec la table, il faut que R soit sur ce réseau. Or, 1.130.0.254 **ET** 255.240.0.0 = 1.128.0.0. R est donc bien sur le réseau 1.128.0.0. Cette nouvelle ligne est donc correcte. ♦

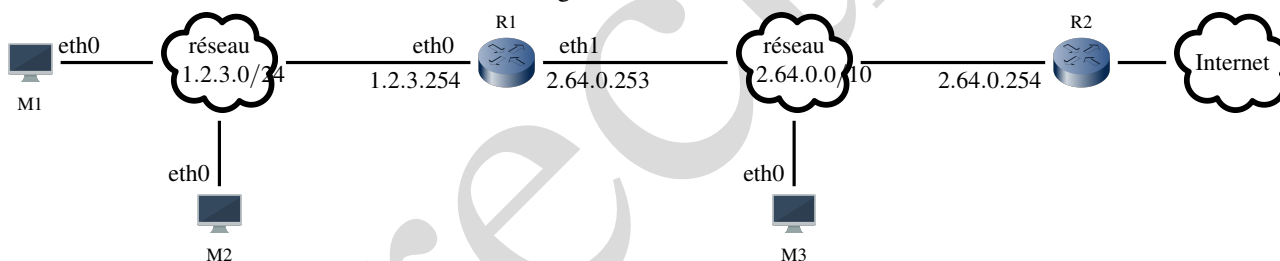
- (d) L'administrateur du réseau rajoute la ligne (D=6.7.8.0, M=255.255.255.0, R=10.0.0.254, I=eth0). Que remarque-t-on ?

Correction

On remarque que cette ligne indique une remise indirecte vers le réseau 6.7.8.0/24 avec le même routeur et la même interface que pour la route par défaut. Cette ligne est donc redondante car pour envoyer un paquet vers le réseau 5.6.7.0/24 il suffit d'utiliser la route par défaut. ♦

Exercice 4 — Écriture de tables de routage

On considère dans cet exercice le réseau de la figure ci-dessous.



Q 4.1 Donner le contenu de la table de routage de M1.

Correction

Destination	Masque	Routeur	Interface
1.2.3.0	255.255.255.0	-	eth0
0.0.0.0	0.0.0.0	1.2.3.254	eth0

Remarque. Il est inutile d'ajouter une route vers le réseau 2.64.0.0/10, celui-ci étant déjà accessible via la route par défaut. ♦

Q 4.2 Donner le contenu de la table de routage de M3.

Correction

Destination	Masque	Routeur	Interface
2.64.0.0	255.192.0.0	-	eth0
1.2.3.0	255.255.255.0	2.64.0.253	eth0
0.0.0.0	0.0.0.0	2.64.0.254	eth0

Remarque. Le masque correspondant à /10 en notation décimale pointée est 255.192.0.0. ♦

Q 4.3 Donner le contenu de la table de routage de R1.

Correction

Destination	Masque	Routeur	Interface
1.2.3.0	255.255.255.0	-	eth0
2.64.0.0	255.192.0.0	-	eth1
0.0.0.0	0.0.0.0	2.64.0.254	eth1

Q 4.4 M1 veut envoyer un paquet à 2.160.0.1. Détailler les opérations effectuées par M1 et R1 pour que le paquet arrive à destination.

Correction

- M1 consulte sa table de routage. 2.160.0.1 n'est pas sur le réseau 1.2.3.0/24. Le paquet va donc suivre la route par défaut de M1. M1 envoie donc le paquet sur son interface eth0 à destination de 1.2.3.254 (eth0 de R1).
 - R1 réceptionne le paquet sur eth0. Le paquet ne lui est pas destiné (l'adresse de destination du paquet ne correspond à aucune des adresses associées à ses interfaces). R1 consulte sa table de routage.
- La première ligne de la table de routage de R1 indique comment joindre le réseau 1.2.3.0/24. L'adresse de destination n'est pas sur ce réseau (car 2.160.0.1 **ET** 255.255.255.0 = 2.160.0.0 $\neq$ 1.2.3.0).

La deuxième ligne de la table de routage de R1 indique comment joindre le réseau 2.64.0.0/10. Le destinataire n'est pas sur ce réseau (car $2.160.0.1 \text{ ET } 255.192.0.0 = 2.128.0.0$).

Le paquet va donc suivre la route par défaut de R1. R1 envoie donc le paquet sur son interface eth1 à destination de 2.64.0.254 (une interface de R2). ♦

Q 4.5 Même question avec un paquet destiné à 2.100.0.1.

Correction

Le traitement effectué par M1 ne change pas.

Par contre, quand R1 va regarder sa table de routage, la deuxième ligne de sa table de routage de R1 est bien la route permettant de joindre le destinataire puisque $2.100.0.1 \text{ ET } 255.192.0.0 = 2.64.0.0$. R1 enverra donc le paquet à 2.100.0.1 (remise directe) sur son interface eth1. ♦

Q 4.6 Supposons que R1 reçoive un paquet à destination de 2.X.Y.Z. Quelle sera la décision de routage en fonction de X, Y et Z ?

Correction

Il faut voir le résultat de $2.X.Y.Z \text{ ET } 255.192.0.0$. Les valeurs de Y et Z n'ont pas d'importance puisque l'octet correspondant dans le résultat de l'opération sera toujours 0.

Si X est dans l'intervalle [64, 127] alors R1 enverra ce paquet en remise directe sur 2.64.0.0/10. Sinon, il l'enverra sur la route par défaut. En effet tous les entiers de cet intervalle ont une représentation binaire de la forme 01xx xxxx. En faisant le ET logique on trouvera donc toujours $X \text{ ET } 192 = 01xxxxx_2 \text{ ET } 11000000_2 = 01000000_2 = 64$. Donc toute adresse de la forme 2.X.Y.Z avec $X \in [64, 127]$ est dans le réseau 2.64.0.0/10. ♦

TD 3 — Services DNS et DHCP

Exercice 1 — DNS

Q 1.1 Rappeler le rôle d'un serveur DNS.

Correction

Donner aux clients les adresses IP correspondant à des noms de machine symboliques (ou noms DNS). C'est ce qu'on appelle la *résolution de noms*. ♦

Q 1.2 On ouvre un navigateur Web à l'URL ci-dessous :

`http://www.iut.edu/rep/fichier.html`

(a) Quels seront les protocoles de service utilisés par le navigateur ?

Correction

DNS pour résoudre `www.iut.edu` en adresse IP et HTTP pour le téléchargement du fichier `/rep/fichier.html`. ♦

(b) Quels seront les messages échangés entre notre machine et les autres serveurs impliqués ?

Correction

Ici on veut récupérer le fichier `/rep/fichier.html` depuis le serveur `www.iut.edu`. Comme on ne connaît pas directement l'adresse IP de ce serveur on doit la demander à un serveur DNS (dont l'adresse IP se trouve dans le fichier `/etc/resolv.conf`). On observera donc l'échange suivant :

- i. notre machine → serveur DNS : quelle est l'adresse IP de la machine appelée `www.iut.edu` ?
- ii. serveur DNS → notre machine : l'adresse IP de `www.iut.edu` est `1.2.3.4`.
- iii. notre machine → machine `1.2.3.4` : je veux le fichier `/rep/fichier.html`
- iv. machine `1.2.3.4` → notre machine : voilà le fichier `/rep/fichier.html`

En réalité, on pourra avoir plusieurs messages DNS à la place des messages 1 et 2 selon le nombre de serveurs DNS interrogés.

Le fichier `/rep/fichier.html` est ensuite affiché dans le navigateur. ♦

Q 1.3 On suppose que le serveur DNS interrogé par le navigateur est celui de la zone `p13.fr`, quels seront alors les serveurs DNS qui seront interrogés par le client en supposant qu'à chaque domaine correspond une zone d'autorité ?

Correction

Le serveur DNS de `p13.fr` va renvoyer sur un serveur racine (de la zone `.`). On va ensuite redescendre dans l'arbre. Le serveur racine va retourner au client l'adresse IP du serveur DNS de la zone `edu`. Le serveur de la zone `edu` va envoyer au client l'adresse IP du serveur DNS de la zone `iut.edu`. Ce dernier connaîtra l'adresse IP de `www.iut.edu` et la renverra au client. ♦

Q 1.4 On suppose que le serveur DNS utilisé par la machine est en panne. Comment l'utilisateur peut-il tout de même récupérer le fichier de la question précédente ?

Correction

Il faut qu'il connaisse l'adresse IP de la machine `www.iut.fr`. En supposant que cette adresse est `1.2.3.4`, le fichier pourra être téléchargé en ouvrant le navigateur à l'URL ci-dessous :

`http://1.2.3.4/rep/fichier.html`

♦

Q 1.5 Soit l'extrait du fichier de zone de `iut.edu` placé sur la machine `dns.iut.edu`.

1	<code>iut.edu.</code>	NS	<code>dns</code>
2	<code>dns</code>	A	<code>10.1.0.1</code>
3	<code>www</code>	A	<code>10.1.0.2</code>
4	<code>mail</code>	A	<code>10.1.0.3</code>
5	<code>max.etud</code>	A	<code>10.2.0.100</code>
6	<code>anna.etud</code>	A	<code>10.2.0.200</code>

Ce fichier se trouve sur le serveur faisant autorité sur la zone `iut.edu` qui contient les domaines `iut.edu` et `etud.iut.edu`. Il contient principalement les adresses IP des machines de cette zone. Chaque ligne de ce fichier contient un enregistrement ayant la forme suivante :

`<nom> <type> <valeur>`

Le nom identifie une machine ou un domaine. Il peut être absolu (terminé par un point) ou relatif (par rapport au nom de la zone). Plusieurs enregistrements peuvent être associés à un nom. On les différencie par un type. La

valeur de l'enregistrement dépend alors du type. Dans notre exemple il n'y a que deux types : A pour définir une adresse IP et NS (Name Server) pour définir un serveur DNS d'une zone.

(a) Quels sont les noms complets des machines dans la zone `iut.edu` dont le serveur connaît les adresses IP ?

Correction

Tous les noms sont donnés sous forme relative, il faut donc compléter par `iut.edu` : `dns.iut.edu`, `www.iut.edu`, `max.etud.iut.edu`,...

(b) On souhaite que le domaine `etud.iut.edu` soit délégué à un autre serveur DNS, appelé `dns.etud.iut.edu` et d'adresse IP 10.2.0.1. Autrement dit, on souhaite découper la zone `iut.edu` et créer une sous-zone `etud.iut.edu`. Comment doit-on procéder ?

Correction

On doit d'abord créer le fichier de la zone `etud.iut.edu` sur la machine `dns.etud.iut.edu`. Ce fichier devra contenir les adresses IP des machines du domaine `etud.iut.edu` :

```
1 etud.iut.edu. NS dns
2 dns A 10.2.0.1
3 max A 10.2.0.100
4 anna A 10.2.0.200
```

On doit ensuite modifier le fichier de la zone `iut.edu` afin qu'il ne contienne plus les adresses IP des machines du domaine `etud.iut.edu` et qu'il indique `dns.etud.iut.edu` comme serveur DNS pour la zone `etud.iut.edu` :

```
1 iut.edu. NS dns
2 dns A 10.1.0.1
3 www A 10.1.0.2
4 mail A 10.1.0.3
5 etud NS dns.etud
6 dns.etud A 10.2.0.1
```

(c) Quel sera alors la réponse du serveur `dns.iut.edu` si on l'interroge sur la machine `anna.etud.iut.edu` ?

Correction

`anna.etud.iut.edu` n'est plus dans sa zone d'autorité. Il répondra négativement et renverra l'adresse IP du serveur DNS qui fait autorité sur `etud.iut.edu` : 10.2.0.1.

Exercice 2 — DHCP

Q 2.1 Rappeler le rôle d'un serveur DHCP et les informations qu'il peut fournir à des clients.

Correction

Le rôle d'un serveur DHCP :

- (a) attribuer des adresses IP aux clients. Ces adresses seront utilisables par le client uniquement pendant un certain laps de temps (le *bail* DHCP). Passé ce délai, le client devra redemander au serveur si il peut continuer à utiliser l'adresse IP fournie.
- (b) fournir aux clients des informations sur le réseau sur lequel il se trouve :
 - l'adresse du réseau et son masque
 - l'adresse du routeur qui connecte le réseau au monde extérieur
 - l'adresse IP d'un serveur DNS
 (Le serveur DHCP peut fournir d'autres informations mais celles qu'on donne ici sont les plus importantes pour que les clients puissent accéder à Internet.)

Q 2.2 On connecte un ordinateur portable au réseau de l'IUT. Comment peut-on alors accéder à Internet si le serveur DHCP du réseau est en panne ?

Correction

Il faut déjà connaître l'adresse et le masque du réseau sur lequel on se trouve. On choisit une adresse IP dans ce réseau (adresse qui ne doit pas être utilisée par une autre machine) et on l'attribue à notre machine en modifiant les fichiers de configuration (voir TP 1). Ensuite, il faut connaître l'adresse IP du routeur qui nous permet de sortir du réseau et modifier la table de routage en conséquence avec la commande `route`. Enfin, il faut connaître l'adresse IP d'un serveur DNS et l'inscrire dans le fichier `/etc/resolv.conf`.

Q 2.3 On considère le fichier ci-dessous.

```
1 # déclaration d'un réseau avec adresse et masque
2 subnet 10.1.2.0 netmask 255.255.255.0 {
3
```

```

4  # adresse du routeur qui mène à l'extérieur
5  option routers 10.1.2.254;
6
7  # adresse du serveur DNS donnée aux clients
8  option domain-name-servers 10.1.2.200;
9
10 # plage des adresses que le serveur peut attribuer aux clients
11 range 10.1.2.1 10.1.2.100;
12 }

```

Celui-ci est un fichier de configuration très simple d'un serveur DHCP. Il contient les informations que le serveur peut fournir à un client qui en fait la demande. Le caractère # permet de commencer un commentaire qui se termine à la fin de la ligne.

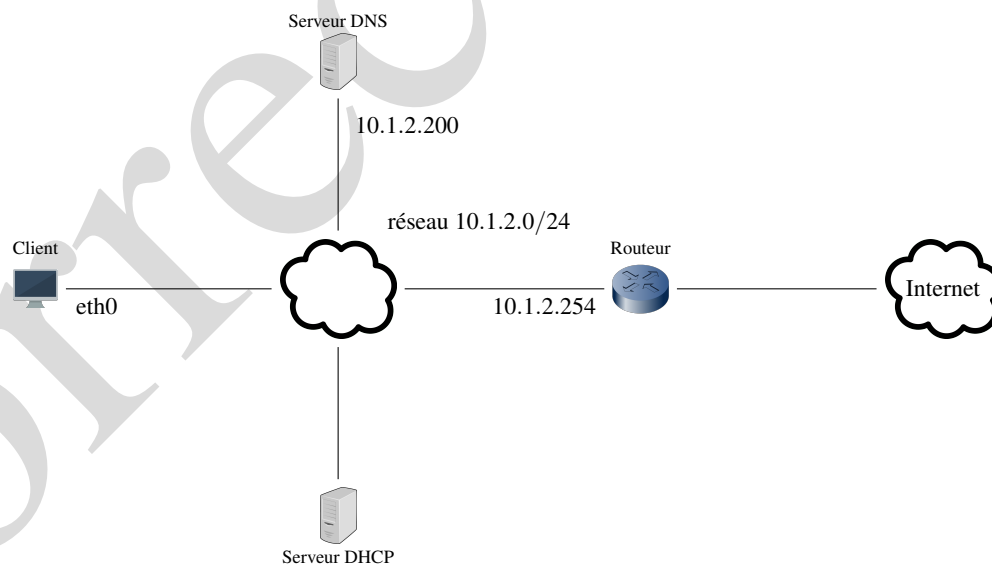
- (a) Combien peut-on avoir au maximum de machines clientes connectées à Internet sur ce réseau ? Que se passera-t-il si davantage de clients essaient de se connecter ?

Correction

Le serveur peut attribuer au maximum 100 adresses IP (les adresses dans l'intervalle indiqué par la ligne range). Si 100 machines ont déjà reçu une adresse IP et qu'une 101^{ème} en fait la demande auprès du serveur DHCP, celui-ci refusera de lui accorder un bail.

- (b) On suppose qu'un client C obtient un bail auprès de ce serveur DHCP. La demande de bail a été envoyée par C sur son interface eth0. Dessiner la topologie du réseau connue par C après l'obtention du bail et donner sa table de routage.

Correction



Destination	Masque	Routeur	Interface
10.1.2.0	255.255.255.0	-	eth0
0.0.0.0	0.0.0.0	10.1.2.254	eth0

- (c) Que contiendra le fichier /etc/resolv.conf après l'obtention du bail par le client C de la question précédente ?

Correction

Ce fichier contient l'adresse IP du serveur DNS utilisé par tous les processus du système pour faire de la résolution DNS. Il contiendra donc l'adresse 10.1.2.200. La ligne exacte dans le fichier sera :

```
1 nameserver 10.1.2.200
```