

SESSION 2022

---

# AGREGATION CONCOURS EXTERNE

Section : INFORMATIQUE

COMPOSITION EN INFORMATIQUE

Durée : 5 heures

---

*L'usage de tout ouvrage de référence, de tout dictionnaire et de tout matériel électronique (y compris la calculatrice) est rigoureusement interdit.*

*Si vous repérez ce qui vous semble être une erreur d'énoncé, vous devez le signaler très lisiblement sur votre copie, en proposer la correction et poursuivre l'épreuve en conséquence. De même, si cela vous conduit à formuler une ou plusieurs hypothèses, vous devez la (ou les) mentionner explicitement.*

**NB : Conformément au principe d'anonymat, votre copie ne doit comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé consiste notamment en la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de la signer ou de l'identifier.**

Tournez la page S.V.P.

A

## INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie.

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

Concours	Section/option	Epreuve	Matière
EAE	6200A	101	9422

**Dépendances.** Ce sujet contient quatre parties indépendantes qui doivent être traitées toutes les quatre. On veillera à bien indiquer sur la copie les changements de partie.

**Attendus.** Il est attendu des candidates et des candidats des réponses construites. Ils seront aussi évalués sur la précision, le soin et la clarté de la rédaction.

## Partie 1 : Système

**Exercice 1** Droits et permissions d'accès aux fichiers.

Ci-dessous, le résultat de la commande linux `ls -al` en ligne de commande sur une installation linux standard :

```
[untel:~/Ag] ls -al
total 36
drwxrwxr-x  3 untel guntel  4096 nov.  7 18:10 .
drwxr-xr-x 75 untel guntel 20480 nov.  7 20:04 ..
lrwxrwxrwx  1 untel guntel    6 nov.  7 18:08 fc.txt -> fi.txt
-rw-rw-r--  1 untel guntel  153 nov.  7 18:10 fi.txt
-rwxrw-r--  1 untel otel   83 nov.  7 18:10 oui.sh
drwxrw-r--  2 untel guntel  4096 nov.  7 18:08 sAg
```

- Quel est l'utilisateur qui utilise le terminal ?
- Quel est le propriétaire des fichiers ?
- Que sont `guntel` et `otel` ?
- Quel est le nom du répertoire courant ?
- Combien contient-il de fichiers ? de répertoires ?
- Pour `oui.sh` et `fc.txt`, expliquer les informations affichées. Des réponses précises sont demandées pour les colonnes 1, 3, 4 ainsi que la dernière.
- Quelles sont les opérations liées à `sAg` que peut faire un utilisateur dont l'identifiant est `trit` (précisez les différentes possibilités) ? Par exemple, que se passe-t-il si, à partir du répertoire `Ag`, `trit` tape la commande `cd sAg`.
- Pourquoi certains systèmes d'exploitation ont-ils une politique de gestion des droits et de permissions d'accès aux fichiers ?

**Exercice 2** Système de Gestion de Fichiers.

On considère un système de gestion de fichiers `ext2` avec une taille de bloc de 2048 octets. Les caractéristiques d'un inode de cet `ext2` sont :

- les 12 (de 0 à 11) premiers champs pointent sur un bloc de données,
  - le champ 12 pointe vers 256 blocs de données (simple indirection),
  - le champ 13 pointe vers  $256^2$  blocs de données (double indirection),
  - le champ 14 pointe vers  $256^3$  blocs de données (triple indirection),
  - le champ longueur : 32 bits, mais le premier bit de poids fort n'est pas utilisé.
- Quelle est la taille maximum d'un fichier sur ce système de gestion de fichiers ?
  - Quel est le nombre maximum de blocs ?

- c. Soit un fichier contenant 4567 octets.  
Combien de blocs occupe-t-il ?
- d. Même question pour un fichier de 573448 octets. Parmi ces blocs, combien y a-t-il de blocs directs, indirects, double indirects et triple indirects ?

### Exercice 3 Performance

Voici un extrait d'article de Roberto Di Cosmo, intitulé *Piège dans le Cyberspace*, mis en ligne le 20 mars 1998.

*“Imaginons maintenant un ministère qui garde ses dossiers dans une énorme armoire avec des millions de tiroirs : on aimerait bien, pour les mêmes raisons qu’avant, que les documents afférents à un même dossier se trouvent dans la mesure du possible rangés dans des tiroirs contigus. Vous devez embaucher une secrétaire et vous avez le choix entre deux candidates aux pratiques assez différentes : la première, quand un dossier est bouclé se limite à vider les tiroirs, et quand un nouveau dossier arrive elle le sépare en petits groupes de documents de la taille d’un tiroir, et range chaque groupe au hasard dans le premier tiroir vide qu’elle trouve dans l’armoire. Lorsque vous lui faites remarquer que ça risque alors d’être bien difficile de retrouver vite tous les documents du dossier du Crédit Lyonnais, elle répond qu’il faut engager tous les week-ends une dizaine de garçons pour tout remettre en ordre. La deuxième secrétaire, par contre, conserve sur son bureau une liste des tiroirs vides contigus, qu’elle met à jour toutes les fois qu’un dossier est clos et qu’on l’enlève des tiroirs ; quand un nouveau dossier arrive, elle cherche dans sa liste une suite de tiroirs vides contigus de taille suffisante et c’est là qu’elle place le nouveau dossier. Ainsi, vous explique-t-elle, s’il y a assez de mouvement, l’armoire restera toujours très bien rangée.”*

Expliquer par une description la plus précise possible et des exemples précis à quelle problématique et à quelles différentes technologies informatiques fait référence cet extrait d'article. Pour cela votre réponse devra au moins inclure des éléments concernant les questions suivantes :

- Que représente *l’énorme armoire du ministère* ?
- Que représentent les *dossiers à ranger dans cette armoire* ?
- Pourquoi *séparer un dossier en petits groupes de documents* ?
- Que représente la *dizaine de garçons qui remettent tout en ordre* ?
- Que représente la *liste des tiroirs vides mise à jour par la deuxième secrétaire*.
- Quelle est la méthode la plus performante ?
- Si vous en connaissez, donnez des exemples concrets correspondants aux deux méthodes décrites.

### Exercice 4 Processus.

- a. 1. Que fait le programme C suivant ? :

```
int main(void)
{
    for (;;)
        fork();
    return 0;
}
```

- 2. Est-ce un problème ? Si oui, donner une solution pour l'éviter.
- b. 1. Soit la fonction C qui calcule  $x^n$  de façon récursive en créant un fils qui va calculer  $x^{n/2}$  et le reste par le père.

```

int calcul (int x, int n)
{
    int a, b, pid;
    int p = n / 2;
    int fds[2];
    if (n == 0) return 1;
    if (n == 1)
    {
        return x;
    }
    if (pipe (fds) == -1)
    {
        perror ("Erreur creation tube");
        exit (1);
    }
    pid = fork ();
    switch (pid)
    {
        case -1:
            perror ("Erreur creation processus fils");
            exit (1);
        case 0: // fils
            close (fds[0]);
            a = calcul (x, p);
            write (fds[1], &a, sizeof (int));
            exit (0);
        default: // pere
            close (fds[1]);
            read (fds[0], &a, sizeof (int));
            b = calcul (x, n - p);
            wait (NULL);
    }
    return (a * b);
}

```

Modifier la fonction en créant deux processus fils qui calculent respectivement  $x^{n/2}$  et  $x^{n-n/2}$  (la division étant entière), le père calculant le produit et ainsi de suite récursivement. On considère le cas où l'on fait le calcul dans deux fils dans tous les cas, même si  $n$  est pair.

**Remarque et données :** l'exactitude de la syntaxe C n'est pas le critère principal pour cette question. Voici les signatures des principales fonctions de gestion de processus utilisées ici :

```

pid_t fork(void);
int pipe(int pipefd[2]);
pid_t wait(int *status);
ssize_t read(int fd, void *buf, size_t count);
ssize_t write(int fd, const void *buf, size_t count);
int close(int fd);

```

2. Que pouvez-vous dire de l'efficacité de ces deux fonctions calculant  $x^n$  ?



**Question 2.** Donner la représentation binaire des valeurs réelles :

- a. 1
- b.  $-1$
- c.  $2^{-53}$
- d.  $1 + 2^{-50}$

**Question 3.** On considère un nombre flottant positif. On peut lire sa représentation mémoire comme étant celle d'un entier non signé 64 bits. Par exemple,  $0 \mid 0 \cdots 0 \mid 0 \cdots 01$  serait interprété par 1 et  $0 \mid 0 \cdots 01 \mid 0 \cdots 0$  par  $2^{53}$ . On considère la fonction  $\mathcal{S}$  qui interprète un nombre flottant comme un entier non signé 64 bits, lui ajoute 1 et le réinterprète comme un nombre flottant. Soit  $x$  de signe  $s$ , d'exposant  $E$  et de fraction  $f$ .

- a. Supposons  $s = 0$ ,  $E = 0$  et  $f \neq 1 \cdots 1$ ; que vaut  $\mathcal{S}(x)$  ?
- b. Supposons  $s = 0$ ,  $1 \leq E \leq 2^{11} - 2$  et  $f \neq 1 \cdots 1$ ; que vaut  $\mathcal{S}(x)$  ?
- c. Supposons  $s = 0$ ,  $E = 0$  et  $f = 1 \cdots 1$ ; que vaut  $\mathcal{S}(x)$  ?
- d. Supposons  $s = 0$ ,  $0 \leq E \leq 2^{11} - 2$  et  $f \neq 1 \cdots 1$ ; que vaut  $\mathcal{S}(x)$  ?
- e. Supposons  $s = 0$ ,  $0 \leq E \leq 2^{11} - 2$  et  $f = 1 \cdots 1$ ; que vaut  $\mathcal{S}(x)$  ?
- f. Conclure sur  $\mathcal{S}$ .

**Question 4.** Dans la suite, on admet (sans le démontrer) que les flottants sont également l'ensemble des  $n2^e$  avec  $n$  et  $e$  entiers,  $|n| < 2^{53}$  et  $-1074 \leq e \leq 970$ .

- a. Prouver que  $0$ ,  $1$ ,  $1/8$ ,  $2^{52}$  et  $-2^{52}$  sont des flottants selon cette caractérisation.
- b. Donner le plus grand nombre flottant et le plus petit nombre flottant strictement positif.
- c. Donner tous les nombres flottants dans l'intervalle  $[1 - 2^{-52}, 1 + 2^{-52}]$ .

Dans la suite du problème, on s'intéresse aux effets de l'arrondi sur les opérations entre flottants. On note  $\mathbb{F}$  l'ensemble des flottants. On voit  $\mathbb{F}$  comme une partie de  $\mathbb{R}$ , l'ensemble des nombres réels. Ainsi, un flottant peut être vu, au besoin, comme un réel.

Il existe plusieurs modes d'arrondi. On considère ici l'arrondi par défaut, qui est l'arrondi au plus proche (en anglais *round to nearest*). Il s'agit de la fonction  $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{F}$  qui, pour un réel  $x$ , renvoie le nombre flottant le plus proche de  $x$ . En cas de point milieu (si  $x$  est à égale distance de deux nombres flottants),  $\mathcal{N}(x)$  est celui qui a la fraction paire (finit par un zéro au sens de l'introduction de cet exercice).

La spécification des flottants garantit que l'addition sur  $\mathbb{F}$ , qu'on note  $\oplus$ , se comporte comme si le calcul avait d'abord été fait dans  $\mathbb{R}$  de façon exacte puis qu'on avait arrondi le résultat avec  $\mathcal{N}$ . On a donc la relation

$$\forall x \in \mathbb{F}, \forall y \in \mathbb{F}, \quad x \oplus y = \mathcal{N}(x + y).$$

On a des relations similaires pour la soustraction  $\ominus$  et la multiplication  $\otimes$ .

**Question 5.** Calculer  $(-2^{52} \oplus 2^{52}) \oplus 1/8$ . Calculer  $-2^{52} \oplus (2^{52} \oplus 1/8)$ . Conclure sur une propriété non respectée par  $\oplus$ .

**Question 6.** Trouver un nombre flottant  $x$  non nul tel que  $x \otimes x = 0$ .

Que penser donc du code Python suivant :

```
if (x != 0):  
    z=1/(x*x)
```

**Question 7.**

- a. Trouver un nombre flottant  $x$  tel que  $1 \oplus x = 1$ .
- b. Trouver le plus grand nombre flottant  $x$  tel que  $1 \oplus x = 1$ . Justifier.

**Question 8.** Dans cette question, on considère l'arrondi vers  $-\infty$ , noté  $\mathcal{D}$  (en anglais *round down*) : pour tout réel  $x$ ,  $\mathcal{D}(x)$  est le plus grand nombre flottant inférieur ou égal à  $x$ .

- a. Trouver un nombre flottant  $x$  tel que  $\mathcal{D}(1 + x) = 1$ .
- b. Trouver le plus grand nombre flottant  $x$  tel que  $\mathcal{D}(1 + x) = 1$ . Justifier.

**Question 9.** Dans cette question, on considère l'arrondi vers  $+\infty$ , noté  $\mathcal{U}$  (en anglais *round up*) : pour tout réel  $x$ ,  $\mathcal{U}(x)$  est le plus petit nombre flottant supérieur ou égal à  $x$ .

- a. Trouver un nombre flottant  $x$  tel que  $\mathcal{U}(1 + x) = 1$ .
- b. Trouver le plus grand nombre flottant  $x$  tel que  $\mathcal{U}(1 + x) = 1$ . Justifier.

**Question 10.** Si le résultat d'une addition est suffisamment petit, alors il n'y a pas eu d'erreur d'arrondi. Prouver le lemme suivant : soient  $x$  et  $y$  deux nombres flottants, si  $|x + y| \leq 2^{-1022}$  alors  $x \oplus y = x + y$ .

**Question 11.** Si l'on soustrait deux valeurs proches, mais résultats d'un calcul arrondi, on peut obtenir un résultat complètement faux. Comparer  $(2^{52} \oplus 1/8) \ominus (2^{52} \ominus 1/8)$  et la valeur mathématique sans arrondi  $(2^{52} + 1/8) - (2^{52} - 1/8)$ .

**Question 12.** Mais ce n'est pas la dernière soustraction qui crée cette erreur, elle ne fait que mettre en lumière les erreurs précédentes. Prouver le lemme suivant : soient  $x$  et  $y$  deux nombres flottants, si  $y/2 \leq x \leq 2y$ , alors  $x \ominus y = x - y$ .



## Partie 3 : Logique, déduction naturelle

Un rappel des règles de déduction naturelle se trouve en annexe de cette partie.

### Exercice 1

Nous avons les faits suivants :

1. Si Informatix réussit sa preuve, il sera content.
2. S'il pleut, Informatix restera chez lui.
3. Si Informatix ne reste pas chez lui, alors il sera content.
4. Chez lui, Informatix s'entraîne.
5. Informatix réussira sa preuve s'il s'entraîne.

- a. Formaliser ces faits à l'aide de formules logiques propositionnelles.
- b. Montrer, en déduction naturelle, qu'Informatix sera content.

### Exercice 2

- a. Montrer en déduction naturelle les deux séquents suivants. L'un des deux séquents est classique, dire lequel.
  1.  $\Gamma \vdash (A \Rightarrow B) \Rightarrow (\neg A \vee B)$
  2.  $\Gamma \vdash (\neg A \vee B) \Rightarrow (A \Rightarrow B)$
- b. Lorsque la formule à éliminer est une hypothèse, on donne les règles alternatives **Hypothèses**, en annexe.  
En vous aidant des règles **Hypothèses**, montrer que :  $\Gamma \vdash \exists x \neg P(x) \Rightarrow \neg(\forall x P(x))$
- c. À partir de la règle  $\vdash A \vee \neg A$ , est-il possible de montrer  $\vdash \neg\neg A \Rightarrow A$ ?  
Si oui, faire la démonstration, sinon expliquer pourquoi.
- d. Expliquer par un (contre-)exemple (intuitif) pourquoi  $\forall x \exists y P(x, y)$  n'est pas la même chose que  $\exists y \forall x P(x, y)$ .

## Annexe : règles de la déduction naturelle

Axiome	Classique	Coupure
$\frac{A \in \Gamma}{\Gamma \vdash A} Ax$	$\overline{\Gamma \vdash A \vee \neg A} EM$	$\frac{\Gamma, A \vdash B \quad \Gamma \vdash A}{\Gamma \vdash B} Cut$

	Introduction	Elimination
$\perp$		$\frac{\Gamma \vdash \perp}{\Gamma \vdash C} \perp e$
$\neg$	$\frac{\Gamma, A \vdash \perp}{\Gamma \vdash \neg A} \neg i$	$\frac{\Gamma \vdash \neg A \quad \Gamma \vdash A}{\Gamma \vdash C} \neg e$
$\wedge$	$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge i$	$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge eg \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge ed$
$\vee$	$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee ig \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vee id$	$\frac{\Gamma, A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C} \vee e$
$\Rightarrow$	$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow i$	$\frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} \Rightarrow e$
$\forall$	$\frac{\Gamma \vdash P \quad x \notin vl(\Gamma)}{\Gamma \vdash \forall x, P} \forall i$	$\frac{\Gamma \vdash \forall x, P}{\Gamma \vdash P[x \leftarrow t]} \forall e$
$\exists$	$\frac{\Gamma \vdash P[x \leftarrow t]}{\Gamma \vdash \exists x, P} \exists i$	$\frac{\Gamma \vdash \exists x, P \quad \Gamma, P \vdash C \quad x \notin vl(\Gamma, C)}{\Gamma \vdash C} \exists e$

	Hypothèses
$\perp$	$\overline{\Gamma, \perp \vdash C} \perp h$
$\neg$	$\frac{\Gamma, \neg A \vdash A}{\Gamma, \neg A \vdash C} \neg h$
$\wedge$	$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C} \wedge h$
$\vee$	$\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \vee h$
$\Rightarrow$	$\frac{\Gamma, B \vdash C \quad \Gamma, A \Rightarrow B \vdash A}{\Gamma, A \Rightarrow B \vdash C} \Rightarrow h$
$\forall$	$\frac{\Gamma, (\forall x, P), P[x \leftarrow t] \vdash C}{\Gamma, (\forall x, P) \vdash C} \forall h$
$\exists$	$\frac{\Gamma, P \vdash C \quad x \notin vl(\Gamma, C)}{\Gamma, (\exists x, P) \vdash C} \exists h$

## Partie 4 : Réseaux

Dans cette partie, nous considérons un réseau composé de 3 machines : la machine A sur laquelle intervient un utilisateur ; et deux autres machines hébergeant respectivement un serveur DHCP et un serveur DNS. Le tout est relié à Internet à travers une passerelle (Routeur R1) afin que l'utilisateur profite notamment de deux autres services différents : WEB et FTP.

Les serveurs DNS, DHCP, WEB et FTP possèdent tous des adresses IP fixes. Le serveur DHCP attribue dynamiquement les adresses dans la plage 174.24.64.32 et 174.24.64.62 et la première adresse disponible est 174.24.64.45. Nous supposons qu'en dehors de la machine A, toutes les autres machines sont correctement configurées, viennent d'être installées (tous les caches sont vides) et opérationnelles. Leur fichier de configuration sont illustrés en partie sur la figure 1.

Nous rappelons que la réalisation d'un service donné sur ce réseau fait parfois appel à d'autres services générant des trafics additionnels. Des rappels sur les protocoles sont donnés en annexe.

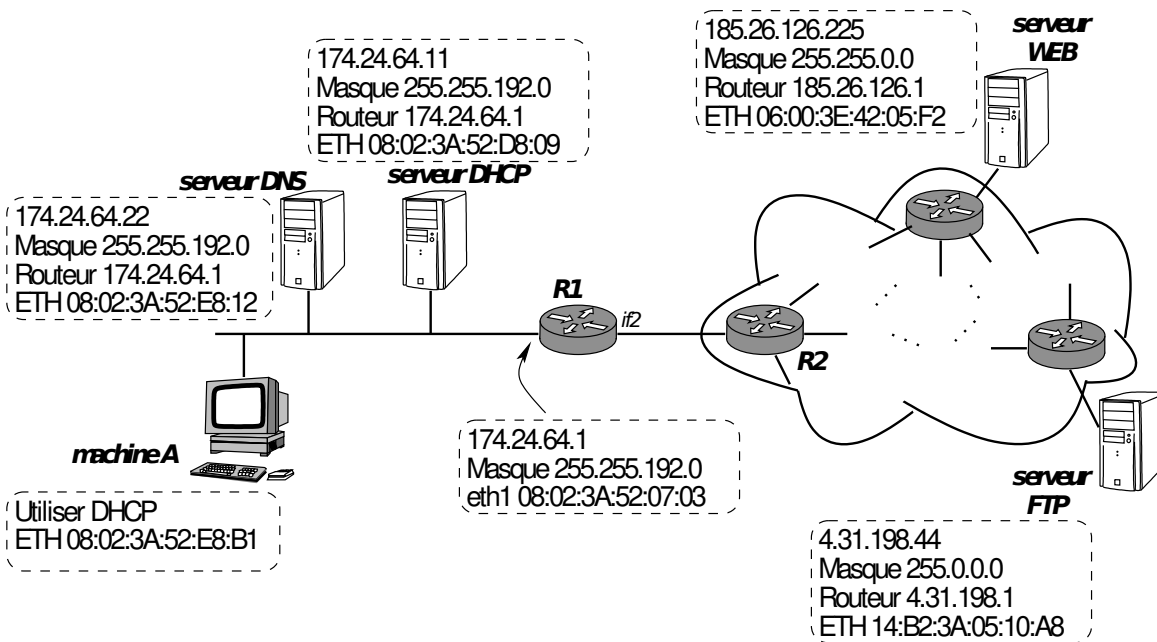


FIGURE 1 – Le réseau étudié.

1. La machine A ne dispose pour l'instant que de son adresse MAC et sait qu'il faut utiliser DHCP pour obtenir une adresse IP. Décrire les trames transmises dans le réseau 174.24.64.0 pour que la machine A obtienne une adresse IP. Pour cela, indiquer le service auquel correspond chaque trame et le plus de détails possibles sur le type d'opération du service et les entêtes des unités de données encapsulées dans la trame (les adresses MAC, les adresses IP, et le cas échéant d'autres identifiants de niveau supérieur).

2. L'utilisateur de la machine A souhaite consulter le site web `www.agreg-info.org`. Décrire les trames échangées lorsque la machine A cherche à établir une session WEB sur le serveur (`www.agreg-info.org`). Nous supposons que le serveur DNS situé sur le même réseau que l'utilisateur dispose, lui, de l'information pour résoudre directement le nom de domaine `www.agreg-info.org` en `185.26.126.225`.
3. Pour les trames identifiées dans les deux premières questions et encapsulant des messages de haut niveau (applicatifs), est ce qu'elles encapsulent toutes des segments du même protocole de transport ? Justifier.
4. Suite à une analyse de trafic sur l'interface `if2` du routeur R1 nous avons capturé les deux datagrammes représentés en figure 2. Visiblement il s'agit de deux fragments d'un même datagramme (numéro 85). En effet, le protocole IP au niveau de R2 a dû fragmenter le datagramme numéro 85 avant de le transmettre sur le lien R2 – R1.

Ver. <b>4</b>	IHL <b>5</b>	TOS <b>0</b>	Total Length <b>1020</b>		Ver. <b>4</b>	IHL <b>5</b>	TOS <b>0</b>	Total Length <b>480</b>	
Identification <b>85</b>		Flags <b>1</b>	Fragment Offset <b>0</b>		Identification <b>85</b>		Flags <b>0</b>	Fragment Offset <b>125</b>	
TTL <b>55</b>	Protocol <b>6</b>	Header Checksum <b>615356</b>			TTL <b>55</b>	Protocol <b>6</b>	Header Checksum <b>725354</b>		
Source IP address <b>185.26.126.225</b>				Source IP address <b>185.26.126.225</b>					
Destination IP address <b>174.24.64.45</b>				Destination IP address <b>174.24.64.45</b>					
Data <b>'x' octets</b>				Data <b>'y' octets</b>					

FIGURE 2 – Deux datagrammes. Les acronymes des champs sont comme suit : Ver. (Version), IHL (Internet Header Length), TOS (Type of Service), TTL (Time To Live).

- a– Donner l'unité maximale de transmission (MTU) sur le lien R2 – R1. Pour quelles raisons peut-on limiter la taille maximale des trames sur les supports de transmission ?
  - b– Même si la MTU dans le réseau `174.24.64.0` est de `1500 octets`, le protocole IP ne réassemble les fragments qu'à la destination finale. Pour quelles raisons ?
  - c– Le champs TTL est utilisé pour détruire les datagrammes lorsque sa valeur arrive à 0. Sachant que la valeur du TTL du datagramme original (à la source) était de 64 au départ, combien de routeurs ont été traversés par ces deux datagrammes ?
  - d– Quelles sont les valeurs de '*x*' et '*y*' dans les champs data des deux fragments ?
  - e– La somme de contrôle "Header Checksum" sert à la fonction du contrôle d'erreurs au niveau de l'entête du datagramme IP. Cette fonction est-elle réalisée uniquement au niveau de la destination ou au niveau de chaque routeur traversé ? Justifier.
5. À présent, l'utilisateur souhaite faire un transfert de fichiers vers le serveur FTP.
    - a– Ce service nécessite également l'établissement d'une connexion. Quelle est l'entité de protocole qui se charge de l'établissement de cette connexion ? Cette même entité existe sur R1, sur R2, ou sur tous les routeurs ?
    - b– Vu que la machine A dispose déjà d'une connexion TCP avec le serveur WEB, peut-il établir une seconde connexion pour le service FTP ? Si oui, comment l'entité protocolaire distingue-t-elle les deux connexions ?

- c- On considère dans la suite que la machine A a normalement terminé le téléchargement et a fermé sa connexion avec le service FTP. La connexion avec le service WEB s'est brutalement interrompue et la machine A en démarre une nouvelle (toujours vers `www.agreg-info.org`). Est-il possible que des segments de la première connexion du service WEB interfèrent avec ceux de la seconde? Justifier.
6. Nous considérons la table de routage du routeur R2. Quatre datagrammes arrivent sur R2 et ont respectivement comme adresse IP destination : 174.24.64.45, 188.114.255.2, 188.114.255.4, 67.44.21.97.

Destination	Prochain saut	Interface
174.24.64.0/18	R1	if1
67.44.16.0/20	R3	if3
67.44.16.0/22	R4	if4
188.114.255.0/30	R3	if3
0.0.0.0/0	R5	if2

- a- Donner le prochain saut de chacun de ces paquets en expliquant le principe de fonctionnement suivi et les opérations réalisées dans la table.
- b- Nous avons, par la suite, constaté d'autres paquets destinés toujours à la machine 174.24.64.45. Ces paquets proviennent de la même source et ont exactement la même taille. En revanche leur latences respectives étaient différentes. Donner les raisons de cette variabilité en décrivant les facteurs impactant les délais de transmission de bout-en-bout.
7. Nous considérons cette fois un système autonome constitué de six routeurs nommés R1, R2, R3, R4, R5 et R6 (voir figure 3). Le protocole de routage utilisé est RIP (Routing Information Protocol) s'appuyant sur l'algorithme de détermination des routes décentralisé Bellman-Ford.

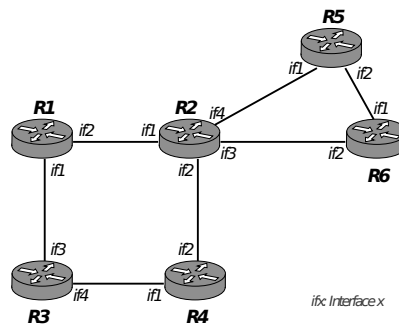


FIGURE 3 – Système autonome à six routeurs.

- a- Établir les tables de routage des routeurs R2 et R3. L'ordre dans lequel circulent les messages RIP a-t-il de l'importance?
- b- La liaison entre R2 et R6 tombe en panne. Expliquer comment les routeurs R2 et R3 détectent la rupture et montrer comment l'information se propage aux autres routeurs.
- c- Expliquer la différence entre une panne d'un routeur et une rupture de lien avec l'incidence que cela pourrait avoir sur les tables de routage.
8. Supposons que la machine A et les deux serveurs DNS et DHCP sont sur un réseau ETHERNET 1 Gbit/s qui utilise une trame minimale de 512 octets.
- a- Quel est le temps d'émission d'une trame de longueur minimale?

- b- Peut-on en déduire la période de vulnérabilité dans un tel réseau ?
- c- Décrire la différence entre ce débit support, le débit utile et le débit réel en s'appuyant sur les facteurs influençant les variations du débit.

## Annexes partie 4

### Annexe A - Encapsulation dans TCP/IP et dans UDP/IP

Dans la figure 4, nous présentons un exemple d'encapsulation dans TCP/IP. Dans cet exemple, un segment TCP contenant un message de données est encapsulé dans datagramme IP, lui même encapsulé à son tour dans une trame ETHERNET. Un segment UDP serait encapsulé de la même manière qu'un segment TCP.

Chacune de ces couches (TCP|UDP, IP, ETHERNET) sont gérées par une entité protocolaire associée : le processus décodant les informations spécifiques à cette couche.

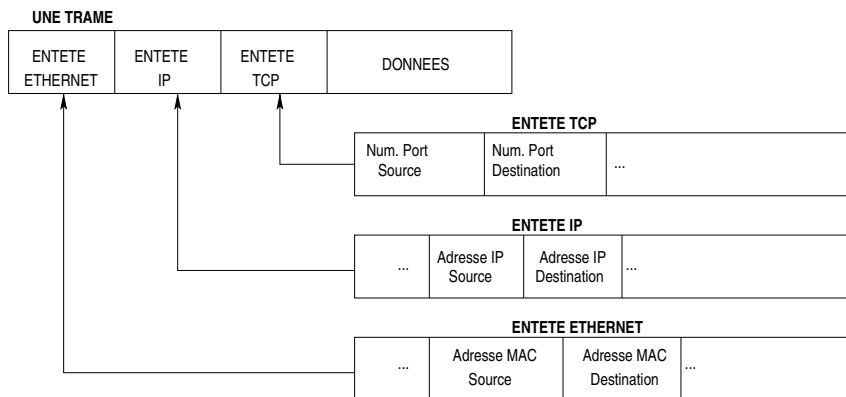


FIGURE 4 – Encapsulation de données dans TCP/IP.

### Annexe B - Principe de DHCP

Le serveur applicatif DHCP attribue dynamiquement des adresses IP dans une plage d'adresses libres. Le serveur fonctionne sur un port UDP 67. Pour la mise en œuvre d'un tel service quatre types de messages sont utilisés :

- Le client diffuse un datagramme *DHCP\_DISCOVER*.
- Tout serveur DHCP ayant reçu ce datagramme, envoie une offre *DHCP\_OFFER* s'il est en mesure de proposer une adresse sur le réseau auquel appartient le client.
- Le client retient une des offres reçues et diffuse sur le réseau un datagramme de requête *DHCP\_REQUEST*.
- Le serveur DHCP élabore un datagramme d'accusé de réception *DHCP\_ACK* qui assigne au client l'adresse IP et son masque de sous-réseau, la durée du bail de cette adresse, etc.

### Annexe C - Principe de DNS

Un serveur DNS (*Domain Name System ou Système de noms de domaine*) opère sur un port UDP numéro 53. Il se charge de la traduction des noms de domaine Internet en adresse IP. Pour la mise en œuvre d'un tel service, nous nous limiterons à l'échange suivant :

- Le client DNS de la machine envoie une requête DNS "Je recherche l'adresse IP de *www.agreg-info.org*".
- Le serveur DNS envoie une réponse au client (*www.agreg-info.org = @IP 185.26.126.225*).