université de BORDEAUX	**ANNÉE UNIVERSITAIRE 2017-2018** **SESSION 1 PRINTEMPS 2018** **MENTION : INFORMATIQUE** **Code UE : 4TIN807** **Intitulé de l'épreuve : Réseaux et protocoles** **Date : 26/4/2018 Heure : 14h30 Durée : 3 heures** **Tous documents autorisés** **Épreuve de M. L. Lamali**	**Collège Sciences et technologies** **Masters**

Il est conseillé de lire attentivement et intégralement le sujet avant de répondre aux questions. Les différentes parties des exercices sont indépendantes, même si la lecture d'une partie peut aider à traiter la partie suivante. Les questions sont numérotées et en gras, le barème est indicatif. Une importance particulière sera accordée à la présentation.

1 Tunnels IP dans IP

1.1 Principes de base (3 points)

Sur une route dans un réseau, un **tunnel IP sur IP** est une portion de la route où un paquet IP est encapsulé dans un autre paquet IP. La Figure 1 représente un tel tunnel. La pile de protocoles (i.e., la suite des en-têtes) est représentée sous chaque lien.

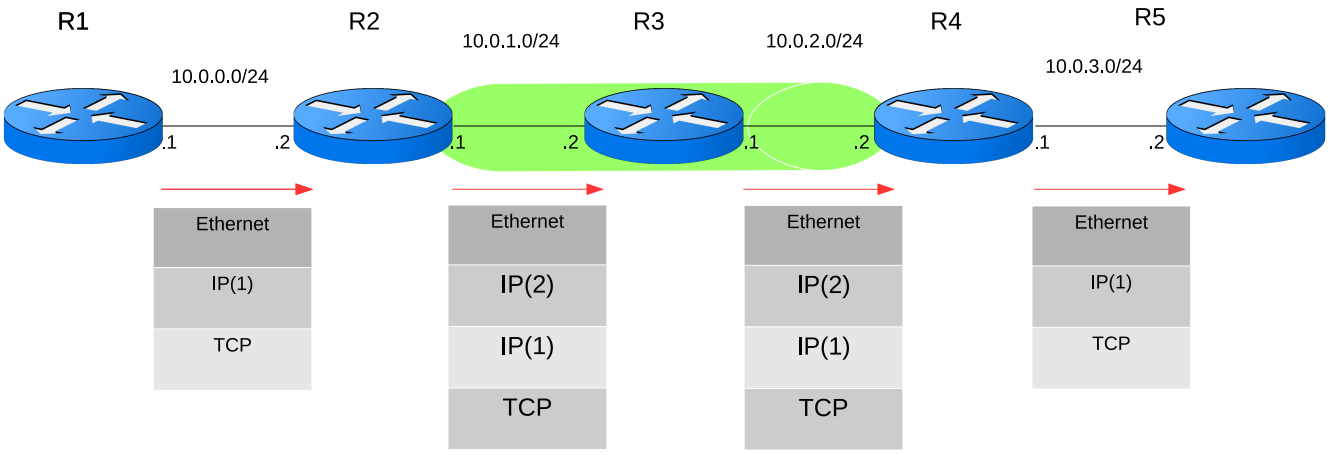

FIGURE 1 – Image représentant un tunnel entre R2 et R4 sur la route entre R1 et R5.

En effet, R1 envoie un paquet IP classique (ici IP1) avec comme adresse émetteur sa propre adresse, et adresse destinataire celle de R5. Mais R2 *encapsule* ce paquet dans un autre paquet IP (ici IP2) où l'adresse émetteur est celle de R2, et l'adresse de destination est celle de R4. Puis il envoie le paquet à R3. R3 n'a accès qu'à l'en-tête IP *externe*, c'est-à-dire IP2. Il croit donc que le destinataire est R4 et lui transmet le paquet. R4, recevant le paquet, *désencapsule* le paquet IP *interne*, constate que le destinataire est R5, et lui transmet.

On dit que la portion R2-R4 est un **tunnel**. R2 est l'*entrée* du tunnel, et R4 est sa *sortie*. Les fonctions des tunnels sont très variées : sécurité pour empêcher les routeurs intermédiaires de connaître la vraie destination, forcer le paquet à passer par un routeur particulier en le désignant comme la sortie du tunnel, permettre à un paquet IPv6 de traverser un réseau IPv4 en étant encapsulé dans un paquet IPv4, etc.

1. **Pour l'exemple de la Figure 1, lorsque R1 envoie un paquet à R5 via le tunnel, donner pour la trame sur chaque lien :**
   - **L'adresse MAC source et destination (on notera @MAC_Ri_G pour l'adresse MAC de l'interface gauche du routeur Ri, et @MAC_Ri_D pour l'interface droite.)**
   - **Les adresses IP source et destination, ainsi que le protocole encapsulé dans le paquet IP (il s'agit ici du paquet directement encapsulé dans la trame ETHERNET.)**
   - **Les adresses IP source et destination, ainsi que le protocole encapsulé dans le paquet IP interne, le cas échéant.**
2. **Sachant que les liens ETHERNET entre les routeurs ne peuvent transmettre que des trames de taille ≤ 1518 octets, quelle est la valeur que devrait avoir le MTU à R1 pour éviter la fragmentation ?**

1.2 Routage avec tunnels (7 points)

Soit le réseau représenté sur la Figure 2.

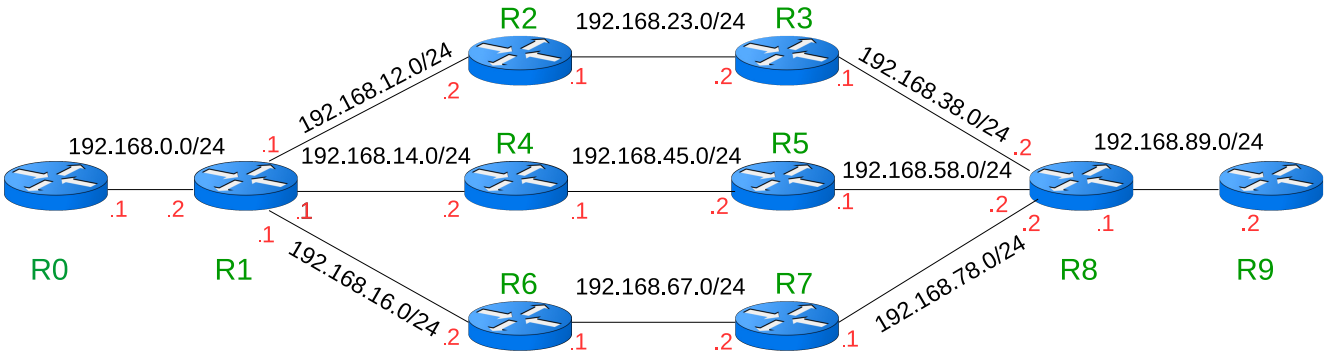

FIGURE 2 –

Et soient les extraits suivants des tables de routage de chaque routeur :

	Destination	Next hop
R0 :	Gateway	192.168.0.2
R1 :	192.168.23.0/24	192.168.12.2
	192.168.67.0/24	192.168.16.2
	192.168.89.0/24	192.168.14.2
R2 :	192.168.89.0/24	192.168.23.2
R3 :	192.168.89.0/24	192.168.38.2
R4 :	192.168.89.0/24	192.168.45.2
R5 :	192.168.89.0/24	192.168.58.2
R6 :	192.168.89.0/24	192.168.67.2
R7 :	192.168.89.0/24	192.168.78.2

De plus, chaque routeur peut joindre ses voisins immédiats, par exemple, R2 peut joindre R3, R6 peut joindre R7, etc.

3. **R0 veut envoyer un paquet à R9. D'après les tables de routages, quel chemin sera emprunté par le paquet ?**

Il se trouve que R0 ne veut pas que son paquet passe par ce chemin. Il va donc créer un tunnel dont il est lui-même l'entrée, c'est-à-dire qu'il va émettre un paquet IP contenant un autre paquet IP. On sait également que R3, R5 et R7 sont capables de désencapsuler un paquet IP d'un autre. Donc ils sont potentiellement des sorties de tunnel. Nous avons capturé le paquet suivant sur le lien R0-R1 :

```
ff ee dd cc bb aa aa bb  cc dd ee ff 08 00 45 00   ........ ......E.
00 47 00 01 00 00 40 04  e2 5e c0 a8 00 01 c0 a8   .G....@. .^......
17 02 45 00 00 33 00 01  00 00 40 06 a0 71 c0 a8   ..E..3.. ..@..q..
00 01 c0 a8 59 01 00 14  00 50 00 00 00 00 00 00   ....Y... .P......
00 00 50 02 20 00 0a ba  00 00 48 45 49 53 54 48   ..P. ... ..HEISTH
45 31 37 54 48                                     E17TH
```

FIGURE 3 – Capture d'une trame sur le lien R0-R1.

Pour le protocole encapsulé dans le paquet IP :

— la valeur $0x01$ signifie que le paquet IP contient un paquet ICMP,

— la valeur $0x04$ signifie que le paquet IP contient un autre paquet IP,

— la valeur $0x06$ signifie que le paquet IP contient un segment TCP,

— la valeur $0x11$ signifie que le paquet IP contient un datagramme UDP.

4. **Quelles sont les adresses MAC source et destination ?**
5. **A priori, quelle est la destination du paquet IP ? (donner l'adresse IP et le nom du routeur)**
6. **Quel est le protocole encapsulé dans le paquet IP ?**
7. **Quelle est la destination réelle du paquet IP ? (donner l'adresse IP et le nom du routeur)**
8. **Quel chemin sera emprunté par ce paquet jusqu'à la destination finale ? Justifier.**
9. **Quel est le protocole de couche 4 ? Quels sont les ports source et destination ?**
10. **Quel protocole de couche application est utilisé ici ?**

1.3 Algorithmes de routage (6 points)

Dans cette partie, on ignore ce qui se passe au niveau ETHERNET et ne traite que la partie IP. On suppose qu'il existe une route entre chaque couple de routeurs.
Soit un réseau avec $\mathcal{R} = \{R1, R2, \ldots, Rn\}$ l'ensemble de ses routeurs, et $\mathcal{L} = (Ri, Rj, \ldots, Rd)$ un sous-ensemble ordonné (une liste) de $\mathcal{R}$. Un routeur particulier, nommé $R0$, veut envoyer un paquet à un routeur destination Rd. Mais il veut que le paquet passe par tous les routeurs de $\mathcal{L}$ dans le bon ordre (plus exactement, il veut qu'il y ait une sous-séquence dans le chemin qui correspond à $\mathcal{L}$, donc certains routeurs peuvent réapparaître, ce n'est pas un problème). On suppose qu'il peut encapsuler un nombre arbitraire de paquets les uns dans les autres.

Pour créer des paquets, il dispose des fonctions suivantes :

— IP(), pour créer un paquet.
Exemple : $p1 \leftarrow$ IP()

— *paquet.set_src*([@IP]), pour affecter l'adresse IP source à un paquet.
Exemple : $p1$.*set_src*("192.168.4.8")

— *paquet.set_dst*([@IP]), pour affecter l'adresse IP destination à un paquet.
Exemple : $p1$.*set_dst*("192.168.15.16")

— *send*(*paquet*), pour envoyer un paquet.
Exemple : *send*($p1$)

— *paquet.encap*([*paquet'*]), pour encapsuler *paquet'* et tout ce qu'il contient, notamment les paquets déjà encapsulés dans *paquet'* dans *paquet*.
Exemple : $p2.encap(p1)$

— *decap*([*paquet*]), pour désencapsuler tout ce que contient *paquet*.
Exemple : $p1 \leftarrow decap(p2)$.
Donc si on fait :
$p2.encap(p1)$
$p1 \leftarrow decap(p2)$
On récupère $p1$ tel qu'il était avant d'être encapsulé.

L'adresse IP d'un routeur Ri est simplement notée `@`Ri. On dispose également des fonctions classiques d'itération sur une liste :

— For x in *liste*, pour parcourir une liste.

— *reverse*(*liste*) pour inverser une liste.

11. **Donner un algorithme qui permet à $R0$ d'envoyer le paquet en passant par tous les routeurs de $\mathcal{L}$ comme il le souhaite.**
12. **Si $R0$ ne veut pas que les routeurs intermédiaires sachent qu'il est l'émetteur, modifier l'algorithme en conséquence.**

On dispose également des fonctions suivantes :

— *paquet.get_src*(), pour lire l'adresse IP source d'un paquet.
Exemple : $a \leftarrow p1.get\_src()$

— *paquet.get_dst*(), pour lire l'adresse IP destination d'un paquet.
Exemple : $a \leftarrow p1.get\_dst()$

— *paquet.get_prot*(), pour lire le protocole encapsulé dans un paquet.
Exemple : $t \leftarrow p1.get\_prot()$, après cette instruction, t contiendra "IP" ou "TCP", etc.

13. **Pour un routeur intermédiaire R_i, qu'il soit ou non dans $\mathcal{L}$, donner un algorithme permettant d'effectuer correctement le routage d'un paquet p reçu.**

1.4 Transmission et tunnels (4 points)

On reprend le réseau de la Figure 1. Chaque routeur R_i a un débit d_i. Le routeur $R1$ émet des données par groupes de t octets vers $R5$, on suppose qu'il n'y a pas de fragmentation. Les données sont à chaque fois encapsulées dans TCP, IP, puis ETHERNET.

14. **Pour chaque groupe, combien d'octets émet réellement $R0$?**
15. **Lors de la transmission de ce groupe de données, combien d'octets émettra réellement chaque routeur Ri ?**
16. **En prenant en compte le tunnel entre $R2$ et $R4$, combien de temps les données vont mettre pour aller de $R0$ à $R5$? (on ignore le temps de propagation). Application numérique : $d_i = 10^6$ bits/s, $t = 10^3$ octets.**

2 Exercice bonus

Soit une liaison avec un BER noté p. On suppose qu'on envoie des données en continu bit par bit.

1. **Quel est, en moyenne, le volume de données envoyées entre deux erreurs successives ?**