**Question 1**
**Question :**
(1pt) Un paquet peut être éliminé dans un routeur à cause de :
a. une surcharge dans les tampons d'entrée seulement
b. une surcharge dans les tampons de sortie seulement
c. une contention dans la matrice de commutation seulement
d. réponses a et b
**Idée simple :** un paquet est jeté quand il n'y a plus de place pour le garder (dans un tampon).
**a)** « surcharge dans les tampons d'entrée seulement »
Les tampons d'entrée = files où les paquets attendent à l'arrivée.
Si c'est plein, le paquet entrant n'a aucune place.
Résultat : le routeur peut jeter le paquet.
Donc a est vrai.
**b)** « surcharge dans les tampons de sortie seulement »
Les tampons de sortie = files où les paquets attendent avant de sortir.
Si le lien sort lentement / congestionné, cette file devient pleine.
Quand c'est plein, le routeur ne peut plus ajouter de paquet à cette sortie.
Résultat : le routeur peut jeter le paquet.
Donc b est vrai.
**c)** « contention dans la matrice de commutation seulement »
La matrice de commutation = la partie interne qui déplace un paquet d'une entrée vers une sortie.
La contention = plusieurs paquets veulent traverser en même temps → certains attendent.
Attendre n'est pas "jeter". On jette seulement si l'attente fait déborder un tampon.
Mais ça impliquerait des tampons pleins, pas "la matrice seulement".
Donc c est faux.
**d)** « réponses a et b »
Puisque a et b sont vraies, la bonne réponse est d.
**Réponse :** d

**Question 2**
**Question :**
(1pt) Le contrôle de parité à une dimension ne permet pas de détecter l'erreur si :
a. le message contient un bit erroné
b. le message contient trois bits erronés
c. le message contient quatre bits erronés
d. le seul bit erroné est le bit de parité
**Idée simple :** avec 1 bit de parité, tu détectes les erreurs si le nombre de bits inversés est impair. Si le nombre est pair, ça peut passer inaperçu.
**a)** « le message contient un bit erroné »
1 bit inversé = impair.
La parité change → on voit l'erreur.
Donc a est faux.
**b)** « le message contient trois bits erronés »
3 bits inversés = impair.
La parité change → on voit l'erreur.
Donc b est faux.
**c)** « le message contient quatre bits erronés »
4 bits inversés = pair.
La parité peut redevenir "comme avant" → l'erreur peut ne pas être détectée.
Donc c est vrai.
**d)** « le seul bit erroné est le bit de parité »
Ça fait 1 bit erroné.
La vérification échoue → on détecte l'erreur.
Donc d est faux.
**Réponse :** c

**Question 3**
**Question :**
(1pt) Le contrôle de parité à deux dimensions permet de détecter :
a. seulement les erreurs sur trois bits ou moins
b. seulement les erreurs sur deux bits ou moins
c. n'importe quelles erreurs sur quatre bits
d. n'importe quelles erreurs sur trois bits
**Idée simple :** on ajoute une parité par ligne et par colonne. Pour qu'une erreur ne soit pas détectée, il faudrait que chaque ligne et chaque colonne reste "correcte".
**a)** « seulement les erreurs sur trois bits ou moins »
2D parité détecte toujours 1, 2, 3 bits.
Mais elle détecte aussi certaines erreurs à 4 bits.
Le mot "seulement" rend l'énoncé trop restrictif.
Donc a est faux.
**b)** « seulement les erreurs sur deux bits ou moins »
2D parité détecte aussi toutes les erreurs à 3 bits.
Donc b est faux.
**c)** « n'importe quelles erreurs sur quatre bits »
Faux : 4 erreurs peuvent passer si elles forment un "rectangle" (2 erreurs par ligne et 2 par colonne).
Donc c est faux.
**d)** « n'importe quelles erreurs sur trois bits »
Avec 3 erreurs, il est impossible de garder toutes les lignes/colonnes correctes.
Donc c'est détecté à coup sûr.
Donc d est vrai.
**Réponse :** d

**Question 4**
**Question :**
(1pt) Un commutateur (switch) Ethernet possédant cinq ports numérotés 1, 2, 3, 4 et 5, reçoit une trame destinée au terminal A sur le port 3.
Dans sa table de transfert, il possède une seule ligne : (Destination : A, Port de sortie : 3).
Le commutateur :
a. élimine la trame
b. diffuse la trame sur les ports 1, 2, 3, 4 et 5
c. diffuse la trame sur les ports 1, 2, 4 et 5
d. retransmet la trame sur le lien duquel elle a été reçue
**Idée simple :** si la destination est sur le même port que l'arrivée, le switch ne renvoie pas la trame.
**a)** « élimine la trame »
Destination A → port 3, et trame reçue sur port 3.
Donc rien à envoyer ailleurs : le switch filtre.
Donc a est vrai.
**b)** « diffuse sur 1,2,3,4,5 »
La diffusion (flood) est pour une destination inconnue.
Ici A est connu.
Donc b est faux.
**c)** « diffuse sur 1,2,4,5 »
Toujours diffusion alors que A est connu.
Donc c est faux.
**d)** « retransmet sur le lien reçu »
Un switch ne renvoie pas sur le port d'entrée.
Donc d est faux.
**Réponse :** a

**Question 5**
**Question :**
(1pt) Un commutateur (switch) Ethernet possédant cinq ports numérotés 1, 2, 3, 4 et 5, reçoit une trame destinée au terminal B sur le port 3.
Dans sa table de transfert, il possède une seule ligne : (Destination : A, Port de sortie : 3).
Le commutateur :
a. élimine la trame
b. diffuse la trame sur les ports 1, 2, 3, 4 et 5
c. diffuse la trame sur les ports 1, 2, 4 et 5
d. retransmet la trame sur le lien duquel elle a été reçue
**Idée simple :** destination inconnue → diffusion sur tous les ports sauf le port d'entrée.
**a)** « élimine la trame »
Destination inconnue → normalement on diffuse, pas on jette.
Donc a est faux.
**b)** « diffuse sur 1,2,3,4,5 »
On ne diffuse pas sur le port d'entrée (3).
Donc b est faux.
**c)** « diffuse sur 1,2,4,5 »
Oui : tous sauf 3.
Donc c est vrai.
**d)** « retransmet sur le lien reçu »
Non, il diffuse sur les autres ports.
Donc d est faux.
**Réponse :** c

**Question 6**
**Question :**
(1pt) La taille de l'adresse IPv6 :
a. est la même que la taille de l'adresse IPv4
b. est deux fois la taille de l'adresse IPv4
c. est trois fois la taille de l'adresse IPv4
d. est quatre fois la taille de l'adresse IPv4
**Idée simple :** IPv4 = 32 bits, IPv6 = 128 bits.
**a)** « même taille »
128 ≠ 32.
Donc a est faux.
**b)** « deux fois »
2 × 32 = 64, pas 128.
Donc b est faux.
**c)** « trois fois »
3 × 32 = 96, pas 128.
Donc c est faux.
**d)** « quatre fois »
4 × 32 = 128.
Donc d est vrai.
**Réponse :** d

**Question 7**
**Question :**
(1pt) La taille de l'entête IPv4 est
a. fixe mais la taille de l'entête IPv6 est variable
b. variable mais la taille de l'entête IPv6 est fixe
c. fixe ainsi que la taille de l'entête IPv6
d. variable ainsi que la taille de l'entête IPv6
**Idée simple :** IPv4 peut avoir des "options" → l'entête peut grossir. IPv6 a un entête de base toujours de même taille, et ce qui "varie" est mis ailleurs (entêtes d'extension).
**a)** « IPv4 fixe, IPv6 variable »
IPv4 n'est pas fixe (il peut y avoir des options).
Donc a est faux.
**b)** « IPv4 variable, IPv6 fixe »
IPv4 variable (options → taille peut changer).
IPv6 fixe (entête de base toujours la même taille).
Donc b est vrai.
**c)** « IPv4 fixe et IPv6 fixe »
IPv4 n'est pas fixe.
Donc c est faux.
**d)** « IPv4 variable et IPv6 variable »
IPv4 variable oui, mais l'entête de base IPv6 est fixe.
Donc d est faux.
**Réponse :** b

**Question 9**
**Question :**
(1pt) Le protocole Ethernet fournit, à la couche réseau, le service suivant
a. un transfert de données fiable
b. une détection d'erreurs
c. un contrôle de congestion
d. réponses a et b
e. aucun des trois premiers choix
**Idée simple :** Ethernet peut détecter qu'une trame est abîmée (CRC/FCS). Mais il ne garantit pas "livré à 100%" et ne fait pas de contrôle de congestion.
**a)** « transfert fiable »
Ethernet peut perdre des trames (collision, surcharge, etc.) et ne garantit pas la livraison par lui-même.
Donc a est faux.
**b)** « détection d'erreurs »
Oui : il y a un champ de contrôle (FCS/CRC) pour détecter si la trame est corrompue.
Donc b est vrai.
**c)** « contrôle de congestion »
Ethernet ne gère pas la congestion comme TCP (pas de fenêtre, pas d'ajustement de débit).
Donc c est faux.
**d)** « a et b »
a est faux, b est vrai → pas les deux.
Donc d est faux.
**e)** « aucun des trois premiers »
b est vrai, donc ce n'est pas "aucun".
Donc e est faux.
**Réponse :** b

**Question 12**
**Question :**
(1pt) Le protocole ALOHA slotted
a. est un protocole d'accès multiple par partitionnement du canal
b. est un protocole d'accès aléatoire totalement décentralisé
c. est un protocole d'accès aléatoire ayant besoin de synchronisation
d. est un protocole toujours plus efficace que CSMA/CD
e. réponses b et c
**Idée simple :** Slotted ALOHA = chacun essaie d'émettre "au hasard", mais seulement au début d'un créneau de temps → donc il faut une synchronisation des créneaux.
**a)** « partitionnement du canal »
Partitionnement = on découpe le canal en parts fixes (temps/fréquence/code réservés).
Slotted ALOHA ne réserve pas une part fixe par utilisateur : c'est du hasard.
Donc a est faux.
**b)** « accès aléatoire totalement décentralisé »
Oui : chaque station décide d'émettre ou non, sans contrôleur central.
Donc b est vrai.
**c)** « accès aléatoire ayant besoin de synchronisation »
Oui : il faut que tout le monde soit d'accord sur les "slots" (début/fin des créneaux).
Donc c est vrai.
**d)** « toujours plus efficace que CSMA/CD »
Non : CSMA/CD est souvent plus efficace que ALOHA en charge moyenne.
Donc d est faux.
**e)** « réponses b et c »
b est vrai et c est vrai.
Donc e est vrai.
**Réponse :** e

**Question 8**
**Question :**
(1pt) Un routeur IP implémentant Network Address Translation (NAT) est placé à la sortie d'un LAN;
quand les hôtes du LAN communiquent avec les hôtes d'Internet, le NAT :
a. laisse les adresses IP des hôtes inaltérées, et laisse le numéro de port TCP inaltéré
b. laisse les adresses IP des hôtes inaltérées, mais change le numéro de port TCP
c. substitue les adresses des hôtes par son adresse IP, mais laisse le numéro de port TCP inaltéré
d. substitue les adresses des hôtes par son adresse IP, et change le numéro de port TCP
**Idée simple :** le NAT "cache" les machines du LAN derrière une seule adresse IP publique. Pour distinguer plusieurs connexions en même temps, il change aussi souvent le port (PAT/NAPT).
**a)** « ne change ni IP ni port »
Si on ne change rien, ce n'est pas du NAT.
Donc a est faux.
**b)** « ne change pas l'IP, change le port »
Si l'IP privée reste la même sur Internet, ça ne marche pas (les IP privées ne sont pas routées sur Internet).
Donc b est faux.
**c)** « change l'IP, ne change pas le port »
Ça peut arriver dans un cas très particulier (1 machine ↔ 1 IP publique dédiée).
Mais à la sortie d'un LAN (plusieurs machines), si deux machines utilisent le même port source, il faut les distinguer.
Donc en pratique pour un LAN, NAT change IP + port.
Donc c est faux ici.
**d)** « change l'IP et change le port »
C'est le comportement typique : IP privée → IP publique du NAT, et port ajusté pour distinguer les sessions.
Donc d est vrai.
**Réponse :** d

**Question 10**
**Question :**
(1pt) Le RIP est un algorithme de routage à
a. vecteur de distances dont la métrique est le nombre de sauts
b. vecteur de distances dont la métrique est le délai
c. état de lien qui utilise comme métrique le nombre de sauts
d. état de lien qui utilise comme métrique le délai
**Idée simple :** RIP compte "combien de routeurs" tu traverses (hop count) et il échange des tables (distance-vector).
**a)** « distance vector + nombre de sauts »
C'est exactement RIP.
Donc a est vrai.
**b)** « distance vector + délai »
RIP n'utilise pas le délai comme métrique principale.
Donc b est faux.
**c)** « état de lien + nombre de sauts »
RIP n'est pas "état de lien".
Donc c est faux.
**d)** « état de lien + délai »
Deux erreurs : pas état de lien, pas délai.
Donc d est faux.
**Réponse :** a

**Question 11**
**Question :**
(1pt) Le protocole DHCP est encapsulé directement dans
a. TCP
b. UDP
c. IP
d. Ethernet
**Idée simple :** DHCP envoie des messages simples (broadcast au début) et utilise UDP.
**a)** TCP
DHCP n'utilise pas TCP.
Donc a est faux.
**b)** UDP
Oui : DHCP fonctionne sur UDP (classiquement ports 67/68).
Donc b est vrai.
**c)** IP
DHCP est au-dessus de UDP : ce n'est pas "directement dans IP".
Donc c est faux.
**d)** Ethernet
DHCP n'est pas directement dans Ethernet (il passe par IP/UDP).
Donc d est faux.
**Réponse :** b

**Question 13**
**Question :**
(2pts) Dans CSMA/CD, après la quatrième collision, quelle est la probabilité qu'un nœud transmette immédiatement après la collision ?
a. 0.5
b. 0.25
c. 0.125
d. 0.0625
e. 0
**Idée simple :** après une collision, CSMA/CD choisit un temps d'attente aléatoire. Après la 4e collision, le nœud choisit K au hasard parmi 0 à 15 (16 choix). "Transmettre immédiatement" = choisir K = 0.
**a)** 0.5
0.5 = 1/2. Ça correspondrait à 2 choix seulement (0 ou 1), pas 16.
Donc a est faux.
**b)** 0.25
0.25 = 1/4. Ça correspondrait à 4 choix (0..3), pas 16.
Donc b est faux.
**c)** 0.125
0.125 = 1/8. Ça correspondrait à 8 choix (0..7), pas 16.
Donc c est faux.
**d)** 0.0625
0.0625 = 1/16.
Ici il y a 16 choix (0..15) et "immédiat" = 1 choix (0).
Donc probabilité = 1/16 = 0.0625.
Donc d est vrai.
**e)** 0
Ce n'est pas impossible : si K = 0, il peut transmettre tout de suite.
Donc e est faux.
**Réponse :** d

## Question 14

**Question :**
(2pts) Dans CSMA/CD, deux nœuds A et B désirant transmettre de nouvelles trames tombent en collision.
En tentant une retransmission, ils tombent en collision une deuxième fois.
Quelle est la probabilité qu'ils tombent en collision à leur 3ème tentative ?
a. 0.125
b. 0.25
c. 0.33
d. 0.5
e. aucune des réponses précédentes
**Idée simple :** après la 2e collision, chaque nœud choisit K au hasard parmi 0,1,2,3 (4 choix). Ils recollisionnent si A et B choisissent le même K.
**a)** 0.125
0.125 = 1/8. Ici on obtient 1/4.
Donc a est faux.
**b)** 0.25
C'est exactement 1/4.
Donc b est vrai.
**c)** 0.33
0.33 ≈ 1/3, ce n'est pas ce qu'on calcule ici.
Donc c est faux.
**d)** 0.5
0.5 = 1/2, beaucoup trop grand pour 4 choix possibles.
Donc d est faux.
**e)** aucune des réponses précédentes
Comme b est correct, e est faux.
**Réponse :** b

## Exercice 3 — Recette a suivre (regle du plus long prefixe)

Objectif : savoir choisir le 'prochain routeur' en appliquant toujours les memes etapes.

### La recette (a refaire a chaque fois)

1 **Etape 1 — Trouver quelles lignes peuvent correspondre**
Dans cet exercice, chaque ligne commence par **128.96** et les masques ne changent que le **3e nombre** (le 3e octet). Donc on regarde seulement le 3e nombre de l'IP destination.

2 **Etape 2 — Utiliser le masque pour connaitre l'intervalle (range)**
Mask **255.255.254.0** → taille de bloc **2** → intervalle = **S a S+1**
Mask **255.255.252.0** → taille de bloc **4** → intervalle = **S a S+3**
(S = le 3e nombre ecrit dans la ligne du tableau.)

3 **Etape 3 — Si plusieurs lignes matchent, choisir la plus specifique**
**/23** (255.255.254.0) est plus specifique que **/22** (255.255.252.0). Donc si les deux matchent, on choisit la ligne en /23.

4 **Etape 4 — Le prochain routeur**
Le prochain routeur est simplement celui ecrit dans la ligne choisie.

### Application aux 4 adresses

```
128.96.167.151 → 3e nombre = 167
Ligne 128.96.164.0 avec masque 255.255.252.0 couvre 164–167 → match
Donc prochain routeur = R3

128.96.163.151 → 3e nombre = 163
Pas dans 160–161, pas dans 164–167, pas dans 168–169, pas dans 170–171 →
aucun match
Donc route par defaut → R4

128.96.169.192 → 3e nombre = 169
Ligne 128.96.168.0 avec masque 255.255.254.0 couvre 168–169 → match
Donc prochain routeur = R2

128.96.240.121 → 3e nombre = 240
Aucune ligne ne couvre 240 → aucun match
Donc route par defaut → R4
```

**Question 5**
**Choisir TCP vs UDP - elements a considerer :**
- **Fiabilite** : TCP retransmet et garantit l'ordre; UDP non.
- **Ordre** : TCP maintient l'ordre; UDP peut arriver desordonne.
- **Latence** : UDP est souvent plus faible (moins de mecanismes).
- **Congestion** : TCP ajuste le debit (controle de congestion); UDP laisse la gestion a l'application.
- **Overhead** : TCP plus lourd; UDP plus leger.
- **Type d'usage** : web/fichiers/courriel -> TCP; voix/video temps reel/jeux -> souvent UDP.

**Question 6**
**Role d'un proxy (fureteur <-> serveur web) :** le navigateur envoie ses requetes au proxy, et le proxy parle au serveur. Le proxy peut aussi renvoyer des reponses deja en cache.
**Pourquoi en avoir un :**
- **Cache** : accelere l'acces en reutilisant des reponses.
- **Controle** : filtrage, politiques d'acces, journalisation.
- **Securite** : masquage, inspection, reduction d'exposition du reseau interne.
- **Performance** : compression, parfois repartition de charge selon l'architecture.

**Question 7**
**Role (au niveau application/transport) :** un numero (0-65535) qui indique **quel programme** sur une machine doit recevoir les donnees. Exemples : HTTP 80, HTTPS 443, DNS 53.

**Question 8**
**Pourquoi FTP n'est pas securitaire :** les identifiants et les donnees peuvent circuler **en clair** (non chiffres). Une personne qui ecoute le reseau peut lire login/mot de passe et fichiers.

**Question 9**
**Pourquoi un changement de domaine prend du temps :** a cause des **caches DNS** partout (FAI, navigateurs). Chaque cache conserve la reponse pendant un **TTL**. Tant que le TTL n'a pas expire, ils verront l'ancienne information.

**Question 10**
**Multiplexage / demultiplexage (Application <-> Transport) :**
**Multiplexage (envoi)** : plusieurs applications envoient; la couche transport utilise des ports pour melanger les flux.
**Demultiplexage (reception)** : la couche transport lit le port destination et remet les donnees au bon programme.

## Question 1

**Qu'est-ce qu'un protocole de communication ?** Un ensemble de regles qui precise : le format des messages, l'ordre d'echange et quoi faire en cas d'erreur ou d'absence de reponse.

**5 protocoles utilises sur Internet + niveau :**
- HTTP / HTTPS - Application
- DNS - Application
- SMTP - Application
- TCP - Transport
- IP - Reseau

## Question 2

**Composants physiques usuels :**
- Cable d'entree du fournisseur (fibre / coaxial / cuivre).
- Modem / ONT (convertit le signal du fournisseur en reseau utilisable).
- Routeur Wi-Fi (partage Internet, NAT, pare-feu de base, Wi-Fi).
- Point(s) d'acces Wi-Fi / Mesh (optionnel, pour augmenter la couverture).
- Switch Ethernet (optionnel, ajoute des ports filaires).
- Cables Ethernet (pour PC, TV, console, etc.).
- Appareils clients (telephone, PC, TV, consoles, imprimantes, objets IoT).

## Question 3

**Modele a 5 couches :**
- **Application** : protocoles des applications (HTTP, DNS, SMTP).
- **Transport** : communication entre programmes via ports; fiabilite (TCP) ou non (UDP).
- **Reseau** : adressage et routage entre reseaux (IP).
- **Liaison** : communication sur le reseau local; trames, MAC, CRC (Ethernet/Wi-Fi).
- **Physique** : transmission des bits sur cable/air (signaux).

## Question 4

**Modele client-serveur (couche application) :** le **client** demande un service (ex. navigateur), le **serveur** fournit le service (ex. serveur web). Le serveur ecoute sur un port connu (ex. 80/443), et le client initie la communication.

### Regle de la division (XOR)

- On travaille avec une fenetre de 5 bits (car G a 5 bits).
- Si la fenetre commence par 1: on fait XOR avec 10011.
- Si la fenetre commence par 0: on fait XOR avec 00000 (donc ca ne change rien).
- Apres le XOR: on garde les 4 derniers bits (on enleve le premier bit) et on 'descend' le prochain bit du dividend.
- A la fin, le reste CRC est exactement 4 bits.

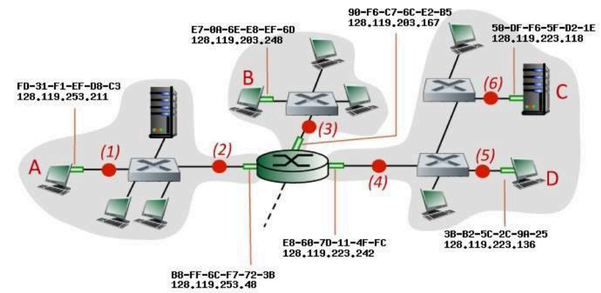
### Division pas-a-pas (fenetre = 5 bits)

| Etape | Fenetre | XOR avec | Resultat XOR | Garder 4 bits | Descendre bit | Nouvelle fenetre |
|---|---|---|---|---|---|---|
| 1 | 10010 | 10011 | 00001 | 0001 | 0 | 00010 |
| 2 | 00010 | 00000 | 00010 | 0010 | 1 | 00101 |
| 3 | 00101 | 00000 | 00101 | 0101 | 1 | 01011 |
| 4 | 01011 | 00000 | 01011 | 1011 | 0 | 10110 |
| 5 | 10110 | 10011 | 00101 | 0101 | 0 | 01010 |
| 6 | 01010 | 00000 | 01010 | 1010 | 0 | 10100 |
| 7 | 10100 | 10011 | 00111 | 0111 | 0 | 01110 |
| 8 | 01110 | 00000 | 01110 | 1110 | - | (fin) |

### Resultat

Reste CRC (4 bits) = **1110**

Message transmis = Data + Reste = **10010011 + 1110 = 100100111110**




a. On suppose que les tables ARP de tous les nœuds sont initialement vides. Nous nous intéressons uniquement à l'adressage des trames. Donnez la liste des trames ARP échangées dans le réseau (indice : il s'agit de quatre trames) en fournissant pour chaque trame les informations suivantes :

**Requête ARP envoyé par A pour rechercher l'@MAC du routeur**

| @MAC source | @ MAC dest. | @ IP de l'interface recherché |
|---|---|---|
| FD-31-F1-EF-D8-C3 | FF-FF-FF-FF-FF-FF | 128.119.253.48 |

**Réponse ARP du routeur**

| @MAC source | @ MAC dest. | @ IP de l'interface recherché |
|---|---|---|
| B8-FF-6C-F7-72-3B | FD-31-F1-EF-D8-C3 | - |

**Requête ARP envoyé par le routeur pour rechercher l'@MAC de D**

| @MAC source | @ MAC dest. | @ IP de l'interface recherché |
|---|---|---|
| E8-60-7D-11-4F-FC | FF-FF-FF-FF-FF-FF | 128.119.223.136 |

**Réponse ARP de D**

| @MAC source | @ MAC dest. | @ IP de l'interface recherché |
|---|---|---|
| 3B-B2-5C-2C-9A-25 | E8-60-7D-11-4F-FC | - |

b. On suppose que chaque nœud connait les adresses MAC de tous les nœuds de son réseau local. On suppose aussi que les commutateurs dans les réseaux de B et C ne disposent pas d'une entrée pour l'adresse MAC de D dans leurs tables de transfert. Donnez les adresses IP et MAC contenues dans les trames (si la trame existe) aux points (1), (2), (3), (4), (5) et (6). Vous devez remplir pour chaque trame le tableau suivant

**Point (1)**

| @ MAC source | @MAC dest. | @IP source | @ IP dest. |
|---|---|---|---|
| FD-31-F1-EF-D8-C3 | B8-FF-6C-F7-72-3B | 128.119.253.211 | 128.119.223.136 |

**Point (2) - (même que Point (1))**

| @ MAC source | @MAC dest. | @IP source | @ IP dest. |
|---|---|---|---|
| FD-31-F1-EF-D8-C3 | B8-FF-6C-F7-72-3B | 128.119.253.211 | 128.119. 223.136 |

**Point (3) - Pas de trame**

**Point (4)**

| @ MAC source | @MAC dest. | @IP source | @ IP dest. |
|---|---|---|---|
| E8-60-7D-11-4F-FC | 3B-B2-5C-2C-9A-25 | 128.119.223. 211 | 128.119. 223.136 |

**Point (5) - (même que Point (4))**

| @ MAC source | @MAC dest. | @IP source | @ IP dest. |
|---|---|---|---|
| E8-60-7D-11-4F-FC | 3B-B2-5C-2C-9A-25 | 128.119.253.211 | 128.119. 223.136 |

**Point (6) - Pas de trame**