

- Expliquez et détaillez les réponses aux questions de l'examen.
- Si cela est nécessaire, précisez les hypothèses que vous avez posées pour résoudre les exercices, il peut parfois exister plusieurs interprétations différentes pour une question.
- Le barème est donné à titre indicatif.
- L'examen doit être rédigé à l'aide d'un traitement de texte et déposé au **format pdf** (exclusivement). Nous n'accepterons pas de photos de document manuscrit.

Correction

Exercice 1 : Ecriture de commandes (7 points)

Dans cet exercice, nous vous demandons d'écrire les commandes demandées dans un terminal shell soit sur la machine virtuelle, soit avec un Linux que vous aurez installé.

Ainsi, pour chaque question :

- donner la ou les commandes exécutées,
- **une copie d'écran** des commandes et de leurs résultats (réalisée sur une fenêtre terminal),
- des explications.

La figure 1 donne l'arborescence de fichiers obtenue pour un étudiant ayant pour nom Dupond.

Question 1 : Créer dans votre répertoire d'accueil (/home/user) un répertoire repNom où Nom est remplacé par votre nom.

Utilisation de la commande `mkdir` pour la création de répertoire

Copie d'écran du résultat de l'exécution :

```
user@cours:~$ mkdir repDupond
user@cours:~$ pwd
/home/user
user@cours:~$ ls
backupsMartin Exercices repDupond RepMartin
user@cours:~$
```

Question 2 : Créer dans le répertoire repNom deux répertoires Exercice1Nom et Exercice2Nom où Nom est remplacé par votre nom.

Utilisation de la commande `mkdir` pour la création de répertoire et de la commande `mkdir` pour se positionner dans le bon répertoire

Copie d'écran du résultat de l'exécution :

```
user@cours:~$ cd repDupond/
user@cours:~/repDupond$ mkdir Exercice1Dupond Exercice2Dupond
user@cours:~/repDupond$ ls
Exercice1Dupond Exercice2Dupond
user@cours:~/repDupond$
```

Question 3 : Créer dans le répertoire `Exercice1Nom` deux fichiers vides `infoNom` et `contenuRepNom` où `Nom` est remplacé par votre nom.

Utilisation de la commande `touch` pour la création de fichier vide

Copie d'écran du résultat de l'exécution :

```
user@cours:~$ cd repDupond/  
user@cours:~/repDupond$ mkdir Exercice1Dupond Exercice2Dupond  
user@cours:~/repDupond$ ls  
Exercice1Dupond Exercice2Dupond  
user@cours:~/repDupond$
```

Question 4 :

1. Afficher le contenu détaillé du répertoire `Exercice1Nom`.

Utilisation de la commande `ls -l` pour afficher le contenu détaillé d'un répertoire

Copie d'écran du résultat de l'exécution :

```
user@cours:~$ cd repDupond/Exercice1Dupond/  
user@cours:~/repDupond/Exercice1Dupond$ ls -l  
total 0  
-rw-r--r-- 1 user user 0 août 25 15:25 contenuRepDupond  
-rw-r--r-- 1 user user 0 août 25 15:25 infoDupond  
user@cours:~/repDupond/Exercice1Dupond$
```

2. Ecrire le contenu détaillé du répertoire `Exercice1Nom` dans le fichier `contenuRepNom`, la première ligne du fichier doit contenir le nombre total de block occupés par la liste de fichiers.

Utilisation de la commande `>` pour re-directionner la série standard d'une commande sur un fichier

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ ls -l > contenuRepDupond  
user@cours:~/repDupond/Exercice1Dupond$ cat contenuRepDupond  
total 0  
-rw-r--r-- 1 user user 0 août 25 15:30 contenuRepDupond  
-rw-r--r-- 1 user user 0 août 25 15:25 infoDupond  
user@cours:~/repDupond/Exercice1Dupond$ _
```

3. Afficher le contenu du fichier `contenuRepNom` sans la première ligne.

Utilisation de la commande `tail -n +2` pour afficher un fichier à partir de sa deuxième ligne

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ tail -n +2 contenuRepDupond  
-rw-r--r-- 1 user user 0 août 25 15:30 contenuRepDupond  
-rw-r--r-- 1 user user 0 août 25 15:25 infoDupond  
user@cours:~/repDupond/Exercice1Dupond$ _
```

4. Ecrire le contenu du fichier `contenuRepNom` sans la première ligne dans un fichier temporaire `tmpNom`.

Utilisation de la commande `>` pour re-directionner la série standard d'une commande sur un fichier

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ tail -n +2 contenuRepDupond > tmpDupond
user@cours:~/repDupond/Exercice1Dupond$ cat tmpDupond
-rw-r--r-- 1 user user 0 août 25 15:30 contenuRepDupond
-rw-r--r-- 1 user user 0 août 25 15:25 infoDupond
user@cours:~/repDupond/Exercice1Dupond$
```

5. Remplacer le contenu du fichier `contenuRepNom` par le contenu du fichier temporaire `tmpNom`.

Utilisation de la commande `cat` pour afficher le contenu d'un fichier et de la commande `>` pour re-directionner la série standard d'une commande sur un fichier

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ cat tmpDupond > contenuRepDupond
user@cours:~/repDupond/Exercice1Dupond$ cat contenuRepDupond
-rw-r--r-- 1 user user 0 août 25 15:30 contenuRepDupond
-rw-r--r-- 1 user user 0 août 25 15:25 infoDupond
user@cours:~/repDupond/Exercice1Dupond$
```

6. Supprimer le fichier temporaire `tmpNom`.

Utilisation de la commande `rm` pour supprimer un fichier

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ rm tmpDupond
user@cours:~/repDupond/Exercice1Dupond$ ls
contenuRepDupond infoDupond
user@cours:~/repDupond/Exercice1Dupond$ _
```

7. En utilisant le contenu du fichier `contenuRepNom`, afficher le nombre total de fichiers/répertoires du répertoire `Exercice1Nom`.

Utilisation de la commande `wc -l` qui compte le nombre de lignes d'un fichier, appliquée au fichier `contenuRepNom` qui contient le contenu du répertoire

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ wc -l contenuRepDupond
2 contenuRepDupond
```

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ set $(wc -l contenuRepDupond) ; echo " nombre de fichiers : $1"
nombre de fichiers : 2
```

Question 5 :

Ecrire un shell script `script1Nom` qui recopie chaque ligne du fichier `contenuRepNom` dans le fichier `infoNom` en ajoutant à la suite de chaque ligne :

1. "F" s'il s'agit d'un fichier ou "R" s'il s'agit d'un répertoire.
2. la taille du fichier (répertoire).

Le script `script1Nom` est sauvegardé dans le répertoire `Exercice1Nom`.

Copie d'écran du script :

```
user@cours:~/repDupond/Exercice1Dupond$ cat script1Dupond
# lecture ligne par ligne du fichier contenuRepDupond
cat contenuRepDupond | while read line
do
  # affectation de la ligne lue aux parametres positionnels
  set -- $line
  # la taille du fichier est en 5eme position
  taille=$5
  # les droits en 1ere position
  droits=$1
  # si le 1er caractere des droits est un d, c est un repertoire
  # sinon c est un fichier
  d=$(echo $droits | cut -c1)
  if [ $d = "d" ]
  then
    type=R
  else
    type=F
  fi
  # ajout d'une ligne a la suite du fichier infoDupond
  # contenant la ligne lue suivie du type de fichier et de sa taille
  echo "$line : $type : $taille" >> infoDupond
done
user@cours:~/repDupond/Exercice1Dupond$ _
```

Copie d'écran du résultat de l'exécution :

```
user@cours:~/repDupond/Exercice1Dupond$ ./script1Dupond
user@cours:~/repDupond/Exercice1Dupond$ cat infoDupond
-rw-r--r-- 1 user user    0 août 25 16:46 contenuRepDupond : F : 0
-rw-r--r-- 1 user user 1206 août 25 16:43 infoDupond : F : 1206
user@cours:~/repDupond/Exercice1Dupond$ _
```

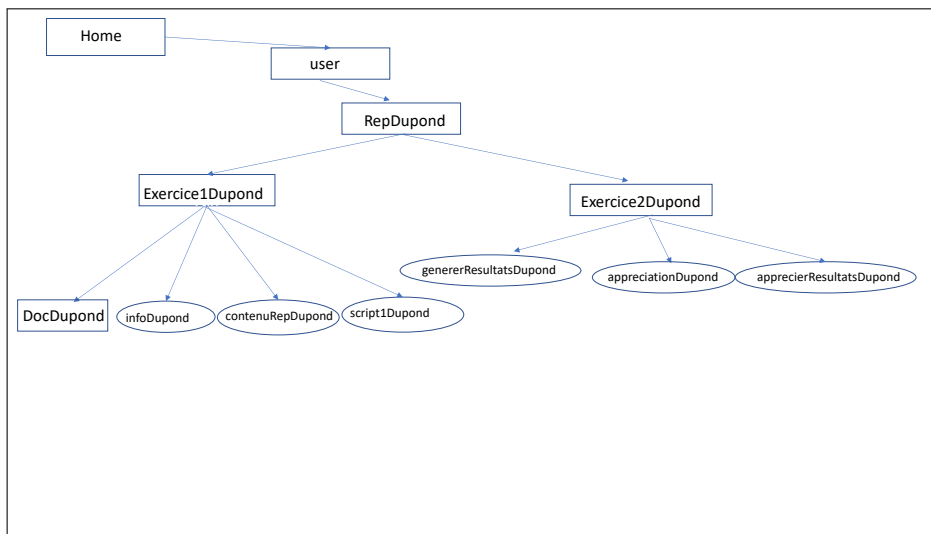


FIGURE 1 – Arborescence de fichiers pour l'étudiant Dupond

Exercice 2 : Ecriture de Script Shell (9 points)

Question 1 : Script 1

1. Écrire un script nommé `genererResultatsNom` (où `Nom` est remplacé par votre nom) qui génère des données au format : *Prénom Nom Note* :

Ce script shell réalise les actions suivantes :

- Calcul du nombre de lignes (variable `n_lignes`) à générer :
 - si un paramètre est passé : ce paramètre correspond au nombre de lignes à générer,
 - sinon, le nombre de lignes à générer est de 10.
 - Affichage de `n_lignes` au format *Prénom Nom Note* :
 - *Prénom* est généré de façon aléatoire : 6 caractères,
 - *Nom* est généré de façon aléatoire : 5 caractères,
 - *Note* est généré de façon aléatoire : de 0 à 20 inclus.
2. Le script `genererResultatsNom` est sauvegardé dans le répertoire `Exercice2Nom` créé à l'exercice 1.
 3. Donner une copie d'écran du script `genererResultatsNom` accompagnée de commentaires et de son résultat d'exécution.

Indications :

— `for ((var=valeurMin; var<=valeurMax; var++)) ; do ; commande ; done`

— Génération de nombres aléatoires : utilisation de la commande `shuf`. Par exemple :
`shuf -i 1-50 -n 5` permet de générer 5 nombres aléatoires entre 1 et 50.

— Génération de caractères aléatoires :

La commande `cat /dev/urandom | head -c 10` permet de générer 10 caractères aléatoires.

La commande `tr -dc 'a-z'` permet de supprimer les caractères non-alphabétiques dans une chaîne de caractères.

Copie d'écran du script :

```

user@cours:~/repDupond/Exercice2Dupond$ cat genererResultatsDupond
# s'il n'y a pas de paramètres, 10 lignes sont générées
# sinon le nombre de lignes à générer est donné par le premier paramètre
if [ $# -eq 0 ]
then
    line=10
else
    line=$1
fi

# Génération des lignes
for ((i=1;i<=$line;i++))
do
    # generation de caractères aléatoires, dans lesquels on conserve les caractères
    # alphabétiques puis conservation des 6 premiers pour le prenom
    # et des 5 premiers pour le nom
    prenom=$(cat /dev/urandom | tr -dc 'a-z' | head -c 6)
    nom=$(cat /dev/urandom | tr -dc 'a-z' | head -c 5)
    # génération d un nombre entre 0 et 20 pour la note
    note=$(shuf -i 0-20 -n 1)
    # affichage des nom prenom et note générés
    echo "$prenom $nom Notes: $note"
done

user@cours:~/repDupond/Exercice2Dupond$ _

```

Copie d'écran du résultat d'exécution

```

user@cours:~/repDupond/Exercice2Dupond$ ./genererResultatsDupond 5
nb lignes demandees : 5
fujrfa styst Notes: 16
vejdpv kuuev Notes: 20
tkskdf ipjcz Notes: 7
eqajby ystln Notes: 13
ozfyeo kcjqv Notes: 14
user@cours:~/repDupond/Exercice2Dupond$ ./genererResultatsDupond
nb lignes demandees : 10
neomuz tfcnc Notes: 17
wckqgt ofquu Notes: 19
zsgeup ezsf1 Notes: 19
xfnbwv poibr Notes: 6
sqgqan nazfo Notes: 0
sydmre twttx Notes: 0
ixpetq vcejp Notes: 18
zujafi gcjid Notes: 7
aoxptp skhgf Notes: 11
mcph1s dyvnp Notes: 14
user@cours:~/repDupond/Exercice2Dupond$

```

Question 2 : Script 2

1. Écrire un script nommé `appreciationNom` (où `Nom` est remplacé par votre nom) qui génère une appréciation automatique en fonction de la note passée en paramètre.

Ce script shell réalise les actions suivantes :

- Récupération de la note passée en paramètre dans la variable `note`
- Affichage sur la sortie standard de l'appréciation calculée ainsi :
 - si la note est supérieure ou égale à 17 : le script affiche "Excellent",
 - sinon, si la note est supérieure ou égale à 14 : le script affiche "Bien",
 - sinon, si la note est supérieure ou égale à 12 : le script affiche "Assez bien",
 - sinon, si la note est supérieure ou égale à 10 : le script affiche "Passable",
 - sinon le script affiche "Insuffisant".

2. Le script `appreciationNom` est sauvegardé dans le répertoire `Exercice2Nom` créé à l'exercice

- 1.
3. Donner une copie d'écran du script `appreciationNom` accompagnée de commentaires et de son résultat d'exécution.

Copie d'écran du script :

```
user@cours:~/repDupond/Exercice2Dupond$ cat appreciationDupond
# la note à apprécier est donnée dans le premier paramètre
note=$1
# Tests en cascade pour déterminer l'appréciation
if [ $note -ge 17 ]
then
    result="Excellent"
else if [ $note -ge 14 ]
then
    result="Bien"
else if [ $note -ge 12 ]
then
    result="Assez Bien"
else if [ $note -ge 10 ]
then
    result="Passable"
else
    result="Insuffisant"
fi
fi
fi
# Affichage de l'appréciation
echo $result
```

Copie d'écran du résultat d'exécution

```
user@cours:~/repDupond/Exercice2Dupond$ ./appreciationDupond 13
Assez Bien
user@cours:~/repDupond/Exercice2Dupond$ ./appreciationDupond 8
Insuffisant
user@cours:~/repDupond/Exercice2Dupond$ _
```

Question 3 : Script 3

Remarque : Si vous ne parvenez pas à générer le résultat attendu avec les deux premiers scripts, écrire un script qui affiche tout le temps la même chose de façon à pouvoir écrire ce troisième script.

1. Écrire un script nommé `apprecierResultatsNom` (où `Nom` est remplacé par votre nom) qui ajoute une colonne appréciation à un fichier créé à partir la commande `genererResultatsNom`. Ce script shell prend un nom de fichier en paramètre et réalise les actions suivantes :
 - teste si le paramètre est un fichier
 - Pour chaque ligne du fichier :
 - extrait la note,
 - calcule l'appréciation avec la commande `appreciationNom`,
 - affiche la ligne suivie d'une virgule et de l'appréciation calculée
2. Le script `apprecierResultatsNom` est sauvegardé dans le répertoire `Exercice2Nom` créé à l'exercice 1.
3. Générer un fichier `fichTestResulatsNom` à partir de la commande `genererResultatsNom`.
4. Donner une copie d'écran du script `apprecierResultatsNom` accompagnée de commentaires et de son résultat d'exécution appliqué au fichier `fichTestResulatsNom` généré précédemment.

Copie d'écran du script :

```
user@cours:~/repDupond/Exercice2Dupond$ cat apprecierResultatsDupond
# Stockage du premier paramètre dans la variable fichier
fichier=$1

# Si ce premier paramètre est un fichier
if [ -f $fichier ]
then
  # lecture du fichier ligne par ligne
  cat $fichier | while read line
  do
    #recupération de la note qui est le second champs après le :
    note=$(echo $line | cut -f2 -d":")
    # calcul de l'appréciation à l'aide du script appreciationdupond
    appreciation="$("./appreciationDupond $note)"
    # affichage de la ligne lue avec ajout de l'appréciation calculée
    echo "$line , $appreciation"
  done
else
  echo"parametre incorrect"
fi
```

Copie d'écran : génération du fichier de test :

```
user@cours:~/repDupond/Exercice2Dupond$ ./genererResultatsDupond 6 > fichTestResulats
user@cours:~/repDupond/Exercice2Dupond$ cat fichTestResulats
gatjzs rfdow Notes: 7
fzgxzu zxsdv Notes: 11
bbukvl sjvlg Notes: 12
yobkw1 himvj Notes: 15
xcsolg oufkb Notes: 14
vnestc yvovm Notes: 9
user@cours:~/repDupond/Exercice2Dupond$
```

Copie d'écran du résultat d'exécution

```
user@cours:~/repDupond/Exercice2Dupond$ ./apprecierResultatsDupond fichTestResulats
gatjzs rfdow Notes: 7 , Insuffisant
fzgxzu zxsdv Notes: 11 , Passable
bbukvl sjvlg Notes: 12 , Assez Bien
yobkw1 himvj Notes: 15 , Bien
xcsolg oufkb Notes: 14 , Bien
vnestc yvovm Notes: 9 , Insuffisant
user@cours:~/repDupond/Exercice2Dupond$
```

Exercice 3 : Mise en place d'un réseau pour un opérateur Télécom

4 points

Enoncé

La société-mère d'un opérateur Télécom va emménager dans un nouvel immeuble. L'opérateur a souhaité séparer ses divisions en quatre étages :

- Le deuxième étage accueillera l'administration (60 machines).
- Le premier étage accueillera la hotline (75 machines).
- Le rez-de-chaussée accueillera la division de recherche et développement disposant d'un espace ouvert pour les bureaux (100 machines).
- Enfin au sous-sol on trouvera la direction des systèmes d'information (80 machines).

Votre rôle est de proposer un découpage réseau pour cet immeuble. Le découpage proposé devra satisfaire les contraintes suivantes :

- Pour des raisons de sécurité d'accès, l'interconnexion entre les différents étages devra être effectuée par un équipement réseau au sous-sol, partie la plus protégée de l'immeuble.
- Pour limiter la diffusion de paquets inutiles et potentiellement confidentiels, chaque étage devra disposer de son propre routeur.
- On considère que, pour communiquer vers l'extérieur, un accès réseau `AccesReseauPublic` est à disposition. La configuration de cet accès n'est pas de votre ressort, mais il doit apparaître sur le plan proposé.
- On souhaite des sous-réseaux dimensionnés au plus juste, pouvant accueillir une demi-douzaine de machines de plus que prévu, de manière à permettre des dépannages ou le raccordement d'équipements temporaires.
- Un unique réseau devra être utilisé comme base de découpage.

Comme base de découpage, on propose l'une des trois plages réseaux suivantes : 172.16.0.0, 192.168.40.0 et 10.0.0.0. On nommera les sous-réseaux obtenus SR1, SR2, SR3...

Questions

1. Quelle est la particularité des trois plages réseaux proposées ? Quelles sont les implications en terme de communication avec Internet ?

Les trois plages réseaux proposées correspondent à des réseaux privés. Cela signifie que ces réseaux peuvent être utilisés librement pour des utilisations internes, mais ne peuvent pas être utilisés sur Internet : Il sera donc nécessaire aux machines clientes de passer par une machine disposant d'une adresse IP publique pour pouvoir accéder à Internet.

2. À quelle classe d'adresses chacune des plages appartient-elle ?

Le réseau 172.16.0.0 est un réseau de classe B, car son premier octet est compris entre 128 et 191 (deux premiers bits à 10). Le réseau 192.168.40.0 est un réseau de classe C, car son premier octet est compris entre 192 et 223 (bits à 110). Enfin, l'adresse 10.0.0.0 correspond à un réseau de classe A (premier octet en 01, inférieur à 128).

3. Compte-tenu des contraintes de l'énoncé, laquelle de ces plages est-il préférable de retenir pour le découpage ? Justifier précisément.

L'énoncé demande de créer un réseau comprenant $(60 + 6) + (75 + 6) + (100 + 6) + (80 + 6)$ soit approximativement (sans compter les routeurs et les adresses réservées) $66 + 81 + 106 + 86$ soit 339 adresses clientes. 192.168.40.0 est un réseau de classe C, qui ne fournit que 255 adresses, ce qui ne suffit pas à notre découpage. 10.0.0.0, en tant que réseau de classe

A, permet plus de 16 millions d'adresses, ce qui est beaucoup trop par rapport au découpage au plus juste demandé par l'énoncé. On retient donc la plage 172.16.0.0, de classe B, pour notre découpage.

4. Combien de sous-réseaux est-il nécessaire de créer, par rapport à l'organisation décrite et aux contraintes de l'énoncé ?

On nous demande un réseau par étage, ce qui correspond à 1 (DSI) + 1 (R & D) + 1 (Hotline) + 1 (Administration) = 4 sous-réseaux. A ces 4 sous réseaux, il est nécessaire d'ajouter ceux nécessaires à l'interconnexion des routeurs situés à chaque étage, soit 3 réseaux puisque le sous-sol est celui qui relie les autres étages, ce qui nous donne un total de 7 sous-réseaux.

5. Quelle sera la taille de chacun de ces sous-réseaux? Expliquer votre méthode de calcul pour un de ces sous-réseaux.

SR1 (DSI) : 80 postes + 6 suppl + 2 adresses réservées + 1 passerelle = 89 adresses minimum. On prend la puissance de 2 immédiatement supérieure ou égale (découpage au plus juste), ce qui nous donne une taille réseau de 128 adresses.

SR2 (R & D) : 100 postes + 6 suppl + 2 réservées + 1 = 109 minimum, on prend donc un sous-réseau de 128 adresses.

SR3 (Hotline) : 75 + 6 + 2 + 1 = 84 minimum, on prend donc un sous-réseau de 128 adresses.

SR4 (Administration) : 60 + 6 + 2 + 1 = 69 minimum, on prend donc un sous-réseau de 128 adresses.

SR5,6,7 (interco) : 2 adresses + 2 réservées = 4 minimum, on prend donc des sous-réseaux de taille 4. Comme il s'agit de réseaux d'interconnexion, il n'est pas nécessaire de prévoir de passerelle, et l'énoncé précise bien que le supplément de six adresses ne s'applique qu'aux réseaux clients.

6. Effectuer le découpage réseau demandé à l'aide de la plage d'adresse retenue dans la troisième question de l'exercice.

On nous demande un découpage au plus juste : Pour garantir le bon alignement des adresses de sous-réseaux obtenues, nous allons découper ces sous-réseaux dans la plage fournie du plus grand au plus petit, en repartant à chaque fois de l'adresse suivante disponible.

Les sous-réseaux étant tous de même taille à l'exception des sous-réseaux d'interconnexion, on découpera dans l'ordre SR1,2,3,4,5,6,7 :

SR1 (DSI), 80 postes

Adresse réseau : 172.16.0.0/25 (32 - 7 = 25)
Masque réseau : 255.255.255.128
Passerelle : 172.16.0.126
Broadcast : 172.16.0.127

SR2 (R & D), 100 postes

Adresse réseau : 172.16.0.128/25 (32 - 7 = 25)
Masque réseau : 255.255.255.128
Passerelle : 172.16.0.254
Broadcast : 172.16.0.255

SR3 (Hotline), 75 postes

Adresse réseau : 172.16.1.0/25 (32 - 7 = 25)
Masque réseau : 255.255.255.128
Passerelle : 172.16.1.126
Broadcast : 172.16.1.127

SR4 (Administration), 60 postes

Adresse réseau : 172.16.1.128/25 (32 - 5 = 27)
Masque réseau : 255.255.255.128
Passerelle : 172.16.1.254
Broadcast : 172.16.1.255

SR5, SR6, SR7 (interconnexion), 2 routeurs

Adresse réseau : 172.16.2.0/30 (32 - 2 = 30)
Masque réseau : 255.255.255.252
Broadcast : 172.16.2.3

Adresse réseau : 172.16.2.4/30 (32 - 2 = 30)
Masque réseau : 255.255.255.252
Broadcast : 172.16.2.7

Adresse réseau : 172.16.2.8/30 (32 - 2 = 30)
Masque réseau : 255.255.255.252
Broadcast : 172.16.2.11

SR1 (128 adresses) : 0.0/25
SR2 (128 adresses) : 0.128/25
SR3 (128 adresses) : 0.0/25
SR4 (128 adresses) : 0.128/25
SR5 (4 adresses) : 2.0/30
SR6 (4 adresses) : 2.4/30
SR7 (4 adresses) : 2.8/30