

**Nom** : .....

**Prénom** : .....

## **DS - M3102**

durée : 1h30 tous documents autorisés

---

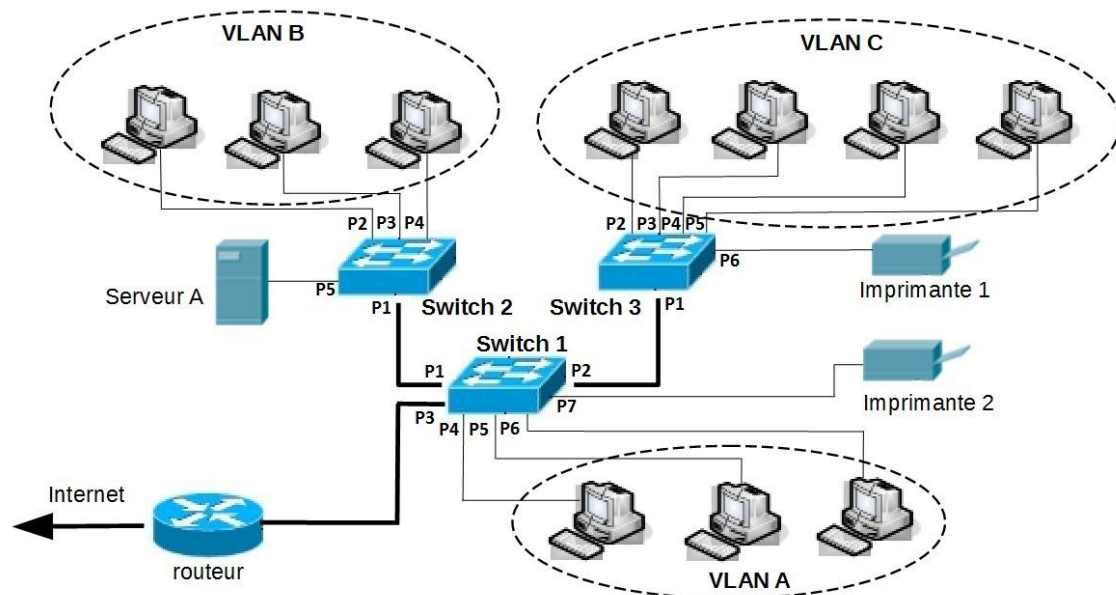
### **1. Choix multiples (2 points)** Pour chaque question, sélectionnez **la** ou **les** réponses correctes.

- Q1. Le mode connecté dans une application client/serveur (TCP) implique :
- A. l'intégrité des données
  - B. l'ordonnancement des données
  - C. la non-duplication des données
  - D. une augmentation des performances vs UDP
  - E. des primitives d'échange de données (read, write) plus compliquées
- Q2. La cryptographie asymétrique :
- A. permet de signer des messages
  - B. utilise uniquement une clé publique
  - C. utilise l'algorithme RSA
  - D. est plus rapide que la cryptographie symétrique
  - E. est utilisée dans SSH
- Q3. FTP, SMTP et Telnet sont des exemples de protocoles qui :
- A. Fonctionnent au niveau de la couche réseau
  - B. Fonctionnent au niveau de la couche application
  - C. Utilisent TCP comme protocole de transport
  - D. Utilisent UDP comme protocole de transport
  - E. Utilisent IP comme protocole de couche réseau
- Q4. Une communication client/serveur est caractérisée par :
- A. un processus serveur
  - B. un processus client
  - C. des processus identiques
  - D. un serveur qui est à l'initiative de l'échange
  - E. un système non-coopératif

### **2. VLAN (2 points)**

On veut diviser le réseau ci-dessous en 3 VLANs par ports avec les contraintes suivantes :

- l'imprimante 1 est accessible aux réseaux B et C,
- le routeur doit appartenir aux VLANs A, B et C,
- le serveur A doit appartenir aux VLANs B et C
- l'imprimante 2 est accessible aux réseaux A et B.



Compléter les tables de configuration de chaque commutateur :

Switch 1

Port	Mode	VLAN

Switch 2

Port	Mode	VLAN

Switch 3

Port	Mode	VLAN

### 3. RSA (3 points)

Dans l'algorithme RSA, on choisit les deux entiers premiers  $p = 17$  et  $q = 11$ . Calculer :

$n = \dots\dots\dots$ ,

$\Phi = (p-1)*(q-1) = \dots\dots\dots$

On choisit  $e = 7$

En déduire le plus petit  $d$  correspondant.

$d = \dots\dots\dots$

Clé publique :  $\dots\dots\dots$

Clé privée :  $\dots\dots\dots$

L'auteur du code rend aussi publique une correspondance entre lettres et chiffres afin de coder des messages textuels il utilise les entiers naturels (a codé par 01, b codé par 02, etc.) l'espace entre deux mots étant codé par 00. Sachant qu'il transmet le message « encore des maths », donner la suite numérique obtenue :

.....

Regrouper en tranches de chiffres conformément à l'algo RSA et donner la suite correspondante :

.....

indiquer ce qu'il faut faire ensuite pour crypter cette suite en utilisant la clé secrète :

.....

#### 4. Matching (2 points)

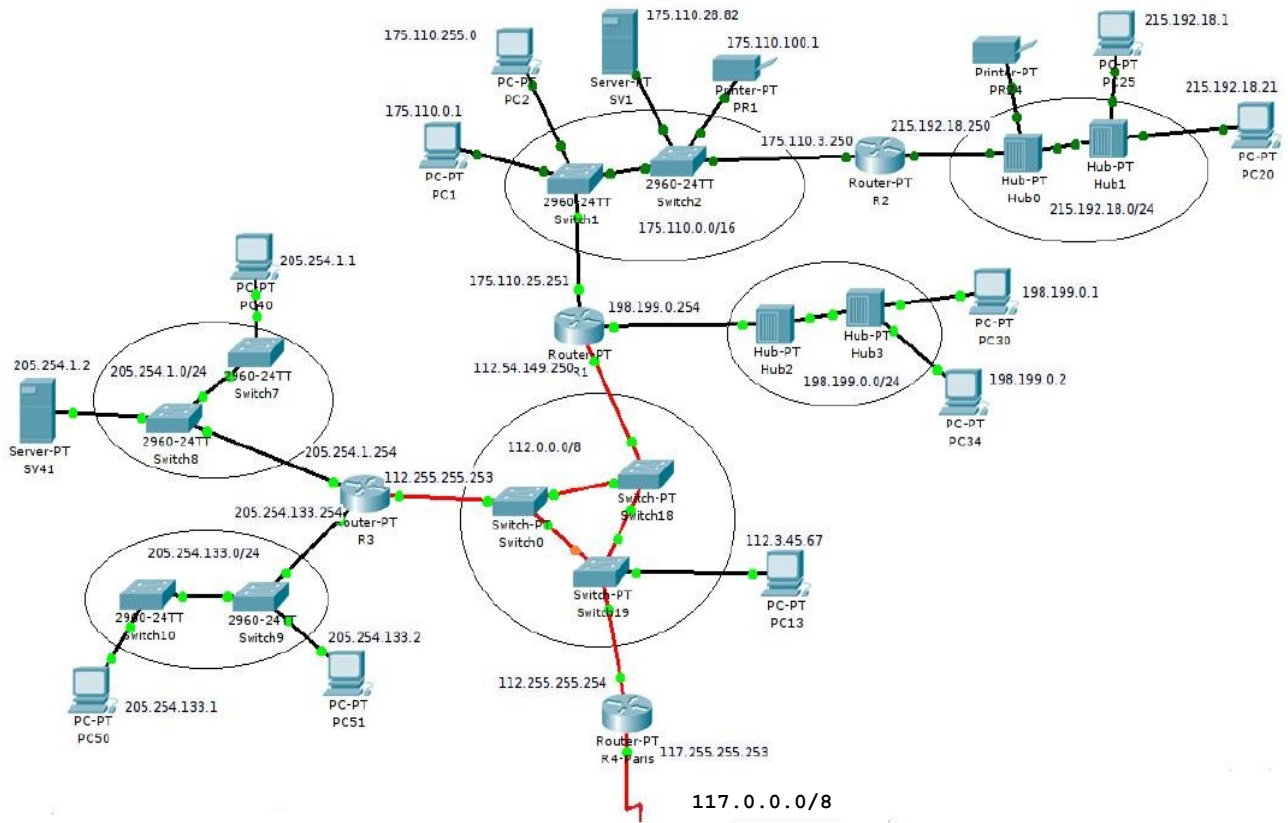
On demande d'associer **un** élément de la colonne 1 avec le meilleur choix possible de la colonne 2. Chaque élément de la colonne 2 sera utilisé **au plus une seule fois**.

Colonne 1	Colonne 2	Col.1	Col.2
1. <code>httpd</code>	A. directive spécifiant les règles à appliquer au répertoire concerné	1	
2. <code>/etc/named.conf</code>	B. contient les directives de contrôle d'accès au répertoire concerné	2	
3. <code>/etc/dhcp</code>	C. fichier client DHCP	3	
4. <code>/var/*/public_html</code>	D. teste si le serveur FTP sécurisé est actif	4	
5. <code>/etc/httpd/conf.d/vhost.conf</code>	E. répertoire des pages web du serveur	5	
6. <code>ps -ef   grep vsftpd</code>	F. fichier de gestion des serveurs web virtuels .	6	
7. <code>.htaccess</code>	G. fichier de configuration du serveur DNS	7	
8. <code>&lt;Directory ...&gt;</code>	H. rép. de gestion de l'allocation dynamique d'adresses	8	
	I. répertoire web des utilisateurs du serveur		
	J. démon du serveur web		

#### 5. Routage IP (3 points)

Dans la structure de réseau ci-dessous, on demande de compléter la table de routage du routeur R4 vers toutes les destinations possibles (rappel : sous Linux, la route 0.0.0.0 correspond au réseau directement accessible – adjacent - au routeur) :

Réseau de destination	Gateway	Subnet mask (/n)	metric
117.0.0.0			
112.0.0.0			
205.254.1.0			
205.254.133.0			
198.199.0.0			
175.110.0.0			
215.192.18.0			
0.0.0.0			



## 6. Etendues DHCP (2 points)

Compléter le tableau suivant à partir d'une adresse IP et d'un masque de sous-réseau. Déterminer la plage d'identificateurs qui inclut cette adresse IP (IP min et IP max) :

Adresse IP	Masque	IP min	IP max
131.107.56.25	/21		
148.53.66.7	/20		
198.53.24.100	/26		
198.54.64.7	/24		

## 7. RIP (3 points)

Le protocole RIP utilise l'algorithme de Bellman-Ford distribué. Chaque routeur envoie le coût et le routeur à utiliser pour atteindre ses voisins, à chaque routeur voisin. Pour le réseau suivant, donner la table de routage de R8 à l'état de convergence des coûts.

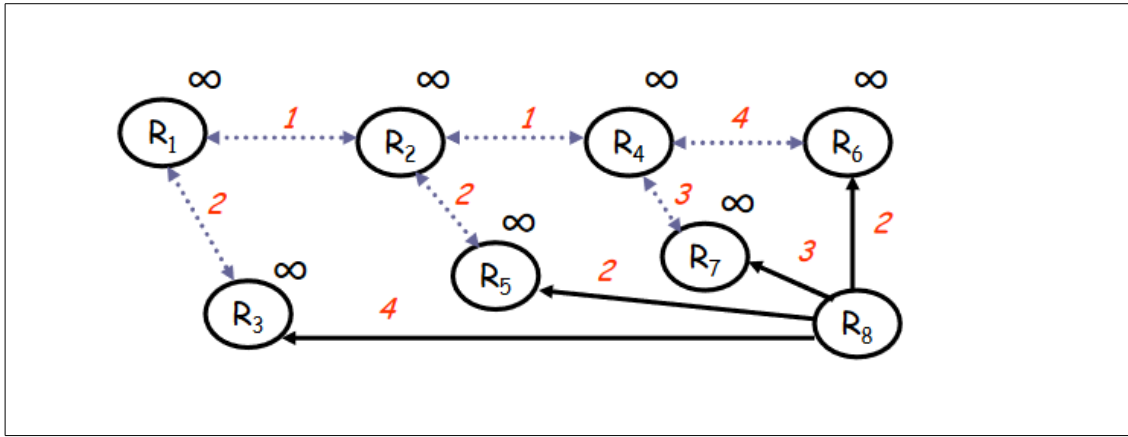


Table de R8 :

Routeur	Coût	Routeur précédent la destination
R1		
R2		
R3		
R4		
R5		
R6		
R7		

### 8. Sockets TCP/IP (3 points)

Pour chaque fragment de code C++, on demande de répondre aux questions suivantes.

```
if ((ListenSocket=socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) == INVALID_SOCKET)
    cout << "socket error : " << WSAGetLastError() << endl;
```

quel est le rôle de cet appel système : .....

quel est le type de protocole niveau 4 : .....

```
int sent = 0;
```

```
sent = connect(client, (SOCKADDR *)&server, sizeof(server));
```

```
if (sent == SOCKET_ERROR) cout << "\nerror connect : " << WSAGetLastError() << endl;
```

quel est le rôle de cet appel système : .....

Par quel entité est-il généré (client ou serveur) : .....

```
if ((client_socket=accept(ListenSocket, (struct sockaddr*)&address,
                           (int*) &addrlen)) < 0)
{
    perror("accept");
    exit(EXIT_FAILURE);
}
```

quel est le rôle de cet appel système : .....

Par quel entité est-il généré (client ou serveur) : .....

Est-il bloquant : .....

```
for (int i = 0; i < MAX_CLIENTS; i++)
{
    s = client_sockets[i];
    if (FD_ISSET(s, &readfds))
    {
        char recvbuf[DEFAULT_BUFLen];
        iResult = recv(s, recvbuf, recvbuflen, 0);
        ....
    }
}
```

Que représente s : .....

De quel type de serveur s'agit-il : .....

Que représente iResult : .....

Quelle est la taille du buffer de réception : .....

Que se passe-t-il si sa taille est inférieure au volume des données  
reçues : .....

.....

```
for (int i = 0; i < MAX_CLIENTS; i++)
{
    s = client_sockets[i];
    if (s > 0) FD_SET(s, &readfds);
}
active = select(0, &readfds, NULL, NULL, NULL);
```

quel est le rôle de cet appel système : .....

## Correction

Ex 1

Q1: A, B, C

Q2: A, C, E

Q3: B, C, E

Q4: A, B

2.

Tables de configuration de chaque commutateur :

Switch 1

Port	Mode	VLAN
P1	trunk	A, B, C
P2	trunk	A, B, C
P3	trunk	A, B, C
P4	access	A
P5	access	A
P6	access	A
P7	trunk	A, B

Switch 2

Port	Mode	VLAN
P1	trunk	A, B, C
P2	access	B
P3	access	B
P4	access	B
P5	trunk	B, C

Switch 3

Port	Mode	VLAN
P1	trunk	A, B, C
P2	access	C
P3	access	C
P4	access	C
P5	access	C
P6	trunk	B, C

3.

Dans l'algorithme RSA, on choisit les deux entiers premiers  $p = 17$  et  $q = 11$  :

$n = 187$ ,

$\Phi = (p-1)*(q-1) = 160$

On choisit  $e = 7$

En déduire le plus petit  $d$  correspondant.

$d = 23$

Clé publique : (187, 7)

Clé privée : (187,23)

L'auteur du code rend aussi publique une correspondance entre lettres et chiffres afin de coder des messages textuels il utilise les entiers naturels (a codé par 01, b codé par 02, etc.) l'espace entre deux mots étant codé par 00. Sachant qu'il transmet le message « encore des maths », donner la suite numérique obtenue :

**05140315180500040519001301200819**

Regrouper en tranches de chiffres conformément à l'algo RSA et donner la suite correspondante : on regroupe en tranches de chiffres inférieures à 187

**051 40 31 51 80 50 0040 51 90 0130 120 081 9**

Il suffit d'élever chaque nombre à la puissance 7 et d'en prendre le reste par la division par 187. On complète les nombre obtenus pour obtenir une suite de nombres sur 3 chiffres.

#### 4. Matching

Col.1	Col.2
1	J
2	G
3	H
4	I
5	F
6	D
7	B
8	A

#### 5. Routage IP (2 points)

Table de routage de R4 :

Réseau de destination	Gateway	Subnet mask (/n)	metric
117.0.0.0	0.0.0.0	/8	0
112.0.0.0	0.0.0.0	/8	0
205.254.1.0	112.255.255.253	/24	1
205.254.133.0	112.255.255.253	/24	1
198.199.0.0	112.54.149.250	/24	1
175.110.0.0	112.54.149.250	/16	1
215.192.18.0	112.54.149.250	/24	2
0.0.0.0	0.0.0.0	/32	0

#### 6.

Tableau complété (IP min et IP max) :

Adresse IP	Masque	IP min	IP max
131.107.56.25	/21	131.107.56.1	131.107.63.254
148.53.66.7	/20	148.53.64.1	148.53.79.254
198.53.24.100	/26	198.53.24.65	198.53.24.126
198.54.64.7	/24	198.54.64.1	198.54.64.254



7. (2 points)

Table de R8 :

Routeur	Coût	Routeur précédent la destination
R1	5	R2
R2	4	R5
R3	4	R8
R4	5	R2
R5	2	R8
R6	2	R8
R7	3	R8

8.

```
if ((ListenSocket=socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) == INVALID_SOCKET)
    cout << "socket error : " << WSAGetLastError() << endl;
```

quel est le rôle de cet appel système : réservation zone mémoire socket

quel est le type de protocole niveau 4 : TCP

```
int sent = 0;
```

```
sent = connect(client, (SOCKADDR *)&server, sizeof(server));
```

```
if (sent == SOCKET_ERROR) cout << "\nerror connect : " << WSAGetLastError() << endl;
```

quel est le rôle de cet appel système : connecter le client au serveur.

Par quel entité est-il généré (client ou serveur) : client

```
if ((client_socket=accept(ListenSocket, (struct sockaddr*)&address,
                           (int*) &addrlen)) < 0)
{
    perror("accept");
    exit(EXIT_FAILURE);
}
```

quel est le rôle de cet appel système : connecte le serveur au client.

Par quel entité est-il généré (client ou serveur) : serveur

Est-il bloquant : oui

```
for (int i = 0; i < MAX_CLIENTS; i++)
{
    s = client_sockets[i];
    if (FD_ISSET(s, &readfds))
    {
        char recvbuf[DEFAULT_BUFLen];
        iResult = recv(s, recvbuf, recvbuflen, 0);
        ....
    }
}
```

Que représente s : identifiant de socket client

De quel type de serveur s'agit-il : serveur parallèle (multi-client)

Que représente iResult : nombre d'octets reçus

Quelle est la taille du buffer de réception : **DEFAULT\_BUFLen**

Que se passe-t-il si sa taille est inférieure au volume des données reçues : ..les données sont découpées en autant de paquets que nécessaires, chacun ayant une taille de **DEFAULT\_BUFLen octets, sauf le dernier qui contient le reliquat**

```
for (int i = 0; i < MAX_CLIENTS; i++)
{
    s = client_sockets[i];
    if (s > 0) FD_SET(s, &readfds);
}
active = select(0, &readfds, NULL, NULL, NULL);
```

quel est le rôle de cet appel système : déclenché par toute activité sur les sockets