Version avec correction

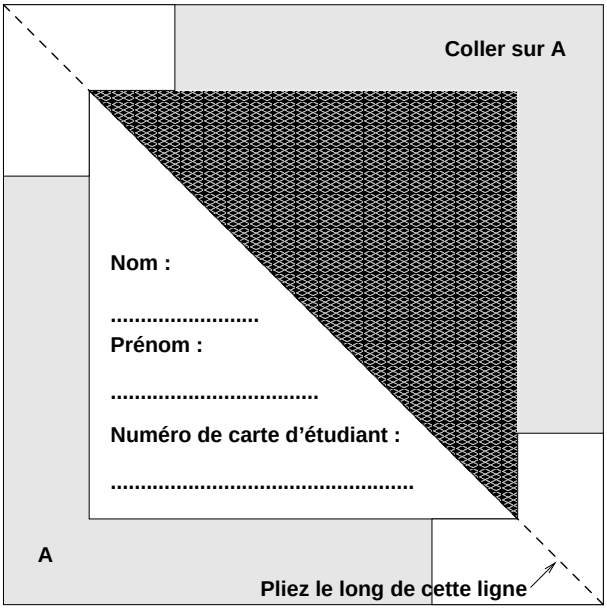
MASTER UPMC INFORMATIQUE 1ÈRE ANNÉE

Partiel U.E. RES

Durée totale: **2h00**
Nombre de pages total: **9**
Documents: **non autorisés**
Calculatrices: **non autorisées**

Avant de commencer, veuillez indiquer vos **Nom**, **Prénom** et **Numéro de carte d'étudiant** dans le cadre de chaque feuille du présent document. Pliez ensuite selon la ligne indiquée et collez avec quelques points de colle (pas de trop... il doit être possible de décacheter facilement après la correction de la copie !)

Vous devez notez vos réponses directement sur ce sujet.



Question 1 *(6 points)*

On souhaite réaliser le transfert d'un fichier `index.html` à partir d'un serveur `S` vers un client `C` selon le protocole applicatif FTP. On suppose qu'un programme FTP client est exécuté sur `C`, que le programme FTP serveur est lancé sur `S` et que le fichier est disponible dans le répertoire de l'utilisateur ayant pour identifiant `alice` et mot-de-passe `hfsdk` sur `S`.

1. Dessinez un schéma général en précisant les numéros de ports utilisés :

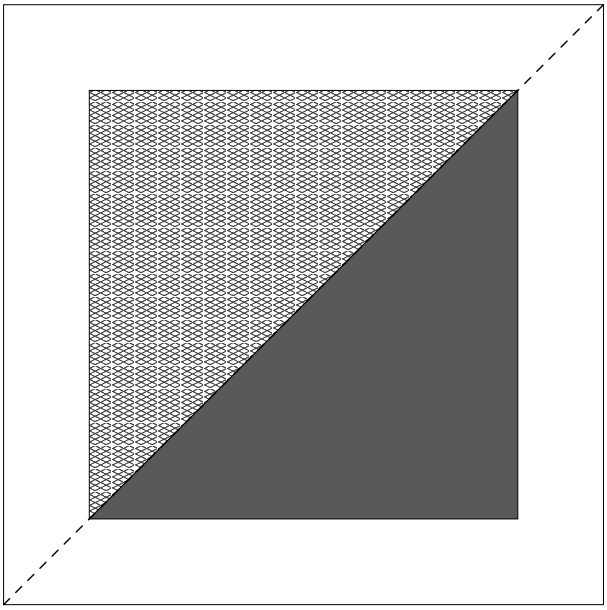
Correction (1.5 pts) : (0.5 pts schéma Client Serveur, 0.5 pts contrôle+data, 0.5 pts port)

```
          Client FTP                                    serveur FTP
      +-----------------------+                  +-------------------------+
      | Interface utilisateur |                  |                         |
      | controle FTP client <port dyn>---TCP---<21> controle FTP serveur  |
      | transfert donnée    <port dyn>---TCP---<20> transfert de donnée   |
      | (systéme de fichier)  |                  |   (système de fichier) |
      +-----------------------+                  +-------------------------+
```

2. Représentez le chronogramme des échanges applicatifs pour la session FTP récupérant le fichier nommé `index.html` (vous n'avez pas besoin de préciser les noms exacts des commandes ou les codes des réponses) :

Correction (1.5pts) : (0.5 pts identification avec USER + PASS, 0.5 pts pour la commande PORT avant RETR, 0.5 pts la connexion data entre deux message de contrôle du serveur)
Les valeurs entre parenthèse ne sont pas demandées mais peuvent compter en plus dans l'évaluation de la réponse. **Ne pas tenir compte des noms exacts**. Si les commandes SYST, TYPE et QUIT sont indiquées, c'est aussi bonus :

```
          Client FTP                          serveur FTP
               <---- (200) Ok ----------------------
               ----------- USER alice ------------->
               <---- (331) password required -------
               ----------- PASS hfsdk ------------->
               <---- (230) user alice logged -------
               ----------- PORT IP_addr_C, Port_FTP_data_C >
               <---- (200) port OK -----------------
               ----------- RETR index.html -------->
               <---- (150) Open data connexion -----
               ............ Echange data ...........
               <---- (226) transfert completed -----
```



3. Décrire le chronogramme des échanges au niveau de la couche transport :

Correction (2pts) : Je n'ai pas indiqué tous les ACK envisageables, plusieurs solutions sont possibles, par exemple après USER, un ACK peut revenir si le serveur ne répond pas assez vite, de même l'entrelacement entre la connexion contrôle et data.

```
Client FTP                                  serveur FTP
     ------------- SYN ------------------> 21
     <------------ SYN + ACK -------------- 21
     ------------- ACK ------------------> 21
     <---- (200) Ok ---------------------- 21
     ------------- ACK ------------------> 21
     ------------ USER alice ------------> 21
     <---- (331) password required ------- 21
     ------------- ACK ------------------> 21
     ------------ PASS hfsdk ------------> 21
     <---- (230) user alice logged ------- 21
     ------------- ACK ------------------> 21
     ------------ PORT IP_addr_C, Port_FTP_data_C > 21
     <---- (200) port OK ----------------- 21
     ------------- ACK ------------------> 21
     ------------ RETR index.html --------> 21
     <---- (150) Open data connexion ----- 21
     <............ SYN ..................... 20
     ------------- ACK ------------------> 21
     ............ SYN + ACK.............> 20
     <............ ACK + DONNES........... 20
     ............ ACK ..................> 20
       ...n segments de données + ACK...   20
     <............ DONNES ................. 20
     <---- (226) transfert completed ----- 21
     ------------- ACK ------------------> 21
     ............ ACK ..................> 20
     <............ FIN ..................... 20
     ............ FIN + ACK ............> 20
     <............ ACK ..................... 20
```

4. Le type d'échange précédent implique un problème pour traverser les *firewall* : Lequel ? Comment y remédier ?

Correction (1pts) : (0.5pts filtrage connexions entrantes, 0.5pts inversion avec PASV)
Les *firewall* interdisent généralement les demandes de connexions entrantes, bloquant ainsi la connexion data crée par les commande PORT + RETR. La solution réside dans l'utilisation de la commande PASV (ouverture de la connexion data a partir du client).

Version avec correction

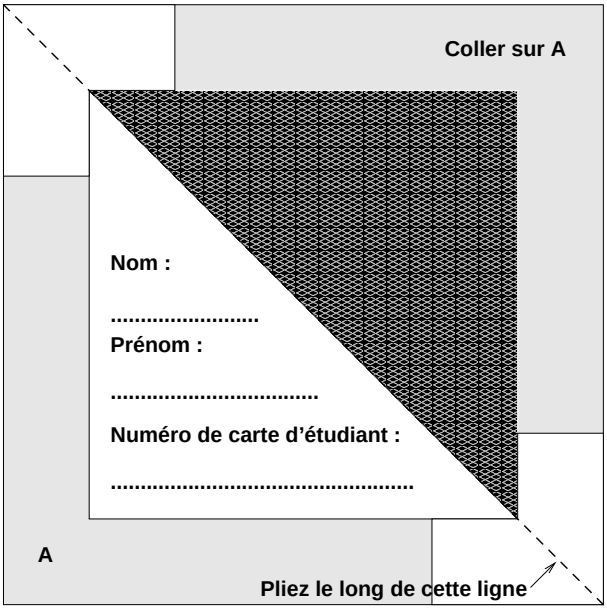
Master UPMC Informatique 1ère année

Partiel U.E. RES

Durée totale: **2h00**
Nombre de pages total: **9**
Documents: **non autorisés**
Calculatrices: **non autorisées**

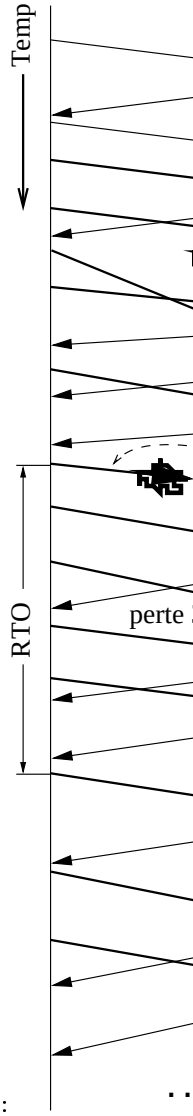
Avant de commencer, veuillez indiquer vos **Nom**, **Prénom** et **Numéro de carte d'étudiant** dans le cadre de chaque feuille du présent document. Pliez ensuite selon la ligne indiquée et collez avec quelques points de colle (pas de trop... il doit être possible de décacheter facilement après la correction de la copie !)

Vous devez notez vos réponses directement sur ce sujet.



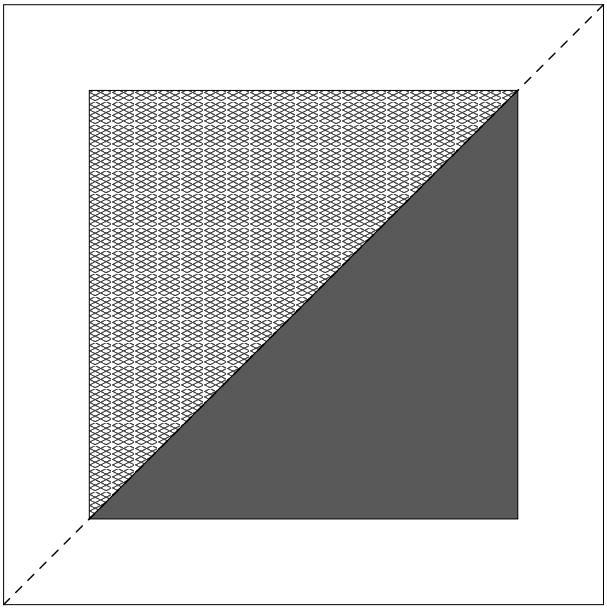
Question 2 *(7 points)*

Considérons le transfert d'un fichier de taille illimité sur une connexion TCP classique (sans *Fast retransmit*). La taille maximum des données d'un segment (MSS) est 200 octets. La transmission démarre, voici le chronogramme : complétez les paramètres associés à chaque segment (Numéros de séquence et d'acquittement, *flags* et Taille des données).



TCP:	No Sequence	No Acquit.	U	A	P	R	S	F	Taille données
→	3999	–						x	0
←	122	4000	x					x	0
→	4000	123	x						0
→	4000	123	x						200
←	123	4200	x						0
→	4200	123	x						200
←	123	4400	x						0
→	4600	123	x						200
←	123	4400	x						0
→	4400	123	x						200
←	123	4800	x						0
→	4800	123	x						200
←	123	5000	x						0
→	5000	123	x						200
→	5200	123	x						200
←	123	5000	x						0
→	5400	123	x						200
→	5600	123	x						200
←	123	5000	x						0
→	5800	123	x						200
←	123	5000	x						0
→	5000	123	x						200
←	123	5400	x						0
→	5400	123	x						200
←	123	6000	x						0
→	5600	123	x						200
←	123	6000	x						0

(4pts) :



Après avoir rempli le schéma précédent, répondez aux quelques questions suivantes :

1. Qu'implique la première perte au niveau des acquittements suivants ?

   Correction (0.5pts) : Aucune impact, grâce aux acquittements cumulatifs, la perte est compensée par l'acquittement suivant.

2. Le déséquencement observé modifie-t-il les émissions de segments de données ?

   Correction (0.5pts) : Non, grâce à la fenêtre d'anticipation, les émissions continuent sans se préoccuper des aléas de réception (l'important étant que les données soient acquitées avant l'expiration du RTO des segments émis).

3. Comment est détectée la première perte d'un segment de données et comment est elle réparée ?

   Correction (0.5pts) : Elle est détectée par l'expiration du RTO puis réparée par la retransmission du premier segment de données perdu.

4. Comment est détectée la seconde perte d'un segment de données et comment est elle réparée ?

   Correction (0.5pts) : Elle n'est pas détectée explicitement, elle est visible par la valeur de l'acquitement du premier segment retransmis, puis elle est récupérée grâce au Go-Back-N.

5. Le *slow start* aide-t-il à limiter les retransmissions ?

   Correction (0.5pts) : Oui, ici après la perte, comme on passe a un MSS, un seul segment est renvoyé. Si d'autres segments succedant immédiatement celui perdu étaient bien arrivés de manière contigue, il seront forcément acquitté avec l'approche cumulative. *On ne demandait pas d'expliquer le controle de congestion, evidement si on limite le trafic on limite l'émission mais pas forcément les retransmission.*

6. Y-a-t'il des retransmissions inutiles ?

   Correction (0.5pts) : Oui, quand la fenêtre passe à 2 MSS, on retransmet deux segments dont un inutile, ces retransmissoin sont liées à la fenêtre d'anticipation supérieure a 1 MSS et aux acquitements cumulatifs.

Version avec correction

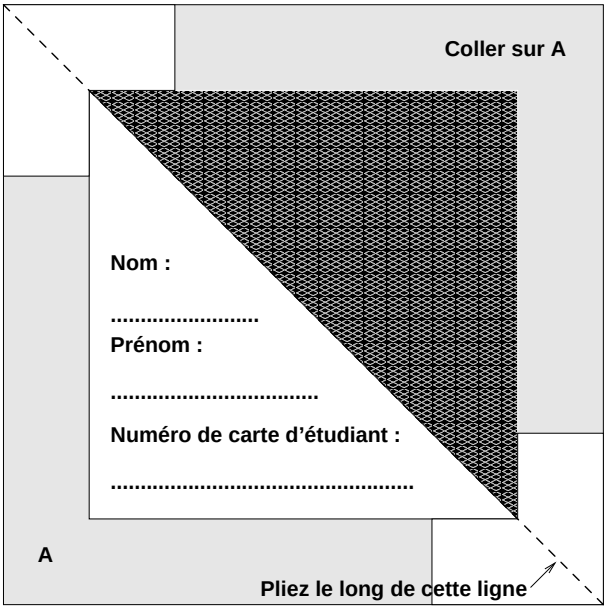
Master UPMC Informatique 1ère année

Partiel U.E. RES

Durée totale: **2h00**
Nombre de pages total: **9**
Documents: **non autorisés**
Calculatrices: **non autorisées**

Avant de commencer, veuillez indiquer vos **Nom**, **Prénom** et **Numéro de carte d'étudiant** dans le cadre de chaque feuille du présent document. Pliez ensuite selon la ligne indiquée et collez avec quelques points de colle (pas de trop... il doit être possible de décacheter facilement après la correction de la copie !)

Vous devez notez vos réponses directement sur ce sujet.

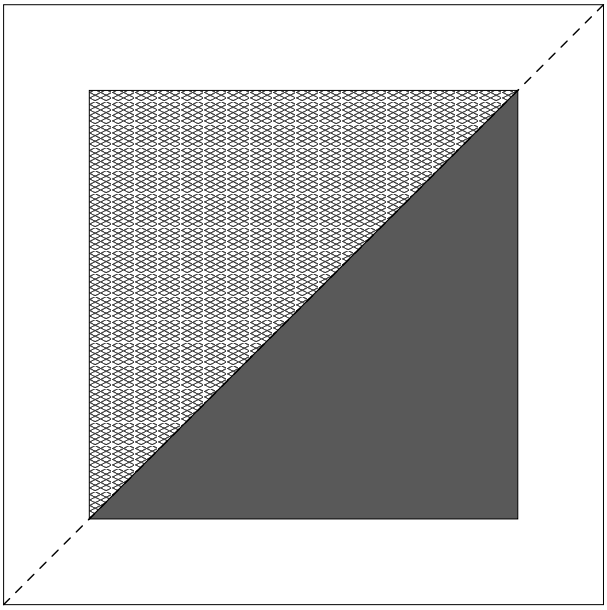


Question 3 *(7 points)*

Voici la capture en représentation hexadécimale d'une trame Ethernet sans préambule (similaire a celles observées en TME). Veuillez délimiter les différentes structures protocolaires directement sur la trace ci-dessous (toutes les informations dont vous avez besoin sont fournis en annexe). Puis répondez aux questions au verso.

Correction (4pts) :

```
      _____MAC_dst_____ _____MAC_src______ EType|vh ts
0000  00 20 ed 87 fd e6 00 07  e9 0c 90 62 08 00|45 00   . ...... ...b..E.
      iplen fr_id no_fr tl udp chsum ip_src_addr ip_dst
0010  00 b3 37 f3 00 00 3f 11  b2 04 84 e3 4a 02 84 e3   ..7...?. ....J...
      _addr|upsrc updst u_len  chsum|dnsid dnsop nques
0020  3d 7a|00 35 80 1f 00 9f  5f c2|57 8b 81 80 00 01   =z.5.... _.W.....
      n_rep n_aut n_add *__q___o__s__m__o__s__#__f__r_
0030  00 01 00 02 00 03|06 71  6f 73 6d 6f 73 02 66 72   .......q osmos.fr
     ___ __mx_ _inet __>*_ ___mx_ _inet __258968s__ __
0040  00 00 0f 00 01|c0 0c 00  0f 00 01 00 03 f3 98 00   ........ ........
     _4o pri10 __>*_ __>*_ ___ns_ _inet ___96541s__ __
0050  04 00 0a c0 0c|c0 0c 00  02 00 01 00 01 79 1d 00   ........ .....y..
     22o ___p__r__o__m__e__t___h__e__e__.__i__p__v__6_
0060  16 09 70 72 6f 6d 65 74  68 65 65 04 69 70 76 36   ..promet hee.ipv6
     _.__l__i__p__6____>#_ __>*__ __ns_ _inet __96541s
0070  04 6c 69 70 36 c0 13|c0  0c 00 02 00 01 00 01 79   .lip6... .......y
     ___ __ns_ __>*_ __>*_ ___A__ _inet ___96541s__ __
0080  1d 00 02 c0 0c|c0 0c 00  01 00 01 00 01 79 1d 00   ........ .....y..
    _4o 217.167.255.6 _>#_ _AAAA_ _inet ___86400s__ __
0090  04 d9 a7 ff 06|c0 37|00  1c 00 01 00 01 51 80 00   ......7. .....Q..
     16o __2001:0660:3302:282b:0a00:20ff:fe87:1a48____
00a0  10 20 01 06 60 33 02 28  2b 0a 00 20 ff fe 87 1a   . ..‘3.( +.. ....
     ___ _>#__ __A__ _inet ____86400s__ __4o_ 132.227.74.34
00b0  48|c0 37 00 01 00 01 00  01 51 80 00 04 84 e3 4a   H.7..... .Q.....J
     ___
00c0  22                                                "
```



1. Quel sont les protocoles contenus dans cette trame ?

Correction (0.5pts) : IPv4, UDP, DNS

2. A quelle étape d'un échange applicatif appartient cette trame ?

Correction (0.5pts) : réponse a une requête DNS

3. Quelles informations applicatives sont contenues dans cette trame ?

Correction (2pts) : Ne pas tenir compte des conversions.

```
identificateur : 0x578b
opcode: QUERY (a ne pas détailler)
QUERY: 1
ANSWER: 1
AUTHORITY: 2
ADDITIONAL: 3

;; QUESTION SECTION:
;qosmos.fr.                     IN      MX

;; ANSWER SECTION:
qosmos.fr.              258968  IN      MX      10 qosmos.fr.

;; AUTHORITY SECTION:
qosmos.fr.              258968  IN      NS      qosmos.fr.
qosmos.fr.              258968  IN      NS      promethee.ipv6.lip6.fr.

;; ADDITIONAL SECTION:
qosmos.fr.              96541   IN      A       217.167.255.6
promethee.ipv6.lip6.fr. 86400   IN      AAAA    2001:660:3302:282b:a00:20ff:fe87:1a48
promethee.ipv6.lip6.fr. 86400   IN      A       132.227.74.34
```

Annexe

Structure de la trame Ethernet

```
Trame présentée sans préambule ni CRC:

+--48-bits--+--48-bits--+16b-+-  -  -  -  -+
|  adresse  |  adresse  |type|   données    |
|destination|   source  |    |              |
+-----------+-----------+----+-  -  -  -  -+

Quelques types : 0x0200 = XEROX PUP
                 0x0800 = DoD Internet
                 0x0806 = ARP
                 0x8035 = RARP
```

Structure ARP

```
+16b-+-16b-+8b+8b+16b+--lgHW--+lgP-+--lgHW--+lgP-+
|type|type |lg|lg|Op |Emetteur|Emt.|Récept. |Rcpt|
|HW  |Proto|HW|P |   |adr. HW |adrP|adr. HW |adrP|
+----+-----+--+--+---+--------+----+--------+----+

Quelques types : 0x0001 = Ethernet
                 0x0800 = DoD Internet

Opérations : 0x0001 = Requête
             0x0002 = Réponse
```

Structure du paquet IP

```
<-----------------32bits----------------------->
<-4b->      <--8bits--><--------16bits-------->
+-----+-----+----------+---------------------+
| Ver | IHL | TOS      | Longueur totale     |
+-----+-----+----------+--+------------------+
| Identificateur        |Fl| FO               |
+-----------+-----------+--+------------------+
| TTL       | Protocole | Somme de ctrl (entête)|
+-----------+-----------+----------------------+
| Adresse Source                               |
+----------------------------------------------+
| Adresse Destination                          |
+----------------------------------------------+
...     Options
+----------------------------------------------+
...     Données
+----------------------------------------------+
```

```
Ver = Version d'IP
IHL = Longueur de l'en-tête IP (en mots de 32 bits)
TOS = Type de service
Longueur totale du paquet IP (en octets)
Fl (3 premiers bits) = indicateurs pour la fragmentation
 * 1er = Reservé
 * 2me = Ne pas fragmenter
 * 3me = Fragment suivant existe
FO (13 bits suivants) = Décalage du fragment
 * valeur a multiplier par 8 octets
TTL = Durée de vie restante

Quelques protocoles transportés :
        1 = ICMP                 8 = EGP
        2 = IGMP                11 = DoP
        4 = IP (encapsulation)  17 = UDP
        5 = Stream              36 = XTP
        6 = TCP                 46 = RSVP
```

Structure du datagramme ICMP

```
<-----------------32bits----------------------->
+-----------+-----------+----------------------+
|  Type     |  Code     | Somme de contrôle (dtg|
+-----------+-----------+----------------------+
|  Variable (généralement non utilisé)         |
+----------------------------------------------+
...     Datagramme original + 8 octets
+----------------------------------------------+

Quelques types ICMP :  0 = Demande d'écho
                       8 = Réponse d'echo
                      11 = Durée de vie écoulée
                      12 = Erreur de paramètre
```

Structure du datagramme DoP

```
<-----------------32bits----------------------->
+----------------------+----------------------+
| Port Source          | Port Destination     |
+----------------------+----------------------+
| Longueur DoP         | Version Doom         |
+----------------------+----------------------+
...     Données
+----------------------------------------------+
```

Structure de datagramme UDP

```
<-----------------32bits----------------------->
+----------------------+----------------------+
| Port Source          | Port Destination     |
+----------------------+----------------------+
| Longueur UDP         | Somme de ctrl (message)
+----------------------+----------------------+
...     Données
+----------------------------------------------+
```

Structure de segment TCP

```
<-----------------32bits----------------------->
<-4b->        <-6bits->><--------16bits-------->
+----------------------+----------------------+
| Port Source          |  Port Destination    |
+----------------------+----------------------+
| Numéro de Séquence                          |
+---------------------------------------------+
| Numéro d'Acquittement                       |
+-----+-------+---------+----------------------+
| THL |       |  Flag   |  Taille Fenêtre      |
+-----+-------+---------+----------------------+
| Somme de ctrl (message)  Pointeur d'Urgence |
+----------------------+----------------------+
...     Options
+---------------------------------------------+
...     Données
+---------------------------------------------+
```

```
THL = Longueur de l'entête TCP sur 4 bits (*32bits)
Flags = indicateur codé sur les 6 bits de droite
        (du plus faible au plus fort)
        * 0 = Fin
        * 1 = Synchronisation (SYN)
        * 2 = Réinitialisation (Reset)
        * 3 = Données immédiates (Push)
        * 4 = Acquittement (ACK)
        * 5 = Données urgentes
```

```
Options = suites d'option codées sur
 * 1 octet à 00 = Fin des options
 * 1 octet à 01 = NOP (pas d'opération)
 * plusieurs octets de type TLV
   T = un octet de type:
     2 Négociation de la taille max. du segment
     3 Adaptation de la taille de la fenêtre
     4 Autorisation des acquittements sélectifs
     8 Estampilles temporelles
   L = un octet pour la taille totale de l'option
   V = valeur de l'option (sur L-2 octets)
```

Services associés aux ports

ftp-data	20/tcp		
ftp	21/tcp		
ssh	22/tcp	ssh	22/udp
telnet	23/tcp		
smtp	25/tcp		
domain	53/tcp	domain	53/udp
		tftp	69/udp
finger	79/tcp		
www	80/tcp	www	80/udp
kerberos	88/tcp	kerberos	88/udp
pop-3	110/tcp	pop-3	110/udp
		snmp	161/udp
		snmp-trap	162/udp

Format des messages DNS

```
<2octets><2octets><2octets><2octets><2octets><2octets><--Qoctets-><-Roctets-><-Soctets-><-Ioctets->
+--------+--------+--------+--------+--------+--------+- - - - -+- - - - --+- - - - --+- - - - --+
|  Ident | Flags  | NbQuest| NbRep  | NbSAut | NbInfo | Questions | Réponses | Serv.Aut.| Info.Add.|
+--------+--------+--------+--------+--------+--------+- - - - -+- - - - --+- - - - --+- - - - --+

* Ident. = Identificateur d'échange
* Flags = Indicateurs de divers paramètres DNS (ne pas détailler)
* NbQuest = Nombre de questions
* NbRep = Nombre de champs réponses
* NbSAut = Nombre de champs désignant les serveurs DNS ayant autorité sur les réponses
* NbInfo = Nombre de champs d'informations additionnelles

Une question:

<----N-octets----><2octets><2octets>
+--- - - - - - ---+--------+--------+
|       Nom       |  Type  | Classe |
+--- - - - - - ---+--------+--------+

Un champ réponse/authorité/information:

<---M-octets---><2octets><2octets><--4octets--><2octets><----D-octets---->
+--- - - - - ---+--------+--------+------------+--------+--- - - - - - ---+
|      Nom      |  Type  | Classe |   T.T.L.   | Taille |    Données      |
+--- - - - - ---+--------+--------+------------+--------+--- - - - - - ---+

* Nom : chaque nom de label est précédé par un octet indicant le nombre de caracteres ASCII le composant.
        (si cet octet est < 63) sinon la valeur 0xC0 indique un renvoi de la valeur de l'octet suivant
        par rapport au début du message. Termine par 0x00.

* Quelques type :  1 = A (adresse IPv4)
                   2 = NS (nom de serveur DNS)
                   5 = CNAME (alias)
                   6 = SOA (zone DNS gérée)
                  12 = PTR (pointeur de nom)
                  13 = HINFO (info sur l équipement)
                  15 = MX (serveur de messagerie)
                  28 = AAAA (adresse IPv6)
                 255 = * (tous les types)

* Classe : 1 = Internet

* T.T.L. : validité en secondes

* Taille : longueur des données en octets

* Données : Nom (chaine de caractère codée comme pour une question) pour NS et CNAME
            Priorité (2 octets) puis nom (chaine de caractère codée comme pour une question) pour MX
            Adresses (valeur numérique) pour les types A (4 octets), AAAA (16 octets)...
```

Brouillon