UPMC - Licence d'Informatique
LI310 – 2010/2011

Collez votre autocollant ici :

Examen LI310 « Réseaux »
Mardi 4 janvier 2011 – Durée : 2 heures
Autorisés : 1 feuille format A4 manuscrite + Calculette

Voici 5 feuilles contenant les énoncés et les zones de réponse à compléter (sans déborder). A la fin de l'épreuve, vous devez nous rendre le tout dans une copie double d'examen vierge.
Afin de garantir l'anonymat, **vous ne devez écrire vos nom, prénom, N° de carte d'étudiant que sur la copie double** et dans le cadre réservé à cet usage.
Des autocollants avec un même numéro aléatoire vous seront distribués pendant l'épreuve : vous en collerez **un sur la copie double** et **un sur chaque feuille du sujet**.

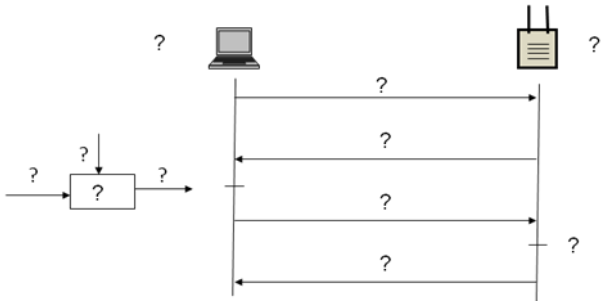
Exercice 1 : Cours (4 points)

1. Quelle est la différence entre la cryptographie à clé secrète et la cryptographie à clé publique ?

2. Quel est l'avantage de la cryptographie à clé publique ?

3. Dans les réseaux 802.11, la sécurité WEP (Wired Equivalent Privacy) se base-t-elle sur la cryptographie à clé secrète ou sur la cryptographie à clé publique ?

4. Compléter le schéma d'authentification avec WEP dans 802.11 en remplaçant les points d'interrogation (i.e. « ? ») par les éléments nécessaires.



5. Dans la sécurité WEP, quelle est l'utilité de l'ICV (*Integrity Check Value*) calculé par l'algorithme CRC-32 ?

Collez votre autocollant ici :

Exercice 2 : Transmission de Données (4 points)

Un signal porteur d'information de fréquences strictement inférieures à 10 kHz est numérisé. Chaque échantillon est quantifié en utilisant une échelle qui possède 128 niveaux quantifiés, codés en binaire. Le message numérique binaire produit est transmis à l'aide d'un code NRZ *M*-aire.

1. Quelle fréquence d'échantillonnage f_e faut-il utiliser si l'on veut minimiser le débit binaire du message numérique produit à l'issue de l'opération de numérisation ? Justifiez.

2. Quelle capacité *C* le canal de transmission devra-t-il posséder au minimum pour permettre la transmission de ce message numérique ?

3. Le canal de transmission possède une largeur de bande passante *B* égale à 90 kHz. Afin d'autoriser l'usage d'un code détecteur d'erreur, la transmission doit s'effectuer à un débit binaire D_b de 200 kbit/s. Quel est le rapport signal à bruit $(S/N)_{\text{dB}}$ minimal nécessaire en réception ?

4. En supposant le rapport signal à bruit suffisant, quelle valeur M faut-il alors choisir ? Quelle est alors la durée T_s des symboles ?

5. On souhaite augmenter la portée de la liaison. Le rapport signal à bruit en réception est alors de 4 dB. On se propose alors de diminuer le nombre n de niveaux de quantification du numérisateur de manière à ce que la transmission reste faisable. Comment adapter la valeur de n ? (On négligera dans cette question le sur-débit engendré par le codage détecteur d'erreur.)

Collez votre autocollant ici :

Exercice 3 : HDLC (4 points)

Un équipement de transmission A situé sur un satellite dialogue avec un autre équipement B au sol à l'aide d'une liaison utilisant le protocole LAP-B (variante de HDLC). Le satellite se situe à 36 000 km d'altitude. On considère que les temps de traitement des données (constitution et réception des trames) sont négligeables et que toute trame de données est instantanément acquittée.
On suppose que seul le transmetteur A situé sur le satellite envoie des données et que les trames émises en continu les unes à la suite des autres contiennent toutes un fanion de début et de fin. Toutes les trames sont acquittées individuellement.
La vitesse de propagation du signal est d'environ 3.10^8 m/s. La taille du champ de données de la trame est ici de 128 octets, l'entête du protocole est codé sur 56 bits et la liaison a un débit de 64 kbit/s.

1. Combien de temps l'émetteur A devra-t-il conserver une trame dans son buffer en émission, si l'on considère que les transmissions se font sans erreur et qu'une trame est transmise dès qu'elle est prête à être envoyée ?

2. Quelle doit être la taille minimale de la fenêtre d'anticipation du satellite pour que la transmission soit la plus efficace possible ?

3. Quelle relation existe-t-il entre la taille *W* de la fenêtre d'anticipation et le nombre *n* de bits sur lequel est codé le numéro de séquence de la trame ?

4. Sachant que, dans la norme HDLC, il est spécifié que le numéro de trame est codé sur 3 ou 7 bits, quelle serait ici la valeur retenue pour *W* ?

5. Compléter les deux schémas d'échanges de trames HDLC.

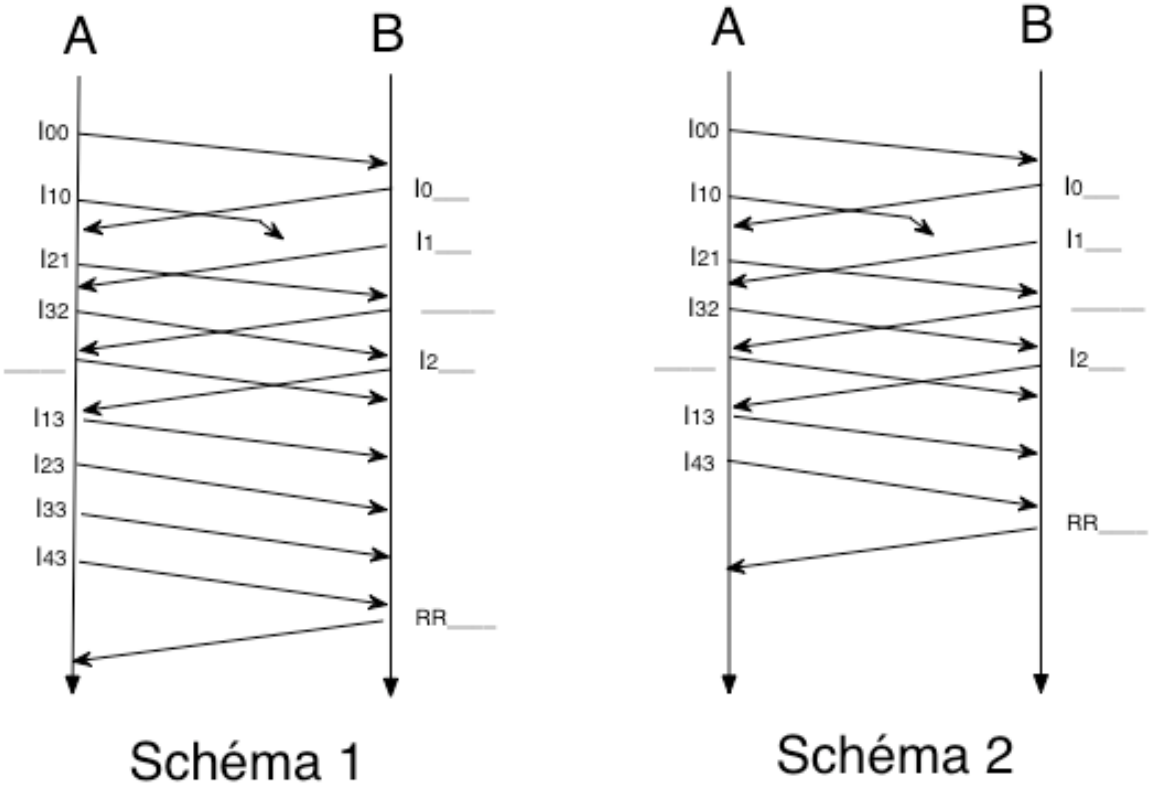


Schéma 1

Schéma 2

Collez votre autocollant ici :

Exercice 4 : Réseaux locaux et CSMA/CD (4 points)

Rappels : une station qui souhaite transmettre des données respecte les étapes suivantes :

1. Ecoute du canal pendant une durée équivalente à la transmission de 96 bits afin de s'assurer que le canal est libre.
2. Si le canal est libre alors la machine transmet ses données.
3. Si la machine détecte une collision, elle continue à émettre 32 bits (séquence de brouillage) puis cesse son émission, elle attend ensuite pendant un certain temps aléatoire (retrait exponentiel) et recommence à l'étape 1. L'algorithme de retrait exponentiel consiste à attendre un temps aléatoire compris entre 0 et Min(2^i-1, 1023) intervalles de temps élémentaires (*slot-time*) après la i-ème collision. Après 16 tentatives d'envoi échouées, l'émetteur abandonne.

Dans les questions ci-dessous on suppose que lorsqu'une station veut transmettre, le canal est toujours libre (ce qui n'empêche pas les collisions par la suite). On prendra soin d'écrire le résultat sous forme littérale avant de passer à l'application numérique :

- Débit = 10 Mbit/s
- Taille maximale du champ de données d'une trame Ethernet = 1500 octets
- Longueur du câble = 2,5 km
- Vitesse de propagation = 100 000 km/s
- Intervalle de temps élémentaire = 51,2 µs

1. A quelle durée correspond le temps d'écoute obligatoire de l'étape 1 ?

2. Quel est le temps de transmission d'une trame Ethernet de taille maximale ?

3. Combien de temps s'écoule *au minimum* entre le moment où une machine veut envoyer une trame de taille maximale et le moment où la trame est intégralement reçue par le destinataire ?

4. On suppose qu'il y a collision. Combien de temps s'écoule *au maximum* entre le moment où une machine veut envoyer une trame et le moment où cette machine détecte la collision ?

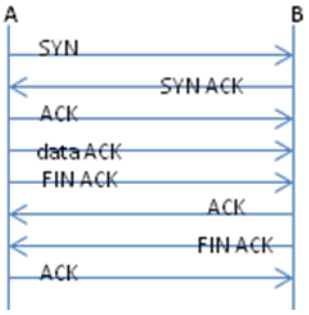
5. Combien de temps s'écoule *au maximum* entre le moment où une machine veut envoyer une trame et le moment où la trame est intégralement reçue par le destinataire en supposant qu'il y a une seule collision ?

6. On suppose qu'il y a collision à chaque envoi. Combien de temps s'écoule *au maximum* entre le moment où la machine veut envoyer une trame et le moment où la machine abandonne (16 échecs) ?

Collez votre autocollant ici :

Exercice 5 : TCP (4 points)

Vous trouverez au verso une trace obtenue par un analyseur réseau et correspondant à l'échange TCP suivant :



1. Quel est le numéro de séquence initial (ISN) de A ? Quel est le numéro de séquence initial (ISN) de B ? (Vous pouvez laisser les valeurs en hexadécimal.)

2. Dans la trace, le segment de données (4ème trame) a été tronqué. Combien d'octets de données transporte-t-il ? Justifiez.

3. Complétez directement sur la trace les valeurs des champs de numéros de séquence et de numéros d'acquittement. (La 1ère colonne correspond à la numérotation des octets.)

```
0000  00 15 17 78 86 b8 00 15 17 50 93 e6 08 00 45 08   ...x.....P....E.
0010  00 3c e6 b3 40 00 40 06 3c e1 0a 0f 02 01 0a 0f   .<..@.@.<.......
0020  01 01 00 14 b8 6b dc a5 2f 97 00 00 00 00 a0 02   .....k../.......
0030  16 d0 8c 30 00 00 02 04 05 b4 04 02 08 0a 00 00   ...0............
0040  c9 24 00 00 00 00 01 03 03 06                     .$........
```

```
0000  00 15 17 50 93 e6 00 15 17 78 86 b8 08 00 45 00   ...P.....x....E.
0010  00 3c 00 00 40 00 40 06 23 9d 0a 0f 01 01 0a 0f   .<..@.@.#.......
0020  02 01 b8 6b 00 14 2d 00 2a 58             a0 12   ...k..-.*X../...
0030  16 a0 61 b0 00 00 02 04 05 b4 04 02 08 0a 00 58   ..a............X
0040  d2 ee 00 00 c9 24 01 03 03 06                     .....$....
```

```
0000  00 15 17 78 86 b8 00 15 17 50 93 e6 08 00 45 08   ...x.....P....E.
0010  00 34 e6 b4 40 00 40 06 3c e8 0a 0f 02 01 0a 0f   .4..@.@.<.......
0020  01 01 00 14 b8 6b                       80 10   .....k../.-.*Y..
0030  00 5c a6 bf 00 00 01 01 08 0a 00 00 c9 24 00 58   .\...........$.X
0040  d2 ee
```

```
0000  00 15 17 78 86 b8 00 15 17 50 93 e6 08 00 45 08   ...x.....P....E.
0010  05 8c e6 b5 40 00 40 06 37 8f 0a 0f 02 01 0a 0f   ....@.@.7.......
0020  01 01 00 14 b8 6b                       80 18   .....k../.-.*Y..
0030  00 5c 83 74 00 00 01 01 08 0a 00 00 c9 24 00 58   .\.t.........$.X
0040  d2 ee 74 6f 74 61 6c 20 39 32 0d 0a 2d 72 77 2d   ..total 92..-rw-
(...)                                 (...)                (...)
0590  6e 2d 65 72 72 6f 72 73 0d 0a                     n-errors..
```

```
0000  00 15 17 78 86 b8 00 15 17 50 93 e6 08 00 45 08   ...x.....P....E.
0010  00 34 e6 b6 40 00 40 06 3c e6 0a 0f 02 01 0a 0f   .4..@.@.<.......
0020  01 01 00 14 b8 6b                       80 11   .....k..4.-.*Y..
0030  00 5c a1 66 00 00 01 01 08 0a 00 00 c9 24 00 58   .\.f.........$.X
0040  d2 ee                                             ..
```

```
0000  00 15 17 50 93 e6 00 15 17 78 86 b8 08 00 45 08   ...P.....x....E.
0010  00 34 f8 b7 40 00 40 06 2a e5 0a 0f 01 01 0a 0f   .4..@.@.*.......
0020  02 01 b8 6b 00 14                       80 10   ...k..-.*Y..4...
0030  00 88 a1 3b 00 00 01 01 08 0a 00 58 d2 ee 00 00   ...;.......X....
0040  c9 24                                             .$
```

```
0000  00 15 17 50 93 e6 00 15 17 78 86 b8 08 00 45 08   ...P.....x....E.
0010  00 34 f8 b8 40 00 40 06 2a e4 0a 0f 01 01 0a 0f   .4..@.@.*.......
0020  02 01 b8 6b 00 14                       80 11   ...k..-.*Y..4...
0030  00 88 a1 39 00 00 01 01 08 0a 00 58 d2 ee 00 00   ...9.......X....
0040  c9 24                                             .$
```

```
0000  00 15 17 78 86 b8 00 15 17 50 93 e6 08 00 45 08   ...x.....P....E.
0010  00 34 e6 b7 40 00 40 06 3c e5 0a 0f 02 01 0a 0f   .4..@.@.<.......
0020  01 01 00 14 b8 6b                       80 10   .....k..4.-.*Z..
0030  00 5c a1 64 00 00 01 01 08 0a 00 00 c9 25 00 58   .\.d.........%.X
0040  d2 ee                                             ..
```

Annexe

Structure d’une trame Ethernet

```
........+-48bits--+-48bits--+16b-+- - - - +......
.(Pré.) | adresse | adresse |type| données  |(CRC).
.       | dest.   | source  |    |          |     .
........+---------+---------+----+- - - - +......

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

Structure d’un paquet IP

```
 <--------------------32bits--------------------->
 <4b->       <--8bits---> <--------16bits------->
+-----+-----+------------+----------------------+
| Ver | IHL | TOS        | Longueur totale (octet)
+-----+-----+------------+--+-------------------+
| Identificateur         |Fl| FO                |
+-----------+------------+--+-------------------+
| TTL       | Protocole  | Somme de ctrl (entête)|
+-----------+------------+-----------------------+
| Adresse Source                                 |
+------------------------------------------------+
| Adresse Destination                            |
+------------------------------------------------+
...                  Options                   ...
+------------------------------------------------+
...                  Données                   ...
+------------------------------------------------+
Ver = Version d’IP
IHL = Longueur de l’entête IP (en mots de 32 bits)
TOS = Type de service (zero généralement)
Fl (3 premiers bits) = Bits pour la fragmentation
 * 1er = Reservé
 * 2me = Ne pas fragmenter
 * 3me = Fragment suivant existe
FO (13 bits suivants) = Position relative du fragment dans le
datagramme initial, le déplacement étant exprimé en mots de 8
octets (seuls un datagramme complet ou un premier fragment
peuvent avoir ce champ à 0)
TTL = Durée de vie restante

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

Structure d’un segment TCP

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

THL = Longueur de l’entête TCP sur 4 bits (*32bits)
Flags = indicateur codé sur 6 bits gauche à droite
   * 1er = Données urgentes
   * 2me = Acquittement (ACK)
   * 3me = Données immédiates (Push)
   * 4me = Réinitialisation (Reset)
   * 5me = Synchronisation (SYN)
   * 6me = Fin
Options = suite d’options 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
bgp	179/tcp		
		snmp	161/udp
		snmp-trap	162/udp
rtraceroute	3765/tcp		