TD1

# Exo 1

Voir feuille

# Exo 2

1. 300 Go = 2400000 Mbps  
   1 jour = 86400 sec  
   débit nécessaire = 2400000/86400 = 27.7 Mbps  
   on dispose que d’un Mbps en upload donc insuffisant
2. Temps pour écouler des 2 To  
   2 To/200 Mbps = 2 \* 8x106 Mb/200 Mbps = 22h  
   Le quota du mois est atteint au bout de 22h
3. Vitesse lumière = 200000 km/s  
   temps pour 1 bit pour traverser = 200 km/200000km/s = 1ms  
   10 Gb/s = 1.25 Go/s  
   Donc en 1 ms on envoie 1.25 Go x10-3 = 1.25 Mo

# Exo 3

Clés USB 32 Go  
temps de vol d’une clé = environ 1 seconde  
temps de préparation = environ 2 secondes  
temps de réception = environ 1 seconde

1. Latence = 1 + 2 + 1 = 4 s  
   Débit = 32 Go toutes les deux secondes = 16 Go/s
2. Temps de vol = 20 s  
   Latence = 20 + 2 + 1 = 23 s  
   Débit = 32 Go / 2s = 16 Go/s
3. Temps de préparation = 0.5s  
   Latence = 20 + 0.5 + 1 = 21.5 sec  
   Débit = 32 Go / 1 s car récepteur plus lent que l’émetteur
4. Vitesse de transfert entre pc et clé = 10 Mo/s

Ordi 1 fou 1 fou 2 ordi 2

|  |  |  |
| --- | --- | --- |
| D1 : 10 Mo/s | D2 : 64 Go/s | D3 : 32 Go/s |
| L1 : 3200 s | L2 : 20.5 s | L3 : 1 s |

Débit global = min {D1, D2, D3, …}  
Latence globale = L1 + L2 + L3 + …

# Exo 4

Connexion ADSL : 2 Mbps / 256 Kbps  
Connexion parents : 18 Mbps / 860 Kbps  
disque USB 1.0 = débit max de 1 Mo/s

1. Latence négligeable
2. Débit = min = 256 Kbps  
   temps de transfert = 1 Go/256 Kbps = 30000 s = environ 10h  
   Débit global = 860 Kbps = 100 Ko/s  
   Temps = 10000 s = 1h30
3. 1 Ko/temps de transfert = 1 Ko/32 Ko/s = 30 ms IL FAUT TENIR COMPTE DE LA LATENCE

TD2

# Exo 1

1. 192.168.122.17/24 (24 « 1 » suivi de 0)  
   192.168.122.3/24  
   192.168.113.1/24  
     
   Deux premiers dans le réseau  
     
   En /16, tous dans le réseau
2. ZZZZZZZZZZZZZZ
3. On peut faire 256 sous réseaux  
   254 machines pour chaque sous réseau (28 – 2 = 254 | -2 -> adresse réseau et broadcast)

# Exo 2

1. 1. 00101101 -> 45
   2. 11100000 -> 128 + 64 + 32 = 224

10000000 -> 128  
11000000 -> 192  
… -> 224  
… -> 240  
… -> 248  
… -> 252  
… -> 254  
… -> 255

* 1. 00100001 -> 00111110  
     33 -> 62

1. Réseau 96 | Masque 240  
   1. 96 -> 01100000
   2. 240 -> 11110000  
      Avec un masque de longueur 4 + 1 on ne peut définir que deux sous réseaux, il faut donc un masque de longueur 4 + 2 = 6 pour définir 4 >= 3 sous réseaux  
      Sous réseau 1 : 01100000  
      Sous réseau 2 : 01100100  
      Sous réseau 3 : 0101000
   3. Il reste 8 – 6 = 2 bits pour la partie machine   
      On a donc 22 – 2 adresses disponibles

# Exo 3

1. 164.32.120.18 / 20  
   1. 255.255.240.0
   2. Adresse réseau = adresse IP ET masque  
      164.32.?.0  
      120 = 01111000  
      240 = 11110000  
      donc 120 ET 240 = 01110000 = 112  
      Adresse = 164.32.112.0

|  |  |  |
| --- | --- | --- |
| TABLE DE VERITE DU ET « bit a bit » | | |
| ET | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |

* 1. 164.32.112.1 -> 164.32.127.254

1. 255.255.252.0 = 8 + 8 + ? + 0  
   252 = 128 + 64 = 192 + 32 = 224 + 16 = 240 + 8 = 248 + 4 = 252  
   -> 11111100  
   donc 8 + 8 + 6 + 0 = 22
2. Besoin de 5 bits en plus car 24 = 16 < 20 <= 25 = 32  
   ça nous donne un masque de longueur 25 et donc 32 – 25 = 7 bits pour les adresses machine  
   c’est insuffisant car 27 -2 = 128 – 2 = 126 < 1000

# Exo 4

1. 2001 : 0660 : 6101 : 0000 : 0000 : 0000 : 0000 : 0000 / 48  
   = 2001 : 660 : 6101 : : / 48  
   1. ffff : ffff : ffff : ffff : :
   2. 16 bits (ou wxyz) pour identifier mes sous réseaux, donc 216 = 65536 sous réseaux
   3. 128 – 64 bits pour identifier mes machines  
      Donc on a 264 = 1019 adresses

Remarque : En IPv6, il n’y a pas d’adresse réseau/broadcast

* 1. Il y a 248 adresses réseaux du même type que celle d’UB

TD 3

# Exo 1

1. Adresse IP source : 11.12.13.14  
   Adresse IP destinataire : 11.12.13.15
2. Port source : 15 en base 16 -> 21 en base 10 (=FTP)  
   Port destination : 401 base 16 -> 1025 base 10

# Exo 2

@IP source @IP desti  
192.168.0.4 192.168.0.1

@Ethernet @Ethernet  
destination source

#port source #port desti  
 txyz 80

data

En-tête TCP

En-tête IP

Trame 6

« GET / HTTP 1.1 »

1. Requête ARP et réponse ARP (Address Resolution Protocol)

TD 4

# Exo 1

1. Premier oui, deuxième non
2. S’il y a deux inversions de bits, on ne peut pas le détecter

# Exo 2

1. 01100011
2. S’il y a une inversion de bit dans le message on verra quelle ligne et quelle colonne a une erreur de parité, ce qui situe la position de l’inversion
3. Oui car elles peuvent pas se compenser a la fois en ligne et en colonne
4. Non car on a plusieurs lignes ou plusieurs colonnes avec erreur et on ne sait pas précisément ou corriger
5. Non parce que
6. Du coup non

# Exo 3

1. 84 56 82 12 74 45  
   0x8456 + 0x8212 = 0x10668  
   RETENUE 0x0668 + 1 = 0x0669  
   0x0669 + 0x7445 = 0x7aae  
   7aae16 = 0111 1010 1010 11102  
   Inversion des bits -> 1000.0101.0101.00012  
   -> 0x8551
2. f1 34 10 23 fe a7  
   0xf134 + 0x1023 = 0x10157 = 0x0158  
   0x0158 + 0xfea7 = 0xffff  
   -> 0x0000 après inversion
3. si on change un chiffre hexa, ça change tout le calcul tout du long jusqu’au résultat final, qui ne retombe donc plus à zero. On ne peut pas les corriger car on ne sait pas où elles sont

Pour aller plus loin

1. l’addition est commutative donc ajouter le checksum au milieu de l’en-tete ou a la fin donne la même somme au final. Et comme zéro est l’element neutre de l’addition, remplacer un champ zéro par le checksum donne aussi la même somme au final.
2. si par exemple un chiffre hexa se retrouve augmenté de 1, et que plus loin dans un autre morceau de 16 bits, un autre chiffre gexa se retrouve diminué de 1…………..

# Exo 4

1. CRC-4  
   La clé K = 0x9 = 1001  
   msg = 0010111011  
   G = 1 | K = 1|1001