

francois.laquédi@labri.fr
dept-inf@labri.fr /~laquédi/

une connection = (@ source , port source), (@ destination , port destination)

exo 1 ftp -d -p ftp.ens.lyon.fr \Leftrightarrow ftp -dp ftp.ens.lyon.fr
port par défaut pour ftp : 21
-p = passive mode pour ne pas déranger le serveur

exo 3 exemple de protocole :

A

connection = "salade"

fin connection = "tomate"

serveur FTP

182.168.54.12, port 21

B

réception connection = "salade verte"

fin réception = "tomate peurries"

meus

173.142.121.3, port 712

connection TCP

220 READY

USER anonymous

331 LOGIN OK

PASS xx@emi.fr

230 OK

transfert de fichier

SIZE Publications.lip

213 4

RETR Publications.lip

PASV

227 entering passive mode (@, 12, 26)
nouveau port = 256 x 12 + 26

on se re-connecte (depuis un nouveau terminal):

Telnet @ Nouv_port

3 modes de transfert :

- flex : les données sont encastrées en rafale et coupe la connexion
- bloc : découpage en blocs qui contiennent des informations sur l'émetteur et un message
- compressé

exo 4:

$$21 \text{ cm} = 8,2677 \text{ inches} = 2480 \text{ points}$$

$$29,7 \text{ cm} = 11,6929 \text{ inches} = 3508 \text{ points}$$

$$A4 = 2480 \times 3508 = 8699840 \text{ points}$$

$$8699840 \div 9600 = 906,5 \approx 15 \text{ min transmission} = \text{données} + \text{débit}$$

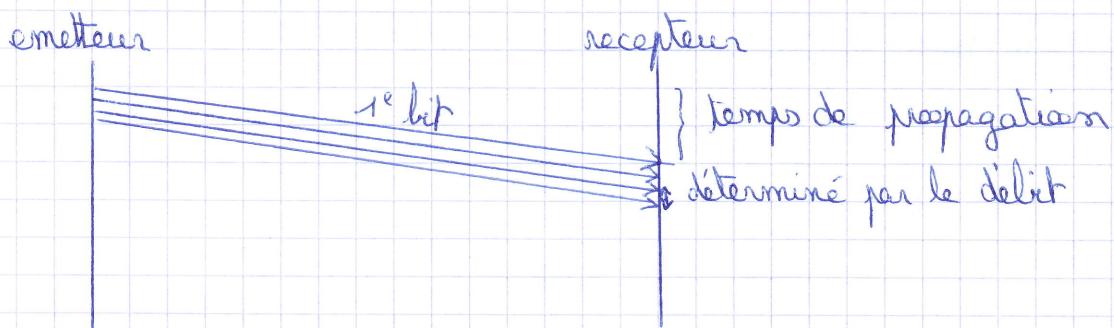
$$\text{compression} : (8699840x) \div 9600 = 1 \Rightarrow x = 1/15$$

TP2 : Transmission et protocoles

exo 1:

20 MBytes (16 bits) : délit

50 ms : temps de réponse = "latence"



Temps de propagation : $L \div V$

Temps de transmission de 512×8 = Temps de transmission du 1^e bit

+ Temps de transmission de tous les autres

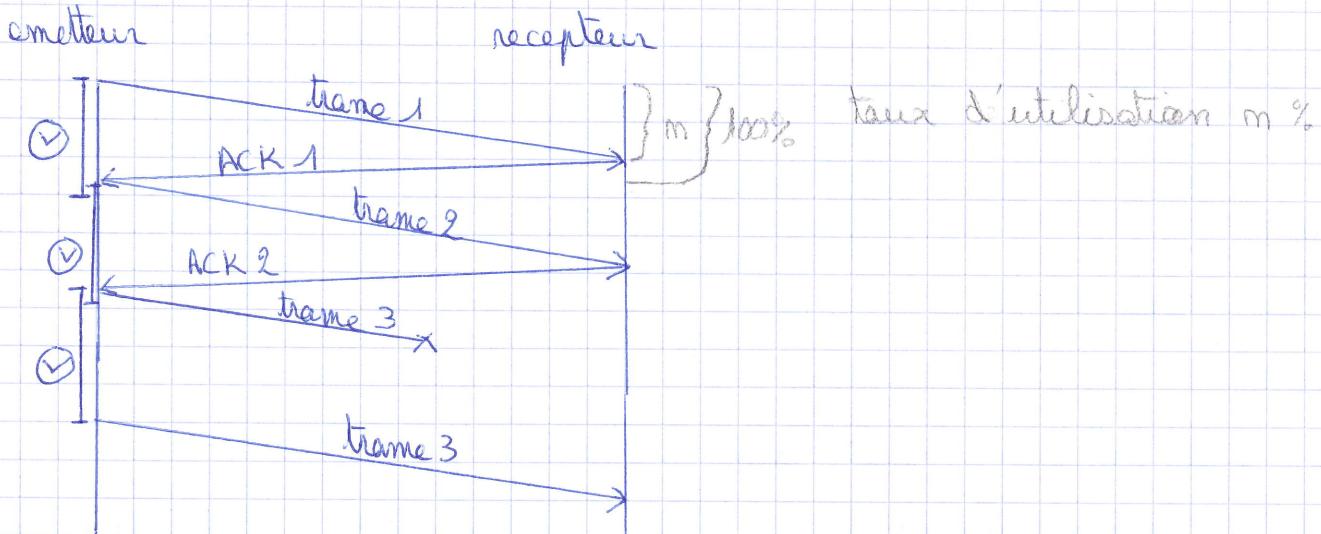
$$= \frac{L}{V} + \frac{512 \times 8}{D}$$

$$= \frac{3 \times 10^6}{2 \times 10^8} + \frac{512 \times 8}{155 \times 10^6}$$

$$= 15,026 \text{ ms}$$

avec une paire torsadée : $\frac{L}{V} + \frac{512 \times 8}{D} = 17,048 \text{ ms}$

exo 3:



le taux d'erreur est négligeable \Rightarrow on acquitte toutes les trames.

Temps de transmission d'une trame = Temps t_1 + temps ACK₁

exo 2:

Soient A et B deux événements indépendants

$$P(A \cup B) = P(A) + P(B)$$

$$P(\bar{A}) = 1 - P(A)$$

$$P(A \cap B) = P(A) \times P(B)$$

Taux d'erreur bit = nombre de bits envoiés correctement en moyenne pour 1 bit en erreur.

$= 10^{-4} \Rightarrow$ en moyenne on reçoit 10^4 bits avant d'en trouver un erroné

= probabilité qu'un bit donné soit erroné

$$P_{\text{bit erroné}} = P^{\text{"au moins 1 bit est en erreur"}}$$

$$= 1 - P^{\text{"tous les bits sont valides"}}$$

$$= 1 - (P_1 \text{ est valide} \cap P_2 \text{ est valide} \cap \dots \cap P_N \text{ est valide})$$

$$= 1 - (1 - \text{taux d'erreur bit})^N$$

$$= 0,1852$$

N : nombre de bits par trame

débit effectif = débit qui tient compte des trames perdues

18,52 trames perdues pour 100 trames envoyées

en 2s, on a effectivement $100 - 18,52 = 81,78$ trames

$$D_{\text{ligne}} = \frac{100 \times 2048}{2} = 102400 \text{ bit/s}$$

$$D_{\text{effectif}} = \frac{81,48 \times 2048}{2} = 83435 \text{ bit/s}$$

exercice 4:

telnet trelawney 2525

parler SMTP en 10 lignes :

1. Dire bonjour : HELO tata = EHLO
2. Dire depuis quelle adresse on souhaite envoyer : MAIL FROM : <tata@labri.fr>
3. Donner le destinataire du message : RCPT TO : <titi@labri.fr>
4. Prévenir le serveur qu'on commence l'édition du message : DATA <
5. Ecrire le message
Subject : ...
Date : ...
6. Terminer l'envoi par l'écriture d'un point seul sur une ligne
7. Dire au revoir : QUIT

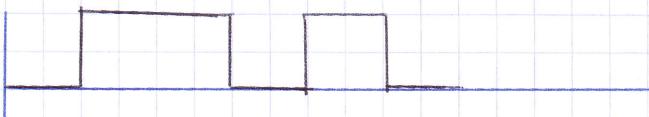
simuler un serveur : nc -v -n -l -p 6854

ouvrir une autre console : telnet localhost 6854

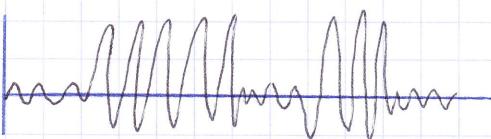
TP 3: Transmission

exercice 1:

011010

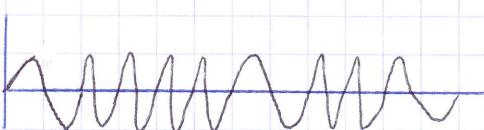


variation d'amplitude

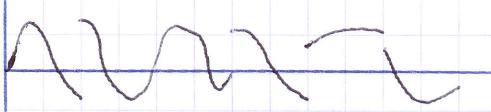


modulations sur un câble téléphonique :

variation de fréquence



phase



Nyquist : calcul du débit en conditions idéales

Shannon : calcul du débit dans la vraie vie

1. on entend les fréquences entre 2 et 20 KHz.

2. $D = 2H \log_2 V$ H = bande passante

$$= 2 \times 3100 \times \log_2 2$$

$$= 6200 \text{ bits/s}$$

V = niveaux significatifs.

on augmente V à 128 niveaux pour coder des 0 et des 1.

exo 2:

$$1. \frac{S}{N} = 30 \text{ dB} \Leftrightarrow 30 = 10 \log_{10} \left(\frac{S}{N} \right) \quad S =$$

$$3 = \log_2 \left(\frac{S}{N} \right)$$

$$10^3 = 10^{\log_{10} \left(\frac{S}{N} \right)}$$

$$\frac{S}{N} = 10^3$$

$$x = \log_2 1000$$

$$2^x = 2^{\log_2 1000} = 1000$$

$$C = 3100 \log_2 (1 + 1000)$$

$$\approx 3100 \times 10$$

$$\approx 31 \text{ kbit/s}$$

$$\text{en sait que } 2^{10} = 1024$$

3. utilisation de filtre

4. on augmente $\frac{S}{N}$ avec la distance.

exo 3

top500.org

problème de confidentialité \Rightarrow filtre

exo 4:

$$2 \text{ Mbit/s} \leftrightarrow 250 \text{ kbit/s}$$

$$250 \text{ kbit/s} \rightarrow 10$$

$$10 \text{ Gbit/s} \rightarrow 40000 \text{ s} \sim 11 \text{ h}$$

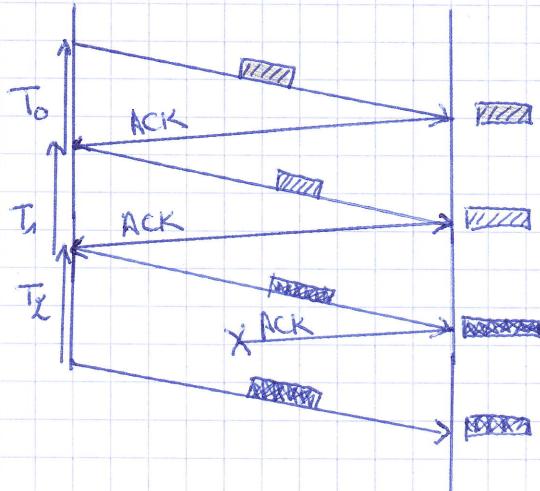
$$\text{durée} = \frac{\text{données}}{\text{débit}} = \frac{8 \times 10^8}{2 \times 10^6} = 4 \times 10^4 \text{ s} \approx 11 \text{ h}$$

TD4: Couche liaison de données (Protocoles)

La couche liaison de données s'assure que la transmission se passe bien.

header	message	queue	= frame
		paquet	tar - xref

liaison-v1.cc : implémente version simple (envoyer et attendre)



Attention de ne pas avoir un timer trop court!
=> le récepteur a des doublons

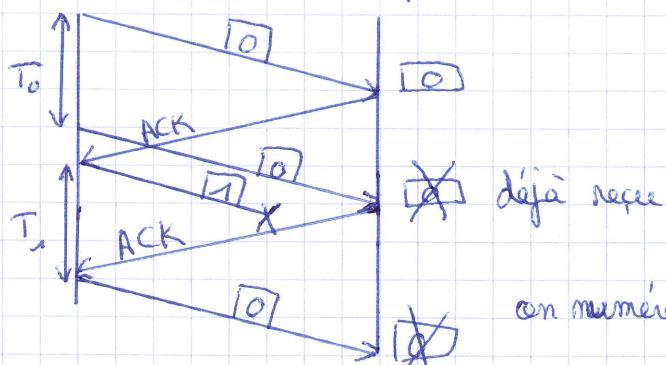
timer trop long => connexion mal utilisée

Problème : le récepteur a deux fois le même paquet si ACK perdu

liaison-v2.cc : version sans doublons en réception

on numérote les paquets

si timer trop court => on envoie tout en doublon, l'émetteur ne sait pas quel paquet est acquitté. Quand R acquitte le second paquet 0, E pense recevoir l'acquittement du premier paquet 1.



on numérote les paquets sur 1 bit.

Il faut donc aussi attendre numérotter les ACK dans liaison-v3.cc.

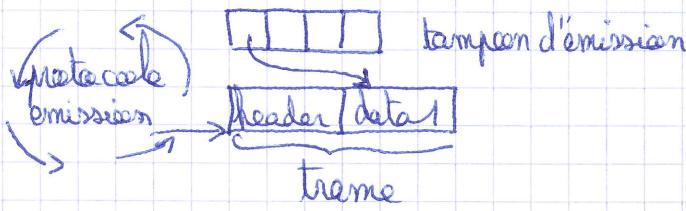
↳ version du lit alterné

Machine 0

"J'ai tenté d'émettre le paquet i"

couche application

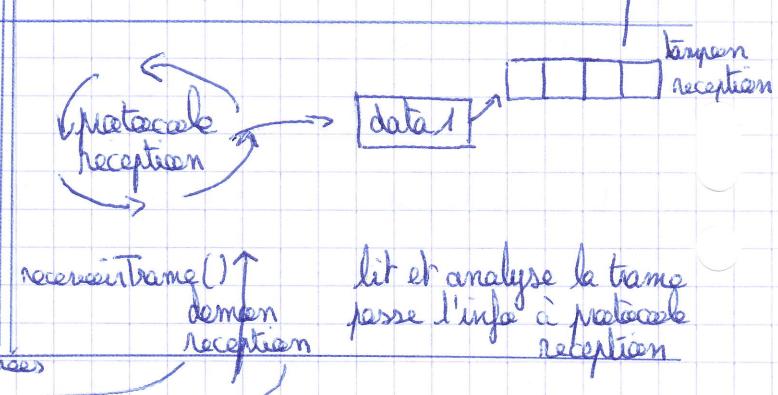
envoyerPaquet()



Machine 1

"J'ai reçu le paquet i"

recevoirPaquet()



couche liaison

envoyerTrame()

type info = données
ack

couche physique

Header|data||Header|data||

fenêtrage :

en étant la taille du tampon d'émission.

↳ envoie en rafale des paquets

problème : 1 seed bit ne suffit plus à différencier deux paquets

↳ utilisation d'un numéro de séquence.

On quitte au fur et à mesure.

⇒ On augmente la taille des tampons d'émission et de réception.

On remplace le tampon du démon réception par une syndicat ()

Treg-P () = -1 signifie qu'il n'y a plus de place dans le tampon.

TP5: Contrôle de flux

TP6: Détection et correction d'erreurs.

Distancce de Hamming: "La distance minimale entre deux mots d'un langage."

ex: soit le langage $L = \{aaa, aba, abb, bbb\}$

(aaa, aba) \rightarrow 1 différence

(aaa, abb) \rightarrow 2 différences

Donc la distance de Hamming pour ce langage est 1.

exercice 1:

1. début du code repéré par la première barre à droite.

2. code postal = 3340S

3. $d_{Hamming} = 2$.

Si d est la distance de Hamming d'un code alors

- on peut détecter $d-1$ erreurs

- on peut corriger $\frac{d-1}{2}$ erreurs (on prend la valeur entière).

On peut détecter 1 erreur.

On ne peut pas corriger d'erreur.

exercice 2:

- On veut envoier "B".

- On calcule une "somme de coefficients" notée "C".

- On envoie "BC" (C accolé à B)

- A la réception, on divise "BC" par G, polynôme générateur.

- On vérifie que le reste de la division est nul.

exemple division :

$$\begin{array}{r} 3x^3 + x^2 + 2x + 1 \\ -3x^3 - 3x^2 \\ \hline 0 \quad -2x^2 + 2x + 1 \\ +2x^2 + 2x \\ \hline 4x + 1 \\ -4x - 4 \\ \hline -3 \end{array}$$

$$\begin{array}{r} x + 1 \\ \hline 3x^2 - 2x - 4 \end{array}$$

degré de $-3 = 0$

plus haut degré de $x + 1 = 1$

$0 < 1$ donc on s'arrête.

$$3x^3 + x^2 + 2x + 1 = (3x^2 - 2x - 4)(x + 1) - 3.$$

$$k(x) = \frac{B(x)x^k}{G(x)} + C(x) \quad G(x) = x^2 + x + 1 \text{ donc } k=2$$

$$MOMI \rightarrow B(x) = x^0 + x^1 + x^2 + 0x^3 + x^4 + x^5 = x^5 + x^4 + x^2 + x^1 + 1$$

$$B(x).x^k = x^7 + x^6 + x^4 + x^3 + x^2$$

$$x^7 + x^6 + x^4 + x^3 + x^2$$

$$-x^7 - x^6 - x^5$$

$$-x^5 + x^4 + x^3 + x^2 \Leftrightarrow x^5 + x^4 + x^3 + x^2 \text{ car } -1 = 1 \text{ en binnaire}$$

$$+ x^5 + x^4 + x^3$$

$$2x^4 + 2x^3 + x^2$$

$2 = 0$ en binnaire

$$-2x^4 - 2x^3 - 2x^2$$

$$-x^2$$

$$\frac{x^2 + x + 1}{x + 1}$$

$$x + 1$$

$$C(x) = x + 1.$$

on envoie MOMIM

exercice 3:

$$1. M = 8 + 2 + 1$$

donc bits qui contrôlent s_M : c_1, c_2 et c_3

$$2. c_1 \text{ contrôle } s_3, s_5, s_7, s_g, s_M. \quad c_1 + s_3 + s_5 + s_7 + s_g + s_M \equiv 0 [2]$$

$$c_2 \text{ contrôle } s_5, s_6, s_7. \quad \text{donc } c_2 \equiv s_3 + s_5 + s_7 + s_g + s_M [2]$$

$$c_4 \equiv s_5 + s_6 + s_7 [2]$$

$$3. c_1 c_2 1 c_4 10 1 c_8 0 M$$

$$c_1 \equiv 1 + 1 + 1 + 0 + 1 [2] = 0$$

$$c_2 \equiv 1 + 0 + 1 + 1 + 1 [2] = 0$$

$$c_4 \equiv 1 + 0 + 1 [2] = 0$$

$$\text{code de } [11010111] = (0, 0, 0, 0).$$

donc 00101010011

5. Supposons que s_7 soit en erreur

$$c_1 \neq s_3 + s_5 + s_7 + s_g + s_M \% 2$$

$$c_2 \neq s_3 + s_6 + s_7 + s_{10} + s_M \% 2$$

$$c_4 \neq s_5 + s_6 + s_7 \% 2.$$

seul s_7 est contrôlé par c_1, c_2 et c_4 , donc on change s_7 pour corriger.

$$6. 11111011100$$

$c_1 c_2 s_3 c_4 s_5 s_6 s_7 c_8 s_g s_{10} s_M$

$$c_1 = 1 + 1 + 1 + 1 + 0 \% 2 = 0$$

$$c_2 = 1 + 0 + 1 + 0 + 0 \% 2 = 0$$

$$c_4 = 1 + 0 + 1 \% 2 = 0$$

$$c_8 = 1 + 0 + 0 \% 2 = 1$$

donc s_7 est faux.

le message est 1100100.

exercice 4:

$$\begin{array}{r}
 101 \mid 1 \\
 10100 \Rightarrow 100 \mid 0 \Rightarrow 101100101010 \\
 \hline
 110
 \end{array}$$

On veut envoier un message de 16 bits.

On veut rajouter $i+j$ bits et on sait que $i \times j = 16$

on prend $i=j=4$

1010	1	
1110	0	
1011	0	
1100	1	
<hr/>		
<u>1100</u>		
		$\Rightarrow 101011101011110010011100$
		message

$$\begin{array}{r} 0 \\ | \\ 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ | \\ 0 \\ \hline 0 \end{array}$$

$$d_{\text{Hamming}} = 3$$

TD 7: les sockets

mode connecté

TCP (fiable)

(analogie avec une communication téléphonique)

mode non connecté

UDP (non fiable)

(analogie avec un envoi par la poste)

exercice 1:

serveur

construction de la ligne par FT

socket()

client

socket()

construction de la ligne

Attribution d'un numéro à votre ligne

bind()

connect()

appelle son correspondant

on branche un téléphone sur la ligne

listen()

on décroche alors que le téléphone sonne

accept()

bind()

on raccroche

close()

close()

en raccroche.

M1

M2

"J'envoie 4"

100 → socket →

GIG-ENDIAN

- le bit de poids fort est à gauche

"J'ai reçu 1"

LITTLE-ENDIAN

- le bit de poids fort est à droite

Quand le serveur fait accept() il crée une nouvelle socket que l'on va utiliser pour l'envoi et la réception de données. Le port 5555 reste libre pour d'éventuelles autres connexions.

int select (size_t mb_desc, ^{plus grand descripteur à scruter}) ⚠ destructeur.

↑
nombre de
sockets ayant
changé d'état

Fd_set * read,
Fd_set * write,
Fd_set * exception,

1 5 7

1	0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---

on veut écouter les
sockets 1, 5 et 7.

ensemble de descripteurs à scruter

struct timeval * timeout); Temps au bout duquel on arrête de

FD_ISSET (&Fd_set, sock); // teste si sock est dans Fd_set

FD_SET (&Fd_set, sock); // met sock dans Fd_set

FD_ZERO (&Fd_set); // initialise Fd_set à zéro.

mb_desc est le plus grand descripteur, ici 7.

while (1) { ^{ensuite de 0 à 5 exclu}

select (5, ^{read} ~~10110111~~, NULL, NULL, NULL); <sup>// bloque jusqu'à qu'un
descripteur dans read change
d'état</sup>

for (i = 0; i < FD_SETSIZE; i++)

if (FD_ISSET (read, i)) {

* accepter les connexions sur la socket i */

}

⚠ pendant un appel à select, read, write et exception sont modifiés
ils contiennent les sockets qui ont changé d'état.

au maximum mb_desc = FD_SETSIZE

Quand on a qu'une socket : select (sock+1, ...);

pour lancer plusieurs clients : ./tcp-client ; ./tcp-client

sock est la socket "principale" qu'on contacte pour toute demande de connexion.

- soit on contacte sock,

→ on accepte la connexion et on ajoute la nouvelle socket à l'ensemble scruté

- soit on contacte une autre socket,

→ on nous a envoyé des données : read_from-client (new-socket);

TP Révisions

TP1 Protocoles, FTP

1) Qu'est ce qu'un protocole ?

- ensemble de règles sur lesquelles se mettent d'accord les parties communicantes pour pouvoir se comprendre.

2) Notion de mode passif ?

pour ne pas monopoliser la socket d'écoute, le serveur ouvre un nouveau port.

TP2 Transmissions

* temps de transmission : temps qui s'écoule entre l'envoi de la 1^e trame et la réception de l'acquittement correspond à la dernière trame.

* temps de propagation : temps qui s'écoule entre l'envoi du message correspond à 1 bit et la réception de l'acquittement correspondant

TP3 Transmission

Nyquist : "Tout va pour le mieux dans le meilleur des mondes"

↳ le débit rendu par l'opérateur

Shannon : conditions réelles

TP4 Couche liaison

Rôle de la couche liaison de données dans le modèle OSI ?

Elle s'assure que la communication se passe bien de bout en bout

se passe bien = éradiquer les pertes, les doublons et les messages erronés
Comment y arriver ?

En rajoutant des informations aux trames

- version avec numérotation des trames émises sur un bit

- version avec numérotation des acquittements et des trames émises

TP5 Contrôle de flux

Fenêtres en émission : ensemble de trames à envoier

Fenêtres coulissantes : on déplace la fenêtre après réception d'un acquittement

contrôle de flux : on ralenti le débit si nécessaire pour ne pas surcharger le récepteur.

TP6 Détection et correction d'erreurs

* Comment y arriver ?

Rajouter de l'information, notamment des bits de contrôle

* Code de Hamming

mot de 7 bits ($s_3, s_5, s_6, s_7, s_9, s_{10}, s_{11}$)

($c_1, c_2, s_3, c_4, s_5, s_6, s_7, c_8, s_9, s_{10}, s_{11}$)

permet de détecter et corriger toute erreur simple.

TP7 des sockets

* TCP : Téléphone

↳ mode connecté

↳ filette

* UDP : Poste

↳ mode non connecté

↳ non filette

Sockets :

↳ manipulées via un descripteur de fichier.

select : "surveiller" un ensemble de descripteurs de fichiers

Fd_set_t ensemble;

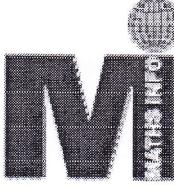
FD_ZERO(&ensemble);

FD_SET (fdin, &ensemble);

select (FD_SETSIZE, &ensemble, NULL, NULL, NULL);

printf ("Yayi"); // sera afficher quand on entrera un caractère au clavier.

Tjacket

 BORDEAUX 1 Sciences Technologies	ANNEE UNIVERSITAIRE 2008/2009 SESSION 1 <p>Diplôme : Master INFO Parcours : Master 1^{ère} année INFO Code de l'UE : INF465 Intitulé de l'UE : Réseaux Responsable : M. Magoni Date : 18/12/2008 Heure : 8h30 – 10h Durée : 1h30 <i>Documents interdits</i></p>	
---	--	---

1. Fournir une description du modèle OSI ainsi que le rôle de chacune de ses couches.
2. Expliquez l'intérêt et le fonctionnement d'un modem. Quelles sont les différences entre un modem standard, un modem câble et un modem ADSL ?
3. Expliquez l'intérêt et le fonctionnement de la méthode d'accès CSMA/CD dans le protocole Ethernet. Expliquez comment fonctionne le protocole *spanning tree* dans la norme IEEE 802.1D.
4. Expliquez l'intérêt et le fonctionnement d'un routeur IP. Dans le protocole RIP, quels sont les mécanismes utilisés pour empêcher la propagation d'informations de routage erronées et en particulier pour éviter la formation de boucles de routage ?
5. Décrire la structure d'une adresse IPv4. A quoi sert un masque ? Quelle est la différence entre une adresse privée et une adresse publique ?
6. Un administrateur possède un réseau IP d'adresse 212.41.80.0/20. Il souhaite créer 6 sous-réseaux IP pouvant contenir chacun un nombre maximal de machines. Donner le nouveau masque à utiliser, les adresses de chaque sous-réseau et les plages d'adresses des machines de chaque sous-réseau.
7. Définir la propriété qui différencie la signature numérique du scellement numérique ? Expliquer pourquoi le scellement ne peut la fournir.
8. Expliquez comment fonctionne le protocole Diffie-Hellman ? Pourquoi est il vulnérable à l'attaque du *man-in-the-middle* ? Comment l'améliorer ?
9. Expliquez l'intérêt et le fonctionnement du protocole IPSec. Décrivez les deux modes utilisables dans IPSec et donnez des exemples de configuration.
10. Expliquez l'intérêt et le principe des numéros de port trouvés dans les en-têtes de la couche transport. Dans le protocole TCP, expliquez comment fonctionnent les acquittements cumulatifs ? Expliquez à quoi servent les acquittements dupliqués ? Expliquez le principe de fonctionnement et l'intérêt d'une fenêtre coulissante de transmission.
11. Expliquez l'intérêt et le fonctionnement du protocole DHCP. Expliquez comment configurer deux serveurs DHCP situés sur deux sous-réseaux différents afin d'assurer une continuité de service dans chacun des sous-réseaux si l'un des deux serveurs tombe en panne ?
12. Expliquez l'intérêt et le fonctionnement du système DNS. Définir ce qu'est une zone d'autorité. Expliquez comment se déroule une requête DNS itérative ?
13. Expliquez l'intérêt et le fonctionnement du protocole NAT. Expliquez comment peuvent fonctionner les protocoles P2P en présence d'un routeur NAT ?
14. Expliquez l'intérêt et le fonctionnement du protocole SMTP. Expliquez comment les e-mails sont ils acheminés vers la destination ?
15. Expliquez l'intérêt et le fonctionnement du protocole HTTP. Qu'est-ce qu'un serveur Web proxy ?