

6 Décembre 2018 (durée : 1h00)

NUMERO D'ETUDIANT(E) (ne pas indiquer votre nom) :

Justifiez et détaillez vos réponses (utiliser si besoin les copies de couleurs).

Exercice 3 : Contrôle des erreurs de transmission (5 points)

Les protocoles de communications mettent en œuvre des algorithmes de contrôle des erreurs binaires de transmission. Nous nous intéressons dans cet exercice, aux 2 algorithmes « bits de parité » et « somme de contrôle », aussi appelé « CHECKSUM » en anglais. On considérera dans l'exercice une parité PAIRE. Le bit de parité pour chaque caractère est ajouté à GAUCHE pour compléter l'octet (bit de poids fort).

On souhaite protéger le message HELLO avant transmission aux moyens de ces 2 algorithmes.

On vous donne les codes ASCII (sur 7 bits et en décimale) des caractères du message à transmettre :
H (72) E(69) L (76) L (76) O (81)

1. dans une première étape, on vous propose de calculer le bit de parité pour chaque caractère du message, et de reporter la valeur décimale du nouveau caractère dans la table ci-dessous.

	Code ASCII du caractère en décimale	Code ASCII avec bit de parité en représentation décimale
H	72	
E	69	
L	76	
L	76	
O	81	

2. dans une seconde étape, on vous demande de calculer la somme de contrôle (en décimale) du message incluant les bits de parité. Sachant que la représentation des caractères est dorénavant sur 8bits. Veuillez détailler vos calculs. **Votre Réponse :**

Valeur du CHECKSUM (en décimale)	
----------------------------------	--

3. La somme de contrôle (ou CheckSum en anglais) permet au protocole IP d'effectuer un **contrôle des erreurs binaires** sur les paquets IP. Quelle(s) parties du paquet IP sont protégées par ce checksum ? Entourer la bonne réponse :

- a) entête IP (header) seul
- b) données (payload) seules
- c) entête IP (header) et données (payload)

Annexe :

• La procédure BACKOFF utilise 3 fonctions :

- random() : tire un nombre réel aléatoire entre 0 et 1.
- int() : rend la partie entière d'un réel
- délai() : calcul le délai d'attente multiple d'un slot_time (51.2 microsec) et est compris entre $[0, 2^k]$.

Avec $k = \min(n, 10)$, n = nbre de ré-émission déjà faites

```
Procédure BACKOFF (no_tentative : entier, VAR maxbackoff : entier)
Const slot_time=51.2 (microsecondes); limite_tentative=16;
Var    délai : entier;

BEGIN
  Si (no_tentative =1)
  Alors maxbackoff =2 (borne de temps d'attente maximale)
  Sinon
    Si (tentative < limite_tentative)
    Alors maxbackoff = maxbackoff*2;
    Sinon maxbackoff = 210 (au delà de 10 essais la borne devient constante)
    fin
  fin
  délai := int(random() *maxbackoff)
  attendre (délai*slot_time)
END
```