M1 Informatique # Protocoles réseaux

Partiel de Protocoles Réseaux

26 octobre 2021

Exercice 1

Un corbeau est capable de faire l'aller-retour entre Sophie Germain (SG) et le campus de Jussieu en 30 minutes (l'aller en 15 minutes, le retour en 15 minutes).

On équipe un corbeau d'une bague contenant une carte micro-SD d'une capacité d'1 TB (octets, pas bits).

(On rappelle qu'un mégaoctet vaut environ 1 000 000 d'octets, un gigaoctet 10^9 octets, un téraoctet environ 10^{12}).

1#

Calculez le temps nécessaire pour transmettre 100MB (octets) de données de SG à Jussieu par corbeau et recevoir un acquittement.

Mo = MB

100Mo : à transmettre
1Mo ≃ 1 000 000 octets = 10^6 octets « un mégaoctet vaut environ 1 000 000 d'octets »
Donc : 100Mo ≃ $100 * 10^6$ octets = $10^2 * 10^6 = 10^{6+2} =$ **10^8 octets**

Capacité de la carte micro-SD : 1 To
1To ≃ **10^{12} octets** « un téraoctet vaut environ 10^{12} »

$10^8 < 10^{12}$: Donc OK !

- $Temp_{SG \to Jussieu}$: $15\ min$
- $Temp_{Jussieu \to SG}(acquittement)$: $15\ min$

$$Total : 30\ min = 1800\ sec$$

2#

Même question pour 100GB.

Go = GB

100Go : à transmettre
1Go $\simeq 10^9$ octets « un gigaoctet 10^9 octets»
Donc : 100Go $\simeq 10^2 * 10^9$ octets = 10^{9+2} = 10^{11} octets

Capacité de la carte micro-SD : 1 To
1To $\simeq 10^{12}$ octets « un téraoctet vaut environ 10^{12} »

$10^{11} < 10^{12}$: Donc OK !

- $Temp_{SG \rightarrow Jussieu}$: $15\ min$
- $Temp_{Jussieu \rightarrow SG}(acquittement)$: $15\ min$

$$Total : 30\ min = 1800\ sec$$

3#

Même question pour 100TB. On suppose que l'on ne dispose que d'un seul corbeau.

To = TB

100To : à transmettre
Capacité de la carte micro-SD : 1 To

Donc : 100 aller-retour ⇒ 100*30 = 3000 minutes = 50 heures

4#

On dispose d'une fibre optique entre SG et Jussieu ayant un débit de 1 Gbit/s (bits, pas octets).

Calculez le temps nécessaire pour transmettre 100Mo, 100Go, 100To. (On négligera la latence du lien).

Transmission : on suppose négligeable la latence, donc :
le temps de transmission = temps d'émission

$$\boxed{\text{Délai d'émission de la trame} = \frac{\text{longueur trame (bits)}}{\text{débit (bit/s)}}}$$

Unités de bits						v · d · m
Système international (SI)			Préfixes binaires (CEI)			Ordre de grandeur
Unité	Notation	Valeur	Unité	Notation	Valeur	
bit	bit	1 bit	bit	bit	1 bit	**1**
kilobit	kbit ou kb	10^3 bits	kibibit	Kibit (ou Kb, par usage)	2^{10} bits	**10^3**
mégabit	Mbit ou Mb	10^6 bits	mébibit	Mibit	2^{20} bits	**10^6**
gigabit	**Gbit** ou Gb	10^9 bits	gibibit	Gibit	2^{30} bits	**10^9**
térabit	Tbit ou Tb	10^{12} bits	tébibit	Tibit	2^{40} bits	**10^{12}**
pétabit	Pbit	10^{15} bits	pébibit	Pibit	2^{50} bits	**10^{15}**
exabit	Ebit	10^{18} bits	exbibit	Eibit	2^{60} bits	**10^{18}**
zettabit	Zbit	10^{21} bits	zébibit	Zibit	2^{70} bits	**10^{21}**
yottabit	Ybit	10^{24} bits	yobibit	Yibit	2^{80} bits	**10^{24}**

Source : https://fr.wikipedia.org/wiki/Gigabit

Débit : 1 Gbit /s = 10^9 bits = 1 000 000 000

100Mo ≃ 10^8 octets = 8*10^8 bits = 800 000 000 bits

$$t_e = \frac{\text{longueur trame (bits)}}{\text{débit (bit/s)}} = \frac{8\cancel{00\ 000\ 000}}{1\ 0\cancel{00\ 000\ 000}} = \frac{8}{10} = 0{,}8\ sec$$

100Go ≃ 10^{11} octets = 8*10^{11} bits = 8*10^2*10^9 bits = 800*10^9 bits

$$t_e = \frac{\text{longueur trame (bits)}}{\text{débit (bit/s)}} = \frac{800 \cdot \cancel{10^9}}{\cancel{10^9}} = 800\ sec$$

1To ≃ 10^{12} octets « un téraoctet vaut environ 10^{12} »

100To ≃ 100*10^{12} octets = 8*100*10^{12} bits = 800*10^{12} bits

$$t_e = \frac{\text{longueur trame (bits)}}{\text{débit (bit/s)}} = \frac{800 \cdot 10^{12}}{10^9} = 800 \cdot 10^{12-9}$$

$$= 800 \cdot 10^3 = 800 \cdot 1000\ sec = \frac{8000\cancel{00}\ sec}{36\cancel{00}\ sec} = \frac{8000}{36}$$

$$\approx 222\ heures$$

5#

Qui a un meilleur débit, la fibre optique ou le corbeau ?

Pour les données de taille « raisonnable », la fibre est plus rapide que le corbeau, en revanche, pour une taille de données importante, le corbeau est plus rapide.

6#

Pourquoi n'utilise-t-on pas davantage le transfert par corbeau ? (Donnez votre réponse en une ou deux phrases.)

Le transfert par corbeau semble moins fiable que la fibre. De plus, il est relativement rare d'avoir des fichiers aussi gros à transmettre, et pour les petits fichiers, la fibre est plus rapide.

Exercice 2 - Code XOR

On considère un message de k octets $a_1 \dots a_k$.

On considère le code suivant :

On envoie $k+1$ octets : les k premiers octets $a_1 \dots a_k$ puis un octet

$$a_{k+1} = a_1 \oplus a_2 \oplus \dots \oplus a_k$$

ou l'opérateur $\oplus$ représente le XOR (ou exclusif) bit-à-bit.

Par exemple, en binaire

$$00010110 \oplus 11001100 = 11011010$$

Ou exclusif

Etant donné deux mots de code pouvant être émis ou reçus, par exemple 10001001 et 10110001, il est possible de déterminer de combien de bits ils diffèrent. Dans notre exemple, ils diffèrent de 3 bits.

Pour évaluer cette différence, il suffit d'effectuer un OU exclusif entre les deux mots de code et de compter le nombre de 1 du résultat.

Par exemple,

10001001
<u>10110001</u>
00111000

Le nombre de bits de différence entre deux mots de code est appelé **distance de Hamming**.

Source : Andrew Tanenbaum, David Wetherall. Computer Networks.. (Edition française)

XOR est l'opération $\boldsymbol{ou\ exclusif}$, notée « ^ » en C ou ⊕ en Mathématiques. C'est une opération classique sur les bits :

- $0 \oplus 0 = 0$
- $0 \oplus 1 = 1$
- $1 \oplus 0 = 1$
- $1 \oplus 1 = 0$

Remarquez les propriétés suivantes :

- $a \oplus b = 0$
- $a \oplus b \oplus b = a$

Comme l'application du $\boldsymbol{ou\ exclusif}$ deux fois avec la même valeur redonne la valeur initiale, le chiffrement et le déchiffrement utilisent exactement le même programme :

- $M \oplus K = C$
- $C \oplus K = M$

Source : Cryptographie appliquée : protocoles, algorithmes et codes source en C / Bruce Schneier

1#

Montrez que ce code permet de détecter une erreur sur un bit

Soient a et b des bits.
D'après la table de vérité du XOR, dans a ⊕b, si uniquement a est flippé ou uniquement b, le résultat différent. L'unique erreur va donc être toujours présente après chaque XOR, y compris le a_{k+1} recalculé. Si l'erreur est dans a_{k+1} , le a_{k+1} calculé par la destinataire différent de celui reçu.

2#

Montrez que ce code ne permet pas de détecter deux erreurs (sur deux bits différents, n'importe où).

Si le bit numéro 3 de deux octets de données est flippé, le résultat final a_{k+1} aura son bit 3 flippé deux fois, ce que est le même résultat que sans erreur.

3#

Combien d'erreurs (sur un bit) ce code permet-il de corriger ?

S'il y a une seule erreur, on peut savoir sur quel bit, mais pas dans quel octet elle est. Ce code ne détecte pas plus d'une erreur. Ce code corrige donc 0 erreur.

4#

On introduit un nouveau modèle d'erreurs : *un octet corrompu*.

S'il y a une erreur sur un ou plusieurs bits de l'octet numéro i (avec $0 \leq i \leq k$), qui est alors dit *corrompu*, le récepteur est prévenu.

Dans ce cas, il reçoit seulement les k octets $a_1 \dots a_{i-1}, a_{i+1} \dots a_{k+1}$ (qui ne sont pas corrompus), et le numéro i de l'octet corrompu.

Montrez que ce code permet de corriger un octet corrompu (donnez l'algorithme de correction).

On va appliquer le XOR octet par octet au lieu de bit par bit. L'octet $\boldsymbol{a}_{k+1}$ stokes le résultat du XOR. Comme le récepteur reçoit le numéro i de l'octet corrompu et tous les autres octets, dont $\boldsymbol{a}_{k+1}$, il peut (en effectuent l'opération XOR) retrouver la valeur du i-ème octet corrompu. Il corrige donc l'erreur.

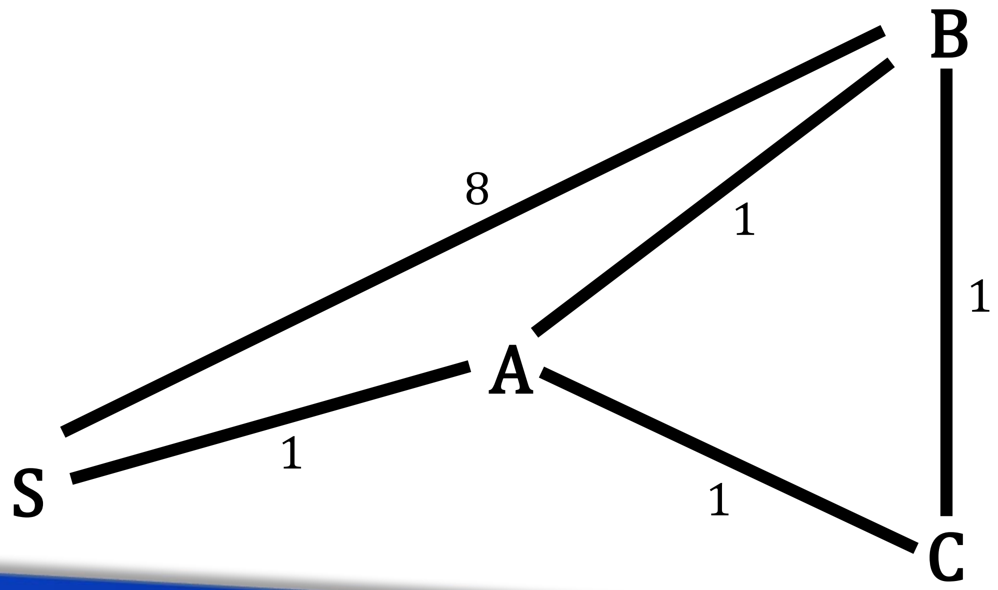
On remarque que $\boldsymbol{a} \oplus \boldsymbol{b} = \boldsymbol{c} \Leftrightarrow \boldsymbol{a} \oplus \boldsymbol{c} = \boldsymbol{b} \Leftrightarrow \boldsymbol{b} \oplus \boldsymbol{c} = \boldsymbol{a}$.
On a que la somme XOR des a_1 à a_k donne a_{k+1}, donc le XOR des a_1 à a_{i-1} avec les a_{i+1} à a_{k+1} donne a_i . Ce code corrige donc l'octet corrompu a_i .

Algorithme

- Réception des octets des $\boldsymbol{k}$ octets $\mathbf{a_1} \dots \mathbf{a}_{i-1}, \mathbf{a}_{i+1} \dots \mathbf{a}_{k+1}$
(qui ne sont pas corrompus), et le numéro $\boldsymbol{i}$ de l'octet corrompu.
- $\boldsymbol{b} = \boldsymbol{a_1} \oplus \dots \oplus \boldsymbol{a_{i-1}} \oplus \boldsymbol{a_{i+1}} \oplus \dots \oplus \boldsymbol{a_k}$
- $\boldsymbol{a_i} = \boldsymbol{b} \oplus \boldsymbol{a_{k+1}}$
- Réinsérer a_i à sa place, le message est corrigé.

Exercice 3

Alice est administrateur réseau. Le réseau qu'elle administre a la topologie suivante, ou les entiers sur les arrêtes sont les couts des liens :



1#

En fixant la destination à S,
faites évoluer le protocole à vecteur de distances naïf depuis l'état initial jusqu'à convergence
(vous pouvez, si vous le désire, supposer que l'implémentation maintient une table de routage redondante).

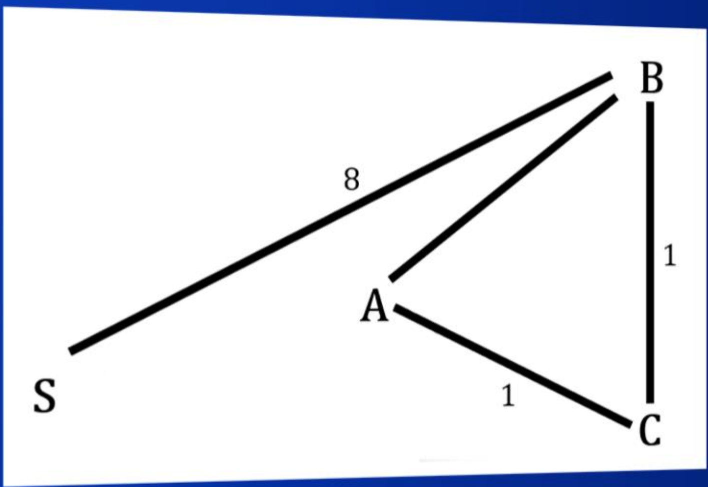
Vous n'avez pas à justifier votre réponse – il suffit de me fournir un tableau ayant la forme suivante :

S	d = ?, nh = ?	d = ?, nh = ?	...
A	d = ?, nh = ?	d = ?, nh = ?	...
B	d = ?, nh = ?	d = ?, nh = ?	...
C	d = ?, nh = ?	d = ?, nh = ?	...

Temp	0	1	2
Qui envoie ?	-	S	A
S	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$
A	$d(s) = \infty$, $nh(s) = \perp$	$d(s) = 1$, $nh(s) = s$	$d(s) = 1$, $nh(s) = s$
B	$d(s) = \infty$, $nh(s) = \perp$	$d(s) = 8$, $nh(s) = s$	$d(s) = 2$, $nh(s) = A$
C	$d(s) = \infty$, $nh(s) = \perp$	$d(s) = \infty$, $nh(s) = \perp$	$d(s) = 2$, $nh(s) = A$

2#

Bertrand, le patron d'Alice, a besoin d'un câble Ethernet, il déconnecte le lien direct entre S et A :



On suppose que l'algorithme avait déjà convergé dans la topologie précédente lorsque Bernard déconnecte le lien S-A. En fixant toujours la source à S, faites évoluer l'algorithme depuis l'état de la topologie précédente jusqu'à la nouvelle convergence.

(On suppose que les routeurs n'implémentent pas l'algorithme de l'horizon scindé).

Qui annonce ?			B	A	B
S	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$
A	$d(s) = 1$, $nh(s) = s$	$d(s) = \infty$, $nh(s) = \perp$	$d(s) = 3$, $nh(s) = B$	$d(s) = 3$, $nh(s) = B$	$d(s) = 5$, $nh(s) = B$
B	$d(s) = 2$, $nh(s) = A$	$d(s) = 2$, $nh(s) = A$	$d(s) = 2$, $nh(s) = A$	$d(s) = 4$, $nh(s) = A$	$d(s) = 4$, $nh(s) = A$
C	$d(s) = 2$, $nh(s) = A$	$d(s) = 2$, $nh(s) = A$	$d(s) = 2$, $nh(s) = A$	$d(s) = 4$, $nh(s) = A$	$d(s) = 4$, $nh(s) = A$

Qui annonce ?	A	B	A	B	A
S	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$
A	$d(s) = 5$, $nh(s) = B$	$d(s) = 7$, $nh(s) = B$	$d(s) = 7$, $nh(s) = B$	$d(s) = 9$, $nh(s) = B$	$d(s) = 9$, $nh(s) = B$
B	$d(s) = 6$, $nh(s) = A$	$d(s) = 6$, $nh(s) = A$	$d(s) = 8$, $nh(s) = A$	$d(s) = 8$, $nh(s) = A$	$d(s) = 10$, $nh(s) = A$
C	$d(s) = 6$, $nh(s) = A$	$d(s) = 6$, $nh(s) = A$	$d(s) = 8$, $nh(s) = A$	$d(s) = 8$, $nh(s) = A$	$d(s) = 10$, $nh(s) = A$

Qui annonce ?	S	B
S	$d(s) = 0$, $nh(s) = S$	$d(s) = 0$, $nh(s) = S$
A	$d(s) = 9$, $nh(s) = B$	$d(s) = 9$, $nh(s) = B$
B	$d(s) = 8$, $nh(s) = S$	$d(s) = 8$, $nh(s) = S$
C	$d(s) = 10$, $nh(s) = A$	$d(s) = 9$, $nh(s) = B$

(L'algorithme est no-déterministe, il est donc possible que C ne fasse jamais d'annonce).