

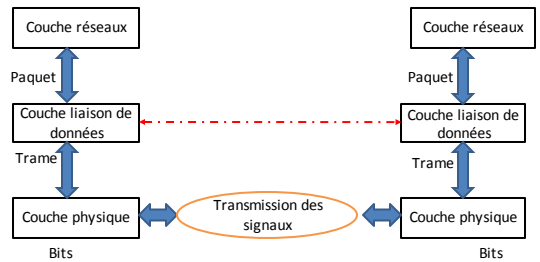
Chapitre 3

La couche liaison de données

1

1.1 .Définition

- La couche liaison de donnée est un ensemble des matériels et logiciels permettant d'assurer une transmission fiable des données sur la liaison physique
- L'unité d'information associée à la couche2 du modèle OSI est la trame



2

1.2.Rôle de la couche liaison de données

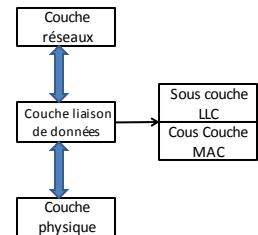
- Découpage en trame : délimiter les données issus de la couche réseaux
- Contrôle d'accès au media de transmission : quelle machine a le droit d'utiliser le support pour envoyer les données
- Adressage : identification physique des machines
- Contrôle d'erreurs : assurer le transfert sans erreurs des données
- Contrôle de flux : assurer un transfert fiable de données

3

1.3. Les deux sous couche de la couche liaison de données

La couche liaison de données est découpée en deux sous couches :

- La sous couche MAC (Media Access Control)
- la sous couche LLC (Logical Link Control)



4

1.3.1 Sous-couche MAC et LLC

- La sous couche MAC l'interface avec la couche physique a pour rôle:
 - gérer l'accès au support physique .
 - structurer les bits d'information en trames
 - gérer les adresses physiques des interfaces de communication
- La sous-couche LLC l'interface avec la couche Réseau a pour rôle:
 - La protection contre les erreurs de transmission .
 - Assurer le transfert des trames et le contrôle de flux entre les stations du réseau.

5

2. Sous couche MAC

Contrôle d'accès

6

2.1 Notion de trame

- Une trame est une suite délimitée de bits .
- La taille d'une trame peut être variable.
- Elle contient plusieurs champs :
 - Délimiteur du début et la fin de la trame : ensemble de bits utilisés pour détecter le début et la fin d'une trame
 - Entête de la trame : ensemble d'informations rajoutées par la couche liaison de données aux données issues de la couche réseaux .
 - Les données : les données reçus de la couche réseaux .

Délimiteur du début de la trame	Entête de la trame	Données	Délimiteur du fin de la trame
---------------------------------	--------------------	---------	-------------------------------

Remarque : la sous couche MAC encapsule les données en prévenance de la couche réseaux

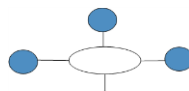
7

2.2 Mode de connexion

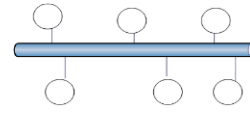
- Connexion point-à-point : deux stations seulement partagent le support de communication



- Connexion multi-points : plusieurs stations utilisent le même support de transmission



TOPOLOGIE EN ANNEAU



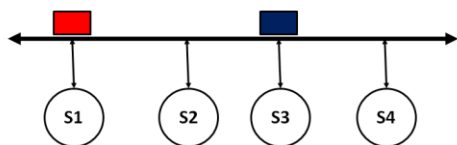
TOPOLOGIE EN BUS

- Dans une liaison multipoint plusieurs stations utilisent le même support de transmission .???Problème de ...

8

2.3 Notion de collision

- Lorsque deux stations (ou plus) envoient des données (trames) simultanément sur le même support on dit qu'il y a collision ou interférence.
- Physiquement : une collision signifie que les signaux qui représentent les données sont mélangés et on arrive pas à reconnaître l'information
- Les données doivent être retransmis ultérieurement



9

2.4 Problème ?

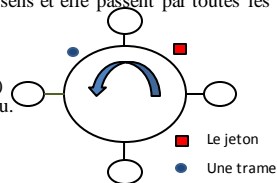
- Comment partager le même support entre les différentes stations ?
- La sous couche MAC (Media Access Control) a pour rôle de :
 - définir des protocoles qui déterminent les stations autorisés à transmettre des données sur le média partagé.
 - régler les problèmes des collisions .
- Il existe deux grandes catégories de protocoles MAC :
 - les protocoles déterministes : jeton
 - Les protocoles aléatoires :
 - Sans écoute de la porteuse :
 - Avec écoute de la porteuse :

10

2.4.1 Les protocoles MAC déterministes : le jeton

- Ce protocole est utilisé dans une topologie en bus ou en anneau.
- Les stations sont organisés de manière à former un anneau.
- Chaque station est reliée à la suivante et à la précédente (liaisons point à point).
- Les trames circulent dans un seul sens et elle passent par toutes les stations (autour de l'anneau).

• Un jeton est une donnée (trame) spécial qui circule autour de l'anneau.



11

Principe du jeton

- Lorsqu'une machine désire transmettre :
 - Elle prend le jeton,
 - transmet les données (trames) pendant un temps limité.
 - remet le jeton dans l'anneau.
- Une machine ne peut pas détenir le jeton plus qu'une durée de temps limité . Au delà elle doit le libérer pour permettre au autre station d'émettre .
- Cette méthode elle à un avantage et un inconvénient :
 - Pas de collisions .
 - problème perte de jeton .

12

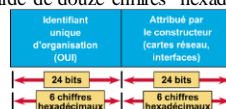
2.4.2 Protocoles MAC aléatoires

- Dans ces protocoles, chaque station peut émettre à n'importe quel moment.
- S'il y a une collision alors retransmission de la trame par la suite (ultérieurement).
- Il existe deux principales méthodes :
 - Sans écoute de la porteuse :
 - Avec écoute de la porteuse :

13

Adressage des machines : Adressage physique

- Chaque ordinateur a une façon unique de s'identifier (soit relié à un réseau ou non, possède une adresse physique).
- Il n'y a jamais deux adresses physiques identiques.
- L'adresse physique, appelée adresse MAC (Media Access Control), se trouve sur la carte réseau.
- Les adresses MAC comportent 6 octets (48 bits) et sont exprimées à l'aide de douze chiffres hexadécimaux :



14

3. Sous couche LLC

Contrôle d'erreurs
Contrôle de flux

15

1. Erreurs de transmission

- Des erreurs sont dues aux canaux de transmission.
- **Un ou plusieurs bits** d'information peuvent être **changés** en même temps durant la transmission des données.
- Les données peuvent être perdues.
- Les erreurs sont causées par
 - L'interférence (Bruit)
 - La distorsion

16

2. Comment prévenir les erreurs

- Pour réduire les interférences
 - Blindage des fils
 - S'assurer que les câbles sont loin des sources d'interférence (Bruit).
- Pour réduire la distorsion
 - Ajuster l'équipement de transmission et améliorer la qualité de la connexion
 - Utiliser des amplificateurs et des répéteurs
 - Utiliser du câble de meilleure qualité
- Mais **le risque d'erreurs existe toujours**, mais il doit pas dépasser un certain seuil 10^{-8} à 10^{-10}

17

3. Contrôle d'erreurs

- Il faut pouvoir détecter et corriger les erreurs.
- De façon générale pour transmettre k bits d'information, on ajoute **r bits dit bits de contrôle**.
On parle de code (n, k) ou de mot de code.
- Les bits de contrôle sont **calculés en fonction** des bits de l'information
Au total on transmet $n = k + r$ bits
- **À la réception** les bits de contrôles **seront recalculés** afin de s'assurer si l'information est bien reçue ou non

18

4. Technique de contrôle d'erreurs

- Il existe plusieurs méthodes de contrôle d'erreurs :
- **Détection (code détecteur) :**
Ce type de code permet de détecter le changement de un ou plusieurs bits d'information. Mais il n'a pas la possibilité de corriger ces erreurs.
- **Détection et correction (code correcteur) :**
Ce type de code permet de détecter le changement de un ou plusieurs bits d'information. En plus il possède la capacité de corriger ces erreurs.

19

4.1 Détection d'erreurs

Les techniques les plus utilisées pour la détection sont :

- VRC (Vertical Redundancy Check) : Parité vertical
- LRC (Longitudinal Redundancy Check) : parité longitudinale
- CRC (Cyclic Redundancy Check) : Vérification polynomiale

20

4.1.1 Parité : VRC(Vertical Redundancy Check)

- Le plus vieux mécanisme
- Calculer la parité est rajouter un bit à l'information envoyé
 - Parité **paire** : si le nombre de 1 dans l'information est **paire** alors le bit de parité est égale à 1, sinon 0
 - Parité **impaire** : si le nombre de 1 dans l'information est **impaire** alors le bit de parité est égale à 1, sinon 0
- **Exemple :**
si on utilise une parité impaire pour l'information 1100100 on rajoute 1.
l'information a envoyé 1100100**1**

21

Détection des erreurs grâce au code VRC

- La détection d'erreur avec le VRC consiste a :
 - **recalculer** le bit de parité a la réception et **le comparer** avec le bit de parité reçu

Exemple : l'information reçues 1100100 **1**

le bit de parité qui correspond à l'information : 1100100 est égal à 1 donc l'information reçues est correcte

- **Capacité de détection :** si un seule bit change alors la parité change → on détecte l'erreur mais on peut pas savoir quel est le bit qui a changé.
- Si le nombre de de bits qui change est pair → impossible de détecter l'erreur
- 1100100 → 11**1**0000

22

4.1.2. Parité longitudinale : LRC

- Appliquer le principe de la parité (paire ou impaire)aux colonnes d'un bloc de données
- Exemple le message DATA:

Caractère	Code ASCII
D	1000100
A	1000001
T	1010100
A	1000001
LRC	1101111

23

- C'est un code meilleur que le VRC
- Impossible de détecter l'erreur si deux bit sont changé en même temps sur la même colonnes
- Pour plus d'efficacité Rajouter un contrôle sur les lignes

Caractère	ASCII	Bit de parité
D	1000100	1
A	1000001	1
T	1010100	0
A	1000001	1
LRC	1101111	1

24

4.1.3. Vérification polynomiale

- Rappel : Une information en binaire peut être écrite sous la forme polynomiale suivant les puissances de 2

$$(1110)_2 = 1 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$$

Dans le cas général :

$$(u_k u_{k-1} \dots u_1 u_0)_2 = u_k \cdot X^k + u_{k-1} \cdot X^{k-1} + \dots + u_1 \cdot X^1 + u_0 \cdot X^0 \text{ avec } u_i \in \{0,1\}$$

Exemple :

La suite 1100101 est représentée par le polynôme

$$1100101 = 1 \cdot X^6 + 1 \cdot X^5 + 0 \cdot X^4 + 0 \cdot X^3 + 1 \cdot X^2 + 0 \cdot X^1 + 1 \cdot X^0 \\ = X^6 + X^5 + X^2 + 1$$

25

Calcul du CRC par le polynôme générateur

- On choisit un polynôme appelé polynôme générateur $G(X)$ de degré n
- Exemple : polynôme générateur $X^4 + X^2 + X$ de degré 4
- Soit une information sur m bits représentée sous la forme d'un polynôme $M(X)$ de degré m
- Pour calculer le CRC :
 - multiplier le polynôme $M(X)$ par X^n (n est le degré du polynôme générateur)
 - effectuer une division de $X^n \cdot M(X)$ par $G(X)$,
 - on obtient le Quotient $Q(X)$ et le reste $R(X)$

$$X^n M(X) = Q(X) \cdot G(X) + R(X)$$
 - Le CRC correspond au reste de la division $R(X)$
- Donc l'information envoyée est égale à : $M(X) \cdot R(X)$

26

Exemple

Soit l'information 11100111 et le polynôme générateur $X^4 + X^2 + X$

$$G(X) = X^4 + X^2 + X$$

$$M(X) = X^7 + X^6 + X^5 + X^2 + X^1 + 1$$

Multiplier $M(X)$ par X^4

$$X^4 \cdot M(X) = X^{11} + X^{10} + X^9 + X^6 + X^5 + X^4 \\ = 111001110000$$

27

1110 0 1 11 0000
10110

010101
10110

111
00000

1111
00000

11110
10110

010000
10110

11000
10110

01110

10110

11001101

L'information à envoyer :
11100111 **1110**

Remarque :
Toutes les opérations se font en binaire modulo 2
1+1 = 0 et 0+0 = 0
1+0 = 1 et 0+1 = 1
Effectuer des ou exclusifs

28

Détection des erreurs

- A la réception de l'information, **diviser le polynôme qui représente** l'information sur le polynôme générateur
- Si le reste de la division **est nul** alors l'information est correcte
- Si non il y a une erreur

29

11100 1111110
10110

010101
10110

111
00000

1111
00000

11111
10110

010011
10110

10110
10110

00000

10110

11001101

L'information est correcte

30

Calcul du CRC par des additions successives

- Soit $G(X)$ un polynôme générateur de degré n . On le transforme en un mot binaire.
- **Exemple :**
Avec le polynôme générateur X^4+X^2+X , on obtient 10110.
- On ajoute n zéros au mot binaire à transmettre où n est le degré du polynôme générateur.
- **Exemple :**
On souhaite transmettre le mot 11100111 en utilisant le polynôme générateur X^4+X^2+X , on obtient alors 111001110000.
- On va **additionner itérativement** à ce mot, le mot correspondant au polynôme générateur jusqu'à ce que le mot obtenu soit inférieur au polynôme générateur.
- Ce mot obtenu correspond au CRC à ajouter au mot avant de l'émettre.

31

```

111001110000
10110
-----
010101110000
10110
-----
000011110000
10110
-----
000001000000
10110
-----
000000011000
10110
-----
000000001110

```

Le code CRC= 1110 donc l'information à transmettre
1110011 **1110**

32

A la réception, refaire la même opération.
Si le résultat est nul alors l'information est correcte.

```

111001111110
10110
-----
010101111110
10110
-----
000011111110
10110
-----
000001001110
10110
-----
000000010110
10110
-----
000000000000

```

Le reste de la division est nul donc l'information est correcte

33

Correction d'erreur par retransmission

- La méthode la plus simple pour corriger une erreur c'est de demander une retransmission
- Un récepteur qui détecte une erreur demande une retransmission jusqu'à ce qu'il n'y est plus d'erreur.

34

4.2. Codes correcteurs

- Le principe des codes correcteurs est le même que celui des codes détecteurs.
- Lors de l'émission rajouter des bits de contrôle supplémentaires.
- A la réception, détecter les erreurs grâce au bits de contrôle et possibilité de corriger ces erreurs.
- Le code de hamming est parmi ces codes. Il permet de détecter une seule erreur.

35

Code de hamming

- Un code de hamming est un code $C(n,k)$ avec $n=2^r-1$ et $r=n-k$.
- Exemple :
 - $n=7 \rightarrow k=4 \rightarrow r=3$
 - $n=15 \rightarrow k=11 \rightarrow r=4 \dots\dots$
- Le codage de hamming se base sur le calcul de la parité.
- Au lieu de rajouter un seul bit, rajouter plusieurs bits de parité.
- chaque bit de contrôle est une fonction de plusieurs bits d'information.

36

Principe du code de hamming

- Soit un mot de code (7,4) → rajouter 3 bits de contrôles notés: C_0, C_1, C_2 .
- et l'information de départ est sur 4 bits $m=U_0U_1U_2U_3$.
- Les bits de contrôle sont insérés dans les bits de l'information de la façon suivante :
 - il prennent les position 2^i { 1,2,4,8,...}.
 - et les bits d'information prennent les autres positions.

C0	C1	U0	C3	U1	U2	U3
2^0	2^1	3	2^3	5	6	7
1	2		4			

37

Comment calculer les bits de contrôle

- Chaque bits de l'information possède une position dans le mot de code final écrite en puissance de 2.

Exemple :

U_0 est dans la position $3=1+2 = 2^0+2^1$

U_1 est dans la position $5=1+4 = 2^0+2^2$

U_2 est dans la position $6=2+4 = 2^1+2^2$

U_3 est dans la position $7=1+2 +4= 2^0+2^1+2^2$

Un bit de l'information ayant la position **J** participe au calcul du bit de contrôle ayant la position 2^i si 2^i **existe dans la décomposition de J en puissance de 2**

38

Comment calculer les bits de contrôle

Dans l'exemple précédant :

- La position de C_0 est 2^0 → donc les bits U_0, U_1, U_3 participe dans le calcul de C_0 → $C_0=U_0+U_1+U_3$
- La position de C_1 est 2^1 → donc les bits U_0, U_2, U_3 participe dans le calcul de C_1 → $C_1=U_0+U_2+U_3$
- La position de C_2 est 2^2 → donc les bits U_1, U_2, U_3 participe dans le calcul de C_2 → $C_2=U_1+U_2+U_3$
- **Application :**
- pour l'information 1010 trouver le code de hamming correspond.
 - $C_0=U_0+U_1+U_3 = 1+0+0 = 1$
 - $C_1=U_0+U_2+U_3 = 1+1+0 = 0$
 - $C_2=U_1+U_2+U_3 = 0+1+0 = 1$
- Donc l'information a envoyée : 1011010

39

Détection de l'erreur

- À la réception du message , **recalculer** les bits de contrôle de la même **manière que lors de l'émission** .
- Si égalité alors passage au bit suivant.
- Si non incrémenter un **compteur C** par la position du bit de contrôle.
- Après avoir recalculer tous les bits de contrôle :
 - Si le compteur est **égale a zéro** alors pas d'erreur
 - Sinon il indique le **numéro du bit erroné**

40

Exemple

- Si on envoi le message 1011010 et on reçoit le message 1011000 normalement il y a une erreur.
- Pour la détecter :
- $C'_0=U_0+U_1+U_3$ → $C'_0=1+0+0 = 1$ correcte , $C=0$
- $C'_1=U_0+U_2+U_3$ → $C'_1=1+0+0 = 1$, erreur $C=2$
- $C'_2=U_1+U_2+U_3$ → $C'_2=0+0+0 = 0$, erreur $C=2+4$
- Donc le bit erroné est le bit N° 6

41