

Chapitre 3

Couche Liaison : Principes

05/12/03

Couche liaison

Page 1

Description

- La couche liaison récupère des paquets de la couche réseau.
- Pour chaque paquet, elle construit une (ou plusieurs) trame(s).
- La couche liaison envoie chaque trame à la couche physique.

05/12/03

Couche liaison

Page 5

Plan

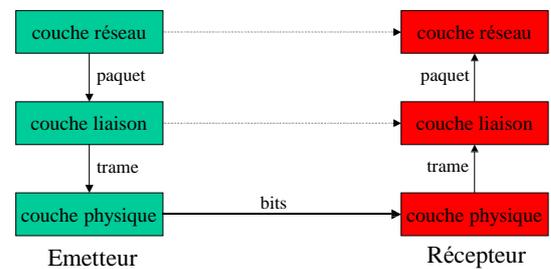
- Introduction
- Délimitation de trames
- Détection/Correction d'erreurs
- Contrôle de flux

05/12/03

Couche liaison

Page 2

Description



05/12/03

Couche liaison

Page 6

Introduction

05/12/03

Couche liaison

Page 3

Services offerts

- Gestion (délimitation) de trames
- Contrôle d'erreurs
- Contrôle de flux
- Contrôle d'accès à un canal partagé (MAC)

05/12/03

Couche liaison

Page 7

Couche liaison

- Cette couche doit assurer une transmission exempte d'erreurs sur un canal de communication.
- Les données sont fractionnées en trames.

05/12/03

Couche liaison

Page 4

Délimitation de trames

05/12/03

Couche liaison

Page 8

Délimitation des trames

- Il existe trois méthodes :
 - Compter les caractères
 - Utiliser des champs délimiteurs de trame
 - Ils se situent en début et en fin de trame
 - Des bits (ou caractères) de transparence sont nécessaires
 - Violer le codage normalement utilisé dans la couche physique

05/12/03

Couche liaison

Page 9

Exemple

- Fanion : 01111110
- Bit de transparence : 0 inséré après toute séquence de cinq 1 successifs dans la trame.
- Technique utilisée dans :
 - HDLC
 - PPP

05/12/03

Couche liaison

Page 13

Compter les caractères

- On utilise un champ dans l'en-tête de la trame pour indiquer le nombre de caractères de la trame
- Problème : si la valeur du champ est modifiée au cours de la transmission
- Méthode rarement utilisée seule

05/12/03

Couche liaison

Page 10

Exemple

Données :

01011001111110

Trame :

01111110 010110011111010 01111110

05/12/03

Couche liaison

Page 14

Exemple

Trames émises

06 'S' 'U' 'P' 'E' 'R' 03 'L' 'E' 06 'C' 'O' 'U' 'R' 'S'

Trames reçues

06 'S' 'U' 'P' 'E' 'R' 04 'L' 'E' 06 43 'O' 'U' 'R' 'S'

code ASCII de 'C'

05/12/03

Couche liaison

Page 11

Utiliser des fanions

- Avantages
 - permet toujours de retrouver la synchronisation
 - permet l'envoi de trames de tailles quelconques
 - technique la plus simple
- Cette technique est utilisée également en considérant des caractères de délimitation et des caractères de transparence.

05/12/03

Couche liaison

Page 15

Utiliser des délimiteurs

- Un fanion (délimiteur) est placé :
 - au début de chaque trame
 - à la fin de chaque trame (en fait, au début de la suivante)
- Un fanion (flag) = séquence particulière de bits
- Des bits de transparence sont alors nécessaires pour qu'une séquence binaire dans la trame ne corresponde accidentellement au fanion.

05/12/03

Couche liaison

Page 12

Violer le codage

- Utilisable lorsque le codage sur le support physique contient des redondances
- Par exemple :
 - 0 = impulsion positive puis négative
 - 1 = impulsion négative puis positive
 - On peut donc utiliser les combinaisons positive-positive et négative-négative pour délimiter les trames
- Utilisée dans la norme 802

05/12/03

Couche liaison

Page 16

Détection/Correction d'erreurs

05/12/03

Couche liaison

Page 17

Types d'erreurs de transmission

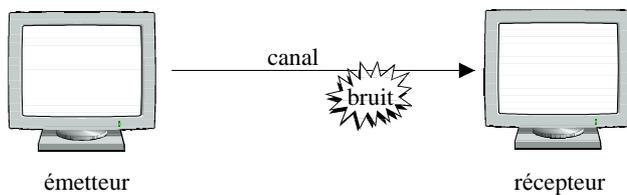
- Erreurs isolées
 - simples à détecter
 - simples à corriger
 - proportion élevée de blocs affectés
- Erreurs en rafales
 - difficiles à détecter
 - difficiles à corriger
 - proportion faible de blocs affectés

05/12/03

Couche liaison

Page 21

Transmission d'information



05/12/03

Couche liaison

Page 18

Exemple

- taille des blocs échangés : 1000 bits
- taux d'erreur : 1/1000
- erreurs isolées
 - la plupart des blocs en erreur
- erreurs en rafale (de longueur 100)
 - un bloc sur 100 en erreur en moyenne

05/12/03

Couche liaison

Page 22

Causes d'erreurs sur un canal

- rayonnement électromagnétique
 - relais
 - émetteurs
- câblage mal isolé
- effet de distorsion

05/12/03

Couche liaison

Page 19

Deux stratégies possibles

La destination peut :

- détecter les erreurs, puis demander une retransmission
 - code détecteurs d'erreurs
- détecter et corriger les erreurs
 - codes correcteurs d'erreur

05/12/03

Couche liaison

Page 23

Taux d'erreur sur un canal

$$\text{taux d'erreur} = \frac{\text{nombre de bits erronés}}{\text{nombre de bits émis}}$$

- 10^{-9} pour les réseaux locaux
- 10^{-5} pour le RTC
- taux élevé pour le téléphone sans fil

05/12/03

Couche liaison

Page 20

Principe des codes

- Exploiter la redondance d'informations
 - ⇒ ajouter des bits de contrôle aux bits de données
- Corriger est plus difficile que détecter
 - ⇒ plus de bits de contrôle

05/12/03

Couche liaison

Page 24

Supr matie des codes d tecteurs

Les codes d tecteurs sont plus souvent utilis s que les codes correcteurs mais :

- lorsque le canal est unidirectionnel
- lorsque la distance est  lev e (satellite)
- lorsque les erreurs sont isol es

les codes correcteurs peuvent (ou doivent)  tre utilis s.

Capacit  d'un code

- Pour qu'un code soit capable de d tecter d erreurs, sa distance de Hamming doit  tre  gal   d+1.
- Pour qu'un code soit capable de corriger d erreurs, sa distance de Hamming doit  tre  gal   2*d+1.

Mot de code

d bits de donn es

+

c bits de contr le

=

n bits d'information (  transmettre)

Un tel mot de n bits est appel  un mot de code

Exercice

Soit le code $C = \{ 0000\ 0000, 0000\ 1111, 1111\ 0000, 1111\ 1111 \}$

- quel est sa distance ?
- si un r cepteur re oit 0000 0111, que peut-il en conclure ?
- est-il certain de pouvoir reconna tre le mot original ?

Distance de Hamming de 2 mots

Il s'agit du nombre de bits qui diff rent entre deux mots de code x et y.

$$\begin{array}{r} 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1 \\ \oplus 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1 \\ \hline = 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0 \end{array}$$

$$\text{Dist}(x,y) = 3$$

Diff rents codes

- Code de contr le de parit 
- Code de Hamming
- Codes polynomiaux
- ...

Distance de Hamming d'un code

Soit C un code (d tecteur et/ou correcteur)

$$\text{Dist}(C) = \min \{ \text{Dist}(x,y) \mid x \in C \wedge y \in C \}$$

Il s'agit de la distance minimale entre 2 mots du code

Code de contr le de parit 

Principe : un seul bit (dit de parit ) est ajout  aux bits de donn es.

- parit  paire : le nombre de bits   1 du mot form  doit  tre pair.
- parit  impaire : le nombre de bits   1 du mot form  doit  tre impair.

Exemple

- parité paire
100 0001 : bits de données
+ 0 : bit de contrôle
= 0100 0001 : mot de code
- parité impaire
101 1001 : bits de données
+ 1 : bit de contrôle
= 1101 1001 : mot de code

05/12/03

Couche liaison

Page 33

Exemple 1/5

$$\begin{array}{r} 1\ 0\ 0\ 1\ 0\ 0\ 0 \\ +\ c_3\ c_2\ c_1\ c_0 \\ \hline \Rightarrow \quad \boxed{1\ 0\ 0\ c_3\ 1\ 0\ 0\ c_2\ 0\ c_1\ c_0} \\ 11\ 10\ 9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array}$$

05/12/03

Couche liaison

Page 37

Code de contrôle de parité

Distance de Hamming = 2

- peut détecter une erreur simple
- ne peut détecter un nombre pair d'erreurs
- peut détecter un nombre impair d'erreurs
- ne peut corriger une erreur simple

Code détecteur

05/12/03

Couche liaison

Page 34

Exemple 2/5

La liste des bits de parité contrôlant un bit donné est fourni par sa représentation binaire

- $11_{10} = 1011_2$
le bit 11 est contrôlé par c_3, c_1 et c_0
- $10_{10} = 1010_2$
le bit 10 est contrôlé par c_3 et c_1

05/12/03

Couche liaison

Page 38

Code de Hamming

Principe : c bits de contrôle (de parité) sont ajoutés aux d bits de données de telle sorte que la séquence des c bits de parité puisse coder :

- la position d'un bit en erreur (soit $d+c$ valeurs)
- ou l'absence d'erreur (soit 1 valeur)

Il faut donc : $2^c \geq d+c+1$

05/12/03

Couche liaison

Page 35

Exemple 3/5

- $9_{10} = 1001_2$
le bit 9 est contrôlé par c_3 et c_0
- $8_{10} = 1000_2$
le bit 8 est contrôlé par c_3 (lui-même)
- $7_{10} = 0111_2$
le bit 7 est contrôlé par c_2, c_1 et c_0
- ...

05/12/03

Couche liaison

Page 39

Mot de code de Hamming

On numérote les bits de chaque mot de code à partir de 1 :

- les bits de contrôle sont placés aux positions représentant des puissances de 2
- les bits de données sont intercalés

05/12/03

Couche liaison

Page 36

Exemple 4/5

La liste des bits contrôlée par chaque bit de parité est donc sur notre exemple :

- c_0 : {1,3,5,7,9,11}
- c_1 : {2,3,6,7,10,11}
- c_2 : {4,5,6,7}
- c_3 : {8,9,10,11}

05/12/03

Couche liaison

Page 40

Exemple 5/5

Le mot transmis est donc (en parité paire) :

1 0 0 1 1 0 0 1 0 0 0

11 10 9 8 7 6 5 4 3 2 1

05/12/03

Couche liaison

Page 41

Exercice

Le codage de Hamming est utilisé. Un récepteur reçoit : 111 0101

Que peut-on en conclure ?

05/12/03

Couche liaison

Page 45

Test à la réception 1/2

- A la réception, les différents test de parité sont effectués.
- Un mot binaire m (de contrôle) constitué d'autant de bits que de bits de contrôle est construit.

05/12/03

Couche liaison

Page 42

Code polynomial

- On considère que les bits d'une séquence binaire sont les coefficients d'un polynôme.
- Exemple :
 $p = 110001$
 $\Rightarrow p(x) = x^5 + x^4 + x^0$
degré de $p = 5$

05/12/03

Couche liaison

Page 46

Test à la réception 2/2

Si le contrôle de parité de c_i est OK alors

$$m[i] \leftarrow 0$$

Sinon

$$m[i] \leftarrow 1$$

Fin si

La valeur de m indique la position du bit en erreur ou 0.

05/12/03

Couche liaison

Page 43

Arithmétique utilisée

Arithmétique modulo 2 (i.e. sans retenue)

- addition et soustraction = ou exclusif
- division
 $11\ 0101\ 1011\ 0000 / 1\ 0011$
quotient = $11\ 0000\ 1010$
reste = 1110

05/12/03

Couche liaison

Page 47

Code de Hamming

Distance de Hamming = 3

- peut détecter 1 ou 2 erreurs
- peut corriger 1 erreur

Code détecteur

05/12/03

Couche liaison

Page 44

Division

Deux méthodes pour l'effectuer à la main :

- division binaire
- division polynomiale

05/12/03

Couche liaison

Page 48

Code polynomial

Principe : l'émetteur et le récepteur se mettent d'accord sur le choix d'un polynôme dit générateur $G(x)$.

Le codage consiste à ajouter des bits de contrôle appelés :

- le total (ou somme) de contrôle
- checksum

05/12/03

Couche liaison

Page 49

Récepteur

- $T(x)$ est le mot transmis par l'émetteur
- $T'(x)$ est le mot reçu par le récepteur
- Le récepteur effectue le test suivant :
 - Si $T'(x) / G(x)$ donne un reste égal à 0 alors
 - pas d'erreur
 - sinon
 - une erreur
 - fin si

05/12/03

Couche liaison

Page 53

Emetteur

- Soit $D(x)$ les données à envoyer et k le degré de $G(x)$
- 1) calculer $D(x) * x^k$
 - revient à ajouter k zéros (poids faibles) à $D(x)$
- 2) calculer $D(x) * x^k / G(x)$
 - on obtient le quotient $Q(x)$ et le reste $R(x)$
- 3) calculer $T(x) = D(x) * x^k - R(x)$
 - revient à remplacer les k zéros par $R(x)$

05/12/03

Couche liaison

Page 50

Normalisation

Plusieurs polynômes ont fait l'objet d'une norme car possédant de bonnes propriétés

$$\text{CRC 12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

$$\text{CRC 16} = x^{16} + x^{15} + x^2 + 1$$

$$\text{CRC CCITT} = x^{16} + x^{12} + x^5 + 1$$

05/12/03

Couche liaison

Page 54

Exemple

- $D(x) = 0101\ 1100$
- $G(x) = 1\ 1000\ 0101$
- $D(x) * x^k =$
0101 1100 0000 0000
- $R(x) =$
0100 1101
- $T(x) =$
0101 1100 0100 1101

05/12/03

Couche liaison

Page 51

Normalisation

Les codes utilisant CRC 16 et CRC CCITT donnent des champs de contrôle d'erreur sur 16 bits.

Ils permettent de détecter :

- toutes les erreurs simples et doubles
- toutes les rafales de longueur ≤ 16
- les rafales de longueur > 17 avec une probabilité supérieure à 99,99%

05/12/03

Couche liaison

Page 55

Propriété

$T(x)$ est toujours divisible par $G(x)$
Le dividende moins le reste est toujours divisible par le diviseur.

Illustration en base 10 :

- 210 278 / 10 941 donne un reste égal à 2399
- 210 278 - 2399 est divisible par 10 941

05/12/03

Couche liaison

Page 52

Réalisation

- Les calculs nécessaires pour obtenir un champ de contrôle semblent complexes.
- Or un simple registre à décalage suffit.
- En pratique, on utilise presque toujours des circuits électroniques pour le codage et décodage polynomial.

05/12/03

Couche liaison

Page 56

Contrôle de flux

05/12/03

Couche liaison

Page 57

Récepteur

Tant que vrai répéter

t ← couchePhysique.donnerTrame()

p ← extrairePaquet(t)

coucheReseau.prendrePaquet(p)

Fin tant que

05/12/03

Couche liaison

Page 61

Contrôle de flux

- Utilisation d'acquittements
- Gestion de temporisateurs
- Numérotation des trames
- Limitation du nombre de trames pouvant être envoyées par l'émetteur

05/12/03

Couche liaison

Page 58

Protocole 2

- Protocole de type « envoyer et attendre »
- Hypothèse levée : mémoire tampon infinie
- Principe
 - le récepteur envoie une trame d'acquittement après chaque trame reçue
 - l'émetteur attend de recevoir un acquittement avant d'émettre la trame suivante

05/12/03

Couche liaison

Page 62

Protocole 1

- Hypothèses
 - mémoire tampon infinie
 - canal parfait (pas de pertes ni d'erreurs)
 - protocole mono-directionnel
 - un émetteur
 - un récepteur

05/12/03

Couche liaison

Page 59

Emetteur

Tant que vrai répéter

p ← coucheReseau.donnerPaquet()

t ← construireTrame(p)

couchePhysique.prendreTrame(t)

couchePhysique.attendreAquittement()

Fin tant que

05/12/03

Couche liaison

Page 63

Emetteur

Tant que vrai répéter

p ← coucheReseau.donnerPaquet()

t ← construireTrame(p)

couchePhysique.prendreTrame(t)

Fin tant que

05/12/03

Couche liaison

Page 60

Récepteur

Tant que vrai répéter

t ← couchePhysique.donnerTrame()

p ← extrairePaquet(t)

coucheReseau.prendrePaquet(p)

couchePhysique.envoyerAcquittement()

Fin tant que

05/12/03

Couche liaison

Page 64

Protocole 3

- Hypothèse levée : canal parfait
⇒ des trames peuvent être erronées
⇒ des trames peuvent être perdues
- Principe
 - Utiliser une méthode de détection d'erreurs
 - Le récepteur émet une trame d'acquittement si la trame arrivée est correcte.
 - L'émetteur ré-émet une trame si aucun ack reçu et si un certain délai de temporisation a expiré

05/12/03

Couche liaison

Page 65

Solution

- En considérant le problème précédent :
 - il s'avère nécessaire de numéroter les trames pour distinguer deux trame successives.
- De plus, comme on le verra en TD :
 - il est préférable que la trame d'acquittement contienne le numéro de la trame qui est acquittée.

05/12/03

Couche liaison

Page 69

Emetteur

```
Tant que vrai répéter
  p ← coucheReseau.donnerPaquet()
  t ← construireTrame(p)
  booléen ack ← faux
  Tant que ack = faux répéter
    couchePhysique.prendreTrame(t)
    ack ← couchePhysique.attendreAcquittement()
  Fin Tant que
Fin tant que
```

05/12/03

Couche liaison

Page 66

Protocole 4

- Protocole à fenêtres d'anticipation (sliding windows).
- Deux fenêtres sont gérées par chaque entité de couche liaison. En effet :
 - Toute entité émettrice possède une fenêtre d'anticipation appelée fenêtre d'émission
 - Toute entité réceptrice possède une fenêtre d'anticipation appelée fenêtre de réception

05/12/03

Couche liaison

Page 70

Récepteur

```
Tant que vrai répéter
  t ← couchePhysique.donnerTrame()
  Si estCorrecte(t) alors
    p ← extrairePaquet(t)
    coucheReseau.prendrePaquet(p)
    couchePhysique.envoyerAcquittement()
  Fin si
Fin tant que
```

05/12/03

Couche liaison

Page 67

Fenêtre d'émission

- La fenêtre d'émission indique la liste des numéros de trames dont on attend l'acquittement.
- Elle possède une taille maximale maxE indiquant le nombre maximal de trames qui peuvent être envoyées sans se préoccuper des acquittements.
- Elle possède une taille courante variable curE de valeur inférieure ou égale à maxE. On a toujours :

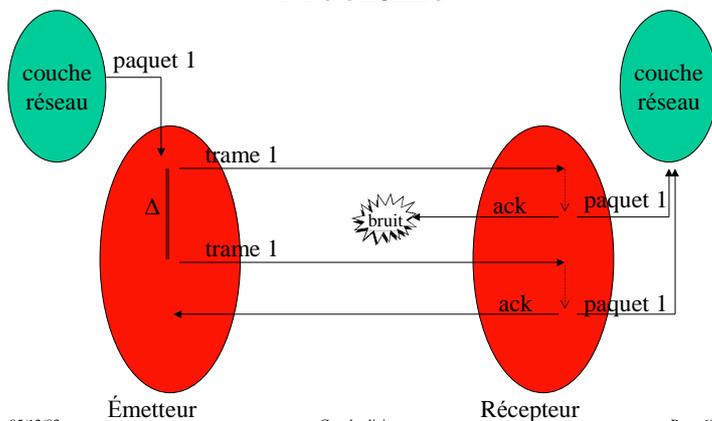
$$0 \leq \text{curE} \leq \text{maxE}$$

05/12/03

Couche liaison

Page 71

Problème



05/12/03

Couche liaison

Page 68

Bornes

- La fenêtre est représentée par deux frontières (bornes).
 - la frontière droite (borne supérieure) est incrémentée de 1 à chaque envoi
 - la frontière gauche (borne inférieure) est incrémentée de 1 à chaque acquittement reçu correspondant à la trame associée à cette frontière

05/12/03

Couche liaison

Page 72

Mémoires tampons

- Coté émetteur, maxE mémoires tampons sont nécessaires pour stocker les trames.
- En effet, il est possible qu'il faille envoyer à nouveau une ou plusieurs trames.

05/12/03

Couche liaison

Page 73

Bornes

- La fenêtre est représentée par deux frontières (bornes).
- les deux frontières sont toutes deux incrémentées de 1 chaque fois qu'une trame correcte correspondant à la frontière droite (borne inférieure) est reçue.

05/12/03

Couche liaison

Page 77

Exemple

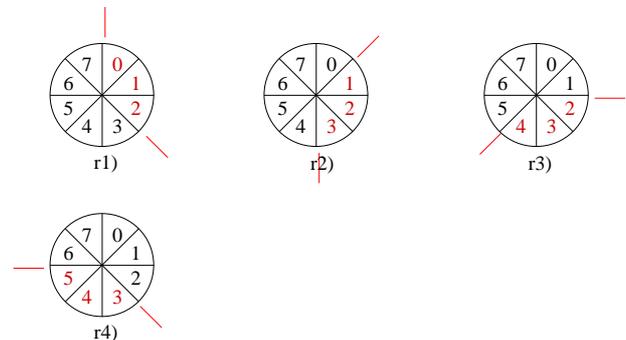
- Considérons que :
 - les trames soient numérotés de 0 à 7 (sur 3 bits)
 - maxE = 3
 - curE = 0 (initialement)

05/12/03

Couche liaison

Page 74

Exemple

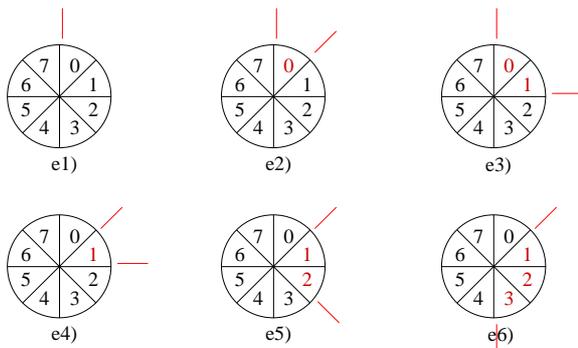


05/12/03

Couche liaison

Page 78

Exemple



05/12/03

Couche liaison

Page 75

Pipelining

- Utiliser des fenêtre d'anticipation permet d'utiliser la technique de pipelining.
- Cela consiste à envoyer plusieurs trames successivement sans attendre de recevoir les trames d'acquittement.
- Il est important de bien régler la largeur des fenêtres pour améliorer l'efficacité du pipelining.

05/12/03

Couche liaison

Page 79

Fenêtre de réception

- La fenêtre de réception indique la liste des numéros de trame attendus.
- La fenêtre de réception possède une taille R qui ne varie pas (sauf cas particuliers).

05/12/03

Couche liaison

Page 76

Calcul de l'efficacité

- Temps d'émission $T_e(t)$ = temps nécessaire pour que la trame t passe sur le canal.
 $T_e(t) = \text{length}(t)/d$ où :
 d = débit du canal (en bits/s)
 $\text{length}(t)$ = longueur de la trame t (en bits)
- Temps de propagation T_p = temps nécessaire pour qu'un bit passe de l'émetteur au récepteur.
- Temps de traitement T_t = temps nécessaire au récepteur pour traiter une trame.

05/12/03

Couche liaison

Page 80

Calcul de l'efficacité

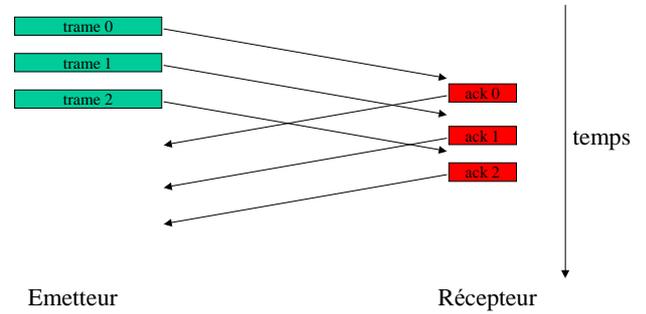
- Hypothèse : toutes les trames sont de longueur m .
- Temps d'attente T_a = temps nécessaire pour recevoir l'acquittement d'une trame émise.
 $T_a = T_e(t) + T_p + T_t + T_e(\text{ack}) + T_p$ où :
 - t désigne une trame quelconque
 - ack désigne une trame quelconque d'acquittement.
- Temps d'émission maximale T_{em} = temps maximal pendant lequel il est possible d'émettre des trames. $T_{em} = \max E * m / d$

05/12/03

Couche liaison

Page 81

Acquittement « individuel »



05/12/03

Couche liaison

Page 85

Efficacité

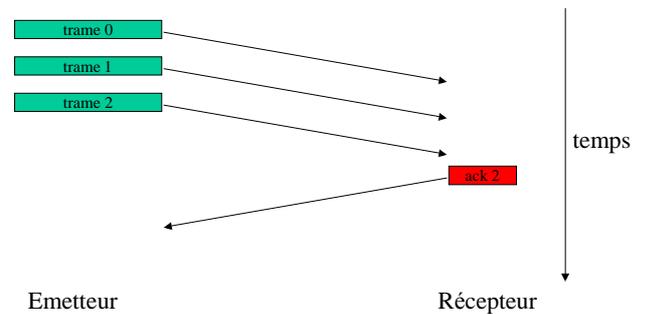
- Efficacité :
 - 100% si $T_{em} \geq T_a$
 - T_{em}/T_a sinon
- NB : calcul théorique ne prenant en compte
 - ni les erreurs de transmission
 - ni les problèmes de surcharge du récepteur

05/12/03

Couche liaison

Page 82

Acquittement « collectif »



05/12/03

Couche liaison

Page 86

Exercice

- Soient $d = 50\text{kb/s}$, $m = 1000\text{ bits}$, $T_p = 250\text{ms}$, T_t et ack considérés comme négligeables (nuls)
- Calculer l'efficacité pour $\max E = 1$
- Calculer l'efficacité pour $\max E = 25$

05/12/03

Couche liaison

Page 83

Contrainte

- Il est nécessaire que :
 - $\max E < n$où n représente le nombre de numéros de trames (allant de 0 à $n-1$) pouvant être utilisés.
- En effet, comme le récepteur peut envoyer plusieurs fois le même acquittement, une ambiguïté peut apparaître.

05/12/03

Couche liaison

Page 87

Acquittements

- Lorsque plusieurs trames doivent être acquittées, il est possible :
 - d'envoyer un acquittement « individuel » pour chaque trame
 - d'envoyer un acquittement « collectif » en indiquant
 - le plus grand numéro de trame parmi celles qui sont acquittées
 - ou le numéro de la prochaine trame attendue.

05/12/03

Couche liaison

Page 84

Exemple

- $\max E = 8$ et $n = 8$
- L'émetteur envoie une série de trames numérotées de 0 à 7,
 - L'émetteur reçoit un acquittement « collectif » avec le numéro 7,
 - L'émetteur envoie une nouvelle série de trames numérotées de 0 à 7,
 - L'émetteur reçoit un acquittement « collectif » avec le numéro 7.

05/12/03

Couche liaison

Page 88

Erreurs de transmission

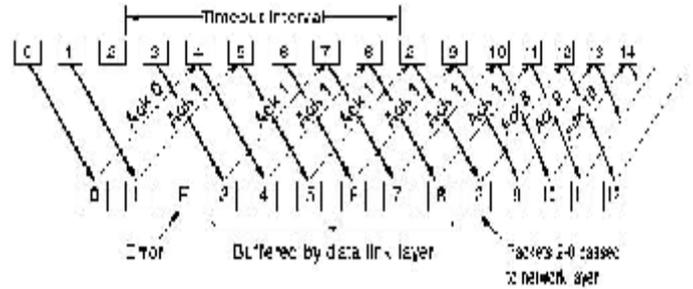
- Si une trame située au milieu d'une série est perdue ou erronée ?
- Deux techniques de rejet sont possibles :
 - technique du rejet total
 - technique du rejet sélectif

05/12/03

Couche liaison

Page 89

Exemple



05/12/03

Couche liaison

Page 93

Technique de rejet total

- Le récepteur rejettent toutes les trames qui suivent celle qui est erronée.
 - inconvénient : le canal est mal exploité
 - avantage : pas besoin de mémoires tampons
- Cette technique correspond à l'utilisation d'une fenêtre de réception de taille 1.

05/12/03

Couche liaison

Page 90

Technique de rejet sélectif

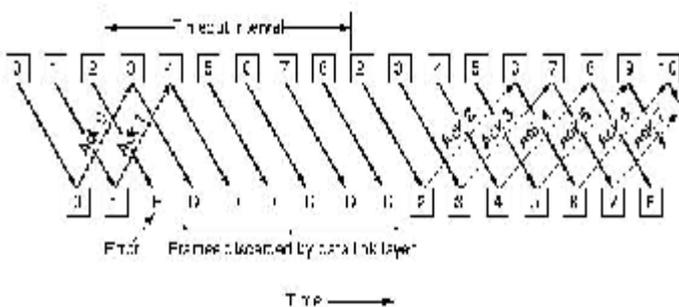
- La technique de rejet sélectif impose une nouvelle contrainte sur la valeur de $\max E$: il faut choisir une valeur au plus égale à $n/2$
- En effet, un problème survient lorsque une nouvelle fenêtre au niveau de l'émetteur se superpose à une ancienne.

05/12/03

Couche liaison

Page 94

Exemple



05/12/03

Couche liaison

Page 91

Superposition

- Lorsque la communication est bidirectionnelle (un émetteur et un récepteur de chaque côté), il est possible d'utiliser un champ des trames de données pour indiquer un acquittement.
- Il s'agit d'une technique de superposition (piggybacking)
- Avantage : quelques bits versus une trame
- Inconvénient : attendre une trame de données
- NB : si pas de trames de données à envoyer dans un certain délai, envoyer une trame de contrôle.

05/12/03

Couche liaison

Page 95

Technique de rejet sélectif

- Le récepteur accepte les suivantes (en les stockant) jusqu'à une certaine limite donnée par R.
 - avantage : le canal est mieux exploité
 - inconvénient : besoin de mémoires tampons
- Cette technique correspond à l'utilisation d'une fenêtre de réception de taille supérieure à 1.
- Le récepteur utilise un acquittement « collectif ».

05/12/03

Couche liaison

Page 92