

Chapitre IV

La couche liaison



4. La couche liaison

1

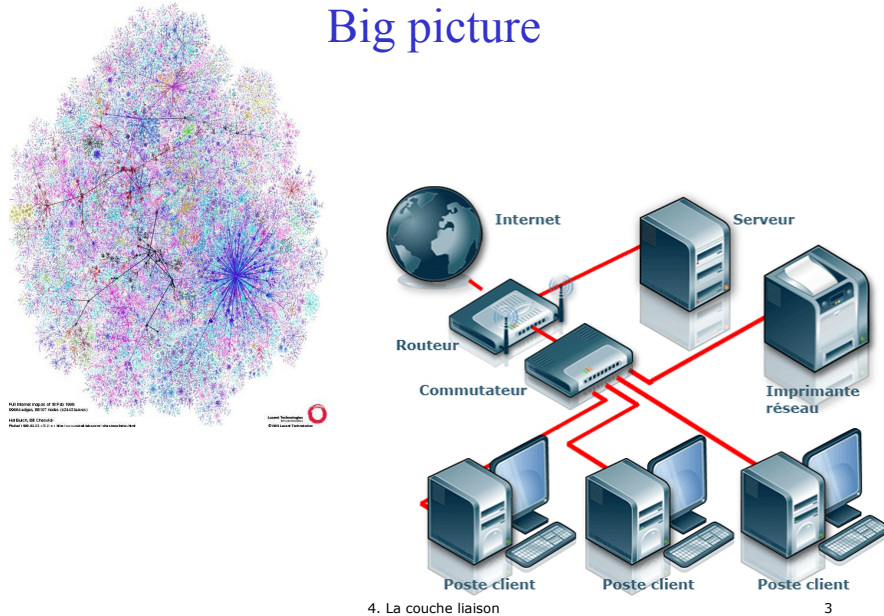
But du chapitre

- Connaître les fonctions et algorithmes de la couche liaison
 - Le **découpage en trames** : orienté bit, orienté caractère
 - Les **méthodes de retransmission** « Envoyer et attendre », « Go-back-N »
 - L'acquittement de données et l'utilisation des numéros de séquence
 - L'établissement et la terminaison de connexion

4. La couche liaison

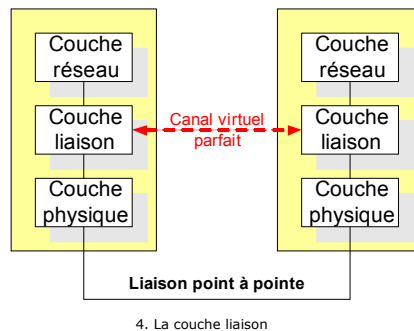
2

Big picture



Introduction

- Transmission point à point entre deux **systèmes adjacents**
- Les protocoles de la couche liaison essaient de fournir un **canal virtuel fiable** à la couche supérieure



Service offert par la couche liaison

Service principal:

Transmission de paquets entre les entités de la couche réseau de deux stations voisines

D'après la fiabilité du service on distingue trois modes

1. **Sans connexion, sans accusé de réception**
 - Utilisé sur les liens fiables, ou si une couche supérieure effectue la correction d'erreurs
2. **Sans connexion, avec accusé de réception**
 - Chaque trame est acquittée avant d'envoyer la suivante
 - Utilisé sur des liens peu fiables et peu performants
3. **Avec connexion, avec accusé de réception**
 - Permet d'utiliser des algorithmes plus complexes pour augmenter la fiabilité et les performances

4. La couche liaison

5

Résumé des fonctions réalisées à la couche liaison

1. Découpage du flot de bits en trames
2. Contrôle d'erreurs
3. Retransmission de trames erronées
 - Acquitements
 - Numéros de séquence
 - Stratégies ARQ
4. Établissement et terminaison de connexions

4. La couche liaison

6

1. Découpage en trames

Couche physique :

Fournit un service de transmission de bits

Couche liaison :

Doit fournir un service orienté paquet à la couche réseau

Trois méthodes principales de découpage en trames

- Comptage de caractères
- Découpage orienté caractère
- Découpage à l'aide d'un fanion (orienté bit)

Comptage de caractères

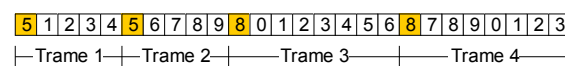
- Méthode la plus simple et la moins fiable
 - En pratique, seulement en combinaison avec d'autres méthodes

Principe

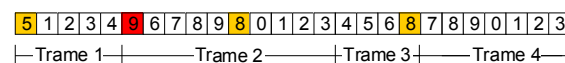
- Chaque trame contient dans l'entête un champ « Longueur de trame » qui permet au récepteur de découper les trames

Problème

- Une erreur bit peut laisser le récepteur sans orientation



a) Sans erreur



b) Avec une erreur

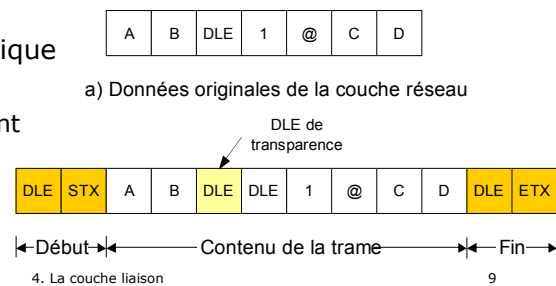
Découpage orienté caractère

- Chaque trame est encadrée par deux séquences
 - DLE + STX au début (Data Link Escape, Start of TeXt)
 - DLE + ETX à la fin (Data Link Escape, End of TeXt)
- « Byte Stuffing » (DLE de transparence)
 - Pour éviter d'interpréter des données de la couche réseau comme début /fin de trame, on utilise une méthode appelée « Byte stuffing »

➤ Souvent utilisé en pratique

- **Inconvénient**

- Fonctionne uniquement avec un codage sur 8 bits

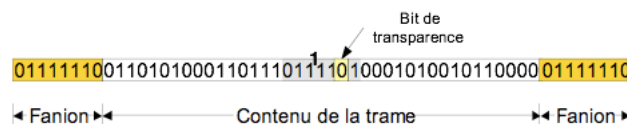


Découpage à l'aide d'un fanion

- Méthode de découpage plus générale
 - Permet à une trame d'avoir une longueur quelconque
 - Utilisable avec tous les codages
- Principe
- Chaque trame commence et finit par un **fanion** (flag) « 01111110 »
 - Bit stuffing (Bit de transparence)
 - Évite d'interpréter des données de la couche réseau comme fanion
 - Après 5 bits consécutifs à 1, ajouter un bit 0 de transparence

011010100011011101111100010100101100000

a) Données originales de la couche réseau



Exercices 1,3,4

2. Contrôle d'erreurs

Problème

- Les protocoles de la couche physique ne sont pas parfaits mais subissent des erreurs bit
- Le **taux d'erreur bit** dépend surtout du média de transmission
 - Fibres optiques: $\sim 10^{-12}$
 - Canal radio: $\sim 10^{-5}$

La couche liaison implémente deux fonctions

- **Détection d'erreurs** (obligatoire)
 - Codes détecteurs
- **Correction d'erreurs** (seulement pour un service fiable)
 - Code correcteurs
 - Retransmission de trames

Critères d'efficacité d'un contrôle d'erreur

1. Capacité de détecter / corriger des **erreurs multiples**
2. Capacité de détecter / corriger des **rafales d'erreurs** d'une certaine longueur
3. Probabilité **d'accepter une trame erronée** comme correcte
4. Rendement:
rapport entre les bits de données et la longueur totale des paquets

Codes simples: les parités

Méthode la plus simple de détection d'erreurs

- Ajouter un seul bit de parité à la fin d'un bloc de données
 - Parité paire :
le nombre total de bits 1 (y compris la parité) est pair
 - Parité impaire :
le nombre total de bits 1 (y compris la parité) est impair
- **Code détecteur** d'erreurs
 - Toutes les erreurs bit sur un **nombre impair de bits** sont détectables pour une **longueur de données quelconque**
 - **En pratique, seulement la moitié des erreurs bit sont détectées**
 - La plupart des erreurs bit apparaissent en rafale, donc un nombre pair d'erreurs est aussi probable qu'un nombre impair

Parités verticales et horizontales

- Permettent de construire un code correcteur simple
 - Arranger la séquence des bits en une matrice
 - Calculer une parité par ligne et par colonne

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	0	1	1	1	0
0	0	1	1	0	0	1	1
1	0	1	1	1	1	1	0

Parités
horizontales

Parités verticales

4. La couche liaison

15

Parités verticales et horizontales

- Permettent de corriger toutes les erreurs simples
- Permettent de détecter toutes les erreurs sur 2 ou 3 bits
- Permettent de détecter toutes les rafales plus courtes que la longueur d'une ligne
- 4 erreurs bit peuvent passer sans être détectées

1	0	0	1	0	1	0	1
0	1	1	1	0	1	0	0
1	1	0	0	1	0	1	0
1	0	0	0	1	1	1	0
0	0	0	1	1	0	1	1
1	0	1	1	1	1	1	0

Parités
horizontales

Parités verticales

4. La couche liaison

16

Codes correcteurs: Bases de la théorie de codage

- Distance de Hamming :
nombre de bits différents des deux mots de code
 - Pour détection d erreurs de bit il faut que le code ait une distance de Hamming minimale de $d+1$
 - Pour corriger d erreurs de bit le code doit avoir une distance de $2d+1$
- Combien de bits de contrôle faut il pour un code correcteur d'erreurs simples ?

$$(m + r + 1) \leq 2^r$$

4. La couche liaison

17

Codes de Hamming

- Vecteur des m bits de données: x^T
- Vecteur des n bits du mot de code: y^T
 - m bits de données suivis de r bits de contrôle

Matrice génératrice du code: H

- Dimensions: $m \times n$
- Composée d'une matrice unité I_m et d'une matrice P de dimensions $m \times r$: $H = [I_m ; P]$
- Génération des mots de code :

$$y^T = x^T H$$

4. La couche liaison

18

Exemple

- Code correcteur pour toutes les erreurs simples sur 4 bits de données
- 3 bits de contrôle sont nécessaire ($4+3+1 \leq 2^3$)
- Matrice génératrice du code

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Bits de contrôle

$$a_1 = x_1 + x_2 + x_3$$

$$a_2 = x_2 + x_3 + x_4$$

$$a_3 = x_1 + x_2 + x_4$$

- Par exemple: $(1010) \cdot H = (1010011)$

Décodage d'un code de Hamming

- Matrice de contrôle de parité : G^T
 - Dimensions : $n \times r$
 - Composée de la matrice P et d'une matrice unité I_r :

$$G^T = \begin{bmatrix} P \\ I_r \end{bmatrix}$$

- Calcul du syndrome s d'un mot : $s^T = y^T G^T$
- Si le syndrome vaut 0, le mot de code est correct

$$y^T G^T = x^T H G^T = x^T \begin{bmatrix} I_m & P \end{bmatrix} \begin{bmatrix} P \\ I_r \end{bmatrix} = 0$$

Exemple (suite)

Matrice de contrôle de parité Calcul du syndrome

$$G^T = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1010011) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (000)$$

4. La couche liaison

21

Exemple – Mot incorrect

Calcul du syndrome

- Si le mot de code est faux, le syndrome indique la position du bit à corriger

$$(1010001) \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (010)$$

4. La couche liaison

22

Exercice 12,14

Efficacité des codes correcteurs

- Rendement d'un code : $R = m/(m + r)$
- Rendement des codes de Hamming :

$$R = \frac{n - r}{n} = 1 - \frac{\log_2(n + 1)}{n}$$

- Exemple :
 - Mot de données sur 8 bits
 - 4 bits de contrôle sont nécessaires
 - Rendement $R = 67\%$

Exemple d'un canal de transmission

- Taux d'erreurs bit : $BER = 10^{-6}$
 - Longueurs des blocs de données : $m = 1000$ bits
 - Nombre de bits de contrôle : $r = 10$ bits
- Transmission de 1 Mb $\rightarrow$ 1000 trames
- Code correcteur Hamming:
 - 10'000 bits de contrôle sont transmis
 - Code détecteur avec parités simples:
 - 1000 bits de contrôle
 - Probabilité d'erreur d'une trame : $p = 1000 \cdot 10^{-6} = 10^{-3}$
 - Retransmission d'une seule trame erronée (en moyenne) : 1000 bits
- **Surcharge totale avec code correcteur et retransmission: 2000 bits**

Code détecteurs d'erreurs

- Codes polynomiaux ou codes cycliques (CRC)
 - Permettent de contrôler des mots d'une longueur variable
 - Réalisation simple et efficace en matériel
 - Très bonnes capacités de détection d'erreurs

Principe des codes polynomiaux

- Un bloc de m bits est vu comme un polynôme $M(x)$ de degré $m-1$ avec des coefficients binaires
 - Exemple : 110001 $\rightarrow M(x) = x^5 + x^4 + x^0$
- Polynôme générateur : $G(x)$ de degré r
- Le mot de code correspond à un polynôme $T(x)$ de degré $m-1+r$ avec $T(x) / G(x) = 0$

Un code polynomial est un code dans lequel tous les mots de code, représentés par des polynômes $T(x)$, sont des multiples d'un polynôme générateur $G(x)$

4. La couche liaison

27

Calcul d'un mot de code

1. Ajouter r zéros après le bit de poids faible du bloc
Il contient ainsi $m + r$ bits, correspondant au polynôme $x^r M(x)$
2. Division modulo 2 du polynôme $x^r M(x)$ par $G(x)$
3. Soustraire modulo 2 le reste de la division de la chaîne de bits correspondant au polynôme $x^r M(x)$
4. Le résultat de cette opération est la trame à transmettre

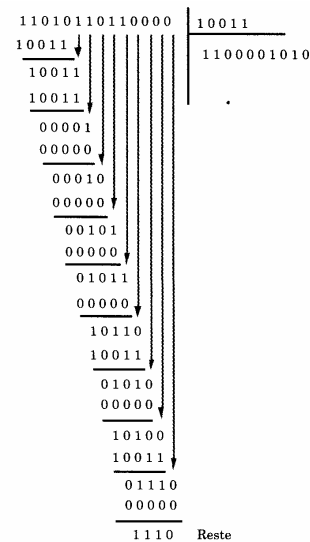
4. La couche liaison

28

Exemple

- Trame: 1101011011
- Générateur: 10011
 $\rightarrow G(x) = x^4 + x + 1$

➤ Trame transmise:
 11010110111110



4. La couche liaison

29

Détection d'erreurs

- Le récepteur calcule $T(x) / G(x)$
- Il accepte la trame comme correcte si le reste est 0
- En cas d'erreurs bit, la trame reçue correspond à un polynôme $T'(x) = T(x) + E(x)$
- Une erreur est détectée si le reste de $E(x) / G(x)$ n'est pas nul

4. La couche liaison

30

Capacité de détection d'erreurs

1. Toute erreur simple est détectée
 - Si le polynôme générateur $G(x)$ comporte plus d'un coefficient non nul
2. Les erreurs doubles sont toutes détectées
 - Si le polynôme générateur $G(x)$ ne divise pas $x^k + 1$, où k peut prendre n'importe quelle valeur comprise entre 1 et $n-1$
3. Une erreur comportant un nombre impair d'erreurs est détectée
 - Si le polynôme générateur comporte $(x + 1)$ en facteur
4. Toutes les rafales d'erreurs de longueur r sont détectées
 - Pour un polynôme générateur $G(x)$ de degré r
5. Les rafales d'erreurs d'une longueur supérieure à r sont détectées avec une probabilité de $1-(1/2^{r-1})$

Polynômes générateurs normalisés

- CRC-12 : $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$
- CRC-16 : $x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT : $x^{16} + x^{12} + x^5 + 1$
- CRC-32 : $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$
- Le CRC-32 est utilisé pour Ethernet
 - Détecte toutes les rafales d'erreurs de 32 bits
 - Probabilité qu'une rafale plus longue ne soit pas détectée:
 $P = 4,6 \cdot 10^{-10}$

Exercices 17,18,15

4. La couche liaison

33

Stratégies de retransmission

- Méthode principale de correction d'erreurs :
[Retransmission des trames erronées](#)
 - Plus efficace que les codes correcteurs

Stratégies ARQ : (*Automatic Repeat reQuest*)

1. « Envoyer et attendre » ou « Stop-and-Go »
2. « Go-back-n » ou « Fenêtre glissante »
3. « Rejet sélectif »

4. La couche liaison

34

Envoyer et attendre

Principe

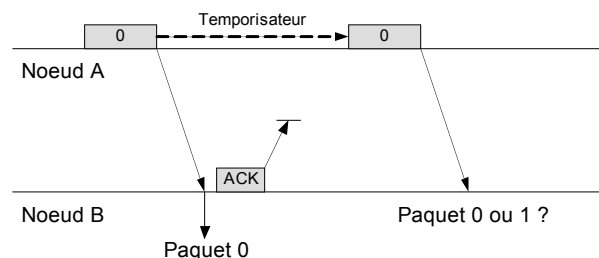
- S'assurer qu'un paquet a été reçu avant d'envoyer le suivant

Protocole simple

- Le paquet contient un CRC qui permet de détecter des erreurs
- Si le paquet est correct, le récepteur renvoie un **acquittement positif** (ACK)
- Si le paquet est erroné, le récepteur renvoie un **acquittement négatif** (NAK)
- ACK et NAK sont également protégés par un CRC
- Lors de la réception d'un NAK le paquet est retransmis

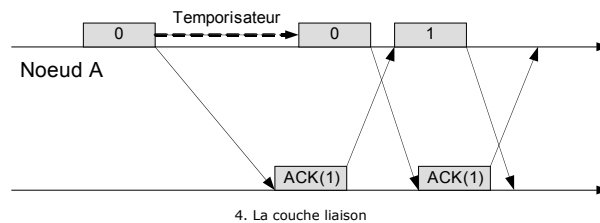
Faiblesse du protocole avec ACK/NAK

- Le protocole simple n'est utilisable que
 - Si aucun paquet ne peut être perdu
 - S'il n'y a pas d'ambiguïté concernant les ACK/NAK
- Il peut créer un comportement incorrect en cas de perte d'un acquittement



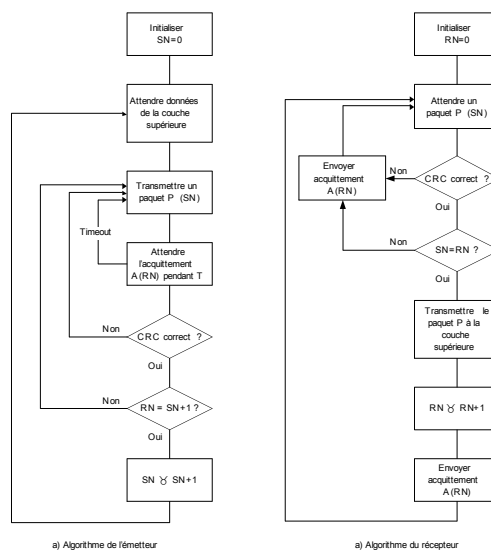
Protocole avec numéros de séquence

- Tous les paquets sont numérotés avec un **numéro de séquence SN**
- Un ACK contient le **numéro de séquence RN du prochain paquet attendu**
 - NAK n'est plus nécessaire, comme le récepteur peut redemander la transmission du paquet précédent avec un ACK
- Un numéro de séquence sur 1 bit est suffisant
 - Un ACK confirme soit le dernier paquet soit son prédécesseur



37

Algorithme



4. La couche liaison

38

Performances de « Envoyer et attendre »

Taux d'utilisation du réseau ρ :

- Rapport entre le débit effectif obtenu D et le débit du lien C

- Calcul

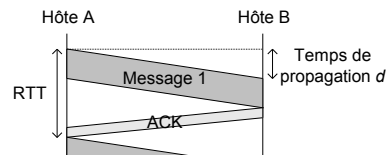
- l_{data} : longueur d'un paquet
- l_{ACK} : longueur d'un ACK

$$RTT = (l_{data} + l_{ACK}) / C + 2d_{prop}$$

$$\rho = \frac{l_{data}}{RTT \cdot C}$$

- Utilisation est inversement proportionnelle au « produit largeur de bande – délai » (*bandwidth delay product*)

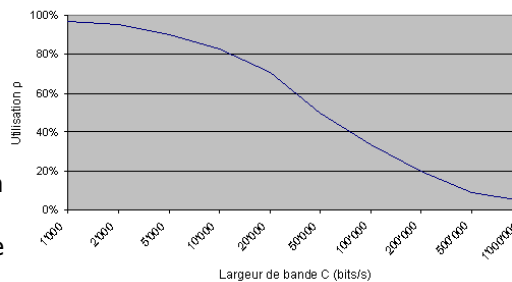
$$BWD = RTT \cdot C$$



Exemple numérique

- Paramètres:

- $l_{data} = 1000$ bits
- $l_{ack} = 10$ bit
- Délai de propagation $d = 10$ ms
- Débit binaire C entre 1 kb/s et 1 Mb/s



- Utilisation inférieure à 5% à $C = 1$ Mb/s
- « Envoyer et attendre » n'est approprié que pour les liens à faible produit « largeur de bande – délai »

Protocole « Go-back-n »

- Méthode ARQ très répandue dans les protocoles de la couche liaison et de la couche transport

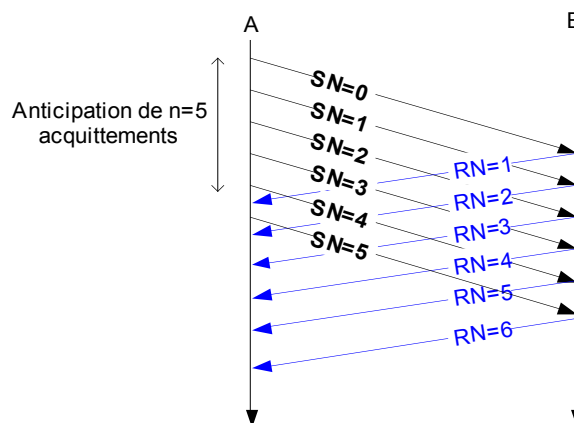
Principe

- L'émetteur envoie **plusieurs paquets de suite** sans attendre d'acquittement
- Tous les paquets contiennent un **numéro de séquence SN**
- L'émetteur conserve une copie des paquets envoyés mais non acquittés afin de pouvoir les retransmettre
- Le paramètre n indique combien de paquets peuvent être envoyés sans recevoir d'acquittement
- Le récepteur **n'accepte que le prochain paquet attendu**
- Le récepteur envoie un acquittement pour chaque paquet correctement reçu

4. La couche liaison

41

Exemple



4. La couche liaison

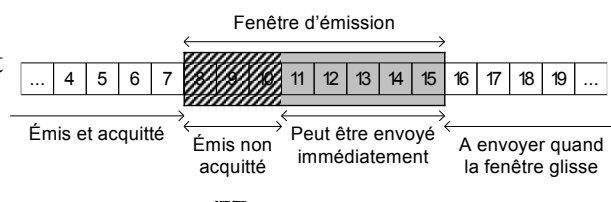
42

La fenêtre glissante

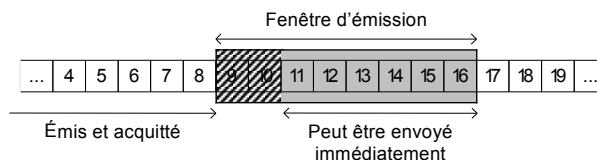
- La méthode « Go-back-n » peut être réalisée à l'aide d'une fenêtre glissante (*Sliding Window*)
 - La fenêtre contient les n paquets après le dernier paquet acquitté $s-1$
 - Numéros de séquence SN autorisés à l'expédition :
$$s \leq SN < s + n$$
 - Après l'acquittement d'un paquet transmis, la fenêtre glisse à droite et autorise la transmission d'un nouveau paquet

Exemple

Dernier acquittement
reçu : RN=8



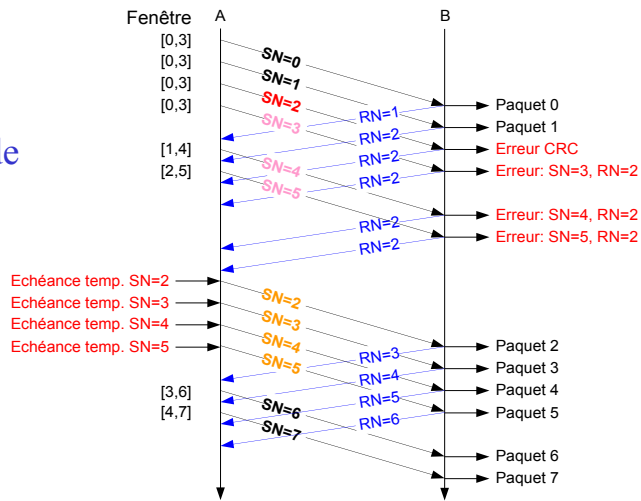
Nouvel acquittement
reçu : RN=9



Retransmission : Gestion passive

La retransmission est déclenchée par un **temporisateur de retransmission**

Tous les paquets à partir du paquet perdu sont retransmis



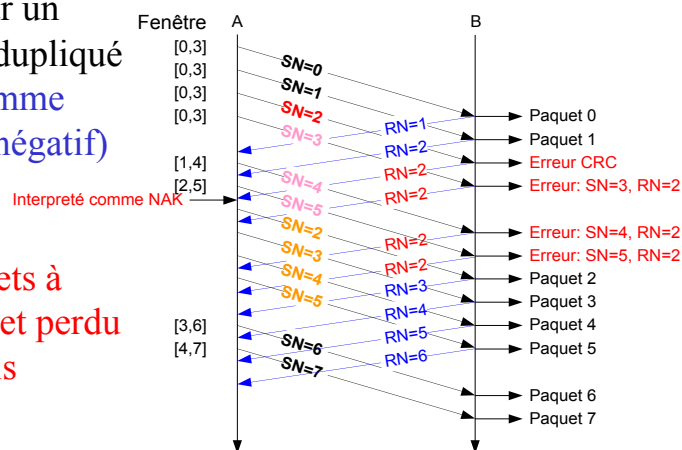
4. La couche liaison

45

Retransmission : Gestion active

La retransmission est déclenchée par un **acquittement dupliqué (interprété comme acquittement négatif)**

Tous les paquets à partir du paquet perdu sont retransmis



4. La couche liaison

46

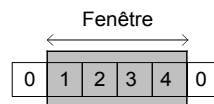
Numéros de séquence dans « Go-back-n »

- Un champ avec une longueur fixe doit être prévu dans l'entête des paquets pour le numéro de séquence
 - Numérotation modulo M

Quelle valeur choisir pour M pour une fenêtre de n ?

- A chaque instant, au maximum n paquets peuvent être en attente d'un acquittement
- Au maximum n+1 acquittements différents sont possibles

➤ $M = n + 1$ est suffisant



Rejet Sélectif

Problème de Go-back-n

Une erreur dans un paquet provoque la retransmission de tous les paquets à partir du paquet erroné

- Le récepteur n'accepte pas de paquet hors séquence

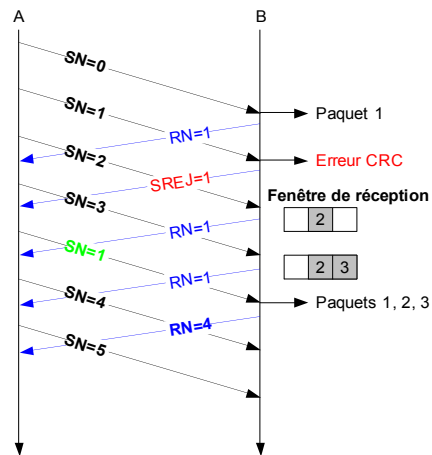
Principe du rejet sélectif

- Le récepteur gère un tampon de réception dans lequel il place les paquets reçus hors séquence
- Un nouveau type d'acquittement négatif SREJ permet d'indiquer le paquet rejeté

Rejet sélectif

Principe

- Lorsqu'un paquet erroné est reçu, le récepteur renvoie un **SREJ** avec le numéro de séquence du paquet erroné
- Les paquets reçus hors séquence sont placés dans une **fenêtre de réception**, avec la même taille que la fenêtre d'émission
- L'émetteur retransmet uniquement les paquets pour lesquels il a reçu un SREJ



4. La couche liaison

49

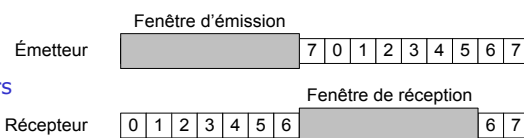
Gestion des numéros de séquence

Numéros de séquence sur 3 bits
Taille de fenêtre = 7

1. L'émetteur transmet toute la fenêtre
2. Les paquets sont reçus correctement mais tous les acquittements sont perdus
3. Après l'expiration du temporisateur l'émetteur retransmet le paquet 0
4. Le paquet 0 se trouve dans la fenêtre de réception. **Il est alors accepté par le récepteur**
5. **Erreur !!!**



a) État initial



b) Après l'émission de 7 paquets et la perte de 7 acquittements

4. La couche liaison

50

Gestion des numéros de séquence

Problème dans l'exemple précédent

- Les numéros de séquence des deux fenêtres se superposent
- L'émetteur a le droit de transmettre tous les numéros de séquence SN avec $SN \in [SN_{\min}, SN_{\min} + n - 1]$
- Après la réception du paquet $SN_{\min} + n - 1$, le récepteur accepte les paquets $SN \in [SN_{\min} + n, SN_{\min} + n + n - 1]$
- En fonction des acquittements reçus, le prochain paquet transmis se trouve dans $SN \in [SN_{\min}, SN_{\min} + 2n - 1]$

Condition pour Rejet sélectif :

$$M \geq 2n$$

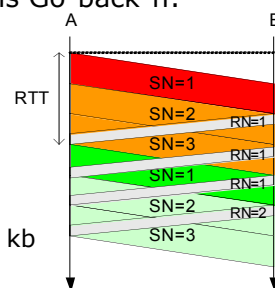
Amélioration des performances par rapport à Go-back-n

- Données retransmises en cas d'erreur dans Go-back-n:
 - $\text{Min}(\text{RTT} \cdot C, \text{taille de la fenêtre})$

Exemple numérique :

- $P(\text{paquet erronée}) = 10^{-4}$
- Longueur de paquet = 1000 bits
- En moyenne une erreur toutes les 10s
- Canal avec $\text{RTT} \cdot C = 10 \text{ ms} \cdot 1 \text{ Mb/s} = 10 \text{ kb}$

- Surcharge pour Go-back-n $\approx 10 \text{ kb} / 10 \text{ Mb} = 0.1 \%$
- Surcharge pour Rejet sélectif $\approx 1 \text{ kb/s} / 10 \text{ Mb/s} = 0.01\%$



Efficacité des protocoles

Envoyer et attendre

$$\rho = \frac{l_{data}}{RTT \cdot C} = \frac{l_{data}}{l_{data} + l_{ACK} + 2dC}$$

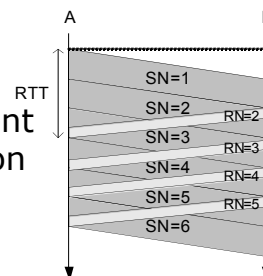
- Taux d'utilisation faible pour les canaux avec un produit largeur de bande – délai élevé
 - Typiquement des canaux avec un délai de propagation élevé

Taille optimale de la fenêtre glissante

- La taille optimale doit permettre à l'émetteur de transmettre en permanence
- La taille de fenêtre (en bytes) correspond alors au produit largeur de bande – délai du canal :

$$n = \left\lceil \frac{RTT \cdot C}{l_{data}} \right\rceil$$

- Go-back-n et Rejet sélectif peuvent alors atteindre un taux d'utilisation (sans erreurs) de $\rho=1$



Taux d'utilisation en cas d'erreurs

Hypothèse :

Uniquement les erreurs dans les paquets de données sont considérées, les erreurs dans les acquittements sont négligées

Rejet sélectif

- Le nombre de paquets reçus correctement est le nombre de paquets émis moins les paquets erronés
(p : probabilité d'une erreur dans un paquet)

- Taux d'utilisation

$$n_{reçu} = n_{émis} - p \cdot n_{émis}$$

$$\rho_{max} = \frac{n_{reçu}}{n_{émis}} = 1 - p$$

4. La couche liaison

55

Taux d'utilisation de Go-back-n

- En cas d'une erreur :
Nombre de paquets retransmis : $\beta = \min(\lceil RTT \cdot C/l_{data} \rceil, n)$
- Espérance no. paquets émis / paquet reçu

Nombre de retransmissions d'un paquet	Probabilité	Nombre de paquets transmis
0	$1-p$	1
1	$(1-p)p$	$1+\beta$
2	$(1-p)p^2$	$1+2\beta$
k	$(1-p)p^k$	$1+k\beta$

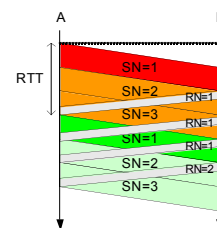
$$E\left[\frac{n_{émis}}{n_{reçu}}\right] = \sum_{k=0}^{\infty} (1+k\beta)(1-p)p^k = \frac{1-p+p\beta}{1-p}$$

- Taux d'utilisation :

$$\rho_{max} = E\left[\frac{n_{reçu}}{n_{émis}}\right] = \frac{1-p}{1-p+p\beta}$$

4. La couche liaison

56



Exemple numérique

Débit binaire	$C=1\text{ Mb/s}$
Longueur de trame	$l_{\text{data}} = 1000\text{ bits}$
Longueur d'ACK	$l_{\text{ACK}} = 1000\text{ bits}$
Délai de propagation	$d=4\text{ ms}$
Taux d'erreurs bit	$\text{BER}=10^{-6}$
Longueur de trame	$l_{\text{data}}=1000\text{ bits}$

Calculer

- La probabilité p d'une trame erronée
- Taille optimale de la fenêtre ($n = \beta$)
- Le taux d'utilisation maximum (avec erreurs) pour
 - Envoyer et attendre
 - Go-back-n
 - Rejet sélectif

Exercices 27,28