

Algorithmes de reprises sur erreurs (ARQ)

Marceau Coupechoux
 Philippe Godlewski
 Philippe Martins
 UE RES222 « Accès et Mobilité »
 Télécom ParisTech
 Département Informatique et Réseaux

Plan

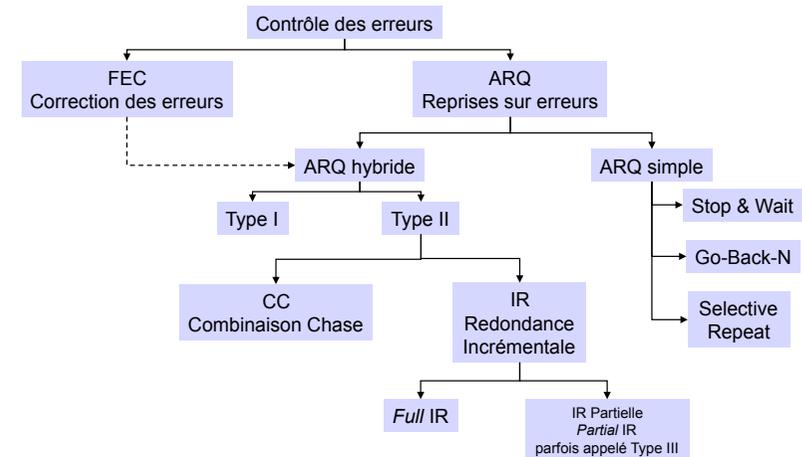
- Introduction
- Contrôle des erreurs
- Détection des erreurs
- Algorithmes d'ARQ
- Algorithmes d'ARQ hybride
- Le protocole LAP-B
- Autres exemples d'implémentations
- Conclusion

Introduction

- ARQ : utilisé depuis plus de 60 ans en radio communications
 - ARQ-M: protocole de télégraphie radio mis en place entre les Pays-Bas et les États-Unis en 1947
 - LAP-B et variantes (par exemple LAP-Dm en GSM)
 - Présent dans toutes les couches RLC des systèmes mobiles (UMTS, LTE, etc.)
- HARQ : utilisé dans tous les systèmes modernes (HSPA, mobile WiMAX, LTE, LTE-A)
 - Type I: Ref: K. Brayer in IEEE Trans. Communications, April 1968
 - Type II: D. M. Mandelbaum, IEEE Trans. Inform. Theory, May 1974
 - Type III: S. Kallel, IEEE Trans. Communications, Jul. 1995

Contrôle des erreurs Classification

- Classification des algorithmes de contrôle des erreurs :



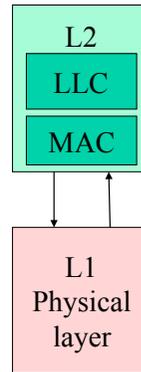
Contrôle des erreurs FEC



- FEC = *Forward Error Correction* = l'émetteur ajoute des informations redondantes au message utile et le récepteur exploite cette redondance pour décider quel message a été envoyé.
- FEC agit au niveau physique (« modem »).

Exemples :

- Codes en bloc linéaires (à parité, cycliques, Reed Solomon,...),
- Codes convolutionnels,
- Codes concaténés,
- Turbo codes,
- LDPC (Low Density Parity Check Codes),
- ...

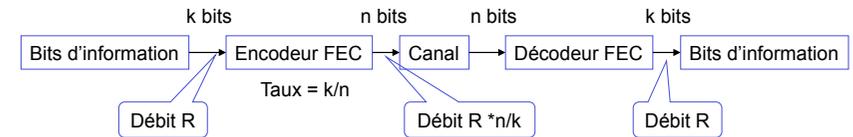


Contrôle des erreurs FEC



Caractéristiques [1] :

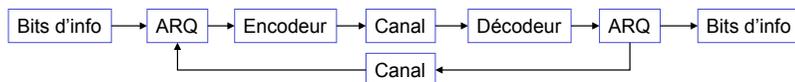
- Il n'y a pas besoin de canal de retour,
- Les besoins en mémoire sont assez réduits (ils dépendent essentiellement de la profondeur de l'entrelacement temporel),
- Il y a un compromis entre les performances et la bande utilisée,
- Le débit maximal est constant et ne dépend que du taux de codage.



Contrôle des erreurs ARQ



- ARQ = algorithme de reprises sur erreurs ; utilisation d'un protocole de fiabilisation.
- ARQ agit soit au niveau 2 (protocoles type HDLC), soit au niveau 4 (TCP).
- Contrôle d'erreur ≠ contrôle de flux.
- Le protocole ARQ est associé à un code de détection des erreurs.
- L'émetteur joint à k bits d'information, (n-k) bits de redondance.
- Si le récepteur détecte des erreurs, le paquet est jeté et une retransmission est demandée à l'émetteur. Sinon, le paquet est transmis aux couches supérieures.
- Le récepteur informe l'émetteur par un mécanisme d'acquittement.
- ARQ hybride: on combine les répétitions avec un code correcteur d'erreurs.



Contrôle des erreurs ARQ



Les performances d'un algorithme d'ARQ dépendent [1] :

- De la fiabilité du canal aller,
- De la fiabilité du canal retour,
- De la probabilité de non détection des erreurs,
- Du temps d'aller-retour entre l'émetteur et le récepteur.

Critères de performance [1] :

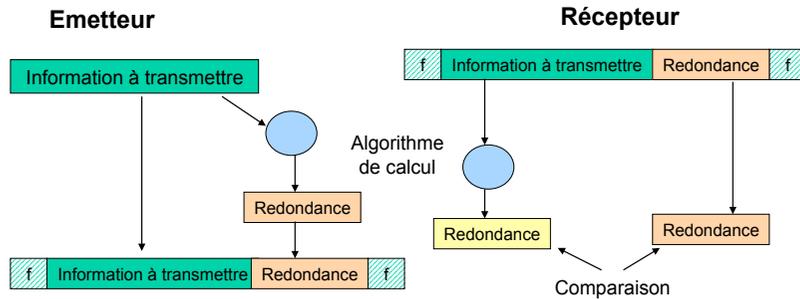
- Le débit du système :

$$\eta = \frac{\text{Nombre de bits utiles par message}}{\text{Temps moyen pour envoyer un message}}$$

- Le délai moyen nécessaire à la transmission d'un message.
- La probabilité de livrer aux couches supérieures une trame erronée.

Détection des erreurs

- Tout algorithme d'ARQ nécessite un code détecteur d'erreurs.



[4]

Détection des erreurs

- Contrôle de parité simple :
 - On joint à une chaîne de bits (mot), la somme modulo 2 des bits de la chaîne.
 - Si la somme est paire, on ajoute 0,
 - Si la somme est impaire, on ajoute 1,
 - Exemple : 100111011101 0
- Propriétés :
 - Le code est capable de détecter une erreur quelque soit la longueur de la chaîne de bits.
 - Le code **peut détecter tout nombre impair d'erreurs** mais tout nombre pair d'erreurs n'est pas détecté.
 - Les performances sont faibles car les erreurs ont généralement tendance à apparaître de manière sporadique (**trains d'erreurs**). Il y a autant de chances qu'un train contienne un nombre pair ou impair de bits erronés.
→ le code ne détecte les erreurs qu'une fois sur deux.

Détection des erreurs

- Contrôle de parité horizontal et vertical :
 - Un bit de parité est ajouté pour chaque mot et pour chaque ensemble de bits de même rang.
 - Exemple [2] :

1001101010	1	1001101010	1
0101010100	0	0101010100	0
1010101101	0	1010101101	0
0101111010	0	0101111010	0
0011111001	1	0011111001	1

Erreurs non détectables

- Propriétés [2] :
 - S'il y a un nombre pair d'erreurs confinées dans une ligne, on peut les détecter grâce aux bits de parité verticaux.
 - Si quatre erreurs forment un carré, elles ne peuvent être détectées.

Détection des erreurs

- Contrôle par redondance cyclique (Cyclic Redundancy Check) : (principaux résultats [2-6])
- Un code cyclique est un code en blocs linéaire tel que :

$$c^{(0)} = [c_0, \dots, c_{n-1}] \in C \Leftrightarrow c^{(1)} = [c_{n-1}, c_0, \dots, c_{n-2}] \in C.$$

En notation polynomiale :

$$c^{(0)}(x) \in C \Leftrightarrow c^{(1)}(x) = xc^{(0)}(x) \bmod (x^n - 1) \in C.$$
- Propriété : les mots de code d'un code cyclique (k,n) sont multiples d'un polynôme générateur g(x) de degré r=n-k diviseur de (x^n-1) :

$$g(x)h(x) = x^n - 1 \text{ et } c(x) \in C \Leftrightarrow \exists a(x) \text{ tel que } c(x) = g(x)a(x).$$

a(x) est le polynôme message et deg(a) < k.
h(x) est le polynôme de contrôle de parité.

Détection des erreurs



- Propriété : pour un mot de code, $c(x)h(x) = a(x)g(x)h(x) = a(x)(x^n-1) = 0$.
 $y(x)$ est un mot de code ssi $y(x)h(x) = 0 \pmod{(x^n-1)}$
- Syndrome correspondant aux données reçues $y(x)$: $s(x) = y(x)h(x)$.
- Encodage non systématique :
on veut transmettre $a(x)$,
le mot de code correspondant est $c(x) = a(x)g(x)$.
- Remarques :
 1. A chaque message correspond un unique mot de code (et vice-versa),
 2. Comme $\deg(a) < k$, il y a 2^k mots de codes.

Détection des erreurs



- Décodage :
On reçoit $y(x) = c(x) + e(x)$, où $e(x)$ est l'erreur introduite par le canal.
Remarque : $\deg(e) < n$.
On calcule le syndrome :
 $s(x) = y(x)h(x) = e(x)h(x)$.
Si $s(x) \neq 0$, alors il y a au moins une erreur.
Si $s(x) = 0$, on décide qu'il n'y a pas d'erreur.
Remarque : on se trompe si $e(x)$ est divisible par $g(x)$, i.e. si e est un mot du code.

Détection des erreurs



- Encodage systématique :
On considère un bloc a de $k=n-r$ bits d'information :
 $a(x) = a_0 + a_1x + \dots + a_{k-1}x^{k-1}$.
On calcule le reste de la division de $x^r a(x)$ par $g(x)$:
 $x^r a(x) = g(x)q(x) + d(x)$.
Remarques : $\deg(d) < \deg(g) = r$ et $\deg(gq(x)) = \deg(x^r a(x)) < r+k = n$.
On transmet le mot de code :
 $c(x) = g(x)q(x) = x^r a(x) - d(x)$.

Bits d'information (k)

Bits de redondance (n-k)

Détection des erreurs



- Exemple [3] :
On désire encoder la séquence : 101011
Elle est représentée par le polynôme : $a(x) = 1 + x^2 + x^4 + x^5$.
Soit $g(x)$ le polynôme générateur de CRC-16 : $g(x) = 1 + x^2 + x^{15} + x^{16}$.
Encodage systématique :
 $x^{n-k}a(x) = x^{21} + x^{20} + x^{18} + x^{16} = g(x)q(x) + d(x)$
 $d(x) = x + x^2 + x^3 + x^4 + x^5 + x^7$.
 $q(x) = x + x^2 + x^5$.
Les bits de redondance sont donc : 011111010000000.
La séquence transmise est : 011111010000000101011

Détection des erreurs



- **Performances** : les performances d'un protocole ARQ dépendent des capacités du code à détecter correctement les erreurs.

Critère de performance : probabilité de ne pas détecter une erreur.

Une borne intuitive est 2^{k-n} [7] : si la probabilité d'erreur bit est $\varepsilon=1/2$, les 2^n n-uplets sont susceptibles d'être reçus avec la même probabilité 2^{-n} . Parmi ces n-uplets, 2^k-1 sont reconnus comme étant des mots de codes et pourtant différents du mot de code émis.

$$P_e(1/2) = (2^k-1)2^{-n} = 2^{k-n} - 2^{-n} < 2^{k-n}.$$

Cette borne n'est vérifiée pour tout ε que pour certaines classes de codes.

Exemple de borne pour certains codes linéaires [7] :

$$P_e(\varepsilon) \leq [1 - (1 - \varepsilon)^k] 2^{-(n-k)}$$

Algorithmes d'ARQ

Introduction



- Pour faire un protocole ARQ
 - nécessité d'initialiser la procédure
 - nécessité de numéroter les PDU (*protocol data unit*) SN ou *sequence number* ou BSN pour "block SN"
 - nécessité de messages de contrôle (acquiescement positif ou acquiescement négatif) ACK ou NACK (*acknowledgement*)
 - Pour améliorer les performances (débit utile) ... nécessité d'anticiper
 - Pour ne pas attendre un délai "aller-retour" (*round trip delay*)
 - Pour transmettre moins de messages de contrôle (ACK)
- Remarque :
 - Dans les protocoles ARQ conventionnels (LLC, *logical link control*) : un DU (*Data Unit*) mal reçue = une DU perdue. (on fait comme si on ne l'avait jamais reçue)

Détection des erreurs



- **Exemple de performance [3]** : CRC-16

$g(x) = 1 + x^2 + x^{15} + x^{16}$ est construit de tel sorte que pour $n \leq 32767$:

- Les erreurs simples, doubles, triples et en nombre impair sont détectables,
- Les salves d'erreurs de longueur au plus 16 sont détectables (NB : longueur d'une salve = nombre de bits entre la première et la dernière erreur, bits erronés compris),
- 99.997% des salves de longueurs 17 et 18 sont détectables.

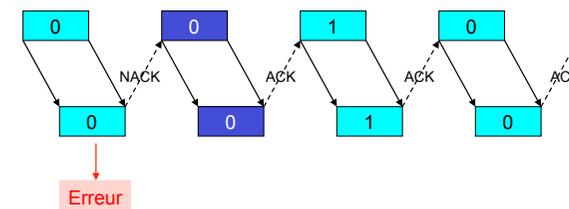
Conclusion : avec assez peu de bits de redondance, on obtient de très bonnes performances en termes de détection d'erreurs.

Algorithmes d'ARQ

Stop and Wait



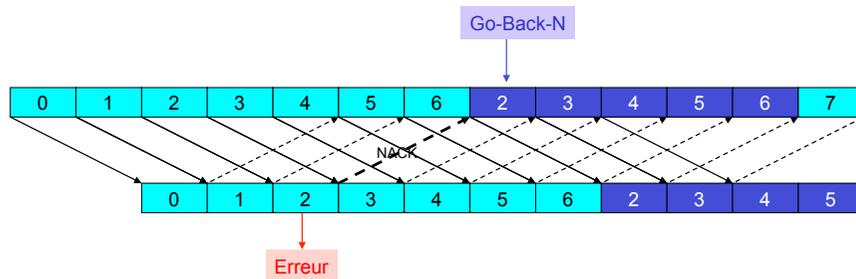
- Exemple le plus simple : le "Stop and Wait"
 - Nécessité de numéroter les DU (*Data Unit*) modulo M pour S&W, on peut prendre $M = 2$



Algorithmes d'ARQ Go Back N



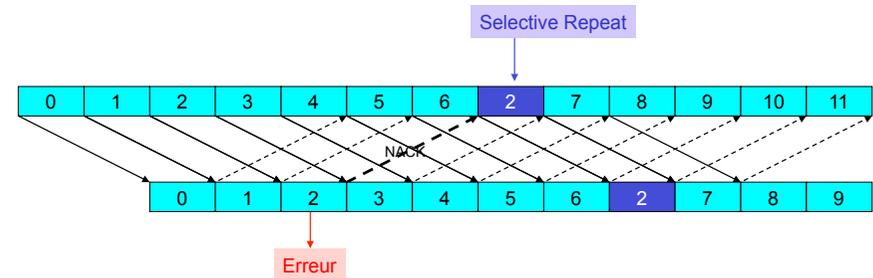
- Exemple d'anticipation : le Go-Back-N
 - Nécessité de numéroter les DU (*Data Unit*) modulo M
 - Go-Back-N pour une fenêtre d'anticipation de taille $W_s = M-1$



Algorithmes d'ARQ Selective Repeat



- Exemple d'anticipation : le Selective Repeat
 - Nécessité de numéroter les DU (*Data Unit*) modulo M
 - SR (*Selective Repeat*), pour une fenêtre de taille $W_s = M/2$



Algorithmes d'ARQ Remarques



Remarques sur la gestion des files d'attente [8] :

- La conception d'un protocole ARQ dépend de la **mémoire** disponible chez l'émetteur et chez le récepteur.
- L'**émetteur** doit mémoriser des paquets (B_1) pour une possible retransmission.
- Le **récepteur** doit mémoriser les paquets (B_2) correctement reçus mais qui ne peuvent pas être livrés en séquence.
- Si $B_1=1$, l'émetteur ne peut mémoriser qu'un paquet → **Stop and Wait**.
- Si $B_1=N$ et $B_2=0$, l'émetteur peut émettre N paquets sans recevoir d'acquittement ; en cas d'erreur il pourra réémettre les N paquets. En cas d'erreur, le récepteur n'est pas capable de livrer les paquets en séquence, sans demander la retransmission de tous les paquets à partir de l'erreur → **Go-Back-N**.
- Si $B_2>0$, le récepteur est capable en cas d'erreur de conserver des paquets correctement reçus mais non en séquence → **Selective Repeat**.

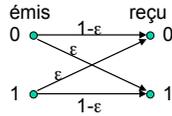
Algorithmes d'ARQ Performances



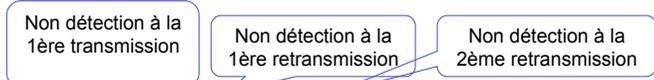
Quelques définitions :

- $P(E)$ = probabilité de livrer une trame erronée (erreurs non détectées),
- P_c = probabilité qu'une trame de n bits ne contienne pas d'erreur.
 - P_c dépend du canal.
- P_d = probabilité qu'une trame contienne des erreurs détectables,
- P_e = probabilité qu'une trame contienne des erreurs indétectables.
 - P_d et P_e dépendent du canal et du pouvoir correcteur du code utilisé,
 - $P_c + P_d + P_e = 1$.
- ε = taux d'erreur bit (*BER*).
- R = débit binaire du canal.

- Canal symétrique binaire (*Binary Symmetric Channel BSC*) :



Probabilité de livraison erronée $P(E)$:



$$\begin{aligned}
 P(E) &= P_e + P_d P_e + P_d^2 P_e + \dots \\
 &= P_e (1 + P_d + P_d^2 + \dots) \\
 &= \frac{P_e}{1 - P_d}
 \end{aligned}$$

$$P(E) = \frac{P_e}{P_c + P_e}$$

On fait l'hypothèse que le canal de retour est sans erreur.
 P_e est généralement très petit devant P_c .

Probabilités de trame correcte et erronée :

On fait ici l'hypothèse que les erreurs sur les bits sont indépendantes.

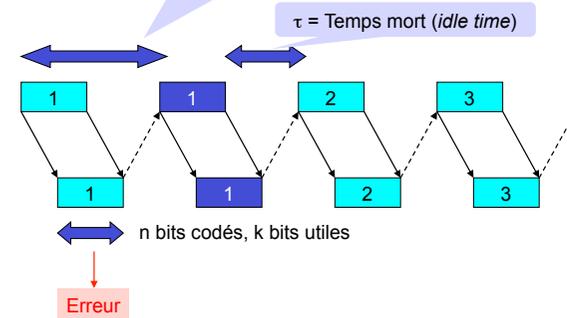
$$P_c = (1 - \varepsilon)^n$$

On dispose d'une borne sur la probabilité d'erreur indétectée de certains codes cycliques :

$$P_e \leq [1 - (1 - \varepsilon)^k] 2^{-(n-k)}$$

Exemple : code BCH(2047,2014) avec $\varepsilon = 10^{-3} \rightarrow P_c = 1.29.10^{-1}$, $P_e \leq 10^{-10}$ et $P(E) \leq 8.10^{-10}$.

- Stop & Wait : T_{AR} = Temps d'aller-retour



- Délai nécessaire à la transmission de k bits utiles s'il n'y a pas de retransmission (probabilité P_c) :

$$\delta_1 = T_{AR} = \frac{n}{R} + \tau$$

- **Stop & Wait :**
- Délai nécessaire à la transmission de k bits utiles s'il y a une retransmission (probabilité $(1-P_c)P_c$) :

$$\delta_2 = 2T_{AR} = 2\left(\frac{n}{R} + \tau\right)$$

- Délai nécessaire à la transmission de k bits utiles s'il y a j transmissions (probabilité $(1-P_c)^{j-1}P_c$) :

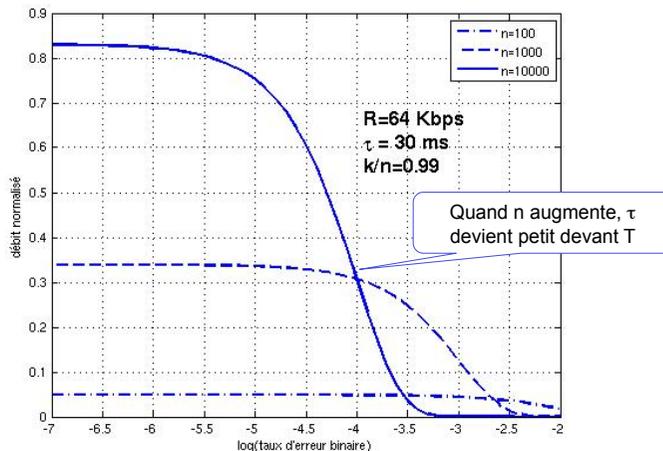
$$\delta_j = jT_{AR} = j\left(\frac{n}{R} + \tau\right)$$

- Délai moyen nécessaire à la transmission de k bits utiles :

$$\delta_{SW} = \sum_j \delta_j P_c (1-P_c)^{j-1} = \sum_j j \left(\frac{n}{R} + \tau\right) P_c (1-P_c)^{j-1}$$

$$\delta_{SW} = \frac{n/R + \tau}{P_c}$$

- **Stop & Wait :** influence de n.



- **Stop & Wait :**
- Débit de Stop & Wait :

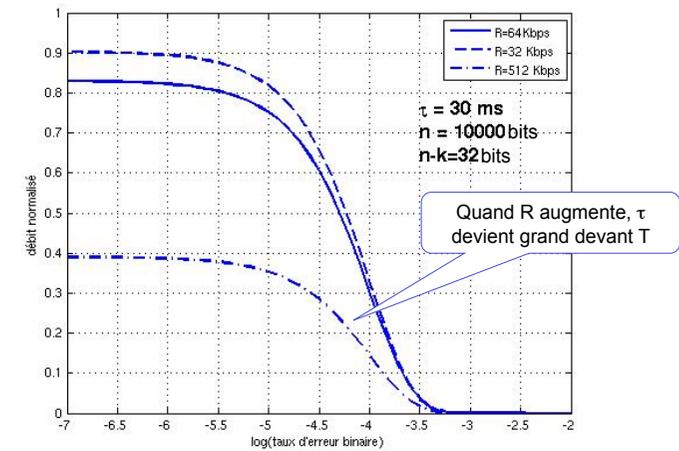
$$\eta_{SW} = \frac{k}{\delta}$$

$$\eta_{SW} = \frac{P_c(k/n)}{1 + \tau R/n} R$$

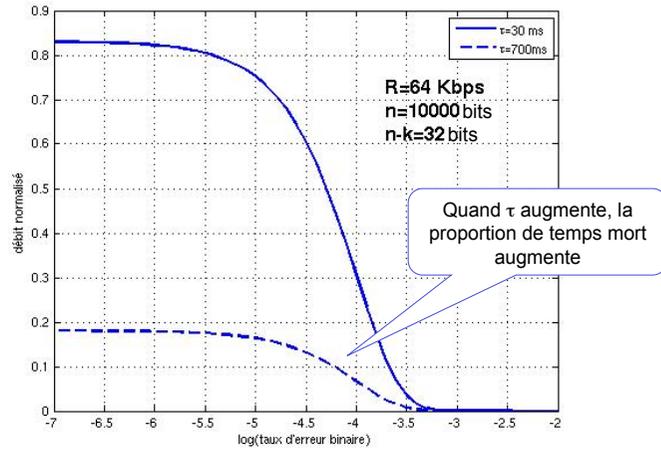
Taux de codage

- Haut débit et temps d'aller-retour élevé : $(1+\tau R/n)$ est grand et le débit de Stop & Wait est faible.
- Exemple : $R=1 \text{ Mbps}$, $\tau=700\text{ms}$ (liaison satellite) $> \tau R = 7 \cdot 10^5$. Pour que $\tau R/n$ soit proche de 1, il faudrait $n \sim 10^6$, ce qui n'est pas possible.
- Si $n=10\,000$ bits, $1 + \tau R/n = 70$ et le débit est négligeable.

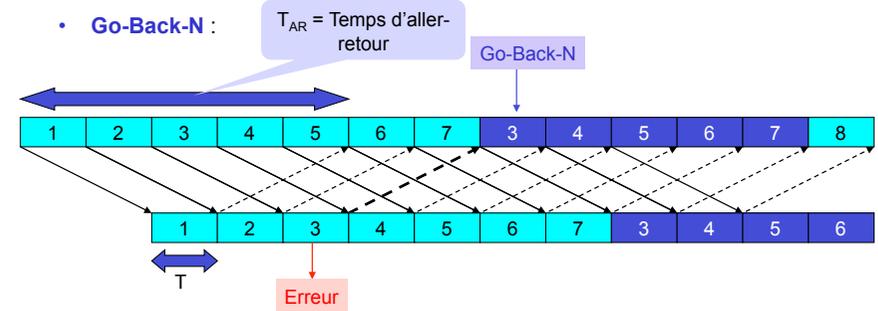
- **Stop & Wait :** influence de R.



- **Stop & Wait** : influence de τ .



- **Go-Back-N** :



- On suppose que la fenêtre est dimensionnée de manière idéale : $W \cdot T = T_{AR}$.
- Sur cet exemple : $W = 5$.

- **Go-Back-N** :

- Délai nécessaire à la transmission de k bits utiles s'il n'y a pas de retransmission (probabilité P_c) :

$$\delta_1 = T$$

- Délai nécessaire à la transmission de k bits utiles s'il y a une retransmission (probabilité $(1-P_c)P_c$) :

$$\delta_2 = (W + 1)T$$

- Délai nécessaire à la transmission de k bits utiles s'il y a j transmissions (probabilité $(1-P_c)^{j-1}P_c$) :

$$\delta_j = ((j - 1)W + 1)T$$

- **Go-Back-N** :

- Délai moyen nécessaire à la transmission de k bits utiles :

$$\begin{aligned} \delta_{GBN} &= \sum_j \delta_j P_c (1 - P_c)^{j-1} \\ &= \sum_j T((j - 1)W + 1) P_c (1 - P_c)^{j-1} \\ &= T P_c \left(\sum_{j=1}^{\infty} W(j - 1)(1 - P_c)^{j-1} + \sum_{j=1}^{\infty} (1 - P_c)^{j-1} \right) \\ &= T P_c \left(\frac{W(1 - P_c)}{P_c^2} + \frac{1}{P_c} \right) \end{aligned}$$

$$\delta_{GBN} = T \left(1 + \frac{W(1 - P_c)}{P_c} \right)$$

Algorithmes d'ARQ Performances



- **Go-Back-N** :
- Débit de Go-Back-N :

$$\eta_{GBN} = \frac{k}{\delta}$$

$$= \frac{k}{T \left(1 + \frac{W(1-P_c)}{P_c} \right)}$$

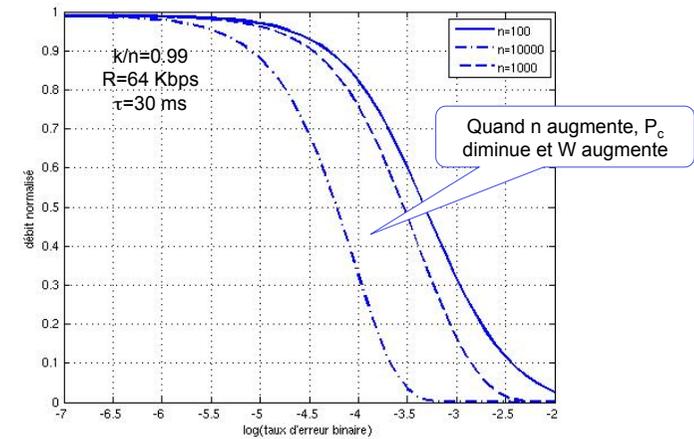
$$\eta_{GBN} = \frac{(k/n)R}{1 + \frac{W(1-P_c)}{P_c}}$$

- Si le débit est faible et le temps d'aller retour est court, la fenêtre d'anticipation, W, peut être rendue petite et Go-Back-N atteint un bon débit.
- Si au contraire, le débit est élevé et le temps de propagation relativement long, le dénominateur est grand et l'efficacité de Go-Back-N faible.
- Exemple : lien satellite à 1Mbps, n=10000 bits, k/n~1, ε=10⁻⁵, τ=700 ms, calculer l'efficacité de Go-Back-N.

Algorithmes d'ARQ Performances



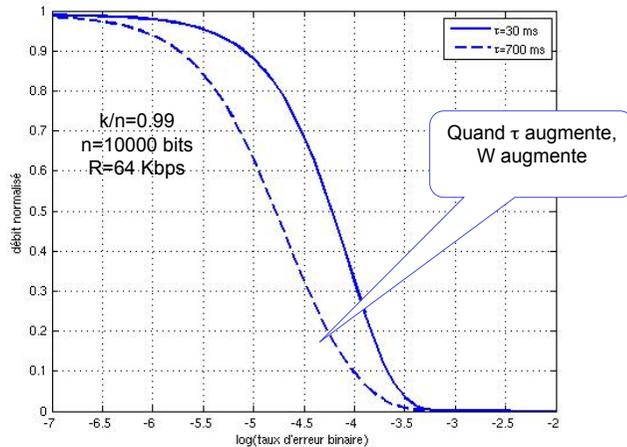
- **Go-Back-N** : influence de n.



Algorithmes d'ARQ Performances



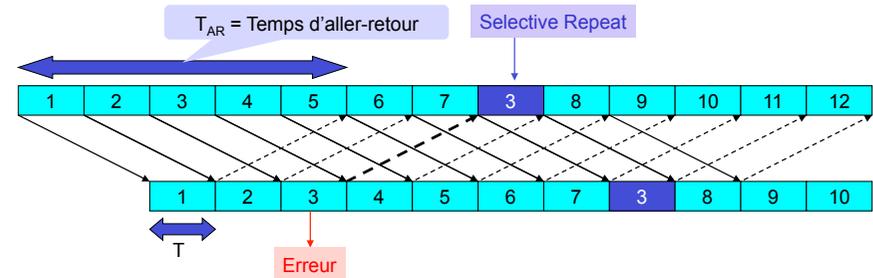
- **Go-Back-N** : influence de τ.



Algorithmes d'ARQ Performances



- **Selective Repeat** :



- On suppose un cas idéal où le récepteur est capable de mémoriser l'ensemble des paquets corrects.

• **Selective Repeat :**

- Délai nécessaire à la transmission de k bits utiles s'il n'y a pas de retransmission (probabilité P_c) :

$$\delta_1 = T$$

- Délai nécessaire à la transmission de k bits utiles s'il y a une retransmission (probabilité $(1-P_c)P_c$) :

$$\delta_2 = 2T$$

- Délai nécessaire à la transmission de k bits utiles s'il y a j transmissions (probabilité $(1-P_c)^{j-1}P_c$) :

$$\delta_j = jT$$

• **Selective Repeat :**

- Délai moyen nécessaire à la transmission de k bits utiles :

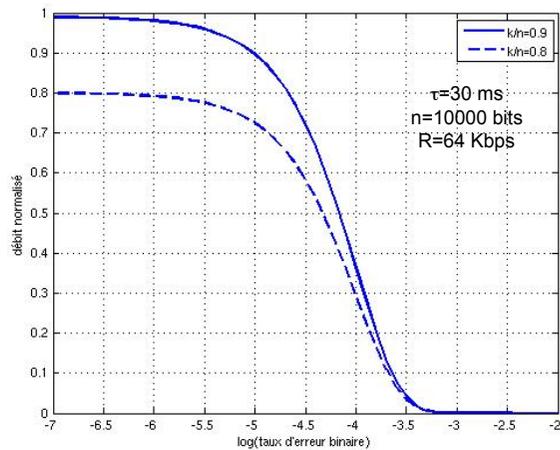
$$\begin{aligned} \delta_{SR} &= \sum_j \delta_j P_c (1 - P_c)^{j-1} \\ &= \sum_{j=1}^{\infty} j T P_c (1 - P_c)^{j-1} \end{aligned}$$

$$\delta_{SR} = \frac{T}{P_c}$$

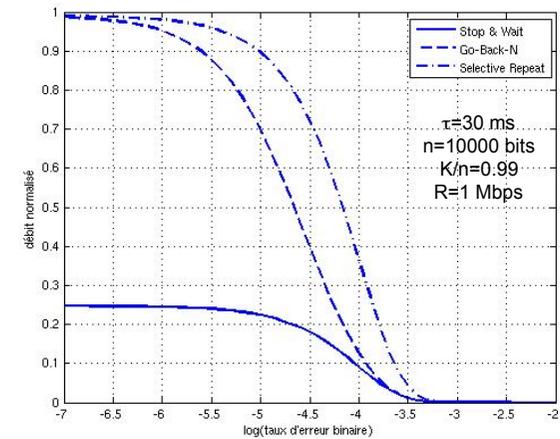
- Débit de Selective Repeat : $\eta_{SR} = \frac{k}{\delta}$
 $= \frac{k}{T/P_c}$

$$\eta_{SR} = P_c (k/n) R$$

• **Selective Repeat :** influence de k/n.



• **Selective Repeat :** comparaison avec les autres algorithmes.



Algorithmes d'ARQ hybride Type I



- **ARQ hybride type I** [1] :

- Les fonctions de correction d'erreur et de répétition restent bien séparées.
- Si le récepteur détecte des erreurs, il tente de les corriger avant de demander une retransmission si nécessaire.
- L'H-ARQ I demande généralement plus de surcharge protocolaire (*overhead*) que l'ARQ simple.
- Si le taux d'erreur introduit par le canal est faible, l'ARQ simple permet d'atteindre des débits plus importants (surcharge moindre).
- Si le taux d'erreur est important, l'H-ARQ I réduit le nombre de retransmissions.
- → adapté aux canaux avec niveau de bruit et interférence à peu près constants.

Algorithmes d'ARQ hybride Type II



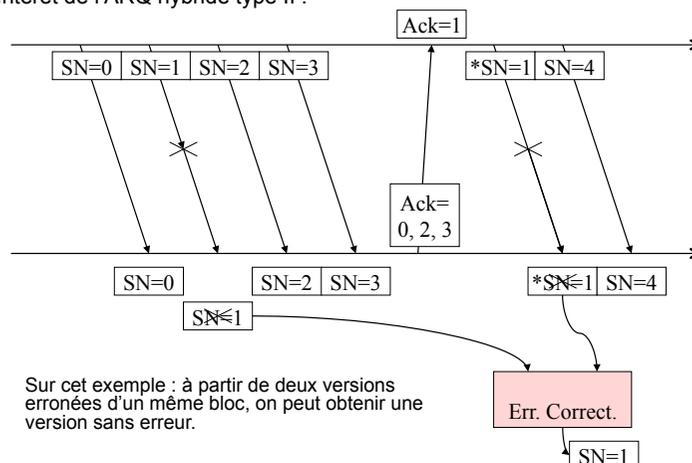
- **ARQ hybride type II** [1] :

- S'il n'est pas capable de corriger les erreurs d'un paquet, le récepteur en conserve tout de même une copie et demande une retransmission.
- Les informations contenues dans les paquets erronés mémorisés améliorent la probabilité de décodage.
- Si le paquet est retransmis de manière identique → **Chase Combining**.
- Si le schéma de codage est modifié → **Redondance incrémentale**.
- → adapté aux canaux à évanouissements rapides.

Algorithmes d'ARQ hybride Type II



- Intérêt de l'ARQ hybride type II :



- Sur cet exemple : à partir de deux versions erronées d'un même bloc, on peut obtenir une version sans erreur.

Algorithmes d'ARQ hybride Type II



- **Principe de Chase Combining** :

- Si le taux de codage du code FEC initial est R, après L transmissions le taux de codage est R/L.
- Soit Y_i le $i^{\text{ème}}$ paquet reçu : $Y_i = (y_{i,1} y_{i,2} \dots y_{i,N})$
- Le paquet combiné après la seconde transmission s'écrit :

$$Y = (y_{1,1} y_{2,1} \dots y_{1,N} y_{2,N})$$

On a donc la superposition du code initial et d'un code à répétition. On décode en utilisant le principe de maximum de vraisemblance et l'algorithme de Viterbi.

- Maximum de vraisemblance : on cherche le symbole s_m tel que la densité de probabilité $p(r|s_m)$ soit maximale (r est le symbole reçu).

Algorithmes d'ARQ hybride Type II

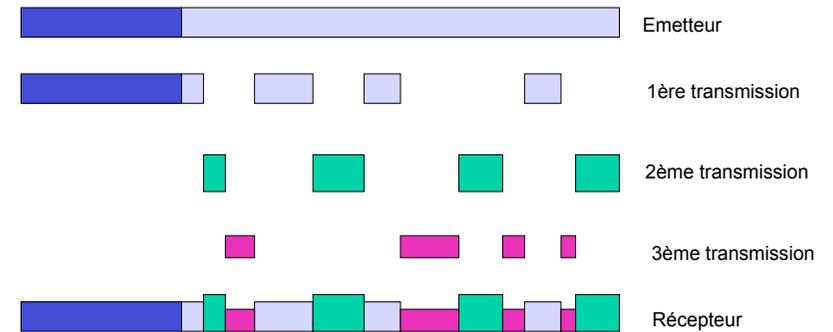


- Principe de la redondance incrémentale [10] :
 - L'information est codée avec un code de faible taux.
 - L'information ainsi qu'un certain nombre de bits de parité sont transmis.
 - Si une transmission échoue :
 - L'émetteur envoie d'autres bits de parité.
 - Le récepteur associe aux nouveaux bits ceux déjà reçus.
 - Chaque retransmission produit un mot d'un code de taux plus faible.
 - Une famille de codes peut être obtenu en poinçonnant un code de faible taux.

Algorithmes d'ARQ hybride Type II



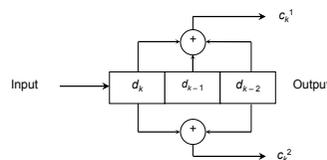
- Exemple de redondance incrémentale *Full IR* [10] :



Algorithmes d'ARQ hybride Type II



- Poinçonnage : retirer périodiquement des bits de la sortie de l'encodeur.
 - Permet d'augmenter le taux du code,
 - Et de décroître la complexité de décodage.
- Exemple :
 - Soit un code de taux $R_1=1/2$:
 $C_1 = (c_0^{(1)}, c_0^{(2)}, c_1^{(1)}, c_1^{(2)}, \dots)$
 - On obtient un code de taux $R_2=2/3$ en poinçonnant un bit sur 4 :
 $C_2 = (c_0^{(1)}, c_0^{(2)}, c_1^{(1)}, -, c_2^{(1)}, c_2^{(2)}, c_3^{(1)}, -, \dots)$
 - On obtient un code de taux $R_3=4/5$ en poinçonnant 3 bits sur 8.

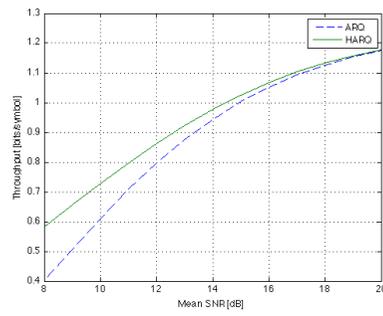
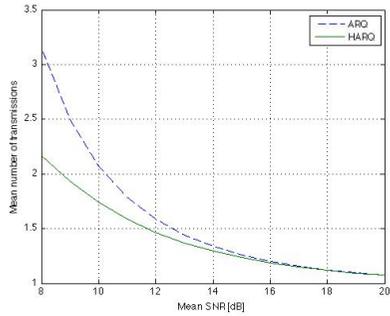


Exemple de code convolutionnel de taux $R=1/2$

Algorithmes d'ARQ hybride Type II

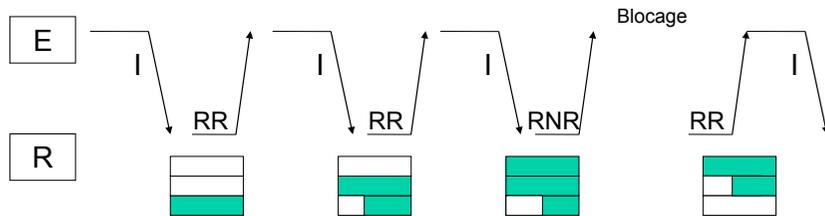


- Rate Compatible Punctured Convolutional Codes (RCPC) :
 - Une famille de codes $\{C_n\}$ est obtenue en poinçonnant un nombre croissant de bits, $R_n \leq R_{n+1}$.
 - Les bits poinçonnés pour obtenir C_n sont inclus dans l'ensemble des bits poinçonnés pour obtenir C_{n+1} .
- Exemple :
 - 0 bit poinçonné : $C_1 = (c_0^{(1)}, c_0^{(2)}, c_1^{(1)}, c_1^{(2)}, \dots)$, $R_1=1/2$
 - 1 bit sur 8 : $C_2 = (c_0^{(1)}, c_0^{(2)}, c_1^{(1)}, c_1^{(2)}, c_2^{(1)}, -, c_3^{(1)}, c_3^{(2)}, \dots)$, $R_2=4/7$
 - 2 bits sur 8 : $C_3 = (c_0^{(1)}, c_0^{(2)}, c_1^{(1)}, c_1^{(2)}, c_2^{(1)}, -, -, c_3^{(2)}, \dots)$, $R_3=4/6$
- C_n offre une protection plus élevée contre les erreurs que C_{n+1} mais avec un débit utile moindre.



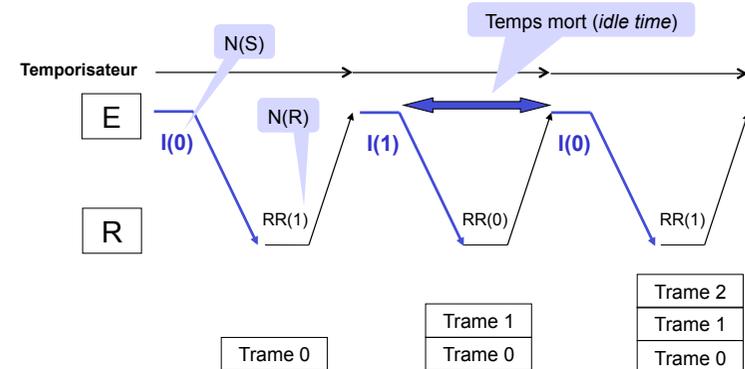
- Paramètres MCS4 de [15] : 16-QAM, taux de codage = 9/16, $R = 2.25$ bits/symbol, $g = 0.664$, $\gamma_{M} = 7,7$ dB

- A chaque réception de trame, la station R envoie
 - une trame RR si elle est prête à accepter d'autres trames,
 - une trame RNR sinon
- Après l'émission d'une trame RNR, R envoie RR dès qu'elle est de nouveau prête à recevoir une nouvelle trame.
- Les trames RR sont utilisées aussi pour l'ARQ.



- Le protocole HDLC (*High-Level Data Link Control*) est un protocole de référence pour la liaison de données (normalisé par l'ISO).
- Plusieurs variantes existent [4] :
 - HDLC (ISO) : version très générale,
 - IEEE 802.2 LLC (IEEE) : liaison de données des réseaux locaux,
 - LAP (*Link Access Procedure*) et LAP-B (ITU) : recommandation X25,
 - LAP-D (ITU) : accès au canal D du RNIS,
 - **LAP-Dm (ETSI) : pour des canaux logiques de signalisation GSM.**

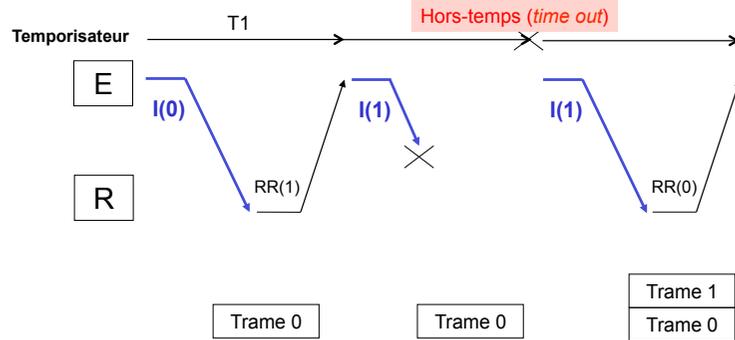
- Stop & Wait est l'algorithme le plus simple :
 - L'émetteur envoie une trame et attend un acquittement,
 - S'il reçoit un RR (*Ready to Receive*), il envoie la trame suivante,
 - Si le temporisateur expire, il réémet la trame erronée.



Le protocole LAP-B Stop&Wait



- Retransmissions sur expiration du temporisateur (T1).



M. Coupechoux – ARQ

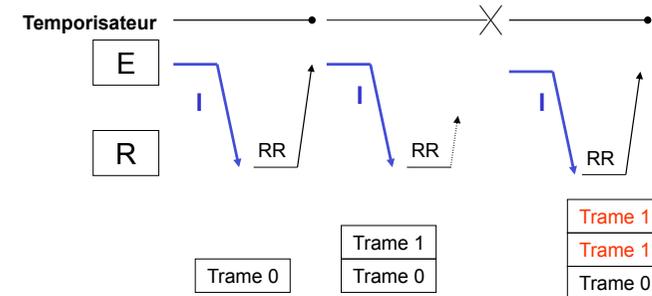
57

Le protocole LAP-B Stop&Wait



- Nécessité de numéroté les trames :

- Il est possible qu'une trame soit bien reçue mais que l'acquittement soit erroné.
- Dans ce cas, la trame sera retransmise et le récepteur aura deux copies de la même trame.
- Il est donc nécessaire de numéroté les trames pour que le récepteur puisse reconnaître deux copies de la même trame.



M. Coupechoux – ARQ

58

Le protocole LAP-B Stop&Wait



- Chaque trame d'information est numérotée.
- Le numéro est placé dans l'en-tête de la trame.
- La variable donnant le numéro de la trame est appelée N(S), codée sur un certain nombre de bits (fini).
- La numérotation se fait alors modulo M (8, 128, 32768 en LAP-B). M=2 pour Stop&Wait.
- La numérotation impose des compteurs au niveau de chaque station, pour mémoriser le numéro de la trame en cours.
- La présence de compteurs impose l'initialisation du dialogue pour que les deux stations se mettent d'accord sur les valeurs initiales des compteurs.
- On parle de **protocole orienté connexion**.

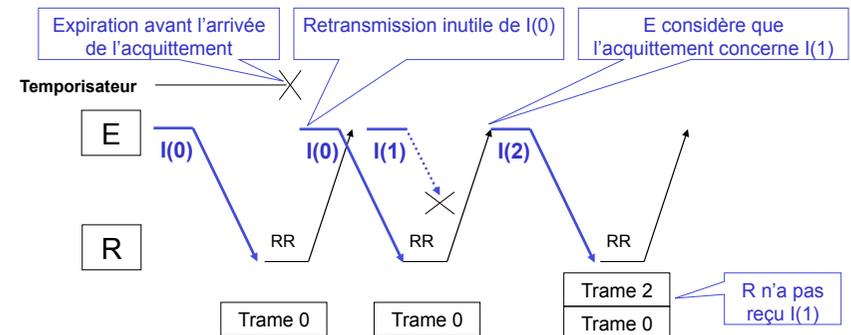
M. Coupechoux – ARQ

59

Le protocole LAP-B Stop&Wait



- Nécessité de numéroté les acquittements :



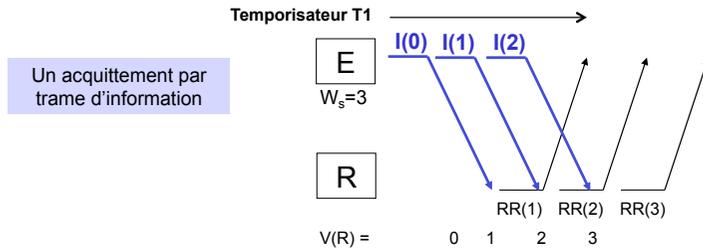
- Une mauvaise valeur du temporisateur peut entraîner un dysfonctionnement du protocole.
- Il faut préciser dans la trame d'acquittement quelle trame d'information elle acquitte.

M. Coupechoux – ARQ

60

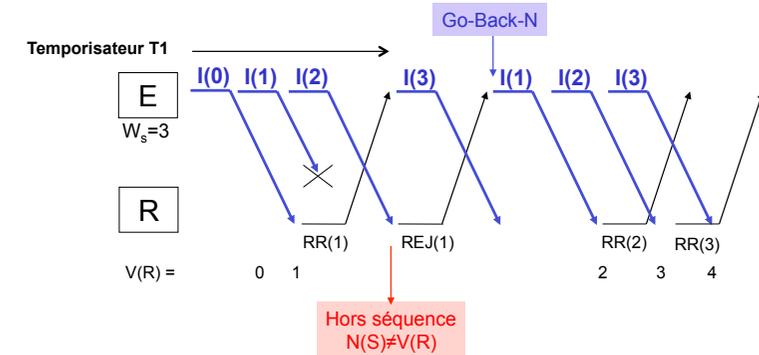
Le protocole LAP-B Go-Back-N

- Go-Back-N : l'émetteur a la possibilité d'envoyer jusqu'à W_s trames sans recevoir d'acquittement.
- Un acquittement RR n'acquies plus une seule trame mais un ensemble de trames.
- En cas d'acquittement négatif REJ (*Reject*), il retransmet l'ensemble des trames à partir de la trame demandée.
- W_s est la taille de la fenêtre d'anticipation.
- Le récepteur maintient un compteur interne $V(R)$.



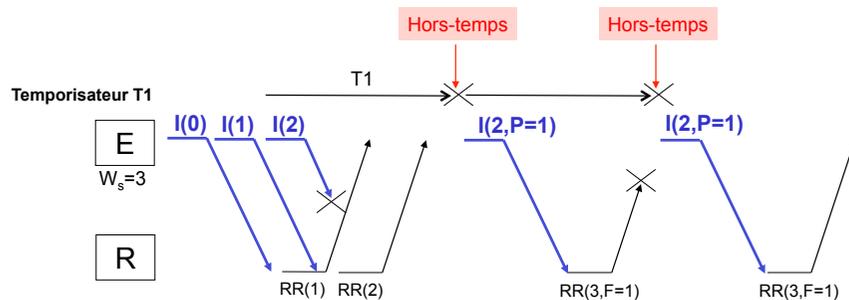
Le protocole LAP-B Go-Back-N

- Retransmission sur acquittement négatif.
 - Après l'émission d'un REJ, le récepteur ignore toutes les trames jusqu'à qu'il ait reçu la bonne (telle que $N(S)=V(R)$).



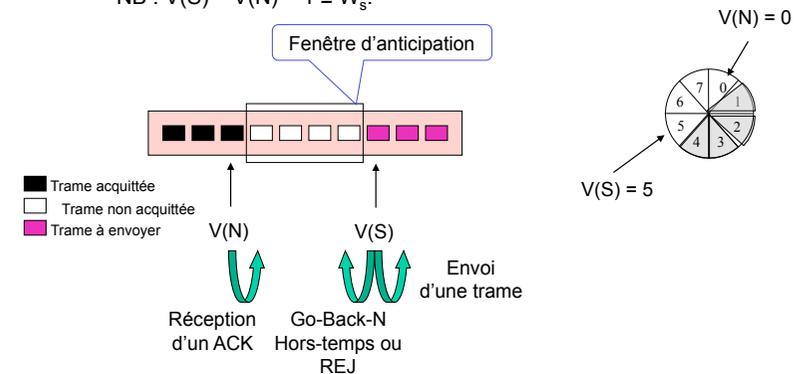
Le protocole LAP-B Go-Back-N

- Retransmission sur expiration du temporisateur (exemple).
 - Bit P/F : demande de réponse immédiate / réponse immédiate.
 - Nombre maximal de retransmissions = N2



Le protocole LAP-B Go-Back-N

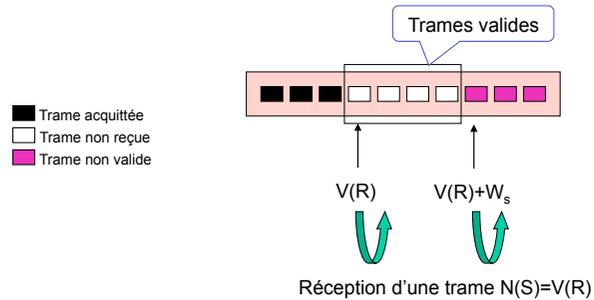
- L'émetteur maintient deux variables $V(N)$ et $V(S)$:
 - $V(N)$ = dernière trame acquittée,
 - $V(S)$ = prochaine trame à envoyer,
 - NB : $V(S) - V(N) - 1 \leq W_s$.



Le protocole LAP-B Go-Back-N



- Le récepteur maintient une variable :
 - $V(R)$ = prochaine trame attendue.

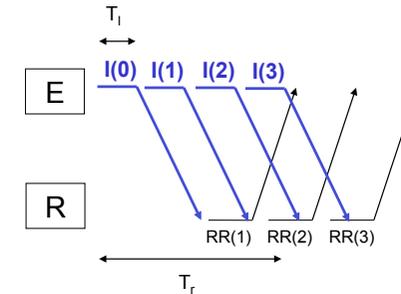


Le protocole LAP-B Go-Back-N



- Dimensionnement de la fenêtre d'anticipation : $W = \lceil T_r/T_t \rceil + 1$, où
 - T_r est le délai entre l'émission d'une trame et la réception de son acquittement,
 - T_t est la durée de transmission d'une trame.
 - On choisit l'entier immédiatement supérieur au quotient.

- Illustration :



Le protocole LAP-B Go-Back-N



- Résumé des outils mis en place :
 - $N(S)$ = numéro de la trame d'information envoyée par l'émetteur
 - $N(R)$ = numéro de la prochaine trame attendue par le récepteur
 - $V(S)$ = compteur de l'émetteur indiquant la prochaine trame à envoyer
 - $V(N)$ = compteur de l'émetteur indiquant la dernière trame acquittée
 - $V(R)$ = compteur du récepteur indiquant la prochaine trame attendue
 - $I(N(S))$ = trames d'information numérotée avec $N(S)$
 - $RR(N(R))$ = trame d'acquiescement indiquant au récepteur la prochaine trame attendue $N(R)$
 - $REJ(N(R))$ = trame de rejet indiquant au récepteur la prochaine trame attendue $N(R)$
 - W_s = taille de la fenêtre d'anticipation
 - T_1 = temporisateur de retransmission en cas de non réponse du récepteur

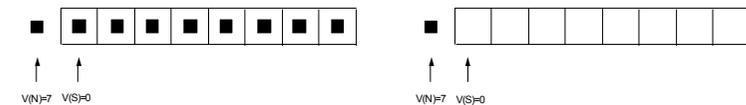
Le protocole LAP-B Go-Back-N



- Remarque sur la taille de la fenêtre : si la numérotation des trames est modulo M , la taille de la fenêtre ne peut dépasser $W_s = M - 1$.

Exemple avec $M = 8$.

- Si la fenêtre est de taille 8, il est possible d'émettre 8 trames successives sans recevoir d'acquiescement. Dans ce cas, $V(S) = 0$ et $V(N) = 7$.
- On peut retrouver les mêmes valeurs de $V(S)$ et $V(N)$ dans le cas où la trame 7 a été acquittée et que la prochaine trame à émettre est la trame 0.
- A cause du modulo, l'émetteur ne peut pas distinguer les cas où la fenêtre est vide et le cas où la fenêtre contient 8 trames.
- Pour lever l'ambiguïté, il est nécessaire de limiter la fenêtre à $N(S)_{\max} = 7$ trames dans notre cas.



Le protocole LAP-B

Format de trame



Fanion	Adresse	Contrôle	Information	FCS	Fanion
01111110	1 octet	1 octet	N octets	2 octets	01111110

- La trame est délimitée par le **fanion** de début et de fin de trame 01111110.
- Le champ **Adresse** (1 octet) identifie l'une des deux extrémités de la liaison.
- Le champ **Contrôle** (1 octet) indique le type de la trame et contient les valeurs de N(S) et N(R). En « mode étendu », ce champ est de 2 octets.
- Le champ **Information** contient les informations utilisateur. Il est facultatif et de longueur variable.
- Le **champ FCS** (Frame Check Sequence) est la séquence de contrôle qui permet la détection des erreurs. Il est obtenu par un contrôle polynomial de polynôme générateur $x^{16}+x^{12}+x^5+1$.

Le protocole LAP-B

Format de trame



- Le protocole LAP-B est utilisé sur le lien entre un terminal abonné et un équipement du réseau public de transmission de données (le commutateur d'accès au réseau).
- Par convention, le commutateur a toujours l'adresse 01.
- Le terminal abonné a toujours l'adresse 03.
- C'est le champ Adresse qui permet de distinguer une commande d'une réponse.
- Par convention, chaque station place dans une commande l'adresse du destinataire, et dans une réponse sa propre adresse.
- Ainsi, l'adresse de la trame est toujours celle de la station secondaire.

Le protocole LAP-B

Types de trames



- Il existe 3 types de trames identifiées par le champ contrôle.
 - **La trame I** : trame d'information qui permet la transmission des données utilisateur.
 - **Les trames S** : trames de supervision qui gèrent les acquittement et le contrôle de flux.
 - **Les trames U** : trames non numérotées (Unnumbered) qui ne transportent pas de données utilisateur. Elles sont utilisées pour les fonctions de contrôle de la liaison, comme l'établissement et la libération des connexions.
- Les trames sont aussi classifiées en « **commandes** » ou « **réponses** ». Une station primaire envoie des commandes à la station secondaire qui doit répondre dans un bref délai par une trame de réponse.
- Certaines trames ne peuvent être que des commandes, d'autres ne peuvent être que des réponses. Certaines trames peuvent être soit une commande, soit une réponse.

Le protocole LAP-B

Types de trames : trames d'information



- La trame I permet la transmission des données utilisateur.
- Le champ contrôle de la trame I a le format suivant

8	7	6	5	4	3	2	1
N(R)		P	N(S)		0		

Le bit numéro 1 de valeur 0 identifie la trame I.

N(S) représente le numéro de séquence de la trame I.

N(R) représente le numéro d'acquittement des trames dans le sens inverse (piggybacking)

Le bit P (Poll) signifie que la trame est une commande qui contient une interrogation à laquelle l'autre extrémité doit répondre dans un bref délai

Le protocole LAP-B

Types de trames : trames de supervision



8	7	6	5	4	3	2	1
N(R)	P/F	S	S	0	1		
N(R)	P/F	0	0	0	1		
N(R)	P/F	0	1	0	1		
N(R)	P/F	1	0	0	1		

Format général des trames S

RR : Receiver Ready

RNR : Receiver Not Ready

REJ : Reject

- **RR** : indique que la station est prête à recevoir des trames et que toutes les trames de numéro de séquence inférieur à N(R) ont été bien reçues.
- **RNR** : indique que la station n'est pas prête à recevoir des trames et que toutes les trames de numéro de séquence inférieur à N(R) ont été bien reçues.
- **REJ** : indique un acquittement négatif. Il faut reprendre la transmission à partir de la trame de numéro de séquence N(R).
- Les trames de supervision peuvent être soit des commandes, soit des réponses.
- Commande => le 5^{ème} bit est un bit P (Poll)
- Réponse => le 5^{ème} bit est un bit F (Final)

Le protocole LAP-B

Types de trames : trames non numérotées



8	7	6	5	4	3	2	1
U	U	U	P/F	U	U	1	1
0	0	1	P	1	1	1	1
0	1	0	P	0	0	1	1
0	1	1	F	0	0	1	1
1	0	0	F	0	1	1	1
0	0	0	F	1	1	1	1

Format général des trames S

SABM : Set Asynchronous Balanced Mode

DISC : Disconnect

UA : Unnumbered Acknowledgement

FRMR : Frame Reject

DM : Disconnect Mode

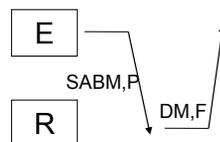
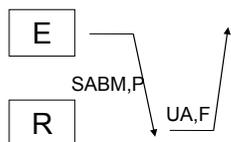
- **SABM** (Set Asynchronous Balanced Mode) : la trame d'initialisation de connexion.
- **DISC** (DISConnect) : la trame de libération de connexion
- **UA** (Unnumbered Acknowledgement) : acquitte les trames SABM ou DISC
- **FRMR** (Frame Reject) : permet de rejeter une commande invalide
- **DM** (Disconnect Mode) : indique l'état de déconnexion d'une station, utilisé pour répondre négativement à une demande de connexion.

Le protocole LAP-B

Etablissement et fermeture de connexion



- LAP-B est orienté connexion. Avant de pouvoir transférer des données utilisateur, il faut initialiser les compteurs et les temporisateurs de chaque entité en établissant une connexion.
- La station qui veut commencer le dialogue envoie une trame SABM avec le bit P mis à 1.
- L'autre station accepte la demande de connexion par l'émission d'une trame UA avec le bit F mis à 1.
- Elle peut refuser l'établissement de la connexion en émettant une trame DM avec le bit F mis à 1.

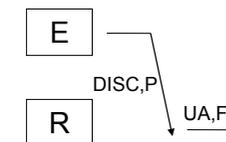


Le protocole LAP-B

Etablissement et fermeture de connexion



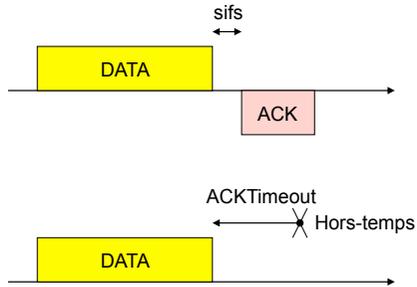
- La fermeture de la connexion se fait avec les trames DISC et UA



Autres exemples d'implémentations IEEE 802.11



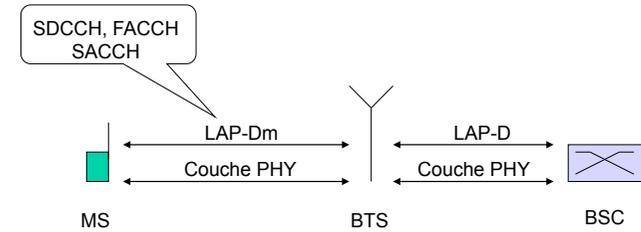
- IEEE 802.11 implémente au niveau MAC un Stop & Wait très simple.
 - L'acquittement suit immédiatement le MPDU,
 - Le MPDU est retransmis tant qu'il n'est pas acquitté,
 - Il y a un nombre maximum de retransmissions.



Autres exemples d'implémentations LAP-Dm - GSM



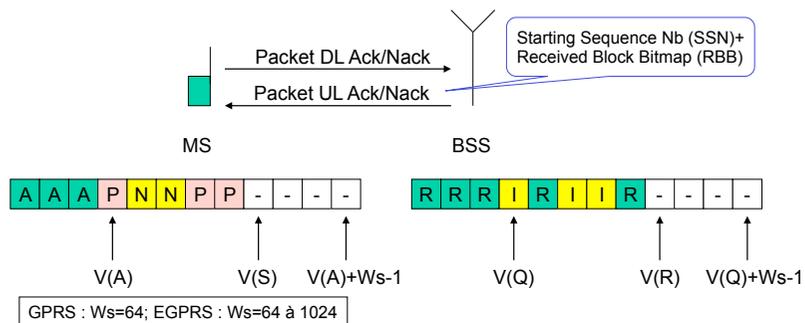
- Le LAP-Dm est utilisé pour certains canaux logiques de signalisation GSM [11] :
 - BCCH, PCH, AGCH : réduit à sa plus simple expression (transparent),
 - SDCCH, FACCH, SACCH : semblable au LAP-D.



Autres exemples d'implémentations (E)GPRS



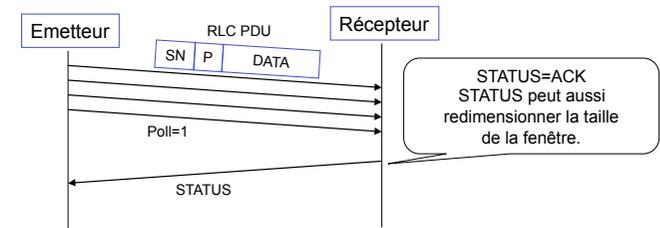
- Le RLC de (E)GPRS implémente un *selective repeat* (mode acquitté).
- En GPRS, les retransmissions sont faites en utilisant le même MCS (*Modulation and Coding Scheme*),
- En EGPRS, il est possible de re-segmenter les blocs RLC et d'utiliser un MCS plus robuste ou un schéma de poinçonnage différent.
- En EGPRS, il y a la possibilité d'activer IR.



Autres exemples d'implémentations UMTS



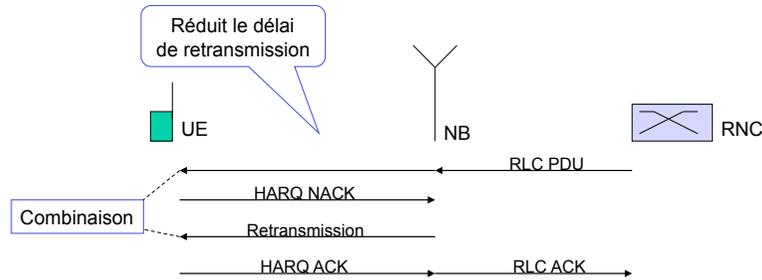
- Le RLC de l'UMTS (mode acquitté) implémente un *selective repeat* [12-14].
- Les acquittements sont demandés par l'émetteur lorsque e.g. :
 - Un temporisateur expire,
 - N PDUs ont été envoyés,
 - Un pourcentage de la fenêtre de transmission a été envoyé.
- De manière optionnelle, le récepteur peut décider l'envoi d'ACK, lorsque e.g. :
 - Un temporisateur expire,
 - Des numéros de séquence manquent.



Autres exemples d'implémentations HSDPA



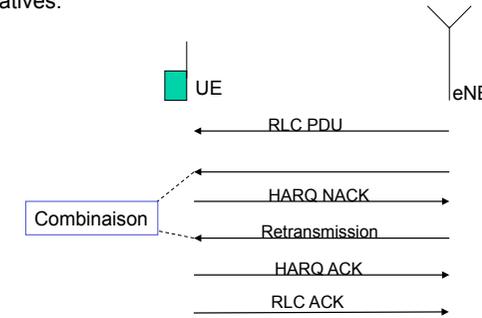
- En HSDPA, un mécanisme d'HARQ au niveau MAC (entre le Node-B et l'UE) se superpose à l'ARQ RLC (entre le RNC et l'UE).
- L'HARQ est composé de plusieurs mécanismes S&W parallèles,
- L'HARQ est de type II avec Chase Combining ou Incremental Redundancy.



Autres exemples d'implémentations LTE



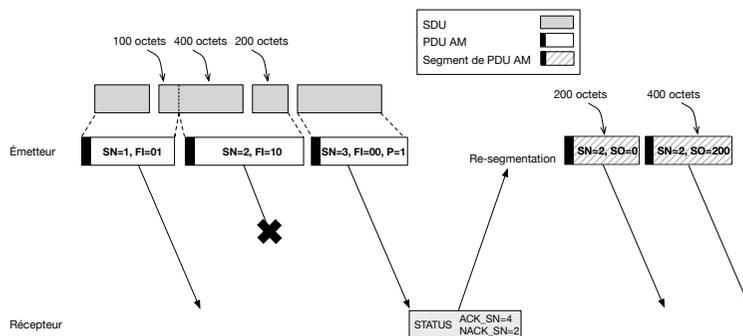
- En LTE, le RNC disparaît, de telle sorte que l'eNode-B est responsable de l'HARQ (couche MAC) et de l'ARQ (couche RLC).
- HARQ : Chase combining et IR, jusqu'à 8 processus S&W parallèles.
- Voie descendante : transmissions et retransmissions HARQ asynchrones et adaptatives.
- Voie montante : transmissions et retransmissions HARQ synchrones et adaptatives.



Autres exemples d'implémentations LTE



- Exemple de transmission ARQ avec re-segmentation



- La taille des PDU dépend des opportunités de transmission offerts par MAC au RLC
- FI=01 indique que le PDU contient un SDU entier et un segment de SDU
- Les PDU de SN<4 ont été correctement reçus à l'exception du PDU de SN=2
- SO indique la position des segments dans le PDU initial

Conclusion



- A retenir :
 - Types de contrôle d'erreur : FEC et ARQ.
 - Types d'ARQ : Simple, Hybride I et II.
 - Principes de l'ARQ simple :
 - Stop and Wait, Go-Back-N, Selective Repeat.
 - LAP-B
 - Performances avec canal BSC.

Annexe : Maximum de vraisemblance



- On note : $r_k = \begin{bmatrix} y_{1,k} \\ y_{2,k} \end{bmatrix}$
- Maximum de vraisemblance : on cherche le symbole s_m tel que la densité de probabilité $p(r|s_m)$ soit maximale.
 - r est le vecteur des symboles reçus.
 - s_m est un vecteur de symboles émis possible.
 - Exemple : $r_k = h_k s_{m,k} + n_k$,
 - $h_k = [h_{1,k} \ h_{2,k}]$ est le gain du canal.
 - $n_k \sim N(0, \sigma)$ est le bruit.
- Maximiser $p(r|s_m)$ est équivalent à
 - minimiser $\sum_k |r_k - h_k s_{m,k}|^2$
 - minimiser $\sum_k (r_k^H - h_k^H s_{m,k}^*)(r_k - h_k s_{m,k})$
 - maximiser $\sum_k \text{Re}[h_k^H s_{m,k}^* r_k] = \sum_k \text{Re}[y_{1,k} h_{1,k}^* + y_{2,k} h_{2,k}^*] s_{m,k}$
- Le décodage se fait donc en appliquant un MRC sur les symboles reçus et l'algorithme de Viterbi.

Algorithmes d'ARQ hybride Performances



• Nombre moyen de transmissions et débit moyen :

- Soit P_k la probabilité que les k premiers paquets soient erronés

$$P_k = \int_0^\infty \dots \int_0^\infty Q_1 \dots Q_k p(\gamma_1) \dots p(\gamma_k) d\gamma_1 \dots d\gamma_k$$

- $P_{k-1} - P_k$ est la probabilité que la $k^{\text{ième}}$ transmission soit réussie
- On obtient le nombre moyen de transmissions :

$$\begin{aligned} N_K &= \sum_{k=1}^K k(P_{k-1} - P_k) + K P_K \\ &= 1 + \sum_{k=1}^{K-1} P_k \end{aligned}$$

- Et le débit moyen : $T_K = \frac{R(1 - P_K)}{N_K}$

Annexe : Algorithmes d'ARQ hybride Performances



• Hypothèses [15] :

- Protocole Stop and Wait avec Chase Combining
- K transmissions au maximum sont permises
- Canal de Rayleigh par bloc (le SNR est i.i.d. de transmission en transmission):

$$p(\gamma) = \frac{1}{\bar{\gamma}} e^{-\frac{\gamma}{\bar{\gamma}}}$$

- Probabilité d'erreur paquet modélisée de la manière suivante :

$$f(\gamma) = \begin{cases} 1 & \text{if } 0 < \gamma < \gamma_M \\ a e^{-g\gamma} & \text{if } \gamma \geq \gamma_M \end{cases}$$

- La probabilité d'erreur paquet à la $k^{\text{ième}}$ transmission est :

- Pour ARQ : $Q_k = f(\gamma_k)$

- Pour HARQ : $Q_k = f(\gamma_{G,k})$, $\gamma_{G,k} = \sum_{i=1}^k \gamma_i$

Algorithmes d'ARQ hybride Performances



• Approximations

- [15] a montré que P_k pouvait s'écrire :

$$P_k = e^{-X} \left(\sum_{i=k}^{\infty} \frac{X^i}{i!} + \sum_{i=0}^{k-1} \frac{X^i}{i!} \prod_{j=1}^{k-i} \frac{1}{1 + jg\bar{\gamma}} \right)$$

- Si on suppose que K est infini, on obtient (avec $X = \gamma_M / \bar{\gamma}$)

$$N = 1 + e^{-X} \sum_{i=1}^{\infty} \frac{X^i}{(i-1)!} + \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{1}{1 + jg\bar{\gamma}}$$

$$N = 1 + X + \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{1}{1 + jg\bar{\gamma}}$$

- Et avec les approximations (avec $F = \gamma_M + 1/g$) suivantes :

$$\prod_{j=1}^i \frac{1}{1 + jg\bar{\gamma}} \approx 1/(i!(g\bar{\gamma})^i), \quad e^{1/g\bar{\gamma}} \approx 1 + 1/g\bar{\gamma}$$

$$\boxed{\begin{aligned} N &\approx 1 + \frac{F}{\bar{\gamma}} \\ T &\approx \frac{R}{1 + F/\bar{\gamma}} \end{aligned}}$$

• Cas ARQ :

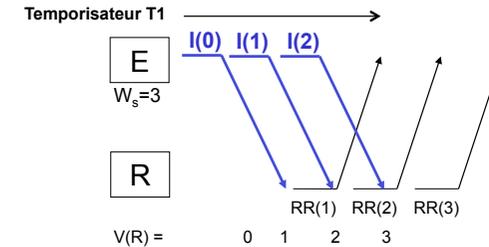
$$P_k = \left(\int_0^\infty Q_1 p(\gamma_1) d\gamma_1 \right)^k$$

$$= \left(1 - \frac{e^{-\gamma_M/\bar{\gamma}}}{1 + 1/g\bar{\gamma}} \right)^k$$

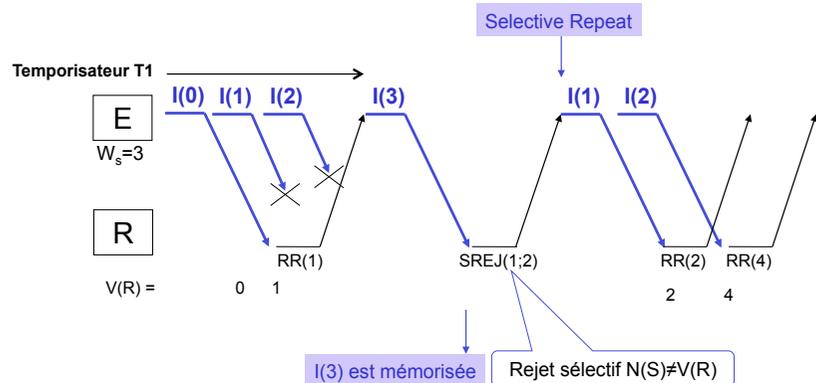
$$N = (1 + 1/g\bar{\gamma}) e^{\gamma_M/\bar{\gamma}}$$

$$T = \frac{R e^{-\gamma_M/\bar{\gamma}}}{1 + 1/g\bar{\gamma}}$$

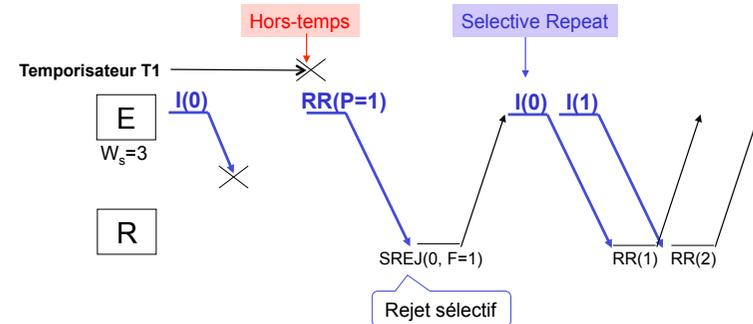
- Selective Repeat (SR) : l'émetteur ne retransmet que les trames erronées.
- Les trames sont susceptibles d'arriver dans le désordre.
- N(S) : numéro de la trame envoyée.
- N(R) : numéro de la prochaine trame attendue en séquence.
- TAB : liste des trames à retransmettre.
- SR est un protocole à fenêtre d'anticipation :



- Retransmission sur acquittement négatif :
 - SREJ (*Selective Reject*) contient un ensemble de numéros de séquence de trames à retransmettre.
 - Un seul SREJ est envoyé sans réponse.

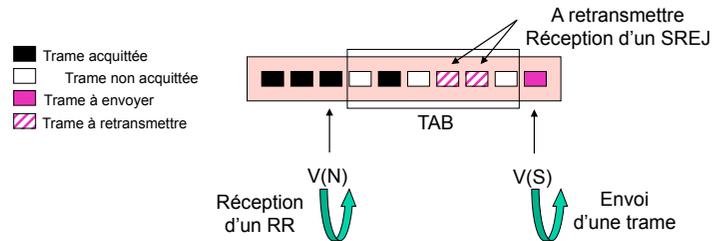


- Retransmission sur expiration du temporisateur (exemple).
 - Une trame RR demande l'état du récepteur.
 - Nombre maximal de retransmissions = N2.



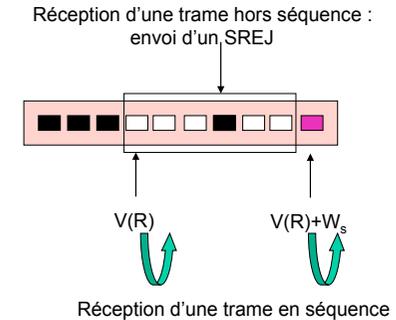
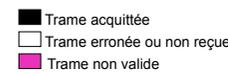
Le protocole LAP-B Selective Repeat

- L'émetteur maintient deux variables et une structure :
 - Toutes les trames avant $V(N)$ ont été acquittées,
 - $V(S)$ = prochaine trame à envoyer en séquence,
 - NB : $V(S) - V(N) - 1 \leq W$.
 - Un tableau TAB représentant les trames entre $V(N) + 1$ et $V(S) - 1$:
 - Acquittées,
 - Non acquittées,
 - A retransmettre (le temporisateur a expiré ou un SREJ a été reçu).



Le protocole LAP-B Selective Repeat

- Le récepteur maintient une variable et une structure de données :
 - Toutes les trames avant $V(R)$ ont été reçues et acquittées,
 - $V(R)$ = prochaine trame attendue en séquence,
 - Un tableau (TAB) des trames de la fenêtre de réception avec leur état :
 - Acquittée,
 - Non acquittée.

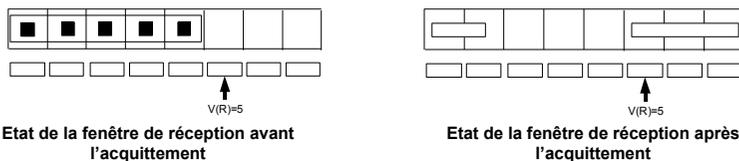


Le protocole LAP-B Selective Repeat

- Remarque sur la taille de la fenêtre : si la numérotation des trames est modulo M , la taille de la fenêtre ne peut dépasser $W_s = M/2$.

Exemple avec $M = 8$.

- On suppose qu'on peut envoyer cinq trames sans recevoir d'acquiescement préalable (trames 0 à 4).
- On suppose que toutes les trames ont été correctement reçues par le récepteur mais que les RR sont perdus.
- L'émetteur retransmet les 5 trames après expiration du temporisateur.
- Mais les trames 0 et 1 sont des trames valides appartenant à la nouvelle fenêtre de réception. Donc la retransmission de ces trames est interprétée comme 2 nouvelles trames valides.



Etat de la fenêtre de réception avant l'acquiescement

Etat de la fenêtre de réception après l'acquiescement

Glossaire

- | | |
|--|------------------------------------|
| ARQ : Automatic Repeat Request | MAC : Medium Access Control |
| BER : Bit Error Rate | MCS : Modulation and Coding Scheme |
| BSC : Binary Symmetric Channel | PDU : Protocol Data Unit |
| BSN : Block Sequence Number | REJ : Reject |
| CC : Chase Combining | RLC : Radio Link Control |
| CRC : Cyclic Redundancy Check | RR : Ready to Receive |
| DU : Data Unit | SN : Sequence Number |
| FEC : Forward Error Correction | SR : Selective Repeat |
| GBN : Go-back-N | SREJ : Selective Reject |
| HARQ : Hybrid ARQ | SW : Stop and Wait |
| HDLC : High Level Data Link Control | TCP : Transport Control Protocol |
| IR : Incremental Redundancy | |
| LAP-B : Link Access Procedure Balanced | |
| LDPC : Low Density Parity Check code | |
| LLC : Logical Link Control | |

[1] S; Lin, D. Costello et M. Miller, « Automatic-Repeat-Request Error Control Schemes », IEEE Communication Magazine, Dec. 1984.

[2] D. Bertsekas et R. Gallager, « Data Networks », Prentice Hall, 1992.

[3] T. Ramabadran et S. Gaitonde, « A Tutorial on CRC Computation », IEEE Micro, 1988.

[4] X. Lagrange et D. Seret, « Introduction aux réseaux », Hermes, 1998.

[5] A. Glavieux et M. Joindot, « Communications numériques, introduction », Masson, 1996.

[6] R. Morelos-Zaragoza, « The Art of Error Correcting Coding », Wiley, 2002.

[7] J.K. Wolf et al., « On the Probability of Undetected Error », IEEE Trans. On Communications, Feb. 1982.

[8] R. Cam et C. Leung, « Throughput Analysis of some ARQ Protocols in the Presence of Feedback Errors », IEEE Trans. On Communications, Jan. 1997.

[9] Recommandation UIT-T X.25.

[10] E. Soljanin, « Hybrid ARQ in Wireless Networks », 2003.

[11] X. Lagrange, Ph. Godlewski, S. Tabbane, « Réseaux GSM », Hermes, 2000.

[12] J. J. Alcaraz et al., « Optimizing TCP and RLC Interaction in the UMTS Radio Access Network », IEEE Com. Mag., Mar. 2006.

[13] Q. Zhang et al., « Performance of UMTS Radio Link Control », Proc. of IEEE ICC, 2002.

[14] R. Bestak, Ph. Godlewski et Ph. Martins, « Interactions bw. The TCP and RLC Protocols in UMTS », Proc. of MMT, 2002.

[15] X. Lagrange, « Performance Analysis of HARQ Protocols with Link Adaptation on Fading Channels », Springer Annals of Telecommunications, Mar. 2011.



Contexte public } sans modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après et à l'exclusion expresse de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage à destination de tout public qui comprend :

- Le droit de reproduire tout ou partie du document sur support informatique ou papier,
- Le droit de diffuser tout ou partie du document au public sur support papier ou informatique, y compris par la mise à la disposition du public sur un réseau numérique.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel, non exclusif et non transmissible.

Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : siltepedago@enst.fr

