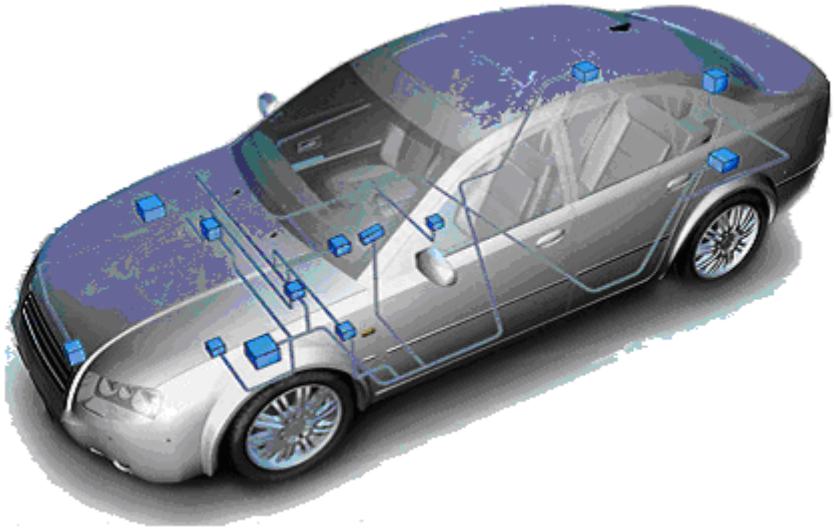
Sciences et technologies
de l'Industrie et du développement durable

SIN : Evolution et paramétrage d'un système réel

Module SIN 31 : Concevoir un système local et permettre le dialogue entre l'homme et la machine

Ressources sur : Le bus de terrain CAN



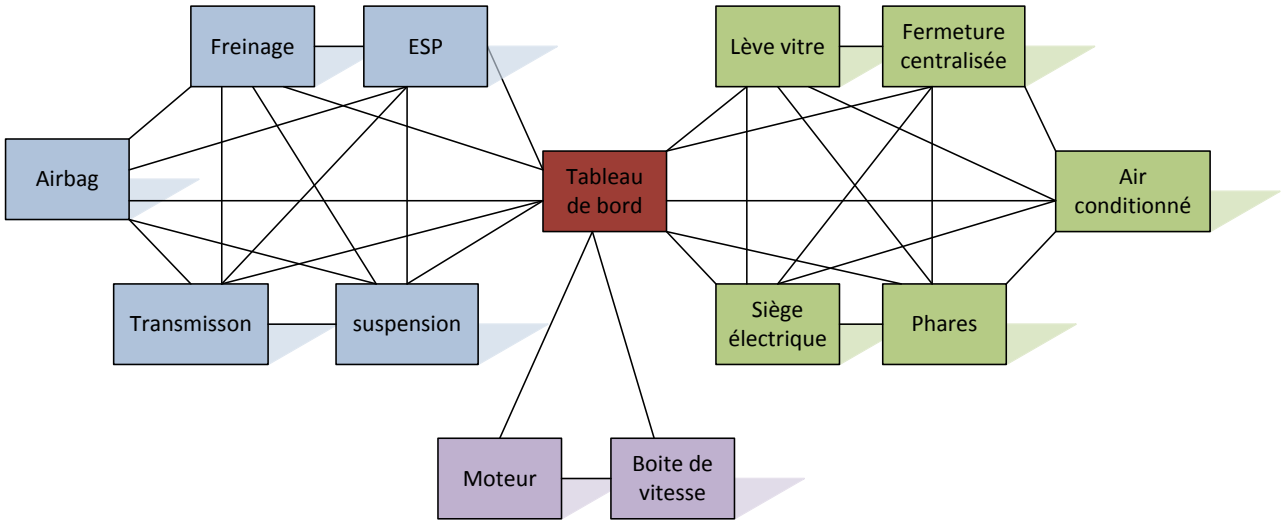
Sommaire

1. Présentation du bus CAN

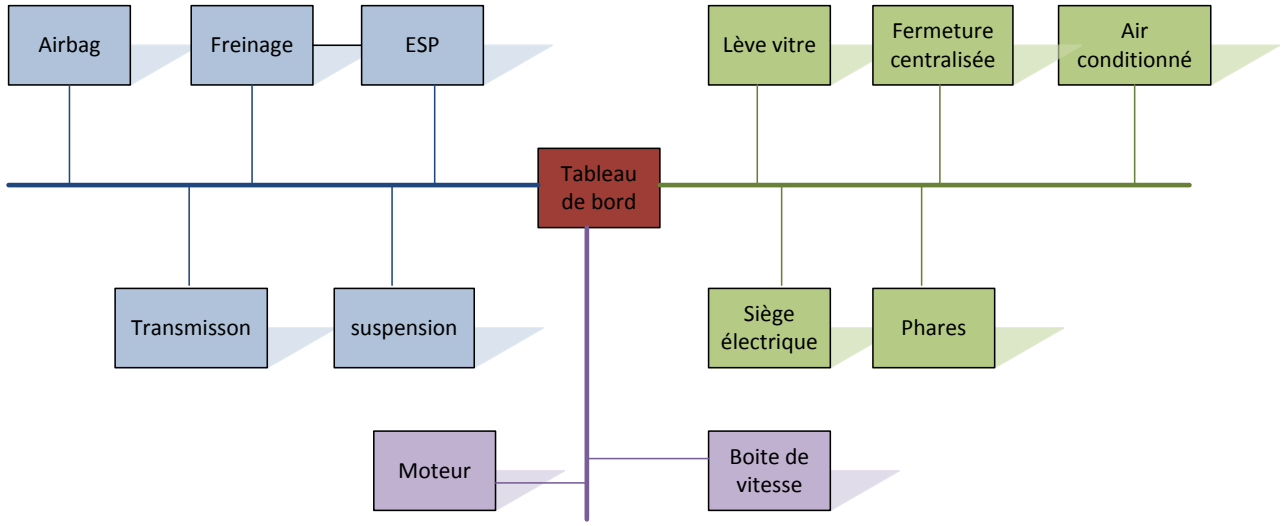
a) Origine

La sécurité et le confort des utilisateurs dans les véhicules ont entrainé une augmentation des organes électroniques.

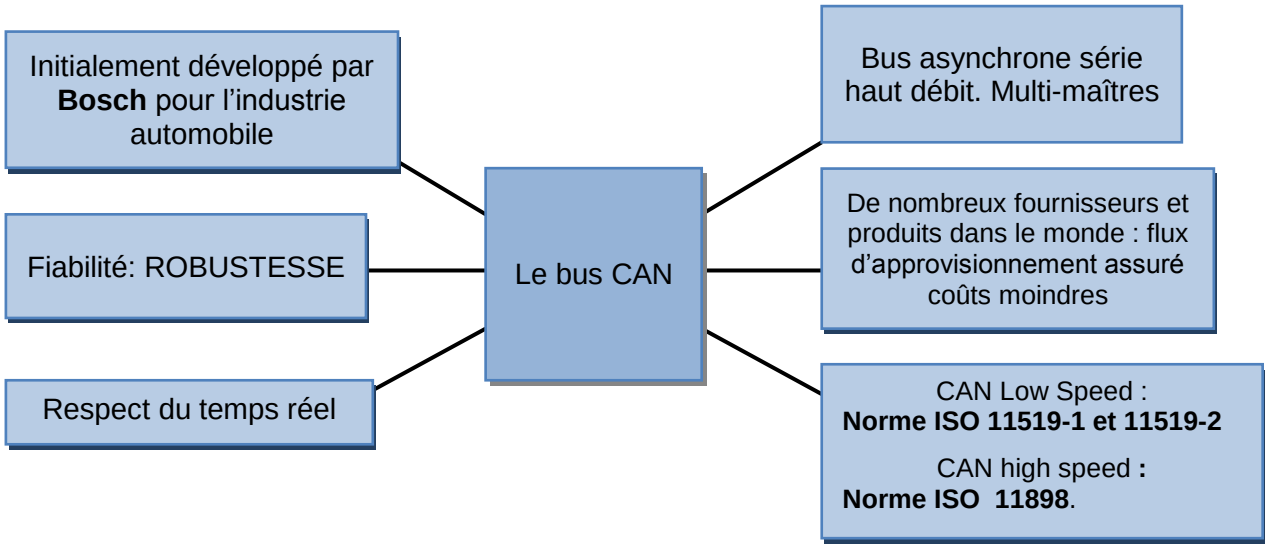
Le câblage point à point (organe à organe) est devenu très complexe et très couteux en câble. On est passé de 200m à 2000m de câble dans un véhicule entre 1960 et 2000.



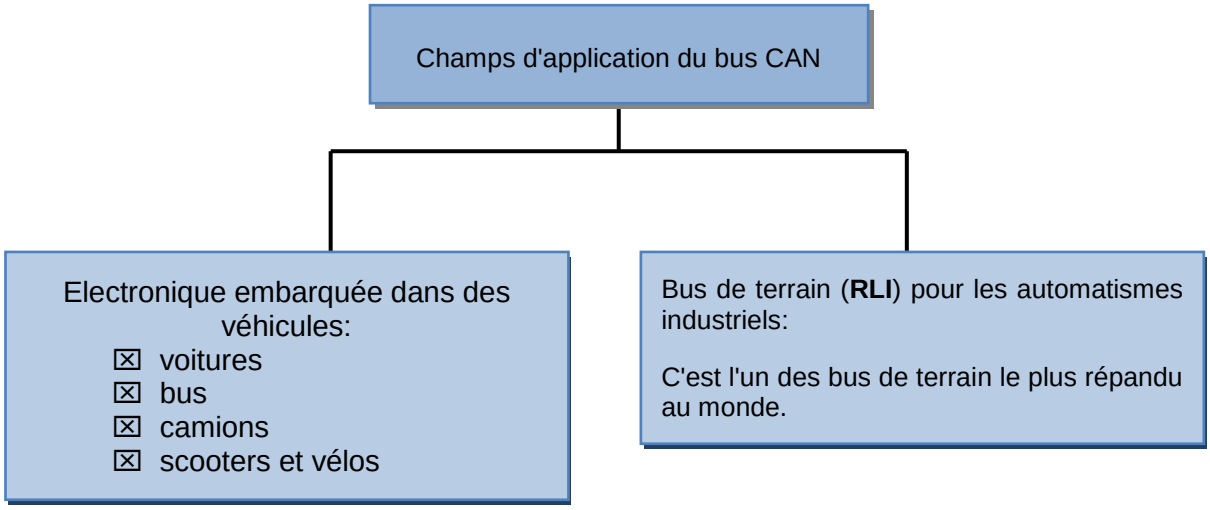
D'où la nécessité d'utiliser le multiplexage qui permet la communication à tour de rôle sur le même câble.



b) La philosophie du CAN

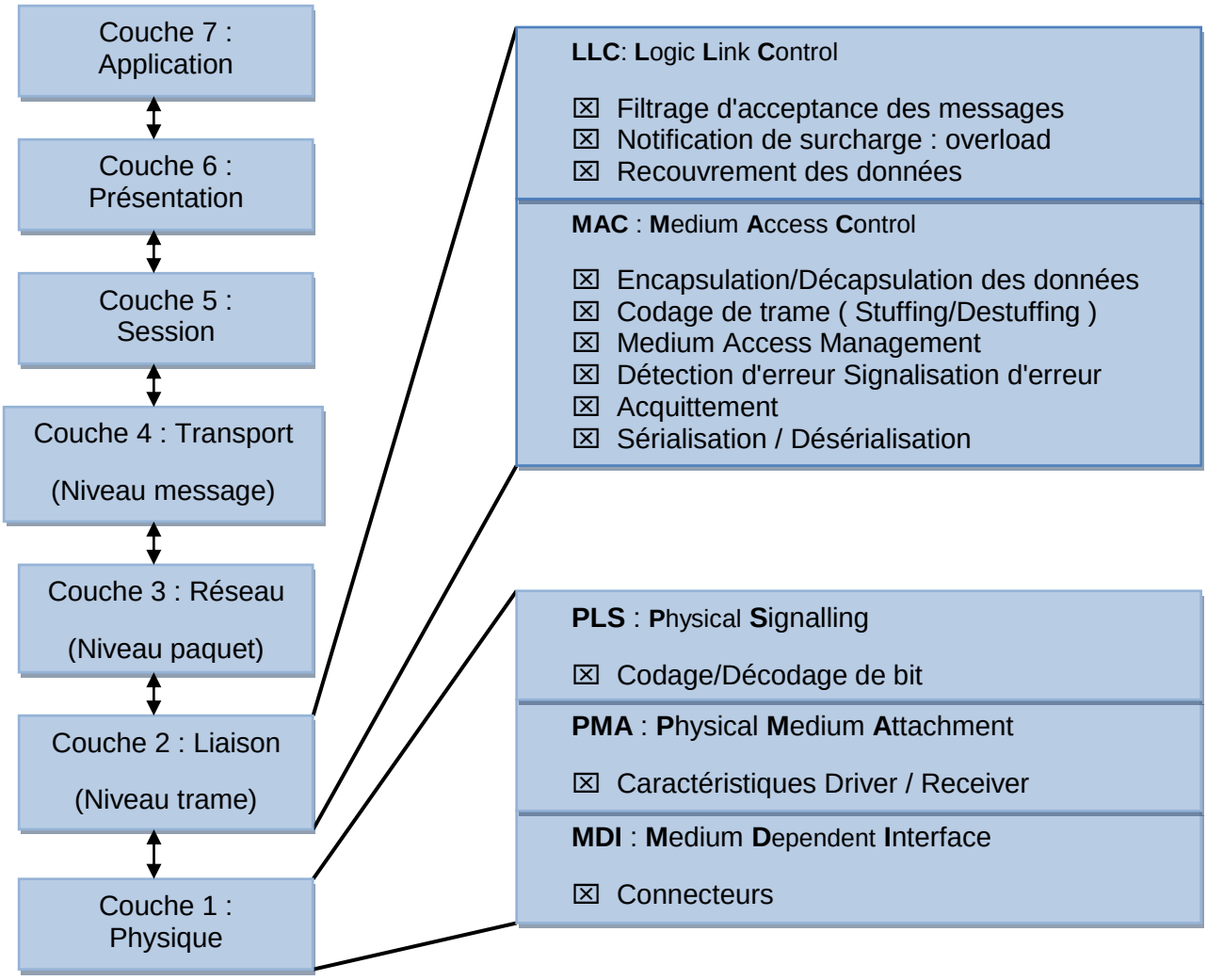


c) Les champs d'application du bus CAN



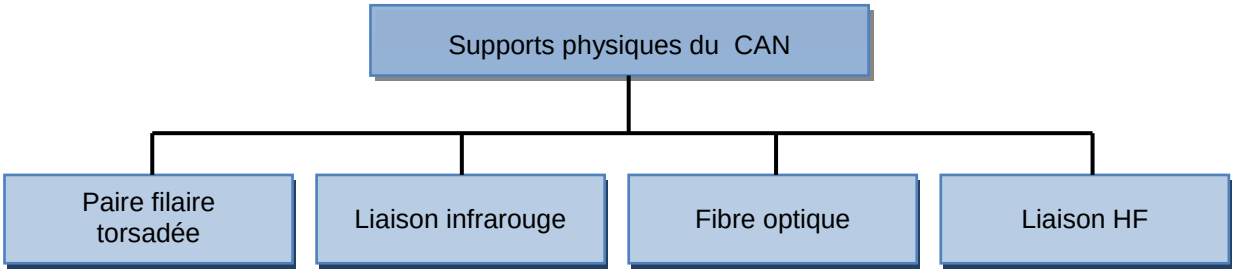
d) Le bus CAN et le modèle OSI

Le protocole CAN ne définit que la couche liaison de données (couche 2) et la couche physique (couche 1).



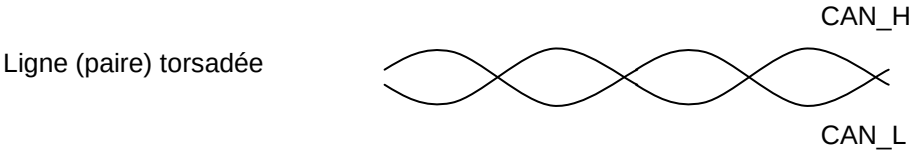
2. Propriétés physiques du CAN.

a) Le support physique

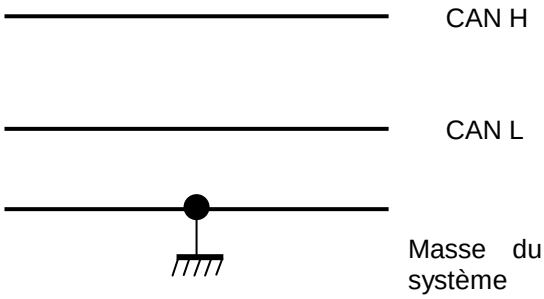


Le CAN embarqué dans un véhicule :

La transmission des données est effectuée sur une paire filaire différentielle.



La ligne est constituée de 2 fils de données ainsi que de la masse.

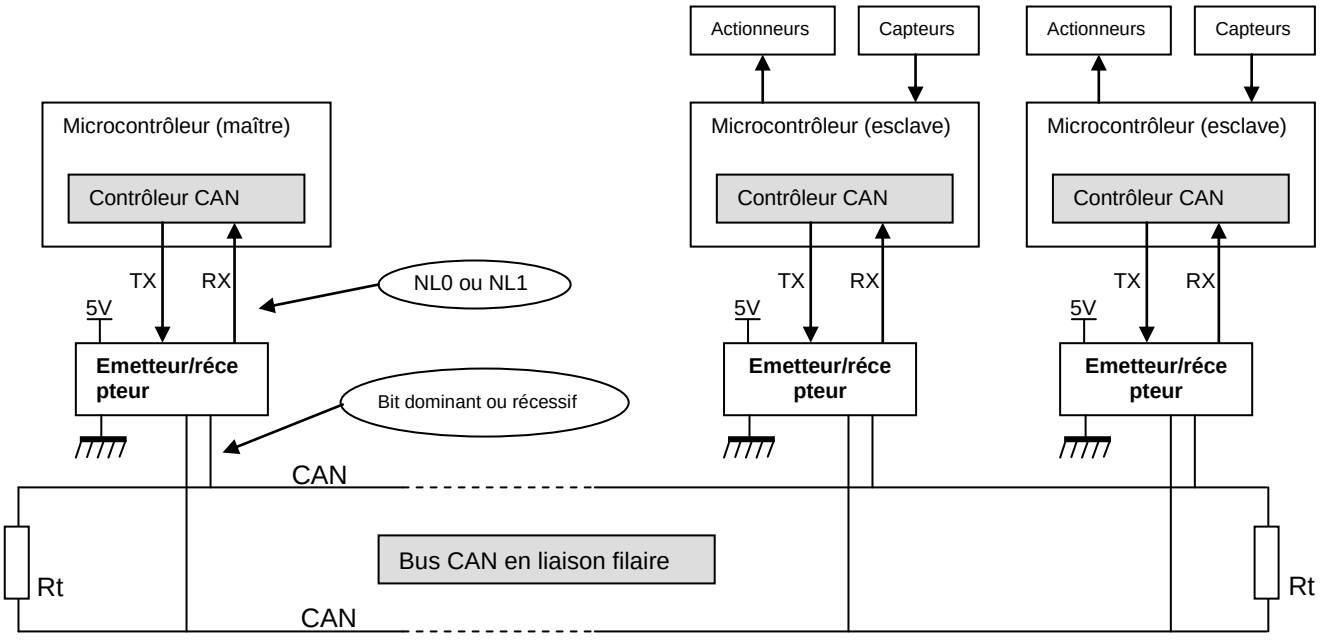


Le **CAN** est un bus de terrain, soumis à des parasites importants. La transmission en paire différentielle permet de s'affranchir de ces problèmes car les montages différentiels ont un fort taux de réjection en mode commun CMRR.

Les débits varient en fonction de la longueur d'une paire torsadée:

Débit	Longueur
1 Mbits/s	40m
500 Kbits/s	100m
100 Kbits/s	500m
20 Kbits/s	1000m

b) Synoptique d'une liaison filaire CAN intégrée dans un véhicule :



En pratique, il y a trois bus CAN différents dans un véhicule, à des débits différents :

- ✓ Un bus rapide pour gérer la sécurité (freinage, ABS, détection chocs, airbags...).
- ✓ Un bus à vitesse moyenne pour gérer le moteur (commandes et capteurs).
- ✓ Un bus lent pour gérer tous les accessoires (lampes, moteurs d'asservissements, boutons...).

Lien entre le modèle OSI et le synoptique d'une liaison filaire CAN:

☒ La couche 2 est réalisée par le contrôleur CAN intégré dans un microcontrôleur.
☒ La couche 1:
- ➔ La strate PLS est réalisée par le contrôleur CAN intégré dans le microcontrôleur.
- ➔ La strate PMA est réalisée par l'émetteur / récepteur (tranceiver).
- ➔ La strate MDI : permet de connecter la carte électronique à la paire filaire.

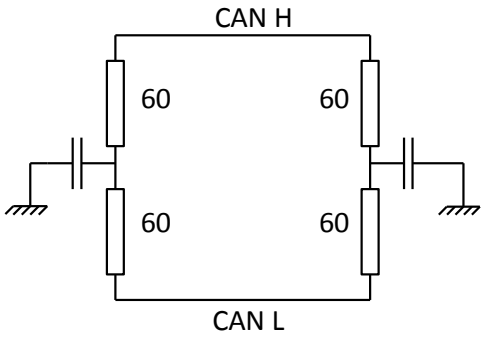
Rôle des résistances de terminaison :

Si l'on fait circuler des signaux de tension sur le bus, sans résistance de terminaison de ligne, les signaux vont se réfléchir sur les extrémités et vont créer des parasites qui risquent de perturber les émissions suivantes sur le bus (identique à une onde qui rebondirait contre un mur).

Pour éviter ces phénomènes de signal réfléchit en bout de câble, on place à l'extrémité une impédance identique à celle du câble. On trouvera donc à chaque extrémité du réseau deux résistances de 120 Ω. Ces résistances de fin de ligne sont intégrées aux extrémités du réseau CAN dans deux participants, en fonction de la topologie et l'architecture du réseau.

Un contrôle rapide de la continuité du réseau peut-être fait en mesurant la résistance entre CAN – H et CAN – L (hors tension et tous les calculateurs branchés).

On mesure deux résistances de 120 Ω en parallèles, soit 60 Ω.



La mesure de toute autre valeur est une anomalie :

☒ R > 60 Ω ➔ coupure de ligne ou si R = 120Ω absence d'une des deux résistances de terminaison.
☒ R < 60 Ω ➔ ligne en court-circuit, ou plus de deux résistances de terminaison dans le réseau.

c) Niveau dominant et niveau récessif

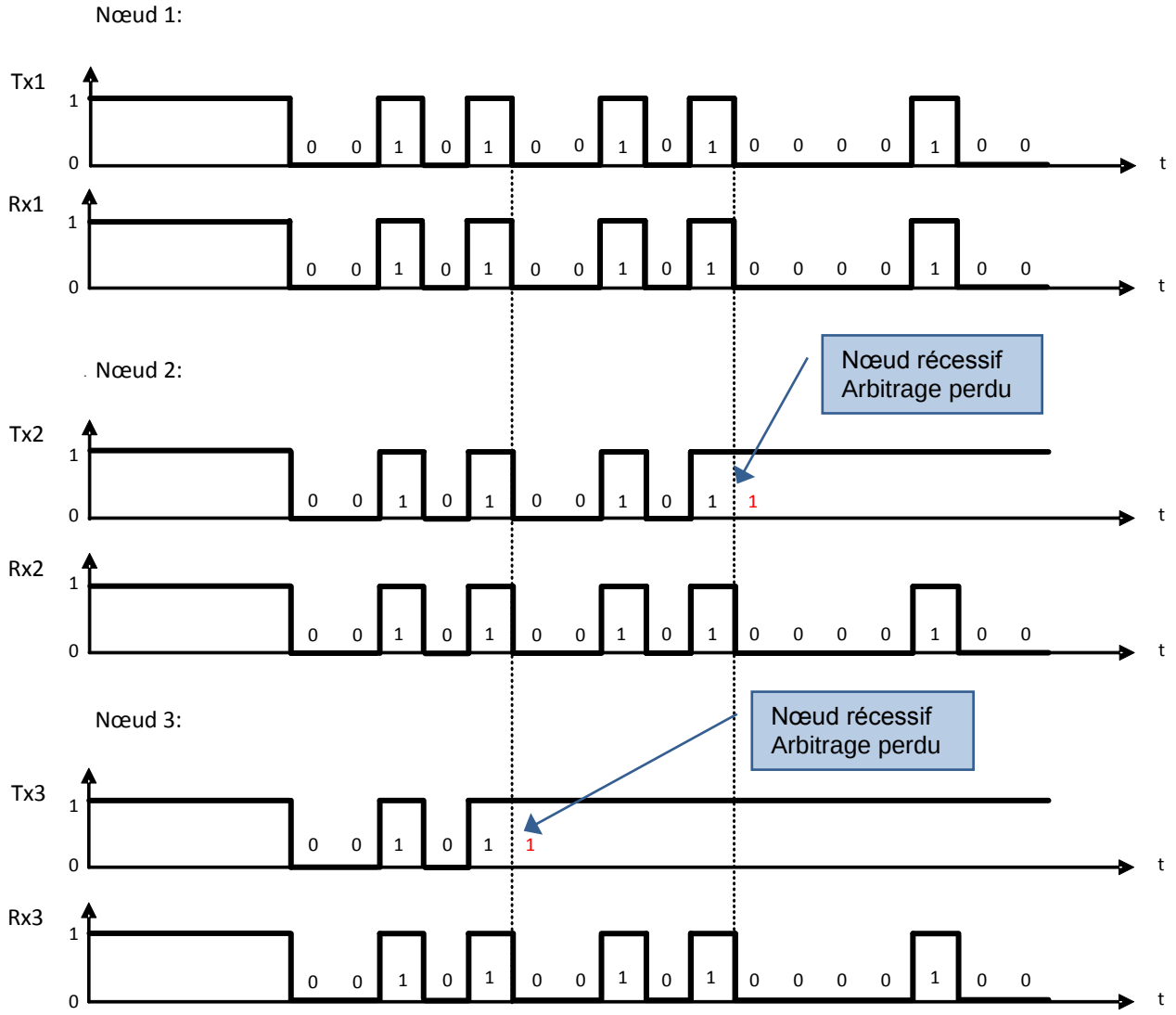
Un niveau électrique récessif sur le bus CAN correspond à un niveau logique haut sur Tx.

Un niveau électrique dominant sur le bus CAN correspond à un niveau logique bas sur Tx.

d) Arbitrage

Tout conflit de bus est résolu par le mécanisme du "ET câblé", c'est-à-dire qu'un état **dominant** écrase un état **récessif**. Si plusieurs nœuds débutent leur trame en même temps, le premier qui présente un bit récessif alors qu'au moins un autre présente un bit dominant perd l'arbitrage.

Exemple : trois nœuds souhaitent prendre simultanément le contrôle de la communication :



Tous les nœuds perdants deviennent automatiquement des récepteurs du message, et ne tentent à nouveau d'émettre que lorsque le bus se libère.

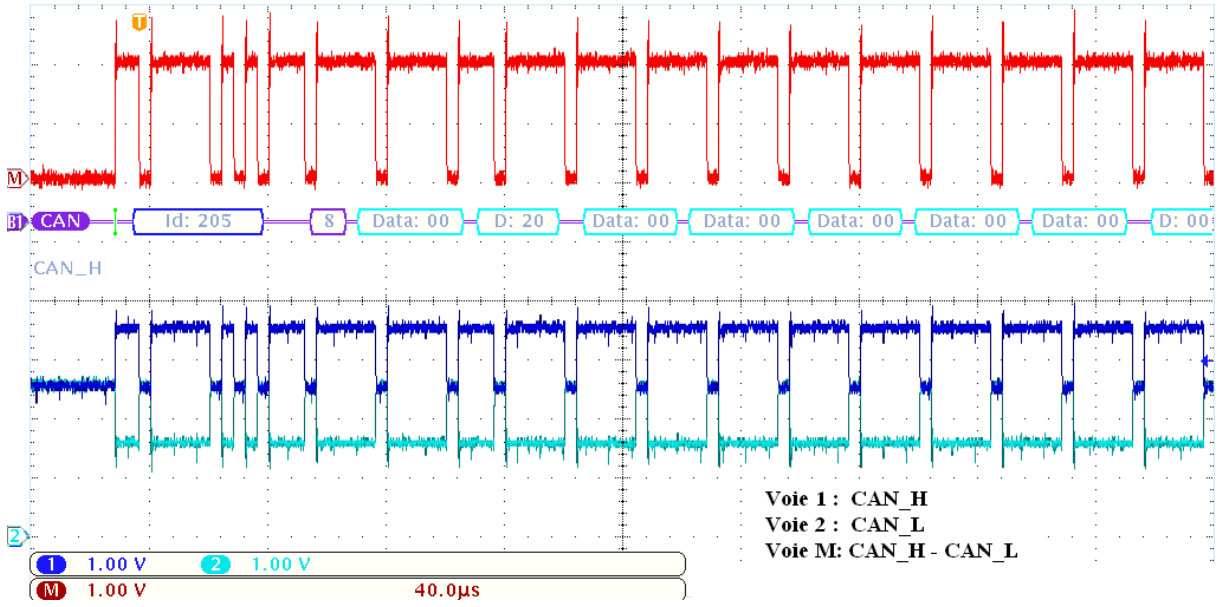
Pour déterminer la priorité des messages, le CAN utilise la méthode CSMA/CR (Carrier Sense, Multiple Access with Collision Resolution) avec la capacité de l'arbitrage non destructif afin d'offrir une disponibilité maximale du bus.

e) Version du bus CAN et niveaux électriques

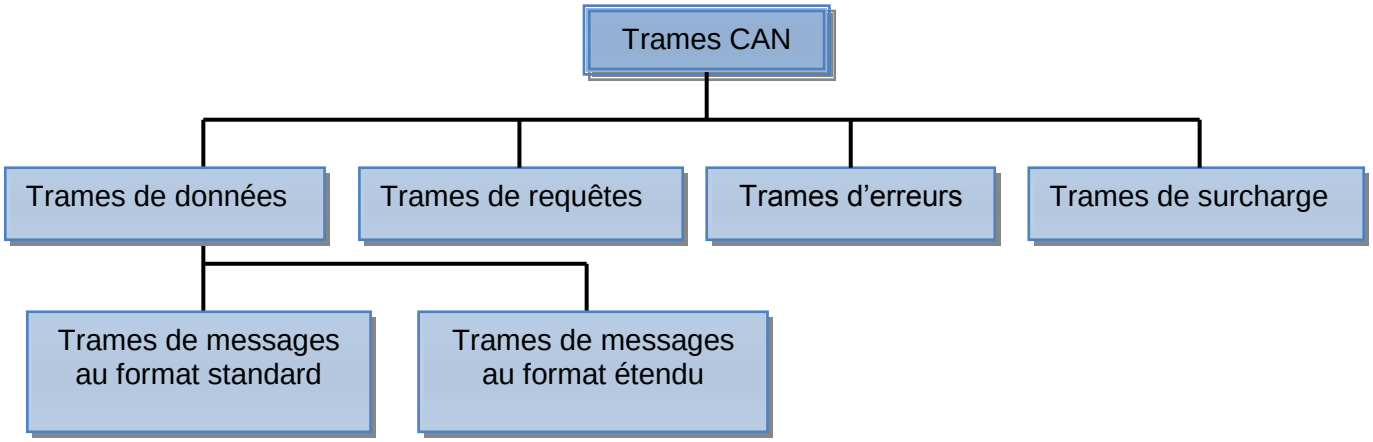
BUS CAN

BUS CAN **LOW SPEED**	BUS CAN **HIGHT SPEED**
☒ Débit : 10kb/s à 125 kb/s	☒ Débit : 250kb/s à 1 Mb/s
☒ Tension d'alimentation: 5V	☒ Tension d'alimentation: 5V
☒ Courant de sortie : > 1 mA sur 2,2 kΩ	☒ Courant de sortie : 25 à 50 mA sur 60Ω
☒ Nombre de nœuds sur le bus : 2 à 20	☒ Nombre de nœuds sur le bus : 2 à 20
☒ Niveau dominant : CAN H = 4V CAN L = 1V	☒ Niveau dominant : CAN H = 3.5V CAN L = 1.5V
☒ Niveau récessif : CAN H = 1.75V CAN L = 3.25V	☒ Niveau récessif : CAN H = 2.5V CAN L = 2.5V
☒ Caractéristique du câble : 30 pF entre les câbles de ligne	☒ Caractéristique du câble : 2 x 120Ω

Oscillogrammes des signaux électriques CAN HIGH SPEED 250 kbits/s :

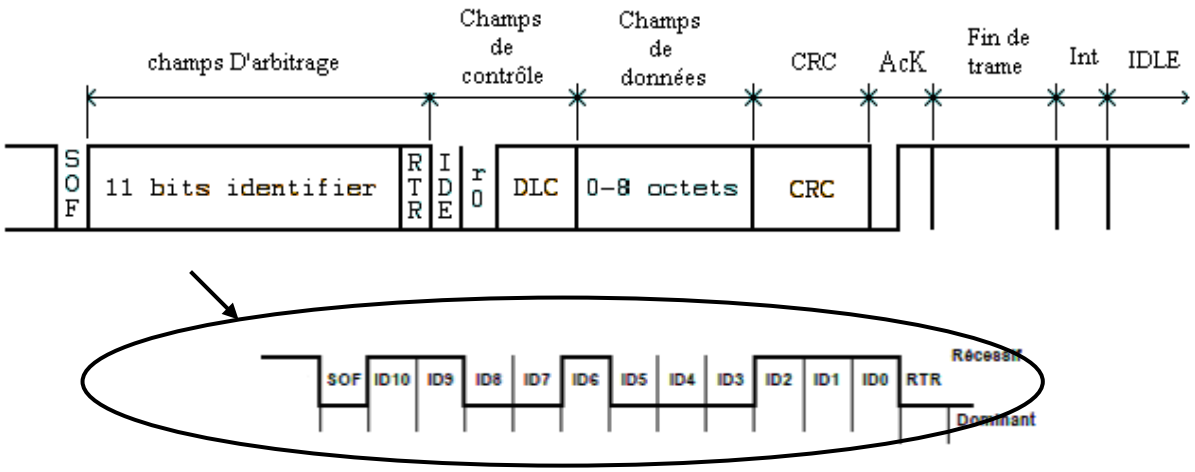


3. Les différentes trames CAN :



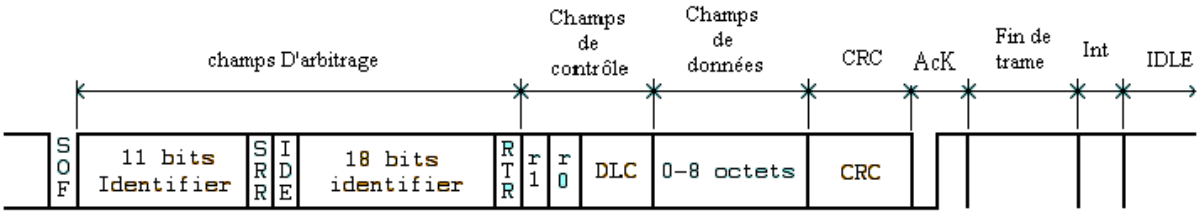
a) Trame de données au format standard CAN 2.0 A :

Le format CAN 2.0A est utilisé dans les voitures.



b) Trame de données au format étendu CAN 2.0 B :

Le format CAN 2.0B est utilisé dans les bus, camions, tracteurs ...

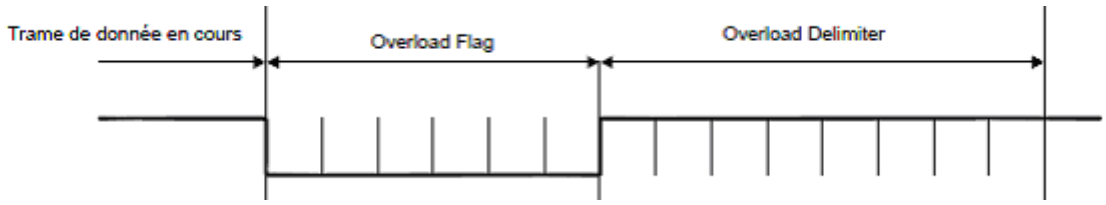


c) Trame de requêtes

Une trame de requête est constituée de la même manière qu'une trame de données sauf que le champ de données est vide.

d) Trame de surcharge

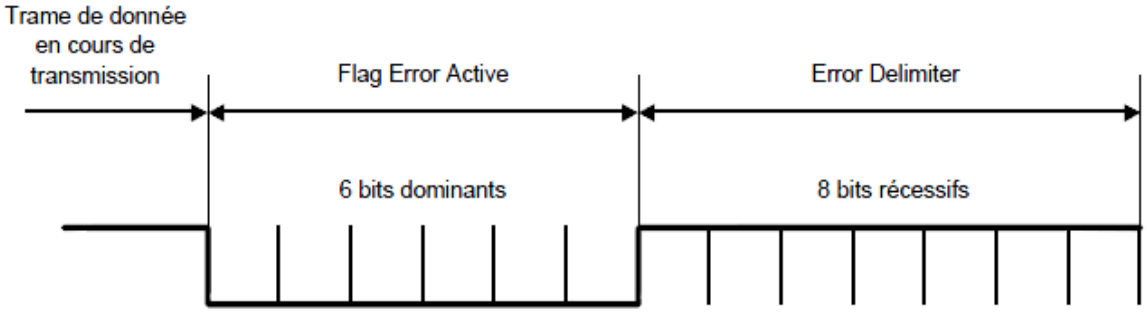
Cette trame indique qu'une station est surchargée pendant un certain laps de temps.



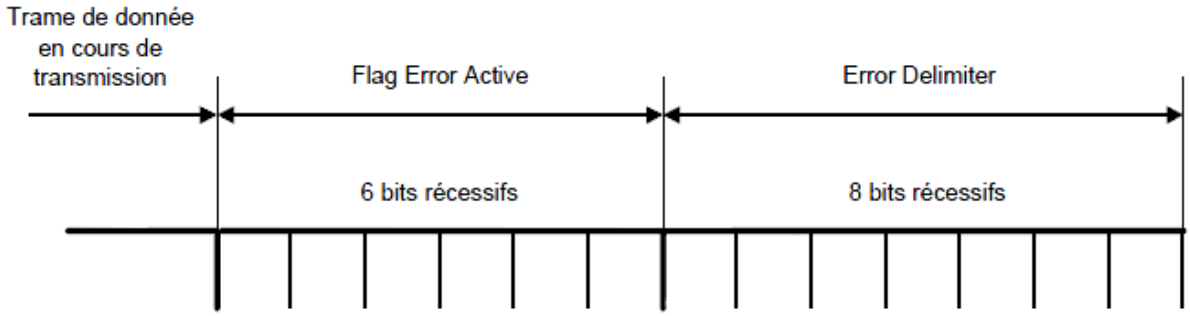
e) Trames d'erreurs :

Il existe deux trames d'erreur possible (pour plus de précision voir chapitre 5).

☒ La trame d'erreur active :



☒ La trame d'erreur passive :



4. Constitution d'une trame de données

a) Le début de trame SOF (*Start Of Frame*)

Le Bit **SOF** **S**tart **O**f **F**rame est dominant, il signale à toutes les stations le début d'un échange.

Il ne peut apparaitre que si le bus était au repos précédemment.

Les nœuds se synchronisent sur le front de ce bit.

b) Le champ d'arbitrage ou identificateur

Le champ d'arbitrage est compose de :

- ☒ L'identificateur **ID** sur 11 bits (CAN 2.0A) ou 29 bits = 11 + 18 (CAN 2.0B).
- ☒ Le type de trame **RTR** (**R**emote **T**ransmission **R**equest*)* sur 1 bit. Il est dominant pour une trame de données et récessif pour une trame de requête.

Les trames de données transmises par un nœud sur le bus ne contiennent ni une adresse du nœud expéditeur ou du nœud destinataire.

L'identificateur **ID** indique la fonction système réalisée par le message.

Chaque nœud recevant un message regarde si celui-ci est intéressant pour lui grâce à l'ID. Si c'est le cas, il le traite, sinon, il l'ignore.

Cet unique ID indique aussi la priorité des messages. Plus la valeur est faible, plus le message sera prioritaire. Si deux nœuds ou plus cherchent à avoir accès au bus en même temps, c'est celui de plus haute priorité qui gagne. Les messages de priorité inférieure seront automatiquement retransmis lorsque le bus sera libre.

> la priorité d'un message est déterminée par la valeur de son **ID**: les adresses basses ont priorité sur les adresses hautes.

Pour l'identificateur les bits sont transmis dans l'ordre, de ID_10 à ID_0 (le moins significatif est ID_0). Par ailleurs les 7 bits les plus significatifs (de ID_10 à ID_4) ne doivent pas tous être récessifs. Pour des raisons de compatibilité avec des anciens circuits, les 4 derniers bits de l'identificateur (ID_3 à ID_0) ne sont pas utilisés, ce qui réduit le nombre de combinaisons possibles.

c) Le champ de contrôle

Il est composé de :

- ☒ La longueur des données **DLC** (**D**ata **L**ength **C**ode) sur 6 bits, indique le nombre d'octets de données de 0 à 8.
- ☒ Les bits r1 et r0 réservés pour un usage ultérieur.

Pour la trame de données, DLC indique le nombre d'octets contenu dans le champs de données.

Pour la trame de requête, DLC indique le nombre d'octets contenu dans le champ de données qui devra être retourné par la trame de données demandée.

d) Le champ de données

Contient les données à transmettre par la trame de donnée, de 0 à 8 octets. Le bit de poids fort est transmis en premier.

e) Le champ de CRC

Il est composé de :

- ☒ La séquence **CRC** (**C**yclic **R**edundancy **C**ode) sur 15 bits
- ☒ Le délimiteur de **CRC** (**CRC D**elimiter) sur 1 bit au niveau récessif.

La séquence de **CRC** est calculée par la procédure suivante :

Le flot de bits (hors *Bit-Stuffing, voir chapitre 5*), constitué des bits depuis le début de la trame jusqu'à la fin du champ de données (pour une trame de données) ou bien la fin du champ de contrôle (pour une trame de requête) est interprétée comme un polynôme f(x) avec des coefficients 0 et 1 affectés à la présence, effective ou non, de chaque bit.

Le polynôme obtenu est alors multiplié par x^{15} complété pour l'ajout du mot de CRC.

le polynôme ainsi formé est divisé (modulo 2) par le polynôme générateur $g(x)=x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+1$.

La chaîne de bits correspondante à ce polynôme est : $(1100010110011001)_2$.

Le reste de la division du polynôme f(x) par le polynôme générateur g(x) constitue la séquence CRC de 15 bits.

f) Le champ d'acquittement (*Acknoledge*)

Il est composé de :

- ☒ L'acquittement **ACK Slot,** indique qu'un nœud a reçu correctement le message et en informe le transmetteur par un bit dominant
- ☒ Le délimiteur d'acquittement **ACK delimiter**., est un bit au niveau récessif.

g) Le champ de fin de trame EOF (*End Of Frame*)

Chaque trame est terminée par une séquence de 7 bits récessifs.

h) Le champ int

C'est l'espace intertrame sur 3 bits au niveau récessif au minimum. Pendant cette durée :

- ☒ Aucun nœud ne peut commencer une nouvelle trame de données ou de requête
- ☒ Seule action permise est signaler une surcharge (overload)

i) IDLE

En mode inactif la ligne reste au niveau récessif.

> **IDLE :** du mot anglais idle inaction inoccupé, oisif.

5. Fiabilité du bus CAN et protection contre les erreurs mises en place

> **To stuff:** Farcir / Bourrer
> **Stuffing:** le bourrage

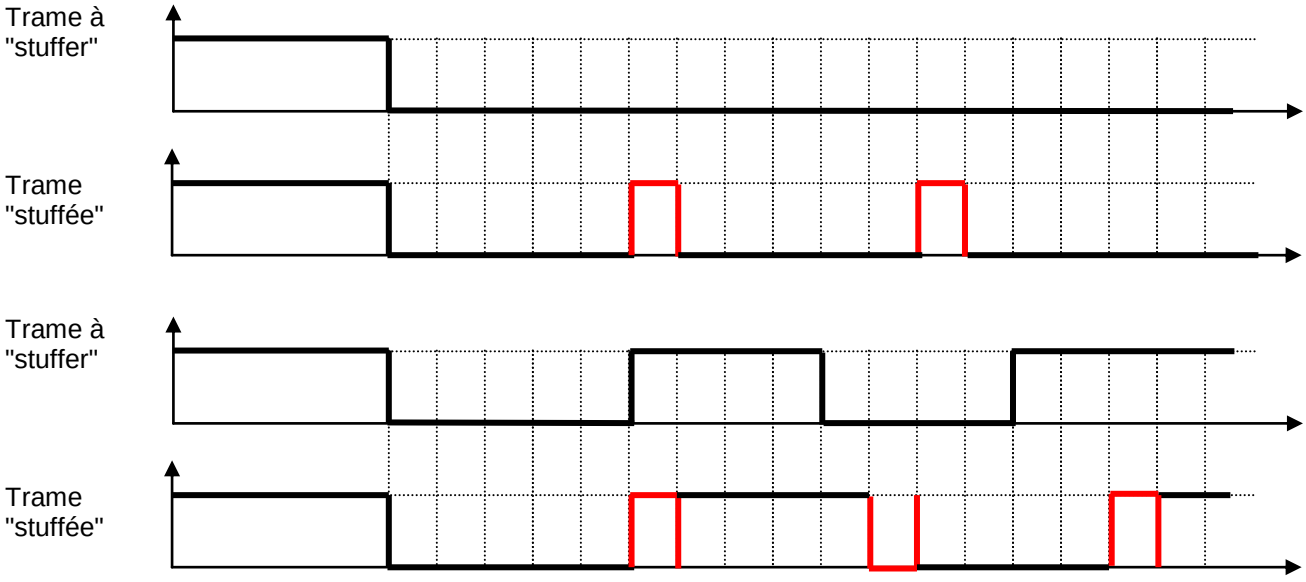
a) Le bit stuffing

Le codage utilisé est de type NRZ. Cela entraine la nécessité d'une horloge stable.

La méthode du bit-stuffing permet la resynchronisation de l'horloge lors de la réception par ajout de transition.

Dès que l'on a émis 5 bits de même polarité sur le bus, on insère un bit de polarité contraire pour casser des chaînes trop importantes de bits identiques.

Cette technique est uniquement active sur les champs de SOF, d'arbitrage, de contrôle, de CRC (délimiteur exclu). Pour un fonctionnement correct de tout le réseau, cette technique doit être implémentée aussi bien à la réception qu'à l'émission.



Dans le pire des cas (trame de 5 bits de même signe puis 4 bits de l'autre signe...), l'ajout du bit-stuffing peut augmenter de 25% le nombre de bit émis.

b) Les erreurs contrôlées par le bus CAN

Différentes erreurs contrôlées par le CAN

Bit error : Chaque fois qu'un émetteur envoie un bit sur le bus, il vérifie en même temps si le niveau émis sur le bus correspond à celui qu'il désire envoyer en faisant une surveillance du bus. Si le niveau ne correspond pas, il le signale par un Bit Error.

Cependant, le Bit Error n'est pas signalé dans les cas suivants :

- Aucune erreur de Bit Error n'est signalée lorsqu'un bit dominant est envoyé dans le champ d'arbitrage à la place d'un bit récessif. Le bit dominant signifie simplement une perte d'arbitrage.
- De même, pour un bit dominant lors de *l'acknowledge slot*, à la place d'un bit récessif.
- Un émetteur envoyant un *flag* d'erreur passive (bit récessif) et recevant un bit dominant, ne doit pas signaler un *Bit Error*.

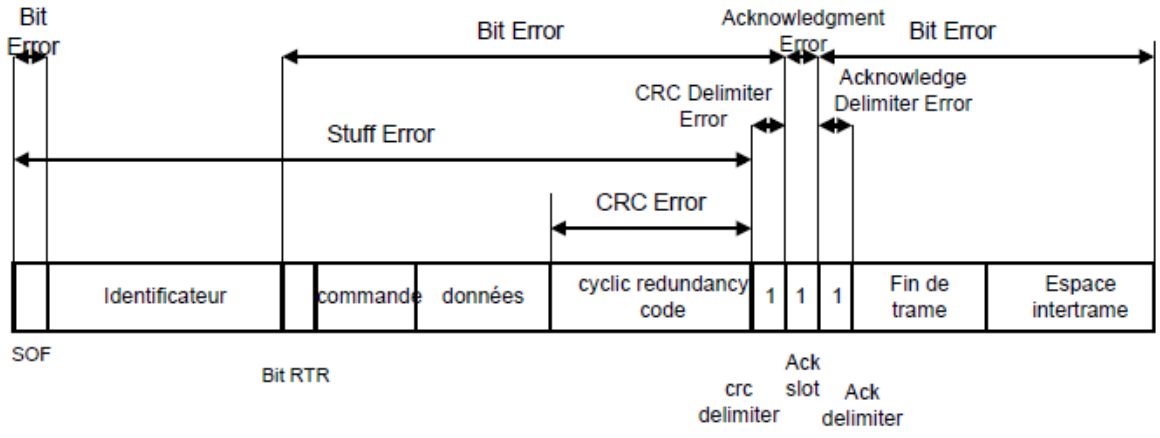
Stuff error : Une erreur de *Stuffing* est détectée à chaque fois qu'il y a 6 bits ou plus consécutifs de même signe sur le bus. Une erreur de *Stuffing* ne doit être signalée que dans les champs d'identificateurs, de commande et de CRC.

CRC error : le CRC calculé par le récepteur est différent de celui envoyé par l'émetteur.

Acknowledge delimiter : le récepteur n'observe pas un bit récessif lors du champ de *Acknowledge Delimiter*. Il en est de même pour le *CRC Delimiter.*

Acknowledgement error : Une erreur de *Slot Acknowledge* est signalée par l'émetteur s'il ne lit pas un bit dominant lors du champ de slot acknowledge.

c) Identification des erreurs dans la trame CAN :



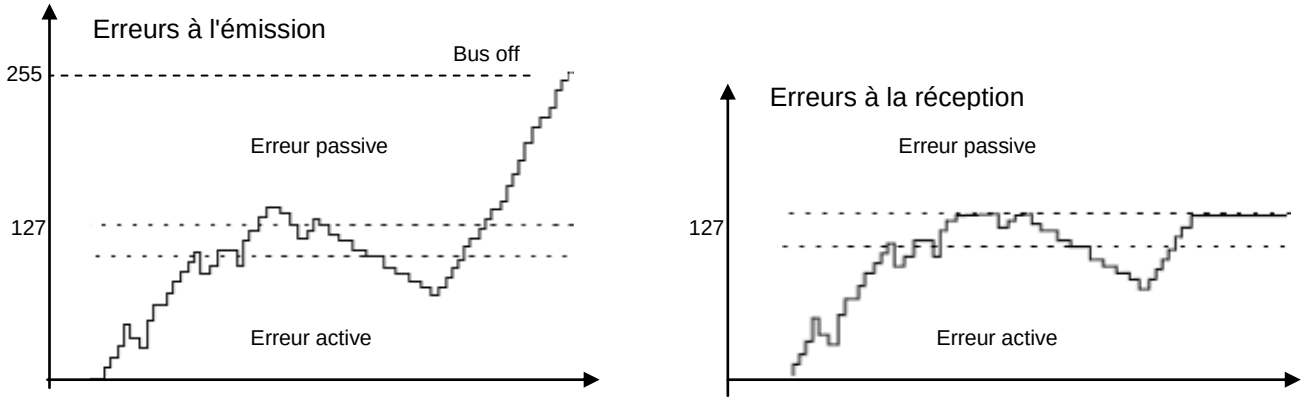
d) Règles de confinement

Chaque nœud possède deux compteurs :

☒ le **T**ransmit **E**rror **C**ounter, **TEC** : compteur d'erreurs de transmission
☒ le **R**eceive **E**rror **C**ounter : **REC** : compteur d'erreurs de réception

Ces compteurs s'incrémentent lorsqu'une erreur est détectées et se décrémentent lorsqu'aucune erreur n'est détectée.

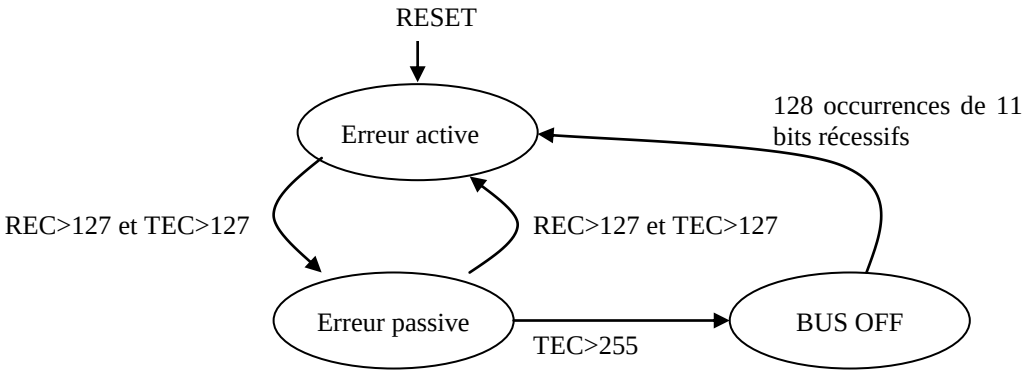
Ils s'incrémentent plus vite qu'ils ne se décrémentent.



Les règles de confinement définissent le mode de fonctionnement du nœud :

☒ Si les deux compteurs sont inférieurs à 127, le nœud travaille en mode **erreur active**.
☒ Si un des deux compteurs est supérieur à 128, le nœud travaille en mode **erreur passive**.
☒ Si un des deux compteurs est supérieur à 255, le nœud doit être déconnecté du bus. Il est **bus-off**. Il pourra se reconnecter ultérieurement après 128 occurrences de 11 bits récessifs sans erreur. (une occurrence de 11 bits récessifs : ACK delimiter + Fin de trame + Intertrame)

La gestion des erreurs et des trames associées est représenté par le diagramme d'état ci-dessous :

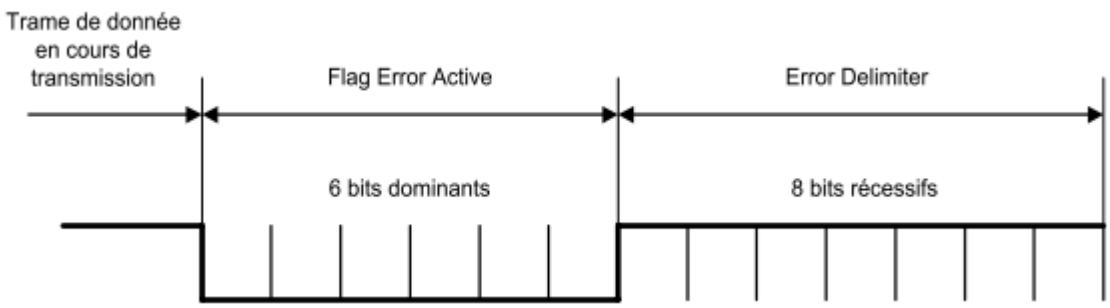


e) Les trames d'erreur

Lorsqu'une erreur est détectée par un nœud, une trame d'erreur est transmise sur le bus à destination des autres noeuds. Cette trame comporte 2 champs : Le **flag error** et le **error delimiter**.

Pour un nœud en mode erreur active :

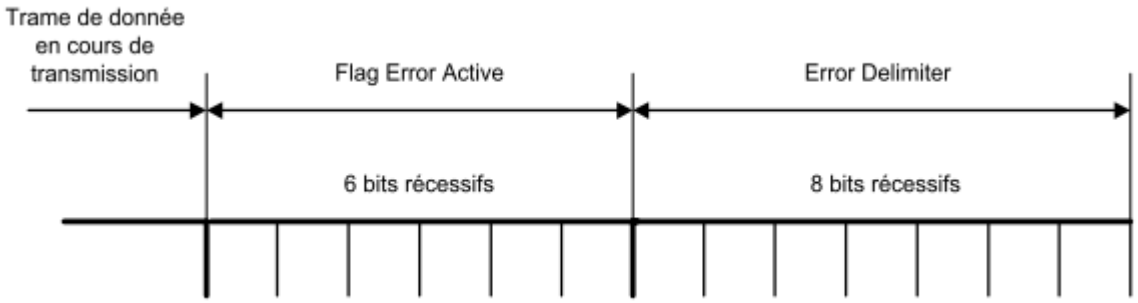
☒ Le champ flag error est composé de 6 bits dominants
☒ Le champ error delimiter est composé de 8 bits récessifs.



Lorsque les autres nœuds (récepteurs) détectent les 6 bits dominants, ils détectent aussi une erreur de bits stuffing et émettent aussitôt leur trame d'erreur. Si d'autres nœuds sont en mode erreur active, il peut donc y avoir de nouveau 6 bits dominants. Dans tous les cas la trame d'erreur se termine par les 8 bits récessifs.

Pour un nœud en mode erreur passive :

☒ Le champ flag error est composé de 6 bits récessifs
☒ Le champ error delimiter est composé de 8 bits récessifs



Lorsque les autres nœuds (récepteurs) détectent les 6 bits récessifs, ils détectent aussi une erreur de bits stuffing et émettent aussitôt leur trame d'erreur (passive ou active selon leur mode). Dans tous les cas la trame d'erreur se termine par les 8 bits récessifs.

f) Recouvrement des erreurs

Lorsque la trame d'erreur est terminée la trame défaillante est de nouveau transmise jusqu'à ce qu'il n'y ait plus d'erreurs ou que le nœud passe en mode bus-off.

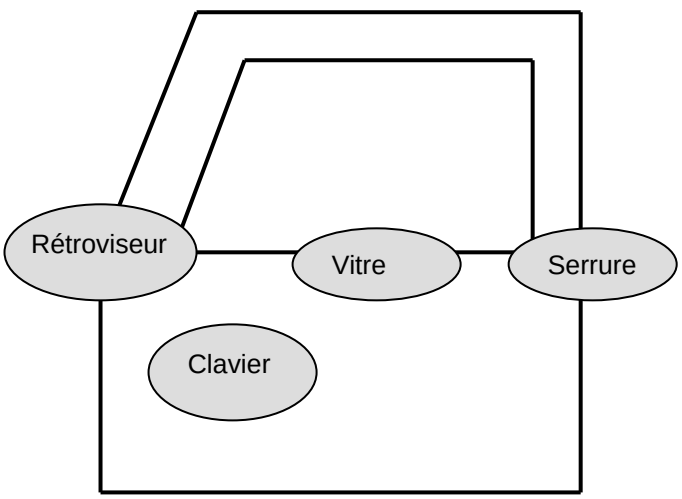
6. Complément du bus CAN : le bus LIN (Local Interconnect Network).

Le bus CAN présente l'inconvénient de travailler en 5V alors que la batterie d'une voiture est de 12V.

De ce fait il impose une interface non normalisée de puissance permettant de commander les actionneurs.

La solution aujourd'hui retenue par l'industrie automobile est de prolonger le bus CAN par le bus LIN : l'interface CAN/LIN et les drivers de puissances peuvent être ainsi normalisés.

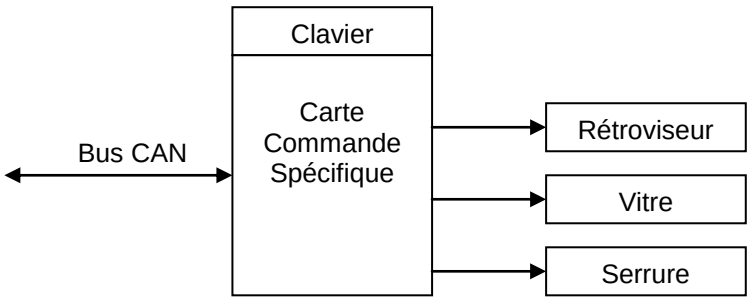
Comparaison des synoptiques électriques d'une portière classique et d'une portière optimisée par bus LIN :



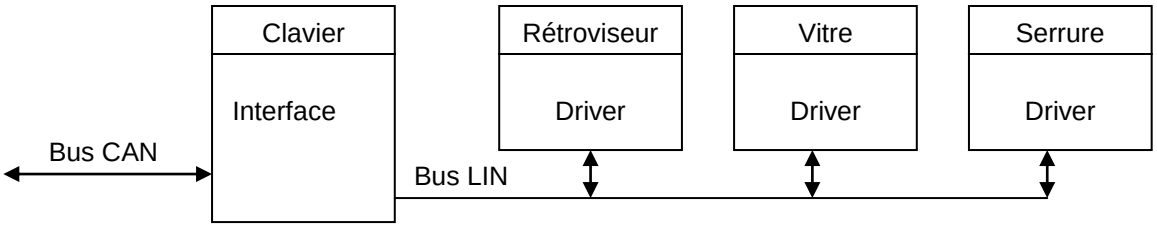
Caractéristiques principales du bus LIN :

☒ Implémentation bas coût pour le matériel ;
☒ Auto synchronisation des esclaves ;
☒ Temps de réponse garanti ;
☒ Débit maximal 20 kbit/s.
☒ Détection d'erreurs :
- 2 bits de parité ;
- CRC à la fin du message ;
- Messages de diagnostic.

Portière classique :



Portière optimisée par bus LIN :



7. Bibliographie et sources

Livre:

Le BUS CAN / Auteur Dominique PARET / éditions DUNOD.

Internet :

Le bus CAN / http://uuu.enseirb.fr/~kadionik/formation/canbus/canbus_enseirb.pdf / auteur Patrick KADIONIK