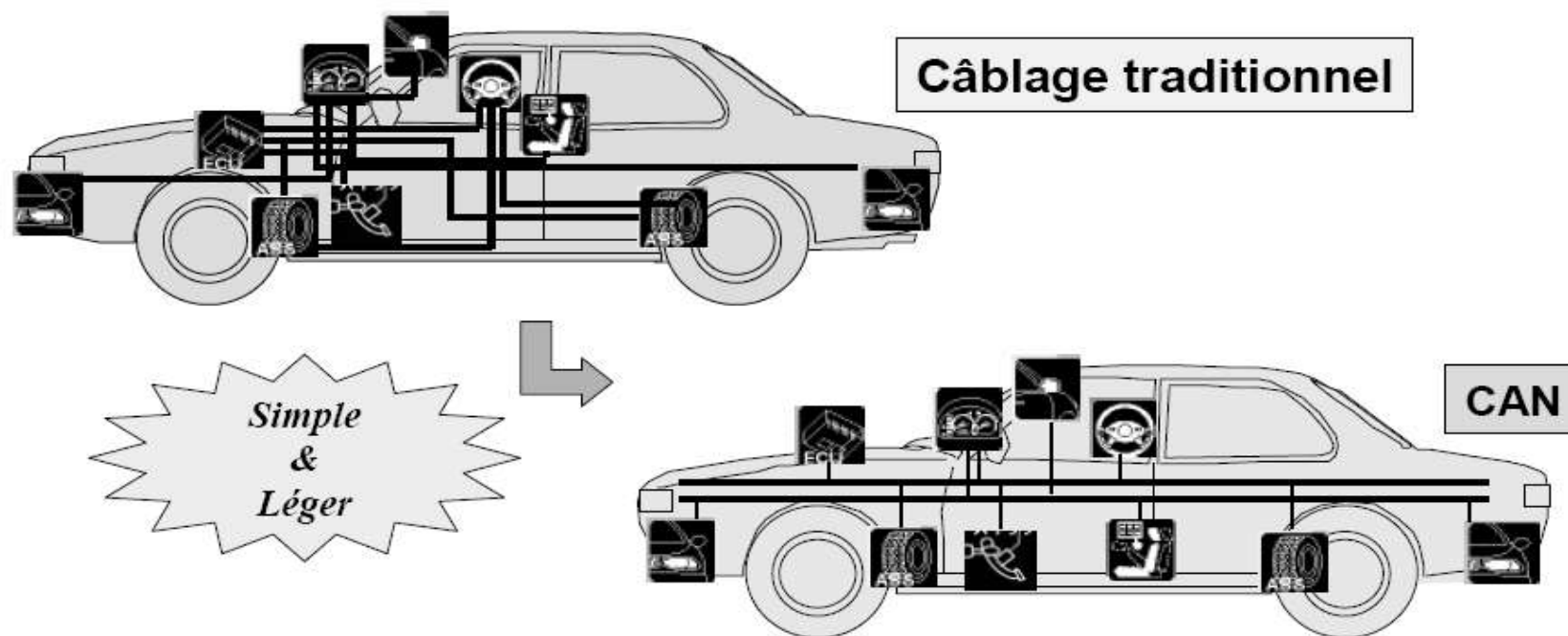


Bus de terrain – Bus CAN

Introduction - Historique

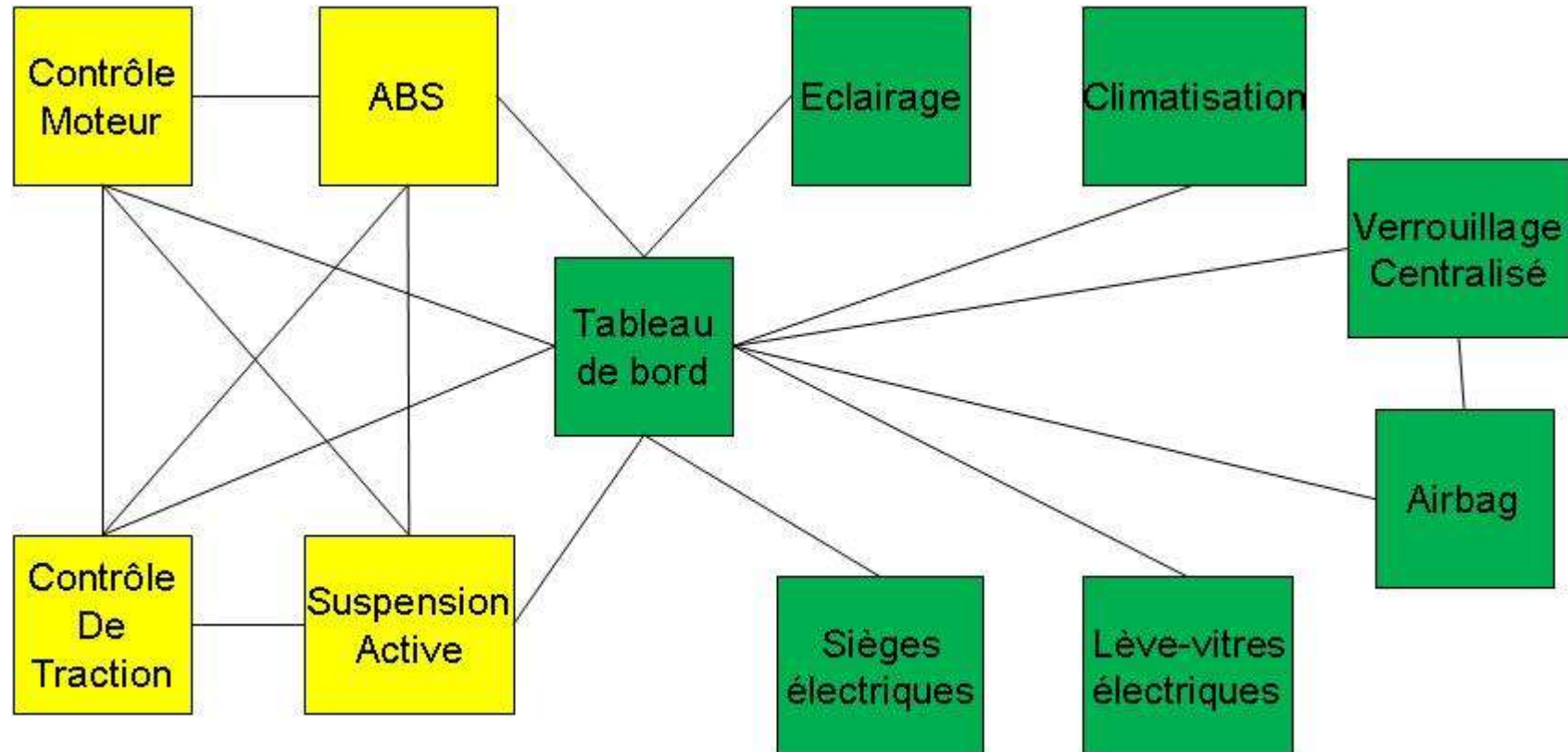
- ▶ Origines : besoin du monde automobile.
- ▶ Concept Robert Bosch Gmbh et université de Wolfenbüttel (1983).
- ▶ Réalisation sur silicium par Intel (1985).
- ▶ Première voiture équipée CAN (1991).
- ▶ Création de CAN in Automation (CiA) en 1992.

Introduction - Evolution du câblages dans l'automobile



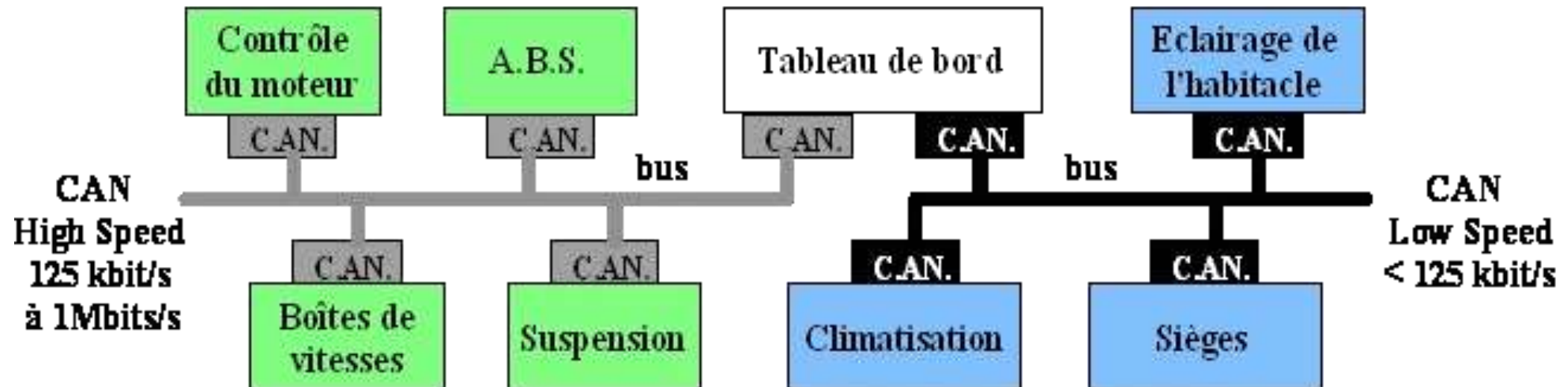
2 câbles pour l'alimentation électrique 12 V DC
1 paire torsadée pour les transferts d'informations

Introduction - Evolution du câblages dans l'automobile



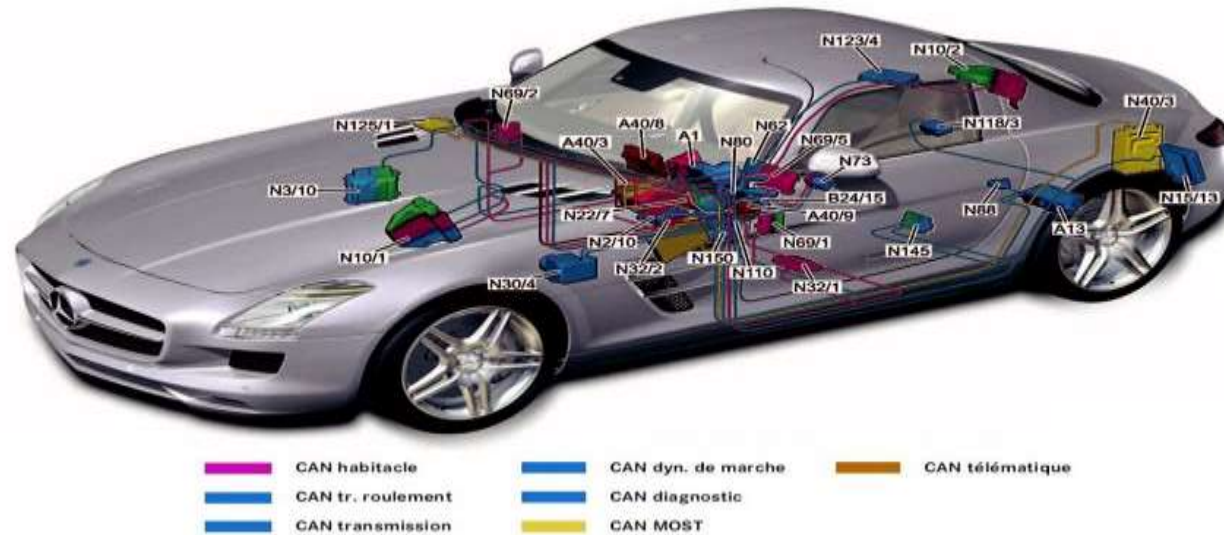
Introduction - Evolution du câblages dans l'automobile

RESEAU C.A.N. AUTOMOBILE



Introduction - Evolution du câblages dans l'automobile

7 réseaux dans un véhicule !



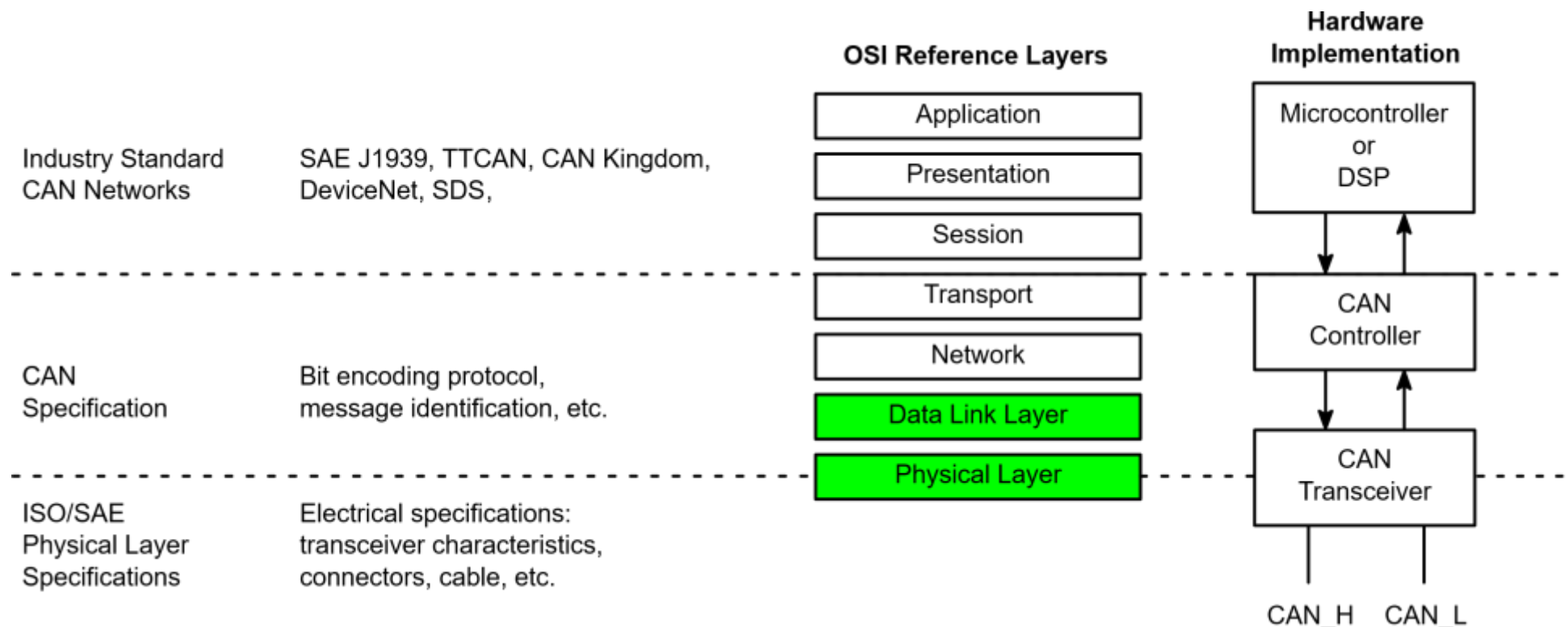
<http://www.forum-mercedes.com/topic-484-sls-amg-mise-en-reseau-calculateurs-bus-can.html>

Introduction – Les avantages de CAN

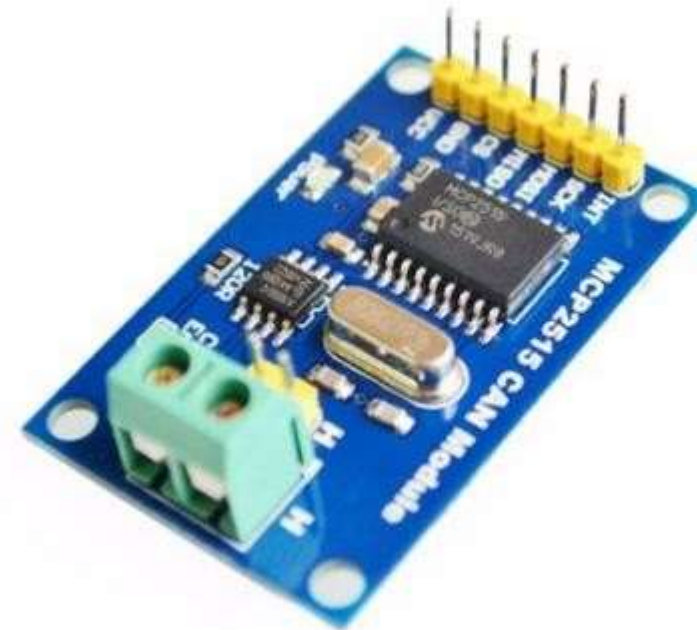
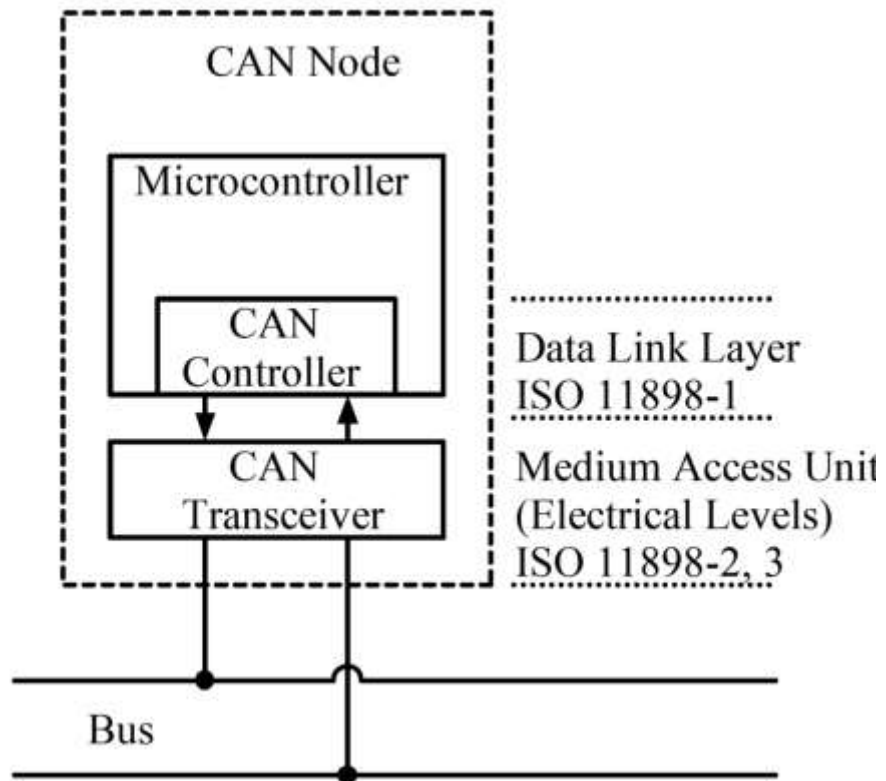
- ▶ Le protocole CAN (Control Area Network) est un protocole de communication série qui supporte des systèmes temps réel avec un haut niveau de fiabilité :
 - ▶ Souplesse de configuration.
 - ▶ Fonctionnement multi-maître.
 - ▶ Détections et signalisations d'erreurs.
 - ▶ Retransmission automatique des messages altérés.
 - ▶ Déconnexion automatique des nœuds défectueux.

Bus CAN et OSI

La norme CAN spécifie les couches Physique et Liaison de données uniquement. D'autres protocoles (devicenet, canopen, ..) spécifient les couches hautes.



La transmission des données dans le bus CAN



La transmission des données dans le bus CAN

Chaque **nœud** est connecté au bus par l'intermédiaire d'une paire torsadée (blindée ou non).



Paire torsadée (blindée ou pas) 2 fils de cuivre isolés de section 0.6 mm²

La transmission des données dans le bus CAN

CAN Low

CAN High

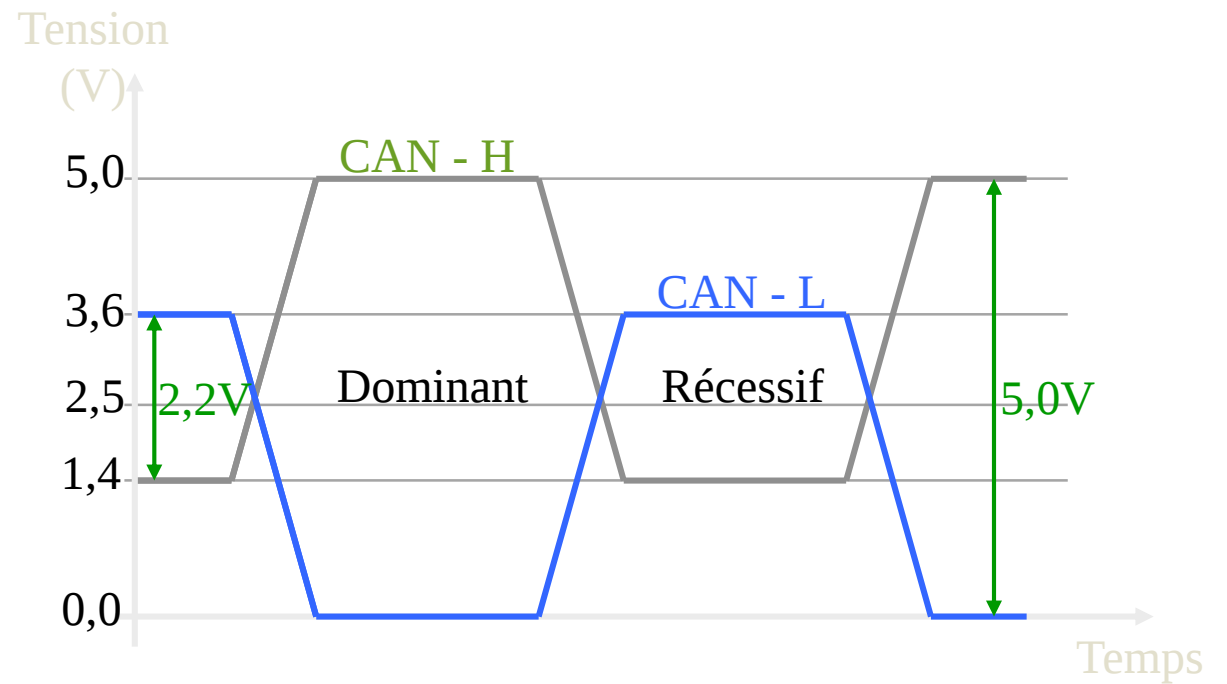
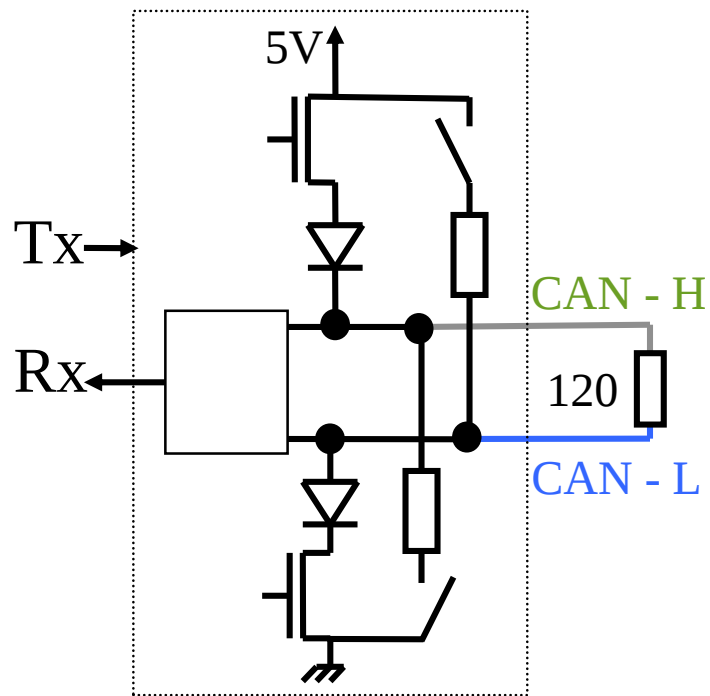
CAN L

Paire filaire

CAN H

La transmission des données dans le bus CAN

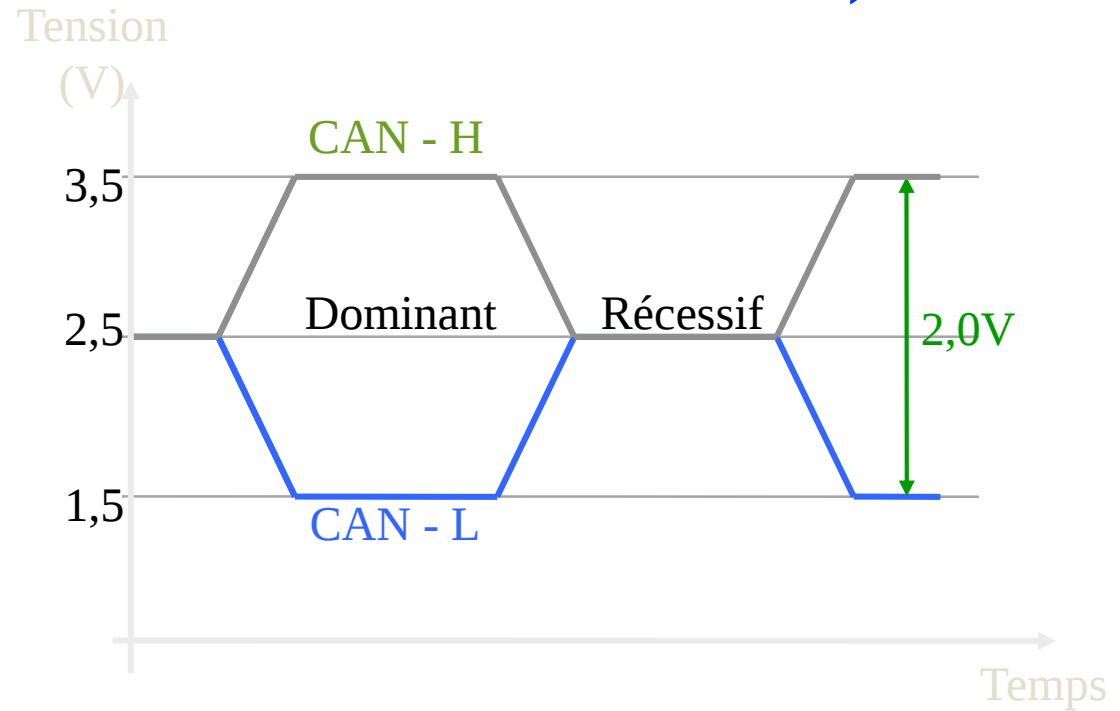
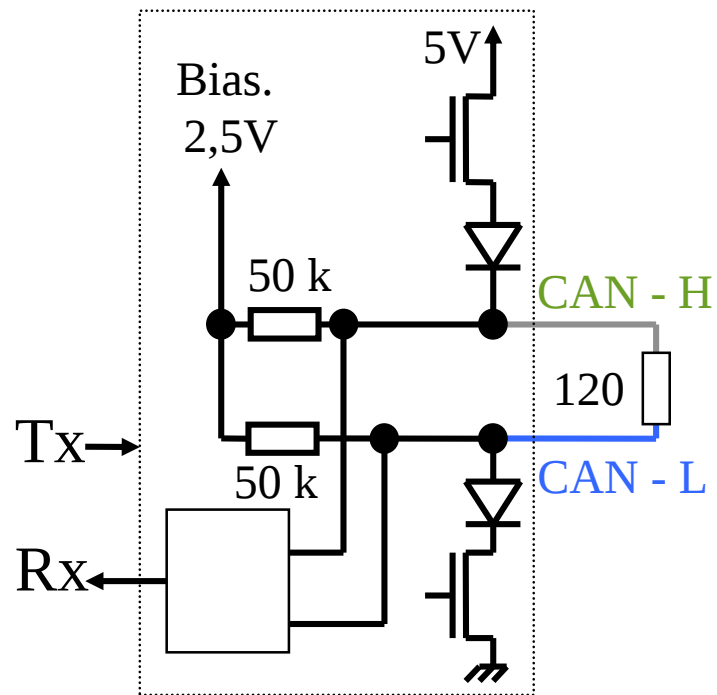
Le CAN low speed (max. 125kbits/s)



La différence de tension entre les deux états est grande

La transmission des données dans le bus CAN

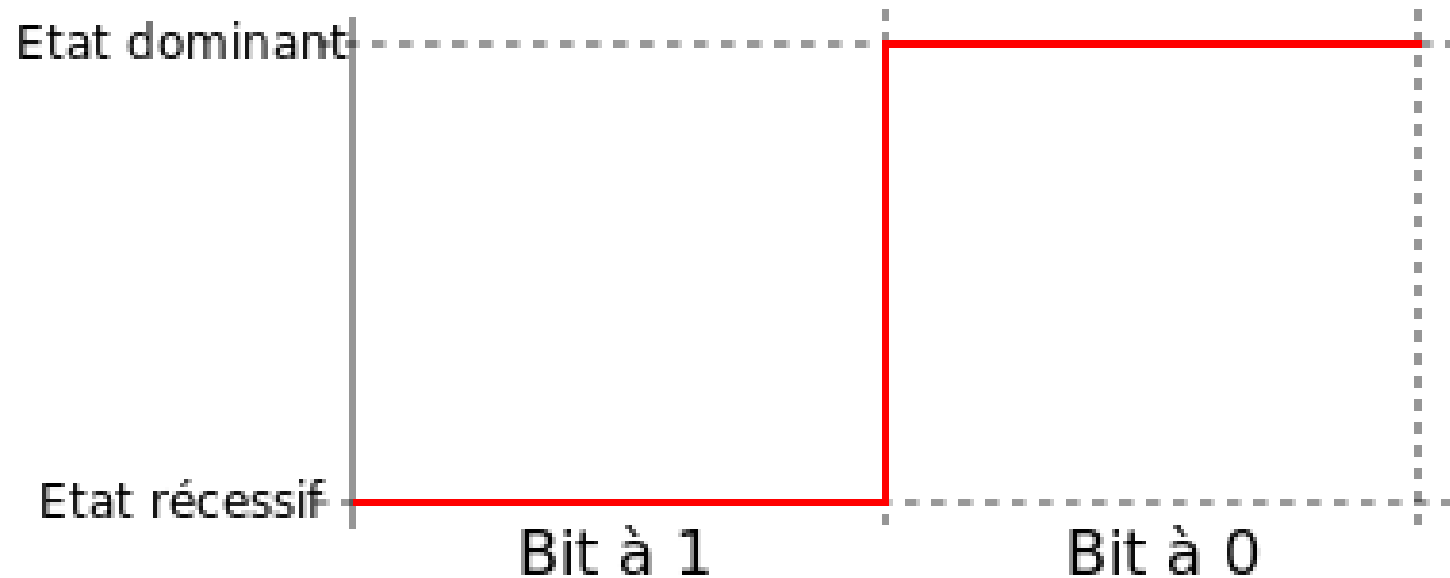
Le CAN *high speed* (max. 1Mbits/s sur 40m)



La différence de tension entre les deux états est faible

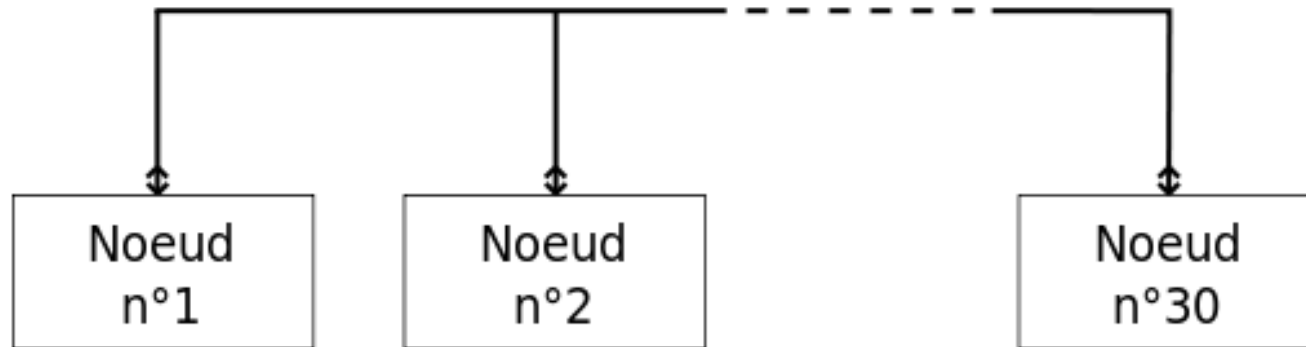
La transmission des données dans le bus CAN

- ▶ Le bus peut avoir l'une des deux valeurs logiques complémentaires définies.
- ▶ Ils sont sous la forme de bit nommé dominant (Bit à 0) ou récessif (Bit à 1).



La transmission des données dans le bus CAN

CAN est un bus de données série bidirectionnel half-duplex dans le domaine automobile,



Pour un bus de données CAN « basse vitesse », le nombre de nœuds est limité à 20.

Pour un bus de données CAN « haute vitesse », il est limité à 30.

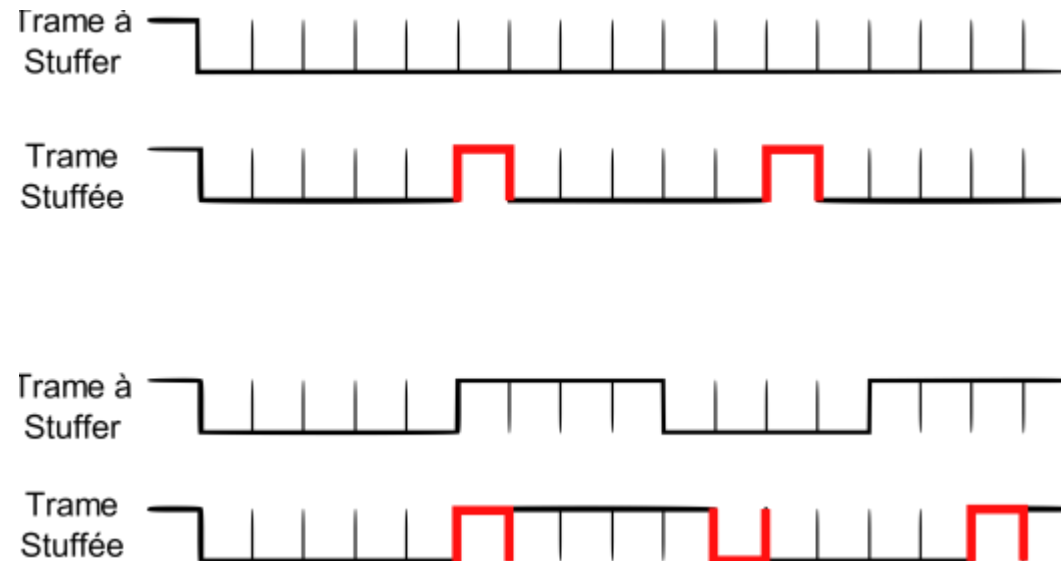
La transmission des données dans le bus CAN

Concepts :

- ▶ **Message** : Chaque information est véhiculée sur le bus à l'aide d'un message (trame de bits) de format défini mais de longueur variable et limitée. Dès que le bus est libre, n'importe quel nœud relié au réseau peut émettre un nouveau message.
- ▶ **Canal de liaison simple** : Le bus consiste en un simple canal bidirectionnel qui transporte les bits. A partir des données transportées, il est possible de récupérer des informations de resynchronisation.
- ▶ **Acquittement** : tous les récepteurs vérifient la validité d'un message reçu, et dans le cas d'un message correct ils doivent acquitter en émettant un flag.
- ▶ **Arbitrage** : En absence d'un hôte, il est impératif de régler les conflits d'accès au bus de communication.
- ▶ **Sécurité de transmission** : Garantir la plus grande sécurité lors de transferts sur le bus. Des dispositifs de **signalisation**, de **détection d'erreurs**, et d'**autotests** ont été implémentés sur chaque nœud d'un bus CAN. On dispose ainsi d'un **monitoring bus** (vérification du bit émis sur le bus), d'un **CRC (Cyclic Redundancy Check)**, d'une procédure de **contrôle de l'architecture du message** et d'une méthode de **Bit-Stuffing**.

La transmission des données dans le bus CAN

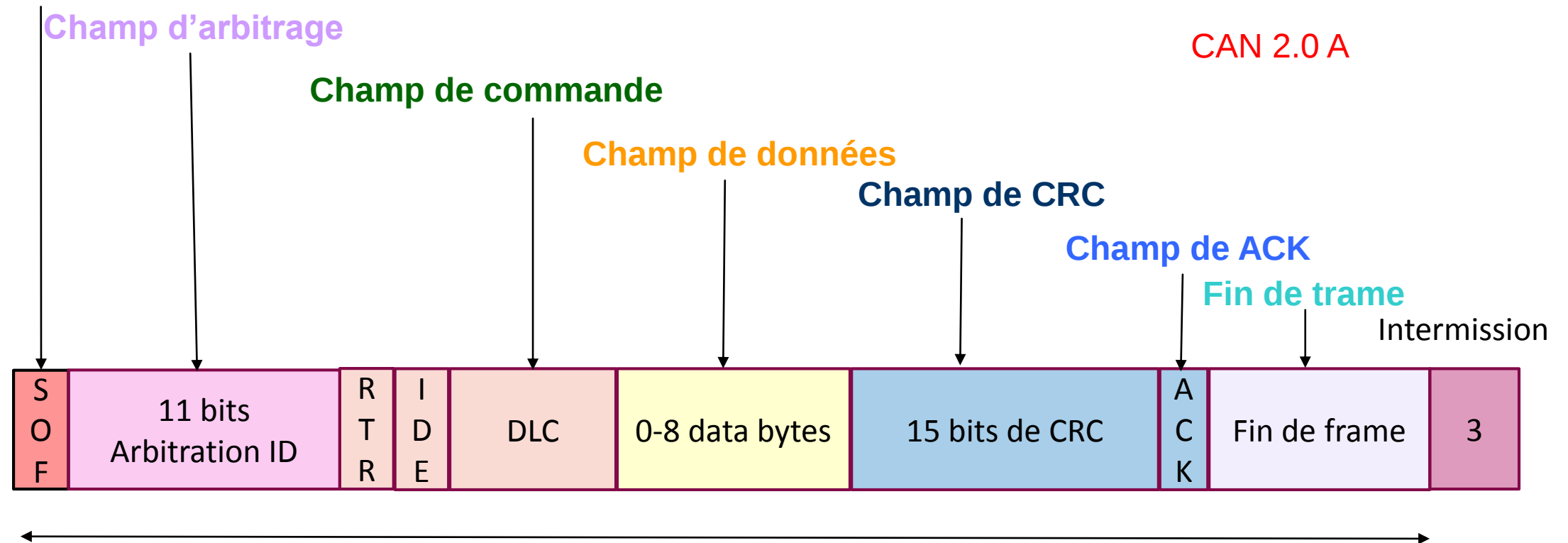
- ▶ Méthode de Bit-Stuffing : Cette méthode consiste, dès que l'on a émis 5 bits de même polarité sur le bus, à insérer un bit de polarité contraire pour casser des chaînes trop importantes de bits identiques
- ▶ Remarque : Cette technique s'applique sur les bits depuis le Start of Frame jusqu'à la séquence de CRC (délimiteur exclu) de la trame.



La trame CAN – Trame de données

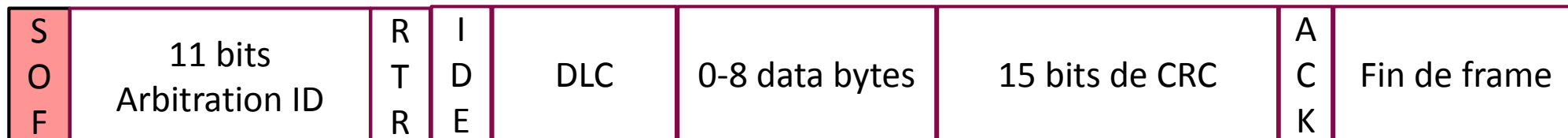
- Chaque nœud peut envoyer des trames de données à intervalles réguliers.

Début de trame



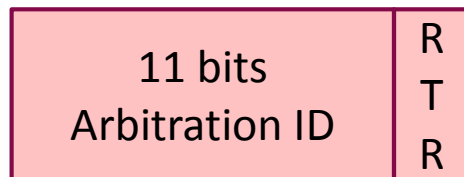
La trame CAN – Trame de données

- ▶ 1 bit de début de trame (Start Of Frame, SOF).
- ▶ Début d'échange après repos du bus.
- ▶ Tous les nœuds doivent se synchroniser sur le flan avant la transition du bit de départ.

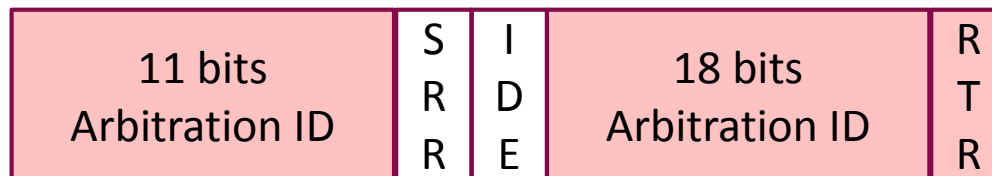


La trame CAN – Trame de données

- ▶ le champ d'arbitrage : 11 bits (ID) + 1 bit (RTR) en standard et, 29 bits (ID) + 2 bits (SRR/IDE) + 1 bit (RTR) en étendue.



CAN 2.0 A



CAN 2.0 B

La signification de certains bits :

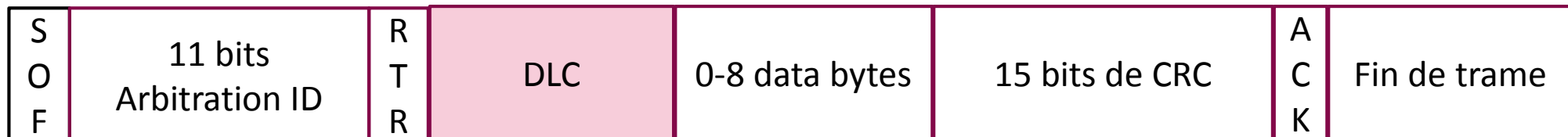
- RTR : Remote Transmission Request bit,
- SRR : Substitute Remote Request bit,
- IDE : Identifier Extension bit.

Le bit RTR (Remote Transmission Request) permet de déterminer le type de trame :

- RTR = état dominant : trame de données
- RTR = état récessif : trame de requête

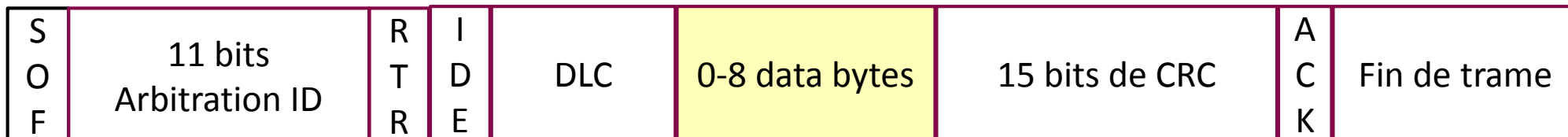
La trame CAN – Trame de données

- ▶ Champ de commande : 6 bits.
- ▶ Bits de réserve : 2 bits permettent d'assurer de futures compatibilités ascendantes (notamment avec CAN 2.0 B).
- ▶ DLC (Data Length Code) : 4 bits Nombre de données contenues dans le champ de données.



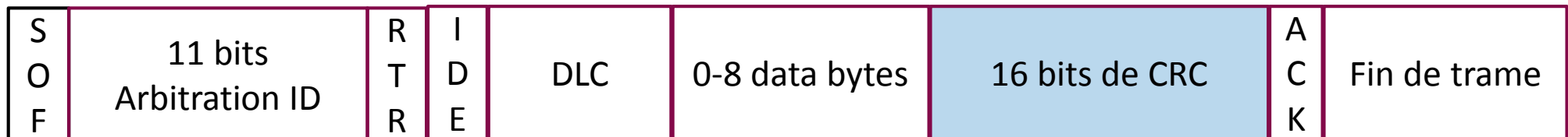
La trame CAN – Trame de données

- ▶ Champ de données : 0 à 64 bits :
 - ▶ Les données transmises (e.g., Valeurs de capteur).
 - ▶ Minimum : 0 octets.
 - ▶ Maximum : 8 octets.
 - ▶ Transmis avec le bit de poids fort en tête.



La trame CAN – Trame de données

- ▶ Champ de CRC : 16 bits.
 - ▶ Cyclic Redudency Code : Code à Redodance Cyclique.
 - ▶ Séquence CRC : 15 bits.
 - ▶ Délimiteur CRC : 1 bits.



La trame CAN – Trame de données

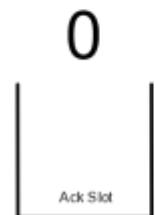
- ▶ Champ d'acquittement : 2 bits.

- ▶ **ACK slot** : À chaque fois qu'un récepteur a correctement reçu un message valide, il superpose lors du time slot du ACK slot (récessif) un bit dominant.
- ▶ **ACK delimiter** : Doit toujours être récessif : lorsqu'un message a été correctement reçu, le bit ACK slot (dominant) est entouré par deux bits récessifs (CRC delimiter et ACK delimiter).

Émetteur



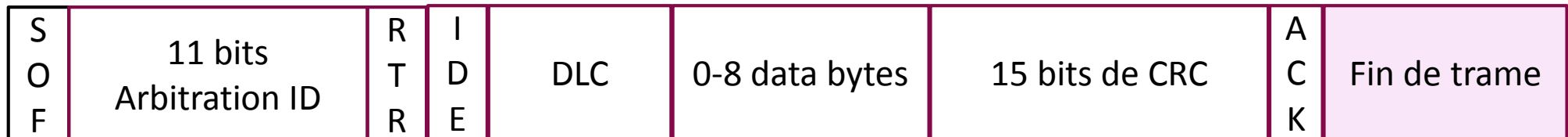
Récepteur



S O F	11 bits Arbitration ID	R T R	I D E	DLC	0-8 data bytes	15 bits de CRC	A C K	Fin de trame
-------------	---------------------------	-------------	-------------	-----	----------------	----------------	-------------	--------------

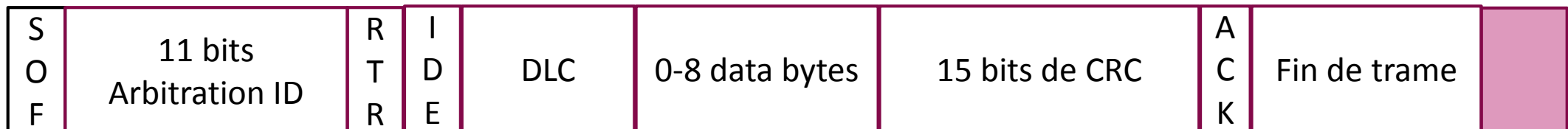
La trame CAN – Trame de données

- ▶ Fin de trame : 7 bits
 - ▶ Séquence de 7 bits récessifs successifs
 - ▶ Structure fixe : les logiques de stuffing et de destuffing doivent être désactivés pendant la séquence de fin de trame



La trame CAN – Trame de données

- ▶ Espace intertrame : 3 bits
 - ▶ 3 bits récessifs
 - ▶ Aucun nœud n'est autorisé à démarrer la transmission de la trame de données ou de la trame de requête.
- ▶ Une seule action permise :
 - ▶ signaler une condition de surcharge (overload)



La trame CAN – Trame de requête

- Demande d'information à un autre nœud.



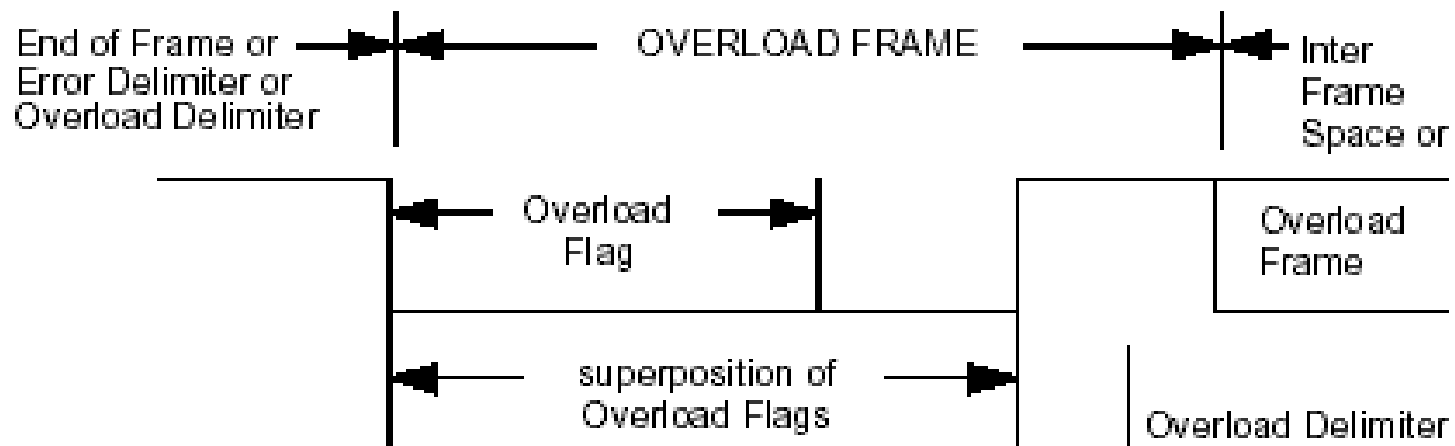
Id du nœud cible.

Pas de champ de
données

La trame CAN – Trame de surcharge

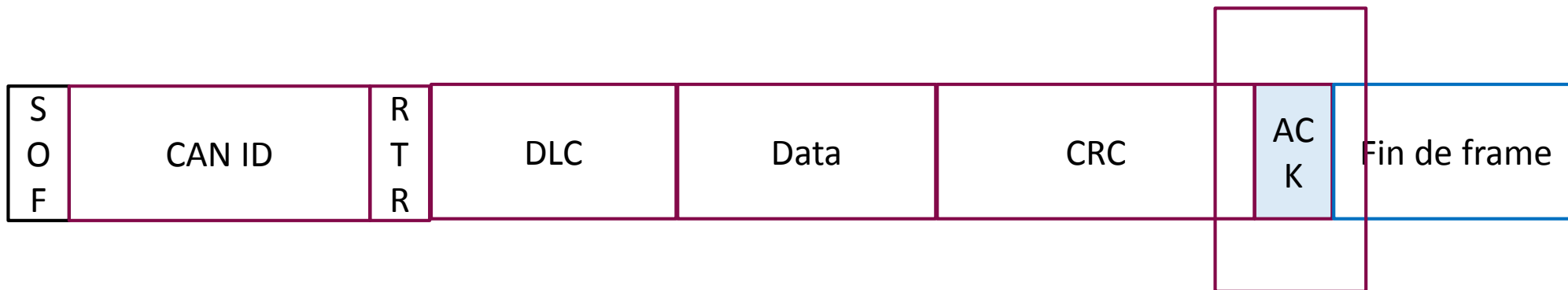
- ▶ La trame de surcharge indique aux autres nœuds qu'une station est surchargée. Elle est formée de deux champs :
 - ▶ le drapeau de surcharge (Overload Flag) avec six bits dominants,
 - ▶ le délimiteur de surcharge (Overload Delimiter) avec huit bits récessifs,

- ▶ Une trame de surcharge est émise sur le bus si
 - ▶ un bit dominant est détecté durant la période d'inter trame.
 - ▶ un récepteur n'est pas prêt pour la réception d'une nouvelle trame de donnée ou de requête (retard sur le traitement des informations circulant sur le bus).



La trame CAN – Trame d'erreur

- ▶ Bit ACK slot dominant : absence d'erreur.
- ▶ Bit ACK Slot récessif : le nœud qui détecte une erreur doit émettre une trame d'erreur pour prévenir l'émetteur qu'une erreur a été détectée.



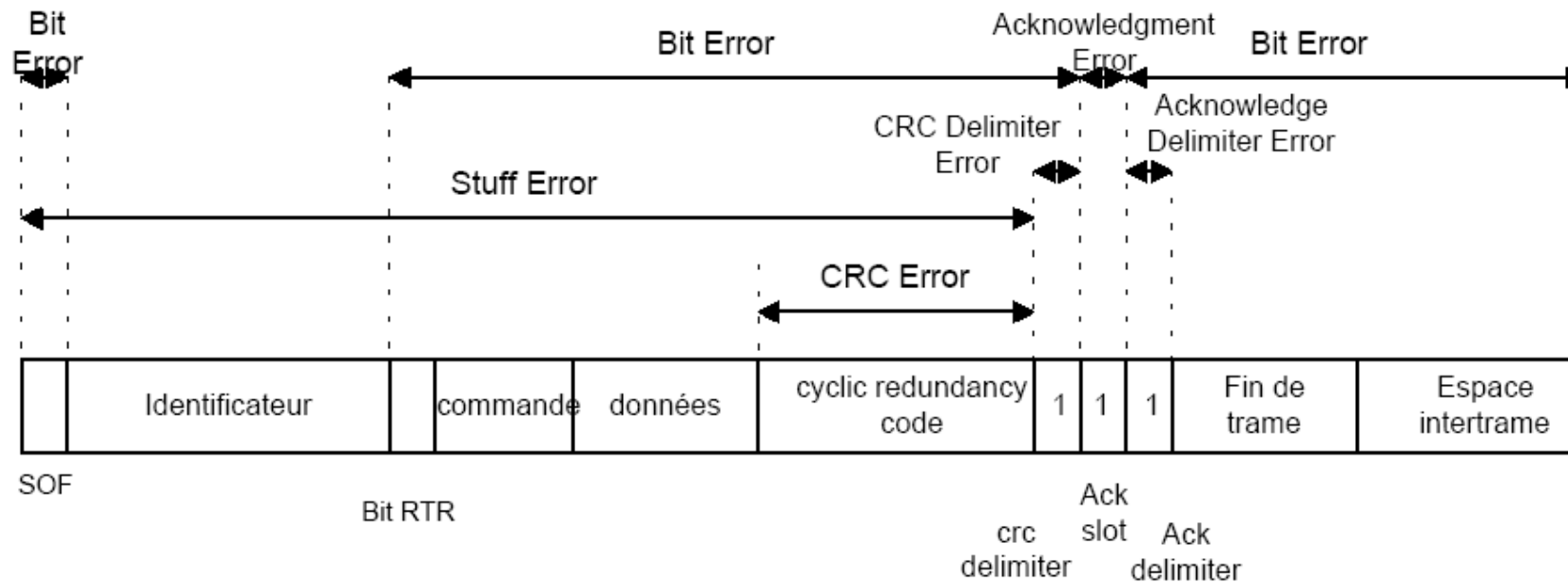
La trame CAN – Trame d'erreur

- ▶ Le CAN implémente cinq mécanismes de détection des erreurs, 2 au niveau bits (le "bit monitoring" et le "bit stuffing").
- ▶ 3 au niveau messages (vérification du CRC, de la forme des trames et de l'acquittement).

La trame CAN – Trame d'erreur

- ▶ Ces cinq types d'erreurs différents qui peuvent être détectée par un nœud sont:
 - ▶ **Bit error:** Un nœud envoyant un bit sur le bus regarde aussi en même temps les bits qu'il reçoit (Bit monitoring). Il considère comme une erreur de bit lorsque le bit envoyé est différent du bit reçu, à l'exception de l'envoi d'un bit récessif durant l'arbitrage (cas de la perte d'arbitrage) ou pendant le ACK Slot (trame acquittée).
 - ▶ **Stuff error:** Le nœud détecte une erreur de stuffing lorsqu'il reçoit 6 bits consécutifs de même valeur dans une partie d'un message qui devrait être codé avec la méthode du bit stuffing.
 - ▶ **CRC error:** Une erreur de CRC est détectée lorsque le CRC calculé par un récepteur est différent de la valeur du CRC contenu dans la trame.
 - ▶ **Form error:** Une "Form error" est détectée lorsqu'un bit qui devrait être à une certaine valeur est à une valeur différente (un délimiteur par exemple).
 - ▶ **ACK error:** Le transmetteur détecte une erreur d'acquiescement lorsqu'il ne reçoit pas de bit dominant pendant le ACK Slot.

La trame CAN – Trame d'erreur

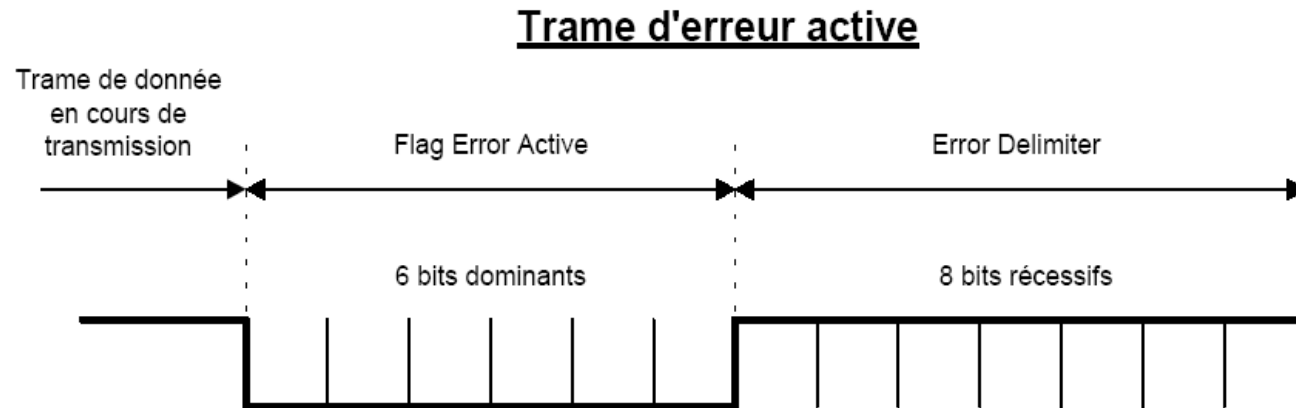


La trame CAN – Trame d'erreur

- ▶ Le contrôle de redondance cyclique (Cyclic Redundancy Check ou CRC) est un mécanisme de contrôle d'intégrité des données puissant mais aussi facile à mettre en œuvre.
- ▶ La plupart des systèmes de contrôle d'erreur au niveau logique sont basés sur un ajout d'information (on parle de « redondance ») permettant de vérifier la validité des données. On appelle somme de contrôle (en anglais checksum) cette information supplémentaire
- ▶ Avec ce mécanisme, on définit un polynôme (appelé polynôme générateur noté $G(x)$) connu à la fois de l'émetteur et du récepteur. On utilise un algorithme sur les bits de la trame pour générer un CRC qui est transmis au récepteur. Il suffit alors au récepteur d'utiliser le polynôme $G(x)$ pour générer un CRC à partir de la trame reçue et le comparer au CRC contenu dans la trame afin de vérifier sa validité.

La trame CAN – Trame d'erreur

- ▶ Une trame d'erreur est constituée de deux parties : La première est formée par la superposition des différents "Error flags" mis par les noeuds du bus. La seconde partie est un délimiteur.
- ▶ Tout nœud qui détecte une erreur la signale par la transmission d'un Error flag. Un Error flag est composé de 6 bits dominants consécutifs violant ainsi la règle du bit stuffing.
- ▶ Ainsi, tous les autres noeuds détectent aussi une erreur et commencent à envoyer un Error flag. La séquence de bits dominants qui existe alors sur le bus est le résultat de la superposition de plusieurs Error flags, et sa longueur varie entre 6 et 12 bits.



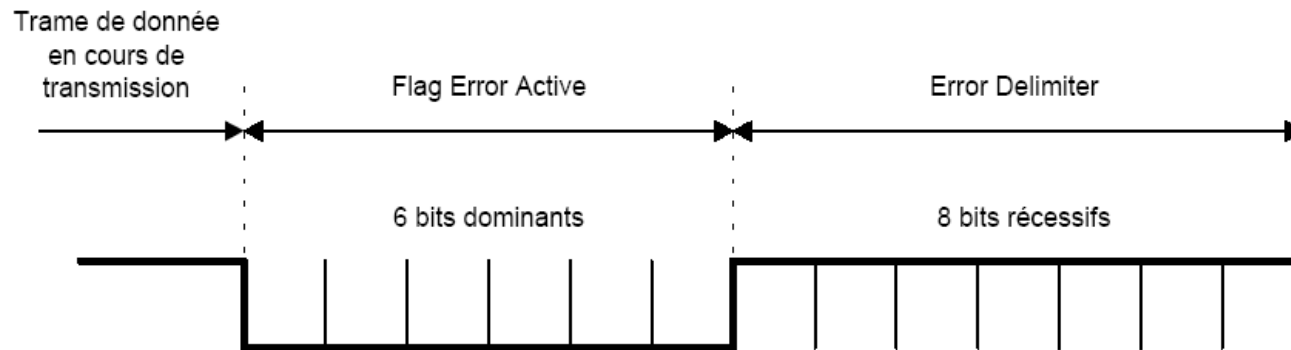
La trame CAN – Trame d'erreur

► Il existe deux types de trames d'erreurs :

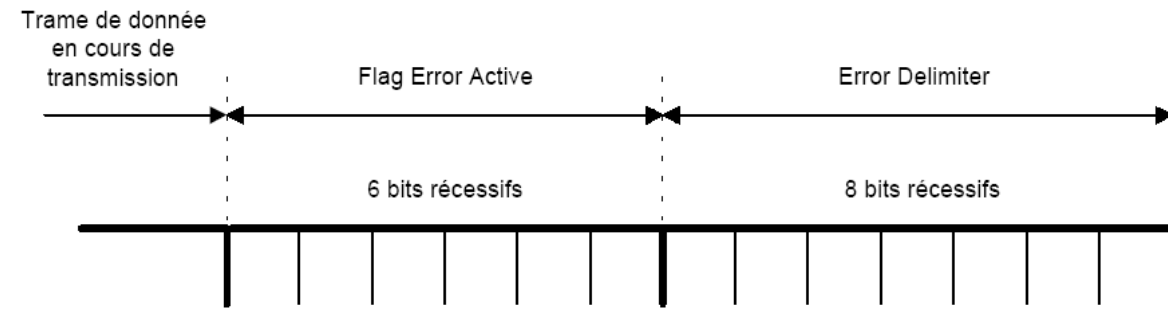
Active error flag : 6 bits dominants consécutifs.

Passive error flag : 6 bits récessifs consécutifs, jusqu'à ce qu'ils soient écrasés par des bits dominants.

Trame d'erreur active



Trame d'erreur passive



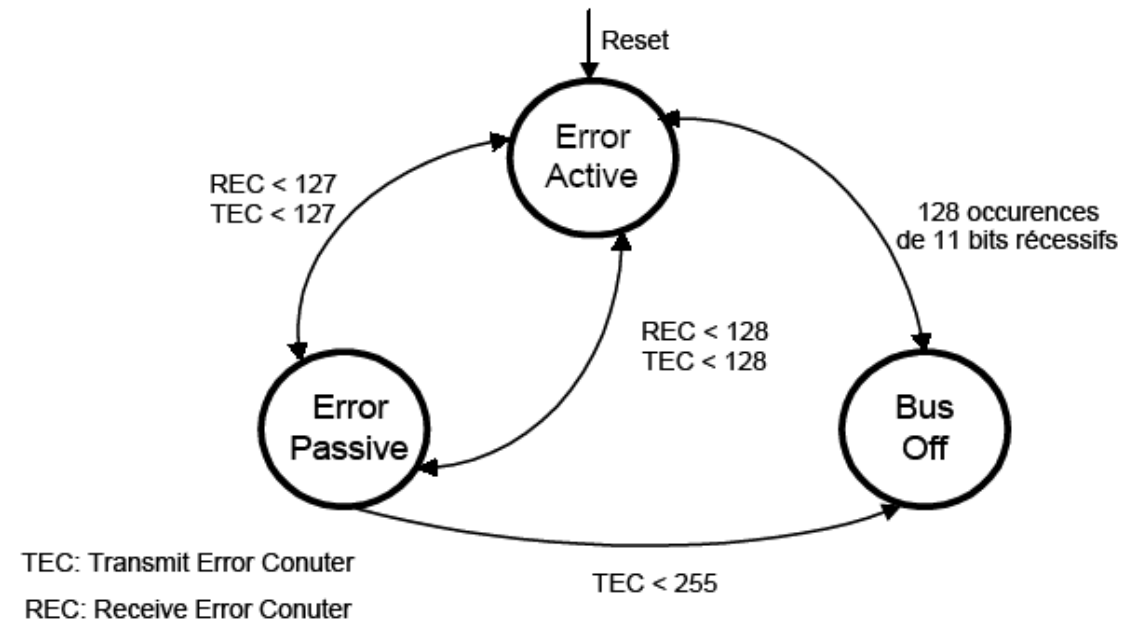
L'Error delimiter est composé de 8 bits récessifs. En fait, après avoir transmis son Error flag, chaque noeuds envoie des bits récessifs et observe le bus jusqu'à ce qu'il détecte un bit récessif, après quoi il envoie encore 7 bits récessifs supplémentaires.

La trame CAN – Trame d'erreur

- ▶ Le confinement des erreurs est un mécanisme qui permet de différencier les erreurs temporaires des erreurs permanentes :
 - ▶ Les erreurs temporaires peuvent être causées par des bugs occasionnels.
 - ▶ Les erreurs permanentes peuvent être due à de mauvaises connections ou à des composants qui sont défectueux. Le système en place permet ainsi de mettre hors communication un nœud défectueux qui sinon aurait pu perturber les autres nœuds.

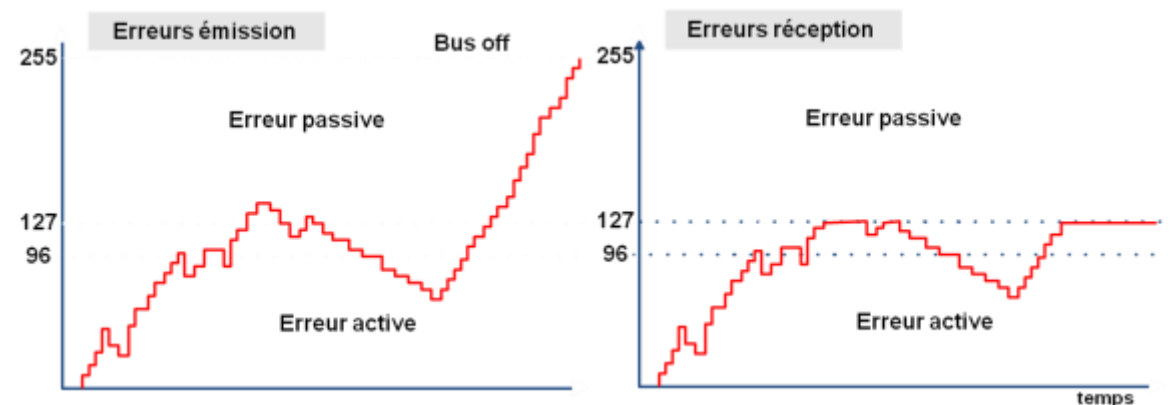
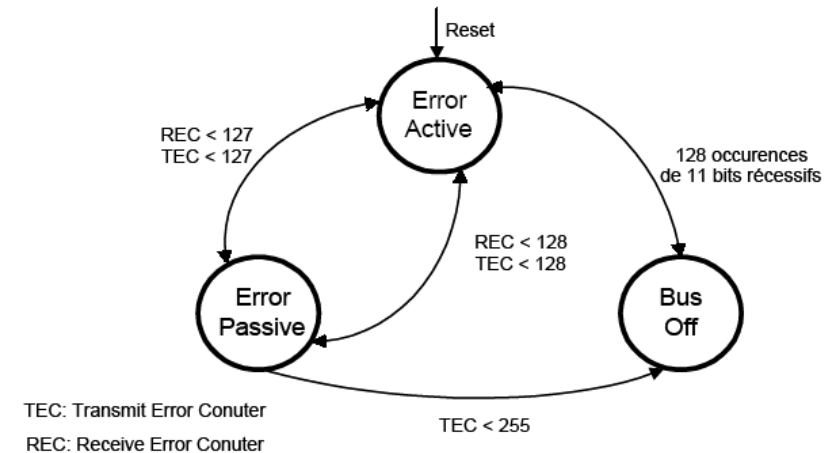
La trame CAN – Trame d'erreur

- ▶ Un nœud en mode d'erreur actif (error-active) peut entrer en communication avec les nœuds du bus. Il transmettra un Active error flag s'il détecte une condition d'erreur.
- ▶ Un nœud en mode d'erreur passif (error-passive) peut entrer en communication avec les nœuds du bus, mais s'il détecte une condition d'erreur sur le bus, il transmettra un Passive error flag. Ce mode indique un nœud à problèmes.
- ▶ Un nœud en mode bus-off n'est pas autorisé à avoir une quelconque influence sur le bus.



La trame CAN – Trame d'erreur

- ▶ Deux compteurs d'erreurs sont implémentés dans chaque noeud : celui des erreurs en transmission (Transmit error count) et celui des erreurs en réception (Receive error count) :
 - ▶ Lorsqu'un récepteur détecte une erreur, son compteur des erreurs en réception est augmenté de 1.
 - ▶ Lorsqu'un transmetteur envoie un Error flag, son compteur d'erreurs en transmission est augmenté de 8.
 - ▶ Après une transmission réussie, compteur d'erreurs en transmission est diminué de 1.
 - ▶ Après une réception réussie, le compteur des erreurs en réception est diminué de 1.



L'arbitrage dans le protocole CAN

- ▶ Le Bus CAN ne dispose pas de maître et permet n'importe quel nœud dans le réseau de faire du broadcasting dans le réseau.
- ▶ Il est impératif ainsi de régler les conflits d'accès.
- ▶ Le concept d'arbitrage permet de régler cette problématique bit à bit de façon non destructif sur la base de l'identifiant d'arbitrage.
- ▶ Grâce au concept de bit récessif et dominant, on garantit un arbitrage sans perte d'information et de temps.

L'arbitrage dans le protocole CAN

