



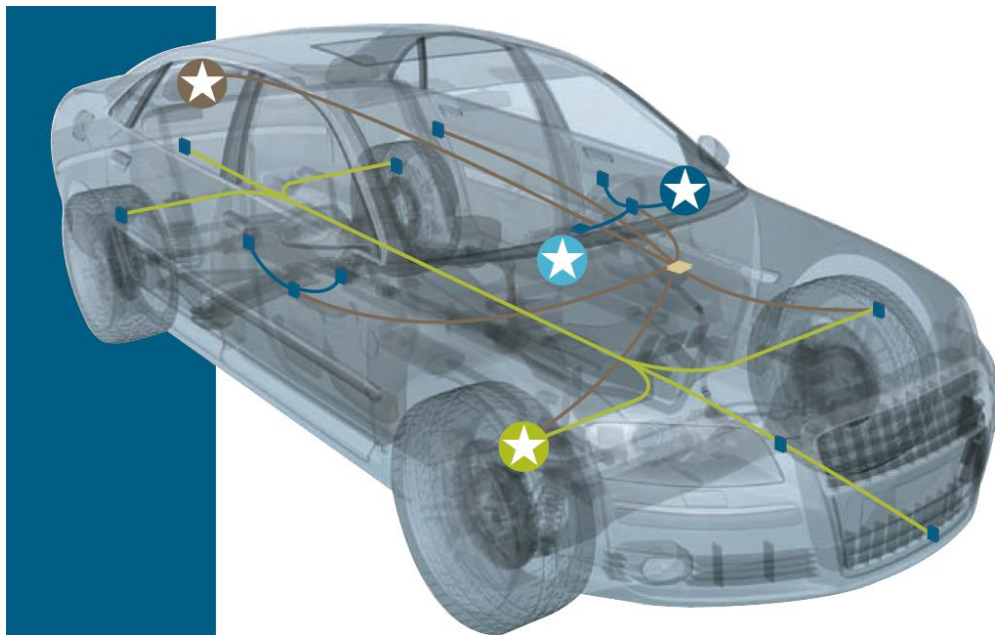
Sommaire

Trente ans de CAN	3
Introduction	3
CAN et modèle OSI	4
La couche Physique	5
Connectique	5
Bit dominant / bit récessif	5
Correspondance avec les niveaux logiques	5
Le Bit Stuffing	5
Signaux CANH et CANL	7
Niveaux de tension	7
Trames CAN	8
Les types de trames	8
Trame standard/étendue	8
Trame de données	8
le Start Of Frame (SOF)	8
Le champ d'arbitrage	8
Le champ de contrôle	9
Le champ de CRC	9
Le champ d'acquiescement	9
Le champ de fin de trame	9
Trame de requête	9
Trames d'erreur	10
Trame d'erreur active	10
Trame d'erreur passive	10
Trame de surcharge	10
Accès au medium, arbitrage et priorité	10
L'arbitrage	11
Priorité	11
Bit et synchro	12
Construction d'un bit	12
La synchronisation	12
Les mode d'erreur	14
Gestion des modes d'erreur	14
Retransmission des messages	15
Validation des messages	15
Retransmission	15

Trente ans de CAN



- 1983 Naissance du projet en interne (Robert BOSCH GmbH)
- 1986 Présentation du CAN au congrès SAE à Detroit
- 1987 Premiers composants CAN (Intel puis Philips)
- 1991 CAN specifications 2.0 publiée par Bosch.
- 1991 CAN Low-Speed devient la norme **ISO1519-2** (standard)
- 1992 MERCEDES utilise CAN sur une classe S
- 1993 CAN High-Speed devient la norme **ISO11898** (CAN 2.0 part A)
- 1995 Amendement de la norme ISO 11898 concernant CAN étendu (CAN 2.0 part B)...



Introduction

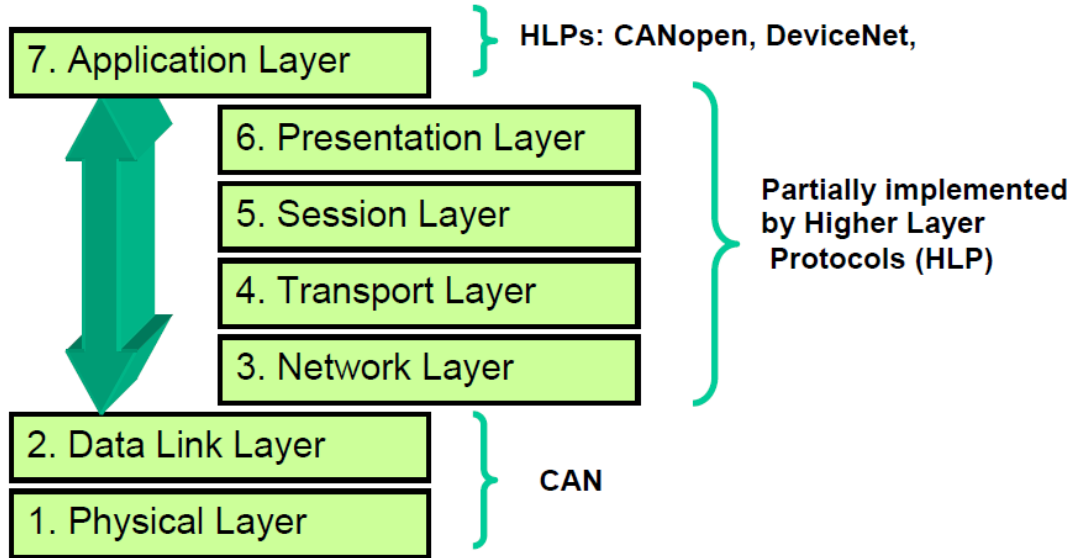
Le protocole CAN (*Control Area Network*) est à classer dans la catégorie des **réseaux de terrain**. Développé à l'origine en vue d'applications dans l'automobile il s'est aussi fait une place dans l'industrie. Il cohabite maintenant avec des évolutions et des standards complémentaires ou concurrents comme LIN, TTCAN, FlexRay...

Il présente les propriétés suivantes :

- transmission série / NRZ,
- fonctionnement distribué multi maître.
- hiérarchisation des messages.
- garantie des temps de latence.
- réception de multiples sources.
- détections et signalisations d'erreurs.
- retransmission automatique des messages altérés.
- distinction des erreurs temporaires ou permanentes au niveau d'un nœud.
- déconnexion automatique des nœuds défectueux...

CAN et modèle OSI

Le bus CAN couvre les deux couches basses du modèle OSI.



	Application	Non implémentées par CAN
	Présentation	
	Session	
	Transport	
	Réseau	
Couche Logique	LLC (<i>Logic Link Control</i>) <ul style="list-style-type: none"> Filtrage d'acceptance des messages Notification de surcharge (<i>overload</i>) Recouvrement des erreurs 	
	MAC (<i>Medium Access Control</i>) <ul style="list-style-type: none"> Encapsulation/Décapsulation des données Codage de trame (<i>Stuffing/Destuffing</i>) Medium Access Management Détection d'erreur Signalisation d'erreur Acquittement Sérialisation/Désérialisation 	Le cœur du standard
Couche Physique	PLS (<i>Physical Signalling</i>) <ul style="list-style-type: none"> Codage/Décodage de bit Bit timing Synchronisation 	Gestion du « nominal bit time »
	PMA (<i>Physical Medium Attachment</i>) <ul style="list-style-type: none"> Caractéristiques Driver/Receiver 	Caractéristiques électriques LS/HS Drivers spécialisés
	MDI (<i>Medium Dependent- Interface</i>) <ul style="list-style-type: none"> Connecteurs 	Propriétaire SubD 9 points

La couche Physique

La couche physique définit comment le signal est transmis entre les différents nœuds.

Elle s'occupe donc :

- de gérer la représentation du bit (codage, timing...).
- de gérer la synchronisation bit.
- de définir les niveaux électriques des signaux.
- de définir le support de transmission.

Attention : le standard défini par Bosch ne fixe pas le type de support physique (simple fil en mode commun, paire différentielle, fibre optique...).

Connectique

Quelques connecteurs standardisés existent :

- DB-9
- M-12
- HE10
- Borniers industriels

Dans les véhicules particuliers on peut accéder au bus sur une prise OBD (On Board Diagnostics).

Sur les cartes de développement la documentation du constructeur est la seule référence valable.

	DB-9	Bornier	M-12	HE10
1	Reserved	GND	Shield (opt.)	Reserved
2	CAN-LOW	CAN-LOW	CAN V+	GND (opt.)
3	GND	Shield (opt.)	GND	CAN-LOW
4	Reserved	CAN-HIGH	CAN-HIGH	CAN-HIGH
5	Shield (opt.)	CAN V+	CAN-LOW	GND
6	GND (opt.)			CAN V+
7	CAN-HIGH			Reserved
8	Reserved			Reserved
9	CAN V+			Reserved
10				Reserved

Bit dominant / bit récessif

Sur un bus CAN les niveaux électriques sont de deux natures : Dominant et Récessif.

Lorsque deux nœuds émettent deux niveaux différents sur le bus c'est le niveau dominant qui sera présent. Le niveau dominant écrase le niveau récessif.

Correspondance avec les niveaux logiques

Sur un bus CAN un bit **dominant** représente un **0** logique et un bit **récessif** un **1** logique

Le Bit Stuffing

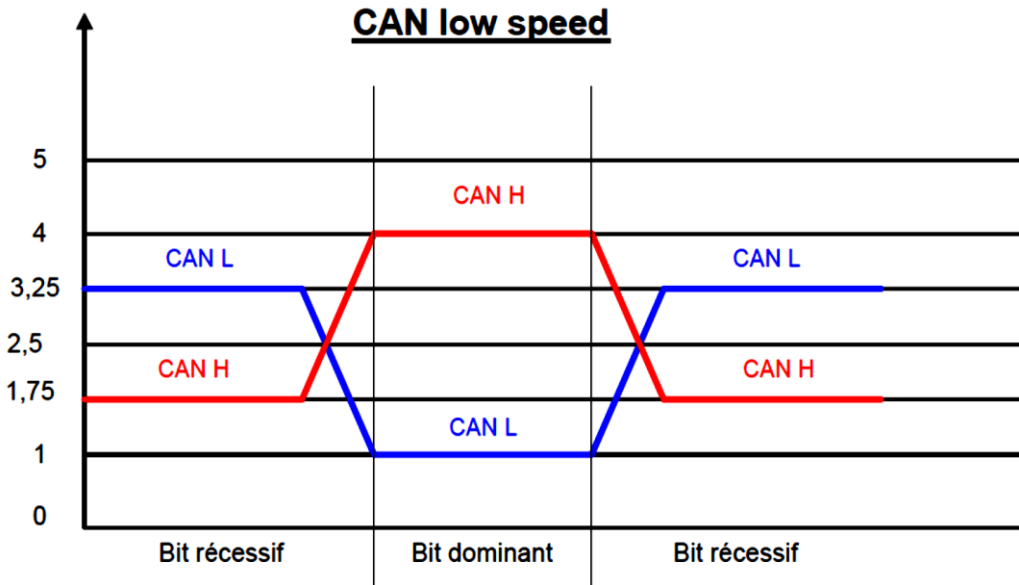
Afin de permettre aux horloges des différents nœuds de se synchroniser régulièrement le **Bit Stuffing** garantit la présence de transitions sur le bus même lors de l'émission d'un grand nombre de bits consécutifs au même niveau. Cette méthode consiste à insérer un bit de niveau opposé dès que l'on a émis 5 bits de même niveau consécutifs.

Signaux CANH et CANL

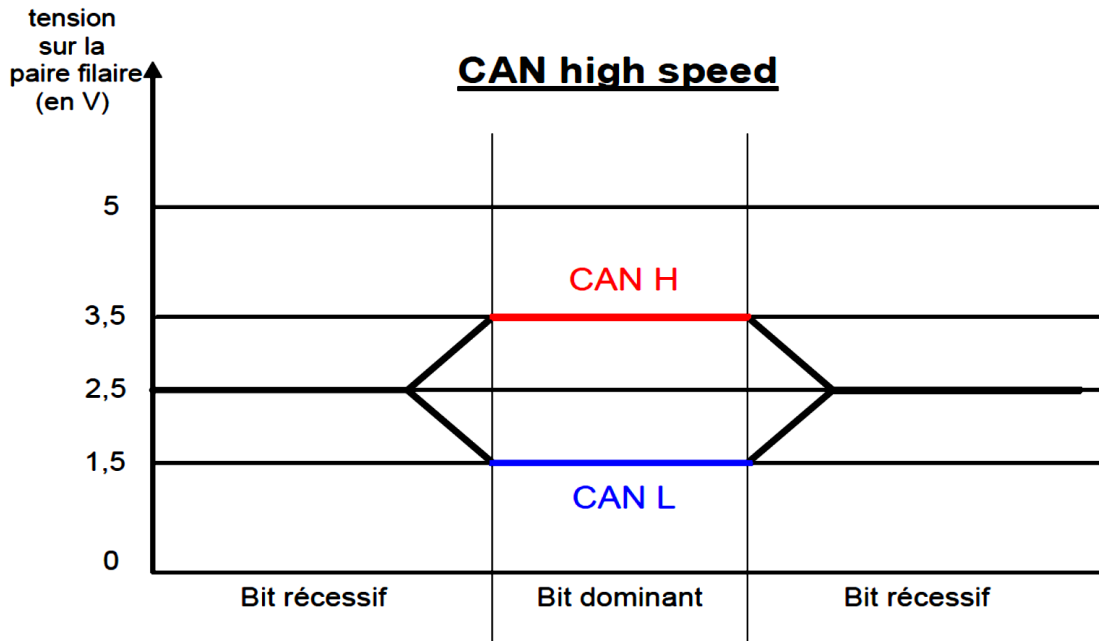
Le bus CAN met en œuvre une transmission différentielle sur deux signaux nommés CANH et CANL.

Niveaux de tension

CAN LS



CAN HS - ISO 11898-3



Trames CAN

L'unité d'échange d'informations du protocole CAN est la trame.

Les types de trames

Il existe quatre types de trames CAN :

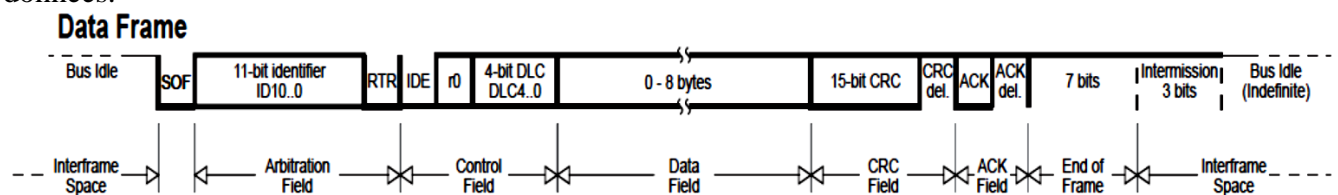
- Trame de données (**data**),
- Trame de requete (**remote**),
- Trame d'erreur (**error**),
- Trame de surcharge (**overload**).

Trame standard/étendue

Deux types de trames peuvent circuler sur un bus CAN : les trames **Standard** et les trames **Étendues**. Elles ne diffèrent l'une de l'autre que par la longueur de l'identificateur (identificateur de 11 bits pour les trames standards, de 29 bits pour les trames étendues). Les trames étendues sont identifiées par le bit IDE qui est au niveau **dominant** pour une trame **standard** et **récessif** pour une trame **étendue**.

Trame de données

La trame de données constitue la partie utile du protocole. C'est elle qui transporte les données.



le Start Of Frame (SOF)

Il est simplement composé d'un bit dominant.

Le champ d'arbitrage

Dans une trame standard, le champ d'arbitrage est composé des 11 bits de l'identificateur (figure 4) et d'un bit de RTR (*Remote Transmission Request*) qui est dominant pour une trame de données et récessif pour une trame de requête.

L'identificateur

L'identificateur est transmis oids forts en tête de ID₁₀ à ID₀.

Pour des raisons de compatibilité avec des anciens circuits, les 4 derniers bits de l'identificateur (ID₃ à ID₀) ne sont pas utilisés et les 7 bits les plus significatifs (de ID₁₀ à ID₄) ne doivent pas tous être récessifs ce qui réduit le nombre de combinaisons possibles sur les systèmes mettant en œuvres ces contrôleurs.

Il n'y a pas de restriction d'ID sur les contrôleurs modernes..

Le bit de requête (RTR)

L'identificateur est suivi du bit de requête qui est dominant (à 0) pour une trame de données et récessif (à 1) pour une trame de requête.

Le champ de contrôle**Les bits réservés (IDE/r0)**

Les deux premiers bits du champ de contrôle sont des bits réservés (pour le futur) dont le premier est d'ores et déjà utilisé pour désigner les trames étendues.

Le bit IDE (r1) doit être dominant pour une trame standard (à 0).

Le code de longueur (DLC)

Les quatre bits suivants indique la longueur du champ de données. Il est transmis poids fort en tête et ne doit pas excéder 8. Les données transmises ne peuvent donc pas excéder 8 octets bien que l'on puisse coder des valeurs allant jusqu'à 15 dans ce champ.

Le champ de données

Ce champs contient autant d'octets qu'indiqué dans le champ de contrôle.

Le champ de CRC

Il est long de 16bits. Il est constitué d'un CRC sur 15 bits suivi d'un délimiteur constitué d'un bit récessif (à 1).

Il est calculé depuis le début de la trame (SOF) 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)

Le CRC est calculé par le contrôleur de bus. La méthode de calcul est donnée en annexe.

Le champ d'acquiescement

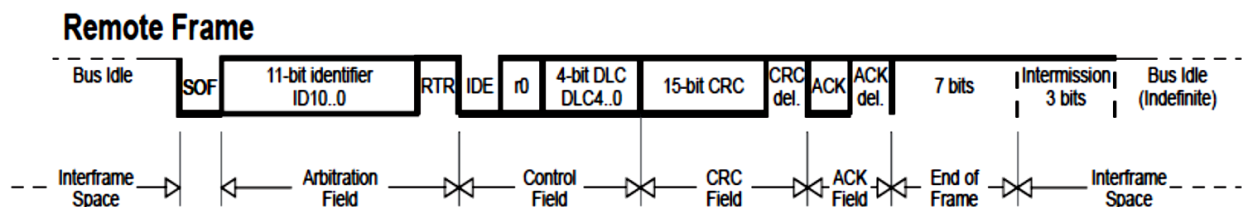
Le champ d'acquiescement est composé de deux bits.

L'émetteur de la trame laisse le bus libre (récessif) durant cas deux bits.

Tout nœud qui a reçu et contrôlé la trame comme valide doit émettre un bit dominant (à 0) durant le premier bit et laisser le bus libre durant le deuxième bit.

Le champ de fin de trame

Ce champ de fin de trame est constitué de 7 bits récessifs, ce qui déroge à la règle de *Bit-Stuffing*. On peut le comparer aux bits de stop des trames série asynchrone RS232. La conséquence est d'espacer les trames.

Trame de requête

Une trame de requête est constituée comme une trame de donnée sans le champ de données et dont le bit RTR (dans le champ d'arbitrage) est récessif (à 1).

On notera que le champ DLC devrait indiquer le nombre d'octets à transmettre dans la réponse.

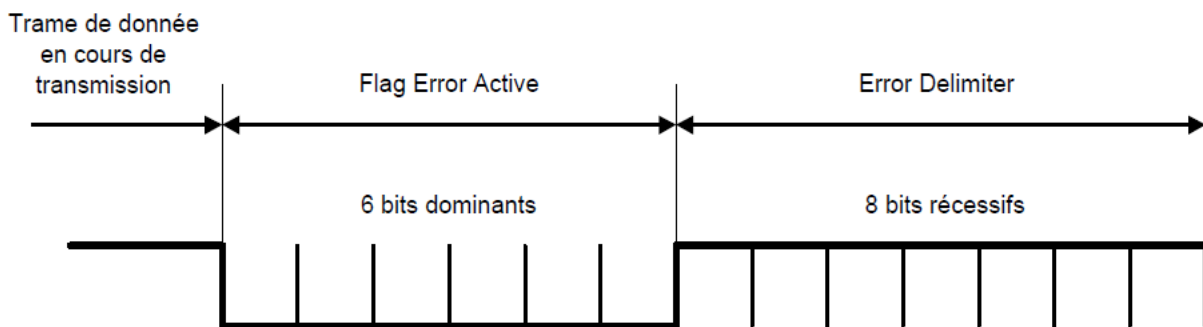
Trames d'erreur

Trame d'erreur active

C'est une trame émise par un nœud qui est en mode « Error-Active ».

Elle est formée de six bits dominant consécutifs suivis de huit bits récessifs pour le délimiteur. Par construction, la trame d'erreur brise la règle du Bit-Stuffing. Les autres récepteurs vont donc se mettre à émettre des trames d'erreurs actives (s'ils sont en mode d'erreur active) à la fin du drapeau de la première station qui a émis la trame d'erreur. Toutes les stations vont donc émettre à tour de rôle la trame d'erreur.

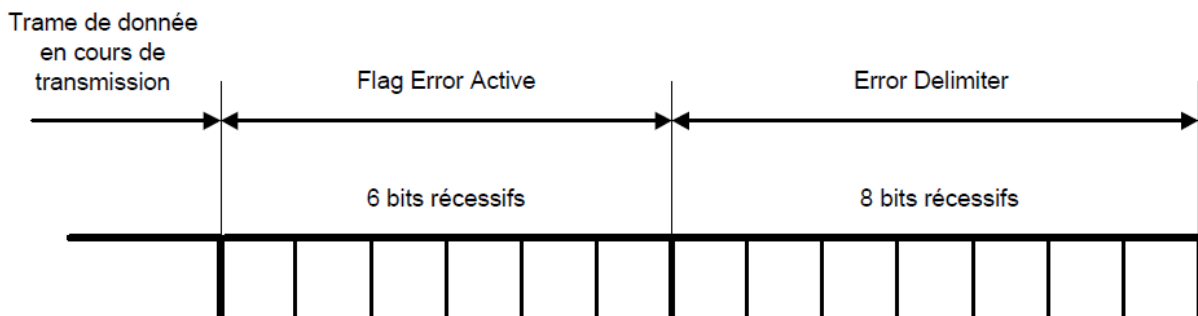
La norme limite le nombre de bits dominant consécutifs à 12 bits. Toute émission devra cesser après le 12^{ème} bit dominant.



Trame d'erreur passive

C'est une trame émise par un nœud qui est en mode « Error-Passive ».

Elle est formée de six bits récessifs suivi de huit bits récessifs soit 16bits récessifs.



Trame de surcharge

Elle sert à un nœud à indiquer qu'un nœud a besoin de temps pour traiter les données. Cela permet donc d'espacer les trames.

Accès au medium, arbitrage et priorité

Le bus CAN met en œuvre une méthode d'accès au medium sur le modèle CSMA/BA : Carrier Sense Multiple Acces / Bitwise Arbitration.

En cas d'accès au bus concurrentiel, le CSMA/BA organise les priorités entre émetteurs de façon à éviter les collisions et donc la réémission superflue de trames. En cas d'échec de l'arbitrage, la détection de collision est aussi à l'œuvre.

L'arbitrage

En cas d'émission simultanée, l'arbitrage permet à chaque nœud en cours d'émission de décider si il peut continuer à émettre ou non.

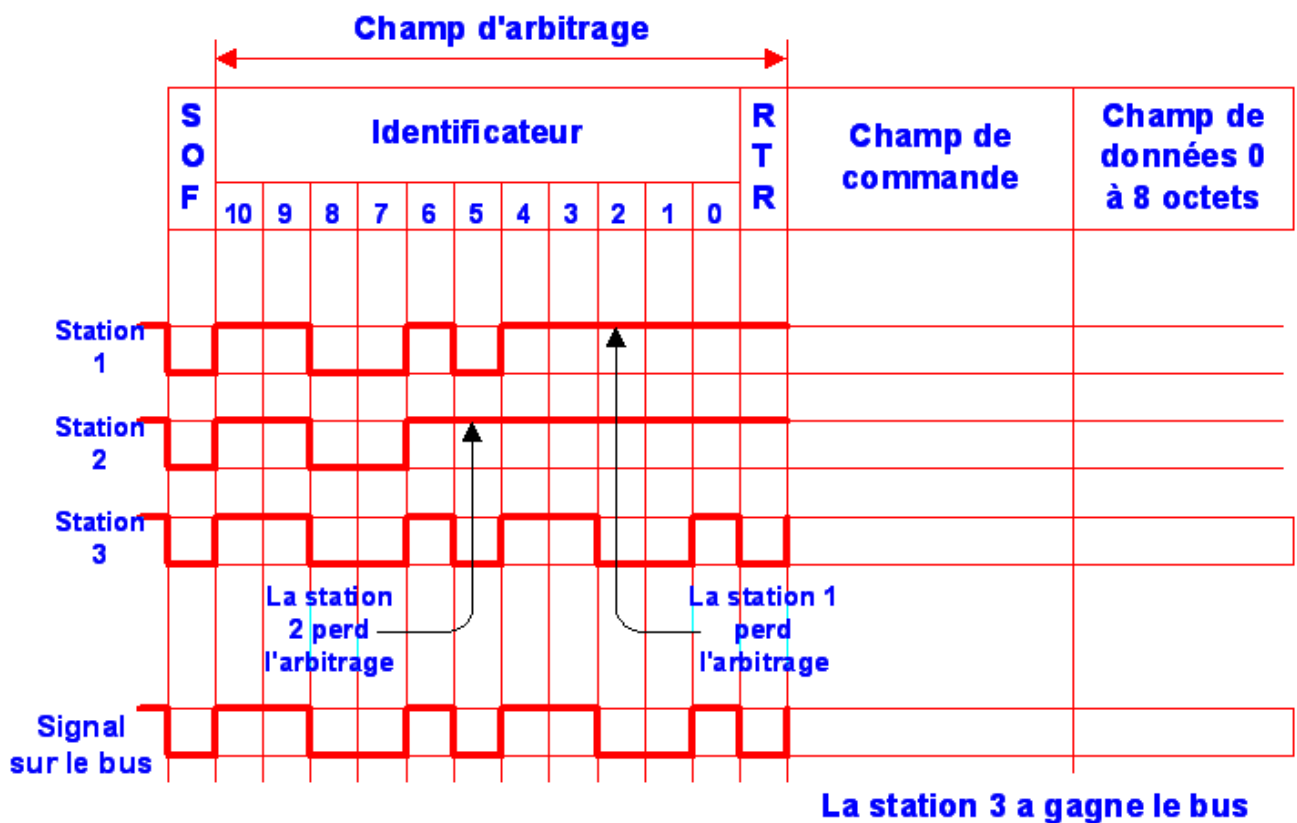
L'arbitrage n'a lieu que sur le « champ d'arbitrage » composé de l'ID et du bit RTR (les douze premiers bits).

Durant l'émission chaque émetteur compare le niveau qu'il est en train d'émettre avec le niveau réellement observé sur le bus. Il cesse l'émission dès qu'il détecte une différence.

Exemple :

Trois nœuds ayant pour identificateurs 0x65F, 0x67F, 0x652, tentent d'accéder au bus simultanément :

- La station 2 émet un ID avec le bit 5 à 1 alors qu'elle observe un 0 sur le bus : elle perd l'arbitrage,
- La station 1 émet un ID avec le bit 2 à 1 alors qu'elle observe un 0 sur le bus : elle perd l'arbitrage,
- La station 3 arrive à émettre l'intégralité de son ID+RTR : elle garde l'accès au bus.



Priorité

Les règles d'arbitrage et le fait qu'un niveau dominant représente un 0 logique ont pour conséquences que :

- un nœud ayant un ID de valeur plus faible qu'un autre gagnera l'arbitrage,
- une trame de données est prioritaire sur une trame de requête.

Bit et synchro

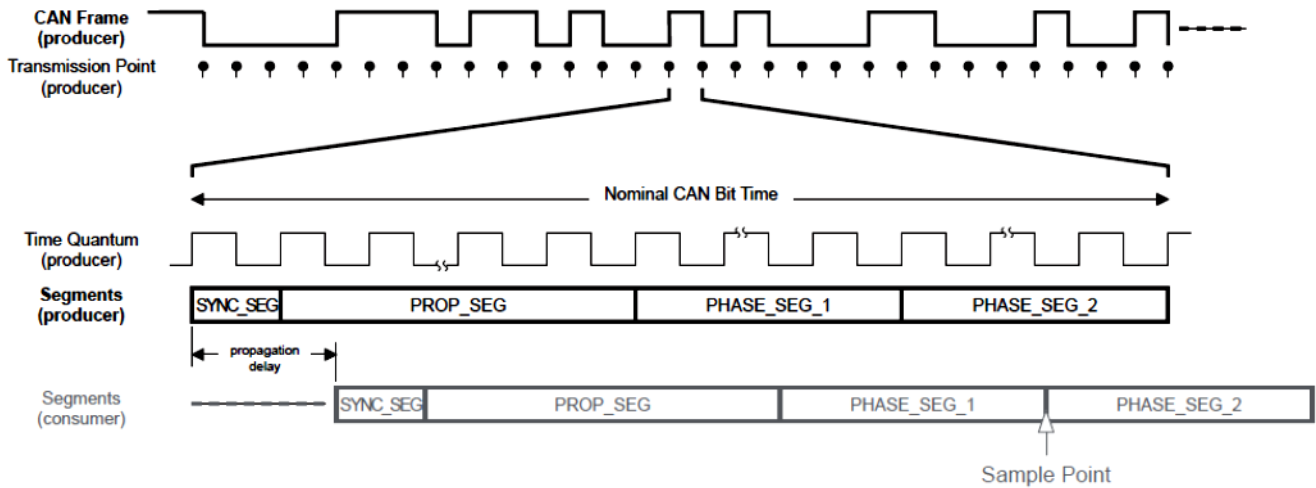
Construction d'un bit

La construction d'un bit (un temps de bit en fait) est basée sur la période d'une horloge de fréquence plus élevée : le *Time Quantum* (Tq).

Le temps de bit nominal (*Nominal Bit Time*) est constitué par la concaténation de quatre phases (*Segments*) basées sur le Time Quantum :

- Le segment de synchronisation (*Sync Seg*),
- Le segment de propagation (*Prop Seg*),
- Les segments de phase #1 et #2 (*Phase Seg 1* et *Phase Seg 2*).

La durée de chacune des phases est un multiple de Tq.



Segment	Durée (Tq)	
SYNC	1	Fixe. Les fronts sont supposés intervenir ici
PROP	1..8	2 x (tps propagation total + délais électronique)
PHASE 1	1..8	Peut être allongé
PHASE 2	2..8	Peut être raccourci
Total		

La synchronisation

Le problème de la synchronisation des horloges est un problème commun à toutes les communications asynchrones.

Deux grands problèmes sont à prendre en compte :

- la différence de fréquence entre les horloges locales,
- les délais dus au temps de propagation du signal sur la ligne de transmission.

Dans le cas d'une communication multi-maîtres comme le bus CAN la prise en compte du temps de propagation est primordiale lorsque celui-ci est significatif par rapport au temps de bit.

La contrainte de la précision des horloges est en relation avec le débit et le prix de revient des oscillateurs : les débits élevés nécessitent des horloges plus précises et stables.

Il découle des règles en vigueur sur le bus que la durée maximum entre deux transitions pouvant être utilisées pour resynchroniser un nœud est de 29 temps de bit.

Sur un bus CAN la synchronisation des nœuds se fait de deux façons : la *Hard-Synchronisation* et la *Resynchronisation*.

La Hard Synchronisation :

Elle consiste à aligner brutalement le *Nominal Bit Time* sur le début du *Sync Seg*.

Elle intervient à deux occasions :

- sur le premier lors de la réception du *Start Of Frame* après une période de repos du bus,
- lors de la réception d'une transition durant le *Sync Seg*.



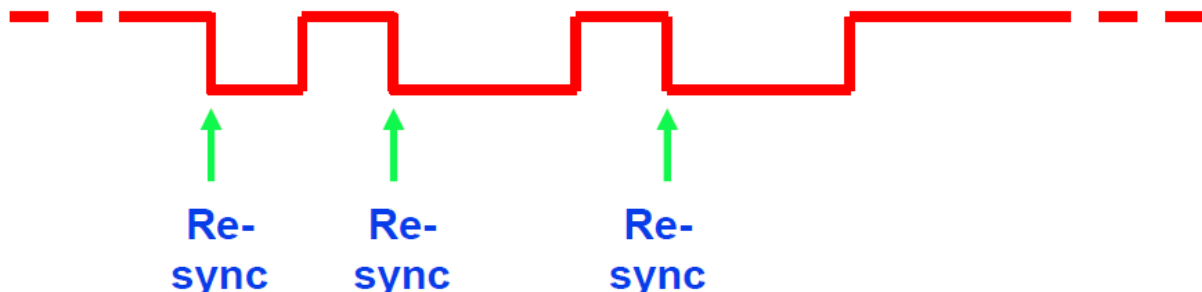
La Resynchronisation :

Nous n'allons pas détailler ici le mécanisme de resynchronisation mais seulement mettre en évidence quelques principes.

Elle consiste à allonger le *Phase Seg1* ou raccourcir le *Phase Seg2* en fonction du décalage entre le point d'échantillonnage et la survenue d'une transition (erreur de phase).

La valeur dont peuvent-être raccourcis ou allongés les *Phase Segs* ne peut excéder une limite appelée **RJW** (*Resynchronisation Jump Width*).

Elle peut intervenir lors des transitions effectives **de récessif à dominant** (sans collision) à l'intérieur de la trame.



C'est un mécanisme prévu pour compenser l'imprécision ou la dérive des horloges mais pas les temps de propagation.

Les mode d'erreur

Tout nœud CAN se comporte conformément à un **Error Mode**.

Celui-ci peut être de trois types :

Error Active

Contrairement à ce que peut suggérer le nom de ce mode c'est le mode normal pour un nœud. Dans ce mode le nœud prend normalement part aux communications sur le bus et peut émettre des trames **Error Active**.

Error Passive

Un nœud en mode **Error Passive** prend part aux communications sur le bus et peut émettre des trames **Error Passive**. Il doit aussi attendre un délai supplémentaire entre trames avant d'émettre une nouvelle trame (champ **Suspend Transmission**).

Bus Off

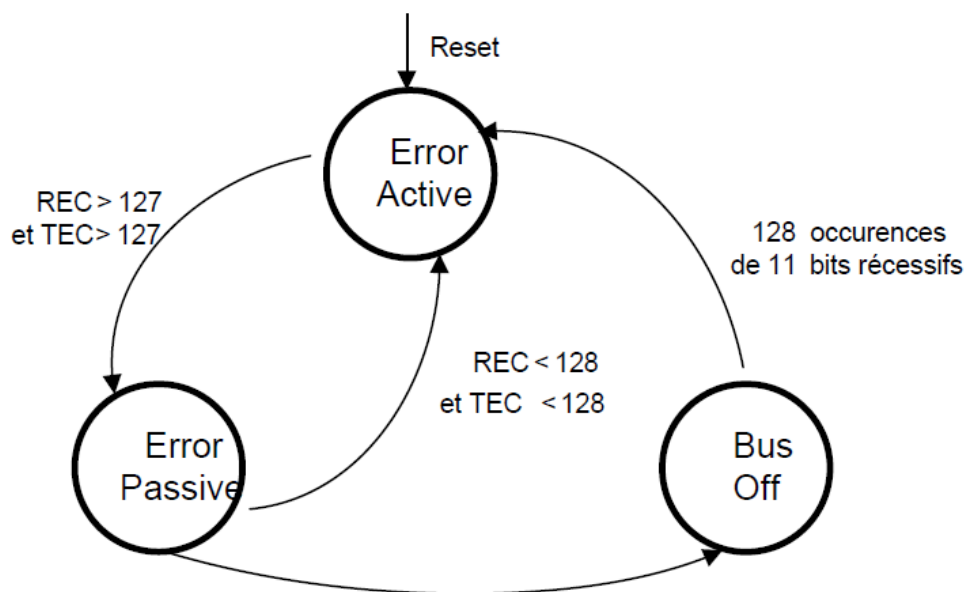
Un nœud en **Bus Off** ne peut avoir aucune influence sur le bus, c'est à dire qu'il doit dévalider (déconnecter) ses drivers de bus (en émission).

Gestion des modes d'erreur

L'évolution de l'état d'un nœud entre les modes d'erreur se fait selon la valeur de deux compteurs :

- **REC** (*Receive Error Counter*) : qui comptabilise les erreurs survenues lors de la réception de trames pour le nœud considéré,
- **TEC** (*Transmit Error Counter*) : qui comptabilise les erreurs survenues lors de l'émission de trames pour le nœud considéré.

Au démarrage les nœuds sont en mode **Error Active**.



TEC: Transmit Error Counter

REC: Receive Error Counter

Il existe des règles d'incrémentation/décrémentation des compteurs en fonction du type d'erreur détectée que nous ne détaillerons pas ici.

Retransmission des messages

Validation des messages

Un message en cours d'émission est considéré comme valide si il ne survient aucune erreur du *SOF* jusqu'au *EOF* (avec quelques différences entre émetteur et récepteur).

Retransmission

Un message corrompu doit être réémis dès que le bus est libre.

Une conséquence de ces deux dernière règles est qu'un contrôleur CAN qui émet seul sur un bus réémet la même trame jusqu'à passer en mode *BUS OFF*.