



Introduction aux Protocoles de communication : Le bus CAN (control Area Network)

Fabrice CAIGNET
LAAS - CNRS
fcaignet@laas.fr



Contenu :

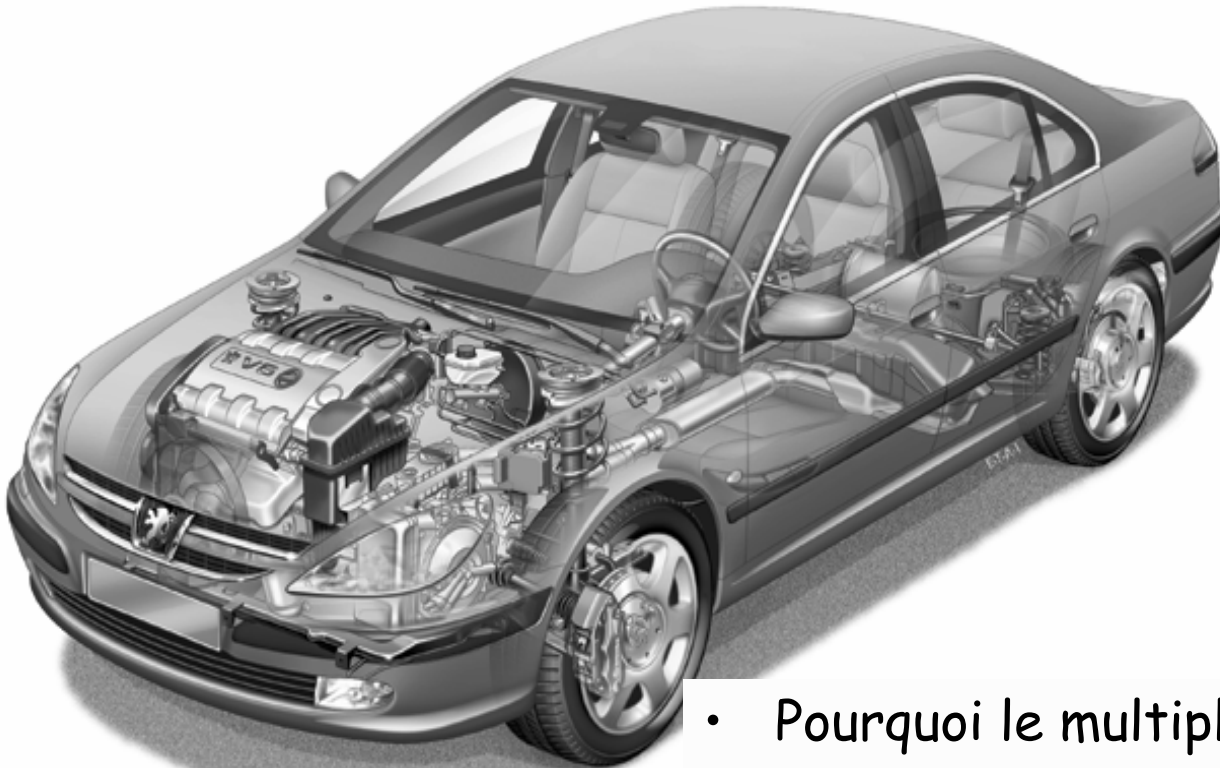
I. Introduction : Notion de multiplexage

II. La notion de bus de communication

III. Le bus CAN

I. Introduction : Notion de multiplexage

Le multiplexage automobile



- Pourquoi le multiplexage ?
- Les principes du multiplexage
- Le protocole CAN



I. Introduction : Notion de multiplexage

Pourquoi le multiplexage ?

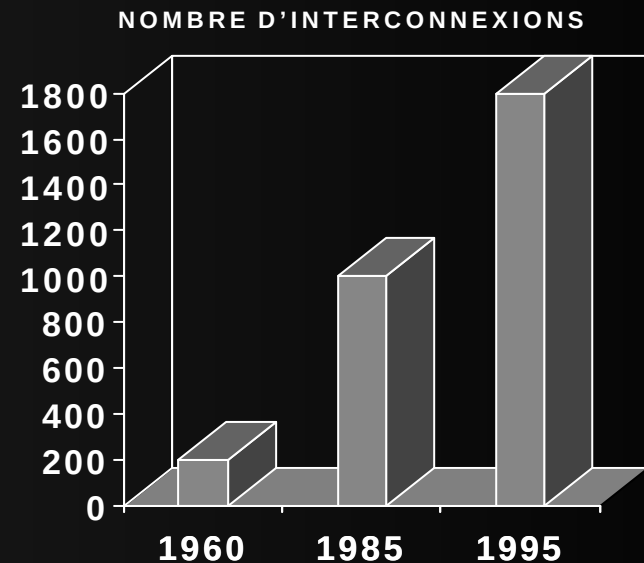
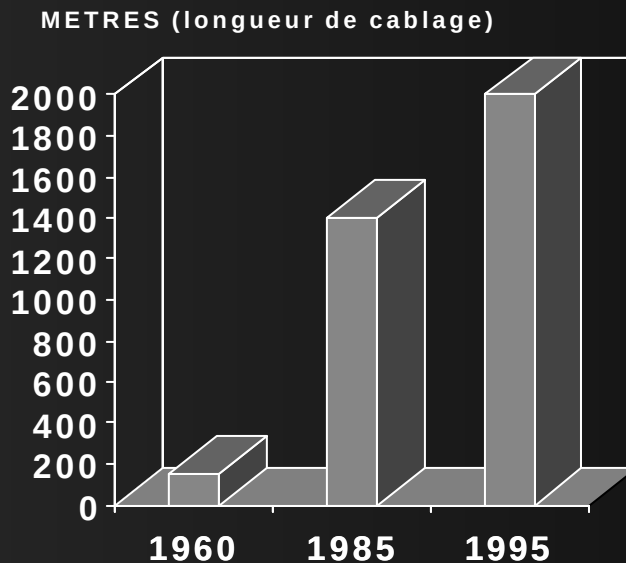
- **L'électronique automobile est en évolution constante :**
 - Exigences de plus en plus sévères en matière de pollution
 - Améliorations en matière de sécurité et de confort
 - Évolution en cours de vie du véhicule (options)
- **D'où une croissance constante, ces 5 dernières années, des fonctions électroniques :**
 - ABS, REF, MSR, ESP, ASR
 - Direction à assistance variable, BVA, suspension pilotée, gestion moteur
 - Airbag, anti-démarrage, clim. régulée, détection du sous-gonflage des roues, aide au stationnement
 - Allumage automatique des feux de croisement, essuie-vitre automatique, correction de site des feux (lampes au Xénon)
 - Allumage automatique des feux de détresse en cas de forte décélération ou de choc
 - Régulation de vitesse avec radar anti-collision, navigation par satellite
 - Et à venir : direction et freins entièrement électrique, guidage du véhicule par rapport aux « bandes blanches », ...

I. Introduction : Notion de multiplexage

Pourquoi le multiplexage ?

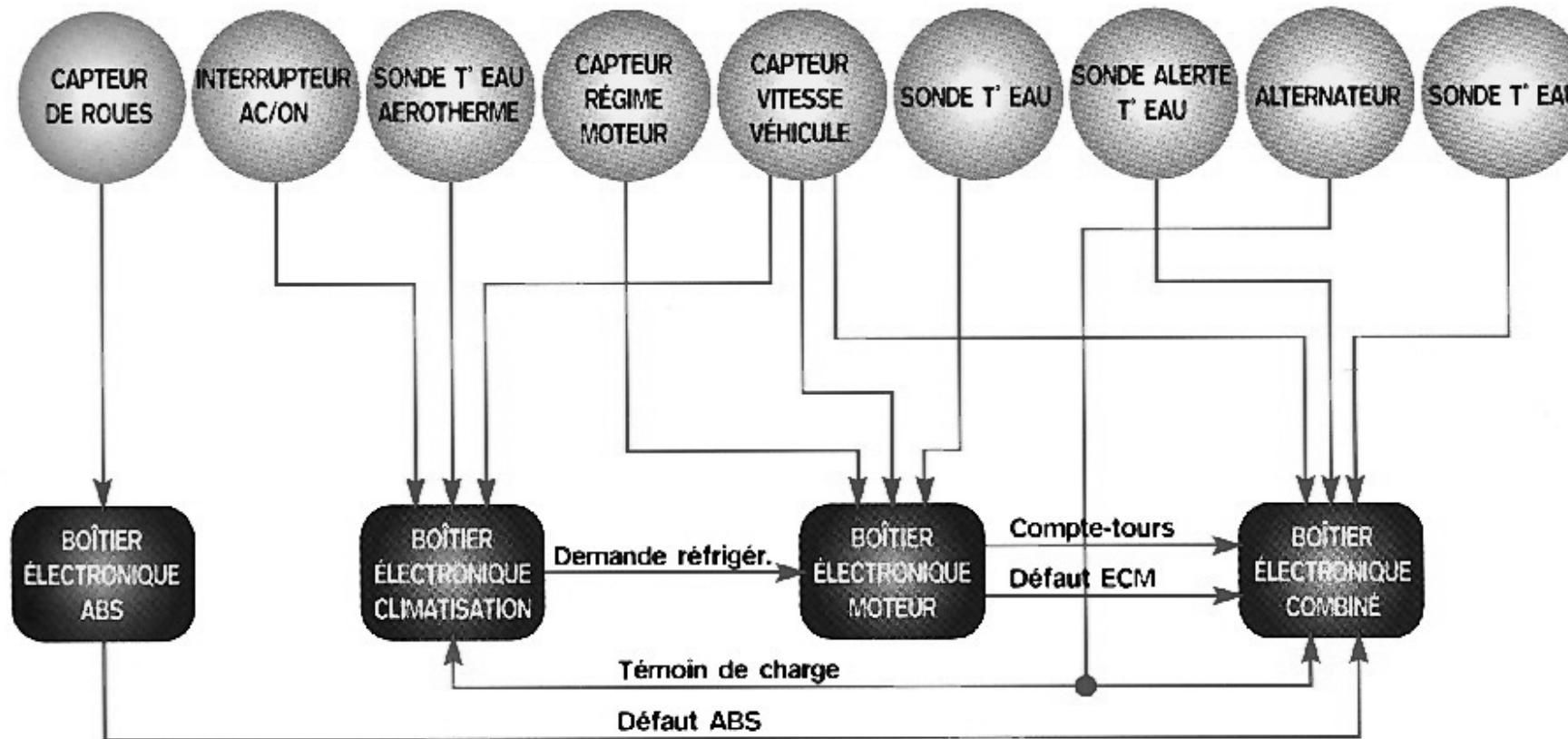
- **Ce renforcement de l'électronique se traduit par :**
 - Une - du nombre de calculateurs
 - Une - du nombre de capteurs
 - Une - des faisceaux de câbles électriques : encombrements, poids et coûts

EVOLUTION DU CABLAGE



I. Introduction : Notion de multiplexage

Pourquoi le multiplexage ? : exemple : câblage classique



Certains capteurs ont des liaisons avec plusieurs calculateurs ou existent en 2 exemplaires en raison de leur localisation. Les liaisons entre boîtiers sont de + en + nombreuses



I. Introduction : Notion de multiplexage

Pourquoi le multiplexage ? Exemple : câblage classique

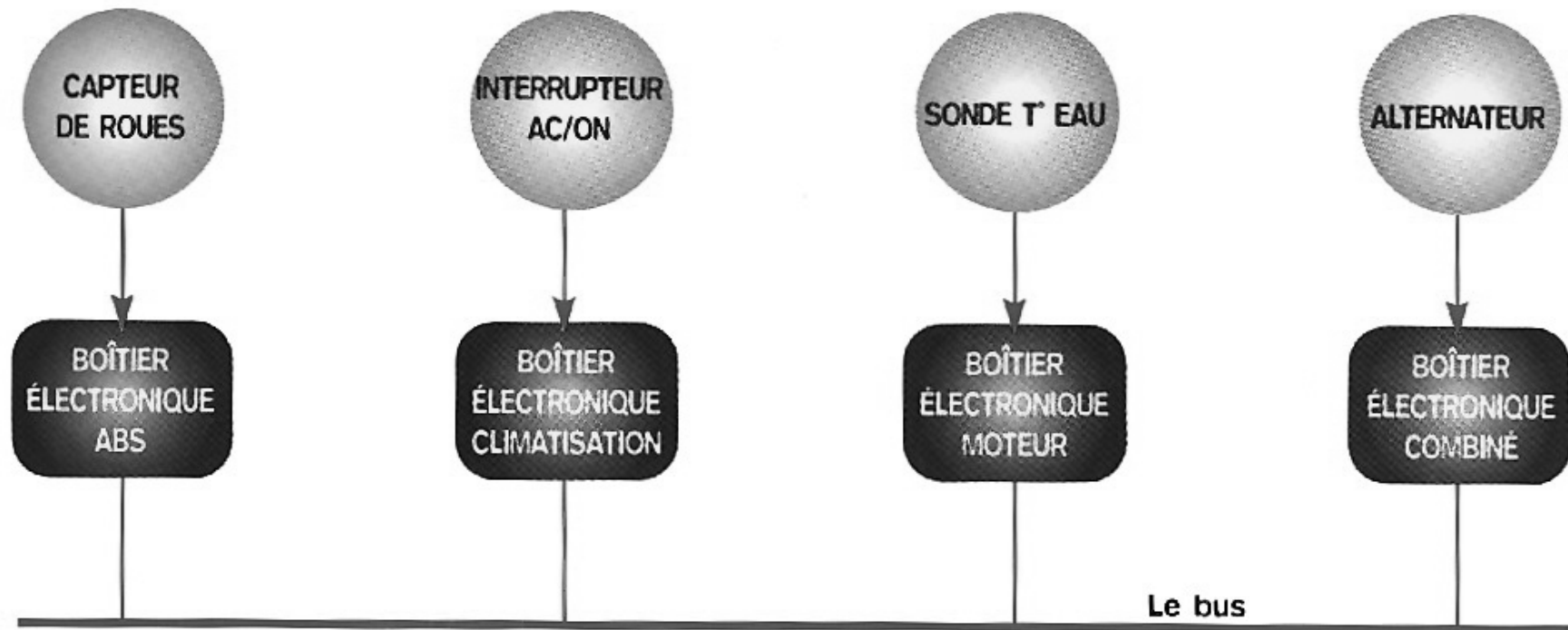
Deux réponses techniques pour limiter « l'inflation » des composants et du volume des câblages :

- **L'intégration** : regrouper plusieurs fonctions dans un seul boîtier (ex : gestion moteur et Boîtier de Servitude Intelligent (BSI))
 - Gestion moteur : injection, allumage, dépollution, refroidissement moteur
 - BSI : fermeture centralisée des portes, alarme, éclairage intérieur, anti-démarrage, essuyage des vitres, gestion des clignotants, ...
 - ESP (contrôle dynamique du véhicule) : ABS, REF, MSR, ASR, ESP

- **Le multiplexage** : faire circuler une multitude d'informations entre divers calculateurs sur un seul canal de transmission appelé le bus (2 fils): le bus CAN

I. Introduction : Notion de multiplexage

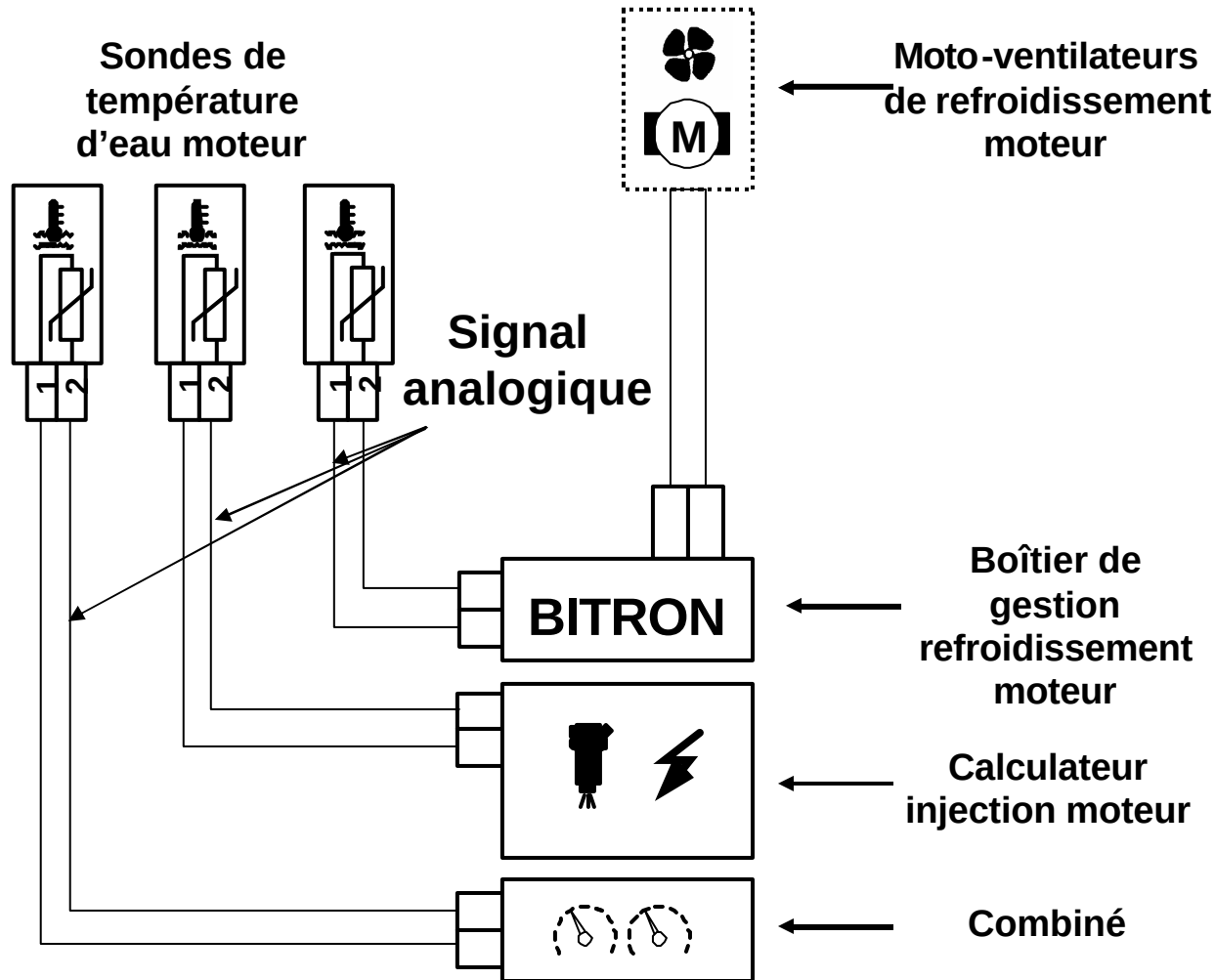
Exemple : câblage avec multiplexage



Diminution du nombre de capteurs et de liaisons entre boîtiers car chacun fournit aux autres, par l'intermédiaire du bus, les infos qu'il reçoit en filaire : c'est le partage des informations

I. Introduction : Notion de multiplexage

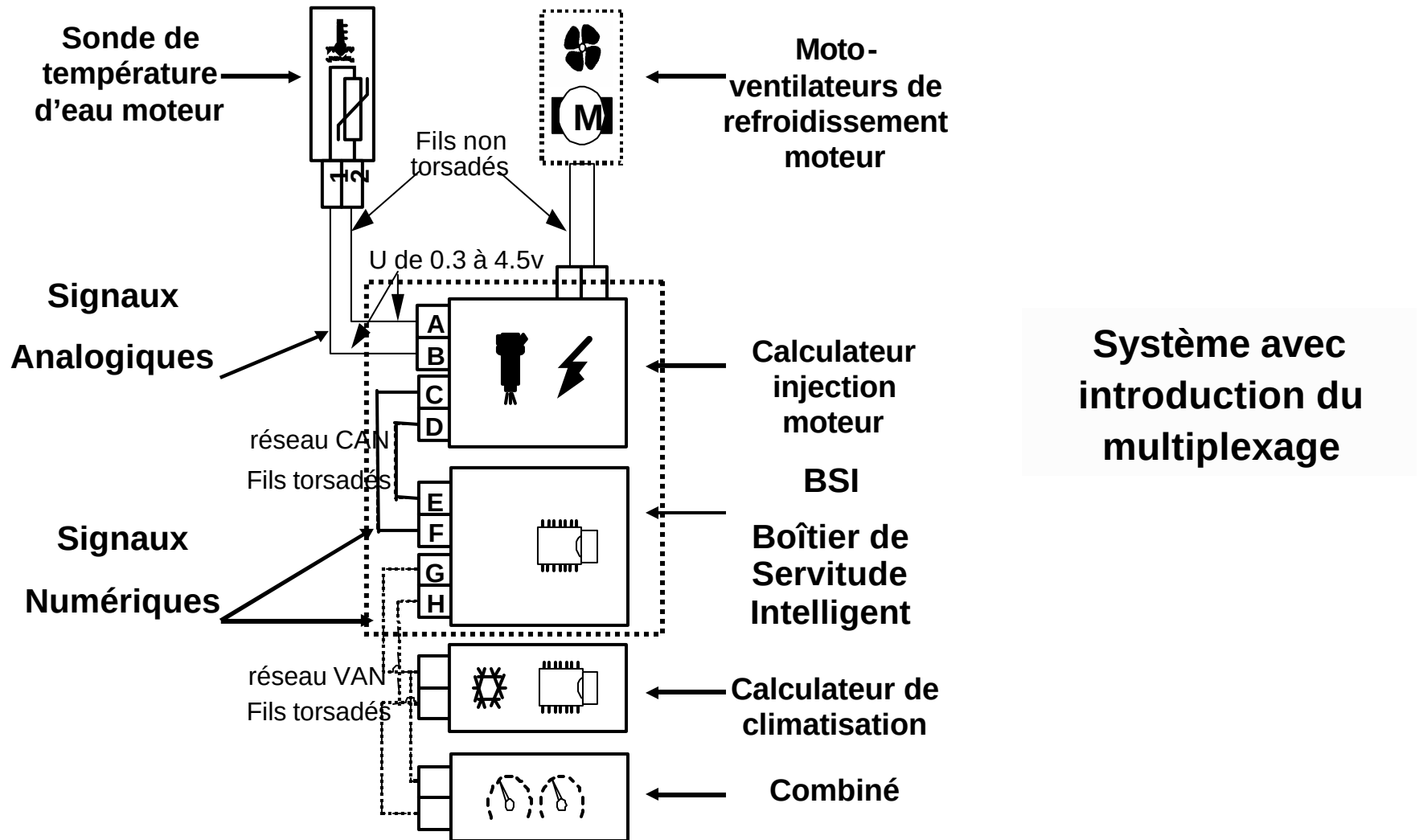
Exemple : Mise en place du multiplexage sur un système de refroidissement



**Système classique
sans multiplexage**

I. Introduction : Notion de multiplexage

Exemple : Mise en place du multiplexage sur un système de refroidissement





I. Introduction : Notion de multiplexage

Les avantages du multiplexage

- Moins de capteurs et/ou de nombres de liaisons avec les boîtiers
- Le poids et les coûts diminuent
- Enrichissement de fonctions sans surcoût important :
 - Faire allumer les feux de croisement lorsque le capteur de pluie détecte une averse (évolution d'un logiciel)
 - Mise en action des feux de détresse lors d'une forte décélération
- Les méthodes répondent à une norme ISO donc fiabilité accrues (théoriquement)
- Les constructeurs « protègent » leur réseau de APV car la plupart des interventions sur les systèmes multiplexés nécessitent l'utilisation d'outils de diagnostic particuliers :
 - Méthodes de recherche de pannes complexes + télé-assistance
 - Téléchargement de mise à jour
 - Apprentissage lors d'une installation ou d'un changement de composants multiplexés (ex : autoradio) : c'est le télé-codage



I. Introduction : Notion de multiplexage

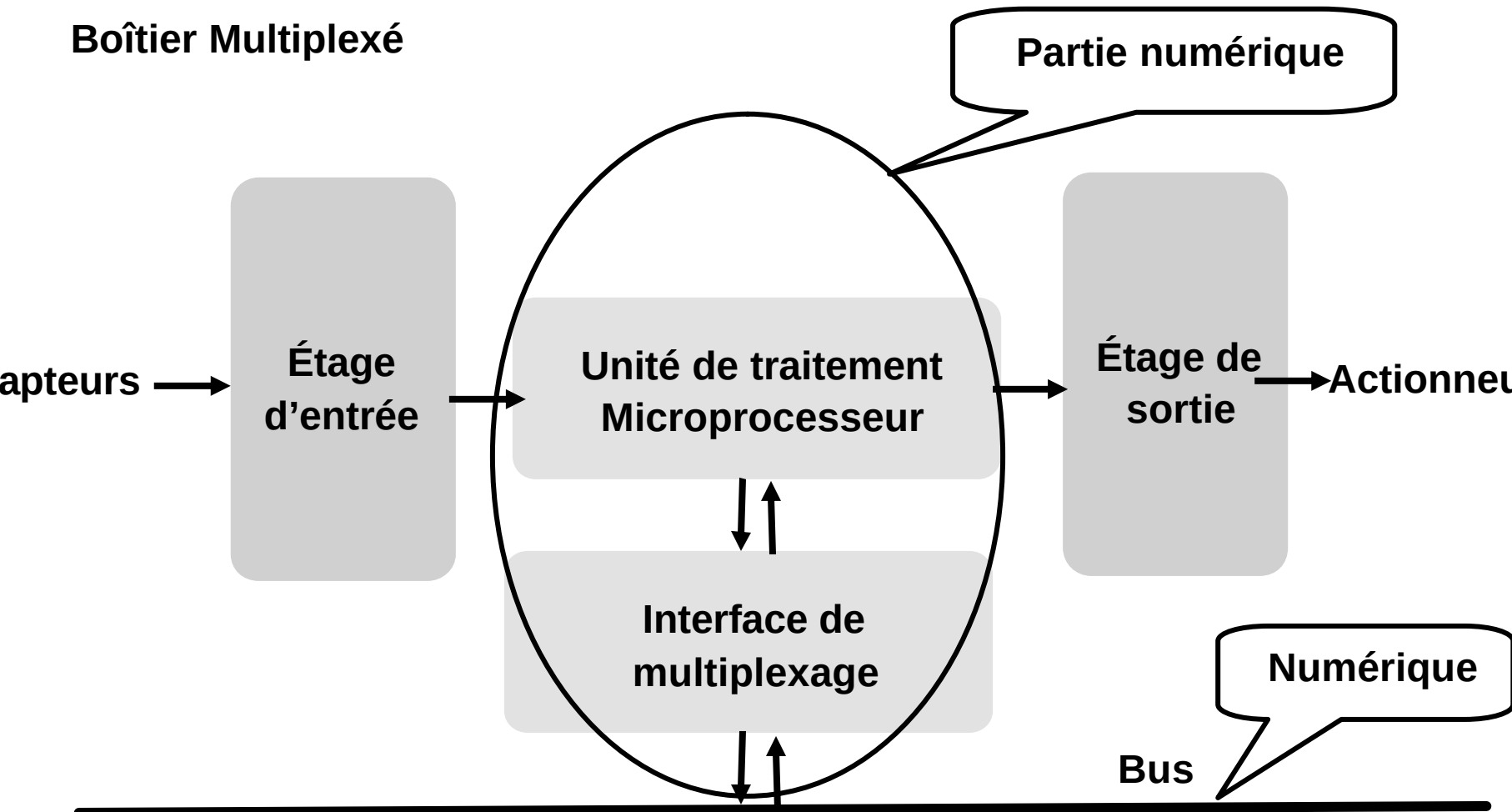
Les principes du multiplexage

- Adaptation des boîtiers
- Le réseau multiplexé
- La transmission des données
- Structure d'une trame
- La synchronisation des horloges des boîtiers
- Arbitrage : gestion des priorités

I. Introduction : Notion de multiplexage

Les principes du multiplexage

Boîtier Multiplexé



I. Introduction : Notion de multiplexage

Les principes du multiplexage

✓ L'étage d'entrée du boîtier :

- Il transforme (il code) les signaux analogiques des capteurs en signaux numériques exploitables par le microprocesseur (ex : signal délivré par la thermistance d'eau)

✓ L'étage de sortie du boîtier : (appelé étage de puissance)

- Il transforme les ordres, fournis par le microprocesseur sous forme de signaux numériques, en signaux analogiques destinés aux actionneurs

✓ L'étage de calcul : le microprocesseur (la puce)

- C'est le composant « intelligent » du boîtier
- Il possède des mémoires qui peuvent être :
 - ROM : mémoire morte qu'on ne peut que lire
 - RAM : mémoire qui disparaît dès que l'alimentation est coupée
 - EEPROM : mémoire morte pouvant être reprogrammées (de + en utilisée)
- La ROM ou l'EEPROM contiennent le ou les programmes à réaliser de la forme : SI, ALORS
- Les signaux traités le sont en général par groupes de 8 bits (ou plus) :

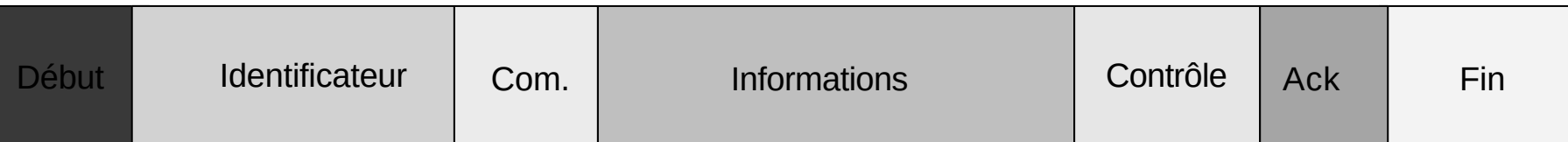


I. Introduction : Notion de multiplexage

Les principes du multiplexage : l'interface

✓ L'interface de multiplexage :

- Elle permet la communication entre le boîtier et le bus
- Les messages qui transitent par l'interface de multiplexage sont numériques et portent le nom de trames
- Ces trames sont découpées en plusieurs champs
- Chacun des champs est composé d'un nombre bien précis de bits à l'état 1 ou à l'état 0



I. Introduction : Notion de multiplexage

Les principes du multiplexage : le réseau

Le réseau est l'ensemble des boîtiers qui communiquent entre eux

Réseau : une architecture + un protocole (VAN, CAN, LIN ...)

C'est la disposition matérielle
des nœuds (boîtiers)

C'est la gestion de la
communication entre les boîtiers
(arbitrage, trame, horloge, débit)

➤ Architecture :

- En étoile (VAN)
- En râteau (VAN)
- En série (CAN)

Maître : peut prendre l'initiative d'une communication sur le réseau

➤ Protocole :

- maître/esclaves
- multi-mâîtres/esclaves
- multi-mâîtres

Esclave : peut seulement répondre à un maître

I. Introduction : Notion de multiplexage

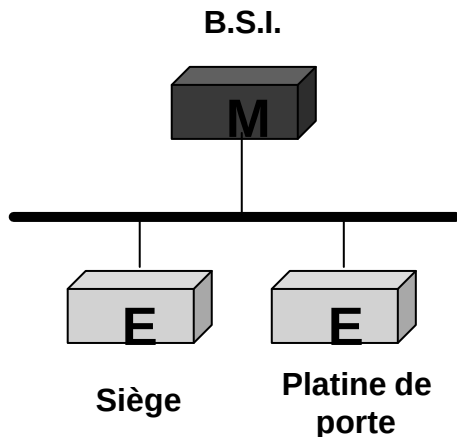
Les principes du multiplexage : le réseau

✓ L'architecture du réseau est adaptée, suivant les besoins en vitesse d'échange d'informations (ex : info passage rapport BVA au boîtier moteur (250 kbits/s) et commande essuie glace arrière (62.5 kbits /s)).

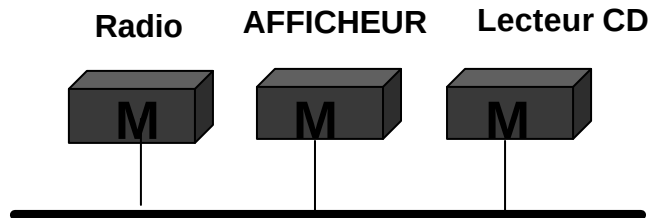
✓ Quelques particularités :

- La vitesse maxi de communication est inversement proportionnelle à la distance entre 2 participants
- Plus il y a de participants sur le réseau, plus la vitesse de communication diminue

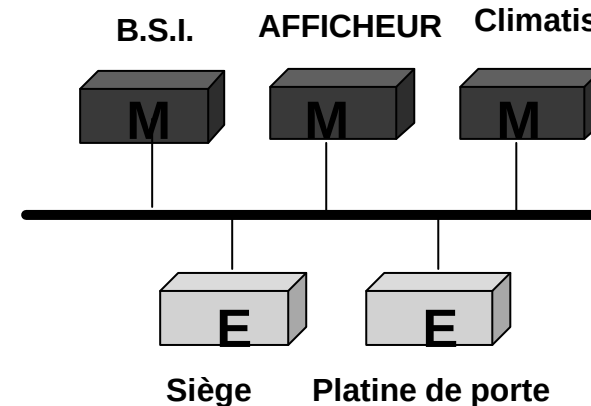
Maître / Esclaves



Multi-Maîtres



Mixte



I. Introduction : Notion de multiplexage

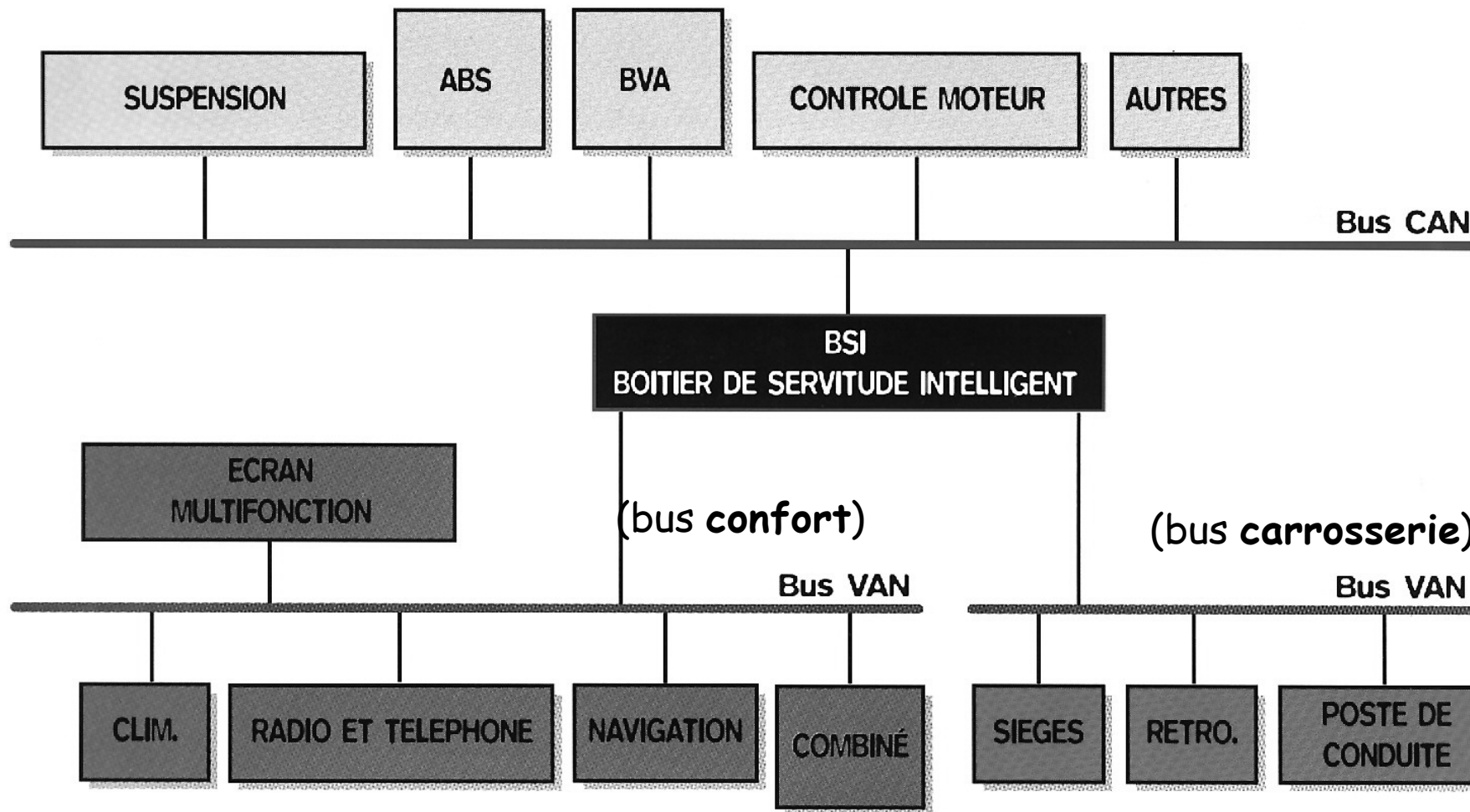
Les principes du multiplexage : le réseau (le protocole)

Le protocole : c'est la « langue » utilisée pour communiquer

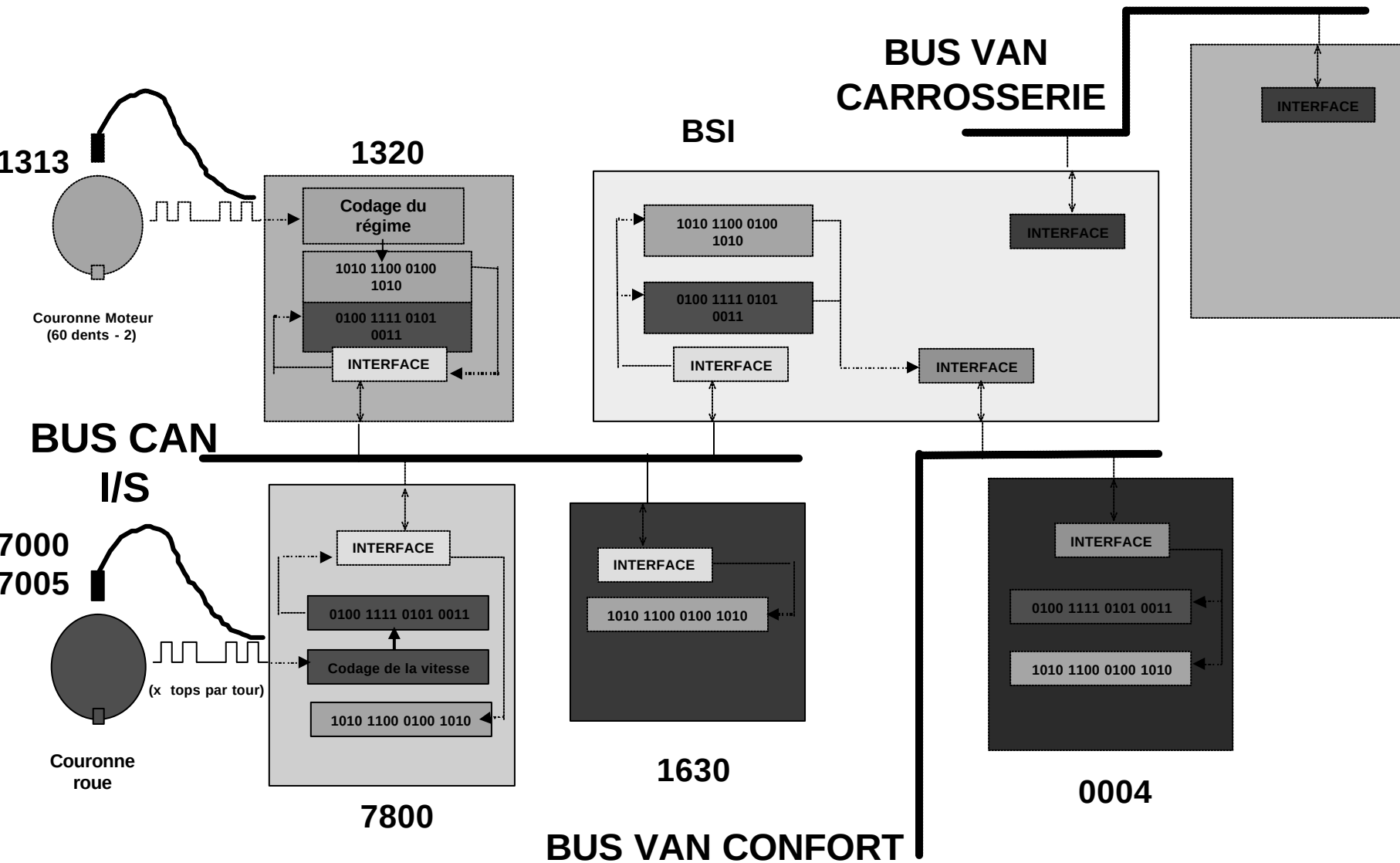
- C'est tout ce qui concerne l'acheminement des trames
- Les trames sont distribuées sur le bus
- Les « récepteurs » consultent l'identité de la trame (champ d'identification de la trame) et seuls ceux qui sont concernés par la trame, utilisent ses informations
- Les échanges de trame, donc de bits, doivent se faire à un rythme bien précis. Pour ce faire chacun des boîtiers possède une horloge interne (quartz)
- Les boîtiers récepteurs doivent caler leur horloge sur celle de l'émetteur
- Il se peut que 2 boîtiers veuillent émettre une trame en même temps sur le bus ; une trame est forcément prioritaire sur l'autre, c'est l'arbitrage.
- Seule la trame prioritaire est émise mais la 2ème n'est pas détruite, elle sera ré-émise dès que le bus sera libre : arbitrage non destructif

II. Notion de bus de communication

le bus : exemple chez PSA



Les principes du multiplexage - La transmission des messages



II. Notion de bus de communication

structure de trame

Début : symbole indiquant le début d'une trame ; les horloges internes des récepteurs se « calent » sur celle de l'émetteur

Identificateur : champ d'identification de la trame qui sert à identifier le contenu du message (ex : régime moteur) et parfois les destinataires

Com. : champ de commande qui annonce la nature du message (données ou requête) pour le VAN, qui annonce le nbre d'octets du champ de données pour le CAN

Informations : champ contenant les données à transmettre (exemple : INFORMATION REGIME MOTEUR envoyée par le boîtier gestion moteur)

Contrôle : champ de contrôle de la cohérence de la trame (l'émetteur calcule un code en fonction des données transmises ; les récepteurs font le même calcul et comparent : si il y a une différence, la trame ne sera pas acquittée)

Ack : champ accusé de réception si aucune erreur détectée en contrôle

Fin : symbole indiquant la fin de la trame

Séparateur de trame : un certain nombre de bits constituent un espace entre 2 trames

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

II. Notion de bus de communication

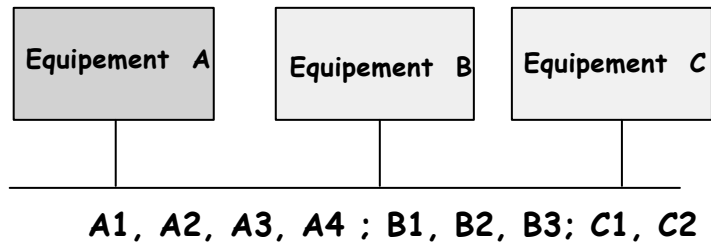
Synchronisation des horloges

- ✓ Sur le réseau, la durée de transmission d'un bit peut varier d'un noeud à l'autre ; les boîtiers doivent donc effectuer une synchronisation pour une bonne réception : c'est la synchronisation des horloges
- ✓ Ce sont les horloges des récepteurs qui se calent sur l'horloge de l'émetteur :
 - En début d'émission de trame sur le bus (: le bus passe de l'état de repos à celui d'activité)
 - Pendant l'émission de la trame : les bits Stuffing sur le CAN
- ✓ Le récepteur compare sa durée de transmission d'un bit avec celle de la trame en cours de lecture.
- ✓ La synchronisation consiste à allonger ou raccourcir la durée de transmission d'un bit du boîtier récepteur, pour l'ajuster avec celle d'un bit du boîtier émetteur

II. Notion de bus de communication

notion d'arbitrage

- Il peut arriver que 2 nœuds (ou plus) émettent simultanément une trame sur le bus
- Au début d'émission pas de conflit, car le champ de début de trame est identique pour tous les boîtiers.
- Mais ensuite il va falloir déterminer laquelle des trames est prioritaire sur les autres, elle sera la seule transmise.



- Arbitrage bit à bit (niveaux Récessif / Dominant)

bit à 0 = Dominant

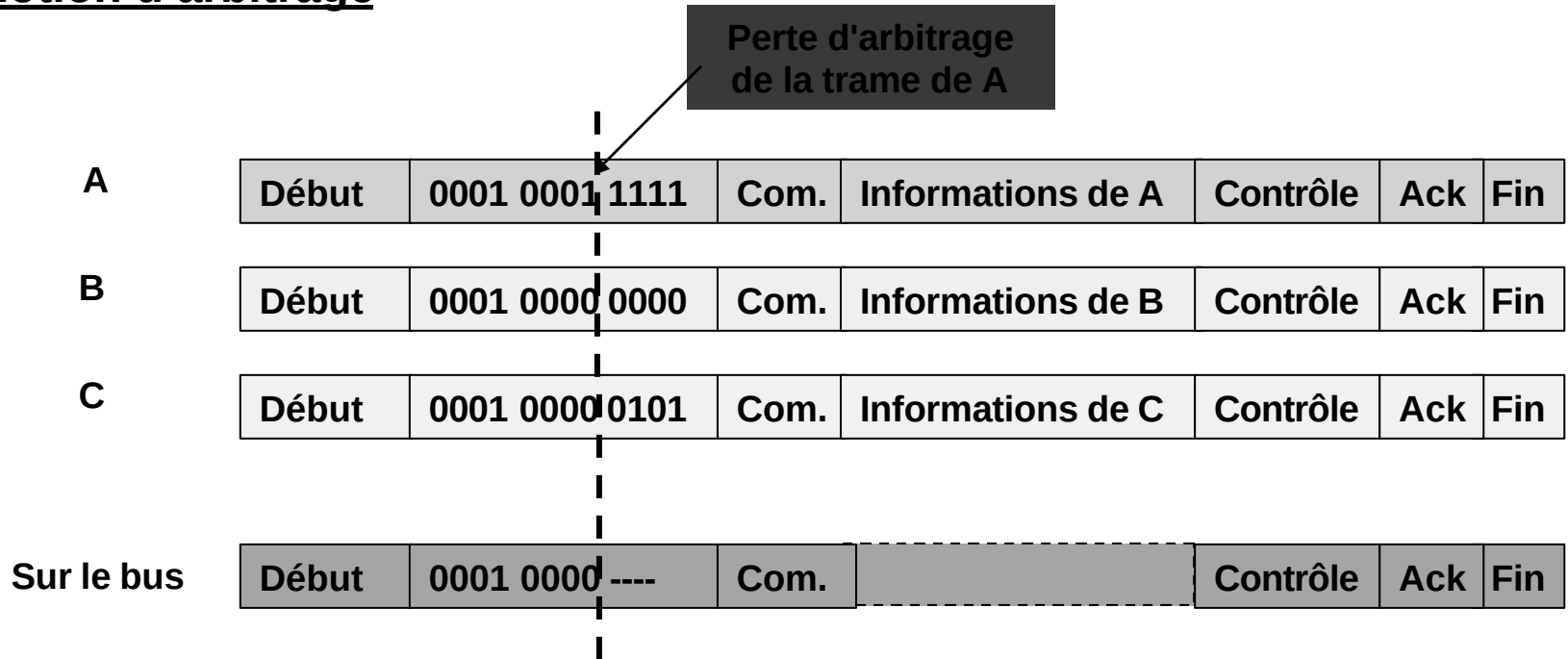
bit à 1 = Récessif

Un niveau Dominant l'emporte toujours sur un niveau Récessif

A	Début	0001 0001 1111	Com.	Informations de A	Contrôle	Ack	Fin
B	Début	0001 0000 0000	Com.	Informations de B	Contrôle	Ack	Fin
C	Début	0001 0000 0101	Com.	Informations de C	Contrôle	Ack	Fin

II. Notion de bus de communication

notion d'arbitrage



➤ Chaque émetteur compare le bit qu'il reçoit avec celui qu'il émet ; tant que ces 2 bits sont identiques les 2 transmissions continuent

➤ Dès que 2 bits diffèrent, le boîtier ayant émis un bit à l'état récessif cesse d'émettre

II. Notion de bus de communication

notion d'arbitrage

B	Début	0001 0000 0000	Com.	Informations de B	Contrôle	Ack	Fin
---	-------	----------------	------	-------------------	----------	-----	-----

C	Début	0001 0000 0101	Com.	Informations de C	Contrôle	Ack	Fin
---	-------	----------------	------	-------------------	----------	-----	-----

Perte d'arbitrage de la trame de C

Sur le bus	Début	0001 0000 0000	Com.	Informations de B	Contrôle	Ack	Fin
------------	-------	----------------	------	-------------------	----------	-----	-----

- **Sur le VAN**, la priorité d'une trame peut-être déterminée sur toute sa longueur
- **Sur le CAN**, la priorité est déterminée sur le seul champ d'identification

III. Le bus CAN (control Area Network)

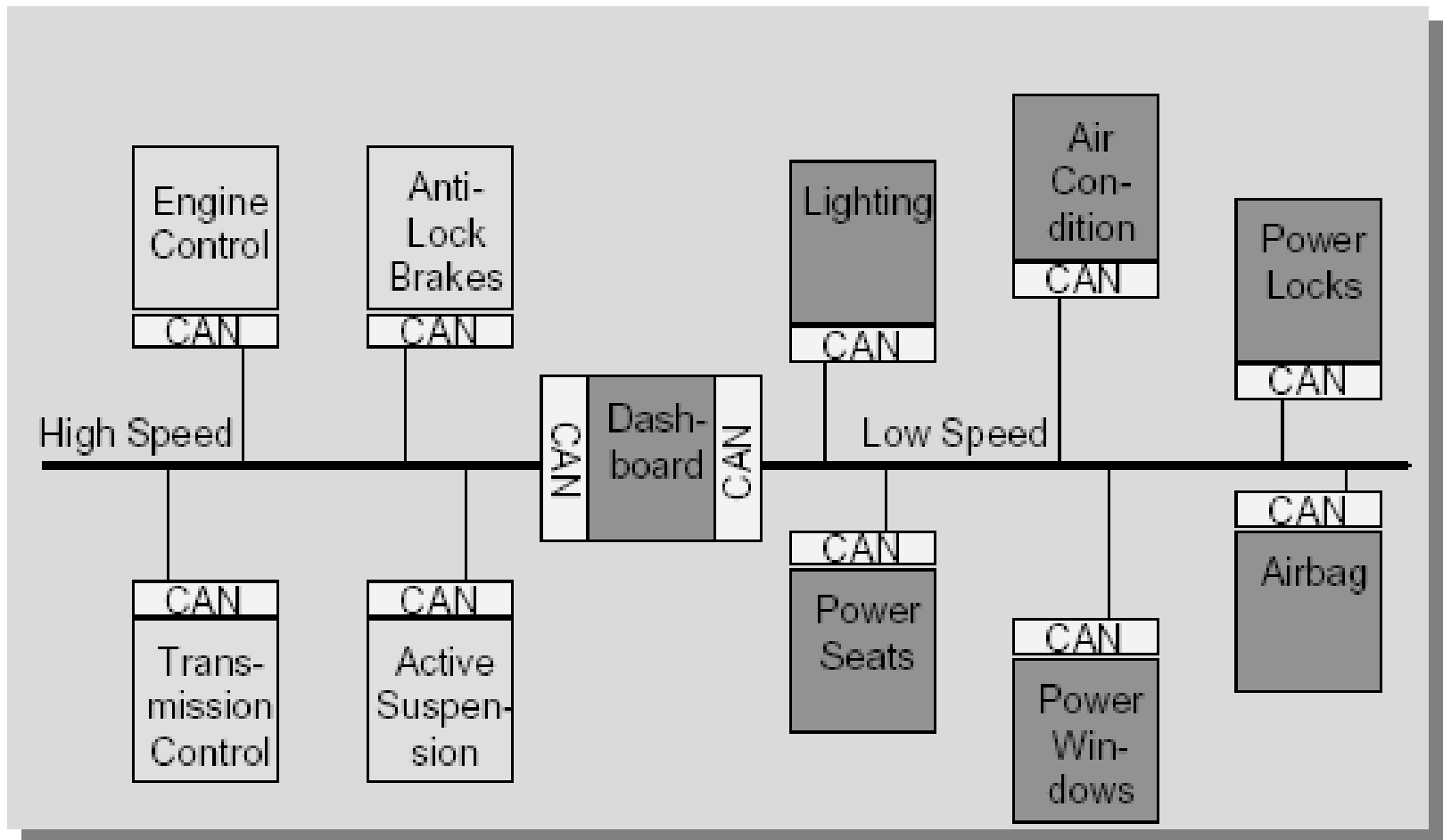
Généralités

Le protocole CAN (*Control Area Network*) est un protocole de communication série qui supporte des systèmes temps réel avec un haut niveau de fiabilité. Ses domaines d'application s'étendent des réseaux moyens débits aux réseaux de multiplexages faibles coûts. Il est avant tout à classer dans la catégorie des réseaux de terrain

- hiérarchisation des messages.
- garantie des temps de latence.
- souplesse de configuration.
- réception de multiples sources avec synchronisation temporelle.
- fonctionnement multimaître.
- détections et signalisations d'erreurs.
- retransmission automatique des messages altérés dès que le bus est de nouveau au repos.
- distinction d'erreurs : d'ordre temporaire ou de non-fonctionnalité permanent au niveau d'un nœud.
- déconnexion des nœuds défectueux.

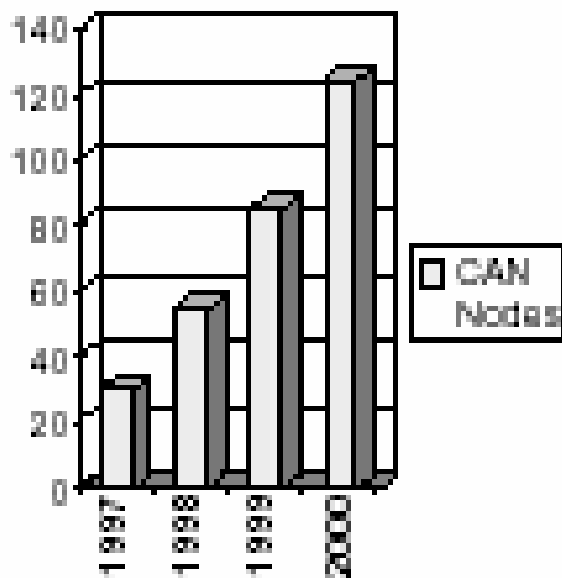
III. Le bus CAN (control Area Network)

Comment tout à commencé



III. Le bus CAN (control Area Network)

Comment tout à commencé



- Le CAN étant un protocole réseau, il s'intègre dans la norme ISO/OSI
(ISO: *International Standards Organization*, OSI: *Open Systems Interconnection*)

- Le CAN est un bus de communication série développé à la fin des années 80 par l'entreprise allemande Robert Bosch. L'objectif était de fournir à l'industrie automobile un bus peu coûteux pour l'électronique embarquée



Specified by Robert Bosch
GmbH, Germany



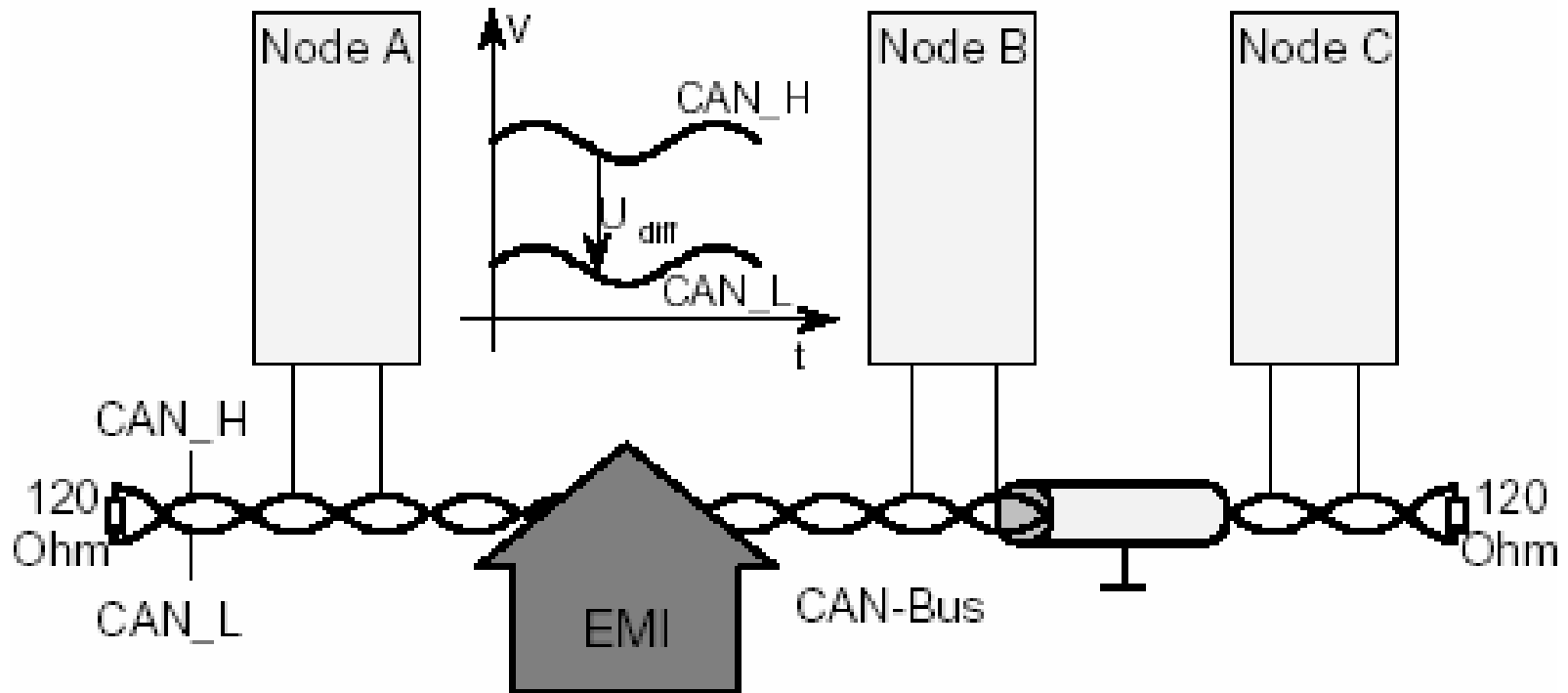
ISO/OSI
SAE

ISO 11898 pour les applications à hauts débits

ISO 11519 pour les applications à bas débits.

III. Le bus CAN (control Area Network)

Caractéristiques physiques du Bus



Le bus série est une paire différentielle connectée sur des résistances 1200 ou des capacités de 30 pF résistant aux perturbations

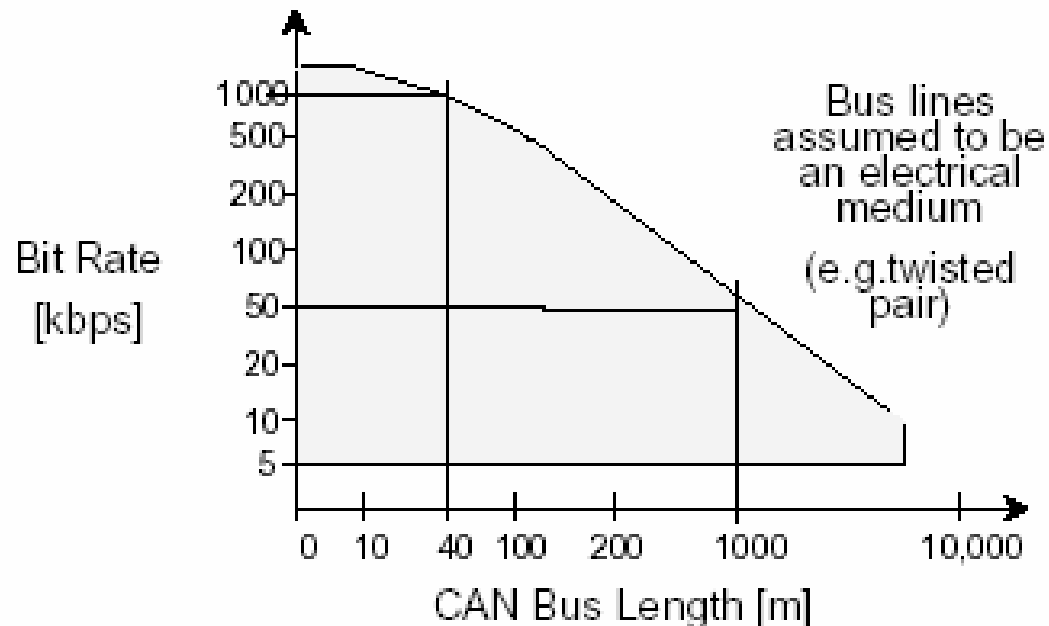
III. Le bus CAN (control Area Network)

Caractéristiques physiques du Bus

- ✓ Le bus CAN utilise deux fils (une paire torsadée ou non) dont les désignations sont CAN L (low) et CAN H (High)
- ✓ Les états logiques (0 ou 1) sont codés par différence de potentiel entre les deux fils : tenue aux perturbations

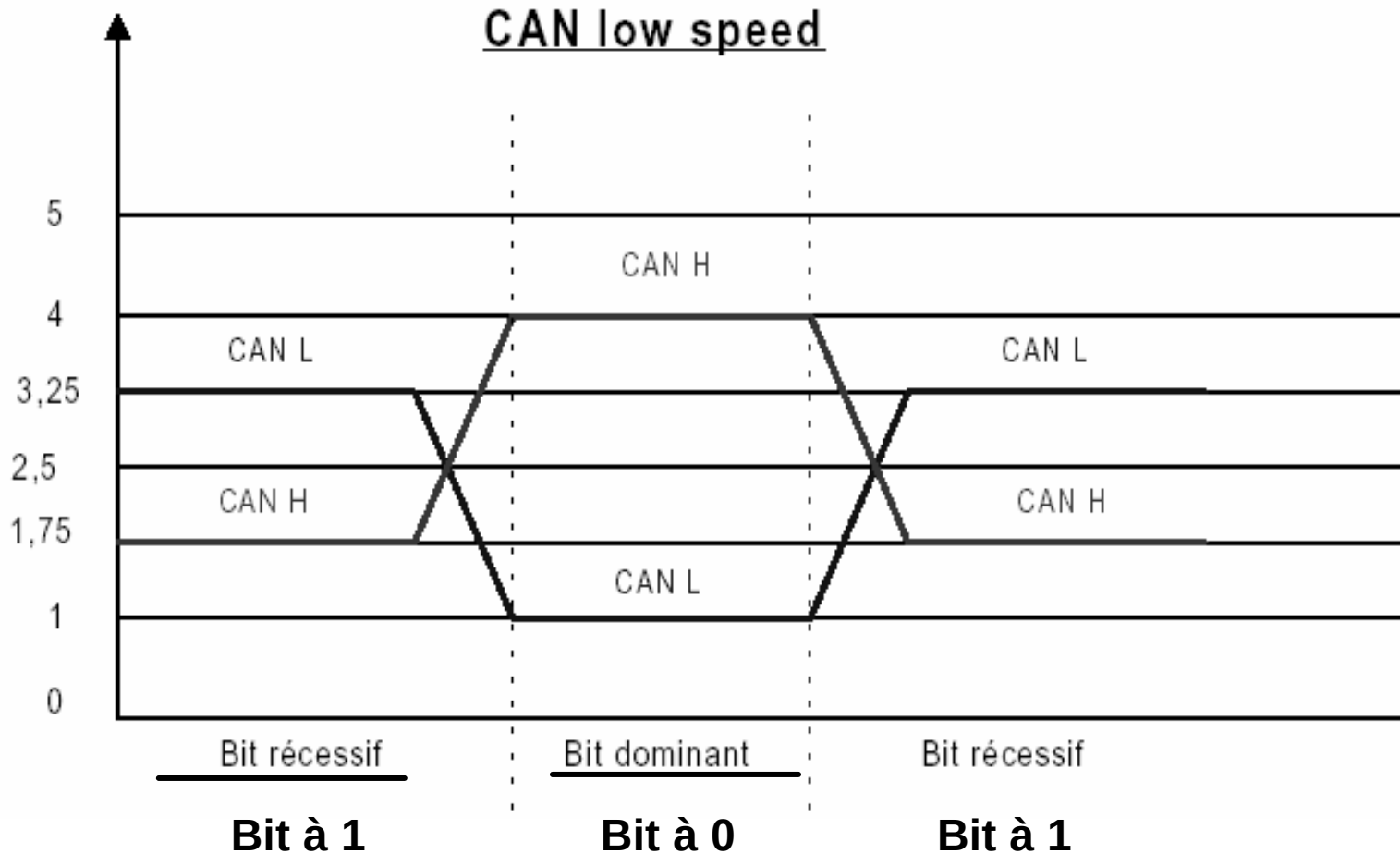
Il existe deux possibilités de configuration du bus suivant que l'on souhaite travailler :

- low speed (125 Kbits)
- high speed (1 Mbits)



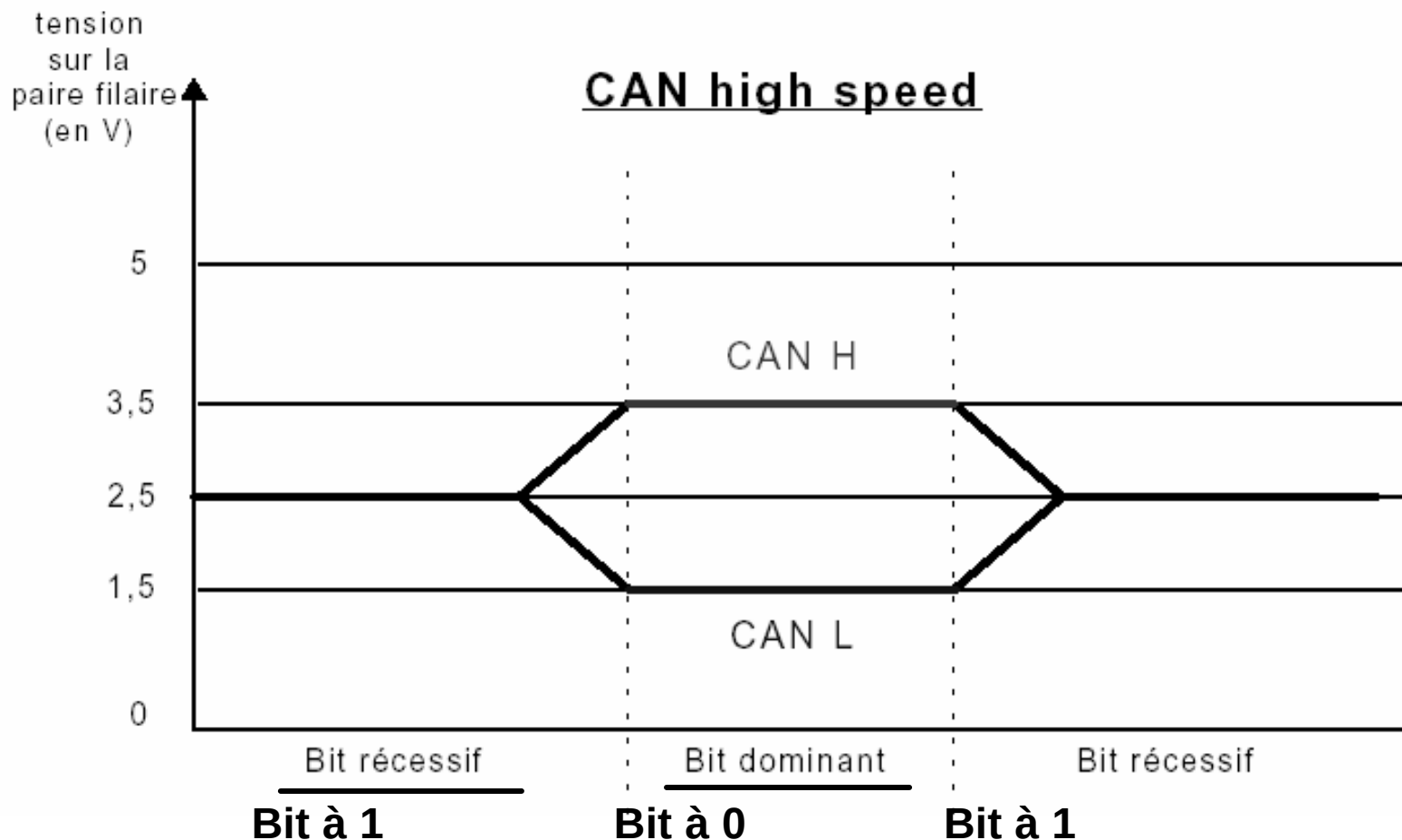
III. Le bus CAN (control Area Network)

Caractéristiques physiques du Bus : low speed



III. Le bus CAN (control Area Network)

Caractéristiques physiques du Bus : high speed





III. Le bus CAN (control Area Network)

Caractéristiques physiques du Bus : résumé

Paramètres	CAN <i>low speed</i>	CAN <i>high speed</i>
Débit	125 kb/s	125 kb/s à 1 Mb/s
Nombre de nœuds sur le bus	2 à 20	2 à 30
Courant de sortie (mode émission)	> 1 mA sur 2,2 k Ω	25 à 50 mA sur 60 Ω
Niveau dominant	CAN H = 4V CAN L = 1V	$V_{CAN\ H} - V_{CAN\ L} = 2V$
Niveau récessif	CAN H = 1,75V CAN L = 3,25V	$V_{CAN\ H} = V_{CAN\ L} = 2,5V$
Caractéristique du câble	30 pF entre les câbles de ligne	2*120 Ω
Tensions d'alimentation	5V	5V

III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus

Il existe uniquement deux états logiques équivalents à

- Dominant : 0
- Récessif : 1

Le bus utilise un fonctionnement de type « Wired-and » qui fonctionne de la façon suivante : un bit dominant (0) écrase un bit Récessif (1)

A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

As soon as one node transmits a dominant bit (zero):

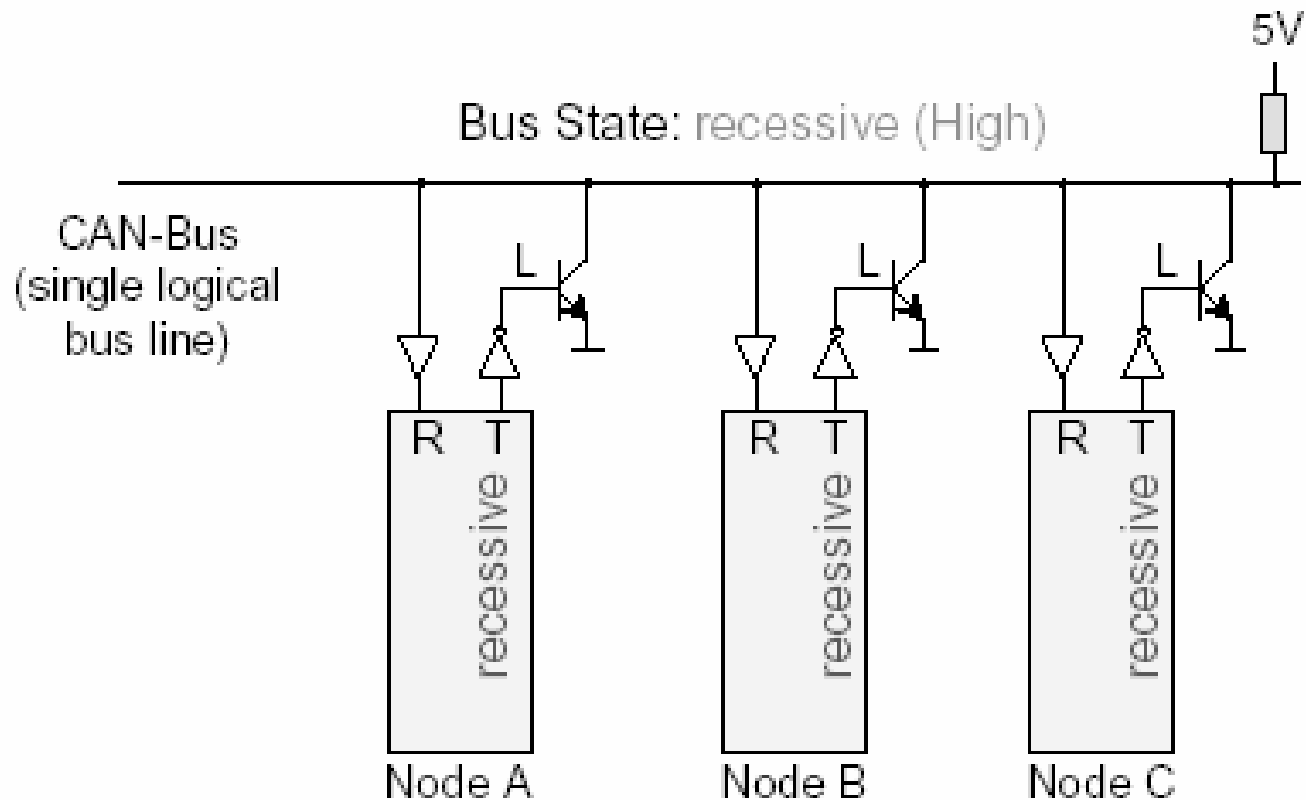
Bus is in the dominant state.

Only if all nodes transmit recessive bits (ones):

Bus is in the recessive state.

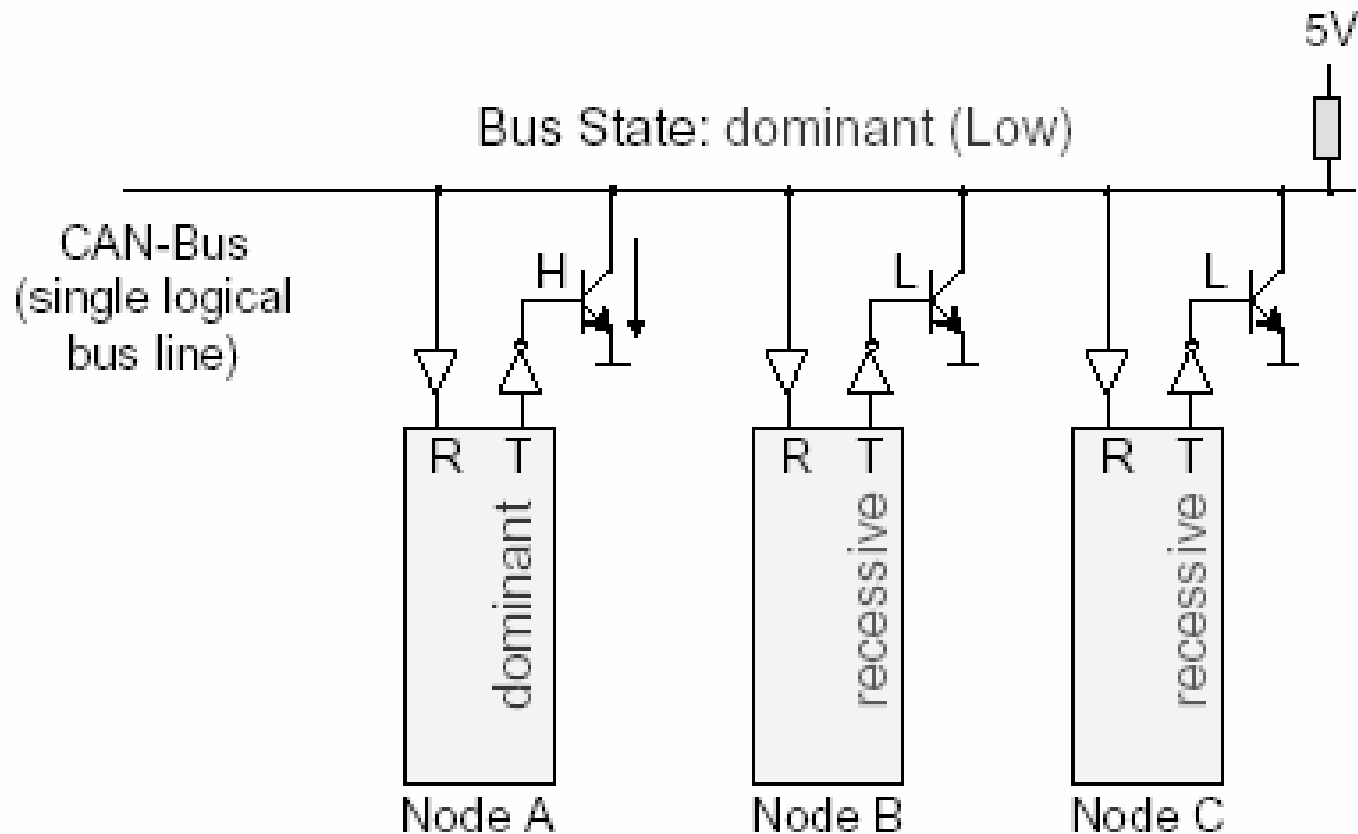
III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus : concept « Wired-and »



III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus : concept « Wired-and »



III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus : Synchronisation d'horloge

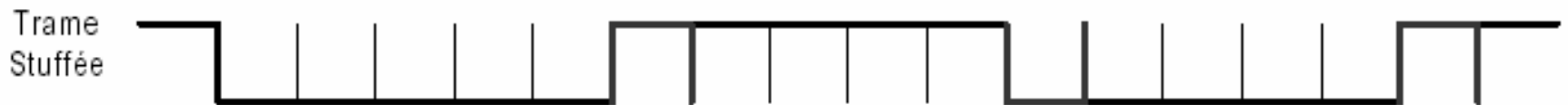
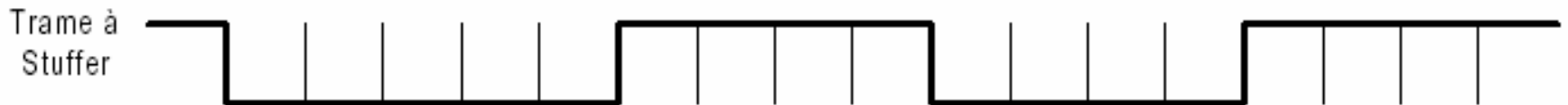
- ✓ Pour que le message soit bien transmis, les horloges de l'émetteur et du récepteur ne doivent pas avoir de décalage. Pour cela il suffit de re-synchroniser régulièrement l'horloge du récepteur sur celle de l'émetteur
- ✓ Le principe consiste à effectuer un bourrage de bit inverse : méthode de bit stuffing
- ✓ Après 5 bits de même niveau, un bit (sans signification) de niveau inverse est ajouté
- ✓ Le récepteur reconnaît ces bits stuffing, cale son horloge, les supprime, et reconstitue le message initial
- ✓ La vitesse de transmission CAN est exprimée en bits/s. Le débit réel des infos ne doit pas tenir compte de ces bits stuffing.



III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus : Synchronisation d'horloge

Technique de Bit-stuffing

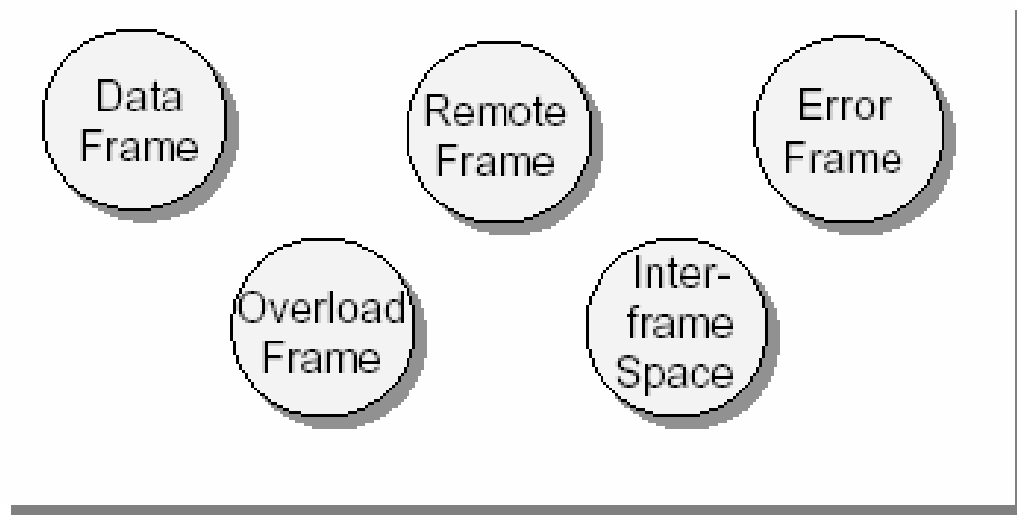


III. Le bus CAN (control Area Network)

Principe de fonctionnement du bus : Structure de trame

Il existe plusieurs format de trames :

- trame de données (data frame)
- trame de requête (remote frame)
- trame de gestion d'erreur (error frame)
- Trame de surcharge (overload frame)
- espace entre trame (inter-frame space)



III. Le bus CAN (control Area Network)

Trame de donnée

Une trame de données se décompose en 7 champs différents (figure 3) :

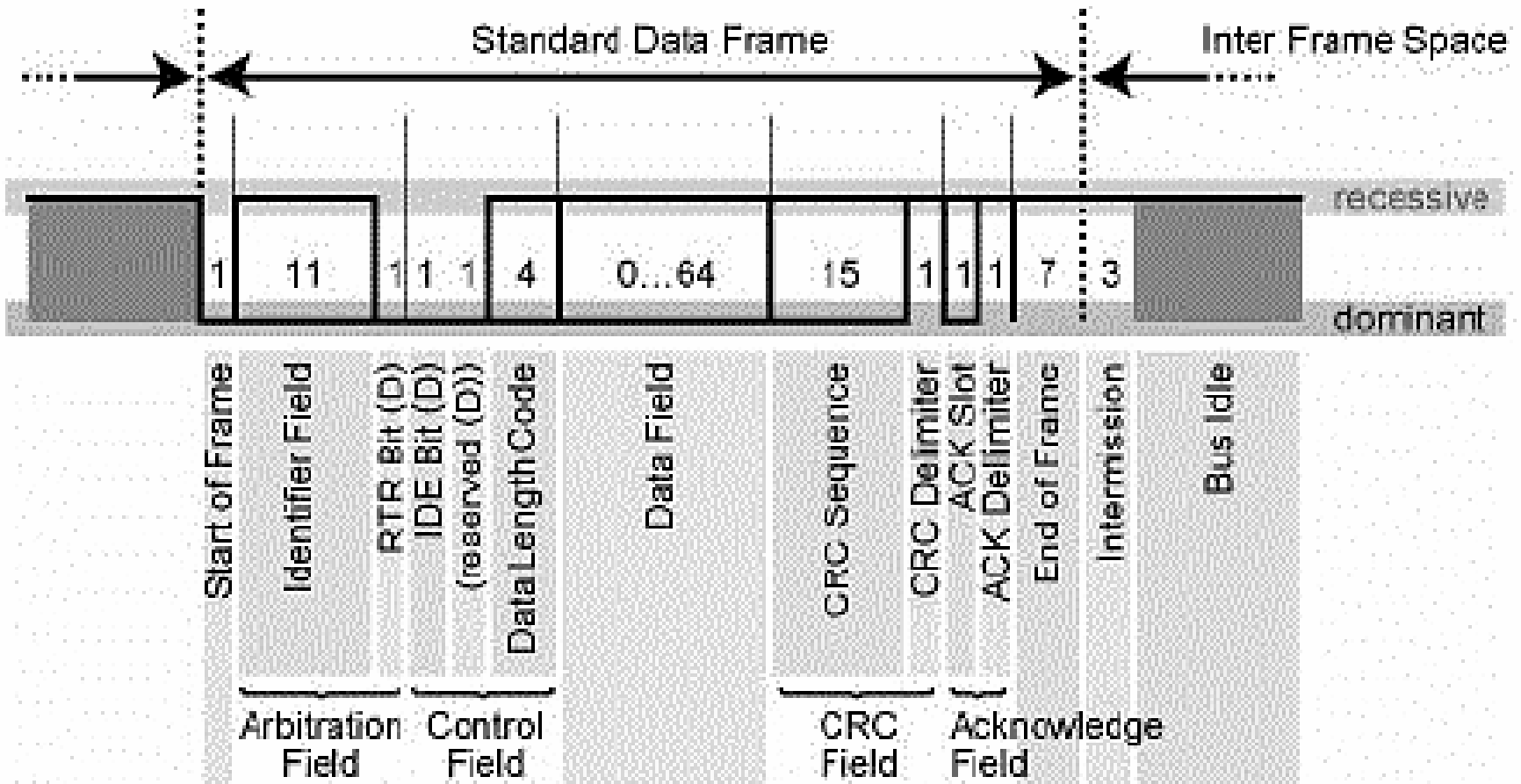
- le début de trame SOF (*Start Of Frame*), 1 bit dominant.
- le champ d'arbitrage, 12 bits (identificateur).
- le champ de contrôle, 6 bits (Commandes).
- le champ de données, 0 à 64 bits (informations).
- le champ de CRC (*Cyclic Redundancy Code*), 16 bits (contrôle).
- le champ d'acquiescement (*Acknowledge*), 2 bits.
- le champ de fin de trame EOF (*End Of Frame*), 7 bits récessifs.

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

SOF (Start Of Frame) : Commence toujours par 1 bit dominant (bit à 0), la ligne étant précédemment au repos. Ce bit ne sert qu'à synchroniser les horloges internes des récepteurs sur celle de l'émetteur : bit de start

III. Le bus CAN (control Area Network)

Trame de donnée



III. Le bus CAN (control Area Network)

Trame de donnée : Champ d'arbitrage

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

Champ composé de 12 bits :

Les 11 premiers indiquent l'identité du contenu du message, et servent également à l'arbitrage (gestion des priorités)

Le dernier bit (RTR : Remote transmission Request bit) permet de coder la nature du message : trame de données (ex : régime moteur) ou trame de requête (demande de T° eau)

➡ bit à 0 (dominant) : trame de données

➡ bit à 1 (récessif) : trame de requête

III. Le bus CAN (control Area Network)

Trame de donnée : Champ de commande

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

✓ **Champ de commande constitué de 6 bits :**

- ✓ Le premier bit IDE (Identifieur Extention bit) est un bit dominant permettant de spécifier qu'il s'agit d'une trame standard
- ✓ Le deuxième bit est réservé est définit comme dominant.
- ✓ Les 4 derniers permettent de coder le nombre d'octets du champ de données

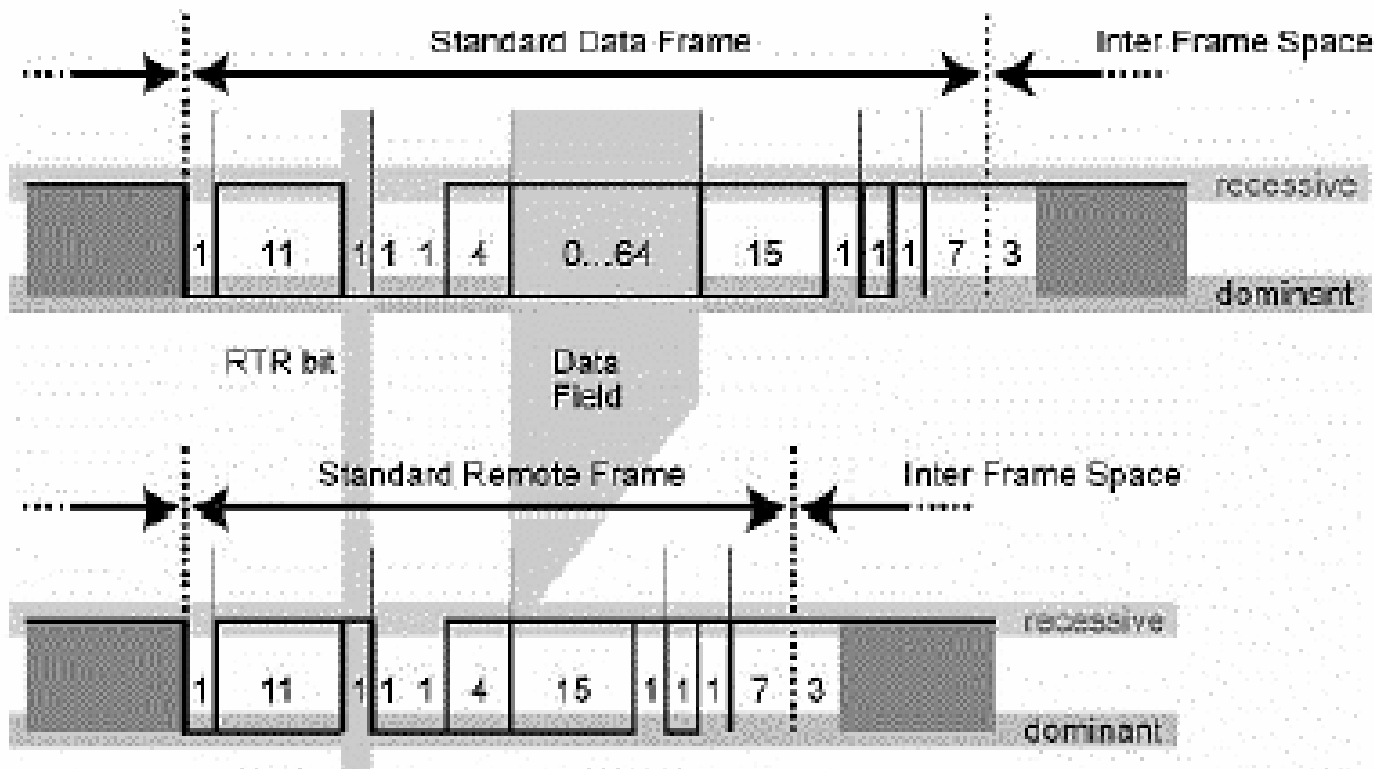
	Nbre d'octets du champ de données								
	0	1	2	3	4	5	6	7	8
Bit n°1	0	0	0	0	0	0	0	0	1
Bit n°2	0	0	0	0	1	1	1	1	1
Bit n°3	0	0	1	1	0	0	1	1	1
Bit n°4	0	1	0	1	0	1	0	1	1

III. Le bus CAN (control Area Network)

Frame de donnée : Champ de données

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

Ce champ contient de 0 à 8 octets de données (64 bits maxi)



III. Le bus CAN (control Area Network)

Trame de donnée : Champ de contrôle

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

Le champ de contrôle : CRC (Cyclic Redundancy) est composé de 16 bits, il permet de vérifier les données transmises

Ce champ de vérification des données est composé de 2 parties :

- **Code de vérification des données** transmises sur 15 bits : le récepteur compare son code à celui de l'émetteur ; si différence : pas d'acquittement
- **Délimiteur** de vérification de données : marque la fin de vérification 1 bit toujours à l'état 1 (récessif)
- Ce contrôle est effectué par tous les boîtiers du réseau
- Ce champ est étudié pour une trame de 127 bits max. Le nombre maximales d'erreurs détectées peut être de 5

III. Le bus CAN (control Area Network)

Trame de donnée : Champ d'acquittement

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

Ce champ d'acquittement est composé de 2 bits :

- Un **bit d'acquittement** à l'état 0 (dominant) si le calcul du code de vérification des données est correct ; si une erreur : bit laissé à l'état haut (recessif)
- Un **bit délimiteur d'acquittement**, toujours à l'état haut (1) (recessif)
- **Tous les boîtiers** du réseau doivent acquitter, même si la trame ne les concerne pas (perte de temps possible)

Pendant cette période, l'émetteur laisse la trame libre...ce sont les récepteurs qui complètent la trame.

III. Le bus CAN (control Area Network)

Trame de donnée : Champ d'acquitement

Début	Identificateur	Com.	Informations	Contrôle	Ack	Fin
-------	----------------	------	--------------	----------	-----	-----

✓ Champ de fin de trame : suite de 7 bits à l'état 1

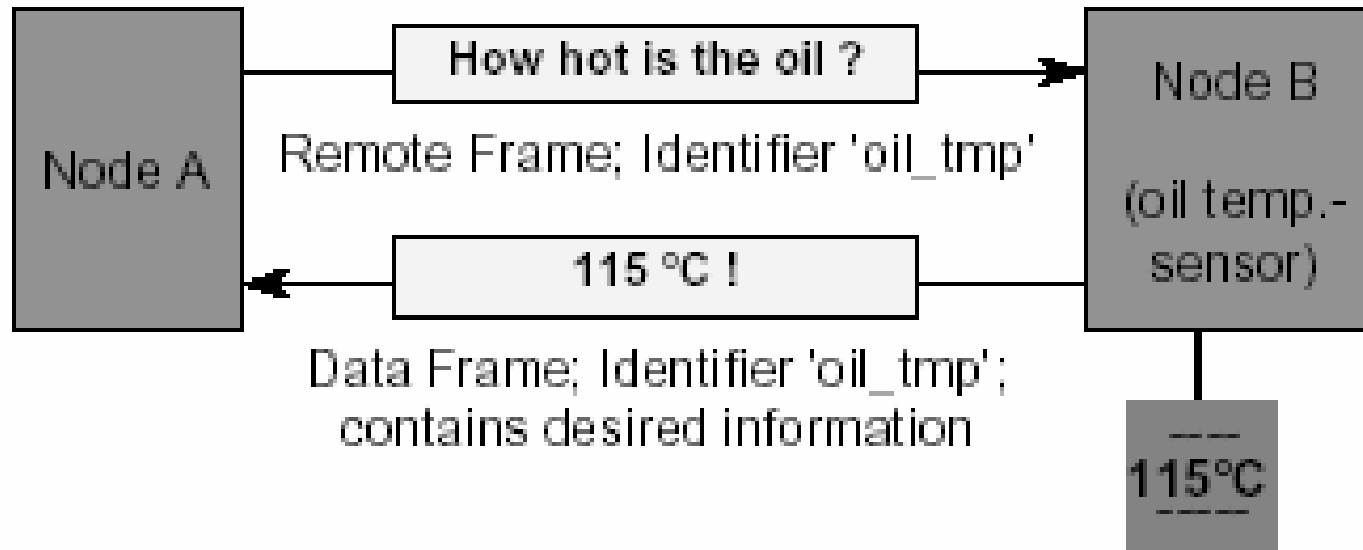
✓ Le codage par bit stuffing est désactivé à partir de cet instant

Remarque :

- 3 bits à l'état 1 séparent obligatoirement 2 trames consécutives
- 108 bits (sans les stuffing) sont nécessaires pour 64 bits de donnée

III. Le bus CAN (control Area Network)

Frame de requête (remote frame)



Scénario : demande d'informations d'un nœud à un autre.

III. Le bus CAN (control Area Network)

Trame de requête

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

Début	Identificateur	Com.	Contrôle	Ack	Fin
-------	----------------	------	----------	-----	-----

➔ Dans le champ d'arbitrage, le bit de RTR est récessif. Par conséquent si deux nœuds émettent chacun une trame possédant le même identificateur (c'est à dire qu'un nœud émet une trame de données et l'autre une trame de requête), l'arbitrage sur le bit de RTR va donner la priorité à la trame de données.

➔ Les règles de construction des autres divers champs d'une trame de requête sont les mêmes que dans le cas d'une trame de données.

III. Le bus CAN (control Area Network)

Trame de gestion des erreurs

Les erreurs de transmission sont détectées par le décodage du champ de contrôle, et sont spécifiées au niveau du champ d'acquittement.

On peut dans le protocole du bus CAN détecter 5 types d'erreurs :

- Erreur de bit
- Erreur de stuffing
- Erreur de CRC
- Erreur d'acknowledge delimiter
- Erreur d'acknowledge slot

III. Le bus CAN (control Area Network)

Trame de gestion des erreurs : Erreur de Bit

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.

Exception :

- présence d'un bit dominant à la place d'un bit récessif dans le champ d'arbitrage (perte d'arbitrage)
- présence d'un bit dominant à la place d'un bit récessif dans le champ d'acquittement (signalisation d'erreur par un receptrur)



III. Le bus CAN (control Area Network)

Trame de gestion des erreurs : Erreur de stuffing

Une erreur de Stuffing est détectée à chaque fois qu'il y a 6 bits ou plus consécutifs de même signe sur le bus.

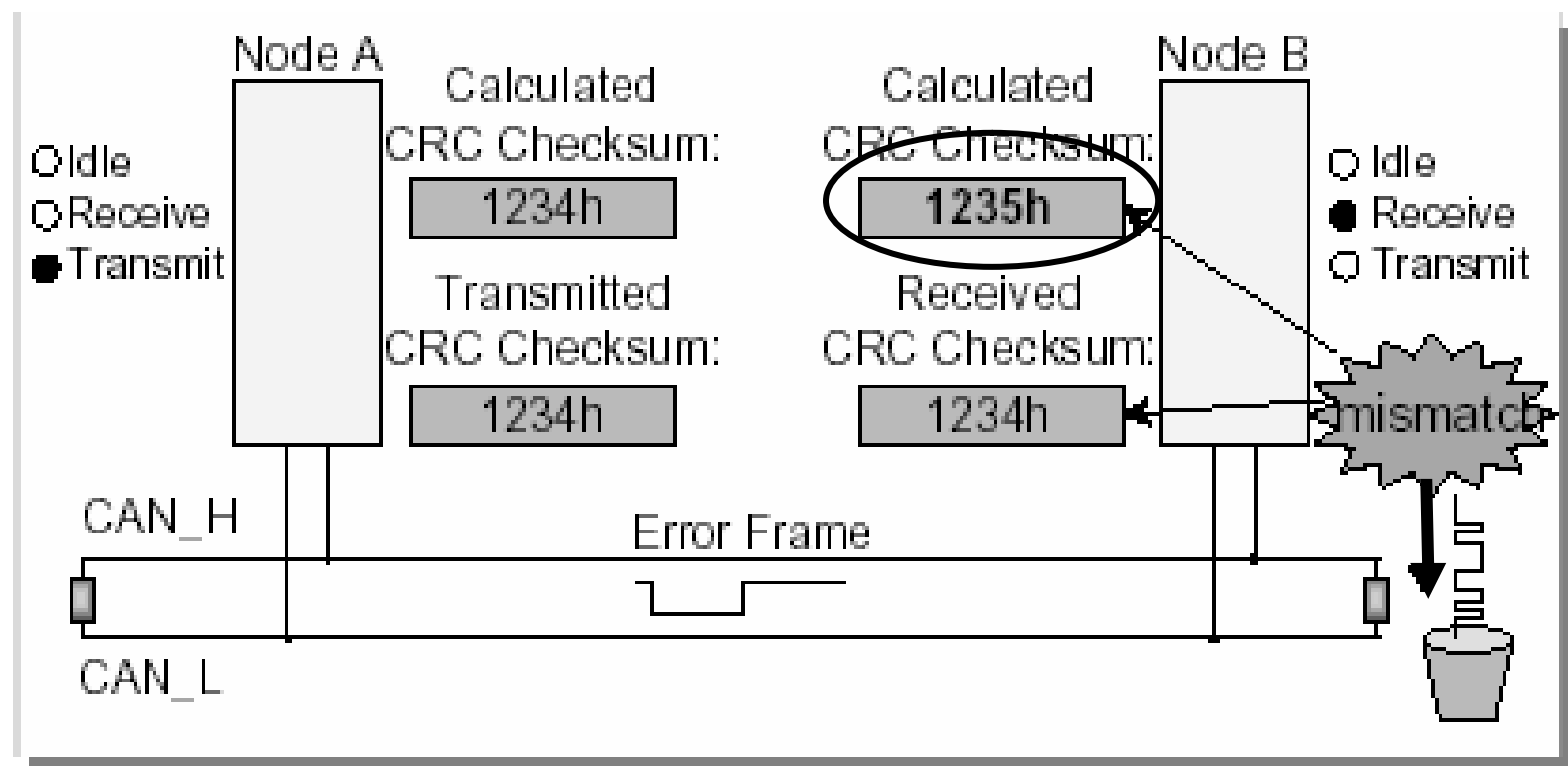
Cependant,

- Une erreur de Stuffing ne doit être signalée que dans les champs d'identificateurs, de commande et de CRC.
- La règle du Bit-Stuffing ne s'appliquant plus après la fin du CRC. En aucun cas, une erreur de Bit-Stuffing ne doit être signalée dans le champ de fin de trame ou dans le champ d'acquiescement.

III. Le bus CAN (control Area Network)

Trame de gestion des erreurs : Erreur de CRC

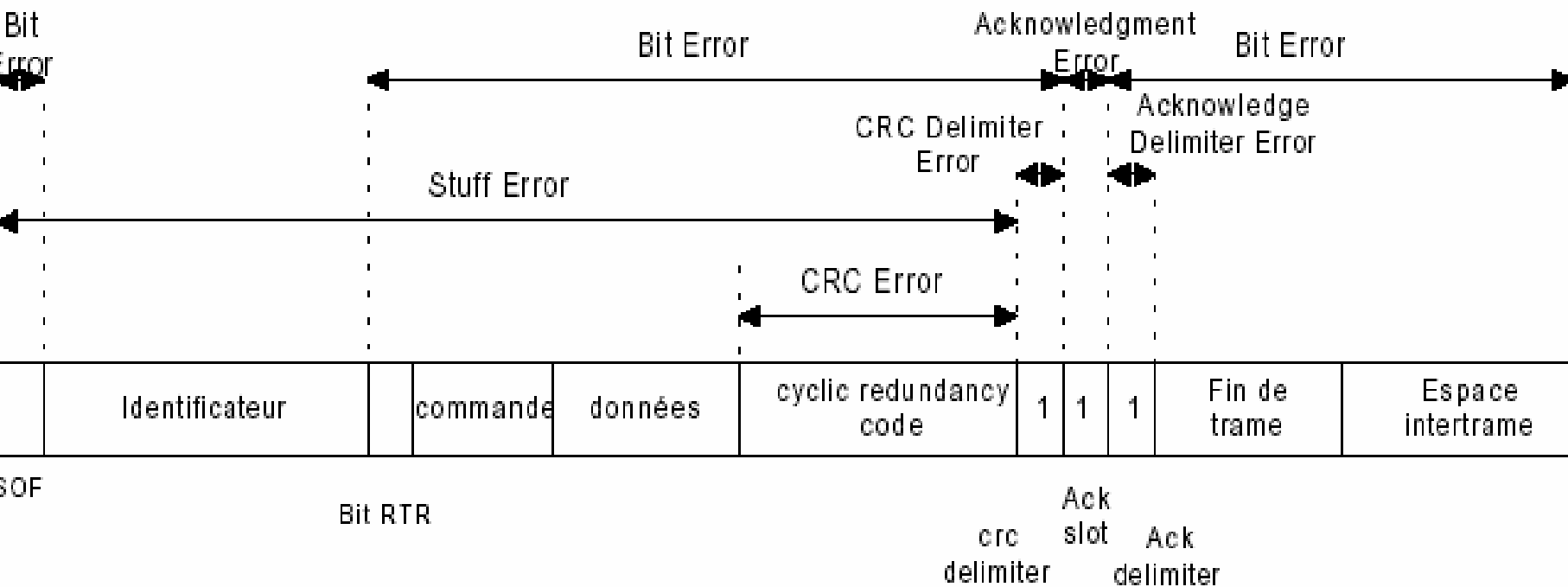
Si la valeur du CRC calculée par le récepteur est différente de celle envoyée par l'émetteur, il y a erreur de CRC (CRC Error).





III. Le bus CAN (control Area Network)

Trame de gestion des erreurs : résumé



III. Le bus CAN (control Area Network)

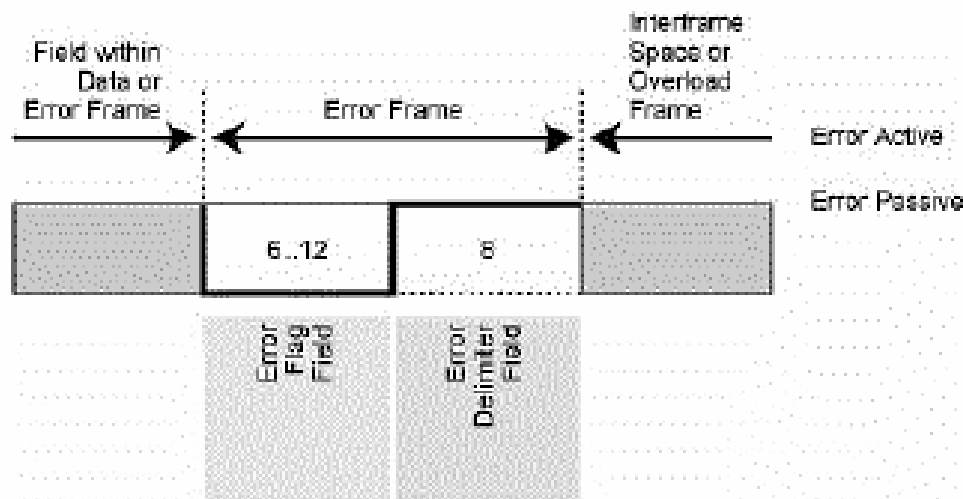
Trame d'erreur

Cette trame est générée par n'importe quel nœud détectant une erreur. La trame d'erreur est constituée de deux champs principaux :

- le drapeau d'erreur,
- le délimiteur de champ.

Le champ des drapeaux peut être constitué de deux sortes de drapeaux

- les drapeaux d'erreur active (Active Error Flag),
- les drapeaux d'erreur passive (Passive Error Flag).



III. Le bus CAN (control Area Network)

Gestion du mode d'erreur

Suivant le nombre d'erreur qu'un nœud comptabilise, l'état du mode de ce nœud peut différer. Un compteur mémorise le nombre d'erreur rencontré lors de la transmission des trames sur le bus. Deux compteurs séparés régissent respectivement le nombre d'erreurs en émission et en réception. Il se nomme :

- Transmit Error Counter pour l'émission (TEC)
- Receive Error Counter pour la réception (REC)

Par défaut (à l'initialisation), le nœud est dans l'état « erreur Active ». Le bus peut émettre et recevoir sans restriction.

Pendant le fonctionnement, les compteurs sont incrémentés lors de la détection d'erreur (variable suivant le type d'erreur).

Pour chaque transmission réussie, le compteur est décrémenté. Le nœud reste dans l'état « erreur active » tant que la valeur du compteur reste inférieure à 127

III. Le bus CAN (control Area Network)

Gestion du mode d'erreur

Si la valeur des compteurs dépasse 127, le nœud passe à l'état « erreur passive ».

Dans ce mode; le nœud peut émettre et recevoir, mais après une transmission, il doit suspendre la transmission et attendre 8bits de plus qu'un nœud en mode erreur active pour pouvoir retransmettre

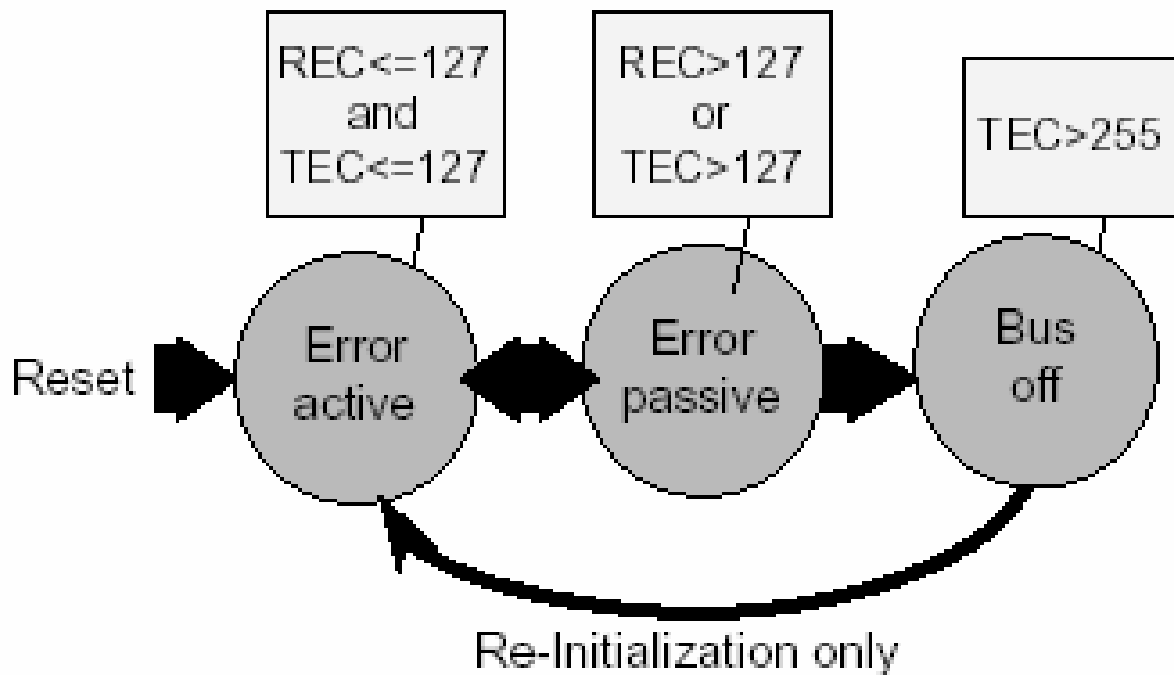
Si les compteurs dépassent la valeur 255, le nœud passe dans le mode « Bus off ».

L'activité du nœud est arrêtée, il ne participe plus à la communication.

Seul un RESET permet de repasser en mode erreur active (127 trames de onze bits récessifs).

III. Le bus CAN (control Area Network)

Gestion du mode d'erreur



III. Le bus CAN (control Area Network)

Gestion du mode d'erreur : gestion du compteurs de réception (REC)

- ➡ Le compteur de réception est incrémenté de 1 si :
- un récepteur détecte une erreur, le compteur d'erreur de réception sera incrémenté de 1, sauf si l'erreur est un Bit Error durant un flag d'erreur active ou un flag de surcharge.
- ➡ Le compteur de réception est incrémenté de 8 si :
- un récepteur reçoit un bit dominant juste après un flag d'erreur.
 - un récepteur voit un Bit Error tandis qu'il reçoit un flag d'erreur active ou un flag de surcharge.

III. Le bus CAN (control Area Network)

Gestion du mode d'erreur : gestion du compteurs d'émission (TEC)



Le compteur de transmission est incrémenté de 8 si :

- un émetteur envoie un flag d'erreur, son compteur d'émission s'incrémente de 8 sauf si l'émetteur est en erreur passive et voit un Acknowledgment Error (il ne détecte aucun bit dominant sur le champ d'acknowledge ou lors de l'envoi de la trame d'erreur passive) et également si l'émetteur envoie un flag d'erreur lors d'une erreur de Bit-Stuffing durant la période d'arbitrage (détection d'un bit dominant au lieu d'un bit récessif situé après le RTR).**
- un émetteur voit un Bit Error tandis qu'il émet un flag d'erreur active ou un flag de surcharge.**

III. Le bus CAN (control Area Network)

Gestion du mode d'erreur : gestion des compteurs REC et TEC

- ➡ Les compteurs d'émission et de réception s'incrémentent de 8 si :
- chaque nœud recevant sept bits dominants consécutifs après réception d'un Active Error Flag, d'un Passive Error Flag ou d'un Overload Flag. Après détection de quatorzième bit dominant consécutif (pour l'Active Error Flag ou l'Overload Flag) ou du huitième bit dominant consécutif suivant le Passive Error Flag, et après toutes les suites de huit bits dominants consécutifs.

III. Le bus CAN (control Area Network)

Gestion du mode d'erreur : gestion des compteurs REC et TEC

➡ Le compteur de réception est décrémenté de 1 si :

- le récepteur reçoit une trame sans erreur (jusqu'au champ d'Acknowledge Slot) et si la valeur du compteur est comprise entre 1 et 127. Si le compteur est à 0, sa valeur ne change pas (pas d'incréméntation). S'il est supérieur à 127, sa valeur est ramenée entre 119 et 127.

➡ Le compteur d'émission est décrémenté de 1 si :

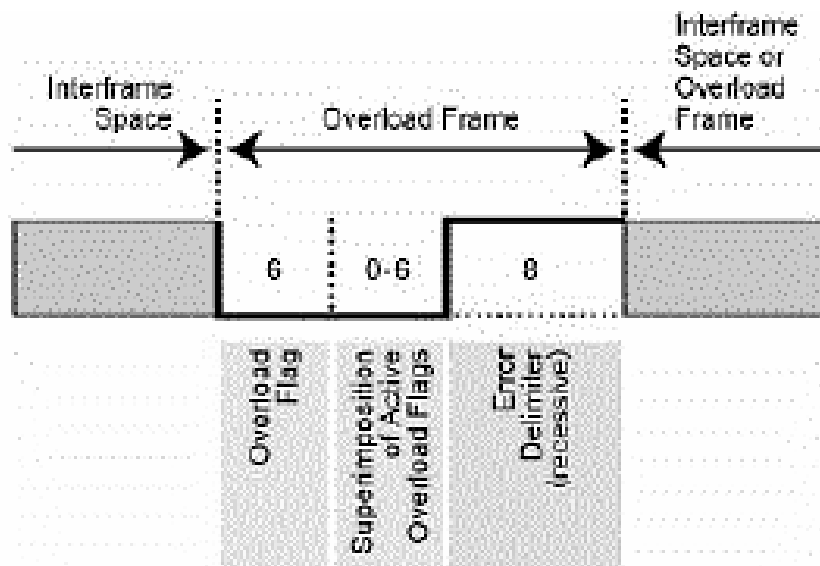
- la transmission d'une trame se déroule sans erreur (jusqu'au champ d'Acknowledge Slot). Si la valeur du compteur est à 0, le compteur ne s'incréménte pas.

III. Le bus CAN (control Area Network)

Trame de surcharge (overload frame)

La trame de surcharge indique aux autres nœuds qu'une station est surchargée. Elle est formée de deux champs :

- le drapeau de surcharge (Overload Frame) avec six bits dominants,
- le délimiteur de surcharge (Overload Delimiter) avec huit bits récessifs.



III. Le bus CAN (control Area Network)

Trame de surcharge (overload frame)

Une trame de surcharge est émise sur le bus si :

- un bit dominant est détecté durant la période d'intertrame.
- 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).

Dès qu'une trame de surcharge est émise, les autres nœuds voient sur le bus une suite de six bits dominants qui ne respectent pas la règle du Bit-Stuffing. Ils émettent à leur tour une trame de surcharge.

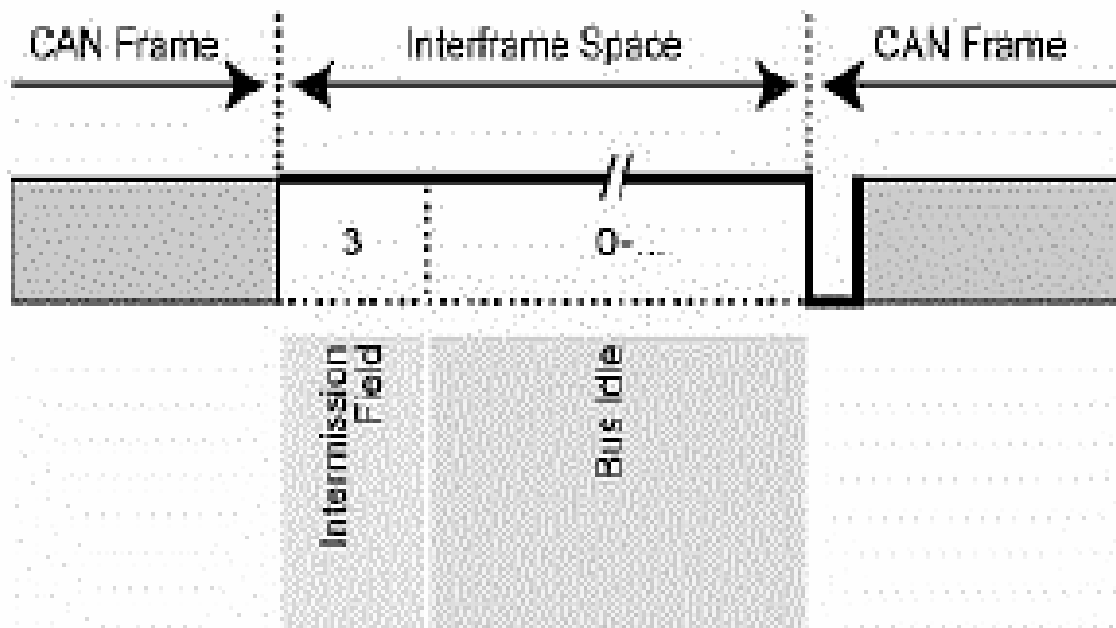
Seulement deux trames de surcharges consécutives sont autorisées sur le bus (pas plus de 12 bits dominants consécutifs émis sur le bus).

III. Le bus CAN (control Area Network)

Période inter-trame

Elle sépare les trames de données ou de requêtes entre elles. Il s'agit d'une suite de plusieurs bits récessifs. Elle est divisée en deux parties:

- le champ d'intermission
- le bus idle



III. Le bus CAN (control Area Network)

Période inter-trame

Le champ d'intermission :

Le champ d'intermission est une suite de 3 bits récessifs consécutifs. Durant la période d'intermission, l'émission de trame n'est pas autorisée. Les gestionnaires de protocole ne sont autorisés à signaler que les conditions de surcharge.

Le champ de Bus Idle :

Le champ de Bus Idle est celui du bus quand il est au repos. Le niveau de repos est le niveau récessif et aucune trame ne circule sur le bus (succession de 5 bit récessifs).

Le champ de suspension de transmission :

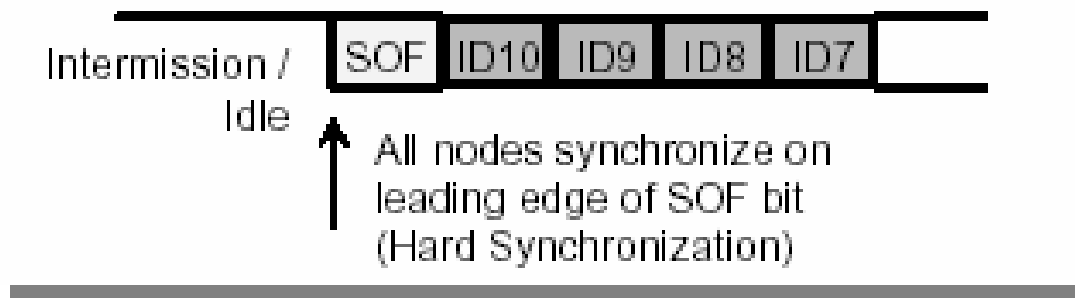
Le champ de suspension de transmission est émis par un nœud lorsque celui-ci envoie une trame d'erreur passive.

III. Le bus CAN (control Area Network)

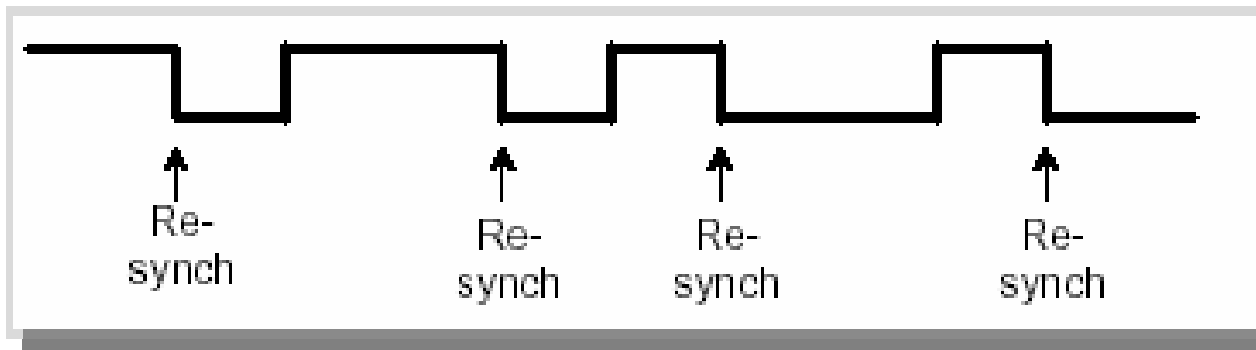
La synchronisation du bus

La synchronisation se fait de 2 façon :

- Hard : sur le bit de start

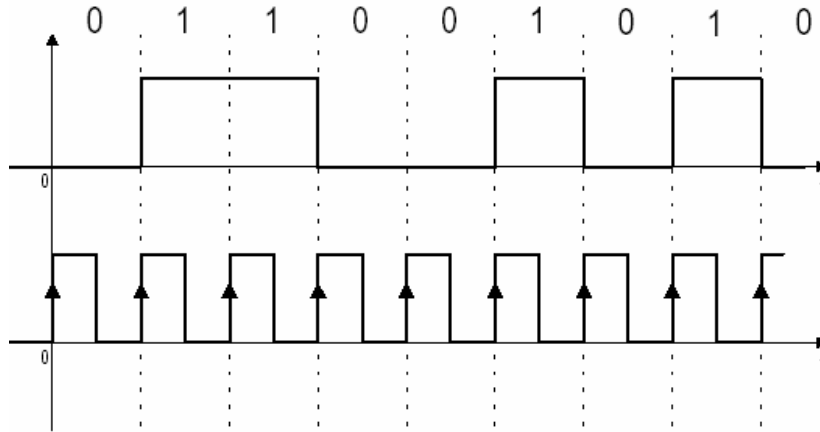


- Re- synchronisation sur chaque front récessif et dominant



III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time »



Les transitions des bits s'effectuent sur chaque front montant de l'horloge. Dans le protocole CAN, une période d'horloge correspond à ce que l'on appelle le Nominal Bit Time.

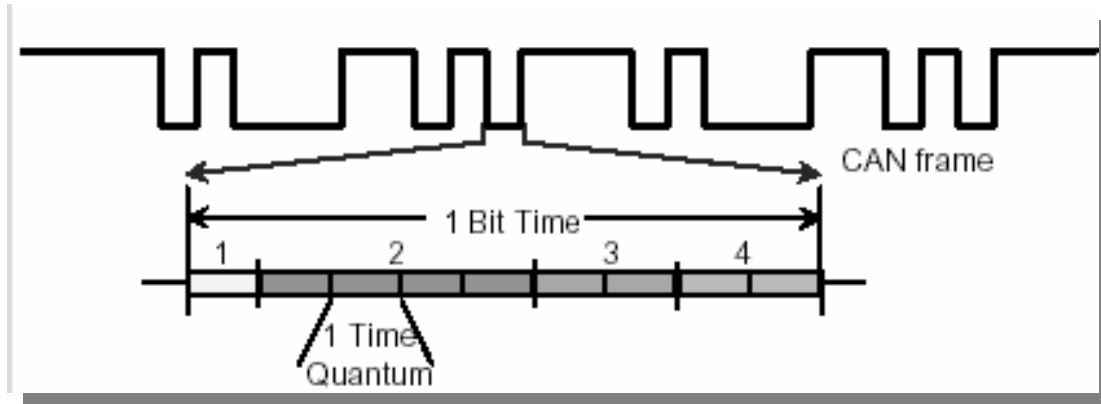
Il représente en fait la durée du bit sur le bus. Cette durée est, comme nous l'avons vu, étroitement liée à la période de l'horloge. Chaque station reliée sur le bus doit être cadencée avec le même Nominal Bit Time pour pouvoir émettre et recevoir correctement les données circulant sur le bus.

III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time »

La durée du bit time de chaque circuit est construite à partir d'un nombre déterminé de périodes d'horloge issue de l'horloge interne de chaque circuit CAN. On définit ainsi un « Time Quantum » qui est une discrétisation plus petite utilisée par le bus CAN

- La longueur du Time Quantum est programmée par la division de l'oscillateur interne du composant d'interface.
- Il y a entre 8 et 25 time quanta par Bit.
- La longueur d'un bit (bit Time) et donc le débit (Bit Rate) est obtenu en programmant la longueur du quantum et en programmant le nombre de quantum par Bit.

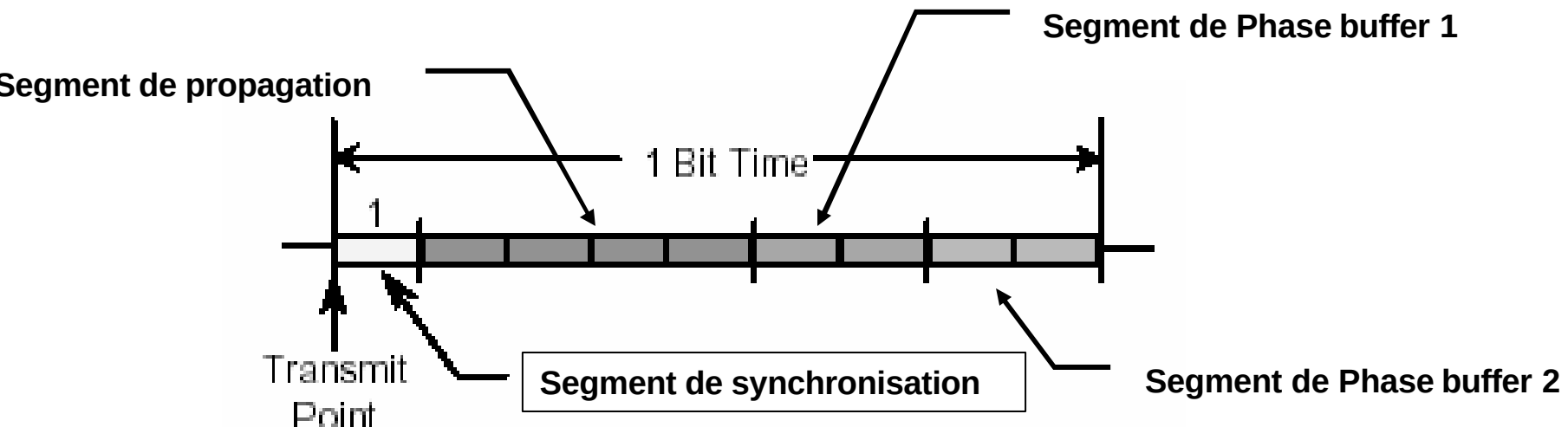


III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time »

La norme BOSCH décrit avec précision la composition de ce Nominal Bit Time qui est divisé en plusieurs segments :

- le segment de synchronisation (SYNC_SEG),
- le segment de propagation (PROP_SEG),
- le segment de phase buffer n°1 (PHASE_SEG1),
- le segment de phase buffer n°2 (PHASE_SEG2).

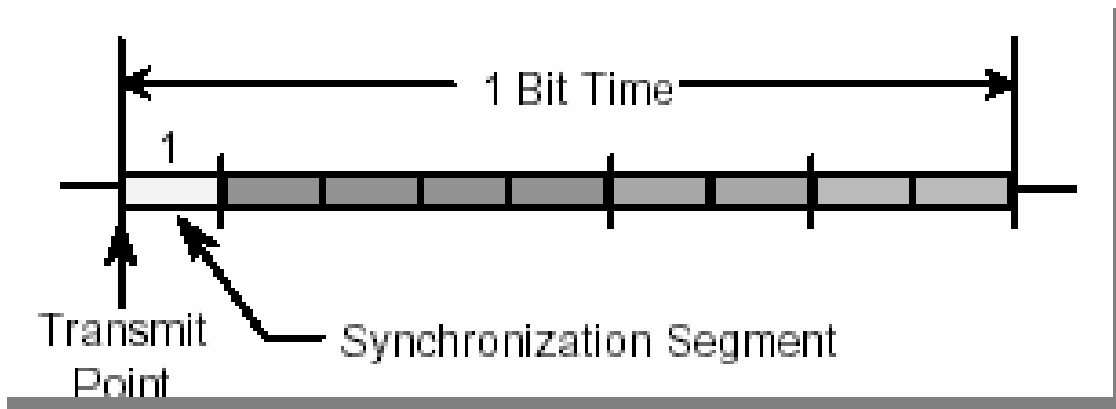


III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : segment de synchronisation

Le segment de synchronisation est utilisé pour synchroniser les différents nœuds du bus. Comme nous le verrons par la suite, une transition (de 0 à 1 ou de 1 à 0) doit s'effectuer dans ce segment pour permettre une re-synchronisation des horloges des différents nœuds en mode de réception de trames.

➡ La durée de ce segment est toujours de 1 quantum



III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : segment de synchronisation

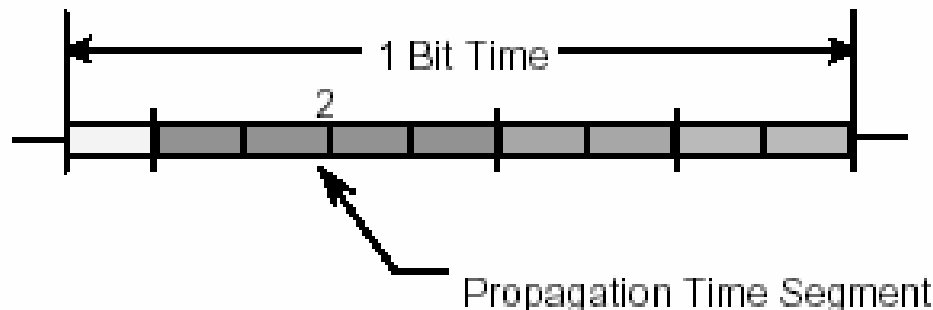
Le segment de propagation est utilisé pour compenser les phénomènes de temps de propagation sur le bus.

Il prend en compte le délai introduit par le driver.

La longueur de ce segment varie entre 1 et 8 quanta.



Durée PROP SEG= 2 (t propag bus + t retard comparateurs + t retard drivers lignes)*



III. Le bus CAN (control Area Network)

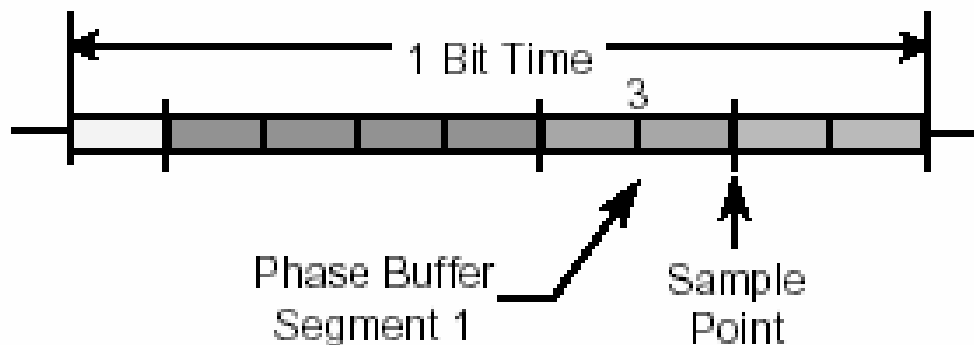
La notion de « Nominal Bit Time » : Le segment de buffer phase1

Il est surtout utilisé pour compenser les erreurs de phase détectées lors des transitions.

Ce segment peut être rallongé pendant la re-synchronisation.

Il est composé de 1 à 8 quanta.

Le point d'échantillonnage (Sample point) est le point auquel le bus est lu, Il est situé à la fin du segment de buffer 1



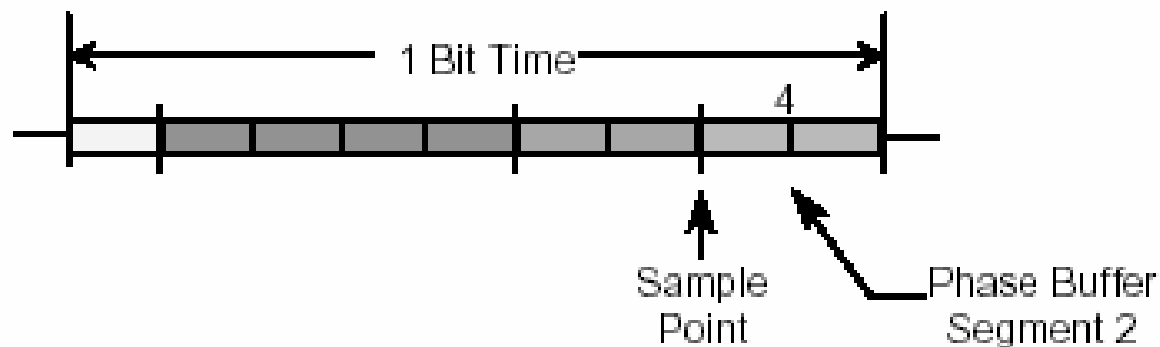
III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : Le segment buffer phase2 :

Il est surtout utilisé pour compenser les erreurs de phase détectées lors des transitions.

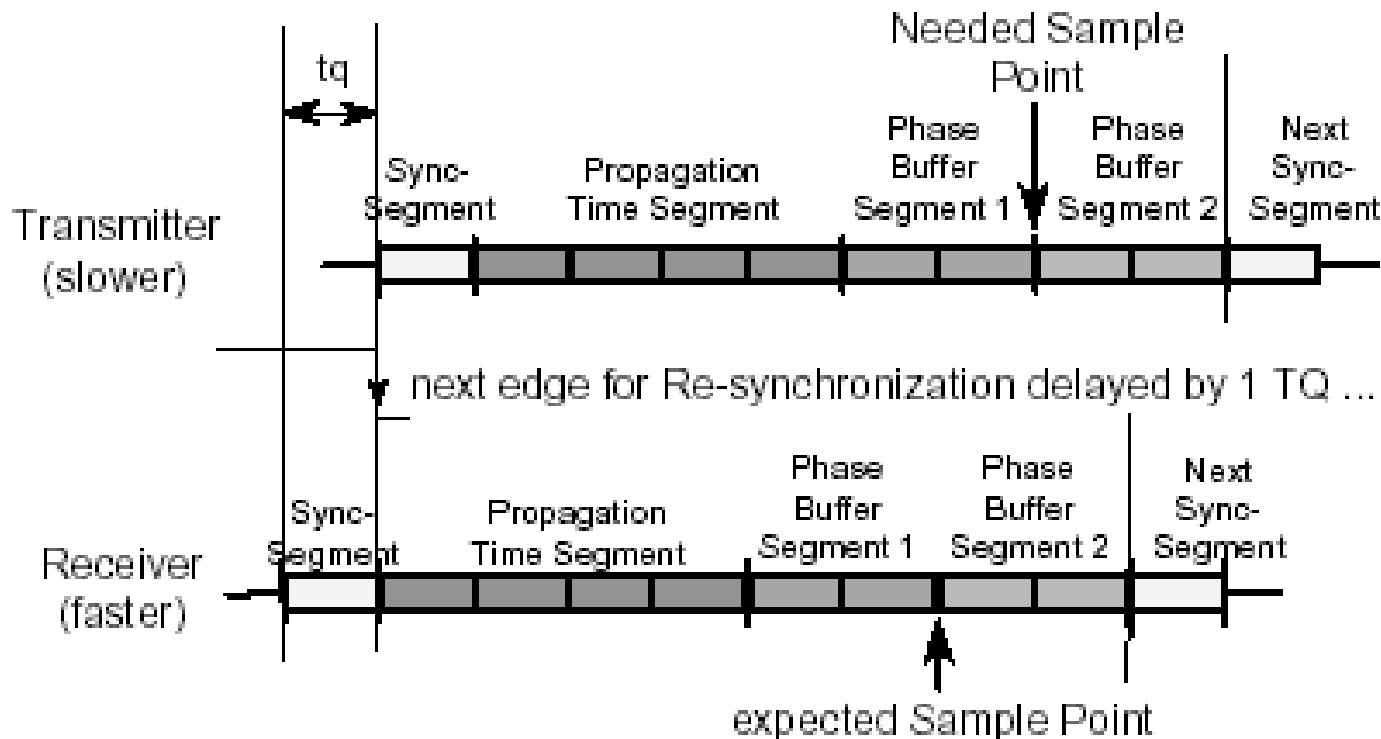
Ce segment peut être raccourci pendant la re-synchronisation.

Il est composé de 1 à 8 quanta et doit être au moins aussi long que le temps de traitement de l'information, sans être plus long que la phase buffer 1.



III. Le bus CAN (control Area Network)

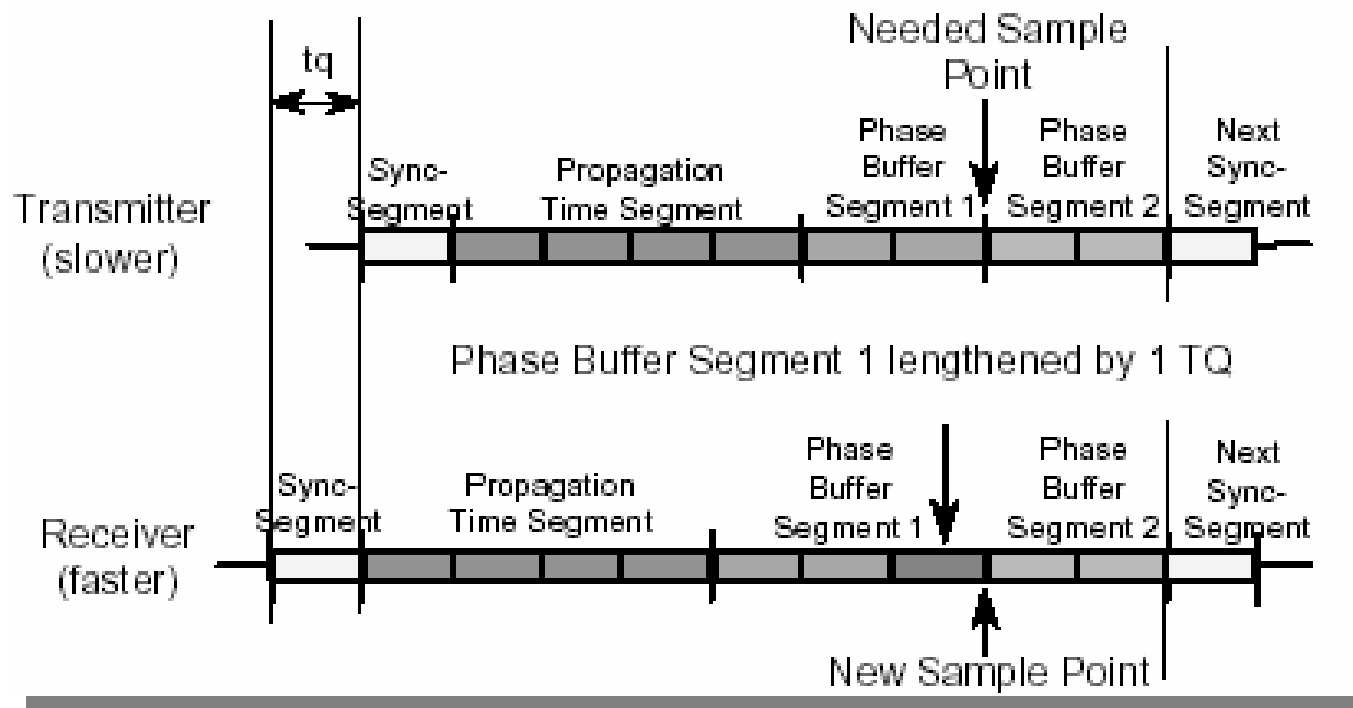
La notion de « Nominal Bit Time » : exemple de re-synchronisation



Ici l'oscillateur de l'émetteur est plus lent que celui du récepteur

III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : exemple de re-synchronisation



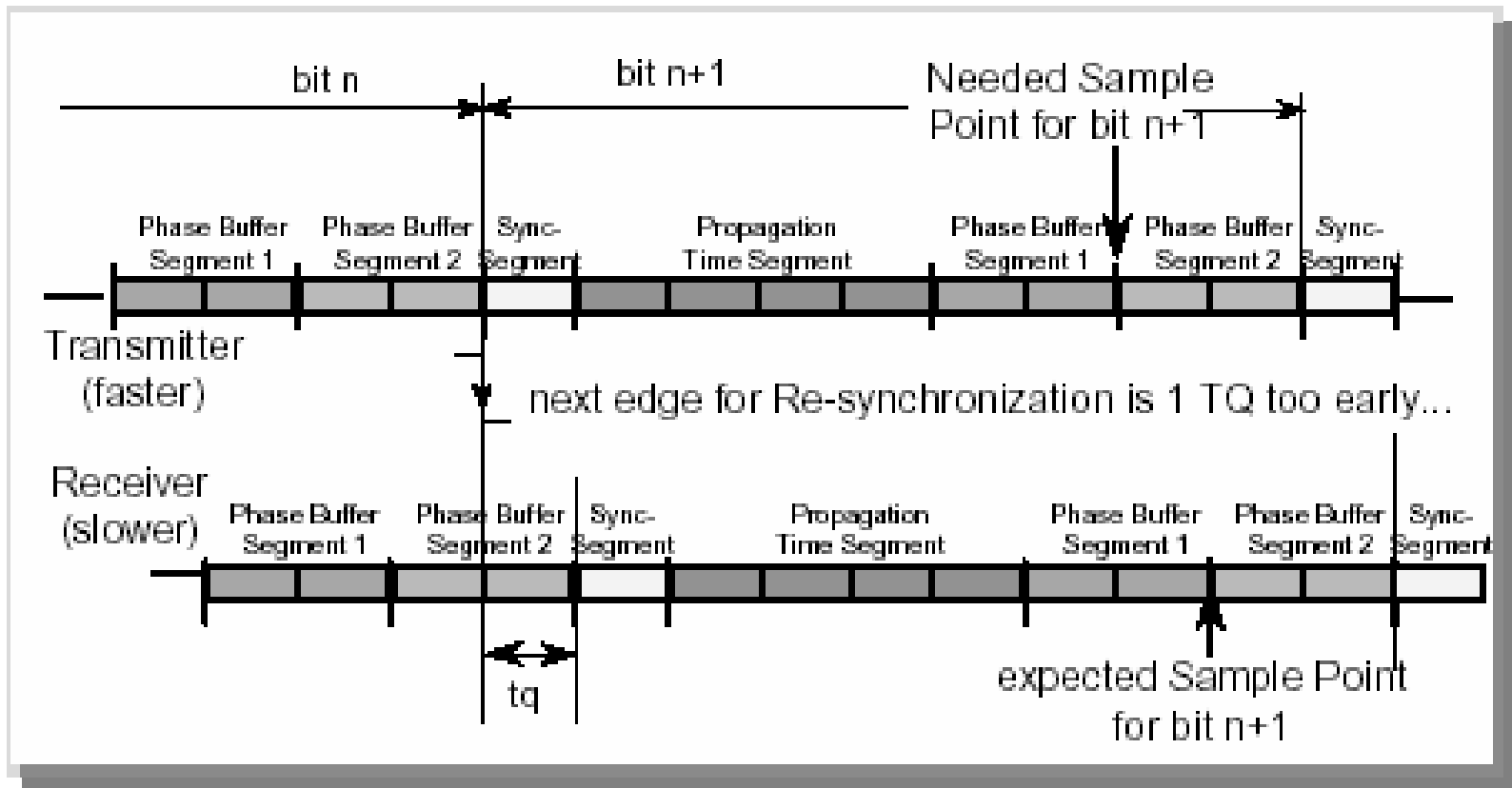
Ici l'oscillateur de l'émetteur est plus lent que celui du récepteur



Ajout d'un quantum au niveau du buffer1

III. Le bus CAN (control Area Network)

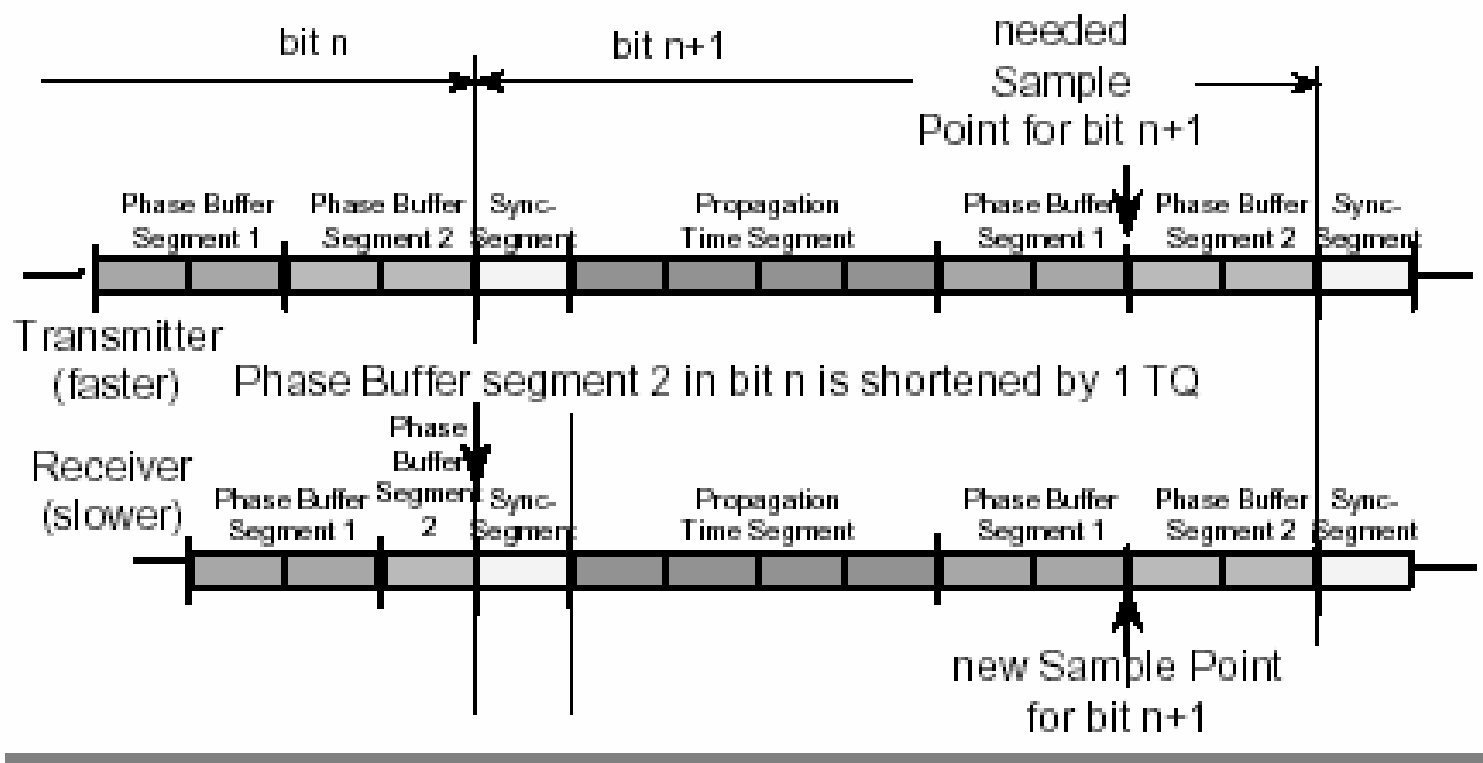
La notion de « Nominal Bit Time » : exemple de re-synchronisation



Ici l'oscillateur de l'émetteur est plus rapide que celui du récepteur

III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : exemple de re-synchronisation



Ici l'oscillateur de l'émetteur est plus rapide que celui du récepteur

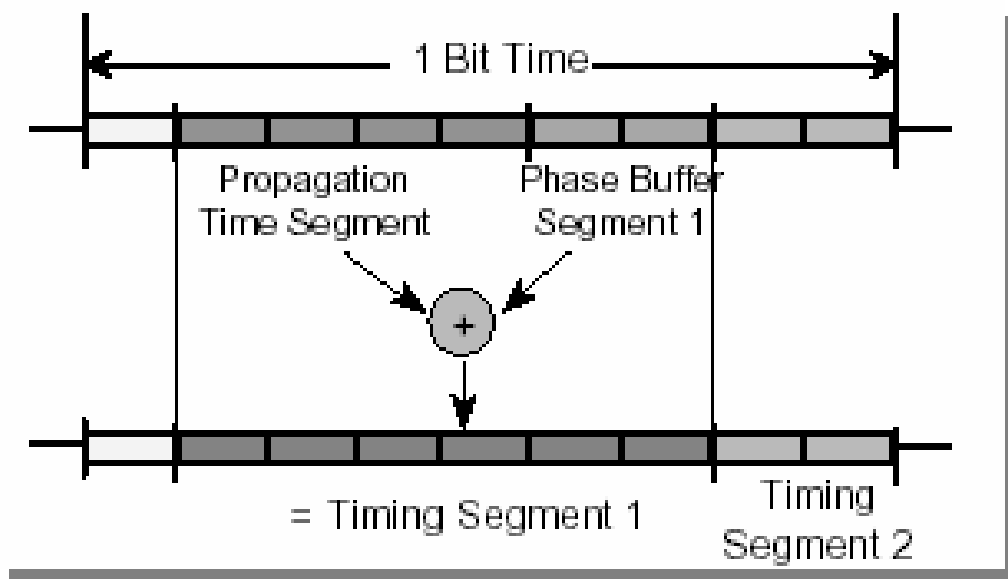
➡ le buffre 2 est raccourci

III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : bit timing

Dans l'implémentation de beaucoup de CAN, le segment de propagation et le segment de buffer 1 sont regroupés pour une programmation plus simple on crée les nouveaux segments :

- timing segment 1
- timing segment 2



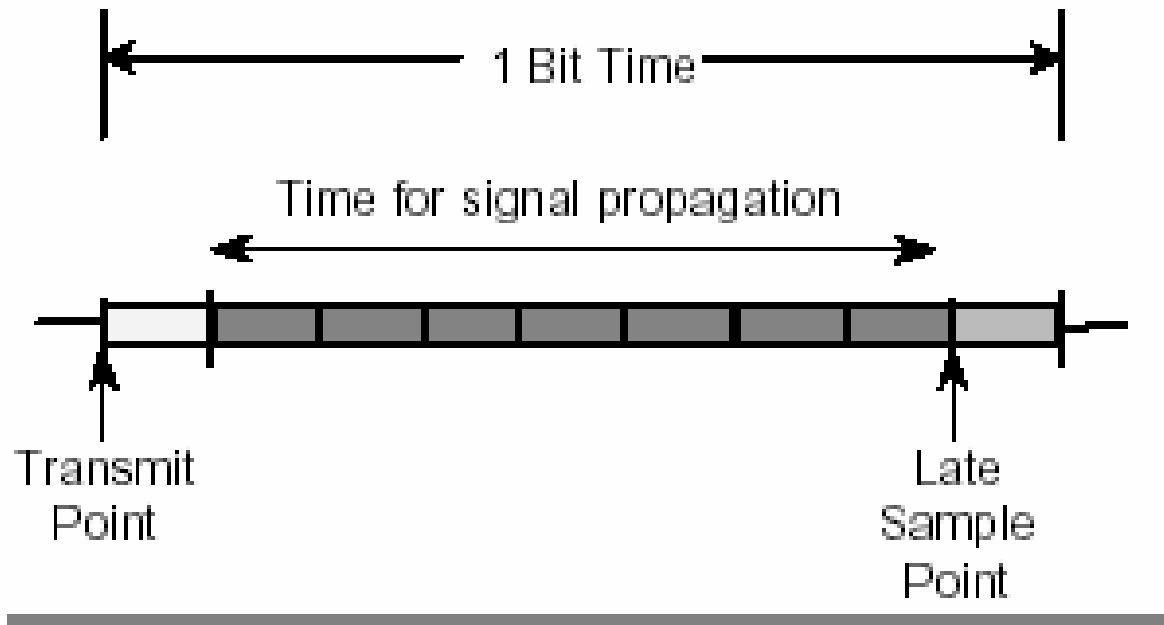
III. Le bus CAN (control Area Network)

La notion de « Nominal Bit Time » : choix de l'échantillonnage

On peut ainsi plus facilement programmer l'échantillonnage.

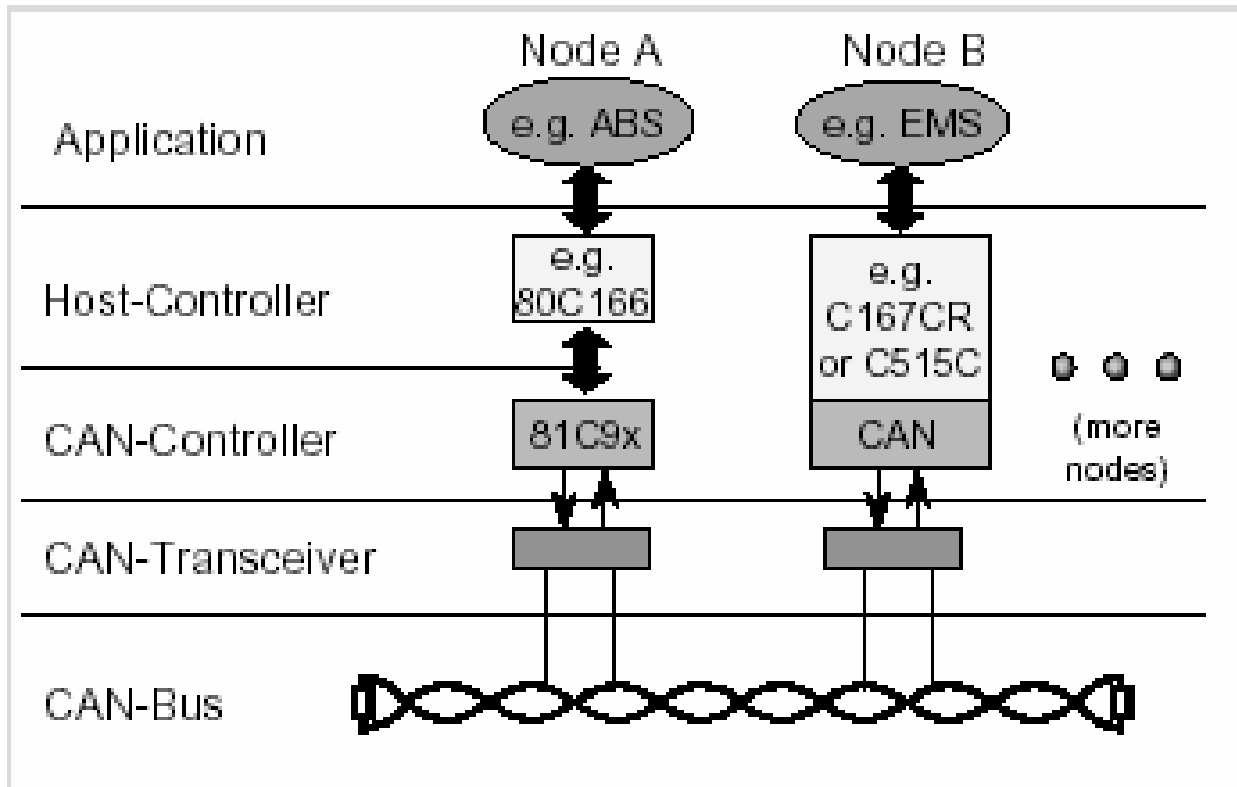
Attention le segment 2 ne peut pas dépassé 4 quanta

Ceci permet d'avoir une taille de bus maximale.



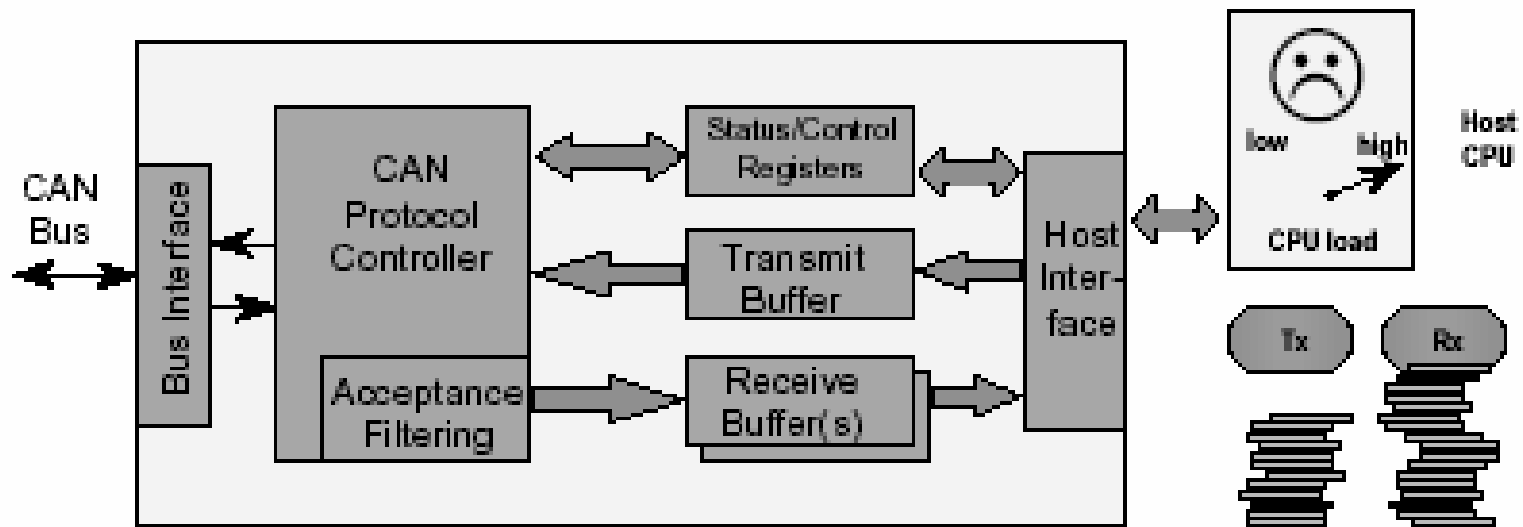
III. Le bus CAN (control Area Network)

Résumé : implémentation



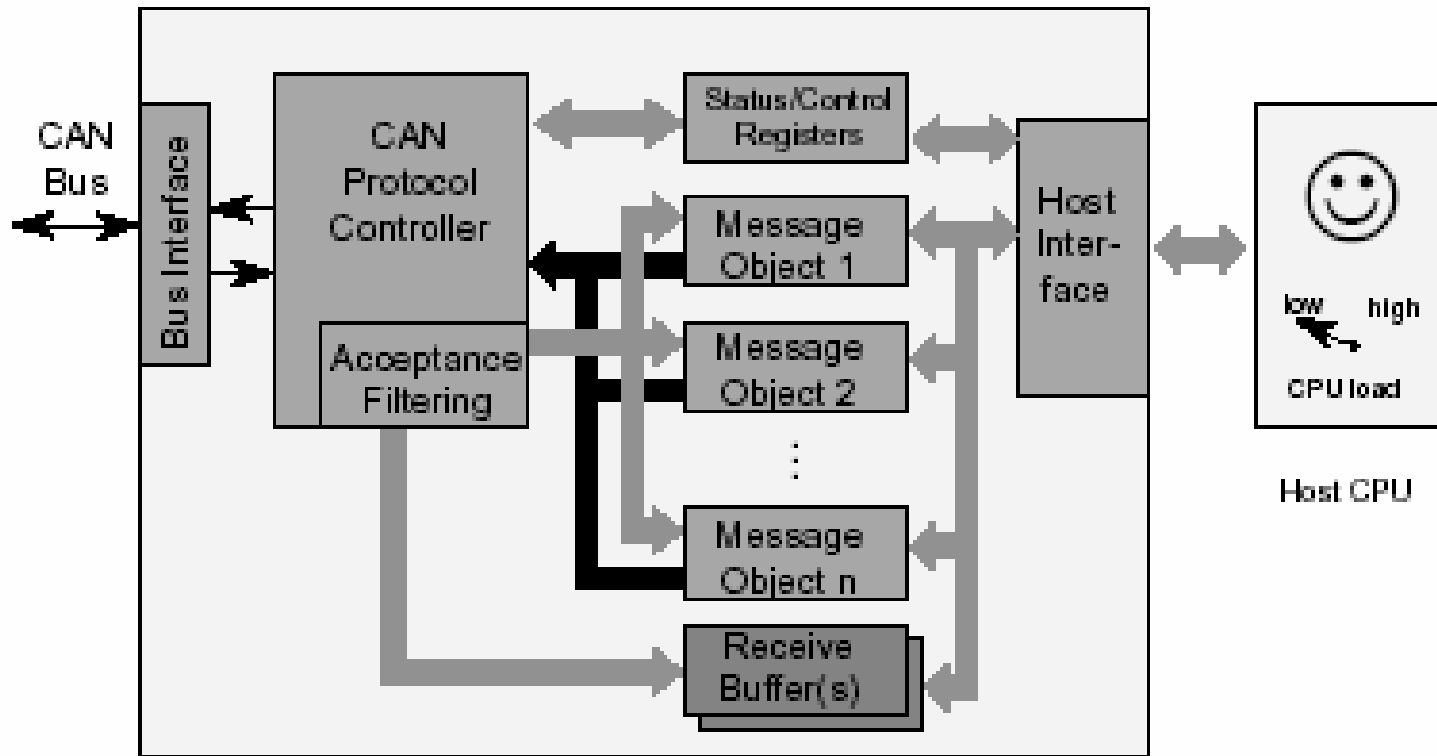
III. Le bus CAN (control Area Network)

Résumé : implémentation le contrôleur CAN basic



III. Le bus CAN (control Area Network)

Résumé : implémentation le contrôleur CAN complet



➔ **Stockage et gestion des messages**

III. Le bus CAN (control Area Network)

Résumé (composant siemens)

