

# Le réseau CAN (Controller Area Network)

Nicolas NAVET

INRIA Lorraine - projet TRIO

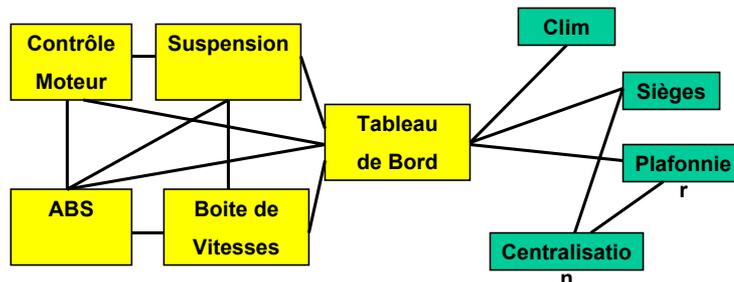
[nnavet@loria.fr](mailto:nnavet@loria.fr) - <http://www.loria.fr/~nnavet>

Certaines images de ce cours proviennent de :

- spécification CAN (CAN in Automation - <http://www.can-cia.de>)
- Cours de P. Koopman (<http://www.ece.cmu.edu/~ece540/lecture/>)
- Slides CAN In Automation (<http://www.can-cia.de>)

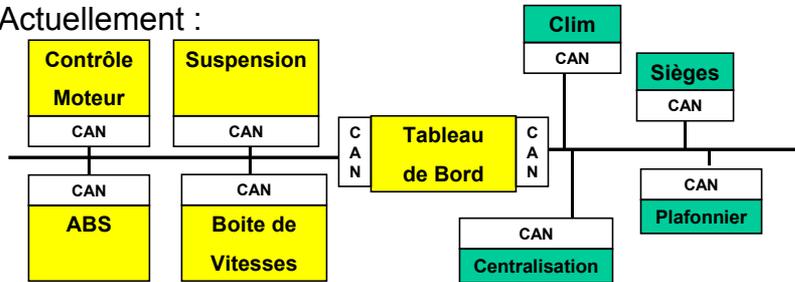
## CAN: un réseau pour l'automobile (1/4)

- CAN conçu pour le **multiplexage véhicule**  
(conception BOSCH 1983 – normalisation ISO 1994)
- Pourquoi multiplexer les communications ?  
Réduction des coûts de câblage et de maintenance
- Situation à la fin des 80's:



## CAN: un réseau pour l'automobile (2/4)

- Actuellement :



CAN High Speed  $\geq 250\text{kb/s}$

CAN Low Speed

- Quelques chiffres :
  - R25 (fin 80's) : 2 km de câbles cuivre ! >80kg
  - MB Classe C (2001) : 12 ECU's – 25 fonctions
  - MB Classe S (2002) : > 50 ECU's !

N. NAVET - ENSEM - Cours CAN

3

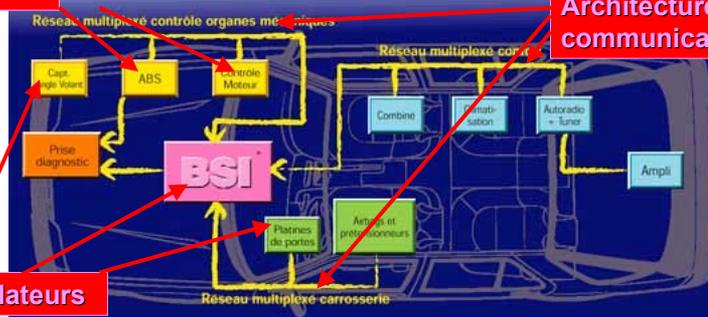
## CAN: un réseau pour l'automobile (3/4)

Des fonctions critiques

Schéma-type d'architecture Electrique/Electronique véhicule

Architecture de communication

Calculateurs



Exemple d'architecture PSA

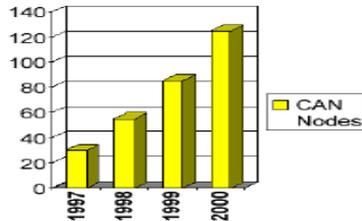
N. NAVET - ENSEM - Cours CAN

4

## CAN: un réseau pour l'automobile (4/4)

---

- Un marché énorme : 50 millions de véhicules par an



- Des contraintes de temps et de sûreté de fonctionnement importantes (ex: correcteur de stabilité, ABS, contrôle-moteur ...)

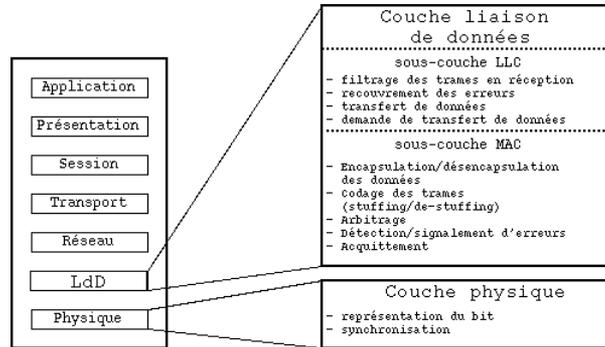
## Caractéristiques Techniques

---

- Bus à diffusion - technique CSMA
- MAC priorisé avec arbitrage non destructif
- Un identificateur unique par message :
  - priorité pour l'accès au bus
  - filtrage des messages en réception
- Retransmission automatique des trames corrompues
- Compteurs d'erreurs sur chaque contrôleur
- 8 octets de données au plus par trame

# CAN dans le modèle OSI

- CAN ne normalise que les couches LDD et physique !

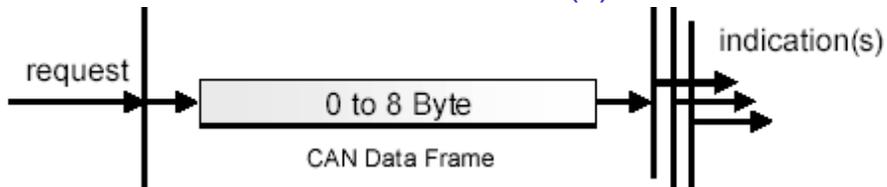


N. NAVET - ENSEM - Cours CAN

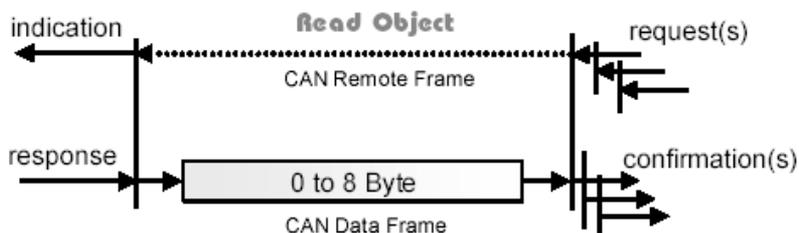
7

# Modèles de coopération

- Producteur – Consommateur(s)



- Client – Serveur



N. NAVET - ENSEM - Cours CAN

8

## Couche Physique : codage NRZ

- Non-Return to Zero (NRZ) Coding :



= 011010

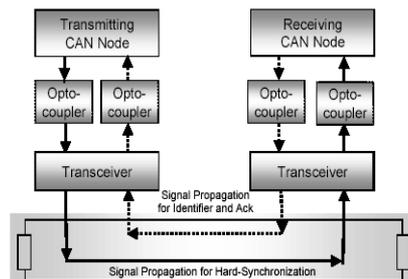
- Niveau logique 0 : **bit dominant**
- Niveau logique 1 : **bit récessif**

## Couche Physique : « et » logique

- En cas de transmissions simultanées :  
niveau 0 (dominant) + niveau 1 (récessif) = 0 sur le bus
- Le temps bit minimum est 2 x temps de propagation

qui est fonction (données CiA) :

- Contrôleurs CAN (50-62ns)
- Interface de ligne (120-250ns)
- Opto-coupleur (40-140ns)
- Câble (environ 5ns/m)



## Les supports physiques

---

- Standard ISO : paire torsadée
- Autres supports envisageables :
  - ✓ Support monofilaire
  - ✓ Courant porteur
  - ✓ Transmission radio
  - ✓ Infra-rouge
  - ✓ Fibre-optique
  - ✓ ...

## Couche Physique : débit vs longueur

---

- En pratique, on peut atteindre (réseau filaire) :

	<b>Débit</b>	<b>Longueur</b>	<b>Temps bit</b>
CAN High Speed ISO 11898	1 Mbit/s	30m	1us
	500kbit/s	100m	2us
	250kbit/s	250m	4us
CAN Low Speed ISO 11519-2	125kbit/s	500m	8us
	10kbit/s	5km	100us

## Couche Physique : Bit-Stuffing (1/2)

- Bit-Stuffing (taille=5) pour créer des fronts sur le signal

Suite de bits à transmettre



Au niveau du médium



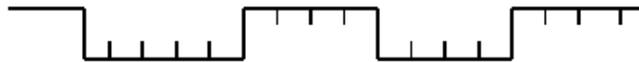
Après «dé-stuffing»



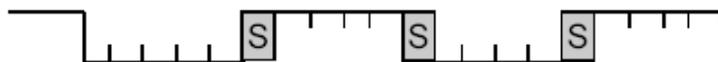
## Couche Physique : Bit-Stuffing (2/2)

- Bit-Stuffing : le pire cas

Unstuffed bit-sequence



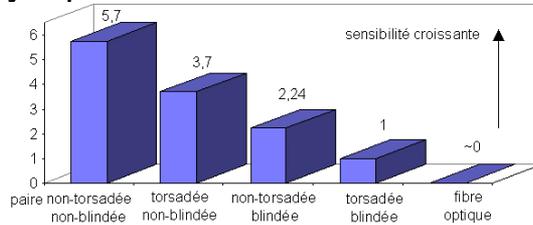
stuffed bit-sequence



Accroissement max de  $\lfloor (n-1)/4 \rfloor$  bits

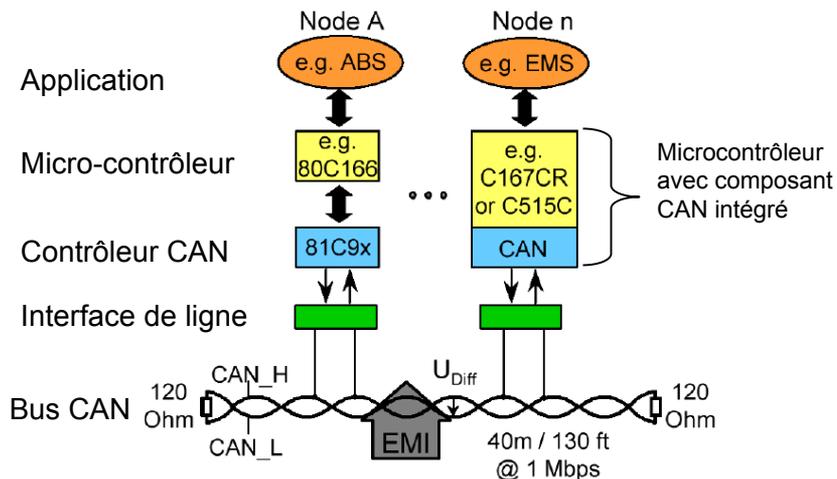
## Couche Physique : sensibilité EMI

- Fct du support physique :



- Fct du débit du bus : + élevé ➡ + de perturbations
- Fct de l'environnement d'utilisation :
  - perturbations d'origine interne
  - Perturbations d'origine externe (radars, haute tension, FM ..)

## Implémentations CAN typiques



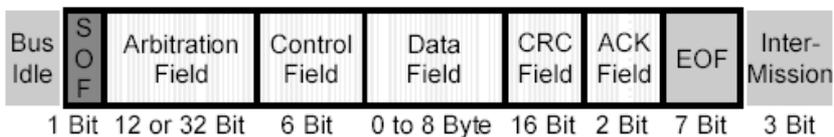
## Les différents types de frame

---

- Trame de données (*data frame*)
  - CAN standard (2.0A) : Ident. sur 11 bits
  - CAN étendu (2.0B) : Ident. sur 19 bits
- Trame de demande de données (*remote transmission request - RTR*)
- Trame d'erreurs (*error frame*)
- Trame de surcharge (*overload frame*)

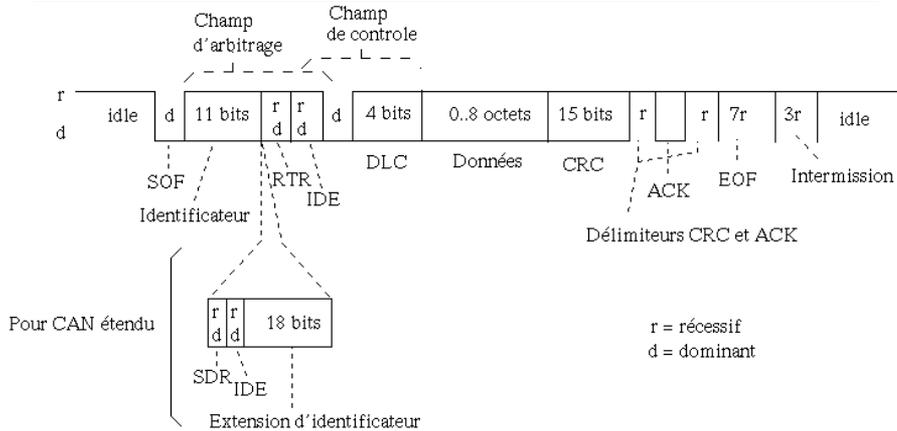
## Format de la trame de données (1/2)

---



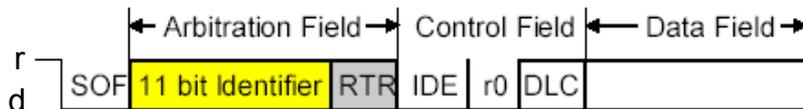
- CAN standard (2.0A) : identificateur de 11 bits  
En théorie: 2048 id différents, en pratique 2032 .. (id dans [2033,2048] interdits)
- CAN étendu (2.0B) : identificateur de 29 bits  
> 500 millions d'id différents

## Format de la trame de données (2/2)

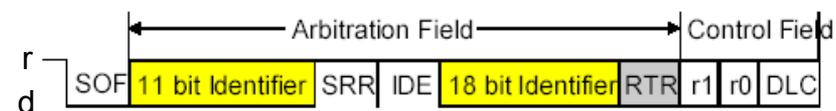


## Chap d'arbitrage

- CAN standard (2.0A)



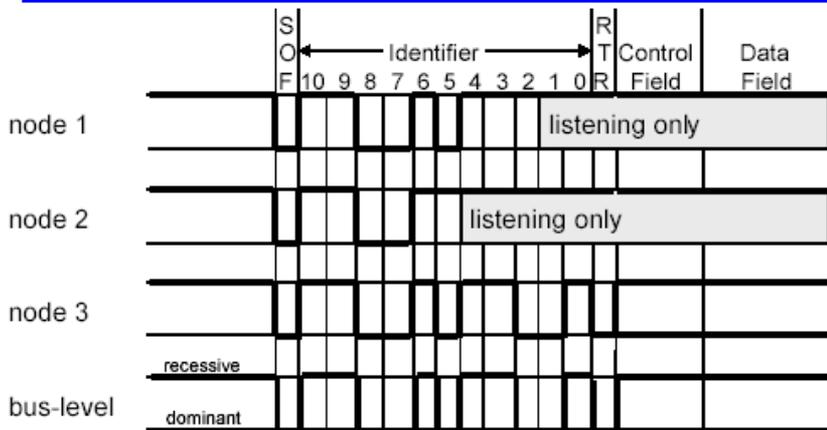
- CAN étendu (2.0B)



## L'accès au bus (1/2)

- La phase d'arbitrage ou de « résolution des collisions » : la trame la plus prio. gagne le bus
- Se fait sur les champs **Identificateur + RTR**
- **Principe** : chaque station émet puis écoute, si la valeur lue est différente de la valeur émise, la station sait qu'elle a perdu l'arbitrage
- **Conséquence** : un aller-retour pour le signal avant l'émission d'un nouveau bit d'où limite sur le débit max.

## L'accès au bus (2/2)



La station 3 remporte l'arbitrage

## Champs de contrôle et de données

---

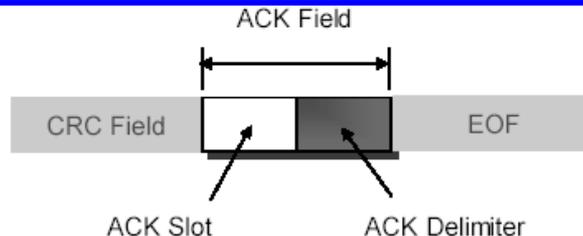
- Champ de contrôle :



- Bit RTR : 0 trame de données, 1 trame de demande.
- IDE(2.0A)/r1(2.0B) : dominant (pour CAN 2.0B, le champ IDE est un champ d'arbitrage)
- r0 : dominant
- DLC : taille des données (octets)
- Champ de données : de 0 à 8 octets

## Champs d'acquittement

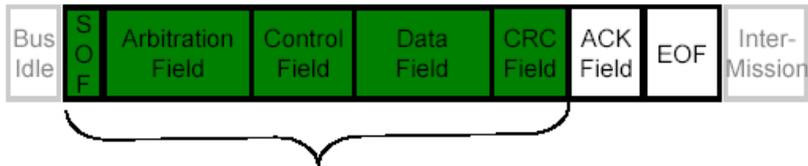
---



- **ACK Slot** : Émis récessif – toute station qui reçoit la trame écrit un bit dominant
- Signification de l'acquittement : au moins une station a reçue la trame mais pas forcément le destinataire !! **L'acquittement est non-fiable**

## Champs soumis au bit-stuffing

---



Bit-Stuffing sur ces champs uniquement

## Exercices

---

1. A un instant donné, le bus devient libre et 2 trames d'identificateurs 31 et 29 (émises respectivement par les stations 1 et 2) sont en concurrence. Représenter les bits émis par la station 1 et 2 et le niveau résultant sur le bus.
2. Quelle est la durée maximale de transmission d'une trame CAN (2.0A) sur un réseau à 125kbit/s ?
3. Sur un bus CAN à 500kbit/s, quel débit utile (données) peut-on espérer ?
4. Est-il possible d'envisager de transmettre le signal qui ordonne le déclenchement d'un airbag sur un réseau CAN à 125kbit/s ?

## Trame de demande de données

---

- Bit RTR à 1 (d'où une priorité inférieure à la trame de données!)
- Ne contient pas de données
- La réponse n'est pas « écrite » dans la trame
- Permet une coopération de type Client-Serveur
- Induit une surcharge sur le réseau
- Aucune garantie sur le délai de la réponse !

## Les erreurs de transmission

---

- Pas de technique de correction automatique
- **Principe** : une station qui détecte une erreur, la signale aux autres par une trame d'erreur (6 bits dominants)
- La trame corrompue participera à un prochain arbitrage (mais ne gagne pas en priorité)
- Probabilité d'erreur résiduelle très faible (de l'ordre de  $10^{-12}$ )

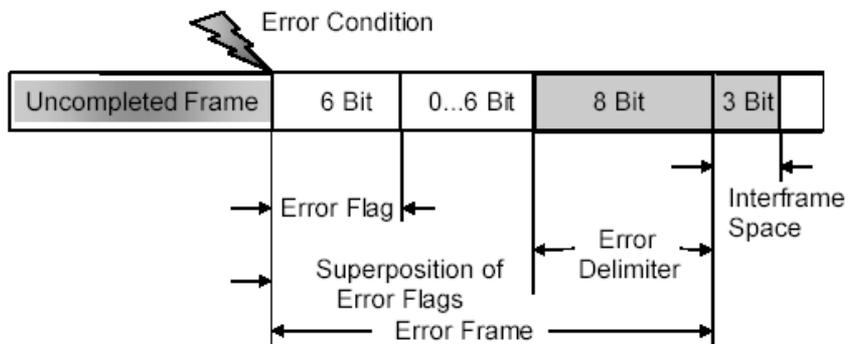
## Les ≠ types d'erreurs

---

- **Bit-stuffing** : 6 bits consécutifs même niveau
- **Bit-error** : dominant + récessif = récessif !
- **CRC error** : CRC calculé ≠ CRC frame
- **Acknowledgement error** : pas d'acquittement
- **Form error** : mauvaise valeur pour un champ fixe

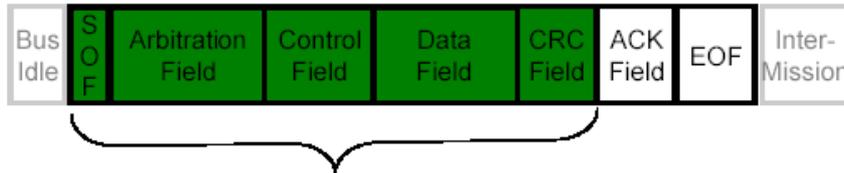
## La trame d'erreur

---



- **Meilleur cas** : retransmission après 17 bits
- **Pire cas** : retransmission après 23 bits

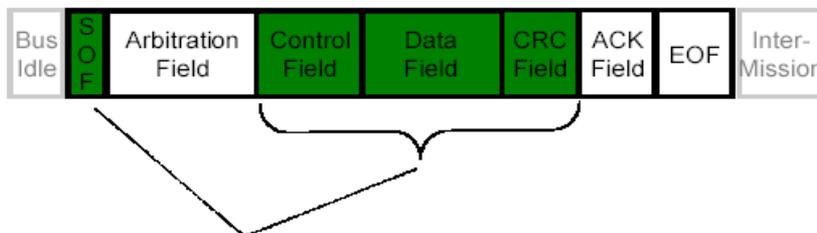
## Erreur de bit-stuffing



Bit-stuffing uniquement sur ces champs

- 6 bits consécutifs de même niveau entraînent une erreur de bit-stuffing
- La trame d'erreur permet de propager l'erreur à toutes les stations ...

## Erreur du niveau d'un bit

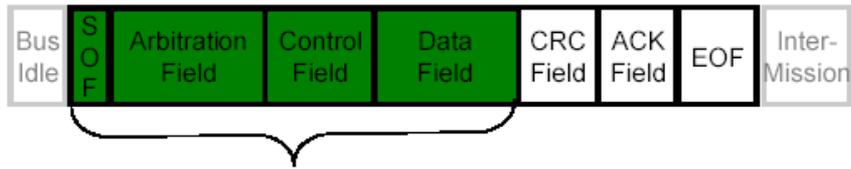


vérification uniquement sur ces champs

- **Bit-error** = émission d'un bit dominant et réception d'un bit récessif

## Erreur de CRC

---

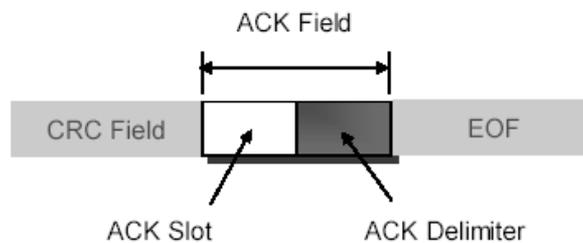


CRC calculé sur ces champs

- **Erreur de CRC** : le CRC reçu est différent du CRC calculé

## Erreur d'acquittement

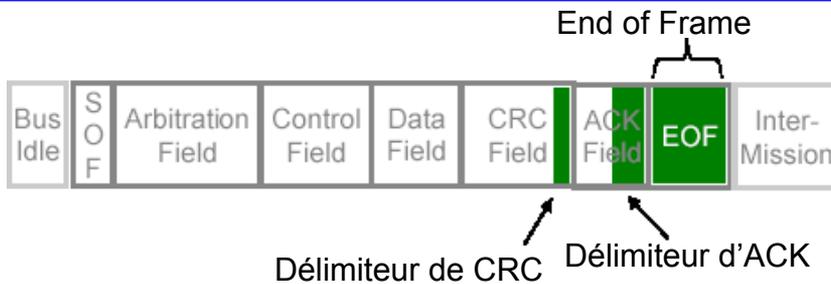
---



- **Erreur d'acquittement** : ACK slot récessif

## Erreur « de forme »

---



- **Erreur de forme** : un champ dont la valeur est fixée par le protocole n'a pas la valeur attendue

## Le confinement d'erreurs

---

- **Problème sur CAN** : une station défectueuse peut perturber le fonctionnement de tout le système .. (ex: envoi ininterrompu de trames d'erreurs)
- **Une solution** : les stations «défectueuses» se déconnectent automatiquement (ou limitent leur prérogatives comme le signalement d'erreurs)
- **Comment détecter un pb ? 2 compteurs d'erreurs de transmission sur chaque contrôleur**
  - sur les trames émises (**TEC** – transmit error counter)
  - sur les trames reçues (**REC** – receive error counter)

## Les 3 états d'une station

---

Selon la valeur des compteurs :

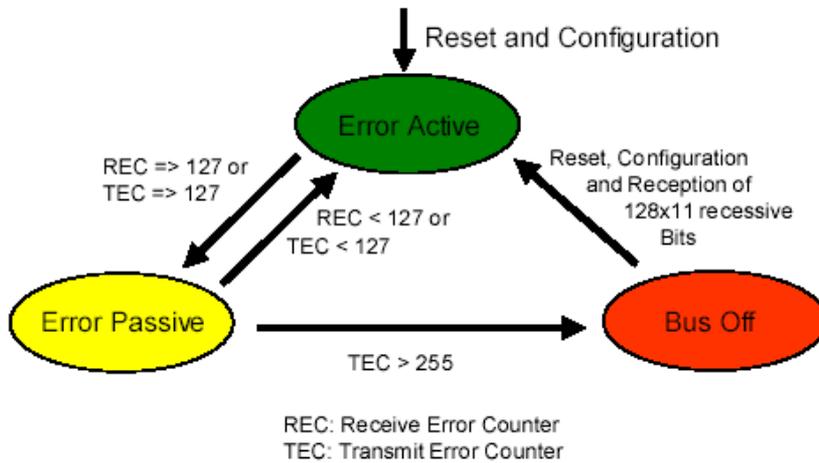
- **Etat Erreur-active** : fonctionnement normal
- **Etat Erreur-passive** :
  - émission possible mais 8 bits après que le bus soit libre (temps de réponse !!)
  - plus de signalement d'erreurs
- **Etat Bus-off** :
  - la station se déconnecte du bus (plus d'émission ni de réception)

## Évolution des compteurs

---

- **REC** :
    - Réception d'une trame corrompue : +1 (jusque 128)
    - Réception d'une trame correcte : -1 (si >0)
  - **TEC** :
    - Emission d'une trame corrompue : +8 (jusque 256)
    - Emission d'une trame correcte : -1 (si >0)
- ⇒ Il existe quelques exceptions mineures à ces règles (ex: quand une station est seule sur le réseau)

## Les règles de changement d'états



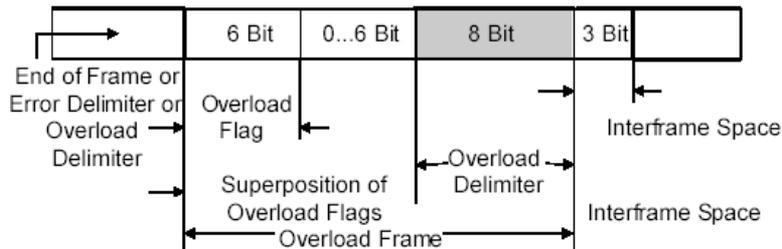
## Conclusion sur le confinement d'erreurs

Le confinement est un plus pour la sûreté de fonctionnement mais ...

- de fortes EMI peuvent conduire à bus-off ou erreur passive sans dysfonctionnement hardware
- le concepteur doit évaluer les risques et prévoir des changements de mode de marche (ex: que faire si le contrôle-moteur est bus-off ??)
- des études ont montrés que le REC ne servait à rien si la station émettait .. (on passe toujours en erreur-passive plus tôt avec le TEC)

## Trame de surcharge

- Une station émet une trame de surcharge (6 bits dominants) pour signaler qu'elle ne peut recevoir la prochaine trame (pas plus de 2 consécutivement)



⇒ En pratique très rarement utilisé ..

## Un « bug » du protocole

Certaines trames peuvent être reçues en double !

1. une perturbation localisée à certaines stations (dont l'émetteur) sur le dernier bit du champ EOF : retransmission ..
2. Toutes les stations non-affectées reçoivent 2x la trame

Conséquences : sur un réseau CAN

- ne jamais utiliser de messages on/off
- pas d'incrément relatifs (+20°)

Vraisemblablement pas de validation formelle du protocole ...

## Une application de multiplexage

---

⇒ Actuellement, 2 sous-réseaux et une douzaine d'ECUs (cf. slides introductifs)

- le réseau « moteur » pour le contrôle temps-réel du véhicule (CAN  $\geq$  250kbit/s)

- Le réseau habitacle ou confort (CAN / VAN / J1850 souvent  $\leq$ 125kbit/s)

⇒ Dans l'avenir, un réseau haut-débit multimédia (ethernet switché, IEEE 1394 ?) et un réseau dédié X-by-Wire (TTP, FlexRay ?)

## Une messagerie « moteur » typique

---

- réseau CAN à 250kbit/s - charge  $\approx$  20%

Trame	Site émetteur	DLC	Période (ms)
1	CM (contrôle moteur)	8	10
2	CAV (capteur angle volant )	3	14
3	CM	3	20
4	BVA (boite vitesse automatique)	2	15
5	ABS	5	20
6	ABS	5	40
7	ABS	4	15
8	Calcul carrosserie	5	50
9	Suspension	4	20
10	CM	7	100
11	BVA	5	50
12	ABS	1	100

## Les contraintes de temps

---

- Informations transmises ont une durée de vie limitée (eg : vitesse, angle volant etc..)
  - Les contraintes temporelles
    - sont issues de la dynamique du véhicule
    - induisent des problèmes d'ordonnement
- ⇒ CAN bien adapté: arbitrage basé sur la priorité permet de garantir des bornes sur le temps de réponse (étudié dans le cours d'ordonnement temps réel)

## Comment «valider» une application ?

---

- ⇒ Valider = vérifier le respect des contraintes
- Mesures sur un système réel ou maquette : coûts, difficultés d'instrumentation, tard dans le cycle de vie
  - La simulation : résultats pire-cas généralement sans valeur
  - Modèles analytiques : peu aptes à refléter toute la complexité d'un système

## Techniques de validation: + et -

---

Validation sur Modèles		Validation sur Prototypes
+ coûts et rapidité de développement - Hypothèses simplificatrices		+ pas d'hypothèse simplificatrice + certaines contraintes ne peuvent être vérifiées que sur prototype ! - construction, instrumentation longues et coûteuses - intervient tard dans la phase de conception - résultats orientés pire-cas généralement sans validité
Modèles Analytiques	Modèles de Simulation	
+ validité du modèle facile à vérifier + bien adaptés aux contraintes dans le pire-cas et aux événements rares - hypothèses simplificatrices fortes - difficultés de modélisation !	+ moins d'hypothèses simplificatrices + facilité de modélisation - résultats orientés pire-cas généralement sans validité - validité des modèles !	

## Techniques analytiques utiles dans la phase de conception

---

- Calcul de bornes sur les temps de réponse
- Fixation optimale des priorités
- Calcul du nombre d'erreurs tolérables par une trame
- Calcul de la probabilité de non-respect des échéances
- Calcul du temps moyen d'atteinte des états bus-off et erreur-passive

## Conclusions

---

- CAN standard dans l'industrie automobile, pourquoi ?
  - bien adapté aux exigences temps réel du domaine
  - très bon marché
  - simple d'utilisation
- Mais :
  - débit limité (par la technique d'accès au bus)
  - peu de services: en particulier, peu de fonctionnalités relatives à la sûreté de fonctionnement