

---

**ARTIGO****TRANSMISSÃO DE MENSAGENS E GERENCIAMENTO DE ERROS EM UMA REDE CAN AUTOMOTIVA<sup>1</sup>****Saulo Marcos Torres de Carvalho<sup>2</sup>  
Gustavo Lobato Campos****RESUMO**

O objetivo deste artigo é ilustrar como é feito o processo de transmissão de mensagens em uma rede CAN. Além de caracterizar o sistema de gerenciamento de erros no CAN, serão apresentados os tipos de erros existentes e os mecanismos para a detecção de erros na rede, juntamente com um mecanismo para o confinamento de falhas. Além disso, este documento propicia um entendimento a respeito da carga do barramento em uma rede CAN, e como esta é influenciada pela quantidade de mensagens, centrais eletrônicas e erros presentes na rede. Este artigo também conta com uma sucinta apresentação do barramento CAN e do formato das mensagens para melhor entendimento do mesmo.

**Palavras-chave:** Barramento CAN. *Controller Area Network* (CAN). Carga do barramento. Identificador (ID). Processo de arbitração. Rede CAN.

**1 INTRODUÇÃO**

O protocolo *Controller Area Network* (CAN) foi criado pela Bosch em meados da década de 80. Seu objetivo a princípio era prover uma maior eficiência na comunicação de redes automotivas com um baixo custo de implementação. No entanto, com o passar dos anos e com o desenvolvimento desta tecnologia, o protocolo CAN tem sido usado em aplicações em diversas áreas, tais como: processos e controle industriais, aplicações no agronegócio, dispositivos médicos, sistemas automotivos e até mesmo em alguns sistemas aviônicos (DI NATALE, 2012).

O motivo pelo qual a rede CAN tem sido vastamente utilizada em diversas aplicações se deve a sua característica robusta e confiabilidade. Em suma, o protocolo CAN possibilita que a rede apresente as seguintes características (DI NATALE, 2012):

- prioridade de mensagens;

---

<sup>1</sup> **Como citar este artigo:** CARVALHO, Saulo Marcos Torres de; CAMPOS, Gustavo Lobato. Transmissão de mensagens e gerenciamento de erros em uma rede CAN automotiva. **ForScience**: revista científica do IFMG, Formiga, v. 6, n. 1, e00341, jan./jun. 2018.

<sup>2</sup> **Autor para correspondência:** Saulo Marcos Torres de Carvalho, e-mail: saulo.eng.eletrica@gmail.com

- tempos de latência garantidos;
- flexibilidade na configuração;
- consistência de dados;
- capacidade *multicast*;
- acesso multimestre ao barramento;
- detecção e sinalização de erros automaticamente;
- retransmissão automática de mensagens;
- elevadas taxas de transferências de dados.

Baseado em um sistema *broadcast*, o protocolo CAN foi desenvolvido para operar em velocidades de 20 kbps até 1 Mbps (DI NATALE, 2012). Por esta razão, o protocolo foi padronizado para aplicações de alta velocidade (até 1 Mbps) *International Standard Organization* (ISO) 11898-2 (ISO, 2003b) e para aplicações de baixa velocidade (até 125 kbps) ISO 11898-3 (ISO, 2006; ASSIS, 2011).

Na indústria automotiva, a rede CAN geralmente é usada em aplicações de alta velocidade (500 kbps), pois possibilita uma maior quantidade de mensagens trafegando no barramento para um mesmo intervalo de tempo e aproxima-se de uma comunicação em tempo real. Por exemplo, a informação de velocidade no odômetro do veículo precisa ser atualizada constantemente em um curto espaço de tempo sem o que usuário perceba, por isto, a velocidade de comunicação e a periodicidade com que essa mensagem é enviada é de suma importância para garantir este comportamento. Porém, a rede CAN de baixa velocidade (até 125 kbps) é utilizada em sistemas como, controle de temperatura, módulo de rádio, painel de instrumentos, módulo de gerenciamento das portas e outros (DAVIS, 2007).

O padrão mais utilizado em veículos comerciais é provido pela *Society of Automotive Engineers*, o SAE J1939, o qual emprega o padrão CAN como camada física, identificador estendido de até 29 bits, taxa de transmissão 250 kbps, comunicação *broadcast* e *peer-to-peer*, funções de diagnóstico, gerenciamento de rede, além de definir os parâmetros a serem utilizados por veículos comerciais, como por exemplo, a taxa de transmissão de dados. Existem vários padrões derivados do SAE J1939, que usam suas funções básicas, no entanto, eles utilizam diferentes grupos de parâmetros e uma camada física modificada de acordo com sua aplicação. Exemplos, ISO 11783, *National Marine Electronics Association* 2000 (NMEA), ISO 11992 e *Fleet Management System* (FMS) (JUNGER, 2010).

Como supracitado, a comunicação em uma rede CAN é feita por um barramento *broadcast*, baseado em uma topologia em linha com um barramento linear, em que um

número de *Engine Control Units* (ECU's), ou centrais eletrônicas de controle, são conectadas via interface CAN. Em um barramento CAN, uma ECU também pode ser referida como um nó CAN (VECTOR E-LEARNING, [201-]).

De forma geral, a transmissão de dados em um barramento CAN realiza-se via troca de mensagens. Essas podem ser transmitidas periodicamente, esporadicamente ou por evento. Uma mensagem pode conter de 0 a 8 bytes de dados, além disso, o conteúdo das mensagens em nó no barramento CAN é dado pelo campo identificador (*Identifier*), ou simplesmente ID, que é usado tanto para filtragem das mensagens de interesse de determinada ECU, assim como para definir a prioridade da mensagem no barramento (TINDELL, 1995).

Uma das vantagens da utilização de uma rede CAN é a robustez de seu mecanismo de detecção de falhas. Para detectar mensagens corrompidas, o protocolo CAN define cinco mecanismos: *Bit Monitoring*, *Bit Stuffing*, *Cyclic Redundancy Check*, *ACK Check* e *Form Check* (VECTOR E-LEARNING, [201-]). Estes serão detalhados posteriormente, na seção deste artigo sobre gerenciamento de erros.

Em suma, o foco deste trabalho é apresentar os principais conceitos da rede CAN, além de, proporcionar um estudo detalhado do processo de transmissão de mensagens e erros existentes em um barramento CAN.

## 2 ESTRUTURA DA MENSAGEM

No protocolo CAN existem quatro tipos de quadro (DI NATALE, 2012):

- **Quadro de dados** – contém dados de um transmissor para um receptor.
- **Quadro remoto** – usado para solicitar que outro nó transmita uma mensagem com o mesmo ID.
- **Quadro de erro** – transmitido por qualquer nó que detecta um erro no barramento.
- **Quadro de sobrecarga** – gera um atraso extra entre os quadros de dados e/ou remotos.

Além disso, são definidos dois formatos de quadros CAN, que se diferenciam pelo tamanho do identificador. Desta forma, o formato padrão (CAN 2.0A), suporta um ID de 11 bits, Figura 1. Enquanto, o formato estendido (CAN 2.0B) suporta ID's de até 29 bits, Figura 2 (COOK, 2007).

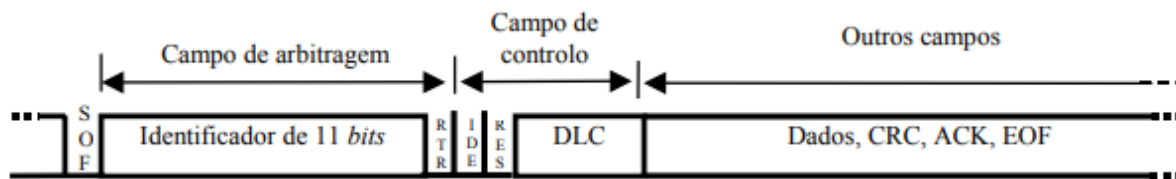


Figura 1 - Formato de um quadro CAN  
Fonte: Metrôlho (1999).

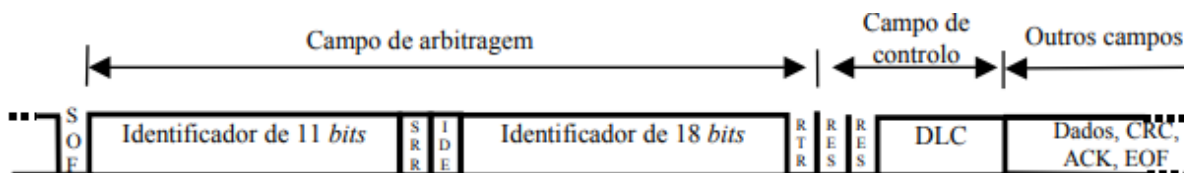


Figura 2 - Formato de um quadro CAN  
Fonte: Metrôlho (1999).

Por um outro lado, o quadro de erro difere significativamente da estrutura do quadro de dados e remoto, como pode ser observado na Figura 2. Ele é transmitido sempre que o barramento detecta um erro (VECTOR E-LEARNING, [201-]). De fato, ele não é um verdadeiro quadro CAN, mas sim o resultado da sinalização de um erro até sua recuperação (DI NATALE, 2012). Em síntese, se um nó que está transmitindo ou recebendo uma mensagem detectar um erro, ele irá abortar imediatamente a transmissão e enviar ao barramento um quadro de erro que consiste em seis bits dominantes e oito bits recessivos (COOK, 2007). Mais detalhes deste quadro serão discutidos na sessão de gerenciamento de erros.

A Figura 3 representa o formato de um quadro de dados podendo este ser o formato padrão ou o estendido. Adicionalmente, um quadro remoto é similar ao quadro de dados, exceto pela à ausência do campo de dados.

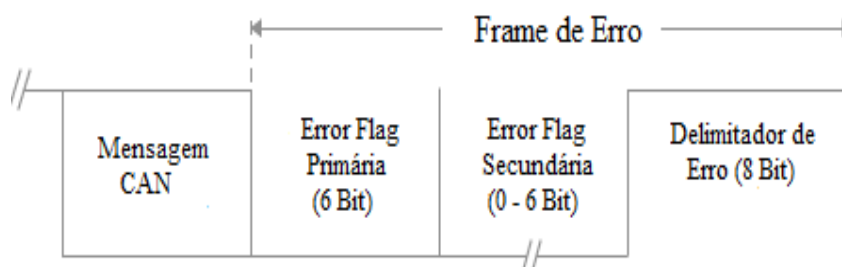


Figura 3 - Formato de um quadro CAN  
Fonte: Vector e-learning ([201-]).

### 3 O BARRAMENTO CAN

A transmissão de sinais em uma rede CAN é baseada na utilização de tensões diferenciais. Isso faz com que o sistema seja imune a ruídos e interferências causadas pelos motores, sistemas de ignição, etc. (VECTOR E-LEARNING, [201-]).

O meio de transmissão físico constituído por um par de fios trançados é chamado de barramento CAN. Neste par trançado existem duas linhas de sinal: CAN\_H (CAN *High Line*) e CAN\_L (CAN *Low Line*). O par trançado também tem a função de reduzir a interferência causada pelas duas linhas de sinal CAN\_H e CAN\_L. Além disso, dois terminadores, que são dois resistores, geralmente de 120 ohms são conectados nos terminais do barramento para garantir a propagação dos sinais elétricos e prevenir o efeito de reflexão existente uma rede CAN de alta velocidade (VECTOR E-LEARNING, [201-]) como pode ser visto na Figura 4.

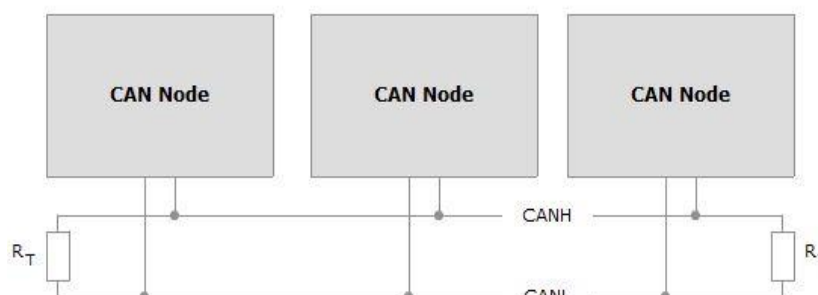


Figura 4 – Princípio de comunicação em uma rede CAN  
Fonte: Vector e-learning, ([201-]).

#### 3.1 Níveis de tensão em barramento CAN.

Para entender o processo de transmissão de uma mensagem em uma rede CAN é imprescindível entender os níveis de tensão com os quais a mesma trabalha.

De acordo com os padrões ISO 11898-2 e ISO 11898-3, um bit dominante é aquele que apresenta o “nível lógico 0” e um bit recessivo corresponde ao “nível lógico 1” (ISO, 2003b, 2006). Esses níveis lógicos de tensão são obtidos através da tensão diferencial entre as linhas CAN\_H e CAN\_L, como observado na Figura 4.

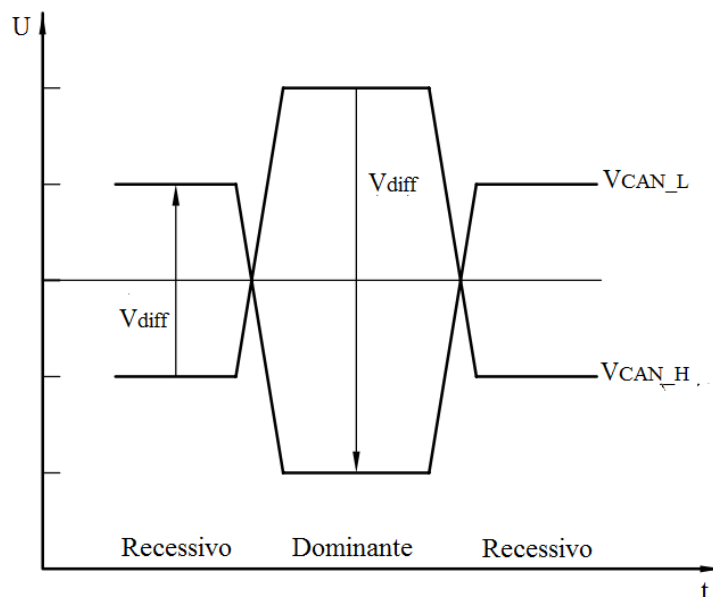


Figura 4 – Níveis de tensão em um barramento CAN  
Fonte: ISO ( 2003b).

Portanto, a ISO 11898-2 define valores de tensões diferenciais maiores que 0,9 Volts como um bit dominante, ao passo que quando a tensão diferencial é menor que 0,5 Volts corresponde a um bit recessivo (VECTOR E-LEARNING, [201-]). Em condições normais uma rede CAN de alta velocidade (*CAN High Speed*) o bit recessivo é definido quando a tensão diferencial é igual a zero. Para que isto ocorra tanto a tensão  $V_{CAN\_H}$  quanto a tensão  $V_{CAN\_L}$  devem assumir o mesmo valor de tensão nominal de 2,5 Volts. Já o bit dominante em condições nominais é obtido quando a tensão diferencial entre as linhas  $CAN\_H$  e  $CAN\_L$  é 2,0 Volts. Assim, a linha  $CAN\_H$  assume o valor nominal de 3,5 Volts, enquanto a linha  $CAN\_L$  assume o valor nominal de 1,5 Volts em relação a ao ponto de terra comum (ISO, 2003b).

De forma análoga, a ISO 11898-3 padroniza níveis de tensões diferenciais que definem os níveis lógicos de tensão para a rede CAN de baixa velocidade (*CAN Low Speed*). Neste caso, em condições nominais, o bit recessivo é obtido quando a tensão diferencial é -5,0 Volts. Para que isto ocorra a linha  $CAN\_H$  deve assumir o valor de 0 Volts e a linha  $CAN\_L$  o valor de 5 Volts, ambas em relação ao ponto de terra comum. Já para o bit dominante, a linha  $CAN\_H$  deve assumir o valor de tensão nominal de 3,6 Volts enquanto a linha  $CAN\_L$  deve apresentar o valor de tensão nominal de 1,4 Volts em relação à terra. Conseqüentemente, o bit dominante é obtido quando a tensão diferencial é 2,2 Volts para condição nominal (ISO, 2003b).

## 4 METODOLOGIA

Inicialmente foi feita uma pesquisa sobre os principais conceitos da rede CAN automotiva e as regulamentações ISO 11898-1, ISO 11898-2 e ISO 11898-3 que padronizam o protocolo.

O objetivo deste primeiro estudo foi justamente familiarizar-se com o protocolo, definições importantes sobre a estrutura da mensagem, barramento CAN e níveis de tensões usados para a comunicação da rede.

As definições supracitadas são de extrema importância para um bom entendimento do processo de transmissão de mensagens em um barramento CAN que constitui um dos principais assuntos deste trabalho.

Posteriormente, foi realizado um estudo detalhado sobre o processo de transmissão de mensagem em um barramento de CAN, bem como do gerenciamento das mensagens em caso de falha na transmissão de mensagens no barramento CAN.

Por fim foi feita uma análise sucinta dos principais fatores que afetam o desempenho de uma rede CAN automotiva.

## 5 RESULTADOS

Nesta seção apresenta-se um estudo detalhado sobre o processo de transmissão e gerenciamento de erros em barramento CAN, destacando-se os principais fatores que influenciam o desempenho de uma rede CAN automotiva.

### 5.1 Processo de transmissão

#### 5.1.1 Endereçamento

A rede CAN se baseia na combinação de uma arquitetura multimestre e uma topologia em linha. Assim, a rede CAN é dirigida por eventos e não por uma sequência pré-determinada de tempo. Desta maneira, transferências de dados em tempo real não são problema (VECTOR E-LEARNING, [201-]).

As mensagens quando são enviadas no barramento CAN ficam disponíveis para todos os nós, caracterizando um sistema *broadcast*. Portanto, é necessário que cada mensagem tenha um identificador único, que é responsável por caracterizar o conteúdo da mensagem,

prioritariamente, além de ser utilizado na filtragem das mensagens. É com base no ID que cada nó decide se aceita ou rejeita aquela mensagem, por meio de um processo de filtragem “*Acceptance Filtering*”, que varia de acordo com os dispositivos controladores utilizados (METRÔLHO, 1999).

Em uma rede CAN o filtro de aceitação (“*Acceptance Filter*”) utiliza o registrador “*Acceptance Mask*” em conjunto com as configurações de filtro para deixar passar um único ID ou um único intervalo de ID’s. O filtro serve tanto para mensagens no formato “*Standard*” quanto no formato “*Extended*” (EASYSYNC, 2016).

Em síntese, o filtro é controlado por dois registradores, via um algoritmo, sendo eles: *Acceptance Mask Register* (AMR) e *Acceptance Code Register* (ACR). Assim, os dados recebidos são comparados bit a bit com os valores armazenados no ACR. O AMR é responsável por determinar a posição dos bits, classificando-os em: (i) bits relevantes, que são os bits que repetem ou, (ii) bits não relevantes, que são os bits que não repetem no campo ID das mensagens que estão sendo comparadas. Eles são assinalados, no registrador, como bit 0 e 1, respectivamente. Feito isso o campo ACR, armazena o valor dos bits que repetem dos ID’s em comparação, ou seja, os bits relevantes, os demais bits ele assina como sendo “*don’t cares*”, X. Portanto, para receber uma mensagem todos os bits relevantes recebidos têm que corresponder aos respectivos bits do ACR (SEMICONDUCTORS, 2000). A Tabela 1, ilustra o processo contido no “*Acceptance Filter*” de um nó X na rede CAN:

Tabela 1 - Exemplo de um Filtro de Aceitação (*Acceptance Filter*) em um Nó X

	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
ID = 0x67	0	0	0	0	1	1	0	0	1	1	1
ID = 0xB4	0	0	0	1	0	1	1	0	1	0	0
AMR	0	0	0	1	1	0	1	0	0	1	1
ACR	0	0	0	X	X	1	X	0	1	X	X
[r] = relevante	[r]	[r]	[r]	[i]	[i]	[r]	[i]	[r]	[r]	[i]	[r]
[i] = irrelevante											

Fonte: Acervo do autor.

De acordo com o exemplo contido na Tabela 1, o campo AMR irá comparar os bits dos ID’s 0x67 e 0xB4, em suas respectivas posições, comparação bit a bit. Assim, ele irá assinar o bit 0 para os bits que repetem, e o bit 1 para um os que não repetem, classificando-os em relevantes e irrelevantes respectivamente, como pode ser visto acima. O campo ACR irá copiar o valor dos bits que repetem entre os ID’s em comparação e armazená-los, os demais ele preenche com “*don’t cares*”. Como exemplo, analisa-se a posição ID5 da Tabela 1, em



que o valor lógico dos bits que repetem é 1, portanto é este valor que campo ACR armazena, e o campo AMR é preenchido com o valor 0, indicando que naquela posição os bits em comparação são os mesmos, sendo assim um bit relevante.

Portanto, é por meio do campo ACR do *Acceptance Filter* que é definido se um único ID, ou um intervalo de ID's é aceito. Tratando-se do exemplo em questão, para um ID ser aceito, as posições ID10, ID9, ID8, ID5, ID3 e ID2 tem que apresentar os valores estabelecidos pelo ACR, pois elas são as posições relevantes. Como as posições ID7, ID6, ID4, ID1 e ID0 são tratadas como “*don't cares*”, isso implica que elas podem assumir qualquer valor binário. Assim, são possíveis 32 opções de ID's diferentes. Neste caso, portanto, não somente os ID's 0x67 e 0xB4 serão recebidos e aceitos no nó X, mas também outros 30 identificadores.

Desta maneira, o nó X receberá e aceitará 32 mensagens com base nesta configuração de filtro. De forma geral, o número de mensagens recebidas e aceitas com base nas configurações de filtro é dada por  $2^{n[i]}$ . Onde,  $n[i]$  se refere a quantidade de bits irrelevantes, ou “*don't cares*” (VECTOR E-LEARNING, [201-]).

Em síntese, o *Acceptance Filter* pode deixar passar um único ID, ou um único intervalo de ID's, como ilustrado no exemplo da Tabela 1.

### 5.1.2 Transmissão e recepção de mensagens na rede CAN

Quando uma ECU deseja enviar uma mensagem na rede CAN, ela envia a mensagem para o controlador CAN desta central eletrônica (Tx *Buffer* - preparar). Assim que o controlador conseguir acesso ao barramento através do processo de arbitração (Enviar Mensagem) todas as outras ECU's passam a ser receptoras desta mensagem (Receber Mensagens). Logo após, todos os receptores fazem o processo de filtragem (Seleção) via *Acceptance Filtering*, onde os dados serão aceitos ou rejeitados naquele nó. Assim que aceitos, os dados são armazenados em um Rx- *Buffer*, para serem processados posteriormente (METRÔLHO, 1999). O processo de transmissão e recepção de mensagens na rede CAN é ilustrado na Figura 5.

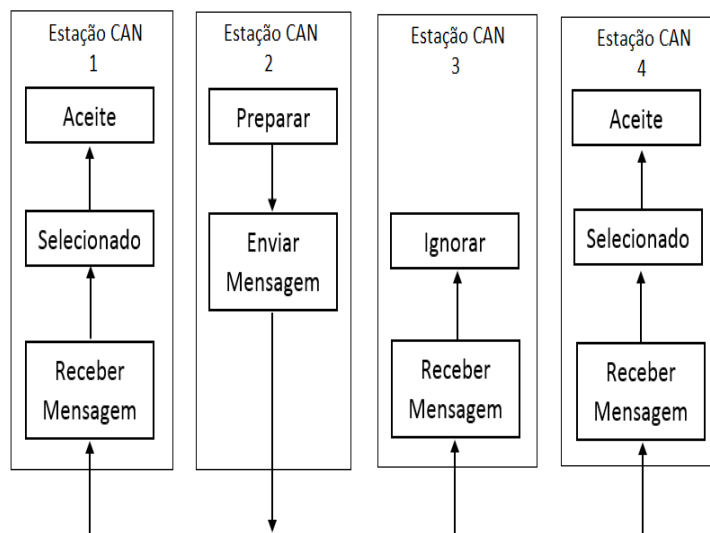


Figura 5 - Transmissão e Recepção de mensagens  
 Fonte: Metrólho (1999).

### 5.1.3 Processo de arbitragem

Em uma rede CAN todos os nós têm direito de acessar o barramento quando for requerido. No entanto, para preservar a comunicação do sistema em tempo real, o padrão ISO 11898-1 (ISO, 2003a), garante o transporte de dados de forma não destrutiva *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). Isso assegura que todos nós querendo transmitir uma mensagem esperem até que barramento esteja livre. Em caso de vários nós tentando acessar o barramento simultaneamente, o método CSMA/CA baseado no processo de arbitragem bit a bit não destrutiva garante que a mensagem com maior prioridade seja transmitida. No protocolo CAN a prioridade da mensagem está diretamente relacionada com o ID da mesma, de forma que, quanto menor o ID maior será a prioridade da mensagem ao acessar o barramento (VECTOR E-LEARNING, [201-]).

Enfim, quando ocorre o acesso simultâneo ao barramento o processo de arbitragem ocorre bit a bit até no final do campo ID. Assim, o nó que envia um bit dominante “0” no último bit do identificador, enquanto os demais enviam um bit recessivo “1”, passa a controlar o barramento. Portanto, este tem o direito de enviar o restante de sua mensagem. Na arbitragem, um bit dominante sempre sobrescreve um bit recessivo. Os nós que perderem a arbitragem irão tentar enviar suas respectivas mensagens quando o barramento estiver disponível novamente (MOURA, 2014). De forma simplificada, o processo de arbitragem pode ser exemplificado pelo fluxograma da Figura 6.

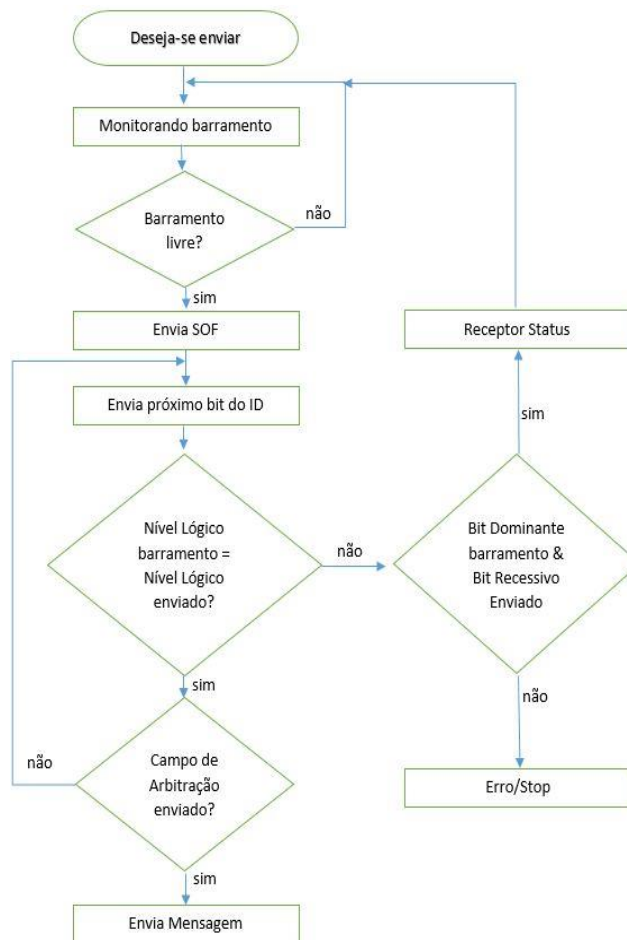


Figura 6 – Fluxograma do acesso ao barramento

## 5.2 Gerenciamento de erros

O CAN é caracterizado por ser uma rede robusta com a capacidade de se adaptar a condições de falhas. A detecção de erros em uma rede CAN é um mecanismo imprescindível para o funcionamento da rede, pois além de detectar mensagens com erro, ele permite retransmiti-las assim que o problema for solucionado. Quando um erro é encontrado, o nó emite uma sinalização “*Error Flag*” que interrompe o tráfico de mensagens no barramento, possibilitando que o nó tome medidas adequadas à aquela situação, por exemplo, descartando a mensagem corrompida (KVASER, 2016).

Existem dois tipos de contadores de erros no CAN: *Transmit Error Counter (TEC)* e *Receive Error Counter (REC)*. A princípio, o nó transmissor quando detecta uma falha incrementa o TEC mais rapidamente do que os nós receptores incrementam o REC. Isso ocorre pois existe uma maior probabilidade de que as falhas aconteçam na transmissão. Como resultado, utilizando os contadores de erros o CAN não só detecta a falha, mas também executa o mecanismo de confinamento de erros (KVASER, 2016).

### 5.2.1 Mecanismos de detecção de erros

Para garantir a robustez e confiabilidade na transmissão de dados o protocolo conta com cinco métodos diferentes de detecção de erros. Sendo, dois deles relativos ao bit, e os outros relacionados a mensagem (KVASER, 2016).

Os relacionados ao bit:

- **Bit Monitoring (monitoramento de bits):** cada transmissor lê o barramento após a transmissão e identifica o sinal transmitido, se for transmitido um bit dominante e o bit lido no barramento for recessivo, um erro de bit é sinalizado (*Bit error*) (GUIMARÃES; SARAIVA, 2002).
- **Bit Stuffing (Inserção de Bit):** quando cinco bits consecutivos com o mesmo nível forem transmitidos por um nó, um sexto bit é adicionado com valor oposto. Ou seja, apenas cinco bits consecutivos podem ter o mesmo nível (dominante ou recessivo). Desta maneira, se mais de cinco bits do mesmo nível forem detectados no barramento, um erro é sinalizado (*Stuff error*) (KVASER, 2016; GUIMARÃES; SARAIVA, 2002).

Os que se levam em conta a mensagem:

- **Frame Check (verificação do quadro):** os receptores analisam alguns bits da mensagem recebida que tem formato fixo, ou seja, seus valores são definidos pela norma. Assim, se o controlador CAN detecta um valor inválido em um desses campos, um erro é sinalizado (*Form error*) (KVASER, 2016; GUIMARÃES; SARAIVA 2002).
- **Acknowledgment Check (ACK):** os nós receptores escrevem um bit dominante no campo ACK para cada mensagem correta recebida. Quando o bit do campo ACK é recessivo, significa, que a mensagem estava corrompida, ou a mesma não foi recebida por nenhum nó. Portanto, um erro de reconhecimento é sinalizado (*Acknowledgement error*) (KVASER, 2016; GUIMARÃES; SARAIVA, 2002).
- **Cyclic Redundance Check (CRC) (verificação cíclica de redundância):** As mensagens têm 15 bits de redundância cíclica (CRC), representado pelo *checksum*. Assim, o transmissor calcula o valor de CRC e o envia junto com a mensagem. Chegando no receptor, o CRC é calculado novamente, se este valor for diferente, um erro é sinalizado (*CRC error*) (KVASER, 2016; GUIMARÃES; SARAIVA, 2002).

Sempre que um nó detecta uma condição de erro, uma *Error Flag* é sinalizada. Isso acontece para cada tipo de erro supracitado.

### 5.2.2 Mecanismos para o confinamento de falhas

O confinamento de falhas tem como principal função preservar a robustez e a confiabilidade do sistema de transmissão. Ele distingue entre erros temporários e falhas permanentes, além de localizar, e em alguns casos desativar, o nó defeituoso da rede (ISO, 2003a).

O mecanismo para o confinamento de falhas é dividido em três estados:

- **Error Active:** o nó comunica normalmente com o barramento e envia uma *Active Error Flag* quando um erro é detectado (DI NATALE, 2012; BOSCH, 1991).
- **Error Passive:** Neste estado, o nó apresenta um comportamento defeituoso, mas ele ainda pode transmitir no barramento. No entanto, sua transmissão é restrita a *Passive Error Flags* (DI NATALE, 2012; BOSCH, 1991).
- **Bus Off:** Neste estado considera-se que o nó está provavelmente corrompido, desta forma, o mesmo não pode ter qualquer influência no barramento. Ou seja, um nó neste estado não pode transmitir nada no barramento (DI NATALE, 2012; BOSCH, 1991). Quando uma *Error Flag* é detectada em um nó, este vai para o estado *Error Active*.

Caso os contadores de erros TEC e REC sejam incrementados a um valor maior do que 127, o nó então passará para o estado de *Error Passive*. Neste estado, se o TEC for incrementado a um valor acima de 255 o nó entrará no estado de *Bus Off* (KVASER, 2016). A transição do Estado *Bus Off* para o estado *Error Active* ocorre quando os contadores de erro são zerados após monitorar 128 ocorrências de 11 bits recessivos consecutivos (ISO, 2003a). Vale ressaltar que os contadores de erros também são decrementados, à medida que as mensagens são corretamente recebidas (GUIMARÃES; SARAIVA, 2002). As regras para incremento e decremento dos contadores de erros são de certa forma complexas, contudo, sua essência é simples: para cada mensagem erroneamente transmitida/recebida o contador de erro REC é incrementado por 1 unidade nos nós receptores, enquanto o contador de erro TEC é incrementado por 8 unidades. E estes contadores são decrementados à medida que as mensagens são recebidas corretamente (KVASER, 2016). As regras para incrementar ou

decrementar os contadores de erros estão descritas detalhadamente em (BOSCH, 1991). De forma simplificada a Figura 7 ilustra o mecanismo de confinamento de falhas.

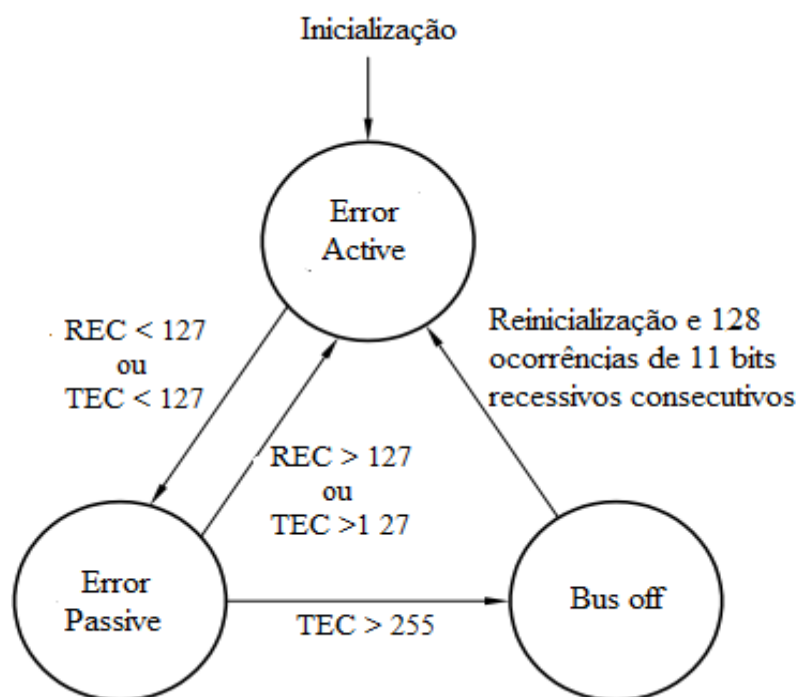


Figura 7 – Confinamento de falhas  
Fonte: ISO (2003°).

### 5.3. Desempenho de uma rede CAN automotiva

Um dos parâmetros mais importantes no que diz respeito a rede CAN automotiva é o *bus load*, ou a carga do barramento - definida como a medida da largura de banda utilizada -, que indica o quanto da capacidade da rede está sendo utilizada em um dado intervalo de tempo. Esta medida é dada em porcentagem. Outro parâmetro de suma importância é a latência, ou o tempo de resposta de uma mensagem, que é o intervalo de tempo medido desde que a mensagem é gerada até ela ser completamente transmitida no barramento (MOURA, 2014).

Como já visto, a comunicação em uma rede CAN é feita por um barramento *broadcast* utilizando uma topologia em linha. Em outras palavras, apenas uma única mensagem é transmitida e disponibilizada para as centrais eletrônicas por vez. Essas mensagens podem ser classificadas em: periódicas, esporádicas ou por evento.

Conseqüentemente, as mensagens que mais influenciam no desempenho do sistema e na carga do barramento são as mensagens periódicas. De forma que mensagens com períodos de amostragem longos geram um alto nível de ociosidade no barramento. Não obstante, mensagens com baixos períodos de amostragem diminuem a ociosidade do barramento.

Considerando-se, por exemplo, um mesmo intervalo de tempo, as mensagens que possuem um período de 10 milissegundos são enviadas no barramento cerca de dez vezes a mais do que uma mensagem com período de 100 milissegundos. Portanto, mensagens que possuem pequenos períodos de amostragem aumentam a possibilidade de ocorrer mais disputas pelo acesso ao barramento, o que por sua vez aumenta o tempo de respostas das mensagens de menor prioridade devido ao processo de arbitragem (GODOY, 2007).

Por isso, em uma rede CAN automotiva a carga do barramento é monitorada constantemente. Justamente para evitar cenários de sobrecarga de mensagens, fazendo com que novas mensagens sejam enviadas para o barramento sobrepondo-se as anteriores que ainda não tinham sido transmitidas. Causando assim uma transmissão constante de erros fazendo a rede entrar em colapso. Por isso, no CAN é importante garantir que a mensagem de menor prioridade seja transmitida antes que uma mensagem subsequente seja enfileirada para a transmissão. Caso o contrário, a mensagem de menor prioridade perderá o processo de arbitragem e nunca será transmitida.

A carga no barramento também pode ser influenciada por mensagens enviadas erroneamente, como observado na sessão 5.2. Pois o mecanismo de recuperação de falhas envolve retransmitir as mensagens que foram corrompidas, o que pode ocasionar uma sobrecarga na rede, além de aumentar o tempo de resposta das mensagens a serem enviadas (BROSTER; BURNS; RODRÍGUEZ-NAVAS, 2002).

Portanto, o desempenho, assim como a carga do barramento da rede CAN automotiva, são susceptíveis a quantidade de mensagens que trafegam no barramento, a periodicidade destas mensagens, além de erros no envio e no recebimento de mensagens.

## **6 CONCLUSÃO**

Em conclusão, o foco deste artigo foi transmitir ao leitor uma visão geral sobre a rede CAN, especificadamente, na parte de transmissão e recepção de mensagens, assim como a parte de detecção e confinamento de erros. Para entender o processo de transmissão de dados entre as ECU's é preciso entender o campo identificador (ID) de cada mensagem. É este que define a prioridade da mensagem, assim como estabelece as condições de filtro para uma central eletrônica. A respeito do campo identificador, é importante ressaltar que a prioridade da mensagem no barramento é inversamente proporcional ao valor em hexadecimal neste campo. Ou seja, quando menor for o valor do campo ID maior será a prioridade da mensagem e vice-versa.

Para garantir a confiabilidade, precisão e rapidez no tempo de resposta do sistema, a rede CAN conta com cinco métodos de detecção de erros, sendo estes alguns referentes a bit, e outros relativos à mensagem. Além disso, o CAN apresenta um avançado mecanismo para o confinamento de falhas, que permite o isolamento dos nós defeituosos, bem como o total desligamento do barramento em uma situação crítica.

Apesar da rede CAN ser largamente utilizada na indústria automotiva devido a sua robustez, as suas aplicações podem ser limitadas a carga no barramento (*bus load*). Isto acaba sendo um grande problema para as montadoras, pois com a evolução natural do setor automobilístico aumenta também a quantidade de centrais eletrônicas nos veículos, à medida que a quantidade de sensores e atuadores crescem tanto para proporcionar maior segurança, bem como para proporcionar maior conforto ao seu usuário. Portanto, o crescimento de centrais eletrônicas no automóvel contribui para o aumento da carga no barramento. Não obstante, há outros fatores que influenciam na carga do barramento CAN, como o período das mensagens cíclicas enviadas e o mecanismo para a recuperação de erros.

Por fim, o CAN é uma rede flexível, que fornece comunicação em tempo real, apresenta tolerância a interferências eletromagnéticas, pode trabalhar em altas velocidades, apresenta um baixo custo de implementação e é extremamente confiável. Além disso, a adição de novas centrais é simples. No entanto, o aumento da carga no barramento pode comprometer o funcionamento do sistema. Sendo assim, para trabalhos futuros torna-se relevante um estudo temporal das mensagens com intuito de procurar uma melhor distribuição probabilística das mesmas, a fim de poder aumentar a carga do barramento sem deixar que o sistema perca sua robustez e confiabilidade.

## MESSAGES TRANSMISSION AND ERROR MANAGEMENT IN AN AUTOMOTIVE CAN NETWORK

### ABSTRACT

The aim of this paper is to illustrate how the message transmission process in a CAN network is done. Besides characterizing the CAN Error Handling System, it will be presented the types of errors and the error detection mechanisms as well as the error confinement mechanism. Additionally, this document provides an understanding of the busload on a CAN network, and how this is influenced by the number of messages, by the number of electronic units, and by the existing errors on the present network. This article also includes a brief description of the CAN bus and the message format in a CAN network for a better understanding of it.

**Keywords:** CAN bus. Controller area network (CAN). Bus load. Identifier (ID). Arbitration logic. CAN network.



## REFERÊNCIAS

ASSIS, A. C. de. **Implementação e avaliação do protocolo FTT-CAN sobre o sistema AUTOSAR**. 2011. 105 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

BOSCH, R. et al. **CAN specification version 2.0**. Rober Bousch GmbH, Postfach, v. 300240, p. 72, Stuttgart, 1991.

BROSTER, I; BURNS, A.; RODRÍGUEZ-NAVAS, G. Probabilistic analysis of CAN with faults. In: REAL-TIME SYSTEMS SYMPOSIUM, 23., 2002, Austin, Texas. **Proceedings... IEEE**, 2002. p. 269-278.

COOK, J. A.; FREUDENBERG, J. S. Controller Area Network (CAN). **EECS**, v. 461, p. 1-5, 2007.

DAVIS, R. I. et al. Controller Area Network (CAN) schedulability analysis: refuted, revisited and revised. **Real-Time Systems**, v. 35, n. 3, p. 239-272, 2007.

DI NATALE, M. et al. **Understanding and using the controller area network communication protocol: theory and practice**. London: Springer Science & Business Media, 2012.

EASYSYNC application Note AN\_103 CAN Plus Message. 2010. Disponível em: <[http://www.easysync-ltd.com/support/documentation/AN\\_103\\_CAN\\_Plus\\_Message\\_Filtering.pdf](http://www.easysync-ltd.com/support/documentation/AN_103_CAN_Plus_Message_Filtering.pdf)>. Acesso em: 16 jan. 2016.

GODOY, E. P. **Desenvolvimento de uma ferramenta de análise de desempenho de redes CAN para aplicações em sistemas agrícolas**. 2007. 93 f. Dissertação (Mestrado em Engenharia) – Escola de Engenharia, Universidade de São Paulo, São Carlos, 2007.

GUIMARÃES, A. A.; SARAIVA, A. M. O protocolo CAN: entendendo e implementando uma rede de comunicação serial de dados baseada no barramento. In: CONGRESSO SAE BRASIL, 11., 2002. São Paulo. **Anais...Controller Area Network**, 2002.

INTERNATIONAL STANDARD ORGANIZATION. **ISO 11898-1**: Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signaling. Geneva, 2003a.

\_\_\_\_\_. **ISO 11898-2**: Road vehicles–Controller area network (CAN) – Part 2: High-speed medium access unit. Geneva, 2003b.

\_\_\_\_\_. **ISO 11898-3**: Road Vehicles – Controller Area Network (CAN) – Part 3: Low – speed, fault-tolerant, medium-depedent interface. Geneva, 2006.

JUNGER, M. **Introduction to J1939**: version 1.1, 2010. Disponível em: <[https:// vector.com/portal/medien/cmc/application\\_notes/AN-ION-1-3100\\_Introduction\\_to\\_J1939.pdf](https://vector.com/portal/medien/cmc/application_notes/AN-ION-1-3100_Introduction_to_J1939.pdf)>. Acesso em: 16 de maio 2017.

KVASER. **The CAN Protocol Tour**, 2016. Disponível em: <<https://www.kvaser.com/can-protocol-tutorial/>>. Acesso em: 16 jan. 2016.

METRÔLHO, J. C. M. M. **Rede CAN para comando de actuadores em estufas agrícolas**. Dissertação (Mestrado) – Universidade do Minho, Portugal, 1999.

MOURA, N. C. F. **Escalonamento de mensagens com offset em redes CAN**. Trabalho de Conclusão de Curso – Universidade Federal de São João Del Rei, Ouro Branco, 2014.

SEMICONDUCTORS, Philips. **SJA1000 Stand-alone CAN controller**, 2000. Disponível em: <<https://www.nxp.com/docs/en/application-note/AN97076.pdf>>. Acesso em: 16 de maio 2017.

TINDELL, K.; BURNS, A.; WELLINGS, A. J. Calculating controller area network (CAN) message response times. **Control Engineering Practice**, v. 3, n. 8, p. 1163-1169, 1995.

VECTOR E-LEARNING. CAN Protocol Reference Chart. [201-]. Disponível em: <[http://elearning.vector.com/vl\\_can\\_introduction\\_en.html](http://elearning.vector.com/vl_can_introduction_en.html)>. Acesso em: 04 mar. 2017.

## **DADOS DOS AUTORES**

### **Saulo Marcos Torres de Carvalho**

E-mail: [saulo.eng.eletrica@gmail.com](mailto:saulo.eng.eletrica@gmail.com)

Currículo lattes: <http://lattes.cnpq.br/4907976478067963>

Graduado em Engenharia Elétrica pelo Instituto Federal Minas Gerais (IFMG) e, atualmente, é Chefe de Projeto na Fiat Chrysler Automobiles, Betim-MG.

### **Gustavo Lobato Campos**

E-mail: [gustavo.lobato@ifmg.edu.br](mailto:gustavo.lobato@ifmg.edu.br)

Currículo Lattes: <http://lattes.cnpq.br/9613750934178733>

Doutor em Ciências Técnicas Nucleares pela Universidade Federal de Minas Gerais (UFMG). Mestre em Engenharia Elétrica pela Pontifícia Universidade Católica de Minas Gerais (PUC - Minas). Graduado em Engenharia Eletrônica e de Telecomunicação pela PUC – Minas. Atualmente é professor efetivo do Instituto Federal Minas Gerais – Campus Formiga.