

**INSTITUTO FEDERAL DE MINAS GERAIS
BACHARELADO EM ENGENHARIA ELÉTRICA
SAULO MARCOS TORRES DE CARVALHO**

**AVALIAÇÃO DO MÉTODO *WORST CASE RESPONSE TIME* PARA O CÁLCULO
DO TEMPO DE RESPOSTA EM MENSAGENS CAN**

**FORMIGA – MG
2017**

SAULO MARCOS TORRES DE CARVALHO

**AVALIAÇÃO DO MÉTODO *WORST CASE RESPONSE TIME* PARA O CÁLCULO
DO TEMPO DE RESPOSTA EM MENSAGENS CAN**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais, como requisito para obtenção do título de bacharel em Engenharia Elétrica.

Orientador: Prof. Msc. Gustavo Lobato Campos

FORMIGA – MG

2017

Carvalho, Saulo Marcos Torres de.
621.3 Avaliação do método do Worst Case Response Time para o cálculo do tempo de resposta em mensagens CAN / Saulo Marcos Torres de Carvalho. – Formiga : IFMG, 2017.
86p.: il.

Orientador: Prof. Msc. Gustavo Lobato
Trabalho de Conclusão de Curso – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. CAN. 2. Carga no Barramento. 3. Pessimismo. 4. Tempo de Resposta no Pior. I. Título.

SAULO MARCOS TORRES DE CARVALHO

**AVALIAÇÃO DO MÉTODO *WORST CASE RESPONSE TIME* PARA O CÁLCULO
DO TEMPO DE RESPOSTA EM MENSAGENS CAN**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais como requisito para obtenção do título de bacharel em Engenharia Elétrica.

Avaliado em: ____ de _____ de _____.

Nota: _____

BANCA EXAMINADORA

Prof. Msc. Gustavo Lobato Campos - Orientador

Prof. Msc. Diego Mello da Silva

Prof. Dr. Washington Santos da Silva

Dedico este trabalho a minha mãe, pelo amor, carinho e incentivo. Por fazer o possível e o impossível para ver os meus sonhos se tornarem realidade.

RESUMO

O protocolo CAN (do inglês, *Controller Area Network*) foi desenvolvido pela Bosch em meados da década de 80, com objetivo de prover maior eficiência na comunicação de redes automotivas com um baixo custo de implementação. Desde então, essa tecnologia foi difundida no setor automotivo, e atualmente todas as montadoras de veículos utilizam o protocolo CAN para realizar a comunicação entre unidades de controle eletrônico (em inglês, *Electronic Control Unit*), cuja sigla é ECU. Com a crescente busca por conforto e segurança nos veículos houve uma expansão na quantidade de ECUs e, conseqüentemente, da quantidade de mensagens que trafegam no barramento CAN. Desta forma, a carga no barramento, também conhecida como *bus load* aumenta. Este aumento é um fator limitante que a maioria das montadoras de automóveis utilizam na escolha da arquitetura veicular. Geralmente os fabricantes de automóveis optam por manter um *bus load* em níveis inferiores à 60% para garantir a confiabilidade e robustez do barramento a fim de evitar que a latência de uma mensagem não seja superior ao seu *deadline*. O *deadline* de uma mensagem é o tempo máximo que a central receptora deve aguardar para receber uma mensagem. Quando este tempo é excedido, a confiabilidade da rede não pode ser garantida. Assim, um método para avaliar a robustez da rede CAN é o *Worst Case Response Time* (WCRT) usado para calcular a latência da mensagem no pior caso, de forma que este valor deve ser menor que o *deadline* da mensagem. Alguns estudos mostram que essa análise, em algumas situações, pode ser otimista. No entanto, o objetivo deste Trabalho de Conclusão de Curso (TCC) é evidenciar que o método do WCRT apresenta uma análise pessimista e que o tempo de resposta fornecido por esta análise é bem superior aos valores práticos. Para atingir tal objetivo optou-se por utilizar o *software* SymTA Vision® para simular um banco de dados de tempos de respostas de um cenário disponível na literatura conhecido como SAE *Benchmark*, e seguido disso apresentar a análise desses dados.

Palavras chave: CAN (*Controller Area Network*), Carga no Barramento, Pessimismo, Tempo de Resposta no Pior Caso.

ABSTRACT

The Controller Area Network (CAN) was designed by Bosch in the mid 80's, whose focus was to provide greater efficiency on the communication of automotive network at a low cost implementation. Since then, this technology has been extensively used by the automotive manufactures to exchange data among Electronic Control Units (ECUs). With the growing demand for comfort and safety in vehicles, the number of ECUs have increased, as well the number of CAN messages broadcasting over the bus. Thus, the *bus load* is a constraint on the development of a vehicle architecture. In general, the automakers usually maintain the bus load at levels below 60% to ensure the bus reliability and robustness to prevent that a given message miss its deadline. When a message miss its deadline the reliability of the network cannot be guaranteed. Hence, to assess the robustness of a CAN network the *Worst Case Response Time* (WCRT) analysis is used to calculate the message response time, this value shall be less than the message deadline. Some studies have shown that this analysis in some cases may be optimistic. However, the aim of this work is to emphasize that the WCRT presents a pessimistic approach because, in practice, the response time provided by this analysis is higher. To accomplish this goal the SymTA Vision® software was chosen to create a database for messages available in a dataset known as SAE Benchmark. The result of this analysis is provided by work..

Keywords: Controller Area Network, Bus load, Pessimism, Worst Case Response Time.

LISTA DE FIGURAS

Figura 1 - Vendas de microcontroladores CAN entre 1999 e 2003.....	14
Figura 2 - Barramento CAN.....	22
Figura 3 - Protocolo CAN dentro do modelo OSI.	23
Figura 4 - Níveis de tensão em um barramento CAN de alta velocidade.	24
Figura 5 - Níveis de tensões diferenciais para o barramento CAN de baixa velocidade.	25
Figura 6 - <i>Frame</i> de Dados	26
Figura 7 - Formato de um <i>Frame</i> de Erro.....	28
Figura 8 - <i>Frame</i> de Sobrecarga.	29
Figura 9 - Processo de <i>bit stuffing</i>	30
Figura 10 - Processo de Arbitragem na rede CAN.	31
Figura 11 - Fluxograma de acesso ao barramento CAN.	32
Figura 12 - Tempos de transmissão de uma mensagem CAN.....	34
Figura 13 - Diagrama temporal da transmissão de uma mensagem CAN.	34
Figura 14 - Modelo temporal para mensagens CAN.	39
Figura 15 - Esquema de modelagem SymTA Vision®.	43
Figura 16 - Mensagens enfileiradas	44
Figura 17 - Configuração CAN frame SymTA®.....	47
Figura 18 - Perfil do tempo de resposta para o WCRT.	55
Figura 19 - <i>Worst Case Gantt</i> mensagem F14.....	56
Figura 20 - <i>Worst Case Gantt</i> mensagem F10.....	57
Figura 21 - <i>Worst Case Gantt</i> mensagem F6.....	57
Figura 22 - Histograma mensagem F8.....	61
Figura 23 - Histograma mensagem F9.....	61
Figura 24 - Histograma mensagem F12.....	62
Figura 25 - Função de Distribuição Acumulada Empírica da mensagem F12.....	65
Figura 26 - <i>Boxplot</i> mensagem F12.	66
Figura 27 - Gráfico de Cullen & Frey para a mensagem F12.....	68
Figura 28 - Gráfico Quartil-Quartil do ajuste da distribuição Weibull.	69
Figura 29 - Gráfico Quartil-Quartil do ajuste da distribuição Gamma.	69
Figura 30 - <i>Boxplot</i> da arquitetura avaliada.....	70
Figura 31 - Perfil WCRT e Rmax das amostras simuladas via SymTA®.	72

Figura 32 - <i>Bus load</i> e contribuição de cada ECU na rede.	73
Figura 33 - Contribuição de cada mensagem para o valor de <i>bus load</i>	74
Figura 34 - Perfil de bus load para uma amostra.	75
Figura 35 - <i>Boxplot Bus Load</i>	76

LISTA DE TABELAS

Tabela 1 - Valores típicos de velocidades de transmissão de acordo com o comprimento do barramento.	21
Tabela 2 – Valores mínimos e máximos para bits recessivos e dominantes no barramento CAN de alta velocidade.	24
Tabela 3 - Valores mínimos e máximos para bits recessivos e dominantes no barramento CAN de baixa velocidade.	25
Tabela 4 – Quantidade de bits sujeitos ao <i>bit stuffing</i>	41
Tabela 5 - Exemplo: mensagens CAN.	43
Tabela 6 - Conjunto de mensagens CAN SAE.	46
Tabela 7 - Configuração dos parâmetros de entrada do SymTA®.	48
Tabela 8 - Fator de <i>stuffing bits</i>	51
Tabela 9 - Comparação WCRT usando as Equações 16 e 18.	52
Tabela 10 - <i>Best Case Response Time SAE Benchmark</i>	53
Tabela 11 - Resultados do SymTA® para WCRT e BCRT.	54
Tabela 12 – Comparação entre os valores do WCRT da literatura e simulação no SymTA®.	59
Tabela 13 – Pessimismo WCRT.	63
Tabela 14 - Grau de pessimismo para as mensagens da arquitetura.	71
Tabela 15 - Relação entre período da mensagem e <i>bus load</i>	77

LISTA DE ABREVIATURAS E SIGLAS

ACK - *Acknowledge*

APAC – *Asia-Pacific*

BCRT – *Best Case Response Time*

CAN – *Controller Area Network*

CRC – *Cyclic Redudancy Check*

DLC – *Data Length Code*

ECU – *Electronic Control Unit*

EMEA – *Europe, Middle East, and Africa*

EOF – *End of Frame*

ID – *Identificador (Identifier)*

IDE – *Identifier Extension*

ISO – *International Organization for Standardization*

LATAM – *Latin America*

RTR – *Requisição de Transmissão Remota (Remote Transmission Request)*

SAE - *Society of Automotive Engineers*

SOF – *Start of Frame*

WCRT – *Worst Case Respose Time*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Justificativa	16
1.2 Objetivos geral e específicos	18
1.3 Estrutura do Trabalho	19
2 REFERENCIAL TEORICO	20
2.1 O Protocolo CAN	20
2.1.1 Modelo OSI e ISO 11898	23
2.1.2 Estrutura da Mensagem – <i>CAN Frame</i>	26
2.1.2 <i>Bit Stuffing</i>	30
2.1.3 Arbitração.....	30
2.1.4 <i>Bus Load</i>	32
2.2 Tempo de Reposta de uma Mensagem CAN.	33
3 ANÁLISE DO TEMPO DE RESPOSTA EM MENSAGENS CAN	36
3.1 Premissas	36
3.2 Análise Clássica do WCRT	37
3.3 SymTA Vision® Análise Temporal.....	41
3.4 Análise Otimista do WCRT	43
3.5 Análise Pessimista do WCRT	44
3.6 SAE <i>Benchmark</i>	45
3.7 Modelagem com SymTA Vision®	46
4 RESULTADOS E DISCUSSÃO	49
4.1 Avaliação dos resultados obtidos por Burns et al., (2002).....	49
4.2 Resultados do SymTA® para o WCRT.....	54
4.3 Resultados SymTA® para diferentes condições de inicialização	59
4.4 Avaliação do <i>Bus Load</i>	73
4.5 Publicações Resultantes Desta Monografia	78
5 CONCLUSÃO.....	80
5.1 Trabalhos Futuros.....	82
REFERÊNCIAS BIBLIOGRÁFICAS	84

1 INTRODUÇÃO

Criado por Robert Bosch no início de anos 80, o protocolo CAN (*Controller Area Network*), trata-se de um meio de comunicação serial, desenvolvido para prover uma maior eficiência e robustez na comunicação de redes automotivas com baixo custo de implementação. Entretanto, somente em 1986, no congresso da SAE (*Society of Automotive Engineers*), o protocolo CAN foi definitivamente apresentado ao mundo (DAVIS et al., 2007).

Os primeiros controladores CAN foram lançados ao mercado em 1987 pela Intel (82526) e Phillips (82C200) (DAVIS et al., 2007). Já no início dos anos 90, a Bosch submeteu a especificação para o protocolo CAN para a normatização (BOSCH, 1991).

Em 1993, foi publicada a primeira normativa para o protocolo CAN (CAN 2.0A). Logo após, em 1995, foi publicada uma extensão da norma CAN 2.0A, denominada CAN 2.0B, a qual permitia dispositivos com identificadores mais longos, ambas baseadas na especificação da fabricante Bosch (COOK et al., 2007). Contudo, no início dos anos 90 houve um aumento considerável da complexidade na eletrônica dos automóveis, conseqüentemente o número de unidades de controle eletrônico, conhecidas por ECUs (*Electronic Control Units*), ou simplesmente centrais eletrônicas, cresceram de cinco ou menos para cerca de quarenta por veículo até o início dos anos 2000. Até então, a topologia para comunicação entre as centrais eletrônicas era feita ponto a ponto, ou seja, o crescimento do número de ECUs ocasionou um aumento na complexidade do cabeamento para interligar as centrais eletrônicas. Eram necessários milhares de conectores e cabos, a instalação era bem mais complexa e onerosa. A primeira montadora de carros que aderiu ao CAN foi a Mercedes, em 1991, para a produção de veículos categoria S (DAVIS et al., 2007).

Como resultado, para facilitar a interação e ligação de dispositivos de diferentes fabricantes, no sentido de reduzir a quantidade de cabos e custos, o protocolo CAN foi rapidamente adotado pela indústria automobilística (DAVIS et al., 2007).

A Figura 1 elucida o crescimento na venda de microcontroladores com o controlador CAN integrado entre os anos de 1999 a 2003.

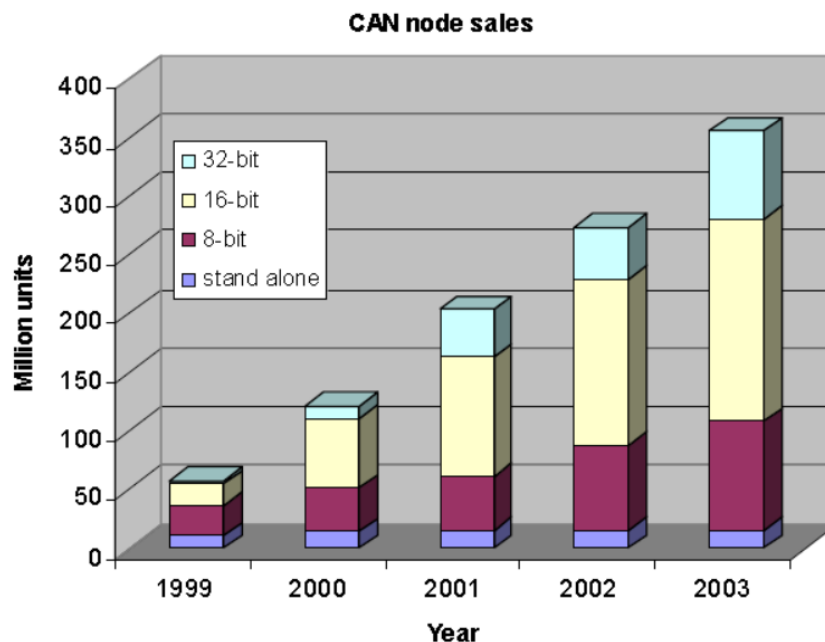


Figura 1 - Vendas de microcontroladores CAN entre 1999 e 2003.
Fonte: (DAVIS et al., 2007)

Deste então, praticamente todo carro fabricado nas regiões de EMEA, NAFTA, LATAM e APAC são equipados com pelo menos um barramento CAN. Atualmente, mais de 400 milhões de controladores CAN são fabricados a cada ano (DAVIS et al., 2007).

Destaca-se também que a rede suporta velocidades de transmissão que podem chegar até a uma taxa de 1 Mbit/s, o que possibilita que o protocolo também seja aplicado em outros domínios, por exemplo: indústrias médicas, controle de processos, sistemas aviônicos, entre outros (MUBEEN et al., 2011).

Como exposto anteriormente, com o desenvolvimento da tecnologia, a complexidade dos sistemas automotivos tem aumentado, levando ao crescimento do número de centrais eletrônicas no veículo que visam prover maior conforto e segurança ao motorista. Atualmente, sistemas inteligentes têm sido desenvolvidos para auxiliar o motorista em diversas situações de trânsito, dentre eles podemos citar: o *Lane Departure Warning* (LDW) ou *Lane Keep Assist* (LKA), cuja função é monitorar e auxiliar o veículo a se manter nas faixas de rolagem; no sistema de monitoramento de pontos cegos, ajuda o motorista a identificar objetos, animais e pessoas que estariam nas proximidades do veículo; O *Adaptive Cruise Control* (ACC), que monitora a distância do veículo da frente, acelerando e controlando os freios de forma automática para evitar a colisão, dentre outros sistemas (BISHOP, 2005).

Desta forma, observa-se que a tendência do setor automotivo está indo a encontro ao que chamamos de sistemas autônomos, o que por sua vez implica em um grande esforço da rede para suportar esta quantidade elevada de sensores e centrais eletrônicas. Assim, devido aos avanços da tecnologia automotiva, o protocolo CAN padrão tende a se saturar, devido às suas limitações. Esta saturação pode ser resultado da grande quantidade, prioridade ou periodicidade das mensagens que trafegam na rede. De modo que, em alguns casos, algumas mensagens podem não ser disponibilizadas no barramento, o que caracteriza uma situação crítica do sistema veicular, comprometendo a segurança e robustez da rede CAN. Para evitar essa condição, novos recursos vêm sendo estudados especialmente do ponto de vista temporal de uma mensagem CAN.

O protocolo CAN adota uma estratégia para a detecção de colisão de mensagens, de forma que a mensagem a ser transmitida no barramento seja escolhida de acordo com o seu identificador. Assim, quando mais de uma central necessita transmitir uma mensagem no barramento, a mensagem que tiver o menor identificador será transmitida. Isto é feito para elencar a prioridade das mensagens. Demais detalhes serão descritos no decorrer deste trabalho.

Devido a esta característica do protocolo CAN, no início dos anos 90, uma análise feita por Tindell et al. (1994a, 1994b) derivada do *Worst Case Latency Evaluation* (avaliação da latência do pior caso) apresentada por Joseph e Pandya (1986), para encontrar o pior caso do tempo de resposta de uma atividade em um sistema em tempo real, foi proposta para avaliar a latência de uma mensagem no protocolo CAN baseada no pior caso. O pior caso do tempo de resposta de uma mensagem WCRT (*Worst Case Response Time*) é definido como o maior tempo de resposta que uma mensagem pode atingir para aquela arquitetura veicular.

Desde então, esta análise temporal proposta por Tindell et al. (1994a, 1994b, 1995) vem sendo vastamente utilizada no setor automotivo para a configuração de novas arquiteturas de rede baseadas no protocolo CAN. Por exemplo, a Volvo tem usado esta teoria para avaliar a latência de comunicação em vários de seus modelos (CASPARSSON et al., 1998). Ao analisar o WCRT foi possível garantir que todas as mensagens e sinais que estas carregam iriam cumprir os seus prazos de transmissão (DAVIS et al., 2007).

Além disso, o método de Tindell et al., tem influenciado o desenvolvimento de controladores CAN pelos fornecedores de microcontroladores, i.e. Motorola; teses de

PhD e diversas pesquisas relacionadas ao escalonamento de mensagens CAN. Em geral, mais de 200 artigos mencionam o trabalho de Tindell et al. (DAVIS et al., 2007).

Antes do trabalho Tindell et al., os valores de utilização do barramento CAN, ou comumente conhecido como *bus load*, eram mantidos em torno de 30 ou 40% para garantir que as mensagens iriam cumprir os seus prazos de transmissão. Não obstante, com advento da metodologia do WCRT foi possível aumentar a capacidade do *bus load* para 80% e ainda garantir que as mensagens não atingiram seus tempos máximos para a transmissão no barramento (DAVIS et al., 2007).

Por fim, a proposta deste trabalho é avaliar método do WCRT usado para o cálculo do tempo de resposta em redes CAN. Para realizar tal objetivo, é realizado um estudo aprofundado do método no sentido de identificar as suas limitações. Além disso, o *software* SymTA Vision® (SYMTA, 2012) é utilizado para gerar um banco de dados, via simulação, de amostras de tempos de resposta para o cenário avaliado. E com base na análise dos dados gerados, estimar o pessimismo causado pela análise do WCRT. Portanto, prover a otimização do barramento, de forma que novas mensagens possam ser adicionadas à arquitetura, aumentando-se os níveis de *bus load*. Afim de proporcionar o aumento na vida útil do protocolo CAN, assim como, reduzir custos no desenvolvimento do projeto.

1.1 Justificativa

O problema do WCRT para calcular o tempo de resposta de uma mensagem, é que ele parte do pressuposto que todas as mensagens CAN disponíveis, são enfileiradas no mesmo instante de tempo de acordo com sua prioridade. Isto significa que as centrais eletrônicas iniciam a sua transmissão de mensagens de forma sincronizada, o que de fato não ocorre na prática. Por exemplo, na arquitetura veicular existem algumas centrais que são alimentadas pela bateria e outras somente no momento de ignição do veículo. A centrais que são alimentadas pela bateria podem transmitir mensagens com a chave na posição OFF (desligado). Quando a chave é comutada para posição ON (ligado), outras centrais irão iniciar a sua comunicação ao barramento CAN e necessariamente estas não estarão sincronizadas com a central que já iniciou sua transmissão de mensagens. Além disso, cada ECU tem um tempo

de inicialização interno, característico de seu *hardware* (DI NATALE, 2008). É importante salientar que como não há a sincronia das mensagens, o enfileiramento destas tentando ganhar o acesso ao barramento não é bem definido. Isto é, como há a variação de mensagens que são enfileiradas por intervalo de tempo, logo o tempo de resposta da mensagem também sofre alteração. Tome-se como exemplo, um cenário onde há três mensagens M1, M2, e M3 em ordem crescente de prioridade. De acordo com análise via WCRT, o tempo de resposta de M3 seria a soma das demais mensagens, uma vez que esta apresenta menor prioridade. Por outro lado, na prática podem ocorrer situações em que M3 seja apenas enfileirada com M1 o que resultaria em uma latência menor. Em um sistema real, situações semelhantes a esta podem ocorrer, assim em alguns casos o WCRT pode apresentar uma estimativa bem pessimista.

Não obstante, Davis et al. (2007) mostrou, em seu trabalho, que a análise de Tindell et al. (1994a, 1994b, 1995) apresenta algumas limitações principalmente em situações de *bus load* elevado. Isso pode resultar em casos que o cálculo do WCRT seja otimista, i.e., tempo de resposta calculado via WCRT menor que o tempo de resposta real da mensagem. E como o próprio nome sugere, este deveria ser o pior caso da latência da mensagem.

Em virtude destas limitações do WCRT supracitadas, simulações computacionais podem ajudar a tecer conclusões e melhorar o cálculo do tempo das mensagens. Com base nisto, simulações de alguns cenários de rede CAN baseados na literatura da área foram desenvolvidos e analisados, a fim de comparar os dados do WCRT. O *software* escolhido para realizar a análise foi o SymTA®, pois além de possibilitar a análise do pior caso ele também incorpora características de uma arquitetura veicular real, ou seja, no momento que a simulação é iniciada as mensagens não são sincronizadas a priori. Possibilitando assim, a simulação de um vasto banco de dados com as variações da latência das mensagens ao longo do tempo.

Por fim, as amostras simuladas pelo SymTA® podem ser utilizadas para obter estimativas intervalares do tempo de resposta das mensagens, e desta forma, avaliar o pessimismo do método do WCRT para estimar o tempo de resposta das mensagens CAN.

1.2 Objetivos geral e específicos

A presente monografia tem por objetivo geral avaliar a análise do WCRT proposta por Tindell et al. (1994a, 1994b, 1995) através de simulações via *software* SymTA® de um cenário conhecido, na literatura, como SAE *Benchmark* (Burns et al., 2002). Além disso, pretende-se propor uma melhoria no equacionamento para o WCRT para eliminar um pouco de seu pessimismo. E por fim, quantificar o pessimismo da teoria do WCRT com base nas amostras simuladas pelo *software* SymTA®.

Para consolidar o objetivo geral, alguns objetivos específicos devem ser alcançados sequencialmente, sendo apresentados a seguir:

- Definir o funcionamento de redes CAN de modo geral, de forma a explicitar como ocorre o tráfego de mensagens no barramento CAN.
- Simulação do WCRT de cenários disponíveis na literatura.
- Comparação das simulações com estudos previamente realizados para a validação dos resultados.
- Avaliar a influência da característica das mensagens em sua latência e *bus load*.
- Confrontar resultados do WCRT com a literatura, a fim de definir se este método é otimista ou pessimista.
- Utilizar resultados das amostras do SymTA® para estimar o pessimismo do WCRT.
- Utilizar resultados das amostras do SymTA® para estimar a variação do tempo de resposta das mensagens, conseqüentemente, determinar um intervalo seguro para o tempo de resposta das mensagens diminuindo o pessimismo causado pelo WCRT.

1.3 Estrutura do Trabalho

Esta monografia é constituída por cinco capítulos, na qual o primeiro, corresponde à introdução do tema. Além disso, nesta seção apresenta-se a justificativa do tema, objetivos a serem atingidos e a estrutura do trabalho.

O segundo capítulo consiste no referencial teórico de alguns tópicos necessários para o entendimento do trabalho, como conceitos básicos referentes ao protocolo CAN, normativas, estrutura de mensagens, carga do barramento e tempo de resposta.

O terceiro capítulo descreve a metodologia do trabalho. Nele é abordado toda a teoria clássica do WCRT. Além disso, este capítulo traz a modelagem que será realizada via SymTA® *software* para posterior análise dos resultados no capítulo seguinte.

O quarto capítulo, faz uma reflexão dos resultados obtidos na literatura pelo WCRT, além de apontar limitações na presente teoria. Adicionalmente, é feita uma avaliação do WCRT com base em amostras obtidas via simulação, para comprovar o pessimismo que o WCRT implica no cálculo do tempo de resposta de mensagens CAN. Também são apresentados os trabalhos resultantes dessa pesquisa.

Por fim, no capítulo cinco são descritas as conclusões finais, bem como propostas de continuidade do tema apresentado.

2 REFERENCIAL TEORICO

Para o melhor entendimento do protocolo CAN é necessário a definição de alguns conceitos básicos que serão de suma importância para as análises realizadas no decorrer deste trabalho. Em vista disso, este capítulo apresenta as principais características desse protocolo para entendimento do comportamento temporal de uma mensagem CAN.

2.1 O Protocolo CAN

Como mencionado previamente, o protocolo de rede CAN foi desenvolvido na década de 80 por Robert Bosch para se adequar à nova necessidade automotiva da época: uma rede de fácil implementação, confiável, robusta e barata.

De modo geral, a rede CAN possui as seguintes características (BARBOSA, 2003):

- Prioridade de mensagens;
- Tempos de latência garantidos;
- Flexibilidade da configuração;
- Consistência de dados;
- Capacidade *multicast*;
- Acesso multi-mestre ao barramento;
- Detecção e sinalização de erros automaticamente;
- Retransmissão automática de mensagens;
- Distinção entre erros temporários e falhas permanentes;
- Elevada taxa de transferência de dados.

Além disso, o CAN baseia-se em um sistema de transmissão de dados *broadcast*, ou seja, a informação é enviada, invariavelmente, para todas ECUs presentes no barramento. Portanto, não é possível enviar uma mensagem para uma central em específico. Todavia, a central receptora opta pela aceitação ou não da

mensagem disponível no barramento CAN através de um filtro local, conseqüentemente, esta central eletrônica processa somente mensagens que lhe interessam (MOURA, 2014).

A velocidade de transmissão de dados no CAN pode chegar até 1 Mbit/s para pequenos tamanhos de barramento (≤ 40 metros) (COOK, 2007). A velocidade de transmissão do barramento é proporcional ao comprimento deste, isso ocorre porque o processo de arbitragem, que será descrito posteriormente, requer que o sinal propagado na central mais remota seja refletido de volta antes que um bit seja amostrado (KVASER, 2017). A Tabela 1, representa os máximos valores de comprimento dos cabos de acordo com a taxa de transmissão.

Tabela 1 - Valores típicos de velocidades de transmissão de acordo com o comprimento do barramento.

Velocidade de transmissão	Tempo p/ transmitir 1 bit	Comprimento do barramento
1 Mbit/s	1 μ s	≤ 40 m
500 kbit/s	2 μ s	100 m
250 kbit/s	4 μ s	200 m
125 kbit/s	8 μ s	500 m
10 kbit/s	100 μ s	6000 m

Fonte: Adaptado KVASER, 2017.

O meio de transmissão físico é constituído por um par de fios trançados que permitem o seu funcionamento em ambientes eletricamente ruidosos. 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 da reflexão existente na CAN de alta velocidade (VECTOR, 2017). A Figura 2 ilustra um exemplo de um barramento CAN.

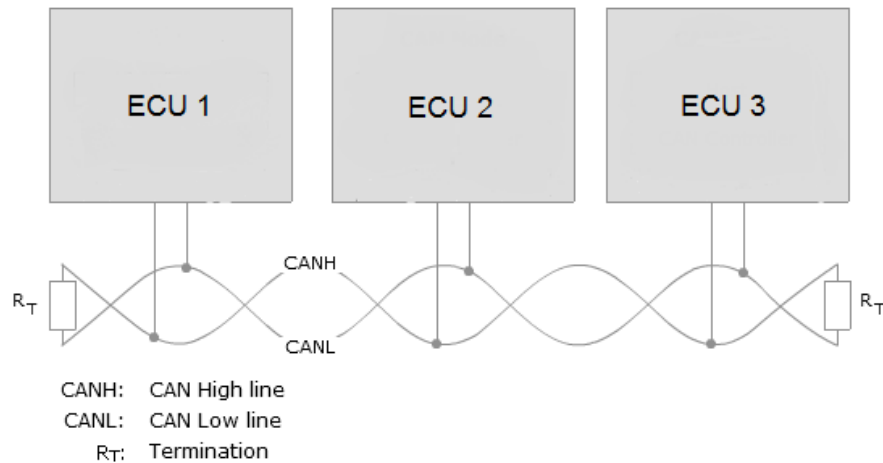


Figura 2 - Barramento CAN.
Fonte: VECTOR, 2017.

Para preservar a capacidade de comunicação do sistema em tempo real, o protocolo CAN utiliza padrão ISO 11898-1, com objetivo de garantir o transporte de dados de forma não destrutiva (CSMA/CA – *Carrier Sense Multiple Access with Collision Avoidance*). Isso assegura que todas as centrais que desejam transmitir uma mensagem esperem até que o barramento esteja livre. Assim, cada ECU monitora a atividade do barramento antes de tentar transmitir (VECTOR, 2017). Tome-se como exemplo: quando duas centrais eletrônicas tentarem enviar uma mensagem no mesmo instante de tempo, será detectada uma colisão, que será resolvida pelo processo de arbitragem, ou arbitração bit a bit, de forma que a mensagem com maior prioridade é quem vencerá a arbitragem e terá acesso ao barramento (MOURA, 2014; SOUZA, 2011).

Por último, a prioridade das mensagens é definida pelo campo identificador (do inglês, *Identifier*), ou simplesmente ID. Ele deve ser único para cada mensagem, e quanto maior for o seu valor, menor será a prioridade da mensagem. Além do mais, este campo é utilizado para filtragem das mensagens de interesse pela ECU receptora (ASSIS, 2011).

2.1.1 Modelo OSI e ISO 11898

O modelo OSI (*Open System Interconnection*) é o modelo que define os protocolos de comunicação em sete camadas, sendo elas: Física, Enlace de Dados, Rede, Transporte, Sessão, Apresentação e Aplicação (TANENBAUM, 2010). De acordo com a norma ISO 11898-1 (ISO, 2003a), o protocolo CAN é definido nas duas camadas mais baixas do modelo OSI, mais especificadamente, a camada Física e a de Enlace de Dados, exibidas na Figura 3.

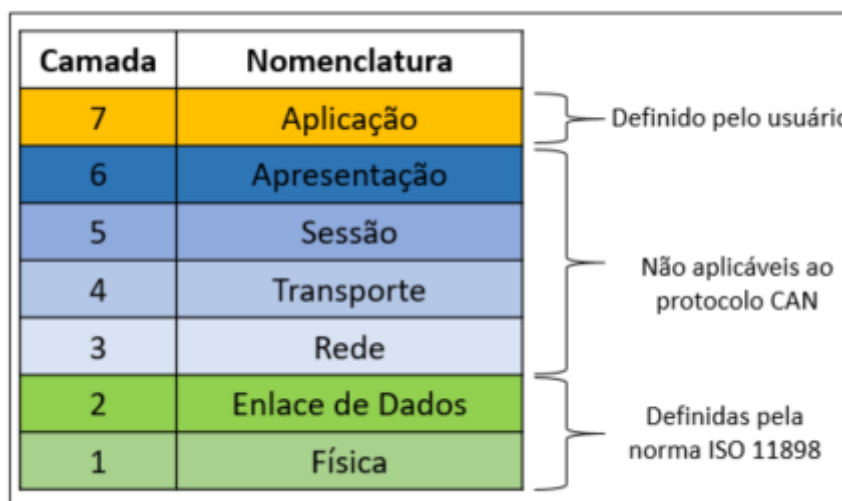


Figura 3 – Protocolo CAN dentro do modelo OSI.
Fonte: Adaptado de Lugli e Santos (2009).

A norma ISO 11898-2 (ISO, 2003b) define o protocolo CAN para aplicações em alta velocidade. E por fim, a norma ISO 11898-3 (ISO, 2006) define aplicações para o protocolo CAN em baixa velocidade. Para ambos padrões, a definição de bit recessivo e dominante é a mesma. Portanto, um bit dominante é aquele que apresenta o “nível lógico 0” e o bit recessivo é aquele que apresenta o “nível lógico 1”. Esses níveis lógicos de tensão são obtidos através da tensão diferencial entre as linhas CAN_H (do inglês, *CAN High Line*) e CAN_L (do inglês, *CAN Low Line*), indicados na Equação 1.

$$V_{diff} = V_{CAN_H} - V_{CAN_L} \quad (1)$$

Para o padrão ISO 11898-2, um bit dominante é aquele que apresenta níveis de tensões diferenciais superiores a 0,9 Volts e um bit recessivo é aquele que por definição apresenta tensões diferenciais menores que 0,5 Volts.

A Figura 4, exibe o comportamento da tensão em uma rede CAN de alta velocidade.

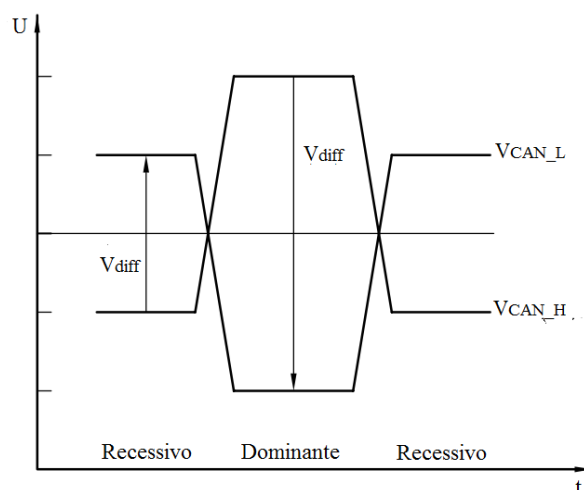


Figura 4 - Níveis de tensão em um barramento CAN de alta velocidade.
Fonte: Adaptado ISO 11898-2, 2003.

A Tabela 2, a seguir, exibe os valores de tensões máximas e mínimos para uma rede CAN de alta velocidade.

Tabela 2 – Valores mínimos e máximos para bits recessivos e dominantes no barramento CAN de alta velocidade.

Bit	Notação	Unidade	Valor		
			Mínimo	Nominal	Máximo
Recessivo	V_{CAN_H}	V	-	2,5	7,0
	V_{CAN_L}	V	-2,0	2,5	-
	V_{diff}	mV	-120	0	12
Dominante	V_{CAN_H}	V	-	3,5	7,0
	V_{CAN_L}	V	-2,0	1,5	-
	V_{diff}	V	1,2	2,0	3,0

Fonte: ISO 11898-2, 2003.

De forma análoga, a ISO 11898-3 padroniza os níveis de tensões diferenciais que definem os níveis lógicos de tensão para a rede CAN de baixa velocidade. Por se tratar de uma tensão diferencial, a norma prevê valores máximos e mínimos de tensão

para considerar se um bit é dominante ou recessivo. Estes parâmetros são apresentados na Tabela 3.

Tabela 3 - Valores mínimos e máximos para bits recessivos e dominantes no barramento CAN de baixa velocidade.

Bit	Notação	Unidade	Valor		
			Mínimo	Nominal	Máximo
Recessivo	V_{CAN_L}	V	$V_{CC} - 0,3$	-	-
	V_{CAN_H}	V	-	-	0,3
	V_{diff}	V	$-V_{CC}$	-	$-V_{CC} + 0,6$
Dominante	V_{CAN_L}	V	-	-	1,4
	V_{CAN_H}	V	$V_{CC} - 1,4$	-	-
	V_{diff}	V	$V_{CC} - 2,8$	-	V_{CC}

Fonte: Adaptado de ISO 11898-3, 2006.

A tensão V_{CC} considerada deve ser 5 Volts, este é o valor nominal segundo norma. Em condições nominais, um bit recessivo é obtido quando a tensão diferencial é -5 Volts. Já para um bit dominante a tensão diferencial deve ser no mínimo 2,2 Volts, logo o valor máximo para o CAN_L deve ser 1,4 Volts e o valor mínimo para o CAN_H é 3,6 Volts. A Figura 5, ilustra os níveis de tensões diferenciais para o barramento CAN de baixa velocidade.

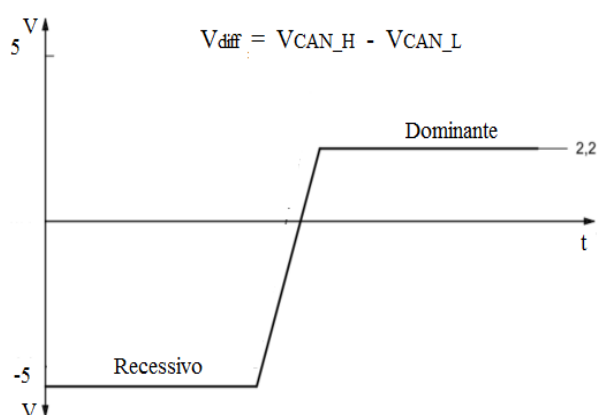


Figura 5 – Níveis de tensões diferenciais para o barramento CAN de baixa velocidade.
Fonte: Adaptado ISO 11898-3, 2006.

2.1.2 Estrutura da Mensagem – CAN Frame

Conforme FREESCALE (1998), as transferências de mensagens no barramento CAN são realizadas e controladas por quatro tipos de quadros de mensagens (*frames*):

- **Frame de dados** – carrega dados de um transmissor para um receptor;
- **Frame remoto** – usado para solicitar que outra ECU transmita uma mensagem com o mesmo identificador;
- **Frame de erro** – transmitido por qualquer ECU que detecta um erro no barramento;
- **Frame de sobrecarga** – gera um atraso extra entre os *frames* de dados e/ou remotos.

Além disso, são definidos dois formatos de CAN *frames*, cuja diferença entre eles está no tamanho do identificador. Desta forma, o formato *Standard* (CAN 2.0A), suporta IDs de 11 bits, enquanto, o formato *Extended* (CAN 2.0B) suporta IDs de até 29 bits (COOK, 2007). O formato *Standard* (11 bits) permite que 2048 identificadores diferentes sejam utilizados, do mesmo modo, o formato *Extended* (29 bits) permite o uso de 537 milhões de identificadores diferentes (CORRIGAN, 2002). Ambos formatos podem ser visualizados em destaque na Figura 6.

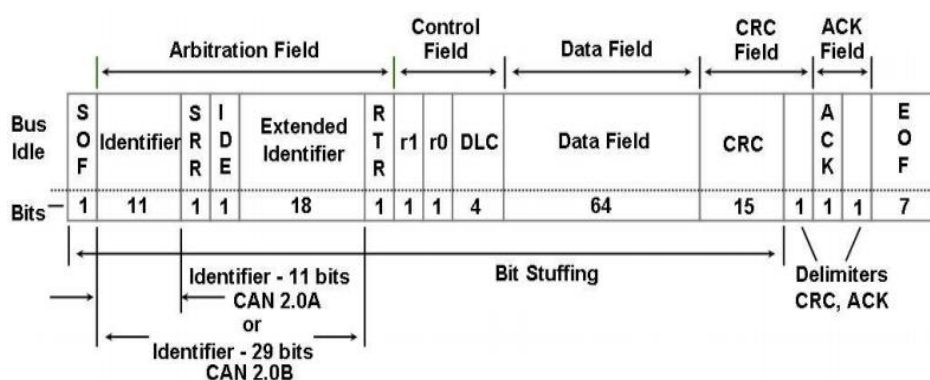


Figura 6 - *Frame* de Dados
Fonte: (GODOY, 2007)

2.1.2.1 *Frame* de Dados

Basicamente o *frame* de dados é formado por sete campos diferentes: “início do *frame*” (SOF – *Start of Frame*), “campo de arbitragem” (*Arbitration field*), “campo de controle” (*Control field*), “campo de dados” (*Data field*), “campo CRC” (*Cyclic Redundancy Check*), “campo ACK” (*Acknowledge*) e “fim do *frame*” (EOF – *End of Frame*) (ASSIS, 2011). A Figura 6, mencionada anteriormente, exhibe o formato de um *frame* de dados.

De acordo com a Figura 6, cada campo da mensagem é responsável por carregar alguma informação, sendo (BOSCH, 1991):

- SOF (1 bit): indica o início da mensagem, tanto para *frames* de dados quanto para *frames* remotos.
- Identificador (11 bits para o formato *Standard* e 29 bits para o formato *Extended*): usado para estabelecer a prioridade da mensagem, quanto menor seu valor binário maior será a prioridade da mensagem.
- SRR (*Substitute Remote Request*, 1 bit) – presente apenas no formato *Extended*, sobrescreve o bit RTR uma mensagem *Standard*.
- IDE (*Identifier Extension*, 1 bit) – identifica se o formato da mensagem é *Standard* (dominante) ou *Extended* (recessivo).
- RTR (Requisição de Transmissão Remota - *Remote Transmission Request*, 1 bit) – usado para requisitar transmissão remota, em um *frame* remoto ele deve ser recessivo.
- r0 (*Reserved bit*, 1 bit): reservado para protocolos variantes do CAN.
- r1 (*Reserved bit*, 1 bit): assim como r0 esse bit é reservado para outras aplicações, presente apenas no formato *Extended*.
- DLC (*Data Length Code*, 4 bits): define a quantidade de *bytes* de dados que serão transmitidos na mensagem CAN.
- Campo de dados (*Data field*, até 8 *bytes*): é o conteúdo útil da mensagem carrega toda a informação que será transmitida de uma central a outra.
- CRC (15 bits): é a verificação de redundância cíclica composta de 15 bits de *checksum*.

- CRC *delimiter* (1 bit): delimitador de um bit utilizado para detectar erros na transmissão de mensagens.
- ACK (2 bits) – indica se a mensagem foi recebida corretamente pela central receptora.
- EOF (7 bits) – sete bits recessivos que indicam o final da mensagem.

2.1.2.2 *Frame Remoto*

O *frame* remoto é semelhante ao *frame* de dados, exceto pelo fato que este não possui o campo de dados. Este *frame* é utilizado para solicitar a transmissão de mensagens de outras centrais eletrônicas. Assim, as ECUs transmissoras podem atuar sob demanda, ou seja, não precisam ficar enviando seus dados a todo momento. Para isto a central receptora envia um identificador igual ao da mensagem desejada (ASSIS, 2011).

2.1.2.3 *Frame de Erro*

O *frame* de erro é usado para indicar se uma central está em estado de falha. Quando uma ECU está transmitindo ou recebendo uma mensagem detectar um erro, ela irá abortar imediatamente a transmissão e enviar no barramento um *frame* de erro que consiste em seis bits dominantes e oito bits recessivos (COOK, 2007). A Figura 7, exibe o formato de um *frame* de Erro no protocolo CAN.

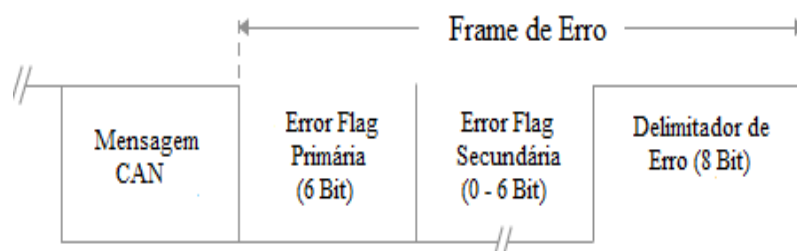


Figura 7 - Formato de um *Frame* de Erro.
Fonte: Adaptado de Vector (2017).

2.1.2.4 *Frame de Sobrecarga*

Este *frame* é gerado quando uma ECU recebe uma mensagem mais rápido do que ela pode processar, o intuito deste *frame* é ganhar tempo para que então a mensagem seja decodificada na central de destino (ASSIS, 2011).

Ele é basicamente composto de três campos: *flag* de sobrecarga, composto de seis bits dominantes, sobreposição das *flags* de sobrecarga, pode chegar a seis bits dominantes, e delimitador de sobrecarga que consiste de oito bits recessivos. Um exemplo de um *frame* de sobrecarga é exibido na Figura 8.

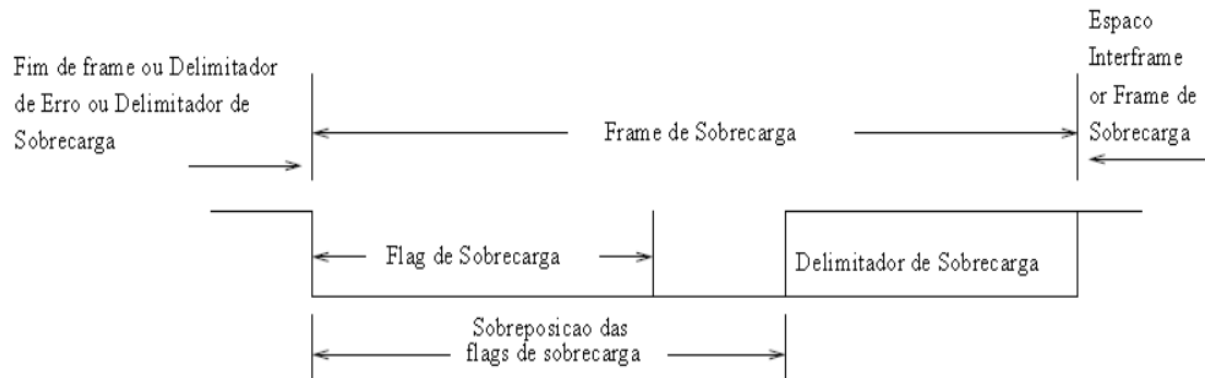


Figura 8 - *Frame* de Sobrecarga.

Fonte: (ASSIS, 2011)

2.1.2.5 Intervalo de *Interframe*

O intervalo de *interframe* consiste em três bits recessivos de intermissão. Durante a intermissão, não é permitido nenhuma central iniciar a transmissão, se por alguma condição um bit dominante é detectado durante o intervalo de intermissão, um *frame* de sobrecarga é gerado (COOK, 2007).

Após o intervalo de *interframe* o barramento ficará livre até que um bit dominante seja percebido para indicar o SOF de uma nova mensagem (COOK, 2007).

2.1.2 Bit Stuffing

Os padrões de bit “000000” e “11111111” são usados no *frame* de erro, portanto, é essencial que esses padrões sejam evitados quando transmitindo uma mensagem. Para isto ocorrer, o protocolo CAN solicita que um bit com a polaridade oposta seja transmitido sempre que detectado 5 bits de mesma polaridade consecutivos na transmissão. Esse processo de adição de um bit com polaridade reversa é referido como *bit stuffing* (DAVIS et al., 2007). A Figura 9, exemplifica como é realizado o processo de *bit stuffing*.

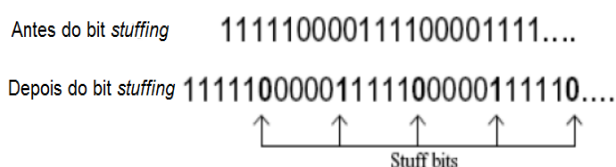


Figura 9 – Processo de *bit stuffing*.
Fonte: Adaptado de Davis et al. (2007)

Quando a mensagem é recebida na central receptora, esta é encarregada, de durante a leitura, retirar automaticamente o *stuff bit* (GUIMARÃES, SARAIVA, 2002).

2.1.3 Arbitragem

De acordo com tutorial do protocolo CAN da KVASER (2017), o processo de arbitragem é extremamente importante para a gestão da largura de banda real disponível para a transmissão de dados. O processo de arbitragem ocorre sempre que duas ou mais ECUs tentam acessar o barramento.

De maneira simplificada sempre que duas ou mais centrais tentam transmitir no barramento CAN, esse conflito é resolvido por uma arbitragem bit-a-bit usando o campo identificador da mensagem. O processo de arbitragem garante que nenhuma informação seja perdida durante a transmissão. Em uma situação, que um *frame* de dados e *frame* remoto de mesmo ID tentam acessar o barramento, o *frame* de dados tem preferência sobre o *frame* remoto (BOSCH, 1991).

A arbitragem utiliza o processo CSMA/CA. Isso faz com que a mensagem de maior prioridade seja a mensagem a ser transmitida no barramento. Com já mencionado, a prioridade de uma mensagem é assinada pelo seu campo identificador tal que quanto maior o seu valor binário menor será a sua prioridade (COOK, 2007).

No processo de arbitragem, um bit dominante, “nível lógico 0”, tem sempre preferência sobre um bit recessivo, “nível lógico 1”. Portanto, quando os IDs das mensagens estão sendo comparados, bit a bit, se uma central que envia um bit recessivo, enquanto as demais enviam um bit a dominante, esta perde o processo de arbitragem e muda seu estado para recepção (MOURA, 2014). A Figura 10, exemplifica o processo de arbitragem no barramento CAN.

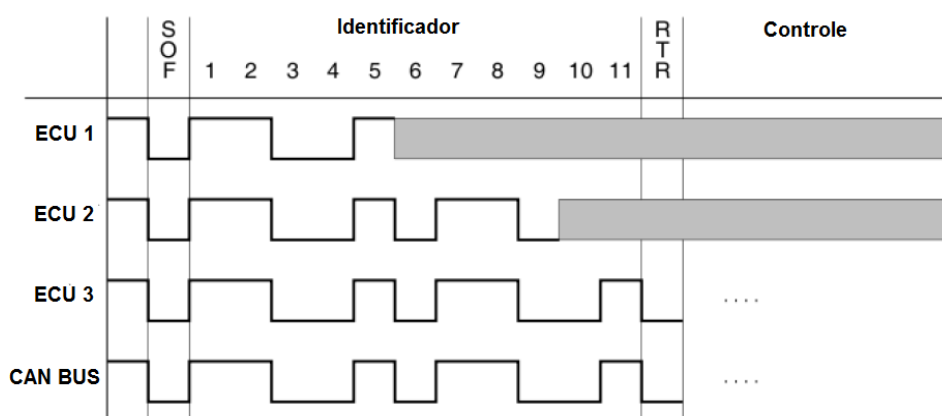


Figura 10 - Processo de Arbitragem na rede CAN.
Fonte: Adaptado de (JOHANSON; TORNGREN; NIELSEN, 2005).

Observa-se que a mensagem enviada no barramento CAN, “CAN Bus”, foi a mensagem proveniente da ECU 3, pois quando a ECU 2 envia um bit recessivo na posição 10 do campo identificador, esta muda o seu status para recepção, assim, a ECU 3 ganha total acesso ao barramento para terminar de enviar sua mensagem.

Uma outra maneira de exemplificar o processo de arbitragem é o fluxograma da Figura 11, na qual o barramento é monitorado o tempo todo. Assim, quando o nível lógico enviado da mensagem é igual ao nível lógico do barramento a mensagem continua a transmitir sua mensagem até o final do seu campo de arbitragem, ao final do campo de arbitragem a mensagem é enviada no barramento. No entanto, se uma das centrais enviar um bit recessivo enquanto no barramento lê-se um bit dominante, essa central muda o seu *status* para recepção e tentará enviar sua mensagem novamente quando o barramento estiver livre.

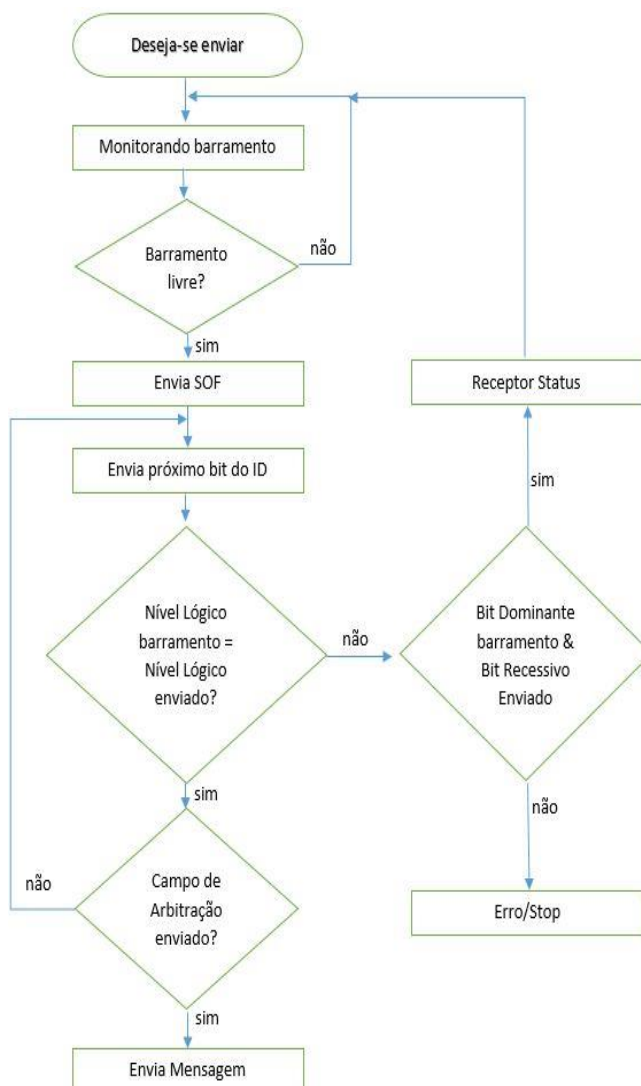


Figura 11 - Fluxograma de acesso ao barramento CAN.
Fonte: Adaptado de VECTOR.

2.1.4 Bus Load

A carga no barramento, ou do inglês *bus load*, é definida como a medida da largura de banda utilizada, indicando quanto da capacidade do sistema está sendo consumida em um dado intervalo de tempo. Esta medida é dada em porcentagem (MOURA, 2014).

Um ponto negativo da rede CAN é que com o aumento do *bus load* a latência, ou tempo de resposta da mensagem tende a aumentar. Uma vez que isto ocorre,

mensagens de prioridade baixa podem sofrer atrasos inesperados, de forma que estas podem deixar de serem enviadas no barramento. Quando uma mensagem deixa de ser enviada, um erro é sinalizado na central eletrônica receptora (COOK, 2007).

De modo geral, o *bus load* pode ser calculado através dos seguintes passos (KVASER, 2017):

- **Passo 1:** Escolha um intervalo de tempo, este deve ser maior que a mensagem menor período (geralmente é usado 1 segundo);
- **Passo 2:** Identificar todas mensagens periódicas;
- **Passo 3:** Calcular a quantidade de bits por mensagem;
- **Passo 4:** Multiplicar o tamanho da mensagem em bits pela quantidade de vezes que ela é enviada no período analisado;
- **Passo 5:** Somar os bits de todas as mensagens periódicas, para estimar a quantidade bits periódicos consumidos.
- **Passo 6:** Multiplicar por um fator de segurança de 1,1;
- **Passo 7:** Dividir o total de bits consumidos pelo total de bits disponíveis.

Os passos supracitados são uma estratégia para o cálculo do *bus load*, este mesmo cálculo também pode ser feito somando os tempos que cada mensagem necessita para ser fisicamente transmitida no barramento pelo tempo avaliado.

2.2 Tempo de Reposta de uma Mensagem CAN.

No domínio de redes embarcadas, a latência de uma mensagem, comumente conhecida como tempo de resposta (do inglês, *response time*) é definido como o intervalo de tempo desde a geração da mensagem em sua central transmissora até o seu recebimento na central receptora (MOURA, 2014).

De forma sucinta, o tempo resposta de uma mensagem pode ser interpretado pelos seus tempos de transmissão através de uma análise do tráfico de rede (GODOY, 2007). A Figura 12, mostra um exemplo dos tempos encontrados na transmissão de uma mensagem CAN, desde sua concepção na ECU de origem, até sua recepção na ECU de destino.

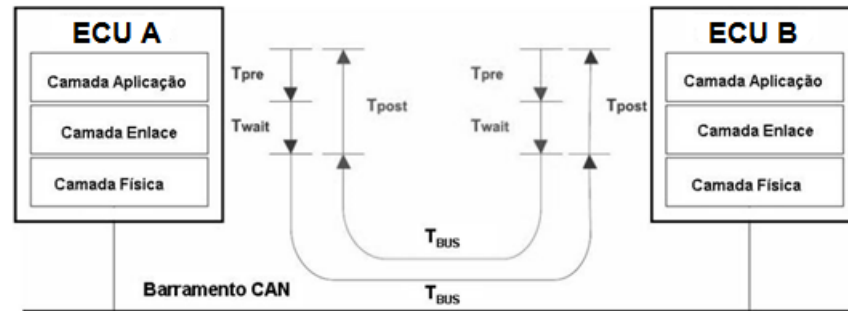


Figura 12 - Tempos de transmissão de uma mensagem CAN.
Fonte: Adaptado de GODOY, 2007.

Com base na Figura 12, a latência da mensagem, pode ser dividida em três partes: atrasos na ECU de origem (T_{pre} e T_{wait}), atrasos no barramento (T_{bus}) e atrasos na ECU de destino (T_{post}). Esses atrasos na transmissão são melhores detalhados na Figura 13.

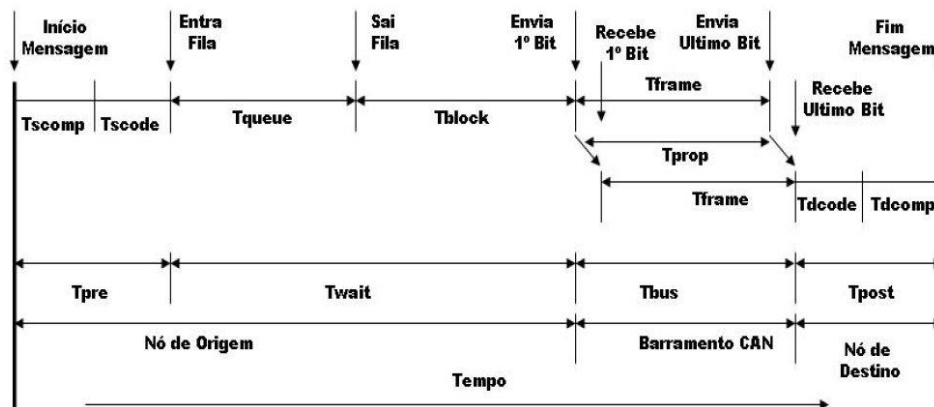


Figura 13 - Diagrama temporal da transmissão de uma mensagem CAN.
Fonte: Adaptado de (GODOY, 2007)

Como observado, o atraso na central é definido pela soma entre o tempo de pré-processamento, T_{pre} , e o tempo de espera, T_{wait} . O tempo de pré-processamento é característico do *hardware*, ele consiste do tempo de computação, T_{scomp} , e o tempo de codificação, T_{scode} . Seguidamente, o tempo de espera total, é a soma do tempo de fila, T_{queue} , o tempo que a mensagem espera no *buffer* da central de origem enquanto outra mensagem está sendo transmitida, com o tempo de bloqueio, T_{block} . De forma geral, o tempo de espera total, depende do período de amostragem das mensagens e da carga de dados a ser processada (GODOY, 2007).

O tempo de atraso no barramento CAN, T_{bus} , é o tempo de transmissão de uma mensagem no barramento, contabilizando o atraso de propagação na rede, T_{prop} , e T_{frame} , que depende do tamanho da mensagem e da velocidade da rede (GODOY, 2007).

Por fim, o atraso na central de destino, T_{dest} , é caracterizado pelo tempo de pós-processamento, T_{post} , que é resultado da soma do tempo de computação, T_{dcomp} , com o tempo de decodificação, T_{dcode} , na central receptora (GODOY, 2007).

Portanto, o tempo de resposta de uma mensagem, R_m , pode ser definido pela soma dos tempos de pré-processamento, T_{pre} , tempo de espera, T_{wait} , tempo de atraso no barramento, T_{bus} e o tempo de pós-processamento, T_{post} , como mostra a Equação 2, (GODOY, 2007).

$$R_m = T_{pre} + T_{post} + T_{wait} + T_{bus} \quad (2)$$

A soma do tempo de pré-processamento com o tempo de pós-processamento é conhecida como *Jitter*, na qual este depende de características de *software* e *hardware* usados nas ECUs transmissoras (GODOY, 2007). De acordo com SANTOS et al. (2004), o *Jitter* é a variabilidade do intervalo de tempo entre mensagens consecutivas inseridas e retiradas da fila de mensagens da rede.

Desta forma, o *Jitter* pode ser representado pela Equação 3, e seu valor é tipicamente constante e pequeno.

$$J_m = T_{pre} + T_{post} \quad (3)$$

Simplificando, o tempo de resposta de uma mensagem CAN pode ser calculado pela Equação 4:

$$R_m = J_m + T_{wait} + T_{bus} \quad (4)$$

A análise feita por Tindel et all. (1994a, 1994b) apenas substitui as variáveis T_{wait} e T_{bus} por W_m e C_m , respectivamente.

3 ANÁLISE DO TEMPO DE RESPOSTA EM MENSAGENS CAN

Como o principal objetivo desta monografia é avaliar o método do *Worst Case Response Time* (WCRT), proposta por Tindell et al. (1994a), utilizando os resultados das simulações do *software* SymTA® de cenários presentes na literatura. É necessário um completo entendimento das premissas estabelecidas por Tindell em seu método.

A seguir serão apresentadas, a metodologia de Tindell et al. (1994a, 1994b, 1995) para o cálculo do WCRT, e posteriormente será listado algumas limitações do método identificada por Davis et al. (2007), do qual ele destaca que o método de WCRT pode ser otimista em algumas situações. Por fim, os resultados provenientes do SymTA®, serão usados para avaliar o pessimismo do WCRT, diferentemente da visão de Davis et al. (2007).

3.1 Premissas

Para a análise do WCRT, algumas características que não ocorrem na prática devem ser consideradas. É evidente que o WCRT não é um método exato para o cálculo da latência da mensagem. Mas mesmo assim, ele é uma excelente ferramenta utilizada pelos engenheiros na confecção de topologia veiculares (DAVIS, 2007).

A método do WCRT assume que todas as mensagens da rede são sincronizadas para o envio, ou seja, a partir do instante inicial $t = 0s$ todas as mensagens já estão em fila prontas para serem transmitidas no barramento. A ordem em que estas mensagens serão enviadas depende de suas respectivas prioridades. Contudo, se tratando de um sistema veicular real, esta condição é extremamente improvável de acontecer. Pois a colisão de mensagens acontece de forma bem dinâmica em uma rede veicular. Por exemplo, existem centrais que iniciam a sua transmissão no momento que o motor é ligado, outras que são alimentadas pela bateria, além disso, as características de *hardware* e *software* podem impactar no tempo de inicialização da central. Portanto, afirmar que todas as mensagens estão

prontas para serem enviadas no mesmo instante de tempo é uma possibilidade remota de ocorrer.

Segundo Tindell et al. (1995), o *Worst Case Response Time*, de uma determinada mensagem, é definido como sendo o maior tempo entre o enfileiramento de uma mensagem e o tempo que esta é recebida na central de destino, indicado como R_m .

Uma mensagem CAN é assinada com um único identificador e possui entre 0 a 8 bytes de dados.

Adicionalmente, o tempo máximo para o recebimento da mensagem na central receptora (do inglês, *deadline*) é denotado D_m , em hipótese alguma, deve ser maior que o período da mensagem, T_m .

Basicamente, estas são as premissas de âmbito geral para iniciar o estudo sobre o WCRT e suas limitações. As considerações específicas do WCRT serão referidas e comentadas quando necessário.

3.2 Análise Clássica do WCRT

No início dos anos 90, um comum equívoco sobre a rede CAN era que embora o protocolo fosse bastante confiável para transmitir mensagens de alta prioridade com latência baixa, não era possível garantir que as mensagens de menor prioridade respeitariam seu tempo máximo para envio.

Não obstante, a abordagem feita por Tindell, Burns & Wellings (1995) deu suporte ao modelo de análise para o cálculo do tempo de resposta em mensagens CAN. Esta análise propôs um método para calcular o tempo de resposta de todas as mensagens na condição de pior caso (WCRT). A partir deste método, foi possível prever o comportamento temporal das mensagens CAN, e garantir que estas fossem recebidas na central receptora no seu tempo esperado (DAVIS et al., 2007).

O escalonamento de mensagens CAN é análogo ao escalonamento de tarefas em um processador de prioridade fixa, proposto por Joseph e Pandya (1986). Contudo, Tindell et al. (1994a, 1994b, 1995) incorpora em sua análise o fator *blocking time* e *jitter*, que serão detalhados no decorrer desta monografia.

O protocolo CAN trabalha com um algoritmo de prioridade fixa. No entanto, nem sempre a mensagem que possui maior prioridade será transmitida, uma vez que uma mensagem de elevada prioridade não pode interromper uma mensagem de menor prioridade que já iniciou sua transmissão, isso justifica o fator *blocking time*. A Equação 5, apresentada a seguir, define o WCRT, denotado R_m , segundo Tindell et al. (1994a, 1994b, 1995).

$$R_m = J_m + W_m + C_m \quad (5)$$

Sendo J_m o *jitter* de enfileiramento de uma mensagem m , derivado de central eletrônica transmissora. Sendo este a máxima variação do tempo gasto para deixar a mensagem pronta para a transmissão. W_m , representa o atraso de enfileiramento de uma mensagem m no pior caso, é o maior tempo que uma mensagem pode ser enfileirada e atrasada devido a outras mensagens de maior e menor prioridade que estão sendo enviadas no barramento CAN. Por fim, C_m , caracterizado por ser o tempo gasto para transmitir a mensagem fisicamente no barramento, uma vez que este está livre, também conhecido como atraso na transmissão, como descrito por Tindell et al. (1995).

Um dos pressupostos explícitos por Tindell et al. (1994a, 1994b, 1995) é que uma mensagem é dita escalável somente se:

$$R_m \leq D_m \quad (6)$$

A *deadline* de cada mensagem deve ser menor ou igual ao seu período.

$$D_m \leq T_m \quad (7)$$

Além disso, uma restrição do WCRT é que a mensagem enfileirada, necessariamente tem que ser enviada antes do término de seu período, T_m , isso é extremamente importante para evitar que a mensagem seja sobrescrita. Por isso, o WCRT deve respeitar a inequação descrita abaixo.

$$R_m \leq T_m - J_m \quad (8)$$

Desta perspectiva, observa-se que o *jitter* necessariamente deve ser menor que o período da mensagem. A Figura 14, explicita as relações supracitadas.

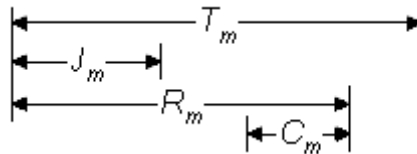


Figura 14 – Modelo temporal para mensagens CAN.
Fonte: (TINDELL et al., 1994b)

Relembrando a Equação 5, o termo W_m , é dado pela Equação 9:

$$W_m = B_m + \sum_{\forall j \in hp(m)} \left[\frac{W_m^n + J_j + \tau_{bit}}{T_j} \right] C_j \quad (9)$$

Na qual $hp(m)$ é o conjunto de mensagens com prioridades superior a m . T_j é o período da mensagem j em análise. Em seguida, o termo J_j caracteriza o *jitter* da mensagem j . O tempo de bit, τ_{bit} , consiste no tempo para transmitir 1 bit no barramento, este é associado com a velocidade da rede CAN (por exemplo, em um barramento CAN de 1 Mbit/s esse valor é 1 μ s). C_j é tempo para transmitir a mensagem j no barramento. Finalmente, B_m , *blocking time*, definido como maior tempo que uma determinada mensagem m pode ser atrasada por mensagens de baixa prioridade. Ele é igual ao tempo gasto para transmitir a maior mensagem de menor prioridade que ainda não foi transmitida no barramento, definido pela Equação 10.

$$B_m = \max_{\forall j \in lp(m)} (C_k) \quad (10)$$

Sendo, $lp(m)$ o conjunto de mensagens com baixa prioridade.

Nota-se que na Equação 9 o termo W_m aparece em ambos os lados, como consequência, não é possível escrever a equação em termos de W_m . Portanto, uma simples solução é utilizar a relação de recorrência descrita pela Equação 11.

$$W_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left[\frac{W_m^n + J_j + \tau_{bit}}{T_j} \right] C_j \quad (11)$$

Devido a relação de recorrência, o termo W_m aumenta uniformemente. O valor zero, para W_m^0 , é adequado para iniciar as iterações. As iterações continuam até atingir a sua convergência, neste caso $W_m^{n+1} = W_m^n$.

Por fim, o último termo da equação do WCRT, tempo gasto para transmitir a mensagem fisicamente no barramento, C_m , é calculado pela Equação 12.

$$C_m = \left(\left\lceil \frac{34+8s_m}{5} \right\rceil + 47 + 8s_m \right) \tau_{bit} \quad (12)$$

O termo s_m fornece o campo de dados da mensagem em *bytes*, assim para converter este valor para bits é necessário multiplicá-lo por 8, como observado na Equação 12. Em uma mensagem CAN, este valor é fornecido pelo campo DLC.

O valor 47, corresponde aos demais bits da mensagem CAN *Standard*, sendo eles: SOF (1 bit), ID (11 bits), IDE (1 bit), RTR (1 bit), r0 (1 bit), DLC (4 bits), CRC (16 bits), ACK (2 bits), EOF (7 bits) e *interframe* (3 bits). Destes 47 bits, 34 bits estão sujeitos ao bit *stuffing*. Portanto, o termo $\left\lceil \frac{34+8s_m}{5} \right\rceil$, corresponde a quantidade de bits de *stuff* adicionados no pior caso.

Não obstante, Punnekkat et al. (2000), aperfeiçoou a Equação 12 apresentada por Tindell et al. (1994a, 1994b, 1995). A fórmula de Punnekkat et al. (2000), apresentada na Equação 13, considera que os bits de *stuffing* também devem ser levados em consideração quando o processo de bit *stuffing* ocorre.

$$C_m = \left(\left\lceil \frac{g+8s_m-1}{4} \right\rceil + g + 13 + 8s_m \right) \tau_{bit} \quad (13)$$

Na qual, g é a quantidade de bits de *overhead* sujeitos ao bit *stuffing*. Em outras palavras, g representa a quantidade de bits de controle submetidos ao processo de bit *stuffing*. Seu valor é normalizado de acordo com o padrão da mensagem, exibido na Tabela 4.

Tabela 4 – Quantidade de bits sujeitos ao bit *stuffing*.
Valores de g

CAN 2.0 A <i>Standard</i>	CAN 2.0 B <i>Extended</i>
34	54

Fonte: Adaptado de DAVIS, 2007.

O valor 13 da Equação 13, corresponde aos bits dos campos das mensagens que não são sujeitos ao processo de bit *stuffing*, sendo eles: CRC *delimiter* (1 bit), ACK (2 bits), EOF (7 bits) e *interframe* (3 bits).

Assim, simplificando a Equação 13, obtém-se, para identificadores de 11 bits, Equação 14:

$$C_m = (55 + 10s_m)\tau_{bit} \quad (14)$$

E para identificadores de 29 bits, Equação 15:

$$C_m = (80 + 10s_m)\tau_{bit} \quad (15)$$

Contudo, Burns et al., (2002) modificou a Equação 12, proposta por Tindell para remover uma pequena fonte de pessimismo. Ele fez isso separando o intervalo de *interframe* do *frame* de dados. Assim, assumindo nenhum erro, máxima condição de bit *stuffing* e não incluindo o intervalo de *interframe* tem se a Equação 16:

$$C_m = \left(\left\lceil \frac{g+8s_m-1}{4} \right\rceil + g + 10 + 8s_m \right) \tau_{bit} \quad (16)$$

3.3 SymTA Vision® Análise Temporal

O SymTA Vision® é uma ferramenta computacional de análise, otimização e verificação temporal para modelos automotivos, dentre outros. Ele faz basicamente três tipos de análises (SYMTAVISION, 2017):

- **Worst Case Analysis:** para planejamento, otimização e verificação das limitações temporais;

- **Distribution Analysis:** para obter estatísticas temporais e probabilidade de violação de *deadline* de mensagens, planejamento, otimização e verificação de limitações temporais;
- **Scenario Analysis:** para planejamento, otimização e verificação de um cenário específico definido pelo usuário.

O SymTA® apresenta uma consistente e eficiente perspectiva temporal para planejamento, implementação, verificação de centrais eletrônicas, redes automotivas e sistemas distribuídos de tempo real.

A abordagem do tempo de resposta (*Response Time Approach*) feita pelo software SymTA Vision® é baseada em dois questionamentos chaves, sendo eles (SYMTA, 2013):

- Em quais condições o WCRT pode acontecer? A situação do pior caso é sempre uma condição crítica para o sistema em análise.
- O que irá ocorrer nesta situação? Quando a situação crítica acontece, o comportamento temporal da mensagem é modificado, onde ela deixa de ser enviada se este valor atinge o *deadline* da mensagem. O tempo de resposta no pior caso é calculado com base no atraso nas mensagens de menor prioridade devido ao enfileiramento de mensagens de maior prioridade.

Desta forma, para executar a análise do WCRT o SymTA® considera que todas as mensagens são enfileiradas no mesmo instante de tempo, logo, o pior caso. Não obstante, paralelamente a isto, o SymTA® simula condições de enfileiramento de forma randômica devido as condições de inicialização e característica do *hardware* de cada central. Assim, o tempo de resposta das mensagens podem sofrer variações se aproximando de uma situação de rede CAN no ambiente automotivo.

Para esta monografia, todas as três vertentes de análise do SymTA® serão utilizadas, como exemplificado na Figura 15. Será modelado um cenário baseado na literatura conhecido como SAE *Benchmark* (*Scenario Analysis*). Feito isto, é realizada a avaliação do WCRT (*Worst Case Analysis*), e por fim simulação de um banco de dados de amostras de tempos de respostas baseadas em simulações randômicas (*Distribution Analysis*).

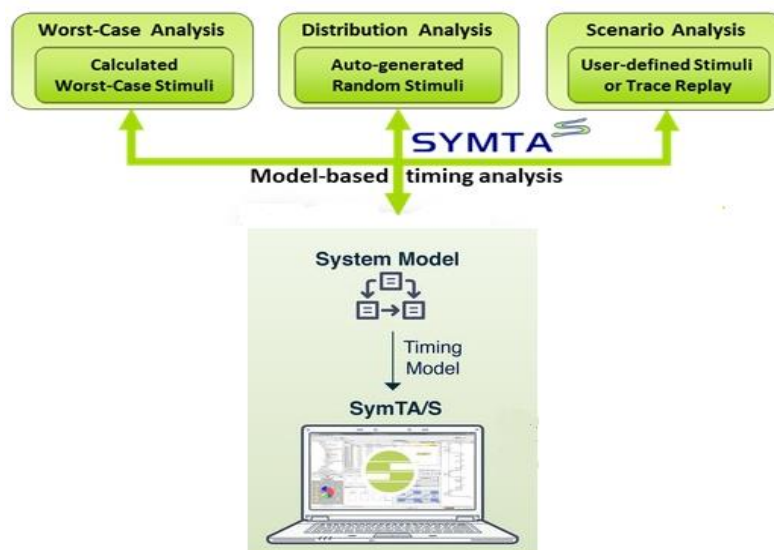


Figura 15 – Esquema de modelagem SymTA Vision®.
 Fonte: Adaptado de SYMTAVISION (2017).

3.4 Análise Otimista do WCRT

Davis et al. (2007) em sua análise afirmou que em alguns casos o método do WCRT apresentado por Tindell et al. (1994a, 1994b, 1995) pode ser otimista, ou seja, o valor do WCRT apresenta um valor menor do que de fato acontece na realidade. Para ilustrar esse fato, ele apresenta o seguinte cenário: uma rede CAN com velocidade 125 Kbit/s, com três mensagens de 7 bytes de dados, considerando o identificador de 11 bits, formato *standard*, e o bit *stuffing* no pior caso. Neste caso, aplicando-se a Equação 14, calcula-se o valor de C_m para as mensagens de 1 milissegundo. Este cenário é melhor ilustrado na Tabela 5:

Tabela 5 - Exemplo: mensagens CAN.

Mensagem	Prioridade	Período	C_m
A	1	2,5 ms	1 ms
B	2	3,5 ms	1 ms
C	3	3,5 ms	1 ms

Fonte: Adaptado de Davis et al. (2007)

Do ponto de vista de Tindell et al. (1994a, 1994b, 1995), o WCRT para as mensagens A, B e C seria, respetivamente, 2 ms, 3 ms e 3 ms. Contudo, como o WCRT para as mensagens B e C é maior do que o período da mensagem de maior prioridade, isto pode causar um atraso nas transmissões subseqüente, especialmente para a mensagem C que é a menos prioritária. A Figura 16, exibe o pior caso para mensagem C.

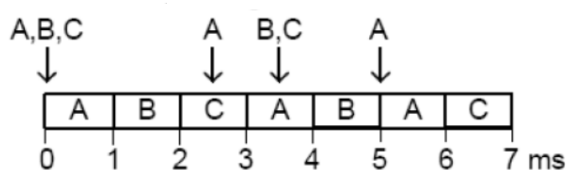


Figura 16 – Mensagens enfileiradas
Fonte: (DAVIS et al., 2007)

Nota-se que a segunda vez que a mensagem C é transmitida ocorre em $t = 7$ ms resultando em um tempo de resposta de 3,5 ms, uma vez que esta é enfileirada novamente em $t = 3,5$ ms. Desta forma, o atual WCRT para a mensagem C é 3,5 ms valor superior ao calculado pela análise de Tindell et al, (1994a, 1994b, 1995).

3.5 Análise Pessimista do WCRT

Um dos objetivos deste trabalho é realizar a avaliação do WCRT, para tal, o uso do *software* SymTA Vision® foi proposto. A principal vantagem deste *software* é que além de realizar a análise do WCRT, ele estabelece premissas reais para o funcionamento da rede CAN veicular. Ou seja, a colisão de mensagens não acontece de forma previsível como na análise de Tindell, na qual todas mensagens são enfileiradas e sincronizadas antes do envio.

Em uma rede CAN implementada fisicamente em um veículo, não é necessário que todas as centrais estejam prontas para transmitir, para a comunicação no barramento se estabelecer. Assim, a medida que as ECUs vão se inicializando elas

vão solicitando acesso ao barramento. Desta forma, a quantidade de mensagens que disputam o barramento por intervalo de tempo é menor. Logo, pressupõe-se que a quantidade de colisões de mensagem também decresça, conseqüentemente que seu tempo de resposta seja menor do que o calculado pelo WCRT.

Para provar este fato, que a análise do WCRT, representa uma análise pessimista o cenário conhecido como SAE *Benchmark*, disponível na literatura, foi escolhido para ser simulado e estudado, sendo o mesmo apresentado na seção 3.6.

3.6 SAE *Benchmark*

O caso a ser avaliado é conhecido como *Automotive Engineers (SAE) Benchmark*, amplamente difundido por Tindell et al., (1994a) e adaptado por Burns et al. (2002). O cenário consiste de 17 mensagens exibidas na Tabela 6. O período das mensagens varia entre 5 ms e 1 s, velocidade da rede de 125 kbit/s e utilização do barramento de aproximadamente 85%, o que por sua vez pode ser considerado um *bus load* elevado. A análise prévia realizada por Burns, evidencia que as mensagens 12, 9, e 8 são particularmente vulneráveis, pois o seu *Worst Case Response Time (R_i)* é muito próxima de seu *deadline*.

Este cenário referido como SAE *Benchmark*, vem sendo usado como base em vários trabalhos que visam exemplificar o comportamento temporal de mensagens na rede CAN. Por conseguinte, ele será usado também como referência neste trabalho.

Por outras palavras, esse cenário será modelado no SymTA Vision®, onde serão comparadas as respostas do WCRT, da literatura com a simulação implementada. Além disso, os dados gerados pelo *software* possibilitarão análise do comportamento temporal das mensagens supracitadas.

Tabela 6 - Conjunto de mensagens CAN SAE.

Mensagem	Prioridade (ID)	S _m (Bytes)	T _i (ms)	D _i (ms)	R _i (ms)
17	1	1	1000	5	1,416
16	2	2	5	5	2,016
15	3	1	5	5	2,536
14	4	2	5	5	3,136
13	5	1	5	5	3,656
12	6	2	5	5	4,256
11	7	6	10	10	5,016
10	8	1	10	10	8,376
9	9	2	10	10	8,976
8	10	2	10	10	9,576
7	11	1	100	100	10,096
6	12	4	100	100	19,096
5	13	1	100	100	19,616
4	14	1	100	100	20,136
3	15	3	1000	1000	28,976
2	16	1	1000	1000	29,496
1	17	1	1000	1000	29,520

Fonte: Adaptado de Burns et al., 2002.

3.7 Modelagem com SymTA Vision®

A modelagem feita no SymTA® foi fundamentada a partir das informações contidas na Tabela 6, supracitada, na qual seus parâmetros foram adaptados para servirem como entrada para o *software*. Adicionalmente, para configurar um modelo de rede CAN no SymTA® é necessário: definir a quantidade de mensagens, o nome das mensagens, o tipo de transmissão, formato CAN (*Standard* ou *Extended*), prioridade de mensagens em hexadecimal, ativação interna (período) e o tamanho da mensagem em *bytes*.

A Figura 17 ilustra como é feita a parametrização de uma mensagem no SymTA®, os campos relevantes estão enumerados de 1 a 6, em vermelho:

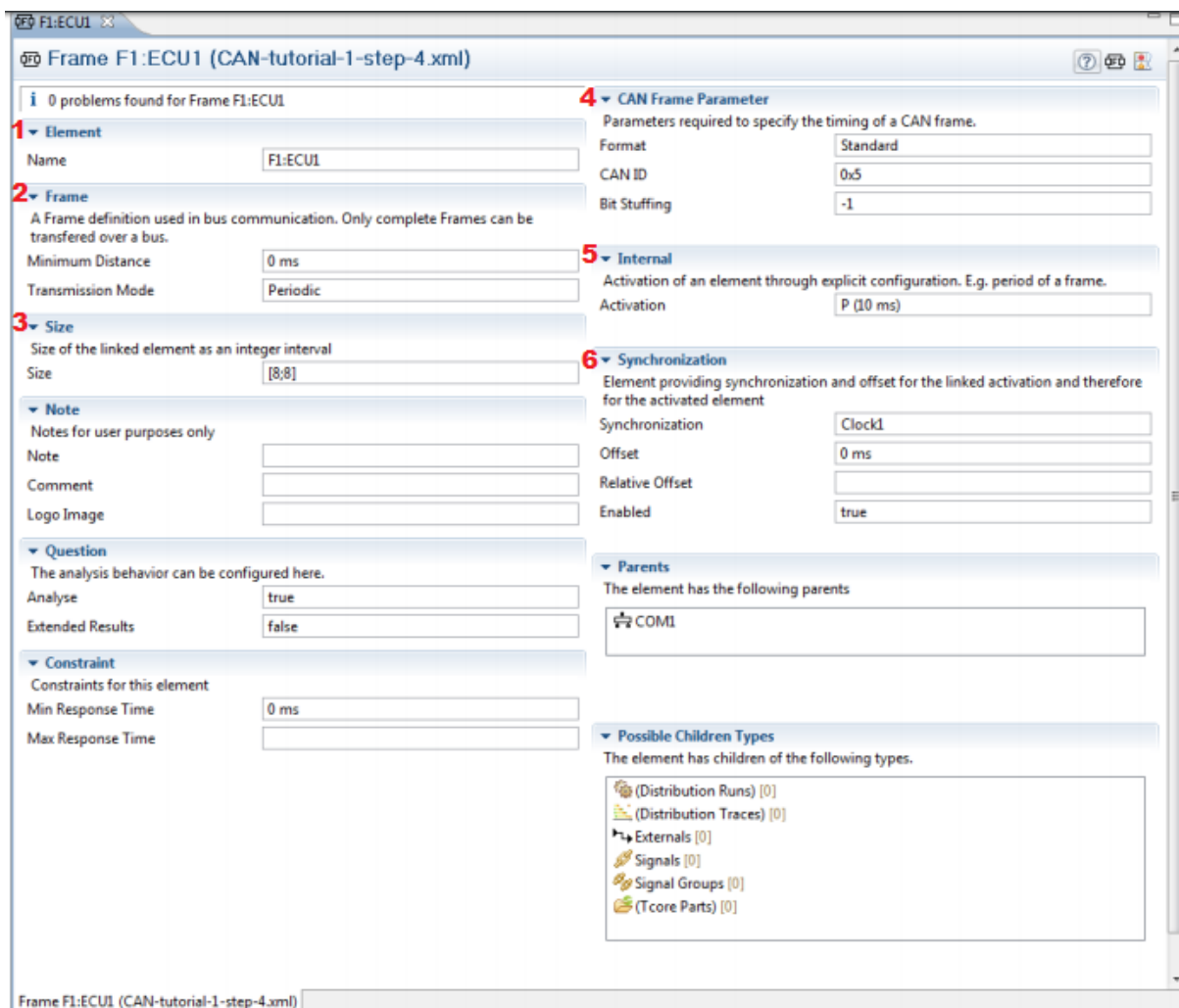


Figura 17 - Configuração CAN frame SymTA®.
Fonte: (SYMTA, 2012)

Depois que uma ECU é criada, as mensagens são criadas conforme a Figura 17. Suas principais características estão numeradas, em vermelho, de 1 a 6 na figura acima:

- **1 – Element:** descreve o nome da mensagem;
- **2 – Frame:** descreve o tipo de transmissão, por exemplo, cíclica, por evento, mista;
- **3 – Size:** define o tamanho da mensagem em *bytes (payload)*;
- **4 – CAN frame parameter:** define o formato da mensagem e prioridade;
- **5 – Internal:** representa a ativação interna de acordo com o tipo de transmissão, por exemplo, se a mensagem é cíclica esse parâmetro fornece o período da mensagem;

- **6 – Synchronization:** define as características de inicialização da central eletrônica;

A Tabela 7, exibida a seguir, ilustra como foi configurado a rede para realizar a análise deste trabalho.

Tabela 7 - Configuração dos parâmetros de entrada do SymTA®.

Nome da Mensagem	Modo de Transmissão	Tipo do Formato CAN	Prioridade CAN ID (Hex.)	Período/ Ativação Interna	Tamanho da Mensagem DLC (bytes)
F17	Periódica	Standard	0x1	P(1000 ms)	1
F16	Periódica	Standard	0x2	P(5 ms)	2
F15	Periódica	Standard	0x3	P(5 ms)	1
F14	Periódica	Standard	0x4	P(5 ms)	2
F13	Periódica	Standard	0x5	P(5 ms)	1
F12	Periódica	Standard	0x6	P(5 ms)	2
F11	Periódica	Standard	0x7	P(10 ms)	6
F10	Periódica	Standard	0x8	P(10 ms)	1
F9	Periódica	Standard	0x9	P(10 ms)	2
F8	Periódica	Standard	0xa	P(10 ms)	2
F7	Periódica	Standard	0xb	P(100 ms)	1
F6	Periódica	Standard	0xc	P(100 ms)	4
F5	Periódica	Standard	0xd	P(100 ms)	1
F4	Periódica	Standard	0xe	P(100 ms)	1
F3	Periódica	Standard	0xf	P(1000 ms)	3
F2	Periódica	Standard	0x10	P(1000 ms)	1
F1	Periódica	Standard	0x11	P(1000 ms)	1

Fonte: Autoria própria.

Configurado todos os parâmetros da Tabela 7 é possível dar início à simulação. A prioridade das mensagens é assinada pelo seu campo identificador em hexadecimal, quarta coluna da Tabela 7. Quanto menor este valor, mais prioritária é a mensagem. Portanto, para o cenário em análise, a mensagem com maior prioridade é a F17. Não obstante, a mensagem com menor prioridade é referida como F1. Todas as mensagens são periódicas com o formato *Standard* do padrão CAN.

4 RESULTADOS E DISCUSSÃO

Após o estudo detalhado do método WCRT, foi modelado com auxílio do *software* SymTA® o cenário apresentado na Tabela 6, com o propósito de avaliar os resultados obtidos para o WCRT na literatura de referência e validar o uso da ferramenta para estimar tal grandeza.

Além de disso, os resultados obtidos para o WCRT são confrontados com a teoria aplicada neste trabalho, para confirmar a veracidade dos resultados apresentados por Burns et al. (2002).

Destaca-se também que para determinar o pessimismo do WCRT foram geradas diversas amostras de tempos de respostas das mensagens da presente arquitetura e comparadas com o resultado do WCRT. Para comparar tal pessimismo utiliza-se a Equação 17:

$$Pessimismo (\%) = \left(1 - \frac{R_{Max(banco\ de\ dados)}^m}{R_{WCRT}^m}\right) \cdot 100\% \quad (17)$$

Na qual, $R_{Max(banco\ de\ dados)}^m$ é o valor máximo do tempo de resposta da mensagem m do banco de dados simulado. E R_{WCRT}^m é o valor do WCRT para a mensagem m . Ao final, o pessimismo é fornecido em porcentagem.

Tal avaliação é apresentada por gráficos e tabelas para melhor exemplificar o comportamento temporal do barramento CAN em situações práticas de funcionamento. Para desta forma evidenciar a hipótese da análise pessimista produzida pelo método do WCRT.

4.1 Avaliação dos resultados obtidos por Burns et al., (2002).

O resultado para o WCRT de Burns et al. (2002) é apresentado na Tabela 6, na seção 3.6. No entanto, aplicando a Equação 16 para o cálculo do tempo para transmitir a mensagem fisicamente no barramento, C_m , como descrito na literatura, resulta em um desvio no valor de R_i apresentado.

Por exemplo, o cálculo do WCRT para mensagem de maior prioridade, M17, Tabela 7. De forma geral, consiste na soma do fator *blocking time*, intervalo de *interframe* e o tempo para transmitir a mensagem no barramento. Analisando o conjunto de mensagens da Tabela 7, para F17, o *blocking time* seria o tempo para transmitir a mensagem F11 fisicamente no barramento. Utilizando a Equação 16 obtém-se para F17 e F11, os respectivos valores de C_m , 0,498 ms e 0,898 ms. O intervalo de *interframe* para uma rede CAN de velocidade de 125 kbit/s corresponde a 0,024 ms. Portanto, o WCRT para mensagem F17 é dado pela soma a seguir:

$$R_{F17} = 0,898 + 0,024 + 0,498 = 1,420 \text{ ms}$$

E o valor apresentado na Tabela 6 é de 1,416 para esta mensagem. Resultando em uma diferença 0,004 ms.

Ao realizar análise mais detalhada da Equação 13 de Punneckkat, Harsson e Norstrom (2000) e da Equação 16 de Burns et al., (2002) foi possível identificar a causa raiz desta diferença nos valores de WCRT. O problema da equação 13 e 16 é devido ao fator relacionado ao pior caso do processo de *stuffing bits*, representado por:

$$\left\lceil \frac{g + 8s_m - 1}{4} \right\rceil$$

A limitação do fator acima é devido ao fato deste não apresentar uma divisão inteira. Por exemplo, em uma mensagem CAN *Standard* de um *byte* de *frame* de dados, o valor deste termo corresponde a 10,25 bits de *stuffing*. Este número representa a quantidade de bits adicionados no processo de *stuffing*, ou seja, não é possível adicionar 0,25 bit de *stuff*. Portanto, deve-se considerar somente a parte inteira, logo a quantidade de bits de *stuff* acrescentado é 10.

Assim, para solucionar este problema, a proposta é substituir o termo “-1” por “-2”, de forma a tornar a divisão inteira. Uma outra forma seria utilizar a função matemática “piso no fator de *stuffing bits*, $\left\lfloor \frac{g+8s_m-1}{4} \right\rfloor$, que retorna a parte inteira de um número fracionário

A Tabela 8 faz um comparativo entre a solução proposta (3º coluna) e a atual (2º coluna).

Tabela 8 - Fator de *stuffing bits*.

	Atual	Proposta
Bytes (s_m)	$\left\lceil \frac{34 + 8s_m - 1}{4} \right\rceil$	$\left\lceil \frac{34 + 8s_m - 2}{4} \right\rceil$
1	10,25	10
2	12,25	12
3	14,25	14
4	16,25	16
5	18,25	18
6	20,25	20
7	22,25	22
8	24,25	24

Fonte: Acervo do autor.

Portanto, para uma mensagem com um *byte* de *frame* dados, desconsiderando o intervalo de *interframe*, e considerado o *bit stuffing* no pior caso, o tamanho desta mensagem constitui-se de 62 bits, sendo eles: 44 bits de controle, 8 bits de dados e 10 bits de *stuffing*. Supondo uma rede de velocidade de 125 kbit/s o tempo gasto para transmitir 62 bits é de 0,496 ms. Utilizando esta abordagem, o valor encontrado para o WCRT para a mensagem F17 corresponde a exatos 1,416 ms, conforme apresentado na Tabela 6.

Deste modo, pelas as razões evidenciadas acima, a Equação 16 deve ser modificada como exibido na Equação 18, possibilitando uma divisão inteira no termo correspondente ao *bit stuffing*:

$$C_m = \left(\left\lceil \frac{g+8s_m-2}{4} \right\rceil + g + 10 + 8s_m \right) \tau_{bit} \quad (18)$$

Por fim, utilizando a Equação 18, proposta nesta monografia, os valores calculados para o WCRT foram condizentes com os valores apresentados na literatura, Burns et al., (2002). A Tabela 9, a seguir, compara os resultados obtidos utilizando as Equações 16 e 18, assim como, o erro entre eles em porcentagem.

Tabela 9 - Comparação WCRT usando as Equações 16 e 18.

Mensagem	R_m (ms)	R_m (ms)	Erro (%)	R_m (ms)	Erro (%)
	Literatura	Equação 16 (Atual)	Equação 16 (Atual)	Equação 18 (Proposta)	Equação 18 (Proposta)
F17	1,416	1,420	0,282	1,416	0,00
F16	2,016	2,022	0,298	2,016	0,00
F15	2,536	2,544	0,315	2,536	0,00
F14	3,136	3,146	0,319	3,136	0,00
F13	3,656	3,668	0,328	3,656	0,00
F12	4,256	4,270	0,329	4,256	0,00
F11	5,016	5,032	0,319	5,016	0,00
F10	8,376	8,404	0,334	8,376	0,00
F9	8,976	9,006	0,334	8,976	0,00
F8	9,576	9,608	0,334	9,576	0,00
F7	10,096	10,158	0,614	10,096	0,00
F6	19,096	19,160	0,335	19,096	0,00
F5	19,616	19,682	0,336	19,616	0,00
F4	20,136	20,232	0,477	20,136	0,00
F3	28,976	29,074	0,338	28,976	0,00
F2	29,496	29,596	0,339	29,496	0,00
F1	29,520	29,622	0,346	29,496	0,08

Fonte: Acervo do autor.

Adicionalmente, é possível também calcular o *Best Case Reponse Time* (BCRT), onde este é o menor tempo na qual a mensagem é transmitida no barramento considerando uma situação de melhor caso. O tempo de resposta no melhor caso é uma situação hipotética que ocorre quando não há a presença de *stuffing bits* e o barramento está *idle* (livre). Desta forma, o valor de R_m é o próprio C_m e a Equação 18 pode ser rescrita da seguinte forma, Equação 19:

$$C_m = (g + 10 + 8s_m)\tau_{bit} = R_m \quad (19)$$

Para ilustrar, o BCRT da mensagem F17 é dado por:

$$R_m = C_m = (34 + 10 + 8 \times 1) \frac{1}{125 \times 10^3} = 0,416 \text{ ms}$$

Aplicando a Equação 19 para as demais mensagens encontra-se os valores exibidos na Tabela 10.

Tabela 10 - *Best Case Response Time SAE Benchmark.*

Nome da Mensagem	Tamanho da Mensagem S_m (bytes)	BCRT Equação 19
F17	1	0,416 ms
F16	2	0,480 ms
F15	1	0,416 ms
F14	2	0,480 ms
F13	1	0,416 ms
F12	2	0,480 ms
F11	6	0,736 ms
F10	1	0,416 ms
F9	2	0,480 ms
F8	2	0,480 ms
F7	1	0,416 ms
F6	4	0,608 ms
F5	1	0,416 ms
F4	1	0,416 ms
F3	3	0,544 ms
F2	1	0,416 ms
F1	1	0,416 ms

Fonte: Acervo do autor.

Em suma, não é possível obter os mesmos resultados para o WCRT utilizando a formulação apresentada por Burns et al., 2002, pela Equação 16. Isto é devido ao fator relacionado ao processo de *bit stuffing*. Equivocadamente, o termo relacionado ao *bit stuffing* da Equação 16 não é uma divisão inteira, conseqüentemente, o resto da divisão quando considerado pode ocasionar um erro na computação do WCRT, aumentando-se ainda mais o seu pessimismo.

Para evitar este problema, a proposta apresentada seria utilizar a Equação 18, tornando a divisão inteira, de modo que só o quociente seria utilizado para representar a adição dos bits de *stuff* no cálculo do tempo gasto para transmitir a mensagem fisicamente no barramento.

Por fim, o BCRT é a condição na qual não há a presença de nenhum bit de *stuff*. Portanto, seu tempo de resposta é em função do campo de dados e de dos 44 bits de controle da mensagem, como exibido pela Equação 19.

4.2 Resultados do SymTA® para o WCRT.

Esta parte da monografia é dedicada a apresentação dos resultados obtidos pelo *software* SymTA Vision®. Para esta etapa, a rede CAN simulada foi configurada conforme descrito na seção 3.4. A Tabela 11 apresenta os resultados obtidos para o WCRT e BCRT.

Tabela 11 - Resultados do SymTA® para WCRT e BCRT.

Mensagem	WCRT	BCRT
F17	1,416 ms	0,416 ms
F16	2,016 ms	0,480 ms
F15	2,536 ms	0,416 ms
F14	3,136 ms	0,480 ms
F13	3,656 ms	0,416 ms
F12	4,256 ms	0,480 ms
F11	5,016 ms	0,736 ms
F10	8,376 ms	0,416 ms
F9	8,976 ms	0,480 ms
F8	9,576 ms	0,480 ms
F7	10,096 ms	0,416 ms
F6	19,616 ms	0,608 ms
F5	20,136 ms	0,416 ms
F4	20,136 ms	0,416 ms
F3	28,976 ms	0,544 ms
F2	29,496 ms	0,416 ms
F1	29,496 ms	0,416 ms

Fonte: Acervo do autor.

Nota-se que o valor para F1 e F2 é o mesmo. Isso ocorre, pois, a mensagem F1 não considera o fator *blocking time* por ser a mensagem de menor prioridade. Além disso, observa-se que valor de F1, na Tabela 11, para o WCRT é 0,024 ms menor que

o valor apresentado pela literatura de 29,520 ms. Esta diferença, corresponde ao intervalo de *interframe*. Como a rede apresenta uma velocidade de 125 kbit/s são necessários 0,024 ms para enviar três bits recessivos ao término da mensagem.

Uma outra forma de visualizar o resultado da análise do WCRT e BCRT é o gráfico do perfil do tempo de resposta para o barramento ordenado de acordo com a prioridade da mensagem, apresentado na Figura 18.

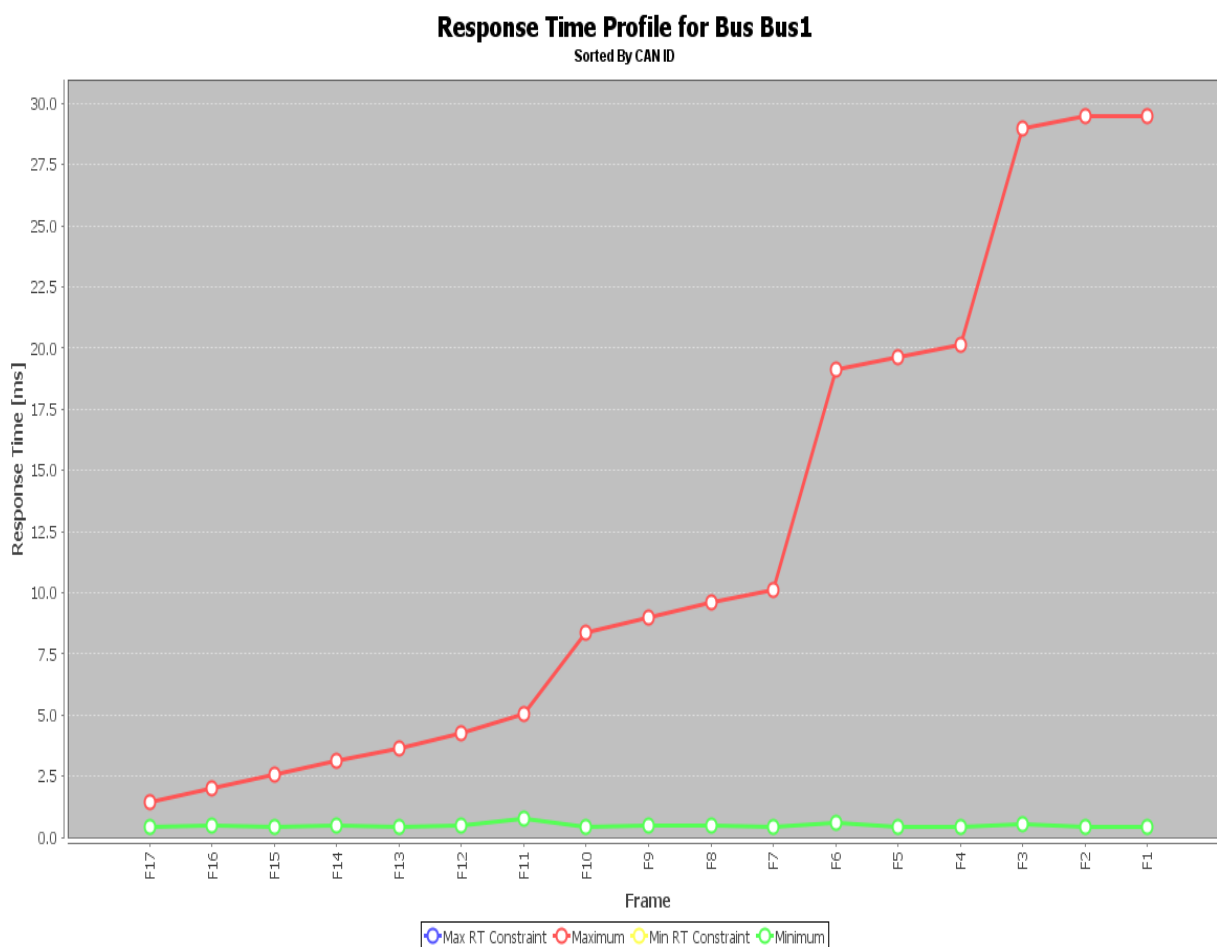


Figura 18 – Perfil do tempo de resposta para o WCRT.
Fonte: Acervo do autor, com uso SymTA®.

Segundo a Figura 18, o WCRT é exibido em vermelho e o BCRT em verde. O BCRT é melhor condição de envio da mensagem, ou seja, o menor tempo de resposta possível que a mensagem pode obter. Para isto ocorrer, o barramento necessita estar livre e nenhum bit de *stuff* ser considerado. Assim o BCRT, só depende do tamanho da mensagem.

Já o WCRT, na Figura 17, é impactado pelo tamanho da mensagem, a pior condição de bit *stuffing*, prioridade e periodicidade da mensagem. Como as

mensagens estão ordenadas de acordo com sua prioridade, ele possui uma característica crescente. Nota-se, que o coeficiente angular da reta, entre as mensagens F11 e F10, aumenta. Isto ocorreu, pois, a mensagem F11 iniciou sua transmissão antes de 5 ms e terminou em 5,016 ms. No entanto, a partir de $t = 5$ ms, as mensagens com período de 5 ms, são enfileiradas novamente e como F10 possui uma prioridade menor, ela tem que esperar o envio destas, resultando em um tempo de resposta bem maior, conforme ilustrado pela Figura 20.

De modo similar, a inclinação acentuada entre as mensagens F7 e F6, é devido ao fato da mensagem F7 ter terminado sua transmissão após 10 ms. Assim, as mensagens de período 5 ms e 10 ms, já estão em espera novamente e possuem prioridade superior a F6, como exibe a Figura 21.

Além disso, uma das vantagens de se utilizar o SymTA® está na possibilidade de verificar as colisões entre as mensagens, no momento da transmissão, via gráfico de Gantt. Esta ferramenta apresenta, visualmente, o atraso no envio das mensagens devido ao enfileiramento de mensagens prioritárias. As Figuras 19, 20 e 21, elucidam o WCRT para as mensagens F14, F10 e F6, respectivamente.

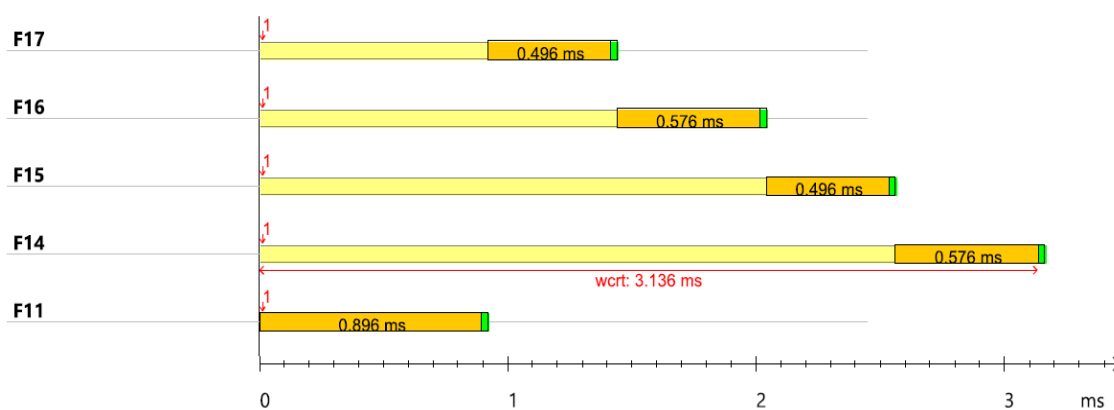


Figura 19 - Worst Case Gantt mensagem F14.
Fonte: Acervo do autor, com uso do SymTA®.

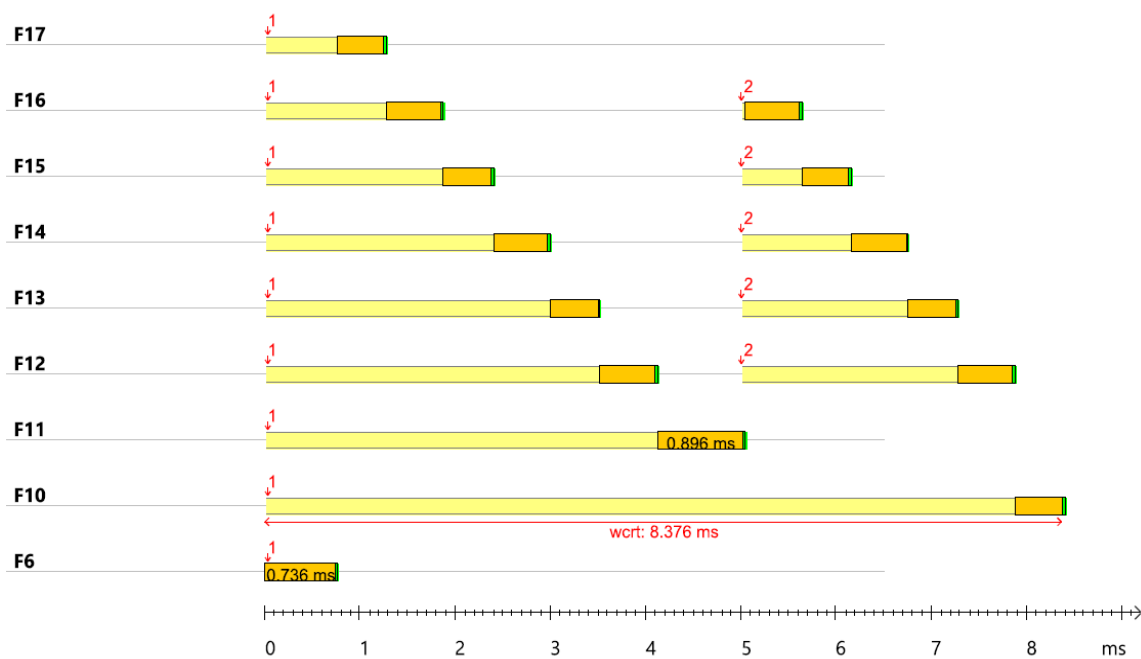


Figura 20 - Worst Case Gantt mensagem F10.
Fonte: Acervo do autor, com uso do SymTA®.

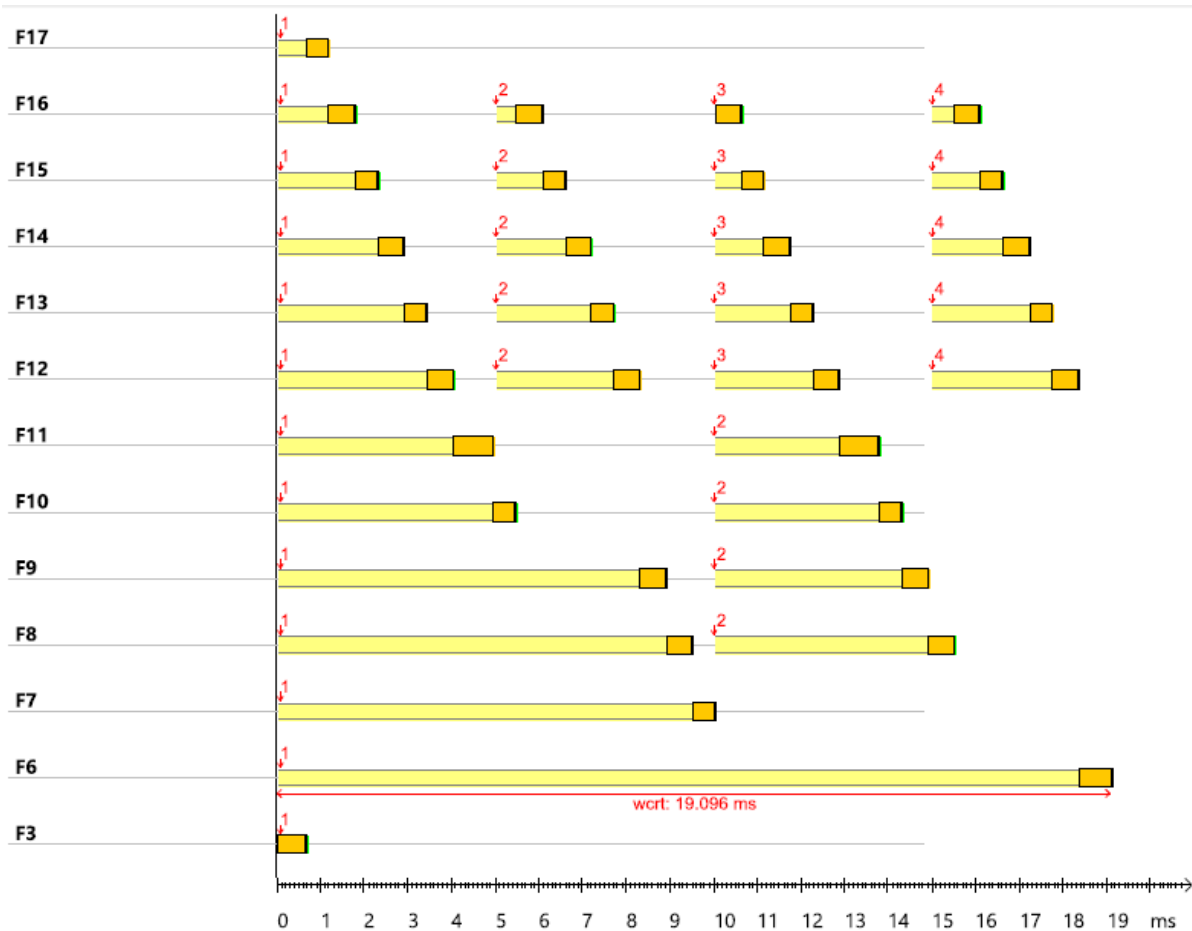


Figura 21 - Worst Case Gantt mensagem F6.
Fonte: Acervo do autor, com uso do SymTA®.

Os gráficos das Figuras 19, 20 e 21 podem ser interpretados da seguinte forma: O número em vermelho indica quando a mensagem foi enfileirada, e o número de vezes; a parte laranja da mensagem é dada pelo valor de C_m , julgando pelo valor obtido tudo leva a crer o SymTA também só considera a parte inteira do processo de bit *stuffing*, validando-se então a Equação 18; a faixa em amarelo representa o atraso causado devido ao envio de mensagens com maiores prioridades; por fim, a área verde, após o termino de envio da mensagem corresponde ao intervalo de *interframe*.

Assim, para a Figura 19, o fator *blocking* é o tempo para transmitir a mensagem F11 fisicamente no barramento. Contudo, para a Figura 20, este fator passa a ser o tempo para transmitir a mensagem F6, pois está é a maior mensagem de menor prioridade para esta situação. De forma análoga, o *blocking time* para a mensagem F6 é o tempo para transmitir F3 fisicamente no barramento acrescidos do intervalo de *interframe*.

Por fim, nota-se que, o resultado do WCRT é formado pela soma do fator de *blocking time*, atrasos causados por mensagens de maior prioridade, tempo necessário para transmitir a própria mensagem no barramento, lembrando-se que no final de cada mensagem o intervalo de *interframe* deve ser contabilizado. Isto é evidenciado nas Figuras 19, 20 e 21, sendo respectivamente o tempo de resposta no pior caso para as mensagens F14, F10 e F6, respectivamente 3,136 ms, 8,376 ms e 19,096 ms.

A visualização do WCRT em forma de gráfico Gantt é uma forma efetiva e visual de mostrar o tempo de resposta de mensagens CAN no pior caso.

Por fim, o SymTA® utiliza-se de ferramentas gráficas para ilustrar os resultados do WCRT e BCRT para deixar esses resultados mais explícitos ao usuário. Os resultados obtidos via SymTA®, para o WCRT, foram de extrema importância para validar método proposto pela Equação 18. De forma que, utilizando a Equação 18 na metodologia do clássica do WCRT, os resultados alcançados para o WCRT calculado, simulado e o exibido na literatura foram equivalentes, conforme exibidos na Tabela 12.

Tabela 12 – Comparação entre os valores do WCRT da literatura e simulação no SymTA®

Mensagem	WCRT (ms)		Diferença (%)
	Simulação	Literatura	
F17	1,416	1,416	0,00
F16	2,016	2,016	0,00
F15	2,536	2,536	0,00
F14	3,136	3,136	0,00
F13	3,656	3,656	0,00
F12	4,256	4,256	0,00
F11	5,016	5,016	0,00
F10	8,376	8,376	0,00
F9	8,976	8,976	0,00
F8	9,576	9,576	0,00
F7	10,096	10,096	0,00
F6	19,616	19,096	0,00
F5	20,136	19,616	0,00
F4	20,136	20,136	0,00
F3	28,976	28,976	0,00
F2	29,496	29,496	0,00
F1	29,496	29,520	0,08

Fonte: Acervo do autor.

A diferença percentual de 0,08% é resultado do intervalo de *interframe*, que não é contabilizado para a última mensagem enviada.

4.3 Resultados SymTA® para diferentes condições de inicialização

Além da análise do WCRT, o SymTA® propicia a análise temporal do sistema em diversas condições de inicialização, ou seja, ele não assume que todas as mensagens são sincronizadas a partir do tempo igual 0 segundos, o que por sua vez gera um alívio no tempo de resposta de mensagens menos prioritárias, pois a quantidade de mensagens disputando a transmissão do barramento naquele instante é menor. Por esta razão, essa aproximação feita pelo *software* se ajusta melhor a uma situação de funcionamento real da rede CAN implementada fisicamente em um veículo.

O SymTA® chama essa análise de *System Distribution Analysis*, essa aproximação considera *offsets* entre as mensagens (*task, frame, ...*) devido a ativações internas e externas. O *offset* é definido como um atraso no tempo de liberação inicial das mensagens no barramento. Por conseguinte, isso aumenta significativamente a precisão do tempo de resposta calculado.

Para análise do cenário *SAE Benchmark*, o SymTA® foi configurado para gerar dados provenientes de 50 simulações (replicações) para alcançar um bom nível de aleatoriedade, propiciando diferentes configurações de inicialização do sistema e consequentemente possibilitando uma variabilidade nas colisões entre as mensagens.

O tempo de simulação de cada replicação é definido com base na mensagem que tem o maior período e menor prioridade, de forma que, essa mensagem seja pelo menos enviada três vezes a cada replicação. Afim de ilustrar, para o cenário em análise, a mensagem com essas características é a mensagem F1, que possui um período de 1000 ms. Logo, o tempo de simulação de cada replicação é 3 segundos. Como foram estabelecidas 50 replicações, esta mensagem possui em seu banco de dados aproximadamente 150 valores de tempo resposta simulados a serem analisados. Pela mesma razão, tomando-se como exemplo, a mensagem F16, de período corresponde de 5 ms, em 1 replicação, esta é enviada 600 vezes aproximadamente. Logo, o banco de dados simulado via SymTA® contém aproximadamente 30000 amostras de tempo de resposta para as 50 replicações.

Desta maneira, para cada mensagem foi gerado, via simulação, seu respectivo banco de dados a ser analisado, de forma que a quantidade de amostras do tempo de resposta de cada mensagem a ser avaliada é dependente do período desta.

Segundo Burns et al., (2002) as mensagens F8, F9 e F12 da Tabela 7 são particularmente vulneráveis, pois o valor do WCRT para essas mensagens é muito próximo de suas respectivas *deadlines*. No entanto, extraíndo as informações destas mensagens no banco de dados simulado pelo SymTA®, é possível notar o comportamento temporal destas mensagens, exibido nos histogramas abaixo, nas Figuras de 22 a 24. Consequentemente, observa-se também que avaliação de Burns et al., apresenta um certo pessimismo.

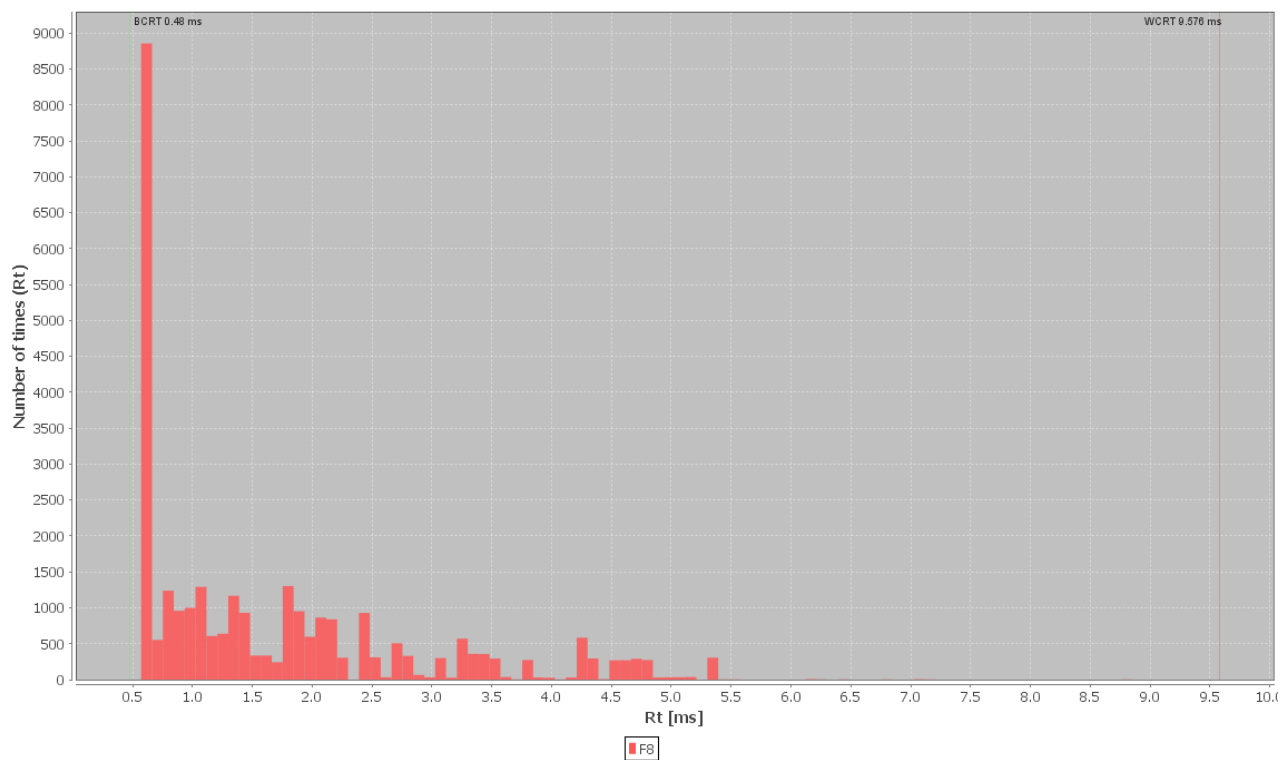


Figura 22 - Histograma mensagem F8.
 Fonte: Acervo do autor, com uso SymTA®.

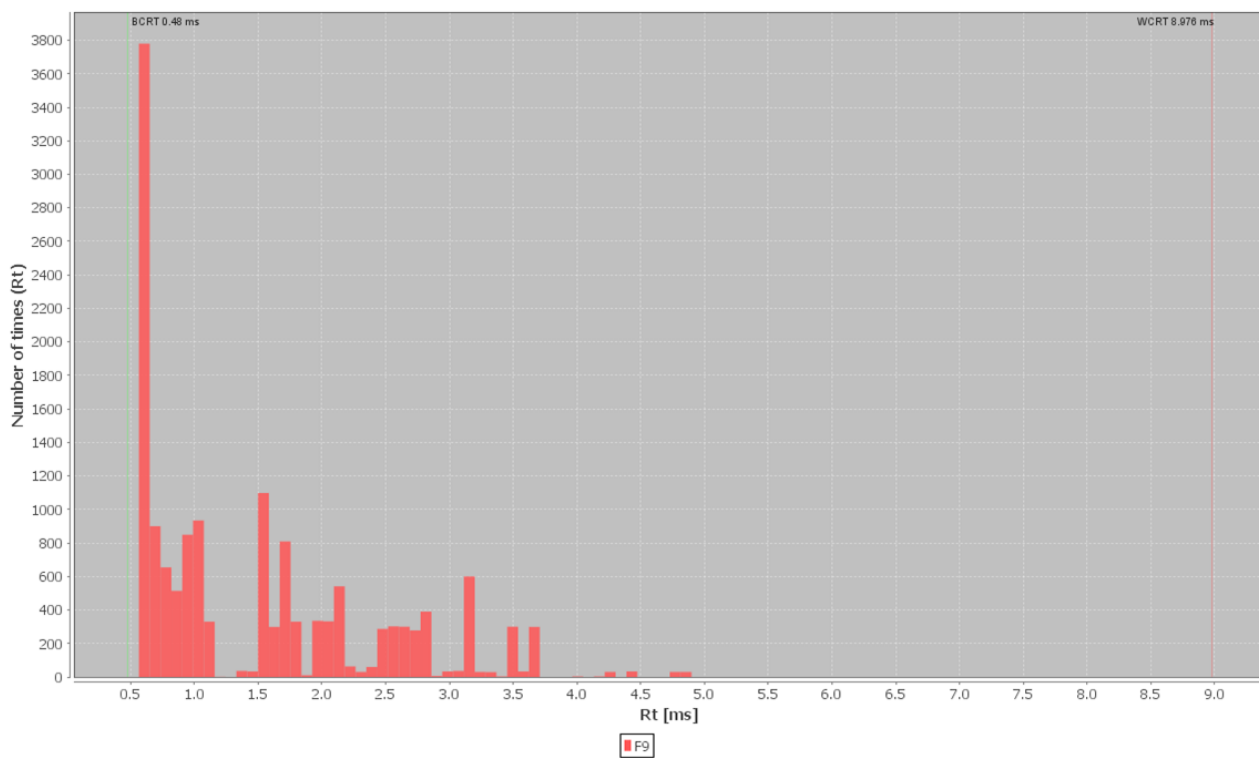


Figura 23 - Histograma mensagem F9.
 Fonte: Acervo do autor, com uso SymTA®.

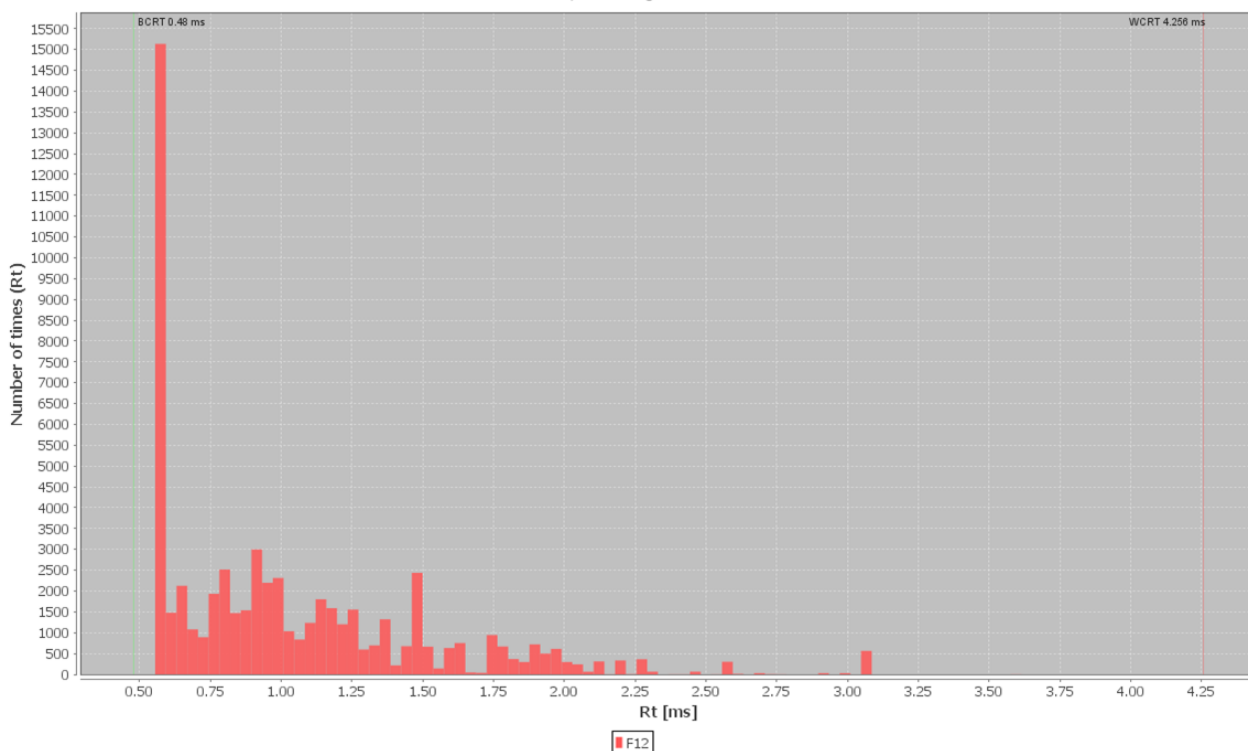


Figura 24 - Histograma mensagem F12.
 Fonte: Acervo do autor, com uso SymTA®.

Os histogramas presentes entre as Figuras 22 a 24, ilustram a distribuição do tempo de resposta para cada mensagem com base nas amostras geradas via simulação no SymTA®.

O eixo das abcissas do histograma representa o tempo de resposta, enquanto o eixo das ordenadas representa a quantidade de mensagens. A linha verde vertical a esquerda, representa o BCRT, enquanto, a linha vermelha vertical a direita, representa o WCRT. Por exemplo, examinando o histograma da Figura 24, constata-se que um pouco mais de 15000 mensagens são enviadas a um tempo de resposta de cerca de 0,55 ms, o que corresponde a aproximadamente 50% das amostras simuladas analisadas para esta mensagem. De forma análoga, para o tempo de resposta de aproximadamente 3,05 ms, obteve-se a recorrência de cerca de 500 mensagens, representando em torno de 1,6% das amostras simuladas do banco de dados.

Além disso, analisando a Figura 23, percebe-se que a incidência de mensagens com tempo de resposta superior a 4,0 ms é extremamente baixa e que os maiores valores encontrados estão abaixo de 5,0 ms, enquanto, o WCRT calcula um valor de

9,576 ms. Portanto, para esta situação, o WCRT apresentou um pessimismo de quase 50% para uma situação prática de funcionamento do sistema.

Observando os demais histogramas, percebe-se que os valores máximos dos tempos de resposta são sempre inferiores ao WCRT, e quanto mais próximo deste, menor a quantidade de mensagens. Como já supracitado, o WCRT é uma situação atípica, onde há a colisão de todas as mensagens no mesmo instante de tempo. Esta é uma possibilidade remota. Existem condições em que o tráfego de mensagens está mais denso, mas esses eventos ocorrem em bem menor escala e ainda são menores do que o WCRT, esses eventos são evidenciados pelos maiores valores de tempo de respostas encontrados nos histogramas das Figuras 22-24. O SymTA® utiliza-se desta aproximação, considerando o *System Distribution Analysis*, para se adequar a uma situação prática de funcionamento do sistema.

Ademais, nota-se que em momento algum as amostras simuladas atingiram o valor calculado pelo WCRT. A Tabela 13, a seguir, complementa as informações dos histogramas, exibindo os valores máximos e mínimos encontrados nas amostras analisadas para as mensagens F8, F9 e F12.

Tabela 13 – Pessimismo WCRT.

Mensagem	Rmim (ms)	Rmax (ms)	WCRT (ms)	T_i (ms)	D_i (ms)	Pessimismo (%)
F8	0,576	8,775	9,576	10	10	8,365
F9	0,576	4,870	8,976	10	10	45,744
F12	0,576	3,736	4,256	5	5	12,218

Fonte: Acervo do autor.

A Tabela 13, evidencia o pessimismo do WCRT para as mensagens analisadas. Esse pessimismo chegou a 45,74% para mensagem F9, pois o maior tempo de resposta foi de 4,870 ms para as 14993 amostras simuladas no banco de dados. Este pessimismo pode ser compreendido como a capacidade que o sistema deixa de usar, uma vez que o WCRT superestima os valores de latência das mensagens. Em outras palavras, na visão do WCRT, adicionar mais mensagens, principalmente com prioridades superiores a mensagem F9, é crítico para o cenário SAE *Benchmark*. Pois, isto pode resultar em WCRT maior que a *deadline* da mensagem. Por um outro lado, se eliminarmos o pessimismo do WCRT, mais mensagens com prioridades elevadas a F9 poderão ser adicionadas, respeitando sempre a *deadline* da mensagem. Isto otimiza o sistema, com relação a quantidade

de mensagens que este suporta, e reduz o custo de desenvolvimento de uma rede adicional, causada pela falta de capacidade de mensagens da rede projetada a priori com base no WCRT.

Portanto, eliminar esse pessimismo proveniente do WCRT é extremamente importante e traz os seguintes benefícios:

- Otimização do sistema e níveis de *bus load*, sendo assim, poder trabalhar em níveis elevados de *bus load* e ao mesmo tempo garantir a confiabilidade da rede.
- Aumento da capacidade de mensagens que a arquitetura suporta;
- Pode evitar o gasto de desenvolvimento de uma rede adicional para suportar a quantidade de centrais eletrônicas e mensagens que o sistema requer.
- Prolongar a vida útil do protocolo CAN, dentre outros.

Uma outra forma de avaliar o comportamento temporal de mensagens CAN são os gráficos de função de distribuição acumulada empírica, conforme exibido pela Figura 25, na qual retrata os valores de tempo de respostas obtidos via simulação para a mensagem F12, que possui período e *deadline* de 5 ms.

A Figura 25, é um exemplo do que se pode inferir analisando o gráfico de distribuição cumulativa empírica. Neste caso, nota-se, que aproximadamente 95% das mensagens são enviadas com o tempo de resposta igual ou inferior a 2 ms. O que corresponde, cerca de 25490 amostras, simuladas, do banco de dados, composto de 29988 valores de tempo de resposta para esta mensagem. Como a *deadline* da mensagem é 5 ms, significa que 95% das amostras analisadas apresentaram valores inferiores a 2 ms. O que comprova que a maior parte das mensagens atingem valores de latência bem menores que os calculados por meio do WCRT (4,256 ms) e *deadline* da mensagem (5 ms). Além do mais, de acordo com gráfico acima, 100% das mensagens são enviadas com o tempo de resposta inferior a 3,74 ms. Sendo este valor, 12,2% menor que o encontrado pelo WCRT e 25,2% menor que a *deadline* da mensagem.

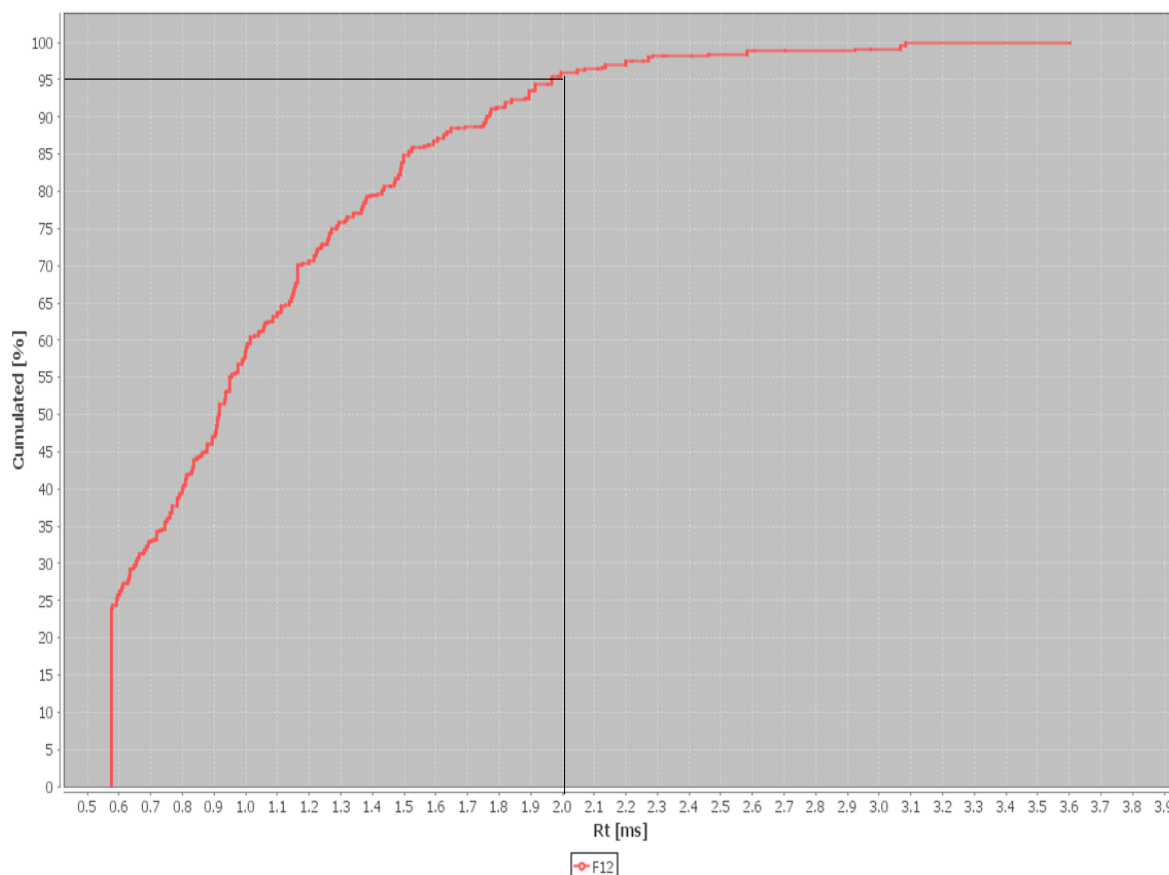


Figura 25 – Função de Distribuição Acumulada Empírica da mensagem F12.
 Fonte: Acervo do autor, com uso SymTA®.

Adicionalmente, para complementar o estudo e justificar a variabilidade do tempo de resposta das mensagens devido as diferentes condições de colisões, a verificação utilizando *boxplot* (gráfico de caixa) pode ser utilizada. O *boxplot* foi escolhido pois ele pode ser usado para avaliar uma distribuição empírica de dados e fornecer informações como: dispersão, assimetria e *outliers* (medidas discrepantes). O grande objetivo do *boxplot* é verificar a distribuição dos dados. De forma que a dispersão é representada pela amplitude de gráfico, assim quanto maior a amplitude maior será a variação dos dados. Além disso, o *boxplot* evidencia a assimetria dos dados, desta forma, se a mediana se aproxima do 1º quartil os dados são positivamente assimétricos e se a mediana se aproxima do 3º quartil os dados são negativamente assimétricos. Por fim, o *boxplot* pode ser usado para caracterizar a presença de *outliers*, o que indica um comportamento complexo de ser analisado em termos probabilísticos.

Com auxílio do MATLAB®, o *boxplot* para mensagem F12 foi obtido, na qual este ilustra a variação do tempo de resposta de F12, apresentado na Figura 26.

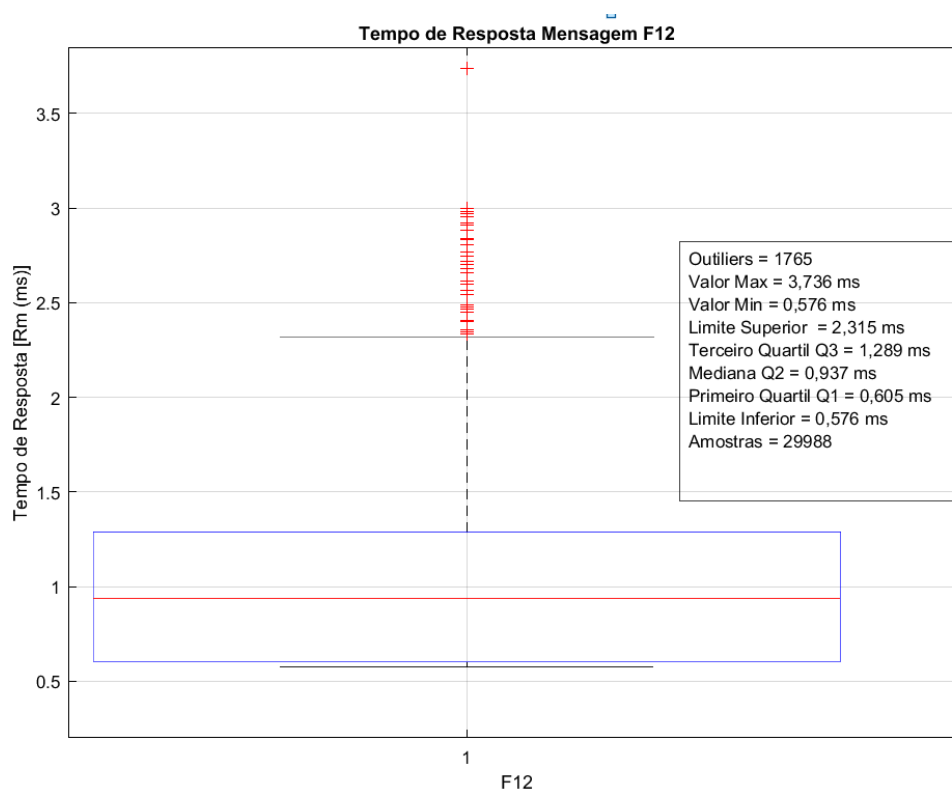


Figura 26 - *Boxplot* mensagem F12.
 Fonte: Acervo do autor, com uso Matlab®.

Tomando-se como exemplo a Figura 26, o intervalo interquartil é a diferença entre o 3º quartil e 1º quartil ($I.I.Q = Q3 - Q1$). Esta diferença para a mensagem acima é igual a 0,684 ms, percebe-se que 50% dos dados estão neste intervalo, pois abaixo do primeiro quartil temos 25% e acima do terceiro quartil temos 25% também. O limite superior é o produto de I.I.Q com o fator de 1,5 somado ao terceiro quartil, para este exemplo, resulta em 2,315 ms, qualquer valor acima deste é considerado um *outlier*. A barra horizontal em vermelho representa a mediana, ou segundo quartil, na qual 50% dos valores simulados do banco de dados estão acima ou abaixo dele. O traço horizontal em vermelho representa o valor mediano do banco de dados, ou segundo quartil, neste caso o valor mediano do tempo de resposta, para F12, este valor é de 0,937 ms, logo 50% das amostras simuladas encontrassem acima e abaixo deste. As cruces em vermelho representam os *outliers* presentes nas amostras analisadas.

Por fim, as estatísticas calculadas e as Figuras 25 e 26 mostram que a distribuição dos tempos de resposta para a mensagem F12 é fortemente assimétrica à direita, com possível comportamento multimodal, mas forte presença de *outliers*, indicando um comportamento muito complexo em termos probabilísticos.

Uma outra alternativa para analisar o perfil do tempo de resposta da mensagem F12, foi utilizando estimativas de intervalos de tolerância. Para esta mensagem, o intervalo de tolerância estimado com 95% de confiança foi [0,576 ms; 2,616 ms] com base nas amostras simuladas pelas 50 replicações. De forma geral, esta estimativa implica que 99% dos tempos de resposta gerados nas mesmas condições estarão entre 0,576 ms e 2,616 ms. Vale ressaltar que o tempo de resposta máximo encontrado nos dados simulados para esta mensagem foi de 3,736 ms, que está acima do limite superior do intervalo estimado, evidenciando assim que a mensagem F12 produziu um padrão de observações extremamente difícil de ser captado com um modelo probabilístico.

Mesmo assim, a possibilidade de se ajustar as amostras dos tempos de resposta a uma distribuição de probabilidade foi investigada.

O primeiro passo antes de se tentar ajustar uma ou mais distribuições de probabilidade a um conjunto de dados é determinar as distribuições candidatas dentre diversas possibilidades.

Essa escolha pode ser guiada pelo conhecimento do processo estocástico que rege a variável modelada ou, na ausência de conhecimento sobre o processo subjacente, pela observação de sua distribuição empírica.

Entretanto, como não se tinha conhecimento suficiente sobre o processo estocástico que produz os tempos de resposta de uma mensagem CAN. Optou-se por utilizar o gráfico de Cullen & Frey (1999), implementado na linguagem R, para selecionar possíveis distribuições candidatas. O resultado é exibido, para a mensagem F12, na Figura 27.

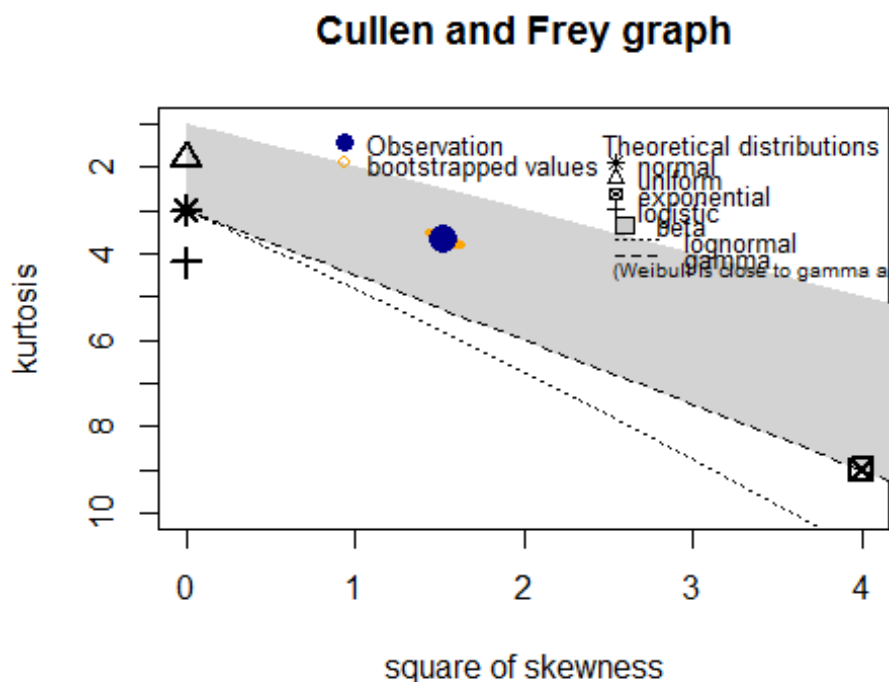


Figura 27 - Gráfico de Cullen & Frey para a mensagem F12.
 Fonte: Acervo do autor, com uso RStudio®.

Com base na teoria de Cullen e Frey (1999) e na posição da observação, em azul na Figura 27, para mensagem F12, observa-se que as possíveis distribuições candidatas são: Weibull, Gamma, Lognormal e Beta, pois os valores das observações, em azul, estão na região da distribuição Beta (em cinza) e acima das retas características das distribuições Weibull, Gamma e Lognormal, o que por sua vez caracteriza que estas distribuições podem ser utilizadas para analisar os dados. As outras demais distribuições seriam indicadas se as observações estivessem concentradas em cima de sua respectiva região.

Dentre essas possibilidades, para esta monografia, as distribuições avaliadas foram Weibull e Gamma, conforme exibido nas Figuras 28 e 29.

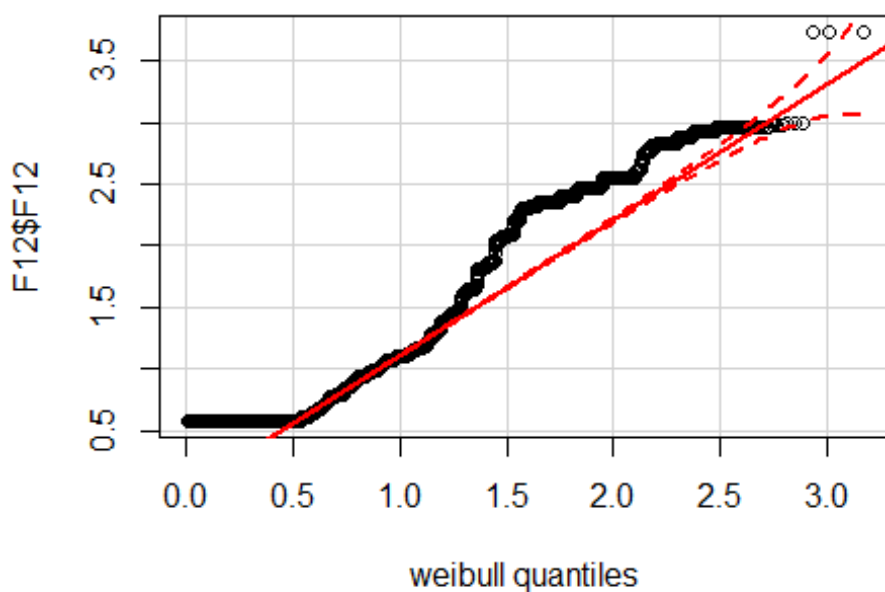


Figura 28 - Gráfico Quartil-Quartil do ajuste da distribuição Weibull.
Fonte: Acervo do autor, com uso RStudio®.

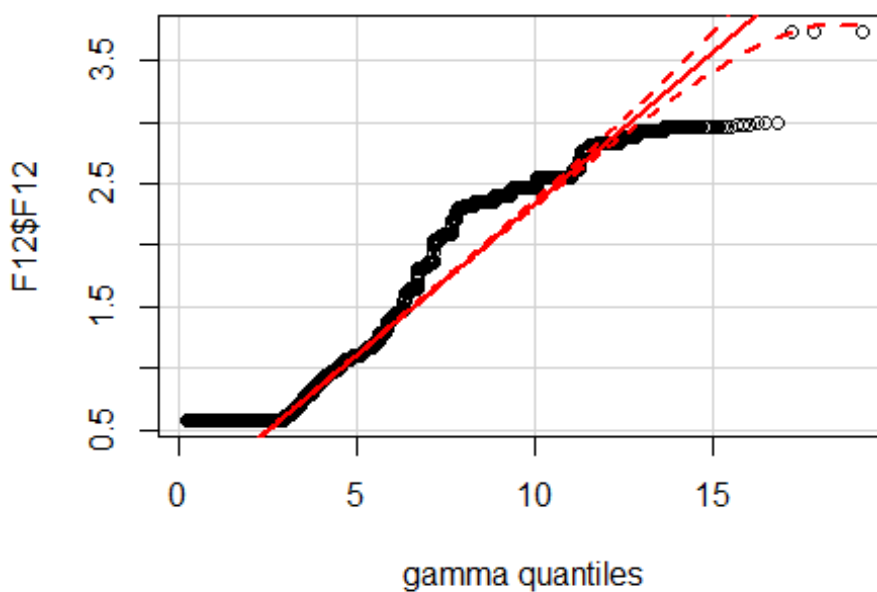


Figura 29 - Gráfico Quartil-Quartil do ajuste da distribuição Gamma.
Fonte: Acervo do autor, com uso RStudio®.

Analisando as Figuras 28 e 29, percebe-se que nenhuma das duas distribuições apresenta um bom ajuste, pois os pontos, que representam os dados, deveriam concentrar-se dentro do intervalo de confiança representado pelas linhas vermelhas.

Portanto, pode-se concluir que nenhuma das distribuições captam adequadamente o padrão complexo, possivelmente multimodal e forte presença de *outliers* que caracterizam o comportamento do tempo de resposta da mensagem F12.

Desta maneira, pelas razões supracitadas, o *boxplot* é uma boa opção para avaliar os dados, uma vez que ele indica a dispersão, simetria e presença de *outliers* entre os dados analisados. Desta forma, ao estender a avaliação para as demais mensagens da arquitetura em estudo, obtém-se os resultados apresentados na Figura 30.

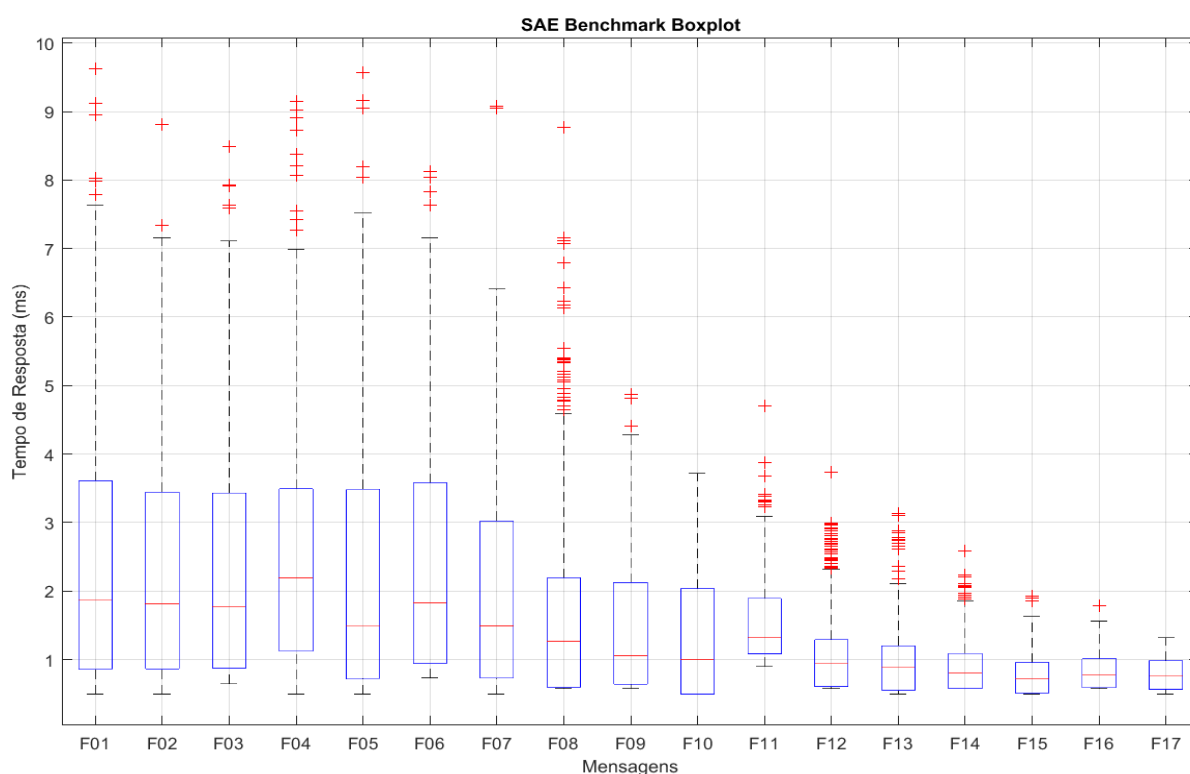


Figura 30 - *Boxplot* da arquitetura avaliada.
Fonte: Acervo do autor, com uso Matlab®.

É evidente que mesmo havendo presença de *outliers*, principalmente nas mensagens de menor prioridade, o valor máximo atingido pelo tempo de resposta para a mensagem ainda é menor que o previsto pelo WCRT, analisando as amostras simuladas. Além disso, devido a esta grande presença de *outliers*, adequar as

amostras das latências em uma distribuição probabilística se torna extremamente complexa. As amostras simuladas do banco de dados não apresentam uma distribuição bem definida, como ilustrado na Figura 30.

A Tabela 14, faz uma comparação entre o valor máximo do tempo de reposta obtido pelas amostras simuladas do banco de dados com os valores calculados para WCRT.

Tabela 14 - Grau de pessimismo para as mensagens da arquitetura.

Mensagem	WCRT	Rmax	Pessimismo (%)
F17	1,416 ms	1,317 ms	6,99
F16	2,016 ms	1,787 ms	11,36
F15	2,536 ms	1,924 ms	24,13
F14	3,136 ms	2,590 ms	17,41
F13	3,656 ms	3,131 ms	14,36
F12	4,256 ms	3,736 ms	12,22
F11	5,016 ms	4,709 ms	6,12
F10	8,376 ms	3,725 ms	55,53
F9	8,976 ms	4,870 ms	45,74
F8	9,576 ms	8,775 ms	8,36
F7	10,096 ms	9,073 ms	10,13
F6	19,616 ms	8,127 ms	58,57
F5	20,136 ms	9,574 ms	52,45
F4	20,136 ms	9,151 ms	54,55
F3	28,976 ms	8,490 ms	70,70
F2	29,946 ms	8,816 ms	70,56
F1	29,946 ms	9,624 ms	67,86

Fonte: Acervo do autor.

Percebe-se que para as mensagens de menor prioridade, tal pessimismo do WCRT é bem elevado, chegando a 70,7 %, especificamente para a mensagem F3. Para elucidar a diferença, entre os valores obtidos no banco de dados simulado e os valores obtidos pelo WCRT, a Figura 31, exibe em vermelho e azul, respectivamente, as curvas para WCRT e tempo de resposta máximo para as mensagens avaliadas.

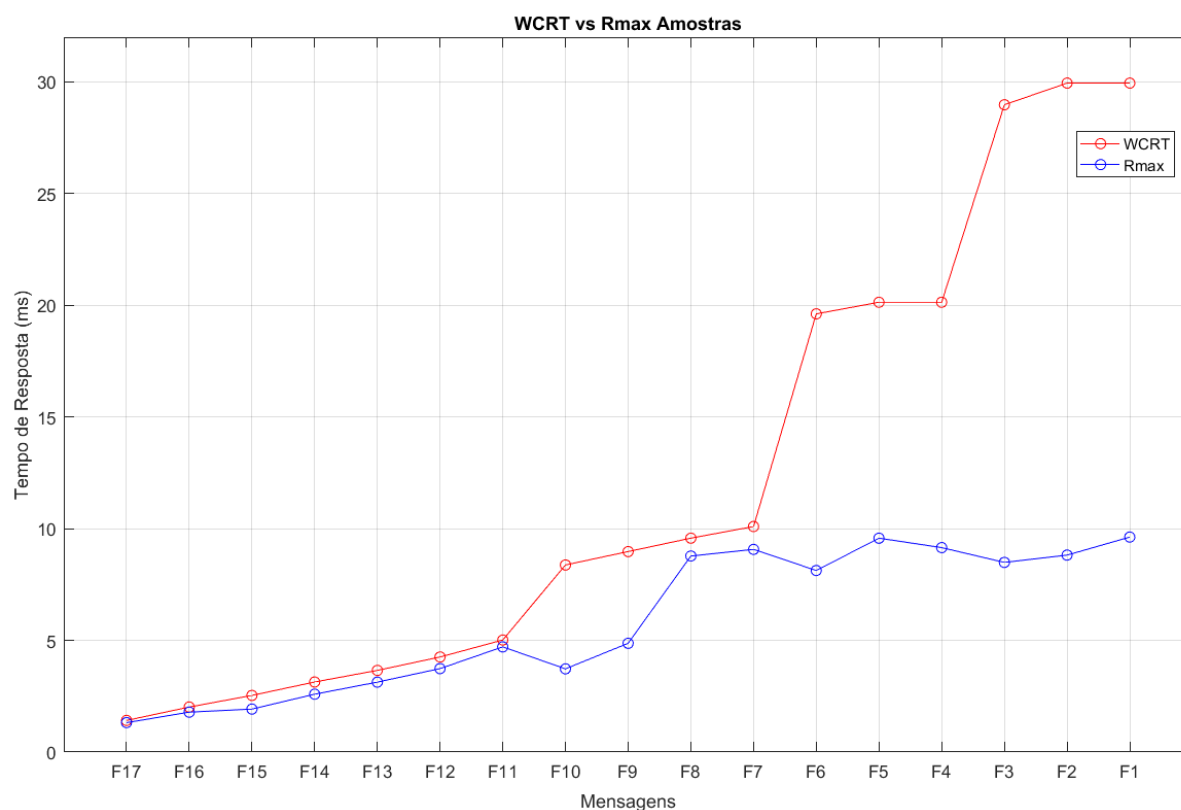


Figura 31 – Perfil WCRT e Rmax das amostras simuladas via SymTA®.

Fonte: Acervo do autor, com uso Matlab®.

Logo percebe-se que o WCRT tem uma tendência em superestimar os valores de tempo de resposta para as mensagens de menor prioridade.

Em síntese, com base nas amostras simuladas nota-se que para as mensagens de maior prioridade o WCRT apresenta uma boa aproximação para o tempo de resposta da mensagem. Todavia, se tratando de mensagens prioridades intermediárias e baixas, a aproximação do WCRT não é tão efetiva. Ao considerar uma arquitetura veicular real, esse pessimismo na análise pode gerar custos adicionais à montadora, pois ele limita a quantidade de centrais e mensagens que trafegam na rede. Logo, dependendo do projeto, para suportar a quantidade de centrais e mensagens requisitadas seria necessário a implementação de redes adicionais, elevando assim o custo do projeto.

4.4 Avaliação do *Bus Load*

Como caracterizados por Burns et al., (2002) a taxa de utilização do barramento para o conjunto de mensagens do SAE *Benchmark* é em torno de 85%. As Figuras 32 e 33, apresentam os resultados provenientes da simulação do cenário SAE *Benchmark*, exibido nas Tabelas 6 e 7, para *bus load* por meio do software SymTA®.

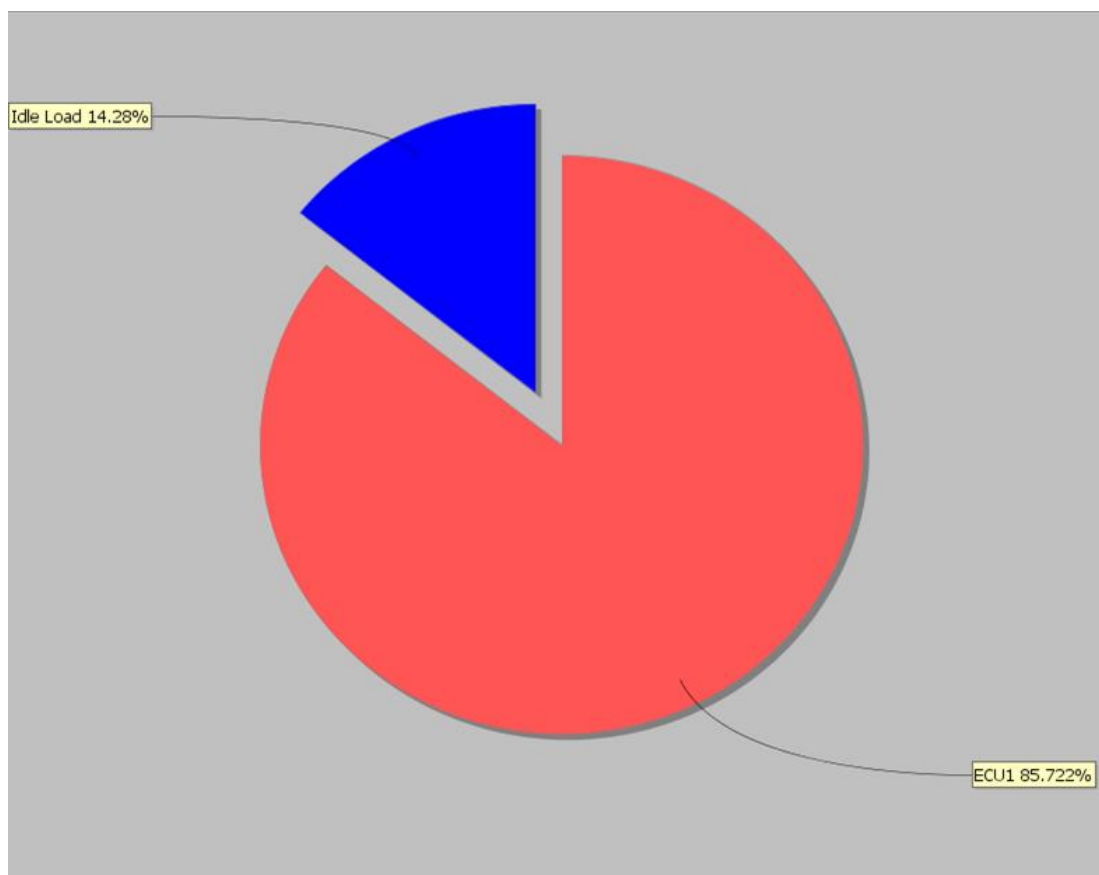


Figura 32 - *Bus load* e contribuição de cada ECU na rede.
Fonte: Acervo do autor, com uso SymTA®.

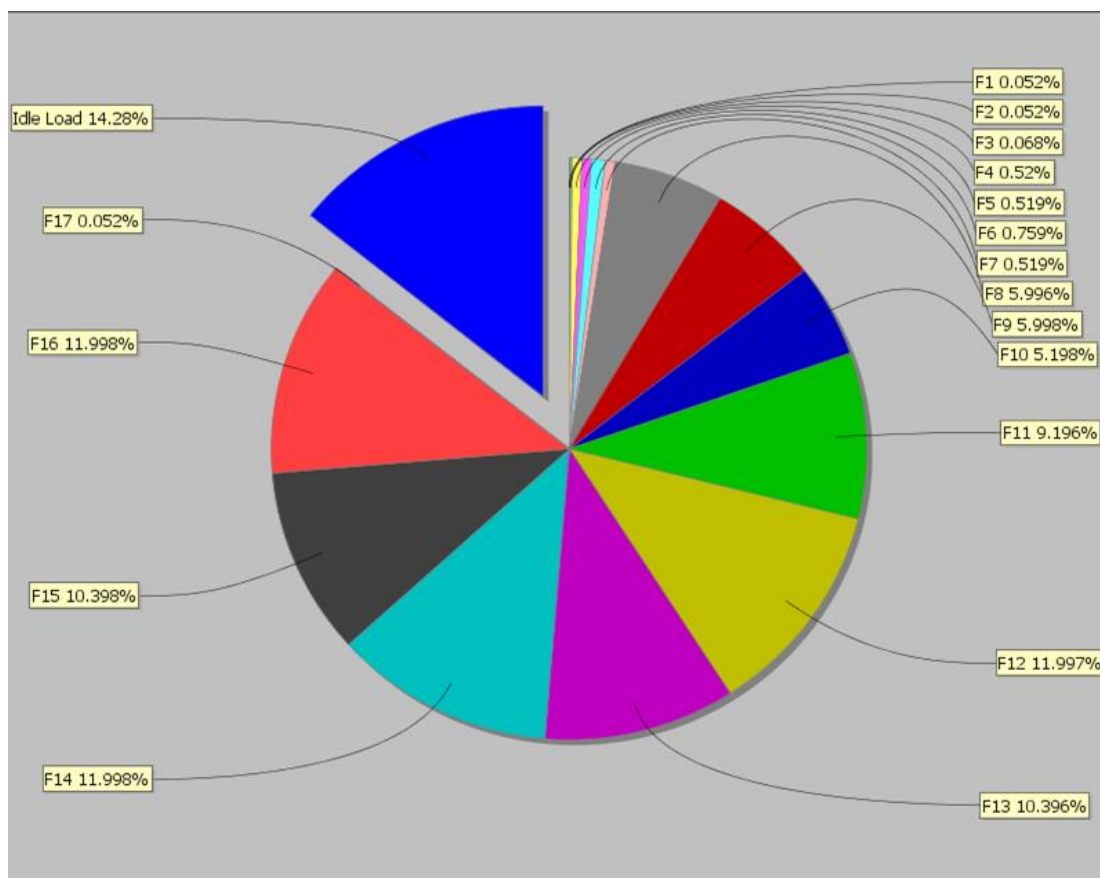


Figura 33 - Contribuição de cada mensagem para o valor de *bus load*.

Fonte: Acervo do autor, com uso SymTA®.

A Figura 32, exibe o valor de *bus load* fornecido com base no banco de dados simulado via SymTA®, o valor encontrado foi de 85,72%, validando assim os resultados da simulação com o da literatura mencionada anteriormente neste texto. Além do mais, utilizando a Figura 33 é possível ver a contribuição de cada mensagem para a carga no barramento. O *bus load* de cada mensagem depende do tamanho e do período da mensagem. Como *bus load* é mensurado em intervalos, se uma mensagem tem um período pequeno, significa que ela é enviada mais vezes no mesmo intervalo de tempo, maior frequência. Logo, ela ocupa uma carga maior do barramento. Assim, analisando a Figura 33, as mensagens que mais contribuem para o valor elevado de *bus load* são as mensagens de periodicidade alta, no caso, 5 ms.

Importante também destacar, que com as amostras simuladas por este trabalho é possível avaliar o comportamento temporal para o *bus load*. A Figura 34, exibe como é o perfil do *bus load* ao longo do tempo para uma replicação de 3 s.

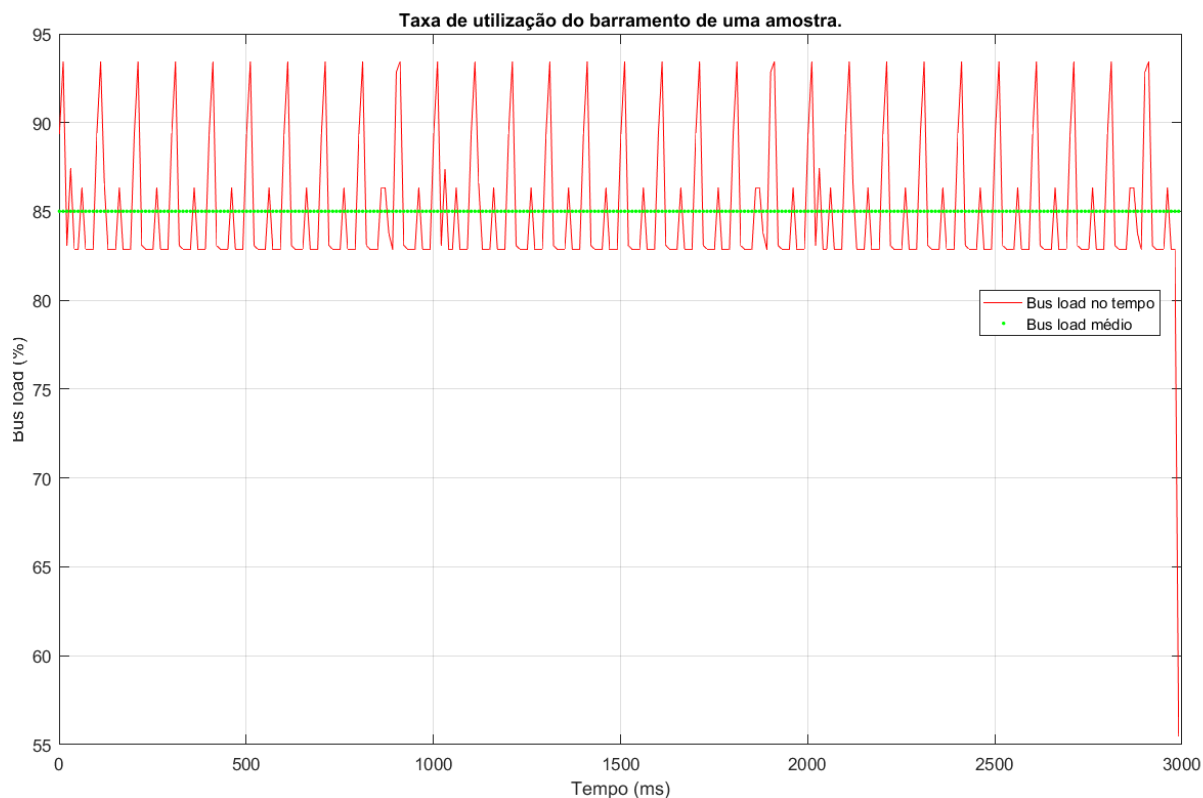


Figura 34 - Perfil de *bus load* para uma amostra.

Fonte: Acervo do autor, com uso Matlab®.

A Figura 34, mostra a variação do *bus load* de uma replicação para o tempo de simulação de 3 s, em vermelho. No qual, a reta, em verde, representa o valor médio do *bus load*, em torno de 85%. Entretanto, ao longo do tempo a carga do barramento pode sofrer picos de variação que vão em torno 94%, quando a densidade de mensagens enfileiradas está elevada, a cerca 83,5%, quando a densidade de mensagens disputando o barramento é menor.

O *boxplot* da Figura 35, mostra que a distribuição razoavelmente simétrica para o *bus load*.

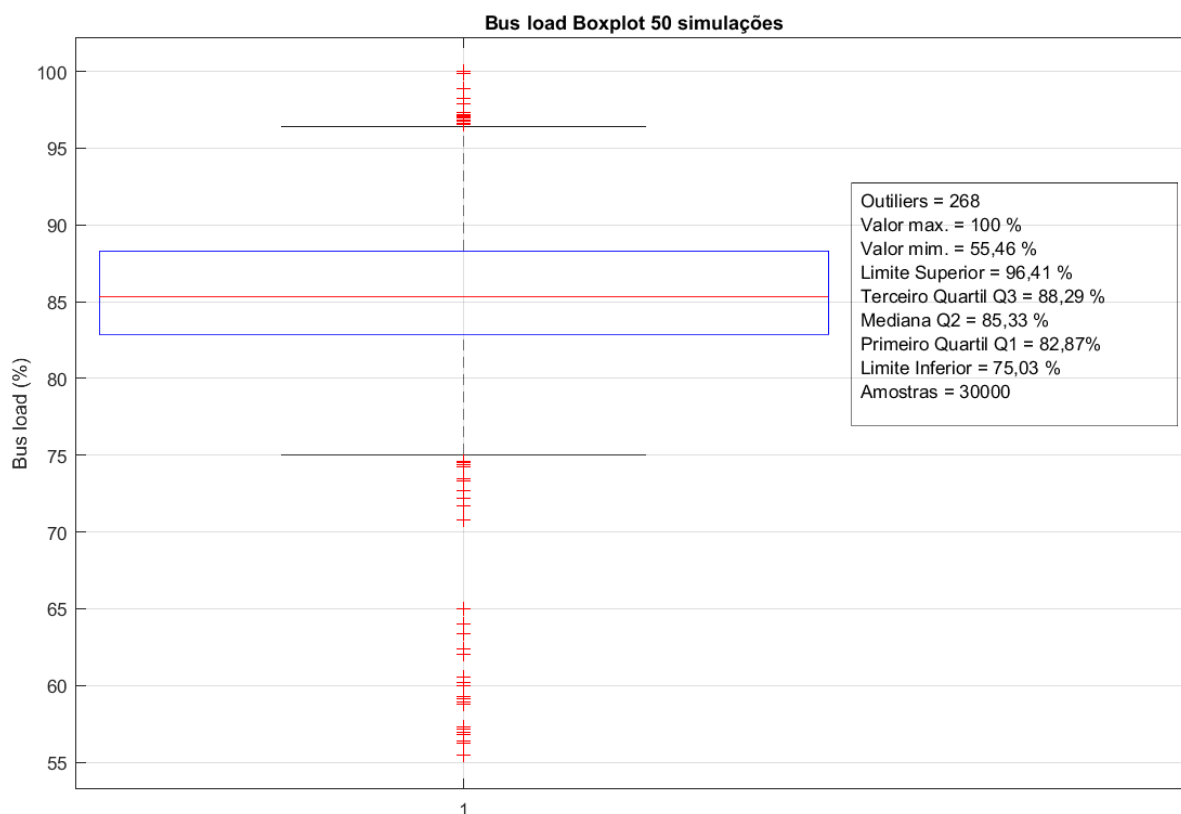


Figura 35 - *Boxplot Bus Load*.
 Fonte: Acervo do autor, com uso Matlab®.

Além disso, das 30000 amostras analisadas 268 estiveram abaixo ou acima dos limites superiores e inferiores, sendo, portanto, consideradas como *outliers*, ou seja, um ponto que está acima do limite superior ou abaixo do limite inferior. Pode ser descrito como ponto distante ou ponto fora da curva.

Uma outra observação que pode ser feita é com relação a influência do período das mensagens na computação do *bus load*. Para apresentar tal relação, serão destacadas as mensagens de um *byte* de dados da presente configuração, uma vez que para os valores de período de 5 ms, 10 ms, 100 ms e 1000 ms pelo menos uma mensagem com essa característica está presente, como mostra a Tabela 15.

Tabela 15 - Relação entre período da mensagem e *bus load*.

Mensagem	Bytes	Período	Bus Load (%)
F17	1	1000	0,052
F15	1	5	10,39
F13	1	5	10,39
F10	1	10	5,198
F7	1	100	0,519
F5	1	100	0,519
F4	1	100	0,520
F2	1	1000	0,052
F1	1	1000	0,052

Fonte: Acervo do autor.

Ao observar a Tabela 15, percebe-se que a relação entre a utilização do barramento pela mensagem e seu período é inversamente proporcional. Por conseguinte, se período da mensagem for reduzido em dez vezes a carga que esta mensagem ocupa no barramento aumenta-se dez vezes. Logo, a seguinte relação, explicitada pela Equação 20, pode ser utilizada:

$$B' = B \times \frac{T}{T'} \quad (20)$$

Sendo,

- B' – *Bus load* novo (%);
- T – Período antigo (ms);
- T' – Período novo (ms);

Para esta relação ser verdadeira, a quantidade de *bytes* do campo de dados deve ser a mesma. No mais, esta é uma equação que pode ser usada para calcular o impacto no *bus load* em diversos cenários de periodicidade. Se forem conhecidos o período, tamanho e *bus load* da mensagem.

Portanto, a nível de mensagem, as características que impactam no *bus load* são o tamanho, quantidade e periodicidade de mensagens. É evidente que quanto maior a mensagem, mais tempo esta leva para ser transmitida, logo, maior sua carga no barramento. Por exemplo, o SymTA®, para mensurar o *bus load*, avalia o uso do barramento em intervalos de 15 ms, assim, se uma mensagem consome 2 ms para

ser transmitida, significa que ela ocupa aproximadamente 13,3% do barramento. Conseqüentemente, uma mensagem com uma maior quantidade de dados ocuparia mais o barramento.

Adicionalmente, o período da mensagem tem um impacto extremamente relevante na parcela que a mensagem ocupa no barramento, como mostrado na Equação 20. O *bus load* da mensagem é inversamente proporcional ao período da mensagem. Portanto, aumentar o período de uma mensagem em dez vezes, pode significar que está ocupará cerca de dez vezes menos o barramento.

Por fim, na concepção da arquitetura veicular deve-se avaliar a necessidade de a periodicidade da mensagem ser elevada, a fim de otimizar a quantidade de mensagens que a rede pode suportar e valores de *bus load*.

Além disso, eliminando o pessimismo do WCRT mais mensagens podem ser adicionadas à rede, conseqüentemente o nível de *bus load* pode aumentar pois ele depende: da quantidade, periodicidade e tamanho das mensagens. Não obstante, uma vez que a *deadline* das mensagens são respeitadas a confiabilidade da rede pode ser garantida.

Adicionalmente, se o barramento apresenta um *bus load* elevado, o valor para o WCRT das mensagens CAN tende a aumentar, principalmente para as mensagens de menor prioridade

4.5 Publicações Resultantes Desta Monografia

As publicações a seguir são resultados desta pesquisa:

- CARVALHO, S. M. T.; CAMPOS, G. L. “Transmissão de mensagens e gerenciamento de erros em uma rede CAN automotiva”. Artigo submetido na Revista ForScience.
- CARVALHO, S. M. T.; CAMPOS, G. L. “Worst Case Response Time Approach Evaluation for Computing CAN Messages Response Times in an Automotive Network”. Artigo submetido COPEP 2017, Brazilian Power Electronics Conference, Juiz de Fora, Minas Gerais, Brasil.

- ADAMSHUK, R. ; ANDRADE, DAVID ; NEME, J. H. ; MARGRAF, E. ; OKIDA, S.; TUSSET, A. M. ; SANTOS, M. M. D. ; AMARAL, R.; VENTURA, A. C.; CARVALHO, S. “On the Applicability of Inverse Perspective Mapping for the Forward Distance Estimation based on the HSV Colormap”. In: IEEE - International Conference on Industrial Technology (ICIT 2017), 2017, Toronto - ON. IEEE - International Conference on Industrial Technology (ICIT 2017), 2017. Disponível em: <http://ieeexplore.ieee.org/document/7915504/>
- ADAMSHUK, R. ; ANDRADE, DAVID ; NEME, J. H. ; MARGRAF, E. ; OKIDA, Sergio ; TUSSET, A. M. ; SANTOS, M. M. D. ; AMARAL, Rodrigo ; VENTURA, A. C. ; CARVALHO, S. “A Novel Strategy for Road Lane Detection and Tracking Based on Vehicle’s Forward Monocular Camera”. Artigo submetido ao IEEE ITS Transactions.

5 CONCLUSÃO

O protocolo de comunicação CAN tem estado em evidência no setor automotivo ao longo dos últimos anos. Para prolongar a vida útil do protocolo e garantir a confiabilidade e a robustez da rede, as montadoras de automóveis de modo em geral optam em manter o *bus load* em valores não superiores a 60%. O principal motivo para se manter o percentual de utilização do barramento a níveis mais amenos é mitigar a disputa pelo acesso ao barramento de mensagens menos prioritárias. Conseqüentemente, evitar com que essas mensagens sejam atrasadas por mensagens mais prioritárias de forma que estas possam perder a seu *deadline*, comprometendo a robustez e segurança da rede veicular.

Para evitar esta situação indesejada, o WCRT é utilizado para calcular o tempo de resposta das mensagens no pior caso. De forma que, se o valor obtido pelo WCRT for menor que o *deadline* da mensagem é possível garantir que a mensagem enviada pela central transmissora será recebida pela central receptora em condições de funcionamento normal da rede, ou seja, sem situações de erro.

Não obstante, os resultados obtidos por esta monografia evidenciam que as aproximações do WCRT apresentam uma análise pessimista, ou seja, ela superestima os valores do tempo de resposta da mensagem, principalmente as mensagens de menor prioridade. Esse pessimismo é transferido para o *bus load*, na qual este é principal parâmetro na definição da arquitetura veicular. Em vista disto, para manter a carga do barramento a níveis menos elevados, devido à grande quantidade de centrais eletrônicas, impulsionadas pela evolução tecnológica do setor, os fabricantes de automóveis utilizam mais de uma rede CAN para comportar todas as centrais e quantidade de mensagens gerando um custo adicional.

Além disso, os resultados mostraram que o comportamento probabilístico dos tempos de respostas das mensagens críticas analisadas é muito complexo, não sendo possível obter um bom ajuste em modelos conhecidos. Portanto, as distribuições dos tempos de resposta das mensagens analisadas não se mostraram facilmente modeláveis por distribuições de probabilidade conhecidas. O que os resultados obtidos sugerem, é que, possivelmente, uma mistura de distribuições associadas a um maior conhecimento de engenharia envolvida pode gerar um modelo complementar relevante para a análise do WCRT, principalmente nas fases iniciais do

delineamento da arquitetura veicular, conforme demonstrado por Zent et. al. (2009, 2010)

O cenário avaliado, com o *bus load* de aproximadamente 85%, era considerado crítico do ponto de vista do WCRT. Todavia, para a mensagem F9 (tabela 12), considerada crítica, seu pessimismo chegou a 45,7%, ou seja, o seu tempo de resposta máximo baseado nas amostras produzidas pelo SymTA chegou a ser 45,7% menor do que tempo calculado pelo WCRT. Isto implica que novas mensagens/centrais eletrônicas poderiam ser adicionadas à arquitetura, o que ocasionaria um aumento do *bus load*, mas de forma consciente. Isto é, sem comprometer confiabilidade e robustez da rede.

Pode-se inferir então, que um *bus load* elevado necessariamente não significa que a rede CAN está comprometida. O que de fato deve ser levado em consideração na escolha de uma arquitetura veicular é a relação entre o tempo de resposta da mensagem e seu *deadline*. É evidente que com o aumento do *bus load*, manter o tempo de resposta da mensagem inferior ao seu *deadline* torna-se mais difícil, mas se esta condição é respeitada é possível assegurar a confiabilidade da rede.

Desta forma, uma análise profunda do tempo de resposta das mensagens CAN pode prover diminuição da complexidade do sistema, devido a redução da quantidade de redes CAN presentes na arquitetura, redução de custo e manutenção, dentre outras vantagens.

Além disso, pensando na otimização do *bus load*, foi avaliado as características das mensagens que impactam neste parâmetro, cujo principal objetivo é ajudar os engenheiros na escolha das características da mensagem no momento da concepção da arquitetura. Dentre as características das mensagens foram avaliadas: prioridade, periodicidade e tamanho da mensagem. Sendo que a única destas que não impacta diretamente no *bus load* é a prioridade. No entanto, quando a tratativa é o tempo de resposta da mensagem esta é extremamente importante. Com relação ao tamanho da mensagem, quanto maior for este valor, mais a mensagem irá ocupar o barramento. Por fim, o parâmetro crítico para o *bus load* é o período da mensagem, de forma que se o período da mensagem for reduzindo em dez vezes, por exemplo, significa que a contribuição desta mensagem para o *bus load* será dez vezes maior.

Uma outra contribuição deste trabalho foi a limitação encontrada na Equação 16, proposta por Burns et al. (2002), para o cálculo de C_m . A proposta é utilizar a Equação 18, eliminando pessimismo extra devido ao processo de bit *stuffing*.

Consequentemente, utilizando a Equação 18 foram encontrados os mesmos valores presentes na literatura e simulação em *software*.

Em suma, este trabalho avaliou por completo o método clássico para o WCRT evidenciando as suas limitações e propondo melhorias para evitar sua análise pessimista. Por fim, com a análise feita neste trabalho é possível o engenheiro responsável por desenvolver a arquitetura avaliar os principais pontos que podem comprometer a confiabilidade da rede, sendo eles: tempo de resposta, *deadline* e características da mensagem, afim de propor uma otimização da rede CAN em valores de carga no barramento elevados, reduzido custos de uma nova rede adicional para trabalhar em níveis de *bus load* menos elevados, prologando assim a vida útil do protocolo CAN no setor automotivo.

5.1 Trabalhos Futuros

O tema que o tema da presente monografia tem sido largamente investigado no setor automobilístico, várias propostas de continuidade para trabalhos futuros podem ser feitas, dentre as quais podem ser citadas.

- Publicar artigos científicos com as avaliações do presente trabalho;
- Complementar o estudo com a adição de mensagens periódicas;
- Analisar o tempo de resposta da arquitetura em diversas condições de sobrecarga, para determinar o nível ótimo entre quantidade de mensagens e *bus load*.
- Estender o estudo para situações de falhas, na qual mensagens devem ser retransmitidas.
- Desenvolver *softwares* para calcular o WCRT em outras plataformas para uso didático;
- Ampliar a análise exploratória dos dados usando diferentes métodos estatísticos e probabilístico para calcular o tempo de resposta de mensagens em redes CAN;
- Analisar métricas alternativas ou complementares ao *bus load* para caracterização da rede veicular.

- Utilizar uma mistura de distribuições de probabilidade para gerar um modelo complementar relevante para a análise do WCRT e tempos de resposta de mensagens CAN

REFERÊNCIAS BIBLIOGRÁFICAS

- ASSIS, A. C. **Implementação e avaliação do protocolo FTT-CAN sobre o sistema AUTOSAR**. 2011.
- BARBOSA, L. R. G. Rede CAN. **Escola de Engenharia da UFMG**. Universidade Federal de Minas Gerais, Belo Horizonte, 2003.
- BISHOP, Richard. **Intelligent vehicle technology and trends**. 2005.
- BOSCH. "**CAN Specification version 2.0**". Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart, 1991.
- BURNS, A.; BROSTER, I; RODRÍGUEZ-NAVAS, G. Probabilistic analysis of CAN with faults. In: **Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE**. IEEE, 2002. p. 269-278.
- CASPARSSON, Lennart et al. Volcano-a revolution in on-board communications. **Volvo technology report**, v. 1, p. 9-19, 1998.
- COOK, J. A.; FREUDENBERG, J. S. **Controller Area Network (CAN)**. 2007.
- CORRIGAN, S. **Introduction to the Controller Area Network (CAN)**. Texas Instruments, Application Report, 2002.
- CULLEN, A.C; FREY, H.C. **Probabilistic Techniques in Exposure Assessment**. 1st edition. Plenum Publishing Co, 1999.
- 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. Understanding and using the Controller Area network" inst. eecs. berkeley. edu/~ ee249/fa08. **Lectures/handout_canbus2. pdf**, 2008.
- FRESCALE. **CAN Bosch Controller Area Network (CAN) Version 2.0 Protocol**. 1998. Disponível em: <http://www.nxp.com/assets/documents/data/en/reference-manuals/BCANPSV2.pdf> Acesso em: 04 de Mar. 2017.
- GUIMARÃES, A. A.; SARAIVA, A.M. **O Protocolo CAN: Entendendo e Implementando uma Rede de Comunicação Serial de Dados baseada no Barramento "Controller Area Network"**. Artigo SAE 2002-01-3569, 2002. In: CONGRESSO SAE BRASIL 2002, 11., São Paulo, 2002.
- GODOY, E. P. **Desenvolvimento de uma Ferramenta de Análise de Desempenho de Redes CAN para Aplicações em Sistemas Agrícolas**. Dissertação de Mestrado apresentada à Escola de Engenharia de São Carlos, Universidade de São Paulo, 2007.

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

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 11898-2**, Road vehicles–Controller area network (CAN)–Part 2: High-speed medium access unit, 2003b.

ISO. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 11898-3**, Road Vehicles – Controller Area Network (CAN) – Part 3: Low – speed, fault-tolerant, medium-depedent interface. 2006.

JOHANSSON, K. H.; TÖRNGREN, M.; NIELSEN, L. Vehicle applications of controller area network. In: **Handbook of networked and embedded control systems**. Birkhäuser Boston, 2005. p. 741-765.

JOSEPH, M.; PANDYA, P. Finding response times in a real-time system. **The Computer Journal**, v. 29, n. 5, p. 390-395, 1986.

KVASER. **CAN Protocol Tutorial**, 2017. Disponível em: <https://www.kvaser.com/can-protocol-tutorial/> Acesso em: 04 de Mar. 2017.

LUGLI, A. B.; SANTOS, M. M. D. **Sistemas FIELDBUS para automação industrial: DeviceNet, CANopen, SDS e Ethernet**. Tatuapé: Erica, 2009

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

MUBEEN, S.; MÄKI-TURJA, J.; SJÖDIN, M. **Response-Time Analysis of Mixed-Type Controller Area Network (CAN) Messages**. 2011.

PUNNEKKAT, S; HANSSON, H.; NORSTROM, Christer. Response time analysis under errors for CAN. In: **Real-Time Technology and Applications Symposium, 2000. RTAS 2000. Proceedings. Sixth IEEE**. IEEE, 2000. p. 258-265.

SANTOS, M. M.; VASQUES, F. Performance Analysis of Networked Control Systems over CAN and Token Passing Networks. In: **VI IEEE International Conference on Industrial Applications (INDUSCON-04)**. 2004

SOUZA, V. A. Introdução ao CAN. **Cerne Tecnologia**. Disponível em http://www.artigos.com/components/com_mtree/attachment.php, 2011.

SYMTA. **Analysis Introduction and Theory**. User Manual SymTA/S Tool Suite. Version 3.4.0. 2013.

SYMTA. **CAN**. User Manual SymTA/S Tool Suite. Version 3.1.0. 2012.

SYMTAVISION. **SymTAS & TraceAnalyzer**. Disponível em: <http://www.symtavision.com/products/symtas-traceanalyzer/>. Acesso em 21 Abr. 2017.

TANENBAUM, A. S. **Redes de Computadores**. 4. ed. Amsterdam: Campus, 2010. 632 p.

TINDELL, K.; BURNS, A. Guaranteeing message latencies on control area network (CAN). In: **Proceedings of the 1st International CAN Conference**. Citeseer, 1994a.

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

TINDELL, K.; HANSSMON, H.; WELLINGS, A. J. Analysing Real-Time Communications: Controller Area Network (CAN). In: **RTSS**. 1994b. p. 259-263.

VECTOR. **CAN Protocol Reference Chart**. Disponível em: http://elearning.vector.com/vl_can_introduction_en.html. Acesso em 04 Mar. 2017.

ZENG, H.; DI NATALE, M.; GIUSTO, P.; SANGIOVANNI-VINCENELLI, A. Stochastic Analysis of CAN-Based Real-Time Automotive Systems. **IEEE Transactions on Industrial Informatics**, 5(4):p. 388-401, 2009.

ZENG, Haibo et al. Using statistical methods to compute the probability distribution of message response time in controller area network. **IEEE Transactions on Industrial Informatics**, v. 6, n. 4, p. 678-691, 2010.