

UM SIMULADOR PARA ANÁLISE DE DESEMPENHO DE REDES VEICULARES COM BARRAMENTO CANBUS

CÂNDIDO, Júlio César de Melo¹; SANTOS, Everthon Valadão²; SILVA; Diego Mello²;
CAMPOS, Gustavo Lobato²

¹Estudante do curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) – *Campus* Formiga, bolsista (PIBITI). E-mail: juliocesarifmg@gmail.com

²Professor orientador do IFMG – *Campus* Formiga. E-mail: everthon.valadao@ifmg.edu.br, diego.silva@ifmg.edu.br, gustavo.lobato@ifmg.edu.br

Resumo: Este documento apresenta resultados sobre o desenvolvimento e teste de um simulador de redes veiculares capaz de operar sobre arquivos de entrada no formato “.dbc”, simular o tráfego de mensagens na rede, assim como computar algumas métricas de desempenho do barramento, como “tamanho dos *bursts*”, “duração dos *burst*”, e “tempo de atraso de pior caso” (WCRT, do inglês *Worst Case Response time*). Resultados são reportados sobre conjuntos de quatro instâncias. Pelo tempo de execução observado conclui-se que é uma ferramenta interessante de ser integrada com outras aplicações.

Palavras-chave: CANBus. WCRT. *Burst*.

1 INTRODUÇÃO

Segundo Junior (2012), o avanço tecnológico das arquiteturas veiculares tem permitido o uso de novos recursos que melhoram a experiência dos condutores e conforto dos demais passageiros, tais como tocadores de DVD, MP3, microfones, amplificadores, alto-falantes, rádio, conversores de áudio, displays, sistemas de informação, sistemas de reconhecimento de voz, sistemas de navegação, telemetria, processamento de dados, dentre outros. Novos modelos de veículos automotores tendem a incluir uma quantidade crescente de sensores em seus sistemas, o que impacta diretamente no desempenho do barramento de comunicações. Outra questão de projeto relevante consiste no uso crescente da multimídia embarcada nos automóveis, que gera uma necessidade de vazão de dados cada vez maiores para suprir a demanda dessas funcionalidades.

Além disso, redes automotivas são redes sensíveis a temporização, onde o aumento da quantidade de mensagens que trafegam no barramento gera um aumento significativo em sua latência, que é uma das maiores preocupações das montadoras, pois a segurança de quem utiliza o veículo está diretamente ligada à velocidade em que informação compartilhada entre as centrais eletrônicas se propaga. O advento dos carros autônomos torna essa preocupação ainda mais crítica.

Atualmente o padrão de barramento utilizado na área automotiva é o CAN (*Controller Area Network*) padronizado como ISO 11898. O CAN é uma rede confiável e veloz, porém com vazão limitada para propósitos de alta/média carga de banda, oferecendo no máximo 1 Mbps de vazão (DI NATALE, 2012). O projeto de uma rede automotiva necessita, portanto, ser bem elaborada, e validada. Uma ferramenta que pode ser usada neste contexto são os simuladores de rede, capazes de verificar, validar e metrificar o desempenho de uma arquitetura.

Diante do exposto, este documento apresenta resultados sobre o desenvolvimento de um simulador CANbus capaz de simular mensagens de uma rede veicular, e gerar métricas de desempenho para avaliação de redes com dado conjunto de mensagens e temporização de entrada. Dentre as vantagens dessa ferramenta está em ser de código aberto, que permite a possibilidade de integração com otimizadores de parâmetros, e comparação de resultados de trabalhos da literatura e com outros simuladores, como o CANOe, da empresa alemã Vector¹ e o NETCARBENCH (BRAUN *et al.*, 2007), o Real Time at Work (RTaW)² em algumas funcionalidades comuns.

2 MATERIAIS E MÉTODOS

O trabalho foi desenvolvido em fases, que estão descritas a seguir. Inicialmente fez-se uma breve revisão de literatura em busca de conceitos sobre CANbus para compreensão sobre o funcionamento do protocolo, em especial sobre aspectos como acesso ao meio (MAC), arbitragem, prioridade de mensagens, quadros cíclicos e outros detalhes importantes que devem ser refletidos na simulação da rede. Em seguida o simulador foi implementado em linguagem C, sendo este composto pelos módulos `get_candb.c`, que implementa a entrada de dados feita mediante arquivos no formato CANdb; `simulador.c`, que calcula todas as métricas de desempenho de saída e implementa o mecanismo de avanço do tempo, e `fifo.c`, que implementa uma estrutura de dados do tipo fila, usada como uma lista de eventos futuros no mecanismo de avanço de tempo de simulação de eventos discreta.

Como entrada o simulador recebe um arquivo no formato CANdb (de extensão “.dbc”) contendo o mapa de mensagem e que descreve propriedades das mensagens como ID, tempo de ciclo, tempo de *deadline*, dentre outros. O simulador filtra os campos de interesse do arquivo e define os parâmetros internos que utilizará para efetuar a simulação das mensagens

¹ Disponível em: <https://www.vector.com/int/en/products/products-a-z/software/canoe/>. Acesso em: set. 2019.

² Disponível em: <http://www.realtimework.com/software/rtaw-sim/>. Acesso em: set. 2019.

na rede. Nesta fase de implementação também se realizou um breve estudo sobre simulação de eventos discretos. Simuladores discretos têm a vantagem de avançar no tempo simulado em taxas mais altas do que ocorreria em tempo real, permitindo replicar simulações com custo menor de tempo, além de gerar saídas determinísticas que permitem validar os cálculos gerados como resposta.

Algumas métricas implementadas como saída do simulador foram o *Worst Case Response Time* (WCRT) (DE CAMPOS; HORA, 2009), o tamanho de filas de mensagens quando ocorre colisão (*bursts size*) e tempo de duração da fila durante a ocorrência de colisão (*burst time*). Em termos de cálculo, o WCRT consiste na soma de todos os piores atrasos experimentados por cada mensagem durante o horizonte de simulação, e representa o pior caso de atraso experimentado na rede. O tamanho de filas foi determinado pela média dos tamanhos das filas ocorridas somada com o produto do desvio padrão por uma constante que calibra a porcentagem de observações cujos valores ocorrem abaixo do produto mencionado, inspirado na desigualdade de Chebychev (GRECHUK *et al.*, 2010), cujo valor desta constante foi 5, garantindo 96 por cento dos casos ocorridos (Teorema de Chebyshev). Já o tempo de duração das filas foi computado pela média dos tempos de duração das filas que ocorreram nas colisões de mensagem devido às diferentes prioridades e tempos de ciclo.

Para validar o funcionamento do simulador implementado utilizou-se arquivos de entrada no formato “.dbc”, que são utilizados em softwares comerciais de projeto de redes automotivas para descrever as mensagens que irão trafegar pela rede, com ID, tempo de ciclo e *offset*. Foram conduzidos experimentos com a instância *SAE.dbc* (KUTLU *et al.*, 1996) utilizado para análise de desempenho em rede *Wireless Control Area Network*, com 53 mensagens, tempo de ciclo médio de 154.34 ms e *payload* de aproximadamente 4 Bytes; com a instância *SAE_23.dbc*, subconjunto com 23 mensagens periódicas do *benchmark* SAE; instância *PSA.dbc*, disponibilizado pela *Peugeot-Citroën Automobiles Company* com 12 mensagens; e uma instância artificial proposta pelos autores (*artificial.dbc*), criada a partir de um mapa de mensagens de rede veicular real contendo 36 mensagens. Para analisar o *runtime* de uma replicação independente do simulador replicou-se cada instância por 100 vezes, com horizonte de simulação de 2 minutos (isto é, simulou-se tráfego de mensagens da rede durante 2 minutos de tempo simulado), gerando como resposta o tempo de execução médio, WCRT, o *burst time* e o *burst size*. Os resultados obtidos foram reportados e discutidos na próxima seção.

3 RESULTADOS E DISCUSSÃO

Os resultados obtidos nos experimentos realizados encontram-se reportados a seguir, na Tabela 1. Nela são apresentados cada arquivo “.dbc” utilizado (coluna Instância), com informações sobre a quantidade de mensagens (coluna #Msg), amplitude dos IDs utilizados (coluna Range), tempo médio de execução do simulador (coluna Avg. Runtime), e resultados das métricas WCRT, *burst time* e *burst size*. Ressalta-se que pelo fato do simulador ser determinístico, os resultados das métricas não se alteraram entre as replicações.

Tabela 1 - Resultados experimentais de desempenho do simulador em diferentes “.dbc”

Instância	# Msg	Range	Avg. Runtime	WCRT	Burst Time	Burst Size
SAE.dbc	53	1 - 53	309.4 ms	141.49253 ms	0.24171 ms	34.05232
SAE_23.dbc	23	1 - 49	148.8 ms	30.79521 ms	0.20471 ms	9.65890
PSA.dbc	12	1 - 12	118.1ms	6.90841 ms	0.39033 ms	58.74480
artificial.dbc	36	129 - 1792	109.7 ms	74.27246 ms	0.14221 ms	4.19146

Fonte: Autores (2019).

Observa-se pelos resultados descritos na Tabela 1 que o tempo médio de execução do simulador foi, em todos os casos, menor do que 1 (um) segundo, tornando-o uma alternativa interessante devido ao desempenho. A diferença no pior tempo de atraso experimentado entre as instâncias SAE.dbc e SAE_23.dbc deve-se ao fato da instância SAE incluir as mensagens não periódicas, isto é, eventos disparados pelo condutor, que foram tratadas como periódicas e, portanto, impactam na colisão das mensagens, visto que as mensagens simuladas possuem IDs de baixo valor (coluna Range).

4 CONCLUSÃO

Este trabalho apresentou breves detalhes sobre a implementação de um simulador de eventos discretos, capaz de simular o tráfego de mensagens de uma rede automotiva do tipo barramento CANBus e calcular seu desempenho em termos do pior tempo de atraso experimentado pela rede (WCRT), do tamanho médio dos *burst* de mensagens quando ocorre colisão, e do tempo médio dos referidos *bursts* medidos sobre o tráfego de mensagens simulado por 2 minutos. O tempo de execução de todas as instâncias foi de menos de um segundo, mostrando-o eficiente o suficiente para utilizar integrado com outras ferramentas. O

simulador operou de maneira determinística, medindo o impacto das colisões. Como trabalhos futuros sugere-se a implementação de mensagens por eventos probabilísticas, isto é, aquelas geradas pelo condutor, que seguem uma determinada distribuição de probabilidades para ocorrer, assim como integração com metaheurísticas para otimizar estes critérios, em especial o WCRT, propondo *offsets* entre as mensagens.

AGRADECIMENTOS

Agradecemos o suporte dado pela Instituição pelo Edital 25/2018 - Programa Institucional de Bolsas de Pesquisa do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) - *Campus Formiga*.

REFERÊNCIAS

BRAUN, C.; HAVET, L.; NAVET, N. Netcarbench: a benchmark for techniques and tools used in the design of automotive communication systems. **IFAC Proceedings Volumes**, v. 40, n. 22, p. 321-328, 2007.

DE CAMPOS, M. F.; HORTA, Lucia R. Franco R. **Analyzing SAE J1939 Messages Worst Case Response Time**. SAE Technical Paper, 2009.

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

GRECHUK, Bogdan; MOLYBOHA, A.; ZABARANKIN, M. Chebyshev inequalities with law-invariant deviation measures. **Probability in the Engineering and Informational Sciences**, v. 24, n. 1, p. 145-170, 2010.

JUNIOR, H. T. **Estudo dos protocolos de comunicação das arquiteturas eletroeletrônicas automotivas, com foco nas suas características e respectivas aplicações, visando o direcionamento para o uso adequado e customizado em cada categoria de veículo**. Centro Universitário do Instituto Mauá, 2012.

KUTLU, A.; EKIZ, H.; POWNER, E. T. Performance analysis of MAC protocols for wireless control area network. *In: Proceedings Second International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'96)*. IEEE, 1996. p. 494-499.

Como citar este trabalho:

CÂNDIDO, J. C. de M. *et al.* Um simulador para análise de desempenho de redes veiculares com barramento canbus. *In: SEMINÁRIO DE PESQUISA E INOVAÇÃO (SemPI), III., 2019. Formiga. Anais eletrônicos [...]. Formiga: IFMG – Campus Formiga, 2019. ISSN – 2674-7111.*