PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Pós-Graduação em Engenharia Elétrica

Luiz Carlos Quintino dos Santos

# PROTEÇÃO CONTRA ATAQUE DO TIPO D.O.S. EM REDES CAN E CAN-FD

Belo Horizonte
2017

Luiz Carlos Quintino dos Santos

# PROTEÇÃO CONTRA ATAQUE DO TIPO D.O.S. EM REDES CAN E CAN-FD

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Alexei Manso Correa Machado

Área de concentração: Redes veiculares

Belo Horizonte

2017

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Santos, Luiz Carlos Quintino dos
S237p Proteção contra ataque do tipo D.O.S. em redes CAN e CAN-FD / Luiz Carlos Quintino dos Santos. Belo Horizonte, 2017.
134 f.: il.

Orientador: Alexei Manso Correa Machado
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica

1. Automóveis - Inovações tecnológicas. 2. Energia - Consumo. 3. Automóveis - Equipamento eletrônico. 4. Hackeres. 5. Engenharia de software. I. Machado, Alexei Manso Correa. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.



CDU: 629.113.064
Ficha catalográfica elaborada por Fernanda Paim Brito– CRB 6/2999

Luiz Carlos Quintino dos Santos

# PROTEÇÃO CONTRA ATAQUE DO TIPO D.O.S. EM REDES CAN E CAN-FD

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

_____

Prof. Dr. Alexei Manso Correa Machado – PUC MG (Orientador)

_____

Prof. Dr. Alexandre Wagner Chagas Faria – UFMG (Banca Examinadora)

_____

Prof. Dr. Gustavo Luís Soares – PUC MG (Banca Examinadora)

_____

Profª. Dra. Rose Mary de Souza Batalha – PUC MG (Suplente)

Belo Horizonte, 09 de Março de 2017

*Dedico esse trabalho a todos aqueles que buscam alguma coisa,*
*porque a vida se constrói enquanto se caminha.*

## AGRADECIMENTOS

*Agradeço primeiramente aos meus pais, que com toda simplicidade e falta de recursos conseguiram educar quatro filhos de forma honrada.*

*Agradeço também a Laice, minha cúmplice nesta jornada, que muito me ensinou e me motivou e que consegue extrair sempre o melhor de mim.*

*Agradeço aos meus filhos por serem pacientes comigo e por me perdoarem pela minha ausência reiterada.*

*Agradeço a FCA LATAM pelo apoio, confiança, pelos equipamentos disponibilizados e pela estrutura de laboratório, sem os quais esse trabalho não seria concluído.*

*Agradeço à PUC/MG pelo acolhimento e pela compreensão. Não é fácil voltar ao banco de escola após, digamos, muito tempo, fora dele.*

*E finalmente, agradeço a Deus pela saúde e por todas as graças que me concedeu e vem me concedendo sempre. Espero continuar merecedor.*

*“O carro vai se transformar em um aplicativo para smartphones”*
*(Ketter, 2016)*

## RESUMO

Já faz tempo que os veículos passaram a adotar sistemas eletrônicos como forma de trazer maior conforto e segurança ativa e passiva aos ocupantes. Com a crescente demanda tecnológica e inovações aplicadas aos veículos, estes passam a ser um alvo interessante para que hackers busquem acessá-los, e com isso o futuro dos automóveis passa por um amplo trabalho, relacionado a avaliar os impactos de alguém mal intencionado tomar controle dos sistemas do veículo. Foi utilizado um ambiente de simulação para avaliar as vulnerabilidades relacionadas ao acesso ao barramento de dados do veículo, CAN e CAN-FD. Os resultados encontrados demonstraram que é possível interferir em sistemas de indicação do painel de instrumentos, transmitindo mensagens erradas ao condutor, e até mesmo comprometer sistemas relacionados à segurança dos ocupantes, como direção e freios. Dentre os diversos tipos de ataque o DoS (negação de serviço), não apresenta solução na literatura para redes veiculares. Este trabalho apresenta uma proposta que visa minimizar os efeitos de um ataque DoS em redes veiculares, e os resultados demonstram que é possível preservar a comunicação entre os módulos do veículo durante um ataque.



Palavras chave — Rede Veicular, Barramento CAN, CAN-FD, Segurança Cibernética

## ABSTRACT

Vehicles have been adopting electronic systems for a long time as a way to bring greater comfort and active and passive safety to the occupants. With the increasing technological demand and innovations applied to the vehicles, these become an interesting target for hackers to seek to access them, and with that the future of automobiles goes through a broad work, related to assessing the impacts of malicious someone take control Of the vehicle's systems. A simulation environment was used to evaluate the vulnerabilities related to access to the vehicle data bus, CAN and CAN-FD. The results showed that it is possible to interfere with instrument panel display systems, transmitting erroneous messages to the driver, and even compromise occupant safety related systems such as steering and brakes. Among the various types of attacks, the D.o.S. (denial of service) does not present a solution in the literature for vehicular networks. This paper presents a proposal to minimize the effects of a DoS attack on vehicular networks, and the results demonstrate that it is possible to preserve communication between vehicle modules during an attack of this type.



Key-words — Vehicle Network, CAN BUS, CAN FD, Cyber Security.

## LISTA DE FIGURAS

## LISTA DE QUADROS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ABNT | Associação Brasileira de Normas Técnicas |
| ACK | Acknowledgement |
| AES | Advanced Encryption Standard |
| ARP | Address Resolution Protocol |
| AUTOSAR | AUTomotive Open System ARchitecture |
| AVB | Audio and Video Bridge |
| CANFD | Controller Area Networt with Flexible DataRate |
| CRC | Cyclic Redundancy Check |
| D.o.S. | Denial of Service |
| DES | Data Encryption Standard |
| DHCP | Dynamic Host Configuration Protocol |
| DoIP | Diagnostic Communication over Internet Protocol |
| ECU | Electronic Control Unit |
| EOF | End Of Frame |
| FCFS | First Come First Server |
| FOTA | Firmwareupdate Over The Air |
| GSM | Global System for Mobile Communications |
| ICMP | Internet Control Message Protocol |
| ID | Identifier |
| IDE | Extended ID |
| IDS | Intrusion Detection System |
| IEEE | Institute of Electrical and Electronics Engineers |
| IP | Internet Protocol |
| ISO | Internacional Organization for Standardization CAN Controller Area Networt |
| LAN | Local Area Network |
| LIN | Local Interconnected Network |
| MM | Mapa de Mensagens |
| MOST | Media Oriented Systems Transport |
| NDP | Neighbor Discovery Protocol |
| OBD | On Board Diagnosis |
| OSI | Open Systems Interconnection |

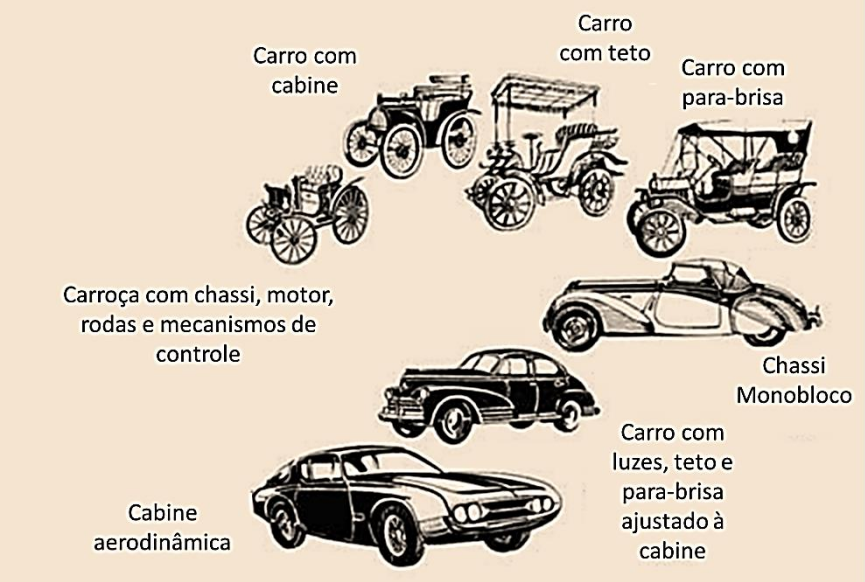
| | |
|---|---|
| PdI | Painel de Intrumentos |
| RR | RoundRobin |
| RX | Recepção |
| SEL | Safety Effect Level |
| SHA | Secure Hash Algorithm |
| SIL | Safety Integrity Level |
| SJF | Shortest Job First |
| SOME | Scalable service Oriented MiddlewarE |
| SPI | Serial Peripheral Interface |
| STB | Standby |
| TCP | Transmission Control Protocol |
| TPMS | Tire Pressure Measurement System |
| TX | Transmissão |
| USB | Universal Serial Board |

**SUMÁRIO**

# 1 INTRODUÇÃO

O automóvel já foi uma máquina puramente mecânica. Era a tradicional carruagem com adição de um motor a combustão interna. O veículo evoluiu desde a sua criação, e conteúdos corriqueiros que conhecemos hoje são frutos de evoluções incrementais (Figura 1). A evolução trouxe módulos eletrônicos aos veículos e os sistemas veiculares atuais possuem módulos capazes de trocar informações e compartilhar recursos em uma rede de comunicação de dados de alta velocidade, como a *Controller Area Networt* (Rede CAN) (ISO, 1996). Essa rede, por ter sido desenhada para veículos, foi projetada para ser robusta, imune contra perturbações eletromagnéticas, tolerante a curto-circuito, a grandes variações de temperaturas, e barata, de forma a resultar em um menor impacto no custo de cada um dos módulos do veículo. Entretanto, não foi pensada do ponto de vista de segurança contra ataques, externos ou internos, por não ser uma preocupação à época (ISO, 1999).

**Figura 1 – A evolução dos veículos**

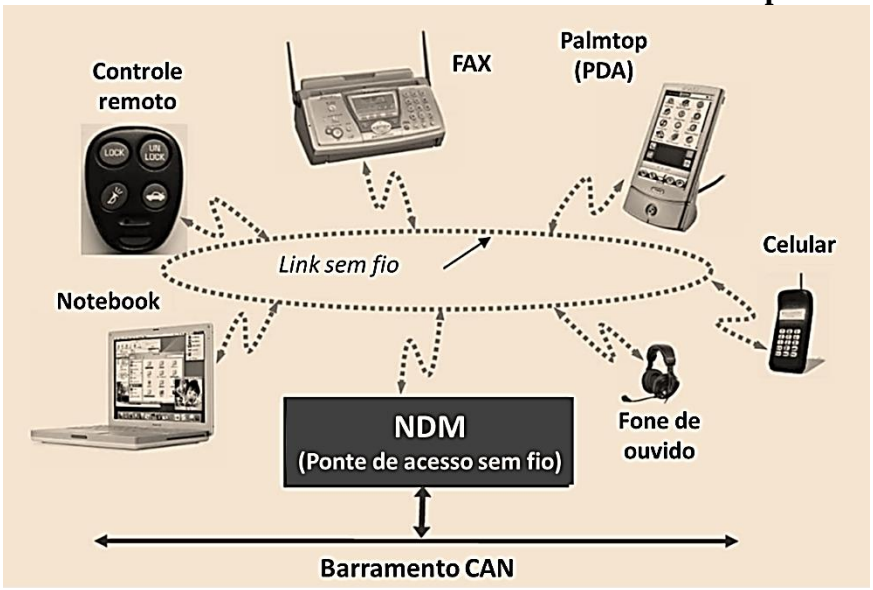



**Fonte: Adaptado de (World-Mysteries, 2013)**

A crescente automação dos veículos e a conectividade com dispositivos externos tornam possíveis que os módulos tomem decisões sobre o controle dos sistemas veiculares, tanto sistemas de conforto quanto de segurança (Larson, et al., 2008). Sistemas como *Emergence Brake* que freia o veículo automaticamente caso seja detectado um pedestre ou algum obstáculo, *Stop&Start* que desliga o motor ao parar o veículo e colocá-lo em ponto morto, religando-o automaticamente quando é pressionada a embreagem para engrenar a 1ª marcha e *Automatic Park Assistance* que estaciona o veículo automaticamente em vagas paralelas e a 90º, são

exemplos em que módulos eletrônicos controlam funções de segurança do veículo autonomamente. Essas características geram uma preocupação com a segurança dos sistemas e dos ocupantes dos veículos. Vários estudos já demonstraram fragilidades na segurança dos módulos eletrônicos e das redes de comunicação atuais (Larson, et al., 2008) (Wampler, et al., 2009) (Kleberger, et al., 2011). A Figura 2 apresenta um exemplo de vulnerabilidade onde a rede do veículo pode ser acessada por dispositivos externos.

**Figura 2 – Vulnerabilidades em uma rede veicular através de dispositivos sem fio**




**Fonte: Adaptado de (Mahmud, et al., 2006)**

A vulnerabilidade em veículos passou a ser um ponto de atenção e os *hackers* já começaram a se interessar por esse tipo de tecnologia. Em uma apresentação sobre segurança veicular, a Black Hat 2015, um *hacker* demonstrou vulnerabilidades em um veículo, ao vivo, acessando remotamente seus sistemas e a partir daí obtendo controle sobre alguns módulos. A montadora do veículo envolvido corrigiu as vulnerabilidades apresentadas, gerando um *recall* em 1,4 milhões de veículos somente nos USA (Miller, et al., 2015).

Diante dessa situação uma série de análises devem ser feitas a fim de verificar e balancear os riscos e os custos envolvidos na aplicação de políticas de segurança para proteção de redes veiculares e os impactos em não adotá-las. É importante classificar os sistemas veiculares quanto aos riscos inerentes de seu funcionamento (perda de conforto, perda de desempenho, risco de acidente), classificar módulos críticos, analisar mensagens e sinais que devem ser protegidos, analisar impactos relacionados à violação de segurança em cada classe, qual seria o plano de contingencia e os esforços envolvidos em cada tipo de rede.

Nesse trabalho é apresentada uma proposta para minimizar os ataques do tipo DoS em qualquer tipo de rede que utilize barramento CAN e CAN-FD por meio da criação de mecanismo na camada de arbitragem para dosar as mensagens, evitando a sobrecarga do barramento que ocorre durante um ataque DoS. Serão apresentados os resultados de testes feitos em um simulador. O objetivo é que se tenha um mínimo de perda de comunicações autênticas diante de um ataque DoS e que sistemas críticos ou relacionados com segurança não deixem de operar.

Serão apresentados no capítulo 2 os principais aspectos relacionados à proteção de redes. São estudadas as redes veiculares largamente utilizadas no mercado automobilístico e os principais tipos de ataque a que estão sujeitas. É feito um estudo sobre o tipo de ataque DoS em redes do tipo LAN e as principais formas sugeridas na literatura para combater tais ataques. É aprofundado o estudo sobre o funcionamento do barramento CAN, suas características de hardware e limitações. São feitos estudos sobre diversos tipos de ataque ao barramento CAN, as formas de acesso ao barramento e as consequências de um ataque em um veículo. É demonstrado como gerar um ataque do tipo DoS em um barramento CAN e as consequências para o barramento e para os nós conectados sobre ele.

No capítulo 3 é apresentado o simulador utilizado para a realização dos testes e demonstrado como é possível criar o tipo de ataque DoS em uma rede veicular.

No capítulo 4 é apresentada uma proposta para minimizar os ataques do tipo DoS em um barramento CAN. São feitos estudos para avaliar sua eficácia e apresentados resultados de testes em simulador. São gerados vários gráficos para analisar o resultado das tabelas geradas e confrontados os resultados com especificações de redes reais para validar a eficácia da proposta.

No capítulo 5 são tecidas as conclusões a respeito do experimento realizado, sua aplicação e limitações e contribuições para o automóvel e são apresentadas propostas para trabalhos relacionados, com base no aprendizado gerado que poderiam dar segmento ao estudo realizado.

# 2 PROTEÇÃO DE REDES VEICULARES

A proteção de redes vem sendo estudada largamente para comunicação entre computadores. A necessidade de estudo veio com a difusão do uso da internet para um número crescente de finalidades. Já as redes veiculares possuem estudos mais recentes, principalmente porque não existia um interesse em atacar este tipo de sistema. Com o aumento do uso da comunicação de rede em veículos e a recente possibilidade de conectar o veículo com o mundo externo o veículo passa a ser alvo de ataques, e os estudos de segurança para este tipo de rede tende a aumentar.

## 2.1 Segurança de redes

Os Sistemas de Detecção de Intrusão (IDS) são processos de acompanhamento e análise dos eventos que ocorrem em um computador e/ou sistema de rede a fim de detectar sinais de problemas de segurança (Kuo, et al., 1999). O objetivo da detecção de intrusão é descobrir intrusões em um computador ou rede observando várias atividades ou atributos de rede. Esse tipo de sistema se baseia no mapeamento de padrões conhecidos e costuma ser desenvolvido utilizando lógica *fuzzy* e redes neurais para identificar, classificar e determinar o nível de proteção para os tipos de acesso e aprender padrões novos.

Ameaças a um ambiente de rede, identificadas por Abraham e Jam (Abraham, et al., 2004) incluem:

a) **Rede não autorizada**: Senha pobre pode facilitar a intrusão em sistemas;
b) **Acesso inadequado a recursos da rede**: Uso de privilégios impróprios para o usuário dando acesso ou controle a arquivos no computador;
c) **Divulgação de dados**: Inclui modificação de dados por criptografia fraca, proteção do anti-virus e ferramentas de proteção deficientes;
d) **Falsificação** ou *spoofing* de tráfego rede: Criptografia deficiente, falha no registro de eventos temporais, autenticação do mecanismo de código deficiente ou falta de uma assinatura digital;
e) **Interrupção das funções de uma *Local Area Network*** (LAN): Inclui falta de habilidade para detectar tráfego com padrão anormal, falta de habilidade para mudar o roteamento, manipular falhas de *hardware*, modificações feitas nos componentes de *hardware*, *hardware* da rede com segurança física inapropriada.

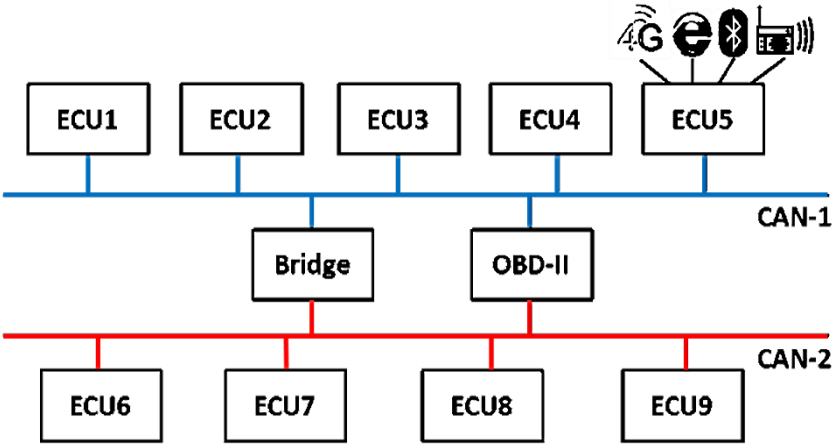
Existem certos elementos que norteiam a política de segurança que devem ser considerados no desenvolvimento de uma rede segura, que inclui, segundo a ABNT (ABNT, 2005):

a) **Disponibilidade**: o sistema deve estar disponível de forma que quando o usuário necessitar possa utiliza-lo. Dados críticos devem estar disponíveis ininterruptamente;
b) **Legalidade**: as informações devem respeitar normas que a regulam;
c) **Integridade**: o sistema deve estar sempre íntegro e em condições de ser usado;
d) **Autenticidade**: o sistema deve ter condições de verificar a identidade dos usuários, e esse ter condições de analisar a identidade do sistema;
e) **Confidencialidade**: dados privados devem ser apresentados somente aos donos dos dados ou ao grupo por ele liberado.

## 2.2 Tipos de redes veiculares

No exemplo da Figura 3 é apresentada uma arquitetura veicular típica, com dois barramentos CAN independentes (CAN-1 e CAN-2). Os barramentos são interconectados através de um *Bridge* (uma ponte, ou *Gateway*). Nesse exemplo, um dos módulos eletrônicos (ECU5) possui diversas interfaces de comunicação externa (internet, *Bluetooth*, *WiFi*, etc), e ainda possui uma conexão física com os barramentos. Nesta arquitetura possui também um conector chamado OBD-II (*On Board Diagnosis*) para comunicação de diagnóstico dos módulos utilizando-se equipamentos externos conectados ao barramento.

**Figura 3 - Exemplo de arquitetura veicular**




**Fonte: Adaptado de (Wang, et al., 2014)**

Em 1991 o Mercedes S-Class foi o primeiro veículo a utilizar uma rede automotiva, a rede CAN (ISO, 1999). As mensagens que trafegam em uma rede veicular transportam informações sobre sensores, diagnósticos de sistemas, informações de controle e comandos. As mensagens trafegam em uma rede em *broadcast* onde uma mensagem é enviada de um nó para todos os outros nós, e não existe colisão de mensagens, uma vez que é utilizado um controle de prioridade baseado no ID (identificador) de mensagens. As mensagens em uma rede CAN podem trafegar a uma velocidade de até 1Mbit/s.

A rede *Local Interconnect Network* – Rede Interconectada Local (LIN), surgiu como uma opção de menor custo para conectar sensores e módulos com eletrônica menos potente, em que o custo era determinante. Devido às suas características não é necessário um *transceiver* (controlador dedicado) e nem um oscilador caro, podendo utilizar um circuito RC para gerar o *clock* da rede e assim definir a frequência de trabalho da rede. A rede LIN opera em uma topologia *Master-Slave*, a uma velocidade de até 20kbps.

As redes *Flexray* e *Media Oriented Systems Transport* (MOST) surgiram como um complemento à rede CAN onde a alta velocidade era determinante, alcançando velocidades de: FLEXRAY (até 10Mbits/s) e MOST (até 150Mbits/s utilizando-se fibra óptica). Entretanto, o uso dessas duas tecnologias é muito restrito devido aos altos custos envolvidos, estando disponível somente em poucos modelos de veículos de alto luxo.

Devido ao número cada vez maior de módulos e funções em um veículo, que chega a utilizar até 100 módulos interconectados em uma rede, o protocolo *Ethernet* vem se apresentando como uma alternativa aos protocolos mostrados anteriormente. Principalmente por ser um padrão já estabelecido de tecnologia em comunicações de escritório, engenharia industrial e indústria aeroespacial. Além da larga utilização possui uma velocidade até 100 vezes superior à CAN, e um custo menor que as atuais alternativas já consolidadas.

Os protocolos *Ethernet* discutidos atualmente para aplicação veicular incluem: DoIP para diagnóstico, AVB IEEE 1722 802.1AS para áudio e vídeo, ISO 15118 para *Smart Grid*, SOME/IP para controle de comunicação, *Bonjour*, DHCPv6, ICMPv6/ICMP e NDP/ARP para resolução de endereços e sensores.

O uso da *Ethernet* na indústria é um exemplo bem sucedido de aproveitamento de sua versatilidade em ambientes adversos com necessidade de alto índice de confiabilidade à exposição de grande variação de ruídos, temperaturas e vibrações. Apesar de seu histórico, para a aplicação veicular existem também dificuldades, como por exemplo:

a) A migração de um sistema conhecido (CAN) para um novo protocolo (*Ethernet Veicular)*;
b) A dificuldade de implantação de sistemas de real time usando protocolos TCP;
c) A gestão de uma pilha TCP/IP é mais complexa que simplesmente receber dados seriais;
d) O tamanho mínimo de um *frame Ethernet* é de 64 bytes, enquanto comunicações veiculares típicas utilizam 0 a 8 bytes. Esse *overhead* pode reduzir a eficiência da transmissão de dados.

Silva (Silva, 2008) apresenta as principais redes utilizadas em veículos, como CAN, LIN, MOST, Zigbee e Bluetooth. Apesar de mencionar em seu trabalho a possibilidade de utilização de redes Zigbee, não existem estudos nas indústrias automobilísticas nesse sentido. A rede MOST, também mencionada, não tem grande volume de utilização e está limitada a alguns poucos fabricantes de automóveis de luxo, dado seu alto custo.

Em trabalhos como o de Kim (Yong Kim, 2011), a *Ethernet* aparece como uma opção viável de barramento para a indústria automobilística e são destacados os sistemas de barramento a serem considerados em veículos e suas características principais (Quadro 1).

**Quadro 1 – Características dos barraemntos utilizados em veículos**

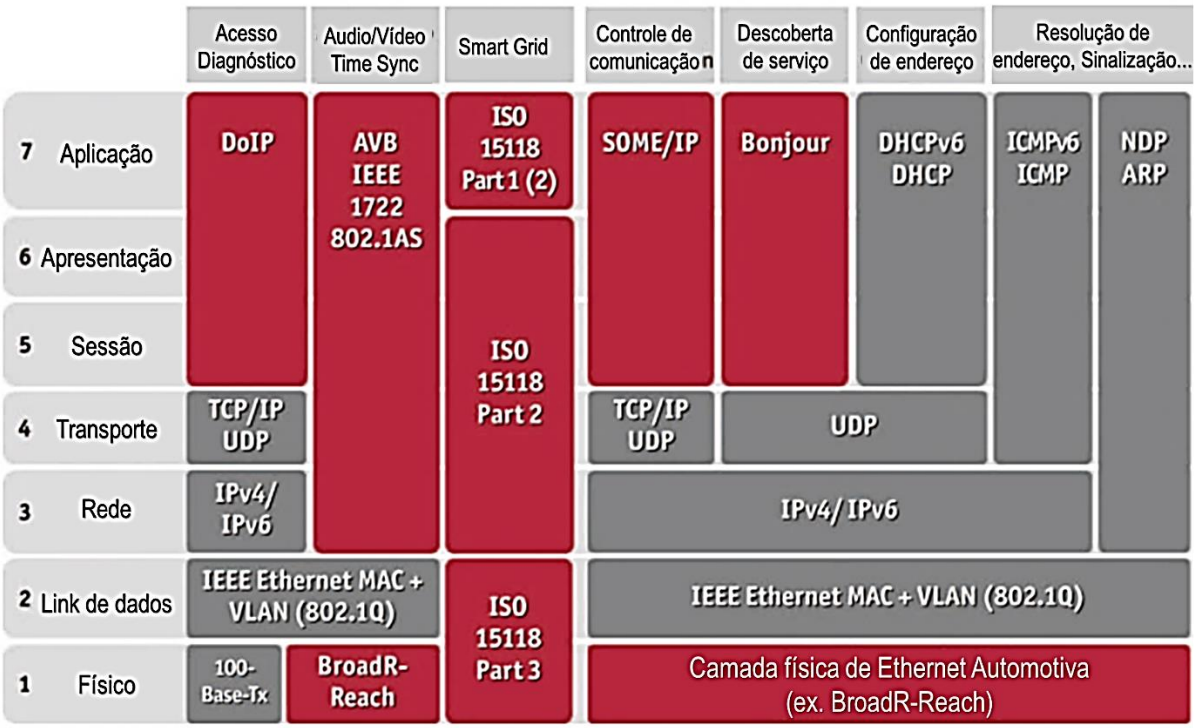
| Rede | Característica | Barramento | Velocidade (Kbps) |
|---|---|---|---|
| **LIN** | Receptor e transmissor universal assíncrono (*Universal Asynchronous Receiver / Transmitter* - UART) | estrela | 20 |
| **CAN** | Acesso múltiplo por sensor de tráfego (*Carrier Sense Multiple Access* - CSMA/CR) | anel ou estrela | 1.000 |
| **FlexRay** | Acesso múltiplo por divisão de tempo (*Time Division Multiple Access* - TDMA) | estrela | 10.000 |
| **MOST** | TDMA síncrona | anel | 25.000<br>50.000<br>(Fibra óptica) 150.000 |
| **Ethernet** | Comutação bidirecional (*Full-Duplex*) | estrela | 100.000 |

**Fonte: (Yong Kim, 2011)**

Segundo o trabalho, a principal vocação da rede *Ethernet* seria para entretenimento, sensores, câmeras, convergência de comunicação, devido ao crescimento no número de módulos eletrônicos, compartilhamento de informações em "zonas" ou "domínios" e diagnóstico. O trabalho trata também de alguns desafios na aplicação da rede *Ethernet* no ambiente veicular e dos vários protocolos que poderiam ser utilizados sobre esse barramento.

Em (Schaal, 2012), são apresentas as motivações para o uso da rede *Ethernet* em veículos e os protocolos distribuídos no modelo OSI, identificando nas colunas as principais utilizações e nas linhas a camada OSI afetada (Figura 4).

**Figura 4 - IP (Protocolos de Internet) em aplicação automotiva mapeada no modelo de referencia OSI**




**Fonte: Adaptado de (Schaal, 2012)**

Merian (Mearian, 2014), destaca os desafios da utilização da rede *Ethernet* em veículos e a criação de grupos de pesquisa, como a *OPEN Alliance Special Interest Group*, um grupo sem fins lucrativos que estuda a padronização de soluções para a utilização da *Ethernet* em veículos. Participam desse grupo Montadoras importantes no mercado mundial, como GM, Honda, BMW, entre outras, além de fornecedores de módulos eletrônicos, sistemas veiculares, fabricantes de microeletrônica. Recentemente, foi formado o grupo IEEE 802.3 para avançar os estudos sobre *Ethernet* em par trançado de 100Mbps.

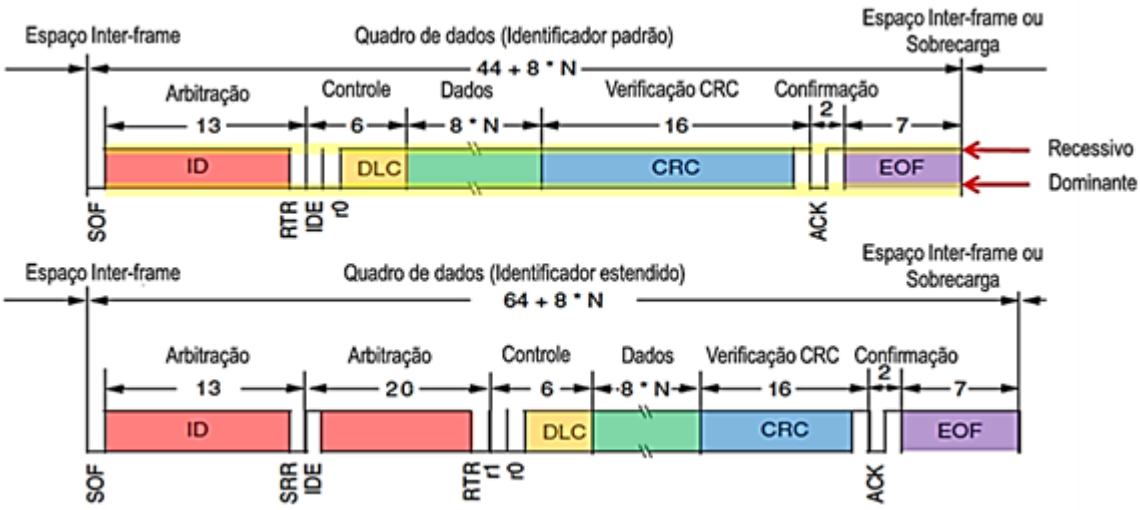
## 2.3 Características de uma Rede CAN

A rede CAN se baseia no modelo OSI e possui uma estrutura em camadas onde a camada inferior é responsável pelo acesso ao barramento. Uma das principais características da rede CAN é de ser *avoid colision*, ou seja, não existe colisão de mensagens no barramento físico.

Uma mensagem CAN (Figura 5), é composta por um ID, alguns bits de controle, 8 bytes de dados, um *Cyclic Redundancy Check* (CRC) ou verificação de redundância cíclica, que é um cálculo matemático para testar a integridade de uma mensagem. Possui também um bit de

*Acknowledgement* (ACK) que nada mais é que uma confirmação dos outros nós sobre a recepção de uma mensagem transmitida e mais alguns bits reservados para controle de erro no *End Of Frame* (EOF) (ISO, 1999).

**Figura 5 - Data frame de uma mensagem**

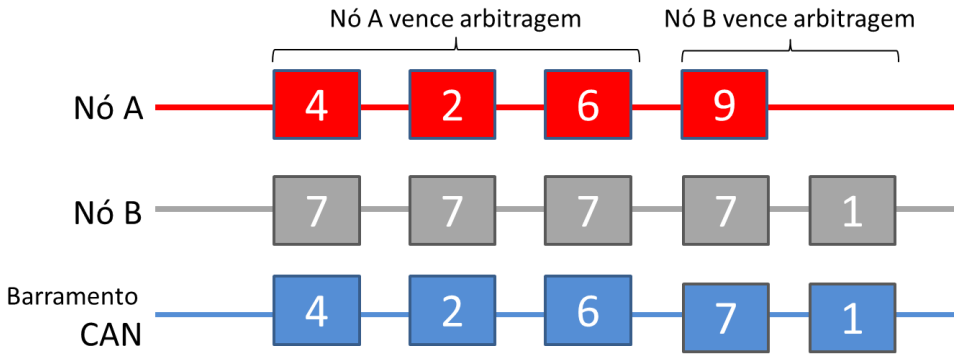



**Fonte: Adaptado de (ISO, 1996)**

O identificativo de uma mensagem (ID) serve, para identificá-la na rede. Desta forma, cada mensagem diferente possui um único ID que determina o que é aquela mensagem. Como na rede CAN não existe informação sobre origem e destino da mensagem, é o ID que vai representar o papel de orientar o nó que recebe a mensagem de filtrar se é uma mensagem que lhe serve ou não.

Além de identificar a mensagem o ID também faz a importante função de determinar a prioridade de acesso ao barramento CAN. Quanto menor o número do ID de uma mensagem, maior será sua prioridade. No caso em que dois ou mais nós queriam transmitir ao mesmo tempo no barramento o ID é a primeira informação a acessar o barramento. À medida que cada bit do ID é transmitido, este bit é lido de volta. Uma vez que o bit 0 é dominante e o bit 1 é recessivo, o ID menor será dominante no barramento. O nó que perceber um bit dominante em relação ao que escreveu irá entender que existe um nó que tem um ID prioritário e irá cessar a transmissão, permitindo que a mensagem prioritária seja transmitida. O nó ou nós que não conseguiram transmitir irão aguardar até o próximo EOF, momento em que os nós que esperam para iniciarem o acesso ao barramento novamente. A Figura 6 apresenta um diagrama que representa dois nós, C e B tentando transmitir ao mesmo tempo. Pode-se observar que o nó C obteve acesso ao barramento todas as vezes que escreveu mensagens com o menor valor de ID.

**Figura 6 - Arbitragem de acesso ao barramento CAN**



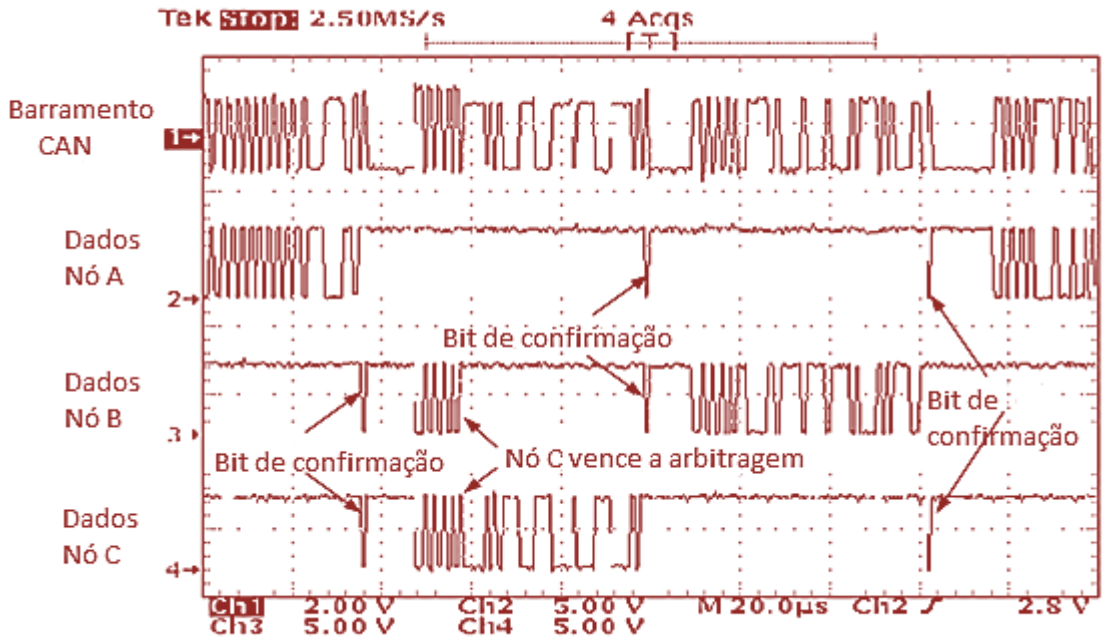

**Fonte: Adaptado de Corrigan (Corrigan, 2008)**

Uma mensagem CAN pode ter ID de 11 ou 29 bits de comprimento. Um mesmo barramento pode conter nós que trabalham com ID's de tamanhos diferentes, desde que o nó que irá receber a mensagem tenha condição de identificá-lo. Um nó que só consegue ler ID de 11 bits não poderá receber uma mensagem de 29 bits, o bit *IDentification Extended* (IDE) serve para identificar quando um ID de 29 bits está sendo utilizado na mensagem.

O barramento CAN tem que trabalhar com um *busload* ou taxa de utilização do barramento, de até 60%, de forma a garantir lacunas no barramento onde todos os módulos possam acessa-lo (Natale, 2008). No desenvolvimento da rede tem que ser calculado o *busload* para se avaliar a necessidade de se adicionar novos barramentos e comportar a quantidade de mensagens.

As mensagens CAN são do tipo *broadcast*, isso quer dizer uma mensagem é enviada para todos os nós do barramento ao mesmo tempo. É impossível determinar quem é o emissor de uma mensagem ou para quem ela vai. Apesar de existir um ID que identifica a mensagem e sua prioridade, é impossível determinar qual dos nós que efetivamente está escrevendo aquela mensagem no barramento ou quem irá utiliza-la. Na Figura 7 podem ser observadas as formas de onda de vários nós tentando acessar o barramento, a sinalização de confirmação dada pelos nós que receberam a mensagem com um ACK, e o tempo de silencio após o ACK, para que algum nó possa informar, durante o EOF se houve algum erro de integridade ou falha no recebimento da mensagem. Caso ocorra algum erro, a mensagem será retransmitida.

**Figura 7 - Trafego no barramento CAN**




**Fonte: Adaptado de (Corrigan, 2008)**

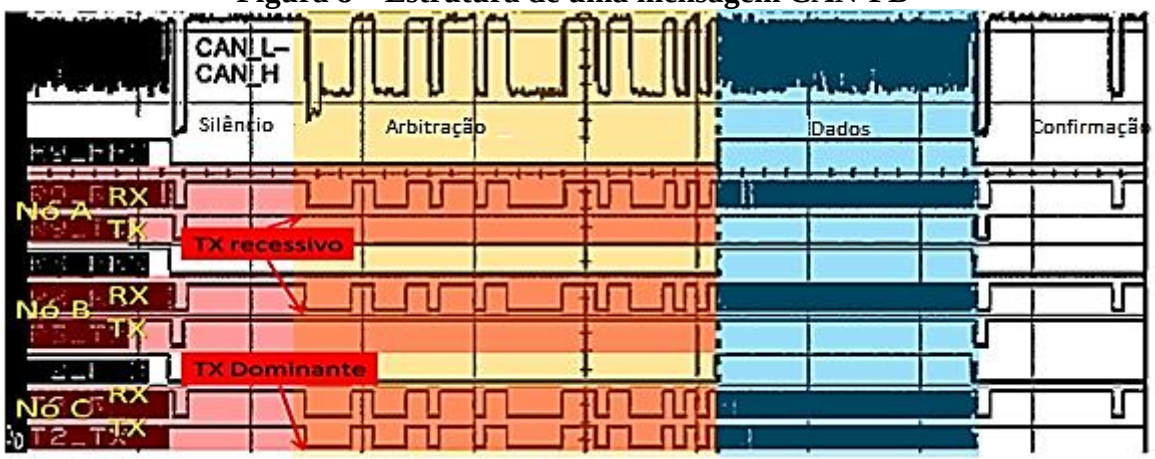
Para garantir que uma mensagem não irá perder informação, devido a falhas no meio físico, existe um mecanismo de controle de integridade da mensagem, através de um cálculo de CRC de 15 bits. O CRC é capaz de detectar erros com 5 ou menos bits modificados, e todos os erros em rajada de até 15 bits de comprimento e todos os erros que afetam um número ímpar de bits. Erros multi-bit fora dessa especificação (6 ou mais bits disturbados ou rajadas mais de 15 bits) são detectados com uma probabilidade de 3x10-5 (Natale, 2008).

Existe também uma ferramenta para garantir o sincronismo dos sinais físicos na rede CAN, chamada *bit stuffing*. Esta ferramenta altera o nível físico de um bit toda vez que houver uma sequência de 5 bits com mesmo sinal lógico. Este artifício garante que, no máximo, a cada 5 bits haverá uma sincronização no tempo de medição dos bits por cada nó que está lendo o barramento.

A CAN-FD utiliza os mesmos princípios da CAN convencional, porém com velocidade e número de bytes maior. Enquanto a CAN convencional pode transmitir no máximo 8 bytes de dados a 1Mbps, a CAN-FD pode transmitir até 64 bytes de dados e um CRC de 127 bits em até 12Mbps. Para que houvesse compatibilidade entre CAN e CAN-FD, foi adotado o mesmo conceito funcional utilizado na CAN, e a velocidade elevada acontece somente na fase de dados, daí o nome FD que significa *Flexible Data-rate*, ou seja, é possível modificar a taxa de transmissão de forma flexível, no meio da transmissão, como apresentado na

Figura 8 em Data Phase. A figura mostra como o CAN FD utiliza exatamente o mesmo processo de arbitragem da rede CAN convencional. O bit dominante “0” se sobrepõe ao recessivo “1”. Quando um nó que está competindo pelo barramento lê um bit diferente do que colocou no barramento este silencia o envio, dando acesso ao nó que tem o ID menor.

**Figura 8 – Estrutura de uma mensagem CAN-FD**



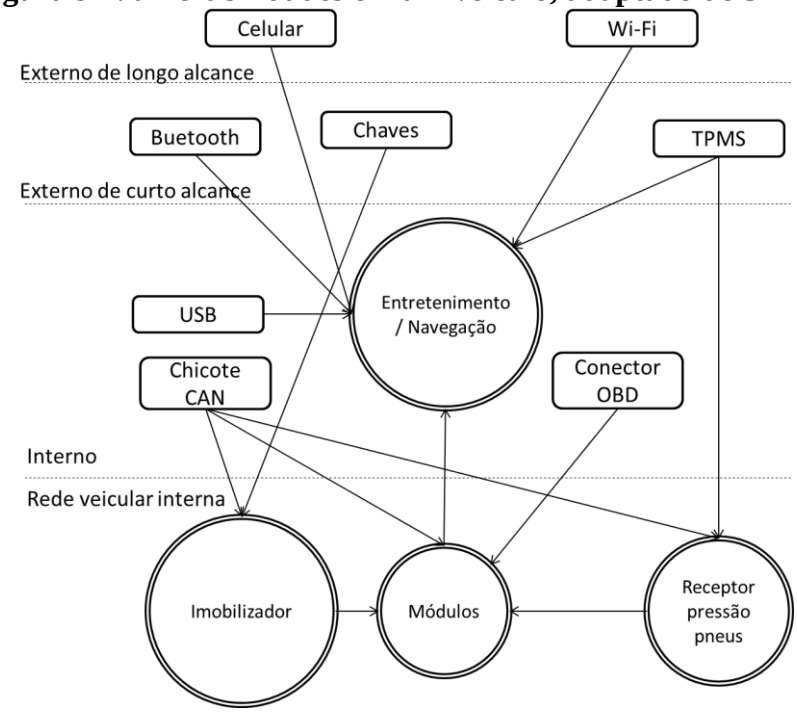

**Fonte: Adaptado de (Florian Hartwich, 2012)**

## 2.4 Formas de acesso não autorizado ao barramento CAN de um veículo

A seguir serão apresentadas as principais formas de acesso ao barramento CAN. Este trabalho não pretende explorar como acessar o barramento, mas apontar quais são as formas usuais. Para todos os efeitos será considerado que algum dos meios de acesso apresentados foram bem sucedidos e que o invasor obteve acesso ao barramento e com isso está apto a realizar algum tipo de ataque.

É preciso ter em mente que nos tempos atuais as pessoas têm mais acesso a dispositivos eletrônicos e o conhecimento gerado por alguma pessoa ou grupo, de forma licita ou ilícita, é rapidamente disseminado (Smith, 2016). Equipamentos que escrevem ou leem memórias de módulos eletrônicos podem ser adquiridos pela internet facilmente. Munidos destes equipamentos é possível fazer *dump* de *firmware*, copiar a programação dos módulos eletrônicos, modificar funções, calibração, adulterar a tabela de códigos secretos, fazer cópia de chaves. Além disso é possível usar algum tipo de solução mais invasiva, através de intervenção no *hardware*. A Figura 9 apresenta um esquema que demonstra as vulnerabilidades de um veículo, relacionadas com a distância de acesso, dentro e fora do veículo.

**Figura 9- Vulnerabilidades em um veículo, adaptado de Smith**




**Fonte: (Smith, 2016)**

### *2.4.1 Acesso através do conector de diagnóstico – On Board Diagnosis (OBD)*

Essa é a forma mais fácil de se ter acesso ao barramento CAN de um veículo. Equipamentos que fazem esse trabalho podem ser facilmente comprados em sites brasileiros e chineses (Figura 10). Além de ler e escrever na rede CAN eles também podem transmitir para um PC ou celular de forma remota utilizando *Bluetooth*, WiFi ou GSM e têm um baixo custo de aquisição. Para atividades mais específicas ou profissionais, é possível também adquirir ferramentas mais caras ou até desenvolver sua própria ferramenta com algum conhecimento de *hardware*, *software* e *kits* de desenvolvimento apropriados.

Uma vez que se tem acesso ao barramento é possível ler e escrever mensagens livremente de forma anônima, acessando informações de trafego de dados do veículo, dando possibilidade de injetar comandos e desta forma comprometer a estabilidade da rede.

**Figura 10- Dispositivos genéricos de acesso ao barramento CAN**


**Fonte: (Imagem: DX.com)**

### *2.4.2 Acesso através de firmware adulterado dos módulos internos*

De posse de equipamentos como os descritos acima ou com algum tipo de acesso remoto ou físico aos módulos do veículo é possível atualizar seu *firmware* ou ainda ler os códigos secretos das chaves que habilitam a partida e reescrevê-los ou apaga-los.

Um módulo adulterado pode inibir algum tipo de funcionamento básico, desabilitar determinadas funções ou ainda atrapalhar o funcionamento de outros módulos que dependem de informações de sensores e atuadores que passam pelo barramento CAN. Dessa forma, é possível adulterar um módulo que tenha alguma vulnerabilidade, mesmo que esse não tenha funções críticas, e através dele comprometer o funcionamento de serviços de conforto além de serviços críticos ou de segurança ativa e passiva que dependam do barramento.

### *2.4.3 Por meio de um módulo adulterado fisicamente*

É possível que alguém mal intencionado construa ou compre um módulo adulterado para realizar determinada tarefa e o substitua pelo módulo original de um veículo tomando controle do barramento. Essa adulteração pode ser conseguida por meio de engenharia reversa em um módulo, leitura da memória e comparação com outros módulos semelhantes, leitura do trafego de comunicação entre o micro controlador e a memória ou outros sistemas eletrônicos presentes na placa de circuitos, e assim por diante. A tarefa de *Hackear* um módulo exige conhecimento de arquitetura de *hardware* e *software*, entretanto, uma vez que alguém descubra como obter acesso ou modificar um módulo o conhecimento necessário pode ser comercializado ou difundido livremente na internet. Módulos adulterados e serviços de adulteração podem ser vendidos para os mais diversos fins.

### *2.4.4 Por meio de Firmware-update Over The Air (FOTA)*

Assim como softwares de computadores e celulares, os firmwares dos módulos eletrônicos dos veículos precisam ser atualizados com uma frequência cada vez maior. Atualmente alguns fabricantes disponibilizam esse serviço nas oficinas credenciadas, através do conector OBD. Está cada vez mais comum os fabricantes de veículos disponibilizarem o serviço de atualização de *firmware* de forma remota. Chamado de FOTA, o sistema permite que os módulos dos veículos sejam atualizados através de *download* do novo *firmware* pelo próprio veículo sem a necessidade de levá-lo a uma concessionária ou oficina, geralmente utilizando uma comunicação de dados por WiFi ou chip de celular (GSM).

### *2.4.5 Por meio de outro tipo de vulnerabilidade*

Existem ainda diversos tipos de vulnerabilidades nos módulos eletrônicos que podem ser exploradas, como módulos com portas de acesso remoto: *bluetooth*, WiFi, celular ou físicas: USB, RS232, SPI, que podem ser usadas para adulterar o *firmware* do módulo e seu comportamento e assim acessar o barramento CAN.

## 2.5 Ataques e proteção às redes veiculares

Nilson (Nilsson, et al., 1996) utiliza alguns critérios para classificar e medir os riscos de ataque que possibilitam modificar o *firmware* de módulos eletrônicos. A classificação se baseia no *Safety Integrity Level* ou Nível de Integridade de Segurança (SIL). Integridade de segurança é a probabilidade de um sistema ligado à segurança realizar de forma satisfatória as funções de segurança requeridas sob todas as condições em um determinado período de tempo. O *Safety effect Level* ou Nível de Efeito de Segurança (SEL) classifica os efeitos de um mau funcionamento nos sistemas em níveis de 1 a 4 (Quadro 2 - SEL de acordo com o efeito da ameaça a segurança).

**Quadro 2 - SEL de acordo com o efeito da ameaça a segurança**

| Efeito na segurança | SEL(Safety effect level) |
|---|---|
| Desastroso | 4 |
| Severo | 3 |
| Medíocre | 2 |
| Gera distração | 1 |

**Fonte: (Nilsson, et al., 1996)**

De acordo com essas categorias, os módulos eletrônicos do veículo foram divididos em cinco categorias, e classificados segundo o nível de risco em caso de ataque (Quadro 3 - Classificação dos módulos eletrônicos).

**Quadro 3 - Classificação dos módulos eletrônicos**

| Categoria do módulo (ECU) | **SIL / SEL** |
|---|---|
| Tração | 4 |
| Segurança | 4 |
| Conforto | 2 |
| Entretenimento | 1 |
| Comunicação | 1 |

**Fonte: Adaptado de (Nilsson, et al., 1996)**

A conclusão de Nilson é que para determinados módulos é desaconselhável a atualização remota sem fio *Firmware Over The Air* (FOTA) ou estudos mais aprofundados devem ser conduzidos para melhorar os aspectos relacionados à segurança, evitando que um módulo crítico possa ser corrompido.

Onishi (Onishi, 2012) faz uma estimativa, baseada em dados dos USA e Canada, em que os danos causados por acidentes envolvendo ataques cibernéticos ou *Cyber attacks* têm um potencial de prejuízo de 56 milhões de dólares por ano para cada 1% de veículos afetados, considerando as vendas de um modelo de veículo. São evidenciadas algumas dificuldades em relação à proteção do veículo contra ataques cibernéticos:

a) Limitação para atualização de *firmware on-line* como FOTA;
b) Veículos possuem módulos eletrônicos com baixa performance computacional e possuem um ciclo de vida longo, acima de 10 anos. Isto quer dizer que veículos obsoletos competem com *hackers* de ultima geração e ainda existe um grande risco de chaves de encriptação serem roubadas;
c) Dispositivos ou aparelhos autênticos podem ser infectados e reconectados ao veículo e influenciar em seu funcionamento;
d) O foco principal deve ser evitar riscos que coloquem em perigo a vida dos ocupantes do veículo. O trabalho sugere que as medidas para tratar veículos infectados são mais importantes que contramedidas para evitar infeção, se comparado a casos de ataques cibernéticos em computadores.

São apresentadas sugestões, como dividir os módulos em domínios, usando *gateway*, de acordo com a vulnerabilidade e conexões externas, separando módulos que envolvem segurança

daqueles que são de conforto e conveniência. Alguma suspeita de ataque ou infecção deveria ser informada imediatamente ao condutor do veículo para evitar complicações maiores. A última sugestão é com relação a módulos críticos, que deveriam manter um funcionamento seguro, mesmo que infectados.

No trabalho de Wolf *et all* (Wolf, et al., 2004), (Nilsson, et al., 2008) são destacadas as fragilidades das redes atuais uma vez que os estudos mostram deficiências de proteção nas redes veiculares suficientes para permitir ataques maliciosos e a possibilidade de sucesso na criação desses ataques.

Mahmud (Mahmud, et al., 2006), afirma que uma série de análises devem ser feitas a fim de estudar e balancear os riscos e os custos envolvidos na aplicação de política de segurança, como por exemplo: classificar os sistemas veiculares quanto aos riscos inerentes de seu funcionamento como perda de conforto, perda de desempenho, risco de acidente, classificar módulos críticos, analisar mensagens e sinais que devem ser protegidas, analisar impactos relacionados à violação de segurança em cada classe e estudar plano de contingência considerando:

a) **Tratamento**: o que deveria ser feito quando um incidente acontece?
b) **Notificação**: quem deveria ser notificado sobre o incidente?
c) **Proteção de evidências e logs de atividades**: que registros deveriam ser mantidos de antes, durante, e depois do incidente?
d) **Contenção**: como o dano pode ser limitado?
e) **Erradicação**: como eliminar as causas do incidente?
f) **Recuperação**: como restabelecer serviço e sistemas?
g) **Sequência**: que ação deveria ser tomada depois do incidente?

Smith (Smith, 2016), ensina técnicas para obter acesso a módulos e redes de um veículo, desde o reconhecimento do *hardware*, redes, *firmware*, arquitetura eletroeletrônica, até como burlar sistemas, introduzir comandos e modificar *firmwares*.

Onishi (Onishi, 2014), destaca que as vulnerabilidades de sistemas veiculares são mais complicados com relação a ataques cibernéticos que quando comparados com computadores e internet. Computadores pessoais possuem cerca de 2 mil componentes enquanto veículos possuem mais de 20 mil componentes e até 100 módulos eletrônicos com centenas de *mega bytes* de código de *firmware*.

Em (Brown, et al., 2014), são discutidos os riscos atuais associados aos veículos, onde "lembrar de trancar o seu carro" não é mais suficiente. São apresentados os pontos de vulnerabilidades no veículo, para os tipos de ataque e proteção, tanto de *hardware* quanto de *software*.

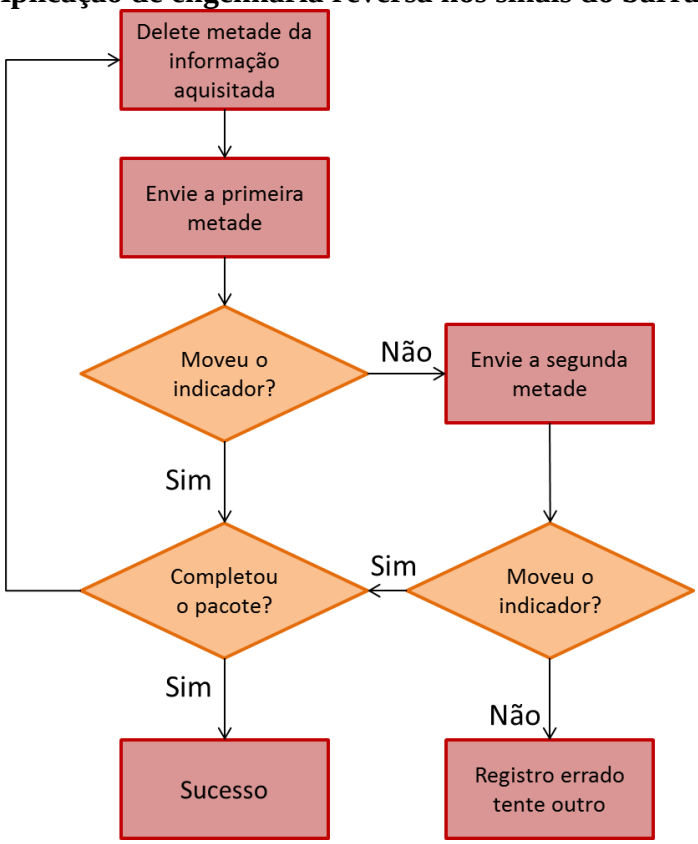
A seguir serão apresentados os principais tipos de ataques, as consequências ao veículo e a forma de evitá-los, segundo a literatura.

### *2.5.1 Ataque do tipo Read*

O objetivo deste tipo de ataque é de conhecer sobre um determinado sistema para um ataque posterior ou para utilizar alguma informação de forma privilegiada. Este ataque é feito monitorando as mensagens para descobrir o significado das mesmas. Existem ferramentas como o CANiBUS (Smith, 2014) para facilitar essa tarefa. É possível variar alguma condição real do veículo, como atuação em um interruptor ou variar a velocidade, e observar qual ou quais mensagens estão se alterando no barramento (Smith, 2016). A partir daí, mesmo que o Mapa de Mensagens daquele veículo não seja conhecido, é possível inferir com uma boa margem quais são as mensagens que controlam determinadas funções.

Existem métodos para decodificar sinais lidos na rede CAN e recriar o Mapa de Mensagens através de engenharia reversa. Como atualmente as informações não são protegidas o trabalho de descobrir o significado das mensagens é facilitado. Sinais binários, como "ligado" e "desligado", pedal do freio acionado, farol aceso, são facilmente descobertos. Sinais que representam valores numéricos, como velocidade, temperatura, ângulo de direção, podem ser decodificados seguindo o procedimento descrito na Figura 11. Os valores numéricos possuem vários bits e deve-se primeiro fazer aquisição na rede normal, variando uma grandeza, como velocidade, por exemplo, verificando quais são os bits que estão variando no barramento. Depois deve-se injetar mensagens na rede variando valores próximos dos bits observados e verificar o resultado no componente que recebe a informação, até se obter o resultado esperado.

**Figura 11- Aplicação de engenharia reversa nos sinais do barramento CAN**




**Fonte: adaptado de (Smith, 2016)**

*2.5.1.1. Consequências para o veículo para um ataque do tipo Read*

Os sistemas veiculares transmitem informações de forma descoberta (Hoppe, et al., 2007). São informações de monitoramento de sensores, estado geral do veículo, falhas, mensagens e avisos, controle de funções distribuídas, comandos. Estas informações podem ser atrativas para estudantes de Engenharia, que podem desenvolver seus próprios sistemas para o veículo, empresas que desenvolvem aparelhos profissionais para vender a clientes ou oficinas de manutenção. Pode ser útil também a empresas que desejam criar ferramentas para burlar sistemas veiculares, como habilitar partida de forma indevida ou modificar módulos eletrônicos, copiar chaves. E pode ser útil para quem quer aprender sobre o funcionamento do veículo com finalidades diversas.

*2.5.1.2. Formas de evitar um ataque do tipo Read*

Este tipo de ataque pode ser evitado com criptografia das mensagens (Larson, et al., 2008). Mas em se tratando de rede CAN não é uma tarefa trivial, uma vez que o comprimento da mensagem permite somente 64 bits de dados. Yoshikawa (Yoshikawa, et al., 2013) tem uma proposta para contornar este inconveniente utilizando menor número de bist nas chaves de segurança, entretanto reduz-se também a segurança.

Jeong (Jeong, et al., 2014), apresenta uma implementação de algoritmos de embaralhamento como o S*ecure Hash Algorithm* (SHA-256), segundo o protocolo IEEE 1609.2 para comunicação veicular. A proposta apresentada usa processamento paralelo e criptografia por *hardware*.

Para muitas das alternativas apresentadas envolveria a modificação na arquitetura veicular da rede CAN para utilização da CAN-FD, que possui maior quantidade de dados, 64 bytes, viabilizando a criptografia.

### *2.5.2 Ataque do tipo Replay*

O ataque do tipo Replay consiste em repetir no barramento uma mensagem autêntica, conhecida, com o objetivo de se obter um comportamento esperado (Hoppe, et al., 2007). Para esse tipo de ataque, são monitoradas as mensagens que trafegavam na rede e observadas as condições do veículo, assim como é feito no ataque *Read*. Esta é a fase de reconhecimento e identificação do comportamento do veículo em função dos estímulos observados nas mensagens que trafegavam no barramento.

Uma vez descoberta a relação entre as mensagens CAN e o comportamento do veículo, basta replicar uma das mensagens escolhidas, mesmo não conhecendo o conteúdo da mesma, para que o veículo adquira o comportamento desejado ou prejudique o funcionamento normal.

*2.5.2.1. Consequências para o veículo para um ataque do tipo Replay*

É possível, utilizando ataque do tipo replay, esterçar a direção do veículo, desligar o motor, mudar o valor de indicação de velocidade, nível de combustível, inserir mensagens de falhas no painel de instrumentos do veículo, entre outras perturbações.

*2.5.2.2. Formas de evitar um ataque do tipo Replay*

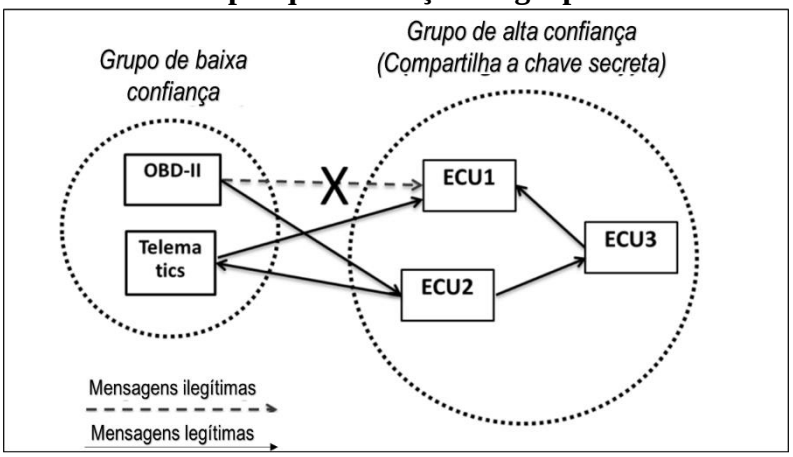
Em (Yoshikawa, et al., 2013), é proposto um sistema de criptografia de mensagens CAN para proteger de mensagens forjadas em ataques do tipo *spoofing* e *replay*. Devido às mensagens CAN possuírem somente 64 bits, as criptografias *Advanced Encryption Standard* (AES) e *Data Encryption Standard* (DES) não podem ser utilizados. O artigo propõe uma alternativa que utiliza menos bytes de dados. O método prevê um contador que deve ser incorporado à mensagem e controlado pelos módulos que as recebem e as enviam.

Em (Wang, et al., 2014), afirmam que a causa raiz para ataques é a falta de autenticação nos barramentos internos dos veículos e propõem uma estrutura de segurança para sistemas veiculares, denominada de VeCure. Wang se baseia nos trabalhos similares de (Szilagyi, et al., 2009) (Szilagyi, et al., 2010) (Lin, et al., 2012). Segundo ele o sistema é capaz de fazer autenticações off-line tendo sido desenhado para trabalhar mesmo em redes já existentes. O trabalho evidencia três desafios na autenticação de mensagens CAN:

a) Muitos módulos no veículo possuem um poder computacional limitado;
b) Cada mensagem CAN pode transportar apenas 8 bytes de dados;
c) Uma solução de autenticação deve ser escalar, uma vez que é crescente o número de módulos eletrônicos nos veículos.

Na proposta de autenticação, enquanto o grupo de baixa confiança ou *Low-trust group*, contém acesso ao mundo externo através de interfaces por fio ou sem fio, o grupo confiável ou *High-trust group* contém módulos que não possuem acesso a interfaces externas. Utiliza-se o grupo confiável para autenticar as mensagens (Figura 12). As mensagens devem ser enviadas em duas etapas porque existe um limite de 64 bits para uma mensagem CAN. Uma primeira mensagem envia os dados e uma segunda mensagem para autenticação com as informações necessárias para validar a primeira.

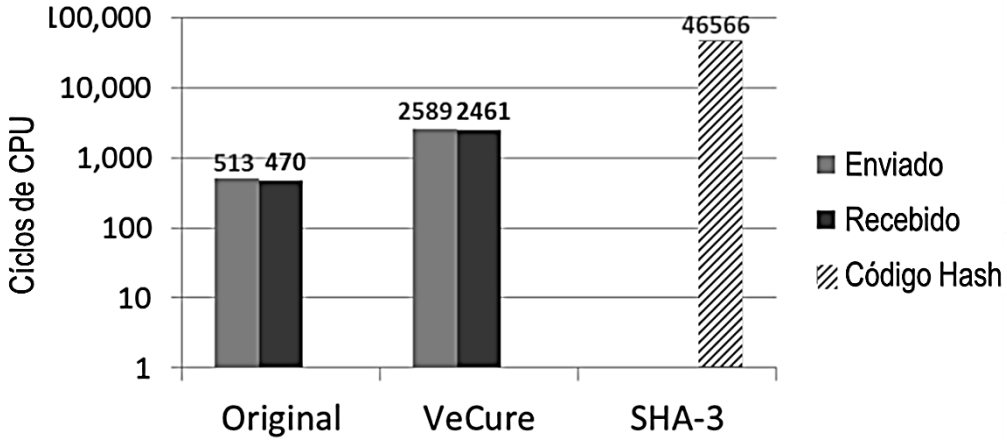
**Figura 12 - VeCure - Exemplo que formação de grupo confiável -Trust Groups**




**Fonte: Adaptado de (Wang, et al., 2014)**

Com a utilização do VeCure existe uma necessidade maior de processamento, entretanto menor que a utilização de um algoritmo de *hash*, como SHA-3 (Figura 13).

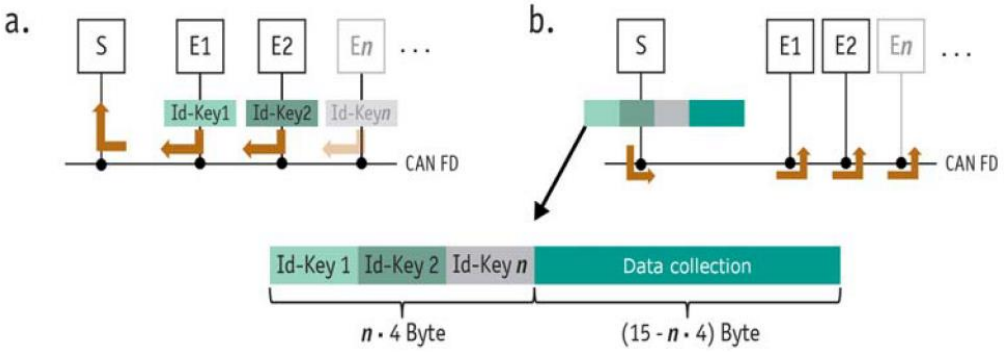
**Figura 13 - Comparação do VeCure com o uso do SHA-3 para autenticação de mensagens**




**Fonte: Adaptado de (Wang, et al., 2014)**

O trabalho de Happel (Happel, 2014), propõe a indexação de mensagens na rede CAN FD para prevenção de *replay attacks*, com a utilização de Id-Keys que funcionariam como contadores de mensagens, permitindo que o receptor descarte mensagens repetidas (Figura 14).

**Figura 14 - Introdução de Id-keys para sequenciar mensagens na rede**




**Fonte: (Happel, 2014)**

A solução proposta não pode ser usada na CAN tradicional, por causa do tamanho extra necessário na mensagem com a finalidade de introduzir os *Id-keys* para cada nó da rede. A ideia por trás dessa proposta é que cada nó controle seu próprio *Id-key*, rejeitando mensagens que estejam fora da sequência determinada.

### *2.5.3 Ataque do tipo Spoof*

É a geração de mensagens falsas, que se sobreponham às originais, levando os módulos a se comportarem de forma errada (Obaidat, et al., 2012). Os conhecimentos adquiridos em *Read* e a geração de ataques do tipo *Replay* podem evoluir ao ponto de escolher valores exatos que se queira enviar livremente em uma mensagem. Dessa forma, ao invés de simplesmente replicar as mensagens sem sequer saber seu valor, é possível criar integralmente novas mensagens e escolher com maior acurácia o momento de enviá-las, obtendo controle maior sobre as consequências desejadas em um ataque.

#### *2.5.3.1. Consequências para o veículo para um ataque do tipo Spoof*

Os resultados deste tipo de ataque tem um potencial de ser mais devastador que um ataque do tipo *Replay*, uma vez que exige um maior conhecimento do conteúdo das mensagens. Desta forma a intrusão pode ser cirúrgica modificando somente os sinais contidos na mensagem em que se quer atacar. Para os sistemas veiculares as mensagens serão tratadas como mensagens autênticas.

*2.5.3.2. Formas de evitar este tipo de ataque um ataque do tipo Spoof*

Em (Szilagyi, et al., 2009) (Groza, et al., 2011) (Obaidat, et al., 2012) (Szilagyi, et al., 2010) (Lin, et al., 2012) são propostas soluções para autenticação de mensagens no barramento CAN, mas todas elas apresentam deficiências, devido a *overhead* elevado, necessidade de maior processamento do micro controlador, se aplicar a comunicação *broadcast* ou *muticast*, entre outros. Também (Yoshikawa, et al., 2013), (Wang, et al., 2014) e (Jeong, et al., 2014) apresentam propostas de criptografia e autenticação que poderiam ser utilizadas para proteger as mensagens deste tipo de ataque. Em todas as propostas existe a limitação do tamanho máximo da mensagem, que pode ser contornado com adoção do CAN-FD.

### *2.5.4 Ataque do tipo DoS*

Segundo Nilson (Nilsson, et al., 2008) , o ataque do tipo DoS consiste em inundar uma rede com mensagens  de forma a impedir o seu trafego normal. Antes de avaliar o ataque do tipo DoS em veículos, vamos estudar como ocorre esse tipo de ataque em uma rede de computadores e conhecer o que a literatura sugere como formas de proteção para esse tipo de ataque. Esse conhecimento prévio é importante, mesmo sabendo de antemão que existe diferença na arquitetura de rede e protocolo da Internet quando comparado com a rede CAN. Da mesma forma, as soluções apresentadas para um tipo de rede não se aplicam diretamente à outra. Verifica-se que não existe uma solução efetiva que garanta 100% de proteção de uma rede de internet contra ataques do tipo DoS e não foi encontrada na literatura nenhuma solução para proteção de uma rede CAN contra um ataque do tipo DoS.

*2.5.4.1. Entendendo o ataque DoS*

Os sistemas de computadores são conectados à internet através de roteadores que ligam um computador a um servidor ou um computador a outro computador. Dessa forma, é possível acessar sites, transferir arquivos, trocar informações ou acessar dados. Para que um computador se conecte a internet ele precisa abrir portas de conexão. Essas mesmas portas permitem também a entrada de dados, em um caminho bidirecional. Usuários das redes, conhecendo sua estrutura e as fragilidades das portas, invadem sistemas e depositam um arquivo que torna um computador, que pode ser um servidor, em um zumbi, que irá obedecer a comandos remotos do seu criador ou estará programado para ser acionado automaticamente em um determinado

momento.

O ataque DoS consiste em vários computadores infectados com esse tipo de vírus, chamados de computadores zumbis tentarem acessar ao mesmo tempo um determinado site ou servidor. Um ataque coordenado gera um grande tráfego de rede ou ainda uma incapacidade do computador atacado de processar tantos acessos simultâneos. A consequência é uma indisponibilidade dos serviços do site, ou do computador alvo, uma vez que mensagens ou tentativas de acesso autênticos não conseguem trafegar na rede inundada. Daí o nome *Denied of Service*, ou Negação de Serviço.

Esse tipo de ataque é possível devido à deficiência estrutural da Internet, criando um enorme trafego de rede onde a vítima do ataque não precisa ser invadida, mas torna-se inoperante devido ao elevado trafego de rede (Xianjun Geng, 2000). Sobretudo em (Debra L. Cook, 2001) é ressaltado que esse tipo de ataque vem sendo negligenciado em ralação a estudos sobre disponibilidade de serviços, Zaroo (Zaroo, 2002) observa que existem três objetivos básicos para um ataque:

a) Ataques para explorar alguma vulnerabilidade ou *bug*, em um servidor ou site, para derruba-lo;
b) Ataques que usam todos os recursos disponíveis na máquina da vítima;
c) Ataques que consomem toda a largura de banda disponível para a máquina da vítima, que são os ataques mais fáceis e mais comuns.

Ressalta-se ainda que os ataques são possíveis devido à falta de segurança na internet, dificuldade de rotear o atacante de volta até a origem, uma vez que são usados pacotes de dados fabricados, onde as identificações do pacote são falsas. Outro fator se refere aos recursos de rede limitados e a um grande número de alternativas de hospedeiros para lançarem um ataque e maior facilidade em se quebrar um sistema do que em construí-los.

*2.5.4.2. Formas de combater um ataque DoS em redes LAN*

Não existe uma proteção efetiva a um ataque do tipo DoS. As proteções a esse tipo de ataque geralmente passam pela criação de cinturões de proteção. Essa proteção acontece nos roteadores próximos, desviando ou limitando o trafego sempre que for percebida uma movimentação acima do normal ou quando um ataque for detectado (Guangsen Zhang, 2006).

Segundo (Xianjun Geng, 2000), os especialistas concordam que uma solução a longo prazo para o ataque DoS seria incrementar o nível de segurança para todos os computadores ligados a internet, de forma a impedir que computadores sejam utilizados como zumbis. Entretanto, esse tipo de solução é inviável do ponto de vista econômico, principalmente porque os computadores zumbis não são prejudicados com o ataque e, dessa forma, seus usuários não veem a necessidade em investir em segurança.

Guangsen (Guangsen Zhang, 2006) ressalta que um ataque do tipo DoS não tem uma característica comum e por isso Sistemas de Detecção de Intrusão (IDS) não têm capacidade de detecta-los com precisão, sendo uma utopia pensar em uma internet 100% segura. Apresenta ainda algumas ferramentas para criar um ataque DoS, como *Trinoo, TFN, TFN2K, Shaft* e *Stachedldraht*.

**Alefiya (Alefiya Hussain, 2003) propõe uma estrutura, de forma a ajudar na criação de sistemas de proteção. Baseando-se em análise espectral é possível caracterizar se o ataque é do tipo DoS, que se caracteriza por ser um ataque de uma única fonte, ou se é do tipo DDoS, que consiste em ataque de múltiplas fontes ou vários computadores zumbis ao mesmo tempo.**

Quadro 4 – Tipos de proteção contra ataque DoS propostos na literatura apresenta algumas propostas encontradas na literatura para se prevenirem ataques do tipo DoS e DDoS.

**Quadro 4 – Tipos de proteção contra ataque DoS propostos na literatura**

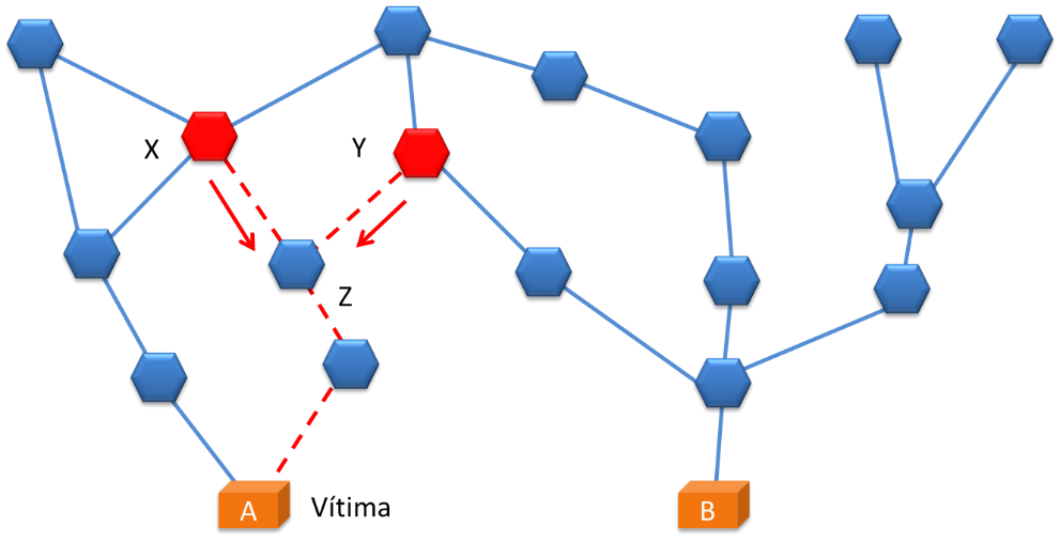
| Autor | Proposta para mitigar um ataque DoS |
|---|---|
| **(Xianjun Geng, 2000)** | Melhorar a segurança da internet para evitar que o computador se torne zumbi;<br>Utilizar filtros coordenados, barrando o ataque aos roteadores;<br>Rastrear o IP dos atacantes, embora geralmente os ataques usem IP's falsos. |
| **(Debra L. Cook, 2001)** | Utilizar uma arquitetura de *Secure Overlay Services* (SOS), que consiste em criar uma lista de usuários confiáveis e barrar usuários que não estejam nessa lista. |
| **(Zaroo, 2002)** | **Antes do ataque**: Prevenção, treinamento das pessoas, auditorias nas redes, *firewall*;<br>**Durante um ataque**: A rede deverá ser desenhada para ter caminhos alternativos que seriam utilizados durante um ataque;<br>**Após um ataque**: Deverá existir uma equipe para avaliar e tomar providencia, inclusive com cooperação das autoridades policiais. |
| **(Piskozub, 2002)** | Utilizar um filtro nos roteadores que aceitem pacotes catalogados e rejeitem pacotes com IP's desconhecidos ou suspeitos. |
| **(Angelos D. Keromytis, 2004)** | Utilizar uma arquitetura SOS para permitir uma comunicação entre um usuário confirmado e o objetivo, barrando comunicação não autenticada. |
| **(Guangsen Zhang, 2006)** | Criar uma rede de proteção cooperativa *gossip* onde os roteadores próximos bloqueariam pacotes com características suspeitas (Figura 06). |

**Fonte: compilação do autor**

A Figura 15 apresenta uma forma de proteção sugerida por (Guangsen Zhang, 2006), onde é criado um cinturão de proteção em que uma suspeita de ataque propagaria informação

para os roteadores próximos (*gossip* ou fofoca). Os roteadores então bloqueariam os pacotes com características suspeitas.

**Figura 15 – Estratégia de proteção *gossip***




**Fonte: Adaptado de (Guangsen Zhang, 2006)**

No caso da rede CAN não existem os *switches* segmentando a rede e a comunicação é feita diretamente de um nó a todos os outros que estejam em um mesmo barramento. É como uma internet sem roteadores e *switches*. Imagine que exista uma grande avenida sem esquinas e que as casas estão à direita ou a esquerda dessa avenida. Se houver um grande congestionamento em frente a uma das casas (casa A), não existirão rotas alternativas. Ninguém consegue chegar à casa A, assim como também não será possível chegar à casa B, do outro lado da cidade, porque a avenida é a mesma.

As proteções pensadas para a internet não se aplicam a rede CAN devido a dois fatores principais:

a) As mensagens CAN não têm um identificador de origem e destino das mensagens. Não se pode determinar de onde está vindo um ataque porque os pacotes não são endereçados. Todas os nós recebem ao mesmo tempo todas as mensagens do barramento, mesmo que não façam uso delas. É impossível rastrear quem enviou uma mensagem;
b) Não existem roteadores segmentando o caminho das mensagens. As mensagens são entregues diretamente ao barramento que liga todos os nós ao mesmo tempo. Não é

necessário um DDoS coordenado, basta um simples nó saturando o barramento para que todos os nós fiquem "isolados" de comunicação.

Embora seja potencialmente muito mais fácil lançar um ataque a um site na internet que em um veículo, pois, no caso da internet já existem ferramentas para fabricar ataques e a infraestrutura de rede está acessível a todos o tempo todo. O ataque a redes veiculares ainda não é tão acessível. Porém, uma vez que o acesso a rede é obtido, como foi visto anteriormente, existe um potencial devastador muito mais grave por colocar em risco a vida humana (carece referência).

*2.5.4.3. Escalonamento de CPU para definição de prioridade*

Com o objetivo de encontrar soluções já existentes para problemas análogos que pudessem ser utilizados para rede CAN, foram estudados algoritmos de escalonamento de CPU. A CPU utiliza este mecanismo para priorizar a execução na fila de tarefas. A questão que envolve diversas mensagens diferentes trafegando em um mesmo barramento com necessidade de se determinar a prioridade de acesso assemelha-se ao trabalho de uma CPU em determinar qual processo será executado em um mesmo processador.

O escalonamento de CPU é uma das principais tarefas dos sistemas operacionais multiprogramados (Silberschatz, et al., 2004). Enquanto em um sistema com um único processador somente um processo pode ser executado de cada vez, o uso da multiprogramação permite que vários processos sejam escalonados aproveitando da ociosidade da CPU para realizar várias tarefas concorrentemente.

Existem vários algoritmos de escalonamento e cada um deles apresenta características diferentes para decidir qual processo, na fila processos, terá atenção da CPU.

- Escalonamento *First Come First Server* (FCFS)

O critério é definido pela ordem de chegada dos processos na fila. É o mais simples, mas apresenta um tempo médio de espera alto.

- Escalonamento *Shortest Job First* (SJF)

Nesse algoritmo o critério é definido pelo processo que apresenta menor tempo de *burst* ou surto. Este processo apresenta, para determinados conjuntos de processos, um menor tempo médio de espera. O valor do surto é estimado com base em processos anteriores. O algoritmo SJF pode ser também preemptivo, o que significa que ao chegar um processo com um tempo de surto menor que o que está sendo executado irá interrompê-lo para ser executado primeiro,

retornando posteriormente seu processamento.

- Escalonamento *Round-Robin* (RR)

Nesse algoritmo é definida uma janela de tempo fixa e os processos são tratados como em uma fila circular. Cada processo é executado pelo tempo definido na janela, em ordem de chegada. Definir janelas muito grandes faz com que o comportamento desse algoritmo seja parecido com o FCFS. Uma janela muito estreita faz com que o tempo necessário pela CPU para trocar de um processo para outro, chamado de troca de contexto, fique significativo o bastante para ser levado em consideração.

- Escalonamento *Multilevel Queue*

Nessa classe de algoritmo é feita uma classificação dos processos e cada grupo possui uma prioridade diferente. Um grupo com prioridade mais baixa somente será atendido após o grupo com prioridade superior tiver atendido toda sua fila. Cada grupo pode ter um algoritmo de escalonamento diferente.

- Escalonamento *Multilevel Feedback-Queue*

A diferença desse algoritmo para o anterior é que um processo pode ser promovido para um grupo de maior prioridade e vice versa. Este algoritmo prioriza os processos com surtos menores.

- Escalonamento por prioridade

No algoritmo de escalonamento por prioridade são definidos pesos para os processos. Esses pesos definem qual processo terá prioridade na fila. O critério para distribuir os pesos é fundamental para determinar a prioridade dos processos. Um processo com peso muito baixo, dependendo do nível de solicitação da CPU, pode ficar aguardando muito tempo até que seja executado. Para evitar esse problema existe uma alternativa que é criar um fator de envelhecimento ou *aging* para os processos. Desta forma, o tempo de espera na fila irá promover sua prioridade até o ponto em que ele tenha prioridade suficiente para ser executado.

Nenhum dos algoritmos apresentados acima descreve corretamente a arbitragem para acesso ao barramento CAN, uma vez que na rede CAN não existe um escalonador gerenciando o acesso ao barramento. O escalonamento por prioridade é o que mais se assemelha, inclusive possuindo o mesmo problema. Processos com baixa prioridade poderão nunca ser executados em uma CPU muito solicitada. Na rede CAN, mensagens com prioridade baixa nunca acessarão o barramento se o *busload* permanecer alto. A solução adotada no escalonamento por prioridade é de promover a prioridade de processos que esperam por longo período de tempo. Esse tipo de solução não é viável em uma rede CAN porque não existe um gerenciador para contar o tempo

de espera de cada mensagem e também porque na rede CAN a prioridade é definida por um ID. O mesmo ID que determina a prioridade também traz a identificação da mensagem para que o seu destinatário a reconheça. O solução para conter um ataque DoS em uma rede CAN não pode depender de um gerenciador centralizado e não pode permitir a alteração do ID, pois indicaria uma mudança no significado da mensagem. Por outro lado, quando a CPU aumenta a prioridade de uma tarefa, ela atrasa a execução de outra, o que permite que todos os processos sejam executados. Este mecanismo poderia ser adaptado na rede CAN. Uma vez que não existe um controle centralizado, cada ECU deveria realizar seu próprio controle.

#### *2.5.4.4. Consequências para o veículo de um ataque DoS*

As consequências de um ataque do tipo DoS em um veículo são tão devastadoras quanto for o nível de tecnologia empregada no veículo. A descentralização das funções obriga que os sistemas se comuniquem para tomar decisões. Isso significa que um ataque DoS em uma rede CAN irá interromper completamente qualquer tipo de comunicação. Quanto mais distribuídos forem os sistemas e quanto mais dependentes de sensores e informações que trafegam pelo barramento CAN maior será a consequência.

A seguir são ilustrados alguns exemplos de sistemas distribuídos em veículos:

- Sistema de conforto

Alguns módulos de conforto podem precisar de informações como velocidade do veículo, temperatura, etc, que estão presentes em outros módulos e com isso comprometer o funcionamento de sistemas como travamento de portas com velocidade ou algoritmos que dependam da velocidade do veículo, como ar condicionado automático, e assim por diante.

- Sistemas de segurança

Da mesma forma, sistemas que dependam da rotação do motor, velocidade do veículo, estado das portas, etc, podem ter suas funcionalidades degradadas ou desabilitadas em caso de perda do barramento CAN. O sistema de localização pode ser inibido, e em uma situação de risco real pode não ser possível enviar uma localização adequada para envio de auxílio. Os sistemas de freios ou controle de estabilidade podem ser afetados, porque dependem de informações de dinâmica do veículo que vêm de outros módulos. Da mesma forma a assistência à direção também pode sofrer perdas que comprometam a segurança do condutor.

- Sistema de entretenimento

As condições do veículo, como localização, temperatura, etc, podem ser afetadas, reduzindo a prestação ou desabilitando completamente o funcionamento dos sistemas.

- Sistema de indicação

Todas as indicações do Painel de Instrumentos (PdI) dependem do barramento CAN. A falta do barramento pode significar a sinalização equivocada sobre a disponibilidade dos serviços que podem levar o usuário a ações que o coloquem em risco. Isso significa que mesmo que algum serviço esteja operando corretamente o PdI, devido a falta de informação, pode sinalizar a falta ou falha daquele sistema. Como resultado pode levar o condutor ao pânico ou ao descrédito do sistema, uma vez que um veículo que sinalize falha no sistema de freios, por exemplo, certamente irá chamar a atenção do condutor.

*2.5.4.5. Resultado dos ataques à rede CAN*

Dos cinco elementos que norteiam a política de segurança que devem ser considerados no desenvolvimento de uma rede segura (ABNT, 2005), é possível ferir todos eles de forma anônima e sem que o sistema tenha condições de detectar a anomalia ou tomar qualquer tipo de providência para coibir ou inibir os ataques.

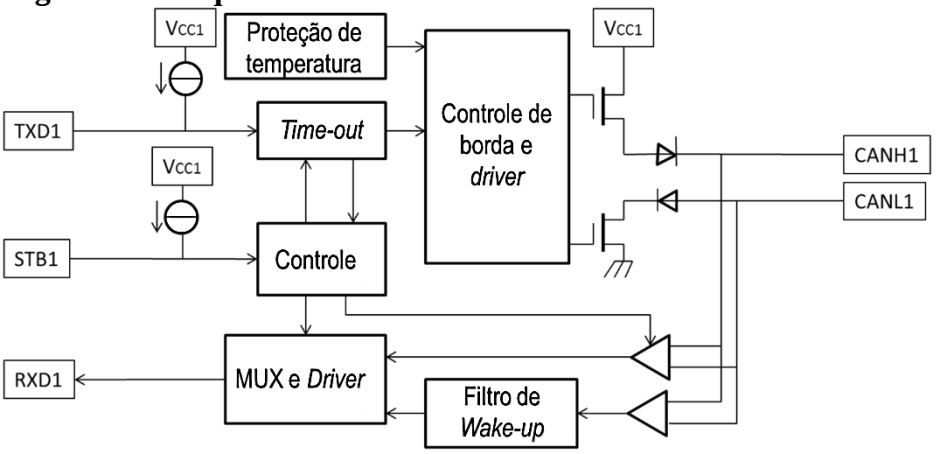
a) Disponibilidade: o sistema deve estar disponível de forma que quando o usuário necessitar possa utiliza-lo. Dados críticos devem estar disponíveis ininterruptamente;
b) Legalidade: as informações devem respeitar normas que a regulam. O protocolo CAN mesmo tendo obedecida sua estrutura formal não impede o sucesso do ataque, entretanto, os aspectos formais do conteúdo de uma mensagem podem ser completamente descaracterizados durante um ataque;
c) Integridade: o sistema deve estar sempre íntegro e em condições de ser usado. A integridade do sistema é violada durante o ataque e uma vez que os ataques cessaram o funcionamento normal voltou a ser estabelecido;
d) Autenticidade: o sistema deve ter condições de verificar a identidade dos usuários, e este terem condições de analisar a identidade do sistema. A rede CAN não oferece condição de identificar os emissores das mensagens ou uma forma de garantir o seu conteúdo, dessa forma, qualquer nó estranho pode enviar mensagens que serão consideradas como autênticas e qualquer nó, autêntico ou não, pode modificar o conteúdo, periodicidade de um mesmo ID de mensagem, sem nenhum tipo de controle;
e) Confidencialidade: dados privados devem ser apresentados somente aos donos dos dados ou ao grupo por eles liberado. Na rede CAN as mensagens enviadas não são cifradas. Existe uma tabela que forma um mapa, determinando todas as mensagens e

identificativos que são trocados entre os nós. Esta tabela pode ter seu conteúdo descoberto e reproduzido através de observação simples das entradas e saídas. Existem ferramentas no mercado para auxiliar nesse trabalho de engenharia reversa. Um ataque DoS não precisa saber o conteúdo das mensagens e nem gerar conteúdos válidos.

Para proteger a rede CAN contra ataques que comprometam a Confidencialidade, Integridade e Autenticidade, existem diversos trabalhos, como os citados no estudo de trabalhos relacionados. Não foram encontrados, contudo, estudos no sentido de proteger a rede CAN contra ataques do tipo DoS.

Não é possível através de software proteger o barramento CAN de um ataque DoS devido à natureza da concepção do barramento. Cada nó tem autonomia para escrever e liberdade para formatar a mensagem do modo que lhe convier, e essa mensagem acessa diretamente o barramento passando pelo transdutor ou *transceiver*. O *transceiver* faz a conexão entre a camada de aplicação e a camada física da rede. A Figura 16 demonstra o esquema interno de um *transceiver* CAN. Os Pinos TX e RX são responsáveis pelo trafego de informação do micro controlador para o barramento e vice-versa passando pelo *transceiver*. O *transceiver* recebe a mensagem através de uma linha TX e coloca a mensagem no barramento respeitando as formalidades do protocolo, níveis de tensão, velocidade e arbitragem. Da mesma forma, quando alguma mensagem chega ao *transceiver* este a decodifica e a envia ao micro controlador através da linha RX. O pino STB é utilizado para o micro controlador colocar o *transceiver* em modo de economia de energia, quando não é necessário enviar ou receber mensagem. Neste modo o *transceiver* irá "acordar" (*Wake-up*) caso perceba trafego de mensagens no barramento.

**Figura 16 – Esquema de um *transceiver* CAN-FD NXP TJA1046TK**




**Fonte: Adaptado de (NXP, 2016)**

Como foi observado nos capítulos anteriores existem diversas formas de acessar e causar danos a comunicação e aos sistemas eletrônicos de um veículo. É possível prejudicar o funcionamento normal dos módulos com potencial de alarmar, causar insegurança e até mesmo acidentes graves, colocando em risco não somente o bem material como também a integridade física e segurança dos ocupantes do veículo.

Da mesma forma foi possível observar que existem vários estudos sugerindo algum tipo de proteção para os ataques cibernéticos às redes veiculares, entretanto, não foram encontrados trabalhos voltados para a proteção de ataques do tipo DoS. Este motivo foi determinante para buscar soluções que pudessem contribuir para minimizar este tipo de ataque.

Nó próximo capítulo será apresentada a proposta de uma solução para resolver ou minimizar os danos em caso de um ataque do tipo DoS e serão apresentados testes feitos em ambiente de simulação para validar a proposta e a avaliação dos parâmetros que melhor atendem à proteção do barramento contra esse tipo de ataque.

# 3 SIMULAÇÃO DE ATAQUE D.O.S. A UMA REDE CAN VEICULAR

Neste capítulo será apresentada a ferramenta de simulação escolhida para a realização dos testes de rede e simulação de ataques ao barramento CAN. Serão apresentadas as justificativas para a escolha da ferramenta e a demonstração do funcionamento da mesma para gerar mensagens e ataques e como analisá-los.

## 3.1 Ferramenta para simulação de uma rede CAN

Para realizar as simulações de ataques ao barramento CAN será utilizada a ferramenta CANoe.

CANoe é uma ferramenta desenvolvida pela empresa Vector com o objetivo específico de simular sistemas de redes veiculares e apresenta uma série de vantagens em relação a outros ambientes de simulação. Podem ser acrescentados pacotes de expansão, como opção de simular outros tipos de barramentos além do CAN, ferramentas avançadas de análise, bibliotecas de utilização com AUTOSAR, que possuem a característica de automatizar camadas de programação de rede, entre outros.

Existem equipamentos de hardware que podem ser adquiridos para utilização em conjunto com o CANoe para serem utilizados na simulação de um sistema físico e lógico de um barramento real, ou mesmo integrar a simulação com um ambiente real de um veículo. Dessa forma é possível desenvolver módulos simulados e conecta-los em redes reais para testar os resultados, antes mesmo de desenvolver o hardware do módulo.

O CANoe possui as seguintes características:

a) É uma das ferramenta mais utilizadas pelas principais montadoras e fornecedores de componentes automotivos para monitoramento e validação de projetos reais;
b) Integra em uma só ferramenta todas as redes pretendidas para esse estudo: CAN e CAN-FD;
c) Possui capacidade de operar com bibliotecas externas (para criptografia, por exemplo), tornando o trabalho produtivo;
d) Possui módulos adicionais para modelagem de sinais, análise de dados e estatística;
e) Os resultados serão mais facilmente acreditados no mundo automotivo, uma vez que a ferramenta dispensa apresentações nesse meio.

O ambiente de simulação possui ferramentas para criação de mapas de mensagem, topologia de rede, definição de velocidade e comunicação e oferece total controle sobre a configuração do ambiente e de cada nó, com a definição de quais mensagens serão enviadas e qual o tamanho e periodicidade de cada uma.

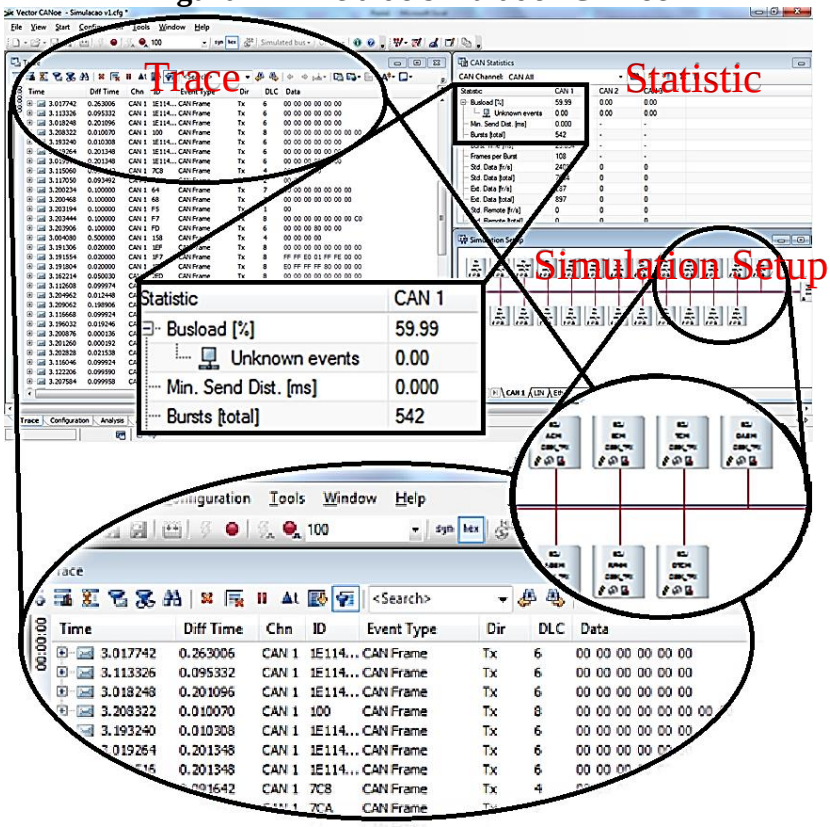
A Figura 17 apresenta uma das telas básicas de simulação, onde aparece à esquerda, na janela *Trace* uma lista com as mensagens que estão trafegando no barramento, com as seguintes informações:

a) *Time*: tempo absoluto de envio de cada mensagem;
b) *Diff Time:* delta tempo entre o envio de duas mensagens de mesmo ID;
c) *Chn:* representa em qual dos barramentos está trafegando a mensagem;
d) *ID:* identificativo da mensagem;
e) *Event Type:* tipo de evento relacionado ao protocolo CAN utilizado;
f) *Dir:* direção da mensagem, que pode ser Tx ou Rx;
g) *DLC:* tamanho da mensagem em bytes;
h) *Data:* são os dados da mensagem.

As informações que aparecem na janela *Trace* podem ser configuradas de forma a excluir ou adicionar novas colunas e mudar a ordem em que aparecem. Nesse exemplo de simulação os dados estão preenchidos com zeros porque nenhuma informação está sendo gerada para transmissão.

Do lado direito temos a janela de CAN *Statistics* onde são apresentadas algumas informações de gerenciamento do barramento, como *bussload*, número de pacotes enviados, erros de envio, etc. A janela *Simulation Setup* apresenta o esquema que representa o barramento, com os retângulos são os nós que estão conectados no barramento.

**Figura 17 – Tela do simulador CANoe**




**Fonte: do autor**

A Figura 18 apresenta uma tela do simulador CANoe com elementos que podem ser criados de forma gráfica para auxiliar na interação e visualização dos eventos que se desejam simular. Nessa imagem são apresentados painéis com imagens de *bitmap* que podem ser animadas de acordo com eventos que estão acontecendo nas mensagens que trafegam no barramento. Da mesma forma é possível interagir com as imagens e gerar eventos que irão criar ou modificar as mensagens no barramento ou variáveis internas. Podem ser utilizados gráficos para analisar as mensagens ou variáveis internas.

É possível também trabalhar em modo programado utilizando uma linguagem chamada *Capl*, que nada mais é que uma linguagem C customizada com comandos específicos para o ambiente de simulação CANoe. Através dessa linguagem é possível interagir com os eventos que estão acontecendo no barramento, criar interrupções, escrever mensagens, modificar mensagens, adicionar DLL's externas, criar *gateway* de mensagens entre barramentos diferentes e mais uma infinidade de possibilidades podem ser exploradas. Também é possível analisar os sinais que trafegam no barramento de forma gráfica. Para isso é necessário conhecer a conversão do sinal, ou seja, saber converter os bits em uma informação numérica.

**Figura 18- Exemplo de ambiente de simulação**


**Fonte: do autor**

## 3.2 Simulando um ataque DoS

Foi criado um ambiente de simulação composto por uma arquitetura formada por vários nós interligados. O barramento foi configurado para operar a 125kbps de velocidade. Foi utilizada a ferramenta do CANoe para criar um mapa de mensagens (MM) com dezenas de mensagens de diferentes tamanhos e ID's. Foi criado no MM um fluxo de trafego de mensagens entre os nós, com a definição dos nós que irão transmitir cada mensagem e a periodicidade das mesmas. (tabela: vel = 125kbps; tam = 64bits; F = 1ms)

Foi utilizada a biblioteca do CANoe para gerenciar o MM e assegurar o envio das mensagens de cada nó nos tempos definidos. Dessa forma foi possível observar um trafego de rede similar a um barramento real com recursos para calcular e medir os parâmetros necessários para a simulação.

Foi testada a condição de ataque utilizando uma ferramenta do CANoe chamada *Interactive Block Generator*. Essa ferramenta permite gerar mensagens, modificar mensagens e sinais internos e definir periodicidade de forma arbitrária. Então, foram feitos testes de ataque DoS e monitorados os resultados no simulador.

Para gerar o ataque foi utilizada uma mensagem de 64bits de dados, o tamanho máximo, e foi definida uma periodicidade igual ao tamanho de uma mensagem, ou seja, 1ms. Isso significa que a cada 1ms será enviada uma nova mensagem. Foi escolhido o ID 0x90 que é um ID menor que os utilizados no MM original, garantindo que nosso ataque terá sempre maior prioridade que as mensagens em condição normal.

Foi utilizado um dispositivo físico, também da Vector, chamado VN1610 para criar um barramento real, com base na simulação e desta forma foi possível medir e observar as mensagens CAN utilizando um osciloscópio analisador de protocolos da *Lecroy*. Uma mensagem CAN vista em um osciloscópio normal apresenta a forma de onda demonstrada na Figura 19 enquanto na Figura 20 são configuradas as informações físicas do barramento para que a função de análise de protocolo consiga reconhecer a forma de onda como sendo uma mensagem CAN. Como pode ser observado o osciloscópio distingue cada segmento da forma de onda, como identificativo, controle, dados, CRC, erros. Da mesmo forma são apresentados os valores hexadecimais lidos na mensagem. É fácil observar que existe uma variação recorrente de bit dentro de um mesmo intervalo de tempo, mesmo que o valor dos bits representem 0x00 (zero). Isto se deve ao mecanismo de *bitstuffing*, estudado anteriormente.

**Figura 19 – Mensagem CAN observada em Osciloscópio normal**


**Fonte: do autor**

**Figura 20 – Mensagem CAN em um Osciloscópio analisador de protocolo**


**Fonte: do autor**

Em um barramento com *busload* de cerca 14% existe um grande intervalo entre as mensagens (Figura 21). Quanto maior o *busload* menor será este intervalo. Segundo Corrigan

(Corrigan, 2008), recomenda-se que uma arquitetura de mensagens CAN seja desenhada para trabalhar com no máximo 60% de *busload* (Figura 22), garantindo que existam espaços suficientes para que todos os pacotes sejam entregues com o mínimo de atraso. O mecanismo de arbitragem da rede CAN garante que mensagens com ID menor terão prioridade de acesso e com isso terão garantia de menores atrasos em relação a mensagens menos prioritárias.

**Figura 21 – Barramento CAN com *busload* a 14,5%**

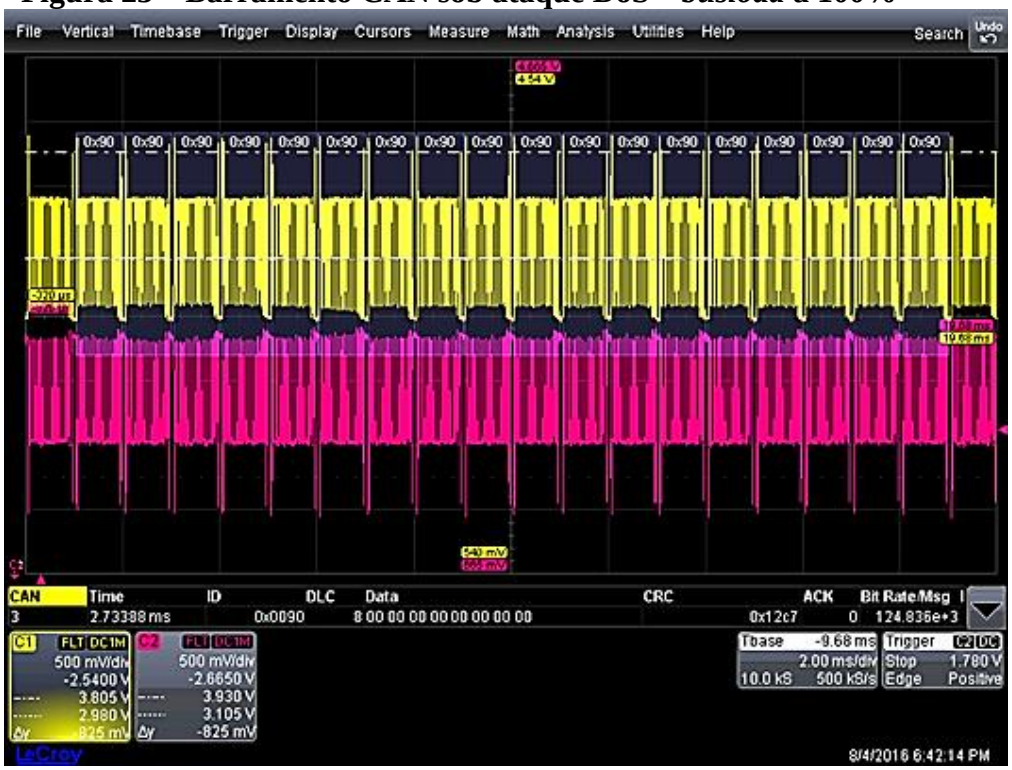

**Fonte: do autor**

**Figura 22 – Barramento CAN com *busload* a 60%**


**Fonte: do autor**

No caso de um ataque DoS ao barramento CAN (Figura 23), pode ser verificado que a única mensagem a ocupar o barramento é a mensagem de ataque com ID 0x90, que representa um ID prioritário em relação às demais mensagens. Nenhuma outra mensagem original consegue acessar o barramento, uma vez que o mecanismo de arbitragem prioriza o ID menor. Este tipo de ataque silencia completamente o barramento, uma vez que nenhuma mensagem autêntica irá chegar ao seu destino.

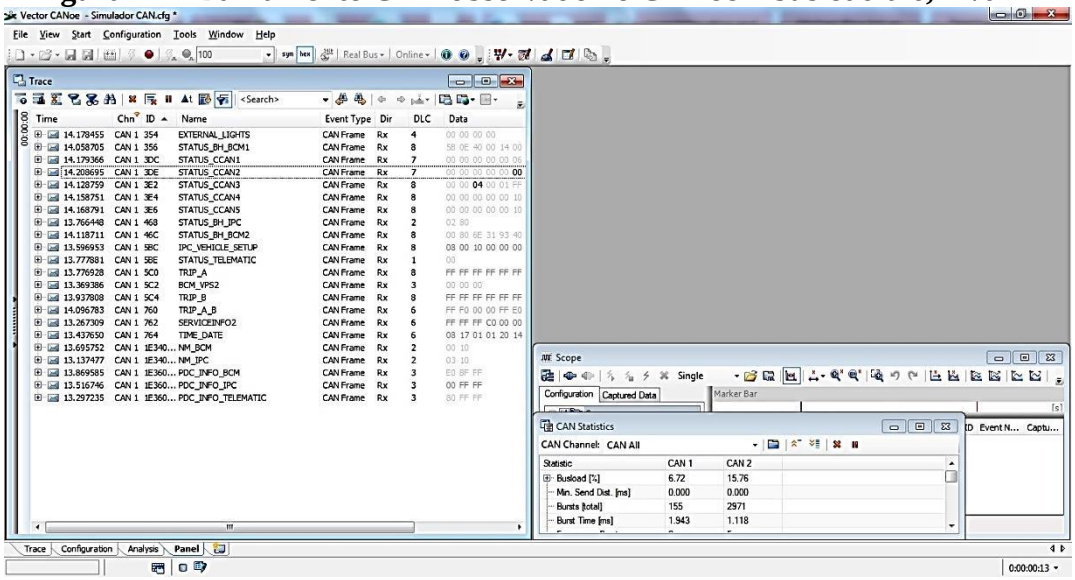
**Figura 23 – Barramento CAN sob ataque DoS – *busload* a 100%**




**Fonte: do autor**

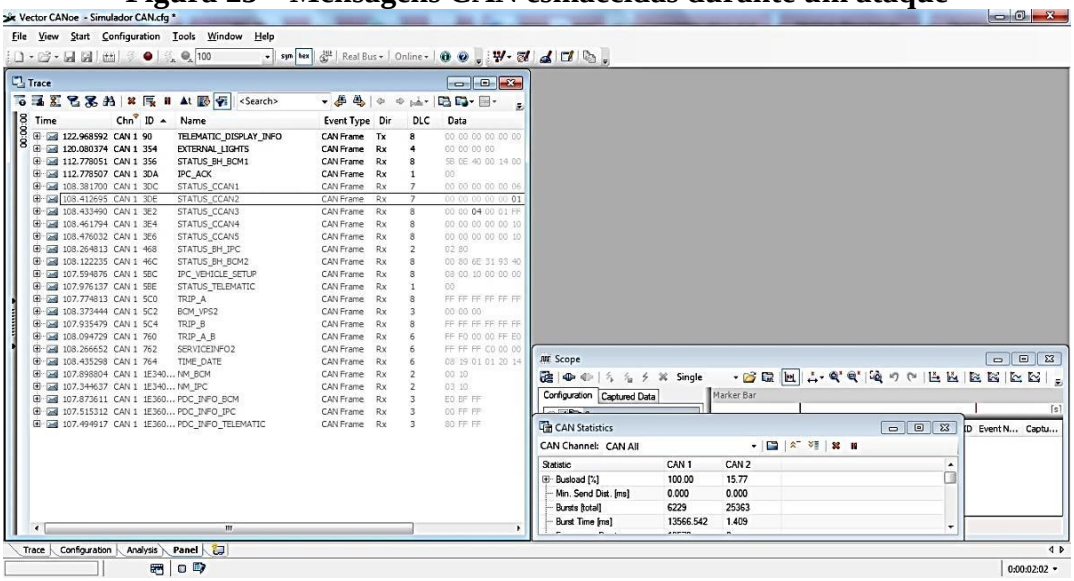
Podemos também observar o comportamento do barramento CAN através do simulador. Na Figura 24 é possível observar um barramento normal, onde todas as mensagens estão sendo transmitidas de forma correta enquanto na Figura 25 mostra o que acontece durante um ataque. As mensagens que não são atualizadas por alguns segundos têm o texto clareado pelo CANoe como uma forma de dar destaque às mensagens que estão sofrendo atualização. Como pode ser observado as mensagens de ID maior começam a ficarem apagadas. Se observados os tempos de transmissão, que representam a "idade" da mensagem fica fácil evidenciar esta situação. No caso de um ataque DoS a única mensagem a ocupar o barramento é a mensagem de ataque com ID 0x90. Como pode ser visto na Figura 26, todas as outras mensagens não conseguem acessar o barramento e com isso o simulador as mostra esmaecidas.

**Figura 24 – Barramento CAN observado no CANoe – busload a 6,72%**



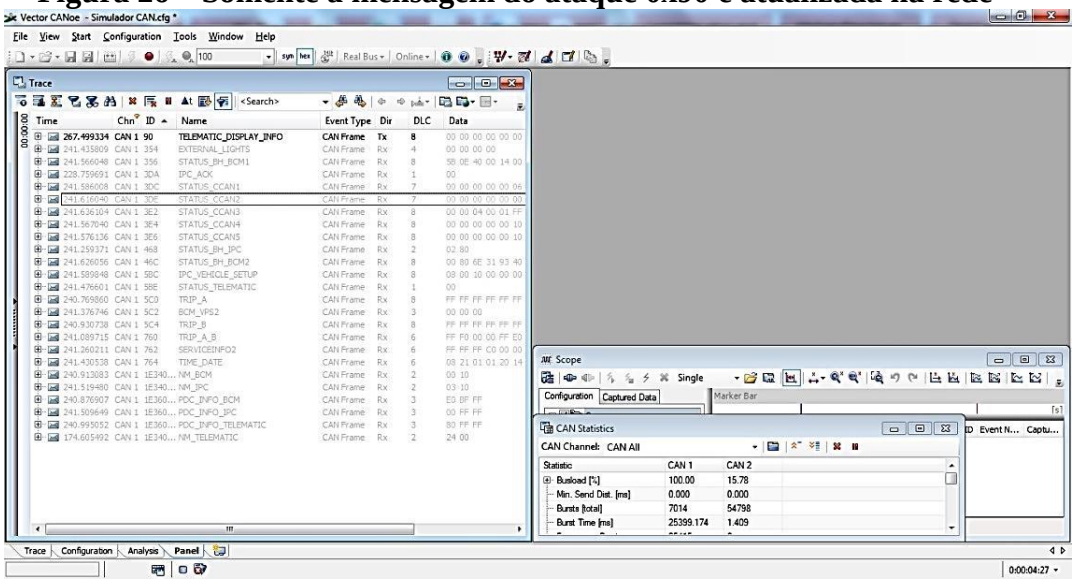

**Fonte: do autor**

**Figura 25 – Mensagens CAN esmaecidas durante um ataque**




**Fonte: do autor**

**Figura 26 – Somente a mensagem do ataque 0x90 é atualizada na rede**




**Fonte: do autor**

Nos testes realizados, apesar de todas as normas formais do protocolo CAN terem sido seguidas, o ataque utilizando mensagem com baixo ID e uma alta periodicidade gera um *busload* superior a 100%, impedindo que qualquer outra mensagem trafegue pelo barramento. Dessa forma, nenhum nó da rede consegue enviar seus pacotes a outros nós, causando indisponibilidade de diversas funções autenticas em um barramento.

Os testes foram repetidos em um barramento de 500kbps. Foram testados barramentos com *busload* inicial de 20% e 60%. Devido às características da rede CAN, descritas anteriormente, não foi testado *busload* superior a 60%. Foram testados tempos de periodicidades diferentes, onde a situação mais critica foi observada quando a periodicidade é igual ou menor que o tamanho da mensagem. Isto significa que a mensagem que está gerando o ataque está sempre concorrendo para acessar o barramento e sempre irá vencer. Tempos maiores representam uma lacuna entre duas mensagens de ataque o que dá tempo para que as mensagens autênticas tenham chance de disputar o barramento e com isso ocorreram perdas de pacotes das mensagens com ID's mais elevados.

Para facilitar a realização dos testes foi criado um painel, dentro do simulador, que permite configurar o ataque DoS. Desta forma é possível determinar o momento do início e fim do ataque, monitor o *busload*, monitorar os pacotes enviados e perdidos, determinar o ID da mensagem e o seu tamanho, escolher entre ID de 11 ou 29 bits. Independente da velocidade do barramento e do *busload* inicial, foi observado que os ataques foram sempre bem sucedidos.

# 4 PROTEÇÃO CONTRA ATAQUE D.O.S. SOBRE UMA REDE CAN E CAN-FD

Com base nos estudos realizados, foi possível observar que uma forma de se conter um ataque DoS em um barramento CAN é através da limitação do número de mensagens transmitidas por uma mesma ECU em um intervalo de tempo. Como não é possível fazer esse tipo de controle de forma centralizada, foi estudada uma maneira em que cada nó tenha seu próprio filtro de controle das mensagens enviadas.

## 4.1 Proposta de filtro de mensagens CAN para conter um ataque D.o.S.

Observando os testes realizados no capitulo anterior, fica evidente que a rede CAN apresenta uma fragilidade devido à forma em que foi projetada para funcionar. Quando se inicia um ataque DoS não existe um mecanismo para interrompê-lo. Não existem *switches*, como na internet ou gerenciadores centralizados operando o trafego de rede. Na rede CAN cada nó faz um autogerenciamento, baseando-se nas regras do protocolo estabelecidas pela norma ISO.

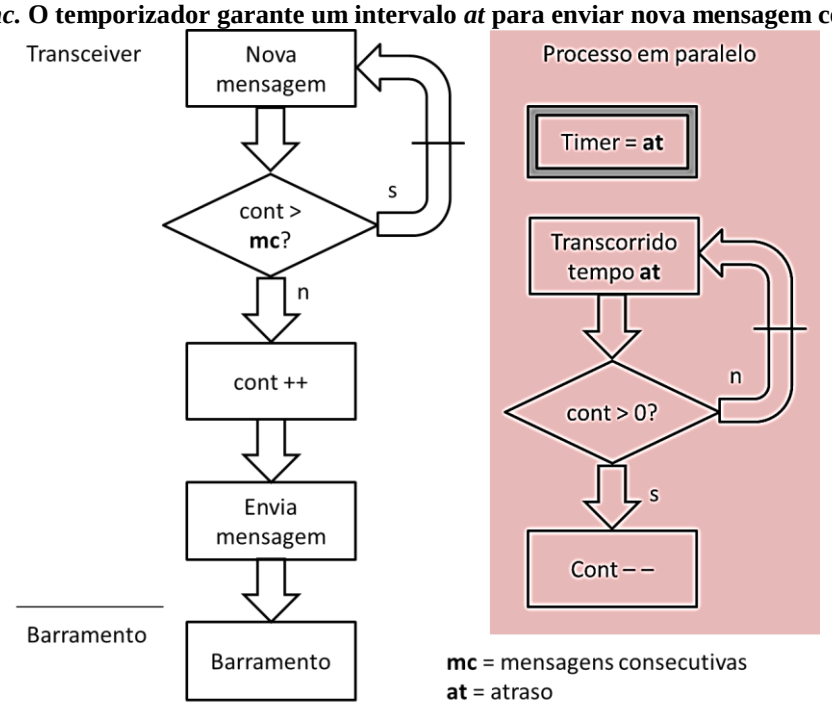
Já que cada nó é responsável pelo gerenciamento do acesso ao barramento e todo nó necessita de um *transceiver*, a proposta apresentada no presente trabalho usa por base o *transceiver* CAN para controlar o fluxo de mensagem durante um ataque. A forma que foi pensada se baseia em um controle de fluxo constante utilizando um filtro por *hardware* implementado no *transceiver* com capacidade para limitar o acesso ao barramento. Como todo nó possui um *transceiver*, é garantido que o acesso ao barramento será controlado. O filtro tem capacidade para limitar o número de mensagens que um nó irá transmitir dentro de um período de tempo, permitindo que, no caso de um ataque, os demais nós também tenham acesso ao barramento.

Cada vez que uma mensagem chega do micro-controlador para ser enviada, o *transceiver* deverá verificar o valor de um contador interno (Figura 27). Caso o contador esteja menor que um valor  definido como um parâmetro configurável do filtro a mensagem é transmitida e o contador é incrementado. Caso o contador seja maior que o valor estabelecido, a mensagem não será enviada e irá aguardar até que o valor do contador seja decrementado, permitindo que um número limitado de mensagens consecutivas sejam enviadas durante um ataque. A forma de decrementar o contador está relacionada com um temporizador configurado para contar um tempo *t*, que também é um parâmetro configurável. Cada vez que o temporizador atingir o tempo estabelecido ele irá decrementar o contador até atingir zero, dessa forma o

temporizador irá gerar um atraso na transmissão da mensagem durante um ataque. O controle de tempo é mais apropriado do que simplesmente contar o número de mensagens enviadas pelo fato do tempo se expirar, mesmo que o barramento não tenha novas mensagens de outros módulos, deixando a ECU pronta para novas transmissões.

**Figura 27- Algoritmo simplificado do mecanismo de filtro.**

**Envia mensagem ao barramento somente se o número de mensagens consecutivas for menor que o contador *mc*. O temporizador garante um intervalo *at* para enviar nova mensagem consecutiva.**



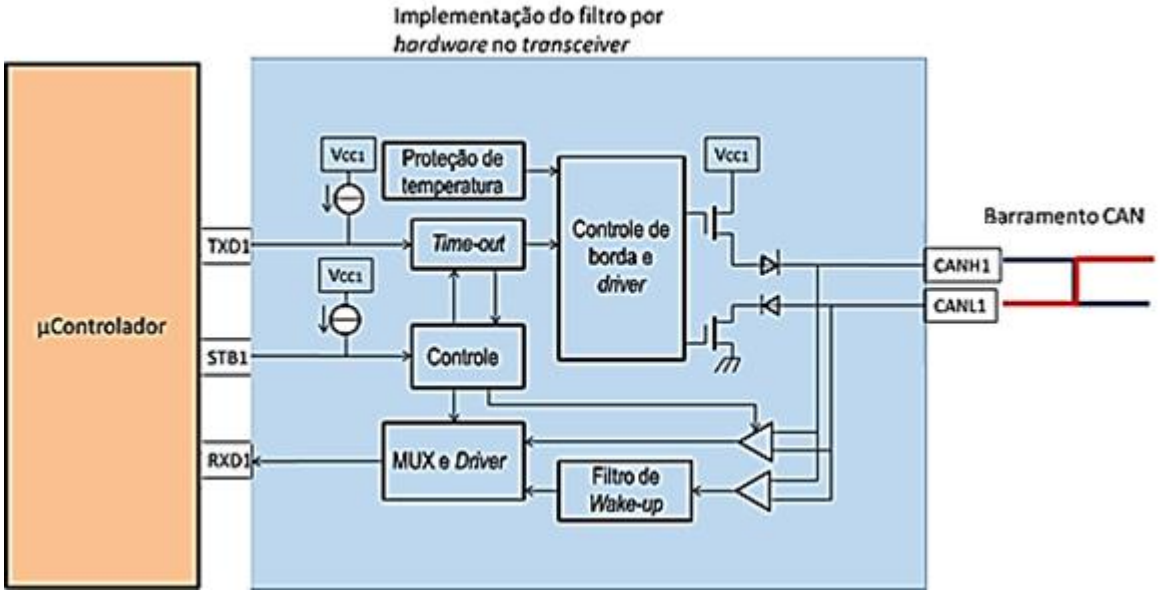

**Fonte: do autor**

Este mecanismo permite que mensagens que são transmitidas dentro de intervalos de comunicação normais, fora de um ataque, não tenham sua transmissão alterada pelo filtro, e com isso não ocorrerá perda de pacotes e nem atrasos na transmissão.

Em caso de um ataque DoS o nó que estiver promovendo o ataque terá suas mensagens submetidas ao controle do filtro e após enviar uma sequência de mensagens consecutivas dentro do limite de tempo estabelecido, deverá então aguardar um tempo de atraso antes de conseguir ter acesso novamente ao barramento. Esse tempo de espera permite que os demais nós possam disputar o acesso ao barramento e com isso as funções principais do veículo serão realizadas mesmo durante um ataque.

O algoritmo de filtro poderia ser incorporado tanto em um *transceiver* CAN quanto em um CAN-FD. A implementação pode ser feita por *software* mas seria mais indicado por *hardware* como visto na Figura 28, por dificultar a possibilidade de adulteração das configurações e pela velocidade de processamento,.

**Figura 28 – Proposta de adaptação de um *transceiver* CAN com adoção de um filtro de mensagens.**
**O filtro intercepta as mensagens antes de transmiti-las ao nível físico do barramento, limitando o fluxo de mensagens em caso de um ataque.**




**Fonte: adaptado de (NXP, 2016)**

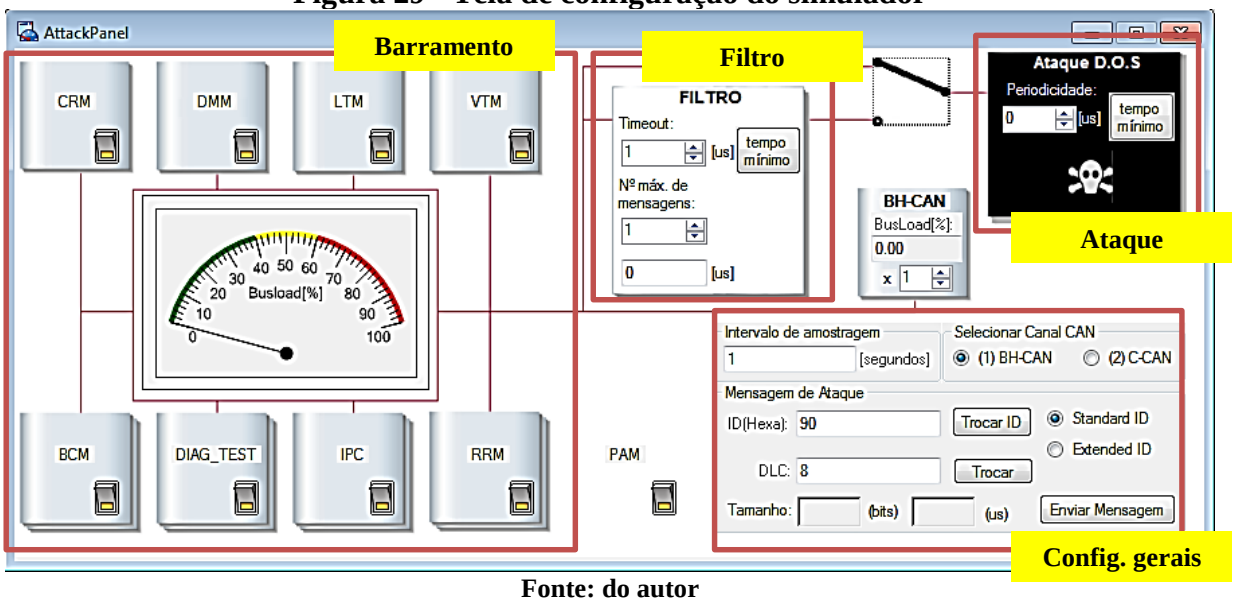
## 4.2 Testes em barramento simulado

Com o objetivo de testar o algoritmo apresentado, foi criado um simulador no ambiente CANoe. A Figura 29 apresenta a imagem da tela de configurações criada no CANoe para simular um barramento CAN e realizar os ataques e testes do algoritmo proposto. O painel criado apresenta, ao centro, um ponteiro que mede o *busload* do barramento, indicando sua saturação. Os parâmetros de configuração do simulador são:

- Ataque
    - Periodicidade da mensagem de ataque
    - ID da mensgem de ataque – com 11 ou 29 bits de identificativo (ID)
    - Ataque liga/desliga
    - Rede a ser atacada na simulação – BH-CAN ou C-CAN
    - Tamanho da mensagem – 0 a 8 bytes de dados (DLC) excluídos cabeçalho e bytes de controle
- Filtro
    - Tempo de atraso
    - Número de mensagens consecutivas
    - Filtro liga/desliga

- Gerais
    - Multiplicador de *busload* – para testar várias cargas de rede
    - Liga/Desliga cada nó simulado – variar carga da rede
    - Intervalo de amostragem – para contagem de pacotes enviados/perdidos

**Figura 29 - Tela de configuração do simulador**




**Fonte: do autor**

Com este simulador é possível testar as condições normais de funcionamento de uma rede CAN com barramento de alta velocidade (C-CAN) e baixa velocidade (BH-CAN). É possível ainda configurar diversos estados de *busloads*, representando situações de saturação de rede que serão encontradas em barramentos reais.

## 4.3 Parametrização

Os valores de configuração de número de mensagens consecutivas envidadas e tempo para decrementar o contador, gerando um atraso no próximo envio de mensagem pelo nó serão variados com o objetivo de encontrar um valor ideal onde se tenha o máximo de eficiência e o mínimo de perdas de pacotes. Dessa forma seria possível que o *transceiver* fosse programado de fábrica com um valor padrão de configuração que não possa ser alterado.

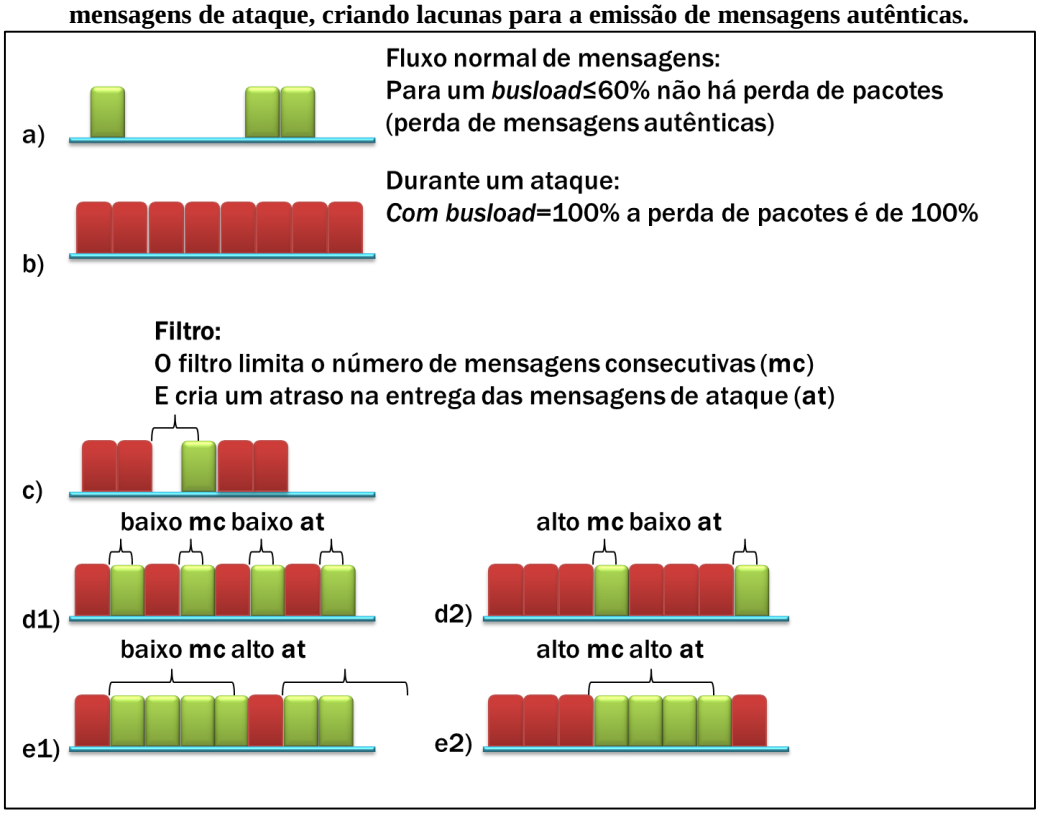
A Figura 30 apresenta um esquema para ilustrar a lógica de configuração do filtro. Podemos ver em (a) que durante um fluxo normal de mensagens o *busload* é menor que 60% e não ocorre perda de transmissão de pacotes. Sempre vão existir lacunas onde outros nós conseguem acesso ao barramento. Quando existe um ataque DoS no barramento somente a

mensagem de ataque ocupa o barramento e nenhum outro nó consegue acesso. A perda de mensagens autênticas é de 100% (b). O filtro proposto permite controlar o número de mensagens consecutivas que um nó pode transmitir e o intervalo de tempo de atraso até que este possa acessar o barramento novamente após ter atingido o número máximo de mensagens consecutivas (d), Os parâmetros de configuração podem ser ajustados para que se tenha um número baixo de mensagens consecutivas e um tempo mínimo de atraso (d1) ou um alto número de mensagens consecutivas, com um baixo tempo de atraso (d2) ou um baixo número de mensagens consecutivas e um alto tempo de atraso (e1) ou ainda um alto número de mensagens consecutivas e um alto tempo de atraso.

As medições no simulador têm o objetivo de encontrar os valores ótimos para os parâmetros que representem o mínimo de perda de pacotes com o mínimo de influência na comunicação normal do barramento.

**Figura 30 – Lógica de configuração do filtro**

**O filtro permite calibrar o número de mensagens consecutivas e cria um atraso na transmissão de mensagens de ataque, criando lacunas para a emissão de mensagens autênticas.**




**Fonte: do autor**

Os demais parâmetros de configuração do simulador serão utilizados para alterar as condições do barramento, buscando variar condições como *busload* inicial, tipo de rede CAN, ID da mensagem de ataque, tempo entre as mensagens de ataque.

Durante os testes foram coletados, primordialmente, dois tipos de informações: a taxa de *busload* da rede e perda de pacotes de mensagens. A taxa de *busload* é importante para analisar a eficiência do filtro em manter o barramento em condição de operação, buscando não variar o fluxo de mensagens comparado ao funcionamento normal. Da mesma forma, a informação de *busload* também ajudará a analisar as condições da rede durante um ataque e a influência do filtro na tentativa de estabelecer um controle no fluxo de mensagens.

As informações de perda de pacotes são importantes durante um ataque e durante a utilização do filtro, de forma a evidenciar os danos do ataque e se o filtro está desempenhando o papel de conter os danos advindos de um ataque.

O Experimento inicialmente foi configurada uma rede BH-CAN, que é uma CAN de 125kbps de velocidade, com um *busload* de ~10%. Foi medido o número de mensagens, ou pacotes transmitidos em condições normais. Foram feitas 5 medições e extraída a média de mensagens autênticas, transmitidas no período de 1 segundo. Na Tabela 1 – Exemplo de condição normal de uma rede B-CAN Tabela 1 pode ser observado o resultado de uma das medições. Os ID’s foram ordenados para que se ver ter uma noção da prioridade das mensagens. Pode-se observar que algumas mensagens têm periodicidade maior do que outras, e por isso são transmitidas mais vezes, dentro de 1 segundo de medição. Na tabela, o número de bytes representa quantos bytes totais foram enviados de cada mensagem. Isso quer dizer que a mensagem de ID 0x34E, por exemplo, tem um DLC de 12 e foi transmitida 6 vezes em um segundo, totalizando 18 bytes enviados.

**Tabela 1 – Exemplo de condição normal de uma rede B-CAN**

**BH-CAN 125kbps, *busload*=11%, média de mensagens normais enviadas=121, mensagens de ataque=zero**

| | | | | | |
|---|---|---|---|---|---|
| ID | 0x34E | nº de mensagens | 6 | Bytes | 12 |
| ID | 0x352 | nº de mensagens | 6 | Bytes | 48 |
| ID | 0x354 | nº de mensagens | 6 | Bytes | 24 |
| ID | 0x356 | nº de mensagens | 5 | Bytes | 40 |
| ID | 0x3DC | nº de mensagens | 12 | Bytes | 84 |
| ID | 0x3DE | nº de mensagens | 12 | Bytes | 84 |
| ID | 0x3E2 | nº de mensagens | 12 | Bytes | 96 |
| ID | 0x3E4 | nº de mensagens | 12 | Bytes | 96 |
| ID | 0x3E6 | nº de mensagens | 12 | Bytes | 96 |
| ID | 0x468 | nº de mensagens | 3 | Bytes | 6 |
| ID | 0x46C | nº de mensagens | 3 | Bytes | 24 |
| ID | 0x5BC | nº de mensagens | 3 | Bytes | 24 |
| ID | 0x5BE | nº de mensagens | 4 | Bytes | 4 |
| ID | 0x5C0 | nº de mensagens | 1 | Bytes | 8 |
| ID | 0x5C2 | nº de mensagens | 1 | Bytes | 3 |
| ID | 0x5C4 | nº de mensagens | 1 | Bytes | 8 |
| ID | 0x760 | nº de mensagens | 3 | Bytes | 18 |
| ID | 0x762 | nº de mensagens | 3 | Bytes | 18 |
| ID | 0x764 | nº de mensagens | 3 | Bytes | 18 |
| ID | 0x1E360000 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360003 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360005 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360008 | nº de mensagens | 1 | Bytes | 3 |
| ID | 0x1E360024 | nº de mensagens | 2 | Bytes | 6 |
| ID | 0x1E360031 | nº de mensagens | 1 | Bytes | 3 |

**Fonte: do autor**

Foi introduzido um ataque utilizando o ID 0x90, como sendo um ID menor que o menor ID utilizado nas mensagens que estavam sendo transmitidas originalmente. O tempo entre as mensagens de ataque foi configurado para o intervalo de 960µs, já que uma mensagem BH-CAN, a uma taxa de 125kbps tem este tempo de duração com DLC de 8 bytes de dados, como foi visto anteriormente na medição do osciloscópio. Esse ataque gerou um *busload* de 100% e uma perda de 100% dos pacotes autênticos, inutilizando completamente o barramento (Tabela 2). Dessa forma, nenhum nó podia se comunicar com os demais e serviços que dependessem de mensagens, comandos ou informações de outros nós estariam indisponíveis. Observa-se também que o número de pacotes ou mensagens, enviadas foi de 993, ao invés do 121, como mostrado na Tabela 1, que significa o número máximo de mensagens enfileiradas que podem trafegar no barramento no período de 1 segundo.

**Tabela 2– Exemplo de introdução de um ataque D.o.S. em uma rede B-CAN desprotegida**

**BH-CAN 125kbps, *busload*=100%, média de mensagens enviadas=zero, mensagens de ataque=993**

| | | | | | |
|---|---|---|---|---|---|
| ID | 0x90 | nº de mensagens | 993 | Bytes | 7944 |
| ID | 0x764 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x762 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x760 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x5C4 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x5C2 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x5C0 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x5BE | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x5BC | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x46C | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x468 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x3E6 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x3E4 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x3E2 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x3DE | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x3DC | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x356 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x354 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x352 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x34E | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360031 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360024 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360008 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360005 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360003 | nº de mensagens | 0 | Bytes | 0 |
| ID | 0x1E360000 | nº de mensagens | 0 | Bytes | 0 |

**Fonte: do autor**

Para demonstrar a influencia da periodicidade sobre o identificativo da mensagem, foi então realizado um teste de ataque com o ID 0x1E40000 que é maior que os ID's autênticos. Na Tabela 3 podemos observar que, a pesar do *busload* ter alcançado 100% de ocupação do barramento, não houve perda de pacotes e todas as 123 mensagens foram transmitidas com êxito e sem atrasos significativos, uma vez que todos os ID's autênticos têm prioridade sobre o ID do suposto ataque.

**Tabela 3 – Exemplo de ataque com um ID muito grande não representa ameaça**
**BH-CAN 125kbps, busload=100%, média de mensagens normais enviadas=123, mensagens de ataque=747**

| | | | | | |
|---|---|---|---|---|---|
| ID | 0x764 | nº de mensagens | 2 | Bytes | 12 |
| ID | 0x762 | nº de mensagens | 2 | Bytes | 12 |
| ID | 0x760 | nº de mensagens | 2 | Bytes | 12 |
| ID | 0x5C4 | nº de mensagens | 2 | Bytes | 16 |
| ID | 0x5C2 | nº de mensagens | 2 | Bytes | 6 |
| ID | 0x5C0 | nº de mensagens | 2 | Bytes | 16 |
| ID | 0x5BE | nº de mensagens | 3 | Bytes | 3 |
| ID | 0x5BC | nº de mensagens | 3 | Bytes | 24 |
| ID | 0x46C | nº de mensagens | 3 | Bytes | 24 |
| ID | 0x468 | nº de mensagens | 3 | Bytes | 6 |
| ID | 0x3E6 | nº de mensagens | 11 | Bytes | 88 |
| ID | 0x3E4 | nº de mensagens | 12 | Bytes | 96 |
| ID | 0x3E2 | nº de mensagens | 12 | Bytes | 96 |
| ID | 0x3DE | nº de mensagens | 12 | Bytes | 84 |
| ID | 0x3DC | nº de mensagens | 12 | Bytes | 84 |
| ID | 0x356 | nº de mensagens | 6 | Bytes | 48 |
| ID | 0x354 | nº de mensagens | 6 | Bytes | 24 |
| ID | 0x352 | nº de mensagens | 6 | Bytes | 48 |
| ID | 0x34E | nº de mensagens | 6 | Bytes | 12 |
| ID | 0x1E360031 | nº de mensagens | 2 | Bytes | 6 |
| ID | 0x1E360024 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360008 | nº de mensagens | 2 | Bytes | 6 |
| ID | 0x1E360005 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360003 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E360000 | nº de mensagens | 3 | Bytes | 9 |
| ID | 0x1E400000 | nº de mensagens | 747 | Bytes | 5976 |

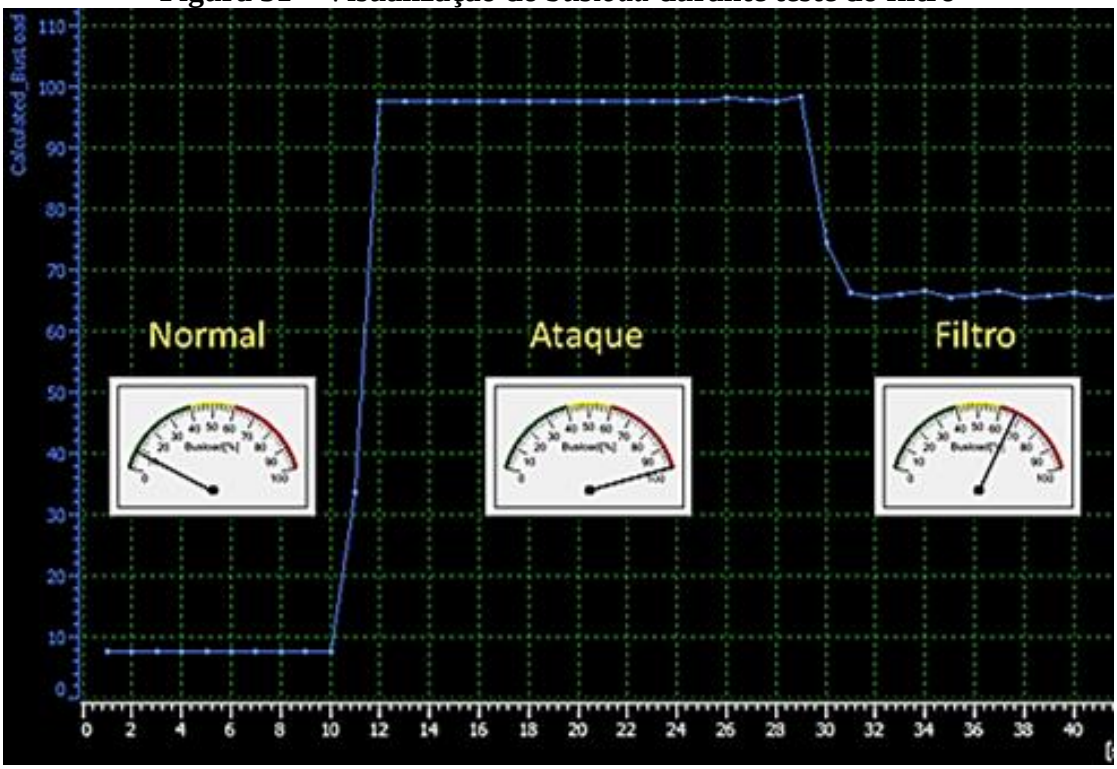
**Fonte: do autor**

Para verificar o funcionamento do filtro, foi feito um teste preliminar (Figura 31). Como pode ser observado, o teste foi feito em três estágios:

- **Condição normal de funcionamento** – Neste estágio o simulador foi configurado para uma comunicação normal. Pode-se observar que o *busload* é de 7,7%;
- **Ataque ao barramento** – Neste estágio foi desencadeado um ataque de forma a corromper o barramento, com ID 0x90, periodicidade de ataque de 960µs e DLC de 8 bytes, alcançando o *busload* de 100% e nenhum pacote autêntico foi transmitido ou seja, perda de 100% da comunicação, conforme visto anteriormente;
- **Controle do tráfego através do filtro** – Neste estágio foi ativado o filtro proposto utilizando parâmetro de 100µs de atraso entre mensagens e permitindo

somente 1 mensagem consecutiva. Pode-se observar que o *busload* se estabilizou em cerca de 67% condição suficiente para que a comunicação autêntica se estabeleça, controlando o ataque.

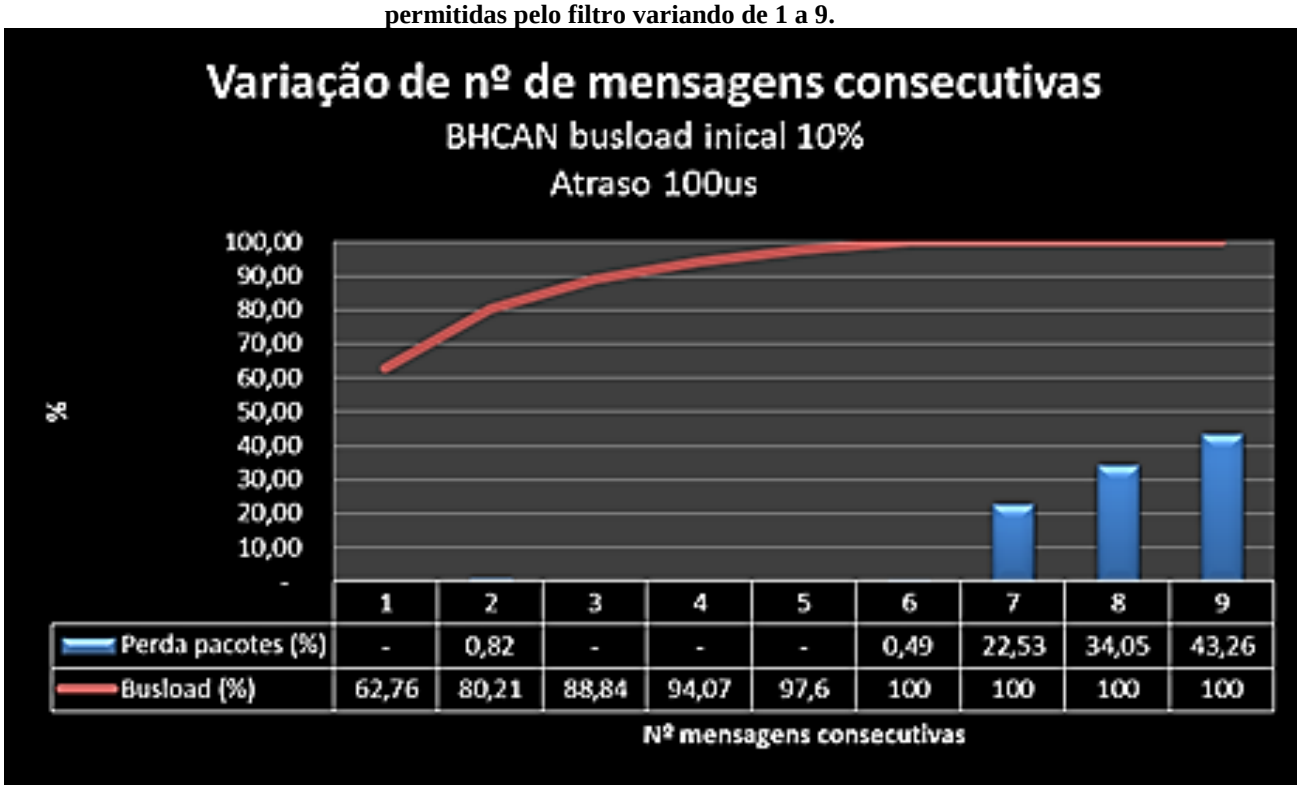
**Figura 31 – Visualização do *busload* durante teste do filtro**




**Fonte: do autor**

Foi também feito um teste considerando o barramento BHCAN com 125kbps e *busload* inicial de 10%, com atraso de mensagem de 100µs e variando-se o número de mensagens consecutivas de 1 a 9. O resultado, Figura 32, pode ser observado que o número de mensagens consecutivas tem influencia direta no *busload* e na perda de pacotes durante um ataque. Um maior número de mensagens consecutivas que o nó envia irá gerar uma maior taxa de *busload* durante um ataque e maior será o número de pacotes perdidos.

**Figura 32 - Teste do filtro com variação do nº de mensagens consecutivas**
**Perda de pacotes (%) e taxa de *busload* (%) em função da variação do número de mensagens consecutivas permitidas pelo filtro variando de 1 a 9.**




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Perda pacotes (%) | - | 0,82 | - | - | - | 0,49 | 22,53 | 34,05 | 43,26 |
| Busload (%) | 62,76 | 80,21 | 88,84 | 94,07 | 97,6 | 100 | 100 | 100 | 100 |

**Fonte: do autor**

Foi observado que durante um ataque, utilizando o filtro, em que há perda de pacotes havia uma expectativa da perda de pacotes igualmente distribuída entre os ID's. O que foi observado, no entanto, é que com a sobrecarga no *busload* e com o mecanismo de prioridade de acesso ao barramento, houve uma linha de corte a partir de um determinado valor de ID. Desta forma, ou um ID não era impactado pelo ataque, e tinha todos os seus pacotes entregues, ou um ID era totalmente comprometido e nenhum pacote era transmitido. A exceção fica por conta do ID que está no limiar da saturação do barramento onde pode ocorrer ou não perdas. A Tabela 4 mostra as perdas de pacotes por ID, durante um ataque, onde o ID 0x3E2 ficou na linha de corte e teve parte de seus pacotes perdidos, entretanto as mensagens de ID menor tiveram prioridade mais alta e foram entregues sem perdas e os ID's maiores, consequentemente com menor prioridade, tiveram perda de 100% dos pacotes.

**Tabela 4 – Perda de pacotes durante um ataque**

**BH-CAN 125kbps, *busload*=100%, média de mensagens normais enviadas=68, mensagens de ataque=933**

| | | | | | |
|---|---|---|---|---|---|
| ID: | 0x90 | nº de mensagens: | 939 | Bytes: | 7512 |
| ID: | 0x34E | nº de mensagens: | 8 | Bytes: | 16 |
| ID: | 0x352 | nº de mensagens: | 8 | Bytes: | 64 |
| ID: | 0x354 | nº de mensagens: | 8 | Bytes: | 32 |
| ID: | 0x356 | nº de mensagens: | 8 | Bytes: | 64 |
| ID: | 0x3DC | nº de mensagens: | 14 | Bytes: | 98 |
| ID: | 0x3DE | nº de mensagens: | 14 | Bytes: | 98 |
| ID: | 0x3E2 | nº de mensagens: | 8 | Bytes: | 64 |
| ID: | 0x3E4 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x3E6 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x468 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x46C | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x5BC | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x5BE | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x5C0 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x5C2 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x5C4 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x760 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x762 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x764 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360000 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360003 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360005 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360008 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360024 | nº de mensagens: | 0 | Bytes: | 0 |
| ID: | 0x1E360031 | nº de mensagens: | 0 | Bytes: | 0 |

**Fonte: do autor**

## 4.4 Ajustes do filtro de mensagens

Para conseguir um valor ótimo de calibração dos parâmetros do filtro foram testadas diversas condições, variando-se o *busload* inicial, tempo de espera, número de mensagens consecutivas e velocidade do barramento. Ao todo foram feitas 21 variações de tempos de atraso, 9 variações de mensagens consecutivas para 3 variações de *busload* iniciais, para B-CAN. Essas medições foram repetidas para C-CAN com 13 variações de tempos, 9 variações de mensagens consecutivas para 2 variações de busload iniciais, repetindo-se as medições 5 vezes para se obter a média de pacotes, perfazendo um total de 4005 medições.

Os valores na Tabela 5, reúnem um extrato com os resultados para BH-CAN com *busload* inicial a 10%. As colunas representam a variação do intervalo de atraso nas mensagens de ataque, enquanto as linhas representam a variação no número de mensagens consecutivas. O

primeiro grupo de dados, em azul, reúne os resultados medidos de *busload* com a aplicação do filtro seguindo os parâmetros das linhas e colunas. O segundo grupo de dados, em vermelho, mede o número de pacotes autênticos transmitidos. O terceiro e último grupo de dados, em verde, medem a perda de pacotes, ou seja, é o inverso da dos dados do segundo grupo. As medições completas estão disponíveis no Anexo A onde são apresentadas as tabelas pra BH-CAN com *busload* de 20% e 60%, além de testes na C-CAN, para *busload* de 20% e 60%.

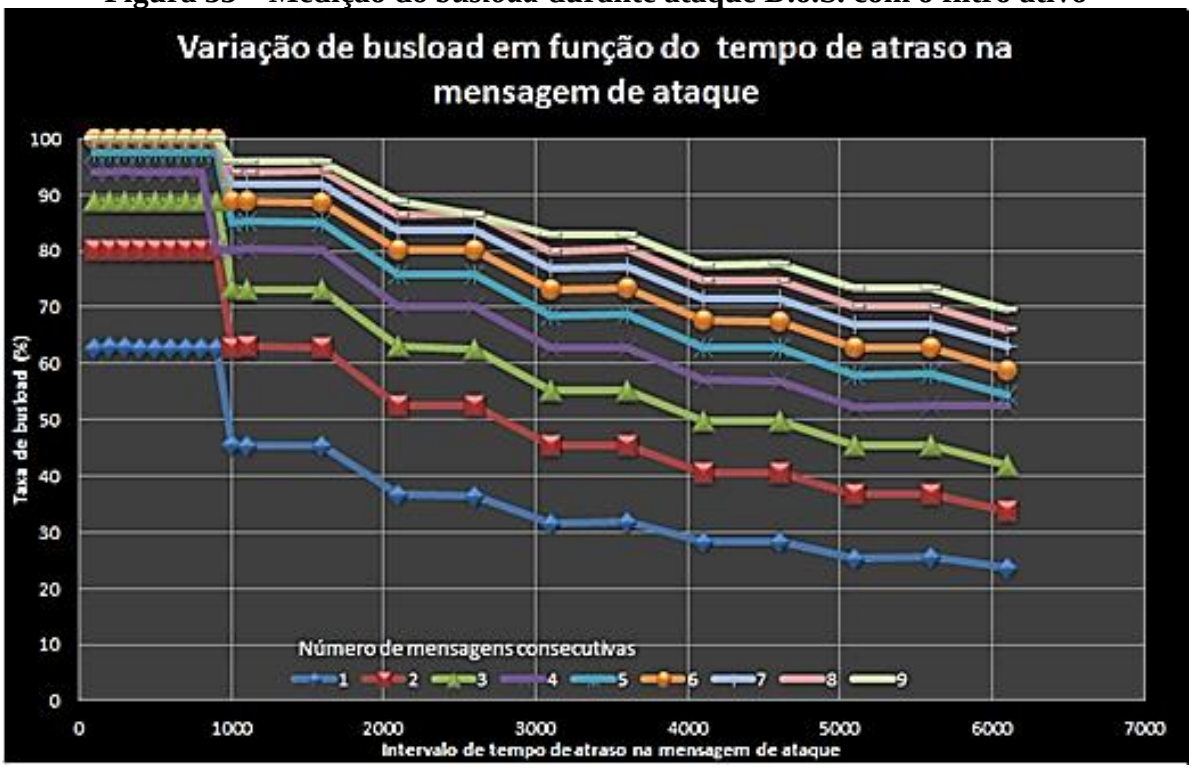
**Tabela 5 – Extração dos valores de *busload* e perda de pacotes para BH-CAN com *busload* inicial de 10%**

| | | Tempo de atraso (µsegundos) | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mensagens Consecutivas | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
| Variação de busload (%) | 1 | 62,76 | 62,94 | 62,94 | 62,84 | 62,88 | 62,73 | 62,77 | 62,85 | 62,77 | 45,58 | 45,54 | 45,47 | 36,78 | 36,62 | 31,60 | 31,78 | 28,28 | 28,28 | 25,27 | 25,68 | 23,69 |
| | 2 | 80,21 | 80,16 | 80,18 | 80,19 | 80,28 | 80,06 | 80,25 | 80,09 | 80,22 | 62,90 | 63,03 | 62,89 | 52,41 | 52,46 | 45,52 | 45,32 | 40,47 | 40,56 | 36,76 | 36,64 | 33,90 |
| | 3 | 88,84 | 88,85 | 88,87 | 88,85 | 88,94 | 88,76 | 88,88 | 88,88 | 88,95 | 73,36 | 73,22 | 73,22 | 63,05 | 62,64 | 55,31 | 55,34 | 49,78 | 49,80 | 45,41 | 45,40 | 41,96 |
| | 4 | 94,07 | 94,02 | 94,20 | 94,08 | 94,13 | 94,08 | 94,12 | 94,06 | 80,13 | 80,13 | 80,33 | 80,17 | 70,10 | 70,23 | 62,90 | 62,78 | 57,14 | 56,96 | 52,35 | 52,60 | 52,60 |
| | 5 | 97,60 | 97,54 | 97,54 | 97,51 | 97,51 | 97,52 | 97,55 | 97,60 | 97,60 | 85,11 | 85,24 | 85,15 | 75,95 | 75,86 | 68,59 | 68,78 | 62,74 | 62,78 | 57,95 | 58,13 | 54,26 |
| | 6 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 88,92 | 88,80 | 88,66 | 80,11 | 80,29 | 73,09 | 73,42 | 67,63 | 67,32 | 62,83 | 62,85 | 58,79 |
| | 7 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 91,77 | 91,84 | 91,71 | 83,76 | 83,60 | 77,02 | 77,10 | 71,47 | 71,46 | 66,84 | 66,78 | 63,00 |
| | 8 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 94,05 | 94,06 | 94,16 | 86,43 | 86,62 | 80,03 | 80,44 | 74,87 | 74,86 | 70,06 | 70,15 | 66,07 |
| | 9 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 95,97 | 96,02 | 95,91 | 88,95 | 86,40 | 82,84 | 82,85 | 77,54 | 77,85 | 73,28 | 73,25 | 69,47 |
| Número de pacotes enviados | 1 | 122,00 | 120,80 | 122,20 | 120,80 | 122,20 | 121,20 | 120,60 | 122,40 | 120,60 | 122,00 | 122,20 | 121,80 | 121,40 | 120,60 | 122,20 | 120,80 | 122,00 | 121,80 | 120,60 | 122,40 | 121,60 |
| | 2 | 120,60 | 122,20 | 121,60 | 121,60 | 122,20 | 120,60 | 122,20 | 119,60 | 121,80 | 122,00 | 122,20 | 122,00 | 120,60 | 122,00 | 122,20 | 122,20 | 120,60 | 121,80 | 122,00 | 120,20 | 122,00 |
| | 3 | 121,80 | 121,00 | 121,80 | 121,00 | 122,40 | 121,20 | 122,00 | 120,80 | 122,40 | 121,20 | 122,40 | 122,20 | 120,40 | 120,20 | 119,40 | 122,20 | 120,60 | 120,60 | 122,00 | 122,00 | 121,80 |
| | 4 | 122,20 | 122,20 | 119,60 | 122,20 | 122,20 | 120,60 | 120,60 | 122,20 | 120,00 | 122,20 | 122,20 | 120,20 | 120,60 | 122,20 | 122,00 | 121,20 | 122,40 | 120,60 | 121,60 | 121,20 | 122,00 |
| | 5 | 121,80 | 122,00 | 122,00 | 122,20 | 122,20 | 121,00 | 120,60 | 121,60 | 122,20 | 122,00 | 122,00 | 121,80 | 121,20 | 122,20 | 122,40 | 122,20 | 122,20 | 120,60 | 120,20 | 120,40 | 121,80 |
| | 6 | 121,00 | 120,80 | 120,40 | 121,20 | 120,60 | 121,60 | 121,20 | 121,20 | 121,20 | 121,40 | 122,00 | 120,00 | 122,00 | 122,20 | 120,60 | 120,00 | 122,20 | 120,60 | 120,40 | 121,80 | 120,40 |
| | 7 | 94,20 | 94,20 | 94,40 | 94,20 | 94,40 | 94,40 | 94,40 | 94,40 | 94,00 | 122,20 | 122,00 | 122,20 | 121,80 | 122,20 | 122,00 | 122,00 | 120,40 | 122,40 | 120,40 | 120,40 | 122,20 |
| | 8 | 80,20 | 80,20 | 80,40 | 80,40 | 79,80 | 80,20 | 80,20 | 80,00 | 80,00 | 122,00 | 122,00 | 122,20 | 122,20 | 122,20 | 122,00 | 121,00 | 121,40 | 121,60 | 122,20 | 120,20 | 122,00 |
| | 9 | 69,00 | 68,80 | 68,80 | 69,00 | 68,80 | 69,20 | 69,20 | 69,00 | 68,80 | 122,00 | 122,00 | 122,00 | 122,00 | 122,20 | 120,80 | 122,20 | 122,20 | 122,00 | 120,60 | 122,00 | 122,00 |
| Perda de pacortes (%) | 1 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | 2 | - | - | - | - | - | - | - | 0,3 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | 3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | 0,5 | - | - | - | - | - | - |
| | 4 | - | - | 0,3 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | 5 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | 6 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | 7 | 21,5 | 21,5 | 21,3 | 21,5 | 21,3 | 21,3 | 21,3 | 21,3 | 21,7 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 8 | 33,2 | 33,2 | 33,0 | 33,0 | 33,5 | 33,2 | 33,2 | 33,3 | 33,3 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 9 | 42,5 | 42,7 | 42,7 | 42,5 | 42,7 | 42,3 | 42,3 | 42,5 | 42,7 | - | - | - | - | - | - | - | - | - | - | - | - |

**Fonte: do autor**

Para representar melhor os valores encontrados na Tabela 5, foi plotado o gráfico, Figura 33. No eixo vertical é apresentada a taxa de *busload* em função do tempo de atraso nas mensagens de ataque. Cada curva representa a variação do número de mensagens consecutivas. A variação no tempo de atraso foi iniciada com intervalo de 100µs. Como pode ser observada esta variação só surtiu efeito na taxa do *busload* após 1.000µs, que representa o comprimento de uma mensagem. Como este teste foi realizado em uma BH-CAN de 125kbps e o comprimento total da mensagem é de 64 bits de cabeçalho + 64 bits de dados para identificativos CAN estendidos (Figura 5), resultando em um comprimento de mensagem em torno de 976µs. Devido a esta característica medida, passou-se a trabalhar com intervalos de 500µs no tempo de atraso, ou seja, 50% do tamanho de uma mensagem.

**Figura 33 – Medição do *busload* durante ataque D.o.S. com o filtro ativo**

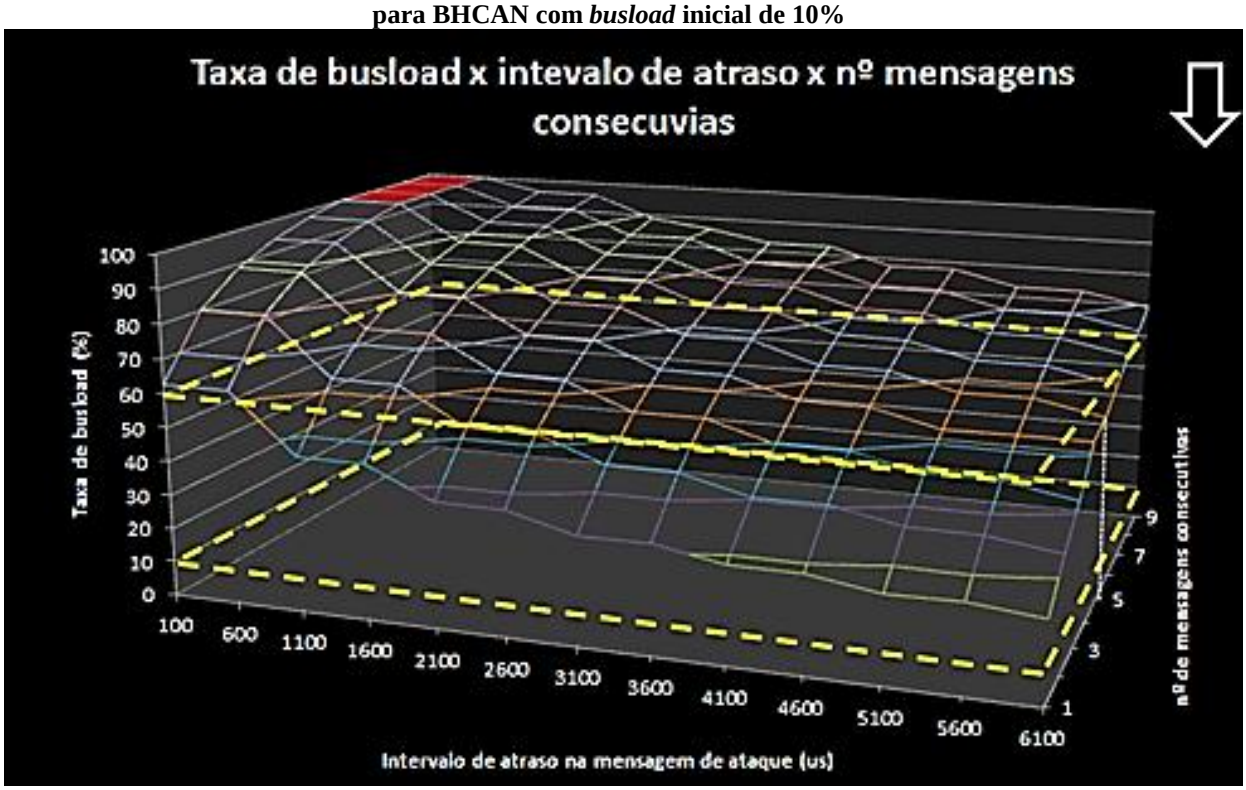



**Fonte: do autor**

O gráfico da Figura 34, apresenta o resultado da medição de *busload* em forma de um mapa 3D que representa a variação do *busload* em função das variações de intervalo de atraso em função do número de mensagens consecutivas. No eixo vertical o *busload* de 0 a 100%, no plano horizontal temos em um dos eixos os valores de calibração do atraso de 100µs a 6.100µs, com intervalo de 500µs e no outro a variação no parâmetro de número de mensagens consecutivas, de 1 a 9. As linhas amarelas no gráfico indicam a referência ao *busload* inicial, de 10%, e o limite recomendado, de 60%. Pode-se observar que o filtro conseguiu trazer a

saturação do barramento para valores inferiores ao limite recomendado para a operação normal do barramento, quando não está sob ataque, para valores do filtro acima 1.000µs de atraso e abaixo de 5 mensagens consecutivas. A faixa em azul indica onde o filtro não obteve êxito em manter o *busload* abaixo de 100%. Os parâmetros de configuração utilizados na faixa azul geraram espaços no barramento, durante um ataque, que não foram suficientes para comportar o volume de mensagens autênticas no período, causando perdas.

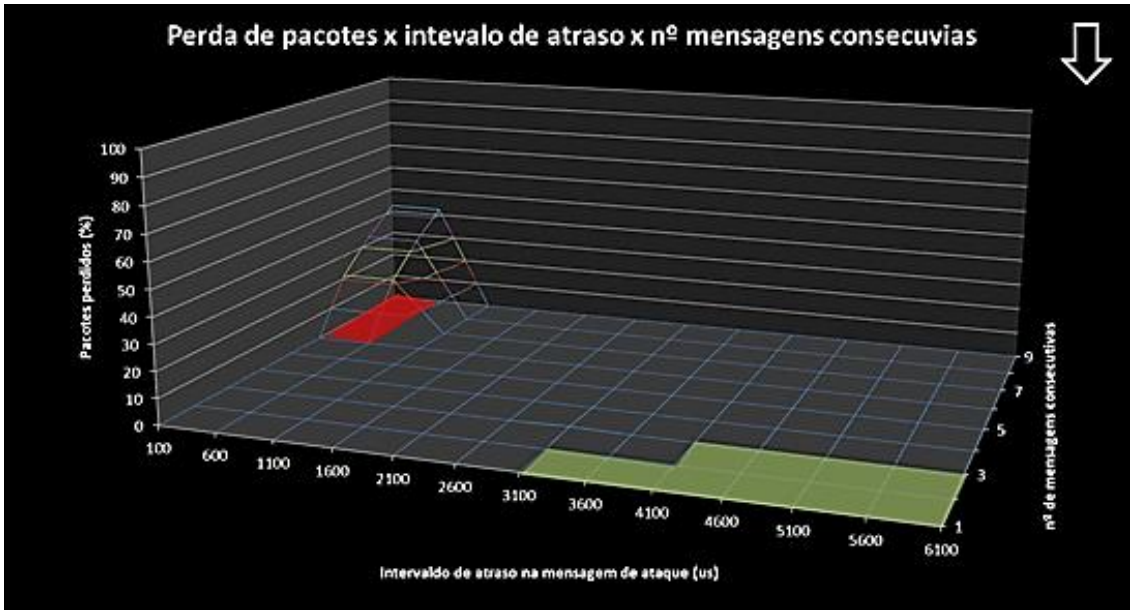
**Figura 34 – Medição de *busload* durante ataque**
**para BHCAN com *busload* inicial de 10%**




**Fonte: do autor**

Para analisar a perda de pacotes foi plotado um gráfico, Figura 35, onde temos no eixo vertical o *busload* de 0 a 100%, no plano horizontal temos em um dos eixos os valores de calibração do atraso de 100µs a 6.100µs e no outro a variação no parâmetro de número de mensagens consecutivas, de 1 a 9. Pode-se observar que somente houve perda de pacotes de mensagens na área em azul, que representa a calibração de tempo de atraso inferior a 1.000µs e o número de mensagens consecutivas maior que 6. As perdas foram de no máximo 45% dos pacotes, no pior caso. Quanto mais mensagens de ataque são permitidas, menor é o espaço para mensagens autênticas. Da mesma forma, o tempo de atraso gera lacunas para as mensagens autênticas. Quanto menor este valor, mais crítica é a realocação de espaço no barramento para as mensagens autênticas.

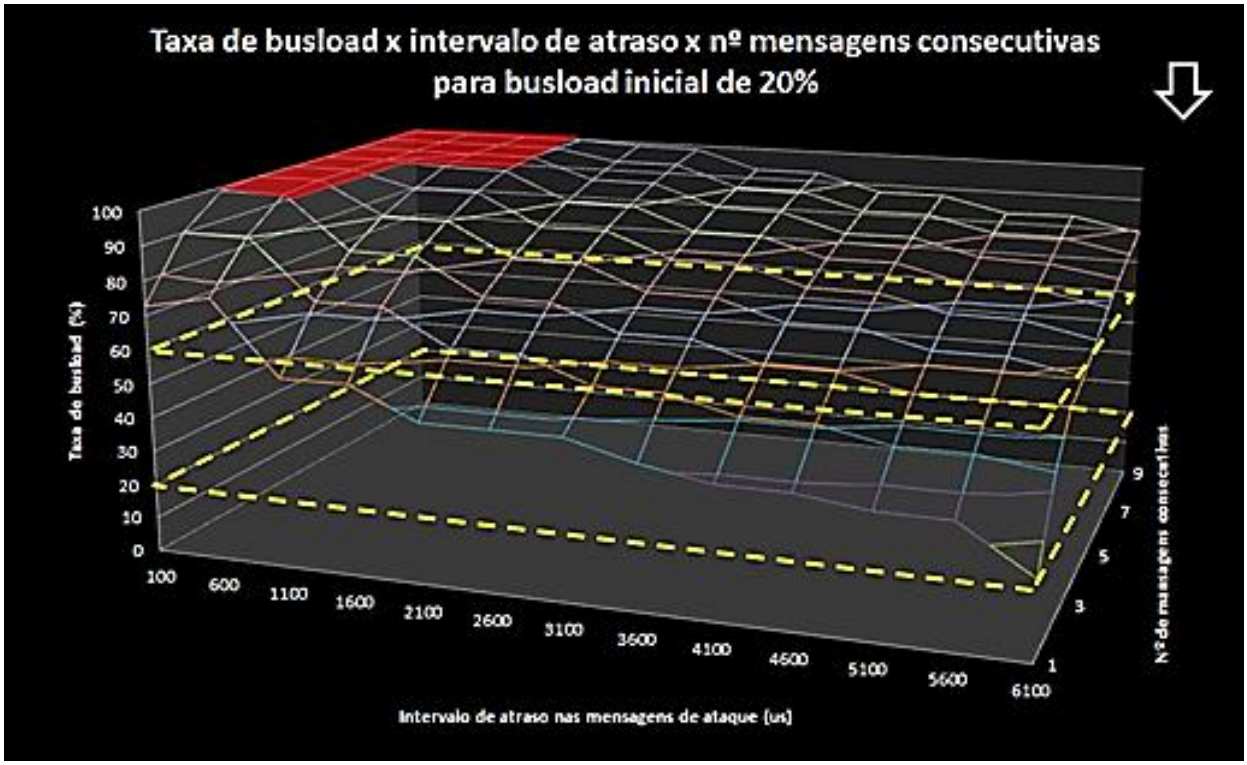
**Figura 35 - Resultado de perda de pacotes durante ataque para BHCAN com *busload* inicial de 10%**



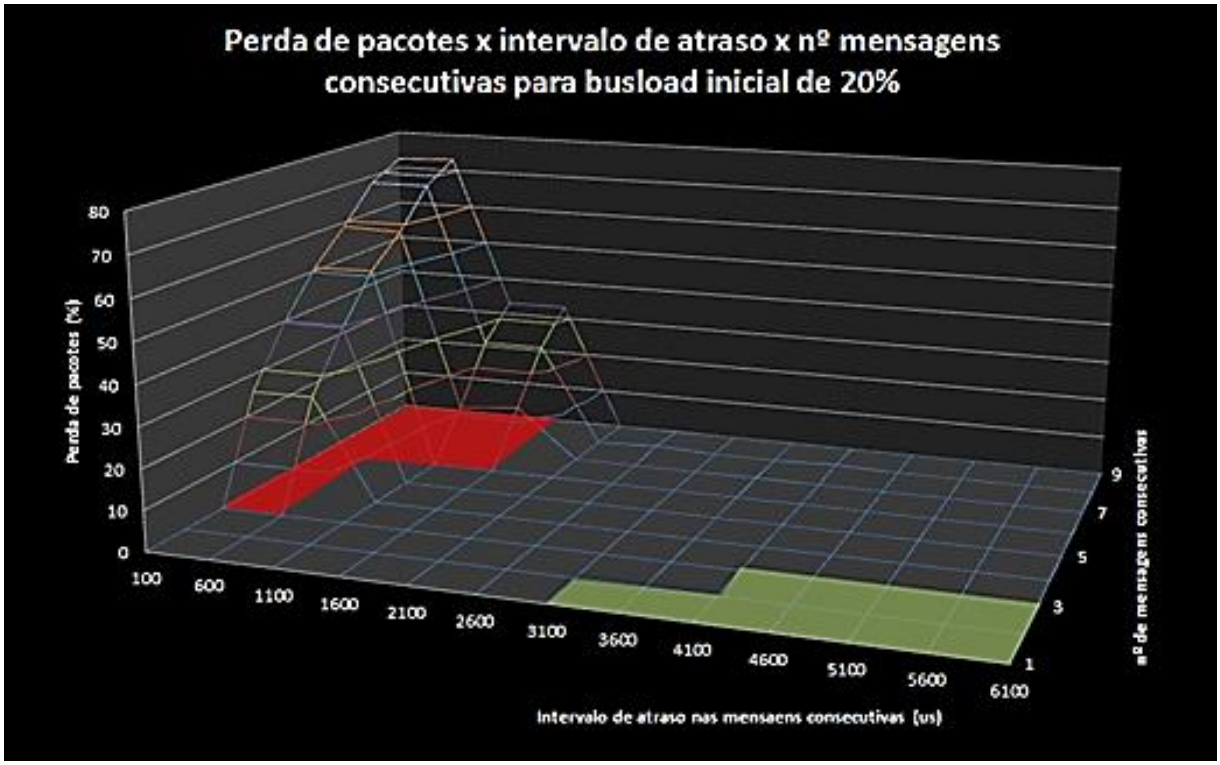

**Fonte: do autor**

Os testes foram repetidos para *busload* inicial de 20% e 60%, As Figura 36(a) e (b) são para 20% de *busload* inicial e Figura 37(a) e (b) para 60%. Pode-se observar que a margem para o filtro agir com o *busload* inicial a 60% é muito pequena, mas ainda assim foi possível encontrar valores de calibração que mantiveram o *busload* abaixo de 100% durante um ataque. Da mesma forma, com 60% de saturação inicial houve perda de pacotes que chegou a 100% para certa faixa de calibração dos parâmetros, entretanto, ainda assim o filtro conseguiu ser eficaz na área roxo destacada no gráfico, em que não houve nenhuma perda de pacote. Essa área roxo foi representada também para as demais saturações iniciais e pode ser observada em cada um dos gráficos, demonstrando que é possível obter valores ótimos de calibração para o filtro proposto para uma rede BH-CAN de 125kbps independente da saturação inicial do barramento.

**Figura 36 – Resultado *busload* e perda de pacotes durante ataque para BHCAN com *busload* inicial de** 20%

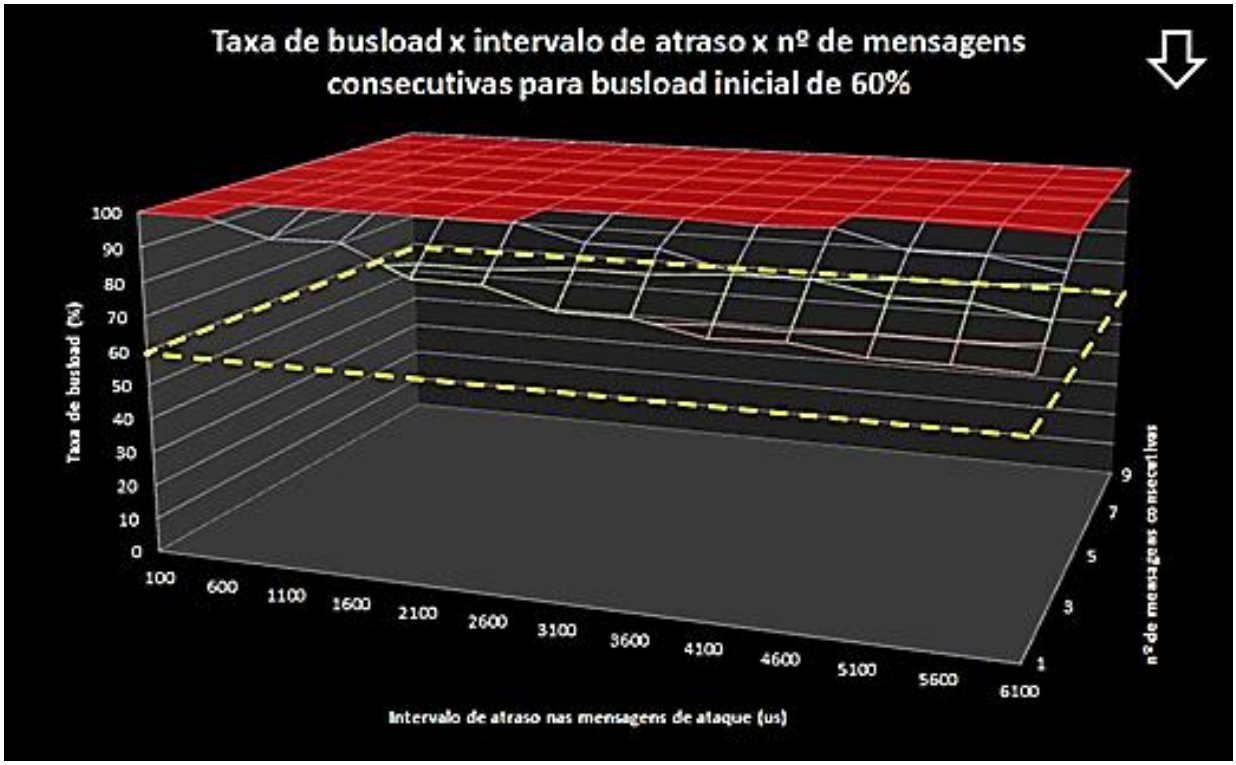



***(a)* busload sob ataque (20%)**

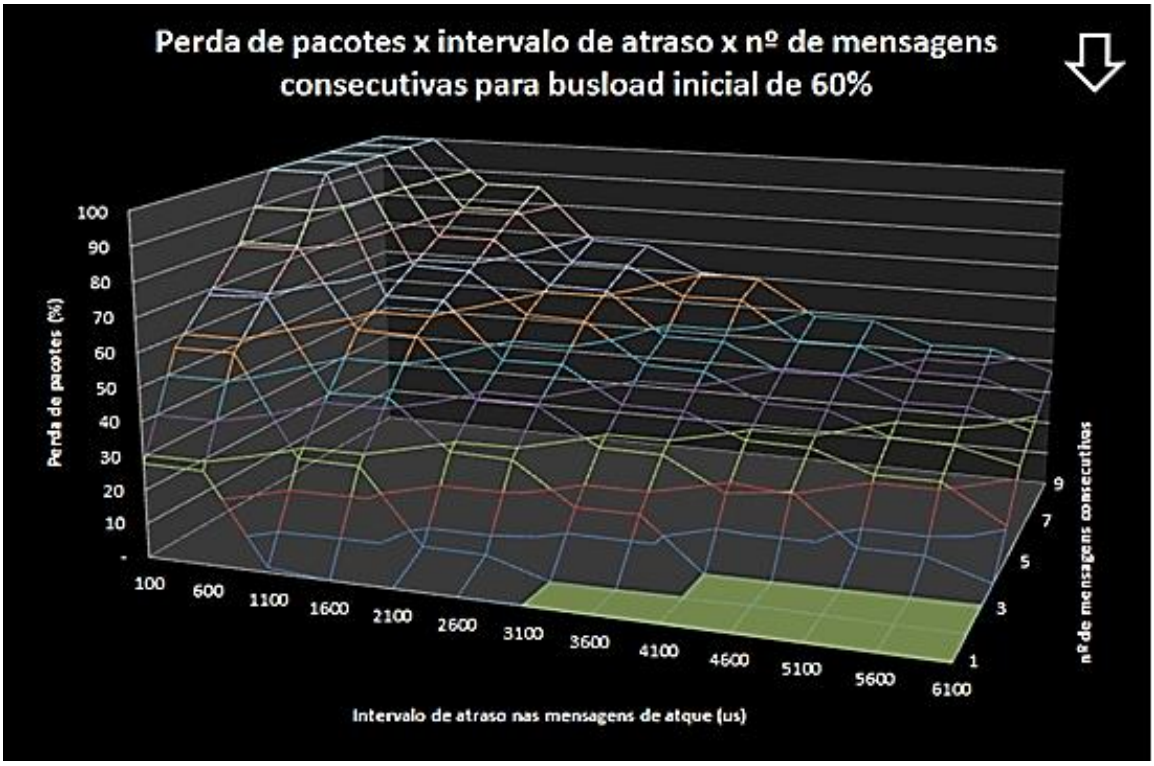



**(b) perda de pacotes (20%)**
**Fonte: do autor**

**Figura 37 - Resultado *busload* e perda de pacotes durante ataque para BHCAN com *busload* inicial de 60%**




***(a) busload* sob ataque (60%)**



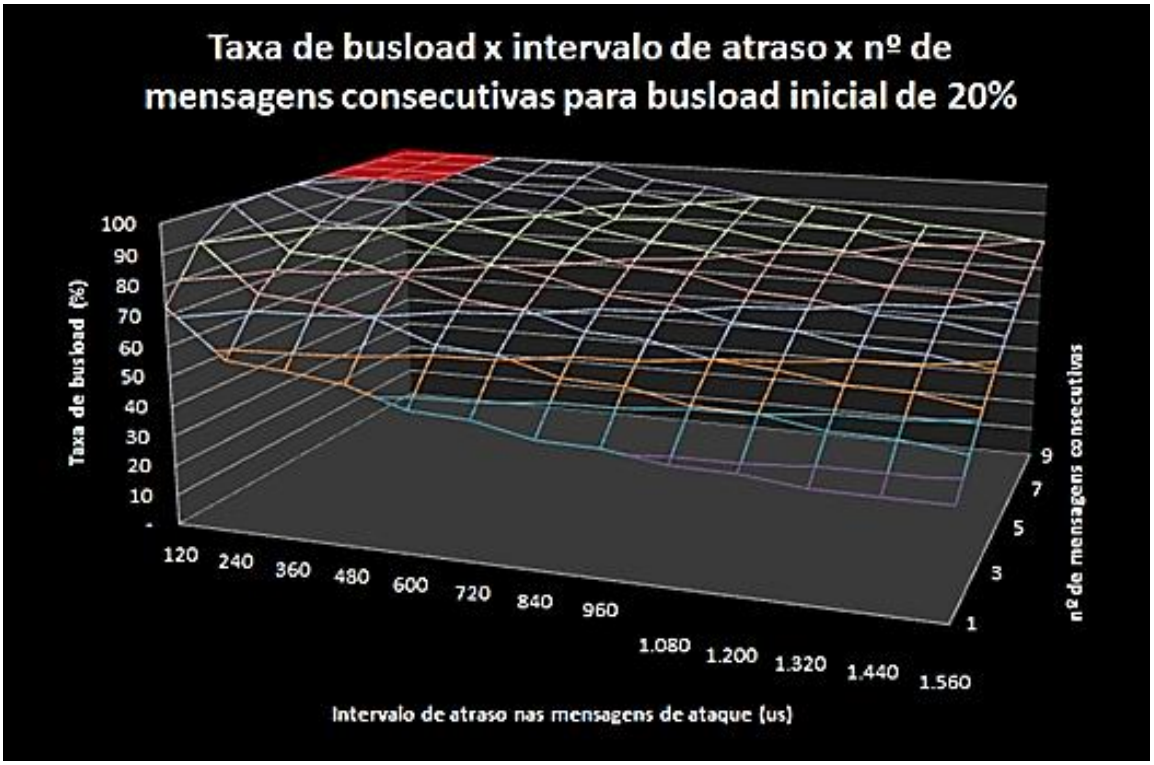

**(b) perda de pacotes (60%)**
**Fonte: do autor**

Uma vez que os dados validaram o filtro para BH-CAN, foi testado também se o filtro é eficaz em uma rede C-CAN com 500kbps. Foram plotados os gráficos para C-CAN, seguindo

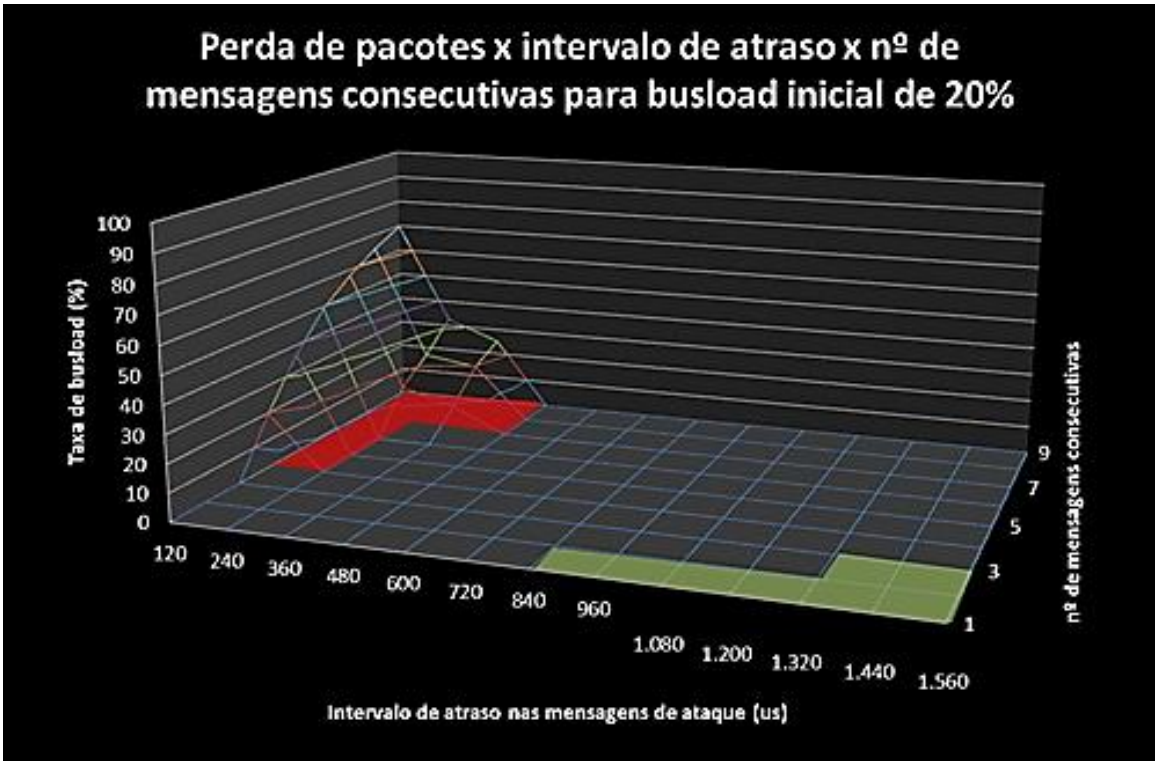
o mesmo critério de variação de tempo de atraso nas mensagens de ataque, ou seja, com intervalo de tempo de atraso de 50% do tamanho da mensagem. Não foi utilizada a variação de 10% do tamanho da mensagem CAN, uma vez que já foi demonstrado que não traz nenhum resultado prático na redução do busload durante um ataque. Para C-CAN com velocidade de comunicação de 500kbps, o tamanho da mensagem CAN é de 240µs, portanto, a variação do parâmetro foi de metade desse tempo, ou seja, 120µs.

Assim como na BH-CAN, a área azul na Figura 38(a) e Figura 39(a), representa a região em que os parâmetros de calibração não deram ao filtro condição de manter o ataque sob controle, e o busload alcançou 100% do barramento. Nas Figuras 38(b) e Figura (39(b) a região em azul representa perda de 100% dos pacotes. A região em roxo demonstra onde os parâmetros de calibração conseguiram manter sob controle o ataque sem que houvesse perda na transmissão dos pacotes.

**Figura 38 - Resultado busload e perda de pacotes durante ataque para C-CAN com busload inicial de** 20%

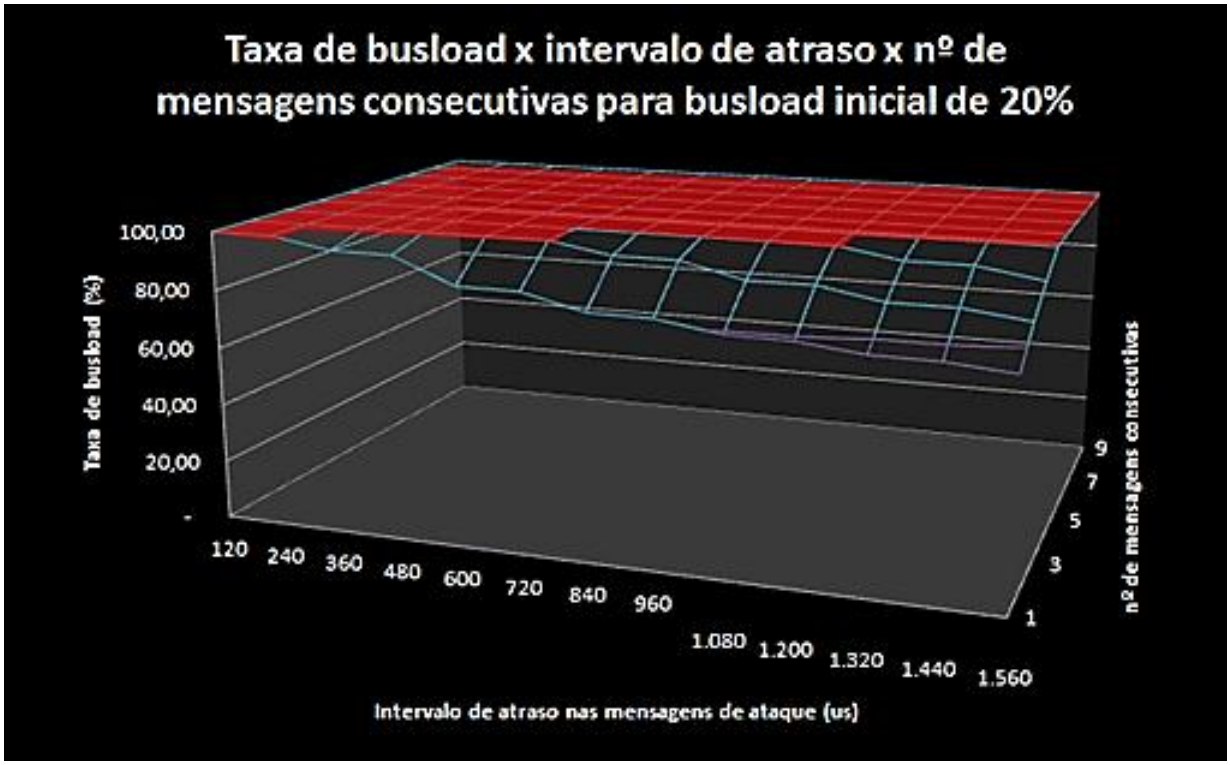



**(a) busload sob ataque (20%)**

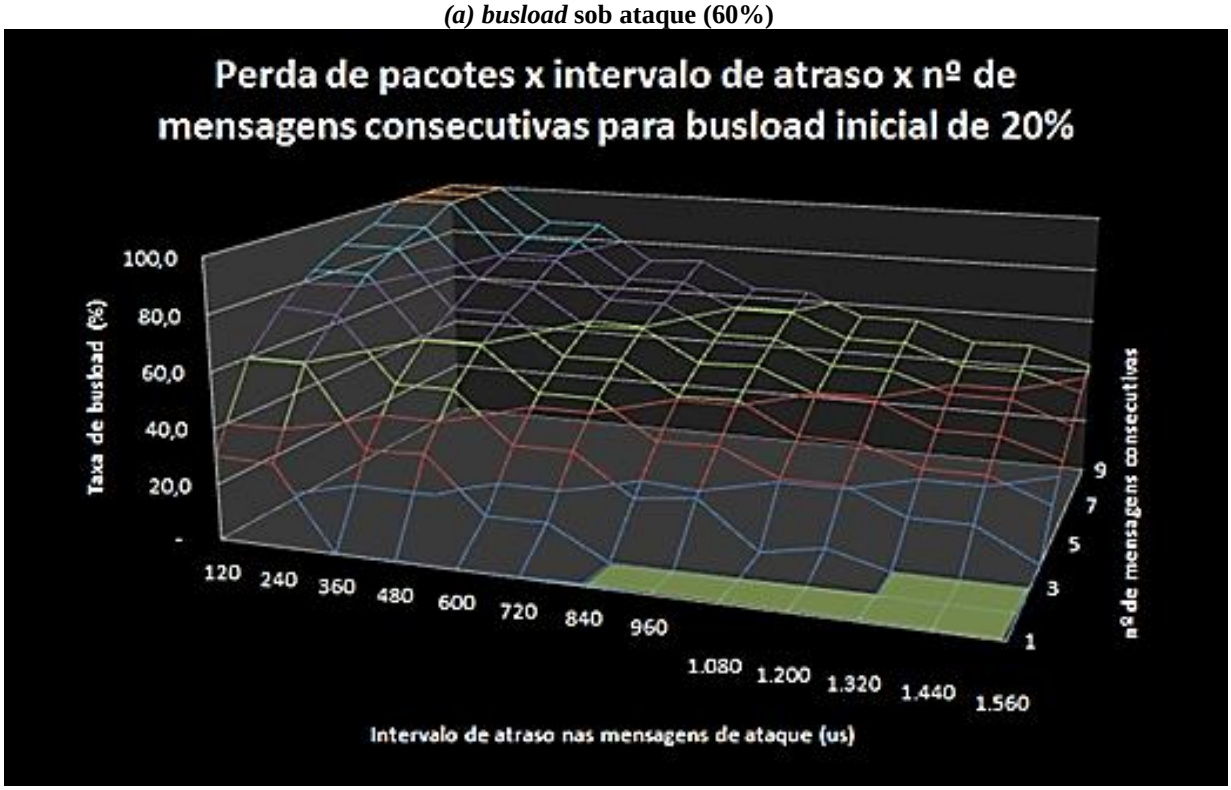



**(b) perda de pacotes (20%)**
**Fonte: do autor**

**Figura 39 - Resultado *busload* e perda de pacotes durante ataque para C-CAN com *busload* inicial de 60%**




***(a) busload* sob ataque (60%)**

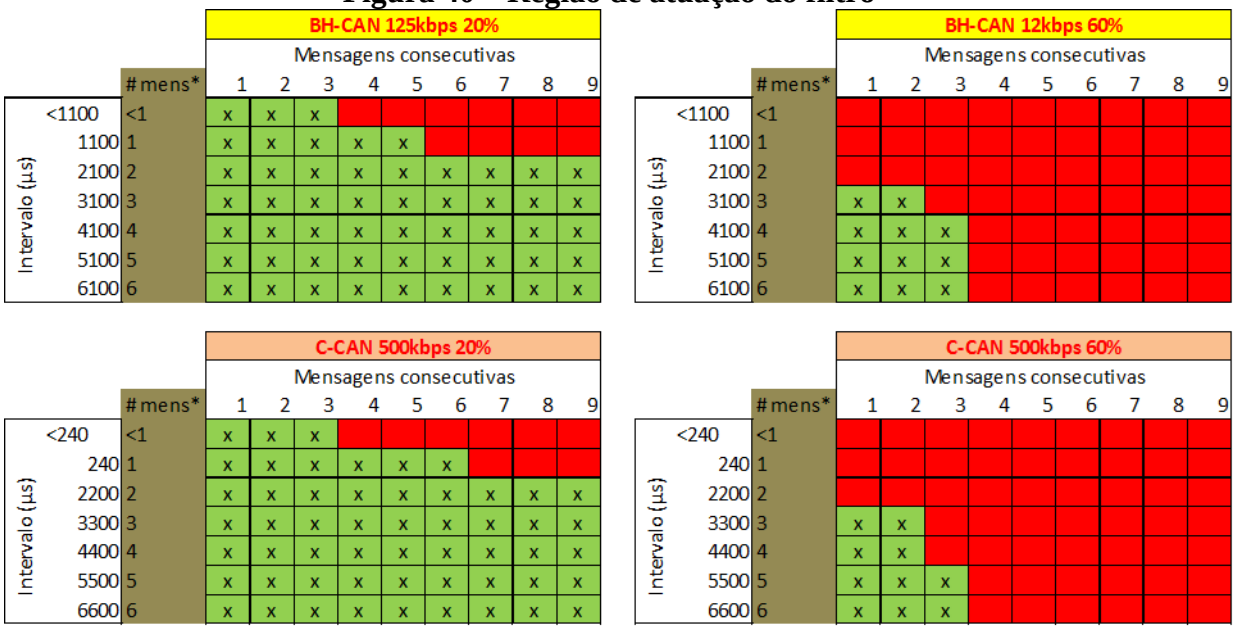



**(b) perda de pacotes (60%)**
**Fonte: do autor**

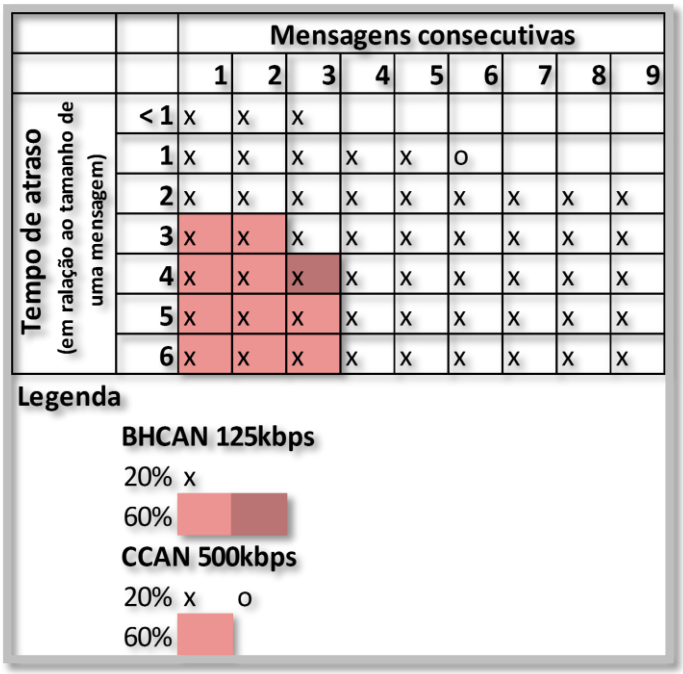
## 4.5 Análise dos resultados

As medições realizadas indicam que o filtro consegue ser eficiente em todas as condições testadas com os devidos ajustes na calibração. Foi possível notar que a saturação inicial do barramento e a taxa de transmissão modifica significantemente a eficácia do filtro. A Figura 40 resume as configurações do filtro de forma sobreposta para todas as condições testadas. Pode ser observado que existe uma região em que o filtro conseguiu conter o ataque em todas as condições do barramento. Isto significa que os valores da região amarela no quadro podem ser utilizados para todas as condições de barramento. No quadro, o tempo de atraso foi indicado em relação ao tamanho de uma mensagem, de forma a referenciar qualquer velocidade de barramento. Como vimos antes o tamanho de uma mensagem no barramento de 125kbps é de 97µs e para 500kbps é de 240µs.

**Figura 40 – Região de atuação do filtro**




(*)mens = equivalente ao tamanho de x mensagens CAN




**Fonte: do autor**

Acontece que um nó pode mandar mais de uma mensagem CAN diferente, com periodicidades diferentes. Desta forma o ideal é que se tenha o máximo número de mensagens consecutivas disponíveis para acessar o barramento, evitando atrasos na entrega dos pacotes. Como foi visto, para que o filtro funcione em qualquer tipo de barramento, poder-se-ia trabalhar com três mensagens consecutivas, mas aí o intervalo entre as três outras novas mensagens seria de 5x o tamanho de uma mensagem, que para um barramento BHCAN de 125kbps significaria 5x 970µs enquanto que para uma rede de 500kbps seria de 5x 240µs.

Estes valores de parâmetros são adequados para uma rede real? Para responder a esta questão, foi verificado em uma rede veicular que utiliza o barramento BHCAN com um *busload* de 15% e observou-se o seguinte: dos 7 nós que trocam informações entre si cada um envia uma média de 8 mensagens diferentes, o que envia mais transmite 14 mensagens e o que envia menos envia 2. O nó que envia mais mensagens transmite a uma taxa média de periodicidade de 350ms, ou seja, transmite 14 mensagens distribuídas em 350ms. A mensagem que tem periodicidade menor é transmitida a cada 100ms e a que tem periodicidade maior a cada 1.000ms. Uma mensagem com periodicidade de 100ms significa que ela é enviada 10 vezes a cada segundo, enquanto uma mensagem com periodicidade de 1000ms é enviada somente uma vez por segundo.

Se considerarmos uma média teríamos 14 mensagens transmitidas em um espaço de tempo de 350ms. Precisaríamos de um intervalo sem transmissão de 5x 970µs entre cada grupo de três mensagens, o que resulta em 4,8ms. Onde 14 mensagens divididos grupos de três mensagens consecutivas de 970µs cada, teríamos então 5 grupos de 3 mensagens cada um, com um tempo de 2,9ms cada grupo. Isto resultaria em (2,9 x 5) + (4,8 x 4) = 33,7ms, onde 2,9 é o tempo de envio das três mensagens consecutivas em 5 grupos mais 4,8 é o intervalo para enviar um novo grupo em 4 intervalos entre um grupo e outro, resultando em um tempo total de 33,7ms. O ciclo de periodicidade é de 350ms. Se considerarmos um ciclo mais crítico, que seria calcularmos com base no menor tempo de periodicidade das mensagens de uma rede BHCAN, teríamos 33,7ms de tempo de transmissão das 14 mensagens em um ciclo de 100ms de periodicidade, o que resulta em uma folga considerável para garantir que o filtro, em condições normais de comunicação, não irá gerar atraso na entrega dos pacotes.

Um *busload* maior significa um tempo de transmissão de mensagem menor e uma periodicidade também menor, entretanto, um *busload* maior não significa que um mesmo nó irá aumentar o número de mensagens no barramento. A principal motivação para um incremento da velocidade de um barramento não é para um mesmo nó enviar um número maior de mensagens e sim para que possam ser incluídos novos nós na rede.

Fazendo a mesma análise para uma rede real de 500kbps, encontramos em um barramento típico, 8 nós com uma média de 7 mensagens por nó. O nó que envia mais mensagens transmite 12 e o que transmite menos mensagens transmite 3. A média de periodicidade das mensagens é de 311ms, a menor periodicidade é de 10ms e a maior de 1.000ms. O tempo de transmissão de uma mensagem é de 240µs, então temos 4 grupos de 3 mensagens por vez, um intervalo de 3x 240µs entre os grupos, desta forma: (4 x 3) x 240 + (3 x 5) x 240 = 6,5ms. Isso quer dizer que teríamos ao todo 6,5ms para enviar todas as 12

mensagens dentro de um espaço de periodicidade de 311ms. Mesmo se considerarmos o caso mais crítico com todas as mensagens a uma periodicidade de 10ms, ainda assim teríamos cerca de 40% de barramento livre.

Existe mais uma análise que precisa ser feita e diz respeito às mensagens de diagnóstico. As mensagens de diagnóstico podem ser extensas, pois transmitem configurações e até mesmo o *firmware* dos módulos eletrônicos. Uma mensagem de diagnóstico pode ter 255bytes de comprimento. Existem serviços de *upload* de *firmware* que segmentam o arquivo permitindo transmitir pequenos pedaços que cada vez até enviar um arquivo inteiro. Nestes casos o filtro poderia gerar um atraso considerável no tempo de transmissão das mensagens de diagnóstico. Como foi visto anteriormente as mensagens de diagnóstico têm prioridade mais baixa que as demais mensagens. Desta forma, ao realizar um serviço de diagnóstico, este utiliza a "sobra" do barramento, ou seja, se o barramento está a 10%, o serviço de diagnóstico pode operar a 90%, mas se o barramento já está a 60%, sobram 40% para o serviço de diagnóstico. Existe um artifício de pedir aos demais nós para silenciarem para que uma operação específica seja realizada utilizando-se 100% do barramento para diagnóstico. Com a utilização do filtro este pedido de liberação do barramento surtiria impacto sobre o filtro e ainda assim teria um atraso significativo na transmissão dos dados.

Vamos imaginar que um arquivo de 2kB, usando uma rede atual a 100% do barramento, considerando ainda que o protocolo de transporte de uma mensagem CAN para diagnóstico transmite 7bytes de dados por vez, pois precisa de 1 byte de controle. Teríamos 2k/7, que dariam 286 pacotes, multiplicado pelo tamanho de uma mensagem CAN em um barramento de 125kbps, seriam necessários 278ms para a transmissão completa. Um arquivo de 2MB levaria 2M/7 = 285.715 pacotes, que levariam 277.144ms para serem transmitidos, ou 278s ou ainda 4,6 minutos. Isto considerando o melhor caso com um barramento dedicado às mensagens de diagnóstico.

Se considerarmos o efeito do filtro teríamos um intervalo equivalente a 5 mensagens a cada 3 mensagens enviadas. A conta ficaria assim: para 2k de dados, teríamos 286 pacotes divididos em grupos de 3 mensagens. Seriam então (96 grupos x 3) x 970µs + (95 x 5) x 970µs = 740ms para a transmissão completa. Para um arquivo de 2MB teríamos 285.715 pacotes divididos em (95.239 grupos x 3) x 970 + (95.238 x 5) x 970 = 739.043ms para a transmissão completa ou 277,14s ou ainda 12,32 minutos. O que seria um tempo considerável de cerca de 265% a mais no tempo de transmissão. Se considerarmos que atualmente existem processos de *upload* de *firmware* que levam 1h20 minutos para serem concluídos teria um salto para 3h30 minutos, o que seria inaceitável.

Uma forma simples de resolver este inconveniente seria o filtro ignorar mensagens de diagnóstico. Como mensagens de diagnóstico têm sempre uma prioridade baixa, existe um ID de corte que determina que todas as mensagens a partir daquele ID é uma mensagem de diagnóstico, normalmente 0x7xx para ID's de 11bits. Como foi demonstrado anteriormente, na Tabela 3, as mensagens de diagnóstico não representam um risco de ataque ao barramento. Desta forma o filtro poderia ser utilizado indistintamente, para todos os processos de comunicação de uma rede veicular.

# 5 CONCLUSÃO E TRABALHOS FUTUROS

Nesse trabalho foi apresentada uma proposta para minimizar os ataques do tipo DoS em qualquer tipo de rede que utilize barramento CAN e CAN-FD através da criação de um filtro na camada de arbitragem para dosar as mensagens. O filtro opera evitando a sobrecarga do barramento e perda de pacotes que ocorre durante um ataque DoS. Foram apresentados os resultados de testes feitos em um simulador confrontados com informações de uma rede real.

O objetivo de ter um mínimo de perda de comunicações autênticas diante de um ataque DoS e que sistemas críticos ou relacionados com segurança não deixem de operar é viável com o filtro proposto. O filtro é capaz de limitar a sobrecarga de tráfego, permitindo que mensagens autênticas possam ser transmitidas entre os nós durante um ataque.

## 5.1 Sugestões para trabalhos futuros

Para trabalhos futuros sugere-se verificar a implementação do filtro apresentado em *hardware* e testar em um veículo real, validando os parâmetros de configurações apresentados neste trabalho.

Sugere-se ainda que seja estudada a utilização de um *Intrusion Detection System* (IDS), na camada de controle do *transceiver* de cada um dos nós que compõem o barramento, verificando taxas anormais de periodicidade das mensagens. Variações anormais de periodicidade em mensagens poderiam ser tratadas como sendo uma falha de transmissão. Isto permitiria ativar o contador de erro do *transceiver* CAN obrigando-o a entrar no modo de BUS OFF que impede o nó de enviar mensagens ao barramento por um período de tempo.

No meio físico, poderia ser avaliada a eficácia em se utilizar redundância de barramento ou barramentos exclusivos para nós que são de extrema relevância para a segurança da vida dos ocupantes do veículo. Seria uma solução mais cara, do ponto de vista de implementação e poderia envolver redundância também de micro controladores, mas garantiria uma via imune a ataques ao barramento principal.

# BIBLIOGRAFIA

**ABNT. 2005.** *Tecnologia da informação — Técnicas de segurança — Código de prática para a gestão da segurança da informação.* s.l. : ABNT NBR ISO/IEC 17799, 2005.

**Abraham, A. and R., Jam. 2004.** *Soft Computing Models for Intrusion Detection Systems, Cryptography and Security, .* s.l. : ACM- class., 2004.

**Alefiya Hussain, John Heidemann, Christos Papadopoulos. 2003.** A Framework for Classifying Denial of Service Attacks. *USC/Information Sciences Institute.* 2003.

**Angelos D. Keromytis, Vishal Misra, Dan Rubenstein. 2004.** SOS: An Architecture for Mitigating DDoS Attacks. *IEEE - DARPA.* 2004.

**Brown, David A, et al. 2014.** Automotive Security Best Practices - Recommendations for security and privacy in the era of the next-generation car. 2014.

**Corrigan, Steve. 2008.** Introduction to the Controller Area Network - CAN. *Texas Instruments - Application Report.* July 2008.

**Debra L. Cook, Willian G. Morein, Angelos D. keromytis, Vishal Misra, Daniel Rubenstein. 2001.** WebSOS: Protecting Web Servers From DDoS Attacks. *Columbia University.* 2001.

**Florian Hartwich. 2012.** The CAN FD protocol overcomes CAN limits. *13º Internacional CAN Conference - iCC.* 2012.

**Groza, B. and Murvay, P.-S. 2011.** Higher layer authentication for broadcastin controller area networks (CAN). *Proceedings of The Sixth International Conference on Security and Cryptography, SECRYPT.* 2011.

**Guangsen Zhang, Manish Parashar. 2006.** Cooperative Defence against DDoS Attacks. *The State University of New Jersey.* Department of Electrical and Computer Engineering, 2006.

**Happel, Armin. 2014.** Secure communication for CAN FD. 2014.

**Hoppe, T. and Dittman, J. 2007.** Sniffing/Replay Attacks on CAN buses: A simulated attack on the electric window lift classified using an adapted. *2nd Workshop on Embedded.* outubro 4, 2007.

**ISO. 1996.** ISO 11989 - CAN Network. *CAN Network.* 1996.

**—. 1999.** Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signaling. s.l. : ISO 11898, 1999.

**Jeong, Chanbok and Kim, Youngmin. 2014.** Implementation of Effi cient SHA-256 Hash Algorithm for Secure Vehicle Communication using FPGA. *IEEE.* 2014, pp. 224-225.

**Ketter, Stefan. 2016.** Draft. *Projeto Draft.* [Online] Draft, 12 8, 2016. http://projetodraft.com/stefan-ketter-presidente-da-fca-latam-o-carro-vai-se-transformar-em-um-aplicativo-para-smartphones/.

**Kleberger, Pierre, Olovsson, Tomas and Jonsson, Erland. 2011.** Security aspects of the in-vehicle network in the connected car. *IEEE - Intelligent Vehicles Symposium IV.* Junho 5-9, 2011.

**Kuo, S. M. and Morgan, D.R. 1999.** Active noise control: atutorial review. *IEEE.* 1999, p. 87.

**Larson, Ulf E., Nilsson, Dennis K. and Jonsson, Erland. 2008.** An Approach to Specification-based Attack Detection for In-Vehicle Networks. *2008 IEEE Intelligent Vehicles Symposium.* Junnho 4-6, 2008.

**Lin, C.-W. and Sangiovanni-Vincentelli, A. 2012.** Cyber-security for the controller area network (CAN) communication protocol. *IEEE Computer - Proceedings of the 2012 International Conference on Cyber Security, ser. CYBERSECURITY.* 2012, pp. 1-7.

**Mahmud, Syed Masud and Shanker, Shobhit. 2006.** In-Vehicle Wiereless Personal Area Network (SWPAN). *IEEE Transactions on vehicular technology.* maio 2006, Vol. 55, 3.

**Mearian, Lucas. 2014.** Ethernet is comming to cars. 2014.

**Miller, Dr. Charlie and Valasek, Chris. 2015.** Remote exploitation of an unaltered passenger vehicle. *Black Hat USA.* Agosto 2015.

**Natale, Marco di. 2008.** Understanding and using the Controller Area Network. October 30, 2008, p. 47.

**Nilsson, D. K. and Larson, U. E. 2008.** Conducting forensic investigations. *First International Conference on Forensic Applications and Tech-.* janeiro 21-23, 2008.

**—. 2008.** Simulated attacks on CAN buses:. *5th IASTED Asian Conference on Communication Systems and Networks.* Langkawi, Malasia : s.n., Abril 2-4, 2008.

**Nilsson, Dennis K., Phung, Phu H. and Larson, Ulf E. 1996.** Vehicle ECU classification based on safety-security characteristics. *Chalmers University of Technology.* 1996.

**NXP. 2016.** TJA1046 - Dual HS CAN w/ standby mode - Product data sheet. 2, 2016.

**Obaidat, M., Sevillano, J. and Filipe, J. 2012.** Broadcast authentication in a low speed controller area network. *E-Business and Telecommunications.* 2012, Vol. 314, pp. 330-344.

**Onishi, Hiro. 2014.** Approaches for Vehicle Cyber Security. 2014.

**—. 2012.** Paradigm change of vehicle cyber security. Torrance, USA : s.n., 2012.

**Piskozub, Adrian. 2002.** Denial of service and distributed denial of service attacks.

*TCSET*. 2002.

**Schaal, Hans-Werner. 2012.** IP and Ethernet in Motor Vehicles - Challenges for the development tool, illustrated by today's applications. 2012.

**Silberschatz, Abraham, Galvin, Peter Baer and Gagme, Greg. 2004.** *Sistemas operacionais: conceitos e aplicações.* Rio de Janeiro : Elsevier, 2004. 85-352-1485-2.

**Silva, Adelino. 2008.** Redes de comunicação no Automóvel. 2008.

**Smith, Craig. 2014.** *Car hacker's handbook.* 2014.

**—. 2016.** The Car hacker's handbook - A guide for penetration tester. 2016.

**Szilagyi, C. and Koopman, P. 2009.** Flexible multicast authentication for time-triggered embedded control network applications. *IEEE.* 2009, pp. 165-174.

**—. 2010.** Low cost multicast authentication via validity voting in time-triggered embedded control networks. *Proceedings of the 5th Workshop on Embedded Systems Security.* 2010. pp. 10:1-10:10.

**Wampler, David and Fu, Huirong. 2009.** Security Threats and Countermeasures for Intra-Vehicle Networks. *IEEE Computer Society - Fifth Inernational Conference of Inforation Assurance and Security.* 2009.

**Wang, Qiyan and Sawhney, Sanjay. 2014.** VeCure: A Practical Security Framework to Protect the CAN Bus of Vehicles. 2014.

**Wolf, M., Weimerskirch, A. and Paar, C. 2004.** Security in Automotive Bus. *Workshop on Embedded IT-Security in Cars.* novembro 11-12, 2004.

**World-Mysteries. 2013.** 127 Years of modern automobile evolution. *World mysteries.* [Online] World mysteries, October 21, 2013. http://blog.world-mysteries.com/.

**Xianjun Geng, Adrew B. Whinston. 2000.** Defeating distributed denial of service attacks. *IEEE - IT Pro Security.* 2000.

**Yong Kim, Masa Nakamura. 2011.** Automotive Ethernet Network Requirements. *IEEE 802.1 AVB Task Force Meeting.* Março 2011.

**Yoshikawa, Masaya, et al. 2013.** Secure in-vehicle systems against Trojan attacks. 2013.

**Zaroo, Puneet. 2002.** A Survey of DDoS attacks and some DDoS defense mechanisms. *Advanced Information Assurance (CS 626).* 2002.

# GLOSSÁRIO

**Arbitragem**: O acesso ao barramento de uma rede CAN é definido pelo valor numérico do identificador da mensagem. O processo que determina qual mensagem tem o menor identificador e chamado de Arbitragem.

**Ataque D.o.S.:** Tipo de ataque em que um sistema tem seu serviço negado a usuários autênticos devido ao atacante sobrecarregar o processamento do sistema ou a rede de comunicação de dados.

**Automatic Park Assistance**: Sistema que auxilia o condutor no memento de estacionar o veículo. Sistemas mais simples indicam a manobra ao condutor, enquanto sistemas mais complexos estacionam de forma autônoma em vagas paralelas ou perpendiculares e também auxiliam no momento de sair de uma vaga apertada.

**Bluetooth**: É uma tecnologia de comunicação sem fio de que permite transmissão de dados e arquivos entre aparelhos diferentes.

**Busload**: Carga do barramento. É determinada pela quantidade de mensagens que trafegam no barramento em um período de tempo e medida em percentual, de zero a 100% de taxa de ocupação.

**Detecção de Intrusão**: Busca identifica trafego de redes fora dos padrões preestabelecidos para determinar se um sistema está sendo acessado de forma não autorizada.

**Dump de firmware**: Cópia do firmware diretamente da memória do hardware.

**Emergence Brake**: Sistema de segurança veicular dotado de sensores que monitoram a diante do veículo de forma constante indicando a presença de obstáculos ao condutor e com capacidade para frear o veículo automaticamente, caso o condutor não tenha uma reação a tempo para evitar uma colisão.

**Firmware**: Conjunto de instruções operacionais programadas diretamente no *harware* de um equipamento. É um tipo de *software*.

**Frame**: É a unidade de um dado, utilizado para definir a composição de uma mensagem.

**Gateway**: Normalmente está contido em um *hardware* com o objetivo de conectar logicamente duas ou mais redes diferentes, mantendo-as separadas fisicamente.

**Hacker**: é um indivíduo que se dedica, com intensidade incomum, a conhecer e modificar os aspectos mais internos de dispositivos eletrônicos e conexões de rede.

**Lógica Fuzzy**: Modelamento de parâmetros para geração de saídas em graus de pertinência em virtude de entradas imprecisas ou instáveis.

**Política de Segurança**: Regras e padrões para o gerenciamento da segurança.

**Rede CAN**: Rede de comunicação de dados largamente utilizada em veículos. Possui também aplicação em automação industrial, máquinas agrícolas e automação residencial.

**Rede em Broadcast**: Uma rede que transmite a todos os nós conectados a ela ao mesmo tempo sem necessidade de especificar um destinatário.

**Risco**: Probabilidade de perigo através de ameaça física ou financeira para a pessoa, o bem ou o meio ambiente.

**Spoofing**: Forjar uma mensagem para parecer autêntica.

**Stop&Start**: Sistema que visa redução de poluentes emitidos por veículos e redução de consumo de combustível que desliga e religa o veículo automaticamente quando o condutor para o veículo, normalmente em situações de trafego intenso ou semáforo.

**Vulnerabilidade**: São falhas no software de computador ou módulos veiculares que criam deficiências na segurança geral do computador, dos módulos ou da rede, permitindo que pessoas mal intencionadas ou não autorizadas tenham acesso de forma a ler informações, corromper ou danificar os sistemas, gerando algum tipo de dano ou prejuízo.

**Wi-Fi**: É uma marca da Wi-Fi Alliance que certifica produtos para interoperabilidade de conexão sem fio.

**Zigbee**: Protocolo para criação de redes sem fio de baixo consumo e baixa potência.

# APÊNDICE A – TABELAS DE RESULTADOS

As medições geraram um grande volume de dados. Ao todo foram feitas 21 variações de tempos de atraso, 9 variações de mensagens consecutivas para 3 variações de *busload* iniciais, para BH-CAN. Estas medições foram repetidos para C-CAN com 13 variações de tempos, 9 variações de mensagens consecutivas para 2 variações de *busload* iniciais, repetindo as medições 5 vezes para obter a média de pacotes, perfazendo um total de 4005 medições.

As tabelas foram disponibilizadas no endereço:

https://drive.google.com/open?id=0B6UD2Cv2R3Yfd0laOTFNcWprb2M

de forma que podem ser consultadas a qualquer momento para que os resultados possam ser criticados e os testes possam ser reproduzidos.

Nas paginas seguintes encontram-se as tabelas com o extrato dos dados de medição que foram utilizados para a geração dos gráficos do capítulo 4.

**Tabela 6 - Extração dos valores de *busload* e perda de pacotes para BH-CAN com *busload* inicial de 20%**

| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 72,71 | 72,94 | 73,94 | 70,90 | 72,26 | 76,68 | 72,90 | 72,88 | 72,92 | 55,54 | 55,41 | 55,64 | 46,67 | 46,75 | 46,74 | 41,64 | 38,15 | 38,12 | 35,57 | 35,47 | 22,65 |
| 2 | 90,02 | 90,16 | 90,02 | 90,34 | 90,14 | 90,29 | 90,25 | 90,29 | 90,21 | 72,90 | 72,74 | 72,87 | 62,34 | 62,54 | 55,49 | 55,34 | 50,62 | 50,60 | 46,62 | 46,72 | 43,96 |
| 3 | 98,90 | 98,91 | 99,74 | 98,83 | 98,50 | 98,93 | 98,82 | 98,79 | 98,66 | 83,01 | 83,45 | 83,32 | 72,72 | 72,76 | 65,37 | 65,46 | 59,68 | 59,74 | 55,41 | 55,56 | 55,56 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 90,07 | 90,30 | 90,08 | 80,16 | 80,37 | 72,89 | 72,83 | 67,00 | 67,14 | 62,37 | 62,36 | 58,59 |
| 5 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 95,31 | 95,07 | 95,26 | 85,78 | 85,96 | 78,66 | 78,61 | 72,98 | 72,77 | 68,31 | 68,14 | 63,98 |
| 6 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 98,99 | 98,96 | 98,73 | 90,18 | 90,32 | 83,19 | 83,15 | 77,50 | 77,67 | 72,86 | 72,67 | 68,83 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 93,74 | 93,70 | 87,22 | 86,98 | 81,38 | 81,57 | 76,73 | 76,11 | 72,96 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 96,52 | 96,62 | 90,20 | 90,23 | 84,88 | 84,93 | 80,41 | 80,34 | 76,42 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 98,94 | 98,67 | 92,94 | 92,93 | 87,66 | 87,62 | 83,10 | 83,48 | 79,40 |

| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 241,20 | 241,40 | 241,20 | 241,20 | 240,40 | 241,20 | 241,20 | 241,20 | 240,60 | 241,20 | 241,40 | 241,20 | 241,20 | 240,80 | 240,00 | 241,00 | 241,20 | 241,20 | 241,20 | 241,40 | 241,00 |
| 2 | 241,00 | 241,40 | 240,80 | 241,40 | 240,80 | 241,20 | 240,60 | 241,20 | 240,80 | 241,00 | 241,00 | 241,20 | 240,80 | 240,80 | 240,80 | 241,20 | 240,80 | 241,20 | 241,20 | 241,00 | 240,80 |
| 3 | 240,80 | 240,40 | 250,00 | 240,60 | 237,60 | 240,40 | 240,40 | 240,60 | 240,60 | 240,80 | 240,80 | 240,60 | 241,20 | 240,80 | 241,20 | 241,20 | 240,20 | 241,00 | 240,60 | 240,80 | 241,20 |
| 4 | 178,60 | 177,80 | 178,80 | 178,80 | 178,40 | 178,40 | 177,80 | 178,00 | 177,00 | 241,20 | 239,80 | 240,80 | 240,40 | 241,20 | 240,80 | 240,40 | 241,40 | 240,40 | 241,40 | 241,40 | 240,20 |
| 5 | 144,60 | 144,20 | 144,60 | 144,40 | 144,00 | 145,00 | 144,60 | 144,80 | 144,40 | 239,40 | 240,80 | 240,80 | 241,20 | 240,80 | 241,20 | 240,80 | 241,40 | 240,80 | 240,20 | 241,40 | 241,00 |
| 6 | 118,00 | 118,60 | 119,40 | 117,60 | 118,80 | 117,40 | 118,80 | 117,60 | 118,20 | 240,20 | 240,40 | 240,40 | 239,80 | 240,40 | 241,40 | 240,80 | 240,40 | 240,40 | 240,60 | 241,00 | 240,40 |
| 7 | 98,60 | 99,60 | 99,80 | 99,60 | 100,80 | 99,80 | 99,20 | 99,60 | 100,00 | 211,00 | 212,80 | 209,40 | 240,20 | 241,00 | 240,20 | 239,60 | 240,80 | 240,60 | 240,40 | 240,60 | 241,40 |
| 8 | 76,40 | 75,80 | 76,40 | 77,20 | 76,60 | 76,20 | 76,40 | 75,60 | 75,60 | 178,00 | 178,80 | 179,00 | 241,00 | 238,40 | 240,60 | 241,40 | 238,80 | 240,00 | 240,80 | 241,20 | 240,20 |
| 9 | 66,00 | 64,00 | 65,40 | 64,20 | 64,80 | 64,60 | 64,20 | 64,80 | 64,00 | 160,60 | 160,60 | 160,40 | 240,00 | 239,60 | 241,00 | 239,80 | 241,00 | 240,40 | 240,80 | 240,20 | 240,40 |

| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 2 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 3 | - | - | - | - | 1,0 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 4 | 25,6 | 25,9 | 25,5 | 25,5 | 25,7 | 25,7 | 25,9 | 25,8 | 26,3 | - | 0,1 | - | - | - | - | - | - | - | - | - | - |
| 5 | 39,8 | 39,9 | 39,8 | 39,8 | 40,0 | 39,6 | 39,8 | 39,7 | 39,8 | 0,2 | - | - | - | - | - | - | - | - | - | - | - |
| 6 | 50,8 | 50,6 | 50,3 | 51,0 | 50,5 | 51,1 | 50,5 | 51,0 | 50,8 | - | - | - | 0,1 | - | - | - | - | - | - | - | - |
| 7 | 58,9 | 58,5 | 58,4 | 58,5 | 58,0 | 58,4 | 58,7 | 58,5 | 58,3 | 12,1 | 11,3 | 12,8 | - | - | - | 0,2 | - | - | - | - | - |
| 8 | 68,2 | 68,4 | 68,2 | 67,8 | 68,1 | 68,3 | 68,2 | 68,5 | 68,5 | 25,8 | 25,5 | 25,4 | - | 0,7 | - | - | 0,5 | - | - | - | - |
| 9 | 72,5 | 73,3 | 72,8 | 73,3 | 73,0 | 73,1 | 73,3 | 73,0 | 73,3 | 33,1 | 33,1 | 33,2 | - | 0,2 | - | 0,1 | - | - | - | - | - |

**Fonte: do autor**

**Tabela 7 - Extração dos valores de *busload* e perda de pacotes para BH-CAN com *busload* inicial de 60%**

| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 95,39 | 95,05 | 95,49 | 86,91 | 86,67 | 81,19 | 81,10 | 77,33 | 78,01 | 75,73 | 75,74 | 75,17 |
| 2 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 95,48 | 95,32 | 90,56 | 90,05 | 86,82 | 86,94 | 83,74 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 99,18 | 99,88 | 95,01 | 94,95 | 92,06 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 98,56 |
| 5 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 6 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | |
| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1.000 | 1.100 | 1.600 | 2.100 | 2.600 | 3.100 | 3.600 | 4.100 | 4.600 | 5.100 | 5.600 | 6.100 |
| 1 | 516,00 | 514,20 | 519,00 | 516,00 | 516,40 | 517,40 | 514,40 | 514,40 | 517,80 | 713,40 | 704,80 | 716,80 | 716,40 | 715,60 | 714,60 | 715,20 | 713,00 | 714,80 | 717,20 | 715,20 | 716,20 |
| 2 | 307,60 | 327,80 | 315,20 | 310,00 | 310,00 | 307,60 | 310,20 | 310,00 | 310,20 | 517,20 | 517,80 | 519,40 | 674,20 | 676,00 | 715,20 | 716,60 | 716,20 | 712,00 | 714,00 | 717,20 | 715,00 |
| 3 | 225,00 | 222,60 | 224,80 | 227,60 | 217,40 | 219,60 | 222,60 | 225,00 | 227,40 | 414,20 | 414,40 | 411,80 | 514,20 | 517,80 | 604,80 | 604,20 | 714,80 | 716,20 | 714,40 | 715,60 | 713,00 |
| 4 | 136,80 | 137,60 | 137,40 | 137,60 | 137,20 | 137,40 | 137,40 | 137,00 | 137,20 | 312,40 | 309,80 | 312,80 | 440,60 | 440,60 | 517,40 | 519,20 | 587,60 | 568,80 | 672,40 | 672,60 | 714,60 |
| 5 | - | - | - | - | - | - | - | - | - | 265,00 | 265,00 | 265,00 | 387,00 | 384,80 | 461,60 | 462,60 | 513,40 | 515,80 | 560,40 | 560,20 | 639,20 |
| 6 | - | - | - | - | - | - | - | - | - | 224,80 | 222,20 | 227,00 | 315,60 | 320,00 | 406,60 | 410,60 | 468,80 | 471,40 | 518,00 | 522,40 | 553,40 |
| 7 | - | - | - | - | - | - | - | - | - | 184,20 | 181,00 | 180,60 | 277,00 | 279,80 | 374,40 | 375,40 | 433,80 | 428,80 | 478,20 | 475,60 | 516,40 |
| 8 | - | - | - | - | - | - | - | 0,40 | - | 137,80 | 137,00 | 137,40 | 248,80 | 248,80 | 312,20 | 309,80 | 397,60 | 395,80 | 444,20 | 444,60 | 485,80 |
| 9 | - | - | - | - | - | - | - | - | - | 100,60 | 102,20 | 102,00 | 222,40 | 227,20 | 283,00 | 290,40 | 367,60 | 368,00 | 413,80 | 411,80 | 454,00 |
| | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | |
| | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 | 1100 | 1600 | 2100 | 2600 | 3100 | 3600 | 4100 | 4600 | 5100 | 5600 | 6100 |
| 1 | 27,9 | 28,1 | 27,5 | 27,9 | 27,8 | 27,7 | 28,1 | 28,1 | 27,6 | 0,3 | 1,5 | - | - | - | 0,1 | 0,1 | 0,4 | 0,1 | - | 0,1 | - |
| 2 | 57,0 | 54,2 | 56,0 | 56,7 | 56,7 | 57,0 | 56,7 | 56,7 | 56,7 | 27,7 | 27,6 | 27,4 | 5,8 | 5,5 | 0,1 | - | - | 0,5 | 0,2 | - | 0,1 |
| 3 | 68,6 | 68,9 | 68,6 | 68,2 | 69,6 | 69,3 | 68,9 | 68,6 | 68,2 | 42,1 | 42,1 | 42,5 | 28,1 | 27,6 | 15,5 | 15,6 | 0,1 | - | 0,2 | - | 0,4 |
| 4 | 80,9 | 80,8 | 80,8 | 80,8 | 80,8 | 80,8 | 80,8 | 80,9 | 80,8 | 56,3 | 56,7 | 56,3 | 38,4 | 38,4 | 27,7 | 27,4 | 17,9 | 20,5 | 6,0 | 6,0 | 0,1 |
| 5 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 63,0 | 63,0 | 63,0 | 45,9 | 46,2 | 35,5 | 35,4 | 28,3 | 27,9 | 21,7 | 21,7 | 10,7 |
| 6 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 68,6 | 68,9 | 68,3 | 55,9 | 55,3 | 43,2 | 42,6 | 34,5 | 34,1 | 27,6 | 27,0 | 22,7 |
| 7 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 74,3 | 74,7 | 74,8 | 61,3 | 60,9 | 47,7 | 47,5 | 39,4 | 40,1 | 33,2 | 33,5 | 27,8 |
| 8 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 99,9 | 100,0 | 80,7 | 80,9 | 80,8 | 65,2 | 65,2 | 56,4 | 56,7 | 44,4 | 44,7 | 37,9 | 37,9 | 32,1 |
| 9 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 100,0 | 85,9 | 85,7 | 85,7 | 68,9 | 68,3 | 60,5 | 59,4 | 48,6 | 48,6 | 42,2 | 42,5 | 36,6 |

**Fonte: do autor**

**Tabela 8 - Extração dos valores de *busload* e perda de pacotes para C-CAN com *busload* inicial de 20%**

| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 72,95 | 57,29 | 55,58 | 53,10 | 46,91 | 45,87 | 41,68 | 41,18 | 38,23 | 37,91 | 35,74 | 35,56 | 35,56 |
| 2 | 90,31 | 74,46 | 72,94 | 69,42 | 62,51 | 61,06 | 55,56 | 54,85 | 50,64 | 50,11 | 46,88 | 46,51 | 44,00 |
| 3 | 98,99 | 84,46 | 83,35 | 78,61 | 72,93 | 70,90 | 65,48 | 64,47 | 59,93 | 58,37 | 55,58 | 55,12 | 52,06 |
| 4 | 100,00 | 91,25 | 90,30 | 85,17 | 80,37 | 77,50 | 72,92 | 71,49 | 67,15 | 65,96 | 62,52 | 61,77 | 58,70 |
| 5 | 100,00 | 96,01 | 95,27 | 89,47 | 85,97 | 83,23 | 78,73 | 77,22 | 72,92 | 71,77 | 68,23 | 67,55 | 64,30 |
| 6 | 100,00 | 100,00 | 99,01 | 93,45 | 90,30 | 86,47 | 83,39 | 81,20 | 77,71 | 76,29 | 72,94 | 72,25 | 68,91 |
| 7 | 100,00 | 100,00 | 100,00 | 95,27 | 93,75 | 86,47 | 87,16 | 84,97 | 81,58 | 80,36 | 77,00 | 76,00 | 72,99 |
| 8 | 100,00 | 100,00 | 100,00 | 97,50 | 97,50 | 92,85 | 90,27 | 87,79 | 84,97 | 83,78 | 80,34 | 79,50 | 76,46 |
| 9 | 100,00 | 100,00 | 100,00 | 99,60 | 98,94 | 94,57 | 92,94 | 90,16 | 87,87 | 86,11 | 83,33 | 82,28 | 79,50 |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
| 1 | 884,00 | 883,80 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,20 | 884,00 | 884,00 | 884,00 | 884,00 |
| 2 | 884,00 | 884,00 | 884,00 | 884,00 | 884,20 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 883,60 | 884,00 | 883,80 |
| 3 | 883,80 | 884,00 | 884,00 | 884,40 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 |
| 4 | 709,20 | 884,20 | 884,00 | 883,80 | 884,00 | 884,00 | 884,00 | 883,80 | 884,00 | 884,00 | 884,40 | 884,00 | 884,00 |
| 5 | 561,40 | 884,00 | 884,00 | 884,00 | 884,00 | 883,80 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 883,80 |
| 6 | 448,20 | 881,80 | 883,80 | 883,80 | 884,40 | 884,00 | 884,00 | 884,40 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 |
| 7 | 373,20 | 751,00 | 803,40 | 883,60 | 883,80 | 883,40 | 884,00 | 884,00 | 884,00 | 884,00 | 884,00 | 883,60 | 884,00 |
| 8 | 315,00 | 683,00 | 709,20 | 884,40 | 884,00 | 884,20 | 884,00 | 884,40 | 884,00 | 884,00 | 884,00 | 884,00 | 884,20 |
| 9 | 267,80 | 595,40 | 661,80 | 884,00 | 884,00 | 884,00 | 884,40 | 884,00 | 884,20 | 884,20 | 884,00 | 883,60 | 884,20 |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
| 1 | - | 0,0 | - | - | - | - | - | - | - | - | - | - | - |
| 2 | - | - | - | - | - | - | - | - | - | - | 0,0 | - | 0,0 |
| 3 | 0,0 | - | - | - | - | - | - | - | - | - | - | - | - |
| 4 | 19,8 | - | - | 0,0 | - | - | - | 0,0 | - | - | - | - | - |
| 5 | 36,5 | - | - | - | - | 0,0 | - | - | - | - | - | - | 0,0 |
| 6 | 49,3 | 0,2 | 0,0 | 0,0 | - | - | - | - | - | - | - | - | - |
| 7 | 57,8 | 15,0 | 9,1 | 0,0 | 0,0 | 0,1 | - | - | - | - | - | 0,0 | - |
| 8 | 64,4 | 22,7 | 19,8 | - | - | - | - | - | - | - | - | - | - |
| 9 | 69,7 | 32,6 | 25,1 | - | - | - | - | - | - | - | - | 0,0 | - |

**Fonte: do autor**

**Tabela 9 - Extração dos valores de *busload* e perda de pacotes para C-CAN com *busload* inicial de 60%**

| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100,00 | 100,00 | 96,46 | 96,35 | 87,79 | 87,69 | 82,65 | 82,47 | 79,23 | 79,00 | 76,72 | 76,58 | 74,82 |
| 2 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 96,40 | 96,42 | 91,46 | 91,56 | 87,79 | 87,77 | 85,23 |
| 3 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 96,45 | 96,45 | 93,07 |
| 4 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 99,75 |
| 5 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 6 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 7 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 8 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| 9 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 | 100,00 |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
| 1 | 1.923,00 | 1.937,80 | 2.761,60 | 2.757,40 | 2.762,80 | 2.759,00 | 2.741,00 | 2.761,40 | 2.764,00 | 2.765,60 | 2.758,60 | 2.762,60 | 2.765,20 |
| 2 | 1.110,20 | 1.112,00 | 1.923,20 | 1.892,20 | 2.444,20 | 2.440,20 | 2.760,60 | 2.764,60 | 2.761,40 | 2.760,80 | 2.757,40 | 2.761,80 | 2.758,00 |
| 3 | 723,80 | 763,20 | 1.418,20 | 1.418,40 | 1.922,80 | 1.924,20 | 2.329,40 | 2.254,20 | 2.720,80 | 2.603,80 | 2.761,80 | 2.761,80 | 2.763,20 |
| 4 | 466,20 | 466,60 | 1.110,00 | 1.115,60 | 1.556,60 | 1.557,00 | 1.935,40 | 1.923,40 | 2.185,80 | 2.186,60 | 2.439,40 | 2.439,80 | 2.762,80 |
| 5 | 300,60 | 301,00 | 911,40 | 912,40 | 1.320,20 | 1.313,60 | 1.636,40 | 1.622,80 | 1.935,00 | 1.947,60 | 2.143,00 | 2.143,00 | 2.378,60 |
| 6 | 181,40 | 182,60 | 714,00 | 763,00 | 1.138,00 | 1.111,00 | 1.436,20 | 1.418,20 | 1.665,60 | 1.679,60 | 1.923,80 | 1.923,60 | 2.113,00 |
| 7 | - | - | 590,40 | 577,20 | 971,00 | 970,80 | 1.265,00 | 1.265,60 | 1.506,20 | 1.506,40 | 1.708,40 | 1.708,40 | 1.929,00 |
| 8 | - | - | 466,00 | 465,80 | 857,40 | 857,60 | 1.115,20 | 1.110,60 | 1.353,20 | 1.354,40 | 1.556,20 | 1.556,40 | 1.733,00 |
| 9 | - | - | 376,00 | 375,60 | 756,80 | 723,80 | 1.003,00 | 1.003,00 | 1.237,60 | 1.237,60 | 1.436,20 | 1.425,20 | 1.593,40 |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | 120 | 240 | 360 | 480 | 600 | 720 | 840 | 960 | 1.080 | 1.200 | 1.320 | 1.440 | 1.560 |
| 1 | 30,4 | 29,8 | - | 0,1 | - | 0,1 | 0,7 | - | - | - | 0,1 | - | - |
| 2 | 59,8 | 59,7 | 30,4 | 31,5 | 11,5 | 11,6 | 0,0 | - | - | 0,0 | 0,1 | - | 0,1 |
| 3 | 73,8 | 72,4 | 48,6 | 48,6 | 30,4 | 30,3 | 15,6 | 18,4 | 1,5 | 5,7 | - | - | - |
| 4 | 83,1 | 83,1 | 59,8 | 59,6 | 43,6 | 43,6 | 29,9 | 30,3 | 20,8 | 20,8 | 11,7 | 11,6 | - |
| 5 | 89,1 | 89,1 | 67,0 | 67,0 | 52,2 | 52,4 | 40,7 | 41,2 | 29,9 | 29,5 | 22,4 | 22,4 | 13,9 |
| 6 | 93,4 | 93,4 | 74,1 | 72,4 | 58,8 | 59,8 | 48,0 | 48,6 | 39,7 | 39,2 | 30,3 | 30,3 | 23,5 |
| 7 | 100,0 | 100,0 | 78,6 | 79,1 | 64,8 | 64,8 | 54,2 | 54,2 | 45,5 | 45,4 | 38,1 | 38,1 | 30,1 |
| 8 | 100,0 | 100,0 | 83,1 | 83,1 | 69,0 | 68,9 | 59,6 | 59,8 | 51,0 | 51,0 | 43,6 | 43,6 | 37,2 |
| 9 | 100,0 | 100,0 | 86,4 | 86,4 | 72,6 | 73,8 | 63,7 | 63,7 | 55,2 | 55,2 | 48,0 | 48,4 | 42,3 |

**Fonte: do autor**

# APÊNDICE B – EQUIPAMENTOS UTILIZADOS

- CANoe – Vector

Software tool for creation and execution of (remaining bus) simulation, communication analysis and testing of ECUs in distributed systems. Supports bus system CAN

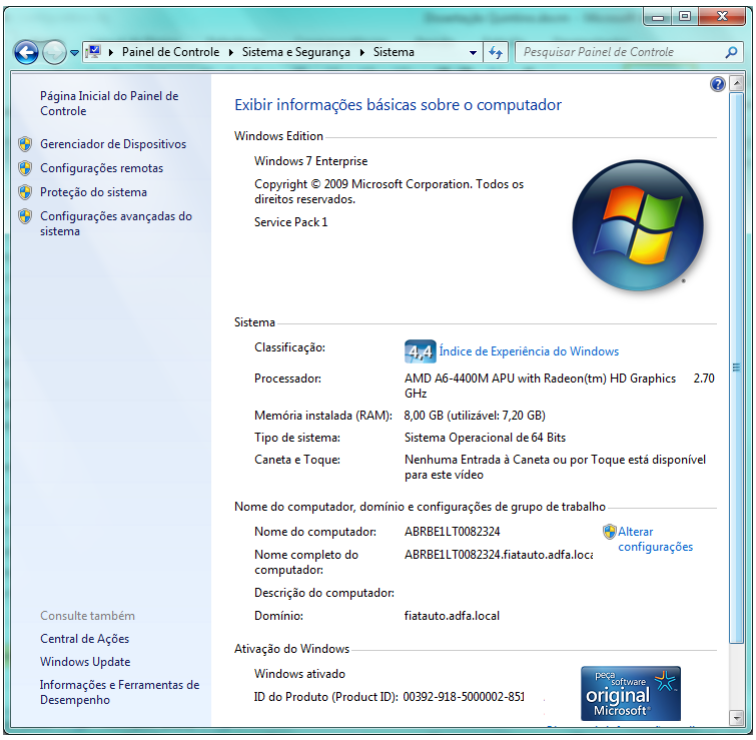

**GENERAL INFORMATION:**
CANoe.LIN.Ethernet
This product is licensed to: Fundação de Apoio ao Desenvolv.
**License number:** 1135310543
**Program version:** [Build 8.5.98 (32bit)]
**Vector channel interfaces connected to User PC:**
Dll Version 9.3.24

| Channel | Type | Driver version |
|---|---|---|
| CAN 1 | VN1630 | - |
| CAN 2 | VN1630 | - |

CANoe RT / VN8900:
Connection to remote runtime kernel is not configured.
information:
**Operating :** Windows 7 Service Pack 1 (64 bit)
**Processor:** AMD A6-4400M APU with Radeon(tm) HD Graphics
**Logical Processors**: 2
**Processor Cores**: 2
**Total physical memory**: 7369 MB
**Available physical memory**: 4681 MB
**Total virtual memory**: 4095 MB
**Available virtual memory**: 3363 MB
**Coprocessor Status**==4020 Control==27F Startup1==27F Startup2==27F Startup3==27F
**THREADING INFORMATION**:

**Maximum numbers of threads**: 0
**Flags**: ParallelizeSignalAccess, ParallelizeSimStop, ParallelizeFileAccess
**BUFFER INFORMATION**:
**Swapping mode**: No use of storage space, large history
**Swapping directory**: D:\users\fa049570\AppData\Local\Temp\
**Swapping space**: 0
**MODULES**:
**EXE Version**: 8.5.98; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\CANoe32.exe]
**Configuration Format Version**: 4.8.3
**PRTapi.DLL Version**: 8.5.98.0; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\PrtApi.dll]
**CAPLCOMP.DLL Version**: CAPL Compiler Version 8.5.98.0; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\CAPLcomp.dll]
**UTSTAT.DLL Version**: 08; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\utstatcn.dll]
**CANOEILNLVECTOR.DLL Version**: CANoe Interaction Layer NL (Vector) Version 3.16.19 (Sep 15 2015 09:49:09) ; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\CANOEILNLVECTOR.DLL]
**VSysChecks.DLL Version**: V100; [vsyschecks.dll Version: 100]
**ChipCfg.dll Version**: 8.5.98.0; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\ChipCfg.dll]
**PortLinkCfg.DLL Version**: 8.5.98.0; [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\PortLinkCfg.dll]
**LOADED PLUGINS**:
**DiVaPlugIn Dynamic Link Library Version**: 3.5.101 [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\DiVaPlugIn.vpi]
**Vector CANalyzer.Ethernet/CANoe.Ethernet Version**: 8.5.98 [C:\Program Files (x86)\Vector CANoe 8.5\Exec32\IPPlugIn.vpi]
**INSTALLED ADD-ONS**:
None
**TRACE WINDOW INFORMATION**:
**Trace**:
**Prioritized data storage**: Not Active
**View buffer size**: 1500
**Current number of records**: 0

- Laptop HP ProBook AMD 2.70 64 Bits 8GB RAM

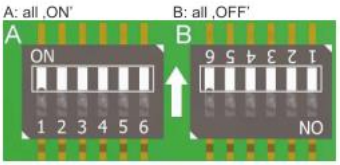


- VN16301 – Vector hardware interface for CAN physical connection

  USB-Interface for CAN, LIN, J1708, K-Line and I/0 (4+1 channels, 2 piggies).


**CAN/LIN Piggyback inserted**
If a CAN- or LINpiggy is inserted, the Piggyback is assigned to CH1 (CH2) and the built-in CAN transceiver is assigned to CH3 (CH4):

(1) 1051cap CAN Low
(2) Piggyback-dependent
(3) Piggyback-dependent
(4) Piggyback-dependent
(5) Shield
(6) 1051cap GND
(7) Piggyback-dependent
(8) 1051cap CAN High
(9) Piggyback-dependent




**D/A Input/Output**
(1) Analog Input
(2) -
(3) -
(4) Digital Input 0
(5) Digital Input 1
(6) Analog GND
(7) -
(8) Digital Output
(9) Digital GND

| | |
|---|---|
| **Analog input** | 10 bit, range 0..18 V, voltage tolerance up to 32 V, sampling rate up to 1 kHz |
| **Digital input** | Range 0..32 V, Schmitt trigger high 2.7 V, low 2.2 V, hysteresis 0.5 V, input frequencies up to 1 kHz |
| **Digital output** | Open drain, external supply up to 32 V, Current max. 500 mA, Short circuit / over voltage protected |

**CONFIGURATION**:
Channel 1 LINpiggy 7269mag
Channel 2 CANpiggy 1051cap
Channel 3 onboard CAN 1051cap
Channel 4 onboard CAN 1051cap
CANoe/CANalyzer activated
Variant CANoe/DENoe FULL
Activated Bussystem CAN
Activated Bussystem LIN
CANoe .Car2x activated
CANoe .Ethernet activated

- Osciloscópio analisador de protocolos Lecroy VBA64i

**SPECIFICATIONS**
Note: Specifications are subject to change without notice.

**VERTICAL SYSTEM**
• **Bandwidth @ 50 ohms (-3 dB):**

| | |
|---|---|
| WaveRunner 44Xi | 10 mV/div to 1 V/div 400 MHz<br>5 mV/div to 9.9 m/div 400 MHz<br>2 mV/div to 4.95 m/div 150 MHz |
| WaveRunner 64Xi | 10 mV/div to 1 V/div 600MHz<br>5 mV/div to 9.9 m/div 500 MHz<br>2 mV/div to 4.95 m/div 150 MHz |
| WaveRunner 62Xi | 10 mV/div to 1 V/div 600 MHz<br>5 mV/div to 9.9 m/div 500 MHz<br>2 mV/div to 4.95 m/div 150 MHz |
| WaveRunner 104MXi and WaveRunner 104Xi | 10 mV/div to 1 V/div 1 GHz<br>5 mV/div to 9.9 m/div 800 MHz<br>2 mV/div to 4.95 m/div 350 MHz |
| WaveRunner 204Xi | 10 mV/div to 1 V/div 2 GHz<br>5 mV/div to 9.9 m/div 1 GHz<br>2 mV/div to 4.95 m/div 350 MHz |

• **Bandwidth @ 1 Mohms (-3 dB) - typical:**

| | |
|---|---|
| WaveRunner 44Xi | 10 mV/div to 10 V/div 400 MHz |
| WaveRunner 64Xi | 10 mV/div to 10 V/div 500 MHz |
| WaveRunner 62Xi | 10 mV/div to 10 V/div 500 MHz |
| WaveRunner 104MXi and WaveRunner 104Xi | 5 mV/div to 10 V/div 500 MHz<br>2 mV/div to 4.95 mV/div 350 MHz |
| WaveRunner 204Xi | 5 mV/div to 10 V/div 500 MHz<br>2 mV/div to 4.95 mV/div 350 MHz |

- **Input Channels**: 4 (model 62Xi: 2)
- **Calculated Rise Time:** 10 mV/div to 1 V/div, 50 ohms (input risetime >/= 50 ps):

| | |
|---|---|
| WaveRunner 44Xi | 875 ns |
| WaveRunner 64Xi | 625 ps |
| WaveRunner 62Xi | 625 ps |
| WaveRunner 104MXi and WaveRunner 104Xi | 400 ps |
| WaveRunner 204Xi | 225 ps |

- **Bandwidth Limiters:**
  - Full
  - 200 MHz
  - 20 MHz
- **Input Capacitance, using PP008 probe:** < 9.5 pF (typical)
- **Input Capacitance of Channel (1/1, 1/10, 1/100):** < 20 pF (typical)
- **Input Impedance:** 1 Mohms // 16 pF or 50 ohms; WR104MXi, WR104Xi, WR204Xi: 1 Mohms // 20 pF or 50 ohms
- **Input Coupling:** 50 ohms: DC, GND; 1 Mohms: AC, DC, GND
- /!\ **Max Input Voltage (1/1, 1/10):** 50 ohms: 5 Vrms ; 1 microsecond pulse, 50% duty cycle: ±10 V peak 1 Mohms: 400 V max. (peak AC: </= 5 kHz + DC) WR104MXi, WR104Xi, WR204Xi: 50 ohms: 5 Vrms; 1 Mohms: 250 V max. (DC + Peak AC </= 10 kHz)
- **Installation (Overvoltage) Category:** CAT I
- **Channel-to-Channel Isolation:** > 40 dB @ < 100 MHz (> 30 dB @ full bandwidth)
- **Vertical Resolution:** 8 bits; up to 11 bits with enhanced resolution (ERES)
- **Sensitivity:** 50 ohms: 2 mV to 1 V/div fully variable; 1 Mohms: 2 mV to 10 V/div fully variable
- **DC Gain Accuracy:** ±1.0% of full scale (typical):

| | |
|---|---|
| ±1.5% | >/= 10 mV/div |
| ±2.5% | 5 mV/div |
| ±3.5% | 2 mV/div |

- **Offset Range:**

| | |
|---|---|
| 50 ohms | ±100 µV @ 2.0 to 10 mV/div<br>±200 µV @ 10.1 to 20 mV/div<br>±500 µV @ 20.1 to 50 mV/div<br>±1 mV @ 51 mV to 100 mV/div<br>±2 mV @ 102 to 200 mV/div<br>±5 mV @ 202 to 500 mV/div<br>±10 mV @ 502 mV to 1 V/div<br>±20 mV @ 1.02 to 2 V/div<br>±50 mV @ 2.02 to 5 V/div<br>±100 mV @ 5.02 to 10 V/div |
| 1 Mohms | ±100 µV @ 2.0 to 10 mV/div<br>±200 µV @ 10.1 to 20 mV/div<br>±500 µV @ 20.1 to 50 mV/div<br>±1 mV @ 51 mV to 100 mV/div<br>±2 mV @ 102 to 200 mV/div<br>±5 mV @ 202 to 500 mV/div<br>±10 mV @ 502 mV to 1 V/div<br>±20 mV @ 1.02 to 2 V/div<br>±50 mV @ 2.02 to 5 V/div<br>±100 mV @ 5.02 to 10 V/div |

- **Offset Accuracy:** Fixed gain setting < 2 V/div: ±(1.5% of offset value + 0.5% of full scale value + 1 mV)
- **Variable gain and settings >/= 2 V/div:** ±(1.5% of offset value + 1.0% of full scale value + 1 mV)
- **Probing System:** BNC or ProBus

**HORIZONTAL SYSTEM**

- **Timebases:** Internal timebase common to all input channels; an external clock can be applied at the auxiliary input
- **Time/div Range:** Real time: 200 ps/div to 10 s/div, RIS mode: 200 ps/div to 10 ns/div (WR104MXi, WR104Xi, WR204Xi: 100 ps/div to 10 ns/div), Roll mode: up to 1,000 s/div
- **Math & Zoom Traces:** 4 math/zoom traces standard

- **Clock Accuracy:** </= 5 ppm at 25 °C (</= 10 ppm at 5 to 40 °C)
- **Jitter Noise Floor:** 2 ps rms typical @ 100 mV/div
- **Time Interval Accuracy:** Clock Accuracy + Jitter Noise Floor
- **Sample Rate & Delay Time Accuracy:** equal to Clock Accuracy WAVERUNNER XI SERIES
- **Trigger & Interpolator Jitter:** </= 3 ps rms (typical)
- **Channel-to-Channel Deskew Range:** ±9 x time/div setting
- **Interpolator Resolution:** 1.2 ps
- **External Sample Clock (2-channel operation only; Ch 2 only in WaveRunner 62Xi):** DC to 600 MHz; 50 ohm (limited BW in 1 Mohm), BNC input, limited to 2 Ch operation (1 Ch in 62Xi), minimum rise time and amplitude requirements apply at low frequencies.

| Threshold | Impedance (ohms) | Minimum V p-p | Minimum Slew Rate (mV/ns) |
|---|---|---|---|
| TTL | 50 | 3 | 250 |
| TTL | 1 M | 3 | 350 |
| ECL | 50 | 0.2 | 150 |
| ECL | 1 M | 0.2 | 250 |
| 0 Cross | 50 | 0.2 | 150 |
| 0 Cross | 1 M | 0.2 | 250 |

- **Roll Mode:** User selectable; available at lower time/div settings

**ACQUISITION SYSTEM**

**Single-shot Sample Rate/Ch:** 5 GS/s

| | WaveRunner 44Xi | WaveRunner 64Xi | WaveRunner 62Xi | WaveRunner 104MXi/Xi | WaveRunner 204Xi |
|---|---|---|---|---|---|
| All Channels | 5 GS/s | 5 GS/s | 5 GS/s | 5 GS/s | 5 GS/s |
| Interleaved | 5 GS/s | 10 GS/s | 10 GS/s | 10 GS/s | 10 GS/s |
| | | | | | |

- **2 Channel Max.:** 10 GS/s

| | Maximum Acquisition Points/Ch - 2 Ch/4 Ch |
|---|---|
| Standard | 10M/20M |
| VL Memory Option | 12.5M/25M |

- **Random Interleaved Sampling (RIS):** 200 GS/s
- **Trigger Rate:** 1,250,000 waveforms per second

**ACQUISITION MODES**

- **Single-shot:** For transient and repetitive signals: 20 ps/div to 1000 s/div
- **Sequence:** 1000 segments standard
- **Sequence Time Stamp Resolution:** 1 ns
- **Intersegment Time:** 800 ns

**ACQUISITION PROCESSING**

- **Time Resolution (minimum, single-shot):** 200 ps (5 GS/s); 100 ps (10 GS/s)
- **Averaging:** Summed averaging to 1 million sweeps; Continuous averaging to 1 million sweeps
- **Enhanced Resolution (ERES):** from 8.5 to 11 bits vertical resolution
- **Envelope (Extrema):** Envelope, floor, roof for up to 1 million sweeps
- **Interpolation:** Linear, (sinx)/x

**TRIGGERING SYSTEM**

- **Modes:** Normal, Auto, Single, and Stop
- **Sources:** Any input channel, External, Ext/10, or line; slope and level are unique to each source (except line)
- **Coupling Mode:** GND, DC 50 ohms, DC 1 Mohms, AC 1 Mohms
- **Pre-trigger Delay:** 0 to 100% of memory size (adjustable in 1% increments or 100 ns)
- **Post-trigger Delay:** 10,000 divisions in real time mode; limited at slower time/div settings
- **Holdoff by Time or Events:** 1 ns to 20 s or from 1 to 99,999,999 events
- **Internal Trigger Range:** ±4.1 div from center (typical)
- **Trigger and Interpolator Jitter:** </= 3 ps rms (typical)
- **Maximum Trigger Sensitivity with Edge Trigger (Ch1-4 + external):**

| 44Xi | 64Xi | 62Xi | 104MXi/Xi | 204Xi |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 2 div @ < 400 MHz<br>1 div @ < 200 MHz | 2 div @ < 600 MHz<br>1 div @ < 200 MHz | 2 div @ < 600 MHz<br>1 div @ < 200 MHz | 2 div @ < 1 GHz<br>1 div @ < 200 MHz | 2 div @ < 2 GHz<br>1 div @ < 200 MHz |

• **Maximum Trigger Frequency with SMART Trigger (Ch1-4 + external):**

| 44Xi | 64Xi | 62Xi | 104MXi/Xi | 204Xi |
|---|---|---|---|---|
| 400 MHz<br>@ >/= 10 mV | 600 MHz<br>@ >/= 10 mV | 600 MHz<br>@ >/= 10 mV | 1 GHz<br>@ >/= 10 mV | 2 GHz<br>@ >/= 10 mV |

• **Trigger Level DC Accuracy:** ±4% of full scale ±2 mV (typical)
• **External Trigger Range:** EXT/10 ±4 V; EXT ±400 mV

**BASIC TRIGGERS**
• **Edge/Slope/Line:** Triggers when the signal meets the slope and level condition.
• **Width:** Triggers on positive or negative pulse widths selectable from 500 ps to 20 s or on intermittent faults (subject to bandwidth limit of oscilloscope).
• **Pattern:** Logic combination (AND, NAND, OR, NOR) of 5 inputs (4 channels and external trigger input – 2 Ch+EXT on WaveRunner 62Xi). Each source can be high, low, or don't care. The High and Low level can be selected independently. Triggers at start or end of the pattern.
• **State or Edge Qualified:** Triggers on any input source only if a defined state or edge occurred on another input source. Delay between sources is selectable by time or events.
• **TV:** Provides stable triggering on standard or custom composite video signals. Use them on PAL, SECAM, or NTSC systems. Optional HDTV Trigger for 1080i, 1080p and 720p formats along with non-standard formats up to 2000 lines.

**SMART TRIGGERS**
• **Dropout:** Triggers if the input signal drops out for longer than a selectable time-out between 1 ns and 20 s.
• **Glitch:** Triggers on positive or negative glitches with widths selectable from 500 ps to 20 s or on intermittent faults (subject to bandwidth limit of oscilloscope).
• **Signal or Pattern Interval:** Triggers on intervals selectable from 1 ns to 20 s.
• **Runt:** Trigger on positive or negative runts defined by two voltage limits and two time limits. Select between 1 ns and 20 s.
• **Slew Rate:** Activates a trigger when the rising or falling edge of a pulse crosses two threshold levels, an upper level and a lower level.

**AUTOMATIC SETUP**
• **Autosetup:** Automatically sets timebase, trigger, and sensitivity to display a wide range of repetitive signals.
• **Vertical Find Scale:** Automatically sets the vertical sensitivity and offset for the selected channels to display a waveform with maximum dynamic range.

**PROBES**
• **Probes:** One PP008 probe per channel standard (WR104MXi, WR104Xi, WR204Xi: one PP007 per channel); optional passive and active probes are available.
• **Probe System - ProBus:** Automatically detects and supports a wide variety of compatible probes
• **Scale Factors:** Automatically or manually selected depending on probe used

**COLOR WAVEFORM DISPLAY**
• **Type:** Color 10.4-inch flat panel TFT LCD with high resolution touch screen
• **Resolution:** SVGA; 800 x 600 pixels; maximum external monitor output resolution of 2048 x 1536 pixels
• **Real Time Clock:** Date, hours, minutes, and seconds displayed with waveform; accurate to ±50 ppm; SNTP support to synchronize to precision internet clocks
• **Number of Traces:** Maximum of eight traces; simultaneously displays channel, zoom, memory, and math traces
• **Grid Styles:** Single, Dual, Quad, Octal, XY, Single+XY, Dual+XY
• **Waveform Display Styles:** Sample dots joined or dots only

**ANALOG PERSISTENCE DISPLAY**

• **Analog and Color-graded Persistence:** Variable saturation levels; stores each trace's persistence data in memory
• **Persistence Selections:** Select analog, color, or 3-D
• **Trace Selection:** Activate Analog Persistence on all or any combination of traces
• **Persistence Aging Time:** From 500 ms to infinity
• **Sweeps Displayed:** All accumulated or all accumulated with last trace highlighted

**ZOOM EXPANSION TRACES**
Display up to 4 Math/Zoom traces

**INTERNAL WAVEFORM MEMORY**
**Waveform:** M1, M2, M3, M4 (Store full-length waveforms with 16 bits/data point.) Or save to any number of files (limited only by data storage media).

**SETUP STORAGE**
**Front Panel and Instrument Status:** Save to the internal hard drive or to a USB-connected peripheral device.

**INTERFACE**
• **Remote Control:** Through Windows® Automation or LeCroy remote command set
• **GPIB Port (optional):** Supports IEEE-488.2
• **Ethernet Port:** 10/100Base-T Ethernet interface (RJ-45 connector)
• **USB Ports:** 5 USB ports (one at front of oscilloscope) support Windows compatible devices.
• **External Monitor Port (standard):** 15-pin D-Type SVGA compatible DB-15; connect a second monitor to use dual monitor display mode
• **Parallel Port:** 1 standard
• **Serial Port:** DB-9 COM1 port (not for remote control of oscilloscope)

**AUXILIARY INPUT**
**Signal Types:** Select External Trigger or Clock input on front panel.

**AUXILIARY OUTPUT**
• **Signal Types:** Select from calibrator signal on front panel or control signals output from rear panel BNC.
• **Calibrator Signal:** 250 Hz to 1 MHz square wave or DC level; 50 mV to 1.0 V (selectable) into 1 kohms
• **Control Signals:** trigger enabled, trigger out, pass/fail status, or off

**MATH TOOLS (STANDARD)**
Display up to four math function traces (F1 to F4). The easy-to-use graphical interface simplifies setup of up to two operations on each function trace. Function traces can be chained together to perform math-on-math.

- absolute value
- average (summed)
- average (continuous)
- copy
- derivative
- deskew (resample)
- difference (-)
- enhanced resolution (to 11 bits vertical)
- envelope
- exp (base e)
- exp (base 10)
- fft (power spectrum, magnitude, phase)
- floor
- histogram of 1,000 events
- integral
- invert (negate)
- ln (log base e)
- log (base 10)
- MATLAB math
- product (X)
- ratio (/)
- reciprocal
- rescale (with units)
- roof
- segment
- segment
- (sinx)/x
- square
- square root
- sum (+)
- trend (datalog) of 1,000 events
- zoom (identity)

**MEASURE TOOLS (STANDARD)**
Display any 8 parameters together with statistics, including their average, high, low, and standard deviations. Histicons provide a fast, dynamic view of parameters and wave shape characteristics.

- amplitude
- area
- base
- cycles
- delay
- delta delay
- delta time @ level
- Dtrig time
- duration
- duty cycle
- fall time (90-10%, 80-20%, @ level)
- first
- frequency
- last
- level @ x
- MATLAB param
- maximum
- mean
- median
- minimum
- number of points
- overshoot+
- overshoot-
- peak-to-peak
- period
- phase
- rise time (10-90%, 20-80%, @ level)
- rms
- std. deviation
- time @ level
- top
- width
- width negative
- x @ minimum
- x @ maximum

**PASS/FAIL TESTING**
Test multiple parameters against selectable parameter limits at the same time. Pass or fail conditions can initiate actions including: document to local or networked files, email the image of the failure, save waveforms, send a pulse out at the front panel auxiliary BNC output, or (with GPIB option) send a GPIB SRQ.

**GENERAL**
• **Auto Calibration:** Ensures specified DC and timing accuracy is maintained for 1 year minimum.
• **Power Requirements:** Single phase, 100 to 240 V rms (±10%) at 50/60 Hz (±5%); or single phase, 100 to 120 V rms (±10%) at 400 Hz (±5%); Automatic AC voltage selection

| | | |
|---|---|---|
| Voltage Range: | 90 to 264 V rms | 90 to 132 V rms |
| Frequency Range: | 47 to 63 Hz | 380 to 420 Hz |

• **Power Consumption:** 340 watts (340 VA) max., WaveRunner 62Xi: 290 W (290 VA), depending on accessories installed (probes, PC port plug-ins, etc.); Standby State: 12 watts
• **Physical Dimensions (HWD):** 260 mm x 340 mm x 152 mm (10.2 in. x 13.4 in. x 6.0 in.); height measurement excludes foot pads
• **Weight:** 6.95 kg (15.3 lbs.)

- Bancada de teste