

Projeto de Redes Intraveiculares Híbridas

Rodrigo Lange¹, Rômulo de Oliveira¹, Nestor Roqueiro¹

¹Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

Abstract. *O FlexRay é um protocolo de comunicação que tem sido fortemente promovido como o futuro padrão de fato para sistemas automotivos de tempo real. Entretanto, a maioria dos dispositivos como sensores e atuadores para fins automotivos atualmente disponíveis utilizam outros protocolos como Controller Area Network (CAN) e Local Interconnect Network (LIN). Caso exista troca de informações entre dispositivos localizados em segmentos de redes que utilizam protocolos diferentes, será necessário um gateway entre estes segmentos. Este artigo descreve os primeiros resultados e observações de um trabalho onde são estudados aspectos relativos ao projeto de redes veiculares que utilizam gateways para trocas de dados.*

1. Introdução

Desde a década de 1970 tem sido observado um crescimento exponencial no número de dispositivos eletrônicos incorporados em veículos automotores, sendo que no ano de 2004 um veículo *high-end* chegava a utilizar mais de 60 microprocessadores que trocavam 2.500 tipos de mensagens de dados. Uma das razões para este crescimento tecnológico é o surgimento de novos *hardwares* e *softwares* que facilitaram a introdução de funções cujo desenvolvimento seria oneroso ou mesmo impossível utilizando apenas componentes mecânicos e hidráulicos. Neste ambiente, o *Controller Area Network* (CAN) tem sido o padrão de fato com mais de 400 milhões de nodos vendidos a cada ano. Mas segundo alguns autores, o CAN não é adequado para uso nos futuros sistemas X-By-Wire que possuem características *hard real-time* por natureza e exigem comunicação de alta velocidade determinística e robusta. Uma das iniciativas para atender às necessidades dessas aplicações foi o desenvolvimento do protocolo FlexRay, frequentemente citado como sendo o protocolo que vai se tornar predominante para as comunicações em sistemas automotivos [Navet and Simonot-Lion 2008].

No entanto, atualmente a disponibilidade comercial de sensores e atuadores com interface FlexRay é reduzida ou mesmo inexistente. Pode ser citado como um exemplo dessa deficiência um catálogo da Bosch Motorsport [Bosch 2010], que oferece diversos dispositivos com interfaces CAN mas nenhum com FlexRay. Um projeto em que o FlexRay seja definido como principal protocolo e no qual se pretenda utilizar sensores e atuadores comerciais possivelmente necessitará de *gateways* para gerenciar a troca de dados entre dispositivos interligados a segmentos de rede que utilizam protocolos diferentes. Para um bom projeto de uma rede veicular, deve-se levar em consideração não apenas o impacto destes *gateways*, mas também as características dos próprios protocolos, como sua velocidade e o formato dos quadros.

Este artigo apresenta os primeiros resultados de um projeto cujo principal objetivo é propor técnicas para o projeto de redes veiculares onde *gateways* gerenciam a troca

de dados entre dispositivos localizados em segmentos de rede que utilizam os protocolos FlexRay e CAN. O trabalho foi motivado pelas necessidades de um projeto maior no qual está sendo desenvolvido um veículo conceitual chamado Flue: um triciclo inclinável para uso urbano com capacidade para dois passageiros (Figura 1). O objetivo é desenvolver um veículo leve e que utilize sistemas *X-by-Wire* para melhorar as características de consumo e baixa emissão de poluentes [de Souza Vieira et al. 2009]. Sendo que a literatura indica o FlexRay para uso em sistemas *X-by-Wire* [Navet and Simonot-Lion 2008], este foi escolhido como sendo o principal protocolo de comunicação do veículo. Mas devido a dificuldade de obtenção de sensores comerciais com interface FlexRay optou-se por utilizar dispositivos CAN disponíveis no mercado. As leituras são enviadas para as *Electronic Control Units* (ECU's) localizadas no segmento FlexRay através de *gateways*, evitando assim o esforço necessário para o desenvolvimento e validação de sensores específicos.

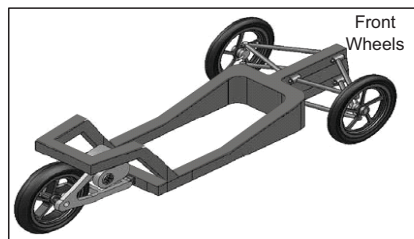


Figura 1: Ilustração do veículo conceitual [de Souza Vieira et al. 2009]

2. FlexRay

O FlexRay é um protocolo desenvolvido por um consórcio de diversos fabricantes com o objetivo de suceder ao CAN em aplicações que exigem determinismo, sincronização, largura de banda e confiabilidade, tais como sistemas *X-by-Wire*.

O controle de acesso ao meio (*Media Access Control*, MAC) do FlexRay é baseado em um ciclo de comunicação periódico com duração predefinida T_{bus} . Um ciclo do FlexRay contém um Segmento Estático (*Static Segment*, ST), um Segmento Dinâmico (*Dynamic Segment*, DYN) e dois segmentos de protocolo *Network Idle Time* (NIT) e *Symbol Window* (SW). Tanto o ST como o DYN são compostos por vários *slots* de tempo, controlados por contadores exclusivos. Durante a fase de projeto é atribuído um identificador *FrameID* para cada mensagem que pode ser transmitida em um *cluster*. Em um dado *slot* ST ou DYN apenas um nó é autorizado a enviar dados para o barramento, e esse é o nó que contém a mensagem com o identificador igual ao valor atual do contador de *slots*. A Figura 2 ilustra um ciclo de comunicação FlexRay.

Um ciclo de comunicação FlexRay sempre contém um ST. Dentro do ST a arbitragem das mensagens é feita através do Time Division Multiple Access (TDMA). O ST contém um número configurável de *slots* estáticos, sendo que todos têm uma duração idêntica. Se uma mensagem não estiver pronta no início de seu *slot*, este permanece vazio e nenhuma outra mensagem pode ser enviada [FlexRay 2005].

Assim como no ST, o DYN tem tamanho fixo e é dividido em *slots*, sendo os identificadores dos *slots* atribuídos aos nós da rede. No entanto, a coordenação de transmissão utiliza um método mais flexível, chamado por alguns autores de *Flexible TDMA* (FTDMA). Neste método, o DYN é dividido em *minislots* (MS) que possuem tamanho

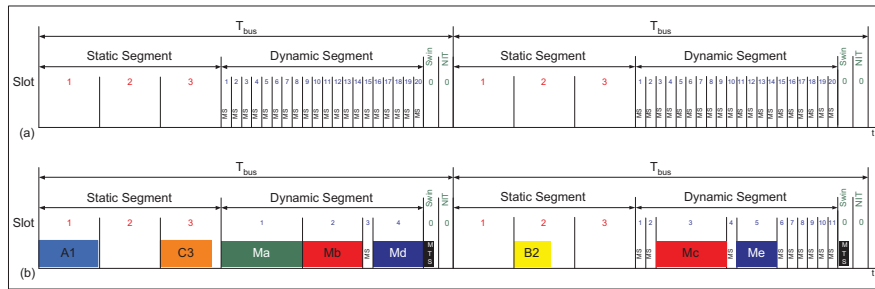


Figura 2: Ciclo de Comunicação FlexRay [Pop et al. 2008]

igual. Assim como no ST, é atribuído um identificador para cada mensagem, e este identificador reflete a prioridade da mensagem. O comprimento de uma mensagem dinâmica pode ser arbitrário mas esta deve caber dentro do DYN. No início de um segmento dinâmico, é permitida a transmissão da mensagem de maior prioridade. Se a mensagem não estiver pronta no início do seu *slot* um quadro vazio com o tamanho de um MS é transmitido. Após a transmissão o acesso é dado para a próxima mensagem, que é enviada se estiver pronta e se couber na parte restante do segmento DYN. Caso contrário, é enviado um quadro vazio com tamanho de um MS. Este processo é repetido para todas as mensagens, ou até o fim do segmento DYN.

Na Figura 2.b o comportamento do FlexRay é exemplificado. As mensagens *A1*, *B2* e *C3* são alocadas no ST, e *Ma*, *Mb*, *Mc*, *Md*, *Me* são alocadas no DYN. Em primeiro lugar, considere-se o comportamento do ST. No primeiro ciclo *A1* e *C3* são enviadas. No segundo ciclo, *B2* (que por algum motivo não estava pronta no primeiro ciclo) é enviada. Considerando-se o DYN, no primeiro ciclo são enviadas *Ma* e *Mb*. Para *Mc* não existe o número necessário de MS, de forma que esta é postergada para o ciclo seguinte e apenas um MS é utilizado. A mensagem *Md* é enviada, não restando mais MS no ciclo. No segundo ciclo, dois MS são utilizados por não existirem mensagens para *Ma* e *Mb*. *Mc* é enviada, outro MS é transmitido devido a *Md* e, por fim, *Me* é enviada.

Na literatura o ST é abordado entre outros em [Schmidt and Schmidt 2008b] e [Grenier et al. 2008]. Aspectos relativos ao DYN são discutido em [Pop et al. 2008], [Schmidt and Schmidt 2008a] e [Chokshi and Bhaduri 2010].

3. CAN

O *Controller Area Network* (CAN) é um barramento de rede projetado para proporcionar uma comunicação simples, robusta e eficiente e é atualmente o padrão de fato para aplicações automotivas. O CAN está presente nos diversos domínios de um veículo, com velocidades variando de 125kbit/s até 1 Mbit/s. Cada mensagem possui um identificador numérico cujo valor determina sua prioridade. Um quadro CAN padrão pode conter até 8 bytes de dados. Um quadro CAN padrão pode conter até 8 bytes de dados de carga útil.

Falando de forma livre, projetar uma rede CAN consiste em atribuir identificadores para as mensagens e verificar se o sistema é escalonável. Um método comum de verificar a escalonabilidade do sistema é calcular o tempo de resposta (*worst-case response time*, wcr) de cada mensagem e então comparar o wcr com seu deadline. Um método bem conhecido para o cálculo do wcr de mensagens CAN é apresentado em [Davis et al. 2007]. Já a atribuição dos identificadores pode ser feita, por exemplo, com a

implementação `Ada Assign_Pri` apresentada em [Burns and Wellings 2001].

4. Gateway

Dos trabalhos existentes na literatura que são relacionados a *gateways* para redes veiculares com FlexRay, pode-se citar [Da 2007] onde é abordado o projeto de *gateways* seguindo o padrão AUTOSAR, [Seo et al. 2008a] e [Seo et al. 2008b], onde são reportadas implementações de *gateways* em placas de desenvolvimento. Entretanto, nenhum dos trabalhos citados aborda questões como análise do tempo de resposta ou de escalonamento.

5. Questões Relacionadas ao Projeto de Redes Veiculares com Gateways

Devido às características dos componentes presentes em uma rede veicular, existem vários pontos e problemas que devem ser considerados no projeto de um sistema que utilize *gateways*. Existem aspectos relacionados especificamente aos *gateways*, como por exemplo os algoritmos dos softwares que precisam considerar as diferenças de quadros e velocidades dos protocolos, ou questões relativas ao uso de padrões como o AUTOSAR [Da 2007]. Outros aspectos dizem respeito especificamente aos sensores e atuadores. Este trabalho não aborda questões relacionadas especificamente aos *gateways* ou dispositivos de forma isolada, mas sim questões relacionadas ao projeto da rede como um todo. Apesar da literatura apresentar trabalhos que abordam todos os pontos que serão levantados aqui, pelo nosso conhecimento não existem trabalhos que apresentem uma solução completa para o problema.

Para um melhor entendimento dos aspectos que serão descritos abaixo, será utilizada como exemplo uma rede veicular composta por um segmento FlexRay de 1Mbps que também atua como *backbone*, um segmento CAN-B com velocidade de 50kbp/s e um segmento CAN-C de 500kbp/s, todos interligados por *gateways* (Figura 3).

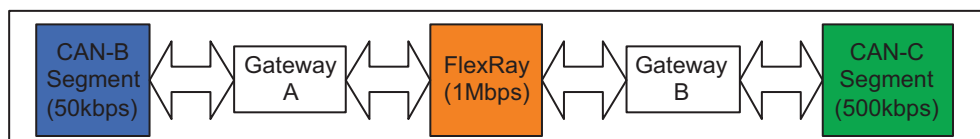


Figura 3: Exemplo de rede com três segmentos

Pode-se dizer que o projeto de uma rede é uma busca de definições para diversos parâmetros, sendo que as escolhas para um dos parâmetros podem influenciar diretamente as escolhas que poderão ser feitas para outros parâmetros. No exemplo, considere-se uma mensagem m gerada por um dispositivo conectado ao segmento CAN-C. A mensagem m também é utilizada por um dispositivo conectado ao segmento CAN-B. Tanto o tamanho do ciclo FlexRay como o modo de transmissão dos dados m no FlexRay (no ST ou no DYN) tem impacto direto na escolha da prioridade dos quadros que conterão m nos barramentos CAN, visto que estas prioridades devem ser altas o suficiente para evitar que m perca seu deadline.

Uma opção para a definição de todos os parâmetros da rede seria através de Programação Linear Inteira onde, obedecendo a restrições fornecidas e informações sobre as mensagens, a resolução de uma formulação desse como resultado definições ótimas

para um sistema completo. No entanto este tipo de técnica é computacionalmente cara, podendo ser inviável para sistemas grandes. Além disso, um sistema resultante pode ser válido apenas para o conjunto inicial de mensagens, sendo necessária a obtenção de um novo conjunto completo de definições mesmo para pequenas modificações. Por estes motivos esta abordagem não será explorada.

Neste trabalho optou-se por procurar uma abordagem com custo computacional baixo que permita a criação de sistemas mais flexíveis (que não necessitem a alteração do sistema como um todo no caso de mudanças). Para que este objetivo seja atingido, uma das etapas foi o levantamento de questões que podem influenciar nas definições para os diversos parâmetros de uma rede. Até o presente momento, as questões identificadas foram: **a)** Como considerar a interferência dos *gateways* no cálculo do wvert de uma mensagem? **b)** É possível a formação de filas de mensagens nos *gateways* devido ao fato dos segmentos possuírem velocidades diferentes? Se sim, como isso pode ser evitado? **c)** Como definir os tamanhos para o ciclo e os segmentos do FlexRay, se considerada a rede como um todo? **d)** Em qual segmento do FlexRay alocar uma mensagem que é originada em um dos barramentos CAN e que deve ser enviada através de um *gateway*? Quais os fatores devem ser levados em conta na decisão sobre a alocação? **e)** Para as mensagens que são transmitidas de um barramento CAN para outro, sem serem aproveitadas por dispositivos no barramento FlexRay, vale a pena fazer algum tipo de agrupamento de mensagens? Qual o impacto deste agrupamento?

No atual estágio deste projeto estão sendo feitos estudos sobre o uso de técnicas de particionamento de deadlines como ferramenta para a definição dos parâmetros de uma rede veicular. Para isso, as técnicas apresentadas em [Kao and Garcia-Molina 1997] tem sido modificadas de forma a abranger os pontos apresentados no parágrafo anterior. Entretanto, os estudos tem esbarrado em um problema relacionado à literatura disponível sobre o FlexRay: até o presente momento, todos os estudos que abordam o cálculo do wvert de uma mensagem do DYN tem sido questionados por apresentarem abordagens que resultam em valores ou otimistas ou muito pessimistas [Chokshi and Bhaduri 2010].

6. Resultados Experimentais

Visando ilustrar alguns dos resultados já obtidos neste trabalho, apresentamos um pequeno exemplo que utiliza alguns dos sensores e atuadores do projeto Flue.

Msg	Descrição	C	T	D	Bus	R
m1	Speed Sensor	8	10.0	10.0	CAN ⇒ FR	0.0570 ms
m2	Yaw-rate Sensor	3	10.0	10.0	CAN ⇒ FR	0.0544 ms
m3	Incln and Acc	8	10.0	10.0	CAN ⇒ FR	0.0570 ms
m4	DataLog 1	8	20.0	20.0	CAN ⇒ FR	0.0570 ms
m5	DataLog 2	6	20.0	20.0	CAN ⇒ FR	0.0560 ms
m6	Left Tilt Ac.	2	5.0	5.0	FR	0.0528 ms
m7	Right Tilt Ac.	2	5.0	5.0	FR	0.0528 ms
m8	Steering Pos.	2	10.0	10.0	FR	0.0528 ms

Tabela 1: Exemplo de Sistema (C em bytes, T, D e R em ms)

Neste exemplo, sensores conectados a um barramento CAN de 50kbp/s enviam suas leituras para dispositivos em um barramento FlexRay de 1Mbp/s. Para o cálculo do wvert R de cada mensagem foi utilizada a estratégia UD [Kao and Garcia-Molina 1997]

em conjunto com as equações presentes nos trabalhos citadas neste documento. Foi considerado um $wert R_{GW}$ do *gateway* como sendo 0,1ms. Para o FlexRay, foram considerados um T_{bus} de 5ms, um ST com 3ms e um DYN com 2ms. Todas as mensagens foram alocadas no ST. A Tabela 1 apresenta os resultados deste exemplo.

7. Agradecimentos

Ao CNPq, CAPES e PPGEAS/UFSC pelo suporte financeiro, e ao Dr. Rodrigo Vieira e demais membros do LI pela ajuda sobre aspectos relacionados ao triciclo.

Referências

- Bosch (2010). Bosch Motorsport Catalog Edition 2010/1. Available at <http://www.bosch-motorsport.de/>.
- Burns, A. and Wellings, A. (2001). *Real-time Systems and Programming Languages: ADA 95, Real-time Java, and Real-time POSIX*. Addison Wesley, third edition.
- Chokshi, D. and Bhaduri, P. (2010). Performance Analysis of FlexRay-based Systems Using Real-Time Calculus, Revisited. In *Symposium On Applied Computing*.
- Da, Z. W. (2007). Performance Analysis of AUTOSAR vehicle Network Gateway. Master's thesis, Ireland.
- Davis, R., Burns, A., Bril, R., and Lukkien, J. (2007). Controller Area Network (CAN) Schedulability Analysis: Refuted, Revisited and Revised. *Real-Time Systems*.
- de Souza Vieira, R., Padilha, R., Nicolazzi, L., and Roqueiro, N. (2009). Five Degrees of Freedom Model for a Tilting Three-Wheeled Narrow Vehicle. *IEEE Transactions on Vehicular Technology*.
- FlexRay (2005). FlexRay Communications System Protocol Specification Version 2.1. Available at <http://www.flexray.com/>.
- Grenier, M., Havet, L., and Navet, N. (2008). Configuring the Communication on FlexRay - The Case of the Static Segment. *Proceedings of ERTS*.
- Kao, B. and Garcia-Molina, H. (1997). Deadline Assignment in a Distributed Soft Real-Time System. *IEEE Trans. Parallel Distrib. Syst.*
- Navet, N. and Simonot-Lion, F. (2008). *Automotive Embedded Systems Handbook*. CRC.
- Pop, T., Pop, P., Eles, P., Peng, Z., and Andrei, A. (2008). Timing Analysis of the FlexRay Communication Protocol. *Real-Time Systems*.
- Schmidt, E. and Schmidt, K. (2008a). Message Scheduling for the FlexRay Protocol: The Dynamic Segment. *IEEE Transactions on Vehicular Technology*.
- Schmidt, K. and Schmidt, E. (2008b). Message Scheduling for the FlexRay Protocol: The Static Segment. *IEEE Transactions on Vehicular Technology*.
- Seo, S., Kim, J., Moon, T., Hwang, S., Kwon, K., and Jeon, J. (2008a). A Reliable Gateway for In-vehicle Networks. *Proceedings of the 17th World Congress The International Federation of Automatic Control*.
- Seo, S., Moon, T., Kim, J., KIM, S., Son, C., Jeon, J., and Hwang, S. (2008b). A Gateway System for an Automotive System: LIN, CAN, and FlexRay. *6th IEEE International Conference on Industrial Informatics INDIN*.