

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciências da Computação

**SISTEMA DE TEMPO REAL**  
**PARA PREVENÇÃO DE COLISÃO EM CRUZAMENTOS**

por

Kattiana Constantino

Samir Palumbo Khalifa

Vitor Venâncio

kattiana@gmail.com

khalifa@dcc.ufmg.br

vitorvd@dcc.ufmg.br

Apresentado como proposta de trabalho na disciplina de **Sistemas de Tempo Real**

Prof.: **Sergio Vale Aguiar Campos**

Belo Horizonte

2013 / 2º semestre

## 1. INTRODUÇÃO

O cenário que engloba o trânsito em ruas de uma cidade proporciona observações interessantes sobre os acontecimentos envolvidos nesse ambiente. Acidentes entre veículos não são raros e na maioria das vezes envolve erro humano que poderia ser evitado com implementações de comportamentos automáticos nas entidades mecânicas.

Uma situação ideal pode ser ilustrada com carros que agem automaticamente de acordo com as características que o trânsito proporciona, principalmente em cruzamentos ou situações de risco em geral. E, seguindo essa modelagem, um sistema de tempo real deve existir para que os componentes sejam monitorados e comandados de maneira eficaz e correta.

A proposta desse trabalho é criar um simulador de trânsito com características de tempo real. Tal simulador possui um sistema centralizado cuja função é orientar os motoristas sobre qual a melhor decisão a ser tomado ao aproximar-se de um cruzamento. O objetivo nesse contexto é evitar colisões com outros veículos e, na medida do possível, minimizar o tempo de espera para atravessar um cruzamento com segurança.

## 2. CARACTERÍSTICAS DO SIMULADOR

As principais características do simulador são:

- **Mapa:** o mapa conterá toda lógica de trânsito funcionando em tempo real. O mapa será composto de quarteirões cujas dimensões serão definidas em um arquivo de entrada. Modelaremos esse mapa como uma grade onde cada célula será um dos seguintes componentes:
  - Via;
  - Cruzamento;
  - Calçada.
- **Radares:** Radares existirão antes de qualquer cruzamento e enviarão informações para o sistema. Os radares detectarão a direção de cada veículo e o tempo para estes chegarem ao cruzamento.
- **Vias:** As vias serão de mão dupla e contínuas. O fim de uma via denota o começo dessa mesma via. Cada veículo terá tráfego livre pelas vias e terá uma velocidade constante. Ao passar por um cruzamento, o veículo decide aleatoriamente a próxima direção a ser seguida.
- **Carros:** O comportamento padrão dos carros é parar impreterivelmente antes de cada cruzamento e esperar retorno do sistema para continuar sua trajetória. Nesse caso, essa tarefa apresenta uma característica *soft real-time* pois o carro ficará ocioso mas não haverá colisão.
- **Ambulâncias:** Uma outra situação desse trabalho envolve observar como o sistema se comporta no cenário onde existem ambulâncias, ou seja, veículos impetuosos que não esperam a decisão do sistema. Nesse caso, temos uma situação *hard real-time* pois se o sistema não responder a tempo haverá uma colisão.

Serão avaliados pontos de quebra do sistema nos quais uma quantidade excessiva de veículos (carros e ambulâncias) seja colocada no simulador. Nessa situações o nosso sistema irá falhar e colisões irão ocorrer mas os limites do sistema serão determinados com experimentos previstos para a etapa posterior a implementação.

### 3. CARACTERIZAÇÃO DO SISTEMA DE TEMPO REAL

O sistema será regido por um relógio cujo iterador será incrementado a cada ciclo. Todas as referências temporais feitas nessa seção devem ser interpretadas como ciclos de relógio (quando no contexto de intervalo de tempo) ou valor do iterador do relógio (quando no contexto de um valor de tempo).

No simulador todos os veículos praticam uma velocidade constante de percorrer uma célula do mapa por ciclo de relógio.

#### 3.1. TAREFAS DO SISTEMA

O sistema terá de tratar diferentes tipos de tarefas que basicamente envolvem um processamento de informação e envio ou recebimento de sinais. Todas essas tarefas são aperiódicas e estão listadas a seguir:

- **TratarInformacaoRadar:** Recebe as informações do radar gerada pela detecção de um veículo passando. Tal informação é uma mensagem contendo:
  - O identificado do veículo;
  - A direção (norte, sul, leste, oeste) em que o veículo está se movimentando;
  - O tempo para a chegada do veículo ao cruzamento subsequente.

O tempo para chegar ao cruzamento é determinado pela distância para o cruzamento pois todos os veículos praticam uma velocidade constante. Tendo como base essas informações o sistema calcula a possibilidade de colisão nos cruzamentos e assim decidirá qual informação enviará para o veículo. O tempo de sistema necessário para realização dessa tarefa será de **1 a 10** ciclos de relógio.

- **SinalizarCarro:** Informa a um veículo que sempre pára que ele está livre para passar no cruzamento a partir de um tempo. O tempo de sistema necessário para realização dessa tarefa será de **1 a 5** ciclos de relógio.
- **SinalizarAmbulância:** Informa a um veículo que sempre segue que ele deve parar por um tempo. O tempo de sistema necessário para realização dessa tarefa será de **1 a 5** ciclos de relógio. Nesse caso, se o tempo de resposta for superior ao deadline da tarefa (tempo em que a ambulância chegará ao cruzamento) essa tarefa não passará no teste de aceitação e não será tratada pelo sistema. Essa é uma decisão de que será discutida posteriormente.

O tratamento de tarefas que envolver veículos impetuosos deve ter uma prioridade maior do que a dos veículos cautelosos para que o sistema priorize a prevenção de colisões.

As tarefas tem início quando um veículo é detectado pelo radar. Nesse instante, o sistema recebe as informações sobre a direção que esse veículo está seguindo, ou seja, em qual cruzamento ele acabará entrando e quanto tempo ele demorará para chegar em tal cruzamento.

O tempo de sistema necessário para a execução de cada tarefa será definido aleatoriamente entre um intervalo de valores pois isto irá simular aspectos da realidade onde o atraso na troca de mensagens e no processamento pode sofrer alguma variação. Outro motivo para essa decisão é que isso torna o escalonamento de tarefas mais heterogêneo e interessante.

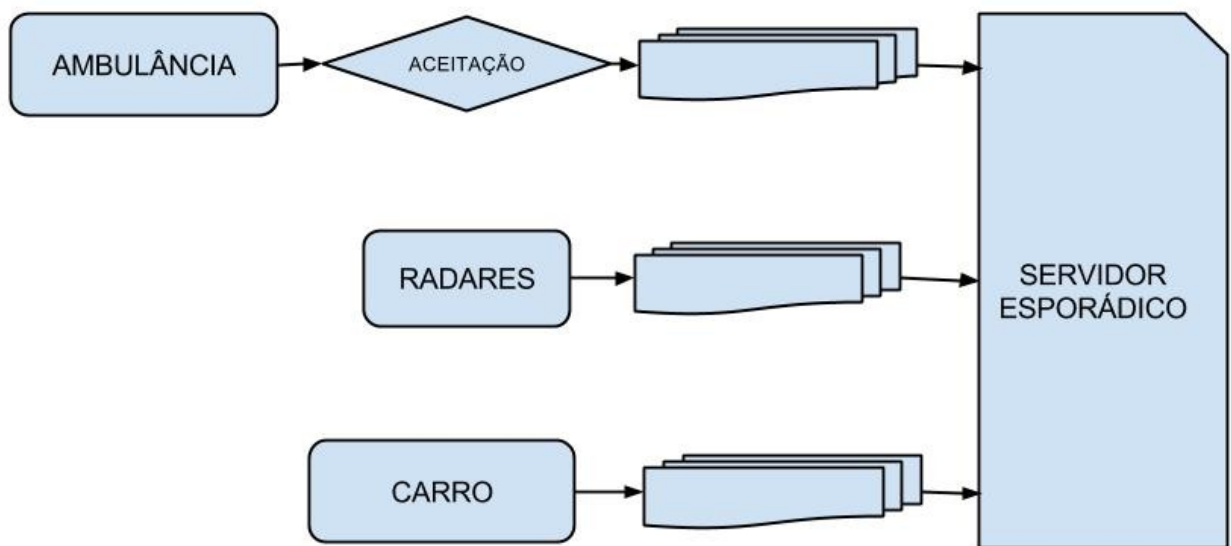
### **3.2. SERVIDOR ESPORÁDICO**

O nosso sistema de tempo real será modelado com um servidor esporádico. Tal decisão foi tomada pois todas os processos existentes não são periódicos e um servidor esporádico é uma solução possível e eficaz para este tipo de situação. Tal servidor tratará a tarefa de enviar a resposta para a ambulância (veículo impetuoso) como a tarefa de maior prioridade. A tarefa de responder ao carro (veículo cauteloso) será a tarefa de menor prioridade. Por sua vez, a tarefa de receber e processar informações dos radares serão de prioridade intermediária. Tais prioridades foram definidas para que o sistema tenha como objetivo maior evitar colisões e como objetivo secundário minimizar o tempo ocioso de espera em cruzamentos.

O tratamento dos carros cautelosos é visto como uma tarefa aperiódica, por estarmos tratando de uma situação de soft real-time. As tarefas de carros cautelosos devem ser completadas o mais rápido possível com tanto que não atrapalhem o tratamento das leituras do radar e muito menos das respostas para carros impetuosos.

Para as tarefas geradas pelos carros impetuosos decidimos que haverá um teste de aceitação como consequência da característica hard real-time dessas tarefas. Como se tratam de tarefas esporádicas se o teste determinar que o sistema não conseguirá completar essa tarefa simplesmente não haverá tratamento.

A figura abaixo é um diagrama que resume os conceitos principais do sistema proposto.



#### **4. DETALHES TÉCNICOS E CRITÉRIOS DE AVALIAÇÃO**

Dentre os detalhes técnicos de implementação do simulador temos que as informações necessárias para a criação do mapa durante a inicialização serão passadas por um arquivo de entrada que conterá informações sobre as distâncias entre cruzamentos, posicionamento de radares e número de veículos impetuosos e cautelosos presentes no cenário.

Como existem situações de hard e soft real-time gerado pelo comportamento dos veículos podemos avaliar o sistema em termos de ineficiência, ou seja, funções de utilidade que foram atendidas fora de seu ponto máximo e também em termos de falha, ou seja, deadlines que foram perdidos gerando situações de quebra do sistema.

##### **Para os veículos que sempre param:**

- A medida de ineficiência será o tempo ocioso do veículo enquanto espera a informação de seguir adiante.
- O sistema mitigará esta medida maximizando o tempo do veículo em movimento.

##### **Para os veículos que sempre seguem:**

- A medida de falha do sistema será o número de possíveis colisões em cruzamento;
- O sistema mitigará esta medida minimizando o número de colisões;
- Uma ideia já discutida para extensão da implementação é incluir no sistema a função que bloqueará temporariamente a via ou cruzamento onde ocorreu a colisão e tratará as devidas implicações desse acidente.

Tais critérios servirão para a avaliação do sistema prevista quando a implementação for terminada.

## 5. IMPLEMENTAÇÃO

A implementação está sendo feita na linguagem Java de forma colaborativa entre os participantes do grupo. O repositório do projeto pode ser encontrado em:

<https://github.com/khalifa/RTCollisionAvoidance>

Uma das frentes de implementação foi criar um mapa na forma de um grafo onde os vértices são os radares e as arestas são as vias. Os veículos passaram a ter origem e destino, então foi desenvolvido o algoritmo de Busca em Largura, para encontrar o menor caminho entre os dois pontos. Cada veículo agora sabe o seu percurso e quando chegar a um radar (vértice do grafo), esperará ou não para seguir em frente.

Ainda nessa etapa, foi criada uma fila com todos os radares para que esse seja tratado pelo servidor.

As próximas etapas dessa frente é criar as filas para os veículos cautelosos e para os veículos de ambulância e o algoritmo de escalonamento de tarefas para o tratamento das filas. Uma extensão desse cenário é ativar ou desativar algum(s) radares, assim as vias para chegar ao(s) radar(es) serão disponibilizadas ou interditadas, respectivamente; e em tempo real um novo percurso deverá ser recalculado.



## **6. REFERENCIAL TEÓRICO**

THOMAS, Jones. Real-time probabilistic collision avoidance for autonomous vehicles, using order reductive conflict metrics. Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics. 2003.

AMMOUN, S; Nashashibi, F.; Laugeau, C. Real-time crash avoidance system on crossroads based on 802.11 devices and GPS receivers. Centre de Robotique de l'Ecole des Mines de Paris. 2008.

BRANNSTROM, Mattias. A Real-time Implementation of an Intersection Collision Avoidance System. Proceedings of the 18th World Congress The International Federation of Automatic Control Milano (Italy) August 28 - September 2, 2011.

MINGUEZ, J.; L. Montano. NEARNESS DIAGRAM NAVIGATION (ND): A NEW REAL TIME COLLISION AVOIDANCE APPROACH. Computer Science and Systems Engineering 2005. International Conference of Robotic and Systems.