

Sistema de Semáforos Inteligentes suportado por comunicações V2X

Catarina Pereira^{1,2[PG53733]}, Inês Neves^{1,2[PG53864]} e Leonardo Martins^{1,2[PG53996]}

¹ Uninversidade do Minho, Campus de Azurém, Guimarães, Portugal

² Mestrado em Engenharia de Telecomunicações e Informática

Resumo Este projeto desenvolveu um sistema de comunicação veículo-para-veículo (V2X) utilizando a plataforma Eclipse MOSAIC, com o objetivo de criar um ambiente onde veículos possam enviar informações de estados e receber comandos de controlo de tráfego (sinais de luz), permitindo uma coordenação eficiente em cruzamentos.

O sistema é composto por duas Road-Side Units (RSUs) instaladas nas interseções, que se comunicam com os veículos através de mensagens V2X. As RSUs recolhem informações sobre o tráfego local e coordenam o controlo dos semáforos virtuais de forma autónoma, sem a necessidade de um semáforo físico central.

Foram implementadas aplicações para veículos e RSUs, que se comunicam através de primitivas de comunicação e mensagens protocolares definidas (PDUs), como StatusMessage enviada pelos veículos com informações de status, e LightMessage enviada pelas RSUs para informar os veículos sobre o estado dos semáforos.

Keywords: V2X · Eclipse MOSAIC · RSU · Semáforos virtuais · Mensagens V2X · StatusMessage · LightMessage · Controlo de tráfego · Simulações de tráfego · Fluxo de tráfego.

1 Introdução

Este relatório faz parte da Unidade Curricular de Redes Veiculares, do 2^o semestre do 1^o ano do Mestrado em Engenharia de Telecomunicações e Informática. Este projeto foi desenvolvido como resposta a um problema apresentado pelos docentes.

Este projeto aborda o desenvolvimento de um sistema de comunicação veículo-para-veículo (V2X) utilizando a plataforma Eclipse MOSAIC [1] e com ajuda do API do Eclipse MOSAIC [2]. O sistema consiste em três componentes principais: o Veículo, a RSU (Road-Side Unit) e o TestApp (uma aplicação de teste para a RSU). O objetivo é criar um ambiente onde veículos possam enviar informações de estados e receber comandos de controlo de tráfego (sinais de luz), permitindo uma coordenação eficiente em cruzamentos.

2 Especificação do sistema e dos protocolos de suporte

O sistema é composto por dois RSU instalados nas interseções (um RSU para cada interseção), que se comunicam com os veículos através de mensagens V2X. As RSUs recolhem informações sobre o tráfego local e coordenam o controlo dos semáforos virtuais de forma autónoma, sem a necessidade de um semáforo físico central.

O sistema é composto por aplicações para veículos e RSUs, que se comunicam através de primitivas de comunicação e mensagens protocolares definidas (PDUs).

Por cada carro presente numa dada estrada da interseção é dado a essa estrada 3 segundos de luz verde com um tempo máximo de 30 segundos para a passagem do tráfego, caso esteja totalmente congestionada.

O RSU avaliará qual estrada tem mais carros, e essa estrada será a primeira a receber o sinal verde. Em seguida, quando essa estrada não tiver mais carros ou quando se completar 30 segundos, o RSU avaliará novamente qual é a estrada mais movimentada, excluindo a que acabou de ter o sinal verde. Esse processo continuará até que todas as estradas tenham tido a sua vez, momento em que a ordem das estradas estará configurada. Se uma estrada não tiver carros, ela será ignorada e a próxima na fila será atendida.

2.1 Primitivas de Comunicação

O sistema utiliza comunicação ad hoc e celular para troca de mensagens V2X. As principais primitivas de comunicação incluem:

- Mensagens V2X enviadas periodicamente pelos veículos, contendo informações como tipo, dimensões, peso, posição (latitude e longitude), velocidade, aceleração e direção. O envio das mensagens possui Multi-Hop de 10 saltos.
- Mensagens enviadas pelas RSUs para informar os veículos sobre o estado dos semáforos (verde ou vermelho).

2.2 Formato das Mensagens Protocolares (PDU)

LightMessage Uma mensagem enviada pela RSU para os veículos, indicando a cor do sinal de trânsito.

- Atributos:
 - `lightcolor`: String representando a cor da luz (e.g., “VERMELHO”, “VERDE”).
- Métodos:
 - `getLightColor()`: Retorna a cor da luz.
 - `getPayload()`: Retorna a carga útil da mensagem.
 - `toString()`: Representação textual da mensagem.
 - `StatusMessage`
 - Uma mensagem enviada pelos veículos para a RSU, contendo informações de status.

StatusMessage Uma mensagem enviada pelos veículos para a RSU, contendo informações de status.

- Atributos:
 - type: Tipo de veículo.
 - dimensions: Dimensões do veículo.
 - weight: Peso do veículo.
 - latitude e longitude: Coordenadas de localização do veículo.
 - speed: Velocidade do veículo.
 - acceleration: Aceleração do veículo.
 - direction: Direção do veículo.
 - road: Identificação da estrada em que o veículo está.
- Métodos:
 - getPayload(): Retorna a carga útil da mensagem.
 - toString(): Representação textual da mensagem.

Interações

- Veículo
 - onStartup(): Configura o módulo ad hoc e celular, e inicializa o veículo.
 - onVehicleUpdated(): Envia uma StatusMessage com os dados atualizados do veículo para a RSU.
 - onMessageReceived(): Processa LightMessage recebidas, controlando o movimento do veículo com base na cor da luz. Também processa StatusMessage recebidas de veículos na mesma estrada
 - onShutdown(): Desliga o veículo, registrando a operação.
- RSU
 - onStartup(): Configura o módulo ad hoc e celular, e inicializa a RSU.
 - processEvent(): Envia um LightMessage com a cor “**VERMELHO**” para todos os veículos e determina a direção com mais veículos para enviar um LightMessage com a cor “**VERDE**”.
 - onMessageReceived(): Processa StatusMessage recebidas, atualizando a lista de veículos e suas direções.
 - onShutdown(): Desliga a RSU, registrando a operação.

3 Implementação

Cada RSU executa uma aplicação que processa as mensagens recebidas dos veículos e mantém um mapa dinâmico do tráfego local. Com base nessas informações, a RSU decide o estado ideal dos semáforos (**verde**, **vermelho**) para cada direção, procurando minimizar o tempo de espera dos veículos.

Quando um veículo se aproxima de uma interseção, a RSU envia uma mensagem V2X informando o estado do semáforo correspondente. O veículo então ajusta a sua velocidade para parar com segurança caso o sinal esteja **vermelho**, ou acelera caso esteja **verde**.

Parâmetros importantes incluem o alcance de comunicação das RSUs, a frequência de envio das mensagens, e os algoritmos de decisão utilizados pelas RSUs para determinar o estado dos semáforos.

A implementação do sistema envolve a configuração de aplicações em veículos e RSUs, utilizando bibliotecas de funções específicas para comunicação e processamento de dados.

- **Detalhes da Implementação:**
 - Aplicações em veículos coletam e transmitem dados em intervalos regulares.
 - RSUs recebem, processam dados e enviam Light Messages.
- **Parâmetros:**
 - Frequência de atualização dos dados dos veículos.
 - Alcance de comunicação das RSUs.
- **Bibliotecas de Funções:**
 - Utilização de bibliotecas de rede para comunicação.
 - Bibliotecas de geolocalização para cálculo de latitude e longitude.

4 Testes e resultados

- **Metodologia de Teste:**
 - Simulações de tráfego com múltiplos veículos e RSUs.
 - Cenários de diferentes condições de tráfego e sinalização.
- **Resultados Obtidos:**
 - Redução de colisões e melhor fluxo de tráfego.
 - Alta precisão na transmissão de dados e na sinalização.

5 Conclusões

O projeto desenvolvido demonstrou a viabilidade e a eficácia de um sistema de comunicação veículo-para-veículo (V2X) utilizando a plataforma Eclipse MO-Saic. Através da implementação de dois RSUs em interseções críticas e da comunicação contínua com veículos, foi possível coordenar os sinais de tráfego de maneira autónoma, resultando numa gestão de tráfego mais eficiente e segura.

Os testes realizados mostraram que o sistema pode reduzir significativamente o número de colisões e melhorar o fluxo de tráfego. A precisão na transmissão de dados e na sinalização foi alta, indicando que as mensagens V2X são uma ferramenta robusta para a coordenação de semáforos e o controlo de tráfego.

- Sucesso na implementação das aplicações de veículos e RSUs.
- Mensagens protocolares eficazes na comunicação e coordenação do tráfego.

6 Trabalho Futuro

Apesar dos resultados promissores, existem várias áreas que podem ser exploradas para melhorar e expandir o sistema:

1. Expansão do sistema para incluir mais tipos de mensagens e sensores.
2. Escalabilidade e Desempenho:
 - Realizar testes em ambientes urbanos mais complexos com um número maior de interseções e veículos para avaliar a escalabilidade do sistema.
 - Otimizar os algoritmos de decisão das RSUs para suportar um maior volume de tráfego sem comprometer o desempenho.
3. Segurança e Privacidade:
 - Implementar medidas de segurança avançadas para proteger as comunicações V2X contra ataques cibernéticos.
 - Garantir a privacidade dos dados transmitidos pelos veículos, especialmente em cenários onde informações pessoais possam ser derivadas.
4. Inteligência Artificial e Machine Learning:
 - Aplicar técnicas de machine learning para melhorar a previsão de tráfego e a tomada de decisão em tempo real das RSUs.
 - Utilizar inteligência artificial para analisar dados históricos e otimizar as rotas e os tempos de sinalização.
5. Feedback dos Utilizadores:
 - Recolha do feedback dos motoristas e outras partes interessadas para identificar áreas de melhoria e ajustar o sistema conforme necessário.
 - Implementar uma interface de utilizador intuitiva para que os motoristas possam entender e interagir com o sistema de forma eficiente.
6. Sustentabilidade:
 - Investigar o impacto ambiental do sistema e explorar maneiras de reduzir o consumo de energia, tanto nos veículos quanto nas RSUs.
 - Considerar a utilização de fontes de energia renovável para alimentar as RSUs.

A continuação deste projeto e a implementação das melhorias sugeridas têm o potencial de transformar significativamente a gestão de tráfego urbano, promovendo um ambiente mais seguro, eficiente e sustentável para todos os usuários das vias.

Referências

1. Eclipse MOSAIC Tutorials, <https://eclipse.dev/mosaic/tutorials/>. Última atualização 29-03-2024, Última vez acedido 24 de maio de 2024
2. Eclipse Foundation, <https://help.eclipse.org/latest/nftopic/org.eclipse.platform.doc.isv/reference/api/index.html>. Última atualização 16-03-2024, Última vez acedido 24 de maio de 2024