

Intelligent Road Safety Based on Fog Computing

Augusto Campos (PG57510), Carlos Silva (PG57518) & Juliana Silvério (PG50006)

Abstract

Este artigo apresenta o desenvolvimento e simulação de uma arquitetura de comunicação veicular baseada em Fog Computing, implementada com recurso à framework Eclipse MOSAIC. O sistema proposto permite que Unidades a Bordo (OBUs) troquem dados com Unidades de Beira de Estrada (RSUs), que posteriormente encaminham a informação para um servidor Fog. O servidor Fog agrega e processa os dados recebidos, obtendo uma visão global e holística do estado da rede, emitindo mensagens de reação destinadas a controlar ou alertar os veículos. O foco do projeto centrou-se no desenvolvimento de um protocolo de encaminhamento multihop personalizado, que garante a comunicação fiável entre os veículos e o servidor Fog, com principal ênfase em alta conectividade e baixa latência. Este protocolo utiliza mensagens de awareness, geradas pelos veículos, para construir um mapa de vizinhança distribuído e determinar os caminhos ótimos de encaminhamento com base no número de saltos e na proximidade às RSUs. Apesar das limitações impostas pelo tempo disponível para o desenvolvimento, o protocolo foi validado com um cenário de proof of concept utilizando o simulador SUMO, no qual veículos regulares reagem à aproximação de um veículo de emergência com base em comandos emitidos pelo Fog. Este cenário demonstra a eficácia do sistema orientado a eventos e estabelece a base para otimizações futuras e uma avaliação quantitativa mais abrangente.

Index Terms

Ad hoc Networks, Vehicular Networks, Fog Computing, V2X, RSU, OBU, Multihop Forwarding, Eclipse MOSAIC, SUMO.

I. INTRODUÇÃO

ESTE projeto teve como objetivo desenvolver e simular uma arquitetura de comunicação veicular utilizando o simulador Eclipse MOSAIC. O sistema implementado permite que veículos (OBUs) troquem mensagens entre si e com unidades de beira de estrada (RSUs), que, por sua vez, retransmitem as mensagens para um servidor de Fog Computing. O Fog processa os dados recebidos, tendo em conta uma visão abrangente dos veículos na sua área de cobertura e, com base nessa análise, tem a capacidade de enviar mensagens de reação aos veículos que se encontrem na rede.

O principal objetivo do nosso grupo foi o desenvolvimento, integração e teste de um protocolo multihop que permitisse a comunicação fiável entre os veículos da rede e o servidor FOG. Consideramos que, para que o servidor FOG possa alertar e atuar eficazmente nos veículos da rede, é absolutamente necessária uma alta taxa de entrega dos pacotes, garantindo assim que o Fog disponha de informações fiáveis e o mais atuais possíveis para suportar a tomada de decisões.

II. ELEMENTOS DO SISTEMA

O sistema implementado assenta em três componentes principais: os veículos equipados com unidades de bordo (On-Board Units), as unidades fixas de beira de estrada (Road-Side Units) e o servidor computacional de proximidade (Fog).

A. OBU – Unidade de Bordo

Cada veículo integra uma OBU que suporta comunicação sem fios V2X (Vehicle-to-Everything), permitindo a troca direta de mensagens com outros veículos e com as RSUs. As OBUs recolhem continuamente dados do veículo, como posição GPS, velocidade, aceleração e direção, que são usados para gerar mensagens de informação transmitidas periodicamente. Estas mensagens são disseminadas por broadcast para os veículos e RSUs ao alcance da OBU. Além disso, as OBUs mantêm um mapa local dos vizinhos na sua proximidade, atualizando-o com as mensagens recebidas.

Existe duas versões da OBU, uma dedicada a veículos sem funcionalidade especial e uma outra para veículos de emergência.

A comunicação utilizada pelas OBUs é exclusivamente ad hoc, permitindo a interligação entre veículos e entre veículos e RSUs sem recorrer a infraestrutura fixa.

B. RSU – Unidade de Beira de Estrada

As RSUs funcionam como nós fixos estrategicamente colocados na berma das estradas, cobrindo áreas específicas da via. O seu papel principal é o de retransmitir mensagens entre veículos e encaminhá-las para o Fog associado, atuando essencialmente como um relay de mensagens. Devido ao protocolo de encaminhamento multihop que implementamos ser baseado na escolha do melhor caminho, espera-se que a carga entre as RSUs não seja equilibrada uniformemente — algumas RSUs podem suportar um tráfego muito superior a outras. Para evitar sobrecarga e garantir a máxima capacidade de transmissão, a arquitetura tenta

retirar às RSUs qualquer processamento computacional significativo, deixando-as livres para receber e enviar um elevado volume de mensagens com baixa latência.

No que respeita aos meios de comunicação, as RSUs usam comunicação ad hoc para enviar e receber mensagens da rede veicular, enquanto a comunicação entre as RSUs e o nó Fog é feita via rede celular. Idealmente, esta ligação seria por cabo, como por exemplo utilizando fibra ótica, para reduzir a latência entre as RSUs e o Fog, mas, devido às limitações do Eclipse MOSAIC, que não possui um módulo de comunicação deste tipo, tiramos partido da comunicação celular e definimos um atraso constante e fixo para a mesma.

C. Fog – Servidor Computacional de Proximidade

O Fog está diretamente ligado às RSUs e realiza o processamento dos dados recebidos, agregando a informação proveniente de vários veículos na sua área de cobertura. Este processamento permite obter uma visão global e em tempo real da situação do tráfego na zona, identificando riscos e gerando alertas de segurança rodoviária. O Fog é responsável por enviar mensagens de reação e alerta aos veículos, através das RSUs, para melhorar a segurança e a eficiência rodoviária. Desta forma, a computação intensiva e a tomada de decisão são centralizadas no Fog.

A comunicação do nó Fog com as RSUs é exclusivamente feita via a rede celular mencionada anteriormente.

III. MENSAGENS

Nesta secção descrevem-se as mensagens utilizadas no protocolo de comunicação veicular implementado, definindo a sua função, conteúdo e fluxo dentro da rede.

A. Beacon Message

A *Beacon Message* é uma mensagem periódica enviada pelas unidades RSU para anunciar a sua presença na rede. Esta mensagem publica a posição da RSU, permitindo aos veículos na sua área de alcance atualizarem o mapa com a localização da infraestrutura fixa.

B. VehInfoMessage

A *VehInfoMessage* é a mensagem que contém a informação do veículo, incluindo dados do plano de dados, como posição GPS, velocidade, aceleração, e direção, bem como informações do plano de controlo, nomeadamente a distância ao RSU mais próximo, o número de saltos necessários para alcançar o RSU, o modo de operação do protocolo (DIRECT ou SEARCH) e o identificador do encaminhador (*FwdId*). Esta mensagem é periodicamente publicada pelo veículo e é reencaminhada pela rede até alcançar o Fog.

C. SlowMessage

A *SlowMessage* é uma mensagem enviada do nó Fog para um veículo específico, com o objetivo de diminuir a velocidade máxima permitida desse veículo. Esta mensagem contém o valor da nova velocidade máxima a ser adotada, e é transmitida através das RSUs e veículos intermediários, assegurando que o veículo destinatário ajuste o seu comportamento em tempo real para aumentar a segurança rodoviária.

D. StopMessage

A *StopMessage* é uma mensagem crítica enviada pelo nó Fog para um veículo, com a instrução de parar imediatamente na berma da estrada. Esta mensagem é utilizada em situações de perigo iminente, onde a paragem do veículo é necessária para evitar acidentes ou para cumprir ordens de controlo de tráfego. Tal como a *SlowMessage*, a *StopMessage* é transmitida via RSUs e encaminhada pela rede veicular até ao veículo alvo.

E. ResumeMessage

A *ResumeMessage* é uma mensagem enviada pelo nó Fog para um veículo, com a instrução de resumir o seu progresso normal. Esta mensagem é utilizada para remotar o normal funcionamento dos veículos após paragem. Esta é também uma mensagem com origem no servidor FOG.

IV. PROTOCOLO DE ENCAMINHAMENTO

Para explicar o protocolo implementado vamos definir uma primeira situação exemplo, e vamos simular como certas mensagens se propagam na rede.

Num primeiro momento temos 4 intervenientes: 2 veículos, 1 RSU e um servidor FOG. O veículo Veh_1 (à direita) está ao alcance da RSU, enquanto o veículo Veh_2 (à esquerda) está fora do alcance da RSU. Definimos também que um qualquer RSU tem uma ligação direta a um servidor FOG sem entraves de distancia.

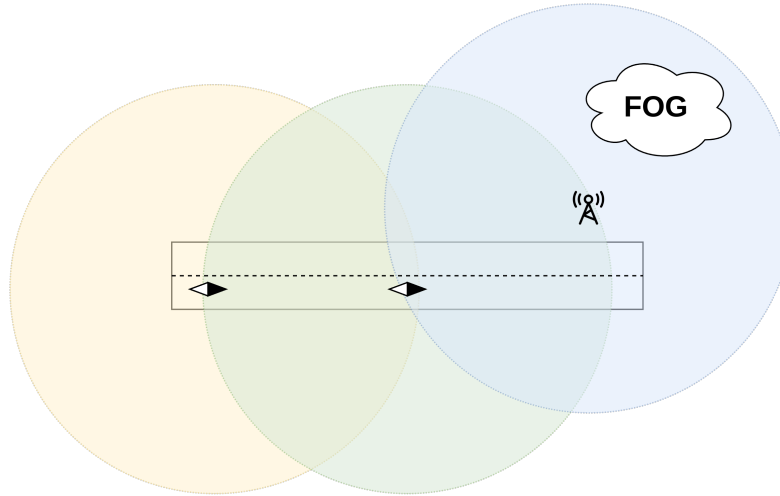


Fig. 1. Área de Alcance.

Em seguida assumimos o envio de uma Beacon Message pela RSU, estas mensagens são periódicas e publicam na rede a presença do RSU que a emite, no exemplo que seguimos, apenas um Veículo (Veh_1) está dentro do alcance do RSU e por isso apenas ele guarda a Posição do mesmo.

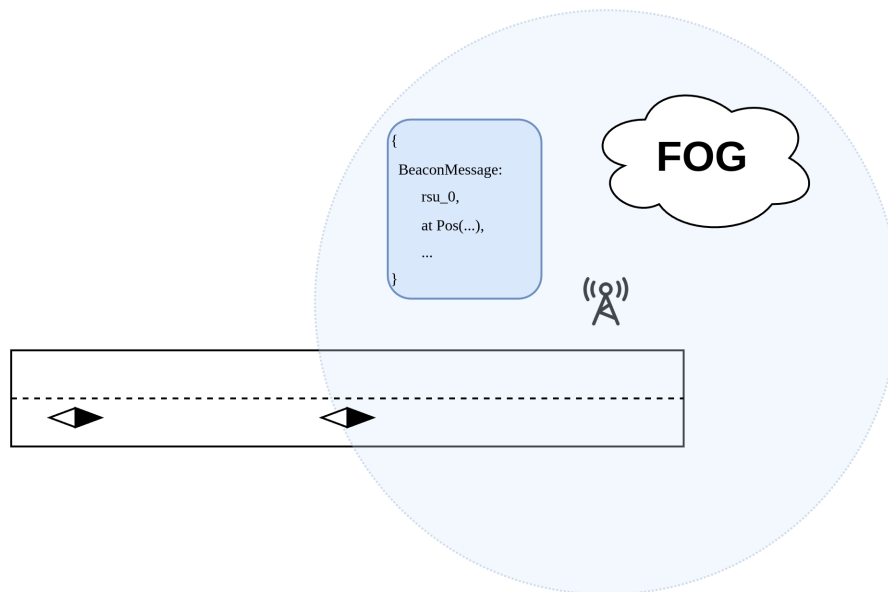


Fig. 2. Mensagem de Beacon.

Em seguida vamos analisar como agora o Veh_1 publica a sua própria mensagem de informação. Para além de informações do plano de dados (posição, velocidade, etc.) esta mensagem contém também informações do plano de controlo (distancia ao RSU, numero de hops para chegar ao RSU, modo do protocolo, identificador do encaminhador), a junção de ambos os planos foi feita com a intenção de tirar também partido da informação do plano de dados para informar a decisão de encaminhamento como em [1], no entanto devido ao período curto de desenvolvimento, isto não foi possível.

Na situação exemplo o Veh_1 publica uma mensagem com a sua distancia ao RSU que é calculada a partir da posição do veiculo e a posição do RSU, o numero de saltos que neste caso é zero, o modo DIRECT e o FwrId é rsu_0.

O objetivo final do protocolo nesta fase é fazer chegar a mensagem ao servidor FOG, assim o veiculo tendo o conhecimento de RSU's ira analisar os mesmos e preferir pelo RSU com menor distancia. O FwrId neste caso é exatamente o RSU do qual o veiculo tem conhecimento.

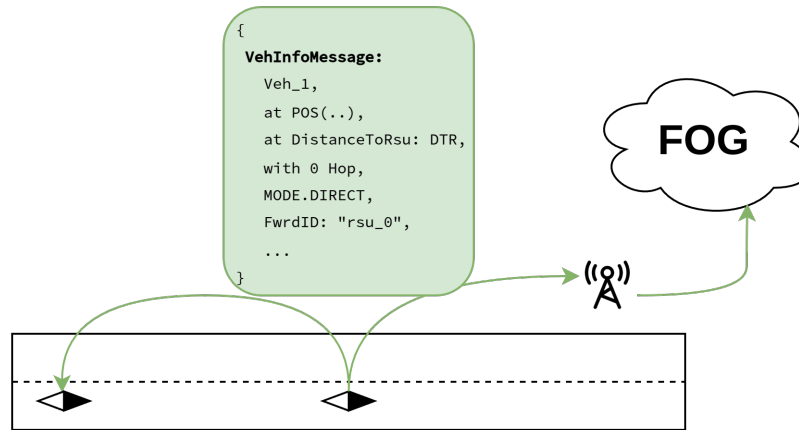


Fig. 3. Mensagem de informação do Veh_1 com conhecimento do RSU.

Em contra ponto no diagrama da figura 4 exemplificamos o que acontece quando o Veh_1 envia uma mensagem sem ter conhecimento do RSU.

A mensagem é publicada com a sua distancia ao RSU de -1, numero de saltos -1, modo SEARCH e o FwrId é "BROADCAST".

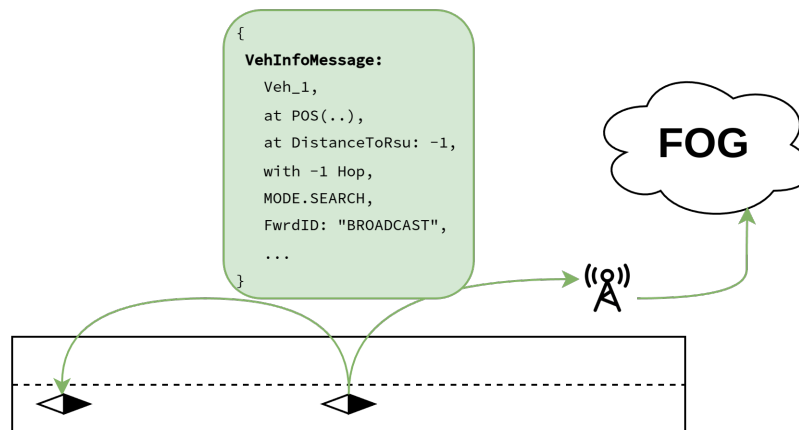


Fig. 4. Mensagem de informação do Veh_1 sem conhecimento do RSU.

O Veh_2 ouve também esta mensagem vinda do Veh_1, ele reage adicionando o veiculo ao seu mapa como vizinho, e guardando a informação que pode ser pensada da seguinte forma : "tenho um vizinho chamado Veh_1, que à distancia DTR tem um RSU e esta a 0 saltos dele".

Em seguida vamos fazer o envio de uma mensagem de informação do Veh_2, este analisa primeiro se conhece algum RSU, como não conhece passa a analisar os vizinhos e verifica se algum deles conhece algum RSU, e em caso afirmativo, analisa também destes qual é o melhor, ou seja, com menos distancia e com menos saltos.

No nosso caso escolhe o Veh_1 e este passa a ser o nosso FwrID da mensagem, ou seja, o próximo salto da mensagem, de maneira a que apenas o Veh_1 realize o reencaminhamento da mensagem.

A mensagem de informação contem também um campo relativo ao publisher, que é o veiculo que esta a publicar a mensagem na rede, este campo é guardado no mapa da rede quando se atualiza a informação de um vizinho e é usado posteriormente para fazer o traceback de mensagens vindas do FOG.

Para alem disso o Veh_2 calcula a sua distancia ao RSU adicionando a distancia de si próprio ao veiculo que é o próximo salto, com a distancia de esse mesmo veiculo ao RSU. Desta maneira se existisse um outro qualquer veiculo que escutasse esta

mensagem, este iria guardar a informação que o melhor caminho para um RSU a passar pelo Veh_2 é um com essa distancia e que esta a 1 salto.

Isto permite construir a rede de forma iterativa, garantindo um invariante fundamental: "Qualquer grupo de veículos e RSUs, em que os elementos estão ligados com pelo menos um outro elemento do grupo, então todos conhecem pelo menos 1 caminho para um RSU", Desta forma garantimos conectividade ao servidor Fog por parte de todos os elementos do grupo, pode existir na nossa rede vários destes grupos que são assim construidos de forma orgânica, e procurando sempre o RSU mais perto e com menos saltos necessários.

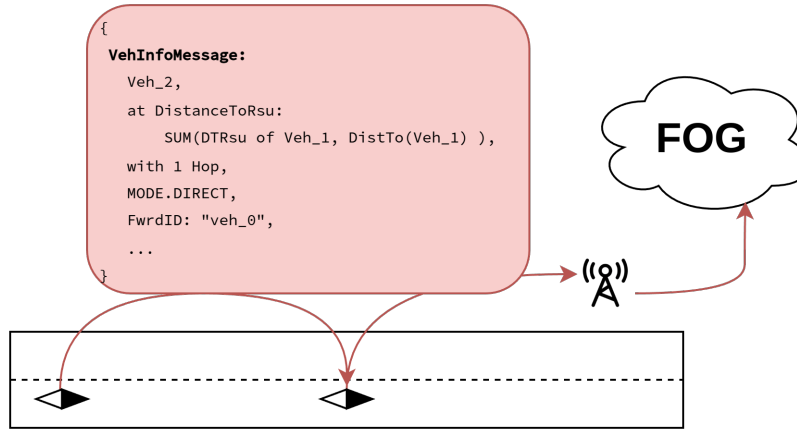


Fig. 5. Mensagem de informação do Veh_2.

Vamos agora exemplificar como uma mensagem vinda do FOG consegue encontrar um veiculo na rede.

Esta definido um delay de mensagens de 200ms para comunicações V2x e Celulares, como uma mensagem pode tomar vários saltos dentro da rede um RTT (Round Trip Time) possível de reação é:

$$RTT = (N_s + 1) \times 2 \times 200 \text{ ms} + D_{\text{RSU-FOG}}$$

onde N_s é o número de saltos, e $D_{\text{RSU-FOG}}$ é o delay entre a RSU e o nó FOG.

No nosso caso exemplo, o RTT teórico para o Veh_2 é 1.2 segundos, neste tempo um veiculo que com velocidade de 50km/h move-se sensivelmente 17 metros. É fácil de perceber que esta mobilidade pode causar problemas nos saltos, em que a conectividade entre os elementos dentro da rede deixam de existir.

No entanto temos também um conhecimento prévio da topologia da rede, e se é verdade que os veículos se movem também é verdade que estes maioritariamente se movem em conjuntos. Assim e tendo como garantia o invariante definido acima, mesmo que um veiculo perca a conectividade pelo caminho que enviou a mensagem de informação, é altamente provavelmente que este tenha um outro caminho pelo qual ele é alcançável.

Na verdade a natureza dinâmica do protocolo que definimos faz com que os elementos do melhor caminho entre um RSU e um veiculo mudem com regularidade mas sempre em procura de minimizar a distancia ao RSU e o numero de saltos para o mesmo.

Vendo agora o percurso de uma mensagem, esta é enviada pelo RSU que também enviou a mensagem de informação, este RSU publica a mensagem em modo SEARCH, em que todos os veículos que escutarem vão tentar fazer a mensagem chegar ao veiculo destino. Se o veículo se encontrar a 0 saltos do RSU, a mensagem é enviada em modo DIRECT.

No nosso caso, o Veh_1 escuta a mensagem que tem por destino o Veh_2 e procura se tem o Veh_2 no seu mapa, se tiver reenvia a mensagem em que o próximo salto é o veiculo que publicou a posição, no nosso caso o próprio Veh_2.

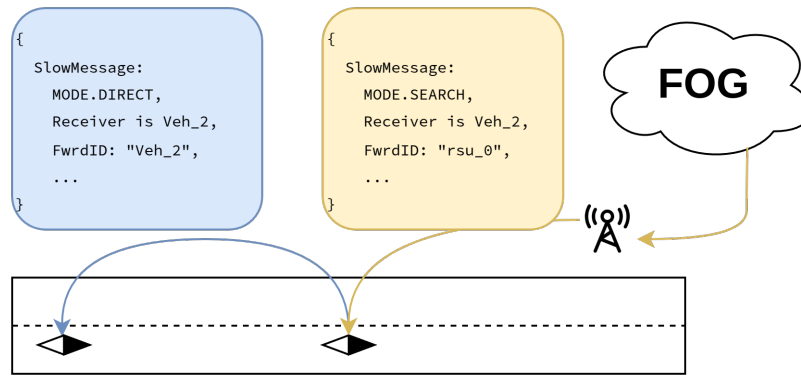


Fig. 6. Mensagem vinda do servidor Fog para o Veh_2.

V. TESTES E RESULTADOS

O objetivo inicial consistia em tirar partido da saída da federation output para extrair estatísticas da simulação, permitindo uma avaliação metódica do protocolo implementado e da sua influência na congestão da rede. No entanto, devido ao tempo limitado disponível para o desenvolvimento do projeto, esta análise não foi possível.

Assim, apresentamos um cenário como proof of concept, executado no simulador SUMO (Simulation of Urban Mobility), envolvendo três veículos (dois com comportamento regular e um veículo de emergência), um RSU e um servidor FOG.

Neste cenário inicialmente os veículos estão em circulação normal (Figura 7), os veículos regulares param para ceder passagem ao veículo de emergência (Figura 8), retomando a circulação assim que este passa (Figura 9). Este comportamento demonstra, de forma simplificada, a aplicação prática do protocolo desenvolvido e das mensagens de eventos criadas.

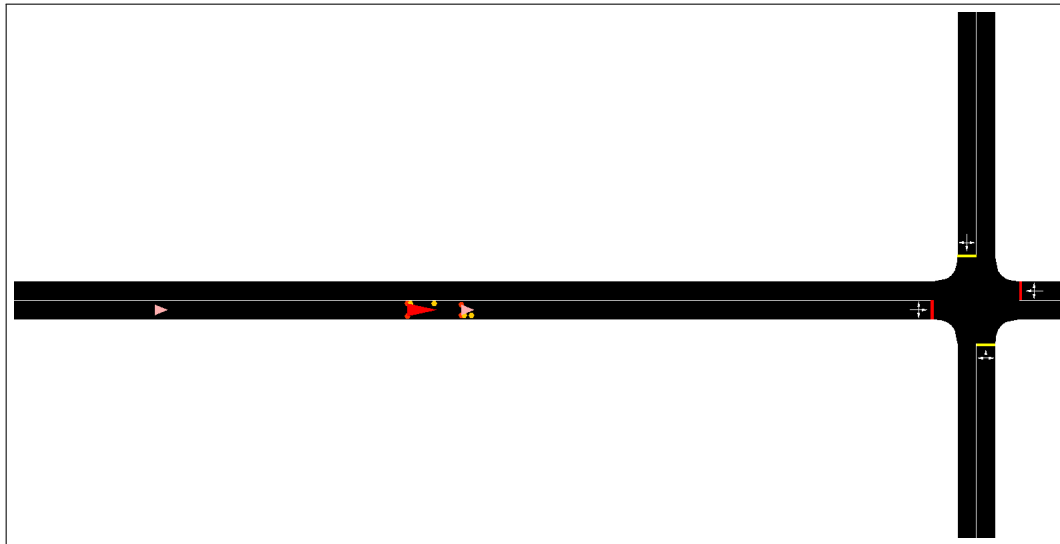


Fig. 7. Momento Inicial.

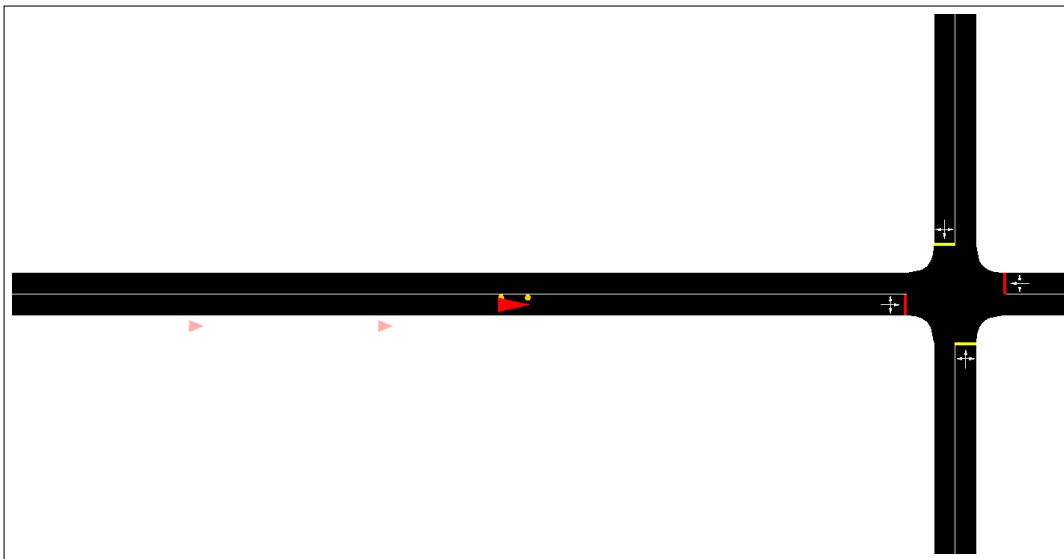


Fig. 8. Paragem dos veículos regulares.



Fig. 9. Retoma da circulação pelos veículos regulares.

VI. TRABALHO FUTURO

No desenvolvimento do nosso sistema, identificamos várias áreas que podem ser aprimoradas no futuro:

A. Implementação do Modelo de Encaminhamento Oportunista Baseado em Conhecimento

Embora tenhamos considerado a implementação dos algoritmos propostos em [1], devido a limitações de tempo, não conseguimos integrá-los ao nosso protocolo. A adoção destes algoritmos poderia melhorar a eficiência do encaminhamento de mensagens, ajustando dinamicamente as rotas com base em mais informação.

B. Avaliação da Atualidade das Mensagens

Atualmente, não realizamos uma avaliação para determinar quais mensagens são mais recentes, tanto no FOG quanto no veículo. A ausência dessa funcionalidade pode resultar em decisões baseadas em informações desatualizadas, comprometendo a eficácia do sistema. A implementação de um mecanismo para verificar que mensagens são mais atuais, como por exemplo descartar VehInfoMessage's de um veículo para o qual já temos uma mensagem com timestamp mais recente ou então tirar partido do messageId desta mesma, que incrementa à medida que novas mensagens são criadas e por isso é possível descartar mensagens com ids inferiores ao que já se recebeu no FOG, semelhante à implementação do numero de sequencia no protocolo Destination-Sequenced DV, este comportamento é essencial para garantir a precisão das decisões.

C. Refatoração do Código para Uniformizar o Tratamento de Mensagens

Refatorar o código para que o comportamento das mensagens no FOG e na rede seja tratado de forma uniforme. Isso facilitará a extensão de funcionalidades e a criação de uma framework para eventos. Embora já tenhamos implementado essa abordagem na RSU e no veículo, ela ainda não foi aplicada no FOG nem na OBU. Idealmente, criar uma nova mensagem deveria consistir apenas em implementar a classe abstrata *Message* e injetá-la na OBU ou no Fog.

D. Avaliação do Desempenho do Protocolo

Pretendemos aproveitar a Federação de Estatísticas e Output para realizar uma avaliação metódica do desempenho do protocolo, analisando métricas como latência, throughput e número de pacotes. Para isso, consideramos utilizar o cenário de BeST, conforme descrito em [2].

REFERENCES

- [1] J. LeBrun, C.-N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *2005 IEEE 61st Vehicular Technology Conference*, vol. 4, 2005, pp. 2289–2293 Vol. 4.
- [2] K. Schrab, R. Protzmann, and I. Radusch, "A large-scale traffic scenario of berlin for evaluating smart mobility applications," 03 2023, pp. 276–287.