
Segurança rodoviária suportada em computação de proximidade

Motivação

Uma rede veicular permite a comunicação dos veículos entre si, espontaneamente, ou com a infraestrutura, planeada e instalada ao longo das estradas. Atualmente, a maioria dos veículos já vem de fábrica equipado com dispositivos de comunicações e espera-se que num futuro muito próximo as redes veiculares sejam tão ubíquas quanto o são as redes móveis celulares atualmente. Dos vários tipos de aplicações que podem ser usadas nas redes veiculares, são as relacionadas com a segurança rodoviária as que mais expectativas geram, nomeadamente na diminuição de acidentes e do número de fatalidades associadas. Ao comunicarem entre si, os veículos aumentam a perceção que têm do ambiente à sua volta, podendo calcular situações de risco com base na informação coligida. De igual modo, existindo uma infraestrutura de rede fixa ao longo das vias, eventualmente com pequenas unidades computacionais próximas (minicentros de dados na periferia da rede), as aplicações podem tirar partido da computação de proximidade, de muito baixa latência, no fornecimento de serviços de segurança mais sofisticados e com melhor desempenho.

Objetivos

Neste trabalho, pretende-se contruir um protótipo funcional de um serviço de segurança rodoviária, suportado pela computação de proximidade. Os veículos enviam mensagens com dados sobre a sua mobilidade, que são recebidas não só pelos veículos seus vizinhos, como também pelas unidades de comunicação fixas instaladas na proximidade das vias de circulação. Unidades computacionais próximas podem criar uma visão da rede veicular da zona onde estão instaladas, mais duradoura, que possa ajudar a alertar os veículos para perigos vários. Uma vez recolhida e processada a informação, as unidades computacionais de proximidade podem enviar mensagens informativas e/ou de alerta de segurança de volta aos veículos, com o objetivo primário de melhorar a segurança rodoviária. A resposta do veículo a essas mensagens, pode ser passiva, de simples alerta do condutor, ou ativa, de atuação sobre o veículo, mas está fora do âmbito deste trabalho. O foco principal do trabalho está essencialmente na arquitetura global do serviço como um todo, com destaque para comunicações entre as várias entidades envolvidas.

Descrição geral

A Figura 1 mostra os componentes fundamentais da rede veicular a considerar. Os veículos (V_i) estão equipados com unidades de comunicação, designadas por OBU (*On-Board Unit*). Alguns, como o V_1 , V_2 , V_3 e V_4 , estão no alcance rádio da unidade de comunicação na berma da estrada, designada por RSU (*Road-Side Unit*). Outros, como o V_5 e V_6 estão fora de alcance da RSU e só comunicam com os vizinhos. V_6 só consegue trocar mensagens com o V_5 , enquanto o V_5 tem 3 veículos no seu raio de alcance: V_3 , V_4 e V_6 . Se for necessário enviar mensagens do V_6 para a RSU, é preciso que outros veículos no percurso (V_5 e V_3) reenviem as mensagens, ou seja, que atuem como routers.

Junto ao RSU, há uma pequena infraestrutura local, com um *switch*, um servidor local (*mini-cloud*) e um *router* para ligação ao resto da rede fixa. É no servidor que a computação mais complexa deve ocorrer. Nomeadamente a agregação de informação e geração de alertas de segurança rodoviária para os veículos que estão na sua zona de influência. A parte fixa da rede tem conectividade garantida. A parte móvel depende da mobilidade e densidade dos veículos e das RSUs. Os veículos podem executar aplicações específicas de

segurança rodoviária, usando protocolos próprios, ou aplicações genéricas TCP/IP. Tipicamente usam o protocolo IPv6, que é auto-configurável.

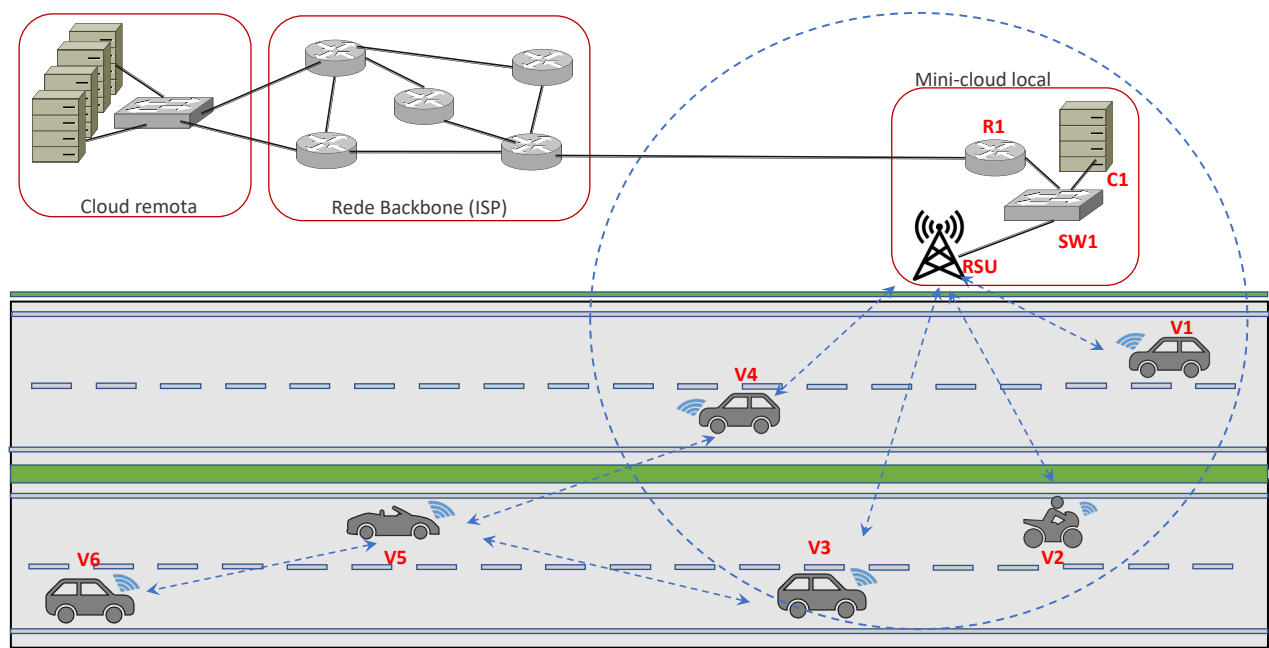


Figura 1: Visão geral da rede veicular e seus componentes principais

Podemos considerar que os veículos também possuem capacidade computacional, além dos dispositivos de comunicação. Podem por exemplo obter dados dos seus múltiplos sensores internos. Sabem a sua posição recorrendo ao serviço GNSS, conhecem a sua velocidade pelos dados do motor, etc. Embora os veículos possam executar internamente o seu módulo de segurança, com base na informação que conseguem obter dos vizinhos, existem vantagens em fazer a agregação de dados e computação de proximidade no servidor local instalado junto da RSU, uma vez que a latência neste caso é muito baixa. O mesmo não se poderia dizer se o serviço estivesse alojado numa *cloud* remota, acessível pela rede de *backbone* de um fornecedor de serviço, como ilustrado na figura. Este é o cenário de partida para o desenho e implementação do serviço proposto.

Requisitos mínimos

- cada veículo deverá estar sempre pronto a receber mensagens de informação e de alerta
- cada veículo deverá coligir informações a respeito de si próprio (tipo de veículo, dimensões, peso, etc), do seu movimento (posição, velocidade, aceleração, direção, etc), do estado dos seus sensores (piso molhado, nevoeiro, etc)
- cada veículo deverá executar uma app que envie periodicamente ou quando se justificar mensagens sobre o seu estado
- cada RSU na berma da estrada deve estar sempre pronta a receber mensagens de informação e de alerta dos veículos no seu raio de alcance
- cada RSU na berma da estrada deve poder enviar periodicamente informações agregadas ou alertas de segurança quando tal se justificar
- as mensagens a enviar resultam do processamento realizado numa unidade de computação de proximidade, diretamente ligada ao RSU na berma da estrada
- a área de intervenção das unidades de comunicação e processamento da berma da estrada pode ser alargada com recurso a encaminhamento adequado na rede veicular, a mais que um salto

Cada grupo deverá estender os requisitos de forma que considerar adequada ao serviço que vai implementar.

Descrição detalhada

Nesta secção propõe-se um conjunto de 3 etapas, com objetivos bem claros em cada etapa, bem como uma sugestão de atividades que podem ser desenvolvidas para atingir esses objetivos.

Etapas 1 – Protótipo do serviço limitado ao raio de alcance

Nesta etapa, espera-se poder testar uma versão simplificada do serviço. Os veículos limitam-se a adquirir os dados e a enviar por *multicast* a um salto de distância. É preciso especificar a informação a enviar, o formato e a periodicidade adequados. Os veículos podem também receber esses mesmos dados dos seus vizinhos. Os dados que a RSU conseguir receber no seu raio de ação, devem ser coligidos no servidor. Para evitar problemas de sincronização na comunicação entre a RSU e o servidor, sugere-se o uso de um *broker* intermédio, para publicação e subscrição de informação. A RSU publica dados dos veículos e consome informação do servidor. Por seu lado o servidor consome informação dos veículos e publica informação agregada. Nesta primeira fase o trabalho do servidor é apenas agregar os dados e sintetizá-los em informação útil simples, como número de veículos na área, densidade de tráfego, tempo de espera e velocidade recomendada, ou outras que sejam consideradas úteis para o serviço base. De igual modo é preciso especificar esses dados, formato e periodicidade.

É também necessário preparar o ambiente de teste e desenvolvimento, instalando as ferramentas necessárias e criando a topologia. Pode-se começar com uma topologia simples como a ilustrada na Figura 1. Com nós móveis, os veículos que se movimentam na via, e nós fixos, que constituem a infraestrutura da berma da estrada e de rede alargada. Na construção da topologia, dois aspetos devem ser tidos em conta:

- endereçamento e encaminhamento – recomenda-se o uso de IPv6 na gama de endereços *unicast* 2001:0690:2280:0820::0/60, e o endereço de grupo *multicast* FF02::1 para comunicação com todos os nós a um salto de distância
- mobilidade – definição de um modelo de mobilidade adequado para os veículos na via; pode-se recorrer numa primeira fase a modelos simples, como movimentação manual dos nós com o rato ou geração automática do movimento aleatório dos nós numa determinada área geográfica, mas o objetivo é ter um movimento realista dos veículos na estrada (são apresentados mais detalhes sobre como implementar a mobilidade mais à frente na secção relativa ao ambiente de desenvolvimento)

Em síntese, sugerem-se as seguintes atividades (a refinar pelos grupos de trabalho):

- criação da topologia híbrida de rede veicular ad-hoc e rede fixa de infraestrutura complementar
- teste de conectividade simples, para validação da topologia
- definição dos dados que o veículo deve enviar, modo de os obter e formato de envio
- criação de uma aplicação para os veículos que implemente uma estratégia de difusão dos dados a um salto de distância
- criação de uma aplicação para receção dos dados na RSU, que se limita a recebe-los e enviá-los para o servidor
- criação do serviço que calcula, por exemplo, algumas informações úteis de síntese, com base nos dados coligidos (ex: número médio de veículos, velocidade recomendada, etc);

No final desta primeira etapa deve ser possível que o servidor local processe os dados dos veículos a um salto de distância.

Etapas 2 – Serviço alargado em rede veicular tolerante a atrasos

Nesta segunda etapa, pretende-se introduzir um novo paradigma, combinando as redes veiculares com redes tolerantes a atrasos. O objetivo é alargar área de ação da RSU a mais que um salto de distância, lidando adicionalmente com as perdas temporárias de conectividade que podem ocorrer na rede veicular ad-hoc. A adaptação das redes tolerantes a atrasos (DTN) às redes veiculares, fornece um mecanismo simples para lidar com interrupções temporárias que podem ocorrer na comunicação. Os veículos podem ter de armazenar as mensagens a enviar, transportando-as algum tempo enquanto esperam por nova conexão, num paradigma designado por *store-carry-and-forward*. Para que a comunicação seja possível a mais que um salto de distância, é necessária uma estratégia de encaminhamento para escolher o próximo salto. Podem-se definir estratégias geográficas, baseadas na posição e no movimento dos veículos, ou topológicas, baseadas nos contactos entre veículos, ou outras. A estratégia deve ser pensada e definida por cada grupo de trabalho

Em síntese, sugerem-se as seguintes atividades (a refinar pelos grupos de trabalho):

- adicionar aos veículos capacidade de armazenamento temporário das mensagens
- especificar e implementar uma estratégia tolerante a atrasos adequada para envio dos dados a mais que um salto de distância, quer dos veículos até à RSU, quer no sentido contrário da RSU até aos veículos; a área máxima deve estar predefinida e não deve ser extensa, pois queremos manter a latência baixa; de igual modo o tempo máximo de atraso também deve ser controlado adequadamente;

Nesta etapa pode ser necessário aumentar a topologia e/ou mexer no modelo de mobilidade dos veículos para testar convenientemente a solução desenvolvida.

Etapas 3 – (Opcional) Serviço alargado em rede de veicular de dados nomeados

Nesta etapa, última e opcional, pretende-se combinar as redes veiculares com as redes de dados nomeados. O objetivo é que a recolha de informação dos veículos para o servidor seja feita com uma rede veicular de dados nomeados, onde os “endereços” perdem a relevância na comunicação, que passa a ser definida apenas por “nomes”. Os veículos são produtores de informação que a RSU vai consumir. O consumidor envia pacotes de interesses. Os produtores respondem com dados que fazem *match* com esses nomes. Adicionalmente os nós podem guardar os dados em cache (*Content Store*) o que significa que estas redes são também, por natureza, tolerantes a interrupções da comunicação. Para simplificar podemos manter a comunicação baseada em nomes apenas na parte móvel da rede, e comunicação baseada em endereços na parte fixa. Sendo assim, a RSU manda pacotes de interesses e recebe dados que depois envia para o servidor em TCP/IP.

Em síntese, sugerem-se as seguintes atividades (a refinar pelos grupos de trabalho):

- especificar os nomes a ser usados, quer nos interesses que a RSU vai enviar, que nos dados que os veículos vão produzir
- adicionar aos veículos os componentes necessários ou envio e receção de pacotes de interesses e de dados
- especificar e implementar uma estratégia de *forwarding* dos interesses e dos dados

Nesta etapa, espera-se que se mantenha a topologia e o cenário de teste da etapa anterior.

Ambiente de desenvolvimento e teste

A implementação e teste do serviço de segurança, como da rede veicular de suporte, protocolos de encaminhamento e aplicação propostos, deverá ser efetuada na plataforma de emulação CORE (*Common Open Research Emulator*). A mesma plataforma CORE deverá ser utilizada para a implementação dos nós móveis, incluindo, a utilização e programação do seu modelo de mobilidade. Como já foi referido, todos os nós da rede devem estar configurados em IPv6, e deverão utilizar *multicast* em vez de *broadcast* na descoberta dos vizinhos.

De acordo com a documentação (<http://coreemu.github.io/core/gui.html#wireless-networks>), o CORE permite vários níveis de emulação da rede sem fios: WLAN Node, Wireless Node e EMANE Node. Os WLAN são os mais simples e que usam menos recursos. Há comunicação se estiver no raio de alcance, não há comunicação se estiver fora. Este é um modelo muito rudimentar, embora aceitável para testar os serviços da camada de rede para cima, sem os detalhes das camadas física e lógica. Os Wireless Nodes modelam perdas de pacotes em função da distância, enquanto os nós EMANE usam um modelo de propagação adequado para o nível físico, consumindo, por isso, muito mais recursos.

Para a mobilidade, também existem três alternativas no CORE: script de mobilidade formato NS-2, aplicação externa a mover os nós usando uma API gRPC e finalmente usando scripts EMANE, no caso dos nós EMANE. Para gerar as scripts de mobilidade no formato ns-2 podem-se usar aplicações simples como o BonnMotion, ou mesmo simuladores de mobilidade como o SUMO, que exportam os eventos nesse formato. Uma vez geradas as scripts, ainda que manualmente, resta associar a script na janela de configuração da rede WLAN.

Avaliação

O trabalho deve ser realizado em grupo, com grupos de três elementos, sendo a constituição dos grupos da responsabilidade dos alunos. O trabalho deverá ser demonstrado em duas fases (*demonstração intermédia* a 29 de março, na semana antes da Páscoa, e *demonstração final* a 24 de maio, na última semana de aulas). Na demonstração final, além da demo, os alunos deverão elaborar um relatório escrito que descreva o trabalho efetuado. Este relatório e o respetivo código deverão ser submetidos até ao dia 22 de maio na plataforma *elearning.uminho.pt*.

O relatório deve ser escrito em formato de artigo com um máximo de 8 páginas (recomenda-se o uso do formato LNCS - *Lecture Notes in Computer Science*). Deve descrever o essencial do desenho e implementação com a seguinte estrutura recomendada: Introdução; Especificação do serviço e dos protocolos de suporte (primitivas de comunicação, formato das mensagens protocolares (PDU), interações); Implementação (detalhes, parâmetros, bibliotecas de funções, etc); Testes e resultados; Conclusões e trabalho futuro.

Referências

- “Common Open Research Emulator (CORE).” <https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/CORE/> (accessed Feb. 21, 2023).
- “CORE Documentation,” *core*. <http://coreemu.github.io/core/> (accessed Feb. 21, 2023).
- “Eclipse SUMO - Simulation of Urban MObility,” *Eclipse SUMO - Simulation of Urban MObility*. <https://www.eclipse.org/sumo/> (accessed Feb. 21, 2023).
- P. Alvarez Lopez *et al.*, “Microscopic Traffic Simulation using SUMO,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Maui, USA, Nov. 2018, pp. 2575–2582. Accessed: Feb. 21, 2023. [Online]. Available: <https://www.itsc2019.org/>
-

- Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 6–9. doi: [10.1109/PERCOMW.2017.7917508](https://doi.org/10.1109/PERCOMW.2017.7917508).
- S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A Survey on Vehicular Edge Computing: Architecture, Applications, Technical Issues, and Future Directions," *Wireless Communications and Mobile Computing*, vol. 2019, p. e3159762, Feb. 2019, doi: [10.1155/2019/3159762](https://doi.org/10.1155/2019/3159762).