

NDN4IVC: Um Arcabouço para Simulação e Experimentação de Aplicações em Redes Veiculares de Dados Nomeados

Guilherme B. Araujo^{1,2}, Maycon L. M. Peixoto¹, Leobino N. Sampaio¹

¹Programa de Pós-Graduação em Ciência da Computação (PGCOMP)
Departamento de Ciência da Computação – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

²Centro de Ciências Exatas – Universidade Federal do Recôncavo da Bahia (UFRB)

guilherme.araujo@ufrb.edu.br, {maycon.leone, leobino}@ufba.br

Abstract. *This paper presents a customized framework (NDN4IVC) for simulating and testing intelligent transportation systems and applications in vehicular named data networking (V-NDN). The project uses two popular simulators in the literature for VANET simulation, a network simulator based on discrete events (Ns-3), with ndnSIM module installed, and Sumo, a simulator for urban mobility. NDN4IVC allows bidirectional communication between Sumo and Ns-3 and integrates the NDN stack and the NFD (NDN Forwarding Daemon) code. The project also brings together a comprehensive set of codes, models, functionalities, and technologies to improve proposals and protocols in V-NDN.*

Resumo. *Este trabalho apresenta um arcabouço (NDN4IVC) personalizado para simular e testar sistemas de transporte inteligentes e aplicações em redes veiculares de dados nomeados (V-NDN). O projeto utiliza dois simuladores bem conhecidos na literatura para simulação de VANET, um simulador de rede baseado em eventos discretos (Ns-3), com módulo ndnSIM instalado, e o Sumo, um simulador de tráfego rodoviário e mobilidade. O NDN4IVC permite a comunicação bidirecional entre o Sumo e o Ns-3 e integra toda pilha NDN e NFD (NDN Forwarding Daemon). O projeto também agrega um conjunto abrangente de códigos, modelos, funcionalidades e tecnologias para facilitar o desenvolvimento de propostas e protocolos em V-NDN.*

1. Introdução

Rede de Dados Nomeados (do inglês, *Named Data Networking* – NDN) é uma arquitetura de rede centrada na informação projetada para suportar cache na camada de rede, entrega *multicast* e *multihoming* nativamente, segurança dos dados, apoio à mobilidade, controle de tráfego e de fluxo, dentre outros benefícios, através das estruturas PIT (*Pending Interest Table*), FIB (*Forwarding Information Base*) e CS (*Content Store*). Em redes veiculares (VANETs), devido a especificidade do ambiente, com alta mobilidade dos nós, constantes mudanças de topologia e desconexões frequentes, essas estruturas e serviços da NDN na camada de rede podem ajudar no encaminhamento eficiente de dados para aplicações distribuídas na rede. No caso de redes veiculares de dados nomeados (do inglês, *Vehicular Named Data Networking* – V-NDN), os veículos podem exercer os seguintes papéis: (i) consumidor de dados, (ii) produtor de conteúdo, (iii) encaminhador, quando conectado à infraestrutura ou a outros veículos, e (iv) “mula de dados”, quando

transporta dados mesmo sem conectividade de rede, espalhando conteúdo entre diferentes áreas [Grassi et al. 2014]. Ao contrário de outros tipos de dispositivos móveis, os veículos se movem sobre uma infraestrutura rodoviária conhecida e não se preocupam com capacidade de armazenamento [Rondon et al. 2020].

Há trabalhos na literatura que utilizam ferramentas para avaliar propostas em V-NDN que não são ambientes adequados e customizados para simulação de VANET, por não respeitar as especificidades do ambiente e mobilidade dos veículos na infraestrutura. O Veins¹, um arcabouço para simulação de VANET, utiliza o Sumo², como simulador de mobilidade urbana, e o OMNeTpp, como simulador de rede. A ferramenta implementa a arquitetura WAVE (*Wireless Access to Vehicular Environment*) para comunicação entre os nós na rede. Entretanto, realizar simulações no Veins de propostas em redes veiculares de dados nomeados não é uma tarefa trivial, pois exige que o usuário implemente toda estrutura de pacotes, armazenamento e encaminhamento de dados da arquitetura NDN. Por outro lado, o ndnSIM³, um módulo para o simulador de rede Ns-3⁴, traz a implementação de toda pilha NDN para o ambiente de simulação. Assim, o módulo permite trabalhar com pacotes no formato de nomes através da primitiva de códigos da biblioteca `ndn-cxx` (NDN C++ *library with eXperimental eXtensions*). Além disso, todo gerenciamento e processamento para encaminhamento de pacotes na rede é feito através do código-fonte do NFD (*Named data networking Forwarding Daemon*) [NDN Project 2021]. No entanto, o ndnSIM não pode ser classificado, conforme discutido em [Martinez et al. 2009], como simulador de VANET. O ambiente possui informações limitadas sobre a infraestrutura rodoviária e mobilidade dos nós nas respectivas vias e trajetos.

A carência de ambientes projetados para avaliação de aplicações em V-NDN pode inviabilizar a reprodução de cenários mais realistas ou gerar resultados de pesquisa inconsistentes. Ademais, a limitação dos ambientes existentes leva à proposição de soluções para VANETs que não exploram todas as propriedades da arquitetura NDN, as quais podem beneficiar o desenvolvimento de aplicações e resolver problemas intrínsecos do ambiente de mobilidade das VANETs. Portanto, diante da falta de um ambiente adequado para simulação de V-NDN [Khelifi et al. 2020], o principal objetivo deste trabalho é apresentar o arcabouço NDN4IVC (NDN *for Inter-Vehicular Communication*), para apoiar experimentações de aplicações em VANETs com suporte nativo da arquitetura NDN.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta o NDN4IVC. Os experimentos e resultados são discutidos na Seção 3. A Seção 4 traz o roteiro de demonstração e, por fim, a Seção 5 conclui o trabalho e lista trabalhos futuros.

2. Arcabouço Proposto

O NDN4IVC⁵ é um arcabouço para simulação e experimentação de sistemas de transporte inteligentes e aplicações em V-NDN. O ambiente utiliza dois simuladores bem conhecidos na literatura, o Ns-3, um simulador de rede baseado em eventos discretos, com módulo ndnSIM instalado, e o Sumo, um simulador de tráfego rodoviário e mobilidade urbana.

¹Disponível em: <http://veins.car2x.org>

²Disponível em: <https://www.eclipse.org/sumo>

³Disponível em: <https://ndnsim.net>

⁴Disponível em: <https://www.nsnam.org>

⁵Disponível em: <https://github.com/insert-br/ndn4ivc>

O projeto integra um conjunto abrangente de códigos, modelos e funcionalidades, para permitir simulação mais realista e apropriada para comunicação veicular no contexto de redes centradas na informação. Além da comunicação entre veículos V2V (*Vehicle-to-Vehicle*), a instalação e utilização de unidades de acostamento (do inglês, *Road Side Units* – RSUs) permite a comunicação dos veículos com infraestruturas computacionais (e.g., *fog, edge e cloud*), conhecida como V2I – *Vehicle-to-Infrastructure* [Peixoto et al. 2021].

A Figura 1 ilustra os principais componentes do arcabouço. A comunicação bidirecional entre Ns-3 e Sumo é parte central, *kernel* e engrenagem de todo sistema. Este projeto para integração do ambiente utiliza a interface TraCI [Wegener et al. 2008] e uma versão modificada do módulo disponível em [Vodafone Chair 2019].

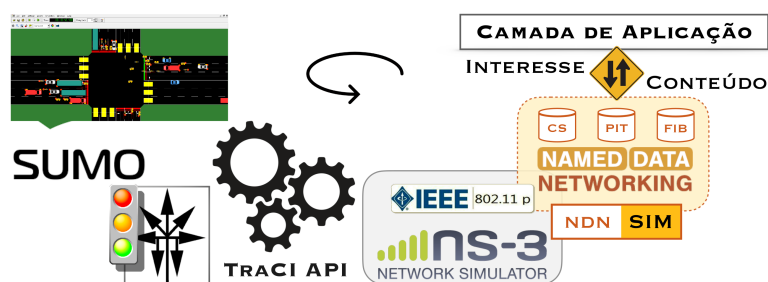


Figura 1. Visão em alto nível dos principais componentes do NDN4IVC

Sobre a integração dos componentes da Figura 1, é preciso destacar que o Sumo e o Ns-3 tem características de projeto distintas e cada simulador interage com os nós no ambiente de forma diferente. No Sumo os veículos são criados e removidos durante todo período de simulação, enquanto que o Ns-3 não suporta a inserção dinâmica de nós na rede. Por isso, para cada execução, o cenário deve ser descrito previamente e criado de forma estática, especificando todos os nós e parâmetros antes de iniciar a simulação. Em resumo, no Ns-3 não é possível adicionar e remover nós durante a simulação de forma nativa. Por outro lado, a simulação de redes veiculares exige a criação dinâmica de nós na rede, pois veículos entram no cenário, percorrem um caminho (rota) e saem do ambiente ao encerrarem a viagem. O módulo em [Vodafone Chair 2019], que também auxilia nesse processo, utiliza um conceito de *pool* de nós para Ns-3, com funções dinâmicas de inicialização e desligamento do nó no simulador. A ideia consiste em ativar e desativar os recursos de um dispositivo na rede, a partir da inserção ou remoção de nós pelo Sumo. Assim, um nó (veículo) no simulador de rede que tenha determinada aplicação desinstalada, seja movido para longe do raio de alcance dos demais veículos e esteja com a interface de rede desativada, não irá receber, processar, nem criar nenhum novo evento no ambiente que altere o comportamento da simulação. Dessa forma, é possível fazer com que todo ambiente consiga trabalhar com inserção e remoção dinâmica de nós durante a simulação, sem precisar fazer alterações profundas no código dos simuladores. Mudanças significativas no funcionamento do Ns-3 para atender objetivos específicos certamente impactarão vários módulos e códigos já validados pela comunidade. As informações sobre inserção, remoção e mobilidade dos veículos são trocadas através da interface TraCI.

A Figura 2 mostra parte do código-fonte utilizado pelo arcabouço para habilitar o mecanismo de inserção dinâmica de nós durante a simulação. As funções lambda, `setupNewSumoVehicle` e `shutdownSumoVehicle`, são responsáveis por ativar e desativar os nós no cenário de forma síncrona com o simulador de mobilidade.

```

251 */
252 std::function<Ptr<Node> (>) setupNewSumoVehicle = [&] () -> Ptr<Node> {
253     if (nodeCounter >= nodePool.GetN ())
254         NS_FATAL_ERROR ("Node Pool empty: "
255             << nodeCounter << " nodes created.");
256
257     NS_LOG_INFO ("Ns3SumoSetup: node [" << nodeCounter<< "] has i
258     Ptr<Node> includedNode = nodePool.Get (nodeCounter);
259     nodeCounter++;
260     Ptr<TmsConsumerApp> tmsConsumerApp = CreateObject<TmsConsumer
261     tmsConsumerApp->SetAttribute ("Frequency", UintegerValue (int
262     tmsConsumerApp->SetAttribute ("Client", (PointerValue) sumoC
263     includedNode->AddApplication (tmsConsumerApp);
264     return includedNode;
265 };
271 */
272 std::function<void (Ptr<Node>)> shutdownSumoVehicle = [&] (Ptr<
273     NS_LOG_INFO ("Ns3SumoSetup: node [" << exNode->GetId ()
274     << "] will be finished an
275     Ptr<TmsConsumerApp> tmsConsumerApp = DynamicCast<TmsConsumerA
276     // App will be removed
277     if (tmsConsumerApp)
278         tmsConsumerApp->StopApplication ();
279
280     for (uint32_t i = 0; i < exNode->GetNDevices (); ++i)
281         if (exNode->GetDevice (i)->GetObject<WifiNetDevice> () ||
282             exNode->GetDevice (i)->GetObject<WifiNetDevice> ()->GetPh
283
284     nodesDisable2Move.emplace (exNode->GetId (), (ns3::Time) ns3:
285 };

```

Figura 2. Processo de ativação e desativação dinâmica de nós no NDN4IVC

A estrutura de diretórios do projeto é apresentada no Quadro 1. A pasta `doc` deve ser utilizada para documentação futura ou específica do código. Em `examples` estão disponíveis cenários de simulação implementados no ambiente. Os diretórios `helper` e `model` contêm boa parte do código do arcabouço para simulação apropriada de redes veiculares de dados nomeados. As estratégias de encaminhamento customizadas para V-NDN, as aplicações de exemplo e arquivos de ajuda, para configuração da camada de enlace e física através do IEEE 802.11p, estão disponíveis nesses dois diretórios.

Em `Traces` estão armazenados cenários de simulação para o Sumo. Os usuários têm a opção de modificar os arquivos para adequar os cenários rodoviários as necessidades de simulação. Por isso, é importante observar que a criação de novos mapas deve seguir o modelo presente no arcabouço, para que seja possível carregar automaticamente os novos cenários no Ns-3, sem a necessidade do usuário ter que fazer ajustes no código. Um arquivo `Makefile` está presente nas subpastas para facilitar o processo de geração sintética de mobilidade nos mapas rodoviários disponíveis na ferramenta. A pasta `results` deve ser utilizada para armazenar arquivos de logs de experimentos, tratamento de dados e apresentação gráfica de resultados.

Quadro 1: Estrutura de diretórios proposta para o arcabouço

Diretório	Descrição
<code>doc/</code>	Documentação adicional
<code>examples/</code>	Cenários de simulação para o Ns-3
<code>helper/</code>	Códigos auxiliares de ajustes e configuração
<code>model/</code>	Aplicações e códigos do usuário
<code>traces/</code>	Dados de mobilidade e mapas rodoviários para o Sumo
<code>results/</code>	Arquivos de log, tratamento de dados e gráficos

2.1. Casos de Uso Propostos

Dois casos de uso em V-NDN foram definidos para demonstrar possibilidades de simulação e testes do NDN4IVC. No primeiro, utiliza-se apenas a comunicação V2V para troca periódica de mensagens *beacons* [Khelifi et al. 2020]. Nesse caso o objetivo é demonstrar possibilidades de troca de informações entre veículos vizinhos para otimizar o fluxo de veículos de urgência, na faixa mais a esquerda das vias. O segundo caso de uso implementa um sistema de gerenciamento de tráfego para troca de informações, através da comunicação V2X (*Vehicle-to-everything*), sobre vias congestionadas, e permite mudanças de rotas de veículos a fim de reduzir o tempo de viagem. Os códigos-fontes dos casos de uso podem ser visualizados no repositório do projeto nos diretórios `model` e `examples`.

É importante enfatizar que os cenários propostos têm objetivos didáticos para demonstração do arcabouço e ambiente customizado para simulação de V-NDN. Otimizações, inserção de novos casos de uso e contribuições no código podem ser feitas através do repositório oficial do NDN4IVC no Github.

Caso de Uso I: segurança de tráfego – comunicação V2V.

A aplicação (i) de segurança rodoviária consiste em enviar periodicamente *beacons* de interesse para propagar informações sobre a localização do veículo aos demais nós vizinhos. Consequentemente, cada nó na rede também consegue manter uma tabela de vizinhança atualizada. Lembrando que *beacons* de interesse são mensagens de controle, em que não há a necessidade de nenhum pacote de dados de retorno.

O esquema de nomeação proposto para aplicação é `/localhop/beacon/<node-id>/<road-id>/<pos-x>/<pos-y>/<pos-z>/<speed>`. Pode-se verificar que o escopo de encaminhamento de mensagens de interesse é *localhop* [NDN Project 2016]. Isso significa que o próprio NFD irá controlar e limitar a propagação dessas mensagens em apenas um salto. Assim, não há necessidade da aplicação (i) ter que tratar e evitar problemas como *broadcast storm* na rede, pois a própria arquitetura NDN oferece isso como um serviço para camada de aplicação. O esquema de nomeação também especifica um rótulo para mensagem, e, cada veículo, ao transmitir informações sobre si, acrescenta ao nome seu id, tipo do veículo, localização e velocidade atual.

Veículos rodando a aplicação (i) processam mensagens (*beacons*) recebidas de outros vizinhos. Se a mensagem for de veículos de emergência, o veículo de passageiro deverá mudar para faixa da direita e dar passagem rapidamente. Nesse caso de uso foi utilizado um cenário rodoviário de autoestrada, disponível no diretório `traces/highway`.

Caso de Uso II: eficiência de tráfego veicular – comunicação V2X.

A aplicação (ii) de gerenciamento de tráfego disponível no arcabouço para testes explora a comunicação V2X. Nesse caso, veículos trocam mensagens e estimativas acerca do nível de congestionamento (velocidade média e taxa de ocupação) nas vias de interesse. As RSUs no cenário são responsáveis por prover informações sobre o tráfego. Os veículos (consumidores) enviam mensagens de interesse sobre as condições do tráfego nas vias do percurso a ser realizado. Se houver congestionamento em alguma via acima de um limiar os veículos podem alterar a rota original, com a finalidade de buscar vias alternativas para viagens mais rápidas. Nesse caso de uso os veículos também participam do processo de retransmissão de mensagens na rede, exercendo os papéis de consumidor de dados, encaminhador e “mula de dados”, espalhando conteúdo na rede. Para essa aplicação foi utilizado um cenário de grade simples, disponível no diretório `traces/grid-map`.

Para a aplicação do caso de uso (ii) o esquema de nomeação proposto para troca de informações é `/service/traffic/<road-id>/<time-window>`. O esquema discrimina o nome do serviço, rua (via) de interesse e janela temporal que representa as condições do tráfego no intervalo. Para ajudar na troca de dados nesse cenário o arcabouço integra e utiliza a biblioteca `JSON Nlohmann C++`⁶. Trabalhar com formato JSON na camada de aplicação se justifica por ser um formato compacto, simples, intuitivo e de fácil descrição para troca de dados entre sistemas e aplicações distribuídas na rede.

⁶Disponível em: <https://github.com/nlohmann/json>

O NDN4IVC promoveu adaptações na estratégia de encaminhamento *multicast*, disponível no módulo ndnSIM, para rodar aplicações em V-NDN de forma adequada. Por padrão a estratégia *multicast* envia mensagens NACK de retorno quando nenhuma rota é alcançável para determinado nome. Além disso, devido a questões de projeto e limitações do ambiente de simulação, o módulo ndnSIM, ao enviar mensagens de interesse na rede utiliza uma função para tratamento de eventos que irá processar apenas uma das seguintes condições: (i) estouro de tempo, (ii) recebimento de NACK ou (iii) dados. Contudo, essa técnica não é adequada para V-NDN e, além de inundar a rede com pacotes NACK (*No Route*), devido as constantes mudanças na topologia da rede, pode impedir o recebimento de mensagens de dados entre nós vizinhos, caso um NACK seja processado primeiro. Para resolver essa questão o arcabouço inclui a estratégia de encaminhamento *multicast-vanet*. Outros ajustes também foram feitos para criação de *faces* NDN em modo *ad-hoc*, mais apropriadas para o IEEE 802.11p e contexto de redes veiculares de dados nomeados.

3. Avaliação da Ferramenta e Caso de Uso II

Para avaliação do trabalho foi feito o planejamento dos experimentos através do projeto fatorial completo [Jain 1991], com os seguintes fatores e respectivos níveis: número de RSUs (1, 2), tamanho da cache (1, 1000), densidade de veículos (1000, 2000) e estratégia de encaminhamento (*multicast*, *multicast-vanet*). A densidade de veículos representa a quantidade de veículos por quilômetro quadrado e as rotas foram geradas de forma aleatória, para cada replicação do experimento. Um evento de acidente foi gerado e os experimentos usam IEEE 802.11p com raio de comunicação de 70 metros. Os resultados mais relevantes serão discutidos a seguir e os gráficos têm intervalo de confiança de 95%.

A Figura 3 mostra um aumento acentuado no número de pacotes enviados na rede para a estratégia de encaminhamento *multicast*, a medida que aumenta a densidade de veículos na rede. Esse mesmo comportamento não é observado para estratégia *multicast-vanet*. O aumento acentuado é explicado pela quantidade excessiva de mensagens NACK de retorno, informando que determinado nome não é alcançável por determinado nó.

Para estratégia *multicast-vanet* é perceptível, Figura 3, que o aumento no número de mensagens transmitidas é baixo ao dobrar a quantidade de veículos. Esse comportamento é atribuído ao plano de encaminhamento ciente de estado (do inglês, *stateful forwarding plane*) intrínseco da arquitetura NDN, em que a cache na rede e a tabela PIT (*Pending Interest Table*) ajudam em um processo de encaminhamento mais otimizado e orientado pelo consumidor, evitando redundância de requisição e dados na rede.

A Figura 4 destaca os resultados para outra métrica avaliada, tempo médio de viagem. Para cenários sem proposta os dados foram obtidos diretamente pelo simulador Sumo, e, estão presentes para fins de comparação (*baseline*) e melhor avaliação dos resultados. É perceptível que nos cenários sem proposta o tempo médio de viagem dos veículos é maior. A estratégia de encaminhamento *multicast* apesar de melhorar o tempo médio de viagem, comparado com cenários sem proposta, apresenta um afastamento grande nos dados, principalmente em experimentos com maior densidade de veículos.

Os cenários com a estratégia *multicast-vanet* possuem um tempo médio de viagem melhor comparado com os demais cenários. Além disso, os dados obtidos possuem uma baixa dispersão, indicando que houve uma melhora significativa para todos os veículos. Esse comportamento se mantém, mesmo com aumento da densidade de veículos na rede.

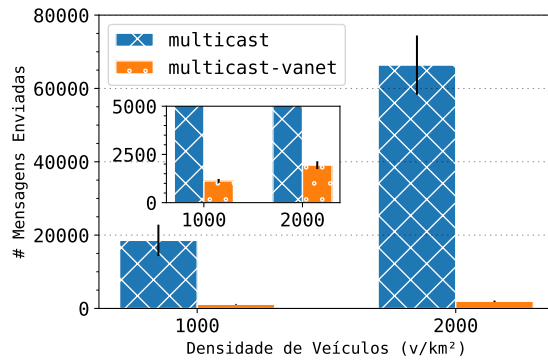


Figura 3. Número de pacotes enviados por densidade de veículos e estratégia de encaminhamento

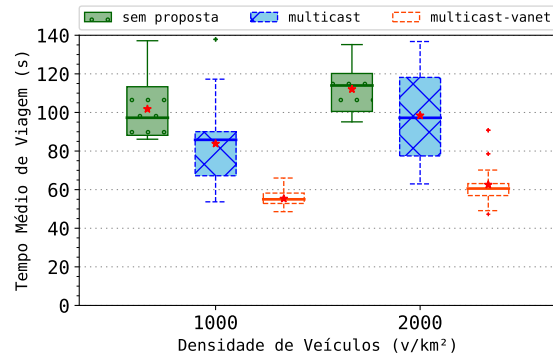


Figura 4. Tempo médio de viagem por densidade de veículos e estratégia de encaminhamento

O tempo médio de viagem é uma métrica estipulada para aplicação (ii), contudo, é perceptível, por refletir nos resultados, a importância em utilizar um ambiente customizado e adequado para simulação de redes veiculares de dados nomeados. Nesse caso, para a mesma aplicação é possível obter resultados bem diferentes se o ambiente não for validado previamente e otimizado para contexto da proposta.

4. Roteiro de Demonstração

O Sumo versão 1.1.0 e Ns-3 versão 3.30.1, com módulo ndnSIM versão 2.8, foram utilizados pelo NDN4IVC na versão 1. Os testes e instalação de todo ambiente foram feitos no Ubuntu 18.04 LTS. Todo código-fonte, roteiro para instalação e configuração de parâmetros de simulação estão disponíveis no repositório oficial da ferramenta no Github.

A Figura 5 mostra a aplicação (ii) executando no ambiente proposto. A Figura 5 (a) exibe a interface gráfica do Sumo com a mobilidade dos veículos na rede rodoviária. A Figura 5 (b) ilustra a comunicação entre os respectivos nós no Ns-3. É importante ressaltar que o módulo gráfico no Ns-3, desenvolvido em Python, por padrão exibe o eixo y de forma invertida, quando comparado com a interface gráfica do Sumo. Contudo, é apenas uma questão visual, pois as coordenadas geográficas no plano são idênticas em ambos os simuladores. Para melhor demonstração da ferramenta e interação com o usuário, os veículos ao processarem informações sobre o tráfego e selecionarem uma rota alternativa, trocam, em tempo real, a cor padrão para azul. No vídeo⁷ demo fica mais intuitivo.

5. Conclusões e Trabalhos Futuros

O NDN4IVC customiza todo ambiente e permite simulações mais eficientes e adequadas de propostas em redes veiculares de dados nomeados (V-NDN), através da integração de um simulador de rede e um simulador de mobilidade urbana. Dois casos de uso propostos exploram o potencial da ferramenta, permitindo discutir e apresentar na prática algumas possibilidades. Os resultados obtidos de simulação e a discussão apresentada endereçam questões e problemas apresentados no trabalho, demonstrando a capacidade e potencialidade do arcabouço. Como trabalhos futuros pretende-se adicionar novos cenários e casos de uso que usem outras características e serviços da arquitetura NDN, como, por exemplo, segurança dos dados. Outras estratégias de encaminhamento disponíveis na literatura e otimizadas para redes veiculares de dados nomeados [de Sousa et al. 2018] também serão implementadas em versões futuras do arcabouço.

⁷Disponível em: <https://www.youtube.com/channel/UCzjOH9dSMYA5aoR-GZkAotw>

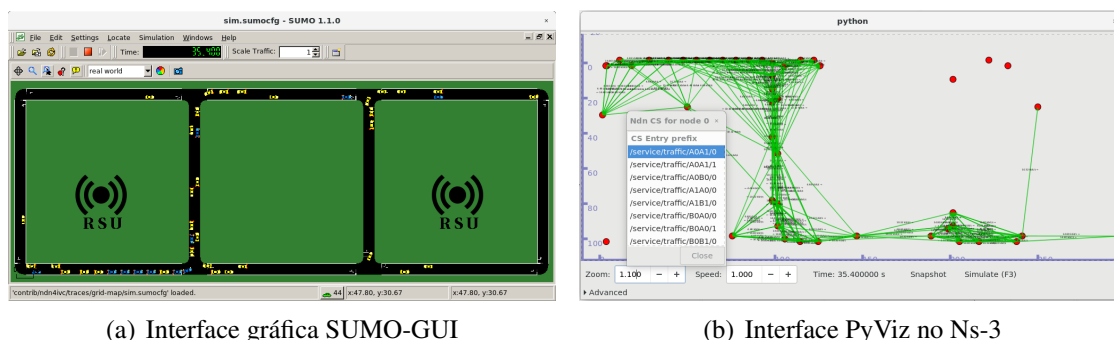


Figura 5. Interfaces gráficas de visualização de simulação ao vivo do NDN4IVC

Agradecimentos

Os autores agradecem o apoio parcial da CAPES, FAPESB e CNPq.

Referências

- de Sousa, A. M., Araújo, F. R., and Sampaio, L. N. (2018). A link-stability-based interest-forwarding strategy for vehicular named data networks. *IEEE Internet Computing*, 22(3):16–26.
- Grassi, G., Pesavento, D., Pau, G., Vuyyuru, R., Wakikawa, R., and Zhang, L. (2014). Vanet via named data networking. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 410–415.
- Jain, R. (1991). *The art of computer systems analysis - techniques for experimental design, measurement, simulation, and modeling*. Wiley professional computing. Wiley.
- Khelifi, H., Luo, S., Nour, B., Mounsla, H., Faheem, Y., Hussain, R., and Ksentini, A. (2020). Named data networking in vehicular ad hoc networks: State-of-the-art and challenges. *IEEE Communications Surveys Tutorials*, 22(1):320–351.
- Martinez, F. J., Toh, C. K., Cano, J.-C., Calafate, C. T., and Manzoni, P. (2009). A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). *Wireless Communications and Mobile Computing*, 11(7):813–828.
- NDN Project (2016). Namespace-based scope control. <https://redmine.named-data.net/projects/nfd/wiki/ScopeControl>. Acesso 15/05/21.
- NDN Project (2021). Named data networking official repositories. <https://github.com/named-data>. Acesso: 10/05/21.
- Peixoto, M., Maia, A., Mota, E., Rangel, E., Costa, D., Turgut, D., and Villas, L. (2021). A traffic data clustering framework based on fog computing for vanets. *Vehicular Communications*, 31:100370.
- Rondon, L. B., Immich, R., Filho, G. P. R., Venâncio Neto, A., Leone Maciel Peixoto, M., and Villas, L. A. (2020). Towards improved vehicular information-centric networks by efficient caching discovery. *Vehicles*, 2(3):453–467.
- Vodafone Chair (2019). A ns3 module for bidirectional coupling with sumo. <https://github.com/vodafone-chair/ns3-sumocoupling>. Acesso: 15/05/21.
- Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., and Hubaux, J.-P. (2008). Traci: An interface for coupling road traffic and network simulators. In *11th Communications and Networking Simulation Symposium*, page 155–163.