ADS IFBA

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

15 ANOS

Caian de Jesus Santana

Comunicação e Orquestração de Semáforos Inteligentes via Redes de Dados Nomeados (NDN)

Salvador, BA

2025

Caian de Jesus Santana

Comunicação e Orquestração de Semáforos Inteligentes via Redes de Dados Nomeados (NDN)

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal da Bahia, como requisito parcial para obtenção do grau de Tecnólogo.

Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA)
Campus Salvador

Prof. Dr. Allan Edgard Silva Freitas

Salvador, BA
2025

Aos meus pais, Silvanir e Carlos.

Resumo

A crescente complexidade da mobilidade urbana exige soluções inovadoras para a gestão de tráfego, e os sistemas de semáforos inteligentes representam uma frente promissora. Contudo, as arquiteturas de rede tradicionais, como a TCP/IP, enfrentam desafios de escalabilidade, segurança e mobilidade em ambientes de Internet das Coisas (IoT). Este trabalho propõe e avalia o uso da arquitetura de Rede de Dados Nomeados (*Named Data Networking* - NDN) como uma alternativa para a comunicação em um sistema de controle de tráfego. Para tal, foi projetado e implementado um protótipo funcional composto por um nó orquestrador central e múltiplos nós semáforos, desenvolvido em C++ com o auxílio das bibliotecas *ndn-cxx* e *yaml-cpp*. O sistema implementa lógicas de otimização, como gestão de cruzamentos, ondas verdes e grupos de sincronia, além de mecanismos de tolerância a falhas. A avaliação do protótipo, realizada em cenários configuráveis e implantada em ambientes local e conteinerizado com *Docker*, demonstrou a viabilidade da arquitetura, apresentando baixa latência de comunicação e robustez. Os resultados sugerem que a NDN é uma arquitetura promissora para sistemas de transporte inteligentes, oferecendo flexibilidade através de sua comunicação centrada em dados e validando seu uso não apenas para nós móveis, mas também para uma infraestrutura de nós estacionários com poder de decisão macro.

Palavras-chaves: Redes de Computadores. Sistemas Distribuidos. NDN. IoT. Semáforos Inteligentes. Semáforos de Tempo Real.

Abstract

The growing complexity of urban mobility demands innovative solutions for traffic management, and smart traffic light systems represent a promising approach. However, traditional network architectures, such as TCP/IP, face challenges in scalability, security, and mobility within Internet of Things (IoT) environments. This work proposes and evaluates the use of the Named Data Networking (NDN) architecture as an alternative for communication in a traffic control system. For this purpose, a functional prototype was designed and implemented, composed of a central orchestrator node and multiple smart traffic light nodes, developed in C++ using the *ndn-cxx* and *yaml-cpp* libraries. The system implements optimization logic, such as intersection management, green waves, and synchronization groups, as well as fault tolerance mechanisms. The prototype evaluation, conducted in configurable scenarios and deployed in both local and containerized *Docker* environments, demonstrated the architecture's feasibility, showcasing low communication latency and robustness. The results suggest that NDN is a promising architecture for intelligent transportation systems, offering flexibility through its data-centric communication and validating its use not only for mobile nodes but also for a stationary node infrastructure with macro-level decision-making capabilities.

Keywords: Computer Networks. Distributed Systems. NDN. IoT. Smart Traffic Lights. Real-Time Traffic Lights.

Sumário

Lista de ilustrações

1 Introdução

1.1 Contextualização

A mobilidade urbana tem se tornado um dos principais desafios enfrentados pelas grandes cidades, impactando diretamente a qualidade de vida da população e a eficiência econômica dos centros urbanos. Através de pesquisas realizadas em 2007 e 2012 constatou-se o expressivo aumento da escolha pelo transporte individual (BERNAL; FERREIRA, 2016), isso tornou os congestionamentos cada vez mais frequentes, resultando em perdas significativas de tempo e oportunidades (CINTRA, 2008); aumento da poluição (FONSECA et al., 2022); e elevação dos níveis de estresse entre motoristas e pedestres (FENERICH, 2016).

Com o avanço das *Tecnologias da Informação e Comunicação* (TIC), diversas iniciativas têm buscado aplicar esses recursos ao cotidiano dos cidadãos, com o objetivo de mitigar problemas urbanos recorrentes. Nesse contexto, surgem as chamadas Cidades Inteligentes, ambientes urbanos que integram tecnologia para otimizar serviços e tomar decisões de forma autônoma e em tempo real. Consequentemente, o foco hoje são projetos que visam tornar a economia, a mobilidade urbana, o meio ambiente, os cidadãos e o governo mais inteligentes. A cidade passa a ser um organismo informacional que reage e atualiza todos sobre suas condições a qualquer hora (LEMOS, 2013).

No setor de mobilidade, os sistemas de controle de semáforo também vêm sendo beneficiados com os avanços da tecnologia (GOMES, 2014).

> Os Semáforos Inteligentes, ou Semáforos de Tempo Real, se destacam como uma aplicação relevante. O princípio de funcionamento é o mesmo dos semáforos atuados, ou seja, os veículos são detectados por sensores e a variação dos tempos procura acompanhar a evolução do tráfego. (NETO, 2016).

A diferença está na comunicação, adaptabilidade e poder de decisão. Os semáforos Atuados apenas detectam e atuam, não são capazes de transformar uma via através do aprendizado e compartilhamento de informações com outros semáforos.

1.2 Motivação

Como visto anteriormente, a integração entre tecnologias inovadoras e conceitos já consolidados oferece novas perspectivas e soluções em direção ao progresso tecnológico. Uma das propostas recorrentes nos últimos anos é a chamada Internet do Futuro. Segundo Moreira et al. (2014):

> É uma nova Internet que deve manter os princípios que levaram ao sucesso atual, tais como a facilidade para implantação de novas aplicações e a adaptabilidade de seus protocolos, mas deverá possuir conceitos novos, tais como autocura e autogerenciamento, podendo absorver alguns princípios de inteligência e conhecimento.

Essa nova abordagem busca superar limitações fundamentais da arquitetura atual, sobretudo frente às crescentes demandas por mobilidade, segurança, escalabilidade, *Internet of Things* (IoT) e aplicações em tempo real. Diante desse cenário, torna-se necessário estudar as possíveis aplicações dessas propostas em diferentes áreas e setores, permitindo que se avance, de forma prática, rumo a uma nova geração de redes.

Um exemplo promissor no contexto da Internet do Futuro é a arquitetura conhecida como *Named Data Networking* (NDN). Ao contrário da Internet atual, centrada em endereços IP e na comunicação entre dispositivos, o NDN adota uma abordagem centrada em dados nomeados, permitindo que os usuários solicitem diretamente o conteúdo desejado, independentemente de sua localização física (JACOBSON et al., 2010) . Essa característica torna o NDN particularmente adequado para contextos em que os dados são mais relevantes do que a identidade dos dispositivos emissores, como ocorre em sistemas IoT, nos quais sensores e atuadores geram e consomem informações de forma contínua.

Nesse sentido, a aplicação da arquitetura NDN em dispositivos IoT, como os semáforos inteligentes, revela-se uma oportunidade valiosa de estudo e desenvolvimento no meio acadêmico, contribuindo para soluções mais eficientes e inteligentes no controle do tráfego urbano.

1.3 Justificativa

Comparada com a arquitetura tradicional TCP/IP, a NDN apresenta peculiaridades promissoras, especialmente no contexto de dispositivos IoT. A natureza heterogênea e restrita desses dispositivos acarreta desafios que não são enfrentados em computadores tradicionais cabeados, como destacam Shang et al. (2016).

Na camada de rede, por exemplo, a baixa capacidade energética da maioria dos dispositivos IoT exige que seus enlaces de comunicação possuam baixo custo energético, o que frequentemente resulta em pequenas Unidades Máximas de Transmissão (MTUs). Essa limitação se torna particularmente problemática no IPv6, cujo cabeçalho possui tamanho fixo de 40 bytes, além de possíveis cabeçalhos de extensão opcionais. Em cenários que demandam pacotes pequenos, esses cabeçalhos tornam-se um grande obstáculo. Para lidar com essa limitação, o protocolo *6LoWPAN* (acrônimo de *IPv6 over Low-power Wireless Personal Area Networks*) introduz uma camada adaptativa entre as camadas de enlace e rede, visando reduzir a sobrecarga e tornar o IPv6 viável em redes de baixo consumo energético.

Na camada de transporte, diversos fatores dificultam o uso do TCP em ambientes IoT. O uso frequente de modos de baixo consumo (*sleep mode*) torna inviável a manutenção de conexões duradouras. Além disso, a maioria das comunicações entre dispositivos IoT envolve pequenas quantidades de dados, o que torna o estabelecimento de conexão via TCP um exagero em termos de custo e complexidade. Por fim, aplicações que exigem baixa latência são prejudicadas pelo atraso introduzido pelo *handshake* do TCP. Como consequência, muitos protocolos IoT optam por integrar funcionalidades típicas da camada de transporte diretamente na camada de aplicação, adotando o UDP como protocolo de transporte principal, o que transforma a camada de transporte em um simples mecanismo de multiplexação e encaminhamento de dados.

Na camada de aplicação, a arquitetura REST (*Representational State Transfer*) tem se mostrado bem-sucedida no contexto da Web, o que levou a comunidade IoT a

adaptá-la para suas aplicações. No entanto, a necessidade de reimplementar funcionalidades essenciais como descoberta, *caching* e segurança na camada de aplicação evidencia as deficiências da arquitetura TCP/IP em oferecer suporte nativo a essas demandas.

"NDN não apenas oferece suporte nativo às funcionalidades essenciais das aplicações IoT, como também enfrenta os desafios das camadas inferiores da pilha TCP/IP" (tradução nossa, (SHANG et al., 2016)).

1.4 Objetivos

1.4.1 Objetivo Geral

Diante do cenário apresentado, que evidencia os desafios da mobilidade urbana e o potencial de novas arquiteturas de rede como a NDN para aplicações de IoT, este trabalho tem como objetivo principal avaliar a viabilidade da comunicação baseada em NDN em um sistema de semáforos inteligentes.

1.4.2 Objetivos Específicos

- Projetar uma arquitetura para um sistema de controle de tráfego centralizado, onde semáforos inteligentes comunicam-se com uma central de controle utilizando os princípios da Rede de Dados Nomeados (NDN).
- Desenvolver um protótipo funcional em C++ utilizando a biblioteca ndn-cxx e o encaminhador NFD. O protótipo deverá contemplar:
  - Uma aplicação para o Semáforo Inteligente, atuando como produtor de dados de fluxo veicular e consumidor de comandos de temporização.
  - Uma aplicação para a Central de Controle, responsável por consumir os dados dos semáforos, processá-los e produzir os comandos de controle.
- Validar o fluxo de comunicação ponta a ponta, demonstrando que os pacotes Interest e Data são trocados corretamente entre os componentes do sistema.
- Coletar e analisar métricas de latência da comunicação para avaliar o desempenho do protótipo em um cenário de troca de mensagens em tempo real.

1.5 Estrutura do Trabalho

Este trabalho está organizado da seguinte forma: O Capítulo 2 apresenta o referencial teórico que fundamenta a pesquisa, abordando os conceitos de Redes de Dados Nomeados (NDN), Internet das Coisas (IoT), Sistemas Inteligentes de Transporte e sistemas distribuídos síncronos. O Capítulo 3 analisa os trabalhos relacionados. O Capítulo 4 detalha a visão arquitetural do sistema proposto. O Capítulo 5 descreve o projeto de implementação do protótipo. O Capítulo 6 apresenta a Prova de Conceito, onde são demonstrados os resultados práticos da implementação e a análise de desempenho da rede. Finalmente, o Capítulo 7 consolida a conclusão do trabalho e aponta sugestões para pesquisas futuras.

2 Referencial Teórico

2.1 Named Data Networking

A arquitetura *Named Data Networking* (NDN) emerge do modelo *Information-Centric Networking* (ICN), que desloca o foco da localização dos *hosts* para o próprio conteúdo. Essa abordagem fundamental contrasta com as redes tradicionais baseadas em TCP/IP, que priorizam a comunicação entre endereços específicos. Conforme Sampaio et al. (2021), essa desassociação entre localizador e identificador de conteúdo na NDN acarreta mudanças significativas em diversos aspectos da rede, incluindo a identificação de *hosts*, encaminhamento, segurança e roteamento.

Na NDN, a preocupação central reside na distribuição e obtenção dos dados requeridos, não na comunicação direta entre dispositivos. A rede opera em um modelo de *publish/subscribe* por demanda: quem deseja um dado expressa esse interesse na rede, e quem o possui responde à solicitação.

2.1.1 Nomes

Os nomes são o cerne da arquitetura Named Data Networking (NDN), funcionando como a identificação fundamental do conteúdo solicitado. Na NDN, os nomes são construídos de forma hierárquica, onde o caractere "/"demarca as divisões entre os componentes de cada nível. Por exemplo, em "/brasil/bahia/salvador/ifba", "/brasil"representa o topo da hierarquia, enquanto "/ifba"indica o nível mais específico. Essa estrutura é análoga à do *Domain Name System* (DNS), onde o ponto (.) separa os níveis hierárquicos, como em . (domínio de topo), seguido por .br (domínio de segundo nível), e assim sucessivamente.

É importante notar que os nomes na NDN não precisam ser globalmente únicos. Em contextos de comunicação local, há flexibilidade para adotar nomes variados, adaptados ao cenário específico. Contudo, para que os dados possam ser disponibilizados e adquiridos globalmente, um certo grau de singularidade nos nomes torna-se necessário para garantir a descoberta e recuperação corretas do conteúdo.

2.1.2 Pacotes

Os pacotes *Interest* são o principal meio de comunicação entre os *hosts*. Eles podem conter diversos parâmetros destinados a otimizar e garantir a condição da solicitação, como prevenção de *loops* e controle do tempo de vida na rede. O objetivo primário de um pacote *Interest* é solicitar um conteúdo específico a qualquer produtor, eliminando a necessidade de conhecer sua localização exata. As respostas a um *Interest* podem ser de três tipos: *Data*, que é o conteúdo solicitado; *Negative Acknowledgment* (NACK), indicando que um *host* não pode satisfazer o pedido ou encaminhá-lo; ou *Timeout*, resultante da ausência de resposta.

Por sua vez, os pacotes *Data* são enviados de volta ao emissor do *Interest*, percorrendo o caminho inverso ao que o pacote *Interest* trilhou. O pacote *Data* carrega consigo as informações de encaminhamento que o pacote *Interest* adquiriu em cada salto entre os roteadores.

2.1.3 Estruturas

A NDN possui três estruturas fundamentais para o processamento de pacotes nos roteadores:

- *Pending Interest Table* (PIT): Registra as entradas com os nomes dos interesses pendentes e o conjunto de interfaces de onde os interesses correspondentes foram recebidos.
- *Forwarding Information Base* (FIB): Utilizada pelo roteador para consultar a próxima interface para alcançar o nome desejado pelo Interest. A FIB é populada por um protocolo de roteamento baseado em nomes.
- *Content Store* (CS): Um *cache* local onde o roteador armazena dados que já foram enviados anteriormente.

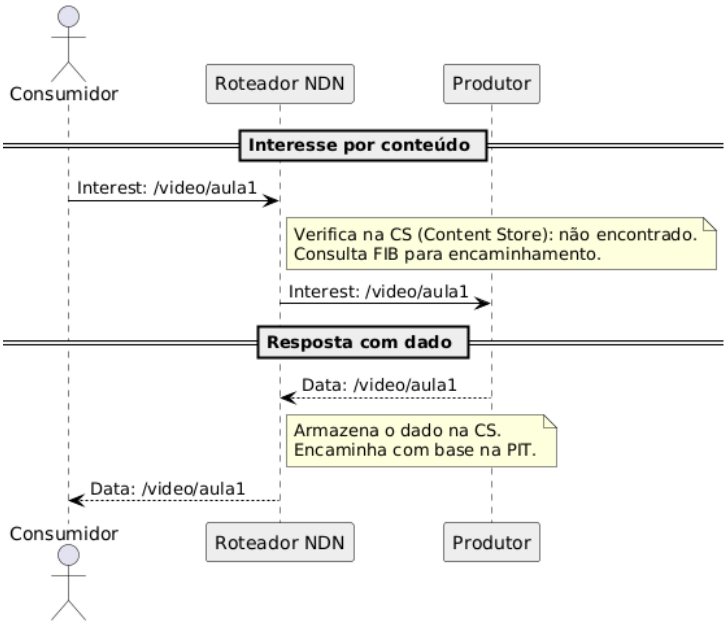

Figura 1 – Funcionamento básico de uma rede NDN.

Conforme a figura 1, quando um roteador NDN recebe um pacote Interest, ele o encaminha para a fonte de dados e o registra na PIT. Ao receber o pacote *Data* correspondente, o roteador encontra a entrada na PIT, encaminha o pacote para todas as interfaces listadas, remove a entrada da PIT e armazena o *Data* no CS.

2.1.4 Funções

Existem três papéis fundamentais na arquitetura NDN, que não são mutuamente exclusivos e podem ser assumidos simultaneamente pelos *hosts*:

- O Produtor: Escuta por pacotes *Interest* em um determinado nome e responde com dados criptografados e assinados digitalmente.
- O Consumidor: Envia pacotes *Interest* para um nome específico e, ao receber os dados, os descriptografa e valida utilizando parâmetros de segurança configurados previamente.

- O Encaminhador: Atua como intermediário, facilitando a troca de pacotes entre produtores e consumidores.

2.1.5 Características inerentes

Uma das características mais poderosas e intrínsecas da arquitetura NDN é sua abordagem de segurança, que é aplicada diretamente no dado:

> "Na NDN, a segurança é feita no próprio dado, em vez de utilizar uma função de onde, ou como, ele foi obtido. Cada pedaço de dado é assinado junto ao seu nome, acoplando-os de maneira segura. Assinaturas de dados são mandatórias. A assinatura, acompanhada com informação do produtor dela, permite que o consumidor confie nos dados, desassociando como (e de onde) o dado foi obtido."(tradução nossa, (JACOBSON et al., 2010)).

Essa abordagem de segurança no próprio conteúdo é um exemplo claro de como a NDN permite que diversas demandas sejam atendidas sem a necessidade de infraestruturas ou protocolos adicionais. Isso inclui a distribuição eficiente de conteúdo para múltiplos usuários requisitando a mesma informação, suporte nativo a *multicast*, facilitação da mobilidade, aumento da robustez de segurança e maior tolerância a redes com alta latência (*delay*).

2.2 Internet of Things

O conceito de *Internet of Things* (IoT) é bastante amplo e não pode ser definido como apenas uma tecnologia, assim como Redes de Computadores ou Sistemas Distribuídos também não podem. A IoT descreve um conjunto de tecnologias que conectam dispositivos entre si, criando uma comunicação efetiva com efeitos que podem ser meramente lógicos ou até físicos. Assim, a IoT pode ser vista como uma rede composta por dispositivos diversos — desde objetos domésticos até aplicações industriais.

> "A Internet das Coisas tem como princípio básico a conectividade, no começo somente entre os objetos do mundo real a um sistema ou ser humano; atualmente e no futuro, haverá objetos com inteligência suficiente para interagir com o meio em que estão inseridos sem necessidade de interação de um ser humano diretamente."(FACHINI et al., 2017)

2.2.1 Estrutura básica

De acordo com Santos et al. (2016), a IoT pode ser descrita como a combinação de diversas tecnologias, as quais se complementam com o objetivo de integrar os objetos do ambiente físico ao mundo virtual. A seguir, uma descrição das três principais tecnologias:

- Sensores: São dispositivos responsáveis por perceber e captar informações do ambiente ao redor do dispositivo. Existem diversos tipos de sensores, cada um com um princípio de funcionamento distinto — por exemplo, sensores de movimento baseados em infravermelho ou micro-ondas, sensores de pressão, de temperatura, entre outros. Independentemente do tipo, os dados coletados por esses sensores são transmitidos

para posterior processamento, podendo ser analisados e/ou utilizados para acionar atuadores.

- Atuadores: Responsáveis por executar ações no mundo físico com base em comandos recebidos, geralmente originados da interpretação de dados coletados por sensores. Eles podem variar de elementos simples, como um LED que acende ao detectar presença, até dispositivos mais complexos, como um relé que aciona uma fechadura eletrônica após o pressionar de um botão.
- Controladores: São sistemas computacionais eletrônicos completamente integrados a um dispositivo ou equipamento, projetados para realizar tarefas específicas de forma dedicada. Podem ser classificados como microcontroladores, com recursos limitados e altamente especializados, ou como computadores de placa única (*Single Board Computers* – SBCs), que possuem maior capacidade de processamento e flexibilidade.

2.2.2 Comunicação

Santos et al. (2016) discorre sobre diversas tecnologias de comunicação, realizando um comparativo entre elas por meio de uma tabela. Entre as tecnologias abordadas, destacam-se:

- Ethernet: Padrão presente na maioria das redes locais com fio atualmente. Entretanto, não é adequada para grande parte dos dispositivos IoT devido à sua limitação em mobilidade.
- Wi-Fi: Tecnologia sem fio amplamente utilizada em residências e escritórios. Apesar disso, não apresenta a melhor compatibilidade com dispositivos IoT, principalmente por não priorizar o consumo energético reduzido.

A comunicação com a camada de aplicação na IoT pode ser realizada de diferentes maneiras. Um dos protocolos utilizados é o *Constrained Application Protocol* (CoAP), mantido pelo *IETF Constrained RESTful Environments (CoRE) Working Group*. O CoAP oferece funcionalidades semelhantes às do HTTP, mas utiliza o protocolo UDP. Ele é dividido em duas camadas: uma que implementa mecanismos de requisição/resposta e outra que assegura a confiabilidade da comunicação sobre o UDP.

Outra alternativa é o *Message Queue Telemetry Transport* (MQTT), um protocolo baseado na arquitetura *publish/subscribe*, projetado para minimizar o uso de recursos computacionais e de largura de banda. O MQTT possui três componentes principais: *publisher*, *subscriber* e *broker*. O funcionamento se dá da seguinte forma: um dispositivo *subscriber* inscreve-se em um tópico de um *broker*, passando a receber todas as mensagens publicadas — os dados expostos — por um dispositivo *publisher*.

2.2.3 Funcionamento

Como descrito anteriormente, o fluxo de operação de um sistema IoT ocorre em etapas:

1. Um dado é capturado por meio de um sensor;

2. Este dado é processado localmente no dispositivo ou transmitido via rede para processamento e/ou armazenamento remoto;

3. Caso tenha sido transmitido, o dispositivo pode receber uma resposta com o dado já tratado;

4. Com base nesse dado processado, um atuador pode ser ativado para realizar uma ação no ambiente físico.

2.3 Sistemas Inteligentes de Transporte

De acordo com Jesus (2001), os Sistemas Inteligentes de Transporte (do inglês, *Intelligent Transportation Systems* – ITS) são compostos por usuários, vias e veículos cada vez mais integrados e tecnologicamente avançados, com o objetivo de aumentar a eficiência dos transportes, promovendo maior conforto, segurança e preservação ambiental.

Quando aplicadas à gestão de sistemas – como o transporte público e as vias urbanas – e ao projeto de veículos, as tecnologias ITS podem contribuir significativamente para a redução do consumo de combustível. Isso ocorre porque esses sistemas possibilitam um planejamento mais eficiente de rotas e horários, além da adaptação dinâmica da sinalização semafórica, reduzindo acelerações e frenagens bruscas, o que promove um fluxo de tráfego mais contínuo, segundo Shaheen e Finson (2013).

Além disso, os ITS viabilizam a coleta automática de pedágios, facilitam a implementação de estratégias de precificação e gestão da demanda, e tornam o transporte público mais atrativo. Também ampliam as oportunidades de mobilidade compartilhada, oferecem informações e opções de pagamento para viagens intermodais, e permitem ajustes nos sistemas de transmissão dos veículos conforme as condições da via e do terreno. Outro benefício relevante é o suporte à tomada de decisão em tempo real, com base nas condições atuais do trânsito. Por fim, os ITS também otimizam o controle de acesso às rodovias por meio do gerenciamento inteligente de rampas. Em conjunto, essas inovações têm contribuído de forma concreta para a redução do consumo de energia no setor de transportes.

2.3.1 Semáforos Inteligentes

Naturalmente, ao longo do tempo, os semáforos passaram a integrar os ITS, evoluindo para dispositivos cada vez mais automatizados e inteligentes. Conforme destacado por Neto (2016), os primeiros semáforos monoplanos e multiplanos eram programados manualmente por controladores. Os modelos monoplanos eram configurados para atender ao horário de maior fluxo, o que resultava em ineficiências nos demais períodos. Já os multiplanos possuíam diferentes programações de acordo com os horários e até os dias da semana.

Além disso, os semáforos podem ser classificados em dois paradigmas operacionais: isolados e em rede. Os isolados operam de forma independente, seguindo seu próprio plano de sinalização. Já os semáforos em rede são utilizados quando é necessária a coordenação entre diferentes cruzamentos, garantindo que os tempos de abertura e fechamento sejam sincronizados. Para isso, todos os controladores devem seguir uma mesma referência de tempo e operar com um tempo de ciclo comum. Essa sincronização pode ser obtida por

meio de cabos dedicados, ajuste de relógio via GPS ou sincronização por computador central.

Outro ponto de distinção está na escolha entre semáforos de tempo fixo e semáforos por demanda. Os primeiros podem ser mono ou multiplanos, exigindo grande volume de dados sobre a via e cálculos detalhados para diferentes períodos do dia e da semana. Apesar de eficazes, esses sistemas não são totalmente capazes de lidar com as variações aleatórias do tráfego cotidiano, ainda que tentem antecipá-las com base em modelos como o de Webster. Como alternativa, os semáforos por demanda oferecem uma resposta mais adaptativa às condições reais do tráfego.

Dentro dessa categoria, destacam-se dois tipos: os semáforos atuados e os de tempo real. Segundo Neto (2016), os semáforos atuados não utilizam programação horária fixa. Em vez disso, sensores detectam o fluxo de veículos e transmitem essas informações ao controlador local, que ajusta os tempos dos sinais com base em parâmetros previamente definidos – como tempos máximos e mínimos de ciclo e extensão do verde. Como o tempo de ciclo não é fixo, não é possível implementar com eficiência estratégias como a onda verde, limitando sua aplicação a cruzamentos isolados.

Os semáforos em tempo real apresentam um funcionamento semelhante aos atuados, pois também dependem da detecção veicular para ajustar os tempos de sinalização. Contudo, diferenciam-se por sua arquitetura centralizada. Nessa abordagem, os dados coletados pelos sensores são enviados a um computador central, que realiza os cálculos e define as alterações a serem aplicadas. Para isso, é necessário o uso de uma Rede de Transmissão de Dados (RTD), que interliga sensores, controladores e o sistema central.

O fluxo de operação pode ser descrito da seguinte forma:

- O veículo é detectado ao passar por uma zona de detecção;
- A informação é enviada ao controlador local;
- O controlador transmite os dados ao computador central por meio da RTD;
- O computador central processa as informações e envia uma decisão de manter ou alterar o estado do sinal;
- O controlador executa a decisão recebida, ajustando a cor do semáforo conforme indicado.

Esse modelo permite uma gestão mais eficiente da rede viária como um todo, com respostas mais rápidas e alinhadas ao comportamento real do tráfego urbano.

2.4 Sistemas Distribuídos Síncronos

Em um sistema distribuído síncrono, os componentes individuais, ou processos, operam sob premissas temporais estritas e conhecidas. A principal característica que define este modelo é a existência de limites superiores finitos e conhecidos para a execução de tarefas e para a entrega de mensagens.

> Os sistemas distribuídos síncronos podem ser construídos, desde que se garanta que os processos sejam executados de forma a respeitar as restrições

> temporais impostas. Para isso, é preciso alocar os recursos necessários, como tempo de processamento e capacidade de rede, e limitar o desvio do relógio. (COULOURIS et al., 2013)

Especificamente, assume-se que o tempo necessário para que um processo execute um passo de um algoritmo está limitado por um valor máximo conhecido. Da mesma forma, o atraso na comunicação — ou seja, o tempo que uma mensagem leva para transitar do processo remetente ao destinatário através da rede — também está sujeito a um limite superior conhecido. Essa previsibilidade temporal é a pedra angular do modelo síncrono, distinguindo-o fundamentalmente de outras abordagens em computação distribuída.

As garantias temporais do modelo síncrono têm implicações profundas no projeto e na análise de algoritmos distribuídos. A consequência mais significativa é a capacidade de detectar falhas de processo de forma confiável. Se um processo não envia uma mensagem esperada dentro do intervalo de tempo máximo estipulado (considerando o tempo de processamento e de transmissão), os demais processos podem concluir inequivocamente que ele falhou. Esse mecanismo, geralmente implementado por meio de timeouts, simplifica significativamente a complexidade de problemas fundamentais em sistemas distribuídos, como o consenso e a exclusão mútua. A previsibilidade do sistema permite que os algoritmos operem em rodadas sincronizadas, nas quais cada processo executa uma série de passos, comunica seus resultados e aguarda as mensagens dos outros — tudo dentro de um ciclo de tempo bem definido.

2.4.1 Algoritmo de Cristian

O algoritmo de Cristian é um algoritmo centralizado que consistem em nós clientes que solicitam informações sobre o tempo a um nó com um relógio mais preciso, geralmente um servidor. Para calcular com maior precisão, o algoritmo leva em conta o *Round-Trip Time*(RTT) da mensagem, isto é, a média do tempo de ida da requisição e da volta da resposta, como mostra a Figura 2.

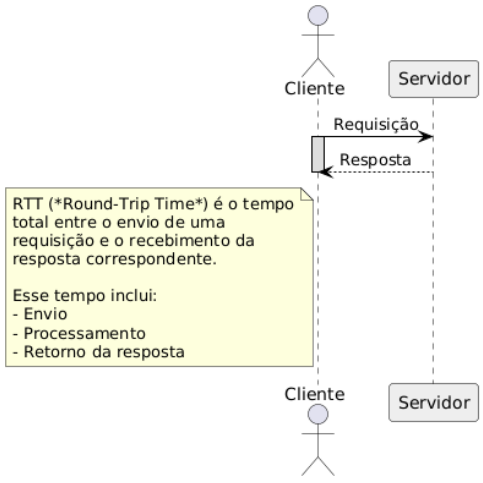

Figura 2 – Diagrama da arquitetura do sistema.

Parafraseando Clemens e A. (2021), temos a seguinte definição:

> *Round-Trip Time* é o conceito mais importante do algoritmo de Cristian. Em suma, o RTT é o tempo decorrido entre o início de uma requisição e o

recebimento da respectiva resposta. A sincronização de relógio utilizada para alinhar o tempo do processo cliente com o tempo do servidor é chamada de Algoritmo de Cristian. Em termos de precisão, redes de baixa latência lidam bem com esse algoritmo, pois o RTT tende a ser pequeno. (Tradução nossa).

3 Trabalhos Relacionados

3.1 Smart traffic lights over vehicular named data networking

O trabalho de Al-qutwani e Wang (2019) explora o potencial da NDN ao propor um sistema de Semáforos Virtuais (VTLs), em que a infraestrutura física é substituída por uma comunicação direta entre veículos e uma Unidade de Beira de Estrada (RSU). O foco da proposta reside em um controle de tráfego dinâmico, baseado na autonomia e no estado individual dos veículos.

Em contraste, o presente trabalho, embora também utilize a NDN como arquitetura de comunicação subjacente, adota um enfoque na otimização da infraestrutura semafórica já existente. O objetivo é demonstrar a viabilidade de um sistema de controle centralizado, no qual os semáforos, equipados com sensores, utilizam a NDN para enviar dados de fluxo veicular a uma central. Esta central processa as informações e retorna comandos de temporização, mantendo o semáforo como principal atuador no controle do cruzamento.

Adicionalmente, esta proposta aborda a lacuna de segurança para pedestres e ciclistas identificada por Al-qutwani e Wang (2019). Enquanto o sistema de VTLs elimina os sinais visuais, tornando o ambiente imprevisível para pedestres, ciclistas e veículos não autônomos, a manutenção dos semáforos físicos neste trabalho garante que a segurança desses usuários não seja comprometida. A previsibilidade dos sinais luminosos é preservada, e a implementação de tempos mínimos e máximos de ciclo assegura que os pedestres sejam atendidos, oferecendo uma solução mais pragmática e segura para ambientes urbanos mistos.

3.2 Controlador de tráfego: semáforo inteligente

Com um enfoque complementar, o trabalho de Araújo (2006) concentra-se no desenvolvimento do hardware e do software embarcado de um controlador de tráfego adaptativo. A proposta visa ajustar os tempos de verde de uma interseção de forma proporcional à demanda, utilizando sensores e lógica de controle local para minimizar o atraso veicular. A solução é eficaz para a otimização de um cruzamento isolado.

Contudo, o próprio autor aponta como uma evolução natural do projeto a implementação da comunicação entre múltiplos semáforos, de forma a criar uma rede sincronizada — como uma “onda verde” —, ponto não abordado em seu escopo. É precisamente nesta lacuna que a presente pesquisa se fundamenta. Enquanto o trabalho supracitado resolve o desafio do controle local, este trabalho propõe uma arquitetura de rede baseada em NDN, que serve como sistema de comunicação para conectar múltiplos semáforos inteligentes. Dessa forma, nossa proposta pode ser vista como a camada de rede que habilita a inteligência coordenada sugerida por Araújo (2006), permitindo que a otimização do tráfego ocorra não apenas localmente, mas em toda a malha viária.

3.3 Semáforo inteligente com comunicação via protocolo ESP-NOW

Explorando uma abordagem de comunicação de baixo nível, o trabalho de Kolls e Simplicio (2022) foca na utilização do protocolo *ESP-NOW* para a comunicação entre semáforos inteligentes. A solução aproveita as vantagens de simplicidade e rapidez do protocolo em microcontroladores ESP para criar uma rede local eficiente, estabelecendo comunicação entre nós a grandes distâncias, mesmo através de obstáculos, além de utilizar a criptografia CCMP. No entanto, essa escolha implica uma forte dependência de uma plataforma de hardware específica, criando um sistema fechado e com barreiras à interoperabilidade. Um semáforo baseado em ESP-NOW, por exemplo, não poderia se comunicar nativamente com um veículo autônomo ou com a infraestrutura de outro fornecedor.

Em contrapartida, a arquitetura NDN proposta neste trabalho resolve essa limitação fundamental por ser, por design, independente de hardware. Ao focar na nomeação de conteúdos em vez de endereços de dispositivos, a NDN permite que um ecossistema de trânsito heterogêneo se comunique de forma padronizada. Isso não só viabiliza a comunicação entre diferentes modelos de semáforos, como também abre portas para uma integração segura e nativa com veículos conectados e outras aplicações de cidades inteligentes, garantindo uma solução mais escalável, flexível e preparada para o futuro.

4 Visão Arquitetural

A arquitetura do sistema, representada na Figura 3, adota um modelo de controle centralizado baseado em comunicação orientada a dados. O sistema é implementado em C++ e composto por duas entidades principais: um nó central denominado **Orchestrator** e múltiplos nós de borda, os **Semáforos Inteligentes** (`SmartTrafficLights`).

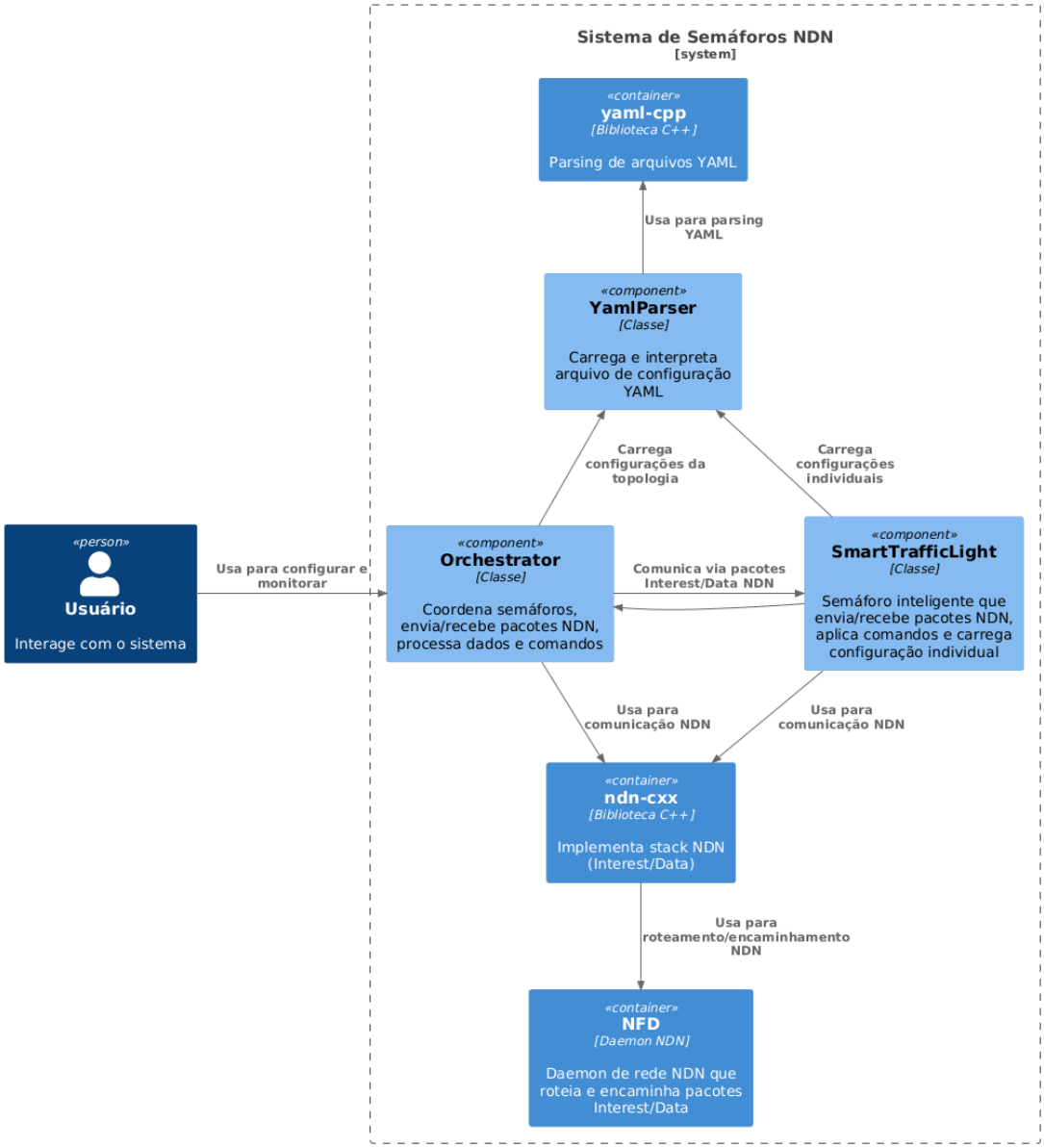

Figura 3 – Diagrama de componentes C4 do Sistema de Semáforos NDN.

A comunicação sobre a rede *Named Data Networking (NDN)* é realizada por meio da biblioteca `ndn-cxx`, que fornece as primitivas para a criação e manipulação de pacotes *Interest* e *Data*. O roteamento e encaminhamento desses pacotes na camada de rede são gerenciados pelo *daemon* `NFD (Named Data Networking Forwarding Daemon)`.

A configuração inicial do cenário, incluindo a topologia da rede e as relações lógicas entre os nós, é carregada a partir de um arquivo no formato `.yaml`. A análise sintática

(*parsing*) deste arquivo é realizada pela biblioteca `yaml-cpp`, que interpreta a topologia geral para o **Orchestrator** e as configurações individuais para cada `SmartTrafficLight`. Ambas as entidades operam simultaneamente como produtoras e consumidoras de dados, engajando-se em um ciclo de comunicação bidirecional.

Para fins de monitoramento, depuração (*debugging*) e análise de comportamento, o sistema foi dotado de um subsistema de *logs* com níveis de verbosidade configuráveis. O usuário pode selecionar, no momento da inicialização, o nível de detalhe desejado, que varia entre `ERROR` (registrando apenas falhas críticas), `INFO` (informações gerais sobre o estado e decisões importantes), e `DEBUG` (informações detalhadas sobre o fluxo de execução). Um nível `NONE` também está disponível para desativar completamente a geração de *logs*.

A seguir, descrevem-se os principais aspectos arquiteturais do sistema, abordando a estrutura dos componentes, o modelo de comunicação baseado em NDN, os algoritmos de controle e sincronização de tráfego, os mecanismos de tolerância a falhas e a modelagem interna de prioridade.

4.1 Modelo de Comunicação NDN

O fluxo de comunicação é projetado para manter o **Orchestrator** com uma visão atualizada do estado da rede e permitir que os semáforos atuem sob seu comando. O processo ocorre em dois fluxos paralelos:

- **Coleta de estado** (**Orchestrator** → `SmartTrafficLight`): o **Orchestrator** expressa pacotes *Interest* periódicos para cada semáforo inteligente, solicitando seu estado operacional. Em resposta, cada semáforo produz um pacote *Data* contendo uma carga útil estruturada com as seguintes informações: o estado atual (por exemplo, `"VERDE"`, `"VERMELHO"`), o tempo remanescente para a próxima transição de estado (em milissegundos) e um valor de prioridade dinâmica calculado localmente.
- **Disseminação de comandos** (`SmartTrafficLight` → **Orchestrator**): de forma recíproca, cada semáforo expressa pacotes *Interest* para o **Orchestrator**, solicitando comandos de atuação. O **Orchestrator** responde com um pacote *Data* contendo uma `string` de comando, que foi previamente processada e armazenada com base nas informações de estado coletadas. Ao receber o comando, o semáforo o interpreta e ajusta seu comportamento.

Para garantir a precisão temporal, o sistema emprega uma abordagem inspirada no *Algoritmo de Cristian* para sincronização de relógios. Ao receber um pacote *Data*, o **Orchestrator** realiza uma correção no tempo de término do estado informado, subtraindo metade do *Round-Trip Time (RTT)* — medido entre o envio do *Interest* e o recebimento do *Data* — para compensar a latência da rede. Esse mecanismo permite que o **Orchestrator** mantenha uma visão mais precisa e sincronizada do estado real de toda a rede semafórica.

4.2 Lógica de Controle e Otimização de Tráfego

O **Orchestrator** implementa múltiplos algoritmos para otimizar o fluxo de tráfego de forma coordenada:

- **Gestão de cruzamentos:** para cruzamentos compostos por dois semáforos, o sistema implementa uma lógica de coordenação onde o tempo de transição do semáforo em estado `VERMELHO` é calculado com base no tempo remanescente do semáforo em estado `VERDE`, incluindo a consideração pelo tempo `AMARELO`. Essa lógica é implementada na função `generateIntersectionCommand()`.
- **Ondas Verdes** (*Green Waves*): o sistema suporta a criação de ondas verdes, nas quais um semáforo *líder* dita o ritmo de abertura, e semáforos *seguidores* ajustam seus ciclos com um atraso (*offset*) proporcional ao tempo de viagem entre eles. Caso um seguidor esteja dessincronizado — por exemplo, com muito tempo restante em vermelho ou envolvido em um cruzamento — o **Orchestrator** antecipa gradualmente seu ciclo. Essa lógica é implementada em `processGreenWaves()`.
- **Grupos de sincronia:** para cenários como travessias de pedestres em vias de mão dupla, o sistema permite definir grupos em que todos os semáforos espelham o estado e o tempo de um semáforo *líder*, garantindo sincronização absoluta. Implementado em `processSyncGroups()`.
- **Ajuste por prioridade dinâmica:** o **Orchestrator** calcula a prioridade média da rede e ajusta, para cada semáforo, a duração de seus tempos de `VERDE` e `VERMELHO`, conforme sua prioridade local em relação à média. Implementado em `assignPriorityCommands()`.

4.3 Mecanismos de Tolerância a Falhas

A robustez do sistema é garantida por meio de múltiplos mecanismos de detecção e mitigação de falhas:

- **Detecção de falhas de comunicação:** o **Orchestrator** identifica a falha de um nó ao não receber respostas a seus *Interests*, registrando múltiplos eventos de *timeout* ou pacotes *NACK (Negative Acknowledgment)*.
- **Gestão de cruzamentos comprometidos:** ao detectar a falha de um semáforo em um cruzamento, o **Orchestrator** marca o cruzamento como comprometido e envia comandos de **alerta** para os demais nós envolvidos, que então entram em um estado seguro (por exemplo, `AMARELO` piscante). Os próprios semáforos também entram nesse estado caso detectem perda de comunicação com o **Orchestrator**.
- **Recuperação de falhas:** quando um nó restabelece a comunicação, o **Orchestrator** comanda todos os semáforos do cruzamento para o estado `VERMELHO` e aplica a lógica de resolução de *deadlock* para reiniciar o ciclo.
- **Resolução de *deadlocks*:** o sistema identifica quando dois semáforos de um cruzamento permanecem no mesmo estado (por exemplo, ambos em `VERMELHO`) e intervém forçando o início de um novo ciclo, promovendo o semáforo de maior prioridade ao estado `VERDE`. Essa lógica é implementada em `forceCycleStart()`.

4.4 Simulação de Tráfego e Prioridade Dinâmica

Para emular um ambiente de tráfego realista, cada `SmartTrafficLight` implementa uma abstração do fluxo veicular:

- **Geração de tráfego:** veículos são gerados de forma estocástica, com probabilidade baseada no parâmetro de intensidade de tráfego definido no arquivo de configuração.
- **Cálculo de prioridade local:** cada semáforo calcula sua prioridade dinâmica com base em dois fatores: a **densidade da fila** (número de veículos esperando, com peso 5) e o **fluxo recente** (número de veículos que passaram no último ciclo, com peso 0,5). Essa abordagem busca não apenas favorecer vias com maior volume, mas também aumentar a prioridade de vias secundárias sujeitas à *starvation*.

Em síntese, a arquitetura proposta combina os princípios de redes orientadas a dados com algoritmos distribuídos de controle de tráfego urbano, oferecendo uma solução escalável, tolerante a falhas e adaptável a diferentes topologias. A modelagem local de tráfego, aliada à coordenação central via NDN, permite decisões eficientes e contextualizadas, otimizando o fluxo urbano em tempo real.

5 Projeto de Implantação

A validação do sistema proposto foi realizada por meio de dois cenários de implantação distintos, projetados para testar desde a funcionalidade básica em um ambiente controlado até uma configuração mais complexa e robusta que simula uma distribuição de serviços em rede. A seguir, são detalhados o processo de compilação do sistema, os cenários de simulação desenvolvidos e as arquiteturas de implantação.

5.1 Compilação do Sistema e Geração de Executáveis

O processo de compilação do sistema é gerenciado pelo *CMake*, que é responsável por localizar as dependências, configurar o projeto e gerar os arquivos de construção (*Makefiles*). A compilação resulta em dois executáveis principais, cujos parâmetros são configurados via linha de comando para permitir flexibilidade na execução:

- **`orchestrator`:** Recebe como argumentos o caminho para o arquivo de configuração *YAML* e o nível de *log* desejado (`none`, `error`, `info` ou `debug`).
- **`trafficLight`:** Recebe os mesmos parâmetros de configuração e *log*, além de um parâmetro adicional para seu identificador único (ID). Este ID é utilizado para que o nó possa extrair suas configurações específicas do arquivo *YAML* geral, com base em sua ordem de definição na lista de semáforos.

5.2 Cenários de Simulação

Para permitir a avaliação do sistema sob diferentes condições de tráfego e topologias de rede, foi desenvolvido um mecanismo de configuração de cenários baseado em arquivos no formato *YAML*.

5.2.1 Estrutura dos Arquivos de Cenário

A configuração de cada cenário é definida em um arquivo *YAML* com até quatro seções principais. A seção `traffic-lights` é obrigatória e define a lista de todos os semáforos, cada um com atributos como `name` (um prefixo NDN único), `cycle_time`, `state` inicial, e parâmetros para o cálculo de prioridade (`columns`, `lines`, `intensity`). As seções opcionais são:

- `intersections`: Define cruzamentos entre pares de semáforos.
- `green_waves`: Define sequências de semáforos que devem abrir em sucessão, incluindo o tempo de deslocamento em milissegundos (`travel_time_ms`) entre eles.
- `sync_groups`: Define grupos de semáforos que devem operar com estado e tempo perfeitamente sincronizados.

5.2.2 Cenários Pré-configurados

Dois cenários baseados em áreas de tráfego intenso da cidade de Salvador, BA, foram desenvolvidos para validar o sistema:

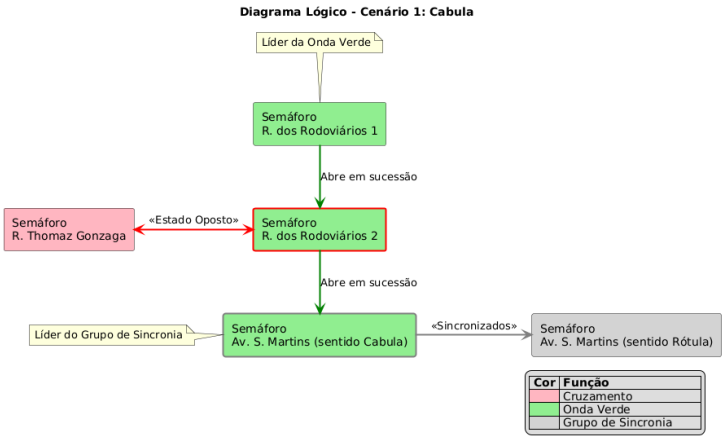

Figura 4 – Diagrama Lógico do Cenário 1: Cabula.

- **Cenário 1: Cabula (`cabula.yaml`):** Este cenário modela uma área de tráfego intenso e complexo envolvendo a Ladeira do Cabula e a saída de Pernambuês. Com um total de cinco semáforos, ele testa a interoperabilidade de todos os mecanismos de controle, contendo um cruzamento, uma onda verde e um grupo de sincronia para travessia de pedestres. A topologia lógica deste cenário é ilustrada na Figura 4.
- **Cenário 2: Dois Leões (`dois-leoes.yaml`):** Este cenário simula o tráfego na região do Largo Dois Leões e Sete Portas. Também com cinco semáforos, seu foco é na coordenação de múltiplos cruzamentos e grupos de sincronia para gerenciar uma interseção principal de grande fluxo. A Figura 5 detalha a complexa rede de inter-relações lógicas deste cenário.

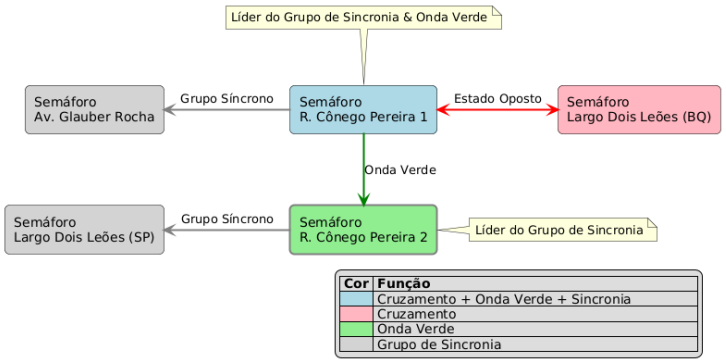

Figura 5 – Diagrama Lógico do Cenário 2: Dois Leões.

5.2.3 Criação de Cenários Customizados

O sistema foi projetado para ser extensível, permitindo a criação de cenários customizados. O processo envolve a criação de um novo arquivo *YAML* na pasta `scenarios/`, a definição de uma nova lista de semáforos e, opcionalmente, a configuração das seções de

grupos lógicos. A aplicação de um novo cenário no ambiente *Docker* é realizada através da alteração de uma variável de ambiente no arquivo `.env` e do ajuste correspondente do número de serviços no arquivo `docker-compose.yml`.

5.3 Cenário de Execução Local (Manual)

O primeiro cenário de implantação, cuja arquitetura está detalhada na Figura 6, consiste em uma execução local em uma única máquina. Nesta configuração, uma única instância do *daemon* NFD é executada para servir como o encaminhador de pacotes. Os executáveis do *Orchestrator* e dos *SmartTrafficLights*, gerados pelo processo de compilação descrito na Seção 5.1, são iniciados manualmente como processos distintos no mesmo ambiente. Este cenário é ideal para depuração rápida e validação da lógica de controle do sistema.

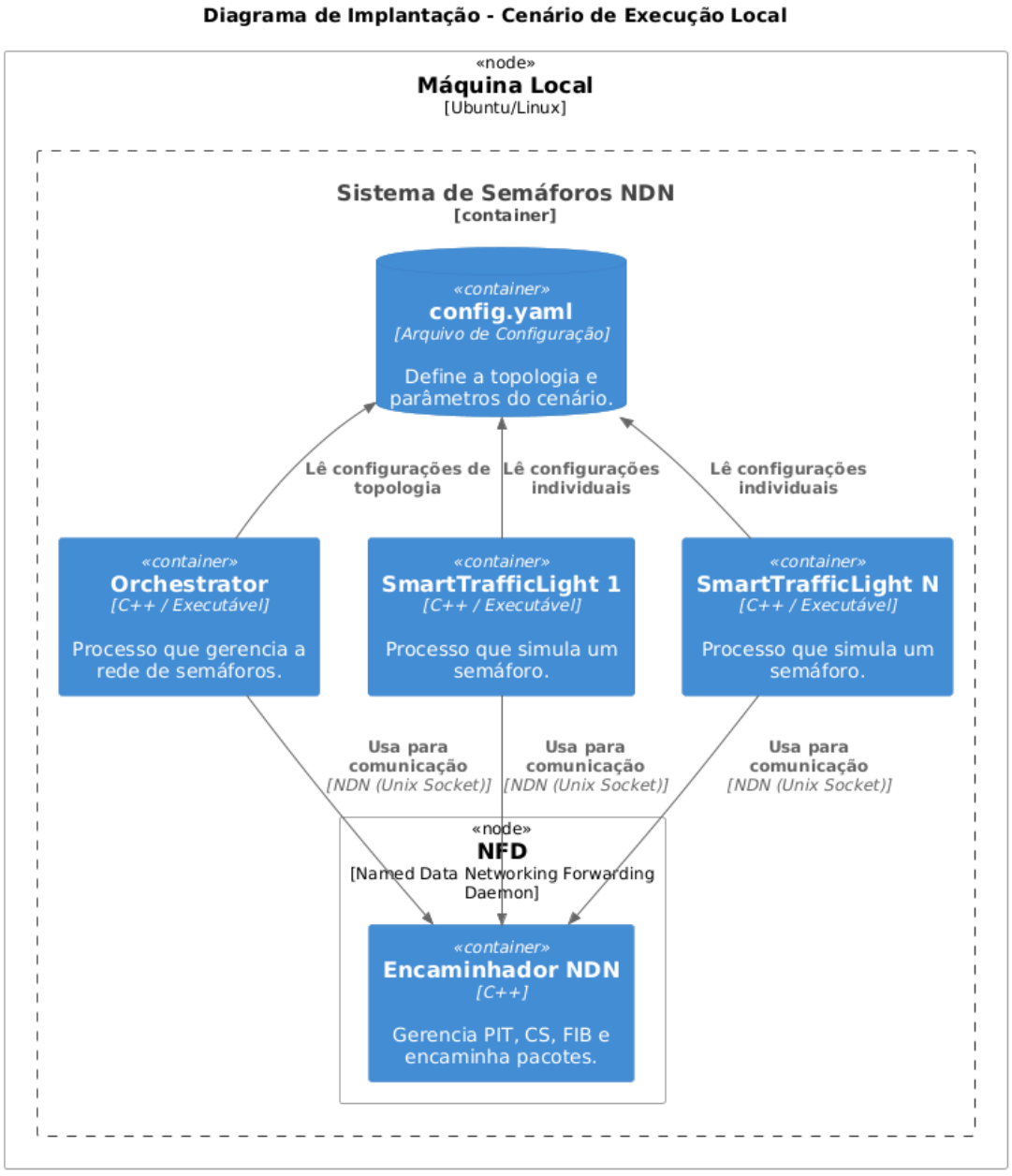

Figura 6 – Diagrama de Implantação - Cenário de Execução Local.

5.4 Cenário de Execução Conteinerizado com Docker

Para garantir um ambiente de execução isolado e reprodutível, foi desenvolvido um segundo cenário, ilustrado na Figura 7, utilizando a tecnologia de contêineres *Docker*.

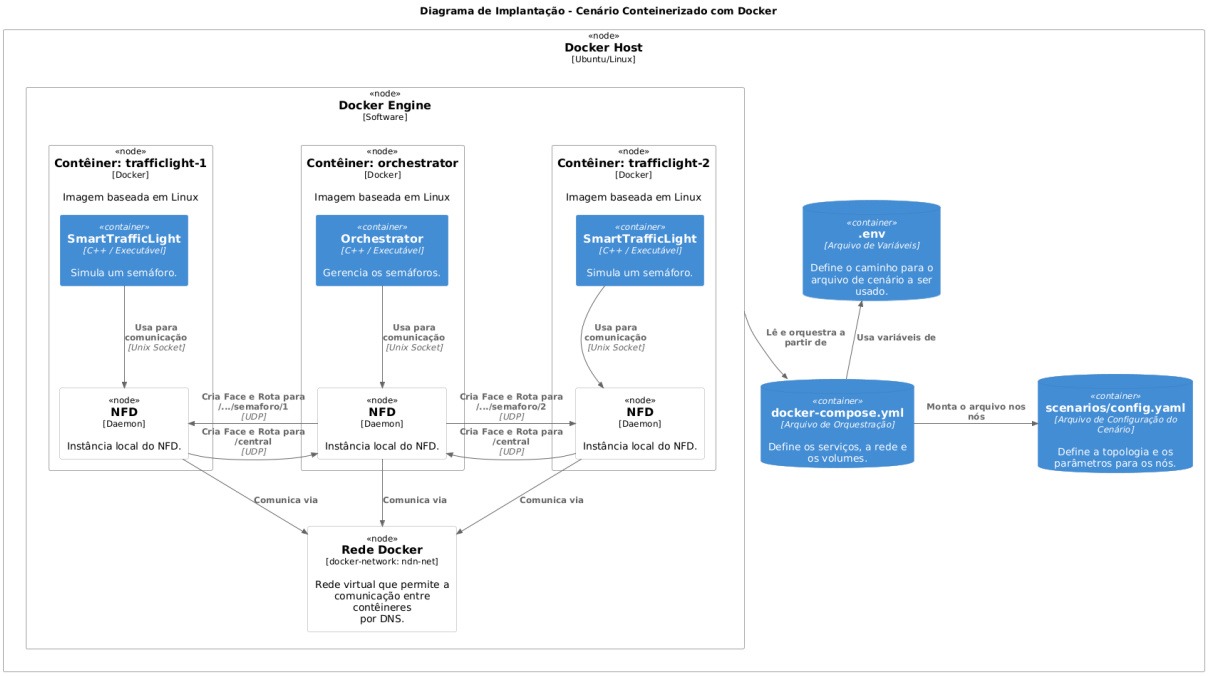

Figura 7 – Diagrama de Implantação - Cenário Conteinerizado com Docker.

5.4.1 Definição da Imagem Docker

Nesta abordagem, a imagem *Docker* é construída através de um `Dockerfile` que utiliza uma abordagem de múltiplos estágios (*multi-stage build*):

- **Estágio 1 (Builder):** Um ambiente de construção completo é utilizado para compilar todas as dependências e o próprio sistema, executando o mesmo processo via *CMake* descrito na Seção 5.1.
- **Estágio 2 (Runtime):** Uma imagem final leve é criada contendo apenas as bibliotecas essenciais, para a qual os executáveis compilados no estágio anterior são copiados.

5.4.2 Orquestração e Configuração de Rede

A orquestração dos contêineres é gerenciada pelo *Docker Compose*. Na inicialização, um *script* `entrypoint.sh` em cada contêiner configura a rede NDN, criando identidades, certificados e estabelecendo *faces* e rotas entre os contêineres através da rede interna do *Docker*, antes de finalmente executar os binários do sistema.

6 Prova de Conceito

Esta seção apresenta os resultados operacionais do sistema, demonstrando as funcionalidades de controle de cruzamento, formação de grupos síncronos e a execução de uma onda verde. Adicionalmente, é realizada uma análise quantitativa do desempenho da rede por meio da métrica de RTT.

6.1 Operação dos Semáforos

Os logs a seguir, capturados de diferentes componentes do sistema, validam a correta implementação das lógicas de controle de tráfego.

6.1.1 Controle de Cruzamento Padrão

Nesta modalidade, os semáforos do cruzamento operam de forma interdependente e exclusiva: enquanto um semáforo permite o fluxo o outro é forçado a um estado de parada para garantir a segurança. A Figura 8 ilustra este cenário, mostrando dois semáforos, `trafficlight-0` e `trafficlight-1`, em um cruzamento. Os logs evidenciam que suas transições de estado são complementares: no instante em que `trafficlight-0` transita para o estado `VERDE`, `trafficlight-1` é obrigatoriamente comutado para o estado `VERMELHO`.

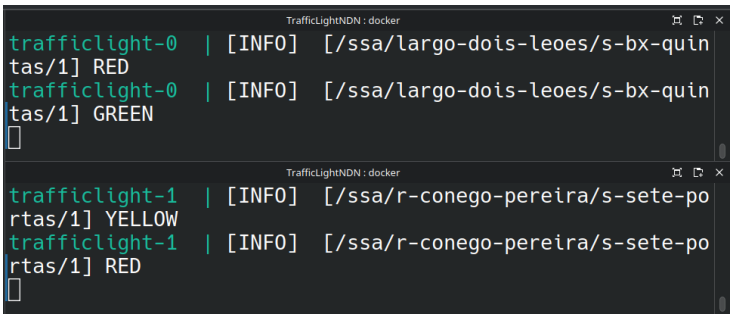

Figura 8 – Logs de semáforos operando de forma interdependente em um cruzamento.

6.1.2 Grupos Síncronos

A funcionalidade de grupos síncronos permite que o orquestrador central comande um conjunto de semáforos para operar de forma coordenada. A Figura 9 exibe o log do orquestrador forçando o semáforo da *Av. Glauber Rocha* a sincronizar com um semáforo líder na *Rua Cônego Pereira*, estabelecendo assim um grupo síncrono.

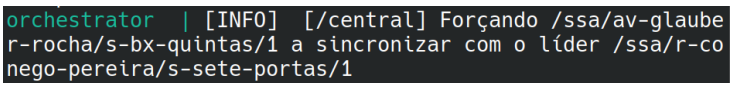

Figura 9 – Log do orquestrador comandando a formação de um grupo síncrono.

O resultado dessa ação pode ser observado na Figura 10, que mostra os logs dos semáforos `trafficlight-1` e `trafficlight-3` alterando seus estados de maneira coordenada, conforme determinado pela lógica do grupo.

Figura 10 – Semáforos membros de um grupo síncrono alterando seus estados.

6.1.3 Onda Verde

A estratégia de Onda Verde é iniciada para otimizar o fluxo em uma via específica. A Figura 11 mostra o momento exato em que o orquestrador inicia o processamento da rotina `onda-verde-conego-pereira`.

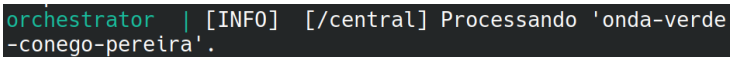

Figura 11 – Orquestrador iniciando o processamento de uma Onda Verde.

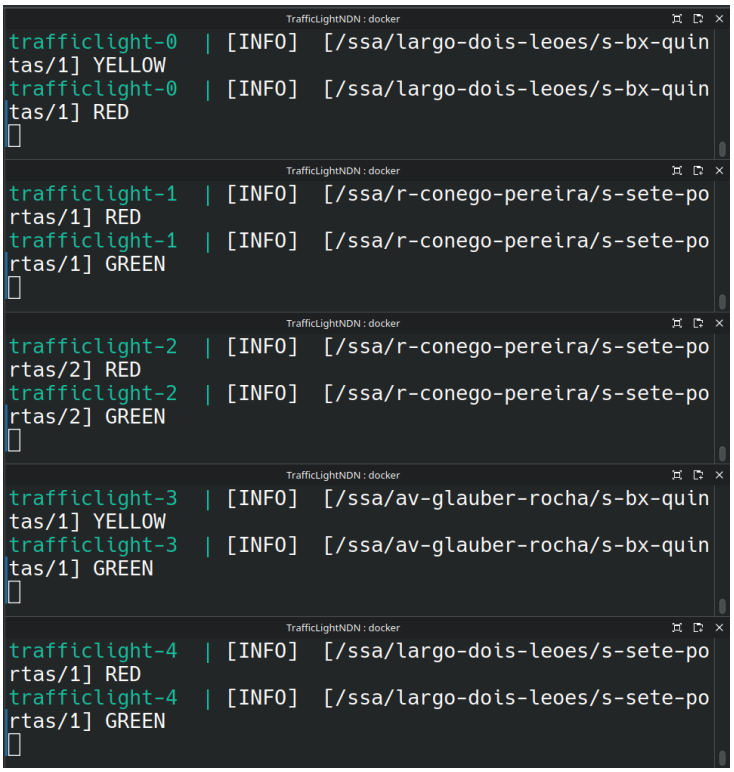

Figura 12 – Execução da Onda Verde com semáforos sequenciais e sincronizados.

A execução da Onda Verde é evidenciada na Figura 12. Nela, observa-se que os semáforos sequenciais na mesma via (`trafficlight-1` e `trafficlight-2` na *Rua Cônego Pereira*) recebem o sinal `VERDE` de forma progressiva. Adicionalmente, como

`trafficlight-3` está em um grupo síncrono com `trafficlight-1`, e `trafficlight-4` com `trafficlight-2`, eles também são forçados a assumir o estado `VERDE` em sincronia com seus respectivos líderes.

6.2 Análise de Desempenho da Rede (RTT)

O RTT é uma métrica crucial que mede o tempo de ida e volta de um pacote de interesse/dados na rede NDN. Ele representa a latência de comunicação entre os nós (orquestrador e semáforos), sendo fundamental para a responsividade do sistema. A Figura 13 apresenta a análise estatística dos RTTs coletados durante a operação do sistema.

```
--- Análise do Arquivo: 'metrics/rtt.csv' ---
Quantidade de valores: 345
Média: 1.7594
Mediana: 2.0
-----------------------------------------
```

Figura 13 – Análise estatística do RTT da comunicação entre os nós.

A partir dos dados, observa-se:

- **Quantidade de valores**: Foram coletadas 345 amostras de RTT ao longo de 1 minuto de execução, fornecendo uma base estatística para a análise. A coleta de dados é realizada pelo `Orchestrator`: ele registra o instante de envio de cada pacote de `Interest` e, ao receber o pacote de `Data` correspondente, calcula a diferença para obter o RTT.
- **Média** ($\bar{x}$): O RTT médio foi de aproximadamente 1.75 ms. Este valor extremamente baixo indica uma alta eficiência e responsividade da rede.
- **Mediana** (M): A mediana foi de 2.0 ms. O fato reforça que a comunicação predominante ocorre em torno de 2.0 ms, com alta estabilidade.

6.2.1 Conclusão da Análise

Os resultados de RTT validam a arquitetura de comunicação baseada em NDN como altamente adequada para aplicações de controle de tráfego em tempo real. A baixa latência, evidenciada pela média e mediana na casa de 2 milissegundos, garante que os comandos de controle sejam entregues e confirmados com a rapidez necessária para uma operação segura e eficiente, mesmo em cenários dinâmicos como a Onda Verde e a formação de Grupos Síncronos.

7 Conclusão

Este trabalho se propôs a investigar e avaliar a viabilidade da arquitetura NDN como base para a comunicação em um sistema de semáforos inteligentes. Através do desenvolvimento e da avaliação de um protótipo funcional, foi possível não apenas validar a aplicação da NDN neste cenário, mas também demonstrar seu desempenho superior por meio de resultados práticos e quantitativos.

A avaliação prática concluiu que a comunicação via NDN é extremamente eficaz para o domínio proposto. A principal vantagem observada foi a dissociação entre o serviço e sua localização física, um princípio fundamental da NDN. Os nós *SmartTrafficLight* não necessitam conhecer o endereço de rede do *Orchestrator*; eles meramente requisitam um dado por seu nome (e.g., `/central/command/...`). Essa agilidade e flexibilidade foram comprovadas pela análise de desempenho da rede, que registrou um RTT mediano de apenas 2.0 ms e médio de 1.75 ms. Estes valores atestam a capacidade do sistema de operar com a alta responsividade exigida por aplicações de controle de tráfego em tempo real. Esta característica, aliada à segurança intrínseca ao dado, confere uma robustez e mobilidade inerentes ao sistema: a central de controle pode ser substituída ou migrada para outra máquina sem que qualquer alteração seja necessária nos semáforos, contanto que a nova instância utilize a mesma identidade criptográfica.

Adicionalmente, este trabalho contribui com uma perspectiva distinta sobre a aplicação de redes NDN. Enquanto grande parte da literatura acadêmica foca no uso da NDN para nós móveis e comunicação veicular (V2V/V2I), este estudo demonstra sua eficácia para uma infraestrutura de nós majoritariamente estacionários. Validou-se o uso da NDN não apenas para a entrega de conteúdo, mas como a espinha dorsal de um sistema de comando e controle com poder de decisão macro (no *Orchestrator*). Conforme detalhado na Prova de Conceito, o protótipo implementado foi capaz de gerenciar com sucesso lógicas complexas de tráfego: as operações de cruzamento, a dinâmica de grupos síncronos e a execução de ondas verdes progressivas foram validadas por meio de logs operacionais, confirmando o controle efetivo do orquestrador sobre a infraestrutura em cenários configuráveis.

Trabalhos Futuros

Apesar dos resultados promissores, este trabalho abre portas para diversas linhas de pesquisa e aprimoramentos futuros. Sugere-se:

- **Expansão da Lógica de Controle:** Aprimorar o *Orchestrator* para suportar cruzamentos complexos com três ou mais vias, que representam um desafio comum em grandes centros urbanos e exigem algoritmos de coordenação mais sofisticados.
- **Integração com Simuladores de Tráfego:** Integrar o sistema com simuladores de micromobilidade, como o SUMO (*Simulation of Urban MObility*), para validar a eficácia dos algoritmos de otimização com um fluxo de veículos mais realista e baseado em dados do mundo real.

- **Aplicação de *Edge/Fog Computing*:** Explorar conceitos de *Edge* e *Fog Computing*, distribuindo instâncias do *Orchestrator* para gerenciar sub-regiões da cidade de forma mais localizada e escalável. Neste contexto, a orquestração dos contêineres poderia evoluir do *Docker Compose* para uma solução mais robusta e adequada para ambientes de borda, como o K3s.

Referências

AL-QUTWANI, M.; WANG, X. Smart traffic lights over vehicular named data networking. *Sensors*, MDPI, v. 19, n. 5, p. 1103, 2019. Disponível em: <https://www.mdpi.com/2078-2489/10/3/83>. Citado na página 19.

ARAúJO, S. C. *Controlador de tráfego: semáforo inteligente*. Dissertação (Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação)) — Centro Universitário de Brasília (UniCEUB), Brasília, DF, dec 2006. Disponível em: <https://repositorio.uniceub.br/jspui/handle/123456789/3290>. Citado na página 19.

BERNAL, L. M.; FERREIRA, M. A. G. Evolução dos indicadores de mobilidade urbana na região metropolitana de são paulo. *Programa de Pós-Graduação em Engenharia Urbana, Universidade Federal de São Carlos (UFSCar)*, São Carlos, SP, Brasil, 2016. Mestranda: bernallu@gmail.com, Prof. Dr.: dmag@ufscar.com.br. Citado na página 8.

CINTRA, M. Os custos do congestionamento na capital paulista. *Revista Conjuntura Econômica*, 2008. Acessado em: 3 jun. 2025. Disponível em: <https://hdl.handle.net/10438/14631>. Citado na página 8.

CLEMENS, J. M.; A., S. Analysis of timing and synchronization algorithms in distributed systems. *International Research Journal of Engineering and Technology (IRJET)*, v. 8, n. 5, May 2021. ISSN 2395-0056. Impact Factor: 7.529, ISO 9001:2008 Certified Journal. Citado na página 17.

COULOURIS, G. et al. *Sistemas Distribuídos: Conceitos e Projeto*. 5. ed. Porto Alegre: Bookman, 2013. Recurso eletrônico. Citado na página 17.

FACHINI, M. P. et al. Internet das coisas: Uma breve revisão bibliográfica. *Conexões - Ciência e Tecnologia*, v. 11, n. 6, p. 85–90, 2017. Disponível em: <https://doi.org/10.21439/conexoes.v11i6.1007>. Citado na página 13.

FENERICH, A. T. *Fatores e nível de estresse no trânsito*. 2016. Trabalho de Conclusão de Curso (Especialização em Engenharia de Segurança do Trabalho). 55 f. Disponível em: <http://repositorio.utfpr.edu.br/jspui/handle/1/17577>. Citado na página 8.

FONSECA, J. P. S. et al. Congestión vehicular y contaminación ambiental en lima metropolitana. *Revista Lasallista de Investigación*, v. 19, n. 1, p. 152–164, 2022. ISSN 1794-4449. Publicado online em 6 fev. 2023. Disponível em: <https://doi.org/10.22507/rli.v19n1a9>. Citado na página 8.

GOMES, A. P. *Os Benefícios da Operação de Semáforos em Tempo Real*. [S.l.], 2014. Acessado em: 3 jun. 2025. Disponível em: <http://sinaldetransito.com.br/artigos/benefÃŋcios_tempo_real.pdf>. Citado na página 8.

JACOBSON, V. et al. *Named Data Networking (NDN) Project*. [S.l.], 2010. Disponível em: <https://named-data.net/techreport/TR001ndn-proj.pdf>. Citado 2 vezes nas páginas 9 e 13.

JESUS, I. M. de Sousa de. *Sistemas Inteligentes de Transporte*. Tese (Doutorado) — Universidade do Porto (Portugal), 2001. ProQuest Dissertations & Theses, No. 31024795. Disponível em: <https://www.proquest.com/openview/3cd973f226755d4b3328876a25dfa21c/1?pq-origsite=gscholar&cbl=2026366&diss=y>. Citado na página 15.

KOLLS, K. H.; SIMPLICIO, L. G. *Semáforo inteligente com comunicação via protocolo ESP-NOW*. Ponta Grossa, 2022. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/26084>. Citado na página 20.

LEMOS, A. Cidades inteligentes. *GV-EXECUTIVO*, v. 12, n. 2, p. 46–49, 2013. Disponível em: <https://doi.org/10.12660/gvexec.v12n2.2013.20720>. Citado na página 8.

MOREIRA, M. D. D. et al. Internet do futuro: Um novo horizonte. In: SBC. *Anais do XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. Florianópolis, Brasil: SBC, 2014. p. 29–42. Disponível em: <https://www.gta.ufrj.br/ensino/cpe728/MFCD09.pdf>. Citado na página 8.

NETO, J. C. *SEMÁFORO: SER OU NÃO SER INTELIGENTE? Uma comparação entre o controle semafórico em tempos fixos e o em tempo real*. [S.l.], 2016. Acessado em: 3 jun. 2025. Disponível em: <https://nou.sinaldetransito.com.br/wp-content/uploads/2021/03/semaforo_ser_ou_nao_ser.pdf>. Citado 3 vezes nas páginas 8, 15 e 16.

SAMPAIO, L. N. et al. Revisitando as icns: Mobilidade, segurança e aplicações distribuídas através das redes de dados nomeados. In: *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. Belo Horizonte, Brasil: SBC, 2021. p. 49–66. Disponível em: <https://books-sol.sbc.org.br/index.php/sbc/catalog/view/81/352/608-1>. Citado na página 11.

SANTOS, B. P. et al. Internet das coisas: da teoriaa prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos*, 2016. Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>. Citado 2 vezes nas páginas 13 e 14.

SHAHEEN, S.; FINSON, R. *Intelligent Transportation Systems*. UC Berkeley: Recent Work, 2013. Accessed: 2025-06-05. Disponível em: <https://escholarship.org/uc/item/3hh2t4f9>. Citado na página 15.

SHANG, W. et al. *Challenges in IoT Networking via TCP/IP Architecture*. Los Angeles, CA, 2016. Revision 1: February 10, 2016. Disponível em: <https://named-data.net/wp-content/uploads/2016/02/ndn-0038-1-challenges-iot.pdf>. Citado 2 vezes nas páginas 9 e 10.