

Gerenciamento Baseado em Intenção para a Open RAN: Automação Inteligente de Configuração de Redes via *Chatbot* *

João Vitor A. Garcês^{1,2}, Nicollas R. de Oliveira^{1,2}, João André C. Watanabe²,
Ricardo M. Tanaka^{1,2}, Daniel O. de Arruda², Breno T. Leite^{1,2},
Caio P. Galdino³, Rodrigo S. Couto³, Igor M. Moraes²,
Dianne S. V. de Medeiros^{1,2}, Diogo M. F. Mattos^{1,2}

¹Laboratório de Redes de Nova Geração (LabGen) - TET/PPGEET/UFF

²Laboratório MídiaCom - TET/PPGEET/TCC/PGC/UFF

Universidade Federal Fluminense - UFF

³Grupo de Teleinformática e Automação (GTA) - PEE/COPPE

Universidade Federal do Rio de Janeiro - UFRJ

Abstract. *The translation of complex policies into practical network configuration actions is often conducted by human operators, resulting in errors that affect network performance. In Open RAN networks, this challenge becomes even more relevant, considering the objective of intelligent configuration automation. This paper proposes an intent-based management system, which receives commands in natural language through a chatbot and converts them into network policies. Proposal evaluation is conducted on a software-defined network emulation on Mininet, controlled by ONOS. The system prototype integrates Rasa chatbot with ONOS controller, allowing the configuration of network flows to be derived from natural language instructions. The results indicate that the system fulfils intents in constant time, up to 100 ms, even with an increasing load of flows in the network switches.*

Resumo. *A tradução de políticas complexas em ações práticas de configuração da rede é, frequentemente, conduzida por operadores humanos, resultando em erros que afetam o desempenho da rede. Em redes Open RAN, esse desafio torna-se ainda mais relevante, considerando o objetivo de automação inteligente da configuração. Este artigo propõe um sistema de gerenciamento baseado em intenção, que recebe comandos em linguagem natural através de um chatbot e os converte em políticas de rede. A avaliação da proposta é conduzida em uma emulação de rede definida por software no Mininet. O protótipo do sistema integra o chatbot Rasa ao controlador ONOS, permitindo que a configuração dos fluxos de rede seja derivada de instruções em linguagem natural. Os resultados indicam que o sistema atende às intenções em tempo constante, de até 100 ms, mesmo com carga crescente de fluxos nos comutadores.*

1. Introdução

O potencial econômico da implantação da quinta geração de redes móveis (5G) no Brasil, destacado pelo Ministério da Economia, representará um benefício de até R\$ 590 bilhões por ano para a economia brasileira. Somente a demanda potencial de software

*Este trabalho foi financiado com recursos da RNP, CNPq, CAPES, FAPERJ.

para a 5G, irá gerar uma expectativa de valor total de R\$ 101 bilhões, até 2031¹. Apesar das promessas revolucionárias da 5G em diversos setores, como indústria e logística, as atuais implementações da rede tendem a ser monolíticas e fechadas, dificultando a inovação e a interoperabilidade. Nesse contexto, iniciativas de redes abertas, como a Rede de Acesso via Rádio Aberta (*Open Radio Access Network* – Open RAN), têm ganhado destaque e possibilitam a interoperabilidade entre equipamentos de diferentes fornecedores [Couto et al., 2023, Bonati et al., 2023]. A arquitetura da Open RAN desagrega software e hardware e padroniza interfaces abertas e interoperáveis, flexibilizando as RANs [de Oliveira et al., 2023, Lacava et al., 2024]. Com a Open RAN, desagrega-se a arquitetura tradicional da estação base, virtualizando funções de rede para implantação em hardware padrão e usando Controladores Inteligentes de RAN (*RAN Intelligent Controllers* - RICs) para otimizar o gerenciamento de recursos em tempo real. Em um cenário de operação complexo, o uso dos RICs automatiza a configuração e monitoramento da RAN e possibilita a implantação do Gerenciamento Baseado em Intenção (*Intent-based Management* - IBM). Com o IBM, o operador não precisa configurar explicitamente a rede, lidando apenas com declarações em alto nível, denominadas intenções para cumprir os objetivos de automação da Open RAN [Couto et al., 2023].

O IBM automatiza a implementação de objetivos de negócios em redes, traduzindo tais objetivos em políticas bem definidas e monitorando a rede para apoiar a rápida adaptação de novos serviços nas RANs abertas e garantir os resultados desejados [Jacobs et al., 2021, Martini et al., 2023]. O IBM compreende a tradução de intenções, a ativação e garantia de execução correta das intenções. Atualmente, as implementações de IBM dependem de tecnologias como Redes Definidas por Software (*Software Defined Networking* – SDN) e técnicas de aprendizado de máquina para traduzir linguagens de alto nível em configurações de rede, aumentando o desempenho da operação automatizada da rede e facilitando a implantação rápida de serviços e integração em todas as camadas da rede [Kavehmadavani et al., 2023, De Alwis et al., 2023].

Este artigo propõe um sistema de gerenciamento de redes baseado em intenção, no qual o usuário se expressa em linguagem natural através de um assistente computacional de diálogo (*chatbot*). O sistema proposto integra o arcabouço de *chatbots* Rasa ao controlador de rede definida por software ONOS e permite que o usuário configure a rede por meio de texto em linguagem natural de alto nível. Um protótipo do sistema proposto foi desenvolvido e é avaliado em uma rede emulada na plataforma Mininet. O objetivo é demonstrar a viabilidade do sistema proposto. Para tanto, verifica-se o tempo decorrido até a instalação das intenções focadas em encaminhamento *Host-to-Host* (H2H), *Multi-to-Single-Point* (M2SP) e *Single-to-Multi-Point* (S2MP). A avaliação do protótipo revela a viabilidade de configuração da rede através das entradas em alto nível via *chatbot* e com baixa latência na aplicação das intenções na rede. Os resultados mostram que o sistema atende 10.000 intenções em aproximadamente 2,5 segundos. Trabalhos anteriores focam o desenvolvimento de modelos de linguagem para o gerenciamento baseado em intenção em redes universitárias [Kiran et al., 2018, Jacobs et al., 2021] e a identificação de conflitos entre intenções [de Oliveira et al., 2023]. Contudo, esses trabalhos carecem de um assistente computacional de diálogo integrado ao controlador de rede, capaz de traduzir

¹Disponível em <https://www.undp.org/pt/brazil/news/levantamento-sobre-5g-no-brasil-indica-beneficio-de-quase-r-600-bi-ao-ano-para-economia-brasileira>.

entradas de texto expressas em linguagem natural em operações de configuração da rede. Assim, a principal contribuição deste artigo é a integração do assistente computacional de diálogo ao controlador da rede.

O restante do artigo está organizado da seguinte forma. A Seção 2 descreve o gerenciamento baseado em intenção e o uso de *chatbots* para definição de intenções. A Seção 3 discute os trabalhos relacionados. A Seção 4 detalha o sistema proposto, sua arquitetura e seus módulos. A Seção 5 descreve os cenários de avaliação e discute os resultados dos experimentos com o protótipo do sistema proposto. Por fim, a Seção 6 conclui o artigo e apresenta direções futuras de pesquisa.

2. Gerenciamento Baseado em Intenção e Assistentes de Diálogo

O Gerenciamento Baseado em Intenção (*Intent-Based Management* – IBM) envolve o uso de instruções de alto nível que definem o comportamento da rede de acordo com as especificações da operadora, expressas por meio de metas de negócios ou Indicadores-Chave de Desempenho (*Key Performance Indicators* – KPIs). A intenção consiste em uma meta declarativa de negócios de alto nível que opera no nível da rede e do serviço, ao invés de se concentrar em dispositivos individuais. Uma rede que utiliza IBM é capaz de alcançar os Objetivos de Nível de Serviço (*Service Level Objectives* – SLOs) sem a necessidade de programação explícita da rede, dado o processo automatizado de tradução das intenções. Assim, a tradução extrai do texto de entrada configurações precisas a nível de dispositivo que podem ser aplicadas à rede. Ao contrário das políticas, as intenções definem resultados e objetivos operacionais amplos, sem detalhar procedimentos ou ações específicas necessárias.

Um Sistema Baseado em Intenção (*Intent-Based System* – IBS) geralmente compreende cinco estágios de processamento, constituindo um sistema de automação de circuito fechado [Zheng et al., 2022, Wei et al., 2020]. O estágio inicial, **Ingestão de Intenções**, reconhece e coleta intenções de rede por meio de interações com o usuário. Posteriormente, no estágio de **Tradução de Intenções**, as intenções coletadas são interpretadas e convertidas em um formato compreensível pelos dispositivos e sistemas subjacentes. Essa conversão envolve a decomposição da intenção original em comandos, parâmetros de configuração ou expressões com uma sintaxe predefinida. No terceiro estágio, a **Resolução de Políticas** valida as intenções em relação às políticas administrativas existentes para detectar e mitigar conflitos, gerando uma nova política operacional para implementação. Esse processo de validação é crucial, pois políticas conflitantes podem desencadear ações destrutivas, resultando em comportamento incorreto da rede. No estágio de **Orquestração de Políticas**, as políticas são implementadas e transformadas em ações tangíveis na infraestrutura de rede. Finalmente, o estágio de **Conformidade de Política** depende de dados de telemetria de rede para verificar a conformidade contínua do desempenho com os requisitos da intenção durante todo o seu ciclo de vida. A detecção de quaisquer desvios no desempenho requer refinamento e reajuste da intenção, garantindo a entrega sustentada dos níveis de qualidade de serviço desejados.

Assistentes computacionais de diálogo podem atuar no estágio de Ingestão de Intenções. Esses assistentes, ou *chatbots*, são aplicações baseadas em inteligência artificial que simulam diálogos com usuários por interfaces de mensagens de texto. A complexidade dessas aplicações está inerentemente ligada à sua capacidade de lidar com a

imprevisibilidade do texto de entrada. Os *chatbots* de baixa complexidade tipicamente empregam uma abordagem de interpretação baseada em regras, pautada na interação segundo um conjunto predefinido de comandos ou perguntas para obter respostas. Essa abordagem, apesar de simples implementação, restringe a interpretação devido à sua natureza rígida. Paralelamente, *chatbots* de alta complexidade são baseados em linguagem natural. Expressar intenções diretamente em linguagem natural oferece vários benefícios, como garantir interfaces de gerenciamento abstratas em diferentes equipamentos, evitar o aprendizado de novas linguagens de programação e reduzir erros humanos na tradução de políticas. No entanto, a constante evolução das línguas naturais torna cada vez mais desafiador estabelecer regras explícitas para computadores [Rahman et al., 2017]. Essa flexibilidade pode introduzir ambiguidade e desafiar os sistemas a captar as intenções dos operadores de forma inequívoca. Para resolver isso, combinar técnicas de Processamento de Linguagem Natural (PLN) com algoritmos de Aprendizado Profundo aprimora a capacidade de tradução do *chatbot*, tornando-o mais robusto e adaptável a diversas estruturas linguísticas, independentemente do nível de abstração.

Internamente, o estágio de Tradução de Intenções pode incluir uma sequência de técnicas de limpeza e modelagem de texto. Entre essas técnicas estão a tokenização, correção ortográfica, Reconhecimento de Entidades Nomeadas (*Named Entity Recognition* - NER) e interpretação de sinônimos. A tokenização adota um critério delimitador para transformar uma intenção, expressa como uma frase contígua, em uma lista de *tokens*. Cada *token*, ou seja, palavra, expressão ou pontuação, pode ser manipulado individualmente e é composto por uma sequência de caracteres. Para lidar com possíveis erros tipográficos, é aconselhável a correção ortográfica, comparando cada *token* com seu equivalente mais próximo em um dicionário adaptado. A técnica de NER visa identificar nomes de software, hardware ou quaisquer nomes próprios ou expressões relevantes na intenção. Paralelamente, a interpretação de sinônimos visa compreender variações de termos, assegurando um tratamento lógico igualitário ou mapeando as múltiplas variações para um termo único [Jacobs et al., 2018, Mahtout et al., 2020].

Independentemente das estratégias de ingestão e tradução escolhidas, as intenções processadas podem ser estruturadas em diversas sintaxes de representação. Dentre essas sintaxes, a *Network Intent Language* (NILE) é uma linguagem especializada que serve como uma camada intermediária entre as intenções em linguagem natural e as políticas de nível inferior [Jacobs et al., 2018, Riftadi e Kuipers, 2019]. Embora as linguagens especializadas possam acomodar requisitos específicos do gerenciamento baseado em intenção, é possível utilizar linguagens mais genéricas. O formato *eXtensible Markup Language* (XML) possui uma natureza estruturada e extensível, facilitando a representação dos vários aspectos da rede [Kiran et al., 2018]. O formato baseado em *Web Ontology Language* (OWL) é indicado quando se deseja uma estruturação semântica rica de intenções de rede [Kiran et al., 2018]. Esse formato permite uma representação diferenciada dos relacionamentos e restrições dentro da rede. O *JavaScript Object Notation* (JSON) pode ser utilizado como um formato que combina a simplicidade e flexibilidade do JSON na estruturação de cenários de comunicação em rede [Cesila et al., 2023, Mahtout et al., 2020].

3. Trabalhos Relacionados

Os trabalhos relacionados referem-se a propostas de criação de sistemas IBM com interfaces baseadas em assistentes conversacionais e na avaliação de desempenho de controladores SDN na implementação de intenções na rede.

Jacobs *et al.* propõem o LUMI, um sistema de refinamento e interpretação de intenções que oferece gerenciamento em alto nível de redes de campi universitários, permitindo que as operadoras comuniquem suas intenções à rede por meio de linguagem natural utilizando um *chatbot* [Jacobs et al., 2021]. Baseado no Rasa e integrado ao Dialogflow, o *chatbot* do LUMI compreende textos não estruturados através de um pipeline que inclui: i) extração de informação, ii) representação estruturada das intenções processadas na sintaxe NILE, iii) verificação de conformidade e detecção de conflitos, e iv) implementação na rede. Mediante várias interações de *feedback*, o usuário participa tanto do refinamento da intenção original, propondo inclusões ou alterações de parâmetros, quanto da verificação e confirmação da ausência de quaisquer estruturas conflituosas. Entre as avaliações conduzidas, os autores quantificaram a usabilidade e a eficiência extrativa segundo a percepção dos usuários. Em uma avaliação quantitativa, métricas de recuperação da informação também foram adotadas para mensurar o impacto das operações de *feedback* na extração de entidades das intenções.

Mahtout *et al.* apresentam o EVIAN, uma plataforma de renderização de intenções que traduz requisitos de rede inseridos por múltiplos usuários em configurações de enlaces. Diferente dos sistemas IBM existentes, o EVIAN incorpora alguns mecanismos de segurança, permitindo que apenas usuários autenticados acessem e implementem novas intenções na rede. Operando em um modelo cliente-servidor-renderizador, o EVIAN possui um funcionamento segmentado em três etapas: a admissão das intenções do usuário através de um *chatbot* integrado à API da plataforma Slack², a resolução de conflitos e a verificação de políticas gerenciadas pelo servidor, e o mapeamento de entidades da intenção em ações de dispositivos de rede executadas pelo renderizador. As interações entre componentes ocorrem após a estruturação das intenções em grafos RDF (*Resource Description Framework*), facilitando o armazenamento e recuperação de parâmetros segundo um esquema ontológico. Antes da implantação da ação, o EVIAN solicita a aprovação do usuário para execução [Mahtout et al., 2020].

Desenvolvido por Kiran *et al.* para ambientes acadêmicos, o iNDIRA é uma ferramenta que automatiza o provisionamento de recursos de rede e a transferência de arquivos com base em intenções expressas em linguagem natural através de uma Interface de Linha de Comando (*Command Line Interface - CLI*) [Kiran et al., 2018]. Utilizando raciocínio de máquina (*Machine Reasoning*) e um analisador sintático, a ferramenta extrai, interpreta e negocia os requisitos de QoS correspondentes aos comandos especificados. O tratamento textual inclui remoção de palavras desnecessárias e reconhecimento, eventualmente seguido de substituição, de palavras-chave por meio de um dicionário de sinônimos. O desempenho da proposta é avaliado pelo tempo de processamento de intenções representadas em grafos RDF, de diferentes tamanhos e complexidades.

Cesila *et al.* implementam o Chat-IBN-Rasa, um assistente conversacional baseado no arcabouço Rasa, capaz de intermediar as solicitações de estabelecimento

²Disponível em <https://slack.com/>.

de conexão entre locais de uma rede óptica, através da seleção de opções predefinidas [Cesila et al., 2023]. Durante uma sequência de interações com o usuário, o *chatbot* coleta parâmetros relacionados ao enlace, como disponibilidade, tipo de proteção, taxa de dados e distância e, posteriormente, avalia a disponibilidade de caminhos disponíveis com requisitos. A pluralidade de opções exibidas, a verificação de disponibilidade das escolhas e a recomendação de rotas alternativas são realizadas mediante consultas a uma base de dados. Caso haja concordância com o operador, a intenção configura o controlador para executar os comandos necessários nos dispositivos físicos ou virtuais. A Tabela 1 sumariza as principais características e funcionalidades associadas a trabalhos anteriores relacionados ao gerenciamento de rede empregando *chatbots*.

Tabela 1. Comparação de trabalhos sobre gerenciamento de redes usando agentes conversacionais (*chatbots*).

Aspectos	Características e Funcionalidades	Lumi	EVIAN	iNDIRA	Chat-IBN-RASA	Proposta
Forma de Interação	Digitação de Textos	✓	✓	✓		✓
	Seleção de Opções				✓	
Expressão das Intenções	Linguagem Natural	✓	✓	✓		✓
	Não Aplicável				✓	
Interface de Interação	Dialogflow	✓				
	Slack		✓			
	Telegram					✓
	WhatsApp					✓
	Linha de Comando			✓		
Técnicas de Processamento de Linguagem Natural	Reconhecimento de Entidades	✓	✓	✓		✓
	Interpretação de Sinônimos			✓		✓
	Mapeamento de Parâmetros			✓	✓	✓
	Correção Ortográfica				✓	✓
Sintaxe de Representação das Intenções	NILE	✓				✓
	XML			✓		
	OWL			✓		
	JSON		✓		✓	
	Não Especificado					
Metodologia de Detecção de Conflitos	Checagem pelo Operador	✓				
	Checagem de Disponibilidade		✓			
	Atualização Reversa					✓
	Não Especificado			✓		
Métrica de Avaliação	Latência	✓		✓		✓
	Usabilidade da Interface	✓				
	Não Aplicada		✓		✓	

Nguyen *et al.* conduziram uma análise sistemática dos relatórios de erros e vulnerabilidades disponíveis publicamente sobre o controlador ONOS, identificados durante a orquestração de redes baseadas em intenções [Nguyen et al., 2022]. Os autores revelaram que 61% dos erros associados à submissão de intenções no controlador são falhas interpretativas decorrentes de intenções sintaticamente corretas, mas que descrevem um comportamento lógico inadequado para a rede. Essas falhas, intituladas de erros semânticos, incluem a perda de intenções, discrepâncias nos dados solicitados pelos usuários e corrupção de estados. Diferentemente dos erros sintáticos, que frequentemente

resultam em corrupção de memória e falhas, a ocorrência de erros semânticos impacta discretamente o desempenho da rede, ocasionando desde falhas de conectividade até a desestruturação da topologia.

Addad *et al.* realizam uma avaliação experimental do desempenho do controlador ONOS ao executar um conjunto de intenções padrão submetidas por diferentes interfaces, RESTful e CLI [Addad et al., 2018]. Variando a carga de trabalho e o tipo de intenções, os autores analisaram o impacto segundo dois critérios: o tempo para instalação bem-sucedida e a quantidade de intenções gerenciáveis por uma única instância do ONOS. Embora a interface RESTful permita que aplicativos e serviços solicitem diretamente ao ONOS a atualização da configuração da rede, resultados revelaram que a interface RESTful é significativamente mais custosa computacionalmente do que a CLI.

Como mostrado na Tabela 1, o sistema proposto contrasta com trabalhos anteriores por executar desde a ingestão das intenções em linguagem natural através de um *chatbot*, o processamento de linguagem natural, a tradução das instruções em políticas de rede até a aplicação das configurações na rede. O sistema proposto prevê a utilização em um ambiente de rede amplo aplicado à Open RAN, enquanto os demais trabalhos focam redes universitárias ou não abordam todas as etapas do gerenciamento baseado em intenção.

4. O Sistema Proposto de Gerenciamento de Rede Baseado em Intenção

O sistema proposto automatiza a configuração de políticas na RAN por intenções expressas em linguagem natural via *chatbot*. A Figura 1 mostra a arquitetura modular do sistema, que permite a interação de dois módulos principais. O módulo Assistente Conversacional é composto por dois componentes: um *frontend*, responsável pela interação com o operador por aplicativos de mensagens instantâneas, como WhatsApp e Telegram, e um *backend*, encarregado do processamento das intenções inseridas e da geração dos diálogos textuais com o operador. Esse *backend* atua como uma xApp dentro do Controlador Inteligente da RAN de Não Tempo Real (*Non-Real-Time RAN Intelligent Controller – Non-RT RIC*) da RAN. As intenções processadas alimentam o módulo iApp, que é uma aplicação ONOS responsável pela tradução para linguagem de baixo nível e pela resolução de potenciais conflitos entre intenções. Por fim, as intenções válidas são mapeadas em políticas aplicáveis pelo *ONOS Intent Framework*, que orquestra a execução das políticas nos dispositivos de telemetria da rede.

O **Assistente Conversacional** é baseado no arcabouço Rasa³, que oferece ferramentas e bibliotecas para criação de fluxos interativos de diálogo entre usuários e máquinas. O Rasa dispõe de três componentes: o Agente, o Rasa NLU (*Natural Language Understanding*) e o Rasa Core, responsáveis pelo gerenciamento do fluxo de solicitação-resposta, pela compreensão da linguagem natural e pelo gerenciamento do diálogo. O Rasa NLU atua na extração de entidades relevantes, classificação de intenções e recuperação de respostas através de rotinas personalizáveis de processamento de linguagem natural. O Rasa Core concentra-se na análise dos comandos processados de cada interação e na orientação das decisões de diálogo, prezando pela manutenção do contexto semântico da conversa.

No Rasa NLU, o modelo DIET (*Dual Intent and Entity Transformer*) é uma

³Disponível em <https://rasa.com/>.

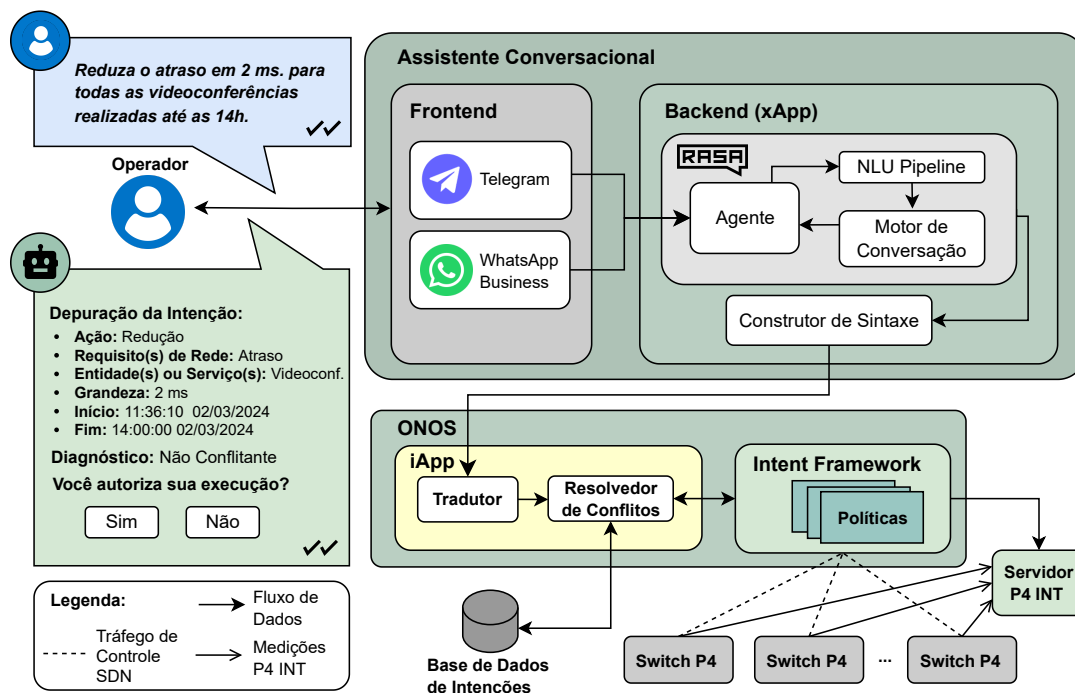


Figura 1. O sistema modular garante a captura e processamento de intenções em linguagem natural por meio de interações com operadores via *chatbot*. Após capturadas, as intenções são traduzidas para uma linguagem compreensível pelo controlador e validadas em relação aos possíveis conflitos com o estado atual da rede. Caso aprovada, a intenção é mapeada em políticas aplicáveis e executada nos dispositivos periféricos de telemetria.

rede neural treinada para classificar as intenções genéricas em intenções válidas, capacitando o sistema a lidar com entradas inéditas após treinamento⁴. Essa classificação é precedida por um *pipeline* de processamento textual que envolve cinco procedimentos básicos: tokenização, correção ortográfica, extração de entidades relevantes, interpretação de sinônimos e vetorização. A tokenização adotou o espaço como caractere delimitador. A correção ortográfica aplicada considera a distância *Levenshtein*, métrica que calcula o número mínimo de operações necessárias para transformar o *token* desconhecido em seu correspondente mais próximo em um dicionário de palavras corretas ortograficamente. Na prática, o reconhecimento de entidades relevantes é viabilizado pela aplicação de expressões regulares, as quais possibilitam a identificação de padrões específicos no texto de intenção. Nesse processo, desconsideram-se estruturas gramaticais e ortográficas irrelevantes para o entendimento da intenção, tais como artigos, pronomes, pontuações e alguns caracteres especiais. A interpretação de sinônimos é essencialmente pautada na consulta à tabela de buscas, uma estrutura de dados contendo palavras relacionadas à palavra alvo dentro de um contexto semântico ou técnico. Para o sistema proposto, a tabela de buscas é populada correlacionando diversos sinônimos possíveis dos termos relacionados ao cenário de redes de computadores, em especial Open RAN. A principal vantagem consiste na diminuição da pluralidade de palavras utilizadas na composição da intenção processada, reduzindo a complexidade de processos subsequentes. Mesmo de-

⁴O treinamento foi realizado adotando 100 épocas e utilizando um conjunto de dados de intenções personalizado.

vidamente padronizada pelos procedimentos prévios, cada intenção não é passível de ser operada matematicamente, visto que ainda é composta pelos fragmentos textuais mais informativos e não por valores mensuráveis. Assim, a ingestão da intenção no classificador depende da vetorização prévia da intenção, realizada pela aplicação do modelo vetorial de Saco-de-Palavras (*Bag-of-Words* – BoW). No Rasa Core, a abordagem de rede neural é desempenhada pelo modelo TED (*Transformer Embedding Dialogue*), capaz de decidir a próxima ação do *chatbot* no diálogo com base nas interações passadas. Após o processamento e classificação, o conjunto de informações de cada intenção é enviado ao construtor da sintaxe NILE, cuja função é representar a intenção em uma estrutura padronizada dos parâmetros como ações, entidades, grandezas e unidades de medida.

Embarcado no controlador ONOS, o módulo **iApp** utiliza o subsistema de gerenciamento de aplicações do ONOS, desenvolvido sobre o Apache Karaf. Esse serviço usa as capacidades de orquestração de pacotes OGSi (*Open Grid Services Infrastructure*) para ativar e desativar dinamicamente as aplicações. A compilação é feita com o Apache Maven, gerando um artefato OAR (*ONOS Application Archive*) contendo todos os recursos e pacotes necessários. A implantação do iApp em todos os módulos do ONOS é realizada por um *shell script* (onos-app) que utiliza a API REST do ONOS. O módulo iApp engloba dois submódulos complementares, o Tradutor e o Resolvedor de Conflitos.

O submódulo **Tradutor** transforma a intenção pré-processada pelo assistente conversacional em uma expressão declarativa de configuração contendo parâmetros de nível mais baixo, que alimentam o *ONOS Intent Framework*. Para isso, o submódulo Tradutor substitui parâmetros incompreensíveis na linguagem de comandos do ONOS por aqueles interpretáveis, consultando uma tabela de sintaxe de comandos do ONOS e tabelas de encaminhamento, caso necessário. O nível de abstração exigido pelo *ONOS Intent Framework* é exemplificado ao analisar a intenção pré-processada expressa por “[*Conectar*] [*Máquina A*] [*Máquina B*]”. Este comando instrui a interconexão entre dois *hosts* na rede, adotando uma linguagem intermediária que pode criar uma barreira interpretativa para o ONOS. A implementação dessa ação requer a redefinição da intenção para *add-host-intent 00:00:00:00:00:01/None 00:00:00:00:00:04/None*, em que são definidos o comando correspondente no ONOS, o MAC e a porta de cada dispositivo envolvido.

O submódulo **Resolvedor de Conflitos** é responsável por detectar e mitigar conflitos entre as intenções inseridas pelo operador e as políticas em execução. É importante destacar que a resolução de conflitos é precedida pela identificação do conflito, cujo objetivo é avaliar se as intenções propostas podem ser aceitas ou devem ser rejeitadas devido a conflitos com políticas já implementadas na rede. Um fator crítico no processo de identificação de conflitos do sistema proposto é identificar corretamente o domínio afetado pela proposta de intenção. A identificação de conflitos é feita através de algoritmos de aprendizado profundo para classificação [de Oliveira et al., 2023].

As ações mapeadas pelo iApp são direcionadas ao *ONOS Intent Framework*, o subsistema do ONOS que viabiliza o IBM. O administrador da rede define o objetivo desejado, sem precisar especificar os métodos e caminhos para sua execução. As intenções pré-existentes no ONOS abrangem encaminhamento e qualidade de serviço. Dentre as intenções que focam o encaminhamento estão a *Host-to-Host* (H2H), *Multi-to-Single-Point* (M2SP) e *Single-to-Multi-Point* (S2MP). O *ONOS Intent Framework* também disponibiliza intenções voltadas para a qualidade de serviço, como a *Optical Intent* e a *Pro-*

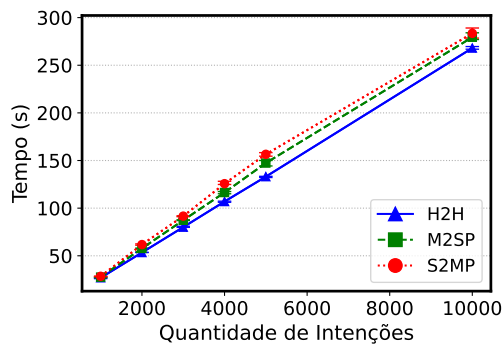
tected Transport. Este artigo concentra-se no primeiro conjunto de intenções. O *ONOS Intent Framework*, além de permitir a criação de novas intenções para ampliar as possibilidades de orquestração da rede, também possibilita a inserção de intenções via diferentes interfaces, tanto pela API RESTful quanto pela própria interface de comandos do ONOS.

Os *switches* P4, embora não sejam componentes integrantes do sistema proposto, podem fornecer informações de telemetria ao sistema por meio da Telemetria de Rede Dentro da Banda (*In-band Network Telemetry – INT*). Essas informações são enviadas para um servidor de telemetria P4 (Servidor INT P4), que encaminha relatórios periódicos de medição para a *iApp*, permitindo o monitoramento contínuo do desempenho da rede. Essa comunicação contínua permite validar se os objetivos da intenção estão sendo alcançados e a adaptar as políticas, se necessário.

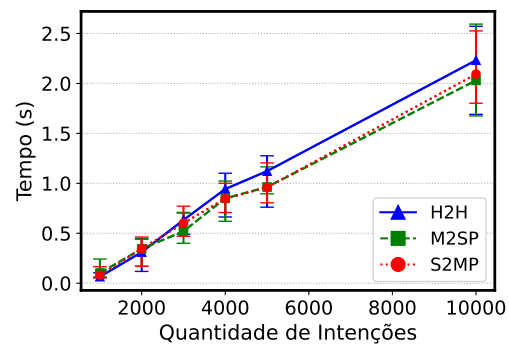
5. Discussão e Análise dos Resultados

O sistema proposto é avaliado em um ambiente emulado com o Mininet versão 2.3.1b4. Os experimentos foram realizados em uma máquina virtual com Ubuntu 22.04 LTS, equipado com um processador QEMU Virtual versão 2.5+ 3,2 Ghz, com 4 GB de RAM e 40 GB de armazenamento. A topologia de teste é uma topologia em árvore, com dois níveis, sendo o primeiro composto por um comutador raiz e o segundo por oito comutadores folha. A cada comutador folha estão conectadas oito estações finais, totalizando 64 estações. Os experimentos verificam o tempo decorrido até a instalação das intenções focadas em encaminhamento, H2H, M2SP e S2MP. O processo para instalação das intenções se inicia com a interação do usuário com o *chatbot*, que formata a entrada em intenções pré-processadas, encaminhadas ao *iApp* no ONOS. A avaliação desconsidera o tempo de interação do usuário com o *chatbot*, contabilizando apenas o tempo a partir do envio da intenção pré-processada para o *iApp*, até a finalização da instalação da intenção no controlador. Após gerar a intenção pré-processada, o *chatbot* pode interagir com o controlador ONOS através da API REST ou diretamente com a CLI do ONOS. A contabilização do tempo ao utilizar a CLI para interagir com o controlador requer modificações nas classes de cada intenção para adicionar marcação de tempo. Ao utilizar a API REST, o tempo das requisições ao ONOS é medido a partir de um código em Python. Três cenários são avaliados. No primeiro, verifica-se o tempo decorrido para instalação das intenções à medida que o número de intenções aumenta. No segundo cenário, a mesma avaliação é realizada em presença de um tráfego de rede crescente. O terceiro cenário mostra o impacto da ativação de uma intenção em fluxos de rede durante o encaminhamento. No primeiro e segundo cenários, os resultados são apresentados com intervalos de confiança com nível de confiança de 95%.

No primeiro cenário, varia-se a quantidade de intenções entre 1.000 e 5.000, com passo de 1.000 intenções, e 10.000 intenções. A Figura 2(a) mostra os resultados obtidos ao utilizar a API REST, enquanto a Figura 2(b) apresenta os resultados para a CLI. A emulação é repetida 10 vezes para cada caso apresentado, a fim de permitir uma comparação de tempo com significância estatística. Observa-se nas figuras que, independentemente do tipo de intenção avaliados, há um aumento no tempo com o crescimento da quantidade de intenções. A tendência de aumento no tempo tende à linearidade. Ao utilizar a API REST, o tempo decorrido para instalação das intenções é mais elevado do que ao utilizar a CLI. Addad *et al.* também reportam esse comportamento [Addad et al., 2018], que sugere que a interface RESTful possui um custo computacional maior. O crescimento

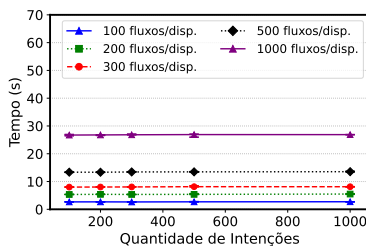


(a) API REST.

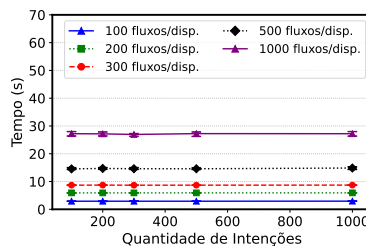


(b) CLI.

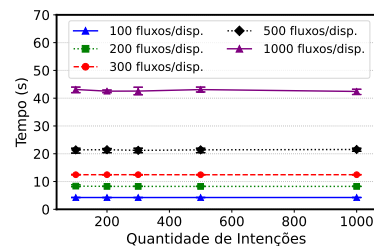
Figura 2. Tempo decorrido para instalação das intenções ao interagir com o controlador ONOS através de (a) API REST e (b) CLI. Há uma tendência de crescimento linear com o aumento da quantidade de intenções e o tempo para instalação das intenções pela CLI é muito menor do que pela API REST. Não há diferença de tempo para instalação de intenções de tipos distintos ao se utilizar a CLI. A API REST apresenta tempo menor para intenções H2H.



(a) H2H – API REST.



(b) M2SP – API REST.



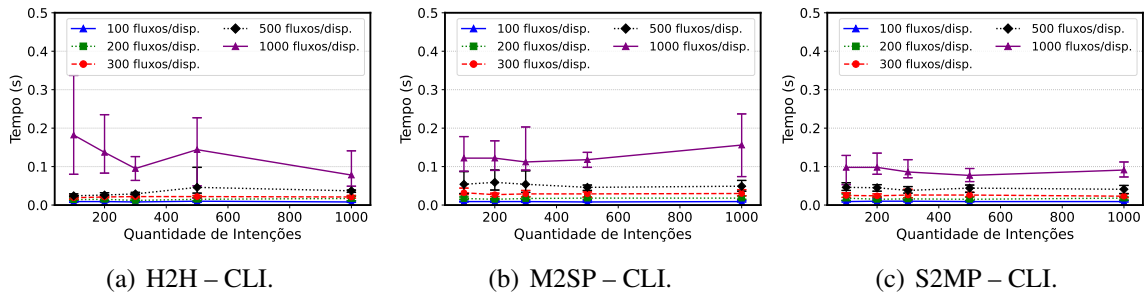
(c) S2MP – API REST.

Figura 3. Tempo decorrido para instalação das intenções ao variar o tráfego na rede. Ao utilizar a API REST, não há influência significativa do aumento da quantidade de intenções, mas o crescimento do fluxo interfere negativamente no tempo necessário para instalação das intenções.

próximo ao linear em todos os casos mostra que o tempo para instalar uma intenção individual não é afetado pela quantidade de intenções. Considerando uma mesma quantidade de intenções, não existem diferenças significativas no tempo para instalação quando se utiliza a CLI. Ao utilizar a API REST, observa-se que as intenções H2H são instaladas mais rapidamente do que as demais.

No segundo cenário de testes, com o objetivo de avaliar a eficácia das intenções com a aplicação de outras atividades na rede, são executados experimentos com a presença de fluxos de tráfego, variando entre 100 e 1.000 fluxos por dispositivo, com passo de 100 fluxos por dispositivo. Os experimentos são executados seis vezes para cada combinação de quantidade de intenções e quantidade de fluxos por dispositivo. As Figuras 3 e 4 mostram os resultados obtidos para cada tipo de intenção avaliado, utilizando, respectivamente, a API REST e a CLI ONOS. Considerando uma quantidade de fluxos por dispositivo constante, observa-se nas Figuras 3(a), 3(b) e 3(c) que o aumento na quantidade de intenções, de 100 até 1.000, não influencia no tempo decorrido para instalação das intenções pelo controlador.

Verifica-se que o crescimento do tráfego aumenta o tempo necessário para a



(a) H2H – CLI. (b) M2SP – CLI. (c) S2MP – CLI.

Figura 4. Tempo decorrido para instalação da intenções ao variar o tráfego na rede. Ao utilizar a CLI, não há influência significativa do aumento da quantidade de intenções e não há alteração significativa no tempo para tráfego abaixo de 300 fluxos por dispositivo. Acima disso, há variabilidade no tempo de instalação em decorrência da ocupação da rede.

instalação das intenções, sendo S2MP o tipo de intenção mais afetado pelo tráfego. O uso da CLI reduz o tempo necessário para instalação das intenções, independentemente do tipo, como mostram as Figuras 4(a), 4(b) e 4(c). Ao aumentar a quantidade de intenções, mantendo a quantidade de fluxos por dispositivo constante, o tempo necessário para instalação das intenções se mantém praticamente constante para uma quantidade e intenções menor do que 1.000, exceto para os casos em que o tráfego é maior do que 500 fluxos por dispositivo. O crescimento do fluxo aumenta a variabilidade de tempo para instalação das intenções, mesmo quando a quantidade de intenções é pequena, por exemplo 100 intenções. Portanto, mudanças no tráfego da rede podem afetar o desempenho de um sistema baseado em intenção.

No terceiro cenário, o objetivo é visualizar o efeito da aplicação de uma intenção *Host-to-Host* (H2H) que altera o caminho dos fluxos entre duas estações na rede. Para esse experimento, o cenário emulado considera uma topologia em anel com quatro comutadores e as estações finais separadas por dois caminhos equidistantes, cada um de tamanho dois. No entanto, um dos caminhos entre as estações está limitado a uma taxa de transmissão de 20 Mb/s, enquanto o outro está limitado a 40 Mb/s. O experimento inicia no tempo $t_0 = 0$ s, com três fluxos compartilhando o caminho de 20 Mb/s. Em $t_1 = 30$ s,

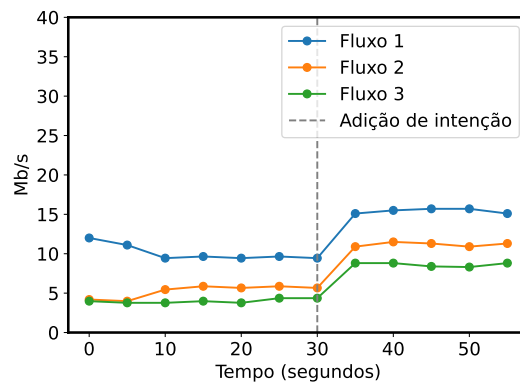


Figura 5. A adição de uma intenção na rede em 30 s resulta no aumento da banda disponível entre as duas estações comunicantes. O experimento considera uma topologia de comutadores em anel e a intenção adicionada é do tipo *Host-to-Host* (H2H).

é adicionada a intenção H2H que altera o caminho de comunicação entre as estações, passando para o novo caminho limitado a 40 Mb/s. A Figura 5 revela que o efeito da intenção é aplicado corretamente na rede, resultando no aumento da banda disponível para os três fluxos considerados.

6. Conclusão

Este artigo propôs um sistema de gerenciamento de redes baseado em intenção do usuário expressas em um assistente computacional de diálogo (*chatbot*), projetado para lidar com as complexidades das Redes de Acesso via Rádio Abertas (Open RAN). A arquitetura do sistema proposto permite que o operador de rede utilize linguagem natural para especificar os objetivos de nível de serviço, eliminando a necessidade de conhecer detalhes de implementação de interfaces dos dispositivos, de protocolos e de topologias para configurar a rede. A partir do pré-processamento da entrada do operador, o sistema gera intenções de baixo nível, que são traduzidas em políticas implementadas pelo controlador ONOS. O desempenho do sistema é avaliado em um ambiente emulado, verificando a latência para a instalação das intenções na rede ao variar a quantidade de intenções a serem instaladas e o volume de tráfego na rede. Os resultados demonstram que a proposta executa corretamente a tradução de intenções de alto nível para fluxos de rede. Além disso, os resultados evidenciam que a latência de instalação das intenções é baixa e que seu crescimento é linear com o número de intenções instaladas. Por fim, também é observado que a interação com o *ONOS Intent Framework* é facilitada por meio da interface de linha de comando em detrimento da API RESTful. Vislumbram-se como trabalhos futuros a implementação do submódulo resolvidor de conflitos, permitindo a avaliação da latência em presença de intenções conflitantes na rede, e o aprimoramento da correção ortográfica do *chatbot* utilizando a API OpenAI.

Referências

- Addad, R. A., Dutra, D. L. C., Bagaa, M., Taleb, T., Flinck, H. e Namane, M. (2018). Benchmarking the ONOS intent interfaces to ease 5G service management. Em *2018 IEEE Global Communications Conference (GLOBECOM)*, p. 1–6.
- Bonati, L., Polese, M., D’Oro, S., Basagni, S. e Melodia, T. (2023). OpenRAN Gym: AI/ML development, data collection, and testing for O-RAN on PAWR platforms. *Computer Networks*, 220:109502.
- Cesila, C. H., Pinto, R. P., Mayer, K. S., Escallón-Portilla, A. F., Mello, D. A., Arantes, D. S. e Rothenberg, C. E. (2023). Chat-IBN-Rasa: Building an intent translator for packet-optical networks based on Rasa. Em *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*, p. 534–538. IEEE.
- Couto, R. S., Mattos, D. M. F., Moraes, I. M., Cruz, P., Medeiros, D. S. V., Souza, L. A. C., Táparo, F. G., Campista, M. E. M. e Costa, L. H. M. K. (2023). Gerenciamento e orquestração de serviços em O-RAN: Inteligência, tendências e desafios. Em *Minicursos do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*.
- De Alwis, C., Kumar, P., Pham, Q.-V., Dev, K., Kalla, A., Liyanage, M. e Hwang, W.-J. (2023). Towards 6G: Key technological directions. *ICT Express*, 9(4):525–533.

- de Oliveira, N. R., Moraes, I. M., de Medeiros, D. S. V., Lopez, M. A. e Mattos, D. M. (2023). An agile conflict-solving framework for intent-based management of service level agreement. Em *2023 2nd International Conference on 6G Networking (6GNet)*, p. 1–8. IEEE.
- Jacobs, A. S., Pfitscher, R. J., Ferreira, R. A. e Granville, L. Z. (2018). Refining network intents for self-driving networks. Em *Proceedings of the Afternoon Workshop on Self-Driving Networks*, p. 15–21.
- Jacobs, A. S., Pfitscher, R. J., Ribeiro, R. H., Ferreira, R. A., Granville, L. Z., Willinger, W. e Rao, S. G. (2021). Hey, Lumi! Using natural language for {intent-based} network management. Em *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, p. 625–639.
- Kavehmadavani, F., Nguyen, V.-D., Vu, T. X. e Chatzinotas, S. (2023). Intelligent traffic steering in beyond 5G Open RAN based on LSTM traffic prediction. *IEEE Transactions on Wireless Communications*, 22(11):7727–7742.
- Kiran, M., Pouyoul, E., Mercian, A., Tierney, B., Guok, C. e Monga, I. (2018). Enabling intent to configure scientific networks for high performance demands. *Future Generation Computer Systems*, 79:205–214.
- Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B. S., Singh, T., Zugno, T., Cuomo, F. e Melodia, T. (2024). Programmable and customized intelligence for traffic steering in 5G networks using Open RAN architectures. *IEEE Transactions on Mobile Computing*, 23(4):2882–2897.
- Mahtout, H., Kiran, M., Mercian, A. e Mohammed, B. (2020). Using machine learning for intent-based provisioning in high-speed science networks. Em *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, SNTA'20, p. 27–30, New York, NY, USA. Association for Computing Machinery.
- Martini, B., Gharbaoui, M. e Castoldi, P. (2023). Intent-based network slicing for SDN vertical services with assurance: Context, design and preliminary experiments. *Future Generation Computer Systems*, 142:101–116.
- Nguyen, M.-T.-A., Souihi, S. B., Tran, H.-A. e Souihi, S. (2022). When NLP meets SDN : an application to Global Internet eXchange network. Em *ICC 2022 - IEEE International Conference on Communications*, p. 2972–2977.
- Rahman, A., Al Mamun, A. e Islam, A. (2017). Programming challenges of chatbot: Current and future prospective. Em *2017 IEEE region 10 humanitarian technology conference (R10-HTC)*, p. 75–78. IEEE.
- Riftadi, M. e Kuipers, F. (2019). P4I/O: Intent-based networking with P4. Em *2019 IEEE Conference on Network Softwarization (NetSoft)*, p. 438–443.
- Wei, Y., Peng, M. e Liu, Y. (2020). Intent-based networks for 6G: Insights and challenges. *Digital Communications and Networks*, 6(3):270–280.
- Zheng, X., Leivadreas, A. e Falkner, M. (2022). Intent based networking management with conflict detection and policy resolution in an enterprise network. *Computer Networks*, 219:109457.