

Desafios de Escalabilidade e Resiliência em SDN sob Mobilidade: Uma Análise Exploratória do OpenFlow

MD Ashik Rezwani¹, Edgar S. Oliveira², Paulo H. L. Rettore^{3,7},
Clayson Celes^{4,5}, Bruno Santos⁶, Jose Marcos Nogueira³, Rodolfo Meneguette²

¹Universidade de Bonn, Bonn - Alemanha

²Universidade de São Paulo (USP), São Carlos - Brasil

³Universidade Federal de Minas Gerais (UFMG), Belo Horizonte - Brasil

⁴Universidade Federal do Espírito Santo (UFES), Vitória - Brasil

⁵Instituto Federal do Estado do Ceará (IFCE), Fortaleza - Brasil

⁶Universidade Federal da Bahia (UFBA), Salvador - Brasil

⁷Universidade Federal de Lavras (UFLA), Lavras - Brasil

s6mdrezw@uni-bonn.de, edgar.oliveira.icmc@usp.br,
rettore@dcc.ufmg.br, clayson.celes@ufes.br,
bruno.ps@ufba.br, jmarcos@dcc.ufmg.br, meneguette@icmc.usp.br

Abstract. *As Software-Defined Networking (SDN) evolves, supporting highly mobile environments has become a significant challenge. This paper evaluates the performance of three widely used SDN controllers—ONOS, RYU, and OpenFlow Reference—under four mobility models: Manhattan Grid, Gauss-Markov, Random Direction, and Random Waypoint. The main objective is to investigate, through an exploratory study not addressed in the current literature, the limitations of OpenFlow-based SDN architectures in mobile scenarios such as vehicular networks, aerial networks, and mobile IoT systems, and to identify potential directions for protocol adoption with appropriate adaptations. The results show that although SDN architectures scale with network size, reactive flow management strategies suffer severe performance degradation in highly mobile environments. A significant behavioral shift was observed in the RYU controller: while it underperformed in small- and medium-scale scenarios, it became more robust and scalable in large-scale, high-load networks when operating proactively. In contrast, ONOS and OpenFlow exhibited greater stability in smaller deployments but reached performance limits under more demanding conditions. These findings highlight the need for proactive flow management and protocol-level adaptations to ensure reliable SDN operation in high-mobility environments.*

Resumo. *Com a evolução das Redes Definidas por Software (SDN), o suporte a ambientes com alta mobilidade tornou-se um desafio significativo. Este trabalho avalia o desempenho de três controladores SDN amplamente utilizados — ONOS, RYU e implementação de referência do OpenFlow — quando combinados com quatro modelos distintos de mobilidade: Manhattan Grid, Gauss-Markov, Random Direction e Random Waypoint. O objetivo é investigar, por meio de um estudo exploratório, as limitações do OpenFlow e das estratégias*

de gerenciamento de fluxos em cenários móveis, como redes veiculares, aéreas e IoT móvel, bem como identificar possíveis caminhos para sua adoção mediante adaptações arquiteturais e operacionais. Os resultados indicam que, embora as arquiteturas SDN apresentem boa escalabilidade, as configurações reativas de gerenciamento de fluxos apresentam perdas de desempenho expressivas em ambientes altamente móveis. Em contrapartida, ONOS e OpenFlow Reference demonstraram maior estabilidade em cenários menores, mas atingiram limites de desempenho em condições mais severas. Os resultados reforçam a importância de estratégias proativas e de adaptações no protocolo para garantir a confiabilidade das redes SDN em cenários de alta mobilidade.

1. Introdução

As Redes Definidas por Software (Software Defined Networking (SDN)) consolidaram-se como um paradigma transformador na arquitetura de redes modernas, proporcionando a flexibilidade necessária por meio do desacoplamento entre os planos de controle e de dados [Chaurasia et al. 2019]. Embora a adoção de protocolos como o OpenFlow tenha atingido maturidade em infraestruturas cabeadas e *datacenters*, a expansão desses princípios para ambientes móveis, resultando em *Software-Defined Wireless Networks* (SDWN), visa atender à demanda crescente por conectividade ubíqua em ecossistemas de IoT e dispositivos móveis [Sornlertlamvanich et al. 2016, Khan et al. 2018].

Contudo, a aplicação de SDN em cenários de mobilidade impõe desafios severos que não existem em redes estáticas. A dinamicidade inerente aos nós móveis acarreta mudanças frequentes de topologia e instabilidade de enlaces, o que degrada a eficiência da arquitetura clássica centralizada [Pentikousis et al. 2013]. As implementações atuais, otimizadas para infraestruturas fixas, frequentemente carecem de mecanismos robustos para o gerenciamento de *handover* e de reconfiguração de rotas em tempo real, resultando em latência elevada e sobrecarga excessiva no plano de controle [Tantayakul et al. 2016].

Neste contexto, este trabalho investiga o impacto da mobilidade no desempenho de protocolos SDN, identificando as limitações críticas das implementações vigentes. A pesquisa avalia métricas fundamentais de Quality of Software (QoS), incluindo o tempo de *handover*, a perda de pacotes e a latência, sob diferentes padrões de mobilidade. Além disso, o artigo avalia as instalações dos fluxos nas tabelas para identificar a sensibilidade dos controladores à detecção da mobilidade, que impacta diretamente as métricas de rede, e propõe direções futuras para o desenvolvimento de protocolos e controladores mais resilientes e adaptativos em cenários dinâmicos.

As principais contribuições deste trabalho são as seguintes:

- Uma avaliação experimental sistemática do desempenho de redes SDN sob cenários de mobilidade, considerando diferentes escalas de rede, modelos de mobilidade e estratégias de gerenciamento de fluxos, utilizando um ambiente emulado reproduzível baseado no Mininet-WiFi;
- Uma análise quantitativa do impacto das estratégias reativa e proativa de gerenciamento de fluxos no tempo de convergência durante eventos de *handover*, evidenciando diferenças significativas de desempenho sob diferentes padrões de mobilidade;

- Uma comparação experimental entre diferentes arquiteturas de controle SDN amplamente utilizadas, incluindo uma implementação de referência do protocolo OpenFlow, o controlador Ryu e o controlador ONOS, destacando seus comportamentos sob diferentes estratégias de instalação de fluxos e escalas de rede;
- A discussão de implicações práticas dos resultados obtidos e a indicação de direções para a evolução do plano de controle SDN, apontando para a adoção de abordagens híbridas que combinem controle centralizado com capacidades de resposta local em cenários de mobilidade.

2. Trabalhos Relacionados

A literatura recente sobre gestão de mobilidade em redes definidas por software tem se diversificado em múltiplas frentes, visando mitigar problemas de *Handover Efficiency (HE)*, *Network Scalability (NE)* e *Quality of Experience (QoE)*. As investigações podem ser amplamente categorizadas em arquiteturas de controle (centralizadas vs. distribuídas) e estratégias de otimização de *handover* (proativas, reativas e híbridas). A Tabela 1 sumariza as principais propostas analisadas, destacando seus cenários de validação e respectivas limitações.

2.1. Arquiteturas de Controle e Aplicações em Redes Móveis

Estudos iniciais focaram na substituição de protocolos tradicionais por uma lógica centralizada SDN para reduzir a latência de sinalização. Contudo, a centralização impõe severos gargalos de escalabilidade. Avaliações experimentais recentes comparando controladores modernos (como Ryu e OpenDaylight) confirmam que, embora arquiteturas leves reduzam a latência em cenários simples, o processamento centralizado degrada rapidamente sob alta carga de requisições [Ahmed et al. 2026].

Para endereçar isso, abordagens distribuídas e hierárquicas têm sido exploradas para redes 5G, integrando SDN ao *Distributed Mobility Management (DMM)*. Entretanto, trabalhos focados em *Intelligent transportation system (ITS)* indicam que, apesar de reduzirem a perda de pacotes, essas arquiteturas distribuídas introduzem um custo elevado de sinalização e latência fim-a-fim quando comparadas a protocolos de mobilidade tradicionais [Das et al. 2024]. A literatura frequentemente omite a análise do comportamento de componentes padronizados (como o plano de controle OpenFlow puro) sob esse ambiente dinâmico, focando em protocolos proprietários. Revisões sobre o estado da arte em 5G e 6G reforçam que a gestão de *handover* reativa ainda falha em garantir QoS em cenários de alta mobilidade [Saoud et al. 2025].

Latency Reduction (LR); Handover Efficiency (HE); Scalability (SC); Performance Optimization (PO); Reliability (RE); Reduced Signaling Overhead (RSO); Scalability with larger network (LSD); Quality of Experience (QoE)

Artigos	Área de Foco									Aspecto da Rede			
	LR	HE	SC	PO	RE	RSO	LSD	QoE	Topologia	Ferramenta	Amb.	Controle	Switch
Gerenciamento de mobilidade baseado em SDN													
[Das et al. 2024]	X	✓	✓	✓	✓	X	✓	X	Vehicular	NS-3	Simulação	SDN-DMM	-
[Bello et al. 2020]	✓	✓	X	✓	✓	✓	X	X	Star/Mesh	Cooja/Contiki	Emulação	OpenFlow Reference	-
Soluções SDN híbridas													
[Silva et al. 2021]	✓	✓	X	✓	X	✓	X	✓	Hybrid	-	Simulação	RYU	-
Otimização de Handover Proativa vs. Reativa													
[Lima et al. 2023]	✓	✓	✓	✓	✓	✓	✓	✓	Hexagonal	Python/AI	Simulação	AI/Proactive	-
[Marquezan et al. 2016]	✓	✓	✓	✓	✓	X	✓	X	Hybrid	Mininet	Emulação	Floodlight	19/89/178
[Labraoui et al. 2017]	X	X	X	✓	✓	X	X	X	Mesh	CORE	Emulação	4,NEON	8
[Tong et al. 2021]	✓	✓	✓	✓	✓	X	X	✓	-	Mininet-WiFi	Emulação	OpenFlow Reference	-
SDN para redes de próxima geração													
[Tantayakul et al. 2019]	✓	✓	✓	✓	✓	X	X	X	-	NS3,SUMO	Simulação	OpenFlow Reference	-

Tabela 1. Revisão da literatura atualizada.

2.2. Estratégias de Otimização de *Handover* e Soluções Híbridas

Para lidar com interrupções de serviço durante a mobilidade, a literatura aborda o dilema entre proatividade e reatividade. Estratégias modernas, baseadas em *Deep Learning* e inteligência computacional, buscam antecipar o *handover* com alta precisão para mitigar a perda de pacotes [Lima et al. 2023, Tong et al. 2021, Labraoui et al. 2017]. Embora eficazes, muitas dessas soluções dependem de lógicas complexas de predição (como redes neurais recorrentes), o que adiciona uma camada de processamento externa aos controladores de mercado.

Paralelamente, surgem arquiteturas híbridas como soluções de transição, permitindo a coexistência de SDN com sistemas legados (LTE/EPC) [Silva et al. 2021]. Embora pragmaticamente valiosas para implantação incremental, essas abordagens tendem a mascarar as restrições inerentes aos controladores SDN puros, pois abstraem ou removem a lógica OpenFlow crítica do caminho de decisão de mobilidade.

2.3. Análise Crítica e Lacunas em Aberto

Apesar dos avanços, observa-se uma lacuna significativa na avaliação de implementações padrão de SDN em cenários realistas. A maioria dos trabalhos revisados baseia-se em simulações simplificadas, em protocolos modificados (não-OpenFlow (OF)) ou em arquiteturas híbridas que evitam o enfrentamento direto das limitações do protocolo OpenFlow em ambientes altamente dinâmicos. Questões críticas sobre LSD, tempo de convergência de topologia e os limites operacionais de controladores *mainstream* permanecem subexploradas. Este trabalho visa preencher essa lacuna, com foco na análise de desempenho de infraestruturas puramente SDN sob condições de mobilidade.

3. Metodologia

Esta seção apresenta os detalhes da metodologia proposta para a avaliação dos desafios de escalabilidade e resiliência em Redes Definidas por Software em cenários dinâmicos. Metodologia adotada prevê a avaliação através de emulação, utilizando o *Mininet-WiFi* (versão 2.6), o que possibilitou a reprodução, de forma controlada e reprodutível, de

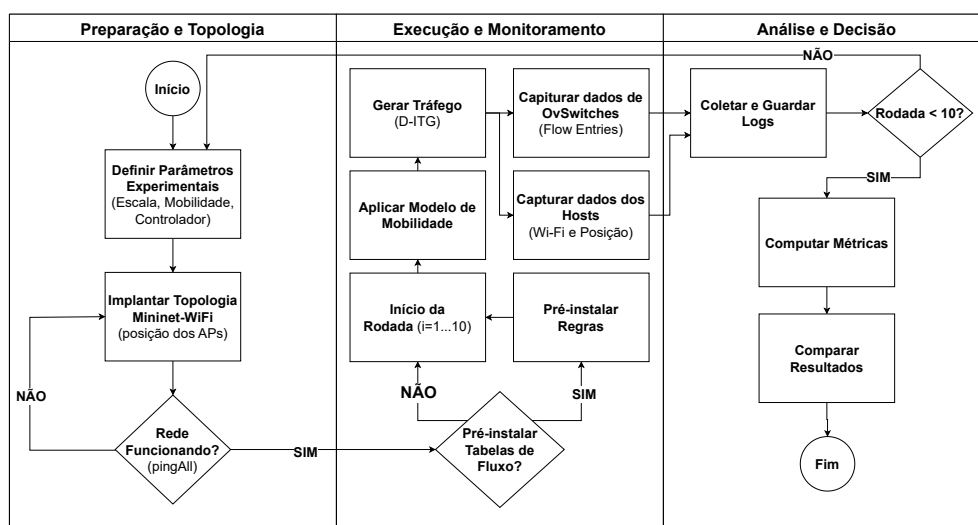


Figura 1. Fluxo de trabalho para a avaliação de desempenho do OpenFlow.

Escala	ID	Configuração	AP gap	Ferramenta de Tráfego	γ (Perda)	Mobilidade	Rodadas
Pequena	Exp A	4 APs, 4 estações	100 m	D-ITG	3.7	Todas móveis	10
Média	Exp B	16 APs, 32 estações	65 m	D-ITG, Iperf	3.9	Todas móveis	10
Grande	Exp C	16 APs, 64 estações	65 m	D-ITG, Iperf	3.9	Todas móveis	10

Tabela 2. Escala e configuração da topologia.

cenários de mobilidade e de propagação de sinal sem fio. O fluxo de trabalho experimental, ilustrado na Figura 1, segue uma abordagem de ciclo fechado composta por três estágios principais: (i) Preparação e Topologia, (ii) Execução e Monitoramento, e (iii) Análise e Decisão. Essa estrutura visa garantir a consistência experimental e o isolamento adequado entre os diferentes cenários avaliados.

Para investigar o impacto da mobilidade e da escalabilidade no desempenho da rede, adotou-se um design fatorial completo, no qual os fatores experimentais exploram três dimensões fundamentais: a escala da rede, a dinâmica de mobilidade e a arquitetura de controle SDN. Adicionalmente, todos os cenários foram avaliados sob duas estratégias de gerenciamento de fluxos — *reativa* (instalação sob demanda) e *proativa* (regras pré-instaladas) — permitindo analisar a sensibilidade de cada arquitetura de controle às variações impostas pela mobilidade. A seguir, cada estágio do fluxo de trabalho proposto é detalhado.

3.1. Preparação e Topologia

Na fase de preparação, o ambiente experimental é instanciado em uma das três configurações de escala detalhadas na Tabela 2. Os cenários diferem no número de estações móveis (STAs) e de Pontos de Acesso (APs), permitindo avaliar o impacto da densidade da rede no desempenho do plano de controle. Os APs são dispostos estaticamente em formato de grade (*grid*), com espaçamento equidistante definido pelo parâmetro *AP gap*, enquanto as estações móveis são organizadas em pares emissor–receptor para a injeção controlada de tráfego. Nos cenários de média e grande escala, o número de pontos de acesso foi mantido constante, variando apenas a densidade de estações móveis por AP, de modo a isolar o impacto da escalabilidade no plano de controle. Além disso, o parâmetro γ representa o expoente de perda do modelo de propagação.

Para analisar o impacto da mobilidade no desempenho do protocolo e das arquiteturas de controle SDN, foram considerados diferentes padrões de mobilidade, conforme apresentado na Tabela 3. Foram selecionados quatro modelos de mobilidade para cobrir distintos comportamentos de deslocamento e os desafios associados ao processo de *handover*. Os modelos *Random Waypoint (RWP)* e *Random Direction (RD)* representam movimentos estocásticos genéricos, servindo como base de comparação para entropia de movimento; *Manhattan Grid (MG)* simula ambientes urbanos estruturados (ruas e avenidas), impondo restrições geográficas realistas ao deslocamento das estações; e *Gauss-Markov (GM)* é utilizado para simular movimentos mais fluidos e temporalmente dependentes, reduzindo as mudanças abruptas de direção típicas de modelos puramente aleatórios.

O desempenho da rede em cenários móveis depende intrinsecamente da capacidade do controlador, no plano de controle, de processar eventos e de reconfigurar as entradas na tabela de fluxos. Dessa forma, para a avaliação exploratória deste trabalho, foram consideradas três arquiteturas de controle SDN distintas, conforme ilustrado na Tabela 4. A implementação de referência do protocolo *OpenFlow* foi utilizada como *baseline*, per-

Modelo de Mobilidade	Ambiente Real Representado	Aplicações Típicas
Manhattan Grid [Mobile 1998]	Ambientes urbanos estruturados	Gestão de tráfego urbano, redes veiculares
Random Waypoint [Johnson and Maltz 1996]	Espaços públicos abertos (campus, parques)	Mobilidade de pedestres e usuários móveis
Random Direction [Bujari et al. 2017]	Patrulha de Fronteira ou de Área	Missões de patrulhamento, busca e salvamento, robôs autônomos
Gauss-Markov [Liang and Haas 2003]	Rodovias e ambientes abertos	Veículos de alta velocidade e drones

Tabela 3. Mapeamento de modelos de mobilidade para ambientes reais.

Controlador	Linguagem	Perfil de Uso Experimental
OpenFlow Reference	C / Python	Implementação de referência utilizada para validar o comportamento padrão do protocolo e estabelecer uma linha de base para a latência de handover, sem otimizações avançadas de controle.
Ryu	Python	Framework SDN modular e extensível adequado para prototipagem rápida e avaliação de estratégias reativas e proativas de gerenciamento de fluxos.
ONOS	Java	Arquitetura distribuída de classe operadora. Focado em alta disponibilidade e escalabilidade para cenários de alta densidade de dispositivos.

Tabela 4. Características dos controladores SDN utilizados nos experimentos.

mitindo validar o comportamento padrão do protocolo, sem a incorporação de otimizações típicas de ambientes de produção. O controlador *RYU*, desenvolvido em Python, foi selecionado devido à sua modularidade e à ampla adoção em pesquisas acadêmicas, o que representa arquiteturas baseadas em componentes. Por fim, o controlador *ONOS* é uma solução *carrier-grade*, implementada em Java, com núcleo distribuído, cuja inclusão permite analisar o *trade-off* entre a robustez de arquiteturas mais complexas e o tempo de convergência em topologias altamente dinâmicas. A seleção dessas arquiteturas permite comparar desde uma implementação de referência do protocolo OpenFlow até controladores com diferentes níveis de complexidade e distribuição, fornecendo uma visão abrangente do impacto do plano de controle em cenários SDN sob mobilidade.

3.2. Procedimento de Execução e Coleta de Dados

Para garantir a confiabilidade estatística dos resultados, cada combinação de cenário foi executada em 10 rodadas independentes. O procedimento experimental, ilustrado na Figura 1, é totalmente automatizado e segue as etapas descritas a seguir:

1. **Inicialização:** Definição dos parâmetros vetoriais e orquestração da topologia no Mininet-WiFi. Estes parâmetros compreendem a escala do experimento, na qual definem-se as posições dos pontos de acesso a serem instanciados, bem como a configuração do expoente de perda de percurso (degradação do sinal) γ . Adicionalmente, instanciam-se as estações configuradas de acordo com o modelo de mobilidade escolhido.
2. **Verificação:** Execução de testes de conectividade (*pingAll*) para assegurar a integridade da malha de controle e de dados antes do início dos experimentos com a geração de fluxos de dados entre as entidades móveis.
3. **Geração de Tráfego:** Injeção de fluxos UDP contínuos com a ferramenta **D-ITG** e *Iperf*, configurada para capturar métricas de QoS.
4. **Coleta (Loop):** Durante a execução, são capturados dados de telemetria do plano de controle (*Flow Entries* e pacotes *Packet-In*) e métricas de rede no receptor (Atraso, Jitter e Vazão).

5. **Fechamento:** Ao final de cada iteração, os *logs* são consolidados e o ambiente é reiniciado para garantir o isolamento entre os experimentos. Os dados coletados são salvos em formato Comma Separated Values (CSV) após a aplicação de um *parser* nos logs do D-ITG, para que sejam analisados posteriormente em conjunto.

3.3. Métricas de Avaliação

Nesta seção, apresentamos as principais métricas utilizadas para avaliar a proposta e verificar o impacto da pré-instalação de tabelas de fluxo em cenários de mobilidade variados e diferentes implementações do protocolo OpenFlow.

3.3.1. Tempo de Convergência

A métrica de tempo de convergência (T_{conv}) foi definida como o intervalo temporal entre o estabelecimento da conexão nas camadas física e de enlace (Layer 1-2) e a efetiva instalação das regras de encaminhamento (Flow Rules) nos dispositivos de encaminhamento. Esse processo é crítico em cenários de mobilidade, pois determina o período de interrupção de serviço durante o *handover*. Ou seja, o tempo de conexão entre o dispositivo móvel e o ponto de acesso e o tempo que o OpenFlow leva para criar a rota por meio das entradas na tabela de fluxo e restabelecer a transmissão de dados do usuário.

O procedimento de medição foi realizado através da correlação de logs distribuídos, capturados em tempo de execução via scripts de monitoramento, conforme descrito a seguir:

1. **Detecção de Associação (L2):** Um agente de monitoramento nas estações móveis inspecionou continuamente o estado da interface sem fio com a ferramenta do sistema *iwconfig*. O instante de associação (T_{assoc}) foi registrado no momento exato em que o BSSID (Basic Service Set Identifier) reportado pela interface alterou-se do AP de origem para o AP de destino (Alvo).
2. **Detecção de Regras de Fluxo (SDN):** Ao mesmo tempo, o estado das tabelas de fluxo nos Pontos de Acesso foi auditado. O instante de atualização (T_{flow}) foi capturado assim que uma nova entrada de fluxo correspondente ao par (src, dst) da estação móvel foi identificada na tabela do AP-alvo.

Dessa forma, o tempo de convergência para cada evento de *handover* (k) foi calculado pela Equação 1:

$$T_{conv}(k) = T_{flow}(k) - T_{assoc}(k) \quad (1)$$

Onde os valores de $T_{conv} \approx 0$ indicam um comportamento proativo ideal (regras pré-instaladas), enquanto valores elevados denotam a latência introduzida pelo processamento reativo do controlador SDN.

3.3.2. Métricas de QoS

Para a avaliação de desempenho da rede sob mobilidade, foram utilizadas ferramentas de geração de tráfego, como o D-ITG (*Distributed Internet Traffic Generator*) e o *Iperf*.

A escolha destas ferramentas justifica-se pela capacidade de gerar fluxos de tráfego com controle granular sobre os parâmetros de transmissão e de gerar estatísticas de transmissão e de recepção. Note que a metodologia utilizou-se dessas ferramentas por critérios de melhor controle e gestão dos fluxos de dados do usuário, contudo, ferramentas similares de geração de tráfego configurável poderiam ser utilizadas para atingir os mesmos objetivos deste trabalho.

A coleta de dados foi estruturada para isolar o plano de dados (avaliado) do plano de controle do experimento. As estações móveis foram configuradas com duas interfaces de rede: uma interface sem fio (`wlan0`) para o tráfego de dados experimental e uma interface cabeada virtual (`eth1`) para a sinalização de controle do D-ITG. Isso garante a troca de pacotes de sinal de início e fim de fluxo, sem interromper a transmissão em caso de desconexões das estações móveis, assegurando que as métricas de QoS sejam coletadas e analisadas a posteriori.

O tráfego foi gerado por meio do protocolo de transporte UDP, configurado para emular fluxos contínuos ao longo de todo o período de simulação (300 segundos por rodada, com taxa de 1 pacote por segundo). Essa configuração visa enfatizar o impacto da mobilidade e dos eventos de *handover* no desempenho da rede, minimizando os efeitos de congestionamento no enlace. As métricas de desempenho foram obtidas a partir da decodificação dos logs gerados pelos nós receptores, com o auxílio de um parser desenvolvido para consolidar os resultados em arquivos CSV.

As métricas analisadas incluem: Vazão (*Throughput*), definida como a taxa média de bits recebidos com sucesso (kbps); Atraso (*Delay*), correspondente à latência unidirecional média entre a transmissão e a recepção dos pacotes; Variação do Atraso (*Jitter*), que representa a variação na latência de entrega e é especialmente crítica para aplicações em tempo real; e Perda de Pacotes (*Packet Loss*), definida como a quantidade de pacotes transmitidos que não foram recebidos com sucesso. O procedimento sistemático de configuração dos fluxos, de execução do experimento e de extração dos dados é detalhado no Algoritmo 1.

4. Resultados Experimentais

Nessa seção são discutidos os principais resultados obtidos considerando o impacto da estratégia de gerenciamento de fluxos no tempo de convergência; confiabilidade e perda de pacotes, por fim o atraso, jitter e vazão.

4.1. Impacto da Estratégia de Gerenciamento de Fluxos no Tempo de Convergência

A análise dos resultados evidencia uma dicotomia clara entre estratégias reativas e proativas de gerenciamento de fluxos. A Figura 2a ilustra o tempo de convergência agregado por modelo de mobilidade. Observa-se que, no Modo Reativo, a rede sofre penalizações severas de latência decorrentes da troca de mensagens Packet-In/Flow-Mod entre os *switches* e o controlador durante cada *handover*.

Em contraste, o Modo Proativo apresentou tempos de convergência muito próximos de zero ($T_{conv} \approx 0$) em todos os modelos de mobilidade. Isso valida a premissa de que, ao eliminar a dependência do controlador no reencaminhamento inicial, a rede torna-se resistente à latência do plano de controle, dependendo apenas da velocidade de associação física (L2) entre as estações.

Algoritmo 1 Procedimento de coleta e extração de métricas de QoS

Require: Topologia Net , Conjunto de Estações S , Tempo de Simulação T_{max}

Ensure: Dataset de Métricas de QoS (D_{qos})

```

1:  $D_{qos} \leftarrow \emptyset$ 
2: Iniciar Topologia e Mobilidade em  $Net$ 
3: for cada par  $(src, dst)$  em  $S$  do                                     ▷ Configuração dos Pares
4:    $IP_{data} \leftarrow ObterIPWireless(dst)$ 
5:    $IP_{sig} \leftarrow ObterIPWired(dst)$ 
6:                                                                                   ▷ Receptor: WLAN e ETH
7:   ExecutarRemoto( $dst$ , "ITGRecv -a  $IP_{data}$  -Si  $IP_{sig}$  -l recv_log")
8:                                                                                   ▷ Emissor: fluxo UDP
9:   ExecutarRemoto( $src$ , "ITGSend -a  $IP_{data}$  -T UDP -t  $T_{max}$ ")
10: end for
11: Aguardar( $T_{max}$ )                                                         ▷ Execução
12: Parar todos os processos D-ITG
13: for cada arquivo de log  $L$  gerado do                                     ▷ Pós-Processamento
14:    $L_{dec} \leftarrow DecodificarLog(L, "ITGDec")$ 
15:    $M \leftarrow ExtrairMétricas(L_{dec})$ 
16:    $D_{qos} \leftarrow D_{qos} \cup M$ 
17: end for
18: Salvar  $D_{qos}$  em formato CSV

```

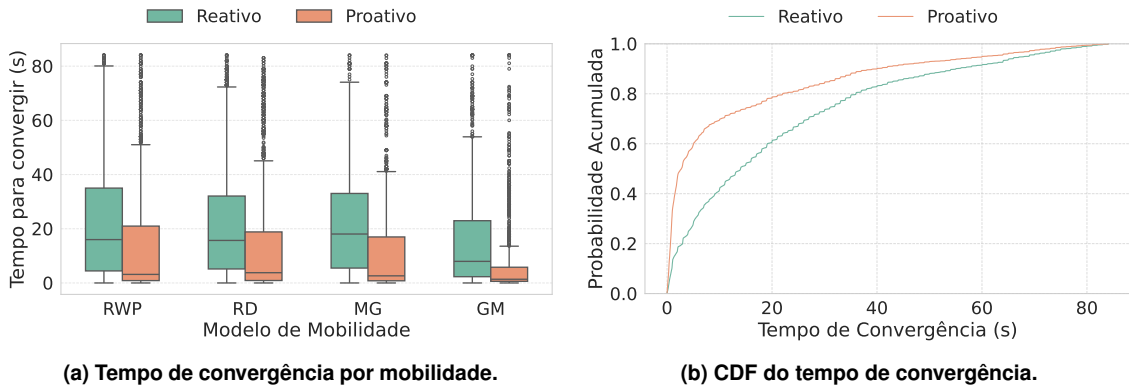


Figura 2. Tempo de convergência

A Figura 2b apresenta a função de distribuição acumulada (CDF) do tempo de convergência. A análise da CDF indica que a degradação no modo reativo não se limita a eventos isolados ou *outliers*, mas caracteriza um comportamento sistêmico. Verifica-se que aproximadamente 90% dos eventos de *handover* apresentam atrasos superiores a 60 segundos, o que evidencia que a saturação do controlador resulta na formação de filas de processamento que afetam a maioria das estações da rede. A curva longa à direita indica que, em momentos de pico de mobilidade simultânea, algumas estações chegaram a enfrentar desconexões superiores à média, evidenciando a incapacidade do controlador centralizado de lidar com rajadas de solicitações Packet-In.

Como mostra a Figura 3, a escalabilidade foi o fator mais crítico para a arquitetura SDN reativa, embora o impacto varie significativamente conforme o controlador utilizado.

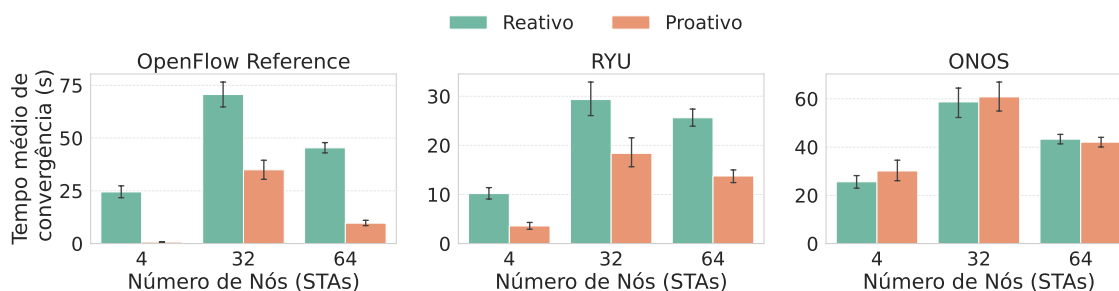


Figura 3. Impacto da escala na carga do controlador.

Conforme o número de estações e APs aumenta, a carga no controlador cresce de forma acentuada devido ao volume simultâneo de eventos de migração, resultando em perfis de saturação distintos: o *OpenFlow Reference* atingiu picos superiores a 70 s em cenários médios, enquanto o RYU apresentou um comportamento mais estável em larga escala quando comparado à sua própria linha de base em cenários menores. Por outro lado, o controlador ONOS, apesar de apresentar latências iniciais elevadas, demonstrou maior resiliência em redes densas (64 STAs), sugerindo que sua arquitetura distribuída absorve melhor o *overhead* de eventos simultâneos.

Juntando todos estes cenários, obteve-se um tempo médio de convergência de 16,56 segundos, com um intervalo de confiança (IC) de 95% entre 16,28 s e 16,84 s. A margem de erro reduzida ($\pm 0,56$ s ou 3,32%) indica que esse atraso elevado não foi um evento isolado (*outlier*), mas sim um comportamento consistente de saturação do plano de controle. Em termos práticos, períodos de interrupção dessa magnitude inviabilizam aplicações de tempo real, como VoIP ou *streaming* de vídeo, o que confirma que estratégias puramente reativas são inadequadas para redes móveis densas, independentemente da robustez do controlador escolhido.

4.2. Confiabilidade e Perda de Pacotes

A correlação entre o tempo de convergência da rede e a sua confiabilidade operacional é evidenciada pelo volume de tráfego descartado. A Figura 4 apresenta o número de pacotes perdidos em função do aumento do número de estações para diferentes modelos de mobilidade e controladores.

Diferentemente do que se observa nas métricas de latência pura, a estratégia de PreInstall (representada pelas linhas sólidas) não elimina a perda de pacotes, mas sim altera significativamente a curva de degradação. Nota-se que, enquanto as configurações padrão (linhas tracejadas) apresentam uma subida íngreme e atingem picos de perda superiores a 80 ou 100 pacotes em cenários de 64 nós, o uso de tabelas pré-instaladas consegue manter essa perda em patamares inferiores, retardando o colapso da comunicação.

Em relação aos modelos de mobilidade:

- Manhattan Grid: Demonstrou ser o cenário de menor estresse para os controladores, apresentando as menores taxas de perda absoluta, possivelmente devido à restrição de movimento em eixos ortogonais que estabiliza as trocas de células.
- Gauss-Markov: Apresentou os maiores picos de perda de pacotes (sup. 40%)

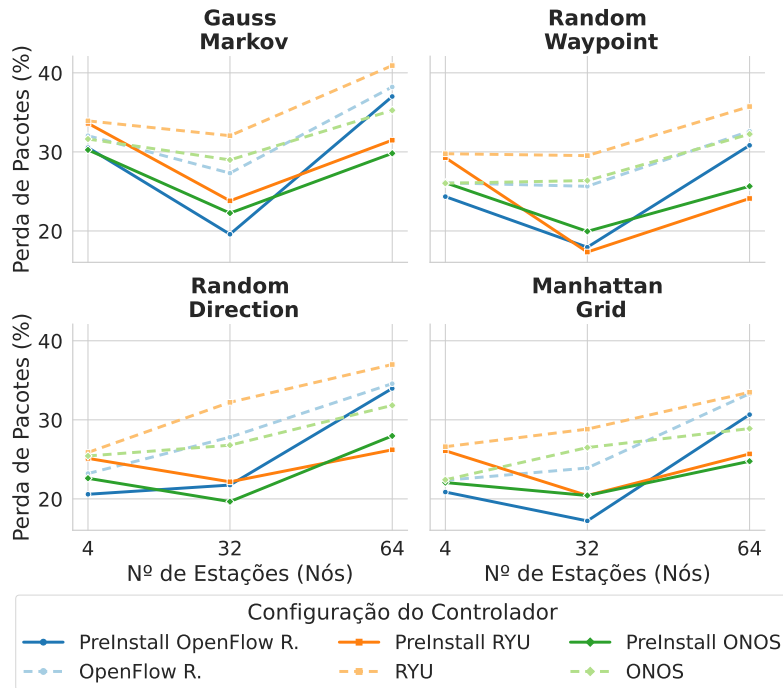


Figura 4. Perda de pacotes.

quando operado sem pré-instalação, evidenciando que a variação contínua de velocidade exige uma frequência de atualização que satura o canal de controle.

Entre os controladores, o ONOS com pré-instalação (linha verde sólida) destacou-se pela estabilidade, evitando a degradação abrupta vista no OpenFlow Reference ao atingir 64 nós. Contudo, a análise visual revela um comportamento não linear: a perda decresce em topologias médias (32 nós) antes de voltar a subir em alta escala. Essa dinâmica sugere que a topologia média representa um ponto de equilíbrio operacional, onde a densidade é suficiente para garantir conectividade sem incorrer no *overhead* excessivo que satura os controladores em cenários massivos.

4.3. Atraso, Jitter e Vazão

Como observado na Figura 5a, o atraso médio aumenta proporcionalmente ao número de estações. Esse comportamento é particularmente evidente nos controladores OpenFlow Reference e RYU em cenários com 64 estações simultâneas. O controlador ONOS destaca-se nessa métrica por apresentar melhor desempenho em cenários de maior escala (32 e 64 nós). Uma explicação provável para esse resultado é a arquitetura distribuída do ONOS, projetada especificamente para gerenciar altas cargas de eventos na rede. Adicionalmente, observa-se que, em cenários com fluxos pré-instalados, não é necessário consultar o controlador, o que resulta em menor atraso no processamento inicial.

O *jitter* é fundamental para aplicações em tempo real, como transmissões de voz e vídeo. A Figura 5b resume os resultados experimentais sobre a variação do atraso. Observa-se que os modelos de mobilidade Gauss-Markov e Random Waypoint apresentam maior instabilidade, conforme evidenciado pelas barras de erro superiores em comparação ao modelo Manhattan Grid. No Manhattan Grid, o movimento das estações é limitado por trajetórias restritas, tornando-as significativamente mais previsíveis para os

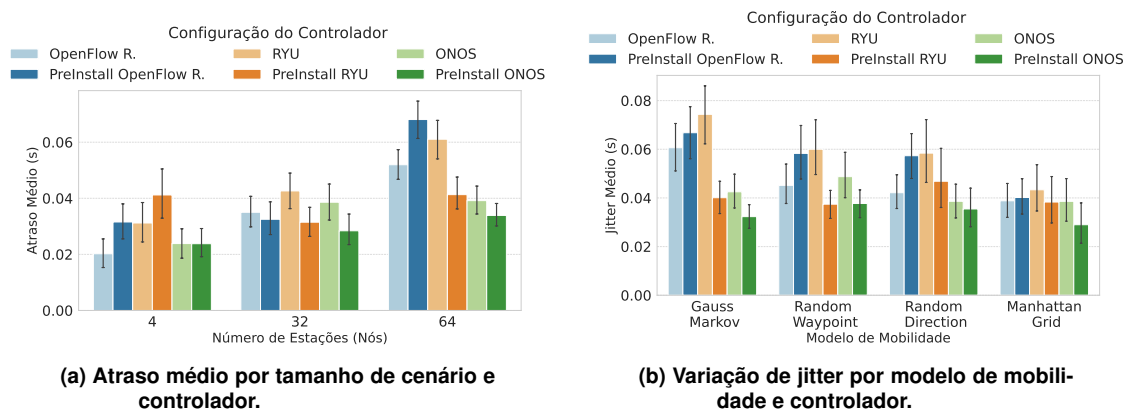


Figura 5. Atraso e jitter.

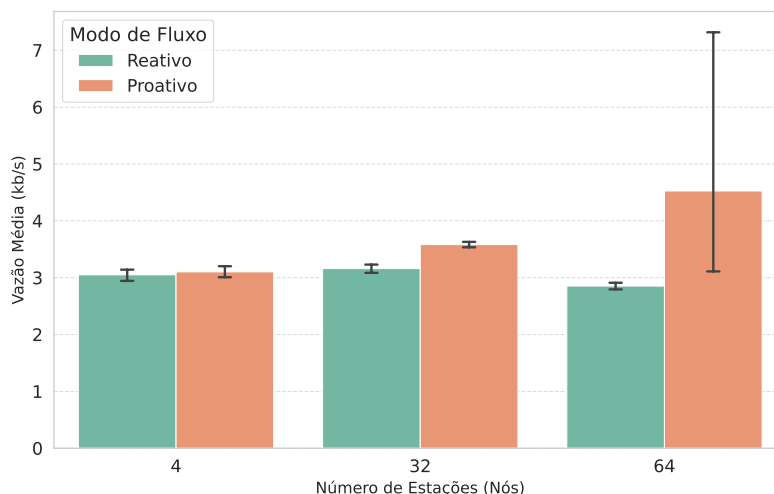


Figura 6. Impacto da instalação de fluxos na vazão média.

controladores.

Como visto na Figura 6, em todos os cenários, a vazão média é maior no modo proativo. No modo reativo, os primeiros pacotes de um novo fluxo geram uma interrupção no controlador para solicitação de regras; dessa forma, em cenários com alta densidade de estações móveis, a rede aloca recursos excessivos para o cálculo de rotas, o que justifica a queda de vazão de até 3 kb/s. Contudo, embora a vazão média dos controladores proativos apresente melhor desempenho em cenários de maior escala, as barras de erro são significativamente maiores, indicando que a rede está operando próxima ao seu limite de saturação.

5. Conclusão e Trabalhos Futuros

Este trabalho apresenta uma avaliação experimental de três controladores no contexto de redes sem fio definidas por software (SDWN) em redes sem fio definidas por software (SDWN), avaliando o impacto de diferentes modelos de mobilidade e de três controladores em modo proativo/reactivo no desempenho da rede. Por meio de um projeto fatorial executado no ambiente Mininet-WiFi, foi possível isolar o peso do processamento do plano de controle na latência de *handover*.

Os resultados experimentais revelaram um gargalo crítico inerente à abordagem reativa padrão. Nos cenários, observou-se que o tempo médio de convergência atingiu 16.56 segundos (IC 95%: [16.28; 16.84]), uma latência proibitiva para aplicações sensíveis ao atraso e duas ordens de grandeza acima daquela dos cenários com regras pré-instaladas.

A comparação entre controladores demonstrou que, independentemente da robustez da arquitetura (ONOS *carrier-grade* versus Ryu ou OpenFlow Reference), o atraso de propagação e de processamento domina o desempenho no modo reativo. Em contrapartida, a estratégia proativa eliminou virtualmente o tempo de interrupção, validando a hipótese de que a inteligência da rede deve antecipar a mobilidade, e não apenas reagir a ela.

Como trabalhos futuros, pretende-se investigar a extensão e a modificação do próprio protocolo OpenFlow para mitigar os atrasos de convergência inerentes ao plano de controle em cenários de alta mobilidade. Pretende-se explorar a implementação de mecanismos de instalação de fluxos, permitindo que o Ponto de Acesso (AP) execute decisões de encaminhamento pré-configuradas, sem necessidade de consulta imediata ao controlador durante o *handover*. Esta modificação visa reduzir o *overhead* de sinalização e a latência de processamento no controlador, evoluindo o protocolo para um modelo híbrido que combine a inteligência centralizada do SDN com uma capacidade de resposta local acelerada a eventos de mobilidade crítica.

Agradecimentos

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (444077/2024-3, 447525/2024-7 e 407568/2025-5), pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo Comitê Gestor da Internet Brasileira (CGI.br), com os auxílios 2018/23097-3, 2020/05182-3 e 2025/00513-5, e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES).

Referências

- Ahmed, N. et al. (2026). Comparative performance evaluation of ryu and opendaylight sdn controllers using mininet. *AlQalam Journal of Medical and Applied Sciences*, pages 64–67.
- Bello, L. L., Lombardo, A., Milardo, S., Patti, G., and Reno, M. (2020). Experimental assessments and analysis of an sdn framework to integrate mobility management in industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 16(8):5586–5595.
- Bujari, A., Calafate, C. T., Cano, J.-C., Manzoni, P., Palazzi, C. E., and Ronzani, D. (2017). Flying ad-hoc network application scenarios and mobility models. *International Journal of Distributed Sensor Networks*, 13(10):1550147717738192.
- Chaurasia, A., Mishra, S. N., and Chinara, S. (2019). Performance evaluation of software-defined wireless networks in IT-SDN and mininet-wifi. In *2019 1st International Conference on Advances in Information Technology (ICAIT)*, pages 315–319.
- Das, D., Rana, M. K., Sardar, B., Pecorella, T., and Saha, D. (2024). A comparative analysis of distributed mobility management schemes for 5g-based intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*.

- Johnson, D. B. and Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer.
- Khan, M. A., Dang, X. T., Doersch, T., and Peters, S. (2018). Mobility management approaches for sdn-enabled mobile networks. *Annals of Telecommunications*, 73:719–731.
- Labraoui, M., Boc, M. M., and Fladenmuller, A. (2017). Self-configuration mechanisms for sdn deployment in wireless mesh networks. *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–4.
- Liang, B. and Haas, Z. J. (2003). Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM Transactions On Networking*, 11(5):718–732.
- Lima, J. P., de Medeiros, Á. A., de Aguiar, E. P., Silva, E. F., de Sousa, V. A., Nunes, M. L., and Reis, A. L. (2023). Deep learning-based handover prediction for 5g and beyond networks. In *ICC 2023-IEEE International Conference on Communications*, pages 3468–3473. IEEE.
- Marquezan, C. C., Despotovic, Z., Khalili, R., Perez-Caparrós, D., and Hecker, A. (2016). Understanding processing latency of SDN based mobility management in mobile core networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7.
- Mobile, U. (1998). Universal mobile telecommunications system (umts); selection procedures for the choice of radio transmission technologies of the umts (umts 30.03 version 3.2. 0).
- Pentikousis, K., Wang, Y., and Hu, W. (2013). Mobileflow: Toward software-defined mobile networks. *IEEE Communications Magazine*, 51(7):44–53.
- Saoud, B., Shayea, I., Alnakhli, M. A., and Mohamad, H. (2025). Mobility and handover management in 5g/6g networks: Challenges, innovations, and sustainable solutions. *Technologies*, 13(8):352.
- Silva, R., Santos, D., Meneses, F., Corujo, D., and Aguiar, R. L. (2021). A hybrid SDN solution for mobile networks. *Computer Networks*, 190:107958.
- Sornlertlamvanich, P., Ang-Chuan, T., Sae-Wong, S., Kamolphiwong, T., and Kamolphiwong, S. (2016). SDN-based network mobility. In *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 1–6.
- Tantayakul, K., Dhaou, R., and Paillassa, B. (2016). Impact of sdn on mobility management. In *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, pages 260–265.
- Tantayakul, K., Dhaou, R., and Paillassa, B. (2019). SDN aided mobility management for connected vehicle networks.
- Tong, H., Wang, T., Zhu, Y., Liu, X., Wang, S., and Yin, C. (2021). Mobility-aware seamless handover with mptcp in software-defined hetnets. *IEEE Transactions on Network and Service Management*, 18(1):498–510.