

# Testes com Aplicativos de Controladores Inteligentes da RAN para o Monitoramento de KPIs e Slicing de Recursos de Rede

João P. S. H. Lima<sup>1</sup>, Paulo R. B. da Silva<sup>1</sup>, Michelle S. P. Facina<sup>1</sup>, Erika C. Alves<sup>1</sup>, Vitalii Afanasiev<sup>1</sup>, Gustavo C. Lima<sup>1</sup>

<sup>1</sup>Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) – Campinas, SP – Brasil

{jsales, prsilva, mfacina, erikaa, vitaliiaf, gcorrea}@cpqd.com.br

**Abstract.** *The smart management of RAN (Radio Access Network) resources in a mobile network ecosystem and the disaggregation of network components can be performed through a central controller known as RIC (RAN Intelligent Controller). The ONF (Open Networking Foundation) and the OAI (OpenAirInterface) provide platforms with open-source solutions for this task. We analyze these platforms by testing KPI (Key Performance Indicator)-monitoring functions, and slicing network resources. Metrics such as throughput, BLER (Block Error Rate), SNR (Signal-to-Noise Ratio), and the network's available bandwidth are presented through graphical interfaces.*

**Resumo.** *A gestão inteligente dos recursos da RAN (Radio Access Network) num ecossistema de redes móveis com virtualização e desagregação de componentes de rede pode ser feita através do controlador central conhecido como RIC (RAN Intelligent Controller). A Open Networking Foundation (ONF) e a OpenAirInterface (OAI) fornecem plataformas com soluções de código aberto para essa tarefa. Este artigo analisa tais plataformas por meio do teste do monitoramento de KPIs (Key Performance Indicator) e slicing de recursos de rede. Métricas como o throughput, a BLER (Block Error Rate), a SNR (Signal-to-noise Ratio) e a banda disponível da rede são mostradas via interfaces gráficas.*

## 1. Introdução

Soluções de mobilidade e telecomunicações estão passando por um processo de virtualização e desagregação. Antes monolíticas, elas estão sendo convertidas em blocos virtualizados coordenados de forma automática e escalável. O novo paradigma tende a usar gerenciadores como o *Kubernetes* e comunicação via APIs (*Application Programming Interfaces*) [O-RAN Alliance 2023]. Isso requer modificações à implementação com uso de componentes agregados em máquinas físicas (instalações *bare metal*), de forma a construir ambientes de testes adequados para a validação de ferramentas virtualizadas e descentralizadas usadas no controle da rede [Benzekki et al. 2016]. As alterações visam a diminuir investimentos em equipamentos e tempo de montagem do *setup*.

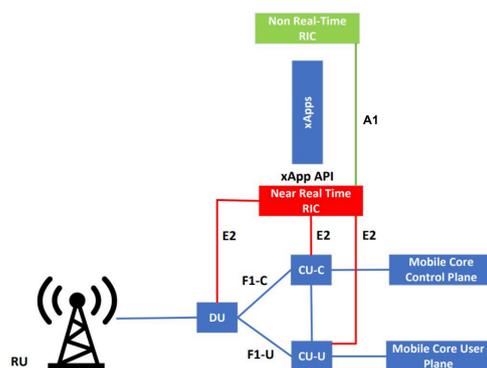
Alguns trabalhos relacionados a esse assunto já foram publicados. Por exemplo, [Singh et al. 2020] discute desafios associados à implementação da O-RAN (*Open-Radio Access Network*) e às possibilidades criadas por essa metodologia. Em [Mungari 2021], uma plataforma O-RAN é usada para obter relatórios periódicos sobre as condições de rede e adaptar dinamicamente a alocação de recursos. Em [Bonati et al. 2021] é apresentada a integração de elementos de *software* compatíveis com as normas da O-RAN

*Alliance* com uma pilha RAN de código aberto. Um emulador de redes *wireless* chamado Colosseum é usado para demonstrar o uso de xApps na coleta de dados na borda, no *scheduling* de *slices* de rede e no controle de métricas de rede em malha fechada via aprendizagem por reforço. Em [Bonati et al. 2022], xApps são instanciados dentro da estrutura de RICs (*RAN Intelligent Controllers*) do OpenRAN Gym, uma plataforma de testes da arquitetura O-RAN que opera em conjunto com o emulador Colosseum. Em [Xavier et al. 2023] demonstra-se um método de detecção de ataques do tipo DoS (*Denial-of-Service*) que usa as medidas da interface aérea para controlar dinamicamente a RAN. Aplica-se uma gama de algoritmos de *Machine Learning* para a análise de tráfego.

Este trabalho tem como objetivo a demonstração das soluções RIC da ONF (*Open Networking Foundation*) e da OAI (*OpenAirInterface*), além da descrição dos testes de monitoramento de KPIs (*Key Performance Indicators*) via CLI (*Command-Line Interface*) e interfaces gráficas do Grafana e dos testes de fatiamento dos recursos físicos da rede (*network slicing*) em *slices* de maior ou menor largura de banda. Segue a organização do restante do artigo: A Seção 2 resume a função do RIC e as características e aplicações das implementações da ONF e da OAI. A Seção 3 descreve os experimentos que nós realizamos para monitorar métricas da RAN e para fatiar recursos de rede. A Seção 4 avalia as diferentes soluções, comparando a implementação da ONF com a da OAI, e traz algumas constatações sobre o processo de implantação e experimentação das plataformas. Por fim, a Seção 5 apresenta as conclusões e possíveis trabalhos futuros.

## 2. Estrutura do RIC e plataformas analisadas

A plataforma RIC pode atuar com diferentes tempos de controle: um quase instantâneo (*near-RT RIC*) e outro de prazo mais estendido (*non-RT RIC*) (ver Figura 1) [O-RAN Alliance 2023]. Neste trabalho, nós tratamos apenas do *near-RT RIC*, dado que o *non-RT RIC* ainda não apresenta soluções maduras de código aberto.



**Figura 1. Arquitetura Open RAN: componentes e interfaces relacionados ao RIC.**

Para atuar em tempo quase real, o *near-RT RIC* opera mais próximo à DU (*Distributed Unit*) e à CU (*Control Unit*), que são componentes da RAN de camada mais elevada [O-RAN Alliance 2023]. As atividades do *near-RT RIC* estão ligadas principalmente a tarefas de RRM (*Radio Resource Management*), e se dão em intervalos entre 10 ms e 1 s [Bonati et al. 2021, ORAN Software Community 2022b].

Seu funcionamento é baseado em aplicativos externos (xApps) e em serviços

e componentes internos necessários para o RRM. Também se baseia em interfaces de comunicação padronizadas pela *O-RAN Alliance*, tal como a interface E2, que o conecta à DU e CU, e a interface A1, que permite ao *non-RT RIC* coordenar as ações do *near-RT RIC* (ver Figura 1) [Open Networking Foundation 2023a]. Além disso, modelos de operação conhecidos como SMs (*Service Models*) são usados para receber dados da RAN, processá-los e enviar respostas com ações de controle [ORAN Software Community 2022a]. A função de controle do *near-RT RIC* se dá por meio de (i) um banco de dados contendo informação da RAN; (ii) uma infraestrutura de envio/recebimento de mensagens para realizar a interface com diferentes componentes de rede; (iii) um sistema de comunicação por meio de APIs e (iv) um mecanismo de atuação para o controle efetivo das funções da RAN [ORAN Software Community 2022a].

## 2.1. ONF

A plataforma *near-RT RIC* da ONF se chama  $\mu$ ONOS-RIC, faz parte do projeto SD-RAN e se baseia em um controlador de SDNs (*Software-Defined Networks*) denominado ONOS (*Open Network Operating System*). Os processos internos e as trocas de mensagens do RIC são coordenadas pelos *pods* do próprio ONOS. Além disso, a solução  $\mu$ ONOS-RIC usa o *protobuf* (*protocol buffer*) para comunicação de dados através dos *endpoints* da API gRPC (*Google Remote Procedure Call*) [Open Networking Foundation 2023a]. O  $\mu$ ONOS-RIC também inclui a integração com ferramentas de aquisição e visualização de dados como o Prometheus e o Grafana [Open Networking Foundation 2022].

## 2.2. OAI

O FlexRIC é um controlador de SDN caracterizado pela flexibilidade e pela baixa utilização de recursos de memória e computação. Pode ser composto apenas por interfaces de comunicação e aplicações internas de mais baixo nível conhecidas como iApps (*internal Applications*) [Schmidt 2021]. O FlexRIC usa a biblioteca *agent*, que consiste de uma interface de comunicação, uma camada de abstração para o protocolo de comunicação da interface E2, um espaço de armazenamento de mensagens recebidas e uma API para acessar funções genéricas da RAN [Schmidt et al. 2021, Schmidt 2021]. A biblioteca *server* gerencia conexões de *agents*, multiplexando mensagens entre eles e seus iApps [Schmidt 2021]. Os iApps são úteis na construção de SMs dedicados a casos de uso específicos, como o *network slicing* [Schmidt et al. 2021, Schmidt 2021], por exemplo.

## 3. Experimentos e Cenários de Testes Utilizados com xApps

### 3.1. ONF

Nós implementamos o RIC ONF em um computador com processador Intel Broadwell de seis núcleos, memória RAM de 32 GB e disco rígido de 100 GB. O simulador de RAN da ONF tem capacidade para cenários com mais de uma centena de UEs (*User Equipment*) e arranjos hexagonais de até 7 células. Esses cenários podem, por exemplo, ser usados no balanceamento de tráfego e na predição de falhas de conexão [Open Networking Foundation 2023b].

Contudo, o SD-RAN pode ser integrado com a pilha da OAI, o que permite a implementação de um *setup* físico com SDRs (*Software-Defined Radios*) e UEs do tipo COTS (*Commercial Off-The-Shelf*). O *setup* físico possibilita mais cenários do que o

simulador, já que esse não possui o plano de usuário. Apesar de ser de difícil depuração, criação de configuração de rede e manipulação de arquivos de instalação, o *setup* físico permite explorar características do plano de dados, tais como dados de *throughput* e o recurso de *slicing* da rede. O xApp de *slicing*, RSM (*RAN Slice Management*), é usado para realizar um teste de vazão com uso da aplicação *iperf3* em uma rede LTE (*Long-Term Evolution*) com um único UE COTS (ver Figura 2). Um *slice* é configurado com maior largura de banda e alcança uma taxa de aproximadamente 6 Mbits/s. Em outro *slice*, diminui-se a largura de banda, reduzindo o tráfego por um fator de aproximadamente 5 para uma taxa próxima a 1,2 Mbits/s. É possível observar essa redução na Figura 2, a qual mostra um *slice* de alta taxa entre 10s e 14s e outro de taxa reduzida entre 47s e 51s.

Time Interval	Data Transferred	Throughput
10.00-11.00 sec	685 KBytes	5.61 Mbits/sec
11.00-12.00 sec	681 KBytes	5.58 Mbits/sec
12.00-13.00 sec	818 KBytes	6.70 Mbits/sec
13.00-14.00 sec	747 KBytes	6.11 Mbits/sec
47.00-48.00 sec	144 KBytes	1.18 Mbits/sec
48.00-49.00 sec	144 KBytes	1.18 Mbits/sec
49.00-50.00 sec	143 KBytes	1.17 Mbits/sec
50.00-51.00 sec	143 KBytes	1.17 Mbits/sec

Figura 2. *Slice* de taxa mais elevada à esquerda e de taxa mais baixa à direita.

### 3.2. OAI

O FlexRIC permite a integração com pilhas de código aberto, tais como a OAI e a srsRAN (*software radio system RAN*), através de *patches* que possibilitam a comunicação entre controlador e pilha. Foram utilizados no nosso experimento uma máquina com 16 GB de memória RAM, processador Intel Core i7 e 256 GB de disco rígido para demonstrar o xApp de monitoramento de KPIs das camadas MAC (*Media Access Control*), RLC (*Radio Link Control*) e PDCP (*Packet Data Convergence Protocol*) operando em uma pilha OAI 5G SA (*Standalone*). Um UE COTS conectado à rede 5G recebe pacotes de informação, de forma que os KPIs possam ser coletados, armazenados no banco de dados InfluxDB e exibidos em painéis elaborados com o Grafana. A Figura 3 mostra um *dashboard* do Grafana com a BLER (*Block Error Rate*) e a SNR (*Signal-to-Noise Ratio*) do canal PUSCH (*Physical Uplink Shared Channel*) para um UE COTS conectado à rede 5G SA executando uma aplicação de *streaming* de vídeo. O painel permite a análise das métricas em forma de séries temporais. Além do BLER e da SNR do PUSCH, é possível monitorar sinais de HARQ (*Hybrid Automatic Repeat Request*), número de pacotes retransmitidos, número de bytes, entre outros.

Quanto ao *slicing*, o FlexRIC possui um modelo de serviço (códigos base para funcionalidades específicas do controlador) chamado *slice\_sm* que gere *slices* da RAN, incluindo as tarefas de criação e remoção, *scheduling* e a atribuição a UEs. Nós não reproduzimos neste artigo experimentos de *slicing* com o FlexRIC por limitações de espaço e por haver testes detalhados dessa funcionalidade em [Schmidt et al. 2021].

## 4. Discussões

A implementação da ONF é feita em um *cluster* Kubernetes, o que permite alta escalabilidade e a orquestração entre os *Pods* da rede. Os *Pods* contêm *containers*, os quais



**Figura 3. BLER e SNR do PUSCH: *streaming* com UE COTS usando o FlexRIC.**

abrigam unidades de *software* necessárias para a implementação do controlador. A vantagem é a capacidade de auto-regeneração e replicação de unidades algorítmicas, possibilitando a regulação dos recursos computacionais. Outro benefício da ferramenta da ONF é sua integração nativa com o Prometheus e o Grafana, ferramentas de armazenamento e visualização de dados. Por outro lado, a instalação e o *deploy* são complexos, já que há muitas dependências e especificidades técnicas. Os processos são delicados e podem sofrer falhas sistêmicas devido a pequenas alterações em códigos componentes.

A solução da OAI não é baseada em *clusters* de *pods*, e, conseqüentemente, não é projetada para escalabilidade. Ela depende de *patches* a pilhas RAN, os quais permitem o controle delas através do FlexRIC. Apesar disso, o FlexRIC é maleável no desenvolvimento de testes e aplicações, pois ele possibilita a modificação de elementos para transmitir dados e receber comandos de uma lógica de controle externa. Ele também pode se integrar com ferramentas como o Grafana e o banco de dados InfluxDB. Dessa forma, dados armazenados podem ser consumidos por outras aplicações. Se alguma das métricas coletadas, como a qualidade de serviço do canal, a taxa de transmissão ou o número de pacotes retransmitidos, servir como entrada para algoritmos de outros xApps, é possível adaptar SMs existentes com as novas medidas. O consumo desses dados pode, por exemplo, ajudar no desenvolvimento de aplicações para detectar a presença de ataques à rede.

## 5. Conclusões e trabalhos futuros

Este artigo discute brevemente a função do RIC, as implementações da ONF e OAI e cenários de testes utilizados para as duas. Para demonstrar o monitoramento de métricas da RAN e o *slicing* de recursos físicos da rede, foi feito o *deploy* de *pods* para a solução da ONF, assim como a integração com bancos de dados (InfluxDB) e interfaces gráficas (Prometheus e Grafana) tanto para a solução da ONF como para a solução da OAI. Nós pretendemos em trabalhos futuros (i) expandir o *setup* com novas células e aumentar a quantidade de usuários e (ii) incorporar o *non-RT RIC*, permitindo a orquestração das funções do *near-RT RIC*. Com base nesses novos *setups*, deseja-se prever as estatísticas da camada física e as condições de canal, monitorar e prever a QoE (*Quality of Experience*) sob condições de *network slicing* e criar rotinas de localização *indoor*.

## Referências

- Benzekki, K., El Fergougui, A., and Elbelrhiti Elalaoui, A. (2016). Software-defined networking (sdn): a survey. *Security and Communication Networks*, 9(18):5803–5833.
- Bonati, L., D’Oro, S., Polese, M., Basagni, S., and Melodia, T. (2021). Intelligence and learning in O-RAN for data-driven nextG cellular networks. *IEEE Communications Magazine*, 59(10):21–27.
- Bonati, L., Polese, M., D’Oro, S., Basagni, S., and Melodia, T. (2022). Intelligent closed-loop ran control with xapps in openran gym.
- Mungari, F. (2021). An rl approach for radio resource management in the o-ran architecture. In *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–2.
- O-RAN Alliance (2023). O-RAN Architecture Description 8.0. Disponível em: <https://orandownloadsweb.azurewebsites.net/specifications>. Acessado: 2023-03-17.
- Open Networking Foundation (2022). onos-exporter. <https://docs.sd-ran.org/master/onos-exporter/README.html?highlight=prometheus>. Acessado: 2022-08-14.
- Open Networking Foundation (2023a). Architecture. <https://docs.sd-ran.org/master/architecture.html?highlight=grpc>. Acessado: 2022-12-23.
- Open Networking Foundation (2023b). RAN Simulator. <https://docs.sd-ran.org/master/ran-simulator/README.html>. Acessado em: 2023-02-05.
- ORAN Software Community (2022a). Near Real time RAN Intelligent Controller (RIC). Disponível em: <https://docs.o-ran-sc.org/en/latest/projects.html#near-realtime-ran-intelligent-controller-ric>. Acessado: 2022-09-09 .
- ORAN Software Community (2022b). OSC Installation Guides. <https://docs.o-ran-sc.org/projects/o-ran-sc-ric-plt-ric-dep/en/latest/installation-guides.html#overview,note> = Acessado: 2022-12-10.
- Schmidt, R. (2021). *Slicing in heterogeneous software-defined radio access networks*. Theses, Sorbonne Université.
- Schmidt, R., Irazabal, M., and Nikaein, N. (2021). FlexRIC: An SDK for next-Generation SD-RANs. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT ’21*, page 411–425, New York, NY, USA. Association for Computing Machinery.
- Singh, S. K., Singh, R., and Kumbhani, B. (2020). The evolution of radio access network towards open-ran: Challenges and opportunities. In *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6.
- Xavier, B. M., Dzaferagic, M., Collins, D., Comarela, G., Martinello, M., and Ruffini, M. (2023). Machine learning-based early attack detection using open ran intelligent controller.