

Uma Avaliação de Desempenho da Implantação de Rádio Definido por Software em Ambientes Virtualizados para Redes de Acesso Sem Fio em Nuvem

Govinda M. G. Bezerra^{1,2}, Tadeu N. Ferreira², Diogo M. F. Mattos¹

¹MídiaCom - Departamento de Engenharia de Telecomunicações
Universidade Federal Fluminense (UFF)

²LAProp - Departamento de Engenharia de Telecomunicações
Universidade Federal Fluminense (UFF)

{govindamgb, tadeu_ferreira, diogo_mattos}@id.uff.br

Abstract. *Fifth generation (5G) mobile networks must be flexible to meet different connectivity requirements of Internet of Things (IoT) devices. In this context, radio virtualization is promising, since it enables multiple heterogeneous virtual radios to coexist on top of the same hardware. One approach to implementing virtual radios is through the use of virtualization tools, such as virtual machines and containers. In this article, we evaluate experimental limits of the Ethernet communication of a SDR device and the impact of the virtualization on its performance, in order to verify the suitability of the virtual environments for the use of SDR applications. The results show that the performance of containers is similar to native Linux.*

Resumo. *A quinta geração (5G) da rede celular deve ser flexível para atender às diferentes demandas de conexão dos dispositivos de Internet das Coisas (IoT). A virtualização de rádios é uma técnica promissora, pois permite que diversos rádios virtuais heterogêneos coexistam em um mesmo hardware, enquanto o processamento é realizado remotamente. Uma abordagem viável para a implementação de rádios virtuais é através da virtualização de sistemas operacionais completos, tais como máquinas virtuais e contêiner. Este artigo avalia os limites experimentais da comunicação Ethernet do dispositivo de Rádio Definido por Software (Software Defined Radio - SDR) e os impactos da virtualização com intuito de verificar a adequabilidade dos ambientes virtuais para uso a criação de redes de acesso sem fio virtuais em nuvem. Os resultados mostram que o desempenho do contêiner se assemelha ao do Linux nativo.*

1. Introdução

Rádios definidos por software (*Software Defined Radio* - SDR) utilizam hardware comercial (*commercial off-the-shelf* - COTS) para implementar funções de rádio e de processamento de sinais. O uso de equipamentos genéricos e produzidos em larga escala tende a diminuir os custos de produção dos sistemas que utilizam esta tecnologia [Dias et al. 2020]. Além disso, a modificação e a atualização de aplicações baseadas em

SDR são mais simples em virtude de as alterações serem apenas em nível de software e, mesmo em casos de avarias de hardware, o reparo é facilitado devido à disponibilidade comercial dos dispositivos.

A quinta geração (5G) da rede celular vislumbra a conexão massiva de dispositivos de Internet das Coisas (IoT), cada um com seus próprios requisitos de latência, largura de banda e mobilidade. Para atender às diversas demandas de diferentes cenários, é importante que as redes móveis sejam flexíveis. Uma das principais tecnologias para a rede de acesso das redes 5G é a Rede de Acesso de Rádio na Nuvem (*Cloud Radio Access Network - C-RAN*). C-RAN permite a operação de rede com eficiência energética e possível economia de custos em infraestrutura de banda base. Além disso, melhora a capacidade da rede através do balanceamento de carga e do processamento cooperativo de sinais provenientes de várias estações base. A ideia principal do C-RAN é prover uma rede de acesso de rádio distribuída, porém com o processamento do sinal de rádio centralizado na nuvem [Checko et al. 2015]. Uma limitação do C-RAN é a necessidade de taxas de transmissão muito altas para o enlace *fronthaul*. Nesse sentido, há um esforço de pesquisa para diminuir as exigências desse enlace, redefinindo a localização das funções de Estações Base. Quanto mais funções são executadas localmente, próximo ao usuário, menor são as exigências de taxa de bit da rede *fronthaul*. Por outro ângulo, quanto mais funções forem centralizadas em *data centers*, maior é a otimização do uso de hardware e mais flexível é a rede para balancear a carga de processamento. [Larsen et al. 2018]. Neste contexto, a virtualização de rádios é interessante, pois permite que diversos rádios virtuais, cada um com suas próprias características, coexistam em um mesmo hardware [Kist et al. 2018], enquanto o controle do hardware é centralizado na nuvem. O estado da arte desta tecnologia de virtualização da rede de acesso de rádio ainda é embrionário e as soluções atuais não atendem a todos os desafios de isolamento, de alocação dinâmica de recursos e de heterogeneidade de hardware.

As tecnologias de virtualização existentes, como máquinas virtuais e contêiner, são ferramentas que podem ser utilizadas no desenvolvimento de rádios virtuais, garantindo o isolamento e a alocação de recurso entre os diferentes usuários. Para que o seu uso seja eficaz, é necessário que o impacto da virtualização na comunicação dos rádios seja conhecido e as suas limitações elencadas. Este artigo analisa o impacto da virtualização na comunicação de dispositivos rádio definidos por software em ambientes virtualizados para a implantação de redes de acesso de rádio na nuvem. Para tanto, utilizam-se duas abordagens de virtualização: máquinas virtuais e contêiner. Para cada tipo de virtualização, são realizados dois diferentes cenários de experimentação que visam aferir o limite da vazão de rede, o uso de CPU e a taxa máxima de amostras por segundo alcançados pelo dispositivo SDR. Os testes também são realizados em um sistema operacional nativo e os seus resultados são utilizados para comparação de desempenho entre ambientes virtualizados. Desta forma, é possível identificar o quanto cada uma das tecnologias de virtualização influencia o desempenho do sistema.

Trabalhos anteriores analisam os limites experimentais alcançados em ambiente SDR utilizando apenas sistemas operacionais nativos [Schmid et al. 2007] ou comparam diferentes arcabouços SDR, como GNU Radio e Cognitive Radios Test System version 2.0 (CRTSv.2) [Becker et al. 2020]. Outros trabalhos buscam quantificar o impacto da virtualização em determinadas aplicações, como roteadores virtuais [Mattos et al. 2012]

e ambientes de computação em nuvem [Huber et al. 2011]. Diferentemente, este trabalho verifica o impacto da virtualização em um cenário específico de SDR.

O restante do artigo está organizado da seguinte forma. A seção 2 apresenta as ferramentas de SDR utilizadas. Na seção 3, propõe-se a metodologia de avaliação e os cenários de testes. A avaliação dos resultados é feita na seção 4. A seção 5 discute os trabalhos relacionados. A seção 6 conclui o artigo.

2. Rádio Definido por Software (*Software Defined Radio* - SDR)

Rádio Definido por *Software* (*Software Defined Radio* - SDR) é um sistema de radiocomunicação cujas funções de camada física são implementadas por *software*. A grande vantagem de utilizar o paradigma SDR, em relação à abordagem tradicional de implementação por *hardware*, é a facilidade em reprogramar o rádio, o que permite construir diferentes tipos de transmissores e receptores utilizando um *hardware* genérico. Além disso, atualizações no SDR, como adição de funções, protocolos e criptografia, também são simplificadas, uma vez que basta realizar mudanças no código do *software* de controle. Assim, a versatilidade oferecida pelo *hardware* SDR diminui o custo de desenvolvimento, o custo de produção e os custos de atualização dos sistemas.

Visto de outro ângulo, uma das grandes limitações da tecnologia SDR é causada pelo uso de dispositivos genéricos de *hardware* que proporcionam a sua flexibilização. Dispositivos de *hardware* projetados como uma função específica são mais eficientes, em termos de processamento e utilização dos componentes, que dispositivos de *hardware* e processadores de uso geral devido à otimização da execução física das funções específicas.

Um rádio definido por *software* idealmente realiza todo o processamento do sinal em software, utilizando uma arquitetura de amostragem direta, na qual o sinal é digitalizado diretamente após a antena, sem nenhuma conversão de frequência. O problema dessa abordagem está na necessidade de uma alta frequência de amostragem, o que implica conversores analógicos-digitais rápidos e eficientes. Outra desvantagem é a grande largura de banda necessária entre o dispositivo de rádio e o dispositivo de processamento do sinal. A banda de transmissão necessária cresce com a banda de frequência amostrada, assim como com a taxa de amostragem usada.

Uma arquitetura mais acessível e bastante utilizada pelos dispositivos de hardware SDR utiliza a conversão de frequência em duas etapas. Conforme ilustrado na Figura 1, a primeira etapa da recepção do sinal ocorre no *front-end* analógico, onde o sinal recebido pela antena é convertido para uma frequência intermediária (FI). A segunda etapa de conversão de frequência ocorre no *front-end* digital, no qual o sinal é convertido de FI para a banda base. Em seguida, o sinal é processado no computador, que consiste no *back-end* digital. De forma simétrica, na transmissão, o sinal é produzido pelo computador e, então, convertido para uma FI pelo *front-end* digital. O sinal digital é transformado em analógico e convertido para a frequência de transmissão no *front-end* analógico.

2.1. Dispositivo de Rádio Definido por Software USRP

USRP (*Universal Software Radio Peripheral*) consiste em um dispositivo de rádio definido por software, cuja arquitetura está ilustrada na Figura 1. Nos dispositivos USRPs, o *front-end* de RF é implementado em placas chamadas placas-filhas (*daughterboards*)

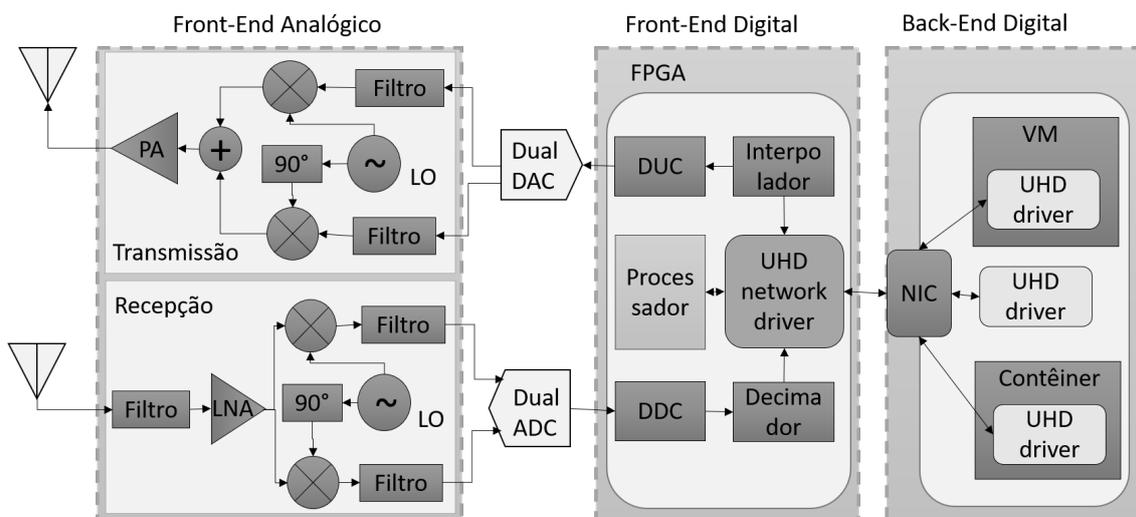


Figura 1. Arquitetura de um Rádio Definido por Software utilizando uma placa de rede Ethernet para comunicação com o computador hospedeiro.

e o *front-end* digital é implementado na placa-mãe do dispositivo. Além da FPGA (*Field Programmable Gate Array*), a placa-mãe possui um oscilador local (LO) e os conversores analógico-digital (ADC) e digital-analógico (DAC). A maioria dos dispositivos USRP disponíveis comercialmente necessitam de um computador hospedeiro que controle o hardware SDR e realize o processamento do sinal em banda básica. É necessário que o computador tenha instalado o *drive* UHD do hardware USRP.

O modelo do USRP utilizado neste trabalho é o N200, cuja conexão com o computador hospedeiro é realizada através da placa de rede Ethernet com capacidade de amostragem de até 50 MS/s de transmissão e recepção operando no modo full-duplex. N200 é composto de uma placa-mãe, responsável pelas funções de geração de relógio (*clock*), sincronização, ajuste de potência e interface com o computador. A placa-mãe do dispositivo N200 opera em uma faixa de frequência de 0 a 6 GHz, contém uma FPGA capaz de processar 100 MHz de banda, um conversor analógico-digital dual de 100 MS/s de 14 bits e um conversor digital-analógico dual de 400 MS/s de 16 bits.

A placa-filha utilizada neste trabalho é a SBX que opera na faixa de 400 MHz a 4,4 GHz e possui dois *front-ends*, um para transmissão ou recepção e outro somente para recepção. Os *front-ends* possuem osciladores locais independentes que permitem uma operação full-duplex com frequências de transmissão e recepção distintas. É recomendável realizar uma auto-calibração no dispositivo para minimizar o nível DC e o desbalanceamento IQ nos sinais. O resultado da calibração é salvo em um arquivo no computador hospedeiro e o *driver* UHD automaticamente realiza as correções quando as aplicações SDR são executadas. A comunicação entre o dispositivo USRP N200 e o computador é realizada através do protocolo da camada de aplicação VRT (*VITA Radio Protocol*) sobre datagramas UDP.

Os dispositivos USRP possuem restrições sobre os valores das frequências de amostragem que podem ser utilizados. É necessário que a razão entre a frequência do relógio e a frequência de amostragem seja um número inteiro, preferencialmente par. Além disso, valores de interpolação/decimação maiores que 128, devem ser divisíveis por

2. Caso a interpolação/decimação seja maior que 256, o valor deverá ser divisível por 4. Considerando que o modelo N200 possui uma frequência de relógio fixa de 100 MHz, as frequências de amostragem possíveis são limitadas pelas restrições mencionadas.

A largura de banda dos dispositivos USRP varia ao longo das cadeias de transmissão e recepção em, basicamente, três fases: no processamento de RF realizado pela placa-filha, no processamento digital realizado pela FPGA e na comunicação com o computador hospedeiro. Cada uma dessas fases tem a sua especificação de largura de banda máxima e, assim, a máxima largura de banda alcançada pelos sistemas desenvolvidos em determinado *hardware* é a menor largura de banda das 3 fases, já que esse será o gargalo de comunicação do sistema. No caso dos dispositivos utilizados no artigo, temos as seguintes características: a placa-filha SBX possui uma largura de banda de 40 MHz; o processamento digital realizado pela FPGA do N200 recebe amostras a uma taxa de 100 MS/s do ADC e de 400 MS/s do DAC; e a conexão Gigabit Ethernet com o computador hospedeiro possui uma taxa máxima teórica de 25 MS/s, utilizando amostras em fase e quadratura (I/Q) com 16 bits em modo full-duplex. Assim, a largura de banda das aplicações desenvolvidas com essa combinação de dispositivos está limitada à largura de banda da conexão Ethernet entre o USRP N200 e o computador hospedeiro.

3. Metodologia de avaliação de desempenho

Os modelos de dispositivos SDR que utilizam conexão Ethernet com o computador hospedeiro têm a sua largura de banda influenciada pela conexão, uma vez que todas as informações de controle e todos os sinais enviados e recebidos pelo hardware percorrem essa mesma interface.

O uso de frequências de amostragem que não sejam compatíveis com o hardware empregado pode levar à perda de pacotes e erros na operação. Por exemplo, pode ocorrer um *overflow* na recepção, no qual a aplicação não é capaz de processar os dados na velocidade em que as amostras são recebidas, ocasionando o descarte de amostras. Outro exemplo é o *underflow* na transmissão, quando a aplicação não consegue produzir amostras suficientes para suprir a taxa de transmissão escolhida. Assim, devido às inúmeras aplicações e aos benefícios que a virtualização pode proporcionar na área de virtualização de rádios, é importante observar o impacto que o uso de máquina virtual e contêiner têm na comunicação Ethernet e como isso pode afetar as aplicações SDR.

Este trabalho estuda o impacto da virtualização no enlace Ethernet, assim como analisa as limitações de comunicação de um rádio SDR que utiliza o padrão Ethernet como meio de conexão ao computador hospedeiro. Além disso, verificam-se as características e limitações no uso de dispositivos SDR em máquinas virtuais e contêiners, comparando com o desempenho de um sistema operacional nativo. Os experimentos são realizados em duas partes: teste do enlace Ethernet em ambientes virtualizados e teste da comunicação do dispositivo USRP com o computador. Em ambos os casos, a vazão de rede e o uso de CPU foram monitorados e correlacionados.

4. Resultados Experimentais

O computador hospedeiro utilizado na análise de desempenho é dotado de um processador I7-4770 CPU @ 3.40 GHz, 8 GB de memória, placa de rede Realtek RTL8111/8168/8411,

utilizando Linux 20.04 como sistema operacional e as plataformas de virtualização VirtualBox 6.1 e Docker 20.10. Ambos os sistemas operacionais virtualizados também são Linux 20.04. A máquina virtual é configurada com 4 GB de memória, 4 núcleos de CPU e seu acesso à rede é realizado através de uma ponte Ethernet (*bridge*) com a placa de rede Ethernet. Já a rede do contêiner é configurada utilizando uma *bridge* e um par de interfaces Ethernet virtuais, conforme mostra a Figura 2. A *bridge* é responsável por encaminhar os pacotes entre as interfaces de rede `eth0`, que corresponde à placa de rede física Ethernet, e `veth0`, que atua como um túnel Ethernet para a interface `veth1` do contêiner. Todos os ambientes são configurados com um endereço IP estático pertencente à mesma sub-rede do dispositivo USRP.

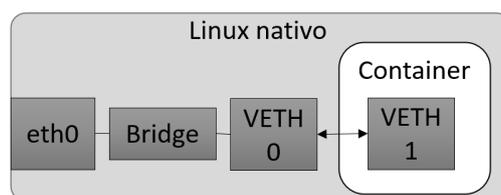


Figura 2. Configuração de rede do contêiner. A *bridge* é responsável por encaminhar os pacotes entre as interfaces de rede `eth0`, que corresponde à placa de rede Ethernet, e `VETH 0`, que atua como um túnel Ethernet para a interface `VETH 1` do container.

4.1. Comunicação Ethernet em ambientes virtualizados

A vazão de rede é altamente dependente do padrão de tráfego e das configurações de rede [VMware 2005]. Assim, a avaliação do impacto que as ferramentas de virtualização exercem na comunicação Ethernet é o primeiro fator estudado nesse trabalho, pois essa característica influencia como o dispositivo SDR comunica através dessa interface. Desta forma, o primeiro teste realizado consiste em testar a largura de banda da comunicação do computador hospedeiro através de sua placa de rede Ethernet, o respectivo uso de CPU para determinada vazão de comunicação e realizar a comparação com o desempenho obtido por uma máquina virtual (VM) e um contêiner nas mesmas condições. O objetivo do teste consiste em verificar como a virtualização e a alocação de recursos de hardware impactam a capacidade de comunicação via Ethernet de um sistema virtualizado. Os testes são realizados utilizando o software `iPerf`¹ que possui uma arquitetura cliente/servidor e permite medir a máxima largura de banda alcançável em redes IP. O servidor `iPerf` é executado em uma máquina com configurações idênticas às do computador hospedeiro que executa o cliente `iPerf` e ambas as máquinas pertencem à mesma rede local, conectadas através de um comutador Gigabit Ethernet.

Como a comunicação entre o dispositivo SDR e computador ocorre através do protocolo UDP na camada de transporte, o teste é realizado utilizando datagramas UDP. Cada ambiente é testado com datagramas variando de 16 a 1500 Bytes, o que possibilita verificar como o tamanho dos pacotes afeta a vazão alcançada pelos sistemas nos dois sentidos de comunicação *full-duplex*. Em um ambiente SDR em que a conexão com o computador hospedeiro é feita através de uma conexão Ethernet, o uso de CPU para processamento dos pacotes de rede é crítico, pois todas as informações recebidas e transmiti-

¹Disponível em <https://iperf.fr/>.

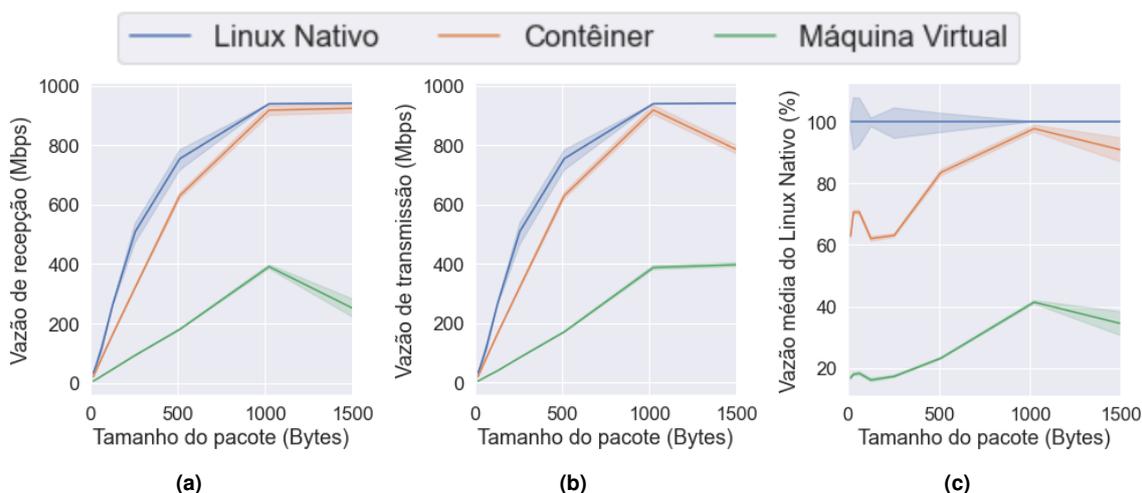


Figura 3. Análise do impacto do tamanho dos pacotes no desempenho do sistema: (a) mostra a vazão de recepção alcançada para diferentes tamanhos de pacotes UDP, (b) mostra a vazão de transmissão alcançada para diferentes tamanhos de pacotes UDP, (c) mostra a comparação da vazão de transmissão dos ambientes em relação à vazão média do Linux Nativo.

das pelo rádio utilizam essa interface e uma grande sobrecarga pode limitar a capacidade de processamento do sinal em banda base pelo hospedeiro.

A Figura 3 mostra a vazão alcançada utilizando diferentes tamanhos de pacotes nos três ambientes: Linux nativo, máquina virtual e contêiner. É possível observar que, quanto menor os pacotes, menor é a vazão alcançada para todos os ambientes, tanto para a recepção, Figura 3 (a), quanto para a transmissão, Figura 3 (b). Ademais, a máquina virtual atinge uma vazão para recepção e transmissão de pacotes consideravelmente menor que nos outros dois ambientes analisados. A Figura 3 (c) mostra a taxa de transmissão em relação ao valor médio da vazão do Linux nativo. Através desse resultado, é possível observar que a vazão do contêiner é acima de 80% da vazão média do Linux nativo para pacotes acima de 500 bytes, enquanto a vazão da máquina virtual atinge no máximo 40% da vazão alcançada com o Linux nativo. Através destes resultados, é possível concluir que a máquina virtual tem um decréscimo considerável em relação à vazão alcançada nativamente, o que pode provocar queda no desempenho de aplicações intensivas em rede executadas nesse ambiente. Esse resultado é reflexo das técnicas usadas para a virtualização das interfaces de rede. Na abordagem baseada em máquinas virtuais, a interface de rede virtual é paravirtualizada [VMware 2005], criando um novo *driver* no contexto da máquina virtual que recebe os pacotes do sistema físico através de cópia de memória executada pelo hipervisor. Na abordagem de contêiner, as interfaces são providas pelo núcleo do sistema Linux e, portanto, evitam-se cópias de memória entre diferentes contextos.

A Figura 4 mostra o uso de CPU durante a realização dos testes. É possível observar que os consumos observados nos ambientes de contêiner e o Linux nativo são muito próximos, porém o uso de CPU é três vezes maior em um ambiente com máquina virtual em função da execução de dois sistemas operacionais distintos (hospedeiro e visitante) e da paravirtualização da interface de rede.

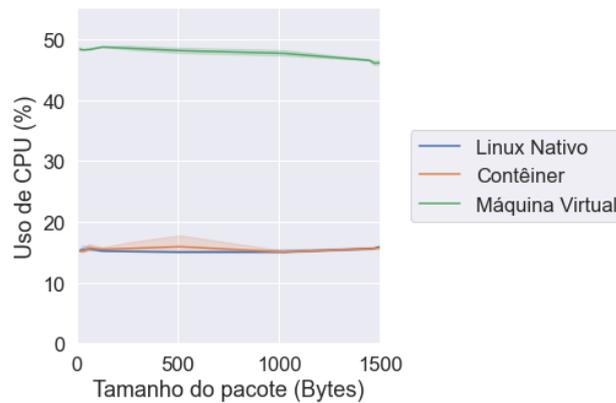


Figura 4. Uso de CPU durante os testes de rede de acordo com os tamanhos dos pacotes.

Esse teste inicial permite concluir que, independentemente da aplicação executada, a máquina virtual apresenta uma inerente sobrecarga de processamento e uma redução na vazão de rede em comparação ao desempenho obtido em um sistema operacional nativo.

4.2. Comunicação do USRP em ambientes virtualizados

A avaliação da comunicação do dispositivo USRP visa estabelecer o limite experimental de largura de banda entre o computador hospedeiro e o USRP N200 e, assim como o teste anterior, verificar o impacto da virtualização na comunicação com o dispositivo SDR. Os testes são realizados através da ferramenta de *benchmark* fornecida juntamente com o *driver* UHD, utilizando diferentes taxas de transmissão e recepção e diferentes modos de comunicação. Conforme mencionado anteriormente, as taxas possíveis de serem configuradas nas aplicações SDR estão limitadas àquelas que, dividindo a frequência de relógio do dispositivo, geram números inteiros. Assim, como o dispositivo possui um relógio de 100 MHz, utilizamos 25 taxas a partir de 1 MS/s até 50 MS/s que consiste no limite máximo especificado na documentação do USRP N200 para amostras de 8 bits. Os testes contemplam seis diferentes cenários, utilizando os três diferentes ambientes: Linux nativo, máquina virtual e contêiner; e três modos de comunicação: transmissão *simplex*, recepção *simplex* e *full-duplex*. São transmitidas amostras em fase e quadratura de 16 bits. Para cada modo e cada ambiente, o experimento foi executado em 10 rodadas e os resultados são representados como médias e intervalo de confiança de 95%.

Ao realizar os testes, verifica-se que, para taxas a partir de 33,33 MS/s, a quantidade de erros inviabiliza a comunicação. Desta forma, apresentamos os resultados até o valor limite de 25 MS/s, que coincide com o valor limite estabelecido na documentação do dispositivo para amostras I/Q de 16 bits. A Figura 5 mostra a quantidade de amostras recebidas em cada um dos cenários de recepção. É possível notar que os desempenhos dos três ambientes são muito próximos no modo de recepção *simplex*, Figura 5 (a), já no modo *full-duplex*, Figura 5 (b) é possível observar uma degradação na quantidade de amostras recebidas pelo contêiner nas taxas a partir de 10 MS/s, enquanto a Máquina Virtual apresenta uma degradação mais acentuada a partir dos 16 MS/s. A Figura 5 (c) mostra a quantidade de amostras descartadas pelo receptor em relação a quantidade total de amostras recebidas no modo *full-duplex*. Observa-se que, em algumas execuções

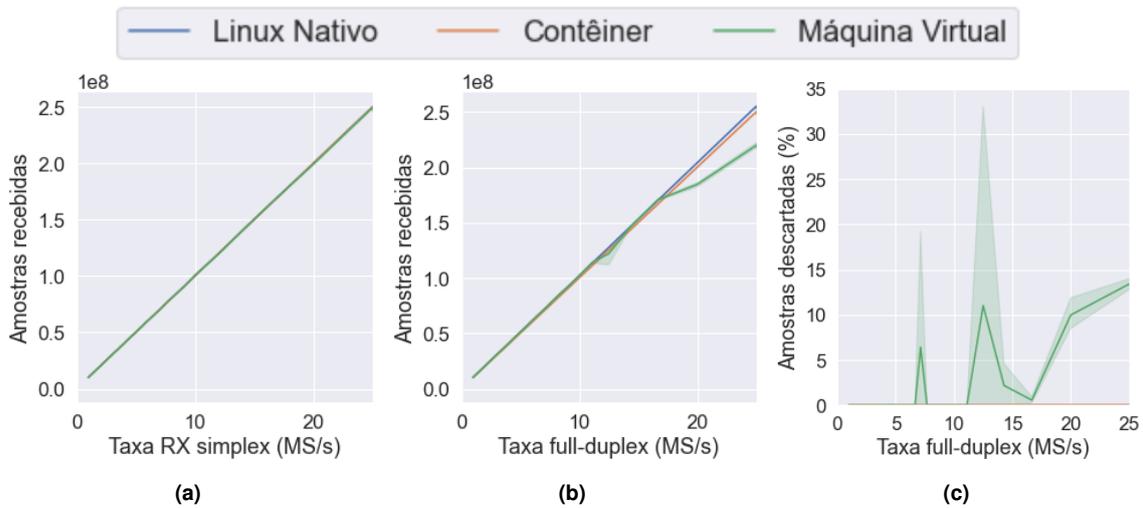


Figura 5. Quantidade de amostras recebidas de acordo com a taxa de recepção: (a) modo de recepção simplex, (b) modo full-duplex, (c) porcentagem de amostras descartadas no modo full-duplex.

do experimento, o descarte de amostras chegou a representar 30% da quantidade total de amostras recebidas pela máquina virtual. Desta forma, é evidente que a virtualização baseada em máquina virtual tem um impacto negativo na quantidade de amostras recebidas em um cenário em que a comunicação realizada é no modo *full-duplex*, o que pode comprometer significativamente o desempenho de aplicações SDR executadas nesse ambiente. Por outro lado, o contêiner obteve resultados bem próximos ao do Linux nativo.

A Figura 6 mostra a quantidade de amostras transmitidas em um cenário com transmissão *simplex* e outro com transmissão *full-duplex*. Assim como na recepção, no modo transmissão *simplex*, todos os três ambientes tiveram resultados semelhantes, porém os testes na máquina virtual apresentaram alguns erros em 10 MS/s, conforme pode ser observado na Figura 6 (a). Já no modo *full-duplex*, a Máquina Virtual obteve um desempenho consideravelmente inferior nos testes de transmissão, apresentando erros em 12 MS/s e nas taxas superiores a 20MS/s, conforme explicitado na Figura 6 (b). A Figura 6 (c) mostra a quantidade de erros *underruns* ocorridos durante os testes no modo *full-duplex*. Esses erros indicam que o computador não está gerando amostras suficientes para atingir a taxa de amostras por segundo estabelecida no teste. No ambiente do contêiner, ocorreram pontualmente erros únicos *underruns* em algumas execuções dos testes em taxas abaixo de 20 MS/s e alguns erros *underruns* mais frequentes em taxas acima de 20 MS/s. Por outro lado, o ambiente de máquina virtual apresenta um crescimento acentuado para taxas acima de 14,29 MS/s, indicando a incapacidade do sistema em operar na taxa selecionada, a taxa nominal de operação seja mais elevada.

Devido à pequena quantidade de erros *underruns* observados na execução do teste no contêiner, é possível concluir que o desempenho obtido nesse ambiente é equivalente ao obtido pelo Linux nativo e não impacta negativamente as aplicações SDR. De maneira oposta, os resultados obtidos no ambiente de máquina virtual indicam um potencial de comprometimento nas aplicações SDR que utilizem taxas acima de 14,29 MS/s.

A Figura 7 mostra o uso de CPU nos diferentes cenários de testes. É possível observar que os ambientes com Linux nativo e contêiner obtiveram resultados similares

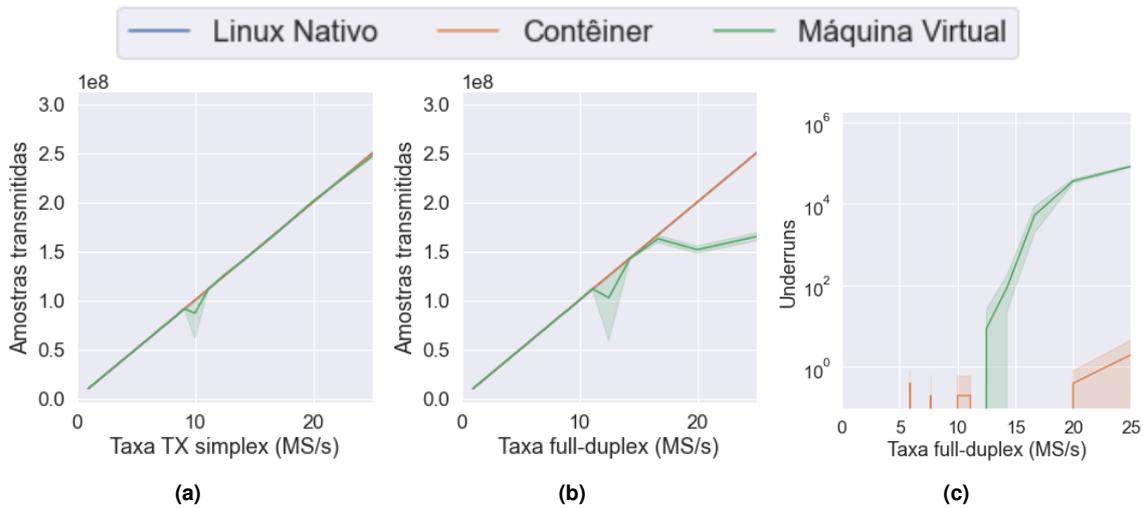


Figura 6. Quantidade de amostras transmitidas de acordo com a taxa de transmissão: (a) modo transmissão simplex, (b) modo full-duplex, (c) quantidade de *underruns* durante o teste full-duplex

em todos os cenários de testes, considerando a variabilidade dos resultados nas diferentes execuções dos testes. Por outro lado, a Máquina Virtual teve o maior consumo de CPU em todos os modos de comunicação, sendo o pior resultado no modo *full-duplex* para taxas maiores que 10 MS/s, com crescimento exponencial para taxas entre 10 e 20 MS/s.

5. Trabalhos Relacionados

Schimid *et al.* estudam o impacto da latência em dispositivos SDR na camada de enlace em dois cenários, o primeiro consiste em um esquema FSK simples e o segundo é baseado no padrão IEEE 802.15.4 [Schmid et al. 2007]. Os autores avaliam o tempo necessário para gerar uma amostra no computador e enviá-la através do USRP e, de forma simétrica, o tempo em que a amostra é recebida pelo USRP até o tempo em que se torna disponível para a camada de enlace no computador. Os testes são realizados com o auxílio de um osciloscópio externo e uma porta paralela do computador. Segundo os autores, a latência final de recepção pode ser calculada como o somatório da latência inserida pelo USRP, pela comunicação USB e pelo software GNU Radio. A latência de transmissão é diretamente afetada pelo buffer de 32 kB existente entre o GNU Radio e a interface USB, de forma que as amostras são transmitidas quando há um número suficiente para preencher um pacote USB. Os resultados obtidos demonstram que as implementações em SDR do protocolo IEEE 802.15.4 possuem uma latência média de 25 ms, considerada alta latência, uma vez que um rádio convencional possui uma latência de 8 ms. Essa alta latência observada inviabiliza a implementação de protocolos com requisitos restritos de tempo.

Becker *et al.* comparam o desempenho de dois arcabouços de SDR, o GNU Radio e o Cognitive Radios Test System version 2.0 (CRTSv.2), em termos de uso de CPU, consumo de memória e latência [Becker et al. 2020]. Bloessl *et al.* implementam o protocolo IEEE 802.11p, que consiste em um padrão de comunicação sem fio entre veículos, utilizando a ferramenta GNU Radio e um dispositivo USRP N210 [Bloessl et al. 2013]. A aplicação desenvolvida suporta modulações BPSK, QPSK, QAM-16 e QAM-64 definidas no padrão, porém não cumpre os requisitos de tempo necessários para a troca de mensa-

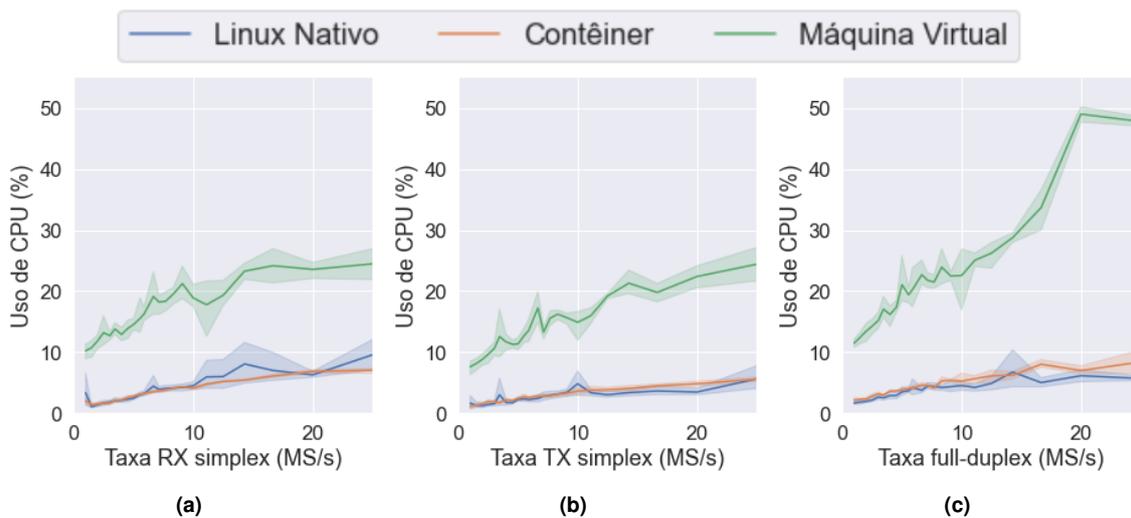


Figura 7. Uso de CPU de acordo com a taxa de transmissão e modo de comunicação: (a) modo de recepção simplex, (b) modo de transmissão simplex, (c) modo full-duplex.

gem de alocação de canal RTS/CTS e para as mensagens ACK que sinalizam o recebimento de quadro. Desta forma, a implementação apresentada se restringe a comunicações de difusão, não sendo capaz de realizar a associação a um ponto de acesso ou comunicação *unicast* com outras estações.

Mattos *et al.* avaliam ferramentas difundidas de virtualização para a implementação de roteadores virtuais em relação ao uso de CPU, memória, rede e disco [Mattos et al. 2012]. Os autores observaram que a ferramenta OpenVZ, baseada em virtualização do espaço de usuário, introduz menor sobrecarga de CPU, disco e memória, porém apresenta uma degradação no desempenho de rede. Por outro lado, a plataforma Xen obteve o melhor resultado na análise de rede, o que é fundamental na utilização de redes virtuais. Os autores evidenciam que os ambientes de virtualização devem ser avaliados de acordo com as aplicações e serviços que serão executados e suas especificidades.

6. Conclusão

Esse artigo avaliou os limites experimentais da comunicação Ethernet de dispositivos SDR e o impacto que a virtualização causa na aplicação. Os experimentos de rede realizados avaliaram a vazão alcançada e o respectivo uso de CPU em um Linux nativo e em dois ambientes virtualizados para diversos tamanhos de pacotes. Os resultados obtidos mostram que o desempenho de rede da virtualização por contêiner é muito próximo ao obtido utilizando um Linux nativo. Por outro lado, a máquina virtual testada apresenta mais erros e mais perdas de pacotes, além de uma menor vazão. A quantidade e a taxa em que é possível transmitir e receber amostras pelo dispositivo USRP foi testada em cada um dos ambientes e em três diferentes cenários de comunicação: transmissão *simplex*, recepção *simplex* e *full-duplex*. Os resultados mostram que, nos modos *simplex*, todos os ambientes possuem comportamentos equivalentes em termos de taxa de transmissão, porém a máquina virtual chega a utilizar mais que o dobro de CPU. No modo *full-duplex*, o desempenho do Linux nativo e do contêiner continuam semelhantes e a máquina virtual apresenta uma piora no desempenho e um consumo ainda maior de uso de CPU. Os resul-

tados dos experimentos demonstram que é viável o uso de máquinas virtuais e contêiner para o uso de SDR, porém a virtualização de ambientes através de máquinas virtuais apresenta uma maior restrição na quantidade de amostras enviadas por segundo, obtendo uma grande sobrecarga de processamento para taxas mais altas. O contêiner obteve um desempenho próximo ao Linux Nativo em todos os cenários, demonstrando atender aos requisitos necessários para uma ampla gama de aplicações de rede de acesso baseadas em SDR na nuvem.

Como trabalho futuro, vislumbra-se o teste de implementações do protocolo IEEE 802.11, avaliando o limite de largura de banda alcançado e as diferenças entre a execução nativa e a dos ambientes virtualizados. Além disso, de acordo com os resultados alcançados, almeja-se propor um sistema que utilize SDR para virtualização de pontos de acesso Wi-Fi alinhado com as iniciativas de redes de acesso de rádio em nuvem.

Referências

- Becker, J. K., Gvozdenovic, S., Xin, L., and Starobinski, D. (2020). Testing and fingerprinting the physical layer of wireless cards with software-defined radios. *Computer Communications*, 160:186–196.
- Bloessl, B., Segata, M., Sommer, C., and Dressler, F. (2013). Towards an open source ieee 802.11 p stack: A full sdr-based transceiver in gnu radio. In *2013 IEEE Vehicular Networking Conference*, pages 143–149. IEEE.
- Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S., and Dittmann, L. (2015). Cloud ran for mobile networks—a technology overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426.
- Dias, W., Ferreira, A., Kagami, R., Ferreira, J. S., Silva, D., and Mendes, L. (2020). 5g-range: A transceiver for remote areas based on software-defined radio. In *2020 European Conference on Networks and Communications*, pages 100–104. IEEE.
- Huber, N., von Quast, M., Hauck, M., and Kounev, S. (2011). Evaluating and modeling virtualization performance overhead for cloud environments. *CLOSER*, 11:563–573.
- Kist, M., Rochol, J., DaSilva, L. A., and Both, C. B. (2018). Sdr virtualization in future mobile networks: Enabling multi-programmable air-interfaces. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Larsen, L. M., Checko, A., and Christiansen, H. L. (2018). A survey of the functional splits proposed for 5g mobile crosshaul networks. *IEEE Communications Surveys & Tutorials*, 21(1):146–172.
- Mattos, D. M., Ferraz, L. H. G., Costa, L. H. M., and Duarte, O. C. M. (2012). Evaluating virtual router performance for a pluralist future internet. In *Proceedings of the 3rd international conference on information and communication systems*, pages 1–7.
- Schmid, T., Sekkat, O., and Srivastava, M. B. (2007). An experimental study of network performance impact of increased latency in software defined radios. In *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 59–66.
- VMware (2005). White paper: Network throughput in a virtual infrastructure. Technical report, VMWARE.