

Alocação de fatias de rede fim-a-fim para usuários móveis utilizando o simulador MobFogSim

Diogo M. Gonçalves¹, Luiz F. Bittencourt¹, Edmundo R. M. Madeira¹

¹Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – São Paulo – Brasil

diogomg@lrc.ic.unicamp.br, {bit, edmundo}@ic.unicamp.br

Abstract. *Network slicing has been presented as one promising technology for resource management in modern networks, e.g., 5G. Based on virtualisation technologies, network slicing creates different virtual networks over one common physical infrastructure. Simulators have been presented as one solution to evaluate new solutions in that context based on scalability, flexibility, and monetary aspects. MobFogSim was proposed to support the evaluation of solutions for resource management in fog computing environments, including service migration, network slicing, and support to users' mobility. This work introduces the new features of MobFogSim in terms of the simulator's scalability and support to end-to-end slices, which includes the management of Fog Nodes' processing and storage resources. Simulations based on realistic parameters show the impact of different slice allocation strategies in the slices' resource allocation as well as the improvements in the simulator's scalability.*

Resumo. *O fatiamento de rede tem se apresentado como uma tecnologia promissora para o gerenciamento de recursos em redes de computadores modernas como o 5G, criando diferentes redes virtuais sobre uma mesma infraestrutura física. Considerando escalabilidade, flexibilidade e aspectos financeiros, simuladores têm se tornado um possível ambiente para avaliar novas soluções desenvolvidas para esse cenário. Nesse contexto, o simulador MobFogSim foi desenvolvido para avaliar soluções em ambientes de Computação em Nuvem e Névoa, incluindo o suporte para migração de serviços, fatiamento de rede e gerenciamento da mobilidade dos usuários. Neste trabalho, apresentamos as novas funcionalidades do simulador em termos de escalabilidade e suporte ao fatiamento ponta-a-ponta, o que inclui recursos de armazenamento e processamento. Simulações considerando dados reais como parâmetros mostraram o impacto de diferentes critérios na alocação de recursos das redes virtuais, além do aumento da escalabilidade do simulador.*

1. Introdução

As redes de telecomunicações constituem uma peça importante para o desenvolvimento e integração de tecnologias para as cidades inteligentes, potencializando soluções que melhoram da qualidade de vida das pessoas. Por outro lado, a já alta e crescente demanda por esses recursos de rede traz grandes desafios. Estima-se que as redes 5G atenderam 580 milhões de dispositivos em 2021 [Ericsson 2021]. Nessas infraestruturas, o tráfego gerado por dispositivos móveis cresceu 46% de 2020 para 2021 e o tráfego gerado somente por celulares superou 10GB em sua média mensal [Ericsson 2021]. Além disso, diferentemente das versões anteriores que se restringiam a atender majoritariamente demandas de transmissão de texto, áudio e vídeo vindos de usuários humanos, as redes 5G

e 6G visam atender também a demanda de dispositivos da Internet das Coisas como, por exemplo, veículos, dispositivos de telemedicina, e máquinas agrícolas e da indústria.

Nesse cenário, a heterogeneidade dos dispositivos, tanto em demandas de rede, como largura de banda e latência, quanto em termos de processamento, armazenamento e confiabilidade, além de peculiaridades em suas características, como mobilidade, tornam mais complexo o gerenciamento dos recursos da rede. A 3GPP sugere uma padronização para o agrupamento desses usuários baseada em suas demandas e características, como URLLC (alta confiabilidade e baixa latência), eMBB (alta largura de banda) e mMTC (alta densidade de dispositivos). Uma solução para este cenário têm sido a utilização de tecnologias de virtualização para flexibilizar e automatizar alterações nas configurações da rede. A partir dessas tecnologias, os recursos de armazenamento, processamento (vindos dos servidores da Nuvem e Névoa) e de rede (como enlaces responsáveis por conectar as entidades da rede) são virtualizados, passando a serem manipulados a partir dessa interface. Nessa arquitetura, diferentes redes virtuais, ou também chamadas de fatias de rede, são criadas sobre uma mesma infraestrutura física. Cada fatia de rede pode ser criada com características diferentes, priorizando determinadas características como latência, largura de banda ou poder de processamento, por exemplo, a fim de suprir a demanda de grupos distintos de usuários. A Figura 1 ilustra o conceito do fatiamento de rede.

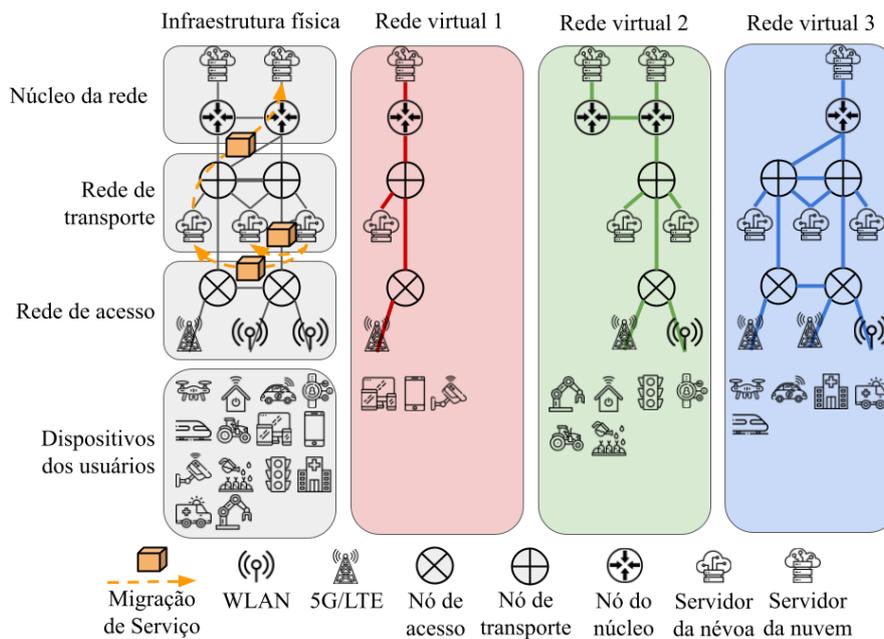


Figura 1. Exemplo de uma arquitetura física com três redes virtuais.

Tendo se apresentado como uma tecnologia promissora, o fatiamento de rede ainda possui diversos tópicos de pesquisa a serem explorados, como a alocação de serviços e funções de rede. Entretanto, para o desenvolvimento, avaliação e validação dessas novas soluções, um ambiente confiável deve ser utilizado para a realização desses testes. Devido a restrições econômicas, o que dificulta a utilização de infraestruturas físicas para este fim, aliada a necessidade de um ambiente flexível e escalável, os simuladores se apresentam como uma possível solução para este prolema.

Visto a falta de um simulador disponível na literatura para suprir essa demanda, em [Puliafito et al. 2020] e [Gonçalves et al. 2020], apresentamos o MobFogSim, um simu-

lador para validar abordagens de gerenciamento de recursos em ambientes de computação em névoa. O simulador possui suporte a mobilidade dos usuários, alocação e migração de serviços (mover um serviço de um servidor para outro), além de um gerenciamento de fatias de rede a nível de recursos de rede. Neste trabalho apresentamos a versão 2.0 do simulador, referente a evoluções relacionadas a (1) inclusão do gerenciamento de recursos de armazenamento e processamento das fatias de rede e (2) escalabilidade do simulador. A partir das atualizações propostas para o simulador, disponibiliza-se agora um ambiente capaz de gerenciar fatias de rede ponta-a-ponta, administrando seus recursos de rede, armazenamento e processamento, desde a rede de acesso (pontos de acesso e nós da névoa), passando pela rede de transporte, até chegar em seu núcleo (nuvem), como exemplificado na Figura 1. Baseado nesses novos cenários, validamos o simulador a partir de simulações baseadas em valores reais para os parâmetros de oferta e demanda de recursos, avaliando o impacto da mobilidade dos usuários, também baseada em dados reais.

O restante deste trabalho está organizado da seguinte forma: A Seção 2 apresenta as características de outros simuladores disponíveis na literatura em comparação ao MobFogSim. A Seção 3 apresenta uma visão geral sobre o MobFogSim e introduz as modificações realizadas no simulador se comparado com sua versão anterior. A Seção 4 discute a metodologia utilizada na validação do simulador enquanto a Seção 5 apresenta os resultados. As conclusões e os trabalhos futuros são apresentados na Seção 6.

2. Trabalhos Relacionados

Nesta seção, é apresentada uma visão geral sobre os simuladores presentes na literatura relacionados ao contexto deste trabalho, apresentando, comparativamente ao simulador MobFogSim. A Tabela 1 apresenta um resumo dos simuladores discutidos. Yet Another Fog Simulator (*YAFS*) [Lera et al. 2019] é um simulador desenvolvido em Python para modelar falhas de rede, sendo elas modeladas de duas maneiras: (i) criando/excluindo dinamicamente nós de névoa e links da rede; (ii) por meio de processos personalizados ou, como eles definem, funções que são invocadas em tempo de execução para a implementação de eventos. O *YAFS* permite modelar a mobilidade dos dispositivos, mas não modela a migração de serviços ou o fatiamento de rede dinâmico.

VirtFogSim [Scarpiniti et al. 2019] não modela migração de serviços nem o fatiamento dinâmico da rede. No entanto, ele rastreia dinamicamente o desempenho do atraso de energia do aplicativo em relação a mudanças ou alterações devido a falhas da rede ou devido a alterações no dispositivo. Por exemplo, alterações induzidas pela variação na largura de banda disponível de *uplink/downlink*. O recurso mais interessante do *VirtFogSim* é a modelagem de acesso à rede celular, útil ao simular cenários para redes 4G/5G.

O foco do simulador *FogNetSim++* [Qayyum et al. 2018] foca na modelagem realista de rede, modelando perdas de pacotes, congestionamento e colisão de canais. O *FogNetSim++* estende o *OMNeT++*¹ para modelar esses aspectos. Apesar de uma modelagem mais realista, o simulador não apresenta suporte para alguns cenários importantes relacionados a computação em névoa e fatiamento de rede, como a migração de serviços e o fatiamento de rede dinâmico, além de permitir modelar a mobilidade dos usuários.

EdgeCloudSim [Sonmez et al. 2018] é um simulador para ambientes de computação em névoa que, como o *iFogSim* [Gupta et al. 2017], estende o *CloudSim*. Os atrasos de rede são modelados no *EdgeCloudSim* de forma mais realista do que no

¹Veja <https://omnetpp.org/>. Último acesso: 26 de Janeiro de 2022.

Tabela 1. Comparativo entre simuladores para névoa e fatiamento de rede

Simulador	Suporte a mobilidade	Migração de Serviços	Fatiamento de rede	Linguagem
YAFS	✓	-	-	Python
VirtFogSim	✓	-	-	MATLAB
FogNetSim++	✓	-	-	C++
EdgeCloudSim	✓	-	-	Java
MobFogSim	✓	✓	✓	Java

iFogSim, onde os atrasos são sempre corrigidos. O EdgeCloudSim atualiza os atrasos da rede com base na carga atual da rede. Porém, o EdgeCloudSim implementa menos funcionalidades do que o iFogSim. Por exemplo, o consumo de energia, os custos operacionais e os preços estão ausentes. A mobilidade do dispositivo é modelada; no entanto, a migração de serviços e o fatiamento de rede dinâmico não estão suportadas pelo simulador.

Nenhum dos simuladores discutidos modela a migração de serviços, o que é importante para dar suporte à mobilidade do usuário. Em [Puliafito et al. 2020], apresentamos o simulador MobFogSim para preencher essa lacuna na literatura. Em [Gonçalves et al. 2020] apresentamos uma extensão ao MobFogSim com a implementação de fatiamento de rede dinâmico, que não é fornecido por nenhum outro simulador de computação em névoa. Assim, o MobFogSim é o único a prover mobilidade, migração de serviços e fatiamento dinâmico de rede. Neste trabalho apresentamos a versão 2.0 do simulador. Nele, apresentamos as atualizações no simulador para incluir o suporte ao fatiamento de rede ponta-a-ponta, incluindo agora aspectos relacionados a gerência de recursos de armazenamento e processamento associados às fatias de rede. Além disso, discutimos e apresentamos as melhorias do simulador em termos de escalabilidade em comparação à sua versão anterior, apresentada em [Gonçalves et al. 2020].

3. MobFogSim

O simulador MobFogSim é baseado nos simuladores iFogSim [Gupta et al. 2017] e MyIFogSim [Lopes et al. 2017], e está disponível como software livre². Em termos de modelagem da infraestrutura de rede, o IFogSim permite modelar dispositivos IoT, pontos de acesso para conexão sem fio entre usuário e arquitetura e servidores da nuvem e névoa. Estendendo os simuladores em que se baseia, o MobFogSim permite o suporte à mobilidade dos usuários, implementando mecanismos de gerenciamento de recursos relativos a essa mobilidade, como a migração de serviços entre os servidores da rede. Com base na mobilidade dos usuários, o serviço acessado por eles que está rodando em um servidor da nuvem/névoa pode ser movido estrategicamente entre os nós/servidores a fim de melhorar alguns parâmetros de Qualidade de Serviço como, por exemplo, latência. O serviço, que está sendo executado sobre uma máquina virtual ou um contêiner, pode ser migrado utilizando as técnicas de migração *cold* ou *live*. A migração *cold* interrompe a execução do serviço enquanto a transmissão dos dados ocorre. Por outro lado, a migração *live* interrompe a execução apenas por uma fração de tempo estritamente necessária. A avaliação de desempenho dessas técnicas de migração pode ser vista com mais detalhes em [Puliafito et al. 2020]. O critério para iniciar uma migração de serviço, definido no simulador como estratégia de migração, pode ser baseados na busca pela menor

²Veja <https://github.com/diogomg/MobFogSim>. Acessado em 30/01/2022

latência possível oferecida pela infraestrutura ou pela menor distância física entre usuário e serviço. O simulador, entretanto, é flexível para adotar outras estratégias de migração.

O suporte ao fatiamento de rede dinâmico foi apresentado em [Gonçalves et al. 2020]. Nesse trabalho, a implementação e avaliação da fatia de rede ficaram restritas aos recursos da rede, como a distribuição de largura de banda. As fatias de rede no simulador podem compartilhar a mesma infraestrutura física. No entanto, cada fatia de rede tem acesso a apenas uma fração desses recursos. O simulador suporta uma realocação de recursos entre essas fatias de rede mas o cenário avaliado no trabalho citado ficou restrito apenas à realocação de recursos ociosos.

Para suportar a mobilidade realista dos usuários, o MobFogSim possui integração com bases de dados reais interpretáveis por outros simuladores de mobilidade, como Simulação de Mobilidade Urbana (SUMO) [Behrisch et al. 2011]. Utilizando as saídas do SUMO como entrada do MobFogSim, é possível avaliar a correlação entre a mobilidade desses usuários e seus impactos no desempenho da rede.

Com base na mobilidade do usuário, a infraestrutura atende a essa demanda dinâmica para colocar e migrar Serviços e criar e reconfigurar as fatias de rede. No MobFogSim, o processo de migração utiliza como entrada os recursos de rede na forma de redes virtuais (servidores e links), os dispositivos do usuário e a estratégia de migração escolhida. Uma representação lógica da arquitetura do simulador, bem como sua relação com as classes herdadas do simulador iFogSim, é apresentada na Seção 3.1. A descrição das políticas de alocação das redes virtuais é apresentada na Seção 3.2.

3.1. Arquitetura lógica e suas novas funcionalidades

O MobFogSim, sendo baseado no simulador iFogSim [Gupta et al. 2017] (baseado no simulador de nuvem CloudSim), herda classes e características desses simuladores. A arquitetura lógica do MobFogSim pode ser classificada em três grandes pacotes: (1) gerenciamento de recursos em nuvem, como recursos de armazenamento, processamento e virtualização (2) gerenciamento de nós da névoa, como criação e manipulação de micro data centers, e criação de sensores e atuadores, e (3) geolocalização dos recursos, gerenciamento de dispositivos móveis, migração de serviços e criação e manipulação de fatias de rede (redes virtuais). A Figura 2 ilustra os módulos do simulador MobFogSim em perspectiva às classes herdadas dos simuladores iFogSim e CloudSim.

Desses três grandes pacotes, em (1) encontram-se as classes herdadas do simulador CloudSim, responsável pelo recursos em nuvem, em (2) encontra-se os módulos herdados do simulador iFogSim, que estende o simulador CloudSim ao introduzir a criação e manipulação de recursos na borda da rede. Finalmente, em (3) encontra-se os módulos característicos do simulador MobFogSim, que estende o simulador iFogSim, sendo responsável, dentre outras coisas, pela criação de um mapa 2D com as entidades simuladas, como nós da névoa e nuvem, pontos de acesso, e usuários móveis, além do gerenciamento dos recursos da rede como resposta à mobilidade dos usuários. Em especial, o módulo *Slice*, responsável pelo gerenciamento das redes virtuais, é um dos tópicos de contribuição deste artigo. Além da adição do módulo *Slice*, outros módulos já existentes na versão 1.0 do simulador foram modificados para atender a nova funcionalidade, sendo eles, destacados em cinza escuro na figura, os módulos *FogDevice*, *MobileDevice* e *APDevice*.

A criação das redes virtuais (fatias de rede) é de responsabilidade do módulo *Slice*. Dado um conjunto de nós de névoa/nuvem, com suas respectivas ofertas de recursos (armazenamento e processamento) vindos do módulo *FogDevice*, e enlaces e pontos

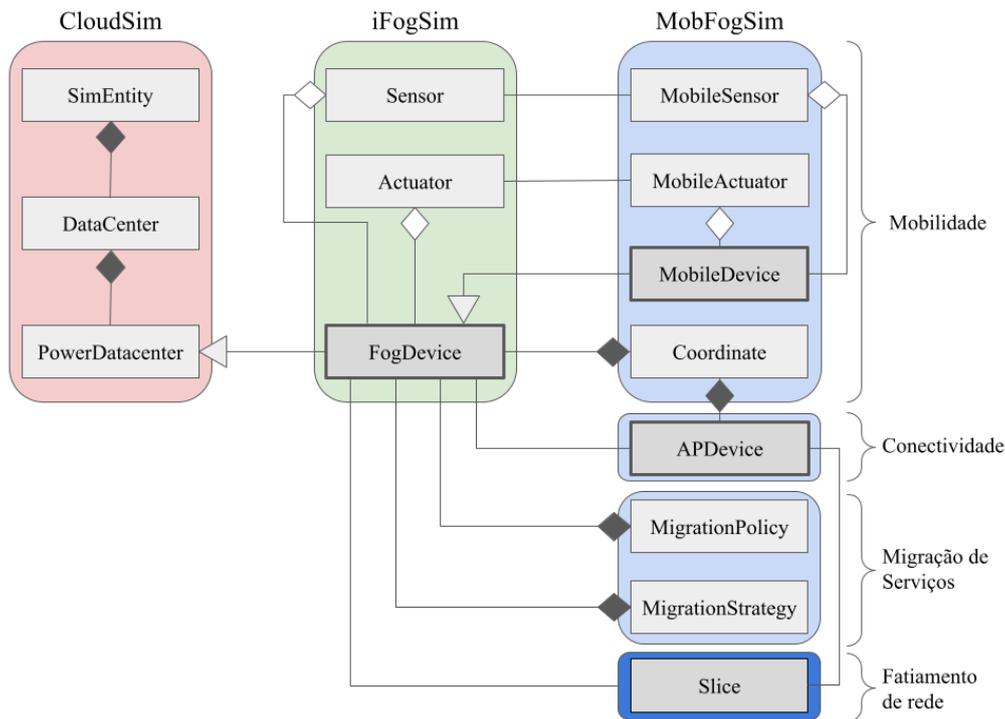


Figura 2. Arquitetura lógica do simulador MobFogSim em comparação aos simuladores CloudSim e iFogSim.

de acesso com suas ofertas de largura de banda e latência, vindos do módulo *APDevice*, o módulo *Slice* define a quantidade de recursos da rede, proporcional ou em números absolutos, para cada fatia de rede. Nesta versão do simulador, os módulos *FogDevice* e *APDevice* passaram a fragmentar os seus recursos, os tornando distribuíveis entre rede virtuais distintas, enquanto o módulo *MobileDevice* passou a discriminar cada usuário, permitindo o agrupamento de usuários por meio de seu identificador.

Cada rede virtual pode atender um ou mais grupos de usuários. Cada grupo de usuários, representados pelos dispositivos da classe *MobileDevice*, possui características e demandas distintas. Baseadas nessas diferentes demandas, como largura de banda, latência ou poder de processamento, as fatias de rede requisitam os recursos necessários para atender seus usuários. Cada *MobileDevice* possui um atributo *groupId* que o identifica como parte de um grupo de usuários em comum. Com base nesse identificador, esse usuário possui acesso a uma rede virtual.

A alocação dos recursos de rede (largura de banda) é feita por meio de uma reserva de tráfego. Cada fatia de rede reserva uma quantidade de largura de banda de um conjunto de enlaces. Esta reserva é definida como a quantidade de pacotes desta rede virtual que possuem o direito de trafegar por aquele enlace. Em outras palavras, uma rede virtual que possui 30% da largura de banda de um determinado enlace, terá direito a enviar até 30% dos pacotes transmitidos por aquele enlace. Esta abordagem garante uma reserva de recursos para cada rede virtual, porém não otimiza o uso de recursos ociosos entre as redes. Uma solução, suportada tanto pelo conceito de fatiamento de rede quanto pelo simulador é a realocação dinâmica de redes. Uma avaliação do impacto dessa abordagem é apresentada em [Gonçalves et al. 2020] e em [Gonçalves et al. 2021] e está fora do escopo deste trabalho. Por sua vez, a alocação dos recursos de névoa para cada

rede virtual é feita por meio de reserva de recursos. Cada rede virtual possui uma lista contendo a quantidade de recursos de cada nó da névoa atribuídos para ela. A partir dessa quantidade de recursos de armazenamento e processamento nesta lista, a infraestrutura pode solicitar a alocação/migração de um serviço que está sendo acessado por um usuário (*MobileDevice*). Tais recursos são reservados para uso exclusivo dos usuários desta rede virtual. A localização desse serviço pode ser definida por diferentes critérios a depender da demanda do usuário como, por exemplo, minimizar a latência.

Detalhes sobre as políticas de alocação das redes virtuais, bem como a descrição das etapas desse processo, são apresentados na Seção 3.2. Detalhes acerca do gerenciamento das máquinas virtuais/containers, bem como a caracterização dos critérios para alocação e migração dos serviços executados sobre essas máquinas virtuais, podem ser encontradas em [Puliafito et al. 2020]. O simulador também possui suporte para o gerenciamento dinâmico das redes virtuais no que diz respeito a realocação de recursos para atender variações de demanda de rede. Tais detalhes estão fora do escopo deste trabalho, mas podem ser encontrados em [Gonçalves et al. 2020] e [Gonçalves et al. 2021].

3.2. Políticas de alocação de recursos

As política de alocação de recursos no processo de fatiamento da rede, ou seja, a criação de diferentes redes virtuais sobre uma mesma infraestrutura física em comum, tem um papel importante na distribuição e no desempenho das mesmas. Dada a porção finita dos recursos de armazenamento, processamento e rede disponíveis na rede, a ordem e quantidade em que esses recursos serão distribuídos deve ser gerenciada pela infraestrutura. Como descrito na Seção 3.1, cada rede virtual servirá um grupo de usuários, tendo eles suas demandas específicas para cada recurso. Assim, as redes virtuais podem ter requisitos diferentes a serem atendidos pela infraestrutura. Cada rede virtual também pode ser atendida com prioridades diferentes pela infraestrutura. Redes prioritárias podem ser alocadas antecipadamente. A partir das demandas das redes virtuais, a infraestrutura pode repartir os seus recursos com base em diferentes políticas.

A política de alocação das redes virtuais adotada nesse trabalho assume a presença de redes prioritárias, colocando elas nas primeiras posições da fila de alocação. Uma vez alcançada sua vez na fila, a infraestrutura tenta alocar os nós da névoa e os enlaces da rede que melhor atendem os requisitos da rede. Assumindo uma alocação sequencial e prioritária, as redes menos prioritárias podem ser injustiçadas nesse processo de alocação. Outras políticas de alocação de Fatias de Redes, como uma alocação coordenada, ou outros algoritmos de alocação como, por exemplo, baseadas em aprendizado de máquina, podem ser implementados no simulador. O Algoritmo 1 apresenta uma visão em alto nível do processo de alocação dessas redes no simulador.

O processo de alocação das fatias de rede se dá no início da simulação. O simulador possui suporte para realizar alterações nos recursos alocados para cada rede virtual, como apresentado em [Gonçalves et al. 2020] e [Gonçalves et al. 2021], porém este tópico não está no escopo deste trabalho. Para a alocação das redes, o Algoritmo 1 recebe como entrada (1) o conjunto *nos_nevoa* de nós da névoa, ou seja, servidores responsáveis pelos recursos de armazenamento e processamento a serem utilizados pelos serviços dos usuários, (2) o conjunto *enlaces* relacionado aos enlaces presentes na rede, responsáveis por conectar as entidades da infraestrutura (nós da névoa e pontos de acesso), e (3) o conjunto *grupo_usuarios* que possui os diferentes grupos de usuários com suas respectivas características e demandas. Como saída desse algoritmo, recebe-se as redes virtuais devidamente alocadas.

Algoritmo 1: Processo de alocação das fatias de rede.

```
input : nos_nevoa, enlaces, grupo_usuarios
output: redes_virtuais
1 redes_virtuais  $\leftarrow$   $\emptyset$ ;
2 grupo_usuarios  $\leftarrow$  OrdenaGruposPorPrioridade(grupo_usuarios);
3 for  $u \in$  grupo_usuarios do
4   nos_da_rv, enlaces_da_rv  $\leftarrow$   $\emptyset$ ;
5   nos_da_rv  $\leftarrow$  SeleccionaNos(nos_nevoa, u.requisitosComputacao);
6   if nos_da_rv  $\neq$   $\emptyset$  then
7     enlaces_da_rv  $\leftarrow$  SeleccionaEnlaces(nos_da_rv, enlaces,
      u.requisitosRede);
8     if enlaces_da_rv  $\neq$   $\emptyset$  then
9       redes_virtuais  $\leftarrow$  redes_virtuais  $\cup$  alocaRedeVirtual(nos_da_rv,
      enlaces, u.requisitosRede);
```

A partir dos grupos de usuários ordenados por sua prioridade (linha 2), o algoritmo inicia seu processo de alocação. Para cada grupo de usuários u , o algoritmo primeiramente seleciona o conjunto de nós da névoa, presentes na infraestrutura física, que atendem as demandas de armazenamento e processamento do grupo de usuários u (linha 5). Caso haja um conjunto de nós que satisfaça tal demanda (linha 6), o algoritmo tenta, então, alocar o conjunto de enlaces para conectar esses nós, também respeitando a demanda de largura de banda e latência do grupo u (linha 7). Se ambos, nós e enlaces, satisfazem os requisitos do grupo de usuários u , o algoritmo finalmente realiza a alocação da rede virtual, removendo então, esses recursos do conjunto de recursos disponíveis.

4. Metodologia de avaliação

A partir das modificações implementadas no simulador MobFogSim, uma parte relevante para a consolidação de processo é a validação do mesmo. Para a validação do simulador, foram elaborados alguns cenários tipicamente encontrados em cidades inteligentes. Os cenários foram configurados para simular o comportamento da rede em um período de 60 minutos, cada um contendo 80 usuários. A arquitetura física definida nas simulações foi posicionada sobre um mapa de 5km x 5km, onde os usuários possuem a liberdade para se locomover. A fim de aumentar a confiabilidade dos cenários, foram utilizados dados reais como entrada para definição dos valores dos parâmetros utilizados nas simulações.

A arquitetura de rede é composta por um conjunto de 16 servidores e 64 pontos de acesso posicionados na borda da rede. Metade das conexões entre os servidores possui uma largura de banda de 74Mbps e a outra metade 148Mbps, e uma latência de 3.47ms. Os servidores possuem uma capacidade de processamento 4.6 BIPS (bilhões de instruções por segundo) e 1024 MB de memória RAM, enquanto hospedam um serviço que demanda 0.29 BIPS e 128 MB de memória RAM. Os valores assumidos para a oferta de demanda de recursos foram baseados em uma infraestrutura física real apresentada em [Puliafito et al. 2020]. A tabela 2 apresenta os parâmetros e seus valores utilizados neste estudo. Em complemento aos parâmetros da infraestrutura de rede, os dados utilizados como mobilidade dos usuários foram extraídos da base de dados pública de veículos de Luxemburgo, disponibilizados por [Codeca et al. 2015]. Foram selecionados 2400 veículos dessa base de dados. Três redes virtuais foram criadas, cada uma priorizando um aspecto da rede: Rede Virtual 1, priorizando a capacidade de processamento e memória

RAM, Rede Virtual 2, priorizando a largura de banda e Rede Virtual 3, priorizando a latência. As redes virtuais foram alocadas nesta ordem.

Tabela 2. Parâmetros e seus valores utilizados nas simulações. Valores baseados na infraestrutura real apresentada em [Puliafito et al. 2020]

Parâmetro	Valor
Demanda de processamento do usuário	2901MIPS
Tamanho dos pacotes de requisição	87B
Tamanho dos pacotes de resposta	54B
Uso de armazenamento do usuário	4MB
Uso de armazenamento do servidor	412MB
Capacidade de processamento do usuário	46534MIPS
Capacidade de processamento do servidor	3234MIPS
Latência entre usuário e servidor via Ponto de Acesso	4.78ms
Latência média entre os servidores	3.47ms
Demanda de RAM no usuário	49MB
Demanda de RAM no servidor	128MB
Largura de banda entre usuário e servidor	13.6Mbps
Largura de banda entre os servidores	74.1 ou 148.2Mbps

Dentre as modificações realizadas no simulador, também inclui-se modificações visando melhorias em termos de escalabilidade. A avaliação de desempenho da nova versão do simulador foi realizada em perspectiva a sua versão anterior publicada em [Gonçalves et al. 2020]. Os resultados foram avaliados com base nas métricas de tempo total de execução, medido em minutos, e o consumo máximo de memória RAM do computador que executava o simulador, medida em Gigabytes. Foram realizados três testes utilizando diferentes cargas ao simulador no que se diz respeito ao número de usuários, contendo 20, 50 e 80 usuários por simulação para avaliação da versão 2.0 do simulador, e 20 e 50 usuários para a versão 1.0 do simulador. Por motivos de limitações de consumo de tempo e recursos computacionais, não foi realizada a avaliação contendo 80 usuários para a versão 1.0 do simulador. Os resultados de cada cenário são apresentados como a média dentre 30 execuções do simulador, considerando um intervalo de confiança de 95%. Nos gráficos, a versão 1.0 do simulador está representada pelas linhas vermelhas enquanto a versão 2.0 é apresentada em linhas azuis. A máquina utilizada para a execução dos testes contém um processador de 14 núcleos Intel(R) Xeon(R) CPU E5-2660 v4, com uma frequência de operação entre 2.0 e 3.2GHz, e 226 GB de memória RAM. Os resultados são apresentados na Seção 5.2.

5. Resultados

A partir dos cenários de avaliação descritos na Seção 4, foram executadas as simulações propostas, tendo seus resultados apresentados a seguir. A Seção 5.1 apresenta os resultados referentes aos cenários de alocação de redes virtuais. A Seção 5.2 apresenta o desempenho da versão 2.0 do simulador MobFogSim em termos de escalabilidade e consumo de recursos se comparado com sua versão anterior. Os resultados representam uma média dos valores obtidos, respeitando um intervalo de confiança de 95%.

5.1. Gerenciamento de recursos

A partir das modificações realizadas no simulador a fim de incorporar recursos de névoa, como armazenamento e processamento, às redes virtuais, assim como descrito

nas políticas de alocação de recursos da Seção 3.2 e nos cenários de simulação apresentados na Seção 4, o desempenho de três redes virtuais com base em suas prioridades foi avaliado e é apresentado nesta seção. Nesta avaliação, as três redes virtuais competem pelos recursos da infraestrutura, resultando na alocação de uma porção deles para cada uma delas. A Figura 3 apresenta um exemplo de alocação dos nós da névoa entre as redes virtuais. Na figura, nota-se que a rede virtual 1 (vermelha) possui acesso a 100% os recursos de névoa dos nós alocados para ela, enquanto as redes virtuais 2 (azul) e 3 (verde) dividem os mesmos recursos físicos, apesar de isoladas virtualmente. A rede virtual 1, ao priorizar recursos de armazenamento e processamento, requisitou e foi atendida, com mais nós de névoa que as demais redes virtuais a fim de suprir sua demanda. Para atender a tal demanda da rede virtual 1, os nós selecionados para essa fatia de rede não dividem seus recursos com as demais redes. Com uma demanda menor por recursos de névoa, as redes virtuais 1 e 2 receberam porções menores desses recursos. Por outro lado, tais fatias requisitaram mais recursos de rede, podendo selecionar os enlaces que melhor as atendessem em termos de largura de banda (rede virtual 2) e latência (rede virtual 3).

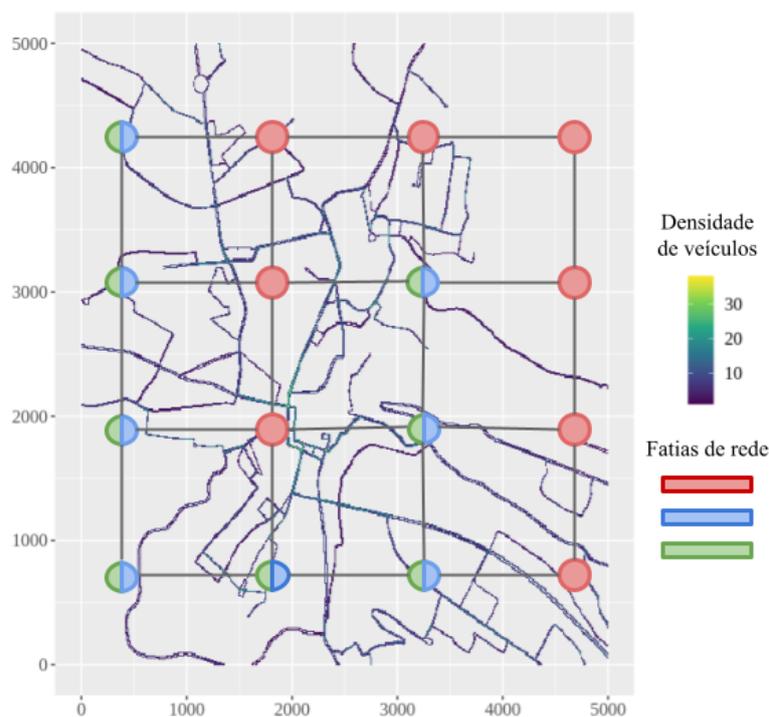


Figura 3. Exemplo de alocação de uma fatia de rede em uma instância simulada.

Com base na alocação das redes virtuais, como apresentado na Figura 3, um conjunto de simulações foi realizado para apresentar uma média de desempenho das três redes virtuais analisadas. A Figura 4 resume os resultados obtidos. A figura apresenta a quantidade de recursos de névoa e de rede alocados para cada rede virtual e, conseqüentemente, o seu desempenho, em termos de número de migrações de serviços realizados, seu tempo médio para realizar cada migração e, por fim, a latência apresentada pelo serviço.

A rede virtual com maior demanda por processamento e memória RAM (rede virtual 1), ilustrada pelas barras vermelhas na Figura 4, apresentou, como esperado, o direito a acessar mais recursos de névoa, como apresentado na Figura 4 (a). Como consequência, a rede virtual 2, ilustrada pelas barras verdes, que prioriza largura de banda e a rede vir-

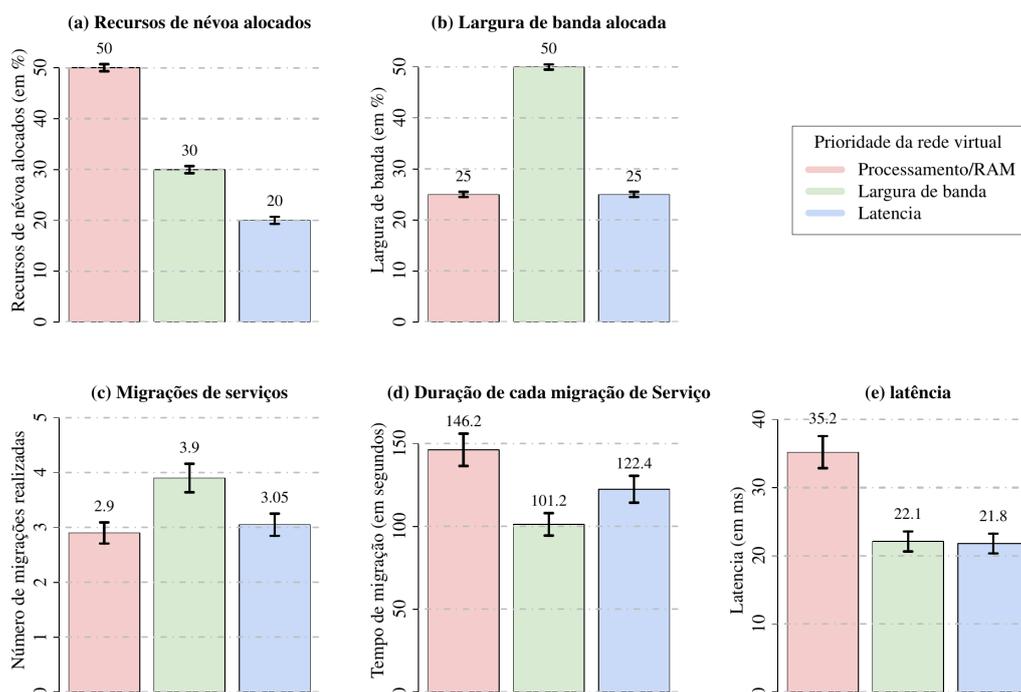


Figura 4. Desempenho das fatias de rede.

tual 3, ilustrada pelas barras azuis, que prioriza uma menor latência, receberam porções menores dos recursos da névoa. Por outro lado, a rede virtual 2 (vermelha), teve direito a alocar a maior porção da largura de banda disponível na rede, como ilustrado na Figura 4 (b), enquanto as demais receberam porções menores. Essa distribuição desproporcional dos recursos é resultado tanto da prioridade de cada fatia de rede, tanto da demanda de cada fatia de rede (Algoritmo 1 nas linhas 5 e 7), quando da prioridade/ordem para alocação de cada fatia de rede (Algoritmo 1 na linha 2).

Como consequência do acesso a diferentes recursos por parte de cada rede virtual, cada fatia de rede apresenta um desempenho distinto para a alocação dos serviços dos seus usuários. Como consequência direta da largura de banda disponível para cada rede virtual, o número de migrações de serviços realizados por cada fatia de rede, ilustrada na Figura 4 (c), bem como a duração de cada processo de migração, apresentado na Figura 4 (d) possuem desempenhos distintos. Nota-se que um maior acesso a recursos de névoa não impactou de forma relevante positivamente o desempenho da migração de serviços para a rede virtual 1, bem como a construção de uma rede virtual priorizando a latência (rede virtual 3). Como esperado, a rede virtual 2, que prioriza a largura de banda, apresentou os melhores desempenhos em termos de tempo de duração para realizar o processo de migração, Figura 4 (d), e número médio de migrações por usuário, Figura 4 (c).

A partir das migrações de serviços, a infraestrutura possui a capacidade de reposicionar os serviços acessados pelos usuários de forma a melhorar seu desempenho. Quanto mais próximo do usuário o serviço está alocado, espera-se menores taxas de latência na entrega dos seus pacotes. A Figura 4 (e) apresenta o desempenho das redes virtuais em termos de latência apresentada na troca de pacotes entre o usuário e seu serviço. Em geral, tanto a rede virtual 3, representada pela cor azul, quanto a rede virtual 2, verde, apresentaram os melhores resultados para esta métrica. Pode-se concluir que selecionar os enlaces com a menor latência ou os enlaces com a maior largura de banda, resulta em

uma diminuição na latência experimentada pelo usuário. Apesar de não necessariamente selecionar os enlaces com a menor latência, a rede virtual 2, ao proporcionar a maior largura de banda para seus usuários, permite um número satisfatório de migrações, mantendo a aplicação próxima ao seu usuário, o que também resulta em bons resultados em termos de latência. Por fim, nota-se que um maior poder de processamento e armazenamento não necessariamente assegura bons resultados para os usuários em termos de latência, uma vez que esta rede virtual nem utiliza os melhores enlaces em termos de latência tão pouco proporciona um número satisfatório de migrações.

5.2. Desempenho do simulador

Assim como descrito nas seções 3.1 e 4, as melhorias na escalabilidade do simulador constituem uma parte importante nas contribuições dessa nova versão do simulador. Como discutido anteriormente, a escalabilidade do simulador foi avaliada com base na sua capacidade de executar cenários com um número crescente de usuários. A Figura 5 apresenta os resultados obtidos para a avaliação do simulador. A primeira métrica avaliada diz respeito ao tempo necessário para a execução completa das simulações, desde a alocação dos recursos até a apresentação dos resultados.

A Figura 5 (a) apresenta os resultados para essa métrica. Na figura, nota-se um desempenho consideravelmente melhor da versão 2.0 do simulador. Para um cenário com 20 usuários, a versão 1.0 do simulador apresentou um tempo total de execução de, em média, 121 minutos, enquanto que na versão 2.0 este tempo foi reduzido para 48 minutos, representando uma diminuição em 60%. Para o cenário 2.5 vezes maior no número de usuários, passando de 20 para 50 usuários, a versão 1.0 teve um desempenho de cerca de 2.8 vezes pior, passando para 342 minutos de execução, em média. Por outro lado, a versão 2.0 do simulador teve um aumento no tempo de execução na ordem de 2.25 vezes, passando para 108 minutos. Para o cenário com 80 usuários, a versão 2.0 apresentou um tempo de execução de cerca de 180 minutos, em média. Com base nessa métrica, nota-se uma significativa melhora na escalabilidade do simulador, permitindo não somente uma diminuição no tempo de execução se comparado o mesmo número de usuários, mas também permitirá uma avaliação de cenários com número maior de usuários.

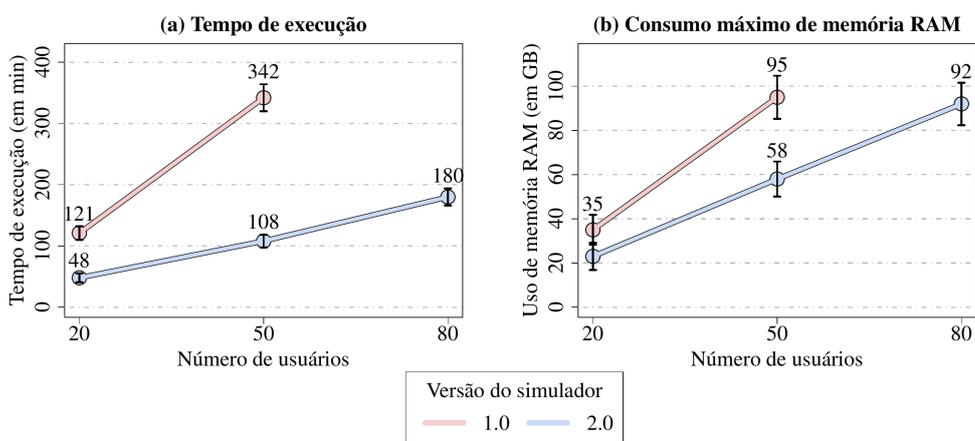


Figura 5. Desempenho do simulador em termos de tempo de execução e consumo de memória RAM.

Outra métrica considerada nessa avaliação diz respeito ao consumo máximo de memória RAM. Um alto consumo de tempo para a execução de uma simulação não necessariamente é um limitante para a viabilidade do simulador. Por outro lado, um alto

consumo de memória RAM pode limitar o número de máquinas capazes de executar o simulador. Dentre as modificações para a melhoria da escalabilidade do simulador, não somente o tempo de execução foi reduzido como também o consumo total de memória RAM. A Figura 5 (b) apresenta os resultados para essa métrica. Na figura, nota-se a superioridade da versão 2.0 do simulador frente a sua versão anterior. O consumo máximo de memória RAM em simulações com 20 usuários foi de em média 35GB para a versão 1.0 enquanto a versão 2.0 consumiu 65% deste valor, 23GB. Para cenários com 50 usuários, a versão 1.0 teve um aumento de 2.7 vezes, subindo para 95GB, enquanto a versão 2.0 teve um aumento de 2.5, subindo para 58GB. A versão 2.0 ainda apresentou um consumo de 92GB de memória RAM para o cenário com 80 usuários. Nota-se que, para patamares similares de consumo, próximos a 92GB, conseguiu-se um aumento de 60% no número de usuários entre as versões, passando de 50 na versão 1.0 para 80 na versão 2.0.

Com a versão 2.0 do MobFogSim, apresentada neste trabalho, pode-se incorporar melhorias ao simulador tanto em termos de função, com a inclusão dos recursos de armazenamento e processamento na criação e manutenção das fatias de rede, discutido na Seção 5.1, como também uma melhoria em termos de escalabilidade do simulador, discutida na Seção 5.2. Assim, o simulador passa a suportar novos cenários de avaliação, seja pela função adquirida, seja pelo aumento no número de usuários em uma simulação.

6. Conclusão

O fatiamento de rede tem se mostrado como uma tecnologia eficiente para gerenciar recursos de rede, em especial, devido à sua capacidade de criar redes virtuais especialmente moldadas para atender usuários com diferentes características e demandas. Apesar de seus benefícios, o fatiamento de rede ainda encontra-se em um estágio de desenvolvimento, possuindo diversos tópicos de pesquisa a serem explorados. Devido a limitações econômicas e de flexibilidade e produtividade, simuladores têm sido utilizados como alternativas a infraestruturas de rede reais no desenvolvimento dessas tecnologias.

Nesse sentido, este artigo apresenta uma extensão do simulador MobFogSim, adicionando a ele a capacidade de gerenciar recursos de armazenamento e processamento das fatias de rede provindos dos nós da névoa e nuvem, complementando o gerenciamento de recursos de rede de sua versão anterior. Além disso, a nova versão do simulador também apresenta melhorias em termos de escalabilidade, permitindo simular uma gama maior de cenários, unindo o fatiamento de rede, mobilidade de usuários, alocação e migração de serviços em ambientes de computação em névoa. Com base nessas características, o MobFogSim se apresenta como o único simulador a oferecer suporte a esses cenários.

Simulações apontaram como o desempenho de diferentes redes virtuais é impactado pela sua política de alocação e sua prioridade na ordem de alocação. Além disso, atualizações no simulador também permitiram um aumento na escalabilidade do software, com uma diminuição de até 65% em termos de tempo de simulação e 60% no consumo máximo de memória RAM. A partir deste trabalho, disponibiliza-se um software aberto para a comunidade científica e da indústria. Como trabalhos futuros, espera-se desenvolver mecanismos mais sofisticados, baseados em aprendizado de máquina, para a alocação de redes virtuais, considerando também variações na demanda, combinações de cenários e diferentes políticas de alocação.

Agradecimentos

Este trabalho é parte do INCT sobre Internet do Futuro para Cidades Inteligentes (CNPq 465446/2014-0, CAPES 88887.136422/2017-00 e FAPESP 2014/50937-1). O presente

trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo nº 2021/05253-0.

Referências

- Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). SUMO - Simulation of Urban Mobility: An Overview. In *3rd International Conference on Advances in System Simulation (SIMUL)*.
- Codeca, L., Frank, R., and Engel, T. (2015). Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research. In *IEEE Conference on Vehicular Networking (VNC)*, pages 1–8.
- Ericsson (June 2021). Ericsson mobility report. www.ericsson.com/en/mobility-report.
- Gonçalves, D., Puliafito, C., Mingozzi, E., Rana, O., Bittencourt, L., and Madeira, E. (2020). Dynamic network slicing in fog computing for mobile users in mobfogsim. In *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, pages 237–246. IEEE.
- Gonçalves, D. M., Bittencourt, L. F., and Madeira, E. R. (2021). Fatiamento dinâmico de redes em computação em névoa para usuários móveis. In *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 57–70. SBC.
- Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., and Buyya, R. (2017). iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in the Internet of Things, Edge and Fog Computing Environments. *Software: Practice and Experience*, 47(9):1275–1296.
- Lera, I., Guerrero, C., and Juiz, C. (2019). YAFS: A Simulator for IoT Scenarios in Fog Computing. *IEEE Access*, 7:91745–91758.
- Lopes, M. M., Higashino, W. A., Capretz, M. A., and Bittencourt, L. F. (2017). Myi-FogSim: A Simulator for Virtual Machine Migration in Fog Computing. In *ACM 6th International Workshop on Clouds and (eScience) Applications Management (CloudAM). Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*, pages 47–52.
- Puliafito, C., Gonçalves, D. M., Lopes, M. M., Martins, L. L., Madeira, E., Mingozzi, E., Rana, O., and Bittencourt, L. F. (2020). Mobfogsim: Simulation of mobility and migration for fog computing. *Simulation Modelling Practice and Theory*, 101:102062.
- Qayyum, T., Malik, A. W., Khattak, M. A. K., Khalid, O., and Khan, S. U. (2018). Fog-NetSim++: A Toolkit for Modelling and Simulation of Distributed Fog Environment. *IEEE Access*, 6:63570–63583.
- Scarpiniti, M., Baccarelli, E., and Momenzadeh, A. (2019). VirtFogSim: A Parallel Toolbox for Dynamic Energy-Delay Performance Testing and Optimization of 5G Mobile-Fog-Cloud Virtualized Platforms. *MDPI Applied Sciences*, 9(6).
- Sonmez, C., Ozgovde, A., and Ersoy, C. (2018). EdgeCloudSim: An Environment for Performance Evaluation of Edge Computing Systems. *Transactions on Emerging Telecommunications Technologies*, 29(11).