Orquestração Multi-Critério de Funções de Serviço em Redes Móveis de Borda para Realidade Aumentada Multiusuário

Rodrigo Flexa[1], **Hugo Santos**[2], **Eduardo Cerqueira**[1], **Denis Rosário**[1]

[1]Universidade Federal do Pará (UFPA) – Belém – PA – Brazil

[2]Universidade Federal Rural da Amazônia (UFRA) – Belém – PA – Brazil

rodrigo.flexa@itec.ufpa.br, {cerqueira,denis}@ufpa.br

hugo.santos@ufra.edu.br

Abstract. *The increasing connection of devices to the internet intensifies the use of network resources, highlighting Collaborative Augmented Reality (RAMU), which combines virtual elements with the real environment, providing an immersive experience. This service can be divided into Service Function Chains (SFCs) and distributed among edge servers, allowing parallel execution and efficient resource sharing, mitigating scalability and consistency limitations. This work proposes the Multi-Criteria and Mobility-Aware Service Function Chain Orchestration (OSFEM), which uses a heuristic to enhance resource efficiency and quality of service (QoS) in mobile scenarios.*

Resumo. *A crescente conexão de dispositivos na internet intensifica o uso de recursos de rede, destacando a Realidade Aumentada Multiusuário (RAMU), que combina elementos virtuais com o ambiente real, proporcionando uma experiência imersiva. Este serviço pode ser dividido em Cadeias de Funções de Serviço (SFCs) e distribuído entre servidores de borda, permitindo a execução paralela e o compartilhamento eficiente de recursos, mitigando limitações de escalabilidade e consistência. Este trabalho propõe a Orquestração de Encadeamento de Funções de Serviço de Múltiplos Critérios e Sensível à Mobilidade (OSFEM), que utiliza uma heurística para aprimorar a eficiência de recursos e a qualidade do serviço (QoS) em cenários móveis.*

1. Introdução

O uso de recursos de rede está aumentando significativamente com a crescente conexão de dispositivos à internet, com o tráfego projetado para atingir aproximadamente 60,192 exabytes por ano em 2030 [Kadir et al. 2021]. Com isso, a Realidade Aumentada (RA) surge como uma forma de avançar na multimídia além dos vídeos sob demanda e comunicações por vídeo simples, proporcionando uma experiência imersiva aos usuários ao combinar elementos virtuais com a visão real. Nesse sentido, a Realidade Aumentada Multiusuário (RAMU) introduz o método de unir usuários na mesma aplicação de RA, utilizando recursos computacionais disponíveis mais próximos dos usuários, na borda da rede. Essa tecnologia promete transformar atividades como compras virtuais, onde consumidores podem visualizar produtos em RA antes da compra, e na medicina, permitindo que médicos de diferentes locais colaborem em cirurgias em tempo real.

A computação de borda, apesar de oferecer recursos poderosos de processamento, rede e armazenamento, geralmente está disponível para um número limitado de usuários, o que compromete sua escalabilidade. Além disso, a falta de sincronização e a inconsistência na entrega de dados podem causar enjoo de movimento nos usuários, especialmente em aplicações de RA [Akhtar et al. 2021]. Para aumentar a escalabilidade e suportar mais usuários com os mesmos recursos computacionais, é essencial habilitar as aplicações de borda para compartilharem esses recursos de forma eficiente, de modo a fazer com que os servidores colaborem mais entre si. Porém, isso requer que as aplicações sejam particionadas em serviços menores, onde parte desses serviços podem utilizar recursos computacionais para múltiplos usuários simultaneamente.

Nesse sentido, serviços que dependem apenas de computação de borda enfretam dificuldades para prover respostas rápidas, gerenciar adequadamente recursos computacionais e entregar uma baixa latência para usuários móveis (6 ms para serviços RAMU) [Siriwardhana et al. 2021]. Uma abordagem é decompor o serviço RAMU em Funções de Serviço (SF, do inglês *Service Function*) e fazer o encadeamento de SFs (SFC, do inglês *Service Function Chaining*) em sequência nos servidores de borda. Esse processo de modelagem inclui desde a captura de imagens até o reconhecimento de objetos, organizando as SFs em cadeias e paralelizando tarefas para melhorar a experiência RAMU. No entanto, a orquestração dessas cadeias em ambientes móveis apresenta desafios como ajustes constantes e roteamento eficiente para manter a Qualidade do Serviço (QoS) e a reutilização das SFs [Akhtar et al. 2021, Medeiros et al. 2022]. A conexão de usuários em diferentes servidores devido à mobilidade e a eficácia na reutilização das SFs são problemas ainda não resolvidos, afetando a latência e QoS.

Este trabalho apresenta a Orquestração de Encadeamento de Funções de Serviço de Múltiplos critérios e sensível a mobilidade (OSFEM). O esquema emprega uma heurística para aprimorar tanto a eficiência no uso dos recursos quanto a QoS. Desta forma, a OSFEM possibilita a reutilização das cadeias de SFs e a adaptação dinâmica às condições variáveis de rede. As simulações conduzidas demonstram que OSFEM melhora a utilização dos recursos enquanto aumenta a taxa de aceitação de serviços em até 4.31%, possibilitando, assim, atender a um contingente maior de usuários.

O restante deste artigo está organizado da seguinte forma: Seção 2 revisa trabalhos relacionados na área, Seção 3 detalha a OSFEM em cenários RAMU, Seção 4 avalia o desempenho do nosso esquema, e Seção 6 conclui o artigo com um resumo de nossas descobertas e direções futuras para pesquisa.

2. Trabalhos Relacionados

A pesquisa em orquestração de SFC para serviços multimídia imersivos em computação de borda foca predominantemente na redução da latência, visando mitigar o enjoo de movimento e aprimorar a experiência do usuário. Santos et al. [Santos, J. et al. 2021] desenvolveram um modelo de programação linear focado em cenários de usuário único para orquestrar cadeias de SFs. Akhtar et al. [Akhtar et al. 2021] examinaram a instanciação ótima de cadeias de SFs e direcionamento de tráfego, embora não abordassem a dinâmica de múltiplos usuários móveis. Wang et al. [L. Wang et al. 2021, T. Wang et al. 2020], exploraram a orquestração online de SFC com foco em mobilidade e aplicaram aprendizado profundo para redução de latência em streaming de vídeo a usuários móveis e RA para

usuários estáticos, respectivamente. Medeiros et al. [Medeiros et al. 2022] e Lin et al. [Lin et al. 2021] abordaram, respectivamente, a orquestração visando latência e eficiência energética para VR e um esquema parcialmente paralelo para minimizar latência e custos de processamento, ambos focados em usuários únicos e estáticos.

Santos et al. [Santos et al. 2022] propôs uma orquestração de serviços de VR visando múltiplos usuários, porém sem suporte para orquestração em tempo real de usuários móveis. Posteriormente, Santos et al. [Santos et al. 2023] propôs uma orquestração de serviços de RA para múltiplos usuários móveis com suporte à mobilidade e compartilhamento de SFs baseadas em localização. Entretanto, o MSF não considera a reutilização de recursos compartilháveis no processo de decisão e, portanto, reutiliza os recursos somente de forma passiva. Identifica-se, portanto, a necessidade de um esquema de orquestração de SFC que integre mobilidade, paralelização, suporte a múltiplos usuários, reutilização eficiente dos recursos limitados da borda e melhoria de QoS.

Tabela 1. Comparação das Características de Trabalhos Relacionados

Artigo	QoS	Mobilidade	Paralelismo	Multiusuário	Reúso Ativo
Santos et al. [Santos, J. et al. 2021]	✓				
Akhtar et al. [Akhtar et al. 2021]	✓				
Wang et al. [T. Wang et al. 2020]	✓				
Wang et al. [L. Wang et al. 2021]	✓	✓			
Medeiros et al. [Medeiros et al. 2022]	✓	✓			
PPC [Lin et al. 2021]	✓		✓		
MusFiCO [Santos et al. 2022]	✓		✓	✓	
MSF [Santos et al. 2023]	✓	✓	✓	✓	
OSFEM	✓	✓	✓	✓	✓

3. Esquema de Orquestração OSFEM

A arquitetura distribuída de borda, o modelo do sistema RAMU SFC e a operação da OSFEM são detalhados nesta seção. O esquema de orquestração OSFEM utiliza um algoritmo de orquestração SFC online para gerenciar grupos de usuários móveis, considerando múltiplos critérios para a instanciação de serviços em rotas de rede ótimas conforme a nova localização dos usuários. Esses critérios incluem a posição das SFs em execução, os recursos computacionais e de comunicação na borda, e o limiar de latência do serviço.

3.1. Arquitetura de Borda Distribuída para Serviços RAMU SFC

Para melhorar a qualidade do serviço (QoS) em ambientes de computação de borda, a RAMU com SFC deve ser implementada em uma arquitetura cliente-servidor enriquecida com servidores de borda, visando otimizar a alocação de funções de serviço (SFs) de maneira geograficamente distribuída, eliminando redundâncias e distribuindo recursos de forma eficiente. Esta abordagem permite que SFs sejam alocadas em servidores descentralizados, mantendo a QoS mesmo quando os usuários estão em movimento. Com tecnologias como 5G, 4G e WiFi, os usuários podem usar dispositivos móveis e óculos de RA para solicitar serviços à Nuvem Central, que, em conjunto com os servidores de borda, gerencia e distribui os recursos RAMU. Assim, as sessões RAMU com SFC são otimizadas com base na proximidade dos usuários e na seleção dinâmica do servidor de borda mais adequado, resultando em maior eficiência e menor latência.

O módulo *Instanciador de SF* decide o roteamento para estabelecer uma rota entre um cliente e um conjunto de SF's paralelizáveis, ordenadas e baseadas em localização, implantadas em um servidor de borda, considerando restrições computacionais, de rede, latência e a posição do cliente móvel. O *Orquestrador* aciona o *Controlador* para implementar a rota RAMU SFC e notifica o *Gerente de Recursos* para atualizar seus estados. O *Gerente de Recursos* monitora os recursos de rede e computação de cada servidor de borda e aciona o *Orquestrador* quando esses recursos são insuficientes para novas solicitações. O *Gerente de Mobilidade* detecta a movimentação dos clientes móveis e notifica o *Orquestrador* para atualizar os SFs baseados em localização. Assim, o *Orquestrador* ajusta dinamicamente as rotas e instancia SFs de acordo com a mobilidade dos clientes e os recursos disponíveis.

A *RAMU SFC* inclui SFs paralelizáveis, ordenadas e baseadas em localização, que seguem uma sequência específica: capturar um quadro de vídeo, extrair características da imagem, reconhecer objetos, sobrepor Objetos Virtuais (OV)s no quadro de vídeo, transformar os quadros em vídeo e exibir o conteúdo para o cliente móvel. Cada sessão move suas SFs de um servidor em nuvem para servidores de borda distribuídos, onde uma *SF IA* atua como servidor do serviço RAMU. Essa SF sincroniza as transformações de OVs para todos os clientes, permitindo interação multi-cliente e alinhando o sistema de coordenadas 3D virtual com o mundo real usando dados de GPS e edifícios.

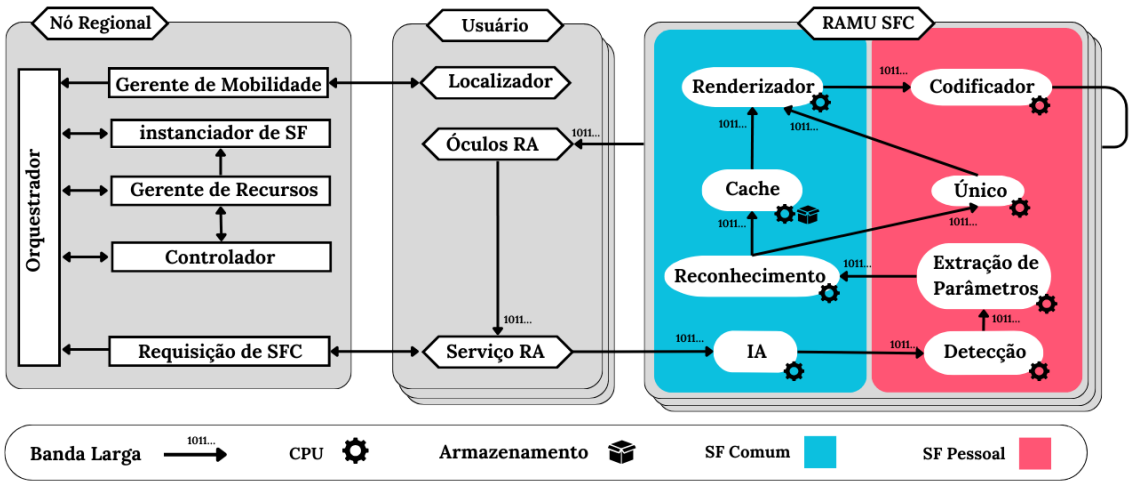


Figura 1. Arquitetura de Borda Distribuída para RAMU SFC

As SF's no sistema são dividas em comuns e pessoais. As SFs comuns, que lidam com dados compartilháveis e processamento de OV para usuários móveis, incluem a *SF de Interação (IA)*, *SF de Reconhecimento*, *SF de Cache* e *SF Renderizador*, conforme mostrado na área azul da Figura 1. SFs pessoais possuem apenas informações específicas do usuário, ou seja, *SF de Extração de Parâmetros*, *SF Único* e *SF de Codificação*, conforme mostrado na área vermelha na Figura 1. Por exemplo, uma *SF de Detecção* destaca OVs usados como referência para posicionar OVs na cena. A *SF de Extração de Parâmetros* pré-processa OVs para encaminhar como entrada para a *SF de Reconhecimento*. A SFC bifurca o fluxo para os fluxos *SF Único* e *SF Cache*. O fluxo *SF Único* refere-se à visualização exclusiva de OVs para um único usuário. O fluxo *SF Cache* refere-se a OVs comumente visualizados no espaço interativo por múltiplos usuários e que possuam as informações previamente armazenadas em cache como, por exemplo, o

horizonte ou o fundo do ambiente. Os fluxos da SFC convergem na *SF Renderizador*, que renderiza os fluxos da visualização atual e a *SF de Codificador* encapsula de forma compacta os OVs para transmissão e visualização nos óculos de RA.

3.2. Modelagem do Sistema

A modelagem do sistema proposta assume que nós de borda, que vão de dispositivos móveis a servidores em nuvem, incluindo *micro data centers* e Unidades de Banda Base (BBUs), são representados pelo grafo não direcionado $G = (V, E)$, com V indicando servidores e E, as conexões. Enlaces entre servidores v e v' possuem latências $d_{vv'}$, e capacidades de banda total ($b^c_{vv'}$), livre ($b^f_{vv'}$) e utilizada ($b^u_{vv'}$). Para cada servidor v, detalham-se capacidades totais, livres e utilizadas de CPU (p^c_v, p^f_v, p^u_v), respectivamente. Usuários móveis U interagem com SFC para RAMU em sessões s, com restrições de latência u^d e mapeamento em um grafo direcionado $G^a = (V^a, E^a)$. O RAMU é organizado em arcos de entrada/saída para Codificador, SF Única para fluxos pessoais, e SF de Cache para comuns, resultando em duas cadeias ordenadas de SFs lineares. Além disso, $D(v, sf_j)$ denota a latência acumulada da rota, enquanto a matriz $Rcpu(v, sf)$ mostra a reutilização de CPU no nó j para o serviço i.

3.3. Operação da OSFEM

A OSFEM utiliza uma heurística que divide o desafio de alocação de recursos em subproblemas menores para melhorar o desempenho do sistema. Essa estratégia permite soluções ágeis e eficazes na distribuição de SFs entre os servidores. Desse modo, cada subproblema avalia fatores críticos como computação (p^c_u), largura de banda (b^c_u) e latência (l^c_u), visando minimizar custos totais e maximizar a eficiência da rede. Esse sistema de custos facilita o reuso de SFs e ajustes dinâmicos na alocação de recursos. O sistema de custos é articulado através de quatro equações principais, delineadas a seguir, onde cada uma aborda um recurso específico a ser otimizado: CPU (1), largura de banda (2) e latência (3). A formulação de cada equação de custo é apresentada com suas respectivas variáveis e parâmetros:

$$C_{\text{cpu}} = \sum_{v=1}^{V} \sum_{j=1}^{J} \delta_{v,sf_j} \cdot (1 - R_{\text{cpu}}(v, sf_j)) \tag{1}$$

$$C_{\text{banda}} = \sum_{(v,v')}^{E} \sum_{j=1}^{J} \gamma_{vv',sf_j} \cdot sf^b_j / (b^c_{vv'} - b^f_{vv'}) \tag{2}$$

$$C_{\text{latencia}} = \sum_{(v,v')}^{E} \sum_{j=1}^{J} \gamma_{vv',sf_j} \cdot d_{vv'} \tag{3}$$

Nestas equações, os termos δ_{v,sf_j} e γ_{vv',sf_j} representam, respectivamente, variáveis binárias acerca da presença de uma SF específica sf_j em um servidor v ou em uma conexão entre servidores v e v'. $R_{\text{cpu}}(v, sf_j)$ representa a matriz de reuso e $d_{vv'}$ expressa a latência entre os servidores. A função objetivo demonstrada na Eq. (4) busca minimizar a soma ponderada dos custos individuais, onde os pesos K_1, K_2 e K_3 refletem a importância relativa dos recursos de CPU, largura de banda e latência, respectivamente, no contexto geral da otimização da rede.

$$\min C = K_1 \cdot C_{\text{cpu}} + K_2 \cdot C_{\text{banda}} + K_3 \cdot C_{\text{latencia}} \tag{4}$$

A orquestração das cadeias de SFs envolve dois passos principais: primeiro, o orquestrador coleta e mantém informações sobre as cadeias de SFs dos usuários móveis, iniciando a re-instanciação da SFC quando necessário. Em seguida, cada SF é mapeada e instanciada em um servidor de borda de forma ordenada, considerando o reúso e a disponibilidade de recursos, como largura de banda e CPU.

O orquestrador processa solicitações de usuários e eventos de mobilidade, atualizando a localização do usuário móvel l em sessões s ($u_l^s \in U$) e adicionando-os à fila apropriada. Ao receber novas solicitações, ele cria uma sessão de cadeia de SFs para RAMU s, agrupando usuários próximos u_l^s para facilitar a interação no serviço RAMU e colocando a localização do usuário na fila de nova instanciação de SF Q_i. Para eventos de mobilidade, ajusta a cadeia conforme a nova localização e move os dados de u_l^s para a fila de re-instanciação Q_m. O orquestrador então executa o algoritmo para roteamento das cadeias de SFs em Q_m. Se uma rota for estabelecida, o controlador a implementa; se falhar, as cadeias de SFs dos usuários em Q_m são canceladas. O mesmo processo é aplicado à fila de novos usuários Q_i, garantindo a interação contínua dos usuários durante a sessão.

O algoritmo distribui as SFs nos servidores de borda, com foco na otimização dos recursos e no atendimento dos requisitos de latência. Utilizando a matriz de latência cumulativa D, o algoritmo monitora o custo de cada alocação possível. A estratégia inclui a reutilização de recursos existentes para reduzir os custos, sempre verificando a viabilidade e a capacidade dos servidores para suportar novas funções sem comprometer o desempenho. O processo também prevê custos futuros para tomar decisões de forma proativa, buscando uma solução ótima que equilibre eficiência de custos e requisitos de desempenho. Caso os critérios não sejam atendidos, o algoritmo sinaliza uma falha na instanciação, sugerindo ajustes na estratégia de alocação ou nos parâmetros da rede.

A solução proposta possui uma complexidade computacional Big O de $O(k \cdot V^2)$, onde k é o número de SFs na solicitação e V é o número de servidores de borda. No entanto, a complexidade inferior é limitada por Omega $\Omega(k \cdot V)$. A heurística realiza k iterações, em cada uma das quais analisa a conectividade entre todos os pares de servidores $v \in V$, totalizando V verificações por iteração.

4. Avaliação

Esta seção descreve a metodologia de avaliação, incluindo a descrição do cenário, parâmetros de simulação e métricas usadas para avaliar o desempenho da OSFEM e de demais abordagens existentes em termos de taxa de aceitação de serviço, utilização de CPU, largura de banda e latência de acordo com diferentes probabilidades de mobilidade.

4.1. Ambiente de Simulação

Foi utilizado um simulador baseado em NetworkX e Python3 [1] para modelar os recursos de borda, latência e capacidade de processamento, coordenado por um orquestrador ciente da mobilidade, conforme discutido na Seção 3. A topologia da rede de borda foi baseada

[1] https://gitlab.com/gercomlacis/fog-vanet/multi-user-sfc

na infraestrutura da Cidade de Luxemburgo, abrangendo 35 nós, com latências geradas por uma distribuição de Poisson com média de 1 ms [Akhtar et al. 2021]. Um terço dos nós, selecionados por sua alta conectividade, foram designados como servidores de borda. Esses servidores estão conectados ao controlador da Nuvem Central através do nó 34, com links de 1 Gbps entre os nós.

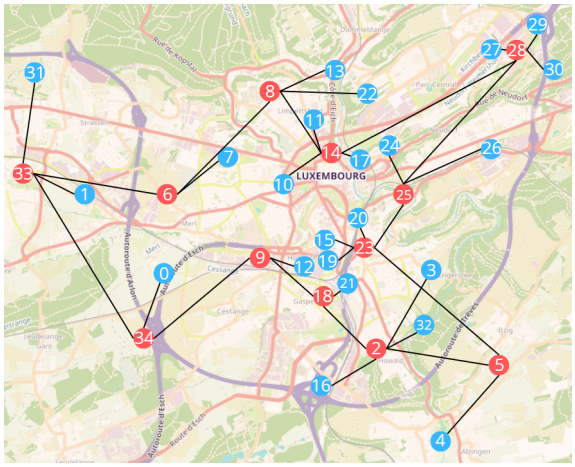


Figura 2. Topologia de borda de Luxemburgo

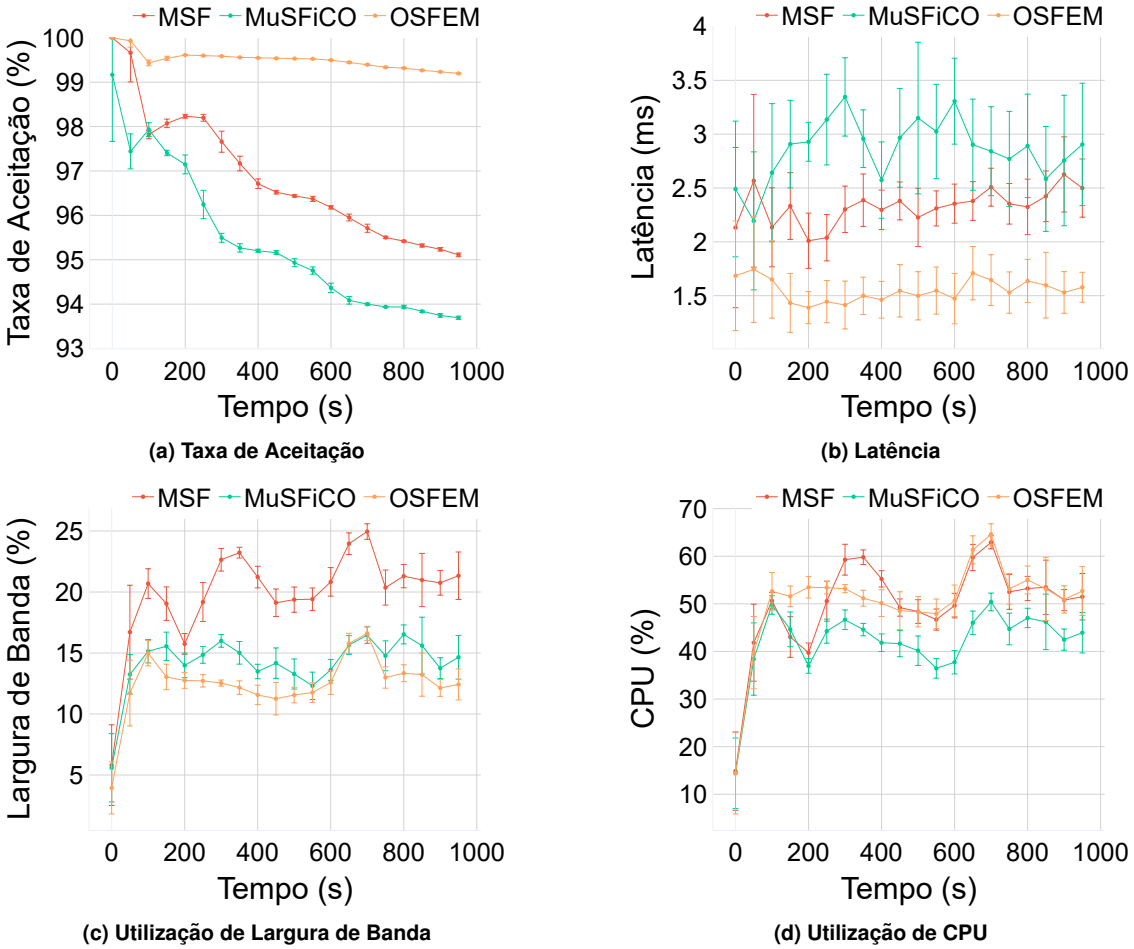
Foi utilizado o SUMO (Simulation of Urban Mobility) e dados do OpenStreetMap de Luxemburgo para simular movimento veicular urbano. O Gerenciador de Mobilidade detecta os *handovers* do usuários para um novo ponto de acesso ou um novo servidor de borda [Ngo, M. et al. 2020]. A RAMU avalia interações em grupos de 4/8/12/16 usuários móveis em serviços como, por exemplo, treinamento, turismo e jogos, com vídeos a 60 quadros por segundo [Liu et al. 2018]. A *SF de Detecção* produz imagens RGB de 400x400 pixels, média de 0.48 MB/quadro [Perronnin et al. 2010]. A *Extração de Parâmetros SF* processa 4 a 12 OV por quadro, 25 KB cada, com bitrate de [100, 300]. Os OVs, armazenados em cache têm até 50 MB com um mínimo médio de 120 MB por sessão e taxa de acerto de cache de 33% [Huang et al 2021]. Cada usuário consome até 33% da CPU do servidor de borda, com demanda de CPU por SF proporcional ao volume de dados, onde cada 10 bits de dados consomem 1 ciclo de CPU [Huang et al 2021]. Foram testados 50 sessões SFC para RAMU, chegadas por distribuição de Poisson (média de 15 s), duração até 120 s, e latência máxima de ida e volta de 12 ms. As sessões RAMU são aceitas se atenderem os requisitos de latência e recursos, dividindo a transmissão em SFs comuns e pessoais. Foram feitas 33 simulações para garantir um intervalo de confiança de 95% para cada esquema de orquestração RAMU.

O desempenho da OSFEM foi analisada junto a outros orquestradores, onde todas eles suportam o mesmo esquema de Gerenciador de Mobilidade [Ngo, M. et al. 2020]. MuSFiCO [Santos et al. 2022] mantém a instanciação de SFC existente, mas adiciona uma nova rota da SF final em direção ao usuário móvel. O MSF [Santos et al. 2023], por outro lado, possui uma instanciação de SFC ciente da mobilidade que adapta continuamente toda a rota da cadeia nos servidores de borda, porém não implementa o reuso ativo dos recursos da rede. Para avaliar abordagens baseadas em RAMU, consideramos métricas essenciais: *(i)* **Taxa de aceitação de sessões:** número de sessões aceitas pelo total de sessões; *(ii)* **Latência:** tempo de transmissão da fonte aos usuários móveis; *(iii)*

Utilização de CPU: uso de CPU pelas solicitações de SFC aceitas versus total de recursos dos servidores de borda; *(iv)* **Utilização de largura de banda:** uso de link pelas cadeias de SFs aceitas versus total dos recursos de conexão de borda;

5. Resultados

A Figura 3a ilustra o desempenho em Taxa de Aceitação ao longo do tempo. Pode-se observar que MSF e MuSFiCO apresentam Taxa de Aceitação similares ao longo do tempo, com MSF levemente superior, embora seja restrito pela sua limitada capacidade de reuso, o que resulta na subutilização de recursos e, portanto, com maior possibilidade de diminuir a aceitação de sessões. OSFEM, gerenciando melhor essas limitações, atinge 99% de aceitação média, superando MSF e MuSFiCO em 2.72% e 4.31%, respectivamente, e mantém desempenho estável ao longo do tempo em contraste com a degradação das outras soluções conforme o número de serviços aumenta.



(a) Taxa de Aceitação

(b) Latência

(c) Utilização de Largura de Banda

(d) Utilização de CPU

Figura 3. Comparação de desempenho dos algoritmos ao longo do tempo.

A Figura 3b ilustra a latência ao longo do tempo. A OSFEM alcançou uma latência média de 1.55 ms, 33.45% e 45.87% menor que MSF e MuSFiCO, respectivamente, graças à sua estratégia de reutilização de recursos para minimizar latência, aproximando os serviços dos usuários. MuSFiCO enfrenta aumento de latência com mobilidade devido a sua instanciação estática sem ajustes dinâmicos, enquanto MSF tem latência mais estável por manter serviços próximos aos servidores de borda.

A Figura 3c ilustra a utilização de banda ao longo do tempo por diferentes algoritmos. A análise dos dados revela que a OSFEM apresentou uma utilização significativamente menor de largura de banda. Especificamente, a OSFEM conseguiu reduzir o consumo de largura de banda em 12,19% em relação ao MuSFiCO e em 37,20% em comparação ao MSF. Essa eficiência é alcançada por meio da adaptação contínua da cadeia de serviços em resposta à mobilidade dos usuários, além da implementação de decisões proativas de reutilização de recursos, o que resulta em uma maior economia nos recursos dos servidores de borda localizados nas proximidades dos usuários.

Os algoritmos têm desempenhos semelhantes quanto a utilização de CPU, porém o MuSFiCO tem uma menor utilização porque aceita consideravelmente menos cadeias de serviço. Entretanto, A OSFEM e o MSF aceitam uma maior quantidade de recursos de processamento e, portanto, tem uma maior utilização de CPU. A OSFEM utiliza uma quantidade similar de recursos quando comparado ao MSF, porém a estratégia de reutiliaçào ativa de recursos da OSFEM faz com que mais serviços sejam aceitos, causando um menor impacto na utilização de recursos de processamento (ver Figura 3d).

Os cenários de mobilidade trazem um desafio adicional para manter os serviços RAMU com baixa latência. Os *handovers* também incrementam o uso de recursos computacionais e elevam o número de sessões bloqueadas. A chegada de mais solicitações de cadeias de SFs nas bordas da rede provoca uma saturação progressiva dos recursos das áreas, e um usuário móvel tem maior probabilidade de se mover para essas áreas, o que amplifica as chances de interrupção de serviço durante o ciclo de vida da SFC. Portanto, deve-se desenvolver um esquema de orquestração de cadeias de SFs para ambientes inteligentes, dinâmicos e multiusuários móveis, considerando mobilidade, paralelização de cadeias de SFs, suporte à QoS e, crucialmente, a integração do reúso de cadeias de SFs no cenário, a fim de aprimorar a eficiência dos recursos de computação distribuída na borda e proporcionar uma experiência aprimorada para os usuários RAMU.

6. Conclusão

Este trabalho apresentou a Orquestração de Encadeamento de Funções de Serviço de Múltiplos critérios e sensível a mobilidade em cenários de RAMU, denominado de OSFEM. Os resultados evidenciam a superioridade da solução apresentada em relação às existentes em termos de taxa de aceitação, eficiência de recursos e redução de latência em cenários de alta mobilidade. Futuramente, exploraremos o impacto do consumo de energia e resiliência da rede, visando ainda o gerenciamento eficiente dos recursos em ambientes dinâmicos, de modo a abrir caminhos para experiências de usuário cada vez mais imersivas e responsivas. Este trabalho desenvolvido por mim, aluno de graduação, e no âmbito da tese de doutorado de [Santos et al. 2022]. A modelagem e a operação da proposta foram desenvolvidas em conjunto, mas escrita, implementação e discussão dos resultados são de minha autoria.

Referências

Akhtar et al. (2021). Managing chains of application functions over multi-technology edge networks. *IEEE Trans. on Network and Service Management*.

Huang et al (2021). Proactive edge cloud optimization for mobile augmented reality applications. In *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE.

Kadir, E. A., Shubair, R., Rahim, S. K. A., Himdi, M., Kamarudin, M. R., and Rosa, S. L. (2021). B5g and 6g: Next generation wireless communications technologies, demand and challenges. In *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, pages 1–6. IEEE.

L. Wang et al. (2021). Change: Delay-aware service function chain orchestration at the edge. In *IEEE International Conference on Fog and Edge Computing (ICFEC)*. IEEE.

Lin, I.-C., Yeh, Y.-H., and Lin, K. C.-J. (2021). Toward optimal partial parallelization for service function chaining. *IEEE/ACM Transactions on Networking*, 29(5):2033–2044.

Liu et al. (2018). An edge network orchestrator for mobile augmented reality. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE.

Medeiros, A., Di Maio, A., Braun, T., and Neto, A. (2022). Service chaining graph: Latency-and energy-aware mobile vr deployment over mec infrastructures. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 6133–6138. IEEE.

Ngo, M. et al. (2020). Coordinated container migration and base station handover in mobile edge computing. In *IEEE Global Communications Conference*, pages 1–6.

Perronnin et al. (2010). Large-scale image retrieval with compressed fisher vectors. In *IEEE computer society conference on computer vision and pattern recognition*. IEEE.

Santos, H., Martins, B., Rosário, D., Cerqueira, E., and Braun, T. (2023). Mobility-aware service function chaining orchestration for multi-user augmented reality. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, pages 1–9. IEEE.

Santos, H., Rosario, D., Cerqueira, E., and Braun, T. (2022). Multi-criteria service function chaining orchestration for multi-user virtual reality services. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 6360–6365. IEEE.

Santos, J. et al. (2021). Efficient orchestration of service chains in fog computing for immersive media. In *17th International Conference on Network and Service Management (CNSM)*, pages 139–145. IEEE.

Siriwardhana et al. (2021). A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications, and technical aspects. *IEEE Communications Surveys & Tutorials*.

T. Wang et al. (2020). Adaptive service function chain scheduling in mobile edge computing via deep reinforcement learning. *IEEE Access*.