Orquestração Dinâmica de Encadeamento de Funções de Serviço 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. *Multi-User Augmented Reality (MUAR) employs edge computing for collaborative 3D interactions on mobile devices. This service can be segmented into Service Function Chainings (SFCs) across edge servers to enable parallel user execution. We propose the Multi-Criteria and Mobility-Aware Service Function Chaining Orchestration (MMASFCO), which minimizes latencies and optimizes network resources. Results indicate that MMASFCO enhances session acceptance, resource efficiency, and latency reduction compared to existing methods.*

Resumo. *A Realidade Aumentada Multiusuário (RAMU) usa computação em borda para colaboração em ambiente 3D em dispositivos móveis. Esse serviço pode ser decomposto em Service Function Chainings (SFCs) distribuídas em servidores de borda, permitindo execução paralela de usuários. Propomos a Orquestração de Encadeamento de Funções de Serviço Multicritério e Sensível à Mobilidade (OSFEM), que reduz latência e otimiza recursos de rede. Os resultados mostram que a OSFEM melhora a aceitação de sessões, a eficiência de recursos e a diminuição de latência em comparação com métodos existentes.*

1. Introdução

A Realidade Aumentada Multiusuário (RAMU) é uma tecnologia emergente que viabiliza colaborações em tempo real em ambientes 3D que mesclam realidade física e virtual. Esta tecnologia promete revolucionar várias práticas cotidianas. Por exemplo, em compras virtuais, consumidores podem visualizar produtos em Realidade Aumentada (RA) antes da compra; em eventos sociais, podem realizar tours virtuais por locais históricos ou museus; e na medicina, a RAMU permite que médicos de diferentes locais colaborem em cirurgias em tempo real. Contudo, a RAMU enfrenta desafios significativos, como limitações de processamento e bateria dos dispositivos de RA, enjoo de movimento devido à latência, e a necessidade de sincronismo entre usuários para colaboração efetiva. Portanto, uma gestão eficiente do processamento é crucial para manter o engajamento do usuário e garantir o desempenho do sistema [Huang et al 2021].

Para implementar efetivamente serviços RAMU, é essencial um sistema computacional que suporte múltiplos usuários com sincronização e baixa latência [Santos et al. 2023]. A estratégia na computação de borda envolve decompor o serviço RAMU em Funções de Serviço (SF, do inglês *Service Function*) e realizar o encadeamento dessas SFs (SFC, do inglês *Service Function Chaining*) para minimizar a latência

e otimizar recurso. Essa abordagem 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 à mobilidade (OSFEM). O esquema utiliza uma heurística para melhorar a eficiência no uso dos recursos e a QoS. Assim, a OSFEM permite a reutilização das cadeias de SFs e adaptação dinâmica às condições variáveis de rede. As simulações mostram que OSFEM aumenta a utilização dos recursos, a taxa de aceitação de serviços e reduz a latência em até 43.87%, atendendo mais 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 5 conclui o artigo com um resumo de nossas descobertas e direções futuras para pesquisa.

2. Trabalhos Relacionados

A pesquisa sobre orquestração de SFC (Service Function Chaining) para serviços multimídia imersivos em computação de borda tem se concentrado na redução da latência para melhorar a experiência do usuário e mitigar o enjoo de movimento. Santos et al. [Santos, J. et al. 2021] e Akhtar et al. [Akhtar et al. 2021] investigaram a orquestração para cenários de usuários únicos e a instanciação ótima de cadeias de SFs, respectivamente. Wang et al. [L. Wang et al. 2021, T. Wang et al. 2020] aplicaram técnicas de aprendizado profundo para reduzir a latência em streaming de vídeo. Medeiros et al. [Medeiros et al. 2022] e Lin et al. [Lin et al. 2021] focaram em latência e eficiência energética, porém limitados a usuários estáticos. Estudos mais recentes por Santos et al. [Santos et al. 2022, Santos et al. 2023] exploraram a orquestração para múltiplos usuários, incluindo suporte à mobilidade e reutilização de SFs baseadas em localização, embora sem considerar a reutilização ativa de recursos atrelada a uma abordagem com critérios múltiplos, ou seja, considerar os recursos da rede na alocação das SFCs. Até onde sabemos, apenas a OSFEM considerou todos os aspectos importantes em uma solução.

Tabela 1. Comparação das Características de Trabalhos Relacionados

Artigo	QoS	Mobilidade	Paralelismo	Multiusuário	Multicritério	Reúso Ativo
Santos et al. [Santos, J. et al. 2021]	✓					
Akhtar et al. [Akhtar et al. 2021]	✓					
Wang et al. [L. Wang et al. 2021]	✓	✓				
Wang et al. [T. Wang et al. 2020]	✓					
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

Esta seção apresenta a arquitetura de borda distribuída, o modelo do sistema RAMU SFC e a operação da OSFEM. O esquema de orquestração OSFEM possui um algoritmo de orquestração SFC *online* para grupos de usuários móveis considerando múltiplos critérios

para instanciação de serviços em rotas de rede ótimas de acordo com a nova posição dos usuários móveis. Os critérios são a posição das SFs em execução, os recursos computacionais e de comunicação da borda, e o limiar de latência do serviço.

3.1. Arquitetura de Borda Distribuída para Serviços RAMU SFC

A implementação da RAMU com SFC deve ocorrer em uma arquitetura cliente-servidor com servidores de borda para otimizar a alocação de SFs geolocalizadas e melhorar a QoS. Esta estratégia minimiza redundâncias e distribui recursos de forma eficiente, adequando-se à mobilidade dos usuários. As SFs são descentralizadas, permitindo conexões via 5G e Wi-Fi para dispositivos móveis e óculos RA. A nuvem central trabalha com servidores de borda para gerenciar os recursos da RAMU e aprimorar sessões SFC, escolhendo servidores próximos aos usuários e aumentando a eficiência.

O módulo *Instanciador de SF* determina a rota entre um cliente e as SFs paralelizáveis e localizadas em um servidor de borda, levando em conta restrições computacionais e de rede, latência e a posição do cliente móvel. O *Orquestrador*, em colaboração com o *Controlador*, define a rota da cadeia de SFs e comunica-se com o *Gerente de Recursos* para verificar a disponibilidade de recursos nos servidores. O *Gerente de Mobilidade* monitora a movimentação dos usuários, orientando ajustes na rota da cadeia de SFs. Assim, o *Orquestrador* adapta as rotas e SFs baseando-se na movimentação dos usuários e recursos disponíveis. *Servidores de Borda Distribuídos* contêm imagens de contêineres para ativação rápida dos serviços RAMU, interconectados para processamento eficiente de vídeo, abrangendo aquisição, reconhecimento de objetos e renderização.

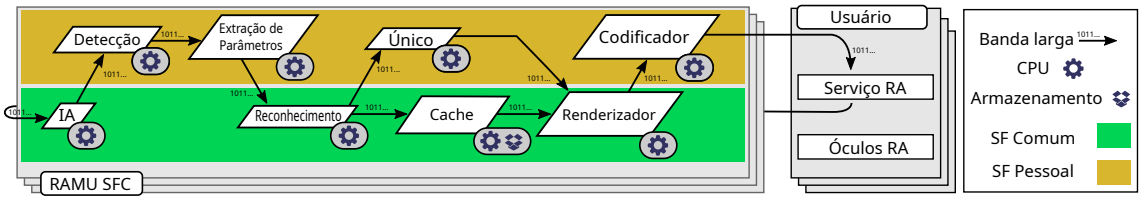


Figura 1. Arquitetura de Borda Distribuída para RAMU SFC

RAMU é composto por SFs comuns e pessoais. SFs comuns possuem dados compartilháveis e processamento de Objetos Virtuais (OV) para usuários móveis, ou seja, o servidor de serviço se divide em *SF de Interação(IA)*, *SF de Reconhecimento*, *SF de Cache* e *SF Renderizador*, conforme mostrado na área verde 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 amarela na Figura 1. Por exemplo, uma *SF de Detecção* destaca OVs usados como referência para posicionar OVs na cena. O *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 com mais recursos interativos em primeiro plano. 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 do 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.

O sistema define duas filas de instanciação, Q_m e Q_i, para gerir a instanciação de SFs, priorizando sessões contínuas de usuários em movimento e novas sessões de novos usuários, respectivamente. 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. Assim, 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 reúso de SFs e ajustes dinâmicos na alocação de recursos. O sistema de custos é articulado por 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).

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

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

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

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 reúso 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 segue dois passos principais: inicialmente, o orquestrador coleta e mantém as informações das cadeias de SFs dos usuários móveis, iniciando o processo de 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, CPU. O orquestrador processa solicitações de usuários e eventos de mobilidade, atualizando a localização do usuário móvel l em sessões s ($u^s_l \in U$) e os adiciona 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 coloca 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 conseguir estabelecer uma rota, o controlador a implementa; se falhar, cancela as cadeias de SFs dos usuários em Q_m. O mesmo se aplica à fila de novos usuários Q_i, assegurando a interação contínua dos usuários durante a sessão

O Algoritmo distribui SFs em servidores de borda, focando na otimização de recursos e no cumprimento dos requisitos de latência. Utilizando a matriz de latência cumulativa D, o algoritmo rastreia o custo de cada alocação potencial. A estratégia incorpora a reutilização de recursos existentes para minimizar custos, sempre verificando a viabilidade e a capacidade dos servidores para suportar novas funções sem degradar o desempenho. O processo também antecipa custos futuros para uma tomada de decisão proativa, buscando uma solução ótima que balanceie eficiência de custo e requisitos de desempenho. Se os critérios não forem atendidos, o algoritmo indica uma falha na instanciação, sugerindo ajustes na estratégia de alocação ou nos parâmetros da rede.

A solução proposta tem uma complexidade computacional Big O $O(k \cdot V \cdot V)$, onde k é o número de SFs na solicitação, e V é o número de servidores de borda, mas tem limite complexidade inferior Omega $\Omega(k \cdot V)$. A heurística percorre k vezes, em cada uma analisando 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, tempo de decisão, quantidade de SF'S compartilhadas e latência de acordo com diferentes probabilidades de mobilidade.

4.1. Ambiente de Simulação

Utilizou-se um simulador baseado em NetworkX e Python3 [1] para modelar recursos de borda, latência e capacidade de processamento sob a coordenação de um orquestrador com ciência de mobilidade, conforme Seção 3. A topologia de rede de borda modela a estrutura da Cidade de Luxemburgo com 35 nós, e latências originadas de uma distribuição de Poisson com média de 1 ms [Akhtar et al. 2021]. Um terço dos nós, os mais conectados, foi designado como servidores de borda. Esses servidores estão conectados ao controlador da Nuvem Central pelo nó 34, com conexões de 1 Gbps entre os nós.

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* dos usuários para um novo ponto de acesso ou um novo servidor de borda [Ngo, M. et al. 2020]. A RAMU avalia colaborações em grupos de 4 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 *SF de Extração de Parâmetros* 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

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

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 comparado com outros orquestradores, todos compatíveis com o mesmo Gerenciador de Mobilidade [Ngo, M. et al. 2020]. O MuSFiCO [Santos et al. 2022] mantém a SFC existente, adicionando apenas uma nova rota para a SF final ao usuário móvel. Já o MSF [Santos et al. 2023] adapta continuamente toda a cadeia de SFC nos servidores de borda, mas não implementa o reúso ativo dos recursos. Para avaliar as 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; *(v)* **SF's Compartilhadas:** referente ao número de SF's reutilizadas por múltiplas SFCs simultaneamente; *(vi)* **Tempo de Decisão:** rapidez com que o algoritmo processa e responde às solicitações de alocação de SFCs.

4.2. Resultados

A Figura 2f exibe uma análise comparativa sobre a eficiência no compartilhamento de SF's entre os algoritmos. A OSFEM adota uma estratégia dinâmica de otimização que favorece rotas com máximo compartilhamento de SF's, levando a uma utilização mais eficaz de recursos computacionais e de rede. Em contraste, o MSF e o MuSFiCO empregam uma reutilização mais passiva das SF's. Como resultado, a OSFEM supera o MSF e o MuSFiCO em 59.19% e 43.65%, respectivamente.

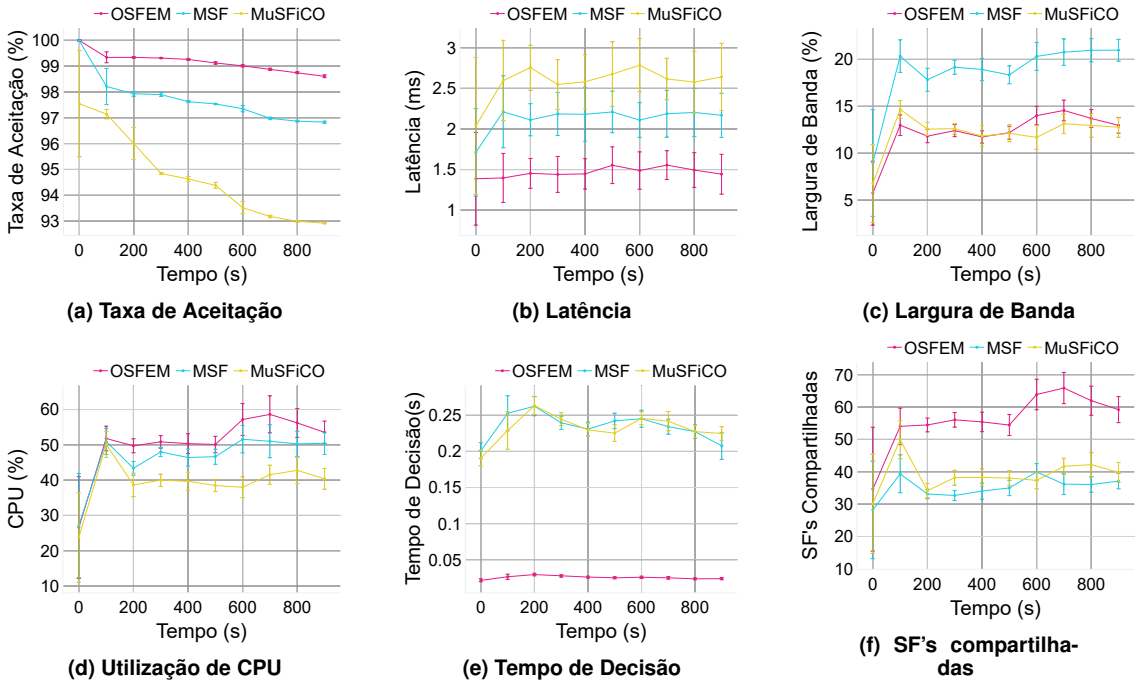
Essa estratégia de otimização reflete-se nos resultados apresentados na Figura 2a. Nesse sentido, tal capacidade da OSFEM de maximizar o compartilhamento de SFs permite uma gestão mais ágil e eficiente dos recursos disponíveis. Isso, por sua vez, amplia significativamente a taxa de aceitação de sessões. Desse modo, a OSFEM obtém uma taxa média de aceitação de 99.15%, superando o MSF e o MuSFiCO, respectivamente, em 1.47% e 4.69%, além de manter o desempenho estável ao longo do tempo.

A Figura 2b ilustra a latência ao longo do tempo. A OSFEM alcançou uma latência média de 1.46 ms, 31.45% e 43.87% menor que MSF e MuSFiCO, respectivamente, devido à modelagem do custo de latência e de reutilização, 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 análise dos dados da figura 2c 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.

A Figura 2e ilustra a evolução do tempo de decisão. Os resultados indicam que, em média, o tempo de decisão para OSFEM foi 89.15% menor em comparação ao MSF e ao MuSFiCO, devido ao menor limite de complexidade Ω e da alta taxa de compartilhamento de SF's na OSFEM reduzir o tempo de alocação de recursos.

Conforme apresentado na Figura 2d, os algoritmos exibem desempenhos similares em termos de utilização de CPU. Entretanto, o MuSFiCO registra menor consumo por aceitar menos cadeias de serviço. Em contraste, a OSFEM e o MSF processam mais recursos, elevando sua demanda de CPU. Apesar de consumirem quantidades similares de recursos, a OSFEM suporta mais serviços, otimizando assim a utilização da CPU.



(a) Taxa de Aceitação **(b) Latência** **(c) Largura de Banda**

(d) Utilização de CPU **(e) Tempo de Decisão** **(f) SF's compartilhadas**

Figura 2. Comparação de desempenho dos algoritmos ao longo do tempo.

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.

5. Conclusão

Este trabalho apresentou a Orquestração de Encadeamento de Funções de Serviço de Múltiplos critérios e sensível à 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, tempo de decisão, 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 [Andrew et al. 2024].

Agradecimentos

Este estudo foi financiado em parte pelo CNPq para o projeto intitulado "INCT Redes de Comunicação e Internet das Coisas Inteligentes (ICoNIoT)", e também pelo projeto intitulado "Computação de Borda em um Mundo de IoT Inteligente com Suporte a Latência Ultra Baixa (EdgeIo2T)". Além disso, este estudo foi financiado em parte pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) para o processo 2023/00673-7

Referências

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

Andrew, S., Bos, H., et al. (2024). *Sistemas operacionais modernos*. Bookman Editora.

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

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