



Projeto Brasil 6G Fase III

Atividade 2.4 - Concepção do Núcleo da Rede 6G - Parte 1



Histórico de Atualizações:

Versão	Data	Autor(es)	Notas
1	27/06/2025	Kleber Vieira Cardoso Sand Luz Correa Luciano de Souza Fraga Cristiano Bonato Both Diego Pivoto Daniely Gomes Silva Fernando Farias Hudson Romualdo João Paulo Esper Mendes Dante Campos Rafael Silva	Elaboração de conteúdo
2	31/07/2025	Kleber Vieira Cardoso Daniely Gomes Silva Luciano Leonel Mendes	Revisão de texto

Lista de Figuras

1	Pontos de acesso com infraestrutura de ICT e redes móveis.	3
2	Infraestrutura híbrida de computação de borda e nuvem.	11
3	Local Optimal Task Orchestration Solution (LOTOS).	12
4	Porcentagem de tarefas completadas.	15
5	Custo da infraestrutura e eficiência do modelo.	16
6	Visão geral da rede de acesso do Projeto Brasil 6G e potenciais aplicações relacionadas à IoT integrada ao agronegócio inteligente.	19
7	Cenário de fazenda inteligente com NS. Três fatias estão instanciadas: veículos autônomos (vermelho), dispositivos de sensoriamento (verde) e veículos aéreos não tripulados (azul).	20
8	Ilustração do processo de escalonamento de 5 TTIs num cenário de NS com três fatias.	21
9	Diagrama de testes do fluxo de comunicação inter-processos entre camada MAC e algoritmo externo.	24

Lista de Tabelas

1	Resumo do conjunto de dados.	8
2	Ganho do +Tour em comparação com o PersTour (RA).	9

Acrônimos

5G	<i>Quinta Geração de Rede Móvel Celular</i>
6G	<i>Sexta Geração de Rede Móvel Celular</i>
AI	<i>Artificial Intelligence</i>
BS	<i>Base Station</i>
GPS	<i>Global Positioning System</i>
ICT	<i>Information and Communication Technology</i>
IoT	<i>Internet of Things</i>
LBSN	<i>Location-Based Social Networks</i>
LoRa	<i>Long Range</i>
LPWAN	<i>Low Power Wide Area Network</i>
MAC	<i>Medium Access Control</i>
MAR	<i>Mobile Augmented Reality</i>
MCS	<i>Modulation and Coding Scheme</i>
MEC	<i>Multi-access Edge Computing</i>
MEC-PTIR	<i>MEC-aware Personalized Tour Itinerary Recommendation</i>
MILP	<i>Mixed Integer Linear Programming</i>
MIMO	<i>Multiple-Input Multiple-Output</i>
MR	<i>Mixed Reality</i>
MTC	<i>Metropolitan Tourism Centre</i>
MVS	<i>Mobile Video Streaming</i>
NS	<i>Network Slicing</i>
PHY	<i>Physical</i>
POI	<i>Point-of-Interest</i>
PRB	<i>Physical Resource Block</i>
PTIR	<i>Personalized Tour Itinerary Recommendation</i>
QoS	<i>Quality of Service</i>
RAN	<i>Radio Access Network</i>
RB	<i>Resource Block</i>
SDR	<i>Software-Defined Radio</i>
SLA	<i>Service Level Agreement</i>
SNR	<i>Signal-to-Noise Ratio</i>
SPPRC	<i>Shortest Path Problem with Resource Constraints</i>
TVWS	<i>TV White Space</i>
TTI	<i>Transfer Time Interval</i>

UE *User Equipment*

UHF *Ultra High Frequency*

VM *Virtual Machine*

Wi-Fi *Wireless Fidelity*

Índice

1	Sumário	1
2	+Tour: Recomendando Roteiros Personalizados para um Turismo Inteligente	2
2.1	Introdução	2
2.2	Solução Proposta	4
2.3	Resultados	7
2.4	Conclusão	9
3	Orquestração eficiente de tarefas, incluindo aplicações de realidade mista em infraestruturas de nuvem e borda	11
3.1	Introdução	11
3.2	Solução Proposta	12
3.3	Resultados	15
3.4	Conclusão	16
4	Escalonamento de recursos de rádio inter-fatias orientado a acordos a nível de serviço aplicado à infraestrutura de atendimento a áreas remotas	17
4.1	Rede de acesso para atendimento a áreas remotas: motivação e casos de uso . .	17
4.2	Escalonamento de recursos de rádio	19
4.3	Avanços obtidos: preparação da camada MAC para integração com escalonador externo	21
4.4	Atividades futuras para a validação e a integração do escalonamento de recursos	24
5	Considerações Finais	26

1 Sumário

A concepção do núcleo da rede 6G representa uma evolução significativa em relação às gerações anteriores de redes móveis, incorporando não apenas avanços em velocidade e latência, mas também uma arquitetura profundamente orientada a serviços, inteligência artificial embarcada, e integração nativa com tecnologias emergentes como computação de borda, realidade estendida e redes definidas por software. Ao contrário das gerações anteriores, o núcleo da rede 6G deverá ser altamente adaptável e personalizável para atender às demandas de diferentes setores, como turismo inteligente, aplicações imersivas e conectividade em áreas remotas. Essa nova abordagem abre espaço para a exploração de mecanismos avançados de orquestração de serviços, alocação de recursos sob demanda e experiências personalizadas baseadas em dados contextuais, criando um ecossistema dinâmico e inteligente que pode ser diretamente conectado a diferentes casos de uso emergentes.

No contexto do turismo inteligente, é apresentado na Seção 2 um resumo do artigo “+Tour: Recomendando Roteiros Personalizados para um Turismo Inteligente”, o qual foi publicado no periódico *Computer Networks*. Esse trabalho ilustra como a concepção do núcleo da rede 6G pode viabilizar experiências altamente personalizadas e contextuais por meio da coleta e processamento em tempo real de dados de mobilidade, preferências dos usuários e demanda das aplicações. A integração de capacidades analíticas diretamente no núcleo da rede permite que itinerários personalizados sejam recomendados de maneira proativa e contínua, considerando informações geolocalizadas e interações anteriores.

No domínio das aplicações imersivas, a Seção 3 apresenta um resumo do artigo do artigo “Orquestração eficiente de tarefas, incluindo aplicações de realidade mista em infraestruturas de nuvem e borda”, o qual foi publicado no 43º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2025). Esse trabalho mostra a importância do núcleo da rede 6G ser projetado para apoiar a orquestração eficiente de tarefas que envolvam altos requisitos de latência e processamento distribuído. A distribuição dinâmica de cargas entre nuvem e borda, viabilizada por um núcleo inteligente e orientado a serviços, é essencial para que experiências imersivas, como de realidade mista, possam ocorrer de forma fluida e responsiva. Além disso, a capacidade de realizar decisões de orquestração baseadas em políticas adaptativas e em tempo real reforça a importância de um núcleo 6G flexível, capaz de oferecer qualidade de experiência (QoE) diferenciada para aplicações sensíveis a atrasos.

Na Seção 4, é apresentado um trabalho em andamento que propõe o escalonamento de recursos de rádio inter-fatias orientado a acordos a nível de serviço, com foco em atender áreas remotas. Esse trabalho destaca outro pilar fundamental da concepção do núcleo da rede 6G: a habilidade de gerenciar eficientemente múltiplas fatias de rede com requisitos heterogêneos. Em regiões remotas, onde os recursos são escassos e a demanda pode variar substancialmente, o núcleo da rede precisa ser capaz de negociar, priorizar e adaptar a alocação de recursos conforme contratos de *Service Level Agreement* (SLA). Isso exige uma visão centralizada e inteligente da rede, com mecanismos autônomos de decisão que garantam tanto a continuidade do serviço quanto a sua qualidade, demonstrando como a evolução do núcleo 6G é essencial para garantir inclusão digital e equidade no acesso a serviços avançados de conectividade.

2 +Tour: Recomendando Roteiros Personalizados para um Turismo Inteligente

Autores

João Paulo Esper (UFMG), Luciano de S. Fraga (UFG), Aline C. Viana (Inria Saclay Île-De-France), Kleber Vieira Cardoso (UFG), Sand Luz Correa (UFG)

2.1 Introdução

O turismo exerce um papel essencial na economia de diversos países. Com o crescimento expressivo dessa indústria, a adoção de tecnologias da informação e comunicação (*Information and Communication Technology* (ICT)) tornou-se indispensável para tornar os serviços mais inteligentes, aprimorar a experiência dos turistas e fortalecer o engajamento dos usuários. Nas últimas décadas, observou-se uma expansão significativa de serviços digitais voltados à construção do turismo inteligente [1], caracterizado pela participação ativa dos viajantes em suas jornadas. Paralelamente, a popularização de dispositivos móveis e os avanços na comunicação sem fio possibilitaram o surgimento das redes sociais baseadas em localização (*Location-Based Social Networks* (LBSN)) [2], que fornecem ricos conjuntos de dados espaciais e temporais relacionados a mobilidade e aos interesses dos usuários. Esses dados têm sido amplamente explorados por sistemas de recomendação de itinerários, com o objetivo de oferecer serviços cada vez mais personalizados [3, 4, 5, 6].

Os sistemas de recomendação de itinerários turísticos personalizados (*Personalized Tour Itinerary Recommendation* (PTIR)) constituem um componente essencial do turismo inteligente. Considerando um conjunto de pontos de interesse (*Point-of-Interest* (POI)) em uma cidade, a disponibilidade de tempo do turista e suas preferências individuais, o principal objetivo dos sistemas PTIR é gerar um itinerário turístico, isto é, uma sequência ordenada de POIs, que maximize a experiência do usuário, respeitando suas restrições temporais. Soluções previamente propostas assumem que a qualidade da experiência está relacionada à popularidade dos POIs [7], ao interesse do usuário por determinadas categorias (como lojas, pontos históricos, entre outras) [8, 9, 10], ou a outras preferências inferidas a partir de redes sociais baseadas em localização (LBSN) [11, 12].

Com a implementação da Quinta Geração de Rede Móvel Celular (5G) em escala global, uma tendência emergente no contexto do turismo inteligente é a utilização de tecnologias de realidade aumentada móvel (*Mobile Augmented Reality* (MAR)). Essas tecnologias permitem que os usuários sobreponham elementos gerados por computador a imagens capturadas do mundo real, por meio de dispositivos móveis, como *smartphones* ou dispositivos vestíveis (*wearables*). No entanto, essa funcionalidade demanda algoritmos avançados de visão computacional e renderização gráfica, os quais requerem alto poder de processamento e que representam uma limitação para dispositivos móveis, cujos recursos computacionais são escassos.

A implementação da realidade aumentada móvel (MAR) exige um equilíbrio entre mobilidade, autonomia da bateria e desempenho computacional. Uma abordagem promissora para alcançar esse equilíbrio é o uso da computação de borda multiacesso (*Multi-access Edge Computing* (MEC)), que permite descarregar o processamento intensivo do MAR para a borda da rede sem fio [13, 14]. Esse descarregamento contribui para uma experiência do usuário mais fluida, além de reduzir a área de cobertura necessária para o tráfego de mídia sobre a rede sem fio [15]. Nesse contexto, à medida que as redes móveis se tornam mais difundidas e os aplicativos de MAR passam a utilizar formatos e resoluções de vídeo mais sofisticados, a qua-

lidade da experiência do usuário ao visitar um ponto de interesse dependerá não apenas dos próprios aplicativos utilizados (como redes sociais, *streaming* de vídeo móvel (*Mobile Video Streaming* (MVS)) e MAR), mas também da capacidade da infraestrutura de ICT em oferecer suporte eficiente a esses serviços.

Nesse contexto, negligenciar a consideração dos recursos de borda ao recomendar itinerários turísticos pode resultar em experiências frustrantes para usuários que buscam uma vivência enriquecida, integrando visitas físicas e interações virtuais avançadas. Assim, a integração entre PTIR e a alocação de recursos MEC constitui a principal motivação deste trabalho.

Considere, por exemplo, um centro metropolitano de turismo (*Metropolitan Tourism Centre* (MTC)) responsável por fornecer serviços de turismo inteligente. O MTC aluga recursos computacionais e de rede de uma operadora por meio de uma fatia de rede (*network slice*), a qual inclui cobertura móvel em cada POI, servidores MEC próximos aos POIs e recursos computacionais remotos em nuvem acessíveis via internet, conforme ilustrado na Figura 1. Além disso, o MTC disponibiliza um sistema PTIR para auxiliar os usuários na construção de itinerários personalizados, baseando-se tanto em suas preferências quanto nos aplicativos que pretendem utilizar durante a visita.

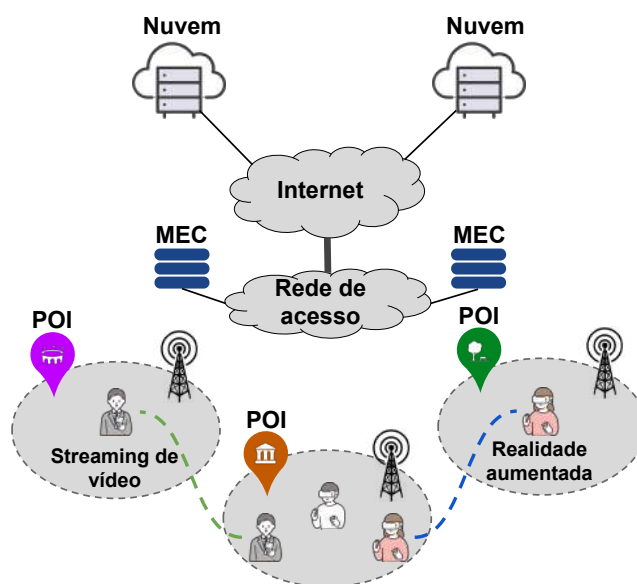


Figura 1: Pontos de acesso com infraestrutura de ICT e redes móveis.

Diferentemente de abordagens tradicionais, em que os itinerários são gerados exclusivamente com base na relevância dos POIs para o perfil do usuário, o sistema proposto considera também a otimização da alocação de recursos MEC ao executar os serviços selecionados. Embora isso possa ocasionar pequenas alterações nos POIs recomendados, o objetivo é alcançar uma maior eficiência no uso dos recursos de borda e, conseqüentemente, proporcionar uma experiência integrada de maior qualidade para múltiplos usuários. Este trabalho avança o estado da arte na área de recomendações de itinerários turísticos personalizados ao incorporar novas dimensões de preferências que emergem com a crescente disseminação de redes móveis e a adoção de serviços MAR.

2.2 Solução Proposta

Dado o grafo $G = (\mathcal{V}, \mathcal{E})$, que representa os POIs da cidade, o conjunto de usuários dispostos a visitar a cidade em um dia específico ($\mathcal{U}_d = \{u \in \mathcal{U} \mid d_u = d\}$), as preferências dos usuários ($int_u(c)$, $\forall c \in \mathcal{C}$, $\forall u \in \mathcal{U}_d$), as restrições de tempo dos usuários (b_u , $\forall u \in \mathcal{U}_d$) e os aplicativos que os usuários consumirão durante o passeio ($\mathcal{A}_u \in 2^{\mathcal{A}}$, $\forall u \in \mathcal{U}_d$), o objetivo é encontrar um único itinerário $I_{u*} = (v_0, \dots, v_{n+1})$, iniciando em v_0 e terminando em v_{n+1} , para cada usuário $u \in \mathcal{U}_d$, de modo que o conjunto de itinerários escolhidos, denotado por $\mathcal{I}_*$, maximize o ganho (*profit*) percebido por todos os usuários, priorizando a alocação de recursos do MEC e respeitando as restrições dos usuários e da infraestrutura.

Denominamos esse problema de *MEC-aware Personalized Tour Itinerary Recommendation* (MEC-PTIR). De fato, o MEC-PTIR incorpora a disponibilidade de recursos de ICT e a demanda dos aplicativos selecionados ao problema clássico de recomendação de itinerários (PTIR). Portanto, o MEC-PTIR é mais complexo do que o PTIR tradicional. Como o problema PTIR é NP-difícil [16], formulamos o problema MEC-PTIR como um problema de otimização em duas etapas, a fim de lidar com sua complexidade de forma eficiente.

MEC-PTIR: primeiro estágio

Definimos o primeiro estágio do MEC-PTIR como um problema de orientação (*orienteering problem*) multiobjetivo, formulado como um problema de programação inteiro. Formalmente, para cada itinerário $I_u = (v_0, \dots, v_{n+1})$, $u \in \mathcal{U}_d$, seja $x_{i,j} \in \{0, 1\}$ uma variável de decisão que indica se o POI $v_i \in \mathcal{V}$ e o POI $v_j \in \mathcal{V}$ são visitados em sequência no itinerário I_u ou não.

O objetivo da primeira etapa do MEC-PTIR é obter, para cada $u \in \mathcal{U}_d$, o conjunto $\mathcal{I}_{u*} = \{I_{u*}^1, \dots, I_{u*}^k\}$ de itinerários de modo que o ganho obtido seja maximizado enquanto o custo da viagem seja minimizado, ou seja:

Primeiro estágio (MEC-PTIR): $\forall u \in \mathcal{U}_d$

$$\text{maximizar } \sum_{i=0}^n \sum_{j=1}^{n+1} x_{i,j} Prof_u(v_i) \quad (1)$$

$$\text{minimizar } \sum_{i=0}^n \sum_{j=1}^{n+1} x_{i,j} Cost_u(v_i, v_j) \quad (2)$$

sujeito a:

$$0 < \sum_{i=0}^n \sum_{j=1}^{n+1} Cost_u(v_i, v_j) x_{i,j} \leq b_u, \quad (3)$$

$$\sum_{j=1}^{n+1} x_{0,j} = \sum_{i=0}^n x_{i,n+1} = 1, \quad (4)$$

$$\sum_{i=0}^n x_{i,r} = \sum_{j=1}^{n+1} x_{r,j} \leq 1, \forall r = 1, \dots, n, \quad (5)$$

$$2 \leq pos(v_i) \leq n+1, \forall i = 1, \dots, n+1, \quad (6)$$

$$pos(v_i) - pos(v_j) + 1 \leq n(1 - x_{i,j}), \forall i, j = 1, \dots, n+1. \quad (7)$$

A Equação (1) tem como objetivo maximizar o ganho total obtido pelo usuário u , enquanto a Equação (2) minimiza o custo da viagem, respeitando o orçamento do usuário (Restrição (3)).

As demais restrições garantem a construção adequada do itinerário. A Restrição (4) assegura que o percurso se inicie no POI v_0 e termine no POI v_{n+1} ; a Restrição (5) garante que o caminho seja conexo e que nenhum POI seja visitado mais de uma vez. Supondo que $pos(v_i)$ represente a posição do POI $v_i \in \mathcal{V}$ no itinerário I_u , as Restrições (6) e (7) asseguram que não haja rotas parciais (sub-tours).

Para um usuário $u \in \mathcal{U}_d$, o conjunto $\mathcal{I}_{u*}$ contém todos os itinerários Pareto eficientes que começam no vértice v_0 , terminam no vértice v_{n+1} e não violam a restrição de tempo do usuário. Uma solução $I_{u*}^k \in \mathcal{I}_{u*}$ é chamada de Pareto eficiente (ou não dominada) se não existir nenhuma solução viável $I_{u*}^{k'}$ que domine I_{u*}^k , ou seja, que seja pelo menos tão boa quanto I_{u*}^k em relação a todas as funções objetivo, e melhor do que I_{u*}^k em pelo menos uma função objetivo. O conjunto $\mathcal{I}_{u*}$ é denominado frente de Pareto.

MEC-PTIR: segundo estágio

O resultado do primeiro estágio do MEC-PTIR representa a frente de Pareto $\mathcal{I} = \bigcup_{u \in \mathcal{U}_d} \mathcal{I}_{u*}$, ou seja, o conjunto de todos os itinerários não dominados que não violam a restrição de tempo do usuário, para todos os usuários em $\mathcal{U}_d$. Cada itinerário $I_{u*}^k \in \mathcal{I}_{u*}$, $\forall \mathcal{I}_{u*} \subset \mathcal{I}$, possui um ganho definido por:

$$Prof(I_{u*}^k) = \sum_{v_i \in I_{u*}^k \setminus \{v_0, v_{n+1}\}} Prof_u(v_i). \quad (8)$$

Dado o conjunto $\mathcal{I}$, o objetivo do segundo estágio do MEC-PTIR é encontrar um conjunto $\mathcal{I}_*$ que maximize a soma dos ganhos percebidos pelos usuários, ao mesmo tempo que priorize a alocação de recursos na borda da rede. Além disso, $\mathcal{I}_*$ deve conter exatamente um itinerário para cada usuário $u \in \mathcal{U}_d$.

Para alcançar esse objetivo, formulamos o segundo estágio do MEC-PTIR como um problema de programação linear inteira mista (*Mixed Integer Linear Programming* (MILP)).

Seja a função indicadora $\phi(I_{u*}^k, v_i) \in \{0, 1\}$ aquela que representa a visita a um POI em um itinerário, sendo $\phi(I_{u*}^k, v_i) = 1$ se o itinerário $I_{u*}^k \in \mathcal{I}$ visita o POI $v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}$, e $\phi(I_{u*}^j, v_i) = 0$ caso contrário.

Assumimos que todo itinerário em $\mathcal{I}$ começa e termina dentro de um período de tempo T , e que os recursos de computação e rede para esses itinerários são alocados em intervalos de tempo discretos de tamanho Δt . Os intervalos de tempo são indexados por $\tau \in \mathbb{Z}$, tal que $1 \leq \tau \leq T$. Dado um itinerário $I_{u*}^k \in \mathcal{I}_{u*}$, um POI $v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}$ e um índice de tempo τ , definimos a função indicadora $\rho(I_{u*}^k, v_i, \tau) \in \{0, 1\}$, sendo $\rho(I_{u*}^k, v_i, \tau) = 1$ se o itinerário I_{u*}^k estiver visitando o POI v_i durante o tempo τ , e $\rho(I_{u*}^k, v_i, \tau) = 0$ caso contrário.

Seja o conjunto de variáveis de decisão $y(I_{u*}^k) \in \{0, 1\}$ que representa a escolha dos itinerários, de modo que $y(I_{u*}^k) = 1$ se o itinerário $I_{u*}^k \in \mathcal{I}_{u*}$ compõe a solução, e $y(I_{u*}^k) = 0$ caso contrário. Um servidor MEC deve fornecer recursos de computação para cada visita ao POI de cada usuário. Definimos a variável de decisão $z(I_{u*}^k, v_i, m) \in \{0, 1\}$ para representar a associação ao servidor MEC, de modo que $z(I_{u*}^k, v_i, m) = 1$ se o servidor MEC $m \in \mathcal{M}$ for responsável por fornecer recursos de computação durante uma visita ao POI $v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}$ no itinerário $I_{u*}^k \in \mathcal{I}_{u*}$, e $z(I_{u*}^k, v_i, m) = 0$ caso contrário.

Sejam as variáveis de decisão $p(I_{u*}^k, v_i) \in \mathbb{R}$ e $q(I_{u*}^k, v_i, m) \in \mathbb{R}$ que representam, respectivamente, a quantidade de recursos de rede e de computação (no servidor MEC $m \in \mathcal{M}$) alocados durante uma visita ao POI $v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}$ no itinerário $I_{u*}^k \in \mathcal{I}_{u*}$. Assumindo $Norm(value)$ como uma função genérica que normaliza um valor, definimos a função objetivo do segundo estágio do MEC-PTIR como:

Segundo estágio (MEC-PTIR):

$$\begin{aligned}
& \text{maximizar } \sum_{\mathcal{I}_{u*} \subset \mathcal{I}} \sum_{I_{u*}^k \in \mathcal{I}_{u*}} y(I_{u*}^k) \text{Norm}(\text{Prof}(I_{u*}^k)) + \\
& \sum_{\mathcal{I}_{u*} \subset \mathcal{I}} \sum_{I_{u*}^k \in \mathcal{I}_{u*}} \sum_{v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}} \frac{\text{Norm}(p(I_{u*}^k, v_i))}{2|I_{u*}^k \setminus \{v_0, v_{n+1}\}|} + \\
& \sum_{\mathcal{I}_{u*} \subset \mathcal{I}} \sum_{I_{u*}^k \in \mathcal{I}_{u*}} \sum_{v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}} \sum_{m \in \mathcal{M}} \frac{\text{Norm}(q(I_{u*}^k, v_i, m))}{2|I_{u*}^k \setminus \{v_0, v_{n+1}\}|}.
\end{aligned} \tag{9}$$

A Equação (9) tem como objetivo maximizar simultaneamente dois objetivos: (i) o ganho total agregado coletado por todos os usuários em $\mathcal{U}_d$; e (ii) o ganho associado a soma da quantidade de recursos de rede alocados e da quantidade de recursos de computação alocados na borda da rede. Juntos, esses objetivos garantem que o conjunto de itinerários gerados maximize a soma dos ganhos percebidos pelos usuários, ao mesmo tempo em que prioriza a alocação de recursos na borda da rede. A função $\text{Norm}(\text{value})$ assegura que ambos os objetivos permaneçam no mesmo intervalo de valores. O ganho total agregado e o ganho associado aos recursos de rede e computação possuem pesos iguais. Assim, a Equação (9) pode escolher itinerários que não possuam o maior lucro total individual, mas que forneçam o melhor equilíbrio com a alocação de recursos, resultando em uma experiência aprimorada para o conjunto de usuários como um todo. A função objetivo representada pela Equação (9) está sujeita às seguintes restrições.

Restrições de escolha de itinerário – Para cada usuário $u \in \mathcal{U}_d$, exatamente um itinerário $I_{u*}^k \in \mathcal{I}_{u*}$ deve ser selecionado, ou seja:

$$\sum_{I_{u*}^k \in \mathcal{I}_{u*}} y(I_{u*}^k) = 1, \quad \forall \mathcal{I}_{u*} \subset \mathcal{I}. \tag{10}$$

Se um itinerário $I_{u*}^k \in \mathcal{I}_{u*}$ for selecionado para compor a solução de um usuário $u \in \mathcal{U}_d$, também é necessário alocar um servidor MEC para cada POI visitado no itinerário, ou seja:

$$\begin{aligned}
& \sum_{m \in \mathcal{M}} z(I_{u*}^k, v_i, m) = y(I_{u*}^k) \phi(I_{u*}^k, v_i), \\
& \forall I_{u*}^k \in \mathcal{I}_{u*}, \quad \forall \mathcal{I}_{u*} \subset \mathcal{I}, \quad \forall v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}.
\end{aligned} \tag{11}$$

Restrições de demanda de serviço – Para cada usuário $u \in \mathcal{U}_d$ e itinerário selecionado $I_{u*}^k \in \mathcal{I}_{u*}$, também é necessário selecionar o mínimo de recursos exigidos para executar os aplicativos selecionados pelo usuário u , evitando alocar mais do que o necessário, ou seja:

$$\begin{aligned}
& p(I_{u*}^k, v_i) \geq \lambda_u^{\min} y(I_{u*}^k) \phi(I_{u*}^k, v_i) \wedge \\
& p(I_{u*}^k, v_i) \leq \lambda_u^{\max} y(I_{u*}^k) \phi(I_{u*}^k, v_i), \\
& \forall I_{u*}^k \in \mathcal{I}_{u*}, \quad \forall \mathcal{I}_{u*} \subset \mathcal{I}, \quad \forall v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}.
\end{aligned} \tag{12}$$

De forma similar, garantimos a alocação adequada dos recursos computacionais na borda da rede com a seguinte restrição:

$$\begin{aligned}
& q(I_{u*}^k, v_i, m) \geq \psi_u^{\min} z(I_{u*}^k, v_i, m) \phi(I_{u*}^k, v_i) \wedge \\
& q(I_{u*}^k, v_i, m) \leq \psi_u^{\max} z(I_{u*}^k, v_i, m) \phi(I_{u*}^k, v_i), \\
& \forall I_{u*}^k \in \mathcal{I}_{u*}, \quad \forall \mathcal{I}_{u*} \subset \mathcal{I}, \quad \forall v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}, \quad \forall m \in \mathcal{M}.
\end{aligned} \tag{13}$$

Restrições de capacidade de recursos – Finalmente, asseguramos que as alocações realizadas não excedam a quantidade de recursos de rede e computação disponíveis, em qualquer intervalo de tempo:

$$\sum_{I_{u*}^k \in \mathcal{I}_{u*}} \rho(I_{u*}^k, v_i, \tau) p(I_{u*}^k, v_i) \leq \lambda_{v_i}, \quad (14)$$

$$\forall v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}, \quad 1 \leq \tau \leq T.$$

$$\sum_{I_{u*}^k \in \mathcal{I}_{u*}} \sum_{v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}} \rho(I_{u*}^k, v_i, \tau) q(I_{u*}^k, v_i, m) \leq \psi_m, \quad (15)$$

$$\forall m \in \mathcal{M}, \quad 1 \leq \tau \leq T.$$

Resolução do +Tour

Para resolver o MEC-PTIR, propomos uma solução para geração de recomendações turísticas assistida por redes móveis (+Tour). Como descrito na Seção 2.2, a primeira etapa do MEC-PTIR é um problema de orientação multiobjetivo com dois objetivos conflitantes: maximizar o lucro total coletado e minimizar o custo de viagem. Este problema pertence à classe de problemas NP-difíceis [17]. Para resolver essa parte do problema de forma ótima e eficiente, +Tour utiliza uma variante do problema do caminho mais curto com restrições de recursos (*Shortest Path Problem with Resource Constraints* (SPPRC)) [18]. A segunda etapa do MEC-PTIR é um problema MILP, que pode ser resolvido diretamente com uma ferramenta otimização linear como Gurobi, IBM ILOG CPLEX ou SCIP [19].

2.3 Resultados

Extraímos dados do Flickr¹, um aplicativo real de compartilhamento de fotos com base em localização, para derivar históricos de viagens de usuários e estatísticas relacionadas a visitas. O uso de dados do mundo real tem três objetivos principais: (i) demonstrar como dados reais podem alimentar nosso modelo de informações sobre POIs e preferências dos usuários; (ii) caracterizar padrões reais de visitação; e (iii) alcançar uma avaliação mais realista do +Tour. Com esse objetivo em mente, coletamos dados de treze cidades em quatro continentes diferentes, a saber: Ásia (Nova Deli e Osaka), Europa (Atenas, Barcelona, Budapeste, Edimburgo, Glasgow, Londres, Madri e Viena), América do Norte (Toronto) e Oceania (Perth e Melbourne). Essas cidades são relevantes do ponto de vista turístico e garantem variedade e diversidade à nossa avaliação. A seguir, descrevemos nossa metodologia de extração de dados. Em seguida, apresentamos uma avaliação experimental da solução de recomendação proposta.

Caracterização dos Dados

Utilizamos a Google Places API¹ para obter a lista de POIs das cidades. Cada cidade identifica cada POI por ID, nome, latitude e longitude, bem como pela categoria à qual pertence. O custo $c_{i,j}$ de uma aresta que conecta dois POIs v_i e v_j na mesma lista é calculado utilizando a Google Matrix Distance API² no modo de caminhada.

¹<https://developers.google.com/places/web-service/intro>

²<https://developers.google.com/maps/documentation/distance-matrix/start>

Utilizamos a Flickr API³ para extrair fotos com geolocalização e derivar os históricos de viagem dos usuários para cada cidade, exceto Melbourne, cuja base de dados é fornecida por [20]. Cada foto do Flickr é associada ao ID do usuário que a publicou, ao carimbo de data/hora e às coordenadas (latitude e longitude).

Para cada cidade, combinamos seu conjunto de fotos com sua lista de POIs para gerar o conjunto $\mathcal{S}$ de históricos de viagem na cidade. Primeiro, associamos cada foto ao POI correspondente utilizando as coordenadas geográficas da foto e do POI. Uma foto é mapeada para um POI se suas coordenadas geográficas diferirem por menos de 100 metros, de acordo com a fórmula de Haversine. Caso essa condição seja satisfeita por mais de um POI, a foto é associada ao POI mais próximo, de modo que cada foto seja mapeada para um único POI. Em seguida, construímos o histórico de viagem S_u para cada usuário, ordenando as fotos por ID do usuário e registro de data/hora e agrupando visitas consecutivas a POIs na mesma sequência de viagem, desde que a duração da sequência não ultrapasse 8 horas. Para cada POI v_i em uma sequência de viagem, consideramos o horário da primeira e da última foto tirada pelo usuário u em v_i como o tempo de chegada ($t_{v_i}^a$) e de partida ($t_{v_i}^d$) nesse POI, respectivamente. Por fim, descartamos as sequências de viagem com ciclos (visitas repetidas a POIs), com POIs com menos de cinco visitas, ou com apenas uma foto (pois resultam em uma duração de sequência de zero segundos). A Tabela 1 detalha o nosso conjunto de dados resultante, que também está disponível publicamente em um repositório no GitHub⁴.

Tabela 1: Resumo do conjunto de dados.

Cidade	País	Continente	Fotos	Usuários	Sequências	Sequências válidas	POIs	Categoria do POI
Atenas	Grécia	Europa	5026	291	516	509	24	5
Barcelona	Espanha	Europa	13654	782	1654	1634	29	8
Budapeste	Hungria	Europa	8090	573	2361	1058	36	6
Edimburgo	Escócia	Europa	20620	879	5028	2405	25	6
Glasgow	Escócia	Europa	8172	367	2227	969	25	7
Londres	Inglaterra	Europa	48519	2198	5018	4949	30	10
Madri	Espanha	Europa	20881	577	1396	1382	30	8
Melbourne	Austrália	Oceania	15255	626	5106	2096	84	9
Nova Déli	Índia	Asia	2961	186	489	278	19	6
Osaka	Japão	Asia	5306	279	1115	540	23	4
Perth	Austrália	Oceania	2503	99	716	307	19	7
Toronto	Canadá	América do Norte	30267	846	6057	2721	29	6
Viena	Itália	Europa	22432	704	3193	1613	28	8
13 cidades	10 países	4 continentes	203686	8407	34876	20461	401	20 (únicos)

Avaliação Experimental

Como atualmente o PersTour [10] é o estado da arte para soluções exatas do PTIR voltadas para o caso de um único viajante em um único dia, comparamos o +Tour com o PersTour. No entanto, o PersTour não considera os recursos MEC nem as demandas de aplicações de serviço ao gerar o itinerário.

Executamos todos os experimentos em uma máquina virtual que executa Debian 11 GNU/Linux, configurada com 4 vCPUs, 32 GB de RAM e 136 GB de disco virtual. A máquina virtual é hospedada em um servidor com 2 processadores Intel Xeon Silver 4114 @2.20GHz. A primeira etapa do +Tour é implementada em C++. A segunda etapa utiliza Python 3.8.2, docplex 2.23.222 e IBM ILOG CPLEX 20.1.0 (como ferramenta de otimização). O PersTour também é implementado com Python 3.8.2, docplex 2.23.222 e IBM CPLEX 20.1.0.

³<https://www.flickr.com/services/developer/api/>

⁴<https://github.com/LABORA-INF-UFG/plusTour>

Apresentamos duas novas métricas, a saber, **Eficiência de Alocação** (*Allocation Efficiency (AE)*) e **Experiência do Usuário** (*User Experience (UE)*), para avaliar o desempenho dos algoritmos na alocação de recursos na borda da rede e na experiência geral do usuário proporcionada pelo itinerário recomendado. Assumindo que S_u é a sequência de viagem real usada como teste e I_{u*}^k é o itinerário gerado, essas métricas são definidas da seguinte forma.

- **Eficiência de Alocação:** quantidade de recursos de rede e computação alocados ao itinerário gerado, relativa à quantidade máxima demandada pelo conjunto de aplicações escolhidas, ou seja:

$$AE(I_{u*}^k) = \sum_{v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}} \frac{p(I_{u*}^k, v_i)}{2|I_{u*}^k \setminus \{v_0, v_{n+1}\}| \lambda_u^{max}} + \sum_{v_i \in \mathcal{V} \setminus \{v_0, v_{n+1}\}} \sum_{m \in \mathcal{M}} \frac{q(I_{u*}^k, v_i, m)}{2|I_{u*}^k \setminus \{v_0, v_{n+1}\}| \psi_u^{max}}. \quad (16)$$

- **Experiência do Usuário:** determina a relação entre a eficiência de alocação associada ao itinerário gerado (experiência virtual) e o ganho percebido pelo usuário para esse itinerário (experiência física), ou seja:

$$UE(I_{u*}^k) = AE(I_{u*}^k) Norm(Prof(I_{u*}^k)). \quad (17)$$

A Tabela 2 apresenta o ganho obtido pelo +Tour (+T) em relação ao PersTour (RA) na comparação entre ambos os algoritmos em termos de **AE** e **UE** para todos os cenários (Londres (LO), Melbourne (ME), Osaka (OS), Perth (PE) e Toronto (TO)). Observa-se que o +Tour supera o PersTour em todos os cenários ao se considerar todos os comprimentos de sequência, especialmente sob alta sobrecarga da rede, apresentando ganhos de até 11% em **AE** e 40% em **UE**. Considerando apenas os itinerários com mais de 3 POIs, os ganhos em **AE** e **UE** são ainda maiores, atingindo, respectivamente, 74% e 65%.

Tabela 2: Ganho do +Tour em comparação com o PersTour (RA).

Todos os comprimentos das sequências															
	Sobrecarga elevada					Sobrecarga média					Sobrecarga baixa				
	LO	ME	OS	PE	TO	LO	ME	OS	PE	TO	LO	ME	OS	PE	TO
+T AE	0.653	0.794	0.584	0.514	0.672	0.746	0.911	0.687	0.602	0.797	0.872	0.942	0.798	0.718	0.874
RA AE	0.649	0.731	0.552	0.461	0.657	0.744	0.852	0.665	0.579	0.793	0.869	0.885	0.791	0.704	0.870
Ganho AE	0.67%	8.64%	5.67%	11.47%	2.29%	0.32%	6.95%	3.24%	3.88%	0.49%	0.27%	6.41%	0.80%	2.03%	0.53%
+T UE	0.106	0.119	0.106	0.135	0.134	0.114	0.131	0.121	0.153	0.152	0.125	0.135	0.138	0.176	0.165
RA UE	0.103	0.085	0.084	0.099	0.123	0.112	0.097	0.108	0.137	0.151	0.124	0.101	0.132	0.171	0.163
Ganho UE	2.25%	40.6%	25.7%	35.6%	8.85%	1.19%	35.5%	12.2%	11.2%	0.84%	1.08%	33.7%	4.42%	2.81%	0.94%
Sequências maiores que 3															
+T AE	0.904	0.863	0.843	0.840	0.845	0.937	0.952	0.917	0.901	0.932	0.941	0.990	0.952	0.949	0.982
RA AE	0.868	0.642	0.603	0.482	0.756	0.917	0.749	0.751	0.756	0.908	0.921	0.796	0.906	0.914	0.954
Ganho AE	4.15%	34.3%	39.7%	74.2%	11.8%	2.18%	27.1%	22.2%	19.2%	2.57%	2.17%	24.4%	5.05%	3.77%	2.91%
+T UE	0.538	0.355	0.419	0.640	0.565	0.555	0.379	0.472	0.726	0.625	0.560	0.389	0.480	0.789	0.654
RA UE	0.519	0.231	0.254	0.389	0.500	0.544	0.261	0.357	0.619	0.617	0.549	0.273	0.441	0.775	0.645
Ganho UE	3.70%	53.4%	65.1%	64.2%	12.9%	2.04%	45.2%	32.2%	17.3%	1.22%	2.02%	42.8%	8.90%	1.80%	1.41%

2.4 Conclusão

Neste trabalho, propomos o +Tour, uma solução em duas etapas para o problema de recomendação de itinerários com restrição de recursos. Nossa abordagem maximiza simultaneamente o lucro e a alocação de recursos, enquanto minimiza os custos de viagem. Apresentamos

uma extensa caracterização dos dados baseada em históricos reais de visitação a POIs de 13 cidades distribuídas em 4 continentes. Esse conjunto de dados inclui 20.461 sequências de passeio válidas geradas por 8.407 usuários que visitaram 401 POIs divididos em 20 categorias únicas, o que nos permite avaliar o +Tour em cenários realistas de grande porte. Também comparamos nossa solução com o estado da arte PersTour, demonstrando que nossa abordagem alcança até 74,2% de melhoria na métrica de Eficiência de Alocação e 65,1% na métrica de Experiência do Usuário. O +Tour induz um comportamento cooperativo nos turistas, o que pode beneficiar tanto os usuários quanto o provedor do serviço turístico. Na prática, sabemos que os turistas podem não seguir as recomendações, o que pode degradar o desempenho estimado. No entanto, essa questão afeta negativamente qualquer sistema de recomendação.

3 Orquestração eficiente de tarefas, incluindo aplicações de realidade mista em infraestruturas de nuvem e borda

Autores

Luciano de S. Fraga (UFG), Leizer de L. Pinto (UFG), Kleber V. Cardoso (UFG)

3.1 Introdução

O futuro das redes de acesso sem fio traz avanços significativos que possibilitam a implantação de serviços com exigências altamente rigorosas em termos de consumo de recursos e qualidade de serviço (*Quality of Service* (QoS)). Entre esses serviços, destacam-se as aplicações de realidade mista (*Mixed Reality* (MR)), que representam um caso de uso relevante, exemplificando as exigências rigorosas desse tipo de serviço devido à sua alta demanda por baixa latência e recursos computacionais [21, 22, 23]. Além dos desafios inerentes à própria classe de aplicações, a crescente densidade de dispositivos conectados em redes sem fio, impulsionada pela proliferação de dispositivos ubíquos da internet das coisas (*Internet of Things* (IoT)), torna ainda mais complexa a tarefa de manter os parâmetros de qualidade exigidos. Cada dispositivo móvel apresenta características únicas e é limitado em termos de poder de processamento, memória, energia e armazenamento, tornando impraticável que muitas cargas de trabalho sejam processadas localmente nos dispositivos dos usuários.

Para lidar com essas limitações, a adoção da computação em borda surge como um elemento-chave para viabilizar aplicações imersivas impulsionadas por tecnologias sem fio, como WiFi 6E/7/8. A computação em borda aproxima os recursos computacionais dos usuários finais, reduzindo a latência e aliviando a carga sobre dispositivos com recursos limitados, desempenhando assim um papel fundamental no atendimento às exigências rigorosas da realidade mista e de outros serviços imersivos. A adoção da computação em borda permite que cargas de trabalho sejam executadas em servidores localizados na borda da rede, oferecendo uma alternativa à computação em nuvem tradicional, na qual os serviços dependem de servidores centralizados em *data centers* distantes. Ao aproximar os recursos computacionais do usuário, a computação em borda atende de forma eficaz aos rigorosos requisitos de serviços que demandam baixa latência e alta capacidade computacional.

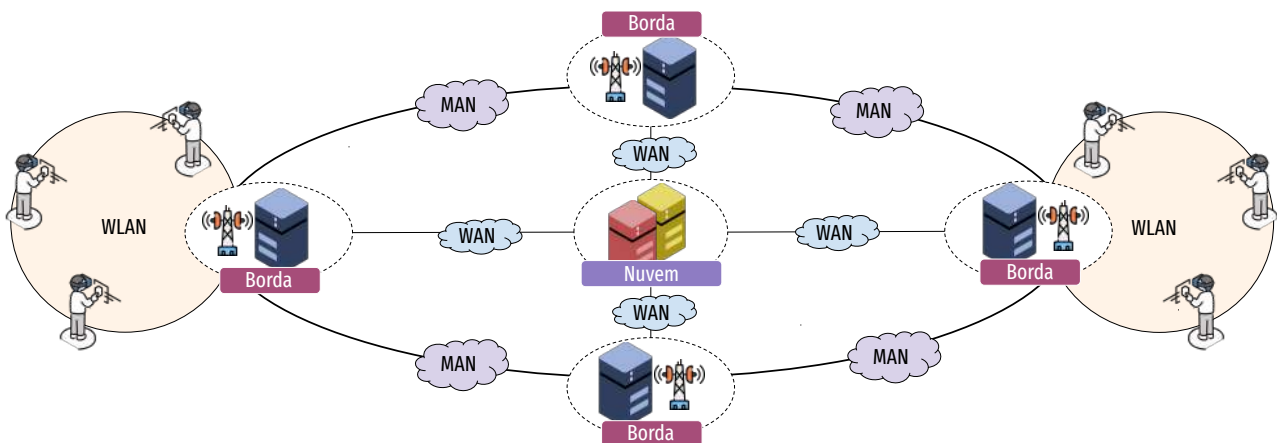


Figura 2: Infraestrutura híbrida de computação de borda e nuvem.

Nesse contexto, propomos o *LOTOS*, uma solução projetada para lidar com o problema

de alocação de recursos por meio da orquestração de um conjunto de tarefas (uma abstração da carga de trabalho dos usuários), descarregando a demanda de processamento em máquinas virtuais (*Virtual Machine* (VM)) na infraestrutura de borda ou na nuvem (Figura 2). O *LOTOS* adota uma abordagem localmente ótima, resolvendo uma parte do problema de forma iterativa para alcançar soluções eficientes. Vale ressaltar que, embora a solução proposta utilize redes WiFi como interface de acesso à infraestrutura de borda, ela pode ser facilmente adaptada para redes de telecomunicações, como a 5G. Neste trabalho, pretendemos: 1) verificar os benefícios de resolver a alocação de recursos para múltiplas tarefas; 2) apresentar uma solução que considere não apenas os requisitos de QoS dos usuários, mas também o custo de consumo dos recursos da infraestrutura; e 3) analisar as melhorias alcançadas pelo *LOTOS* em comparação com soluções existentes na literatura.

3.2 Solução Proposta

Consideramos que um usuário $u \in \mathcal{U}$ está conectado a um ponto de acesso $a \in \mathcal{A}$, em uma infraestrutura semelhante à apresentada na Figura 2. Cada usuário gera um conjunto de tarefas, que são um conceito abstrato representando uma unidade de carga de trabalho composta por instruções de processamento, demanda de memória e CPU, e a quantidade de dados gerados por uma aplicação. Definimos uma VM como $v \in \mathcal{E} \cup \mathcal{C} \cup \mathbb{O}$, onde $\mathbb{O}$ é uma máquina localizada próxima ao ponto de acesso ao qual o usuário u está conectado, $\mathcal{E}$ é uma VM de borda vizinha, e $\mathcal{C}$ é uma VM na nuvem. O objetivo geral do modelo é alocar as tarefas dos usuários às VMs de forma a maximizar o número de tarefas executadas com sucesso, em conformidade com as restrições de rede, atraso, processamento e memória, ao mesmo tempo em que se minimiza o custo de uso da infraestrutura.

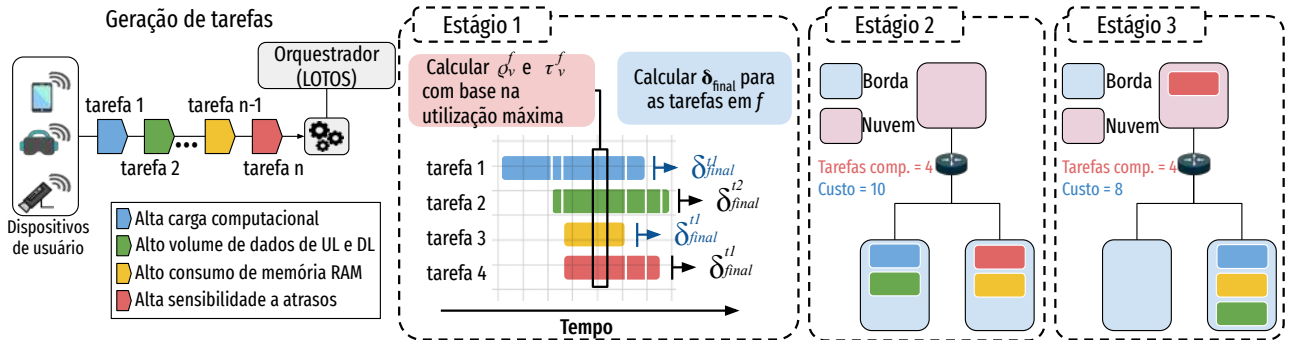


Figura 3: Local Optimal Task Orchestration Solution (LOTOS).

A solução proposta é composta por três estágios. No primeiro estágio, calculamos a quantidade de recursos consumidos por um conjunto de tarefas alocadas a uma VM. Simultaneamente, considerando os recursos disponíveis, determinamos o tempo de processamento das tarefas que compartilham os recursos dentro da mesma VM. No segundo estágio, tendo como entrada a saída do estágio 1, o objetivo é maximizar o número de tarefas executadas com sucesso, decidindo onde cada tarefa será alocada. Por fim, no terceiro estágio, a meta é minimizar o custo monetário ao processar as tarefas, garantindo a qualidade da solução gerada no estágio 2. A Figura 3 apresenta uma visão geral da solução em três estágios proposta neste trabalho.

Estágio 1: No primeiro estágio do *LOTOS*, precisamos calcular o consumo de recursos e o tempo necessário para processar cada tarefa em uma determinada VM. Para isso, implementamos os Algoritmos 1 e 2, que realizam os cálculos para um conjunto específico de tarefas f

alocadas em uma VM v . Definimos f como um elemento do conjunto das partes $\mathcal{F} = \mathbb{P}(\mathcal{T})$, representando uma combinação específica de tarefas, onde os recursos disponíveis na VM v são compartilhados entre elas. As saídas desses algoritmos para todos $f \in \mathcal{F}$ e $v \in \mathcal{V}$ serão utilizadas na execução dos estágios 2 e 3.

Algoritmo 1: Calculo de demanda da configuração

Entrada: $v, f, \Phi^v, \Omega^v, \omega_u^t, \phi_u^t, \delta_{inicial}^{t,u} \quad \forall t \in f$
Saída: $\delta_{final}^{t,u}, \rho_v^f, \tau_v^f$

- 1 $\mathcal{T}_{ordenado}^v \leftarrow \text{Ordenar}(\delta_{inicial}^{t,u} \quad \forall t \in f);$
- 2 $\delta_{atual} \leftarrow \text{Min}(\delta_{inicial}^{t,u} \quad \forall t \in f);$
- 3 $k \leftarrow 1 \quad i \leftarrow \mathcal{T}_{ordenado}^v[k] \quad j \leftarrow \mathcal{T}_{ordenado}^v[k+1] \quad \mathcal{T}_{concorrente}^v \leftarrow i;$
- 4 **enquanto** $(k \neq |\mathcal{T}_{ordenado}^v|)$ **faça**
 - 5 $\delta_{dif} \leftarrow \delta_{inicial}^j - \delta_{inicial}^i;$
 - 6 $\text{Atualização da carga de trabalho}();$
 - 7 $\mathcal{T}_{concorrente}^v \leftarrow \mathcal{T}_{concorrente}^v \cup j \quad k \leftarrow k+1 \quad \delta_{atual} \leftarrow \delta_{atual} + \delta_{dif};$
 - 8 $\rho_v^f \leftarrow \text{Max}(\rho_v^f, \text{CPU}(\mathcal{T}_{concorrente}^v)) \quad , \quad \tau_v^f \leftarrow \text{Max}(\tau_v^f, \text{RAM}(\mathcal{T}_{concorrente}^v));$
 - 9 $i \leftarrow \mathcal{T}_{ordenado}^v[k] \quad j \leftarrow \mathcal{T}_{ordenado}^v[k+1];$
- 10 **fim**
- 11 **enquanto** $(\mathcal{T}_{concorrente}^v \neq \emptyset)$ **faça**
 - 12 $i \leftarrow \text{Min}(\omega_u^t \times \phi_u^t \quad \forall t \in \mathcal{T}_{concorrente}^v);$
 - 13 $cpt \leftarrow \frac{\Phi^v}{|\mathcal{T}_{concorrente}^v| + |\mathcal{T}_{legado}^v|} \quad cap_u^i \leftarrow \frac{cpt}{\text{Max}(\phi_u^i, cpt)} \times \Omega^v \quad \delta_{dif} \leftarrow \frac{\omega_u^i}{cap_u^i};$
 - 14 $\text{Atualização da carga de trabalho}();$
 - 15 $\delta_{atual} \leftarrow \delta_{atual} + \delta_{dif};$
- 16 **fim**

Algoritmo 2: Atualização da carga de trabalho

Entrada: $\mathcal{T}_{concorrente}^v, \mathcal{T}_{legado}^v, \delta_{dif}, \delta_{atual}, \Phi^v, \Omega^v, \phi_u^t, \omega_u^t \quad \forall t \in \mathcal{T}_{concorrente}^v$
Saída: Atualização de $\mathcal{T}_{concorrente}^v, \delta_{final}^{t,u}$ e ω_u^t

- 1 **para** $t \in \mathcal{T}_{concorrente}^v$ **faça**
 - 2 $cpt \leftarrow \frac{\Phi^v}{|\mathcal{T}_{concorrente}^v| + |\mathcal{T}_{legado}^v|} \quad cap_u^t \leftarrow \frac{cpt}{\text{Max}(\phi_u^t, cpt)} \times \Omega^v;$
 - 3 **se** $\delta_{dif} \times cap_u^t \geq \omega_u^t$ **então**
 - 4 $\mathcal{T}_{concorrente}^v \leftarrow \mathcal{T}_{concorrente}^v - t;$
 - 5 $\delta_{final}^{t,u} \leftarrow \delta_{atual} + \frac{\omega_u^t}{cap_u^t};$
 - 6 **senão**
 - 7 $\omega_u^t \leftarrow \omega_u^t - (\delta_{dif} \times cap_u^t)$
 - 8 **fim**
- 9 **fim**

Estágio 2: As Equações 18-26 representam o problema a ser resolvido no estágio 2 do LOTOS. O objetivo, definido na Equação 18, é maximizar o número de tarefas executadas com sucesso, alocando-as em qualquer VM disponível na infraestrutura. A variável binária de decisão x_v^f indica a alocação do conjunto de tarefas f à VM v , onde $x_v^f = 1$ se o conjunto f for alocado à VM v , e $x_v^f = 0$ caso contrário. $A(x_v^f)$ retorna o número total de tarefas em f .

$$\text{maximizar} \quad \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} x_v^f \times A(x_v^f) \quad (18)$$

$$\text{sujeito a:} \quad \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} (x_v^f \times \rho_v^f) \leq P_v, \quad (19)$$

$$\sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} (x_v^f \times \tau_v^f) \leq T_v, \quad (20)$$

$$\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{E}} (x_v^f \times W_{dev}^{wlan}(x_v^f, a)) \leq \zeta_a^{wlan}, \quad (21)$$

$$\sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{C}} (x_v^f \times W_{dev}^{wan}(x_v^f, a)) \leq \zeta_a^{wan}, \quad (22)$$

$$\sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{E}} (x_v^f \times W_{tasks}(x_v^f)) \leq \text{Min}(\zeta_{ul}^{man}, \zeta_{dl}^{man}), \quad (23)$$

$$(\Gamma_{com}^{t,u,v} + \Gamma_{proc}^{t,u,v} + \Gamma_{wait}^{t,u,v}) \times x_v^f \times M(x_v^f, t, u) \leq \Delta_u^t, \quad \forall f \in \mathcal{F}, t \in \mathcal{T}, u \in \mathcal{U}, v \in \mathcal{V}, \quad (24)$$

$$\sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} x_v^f \times M(x_v^f, t, u) \leq 1, \forall t \in \mathcal{T}, u \in \mathcal{U}, \quad (25)$$

$$\sum_{f \in \mathcal{F}} x_v^f \leq 1, \forall v \in \mathcal{V}. \quad (26)$$

Estágio 3: A solução apresentada no estágio 2 determina o destino de cada tarefa, com o objetivo de minimizar falhas e otimizar a utilização dos recursos de armazenamento, processamento e comunicação, garantindo ao mesmo tempo o cumprimento dos requisitos de qualidade específicos da aplicação. No estágio 3, a meta é minimizar o custo de implementação da infraestrutura, sem deixar de considerar os benefícios alcançados pela solução desenvolvida no estágio 2.

A Equação 27 define a função objetivo para minimizar o custo total do uso das máquinas virtuais no processamento das tarefas. O custo c^v representa o valor monetário por unidade de tempo que uma VM incorre para processar um conjunto de tarefas. A função $T(x_v^f)$ indica o tempo total que a VM é utilizada para processar toda a carga de trabalho das tarefas em f . A Equação 28 garante que o número de tarefas executadas sem falhas permaneça consistente com o resultado obtido no estágio 2. Neste contexto, x_v^{f*} representa as variáveis de decisão correspondentes à solução ótima obtida no estágio 2.

$$\text{minimizar} \quad \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} x_v^f \times T(x_v^f) \times c^v \quad (27)$$

$$\text{sujeito a:} \quad \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} x_v^f \times A(x_v^f) = \sum_{f \in \mathcal{F}} \sum_{v \in \mathcal{V}} x_v^{f*} \times A(x_v^{f*}), \quad (28)$$

$$\text{Equações} \quad 19-26. \quad (29)$$

3.3 Resultados

Neste trabalho, realizamos uma avaliação extensiva do desempenho do *LOTOS*, comparando-o com três abordagens alternativas: 1) um método que prioriza as VMs menos carregadas (UTIL.); 2) uma solução que seleciona a VM com menor custo (MIN COST); e 3) uma abordagem baseada em lógica difusa (Fuzzy), proposta em [24]. Para essa avaliação, utilizamos o EdgeCloudSim, um simulador de redes construído sobre o CloudSim [25], que permite a modelagem de enlaces de comunicação sem fio, a caracterização de diversas aplicações, a criação de modelos de mobilidade de usuários e a personalização de estratégias de orquestração de tarefas de forma flexível. Para representar aplicações de realidade mista, empregamos o gerador de carga apresentado na caracterização de uma aplicação chamada MR-Leo [26]. A implementação foi realizada utilizando Java 21.0.5 e Python 3.10. Além do EdgeCloudSim, utilizamos a ferramenta de otimização CPLEX (versão 22.1.0) e o docplex 2.23.221 para desenvolver o modelo de otimização. O código completo deste trabalho está disponível em um repositório no GitHub ⁵.

A Figura 4 ilustra o percentual de tarefas concluídas com sucesso, em dois cenários distintos. No cenário original, os limites de uso de RAM e de atraso para cada tarefa não são considerados, e todas as VMs de borda são configuradas com o mesmo número de núcleos de CPU. Essa configuração foi escolhida para garantir um ambiente de avaliação consistente, correspondendo ao apresentado em [24]. Como mostrado na Figura 4(a), nossa solução supera todas as demais abordagens nessas condições. O segundo cenário representa um ambiente de VMs com capacidades variáveis, refletindo as especificações de máquinas virtuais em um servidor da AWS. Todos os resultados subsequentes deste trabalho são avaliados com base nesse cenário.

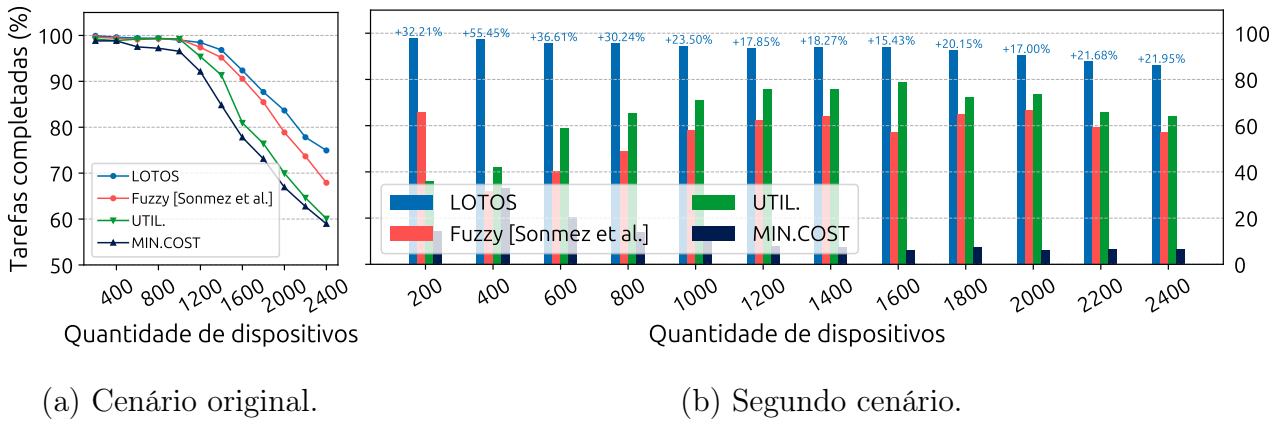


Figura 4: Porcentagem de tarefas completadas.

Conforme ilustrado na Figura 4(b), nossa solução apresenta uma melhoria significativa na execução bem-sucedida de tarefas, com aumentos que variam de 15,43% a 55,45% em todas as instâncias analisadas. Por outro lado, a abordagem Fuzzy não considera fatores como latência ou utilização de RAM em seu processo de decisão, o que compromete sua capacidade de alocar recursos de forma eficiente e evitar gargalos de desempenho. Em contraste, nossa solução leva em conta esses fatores críticos, possibilitando uma alocação de recursos mais eficaz. Além disso, a capacidade da nossa abordagem de avaliar múltiplas tarefas simultaneamente durante

⁵<https://github.com/LABORA-INF-UFG/paper-LLK-2025>

o processo de tomada de decisão permite uma estratégia de alocação mais precisa e proativa, diferentemente da natureza imediatista e reativa do método Fuzzy.

A Figura 5(a) apresenta o custo por tarefa concluída, demonstrando que nossa solução alcança maior eficiência na alocação de recursos, ao mesmo tempo em que completa um número mais elevado de tarefas sem erros. Notavelmente, nossa abordagem reduz os custos em até aproximadamente 8 vezes em comparação com o modelo UTIL., mesmo apresentando uma eficiência superior na conclusão das tarefas. A Figura 5(b) ilustra a eficiência na utilização de recursos, medida como o percentual de tarefas que consomem recursos e são executadas com sucesso. Essa eficiência é calculada dividindo-se o número de tarefas aceitas pelo número total de tarefas enviadas a uma VM. Com o *LOTOS*, pelo menos 90% das tarefas enviadas a uma VM são executadas com sucesso, enquanto todas as demais soluções apresentam níveis de eficiência inferiores a 90% em todas as instâncias.

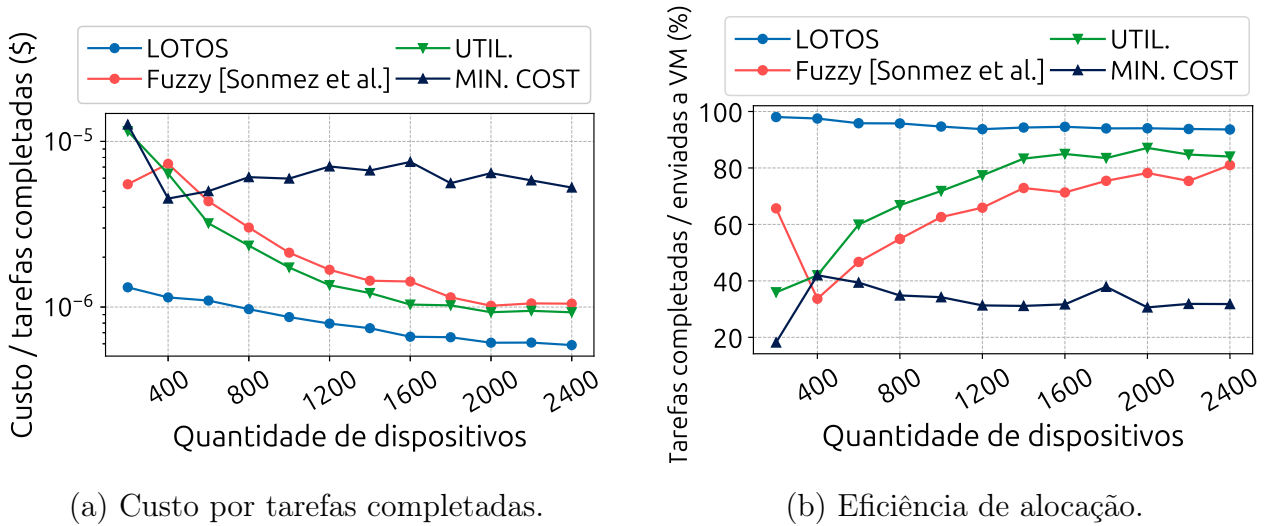


Figura 5: Custo da infraestrutura e eficiência do modelo.

3.4 Conclusão

Neste trabalho, apresentamos o *LOTOS*, uma solução em três estágios projetada para lidar com o problema de orquestração de tarefas, maximizando a execução bem-sucedida das tarefas ao mesmo tempo em que minimiza os custos de infraestrutura. A solução adota uma abordagem localmente ótima, que considera a alocação simultânea de recursos para um conjunto de tarefas. Nossa avaliação demonstrou que o *LOTOS* melhora significativamente a taxa de tarefas concluídas com sucesso e alcança uma eficiência de custo de até 8 vezes em comparação com as abordagens alternativas. Além disso, nossa solução garante que pelo menos 90% das tarefas alocadas a uma VM sejam executadas sem erros, em contraste com a eficiência máxima abaixo de 90% alcançada pelos trabalhos comparativos. Ademais, o *LOTOS* otimiza a utilização dos recursos da infraestrutura e os custos ao identificar potenciais erros antes da fase de simulação, evitando, assim, a alocação de tarefas em VMs quando a execução bem-sucedida não é viável.

4 Escalonamento de recursos de rádio inter-fatias orientado a acordos a nível de serviço aplicado à infraestrutura de atendimento a áreas remotas

Autores

Daniel Campos da Silva (UFG), Diego Gabriel Soares Pivoto (Inatel), Luciano Leonel Mendes (Inatel), Kleber Vieira Cardoso (UFG)

4.1 Rede de acesso para atendimento a áreas remotas: motivação e casos de uso

A conectividade digital em áreas remotas do território brasileiro ainda representa um dos principais desafios para a inclusão socioeconômica, educacional e produtiva de populações isoladas. O Brasil, com dimensões continentais e uma topografia diversificada, possui extensas regiões rurais, ribeirinhas e de difícil acesso logístico, nas quais a ausência de infraestrutura de telecomunicações compromete o acesso a serviços essenciais, como saúde, educação e apoio à produção agropecuária. A baixa densidade populacional, os altos custos de implementação e manutenção de redes convencionais e a falta de retorno financeiro para operadoras comerciais tornam essas regiões historicamente negligenciadas sob o ponto de vista da conectividade.

Nesse contexto, o Projeto Brasil Sexta Geração de Rede Móvel Celular (6G) surge como uma proposta inovadora voltada à democratização do acesso à conectividade, com foco no atendimento eficiente a essas áreas remotas. Para isso, é proposta uma infraestrutura de rede de acesso programável baseada em *Software-Defined Radio* (SDR), o qual oferece flexibilidade, reconfigurabilidade e controle dinâmico dos parâmetros de comunicação. A arquitetura da rede permite que múltiplos usuários sejam atendidos simultaneamente, mesmo sob condições adversas de cobertura, ao mesmo tempo em que proporciona integração com tecnologias locais de comunicação, promovendo interoperabilidade e adaptabilidade.

Um dos diferenciais técnicos da rede de acesso do Projeto Brasil 6G é a utilização de enlaces de radiofrequência de longas distâncias, baseados em canais do espectro não licenciado conhecidos como *TV White Spaces* (TVWSs). Os TVWSs referem-se a porções de espectro ocioso na faixa de *Ultra High Frequency* (UHF) originalmente destinadas à radiodifusão televisiva, mas que, por estarem desocupadas em determinadas regiões, podem ser aproveitadas para a transmissão de dados com excelente alcance e penetração do sinal. Essa característica é particularmente vantajosa para áreas rurais, pois permite cobrir grandes extensões territoriais com um número reduzido de estações rádio-base, reduzindo custos de infraestrutura e aumentando a viabilidade de implantação.

Testes de desempenho realizados com a rede de acesso demonstraram que é possível estabelecer comunicações confiáveis entre uma *Base Station* (BS) e um terminal de usuário (*User Equipment* (UE)) a distâncias de dezenas de quilômetros em cenários com linha de visada desobstruída. Essa capacidade amplia significativamente o raio de cobertura da infraestrutura, sendo um elemento-chave para o atendimento de comunidades dispersas e propriedades agrícolas localizadas em regiões afastadas dos centros urbanos.

Além da robustez física do enlace de rádio, os terminais de usuário da rede 6G são projetados com capacidade de interconectar redes locais heterogêneas baseadas em tecnologias como *Wireless Fidelity* (Wi-Fi), *Long Range* (LoRa) para *Low Power Wide Area Network* (LPWAN), *ZigBee*, entre outras. Dessa forma, é possível formar ecossistemas de conectividade local nas

comunidades ou fazendas remotas, em que os dados gerados localmente são agregados e encaminhados à Internet ou a redes privadas por meio do enlace de longa distância provido pela infraestrutura do Projeto Brasil 6G.

Considerando esse cenário de conectividade ampliada, diversos casos de uso passam a ser viabilizados com a infraestrutura proposta. Entre os exemplos mais relevantes, destacam-se:

- educação remota: conexão de escolas em áreas rurais para acesso a conteúdos educacionais, plataformas de ensino à distância e capacitação de professores;
- telemedicina: atendimento médico remoto, coleta e envio de exames, monitoramento de pacientes crônicos;
- redes de IoT para proteção ambiental: sensores de umidade do solo, detecção de incêndios florestais, monitoramento de qualidade do ar;
- operações logísticas e de segurança pública: comunicação entre *drones* e agentes em campo, rastreamento de veículos, suporte a vigilância territorial.

Entretanto, para fins deste capítulo, será dado foco especial ao cenário de IoT aplicada a *Smart Farming*, área com grande potencial de impacto socioeconômico. A conectividade rural viabilizada pela rede de acesso permite a digitalização de processos agrícolas, otimizando a produção e promovendo a sustentabilidade. Dentre as aplicações possíveis, destacam-se:

- monitoramento ambiental e climático, com sensores distribuídos para coleta de dados de temperatura, umidade do ar e do solo, pressão atmosférica, índice pluviométrico, entre outros, auxiliando na previsão e planejamento agrícola;
- localização e rastreamento de maquinário agrícola, permitindo gestão em tempo real da frota, redução de custos operacionais e prevenção de furtos;
- controle e acompanhamento de rebanhos, através de coleiras inteligentes com sensores de movimento, temperatura corporal e localização *Global Positioning System* (GPS), que permitem detectar alterações de comportamento ou doenças;
- aplicações com *drones*, utilizados para reconhecimento aéreo da plantação, mapeamento do terreno, identificação precoce de pragas ou áreas com deficiência de nutrientes, além de permitir pulverização seletiva e outras ações automatizadas baseadas em algoritmos de *Artificial Intelligence* (AI).

Essas aplicações são ilustradas na Figura 6, a qual apresenta uma visão geral das possibilidades habilitadas pela integração da rede de acesso 6G com tecnologias de IoT para o agronegócio inteligente.

Considerando a natureza dinâmica da rede e a diversidade de aplicações com diferentes requisitos de QoS, surge a necessidade de um escalonamento inter-fatias (*inter-slice*) inteligente dos recursos de rádio. Essa abordagem visa garantir o cumprimento dos SLAs firmados para cada tipo de aplicação ou perfil de usuário, alocando dinamicamente a capacidade da rede conforme as demandas variáveis do sistema. No contexto descrito, o desafio está em compatibilizar o uso eficiente do espectro com a garantia de desempenho para aplicações críticas, como o monitoramento de clima ou o rastreamento de gado, sem comprometer a estabilidade global da rede ou levar aplicações menos prioritárias à inanição.

Rede de acesso do Projeto Brasil 6G e potenciais aplicações de IoT integrado à Smart Farming

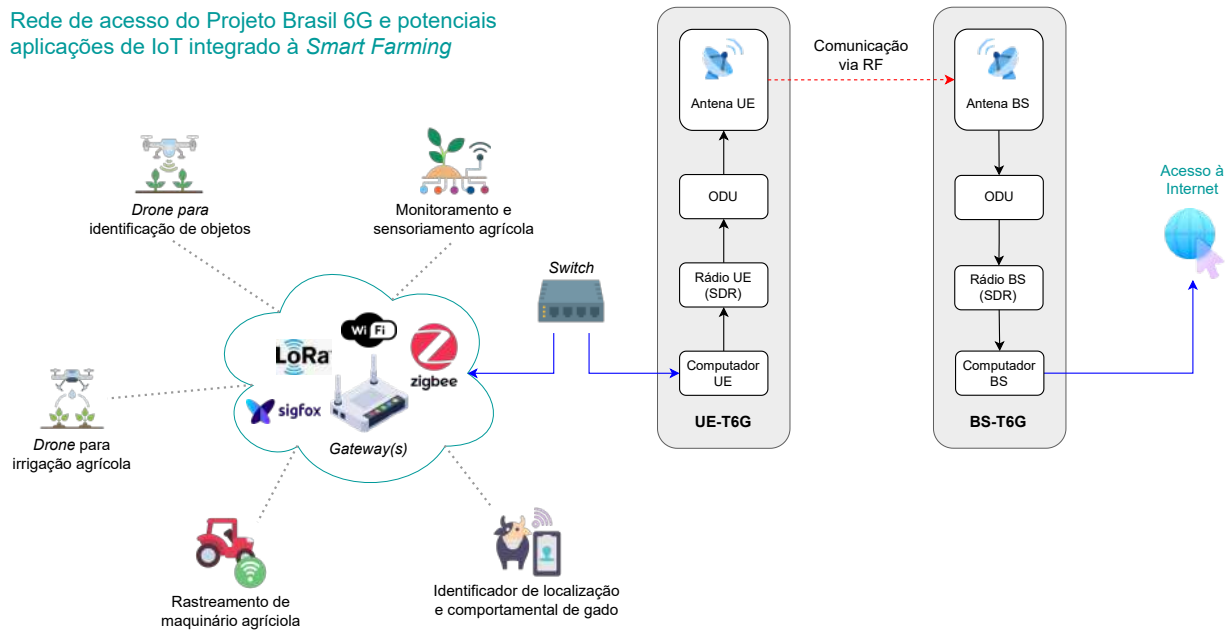


Figura 6: Visão geral da rede de acesso do Projeto Brasil 6G e potenciais aplicações relacionadas à IoT integrada ao agronegócio inteligente.

As próximas seções deste capítulo irão abordar, de forma detalhada, os fundamentos do escalonamento de recursos de rádio, contextualizando o papel do escalonador dinâmico inter-fatias e seu processo de implementação na camada *Medium Access Control* (MAC) dos transceptores da rede de acesso do Projeto Brasil 6G.

4.2 Escalonamento de recursos de rádio

Para habilitar a comunicação via rádio entre estação base e terminal de usuário, é necessário realizar o escalonamento de recursos de rádio, que consiste em alocar faixas de frequência para os terminais de usuário conectados a cada instante de tempo. Discretizamos tal escalonamento como uma matriz em que as colunas contam o tempo em unidades de *Transfer Time Interval* (TTI) e as linhas representam faixas de frequência da estação base, resultando em células que chamamos de *Physical Resource Block* (PRB). Portanto, o escalonador é o algoritmo que define quais PRBs são alocados a quais terminais de usuário em cada TTI.

Como a alocação de PRB define diretamente a capacidade e vazão atingidas pelos terminais de usuário, um escalonador eficiente é essencial para atingir requisitos de QoS. Entretanto, num cenário de alta heterogeneidade entre aplicações como o nosso, o qual pode incluir de veículos autônomos a dispositivos de sensoriamento remoto atrelados aos terminais, os padrões de QoS são discrepantes, dificultando o desenho de um escalonador geral. Portanto, utilizamos a tecnologia de *Network Slicing* (NS) para agrupar aplicações similares em “fatias”: redes lógicas com recursos isolados instanciadas numa mesma infraestrutura compartilhada. Utilizando o exemplo anterior, podemos definir a fatia 1 como agrupando veículos autônomos, enquanto dispositivos de sensoriamento remoto estão associados à fatia 2. Nos baseamos na possibilidade de um mesmo usuário estar associado a múltiplas fatias [27] para, num mesmo enlace entre estação base e terminal de usuário, isolarmos recursos entre diferentes fatias. A Figura 7 ilustra um cenário de NS com dois terminais de usuários conectando dispositivos de fazenda inteligente

a uma estação base, associando aplicações distintas a fatias distintas.

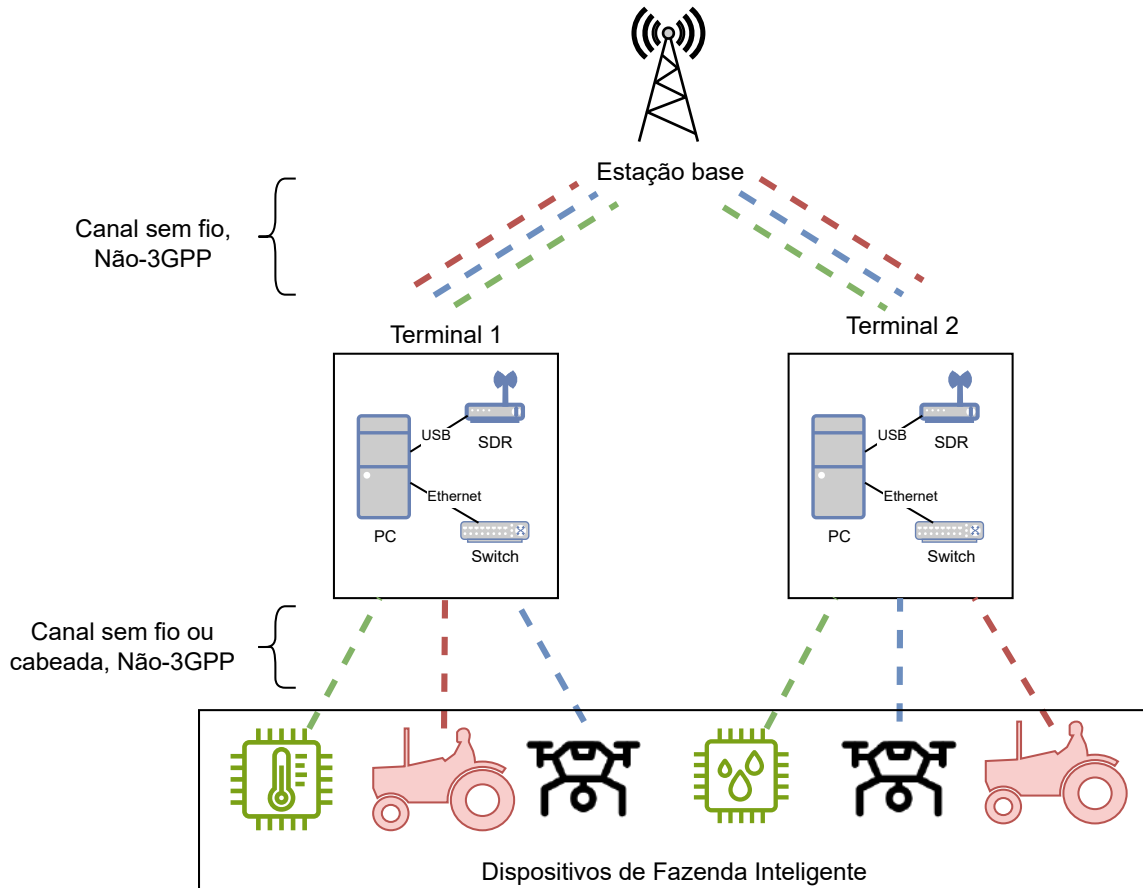


Figura 7: Cenário de fazenda inteligente com NS. Três fatias estão instanciadas: veículos autônomos (vermelho), dispositivos de sensoriamento (verde) e veículos aéreos não tripulados (azul).

No contexto de NS, o escalonamento passa então a possuir dois estágios. Primeiramente, o escalonador inter-fatias aloca PRBs entre as fatias, levando em consideração diferentes tipos de aplicação. Em seguida, cada fatia distribui os PRBs recebidos entre seus terminais de usuário associados, utilizando, como escalonador intra-fatia, um algoritmo que se adeque ao tipo de aplicação (como *maximum throughput* ou *proportional fair*). A Figura 8 ilustra o processo de escalonamento de recursos de rádio considerando um cenário de NS com três fatias.

[28] estuda o escalonamento ciente de canal, ou seja, o escalonamento em que é sabida a qualidade de canal que cada PRB tem para cada usuário, de forma que é definida não só a quantidade de PRBs alocados, mas são selecionados os melhores PRBs para cada usuário. O artigo aponta que, enquanto escalonadores intra-fatia costumam ser cientes de canal, escalonadores inter-fatia dificilmente utilizam de tal informação por não saberem para qual usuário o escalonador intra-fatia irá passar cada PRB alocado à fatia. Entretanto, ao considerar apenas algoritmos gulosos como escalonadores intra-fatia, é possível prever a alocação a nível de usuário durante a alocação entre fatias, habilitando um escalonador inter-fatias ciente de canal.

Além da ciência de canal, outro fator importante para o escalonador é o seu objetivo. En-

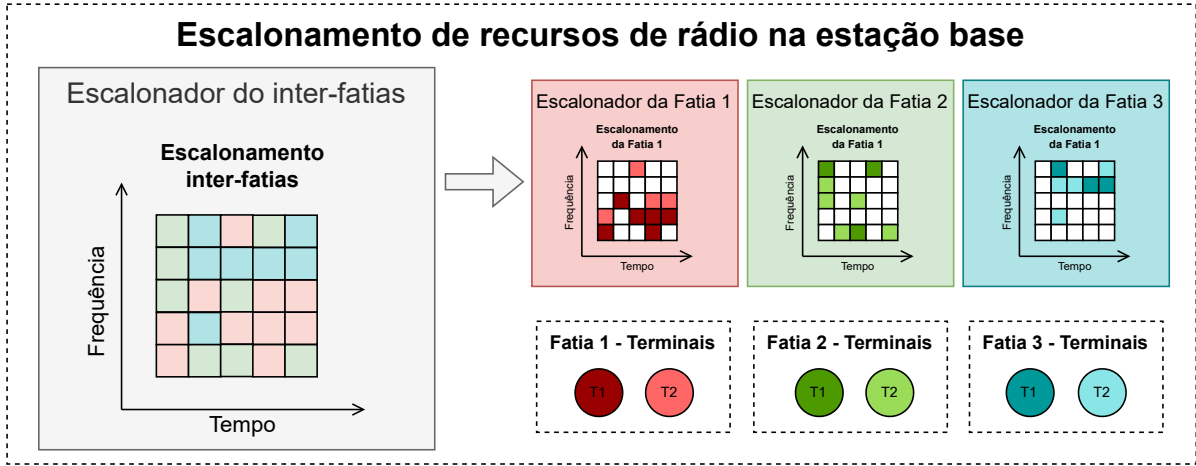


Figura 8: Ilustração do processo de escalonamento de 5 TTIs num cenário de NS com três fatias.

contramos na literatura diversos escalonadores focados em maximizar vazão, minimizar latência ou maximizar uso de banda. No entanto, tais objetivos podem levar ao superprovisionamento de usuários em cenários de abundância de recursos, em que a demanda é muito menor que a banda disponível. Portanto, optamos por guiar nosso escalonamento a garantir SLAs, expressados para cada fatia como uma coleção de valores almejados para métricas de rede, como vazão a longo prazo e latência de *buffer*. Para quantificar a garantia de SLA e diferenciar opções de alocação de acordo com o quão próximos estamos de cumprir os SLAs dos usuários, calculamos o *SLA-drift*: uma métrica indicando 0 caso o SLA foi cumprido e um valor positivo normalizado caso contrário, sendo 1 o valor para o maior descumprimento de SLA. Baseamos o *SLA-drift* no *intent-drift* desenvolvido em [29] e [30], também formulado de maneira similar em [31]. Agregamos o *SLA-drift* de um usuário como a média ponderada entre os *SLA-drifts* de seus requisitos de SLA, enquanto o *SLA-drift* de uma fatia é a média dos *SLA-drifts* de seus usuários associados e o *SLA-drift* da estação base é a média ponderada dos *SLA-drifts* das fatias instanciadas. Consideramos que os pesos das fatias e dos requisitos de SLA são configurados pelo operador de rede. Dessa forma, propomos implementar um escalonador inter-fatia que seja ciente de canal e guiado a minimizar o *SLA-drift* da estação base para simultaneamente garantir a *QoS* de diferentes tipos de aplicação em nosso cenário.

Nosso escalonador estará incluso na camada MAC da pilha de rádio da *Radio Access Network* (RAN), recebendo como entrada os SLAs das fatias instanciadas, os dados de canal dos PRBs de cada terminal de usuário e dados de demanda e estado do *buffer* dos usuários na estação base. Como saída, o escalonador definirá quais PRBs serão alocados para cada fatia, prevendo qual usuário receberá cada PRB e minimizando o *SLA-drift* da estação base.

4.3 Avanços obtidos: preparação da camada MAC para integração com escalonador externo

A necessidade de um mecanismo eficiente de escalonamento de recursos na rede de acesso do Projeto Brasil 6G torna-se ainda mais evidente diante das particularidades de seu cenário

de aplicação: o atendimento a regiões remotas, com demandas heterogêneas de conectividade e recursos de espectro escassos. Um escalonador inteligente e dinâmico permite não apenas maximizar a utilização do espectro disponível, mas também garantir qualidade de serviço diferenciada para cada tipo de aplicação, otimizando a experiência do usuário final, a eficiência espectral e o desempenho geral da rede.

Nesse contexto, a integração de um algoritmo escalonador externo à infraestrutura de rede requer uma interface eficiente com a camada MAC dos transceptores, de modo que essa camada seja capaz de transmitir ao escalonador informações relevantes sobre o estado atual da rede. Entre os parâmetros fundamentais a serem compartilhados destacam-se: (i) informações provenientes da camada *Physical* (PHY), como o valor de *Signal-to-Noise Ratio* (SNR) por recurso ou por usuário, e resultados de sensoriamento espectral local, indicando a ocupação de canais; (ii) e informações da própria camada MAC, como a vazão atual (*throughput*) por dispositivo, o *Modulation and Coding Scheme* (MCS) atualmente utilizado nas transmissões de *uplink* e *downlink*, bem como métricas relacionadas à ocupação e uso dos *Resource Blocks* (RBs).

No estado atual da rede de acesso do Projeto Brasil 6G, os transceptores operam com uma camada MAC em que todos esses parâmetros já são coletados e utilizados internamente por um algoritmo de alocação de recursos. Esse algoritmo atribui blocos de recursos RBs aos terminais conectados com base em:

- condições de canal, especialmente o MCS estimado a partir do SNR;
- estado atual de disponibilidade do espectro, obtido via sensoriamento espectral;
- configurações do transceptor, incluindo capacidades *Multiple-Input Multiple-Output* (MIMO);
- e quantidade de dados a serem transmitidos por cada terminal de usuário em um determinado instante.

Com base nesse funcionamento interno, vislumbra-se a viabilidade de desacoplar a lógica de escalonamento da própria camada MAC, permitindo a adoção de novas alternativas, como algoritmos mais avançados e de maior complexidade, executados em ambientes computacionais mais flexíveis, escritos em linguagens distintas do C++ utilizado atualmente na camada MAC, e até mesmo alocados em máquinas dedicadas. Essa estratégia também pode reduzir a carga computacional sobre a unidade que processa as funções PHY/MAC, viabilizando maior escalabilidade da solução.

Para possibilitar essa integração com um escalonador externo, a camada MAC foi preparada para implementar um mecanismo de comunicação inter-processos, responsável por:

- organizar os parâmetros de entrada relevantes, obtidos tanto da PHY quanto da própria MAC, de forma estruturada;
- transmitir essas informações ao algoritmo externo por meio de mensagens bem definidas;
- e receber a resposta do algoritmo com as decisões processadas, aplicando-as no funcionamento da rede.

Como etapa preliminar para validar o fluxo de comunicação inter-processos e preparar a futura integração do escalonador externo, optou-se por testar, inicialmente, o desacoplamento de uma das funções já existentes na camada MAC, antes de realizar diretamente a migração

da lógica de alocação de recursos. Essa abordagem progressiva visa garantir a viabilidade técnica do desacoplamento funcional, minimizando riscos e complexidade nos estágios iniciais de desenvolvimento. Para esse fim, foi selecionada a função de cálculo do MCS para transmissões de *downlink*, que atualmente é executada internamente na MAC por meio de uma *look-up table*, com base no valor médio de SNR reportado pela camada física. A escolha dessa função como primeiro caso de teste justifica-se por sua menor complexidade e por não exigir decisões dependentes de múltiplos usuários, como ocorre no escalonamento de recursos propriamente dito. Assim, o objetivo principal desta etapa foi verificar a viabilidade da transferência de decisão para um ambiente externo, mantendo a integridade funcional da camada MAC.

Para conduzir esse teste inicial, foi implementado um sistema de comunicação baseado no modelo *publish/subscribe*, em que ambos os componentes – a camada MAC e o algoritmo externo – assumem papéis de publicador e assinante em diferentes momentos do ciclo de comunicação. Nesse modelo, a camada MAC atua inicialmente como *publisher*, organizando e publicando uma mensagem contendo os parâmetros de entrada necessários, como o SNR médio, para o cálculo do MCS. O algoritmo externo, por sua vez, atua como *subscriber*, recebendo esses parâmetros, processando-os e determinando o MCS apropriado. Em seguida, o algoritmo externo assume o papel de *publisher*, publicando a resposta com o MCS calculado, enquanto a MAC, agora como *subscriber*, aguarda essa resposta e aplica a reconfiguração do sistema de transmissão de acordo com o novo valor recebido. Esse fluxo bidirecional de mensagens garante a integração efetiva entre os componentes e serve como prova de conceito para a execução externa de decisões que, até então, eram restritas ao escopo interno da MAC.

A definição dos protocolos de transmissão foi organizada em dois tipos de mensagens, contendo parâmetros para *uplink* e *downlink*. Cada uma dessas mensagens inclui métricas como a vazão, o MCS previamente utilizado e o valor médio de SNR do canal. Embora, neste estágio, o cálculo do MCS tenha sido realizado com base no SNR médio, os protocolos foram projetados para incluir também métricas adicionais, como o SNR por RB, que serão fundamentais, em etapas posteriores, para viabilizar decisões de escalonamento de recursos mais eficientes e precisas.

A Figura 9 ilustra o diagrama do fluxo de comunicação inter-processos entre a camada MAC e o algoritmo externo utilizado nos testes e previamente descrito, detalhando os protocolos de mensagens definidos, bem como a estrutura das informações publicadas e recebidas ao longo da interação.

Os testes foram conduzidos por meio de simulações com abstração da camada PHY, considerando um cenário básico de comunicação entre uma estação rádio-base (BS) e um terminal de usuário (UE), com a tarefa de seleção de MCS realizada externamente. Os resultados demonstraram o êxito da proposta: a camada MAC foi capaz de publicar corretamente os parâmetros exigidos e o algoritmo externo foi capaz de processar as informações e retornar a resposta. Além disso, a camada MAC também aplicou corretamente a decisão recebida na configuração da transmissão de *downlink*. Esse fluxo validou, portanto, o mecanismo de comunicação, a estrutura dos protocolos de mensagem e a viabilidade de execução de funções de decisão fora do domínio MAC.

Este avanço viabiliza a inicialização da etapa de integração de um escalonador de recursos inteligente e dinâmico, o qual substituirá o cálculo do MCS por uma lógica de alocação de RBs otimizada. A próxima seção apresenta os trabalhos futuros planejados, detalhando os passos necessários para a consolidação da integração entre a camada MAC e o escalonador, as adaptações previstas no algoritmo externo para sua operação na rede de acesso do Projeto Brasil 6G e a operação prática com cenários mais realistas, incluindo SDRs e múltiplos usuários.

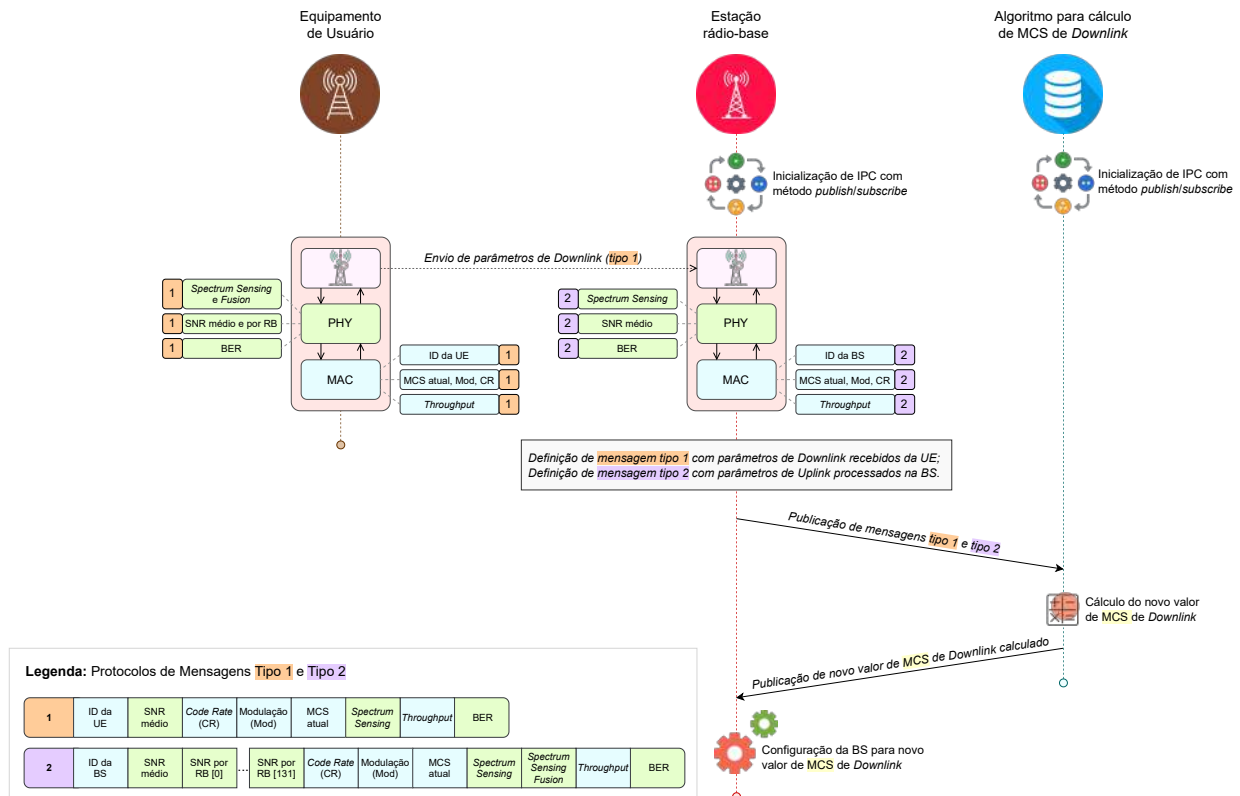


Figura 9: Diagrama de testes do fluxo de comunicação inter-processos entre camada MAC e algoritmo externo.

4.4 Atividades futuras para a validação e a integração do escalonamento de recursos

Realizada a validação inicial do mecanismo de comunicação inter-processos entre a camada MAC e um módulo externo de processamento, as próximas etapas do projeto concentram-se na transição do ambiente de simulação para condições operacionais mais realistas, bem como na consolidação da integração com o algoritmo de escalonamento de recursos.

Uma das atividades previstas consiste na migração dos testes para um ambiente prático com SDRs, substituindo a abstração da camada física utilizada nas simulações atuais. Essa etapa é fundamental para avaliar o impacto da operação em *hardware* real, considerando os aspectos de sincronização temporal, latência de comunicação, variabilidade do canal físico e eventuais limitações de desempenho dos dispositivos SDR utilizados. A análise deverá considerar, por exemplo, o tempo de resposta do sistema ao receber decisões externas de alocação, bem como a estabilidade do enlace em cenários reais de propagação e mobilidade.

Em paralelo, será incorporada a presença de múltiplos usuários simultâneos (UEs) à rede de acesso, dado que os testes realizados até o momento envolveram apenas um terminal. Com múltiplos UEs ativos, o algoritmo escalonador deverá lidar com demandas concorrentes de aplicações distintas, condições de canal variáveis e distribuição eficiente dos recursos de rádio entre os usuários. Essa evolução traz desafios adicionais ao sistema, como a priorização de tráfego com base em critérios de QoS, o balanceamento dinâmico do espectro e a manutenção da equidade no atendimento aos dispositivos conectados.

Outro foco essencial das atividades futuras é a integração plena com o algoritmo externo de escalonamento de recursos, substituindo o teste preliminar de cálculo de MCS por decisões reais de alocação de RBs. Para isso, será necessário definir com precisão o conjunto de parâmetros a serem coletados e organizados na camada MAC, incluindo métricas de desempenho da rede e informações de canal em tempo real. Esses dados deverão ser transmitidos ao algoritmo escalonador de forma estruturada, permitindo que este realize o processamento adequado e retorne decisões otimizadas de alocação de recursos.

Além disso, o próprio algoritmo escalonador deverá ser adaptado para garantir compatibilidade com o sistema de rádio da rede de acesso do Projeto Brasil 6G, respeitando, por exemplo, os intervalos de tempo de transmissão (TTIs) utilizados na camada MAC, de modo que as decisões de escalonamento sejam tomadas dentro de janelas temporais coerentes com a dinâmica do sistema. Essa adaptação requer a definição de uma política de sincronização entre o escalonador e a MAC, bem como o tratamento de casos de latência alta na resposta ou perda de mensagens no canal de comunicação inter-processos.

Essas atividades representam um avanço significativo rumo à implementação completa de um escalonamento de recursos inter-fatias inteligente e dinâmico, plenamente integrado à rede de acesso programável do Projeto Brasil 6G, capaz de operar com múltiplos usuários, sob condições reais de canal, e com suporte a diferentes perfis de aplicação.

5 Considerações Finais

Este relatório apresentou as atividades do grupo 2.4: *Concepção do Núcleo da Rede 6G do Projeto Brasil 6G*, destacando avanços significativos para a concepção de um núcleo de rede 6G flexível, inteligente e orientado a serviços. A partir de três frentes complementares, i.e., turismo inteligente, orquestração de tarefas imersivas em infraestruturas de nuvem e borda, e escalonamento de recursos de rádio em áreas remotas, o grupo de pesquisadores evidenciou como a rede 6G poderá sustentar aplicações cientes de contexto e sensíveis a latência, viabilizando experiências personalizadas, processamento distribuído eficiente e inclusão digital.

A proposta do +Tour demonstrou, por meio de dados reais e formalização do problema, ganhos expressivos em eficiência de alocação de recursos e experiência do usuário ao incorporar capacidades da borda na recomendação de roteiros turísticos. Já o *Local Optimal Task Orchestration Solution* (LOTOS) *framework* evidenciou o potencial de estratégias de orquestração locais para reduzir custos e aumentar a taxa de execução bem-sucedida de tarefas em ambientes de computação de borda, superando abordagens clássicas e lógica fuzzy em diversos cenários simulados. Por fim, os avanços na camada MAC para integração com escalonadores externos, bem como os testes com comunicação inter-processos, preparam o caminho para um mecanismo de escalonamento dinâmico e ciente de SLA que, além de eficiente, é compatível com as particularidades do cenário brasileiro de conectividade remota.

Esses resultados reforçam a importância da modularidade, programabilidade e inteligência distribuída na arquitetura do núcleo 6G, pavimentando o caminho para uma próxima etapa do projeto focada na validação prática com múltiplos usuários, integração de *hardware* real, e testes com ambientes heterogêneos e realistas. As contribuições científicas e experimentais consolidadas neste relatório fornecem uma base sólida para a continuidade do projeto e sinalizam um compromisso consistente com a geração de impacto acadêmico e tecnológico.

Referências

- [1] European Commission, “European Capitals of Smart Tourism,” https://smart-tourism-capital.ec.europa.eu/index_en, 2023, [Last access: Mar-26-2023].
- [2] T. H. Silva, A. C. Viana, F. Benevenuto, L. Villas, J. Salles, A. Loureiro, and D. Quercia, “Urban Computing Leveraging Location-Based Social Network Data: A Survey,” *ACM Comput. Surv.*, vol. 52, no. 1, 2019.
- [3] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, “Tour recommendation and trip planning using location-based social media: a survey,” *Knowledge and Information Systems*, vol. 60, no. 3, pp. 1247–1275, 2018.
- [4] S. Renjith, A. Sreekumar, and M. Jathavedan, “An extensive study on the evolution of context-aware personalized travel recommender systems,” *Information Processing & Management*, vol. 57, no. 1, p. 102078, 2020.
- [5] P. Yochum, L. Chang, T. Gu, and M. Zhu, “Linked Open Data in Location-Based Recommendation System on Tourism Domain: A Survey,” *IEEE Access*, vol. 8, pp. 16 409–16 439, 2020.
- [6] S. Halder, K. H. Lim, J. Chan, and X. Zhang, “A survey on personalized itinerary recommendation: From optimisation to deep learning,” *Applied Soft Computing*, vol. 152, 2024.
- [7] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu, “Automatic Construction of Travel Itineraries Using Social Breadcrumbs,” in *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, 2010, p. 35–44.
- [8] I. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso, “Where Shall We Go Today?: Planning Touristic Tours with Tripbuilder,” in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, 2013, pp. 757–762.
- [9] Z. Yu, H. Xu, Z. Yang, and B. Guo, “Personalized Travel Package With Multi-Point-of-Interest Recommendation Based on Crowdsourced User Footprints,” *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 151–158, 2016.
- [10] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, “Personalized trip recommendation for tourists based on user interests, points of interest, visit durations and visit recency,” *Knowledge and Information Systems*, vol. 54, no. 2, pp. 375–406, 2018.
- [11] L. Chen, L. Zhang, S. Cao, Z. Wu, and J. Cao, “Personalized itinerary recommendation: Deep and collaborative learning with textual information,” *Expert Systems with Applications*, vol. 144, p. 113070, 2020.
- [12] L. Chen, J. Cao, H. Tao, and J. Wu, “Trip Reinforcement Recommendation with Graph-Based Representation Learning,” *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 4, 2023.
- [13] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, P. Pillai, R. Klatzky, D. Siewiorek, and M. Satyanarayanan, “An Empirical Study of Latency in an Emerging Class of Edge Computing Applications for Wearable Cognitive Assistance,” in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, 2017, pp. 1–14.

- [14] P. Y. Zhou, S. Fu, B. Finley, X. Li, S. Tarkoma, J. Kangasharju, M. Ammar, and P. Hui, “5G MEC Computation Handoff for Mobile Augmented Reality,” in *2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*, 2024, pp. 129–136.
- [15] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, “A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.
- [16] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “A survey on algorithmic approaches for solving tourist trip design problems,” *Journal of Heuristics*, vol. 20, pp. 291–328, 2014.
- [17] D. Feillet, P. Dejax, and M. Gendreau, “Traveling Salesman Problems with Profits,” *Transportation Science*, vol. 39, no. 2, pp. 188–205, 2005.
- [18] S. Irnich and G. Desaulniers, *Shortest Path Problems with Resource Constraints*. Boston, MA: Springer US, 2005, ch. 2, pp. 33–65.
- [19] B. Meindl and M. Templ, “Analysis of commercial and free and open source solvers for linear optimization problems,” *Eurostat and Statistics Netherlands within the project ES-Snet on common tools and harmonised methodology for SDC in the ESS*, vol. 20, p. 64, 2012.
- [20] K. Dongwoo *et al.*, “Melbourne data set,” <https://github.com/dongwookim-ml/flickr-photo>, 2017, accessed: 2022-01-05.
- [21] T. W. Nowak *et al.*, “Verticals in 5G MEC-Use Cases and Security Challenges,” *IEEE Access*, vol. 9, pp. 87 251–87 298, 2021.
- [22] H. Wang *et al.*, “A Survey on the Metaverse: The State-of-the-Art, Technologies, Applications, and Challenges,” *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 671–14 688, 2023.
- [23] K. Toczé *et al.*, “Characterization and modeling of an edge computing mixed reality workload,” *Journal of Cloud Computing*, vol. 9, no. 1, p. 46, Dec. 2020.
- [24] C. Sonmez *et al.*, “Fuzzy Workload Orchestration for Edge Computing,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 769–782, 2019.
- [25] R. Buyya *et al.*, “Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,” in *2009 International Conference on High Performance Computing & Simulation*, 2009, pp. 1–11.
- [26] K. Toczé *et al.*, “Performance Study of Mixed Reality for Edge Computing,” in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, ser. UCC’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 285–294. [Online]. Available: <https://doi.org/10.1145/3344341.3368816>

- [27] 3GPP, “5G; System architecture for the 5G System (5GS),” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 2025, version 18.9.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>
- [28] Y. Chen, R. Yao, H. Hassanieh, and R. Mittal, “Channel-Aware 5G RAN Slicing with Customizable Schedulers,” in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 1767–1782.
- [29] C. V. Nahum, V. H. Lopes, R. M. Dreifuerst, P. Batista, I. Correa, K. V. Cardoso, A. Klautau, and R. W. Heath, “Intent-aware radio resource scheduling in a RAN slicing scenario using reinforcement learning,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2253–2267, 2023.
- [30] C. Nahum, S. D’Oro, P. Batista, C. Both, K. Cardoso, A. Klautau, and T. Melodia, “Intent-based radio scheduler for ran slicing: Learning to deal with different network scenarios,” *arXiv preprint arXiv:2501.00950*, 2025.
- [31] A. Balasingam, M. Kotaru, and P. Bahl, “{Application-Level} service assurance with 5g {RAN} slicing,” in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 841–857.