Brasil 6G

# Projeto Brasil 6G
# Fase III

# Atividade 2.4 - Concepção do Núcleo da Rede 6G - Parte 2


31/07/2026

**Histórico de Atualizações:**

| Versão | Data | Autor(es) | Notas |
|---|---|---|---|
| 1 | 13/07/2026 | Kleber Vieira Cardoso<br>Luciano de Souza Fraga<br>Rafael Rodrigues Silva<br>Cristiano Bonato Both | Elaboração de conteúdo |
| 2 | 31/07/2026 | Kleber Vieira Cardoso<br>Daniely Gomes Silva<br>Luciano Leonel Mendes | Revisão de texto |

# Lista de Figuras

# Lista de Tabelas

# Acrônimos

**3GPP** *3rd Generation Partnership Project*
**5G** Quinta Geração de Rede Móvel Celular
**6G** Sexta Geração de Rede Móvel Celular
**API** *Application Programming Interface*
**AWGN** *Additive White Gaussian Noise*
**C2N** *CAMARA-to-NEF*
**CAPIF** *Common API Framework for 3GPP Northbound APIs*
**DNAI** *Data Network Access Identifier*
**DNN** *Data Network Name*
**FE** *Far Edge*
**FLOPs** *Floating-point Operations Per Second*
**gNB** *Next-Generation NodeB*
**JSCC** *Joint Source-Channel Coding*
**MEC** *Multi-access Edge Computing*
**MINLP** *Mixed Integer Nonlinear Programming*
**NaaS** *Network as a Service*
**NEF** *Network Exposure Function*
**NE** *Near Edge*
**OFDMA** *Orthogonal Frequency-Division Multiple Access*
**QoE** *Quality of Experience*
**RB** *Resource Block*
**RTT** *Round-Trip Time*
**SemCom** *Semantic Communication*
**SNR** *Signal-to-Noise Ratio*
**S-NSSAI** *Single Network Slice Selection Assistance Information*
**UE** *User Equipment*
**UPF** *User Plane Function*
**ViT** *Vision Transformer*
**XR** *Extended Reality*

# Índice

# 1 Introdução Geral

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 entre comunicação, computação e dados. Ao contrário das gerações anteriores, o núcleo da rede 6G deverá ser altamente adaptável e programável, permitindo que aplicações e serviços interajam dinamicamente com funcionalidades da rede e recursos computacionais distribuídos. Essa nova abordagem abre espaço para a exploração de mecanismos avançados de exposição de capacidades de rede, orquestração de serviços e alocação inteligente de recursos, criando um ecossistema capaz de atender às demandas de aplicações emergentes, como sistemas distribuídos sensíveis à latência, serviços baseados em computação de borda e aplicações orientadas à qualidade de experiência (QoE).

No contexto da programabilidade da rede, a Seção 2 apresenta um resumo do trabalho "C2N: uma Função de Transformação modular para a exposição de capacidades de rede via APIs CAMARA", publicado no 44º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2026). Esse trabalho ilustra como a concepção do núcleo da rede 6G pode viabilizar o paradigma *Network as a Service* (NaaS), permitindo que capacidades avançadas da rede sejam disponibilizadas para aplicações externas por meio de interfaces padronizadas e orientadas ao desenvolvedor. A proposta de uma Função de Transformação modular para traduzir chamadas das *Application Programming Interfaces* (APIs) CAMARA para operações do núcleo da rede contribui para a construção de um ambiente mais aberto, interoperável e flexível. Dessa forma, o trabalho reforça a visão de um núcleo 6G capaz de expor funcionalidades de rede de maneira transparente, simplificando o desenvolvimento de novos serviços e ampliando as possibilidades de inovação sobre a infraestrutura de telecomunicações.

No domínio da integração entre comunicação e computação, a Seção 3 apresenta um resumo do trabalho "Offloading de Tarefas com Comunicação Semântica para Alocação de Recursos Orientada à Qualidade de Experiência em Computação de Borda", atualmente sob avaliação para publicação no Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2026). Esse trabalho evidencia a importância de um núcleo da rede 6G capaz de coordenar eficientemente recursos de comunicação e processamento distribuídos entre dispositivos, borda e nuvem. Ao integrar conceitos de comunicação semântica, computação de borda e otimização orientada à QoE, a proposta demonstra como decisões inteligentes sobre descarregamento (*offloading*), compressão semântica e alocação de recursos podem melhorar simultaneamente latência, consumo energético e acurácia das aplicações. Essa capacidade de orquestração adaptativa e orientada ao contexto reforça o papel do núcleo 6G como elemento central na viabilização de serviços inteligentes, escaláveis e sensíveis às necessidades dos usuários.

# 2 C2N: uma Função de Transformação modular para a exposição de capacidades de rede via APIs CAMARA

*Rafael Rodrigues Silva (UFG), João Paulo Esper (UFG), Leandro C. de Almeida (IFPB), Fábio L. Verdi (UFSCar), Kleber Vieira Cardoso (UFG)*

## 2.1 Introdução

A evolução do núcleo da rede em direção ao 6G não se limita ao aumento de capacidade e à redução de latência: ela pressupõe um núcleo profundamente orientado a serviços, capaz de expor suas capacidades a aplicações de terceiros de forma programável. Esse princípio é a base do paradigma de NaaS, no qual o operador deixa de ser apenas provedor de conectividade e passa a oferecer funcionalidades de rede, direcionamento de tráfego, qualidade de serviço sob demanda, localização de dispositivos, por meio de APIs [1]. A concretização do NaaS é, portanto, um requisito direto da concepção de um núcleo 6G flexível e monetizável.

Historicamente, as APIs de rede eram complexas e proprietárias, resultando em um ecossistema fragmentado que dificultava a adoção por desenvolvedores. Para endereçar essa fragmentação, a iniciativa GSMA Open Gateway, em colaboração com o projeto CAMARA da Linux Foundation [2], padroniza um conjunto de APIs orientadas ao desenvolvedor (denominadas *Service APIs*) que abstraem os parâmetros específicos das interfaces de rede. No núcleo 5G/6G, o ponto de entrada padronizado para aplicações externas é a *Network Exposure Function* (NEF), especificada pelo *3rd Generation Partnership Project* (3GPP) [3]. Configurar capacidades diretamente pela NEF exige o fornecimento de parâmetros de baixo nível, como *Single Network Slice Selection Assistance Information* (S-NSSAI), *Data Network Name* (DNN) e descrições de fluxo em formato de 5-tupla IP, o que pressupõe a familiaridade do desenvolvedor com a especificação 3GPP.

A arquitetura CAMARA prevê um componente, denominado Função de Transformação, responsável por traduzir as chamadas das *Service* APIs orientadas ao desenvolvedor em operações das *Network* APIs do núcleo da rede. Entretanto, as especificações CAMARA definem apenas o papel conceitual desse componente, deixando seu projeto concreto em aberto. Como consequência, cada operadora desenvolve sua própria implementação proprietária, cujos padrões de mapeamento semântico e decisões arquiteturais não são documentados de forma acessível à comunidade acadêmica. Essa lacuna dificulta a experimentação prática com APIs CAMARA, a comparação de estratégias de tradução e o estudo de propriedades como a reutilização modular entre diferentes APIs e domínios de *backend*. Trabalhos recentes endereçam partes do problema da exposição de capacidades de rede, como implementações nativas em nuvem da NEF integradas ao *Common API Framework for 3GPP Northbound APIs* (CAPIF) [4], mas não tratam da tradução para as *Service APIs* orientadas ao desenvolvedor.

Este trabalho apresenta o *CAMARA-to-NEF* (C2N), uma Função de Transformação modular e de código aberto, validada por meio da *Traffic Influence* API, a API CAMARA que permite a uma aplicação solicitar o direcionamento do tráfego do plano de usuário para localizações específicas da rede, capacidade essencial para serviços sensíveis à latência que se beneficiam de computação de borda. O trabalho foi aceito na trilha principal do 44º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2026).

## 2.2 Solução Proposta

O C2N implementa uma arquitetura de *gateway* modular projetada para suportar múltiplas *Service* APIs CAMARA sobre múltiplos *backends* de *Network* APIs, separando a infraestrutura reutilizável da lógica de tradução específica por API. O *gateway* é posicionado no domínio do provedor de APIs, recebendo chamadas autenticadas pelo CAPIF [5] e encaminhando as requisições traduzidas ao *backend*, que na instância atual é a NEF. A Figura 1 apresenta o posicionamento do C2N no ecossistema CAMARA e sua organização interna.

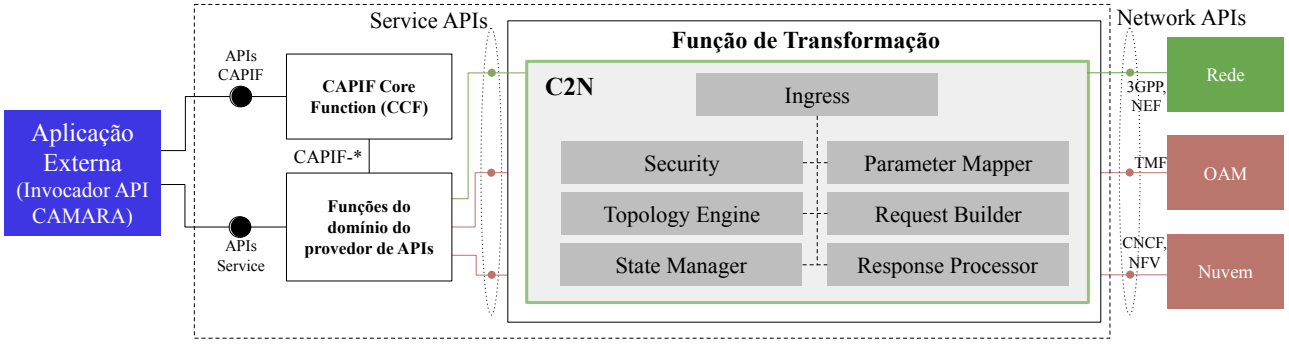


Figura 1: Visão geral da arquitetura do C2N no ecossistema CAMARA. A Função de Transformação conecta as *Service* APIs às *Network* APIs expostas pela NEF 3GPP (Rede), por sistemas de Operação e Gerenciamento (OAM) e pela infraestrutura de nuvem (Nuvem).

A arquitetura é composta por sete módulos organizados em quatro camadas de processamento. A camada de ingresso reúne o módulo de segurança, responsável pela autenticação e autorização (verificação de *tokens*, controle de acesso baseado em papéis, limitação de taxa), e o módulo de ingresso, que valida a carga útil da requisição contra a especificação da *Service* API. A camada de enriquecimento e mapeamento contém o *Topology Engine*, que resolve identificadores abstratos em parâmetros de rede, e o *Parameter Mapper*, que executa a tradução semântica propriamente dita. A camada de construção e estado abrange o *Request Builder*, que monta a requisição para o *backend*, e o *State Manager*, que mantém os mapeamentos bidirecionais entre identificadores de recursos CAMARA e do *backend*. Por fim, a camada de egresso contém o módulo de resposta, que converte a resposta do *backend* de volta ao formato CAMARA.

A tradução semântica é organizada em uma estratégia de três fases, ordenadas por complexidade crescente. O mapeamento direto aplica-se a campos com correspondência um-para-um entre as duas APIs, sem transformação de conteúdo. A transformação lógica aplica-se a campos que exigem resolução de contexto de topologia, por exemplo, um único identificador abstrato de zona de borda é expandido em múltiplos parâmetros de rede (DNN, S-NSSAI e *Data Network Access Identifier* (DNAI)). O mapeamento estrutural aplica-se a conversões que envolvem reestruturação de dados, como a geração de descrições de fluxo em formato de regra de filtro IP a partir do modelo simplificado de filtros de tráfego do CAMARA. Essa taxonomia de três categorias cobre todos os mapeamentos identificados para a *Traffic Influence* API e é proposta como generalizável a outras APIs CAMARA.

Uma contribuição arquitetural é a separação entre componentes agnósticos à API e específicos por API. A análise da implementação indica que cinco dos sete módulos (*Security*, *Ingress*, *Request Builder*, *State Manager* e *Response*, são reutilizáveis entre diferentes APIs CAMARA, pois tratam preocupações comuns a qualquer tradução (autenticação, validação, operações de

rede, correlação de identificadores e formatação de resposta). Apenas o *Topology Engine* e o *Parameter Mapper* precisam ser reimplementados por API. Essa proporção de aproximadamente 70% de módulos reutilizáveis sustenta a hipótese de que a adição de uma nova *Service* API exige esforço de implementação reduzido. O C2N é implementado na linguagem Go, com o arcabouço Gin para roteamento HTTP, e adota uma política de dependências mínimas e configuração externa via variáveis de ambiente e arquivos JSON, o que permite que operadores adaptem o *gateway* ao seu ambiente sem modificar o código-fonte. A implementação está disponível publicamente para reprodutibilidade.

## 2.3 Resultados

O C2N foi avaliado em um ambiente de testes 5G composto inteiramente por software livre, baseado no núcleo free5GC e no simulador de rede de acesso UERANSIM. O ambiente, ilustrado na Figura 2, adota uma topologia com duas *User Plane Functions* (UPFs): um caminho padrão, no qual é introduzida latência artificial para simular degradação de rede, e um caminho alternativo otimizado para o qual o tráfego deve ser redirecionado. A própria figura ilustra a lacuna de complexidade que o C2N reduz: a requisição CAMARA (205 bytes), com campos intuitivos, é convertida em uma requisição NEF de 402 bytes, com os parâmetros técnicos detalhados exigidos pelo *backend*.

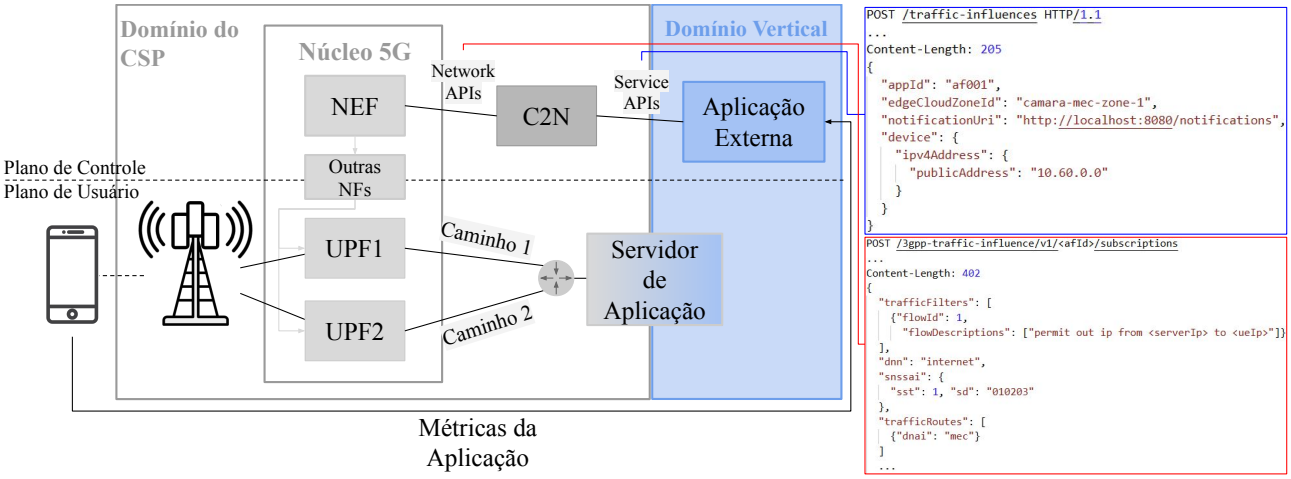


Figura 2: Arquitetura do ambiente de testes. O C2N opera no domínio do provedor de comunicação, entre a aplicação externa e a NEF, com topologia de dois UPFs para validação do redirecionamento de tráfego.

A avaliação foi organizada em dois eixos. No eixo funcional, um protocolo de três fases (referência, degradação e ativação do *Traffic Influence*) demonstrou que o C2N traduz corretamente as chamadas CAMARA em subscrições NEF que produzem redirecionamento real do tráfego no plano de dados. A Figura 3 apresenta o tempo de ida e volta (*Round-Trip Time* (RTT)) ao longo de um experimento de 60 segundos: após a ativação do *Traffic Influence*, o RTT retorna a valores próximos aos da fase de referência, confirmando a migração do caminho degradado para o caminho otimizado.

No eixo de desempenho, a instrumentação do ambiente permitiu decompor a latência do fluxo de sinalização do plano de controle, ilustrado na Figura 4. O procedimento abrange três fases: a criação da subscrição (síncrona, percebida pela aplicação), o controle de política (assíncrono) e a configuração do plano de usuário (assíncrona). A sobrecarga de tradução

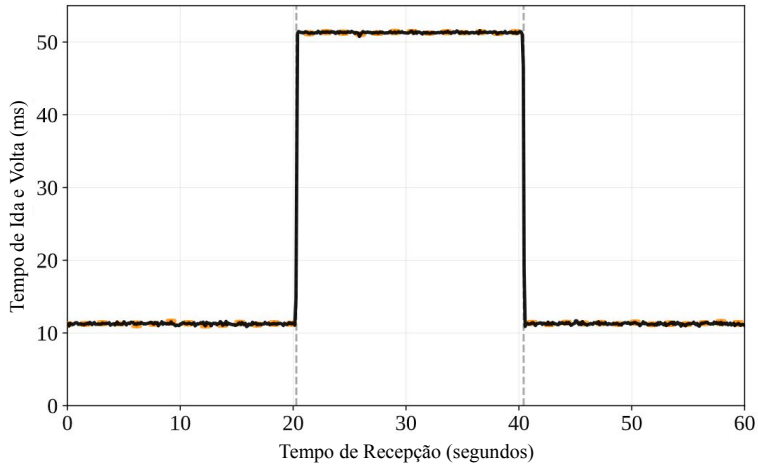


Figura 3: Experimento de 60 segundos: RTT médio por janela de pacotes. A fase de referência e a fase com *Traffic Influence* ativo apresentam latência equivalente, evidenciando o redirecionamento bem-sucedido do tráfego.

introduzida pelo C2N foi de 3,22 ms, o que representa menos de 2% da latência total de ponta a ponta de aproximadamente 172 ms. A maior parcela dessa latência total corresponde à fase de configuração do plano de usuário (cerca de 150 ms), inerente ao procedimento 3GPP em [6] e independente da existência da camada de mediação. Conclui-se, portanto, que a Função de Transformação não constitui gargalo de desempenho no fluxo de sinalização 5G.

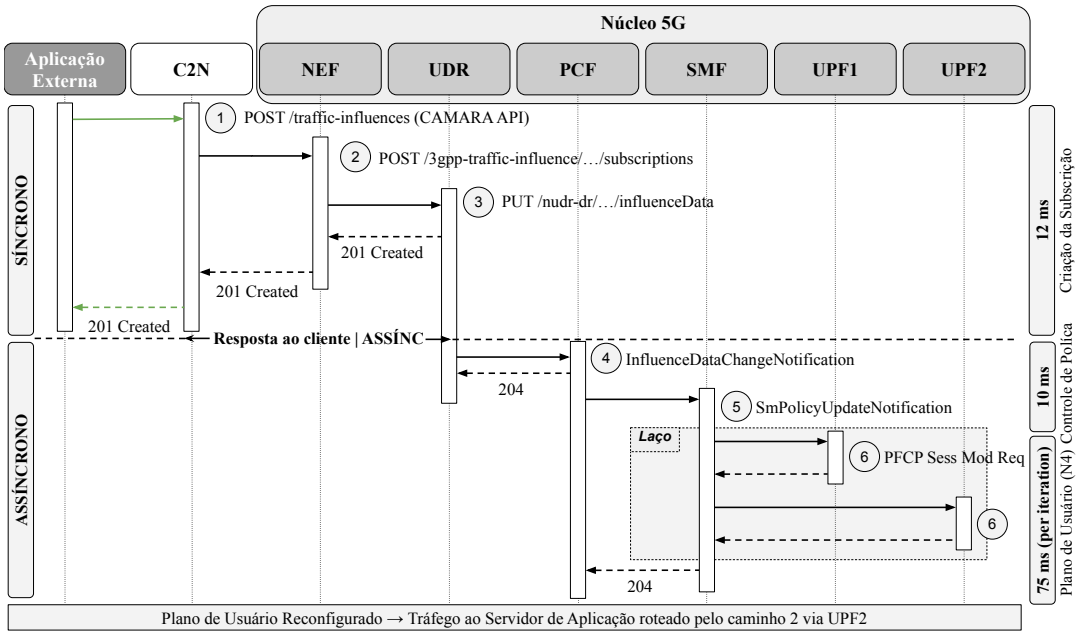


Figura 4: Fluxo de sinalização do núcleo 5G para a ativação do *Traffic Influence*, com as três fases do procedimento e os tempos medidos em cada uma.

## 2.4 Conclusão

O C2N demonstra que a Função de Transformação do ecossistema CAMARA, componente essencial para a exposição programável de capacidades do núcleo da rede, pode ser implementada de forma modular, documentada e reproduzível. O trabalho oferece três contribuições principais: uma taxonomia de transformações semânticas em três categorias (direta, lógica e estrutural); uma arquitetura de sete módulos em quatro camadas que separa a infraestrutura reutilizável da lógica específica por API; e uma implementação de código aberto validada experimentalmente, com caracterização da sobrecarga de tradução e do comportamento de sinalização do plano de controle. Os resultados confirmam que a camada de mediação introduz sobrecarga desprezível frente à latência total do procedimento de sinalização 5G.

Esse trabalho reforça, no contexto da concepção do núcleo da rede 6G, a importância de uma camada de exposição transparente e padronizada: ao tornar a Função de Transformação um artefato aberto e documentado, o C2N viabiliza a experimentação acadêmica com o paradigma NaaS e fornece uma base extensível para a integração de novas APIs e domínios de *backend*. Como continuidade, planeja-se a expansão do *gateway* para o domínio de nuvem, por meio da *Edge Application Management* API sobre plataformas de computação de borda, o que permitirá verificar empiricamente a reutilização modular da arquitetura em um cenário no qual uma única chamada CAMARA coordena operações em múltiplos *backends*.

# 3 Descarregamento de Tarefas com Comunicação Semântica para Alocação de Recursos Orientada à Qualidade de Experiência em Computação de Borda

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

## 3.1 Introdução

A Sexta Geração de Rede Móvel Celular (6G) promete impulsionar o crescimento na geração de dados e nas demandas computacionais de aplicações inteligentes, como direção autônoma e realidade estendida (*Extended Reality* (XR)) [7, 8]. Nesse contexto, a computação de borda multiacesso (*Multi-access Edge Computing* (MEC)) permite que equipamentos do usuário (*User Equipment* (UE)) realizem descarregamento (*offloading*) de tarefas intensivas para servidores remotos, reduzindo atrasos de processamento e consumo de energia [9, 10]. Entretanto, a eficiência dos sistemas MEC é limitada pelo gargalo de comunicação dos enlaces sem fio, especialmente em cenários com grande volume de dados e condições de canal adversas [11, 12, 13].

Os sistemas de comunicação convencionais seguem o paradigma de *Shannon*, focado na transmissão confiável de bits independentemente do significado da informação. Embora adequado para aplicações centradas em dados, esse modelo é ineficiente em aplicações orientadas a tarefas, nas quais apenas parte das informações transmitidas é relevante para o resultado final [8, 7]. Como alternativa, a comunicação semântica (*Semantic Communication* (SemCom)) busca transmitir significado em vez de bits, reduzindo a carga de comunicação por meio da extração de características relevantes à tarefa [7, 14, 15, 16]. Quando integrada à MEC, a SemCom pode melhorar significativamente a eficiência no processo de *offloading* [17, 18, 11].

Apesar dos avanços recentes, a maioria dos trabalhos que usam MEC sensível à semântica não modela explicitamente a relação entre condições do canal, compressão semântica e acurácia de inferência [11, 13, 17]. Além disso, muitos estudos assumem um único servidor de borda, ignoram a latência da rede cabeada e utilizam métodos baseados em aprendizado por reforço, frequentemente complexos, implicando maior latência no processo de decisão [17, 18].

Diante dessas limitações, este trabalho propõe um modelo de *offloading* sensível à semântica que considera conjuntamente compressão semântica, condições de comunicação e recursos computacionais distribuídos. As principais contribuições apresentadas são: (1) investigação da relação entre extração semântica, condições do canal e acurácia de inferência; (2) modelo de alocação de recursos que considera múltiplos pontos de *offloading*; e (3) desenvolvimento de uma heurística para resolver eficientemente o problema de otimização não linear (MINLP).

## 3.2 Solução Proposta

Consideramos um sistema de *offloading* de tarefas sensível à semântica composto por usuários $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$ conectados a uma *Next-Generation NodeB* (gNB) em uma rede 5G/6G. A gNB utiliza *Orthogonal Frequency-Division Multiple Access* (OFDMA) para alocar recursos aos usuários, assumindo-se, por simplicidade, um bloco de recursos (*Resource Block* (RB)) por UE e disponibilidade suficiente de RBs.

Cada UE gera tarefas de classificação de imagens que podem ser executadas localmente ou alocadas em servidores remotos $\mathcal{S} = \{1, \ldots, |\mathcal{S}|\}$, incluindo servidores de borda e de nuvem com

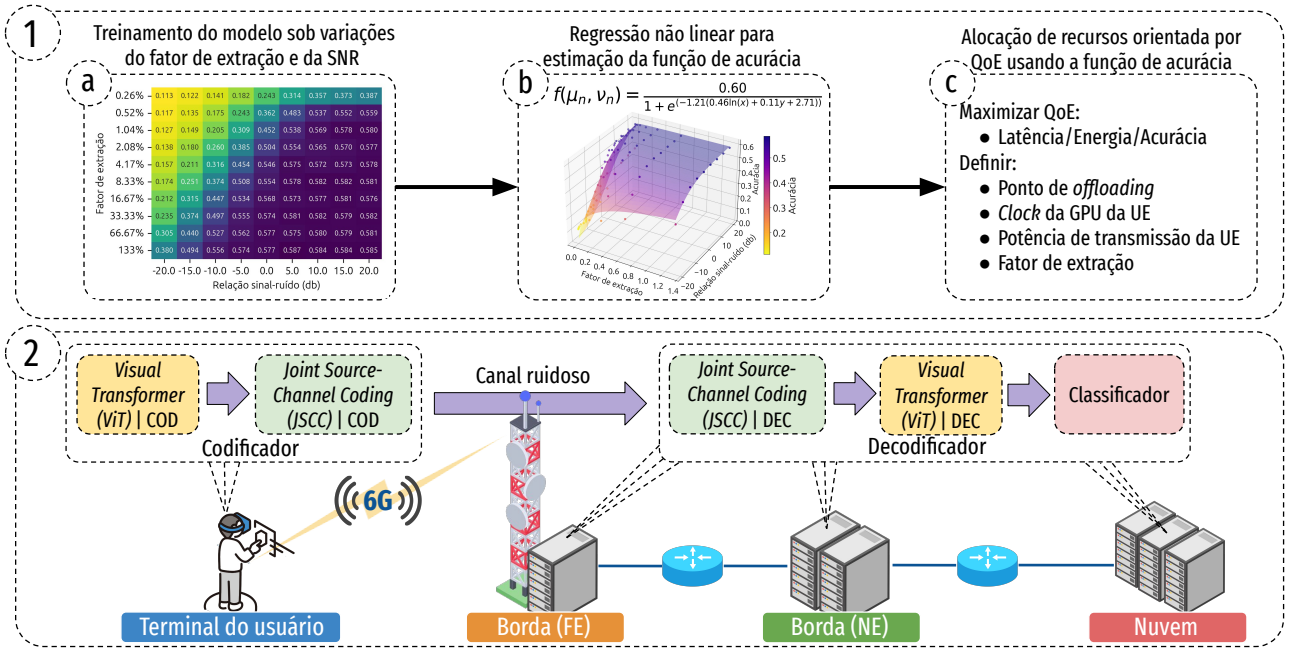


Figura 5: Sistema de comunicação semântica em que a função de acurácia é obtida para aplicação em sistemas de alocação de recursos (parte superior), considerando uma arquitetura de *offloading* de tarefas com múltiplos pontos de alocação (parte inferior).

diferentes capacidades computacionais. Cada UE $n$ aplica um codificador semântico com fator de extração $\mu_n \in [0, 1]$, responsável pela compressão semântica dos dados. Quando o *offloading* é realizado, a potência de transmissão $p_n$ também influencia a acurácia da inferência. Para modelar esse efeito, utilizamos regressão não linear sobre dados obtidos de modelos treinados sob diferentes fatores de extração e relação sinal-ruído (*Signal-to-Noise Ratio* (SNR)), resultando em uma função de acurácia que pode ser empregada no processo de otimização.

A Figura 5 apresenta o sistema proposto. Inicialmente, são coletados dados de acurácia para diferentes configurações de extração semântica e SNR. Em seguida, aplica-se regressão não linear para obter uma função de acurácia que relaciona compressão semântica e qualidade da transmissão. Essa função é então incorporada a um mecanismo de alocação de recursos que otimiza conjuntamente latência, consumo energético e acurácia. O sistema de classificação é composto por codificadores semântico e de canal executados no UE, enquanto os módulos de decodificação e classificação podem ser processados localmente ou em servidores remotos. Foi utilizado um *Vision Transformer* (ViT) [19] e um codificador de fonte e canal (*Joint Source-Channel Coding* (JSCC)) [20] na implementação do codificador e decodificador.

**Extração semântica** - A quantidade de dados transmitidos pelo usuário $n$ após a aplicação de um codificador semântico com uma taxa de compressão $\mu_n$ é $\widetilde{D}_n = \mu_n D_n$, onde $D_n$ representa o tamanho original dos dados. O tempo de extração semântica pode ser definida como $t_n^{\text{ext}} = 0$ se $\mu_n = 1$, e $t_n^{\text{ext}} = \frac{l_n^{SE}}{\mu_n c_n}$ se $\mu_n^{\min} \leq \mu_n < 1$, onde $l_n^{SE}$ é o consumo de GPU no UE para a extração semântica. $c_n = \alpha NU f_n$ representa a capacidade computacional da GPU do UE $n$. O parâmetro $\alpha = 2$ corresponde ao número de operações de ponto flutuante (*Floating-point Operations Per Second* (FLOPs)) por ciclo por núcleo (FLOPs/ciclo/núcleo). $NU$ representa o número de núcleos CUDA disponíveis no UE $n$. $f_n$ denota a frequência de clock da GPU do UE $n$.

**Modelo de comunicação** - A taxa de transmissão alcançável pelo o usuário $n$ é $R_n =$

$B \log_2 \left(1 + \frac{|h_n|^2 p_n}{\sigma^2}\right)$, onde $B$ é a largura de banda, $p_n$ a potência de transmissão, $h_n$ o ganho de canal, e $\sigma^2$ a potência do ruído branco gaussiano aditivo (*Additive White Gaussian Noise* (AWGN)). A latência da rede é composta pela latência de transmissão sem fio e pela latência da rede cabeada entre a gNB e o servidor remoto $s$, denotada por $t_s^{\text{wired}}$, e pode ser formulada como $t_n^{\text{tx}} = \frac{\widetilde{D}_n}{R_n} + t_s^{\text{wired}}$.

**Modelo de computação** - Caso seja realizado o *offloading*, o tempo de execução no servidor remoto $s$ é $t_n^{\text{rem}} = \frac{\sum_{n=1}^{\mathcal{N}} l_n^s}{c_s}$, onde $c_s = \alpha NS f_s$ representa a capacidade computacional da GPU do servidor $s$. O parâmetro $l_n^s$ corresponde ao consumo de GPU da tarefa associada ao UE $n$ no servidor $s$. $NS$ denota o número de núcleos CUDA disponíveis no servidor, enquanto $f_s$ representa a frequência de *clock* da GPU do servidor. A latência de execução local pode ser expressa como $t_n^{\text{loc}} = \frac{l_n^U}{c_n}$, onde $l_n^U$ representa a carga total de processamento da tarefa alocada ao UE $n$. Por fim, a latência total de processamento e de rede pode ser formulada como $t_n = \rho_n\left(t_n^{\text{ext}} + t_n^{\text{tx}} + t_n^{\text{rem}}\right) + (1-\rho_n) t_n^{\text{loc}}$, onde $\rho_n = \sum_{s=1}^{S} x_n^s$ representa o indicador de *offloading* (1 = *offloading* após a extração semântica, 0 = sem *offloading*). A variável de decisão $x_n^s = 1$ indica que a carga de processamento do usuário $n$ é alocada no servidor $s$; caso contrário, $x_n^s = 0$.

**Consumo energético** - O gasto total de energia para o processamento de uma tarefa é determinado pela energia consumida no dispositivo do usuário e no servidor remoto. O consumo de energia no lado do usuário, $E_n^U$, considera tanto a computação local e a extração semântica quanto a transmissão sem fio: $E_n^U = \kappa_n \left(\rho_n t_n^{\text{ext}} + (1-\rho_n) t_n^{\text{loc}}\right) f_n^3 + p_n (t_n^{\text{tx}} - t_s^{\text{wired}})$. O primeiro termo representa o consumo de energia da GPU do usuário $n$, considerando a extração semântica e o processamento local. $\kappa_n$ corresponde ao coeficiente energético do UE $n$. O segundo termo contabiliza a energia consumida durante a transmissão sem fio dos dados codificados semanticamente. A energia computacional consumida pelo servidor $s$ é dada por: $E_s = \kappa_s t_n^{\text{rem}} f_s^3$, onde $\kappa_s$ é o coeficiente energético do servidor remoto $s$. A energia total consumida para atender às demandas do UE $n$ é dada por: $E_n = E_n^U + E_s$.

**Função de acurácia** - Para capturar a relação entre a extração semântica $\mu_n$, SNR $\nu_n$ e a acurácia de inferência $\varepsilon_n$, aplicamos regressão não linear aos dados coletados durante a avaliação do modelo de classificação de imagens (Figura 5(a)). A função utilizada como base para a regressão pode ser definida como $f(\mu_n, \nu_n) = \frac{\mathbf{L}}{1+e^{-\mathbf{k}(\mathbf{a}(ln(\mu_n)+\mathbf{b}\nu_n+\mathbf{c}))}}$, onde os parâmetros $\mathbf{L}$, $\mathbf{k}$, $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$ são obtidos ao final do processo de regressão.

**Função de QoE** - O desempenho do modelo de alocação de recursos está relacionado à latência, ao consumo energético e à acurácia da inferência. A Equação 9 representa a função de QoE normalizada, onde $\sigma(x) = 1/(1+e^{-x})$ é a função logística, $\{\omega_t, \omega_e, \omega_a\}$ são pesos, tais que $\omega_t + \omega_e + \omega_a = 1$, e $t_n^l$, $E_n^l$ e $\varepsilon_n^l$ são valores de referência. Por fim, a função de acurácia é definida como $\varepsilon_n = f(\mu_n, \nu_n)$.

**Formulação do problema** - O objetivo geral do problema é maximizar a QoE agregada por meio da escolha de $\{\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f}\}$, conforme definido nas Equações 1-9. O problema pode ser classificado como um problema de programação não linear inteira misto (MINLP), contendo variáveis de decisão inteiras e contínuas, bem como diversas funções não lineares, o que torna desafiadora a obtenção de sua solução ótima.

**Heurística** - Propomos uma heurística gananciosa, apresentada no Algoritmo 1, para maximizar a QoE por meio de duas etapas: otimização de decisões de *offloading* e das variáveis contínuas. Inicialmente, assume-se inferência local para todos os usuários, enquanto potência

$$\underset{\{\boldsymbol{x},\boldsymbol{\mu},\boldsymbol{p},\boldsymbol{f}\}}{\text{maximize}} \quad \sum_{n=1}^{\mathcal{N}} \text{QoE}_n \tag{1}$$

s.a:

$$\sum_{s=1}^{S} x_n^s \leq 1, \forall n \in \mathcal{N}, \tag{2}$$

$$\mu^{\min} \leq \mu_n \leq 1, \forall n \in \mathcal{N}, \tag{3}$$

$$p_n \leq p^{\max}, \forall n \in \mathcal{N}, \tag{4}$$

$$f_n \leq f^{\max}, \forall n \in \mathcal{N}, \tag{5}$$

$$t_n \leq t^{\max}, \forall n \in \mathcal{N}, \tag{6}$$

$$E_n \leq E_n^{\max}, \forall n \in \mathcal{N}, \tag{7}$$

$$\varepsilon_n \geq \varepsilon^{\min}. \tag{8}$$

$$\begin{aligned} \text{QoE}_n = {} & \omega_t \sigma\big(\lambda(t_n^l - t_n)\big) + \\ & \omega_e \sigma\big(\beta(E_n^l - E_n)\big) + \\ & \omega_a \sigma\big(\eta(\varepsilon_n - \varepsilon_n^l)\big), \end{aligned} \tag{9}$$

**Algorithm 1** Heurística gananciosa para maximização da QoE

**Require:** Número de usuários $\mathcal{N}$, limite de iterações $I_{\max}$
**Ensure:** Melhor valor de QoE e solução correspondente
1: $(x_n^s, p_n, f_n, \mu_n) \leftarrow (0, p^{\min}, f^{\min}, \mu^{\min})\ \forall n \in \mathcal{N}, \forall s \in \mathcal{S}$
2: $best\_qoe \leftarrow \text{QoE}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f})$
3: $best\_sol \leftarrow (\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f})$
***# Otimização da decisão de offloading***
4: **for all** $s \in \mathcal{S}$ **do**
5: **for** $n = 1$ to $\mathcal{N}$ **do**
6: $x_n^s \leftarrow 1$ e remove as demais associações do UE $n$
7: $qoe \leftarrow \text{QoE}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f})$
8: **if** $qoe \neq$ `None` and $qoe > best\_qoe$ **then**
9: Atualiza $best\_qoe$ e $best\_sol$
10: **else**
11: **break**
***# Otimização das variáveis contínuas***
12: **for all** $v \in \{\boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f}\}$ **do**
13: **for** $i = 1$ to $I_{\max}$ **do**
14: **for** $n = 1$ to $\mathcal{N}$ **do**
15: Escolhe aleatóriamente $v_n \sim U(v^{\min}, v^{\max})$
16: $qoe \leftarrow \text{QoE}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{f})$
17: **if** $qoe \neq$ `None` and $qoe > best\_qoe$ **then**
18: Atualiza $best\_qoe$ e $best\_sol$
19: **else**
20: **break**
21: **return** $best\_qoe$, $best\_sol$

de transmissão, frequência computacional e fator de extração são definidos em seus valores mínimos. A QoE resultante é então armazenada como a melhor solução inicial. Na primeira etapa, o algoritmo avalia associações usuário-servidor de forma gananciosa. Uma nova associação é aceita apenas se for viável e resultar em aumento da QoE. Caso contrário, a busca é interrompida. Na segunda etapa, as variáveis contínuas são ajustadas por amostragem aleatória dentro de seus intervalos viáveis durante, no máximo, $I_{\max}$ iterações. A cada atualização, a solução é aceita somente quando melhora a QoE. A função de QoE utilizada nas linhas 7 e 16 é definida pela Equação 9.

## 3.3 Resultados

Nesta seção, avaliamos o desempenho do sistema de *offloading* de tarefas sensível à semântica sob condições dinâmicas da rede sem fio. Comparamos nossa abordagem com duas soluções alternativas: (i) *offloading* de tarefas sem extração semântica (*Não semantica*); e (ii) *Local*, em que todas as tarefas são processadas no próprio dispositivo.

O sistema foi implementado utilizando uma combinação de PyTorch para a codificação e decodificação semântica e scikit-learn para aplicar regressão não linear. O problema MINLP foi modelado utilizando o IBM CPLEX 22.1.0.0. O modelo de canal segue um perfil de desvanecimento de Rayleigh com AWGN. A fim de calcular a demanda computacional do codificador e do decodificador, utilizamos a ferramenta *ptflops* [21]. Consideramos uma RTX 2080 Ti, uma RTX 3090 Ti, uma RTX 4090 e uma RTX 5090 como o UE, a borda distante (*Far Edge* (FE)), a borda próxima (*Near Edge* (NE)) e o servidor em nuvem, respectivamente. O código completo deste trabalho está disponível publicamente em um repositório do GitHub [1].

[1] https://github.com/LABORA-INF-UFG/SATO

A Figura 6((i) e (ii)) apresenta o desempenho da QoE sob diferentes condições de largura de banda e ruído. À medida que a largura de banda aumenta, todos os métodos apresentam melhora de desempenho, porém o *offloading* de tarefas sensível à semântica alcança valores mais elevados de QoE devido ao menor atraso de transmissão e à preservação da acurácia de inferência. Sob o aumento da potência do ruído, seu desempenho se degrada de forma mais lenta do que o *offloading* sem extração semântica, uma vez que as características semânticas comprimidas requerem menos bits transmitidos e são mais robustas às degradações do canal. De modo geral, o *offloading* de tarefas sensível à semântica proporciona desempenho superior tanto em cenários com limitação de largura de banda quanto em cenários com canais ruidosos.

A Figura 6((iii) e (iv)) avalia o desempenho da otimização e o custo computacional do modelo proposto. O resultado em 6(iii) compara a QoE obtida pelas abordagens heurística e MINLP. O *gap de otimalidade* mede a diferença entre a solução atual e a solução ótima, enquanto o *gap heurística–MINLP* representa a diferença entre os resultados da heurística e do MINLP. A heurística proposta alcança uma QoE próxima da ótima com complexidade significativamente menor. O resultado em 6(iv) apresenta o tempo de execução da heurística para diferentes quantidades de usuários, considerando que todas as instâncias foram limitadas a 2 horas de execução.

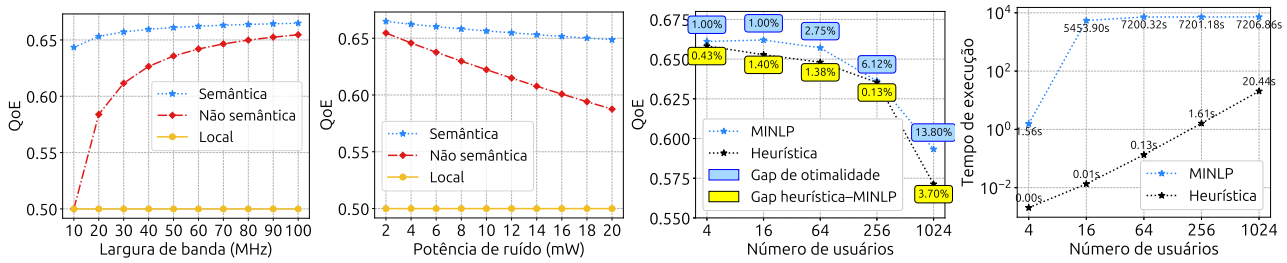


Figura 6: Avaliação da QoE sob variações da largura de banda (i), e potência do ruído (ii). Comparação da qualidade da solução (iii) e do tempo de execução (iv) entre as abordagens MINLP e heurística.

## 3.4 Conclusão

Neste trabalho, propomos um modelo de alocação de recursos e *offloading* de tarefas sensível à semântica que integra conjuntamente a extração de características semânticas, a transmissão sem fio e o escalonamento computacional em um modelo orientado à QoE. Também investigamos o impacto da extração semântica e da qualidade da transmissão sobre a acurácia de inferência para fins de alocação de recursos. Os resultados mostram que o *offloading* de tarefas sensível à semântica supera consistentemente as abordagens sem extração semântica e de execução local, especialmente em condições de largura de banda limitada e canais ruidosos. Esses resultados destacam o potencial da comunicação semântica para serviços de borda inteligentes e escaláveis em redes 6G. Como trabalhos futuros, podem ser investigados cenários de *offloading* em múltiplas camadas e de inferência multimodal.

# 4 Conclusão Geral

Com base nos resultados apresentados nas duas frentes de pesquisa sobre o núcleo da rede 6G, i.e., (i) exposição programável de capacidades de rede por meio do paradigma *Network as a Service* e (ii) alocação inteligente de recursos por meio de comunicação semântica e computação de borda, observa-se que a concepção do núcleo da rede 6G exige uma arquitetura capaz de integrar, de forma nativa, comunicação, computação, inteligência e programabilidade. Os trabalhos desenvolvidos demonstram que o núcleo 6G deve evoluir para além de suas funções tradicionais de conectividade, tornando-se uma plataforma aberta e adaptativa para a orquestração de serviços e recursos distribuídos.

No contexto da exposição de capacidades de rede, a proposta do *CAMARA-to-Network Exposure Function* evidencia a viabilidade de mecanismos padronizados que permitam a aplicações externas acessar funcionalidades avançadas do núcleo de rede por meio de APIs de alto nível, reduzindo a complexidade associada às interfaces nativas do 3GPP e favorecendo a adoção do paradigma *Network as a Service*. Os resultados obtidos demonstram que essa camada de abstração pode ser implementada de forma modular, reutilizável e com impacto desprezível no desempenho do sistema, contribuindo para a construção de ecossistemas mais abertos, interoperáveis e orientados à inovação.

A investigação sobre comunicação semântica aplicada ao *offloading* de tarefas mostra que a integração entre comunicação e computação será um elemento central das futuras redes 6G. A utilização de informações semânticas para orientar decisões de alocação de recursos e de processamento distribuído permitiu alcançar ganhos significativos na qualidade da experiência, especialmente em cenários com restrições de largura de banda e condições adversas do canal. Esses resultados reforçam a necessidade de mecanismos inteligentes de gerenciamento de recursos capazes de considerar simultaneamente os requisitos de comunicação, processamento, consumo de energia e desempenho das aplicações.

Os resultados desta atividade convergem para a visão de um núcleo 6G orientado a serviços, programável, inteligente e fortemente integrado aos ambientes de borda e de nuvem. A combinação de mecanismos de exposição de capacidades de rede com técnicas avançadas de otimização e comunicação semântica representa um caminho promissor para viabilizar aplicações emergentes que demandam elevada flexibilidade, baixa latência, eficiência energética e adaptação dinâmica ao contexto operacional. Assim, os avanços apresentados contribuem para a consolidação dos princípios arquiteturais que deverão nortear o desenvolvimento das futuras redes 6G e estabelecem bases sólidas para investigações futuras envolvendo automação baseada em inteligência artificial, orquestração fim a fim de serviços e integração entre múltiplos domínios de comunicação e computação.

# Referências


[1] J. Ordonez-Lucena and F. Dsouza, "Pathways towards network-as-a-service: The CAMARA project," in *Proc. ACM SIGCOMM Workshop on Network-Application Integration (NAI)*, 2022, pp. 53–59.

[2] GSMA, "CAMARA project: Network APIs for developers," https://www.gsma.com/solutions-and-impact/technologies/networks/camara/, 2023.

[3] 3GPP, "TS 29.522: 5G system; network exposure function northbound APIs; stage 3," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2018, release 18.

[4] J. Fernandes, L. Rosa, J. Gameiro *et al.*, "Pushing the boundaries of scalable 5G core networks: Cloud-native NEF and CAPIF interplay," in *Proc. 3rd International Conference on 6G Networking (6GNet)*, 2024, pp. 201–205.

[5] 3GPP, "TS 29.222: Common API framework for 3GPP northbound APIs; stage 3," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2018, release 18.

[6] ——, "TS 23.502: Procedures for the 5G system (5GS)," 3rd Generation Partnership Project (3GPP), Tech. Rep., 2016, release 18.

[7] Y. Li, F. Zhou, L. Yuan, Q. Wu, and N. Al-Dhahir, "Cognitive semantic communication: A new communication paradigm for 6g," *IEEE Communications Magazine*, vol. 63, no. 6, pp. 122–129, 2025.

[8] C. Zhang, L. Huang, and Q. Ning, "Resource allocation in wireless semantic communications: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.

[9] Y. Zheng, T. Zhang, X. Mu, Y. Liu, and R. Huang, "Joint semantic transmission and resource allocation for intelligent computation task offloading in mec systems," *IEEE Transactions on Wireless Communications*, vol. 24, no. 10, pp. 8756–8770, 2025.

[10] Y. Zheng, T. Zhang, R. Huang, and Y. Wang, "Computing offloading and semantic compression for intelligent computing tasks in mec systems," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[11] Y. Cang, M. Chen, Z. Yang, Y. Hu, Y. Wang, C. Huang, and Z. Zhang, "Online resource allocation for semantic-aware edge computing systems," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 094–28 110, 2024.

[12] Z. Ji and Z. Qin, "Computational offloading in semantic-aware cloud-edge-end collaborative networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 7, pp. 1235–1248, 2024.

[13] Z. Ji, Z. Qin, X. Tao, and Z. Han, "Resource optimization for semantic-aware networks with task offloading," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 12 284–12 296, 2024.

[14] G. Zhang, Q. Hu, Z. Qin, Y. Cai, G. Yu, and X. Tao, "A unified multi-task semantic communication system for multimodal data," *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 4101–4116, 2024.

[15] C. Wang, R. Olsson, S. Forsström, and Q. He, "Deep semantic inference over the air: An efficient task-oriented communication system," 2025. [Online]. Available: https://arxiv.org/abs/2508.12748

[16] H. Zhang, S. Shao, M. Tao, X. Bi, and K. B. Letaief, "Deep learning-enabled semantic communication systems with task-unaware transmitter and dynamic data," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 170–185, 2023.

[17] X. Chen, D. Feng, W. Jiang, Q. Luo, G. Chen, and Y. Sun, "Qoe-driven multi-task offloading for semantic-aware edge computing systems," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 7100–7117, 2026.

[18] L. Yan, Z. Qin, C. Li, R. Zhang, Y. Li, and X. Tao, "Qoe-based semantic-aware resource allocation for multi-task networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 9, pp. 11 958–11 971, 2024.

[19] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," 2020. [Online]. Available: https://arxiv.org/abs/2006.03677

[20] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[21] V. Sovrasov, "Ptflops: A flops counting tool for neural networks in pytorch framework," *GitHub repository*, 2018.