Alocação de Recursos em *Edge* e *Cloud Computing* para Atender Dispositivos de IoT: Uma Análise Rumo ao 6G

Samuel Moreira Abreu Araújo[1], **Mayron César de Oliveira Moreira**[2,3]
Geraldo Robson Mateus[3]

[1]Departamento de Tecnologia em Engenharia Civil, Computação,
Automação, Telemática e Humanidades
Universidade Federal de São João Del-Rei (UFSJ) – Ouro Branco, MG - Brasil

[2]Departamento de Ciência da Computação
Universidade Federal de Lavras (UFLA) – Lavras, MG - Brasil

[3]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG - Brasil

sabreu@ufsj.edu.br,mayron.moreira@ufla.br,mateus@dcc.ufmg.br

Resumo. *Dispositivos de Internet das Coisas (IoT) demandam atualmente processamento rápido, sendo as tecnologias de Edge Computing (EC) e Cloud Computing (CC) frequentemente utilizadas para este fim. Este artigo apresenta uma definição e um modelo matemático para o problema de integração de sensores de IoT, EC e CC em ambientes de Cidades Inteligentes. O modelo prevê que, caso seja oportuno, as demandas dos sensores podem ser processadas em dispositivos de CC alugados sob demanda. Adicionalmente, é apresentado um estudo sobre as gerações de tecnologias de comunicação, até o 6G. Em experimentos computacionais, considerando essas tecnologias, ao adotar a tecnologia 6G, o atraso fim a fim no atendimento da demanda de um sensor é de* $\approx 9ms$*, significativamente menor em comparação a tecnologia 4G (*$\approx 410ms$*). Além disso, a função objetivo que minimiza os custos conseguiu reduzi-los em até* $\approx 123.81\%$ *em comparação à função que minimiza os atrasos fim a fim.*

Abstract. *Internet of Things (IoT) devices currently require fast processing, and Edge Computing (EC) and Cloud Computing (CC) technologies are employed for this purpose. This article introduces a definition and mathematical model for integrating IoT, EC, and CC sensors in Smart City environments. The model permits that, if necessary, sensor demands can be processed in CC servers rented on demand. Additionally, a study is presented on the generations of communication technologies, up to 6G. In computational experiments, considering these technologies, when adopting 6G technology, the end-to-end delay in addressing the demand of a sensor is* $\approx 9ms$*, significantly lower compared to 4G technology (* $\approx 410ms$*). Furthermore, the cost-minimizing objective function managed to reduce them by up to approximately* 123.81% *compared to the function that minimizes end-to-end delays.*

1. Introdução

A era das tecnologias inteligentes é uma realidade e pode ser considerada um dos principais desenvolvimentos tecnológicos de nossos tempos. Nesse cenário, a IoT

refere-se à conexão de diversos objetos à internet, ampliando suas possibilidades de aplicação e destacando-se como uma tecnologia promissora que impulsiona a inovação digital. Embora seja familiar há anos, sua visibilidade tem aumentado significativamente, em parte devido aos avanços nas redes, que buscam atender à mobilidade dos usuários e objetos [Alsabah et al. 2021]. Como exemplo, o monitoramento da poluição em cidades e as indústrias são setores que atualmente utilizam sensores de IoT [Queiroz et al. 2022].

Com o desenvolvimento da IoT, as cidades e corporações passaram a empregar tal tecnologia, abrangendo setores como o de automação do tráfego e o de gerenciamento de mobilidade, amadurecendo assim o conceito de Cidades Inteligentes. A IoT tem como premissa permitir o monitoramento, controle e automação de dispositivos, visando aprimorar a qualidade de vida das pessoas. Ela tem o potencial de transformar diversos aspectos da vida urbana [Santos et al. 2021]. Essas características colocam a IoT com papel fundamental na implantação de ambientes inteligentes e integrados, presentes nas Cidades Inteligentes [Khan et al. 2020]. Exemplificando, uma lixeira pode possuir um sensor e enviar um sinal para o serviço de limpeza esvaziá-la, ou câmeras podem ser utilizadas para medir a quantidade de pessoas em ambientes [Queiroz et al. 2022].

A implementação da IoT requer a conexão dos dispositivos utilizados em diferentes áreas de uma cidade por meio de uma rede confiável [Queiroz et al. 2022]. Diante dessa necessidade, e considerando aplicações sensíveis a atrasos, como realidade virtual e carros autônomos, é fundamental uma infraestrutura de rede capaz de atender às demandas desses serviços, com altas taxas de transferência e baixos atrasos fim a fim. Como dificultador, prevê-se que entre 2020 e 2030 o número de dispositivos de IoT no mundo dobre, atingindo mais de 29 bilhões, com a expectativa de que essa taxa de crescimento se mantenha consistente até 2040 [Vailshery 2023]. Esse aumento, combinado com exigências mais rigorosas em relação a taxas de transmissão e atrasos, impulsiona a necessidade de aprimoramento das tecnologias de comunicação.

Em resposta aos citados desafios, existe um amplo conjunto de pesquisas dedicadas ao desenvolvimento de tecnologias redes avançadas, incluindo as da sexta geração, conhecidas como 6G [Alwis et al. 2021]. Observa-se também um crescimento relevante no tráfego mundial de dados [Alsabah et al. 2021]. Projeções indicam que esse tráfego atingirá 5016 Exabytes até 2030 [ITU-R 2015]. O referido aumento apresenta alguns desafios, entre eles: prover ubiquidade, aprimorar o processamento das demandas dos dispositivos de IoT, e melhorar as infraestruturas de redes [Khan et al. 2020].

1.1. *Background*

As redes de quinta geração (5G) foram desenvolvidas para proporcionar altas taxas de transferência de dados, superando as redes de quarta geração (4G). Contudo, existe um notável aumento no número de dispositivos IoT, demandando comunicação confiável e baixos atrasos. Esses aspectos colocam diante das redes 5G desafios a serem superados, impulsionando o desenvolvimentismo de novas tecnologias, como as de sexta geração (6G) [Alsabah et al. 2021, Alwis et al. 2021]. A Tabela 1 apresenta um resumo da evolução das tecnologias de redes do 1G ao 6G. Dentre outras diferenças, percebe-se que o raio de cobertura do sinal diminui à medida que as tecnologias avançam, entre outros fatores, devido aos diferentes espectros de frequência utilizados. Isto implica por exemplo, na necessidade de um maior número maior número de estações para para cobrir uma área com a tecnologia 5G em comparação com a tecnologia 4G.

Tabela 1. Evolução das redes comunicações sem fio, do 1G ao 6G

Rede	Década	Destaques e aplicações	Largura de banda teórica	Largura de banda média[1]	Raio de cobertura[1]
1G	1980	Redes de comunicação analógica para chamadas de voz	2.4 Kbps	2 Kbps	até 20 km
2G	1990	Redes de comunicação digital, suporte para mensagens de texto	64 Kbps	40 Kbps	5 - 10 km
3G	2000	Banda larga móvel, mensagens multimídia, videochamadas, etc	2 Mbps	1 Mbps	2 - 5 Km
4G	2010	Banda larga móvel aprimorada, *streaming* de vídeo, jogos e IoT	200 Mbps	10 Mbps	1 - 3 Km
5G	2020	IoT, serviços de alta largura de banda para conectividade sem fio, comunicação ultra confiável de baixa latência, redes definidas por *software*, aplicações de Realidade Virtual, carros autônomos, etc	10 Gbps	120 Mbps	até 600 metros
6G	2030	Rumo à Internet de Tudo (mundo totalmente digital e conectado) era da inteligência artificial, aplicações de realidade estendida, telepresença holográfica, direção colaborativa, indústria 5.0, IoT hiperinteligente, robôs colaborativos, etc	até 1Tbps	1Gbps	até 320 metros

[1]A largura de banda média e a cobertura são influenciadas por diversos fatores, incluindo frequência e interferências. Os valores fornecidos são aproximações com base nos trabalhos [Alsabah et al. 2021, Alwis et al. 2021, Four-Faith 2022, Shah et al. 2021]

Para superar alguns dos desafios levantados, uma tendência é o processamento das demandas dos dispositivos de IoT em uma infraestrutura de computação em nuvem (CC). Contudo, bilhões de dispositivos serão integrados ao ecossistema de IoT nos próximos anos. Todos esses dispositivos estarão conectados à rede, enviando e recebendo dados para a CC, o que tornará as atuais soluções de CC centralizadas impraticáveis [Santos et al. 2021]. Assim, propõe-se o uso de servidores nas extremidades da rede (EC). A EC refere-se à inserção de uma camada de servidores intermediária entre os sensores de IoT e a CC [Khan et al. 2020]. A Figura 1 mostra uma estrutura hierárquica de rede com sensores de IoT e CC, e a introdução da camada de EC. Além disso, a CC oferece recursos tecnológicos, como processamento e armazenamento, de maneira flexível e sob demanda, por meio da internet, com uma precificação ajustada conforme a utilização.

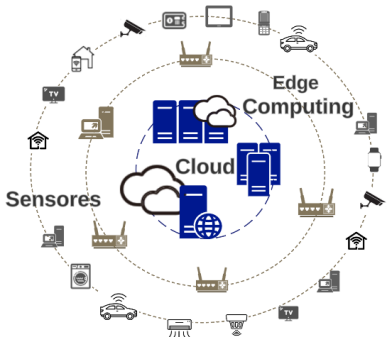


Figura 1. Exemplo de hierarquia: sensores, EC e CC

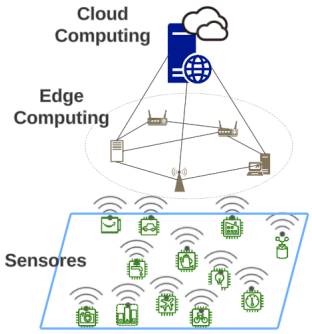


Figura 2. Exemplo de conexões: sensores, EC e CC

No contexto de integração entre IoT, Cidades Inteligentes, EC e CC, à medida que avanços ocorrem, surgem novos desafios, tais como a otimização da alocação de recursos e estabelecimento de conexões entre os dispositivos [Santos et al. 2021]. Neste trabalho, conforme mostrado na Figura 1 e em consonância com [Queiroz et al. 2022], é explorado um cenário contextualizado em tal integração. Contrastando com os aspectos levantados, é necessário que a definição do problema aborde conceitos de alocação de recursos de processamento e também de roteamento. Assim, em complemento aos estudos [Premsankar et al. 2018, Queiroz et al. 2022], é proposta uma definição do problema e uma modelagem matemática abrangendo tanto a alocação de recursos quanto

o roteamento entre sensores e servidores da rede. Adicionalmente a [Santos et al. 2021], e complementarmente ao estado da arte, o modelo proposto é avaliado considerando questões evolutivas das tecnologias de rede até o 6G.

A problemática levantada anteriormente induz a algumas questões sobre: o impacto das tecnologias de rede móvel no atendimento das demandas dos sensores de IoT; a necessidade de envolvimento de vários componentes de CC para processar eficientemente os dados; e o impacto da evolução do 5G para o 6G em métricas de QoS. Para abordar essas questões, como contribuição são apresentadas: *i*) uma definição formal do problema, incluindo alocação de recursos e roteamento; *ii*) um modelo matemático utilizando Programação Linear Inteira (*Integer Linear Programming*, ILP); e *iii*) uma análise de experimentos, empregando um algoritmo exato em instâncias reais da literatura, considerando variações entre as tecnologias de redes. A opção por um algoritmo exato é justificada pela capacidade de proporcionar soluções com garantias de otimalidade. Dessa forma, as análises dos experimentos são conduzidas de maneira assertiva, eliminando potenciais erros de decisão e vieses que poderiam surgir ao empregar heurísticas.

O restante do artigo está organizado como segue. É apresentada na Seção 2 uma revisão dos trabalhos da literatura. O problema é definido na Seção 3, e o modelo apresentado na Seção 4. Os cenários e experimentos computacionais são mostrados na Seção 5. Por fim, as conclusões e trabalhos futuros são discutidos na Seção 6.

2. Trabalhos Relacionados

Um levantamento da evolução das pesquisas sobre Cidades Inteligentes integradas com EC e CC é apresentado em [Khan et al. 2020]. Os autores destacam os avanços na EC, abordando aspectos de otimização e métricas. Para os autores, a CC é um paradigma fundamental para a implementação das Cidades Inteligentes. No entanto, o atraso inerente à CC motiva a transferência dos recursos de computação para dispositivos de EC, aspectos estes explorados neste artigo. A implementação da EC em Cidades Inteligentes apresenta desafios, sendo um deles explorado neste artigo e não tratado em [Khan et al. 2020, Santos et al. 2021, Queiroz et al. 2022], que consiste na análise dos algoritmos em conjunto com as tecnologias de redes recentes, como o 5G e 6G.

Uma revisão sobre as tecnologias de comunicação é encontrada nos artigos [Alwis et al. 2021, Alsabah et al. 2021]. Alwis *et al.* destacam conceitos emergentes em redes e expõem limitações do 5G, como atrasos e transferência de dados. Segundo os autores, tais limitações dificultam o desenvolvimento de aplicações inovadoras, como robôs colaborativos e aplicações da Internet de Tudo. As limitações identificadas por Alwis *et al.* serviram como motivação para as investigações realizadas neste artigo, com foco nos atrasos fim a fim gerados pelo atendimento dos sensores de IoT.

O processamento das demandas de dispositivos IoT em EC e CC, e abrangendo a Computação Contínua, é discutido em [Rosendo et al. 2020]. Segundo os autores, a Computação Contínua refere-se a uma infraestrutura utilizada para processar fluxos de trabalho complexos de dispositivos IoT, abordando computação em tempo real. Rosendo *et al.* assumem que a maioria das aplicações IoT possui uma velocidade de transmissão média de 22 Mbps e um atraso fim a fim de $100ms$, mas estes valores podem ser inviáveis em um contexto atual. Adicionalmente ao proposto por Rosendo *et al.*, este artigo discute os impactos desses parâmetros em um contexto das tecnologias de redes sem fio atuais.

O posicionamento de dispositivos de EC para atender às demandas de aplicativos veiculares em Cidades Inteligentes é abordado em [Premsankar et al. 2018]. Os autores destacam desafios, como garantir conectividade, especialmente em áreas com problemas de cobertura, exigindo um planejamento preciso de rede. A análise da cobertura é um aspecto explorado neste artigo, no qual esse parâmetro é avaliado, abrangendo variações até o 6G. Para tratar o problema, os autores propõem um modelo de Programação Inteira Mista (*Mixed Integer Programming*, MIP) com o objetivo de minimizar os custos de instalação da rede. Contudo, diferentemente de Premsankar *et al.* e em consonância com [Rosendo et al. 2020, Queiroz et al. 2022], considerando os custos de instalação da rede e assumindo que as demandas dos dispositivos de IoT podem ser originadas de diferentes lugares, o problema tratado neste artigo busca não posicionar novos servidores, mas sim tomar decisões sobre a utilização eficiente dos dispositivos de EC e CC já existentes.

O trabalho de [Queiroz et al. 2022] apresenta uma definição do problema que aborda a integração de sensores de IoT e EC em uma Cidade Inteligente. Em tal contexto, servidores não utilizados podem ser desligados para minimizar os custos de operação, princípios esses adotados neste artigo. Os autores apresentam um modelo em ILP e uma heurística para tratar o problema; no entanto, eles não consideram aspectos relacionados ao roteamento das conexões, não aplicam conceitos de consumo de memória e não avaliam diferentes tecnologias de comunicação. Ressalta-se que esses pontos podem impactar diretamente nos atrasos para atender às demandas dos sensores. Assim como Queiroz *et al.*, este artigo propõe um modelo em ILP, sendo os experimentos realizados por simulações, mas incluindo os aspectos citados.

Segundo [Long et al. 2018], ao abordar os trabalhos de integração EC e CC, grande parte das pesquisas assumem que um servidor de CC pode ser considerado um servidor especial, utilizado para auxiliar no processamento das demandas. Os autores propõem uma abordagem cooperativa EC-CC, na qual servidores de CC podem ser alugados para auxiliar no processamento das demandas. O problema tratado por Long *et al.* é modelado com MIP e tem como objetivo minimizar o consumo de energia. Similarmente a Long *et al.*, este artigo propõe que exista tal cooperação entre EC-CC, onde servidores de CC podem ser alugados sob demanda. No entanto, Long *et al.* não abordam restrições de roteamento, não abordam a possibilidade de contratar diferentes configurações de CC e não investigam variações nos parâmetros de transmissão.

3. Definição Formal do Problema

A definição formal do problema descreve a interação entre sensores de IoT, servidores de EC e de CC em uma Cidade Inteligente. Tal definição pode ser concebida como uma derivação da proposta em [Queiroz et al. 2022]. Nesse contexto, os sensores de IoT produzem dados que precisam ser enviados e processados em algum ponto da rede, seja em um servidor de EC ou CC. Sejam S, E, C respectivamente os conjuntos de sensores, servidores (nós) de EC e servidores (nós) CC existentes. Seja ainda $\mathcal{A}_i$ o conjunto de nós cobertos (alcançáveis) por um nó i, tal que $\mathcal{A}_i \subseteq E$, se $i \in S$; e $\mathcal{A}_i \subseteq C$, se $i \in E \cup C$. No caso de um nó $s \in S$, para ser coberto por um nó $e \in E$, ele deve estar dentro do raio de cobertura existente, determinado pela tecnologia adotada (Tabela 1).

Cada sensor $s \in S$ possui diferentes demandas que precisam ser atendidas. Essas demandas requerem capacidades de processamento e memória, representadas

respectivamente por $p_s \in \mathbb{Z}_+$ (*cores*) e $m_s \in \mathbb{R}_+$ (MB). A rede física possui nós de EC e nós de CC. Cada nó $e \in E$ e $c \in C$ possui, respectivamente, capacidades máximas de processamento P_e e P_c (*cores*), bem como de memória M_e e M_c (MB). Tanto os sensores $s \in S$, quanto os servidores $c \in C$ e $e \in E$, possuem posicionamento geográfico definido por suas respectivas latitude e longitude. Cada servidor $c \in C$ e $e \in E$ possui ainda um atraso de processamento da demanda solicitada, ou, caso a demanda não seja atendida no nó em questão, um atraso para examinar os dados e repassar para um próximo nó na rede.

Seja $bw_s \in \mathbb{R}_+$ a largura de banda demandada pelo sensor $s \in S$ (Mb/s). Assume-se que bw_s é definido como a média percebida pela tecnologia de rede adotada (Tabela 1). Sejam $BW_{se} \in \mathbb{R}_+$, $BW_{ec} \in \mathbb{R}_+$ e $BW_{cc'} \in \mathbb{R}_+$ parâmetros, em Mb/s, que representam, respectivamente, a largura de banda disponível para a conexão entre um sensor $s \in S$ e um nó $e \in E$, um nó $e \in E$ e um nó $c \in C$, e dois nós $c, c' \in C$. Cada sensor pode se conectar a um nó de EC que esteja dentro do raio de cobertura, definido como $\mathcal{A}_s$. Cada servidor $e \in E$ pode se conectar a um servidor $c \in C$, desde que exista uma conexão, definida pelo conjunto $\mathcal{A}_e$. Cada servidor $c \in C$ pode se conectar a um servidor $c' \in C$, desde que exista uma conexão, definido pelo conjunto $\mathcal{A}_c$. Em todos os casos, a conexão só pode acontecer se existir largura de banda residual suficiente. A cada conexão realizada existe uma latência relacionada à transmissão dos dados.

4. Modelo em Programação Linear Inteira

As variáveis, restrições e funções objetivo utilizadas são apresentadas na sequência desta Seção. Os parâmetros e terminologias utilizados são os mesmos apresentados na Seção 3.

4.1. Variáveis de decisão

- $w_s \in \{0, 1\}$, igual a 1 se as demandas de s foram atendidas com sucesso;
- $z_e \in \{0, 1\}$, igual a 1 se o nó e está ativo para processar as demandas de algum sensor s, ou se é usado como nó de roteamento para algum nó de CC;
- $z_c \in \{0, 1\}$, igual a 1 se o nó e está ativo para processar as demandas de algum sensor s, ou se é usado como nó de roteamento para algum outro no CC;
- $y_{se} \in \{0, 1\}$, igual a 1 se o nó e atende a demanda do nó sensor s;
- $y_{sc} \in \{0, 1\}$, igual a 1 se o nó c atende a demanda do nó sensor s;
- $x_{se} \in \{0, 1\}$, igual a 1 se existe fluxo do nó sensor s para o nó e;
- $x^s_{ec} \in \{0, 1\}$, igual a 1 se existe fluxo do sensor s, entre o nó e e o nó c;
- $x^s_{cc'} \in \{0, 1\}$, igual a 1 se existe fluxo do sensor s, entre os nós c e c'.

4.2. Restrições

$$\sum_{e \in \mathcal{A}_s} y_{se} + \sum_{c \in C} y_{sc} = w_s, \quad \forall s \in S \tag{1}$$

$$\sum_{s \in S | e \in \mathcal{A}_s} y_{se} p_s \leq P_e z_e, \quad \forall e \in E \tag{2}$$

$$\sum_{s \in S | e \in \mathcal{A}_s} y_{se} m_s \leq M_e z_e, \quad \forall e \in E \tag{3}$$

$$\sum_{s \in S} y_{sc} p_s \leq P_c z_c, \quad \forall c \in C \tag{4}$$

$$\sum_{s \in S} y_{sc} m_s \leq M_c z_c, \quad \forall c \in C \tag{5}$$

$$\sum_{s \in S | e \in \mathcal{A}_s} x^s_{ec} bw_s \leq BW_{ec}, \quad \forall e \in E, \forall c \in \mathcal{A}_e \tag{6}$$

$$\sum_{s \in S} x^s_{cc'} bw_s \leq BW_{cc'}, \quad \forall c \in C, \forall c' \in \mathcal{A}_c \tag{7}$$

$$\sum_{e \in \mathcal{A}_s} x_{se} = w_s, \quad \forall s \in S \tag{8}$$

$$x_{se} - \sum_{c \in \mathcal{A}_e} x^s_{ec} = y_{se}, \quad \forall s \in S, \forall e \in \mathcal{A}_s \tag{9}$$

$$\sum_{e \in E | c \in \mathcal{A}_e} x^s_{ec} + \sum_{c' \in C | c \in \mathcal{A}_{c'}, c' \neq c} x^s_{c'c} - \sum_{c' \in C | c' \in \mathcal{A}_c, c' \neq c} x^s_{cc'} = y_{sc}, \quad \forall c \in C, \forall s \in S \tag{10}$$

$$y_{se} \leq z_e, \quad \forall s \in S, \forall e \in \mathcal{A}_s \tag{11}$$

$$x^s_{ec} \leq z_e, \quad \forall s \in S, \forall e \in \mathcal{A}_s, \forall c \in \mathcal{A}_e \tag{12}$$

$$y_{sc} \leq z_c, \quad \forall s \in S, \forall c \in C \tag{13}$$

$$x^s_{cc'} \leq z_c, \quad \forall s \in S, \forall c \in C, \forall c' \in C \tag{14}$$

As Restrições (1) asseguram que as demandas de cada sensor $s \in S$ devem ser atendidas por um nó $e \in E$ ou $c \in C$. Por sua vez, as Restrições (2) a (5) garantem, respectivamente, que as capacidades de memória e processamento dos nós $e \in E$ e $c \in C$ devem ser respeitadas. As Restrições (6) e (7) são estabelecidas para garantir o atendimento à largura de banda requerida por cada sensor. No cenário apresentado, não é considerada a comunicação entre $s \in S$ e $e \in E$. Assume-se que a capacidade bw_s demandada já é estipulada pelo próprio sensor que solicita o serviço. As Restrições (8) a (10) garantem a conectividade entre sensores, nós de EC e CC, aplicando princípios de conservação de fluxo. Cada sensor $s \in S$ está conectado a um nó $e \in E$, que pode encaminhar a demanda para um nó de $c \in C$. Um nó $c \in C$ pode, por sua vez, redirecionar o fluxo para outro nó $c' \in C$. As Restrições (8) asseguram a conservação de fluxo de sensores para nós de EC, enquanto as Restrições (9) garantem a conservação de fluxo de nós EC para nós de CC, e as Restrições (10) aplicam a conservação de fluxo entre nós de CC. As Restrições (11) e (12) garantem que cada nó $e \in E$ seja determinado como ativo, seja para processamento ou para encaminhar o fluxo para outro nó. Por fim, as Restrições (13) e (14) desempenham a mesma função, mas em relação aos nós $c \in C$.

4.3. Função Objetivo

Duas funções objetivo são formuladas, uma considerando os custos relacionados aos servidores ativos, e outra o atraso fim a fim. A função expressa pela Equação (15) fornece um indicador de quanto está sendo gasto para o processamento das demandas dos sensores. Similarmente a [Queiroz et al. 2022], tal função minimiza os custos associados aos servidores ativos. Ressalta-se que a infraestrutura de rede possui uma diversidade de custos de operação, tais como o consumo e custo de energia, manutenção e licenças de *software*. Neste artigo, a minimização dos custos é feita abstraindo esses diversos valores envolvidos e considerando a capacidade de colocar servidores não utilizados em *stand-by*. O parâmetro α_i representa o custo associado ao uso de um servidor da rede $i \in E \cup C$ em

unidades monetárias (\$). As equações (15) e (16) incorporam uma constante *M*, com um valor alto, que atua como uma penalidade para cada sensor $s \in S$ de IoT não atendido.

$$\text{Minimizar: } \sum_{e \in E} \alpha_e z_e + \sum_{c \in C} \alpha_c z_c + M \cdot \sum_{s \in S} (1 - w_s) \tag{15}$$

Cada demanda originada de um sensor $s \in S$ gera um atraso fim a fim, resultante da alocação de recursos em servidores, processamento, e das conexões utilizadas na transmissão de dados (roteamento). A minimização desse atraso é fundamental e pode influenciar na Qualidade de Experiência (*Quality of Experience*, QoE) do serviço. Nesse contexto, uma função é formulada para minimizar tal atraso, mostrada na Equação (16).

$$\begin{aligned} \text{Minimizar: } \sum_{s \in S} \Bigg(& \sum_{e \in \mathcal{A}_s} atraso(s,e) x_{se} + \sum_{s \in S} \sum_{e \in \mathcal{A}_s} \sum_{c \in \mathcal{A}_e} atraso(e,c) x^s_{ec} + \\ & \sum_{c \in C} \sum_{c' \in \mathcal{A}_c} atraso(c,c') x^s_{cc'} + \sum_{e \in E} d_{proc}(e) y_{se} + \sum_{c \in C} d_{proc}(c) y_{sc} + M \cdot (1 - w_s) \Bigg) \end{aligned} \tag{16}$$

Na Equação (16), a função $d_{proc}(i)$ representa o tempo necessário para processar os dados na rede. Se o nó i for utilizado apenas como um nó para repassar os dados, o atraso de processamento é da ordem de microssegundos [Kurose and Ross 2021]. Neste trabalho adota-se $0.002ms$. Caso o nó i esteja efetivamente processando as demandas do sensor $s \in S$, o tempo considerado é de $5ms$, correspondente ao processamento de um serviço de IoT, que envolve funções de rede como: *Network Address Translation* (NAT), *Firewall* (FW), e *Intrusion Detection System* (IDS) [Askari et al. 2019, Araujo et al. 2022]. Ainda segundo Kurose e Ross, a função $atraso(i,j)$ pode ser decomposta em $d_{trans}(i,j) + d_{prop}(i,j)$, sendo:

• $d_{trans}(i,j)$, tempo gasto para enviar os dados entre os nós i, j da rede. Dado por: $(\text{tamanho do arquivo (MB)}/\text{velocidade de transmissão entre } i \text{ e } j)/\text{dividido por 8 (bits)}$. Neste caso, a velocidade de transmissão é definida pela tecnologia utilizada;

• $d_{prop}(i,j)$, tempo necessário para propagar os bits dos dados entre os pontos i e j da rede. Calculado pela razão da distância[1] entre os pontos i e j, e a velocidade de propagação no meio ($\approx 2 \cdot 10^8 metro/segundo$).

5. Experimentos Computacionais

Os experimentos foram realizados em um computador Intel Core i3-8300, 16GB de RAM, e o SO Ubuntu 20.04.3. O simulador utilizado nos experimentos foi implementado em C++, e os modelos foram executados por meio da API IBM ILOG CPLEX V12.6.3[2].

[1]Tal distância é calculada utilizando a fórmula de Haversine, a qual considera a curvatura da Terra para calcular a distância entre dois pontos, a partir de suas latitudes e longitudes.

[2]Código e instâncias disponíveis em `http://tiny.cc/CodeInstance`

5.1. Métricas de Avaliação em Redes

As métricas utilizadas são retiradas de [Araujo et al. 2022, Queiroz et al. 2022], sendo:

- Taxa de Aceitação: razão entre o número de solicitações atendidas e demandadas;
- Tempo de processamento: medido através da biblioteca *chrono.h*, dado em ms;
- Custo de operação: Equação (15), sem considerar a penalidade de não atendimento;
- Atraso: Equação (16), sem considerar a penalidade de não atendimento.

5.2. Cenário de Simulação e Algoritmos Avaliados

As instâncias utilizadas são as mesmas adotadas em [Queiroz et al. 2022], sendo derivadas do projeto de uma Cidade Inteligente em Modena, na Itália. Similarmente a Queiroz *et al.*, neste trabalho é adotada uma aplicação que monitora o tráfego por meio de sensores distribuídos pela cidade, capturando imagens para análise. Assim como em Queiroz *et al*, os servidores de EC e CC estão situados em edifícios municipais, sendo o de CC municipal baseado em *hardware* proprietário. Adicionalmente ao trabalho de Queiroz *et al.*, propõe-se experimentalmente que o servidor de CC municipal possa enviar dados, se necessário, para o processamento em servidores de CC da Amazon EC2, em Milão[3].

Do mesmo modo que [Queiroz et al. 2022], os sensores possuem conectividade sem fio, mas, diferentemente, neste trabalho, o alcance de um sensor é determinado pela sua tecnologia específica. Os atrasos das conexões da rede são definidos pela função $atraso(i, j)$ e, similarmente a [Jia et al. 2018], perturbados pela multiplicação de um número aleatório entre $[0.8, 1.2]$. Nos experimentos, são considerados 100 servidores de EC, 1 servidor de CC do município, e 3 servidores de CC da EC2 contratados sob demanda. O número de sensores é variado em $|S| \in \{100, 150, 200, 250, 300\}$.

Cada sensor de IoT possui uma posição geográfica e demanda $p_s = 8$ *cores* e $m_s = 1600$ MB para atender as funções de rede NAT, FW e IDS [Askari et al. 2019, Bari et al. 2019, Araujo et al. 2022]. Cada servidor de EC é atribuído, seguindo uma distribuição uniforme, a uma máquina com as configurações descritas em [CISCO 2023] (Tabela 2). O custo α_e de cada servidor é definido com base no consumo e no custo da energia elétrica na Itália (relação *kWh*). O servidor de CC da infraestrutura municipal é equipado com configurações do Cisco UCS C240 M7, com $P_c = 96$, $M_c = 8$ TB e $\alpha_c = 0.504$ USD$. Os servidores de CC contratados sob demanda possuem configurações da Amazon Elastic Compute Cloud (EC2, Tabela 3). A demanda bw_s de cada sensor é determinada pela taxa média de conexão, de acordo com a tecnologia. Nas conexões entre servidores de EC e CC, adota-se $BW_{ec} = 10$ Gbps, e entre servidores de CC, considera-se $BW_{cc'} = 1$ Tbps. Os parâmetros adotados são para fins de experimentação e podem ser ajustados caso o modelo seja adotado em um ambiente real.

Tabela 2. Configuração dos servidores de EC

Modelo	p_e (*cores*)	m_e (GB)	Custo/hora (α_e) (USD $)
HX-CPU-I8380	40	256	0.1134
HX-CPU-I6314U4	32	256	0.0861
HX-CPU-I4314	24	256	0.0567
HX-CPU-I4309Y	8	256	0.0441

Tabela 3. Configuração dos servidores da Amazon EC2

Modelo	P_c (*cores*)	M_c (GB)	Custo/hora (α_c)[4] (USD $)
m6a.48xlarge	192	768	9.6768
m5.24xlarge	96	384	5.376
m6g.12xlarge	48	192	2.1504

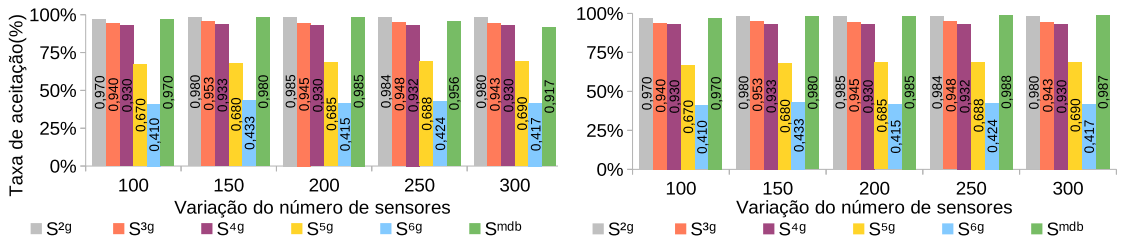
[3]Serviço que fornece recursos computacionais sob demanda, disponível em `http:tiny.cc/AMZEC2`

Os algoritmos utilizados na experimentação são desenvolvidos com base no modelo em ILP (Seção 4) e realizam a alocação com base nos recursos existentes na rede. O *solver* utilizado resolve cada modelo em sua otimalidade. Os algoritmos avaliados são: ILP^c, que utiliza a função objetivo de minimizar custos, expressa pela Equação (15); e ILP^a, que utiliza a função objetivo de minimizar atrasos, expressa pela Equação (16).

Para avaliar os algoritmos, são definidos alguns cenários. O cenário S^{2G} assume que todos os sensores e nós de EC podem se comunicar com os parâmetros da tecnologia 2G (Tabela 1). Analogamente, os cenários S^{3G} ao S^{6G} são definidos. No cenário S^{mbd}, assume-se que tanto os sensores quanto os nós de EC podem operar com qualquer tecnologia. Neste caso, a comunicação adapta-se à tecnologia disponível na cobertura do sensor, priorizando aquela com maior largura de banda.

5.3. Análise de Desempenho

Em cada gráfico apresentado, é possível observar a métrica em questão no eixo y e uma variação no número de sensores no eixo x. Em termos de taxa de aceitação (Figura 3), em situações com poucos sensores processados, tal taxa é próxima, independentemente do número de servidores de CC. Neste caso, existe sobra de recursos, sendo a maioria das rejeições causada por inviabilidades de cobertura. Devido à similaridade nos resultados, optou-se por mostrar apenas os gráficos referentes ao algoritmo ILP^c.



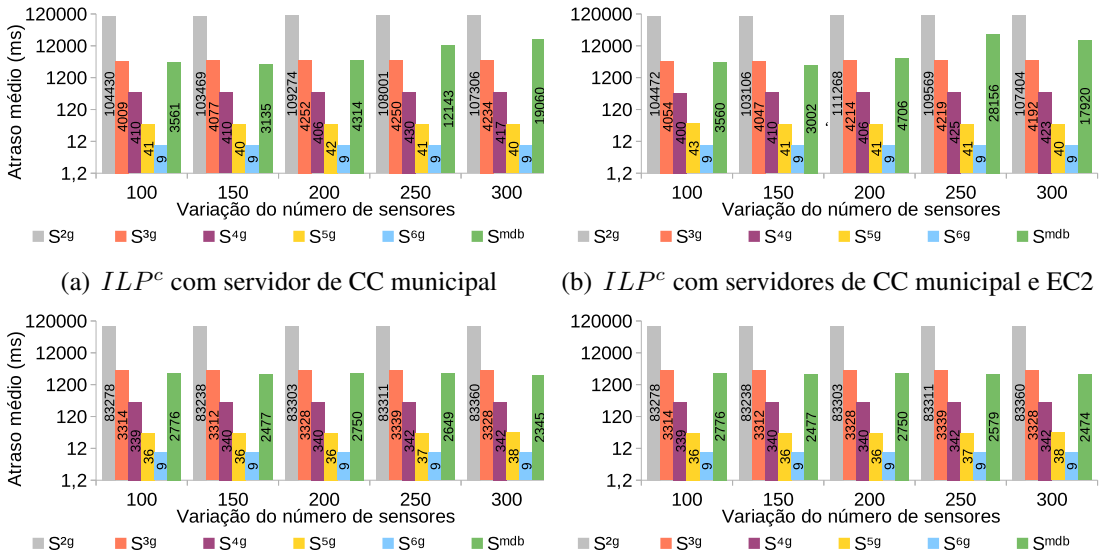
(a) ILP^c com servidor de CC municipal (b) ILP^c com servidores de CC municipal e EC2

Figura 3. Taxa de aceitação, com o algoritmo ILP^c e variando cenários

Nota-se nas Figuras 3 que, no cenário S^{2G}, que a ampla cobertura dos dispositivos resulta em mais opções de atendimento, e, consequentemente, em uma taxa de aceitação alta. Entretanto, há um aumento significativo no atraso fim a fim em comparação com outros cenários (Figura 4), devido às características da tecnologia. No caso, a tecnologia $2G$ oferece uma cobertura com raio maior, mas com uma taxa de transmissão baixa. Por exemplo, comparando os cenários S^{2G} e S^{6G}, ambos com $|S|= 150$ (Figura 3(b)), a taxa de aceitação diminui de 98% para 43%, e o atraso médio de $103106ms$ para $9ms$ (Figura 4(b)). Devido à parametrização adotada, não houve rejeições devido à falta de largura de banda. No entanto, em cenários com mais sensores sendo atendidos simultaneamente, é importante considerar que esse comportamento pode não se repetir. Comparando todos os cenários, destaca-se que a diminuição no atraso médio é significativa, indicando que as tecnologias 6G são recomendadas para evitar atrasos fim a fim na comunicação.

Na Figura 3(b), a queda na taxa de aceitação no cenário S^{6G} ocorre independente da quantidade de sensores ou do algoritmo, devido ao baixo raio de cobertura. Tal comportamento indica a necessidade de replanejamento da topologia de rede para atender dispositivos com a tecnologia $6G$. Adicionalmente, mesmo no cenário S^{2G}, tal taxa não atinge 100%, indicando que alguns sensores estão fora da cobertura dos nós de EC.

Observando a Figura 4 (cenário S^{mdb} à parte), nota-se que o atraso médio tem uma baixa variação, independentemente do número de sensores. Isso ocorre devido à propagação gerada pela função $d_{prop}(i,j)$ não ser significativa em termos de atrasos, devido à alta velocidade de propagação no meio. Além disso, o número de saltos realizados para o processamento na topologia não é elevado, pois as demandas são processadas nos servidores da EC, do município ou da EC2. A maior variação percebida ocorre no cenário S^{mdb}, principalmente ao comparar as funções objetivo. Os resultados mostrados no cenário S^{mdb} são compostos por demandas de sensores atendidas com vários tipos de tecnologias. Isso gera uma variação na largura de banda considerada para a transmissão, afetando diretamente os valores gerados pela função $d_{trans}(i,j)$. Neste caso, as rejeições por falta de cobertura, presentes nos cenários S^{4G} e S^{5G}, são menores.



(a) ILP^c com servidor de CC municipal

(b) ILP^c com servidores de CC municipal e EC2

(c) ILP^a com servidor de CC municipal

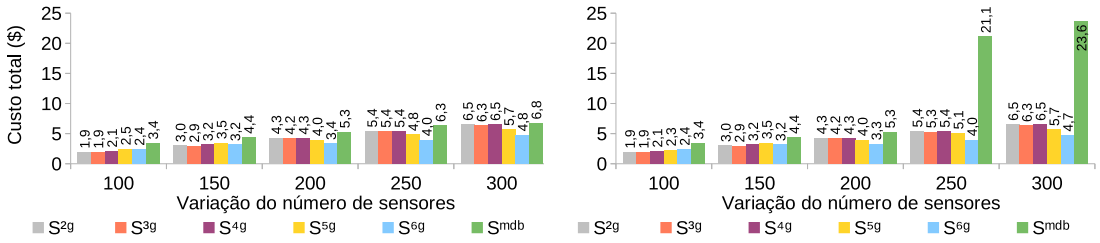
(d) ILP^a com servidores de CC municipal e EC2

Figura 4. Atraso médio fim a fim (Equação 16), variando os algoritmos e cenários

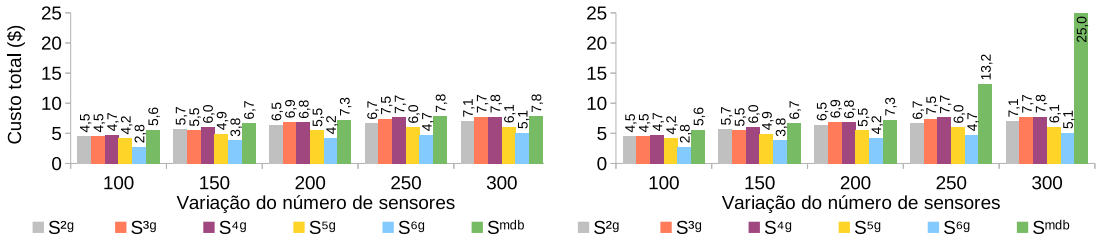
Contrastando as Figuras 3(a) e 3(b), a partir de $|S| = 250$, para manter alta a taxa de aceitação no cenário S^{mbd} é necessário utilizar servidores de CC sob demanda. Isso ocorre porque os sensores priorizam enviar dados para a rede usando a tecnologia com maior largura de banda, e consequentemente menores atrasos, mas com um raio de cobertura menor (Tabela 1). Assim, as opções de atendimento em servidores de EC são limitadas em comparação com o cenário S^{2G}, levando à sobrecarga nos nós de EC disponíveis e transferindo o processamento para a CC. Este ocorrido impacta diretamente nos custos, já que os custos de servidores da EC2 são superiores aos dos servidores municipais (Tabela 3). Observando as Figuras 5(b) e 5(d), em momentos específicos, o custo do algoritmo ILP^c é mais elevado em comparação com o ILP^a, devido a uma taxa de aceitação maior. Neste caso, com uma taxa de aceitação maior, mais servidores estão ativos para efetuar o processamento, aumentando consequentemente o custo operacional.

A função objetivo de minimizar custos teve um impacto significativo em comparação com a que visa reduzir atrasos. Como exemplo, no cenário S^{mdb} com $|S| = 200$, existe uma diferença de $\approx 71.3\%$ no aumento do atraso fim a fim do ILP^a para o ILP^c (Figuras 4(b) e 4(d)), enquanto o custo aumentou $\approx 37.74\%$ do ILP^c para o ILP^a

(Figuras 5(b) e 5(d)). Como exemplo mais significativo, custo aumentou $\approx 123.81\%$ com $S = 100$ e S^{4G} (Figuras 5(b) e 5(d)). Isso destaca a existência de um *trade-off*, logo a escolha da função objetivo deve considerar como o atraso impacta na QoE do serviço.



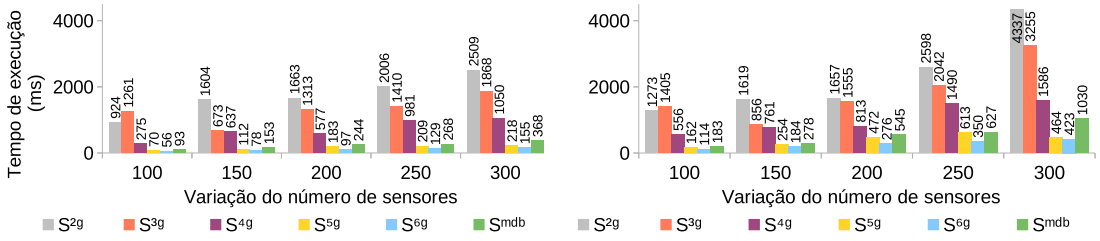
(a) ILP^c com servidor de CC municipal (b) ILP^c com servidores de CC municipal e EC2



(c) ILP^a com servidor de CC municipal (d) ILP^a com servidores de CC municipal e EC2

Figura 5. Custo de operação (Equação 15), variando os algoritmos e cenários

Observando o algoritmo ILP^c nas Figuras 6(a) e 6(b), nota-se que o aumento no número de sensores e servidores de CC processados resulta em um aumento no tempo de execução, fato decorrente da aplicação do algoritmo exato. Aumentos mais significativos são observados no cenário S^{2G} (Figura 6(a)), chegando a $4337ms$. Fato ocorre devido ao maior número de opções de servidores de EC disponíveis para cada sensor se comunicar, em comparação, por exemplo, ao cenário S^{6G}. Neste caso, mais opções de atendimento aumentam o número de variáveis e restrições a serem resolvidas no modelo matemático, elevando o tempo de execução do algoritmo. Esse comportamento de tempo crescente sugere a necessidade, em trabalhos futuros, de desenvolver estudos sobre complexidade e propor heurísticas para gerar soluções em tempos mais baixos. Pelo comportamento similar do ILP^a, optou-se por mostrar apenas os gráficos referentes ao algoritmo ILP^c.



(a) ILP^c com servidor de CC municipal (b) ILP^c com servidores de CC municipal e EC2

Figura 6. Tempo de execução, com o algoritmo ILP^c e variando cenários

6. Conclusões e Trabalhos Futuros

Neste artigo, foi apresentada uma definição do problema de alocação de recursos em EC e CC para atender dispositivos de IoT em uma Cidade Inteligente. Foi formulado um

modelo com ILP, considerando restrições de capacidades de processamento e memória, além do roteamento entre os componentes da rede. Duas funções objetivo foram propostas: minimizar os custos dos servidores utilizados e minimizar os atrasos fim a fim. O problema foi abordado com um algoritmo exato, baseado no modelo em ILP. O tratamento com o algoritmo evitou potenciais erros de decisão e vieses que poderiam surgir com heurísticas, propiciando uma análise assertiva. Foi realizada uma análise das tecnologias de redes. Extensos experimentos foram conduzidos em instâncias da literatura, considerando aspectos atuais das tecnologias de comunicação, incluindo o 6G.

Os experimentos com a tecnologia 6G mostraram um atraso fim a fim de $9ms$, notavelmente inferior a outras tecnologias, como o exemplo do 4G, que apresentou um atraso médio de $410ms$ com o algoritmo ILP^c, e $340ms$ com o ILP^a, tornando-os inviáveis para aplicações de IoT com requisitos de baixo atraso. Os resultados revelaram que a distribuição dos servidores de EC na topologia utilizada não atendeu satisfatoriamente à comunicação entre os dispositivos, especialmente considerando o 6G, resultando em uma baixa taxa de aceitação. Destaca-se que a utilização de componentes de CC alugados sob demanda, apesar de afetar os custos, mostrou-se promissora para absorver demandas não atendidas nos dispositivos originais da rede, mantendo uma alta taxa de aceitação.

A função objetivo que minimiza os atrasos fim a fim foi eficaz, com uma diferença de até $\approx 624.33\%$ em alguns casos em comparação à função que minimiza os custos. Em contrapartida, em alguns cenários, a função que minimiza os custos conseguiu reduzi-los em até $\approx\ 123.81\%$ em comparação àquela que minimiza os atrasos. Considerando que os impactos na QoE podem ser variados e relacionados a aspectos como atrasos e disponibilidade, percebe-se um *trade-off*, no qual a escolha da função objetivo a ser aplicada deve considerar esses impactos.

Em trabalhos futuros, pretende-se investigar a complexidade do problema e desenvolver algoritmos com menor tempo de execução. Por se tratar de um problema de otimização com restrições de capacidade e conservação de fluxo, a resolução exata para instâncias maiores pode demandar um tempo de processamento alto, o que se torna inviável em cenários que requerem tomadas de decisão rápidas. Dessa forma, planeja-se criar algoritmos que combinem técnicas de Inteligência Artificial e heurísticas, visando manter um tempo de resolução baixo, mesmo em instâncias maiores.

Agradecimentos

Os autores agradecem ao CNPq, CAPES e FAPEMIG.

Referências

Alsabah, M., Naser, M. A., Mahmmod, B. M., Abdulhussain, S. H., Eissa, M. R., Al-Baidhani, A., Noordin, N. K., Sait, S. M., Al-Utaibi, K. A., and Hashim, F. (2021). 6G Wireless Communications Networks: A Comprehensive Survey. *IEEE Access*, 9.

Alwis, C. D., Kalla, A., Pham, Q.-V., Kumar, P., Dev, K., Hwang, W.-J., and Liyanage, M. (2021). Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society*, 2:836–886.

Araujo, S. M. A., de Souza, F. S. H., and Mateus, G. R. (2022). A demand aware strategy for a machine learning approach to VNF-PC problem. In *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*, pages 211–219.

Askari, L., Musumeci, F., and Tornatore, M. (2019). Latency-Aware Traffic Grooming for Dynamic Service Chaining in Metro Networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6.

Bari, M. F., Chowdhury, S. R., and Boutaba, R. (2019). ESSO: An Energy Smart Service Function Chain Orchestrator. *IEEE Transactions on Network and Service Management*, 16(4):1345–1359.

CISCO (2023). Cisco HyperFlex - All Flash and Hybrid Server Nodes Spec Sheet. Spec Sheet REV A.25, CISCO SYSTEMS.

Four-Faith (2022). 5Ghz WiFi Router Range Standard. [Online]; acessado 20/12/2023, disponível em https://www.fourfaith.com/industry-news/5g-wifi-range.html.

ITU-R (2015). IMT Traffic Estimates for the years 2020 to 2030. M.2370-0, International Telecommunication Union.

Jia, Y., Wu, C., Li, Z., Le, F., and Liu, A. (2018). Online Scaling of NFV Service Chains Across Geo-Distributed Datacenters. *IEEE/ACM Trans. Netw.*, 26(2):699–710.

Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. M. A., Dang, T. N., and Hong, C. S. (2020). Edge-Computing-Enabled Smart Cities: A Comprehensive Survey. *IEEE Internet of Things Journal*, 7(10):10200–10232.

Kurose, J. F. and Ross, K. W. (2021). *Redes de Computadores e a Internet*. Bookman, Brasil, 8 edition.

Long, X., Wu, J., and Chen, L. (2018). Energy-Efficient Offloading in Mobile Edge Computing with Edge-Cloud Collaboration. In *Algorithms and Architectures for Parallel Processing*, pages 460–475, Cham. Springer International Publishing.

Premsankar, G., Ghaddar, B., Di Francesco, M., and Verago, R. (2018). Efficient Placement of Edge Computing Devices for Vehicular Applications in Smart Cities. In *NOMS - IEEE/IFIP Network Operations and Management Symposium*. IEEE Press.

Queiroz, T. A. d., Canali, C., Iori, M., and Lancellotti, R. (2022). *An Optimization View to the Design of Edge Computing Infrastructures for IoT Applications*, pages 1–30. Springer International Publishing, Cham.

Rosendo, D., Silva, P., Simonin, M., Costan, A., and Antoniu, G. (2020). E2Clab: Exploring the Computing Continuum through Repeatable, Replicable and Reproducible Edge-to-Cloud Experiments. In *2020 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 176–186.

Santos, J., Wauters, T., Volckaert, B., and De Turck, F. (2021). Towards end-to-end resource provisioning in fog computing over low power wide area networks. *Journal of Network and Computer Applications*, 175:102915.

Shah, A. F. M. S., Qasim, A. N., Karabulut, M. A., Ilhan, H., and Islam, M. B. (2021). Survey and performance evaluation of multiple access schemes for next-generation wireless communication systems. *IEEE Access*, 9:113428–113442.

Vailshery, L. S. (2023). Number of IoT connected devices worldwide 2019-2023, with forecasts to 2030. [Online]; acessado 20/12/2023, disponível em https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide.