

# Um Esquema para Alocação Justa de Dispositivos IoT Multifuncionais Ciente dos Recursos Computacionais

Diego M. Rocha<sup>1</sup>, Ayrton M. Porto de Gois<sup>1</sup>, Leandro H. Batista da Silva<sup>1</sup>,  
Fernando Matos<sup>2</sup>, Aldri Santos<sup>3</sup>, Paulo Ditarso Maciel Jr.<sup>1</sup>

<sup>1</sup>Instituto Federal da Paraíba (IFPB) – João Pessoa – PB – Brasil

<sup>2</sup>Universidade Federal da Paraíba (UFPB) – João Pessoa – PB – Brasil

<sup>3</sup>Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte – MG – Brasil

{diego.rocha, ayrton.porto, leandro.batista}@academico.ifpb.edu.br  
fernando@ci.ufpb.br, aldri@dcc.ufmg.br, paulo.maciel@ifpb.edu.br

**Abstract.** *Many IoT scenarios demand continuous capture of information from multifunctional sensors and smart units, as well as the data sending to cloud centers. However, managing the allocation of tasks to these sensors is not simple due to the urgency and priority that each type of data collection requires depending on the needs of the urban environment. This paper presents the POSITRON scheme for managing the sensing allocation in a multifunctional IoT network from previously defined policies. The policies consider the applications characteristics running on the network and the different specifications of the available devices. The POSITRON implementation in a network simulator aims to show its efficiency in allocating resources. The results point out that considering the requirements demanded by applications and the distinct characteristics of multifunctional IoT devices brings benefits in the resource allocation.*

**Resumo.** *Muitos cenários de IoT demandam uma captura contínua de informações de sensores multifuncionais e unidades inteligentes, além do envio desses dados para centros de nuvem. Entretanto, a gestão da alocação das tarefas a esses sensores não é simples em razão da urgência e da prioridade que cada tipo de coleta de dados exige dependendo da necessidade do ambiente urbano. Este artigo apresenta o esquema POSITRON para o gerenciamento da alocação de sensoriamento em uma rede IoT multifuncional a partir de políticas previamente definidas. Tais políticas consideram as características das aplicações executadas na rede, bem como as especificações distintas dos dispositivos disponíveis. A implementação do POSITRON em um simulador de redes visa demonstrar a sua eficiência na alocação dos recursos. Os resultados apontam que levar em conta os requisitos demandados pelas aplicações e as características distintas dos dispositivos IoT multifuncionais traz benefício na alocação dos recursos.*

## 1. Introdução

Nas últimas décadas, os centros urbanos têm vivenciado um crescimento populacional sem precedentes. Estima-se que em 2050, cerca de 68% da população mundial viverá em cidades [Klein and Anderegg 2021]. O aumento natural da população nas áreas urbanas, além do crescimento devido à migração das áreas rurais, afetam de forma direta o

cotidiano das pessoas e dos serviços oferecidos aos cidadãos. Uma população maior significa filas maiores, mais trânsito, maior concorrência para conseguir emprego, necessidade de mais hospitais e postos de saúde, mais moradias, aumento na geração, captação e distribuição de energia e água, dentre outros problemas [United Nations 2019].

Para possibilitar esta gestão, faz-se necessária a implantação de equipamentos que permitam acessar, monitorar e disponibilizar informações de condições climáticas, tráfego, saúde, dentre outros. Neste contexto, a Internet das Coisas (IoT, do inglês *Internet of Things*) tornou-se uma tecnologia básica e essencial para a realização dos diversos cenários urbanos inteligentes [Pedroso et al. 2021]. Nela todos os objetos (“coisas”) e todos os ambientes podem apresentar um melhor desempenho através da incorporação de tecnologia, fazendo com que tenham uma identificação única e comuniquem-se entre si para a transferência de dados. Esta interação, guiada pelas ações que estes objetos executam, suporta a criação de infraestruturas inteligentes e autônomas; e cuja a incorporação dos objetos nos ambientes de forma não perceptível aos usuários busca prover uma inteligência funcional [El Bouanani et al. 2019]. Entretanto, apesar de ser uma tecnologia já consolidada e com inúmeras aplicações bem-sucedidas, as redes IoT sofrem de limitações em decorrência de suas características. Uma destas limitações é a má utilização de recursos [Pedroso et al. 2020]. Por exemplo, muitas vezes uma rede IoT implantada em um determinado local é responsável pelo provisionamento de apenas um único serviço, subutilizando os recursos existentes.

Além disso, os dispositivos IoT de diferentes fabricantes podem possuir características físicas diferentes, como capacidade energética da bateria e alcance de transmissão, o que poderia limitar os tipos de aplicações que determinado conjunto de dispositivos pode atender [Perera et al. 2021]. Assim, uma única rede IoT multifuncional que pudesse atender a diferentes aplicações e suportasse diferentes tipos de dispositivos, poderia substituir várias outras redes IoT, diminuindo custos. Logo, os desenvolvedores de aplicações aproveitariam de uma mesma infraestrutura já existente de dispositivos IoT. As iniciativas de pesquisa, entretanto, desconsideram a potencialidade de tal infraestrutura e se concentram em ações de *offloading* das aplicações em algum ambiente de *Edge/Cloud*, como forma de alocar os recursos e reduzir métricas de utilização como consumo energético [Narayanan et al. 2020].

Este trabalho apresenta o esquema POSITRON para o gerenciamento da seleção e alocação de maneira “justa” das demandas de dispositivos IoT multifuncionais em ambientes de sensoriamento urbano. O POSITRON leva em conta as características das aplicações executadas na rede, bem como as especificações distintas dos dispositivos disponíveis. Ele emprega dois níveis de gerenciamento através de uma política que, no primeiro nível, mapeia as aplicações a um perfil previamente definido de carga de trabalho (consumo de recurso). Em um segundo nível de gerência, ele realiza o balanceamento ou a concentração da carga nos dispositivos possíveis. Ressalta-se que a característica de “justiça” na alocação de recursos trata de uma melhor adequação entre o perfil da aplicação e o dispositivo onde é executada. Assim, é intuitivo presumir uma utilização mais eficiente dos recursos em rede. A implementação do POSITRON em um simulador de redes demonstrou a sua eficiência na alocação dos recursos. Os resultados apontam que considerar os requisitos demandados pelas aplicações e as características distintas dos dispositivos IoT multifuncionais traz benefício na alocação dos recursos.

O restante deste artigo está organizado como descrito a seguir. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o esquema POSITRON. A Seção 4 detalha a implementação, a avaliação do POSITRON e os resultados alcançados. Finalmente, a Seção 5 apresenta as conclusões do trabalho.

## 2. Trabalhos Relacionados

Embora a literatura apresente várias soluções para lidar com a alocação de recursos em redes IoT, em geral estão focadas na redução de métricas específicas como latência [Tsai 2018, Zhao et al. 2019, Sangaiah et al. 2020] e consumo energético [Guim et al. 2022, Xavier et al. 2022]; na otimização via modelos matemáticos [Li et al. 2021, Bashir et al. 2022]; ou são normalmente voltadas a um ambiente específico como MEC [Xavier et al. 2020, Bolettieri et al. 2021, Wang et al. 2022], entre outros. Adicionalmente, estas propostas se concentram no *offloading* de aplicações como forma de alocação de recursos.

Em [Tsai 2018], os autores consideram várias restrições em um sistema IoT de alto desempenho na alocação de recursos, a saber, requisitos distintos do usuário, diferentes tipos de dispositivos, enorme necessidade de comunicações, largura de banda de rede e poder computacional limitados. O artigo propõe um algoritmo para resolver o problema da alocação de recursos em tal sistema, objetivando reduzir o custo total de comunicação entre os dispositivos. Para tanto, conceitos de agrupamento de dados e meta-heurística são empregados, além do uso de *gateways* IoT para comunicação mais eficiente. Diferente da proposta aqui descrita, o artigo não considera as características multifuncionais dos nós da rede, nem se utiliza de políticas para um gerenciamento autônomo da rede IoT. Além disso, a eficiência é mensurada em termos da comunicação de dados, desconsiderando outros aspectos dos recursos como processamento e armazenamento.

Em [Zhao et al. 2019], os autores assumem o desafio de alocar os recursos em servidores de borda de forma eficiente, em um cenário crescente do número de dispositivos IoT e do número de serviços em cidades inteligentes. Eles buscam otimizar as aplicações hospedadas com o objetivo de obter um desempenho satisfatório e comparam algoritmos de *offloading* de carga de trabalho, observando o tempo de resposta como a principal métrica de desempenho. Em [Sangaiah et al. 2020], os autores propõem um algoritmo de otimização para a alocação e o agendamento de recursos em sistemas IoT de modo a reduzir o custo total de comunicação entre dispositivos e um *gateway* próximo, visto que a heterogeneidade dos dados e quantidade de recursos a ser alocada em tais sistemas inviabiliza uma abordagem manual. O esquema de alocação justa aqui proposto, por outro lado, observa diferentes aspectos para alocar de forma mais adequada as aplicações nos dispositivos IoT multifuncionais existentes, dentre os quais o aspecto de comunicação pode ser considerado. No entanto, outros requisitos de alocação também podem ser usados, além de uma política de balanceamento ou saturação de carga.

Em [Guim et al. 2022], os autores propõem o gerenciamento autônomo do ciclo de vida para plataformas de borda convergentes, permitindo a orquestração da carga de trabalho com eficiência de recursos em relação à computação verde. Eles apresentam uma solução para configuração de recursos dinâmicos e inteligentes em plataformas de computação de borda que hospedam serviços de múltiplos usuários, garantindo os SLOs para cada serviço. Contudo, os autores definem a eficiência em termos de consumo energético e não em relação aos demais recursos computacionais. Além disso, não consi-

deram dispositivos multifuncionais. Em [Xavier et al. 2022], os autores propõem um algoritmo de alocação de recursos totalmente distribuído para o ambiente IoT, que considera a borda e a nuvem. O algoritmo gerencia o uso de recursos de infraestrutura ao promover a colaboração entre nós de borda. Também suporta a heterogeneidade e os requisitos genéricos das aplicações, bem como reduz a latência e aumenta a eficiência energética na borda. Contudo, estes trabalhos têm o objetivo de otimizar a gestão energética dos dispositivos, enquanto nossa proposta trata da alocação justa de diversos recursos computacionais, podendo, inclusive, balancear a carga de trabalho para economizar bateria.

Em [Li et al. 2021], os autores também consideram diferentes tipos de aplicações em um cenário de rede IoT, além de dispositivos com capacidades computacionais limitadas que afetam a disponibilidade da rede. Para contornar tais desafios, um modelo de *offloading* cooperativo de aplicações IoT multiusuário para um sistema MEC é proposto, com base em uma formulação matemática de programação não linear inteira mista. Os resultados indicam a eficiência nas decisões de alocar parte das aplicações na MEC, bem como a melhoria em métricas de consumo de energia, latência e utilização da rede. Contudo, a característica multifuncional dos dispositivos também não é explorada neste trabalho, muito menos é disponibilizada uma implementação de referência. Em [Bashir et al. 2022], similar à abordagem do artigo anterior no contexto de *offloading*, os autores elaboram uma estratégia de alocação dinâmica de recursos para a *cloud* e *fog*, fazendo uso de regressão logística para o cálculo de carga necessária na borda. Um arcabouço para ranqueamento dos nós disponíveis é utilizado, onde a tomada de decisão é realizada pelo algoritmo TOPSIS. Entretanto, as formulações matemáticas propostas nestes artigos não relacionam diferentes perfis de carga de trabalho com dispositivos multifuncionais distintos. Além disso, utilizam uma abordagem inversa ao esquema aqui proposto, que realiza uma alocação mais adequada das aplicações nos nós finais a partir de suas especificações.

Em [Xavier et al. 2020], os autores propõem um algoritmo de alocação de recursos para um ambiente de CoT (*Cloud of Things*), que combina uma arquitetura de três camadas com dispositivos IoT, nós de borda e a nuvem. A proposta suporta a heterogeneidade de dispositivos e aplicações, aproveitando a natureza distribuída dos nós de borda para promover a colaboração durante o processo de alocação. O algoritmo colaborativo segue uma abordagem baseada em heurística inspirada em um modelo econômico para resolver o problema de alocação de recursos em CoT. Os resultados simulados indicam o uso eficiente dos recursos do sistema enquanto atende aos requisitos de latência e assume diferentes prioridades de aplicações IoT, em comparação com uma abordagem baseada em nuvem de duas camadas. Diferente do esquema proposto, que funciona de maneira centralizada, o algoritmo colaborativo depende da utilização de recursos externos para uma maior eficiência na alocação de recursos. Em [Bolettieri et al. 2021], o problema de múltiplo acesso à borda (MEC) por aplicações executadas em dispositivos IoT é investigado. A ênfase do artigo está na modelagem do transporte e gerenciamento de dados, para os quais é proposta uma arquitetura compatível com MEC, visando o suporte e a operação de vários provedores de serviços de IoT em uma plataforma comum. Com a perspectiva nas aplicações, os autores modelam a colocação dos serviços (*placement*) e o gerenciamento dos dados, levando em conta as dependências à nível de dados entre os serviços IoT e recursos de sensoriamento. Embora admita as características das aplicações no compartilhamento dos recursos na MEC, a arquitetura não considera a multi-funcionalidade dos dispositivos IoT, bem como as distintas especificações de hardware. Um algoritmo

distribuído de *offloading* para alocação de recursos é proposto em [Wang et al. 2022], baseado em aprendizado profundo distribuído no cenário SD-MEC IoT. Várias redes neurais profundas são invocadas em paralelo para gerar a decisão ideal e o agendamento de recursos. Similar ao esquema proposto neste trabalho, características das aplicações são levadas em consideração na decisão de *offloading*. Porém, a estratégia de utilizar MEC busca reduzir a taxa de utilização do sistema, em contrapartida ao que se propõe essa pesquisa com a utilização mais eficiente dos recursos disponíveis.

Apesar de diversas propostas, ora considerando apenas requisitos das aplicações, ora métricas relacionadas a um único recurso computacional, praticamente nenhuma delas aborda dispositivos IoT multifuncionais. Com o POSITRON, por outro lado, busca-se uma alocação justa em tais dispositivos multifuncionais, onde os recursos são usados de modo mais adequado, mitigando a necessidade de realizar *offloading*.

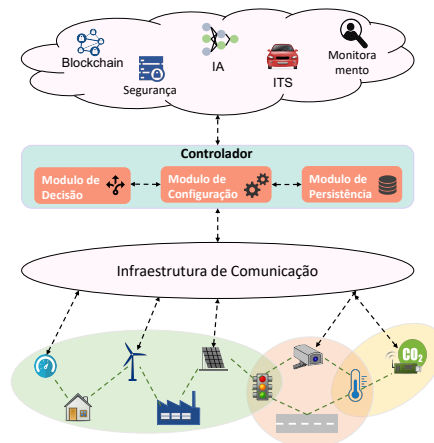
### 3. O Esquema POSITRON

Esta seção detalha o esquema POSITRON para a gerenciamento da alocação justa de dispositivos IoT multifuncionais, em um cenário de aplicações urbanas. Inicialmente, descreve-se a organização hierárquica da infraestrutura de redes onde o POSITRON atua no ambiente urbano IoT e os módulos que compõem o componente Controlador. Em seguida, as políticas de funcionamento e interação entre os dispositivos multifuncionais são apresentadas, de modo a atender as demandas de sensoriamento das aplicações.

#### 3.1. Organização do Ambiente e da Rede IoT Multifuncional

A Figura 1 demonstra a topologia lógica da rede IoT multifuncional considerada nesta pesquisa. A característica multifuncional dos dispositivos se deve ao fato de existirem diversos modelos possíveis, dos mais variados tipos de configurações. Com o avanço tecnológico dos chamados *Single Board Computers*, tais como Raspberry Pi, NVIDIA Jetson Family e BeagleBone, é possível configurar ambientes de rede completamente heterogêneos e modulares, proporcionando dispositivos que podem ser utilizados para diversos fins computacionais. A figura demonstra a topologia de rede do ambiente IoT multifuncional considerado nesta pesquisa, onde tipos distintos de aplicações requisitam serviços de diferentes contextos possíveis (p.ex: IA, *blockchain*, ITS, dentre outros), como esperado em um cenário de cidades inteligentes. Como espera-se também, essas aplicações possuem demandas das mais variadas em termos de hardware. Pode-se imaginar, por exemplo, que o monitoramento de vias públicas exige uma capacidade de armazenamento mínima disponível nos nós da rede. Já aplicações que realizam alguma criptografia podem usufruir de nós com um maior poder computacional em termos de processamento. Ou mesmo aplicações que envolvem sensoriamento de algum fenômeno físico e precisam transmitir os dados de maneira constante na rede. Ainda nesse contexto, por mais que os dispositivos IoT da rede apresentem características heterogêneas, é provável encontrar grupos de nós que possuam configurações semelhantes. Tais semelhanças podem também indicar que certos grupos de nós são mais ou menos apropriados para executar uma ou outra aplicação.

A camada de controle coordena o esquema de alocação justa proposto no POSITRON. No ambiente representado na figura percebe-se que o Controlador, através de seus módulos, serve como interface entre as possíveis aplicações que demandam por recursos e



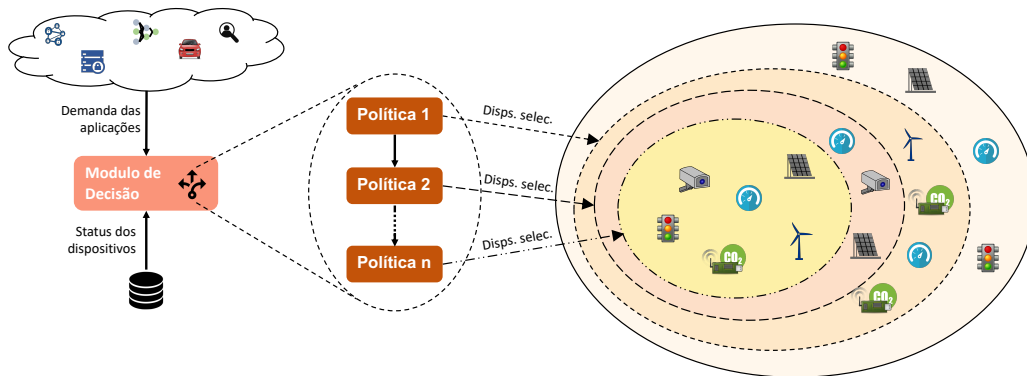
**Figura 1. Topologia da rede IoT multifuncional em um cenário urbano.**

a infraestrutura de rede adjacente. As demandas das aplicações são avaliadas pelo Módulo de Decisão (MD). O MD utiliza políticas para determinar o conjunto mais adequado de dispositivos para atender as aplicações, levando em conta suas demandas e os estados atuais dos dispositivos. Esta abordagem baseada em políticas permite que o sistema tome decisões adaptativas de acordo com o estado atual da rede. O resultado da análise do MD serve de entrada para o módulo seguinte, chamado de Módulo de Configuração (MC). Este módulo é responsável pela comunicação com os dispositivos através da camada de infraestrutura e por aplicar a alocação sugerida pelo módulo anterior. Por fim, o Módulo de Persistência (MP) cadastra e atualiza todas as informações referentes aos nós da rede em uma base de dados. O MP realiza esta atualização periodicamente de forma que o POSITRON tenha uma visão geral e mais atualizada possível do estado da rede. Como exemplo das informações que são registradas, além dos atributos de cada nó relacionados anteriormente, há também o histórico de execução que associa as aplicações aos respectivos dispositivos que as executam.

### 3.2. Políticas de Seleção

A seleção dinâmica de nós do POSITRON, através do MD, utiliza uma hierarquia de políticas para selecionar os dispositivos de acordo com o estado geral da rede e as demandas das aplicações, como ilustra a Figura 2. A utilização de políticas permite que o sistema se auto-configure de acordo com regras pré-estabelecidas, não havendo necessidade de reinicialização ou reimplantação do sistema [Gonçalves et al. 2014]. Além disso, ao empregar esta hierarquia de políticas, o POSITRON consegue realizar sucessivos refinamentos na seleção a fim de buscar o sub-conjunto de nós mais apropriado para atender uma solicitação. Esta abordagem flexibiliza a adição de novas políticas visando diferentes objetivos, como custo financeiro, custo computacional, acordos comerciais, dentre outros.

Utilizou-se três políticas simples em dois níveis hierárquicos para validar o POSITRON, conforme Figura 3. A função *atende(listaNos, recursos, requisitos)* seleciona todos os nós cujos recursos atendem minimamente os requisitos de uma aplicação. Por exemplo, um nó possui a quantidade de mínima de processamento para suportar uma aplicação de criptografia. A função *seleciona(n, listaNos)* seleciona os primeiros  $n$  nós da lista *listaNos*. A função *ordena(listaNos, criterio, ordem)* ordena a lista *listaNos* de acordo com um *criterio* em ordem ascendente ou descendente. Inicialmente, o sistema



**Figura 2. Hierarquia de políticas para seleção de dispositivos IoT.**

aplica uma política de *adequação*, onde escolhe dentre os nós IoT disponíveis, aqueles cujas especificações atendem os requisitos mínimos das aplicações. Já no segundo nível hierárquico, há um refinamento daqueles nós escolhidos anteriormente de acordo com critério desejado pela aplicação. O POSITRON considera então duas políticas, *balanceamento* ou *saturação*. O balanceamento de carga busca equilibrar a utilização dos recursos computacionais nos nós, ordenando a lista de forma ascendente de acordo com o *critério*. Então, os  $n$  primeiros nós são os que atendem aos requisitos da aplicação e que estão menos sobrecarregados. Já na saturação, a lista de nós é ordenada de forma decrescente, fazendo com que os  $n$  primeiros nós também atendam aos requisitos da aplicação, porém os sobrecarregando. Assim, a rede tem mais nós livres para atender futuras requisições.

```

if politica == ADEQUACAO
    then listaNos = atende(listaNos, nos.recursos, app.requisitos)
if politica == BALANCEAMENTO
    then listaNos = seleciona(n, ordenaLista(listaNos, criterio, ASC))
if politica == SATURACAO
    then listaNos = seleciona(n, ordenaLista(listaNos, criterio, DESC))

```

**Figura 3. Políticas de fluxo de trabalho.**

---

### Algoritmo 1 Processo de Seleção de Dispositivo Multifuncional

---

**Require:** AppDesc

- 1: **enquanto** Verdade **faça**
  - 2:   AppDesc  $\leftarrow$  aguardaApp() ▷ linhas 2, 3 e 4 executadas pelo MD
  - 3:   requisitos  $\leftarrow$  estimaReq(AppDesc)
  - 4:   perfil  $\leftarrow$  definePerfil(requisitos)
  - 5:   listaNós  $\leftarrow$  aplica política de *adequação* conforme perfil ▷ MC consulta o MP
  - 6:   **se** (listaNós = Vazia) **então** ▷ MC executa deste ponto em diante
  - 7:     ecoa("Não há nó disponível")
  - 8:     configuraAlertaDisponibilidade()
  - 9:   **senão**
  - 10:    listaNós  $\leftarrow$  aplica política de *balanceamento* ou *saturação*
  - 11:    alocaApp(AppDesc, listaNós)
  - 12:   **fim se**
  - 13: **fim enquanto**
- 

O Algoritmo 1 demonstra a sequência de passos para alocação de uma aplicação, a partir da sua chegada, em um determinado nó da rede. Para seu funcionamento, o algoritmo necessita como entrada de uma descrição da referida aplicação. De uma forma

geral, o procedimento consiste em aguardar a chegada de uma aplicação específica (linha 2) e extrair sua descrição com dados sobre tipo e requisitos mínimos exigidos (p.ex. em termos de processamento, memória e disco). Apesar desta definição nem sempre ser fácil, assume-se neste trabalho que é possível estimar estes requisitos mínimos (linha 3) e, a partir deles, traçar um perfil (linha 4) que será utilizado para o primeiro nível hierárquico da política de gerenciamento. Em seguida, é realizada uma consulta à base de dados (linha 5) para retornar uma lista de nós que atendam minimamente tais requisitos. Importante observar que essa consulta considera a capacidade livre dos nós. Ou seja, todos que possuam capacidade livre maior ou igual aos requisitos mínimos, mesmo que estejam executando quaisquer outras aplicações. Caso esta lista esteja vazia, uma mensagem de indisponibilidade (linha 7) é enviada para os possíveis interessados e um alerta de disponibilidade de nós é configurado (linha 8). Considerando uma lista não vazia, a sequência do processo consiste em testar se o segundo nível hierárquico a ser usado para alocar a aplicação é uma política de balanceamento ou de saturação (linha 10), novamente conforme o perfil da aplicação, de onde são extraídos a quantidade necessária de nós e os critérios para sua seleção. Dependendo do resultado deste teste, a lista de nós será ordenada considerando uma ordenação ascendente, conforme discutido anteriormente. Por fim (linha 11), a aplicação é alocada nos nós da lista ordenada, em concordância com os seus requisitos e com um segundo nível de política de utilização dos nós.

#### **4. Implementação e Avaliação**

Esta seção apresenta a implementação e uma avaliação de desempenho do esquema proposto. Os parâmetros utilizados para configurar os cenários simulados de avaliação são apresentados. Em seguida, os resultados numéricos alcançados são descritos e discutidos.

O POSITRON foi implementado em linguagem C++, no simulador de redes NS-3 versão 3.35. Com relação à representação dos nós IoT, a classe do simulador que implementa os nós na rede foi estendida com atributos comuns aos dispositivos multifuncionais. No geral, todos os nós IoT da rede utilizam os protocolos LrWPAN, 6LoWPAN e IPv6, já implementados no NS-3; além dos novos atributos que caracterizam um nó multifuncional: nível de bateria, taxa de consumo inicial, taxa de consumo corrente, CPU, memória, taxa de transmissão e capacidade de armazenamento. Estas especificações foram codificadas no NS-3, bem como a camada de controle considerando os três módulos funcionais ilustrados na Figura 1. Esta camada é executada em um nó especial chamado de Controlador, que aplica todas as funções correspondentes às políticas de gerenciamento. Para tal, um modelo conceitual de uma aplicação que ecoa pacotes UDP foi utilizado como base, com a reutilização do código responsável por gerar, enviar e tratar os pacotes na rede. O código do esquema proposto está disponível em repositório público<sup>1</sup>.

A Figura 4 mostra um diagrama de sequência que apresenta a comunicação entre os módulos do POSITRON, a partir da chegada de duas aplicações, passando pelo processo de seleção e alocação justa dos nós e finalizando com a execução das mesmas. Destaca-se a fase crítica de seleção do dispositivo, que consiste na consulta pelos possíveis candidatos, na ordenação com base no perfil da aplicação e na política de gerenciamento previamente definida. Neste exemplo em particular, representa-se também a situação hipotética de perda da bateria de um nó escolhido (nó X). Assim, uma vez

---

<sup>1</sup><https://github.com/ifpb/positron>



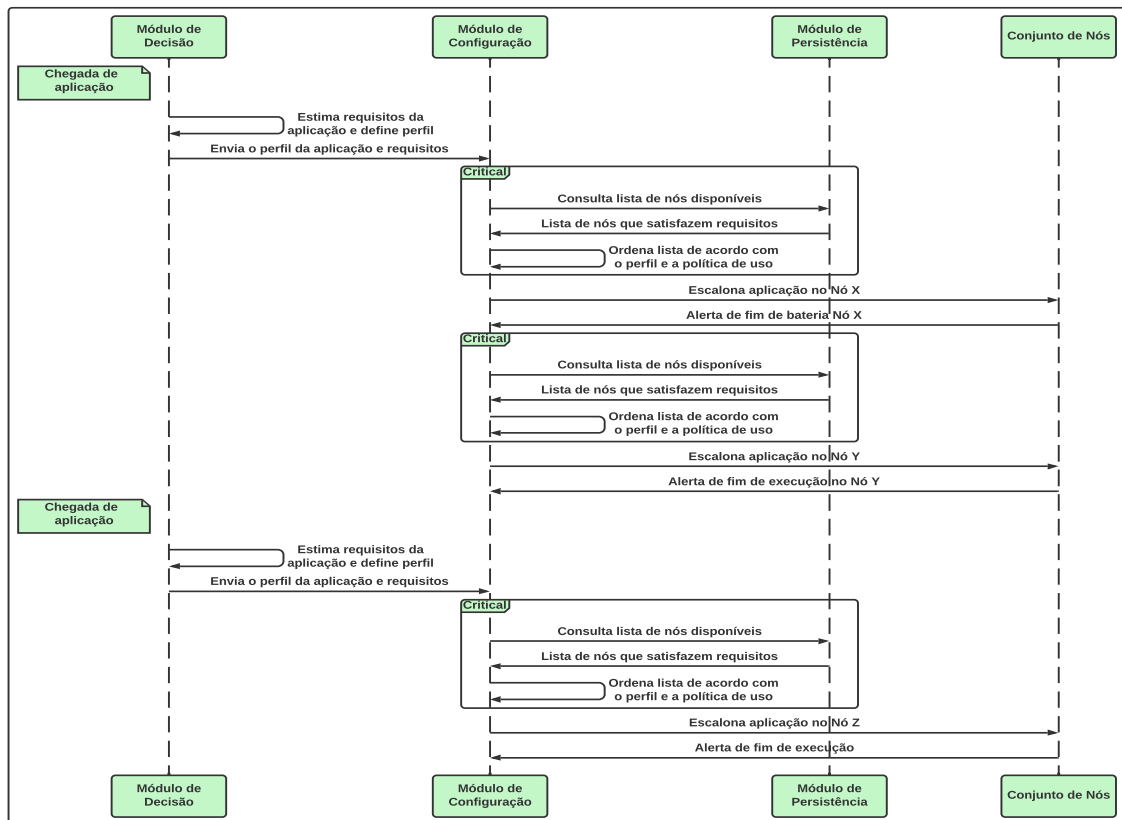


Figura 4. Interação entre módulos do POSITRON para seleção de nós.

recebida a mensagem de alerta de perda de bateria, o MC inicia novamente a sequência crítica de mensagens a fim de selecionar de forma justa um novo dispositivo (nó Y) para executar a aplicação em questão. Também é importante observar que, após a execução completa das aplicações (pelos nós Y e Z), mensagens de alerta também são enviadas ao MC para que o mesmo requisite atualização de estado dos nós junto ao MP. Dessa forma, novas seleções poderão contar com informações mais acuradas a respeito da utilização atual dos nós. Por questão de simplificação, as mensagens de atualização de estado, bem como mensagens de cadastro e saída dos nós no MP, foram omitidas da figura.

#### 4.1. Cenários de Avaliação

A configuração comum a todos os cenários de simulações envolve um nó Controlador e um número de outros nós representando os dispositivos IoT multifuncionais. Adicionalmente, a comunicação em rede acontece através dos protocolos LrWPAN, 6LoWPAN e IPv6. Nesta configuração comum, as especificações dos dispositivos IoT são diferenciadas através de grupos com atributos equivalentes do tipo: *nível de bateria*, *taxa de consumo inicial*, *taxa de consumo corrente*, *CPU*, *memória*, *taxa de transmissão* e *capacidade de armazenamento*. De maneira similar, os grupos de aplicações distintas são diferenciados pelos requisitos mínimos de execução das mesmas, como detalhado a seguir. A chegada dos nós na rede acontece no início da simulação e, conseqüentemente, são cadastrados no Módulo de Persistência. O tempo de chegada das aplicações também são configurados no início das simulações.

Foram considerados três tipos distintos com relação às demandas das aplicações:

um tipo que requisita alto poder de processamento, outro que requisita maiores capacidades de armazenamento e, por fim, um que necessita alta taxa de transmissão de dados em rede. Por outro lado, foram configurados no cenário três tipos distintos de nós IoT, cujas especificações variam de acordo com as demandas solicitadas pelas aplicações. Mais especificamente, um grupo de nós é mais apropriado para as aplicações que requisitam poder de processamento, outro para as aplicações que requisitam armazenamento e um terceiro mais apropriado para aplicações que necessitam alta taxa de transmissão.

Já o número total de nós variou entre 2 e 6 vezes em cada um dos 3 grupos, ou seja, um número total de 6, 9, 12, 15 ou 18 nós no sistema. Foi considerado que um lote com três aplicações, uma de cada tipo, chega ao sistema em intervalos de 10s, após os 5 primeiros segundos de simulação e até o tempo de 125s. Isso configura a chegada de 13 lotes de aplicações, com um número total de 39 aplicações inicializadas (13 de cada tipo considerado). O tempo de duração de cada aplicação foi configurado de acordo com uma distribuição normal, com média de 100s e variância de 30. Dado o esquema de chegada das aplicações ao longo de dois minutos e o tempo médio de 100s destas, o tempo de simulação de cada cenário pode variar entre aproximadamente 3 e 5 minutos. A execução de cada cenário foi repetida 30 vezes para as devidas análises estatísticas e o nível de confiança utilizado foi de 95%. Por questões de simplificação dos cenários, foi desconsiderado o consumo de bateria dos nós a princípio. A Tabela 1 apresenta os valores utilizados nos parâmetros de simulação.

**Tabela 1. Parâmetros utilizados nas simulações.**

Parâmetros	Valores
Número de aplicações	39 (13 de cada tipo) <sup>a</sup>
Número de nós	$n \times 3$ (total de grupos) <sup>b</sup> ( $n = \{2, 3, 4, 5, 6\}$ )
Políticas de gerenciamento	2 níveis (demanda <sup>c</sup> e carga <sup>d</sup> )
Chegada das aplicações	3 (uma de cada tipo <sup>a</sup> ) a cada 10s, durante 2 minutos
Duração das aplicações	distribuição normal com 100s de média e 30 de variância

<sup>a</sup> Criptografia (cri), monitoramento (mon) e sensoriamento (sen).

<sup>b</sup> Processamento (pro), armazenamento (arm) e transmissão de dados (tra).

<sup>c</sup> Associação entre a demanda da aplicação e especificação do dispositivo.

<sup>d</sup> Saturação na utilização dos nós ou balanceamento de carga.

## 4.2. Métricas

Para avaliar o efeito das políticas de alocação nos dispositivos IoT e o desempenho da alocação justa, foram elaboradas as duas métricas descritas a seguir. A primeira delas demonstra quantos nós são efetivamente utilizados, como uma maneira de distinguir a política de balanceamento de carga ou saturação em quantidade menor de dispositivos. Para tanto, assume-se primeiramente um conjunto finito de perfis de aplicações  $P$  e outro que define a quantidade de aplicações que chegaram em cada perfil  $I_p$ . Adicionalmente, os nós IoT são classificados a partir de um conjunto finito de tipos  $G$  e a quantidade de cada nó, em cada tipo, é representada pelo conjunto  $J_g$ . Dessa forma, é possível definir os conjuntos de aplicações e nós IoT que chegam ao sistema, além do número total de nós, respectivamente, como:

$$A = \{a_{p,i} \mid p \in P; i \in I_p\}, N = \{n_{g,j} \mid g \in G; j \in J_g\} \text{ e } |N| = \sum_{g=1}^{|G|} \sum_{j=1}^{|J_g|} (g+j)^0.$$

A primeira métrica, portanto, indica o percentual de nós IoT utilizados ao longo da simulação e é representada por:

$$\mathcal{P}_{utilizados} = \frac{10^2}{|N|} \sum_{g=1}^{|G|} \sum_{j=1}^{|J_g|} \mathcal{U}(n_{g,j}), \quad (1)$$

onde uma função de utilização é definida por,

$$\mathcal{U}(n_{g,j}) = \begin{cases} 1, & \text{se } n_{g,j} \text{ foi utilizado para executar uma aplicação qualquer;} \\ 0, & \text{caso contrário.} \end{cases}$$

Ressalta-se que um percentual de utilização maior ou menor está relacionado à política de alocação dos recursos físicos e não necessariamente indica um valor bom ou ruim.

A segunda métrica diz respeito ao percentual de alocação “justa” das aplicações nos nós IoT, o que representa o índice de “acerto” ao executar tais aplicações nos dispositivos mais adequados. Em outras palavras, dispositivos com especificações de hardware mais apropriadas para os perfis de aplicações que estes executaram. Logo, defini-se o percentual de alocação justa em um determinado nós  $g$  como:

$$\mathcal{P}_{justa}^g = \frac{10^2}{|N|} \sum_{p=1}^{|P|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \mathcal{U}(n_{g,j}, a_{p,i}) \cdot \mathcal{J}(a_{p,i}, g), \quad (2)$$

onde utiliza-se uma especificação da função utilização anterior que define,

$$\mathcal{U}(n_{g,j}, a_{p,i}) = \begin{cases} 1, & \text{se } n_{g,j} \text{ foi utilizado para executar a aplicação específica } a_{p,i}; \\ 0, & \text{caso contrário.} \end{cases}$$

Adicionalmente, utiliza-se a seguinte função que relaciona o perfil da aplicação ao grupo de dispositivos mais adequado:

$$\mathcal{J}(a_{p,i}, g) = \begin{cases} 1, & \text{se o grupo } g \text{ é o mais adequado para executar o perfil } p; \\ 0, & \text{caso contrário.} \end{cases}$$

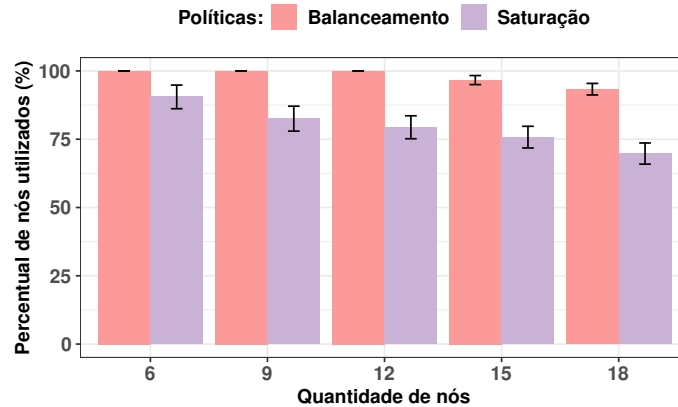
Por fim, defini-se o percentual total de alocação justa como o somatório dos percentuais de cada grupo de nós IoT:

$$\mathcal{P}_{justa} = \sum_{g=1}^{|G|} P_{justa}^g.$$

De acordo com os parâmetros utilizados na simulação (Tabela 1), os conjuntos foram instanciados da seguinte maneira:  $P = \{cri, mon, sen\}$ ,  $I_p = \{1, 2, \dots, 13\} \forall p$ ,  $G = \{pro, arm, tra\}$  e  $J_g = \{1, \dots, n\} \forall g \mid n = \{2, 3, 4, 5, 6\}$ ; este último definido como parâmetro variado em cada cenário de simulação.

### 4.3. Resultados

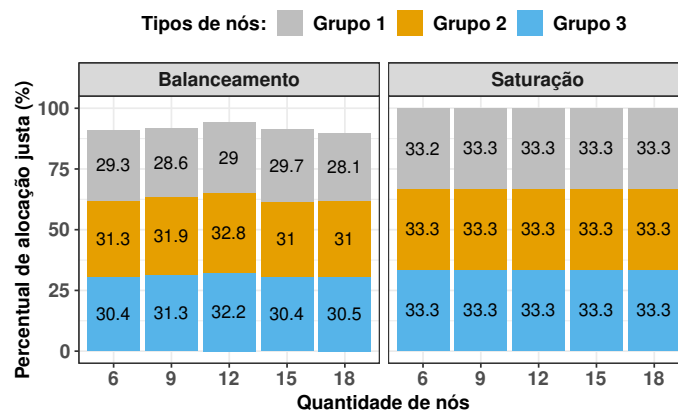
Como validação preliminar do esquema implementado, os resultados a seguir ilustram a execução de cenários de simulação envolvendo três tipos distintos de aplicações, três especificações diferentes de dispositivos IoT e dois níveis de política de alocação dos recursos. Como visto na Figura 5, há uma diferenciação no uso total dos recursos a partir da política aplicada, seja ela de balanceamento ou de saturação dos mesmos. Como esperado, uma política que preza pelo balanceamento dos recursos emprega uma quantidade maior dos nós disponíveis e pode ser aplicada em situações onde os vários nós de rede possuem carga (bateria) suficiente para execução das aplicações. Por outro lado, em uma situação onde é mais vantajoso usufruir de um número menor de nós dado, por exemplo, um cenário de economia de bateria, a política de saturação de determinados nós utiliza sempre um percentual menor destes.



**Figura 5. Percentual de utilização dos nós a partir das políticas de balanceamento ou saturação dos dispositivos (Equação 1).**

Os dois gráficos na Figura 6 ilustram a capacidade de alocar as aplicações devidamente, considerando as demandas específicas e as configurações dos nós existentes. Como pode ser visto no cenário onde se busca pela saturação dos dispositivos, as aplicações são alocadas da maneira mais “justa”, levando em consideração as configurações dos nós correntes. Isso significa que praticamente 100% das aplicações foram alocadas nos nós mais adequados para sua execução, validando o funcionamento do esquema de gerenciamento proposto. Por fim, no cenário onde a política de balanceamento foi empregada, também houve um alto percentual de acerto, com não menos do que 89% das aplicações alocadas devidamente. Conclui-se que, mesmo utilizando uma política que busca balancear o uso dos recursos e, dessa forma, economizar outros, ainda assim é

possível um bom cenário de eficiência na alocação das aplicações. O preço que se “paga” pelo balanceamento não traz grande prejuízo na “justiça” exercida durante a alocação.



**Figura 6. Percentual de alocação justa das aplicações nos nós (Equação 2).**

A partir dos resultados apresentados nas duas figuras, é importante ressaltar que uma alocação “injusta” de uma aplicação a um respectivo dispositivo, não significa que a mesma não foi executada. Esta situação representa uma alocação onde a aplicação seria mais apropriada para outro tipo de dispositivo, contudo não deixou de ser executada. Na média dos experimentos realizados, por exemplo, um máximo de aproximadamente 10% das aplicações são alocadas “injustamente”, no cenário balanceado. No geral, a taxa de execução das aplicações foi de praticamente 100% considerando todos os cenários.

## 5. Conclusões

Este trabalho propôs o esquema POSITRON para o gerenciamento da alocação de sensoriamento em uma rede IoT multifuncional a partir de políticas previamente definidas que consideram, não apenas as características das aplicações executadas nesta rede, mas também as especificações distintas dos nós disponíveis. O esquema POSITRON foi implementado no simulador de redes NS-3 com o intuito de apresentar indícios da eficiência alcançada em um cenário de redes IoT. A partir dos experimentos executados, concluiu-se que o POSITRON realiza de forma satisfatória o gerenciamento dos dispositivos multifuncionais através das políticas propostas. Contudo, pretende-se como trabalho futuro caracterizar melhor os requisitos demandados pelas aplicações, bem como funcionalidades de sensoriamento dos nós IoT. Além disso, uma investigação de como propor políticas de gerenciamento para dispositivos IoT heterogêneos também se faz necessária.

## Agradecimentos

Este trabalho foi realizado com apoio do Edital PRPIPG/IFPB nº 17/2020 PIBITI/CNPq, da Chamada Interconecta nº 02/2021 PIBICT/IFPB, do Edital PROPESQ/PRPG/UFPB nº 03/2020 e do Edital Universal FAPESQ nº 09/2021. A estes programas de fomento são prestados os devidos agradecimentos.

## Referências

Bashir, H. et al. (2022). Resource allocation through logistic regression and multicriteria decision making method in IoT fog computing. *Trans. on Emerging Telecom. Technologies*, 33(2).

- Bolettieri, S. et al. (2021). Application-aware resource allocation and data management for MEC-assisted IoT service providers. *Journal of Network and Computer Applications*, 181:103020.
- El Bouanani, S., El Kiram, M. A., Achbarou, O., and Outchakoucht, A. (2019). Pervasive-Based Access Control Model for IoT Environments. *IEEE Access*, 7:54575–54585.
- Gonçalves, N. M. F., Santos, A. L. d., and Hara, C. S. (2014). A Policy-based storage model for sensor networks. In *IEEE Network Operations and Management Symp. (NOMS)*, pages 1–8.
- Guim, F., Metsch, T., et al. (2022). Autonomous Lifecycle Management for Resource-Efficient Workload Orchestration for Green Edge Computing. *IEEE Transactions on Green Communications and Networking*, 6(1):571–582.
- Klein, T. and Anderegg, W. R. (2021). A vast increase in heat exposure in the 21st century is driven by global warming and urban population growth. *Sustainable Cities and Society*, 73:103098.
- Li, X., Zhao, L., et al. (2021). A cooperative resource allocation model for IoT applications in mobile edge computing. *Computer Communications*, 173:183–191.
- Narayanan, A. et al. (2020). Key Advances in Pervasive Edge Computing for Industrial Internet of Things in 5G and Beyond. *IEEE Access*, 8:206734–206754.
- Pedroso, C., de Moraes, Y. U., Nogueira, M., and Santos, A. (2020). Managing consensus-based cooperative task allocation for IIoT networks. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6.
- Pedroso, C., de Moraes, Y. U., Nogueira, M., and Santos, A. (2021). Relational Consensus-Based Cooperative Task Allocation Management for IIoT-Health Networks. In *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 579–585.
- Perera, A. et al. (2021). Light-based Internet of Things: Implementation of an Optically Connected Energy-autonomous Node. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7.
- Sangaiah, A. K. et al. (2020). IoT Resource Allocation and Optimization Based on Heuristic Algorithm. *Sensors*, 20(2).
- Tsai, C.-W. (2018). SEIRA: An effective algorithm for IoT resource allocation problem. *Computer Communications*, 119:156–166.
- United Nations (2019). Department of Economic and Social Affairs, Population Division (2019). Technical report, World Population Prospects 2019: Highlights (ST/ESA/SER.A/423).
- Wang, Z. et al. (2022). Computation offloading and resource allocation based on distributed deep learning and software defined mobile edge computing. *Computer Networks*, 205:108732.
- Xavier, T. C. et al. (2020). Collaborative resource allocation for Cloud of Things systems. *Journal of Network and Computer Applications*, 159:102592.
- Xavier, T. C. et al. (2022). Managing Heterogeneous and Time-Sensitive IoT Applications through Collaborative and Energy-Aware Resource Allocation. *ACM Trans. Internet Things*, 3(2).
- Zhao, L., Wang, J., et al. (2019). Optimal Edge Resource Allocation in IoT-Based Smart Cities. *IEEE Network*, 33(2):30–35.