UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Felipe Backes Kettl

Gerência autonômica para aperfeiçoar a QoS (Qualidade de Serviços) em cloud computing (Fog e Edge) e Internet of Things

Florianópolis
2022

RESUMO

Nos últimos anos vimos uma rápida expansão na quantidade de dispositivos IoT (Internet of Things) sendo utilizados, o que acaba resultando em redes de larga escala. Com essas redes, surgem tanto novas possibilidades e paradigmas, quanto novos problemas. Melhorar a latência, a distribuição de tarefas, o throughput de um sistemas e outros problemas nesses sistemas é um desafio presente. Este trabalho estuda os ambientes Fog-IoT que utilizam gerenciamento autônomo para melhorar a qualidade do serviço ao usuário e apresenta uma breve *survey* acerca do assunto.

1 INTRODUÇÃO

1.1. MOTIVAÇÃO

Conforme as tecnologias de comunicação e os dispositivos inteligentes evoluíram, surgiram novas possibilidades de tornar diferentes tarefas, de diferentes áreas, autonômicas. O advento da Fog Computing permitiu que diversos ambientes, antes inviáveis devido a baixa qualidade de serviço para os usuários, se tornassem viáveis.

Segundo Mehta e Elmroth (2018), a Fog Computing pode ser utilizada para permitir que aplicações IoT (Internet of Things) interajam com os recursos da Cloud de forma eficiente, e, portanto, pode ser considerada uma extensão do conceito de Cloud computing.

Nas últimas décadas houve uma grande exploração de ambientes autonômicos, como, por exemplo, smart houses. Esses ambientes possuem uma vasta gama de possibilidades, desde controle autonômico de alarmes de incêndio para conservar energia, até portas e janelas que abrem sozinhas. Para isso, eles contam com inúmeros dispositivos IoT, tornando a Fog indispensável para manter a qualidade do serviço (MOCRII; CHEN; MUSILEK, 2018).

Um dos problemas que surgem é que a quantidade de dispositivos IoT em uma rede cresce muito rápido. Se em apenas um apartamento houver algumas dezenas de dispositivos, então um prédio inteiro teria centenas deles. Por isso, fizemos uma *survey* no estado da arte atual em questões de soluções autonômicas para melhorar a qualidade de serviço em ambientes Fog-IoT.

1.2 JUSTIFICATIVA

O trabalho se justifica na crescente quantidade de sistemas Fog-IoT, o que traz consigo diversos desafios. Nesses ambientes cada vez maiores torna-se difícil garantir a qualidade de serviço para os usuários, ainda mais se feito de forma manual. Portanto, é essencial aprofundar na melhoria da QoS por meio de gerência autonômica.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo principal é trazer pesquisas no estado da arte acerca de gerência autonômica para melhorar a qualidade de serviço de sistemas Fog-IoT. Além disso, verificar problemas que atingem esses sistemas e ver possíveis soluções.

1.2.2 OBJETIVOS ESPECÍFICOS

- Fazer a pesquisa bibliográfica e construir uma tabela com pesquisas atuais dentro do escopo do problema trabalhado;
- Relacionar e discutir os trabalhos selecionados, elencando trabalhos futuros e possíveis problemas;
- Definir conceitos no estado arte sobre *Internet of things, Fog computing, Cloud computing* e *Autonomic management*.

1.3 ORGANIZAÇÃO DO ARTIGO

O trabalho é organizado em seis seções, sendo elas: 1. introdução, 2. conceitos básicos, 3. trabalhos correlatos, 4. aspectos relevantes, 5. problemas existentes e 6. soluções possíveis. Na seção 1, comenta-se acerca das motivações, justificativas e objetivos desse projeto. Na seção 2, são descritos os principais conceitos do trabalho. Na seção 3, é feita uma revisão bibliográfica sistemática e apresentado dois trabalhos correlatos. Na seção 4, dissertamos sobre os trabalhos lidos, buscando destacar partes importantes dos mesmos, em 5., listamos os problemas existentes dentro do tema escolhido e em 6, comentamos quais são as possíveis soluções explicitadas na literatura.

2 CONCEITOS BÁSICOS

2.1 INTERNET OF THINGS (IOT)

A Internet of Things (IoT) é a conexão entre a internet e os mais diversos objetos, formando um ambiente de intercomunicação inteligente. Tal ambiente é regido por três princípios: seus objetos são identificáveis, comunicáveis e conseguem interagir com o meio em que estão (CARDOSO et al., 2019).

As limitações da capacidade computacional dos aparelhos IoT tornam a Cloud uma grande aliada desse ambiente, possibilitando uma solução prática para essas limitações. Porém, a grande distância entre esses objetos e a Cloud pode causar atrasos muito altos, portanto um ambiente puramente Cloud-IoT é mais limitado e pode ter uma baixa qualidade de serviço para certas aplicações (CRUZ et al., 2021).

Nas aplicações IoT que requerem processamento de dados em tempo real, é esperado uma grande quantidade de sensores, os quais geram problemas para o processamento de dados, serviços de gerenciamento, tomadas de decisão e, consequentemente, latência nos serviços para o usuário (CHEGINI et al., 2021).

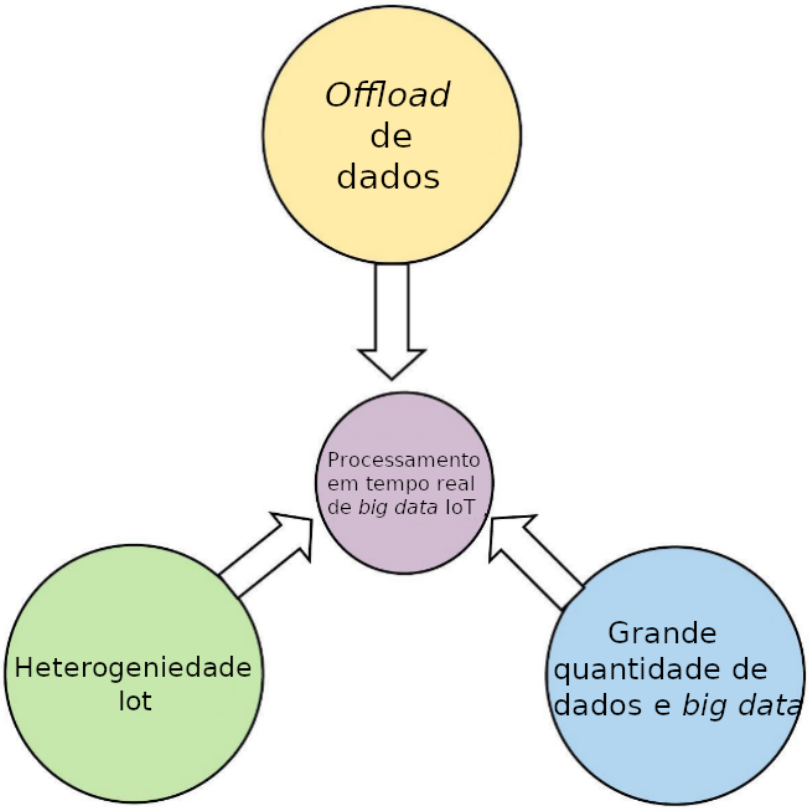
Na Figura 1 é possível ver os principais desafios dos ambientes IoT que requerem processamento em tempo real:

- Offloading de dados: Latência muito alta é inaceitável nos processos IoT real-time. Para melhorar isso, uma das soluções possíveis é diminuir a troca de dados entre os aparelhos IoT e a Cloud por meio do uso de Fog computing.

- Heterogeneidade: Os inúmeros dispositivos que compõem esse meio possuem diferentes arquiteturas, requerimentos e capacidades, o que torna a integração com a Cloud e a Fog mais complexa.

- Grande quantidade de dados e Big Data: O enorme fluxo de dados, resultante da conexão de um número cada vez maior de sensores e objetos à Internet, torna as tarefas de hospedagem, agendamento, processamento e tomada de decisão difíceis de lidar.

Figura 1 – Desafios das aplicações IoT



Fonte: Adaptado de Chegini et al. (2021).

2.2 CLOUD COMPUTING

A Cloud Computing é uma infraestrutura composta por inúmeros computadores que providencia recursos computacionais de forma remota, a qual consegue processar uma grande quantidade de dados, sendo portanto uma grande aliada das aplicações IoT. O serviço é rápido de ser configurado e providenciado, porém sua centralização leva ao aumento da distância com o usuário, o que causa aumento na latência (CARDOSO et al., 2019).

Esse paradigma possui IaaS, PaaS e SaaS como seus três modelos mais famosos de serviço:

- Infrastructure-as-a-Service (IaaS): O provedor fornece todos os recursos computacionais em um ambiente virtual, junto com todas as dependências e frameworks necessários, fornecendo uma grande flexibilidade ao usuário (FARIDI et al., 2021). Temos a Amazon EC2 como exemplo de um provedor IaaS.

- Platform-as-a-Service (PaaS): É fornecido um ambiente de desenvolvimento altamente equipado, providenciando ferramentas de processamento, assim como acesso a sistemas operacionais (como windows e linux) sem a necessidade de uma VM (FARIDI et al., 2021). Dois exemplos de PaaS são o Windows Azure e a Amazon Web.

- Software-as-a-Service (SaaS): Trata-se do fornecimento de um software em conjunto com suas configurações e suporte necessário, com funções adicionais, como armazenamento de dados na Cloud de forma segura. Algumas companhias ainda disponibilizam serviços adicionais como gerenciamento de projetos, exemplificado por Faridi et al. (2021). Um exemplo bem conhecido é o Google Drive.

2.3 FOG COMPUTING

A Fog computing é um paradigma que aproxima as funções da Cloud aos dispositivos IoT, diminuindo a distância entre os objetos do meio e o processamento de dados. Essa proximidade permite que esses objetos localizem uns aos outros com maior rapidez do que utilizando apenas a Cloud. Além disso, a Fog ajuda a reduzir os problemas de escalabilidade e de carga que ocorrem ao aumentar o número desses dispositivos (AAZAM et al., 2022).

Ambientes compostos por Fog e Cloud permitem que os processos sejam divididos entre ambas, de forma que apenas uma fração dos dados precisa viajar pela Wide Area Network (WAN), diminuindo a carga da Cloud (TADAKAMALLA; MENASCé, 2019). A principal vantagem desse paradigma é reduzir o tráfego de dados necessário entre um ambiente IoT e a Cloud, o que resulta em uma latência menor e, consequentemente, uma qualidade de serviço melhor (CRUZ et al., 2021).

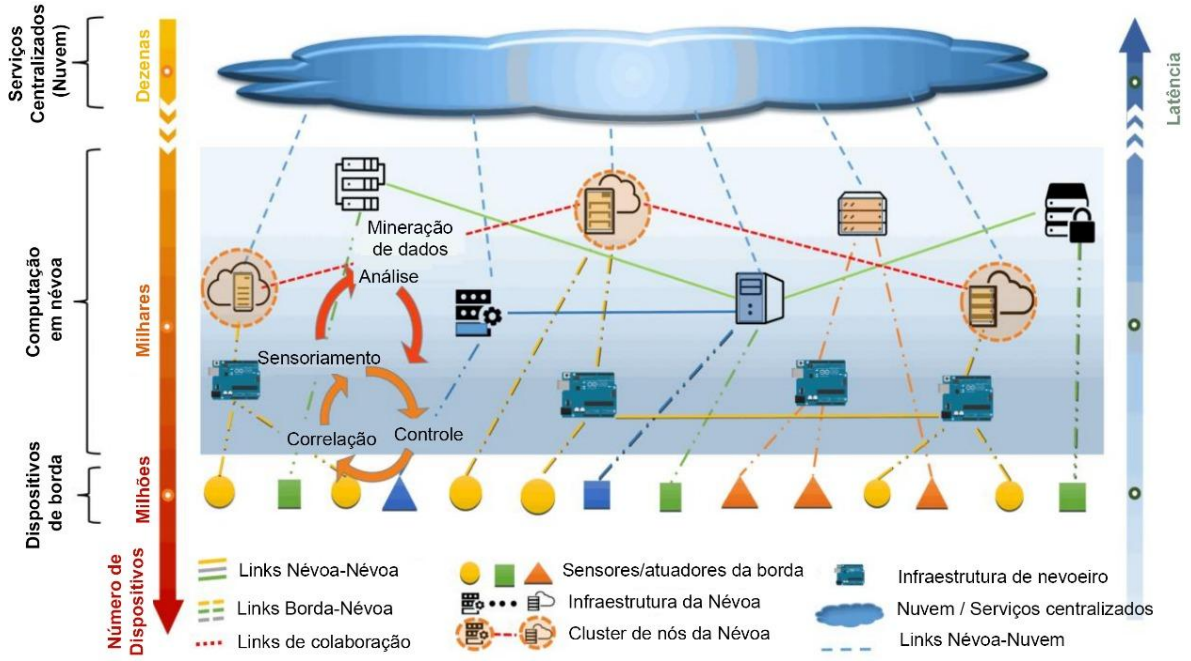
A Figura 2 demonstra como a Fog computing interage com a Cloud para atender dispositivos inteligentes. A Fog não é uma camada obrigatória nesse ambiente, assim como a Cloud não é obrigatória para o atendimento das demandas da Fog. Portanto, a arquitetura é dinâmica, podendo ser moldada conforme as necessidades da aplicação.

2.4 SMART ENVIRONMENTS AND AUTONOMIC MANAGEMENT

Smart Environments incluem desde casas até cidades inteiras. Essas áreas possuem recursos IoT que permitem automatizar diversas funcionalidades, melhorando a qualidade de vida. Em casas inteligentes, por exemplo, é possível reduzir gastos de energia automatizando luzes e alarmes de incêndio (SAMPAIO et al., 2021).

Sistemas capazes de realizar tarefas com pouca ou nenhuma intervenção humana são considerados sistemas autonômicos (DEHRAJ; SHARMA, 2021). Além disso, esse sistema deve ser capaz de se adaptar a mudanças imprevisíveis ao mesmo tempo em que abstrai as necessidades do usuário (SAIF; NIRANJAN; AL-ARIKI, 2021).

Figura 2 – Modelo de computação em névoa.



Fonte: Adaptado de Iorga et al. (2018).

3 TRABALHOS CORRELATOS

Para este trabalho, procuramos dissertar sobre o estado da arte de gerência autonômica para melhorar a qualidade de serviço de ambientes Fog-IoT. Para isso, fizemos uma revisão bibliográfica sistemática e apresentaremos 2 trabalhos relevantes: “Content-centric data and computation offloading in ai-supported fog networks for next generation iot” e “Fog node self-control middleware: enhancing context awareness towards autonomous decision making in fog colonies”.

3.1 Revisão Bibliográfica Sistemática

Tabela 1 – Revisão Bibliográfica

Palavra-Chave Total	Total
“Fog”	110.000
“IoT”	375.000
“QoS”	47.800
“Autonomic management”	1.150
“Fog” and “IoT”	36.000
“Fog” and “QoS”	16.400
“Fog” and “Autonomic management”	278
“IoT” and “QoS”	23.600
“IoT” and “Autonomic management”	522
“Fog” and “IoT” and “Autonomic management”	229
“Fog” and “IoT” and “QoS”	16.100
“Fog” and “IoT” and “Autonomic management” and “QoS”	172

A revisão foi realizada utilizando o Google Scholar. As palavras-chave foram pesquisadas em inglês e com aspas duplas. A pesquisa foi limitada a artigos publicados após 2018. Como visto nos dados acima, a quantidade de trabalhos cai consideravelmente ao adicionarmos “Autonomic management” em qualquer combinação, e há um número ainda mais escasso quando se relaciona com “QoS”(Qualidade de serviço) , o que reforça a necessidade de mais pesquisas na área.

3.2 CONTENT-CENTRIC DATA AND COMPUTATION OFFLOADING IN AI-SUPPORTED FOG NETWORKS FOR NEXT GENERATION IOT

Segundo Kök e Özdemir (2022), os dispositivos IoT atuais facilitam a coleta de conteúdo e a disseminação de diversas aplicações, como sensores ambientais, monitoramento, download de vídeos, etc, gerando quantidades massivas de dados. Porém, transferir todos esses dados para um servidor Cloud pode resultar em congestionamento da rede, alta latência e uma baixa qualidade de serviço em aplicações IoT de tempo real. Diferente de Fog computing, redes tradicionais com arquiteturas baseadas em IP não foram construídas para suportar os requerimentos dos futuros sistemas IoT. Nesse contexto, modelos de redes mais recentes, como o Content-Centric Networking(CCN) têm potencial de resolver esses problemas.

No trabalho os autores introduzem uma camada virtual baseada em IA, chamada FogOrch, que permite recursos da Fog a auto-otimização em termos de transferência de dados de acordo com os requerimentos do sistema. Também propõem um algoritmo DRL baseado em Deep Q-Network (DQN) que gerencia os diferentes recursos da Fog de forma dinâmica e eficiente via FogOrch. Por fim, eles apresentam o design da arquitetura que suporta a FogOrch.

Nas avaliações de performance, o algoritmo DRLOS apresentou custo computacional menor que os outros algoritmos testados, CNOS, ROS e CMOS. As razões para isso é que o algoritmo DRLOS não impõem carga extra no sistema, visto que os servidores Fog distribuem o trabalho de forma balanceada de acordo com a capacidade computacional. Por fim, o autor conclui que a arquitetura e o algoritmo apresentado são promissores para os problemas apresentados no trabalho.

3.3 FOG NODE SELF-CONTROL MIDDLEWARE: ENHANCING CONTEXT AWARENESS TOWARDS AUTONOMOUS DECISION MAKING IN FOG COLONIES

Nikolopoulos et al. (2022) apresentam outro aspecto da *Fog computing,* as *Fog Colonies.* Os *nodes* Fog são agrupados em colônias, permitindo que eles compartilhem recursos e implementem as mesmas políticas . Neste trabalho os autores discutem o desenvolvimento de um "*Fog Node Self-Control Middleware*", que aprimora a percepção de colônias sem um controle central e permite que *nodes* façam decisões sofisticadas quando realizarem ações que podem ser melhoradas pelo contexto operacional, como transferência de recursos e distribuição de tarefas.

O middleware foi desenvolvido usando Java, por isso os Fog nodes precisam ser capazes de rodar a JVM. Porém ele poderia ser desenvolvido em qualquer linguagem, os autores recomendam fortemente o uso de linguagens baseadas em C para poder implementar o middleware em hardwares menos potentes.

Os resultados dos testes realizados apontaram que para colônias com 5 ou menos *nodes* não houveram melhoras perceptíveis, porém os ganhos aumentam drasticamente ao atingir 10 ou mais *nodes*. Por fim, os autores concluem que para melhorar a escalabilidade das *Fog colonies* é essencial que os *Fog nodes* operem de maneira autônoma, sem uma unidade central de cooperação. Ademais, os resultados demonstram que a descentralização de *Fog colonies* é algo relevante e deve ser explorado com mais pesquisas.

4. ASPECTOS RELEVANTES

Percebe-se que apesar de eficiente, o uso de gerência autonômica no contexto do IoT ainda tem muitos desafios no que tange custo, facilidade na implementação e na escassa quantidade de pesquisas se comparado com soluções mais populares. Ademais, fica claro que a gerência autonômica é extremamente abrangente e pode ser utilizada para diversos tipos de sistemas Fog-IoT.

No trabalho de Kök e Özdemir (2022), é utilizado algoritmos de deep learning e considerada a utilização de inteligências artificiais na rede. Já em Nikolopoulos et al. (2022), vemos um *middleware* que foca em sistemas de *Fog colonies.* Em Sampaio et al. (2021) é proposto um sistema autonômico baseado em Fog computing que gerencia o ciclo de ativação dos dispositivos IoT da rede, a *Central IoT* (CIoT). Hou et al. (2016) propõe a ideia de "*Vehicular Fog Computing* (VFC)", em que se usa um veículo como infraestrutura para computação e comunicação.

Além disso, é interessante destacar a diversidade no que tange a solução de problemas desses sistemas. Sampaio et al. (2021) utiliza seu sistema para reduzir o consumo energético de smart *houses*, já o sistema de Jeurkar et al. (2020) vê aplicações na indústria na redução do consumo de água. Rahmani et al. (2018) propôs uma arquitetura assistida por Fog para o sistema de saúde. Ni et al. (2017) trouxe uma estratégia de alocação de recursos baseada em *"priced timed Petri nets* (PTPNs)" para *Fog computing.*

Considerando que tanto IoT quando Fog computing são tecnologias relativamente novas e - somado ao fato de que redes estão constantemente evoluindo, vide 6G e 7G - podemos considerar ambos como campos pouco explorados e, portanto, mais pesquisas são necessárias.

5. PROBLEMAS EXISTENTES

Os diferentes trabalhos apresentados possuem lacunas em relação à solução de problemas. Em Kök e Özdemir (2022), os autores mencionam a falta de soluções que de fato trazem como implementá-las em cenários reais, por isso, eles trazem uma estrutura de rede inovadora, a FogOrch, a qual o algoritmo pode ser implementado em arquiteturas do mundo real. Além disso, há diversas categorias de problemas. Naha et al. (2018) apresentam algumas delas: Problemas de infraestrutura, plataforma e aplicação.

Os problemas de infraestrutura remetem à questão de implementação, falta de padronização na arquitetura para *Fog Computing* e a interoperabilidade dos sistemas. A rápida evolução da tecnologia e a necessidade de se adaptar ao mercado torna a infraestrutura um dos desafios da *Fog Computing* .

Além disso, existem os problemas relacionados à plataforma, são eles: Gerenciamento de recursos, falha no gerenciamento, comunicação entre diversas camada, necessidade de participação do usuário e privacidade e segurança. A plataforma permite o software e o hardware a coexistirem remotamente, sendo indispensável nessas arquiteturas.

Por fim, vemos os problemas de aplicação: Gerenciamento do serviço e modelagem da aplicação. No paradigma da *Fog Computing* é esperado bilhões de dispositivos IoT, divididos entre sensíveis ou não a tempo, por isso, as necessidades das diversas aplicações variam bastante e gerenciar isso se torna um desafio.

6. SOLUÇÕES POSSÍVEIS

Tendo em vista os problemas levantados, divididos entre problemas de infraestrutura, plataforma e aplicação, há diversos trabalhos que demonstram soluções promissoras para parte do, ou todo, problema.

Hou et al. (2016) traz uma ideia muito interessante ao introduzir veículos no sistema, permitindo nodes móveis, os quais podem se alocar baseado na demanda, fornecendo menor QoS para seus usuários. Em relação às dificuldades de implementação, seria interessante o incentivo a estudos que estimem o custo e a dificuldade de implementar *novels architectures* promissoras.

Além disso, podemos mencionar o algoritmo de Ni et al. (2017) que realiza a alocação de recursos em ambientes de Fog Computing. Ao utilizar esse algoritmo em sistemas que contêm uma grande heterogeneidade de dispositivos IoT, devemos conseguir distribuir melhor os recursos do sistema, permitindo atender aos diversos requisitos de cada aparelho.

Outro aspecto promissor é a gerência autonômica, a qual nos permite criar *Fog Nodes* com *context-awareness* como visto em Sampaio et al. (2021), permitindo que o sistema seja regulado de acordo com as suas condições.

Por fim, o uso de simulações torna mais fácil testar e aprimorar sistemas IoT, por isso vale a menção do trabalho de Kodali e Kirti (2020), onde é feita uma simulação de uma rede IoT utilizando o simulador NS-3. Simulações mais complexas podem ser feitas e há diversas ferramentas que permitem a análise dos dados obtidos, como o NetAnim, o qual permite a visualização de uma animação do funcionamento da rede.

7. PROJETO E DESENVOLVIMENTO DE UMA PROPOSTA

Todos os problemas apresentados na seção 5 possuem uma barreira semelhante, a dificuldade em reproduzi-los e estudá-los. Por isso, para este trabalho, propomos o uso de simulações para verificar e simular problemas que podem ocorrer em redes de larga escala. Para isso, foi realizado um estudo, utilizando simulações, acerca dos problemas das redes de larga escala. O simulador escolhido foi o NS-3. O foco deste estudo foi entender o funcionamento do NS-3 e montar algumas simulações que permitam entender quais problemas afetam consideravelmente uma rede de larga escala.

Para as simulações, foram utilizados dois NodeContainer, um para o servidor, contendo apenas um node, e outro para os dispositivos, contendo um número arbitrário de nodes, como visto na figura 12.

Figura 12 – Contêiner dos *nodes*.

```
46  // Create N nodes
47  NodeContainer devices;
48  devices.Create (iotDevices);
49
50  // server node
51  NodeContainer serverNode;
52  serverNode.Create (1);
```

Fonte: Produção original.

Além disso, a conexão escolhida foi o WiFi. Para o channel e a camada PHY, utilizamos as configurações padrão do YansWifiChannelHelper e o YansWifiPhyHelper, respectivamente. Na figura 13 observa-se a criação desses módulos, assim como a definição do SSID da rede e a declaração da classe que irá configurar a camada MAC.

Figura 13 – Configurações gerais do WiFi.

```
// Set wifi channel, mac and phy
YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy;
phy.SetChannel (channel.Create ());

WifiMacHelper mac;
Ssid ssid = Ssid ("ns-3-ssid");

WifiHelper wifi;
```

Fonte: Produção original.

O próximo passo foi criar os NetDeviceContainer e configurar a camada MAC. Tanto os dispositivos quanto o servidor foram instalados na mesma camada PHY. Com ajuda do WifiHelper, é instalada a rede WiFi no servidor e em cada dispositivo. O código utilizado está na figura 14.

Figura 14 – Instalação do WiFi nos *nodes*.

```
// Install server (station node)
NetDeviceContainer serverDevice;
mac.SetType ("ns3::StaWifiMac",
             "Ssid", SsidValue (ssid),
             "ActiveProbing", BooleanValue (false));
serverDevice = wifi.Install (phy, mac, serverNode);

// Install wifi on devices (AP nodes)
NetDeviceContainer apDevices;
mac.SetType ("ns3::ApWifiMac",
             "Ssid", SsidValue (ssid));
apDevices.Add(wifi.Install (phy, mac, devices));
```

Fonte: Produção original.

Na figura 15 podemos ver a instalação de internet nos nodes, utilizando a classe InternetStackHelper. Ademais, foi utilizado IPv4, escolhido um endereço base e, com ajuda do Ipv4AddressHelper, atribuindo um endereço para cada dispositivo.

Figura 15 – Configuração da internet e do IPv4

```
InternetStackHelper stack;
stack.Install (serverNode);
stack.Install (devices);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer serverInterfaces;
serverInterfaces = address.Assign (serverDevice);
address.Assign (apDevices);
```

Fonte: Produção original.

Por fim, utilizamos um echo server em conjunto com echo clients. O echo server foi instalado no node referente ao servidor e um echo client foi instalado em cada node referente aos dispositivos IoT. Foi definido um máximo de 1 pacote por cliente com 1024 bytes cada, sendo enviado 1 por segundo, como visualizado na figura 16.

Figura 16 – Criação e atribuição do *echo client* e *echo server*.

```
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps =  echoServer.Install (serverNode.Get(0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (15.0));

UdpEchoClientHelper echoClient (serverInterfaces.GetAddress(0), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

Fonte: Produção original.

Para monitorar o atraso, foi usado o módulo "flow monitor"do NS-3, e com ajuda da classe AnimationInterface, foram gerados arquivos XML para visualização no NetAnim.

Na figura 17 há a representação da primeira simulação. Foram criados apenas dois dispositivos IoT, sendo eles representados pelos nodes 0 e 1. O servidor está representado pelo node 2. Podemos notar que o dispositivo 1 manda o seu pacote para todos os nodes em seu alcance, mesmo que eles não sejam o destino do pacote. A distância dos dispositivos 0 e 1 até o servidor é 35 e 25 metros, respectivamente. O atraso total para cada um dos dispositivos mandar 1 pacote e receber a resposta do servidor foi de 2,96206 milissegundos.

Para a segunda simulação, manteve-se os mesmos atributos, alterando apenas a posição dos dispositivos e do servidor (figura 18). A distância agora, entre dispositivos e servidor, é de, aproximadamente, 50 metros. Nesse cenário, o tempo de atraso total subiu para 3,03167 milissegundos.

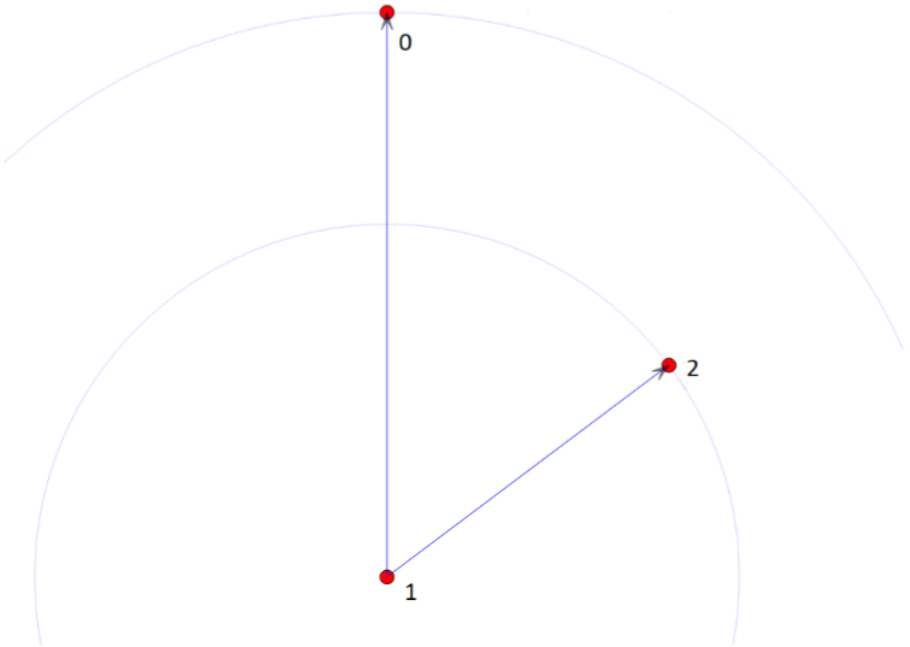
Na terceira simulação, mantivemos a distância de 50 metros entre dispositivos e servidor, porém alteramos a posição do dispositivo 1 de forma com que ele não estivesse no alcance do dispositivo 0 (figura 19). O tempo de atraso desceu para 3,02967 milissegundos.

Na quarta simulação, criamos 6 novos dispositivos, totalizando 8 dispositivos IoT. O servidor está representado pelo node no centro, número 8, enquanto os dispositivos foram representados pelos nodes de número 1 a 7. A maior distância entre dispositivo e servidor é 35 metros. Com o aumento no número de dispositivos, o atraso subiu para 9,14837 milissegundos. Por fim, montamos a tabela 2 com o resumo das quatro simulações.

Tabela 2 – Resumo das simulações

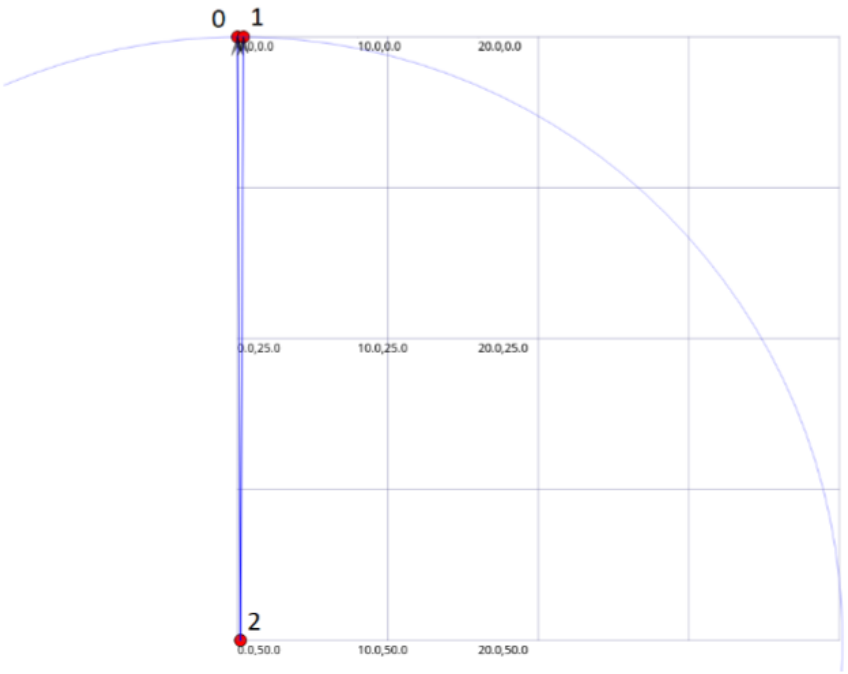
Simulação	Quantidade de dispositivos	Distância máxima entre servidor e dispositivo (em metros)	Dispositivos isolados dos outros?	Tempo de atraso (em milissegundos)
Nº 1	2	35	Não	2,96206
Nº 2	2	50	Não	3,03167
Nº 3	2	50	Sim	3,02967
Nº 4	8	35	Não	9,14837

Figura 17 – Primeira simulação.



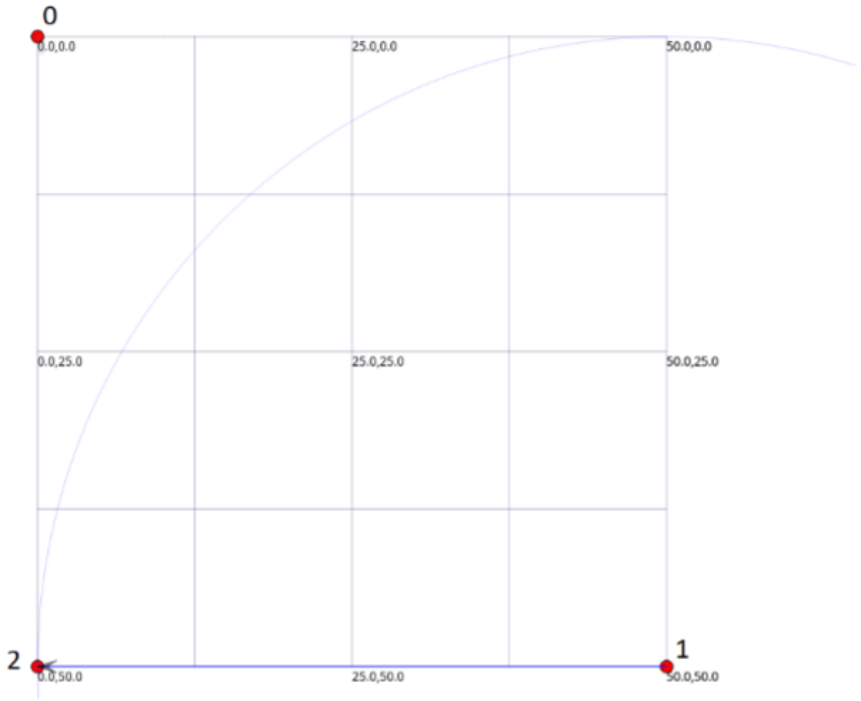
Fonte: Produção original.

Figura 18 – Segunda simulação.



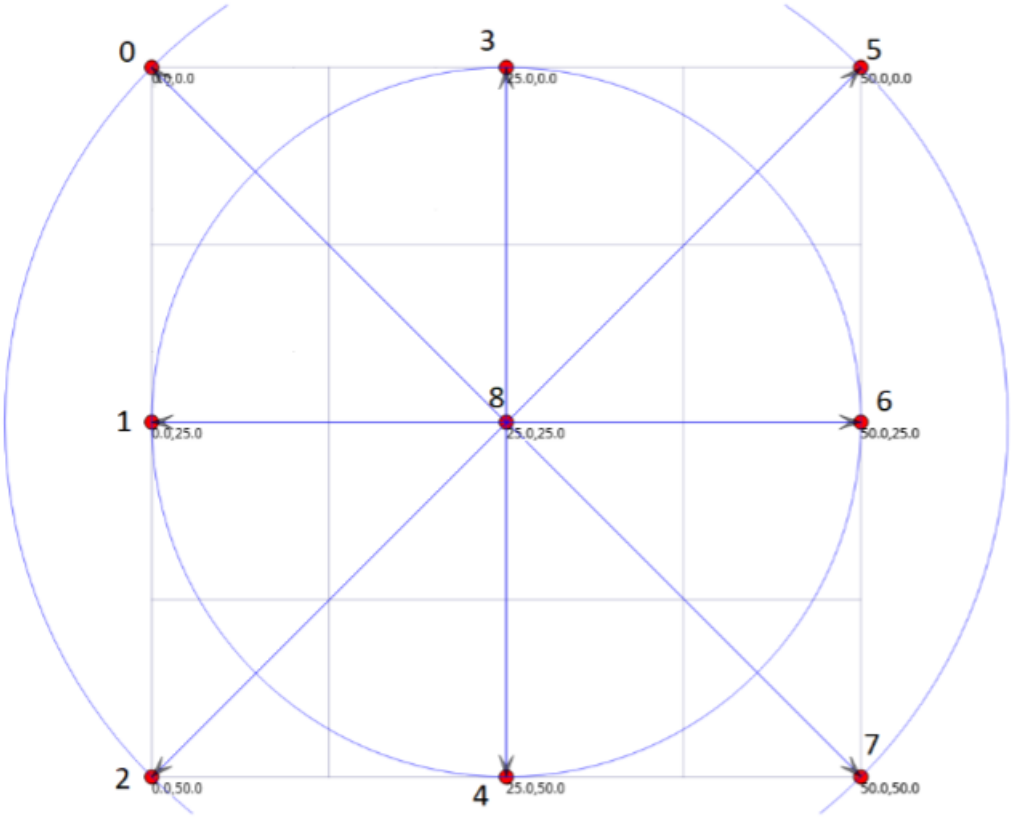
Fonte: Produção original.

Figura 19 – Terceira simulação.



Fonte: Produção original.

Figura 20 – Quarta simulação.



Fonte: Produção original.

8. CONCLUSÕES E TRABALHOS FUTUROS

A partir das simulações realizadas demonstraram alguns dos principais problemas que encontramos em redes de larga escala e a importância da utilização de simulações para explorar tal área. Comparando a primeira simulação com a segunda, vemos como o aumento da distância entre dispositivos e servidor impactam no tempo de resposta. Ao compararmos a segunda e a terceira simulação, podemos ver que a densidade de dispositivos também é um fator relevante, pois os dispositivos gastam tempo, mesmo que pequeno, interceptando pacotes o qual não são o destinatário. Por fim, quando comparamos as três primeiras simulações com a quarta, é perceptível que a quantidade de dispositivos na rede afeta drasticamente o tempo de resposta.

Problemas difíceis de resolver na prática podem ser estudados de maneira mais conveniente utilizando simulações. A falta de trabalhos na área, como mencionado na seção 3.1, mostra que há a necessidade de explorar mais essas ferramentas, possibilitando avanços no estado da arte. Por fim, para trabalhos futuros, seria interessante aprofundar os estudos dessas ferramentas para gerar simulações mais complexas e com cenários diferentes. Até mesmo reproduzir os sistemas dos trabalhos relacionados, seção 3.2 e 3.3, podendo comparar valores, verificar a escalabilidade e até mesmo juntar sistemas de trabalhos distintos.

REFERÊNCIAS

AAZAM, M. et al. Cloud of things (cot): Cloud-fog-iot task offloading for sustainable internet of things. **IEEE Transactions on Sustainable Computing**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 87–98, 2022. ISSN 23773782.

A. M. RAHMANI et al., ‘‘Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach,’’ Future Gener. Comput. Syst., vol. 78, pp. 641–658, Jan. 2018.

CARDOSO, J. V. et al. Dos attack detection and prevention in fog-based intelligent environments. **Brazilian Journal of Development**, Brazilian Journal of Development, v. 5, p.23934–23956, 2019. ISSN 25258761.

CHEGINI, H. et al. Process automation in an iot–fog–cloud ecosystem: A survey and taxonomy. **IoT**, MDPI AG, v. 2, p. 92–118, 2 2021.

CRUZ, R. N. S. et al. Experiments on energy optimization in smart residences. In: 2021 **International Conference on Software, Telecommunications and Computer Networks (SoftCOM)**. [S.l.: s.n.], 2021. p. 1–6. ISSN 1847-358X.

DEHRAJ, P.; SHARMA, A. A review on architecture and models for autonomic software systems. **Journal of Supercomputing**, Springer, v. 77, p. 388–417, 1 2021. ISSN 15730484.

FARIDI, F. et al. Cloud computing approaches in health care. In: . [S.l.]: Elsevier Ltd, 2021.v. 51, p. 1217–1223. ISSN 22147853.

JEURKAR, V. et al. Iot based water management system. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2020. p. 141–144. ISBN 9781728150031.

KODALI, R. K.; KIRTI, B. Ns-3 model of an iot network. In: 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA). [S.l.: s.n.], 2020. p. 699–702.

KöK İbrahim; ÖZDEMIR, S. Content-centric data and computation offloading in ai-supported fog networks for next generation iot. **Pervasive and Mobile Computing**, v. 85, p. 101654, 2022. ISSN 1574-1192. Disponível em: https://www.sciencedirect.com/science/article/pii/S157411922200075X.

L. NI, J. Zhang, C. Jiang, C. Yan, and K. Yu, ‘‘Resource allocation strategy in fog computing based on priced timed petri nets,’’ IEEE Internet Things J., vol. 4, no. 5, pp. 1216–1228, 2017

MEHTA, A.; ELMROTH, E. Distributed cost-optimized placement for latency-critical applications in heterogeneous environments. In: **2018 IEEE International Conference on Autonomic Computing (ICAC)**. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 121–130. ISBN 9781538651391. ISSN 2474-0756.

MOCRII, D.; CHEN, Y.; MUSILEK, P. Iot-based smart homes: A review of system architecture, software, communications, privacy and security. **Internet of Things**, v. 1-2, p. 81–98, 2018. ISSN 2542-6605. Disponível em: https://www.sciencedirect.com/science/article/pii/S2542660518300477.

NIKOLOPOULOS, V. et al. Fog node self-control middleware: Enhancing context awareness towards autonomous decision making in fog colonies. **Internet of Things**, v. 19, p. 100549, 2022. ISSN 2542-6605. Disponível em: https://www.sciencedirect.com/science/article/pii/S254266052200049X.

R. K. Naha et al., "Fog Computing: Survey of Trends, Architectures, Requirements, and Research Directions," in IEEE Access, vol. 6, pp. 47980-48009, 2018, doi: 10.1109/ACCESS.2018.2866491.

SAIF, M. A. N.; NIRANJAN, S. K.; AL-ARIKI, H. D. E. Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis. **Wireless Networks**, Springer, v. 27, p. 2829–2866, 5 2021. ISSN 15728196.

SAMPAIO, H. V. et al. Autonomic energy management with fog computing. **Computers and Electrical Engineering**, Elsevier Ltd, v. 93, 7 2021. ISSN 00457906.

TADAKAMALLA, U.; MENASCé, D. A. Autonomic resource management using analytic models for fog/cloud computing. In: **2019 IEEE International Conference on Fog Computing (ICFC)**. [S.l.: s.n.], 2019. p. 69–79.

X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, ‘‘Vehicular fog computing: A viewpoint of vehicles as the infrastructures,’’ IEEE Trans. Veh. Technol., vol. 65, no. 6, pp. 3860–3873, Jun. 2016.