

Uma solução em névoa via objetos inteligentes para lidar com a heterogeneidade dos dados em um ambiente residencial

Artur H. Brandão de Souza¹, Renato A. Nobre¹,
Vinícius P. Gonçalves¹, Geraldo P. Rocha Filho¹

¹Universidade de Brasília – UnB
Caixa Postal 4466 CEP 70910-900 – Brasília – DF – Brasil

arturhbsouza@gmail.com, rekanobre@gmail.com

{vpgvinicius, geraldof}@unb.br

Abstract. *This work proposes HOST – a solution that addresses the problems of data heterogeneity and the interoperability of smart objects in the context of a smart home. HOST was modeled to compose a set of intelligent objects to form a computational infrastructure in fog. To disseminate heterogeneous information, a Publish/Subscribe communication module was implemented to abstract the details of communication between objects. The results of the performance evaluation show evidence of efficiencies in the computational resources of the devices, as well as in the communication infrastructure of HOST.*

Resumo. *Este trabalho propõe o HOST – uma solução que trata dos problemas da heterogeneidade de dados e da interoperabilidade de objetos inteligentes no contexto de uma casa inteligente. O HOST foi modelado com o intuito de compor um conjunto de objetos inteligentes para formar uma infraestrutura computacional em névoa. Para disseminar as informações heterogêneas, implementou-se um módulo de comunicação Publish/Subscribe que permite abstrair os detalhes de comunicação entre os objetos. Os resultados da avaliação de desempenho mostram evidências de eficiências nos recursos computacionais dos dispositivos, bem como na infraestrutura de comunicação do HOST.*

1. Introdução

Nos últimos anos, a informação se tornou um dos elementos mais valiosos do mundo [Vitorino 2018]. Tais informações são provenientes de dados crus, observações documentadas, ou resultados de alguma medição [Becker 2015]. Há diversas maneiras de se obter dados, e uma delas é por meio dos objetos inteligentes. Esses objetos podem: (i) determinar o valor de algum evento local; (ii) detectar eventos de interesse; (iii) processar tais eventos dentro ou fora da rede; (iv) detectar um objeto de interesse; e (v) prover comunicação de dados e escalabilidade.

Com os avanços tecnológicos na área de micro sistemas eletro-mecânicos, houve um crescimento significativo no uso dos objetos inteligentes dentro do contexto residencial [Gonçalves et al. 2021]. Nesse sentido, as casas inteligentes, também conhecida como *smart home*, vêm surgindo como fonte de informação promissora e despertam o interesse de pesquisadores em propor modelos e aplicações para realizar

tomada de decisões inteligentes. Uma casa inteligente pode ser definida como um conjunto de objetos inteligentes presentes em uma residência para automatizar tarefas diárias e manter um controle sobre dados que o ambiente consegue prover para o usuário [Jiang et al. 2018, Filho et al. 2019].

Vale salientar que devido a escassez de recursos computacionais presentes nos objetos inteligentes, desenvolver serviços no contexto residencial que utilizam a intercomunicação entre tais objetos não é uma tarefa trivial. Essa limitação se torna ainda mais desafiadora de ser resolvida devido a heterogeneidade dos dados e interoperabilidade entre os objetos presentes na residência. Para lidar com tais desafios, um dos caminhos promissores é o uso da computação em névoa [Yi et al. 2015] aliada com o paradigma de comunicação *Publish/Subscribe* (Pub/Sub) [Eugster et al. 2003]. A computação em névoa traz o processamento e armazenamento dos dados para a borda da rede, ou seja, essas tarefas são executadas de maneira cooperativa entre os objetos inteligentes contidos no ambiente por meio da comunicação [Yi et al. 2015]. O paradigma Pub/Sub permite a comunicação confiável entre os objetos de maneira assíncrona a partir de múltiplas fontes de eventos para seus respectivos grupos de interesse.

Diferentes abordagens foram propostas para lidar com o problema da heterogeneidade dos dados e a interoperabilidade entre os dispositivos em um mesmo ambiente [Kumar 2014, Jutadhamakorn et al. 2017, Kang et al. 2017, de Sousa et al. 2018, Rocha Filho et al. 2018b, Rocha Filho et al. 2020]. Apesar dos esforços significativos nessa vertente, investigar a interoperabilidade de comunicação entre os dispositivos, aproveitando os conteúdos que são disseminados de maneira implícita em um contexto residencial com base na computação em névoa é um campo de pesquisa em aberto que este trabalho investiga.

Com isso em mente, este artigo apresenta HOST, uma solução para lidar com o problema da Heterogeneidade dos dados e com a interoperabilidade dos objetos inteligentes no contexto de uma casa inteligente. Para isso, modelou-se um ambiente computacional em névoa por meio dos objetos inteligentes providos da residência para formar a infraestrutura do HOST. Para disseminar as informações, implementou-se nessa infraestrutura um módulo de comunicação Pub/Sub com o intuito de melhorar o custo computacional e operacional procedente da variação da quantidade de nós da névoa e da quantidade de dados trocados entre os objetos. Os resultados evidenciam que o HOST possui um desempenho satisfatório dos recursos computacionais dos dispositivos e da infraestrutura da solução proposta.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta os trabalhos relacionados. Seção 3 descreve o funcionamento do HOST e a sua validação é apresentada na Seção 4. Por fim, a Seção 5 apresenta as conclusões e trabalhos futuros.

2. Trabalhos relacionados

Dentro dos trabalhos analisados, Sousa *et al.* [de Sousa et al. 2018] propuseram uma plataforma para lidar com grandes volumes de dados heterogêneos disseminados por dispositivos inteligentes com base no *Apache Kafka*. Para tanto, foi desenvolvida uma interface *webservice* REST para abstrair o uso da plataforma com os dispositivos inteligentes. Ademais, Kang *et al.* [Kang et al. 2017] construíram um MQTT *broker* no *Amazon Web Services* para fornecer serviços para aplicações de IoT (*Internet of Things*). Como

prova de conceito, os autores modelaram um cenário de uma casa inteligente ao fazer o uso de um Arduíno para capturar os dados do ambiente e validar o funcionamento do sistema. Visando a escalabilidade e o baixo custo através da grande quantidade de dispositivos IoTs que podem ser conectados em um ambiente inteligente, Jutadhamakorn *et al.* [Jutadhamakorn et al. 2017] propuseram um sistema de *cluster* com base no MQTT *broker*. Para tanto, foi modelado um sistema de visualização com a ideia de analisar os dados do cluster quanto ao uso da cpu, memória e o tráfego de rede. Salienta-se que as soluções citadas anteriormente, limitam-se por proporem maneiras de enviar os dados diretamente para a nuvem, não expandido o estudo para a comunicação em névoa, como explorado nesta pesquisa. Além disso, tais soluções não aproveitam o processamento local que os dispositivos podem oferecer.

Com a proposta de gerenciar aplicações de uma residência por meio da computação em névoa, Rocha Filho *et al.* [Rocha Filho et al. 2018a] propõem um sistema de controle neuro-fog via objetos inteligentes. O trabalho tem como finalidade reduzir o tempo da tomada de decisão ao executar seus processos decisórios na infraestrutura da residência. Ainda com foco na computação em névoa, Rocha Filho *et al.* [Rocha Filho et al. 2018b] propõem um sistema de automação residencial de baixo custo para melhorar o processo de tomada de decisão do morador. Para tanto, o sistema proposto possui como objetivo o aprendizado das aplicações por meio de um comitê de classificadores. Já Kumar *et al.* [Kumar 2014] propõem um sistema de baixo custo e flexível no contexto de casa inteligente. Os autores apresentam uma aplicação para Android que comunica com um *micro-web server*, provendo funcionalidades de interação entre os dispositivos utilizando *Arduino Ethernet* como forma de conectar sensores. Dessa maneira, a solução apresentada por esta pesquisa é uma extensão de todos os trabalhos anteriores, diferenciando-a devido a interface de comunicação para lidar com o problema da heterogeneidade dos dados e com a interoperabilidade dos dispositivos em uma casa inteligente.

3. Uma solução em névoa para uma casa inteligente

Esta seção apresenta o HOST¹, uma solução em névoa para lidar com o problema da heterogeneidade dos dados e com a interoperabilidade dos objetos inteligentes no contexto residencial. O HOST foi desenvolvido com o intuito de ter um conjunto de objetos inteligentes providos pela própria residência, os quais estão conectados à internet para formar um ambiente computacional em névoa. Esse ambiente, também chamado de computação em borda, gera uma cooperação entre o uso da internet e os dispositivos heterogêneos, que trabalham juntos para prover uma melhora na performance de armazenamento e no processamento de tarefas por estarem na camada de borda da rede [Yi et al. 2015, Fernando et al. 2013].

A partir desse espaço formado no HOST, é implementado um módulo que abstrai os detalhes de comunicação dos objetos inteligentes com base no módulo de comunicação Pub/Sub. Devido ao desacoplamento oferecido entre as partes finais por meio do Pub/Sub, é possível criar grupos de interesses para cada dado coletado. Nesse sentido, é factível garantir que os módulos de ambiente, objetos e aplicações possam se comunicar entre si, auxiliando no processo de disseminação de dados, como apresentado na Figura 1a.

¹Disponível em <https://github.com/arturhbs/HOST>

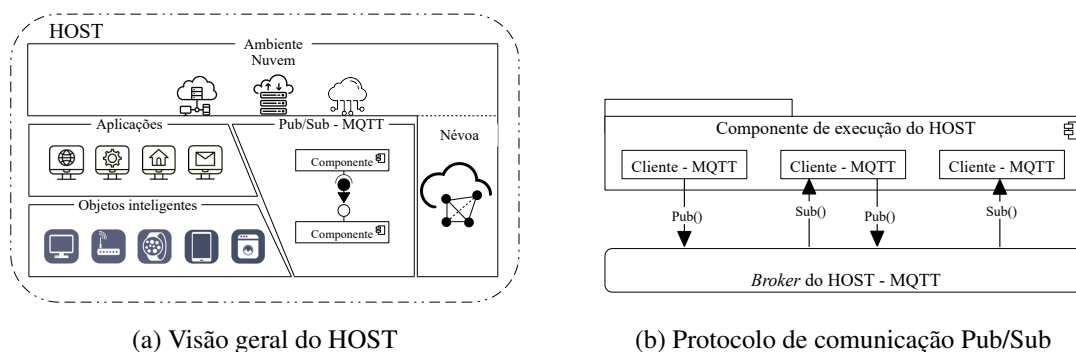


Figura 1. Cenário de funcionamento do HOST

O HOST contém objetos inteligentes que produzem dados e os enviam para o módulo de ambiente através da conexão MQTT Pub/Sub, como apresentado na Figura 1a. Além disso, as aplicações atuam em cima desses dados coletados no ambiente. Para tanto, a infraestrutura de funcionamento do ambiente do HOST foi dividida em duas, sendo elas: (i) névoa; e (ii) nuvem. No ambiente em névoa, o armazenamento e o processamento das tarefas são realizadas localmente por meio dos nós da névoa. Por outro lado, o ambiente em nuvem é um servidor remoto com maior recurso computacional que, por exemplo, armazena dados históricos e processa tarefas que requerem um alto custo tecnológico. Em síntese, o HOST tem como objetivo principal garantir eficiência na comunicação para a transferência de dados entre dispositivos em um mesmo ambiente, ao mesmo tempo em que consegue prover eficiência nos recursos computacionais limitados dos objetos sem que haja problemas de escalabilidade nos processos executados.

3.1. Ambiente computacional em névoa

No ambiente computacional em névoa, assume-se que cada objeto inteligente é equipado com capacidade de armazenamento, poder de processamento e interface de comunicação sem fio. No HOST, os objetos inteligentes são chamados de nós de névoa que são responsáveis por coletar, processar e disseminar os dados no ambiente residencial. Ainda, esses objetos estão na borda da rede e são responsáveis por realizar uma comunicação entre os usuários e o servidor em nuvem. O objetivo do conjunto dos nós de névoa é formar um ambiente computacional capaz de realizar tomada de decisões em uma área que necessita enviar dados com uma performance superior a que um ambiente em nuvem consegue alcançar. Salienta-se que não é o escopo deste trabalho propor modelos e aplicações para realizar tomada de decisões, mas propor uma solução que possa ser utilizada para isso.

Os nós de névoa estão distribuídos no ambiente e atuam como produtor e consumidor de informações. Cada equipamento inteligente possui o conhecimento local e global do ambiente. Para isso, o HOST adota um mapa virtual de névoa com base no trabalho [Rocha Filho et al. 2018a]. Para gerar o mapa, o nó de névoa compartilha a informação mediante um modelo *request-reply* do Pub/Sub. O compartilhamento é feito via um mecanismo de comunicação implementado no HOST que será apresentado a seguir.

3.2. Mecanismo de comunicação

A solução proposta foi desenvolvida com base no MQTT (*Message Queue Telemetry Transport*) [Liu et al. 2003] no ambiente em névoa. A escolha pelo MQTT é condicionada devido a mínima utilização da largura de banda, da baixa imposição de recursos

sobre equipamentos que ainda assim conseguem transmitir com uma confiabilidade e garantia de entrega das mensagens enviadas sobre ele, além do desacoplamento entre os componentes que fazem parte do ambiente, características presentes nesta pesquisa. Já o Publicador e o Subscritor são dispositivos que enviam e recebem mensagens, respectivamente, de maneira desacoplada e assíncrona entre os dispositivos [Liu et al. 2003]. Assim, há três componentes principais no HOST atuando no processo do MQTT, Figura 1b, sendo eles: (i) publicador, componente que envia dados por meio de um evento; (ii) evento, componente que gera uma notificação; e (iii) subscritor, componente que recebe dados de interesse sobre um evento.

No HOST, o desacoplamento que há entre os objetos é uma dissociação de espaço, tempo e de sincronização. Quanto ao espaço, é relacionado com a não necessidade dos nós saberem sobre a quantidade de publicadores e subscritores pertencentes à névoa, uma vez que há uma independência na atividade de cada equipamento que faz o uso do MQTT. Além disso, não há necessidade dos publicadores e subscritores estarem ativos ao mesmo tempo para que haja uma troca de mensagens entre si. Com isso, ao publicar um evento, mesmo que o subscritor esteja inativo no momento que é gerado a notificação, este poderá ser acionado e não há necessidade do publicador estar ativo para que o evento seja direcionado a seu subscritor, uma vez que quem faz essa função é o *broker* do MQTT no HOST. Além disso, o HOST por meio do MQTT traz aos publicadores a possibilidade de produzirem eventos sem a necessidade de que alguns deles sejam bloqueados ou impedidos de enviar mensagens. De maneira semelhante ocorre com os subscritores, quanto ao recebimento desses eventos, em que, o *broker* está encarregado de garantir a remessa das mensagens para os respectivos subscritores. Em razão disso, a solução descrita consegue trazer facilidades através da execução assíncrona de suas atividades, o que possibilita realizar tarefas concorrentes entre todos os nós de névoa presentes no ambiente para lidar com o problema da heterogeneidade dos dados e interoperabilidade entres os nós.

Para o manuseio dos eventos, o HOST utiliza o *broker* da biblioteca *Eclipse Paho MQTT*², que constitui em um componente do protocolo MQTT, Figura 1b. A biblioteca trabalha na captura dos dados enviados pelos publicadores, gera uma notificação e a direciona para seus respectivos subscritores. Para que isso ocorra, tanto os publicadores quanto os subscritores utilizam os comandos *publish()* e *subscribe()*, respectivamente, ao importar a biblioteca para que consigam se comunicar com o *broker*.

4. Avaliação de desempenho

O HOST foi validado em duas etapas, sendo elas: (i) avaliação dos recursos computacionais da infraestrutura; e (ii) avaliação da disseminação dos dados na rede. Por meio dessas avaliações, foi possível entender o funcionamento do HOST e a sua vantagem em um ambiente em névoa. A seguir, será apresentado o cenário modelado, os parâmetros selecionados e as métricas utilizadas para gerar os resultados.

4.1. Cenário

Para avaliar a proposta, utilizou-se o contêiner do *docker* como um ambiente para simular os dispositivos *Pub/Sub* em que tais dispositivos representam os nós de névoa no HOST. A linguagem utilizada foi o Python 3.6 com a biblioteca *paho-mqtt*³ para atuar como *broker*.

²<https://www.eclipse.org/paho/>

³<https://pypi.org/project/paho-mqtt/>

Como métricas utilizadas para avaliar os recursos computacionais do HOST e computar a disseminação dos dados na rede, destacam-se: (i) **uso do disco** pelo sistema operacional; (ii) **informações da memória**, que consistem na soma da memória física não trocada com o total de memória virtual utilizada pelo processo; (iii) **tempo gasto da cpu** que representa a soma do tempo do usuário com o tempo em modo *kernel* pelo processo; (iv) **tempo de resposta** para enviar um tópico; (v) **taxa de dados** entre o dispositivo e infraestrutura; e (vi) **taxa de entrega** por segundo dos tópicos na infraestrutura. Para obter tais métricas, foi utilizada a biblioteca *psutils*⁴ no *Python* que tem como objetivo coletar informações dos processos e do próprio sistema operacional.

Na geração dos resultados, foi realizada uma variação do número de nós de névoa (5, 10, 20 e 30), e uma variação do número de tópicos (8, 13, 21 e 34) para investigar o impacto no HOST. Neste caso, os nós de névoa, publicadores, produziram simultaneamente seus próprios dados, enquanto um subscritor recebe por meio do *broker*. Ainda, variou-se o *payload* dos tópicos (100kB, 200kB, 400kB, 800kB e 1600kB) instanciando um *broker* e os dados foram normalizados quanto ao tempo de resposta (ms) e tráfego de dados (kB/s). Cada experimento foi executado 32 vezes com um intervalo de confiança de 95% usando a distribuição *t-student*, como apresentado nas subseções a seguir.

4.2. Impacto dos recursos computacionais dos dispositivos no HOST

Na Figura 2, é apresentada a avaliação de desempenho dos recursos computacionais do HOST variando a quantidade de tópicos. Inicialmente, foi avaliado o uso do disco do sistema operacional em função do CDF (*Cumulative Distribution Function*), como apresentado na Figura 2a. Independentemente da quantidade de tópicos enviados, observa-se que o uso do disco possui estabilidade no seu comportamento, além de uma escalabilidade no seu uso à medida que aumenta a quantidade de recursos utilizados no HOST. Isso faz sentido uma vez que a solução proposta aproveita o processamento local provido do ambiente computacional em névoa. Assim, é possível aumentar a quantidade de tópicos sem se preocupar em um crescimento do uso do disco que pudesse afetar o publicador.

Após avaliar o uso do disco, analisou-se o tempo gasto de cpu para executar o processo, como apresentado na Figura 2b. Por meio dos resultados, observa-se que durante todo o processo há um crescimento suave em função do tempo de execução do usuário. Esse comportamento pode ser explicado devido ao pub/sub não enviar os tópicos diretamente para os dispositivos. Isto é, apenas os subscritores que manifestam interesse podem receber o tópico enviado. Conseqüentemente, não haverá um crescimento inesperado do uso da cpu na solução proposta.

Em relação a métrica de memória, nota-se que não há uma diferença no comportamento da métrica independentemente da quantidade de tópicos, como apresentado na Figura 2c. Em outras palavras, a memória física e virtual utilizada pelo processo permanece inalterada independente do aumento de tópicos enviados. Portanto, a quantidade de memória a ser utilizada para o publicador pode ser próxima a constante que ela atinge durante sua execução, não necessitando de buscar alternativas caso venha a necessitar alterar a quantidade de envio de tópicos.

⁴<https://pypi.org/project/psutil/>

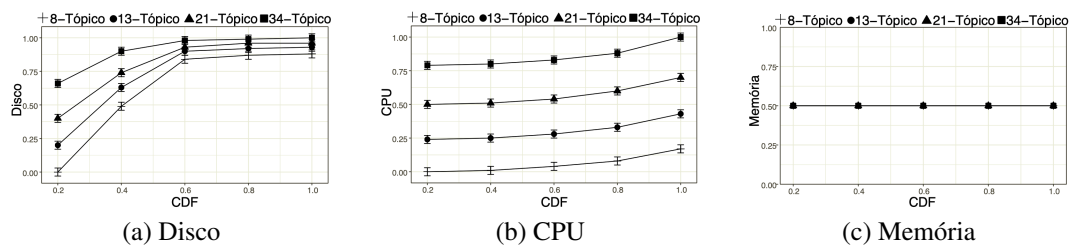


Figura 2. Impacto de desempenho do uso do disco, tempo gasto de CPU e memória.

4.3. Impacto dos recursos computacionais da infraestrutura do HOST

Avaliamos o desempenho do HOST quanto ao aumento da quantidade de nós de névoa em função do *payload* enviados por eles. Conforme a quantidade de *payload* aumenta, observa-se que há um aumento no tempo de resposta para enviar a mensagem, como apresentado na Figura 3a. Isso está relacionado com o *broker*, o qual necessita de um tempo maior de retorno devido a captação e envio das mensagens.

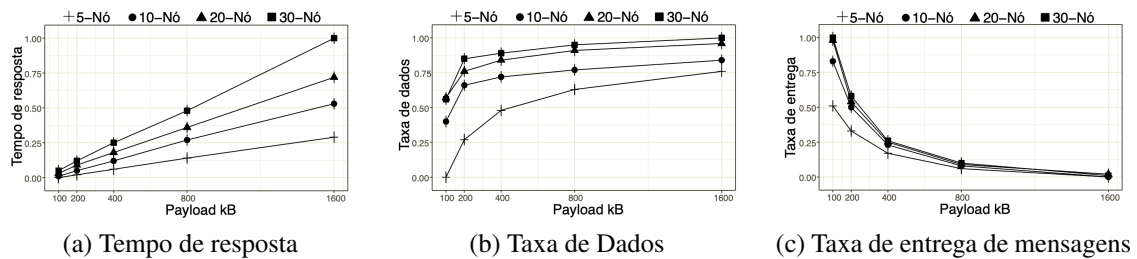


Figura 3. Impacto do tempo de resposta, taxa de dados e entrega de mensagens.

Quanto à taxa de dados, analisada na Figura 3b, nota-se uma estabilidade no crescimento após 400 kB de *payload*. Esse fato ratifica que é possível aumentar a quantidade de dispositivos sem sobrecarregar a infraestrutura do HOST. Além disso, a avaliação corrobora o fato da não utilização de memória pelo *broker*, como apresentado na Figura 2c e, assim, não gerando um obstáculo para o tráfego de dados que ocorre durante o processo. Vale salientar que com o aumento da quantidade de dispositivos em função do *payload*, a taxa de entrega de mensagens diminui, como observado na Figura 3c. Esse resultado advém da velocidade e quantidade de dados que o *broker* consegue recebê-los e repassar para os demais subscritos nos tópicos em questão. Contudo, mesmo que a taxa de entrega diminua conforme o *payload* aumenta, é notório que a quantidade de nós não afeta de modo significativo o resultado, chegando a uma estabilidade da taxa de entrega. Portanto, o HOST além de ser eficiente quanto a taxa de dados, também demonstra bons resultados quanto ao tempo de resposta e entrega de mensagens com o aumento de *payload*.

5. Conclusão

Este artigo propôs o HOST, uma solução para lidar com o problema da heterogeneidade dos dados e com a interoperabilidade dos objetos inteligentes no contexto residencial. A avaliação de desempenho apresenta evidências da eficiência nos recursos computacionais dos dispositivos e da infraestrutura de comunicação do HOST. Os resultados mostraram que o HOST proporciona escalabilidade no que diz respeito à quantidade de dispositivos atuando simultaneamente, além de demonstrar estar hábil a funcionar com diferentes tipos de dispositivos e dados. Como trabalhos futuros planeja-se, além de expandir os estudos deste artigo, aplicar o HOST em diferentes contextos voltados para ambientes inteligentes.

Referências

- Becker, J. L. (2015). Estatística básica: transformando dados em informação. Bookman.
- de Sousa, A. B., Neto, J. R. T., Filho, G. P. R., and Ueyama, J. (2018). Uma plataforma de iot para integração de dispositivos baseada em nuvem com apache kafka. In Anais Estendidos do XXXVI SBRC, Porto Alegre, RS, Brasil. SBC.
- Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A.-M. (2003). The many faces of publish/subscribe. ACM computing surveys (CSUR), 35(2):114–131.
- Fernando, N., Loke, S. W., and Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29:84–106. Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.
- Filho, G. P., Villas, L. A., Gonçalves, V. P., Pessin, G., Loureiro, A. A., and Ueyama, J. (2019). Energy-efficient smart home systems: Infrastructure and decision-making process. Internet of Things, 5:153–167.
- Gonçalves, V. P., Geraldo Filho, P., Mano, L. Y., and Bonacin, R. (2021). Flexpersonas: flexible design of iot-based home healthcare systems targeted at the older adults. AI & SOCIETY, pages 1–19.
- Jiang, H., Cai, C., Ma, X., Yang, Y., and Liu, J. (2018). Smart home based on wifi sensing: A survey. IEEE Access, 6:13317–13325.
- Jutadhamakorn, P., Pillavas, T., Visoottiviset, V., Takano, R., Haga, J., and Kobayashi, D. (2017). A scalable and low-cost mqtt broker clustering system. In 2017 2nd International Conference on Information Technology (INCIT), pages 1–5. IEEE.
- Kang, D.-H., Park, M.-S., Kim, H.-S., Kim, D.-y., Kim, S.-H., Son, H.-J., and Lee, S.-G. (2017). Room temperature control and fire alarm/suppression iot service using mqtt on aws. In 2017 International Conference on Platform Technology and Service. IEEE.
- Kumar, S. (2014). Ubiquitous smart home system using android application. arXiv preprint arXiv:1402.2114.
- Liu, Y., Plale, B., et al. (2003). Survey of publish subscribe event systems. Computer Science Dept, Indian University, 16.
- Rocha Filho, G., Neto, J. R. T., Valejo, A., Meneguette, R. I., Villas, L. A., and Ueyama, J. (2018a). Um sistema de controle neuro-fog para infraestruturas residenciais via objetos inteligentes. In Anais do XXXVI SBRC. SBC.
- Rocha Filho, G., Yukio Mano, L., Demetrius Baria Valejo, A., Aparecido Villas, L., and Ueyama, J. (2018b). A low-cost smart home automation to enhance decision-making based on fog computing and computational intelligence. IEEE Latin America Transactions, 16(1):186–191.
- Rocha Filho, G. P., Meneguette, R. I., Maia, G., Pessin, G., Gonçalves, V. P., Weigang, L., Ueyama, J., and Villas, L. A. (2020). A fog-enabled smart home solution for decision-making using smart objects. Future Generation Computer Systems, 103:18–27.
- Vitorino, M.S.A., M. L. C. H. (2018). Sociedade da informação, quando os dados pessoais viram moeda de troca: A polêmica da nova moeda.
- Yi, S., Li, C., and Li, Q. (2015). A survey of fog computing: Concepts, applications and issues. In Proceedings of the 2015 Workshop on Mobile Big Data, Mobidata '15, page 37–42, New York, NY, USA. Association for Computing Machinery.