

STEER: Redes IoT Dirigidas por Intenções e Baseadas em SDN

Bruna M. O. S. Cordeiro, Roberto Rodrigues Filho, Iwens G. S. Júnior, Fábio M. Costa

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 74690-900 - Goiânia - GO - Brasil

brunamichelly@discente.ufg.br, {robertovito,iwens}@ufg.br, fmc@inf.ufg.br

Abstract. *IoT infrastructures are becoming increasingly more difficult to manage. One of the main issues is the high volatility present in the infrastructure, which increasingly demands self-adaptive solutions. As a response, this work presents STEER (Sdn-based inTEnt drivEn iot netwoRks), a new approach for the dynamic adaptation of IoT networks, based on the unification of Intent-Driven Networks (IDN) and Software-Defined Networks (SDN). Particularly, we explore the ability of IDNs to interpret an application's intent, using an IDN-based mediator attached to SDN-controllers to autonomously change the IoT network behavior at runtime, thus realizing the intent according to the current operating context of the network.*

Resumo. *As infraestruturas de IoT estão ficando cada vez mais difíceis de gerenciar. Um dos principais problemas é a alta volatilidade presente na infraestrutura, que demanda, cada vez mais, soluções auto-adaptativas. Como resposta, este trabalho apresenta STEER (Sdn-based inTEnt drivEn iot netwoRks), uma nova abordagem para adaptação dinâmica de redes IoT, baseada na união dos conceitos de Redes Dirigidas por Intenções (IDN) e Redes Definidas por Software (SDNs). Especificamente, exploramos a capacidade de IDNs para interpretar a intenção de uma aplicação, usando um Mediador de IDN em conjunto com controladores SDN para modificar o comportamento da rede, em tempo de execução e de forma autônoma, permitindo realizar a intenção de acordo com o contexto operacional atual da rede.*

1. Introdução

As infraestruturas de IoT estão ficando cada vez mais complexas e difíceis de gerenciar. Um dos principais motivos é a alta volatilidade de sistemas distribuídos contemporâneos [Blair 2018], causada, notadamente, pelas mudanças constantes nas infraestruturas de rede. Assim, para se manterem funcionais e atenderem as demandas das aplicações, as redes IoT precisam se adaptar em tempo de execução, levando em conta tanto os objetivos das aplicações ao interagirem com a rede, quanto às suas próprias restrições, à medida que mudanças ocorrem em seu ambiente operacional. É notória a necessidade de abordagens autônomas que permitem às redes identificarem a necessidade de mudança e se adaptarem para uma melhor configuração, de forma rápida e assertiva, sem depender da análise manual e a tomada de decisão de um operador humano [Júnior et al. 2021].

Nesse contexto, redes auto-adaptativas, como *Self-driving Networks* [Mai et al. 2021], têm ganhado popularidade. O foco na realização dessas redes se

dá, majoritariamente, em mecanismos que permitem a adaptação da rede em tempo de execução e sem interrupção na prestação de serviços, assim como em técnicas de classificação do contexto de execução da rede e em técnicas de aprendizado de máquina para prever mudanças e adaptar a rede de acordo. Um aspecto importante e pouco explorado no processo de tomada de decisão, no entanto, refere-se aos objetivos das aplicações que interagem com a rede.

Por meio da junção dos conceitos de Redes Definidas por Software (SDN) e Redes Dirigidas por Intenções (IDN), este trabalho¹ tem por objetivo a proposta de uma abordagem inédita para redes IoT, capaz de auto-adaptar a rede em tempo de execução sem o uso de regras predefinidas, levando em consideração apenas o contexto operacional da rede e as metas alto-nível definidas pelas aplicações. Exploramos SDN como o mecanismo de adaptação do comportamento da rede, assim como IDN para viabilizar a interpretação das metas definidas pelas aplicações usando uma linguagem declarativa, com a finalidade de nortear o processo adaptativo da rede. Para realizar o conceito de IDN, nos baseamos na figura do **Mediador**, elemento que interage com os controladores SDN e que tem a função de receber e interpretar as intenções das aplicações e, em tempo de execução, decidir pelo comportamento que será executado na rede.

Este artigo está organizado da seguinte forma: os trabalhos relacionados são discutidos na Seção 2; a proposta de redes IoT baseadas em SDN e dirigidas por intenções, juntamente com sua arquitetura, é descrita na Seção 3; os resultados obtidos por meio de experimentos em um ambiente de rede simulado são apresentados na Seção 4; as conclusões e trabalhos futuros são discutidos na Seção 5.

2. Trabalhos Relacionados

Esta seção apresenta trabalhos relevantes da literatura nas seguintes vertentes, contrastando-os com o presente trabalho: mecanismos de adaptação em redes IoT; gerenciamento autônomo de redes IoT; e redes dirigidas por intenções (IDN).

A adaptação de uma rede IoT pode se dar em diferentes níveis: no nível do sistema operacional que roda nos dispositivos da rede [Rodriguez-Zurrunero et al. 2018]; no nível do protocolo de comunicação [Junior et al. 2020]; no nível da aplicação que roda nos dispositivos [Shafi et al. 2012, Aschenbruck et al. 2012]. Há também abordagens que permitem a (re)programação do comportamento da rede como um todo [Noor et al. 2019, Azzara et al. 2014]. Neste contexto, uma abordagem que vem se destacando é o uso de SDN para (re)configurar funções de rede (como roteamento) nos nós que compõem uma rede IoT [Galluccio et al. 2015, Bera et al. 2017]. No presente trabalho, de forma semelhante, fazemos uso de controladores SDN para (re)configurar as aplicações que executam nos nós da rede.

O gerenciamento autônomo de redes IoT, por sua vez, também tem ganhando destaque na literatura. Devido à volatilidade da infraestrutura de IoT, torna-se necessária a adaptação da rede de forma rápida e assertiva, sem depender da ação de gerentes de redes humanos. Dentro desta vertente, as *Self-driving Networks* [Júnior et al. 2021, Mai et al. 2021, Jacobs et al. 2018] têm ganhado grande notoriedade. Um elemento importante dessas abordagens refere-se ao uso de técnicas de

¹Disponibilizamos o código e o dataset para replicação dos experimentos mostrados neste artigo: <https://github.com/brunacordeiro/steer>

aprendizado de máquina [Osman et al. 2021] e de intenções definidas pelo gerente da rede seguindo o conceito de *Intent-Driven Networks* [Davoli et al. 2019]. O presente trabalho, contudo, emprega um algoritmo de tomada de decisão que age em tempo de execução à medida que coleta métricas da rede em tempo real. Esse algoritmo é inspirado na abordagem de sistemas emergentes descrita em [Rodrigues-Filho and Porter 2017].

Por sua vez, a definição de intenções para dirigir a adaptação em redes tem sido utilizada de diversas formas [Cerroni et al. 2017, Pang et al. 2020]. Uma decisão de projeto importante refere-se ao ator que define as intenções no sistema, que pode ser o gerente de rede ou o desenvolvedor de aplicações, assim como o escopo de uma intenção. Neste trabalho, seguimos a definição de IDN proposta em [Elkhatib et al. 2017], na qual intenções são definidas pelo desenvolvedor de aplicações, com o objetivo de informar a rede sobre como uma dada aplicação pretende usá-la, permitindo que a rede se adapte para melhor acomodar as demandas específicas de cada aplicação. Essa abordagem contrasta com outras propostas semelhantes, em que intenções são definidas pelo gerente de rede e especificam políticas de alto-nível que ditam como a rede deve se comportar com respeito a todas as aplicações em execução [Jacobs et al. 2018].

3. Redes IoT Baseadas em SDN e Dirigidas por Intenções - STEER

Uma rede IoT definida por software e dirigida por intenção é caracterizada, principalmente, pela auto adaptação dirigida por metas alto-nível (i.e., intenções) providas e definidas pelas aplicações que interagem com a rede. As funcionalidades da rede são definidas na forma de um conjunto predefinido de *comportamentos*, podendo haver vários comportamentos alternativos para a mesma funcionalidade. Os requisitos e objetivos das aplicações com respeito a uma funcionalidade específica são expressos na forma de *intenções* e traduzidos transparentemente (tanto para as aplicações quanto para a rede) em termos do comportamento mais apropriado em função do contexto operacional da rede, permitindo a adaptação da funcionalidade da rede. O gerente de rede é responsável por definir e implementar o conjunto de comportamentos disponíveis, enquanto a especificação das intenções é realizada pelo desenvolvedor de aplicações para o usuário final, fazendo com que não haja a necessidade do desenvolvedor em conhecer a configuração da rede.

A Figura 1 apresenta a arquitetura STEER, composta por três camadas: uma solução já existente de SDN para IoT dividida em duas subcamadas, a subcamada de infraestrutura e a subcamada de controle; uma implementação do conceito de IDN para IoT, composta por um *Gerenciador*, um *Mediador* e um componente que realiza a *Comunicação* entre as camadas da IDN e SDN; e a camada de Aplicação composta pelas aplicações, que representam a diversidade de sistemas IoT, juntamente com as respectivas intenções e comportamentos.

Na camada SDN para IoT, utilizamos o SDN-WISE [Galluccio et al. 2015], cujo plano de controle permite a instalação e adaptação de funções de aplicação no plano de dados dos nós de uma rede IoT. Esta escolha se deu também pelo fato de o SDN-WISE ser de código aberto, o que permitiu sua adaptação para o propósito deste trabalho, como descrito a seguir. A subcamada de infraestrutura é composta pelos nós da rede IoT, que podem ser de dois tipos: nós sensores, que fazem a leitura dos dados no ambiente; e o nó *sink*, responsável por enviar e receber as solicitações do controlador SDN. O SDN-WISE foi alterado para adição de um agente nos nós da rede, o qual realiza a mudança

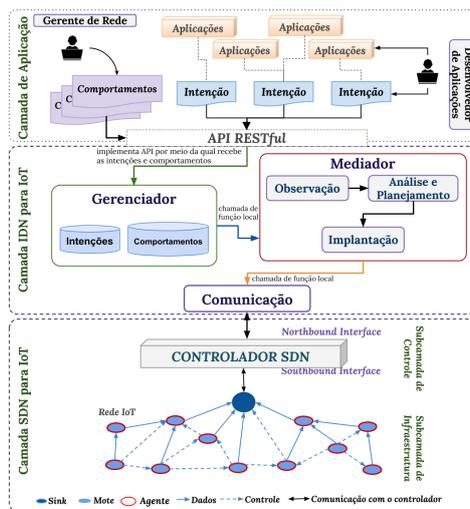


Figura 1. Arquitetura STEER

dinâmica do comportamento da rede, no nível de aplicação, em cada nó. Na subcamada de controle, por sua vez, o controlador SDN realiza o gerenciamento da rede por meio de pacotes de controle, os quais são enviados aos nós por meio da interface *southbound*, enquanto a interface *northbound* fornece um conjunto de operações que permitem à camada de IDN enviar os comandos resultantes da interpretação da intenção, por meio de pacotes de dados, para serem executados na rede pelo controlador SDN. O plano de controle permite que as mensagens de configuração, produzidas pelo Mediador, contendo as funcionalidades dos comportamentos, possam ser implantadas nos *notes*.

Na camada de IDN para IoT, que representa a principal contribuição deste trabalho, o *Gerenciador* é o módulo responsável por receber da *API RESTful* as intenções e comportamentos, que ficam armazenados em repositórios, e por encaminhá-los ao Mediador. Este, por sua vez, executa as etapas de *Observação*, *Análise e Planejamento* e *Implantação* de cada comportamento na rede para determinar o quão bem (ou não) o comportamento atende a respectiva intenção, conforme a métrica utilizada em sua definição. Durante a execução desse *loop* no Mediador, um algoritmo de tomada de decisão é executado a fim de determinar o melhor comportamento da rede. As fases do *loop* do Mediador são inspiradas no *loop de controle* de sistemas de software emergentes proposto em [Rodrigues-Filho and Porter 2017].

O algoritmo de tomada de decisão utilizado no planejamento é independente da métrica utilizada na definição da intenção. Para este artigo, no entanto, consideramos apenas a métrica “número de mensagens”. Para cada comportamento, a fase de monitoramento contabiliza todas as mensagens produzidas pelo comportamento durante um intervalo de tempo predefinido, aqui chamado de *janela de observação*. Assim, o melhor comportamento será aquele que produzir a menor quantidade de mensagens, sendo então escolhido para ser concretizado na rede.

Na camada de aplicação, estão presentes os comportamentos, as aplicações do usuário final e as intenções. Os comportamentos, que definem as funcionalidades da rede, são enviados à camada IDN para IoT por meio da *API RESTful*. O Desenvolvedor de Aplicações, por sua vez, deve realizar uma consulta a essa API para descobrir os tipos de

sensores que a rede possui. De posse dessas informações, ele pode especificar a intenção, que, por sua vez, é encaminhada à camada IDN para IoT por meio da mesma interface. Intenções são definidas na forma de pares atributo-valor utilizando a notação JSON (*JavaScript Object Notation*). Um exemplo de intenção que solicita a leitura dos dados dos sensores de forma periódica é mostrado na Figura 2. No exemplo, a intenção solicita os dados de temperatura a cada 30 segundos; além disso, a intenção indica que a quantidade de mensagens trafegadas na rede deve ser minimizada.

```
1  "Intent": {
2    "id": "001",
3    "funcionalidade": "solicitarDados",
4    "tipo_de_dado": "temperatura",
5
6    "periodicidade": {
7      "valor_periodicidade": "30",
8      "unidade_medida": "segundos"
9    }
10   "metrica": "quantidadeDeMensagens",
11   "objetivo": "minimizar"
12 }
```

Figura 2. Exemplo de definição de intenção

4. Avaliação

Esta seção está dividida em duas partes. A primeira descreve os diferentes cenários nos quais os experimentos serão executados. Já a segunda parte relata os resultados dos experimentos propriamente. Os experimentos foram projetados para explorar as seguintes questões de pesquisa:

- Q1. O Mediador consegue tomar a decisão correta e escolher o melhor comportamento, dada uma intenção e sua respectiva métrica, sem a necessidade de regras específicas pré definidas?
- Q2. Como o tamanho da janela de observação influencia o algoritmo de tomada de decisão no Mediador?

Para demonstrar o potencial do Mediador, é necessário avaliar a assertividade e o desempenho de seu algoritmo de tomada de decisão. Um fator importante nessa análise é a definição da janela de observação, *i.e.*, o tempo que o Mediador gasta para “observar” um dado comportamento em execução na rede e coletar as métricas que serão usadas para determinar a adequação do comportamento ao contexto de operação da rede. A análise, portanto, considera a execução do Mediador com diferentes janelas de observação para determinar o impacto do tamanho da janela na assertividade e no tempo de execução. A avaliação foi feita com o uso de uma implementação dos elementos essenciais da arquitetura, notadamente o Mediador e sua interface com a camada SDN para IoT.

4.1. Cenários utilizados no Experimento

O domínio de aplicação relacionado ao monitoramento da qualidade do ar foi considerado para fins de construção dos cenários utilizados para a validação da proposta. Dois cenários foram simulados para apresentar a definição de uma intenção e a tomada de decisão do Mediador na escolha do melhor comportamento, respeitando uma métrica específica, nesse caso, o número de mensagens trocadas. O primeiro cenário tem como

base o monitoramento da temperatura em um ambiente. Nesse caso, os dados lidos dos sensores não variam com muita frequência. O segundo cenário, por sua vez, representa o monitoramento de Compostos Orgânicos Voláteis Totais (COVT) no ambiente, sendo este um caso em que os dados lidos dos sensores variam com alta frequência.

Para este trabalho, foi implementado um conjunto representativo de comportamentos para coleta de dados na rede, sendo que cada comportamento é capaz de satisfazer a intenção da aplicação por completo. O primeiro comportamento, *Periódico*, envia os dados dos sensores em intervalos de tempo estabelecidos na intenção. O segundo comportamento, *Polling*, consiste no padrão de requisição e resposta, sendo necessárias duas mensagens, uma de solicitação e outra de resposta, para cada leitura de dados. O último comportamento, denominado *Alerta*, consiste em uma interação de notificação, combinada com uma *cache*; sempre que os sensores detectam uma variação significativa nos dados monitorados, uma notificação é gerada e o novo valor é armazenado em uma cache no Mediador, onde fica disponível para a aplicação. A configuração de cada comportamento, que inclui o tipo de dados monitorados e a periodicidade de leitura, é obtida a partir da interpretação da intenção especificada pelo desenvolvedor da aplicação.

Os cenários analisados tiveram o objetivo de demonstrar a exploração do conjunto de comportamentos pelo Mediador para escolher aquele que melhor se aplica à coleta de dados na rede em cada contexto operacional. O cenário de monitoramento de temperatura representa um contexto operacional com pouca variação dos dados monitorados, enquanto o cenário de monitoramento de COVT representa um contexto com alta variabilidade. A intenção aplicada está apresentada na Figura 2. Os experimentos foram realizados em uma rede criada no simulador Cooja 3.0, composta por 5 nós sensores (*motés*) e um nó sorvedouro (*sink*), os quais emulam o dispositivo EMB-Z2530PA, com o sistema operacional Contiki e o plano de controle do SDN-WISE. O ambiente de execução consiste de um computador Intel Core i7-4700MQ 2.40GHz, com 16GB de memória RAM e sistema operacional Windows 10 64 bits. O simulador foi executado em uma VM com Ubuntu 14.04 e 4GB de RAM, enquanto o controlador SDN e o Mediador foram executados diretamente na máquina física.

Os dados produzidos pelos sensores utilizados na simulação foram retirados de um *dataset* com dados reais coletados por uma solução IoT denominada Air-Pure [Cordeiro et al. 2019], um dispositivo com sensores de temperatura, dióxido de carbono e COVT. Os dados foram coletados por um único dispositivo, que realizou a coleta dos dados no período de cinco meses no corredor de um prédio da administração pública municipal. Apenas um conjunto dos dados foi utilizado na avaliação deste trabalho.

4.2. Resultados dos Experimentos

No experimento, o conjunto de comportamentos foi explorado pelo Mediador nos dois cenários descritos acima (monitoramento de temperatura e monitoramento de COVT), com dois tamanhos de janela de observação: 30 e 60 segundos. Cada execução do experimento foi dividida em duas fases: a fase de exploração, em que o Mediador executa cada comportamento por uma duração correspondente à janela de observação para monitorar o desempenho do comportamento com respeito à métrica adotada (quantidade de mensagens trocadas); e a fase de operação da rede, em que o comportamento escolhido pelo Mediador é executado na rede por um período mais longo (equivalente a 7 vezes a

duração da janela de observação) de modo a evidenciar seu desempenho em regime de uso da rede. Portanto, para cada cenário, o experimento teve a duração de 5 ou 10 minutos, dependendo do tamanho da janela de observação.

Para efeito de comparação, em cada cenário, cada comportamento foi executado separadamente de forma estática (isto é, como um comportamento fixo) por um período de tempo equivalente (5 ou 10 minutos). Desta forma, é possível observar que o comportamento escolhido pelo Mediador corresponde ao melhor comportamento em cada caso. Para relevância estatística, cada execução foi repetida 10 vezes, obtendo-se a média dos números de mensagens trocadas durante o intervalo correspondente à janela de observação. Os resultados obtidos são mostrados na Figura 3, onde a curva denominada “Mediador” corresponde à execução inicial dos três comportamentos na fase de exploração, seguida da fase de operação, onde ocorre a implantação e execução do comportamento escolhido. Para facilitar a identificação das duas fases nos gráficos, elas são delimitadas por uma linha pontilhada vertical.

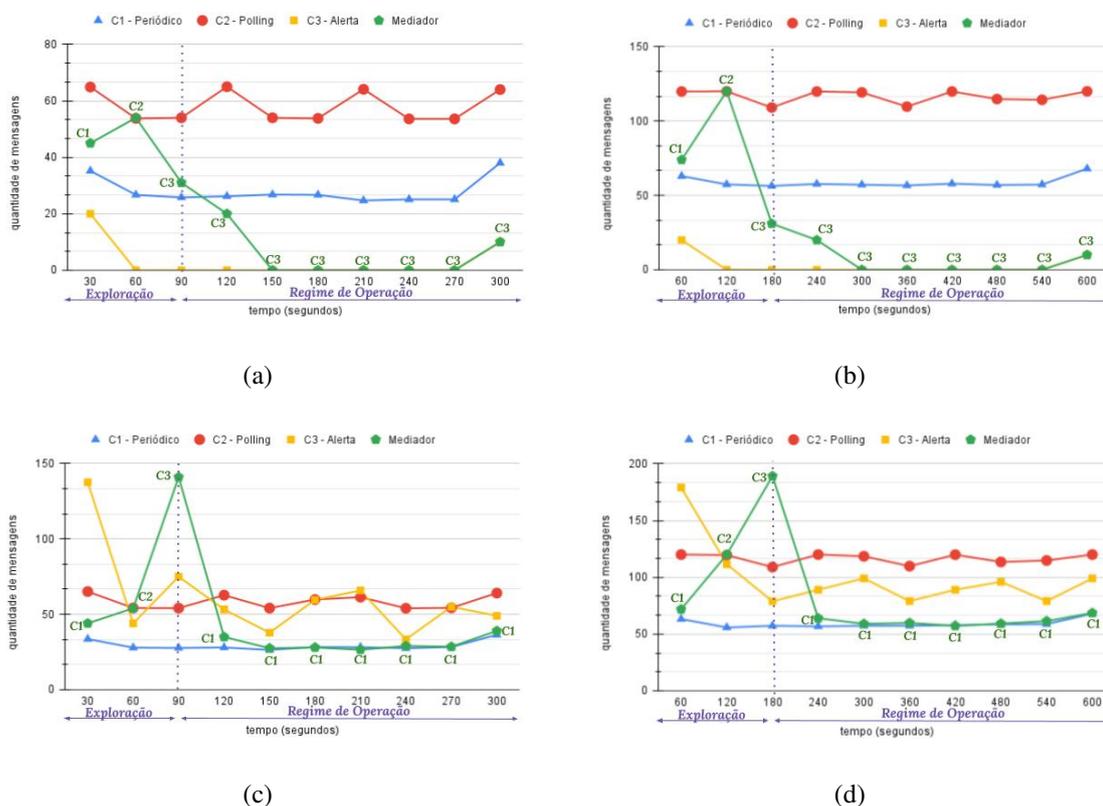


Figura 3. Resultados do Experimento. (a) Primeiro cenário com janela de observação de 30 s. (b) Primeiro cenário com janela de observação de 60 s. (c) Segundo cenário com janela de observação de 30 s. (d) Segundo cenário com janela de observação de 60 s.

As figuras 3(a) e 3(b) apresentam os resultados para o primeiro cenário, com janelas de observação de 30 segundos e 60 segundos respectivamente. É possível notar que o melhor comportamento foi o C3 - Alerta para ambos os tamanhos de janela, pois ele produziu a menor quantidade de mensagens, conforme esperado, uma vez que o cenário trata do monitoramento de temperatura, que apresenta pouca variação entre as leituras. O

comportamento *C1 - Periódico*, que a princípio deveria ser uma boa escolha para satisfazer intenções que requerem dados com regularidade, não se mostrou satisfatório quando a variabilidade dos dados é baixa.

Os resultados para o segundo cenário são apresentados nas figuras 3(c) e 3(d). Pode-se observar que o comportamento *C3 - Alerta* apresenta um desempenho bastante diferente daquele apresentado no cenário anterior. Isto se deve à variação frequente nas leituras dos sensores, ou seja, sempre que os dados monitorados por um sensor atingem os limites estabelecidos na intenção, uma mensagem é enviada para registro na *cache* do Mediador, mesmo que não seja utilizada pela aplicação. Portanto, em cenários onde a variabilidade é mais elevada do que o intervalo de leitura dos dados pela aplicação, o comportamento periódico se torna o mais adequado, visto que o envio das mensagens ocorre apenas sob demanda. A exploração realizada pelo Mediador, para ambas as janelas de observação, foi capaz de identificar o melhor comportamento e convergir corretamente para o comportamento *C1 - Periódico*, conforme pode ser visto na fase de operação.

Os pontos 120 e 240 nos gráficos indicam o primeiro intervalo de execução do comportamento escolhido pelo Mediador após o período de exploração, ou seja, a primeira execução do comportamento em regime de operação. Nas figuras 3(a) e 3(b) o comportamento escolhido, *C3 - Alerta*, possui alto custo de inicialização, o que explica a elevada quantidade de mensagens produzidas no instante inicial de execução do comportamento. A mesma observação é verificada no segundo cenário (figuras 3(c) e 3(d)), embora o comportamento escolhido neste caso, *C1 - Periódico*, tenha um custo de inicialização menor, resultando em um *overhead* inicial menos expressivo.

Os experimentos não mostram diferença significativa com os dois tamanhos da janela de observação, o que significa que uma janela de menor tamanho é suficiente e permite a escolha do melhor comportamento em menos tempo. Como trabalho futuro, é necessário investigar a configuração do tamanho da janela, de modo a reduzir a duração da fase de exploração ao mínimo necessário.

Finalmente, na Figura 3, observa-se que o comportamento *C2 - Polling* produz a maior quantidade de mensagens, tornando-o o pior comportamento para ambos os cenários. Isso ocorre pois ele utiliza o padrão de requisição e resposta para coletar os dados de cada *mote* na rede. Logo, para realizar intenções que solicitam dados de forma periódica, como a que foi utilizada no experimento, este comportamento não é indicado. Porém, para intenções que resultam na coleta de dados em momentos arbitrários, esse comportamento pode ser o mais indicado.

5. Considerações Finais

Este trabalho apresentou uma primeira incursão em uma nova abordagem para adaptação autônoma de redes IoT, resultante da união de dois conceitos que tem ganhado importância na literatura, IDN e SDN. A combinação dos dois conceitos foi realizada na forma de uma arquitetura em camadas, onde a camada de SDN para IoT é baseada em uma solução existente (SDN-WISE), enquanto a camada de IDN para IoT foi definida neste trabalho. Esta camada recebe e interpreta as intenções das aplicações, realizando-as por meio da escolha do melhor comportamento, dentre um conjunto de comportamentos disponíveis, considerando o contexto operacional da rede. Uma implementação inicial da proposta foi desenvolvida para fins de avaliação.

Os resultados mostram que o Mediador, que é o principal componente da camada IDN para IoT, foi capaz de tomar a decisão correta e escolher o melhor comportamento dada a intenção e uma métrica nela definida, sem o uso de regras específicas predefinidas. Os experimentos apresentados indicam a convergência correta do Mediador com janelas de observação a partir de 30s, demonstrando seu potencial para adaptar a rede de acordo com as intenções e o contexto operacional. A métrica utilizada na avaliação, número de mensagens, mostra-se útil, pois a redução da quantidade de mensagens trocadas em uma rede IoT pode evitar congestionamento, e por consequência, a perda de mensagens durante o regime de operação. Ademais, a redução do número de mensagens pode influenciar diretamente na economia de energia nos nós da rede.

Como trabalho futuro, o tratamento de possíveis conflitos de intenção deve ser levado em consideração, pois apesar da intenção ser específica da aplicação para o usuário final, podem haver intenções de outras aplicações com objetivos semelhantes. Além disso, o uso de diferentes métricas para avaliação dos comportamentos na camada IDN para IoT, como consumo de energia e confiabilidade, acrescentará maior flexibilidade à abordagem. Um outro trabalho futuro refere-se ao desenvolvimento da API RESTful da camada, para permitir a criação de intenções arbitrárias, bem como a inclusão de novos comportamentos. Finalmente, para permitir a adaptação dinâmica da rede, pode-se acrescentar um laço autônomo na fase de operação da rede, de modo a monitorar seu desempenho (com respeito à métrica usada na escolha dos comportamentos) e, se necessário, disparar a execução de uma nova fase de exploração pelo Mediador.

Agradecimentos

Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes, financiado pela CNPq (proc. 465446/2014-0), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e FAPESP (procs. 14/50937-1 e 15/24485-9). Bruna Michelly de O. S. Cordeiro agradece a CAPES pela bolsa de mestrado. Roberto Rodrigues Filho agradece à FAPESP por financiar seu trabalho de pós-doutorado através da bolsa vinculada ao processo 2020/07193-2.

Referências

- Aschenbruck, N., Bauer, J., Bieling, J., Bothe, A., and Schwamborn, M. (2012). Selective and secure over-the-air programming for wireless sensor networks. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE.
- Azzara, A., Alessandrelli, D., Bocchino, S., Petracca, M., and Pagano, P. (2014). Pyot, a macro-programming framework for the internet of things. In *Proceedings of the 9th IEEE international symposium on industrial embedded systems (SIES 2014)*, pages 96–103. IEEE.
- Bera, S., Misra, S., and Vasilakos, A. V. (2017). Software-defined networking for internet of things: A survey. *IEEE Internet of Things Journal*, 4(6):1994–2008.
- Blair, G. (2018). Complex distributed systems: The need for fresh perspectives. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1410–1421.
- Cerroni, W., Buratti, C., Cerboni, S., Davoli, G., Contoli, C., Foresta, F., Callegati, F., and Verdone, R. (2017). Intent-based management and orchestration of heterogeneous openflow/iot sdn domains. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–9.

- Cordeiro, Bruna M. O., S., Junior, M. B. R., and Junior, I. G. S. (2019). Air-pure: Uma solução iot da qualidade do ar interior. Monografia (Bacharel em Engenharia de Computação), UFG (Universidade Federal de Goiás), Goiânia, Brazil.
- Davoli, G., Cerroni, W., Tomovic, S., Buratti, C., Contoli, C., and Callegati, F. (2019). Intent-based service management for heterogeneous software-defined infrastructure domains. *International Journal of Network Management*, 29(1):e2051.
- Elkhatib, Y., Coulson, G., and Tyson, G. (2017). Charting an intent driven network. In *2017 13th International Conference on Network and Service Management (CNSM)*, pages 1–5. IEEE.
- Galluccio, L., Milardo, S., Morabito, G., and Palazzo, S. (2015). Sdn-wise: Design, prototyping and experimentation of a stateful sdn solution for wireless sensor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 513–521. IEEE.
- Jacobs, A. S., Pfitscher, R. J., Ferreira, R. A., and Granville, L. Z. (2018). Refining network intents for self-driving networks. In *Proceedings of the Afternoon Workshop on Self-Driving Networks*, pages 15–21.
- Júnior, J. C., da Cunha, D. C., and Ferraz, C. A. (2021). Integrating context awareness and sdn for a lightweight approach to adaptive networking. In *Anais do XIII Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, pages 91–101. SBC.
- Junior, S., Riker, A., Silvestre, B., Moreira, W., Oliveira-Jr, A., and Borges, V. (2020). Dynasti—dynamic multiple rpl instances for multiple iot applications in smart city. *Sensors*, 20(11):3130.
- Mai, T., Garg, S., Yao, H., Nie, J., Kaddoum, G., and Xiong, Z. (2021). In-network intelligence control: Toward a self-driving networking architecture. *IEEE Network*, 35(2):53–59.
- Noor, J., Tseng, H.-Y., Garcia, L., and Srivastava, M. (2019). Ddflow: visualized declarative programming for heterogeneous iot networks. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 172–177.
- Osman, M., He, J., Mokbal, F. M. M., Zhu, N., and Qureshi, S. (2021). MI-lgbm: A machine learning model based on light gradient boosting machine for the detection of version number attacks in rpl-based networks. *IEEE Access*, 9:83654–83665.
- Pang, L., Yang, C., Chen, D., Song, Y., and Guizani, M. (2020). A survey on intent-driven networks. *IEEE Access*, 8:22862–22873.
- Rodrigues-Filho, R. and Porter, B. (2017). Defining emergent software using continuous self-assembly, perception, and learning. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 12(3):1–25.
- Rodriguez-Zurrunero, R., Tirado-Andrés, F., and Araujo, A. (2018). Yetios: An adaptive operating system for wireless sensor networks. In *2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops)*, pages 16–22. IEEE.
- Shafi, N. B., Ali, K., and Hassanein, H. S. (2012). No-reboot and zero-flash over-the-air programming for wireless sensor networks. In *2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 371–379. IEEE.