Protocolos de Redundância de Gateway Aplicados em Redes IoT

Cassio Fabius C. Ribeiro[1], Francisco L. de Caldas Filho[1], Lucas M. C. e Martins[1], Cláudia J. Barenco Abbas[1], Rafael T. de Sousa Júnior[1]

***Resumo*— Os *gateways* para Internet das Coisas são componentes de extrema importância para conectar dispositivos com baixa capacidade de recursos (processamento, memória e/ou energia) a *middlewares* na Internet. Para minimizar o tempo e o impacto de sua indisponibilidade, este trabalho apresenta uma abordagem de redundância num *gateway* semântico. Com a proposta desenvolvida, conseguimos produzir uma solução de contingência na qual o *gateway* passivo assume todas as funções desempenhadas pelo ativo em pouco mais de um segundo, garantindo alta disponibilidade e aumentando a confiabilidade de redes de Internet das Coisas.**

***Palavras-Chave*— Gateway, Internet das Coisas, Redundância, Alta Disponibilidade.**

***Abstract*— Internet of Things gateways are extremely important components to grant connectivity for devices with low availability of resources (processing, memory and/or energy) to middlewares on the Internet. To minimize the time and impact of their unavailability, this work presents a redundancy approach in a semantic gateway. With the developed proposal, we are able to provide a contingency solution in which a slave gateway assumes all the functions performed by the master in a little bit more than one second, ensuring high availability and improving Internet of Things networks reliability.**

***Keywords*— Gateway, Internet of Things, Redundancy, High Availability.**

I. Introdução

As redes de computadores e, em especial, a Internet abarcaram diversos aspectos da vida cotidiana. A partir do final dos anos 2000, emergiu um modelo de integração da Internet com objetos do mundo real ligados a dispositivos RFID. Em 2010, [1] compilou o estado da arte da pesquisa nessa área, descrevendo o paradigma Internet das Coisas (*Internet of Things* ou IoT), reforçando a necessidade de convergência entre os objetos físicos, a Internet e a semântica. Em [2], corrobora-se que IoT tem foco nas informações e não nos recursos tecnológicos e de comunicação que o suportam.

Por se tratar de um paradigma de integração entre tecnologias e plataformas, as soluções de IoT tendem a ser de silo, segundo as especificações de seu fabricante. Ou seja, são soluções isoladas que não conversam entre si. Mesmo quando o fabricante disponibiliza seu protocolo de forma interoperável, esbarra-se na limitação de que alguns dispositivos não possuem a capacidade de implementar outros protocolos além dos seus originais. A utilização de um *gateway* IoT semântico para permitir a interoperação entre dispositivos de diferentes protocolos e *middlewares* específicos é proposta em [3], [4], [5]. Um *gateway* IoT semântico é um elemento responsável pela comunicação entre dispositivos e/ou redes que utilizam protocolos diferentes, realizando as traduções necessárias e repassando pacotes para o próximo nó ou cliente, possibilitando que a troca de dados ocorra. Ele pode ser um roteador, um computador ou simplesmente um software.

Em redes IoT, ele se torna um componente essencial, como se fosse o coração da rede. O **gateway** IoT é a entidade responsável por realizar a conexão entre os dispositivos IoT e a rede onde o sistema que os gerencia está localizado. A Figura 1 ilustra esse papel para os ambientes IoT. Devido à sua importância, pode-se concluir também que o *gateway* IoT se torna um ponto único de falha desse ambiente, ou seja, qualquer problema de indisponibilidade do *gateway* fará com que toda a rede IoT fique incomunicável.

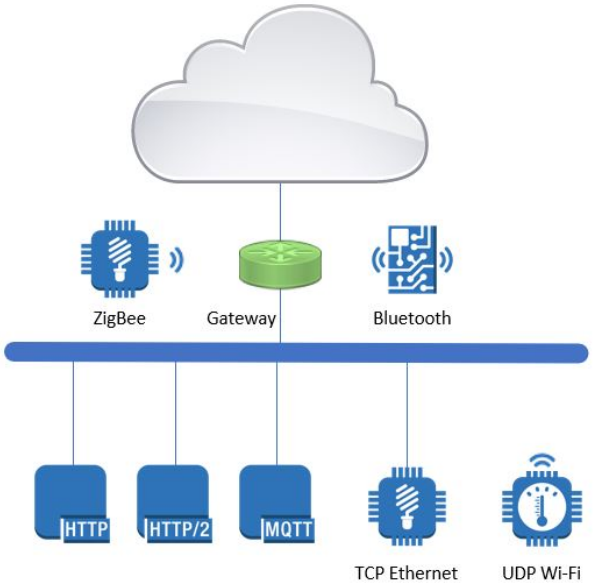


Fig. 1: O papel do *Gateway IoT* na rede é permitir que dados recebidos por meio de diferentes protocolos sejam enviados para redes externas

Conforme destacam [6], esse cenário evidencia a necessidade de se endereçar formas de evitar ou minimizar o impacto de indisponibilidade do *gateway* IoT. Assim, nesse trabalho, apresentamos uma proposta de *gateway* IoT com redundância que, em caso de falha na sua unidade principal, permita que uma instância secundária assuma seu papel de forma transparente aos seus clientes.

[1]Departamento de Engenharia Elétrica, Universidade de Brasília (UnB), E-mails: {cassio.fabius, francisco.lopes, lucas.martins} @uiot.org, {barenco, desousa} @unb.br

Além dessa seção introdutória, esse trabalho está organizado da seguinte forma: a Seção II traz um resumo da literatura sobre protocolos de redundância e conceitos importantes a fim de subsidiar nossa proposta, a Seção III apresenta os trabalhos relacionados, a Seção IV descreve nossa proposta de *gateway* IoT com redundância, a Seção V mostra como o *gateway* foi avaliado, bem como os resultados que foram obtidos e, por fim, apresentamos algumas conclusões e trabalhos futuros na Seção VI.

II. Protocolos de redundância

Diz-se que um sistema ou serviço possui **alta disponibilidade** quando há mecanismos capazes de mantê-lo em funcionamento mesmo em caso de falhas, seja tanto a nível de hardware quanto de software. Uma prática comum para prover essa garantia contra indisponibilidade é a **redundância**, que pode ser implementada pela presença de múltiplos elementos que realizam a mesma função.

Há diferentes maneiras de dispor estes componentes pelo sistema. As configurações mais conhecidas são:

- **Ativo-Ativo**: dois ou mais componentes estão ativos ao mesmo tempo e a carga que antes era destinada ao que falhou passa a ser distribuída entre os que ainda estiverem em funcionamento.
- **Ativo-Passivo**: apenas um componente permanece ativo enquanto um ou mais ficam em espera, preparados para tomar seu lugar em caso de falha.

A redundância também pode ser aplicada a diversos segmentos de um sistema ou serviço de rede:

- um servidor com duas fontes de energia possui redundância para o caso de um dos cabos ou dos circuitos elétricos falhar;
- dados são sincronizados entre dois equipamentos distintos de uma empresa para garantir que não haja perda total de informação em caso de falha num deles;
- sites são hospedados em data centers longes um do outro, para evitar indisponibilidade de páginas da web em caso de desastre num dos locais ou cidades.

Como os *gateways* são os componentes responsáveis por permitir a comunicação de uma rede com outra, a sua indisponibilidade provoca o isolamento de todos os elementos da sua rede local.

Protocolo de Redundância de Primeiro Salto (ou *First Hop Redundancy Protocol* - FHRP) é um termo utilizado para se referir ao tipo de protocolos que visam gerar uma redundância no *gateway* padrão de uma rede, ou seja, possuem o objetivo de promover alta disponibilidade na camada três do modelo *Open System Interconnection* (OSI), entre clientes (*hosts*) e a porta de saída da rede local, garantindo que, mesmo que um roteador apresente falha, outro em perfeito funcionamento tome seu lugar.

Em geral, há dois ou mais roteadores que se unem para formar um grupo que fica responsável pelas funções de *gateway*. Alguns protocolos oferecem apenas redundância, enquanto outros oferecem também balanceamento de carga (*load balancing*).

Os protocolos mais conhecidos serão descritos a seguir, para que então haja uma análise sobre a aplicação de cada um com foco em IoT.

1) *Hot Standby Router Protocol* (HSRP): o Protocolo de Roteador de Espera a Quente é um protocolo desenvolvido pela Cisco. Sua primeira versão foi documentada no RFC 2281 [7], mas sua segunda versão, apesar de trazer algumas melhorias, se tornou proprietária da Cisco. O HSRP não realiza o balanceamento de carga. Dois ou mais roteadores podem operar como um único elemento virtual, compartilhando os endereços IP e MAC. A definição de qual membro terá o papel ativo é baseada em eleição, utilizando critérios como prioridade e endereço IP. Os membros do grupo trocam mensagens entre si e, em caso de falha no ativo, um dos passivos assume o papel de forma transparente e automática.
2) *Virtual Router Redundancy Protocol* (VRRP): o Protocolo de Redundância de Roteador Virtual está na versão 3, a qual é documentada na RFC 5798 [8]. Por ser um padrão aberto, o VRRP é utilizado em diferentes dispositivos de rede como roteadores, *switches* de camada 3, *firewalls* e balanceadores de carga. Seu funcionamento é similar ao do HSRP. Um equipamento funciona como ativo, respondendo ao endereço IP virtual da instância VRRP, enquanto um ou mais membros passivos recebem anúncios do principal e que podem assumir o papel de ativo caso parem de receber anúncios.
3) *Gateway Load Balance Protocol* (GLBP): o Protocolo de Balanceamento de Carga de Gateway é outro protocolo proprietário da Cisco. É semelhante aos dois anteriores, mas com a função nativa de aproveitar os recursos disponíveis dos outros equipamentos que estão apenas aguardando uma possível falha do equipamento ativo. Diferente do HSRP e do VRRP, todos os roteadores do *cluster* podem operar no modelo ativo-ativo, promovendo a redundância e o balanceamento de carga. O balanceamento ocorre por fluxo de dados, utilizando critérios de endereço IP de origem/destino, endereço MAC de origem/destino ou porta TCP/UDP.
4) *Common Address Redundancy Protocol* (CARP): criado pela OpenBSD para fugir da guerra de patentes entre a Cisco e o IETF por conta do HSRP e o VRRP, o Protocolo de Redundância de Endereço Comum possui como diferencial o foco em segurança, criptografando seus pacotes de anúncio para evitar que intrusos passem a fazer parte de uma instância CARP.

Os protocolos proprietários HSRP e GLBP serviram apenas para conhecimento geral e como comparação entre as diversas opções existentes. Eles não podem ser utilizados numa rede IoT com produtos de diferentes fabricantes, logo não são aplicáveis aos objetivos do artigo e do projeto.

Dos que restaram, VRRP e CARP, ambos realizam bem a função de gerar alta disponibilidade do *gateway* padrão, além de terem sua especificação aberta. O VRRP foi escolhido para testes por atender os requisitos da aplicação, por ser mais simples de configurar e por não haver foco em segurança neste primeiro momento.

III. Trabalhos relacionados

Em geral, os trabalhos sobre IoT que abordam o tema da disponibilidade têm focado sua solução do lado do *middleware*, por meio da utilização da nuvem. Como exemplo, podemos citar [9], [10]. Esse tipo de solução é importante para endereçar a questão da disponibilidade do *middleware* e seus componentes. Porém, esse tipo de trabalho não apresenta nenhuma consideração sobre a possibilidade da indisponibilidade ocorrer no *gateway* IoT.

Em [11], apresenta-se uma opção para a indisponibilidade do *middleware* ou do canal entre o *gateway* e o *middleware*. No seu *gateway* IoT, utiliza-se do armazenamento local para garantir a operação do sistema nesses momentos de indisponibilidade. Porém, assim como nos demais casos, não trata a indisponibilidade do *gateway* IoT.

Os autores em [12] criaram um novo tipo de mensagem VRRP, visando reduzir o tempo de comutação entre o ativo e o passivo. Esta melhoria reduz o período de indisponibilidade da rede, entretanto não está focado em aplicações para redes IoT.

Os autores em [13] propõem uma estrutura de *gateways* de borda redundantes, responsáveis por receber e encaminhar dados utilizando o protocolo IPv6. O que difere esta proposta da apresentada em nosso artigo é a área de atuação. O trabalho dos autores está focado na camada de distribuição, enquanto a nossa proposta é criar uma redundância na camada de acesso, mais próxima aos sensores.

IV. Gateway IoT com redundância

Nesta seção é apresentada a proposta de *gateway* IoT com redundância. Inicialmente descrevemos a solução encontrada para a disponibilidade do equipamento em si. Em seguida, discutimos o tratamento realizado para os dados gerenciados pelo *gateway* IoT.

Disponibilidade do gateway: A primeira etapa para aumentar a disponibilidade desta solução foi a implementação de um protocolo FHRP no grupo de *gateways*. Conforme discutido na Seção II, o protocolo escolhido para isso foi o VRRP, que permite que dois ou mais elementos de rede respondam por um único endereço IP. Este endereço é o IP de *gateway* configurado em todos os clientes da rede. Ele deve sempre ser atendido pelo *gateway* ativo e, quando este estiver indisponível, um dos *gateways* passivos deve tomar o seu lugar, conforme ilustra a Figura 2.

A troca do papel de passivo para o de ativo ocorre sem nenhum sobressalto e de maneira automática, sem a necessidade de intervenção manual, aumentando a disponibilidade e a confiabilidade da rede. Durante a comutação automática, podem ocorrer perdas de comunicação. Estas perdas podem ser reduzidas alterando parâmetros de configuração. Os *gateways* foram configurados utilizando o *daemon* Keepalived, um programa *open source* que permite a configuração do protocolo VRRP em equipamentos com sistema operacional Linux.

Alguns parâmetros podem ser alterados no Keepalived. Alguns deles são: o papel do *gateway*, o intervalo de anúncio e a prioridade.

O primeiro parâmetro define o papel inicial do *gateway*, sendo ativo ou passivo. O segundo define a periodicidade que

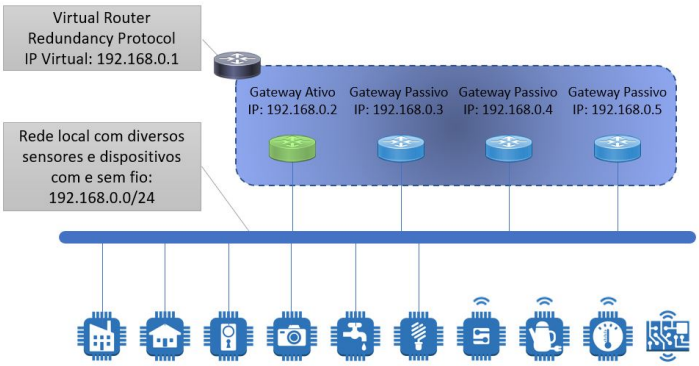


Fig. 2: Exemplo de configuração de *gateways* com VRRP

o *gateway* ativo anuncia ao grupo que ainda está ativo, técnica chamada de "*heartbeating*". O terceiro diz qual é a prioridade que cada membro do grupo do VRRP tem para responder caso seja detectada alguma indisponibilidade do ativo atual. Essa configuração é interessante caso seja necessário determinar em que ordem os passivos assumirão o papel de *gateway* ativo, considerando que os membros com prioridades superiores estejam indisponíveis.

Por padrão, o *gateway* que é inicializado primeiro assume o papel de ativo e os demais se tornam passivos. Caso mais de um *gateway* seja inicializado ao mesmo tempo e ainda não exista um ativo, o VRRP utiliza o parâmetro de prioridade para decidir qual deve assumir esse papel. Quando há alguma indisponibilidade do ativo, o passivo com maior prioridade torna-se o *gateway* ativo. Se o antigo *gateway* ativo retornar à rede, ele torna-se passivo. Dessa forma, como qualquer um dos *gateways* presentes pode se tornar o ativo, a disponibilidade é garantida enquanto houver ao menos um *gateway* ativo no grupo.

Dados gerenciados pelo gateway: gateways IoT não possuem apenas a função de encaminhar pacotes como os roteadores convencionais. Eles possuem diversas tarefas extras como o controle de admissão de sensores, permitindo que novos membros realizem a transmissão dos dados para nuvem e a tradução dos valores recebidos pelos sensores para chamadas REST ou para alguma outra API utilizada pelo *middelware*. Desta forma, a redundância garantida pelos protocolos FHRP são apenas parte da solução, pois se um sensor passasse a transmitir dados por um *gateway* que estava atuando como passivo, todo o processo de admissão teria de reiniciado, para só então ser possível dar continuidade à comunicação sem interrupções.

Em nossa proposta, para evitar este tipo de problema, o *gateway* passivo realiza a sincronização da base de dados do *gateway* ativo com a sua própria base de dados, utilizando-se da ferramenta rsync. Dessa forma, ele já possui uma base atualizada quando torna-se necessário assumir o papel de ativo e não precisa registrar novamente todos os dispositivos e serviços já autenticados no *middleware*. O software desenvolvido para atuar como *gateway* IoT, comunicando-se com um *middelware* e realizando o controle de admissão de dispositivos IoT, é chamado de **Universal IoT Gateway - UIoT Gateway**.

V. RESULTADOS

Após a implementação da proposta descrita na Seção IV, ela foi testada num cenário controlado num ambiente virtualizado no Oracle VM VirtualBox. Foram configuradas apenas duas máquinas virtuais ("VM") para atuarem como *gateways*, ambas com sistema operacional Linux Ubuntu Server 16.04 LTS, com os serviços Keepalived, rsync e UIoT Gateway instalados, configurados e habilitados. A versão do VRRP utilizada nos testes foi a versão 3, definida pela [8]. Ambos *gateways* possuíam prioridade 100, com a opção *nopreempt*, de modo que a VM que fosse inicializada primeiro se tornaria o *gateway* ativo da instância de VRRP e, ao cair e retornar após algum tempo, assumiria o papel de passivo se o outro *gateway* já tiver assumido o papel de ativo.

Tempo para comutação de gateway: a análise de comutação de *gateway* procurou obter o tempo de indisponibilidade do mesmo enquanto ocorre a troca entre o ativo e o passivo para diferentes valores de intervalo de anúncio do VRRPv3.

A Tabela I exibe os resultados do tempo de indisponibilidade com base nos valores escolhidos para testes: 0,1 s, 0,5 s, 1,0 s, 2,0 s, 5,0 s e 10,0 s. A coluna "calculado" exibe os valores que calculamos, conforme definido em [8]. A coluna "medido" demonstra o tempo coletado de simulações feitas no cenário de testes, com base no *timestamp* de *pings* enviados a cada 200 ms para o endereço IP virtual da instância de VRRP criada. Os valores apresentados são as médias de 10 comutações, que foram forçadas ao desativar a interface de rede do *gateway* ativo naquele momento. A Figura 3 apresenta os mesmos dados num gráfico de caixa, permitindo uma análise mais visual.

TABELA I: Indisponibilidade: Valores calculados VS valores medidos

Anúncio (s)	Prioridade	Skew Time	Calculado (s)	Medido (s)
0,1	100	0,0609375	0,3609375	0,632577
0,5	100	0,3046875	1,8046875	2,228022
1,0	100	0,6093750	3,6093750	4,312428
2,0	100	1,2187500	7,2187500	8,620555
5,0	100	3,0468750	18,0468750	21,353289
10,0	100	6,0937500	36,0937500	41,069723

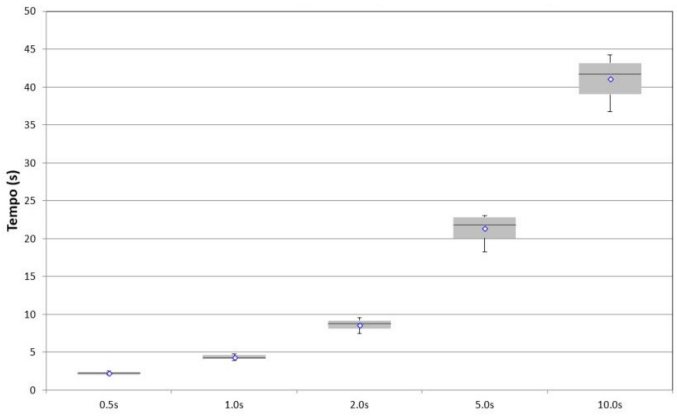


Fig. 3: Tempo de indisponibilidade para cada confirmação de tempo de anúncio

Impacto da indisponibilidade para os clientes: com o objetivo de obter um comportamento regular e homogêneo, desenvolvemos um programa em Python para representar o comportamento de um dispositivo IoT. Sua função era enviar a cada segundo uma leitura fictícia de um único serviço (como temperatura, umidade, etc) para o UIoT Gateway, sendo que a leitura corresponde a um valor numérico pseudo-aleatório. Os protocolos testados foram o TCP e o UDP. Esse programa foi executado num computador com sistema operacional Linux, distribuição Ubuntu 14.04, conectado na rede cabeada via interface Ethernet no laboratório onde os testes estavam sendo realizados.

No cenário de envio constante e monitorado de requisições do dispositivo virtual ao *gateway*, nosso experimento consiste em desativar a interface de rede do *gateway* ativo e observar por quanto tempo o serviço do UIoT Gateway permanecerá indisponível; o impacto do tempo de troca nos clientes; e se houve alguma perda de registro nos *gateways* quando houve a troca.

Para isso, consideramos os mesmos tempos de anúncio descritos na Tabela I. Para cada protocolo, apresentamos o tempo médio obtido após a observação de 10 comutações para cada caso, bem como os tempos para o melhor e o pior caso. Os resultados obtidos estão sumarizados na Tabela II. Todos os valores estão em segundos. A Figura 4 apresenta os mesmos dados num gráfico de caixa.

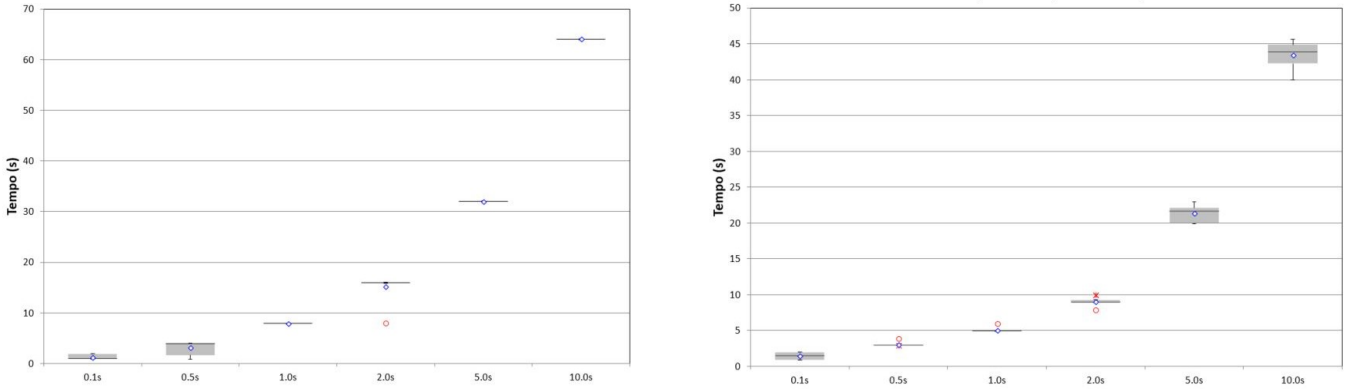
TABELA II: Tempo de indisponibilidade do UIoT Gateway para os clientes

Anúncio (s)	TCP			UDP		
	Menor	Maior	Média	Menor	Maior	Média
0,1	0,897	1,941	1,232	0,875	1,943	1,430
0,5	0,845	4,032	3,132	2,916	3,942	3,034
1,0	7,924	7,956	7,941	4,916	5,964	5,037
2,0	8,024	16,048	15,185	7,881	10,019	9,056
5,0	31,939	32,078	32,016	19,935	22,974	21,333
10,0	64,022	64,084	64,049	39,987	45,656	43,444

É interessante destacar que a indisponibilidade do serviço varia de acordo com o protocolo que está sendo utilizado na comunicação entre o dispositivo e o *gateway*. Numa conexão TCP, devido ao controle de entrega de pacotes, mesmo com mais de um minuto de indisponibilidade, há apenas um atraso na entrega da comunicação, sem que exista qualquer tipo de perda durante a comutação de *gateways*. Durante testes preliminares não registrados, entretanto, houve casos para 10 segundos de intervalo de anúncio em que o número de retransmissões na conexão TCP passou do limite e o pacote não foi entregue, mas não ocorreu tal situação durante a aquisição de dados. Além disso, é possível verificar o recuo exponencial do TCP em funcionamento nos dados coletados, já que os valores obtidos são múltiplos de 2.

Para conexões utilizando o protocolo UDP, o tempo de indisponibilidade representa um período no qual todas as comunicações serão perdidas, mas os pacotes passam a ser recebidos tão cedo um novo *gateway* ativo é designado.

Perda de registro entre gateways: por fim, analisou-se o registro das transações do serviço UIoT Gateway de ambas VM para verificar se a sincronização de suas bases de dados foi



(a) Conexões TCP (b) Conexões UDP

Fig. 4: Intervalo de anúncio VS Tempo de indisponibilidade

feita adequadamente. Em todos os casos analisados, verificou-se que o dispositivo IoT virtual era registrado corretamente pelo *gateway* ativo na sua primeira requisição e, quando a troca ocorreu, o novo *gateway* ativo utilizou o registro existente em seu banco de dados sincronizado para continuar enviando as informações do cliente ao *middleware*, sem a necessidade de realizar uma nova autenticação.

VI. Conclusões e Trabalhos Futuros

Conforme podemos observar na Seção V, a solução proposta conseguiu promover não apenas redundância na camada três do modelo *Open System Interconnection* por meio da utilização do *Virtual Router Redundancy Protocol*, como também alta disponibilidade para sensores, preservando o controle de admissão feito pelo *gateway* ativo fazendo sincronismo de banco de dados com o programa rsync.

Destacamos que o tempo médio de convergência foi de 1,23 segundos para sensores com transmissão TCP e 1,9 segundos para UDP. Esta solução se mostra robusta para atender ambientes que exijam alta disponibilidade, havendo a possibilidade de perdas de dados coletados pelos sensores apenas se o *gateway* receber dados durante o intervalo de convergência.

Como trabalhos futuros, vislumbramos incluir redundância para sensores *ZigBee*. Atualmente o UIoT Gateway recebe dados de sensores *ZigBee* e realiza a transferência destes para o *middleware*, porém não desenvolvemos a redundância para este tipo de dispositivos.

Também planejamos aproveitar a capacidade ociosa dos *gateways* passivos. Verificamos que estes elementos não aproveitam toda a sua capacidade de processamento, pois ficam ociosos, aguardando uma falha no ativo. Nosso objetivo é fazer com que o *gateway* passivo possa realizar o pré-processamento dos dados antes de enviá-los para a nuvem, evoluindo de uma solução de redundância para *Edge Computing*, realizando tarefas de inferência e compactação de dados.

Agradecimentos

Os autores agradecem o apoio das agências brasileiras de pesquisa, desenvolvimento e inovação CAPES (Projeto FORTE 23038.007604/2014-69), CNPq (Projeto INCT em Segurança Cibernética 465741/2014-2) e Fundação de Apoio à Pesquisa do Distrito Federal FAPDF (Projetos UIoT 0193.001366/2016 e SSDDC 0193.001365/2016), bem como ao Ministério do Planejamento, Desenvolvimento e Gestão/SPO (TED 005/2016 DIPLA e TED 011/2016 SEST), ao Gabinete de Segurança Institucional da Presidência da República (TED 002/2017) e à Defensoria Pública da União (TED DPGU 066/2016).

Referências

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, pp. 2787–2805, oct 2010.

[2] C. C. M. Silva, F. L. d. Caldas, F. D. Machado, F. L. L. Mendonça, and R. T. de Sousa Júnior, "Proposta de auto-registro de serviços pelos dispositivos em ambientes de IoT," (Santarém-PA), Sep 2016.

[3] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in *Mobile Services (MS), 2015 IEEE International Conference on*, pp. 313–319, IEEE, 2015.

[4] F. L. d. Caldas Filho, L. M. C. e. Martins, I. P. Araújo, F. L. L. d. Mendonça, J. a. P. C. L. da Costa, and R. T. de Sousa Júnior, "Design and evaluation of a semantic gateway prototype for iot networks," in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, UCC '17 Companion, (New York, NY, USA), pp. 195–201, ACM, Dec 2017.

[5] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The internet of things has a gateway problem," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 27–32, ACM, 2015.

[6] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and paradigms*. Elsevier, 1 ed., 2016.

[7] "Glbp - gateway load balancing protocol."

[8] S. Nadas, "Virtual router redundancy protocol (VRRP) version 3 for IPv4 and IPv6," RFC 5798, Internet Engineering Task Force (IETF), March 2010.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[10] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, Dec. 2014.

[11] A.-M. Rahmani, N. K. Thanigaivelan, T. N. Gia, J. Granados, B. Negash, P. Liljeberg, and H. Tenhunen, "Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems," in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, pp. 826–834, IEEE, 2015.

[12] Q.-D. Nguyen, J. Montavont, N. Montavont, and T. Noël, "RPL border router redundancy in the internet of things," in *International Conference on Ad-Hoc Networks and Wireless*, pp. 202–214, Springer, 2016.

[13] R. Jesuraj, "Method and apparatus for learning VRRP backup routers," Abr 2011.