Avaliação de Redes *OSPF* com Roteamento Sensível ao Contexto e Aplicação de Conceitos *SDN*

Vitor S. Goulart, Luiz F. G. de Oliveira, Jorge G. S. dos Santos, Ugo S. Dias e Rafael T. de Sousa Jr.

***Resumo*— Este artigo propõe uma nova abordagem para o uso de medições obtidas por sensores em redes de computadores, com aplicação em topologias de Internet das Coisas (*IoT*) e em redes definidas por *software* (*SDN*). O desempenho da abordagem proposta é comparado a uma típica topologia de rede *OSPF* tradicional. O papel do processamento distribuído das informações de contexto e a inferência típica de ambientes *IoT*, bem como uma proposta para a ampliação do uso de dados de sensores em redes de computadores formam o tema central deste trabalho.**

***Palavras-Chave*— Internet das coisas, Redes Definidas por *Software*, Redes de Sensores, Roteamento, Sensibilidade ao Contexto, Sensoriamento Distribuído.**

***Abstract*— This paper proposes a new approach for the use of sensor network data, increasing the applicability of IoT networks and SDN applications, after an investigation of the utilization of the context environment information in network routing. Based on information obtained using local sensors, network nodes are able to modify routes and routing tables. This new approach is compared with traditional *OSPF* networks. The role of distributed processing of information for inference based on the context and the broader use of the sensed data and a proposed solution to this problem form the central theme of this work.**

***Keywords*— Context-awareness, wired networks, sensor networks, *OSPF*, distributed sensing, Internet of things, *SDN*, routing performance.**

I. Introdução

A evolução dos equipamentos eletrônicos digitais, em termos de processamento, baixo custo e consumo energético, tem sido responsável pelo crescimento da utilização de sensores nas redes de computadores. Arquiteturas distribuídas para diferentes tipos de aplicações são uma nova oportunidade para obter sistemas econômicos, flexíveis, escaláveis e confiáveis [1].

Em redes de computadores, sensores são dispositivos eletrônicos que normalmente contêm um microcontrolador, um *chip* de comunicação de rádio e outros periféricos, além de dispositivos para realizar determinada medida [2]. São capazes de se comunicar para formar redes sensoriais complexas [3].

Um ambiente de *Internet* das coisas (*IoT*, do inglês *Internet of Things*) tem como características: baixo consumo de energia, adaptabilidade e flexibilidade [4]. Tais sistemas podem ser encontrados em diversas aplicações pelo mundo, inclusive em redes de computadores, finalidade para qual servem de catalisador. Funcionam, portanto, como sistema integrado, tornando escaláveis, mais velozes e precisas topologias de redes clássicas de mercado [5].

Os autores pertencem ao Departamento de Engenharia Elétrica Universidade de Brasília, Brasília, Brasil 70919-970. E-mail: 170165396@aluno.unb.br, ziul@unb.br, jorge.guilherme@redes.unb.br, udias@unb.br, desousa@unb.br

É possível, portanto, unir os conceitos de *IoT* e redes de sensores com o objetivo de melhorar o funcionamento de uma rede cabeada tradicional. Para isto, deve-se utilizar as informações de contexto (dados de ambiente obtidos pelos sensores, que podem ser espalhados ou não pela rede) [6]. Tais dados devem ser interpretados com base em limiares e definições específicas. Estas, no entanto, devem ser definidas com base nas necessidades de cada aplicação ou cenário de rede [7].

Comparações entre aspectos de redes de sensores e *IoT* foram feitas em diversos trabalhos e artigos acadêmicos. Em [8] e em [9], o uso de *OSPF* (do inglês *Open Shortest Path First*) em redes com inteligência e interpretação de informações é estudado. Outros trabalhos, como [10] e [11], focam em comparações entre redes definidas por *software* (*SDN*, do inglês *Software Defined Networks*) e redes *IoT*.

Neste artigo, no entanto a proposta central é unir os conceitos acima apresentados. Em uma rede de computadores cabeada tradicional, sensores podem ser adicionados a cada nó para coletar informações de ambiente como temperatura, umidade, entre outras. Tais dados formam o contexto físico na qual cada dispositivo está inserido, e serão levados em conta na hora de construir tabelas de roteamento, além das métricas já utilizadas no protocolo *OSPF*.

O restante deste trabalho está organizado de forma a elucidar a solução proposta. Na seção II, uma rede *OSPF* tradicional é demonstrada. Na seção III, o cenário com a utilização de interpretação de informações sensoriais é proposto. É traçada uma comparação de desempenho entre as duas topologias na seção IV. Finalmente, na seção V, chega-se a uma conclusão a respeito dos cenários aqui abordados.

II. Protocolo *OSPF* em redes de sensores

OSPF é o IRP (do inglês *Interior Routing Protocol*) mais utilizado na *Internet*, tanto em redes tradicionais, quanto em sistemas *IoT*. É conhecido por ser um protocolo de estado de enlace, ou seja, cada nó da rede mantém um mapa da topologia [9]. Consequentemente, cada roteador pode construir localmente sua tabela de roteamento, para que os pacotes sejam encaminhados de forma otimizada por toda a rede [12].

No *OSPF*, um roteador percebe uma mudança na topologia ao receber uma mensagem de aviso de estado de enlace. Estas

mensagens são definidas no padrão do protocolo, e trocadas em intervalos determinados. Após uma alteração na rede, é necessário um tempo até que todos os nós sincronizem suas tabelas de roteamento, ou seja, até que as mensagens de aviso cheguem a todos os roteadores. Este tempo é chamado "tempo de convergência". Além disso, o protocolo utiliza o algoritmo de *Dijkstra*, juntamente com um atributo de custo por rota para calcular o melhor caminho entre dois nós [9].

Na Figura 1, é desenhada uma típica topologia de rede cabeada. O protocolo *OSPF* funciona nesta rede, proporcionando a comunicação entre as diferentes sub-redes dispostas. Os roteadores fazem parte da mesma área *OSPF*, o que facilita a troca de mensagens entre eles. Desta forma, todos os roteadores deste cenário conhecem todas as sub-redes. As demais áreas existem nesta topologia como simulações de redes externas.

O tempo de convergência médio para uma rede como a proposta é definido com base nos intervalos em que as mensagens de estado de link são trocadas entre roteadores, e também no tempo que cada roteador leva para detectar uma mudança na topologia. No *OSPF*, ao detectar que uma rota não é mais funcional, o roteador espera um determinado tempo para declará-la inutilizável. Este intervalo de tempo é chamado "*dead time*" completo [12].

Quando uma rota não é mais alcançável por um roteador, e o *dead time* se esgota, o roteador que percebe a mudança remove a rota de sua tabela de roteamento e envia avisos aos seus vizinhos com a mudança no cenário da rede. Tais avisos se espalham pela topologia e, depois de todo este tempo, a rede volta a convergir [13].

Sabe-se que o tempo de convergência é, portanto, a medida de performance mais clara de um protocolo de roteamento. Protocolos com longos tempos de convergência tendem a cair em desuso. Há vários fatores que podem afetar o desempenho geral do roteamento em uma rede. Dentre eles, temperatura e umidade estão entre os principais, pois podem afetar o funcionamento dos circuitos e elementos eletrônicos de um roteador, tornando mais lenta sua capacidade de enviar pacotes, bem como aumentando o tempo para que o dispositivo perceba alterações na rede ao seu redor [13]. A luminosidade também pode afetar os nós de uma rede, seja ela tradicional ou de sensores, que muitas vezes são energizados por baterias solares, ou dependem da luz para funcionar [14]. Todas essas métricas são entendidas como informações de contexto, e podem ser obtidas através de sensores.

III. Protocolo de roteamento sensível ao contexto

Para obter as informações de contexto, podem ser utilizados diversos tipos de sensores para as mais variadas métricas. Neste trabalho, especificamente, foram utilizados sensores de temperatura, umidade e luminosidade. Para demonstrar que as informações obtidas podem afetar positivamente o desempenho da rede, os sensores foram ligados a cada um dos nós da topologia proposta.

Para a temperatura e umidade, foi utilizado o sensor DHT11, e para a luminosidade, o sensor LDR [15]. Estes são simplesmente equipamentos de medição, não possuindo qualquer capacidade de interpretar os dados medidos. Ocupam pouco espaço físico e exigem pouca energia elétrica [14]. Desta forma, são a melhor opção para redes *IoT* escaláveis e adaptáveis.

Para os nós, ou seja, os roteadores da rede, foi montada uma estrutura de quatro microcontroladoras *Raspberry Pi*. Estes dispositivos possuem um sistema *linux*, o que possibilita sua programação com facilidade. Podem, portanto, ser transformados em roteadores com a utilização do software livre *Quagga* [16].

Quando todos os nós são ligados conforme a Figura 1, uma rede *OSPF* é configurada e todos os roteadores trocam mensagens nos tempos padrões do *OSPF*. Os pacotes de aviso de estado de enlace são trocados a cada dez segundos, e cada roteador espera 5 segundos antes de retransmitir uma mensagem deste tipo. Adicionalmente, o *dead time* padrão do *OSPF* é de quatro vezes o tempo de mensagem padrão, ou seja, quarenta segundos.

O tempo de convergência total é definido por

$$Ct = Dt + EPt + IPt, \tag{1}$$

em que *Ct* é o tempo de convergência total do *OSPF*, *Dt* é o *dead time*, *EPt* o tempo de propagação de um evento pela rede (*Event Propagation time*) e *IPt* o tempo de processamento interno médio de um pacote por um nó (*Internal Processing time*). Mesmo assim, não é levado em consideração se um dos roteadores encontra-se em um estado físico pior que algum outro, isto é, se tem seu desempenho prejudicado devido a altas temperaturas, por exemplo. Desta forma, uma base de comparação pode ser definida.

No protocolo sensível ao contexto, não há acréscimo de complexidade na estrutura de rede. Isto é, o número de roteadores ativos permanece o mesmo [17]. Portanto, o tempo de convergência não deve sofrer alterações após a instalação dos sensores. Além disso, as informações coletadas podem servir de base para a definição dos limites que caracterizam falha em um dispositivo. Ainda assim, podemos utilizar a equação (1) para calcular o tempo médio de convergência da rede.

Em um protocolo sensível ao contexto, as mudanças ambientais devem ser percebidas e, a partir de inferências realizadas, decisões devem ser tomadas visando fatores como disponibilidade e segurança do sistema como um todo. Estas também são características importantes de sistemas *IoT*. Neste trabalho, especificamente, as aplicações dos conceitos abordados podem ser estendidas a outros tipos de rede que utilizem um protocolo de roteamento interno para comunicação. Sistemas de Internet das coisas podem ser compostos por redes sem fio, *ad hoc*, além de redes de longa distância. Todos estes tipos de redes podem usufruir dos resultados obtidos neste artigo [17].

A topologia proposta de rede de computadores cabeada serve como dois casos de estudo e experimento prático para a obtenção dos dados de desempenho do protocolo. Afinal, os sensores podem simplesmente ser desligados do sistema para que ele funcione como uma rede *OSPF* tradicional. Assim, ao

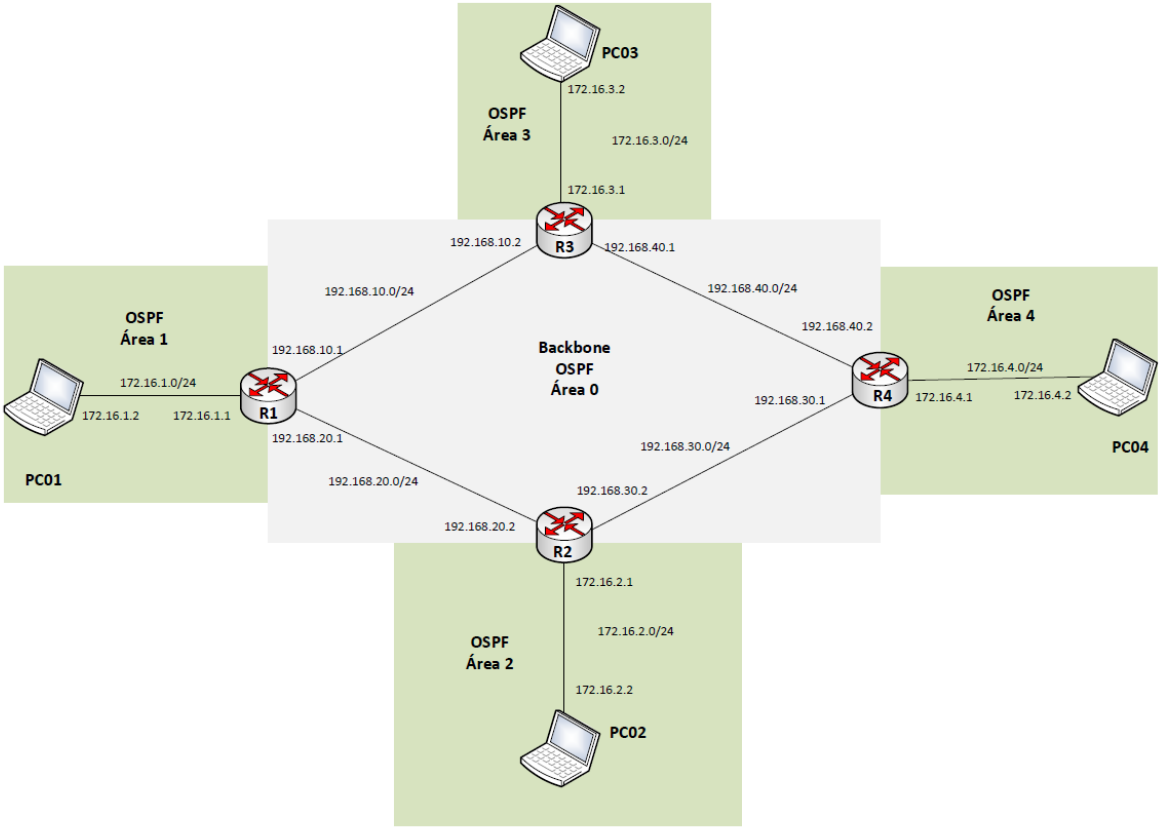

Fig. 1: Topologia de rede cabeada *OSPF* com quatro nós

observar o desempenho dos indicadores da rede e compará-los durante o funcionamento com e sem os sensores, é possível demonstrar a otimização realizada pelo protocolo proposto.

IV. Resultados

Quando os sensores são conectados a cada nó da rede, o tempo de convergência do *OSPF* permanece o mesmo, pois não foram alteradas as topologias física ou lógica do sistema. No entanto, os sensores passam a medir os dados de temperatura, umidade e luminosidade local em cada roteador.

As métricas escolhidas neste trabalho são conhecidas por afetar o funcionamento de dispositivos eletrônicos. Quando um roteador passa da temperatura indicada pelo fabricante, seu desempenho cai significativamente, pois os circuitos que fazem o encaminhamento de pacotes podem apresentar falhas. O mesmo vale para umidade. Desta forma, foram definidos os limiares de 50 graus Celsius e umidade maior do que 30%. Estes são limiares a partir dos quais roteadores de mercado já podem sofrer prejuízo em seu desempenho [18].

No cenário prático construído, limiares estáticos foram adotados afim de observar a reação do sistema a uma mudança de rota causada por uma medida dos sensores. Para tal, um programa capaz de interagir com os roteadores e, baseado nos dados obtidos pelos sensores, alterar suas tabelas de roteamento, se faz necessário.

Durante o funcionamento normal da rede, as tabelas de roteamento são construídas de acordo com o aprendizado das rotas pelos roteadores. Um *script*, escrito na linguagem de programação *Python*, é ativado em cada um dos nós, e passa a funcionar como uma controladora distribuída, interpretando as informações recebidas. No código, é definido o limiar para cada uma das medições (temperatura, umidade e luminosidade). Quando os sensores atingem dado limiar, o *script* altera o custo da rota *OSPF* no roteador em que o problema foi detectado. A alteração pode ser observada na Figura 2.

A alteração das rotas ocorre conforme demonstrado nas Figura 3. A tabela de roteamento do roteador R4 é alterada após a rota via R3 ser considerado pior. Esta rota foi alterada pelo *script* que interpretou as informações dos sensores. Dessa forma, os pacotes de um lado do anel que tem como destino o outro lado, não mais passam pelo nó com problemas físicos. Evita-se assim a perda de pacotes e de tempo, pois quando há alguma falha física na rede, ocorrem retransmissões.

Para calcular o tempo médio de convergência da rede, aplica-se a equação agora definida por

$$Ct = FDtS + EPt + IPt, \tag{2}$$

em que *FDtS* é o tempo de detecção de uma falha de um nó com sensores (*Failure Detection time with Sensors*), consideravelmente menor do que o *dead time*. O resultado é um tempo de convergência menor, cuja diferença em segundos para o protocolo tradicional tende a aumentar com o número de falhas na rede, visto que o *OSPF* espera um período de

```
root@R4:~Roteador# rcontext

interfaces descobertas:
 ['eth0',eth2']

Reading result of 2 sensors
 reduzindo custo para interface eth0
 reduzindo custo para interface eth2
 reduzindo custo para interface eth0
 reduzindo custo para interface eth2
```

```
LDR Value: 0.8560428619384766
reduzindo custo para interface eth0
reduzindo custo para interface eth2
Temp 21.0; Humidity: 17.0n
LDR Value: 0.8803615570068359
Temp 21.0; Humidity: 17.0n
LDR Value: 0.8504687956678986
Temp 21.0; Humidity: 17.0n
LDR Value: 0.8700567895456231
Temp 21.0; Humidity: 17.0n
LDR Value: 0.8727083206176758
```

Fig. 2: *script Python* alterando os roteadores

dead time inteiro para decretar uma rota como falha.

Assim, um roteador deste sistema, através dos próprios anúncios do *OSPF*, é capaz de compreender que há um nó com problema na rede, e encaminhar pacotes por outra rota. Desta forma, o tempo de convergência do protocolo permanece o mesmo, mas seu desempenho melhora significativamente, evitando perda de pacotes e a necessidade de retransmissões, conforme demonstrado na Figura 4. A perda de pacotes na rede devido a falhas aleatórias de equipamentos foi reduzida de forma significativa, pois os pacotes não são mais enviados para nós problemáticos.

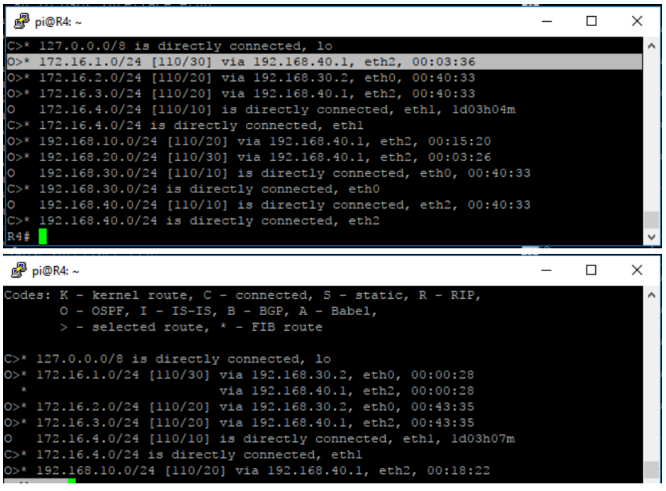

Fig. 3: Comprovação das rotas alteradas

Quando um nó se recupera da falha, o custo *OSPF* da rota volta ao valor original, fazendo com que o fluxo de pacotes

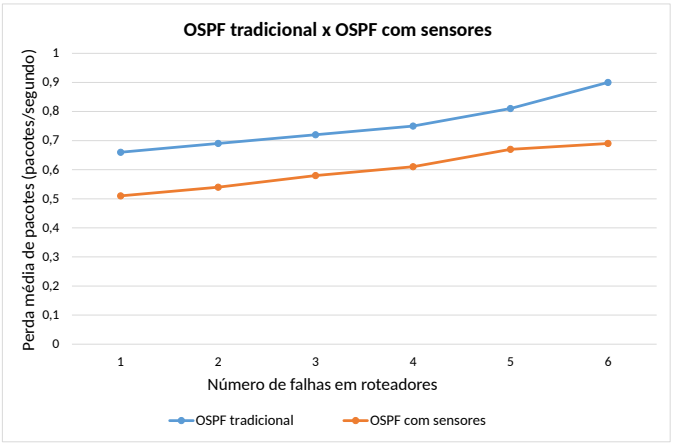

Fig. 4: Comparação entre perda de pacotes do *OSPF* com e sem os sensores

volte a seguir o caminho original determinado pelo protocolo de roteamento. Isto acontece pois o *script* lê continuamente as informações dos sensores, fazendo com que o roteamento seja alterado tão logo haja mudança no ambiente físico.

Outra comparação que pode ser feita para comprovar o melhor funcionamento da rede com a utilização dos sensores e da aplicação distribuída é a medida de latência. Dado que a rede *OSPF* tradicional trafega com uma certa taxa de pacotes por segundo, a rede que utiliza a aplicação com sensores proposta por este trabalho trafegará com a mesma latência enquanto não houver falhas de transmissão causadas pelas medidas físicas aqui adotadas.

No entanto, no momento de um problema devido a alta temperatura por exemplo, a rede com os sensores imediatamente se recupera do problema, sem necessidade de qualquer retransmissão. Afinal, a rota para o nó problemático foi alterada proativamente graças a aplicação aqui descrita. Conforme os limiares de temperatura, umidade e luminosidade são ajustados para melhor atender às necessidades do local em que a rede está instalada, o sistema tende a se tornar cada vez mais eficaz que o protocolo *OSPF* sem a utilização de sensores. O protocolo sensível ao contexto também acrescenta escalabilidade a rede, visto que a aplicação altera apenas o nó em que um problema foi detectado, a operação original do protocolo de roteamento permanece inalterada. Trata-se portanto de uma aplicação distribuída com ênfase na disponibilidade geral do sistema.

Uma comparação pode ser feita entre o protocolo *OSPF* tradicional e a rede com sensores e aplicação *IoT* em funcionamento: enquanto o protocolo *OSPF* tem como critério primário para seleção de rotas a velocidade do link, o protocolo sensível ao contexto aqui abordado utiliza-se primeiro das informações dos sensores para definir uma rota, ou seja, leva em conta as informações medidas com uma prioridade maior para construir tabelas de roteamento.

Adicionalmente, a aplicação tem a capacidade de alterar automaticamente e proativamente as rotas da rede, sem que haja necessidade de esperar por anúncios de estado de enlace ou aguardar um tempo pré-definido para não mais encaminhar pacotes por um caminho com problemas. Desta forma,

observa-se um ganho de desempenho com relação ao protocolo *OSPF*.

O sistema todo pode ser entendido como um típico sistema *IoT*, pois possui características como inferência e sensibilidade ao contexto em que está inserido. Com a interpretação dos dados obtidos dos sensores com base em comparações com limiares pré-definidos, a rede não precisa de interação humana para tomar decisões de roteamento com base em informações obtidas externamente. O diagrama sequencial da aplicação desenvolvida é mostrado na Figura 5.

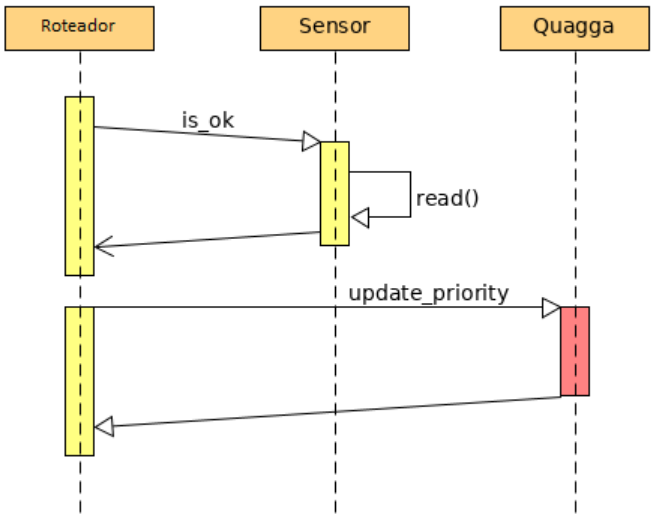

Fig. 5: Diagrama sequencial da aplicação

V. Conclusões

Este trabalho teve como proposta uma nova abordagem para redes de computadores tradicionais, com a utilização de sensores e conceitos de *IoT*. O uso de redes de sensores é adaptado para fornecer dados que são usados tanto para fins externos à rede quanto para contextualizar a própria rede, o que redefine as possibilidades de tráfego de dados em sistemas *IoT*.

Simulações anteriores foram implementadas e executadas para garantir o funcionamento correto do *firmware* e, finalmente, os testes em uma plataforma real foram implementados e executados com sucesso. O trabalho fornece uma nova abordagem ao roteamento de dados, melhorando o uso de todas as informações disponíveis para aumentar a inteligência no tráfego de rede.

O cenário proposto define uma maneira de combinar os protocolos de rede tradicionais, como o *OSPF*, com os conceitos de *SDN* e *IoT*, para desenvolver uma nova abordagem para redes de roteamento. Nesse esquema, a rede é capaz de aprender sobre si mesma e implantar mudanças em tempo real, com pouca ou nenhuma interação humana. Não há nenhum prejuízo para o tempo de convergência da rede, e há uma melhora significativa de desempenho, pois são evitadas retransmissões e perda de pacotes.

A abordagem proposta tem aplicações tanto em redes tradicionais de computadores, quanto em sistemas sem fio, *Data Centers* ou redes de longa distância. Quando um *software* é capaz de alterar automaticamente o encaminhamento de pacotes na rede sem interação humana, conceitos de *SDN* e *IoT* se unem para tornar as redes mais escaláveis e resilientes.

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] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microeletronics Journal*, pp. 1322–1336, 2009.

[2] E. C. Ian F. Akyildiz, Yogeh S. and W. Su, "A survey on sensor networks," *IEEE Communications Magazine, August 2002*, pp. 102–114, 2002.

[3] V. Q. Son, B.-L. Wenning, A. Timm-Giel, and C. Görg, "A model of wireless sensor networks using context-awareness in logistic applications," in *Intelligent Transport Systems Telecommunications,(ITST), 2009 9th International Conference on*. IEEE, 2009, pp. 2–7.

[4] J. C. Araiza Leon, "Evaluation of ieee 802.11 ah technology for wireless sensor network applications," 2015.

[5] T. B. T. B. E. C. e. M. P. Paulo F. Pires, Flavia C. Delicato, "Plataformas para a internet das coisas," *Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos - SBRC*, 2015.

[6] S. Yinbiao, P. Lanctot, and F. Jianbin, "Internet of things: wireless sensor networks," *White Paper, International Electrotechnical Commission, http://www. iec. ch*, 2014.

[7] C. C. N. L. . L. Y. Farabet, C., "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1915–1929, 2013.

[8] I. Akyildiz and X. Wang, "Wireless mesh networks (advanced texts in communications and networking)," 2007.

[9] K. Holter, A. Hafslund, F. Y. Li, and K. Øvsthus, "Design and implementation of wireless ospf for mobile ad hoc networks," in *Scandinavian Workshop on Wireless Ad-hoc Networks (ADHOC 06)*, 2005.

[10] J. N. Tayyaba S. K., Tanvir S., "Routing techniques in software defined networks: A survey," *13th IEEE International Bhurban Conference on Applied Sciences and Technology*, 2016.

[11] K. G. Thomas D. Nadeau, "Sdn: Software defined networks," 2013.

[12] M. K. U. S. Nurul Absar, Md. Abdul Wahab, "International journal of engineering research," *International Journal of Engineering Research*, vol. 6, no. 2, pp. 110–115, 2017.

[13] R. Mishra, M. Keimasi, and D. Das, "The temperature ratings of electronic parts," *Electronics Cooling*, vol. 10, no. 1, p. 20, 2004.

[14] D. Goyal and M. R. Tripathy, "Routing protocols in wireless sensor networks: A survey," in *Advanced Computing & Communication Technologies (ACCT), 2012 Second International Conference on*. IEEE, 2012, pp. 474–480.

[15] J. Mendes Junior and S. Jr, "Ldr e sensores de luz ambiente: Funcionamento e aplicacoes," *Semana de EletrÃ´nica e AutomaÃ§Ã£o 2013, At Ponta Grossa - PR*, 06 2013.

[16] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," in *Network, IEEE*, March 2014, vol. 28.

[17] Y. M. Akkaya, K., "A survey on routing protocols for wireless sensor networks," *AD HOC NETWORKS*, vol. 3, no. 3, pp. 325–349, 2005.

[18] P. S. MARIN, "Data centers: Desvendando cada passo: conceitos, projeto, infraestrutura física e eficiência energética," *1a ed. São Paulo: Ed. Érica*, 06 2011.