

Roteamento ciente de energia em redes de sensores sem fio definidas por software

Doriedson A. G. Oliveira e Cintia B. Margi

Resumo— Redes de sensores sem fio compõe-se de dispositivos com recursos limitados. A comunicação possui o maior consumo de energia, sendo importante otimizar o roteamento de dados. Apesar de propostas de protocolos cientes de energia, existe uma lacuna quando se trata do uso de energia como métrica de roteamento em redes de sensores sem fio definidas por software (SDWSN). Assim, apresentamos uma abordagem de roteamento ciente de energia em SDWSN. Os resultados obtidos, considerando o arcabouço TinySDN, demonstram que combinar as métricas de energia e ETX (*Expected Transmission Count*) melhora a taxa de entrega de dados e o tempo de vida da rede.

Palavras-Chave— Rede de Sensores Sem Fio; redes definidas por software; roteamento ciente de energia.

Abstract— Wireless Sensor Networks are composed by resource constrained devices. Communication is a key issue concerning energy consumption, and thus it is important to optimize data routing. Despite the energy-aware routing protocols proposed, there is a gap concerning the use of energy as a routing metric in software-defined wireless sensor networks (SDWSN). This paper presents an SDWSN energy-aware routing approach. Results obtained using the TinySDN framework show that combining energy and ETX (*Expected Transmission Count*) metrics improve data delivery rates and increases the network lifetime.

Keywords— Wireless Sensor Networks; software-defined networking; energy aware routing.

I. INTRODUÇÃO

Redes de Sensores Sem Fio (RSSF) abrangem diversas aplicações, como: detecção e acompanhamento, monitoramento de ambientes e indústrias, acompanhamento médico.

As RSSFs podem ser classificadas como um tipo especial de rede *ad hoc* de múltiplos saltos, já que apresentam características em comum com estas, mas possuem desafios adicionais a serem tratados, tais como fonte de energia e recursos computacionais limitados [1].

Uma das maiores preocupações ao se programar um dispositivo para RSSF é o consumo de energia, uma vez que o nó sensor possui restrições de energia já que depende de uma bateria.

Existem abordagens para redução no consumo de energia agnósticos aos protocolos de roteamento como o controle do ciclo de trabalho do rádio em ContikiMAC [4]. Essa abordagem independe dos protocolos em uso, mas pode afetar o seu desempenho, aumentando o atraso na entrega de dados.

Em relação aos protocolos de roteamento, uma das abordagens na literatura propõe o uso de *cluster heads* para centralizar o roteamento em nós com mais energia [15] dentro

de um grupo. Também é possível fazer uso de algoritmos naturais como o protocolo EEABR (*Energy-Efficient Ant-Based Routing*) [16], desenvolvido para usar formigas (pequenos pacotes) para descobrir rotas energeticamente eficientes de forma descentralizada.

O protocolo de roteamento IPv6 para redes de baixo consumo e com perdas (RPL - IPv6 Routing Protocol for Low-Power and Lossy Networks) [6] utiliza uma função conhecida como função objetiva zero (OF0 - *Objective Function Zero*) para montar a topologia e definir o roteamento. Tipicamente, essa função é análoga a contagem de saltos.

Lassouaoui *et al.* [7] apresentam uma análise comparando o uso de ETX [11], energia remanescente e mescla dos dois como funções objetivas do RPL. ETX (*Expected Transmission Count*) [11] é uma medida da qualidade do enlace entre dois nós em uma rede sem fio. Os autores concluem que a abordagem que utiliza a combinação do ETX e a energia remanescente mostra bom desempenho em termos de tempo de vida da rede e atraso fim-a-fim.

A descoberta de topologia do SDN-WISE [10], uma proposta de redes definidas por software (SDN - *Software-Defined Networking*) para RSSF, avalia três parâmetros em sequência: o número de saltos até o *sink*, o valor RSSI (*Receive Signal Strength Indicator*) recebido com a mensagem, e o nível residual da bateria. Os autores indicam que utilizam o algoritmo de Dijkstra para o cálculo de rotas no controlador SDN, mas não indicam a métrica usada.

As abordagens descritas utilizam energia como uma das métricas para o cálculo de rota realizado de maneira distribuída pelos nós (paradigma de roteamento tradicional) ou para a descoberta de vizinhos no SDN-WISE. Assim, dada essa lacuna do uso de energia como métrica de roteamento em SDWSN, este artigo apresenta uma abordagem de roteamento ciente de energia em SDWSN.

Como o foco do artigo são as SDWSN, os experimentos realizados comparam a abordagem ciente de energia proposta com o uso da métrica ETX somente, considerando o arcabouço TinySDN [2]. Os resultados obtidos demonstram que combinar as métricas de energia e ETX levam a aumento da taxa de entrega de dados e do tempo de vida da rede.

O restante do artigo é estruturado como segue. A Seção II apresenta uma visão geral das redes de sensores sem fio definidas por software, e as decisões de roteamento. O método de roteamento ciente de energia proposto é descrito na Seção III. A Seção IV descreve a metodologia e configuração das simulações, enquanto os resultados das simulações e sua análise são descritos na Seção V. A Seção VI conclui o artigo, apresentando as considerações finais e trabalhos futuros.

II. TRABALHOS RELACIONADOS

Esta seção apresenta as formas de roteamento abordados pelos protocolos que utilizam as premissas de SDN (*Software-Defined Networking*) para RSSF.

O SensorOpenFlow [17] é uma implementação do paradigma OpenFlow com ajustes para RSSF. Os autores especificam que há duas soluções: (i) uma solução sem IP ao qual o operador deve definir um protocolo de transporte e, (ii) outra com IP ao qual o SensorOpenFlow já está preparado para trabalhar utilizando a *stack* TCP/IP que utiliza o RPL [6] como protocolo de roteamento.

O FlowSensor [18] é outra aplicação dos paradigmas do OpenFlow para RSSF com o objetivo de alcançar confiabilidade na entrega de pacotes. Os autores denotam o uso de um algoritmo de roteamento nomeado "otimizador de rota", que leva em consideração o número de saltos e o custo da rota, porém não especificam a forma de cálculo de custo da rota. Também indicam como trabalhos futuros o estudo sobre o consumo de energia.

Wenxin *et al.* [19] propõem uma abordagem diferente para roteamento, na qual os nós são divididos em camadas de espaços lógicos de energia. Os nós nas camadas inferiores, com baixa energia, não roteiam pacotes de camadas superiores, com mais energia.

O TinySDN [2] é uma ferramenta desenvolvida para o TinyOS [3] que habilita o uso do paradigma SDN em RSSF. O TinySDN possui dois componentes principais: o nó sensor (*SDN-enabled sensor node*) que compõe um roteador SDN e um dispositivo de uso final e o nó controlador (*SDN controller node*) que centraliza a programação dos planos de controle. O TinySDN utiliza ETX como métrica de roteamento, obtido através do protocolo CTP (*Collection Tree Protocol*) [5] que é usado como protocolo adjacente para descoberta de controlador (CD - *Controller Discovery*) e descoberta de vizinhos (ND - *Neighbor Discovery*).

O SDN-WISE [10] utiliza o algoritmo de Dijkstra para o cálculo de rotas no controlador, mas não explica quais as métricas usadas. Também considera o uso eficiente de energia ao suportar ciclos de trabalho (*duty cycle*) e ao realizar agregação de dados.

A Tabela I resume as propostas de SDWSN na literatura, identificando os algoritmos e métricas utilizados para a determinação de rotas no controlador.

TABELA I

CRITÉRIOS DE ROTEAMENTO EM PROPOSTAS DE SDWSN

Abordagem	Algoritmo	Métrica
SensorOpenFlow	baseado no RPL	Função Objetiva 0
FlowSensor	Não identificado	Número de Saltos e Custo da Rota
Multi-dimensional energy space	Próprio	energia residual
TinySDN	Não identificado	ETX
SDN-Wise	Dijkstra	Não identificado

Em resumo, existe uma lacuna na literatura em se tratando do uso de energia como métrica de roteamento em SDWSN. A proposta descrita neste artigo é a primeira a aplicar esta abordagem para SDWSN e apresentar resultados de roteamento

ciente de energia em SDWSN.

III. ROTEAMENTO CIENTE DE ENERGIA PROPOSTO

Dada a lacuna do uso de energia como métrica de roteamento em SDWSN, este artigo apresenta uma abordagem de roteamento ciente de energia em SDWSN.

A configuração base considera como métrica o ETX, seguindo a especificação do TinySDN.

A primeira estratégia considerada foi utilizar a energia disponível como métrica no lugar do ETX. Desta forma, a decisão de rotas dependeria somente do nível de energia disponível nos nós. Isso traria um balanceamento do consumo de energia da rede. Porém, o nó deve reportar ao controlador SDN a sua energia residual.

No entanto, a variação do ETX ou da energia de um nó leva ao recálculo das rotas, o que implica em maior *overhead* de controle, já que é preciso reconfigurar as tabelas nos nós.

Assim, a estratégia proposta consiste em definir uma função que combine energia disponível no nó e ETX como métrica para que um controlador SDN possa calcular a melhor rota utilizando o algoritmo de Dijkstra. A estratégia é similar ao conceito proposto para o RPL por Lassouaoui *et al.* [7].

A função usada no algoritmo no controlador é definida pela Equação 1, sendo definida como uma média ponderada entre ETX e energia.

$$m = P_{ETX} \times ETX + P_{energia} \times EnDisp \quad (1)$$

A possibilidade de variar o peso das métricas traz maior flexibilidade a esta proposta, permitindo uma análise do impacto da seleção de seus pesos no desempenho da rede.

IV. METODOLOGIA

Escolhemos o arcabouço TinySDN [2] para avaliar o desempenho da nossa proposta. Como o TinySDN utiliza a métrica ETX por padrão, são necessárias alterações no seu mecanismo de definição de rotas para utilizar a função proposta.

O TinySDN define duas tabelas de fluxos em seus nós: (i) uma para armazenar rotas de pacotes de controles e (ii) outra para armazenar rotas de pacotes de dados conforme retratado na Tabelas II e III.

TABELA II

EXEMPLOS DE TABELA DE FLUXO DE DADOS

Fluxo ID	Ação	Parâmetro	Contador
2	Encaminhar	5	2
9	Receber	-	5
11	Ignorar	-	2
34	Encaminhar	10	50

TABELA III

EXEMPLOS DE TABELA DE FLUXO DE CONTROLE

Destino Nó ID	Próximo salto Nó ID
5	5
10	5
13	5
15	3

O protocolo SB (*South Bound*) do TinySDN é responsável pela comunicação entre o controlador e os nós e contempla: definição de fluxos, requisição de fluxos e anúncios de vizinhos.

O controlador disponibilizado junto ao código do TinySDN¹ é uma versão simples e funciona apenas para uma topologia em linha e com endereço sequencial.

Assim, desenvolvemos um *software* para o controlador que armazena a topologia da rede reportada pelos pacotes de anúncios de vizinhos (*Neighbor Report packet*) e usa o algoritmo de Dijkstra para o cálculo de rotas.

Os experimentos foram executados como simulações no COOJA [8], um simulador para RSSF que possibilita a emulação de código binário compilado para plataformas específicas.

Conectamos o *software* do controlador via TCP ao nó controlador no COOJA [8], usando assim o nó do controlador disponibilizado pelo TinySDN apenas como uma interface de rede para o *software* do controlador desenvolvido.

Para obter a energia residual de cada nó, adicionamos um método para estimar o consumo de energia no nó, levando em consideração apenas os tempos gastos para transmissão e recepção de mensagens do rádio. O valor obtido é reportado ao controlador através do pacote de anúncio de vizinhos.

Escolhemos o dispositivo TelosB [14], comumente usado em RSSF, para validação da proposta.

Para a simulação utilizamos uma topologia em grid (4x4), onde o controlador foi identificado como nó #0 (zero) e o *sink* como nó #1 (um). Os nós #2, #3, #4, #5, #8, #9, #12, #13, #14, #15 e #16, que formam o perímetro da rede, formam o grupo dos nós emissores responsáveis por enviar dados sensoreados para o *sink*, e o restante dos nós foram utilizados como roteadores.

Os nós foram configurados com o mesmo alcance do sinal de rádio. Assim os nós mais centralizados possuem 4 nós adjacentes como vizinhos, e os nós nas bordas da rede possuem 2 nós adjacentes como vizinhos conforme exibido na Figura 1.

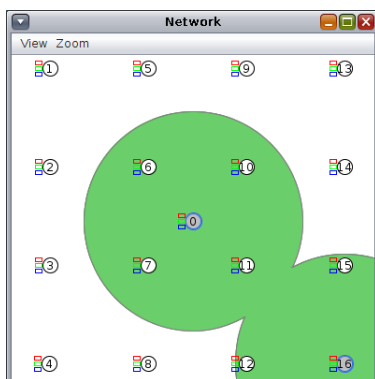


Fig. 1. Topologia da simulação (grid 4x4).

As simulações foram programadas para finalizarem ao detectarem o particionamento da rede. Entende-se por

particionamento da rede o momento no qual não é possível a comunicação do *sink* e do controlador com pelo menos um nó emissor.

Os pacotes de anúncio de vizinhos foram configurados para serem enviados com intervalos de um minuto, e os dados sensoreados com intervalos de cinco segundos.

O nó do controlador foi configurado com uma fonte de energia irrestrita, ou seja, sua energia sempre está em 100% durante a simulação.

As simulações foram realizadas em seis cenários. Para cada cenário foram dados pesos diferentes entre ETX e energia, conforme a Tabela IV, para obtenção da métrica usada no cálculo de rotas conforme a Equação 1.

TABELA IV
CENÁRIOS DA SIMULAÇÃO E PESOS EMPREGADOS.

	ETX	Energia
Cenário 1	100%	0%
Cenário 2	80%	20%
Cenário 3	60%	40%
Cenário 4	40%	60%
Cenário 5	20%	80%
Cenário 6	0%	100%

As métricas obtidas incluem: tempo até particionamento da rede, tempo do esgotamento da energia do primeiro nó, relação entre pacotes de dados enviados e recebidos, e relação entre pacotes de controle enviados e recebidos.

Cada resultado apresentado é a média de 10 execuções, e o desvio padrão é apresentado nos gráficos.

V. RESULTADOS OBTIDOS

Nesta seção são apresentados os resultados obtidos para as simulações dos cenários descritos utilizando o TinySDN [2].

A Figura 2 ilustra a relação entre o tempo de esgotamento da energia do primeiro nó e o tempo decorrido até o particionamento da rede. Observamos um sensível aumento no tempo referente ao esgotamento do primeiro nó conforme aumentamos o peso da energia como métrica de roteamento. Isso ocorre devido ao balanceamento do tráfego relacionado às mudanças no nível de energia de cada nó.

A linha de tempo relacionada ao particionamento da rede indica um aumento no tempo, seguida de um decaimento e logo após outro aumento. Analisando os *logs* das simulações identificamos que essa anomalia decorre por conta do tráfego intermitente ocasionado pelo protocolo adjacente de ND e CD utilizado pelo TinySDN. Verificamos que em algumas simulações houve pouco tráfego de pacotes de controle para ND e CD, assim economizando energia e permitindo os nós ficarem ativos por mais tempo. Em contrapartida, em outras simulações ocorreu o aumento de tráfego de pacotes do protocolo CTP, ocasionando o esgotamento prematuro de energia nos nós. O protocolo CTP é responsável pela comunicação *broadcast* para descoberta da raiz (controlador).

O protocolo CTP [5] utiliza um componente nomeado *link estimator* para estimar a qualidade de um enlace com outro nó vizinho através do envio e confirmações recebidas para uma série de pacotes, resultando na métrica ETX [11].

¹Disponível em <http://www.larc.usp.br/~cbmargi/tinysdn/>

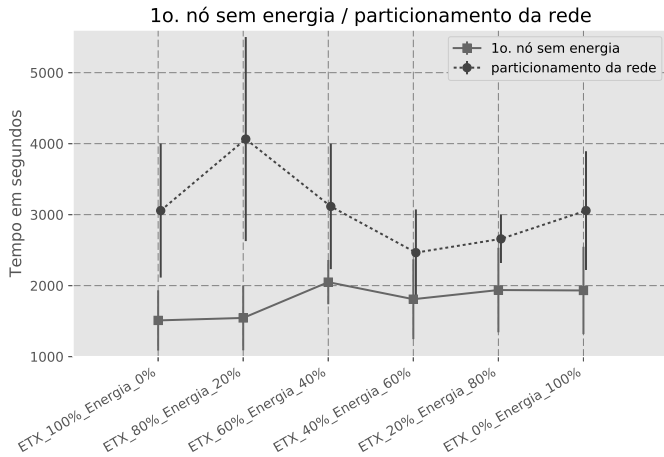


Fig. 2. Relação entre o tempo até o particionamento da rede e o tempo até o esgotamento da energia do primeiro nó na rede.

O resultado obtido no nó para cada avaliação do ETX é incluído em uma média histórica, de modo que variações ocasionais não interfiram na qualidade do enlace. Dessa forma, o cálculo de ETX do vizinho de um nó que esgotou a energia demorará a refletir a ausência desse enlace. Assim, quando o CTP atua como protocolo de ND para o TinySDN, as informações de vizinhos recebidas pelo controlador podem não refletir o estado atual da rede, seja devido a ausência de um nó pelo término de sua energia disponível ou devido a mobilidade do nó. Como a informação de topologia do controlador pode não estar sincronizada com o estado atual da rede, o cálculo de rotas poderá incluir um nó não disponível.

Um nó, que já possua uma entrada em sua tabela de fluxos determinada pelo controlador TinySDN, irá tentar transmitir um pacote de dados através do próximo salto indicado na sua tabela. Caso não ocorra confirmação do recebimento deste pacote pelo vizinho, o nó irá retransmitir o pacote. Dado que nossos cenários consideram que os nós podem ficar sem energia, a ausência de um vizinho fará com que um nó fique tentando reenviar um pacote continuamente, até sua energia acabar. Para evitar situações como essas, fizemos uma modificação no TinySDN para limitar o número de tentativas de retransmissão e ao atingir esse número o nó irá remover a entrada da tabela de fluxos, e efetuar uma nova requisição de rota ao controlador SDN. Um efeito esperado caso a visão da topologia do controlador ainda não esteja atualizada, devido ao modo como o ETX é calculado, será que o controlador continuará a configurar a tabela de fluxos do nó para utilizar a mesma rota através do vizinho que não está mais disponível. Esse processo pode se repetir várias vezes, até que a visão de topologia do controlador esteja correta.

Essa situação foi observado nos *logs* das simulações realizadas. Desta situação decorrem duas consequências: (i) aumento considerável no tráfego de pacotes de controle, e (ii) maior consumo de energia nos nós devido ao aumento da comunicação.

A Figura 3 ilustra este efeito. Observa-se mais pacotes de controle para o cenário que usa somente ETX como métrica de

roteamento do que para o cenário que utiliza somente energia como métrica de roteamento.

Para lidar com essa limitação da atualização da topologia decorrente do modo como o ETX é calculado, em um cenário no qual o controlador é ciente do nível de energia disponível nos nós da RSSF, é necessário que o controlador utilize um limite mínimo de energia disponível no nó para que ele seja usado em uma rota. Por exemplo, um nó que tenha um nível de energia menor ou igual à 5% não deverá fazer parte de uma rota.

O envio de dados está relacionado diretamente ao tempo em que os nós emissores estão ativos, já que os nós sensores foram configurados para enviarem dados em intervalos de cinco segundos. Podemos confirmar essa relação sobrepondo o gráfico de envio de dados ao de particionamento da rede e verificar a existência de uma correlação entre eles. No entanto, o recebimento de dados está relacionado diretamente ao tempo de existência de rotas para o fluxo de dados.

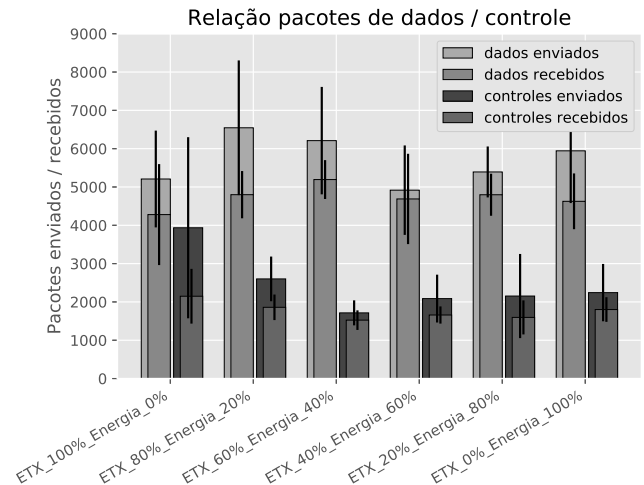


Fig. 3. Relação entre pacotes de dados e pacotes de controle enviados/recebidos na rede.

VI. CONCLUSÃO E TRABALHOS FUTUROS

A literatura inclui propostas de protocolos cientes de energia para as RSSF. Porém as propostas de redes de sensores sem fio definidas por software (SDWSN) não consideram energia como métrica na escolha de rotas. Assim, apresentamos uma abordagem do uso de roteamento ciente de energia em RSSF baseadas em SDN.

Os resultados obtidos, se desconsiderarmos os pacotes de controle enviados devido ao atraso da atualização de topologia por conta do ETX, mostram que usar somente a energia como métrica de roteamento aumenta sensivelmente o tráfego de pacotes de controle na rede, e por consequência ocorre maior consumo de energia, diminuindo o tempo de vida da rede. Ao usar somente o ETX como métrica de roteamento, observamos um menor tráfego de pacotes de controle, e por consequência uma economia de energia para o uso no tráfego de dados.

Assim, utilizar uma combinação da energia residual e do ETX como métrica de roteamento, conforme propusemos,

permite obter um equilíbrio entre o tempo de vida da rede como um todo, pois o tempo de vida entre os nós é mais homogêneo e demora mais até ocorrer o particionamento da rede.

Como trabalho futuro seria interessante avaliar um mecanismo para determinar os pesos das métricas ETX e energia residual dinamicamente, conforme o estado da rede (atraso, energia disponível nos nós, ocupação). Também é interessante avaliar o desempenho da proposta considerando uso de ciclos de trabalho para o rádio.

AGRADECIMENTOS

C. B. Margi is supported by CNPq research fellowship #307304/2015-9.

REFERÊNCIAS

- [1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer Magazine*, vol. 37, no. 8, pp. 41–49, 2004.
- [2] B. T. De Oliveira and C. B. Margi, "TinySDN: Enabling TinyOS to Software-Defined Wireless Sensor Networks," in: 34^o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) - Salão de Ferramentas, 2016, Salvador.
- [3] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, W. Weber, J. Rabaey, and E. Aarts, "TinyOS: An Operating System for Wireless Sensor Networks," *Ambient Intell.*, pp. 115–148, 2005.
- [4] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," SICS Technical Report T2011:13, ISSN 1100-3154, 2011. [Online]. Available: <http://dunkels.com/adam/dunkels11contikimac.pdf>. [Accessed: 04-Apr-2017].
- [5] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM Trans. Sens. Networks*, vol. 10, no. 1, pp. 1–49, 2013.
- [6] "RFC 6550", T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, R. Alexander, Rpl: Ipv6 routing protocol for low-power and lossy networks" Internet Requests for Comments, IETF, 2012.
- [7] Lassouaoui, L., Rovedakis, S., Wei, A., National, C., Martin, S., & Rance, F. (2016). Evaluation of Energy Aware Routing Metrics for RPL. In 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 1–8). IEEE.
- [8] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with COOJA. In Local Computer Networks, Proceedings 2006 31st IEEE Conference on, pages 641–648, Nov 2006.
- [9] NOX: Towards an Operating System for Networks, Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martin Casado, Nick McKeown, Scott Shenker, CCR 2008. [Online]. Available: <http://yuba.stanford.edu/casado/nox-ccr-final.pdf>
- [10] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Comput. Electr. Eng.*, 2017.
- [11] De Couto, D. S. J., Aguayo, D., Bicket, J., & Morris, R. (2003). A high-throughput Path Metric for Multi-hop Wireless Routing. In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (pp. 134-146). <http://doi.org/10+1145/968985.969000>
- [12] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 2, pp. 551-591, 2013.
- [13] H. I. Kobo, A. M. Abu-Mahfouz and G. P. Hancke, "A Survey on Software-Defined Wireless Sensor Networks: Challenges and Design Requirements," in *IEEE Access*, vol. 5, no. , pp. 1872-1899, 2017.
- [14] MotelV, tmote sky Low Power Wireless Sensor Module, Datasheet. [Online]. Available: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>
- [15] Muruganathan, S. D., Ma, D. C. F., Bhasin, R. I., & Fajokuwo, A. O. (2005). A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine*, 43(3), S8-13.
- [16] Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. (2006). An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks. *Ant Colony Optimization and Swarm Intelligence*, 49–59.
- [17] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks." *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.
- [18] A. Mahmud, R. Rahmani, "Exploitation of openflow in wireless sensor networks", *Computer Science and Network Technology (ICCSNT) 2011 International Conference on*, vol. 1, pp. 594-600, 2011.
- [19] Liao Wenxing, Wu Muqing, & Wu Yuwei. (2016). Energy-efficient algorithm based on multi-dimensional energy space for software-defined wireless sensor networks. In 2016 International Symposium on Wireless Communication Systems (ISWCS) (pp. 309–314). IEEE.