Journal of Network and Computer Applications 59 (2016) 39–45

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

An energy-efficient SDN based sleep scheduling algorithm for WSNs

Yanwen Wang [a], Hainan Chen [b,a], Xiaoling Wu [a,b,c,d,e,*], Lei Shu [c]

[a] Guangzhou Institute of Advanced Technology, CAS, No. 1121, Haibin Rd, Nansha District, Guangzhou 511458, China
[b] Guangdong University of Technology, No. 100, Huanchengxi Road, University Town, Guangzhou 510006, China
[c] Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, China
[d] Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin 300300, China
[e] Shenzhen Institutes of Advanced Technology, CAS, No. 1068, Xueyuan Rd, Shenzhen University Town, Nanshan District, Shenzhen 518055, China

ARTICLE INFO

Available online 20 May 2015

Keywords:
Energy efficiency
WSNs
Sleep scheduling
SDN-ECCKN
EC-CKN

ABSTRACT

Energy efficiency in Wireless Sensor Networks (WSNs) has always been a hot issue and has been studied for many years. Sleep Scheduling (SS) mechanism is an efficient method to manage energy of each node and is capable to prolong the lifetime of the entire network. In this paper a Software-defined Network (SDN) based Sleep Scheduling algorithm SDN-ECCKN is proposed to manage the energy of the network. EC-CKN is adopted as the fundamental algorithm when implementing our algorithm. In the proposed SDN-ECCKN algorithm, every computation is completed in the controller rather than the sensors themselves and there is no broadcasting between each two nodes, which are the main features of the traditional EC-CKN technique. The results of our SDN-ECCKN show its advantages in energy management, such as network lifetime, the number of live nodes and the number of solo nodes in the network.

1. Introduction

Wireless Sensor Networks (WSNs) have been widely adopted to collect, process, transmit and receive on-the-spot data instead of human labors, especially in hush environment. Normally, nodes in a WSN are very hard or even impossible to be recharged or replaced. Moreover, for a sensor node, usually limited energy is supplied with batteries. These challenges need high level of autonomy and self-organization of each sensor node in the network (Baccour et al., 2012). Hence, each node must be capable to acquire some other nodes' information deployed in the same region of interest. By processing these information, nodes are able to automatically make decisions and change their sleep status. In recent research, many literatures about Energy balancing techniques for WSNs have been discussed (Zhangbing et al., 2014a; Kai et al., 2012) and the fusion of WSNs and other network techniques is increasingly studied (Zhangbing et al., 2014b; Luís et al., 2014; Joel and Paulo, 2010).

Sleep Scheduling mechanism is currently an efficient method to manage the entire network and make the energy management more efficient (Zhu et al., 2014). To save the energy, the key point of Sleep Scheduling mechanism is to automatically and deliberately shut down subsets of nodes while remain other nodes alive in each given time interval. By applying SS mechanism, each node in the network has opportunity to "sleep" instead of "being awake" all the time, while the connectivity of the entire network is not affected during the lifetime of the network (Zhu et al., 2012). However, the discovery of other nodes is implemented by broadcasting the relevant information from each node, which is called beacon data, to all its neighbors, and all neighbors must then broadcast their information back in every time interval, which cost a lot of communication energy. Furthermore, in WSNs, the energy consumption of sending a single bit of data is at least 480 times as much as performing one addition instruction by CPU (Kimura and Latifi, 2005), which means if the total transmission times of a network during its lifetime is reduced by one, 480 addition instructions can be completed. How to reduce the transmission times of a network while keeping the network connectivity becomes a difficulty in the study of energy management in WSNs.

Motivated by the challenges above, in this paper we propose a Software-Defined Network (SDN) based SS algorithm to reduce the total transmission time of a network during its lifetime while maintaining the network connectivity, hence prolong the network lifetime. EC-CKN algorithm is regarded as the prototype of our algorithm since the residual energy is the criterion considered by each node when judging its status in the current interval. The rest of this paper is organized as follows. In Section 2, the related work including EC-CKN algorithm and SDN technology will be briefly

---

* Corresponding author.

http://dx.doi.org/10.1016/j.jnca.2015.05.002

introduced. Then in Section 3, the proposed SDN based SS will be presented in detail. The results and analysis are shown in Section 4. Finally, we conclude the paper in Section 5.

2. Related work

It is important to prolong the lifetime of the entire network and manage the energy of the network more efficient. Multiple Sleep scheduling algorithms have been proposed by researchers in recent study to balance the energy consumption and prolong the lifetime of the network.

2.1. Sleep scheduling mechanisms

Yaxiong and Jie (2010) proposed a generic duty-cycling scheduling method based on stochastic theory; Chih-fan and Mingyan (2004) presented specific scheduling algorithms within each approach and analyzed their coverage and duty cycle properties. In Nath and Gibbons (2007) and Zhuxiu et al. (2011), two famous Sleep Scheduling algorithms were designed, which are called Connected K-Neighborhood (CKN) algorithm and an improved CKN: Energy Consumed uniformly-Connected K-Neighborhood (EC-CKN) algorithm, respectively. Both Sleep Scheduling algorithms can efficiently close the relatively low-power nodes while maintaining both network connectivity and reasonable routing latency. In CKN algorithm, each node is K-connected which means that for any node in the network, if it has more than *K* alive neighbors, then it decides to close itself; and if its number of neighbors is less than *K*, the node remains awake. CKN algorithm is a distributed SS algorithm which can effectively prolong the lifetime of each node and the entire network. However, the energy in CKN algorithm cannot guarantee to be uniformly consumed (Zhuxiu et al., 2011). Different from CKN algorithm, EC-CKN algorithm considered the remaining energy of each node on the basis of CKN, which can balance the energy consumption of the entire network and simultaneously keep the network K-connected.

Detailed procedure of EC-CKN is shown in Fig. 1 (Zhuxiu et al., 2011). We decide to choose EC-CKN as the fundamental algorithm of our SDN based algorithm because its judgement criterion is relevant to the nodes' residual energy, which can directly reflect the energy consumption of the entire network. In our SDN-ECCKN algorithm, SDN based architecture is adopted instead of traditional WSN to remove the broadcasting procedures and hence reduce the total transmission times. In Table 1, from step 2 to step 3, each node broadcasts twice to obtain its 1-hop and 2-hop nodes' status, which are used for the later judgement of its own status. These two broadcasting procedures for each node in each interval cost a lot of communication energy.

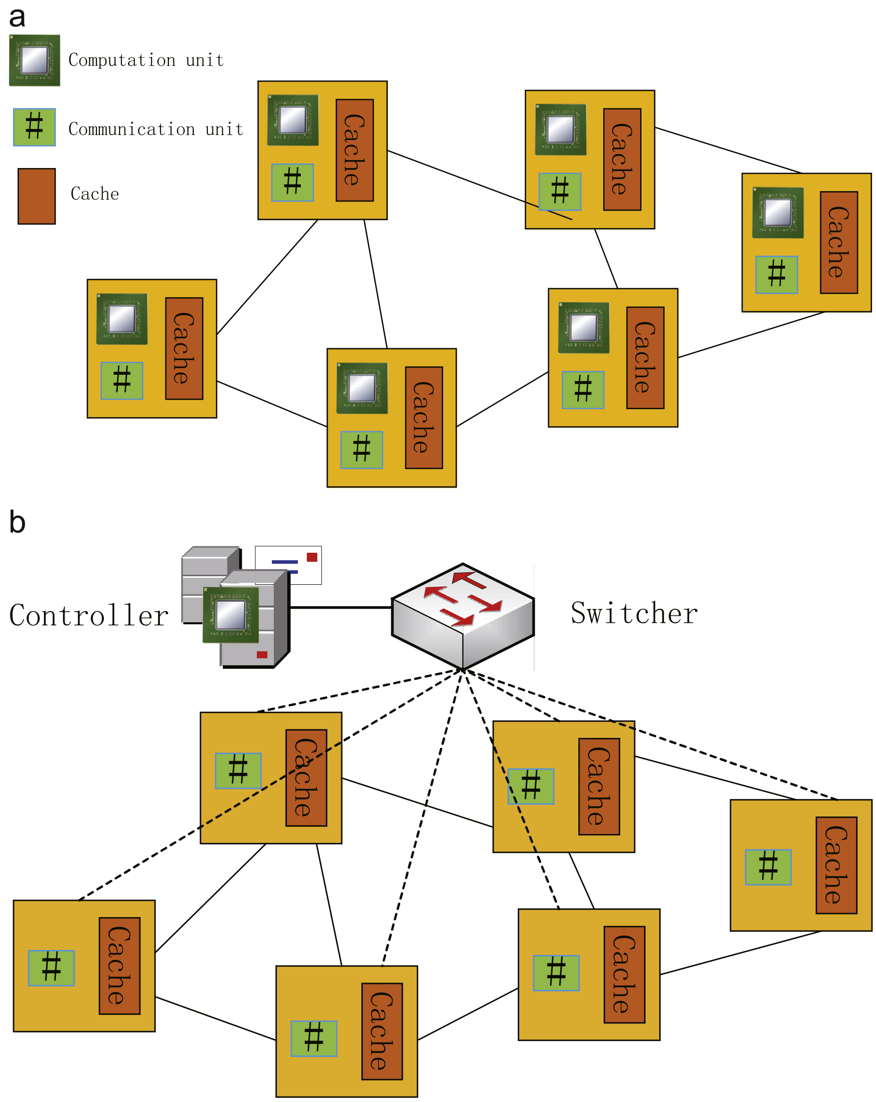

Fig. 1. Differences between traditional and SDN architecture. (a) Traditional WSN architecture. (b) SDN architecture.

2.2. Software-defined network

Since firstly proposed by McKeown et al. (2008), SDN has been paid more and more attention due to its unique and innovative features and superiorities, which has been deployed to multiple types of networks instead of traditional network architectures, such as campus networks (McKeown et al., 2008), cellular network (Li et al., 2012), data centers (Banikazemi et al., 2013), etc. The key technology in SDN is called OpenFlow (Lara and Kolasani, 2014), which is likewise the most different part compared to traditional network. OpenFlow is able to separate the computational unit and transmission unit of a single device. Every computation is completed in the controller rather than the devices themselves and all information is exchanged only through switch (Hata, 2013). In SDN, people are able to manage and configure the network device on demand via controller. Multiple network parameters, for example bandwidth or throughput, can be directly controlled by users rather than ISP (Xiangxin et al., 2013), which means the entire network can be managed more flexibly. Figure 2 shows the traditional network architecture (a) and the SDN architecture (b).

In Fig. 1(a), each node in traditional WSN architecture consists of 3 modules: a computation unit (usually a micro CPU), a communication unit (transceiver) and a cache that is used for storing collected data and corresponding forwarding tables. In traditional WSN, each node is directly connected to each other or indirectly connected but through intermediate nodes (Al-Turjman et al., 2013). At the beginning of data transmission, every node needs to calculate on demand by computation unit and broadcast the relevant data stored in cache several times to its every neighbor (Lee et al., 2013) via communication unit in each time interval, which consumes high communication energy.

However, in Fig. 2(b), the computation unit has been removed from each node. All computation is processed by the controller, which usually has powerful computation capability and constant power supply. Decisions are made only by the controller and then the switch forwards these decisions to each device. Each device only consists of a communication unit and a cache unit to communicate with switch and each device is directly controlled by the controller. There are no broadcasting procedures at the beginning of the data transmission, which decreases the total transmission time of the network.

Our SDN-ECCKN algorithm adopts the characteristics of SDN, hoping to reduce the total transmission times during the network lifetime and hence prolong the lifetime of the network. We use EC-CKN algorithm as our fundamental algorithm, while apply SDN architecture instead of traditional one to implement Sleep Scheduling mechanism.

Table 1
Energy consumed uniformly-CKN (EC-CKN).

```
1.  Get the information of current remaining energy $Erank_u$;
2.  Broadcast $Erank_u$ and receive the energy ranks of its currently
    awake neighbors $N_u$. Let $R_u$ be the set of these ranks.
3.  Broadcast $R_u$ and receive $R_v$ from each $s_v \in N_u$
4.  If $|N_u| < k$ or $|N_v| < k$ for any $s_v \in N_v$, remain awake. Return.
5.  Compute $E_u = \{s_v | s_v \in N_u$ and $Erank_v > Erank_u\}$;
6.  Go to sleep if both the following conditions hold. Remain awake
    otherwise.
    • Any two nodes in $E_u$ are connected either directly themselves
      or indirectly through nodes which is in the $s_u$'s 2-hop
      neighborhood that have $Erank_v$ larger than $Erank_u$;
    • Any node in $N_u$ has at least k neighbors from $E_u$.
7.  Return.
```

(*Run the following at each node s_u).

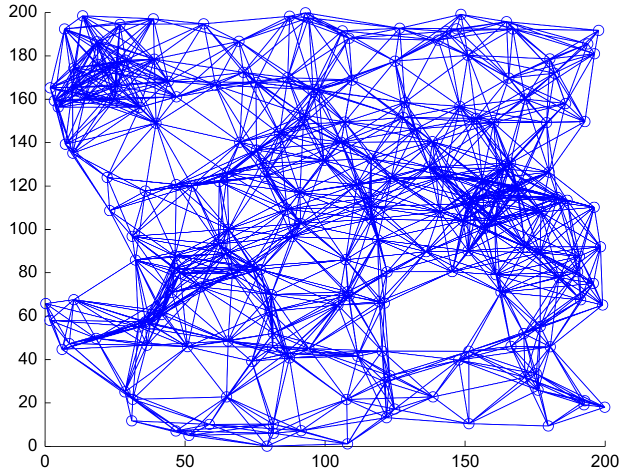

Fig. 2. A simple example of a full-connected network.

3. SDN-ECCKN algorithm

3.1. Simulation model

In this paper, the simulation area is a square with 200 × 200, in which 150 sensor nodes are randomly deployed. A single radio interface is embedded in each node, in which the same energy level is allocated at the beginning of the simulation. In this simulation, nodes are automatically alive or closed in different time intervals according to our algorithm, and the length of time interval is T. This type of network is called time-varying connectivity (TVC) network (Nath and Gibbons, 2007). In TVC network, the number of awake nodes in the global topology and the number of awake neighbors for each node are highly dynamic in each time interval T throughout the entire simulation. Assume that the packets' size keeps the same throughout the simulation time. The energy that a node transmits an l-bit long packet over distance d is defined as follows (Zhuxiu et al., 2011):

$$E_T(l,d) = E_{elec} \cdot l + \epsilon_{amp} \cdot l \cdot d^2 \tag{1}$$

where E_{elec} is the electrical energy that a node sends or receives 1-bit data and ϵ_{amp} is the transmit amplifier. The energy that a node needs to receive this packet is

$$E_R(l) = E_{elec} \cdot l \tag{2}$$

The network model is described as follows:

A WSN W can be described as:

$$W = \{(N_i, E_i)\}, \quad i \in [1, n] \tag{3}$$

where N_i denotes the current node and E_i denotes the energy of N_i. The total number of nodes in W is n.

Assume the neighbor nodes of N_i is

$$Neighbor_i = \{N_i(j), E_i(j)\}, \quad j \in [1, q_i] \tag{4}$$

where q_i is the number of neighbors of node N_i. The lifetime of a WSN is defined as

$$W_{lt} = \frac{\sum_{i=1}^{n} T(N_i)}{n} \tag{5}$$

where $T(N_i)$ is the lifetime that node N_i exhausts from the beginning.

At the beginning, the transmission radius of each node in the WSN topology is set to the same, r. Actually, r must satisfy that the connectivity of the entire WSN must be close to 1, which means the WSN should be full-connected (every node in the WSN topology must be directly connected or indirectly connected via intermediate nodes). A simple example of a full-connected WSN is

Table 2
Network parameters.

l	Packet size in bits
d	Transmission distance between 2 nodes
$E_T(l,d)$	The energy that a node transmits an l-bit long packet over distance d
E_{elec}	The electrical energy that a node sends or receives 1-bit data
ϵ_{amp}	Transmit amplifier
$E_R(l)$	The energy that a node needs to receive a l-bit packet
W	A WSN model
N_i	The current node
E_i	The residual energy of N_i
n	The total number of nodes in W
$Neighbor_i$	The set of neighbor nodes of N_i
q_i	The number of neighbors of node N_i
$T(N_i)$	The lifetime that node N_i exhausts from the beginning
W_{lt}	The lifetime of a WSN W

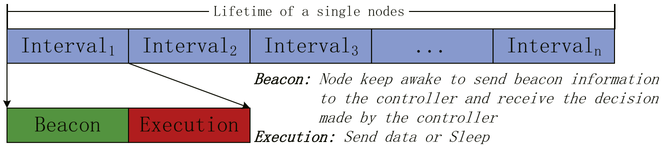

Fig. 3. Lifetime of a single node.

shows in Fig. 2 and detailed simulation parameters are shows in Table 2.

3.2. SDN-ECCKN algorithm

In the proposed SDN-ECCKN algorithm, the lifetime of a single node consists of multiple intervals. Each time interval can be divided into two slices: beacon slice and execution slice, which is shown in Fig. 3. In the beacon slice, each node keeps awake to send beacon information to the controller following the route in the initial network topology (since all nodes are awake). Because only the transmission energy consumption is considered, we assume that the full network topology has already been acquired by the controller at the beginning of our simulation. The energy cost of the full topology acquisition procedure can be neglected since the controller is usually connected to a constant power source. Different from the main data, beacon data has much smaller size which only includes the status of the nodes such as sleeping status, the node ID of the next hop and its residual energy. Then the controller calculates the sleeping status of each node and updates the routing topology of the global network according to these received beacon data by applying EC-CKN algorithm. Then the controller sends these decisions back only to the affected nodes including the nodes that will change their sleeping status or the nodes that will change their transmission routes in the execution slice. Finally, nodes in the network will choose to remain awake or go to sleep based on these received decisions or change their next-hop nodes to forward the main data.

Each interval can be described in Fig. 4 (*the following steps are applied for every node N_i).

We can see from Fig. 4 that SDN based architecture is applied to each node, which has the following advantages:

1. All the decisions are made by the controller which is usually connected with constant power supply.
2. Each node transmits its beacon data via certain route in the initial full topology and transmits its main data only to the next-hop node based on the decision made by the controller.
3. There is no data exchange (broadcasting) between nodes to calculate its own status during the whole interval. Hence, there is no broadcasting during the entire network lifetime, which dramatically reduces the total transmission times during the network lifetime.
4. Controller only transmits decision packet to the nodes whose sleep status or next-hop nodes will change.

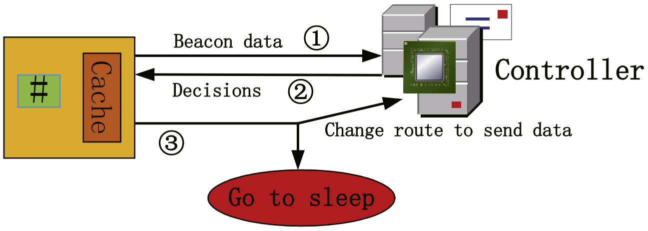

Fig. 4. Data transmission between node and controller in each interval.

Detailed differences between SDN-ECCKN and EC-CKN algorithm are summarized in Table 3.

From Table 3 it is clear to see that according to these differences, the proposed SDN-ECCKN algorithm removes the broadcasting procedures within each time interval. Each node transmits beacon data to the controller via certain route, which has already been pre-defined initially. There is no data exchange (broadcasting) between nodes during the entire lifetime of the network. Hence, the total transmission times during the network lifetime decrease which saves energy and prolongs the lifetime of the entire network.

4. Results and analysis

We simulate the entire network and test several important parameters which can directly reflect the performance of the network. The total simulation area is a square with 200 × 200 and the number of nodes is 150. Nodes are randomly deployed in this area and each node is allocated with the same energy at the beginning of the simulation.

Firstly, EC-CKN algorithm is applied to the randomly deployed network and then SDN-ECCKN algorithm is implemented in the same network topology.

4.1. Network residual energy

We compare network Residual Energy Rank after each time interval for EC-CKN and SDN-ECCKN algorithm under different K values. Figure 5 shows the simulation result. Residual Energy Rank is defined as

$$Residual\ energy\ rand = \frac{\sum_{i=1}^{n} E_i}{n \cdot E_{initial}} * 100\% \tag{6}$$

Table 3
Differences between SDN-ECCKN and EC-CKN algorithm.

Interval	Step	SDN-ECCKN	EC-CKN
Beacon	Begin	Get the information of current remaining energy	Get the information of current remaining energy
	①	Each node keeps awake to *send beacon data to the controller* following the route in the initial network topology	Each node *broadcasts the beacon data twice* to obtain the energy ranks of its current 1-hop and 2-hop neighbors
	②	Controller makes decision of sleep status of every node in the network according to the beacon data sent by nodes and updates the network topology based on the ECCKN algorithm. Then controller sends these decisions to all the affected nodes	Each node makes decision of its sleep status based on the energy ranks acquired by broadcasting the beacon data twice
Execution	③	Each node executes the decision: go to sleep or keep awake to send main data in each interval	Each node executes the decision: go to sleep or keep awake to send main data in each interval

*Run the above at each node.

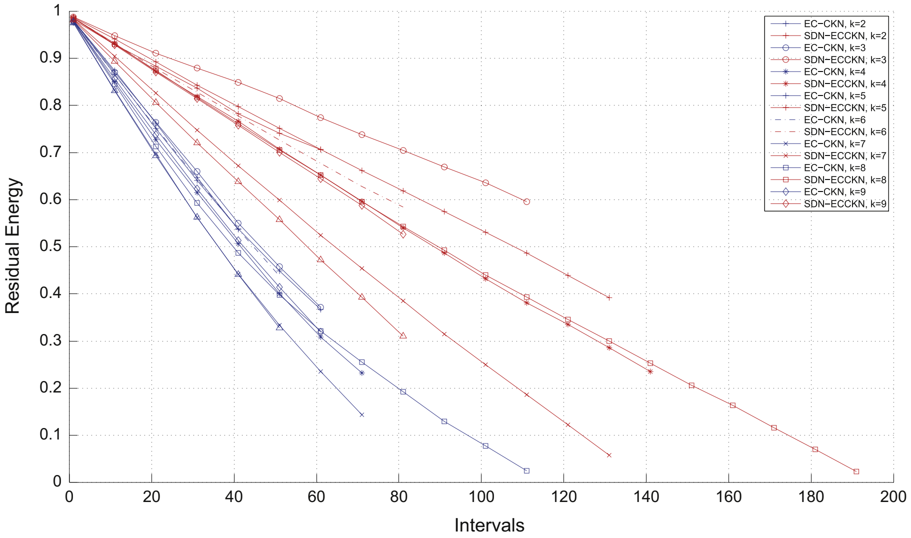

Fig. 5. Network residual energy after each interval under different K values.

where $E_{initial}$ is the initial energy allocated to each node at the beginning which is the same for each node.

By applying the proposed SDN-ECCKN algorithm, the total number of intervals experienced in the same network topology dramatically increase for all scenarios when $K = 2, 3, \ldots, 10$, which means the lifetime of the network increases as well. The residual energy after each interval by applying SDN-ECCKN algorithm is much higher than EC-CKN algorithm. In addition, the relationship between residual energy and the number of intervals is almost linear in SDN-ECCKN algorithm, which reveals that the energy consumption of each interval is almost the same, leading to the easier management of the network energy. Our SDN-ECCKN algorithm is able to increase the entire network lifetime efficiently.

4.2. Number of alive nodes

The number of alive nodes after each interval is another important parameter that can directly reflects the network performance. Simulation results in Fig. 6 apparently show the superiority of the SDN-ECCKN algorithm under different K values. After each interval, the number of alive nodes is obviously higher when it is applied with the proposed SDN-ECCKN algorithm. Even near the end of the network lifetime, the number of alive nodes is much higher, which indicates that the energy consumption in the network is well balanced at each node during the entire WSN lifetime.

4.3. Number of solo nodes

Solo nodes are the nodes that do not have any neighbors. Although solo nodes might have enough residual energy, these nodes are isolated from the network, which greatly affect the network connectivity and the balance of network energy consumption. Figure 7 shows the advantage of the proposed SDN-ECCKN algorithm with less solo nodes during the entire network lifetime under different K values. This advantage is even more distinct on the latter period of the network lifetime.

4.4. Network lifetime vs. K value

We also compared the entire network lifetime under different K values. The value of K greatly affects the network lifetime. However, the relationship between K and the network lifetime is nonlinear. Figure 8 depicts this relationship. Overtly, the proposed SDN-ECCKN algorithm achieves remarkable preponderance on network lifetime.

In summary, the proposed SDN-ECCKN algorithm has longer network lifetime, more alive nodes, and less solo nodes during the network lifetime under different scenarios when $K = 2, 3, \ldots, 10$.

5. Conclusion

In recent WSN research, network energy management has always been a difficulty. In this paper, a Sleep scheduling algorithm named SDN-ECCKN is proposed to address this issue. The key point of SDN-ECCKN is to adopt SDN based network architecture to the network instead of traditional WSN architecture for the purpose of reducing the total transmission times during the network lifetime. The famous ECCKN algorithm is regarded as the fundamental algorithm because of its unique feature that direc-

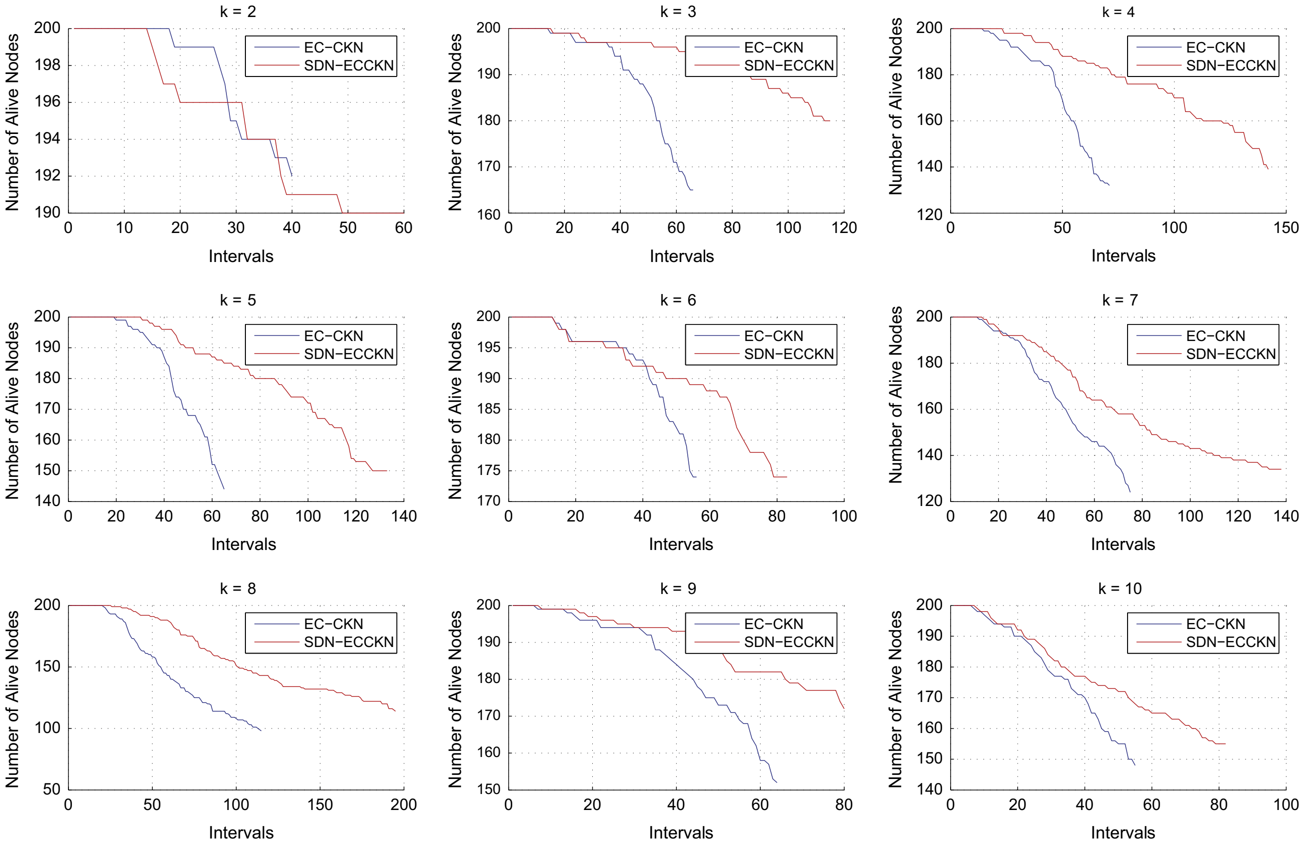

Fig. 6. The number of alive nodes after each interval under different *K* values.

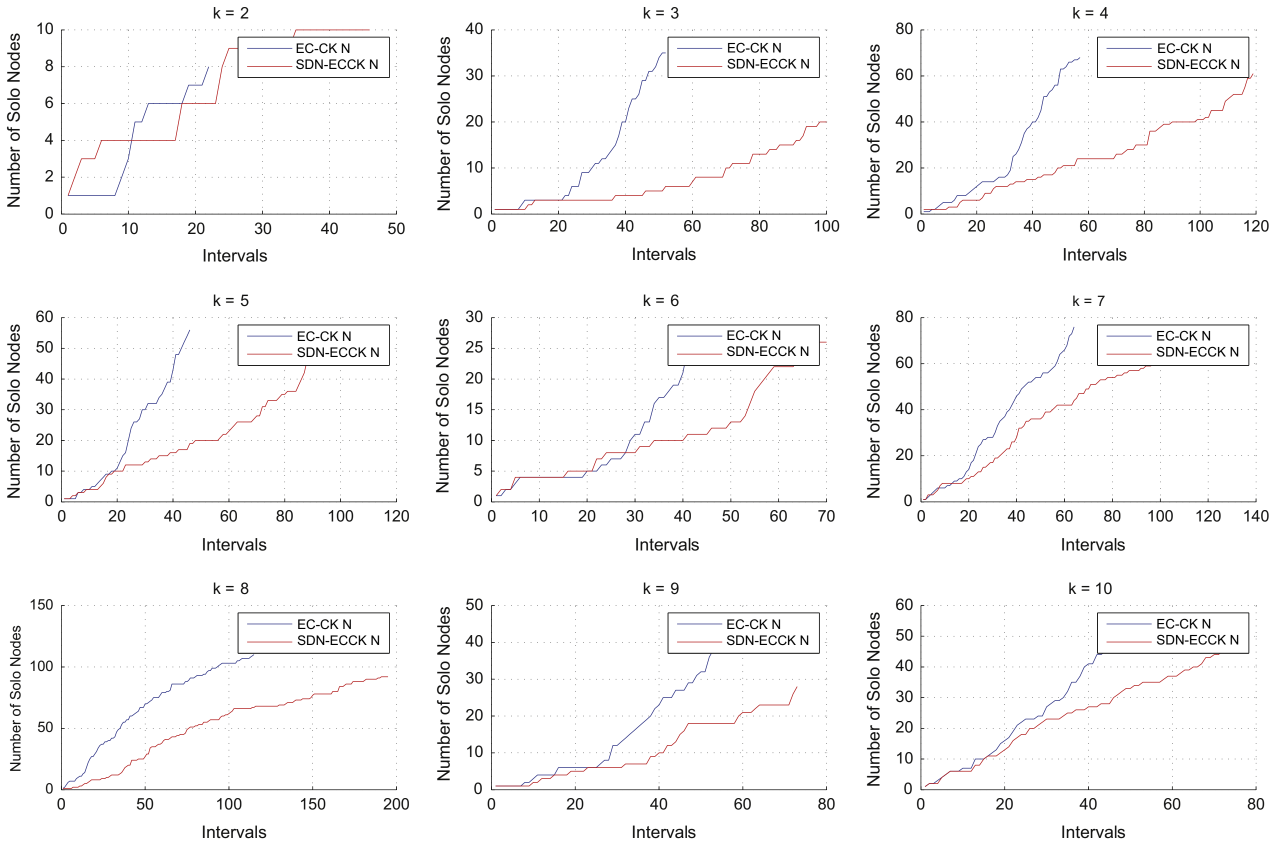

Fig. 7. The number of solo nodes after each interval under different *K* values.

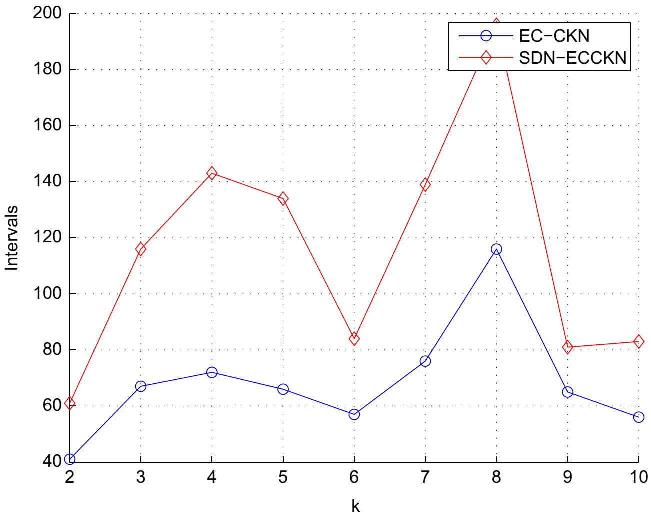

Fig. 8. The relationship between *K* value and network lifetime.

tly reflects the energy consumption of the entire network. The detailed network model and relevant energy parameters are presented as well. Finally, by applying the proposed SDN-ECCKN algorithm, the network performance has been prominently reinforced. Our simulation results show the great improvement when compared to the EC-CKN algorithm especially in network lifetime, number of alive nodes and number of solo nodes. In future work, how to apply our SDN-ECCKN algorithm in dynamic WSNs will be further investigated.

Acknowledgments

This work is supported by the 2014 Pearl River S&T Nova Program of Guangzhou (No. 2014J2200023), the Natural Science Foundation of Guangdong Province (No. 2014A030313685), the Open Fund of Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis (No. GDUPTKLAB201304), the 2014 Shenzhen City “Knowledge Innovation Program” Project (No. JCYJ20140417113430604), the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201406), the 2014 Foshan Science and Technology Project (No. 2014HK100103), the National Natural Science Foundation of China (Grant no. 61401107), the 2013 Special Fund of Guangdong Higher School Talent Recruitment, the Educational Commission of Guangdong Province, China Project (No. 2013KJCX0131), the special funds of Guangdong high-tech development project (No. 2013B010401035), and the Guangdong University of Petrochemical Technology's Internal Project no. 2012RC0106.

References

Al-Turjman S, Fadi M, Hossam S Hassanein, Mohamad Ibnkahla. Quantifying connectivity in wireless sensor networks with grid-based deployments. Journal of Network and Computer Applications 2013;36(1):368–77.

Baccour N, Koubaa A, Mottola L, Zuniga MA, Youssef H, Boano CA, et al. Radio link quality estimation in wireless sensor networks: a survey. ACM Trans Sensor Netw TOSN 2012;8(4):34.

Banikazemi M, David O, Anees S, John T, Guohui W. Meridian: an sdn platform for cloud network services. Communications Magazine, IEEE 2013;51(2):120–7.

Chih-fan H, Mingyan L. Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms. In: IPSN 2004. Third international symposium on information processing in sensor networks; 26–27 April 2004. p. 433–42.

Hata H. A study of requirements for SDN switch platform. In: International symposium on intelligent signal processing and communications systems (ISPACS); 12–15 November 2013. p. 79–84.

Joel JPC Rodrigues, Paulo ACS Neves. A survey on IP-based wireless sensor networks solutions. Int J Commun Syst 2010;23(8):963–81.

Kai Lin, Min Chen, Sherali Zeadally, Joel JPC Rodrigues. Balancing energy consumption with mobile agents in wireless sensor networks. Future Gener Comput Syst —Int J Grid Comput eSci 2012;28(February(2)):446–56.

Kimura N, Latifi S. A survey on data compression in wireless sensor networks. In: ITCC 2005. International conference on information technology: coding and computing, vol. 2; 4–6 April 2005. p. 8–13.

Lara A, Kolasani A. Communications Surveys & Tutorials, IEEE 2014;16(1):493–512 First Quarter 2014.

Lee Chao-Yang, Liang-Cheng Shiu, Fu-Tian Lin, Chu-Sing Yang. Distributed topology control algorithm on broadcasting in wireless sensor network. Journal of Network and Computer Applications 2013;36(4):1186–95.

Li LE, Zhuoqing MM, Jennifer R. Toward software-defined cellular networks. In: European workshop on software defined networking (EWSDN). IEEE; 2012. p. 7–12.

Luís ML Oliveira, Joel JPC Rodrigues, André GF Elias, Guangjie Han. Wireless sensor networks in IPv4/IPv6 transition scenarios. In: Wireless personal communications, vol. 78, No. 4. Springer; October 2014. p. 1849–62.

McKeown N, Tom A, Hari B, Guru P, Larry P, Jennifer R, et al. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput Commun Rev 2008;38(2):69–74.

Nath S, Gibbons PB. Communicating via fireflies: geographic routing on duty-cycled sensors. In: IPSN 2007. Sixth international symposium on information processing in sensor networks; 25–27 April 2007. p. 440–9.

Xiangxin K, Zhiliang W, Xingang S, Xia Y, Dan L. Performance evaluation of software-defined networking with real-life ISP traffic. In: IEEE symposium on computers and communications (ISCC); 7–10 July 2013. p. 541–7.

Yaxiong Z, Jie W. Stochastic sleep scheduling for large scale wireless sensor networks. In: 2010 IEEE international conference on communications (ICC); 23–27 May 2010. p. 1–5.

Zhangbing Zhou, Jine Tang, Liang-Jie Zhang, Ke Ning, Qun Wang. EGF-Tree: an energy efficient index tree for facilitating multi-region query aggregation in the Internet of things. Person Ubiquitous Comput 2014a;18(4):951–66.

Zhangbing Zhou, Deng Zhao, Xiaoling Xu, Chu Du, Huilin Sun. Periodic query optimization leveraging popularity-based caching in wireless sensor networks for industrial IoT applications. Mobile Netw Appl 2014b;20(2):124–36.

Zhu CS, Yang LT, Shu L, Duong TQ, Nishio S. Secured energy-aware sleep scheduling algorithm in duty-cycled sensor networks. In: 2012 IEEE international conference on communications (ICC); 10–15 June 2012. p. 1953–7.

Zhu CS, Yang LT, Shu L, Leung VCM, Rodrigues JJPC, Wang L. Sleep scheduling for geographic routing in duty-cycled mobile sensor networks. IEEE Trans Ind Electron 2014;61(November (11)):6346–55.

Zhuxiu Y, Lei W, Lei S, Hara T, Zhenquan Q. A balanced energy consumption sleep scheduling algorithm in wireless sensor networks. In: Seventh international conference on wireless communications and mobile computing conference (IWCMC); 4–8 July 2011. p. 831–5.