



# A software defined network routing in wireless multihop network



Junfeng Wang<sup>a</sup>, Yiming Miao<sup>b</sup>, Ping Zhou<sup>b,\*</sup>, M. Shamim Hossain<sup>c</sup>, Sk Md Mizanur Rahman<sup>d</sup>

<sup>a</sup> School of Information Engineering, Zhengzhou University, Zhengzhou, China

<sup>b</sup> School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

<sup>c</sup> Department of Software Engineering, College of Computer and Information Sciences (CCIS), King Saud University, Riyadh 11543, Saudi Arabia

<sup>d</sup> Department of Information Systems, College of Computer and Information Sciences (CCIS), King Saud University, Riyadh 11543, Saudi Arabia

## ARTICLE INFO

### Keywords:

Software defined network(SDN)  
Wireless multihop network  
Routing  
Multipath

## ABSTRACT

SDN has been touted as one of the most promising solutions for future Internet with the innovative design ideas that the control plane is logically centralized and decoupled from the data plane. Current research on SDN mainly focuses on wired network and data center, while software-defined wireless multi-hop network is put forth in a few researches, but only at stage of putting forth models and concepts. In this paper, we propose a novel routing protocol applied SDN in wireless multi-hop network. The implementation of the protocol is given in detail, and OPNET is used to build the model and carry on the simulation experiment. A large number of simulation experiments are performed to compare the key parameters of different networks. Simulation results show that our proposed routing protocol provide shortest path and disjoint multipath routing for nodes, and its network lifetime is longer than existing algorithms (OLSR, AODV) when traffic load reaches a certain value.

## 1. Introduction

Because of node energy depletion and environment noise and other obstacles, the high dynamic characteristics of wireless link cause poor quality and low stability for link, which poses a challenge to throughput and transmission reliability of multi-hop wireless network. Otherwise, restricted energy and mobility requirements of node also bring difficulties to design and optimization of routing protocol.

Traditional multi-hop wireless routing is divided into active routing and passive routing. Active routing such as OLSR (Clausen and Jacquet, 2003) is based on broadcast information in each node, the routing information from that node to all other nodes is saved, which incurs a large memory overhead. Therefore, active routing is not adapted to high dynamic network. As for passive routing such as AODV (Perkins and Royer, 2010; Galzarano et al., 2013), it is using flooding broadcast for the establishment and maintenance of routing. Therefore the waste of network resources is larger.

Recently, software-defined networking (SDN) (Xia et al., 2015) has been touted as one of the most promising solutions for future Internet. In SDN, the control plane is logically centralized and decoupled from the data plane, and open uniform interface (such as OpenFlow) is adopted for interaction (Chen et al., 2015a, 2016c). Control layer is responsible for programming to manage & collocate network, to deploy new protocols and etc. Through centralized control of SDN,

an overall view of the network topology can be obtained for the controller, and dynamic allocation may be conducted to network resources when network flow changes. Currently, most studies of routing for software-defined network are with respect to wired network and data center (Chen et al., 2013; Liu et al., 2016), though software-defined Internet of Things (Fortino and Trunfio, 2014) and software-defined wireless sensor network (Aloi et al., 2015) are put forth in a few researches, but only at stage of putting forth models and concepts.

In researches on software-defined multi-hop wireless network, the characteristics of wireless network, such as broadcast characteristics, hidden terminal, node mobility and etc shall be taken into consideration. OpenFlow is only applicable to route selection, and it can not manage more functions such as aperiodic data collection, duty cycle of sensor node.

Some researchers propose transforming sensing node, for instance, the concept of Flow-Sensor and utilization of OpenFlow between Flow-Sensor and controller is put forth in Mahmud and Rahmani (2011). Realization of SDN sensor based on Microcontroller Units (MCUs) and Field-Programmable Gate Arrays (FPGAs) with super low power consumption is put forth in Miyazaki et al. (2014). In some researches, the framework of software defined wireless sensor network (SD-WSN) and Sensor OpenFlow Protocol (Luo et al., 2012) that applies in WSN are proposed. Lightweight IP Protocol such as uIP or uIPv6 based on Contiki operating system are utilized in WSN. From the point of

\* Corresponding author.

E-mail addresses: [iewangjf@zzu.edu.cn](mailto:iewangjf@zzu.edu.cn) (J. Wang), [yiming.epic@qq.com](mailto:yiming.epic@qq.com) (Y. Miao), [pingzhou.cs@qq.com](mailto:pingzhou.cs@qq.com) (P. Zhou), [mshossain@ksu.edu.sa](mailto:mshossain@ksu.edu.sa) (M.S. Hossain), [mizan@ksu.edu.sa](mailto:mizan@ksu.edu.sa) (S.M.M. Rahman).

<http://dx.doi.org/10.1016/j.jnca.2016.12.007>

Received 5 August 2016; Received in revised form 23 November 2016; Accepted 2 December 2016

Available online 06 December 2016

1084-8045/ © 2016 Elsevier Ltd. All rights reserved.

application fields, there are campus WLAN (Lei et al., 2014), VANET (Ku et al., 2014; Carlos et al., 2012), BAN (Chen, 2014; Chen et al., 2015b), network between mobile base station and base station controller (Amani et al., 2014), WSN (Jayashree and Princy, 2015; Fortino et al., 2012, 2013), MAC layer (De Gante et al., 2014; Stefano Galzarano and Liotta, 2013; Stefano Galzarano et al., 2014) in WSN, novel mobile services and applications (Zhang et al., 2014; Chen et al., 2015c; Zhang, 2016) and etc.

The common problem for above researches is that only concepts and simple models are put forth, and simulation is simple or even not realized. The description on detailed design and realization algorithms of controller for SDN routing is relatively obscure, and there is no systematic description or realization (Li and Chen, 2015). In this paper, a multi-hop wireless network routing protocol is designed and realized (Chen et al., 2016b), detailed description is conducted to the realization process of protocol, model is established and simulation verification is conducted with OPNET (Chen et al., 2016a). The contributions of this document are as follows:

### 提出了一种多跳网络路由协议

- A multi-hop wireless routing protocol based on SDN is proposed, the controller has a global view of the network and provides single-path routing or multipath routing for other nodes.
- The residual energy of nodes in controller is updated in real time, and the shortest path is generated based on energy and hop count.
- The generation algorithm for disjoint multipath from source to target is put forth. 提出了算法

The other parts of this paper are arranged as below: routing protocol scheme is introduced in Part 2, simulation verification is conducted in Part 3, and Part 4 summarize the whole paper.

## 2. Routing scheme

Exclusive SDN controller node (hereinafter “controller” for short) is added in network. Controller broadcasts information to each sensing node, normal node sends node information to controller, controller generates the global network view as per information of normal nodes. When source node requests the transmission path from the controller, controller calculates the shortest path with Dijkstra algorithm and sends the path information to source node. The premise of routing design is that nodes in network are not aware of their locations, that controller is located in middle of network and not restricted by energy, and that source node and target node in network are not fixed at certain node. 前提假设

### 2.1. Routing process design

The flow diagram of routing protocol is shown in Fig. 1, and the specific description is as below:

1. Controller broadcasts information to each sensing node, normal node forms the backward path to controller as per broadcast path;
2. Normal node sends node information (residual energy, neighbor nodes) to controller through backward path, and controller establishes network topology as per node information received;
3. When source node is to send data without path to target node, it shall send routing information request to controller;
4. Controller calculates the shortest path from source to target (based on hop count and residual energy) as per network-wide view and Dijkstra algorithm, then sends path information to source node;
5. Source node sends data to target node as per path information;
6. When the change of information of neighbor node is discovered by some node, they would report it to controller; 动态环境下开销大
7. When target node have data received, statistical information should be reported to controller periodically.

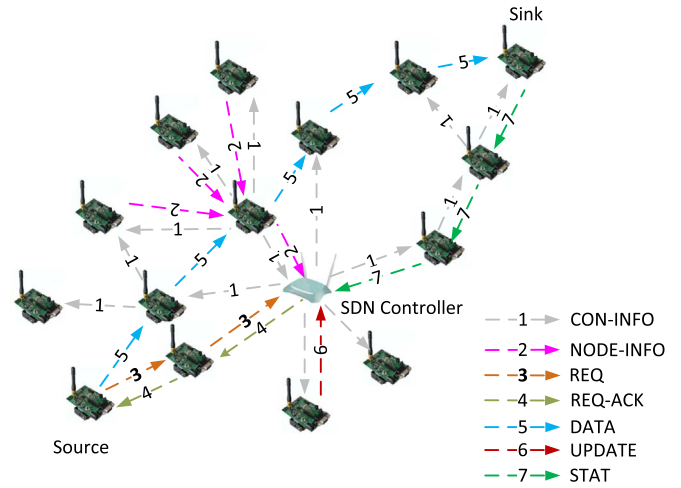


Fig. 1. Schematic diagram of protocol flow.

The information on DATA package to be used in routing algorithm is shown in Table 1.

### 基于剩余能量和跳数计算最短路径

#### 2.2. Storage structure for protocol

##### 2.2.1. Controller

##### 二维邻居矩阵和剩余能量矩阵

Global view shall be saved by controller node. Assume the number of normal nodes in network is  $n$ , then two-dimensional array  $NodeInfo[n][n]$  is required to record global view. When node  $i$  is neighbor of node  $j$ , the value of  $n[i][j]$  and  $n[j][i]$  is 1, or the value is infinity. Another array  $ResEnergy[n]$  is needed for controller to save residual energy of each node.

##### 2.2.2. Normal node

Normal nodes require two kinds of information: list of neighbor information and routing list.

List of neighbor information is formed with periodical Hello information exchange. During initialization, node needs to report self information and neighbor information to controller. If broadcast algorithm with greedy neighbors is selected, information for 2-hop neighbors shall be saved by node.

Routing list records the next hop routing information between certain node to target node. It includes serial number (SN), target node, next hop and hop count. When node receives controller package or request package from source node, the backward routing to controller or source node shall be recorded. If node A is to send data to node B, but there is no target routing path in routing list, then node A needs to request routing from controller, the REQ-ACK package of controller includes hop count for one or multiple disjoint path(s) and list of relay nodes, which shall be saved by source node A. The schematic diagram for structure of routing list is shown in Fig. 2.

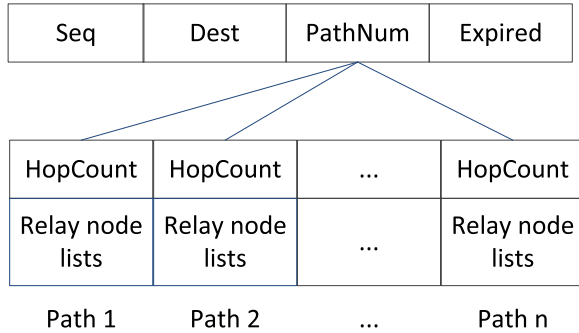
#### 2.3. Controller broadcast

In order to clearly define path to controller for nodes in network, controller broadcasts packages in advance. Nodes establish backward routing as per control package received from controller. After receiving a broadcast package, one node shall check whether it has received that package before as per SN, if that broadcast package is new, that node would broadcast it. If that node has received that package before, then there would be no broadcast, but the hop count would be updated at that node.

Simply flooding broadcast package in network would cause problems such as rebroadcast & redundancy, signal collision, broadcast storm and etc. These problems would be more outstanding especially when network nodes are relatively dense. Generally, multi-hop wireless

**Table 1**  
Packages for routing protocol.

Package	Main Fields of Package	Send Mode	Pathway
CON-INFO	Seq, Previous Hop, Hop Count	Broadcast	Controller → Other node
NODE-INFO	Residual Energy and All Neighbors of Node	Unicast	Other node → Controller
REQ	Seq, Path Number, Source Node, Target Node	Unicast	Source → Controller
REQ-ACK	Seq, Source Node, Target Node, List of Relaying Nodes, Expiration Time	Unicast	Controller → Source
DATA	Seq, Source Node, Target Node, List of Relaying Nodes, Sensing Information	Unicast	Source → Sink
UPDATE	Residual Energy and List of Neighbors	Unicast	Other node → Controller
STAT	Source Node, Target Node, Package Size, List of Relaying Nodes	Unicast	Sink → Controller



**Fig. 2.** Schematic diagram for list of multipath source routing.

只有部分节点对控制器的广播包进行广播

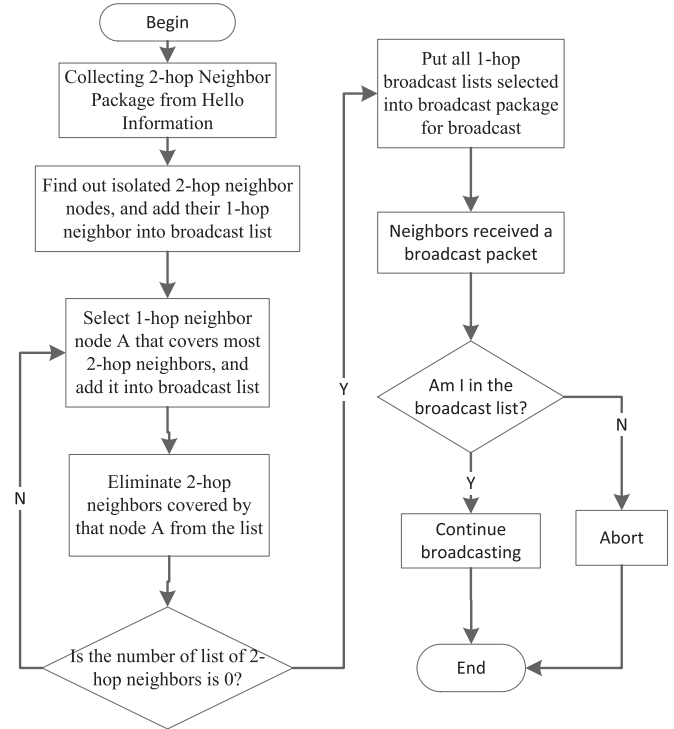
network is deployed densely, and there are a lot of redundant nodes, and system bears stronger fault-tolerant performance. If only a part of nodes are selected for rebroadcast on premise that all nodes should receive broadcast, the problem of broadcast storm would be relieved.

At present, there are a variety of researches that aim to solve the problem of broadcast storm, including algorithms based on probability, counter, distance, location, neighbor information and etc. As for probability-based method (Lichtblau and Dittrich, 2014), nodes conduct broadcast based on certain probability. However, this method could not be adapted to change in node density, if the node density is low, the area covered by broadcast decreases. As for counter-based algorithm (Chen et al., 2003), after the number of broadcast received by a node exceeds a certain threshold, the broadcast at that node would be canceled. This algorithm is not influenced by node density in network, but there is much broadcast delay. As for broadcast algorithm based on neighbor information, a part of nodes are selected for broadcast as per neighbor information. This kind of broadcast algorithm need to save neighbor information for nodes.

In the algorithm based on neighbor information, the algorithm where MPR nodes are selected by OLSR routing is taken into our reference, the neighbors of a part of nodes are selected for broadcast. 1-hop and 2-hop neighbors of some nodes are utilized in this algorithm. As for this algorithm, neighbor information should be added into Hello information, and a node obtains list of 2-hop neighbors through Hello information of neighbor nodes, the processes of this algorithm are shown in Fig. 3.

Simulations are conducted for 4 algorithms(3 broadcast methods and full-node broadcast), and a large times of simulation experiment are performed to figure out mean value. The performance comparison results of the four methods are shown in Table 2.

There are 800 nodes in total in simulation network, the number of nodes in full-node broadcast is the number of total nodes, while the number of broadcast in the other 3 methods is largely reduced, thereinto, the number in counter method is more than that in greedy neighbor method but less than that in probability method. It can be seen that the less the number of broadcast is, the longer the network lifetime is. What should be noticed is that as for probability method and counter method, if different parameters are set up, the results are different; if the probability set up in probability method is larger, or if the threshold set up in counter method is larger, the number of



**Fig. 3.** Selecting next hop of broadcast node with greedy algorithm.

**Table 2**  
Performance comparison of four broadcast methods.

Method	Number of Broadcast	Lifetime (s)	Parameter
All nodes	800	744	p=0.3 threshold=3, waittime=0.02 s
Probability	233	773	
Counter	201	784	
Greedy Neighbor	159	798	

broadcast is larger. The parameters for probability method and counter method in the table are values with better performance in experiment.

During actual simulation, even greedy neighbor algorithm has multiple redundancies, because overlap exists for greedy neighbor of multiple nodes in transmission distance after multiple hops, and there is still margin for reduction.

Node forms the backward path to controller as per broadcast package received, and sends NODE-INFO package along the backward path. If the information of each node is sent separately along the backward path, then midway node could finish sending information of downstream node through sending for many times. In this paper, it is designed that the upstream node shall combine information of all next-hop nodes for sending, after information of downstream node arrives at upstream node. The node information package is shown in Fig. 4.



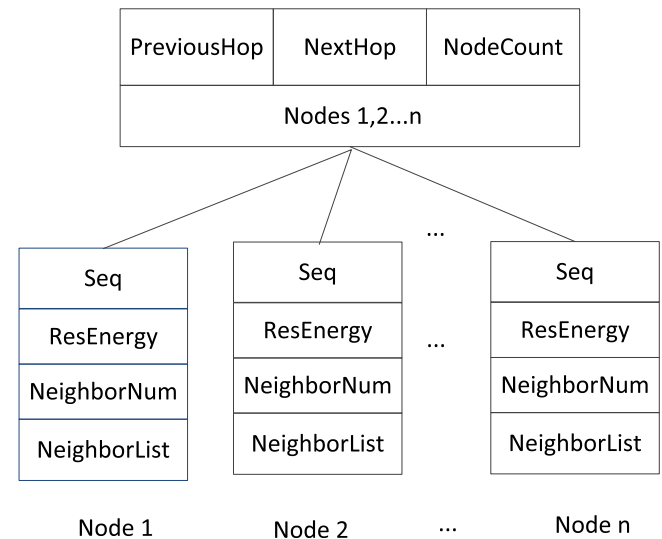


Fig. 4. Structure of NODE-INFO package.

Table 3

Performance comparison before and after combination.

Method	Lifetime (s)	EnergyPerPacket (mw/packet)
Before Combination	79.8	0.933
After Combination	83.3	0.812

After receiving SDN broadcast package for a node, there is certain delay for sending NODE-INFO package. It is designed that the delay time of node is inversely proportional to hop count of the node to controller. The larger the hop count is, the shorter the delay for sending node information package is. Therefore, the information of nodes located at the edge would be reported in advance, gradually converging from the edge to the center.

When a node is sending NODE-INFO package, energy consumption required shall be firstly estimated as per size of information package received and size of information package to be sent. Residual energy would be obtained by deducting estimated energy consumption from

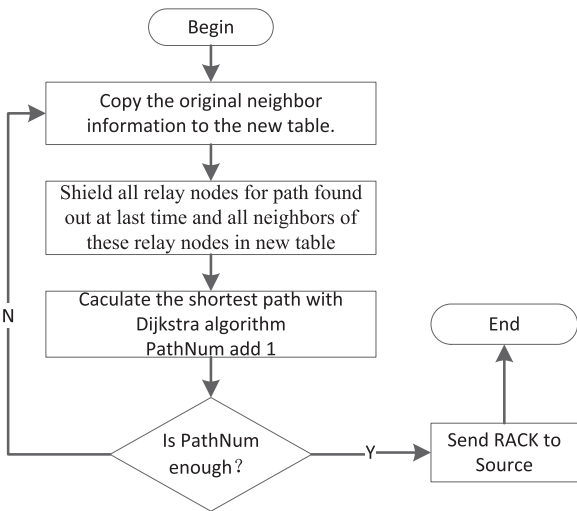


Fig. 6. Multipath routing generation flow chart.

node energy, then information on residual energy and neighbors of that node shall be sent in the end of NODE-INFO package.

What should be noticed is that the size of package after combination could not exceed the maximum limit of MAC layer, or that package would be abandoned. Therefore, large package shall be sent after splitting. The size of information package is related to node density and transmission distance of node, and the largest size of information package could be estimated. Assume a simulation scenario, the area of simulation scene is 1000 m\*500 m, with 800 nodes, and the transmission distance of node is 70 m, the size of each field is 16 bits, the size of maximum package in MAC layer is 18,432 bits, the estimation method is as below:

1. Node density=number of nodes/scene area=800/(1000\*500)=0.0016 pieces/m<sup>2</sup>;
2. The estimated number of node neighbors= $\pi \times \text{transmission radius}^2 \times \text{node density} = 3.14 \times 70^2 \times 0.0016 = 24.6 \approx 25$ ;
3. The size of information of each minor node=minor node area \* 16=(3+25)\*16=448 bit;

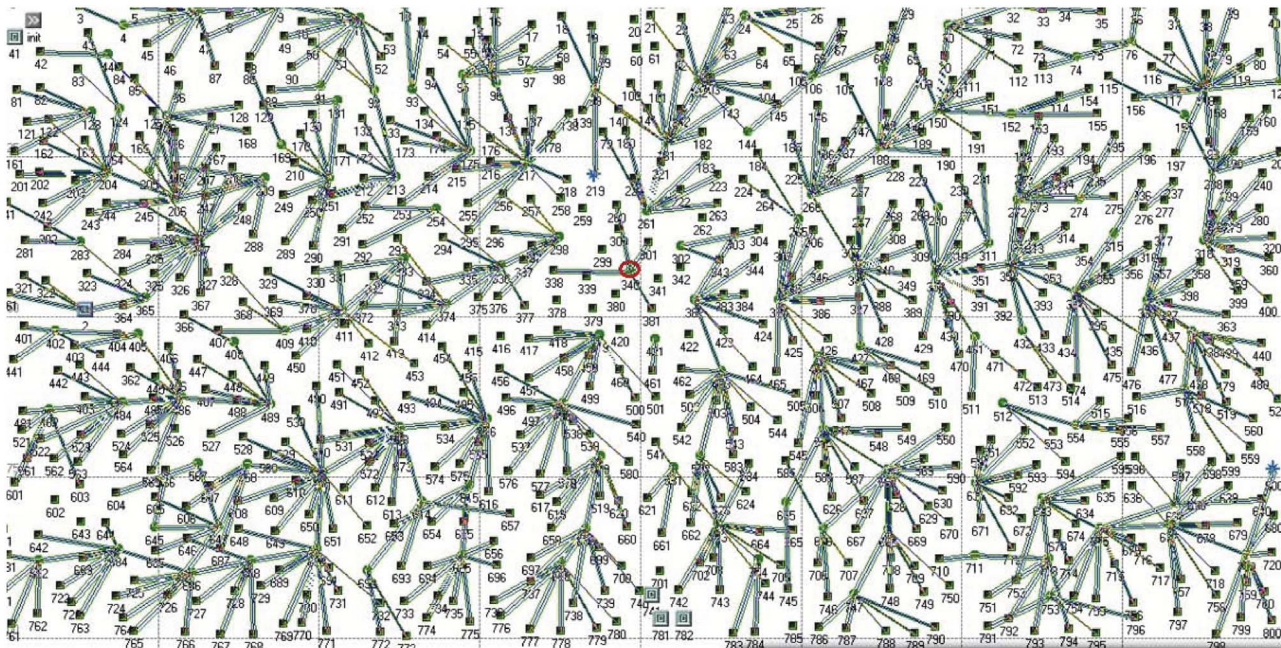


Fig. 5. Simulation screenshot for combined node information.

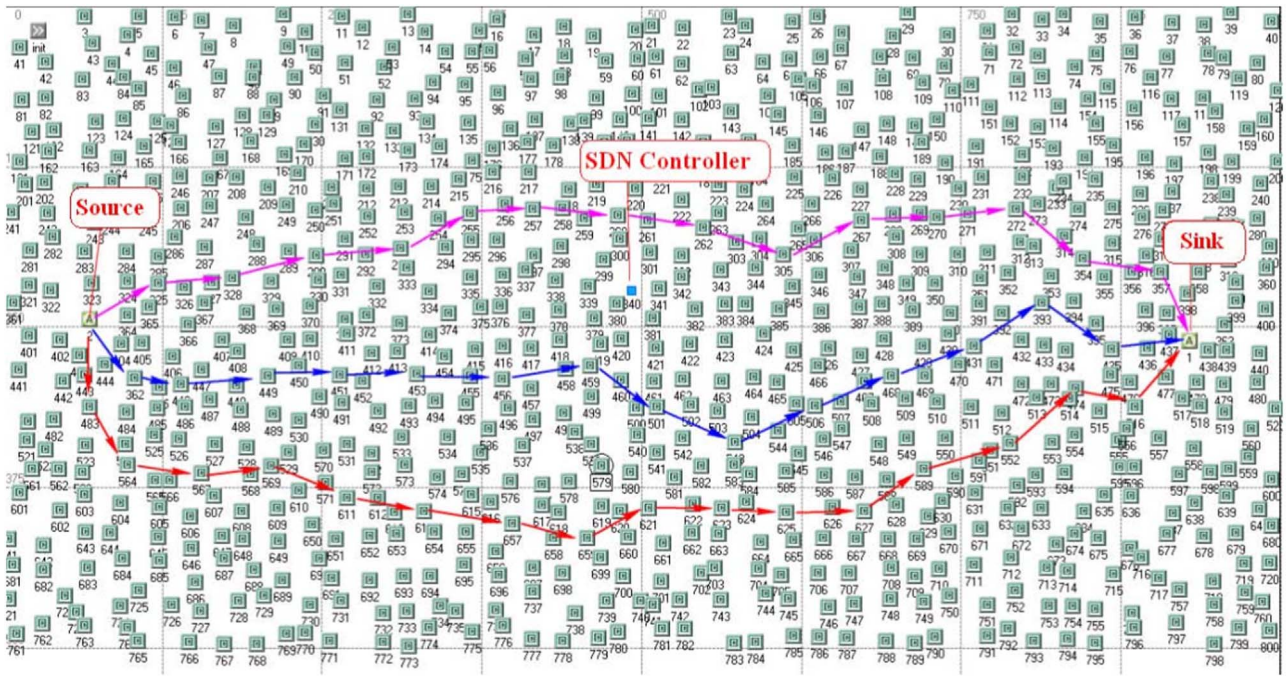


Fig. 7. Schematic diagram of disjoint multipath routing.

**Table 4**  
Simulation parameter.

Parameter	Value	Parameter	Value
Network Size(m)	1000*500	Simulation Time(s)	100
Node	200–900	Transmission Rate(Mbps)	1
Mac Protocol	IEEE 802.11b	Packet Size(Byte)	1024
Beacon interval(s)	2	Route expiration interval (s)	30

**Table 5**  
Parameter of energy power.

Parameter	Value
Initial Energy of Common node	2 w
Initial Energy of Controller	infinite
Transmission Power	0.660 mw/s.bit
Reception Power	0.395 mw/s.bit
Overhearing Power	0.195 mw/s.bit
Idle Power	0.035 mw/s.bit

**Table 6**  
Simulation test.

Parameter	Network density	Connections
Node	200–900 step by 100	800
Connection	1	2–10 step by 2

4. Number of minor nodes that can be accommodated=maximum permissible bits/number of bits at each minor node=18432/448=41.1.

During actual simulation, it is set up that information of 40 minor nodes would be sent at most. After combination, the relay node's frequently sending DATA package could be avoided, and energy consumption could be reduced. Table 3 shows contrast on network lifetime and energy consumption before and after combination. The simulation screenshot is shown in Fig. 5, it can be seen that node combines information of downstream node for sending to upstream

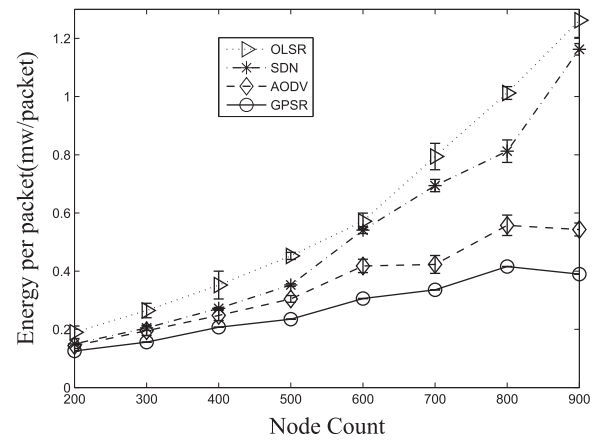


Fig. 8. Contrast on energy consumption and hop count for each package in different network size.

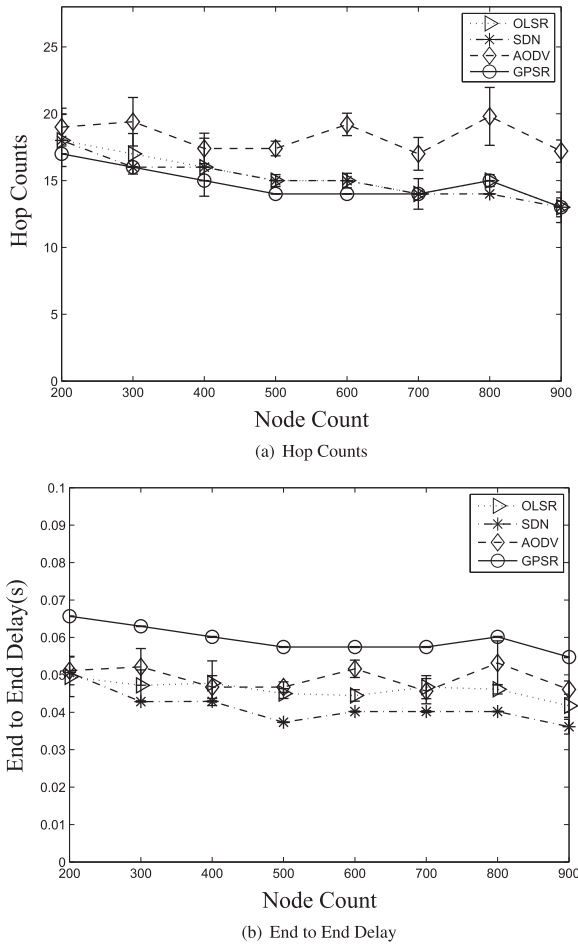
node, and within the red circle in the figure is controller node.

After controller receives NODE-INFO package, node information shall be saved into array of node information list, and residual energy of node shall be saved into array of residual energy. Thus the controller has a global view of network, and is able to provide routing for other nodes.

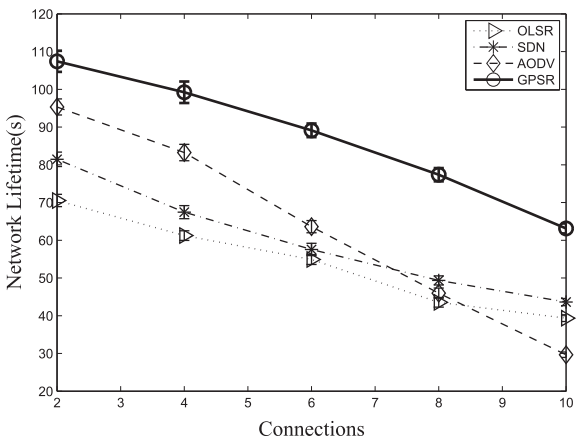
#### 2.4. Request and ACK of node's routing

If node A is to send data to node B, but there is no routing to node B in routing list, then node A shall send routing request to controller. The information of REQ package includes: SN, source node, target node and number of path requested. After receiving REQ package, relay node shall record the backward path to source node. When controller finishes calculating a shortest path or multiple disjoint multi-path routing, it generates REQ-ACK package and forwards this package back to source node. The algorithm for generation of shortest path and disjoint multi-path routing would be illustrated below:





**Fig. 9.** Contrast on mean hop count and delay in different network size. (a) hop counts (b) end to end delay.



**Fig. 10.** Contrast on network lifetime in case of multiple connections.

**Table 7**  
Disjoint multipath of SDN.

PathNum	Delay (s)	HopCount	Energy (mw/pk)	Lifetime (s)
1	0.038388	14.33333	0.80373	83.44031
2	0.062044	15.66667	1.26562	76.12225
3	0.083221	17.33333	1.476499	66.44215
4	0.097444	18.66667	1.647883	54.46031

#### 2.4.1. Caculation of shortest path

After receiving REQ package, controller shall run Dijkstra algorithm of shortest path to calculate the path from source node to target node. Here two parameters (hop count and energy) are adopted for measurement. Assume node  $j$  is neighbor of node  $i$ , and metric function  $f(j)$  of node  $j$  with respect to node  $i$  is shown in Eq. (1).

$$f(j) = \begin{cases} 1 - \frac{E_r(j)}{E_t} & j \text{ is neighbor of } i, \\ 0 & j \text{ isn't neighbor of } i. \end{cases} \quad (1)$$

Thereinto,  $E_r(j)$  stands for residual energy of node  $j$ , and  $E_t$  stands for primary energy of node. The larger the residual energy of node is, the smaller  $f(j)$  is, and the higher the possibility where node  $j$  is selected as forwarding node is. Thus, Dijkstra could calculate the shortest path as per comprehensive measurement on energy and hop count.

The problem here is that controller needs to know residual energy of node in time, the energy of node may be known at initialization of node, and residual energy of node may also be collected and estimated by controller as per UPDATE package and STAT package of node.

#### 2.4.2. Disjoint multipath routing

When source node requests multi-path routing to target node from controller, Dijkstra algorithm shall be invoked by controller for many times as per number of routing requested. The processes are shown in Fig. 6. In order to generate uncrossed disjoint multi-path routing, the forwarding nodes for paths generated and all neighbors of forwarding nodes (except second-hop nodes or nodes 2-hop away from the target) need to be shielded, Fig. 7 is the effect picture for 3 disjoint multi-path generated.

Lest the energy of node should be exhausted prematurely because node sends data with fixed path, expiration time is set up in REQ-ACK package. When routing expires, the information on routing to target node saved in routing list shall be deleted at source node.

#### 2.5. Data transmission

After receiving REQ-ACK package of controller, source node shall save it into routing list, and then send data to target node. Source node shall encapsulate all relay nodes into the package for sending to next-hop node. After receiving package, relay node shall record the routing to target node, and find out next-hop forwarding node in DATA package for continuous sending. After receiving data package, target node shall count the information such as hop count, delay and etc.

#### 2.6. Update of node information

One node shall check the change in neighbor node from periodical Hello information, there would be changes in neighbor information of multiple nodes in case of movement or invalidation of node. These nodes shall send UPDATE package to controller. After receiving UPDATE package, controller shall update array of connectivity and of residual energy of nodes.

#### 2.7. Statistical information

In order to conduct real-time statistics for number of information sent in network and residual energy of node, after receiving DATA package, target node shall conduct statistics for information such as hop count, delay, size, list of relay nodes and etc. of that package, and start timer for periodical reporting to controller. After receiving reports, controller may estimate residual energy for all nodes in the path as per information in STAT package, thus to provide basis for subsequent routing decision.

### 3. Routing simulation

Model is established with OPNET, and simulation is conducted. The contrast among four routing protocols (AODV, OLSR, our SDN routing and GPSR (Karp and Kung, 2000)) are made, where GPSR is introduced for contrast of shortest path routing (here the energy consumption of the GPSR for acquiring location information is ignored). In order to increase dependability of simulation, each simulation parameter is operated for 20 times.

#### 3.1. Simulation parameters and scenario

Simulation parameters are shown in Table 4, and energy consumption parameters are shown in Table 5.

Two kinds of simulation scenes are taken into consideration, and changes are made to network size and connection number, as shown in Table 6. (1) Consider data transmission between source node and target node, and change node density; (2) Under the circumstance where network size is invariant, utilize multiple connections to send data.

#### 3.2. Simulation results

##### 3.2.1. Different node density

The contrast among values of energy consumption for each package is shown in Fig. 8, it can be seen that the energy consumption for each package becomes higher as node density increases. As for SDN routing, the energy consumption is larger due to information exchange between controller and nodes, but the value for SDN routing is smaller than that for OLSR. In traditional routing protocol, the energy consumption for OLSR is higher because the network throughput required to construct routing at preliminary stage is higher. AODV also needs to form routing through broadcasting, so its energy consumption for each package is ranked the third. GPSR does not require broadcast, it only calculates and seeks next-hop forwarding node as per coordinates of neighbor nodes, so its energy consumption is the lowest.

Fig. 9 shows the contrast on hop count and delay among different algorithms. It can be seen that the higher the node density is, the more the number of forwarding nodes that may be selected is. One node may select the next-hop node that is more suitable for forwarding, thus the hop count decreases as node density increases. AODV could not provide optimal hop count because it does not have global view, the hop count is higher and unstable as well. However, as for OLSR and SDN, the shortest path could be calculated, thus their hop counts are close to that of GPSR. It can be seen from the delay figure that delay decreases as node density increases. As for each hop of GPSR, time is needed to calculate the next-hop neighbors, so its delay is the longest. Because the hop count of AODV is higher, so the delay is longer. Because SDN is constructed as per the shortest path, and forwarding nodes are put into DATA package that is available for direct reading and forwarding, the end-to-end delay is the lowest.

##### 3.2.2. Multiple connections

Fig. 10 shows the contrast on network lifetime among different algorithms in case of multiple connections. The network lifetime means the time period when the first node uses up energy and quits from network. As the routing of GPSR is established as per coordinates of next-hop node rather than broadcast, its energy consumption is the lowest, and its network lifetime is the longest. AODV is on-demand routing, when there is no data for sending, it does not establish routing, but it needs broadcast each time of connection establishment. The more connections there are, the heavier its network load is, and the faster its network lifetime descends. In the figure, when the connection number is more than 8, its network lifetime is lower than that of SDN and OLSR. SDN only needs to broadcast for one time at preliminary stage, when there is change in network, only a part of nodes need to

transmit neighbor information to SDN, the energy initially consumed is lower than that of OLSR, so the network lifetime of SDN is longer than that of OLSR.

##### 3.2.3. Disjoint multipath of SDN

Multiple connections mean that multiple different source nodes send information to multiple target nodes in the meantime. Multipath emphasizes that the source node/target node for different paths is the same. Table 7 shows the results of multipath simulation for SDN. It can be seen that the detour of paths searched is further as the number of paths increases, thus delay, hop count and energy consumption for each package gradually increase, and network lifetime is shorter.

### 4. Conclusion

In this paper, We proposed a routing protocol which applied SDN to multi-hop wireless network. The proposed protocol is implemented using OPNET simulation and compared with other algorithms (including AODV, OLSR, and GPSR). The simulation results show that with global view, SDN centralized control can provide shortest path and disjoint multipath routing for nodes, and that its network lifetime is longer than existing algorithms (OLSR, AODV) when traffic load reaches a certain value. With the expansion of network size, it is necessary to introduce multiple controllers to the wireless network. In the future, deployment of multiple controllers with intelligent agents and node mobility will be the focus of our research.

### Acknowledgments

The authors would like to extend their sincere appreciations to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for its funding of this research through the Profile Research Group project (PRG-1436-17).

### References

- Aloi, Gianluca, Fortino, Giancarlo, Pace, Pasquale, 2015. A software defined network solution for spontaneous wireless access extension. *IoT* 360 (2), 515–520.
- Amani, M., Mahmoodi, T., Tatipamula, M., et al., 2014. Programmable policies for data offloading in LTE network. In: Proceedings of the ICC 2014 – 2014 IEEE International Conference on Communications, pp. 3154–3159.
- Carlos, Miguel, Tavares, Calafate, Giancarlo, Fortino, Sascha, Fritsch, Monteiro, Janio M., Juan-Carlos, Cano, Pietro, Manzoni, 2012. An efficient and robust content delivery solution for IEEE 802.11p vehicular environments. *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 753–762.
- Chen, M., Zhang, Y., Hu, L., Taleb, T., Sheng, Z., 2015a. Cloud-based wireless network virtualized, reconfigurable, smart wireless network. *ACM/Springer. Mob. Netw. Appl.* 20 (6), 704–712.
- Chen, M., Hao, Y., Qiu, M., Song, J., Wu, D., Humar, I., 2016a. Mobility-aware caching and computation offloading in 5G ultradense cellular networks. *Sensors* 16 (7), 974–987.
- Chen, M., Ma, Y., Song, J., Lai, C., Hu, B., 2016b. Smart clothing connecting human with clouds and big data for sustainable health monitoring. *ACM/Springer. Mob. Netw. Appl.* 21 (5), 825–845.
- Chen, M., Qian, Y., Mao, S., Tang, W., Yang, X., 2016c. Software-defined mobile networks security. *ACM/Springer. Mob. Netw. Appl.* 21 (5), 729–743.
- Chen, M., Zhang, Y., Li, Y., Hassan, M., Alamri, A., 2015b. AIWAC affective interaction through wearable computing and cloud technology. *IEEE Wirel. Commun. Mag.* 22 (February (1)), 20–27.
- Chen, M., Wen, Y., Jin, H., Leung, V., 2013. Enabling technologies for future data center networking a primer. *IEEE Netw.* 27 (July (4)), 8–15.
- Chen, M., Zhang, Y., Hu, L., Mao, S., 2015c. EMC emotion-aware mobile cloud computing in 5G. *IEEE Netw.* 29 (March (2)), 32–38.
- Chen, M., 2014. NDNC-BAN supporting rich media healthcare services via named data networking in cloud-assisted wireless body area networks. *Inf. Sci.* 284 (November (10)), 142–156.
- Chen, M., et al., Ma, Y., Li, Y., 2016. Wearable 2.0: Enable Human-Cloud Integration in Next Generation Healthcare System. *IEEE Communications* 54 (12), 3–9.
- Clausen, T., Jacquet, P., 2003. Optimized link state routing protocol (OLSR). *Manet Work. Group*.
- De Gante, A., Aslan, M., Matrawy, A., 2014. Smart wireless sensor network management based on software-defined networking. *IEEE Commun.*, 71–75.
- Fortino, Giancarlo, Trunfio, Paolo, 2014. Internet of things based on smart objects, technology, middleware and applications. Springer, (ISBN 978-3-319-00490-7).

- Fortino, Giancarlo, Guerrieri, Antonio, OHare, Gregory M.P., Ruzzelli, Antonio G., 2012. A flexible building management framework based on wireless sensor and actuator networks. *J. Netw. Comput. Appl.* 35 (6), 1934–1952.
- Fortino, Giancarlo, Giannantonio, Roberta, Gravina, Raffaele, Kuryloski, Philip, Jafari, Roozbeh, 2013. Enabling effective programming and flexible management of efficient body sensor network applications. *IEEE Trans. Hum.-Mach. Syst.* 43 (1), 115–133.
- Galzarano, S., Savaglio, C., Liotta, A., et al., 2013. Gossiping-based aodv for wireless sensor networks. In: *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 26–31.
- Jayashree, P., Princy, F.I., 2015. Leveraging SDN to conserve energy in WSN-An analysis. In: *Proceedings of the International Conference on Signal Processing, Communication and Networking*. IEEE, pp. 1–6.
- Karp, B., Kung, H.T., 2000. GPSR: Greedy perimeter stateless routing for wireless networks. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, pp. 243–254.
- Ku, I., Lu, Y., Gerla, M., et al., 2014. Towards software-defined VANET: Architecture and services. In: *Proceedings of the Ad Hoc NETWORKING Workshop*. IEEE, pp. 103–110.
- Lei, T., Lu, Z., Wen, X., et al., 2014. SWAN: An SDN based campus WLAN framework. In: *Proceedings of the International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems*, pp. 1–5.
- Li, Y., Chen, M., 2015. Software-defined network function virtualization: a survey. *IEEE Access* 3, 2542–2553.
- Lichtblau, B., Dittrich, A., 2014. Probabilistic Breadth-First Search-A Method for Evaluation of Network-Wide Broadcast Protocols. In: *Proceedings of the Ieee/acm/ ifip International Conference on New Technologies, Mobility and Security*, pp. 1–6.
- Liu, Q., Ma, Y., Alhussein, M., Zhang, Y., Peng, L., 2016. Green data center with IoT sensing and cloud-assisted smart temperature controlling system. *Comput. Netw.* 101 (June), 104–112.
- Luo, T., Tan, H.P., Quek, T.Q.S., 2012. Sensor OpenFlow: enabling software-defined wireless sensor networks. *IEEE Commun. Lett.* 16 (11), 1896–1899.
- Mahmud, A., Rahmani, R., 2011. Exploitation of openflow in wireless sensor networks. In: *Proceedings of the , IEEE International Conference on Computer Science and Network Technology* no. 1, pp. 594–600.
- Miyazaki, T., Yamaguchi, S., Kobayashi, K., et al., 2014. A software defined wireless sensor network. In: *Proceedings of the IEEE, International Conference on Computing, Networking and Communications*, pp. 847–852.
- Perkins, B.C., Royer, E., 2010. Ad hoc on demand distance vector (AODV) routing. In: *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, New.
- Stefano, Galzarano, Antonio, Liotta, Giancarlo, Fortino, 2013. QL-MAC: A Q-Learning Based MAC for Wireless Sensor Networks. In: *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, Springer International Publishing, pp. 267–275.
- Stefano, Galzarano, Giancarlo, Fortino, Antonio, Liotta, 2014. A Learning-Based MAC for Energy Efficient Wireless Sensor Networks. *Proceedings of the International Conference on Internet and Distributed Computing Systems*, Springer International Publishing, pp. 396–406.
- Xia, W., Wen, Y., Foh, C.H., et al., 2015. A survey on software-defined networking. *IEEE Commun. Surv. Tutor.* 17 (1), 27–51.
- Zhang, Y., Chen, M., Mao, S., Hu, L., Leung, V., 2014. CAP crowd activity prediction based on big data analysis. *IEEE Netw.* 28 (July (4)), 52–57.
- Zhang, Y., 2016. Grorec a group-centric intelligent recommender system integrating social, mobile and big data technologies. *IEEE Trans. Serv. Comput.* 9 (5), 786–795.