Software Defined Network Routing in Wireless Sensor Network

Junfeng Wang[1], Ping Zhai[1], Yin Zhang[2](✉), Lei Shi[1], Gaoxiang Wu[3], Xiaobo Shi[3], and Ping Zhou[3]

[1] School of Information Engineering, Zhengzhou University, Zhengzhou, China
{iewangjf,iepzhai,ielshi}@zzu.edu.cn
[2] School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China
yinzhang@zuel.edu.cn
[3] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
{gaoxiangwu.epic,xiaoboshi.cs,pingzhou.cs}@qq.com

Abstract. Software-Defined Networking (SDN) is currently hot research area. The current researches on SDN are mainly focused on wired network and data center, while software-defined wireless sensor network (WSN) is put forth in a few researches, but only at stage of putting forth models and concepts. In this paper, we have proposed a new SDN routing scheme in multi-hop wireless network is proposed. The implementation of the protocol is described in detail. We also build model with OPNET and simulate it. The simulation results show that the proposed routing scheme could provide shortest path and disjoint multipath routing for nodes, and its network lifetime is longer than existing algorithms (OLSR, AODV) when traffic load is heavier.

Keywords: Software Defined Network (SDN) · Wireless Sensor Network (WSN) · Routing · Multipath

1 Introduction

In wireless sensor network, each node may act as data source & target node, and forwarding node as well. 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 wireless sensor network. Otherwise, restricted energy and mobility requirements of node also bring difficulties to design and optimization of routing protocol [1].

Traditional multi-hop wireless routing is divided into active routing and passive routing; active routing such as OLSR [2] is based on broadcast information; in each node, the routing information from that node to all other nodes is saved, so there is so much routing information that requires to be saved in each node, and too much internal storage is occupied; therefore, active routing is not adapted to high dynamic network. As for passive routing such as AODV [3], the

J. Wan et al. (Eds.): CloudComp 2016, SPNCE 2016, LNICST 197, pp. 3–11, 2018.
https://doi.org/10.1007/978-3-319-69605-8_1

routing is searched with broadcast each time when sending data is required by node; when multiple nodes require sending routing, nodes need broadcasting for many times to search routing; when there are too many links for a node, too much energy is consumed by broadcast.

SDN separates control from data, and open uniform interface (such as OpenFlow) is adopted for interaction. Control layer is responsible for programming to manage & collocate network, to deploy new protocols, and etc. Through centralized control of SDN, uniform network-wide view may be obtained, and dynamic allocation may be conducted to network resources as per changes in network flow [4]. Currently, the most routing researches for software-defined network are with respect to wired network and data center [5,6]; though software-defined Internet of Things and software-defined wireless sensor network are put forth in a few researches, but only at stage of putting forth models and concepts.

In researches on SDN based on wireless network, the characteristics of wireless network, such as broadcast characteristics, hidden terminal, node mobility and etc. shall be taken into consideration. OpenFlow Protocol is only applicable to route selection, however, applying more functions such as perceiving a variety of sensor data, sleep, aperiodic data collection and etc. in wireless network node, cannot be realized with OpenFlow Protocol and Standard.

Transforming original sensing node is put forth by some researchers, for instance, the concept of Flow-Sensor and utilization of OpenFlow Protocol between Flow-Sensor and controller is put forth in document [7]. Realization of SDN sensor based on MCUs and FPGAs with super low power consumption is put forth in document [8]. In some researches, the framework of SD-WSN and Sensor OpenFlow Protocol [9] that applies in WSN are put forth; lightweight IP Protocol such as uIP and uIPv6 based on Contiki operating system shall be utilized in WSN. From the point of application fields, there are campus WLAN [10], VANET [11], network between mobile base station and base station controller, WSN, MAC laye in WSN, and etc.

The common problem for above researches is that only concepts and simple models are put forth in most researches, and that simulation is not realized or only simple simulation is realized. The description on detailed design and realization algorithms for SDN routing and controller is relatively obscure, and there is no systematic description or realization. In this paper, a novel wireless sensor network routing protocol is proposed, detailed description is conducted to realization process and details of protocol, and model is established with OPNET and simulation verification is conducted to it. The contributions of this document are as follows:

- A WSN routing protocol based on SDN is put forth; the controller has network-wide view and provides single-path routing or multipath routing for other nodes.
- The residual energy of nodes in controller is updated in real time by routing protocol; the shortest path is generated based on energy and hop count.
- The generation method for disjoint multipath from source to target is put forth.

The other parts of this document are arranged as below: routing protocol scheme shall be introduced in Part 2, simulation verification shall be illustrated in Part 3, and Part 4 is summarization to the whole document.

2 Routing Scheme

Exclusive SDN controller node (hereinafter controller for short) is added in network; the broadcast information of controller is reported to each sensing node, normal node sends node information to controller, controller generates the whole network view as per information of normal nodes; when source node requires controller to transmit path, controller calculates the shortest path with Dijkstra algorithm and sends 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.

2.1 Routing Process Design

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

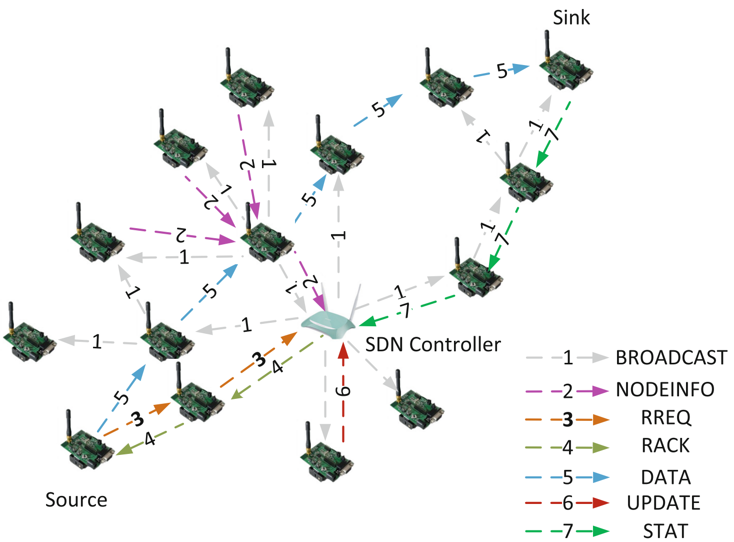

Fig. 1. Schematic diagram of protocol flow

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 picture 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 with 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 in neighbor node information is discovered by some node, that node would report that change to controller;
7. When there is data receipt at target node, statistical information should be reported to controller periodically.

2.2 Controller Broadcast

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

Simply flooding broadcast package in network would cause problems such as rebroadcast & redundancy, signal collision, broadcast storm and etc. Especially when network nodes are relatively dense, these problems would be more outstanding. Generally, wireless sensor 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, thereinto, there are algorithms based on probability, counter, distance, location, neighbor information and etc. As for probability-based method [12], 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 [13], 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 needs neighbor information.

In the algorithm based on neighbor information, the algorithm where MPR nodes are selected by OLSR routing is taken into reference; the neighbors of a part of nodes are selected for broadcast. 1-hop and 2-hop neighbor nodes of some node are utilized in this algorithm.

Tests were conducted for 4 algorithms (3 broadcast methods and full-node broadcast) in simulation scene; the results of performances contrast are shown in Table 1.

Table 1. Performance comparison of four broadcast methods

Method	Number of broadcast	Lifetime(s)	Parameter
All nodes	800	744	
Probability	233	773	p = 0.3
Counter	201	784	Threshold = 3, Wait time = 0.02s
Greedy neighbor	159	798	

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 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 NODEINFO 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.

After a node receives SDN broadcast package, there is certain delay before it sends NODEINFO 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 firstly, and summarization would occur gradually from edge to center. After combination, the relay nodes frequently sending DATA package may be avoided, and energy consumption may be reduced.

After controller receives NODEINFO 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 there is global view at controller, and controller is able to provide routing for other nodes.

2.3 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 RREQ package includes: SN, source node, target node and number of path requested. After receiving RREQ 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 RACK package and forwards this package back to source node.

After receiving RREQ package, controller shall operate 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.

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

Thereinto, stands for residual energy of node j, and 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 may 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, otherwise, residual energy of node may also be collected and estimated by controller as per UPDATE package and statistical package of node.

When source node requests multi-path routing to target node from controller, Dijkstra algorithm shall be invoked for many times as per number of routing requested.

3 Simulation Results

Model is established with OPNET, and simulation is conducted. The contrast among four routing protocols (AODV, OLSR, our SDN routing and GPSR are made, GPSR is introduced as the routing with shortest path for contrast (here the energy consumption when GPSR obtains location information).

3.1 Different Node Density

The contrast among values of energy consumption for each package is as shown in Fig. 2, 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 broadcast, so its energy consumption for each package is ranked the third; thereinto, GPSR with shortest path 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.

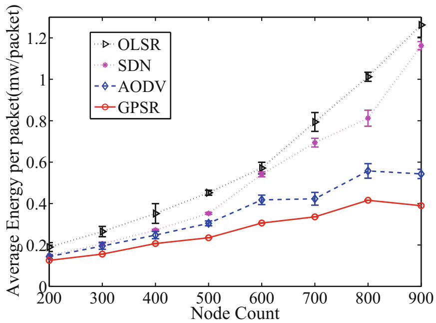

Fig. 2. Contrast on energy consumption and hop count for each package in different network size

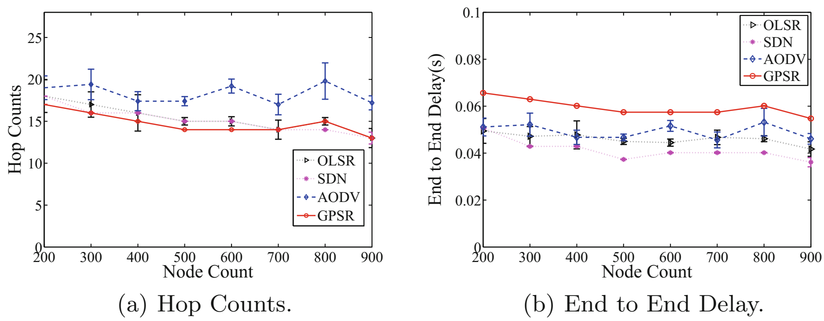

Fig. 3. Contrast on mean hop count and delay in different network size.

Figure 3 shows the contrast on hop count and delay among different algorithms; it can be seen from the hop count figure that the higher the node density is, the number of forwarding nodes that may be selected is more; 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 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, so the end-to-end delay is the lowest.

4 Conclusion

In this document, a kind of routing protocol where SDN is applied in wireless sensor network is put forth, the protocol put forth is realized with OPNET simulation and contrast is made among this protocol and other algorithms. The simulation results show that with global view, SDN centralized control may provide shortest path and disjoint multipath routing for nodes, and that its network lifetime is longer than existing algorithms (OLSR, AODV) when load reaches a certain value. In the future, deployment of multiple controllers and node mobility will be taken into consideration.

References

1. Chen, M., Ma, Y., Song, J., Lai, C.F., Hu, B.: Smart clothing: connecting human with clouds and big data for sustainable health monitoring. Mob. Netw. Appl. **21**, 1–21 (2016)
2. Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A., Viennot, L.: Optimized link state routing protocol (OLSR) (2003)
3. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing, Technical report (2003)
4. Li, Y., Chen, M.: Software-defined network function virtualization: a survey. IEEE Access **3**, 2542–2553 (2015)
5. Chen, M., Hao, Y., Qiu, M., Song, J., Wu, D., Iztok, H.: Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks. Sensors **16**(7), 974 (2016)
6. Chen, M., Qian, Y., Mao, S., Tang, W., Yang, X.: Software-defined mobile networks security. Mob. Netw. Appl. **21**, 1–15 (2016)
7. Mahmud, A., Rahmani, R.: Exploitation of openflow in wireless sensor networks. In: 2011 International Conference on Computer Science and Network Technology (ICCSNT), vol. 1, pp. 594–600. IEEE (2011)
8. Miyazaki, T., Yamaguchi, S., Kobayashi, K., Kitamichi, J., Guo, S., Tsukahara, T., Hayashi, T.: A software defined wireless sensor network. In: 2014 International Conference on Computing, Networking and Communications (ICNC). IEEE, pp. 847–852 (2014)
9. Luo, T., Tan, H.-P., Quek, T.Q.: Sensor openflow: enabling software-defined wireless sensor networks. Commun. Lett. **16**(11), 1896–1899 (2012). IEEE

10. Lei, T., Lu, Z., Wen, X., Zhao, X., Wang, L.: Swan: an SDN based campus WLAN framework. In: 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), pp. 1–5. IEEE (2014)
11. Ku, I., Lu, Y., Gerla, M., Ongaro, F., Gomes, R.L., Cerqueira, E.: Towards software-defined vanet: architecture and services. In: 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), 103–110. IEEE (2014)
12. Cartigny, J., Simplot, D.: Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences, p. 10. IEEE (2003)
13. Levis, P.A., Patel, N., Culler, D., Shenker, S.: Trickle: a self regulating algorithm for code propagation and maintenance in wireless sensor networks. University of California, Computer Science Division (2003)