



Research on Backbone Routing Protocol of Ad Hoc Network Based on SDN

Yiming Yang, Ding Wang, Mao Yang^(✉), Zhongjiang Yan, and Bo Li

School of Electronics and Information, Northwestern Polytechnical University,
Xi'an, China
yangmao@nwpu.edu.cn

Abstract. In recent years, with the development of Internet of Things and UAV network, Ad Hoc networks have increased the number of nodes, enhanced node mobility, and become more sensitive in delay. However, the existing Ad Hoc network routing protocols have some disadvantages, such as high signaling overhead, slow response speed and slow convergence speed. In order to improve the efficiency of routing protocols, we propose a backbone routing protocol based on SDN (software defined network). The core idea of the protocol is as follows. Firstly, there is a logical centralized controller to implement the control plane strategy such as backbone node selection algorithm, and the centralized controller can switch between deep control and shallow control flexibly according to the network state. Secondly, the backbone nodes exchange control signaling, and other nodes transmit data through the backbone nodes, which can reduce the signaling cost and ensure the response speed and convergence speed of routes. Simulation results in multiple topologies show that the centralized backbone routing protocol significantly reduces the overhead of establishing routes and improves the throughput.

Keywords: Backbone routing protocol · SDN · High efficiency · Ad hoc network

1 Introduction

Ad Hoc network is a wireless network that contains no infrastructure. In Ad Hoc network, each node has the routing function. Nodes exchange signaling to form the routing table. Because nodes in Ad Hoc networks can move freely and do not depend on infrastructure, Ad Hoc networks are widely used in scenarios with complex communication requirements such as emergency communications, military exercises, UAV communications, disaster relief, and intelligent communications [1].

Ad Hoc networks have the advantages of easy to deploy, unlimited topology, and fast to organize network. However, due to the lack of infrastructure, end-to-end transmission between nodes needs to be established based on routing

protocols. Nowadays, routing protocols in Ad Hoc networks can be divided into two categories: proactive routing protocol (table driven routing protocol) and on-demand routing protocol. In recent years, with the development of the needs of the Internet of Things and UAV network, Ad Hoc networks have increased the number of nodes, enhanced node mobility, and become more sensitive to delay. Therefore, how to improve the efficiency of routing protocols has become the focus of academic research.

On-demand routing protocols only look for paths when nodes need to send data. The location-based anonymous routing proposed in literature [2] achieves the anonymity and security of nodes in Ad Hoc networks. The Ad Hoc on-demand distance vector routing proposed in literature [3] avoids the problem of DSDV, makes nodes store the required routes, minimizes broadcast demand, reduces memory demand and unnecessary duplication to save resources. Although the existing on-demand routing protocols have been improved in terms of network security and resource saving, they have not improved in terms of route response speed and convergence speed. This is because the on-demand routing method needs to find the path before sending data each time, and then sends the data packet after finding the path, which greatly increases packet delay. When the path fails to find, the resources and time of the protocol package will be wasted, and the network stability will be greatly reduced.

In the proactive routing protocol, the node broadcasts and forwards protocol packets to form a routing table regardless of whether it needs to send packets. The OLSR model based on security clustering proposed in reference [4] extends the network lifetime by motivating the normal behavior of nodes. The resource-aware OLSR routing mechanism in mobile Ad Hoc networks proposed in literature [5] relies on battery power and available bandwidth to select MPRs, which improves the availability of the network and prolongs the service time of the network. Although the existing proactive routing protocols have good availability, they still have the problem of excessive signaling. This is because whether the node sends packets or not, it will broadcast protocol packets periodically and forward these to form a routing table. A large number of protocol packets will cause congestion in the network, occupy channel resources, affect packet sending and receiving, and bring huge overhead to the node.

According to the survey, the existing studies on these two protocols are formed in a certain perspective, which cannot meet the Ad Hoc network scenario. In order to deal with the scenario of Ad Hoc network with increasing number of nodes and enhanced mobility, and solve the problems of large overhead and slow convergence rate of existing routing protocols, we design a backbone routing protocol based on SDN. The core idea of the protocol is as follows. There is a logical controller in the whole network, which is responsible for the selection of backbone nodes. The controller also uses the Dijkstra algorithm to select bridge nodes which will establish full connectivity between backbone nodes through the bridge nodes. The current network status is judged by collecting network information periodically. The shallow control function is used for networks with rapidly topology changes, and the deep control function is used for networks with

slowly topology changes. Only backbone nodes need to forward routing control information, and the controller adjusts the policy based on the network status. In this way, the signaling overhead is effectively reduced, the network service life is prolonged, and the route response speed and convergence speed are improved to ensure the reliability of network.

The structure of this paper is as follows: Sect. 2 describes the backbone routing architecture based on SDN; Sect. 3 introduces the design of the protocol from the control plane and the data plane. Section 4 shows the simulation verification results in different scenarios. Section 5 gives the final conclusion.

2 Backbone Routing Architecture Based on SDN

In the traditional network layer, the control plane is combined with the data plane and deployed on each node in a distributed architecture. This architecture makes it difficult to update the version of the control plane and has few functions. SDN separates the data plane from the control plane. Data planes are deployed in distributed mode, and the control planes of all nodes are centralized. In this way, the network is easy to manage and can be programmed to meet special service requirements. In addition, routes can be allocated according to traffic to realize load balancing.

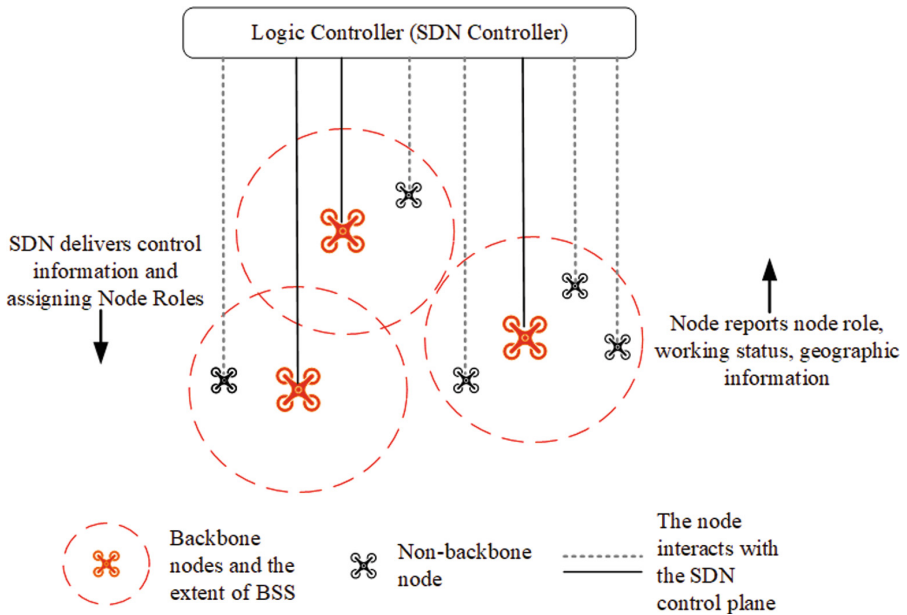


Fig. 1. Backbone routing architecture based on SDN.

Referring to the traditional SDN architecture, we design a backbone routing protocol based on SDN. The centralized control backbone routing architecture

consists of two parts: the controller and the nodes in the network. The SDN controller obtains node information, link status, working status and other information reported by the nodes in the network through the interface. Based on the information, the SDN controller makes decisions and sends them to nodes to realize the configuration of network resources and the setting type of nodes. The above figure shows the specific architecture diagram (see Fig. 1).

The SDN controller in the centralized control backbone routing protocol designed in this paper has the function of adjusting the management network policy in real time. By collecting node information in the network regularly, the controller determines the network status and decides whether the shallow control strategy or the deep control strategy should be adopted.

When the topology changes slowly, the SDN controller will determine that the current network is stable and the position of nodes changes very little in a short time. In this case, deep control is adopted. SDN controller will collect the position, power, working status and other information of the nodes in the network, according to the selection algorithm of backbone node, selecting the backbone nodes and its corresponding non-backbone nodes in a BSS. At the same time, according to the topology information of the nodes in the network, the routing table of the backbone nodes is calculated and updated, and the information is delivered to each node in the network through the interface to complete the node role division and the update of the backbone routing table. Since the SDN controller calculates the routing table from the global perspective, the path result may be shorter than the path obtained by the method of exchanging protocol packets, which accelerates the response speed and convergence speed of the route.

When the topology changes rapidly and nodes move quickly, the SDN controller will determine that the current network is unstable and nodes will join or leave the network at any time. In this case, shallow control is adopted. The SDN controller selects backbone nodes based on the collected information to complete node role assignment. The backbone routing table is established and maintained by the Ad Hoc network through exchanging protocol packets. Nodes send and receive protocol packets automatically according to the specified process. The shallow control has more functions and can flexibly respond to topology changes.

In any scenes, the SDN controller will collect the information of the nodes in the network regularly, make real-time responses to the changes in the network, and make different decisions according to different network states, thus reducing the overhead of exchanging protocol packets and ensuring the reliability of the network.

3 Protocol Design

3.1 Control Plane

In this paper, nodes in Ad Hoc network are divided into different BSS (Basic Service Set). Backbone node refers to BSS header node and bridge node in each BSS. Non-backbone node refers to other nodes in the BSS. The most important

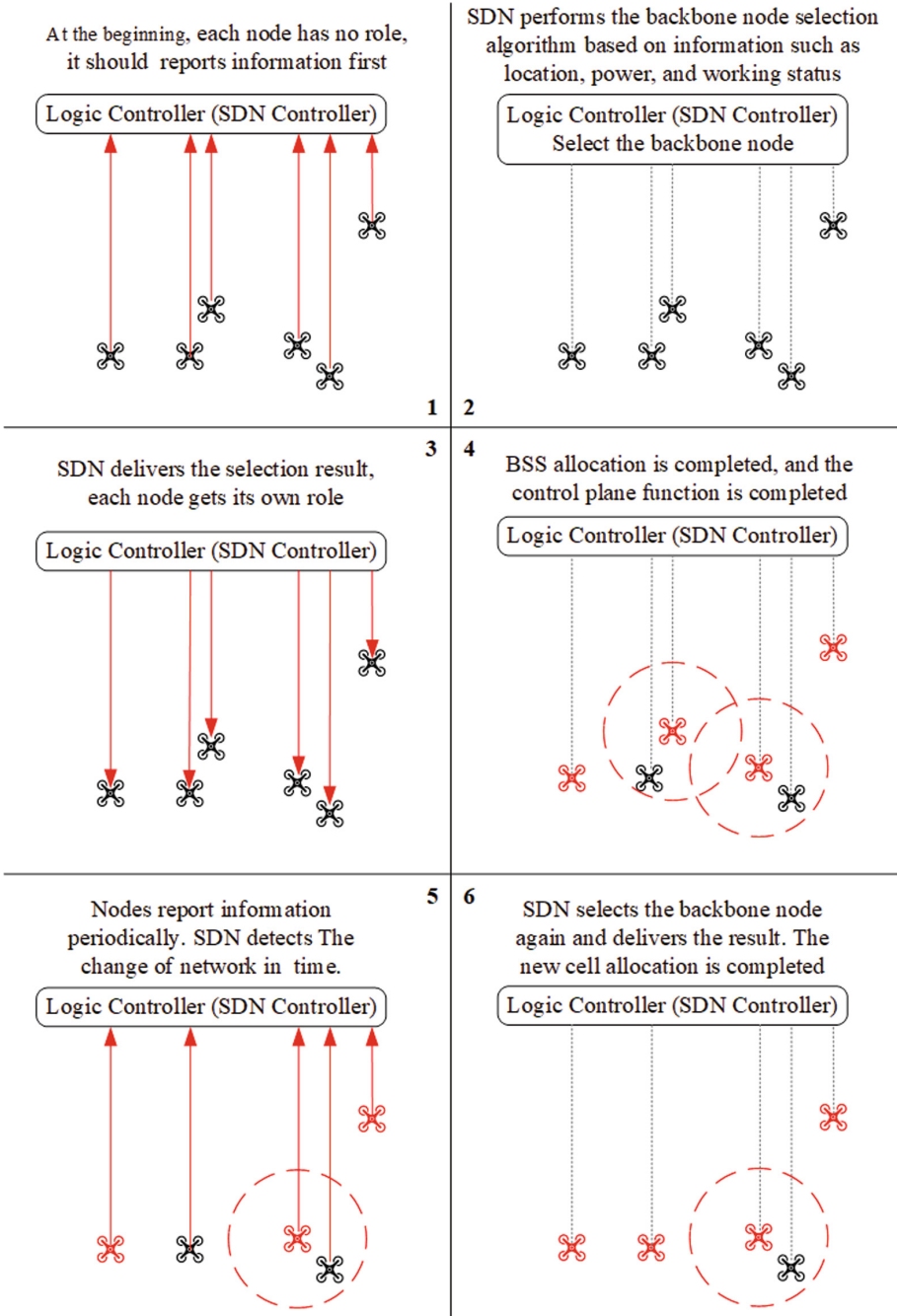


Fig. 2. The control plane functions.

function of the SDN controller is to select the backbone node. The backbone node algorithm used in this paper is as follows. According to the location, power and other information of nodes, SDN controller preferentially sets nodes which have high power, large coverage area and more covered nodes as backbone nodes. According to the coverage of the backbone nodes, the controller uses the Dijkstra algorithm to select the bridge nodes while will enable the backbone nodes to establish connections. In addition, the nearest backbone node is selected as the BSS header for the remaining nodes. In this way, BSS partition is completed.

Another important function of logic controller is to select different control strategies according to different network states.

In deep control, the SDN controller uses the backbone node selection algorithm to select the backbone nodes, and then uses the existing routing protocol to obtain the backbone routing table according to the topology of the backbone nodes in the network. The node roles and routing table are delivered to nodes to realize the reasonable allocation of network resources. In this mode, backbone nodes can be selected to reduce the overhead of exchanging protocol packets, and routing table can be formed in a short time to accelerates route response speed and convergence speed.

In shallow control, the SDN controller uses the backbone node selection algorithm to select the backbone nodes, and then delivers the node roles to the nodes to complete the division of node types and cells in the network. This mode reduces the overhead of exchanging protocol packets by selecting backbone nodes, reduces the instability caused by rapid topology changes, and improves the network throughput.

The following picture shows the function diagram of the control plane, which contains six UAV nodes (see Fig. 2). At the beginning, each node has no role and the color is black, and each node needs to report information to the SDN controller. The SDN controller performs backbone node selection and delivers the results. In part 4, the red UAV is the selected backbone node, and the black UAV is the BSS member node. By reporting information periodically, the SDN controller can discover the changes of network topology and update the role information timely. Part 5 to Part 6 describes the process of the SDN controller from discovering the change of topology to completing the update of node roles.

3.2 Data Plane

The design of data plane is divided into two parts: protocol packet design and data packet design. The following figure shows the interaction flow of protocol packet and data packet between backbone nodes and backbone nodes or between backbone nodes and non-backbone nodes in the same BSS (see Fig. 3).

The protocol packet includes the HELLO packet and TC packet. The HELLO packet is used to maintain the link status with other nodes, one-hop neighbor table, and two-hop neighbor table. The TC packet is used to form the backbone routing table. All nodes on the network need to send HELLO packets periodically. Only backbone nodes need to periodically send TC packets to generate the backbone routing table.

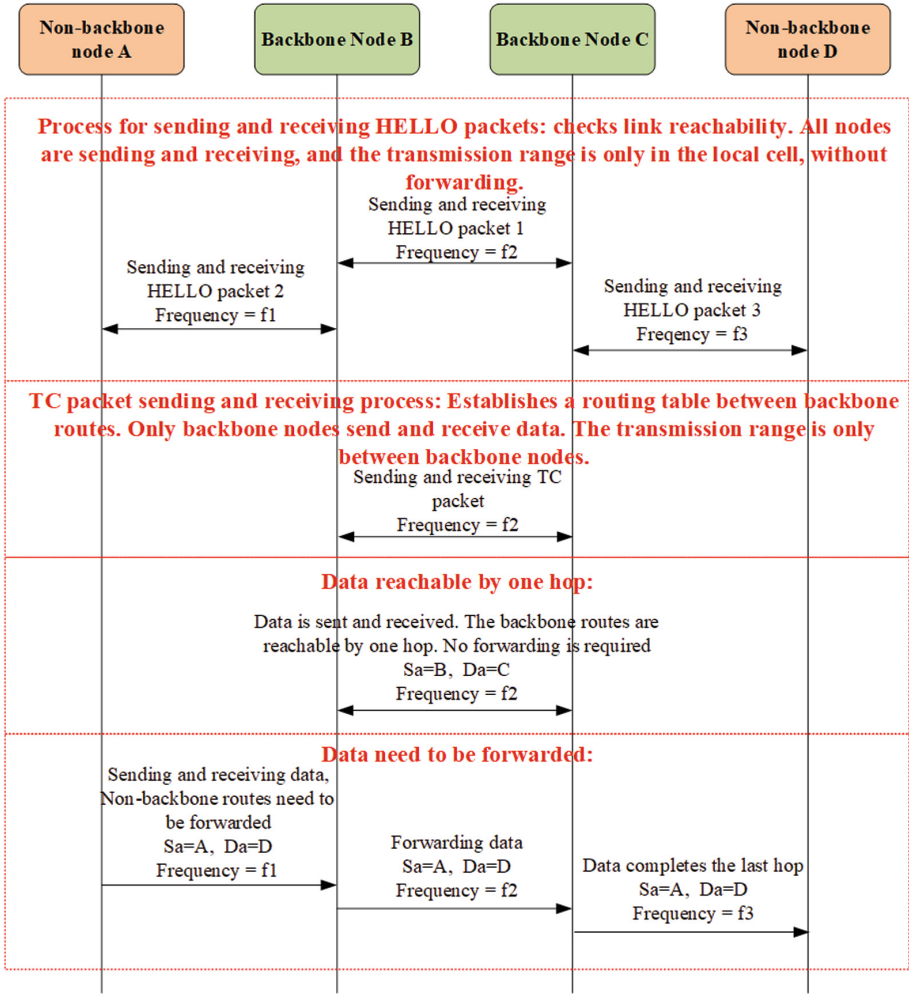


Fig. 3. The data plane functions.

For data packet, according to the different types of source nodes, the data forwarding process can be divided into two categories:

- (1) If the source node is a non-backbone node, the data needs to be forwarded to the BSS header node of the same BSS. The BSS header node queries the next backbone node to be forwarded according to the backbone routing table, and the BSS header node of the destination BSS forwards the data to the destination node.
- (2) If the source node is a backbone node, the next backbone node to be forwarded is queried according to the backbone routing table, and the BSS header node in the destination BSS is forwarded to the destination node.

In this protocol, the candidate range of MPR nodes is narrowed to backbone nodes, and non-backbone nodes will not be selected as MPR nodes. This is because only the backbone node needs to maintain the routing table and is responsible for the routing and forwarding of data, while the non-backbone node does not need to maintain the routing table. The data of a BSS member node only needs to be sent to the BSS header node, and the routing function is completed by the BSS header node. Reducing the number of MPR nodes can reduce TC packet flooding and save channel resources.

4 Simulation Verification

To compare the performance of backbone routing protocol and OLSR, we design two different scenarios: chain scenario and random distribution scenario. All variables are the same except the routing protocol. The throughput and protocol packet proportion of the two protocols are compared.

4.1 Simulation Scenario One: Chain Scenario

In this scenario, backbone nodes are arranged in a chain and non-backbone nodes are randomly distributed around backbone nodes, as shown in the following figure (see Fig. 4). The yellow node is a backbone node (BSS header), and the green node is a non-backbone node (BSS member). The line between the yellow node and the green node indicates that they belong to the same BSS.

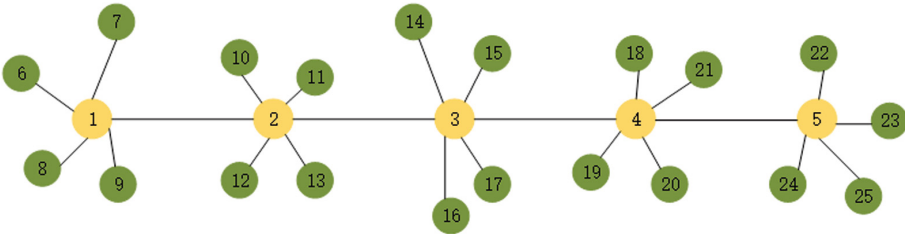


Fig. 4. Chain scenario.

4.2 Simulation Scenario One: Result Analysis

In different protocols, the throughput varies with the traffic rate. As can be seen from the figure (see Fig. 5), when the rate is less than 10 Mbps, the channel is not saturated, so the throughput of both protocols increases with the increase of the traffic rate. When the rate is 10 Mbps, the throughput reaches the maximum. When the rate is greater than 10 Mbps, the channel reaches saturation and the throughput remains stable. At different traffic rate, the throughput of the backbone routing protocol is always higher than that of OLSR (Table 1).

Table 1. Simulation parameters.

Parameter	Value
Traffic Rate	0.01 Mbps to 50 Mbps
Simulator Duration	10 s
Number of nodes	25

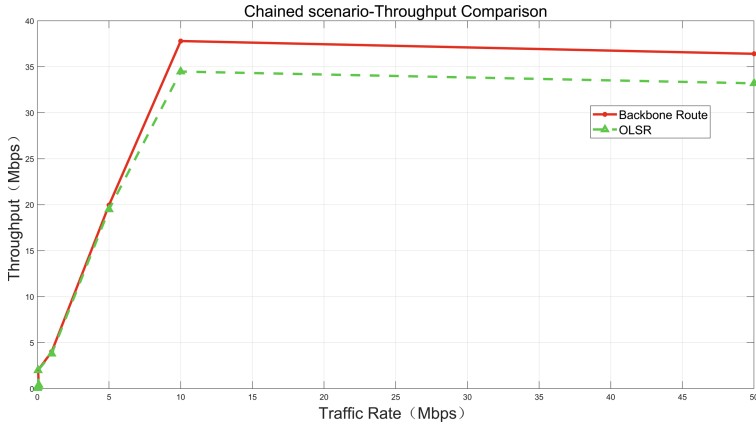


Fig. 5. The throughput of chain scenario.

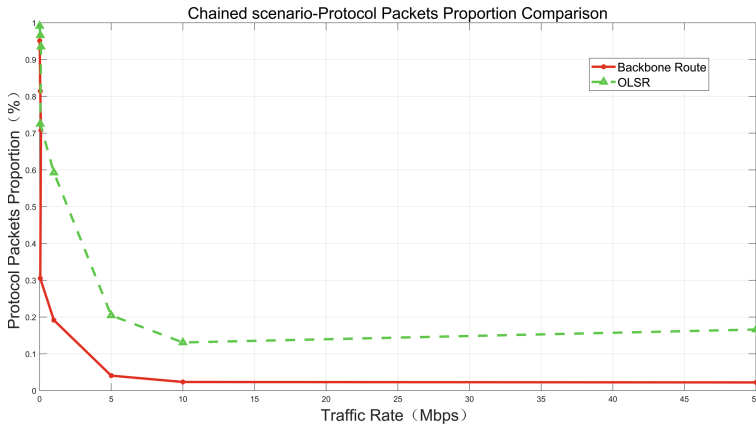


Fig. 6. The protocol packet proportion of chain scenario.

In different protocols, the protocol packet proportion varies with the traffic rate. As can be seen from the figure (see Fig. 6), protocol packet proportion decreases first and then tends to be flat. When the rate is less than 10 Mbps, the number of data packet increases with the rate, so the protocol packet proportion decreases gradually. When the rate is 10 Mbps, the number of packets

tends to saturate. When the rate is greater than 10 Mbps, the proportion of protocol packets remains unchanged. At different traffic rate, the protocol packet proportion of the backbone routing protocol is always lower than that of OLSR.

4.3 Simulation Scenario Two: Random Distribution Scenario

In the random distribution scenario, the coordinates of twenty-five nodes are randomly generated. The backbone nodes are selected by SDN controller, and then the bridge nodes are selected by Dijkstra algorithm. The network topology of scenario two is generated, as shown in the following figure (see Fig. 7 and Table 2).

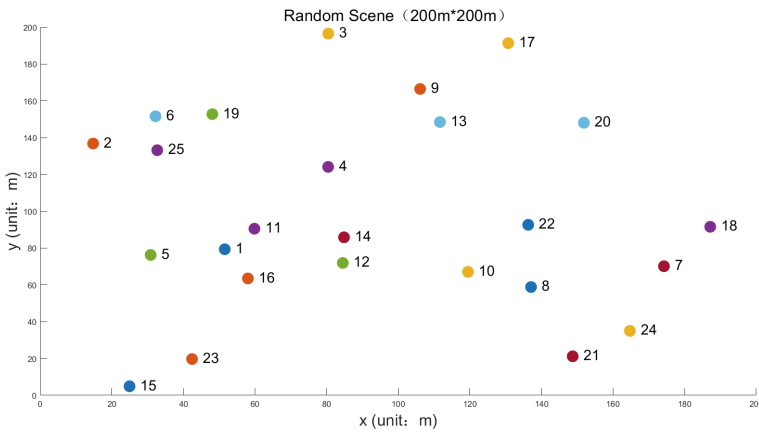


Fig. 7. Random distribution scenario.

Table 2. Simulation parameters.

Parameter	Value
Traffic Rate	0.01 Mbps to 100 Mbps
Simulator Duration	10 s
Number of nodes	25
Backbone Node ID	8, 9, 14, 16, 18, 23, 25
Bridge Node ID	4, 7, 10

4.4 Simulation Scenario Two: Result Analysis

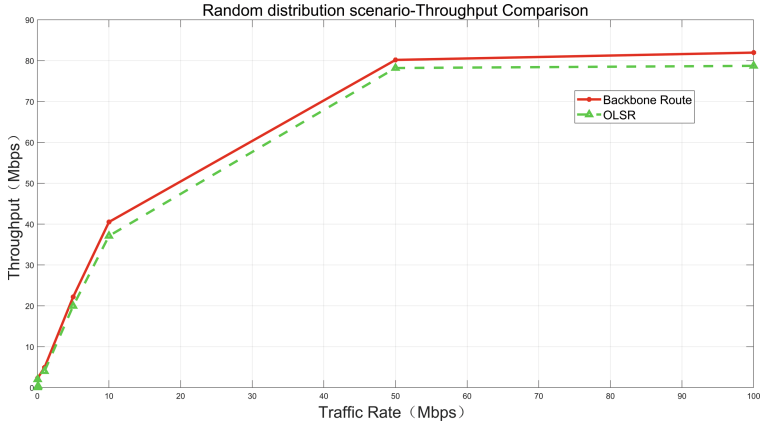


Fig. 8. The throughput of the random distribution scenario.

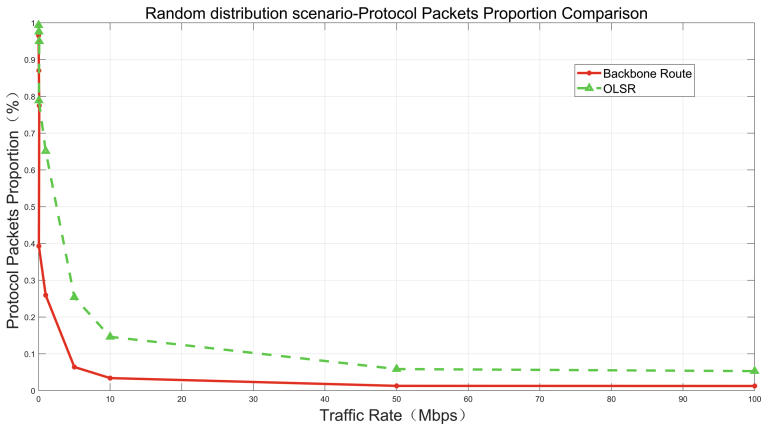


Fig. 9. The protocol packet proportion of the random distribution scenario.

In different protocols, the throughput varies with the traffic rate. As can be seen from the figure (see Fig. 8), when the rate is less than 50 Mbps, the channel is not saturated, so the throughput of both protocols increases with the increase of the traffic rate. When the rate is 50 Mbps, the throughput reaches the maximum. When the rate is greater than 50 Mbps, the channel reaches saturation and the throughput remains stable. At different traffic rate, the throughput of the backbone routing protocol is always higher than that of OLSR.

In different protocols, the protocol packet proportion varies with the traffic rate. As can be seen from the figure (see Fig. 9), protocol packet proportion decreases first and then tends to be flat. When the rate is less than 50 Mbps, the number of data packet increases with the rate, so the protocol packet proportion decreases gradually. When the rate is 50 Mbps, the number of packets tends to saturate. When the rate is greater than 50 Mbps, the proportion of protocol packets remains unchanged. At different traffic rate, the protocol packet proportion of the backbone routing protocol is always lower than that of OLSR.

5 Conclusion

In this paper, a backbone routing protocol based on SDN is designed to solve the problems of excessive signaling overhead, slow response speed and slow convergence speed of existing Ad Hoc network routing protocols. A logical centralized controller performs backbone node selection, switches control policy between deep control and shallow control according to the network status flexibly and timely, reduces signaling overhead, and ensures the response speed and convergence speed of routes. Simulation results in multiple topologies show that the proposed centralized control backbone routing protocol significantly reduces the overhead of route establishment and improves the throughput.

Acknowledgements. This work was supported in part by the National Natural Science Foundations of CHINA (Grant No. 61871322, No. 61771392, and No. 61771390), and Science and Technology on Avionics Integration Laboratory and the Aeronautical Science Foundation of China (Grant No. 20185553035 and No. 201955053002).

References

1. Hashim, A.-A., Farhan, M.M., Alshybani, S.: Performance evaluation of OLSR and AODV routing protocols over mobile ad-hoc networks. In: 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), pp. 1–8 (2019). <https://doi.org/10.1109/ICOICE48418.2019.9035171>
2. Kulkarni, N.S., Gupta, I., Raman, B.: On demand routing protocols for mobile ad hoc networks: a review. In: 2009 IEEE International Advance Computing Systems Conference, pp. 586–591 (2009). <https://doi.org/10.1109/IADCC.2009.4809077>
3. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings WMCSA 1999. Second IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999). <https://doi.org/10.1109/MCSA.1999.749281>
4. Chriqi, A., Otrok, H., Robert, J.: SC-OLSR: secure clustering-based OLSR model for ad hoc networks. In: 2009 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 239–245 (2009). <https://doi.org/10.1109/WiMob.2009.48>
5. Romanik, J., Kraśniewski, A., Golan, E.: RESA-OLSR: resources-aware OLSR-based routing mechanism for mobile ad-hoc networks. In: 2016 International Conference on Military Communications and Information Systems (ICMCIS), pp. 1–6 (2016). <https://doi.org/10.1109/ICMCIS.2016.7496549>

6. Rahman, M., Mambo, M., Inomata, A., Okamoto, E.: An anonymous on-demand position-based routing in mobile Ad Hoc networks. In: International Symposium on Applications and the Internet (SAINT 2006), pp. 7, 306 (2006). <https://doi.org/10.1109/SAINT.2006.13>
7. Kreutz, D., Ramos, F.M., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. *Proc. IEEE* **103**(1), 14–76 (2015). <https://doi.org/10.1109/JPROC.2014.2371999>
8. Gupta, L., Jain, R., Vaszkun, G.: Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutorials*, **18**(2), 1123–1152. Secondquarter (2016). <https://doi.org/10.1109/COMST.2015.2495297>
9. Xia, W., Wen, Y., Foh, C. H., Niyato, D., Xie, H.: A survey on software-defined networking. *IEEE Commun. Surv. Tutorials*, **17**(1), 27–51. Firstquarter (2015). <https://doi.org/10.1109/COMST.2014.2330903>
10. Sharma, P.K., Chen, M.Y., Park, J.H.: A software defined fog node based distributed blockchain cloud architecture for IoT. *IEEE Access* **6**, 115–124 (2018). <https://doi.org/10.1109/ACCESS.2017.2757955>
11. Perkins, C.E., Royer, E.M., Das, S.R., Marina, M.K.: Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Pers. Commun.* **8**(1), 16–28 (2001). <https://doi.org/10.1109/98.904895>
12. Zhang, H., Guo, J.: Application of manet routing protocol in vehicular ad hoc network based on NS3. In: 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 391–394 (2017). <https://doi.org/10.1109/ICEIEC.2017.8076589>
13. Hao, J., Duan, G., Zhang, B., Li, C.: An energy-efficient on-demand multicast routing protocol for wireless ad hoc and sensor networks. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 4650–4655 (2013). <https://doi.org/10.1109/GLOCOMW.2013.6855685>
14. Zeng, Y., Zhang, R., Lim, T.J.: Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Commun. Mag.* **54**(5), 36–42 (2016). <https://doi.org/10.1109/MCOM.2016.7470933>
15. Shakhathreh, H., et al.: Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. *IEEE Access* **7**, 48572–48634 (2019). <https://doi.org/10.1109/ACCESS.2019.2909530>