Research on QoS Centralized Routing of Software-Defined Aeronautic Swarm Network

Hao Zhang¹,a*, HaiTao Li¹,b

¹Beijing University of Technology, Beijing, China
*a-e-mail: s2018zhanghao@emails.bjut.edu.cn
*Corresponding author: b-e-mail: lhaitao@bjut.edu.cn

Abstract. To address the problem of unstable data transmission caused by frequent topology changes in Aeronautic Swarm Network (ASNET), a centralized routing algorithm for Quality of Service (QoS) of the aeronautic swarm network based on Software Defined Network (SDN) is studied. First, this paper designs a Software Defined Aeronautic Swarm Network (SD-ASNET) model. Then, taking advantage of the SDN controller's ability to obtain global link-state information, we propose a Multiple Link-state Awareness routing (MLSA) based on the shortest path algorithm, taking into account link bandwidth, transmission delay and other factors. The simulation results show that the proposed centralized routing protocol makes full use of the centralized decision-making capability of the SDN controller, and effectively optimizes the link bandwidth utilization and transmission delay by using link parameter information.

1. Introduction

The aeronautic domain has long been an important part of the systematic global battlefield. With the continuous advancement of war concepts and technologies, the future aeronautic combat environment will face highly dynamic and highly confrontational issues. In order to cope with challenging aeronautic combat missions and exploit diverse aeronautic combat capabilities, researchers have proposed the concept of aeronautic swarm. The Aeronautic Swarm Network (ASNET) was inspired by the behavior of biological swarms in the context of swarm combat applications. It is a new type of airborne tactical network composed of multiple drones[1]. Airborne cluster networks are network-centric and use airborne networks to achieve large-scale platform networking, thus providing efficient organization and collaboration capabilities and flexible response capabilities. The currently prevalent airborne networks use traditional distributed network architectures to design aviation self-organizing networks, but its limited network management capabilities cannot meet the flexible operational requirements of future aviation cluster networks. SDN-based centralized network control[2] achieves complete separation of control plane entities from data or forwarding plane entities, providing the ability to adapt to rapid network changes, which is a critical feature for aviation cluster networks.

To meet the needs of the future aeronautic swarm airborne network, this paper extends the concept of SDN to the airborne tactical network (ATN), and proposes the Software Defined Aeronautic Swarm Network (SD-ASNET). To enable the aeronautic swarm network to cope with the instability of the highly dynamic network environment, a multi-link state-aware centralized routing protocol (MLSA) is proposed in this paper. Links satisfying QoS constraints are selected and effective routes are calculated based on link costs.
2. Routing Algorithm

2.1. System structure
The aeronautic swarm consists of a certain number of aircraft (usually tens to hundreds), including multiple types of combat platforms such as reconnaissance aircraft, fighter jets, and UAVs. In the SD-ASNET architecture designed in this paper, the control plane is a centralized controller configured on a mainframe command platform such as reconnaissance aircraft, which provides a global view of the underlying network and centralized management of data forwarding. The data plane is deployed on non-command platforms such as UAVs and consists of switches responsible for processing data forwarding. In the aeronautic swarm network, due to the mobility of drone nodes, the interconnection between the controller and the nodes in the data plane is very unstable, which is very different from traditional SDN application scenarios\(^3\). Therefore, we consider introducing QoS into the routing algorithm of the SDN controller to meet the flexible application of the cluster network, which aims to discover multiple valid routes based on the metrics of each link. Real-time link parameters such as transmission delay and available bandwidth are considered to calculate the link weights and select the optimal route based on the current network state. Figure 1 shows the structure of the MLSA routing system proposed in this paper, which is mainly implemented by three modules. They are topology awareness module, state processing module, and route selection module.

![Figure 1 MLSA system module](image)

2.2. Topology awareness module
The main function of the topology management module is network topology discovery and link state information collection, and provides information storage for the subsequent routing calculation module. This module mainly implements the following three points: (1) Use LLDP protocol\(^4\) to discover network topology and obtain node information in the network, including switch node information and wireless link information. And it constructs virtual topology in the controller to keep it updated in real time. (2) Use the Networkx function to store a two-dimensional network topology \(G=(K,E)\) containing nodes and links. (3) Call the K-Shortest Paths (KSP) algorithm in the Networkx library, calculate the first K shortest paths and store them in the matrix function. In the topology discovery phase, the global network topology is discovered through the LLDP protocol, and a virtual topology is constructed in the controller to keep real-time updates.

2.3. State processing module
The state processing module is to provide network information for routing calculations, mainly to complete the statistics and analysis of switch data, and calculates the evaluation index of route selection...
including bandwidth and transmission delay.

(1) Transmission Delay

Transmission delay refers to the transmission time of data from the sender to the destination of the network[5]. In order to ensure the timeliness of data transmission between SDN switch nodes, the transmission delay should be reduced as much as possible. The process of MLSA's discovery of the lowest-latency route is as follows. First, read the path set \( P \) from the topology awareness module. Among them, a shortest path is composed of \( n+1 \) nodes and \( n \) links. The corresponding link delay in each path is \( D_1 = \{d_1, d_2, \ldots, d_m\} \), and the delay of each link is \( d_j \). The controller sends a message carrying the time stamp message Packet_Out to the switch \( OV_i \), and the switch forwards the message to another switch port \( OV_j \) according to the target node in the message. Since the switch \( OV_j \) has no matching flow entry, it will uploads the Packet_In message to the controller. After the controller parses, it obtains the sending and arrival time \( t_{sent} \) and \( t_{arrival} \) on the forwarding path between the controller and the switch. Then, use the EchoRequest message to calculate the round-trip transmission time \( RTT \) of the data packet between the controller and the switch. The delays required for sending an Echo message from the controller to the switch are \( RTT(OV_i) \) and \( RTT(OV_j) \) respectively. Finally, calculate the transmission delay of the target link according to the following formula:

\[
d_j = \frac{(t_{arrival} + t_{sent} - RTT(OV_i) - RTT(OV_j))}{2}
\]

The actual delay of each path is the sum of the delays of all links included in the path, as shown by equation (2).

\[
D_1 = \sum_{j=1}^{n} d_j
\]

(2) Available bandwidth

Since adjacent nodes in the network may transmit data at the same time, interference will occur between the links, resulting in the actual bandwidth consumed by the transmission data packet will far exceed the set bandwidth, and the highest available bandwidth needs to be found between any two points in the network[6]. The SDN controller periodically collects information about each port of the switch. Each switch port has a counter to count the number of data packets sent and received by this port and the number of bytes sent and received. Through the OpenFlow protocol message[7], the SDN controller can obtain the number of bytes sent, the number of bytes received, and the survival time. According to the interval time of OFP messages sent out regularly, and the number of bytes sent by the current port from the beginning and the number of bytes changed by the port counter in a period, the controller calculates the used bandwidth of the link.

\[
BandWidth_{used} = \frac{S_i - S_j}{T}
\]

Among them, \( s_i \) represents the value of the counter, and \( T \) represents the statistical time interval. The difference between the predetermined bandwidth of the link and the used bandwidth of this link is expressed as the available bandwidth of the link.

\[
BandWidth_{available}(i, j) = BandWidth_{total}(i, j) - BandWidth_{used}(i, j)
\]

\[
BandWidth_{total}(i, j)\text{ is the total bandwidth capacity of the link \( (i, j) \). For each complete path }\text{path} = \{\text{Node}_1, \ldots, \text{Node}_i, \text{Node}_j, \ldots, \text{Node}_n\}\text{ from the source node } src \text{ to the destination node } dst, \text{ the available bandwidth } BandWidth_{available}\text{ of the path }\text{ is the minimum available bandwidth of all links } e_{ij}\text{ included in the path, as shown in equation (5).}
\[ B_{\text{available}}^{\text{max}} = \min \{ B_{\text{available}}(\text{Node}_i, \text{Node}_j) \}, \quad i, j \in n \]  
\[ e_{ij} = \{ \text{Node}_i, \text{Node}_j \}, \quad e_{ij} \in \text{path} \]  

(5)

Where \( \text{Node}_n \) is the switch node that may be included in the path. For the link state indicator of bandwidth, the available bandwidth of the link in each path and the available bandwidth of the path are calculated by formulas (4) and (5). And among the transmission paths in the set \( \text{path} \), the path with the largest available bandwidth is selected as the optimal choice.

2.4. Route selection module

This module is the core module of the MLSA routing system. Based on the data provided by the first two modules, the main applications for completing optimal routing and data forwarding are as follows: (1) Obtain the network topology information and the set of the first K shortest paths stored by the topology awareness module. Then obtain the available bandwidth and delay information of all links calculated in the path acquisition module, and integrate them into the delay matrix and bandwidth matrix corresponding to the network topology \( [8] \). (2) According to different service quality requirements, the controller selects the K shortest paths in the path set \( P \) that meet the QoS constraints. (3) Calculate the transmission delay and available bandwidth of the path, calculate the weight of the path according to the comprehensive weight formula, select the optimal route and issue the flow entry to complete the data flow forwarding.

The weight can reflect the network performance of the link and also serve as a basis for routing path performance. The MLSA routing protocol selects the optimal route through comprehensive weight calculation. Before calculating the network information, the difference between related parameters should be eliminated first, and the link performance parameters should be dimensionlessly processed \( [9] \). Define an evaluation index \( C = (c_b, c_d) \), where \( b \) and \( d \) respectively represent the maximum available bandwidth and transmission delay of the path. Link evaluation indicators can be divided into two categories: one is called a positive indicator, which is directly proportional to the expected performance; the other is inversely proportional to the expected performance, and is called a negative indicator. Obviously, higher transmission delay means worse link performance, and higher available bandwidth means better link performance. Therefore, bandwidth is a positive indicator, and delay is a negative indicator. Use formula (6) to deal with the dimensionlessness of positive and negative indicators.

\[ u_n = \begin{cases} \frac{c_n - c_n^{\text{min}}}{c_n^{\text{max}} - c_n^{\text{min}}}, & n \text{ is a positive indicator} \\ \frac{c_n^{\text{max}} - c_n}{c_n^{\text{max}} - c_n^{\text{min}}}, & n \text{ is a negative indicator} \end{cases} \]  

(6)

The indicator is expressed as \( n \in (b, d) \), \( c_n \) is the actual measured value of index \( n \), \( c_n^{\text{min}} \) is the lowest satisfactory value of index \( n \), and \( c_n^{\text{max}} \) is the highest satisfactory value of index \( n \). Define a column vector \( \mathbf{U} = [u_b, u_d]^T \) to represent the value obtained after dimensionless index \( n \). Then, based on the above state information, the path acquisition module calculates the delay and bandwidth information of all available paths. Define a vector \( \mathbf{W} = [w_b, w_d]^T \), where \( \sum w_i = 1, \quad i \in (b, d) \) represents the weight of the evaluation index. Then the actual comprehensive evaluation weight \( V \) of each path can be represented by the product of the index weight vector \( \mathbf{W} \) and the evaluation index vector \( \mathbf{U} \), as shown in equation (7).

\[ V = \mathbf{W} \times \mathbf{U} = w_b \times u_b + w_d \times u_d \]  

(7)
A higher value of $V$ represents better performance of the link. The metric weight vector $W$ can be scaled according to the QoS requirements.

3. Results

3.1. Experimental equipment

The MLSA algorithm proposed in this paper is deployed in the RYU controller, and the simulation experiment is used to verify the overall performance of the SD-ASNET network architecture. The topology is implemented on Mininet-WiFi (version 2.2.1d1). Six UAV nodes from S1 to S6 are configured with OpenFlow switches and connected to the controller. Then configure the interval time of the RYU controller module, including a topology discovery interval of 5 seconds and a path information collection interval of 5 seconds. The network performance was tested under different traffic numbers through the traffic generation tool Iperf, and compared with the performance of the RYU native shortest path routing algorithm. The performance of the routing system is verified by bandwidth utilization and end-to-end delay.

![Network topology for experiment](image)

Figure 2 Network topology for experiment

3.2. Simulation results

| Routing Algorithm | 10 data streams | 20 data streams | 30 data streams | 40 data streams |
|-------------------|----------------|----------------|----------------|----------------|
| MLSA              | 47.45ms        | 51.33ms        | 55.89ms        | 63.24ms        |
| Dijkstra          | 49.23ms        | 57.34ms        | 66.44ms        | 78.48ms        |

Transmission delay can reflect the congestion of the network. The larger the value, the more serious the network congestion. As the number of data streams increases, competition for link resources intensifies, resulting in a gradual increase in transmission delay. The Dijkstra routing algorithm transfers data traffic to each path with equal probability, which can easily lead to congestion in a certain path to a certain extent.

| Routing Algorithm | 10 data streams | 20 data streams | 30 data streams | 40 data streams |
|-------------------|----------------|----------------|----------------|----------------|
| MLSA              | 12.87%         | 29.64%         | 42.77%         | 53.47%         |
| Dijkstra          | 11.45%         | 26.58%         | 34.48%         | 39.86%         |
Bandwidth utilization is mainly affected by redundant routing in the network topology. The more redundancy, the higher the bandwidth utilization. It can be seen from the results that the bandwidth utilization of MLSA is higher than that of Dijkstra's model. In the initial stage of the simulation, the number of data stream requests is low, and there is little difference in bandwidth utilization between the two routing algorithms. With the massive increase in data flow in the network, the MLSA model showed better performance.

4. Conclusions
This paper uses the ability of SD-ASNET to globally manage and obtain the link state, and proposes a multiple link-state awareness routing protocol constrained by the quality of service. Experiments show that the MLSA algorithm proposed in this paper can improve the ability of routing strategies, and at the same time make more reasonable use of link bandwidth resources, which improves link utilization compared to other routing algorithms. However, due to the limited ability of the controller node to process information, when there are too many load flow entries of the drone nodes in the aviation cluster, the controller nodes in the cluster will be overloaded, and the performance of the entire aviation network will be degraded. However, the influencing factors considered by the MLSA algorithm are not comprehensive enough. In the next step, the overhead and processing load capacity of the control information of the switch nodes in the cluster will be studied and the corresponding node deployment scheme will be proposed.

Acknowledgments
I sincerely thank my committee for providing me with the continued support. I cannot express enough thanks to Dr. Hai Tao Li for guiding me in every stage of this research paper. Without his expert advice and encouragement throughout this project, it would have been very difficult for me to prepare the paper so meaningful.

References
[1] Cao, F.B., Lv, N., Chen, K.F. (2019) Routing Algorithm of Congestion Control for Aeronautic Swarm Network. Journal of Communications, 14.9:813-818.
[2] Ahmad, I., Namal, S., Ylianttil, A.M. (2015) Security in software defined networks: A survey. IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp.2317–2346.
[3] Lee, M.C., Sheu, J.P. (2016) An efficient routing algorithm based on segment routing in software-defined networking. Computer Networks, vol. 103, pp. 44–55.
[4] Mtawa, Y. A., Memari, A., Haque, A. (2019) Evaluating QoS in SDN-Based EPC: A Comparative Analysis. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE.
[5] Peng, S., Azzedine, B. (2018) Performance modeling and analysis of a UAV path planning and target detection in a uav-based wireless sensor network. Computer Networks, 146:217–231.
[6] Zheng, X.H., Pan, Q.F. (2015) Multi-path sdn route selection subject to multi-constraints. Third International Conference on Cyberspace Technology (CCT 2015) IET.
[7] Bagaa, M., Dutra, D., Taleb, T., Samdanis, K. (2020) On sdn-driven network optimization and qos aware routing using multiple paths. IEEE Transactions on Wireless Communications, PP(99), 1-1.
[8] Bagci, K.T., Tekalp, A.M. (2019) Sdn-enabled distributed open exchange: dynamic qos-path optimization in multi-operator services. Computer Networks, 162, 106845-.
[9] Jhaveri, R.H., Tan, R., Ramani, S.V. (2019) Real-time QoS-aware Routing Scheme in SDN-based Robotic Cyber-Physical Systems. 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR). IEEE.