Two-stage Topology Control Strategy for Software-defined UAV Networks Against Jamming Attack

Zhiwei Li1,2, Wenxin Qiao1*, Yu Lu1, Donghao Zhao1, Runze Shi3

1Shijiazhuang Campus, Army Engineering University of PLA, Shijiazhuang, 050000, China
2Aircraft Maintenance Center, 93413 Units, Yongji, CO 044500, China
3Troop 32382, China
qiaowenxinphd@163.com, qiaowenxin@bupt.edu.cn

Abstract. Software-defined network (SDN) technology brings new methods to the autonomous control of UAV networks. Aiming at the deterioration of UAV network wireless communication quality caused by jamming attacks, we propose a two-stage topology control algorithm to improve network service quality. The first stage is the redeployment of the control plane consisted by the relay UAVs. We calculate the position of the jammers based on Received Signal Strength Indication (RSSI), and then we first redeploy the position of the Relay UAVs. The second stage is the redeployment of the data plane consisted by the mission UAVs. Mission UAVs that lose control links due to jamming are redeployed according to the historical state of controllers, and other UAVs are redeployed according to the new state of controllers. The performance metric is the weighted average of load balancing, redeployment delay and Signal to Interference plus Noise Ratio (SINR). The experimental results show that compared with the Distance Closest Migration (DCM) strategy and the Distributed Hopping Algorithm (DHA), the comprehensive performance of the proposed algorithm is better.

1. Introduction

The UAV network is in an open wireless network environment and is vulnerable to jamming attacks. Traditional drones are remotely controlled by ground operators or controlled in a pre-programmed manner. When the UAV loses the control link due to jamming attack, the UAV often hovers in the air waiting for the control link to recover or returns to the ground station according to pre-programmed instructions, thus losing the ability to perform tasks. Software-defined network (SDN) technology separates the control plane from the data plane, giving the UAV network programmable control capabilities. Utilizing the flexible control capabilities of SDN, the UAV network based on SDN enables UAVs to make autonomous decisions and redeploy the location of UAVs in the face of jamming attacks, thereby avoiding jamming.

The loss of the control link of the UAV caused by the jamming attack can be analogized to the controller failure in the SDN network. The current research on controller failures is mainly divided into two categories. One type of method is controller backup. This method refers to deploying a backup controller in the network in advance, and automatically activates the backup controller when the control link is lost. Chaudhary et al. [1] propose a Placement Availability Resilient Controller (PARC) scheme to minimum the extra backup controllers. Zhang et al. [2] propose a failure recovery system for traffic engineering in software-defined WANs, and use backup tunnels to accelerate failure recovery. Hu et al.
[3] propose a dynamic slave controller assignment algorithm that can improve the robustness of control plane. Calle et al. [4] study the method of adding a few additional controllers to increase its resilience to target attacks. Sahoo et al. [5] use Particle Swarm Optimization (PSO) method and FireFly Algorithm (FFA) to design a seamless backup mechanism against single link failure. The controller backup method is not suitable for jamming attack scenarios, because when the UAV is under close jamming attack, if it is not far away from the jammer, it will be useless even to switch the backup controller.

Another method of controller failure recovery is data plane migration. We know that the load balancing constraints must be considered when migrating the nodes whose control links are lost to other control domains that work normally, otherwise it is easy to cause high load crashes. Wang et al. [6] comprehensively considers the migration cost and load balancing rate, and proposes a switch migration algorithm based on the greedy algorithm. Hock et al. [7] weakened the constraints of load balancing, and designed switch migration algorithms based on migration distance and delay. Zhou et al. [8] model the switch migration problem as a signature matching problem so as to protect the most important controllers. Xu et al. [9] study the dynamic mapping algorithm and propose BalCon and BalConPlus, two SDN switch migration schemes to achieve load balance with low cost. Duy et al. [10] propose a mechanism of load balancing and failure recovery named Aloba, whose advantage is to reduce the overhead of communication among controllers. Current research on switch migration is logical migration, but migration in software-defined UAV networks also involves the migration of physical locations. In addition, current research is limited to the migration of the data plane. In the face of jamming attacks, UAVs on the control plane will also be attacked.

To the best of our knowledge, this article is the first attempt to migrate the data plane and control plane at the same time under the constraints of both migration delay and load balancing rate.

2. System Model

2.1. Network Model

The scenario considered in this article is shown in Figure 1. UAV platforms in the network are divided into two categories: relay UAV and mission UAV. The relay UAV has strong communication performance and maneuverability. Therefore, the SDN controller is deployed on these platforms to form the control plane of SDN. The mission UAV has weak communication performance and maneuverability, and is divided into multiple control domains under the management of the relay UAV. When UAVs are under jamming attack, UAVs that have lost control links need to migrate to the neighbour control domain to restore network control.

![Fig. 1 Schematic diagram of the control relationship in software-defined UAV networks](image)

The topology of the software-defined UAV network is denoted as $G = \{V, E\}$ where $V$ is the node set and $E$ is the edge set. $V$ UAV nodes consist of $M$ relay UAV act as control node denoted as...
\[ C = \{c_1, \ldots, c_m, \ldots, c_M\} \text{ and } N \text{ mission UAV act as transmission node denoted as } S = \{s_1, \ldots, s_n, \ldots, s_N\} \]

where \( V = M + N \). The set of transmission nodes that lost control links due to jamming attacks is \( F = \{c_1, \ldots, c_j, \ldots, c_f\} \), and the set of other transmission nodes is \( Z = \{c_1, \ldots, c_z, \ldots, c_z\} \) where \( N = F + Z \).

The set of transmission nodes that controlled by \( c_m \) is \( W_c, c_m \in S \). The Boolean variable \( l_{m,n} \) indicates whether \( c_m \) controls \( s_n \). Assuming that the jammers are far apart, it can be considered that a control domain is affected by at most one jammer \( J = \{c_j\} \). The flight height of the jammer and the mission UAVs are both \( h_s \), and the flight height of the relay UAVs are \( h_m \). The distance between node \( i \) and node \( j \) is \( d_{ij} \) where \( i, j \in \{C, S, J\} \). The transmit power of the relay UAVs, mission UAVs and jammer are \( P_c, P_s \) and \( P_j \). The SINR of node \( i \) is \( \Gamma_i, i \in \{C, S, J\} \). The minimum SINR required by the mission UAV to maintain its control link is \( \sigma \). We denote the set of relay UAVs with qualified communication links as \( C_{\text{in}} \), and denote the set of transmission UAVs that lost control link as \( S_{\text{os}} \). The maximum velocity of UAV is denoted as \( v_i, i \in \{C, S, J\} \). We assume that the motion mode of UAV is only uniform motion and hovering, and its velocity can be changed every \( T \) time. We denote that the velocity of \( n_s \) at \( k^{th} \) time slot is \( v_{n_s}^{(k)} \).

2.2. Path Loss Propagation Model

The path loss propagation model of air-to-air fading can be modelled as a log-normal shadowing model with constant channel power gains. The path loss denoted as \( PL \) can be calculated by

\[
PL(dB) = 10 \rho \log \left( \frac{d}{d_0}\right), \quad d > d_0
\]

where \( d_0 \) is the reference distance, \( \rho = 2 \) is the path loss exponent of air-to-air propagation, and \( \nu \) is the SINR at \( d_0 \). The SINR of mission UAV \( \Gamma_s \) can be calculated by

\[
\Gamma_s = \frac{P_c - \nu - 10\rho \log(d_{c,s}/d_0)}{P_s - \nu - 10\rho \log(d_{s,c}/d_0) + \sigma^2}
\]

where \( \sigma^2 \) is the noise power, \( d_{c,s} \) is the distance from control node \( c_m \) to transmission node \( s_n \) and \( d_{s,c} \) is the distance from transmission node \( s_n \) to jammer \( c_j \).

3. Topology Control Strategy

The main idea of the two-stage topology control strategy is as follow. Each control node periodically checks the connectivity of its transmission nodes. We distinguish between the two situations where the jammer approaches the control node or the task node. When the jammer approaches the control node, it will happen that all transmission nodes in the control domain lose the control link. When the jammer approaches the transmission nodes, the control node of these jammed transmission nodes will sense the existence of the jammer. Because the second case will not cause a large-scale loss of the control link, and the position of the jammer is immediately exposed to the nodes of the entire network. Therefore, this situation is not harmful, we focus on the first situation.

When the control link of a certain control domain is completely jammed, the neighbouring control domain of this jammed control domain will form the set \( C_{\text{in}} \). The topology control algorithm we proposed includes 2 stages. The first stage is the redeployment of the controller. The neighbour controllers of the jamming control domain flying around the original location to ensure that they can receive the jammed transmission node. Other controllers fly away from the jammer. The second stage is the redeployment of transmission nodes. Since the nodes in the \( S_{\text{os}} \) domain have lost control, the
migration goal of these nodes is to restore the control link as soon as possible, that is, the migration delay and SINR are used as performance metrics. However, the migration of a large number of nodes in the $S_{os}$ domain may cause load balancing problems, so the migration goal of transmission nodes in the $C_{im}$ domain is load balancing.

3.1. Performance Metrics

3.1.1. Load balance of the controllers.

Table 1 The two-stage migration algorithm

| Algorithm: two-stage jammed transmission nodes migration algorithm |
|---------------------------------------------------------------|
| **Input:** the set of transmission nodes that need to be migrated $S_{os}$, calculated by SINR $<\sigma$ |
| **Output:** the set of control domain that receive migrating nodes $C_{im}$, control relationship matrix $l_{n,m}$ |
| 1 The nodes in $S_{os}$ estimate healthy control domain $\Lambda$ from set $C_{im}$ satisfying SINR $\Gamma_{mc} > \sigma$ using RSSI value |
| 2 Calculate the load balancing rate $\beta$ and load difference matrix $H$ via (3), (4) |
| 3 for $c_s \in S_{os}$ do |
| // implement by $c_s \in F$ |
| 4 for $c_m \in \Lambda$ do |
| 5 Calculate $d_{c_s c_m}$, $T_{c_s c_m}$, $\beta$, and estimate SINR $\hat{\Gamma}_{c_m}$ using historical RSSI value |
| 6 Choose control domain $c_m^* = \arg\min_{c_m \in \Lambda} \left( T_{mc} - \sigma \right)$ |
| 7 $c_s$ implement the migration to $c_m^*$ |
| 8 Restore the control link of $c_s$ |
| 9 Update load difference matrix $H$ |
| // implement by $c_s \in Z$ |
| 10 Estimate the number of nodes $\tilde{Q}$ that will migrate to this domain |
| 11 Form a virtual load difference matrix $\tilde{H}$ assuming $\tilde{Q}$ has joined this domain |
| 12 if $\tilde{h}_{c_s c_m} > \delta$ |
| 13 Choose any one node $c_s \in Z$ and migrate it to other control domain according to step 4 |
| 14 end if |
| 15 if $h_{c_s c_m} > \delta$ |
| 16 Choose any one node $c_s \in Z$ and migrate it to other control domain according to step 4 |
| 17 end if |
| 18 end for |
| 19 end for |
| 20 if $\sum_{m=1}^{M} l_{n,m} = 1 \ \forall n \in S_{os}, m \in C$ |
| 21 update $C_{im}$ and $l_{n,m}$ |
| 22 end if |

The load of control node $c_m$ is mainly related to the number of transmission nodes belonging to the control node $Q_{cm}$ and the average packet transmission rate $B$. The definition of load of $c_m$ is
$L_m = w_1Q_m + w_2B_m$ where $w_1 + w_2 = 1$. The average load of each control node is $\bar{L} = \frac{1}{Z} \sum_{z=1}^{Z} L_z$, and the load balancing rate can be defined as

$$\beta = \left( \frac{1}{Z} \sum_{z=1}^{Z} (L_z - \bar{L}) \right)^{\frac{1}{2}}.$$  \hspace{1cm} (3)

It can be seen that the smaller the value of $B$, the more balanced the network load. To simplify the calculation, we define the load difference as $h_{ij} = |L_i - L_j|$, then the load difference matrix $H$ is

$$H_{z,x} = \begin{bmatrix}
0 & h_{12} & \cdots & h_{1y} \\
h_{12} & 0 & \cdots & h_{1y} \\
\vdots & \vdots & \ddots & \vdots \\
h_{12} & h_{1y} & \cdots & 0
\end{bmatrix}.$$  \hspace{1cm} (4)

We determine the threshold of load difference for load balancing $\delta$ as follow.

$$\delta = \frac{\max H_{z,x} - \min H_{z,x}}{\max H_{z,x}}$$  \hspace{1cm} (5)

3.1.2. Migrating Delay
Assuming that transmission node $s_n$ needs to migrate to control node $c_m$, the migrating delay is the flying time $T_{mn} = \tau d_{c_m}/v_n$ where $\tau = 0.001$ is an adjustment constant.

3.2. Implementation of Topology Control

3.2.1. Jamming situation awareness.
The position of the jammer can be estimated by using the Received Signal Strength Indication (RSSI) value of the continuous sampling points obtained by the UAVs. With the help of [11], we can obtain the estimated position of jammer $p_j = (x_j, y_j, h_j)$ using RSSI value.

3.2.2. Control algorithm.
The algorithm first estimates the feasible migration control nodes whose SINR meets the requirements based on the RSSI value. Then the algorithm starts a parallel execution process. On the one hand, for nodes in the $S_m$ domain, redeployment is implemented according to SINR and migrate delay. On the other hand, the control nodes in the $C_m$ domain predict the number of nodes that will migrate into this domain, and according to the predicted load conditions, pre-migrate some nodes to other control domains. The detailed process of the algorithm is shown in Table 1. Lines 10-15 in italics in Table 1 are capable of concurrent execution.

4. Simulation

4.1. Simulation Parameters
We use python to write a simulation program to verify the effectiveness of the proposed algorithm. We compare the proposed algorithm with Distance Closest Migration (DCM) algorithm and Distributed Hopping Algorithm (DHA). The DCM algorithm performs migration in a manner that minimizes
migration delay. The DHA algorithm preferentially migrates nodes to nodes with large remaining capacity.

The size of our simulated airspace is 4Km*3km. The minimum SINR required by the receiver is 3dB. The velocity of jammer and mission UAV is 300m/s, and the velocity of relay UAV is 600m/s. The average packet request rate is 3~5 packet/ms. The transmit power of mission UAV and jammer is 0.4W and the transmit power of relay UAV is 1W. The number of relay UAV is 10, and the number of jammers is 1~3. Each relay UAV controls 5, 10, 15 mission UAVs respectively.

4.2. Results Analysis
We conduct simulation experiments for the three cases where the number of transmission nodes in the control domain is 5, 10, and 15 respectively. It can be seen from Figure 2 that compared to the DCM and DHA algorithms, the migration algorithm proposed in this paper can obtain the best comprehensive performance under three different network scales. There are three main reasons for the performance improvement. The first is because our algorithm not only migrates the data plane, but also optimizes the control plane. The second is because the performance metrics we designed are more comprehensive and reasonable. The third is because we designed the algorithm to run in parallel to speed up the convergence speed.

(a) 5 mission UAVs each domain (b) 10 mission UAVs each domain (c) 15 mission UAVs each domain

Fig.2 Performance comparison of migration algorithms under different network scales

5. Conclusion
This paper takes advantage of the flexible control brought by software-defined network technology to design a two-stage anti-jamming attack topology control strategy for UAV networks. Compared with traditional strategies, this method can optimize the overall control plane and data plane. The optimization goal we designed is more reasonable, taking into account the performance of load balancing rate, migration delay and SINR, and the algorithm has better parallelism.

References
[1] Chaudhary Rajat and Kumar Neeraj PARC: Placement Availability Resilient Controller Scheme for Software-Defined Datacenters[J]. IEEE Transactions on Vehicular Technology.2020, 69(8) 8985-9001.
[2] Zheng Jiaqi , Xu Hong , et al. Sentinel: Failure Recovery in Centralized Traffic Engineering[J]. IEEE-ACM Trans. Netw.2019, 27(5) 1859-72.
[3] Hu Tao, Yi Peng , et al. Dynamic slave controller assignment for enhancing control plane robustness in software-defined networks[J]. Futur. Gener. Comp. Syst.2019, 95(1) 681-93.
[4] Calle Eusebi, Cosgaya Sergio G. , et al. 2019 Solving The Backup Controller Placement Problem In SDN Under Simultaneous Targeted Attacks (New York: Ieee)
[5] Sahoo Kshira Sagar , Puthal Deepak , et al. On the placement of controllers in software-Defined-WAN using meta-heuristic approach[J]. J. Syst. Softw.2018, 145(1) 180-94.
[6] Wang Chuanan, Hu Bo , et al. A Switch Migration-Based Decision-Making Scheme for Balancing Load in SDN[J]. IEEE Access.2017, 5(1) 4537-44.
[7] Hock David , Gebert Steffen , et al. 2014 Ieee Network Operations and Management Symposium, (New York: Ieee)
[8] Zhou Yang, Zheng Kangfeng, et al. Elastic Switch Migration for Control Plane Load Balancing in SDN[J]. IEEE Access. 2018, 6(1) 3909-19.

[9] Xu Yang, Cello Marco, et al. Dynamic Switch Migration in Distributed Software-Defined Networks to Achieve Controller Load Balance[J]. IEEE J. Sel. Areas Commun. 2019, 37(3) 515-29.

[10] Duy Phan The, Qui Huynh Phu, et al. 2019 Aloba: A Mechanism of Adaptive Load Balancing and Failure Recovery in Distributed SDN Controllers. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT): IEEE pp 1322-6

[11] Vazquez-Rodas A., Astudillo-Salinas F., et al. Experimental evaluation of RSSI-based positioning system with low-cost LoRa devices[J]. Ad Hoc Netw. 2020, 105(1) 13.