Research on a High Reliability Multi-Controller Deployment Method

Hanghang Gao*, Xiang Wang, Shanghong Zhao, Kun Zhang and Guodong Wang

1Information and Navigation College, Air Force Engineering University, Xi’an 710077, China

*Corresponding author’s e-mail: gaohh321@126.com

Abstract. In the software-defined aeronautical information network architecture, a multi-controller deployment scheme was proposed for the control plane scalability problem, which includes two phases: cluster domain partitioning and intra-domain deployment controller. An improved k-means algorithm was proposed in the cluster domain segmentation phase, which can effectively avoid discrete points from becoming cluster center. When the controller was deployed in the cluster domain, the minimum failure rate of the control path in the network was targeted, and an improved discrete particle swarm optimization algorithm was used to solve the problem to obtain effective control of the network. The simulation results show that the algorithm can effectively reduce the failure rate of the control path based on the reasonable division of the cluster domain. For example, when the number of controllers was increased from 3 to 15, the total failure rate was correspondingly reduced from 0.242 to 0.092, and good results are obtained. This solution provides a new way to solve the problem of multi-controller deployment under dynamic and large-scale networks.

1. Introduction
The aeronautical information network includes an aviation platform for information forwarding and a wireless link between nodes. It has the characteristics of high dynamics, wide coverage, diversified combat tasks, and large amount of access data[1]. However, with the expansion of the network and the complexity of the operational environment, the current aviation information network has gradually exposed a series of problems. For example, how to effectively share the battlefield information obtained by the aviation platform and flexible scheduling control. Different combat missions have different QoS in terms of delay and reliability, and require the network to have strong differentiated service capabilities. Traditional network equipment with tight control of software control and hardware forwarding is also increasingly difficult to meet the needs of rapid network development. The existing closed network equipment integrates a large number of protocols, and various technical standards are complex and diverse. It is difficult for each network equipment to interoperate in different operating environments, which greatly hinders information sharing in the network.

The emergence of a software-defined network provides a new idea for solving the above problems[2]. The SDN decouples the control plane and the data plane in the traditional network device, and uses the SDN controller of the logical centralized control to perform unified control on the underlying forwarding device. The control plane is responsible for policy formulation and resource allocation, and the underlying forwarding device forwards data services. However, the centralized control plane has shortcomings such as single point failure and limited processing capability.
Therefore, it is necessary to pay attention to the scalability of the control plane. At present, the research on the deployment of SDN multi-controllers is mainly based on terrestrial networks. The research content mainly includes network delay, reliability, traffic overhead and other performance indicators[3]. Heller[4] first studied this problem. He defines the average transmission delay and the maximum transmission delay, and uses the actual network topology to test. Yao[5] improved the existing transmission delay model and proposed a transmission algorithm and a delivery algorithm depending on whether or not transmission delay is included. The simulation shows that the transmission algorithm is superior to the delivery algorithm in terms of network response speed and stability. Zhao[6] proposed an improved transmission delay model and simulation algorithm and a delivery algorithm depending on whether or not transmission delay is included. The simulation results show that the convergence time of the algorithm is shorter and can obtain better values. The disadvantage is that the load balancing of the controller is not considered. For the average failure rate of the control path, Hu[7] proposes an indicator of the effective percentage of the effective control path. It achieves the robustness of the network control path by maximizing the expected percentage, but the average failure rate can only reflect the entire network state and don't reflect the worst state of the network. Compared to traditional terrestrial networks, aeronautical information networks have features such as highly dynamic network topologies, longer transmission ranges and unstable link quality. Therefore, multi-controller architecture and deployment algorithms based on terrestrial networks will no longer be applicable to aeronautical information networks. In view of the above problems, this paper firstly improves the traditional planar and hierarchical SDN multi-controller architecture based on the characteristics of the aviation information network. Secondly, an improved cluster domain partitioning algorithm is proposed for the problem of multi-controller load imbalance in the network. Finally, with the node and link outage probability as variables, and the minimum control path failure rate as the goal, an improved discrete particle swarm optimization algorithm is used to solve the problem.

2. Model analysis

2.1. Software-defined aeronautical information network architecture features

As shown in Figure 1, the aeronautical information network is located between the space-based network and the terrestrial network. On the one hand, it can establish an information link with the space-based platform to realize timely injection of space-based information into the aviation information network and provide information support for the aviation platform. On the other hand, an information link can be established with the terrestrial information system to ensure the input of ground allegation information. The aeronautical information network is developing in the direction of network isomerization, service diversification, and functional complexity. In addition, the mobility of the network nodes and the instability of the link put forward higher requirements for the current network performance, and the construction of the SDN-based aviation information network will be more in line with the requirements of future operations on all aspects of the network performance.

Figure 1. Structure diagram of aviation information network
The software-defined aeronautical information network utilizes the centralized control strategy of SDN logic to grasp the global view of the network in time, and meet the multi-user demand in the network. The controller deployment is a prerequisite for building a control plane and it is important for improving network performance. However, single controllers in the control plane usually have single-point failures and limited processing capabilities. Therefore, a multi-controller deployment architecture is adopted. The multi-controller deployment architectures typically include a flat architecture, a vertical architecture, and a hybrid architecture. Because the mobility of the aeronautical platform and the unreliability of the link are likely to cause network failure risks, the planar multi-controller architecture in the terrestrial network will no longer be applicable. Taking into account the above factors, the aeronautical information network should adopt a hybrid controller architecture to centrally control the network. With reference to the hybrid controller architecture model proposed in the paper [8], this paper constructs a hybrid multi-controller architecture under aviation information network based on specific application scenarios, as shown in figure 2:

The control plane in the hybrid multi-controller architecture consists of a Global Controller (GC) and a Local Controller (LC). The GC can implement centralized management and control of the aeronautical information network from the perspective of the global battlefield. Therefore, the GC should be deployed on aircraft nodes with strong information processing capability and survivability, with the highest control priority. The LC is responsible for controlling the network nodes in its own control area. For the actual needs and characteristics of the aviation information network, the LC can be arranged on each aviation platform. The network can enable or disable the LC according to its own state and the deployment strategy of the GC to implement dynamic deployment of the controller.

2.2. Mathematical model establishment
The following describes the deployment of multiple controllers in a software-defined aeronautical information network:

(1) \( G(V, E, V_c, E_c) \) represents an aeronautical information network topology, where \( V \) represents a set of aircraft nodes in the network, \( E \) represents a set of communication links, \( V_c \) and \( E_c \) represent sets of control nodes and control paths respectively, and \( V_c \subseteq V, E_c \subseteq E \).

(2) This paper assumes that the number and location of GCs in a hybrid multi-controller architecture are known. Only the LCs in the control plane are deployed. The control nodes and LC nodes mentioned below have the same concept.

(3) Controllers should be placed on all nodes in the network, and the controllers need to be turned on or off according to the specific deployment strategy. When the controller on node \( i \) is open, \( i \) is the control node, otherwise it is the switching node.

(4) The control path includes the path between the LC and the GC and the path between the switching node and the LC. Because the GC has the highest priority, the control path is configured separately between the GCs to share the network view information. The remaining control paths are no
longer configured separately, and the control information and data information are transmitted along the same path.

(5) Each LC in the network can only be controlled by a unique GC at the same time, and each switching node can only be controlled by a unique LC.

To facilitate the description of multi-controller deployment issues, the following is a description of the symbols in a specific mathematical model:

Table 1. Symbol Description

| Symbol | Physical meaning |
|--------|------------------|
| $p_v$  | Node failure probability |
| $p_e$  | Link outage probability |
| $g$    | GC node |
| $k$    | Number of controllers |
| $h_{ij}'$ | $l$ between the $i,j$ nodes |
| $x_{ij}$ | Switch node $i$ controlled by control node $j$ |

Based on the above analysis, the multi-controller deployment problem in the software-defined aeronautical information network can be established as follows:

$$\min f = f_1 + f_2$$

$$f_1 = 1 - \prod_{i \in j} x_{ij} \prod_{i \in j} h_{ij}'(1 - p_i)$$

$$f_2 = 1 - \prod_{i \in j} x_{ij} \prod_{i \in j} h_{ij}''(1 - p_s)$$

$$\sum_{j \in i} x_{ij} = 1; \forall i \mid y_i = 0, i \in V$$

$$\sum_{i \in j} x_{ij} = \frac{k}{2}(k - 1)$$

Where equation (1) represents the optimization goal is to minimize the control path failure rate. equation (2) is the control path failure rate between the LC node and the switching node. equation (3) is the control path failure rate between the LC node and the GC node. Equation (4) indicates that each switching node is controlled by only one control node. Equation (5) represents the range of values of variables in each formula. Equation (6) represents the number of paths between adjacent control nodes.

3. Algorithm design

3.1. Cluster domain partitioning algorithm based on improved k-means

Since the nodes in the network are dynamic, this paper refers to [8], assuming that the network topology remains relatively stable during the execution of a certain combat mission, and the network topology changes accordingly as the combat mission changes. On this basis, the aviation information network is divided.

The traditional k-means clustering algorithm is a greedy algorithm, which is easy to fall into local optimum, and the initial clustering center selected by the algorithm is very likely to deviate from the data-intensive area. If the initial cluster center is located at an orphan point or a remote point, it will result in poor performance of the divided cluster domain. Aiming at this problem, this paper proposes an improved k-means algorithm. By selecting the appropriate initial cluster points, the data is finally divided into multiple cluster domains.

In view of the characteristics of node initial cluster randomness, in order to avoid the discrete point becoming the initial cluster center, the concept of data potential is introduced. For node $x_i$, the calculation formula is as follows:
\[ D_k(x_i) = \sum_{j=1}^{N} \exp\left(-\frac{1}{\xi} \left\| x_i - x_j \right\|^2 \right), \quad k=0 \]  \hspace{1cm} (7)

\[ D_k(x_i) = D_{k-1}(x_i) - D_{k-1} \cdot \exp\left(-\frac{1}{\xi} \left\| x_i - x_{k-1} \right\|^2 \right), \quad k \geq 1 \] \hspace{1cm} (8)

Where equation (7) represents the initial potential of data \( x_i \), and \( \xi \) is a constant between [0,1]. Equation (8) represents the potential of \( x_i \) in the \( k \)-th iteration, \( D_{k-1} \) is the maximum potential obtained in the \( k \)-1th iteration, \( x_{k-1} \) is the node corresponding to the maximum potential in the \( k \)-1th iteration, and \( \xi \) is a constant which is between \([1.25\xi; 1.5\xi]\).

For the set of nodes \( X = \{x_1, x_2, \ldots, x_n\} \), the potential \( D_k(x_i) \) of the node \( x_i \) is calculated in turn, and the initial cluster centers are finally obtained through iteration. The following describes the constraints in the iterative process:

1. The greater the potential of \( x_i \), the more nodes are distributed around the point, and the smaller the data potential for discrete points.
2. When the potential of the node \( x_i \) satisfies \( D_k(x_i) \leq 0 \), it no longer participates in the iterative process.
3. After \( k \)-1 iterations, \( k \) node maximum potentials are obtained, and the data \( k \) is the cluster initial convergence, as shown in equation (9):

\[ X = \{x_m \mid m = 0, 1, \ldots, k - 1\} \] \hspace{1cm} (9)

From the above analysis, it can be seen that there are many nodes around each object in the maximum potential set, and the objects are far apart. Therefore, selecting the data in \( X \) as the initial clustering center can effectively avoid the initial cluster center distance obtained by the traditional k-means algorithm being too close or the discrete point becoming the initial clustering center. The algorithm is mainly based on the distribution characteristics of nodes to build a cluster domain. In actual deployment, due to problems such as limited capacity of the LC, the LC load is overloaded or underloaded, resulting in data congestion or insufficient resource usage. Therefore, there is a need to increase the constraint that the LC processing capability is limited. This paper introduces the load balancing index \( B \) to constrain the results of the divided cluster domains. The expression is as follows:

\[ B = \max_p \sum_{x \in X_p} a(x) - \min_q \sum_{x \in X_q} a(x) \] \hspace{1cm} (10)

\[ \sum_{x \in X_p} a(x) \leq A(p) \] \hspace{1cm} (11)

Where \( a(x) \) represents the data request information submitted by the node \( x_i \) to the LC in the cluster domain, and the total data request information submitted by the nodes included in the cluster domain cannot exceed the rated load capacity of the LC in the cluster domain.

Algorithm 1 is an improved k-means algorithm based on node potential. The algorithm calculates the initial potential value of each data in set \( X \) and excludes discrete data points. The maximum potential data set and \( k \) initial cluster centers can be obtained by iteration. Finally the remaining data is divided into corresponding clusters by Dijkstra algorithm, as follows:

### Table 2. Improved k-means cluster domain partitioning algorithm

| Algorithm 1: Improved k-means algorithm |
|----------------------------------------|
| **Input**: node \( X \), clusters \( n, \xi, \zeta \) | **Output**: Cluster domain \( X_j \) \((j=1, 2, \ldots, n)\) |
| 1. for \( x_i \in X \) | 10. conclude \( D_k \) |
| 2. compute \( D_0(x_i) \) | 11. \( \vec{X} \leftarrow \vec{x}_k \) |
| 3. if \( D_0(x_i) \leq 0 \) | 12. update \( \vec{X} \) |
| 4. delete \( x_i \in X \) | 13. for \( x_i \in \{X \setminus \vec{X}\} \) |
| 5. update \( X \) | 14. Dijkstra |
| 6. for \( k=1:n \) | 15. conclude \( X_j \) |
| 7. \( \vec{X} \leftarrow \vec{x}_k \) | 16. for \( X_j \) |
| 8. Dijkstra | 17. compute \( LC_{load} \) \( B \) |
3.2. Controller deployment algorithm in the cluster domain

After the aeronautical information network is divided into multiple cluster domains, the LC domain is deployed in the cluster domain with the minimum total failure rate of the control path. The objective function is as shown in equations (1) ~ (7). In this paper, an improved discrete particle swarm optimization algorithm is used to solve the objective function.

Discrete particle swarm optimization is an optimization algorithm based on swarm intelligence theory, in which the particles in space represent the solution of the problem, and the particle function is judged according to the fitness function. The particle is updated according to the individual optimal and global optimal position. It has the characteristics of fast convergence and easy implementation. The particle $X_i$ consists of $d$-dimensional binary code, which limits each dimension of the particle to 0 or 1, and the speed is not limited. Each bit calculates the velocity according to equation (12), and converts the velocity into a probability that the bit variable takes 1 according to equation (13). The change in particle position is calculated with reference to equation (14).

$$v_d = w \cdot v_d + c_1 \cdot r_1 \cdot (pBest_d - x_d) + c_2 \cdot r_2 \cdot (gBest_d - x_d)$$  \hspace{1cm} (12)

$$S(v_d) = \frac{1}{1 + \exp(-v_d)}$$  \hspace{1cm} (13)

$$x_d = \begin{cases} 1, & r \leq S(v_d) \\ 0, & \text{other} \end{cases}$$  \hspace{1cm} (14)

Equation (12) represents the velocity change equation of the particle, where $w$ represents the inertia weight of the particle, $c_1$ and $c_2$ represent the acceleration factor, $r_1$ and $r_2$ both represent the random number in the interval $[0, 1]$. $pBest_d$ and $gBest_d$ are the individual historical optimal values and global historical optimal values in the particle iterative process, and $d$ represents the dimensionality.

However, the sigmoid function in the algorithm can only solve the probability that the particle bit takes 1 or takes 0, and the magnitude of the change of the particle bit cannot be obtained. In addition, the algorithm will be affected by the previous iteration when calculating $x_d$, so the randomness of the algorithm will increase with the number of iterations, which will lead to the algorithm not achieving the desired result. Therefore, an improved discrete particle swarm optimization algorithm is proposed to solve this problem. Since $pBest_i$ and $gBest_i$ already contain historically optimal information of particles, they have high credibility, so each change in particle position is related to $pBest_i$ and $gBest_i$. For the LC deployment problem in the cluster domain, it is assumed that there are $n$ nodes in the network, and each particle represents an LC deployment scheme. For particle $X_i = [x_{i1}, x_{i2}, \cdots, x_{in}], x_{im}$ indicates whether node $m$ is a control node in the $i$-th deployment scenario. The particle updates its position according to $pBest_i$ and $gBest_i$, and defines that the probability that node $i$ becomes a control node when updating at the $T$-th position is $P(x_{it} = 1)$, otherwise it is $P(x_{it} = 0)$. Assume that $pBest_i$ and $gBest_i$ are independent in the iteration, where $pBest_i$ has a trust of $p_p$ and $gBest_i$ has a trust of $p_g$. The Bayes formula can be used to determine its specific probability value:

$$P(x_{it} = 1 | pBest_i = 1, gBest_i = 1) = \frac{p_p \times p_g}{p_p \times p_g + (1 - p_p) \times (1 - p_g)}$$  \hspace{1cm} (15)

$$P(x_{it} = 1 | pBest_i = 0, gBest_i = 0) = \frac{(1 - p_p) \times (1 - p_g)}{p_p \times p_g + (1 - p_p) \times (1 - p_g)}$$  \hspace{1cm} (16)

$$P(x_{it} = 1 | pBest_i = 1, gBest_i = 0) = \frac{p_p \times (1 - p_g)}{p_p \times p_g + (1 - p_p) \times (1 - p_g)}$$  \hspace{1cm} (17)

$$P(x_{it} = 1 | pBest_i = 0, gBest_i = 1) = \frac{(1 - p_p) \times p_g}{p_p \times p_g + (1 - p_p) \times (1 - p_g)}$$  \hspace{1cm} (18)
Since both $p_{\text{Best}}$ and $g_{\text{Best}}$ are historically optimal values in the iterative process of particles, their trust is higher than the average value, and in order to avoid the algorithm falling into local optimum, $p_p > p_g$ should be guaranteed. Referring to [9], set $p_p=0.8$, $p_g=0.7$, and the algorithm can obtain the optimal performance. Where $\alpha$ represents the effect of both on the value of $x$, when $p_{\text{Best}}$ and $g_{\text{Best}}$ are the same, and $\beta$ represents the effect of both on the value of $x$ when $p_{\text{Best}}$ and $g_{\text{Best}}$ are different. The improved BPSO algorithm does not need to calculate the velocity of the particles, and can directly calculate the probability of particle bit values according to equations (15) $\sim$ (18). In addition, the improved BPSO algorithm directly uses the probability of $p_{\text{Best}}$ and $g_{\text{Best}}$ to determine the value of the bit, effectively avoiding the influence of the previous generation value on the late value.

In the process of finding the optimal solution, the degree of goodness of the particles is evaluated by the fitness function. Equation (1) is used as the fitness function of the improved algorithm. If the fitness function value of a particle is small, it means that the solution corresponding to the particle is better.

| Table 3. Controller deployment algorithm in a cluster domain |
|-----------------------------------------------------------|
| Algorithm 2: Controller deployment strategy for improved algorithm |
| Input: Cluster domain $U_i$ |
| Output: Controller deployment scenario |
| (1) Set the particle population size $N$, the number of iterations $T_{\text{max}}$, the number of controllers $n$ |
| (2) Initialize particle position $X_i$, velocity $V_i$, individual optimal $p_{id}$ and calculate global optimal $p_{gd}$ |
| (3) Calculate the fitness value of the initial particle $F$ |
| (4) Compare the fitness value $F$ of the current time with the fitness value of the individual optimal and the fitness value under the global optimal, and judge whether to update the individual optimal and global optimal |
| (5) Update particle velocity $V_i$ and particle position $X_i$ |
| (6) Whether the number of iterations is $T_{\text{max}}$, if not, execute step (3) |
| (7) Whether the number of controllers satisfies $i < n$, if yes, execute step (2) |
| (8) Conclude different controller numbers $k$ and LC deployment strategy |

4. Simulation analysis

4.1. Simulation Settings
In this paper, the scope of the aeronautical information network is set to 500 (km) $\times$ 500 (km), and 46 nodes and 136 links are randomly generated within this range. It is assumed that there is only one GC in the network and ignore the impact of the number and location of GCs on network performance and deployment results. During the deployment of the LC, it is assumed that all LCs have the same processing power and load capacity, and the probability of node and link failure is a random number in $[0, 0.03]$ and $[0, 0.05]$. In the simulation, the Random algorithm and ANIGA algorithm are compared with the improved algorithm for performance comparison. The Random algorithm randomly selects nodes to deploy LCs in the network. ANIGA is a heuristic search algorithm that finds the number and location of LCs when the condition of the outage probability is met by loop iteration. The average of 20 experiments was taken as the final result to avoid interference factors.

4.2. Analysis of results
Figure 3 shows the clustering quality values of the traditional k-means algorithm and the improved algorithm based on node potential. It can be seen that the fluctuation range of the improved algorithm results is smaller than that of the k-means algorithm, and the overall value is lower than the k-means algorithm. This is because the load balancing module is added to the improved algorithm, and the difference in the number of nodes in each cluster domain is small, and the calculated fluctuation range
of the result is small. In addition, since the improved algorithm fully considers the potential value of each node in the network when selecting the initial clustering center, it effectively avoids the discrete point becoming the initial clustering center, and fully demonstrates that the clustering domain with improved algorithm is more reasonable.

Figure 3. Graph of cluster evaluation function

Figure 4. Graph of control path failure rate change

Figure 4 shows the trend of network control path failure rate changes. As the number of controllers increases, the failure rate of the control path under several algorithms gradually decreases. For example, when the number of controllers in the algorithm is 3, 9 and 15, the corresponding control path failure rates are 0.242, 0.122 and 0.092 respectively. This is because as the number of controllers increases, the control path between the control node and the switching node increases, so the reliability of the control path is enhanced. In addition, as the number of controllers increases, the control path failure rate changes slowly, indicating that the number of controllers cannot be increased indefinitely in the network.

Figure 5 shows a time delay between control nodes in a network cluster domain. When the number of cluster domains is the same, as the number of control nodes in each cluster domain increases, the delay between control nodes increases too. As the number of control nodes increases, the control path increases, and the delay of the control nodes in the network that needs to be synchronized increases. When the proportion of the control node remains unchanged, the synchronization delay decreases as the number of cluster domains increases. As the number of cluster domains increases, the number of control nodes in the cluster domain decreases, so the synchronization delay decreases. Figure 6 shows
the total delay between controllers. As the number of controllers increases, the total delay increases. Because the number of paths between controllers increases, the time required to synchronize information increases. For example, when the number of controllers is 5 and 13, the corresponding total delay is 3.01 ms and 4.38 ms respectively.

5. Conclusion
This paper studies the scalability of the control plane in the software-defined aeronautical information network architecture. In this paper, a node-based k-means algorithm is firstly used to divide the aeronautical information network into multiple cluster domains. Secondly, when deploying LC in the cluster domain, aiming at the minimum control path failure rate, an improved BPSO algorithm is proposed to solve it. Finally, the influence of different controller numbers and strategies on network performance is obtained, which provides a feasible solution for multi-controller deployment problems. This article focuses on the deployment strategy of the controller in the absence of failure, and future work will study the survivability of the controller to ensure the continuity of centralized control.

References
[1] Zhao S H, Chen K F, Lv N, Wang X, Zhao J. Soft-defined airborne tactical network of aviati-on clusters[J]. Journal on Communications,2017,38(08):140-155.
[2] Zuo Q Y, Chen M, Zhao G S, Xing C Y, Zhang G M, Jiang P C. Research on SDN Technology Based on OpenFlow[J]. Journal of Software, 2013(5):1078-1097.
[3] Gao X M, Wang B S, Deng W P,Tao J. A Survey of Controller Placement Problems in SDN Networks[J]. Journal on Communications,2017,38(07):155-164.
[4] Heller B, Sherwood R, Mckeown N. The controller placement problem[J]. Acm Sigcomm Compu-ter Communication Review, 2012, 42(4):473-478.
[5] Yao L Y, Chen Y, Song F, Zhang H K. Deployment of SDN Fast Response Controller Based on Time Delay [J]. Journal of Electronics & Information Technology,2014,36(12):2802-2808.
[6] Zhao J, Zhao S H, Zhao W H, Li Y J. Controller Distribution Optimization Strategy for Aeronauti-cal Optical Information Network Defined by High Reliability Software[J]. Systems Engineering a-nd Electronics, 2018, 40(4): 885-890.
[7] Hu Y N, Wang W D, Gong X Y, Que X R, Cheng S D. On the placement of controllers in software defined networks[J]. Journal of China Universities of Posts & Telecommunications, 2012, 19(19):92-97
[8] Lv N, Liu C, Chen K F, Cao F B. A centralized control network deployment method for aviation clusters[J]. Acta Aeronautica Sinica,2018,39(07):172-184.
[9] Xu Y C, Xiao R B. An Improved Binary Particle Swarm Optimization Algorithm[J]. Pattern Reco-gnition and Artificial Intelligence, 2007, 20(06): 788-793.