

Research Article

Network Monitoring Information Collection in the SDN-Enabled Airborne Tactical Network

Kefan Chen , Shanghong Zhao , and Na Lv 

School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China

Correspondence should be addressed to Kefan Chen; 1148180199@qq.com

Received 2 May 2018; Accepted 25 July 2018; Published 9 September 2018

Academic Editor: Enrico Cestino

Copyright © 2018 Kefan Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aviation swarm is considered to be a promising organization of air combat forces to execute combat and noncombat air missions. As a critical component of an aviation swarm, airborne tactical network (ATN) provides the communication capability. Considering the deficiencies of today's ATNs, it is unable to meet the communication demands of an aviation swarm, which motivates us to employ the software-defined networking (SDN) paradigm and design a SDN-enabled airborne tactical network (SD-ATN). For the SDN paradigm, network monitoring information (M-info) is the source of knowledge that forms the globe network view of the control plane; therefore, how to ensure that the control plane collects M-info from the data plane in a reliable and real-time way is a fundamental problem of designing the SD-ATN. To address this issue, a transmission framework called the MCF-SD-ATN is first designed, which makes it practical to provide dedicated quality of service (QoS) guarantees for the M-info collection. Then, a communication protocol called the MCP-SD-ATN is designed to implement the M-info collection work based on the MCF-SD-ATN. We implement the MCF-SD-ATN and the MCP-SD-ATN in EXata 5.1 for simulation. Simulation results show that the proposed solution is appropriate for M-info collection in the SD-ATN.

1. Introduction

With the development of technology and war concept, battlefield environment is becoming much more challenging. Under this background, employing a swarm of the manned or unmanned aircraft with various warfare capabilities to cooperatively execute combat or noncombat air missions is paid much attention recently. We term the new organization of air warfare forces as the *aviation swarm*. An aviation swarm is composed of a certain number of aircraft (typically dozens of aircraft), including aviation platforms equipped with the airborne early warning and control system (AP-AEW&C), fighter jets, and reconnaissance aircraft. Obviously, the superiority of the aviation swarm relies on efficient collaborations among swarm members, requiring deep and extensive interactions among mission systems, weapon systems, and sensors carried on different aircraft. In such a context, a more efficient, more appropriate, and more adaptable communication capability is needed by the aviation swarm.

Airborne tactical network (ATN) has been a critical communication capability for many decades, enabling information sharing between both manned and unmanned military aircraft as well as surface and ground platforms. However, today's ATNs are embraced a tightly coupled and vertically integrated architecture [1]. To meet the communication demands of the aviation swarm, network engineers have to patch the network in a slow and costly way, which makes the network cumbersome, rigid, and difficult to manage. Therefore, there is an urgent need for a new ATN architecture designed for the aviation swarm, and the software-defined networking (SDN) paradigm provides ideas.

SDN is a promising paradigm that distinctly decouples the control plane and data plane of the network. The data plane is only in charge of forwarding data flows, while in the control plane, a logically centralized network controller dictates overall network behaviors [2, 3]. The SDN paradigm simplifies networking operations, optimizes network management, and introduces innovation and flexibility compared with legacy networking architectures. The SDN paradigm has

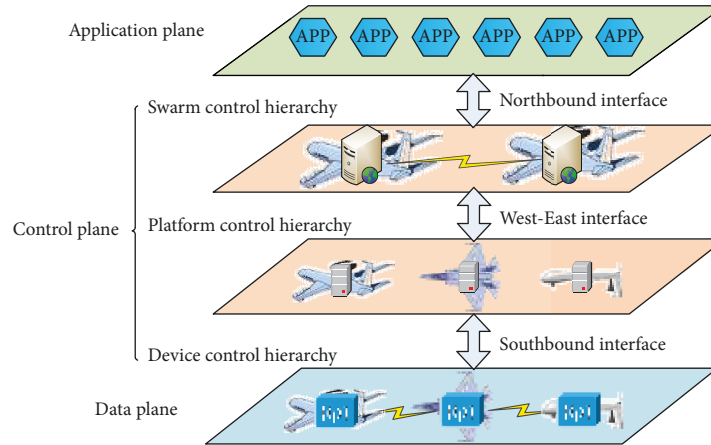


FIGURE 1: The SD-ATN architecture.

been successfully applied to the data center network (DCN) [4], meanwhile applying the SDN paradigm to other network domains, such as wireless cellular network (WCN), vehicular ad hoc network (VANET), and wireless sensor network (WSN), and has been grabbing attentions of both industry and academia [5–8]. Given the advantages of the SDN, we intend to use it as a means of compensating for the deficiencies of the current ATN architecture. In this article, we term the SDN-enabled ATN as *SD-ATN*.

For any network employing the SDN paradigm, the control plane must build a global network view to provide necessary knowledge for dictating network behaviors. Therefore, collecting the network monitoring information (M-info) from the data plane, which reflects the actual network status, is a basic step to manage the network. Current research works focusing on the SDN's network monitoring are mainly to find an optimal point between the adequate monitoring accuracy and monitoring overhead. Tootoonchian et al. [9] present a traffic matrix estimation system called OpenTM that reads byte and packet counters maintained by OpenFlow switches to directly and accurately measure the traffic matrix with a low overhead. Yu et al. [10] propose FlowSense, a push-based network monitoring method where Packet_In and Flow_Removed messages are used to estimate the utilization of links between OpenFlow switches, which reduces the network monitoring overhead. OpenNetMon [11] provides per-flow metrics monitoring of throughput, packet loss, and delay by polling the edge switches at an adaptive rate, reducing network and switch CPU overhead while optimizing measurement accuracy. PayLess [12] provides an API for flow statistics collection at different aggregation levels and uses an adaptive statistics collection algorithm delivering highly accurate information in real-time without generating too much network overhead. SCREAM [13] and DREAM [14] are both systems for dynamically adjusting the resources allocated to each monitoring task, while ensuring a user-specified level of accuracy. Liu et al. [15] present a practical architecture named UnivMon that can leverage previous theoretical advances to serve as the basis for a general and accurate monitoring framework.

Since the control plane and data plane of the SDN paradigm are deployed independently, the communication efficiency between the two planes seriously impacts the benefits provided by the SDN. In fact, all the above research works are carried out for terrestrial wired networks, where the control plane and data plane can efficiently communicate with each other via a stable and high-bandwidth wired network, leading the researchers pay less attention on the communication efficiency between the two planes. Unlike the terrestrial networks, the SD-ATN is a network with high-speed nodes, intermittent wireless links, variable network topology, limited bandwidth, and heterogeneous communication systems, making it challenging for the SD-ATN's control plane to communicate with the data plane efficiently; thus, the quality of service (QoS) requirements of the M-info are hard to be guaranteed.

Given the critical role of the M-info, having the control plane collects M-info from the data plane in an unreliable and high latency way severely undermines the efficiency of the SD-ATN. In such a context, how to ensure that the control plane collects the M-info from the data plane in a reliable and real-time way becomes a fundamental problem of designing the SD-ATN. To address this issue, a transmission framework called the MCF-SD-ATN (M-info collection framework for SDN-enabled airborne tactical network) is first designed, giving a fundamental infrastructure for M-info collection in the SD-ATN. Based on the transmission framework proposed, we design a communication protocol called the MCP-SD-ATN to make the control plane of the SD-ATN collects the M-info from the data plane efficiently.

The rest of this paper is structured as follows. Section 2 presents the preliminaries related to our work. Section 3 describes the transmission framework designed. Section 4 presents the details of the MCP-SD-ATN. Extensive experimental results are presented in Section 5, and Section 6 concludes this article.

2. Preliminaries

The architecture of the SD-ATN is shown in Figure 1, which contains the basic planes of the SDN paradigm. The network

applications that formulate operation policies for the SD-ATN constitute the application plane of the SD-ATN. The control plane of the SD-ATN includes three control hierarchies: swarm control hierarchy, platform control hierarchy, and device control hierarchy. The data plane of the SD-ATN is composed of the SD-ATN transmission systems of different aircraft.

The swarm control hierarchy centrally controls the overall SD-ATN, and the controller that serves this control hierarchy is called the SD-ATN controller. The network applications all work on the SD-ATN controller. The platform control hierarchy is responsible for controlling the SDN-enabled avionics network (employing the SDN paradigm in the avionics network is a promising way to improve the efficiency of the avionics network [16, 17]), and it also works as a proxy to configure the heterogeneous SD-ATN transmission system according to the policies given from the swarm control hierarchy, which can simplify the network control operations and protocol design. The controller that serves this control hierarchy is called the platform controller. It is not always efficient and robust for the SD-ATN to rely on the logical centralized controller to manage the network. Therefore, the device control hierarchy embedded in the data plane is also defined in the SD-ATN, providing the SD-ATN transmission system with local network control logic to enable the SD-ATN to work in a distributed manner.

Each aircraft carries a platform controller. To improve the network robustness and facilitate the communications between the SD-ATN controller and the mission system (the latter purpose is helpful in formulating appropriate mission plans and network control strategies [13]), the SD-ATN controller is only carried on the AP-AEW&C, such as the E-3 and the KJ-2000. Each AP-AEW&C in an aviation swarm carries an SD-ATN controller (multiple SD-ATN controllers can work in a cluster mode), but only one AP-AEW&C works as the active network administrator. The remaining AP-AEW&Cs work as the standby network administrators in case the active network administrator is down. The details of the election and handover mechanisms that enable a new network administrator to work are not the focus of this article; thus, these mechanisms are not discussed. For ease of description, the AP-AEW&C working as the active network administrator is called the *active control node* and the remaining aircraft are called the *common nodes*.

Two M-info collection modes are considered: one is to make the active control node retrieves the M-info (“pull” mode) and the other is to make the common nodes proactively report its M-info to the active control node (“push” mode). Without loss of generality, information aggregation is not considered in this article. Moreover, how the data plane generates the M-info that needs to be reported to the active control node is ignored in this paper, since there are diverse methods can be used.

The M-info is sent in an omnidirectional and half-duplex transmission mode. For ease of protocol design, the SD-ATN is modeled as a three-dimensional undirected graph $G_3 = (V, E)$, where V denotes the set of nodes in the

network and $E = \{(i, j) \mid i, j \in V\}$ denotes the set of edges representing the wireless links. All the nodes in the network are assumed to have the same transmission range R . When designing the communication protocol, the R is taken as an empirical value, within which two nodes can communicate reliably in most cases. The protocol interference model is used to determine the transmission interference, in which concurrent transmission on two edges interfere with each other if and only if (1) the two edges are adjacent and (2) a receiver is within the interference range of a nonintended transmitter which is transmitting at the same time slot and frequency with the receiver’s receiving duration.

3. Framework for M-info Collection in the SD-ATN

The M-info is the source of knowledge for the SD-ATN controller to form the global network view; the reliability and latency of collecting M-info have a direct impact on the network performance [18]. Actually, besides the M-info, there are various types of information (e.g., voice, images, videos, commands, and sensor data) transmitted in SD-ATN and different types of information have different priorities and QoS requirements. If the M-info is transmitted mixed with the other types of information, its QoS requirements are hard to be guaranteed. Meanwhile, some M-info is hard to be collected in an in-band manner, for example, the M-info of the IFDL (Intra-Flight Data Link). Moreover, the M-info is not only critical for the network management but also helpful for battlefield situation awareness and mission planning, which attracts us to fully utilize its value. Therefore, we separate the M-info collection from the transmission of the other types of information, which makes it practical to give the M-info collection dedicated QoS guarantees.

As shown in Figure 2, we design a transmission framework called the *MCF-SD-ATN* for the M-info collection of the SD-ATN. In the *MCF-SD-ATN*, the platform controller collects the M-info from the network devices within the same platform and then submits the collected M-info to the SD-ATN transmission system for further transmission. The SD-ATN controller of the active control node collects the M-info generated by different common nodes and extracts the information useful for updating the global network view. Similar to the system framework presented in [19], a SD-ATN transmission system is composed of a common SD-ATN router and different radios. The SD-ATN router communicates with the radios through a common radio-to-router interface (R2RI), which improves the interoperability by separating the radio capability (one RF hop) from the router functionality (multihop) and is helpful for routing computing. The difference of the SD-ATN transmission system is that its transmission behaviors can be managed by the SD-ATN controller through the platform controller and the SBI; the implementation details are not discussed in this article. Because the RF-integrated technology and the software-defined radio technology allow for flexible utilization of hardware resources, a new

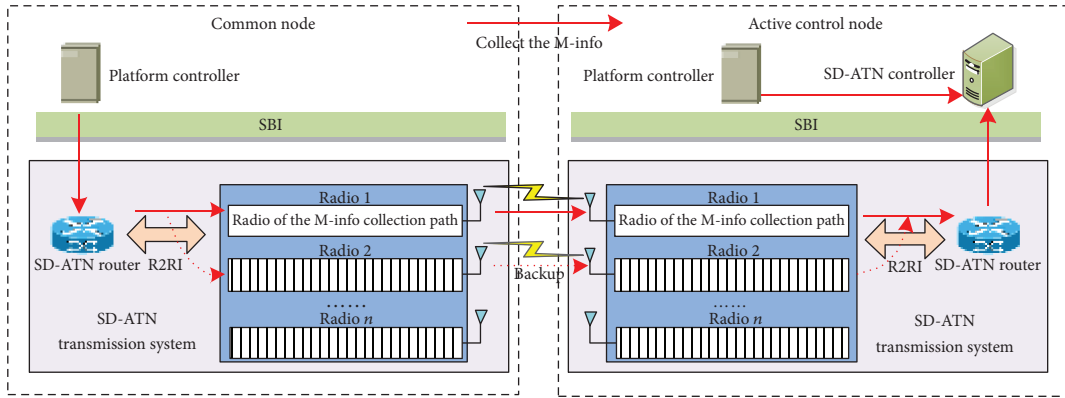


FIGURE 2: Overview of the MCF-SD-ATN.

transmission path that only focuses on the M-info collection is added to the SD-ATN transmission system, enabling the out-of-band collection process. The radio of the M-info collection path works within an independent frequency band. In the SD-ATN router, we add two types of dedicated queues for the new added transmission path: one type is used for caching the information that configures the M-info collection path and another type is used for caching the M-info. A dedicated routing table that stores the rules of forwarding the M-info is also constructed in the SD-ATN router.

In particular, to cope with the abnormalities of the new added transmission path, we also define a backup transmission path that is implemented by the off-the-shelf transmission capabilities of the SD-ATN transmission system and sends the M-info together with the tactical flows. For instance, the transmission path implementing the tactical targeting network technology (TTNT) can be employed as the backup transmission path; with the help of which, the M-info can still be collected normally even if the M-info collection path does not work.

By adding the M-info collection path in the SD-ATN transmission system, it is available to design a dedicated transmission solution for the M-info collection, which provides special QoS guarantees and prevents the M-info from being overwhelmed by the tactical flows. Meanwhile, the new added transmission path saves the bandwidth by making it easy to allocate channel resources according to the transmission demands of the M-info, while the needed bandwidth is difficult to learn if the M-info is transmitted together with the tactical information.

4. Design of MCP-SD-ATN

Based on the MCF-SD-ATN, we design a communication protocol called *MCP-SD-ATN* to make the active control node collect the M-info in a reliable and real-time manner. In this section, we elaborate the details of the MCP-SD-ATN.

4.1. Overview of the MCP-SD-ATN. In the general view of the SDN paradigm, the active control node should be the formulator of the M-info transmission policy of each node.

However, we find that it generates much overhead since many control packets that configure the M-info collection path of each node are needed to be broadcast to the whole network, which reduces the efficiency of collecting the M-info. Considering that different nodes can independently formulate a common M-info transmission policy if they have the same policy formulation processes and input information, we do not consider relying on the active control node to be the formulator and broadcaster of the M-info transmission policy. Instead, we make the active control node broadcast auxiliary information that is essential for formulating the M-info transmission policy to each common node, which significantly reduces the network control overhead. Any common node that receives the auxiliary information can independently formulate a common M-info transmission policy.

For the MCP-SD-ATN, we make the auxiliary information reflect the location and M-info traffic of each common node; the former can be used to construct the network topology by combining the transmission range of each node, and the latter indicates the number of M-info packets waiting to be collected from different common nodes. On the other hand, the active control node obtains each common node's location information and M-info traffic information from the M-info collected.

The MCP-SD-ATN is designed for ensuring the reliable and real-time M-info collection through the M-info collection path, and it enables the backup transmission path to collect M-info to cope with the exceptions of the M-info collection path. As shown in Figure 3, the MCP-SD-ATN divides the operations of collecting the M-info into two main phases, respectively, named preparation phase and M-info collection phase. The preparation phase is reserved for the common nodes to formulate its M-info transmission policy and configure the network devices according to the formulated policies. During the M-info collection phase, different common nodes transmit their M-info to the active control node according to the configured policies. The two phases are carried out periodically, and the time needed for one round of preparation phase and M-info collection phase is termed the *transmission period*. A transmission period's duration is determined according to the network status and M-info traffic size.

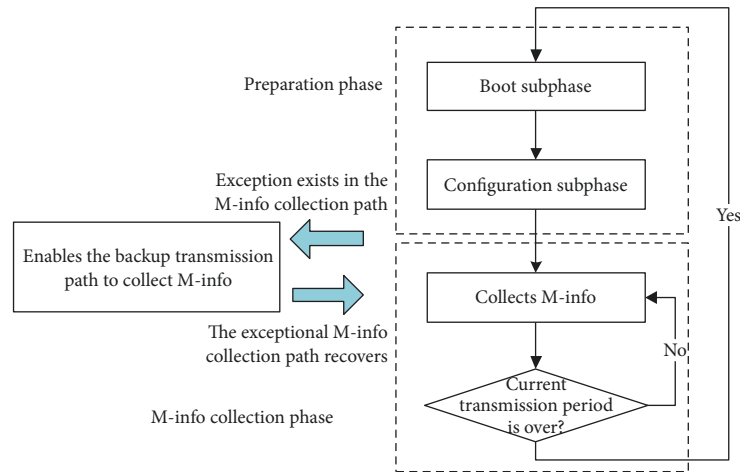


FIGURE 3: Overview of the MCP-SD-ATN.

The preparation phase is divided into the boot subphase and configuration subphase. During the boot subphase, the active control node broadcasts the boot packet (described in Section 4.2) to the whole network. Triggered by the boot packet, the prior works for the M-info collection during the current transmission period are carried out during the subphase. These prior works include (1) synchronizing the time slot ID of the whole network, (2) informing the common nodes of the active control node's M-info content demand, which is a critical step of the "pull" collection mode, and (3) informing the common nodes of the packet transmission policy during the preparation phase. During the configuration subphase, the active control node broadcasts the assistant packet (described in Section 4.2) that contains the auxiliary information to the whole network and the network configuration for M-info transmission is implemented in this subphase. There is overlap in time between the two subphases, which means that some nodes maybe in the boot subphase and some nodes are in the configuration subphase, which depends on if a node has received the boot packet or not during the preparation phase. When a common node's M-info collection path becomes exceptional, this node enables its backup transmission path to transmit its M-info to the active control node until the exception is eliminated.

To route a common nodes' M-info to the active control node, each common node constructs a common routing tree which is rooted at the active control node according to the auxiliary information received. It is well known that the contention-free multiple access technology named time division multiple access (TDMA) is better suitable for the periodic traffic and reliable transmission. Considering that the M-info collection process is approximately periodic and the reliability requirement of collecting M-info, the MCP-SD-ATN chooses the TDMA as the fundamental multiple access technology. Moreover, multichannel communication is an efficient method to eliminate interference and improve throughput by enabling concurrent transmissions over different channels; thus, the MCP-SD-ATN uses multichannel transmission technology to improve the transmission

capability. The multichannel transmission technology requires the radio of the M-info collection path to support the frequency hopping which is easy to be implemented.

The frame structure employed by the MCP-SD-ATN is shown in Figure 4. All the nodes in the network transmit packets over a common channel during the boot subphase, since the channel assignment result for broadcasting boot packet cannot be known to the common nodes previously. In contrast, the assistant packet and M-info packet can be transmitted over multiple channels.

Considering the transmission delay of the avionics network and the process delay of the controller, every time slot in the boot subphase has the same number of matched guard time slots and there are also guard time slots at the end of the configuration subphase, which provides sufficient time for policy formulation, network configuration, and M-info packet generation. Besides, there are also guard time slots at the end of each transmission period, and these guard time slots are used to provide for the SD-ATN controller sufficient time to process the collected M-info and encapsulate packets (boot packet and assistant packet) needed to be broadcast during the following transmission period. The number of the matched guard time slots is previously fixed according to the communication efficiency of the avionics network and the working efficiency of the controller.

When a transmission period is over, except the active control node, all the common nodes eliminate their current frame structure and routing tree. Then, they keep synchronization in units of time slot which has the same duration and switch the radio of the M-info collection path into reception status to receive the boot packet broadcast during the following transmission period. After receiving the boot packet, a common node can obtain the packet relaying policy during the preparation phase. Until receiving the assistant packet and after formulating the M-info transmission policy, a common node can construct the frame structure and the routing tree for the current transmission period.

4.2. Design of Packet. There are two types of packets broadcast during the preparation phase, which are named boot

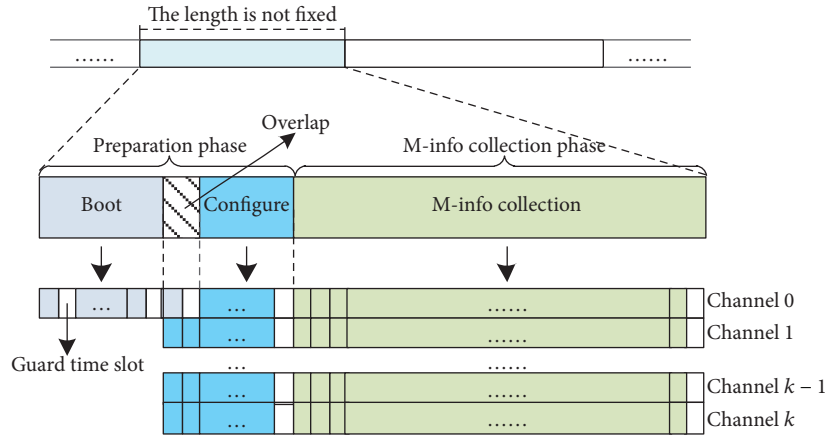


FIGURE 4: The frame structure employed by the MCP-SD-ATN.

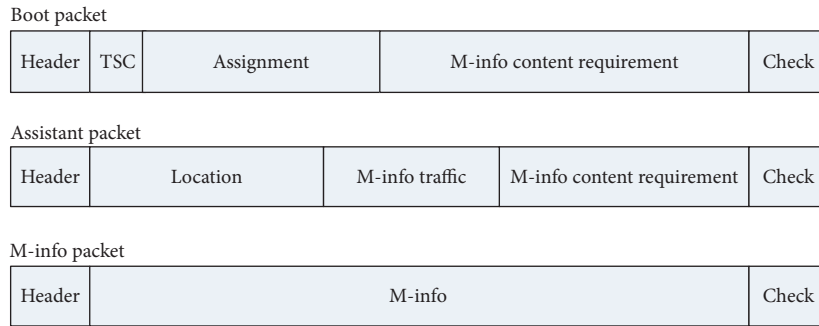


FIGURE 5: The structure of different packet types.

packet and assistant packet, respectively. The structure of the boot packet is shown in Figure 5. The time slot calibration (TSC) field is used to calibrate time slot ID of different common nodes; the assignment field stores the scheduling result for broadcasting the boot packet and assistant packet during the preparation phase, and the M-info requirement field announces the M-info content requirements of the active control node. According to the structure of the assistant packet shown in Figure 5, the location field is used to inform the locations of all nodes in the network and the M-info traffic field is used to inform the number of M-info packets needed to be collected from different common nodes. Moreover, the assistant packet also contains the M-info requirement field, in case that the boot packet cannot contain all the M-info content requirements of the active control node due to the packet size limitation. The M-info packet is used to carry the M-info of each common node, and the location information and M-info traffic information of a common node are also contained in the M-info packet. The packet type identifier and packet sequence number are added to each packet’s header to distinguish different packets received, and then, the corresponding operations will be carried out.

4.3. Scheduling for the Preparation Phase. To reduce the M-info collection latency, it is important to shorten the

duration of the preparation phase in each transmission period, which means that the boot packet and assistant packet can be broadcast to the whole network as soon as possible. We formulate the problem of how to quickly broadcast the boot packet and assistant packet to the whole network as the minimum latency broadcast scheduling (MLBS) problem [20]. According to Figure 4, the boot packet and assistant packet are broadcast over a common channel and multiple channels, respectively. The TDMA-based MLBS problem is proven to be NP-hard [20, 21]. To reduce the broadcast latency, we make the active control node first selects the part of the common nodes out to help relaying its packet to the whole network, which reduces the channel resources needed. These selected common nodes are termed the *broadcast node*. The broadcast node selection algorithm is given in Algorithm 1.

Before elaborating the following context, we summarize the notations used in the following algorithms in Table 1.

Initially, the common nodes in the network are divided into multiple levels according to its hop count to the active control node. The common nodes that are only one hop away from the active control node belong to the highest level. Algorithm 1 is iterated every two adjacent levels. For each time of iteration, the nodes in the higher level are temporarily

```

Input:  $G_3, C$ 
Output:  $BN$ 
1. Divide the common nodes in the network into multiple levels based
   on its hop count from the  $C$ ;
2. Obtain  $L$  and  $nL$ ;
3. initialize  $BN \leftarrow \phi, Q_1 \leftarrow \phi, Q_2 \leftarrow \phi, ll \leftarrow \phi, nll \leftarrow \phi, hl \leftarrow \phi, nhl \leftarrow \phi$ ;
4. for  $i \leftarrow 1:nL$ 
5.   Calculate  $ll, hl, nll, nhl$ ;
6.    $Q_1 \leftarrow Q_1 \cap hl, Q_2 \leftarrow Q_2 \cap ll$ ;
7.   for each  $v \in ll$ 
8.     if  $nhl(v) \leftarrow 1$ 
9.        $BN \leftarrow BN \cup hl(v)$ ;
10.      Remove  $ll(hl(v))$  and  $hl(v)$  from  $Q_2$  and  $Q_1$ , respectively;
11.      Update  $nll$  and  $nhl$ ;
12.     end if
13.   end for
14.   while  $Q_1 \neq \phi$  and  $Q_2 \neq \phi$ 
15.     Sort the nodes in  $Q_1$  in decreasing order according to  $nll$ ;
16.     Take out the first node say  $u$  in  $Q_1$ ;
17.      $BN \leftarrow BN \cup u$ ;
18.     Remove  $ll(u)$  and  $u$  from  $Q_2$  and  $Q_1$ , respectively;
19.     Update  $nll$  and  $nhl$ ;
20.   end while
21.    $Q_1 \leftarrow \phi, Q_2 \leftarrow \phi, ll \leftarrow \phi, nll \leftarrow \phi, hl \leftarrow \phi, nhl \leftarrow \phi$ ;
22. end for

```

ALGORITHM 1: Broadcast node selection.

stored in a buffer Q_1 and the nodes in the lower level are stored in another buffer Q_2 . The nodes, which are in Q_2 and have only one connected node in Q_1 , are found first, and these nodes' connected nodes in Q_1 are selected as the broadcast nodes. Then, the selected broadcast nodes along with all its connected nodes in the lower level are, respectively, removed from Q_1 and Q_2 , after which nhl and nll are updated. If there are nodes remaining in Q_1 and Q_2 , a greedy method is used to select as few broadcast nodes as possible. First, the remaining nodes in Q_1 are sorted in decreasing order according to the number of a node's connected nodes in Q_2 . Then, the first node in Q_1 is selected as the broadcast node, followed by the removal and update operations as above.

After selecting the broadcast nodes out, the channel resources of the preparation phase are assigned to the broadcast nodes by the active control node. To ensure the collision-free and efficient broadcast transmission, a channel/slot assignment algorithm is proposed. Before describing the details of the proposed algorithm, the broadcast constraints that ensure correct broadcast transmission are presented:

- (1) A broadcast node can be scheduled to send only if it has received a packet.
- (2) A broadcast node can be scheduled to send the assistant packet only if it has received the boot packet.
- (3) A broadcast node can be scheduled to send at a time slot if it is not scheduled to receive during the same time slot.

- (4) A broadcast node can be scheduled to send if it does not interfere with the reception of another node according to the protocol interference model.

In Algorithm 2, the common nodes in the network are also divided into multiple levels. Then, llb and $nllb$ are calculated according to G_3 and L_i . The broadcast of the boot packet is scheduled first. Initially, the active control node C is stored in a buffer Q . As long as Q is not empty, the nodes in it are sorted in decreasing order according to $nllb$, to make more time slots can be multiplexed since more broadcast node can be scheduled to send according to the first broadcast constraint. The node, which is in Q and has the largest number of connected broadcast nodes in the lower level, is scheduled first to send over the common channel. Meanwhile, the sending node's all connected nodes in the lower level are scheduled to receive. The node that is already scheduled to send is removed from Q , and the broadcast nodes, which are in the lower level and are connected with the scheduled node, are newly stored into Q . After scheduling the broadcast of the boot packet, llb and $nllb$ are recalculated and Q is initialized to include only C . Afterwards, the broadcast of the assistant packet is scheduled over multiple channels with the similar steps.

The scheduling result is stored in the assignment field of the boot packet. The platform controller of a common node that receives the boot packet determines how to process the received boot packet according to the information of the packet assignment field.

TABLE 1: List of notations used in the following algorithms.

C	Active control node
L_i	Set of nodes at level i
nL	Number of levels
BN	Set of broadcast nodes
$ll(u)$	Set of nodes that are in the lower level and connected with node u
$nll(u)$	Size of $ll(u)$
$hl(u)$	Set of nodes that are in the higher level and connected with node u
$nhl(u)$	Size of $hl(u)$
$llb(u)$	Set of the broadcast nodes that are in the lower level and connected with node u
$nllb(u)$	Size of $llb(u)$
T	Routing tree for collection the M-info
$B(k)$	Set of nodes that are in the higher level and included to branch k
Bl	Set of connections between the nodes in the lower level and different branches
M	Set of matched pairs between the branches and nodes
$wb(k)$	Weight of branch k
$wn(u)$	Weight of node u
$child(u)$	Set of receiver u 's children
D_u	Number of the M-info packets needed to be transmitted by receiver u
$Nei_2(u)$	Set of neighbors within receiver u 's two-hop distance
$I(u, v)$	Interference degree between node u and node v
$I'(u, v)$	Interference degree of node u on node v
$buffer(u)$	Number of the M-info packets remaining in node u 's buffer
$bc(u)$	Change status of $buffer(u)$

4.4. The M-info Collection Scheduling Problem. Shortening the duration of the preparation phase is not efficient enough to reduce the M-info collection latency. Therefore, we further try to shorten the duration of the M-info collection phase by carefully scheduling the transmission of the M-info packets, which is called *M-info collection scheduling problem*.

Lemma 1. *The M-info collection scheduling problem is NP-hard.*

Proof 1. Another problem that is similar to the M-info collection scheduling problem is first presented: Given a routing tree T on an arbitrary graph G and two positive integers p and q , is there an assignment of time slots to the edges in T using at most p frequencies such that the scheduling length is at most q ? This problem is proven to be NP-complete in [22].

Unlike the problem described in [22], the routing tree has been constructed in advance and the information aggregation is considered. For the M-info collection scheduling problem, a feasible routing tree rooted at the active control node needed to be constructed and each common node can send any number of M-info packets. Based on the facts above, if the routing tree rooted at the active control node is constructed in advance and any node of the SD-ATN can send no more than one M-info packet, the M-info collection scheduling problem is equal to the problem described in [22]. Since the problem described in [22] is a special case of the M-info collection scheduling problem, as a special case of the M-info collection scheduling problem is NP-hard, the general M-info collection scheduling problem is NP-hard.

We solve the M-info collection scheduling problem in two phases called *routing tree construction phase* and *channel/slot assignment phase*, respectively. Any common node's routing path to the active control node is determined by the routing tree construction phase, and the channel/slot assignment result for transmitting the M-info packets is given in the channel/slot assignment phase. Both of the two above phases are needed for M-info collection and have the nonneglected impacts on the M-info collection latency.

4.5. Routing Tree Construction Phase for the M-info Collection. The routing tree construction principle is formulated according to the Lemma 2.

Lemma 2. *If all the interfering links are eliminated, the scheduling length for the M-info collection phase is lower bounded by $\max(2D_{c-ch_k} - d_{c-ch_k}, D_c)$, where $c - ch_k$ denotes the node that needs to transmit the maximum number of M-info packets to C and is a child of C , D_c denotes the number of M-info packets needed to be collected by C during current transmission period, and d_{c-ch_k} denotes the number of M-info packets generated by node $c - ch_k$ itself.*

Proof 2. Let D_{c-ch_i} denote the number of M-info packets needed to be transmitted by C 's child i . Sorting C 's children in nonincreasing order according to D_{c-ch_i} , which is represented by $D_{c-ch_k} \geq D_{c-ch_{k-1}} \geq \dots \geq D_{c-ch_1}$. Since no node can receive multiple packets simultaneously, C needs at least D_c time slots to collect all the M-info packets. For child node k , it needs at least D_{c-ch_k} time slots to send the M-info packets to C and $D_{c-ch_k} - d_{c-ch_k}$ time slots to receive the M-info packets from its children. Because of the half-duplex transmission mode, the time slots assigned to node k for sending must be distinct from its children's sending time slots, which makes node k need at least $D_{c-ch_k} + (D_{c-ch_k} - d_{c-ch_k})$ time slots in total. Therefore, the schedule length for the M-info collection phase is lower bounded by $\max(2D_{c-ch_k} - d_{c-ch_k}, D_c)$.

To reduce the number of time slots occupied by the M-info collection phase, we intend to construct a routing

Input: G_3, BN, C
Output: channel/slot assignment result of broadcasting the boot packet and assistant packet

1. **Initialize** $llb \leftarrow \phi, nllb \leftarrow \phi, Q \leftarrow Q \cap C$;
2. Calculate $llb, nllb$ and ll ;
3. Schedule all the common nodes to be idle;
4. **while** $Q \neq \phi$, schedule the broadcast of boot packet
5. Sort the nodes in Q in decreasing order according to $nllb$;
6. According to the broadcast constraints, take the first node say u from Q and schedule it to send over the common channel;
7. $Q \leftarrow Q \cup llb(u)$;
8. Remove u from Q ;
9. Update llb and $nllb$;
10. **end while**
11. **if** $Q = \phi$
12. Recalculate llb and $nllb$;
13. $Q \leftarrow Q \cap C$;
14. **end if**
15. **while** $Q \neq \phi$
16. According to the broadcast constraints, schedule the broadcast of the assistant packet with the similar steps;
17. **end while**

ALGORITHM 2: Channel/slot assignment for the preparation phase.

tree that reduces the number of M-info packets transmitted by each child node of the active control node, that is,

$$\min \left\{ \max_{\forall c-ch_i \in ll(c)} \{D_{c-ch_i}\} \right\}. \quad (1)$$

The routing tree construction operations are presented in Algorithm 3. Similar to Algorithm 1, the nodes in the network are also divided into multiple levels. We define *branch* which is the tree rooted at one child node of the active control node. The number of the branches equals to the number of C 's children. To reduce the number of the M-info packets transmitted by each child of C , we try to balance the M-info packets transmitted via different branches. According to Algorithm 3, the routing tree is constructed in a top-down approach by iterating every two adjacent levels. A branch only includes one different child of C during the first iteration, and the nodes in the lower level (these nodes will belong to the higher level in the next iteration) are included into different branches at the end of this iteration.

For each time of iteration, Bll is computed according to the ll and B and the nodes in the lower level are stored in a buffer Q . After computing Bll , the nodes, which are in Q and connected to only one branch, are found first (e.g., u_1 in Figure 6). Then, these nodes are matched with their connected branches and removed from Q . The matching results are stored in M . A branch that has been matched is marked as occupied to indicate that it cannot be matched any more, and then the weight of the matched branch is updated, which equals to the number of the M-info packets generated by the nodes that are in the higher level and belong

to the branch and the nodes that are in the lower level and matched to the branch. Then, the remaining nodes in Q are sorted in decreasing order according to its weight, for example, $wn(v)$ which equals to the number of the M-info packets generated by v and the nodes belong to $ll(v)$. The nodes in Q are taken out in order to match the branches according to the principle described in Algorithm 3 (lines 12–18). Specifically, the augmenting path (e.g., $\{u_3, B(2), u_2, B(3)\}$ in Figure 6(b)) is the path that contains a matched pair, a nonmatched node, and a nonmatched branch. When the augmenting path is found, the matched pair in the augmenting path is removed from M and the nonmatched node and nonmatched branch are, respectively, matched to the branch and node that belong to the previous matched pair in the augmenting path, which is followed by the update of M and wb .

Through the processes above, the nodes in the lower level are all matched to different branches. Then, the next hop of a node which needs to forward the M-info to the active control node and is in the lower level can be found from the connected nodes that belong to its matched branch. Since a node has to be scheduled to avoid interference not only from other children of its parent in the routing tree but also the parent's other neighbors [23], the node with a minor number of neighbor nodes is preferred as the next hop destination, breaking ties based on the node ID. After finding the connections between the nodes in two adjacent levels, the nodes in the lower level become the nodes in the higher level and are included to different branches according to the branches they are matched with. B is updated to make any branch only includes the nodes newly included.

Input: G_3, C
Output: T

1. **Initialize** $B, wb, Q \leftarrow \phi$;
2. **for** $i \leftarrow 1 : nL$
3. Store the nodes that belong to the lower level into a buffer Q ;
4. Calculate Bll according to ll and B ;
5. According to Bll , find the nodes that are in Q and connected to only one branch;
6. Match the found nodes with the connected branches;
7. Remove the matched nodes from Q and mark the matched branches as occupied;
8. Update M and wb ;
9. Sort the nodes in Q in decreasing order according to w_n ;
10. **while** $Q \neq \phi$
11. Take out the first node in Q , say u ;
12. **if** there is a nonoccupied branch that has the minimum weight and is connected with u (breaking ties based on the branch ID)
13. Match u with this branch;
14. **else if** there is an augmenting path of which u is an end vertex, and the total weight of the augmenting path can be reduced by matching u
15. Match u with the branch in the augmenting path and change the previous matched pairs;
16. **else**
17. Match u with the branch that has minimum weight (breaking ties based on the branch ID);
18. **end if**
19. mark the matched branch as occupied;
20. update M and wb ;
21. remove u from Q ;
22. **end while**
23. For any node in the lower level, it should select its connected node that belongs to the matched branch and has the minimum number of neighbors as its next hop.
24. Update $B, Q \leftarrow \phi$;
25. **end for**

ALGORITHM 3: Construction of routing tree.

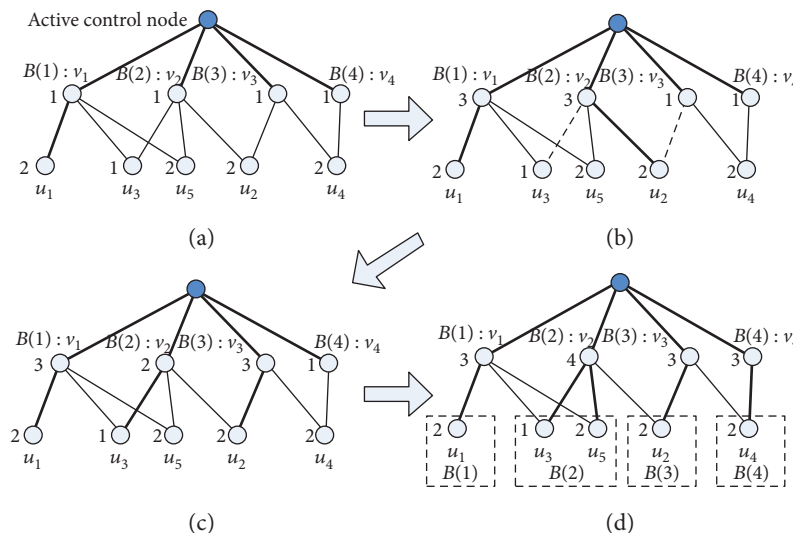


FIGURE 6: Example for routing tree construction.

```

Input:  $G_3, T, M$ -info traffic information, receiver  $u$ 
Output: interference map of the receiver  $u$ 
1. Get the set of children of each receiver;
2. Get the number of  $M$ -info packets that are needed to transmit by
   different node;
3. Get the set of two-hop neighbor receivers of the receiver  $u$ ;
4. Initialize  $Q_1 \leftarrow \phi, Q_2 \leftarrow \phi, Q_3 \leftarrow \phi, I' \leftarrow 0, I \leftarrow 0$ ;
5.  $Q_1 \leftarrow Q_1 \bigcap Nei_2(u)$ ;
6. while  $Q_1 \neq \phi$ 
7.   Take out a node say  $v$  from the  $Q_1$ ;
8.    $Q_2 \leftarrow Q_2 \bigcup child(u), Q_3 \leftarrow Q_3 \bigcap child(v)$ ;
9.   while  $Q_2 \neq \phi$  (calculate the  $I'(u, v)$  first)
10.    Take out a node say  $w$  from the  $Q_2$ ;
11.    if the node  $w$ 's sending interferes with the reception of the
        node  $v$ 
12.       $I'(u, v) \leftarrow I'(u, v) + D_w$ ;
13.      Remove the  $w$  from the  $Q_2$ ;
14.      if  $I'(u, v) \geq D_v - d_v$ 
15.         $I'(u, v) \leftarrow D_v - d_v$ ;
16.        break;
17.      end if
18.    else
19.      Remove the  $w$  from the  $Q_2$ ;
20.    end if
21.  end while
22.  while  $Q_3 \neq \phi$ 
23.    calculate the  $I'(v, u)$  via the same steps of calculating the  $I'(u, v)$ ;
24.  end while
25.   $Q_2 \leftarrow \phi, Q_3 \leftarrow \phi$ ;
26.   $I(u, v) \leftarrow \max \{I'(u, v), I'(v, u)\}$ ;
27.  Remove the  $v$  from the  $Q_1$ ;
28. end while

```

ALGORITHM 4: Interference map construction.

4.6. Channel/Slot Assignment Phase for the M -info Collection.

After constructing the routing tree, the channel resources are assigned to the common nodes. The operations of assigning the channel resources are divided into two phases. First, we assign different channels to the receivers in the routing tree, which makes the children of a common receiver send over the same channel. Considering that the spectrum resources in the battlefield are strained, the number of channels that can be assigned to the receivers is limited. To multiplex more time slots, we try to minimize the interference between the receivers that are assigned to receive over the same channel. Afterwards, according to the local time slot assignment algorithm in [24], we design a buffer change-based time slot assignment algorithm and prove that this algorithm can achieve the lower bound described in Lemma 2 if all the interfering links are eliminated.

In order to ensure the collision-free M -info transmission, we need to respect the following two constraints termed *transmission constraints*:

- (1) A node can be scheduled to send at a time slot if it is not scheduled to receive during the same time slot.

- (2) A node can be scheduled to send if it does not interfere with the reception of another node according to the protocol interference model.

For assigning channels to the receivers in an optimal manner, different receivers independently construct a common interference map according to the common auxiliary information received, which reflects the interference degree between different receivers. The steps of calculating the interference degree between two receivers are presented in Algorithm 4. First, the children of each receiver and the number of the M -info packets needed to be transmitted by each receiver are both obtained according to the routing tree constructed. Moreover, the reception process of a receiver in the routing tree can only be interfered by the receptions of the receivers within two-hop distance. Therefore, the set of neighbor receivers within the two-hop distance is also required in our algorithm. The number of the M -info packets that can be interfered is used to measure the interference degree between two receivers. As shown in Figure 7, for a receiver u and a receiver v within u 's two-hop distance, $I'(u, v)$, which denotes the interference degree of u on v , is first computed according to the principles described in

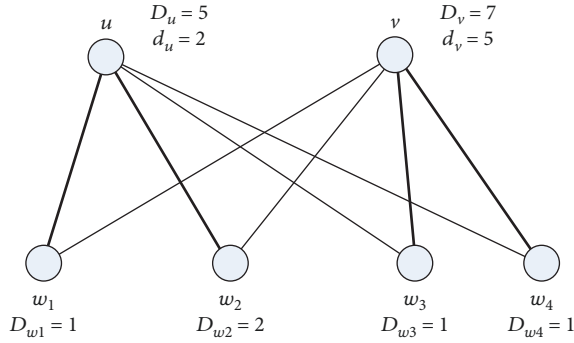


FIGURE 7: Example for measuring interference degree.

Algorithm 4 (lines 9–21). Then, $I'(v, u)$ is computed via the same steps of computing the $I'(u, v)$. The maximum value of $I'(u, v)$ and $I'(v, u)$ is taken as the interference degree between the node u and node v . In Figure 7, $I'(u, v) = D_v - d_v = 2$ since $D_{w_1} + D_{w_2} \geq D_v - d_v$ and $I'(v, u) = D_{w_3} + D_{w_4} = 2$ since $D_{w_3} + D_{w_4} \leq D_u - d_u$. Then, we get the interference degree between u and v , which equals to 2 according to $\max \{I'(u, v), I'(v, u)\}$ that equals to 2.

The channels are assigned to the receivers according to the interference map. For an assignment result m , we use $m(f_k)$ to denote the set of receivers that are scheduled to receive at the channel k . Thus, the interference degree of channel k can be expressed as

$$L_m(f_k) = \sum_{u, v \in m(f_k), u \neq v} I(u, v). \quad (2)$$

The goal of our channel assignment algorithm is to find a channel assignment result to achieve

$$\min \{ \max_{m, k} \{ L_m(f_k) \} \}, \quad (3)$$

and the details of the proposed channel assignment algorithm are presented in Algorithm 5.

In Algorithm 5, we calculate SI for each receiver, where

$$SI(u) = \sum_{v \in \text{Nei}_2(u)} I(u, v). \quad (4)$$

The receivers in Q are sorted in decreasing order according to SI. For each time, we take out a receiver say u from Q in order and find the channel that has the minimum interference degree when the receiver u is assigned to receive over this channel. Then, the receiver u is assigned to receive over this channel with all its children are assigned to send over the same channel. Once all the receivers in the routing tree have been assigned channels according to Algorithm 5, the sending channel and receiving channel of all nodes in the network are determined.

Afterwards, we give the buffer change-based time slot assignment algorithm to assign the time slots. In Algorithm 6, $\text{bc}(u) = 0$ indicates that the size of $\text{buffer}(u)$ is nondecreased

and $\text{bc}(u) = 1$ indicates that the size of $\text{buffer}(u)$ is decreased. For a given time slot, it is first considered to be assigned to the active control node's child whose buffer is not empty and which has the maximum number of M-info packets needed to be transmitted (lines 8–15). If this time slot is not assigned, the time slot is then considered to be assigned to the node that is not the child of the active control node (lines 16–23). If a node is scheduled to send, the number of the M-info packets in the node's buffer decreases and that of the corresponding receiver increases. For a node that is not the child of C , if the number of the M-info packets in its buffer is decreased at a time slot, one of its children will be scheduled to send during the next time slot.

Lemma 3. For a common node which is not the child of the active control node, if it is identified as decreased, this node always has a parent and a child that are identified as nondecreased.

Proof 3. For a given time slot say $t = 1$, the lemma is trivially true since the buffer size of all common nodes is initialized to be nondecreased. Suppose the lemma holds for the time slot $t = k (k \geq 1)$, at the time slot $t = k + 1$, one of the following two cases should occur:

- (1) $\text{bc}(u) = 0$, meanwhile $\text{bc}(u_p) = 1$ and $\text{bc}(u_c) = 1$.
- (2) $\text{bc}(u) = 0$ and $\text{bc}(u_p) = 0$, meanwhile $\text{bc}(u_c) = 1$.

For the first case, we show that both u_p and u_c have a parent and a child that are identified as nondecreased. It is clear that u_p has a child that is identified as nondecreased since $\text{bc}(u) = 0$. Similarly, u_p also has a father that is identified as nondecreased, since u_p sends an M-info packet to its parent. It is also clear that the parent of u_c is identified as nondecreased since $\text{bc}(u) = 0$. If u_c has a child (denoted by u_{gc}) that is identified as nondecreased at time slot k , then u_{gc} is also identified as nondecreased at time slot $k + 1$ since $\text{bc}(u_c) = 0$ at time slot k . Otherwise, if $\text{bc}(u_{gc}) = 1$ at time slot k , then $\text{bc}(u_{gc}) = 0$ at time slot $k + 1$ since one child of u_{gc} sends an M-info packet to u_c at time slot $k + 1$. For the second case to happen, the parent of u_p (denoted by u_{gp}) is identified either decreased or nondecreased at time slot $k + 1$. If $\text{bc}(u_{gp}) = 1$, the parent of u_{gp} is identified as nondecreased since u_{gp} transmits an M-info packet to its parent at time slot $k + 1$ and there is a child of u_f is also identified nondecreased since $\text{bc}(u) = 0$. If $\text{bc}(u_{gp}) = 0$, u , u_f , and u_{gp} are all identified as nondecreased at time slot $k + 1$, $\text{bc}(u_c) = 1$, and $\text{bc}(u_{gc}) = 1$ as we show in the first case. Therefore, the lemma holds for time slot $k + 1$, and the proof follows.

Lemma 4. If all the interfering links are eliminated, the schedule length achieved by Algorithm 6 equals to the lower bound mentioned in Lemma 2.

Input: G_3, I, T , all available channels
Output: channel assignment result m

1. **Initialize** $Q \leftarrow \phi$;
2. Get the receivers in T and store them in Q ;
3. Get SI according to the interference map;
4. **while** $Q \neq \phi$
5. Sort the receivers in Q in decreasing order according to SI ;
6. Take out the first receiver say u from Q ;
7. Find the channel that still has the minimum interference degree if u is assigned to receive over this channel. Assigning u to receive over the found channel and the children of u to send over the same channel;
8. Remove u from Q ;
9. Update the interference degree of each channel
10. **end while**

ALGORITHM 5: Assignment of channels.

Input: G_3, T, C, M -info traffic information
Output: time-slots assignment result of M -info collection

1. **initialize** $Q_1 \leftarrow \phi, Q_2 \leftarrow \phi, bc \leftarrow 0$;
2. $Q_1 \leftarrow Q_1 \cup \text{child}(c)$;
3. Store the nodes in the network into Q_2 except C ;
4. **while** there is a node's buffer is not empty
5. **if** there is node in Q_1 whose buffer is not empty
6. Sort the nodes in Q_1 in decreasing order of the number of M -info packets needed to be transmitted;
7. Take out a node say u from Q_1 , which has a nonempty buffer;
8. Schedule link (u, c) respecting the transmission constraints;
9. $bc(u) \leftarrow 1$;
10. $D_u \leftarrow D_u - 1$;
11. $buffer(u) \leftarrow buffer(u) - 1$;
12. **end if**
13. **if** there is a node in Q_2 , which has a nonempty buffer;
14. Find a node say v in Q_2 whose buffer is identified as decreased;
15. Find a child node of v say w whose buffer is not empty, and schedule link (w, v) respecting the transmission constraints;
16. $buffer(w) \leftarrow buffer(w) - 1$;
17. $buffer(v) \leftarrow buffer(v) + 1$;
18. $bc(w) \leftarrow 1$;
19. $bc(v) \leftarrow 0$;
20. **end if**
21. **end while**

ALGORITHM 6: Buffer change-based time slot assignment.

Proof 4. According to Lemma 3, for a child node of the active control node, if it sends an M -info packet during the current time slot and all the interfering links are eliminated, it receives another M -info packet from its children during the next time slot. Thus, if the active control node has at least two child nodes of which the M -info packets have not been all transmitted, the active control node can receive an M -info packet during every time slot. Suppose $D_{c-ch_k} \geq D_{c-ch_{k-1}} \geq \dots \geq D_{c-ch_1}$, considering the following two cases:

$$(1) D_{c-ch_k} - d_{c-ch_k} \geq \sum_{i=1}^{k-1} D_{c-ch_i}.$$

$$(2) D_{c-ch_k} - d_{c-ch_k} < \sum_{i=1}^{k-1} D_{c-ch_i}.$$

For the first case, $\max(2D_{c-ch_k} - d_{c-ch_k}, D_c) = 2D_{c-ch_k} - d_{c-ch_k}$ and $c - ch_k$ needs $D_{c-ch_k} - d_{c-ch_k}$ time slots to receive M -info packets from its children. Since $D_{c-ch_k} - d_{c-ch_k} \geq \sum_{i=1}^{k-1} D_{c-ch_i}$ and the active control node keeps receiving, the children of the active control node except $c - ch_k$ can transmit all the M -info packets they are required to transmit during the receiving time slots of $c - ch_k$. Therefore, the

schedule length is bounded by the time slots needed by $c - ch_k$ and is equal to $2D_{c-ch_k} - d_{c-ch_k}$ since $c - ch_k$ is scheduled to send every two time slots. For the second case, $\max(2D_{c-ch_k} - d_{c-ch_k}, D_c) = D_c$, the active control node keeps receiving M-info packets during every time slot until there is no M-info packet remaining in any nodes' buffer, which makes the schedule length equals to D_c . Therefore, the lemma follows.

Each node in the network accomplishes all the above scheduling processes independently according to the auxiliary information obtained.

4.7. Operation Details of the MCP-SD-ATN. We develop a global MCP-SD-ATN module for the SD-ATN controller and a local MCP-SD-ATN module for the platform controller to implement the operations of the MCP-SD-ATN, which are, respectively, performed by the SD-ATN controller and platform controller.

At the end of a transmission period, the global MCP-SD-ATN module extracts the auxiliary information from the received M-info and formulates the transmission policy for the following transmission period. Then, the local MCP-SD-ATN module within the same platform configures the network devices according to the transmission policy formulated. The general MCP-SD-ATN module also generates the boot packet and assistant packet; it stores the generated packets into the dedicated queue that stores the configuration information of the M-info collection path. Each common node in the network switches the radio of its M-info collection path to reception status to receive the boot packet broadcast from the active control node. During the preparation phase, the local MCP-SD-ATN module of the common node takes the responsibility of processing the received boot packet and assistant packet. The local MCP-SD-ATN module formulates transmission policies according to the received packet and makes the platform controller configure the SD-ATN transmission system to implement the transmission policies. During the M-info collection phase, according to the M-info transmission policy formulated, a common node transmits its M-info packets to the active control node via the M-info collection path. The operation details of the MCP-SD-ATN are shown in Figure 8.

For the SD-ATN, packet loss is unavoidable since the links of the SD-ATN are unstable and intermittent, which should be considered in the MCP-SD-ATN. For the MCP-SD-ATN, the packet loss cases that impact include (1) loss of the boot packet, (2) loss of the assistant packet, and (3) loss of the M-info packet that carries the auxiliary information that indicates the status of the sending node. The above cases may cause a common node to unable to successfully transmit its M-info packets to the active control node. We term the common node that cannot successfully send its M-info packets to the active control node as *missing node* in this article.

To cope with the exceptions caused by the packet loss cases, the first step is to perceive the exceptions. For the local MCP-SD-ATN module, an exception is perceived if the local MCP-SD-ATN module cannot receive a boot packet for a

fixed number of time slots during the preparation phase or it cannot receive an assistant packet at the assigned time slot. The general MCP-SD-ATN module perceives an exception if it cannot receive the M-info packet that carries the auxiliary information of the sending node. Thus, the general MCP-SD-ATN module assigns none time slot to the corresponding common node. Then, the local MCP-SD-ATN module of the common node can perceive the exception when it cannot find a time slot assigned to itself from the boot packet received. When an exception is detected, the local MCP-SD-ATN module of the missing node switches the radio of its M-info collection path to the reception status. Meanwhile, the backup transmission path of the missing node is enabled to transmit its M-info packets to the active control node. If the active control node receives the missing node's auxiliary information, it tries to make the missing node transmit the M-info packets via the M-info collection path again by scheduling for the missing node. Particularly, the missing node does not stop transmitting via the backup transmission path until it is twice scheduled.

5. Performance Evaluation

In this section, simulation results are given based on EXata 5.1 to illustrate the performance of our solution.

5.1. Simulation Scenario. As shown in Figure 9, we build a simplified avionics network in EXata 5.1 for every aircraft and regard an avionics network as a node of the SD-ATN. The simplified avionics network, which is regarded as the common node, includes two devices that implement the functions of the platform controller and SD-ATN transmission system, respectively. The avionics network that represents the active control node includes one more device that implements the functions of the SD-ATN controller. The devices within an aircraft are connected through wired links. The 802.3 MAC protocol and the open shortest path first routing protocol (OSPF) are used to ensure efficient communications between the different devices.

The general MCP-SD-ATN module and local MCP-SD-ATN module are implemented in the devices that represent the SD-ATN controller and platform controller, respectively. A simple SBI that implements the configuration functions needed by the platform controller is designed referred to the OpenFlow protocol. Since an aircraft is represented by its avionics network in our experiment, a new random waypoint mobility model is developed to simulate the mobility of different aircraft, which makes the aircraft move in a three-dimensional random way, meanwhile the devices belong to the same aircraft remain relatively stationary.

The experiment scenario dimension is set in Cartesian coordinate, where X and Y are both equal to 6×10^5 meters and the altitude equals 1×10^4 meters above sea level. In the aviation swarm deployed in our simulation scenario, only one aircraft is the active control node and the remaining aircraft are the common nodes. Some aircrafts are statically deployed in the scenario to ensure the connectivity between the aviation swarm members from the topology view. The minimum speed of an aircraft is set to 200 m/s. Three

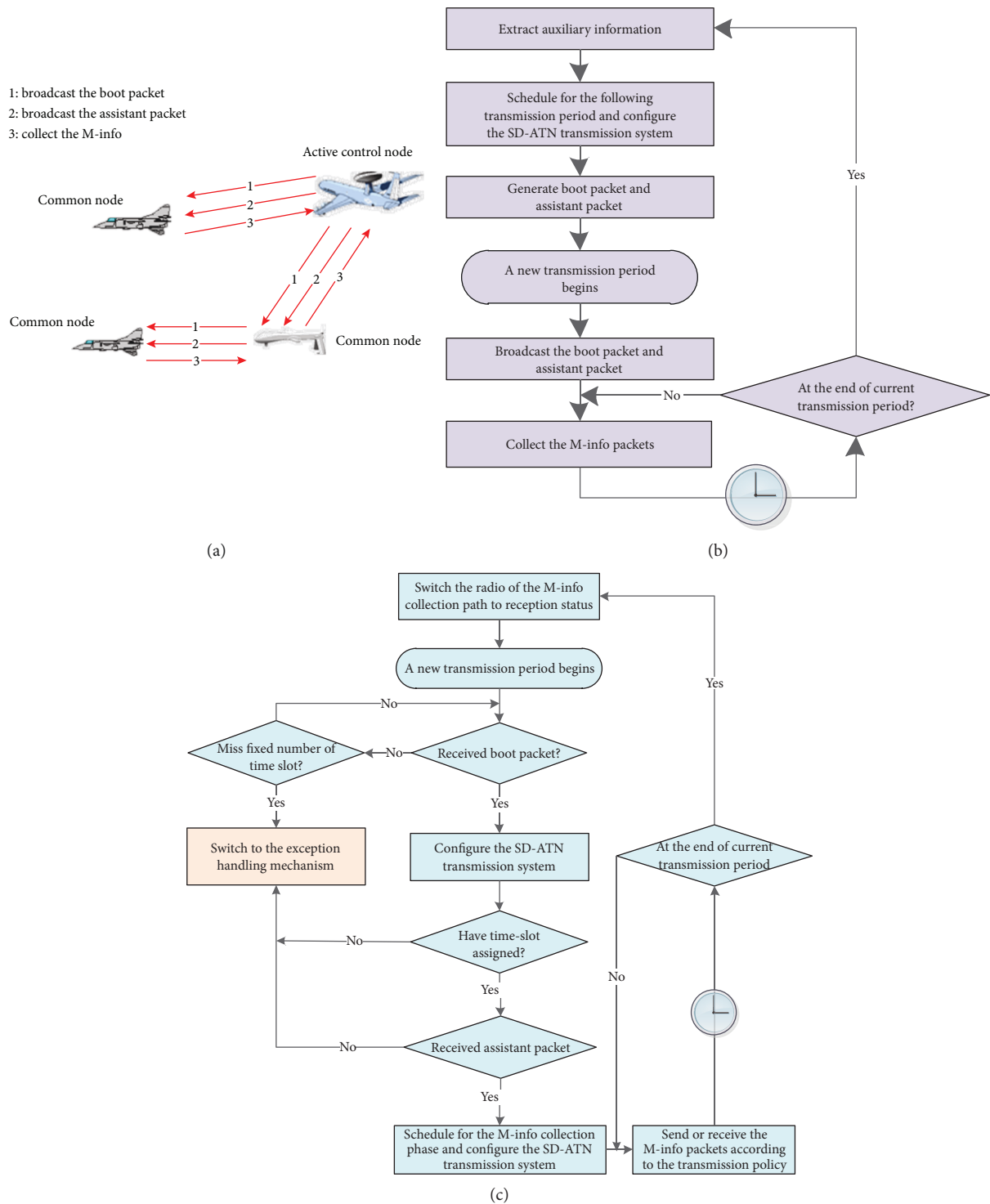


FIGURE 8: Operation details of the MCP-SD-ATN. (a) General operations. (b) Operation details of the active control node. (c) Operation details of the common node.

channels are allocated to the M-info collection path, and the data rate of a channel is set to 2 Mbps. The duration of a time slot is set to 2.5 ms. The bandwidth of the avionics network's wired links is set to 1 Gbps. The aeronautical channel model [25] is used to simulate the aviation channel environment.

First, based on the proposed transmission framework MCF-SD-ATN, the reliability and real-time performance of

the designed communication protocol MCP-SD-ATN are simulated and the simulation results are compared with the performance performed by the combinations of classic routing and MAC protocols, which is also simulated based on the MCF-SD-ATN. Afterwards, we simulate the reliability and real-time performance of the system which integrates the MCF-SD-ATN and the MCP-SD-ATN and compares the

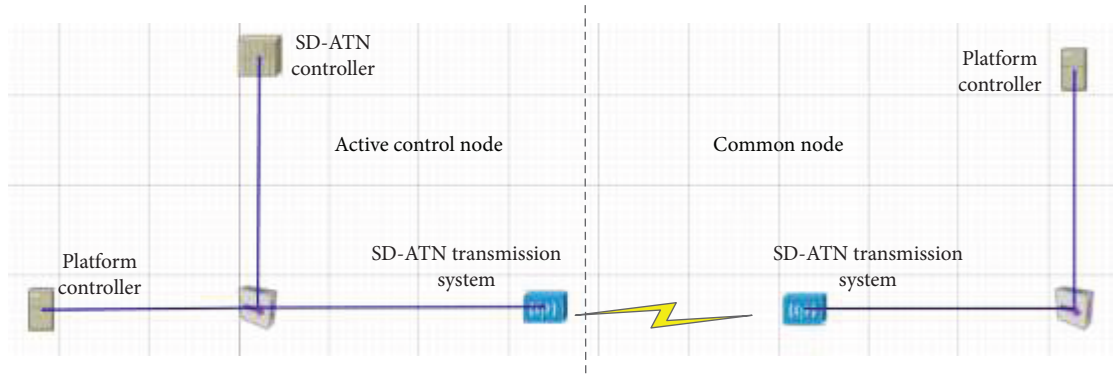


FIGURE 9: Example for simulation scenario.

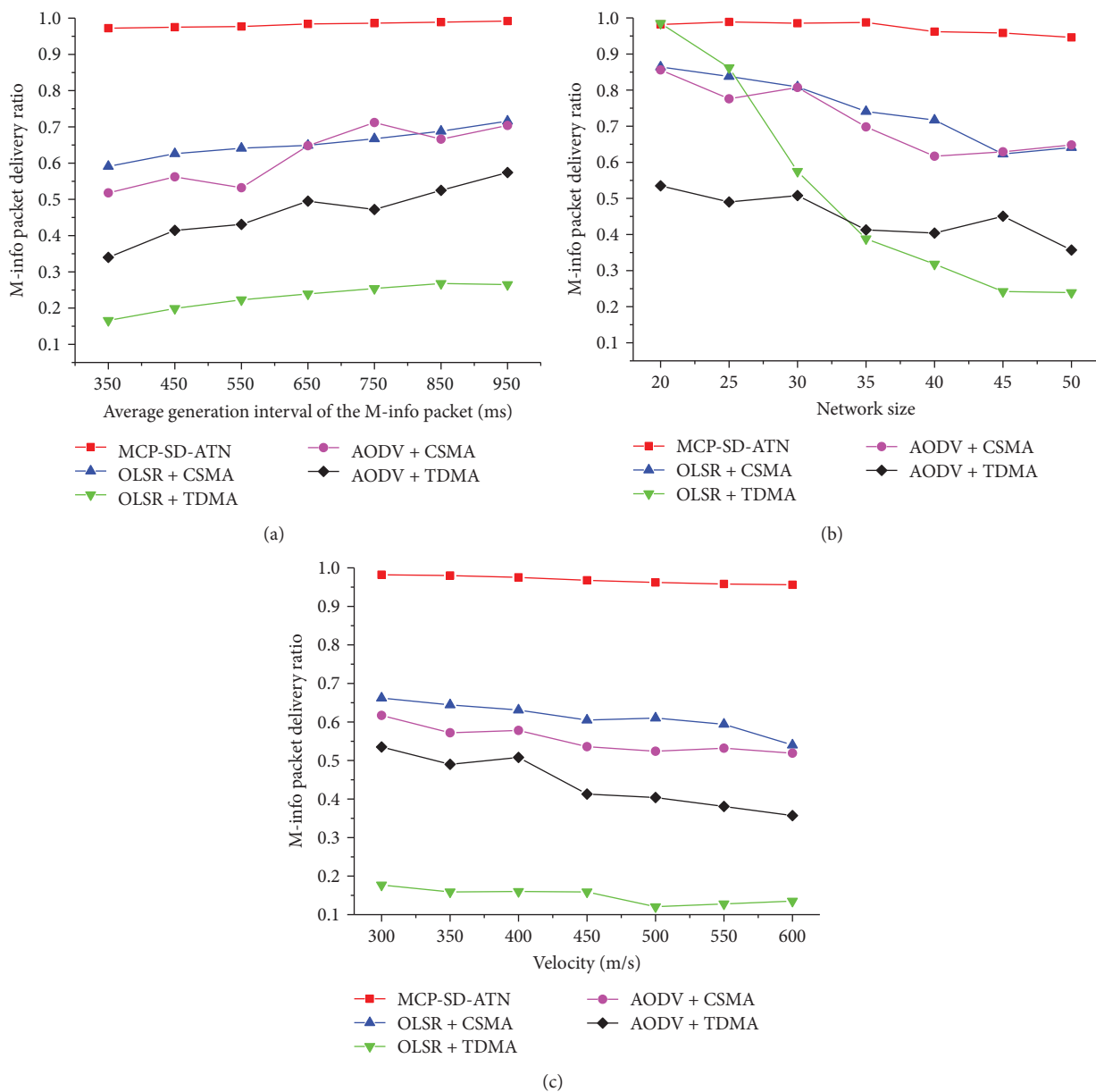


FIGURE 10: M-info packet delivery ratio of the MCP-SD-ATN. (a) With respect to the M-info traffic. (b) With respect to the network size. (c) With respect to the aircraft's moving velocity.

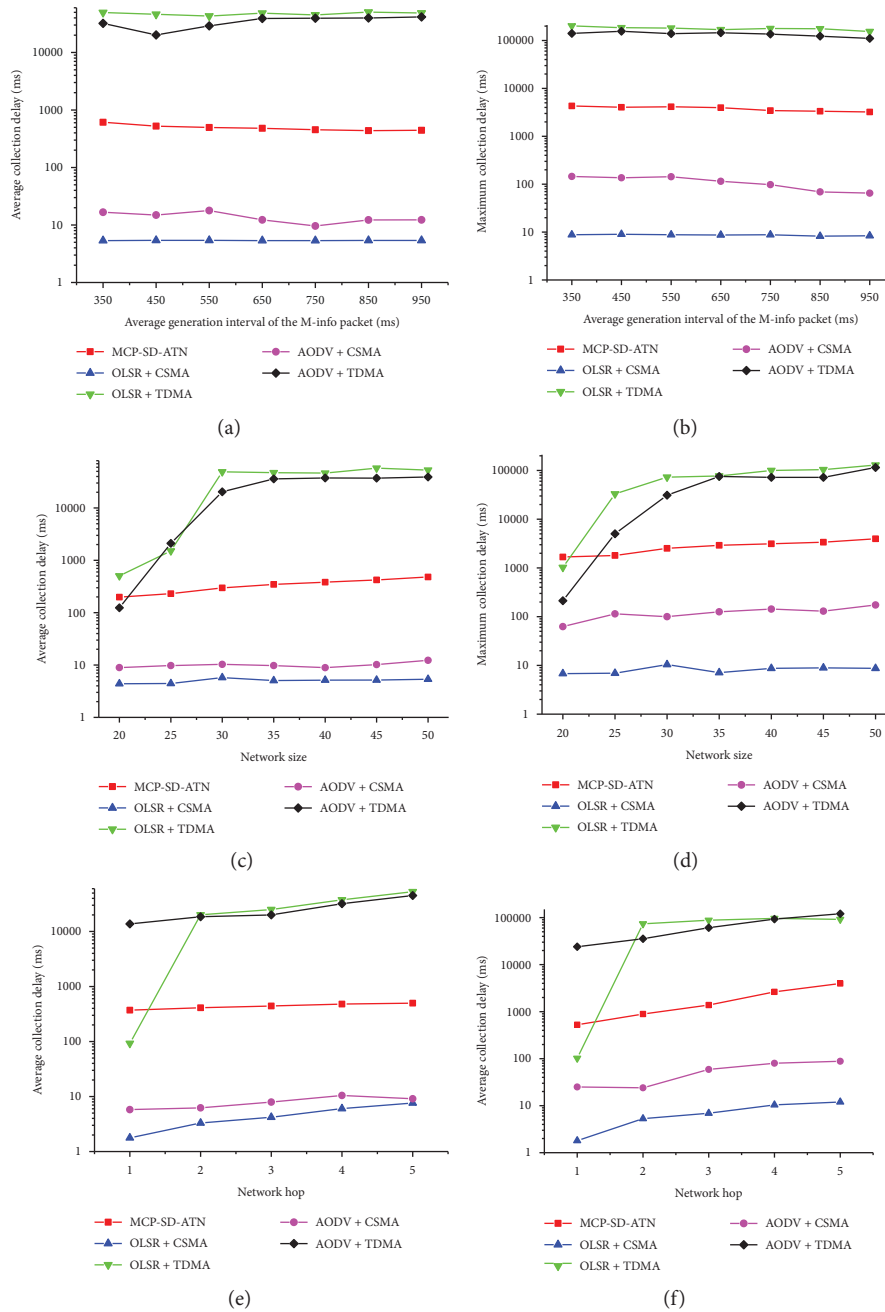


FIGURE 11: M-info collection delay. (a) Average collection delay with respect to the M-info traffic. (b) Maximum collection delay with respect to the M-info traffic. (c) Average collection delay with respect to the network size. (d) Maximum collection delay with respect to the network size. (e) Average collection delay with respect to the network hop count. (f) Maximum collection delay with respect to the network hop count.

simulation results with the widely used military communication systems of today’s ATN.

5.2. Simulation Results. The reliability performance of the MCP-SD-ATN is first simulated. Figure 10 shows the M-info packet delivery ratio of the MCP-SD-ATN. We observed that the MCP-SD-ATN showed better reliability performance under different circumstances. According to Figure 10, with the increase of the M-info traffic, the M-info packet delivery ratio of the MCP-SD-ATN is always higher than 95% and the classic protocols can

only achieve a delivery ratio lower than 80%. When the network size (the number of common nodes in the network is used to characterize the network size) is small, the M-info packet delivery ratio of the OLSR-TDMA combination is better (OLSR, optimized link state routing), but it decreases dramatically as the network size expands. Compared with the other combinations of classic protocols, the M-info packet delivery ratio of the MCP-SD-ATN is at least 10% higher under different network sizes. Although the increase of the aircraft’s velocity increases the changing frequency of network topology, the M-info packet delivery ratio of the

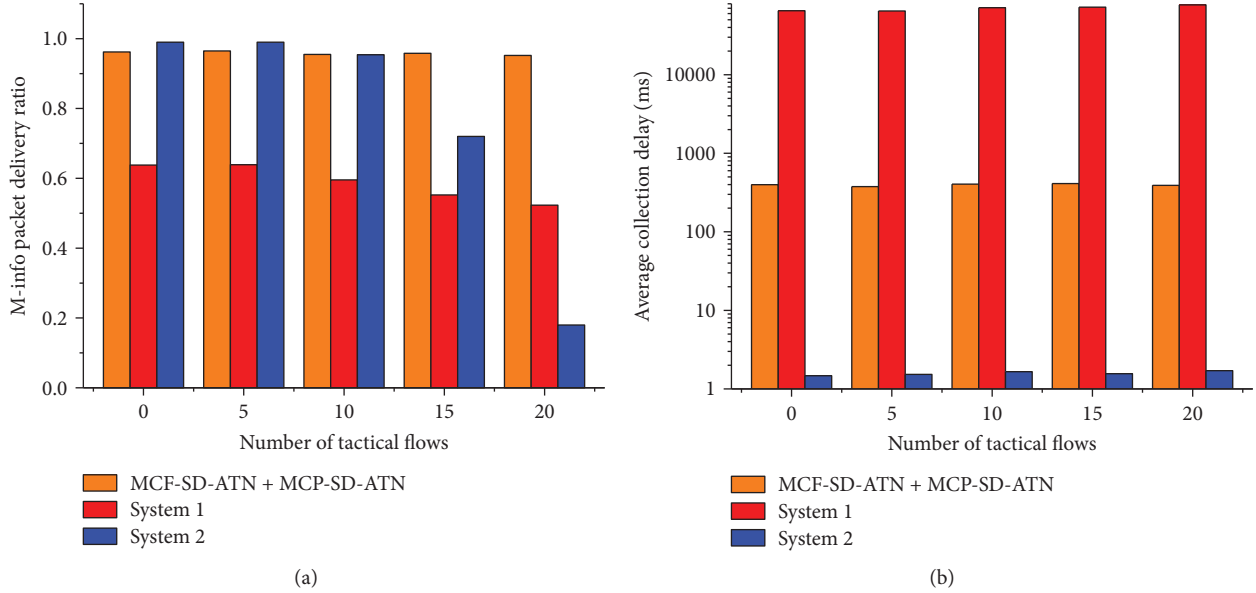


FIGURE 12: M-info collection performance of different communication schemes. (a) Performance of the M-info packet delivery ratio. (b) Performance of the average M-info collection latency.

MCP-SD-ATN is maintained higher than 95%, which means a good adaptability to network topology changes.

Figure 11 shows the M-info collection latency of the MCP-SD-ATN. It can be seen from Figure 10 that a very low M-info collection latency can be achieved if the CSMA (carrier sense multiple access) protocol is employed, which keeps the average M-info collection delay under 50 ms and the maximum M-info collection delay under 200 ms. It is because that the random channel access mechanism of the CSMA protocol makes a packet transmitted to the channel as soon as possible. But the CSMA protocol cannot address the packet collision problem, which reduces the M-info packet delivery ratio as shown in Figure 10. For the MCP-SD-ATN, it keeps the average M-info collection delay under 1 s and the maximum M-info collection delay under 5 s. Although the real-time performance of the MCP-SD-ATN is not as good as the CSMA, it outperforms the TDMA protocol and ensures the reliable M-info collection. Further, we noted that the network's hop count had great impact on the maximum M-info collection delay of the MCP-SD-ATN, which means that the M-info collection delay jitter of the MCP-SD-ATN can be effectively reduced by limiting the network hop count.

In general, the MCP-SD-ATN helps the active control node to collect the M-info from the whole network in a relatively reliable and real-time way. This good performance mainly contributes to the following reasons:

- (1) The MCP-SD-ATN requires less transmission overhead for routing computation compared to the OLSR and AODV (ad hoc on-demand distance vector) protocols. Moreover, compared to the CSMA or TDMA protocol, the MCP-SD-ATN reasonably schedules the use of the channel resources, which makes the

MCP-SD-ATN take full advantage of the reliability merit of TDMA technology with no packet overflow.

- (2) There is a backup transmission path in the MCP-SD-ATN to cope with the exceptions of the M-info collection path, and the MCP-SD-ATN exploits the capability of the backup transmission path, which improves the robustness of collecting the M-info.

To illustrate the superiority of our scheme (MCP-SD-ATN and MCF-SD-ATN) in collecting the M-info, we simulate the proposed scheme's performance from a system view, and the performance of two other communication systems that are widely employed in today's ATN are simulated for comparison; because the two communication systems for comparison are for military domain, only the general characteristics are listed:

- (1) System 1: a communication system that employs the TDMA protocol and achieves the multihop transmission by configuring fixed relaying nodes.
- (2) System 2: a communication scheme that employs the TxRx^N waveform [26], random access technology, and proactive routing protocol.

For the system 1 and system 2, the M-info is transmitted mixed with other types of information like the tactical information and the priorities of different information types are not considered. Under the premise of the same channel resource consumption, Figure 11 shows simulation results obtained from a 2-hop network and takes the number of tactical flows as the variable.

According to Figure 12, the M-info packet delivery ratio of the system 1 is lower than 65% and the M-info collection latency is higher than 60s since it needs a long waiting time

for system 1 to relay the M-info packets to the active control node. Thus, the system 1 is not suitable for collecting the M-info. When the number of tactical flows is small, the system 2 has the highest M-info packet delivery ratio that is higher than 99% and the lowest M-info collecting latency lower than 2 ms. When the tactical traffic becomes higher, the M-info packet delivery ratio of the system 2 drops severely since the collision of packets becomes serious. Thus, the M-info collection performance of the system 2 can be seriously affected by the interaction of other information types. Compared to the system 1 and system 2, our scheme collects the M-info in a relatively reliable and real-time way, which protects the transmission of the M-info from being interfered by the other information types. In general, our scheme is more appropriate for the M-info collection in the SD-ATN.

6. Conclusion

The aviation swarm needs an ATN that provides more flexible and efficient communication capability. To design the ATN that satisfies the communication demands of the aviation swarm, applying the SDN paradigm to the ATN architecture is a promising way owing to the SDN's flexibility, programmability, and openness features. Remarkably, one of the fundamental problems of constructing the SD-ATN is how to ensure that the SD-ATN's control plane collects the M-info from the data plane reliably and timely in the aviation environment.

To address this issue, we propose a transmission framework called the MCF-SD-ATN for collecting the M-info in the SD-ATN and design a communication protocol called the MCP-SD-ATN based on the MCF-SD-ATN. The MCF-SD-ATN makes it practical to provide dedicated QoS guarantees for collecting M-info; meanwhile, the MCP-SD-ATN employs the communication capability provided by the MCF-SD-ATN and has good reliability and real-time performance at collecting M-info.

Generally, the work in this article is carried out under a basic SD-ATN architecture, which makes it a fundamental research work for satisfying the SD-ATN's M-info collection requirements. In future work, we will further optimize our framework and protocol by considering more holonomic network architecture.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 91638101 and Grant 61472443.

References

- [1] B.-N. Cheng, F. J. Block, B. R. Hamilton et al., "Design considerations for next-generation airborne tactical networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 138–145, 2014.
- [2] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [3] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.
- [4] S. Jain, M. Zhu, J. Zolla et al., "B4: experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.
- [5] T. Bilen, B. Canberk, and K. R. Chowdhury, "Handover management in software-defined ultra-dense 5G networks," *IEEE Network*, vol. 31, no. 4, pp. 49–55, 2017.
- [6] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: challenges and design requirements," *IEEE Access*, vol. 5, no. 5, pp. 1872–1899, 2017.
- [7] X. Ge, Z. Li, and S. Li, "5G software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017.
- [8] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.
- [9] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: traffic matrix estimator for OpenFlow networks," in *Passive and Active Measurement*, pp. 201–210, Springer, Berlin, Heidelberg, 2010.
- [10] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: monitoring network utilization with zero measurement cost," in *Passive and Active Measurement*, pp. 31–41, Springer, Berlin, Heidelberg, 2013.
- [11] N. L. M. van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: network monitoring in OpenFlow software-defined networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–8, Krakow, Poland, 2014.
- [12] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "Pay-Less: a low cost network monitoring framework for Software Defined Networks," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–9, Krakow, Poland, 2014.
- [13] M. Moshref, M. Yu, and A. Vahdat, "SCREAM: sketch resource allocation for software-defined measurement," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies - CoNEXT '15*, p. 14, Heidelberg, Germany, 2015.
- [14] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "DREAM: dynamic resource allocation for software-defined measurement," in *Proceedings of the 2014 ACM Conference on SIGCOMM - SIGCOMM '14*, pp. 419–430, Chicago, IL, USA, 2014.
- [15] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, "One sketch to rule them all: rethinking network flow monitoring with UnivMon," in *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*, pp. 101–114, Florianopolis, Brazil, 2016.

- [16] Z. Li, Q. Li, L. Zhao, and X. Hiong, "Openflow channel deployment algorithm for software-defined AFDX," in *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, pp. 4A6-1-4A6-10, Colorado Springs, CO, USA, 2014.
- [17] P. Heise, F. Geyer, and R. Obermaisser, "Deterministic OpenFlow: performance evaluation of SDN hardware for avionic networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 372-377, Barcelona, Spain, 2015.
- [18] M. Aslan and A. Matrawy, "On the impact of network state collection on the performance of SDN applications," *IEEE Communications Letters*, vol. 20, no. 1, pp. 5-8, 2016.
- [19] B. N. Cheng, R. Charland, P. Christensen, L. Veytser, and J. Wheeler, "Evaluation of a multihop airborne IP backbone with heterogeneous radio technologies," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 299-310, 2014.
- [20] L. You, Z. Yuan, B. Tang, and G. Chen, "Minimum latency broadcast scheduling in single-radio multi-channel wireless ad-hoc networks," 2013, <http://arxiv.org/abs/1304.5330v1>.
- [21] I. Chlamtac and S. Kutten, "On broadcasting in radio networks - problem analysis and protocol design," *IEEE Transactions on Communications*, vol. 33, no. 12, pp. 1240-1246, 1985.
- [22] A. Ghosh, O. D. Incel, V. S. A. Kumar, and B. Krishnamachari, "Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pp. 363-372, Macau, China, 2009.
- [23] J. Gagnon and L. Narayanan, "Efficient scheduling for minimum latency aggregation in wireless sensor networks," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1024-1029, New Orleans, LA, USA, 2015.
- [24] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86-99, 2012.
- [25] E. Haas, "Aeronautical channel modeling," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 254-264, 2002.
- [26] S. M. Clark, K. A. Hoback, and S. J. F. Zogg, "Statistical priority-based multiple access system and method," US Patent 7680077, 2010.



Hindawi

Submit your manuscripts at
www.hindawi.com

