Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 000 (2017) 1–25

www.elsevier.com/locate/jfranklin

A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers

Weijing Qi[a], Qingyang Song[a,*], Xiangjie Kong[b,*], Lei Guo[a]

[a] *School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China*
[b] *Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China*

Received 1 May 2017; received in revised form 6 September 2017; accepted 11 November 2017
Available online xxx

Abstract

Recently, Flying Ad-hoc Sensor Networks (FASNETs), which are basically ad hoc networks between Unmanned Aerial Vehicles (UAVs), have been playing a great role in many sensing fields. To further improve the utilization of network resources, researchers offer the possibility of applying Software Defined Networking (SDN) technology to FASNETs, enabling various applications to be supported over the same platform. Since these concurrently executed applications might have diverse Quality of Service (QoS) requirements, it brings new challenges to network management and resource scheduling. In this paper, we propose a novel clustering FASNET architecture with SDN cluster controllers and also a collaborative controller, in order to realize hierarchical management and unified dispatch. Based on our designed architecture, we also propose a centralized *traffic-differentiated routing* (TDR) executed in each cluster, which aims to guarantee the specific QoS requirements of delay-sensitive and reliability-requisite services. Different weights are assigned to various flows according to their sensitivity to delay and also levels of importance. In particular, we introduce a transmission reliability prediction model to TDR, in which we consider both link availability and node's forwarding ability. Simulation results show our proposed TDR has good performances in terms of end-to-end delay, packet dropping ratio and network throughput.

* Corresponding authors.
E-mail addresses: songqingyang@cse.neu.edu.cn (Q. Song), xjkong@ieee.org (X. Kong).

https://doi.org/10.1016/j.jfranklin.2017.11.012
0016-0032/

1. Introduction

Unmanned Aerial Vehicles (UAVs) have received widespread adoption for military and civilian purposes due to their versatility, flexibility and relatively small operating expenses. With various onboard sensors, UAVs have been playing great roles in the fields of reconnaissance, forestry fire monitoring, Earth science research, volcanic gas sampling, humanitarian observations, etc. [1]. Instead of developing and operating one large UAV, using a group of small UAVs shows a tremendous advantage in terms of scalability, survivability and also efficiency [2]. Recent developments in relevant communication technologies make Flying Ad-hoc Sensor Networks (FASNETs), which are basically ad hoc networks composed of UAVs, become possible.

In some realistic scenarios, multiple FASNETs may need to be deployed for respective sensing tasks in the same area. Without sharing common physical infrastructures, different service providers develop their application-specific FASNETs in an isolated manner, leading to high deployment cost. In order to facilitate flexible deployment of new services and improve the utilization of network resources, Gupta et al. [3] showed the possibility of applying Software Defined Networking (SDN) to FASNETs. These SDN-FASNETs with multiple types of sensors enable various on-demand sensing tasks to be executed over the same network platform by loading different programs. Since these simultaneous applications might have diverse Quality of Service (QoS) requirements, it brings new challenges to the network management and resource scheduling, especially from the perspective of routing designs. In particular, for a heavily loaded network with limited resources, an unreasonable routing scheme may cause traffic congestions, resulting in degradation of network performance. Take Fig. 1 as an example, and suppose there are two types of packets from source nodes to the gateway: delay-sensitive and reliability-requisite ones. In Fig. 1(a) and (b), packets are routed without being differentiated. As a result, in the Shortest Path Tree (SPT) routing, the network may have bottleneck nodes, causing massive losses of integrity-requisite packets and long end-to-end delays of delay-sensitive packets. While for the multipath routing in Fig. 1(b), although the suboptimal paths relieve the cache pressure of the bottleneck nodes, the long paths will hardly guarantee QoS for real-time services [4–6].

To make full use of the limited network resources and satisfy the QoS requirements of each individual application, active research efforts have been done on providing traffic-differentiated services in Wireless Sensor Networks (WSNs). The current works mainly adopt the

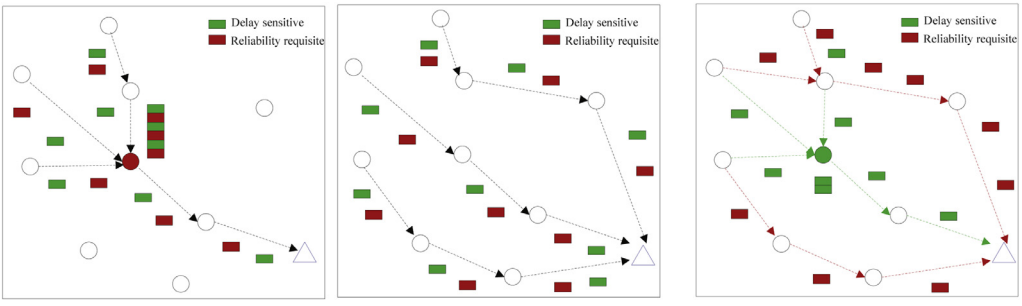


Fig. 1. Illustration of network traffic under different routing methods.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

strategy that packets with different QoS requirements are routed in different paths, as shown in Fig. 1(c). Many of them consider delay and reliability as QoS metrics. Qian et. al. considered stabilization of systems with varying delay [7,8]. Specifically, the delay includes transmission delay along links and also queuing delay in nodes' caches. The delay-sensitive traffic is generally arranged to have lower queuing delay. For example, the EDCA mechanism of the IEEE 802.11e protocol provides four kinds of queues with various priorities: the AC0 and AC1 queues with high priority are for real-time video services, the AC2 and AC3 queues with low priority are for non-sensitive FTP/Email [9]. One drawback is that it works only within a one-hop range. For the aspect of reliability, the current link reliability estimation methods mainly includes Exponential Weighted Moving Average (EWMA) [10–13], Window Mean Exponential Weighted Moving Average (WMEWMA) [11,12,14], and Expected Transmission Count (ETX) [15–18], etc. Unfortunately, a lot of additional probe messages need to be sent during these distributed estimation processes, leading to the increase in node energy consumption. Besides, only the link quality is taken into account, but node forwarding failures caused by malicious nodes' selfishness are not considered [19,20].

To solve the above challenges, in this paper, we propose a novel hierarchical FASNET architecture with SDN cluster controllers and also a collaborative controller. Based on our designed architecture, we also propose a centralized *traffic-differentiated routing* (TDR) executed in each cluster. Specifically, in order to increase the network scalability, the UAVs are grouped into several cluster domains in our design, each of which is controlled by an upper stationary airship. The controllers are in charge of acquiring the whole abstract cluster network views, implementing the unified scheduling of the network resources and also guiding the data processing and forwarding. There is also a collaborative airship controller which aims to realize the interactions between the single-domain controllers. Once multiple modules are written on the application layer, our designed FASNET architecture may perform different tasks at the same time. On each cluster network, we run a data-driven optimization algorithm TDR to update traffic routing paths periodically. The objective is to minimize the total cost of all traffic flows in the current time, which is related to the delay and reliability. In particular, to guarantee the transmission reliability, the methods for link availability and node's forwarding ability are introduced to TDR. We assign different weights to various flows according to their sensitivity to delay and also levels of importance, in order to guarantee the specific QoS requirements of delay-sensitive and reliability-requisite services. The key contributions are listed as follows:

- We design a hierarchical FASNET architecture with SDN cluster controllers and a coordination controller, which shows a tremendous advantage in terms of scalability, flexibility, and efficiency.
- We propose a transmission reliability prediction model, in which we consider both link availability and node's forwarding ability. An incentive mechanism is added to the nodes' interactive evaluation to avoid malicious nodes.
- Based on the proposed transmission reliability prediction model, we design a centralized TDR algorithm, in order to ensure the respective QoS of different applications.

The following paper is organized as follows. In Section 2, we review the related work concerning SDN architecture for WSNs and traffic differentiated routing in WSNs. In Section 3, an overview of our proposed hierarchical FASNET architecture with SDN cluster controllers is described. The traffic differentiated routing problem in FASNET is described in Section 4.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

In Section 5, a forwarding reliability prediction model is introduced. And we propose the TDR algorithm running in the FASNET cluster in Section 6. Simulation results are shown in Section 7 before conclusion and prospect of future work in Section 8.

2. Survey of current work

WSN is a relatively mature research field. As reference, we review the related works concerning SDN architecture for WSNs and traffic differentiated routing in WSNs here.

2.1. SDN architecture for WSNs

Recently, ideas of applying the SDN architecture to WSNs have been proposed. Early attempts to make networking protocols in WSNs programmable have been done in [21]. In [22,23], the SDN was taken as a way to tackle inherent problems in WSNs such as the difficulty in changing network management strategies. Later on, the advantages of the SDN approach in WSNs have been discussed in [24]. As a new WSN paradigm, the concept of Software-Defined Sensor Network (SDSN) was introduced in [25]. Zeng et al. studied the evolution of the SDSN, and also integrated the sensor nodes into cloud computing using the Sensing-as-a-Service (SaaS) model. Different from the existing SDN solutions for WSNs, the proposed SDN-WISE approach in [26] has reduced the amount of information exchanged between sensor nodes and the SDN controller. Based on SDN-WISE, Anadiotis et al. [27] proposed a comprehensive solution to support MapReduce function in WSNs, leading to a significant reduction of communication cost such as energy consumption. Apart from the architecture design and implementation, Some algorithms have been proposed to improve network performance in SDSNs. In [28], taking advantage of the SDN paradigm, Fotouhi et al. proposed the traffic monitoring and load balancing algorithms. Based on the global traffic information, the proposed protocol can significantly reduce packet loss. Some sleep scheduling algorithms in SDSNs were introduced in [29–31] to reduce network overhead and prolong network lifetime. Besides, it is worth mentioning that a software architecture of UAV-based sensor networks was proposed in [32,33]. This is an early attempt to combine UAV, WSN, and SDN. *However, there is little literature relating to hierarchical or clustering SDN architecture in WSNs, which is conducive to network management and expansion.*

2.2. Traffic differentiated routing in WSNs

Due to the existence of service differentiation in WSNs, QoS aware routing has drawn many attentions and some research has been done. As early as 2003, Akkaya and Younis [34] first proposed an energy-aware QoS routing protocol in order to ensure efficient usage of the sensors and effective access to the gathered measurements. The protocol found a delay-constrained path for real-time data and also maximized the throughput for non-real-time data by assigning various r-value, i.e., bandwidth ratios to them. The protocol was extended in [35] by a multi-r mechanism, where each node calculates its own r-value according to the information obtained from the gateway, achieving a better utilization of the link resources. To solve the excessive overhead and energy consumption when sending r-value to each node, Hamid et al. in [36] improved the above schemes by locally adjusting the bandwidth and delay requirement based on the incoming traffic and path length. Furthermore, besides timeliness, reliability is taken into account to differentiate traffic types in [10,11,37,39]. In [10], QoS

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

levels are guaranteed by multiple data delivery speed options and also probabilistic multi-path forwarding. Besides reliability and delay, the Distributed Aggregate Routing Algorithm (DARA) [37] considers residual energy in the routing metric. A set of candidate nodes with higher transmission powers are selected to route critical packets. Based on [38], Djenouri and Balasingham [11] further considered transmission power as QoS metric. They adopted EWMA for link latency estimation considering queuing time and transmission delay while in DARA only queuing time is considered. Recently, Zhang et al. [39] designed a novel integrity and delay-differentiated routing algorithm, based on the concept of potential field in the discipline of physics. The algorithm improves the fidelity of high-integrity data and decreases the average delay of delay-sensitive data. Since only local information is required, it provides good scalability. *However, in these distributed routing algorithms, the selected path may not be optimal since the overall information is not comprehensive. Numerous probe packets have to be sent in the link parameter estimation, which increases network load and affects the network throughput. Besides, node forwarding failures caused by malicious nodes' selfishness are not considered. The aforementioned challenges are the major considerations of the proposed TDR algorithm in this paper.*

3. Network architecture

3.1. Motivation

Traditionally, once the infrastructure of a network is deployed, its functionality remains changeable. However, along with the rapid increase of application demands in this information-intensive world, more diversified and complicated services need to be provided, which requires FASNETs to be flexible, extensible and dynamically upgradable. Besides, the limited carrying load of UAV nodes raises higher requirement on the effective utilization of nodes' resources such as computation and communication.

At this point, the concept of Software Defined Networking (SDN) turns up correspondingly. The main idea is to decouple the control plan and data plan, so that the network traffic could be controlled flexibly. It also provides open interfaces for users to customize their network services arbitrarily. Aside from the SDNs advantages in terms of simplifying hardware structure, realizing centralized management and improving network resource utilization, the cluster architecture also enhances the network scalability. In addition, deploying multiple SDN controllers overcomes the problems caused by the single-controller mode, such as traffic congestions and high risk of network breakdown [40]. Controllers being carried by stationary airships also decrease the overhead for topology management.

3.2. SDN-based FASNET architecture

In this paper, we propose a novel hierarchical FASNET architecture based on the SDN technology, as shown in Fig. 2. In the following we will introduce it from the the respective of the three layers in SDN architecture: data layer, control layer, and application layer.

The data layer reflects the deployment of the whole network infrastructure. In our design, to increase the network scalability, the UAVs are grouped into several clusters, each of which is controlled by an upper stationary airship. Laser (when pointing) or radio communication links may be established between various devices, in order to implement data forwarding, according to the flow table acquired from the controller.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

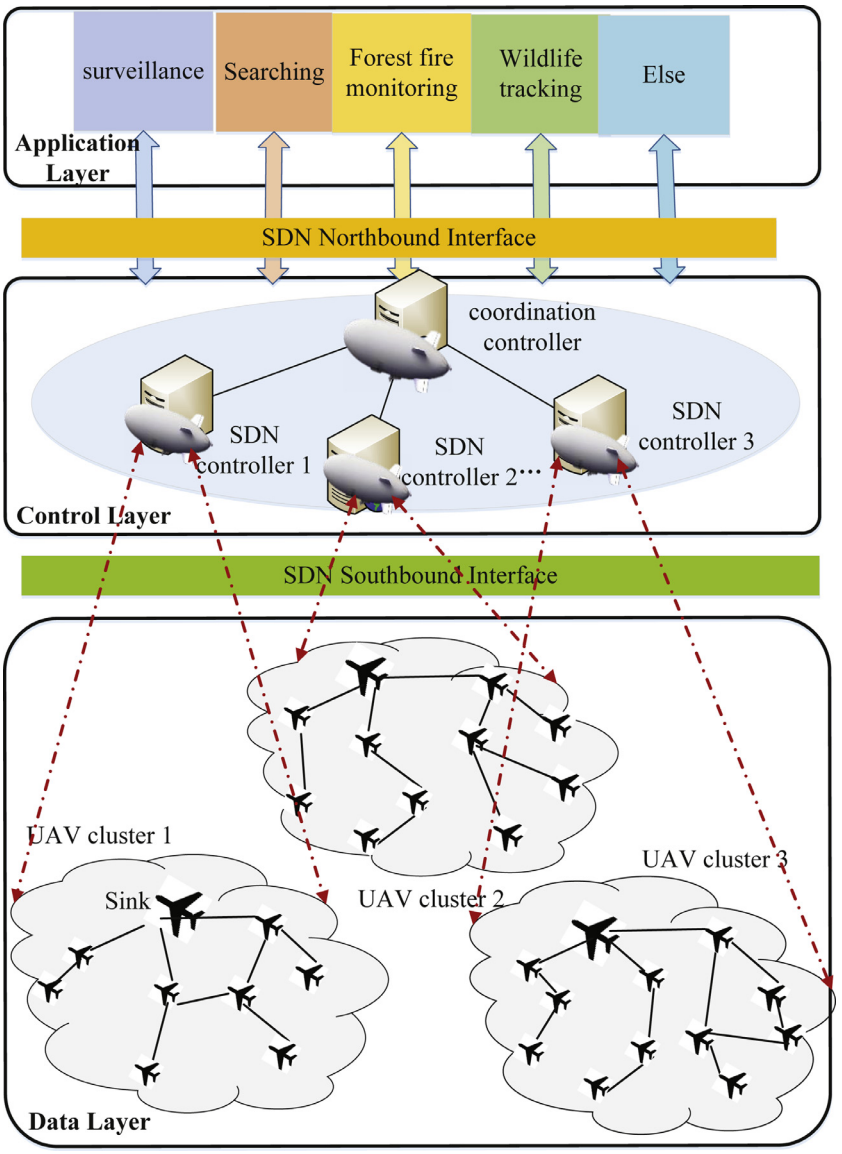


Fig. 2. Illustration of SDN-based FASNET architecture.

The control layer is the core part of the network architecture, which is isolated from the data layer by a Southbound Interface (SBI). Shielding the details of physical devices, the NBI makes the controllers logical entities, which are in charge of acquiring the whole abstract UAV cluster network views, implementing the unified scheduling of network resources and also guiding data processing and forwarding. Each upper stationary airship acts as a single-domain controller. And also, there is a collaborative airship controller which aims to realize the interactions between the single-domain controllers. The special distributed controller structure is of a great significance for network extension.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

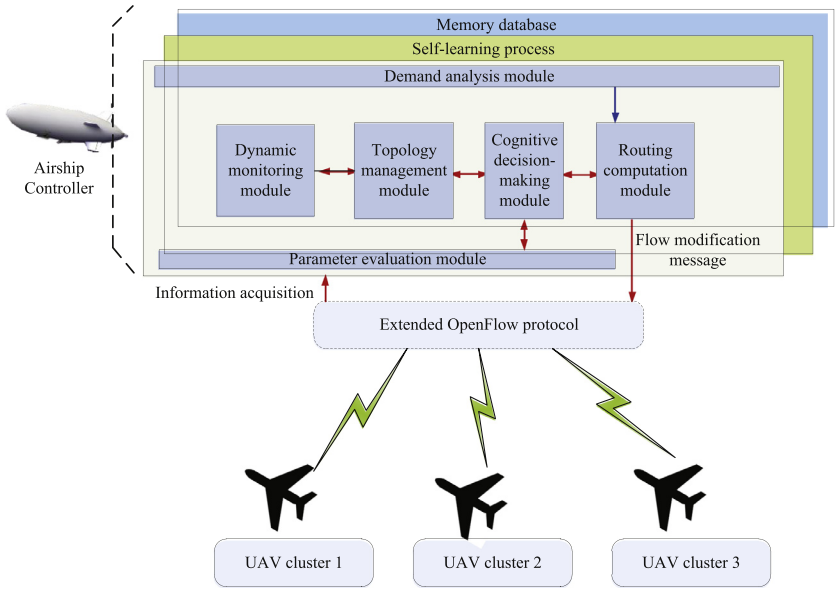


Fig. 3. Illustration of controller function extension.

On the application layer, network functions are written in a software module manner. Once multiple function modules are written into the application layer, the FASNET could perform different tasks at the same time [41]. When a certain network function cannot meet the requirements of the current situation, we just need to upgrade or add the corresponding modular, instead of replacing the entire hardware devices. The way of packing instructions into application modules facilitates the unified network management and network resources scheduling. There is a Northbound Interface (NBI) which isolates the application layer from the control layer, shielding the complexity of network application services. Therefore, when programming an application, the only thing needed to consider is its function rather than how it performs, which is therefore contribute to faster application upgrades.

3.3. Controller function extension

We expand the existing controller in the FASNET to include various modules so that it can realize the unified network management and improve the utilization efficiency of network resources. As shown in Fig. 3, the extended airship controller mainly includes the demand analysis module, dynamic monitoring module, topology management module, routing computation module, and cognitive decision-making module. A brief description is as follows:

The dynamic monitoring module is responsible for online surveillance of network nodes and link information, including information collection for link reliability calculation. Once there is a node or link failure, it should be able to locate and record the fault point quickly and precisely. Based on the monitoring data, the topology management module maintains and updates the topology of the FASNET, providing a global network view for all the other modules. Meanwhile, the demand analysis module makes an analysis on the packets received by controllers, including the traffic flow rate, QoS requirements and other parameters. According

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

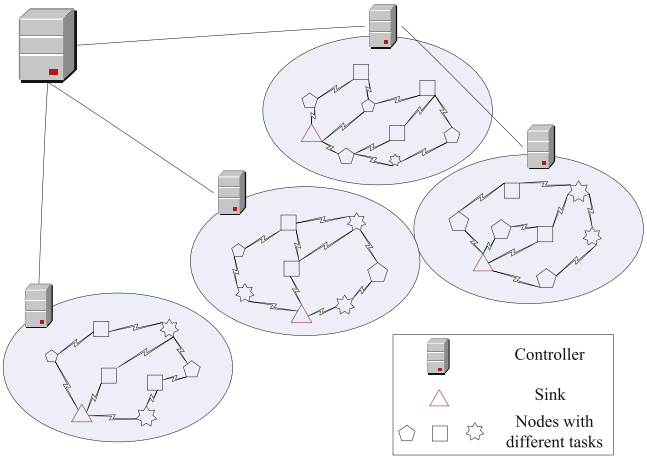


Fig. 4. Illustration of FASNET model.

to the analysis, the routing computation module calls a suitable routing algorithm for network traffic. In addition, to effectively alleviate the heavy workload, a cognitive decision-making module is introduced so that the controller can make decisions in advance based on the historical experience in a memory library. The memory library is constructed to store the historical events such as a packet's forwarding or a full path. Based on numerous similar events in the memory library, the controller makes some prediction before the network deterioration.

In summary, the working mechanism of this SDN-based network architecture is as follows: once there are new nodes deployed in the FASNET, the HELLO and ECHO packets will be transmitted between each controller and its cluster nodes to guarantee the links between UAVs and their controllers, based on the handshake progress in the OpenFlow protocol. Then, the controllers will send the request messages for network topology to nodes in their cluster. The node capacity and link information can be analyzed based on the nodes' reply messages. According to the service request (including source/destination node, traffic flow rate, service parameters, etc.) from the source node sending to the controller, the routing computation module in the controller calls a suitable algorithm to decide an optimal path, center frequency, resource granular and modulation format.

4. Network model and problem description

4.1. Preliminaries

Based on the above SDN-based architecture, we consider a hierarchical FASNET model as illustrated in Fig. 4. Under the management of the cluster controllers, the software-defined UAV sensor nodes equipped with multiple functional sensors are able to perform multitask simultaneously.

In each cluster, all sensed data is transmitted to its sink node along the paths computed by the routing calculation module. There are multiple types of services with different QoS requirements. Unfortunately, the optimal routing for multi-QoS services is not the superposition of routing for those individual service because such simultaneous transmissions may

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

Table 1
Notation list.

Notations	Meanings
$G(V, E)$	FASNET topology, where V is node set and E is link set.
$e(i, j)$	Unidirectional link from node i to j, which belongs to E.
L_t	Total weighted delay of all traffic flows.
R_t	Total weighted reliability of all traffic flows.
s	A source node with traffic demands.
S	Set of nodes with traffic demands.
ϵ	Traffic type index, delay-sensitive or reliability-requisite.
f_s^ϵ	A traffic flow with type ϵ from source node s.
$\omega_d^{s\epsilon}$	Delay-sensitivity level of traffic flow f_s^ϵ.
$\omega_r^{s\epsilon}$	Important level of traffic flow f_s^ϵ.
$path(s, \epsilon)$	Path for traffic flow f_s^ϵ.
L_{ij}	Delay of link $e(i, j)$.
R_{ij}	Transmission reliability of link $e(i, j)$.
f_{ij}	Aggregated traffic in link $e(i, j)$.
C_{ij}	Link capacity of $e(i, j)$.
B_{ij}^t	Bandwidth threshold of $e(i, j)$.
d_{ij}	Transmission delay along link $e(i, j)$.
MAT_{ij}	Possible available duration of link $e(i, j)$.
gr_i	Global trust value of node i.
D_i	Overall traffic request of node i.
M_i^t	Number of transmitters node i has.
M_i^r	Number of receivers node i has.
C_{ij}	Capacity of link $e(i, j)$.
De^ϵ	Maximum end-to-end delay limit for traffic flows of type ϵ.
Re^ϵ	Minimum path reliability limit for traffic flows of type ϵ.

produce some interaction effects and cause overall network performance degradation. Without loss of generality, we just consider one UAV cluster in our network model for simplification, while inter-cluster routing is beyond the scope of this article. The following assumptions are made: (1) The UAV nodes are aware of their positions and speeds through internal global positioning system (GPS). (2) The controller can realize interactions with all UAVs in its cluster and obtain their current states (e.g. ID, position, speed etc) via HELLO and ECHO messages. (3) A certain number of directed antennas are equipped on each UAV in order to reach a long distance for data transmission.

4.2. Problem formulation

We model one FASNET cluster as a graph $G(V, E)$, where V is the set of UAV nodes and E is the set of links in the cluster. $e(i, j) \in E$ is a link between nodes i and j (a notation list is shown in Table 1). There are $(|V| - 1)$ ordinary nodes and one sink node. We run a data-driven optimization framework to update traffic routing paths on the graph periodically. Given delay sensitivities and importance levels of various flows, our objective is to find a routing spanning tree which reduces the average delay for delay-sensitive applications and also improves the data integrity for reliability-requisite applications as much as possible. The objective function is defined as follows:

$$\min (L_t - k \cdot \ln R_t), \tag{1}$$

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

where L_t is the total weighted delay of all traffic flows and R_t is the total weighted reliability of all traffic flows. k is a correction factor. We can see that minimizing the cost function is equivalent to reducing the aggregate delay and increasing the integrated forwarding reliability as much as possible simultaneously.

Furthermore the objective function can be written as follows:

$$\min\left(\sum_{s\in S}\sum_{\epsilon\in\Gamma}\omega_d^{s\epsilon}\sum_{e(i,j)\in E}L_{ij}\cdot x_{ij}^{s\epsilon}-k\cdot\ln\sum_{s\in S}\sum_{\epsilon\in\Gamma}\omega_r^{s\epsilon}\prod_{e(i,j)\in E}R_{ij}\cdot x_{ij}^{s\epsilon}\right), \tag{2}$$

where obviously the first summation item corresponds to L_t and the second one does to R_t. We let f_s^{ϵ} be a traffic flow with type ϵ from source node s. S is the set of nodes with traffic requests. Here $\epsilon\in\Gamma=\{delay\text{-}sensitive, reliability\text{-}requisite\}$. Correspondingly, $\omega_d^{s\epsilon}$ and $\omega_r^{s\epsilon}$ are weights representing the delay sensitivity and importance level of traffic flow f_s^{ϵ} respectively. The value of $x_{ij}^{s\epsilon}$ is a binary variable as follows:

$$x_{ij}^{s\epsilon}=\begin{cases}1, & if\ e(i,j)\in path(s,\epsilon)\\ 0, & otherwise\end{cases}, \tag{3}$$

where $path(s, \epsilon)$ represents the path for traffic flow f_s^{ϵ}.

L_{ij} is the expected delay of packets in the link $e(i, j)$ including queuing delay and transmission delay as follows:

$$L_{ij}=\xi_1\max\left(0,1-\frac{C_{ij}-f_{ij}}{B_{ij}^t}\right)+\xi_2 d_{ij}, \tag{4}$$

where f_{ij} is the aggregated traffic in the link $e(i, j)$. When the residual bandwidth is lower than a threshold B_{ij}^t, queuing delay is induced. d_{ij} is the transmission delay along $e(i, j)$, which relates to the distance between nodes i and j.

R_{ij} is the transmission reliability of $e(i, j)$, which is related to the link availability of $e(i, j)$ and also forwarding ability of node i. It is expressed as follows:

$$R_{ij}=\begin{cases}gr_i, & if\ MAT_{ij}\geq L_{ij}\\ 0, & otherwise\end{cases}, \tag{5}$$

where MAT_{ij} is the possible duration of link $e(i, j)$, reflecting the tendency of link availability. gr_i is the global trust value of node i, reflecting the node's forwarding ability.

The UAVs' position and also speed can be obtained in real time through GPS, according to which, the current or even possible future link availability can be computed. Besides, we utilize the global trust value to represent each node's forwarding ability. Prediction methods for link availability and node's global trust value will be introduced in the next section.

For one node with a traffic request, the traffic flows out of it must equal to those into it, thus we have the following flow conservation constraint:

$$\sum_{v:e(i,v)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma}f_s^{\epsilon}x_{iv}^{s\epsilon}-\sum_{u:e(u,i)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma}f_s^{\epsilon}x_{ui}^{s\epsilon}=D_i,\ i\in S \tag{6}$$

For the sink node acting as a gateway, we have

$$\sum_{v:e(g,v)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma}f_s^{\epsilon}x_{gv}^{s\epsilon}-\sum_{u:e(u,g)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma}f_s^{\epsilon}x_{ug}^{s\epsilon}=-\sum_{i\in S}D_i \tag{7}$$

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

While for the other nodes, we have

$$\sum_{v:e(g,v)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma} f_s^{\epsilon} x_{iv}^{s\epsilon} - \sum_{u:e(u,g)\in E}\sum_{s\in S}\sum_{\epsilon\in\Gamma} f_s^{\epsilon} x_{ui}^{s\epsilon} = 0 \tag{8}$$

We assume single-path routing is adopted in this paper, hence each ordinary node must have one out-going link for one traffic flow:

$$\sum_{v\in V} x_{iv}^{s\epsilon} = 1, i \in V, s \in S, \epsilon \in \Gamma \tag{9}$$

In order to make sure the number of out-going and in-coming links is no more than the number of transmitters and receivers, respectively:

$$\sum_{v\in V} x_{iv} \le M_i^t, \sum_{u\in V} x_{ui} \le M_i^r, i \in V, \tag{10}$$

where $x_{iv} = 1$ if f_{iv} is non-zero, otherwise $x_{iv} = 0$.

The capacity of each link is known in advance and the amount of the aggregated traffic carried by a link does not exceed its capacity:

$$\sum_{s\in S}\sum_{\epsilon\in\Gamma} f_s^{\epsilon} x_{ij}^{s\epsilon} \le C_{ij}, e(i,j) \in E \tag{11}$$

For each traffic request, there are corresponding maximum end-to-end delay and minimum reliability, respectively:

$$\sum_{e(i,j)\in path(s,\epsilon)} L_{ij} \cdot x_{ij}^{s\epsilon} \le De^{\epsilon}, s \in S, \epsilon \in \Gamma \tag{12}$$

$$\prod_{e(i,j)\in path(s,\epsilon)} R_{ij} \cdot x_{ij}^{s\epsilon} \ge Re^{\epsilon}, s \in S, \epsilon \in \Gamma \tag{13}$$

$$\xi_1, \xi_2 \ge 0 \tag{14}$$

In the Multiple Objective Shortest Path (MOSP) problem, the optimal solution for one objective is probably not optimal for the other objectives. There is only a satisfactory solution in the MOSP problem, which is a Pareto optimal solution set [42]. It is NP-hard with great solving difficulty. In the following sections, we will propose a heuristic algorithm to solve it.

5. Transmission reliability prediction model

In order to quantitatively evaluate the quality of links, the routing metric of L_t and R_t need to be obtained. Since for the former one, it could be computed easily, this section presents a novel prediction model for transmission reliability R_t. Taking advantage of the cognitive decision-making module in the controller, our proposed prediction process of transmission reliability including link availability and also nodes forwarding ability would reflect the link and node state in advance, based on the historical information.

Fig. 5 illustrates the framework of our transmission reliability prediction model. Based on the topology information, the prediction process is performed by each cluster SDN controller through the Packet-In and Packet-Out messages. Specifically, based on the acquired node position and velocity by GPS, the link available duration under hypothetical constant velocities and directions could be computed. Then consider the changes of velocities and directions,

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

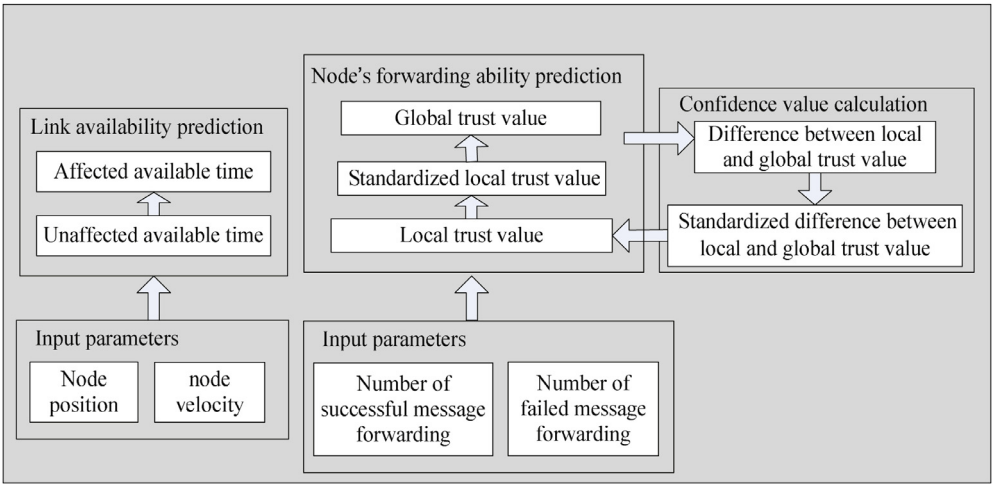


Fig. 5. Framework of transmission reliability prediction model.

affected available time could further be obtained. Besides, for the node's forwarding ability prediction module, the statistics of successful and failed node forwarding among the current neighboring node set is taken as input of local trust value computation. Further, the global trust value could be estimated by iterations. In order to improve the prediction accuracy, an incentive module that is node confidence value calculation is added, whose results feed back to the local trust value calculation module. In the following, we will introduce the prediction methods of link availability and node's forwarding ability.

5.1. Link availability prediction

Assuming that the maximum transmission ranges of nodes are known in advance, and the respective locations of nodes are known by GPS, then the unaffected link available time could be computed if no change in velocities and directions happens [43].

Here we have two mobile nodes n_i and n_j, and both of them have equal maximum transmission range d_{max}. Assuming that at time t_0, (x_i, y_i, z_i) and (x_j, y_j, z_j) are their positions, also (v_{xi}, v_{yi}, v_{zi}) and (v_{xj}, v_{yj}, v_{zj}) are their velocity vector. Then, we can obtain the distance between them is:

$$d_{ij}(t_0) = \left[\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2 + \left(z_j - z_j\right)^2\right]^{\frac{1}{2}} \tag{15}$$

We assume that both nodes have no change in velocity. After a certain time period t, the distance between these two nodes is as follows:

$$\begin{aligned} d_{ij}(t_0 + t) = \Big\{&\left[\left(x_j + v_{xj}t\right) - \left(x_i + v_{xi}t\right)\right]^2 \\ &+\left[\left(y_j + v_{yj}t\right) - \left(y_i + v_{yi}t\right)\right]^2 \\ &+\left[\left(z_j + +v_{zi}t\right) - \left(z_j + v_{zi}t\right)\right]^2\Big\}^{\frac{1}{2}} \end{aligned} \tag{16}$$

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

When $d_{ij}(t_0 + t) = d_{max}$, the two nodes may begin to move beyond the communication range of each other. Hence, the expected available time T_{ij} of link $e(i, j)$ can be obtained by solving the quadratic equation for t above.

In [44], the authors consider possible changes in velocities occurring during the period T_{ij} and try to estimate the probability $P(T_{ij})$ that $e(i, j)$ may really last for T_{ij}, The basic assumptions are: (1) UAV mobility is uncorrelated and (2) epochs (an epoch is a random time interval during which a node moves in a constant direction at a constant speed) are exponentially distributed with mean as λ^{-1}:

$$F(x) = P(epoch \leq x) = 1 - e^{-\lambda x} \tag{17}$$

Further, $P(T_{ij})$ is shown as follows:

$$P(T_{ij}) \approx e^{-2\lambda T_{ij}}\left(\frac{\lambda T_{ij}}{2} - \frac{1}{\lambda T_{ij}}\right) + \frac{1}{2\lambda T_{ij}}\Big)\Big) \tag{18}$$

Considering changes of node speeds and directions, we take the following equation as the lifetime of link $e(i, j)$:

$$MAT_{ij} = T_{ij} \cdot P\left(T_{ij}\right) \tag{19}$$

5.2. Node's forwarding ability prediction

We define M_{ij} and F_{ij} as successful and failed data transmissions respectively between nodes i and j, which are recorded in the local history database, based on which, node i can determine its local trust lr_{ij} to the node j. We determine lr_{ij} as follows:

$$lr_{ij} = \left(1 - e^{-\alpha M_{ij}}\right) \cdot e^{-\beta F_{ij}}, \tag{20}$$

where α and β are regulatory factors and $0 < \alpha < \beta < 1$. The increasing rate of trust caused by good behaviors is lower than the increasing rate of trust triggered by malicious behaviors, which conforms to the trust constructing rule.

Then the local trust value is normalized by Eq. (21):

$$\hat{lr}_{ij} = \frac{lr_{ij}}{\sum_{j=1}^{n} lr_{ij}}, \tag{21}$$

where $n = |V| - 1$ is the number of regular nodes in a network cluster.

With the above normalization, $0 \leq \hat{lr}_{ij} \leq 1$, and $\sum_{j=1}^{n} \hat{lr}_{ij} = 1$, effectively ensuring the convergence of the calculation.

The global trust to a certain node is made by the whole network. It is the sum of the product of all nodes' trusts made to the object node and their own global trust as follows:

$$gr_i = \sum_{j=1}^{n}\left(\hat{lr}_{ji} \cdot gr_j\right) \tag{22}$$

In order to estimate the local trust more accurately, we add an incentive item—an evaluation credibility C_i which represents the credible degree of the evaluations node i makes to other

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

nodes, is as follows:

$$C_i = \begin{cases} C_i + \frac{1-C_i}{2} \cdot (1 - 2AD_i), & if\ AD_i < 0.5 \\ C_i - \frac{C_i}{2} \cdot \left(1 - \frac{1}{2AD_i}\right), & if\ AD_i \geq 0.5 \end{cases} \tag{23}$$

where

$$AD_i = \frac{1}{n} \sum_{j=1}^{n} \frac{|gr_j - lr_{ij}|}{gr_j} \tag{24}$$

AD_i represents the mean absolute difference of trust values node i makes to all other nodes and their global value. An honest node has a AD_i smaller than 0.5 and there will be a small increase in C_i. AD_i's being larger than 0.5 indicates that node i is a dishonest node and hence there is a great decrease in C_i.

We modify Eq. (20) as follows to make the incentive C_i feed back to the local trust value:

$$lr_{ij} = e^{C_j}\left(1 - e^{-\alpha M_{ij}}\right) \cdot e^{-\beta F_{ij}} \tag{25}$$

The whole iterative calculation process of the global trust value gr_{ij} is shown in Algorithm 1.

So far, we have obtained R_{ij} in Eq. (5) by computing MAT_{ij} and gr_{ij} respectively.

6. Ant-colony-based routing algorithm

Due to the NP-hard nature of our problem characterized in Section 4, we propose a heuristic algorithm base on ant-colony to solve it.

Ant colony algorithm simulates ants' behavior of seeking food with positive feedback [45]. Aiming to find the shortest path from their caves to the food source, ants release pheromone on the path they pass by, which evaporates over time. The concentration of pheromone is inversely proportional to the length of the path. The shorter path with more pheromone attracts more ants going along it. After a period, the shortest path will always be selected. Ant colony algorithm seldom relies on initial routes, and no manual intervention occurs in the searching process. Therefore, it is easy for us to solve the MOSP problem using ant colony algorithm.

In this paper, we propose a multi-layer network model for traffic differentiated routing. As shown in Fig. 6, the physical topology $G(V, E)$ is divided into two layers: delay-sensitive and reliability-requisite. Given $|S|$ delay-sensitive traffic demand $\{f_s^{\epsilon}, s \in V, \epsilon = delay - sensitive\}$ and $|S'|$ reliability-requisite traffic demand $\{f_s^{\epsilon}, s \in V, \epsilon = reliability - requisite\}$. Two spanning trees which are all paths from source nodes to the gateway node are generated in each layer respectively. The two spanning trees are merged into a single one T_m in the hybrid graph. The ant colony algorithm helps to find a merged spanning tree that closely satisfies the objective function.

The pseudo code of the algorithm is shown in Algorithm 2. We imagine τ_{ij} is the pheromone intensity on link $e(i, j)$ which evaporates with rate ρ. In each iteration, there are ants starting from the source nodes to find a path to the gateway in each layer. In this process, each ant keeps a node list called *Visited* to avoid visiting a node twice and also a node list called *Route* to record the nodes it passed by. Once the optional neighboring node list is determined, an ant at node i chooses j as its next-hop node with a probability as follows:

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

Algorithm 1 Node's forwarding ability algorithm.

Require: Initial values of M_{ij}, F_{ij}, C_i, gr_i;
Ensure: global trust value gr_i;
for all iteration $k = 0$ **to** T **do**
  if nodes i and j communicate successfully **then**
    $M_{ij} \leftarrow M_{ij} + 1$;
  else
    $F_{ij} \leftarrow F_{ij} + 1$;
  end if
  for i=1,2,...,$|V| - 1$ **do**
    for j=1,2,...,$|V| - 1$ **do**
      $lr_{ij} \leftarrow e^{C_j}(1 - e^{-\alpha M_{ij}}) \cdot e^{-\beta F_{ij}} + \delta$;
      $lr_i \leftarrow lr_i + lr_{ij}$;
    end for
  end for
  for i=1,2,...,$|V| - 1$ **do**
    for j=1,2,...,$|V| - 1$ **do**
      $\hat{lr}_{ij} \leftarrow lr_{ij}/lr_i$;
      $gr_j \leftarrow gr_j + lr_{ij} \cdot gr_i$;
      $AD_{ij} \leftarrow |gr_j - lr_{ij}|/gr_j$;
      $AD_i \leftarrow TD_i + TD_{ij}$;
    end for
    $AD_i \leftarrow AD_i/(|V| - 1)$;
  end for
  if $AD_i < 0.5$ **then**
    $C_i \leftarrow C_i + (1 - C_i)/2 \cdot (1 - 2AD_i)$;
  else
    $C_i \leftarrow C_i - C_i/2 \cdot (1 - 1/2AD_i)$;
  end if
  Return gr_i
end for

$$P_{ij} = \frac{\tau_{ij} \cdot \eta_{ij}^{\varphi}}{\sum_{j \in N(i)} \tau_{ij} \cdot \eta_{ij}^{\varphi}}, \tag{26}$$

where $\eta_{ij} = 1/d_{ij}$.

Two spanning trees are generated when all ants stop at the gateway, denoted as T_1 and T_2. Merge the spanning trees into a single one T_m in the hybrid graph and the cost of T_m could be easily obtained by Eq. (2). A best tree with minimum objective function value is selected after each M iterations. And then the pheromone on the best tree's links will be enhanced as follows:

$$\tau_{ij} = \begin{cases} (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}, & if\ e(i, j) \in T_{opt} \\ (1 - \rho)\tau_{ij}, & otherwise \end{cases}, \tag{27}$$

where $\Delta\tau_{ij} = 1/cost(T_{opt})$.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

Algorithm 2 Ant-colony-based routing algorithm.

Require: $G(V, E)$, L_{ij}, R_{ij}, MAT_{ij}.
Ensure: T_{opt}

$T_{min} = \varnothing$, $T_{opt} = \varnothing$, $T_{set} = \varnothing$;
while convergence condition not met **do**
  if $cost(T_{min}) < cost(T_{opt})$ **then**
    $T_{opt} \leftarrow T_{min}$;
  end if
  $T_1 \leftarrow AntFind(S(D))$
  $T_2 \leftarrow AntFind(S(R))$
  $T_m \leftarrow T_1 + T_2$
  $T_{set} \leftarrow T_{set} \bigcup \{T_m\}$;
  $T_{min} \leftarrow argmin_{T \in T_{set}} cost(T)$;
  if i mod $M = 0$ **then**
    for each $e(i, j) \in E$ **do**
      $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho \Delta \tau_{ij}$;
    end for
  end if
end while
Return T_{opt}
Function AntFind(S)
$T = \varnothing$, $Route = \varnothing$;
for all $s \in S$ **do**
  $Visited \leftarrow s$;
  $i \leftarrow s$;
  while $i \neq g$ and $i \notin Route$ **do**
    $Route \leftarrow Route \bigcup \{i\}$;
    for all $j \in N(i)$ **do**
      if $MAT_{ij} \geq L_{ij}$ and $j \notin Visited$ **then**
        $N'(i) \leftarrow N(i) \cup j$;
      end if
    end for
    for all $j \in N'(i)$ **do**
      choose l as the next hop by Eq. (22);
      $T \leftarrow T \bigcup \{e(i, l)\}$;
      $Visited \leftarrow Visited \bigcup \{l\}$;
      $i \leftarrow l$;
      break;
    end for
  end while
end for
Return T
End Function

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

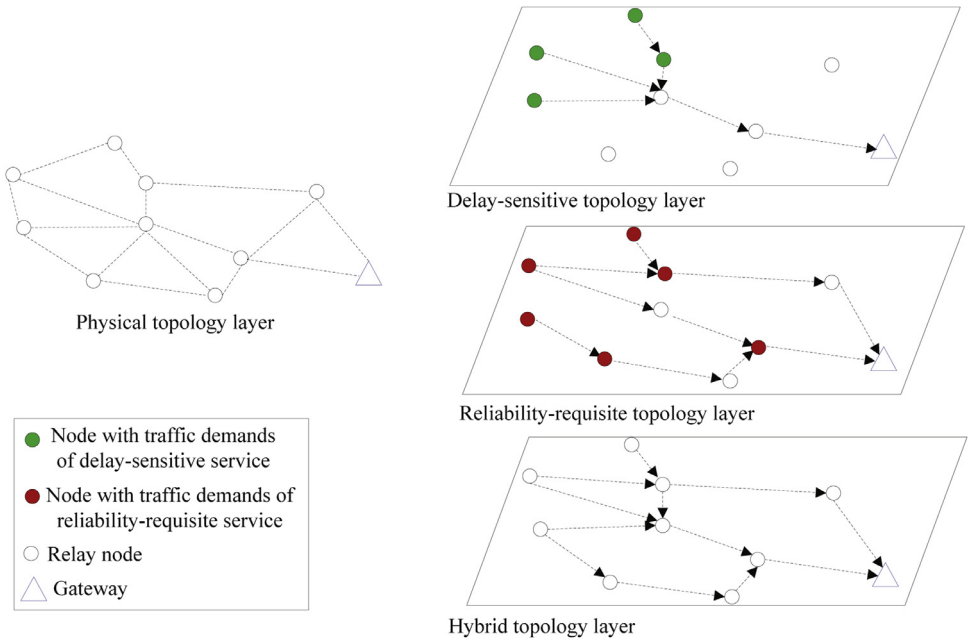


Fig. 6. Multi-layer network model for traffic differentiated routing.

After multiple iterations, when the system converges to a stable state, we can obtain the near-optimal tree and the MOSP problem is solved. Since the algorithm is executed in the controller with powerful computations, the complexity is not a serious issue here.

The time complexity of ant colony algorithm is $T(n) = O(N_c n^2 m)$, where N_c is the number of iterations, n is the number of UAVs and m is the number of ants. Since the UAV networks is clustered, the number of UAVs in each cluster is limited and the time complexity is acceptable. The algorithm is executed in the airship SDN controller, hence there is no constraints on the space complexity.

7. Performance evaluation

In this section, we evaluate the effectiveness of our TDR algorithm in a cluster network. We first describe the simulation settings, and then compare TDR with the MintRoute algorithm [46], which is the TinyOS's standard routing algorithm, in terms of end-to-end delay, packet dropping ratio and network throughput. In MintRoute, each node sends probe packets to its neighboring nodes. Based on these probe packets, the neighboring nodes update the link state information and the next-hop node will be selected according to link states. For comparison, here we make each node select next-hop based on the prediction result of link availability in this paper.

7.1. Simulation setup

In our simulation, the parameters of the network environment are listed in Table. 2. It is worth noting that in the exponential correlated random model, a node selects a constant

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

Table 2
Parameter setting of the network environment.

Parameter	Value
Number of nodes (UAVs)	40
Network range	2 km*2 km
Node mobility model	Exponential correlated random model
Mean epoch of constant movement λ^{-1}	10 s
Node movement speed	$0-20$ m/s
Percentage of malicious nodes	20%
Channel capacity	20 Mbps
Maximum transmission range	600 m
Traffic type	CBR
Number of traffic flow of APP1	10
Number of traffic flow of APP2	10
Rate of traffic flow	1.5 Mbps
Weights of sensibility and reliability for APP1	(1, 5)
Weights of sensibility and reliability for APP2	(5, 1)

Table 3
Parameter setting of the algorithms.

Parameter	α	β	φ	ρ	τ_{ij}^0	M
Value	0.2	0.3	0.2	0.1	0.1	100

direction from the given space and a speed from 0 to 20 m/s for its next movement. The epochs of the above movement are exponentially distributed with mean λ^{-1}. We make APP1 as the delay-sensibility applications and APP2 as the reliability-requisite applications. We assign 1 and 5 as weights of sensibility and reliability for all traffic of APP1, and they are reversed for APP2. In fact, our proposed routing algorithm is suitable for other types of services by setting different weights of sensibility and reliability, and traffic flows of more than two types of applications are also allowed.

The parameter setting of the algorithms is listed in Table 3. α and β are regulatory factors in Eq. (20). We imagine that a node has 10 transactions with another node during a period, and the positive evaluation is set to be 0.85 if all of them are successful, that is $1-e^{(-\alpha\times 10)}=0.85$. We obtain that $\alpha=0.2$. The negative evaluation is set to be 0.05, that is $1-e^{(\beta\times 10)}=0.05$ and thus $\beta=0.3$.

It is noteworthy that the parameter k in Eq. (1) determines the proportions of delay and packet dropping ratio in the total tree cost. To obtain a global optimum spanning tree, we need to balance the weights of delay and packet dropping ratio by k. In the cluster network, take the gateway node as the root and the corresponding minimum spanning trees are calculated under different values of k. Then the relation between the delay and packet dropping ratio is recorded in Fig. 7. We can see that when $k=0.7$, the values of delay and packet dropping ratio are balanced and hence we take it as our simulation parameter.

At first, our TDR program runs for multiple iterations to collect enough historical information until it converges to a stable state [47].

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

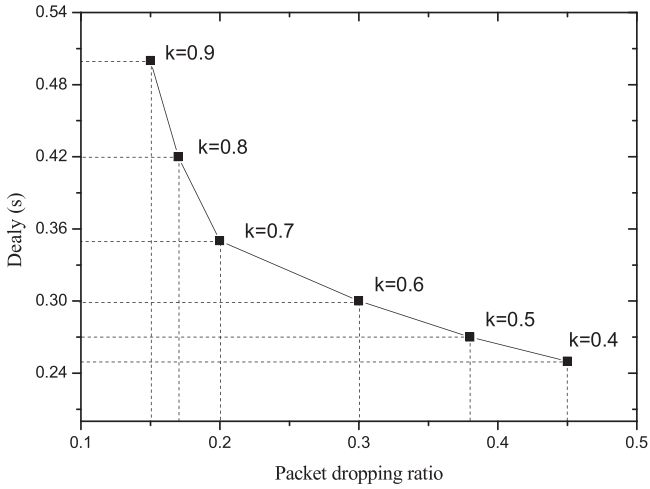
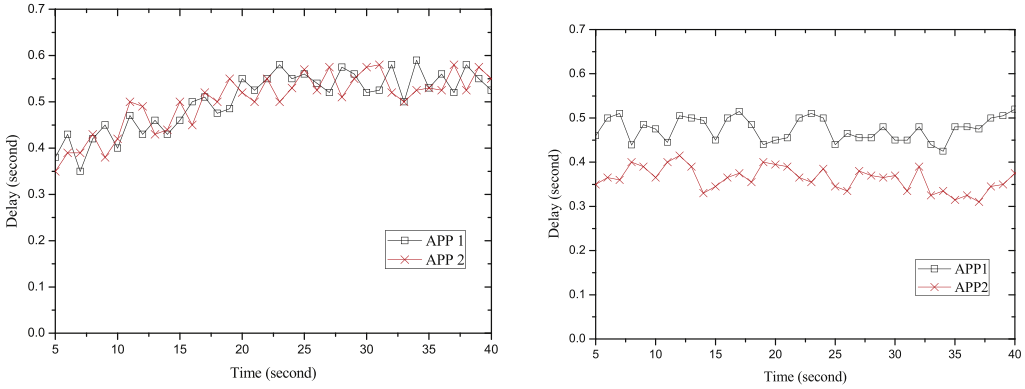


Fig. 7. Relationship between delay and packet dropping ratio value under different k.



(a) MintRoute (b) TDR

Fig. 8. Average end-to-end delay of each application over time in MintRoute and TDR algorithms.

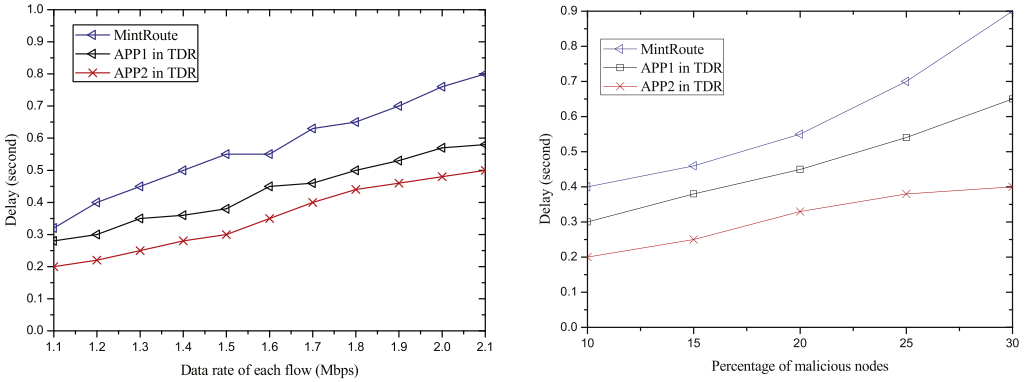
7.2. Simulation results

7.2.1. Average end-to-end delay

The end-to-end delay is the sum of transmission time and queuing time along the path from a source node to the gateway.

By tracking the average end-to-end delay of the two types of traffic, we record the performance of MintRoute and TDR during the periods of from 5 to 40 s in Fig. 8 (a) and (b), respectively. We can see that in MintRoute, there is almost no delay difference between the two kinds of traffic. All flows are routed along the most reliable paths from their sources to the gateway. In fact the most reliable may not always be reliable since MintRoute give no considerations to the nodes forwarding ability and also link congestions. The constant data streams getting into a certain group of links result in more serious link congestion and longer

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012



(a) Delay versus data rate of each flow (b) Delay versus percentage of malicious nodes

Fig. 9. Comparisons of average end-to-end delay of each application in MintRoute and TDR algorithms.

packet queues, leading to the gradual increase of end-to-end delay. While in Fig. 8 (b), the average delays of APP1 and APP2 in TDR show great difference. The reason is obvious that different paths are selected for the two types of traffic flows according to their allocated weights. The packets of APP2 are delay-sensitive, which means a larger weight of sensibility is assigned in the objective function. When the network is with a light load, they will choose the shorter path with lower transmission delay, otherwise when an overload occurs, proper longer path with smaller queue length will be selected. For APP1, which is reliability-requisite, the average delay is higher than that of APP2. However, it is still superior to that in MintRoute since the various paths make the traffic more disperse and the longer paths may have smaller queuing delay.

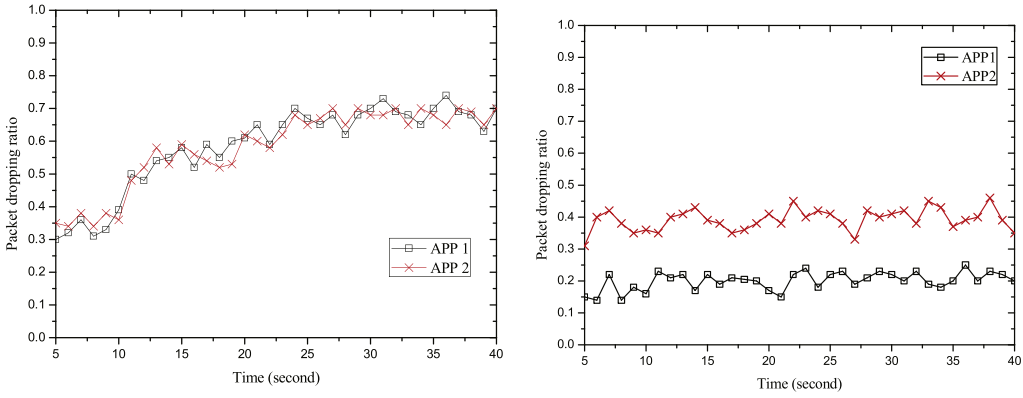
Fig. 9 shows the average end-to-end delay of traffic of each application in the MintRoute and TDR algorithm. The delay changes along with the traffic load per flow. For the same reason as stated above, traffic of type APP2 in TDR performs best, traffic of type APP1 comes the second and traffic in MintRoute is the worst. With the increase of the traffic load per flow, the network congestion becomes increasingly serious and results in a longer delay. All traffic being carried by part of links lead to larger end-to-end delay in MintRoute. The changing trends with the percentage of malicious nodes are demonstrated in Fig. 9(b). In MintRoute, the average delay is significantly affected by the percentage of malicious nodes since it provides no means to recognize the malicious nodes, leading to packets losses and retransmissions. While in TDR, especially for traffic of type APP1, the negative effect is weakened due to our trust model. The packets with reliability requirement will be routed in a path avoiding nodes with low degrees of trust.

7.2.2. Packet dropping ratio

Packet dropping ratio is the ratio of the number of successfully delivered packets over the total number of packets in the network. It is relative to delay since some packets become expired under long delay.

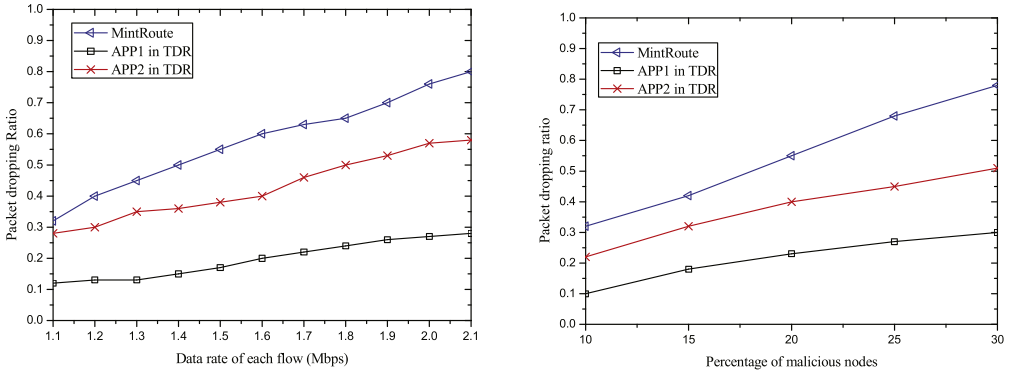
The packet dropping ratio of traffic of each application over time in MintRoute and TDR algorithms is shown in Fig. 10. We can see that in MintRoute, there is no packet-dropping-ratio difference between the two types of applications. The reason lies in the same path selection

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012



(a) MintRoute (b) TDR

Fig. 10. Packet dropping ratio of each application over time in MintRoute and TDR algorithms.



(a) Packet dropping ratio versus data rate of each flow

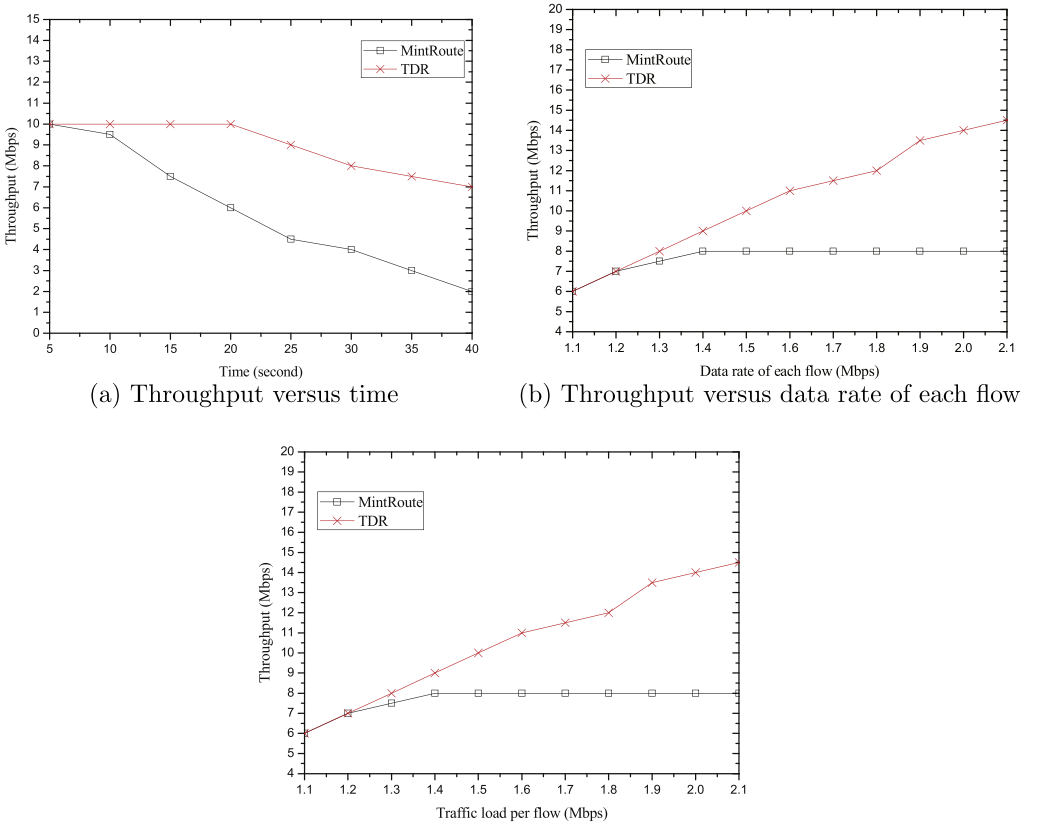
(b) Packet dropping ratio versus percentage of malicious nodes

Fig. 11. Comparisons of packet dropping ratio of each application in MintRoute and TDR algorithms.

method as mentioned above. There is an increase in packet dropping ratio along with time. That is because that a buffer overflow will occur once the link traffic load is larger than a threshold. While in TDR, the dropping ratio of traffic of type APP1 is obviously smaller than that of type APP2. That is because the packets of type APP1 have a larger weight of reliability in the objective function. They will choose paths consisting of nodes with higher forwarding reliability and partly avoid malicious nodes. Though the packet dropping ratio of the delay-sensitive traffic of type APP2 is higher than that of APP1, it is still superior to that in MintRoute since the disperse traffic alleviates the network congestion.

Fig. 11 shows the comparative results of packet dropping ratio for diverse type of applications with varying traffic load per flow and percentage of malicious nodes, in MintRoute and TDR algorithms. From Fig. 11(a) we can observe that the traffic of type APP1 in TDR performs best, that of type APP2 in TDR comes the second and traffic in MintRoute is the worst. With the increase of the traffic load per flow, the network congestion becomes increas-

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012



(a) Throughput versus time

(b) Throughput versus data rate of each flow

(c) Throughput versus percentage of malicious nodes

Fig. 12. Comparisons of packet dropping ratio of each application in MintRoute and TDR algorithms.

ingly serious and results in extensive packet losses. Fig. 11(b) shows that the packet dropping ratio in MintRoute is much higher than that in TDR and also significantly affected by the percentage of malicious nodes. TDR provides a prediction method to evaluate nodes' trust values according to their behaviors. For the packets of type APP1, the nodes with higher trust values will be selected as relay nodes to guarantee the forwarding reliability.

7.2.3. Network throughput

Network throughput is defined as the amount of data transmitted by the network per time slot. Fig. 12 demonstrates the network throughput of MintRoute and TDR varying with time, traffic load per flow and percentage of malicious nodes. TDR shows great advantages in these three aspects. Specifically, For the MintRoute algorithm, the network throughput decreases as time goes by due to the network congestion caused by overloaded links. The network throughput of our TDR algorithm shows a much slower decline over time because the routing paths can be adjusted dynamically to avoid overloaded links. In Fig. 12(b), as the traffic load per flow increases, the network throughput also increases in TDR. While in MintRoute, the traffic load is more easily to reach saturation since it doesn't take the forwarding reliability and queuing delay into consideration. In addition, Fig. 12(c) demonstrates that the increase

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

in the percentage of malicious nodes directly degrades throughput performance. Fortunately, the trust model in TDR alleviates the situation.

8. Conclusion

In this paper, we have designed a hierarchical FASNET architecture with some SDN cluster controllers and one collaborative controller, where the network resources in one cluster are managed by the upper airship. Based on this architecture, we have further proposed the TDR algorithm executed in each cluster. In order to fulfill the specific QoS requirements of delay-sensitive and reliability-requisite services, different weights have been assigned to various flows according to their sensitivity to delay and also levels of importance. The transmission reliability prediction model has been introduced to TDR. Simulation results have shown that TDR not only reduces the average delay for delay-sensitive applications but also improves the data integrity for reliability-requisite applications.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (91438110, 61401082, 61501104, 61501105, 61775033, 61771120) and the Fundamental Research Funds for the Central Universities (N161606001, N150401002, N161608001).

References

[1] I. Bekmezci, O.K. Sahingoz, c. Temel, Flying ad-hoc networks (FANETs): a survey, J. Ad Hoc Netw. 11 (3) (2013) 1254–1270.
[2] W.L. Teacy, J. Nie, S. McClean, J. Parr, Maintaining connectivity in UAV swarm sensing, in: Proceedings of the IEEE GLOBECOM Workshop, 2010, pp. 1771–1776.
[3] L. Gupta, R. Jain, G. Vaszkun, Survey of important issues in UAV communication networks, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1123–1152.
[4] Z. Wang, J. Liao, Q. Cao, H. Qi, Z. Wang, Friendbook: a semantic-based friend recommendation system for social networks, IEEE Trans. Mob. Comput. 14 (3) (2015) 538–551.
[5] Z. Ning, F. Xia, N. Ullah, X. Kong, X. Hu, Vehicular social networks: enabling smart mobility, IEEE Commun. Mag. 55 (5) (2017a) 49–55.
[6] Z. Ning, X. Hu, Z. Chen, M. Zhou, B. Hu, J. Cheng, M. Obaidat, A cooperative quality-aware service access system for social internet of vehicles, IEEE Internet Things J. (2017b), doi:10.1109/JIOT.2017.2764259.
[7] W. Qian, L. Wang, M.Z. Chen, Local consensus of nonlinear multiagent systems with varying delay coupling, IEEE Trans. Syst. Man Cybern. Syst. (2017a), doi:10.1109/TSMC.2017.2684911.
[8] W. Qian, M. Yuan, L. Wang, X. Bu, J. Yang, Stabilization of systems with interval time-varying delay based on delay decomposing approach, ISA Trans. 70 (2017b) 1–6.
[9] J. Hui, M. Devetsikiotis, A unified model for the performance analysis of IEEE 802.11 e EDCA, IEEE Trans. Commun. 53 (9) (2005) 1498–1510.
[10] E. Felemban, C.G. Lee, E. Ekici, MMSPEED: multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, IEEE Trans. Mob. Comput. 5 (6) (2006) 738–754.
[11] D. Djenouri, I. Balasingham, Traffic-differentiation-based modular QoS localized routing for wireless sensor networks, IEEE Trans. Mob. Comput. 10 (6) (2011) 797–809.
[12] T.S. Prakash, K.B. Raja, K.R. Venugopal, S.S. Iyengar, L.M. Patnaik, Traffic-differentiated two-hop routing for QoS in wireless sensor networks, in: Proceedings of the IEEE Conference of Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013, pp. 356–363.
[13] L. Wang, W. Qian, Q.G. Wang, Exponential synchronization in complex networks with a single coupling delay, J. Frankl. Inst. 350 (6) (2013) 1406–1423.
[14] Y. Li, C.S. Chen, Y.Q. Song, Z. Wang, Y. Sun, Enhancing real-time delivery in wireless sensor networks with two-hop information, IEEE Trans. Ind. Inform. 5 (2) (2009) 113–122.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

[15] N. Javaid, A. Javaid, I.A. Khan, K. Djouani, Performance study of ETX based wireless routing metrics, in: Proceedings of the IEEE Conference of Computer, Control and Communication, 2009, pp. 1–7.
[16] Z. Ning, F. Xia, X. Hu, Z. Chen, M. Obaidat, Social-oriented adaptive transmission in opportunistic internet of smartphones, IEEE Trans. Ind. Inform. 13 (2) (2017a) 810–820.
[17] Z. Ning, L. Liu, F. Xia, B. Jedari, I. Lee, W. Zhang, CAIS: a copy adjustable incentive scheme in community-based socially-aware networking, IEEE Trans. Veh. Technol. 66 (4) (2017b) 3406–3419.
[18] Y. Hu, Z. Jin, S. Qi, C. Sun, Estimation fusion for networked systems with multiple asynchronous sensors and stochastic packet dropouts, J. Frankl. Inst. 354 (1) (2017) 145–159.
[19] W. Hou, Z. Ning, L. Guo, Z. Chen, Novel framework of risk-aware virtual network embedding in optical data center networks, IEEE Syst. J. PP (99) (2017) 1–10, doi:10.1109/JSYST.2017.2673828.
[20] A.M. Ahmed, X. Kong, L. Liu, F. Xia, S. Abolfazli, Z. Sanaei, A. Tolba, BoDMaS: bio-inspired selfishness detection and mitigation in data management for ad-hoc social networks, Ad Hoc Netw. 55 (2017) 119–131.
[21] F. Dressler, I. Dietrich, R. German, B. Krüger, A rule-based system for programming self-organized sensor and actor networks, J. Comput. Netw. 53 (10) (2009) 1737–1750.
[22] P. Dely, A. Kassler, N. Bayer, OpenFlow for wireless mesh networks, in: Proceedings of the IEEE Computer Communications and Networks Conference (ICCCN), 2011, pp. 1–6.
[23] T. Luo, H.P. Tan, T.Q. Quek, Sensor OpenFlow: enabling software-defined wireless sensor networks, IEEE Commun. Lett. 16 (11) (2012) 1896–1899.
[24] S. Costanzo, L. Galluccio, G. Morabito, S. Palazzo, Software defined wireless networks: unbridling SDNs, in: Proceedings of the IEEE Software Defined Networking Conference (EWSDN), 2012, pp. 1–6.
[25] D. Zeng, T. Miyazaki, S. Guo, T. Tsukahara, J. Kitamichi, T. Hayashi, Evolution of software-defined sensor networks, in: Proceedings of the Mobile Ad-hoc and Sensor Networks Conference (MSN), 2013, pp. 410–413.
[26] L. Galluccio, S. Milardo, G. Morabito, S. Palazzo, SDN-WISE: design, prototyping and experimentation of a stateful SDN solution for wireless sensor networks, in: Proceedings of the IEEE Computer Communications Conference (INFOCOM), 2015, pp. 513–521.
[27] A.C.G. Anadiotis, G. Morabito, S. Palazzo, An SDN-assisted framework for optimal deployment of MapReduce functions in WSNs, IEEE Trans. Mob. Comput. 15 (9) (2016) 2165–2178.
[28] H. Fotouhi, M. Vahabi, A. Ray, M. Björkman, SDN-TAP: an SDN-based traffic aware protocol for wireless sensor networks, in: Proceedings of the IEEE e-Health Networking, Applications and Services Conference (Healthcom), 2016, pp. 1–6.
[29] Y. Wang, H. Chen, X. Wu, L. Shu, An energy-efficient SDN based sleep scheduling algorithm for WSNs, J. Netw. Comput. Appl. 59 (2016) 39–45.
[30] M. Mukherjee, L. Shu, T. Zhao, K. Li, H. Wang, Low control overhead-based sleep scheduling in software-defined wireless sensor networks, in: Proceedings of the IEEE HPCC/SmartCity/DSS Conference, 2016, pp. 1236–1237.
[31] Y. Zhu, F. Yan, Y. Zhang, L. Shen, SDN-based anchor scheduling scheme for localization in heterogeneous WSNs, IEEE Commun. Lett. 21 (5) (2017) 1127–1130.
[32] E.D. Jones, R.S. Roberts, T.S. Hsia, STOMP: a software architecture for the design and simulation of UAV-based sensor networks, in: Proceedings of the IEEE Robotics and Automation Conference, 2003, pp. 3321–3326.
[33] M. Sara, I. Jawhar, M. Nader, A softwarization architecture for UAVs and WSNs as part of the cloud environment, in: Proceedings of the IEEE Cloud Engineering Workshop Conference (IC2EW), 2016, pp. 13–18.
[34] K. Akkaya, M. Younis, An energy-aware QoS routing protocol for wireless sensor networks, in: Proceedings of the International Conference on Distributed Computing Systems Workshops, 2003, pp. 710–715.
[35] K. Akkaya, M. Younis, Energy and QoS aware routing in wireless sensor networks, J. Clust. Comput. 8 (2–3) (2005) 179–188.
[36] M.A. Hamid, M.M. Alam, C.S. Hong, Design of a QoS-aware routing mechanism for wireless multimedia sensor networks, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM), 2008, pp. 1–6.
[37] M.A. Razzaque, M.M. Alam, M. Mamun-Or-Rashid, C.S. Hong, Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks, IEICE Trans. Commun. 91 (8) (2008) 2589–2601.
[38] L. Wang, H. Shi, Y.X. Sun, Power adaptation for a mobile agent network, Europhys. Lett. 90 (1) (2010) 10001.
[39] J. Zhang, F. Ren, S. Gao, H. Yang, C. Lin, Dynamic routing for data integrity and delay differentiated services in wireless sensor networks, IEEE Trans. Mob. Comput. 14 (2) (2015) 328–343.
[40] N. Sirdeshpande, V. Udupi, Fractional lion optimization for cluster head-based routing protocol in wireless sensor network, J.Frankl. Inst. 354 (11) (2017) 4457–4480.
[41] W. Xiang, N. Wang, Y. Zhou, An energy-efficient routing algorithm for software-defined wireless sensor networks, IEEE Sens. J. 16 (20) (2016) 7393–7400.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012

[42] S. Hu, X. Xu, D. Zhan, A genetic algorithm for the Pareto optimal solution set of multi-objective shortest path problem, J. Harbin Inst. Technol. 12 (6) (2005) 721–726.
[43] W. Su, M. Gerla, IPv6 flow handoff in ad hoc wireless networks using mobility prediction, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM), 1999, pp. 271–275.
[44] S. Jiang, D. He, J. Rao, A prediction-based link availability estimation for routing metrics in MANETs, IEEE/ACM Trans. Netw. (TON) 13 (6) (2005) 1302–1312.
[45] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[46] MintRoute. http://www.tinyos.net/tinyos-1.x/tos/lib/MintRoute.
[47] L. Wang, M.Z. Chen, Q.G. Wang, Bounded synchronization of a heterogeneous complex switched network, Automatica 56 (2015) 19–24.

Please cite this article as: W. Qi et al., A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers, Journal of the Franklin Institute (2017), https://doi.org/10.1016/j.jfranklin.2017.11.012