An SDN Based Framework for Load Balancing and Flight Control in UAV Networks

Zohaib Latif
Department of Computer Science, Hanyang University, Korea

Choonhwa Lee
Department of Computer Science, Hanyang University, Korea

Kashif Sharif
School of Computer Science and Technology, Beijing Institute of Technology, China

Fan Li
School of Computer Science and Technology, Beijing Institute of Technology, China

Saraju P. Mohanty
Department of Computer Science, University of North Texas, USA

***Abstract*—Unmanned Aerial Vehicles (UAVs) are gaining tremendous attention due to their flying nature. To complete the task efficiently, multi UAV systems are a good choice as compared to a single UAV system. However, multi-UAV systems introduce issues such as high dynamics, limited battery, and frequent changes in topology. Software control is required to solve these issues. Thus, Software Defined Networking (SDN) is an excellent candidate to separate control logic from forwarding elements and provide high-level programming abstractions. However, due to architectural constraints, applying SDN introduces some new challenges, including uneven load on multiple links between source and destination. This irregular load also affects UAVs' battery consumption, necessitating an adequate solution to meet these challenges fully. This paper proposes an SDN-based framework for UAV elements that monitors frequent changes in the network topology. Based on this monitoring, an algorithm is designed which distributes traffic load evenly on different links of multi-UAV systems. UAV networks have limited resources; therefore, battery limitations are also considered, and traffic is shifted to a path where elements have more battery. Moreover, a flight control mechanism is proposed to avoid collisions due to the high dynamics of UAVs. Extensive simulation results show that the traffic load is distributed evenly on multiple links connecting different systems with less battery consumption.**

IN RECENT YEARS, Unmanned Aerial Vehicles (UAVs) attracted significant interest and research in consumer and military domains [1]. They can provide disaster warnings on time, speed up recovery and rescue operations, and carry medical supplies in case of a disaster or the absence of public communication. Public safety, police, and transportation management are public uses. Similarly, in the military, these elements are deployed to perform operations like; surveillance, battlefield inspection, and mapping of inaccessible areas. Compared to a single UAV system, multiple UAV systems are more reliable due to redundancy and survivability to efficiently complete a task. These multiple UAVs form a UAV Network (UAVNet) [2] which introduces issues such as uneven load on wireless links, battery limitations, frequent changes in topology, and high dynamics. To overcome these issues, a centralized and programmable solution is required. Software Defined Networking (SDN) [3] is a good candidate that separates control logic from data plane elements and provides high-level programming abstractions. Data plane elements become simple forwarding nodes, whereas decision-making is shifted to a centralized control plane. Data plane updates control plane by using well-defined Application Programmable Interfaces (APIs) [4]. In response to this information, a global view is generated, and forwarding rules are pushed by the control plane. In this article, considering the problems mentioned above, a dynamic SDN framework in UAV communication is proposed for path selection based on the UAV elements' link load and battery level. Moreover, a flight control mechanism is proposed. The main contributions of this paper are;

- A novel load balancing algorithm is proposed for path selection based on link load and battery level of UAV elements.
- It uses OpenFlow statistics to dynamically adjust the paths in case of congestion, link failure, and/or availability of a better path.
- The battery consumption of UAV elements is reduced by shifting the load on elements with high battery levels and validating the effectiveness of the proposed approach with an example.
- The proposed solution takes decisions based on the link load and battery level. It also takes heuristic decisions, but the objective is to select the optimal path.
- It handles the high dynamics of UAV flights and uses the programmability of SDN to avoid collisions.

The rest of the article is organized as the following section describes the related work. Then background and problem statement are presented, followed by the system overview. After that, the performance evaluation and analysis are described. Finally, the conclusion and future work are discussed.

Related Work

SDN has been extensively deployed in wired networks; however, in recent years, it has also been applied to wireless scenarios. For example, it has diverse applications (e.g., path selection, channel allocation, etc.) in MANET, VANET, and UAVNets. In UAVNets, there are frequent changes in topology due to battery drainage and intermittent links. Moreover, the programmability of UAVNet can reduce the threat of collisions, performance improvement, path optimization for data routing, and change packet transmission range due to energy constraints. Gupta et al. [5] exploits a survey on various issues in UAVNets and discusses different SDN capabilities to solve these issues. For example, mobility support, flexible strategies for routing and switching, dealing with unreliable wireless links, network greening, and interference reduction. Rehman et al. [6] address the controller placement problem in SDN-based UAVNets. The communication overhead and end-to-end delay of the control packets exchanged between UAV elements and the controller is considered. Results indicate that controller placement should be in a central location to minimize the number of hops. Moreover, a merging approach has discussed that buffer and merge the control packets to reduce the overhead. In [7] same authors propose an algorithm to optimize the UAV position so that overall throughput is maximized in disaster area UAVNets using SDN. Since the information is insufficient at the time of the initial deployment of UAVNets, UAV positions may not be optimal. Later, the UAVs can share the user position information with the controller. In response to this information, the controller can use a centralized algorithm to reassign the positions to UAVs. Recently, Guerber et al. [8] address security issues on a swarm of drones using IP-based table filtering (a well-known approach for ad hoc networks) and SDN. It

Digital Object Identifier 10.1109/MCE.2022.Doi Number

Date of publication 00 xxxx 0000; date of current version 00 xxxx 0000

not only tackles attacks from outside the network but also within the network and enforces security rules against an attack. In [9], the authors propose a monitoring platform in SDN controller that analyzes and manages the information of UAV elements. Based on the information provided by the monitoring system, a load-balancing algorithm is proposed based on the local and global variance. Moreover, it shifts the load to other paths once the battery of a particular UAV is finished. Secinti et al. [10] propose a protocol that computes multiple paths from source to destination where these multiple paths are working on different access technologies to overcome the intentional jamming or environmental obstructions. In case of obstructions, UAVs use an alternate path and improve resilience in the network. In [11], authors propose a solution for topology management using SDN to ensure user connectivity and load balancing. In addition, the authors use a spring virtual force approach to avoid collisions, maintain adequate distance among UAVs for communication, and maximize the coverage area. Another study, in [12], authors reuse the delivery drones for crowdsensing. For this purpose, authors jointly optimize delivery weights, sensing time, and route selection while considering the limited battery of UAVs. In [13], UAVs are used to collect the data from IoT devices to save the energy consumption of these devices. Moreover, the data freshness is ensured while determining the minimum number of UAVs. *LB-UAVnet* [14] switches the communication links and perform packet routing dynamically. The authors use a threshold flow load to compare with a load of each UAV. Based on these threshold values, the flow is shifted from overloaded UAVs to underutilized UAVs.

However, a different approach is proposed for load balancing concerning link load and battery level of UAVs. It computes the weight value, including link load and battery level, at each link and node. Also, a flight control mechanism is proposed that extracts the current position of the UAV. Based on the distance between nodes, it controls the UAV flights to avoid collisions.

Background and Problem Statement

This section provides some details of the challenges while deploying UAVNets and how SDN can help to address these challenges. In addition, the problem formulation is also presented in this section.

Background

There are several challenges of UAVNets which require adequate consideration. Some details of these challenges are as follows;

Power Limitations: Since fixed power cannot be supplied to UAVs, batteries are carried out to complete the missions. Due to battery limitations, these elements cannot operate for a long time if energy utilization is not scheduled correctly.

Routing Strategies: In case of battery exhaustion, UAVs are replaced with new ones, which results in link disruption. Also, links can be affected by environmental conditions or interference. Due to intermittent connections, efficient routing, reducing latency, and ensuring reliability become challenging.

High Dynamics: UAV flights are either proactive or dynamic. In a proactive approach, UAVs follow a preprogrammed flight plan, whereas complex automation systems and intelligent inter-UAV coordination protocols are used in a dynamic approach. Since the movement of UAVs is usually in 3 dimensions with a higher speed than Vehicular Ad hoc Network (VANET) and Mobile Ad hoc Network (MANET) thus, optimization of UAV location and trajectories is an important issue.

To address these challenges, centralized control, external computation, and programmability are required, which is provided by SDN. As SDN is a centralized architecture thus, all the computation is shifted to the controller, which reduces the power consumption of UAVs in SDN-based UAVNets. In addition, radios can be turned off for power conservation. Furthermore, the global view of the network makes path selection simple and straightforward [15]. Since SDN provides an elastic and programmable network; therefore, programmability helps to reduce collision risks. However, applying SDN in UAVNets introduces new challenges such as uneven load over the links, energy consumption, and collision among UAVs.

Problem Statement

In SDN, only programmable switches are used to shift decision-making to a centralized controller. These switches provide their information to the controller. In response to this information, the controller generates a global view of the network and installs flow rules on underlying devices. Once a flow rule is installed, devices forward data according to these rules till the end of the flow. In the case of multiple paths, there can be an uneven distribution of load. This situation also

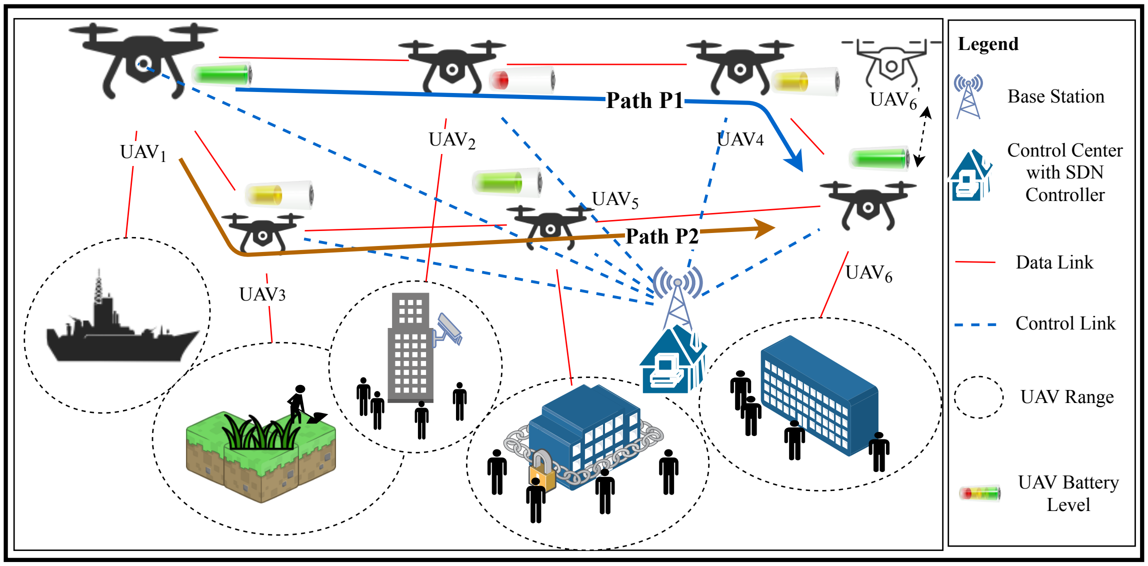

Figure 1. Sample Topology

results in UAVs' battery exhaustion, which ultimately needs to be replaced.

Assume that a network has a topology graph G that can be represented as G=(V,E), where V = $\{v_1, v_2, v_3...v_m\}$ represents UAVs, and E = $\{e = (u,v) : u,v \in V\}$ is a set of edges to connect V number of UAVs. Any path between consumers (i.e., a source s and destination t) is represented as P_s^t = $\{v_s, v_1, v_2...v_t\}$. Notably, traffic originator is a consumer, however, from a controller's perspective, it is UAV from where the flow begins. Hence, UAVs are used to represent source and destination. Moreover, every node V has a weight W_e, where $e \in$ [1-M] is a series of M non-negative weights or cost functions. Finally, the total weight of path p_i can be calculated as; W_{p_i}=$\sum_{en=0}^{m} W_{en}$ where W_{en} is weight of each link e or node n in path p_i and can be represented as; W=α_e+$\frac{1}{\beta_n}$ where α_e represents the load of link e whereas, β_n is the battery level of each node n. Notably, a higher value of battery has a less weighted value. One major reason to use this fraction is to handle the overall weight values of different paths. For example, if two paths have similar link load but significantly different battery levels or vice versa, then the overall weight value can help to select the best path. Formally, the problem can be defined as; a source v_s intends to communicate with v_t, where v_s, $v_t \in V$ and path p_i is said to be best optimal path if: $W_{p_i} \leq W_{p_j}$ where p_i, p_j∈P∧ $p_i \neq p_j$.

The problem statement can be presented with the help of sample topology as shown in Figure 1. These UAVs with different coverage areas and battery levels are connected using a data link to form a UAVNet. The SDN-based control center is associated with these UAVs using a control link. In addition, the UAVNet has multiple paths between the source and destination. SDN-based UAV networks have four components: UAVs, consumers, UAV control center, and SDN controller. UAVs are equipped with WiFi Access Points (APs) with different transmission ranges and provide various services to consumers in the sky. These consumers are terminal devices (e.g., mobile nodes or sensors) distributed unevenly on the ground using UAVs to transfer their data. The position of UAVs can be controlled with the help of a UAV control center which also has information about the current location and battery level of UAVs.

SDN controller communicates with all UAVs coming under its domain using OpenFlow protocol to receive packets from UAVs. It also extracts topology updates, link states, and statistics (e.g. latency, packet loss ratio, and link utilization). After collecting this information, the controller can forward it to management plane applications. Since there can be multiple paths between source and destination, for example, let UAV_1 and UAV_6 are the source and destination nodes and there are two possible routes among this pair which can be represented as path *P1* and *P2* (i.e., $UAV_1 \rightarrow UAV_2 \rightarrow UAV_4 \rightarrow UAV_6$ and $UAV_1 \rightarrow UAV_3 \rightarrow UAV_5 \rightarrow UAV_6$) as shown in Figure 1.

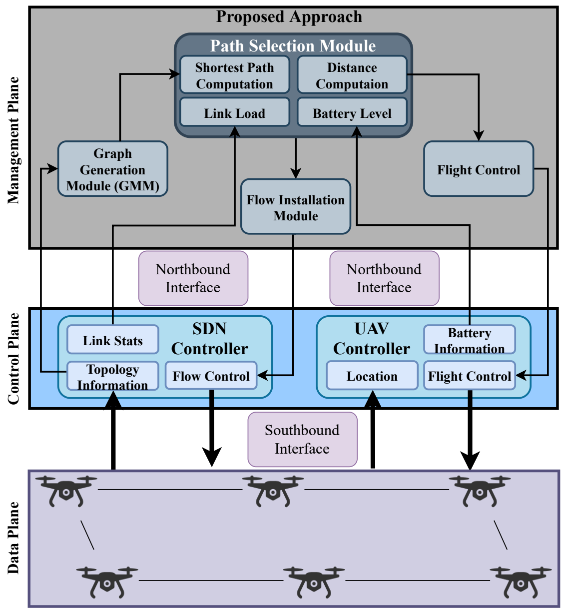

Figure 2. System architecture

As soon as UAV_1 receives a packet, it forwards the packet to the controller as Packet_IN message. In response to the Packet_IN request, the controller generates a Packet_OUT message which contains flow rules for each UAV of path *P1*. After the flow rule installation, the same path is followed for the flow duration. This situation may cause an uneven load on path *P1* and path *P2*, as all the traffic follows path *P1*, which makes it overloaded and path *P2* becomes idle. The transmission of packets directly affects the battery of UAVs; therefore, battery consumption of the overloaded path (i.e., *P1*) becomes higher as compared to the idle path (i.e., *P2*). A dynamic and adequate solution is required to find and redirect traffic to the best optimal path, which can be robust in determining the network state and adjusting the flow path to optimize the load.

Due to high dynamics, UAVs can collide with each other or go in the wrong direction. For example, UAV_6 in Figure 1 starts moving in wrong direction with its current position at $UAV_6^{'}$. In this scenario, UAVs can collide with the ground or each other. Also, UAVs can go out of range in swarm-based operations. The collision issue can be resolved with the help of a flight control mechanism based on the distance between two nodes or UAV distance from the ground. Similarly, the transmission range of neighboring UAVs can be used if a UAV goes out of range.

System Overview

This section discusses the load balancing among UAV elements inside the control of its domain. Each source and destination pair is connected via multiple paths, as shown in the sample topology (Figure 1). The proposed framework works as a management application in the management plane and an integral part of the controller, as shown in Figure 2. It utilizes information forwarded by the data plane elements to find an optimal path for a particular flow and pushes forwarding rules to UAVs using the SDN controller. It also uses this information for the flight control of UAVs. There are various modules of the proposed approach that are discussed below.

Load Balancing

The load balancing collects the information from the controller and installs flow rules on forwarding elements with the help of the following modules.

Graph Generation Module: The Graph Generation Module (GGM) generates the graph G=(V,E), which represents the topology of the entire domain. It collects the information (e.g. *IP and MAC addresses*, and *port connectivity of UAVs*) of network devices. The network state is collected in JSON format with the help of interfaces. It also retrieves each UAV's port information, which helps find link load. Finally, this module generates a graph that helps to find the shortest paths from source to destination using Dijkstra's algorithm. These paths are forwarded to the path selection module for optimal path selection.

Path Selection Module: The main objective of this module is to select the best optimal path for flow installation. The best optimal path selection is based on two parameters, link load and battery level of each node.

1) **Link Load:** OpenFlow statistics are used to find the load on each link. Since the port statistics of each UAV can be collected from the SDN controller, which is further used to find the link load. The weight value of a path with respect to link load is computed as a sum of load on each link. For example, for path 2 in Figure 1 (i.e. $UAV_1 \rightarrow UAV_3 \rightarrow UAV_5 \rightarrow UAV_6$) the weight can be computed as; $W_{p_2} = Load_{1,3} + Load_{3,5} + Load_{5,6}$ where $Load_{1,3}$, $Load_{3,5}$, $Load_{5,6}$ represents the link load between the pairs (UAV_1, UAV_3), (UAV_3, UAV_5), and (UAV_5, UAV_6) respectively.

2) **Battery Level:** Similar to link load, battery level of each node is retrieved from UAV controller. Since all shortest paths have same number of nodes therefore, to find the best optimal path with respect to battery, the battery levels of all nodes (except source and destination nodes) are compared with respective node of alternate path. For example, in paths $UAV_1 \rightarrow UAV_2 \rightarrow UAV_4 \rightarrow UAV_6$ and $UAV_1 \rightarrow UAV_3 \rightarrow UAV_5 \rightarrow UAV_6$ the battery levels of UAV_2 and UAV_3 are compared and battery level of UAV_4 is compared with UAV_5. Moreover, if battery of a particular UAV is less than a threshold value then that path is not selected. In this article, threshold value is set to 10%. For example, battery level of UAV_2 in path 1 is less than threshold value hence, this path will not be selected. Finally, these weight values are used to find the optimal path between the source and destination pair, where a less weight value signifies better optimality. Moreover, the list of associated UAVs is forwarded to the flow installation module.

Flow Installation Module: The flow installation module is responsible for installing flow rules on UAVs forwarded by the path selection module. It translates the path information into OpenFlow rules and adds/removes them at each UAV. It removes the old path after the installation of a new path.

Flight Control

A UAV controller collects the information for data plane elements in the flight control mechanism. This information contains the current position and battery level of UAVs. As mentioned above, battery level information is used to find the optimal path. However, the location of UAVs is forwarded to the distance computation module. This module finds the distance among nodes and nodes from the ground. If the distance between two nodes continuously increases, it checks the transmission range of neighboring UAVs. Similarly, a decrease in distance between two nodes may result in a collision. Based on this information, the distance computation module takes a decision. It forwards the position of the UAV to the flight control module, which provides instructions to UAV elements with the help of the UAV controller.

Performance Evaluation and Analysis

The performance of the proposed solution is evaluated by designing the topological setup in *Mininet_WiFi* [1]. The *Floodlight* [2] SDN controller is adopted. 6 UAVs (APs) are in the air with mobility and make the same topology as shown in Figure 1. It is worth mentioning that UAVs are designed by modifying Mininet_WiFi while considering their battery lives by combining the station and access point classes. Moreover, a 3D mobility model is used to ensure the real scenario of UAVs. The battery level of UAVs is set to 90%. The battery reduction of UAV is based on; fixed-rate, distance, and packet transmission rates. Due to flight, the battery of each UAV is reduced at a fixed rate. Since transmission power is directly proportional to distance, higher distance consumes more battery. Similarly, battery life relates to the number of packets transmitted; thus, more battery is consumed if any UAV transmits more packets. Each UAV is connected to one or more UEs (stations). *Iperf* [3] generates traffic between the source and destination pair. To show the effectiveness of the proposed approach, a comparison with the state-of-the-art LB-UAVnet is presented. In addition, the proposed approach is compared with the default controller decisions (i.e., without applying the proposed approach).

In this article, one of the significant parameters for evaluation is Round Trip Time (RTT), as shown in Figure 3 where the green axis represents minimum, average, and maximum RTT without applying load balancing algorithm (w/o LBA). In contrast, the blue axis shows minimum, average, and maximum RTT when the load balancing algorithm (w LBA) is applied. The upper value indicates the maximum, the lower value shows the minimum, and the middle value represents the average RTT. In each experiment, five packets are transmitted between the source and destination pair. It can be observed that the minimum RTT in both cases is almost similar, whereas there is a significant difference in maximum RTT. The maximum RTT without applying LBA is more than 6 ms in all experiments, whereas, in experiment 3, this value is 8 ms. On the other hand, when LBA is applied, this value does not exceed 4.9 ms. Similarly, the average RTT is above 5 ms when LBA is not applied, but it is reduced to 4.84 ms after applying the algorithm.

Figure 4 shows the percentage of packets processed on each UAV working as APs. The percentage of packets processed on each UAV without applying

---

[1] https://github.com/intrig-unicamp/mininet-wifi

[2] http://www.projectfloodlight.org/floodlight/

[3] https://iperf.fr/

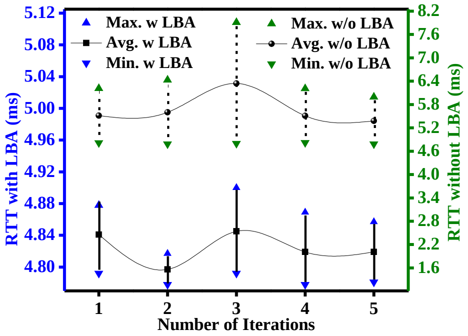

Figure 3. Round Trip Time with and without using load balancing algorithm.

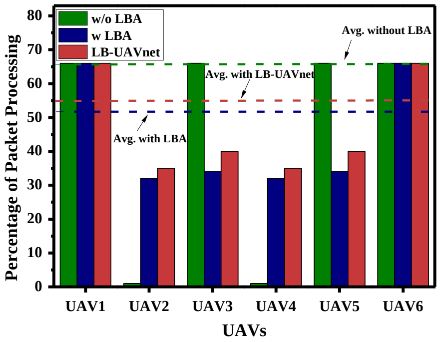

Figure 4. Percentage of packets processed on each node.

LBA (w/o LBA) is indicated as a green bar whereas the red bar shows LB-UAVnet [14], and the blue bar shows when LBA is applied. Moreover, the average values of each approach are presented in the dashed lines. It can be observed that without LBA the path selected is $UAV_1 \rightarrow UAV_3 \rightarrow UAV_5 \rightarrow UAV_6$ (i.e., Path 2 as shown in Figure 1). It can be seen that the percentage of packet processing is more than 60% on these UAVs. On the other hand, packet processing over UAVs of other path (that includes UAV_2 and UAV_4) is almost 0%. Notably, the average value without LBA is computed by considering the UAVs processing the packets, whereas the UAVs with zero workloads (i.e., UAV_2 and UAV_4) are neglected. Similarly, the packet processing on each UAV with LB-UAVnet is between 35% to 40%. On the other hand, it can be evident that when the proposed approach is applied, the packet processing of UAV_2 and UAV_4 is also increased to 30% whereas load on UAV_3 and UAV_5 is reduced. The packet processing after applying LBA is between 30% to 35%. It indicates that the proposed approach distributes load evenly on multiple paths. Furthermore, the proposed approach outperforms LB-UAVnet as well. One possible reason is that LB-UAVnet balances the load based on flow rules rather than network traffic.

To compute link utilization on all possible paths from source to destination without battery considerations, two applications are developed. One application generates the observed data flow, while the other generates random traffic across the network to model load on links. The same model is also applied with the LB-UAVnet [14] to show the effectiveness of the proposed approach. It can be observed from Figure 5 that without LBA, the usage of the initial path is maximum, whereas the second path is neglected. For example, path *P2* is overloaded with more than 50% utilization, whereas path *P1* usage is 0% for the observed flow. It is important to note that the load-generating application directs traffic randomly on the different links. Hence it is not possible that the other path is overloaded with it. Similarly, LB-UAVnet is balancing the load, but it is not optimal. Contrary to these, the path of the observed flow utilizes both the links evenly after applying LBA, and link utilization does not exceed 30%.

Since UAVNets are energy-constrained, the power limit of UAVs is also considered. It can be evident from Figure 6 that the battery level of each UAV is set to 90%, and the flow completion time is 25 seconds. Initially, the controller selects path *P2* without applying LBA. Therefore, the battery level of this path is scaled down to 0% within 13 seconds. Since path *P2* is overloaded, and UAVs on this path are forwarding more packets thus, the battery consumption rate is very high. It can also be observed that the battery level of path *P1* is reducing slower during this period. As soon as the battery of path *P2* reaches 0%, battery consumption of path *P1* becomes higher and approaches 0% in the next 10 seconds. One possible reason can be that all the traffic transmitted via path *P2* is now shifted to path *P1*. However, when similar traffic is transmitted after applying LBA, the battery consumption of both paths is stable. It also shows that resources are adequately utilized, and 5% battery remains for the observed flow.

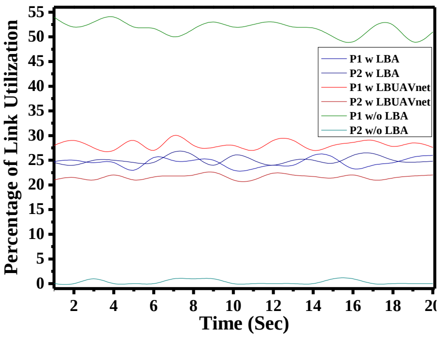

Figure 5. Link utilization of UAVs in different paths.

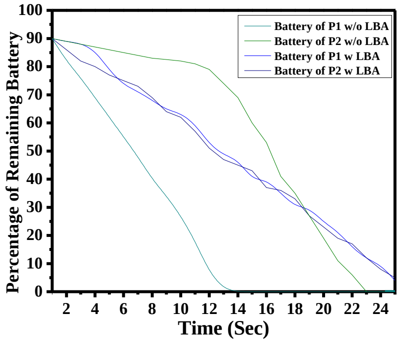

Figure 6. Battery life of UAVs in different paths.

Conclusion

This article proposes an SDN-based framework for UAVNets, which works in the management plane of SDN architecture. The proposed approach works with SDN and UAV controllers for load balancing and flight control mechanisms. For load balancing, it collects the information from the SDN controller (e.g., topology information and link stats) whereas battery information from the UAV controller. Based on this information, it redirects the traffic to a less loaded path or path with UAVs having more battery. Since there are limited resources in UAVNets, the proposed work utilizes the resources properly. The flight control mechanism uses information (e.g., location and position of UAV) and pushes instructions if UAVs are going in the wrong direction. In SDN, a single controller can handle a limited number of UAVs; hence, scalability is an issue in SDN. In the future, the load can be balanced in multiple SDN domains with distributed SDN controllers with further optimization using intelligent algorithms.

Acknowledgement

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2020R1A2B5B01001758) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00590, Decentralized High Performance Consensus for Large-Scale Blockchains), National Natural Science Foundation of China No. 61772077, 61370192, and the Beijing Natural Science Foundation No. 4192051.

REFERENCES

1. T. D. e Silva, C. F. E. de Melo, P. Cumino, D. Rosario, E. Cerqueira, and E. P. De Freitas, “Stfanet: Sdn-based topology management for flying ad hoc network,” *IEEE Access*, vol. 7, pp. 173 499–173 514, 2019.
2. S. Morgenthaler, T. Braun, Z. Zhao, T. Staub, and M. Anwander, “Uavnet: A mobile wireless mesh network using unmanned aerial vehicles,” in *2012 IEEE Globecom Workshops*, Dec 2012, pp. 1603–1608.
3. D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
4. Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y. Wang, “A comprehensive survey of interface protocols for software defined networks,” *Journal of Network and Computer Applications*, vol. 156, p. 102563, 2020.
5. L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in uav communication networks,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1123–1152, Secondquarter 2016.
6. S. ur Rahman, Y. Cho, A. Khan, and G. Kim, “Deployment of an sdn-based uav network: Controller placement and tradeoff between control overhead and delay,” in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2017, pp. 1290–1292.
7. S. ur Rahman, G. Kim, Y. Cho, and A. Khan, “Positioning of uavs for throughput maximization in software-defined disaster area uav communication networks,” *Journal of Communications and Networks*, vol. 20, no. 5, pp. 452–463, Oct 2018.
8. C. GUERBER, N. LARRIEU, and M. ROYER,

“Software defined network based architecture to improve security in a swarm of drones,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2019, pp. 51–60.

9. X. Zhang, H. Wang, and H. Zhao, “An sdn framework for uav backbone network towards knowledge centric networking,” 04 2018, pp. 456–461.
10. G. Secinti, P. B. Darian, B. Canberk, and K. R. Chowdhury, “Sdns in the sky: Robust end-to-end connectivity for aerial vehicular networks,” *IEEE Communications Magazine*, vol. 56, no. 1, pp. 16–21, Jan 2018.
11. F. Pasandideh, A. A. S. d. Silva, E. Pignaton de Freitas *et al.*, “Topology management for flying ad hoc networks based on particle swarm optimization and software-defined networking,” *Wireless Networks*, vol. 28, no. 1, pp. 257–272, 2022.
12. C. Xiang, Y. Zhou, H. Dai, Y. Qu, S. He, C. Chen, and P. Yang, “Reusing delivery drones for urban crowdsensing,” *IEEE Transactions on Mobile Computing*, 2021.
13. J. Zhang, Z. Li, W. Xu, J. Peng, W. Liang, Z. Xu, X. Ren, and X. Jia, “Minimizing the number of deployed uavs for delay-bounded data collection of iot devices,” in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.
14. C. Singhal and K. Rahul, “Lb-uavnet: Load balancing algorithm for uav based network using sdn,” in *2019 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC)*. IEEE, 2019, pp. 1–5.
15. L. Zhu, M. M. Karim, K. Sharif, C. Xu, F. Li, X. Du, and M. Guizani, “Sdn controllers: A comprehensive analysis and performance evaluation study,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–40, 2020.

Zohaib Latif is a Postdoc Research Associate with Department of Computer Science, Hanyang University, Seoul, South Korea. Contact him at zohaiblatif@hanyang.ac.kr.

Choonhwa Lee is a Professor with the Department of Computer Science, Hanyang University, Seoul, South Korea. He is the corresponding author of this article. Contact him at lee@hanyang.ac.kr.

Kashif Sharif is an Associate Professor (Research) with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Contact him at kashif@bit.edu.cn.

Fan Li is a Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Contact her at fli@bit.edu.cn.

Saraju P. Mohanty is a Professor with the Department of Computer Science, University of North Texas, USA. Contact him at saraju.mohanty@unt.edu.