SAN: Self-Adaptive Navigation for Drone Battery Charging in Wireless Drone Networks

Jinyong Kim*, Jinho Lee*, Jaehoon (Paul) Jeong†, Hyoungshick Kim*, Jung-Soo Park‡, and Taeho Kim‡
* Department of Computer Science & Engineering, Sungkyunkwan University, Republic of Korea
† Department of Interaction Science, Sungkyunkwan University, Republic of Korea (Corresponding Author)
‡ Electronics and Telecommunications Research Institute (ETRI), Republic of Korea
Email: {wlsdyd0930,jinholee,pauljeong,hyoung}@skku.edu, {pjs,taehokim}@etri.re.kr

Abstract—**This paper introduces an optimal path finding problem for drone battery charging where their batteries should be charged for the travel from a source to a destination as needed. We present a practically reasonable heuristic to solve the problem by monitoring drones' battery status and traffic conditions in real time through a cloud-based service called traffic control center. This study will be the cornerstone of path finding problems for drone battery charging in drone networks.**

I. Introduction

Recently, drones have been aggressively developed for hard duties (e.g., military service, delivery service, and disaster relief), so people expect drones to be commercialized in the near future. Drones can use various power source (e.g., battery, solar fuel, hydrogen fuel), but they usually use battery instead of other energy source because other energy resources are unsafer than the battery. With this reason, people prefer to use safe battery. Furthermore, automated battery replacement and recharge system are actively researched. However, the battery-based approach has one disadvantage. The disadvantage is that operation time is short. Of course, operation time is increasing, but it is evident that this time is still too short to execute various services. Therefore, it is necessary that drones can recharge the battery during the execution time of various services [1]–[3].

This paper is the first work to propose a method about efficient battery charging for drones during their services. Our method is is based on Dijkstra algorithm considering charging time and queuing time at QCMs. In order to reduce the queuing delay, we suggest to utilize a virtual metric called *Congestion Contribution* [4]. This Congestion Contribution measures the impact of each drone on the whole drone networks. Drones report their information (e.g., speed, current position, and destination position) to the cloud-based management system called Traffic Control Center (TCC). Then, TCC broadcasts the gathered statistics to the drones. With this information, the drones can decide appropriate battery charge stations while they avoid to cause excessive queuing delay.

The rest of this paper is organized as follows. First of all, we summarize related work in Section II. Section III describes problem formulation in an optimal path finding algorithm for charging drone battery. Section IV explains our drone battery charging scheme. Section V discusses challenging research issues. We finally conclude this paper along with future work in section VI.

II. Related-Work

Recently, a drone (called unmanned aerial vehicle) industry emerges as a new industry. Originally, a drone was developed to perform military service (e.g., reconnaissance, monitoring, and bombing). However, lately, the drones are rapidly being developed for various purposes because various enterprises (e.g., Amazon and Google) declare to make use of drones for a commercial use (e.g., shooting, agriculture, infrastructure management, data sharing, outdoor/indoor navigation, delivery, and rescue) [5], [6]. By new services of the drone such as this, many countries and enterprises (e.g., Parrot [7], Nixie [8], Dji [9], and Airinov [10]) have put a lot of efforts into reserach (e.g., software platform, battery, battery recharging machine, and communications) for drone development.

The software platforms of the drone are researched to utilize drones for various services [11]. Initially, the software platforms of drone were developed only for flight control. However, recently, the software platforms of the drone are developed to carry out not only flight but also various services [5], [6]. Because of this change, the software platforms of the drone are actively researched.

Because drones usually use a battery to fly, the battery of the drone is researched. Life time of drones is normally about 15-40 minutes [12]. However, this life time is too short to execute various services. Because of this disadvantage, the battery charging of drones at battery charging machines is necessary for smooth operation. In other to solve this problem, much research has been done about battery life time and charging machine of drones [2], [13], [14].

Research about communications and control of drones has been performed [15]. For various services, drones are controlled by the Traffic Control Center (TCC). The drones are researched about communications with other devices by using satellites and wireless communications (e.g., WiFi). Furthermore, much research has been done about communications security in other to prevent malicious users from accessing drones.

The virtual metric *Congestion Contribution* is introduced in [4]. This idea is developed to estimate congestion and prevent future congestion. Vehicles can cause a congestion as all of them take a greedy choice. [4] evaluates the impact of each vehicle on congestion. Then, vehicles takes paths considering the future congestion.

III. Problem Formulation

In this section, we articulate the goal, drone network architecture, and assumptions to recharge the battery of drones in wireless drone networks. Given the trajectories of drones in wireless drone networks, our goal is to assign appropriate battery charge stations to drones in order to reduce overall travel delay. Fig. 1 shows a network of flying drones and Quick Battery Charge Machine (QCM). Since this QCM is located in land, drones should land to recharge the battery.

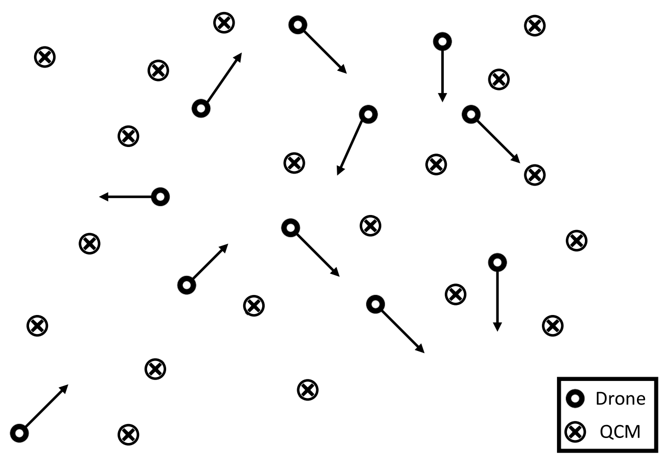

Fig. 1. Wireless Drone Networks

A. Drone Network Architecture

We describe a drone network architecture to support this paper in wireless drone networks. Our drone network architecture consists of (i) Traffic Control Center (TCC), (ii) Quick Battery Charge Machine (QCM), and (iii) Drone:

- **Traffic Control Center (TCC)**: TCC is a drone traffic management node that maintains the trajectories and locations of drones for the location management as used in Mobile IPv6 [16]. the TCC has up-to-date traffic statistics of drones, such as average speed, current position, and destination position in the wireless drone network under its management.
- **Quick Battery Charging Machine (QCM)**: QCM is a station which can recharge a drone's battery. It has one in coming queue for battery charging drones.
- **Drone**: Drone is an unmanned aerial vehicle flying from a source position to a destination position without a pilot. Drones know waiting delay for each QCM through the communication with TCC.

B. Assumptions

We assume the following to design an efficient path finding algorithm in wireless drone networks:

- All drones can communicate with the TCC and periodically report their own mobility information (e.g., speed, current position, destination position), to the TCC.
- The TCC knows QCM position and mobility information of all drones.
- Each QCM can accommodate only one drone for charging battery at a time.

IV. Drone Path Planning Scheme

In this section, we propose our drone battery charge scheme. This scheme uses estimated delays (i.e., waiting delay at QCMs and travel delay from source position to destination position) and an enhanced shortest path algorithm from source position to destination position. When a drone ia already using the QCM, other drones have to wait for their turn for battery charge in a waiting queue at the QCM, so will have waiting delay. By this waiting delay, the enhanced shortest path algorithm considers estimated waiting delay of drones for each QCM. With this shortest path algorithm, we can assign an appropriate QCM to each drone in order to reduce the overall delay from source position to destination position.

A. Time Prediction

We explain how to estimate time to assign appropriate QCMs to drones. If drones need to recharge their battery, drones report their own mobility information (e.g., speed, current position, destination) to the TCC. With this information, the TCC can calculate waiting delay t QCMs and travel delay from source position to destination position because distance and the speed are used to compute the travel delay between two positions. Also, because drones are unmanned aerial vehicles, the speed of drones can continually be kept to a constant. Thus, we can estimate the overall travel delay.

B. An Optimal Path Planning Algorithm for Drone Battery Charging

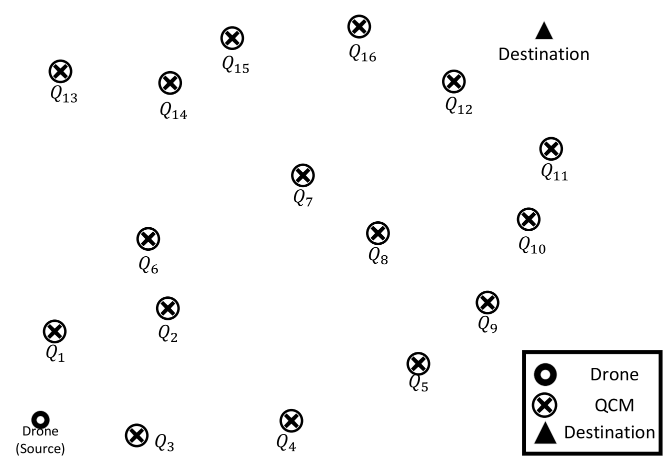

Fig. 2. Finding Path to the Destination

We assume that a drone is planning its path to a destination and QCMs are located as shown in Fig. 2. The drone can get this position information from TCC. With this position information of QCMs, a drone can construct a reachability graph based on the maximum travel distance with its fully charged battery. Unlike conventional vehicles, aerial drones reach any QCM within its maximum travel distance. Then, a reachability graph is a mesh network. However, drones can reach limited number of QCMs due to its constrained battery budget. Thus, the reachability graph is not constructed as a full mesh. Therefore, we need to find a shortest path on a

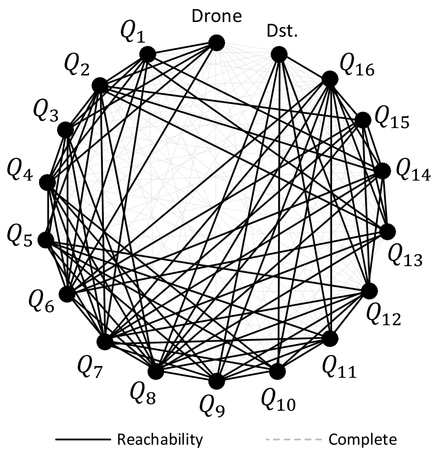

Fig. 3. Reachability Graph

partial mesh topology. The reachability graph is shown in Fig. 3. If we do not consider the recharging time and the waiting delay at QCMs, we can find the shortest path with Dijksta algorithm. Here, we propose our enhanced path finding algorithm considering recharging time and the waiting delay. Furthermore, we utilize a virtual metric called *Congestion Contribution* (CC) [4] to indicate the congestion level at the QCM for drones. This CC prevents the situation that many drones try to recharge at a QCM simultaneously, which cause a long queueing delay. This case may happen if drones take a greedy choice with Dijkstra path.

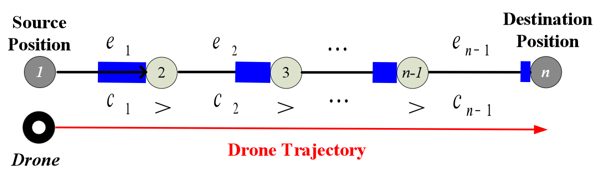

Fig. 4. Congestion Contribution Model

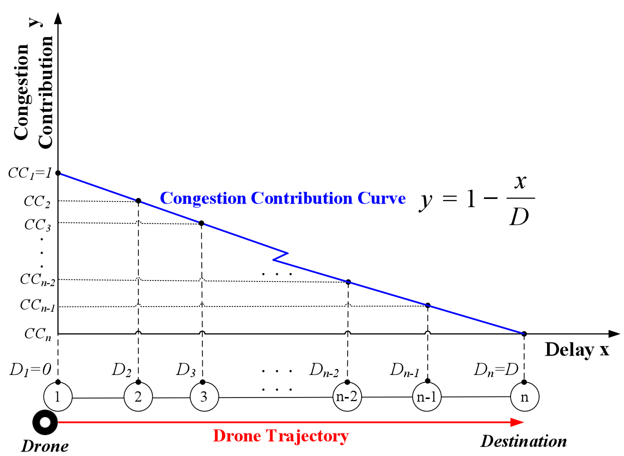

Fig. 5. Congestion Contribution

Congestion Contribution (CC) is illustrated in Fig. 5. If a drone set its path and report it to TCC, TCC calculate and add CC for each QCM on the path. CC is proportional to $-\frac{1}{D_i}$ where D_i is the distance between the source position and the Q_i. Then,

$$y = 1 - \frac{x}{D}. \tag{1}$$

Algorithm 1 Queueing-Constrained Shortest Path Algorithm

1: **function** CONSTRUCT-QUEUE-CONSTRAINED-SHORTEST-PATH(G, u, v, α)
2: $P_{qsp} \leftarrow \emptyset$ ▷ P_{qsp} will contain the list of QCMs for a queue-constrained shortest path.
3: $D_{uv} \leftarrow$ Compute-Dijkstra-Path-Value(G, u, v) ▷ D_{uv} is the time-wise shortest delay from u to v in G by Dijkstra's shortest path algorithm.
4: $\Delta_{uv} \leftarrow \alpha \times D_{uv}$ ▷ Δ_{uv} is the α-percent delay increase for D_{uv}.
5: $K \leftarrow$ Compute-k-Smallest-Congestion-Increase-Paths(G, u, v) ▷ compute the next k smallest congestion increase paths arranged in nondecreasing order by Yen's k-shortest-path algorithm.
6: $n \leftarrow$ Count-Path-Numbers(K) ▷ count the number of paths in K.
7: **for** $i \leftarrow 1, n$ **do**
8: $D \leftarrow$ Compute-Path-Value(K, i) ▷ compute the overall delay for the ith path in K.
9: **if** $D \leq D_{uv} + \Delta_{uv}$ **then** ▷ check the overall delay constraint of α-percent increase.
10: $P_{qsp} \leftarrow$ Get-Path(K, i) ▷ get the ith path in K.
11: **return** P_{qsp}
12: **end if**
13: **end for**
14: $P_{qsp} \leftarrow$ Compute-Dijkstra-Path(G, u, v) ▷ set P_{qsp} to the time-wise shortest path from u to v in G by Dijkstra's shortest path algorithm.
15: **return** P_{qsp}
16: **end function**

As a drone make a decision based on this CC value, it can avoid long queueing delay and prevent it at the same time. Algorithm 1 is based on the Delay-Constrained Shortest Path Algorithm in [4]. Let us explain our Delay-constrained Shortest Path algorithm in details as follows:

In Algorithm 1, a queueing-constrained shortest path P_{qsp} is returned for the input of the reachability graph G, source u, destination v, and α-increase value. In line 2, P_{qsp} a list of QCMs for the queueing-constrained shortest path. In line 3, D_{uv} is calculated as follows:

$$D_{uv} = \sum_{(i,j) \in E_p} T_{ij} + \sum_{k \in V_p} C_k + \sum_{k \in V_p} W_k, \tag{2}$$

where

- P: Dijkstra path from u to v. P=(V_p, E_p),
- D_{uv}: overall delay for P,
- T_{ij}: travel delay from Q_i to Q_j,
- C_k: charging delay at Q_k, and
- W_k: waiting delay at Q_k.

4 computes the α-percent delay which is α-percent increased from D_{uv}. In line 5, k shortest paths [17] are calculated by *Compute-k-Smallest-Congestion-Increase-Path()* in terms of *Congestion Contribution*. In line 6, we set n to the number of paths in K. In lines 7-13, a queueing-constrained shortest path P_{qsp} is selected. P_{qsp} is the smallest congestion increase

path, while satisfying the delay constraint of $D_{uv} + \Delta_{uv}$. In line 8, D is the overall delay computed in the same way with 2. In line 10, *Get-Path*(K, i) imposes CC for each QCM in P_{qsp}. If there is no such path in K, in line 14, the shortest path computed by Dijkstra's algorithm is set to P_{qsp}.

The time complexity of Algorithm 1 is $O(kN(M + N \log N))$, where $N = |V(G)|$, $M = |E(G)|$, and k is the number of paths in k-shortest-path algorithm in line 5 [17]. k-shortest-path algorithm is the dominant function in Algorithm 1. Its time complexity is $O(kN(M + N \log N))$ [17]. Thus, the time complexity is $O(kN(M + N \log N))$.

V. Research Issues

We have the following research issues for drones in drone networks.

1) We should consider the environment factors such as wind and terrain (e.g., mountain, obstacle, infrastructure, resident area, etc.) which may affect the travel time of drones.
2) In this paper, we consider the charging time as constant. We can enhance our path planning scheme as we consider the charging time as a variable.
3) For gathering and disseminating statistics between drones and TCC, we need to consider the wireless networks through IEEE 802.11p or IEEE 802.11a.
4) Besides the queueing at QCMs, closely-packed drones may collide each other. We need to consider a control system to avoid collision between drones.
5) We need to optimize the congestion contribution curve that considers a drone's travel delay, waiting delay, and charging delay.

VI. Conclusion

In this paper, we proposed our scheduling algorithm to assign appropriate QCMs to drones. Our algorithm aims at reducing overall travel delay for drones at QCMs. Our algorithm is suggested based on [4], [17]. We believe that our algorithm will contribute to commercialize drones. As future work, we will evaluate the performance of our algorithm by comparing ours with other baseline algorithms through simulation, and enhance our drone battery charge algorithm.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2014006438). This research was supported in part by MSIP/IITP [R0166-15-1041, Standard Development of Network Security based SDN], and MSIP/IITP [B0101-15-0663, Safety-critical Distributed Modular SW Platform]. Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] R. Austin, *UNMANNED AIRCRAFT SYSTEMS*. John Wiley and Sons Ltd, 2010.

[2] T. Toksoz, J. Redding, M. Michini, B. Michini, J. P. How, M. A. Vavrina, and J. Vian, "Automated Battery Swap and Recharge to Enable Persitent UAV Missions," in *American Institute of Aeronautics and Astronautics*, 2011.

[3] Korea Internet & Security Agency, "Power Review," *KISA Report*, May 2015.

[4] J. P. Jeong, H. Jeong, E. L. an Tae (Tom) Oh, and D. H. Du, "SAINT: Self-Adaptive Interactive Navigation Tool for Cloud-Based Vehicular Traffic Optimization," *IEEE Transactions on Vehicular Technology*, Sep. 2015.

[5] Birk, Andreas, et al, "Safety, security, and rescue missions with an Unmanned Aerial Vehicle (UAV)," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 1, pp. 57–76, 2011.

[6] Bristeau, Pierre-Jean, et al, "The navigation and control technology inside the ar. drone micro uav," in *18th IFAC world congress*, vol. 18, no. 1, 2011.

[7] Parrot, "Parrot," http://www.parrot.com.

[8] Nixie, "Nixie," http://flynixie.com.

[9] Dji, "Dji," https://store.dji.com.

[10] Parrot, "Parrot," http://www.airinov.fr.

[11] Saska, Martin, et al, "Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation," in *Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[12] J. Irizarry, M. Gheisari, and B. N. Walker, "USABILITY ASSESSMENT OF DRONE TECHNOLOGY AS SAFETY INSPECTION TOOLS," *ITCON*, vol. 17, no. 6, Sep. 2012.

[13] Katsuya Fujii and Keita Higuchi and Jun Rekimoto, "Endless Flyer: A Continuous Flying Drone with Automatic Battery Replacement," in *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*. IEEE, 2013.

[14] Suzuki and Koji AO and Paulo Kemper Filho and James R. Morrison, "Automatic battery replacement system for UAVs: Analysis and design," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 563–586, 2012.

[15] Wu, Zhiqiang, Hemanth Kumar, and Asad Davari, "Performance evaluation of OFDM transmission in UAV wireless communication," in *System Theory, 2005. SSST'05. Proceedings of the Thirty-Seventh Southeastern Symposium on. IEEE*. IEEE, 2005.

[16] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," *RFC6275*, Jul. 2011.

[17] J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network," *Management Science*, vol. 17, no. 11, pp. 712–716, Jul. 1971.