Computers and Electrical Engineering 75 (2019) 77–89

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Dynamic self-reconfiguration of unmanned aerial vehicles to serve overloaded hotspot cells☆,☆☆

Amar Nath Patra*, Paulo Alexandre Regis, Shamik Sengupta

Department of Computer Science and Engineering, University of Nevada, Reno, USA

ARTICLE INFO

Article history:
Received 1 September 2018
Revised 13 February 2019
Accepted 13 February 2019
Available online 21 February 2019

Keywords:
Unmanned Aerial Vehicles
Hotspot cell
Hot zone
Delaunay triangulation
Wireless coverage
Bandwidth

ABSTRACT

Unmanned Aerial Vehicle (UAV) networks can provide wireless coverage to users when cellular infrastructure is unavailable. Each UAV covers a circular area, called a *hotspot cell*, and serves the users in it. Natural disasters can cause unpredictable mobility of users, randomly overloading some cells, called *hot zones*, for different time intervals. Along with an efficient initial deployment, this paper presents a two-fold approach to handle hot zones. First, the loads in overlapped sections are redistributed to neighbors. Second, a distributed algorithm is run that dynamically repositions the least number of UAVs to serve the maximum possible users. Additionally, to balance the user demand and supply of wireless coverage, adjacent UAVs are allowed to swap their positions. Simulation results show improvement in the network performance using the parameters: packet delivery ratio, data served per unit of depleted energy, service discontinuity time, and the total amount of data served.

1. Introduction

A prominent application of Unmanned Aerial Vehicle (UAV) networks is in providing wireless coverage to ground users. Offloading ground base stations is one of the implemented approaches to harvest the UAVs' potential in providing wireless services [2]. However, due to the impracticability of ground conditions or time constraints, these base stations cannot be established. In such scenarios, the users can be served by a swarm of UAVs that can be deployed on-the-fly [3].

In 2011, the US government considered the deployment of a drone network (as an extemporaneous communication system) in emergency situations to be promising [4]. The National Public Safety Telecommunications Council started investigating drone based communication four years later [5]. Virtual Network Communications [6], a startup, developed a scalable LTE (Long-Term Evolution) base station called a *Green Cell*. It contained a credit-card-size component employing LTE technology to form an ad hoc network with neighboring radios, subsequently connecting to a nationwide cellular network. This can support 128 users at a time on any LTE frequency. The Green Cell along with its battery weighed only 2 kg, light enough for a drone to carry. The authors in [7] proposed a drone-mounted base station mechanism by mounting LTE femtocells on drones to offer an alternative for the overloaded existing wireless infrastructure. The mechanism calculates the required number of drones and their optimal locations to maximize user coverage.

☆ A preliminary version of this work was presented in IEEE SPECTS 2017 [1].

☆☆ This paper is for CAEE special section SI-fadn. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. C. A. Kerrache.

* Corresponding author.

E-mail addresses: apatra@nevada.unr.edu (A.N. Patra), pregis@nevada.unr.edu (P.A. Regis), ssengupta@unr.edu (S. Sengupta).

https://doi.org/10.1016/j.compeleceng.2019.02.014

Nevertheless, initial deployment of these UAV networks is not sufficient to provide efficient coverage. The users' mobility causes non-uniform densities at various locations at different instants, creating overloaded hotspot cells, called *hot zones*. Limitations in the available UAV count, UAVs' battery life constraint, and the requirement of an uninterrupted service to users call for a judicious load distribution, and repositioning of these UAVs. When an UAV moves from its current position, some or all of its users may remain unattended until another UAV comes to serve them. These challenges overshadow the unparalleled capability of UAV networks in providing wireless coverage.

Considering the above problems, this paper's objective is to present a UAV deployment scheme to serve the maximum possible number of users by minimizing the service interruption time. An initial deployment for an efficient layout of UAV positions is presented here. The paper then highlights the effect of user mobility and irregular density (causing hot zones), in disabling the affected UAVs to serve all of their respective users. The proposed solution allows these UAVs to prudently choose their peers so that they can dynamically reposition themselves to maximize the users served. The algorithms are presented for: (1) redistributing loads (user demand) in shared areas of adjacent cells; (2) the dynamical reconfiguration of the network topology based on overloaded cells; and (3) swapping the positions of UAVs with their 1-hop neighbors.

The rest of the paper is organized as follows: Section 2 showcases recent research in this area. Section 3 explains the proposed mechanism of the initial deployment of UAVs, the three algorithms discussed before and the various assumptions that are taken. Section 4 explains the simulation environment, parameters, and results. Finally, Section 5 concludes the paper.

2. Related work

In [8], the authors propose a placement scheme to minimize the number of UAV mounted mobile base stations for providing wireless coverage to ground terminals. To position the UAVs sequentially in a spiral, starting from the boundary of the area and finally reaching the center, it takes a polynomial time. In [9], the authors present an algorithm based on electrostatic forces for the deployment of UAVs in 3D space. The authors in [3] discuss two problems of optimal deployment: minimizing the maximum deployment delay among all UAVs (min-max) for fairness consideration, and minimizing the total deployment delay (min-sum) for efficiency consideration. They consider UAV heterogeneity in flying speed, coverage radius and operating altitude.

The authors [10] propose an energy-efficient 3D placement of UAV mounted base stations involving the onboard power consumption and considers varying users' density. A proactive UAV deployment to reduce overload conditions caused by flash crowd traffic in 5G networks is presented by the authors in [11]. In this paper, three kinds of flash crowd traffic are developed along with a hybrid distribution. This is followed by prediction and operating control schemes for the deployment. The authors in [12] propose a Hovering Ad-Hoc Network to solve the problem of irregular capacity demand, generated because of the mass movement of users. In [13], the authors present two distributed algorithms for dynamic drone repositioning. In the first, a drone makes movement decisions based on only the information of its current users' positions, while the second algorithm requires information about user locations in neighbor cells as well to maximize the spectral efficiency of the complete network.

The aforementioned studies do not evaluate the actual new position of the moving UAVs and the associated change in the topology, at different instants of the deployment time, to serve maximum possible users. In our previous work [1], a distributed algorithm was proposed where the UAVs assisted their immediate neighbors in serving hot zones. The algorithm did not consider multi-hop peers and, moreover, always required the UAVs to move to share the load in hot zones. Hence, this paper proposes a comprehensive solution, executing various components in a stepwise manner to mitigate the problem of hot zones. Load redistribution in overlapped sections, multi-hop away UAV assistance, and swapping of the UAV locations (independent of hot zones) are the three prominent components that were not previously considered.

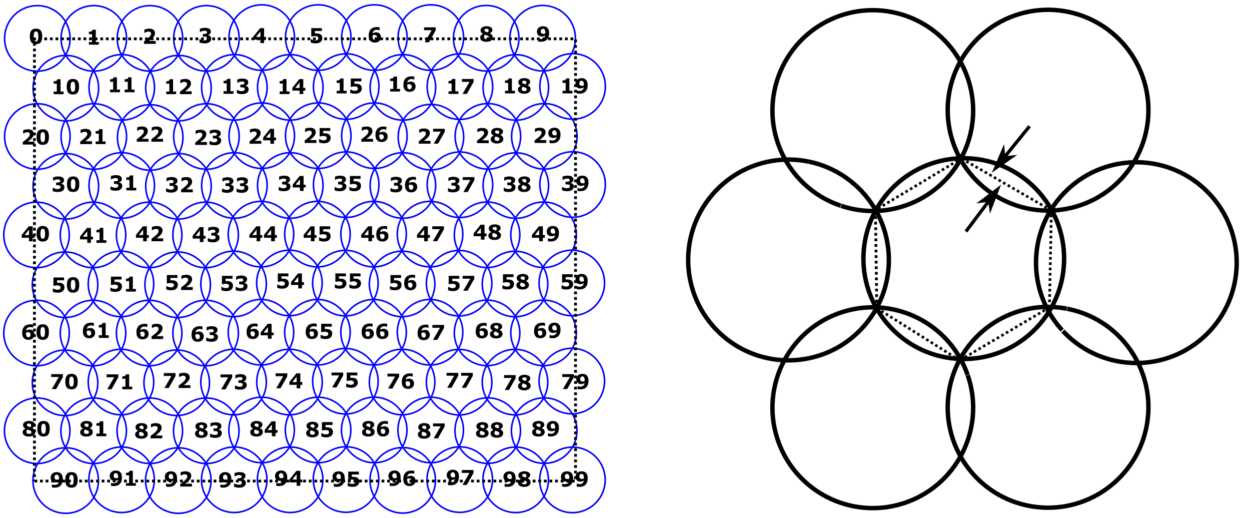
3. Proposed methodology

The UAVs should be efficiently deployed to provide seamless wireless coverage to the maximum possible number of users. The users move in the covered region, forming variable densities in different subareas. The user density in a sub-area may also change with time. The challenge is to serve as many users as possible without over-consuming energy (spending more energy in hovering or flying than in serving users). To this end, the proposed method strives to increase the average number of users served per unit of energy consumed by a UAV.

The proposed method first deploys UAVs to efficiently cover the entire region. Consequently, UAVs redistribute their loads to serve users whose mobility and bandwidth requirements are unpredictable. Later, if required, they move to new positions.

3.1. Initial deployment

The initial deployment utilizes Delaunay Triangulation with equilateral triangles. This technique offers an efficient coverage of the region by maximizing the area covered while minimizing (the area of) overlapped sections between adjacent hotspot cells, and ensuring that there are no gaps between them [1,14]. Furthermore, the greedy approach is adopted to ensure complete coverage of a given region with any shape [15]. Fig. 1(a) shows a deployment of 100 UAVs in a rectangular region represented by a dotted perimeter.



(a) Example deployment (100 UAVs)

(b) Load sharing

Fig. 1. Initial deployment and load sharing in overlapped sections.

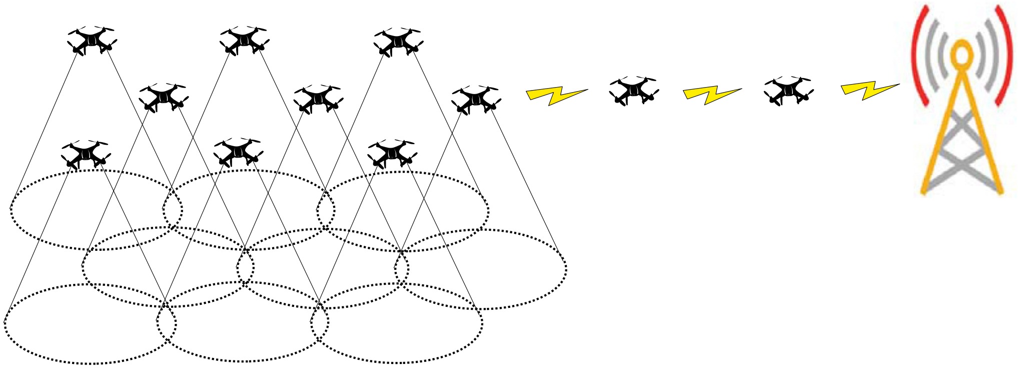


Fig. 2. UAV deployment and internet connectivity.

Each cell circumscribes a regular hexagon (Fig. 1(b)). The overlapping cells form a framework of non-intersecting hexagons that share their sides with neighboring hexagons. The objective is to determine the minimum UAV count required to cover the entire region. The total area covered by all the hexagons can be computed to determine the required UAV count. Let $Area_{Hex}$ represent the area of a hexagon, given by $\frac{3\sqrt{(3)}R^2}{2}$, where R is the radius of the cell circumscribing it. $Area_{Region}$ denotes the area of the region. Then, the minimum required UAV count, N, for covering the region is given by

$$N = \arg\min(N.Area_{Hex} - Area_{Region}) \tag{1}$$

$$\text{st:} (N.Area_{Hex} - Area_{Region}) >= 0 \tag{2}$$

3.2. Assumptions

The following assumptions are made to streamline the proposed method.

i) All the UAVs cover the same area and fly at the same altitude. The bandwidth requirement of all users are identical and fixed.
ii) The UAVs called *Relays*, connect the deployed UAVs to a LTE BTS (Base Transceiver Station) tower, providing an Internet connection (Fig. 2 shows two *Relays*). (UAV-UAV communication is not shown for the sake of simplicity.)
iii) In the beginning of the deployment, there are N number of UAVs (excluding Relays) that decreases over time due to energy consumption.

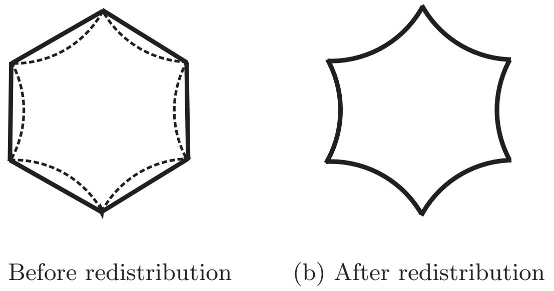

(a) Before redistribution (b) After redistribution

Fig. 3. Load redistribution in overlapped sections.

iv) The UAVs divide the users in the overlapped section of adjacent cells by forming hexagonal cells. Fig. 1(b) shows a circular cell in the center, taking the shape of a (dotted) hexagon after sharing the loads with its neighbors. The arrows signify the load distribution among the UAV serving the central cell and one of its neighbors.
v) Each UAV determines its current location using GPS (Global Positioning System) and updates the neighborhood information in its neighbor table after every hello timer.
vi) The user density within a cell always follows uniform distribution.

3.3. *Dynamic self reconfiguration of UAVs to serve overloaded hotspot cells*

An overloaded UAV redistributes its load in overlapping sections. Later, it may request for one or more peers (to move closer to share its load).

3.3.1. *Phase 1: Load redistribution in overlapped sections*

The users may enter or leave the cell of a serving UAV (or even the entire region), thereby making the user density and mobility dynamic. The increase in user density in a cell may render the UAV incapable of serving all of its users. Contracting the cell in such a situation, reduces the number of users and may help the UAV in meeting the users' requirements.

Expansion and contraction of cells in asymmetric traffic have been discussed in past research [16]. A decentralized cell-site selection algorithm for the users in overlapped sections of adjacent cells is proposed in [17]. The users are assigned one of two cell-sites depending on their traffic levels. A dynamic load-sharing method based on a reconfigurable overlapped cellular coverage is analyzed in [18]. This work focuses on localized congestion (caused by accidents and roadwork) that contributes to asymmetrical teletraffic load situations.

An UAV, U_i, periodically computes its current load (user count), l_i. The load capacity, c_i, is the number of users it can serve. If $l_i > c_i$, then, U_i cannot serve all of its users, which means a hot zone is created in its cell.

U_i successively releases its loads in the overlapped sections to its 1-hop neighbors. In the initial deployment, an UAV has a maximum of six neighbors, sharing one overlapped section with each. Suppose U_i is the central UAV in Fig. 1(b). It starts handing over its load in these sections to its neighbors, considering them individually. While releasing the load, it ensures that no hot zone is created in each considered neighbor's cell. Thus, prior to handing over the loads, U_i has a hexagonal coverage area as shown in Fig. 3(a). If it succeeds in giving away its load in all of the overlapped sections, its coverage area reduces to the shape as shown in Fig. 3(b).

Because the user density in a cell is assumed to be uniform, a reduction in the coverage area implies a lower load. The area of the overlapped section, *ovlp*, is determined by the following Eq. (3), where d_{ij} is the distance between U_i and U_j, and R is the radius of U_i's cell [19].

$$ovlp = \frac{2R^2 \arccos(\frac{d_{ij}}{2R}) - \frac{d_{ij}}{2}\sqrt{4R^2 - d_{ij}^2}}{\pi R^2} \tag{3}$$

Therefore, the cell's area can be mapped to its user count. Eq. (4) gives the number of users (load) per unit area.

$$l_{i_{unitArea}} = \frac{l_i}{\pi R^2} \tag{4}$$

$handOver_i$ deontes the amount of load that U_i can release to a neighbor U_j. U_i uses Eq. (5) to show the relationship between its cell area and its current load.

$$handOver_i = \frac{l_i}{\pi R^2} \cdot \frac{ovlp}{2} \tag{5}$$

Algorithm 1 shows the steps of redistribution that are performed.

Algorithm 1: Load redistribution.

```
Input: Current load of $U_i$
Output: Reduced load of $U_i$ after redistribution
1  while For each 1-hop neighbor $U_j$ do
2      if $l_j$ + $handOver_i$ < $c_j$ then
3          $l_i$ = $l_i$ - $handOver_i$;
4          $l_j$ = $l_j$ + $handOver_i$;
5          exit;
```

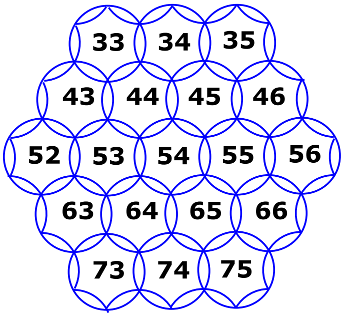


Fig. 4. Example figure.

3.3.2. Phase 2: dynamic repositioning of UAVs

After handing over all possible loads to its 1-hop neighbors (Phase 1), U_i recomputes its current load, l_i. If l_i is still greater than c_i, then U_i requires assistance from its peers to share its load (by moving closer) so that all of its users can be served. It computes its excess load, l_i^{ex}, which is equal to $l_i - c_i$.

The following example explains the process of selecting suitable peers and computing their new locations. Suppose U_{54} in a deployment (Fig. 4) observes that its cell has become a hot zone. It is assumed that after redistribution of load, the number of users in its cell are 165, and it can serve a maximum of 100 users. This means l_{54}^{ex} is 65 (165–100). It becomes the requester UAV, U_{req}, and starts executing Algorithm 2 by broadcasting a 1-hop *Req_Msg* (hcnt = 1) to advertise its excess

Algorithm 2: Dynamic repositioning.

```
Input: All UAVs (excluding Relays), Assumption: $l_i > c_i$ for $U_i$
Output: Change in UAV network topology
1  $U_i$ sets hop count, hcnt = 1 and excess load, $l_i^{ex}$ = $l_i$ - $c_i$ in Req_Msg;
2  $U_i$ broadcasts Req_Msg with hop count = hcnt, advertising the excess load, $l_i^{ex}$;
3  A hcnt-hop neighbor $U_j$ responds with a Resp_Msg only if $l_i^{ex} > l_j$;
4  if hcnt < hop-limit then
5      if $U_i$ does not receive any response within hcnt hello timers then
6          it increments hcnt by 1 and goes back to Step 2;
7  else
8      exit.
9  $U_i$ chooses $U_{aide}$ base on highest remaining energy, calculates its new position, and broadcasts an Ack_Msg (hop count
   = hcnt);
10 $U_{aide}$ broadcasts Req_Msg to its 1-hop neighbors advertising the excess load, $l_{U_{aide}}^{ex}$ = $l_{U_{aide}}$;
11 if $U_{aide}$ receives a Resp_Msg within one hello timer then
12     it selects the neighbor with highest remaining energy;
13 $U_{aide}$ proceeds and acquires the new position;
14 After collocation and sharing of load, $U_i$ recomputes excess load, $l_i^{ex}$ = ($c_i$ + $c_{U_{aide}}$) - ($l_i$ + $l_{U_{aide}}$);
15 if $l_i^{ex}$ > 0 then
16     $U_i$ reexecutes Algorithm 2 by going to step 1;
17 exit.
```

Table 1
Current loads and remaining energy of 1-hop neighbors of U_{54} and U_{44}.

U_i	l_i	**Energy**(10^3 **J**)
44	48	920
45	70	680
53	79	580
55	23	870
64	52	790
65	82	950
33	22	570
34	85	630
43	34	710
45	78	290
53	93	880

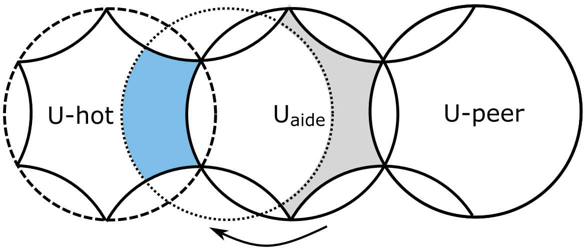


Fig. 5. Case 1: movement of 1-hop away U_{aide}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

load, l_{54}^{ex}. On receiving the message, U_{44}, U_{55} and U_{64} respond with a *Resp_Msg*, as these are the only 1-hop neighbors with current loads less than l_{54}^{ex}, as shown in Table 1. These responding UAVs form a set, $RSet_{hcnt}$, as defined in Eq. (6).

$$RSet_{hcnt} = \{U_j | U_j \text{ is a} hcnt - \text{hop neighbor} \wedge l_i^{ex} > l_j\} \tag{6}$$

$$U_{aide} = \arg\max_{j \in RSet_{hcnt}} e_j \tag{7}$$

On receiving this message, U_{54} chooses U_{44} to be its U_{aide} because U_{44} has the highest remaining energy as compared to the other two prospective peers ($e_{44} > e_{55} > e_{64}$), following Eq. (7). U_{54} broadcasts a 1-hop *Ack_Msg* to notify its 1-hop neighbors about the chosen U_{aide} (U_{44}) and its new location, computed using Eq. (8) [19]. *ovlp* is the required overlap between U_{54} and U_{44} so that the latter can take over the load, l_{54}^{ex}, from U_{54}. *ovlp* is computed based on l_{54} (current load of U_{54}), using Eqs. (3) and (4).

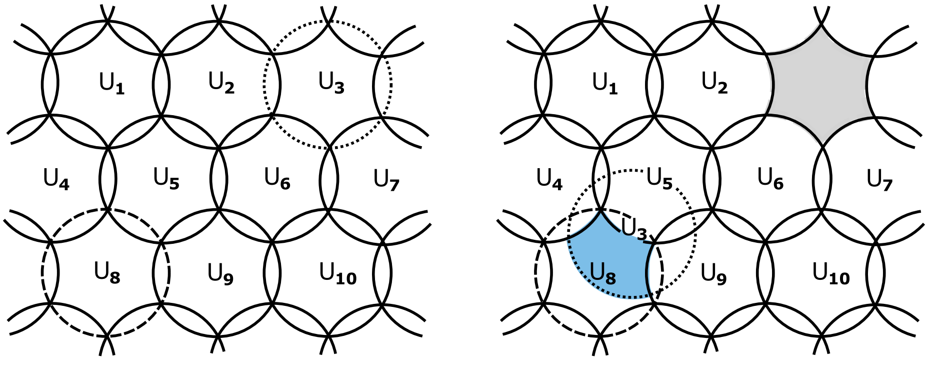
$$ovlp.\pi R^2 - 2R^2 \arccos(\frac{d_{ij}}{2R}) - \frac{d_{ij}}{2}\sqrt{4R^2 - d_{ij}^2} = 0 \tag{8}$$

Movement of U_{44} from its original location will result in a pseudo-hot zone as its users will not be served due to its relocation. Hence, U_{44} redistributes its load, and then starts its own Algorithm 2 by broadcasting a *Req_Msg* advertising l_{44}^{ex} (l_{44} minus the load represented by the area that it will overlap after movement). If the load represented by this overlapped section is 8; thus, l_{44}^{ex} will be set as 40 (48 - 8). It waits for one hello timer to receive any *Resp_Msg*. If the message is received, it chooses a peer with the highest remaining energy as its U_{aide} and broadcasts a 1-hop *Ack_Msg* in a similar fashion, and then moves to its new location. It acquires the new position even when no responses are received, as its primary goal is to assist the U_{req}.

Based on the information provided in Table 1, U_{44} receives responses from U_{33} and U_{43}, and selects the latter. This chained process continues until no suitable 1-hop neighbor is available. However, if the original U_{req}, U_{54}, does not receive any response in the first cycle, then it increases the hop count by one, and rebroadcasts the *Req_Msg*, ensuring the advertisement is received by 2-hop neighbors. The hop count is incremented by 1 until at least one response is obtained in a current cycle, or all the UAVs in the network receive the *Req_Msg*.

In the above example, the U_{aide}, U_{44} was a 1-hop neighbor of the U_{req}. However, a U_{aide} may not always be 1-hop away. Hence, there are two cases.

Case 1 (1-hop away U_{aide}): Fig. 5 illustrates an example of this context. The area in blue represents the required overlap, whereas the area in gray represents the users who will no longer be served by the U_{aide} (unless another UAV is found). Before acquiring the new location, U_{aide} broadcasts a 1-hop *Req_Msg* advertising its current load. If the 1-hop neighbors do



(a) Before movement

(b) After movement

Fig. 6. Case 2: movement of multi-hop away U_{aide}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

not receive an *Ack_Msg* from the U_{aide} in a set time, they will assume that the U_{aide} could not find a peer to take its place. Thus, they broadcast the *Req_Msg* that they had received from the U_{aide} to their respective 1-hop neighbors. This process continues until all the UAVs receive the *Req_Msg*, or a suitable peer is found. If no UAV is found, then the users in the gray area will remain unserved until U_{aide} comes back to its initial location.

Case 2 (Multi-hop away U_{aide}): This scenario is portrayed in Fig. 6(a). Here, the gray area is much larger as compared to that in Case 1, because the movement of the U_{aide} will not result in overlapping its existing cell. Fig. 6(b) shows the effect of movement of the U_{aide}, U_3, from its original location to the new one, depicting the gray and blue areas. As in Case 1, finding the subsequent neighbors also follows the steps as enumerated in Algorithm 2. The difference between the two cases is that now U_{aide} shares the loads with some of the neighboring UAVs of U_{req}; U_3 now shares the load of U_5 and U_9 in the corresponding overlapped sections.

If a U_{req} does not receive any *Resp_Msg* from its peers, it increments the hop count by 1 in the *Req_Msg* until hop count $<$ *hop-limit*. If hop count equals *hop-limit*, U_{req} stops broadcasting any *Req_Msgs*. The *hop-limit* is set based on the radius of the UAV network. Once U_{req} no longer requires assistance from the U_{aide} to serve its cell, it notifies the latter (through *Hello Message*) to retreat to its original position. In a recursive process, the subsequent neighbors also retreat to their previous positions.

3.4. Swapping of UAVs

An UAV, U_i, calculates its *time-to-die*, T_i, based on its current consumption rate, puts this data in its *Hello Message*, and broadcasts to its 1-hop neighbors. If the received *Hello Message* from U_j gives a difference between T_i and T_j of more than τ time units, then U_i swaps its position with U_j (Algorithm 3).

Algorithm 3: Swapping.

```
Input: Current locations of $U_i$ and $U_j$
Output: Swapped locations of $U_i$ and $U_j$
while For all pairs of 1-hop neighbors, $U_i$ and $U_j$ do
    if $T_i$ - $T_j$ >= $\tau$ time units then
        $U_i$ swaps position with $U_j$;
        exit.
```

During the swapping process, some of the users in the cells of both UAVs will not be served for a small duration of time. Swapping of positions between two arbitrary 1-hop neighbors, U_{Blue} and U_{Green} is shown in Fig. 7. Here, it is assumed that each one requires 4 s to take the other's position. The trajectories of each of the UAVs are shown along with their corresponding coverage area movements. The new coverage areas in the first three seconds are shown with dotted circles (U_{Blue}) and dashed circles (U_{Green}) for the corresponding UAVs. In the fourth (final) second, the new coverage areas completely exchange each other's positions.

Fig. 8 highlights the effect on the original coverage area of U_{Blue}. After 1 s, U_{Blue} loses some of its users, shown in the gray shaded area (unattended users). During this time, U_{Green} could cover a very small area that U_{Blue} has already lost, as shown in Fig. 8(a). After 2 s, U_{Green} could substantially cover the lost area of U_{Blue}; however, the unattended area is greater than in Fig. 8(a) (depicted by the gray area in Fig. 8(b)).

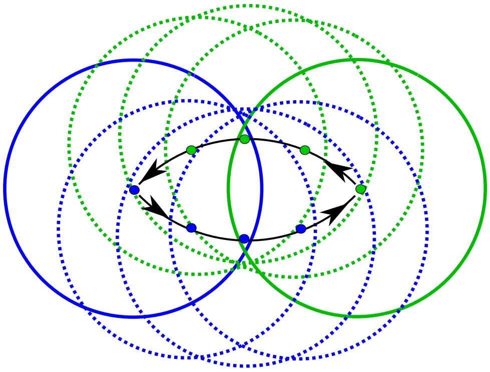
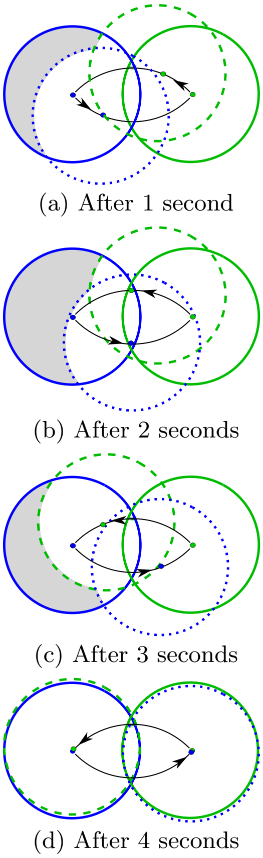


Fig. 7. Swap between U_{Blue} and U_{Green}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) After 1 second

(b) After 2 seconds

(c) After 3 seconds

(d) After 4 seconds

Fig. 8. Unattended (shaded) area of U_{Blue} while swapping with U_{Green}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Eventually, after 3 s, Fig. 8(c), U_{Green} covers an even larger piece of the U_{Blue}'s original coverage area, leaving a smaller unattended section, shown by the gray shaded portion. This area is approximately equal to that depicted in Fig. 8(a) (first second). Finally, after 4 s (Fig. 8(d)), both UAVs exchange each other's original positions; hence, completing the swap. Furthermore, U_{Green} completely acquires the coverage area of U_{Blue} and there is no unattended section, which means all of the original users of U_{Blue} are now being served by U_{Green}. As it is assumed that all the UAVs cover the same area, there will be a similar effect on U_{Green}'s coverage area.

Although swapping can improve the overall performance and increase the lifetime of the network, highly frequent swaps throughout the network will, in fact, increase the overall unserved time along with the number of affected users. This in

Table 2
Simulation parameters.

Parameter	Value
Coverage radius of each UAV	100 m
Initial distance between any two adjacent UAVs	$\sqrt{3} \times 100$ m
Area of region	2×2 km^2
Results accumulative interval	600 s
Tx current	8A
Idle/Rx current	6.5A
Communication range	210 m
Constant error rate	0.01
Initial energy source capacity	12000 J
User data requirement	1000 B
Hello Interval (OLSR)	2 s

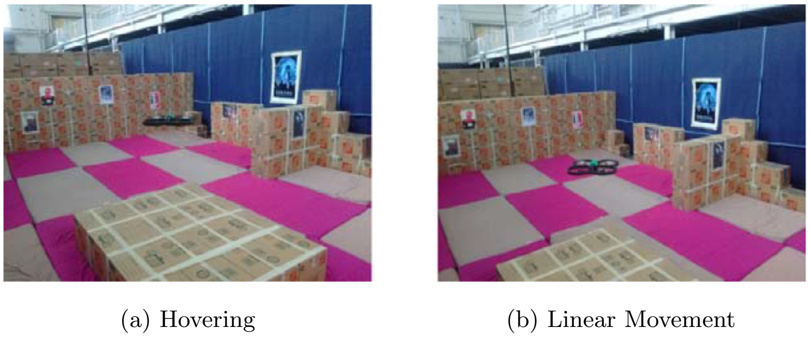

(a) Hovering (b) Linear Movement

Fig. 9. Laboratory experiments. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

turn will decrease the total amount of data transmitted by all the UAVs. Hence, the parameter τ, as described above, should be chosen carefully.

4. Performance evaluation and results

To validate the proposed approach, a simple scenario was simulated using ns-3 [20]. 100 UAVs were initially deployed through Delaunay triangulation with equilateral triangles as shown in Fig. 1(a). The UAV ad hoc network was distributed at the same height, with all traffic directed to an abstract *sink* node which was placed in the center of the scenario, 100 m above the UAV plane. The *sink* did not contribute to the measured results. All nodes, including the *sink*, were part of the same mesh network. This setup was chosen due to its simplicity. A more advanced architecture could potentially improve the mechanism even further; however, the architecture itself is beyond the scope of this paper.

It was assumed that each UAV could serve a maximum of 50 users and would experience a hot zone when there were more than 50 users in its cell. A threshold of 5 users was applied as a trigger for load redistribution. This means that an UAV would start the process of load redistribution in its cell when it had more than 55 users. 1 to 10 hot zones were dynamically created, with a uniform distribution, to account for the unpredictability of user mobility. Further, these hot zones were created for random durations following a uniform distribution.

A range-based model for the network communication channel was used (if two nodes were within communication range, then there was a link between them). The transmission data rate was fixed to 1Mbps. Each UAV aggregated the entire traffic generated by its served users and forwarded it to the *sink* either directly or relaying through other nodes. Optimized Link-State Routing (OLSR) protocol was used to enable the multi-hop relay aspect of the architecture. The simulation time was set for 2 h based on the recent achievement in flight time [21]. Most of the simulation parameters were based on [22] and can be seen in Table 2.

The energy model assumed that the node could be in two states: *transmitting* or *idle/receiving*. When transmitting, the energy depletion is greater than when receiving. It was assumed that the UAVs consume an equal amount of energy when flying horizontally or when hovering over a fixed position. This assumption was based on the laboratory experiments as shown in Fig. 9. To find out the energy consumed in providing services to users, Raspberry Pi were used as clients to the drones receiving data at a rate of 1Mbps.

The UAVs can carry a weight of around 2 kgs to be able to provide wireless coverage (discussed in Section 1). Amazon intends to fly its drones carrying a payload of the same weight at a speed of 22.352 m/s [23]. Hence, the flying speed of 20 m/s was chosen for the UAVs in the simulations. Moreover, the current record of the fastest drone is above the rate of 70 m per second [24]. Thus, in future applications of the proposed method, higher flying speed for UAVs can be set, which will further improve the network performance.

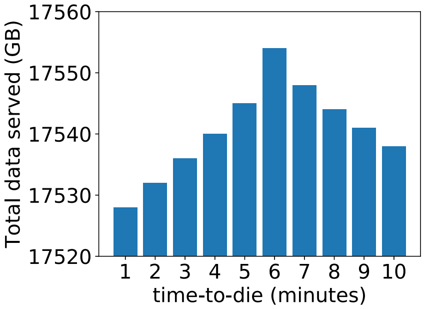


Fig. 10. Total data served with varying time-to-die time units.

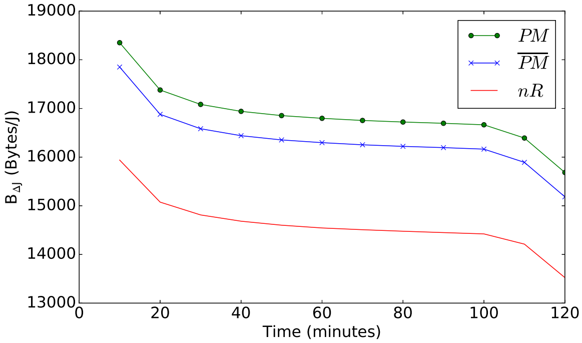


Fig. 11. Data served per unit of depleted energy.

Keeping the above-mentioned parameters constant, several simulation repetitions were run with different τ time units. The total amount of data served by the UAV network with varying *time-to-die* time units are shown in Fig. 10. It can be seen that the total data increased when τ was increased from 1 to 6 min. This is because a highly frequent swapping operation (lower τ) has a detrimental effect on the UAV networks' capability to serve the users, as discussed in Section 3.4. However, the total data decreased by further increasing τ as the UAVs were not prompt enough to balance the user demand and wireless supply in the network. The optimal value of τ was observed to be 6 min. Thus, in the simulations, two adjacent UAVs swapped their positions when their *time-to-die* values differed by at least 6 min.

One hundred independent simulation repetitions were run for each case: (1) Proposed Method (*PM*) (includes swap); (2) Proposed Method without swap ($\overline{PM}$); and (3) no Reconfiguration (*nR*). Case (1) and case (2) show the effect of swapping (which was not considered in the previous work [1]) on the results when proposed methods were used. In case (3), the UAVs do not move or redistribute the load among themselves. However, the initial deployment of UAVs for all the three cases were identical. Results were averaged from the simulation repetitions for each individual case.

A parameter, $B_{\Delta J}$, is used to compare the three cases in terms of the quantity of energy consumed in serving data and in hovering/flying. $B_{\Delta J}$ is the total data served to all users per unit of energy depletion (considering all the UAVs), and computed after every 10 min, as given in Eq. (9).

$$B_{\Delta J} = \frac{\sum_{i=1}^{\hat{M}} data_{u_i}}{\sum_{j=1}^{\hat{N}} \Delta J_{U_j}} \tag{9}$$

where, $data_{u_i}$ is the data served to user, u_i, ΔJ_{U_j} is the quantity of depleted energy at UAV, U_j, $\hat{N}$ is the total number of active UAVs during the calculation of ΔJ_{U_j} and $\hat{M}$ is the user count served by $\hat{N}$. $data_{u_i}$ and ΔJ_{U_j} are calculated every 100 ms and accumulated throughout the simulation time.

Fig. 11 shows $B_{\Delta J}$ calculated after every 10 min for the three cases. The proposed methods (case 1 and case 2) outperform the *nR* case (case 3) throughout the simulation time. As they focus on efficiently utilizing the UAVs' energy while serving the users, the UAVs are able to better handle the hot zones in the network. This efficacy is achieved by prudent load redistributions, followed by choosing of suitable neighbors to share the load and repositioning of fewer UAVs to serve hot zones. By adding the swapping component, the energy consumption of the UAVs in serving users is further improved, than in hovering. Periodic swap ensures a balance between the user requirement and availability of UAVs.

Fig. 12 compares the Packet Delivery Ratio (PDR) in the three cases. The proposed methods have higher PDR than the *nR* case after around 10 min of simulation time. Efficient load redistribution and network topology changes triggered by hot zones, allow the UAVs to serve users in hot zones better as compared to a network without reconfigurations. Adding the swapping component to the proposed method further increases the PDR.

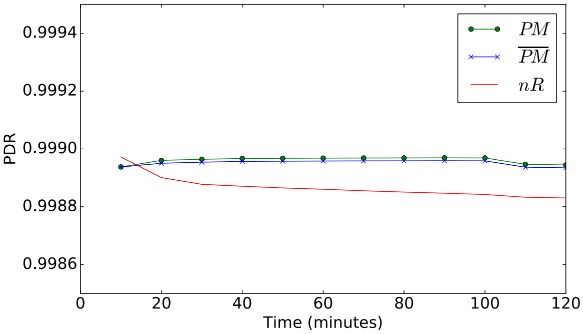


Fig. 12. Packet delivery ratio.

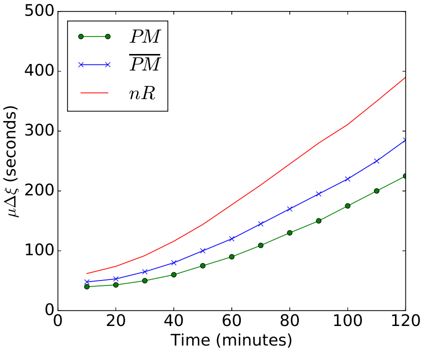


Fig. 13. Average service discontinuity time.

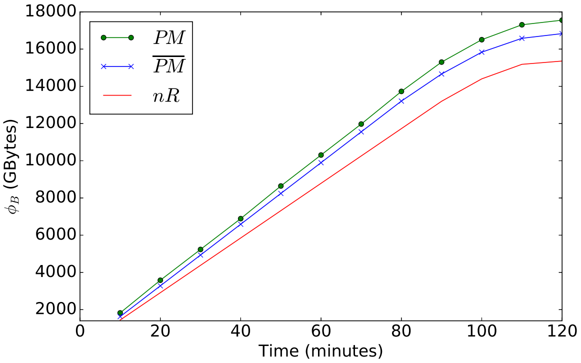


Fig. 14. Total data served.

The users are affected in hot zones because the serving UAVs cannot meet the requirements. The proposed methods allow the UAVs to redistribute their loads effectively so that they can better serve the users. The average service discontinuity time observed by users, $\mu\Delta\xi$ (Eq. (10)), was added up until the end of the simulation. Fig. 13 compares the three cases. The users encountered less interruption in receiving data from the UAV network when the proposed methods were used throughout the simulation time. The plots are exponential since the service discontinuity time is summed up from the beginning until the end of the simulations.

$$\mu\Delta\xi^{t_k} = \sum_{t=1}^{t_k} \frac{\sum_{i=1}^{M^{t_k}} \xi_{u_i}^{t_k}}{M^{t_k}} \tag{10}$$

where, M_{t_k} is the user count served by the UAV network at time t_k and $\xi_{u_i}^{t_k}$ is the interruption check for the user, u_i, at time t_k ($\xi_{u_i}^{t_k} = 0$ if u_i is served at t_k, 1, otherwise).

Similarly, the total data served by all the UAVs, ϕ_B (Eq. (11)), is added up too throughout the simulation. ϕ_B is higher when the proposed methods are followed as compared to the *nR* case (Fig. 14). In due course of time, the UAVs consumed their energy. After around 100 min of simulations, most of the UAVs had completely exhausted their batteries, thus, leaving

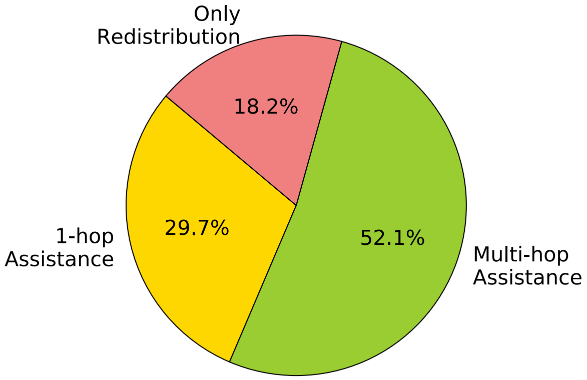


Fig. 15. Percentages of Only Redistribution, 1-hop assistance, and Multi-hop assistance.

Table 3
Confidence intervals.

Parameters	Cases	600 s	3600 s	7200 s
$B_{\Delta J}$ (B/J)	PM	18,351 ± 163	16,797 ± 129	15,684 ± 182
	$\overline{PM}$	17,827 ± 116	16,485 ± 147	15,165 ± 215
	nR	15,939 ± 152	14,544 ± 229	13,524 ± 186
$\mu\Delta\xi$ (s)	PM	40 ± 0.43	90 ± 0.85	225 ± 3.38
	$\overline{PM}$	48 ± 0.35	127 ± 2.23	284 ± 4.61
	nR	62 ± 0.59	177 ± 5.37	390 ± 9.64
ϕ_B (GB)	PM	1824 ± 25.7	10,307 ± 194.4	17,554 ± 236.6
	$\overline{PM}$	1636 ± 22.9	9896 ± 102.8	16,832 ± 429.1
	nR	1441 ± 20.9	8787 ± 85.6	15,357 ± 298.6

only a few of them in the network. For this reason, the linear behavior of the plots stop at this time in the simulations as the contribution of the data served by these remaining UAVs to the cumulative total data is comparatively very low than that of the prior total data.

$$\phi_B^{t_k} = \sum_{t=1}^{t_k} \sum_{i=1}^{M^{t_k}} B_{u_i}^{t_k} \tag{11}$$

where, $B_{u_i}^{t_k}$ is the data served to user, u_i, at time, t_k.

Finally, Fig. 15 shows the contribution percentages of various components in the proposed mechanism. *Only Redistribution* refers to the case when only load redistributions were sufficient to serve the hot zones. Whereas, the *1-hop assistance* and *Multi-hop assistance* refer to the cases when one or more UAVs were repositioned, respectively, after performing load redistributions.

Redistribution of load during hot zones can avoid UAV movements and in turn disruption in the network topology. *Only Redistribution* refers to this aspect of solving the hot zone problem which was not considered in the previous work [1]. The UAV movements were avoided for 18.2% of the hot zone instances, thus, signifying the importance of this component in the proposed method. However, the UAV movement occurred around four times than the *Only Redistribution* (81.8%). This shows that repositioning the UAVs is paramount to serve users in unpredictable events. Further, the much higher occurrence of *Multi-hop assistance* over the *1-hop assistance* (by 22.4%), emphasizes the required movement of multi-hop peers. As the previous work [1] did not consider *Multi-hop assistance*, this paper shows the advantage of including this component in the network performance.

The 95% confidence intervals for the three accumulative parameters, $B_{\Delta J}$, $\mu\Delta\xi$ and ϕ_B at the beginning (600 s), middle (3600 s) and final (7200 s) stages of the simulation are shown in Table 3.

5. Conclusions and future work

The proposed method of providing coverage to users by an Unmanned Aerial Vehicle network follows a two-fold scheme when hot zones occur in the region. In the first process, the Unmanned Aerial Vehicle serving the hot zone tries to release its load to its 1-hop neighbors in the corresponding overlapped sections. If the Unmanned Aerial Vehicle is still unable to serve the remaining users, then it initiates the second process to find a suitable Unmanned Aerial Vehicle (whose current user count is less than what it would share), as the objective is to serve the maximum possible number of users at any instant. Finally, in the swapping process (independent of hot zone occurrence), every Unmanned Aerial Vehicle compares its predictive lifetime with its 1-hop neighbors based on the current loads. It swaps its position (with one of the neighbors) when the difference is more than τ time units. This process improves network performance as the Unmanned Aerial Vehicles with more resources (energy) move to a more crowded location. The proposed methods outperform the *nR* case by serving more data throughout the simulation time.

An entire cell is considered as a hot zone. However, a hot zone can encompass multiple *hotspot cells*, either entirely or partially as user density or movement is unpredictable. The future work will be focused in this direction.

Declaration of interest

None.

Acknowledgment

This research was supported by NSF Grant Award #1723814.

References

[1] Patra AN, Sengupta S. Dynamic deployment of uav-enabled floating access points for serving hot zones. International symposium on performance evaluation of computer and telecommunication systems. IEEE; 2017.
[2] Galkin B, Kibilda J, DaSilva LA. Deployment of uav-mounted access points according to spatial user locations in two-tier cellular networks. In: Wireless days (WD). IEEE; 2016. p. 1–6.
[3] Zhang X, Duan L. Optimal deployment of uav networks for delivering emergency wireless coverage. arXiv:171005616v1 2017.
[4] Deployable aerial communications architecture in emergency communications. goo.gl/EPDq79.
[5] National public safety telecommunications council. http://www.npstc.org/.
[6] Virtual network communications. http://www.virtualnetcom.com/index.html.
[7] Deruyck M, Wyckmans J, Martens L, Joseph W. Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario. International conference on wireless and mobile computing, networking and communications. IEEE; 2016.
[8] Lyu J, Zeng Y, Zhang R, Lim TJ. Placement optimization of uav-mounted mobile base stations. IEEE Commun Lett 2017;21(3):604–7.
[9] Sharafeddine S, Islambouli R. On-demand deployment of multiple aerial base stations for traffic offloading and network recovery. arXiv:180702009v1 2018.
[10] Lu J, Wan S, Chen X, Fan P. Energy-efficient 3d uav-bs placement versus mobile users' density and circuit power. arXiv:170506500v1 2017.
[11] Yang P, Cao X, Yin C, Xiao Z, Xi X, Wu D. Proactive drone-cell deployment: overload relief for a cellular network under flash crowd traffic. IEEE Trans Intell Transp Syst 2017;18(10):2877–92.
[12] Mehta PL, Prasad R. Aerial-heterogeneous network: a case study analysis on the network performance under heavy user accumulations. Wireless Pers Commun Springer 2017;96(3):3765–84.
[13] Fotouhi A, Ding M, Hassan M. Dynamic base station repositioning to improve spectral efficiency of drone small cells. arXiv:170401244v1 2017.
[14] Ma M, Yang Y. Adaptive triangular deployment algorithm for unattended mobile sensor networks. IEEE Trans Comput 2007;56(7). 946–847.
[15] Regis PA, Patra AN, Sengupta S. Unmanned aerial vehicles positioning scheme for first-responders in a dynamic area of interest. In: Vehicular technology conference. IEEE.
[16] Yates RD, Huang C-Y. Integrated power control and base station assignment. IEEE Trans Veh Technol 1995;44(3):638–44.
[17] Hanly SV. An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity. JSelAreas Commun. 1995;13(7):1332–40.
[18] Kuek S, Wong W. Dynamic load sharing for asymmetrical microcellular teletraffic conditions. Electron Lett 1991;27(9):765–7.
[19] Circle-circle intersection. http://mathworld.wolfram.com/Circle-CircleIntersection.html.
[20] ns-3 Consortium. Network simulator, ns-3. https://www.nsnam.org/.
[21] Flight time. https://www.forbes.com/sites/jeremybogaisky/2018/09/10/impossible-aerospace-drone-2-hours/#2825f4a75554.
[22] Fadlullah ZM, Takaishi D, Nishiyama H, Kato N, Miura R. A dynamic trajectory control algorithm for improving the communication throughput and delay in uav-aided networks. IEEE Netw 2016;30(1):100–5.
[23] Amazon prime. https://en.wikipedia.org/wiki/Amazon_Prime_Air.
[24] New drone claims guinness world record with a top speed of 163 mph. goo.gl/gZca57.

Amar Nath Patra is a Ph.D. student in the Department of CSE at the University of Nevada, Reno and received his masters from Indian Institute of Technology, Dhanbad, India. His research interests include mobile ad-hoc networks. He is the recipient of International Symposium on Performance Evaluation of Computer and Telecommunication Systems 2017 best paper award.

Paulo Alexandre Regis is a Ph.D. Candidate at the University of Nevada, Reno. Paulo obtained his B.S. in Telecommunication Engineering in 2010 at the Regional University of Blumenau, Brazil. His research interests are optimization, security and application of mobile ad hoc networks. His background experience includes mobile development and network systems administration.

Shamik Sengupta is an Associate Professor in the Department of Computer Science and Engineering and Executive Director of the Cybersecurity Center at the University of Nevada, Reno (UNR). His research interests include cognitive radio and DSA networks, game theory, cybersecurity, network economics and self-configuring wireless mesh networks.