Distributed Allocation and Dynamic Reassignment of Channels in UAV Networks for Wireless Coverage

Amar Nath Patra*, Paulo Alexandre Regis, Shamik Sengupta

Department of Computer Science and Engineering
University of Nevada, Reno, Nevada 89557, USA

Abstract

Providing wireless coverage to users using Unmanned Aerial Vehicles (UAVs) encounters two major challenges: deployment and channel allocation. For this, initial deployment and channel allocation plans are proposed in this paper. An overloaded UAV first tries to acquire more channels by performing channel bonding/aggregation followed by requesting its chosen peers to move closer for sharing load. The proposed channel reallocation schemes minimize interference caused by channel reassignments or change in network topology. The simulation results show that on employing proposed reallocation schemes, more data is served with less discontinuous service time along with efficient usage of limited battery power.

Keywords: Unmanned Aerial Vehicles, Delaunay Triangulation, Channel Bonding, Channel Aggregation, Radio Access Technology

---

*Corresponding author

Email addresses: apatra@nevada.unr.edu (Amar Nath Patra), pregis@nevada.unr.edu (Paulo Alexandre Regis), ssengupta@unr.edu (Shamik Sengupta)

1. Introduction

A cellular network outage in a region caused by a natural calamity increases the threat to human lives by affecting emergency response and restoration. An outage also causes social and economical repercussions on the populace. Troubleshooting and recovering from the downtime is time-consuming. In such emergency events, a flock of Unmanned Aerial Vehicles (UAVs) can be deployed. Each deployed UAV covers an area on the ground, called a hotspot cell, and serves the users in it. Since no infrastructure is required for the deployment, this solution is effective in restoring communication. Centralized Control (CC) can help in initial deployment but may not efficiently reposition the deployed UAVs because some areas (harsh terrains) may be unreachable. This state of remoteness requires UAVs' autonomous decision making which includes both individual and collaborative work.

The users' mobility and bandwidth requirement can be unpredictable, requiring the UAVs to change their initial positions for sharing the load dynamically. While moving, an UAV should ensure that it has at least one 1-hop neighbor. The reader is forwarded to [1] for a possible solution to these problems. The UAVs should be initially assigned non-interfering channels to serve their users. Finally, the mobility of UAVs, and a restricted or unavailable CC would require dynamic and distributed channel reallocation schemes which the UAVs should follow.

An overburdened UAV first tries to combine multiple channels to get an aggregated channel with a higher bandwidth by performing link aggregation. Later, if required, it requests one or more peers to move closer to share its load. UAVs have limited energy and consume it at a high rate to be airborne. Thus, the UAVs should dynamically reposition themselves to maximize the user count in their cells. Link aggregation and UAV movements increase chances of channel interference since they change the initial channel allocation. Reallocation can further cause interference among others, generating a ripple effect in the UAV network. Hence, this paper proposes reallocation methods that dynamically reassign channels to reduce the ripple effect.

In an emergency event, first responders may collaborate from different teams and thus may have UAVs with different capabilities and coverage. Thus, in the proposed method, two types of UAVs are considered: multi-Radio Access Technology (RAT) and uni-RAT. The former ones are capable of communicating through multiple RATs, unlike the latter ones. Two far-away multi-RAT UAVs can communicate through the RAT which has a higher range. The initial deployment positions the UAVs so that the multi-RAT UAVs simultaneously initiate channel allocation to their 1-hop neighbors, resulting in concurrent allocations to UAVs throughout the network.

The rest of the paper is organized as follows. Section 2 overviews work on channel allocation and UAV networks in providing wireless coverage. Section 3 presents the proposed intelligent scheme of channel allocation considering the cases: CA, CB, and UAV movements. The simulation parameters and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

Mozaffari et al. [2] proposed UAV deployment considering downlink coverage probability, altitude and antenna gain. Lyu et al. [3] minimized UAV count to serve ground terminals and proposed a polynomial-time deployment algorithm. Huo et al. [4] discussed UAVs in 5G network and presented a hierarchical deployment architecture. Moraes et al. [5] presented a distributed repositioning algorithm for UAV swarms (self-organizing UAV network) in communication relay networks for surveillance missions. Orfanus et al. [6]

proposed a self-organizing paradigm to design efficient UAV relay networks to support military operations. These works proposed efficient deployments but did not consider effective channel allocation and reallocation schemes (to resolve probable interferences) due to the UAV movements.

Wang et al. [7] presented list-coloring based channel allocation scheme for wireless networks, considering opportunistic spectrum availability. Zeng et al. [8] analyzed pairing stability in device-to-device (D2D)-relay networks and showed the positive correlation between the proposed metric and the system performance. Xu et al. [9] analyzed the impact of fast time-varying channels to statistical signal transmission and proposed a channel condition aware detection scheme. These works consider only static environments.

To the best of our knowledge, this is the first work that discusses channel allocation in a UAV network considering the scenario of variable user movement and requirements and resolves interferences due to changes in initial allocation plan and network topology.

3. Proposed Methodology

To provide wireless coverage in an emergency event, a three-fold approach is proposed here: (1) deployment of available UAVs to cover maximum possible continuous area; (2) allocation of air to ground channels to UAVs; and (3) efficient reallocation of these channels (by minimizing possible interferences) when one or more UAVs are unable to serve an increased number of users in their cells. The reallocation schemes support link aggregation methods when multiple channels are assigned to UAVs.

3.1. System Model

UAVs are classified based on RATs: *L_drones* (multi-RAT: LTE, WiFi), and *u_drones* (uni-RAT: WiFi). Two 1-hop *u_drones* communicate via WiFi

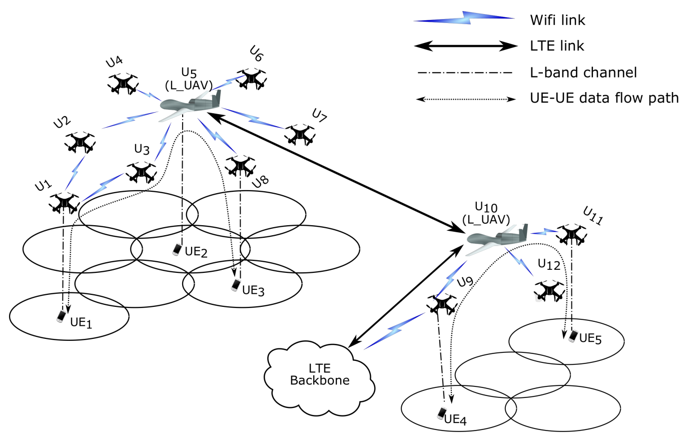


Fig. 1: System Model

links, whereas the *L_drones* communicate via LTE link. As LTE has a longer coverage range, two 1-hop *L_drone* neighbors can be comparatively farther from each other. A *L_drone* communicates with a 1-hop *u_drone* via WiFi links. Fig. 1 shows a deployment of 12 UAVs (2 *L_drones* and 10 *u_drones*), and gives an outline of the communication scheme of the overall system. A User Equipment (UE) communicates with another UE in various ways [10]. The proposed work focuses only on UAV-UE (air to ground) channel allocations and communication, and the associated UAV-UAV communication required for these channel allocations. In the figure, UE_1 and UE_3 communicate following the path: U_1, U_3, U_5 and U_8, whereas, UE_4 and UE_5 communicate following the path: U_9, U_{10}, and U_{11}.

Further, to meet a surge in user demand, an overloaded UAV tries to increase its bandwidth by performing the link aggregation methods: channel bonding (CB) and channel aggregation (CA). CB combines contiguous channels to get a combined channel with higher bandwidth. Whereas, in CA, data is transmitted simultaneously on all available channels, thus providing load balancing [11]. CB is possible only when the adjacent channels are available, while CA can be performed only on those channels which are in the range supported by the hardware. If the user demand is still not met, then the UAV requests one or more peers to move closer to share its load. Since the battery life of the UAVs is limited, it considers the peers with higher remaining energy as they can share its load for a longer period.

3.2. Assumptions

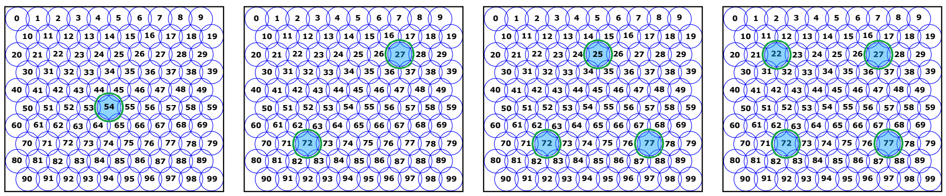
i) There are N L-band channels ([12]), c_1 to c_N (known to all UAVs), used by UAVs for air to ground links.

ii) UAVs fly at the same altitude and cover the same area. They share their locations (determined by GPS) with the 1-hop neighbors along with the assigned channel number through *Hello_Msgs*. Hence, an UAV knows the spatial location of its neighbors and maintains the entries in its neighbor table (NT) to represent their clockwise positions around it.

iii) *Hello_Msgs* are retransmitted periodically (hop count set to 1) to enable every UAV to have the knowledge of the positions and assigned channel numbers of its 2-hop neighbors [13] also.

3.3. Initial Deployment of UAVs

A fixed count of UAVs is considered for deployment to cover as much area as possible while making sure that they are connected to each other. For this, the Delaunay Triangulation with equilateral triangles is followed to provide maximum coverage area with minimum overlaps and no gaps between adjacent hotspot cells [14]. The *L_drones* are deployed in the region according to their availability. As they are going to (simultaneously) initiate the channel allocation process (explained in the next subsection), they are positioned approximately twice as far from each other as they are from the



(a) 1 *L_drone* **(b)** 2 *L_drones* **(c)** 3 *L_drones* **(d)** 4 *L_drones*

Fig. 2: Initial Deployment (100 UAVs)

nearest corner of the region to ensure a fast convergence of channel allocation in the entire network. Later, the *u_drones* take the remaining positions.

Fig. 2 shows example deployments of 100 UAVs with a different set of *L_drones* in each case. The *L_drones* are represented by their hotspot cells, shown as blue shaded circles, encircled in green rings. The outer green rings represent the LTE RAT whereas the inner blue circles represent WiFi RAT. (The green rings do not reflect the actual coverage range of LTE RAT and only distinguishes *L_drones* from the *u_drones*.)

3.4. Channel Allocation in Initial Deployment

After the deployment, Ground Control Station (GCS) sends a message to the nearest *L_drone* to start allocation process. This *L_drone* communicates with other *L_drones* through a shared LTE channel. They all assign the same predetermined L-band channel to themselves, after which they simultaneously offer a single channel (from the remaining channels) to each of their 1-hop neighbors. These channels are concurrently offered to them. This completes I round. In the rest of the paper, *nbr* of an UAV is going to refer any of it's 1-hop/2-hop neighbors and *nbrhood*, the set of all the *nbrs*.

Subsequently, the 1-hop neighbors of the *L_drones* offer channels to their own 1-hop neighbors. However, these channels are offered sequentially. To elaborate, an UAV, U_i, starts with the first 1-hop neighbor in its NT and eventually offers channels to the remaining neighbors, successively. While doing so, it considers only those neighbors which do not have any channels assigned to them (*unassigned neighbors*). To offer channels, it prepares an *offer-list* of channel numbers, excluding its own channel number and those assigned to its 1-hop neighbors (*assigned neighbors*), and sends (unicasts) an *Offer_Msg* to the first 1-hop unassigned neighbor, $U_{i_{n1}}$, in its NT. $U_{i_{n1}}$ waits to receive *Offer_Msgs* from all of its 1-hop assigned neighbors. It accepts the smallest channel number which is common in all the received *offer-lists* and is not assigned to any of its *nbrs*. Consequently, $U_{i_{n1}}$ replies (1-hop broadcast) with an *Alloc_Msg* (Its 2-hop neighbors will receive this information from the 1-hop neighbors through *Hello_Msgs*).

U_i then moves on to the next 1-hop unassigned neighbor $U_{i_{n2}}$, whereas $U_{i_{n1}}$ simultaneously starts sending *Offer_Msgs* to its own 1-hop unassigned neighbors, initiating or contributing to the next round. Thus, *Offer_Msgs* are sent by different UAVs concurrently throughout. This process continues until all the UAVs are assigned with channels (Algorithm 1).

To avoid interferences due to the concurrency of channel allocation process by different UAVs, a UAV sequentially offers channels to its 1-hop neighbors in the subsequent rounds. Further, the UAVs which are within the *nbrhood* do not set the same channel: only those two UAVs which are at least 3-hops away can set the same channel. This constraint enables a faster channel reallocation as opposed to when only 1-hop neighbors are not allowed to set the same channel. This is because the case of 1-hop constraint will have same channels allocated in nearer cells and hence will result in a higher cascaded effect of channel interference during channel reallocation.

The example in Fig. 3(a) considers 4 *L_drones* which start the channel allocation process after assigning the channel, c_1, to themselves. Each *L_drone* prepares its own *offer-lists* containing a single channel (in each) and

Algorithm 1: Initial Channel Allocation

Input: channel information of *nbrhood*

Output: channel allocation in entire network

GCS sends message to the nearest *L_drone*, L_i;

`/* I Round: */`

L_i connects with other *L_drones* via shared LTE channel and all the *L_drones* assign a common L-band channel to themselves;

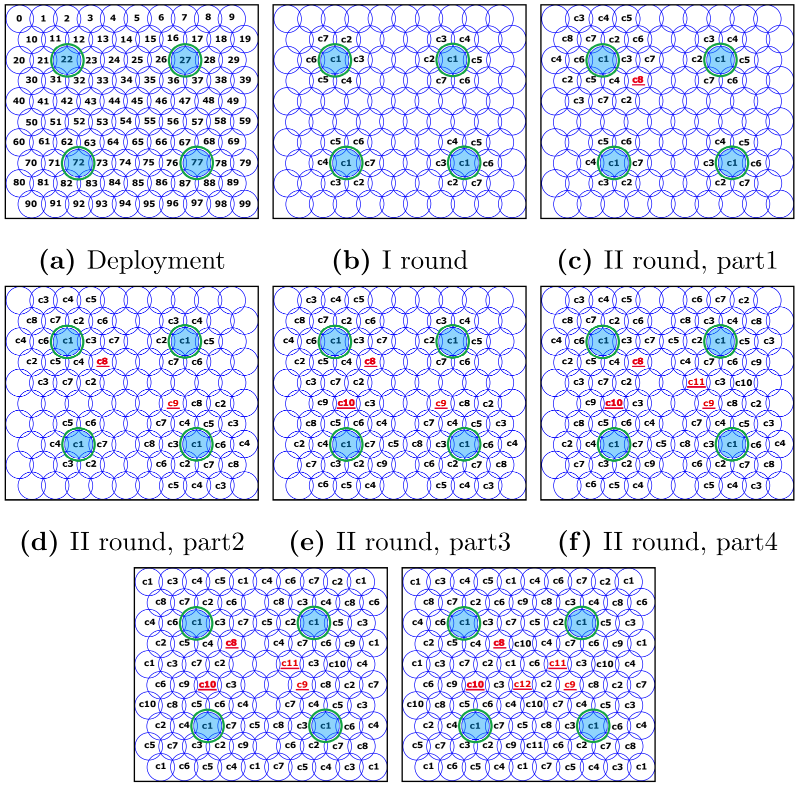
L_drones concurrently assign channels from remaining set to their 1-hop neighbors simultaneously;

`/* Subsequent Rounds: */`

while *All the UAVs are not assigned with a channel* **do**

- An UAV, U_i, prepares an *offer-list* sends *Offer_Msg* to $U_{i_{n1}}$;
- $U_{i_{n1}}$ waits for *Offer_Msgs* from 1-hop *assigned neighbors*;
- $U_{i_{n1}}$ sets lowest non-interfering channel number to itself and broadcasts 1-hop *Alloc_Msg* as response;
- U_i moves to its next 1-hop *unassigned neighbor*, $U_{i_{n2}}$, and $U_{i_{n1}}$ starts sending *Offer_Msgs* to its own 1-hop *unassigned neighbors* (initiating or contributing to the next round);

sends *Offer_Msgs* to all of its 1-hop neighbors simultaneously, completing the I round. A *L_drone* offers the channels to these neighbors concurrently as it knows the spacial arrangement of them through its NT. So, c_2 is offered to first 1-hop neighbor in the NT, c_3 to the second, until c_7 is offered to the sixth 1-hop neighbor. All of these neighbors reply with *Alloc_Msgs* as a confirmation. Fig. 3(b) shows the I round of the process (The UAV IDs are replaced by the allocated channel numbers). Until now seven channels,



(a) Deployment **(b)** I round **(c)** II round, part1

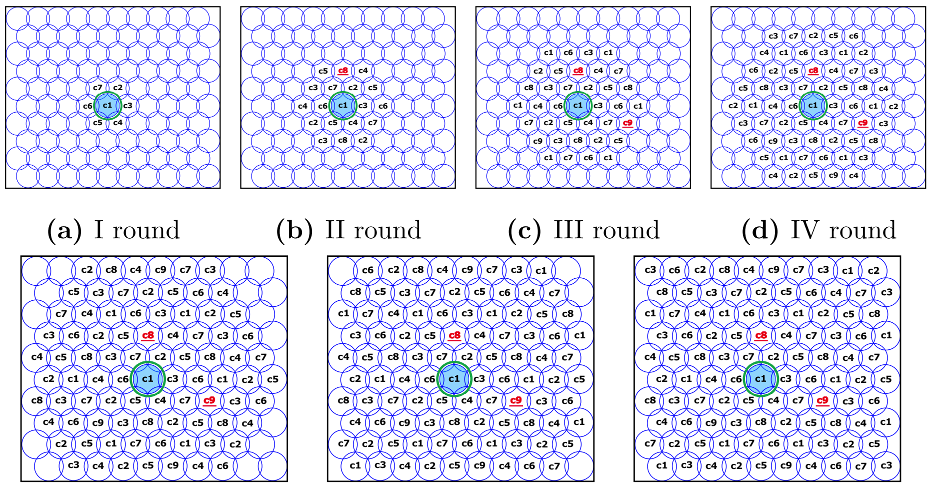
(d) II round, part2 **(e)** II round, part3 **(f)** II round, part4

(g) III round, part1 **(h)** III round, part2

Fig. 3: Example: Channel Allocation with four *L_drones*

c_1 - c_7, are sufficient to avoid interference. Fig. 3(c)-(f) show the channel allocation in the II round when the 2-hop neighbors of the four *L_drones* set their channels. Although the allocation in this round occurs simultaneously, it is shown in four different sub-figures for clarity. Here, four new channels, c_8 - c_{11}, are added. The first instances are in red font and underlined. Fig. 3(g)-(h) show the III (final) round in which another channel, c_{12}, is added.

To explain the addition of a new channel, II round (part 2), shown in Fig. 3(d) is examined here. Assuming that the UAV, $U_{i_{n1}}$ (shown with the assigned channel c_8), has not set any channels yet, it will receive *Offer_Msgs* from four of its 1-hop *assigned neighbors*. These neighbors are considered



(a) I round **(b)** II round **(c)** III round **(d)** IV round

(e) V round **(f)** VI Round **(g)** VII Round

Fig. 4: Example: Channel Allocation with one *L_drone*

clockwise (starting from bottom left) with channel numbers, c_2, c_4, c_3 and c_7. These UAVs send the following *offer-lists* in their *Offer_Msgs*: {c_1, c_3, c_5, c_6, c_8, c_9, ..., c_N}, {c_6, c_8, c_9, ..., c_N}, {c_5, c_8, c_9, ..., c_N} and {c_1, c_2, c_4, c_5, c_8, c_9, ..., c_N}, respectively. The lowest channel number common to all these lists is c_8 which is also not allocated by any of the $U_{i_{n1}}$'s *nbrs*, so, it assigns c_8 to itself and broadcasts an *Alloc_Msg* as a response. Similar analysis can be done for the new channels, c_9 - c_{12} in Fig. 3(d)-(h).

Comparatively, when there is one *L_drone*, the process converges slower but requires a fewer set of channels. Fig. 4(a)-(g) show all the rounds. Evidently, comparing with the previous example, 7 rounds are needed instead of 3 for the convergence. However, only 9 channels are sufficient to avoid any interference as opposed to the 12 channels in the previous example. Hence, the convergence of the process depends on the number of rounds which is the number of minimum hops from the *L_drone* to the farthest UAV which sets its channel due to the process initiated by the *L_drone*.

The *L_drones* are deployed in an evenly scattered manner so that they are approximately twice as far from each other as they are from the nearest corner of the region. Such a deployment of *L_drones* will ensure that the farthest UAVs receive an *Offer_Msg* the earliest. Although an increase in the count of *L_drones* makes the allocation process faster due to parallelism, a greater number of channels would be required for allocation because an increased level of concurrency results in more conflicts in channel allocation.

Time Analysis of Complete Execution: The total delay (transmission, propagation and processing times) for 1-hop message is denoted by τ_d. The I round will complete after a *L_drone* receives *Alloc_Msgs* from all the 1-hop neighbors. Since the *Offer_Msgs* are sent simultaneously, the time taken for completion of the I round, $T_f = 2\tau_d$.

For subsequent rounds, a U_i sends *Offer_Msgs* to its 1-hop neighbors sequentially. There can be a maximum of three such neighbors (Fig. 3(c)), thus, the maximum time required to complete a round, T_{sub} is given by

$$T_{sub} = 3 \times 2\tau_d = 6\tau_d \tag{1}$$

Now, the number of rounds taken by each *L_drone* to allocate channels to all of its peers is equal to the minimum distance in hops between the *L_drone* and the farthest peripheral UAV which receives an *Offer_Msg* through the allocation process initiated by it. This parameter is named *local radius*, denoted by R_{local}. In case the *L_drones* are not having the same *local radius* values (due to the irregular shape of the region), *local radius* is set with the largest value among them for further calculations. The total time taken in allocating channels to all the UAVs, T_{total} is given by

$$T_{total} = T_f + (R_{local} - 1)T_{sub} \tag{2}$$

In the first example (Fig. 3), R_{local} is 3. Thus, using Eq. 2, the total

Table 1: Varying count of *L_drones*

Number of L_drones	Convergence time	Number of channels
1	38 τ_d	9
2	32 τ_d	10
3	32 τ_d	11
4	14 τ_d	12

time required is 14 τ_d. However, in the second example (Fig. 4), R_{local} is 7, giving the total time as 38 τ_d. Table 1 compares the convergence time and the number of required channels when different count of deployed *L_drones*. It shows that by increasing the count of *L_drones*, the convergence time decreases, however the number of required channels, increases. Although for both the cases of 2 and 3 *L_drones*, the convergence times are same, a higher count of UAVs are allocated with channels in each round when there are three *L_drones* as opposed to when there are two *L_drones*.

The proposed allocation procedure ensures that no channel is revoked as it determines the next channel to be allocated by using *neighbor* information, *Offer_Msgs*, and *Alloc_Msgs*. Availability of information about the allocated channels by peers and a sequential exchange of *Offer_Msg* and *Alloc_Msg* avoids any chance of channel interference. Once an UAV sets a channel for itself, it immediately starts communication with the users in its cell. The robustness of the proposed approach allows fast and seamless communication for the users as no UAV has to wait for the channel allocation to other UAVs.

However, an allocation process in not sufficient as the users move in the region and show random service requirement. This uncertainty of mobility and demand may result in some UAVs to experience overload. These UAVs then try to acquire additional channels and later request some of the peers to move closer to share the load. Attaining additional channels and changing the network topology disrupts the initial channel allocation plan and thus resulting in channel interferences, degrading the network performance as the users observe a discontinuity in the service. Allocating different channels to the adjacent UAVs sharing the same channel will resolve interference.

3.5. Channel Reallocation Triggered by Overloaded Hotspot Cells

An UAV, U_i computes its QoS affect ratio, $Q_i = \frac{B^*}{B}$, periodically, where B^* is the current total bandwidth allocated to its users and B is its total bandwidth capacity. $Q_i > 1$ implies U_i can not satisfy the QoS requirements of its users and thus a Hot Zone is created in its hotspot cell [1]. U_i first tries to acquire more channel(s) to satisfy the users' demands. However, if it needs more channels than its air-to-ground RAT interface can support, U_i chooses one or more peers. The chosen peer(s) come closer and overlap U_i's cell to share its load [1]. Unlike the first case, the second case involves UAV movements. However, in both the cases, channel reallocation is necessary to minimize (if not completely remove) the interference effects.

For analyzing the two cases, a parameter is proposed here, called *cdnlty* of a channel, c_α, which represents the count of *nbrs* of U_i that have assigned c_α to themselves.

$$cdnlty(c_\alpha) = |\{UAV^\alpha_{n_1}, UAV^\alpha_{n_2}, ...,\}| \tag{3}$$

where $UAV^\alpha_{n_i}$ is a *nbr* of U_i which has assigned c_α to it.

3.5.1. ***No UAV Movements***

Since a UAV knows the total number of channels, N, and the assigned channels in its *nbrhood*, U_i can determine if there is any unassigned channel in its *nbrhood*. If found, it checks whether it can aggregate it with its own

Algorithm 2: Channel_Reallocation-I: No UAV Movements

Input: current information of assigned channels

Output: updated channel allocation

U_i checks for non-interfering channel available in the *nbrhood*;
if *found* **then**
    if *channel aggregation can be performed* **then**
        keeps a record of such channels;
        accepts the channels successively till required;
if *another channel required* **then**
    calls Channel_Bonding procedure;
    if *Channel_Bonding successful* **then**
        accepts the channel;
while *another channel required* **do**
    calls Channel_Aggregation procedure;
    if *Channel_Aggregation successful* **then**
        accepts the channel;
    if *aggregation not possible any more* **then**
        exit;

channel. Otherwise, it tries to acquire a channel adjacent to its current channel to perform CB (CB procedure call). If CB fails, then U_i requests channels from its *nbrs* to aggregate with its own channel (CA procedure call). These steps are listed in Algorithm 2. Taking channels from *nbrs* will result in interference and requires reallocation for the affected UAVs.

Subcase1 - Attempting Aggregation Prior to CB/CA Proce-

Algorithm 3: Channel_Bonding

```
Input: channel information of nbrhood
Output: success of channel bonding
U_i checks for both of the adjacent channels unassigned in nbrhood;
if both unassigned then
    bonds the lower channel number;
else
    if either unassigned then
        bonds this channel;
        returns true;
    else
        sends Req_Msg to the nbrs associated with the lower cdnlty;
        if Positive Resp_Msgs are received from all these nbrs then
            bonds this channel;
            returns true;
        else
            sends Req_Msg to the nbrs associated with higher cdnlty;
            if Positive Resp_Msgs are received from all these nbrs
             then
                bonds this channel;
                returns true;
            else
                returns false;
```

dure Calls: U_i checks for an unassigned channel in its *nbrhood* and ascertains the possibility of CA. CA with such a channel prevents any interference

Algorithm 4: Channel_Aggregation

Input: channel information of *nbrhood*

Output: success of channel aggregation

U_i prepares Clist, and starts checking with its first member;
while *end of Clist not reached* **do**
    U_i sends *Req_Msgs* to *nbrs* assigned with considered channel;
    if *Positive Resp_Msgs received from all these nbrs* **then**
        aggregates this channel;
        returns **true**;
        exit;
    else
        U_i considers next channel in Clist;
returns false;

and the associated ripple effects. If U_i is successful in acquiring the required number of channels in this case, it does not proceed further.

Subcase2 - Attempting Bonding: Assuming U_i has channel, c_α, it checks if any of its *nbrs* have the bonding candidates ($c_{\alpha-1}$ and $c_{\alpha+1}$). If either is unassigned, it bonds the channel. Otherwise, it tries to obtain the one with least *cdnlty* (assigned to a minimal count of *nbrs*) by sending *Req_Msg* to these *nbrs*. A *nbr* will respond with a *Positive Resp_Msg* if it is not overloaded otherwise it replies with a *Negative Resp_Msg*. If U_i receives positive responses from all of them, it bonds this channel. Otherwise, it tries to obtain the other channel in a similar way (Algorithm 3).

Subcase3 - Attempting Aggregation: If CB fails, U_i obtains a channel comparing the *cdnlty* of the aggregation candidates. It does so by arranging them in increasing order of *cdnlty* (ties are broken by giving higher priority to a lower channel number), creating a *cdnlty* list, Clist, and starts checking with its first member. If all the *nbrs* of a considered Clist member respond with *Positive Resp_Msgs* then CA is successful (Algorithm 4).

Comparing Channel Bonding and Channel Aggregation: Since channels far apart in the spectrum cannot be aggregated, U_i considers only those for CA which its interface can support. CB, on the other hand, provides an additional 10% capacity as the guard bands between the adjacent channels can be used for transmission [11]. However, as CB is possible only with adjacent channels, there are not more than two possible channels; hence, the probability of acquiring them is far lesser than performing CA.

U_i may allocate more than one additional channel by employing CB/CA, following the previous steps. It is assumed that the users can receive data if multiple channels are bonded or aggregated. However, if it cannot further bond or aggregate additional channels, due to unavailability, hardware constraints, or diminished battery life (the higher the bandwidth, more the power consumption), then it looks for a suitable peer to come to its rescue.

*3.5.2. **UAV Movements***

U_i broadcasts a *Req_Msg* to all of its *nbrs* advertising the excess load. Only those *nbrs* whose current load is less than the advertised load reply with a *Positive Resp_Msg.* U_i selects one of them, considering their current QoS affect ratios, remaining energies, and hop-count distances and then sends it a *Req_Msg.* Moving a peer with a lower QoS affect ratio will result in a comparatively fewer number of affected users in the *nbrhood.* Additionally, a peer with a higher remaining energy can share the load with U_i for a longer time. Finally, U_i prefers a peer nearer to it than a farther one. As the

Algorithm 5: Channel_Reallocation-II: UAV Movements

Input: channel information of *nbrhood*

Output: updated channel allocation

nbr_{chosen} checks its new *nbrhood* for any interference;
if *none* **then**
    continues using its original channel for sharing the load of U_i;
else
    checks *nbrhood* for unassigned channels;
    if *found* **then**
        accepts it;
    else
        accepts channel with least *cdnlty* (by sending *Req_Msgs*);

objective is to provide seamless coverage to as many users as possible, QoS affect ratio is given the higher priority than the remaining energy. Since, the UAVs can move very fast, the least priority is given to the parameter, hop-count. U_i computes a function, *select*, based on these three parameters and their priorities, given in Eq. 4 and selects the one with the highest value as given in Eq. 5. E_j refers to the remaining energy of the peer U_j whereas hop_{ij} denotes the hop-count between U_i and U_j. Variables, a, b and c are the weighting coefficients of the three parameters (Section 4).

$$select_j = \frac{b \times E_j}{(a \times Q_j) \times (c \times hop_{ij})} \tag{4}$$

$$nbr_{chsn} = \arg\max_{i \neq j} (\ select_j) \tag{5}$$

Chosen *nbr*, nbr_{chsn} after moving to the new location, checks if it can

Algorithm 6: Channel_Reallocation-III: Affected UAVs

Input: channel information of *nbrhood*

Output: updated channel allocation

after relinquishing its channel, *nbr* checks if any non-interfering channel available in the *nbrhood* which could be aggregated;

if *found* **then**

  accepts it;

else

  accepts the channel with least *cdnlty* (by sending *Req_Msgs*);

  the UAVs to which this channel was assigned, excecute their own **Channel_Reallocation-III** procedures;

continue to use its channel in the new *nbrhood*. If interference is observed, it chooses a channel with least *cdnlty* (Algorithm 5). These checks, extensive message exchanges, operations, and steps are followed in the same way as in the previous case of no UAV movements.

In the above cases, when U_i or nbr_{chosen} is successful in acquiring a new channel, the *nbrs* to which this channel was allocated avoid interference with U_i by checking their respective *nbrhoods* to acquire a non-interfering channel with respect to the *nbrs* (including U_i and nbr_{chosen}). However, if none are found, then such a *nbr* obtains the channel with the least *cdnlty* by following similar process of Req_ and Resp_Msg exchanges, creating a chain reaction (Algorithm 6). This causes a *ripple effect* around the Hot Zone. By always preferring a channel with the least *cdnlty*, the UAVs strive to reduce ripple effects. Moreover, to curb it, a threshold on the permissible maximum *cdnlty*, $cdnlty_{max}$ is set. For example, if $cdnlty_{max}$ is set to 3, then a channel with a $cdnlty > 3$ is never considered as a candidate. It may happen that an UAV does not find any channels and thus, interference cannot be avoided. Such an UAV then shares the channel with one or more *nbrs.* The interaction of Algorithm 2-6 are shown in Fig. 5.

An UAV, U_j on receiving requests from multiple UAVs who are experiencing Hot Zones, accepts the first request.

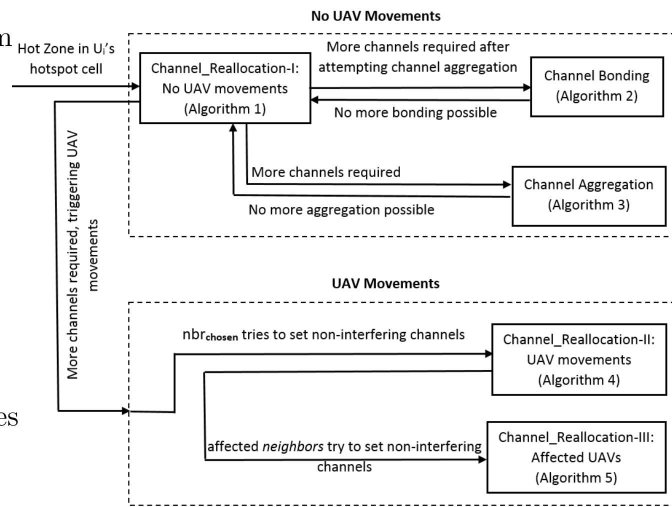
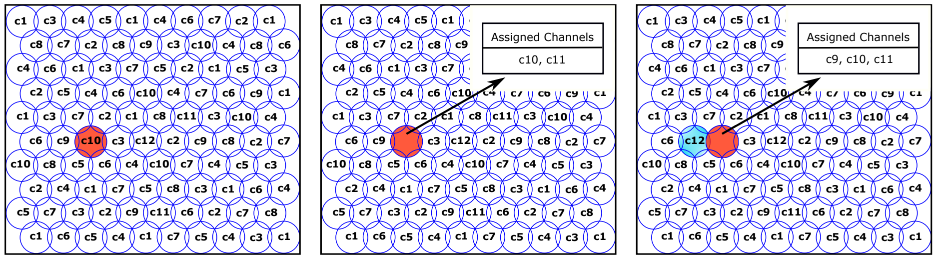


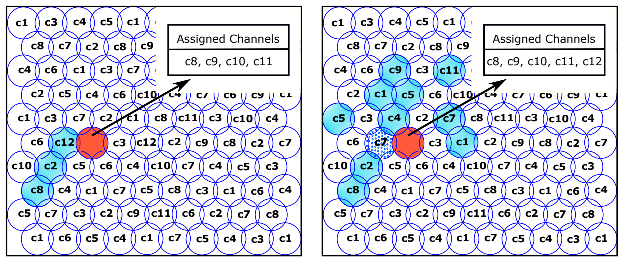
Fig. 5: Interaction of reallocation algorithms

Example Scenario: Fig. 6(a) discusses different cases of the proposed reallocation method. A Hot Zone, depicted by the red cell, is served by the UAV U_i and has channel c_{10} assigned to it. It is assumed that there are N = 12 channels. For CA, the constraint is that an UAV with channel c_α can aggregate only $c_{\alpha-2}$ and $c_{\alpha+2}$. U_i checks its NT to determine if the CA candidates, c_8 or c_{12} are unassigned in its *nbrhood*. As can be seen in Fig. 6(a), both of these channels are assigned to two of its *nbrs*. U_i tries to perform CB during which it checks the better alternative between c_9 and c_{11} by comparing their respective *cdnlty*. Since c_{11} has lower (zero) *cdnlty*, it bonds its channel, c_{10} with c_{11}. A zero *cdnlty* implies no ripple effect, thus requiring no reallocation at any of the *nbrs*. Now U_i has channels, c_{10} and c_{11}, shown in Fig. 6(b).

U_i tries CA when it needs more channels and chooses between c_8 and c_{12} based on their *cdnltys*. It compares these *cdntlys* along with that of the



(a) Initial channels **(b)** I reallocation: CB **(c)** II reallocation: CA



(d) III reallocation: CA **(e)** IV reallocation: CA

Fig. 6: Example: Channel Reallocation

CB candidate which was not considered (c_9). As the channels c_8, c_9 and c_{12} have the same *cdnlty* of 1 (Fig. 6(a)), U_i chooses c_9 as it can be bonded with the current pair of channels, (c_{10}, c_{11}) giving 10% additional bandwidth. U_i sends a *Req_Msg* to U_j, the current UAV to which c_9 is assigned. Assuming U_j sends a *Positive Resp_Msg*, it starts looking for a possible non-interfering channel for itself. U_j checks its NT and determines that c_{12} is the non-interfering channel with the lowest number and assigns the channel to itself. The newly allocated channels of U_i and U_j can be seen in Fig. 6(c). U_j's cell is denoted by blue area to represent channel reallocation.

Needing additional channels, U_i considers c_8 over c_{12} since it has a lower channel number. In the remaining part of this example, it is assumed that *Positive Resp_Msgs* are received for every *Req_Msg* sent, for the sake of simplicity. The ripple effect is larger in this case affecting two UAVs, both represented by blue areas. These reallocated channels are shown in Fig.

6(d). U_i proceeds to acquire another channel, assuming it still has a Hot Zone, by aggregating c_{12} with its current channel set. This aggregation is again essentially a bonding operation because c_{12} (new channel) and c_{11} (existing channel) are contiguous. Since c_{12} has a *cdnlty* of 2, the ripple effect is stronger, affecting 9 additional UAVs, all shown in blue areas. One of these UAVs encountered channel reallocation twice, so it is emphasized with a blue dotted area (left *nbr* of U_i). These newly allocated non-interfering channels are shown in Fig. 6(e).

The above was a simple example to explain the subcases of the reallocation scheme in the *no UAV movements* case. CA may not always result in bonding unlike in the example, especially when the CB procedure call is unsuccessful. CB/CA procedure calls are unsuccessful when a *nbr* replies with a *Negative Resp_Msg*. Moreover, it will not always be possible for an UAV to get a non-interfering channel during channel reallocation. In such a case, interference is inevitable when two or more adjacent UAVs share the same channel, negatively affecting their users' requirements. These UAVs check their NTs periodically for any possible unassigned channels in the *nbrhood*.

Number of Channels and Interference: The minimum number of channels required to avoid interference is analyzed here. First, the case of a single Hot Zone is considered which is served by U_i with a set of assigned channels, c_{set}. For further simplicity, it is assumed that there are no UAV movements. The minimum number of remaining channels necessary to avoid any interference should be at least equal to the count of U_i's 1-hop neighbors. This requirement comes from the fact that each of these 1-hop neighbors is a *nbr* of at least one of the remaining neighbors. With these minimum re-

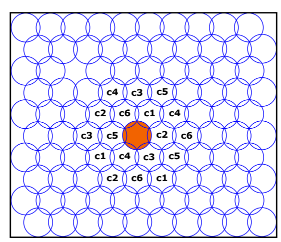


Fig. 7: Simple Case

Table 2: Simulation Parameters

Parameter	Value
Hello Interval	1 second
Initial distance between any two adjacent UAVs	$\sqrt{3} \times 100$ meters
Coverage radius of each UAV	100 meters
Initial energy of each UAV	1000×10^3 Joules
Simulation time	140 to 160 minutes
Energy consumption: Hovering/Travelling	98 Joules/second
Energy consumption: Serving per user	5 Joules/second

maining channels and the assumptions, a non-interfering channel allocation is obtained. Fig. 7 shows such an allocation to all the *nbrs* of U_i (red circle) achieved by the proposed reallocation scheme. Here, c_1-c_6 are the remaining channels, while c_7-c_N (c_{set}) are assigned to U_i. The allocation for the non-*nbrs* is not shown as they have more options due to the availability of channels in the c_{set} to them. Further, when there are multiple Hot Zones and the UAVs move, they change the topology and hence the minimum number of channels required will be higher.

Releasing Channels: An UAV releases channels when it does not need or has to replace them during reallocation. These are not given back to their previous owners to avoid interference, and become available to any UAV.

4. Performance Evaluation and Results

The simulations were performed in C++ on Ubuntu 16.04. 100 UAVs (96 *u_drones* + 4 *L_drones*), were deployed. 15 2MHz-wide L-band channels were considered with a spectral efficiency of 2.5 bps/Hz [12], thus each channel provided a data rate of 5 Mbps to users. The simulations focused on UAV - UE communication and the required UAV - UAV WiFi/LTE links for allocating L-band channels to the UAVs for serving the ground users. To generate the user traffic, 1 to 10 Hot Zones were created with uniform distribution for arbitrary durations (also uniform distribution) to exhibit randomness of user mobility and requirements in an emergency event.

Total delay, τ_d, was set to 500 μseconds [15]. The average convergence time of channel allocation in the UAV network was determined to be 7538 μseconds after running several simulation repetitions. This value is close to the expression, 14 τ_d (7000 μseconds, Table 1). The difference of 538 μseconds attributed to the wait time that the UAVs endured to receive all the *Offer_Msgs* from their corresponding 1-hop *assigned neighbors*, which was not considered in the convergence time calculations in Section 3.4.

With one L-band channel, an UAV was assumed to serve at most 50 users and experience a Hot Zone when it had more than that. A UAV started the process of channel reallocation when there were more than 55 users in its cell (threshold set to 5 users). Further, to account for CA hardware constraint, a range of 4 channels was considered: UAV with a channel, c_α, could aggregate $c_{\alpha-2}$, $c_{\alpha-1}$, $c_{\alpha+1}$ and $c_{\alpha+2}$. Since aggregating channels $c_{\alpha-1}$ or $c_{\alpha+1}$ is essentially performing CB, only two channels for CA were allowed, $c_{\alpha-2}$ and $c_{\alpha+2}$. To curb the ripple effect, the $cdnlty_{max}$ was set to 3.

It was also assumed that the energy spent in traveling a distance horizontally was same as in hovering. This was based on the laboratory experiments. To determine the energy consumed in serving the users, Raspberry Pi (users) were used to receive data at 1Mbps from the UAVs (Table 2).

The proposed channel reallocation method was compared with that of no channel reallocation. 1000 repetitions were executed for each scenario. In the latter, the UAVs were initially allocated with non-interfering channels.

Fig. 8 shows the cumulative total data served by the UAVs every 10 minutes. The total amount of data served throughout the time by all the UAVs is the value at 160 minutes. The proposed method served more data at every instant in comparison to the no reallocation scenario. With time, the UAVs started depleting their energies and after around 80 minutes most of them had no remaining energy, leaving few UAVs in the network. This is why both the plots stop growing linearly after 80 minutes because the data served by the remaining UAVs had a negligible contribution to the cumulative total amount calculated before.

The improvement in the total data served is due to the application of intelligent reallocation through CB, CA and UAV movement, which reassigned channels quickly to reduce channel interferences among the adjacent hotspot cells. For the movements, the UAVs were selected based on Eq. 4. With extensive simulations, the optimal values of the parameters, a, b, c were found to be 3.5, 2, and 0.5, respectively. The new location of an assisting UAV was determined by the equations from [1].

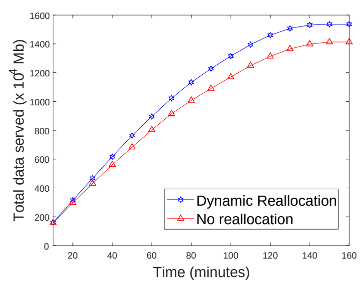


Fig. 8: Cumulative data served

Since the UAVs consume around 20 times more energy in flying (hovering) than in serving a user (Table 2), their limited flight time should be efficiently used in serving users. Reallocating channels faster and resolving interferences efficiently improves energy utilization by reducing the average discontinuity time of service. However, interferences cannot be resolved when non-interfering channels are unavailable in the *nbrhood*, particularly when there are several Hot Zones in the network. Two adjacent UAVs with a common channel transmitted data in round-robin fashion with a wait time of 30 seconds. The time when service was not provided due to channel interference was captured for every user.

The average discontinuity time was computed for all such users (who were being served by an UAV) per 10 minutes, shown in Fig. 9. This figure highlights how reassigning channels to affected UAVs minimize the interferences, reduces the discontinuity time and hence, improves the network performance by providing as much continuous service to the users as possible.

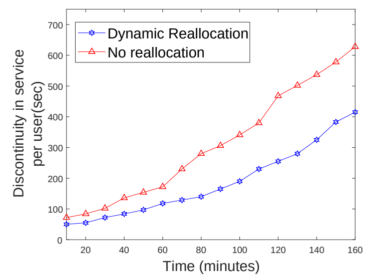


Fig. 9: Average service discontinuity time

To evaluate the efficiency of energy consumption, the number of users served per unit of depleted energy was computed every second and averaged after every 10 minutes. Fig. 10 shows that more users were served per unit of depleted energy when the proposed method was used as compared to the no reallocation scenario. This shows an improved network performance in serving users. However, the downward slope of the graphs shows a decrease in the number of users served. This is due to the accumulated effect of discontinuities in serving the users because of channel reallocations induced by interferences and the UAV movements triggered by several Hot Zones.

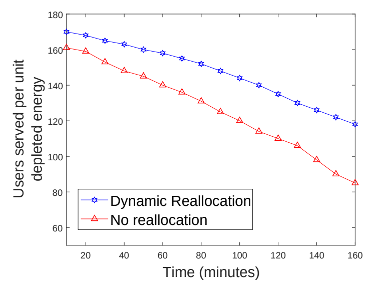


Fig. 10: Average number of users served per unit of depleted energy

Fig. 11 compares the number of times CB, CA and UAV movements occurred. The number of occurrences of CA was more than that of CB. As bondings induced by CA calls are essentially CB, hence, they were not considered for counting the CA instances. The combined contribution of CB and CA in reallocations without UAV movements was 45.5%. Prompt reassignments improve user satisfaction by reducing discontinuity in service time since no UAVs have to move. Since CB/CA have their limitations, the movement of UAVs is inevitable. Moving a UAV is a time-consuming process because it involves UAV selection, actual displacement, and the resolution of possible channel interferences due to the movement. The movements accounted for 54.5% of the reallocation contribution, which is higher than that of CB/CA (45.5%). The higher influence of physical relocation of the UAVs on the proposed solution reinforces the requirement of a self-organizing network to serve users in an emergency event.

5. Conclusion

UAV deployment and channel allocation schemes are presented to provide coverage to users as a makeshift solution where infrastructure is unavailable. UAVs are classified as uni-RAT and multi-RAT, based on RAT interfaces. Multi-RAT UAVs initiate the allocation process and are deployed effectively for a faster convergence. The convergence time and number of channels required are compared when the count of multi-RAT UAVs vary. The reallocation scheme is based on multiple channel allocation to an UAV and UAV movements. Later, this scheme is compared with the case of no channel reallocation. It is shown that the proposed scheme performs better in terms of total data transmitted, effective utilization of battery power, and lesser discontinuous service time.

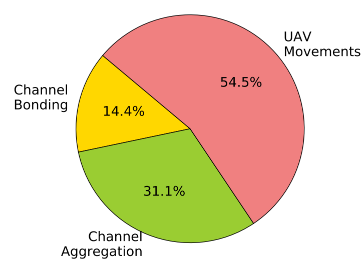


Fig. 11: Percentages of CB, CA and UAV Movements

6. References

[1] A. N. Patra, S. Sengupta, Dynamic deployment of uav-enabled floating access points for serving hot zones, in: International Symposium on Performance Evaluation of Computer and Telecommunication Systems, IEEE, 2017, pp. 1–8.

[2] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage, IEEE Communications letters 20 (8) (2016) 1647–1650.

[3] J. Lyu, Y. Zeng, R. Zhang, T. J. Lim, Placement optimization of uav-mounted mobile base stations, IEEE Communications letters 21 (3) (2017) 604–607.

[4] Y. Huo, X. Dong, T. Lu, W. Xu, M. Yuen, Distributed and multi-layer uav network for the next-generation wireless communication, arXiv preprint arXiv:1805.01534, 2018.

[5] R. de Moraes, E. de Freitas, Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services, Journal of Intelligent & Robotic Systems (2017) 1–12.

[6] D. Orfanus, E. P. de Freitas, F. Eliassen, Self-organization as a supporting paradigm for military uav relay networks, IEEE Communications letters 20 (4) (2016) 804–807.

[7] W. Wang, X. Liu, List-coloring based channel allocation for open-spectrum wireless networks, in: IEEE Vehicular Technology Conference, Vol. 1, 2005, pp. 690–694.

[8] Y. Zeng, H. Hu, T. Xu, B. Jia, User pairing stability in d2d-relay networks, IEEE Communications letters 21 (10) (2017) 2278–2281.

[9] T. Xu, M. Zhang, S. Yao, H. Hu, H.-H. Chen, Channel condition aware detection in statistical signal transmission, Transactions on Wireless Communications 16 (11) (2017) 7221–7234.

[10] ABSOLUTE (2015, Accessed October 1, 2018).
URL http://nomor.de/resources/research-projects/absolute-eu/

[11] S. H. R. Bukhari, M. H. Rehmani, S. Siraj, A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks, IEEE Communications Surveys & Tutorials 18 (2) (2016) 924–948.

[12] R. Jain, F. Templin, K.-S. Yin, Analysis of l-band digital aeronautical communication systems: L-dacs1 and l-dacs2, in: Aerospace Conference, IEEE, 2011, pp. 1–10.

[13] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, in: International Multi-Topic Conference, IEEE, 2001, pp. 62–68.

[14] M. Ma, Y. Yang, Adaptive triangular deployment algorithm for unattended mobile sensor networks, IEEE Transactions on Computers 56 (7) (2007) 946–847.

[15] Y. Xiao, J. Rosdahl, Throughput and delay limits of ieee 802.11, IEEE Communications letters 6 (8) (2002) 355–357.