Abstract—Due to the expansion of the unmanned aerial vehicle (UAV) market, there is the issue of spectrum scarcity for unmanned aerial system (UAS). Thus, it needs to allocate the resource effectively in the limited bandwidth considering the network environment. In this paper, we propose the frame structure and the resource allocation algorithm which can maximize the network throughput as well as satisfy the minimum data rate requirement. By performance analysis, we show that the proposed algorithm can allocate the resource to satisfy the high network throughput as well as the minimum data requirement in the given network environment.

Keywords—UAV; UAS; TDMA; Resource Allocation; Scheduler

I. INTRODUCTION

During the last few years, the growth of unmanned aerial vehicles (UAV) market shows a significant increase due to the various applications in civil and military areas and its increase trend will continue in next few decades. In Teal Group’s report, its analysts forecast that worldwide military and civil unmanned aerial systems (UAS) combined production will soar to $20.3 billion in 2025, up from $5.4 billion in 2016, and total $135 billion over the next 10 years [1, 2].

Although the spectral requirement for UAS is derived as mentioned earlier, there are the lack of sufficient bandwidth to allocate to the UAS communications links because not only the UAS market explosively increases, but also the existing manned aerial systems (MAV) are operated in current bandwidth, especially in the L-band [7]. In addition, the data types and requirements are different depending on the mission of the UAS. Furthermore, the environment of UAS communication links, such as the number of UAVs, the weather and the communication channel condition is changed frequently because of the high mobility of UAV [8].

Thus, we propose the frame structure dynamically choosing optimal unit timeslot and resource allocation algorithm which can maximize the network throughput as well as satisfy the minimum data rate requirement for the UAS operation in the given network environment based on Time Division Multiple Access (TDMA).

II. RELATED WORK

Before the presenting the proposed algorithm, we study the related works of two categories: TDMA-based UAS MAC protocols and other multiple access-based UAS MAC protocols.

A. TDMA-based MAC protocols for UAS

There are a lot of researches based on the TDMA MAC protocol. Jang et al. proposed a location-based TDMA MAC protocol to solve the transmission delay problems where it uses multiple UAVs environment [9]. Jang et al. proposed a new TDMA MAC protocol for the transmission delay problems. The proposed MAC protocol is used only one guard time in a frame using Piggy-backing algorithm [10]. Young et al. proposed a unifying slot assignment protocol (USAP) using distance 2 vertex-coloring problem in slot segments divided into Bootstrap, Broadcast and Reservation to avoid interference from 2-hop neighbor nodes [11]. Younis et al. proposed a cognitive USAP (C-USAP) for time slot scheduling and...
channel allocation at the same time using distributed method. A proposed protocol in [11] is used radio interference model [12].

B. Other Multiple Access-based MAC protocol for UAS

There are a lot of researches based on other multiple access-based MAC protocol. Ho et al. proposed a frame based random access (FRA) method using prioritized frame selection (PFS) method to reduce packet error rate (PER) between UAS and wireless sensor networks [13]. Temel et al. proposed a location oriented directional MAC (LODMAC) protocol using directional antennas based CSMA/CA. A proposed protocol is used Busy to Send (BTS) packet along with the Request to Send (RTS) and Clear to Send (CTS) to overcome directional deafness problem [14]. Gu et al. proposed a centralized intelligent channel assigned multiple access (C-ICAMA) combined Slotted ALOHA and TDMA for ground node to access UAV to solve the asymmetric traffic. A proposed method can dynamically allocate bandwidth for uplink and downlink using intelligent scheduling algorithm [15].

III. PROPOSED RESOURCE ALLOCATION ALGORITHM

We propose the dynamic resource allocation algorithm which determine the timeslot allocation for unmanned aerial system (UAS) based on the TDMA. In the algorithm, we determine the optimal unit timeslot size without timeslot allocation waste and then allocate the timeslot for each data type to maximize the network throughput under given network condition and data rate requirement.

First, we propose the frame structure for this algorithm as shown in Fig. 1. In this paper, we decide the frame size as 100 ms because the maximum message update rate for UAS is 10 Hz i.e. it should update messages at least once per 100 ms [6]. Then, a frame includes a number of timeslots for critical data and uncritical data, and each timeslot consists of a few unit timeslots and guard time. The size of guard time defines as about 25% of the unit timeslot size, based on [16].

The proposed resource allocation algorithm is described in Algorithm 1. The algorithm has 8 phases. In phase 1, we calculate the requirement data volume for each data type per a frame, which means that a UAV should transmit the data volume within a frame to satisfy the requirement data rate and it is given by

\[ D_i = R_i \times T_{frame}, \quad \forall i \in I. \]  

where \( D_i \) is the requirement data volume size for data type \( i \), \( R_i \) is the requirement data rate of data type \( i \), \( T_{frame} \) is the frame size, which is defined as 100 ms in this paper. Data type \( i \) means one of data types in set \( I \) and this data type set includes critical data types to operate the UAS such as uplink control message and downlink control message, in addition, the various other data types such as voice, video, and so on according the mission of UAS.

Next, in phase 2, we determine the optimal unit timeslot size. First, we choose the initial unit timeslot size \( T_{unit-fixed} \), which should be much smaller than a frame size considering the number of data types. Next, we calculate the requirement time for each data type to transmit requirement data volume \( D_i \) within a frame and it is defined as

\[ t_{req, i} = \frac{D_i}{\text{spectral efficiency} \times \text{bandwidth}}. \]  

And then, we define the unit timeslot size for data type \( i \), \( T_{unit, i} \), that it rounds \( t_{req, i} \) up to the first digit place of \( t_{req, i} \), which describes in line 5-12 of algorithm 1. Finally, we choose the minimum value of \( T_{unit-fixed} \) and \( T_{unit, i} \) as unit timeslot size, \( T_{unit} \).

After decide the unit timeslot size, in phase 3, we calculate the unit data volume per a unit timeslot, which mean that it can transmit the data volume within a unit timeslot under the network environment and it is given by

\[ D_{unit} = \frac{\text{spectral efficiency} \times \text{bandwidth}}{1/T_{unit}}. \]  

In phase 4, we calculate the allocation requirement time for each data type per a frame, which means that it needs to allocate the time to transmit the data within a frame to satisfy the requirement data rate and it is given by

\[ t_i = \left[ \frac{D_i}{D_{unit}} \right] \times T_{unit} + T_{guard}, \quad \forall i \in I, \]  

where \( t_i \) is the allocation requirement time for data type \( i \) per a frame, \( T_{guard} \) is guard time size and is defined as a quarter of \( T_{unit} \). \( \lceil x \rceil \) is ceiling function, which is the least integer greater than or equal to \( x \).
Algorithm 1 DTDMA UAV Resource Allocation

1: **Phase 1** Calculate the requirement data volume per frame
2: \[ D_i = R_i \times T_{\text{frame}}, \forall i \in I \]
3: **Phase 2** Determine the unit timeslot size
4: \[ t_{\text{req}, i} = \frac{D_i}{\text{spectral efficiency} \times \text{bandwidth}} \]
5: for \( i = 1 \) to \( I \) do
6: \( j = 0 \)
7: while \( t_{\text{req}, i} < 1 \)
8: \( t_{\text{req}, i} = t_{\text{req}, i} \times 10 \)
9: \( j = j + 1 \)
10: end while
11: \[ T_{\text{req}, i} = \left \lfloor t_{\text{req}, i} \right \rfloor / 10^j \]
12: end for
13: \[ T_{\text{unit}} = \min \left \{ T_{\text{unit, fixed}} \times \left \{ T_{\text{unit, i}} \right \}, \forall i \in I \right \} \]
14: \[ T_{\text{guard}} = T_{\text{unit}} \times 0.25 \]
15: **Phase 3** Calculate the unit data volume per unit timeslot
16: \[ D_{\text{unit}} = \frac{\text{spectral efficiency} \times \text{bandwidth}}{1/T_{\text{unit}}} \]
17: **Phase 4** Calculate the allocation requirement time per frame
18: \[ t_i = \left \lfloor \frac{D_i}{D_{\text{unit}}} \right \rfloor \times T_{\text{unit}} + T_{\text{guard}}, \forall i \in I \]
19: **Phase 5** Calculate the maximum number of UAVs to be acceptable for the critical data allocation
20: \[ n_{\text{max}} = \left \lceil \frac{T_{\text{frame}}}{\sum_{i \in I_1} t_i} \right \rceil \]
// \( I_1 \) is the set of critical data
21: **Phase 6** Decide the number of timeslot to allocate for critical data
22: \( n_i \in \{0, n_{\text{max}}\}, \forall i \in I_1 \)
23: **Phase 7** Decide the number of timeslot to allocate for uncritical data
24: if \( T_{\text{frame}} - \sum_{i \in I_1} (t_i \times n_i) < \min \left \{ t_i \mid \forall i \in I_2 \right \} \)
// \( I_1 \cup I_2 = I, I_1 \cap I_2 = \phi \)
25: \( n_i = 0 \), \forall i \in I_2
26: else
27: \[ D_{\text{rev}} = T_{\text{frame}} - \sum_{i \in I_2} (t_i \times n_i) \times \text{spectral efficiency} \times \text{bandwidth} \]
28: \( \mathbf{w} = \arg \max_{\mathbf{w} \in [0,1]} \frac{\sum_{i \in I_1} (D_i \times n_i) + \sum_{i \in I_2} (w_i \times D_{\text{rev}})}{T_{\text{frame}}} \]
29: \[ n_i = \left \lfloor w_i \times \left( T_{\text{frame}} - \sum_{i \in I_1} (t_i \times n_i) \right) / t_i \right \rceil, \forall i \in I_2 \]
30: end if
31: **Phase 8** Calculate the allocation time per frame
32: \[ T_i = t_i \times n_i, \forall i \in I \]

In phase 5, we calculate the maximum number of UAVs to be acceptable when it can only allocate the timeslot for critical data types such as control messages and it is defined as
\[ n_{\text{max}} = \left \lceil \frac{T_{\text{frame}}}{\sum_{i \in I_1} t_i} \right \rceil, \]

where \( n_{\text{max}} \) is the maximum number of acceptable UASs and \( I_1 \) is the set of critical data type. The critical data type means that it is positively necessary to operate the UAS and this data type varies according to the purpose of UAS. \( \left \lfloor x \right \rfloor \) is floor function, which is the greatest integer less than or equal to \( x \).

In phase 6, we decide the number of timeslots to allocate for each critical data type, which is not larger than \( n_{\text{max}} \).

In phase 7, \( n_i \) is the number of timeslots for data type \( i \), \( I_1 \) is the set of critical data type. The union of \( I_1 \) and \( I_2 \) is the set of critical data type. The critical data type means that it is positively necessary to operate the UAS and this data type varies according to the purpose of UAS. \( \left \lfloor x \right \rfloor \) is floor function, which is the greatest integer less than or equal to \( x \).

In phase 8, we allocate the timeslot according to the weighting factor of each data type to maximize the network throughput. To find this weighting factor, we calculate the data volume which is able to transmit during the remaining time and it is given by
\[ D_{\text{rev}} = \left \lceil T_{\text{frame}} - \sum_{i \in I_2} (t_i \times n_i) \right \rceil \times \text{spectral efficiency} \times \text{bandwidth}. \]

where \( D_{\text{rev}} \) is the transmission-available data volume within the remaining time. And then, we find the optimal weighting factor set to maximize the network throughput as follows,
\[ w = \arg \max_{w_i \in [0,1]} \sum_{i \in I_1} (D_i \times n_i) + \sum_{i \in I_2} (w_i \times D_{cen}) \bigg/ T_{frame}, \]  

where \( w \) is the optimal weighting factor set, \( w = \{w_i\}, \forall i \in I_1 \), and total sum for the elements of \( w \) is 1. Based the optimal weighting factor, we calculate the number of timeslot to allocate for uncritical data and it is defined as

\[ n_i = \left\lfloor w_i \times \left( T_{frame} - \sum_{i \in I_1} (t_i \times n_i) \right) / t_i \right\rfloor, \forall i \in I_2. \]  

Finally, in phase 8, we calculate the allocation time for each data type per frame and it is given by

\[ T_i = t_i \times n_i, \forall i \in I_1. \]  

where \( T_i \) is the allocation time for data type \( i \).

IV. PERFORMANCE ANALYSIS

To evaluate the performance of the proposed algorithm, we perform the simulation using MATLAB. We compare the proposed algorithm with the algorithm which has 1 ms fixed unit timeslot size. The performance analysis environment refers to [17] and [6], and set them as Environment 1 and Environment 2. The bandwidth sets 2 MHz in Environment 1 and 6 MHz in Environment 2. There are lower than the required bandwidth of [17] and [6] in order to make the assumption that the required bandwidth is insufficient. The other parameters are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Environment 1</th>
<th>Environment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulation</td>
<td>QPSK</td>
<td>OQPSK</td>
</tr>
<tr>
<td>Code rate</td>
<td>0.646</td>
<td>Uncoded</td>
</tr>
<tr>
<td>Spectral efficiency</td>
<td>1.287</td>
<td>0.88</td>
</tr>
<tr>
<td>Bandwidth (MHz)</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirement data rate (bps)</th>
<th>Uplink control</th>
<th>Uplink control</th>
<th>Downlink control</th>
<th>Downlink control</th>
<th>Voice</th>
<th>Voice</th>
<th>Video</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment 1</td>
<td>30.794 k</td>
<td>Uplink control</td>
<td>44.734 k</td>
<td>Downlink control</td>
<td>27.7 k</td>
<td>Voice</td>
<td>270 k</td>
<td>Video</td>
</tr>
<tr>
<td>Environment 2</td>
<td>44.734 k</td>
<td>Uplink control</td>
<td>28.774</td>
<td>Downlink control</td>
<td>27.7 k</td>
<td>Voice</td>
<td>270 k</td>
<td>Video</td>
</tr>
</tbody>
</table>

Table 2 is the result of the maximum number of UAVs to be acceptable for the critical data allocation (\( n_{\text{max}} \)) which are resulted from Phase 5 in the Algorithm 1. The legend of ‘Dynamic’ refers to the proposed resource allocation algorithm and the legend of ‘Fixed’ refers to the resource allocation algorithm which has 1 ms fixed unit timeslot size.

Table 3 is the result of the average data rate per UAV. The uncritical data (Voice, Video) may not be allocated due to the optimal weighting combination. So the values are averaged out only when data is allocated. We have confirmed that the values are above the required data rate.

In Fig. 2, we show the total network throughput according to the number of UAVs in Environment 1. In Environment 1, the total network throughput is almost same in ‘Dynamic’ and ‘Fixed’, because there’s no redundancy time in unit timeslot. So, throughput may be increased by selecting the large value of unit timeslot.

In Fig. 3, we show the total network throughput according to the number of UAVs in Environment 2. In Environment 2, the total network throughput which case of ‘Dynamic’ is higher than ‘Fixed’ when the number of UAVs is increased, because the size of the unit timeslot is fixed largely so that data can be transmitted within a unit timeslot due to the data rate of uplink control and voice is low in Environment 2. That is, the redundancy time occurs until next unit timeslot when use ‘Fixed’. Thus, the throughput can be increased by selecting the unit timeslot for the required data rate.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Dynamic</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uplink Control</td>
<td>38,435</td>
<td>38,409</td>
</tr>
<tr>
<td>Downlink Control</td>
<td>50,460</td>
<td>50,373</td>
</tr>
<tr>
<td>Voice</td>
<td>34,633</td>
<td>34,546</td>
</tr>
<tr>
<td>Video</td>
<td>282,502</td>
<td>282,515</td>
</tr>
<tr>
<td>Environment 1 (bps)</td>
<td>38,435</td>
<td>38,409</td>
</tr>
<tr>
<td>Downlink Control</td>
<td>50,460</td>
<td>50,373</td>
</tr>
<tr>
<td>Voice</td>
<td>34,633</td>
<td>34,546</td>
</tr>
<tr>
<td>Video</td>
<td>282,502</td>
<td>282,515</td>
</tr>
<tr>
<td>Environment 2 (bps)</td>
<td>5,263</td>
<td>6,228</td>
</tr>
<tr>
<td>Downlink Control</td>
<td>31,384</td>
<td>35,961</td>
</tr>
<tr>
<td>Voice</td>
<td>5,230</td>
<td>5,986</td>
</tr>
<tr>
<td>Video</td>
<td>1,003 k</td>
<td>1,003 k</td>
</tr>
</tbody>
</table>

TABLE II. THE MAXIMUM NUMBER OF UAVS

<table>
<thead>
<tr>
<th>Environment</th>
<th>Dynamic</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment 1</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>Environment 2</td>
<td>133</td>
<td>40</td>
</tr>
</tbody>
</table>

TABLE III. AVERAGE DATA RATE PER UAV
V. CONCLUSION

We propose the new frame structure based on the dynamic selection unit timeslot and TDMA resource allocation algorithm in given network environment, which can maximize network throughput while satisfy the minimum data rate requirement of UAS. In performance analysis, the network throughput of proposed algorithm is higher than the algorithm using fixed unit timeslot while satisfy the minimum data rate requirement in given network environment.

ACKNOWLEDGMENT

This work has been supported by the Future Combat System Network Technology Research Center program of Defense Acquisition Program Administration and Agency for Defense Development (UD160070BD).

REFERENCES


388