

Scheduling Design and Performance Analysis of Carrier Aggregation in Satellite Communication Systems

Hayder Al-Hraishawi^{1b}, *Member, IEEE*, Nicola Maturo^{1b}, *Member, IEEE*, Eva Lagunas^{1b}, *Senior Member, IEEE*, and Symeon Chatzinotas^{1b}, *Senior Member, IEEE*

Abstract—Carrier Aggregation is one of the vital approaches to achieve several orders of magnitude increase in peak data rates. While carrier aggregation benefits have been extensively studied in cellular networks, its application to satellite systems has not been thoroughly explored yet. Carrier aggregation can offer an enhanced and more consistent quality of service for users throughout the satellite coverage via combining multiple carriers, utilizing the unused capacity at other carriers, and enabling effective interference management. Furthermore, carrier aggregation can be a prominent solution to address the issue of the spatially heterogeneous satellite traffic demand. This paper investigates introducing carrier aggregation to satellite systems from a link layer perspective. Deployment of carrier aggregation in satellite systems with the combination of multiple carriers that have different characteristics requires effective scheduling schemes for reliable communications. To this end, a novel load balancing scheduling algorithm has been proposed to distribute data packets across the aggregated carriers based on channel capacities and to utilize spectrum efficiently. Moreover, in order to ensure that the received data packets are delivered without perturbing the original transmission order, a perceptive scheduling algorithm has been developed that takes into consideration channel properties along with the instantaneous available resources at the aggregated carriers. The proposed modifications have been carefully designed to make carrier aggregation transparent above the medium access control (MAC) layer. Additionally, the complexity analysis of the proposed algorithms has been conducted in terms of the computational loads. Simulation results are provided to validate our analysis, demonstrate the design tradeoffs, and to highlight the potentials of carrier aggregation applied to satellite communication systems.

Index Terms—Carrier aggregation, load balancing, packet scheduling, scheduling algorithms, satellite communication.

Manuscript received November 30, 2020; revised January 27, 2021 and March 14, 2021; accepted June 1, 2021. Date of publication June 29, 2021; date of current version August 13, 2021. This work was supported in part by Luxembourg National Research Fund (FNR) under the project FlexSAT C19/IS/13696663 and the European Space Agency (ESA) funded activity CADSAT: Carrier Aggregation in Satellite Communication Networks. The views of the authors of this article do not necessarily reflect the views of FNR and ESA. This work in part has been presented at IEEE International Conference on Communications (ICC), 2020 [1]. The review of this article was coordinated by Prof. Riku Jantti. (*Corresponding author: Hayder Al-Hraishawi.*)

The authors are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg (e-mail: hayder.al-hraishawi@uni.lu; nicola.maturo@uni.lu; eva.lagunas@uni.lu; symeon.chatzinotas@uni.lu).

Digital Object Identifier 10.1109/TVT.2021.3093117

I. INTRODUCTION

THE demand of data traffic on satellite systems is skyrocketing, and data usage is expected to continue increasing for the foreseeable future. Satellite communication is witnessing this outgrowth of the traffic demand due to its seamless high speed connectivity and ubiquitous broadband coverage [2]. The ability to provide telecommunication services in a wide range of sectors such as aeronautical, maritime, military, rescue and disaster relief has led to a dramatic increase in the demand for satellite data traffic [3], [4]. Moreover, in many emerging applications for future wireless systems such as interactive multimedia services, distributed IoT networks, and content delivery networks (CDNs), satellite technology can support this expansion, and thus, contribute to 5G and beyond ecosystems toward highly reliable networks with global mobile coverage[5]. Specifically, 5G and beyond systems are mostly about high-speed backhaul connectivity to the in-motion terminals on airplanes, vehicles, trains, and vessels over large coverage areas. However, this goal cannot be achieved by only developing the terrestrial networks owing to the fact that high-speed and multicast-enabled satellite links, direct to the airplane, vehicles, train, or vessel, will complement existing terrestrial connectivity [6].

The advantage of satellite systems in provisioning a wide coverage and far-reaching access to a large number of various users makes the traffic demands more heterogeneous and spatially distributed. The flexibility to adapt to such spatial diversified traffic demands is a crucial requirement for future broadband satellite systems [7], and the resilience in assigning the constrained satellite resources is essential to satisfy the uneven demands [8]. Many important contributions to the developments of flexible resource allocation have been conducted in this context. For instance, an optimal dynamic capacity allocation scheme is investigated in [9] by utilizing smart gateway diversity setup and considering users' requested and gateways' offered capacities to minimize system capacity losses and improve rate matching performance.

The fundamental satellite network parameters such as uplink and downlink antenna gains, user receiving gains and noise temperature, path losses, and data rates have been jointly optimized in [10] to increase system resource utilization. The concept of beam-hopping has been particularly studied in [11]–[13] to harness the benefits of flexible system architecture in order to

fulfill the inconstant traffic demands over time and geographical locations. Beam-hopping system allows sharing in time, power, and frequency resources among multiple beams to offer higher usable throughput. Reference [14] studies system resource allocation in the forward link of multibeam satellite networks and proposes an algorithm for resource allocation with objective of satisfying the requested traffic across different beams with taking fairness into consideration. A dynamic on-board signal processing scheme in a multiple gateway multibeam satellite system has been designed in [15] to render flexible resource allocation, although this architecture imposes high complexity to the satellite payload.

Furthermore, carrier aggregation technique has been introduced to terrestrial networks in Long Term Evolution-Advance (LTE-A) standard to allow multiple component carriers across the available spectrum bands to be flexibly aggregated to support wider transmission bandwidth, and thus, increasing the overall system capacity [16]. Carrier aggregation in terrestrial networks has achieved a considerable enhancement in performance through maximizing the spectrum utilization and satisfying the extremely high throughput requirements [17]. Carrier aggregation does not only efficiently exploit the available spectrum but it also maintains users' quality of service through effective interference management and avoidance capabilities [18]. Therefore, a natural step would be combining carrier aggregation with satellite system architectures in synergy to harness the multiplexing gain by distributing the traffic dynamically over multiple carriers.

In heterogeneous wireless networks, traffic aggregation is set to play an essential role in leveraging radio resources to improve user peak data rate and quality of service [19]. However, traffic aggregation solutions at the upper layers, e.g., transport and application layers, may not be very efficient in terms of performance. The lack of instantaneous channel information at these layers makes them incompetent under variable channel conditions [20], [21]. In contrast, several designing insights have been drawn from other works indicating that aggregation schemes at the link layer accomplish a major performance enhancement owing to the availability of feedback instantaneous channel information [22]. For instance, in LTE protocol structure, carrier aggregation is essentially handled by the medium access control (MAC) entity that is responsible for distributing data packets across the component carriers based on a specific scheduling approach, where it can achieve an overall wider bandwidth and correspondingly higher per-link data rates [23]. Therefore, this work aims at developing a traffic scheduler operates at the link layer to maximize the resource utilization and improve peak data rates.

Notwithstanding that deployment of carrier aggregation technique in terrestrial networks has been widely investigated, its integration into satellite communication systems has not attracted a worthwhile attention in academia. Meanwhile, the European Space Agency (ESA) has funded a project titled as CADSAT [24] that deals with introducing carrier aggregation to satellite systems, where multiple potential scenarios have been extensively explored and analyzed based on market, business, and technical feasibility standpoints. The works in [25] and [26] focus on designing an optimal carrier-user assignment scheme

for carrier aggregation in a multibeam Geostationary Earth Orbit (GEO) satellite system. Specifically, both inter-transponder and intra-transponder carrier aggregation have been considered at the satellite payload level of the communication stack to address the difficulty of carrier-user assignment in an environment of multiple users that can be multiplexed in each carrier. Reference [27] devises a carrier aggregation scheme for satisfying the non-uniform user traffic demands across the system, and thus, improving the overall spectral resource utilization of satellite systems. Beyond this, the authors' work in [1] and its extension in this paper focus on load balancing and scheduling design at the gateway side.

On the other hand, channel bonding for satellite systems has been already deployed and defined in DVB-S2X standard [28] as a scheme where a single data stream is transmitted over different RF channels through different transponders lying in the same frequency band. In contrast, carrier aggregation refers to simultaneously combining multiple contiguous and non-contiguous carriers in different spectrum bands to constitute a larger transmission bandwidth. Comparing to carrier aggregation, channel bonding has numerous inherent limitations for broadband applications that would hinder resource allocation flexibility. Moreover, channel bonding utilizes constant modulation and coding (MODCOD) schemes, where all the services employ same coding and modulation procedures, which is a very deterrent factor for its deploying in future broadband applications. These limitations have motivated this work to consider carrier aggregation in order to jointly enhance system throughput and flexibility.

Introducing carrier aggregation to satellite systems requires the resource allocation module to be more flexible while scheduling radio resources for users. Besides, the assignment of carriers has to be tailored with available resources and channel characteristics of each user because it has a significant impact on network performance. Specifically, within the MAC layer, a scheduler has to be carefully designed for load balancing and Protocol Data Unit (PDU) scheduling among the aggregated carriers. More importantly, an appropriate PDU scheduling algorithm is a prime part of carrier aggregation mechanism that takes into consideration the quality of service requirements of each user and attains a certain level of fairness, and hence, leads to an enhancement in the achievable system capacity and service performance.

This paper focuses on establishing an architecture with a detailed design for embedding carrier aggregation in satellite systems from a link layer viewpoint. In other words, we will demonstrate the main essential blocks that enable carrier aggregation, namely the PDU scheduler at the gateway, and the traffic merging unit at the receiver. The developed structure is mainly aiming at focusing the implementation efforts on the gateway side, so that the user terminal stays as simple as possible with minimum changes required to support carrier aggregation. To this end, the traffic merging block at the user terminal is simply a First-In First-Out (FIFO) system. More importantly, the proposed modifications at the link layer is making carrier aggregation invisible to the upper layers, i.e., no further modifications are needed as the output packet streams will be identical to the non-aggregation case. Hence, it is of extremely importance that

the PDU scheduler module at the gateway side is perceptively distributing the incoming PDUs among the aggregated carriers to ensure that they can be easily merged in a single stream with a simple FIFO buffer such that the reconstructed PDUs at the user terminal are in the correct order with no missing PDU. Therefore, we will mainly concentrate our investigating and designing efforts in this work on a scenario where a certain user is aggregating two carriers, and then, the extension for including more carriers/users can be readily conducted based on these guidelines.

Contributions: Our technical contributions can be explicitly summarized as follows

- 1) Design guidelines are provided for deploying carrier aggregation technique into satellite communication systems. Specifically, the proposed modifications at the link layer of the system architecture have been thoroughly analyzed and efficiently designed to make carrier aggregation transparent to the upper layers, and thus, no further stringent adjustments are needed.
- 2) Novel load balancing scheduling algorithms have been proposed to distribute data packets efficiently across the aggregated carriers based on their capacities. Specifically, load balancing algorithms for two carriers and three carriers aggregation scenarios are elaborated, and their tightness to the load balancing theoretical model is also evaluated.
- 3) Aiming at delivering data packets without perturbing their original transmission order while keeping user terminal design simple, a perceptive scheduler has been jointly developed alongside with the load balancing method through taking into account not only channel characteristics but also the instant accessible resources of the aggregated carriers. Thus, in addition to achieving intelligent packet distribution, this procedure avoids any ordering problem that may occur at the receiver.
- 4) Deploying schedulers for carrier aggregation inherently introduces complexity and delay variation. Therefore, the complexity of the proposed algorithms is analyzed in terms of the computational loads, and the additional complexity of the established scheduling block within the other link layer entities is quantified as well.
- 5) The performance of the proposed scheduling algorithms has been investigated in terms of the achievable peak data rate and in-order delivery success rate. Simulation results including some performance comparisons are provided to demonstrate the validity and gains of the proposed algorithms.

The rest of the paper is structured as follows. The system model is presented in Section II. Next, a detailed description and presentation of the proposed scheduling algorithms are provided in Section III, along with their complexity analyses. Numerical examples and demonstrations of various simulation results are given in Section IV. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL

We consider the forward link of a broadband multibeam GEO satellite system that employs multi-carrier transponders. In this

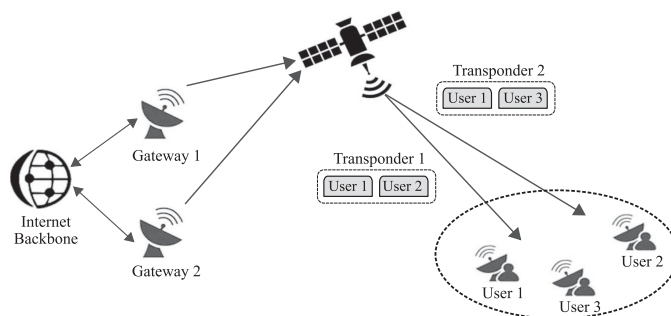


Fig. 1. Schematic diagram of intra-beam carrier aggregation in a multi-carrier satellite system

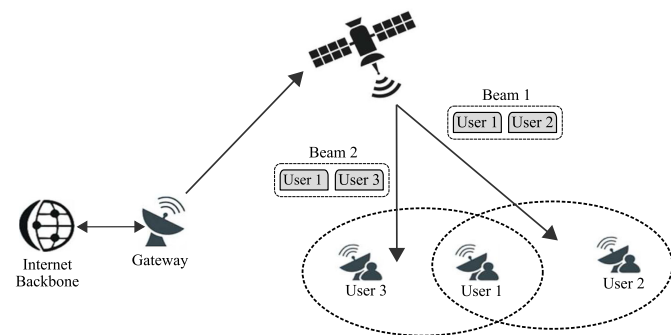


Fig. 2. Schematic diagram of inter-beam carrier aggregation in a multibeam satellite system.

setting, carrier aggregation can be applied for both intra-beam and inter-beam scenarios. More precisely, in the use case of enabling carrier aggregation within a single beam employing multiple carriers that cover the same region, a user may aggregate carriers across or within transponders that might originate from the same or different gateways as depicted in Fig. 1. Load balancing between transponders in a single-beam can be achieved via carrier aggregation to leverage the underutilized spectrum and offer extended bandwidth to the users with high demand. In this context, the transponders are assumed to be coexisted for covering the same region, which can be done either through Frequency Division Multiplexing (FDM) or using two polarizations (e.g., horizontal and vertical polarizations) in the same bandwidth.

Similarly, in multibeam systems, inter-beam carrier aggregation provides flexibility to adapt to heterogeneous traffic demands, namely congestion on the so-called hot beams (high demand beams) can be relieved by utilizing resources from the neighboring cold beams (low demand beams), which usually happens at the beams' edges as illustrated in Fig. 2. Thus, carrier aggregation can boost the peak data rates for the edge users, and hence, provisioning a steady quality of service over the entire coverage area. Additionally, this scenario is particularly relevant to the aeronautical mobile satellite services as it can ensure the continuity of service without degradation while traveling through the beams' edges. In this, a 4-color reuse frequency scheme is considered, that is allowing frequency reuse with minimal interference between neighboring spot beams. It is worth mentioning that in a fully loaded network, carrier aggregation may offer a very limited advantage because offloading can hardly increase the sum capacity of the system.

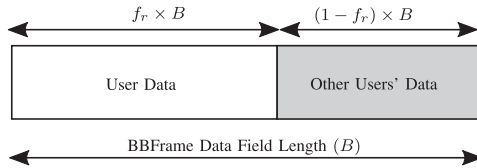


Fig. 3. User fill rate representation in BBFrame structure.

Furthermore, the fluctuation in satellite channel conditions has been incorporated in this work through employing the Digital Video Broadcasting-Satellite (DVB-S) (DVB-S2 and DVB-S2X) standardization, which are used by most satellite operators worldwide. These standards are very flexible and covering a variety of satellite applications. They are also characterized by several important features such as a flexible input stream adapter that is suitable for operation with single and multiple input streams of various formats. Additionally, a wide range of code rates and constellations that are optimized for operation over non-linear transponders are also adopted in DVB-S2 standards alongside with a set of multiple spectrum shapes for different roll-off factors [28].

Satellite communications impose different constraints compared to terrestrial systems in terms of attenuation, propagation delays, fading, etc., namely satellite channels are characterized by excessive propagation delays and intense fading phenomena. Thus, long Forward Error Correction (FEC) based on LDPC (Low-Density Parity Check) codes concatenated with BCH codes and fading mitigation techniques have been employed based on DVB-S2 and DVB-S2X standards to address these channel limitations. These techniques essentially rely on an adaptive link layer design that is conducted by providing each user with the most suitable MODCOD scheme according to the link Signal-to-Noise Ratio (SNR). In this context, the Adaptive Coding and Modulation (ACM) provides efficient channel protection and dynamic link adaptation to different propagation conditions, targeting each individual receiving terminal. In the proposed integration scheme of carrier aggregation into satellite systems, ACM technique has been considered to cope with the long-distance fading and long-transmission delay of satellite channels.

The focus of this work is to design the two main critical blocks that are required to deploy carrier aggregation technique in satellite systems, namely packet (PDU) scheduler block at the gateway and traffic merging procedure at the user terminal. For this objective, the process of establishing carrier aggregation in a single user is thoroughly and elaborately analyzed while the presence of other users within the same Base Band Frames (BBFrames) is not neglected, where zero-padding is applied to fill up the remaining part of BBFrame data fields. To this end, we have introduced a parameter that determines the percentage of the BBFrame data field that can be utilized by the intended user for transmission, which is called fill rate and denoted as f_r . Fig. 3 depicts an illustration for the fill rate in a BBFrame.

In this system, two carriers are aggregated to provide a wider bandwidth for a single user. Fig. 4 shows the changes required at the MAC layer protocol so as to ensure compatibility with DVB-S2 and DVB-S2X specifications. As the link layer is the

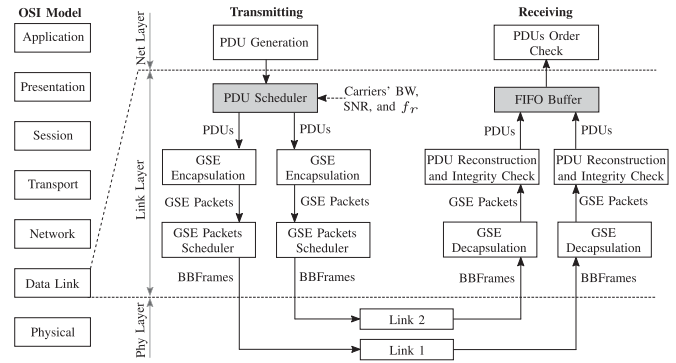


Fig. 4. Carrier aggregation link layer design.

core of this system and for the purpose of keeping upper layers intact and not aware of the carrier aggregation, the functionality of network layer at the transmitter side is simply represented by a PDU generator. Hence, the introduced PDU scheduler block takes the generated PDUs as input together with the parameters of the aggregated carriers, i.e. bandwidth (BW) and SNR, to bring forth two separated PDU streams, one for each carrier. The SNR here accounts for additive noise, as well as the radiated signal strength and the attenuation throughout the channel and the system. In addition to BW and SNR, fill rate factor is also taken into consideration as another input parameter to the PDU scheduler because it reflects the actual available capacity offered to the user by each carrier individually, and thus, the dimension of the Generic Stream Encapsulation (GSE) packets will be changed accordingly.

The function of PDU scheduler is to distribute the incoming PDUs from the upper layer between the two carriers according to their channel characteristics. Afterwards, each PDU flow goes to a typical GSE encapsulation block. In this function, PDUs are encapsulated in GSE packets and according to their dimension, they could be encapsulated in a single GSE packet or sliced into PDU fragments and encapsulated in several GSE packets. Since the maximum dimension of a GSE packet is limited to 4KB, while the maximum dimension of a PDU can be up to 65536 bytes [29], PDU fragmentation might be required before the encapsulation. The encapsulated packets for each carrier are then placed together to create two streams of BBFrames. Whereas, at the link layer of the receiver side, conventional GSE decapsulation is performed and then the PDUs are reconstructed independently in both links. PDU reconstruction includes de-fragmentation of the segmented PDUs at the transmitter side. The integrity of the reconstructed PDUs is also checked at the link layer using a CRC (Cyclic Redundancy Check) code. Then, the two streams of the reconstructed PDUs are merged at the FIFO buffer and passed to the network layer.

III. PROPOSED SCHEDULING ALGORITHMS

In this section, the concept of load balancing between the aggregated carriers is first described in brief. Next, the proposed load balance scheduling algorithms for two carriers and three carriers aggregation are presented, respectively. Thereafter, a perceptive scheduler design is introduced to avoid the potential pitfalls of the load balancing schedulers and to obtain the best

possible link adaptation. Finally, the added complexity and computational load of the proposed scheduling algorithms are analyzed.

A. Load Balancing Scheduler

In this subsection, a load balancing scheduler that distributes the incoming PDUs across the aggregated carriers is proposed based on channel conditions and the available resources. To this end, the following algorithms are developed for efficient and fair scheduling that takes into account capacity and fill rate of the considered carriers. To begin with, let us define the ratio between capacities and fill rates of the aggregated carriers as the *load balancing factor* (α) for the case of aggregating two carriers, which can be calculated as follows:

$$\alpha = \frac{C_2 f_{r,2}}{C_1 f_{r,1}}, \quad \text{where } C_1 f_{r,1} \geq C_2 f_{r,2} \quad (1)$$

where C_i and $f_{r,i}$ account for the capacity and fill rate of the i -th carrier, respectively, and $i \in \{1, 2\}$. Basically, carrier capacity is determined by the bandwidth, modulation order, and code rate, which are obtained based on the SNR of the carrier and in accordance with DVB-S2 and DVB-S2X standards, specifically Table (13) in [30] and Table (20a) in [31]. Both DVB-S2 and DVB-S2X use several MODCOD schemes to take advantage of the variable transmission conditions of each carrier, and therefore they feature different capacity and transmission time with each MODCOD.

In order to expedite the scheduling process and suppress any incurred delay due to PDUs assignment, the scheduling is done by using a look-up table or an *allocation table* that contains a mapping of carrier allocation sequences and its corresponding load balancing factor (α) values. The length of the allocation table mainly depends on the selected granularity (β) of α . Thus, based on a range of α values with a reasonable granularity, the allocation table can be generated beforehand and stored. Then, these carrier allocation patterns are ultimately used to schedule the incoming PDUs between the aggregated carriers. The allocation table can be dynamically modified based on the application, which gives a degree of resilience to the system. The procedure to generate the allocation table, when aggregating two carriers, is presented in **Algorithm 1**.

Fundamentally, the conception behind the load balancing algorithm is simply to prioritize allocating more PDUs to the high capacity carrier and vice versa. The inputs to the proposed algorithm are the number of PDUs (μ) that need to be scheduled, and the granularity (β) of the load balancing factor (α). Then, two similar sequences of length μ , $\mathbf{v}_1 = \alpha\{1, 2, \dots, \mu\}$ and $\mathbf{v}_2 = \{1, 2, \dots, \mu\}$ are generated that correspond the first and second carriers, respectively. The vector \mathbf{v}_1 is multiplied by α , where $0 < \alpha \leq 1$, in order to favour the carrier with the higher capacity and fill rate. Next, these two vectors are merged to produce $\mathbf{w} = [\mathbf{v}_1 \mathbf{v}_2]$, which will be later sorted in an ascending order. At the first duplicate values in vector $[\mathbf{w}]$ the sequence will be cut and stored in new vector $[\mathbf{a}]$. All elements in $[\mathbf{a}]$ are tracked and labeled according to their originating carrier.

Algorithm 1: Generating the Allocation Table for Two Carriers Aggregation.

Input: $\alpha = 0.01$
 $\beta = 0.01$ (Granularity of α)
 $\mu = \text{Number of PDUs}$

- 1 Generate $\mathbf{v}_2 = 1 : 1 : \mu$
- 2 **while** $\alpha \leq 1$ **do**
- 3 Generate $\mathbf{v}_1 = \alpha : \alpha : \alpha\mu$
- 4 $\mathbf{w} = [\mathbf{v}_1 \mathbf{v}_2]$ Merge the two vectors
- 5 Sort $[\mathbf{w}]$ Ascendingly
- 6 Find j , index of the first duplicate values in $[\mathbf{w}]$
- 7 $\mathbf{a} = \mathbf{w}(1 : j + 1)$
- 8 $\mathbf{s} = [\alpha, \mathbf{a}]$ Store *Allocation Sequence* with α
- 9 $\alpha = \alpha + \beta$
- 10 **end**

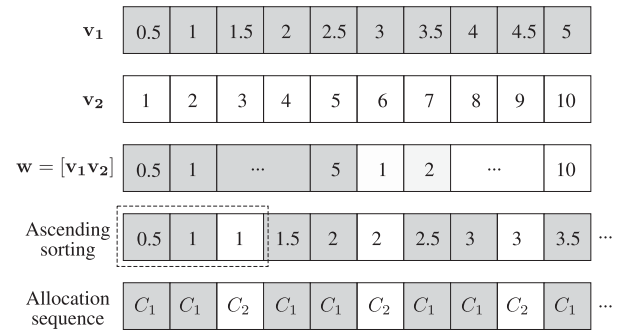


Fig. 5. Example of generating carrier allocation sequence based on load balancing algorithm.

Finally, the produced allocation sequence $[\mathbf{a}]$ is collected with its α value, and then the same steps are repeated for $\alpha = \alpha + \beta$.

To elaborate more about Algorithm 1, a numerical example is given in the following. Assume that there are two carriers (C_1 and C_2) to be aggregated and their load balancing factor is $\alpha = 0.5$, namely, the first carrier offers twice the capacity of the second carrier. Thus, for a number of PDUs equals to 10, \mathbf{v}_1 and \mathbf{v}_2 can be created as shown in Fig. 5. Next, these two vectors are combined to make vector \mathbf{w} that will be sorted ascendingly in the next step. Obviously, the elements in the sorted sequence provide the allocation order to be followed for carrier assignment, which is $[C_1, C_1, C_2]$. Specifically, it corresponds to sending a single PDU through the second carrier for every two PDUs sent through the first carrier.

To further extend the design of the load balancing algorithm to deal with more than two carriers, we can follow the aforementioned methodology that we used for designing the two carriers case, but we will have multiple load balancing factors (e.g. $\alpha_1, \alpha_2, \dots$ etc.), instead of only one factor in the case of aggregating two carriers. Specifically, the number of load balancing factors equals to the number of aggregated carriers (N_C) minus one. The carrier with highest capacity among other carriers will be the reference point to calculate the first load balancing factor (α_1) with respect to the other slower carriers, and the carrier with the higher capacity after the first carrier will be the next reference

Algorithm 2: Generating the Allocation Table for Three Carriers Aggregation.

Input: $\alpha_1 = 0.01$ and $\alpha_2 = 0.01$
 $\beta = 0.01$ (Granularity of α_1 and α_2)
 $\mu =$ Number of PDUs

```

1 Generate  $\mathbf{v}_2 = 1 : 1 : \mu$ 
2 while  $\alpha_1 \leq 1$  do
3   Generate  $\mathbf{v}_1 = \alpha_1 : \alpha_1 : \alpha_1 \mu$ 
4    $\mathbf{w}_1 = [\mathbf{v}_1 \mathbf{v}_2]$  Merge the two generated vectors
5   Sort  $[\mathbf{w}_1]$  Ascendingly
6   Find  $j$ , index of the first duplicate values in  $[\mathbf{w}_1]$ 
7    $\mathbf{a} = \mathbf{w}_1(1 : j + 1)$ 
8    $\mathbf{a} = \mathbf{a}(\text{mod}(0 : \mu - 1, \text{length}(\mathbf{a})) + 1)$ 
9   while  $\alpha_2 \leq 1$  do
10    Generate  $\mathbf{v}_3 = \alpha_2 : \alpha_2 : \alpha_2 \mu$ 
11    Generate  $\mathbf{v}_4 = 1 : 1 : \mu$ 
12     $\mathbf{w}_2 = [\mathbf{v}_3 \mathbf{v}_4]$  Merge the two vectors
13    Sort  $[\mathbf{w}_2]$  Ascendingly
14    Find  $j$ , index of first duplicate values in  $[\mathbf{w}_2]$ 
15     $\mathbf{b} = \mathbf{w}_2(1 : j + 1)$ 
16     $\mathbf{b} = \mathbf{b}(\text{mod}(0 : \mu - 1, \text{length}(\mathbf{b})) + 1)$ 
17     $m = 1$ 
18    for  $k = 1 : \mu$  do
19      if  $\mathbf{a}(k) \neq 1$  then
20        if  $\mathbf{a}(k) \neq \mathbf{b}(m)$  then
21           $\mathbf{a}(k) = \mathbf{b}(m)$ 
22        end
23         $m = m + 1$ 
24      end
25    end
26     $\mathbf{s} = [\alpha_1, \alpha_2, \mathbf{a}]$  Store Allocation Sequence with
       $\alpha_1$  and  $\alpha_2$ 
27     $\alpha_2 = \alpha_2 + \beta$ 
28  end
29   $\alpha_1 = \alpha_1 + \beta$ 
30 end

```

point to calculate the next load balancing factor (α_2) with the rest of the slower carriers, and so forth. For instance, in case of aggregating three carriers ($N_C = 3$) the number load balancing factors is two, which can be calculated as follows

$$\alpha_1 = \frac{C_2 f_{r,2} + C_3 f_{r,3}}{C_1 f_{r,1}}, \quad \text{and} \quad \alpha_2 = \frac{C_3 f_{r,3}}{C_2 f_{r,2}}$$

where $C_1 f_{r,1} \geq C_2 f_{r,2} \geq C_3 f_{r,3}$. (2)

Moreover, according to the findings of [26], the maximum number of carriers that can be aggregated to achieve an optimal performance is three. Therefore, generating the allocation table for three carrier aggregation case is given in **Algorithm 2**.

Theoretically, to fully utilize the available resources in the aggregated carriers and realize an ideal load balancing, the number of PDUs that should be scheduled over each carrier has been calculated by the theoretical load balancing model as follows

[32].

$$\eta_i = \text{round} \left(\frac{C_i f_{r,i}}{\sum_{j=1}^{N_C} C_j f_{r,j}} \mu \right), \quad (3)$$

where η_i represents the number of the assigned PDUs to the i -th carrier, and N_C accounts for the number of aggregated carriers. To validate the load balancing algorithms, a comparison between the theoretical model and the proposed algorithms will be illustrated in performance evaluation section.

B. Perceptive Scheduler

Intuitively, scheduling PDUs to their corresponding carriers is not necessarily reflecting their transmission order in the BBFrame because they will be fragmented/encapsulated according to the assigned carrier capacity and fill rate. For instance, when a carrier has more capacity that allows to convey two PDUs or more in one BBFrame while the other carrier has space only for one PDU in a BBFrame that will bring about an ordering problem at the receiver side. Therefore, BBFrame length and the instant available space in each BBFrame should be taken into consideration in the scheduling process. To this end, we have further investigated such cases and developed a supplementary procedure to track the allocation process that is done by the load balancing scheduler. The proposed procedure is conducting in advance comparison between the PDU length, the occupied bits in current BBFrame of each carrier, and the number of transmitted BBFrames in both carriers. Practically, this scheduler can minimize packet reordering at the receiver side and improve the performance of transport protocols, namely it can enhance communication reliability when using User Datagram Protocol (UDP) and reduce network latency in Transmission Control Protocol (TCP).

Our basic idea is to incorporate the tradeoff between channel capacity and the instantaneous available resources when allocating a PDU to a carrier, and hence, do the scheduling and link adaptation collectively. In this context, BBFrame transmission is carrying out in a dynamic manner, where it has to constantly check the BBFrame creation process of the two carriers in a parallel manner to verify if any further adjustment to the allocation sequence is required. Particularly, three factors are organizing and deciding this precise scheduling, (i) length of PDU, (ii) count of the transmitted BBFrames, and (iii) the remaining bits at the current BBFrame. The developed perceptive scheduling procedure is detailed in **Algorithm 3**, where the occupied space in each BBFrame (ψ_i) is updating after every transmitted PDU.

In **Algorithm 3**, λ symbolizes the length of the PDUs in bits, and BBFrame useful bits account for the usable payload that a user can exploit for data transmission after eliminating overheads. There are two counters tracking the number of transmitted BBFrames in each carrier and they are represented by Q_1 and Q_2 , respectively. To prioritize the carrier that has sufficient space in its current BBFrame for the upcoming PDU, the factor ψ_i is introduced in order to keep under observation the occupied space in both BBFrames. This algorithm first considers the incipient phase when both carriers have equal numbers of transmitted BBFrames ($Q_1 = Q_2$), afterwards, the subsequent phases are

Algorithm 3: Perceptive Allocation for Two Carriers Aggregation.

Input: λ = PDU length
 B_1 = BBFrame1 useful bits
 B_2 = BBFrame2 useful bits
 $Q_1 = 1$, First carrier BBFrame counter
 $Q_2 = 1$, Second carrier BBframe counter
 $\psi_1 = 0$, occupied bits in BBFrame1
 $\psi_2 = 0$, occupied bits in BBFrame2
 $j = 1$

```

1 while  $j \leq \mu$  do
2   if  $Q_1 = Q_2$  then
3     if  $B_1 - \psi_1 > \lambda$  and  $B_2 - \psi_2 > \lambda$  then
4       if  $B_1 - \psi_1 \geq B_2 - \psi_2$  then
5         Allocate PDU( $j$ ) to carrier 1
6       else
7         Allocate PDU( $j$ ) to carrier 2
8       end
9     else if  $B_1 - \psi_1 > \lambda$  and  $B_2 - \psi_2 < \lambda$  then
10      Allocate PDU( $j$ ) to carrier 1
11    else if  $B_1 - \psi_1 < \lambda$  and  $B_2 - \psi_2 > \lambda$  then
12      Allocate PDU( $j$ ) to carrier 2
13    else
14      if  $B_1 - \psi_1 \geq B_2 - \psi_2$  then
15        Allocate PDU( $j$ ) to carrier 1
16      else
17        Allocate PDU( $j$ ) to carrier 2
18      end
19    end
20  else if  $Q_1 > Q_2$  then
21    if  $2B_2 - \psi_2 < \lambda$  then
22      Allocate PDU( $j$ ) to carrier 1
23    else if  $\psi_1 + \lambda \geq \psi_2 + \lambda$  then
24      Allocate PDU( $j$ ) to carrier 2
25    else
26      Allocate PDU( $j$ ) to carrier 1
27    end
28  end
29   $j = j + 1, \psi_i = \psi_i + \lambda$ 
30 end

```

addressed by either the case of $Q_1 > Q_2$ or $Q_1 = Q_2$. The case of $Q_1 < Q_2$ is disregarded here because of the presumption of considering the first carrier always has a higher combination of capacity and fill rate than the second carrier ($C_1 f_{r,1} \geq C_2 f_{r,2}$).

To illustrate the scheduling decision process in the perceptive algorithm, the following basic examples are provided as shown in Fig. 6. Specifically, we have a PDU of length λ that must be transmitted either through carrier 1 or carrier 2. First, the initial phase is considered when $Q_1 = Q_2$. In the first case (a), the scheduler checks the space availability in both carriers' BBFrames and learns that both carriers have enough space to carry this PDU, i.e., $B_1 - \psi_1 > \lambda$ and $B_2 - \psi_2 > \lambda$. Since the first carrier has more unoccupied space in its current BBFrame than the second carrier, then this PDU is allocated to the first

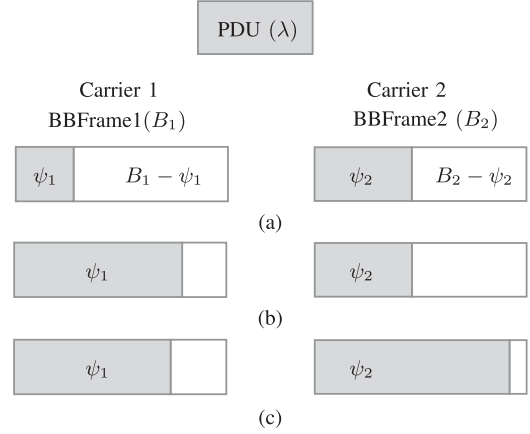


Fig. 6. Examples to illustrate the mechanism of the perceptive scheduling algorithm.

carrier. Next example in case (b) clearly shows that the second carrier has to handle this incoming PDU due to the higher unoccupied capacity, which fits the entire length of the PDU within its current BBFrame. In the third case (c), both carriers have smaller unoccupied spaces than λ , thus the decision will be to send it through the first carrier because it has larger available space ($B_1 - \psi_1$) than the second carrier ($B_2 - \psi_2$). Second, when more PDUs are already transmitted through the first carrier than the second carrier ($Q_1 > Q_2$), the decision for case (c) will be different than the previous scenario. Since both current BBFrames cannot fully handle the entire PDU, then the priority now is for the second carrier as long as it can be entirely sent over the next BBFrame.

C. Complexity Analysis

The proposed design modifications in this work to enable carrier aggregation in satellite communication systems are entirely restricted within the link layer, which limits the added complexity to the system architecture. However, the essential block to support multiple carriers is the packet scheduler that inherently adds a layer of complexity and introduces some delay variations. Therefore, the complexity of the proposed scheduling algorithms is analyzed in terms of their computational loads. Specifically, packet allocation with load balancing scheduling algorithm is executed using a look-up table, which makes its contribution to the processing load is very marginal comparing to other link layer functions such as PDU fragmentation and GSE encapsulation. Nevertheless, generating the allocation sequences that are constituting the allocation table is obtained from Algorithms 1 and 2 with a time complexity depends on the number of aggregated carriers, namely $\mathcal{O}(n^{N_C-1})$. The complexity of the load balancing algorithms 1 and 2 are $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively.

Moreover, the complexity of the perceptive scheduling Algorithm 3 is very low as it is just a supplementary procedure added to the load balancing algorithm to carry out in advance comparison between PDU lengths, occupied bits in carriers' BBFrames, and numbers of transmitted BBFrames in both carriers. This comparison takes insignificant processing time and

its impact on the complexity of the scheduling algorithm is minimal. Additionally, the computation of the perceptive scheduler speeds up when considering balanced carriers ($\alpha \rightarrow 1$) for aggregation because the load balancing scheduler causes less ordering problems in such cases, and then a large reduction can be expected in the number of calls to the perceptive procedure. Therefore, the complexity of the proposed algorithms is fairly low. In the next section, more experiments will be run to quantify the additional complexity of the developed schedulers by testing various settings with the consideration of balanced and unbalanced carriers.

IV. PERFORMANCE EVALUATION

This section presents the simulation results. We investigate the validity of the proposed algorithms and evaluate their performances using a software-based demonstrator laboratory testbed that is delineated in [33]. This demonstrator is *Matlab* experimental platform built based on the conventional GSE protocol as detailed in [29]. Specifically, a GSE encapsulation function is constructed at the transmitter side as it performs PDU fragmentation and encapsulation to constitute the GSE packets. Afterwards, a GSE packet scheduler function is implemented to put together the completed GSE packets based on the available space in each BBFrame data field of the aggregated carriers. At the receiver side, a GSE decapsulation function is implemented to process the received BBFrames and extract the GSE packets. These extracted GSE packets are passed to another function that is responsible for PDU reconstruction at the receiver side, where in case of fragmented PDUs, the CRC and all the elements needed to recalculate the CRC are passed as well in order to check the integrity of the reconstructed PDUs.

In this section, the performance of the proposed carrier aggregation scheme will be evaluated in terms of the achievable peak data rate and compared to the conventional channel bonding technique in satellite systems. Next, both load balancing algorithms for two carriers and three carriers aggregation will be validated by examining their tightness to the theoretical load balancing model. Further, the performance of the proposed scheduling algorithms in terms of delivering PDUs without perturbing their original transmission order will be compared to a traditional scheduler. Finally, the complexity of each proposed scheduling algorithm is quantified.

In Fig. 7, the achievable peak data rate of a user terminal utilizing three different carriers is scrutinized. The peak data rate is plotted against the SNR (γ_1) of the first carrier, whereas the other carriers have lower SNRs. The second and third carriers have $\gamma_2 = \gamma_1 - 3$ dB and $\gamma_3 = \gamma_1 - 5$ dB, respectively. For the sake of comparison, we consider all carriers have similar bandwidth and fill rate, namely, $BW = 10$ MHz and $f_r = 50\%$. Fig. 7 shows comparisons between carrier aggregation and channel bonding from the achievable peak data rate viewpoint. The channel bonding is simulated here based on the specifications of DVB-S2X standard [28]. It can be readily seen that carrier aggregation outperforms channel bonding in both cases of combining two carriers ($N_C = 2$) and three carriers ($N_C = 3$). Interestingly, in the useful SNR regime lower than 10 dB, aggregating two

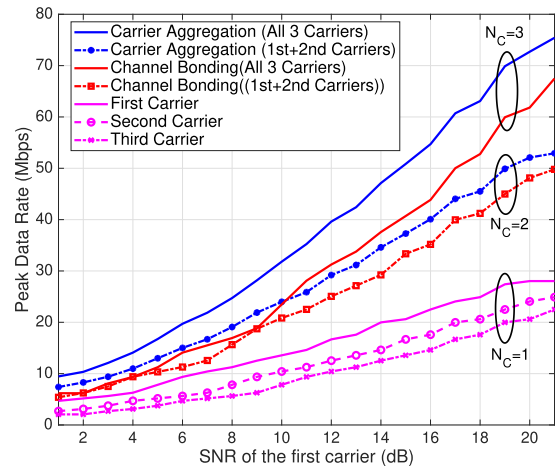


Fig. 7. The achievable peak data rate versus SNR. A comparison between single carrier, channel bonding, and carrier aggregation.

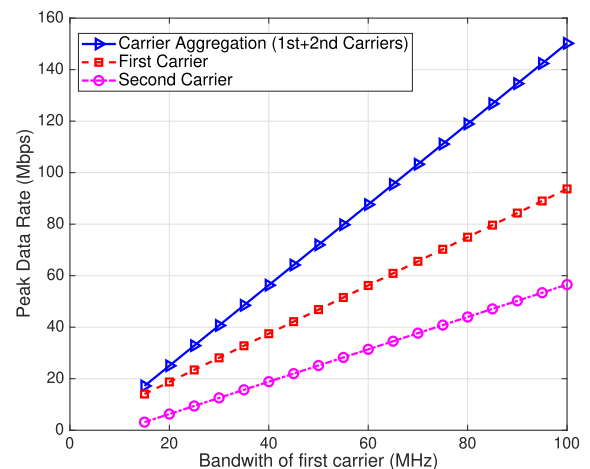


Fig. 8. The achievable peak data rate versus bandwidth. A comparison between single carrier and carrier aggregation.

carriers achieves a slightly higher peak data rate than bonding three carriers.

In Fig. 8, the peak data rates of two carriers having different bandwidths are plotted against bandwidth of the first carrier, and then compared to the achievable peak data rate of aggregating both carriers. The second carrier has a 5 MHz lower bandwidth than the first carrier $BW_2 = BW_1 - 5$ MHz, whereas the bandwidth of the first carrier increases as $BW_1 = 15 : 5 : 100$ MHz. For the sake of comparison, all other parameters of the carriers are fixed, namely $\gamma_1 = 6$ dB, $\gamma_2 = 4$ dB, and $f_{r,1} = f_{r,2} = 50\%$. Obviously, aggregating different carriers offers a higher operational bandwidth through creating larger ‘virtual’ carrier bandwidths for satellite services, which will achieve a higher peak data rate. This makes carrier aggregation technique as an essential approach to meet the growing satellite traffic demand in a well-timed manner.

The performance of carrier aggregation with respect to fill rate is investigated by considering two carriers, one allocates fixed resources to the investigated user while the second carrier offers an increasing space in the BBFrame data field that can be utilized

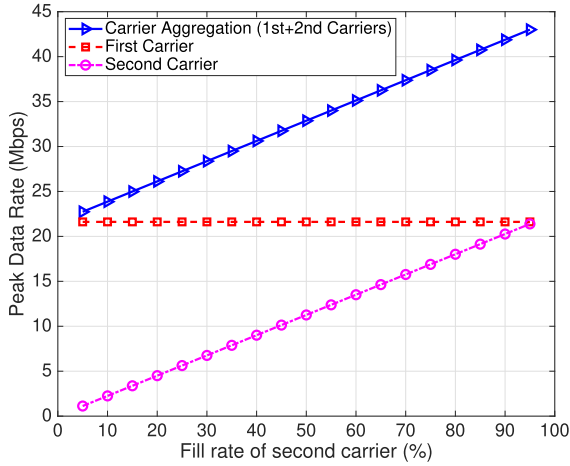


Fig. 9. The achievable peak data rate versus fill rate. A comparison between single carrier and carrier aggregation.

for data transmission, which is represented by the fill rate parameter. Specifically, the parameters of first carrier are $BW_1 = 12$ MHz, $\gamma_1 = 8$ dB, and $f_{r,1} = 80\%$, while the second carrier has $BW_2 = 12$ MHz, $\gamma_2 = 8$ dB, and $f_{r,2} = [5 : 5 : 95]\%$. The peak data rates of these carriers individually along with result of aggregating them are plotted versus $f_{r,2}$ in Fig. 9. Fill rate, as earlier defined in this paper, is the parameter that determines the percentage of the BBFrame data field in a carrier that can be utilized by a user for carrier aggregation. Accordingly, the cases of $f_r = 0\%$ and $f_r = 100\%$ mean that there is no space in that carrier to be utilized for carrier aggregation and the user is fully assigned to that carrier and carrier aggregation cannot be applied, respectively. Clearly, the relationship between fill rate and peak data rate is linear when assuming all other carrier parameters are invariant. It can be noticed that carrier aggregation always achieves a higher peak data rate than a single carrier. Beyond this, it also reveals that spectrum efficiency can be improved through resource sharing between radio channels. In short, the observations about the peak data rate that can be accomplished when employing carrier aggregation validate the discussion that is given in Section I about the benefits of deploying carrier aggregation in satellite communications. Hence, our analysis and designing insights in this paper can be useful as a benchmark for performance comparison purposes with other practical data rate enhancement techniques in high throughput satellite systems.

Next, the performance of the proposed load balancing scheduler in Algorithm 1 is evaluated in Fig. 10 by considering two unbalanced carriers to be aggregated. The numbers of allocated PDUs to each carrier (η_1 and η_2) are plotted versus the load balancing factor (α). Theoretical curves are obtained using equation (3), and the curves of the load balancing scheduling are calculated using the sequences in the generated allocation table for each corresponding α value. Here we consider a stream of PDUs ($\mu = 200$) to be distributed across carrier 1 and carrier 2 of bandwidths 50 MHz and 40 MHz, respectively, while the SNR of the first carrier is 10 dB and of the second carrier is 8 dB. Further, carriers' fill rates are setup as fixed ($f_{r,1} = 70\%$) for the first carrier, whilst in the second carrier varies as $f_{r,2} = [10 : 10 : 100]\%$

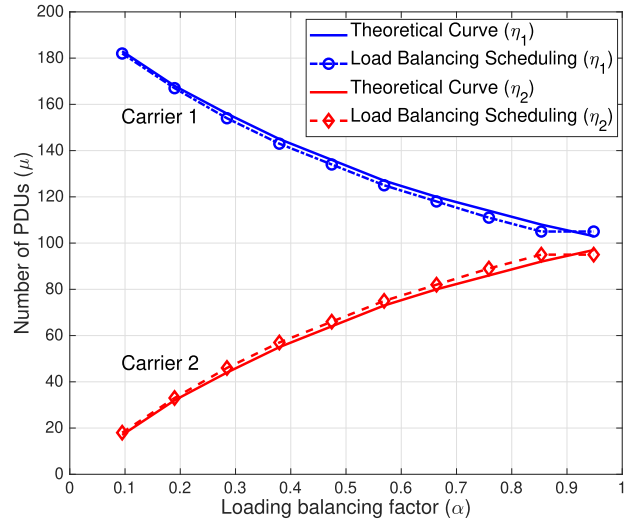


Fig. 10. Validation of load balancing algorithm for $N_C = 2$.

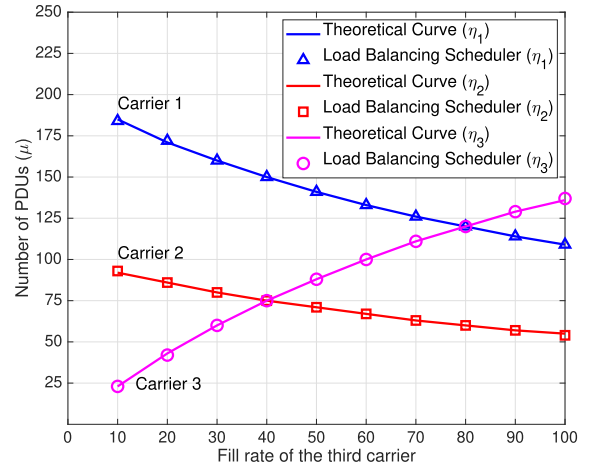


Fig. 11. Validation of load balancing algorithm for $N_C = 3$.

to attain the shown α ranging. Fig. 10 reveals that when α increases the difference between η_1 and η_2 decreases till the balance point when $\alpha = 1$ where both carriers convey same number of PDUs, i.e., $\eta_1 = \eta_2$. More importantly, we can clearly observe that our algorithm is significantly tight to the theoretical distribution. Additionally, the proposed algorithm provides the actual chronological sequence for transmission.

In Fig. 11, Algorithm 2 is evaluated by comparing its tightness with the theoretical load-balancing formula in (3). To this end, we consider a data stream consists of 300 PDUs (i.e., $\mu = 300$) to be scheduled over three aggregated carriers that have the following parameters: $BW_1 = BW_2 = BW_3 = 40$ MHz, $\gamma_1 = \gamma_2 = \gamma_3 = 10$ dB, and fill rates of the first and second carriers are fixed as $f_{r,1} = 80\%$ and $f_{r,2} = 40\%$, while the third carrier has a variable fill rate varies as $f_{r,3} = [10 : 10 : 100]\%$. The number of the allocated PDUs per each carrier versus the variable parameter $f_{r,3}$ is plotted in Fig. 11. Clearly, the proposed algorithm is very tight to the theoretical distribution, which corroborates its validity. Moreover, it can be clearly seen that the number of allocated PDUs to the third carrier is monotonically increasing with fill rate. Additionally, the curves of second and

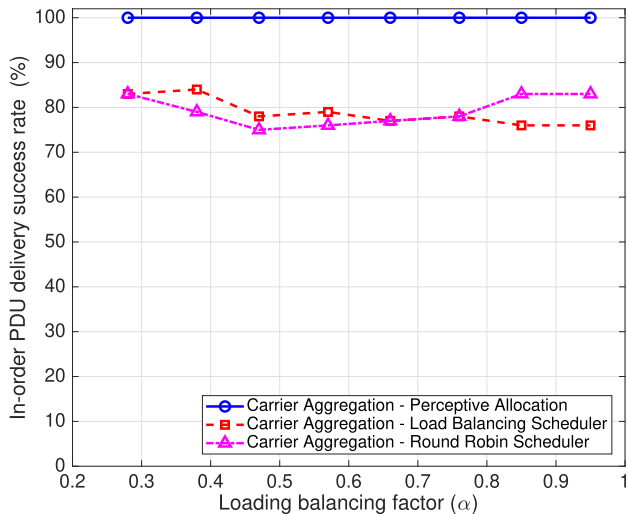


Fig. 12. In-order PDU delivery success rate versus load balancing factor (α). A comparison between perceptive allocation, load balancing, and Round Robin schedulers.

third carriers are intersected at $f_{r,3} = 40\%$ that equals to $f_{r,2}$, which means that they process same number of PDUs ($\eta_2 = \eta_3$). Similar observation can be noticed when first and third carriers have identical parameters at $f_{r,3} = 80\%$ leading them to convey same number of PDUs ($\eta_1 = \eta_3$).

The performance of the developed schedulers is evaluated for delivering the transmitted PDUs in a correct order. Two metrics are used here, namely the in-order delivery success rate and the number of out-of-order PDUs, the former is calculated as the ratio of the in-order received PDUs to the total number of the transmitted PDUs. In addition to the proposed schedulers, we consider the traditional Round-Robin scheduler [34] for comparison purposes. Fig. 12 presents the success rate curves of delivering in-order PDUs versus α by employing the aforementioned schedulers. Two carriers are considered with bandwidths 50 MHz and 40 MHz, respectively, and their SNRs are 10 dB and 8 dB, respectively. Further, we fixed the fill rate of the first carrier to 70%, while the fill rate of the second carrier is ranging between $f_{r,2} = [30, 100]\%$. A stream of 100 PDUs of length 1400 bytes are processed through the implemented simulator.

Fig. 12 clearly reveals that our perceptive scheduler in Algorithm (3) for carrier aggregation achieves the highest success rate with zero out-of-order PDU. It can also be seen that however the load balancing scheduler distributes PDUs efficiently between the considered carriers but it is not enough by itself to deliver them in-order at the receiver side. Similarly, Round-Robin scheduler cannot guarantee the order of the received PDUs unless both carriers are balanced because it is neither considering the available spectrum resources nor tracking the BBFrame creation process. Further, the occurrence of the ordering problems when using Round Robin and load balancing schedulers is completely different because they consider different allocation schemes. Since Round Robin uses a fixed allocation sequence, the ordering problems occur continuously in a repetitive pattern, whereas ordering errors with the load balancing method behaves

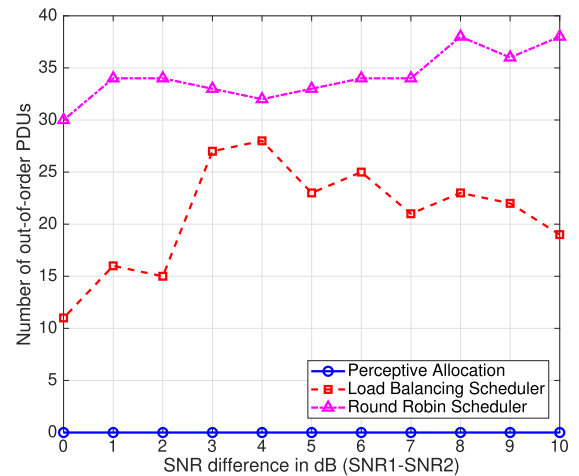


Fig. 13. Number of out-of-order PDUs versus the SNR difference between carriers. A comparison between perceptive allocation, load balancing, and Round Robin schedulers.

more arbitrary because the allocation sequences are influenced by the carrier parameters' combinations. Thus, both schedulers show a clear variation in their packet delivery curves even for the same carrier parameters. Consequently, the proposed perceptive scheduling algorithm can serve as an efficient mean for carrier aggregation in designing practical satellite systems.

Furthermore, performance of the three schedulers in terms of the number of out-of-order PDUs are evaluated versus some parameters other than alpha. For instance, the SNR difference between the aggregated carriers while all other parameters are fixed. Specifically, we consider 100 PDUs ($\mu = 100$), each of length 1200 bytes, and two carriers with equal bandwidths $BW_1 = BW_2 = 10$ MHz, and different fill rates $f_{r,1} = 60\%$ and $f_{r,2} = 50\%$, respectively. Meanwhile, the SNR of first carrier increases as $\gamma_1 = [2 : 1 : 12]$ dB and the second carrier is on steady condition with SNR $\gamma_2 = 2$ dB. The PDUs are transmitted when both carriers are aggregated and scheduling is performed by using Round Robin, load balancing, and perceptive allocation. The number of out-of-order PDUs are examined at the PDUs order check block at the receiver side. The obtained results are plotted in Fig. 13 against the difference between carriers' SNRs. Clearly, perceptive scheduler gives a steady and perfect performance comparing to other schedulers. Moreover, Round-Robin performance degrades with the growth of SNR difference.

Fig. 14 investigates the number of out-of-order PDU versus carrier fill rate while all other parameters are fixed. Performance of the studied schedulers are evaluated under the following parameters. A stream of PDUs $\mu = 200$ with a length of each is 1300 bytes is considered for communication through two different carriers. Specifically, carriers' bandwidth are $BW_1 = 12$ and $BW_2 = 10$ MHz, respectively, and their SNRs are assumed to be constant as $\gamma_1 = 10$ dB and $\gamma_1 = 7$ dB, respectively. Fill rate of the first carrier is $f_{r,1} = 50\%$, while it varies for second carrier as $f_{r,2} = [30 : 5 : 65]\%$. These combinations of carrier parameters produce an incremental load balancing factor ranging from $\alpha = 0.46$ to $\alpha = 1$ corresponds to the growth of $f_{r,2}$. Fig. 14

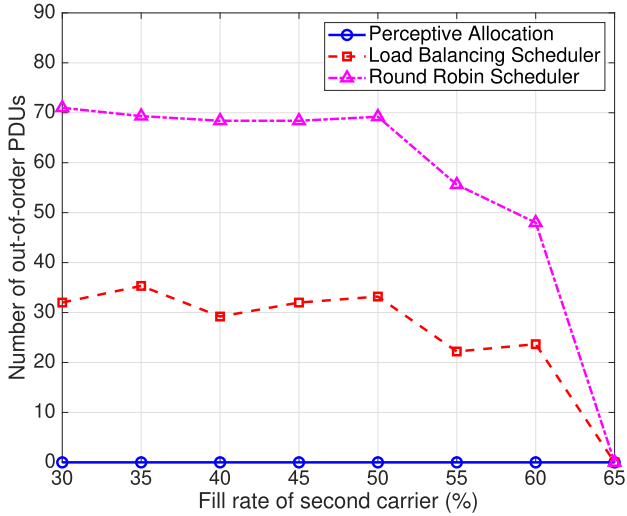


Fig. 14. Number of out-of-order PDUs versus carrier's fill rate. A comparison between perceptive allocation, load balancing, and Round Robin schedulers.

depicts the obtained results, where it can be clearly seen that perceptive scheduling method outperforms other schedulers, and load balancing algorithm is exceeding the performance of traditional Round Robin scheduler. However, a gradual performance enhancement can be noticed when fill rate increases. This incremental fill rate moves the aggregated carriers from being unbalanced to the condition of being perfectly balanced when $\alpha = 1$, which makes the incompetent schedulers perform well.

In order to quantify the complexity of the proposed algorithms and illustrate the differences in terms of the processing time, the execution time of the established PDU scheduler block (shown in Fig. 4) has been tracked and measured by the *profile* report function of Matlab. Specifically, we have tested 300 random carrier parameters' combinations and extracted the time consumed by the scheduler block solely when carrier aggregation is applied, first by using load balancing and next when the perceptive allocation is performed. These experiments were conducted on an average laptop PC (Dell Latitude 5490, Intel Core™ i7-8650U CPU @ 1.90GHz, 16GB RAM), using Matlab R2020a. The obtained results are averaged with respect to each unique α value and plotted in Fig. 15. Interestingly, the report shows that the load balancing takes an average of 0.08% of the whole processing time of transmitting and receiving 100 PDUs, while the perceptive scheduler takes an average of 0.26%. Thus, enabling carrier aggregation slightly increases the processing time but still it is a fair tradeoff comparing with the achieved gain in terms of peak data rate. The extra time complexity of the perceptive scheduler is resulting from tracking the BBFrame creation process. It can be readily seen that when the aggregated carriers are balanced ($\alpha \rightarrow 1$), the computation of the perceptive scheduler decreases where the load balancing scheduler causes less ordering problems.

Next, processing time of transmitting and receiving a stream of PDUs ($\mu = 200$), with maximum length for each PDU (1500 bytes), is measured under two different scenarios for comparison purposes. First, when the communication is conducted through a single carrier has a variable capacity, namely $BW_1 = 10$ MHz,

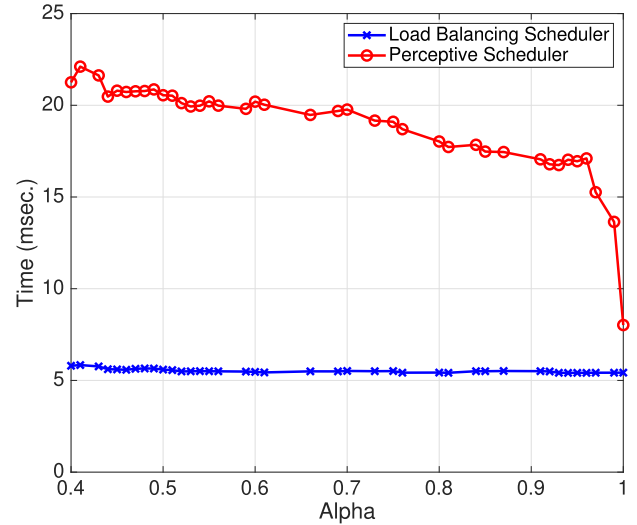


Fig. 15. Execution time of the PDU scheduler block when using load balancing and perceptive allocation algorithms versus load balancing factor (α).

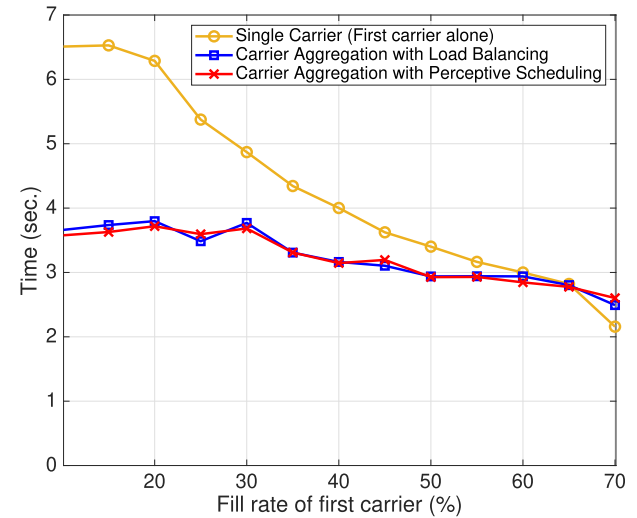


Fig. 16. Overall processing time versus carrier's fill rate. A comparison between single carrier communication and carrier aggregation when using load balancing and perceptive schedulers.

$\gamma_1 = 8$ dB, and $f_{r,1} = [20 : 5 : 80]\%$. Second, this single carrier is aggregated with another fixed carrier with a larger capacity ($BW_2 = 10$ MHz, $\gamma_2 = 8$ dB, and $f_{r,2} = 80\%$) and employed for communicating the same stream of PDUs. The processing time for both cases are analyzed and plotted versus the fill rate of the first carrier in Fig. 16. Surprisingly, the single carrier case takes a longer communication time comparing with carrier aggregation for the same PDU stream due to the limited capacity offered by this carrier to convey these PDUs. Specifically, when the PDU size is greater than the available BBFrame useful bits, the PDU will be further fragmented into smaller segments, which eventually adds computational loads and delays to perform the additional GSE encapsulation/decapsulation and PDU fragmentation/defragmentation. Meanwhile, the processing time decreases when the capacity of the first carrier starts to increase, i.e., $f_{r,1} > 60\%$. Therefore, enabling carrier aggregation for

TABLE I
PERFORMANCE SUMMARY OF ROUND ROBIN, LOAD BALANCING, AND
PERCEPTIVE SCHEDULING ALGORITHMS

Metric / Scheduler	Round Robin	Load balancing	Perceptive scheduling
Scheduling criteria	Blind distribution	Channel capacity	Channel capacity and BBFrame size
Spectral efficiency	Not efficient	Efficient	Efficient
Performance with balanced carriers	Bad (unless perfectly balanced)	Better	Superior
Performance with unbalanced carriers	Bad	Good	Superior
Computational complexity	Very low	Low	Low
Applications with low tolerance to out-of-order delivery	Not suitable	Not suitable	Suitable

satellite communications will not only improve system capacity but it also can help alleviating complexity and queuing through offloading the congested carriers, and then providing a seamless user experience.

Table I summarizes the pros and cons of Round Robin, load balancing, and perceptive scheduling approaches for carrier aggregation. On one hand, Round Robin and load balancing can be used when the transport layer protocol considers UDP with high tolerance to out-of-order PDUs, although load balancing is better in terms of the spectral utilization. On the other hand, the perceptive approach can be considered as a feasible option for reliable and efficient communications because it is capable of avoiding the delays from calling TCP to reassemble the received packets in the correct order.

V. CONCLUSIONS

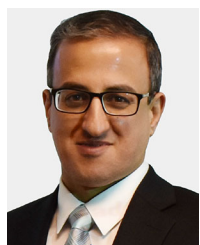
This paper constructed a new link layer structure to integrate carrier aggregation technique into satellite systems without disturbing the architecture of the upper layers. In other words, carrier aggregation is invisible to the other layers in view of that the entire set of aggregated carriers can be seen as a single carrier communications. In link layer entity, data packet scheduling algorithms are developed to efficiently utilize the considered carriers. The proposed load balancing scheduler allocates the incoming packets based on link capacity and fill rate of the aggregated carriers. This scheduler provides fair allocation to the transmitted packets and efficiently utilizes carriers' resources but it brings about an ordering issue at the receiver side. Thus, it can be beneficial for some applications that have high tolerance to out-of-order delivery. To address the ordering challenge, a perceptive scheduling algorithm has been developed that incorporates the tradeoff between channel capacity and the instantaneous availability of the spectral resources. The latter scheduler outperforms the proposed load balancing allocation scheme and the traditional Round-Robin scheduler in terms of in-order packet delivery. Moreover, the added complexity due to enabling carrier aggregation is marginal comparing to single carrier communication and it is still a fair tradeoff when looking

at the achievable peak data rate. Finally, carrier aggregation helps alleviating complexity and queuing delays through offloading the congested carriers, and thus providing an ameliorated user experience.

REFERENCES

- [1] H. Al-Hraishawi, N. Maturo, E. Lagunas, and S. Chatzinotas, "Perceptive packet scheduling for carrier aggregation in satellite communication systems," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2020, pp. 1–6.
- [2] S. H. Chae, C. Jeong, and K. Lee, "Cooperative communication for cognitive satellite networks," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5140–5154, Nov. 2018.
- [3] F. Li, K. Lam, N. Zhao, X. Liu, K. Zhao, and L. Wang, "Spectrum trading for satellite communication systems with dynamic bargaining," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4680–4693, Oct. 2018.
- [4] H. Al-Hraishawi, E. Lagunas, and S. Chatzinotas, "Traffic simulator for multibeam satellite communication systems," in *Proc. 10th Advanced Satellite Multimedia Syst. Conf. and 16th Signal Process. Space Commun. Workshop*, 2020, pp. 1–8.
- [5] M. Bacco, P. Cassarà, M. Colucci, and A. Gotta, "Modeling reliable M2M/IoT traffic over random access satellite links in non-saturated conditions," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1042–1051, May 2018.
- [6] A. Guidotti *et al.*, "Architectures and key technical challenges for 5G systems incorporating satellites," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2624–2639, Mar. 2019.
- [7] Y. Gao, X. Yang, X. Geng, J. Li, and J. Wu, "A multi-band carrier aggregation scheme for mobile satellite communications," *Appl. Mech. Mater.*, vol. 548–549, pp. 1282–1285, Apr. 2014.
- [8] J. Du, C. Jiang, H. Zhang, Y. Ren, and M. Guizani, "Auction design and analysis for SDN-based traffic offloading in hybrid satellite-terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2202–2217, Oct. 2018.
- [9] A. J. Roumeliotis, C. I. Kourogiorgas, and A. D. Panagopoulos, "Optimal dynamic capacity allocation for high throughput satellite communications systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 596–599, Apr. 2019.
- [10] J. J. Knab, "Optimization of commercial satellite transponders and terminals," *IEEE Trans. Aeronaut. Navig. Electron.*, vol. 49, no. 1, pp. 617–622, Jan. 2013.
- [11] C. Rohde *et al.*, "Beam-hopping systems for next-generation satellite communication systems," in *Satellite Commun. 5G Era*. Institution of Engineering and Technology, 2018, ch. 10, pp. 277–305.
- [12] V. Joroughi *et al.*, "Deploying joint beam hopping and precoding in multi-beam satellite networks with time variant traffic," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Anaheim, CA, USA, 2018, pp. 1081–1085.
- [13] L. Lei, E. Lagunas, Y. Yuan, M. Kibria, S. Chatzinotas, and B. Ottersten, "Deep learning for beam hopping in multibeam satellite systems," in *Proc. IEEE Veh. Technol. Conf.*, May 2020, pp. 1–5.
- [14] G. Cocco, T. de Cola, M. Angelone, Z. Katona, and S. Erl, "Radio resource management optimization of flexible satellite payloads for DVB-S2 systems," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 266–280, Jun. 2018.
- [15] V. Joroughi *et al.*, "Deploying dynamic on-board signal processing schemes for multibeam satellite systems," in *Proc. IEEE Global Commun. Conf.*, 2019, pp. 1–6.
- [16] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2*, 3rd Generation Partnership Project (3GPP), Standard ETSI TS 136 300, 2012.
- [17] Y. Rui, P. Cheng, M. Li, Q. T. Zhang, and M. Guizani, "Carrier aggregation for LTE-advanced: uplink multiple access and transmission enhancement features," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 101–108, Aug. 2013.
- [18] H. Lee, S. Vahid, and K. Moessner, "A survey of radio resource management for spectrum aggregation in LTE-advanced," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 745–760, 2014.
- [19] D. López-Pérez *et al.*, "Long term evolution-wireless local area network aggregation flow control," *IEEE Access*, vol. 4, pp. 9860–9869, Jan. 2017.
- [20] S. Singh, M. Geraseminko, S. Yeh, N. Himayat, and S. Talwar, "Proportional fair traffic splitting and aggregation in heterogeneous wireless networks," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1010–1013, May 2016.

- [21] S. Goyal, T. B. Le, A. Chincholi, T. Elkourdi, and A. Demir, "On the packet allocation of multi-band aggregation wireless networks," *Wireless Netw.*, vol. 24, no. 7, p. 2521–2537, Oct. 2018.
- [22] M. Kim and C. Choi, "Probing-based link adaptation for high data rate wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 11, no. 7, pp. 2382–2390, Jul. 2012.
- [23] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G*, 3rd ed. Orlando, FL, USA: Academic Press, Inc., 2016.
- [24] European Space Agency (ESA), "CADSAT - Carrier aggregation in satellite communication networks," 2019. [Online]. Available: <https://artes.esa.int/projects/cadsat>
- [25] M. Kibria, E. Lagunas, N. Maturo, D. Spano, H. Al-Hraishawi, and S. Chatzinotas, "Carrier aggregation in multi-beam high throughput satellite systems," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.
- [26] M. Kibria, E. Lagunas, N. Maturo, H. Al-Hraishawi, and S. Chatzinotas, "Carrier aggregation in satellite communications: Impact and performance study," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1390–1402, 2020.
- [27] E. Lagunas, M. Kibria, H. Al-Hraishawi, N. Maturo, and S. Chatzinotas, "Dealing with non-uniform demands in flexible GEO satellites: The carrier aggregation perspective," in *Proc. 10th Adv. Satellite Multimedia Syst. Conf. 16th Signal Process. Space Commun. Workshop*, Oct. 2020, pp. 1–5.
- [28] *Digital Video Broadcasting (DVB): Implementation Guidelines for the Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: S2 Extensions (DVB-S2X)*, European Telecommunications Standards Institute, Standard ETSI TR 102 376-2, 2015.
- [29] *Digital Video Broadcasting (DVB): Generic Stream Encapsulation (GSE); Part 1: Protocol Technical Specifications*, European Telecommunications Standards Institute, Standard ETSI TS 102 606-1 V1.2.1, 2014.
- [30] *Digital Video Broadcasting (DVB): Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications; Part 1: DVB-S2*. European Telecommunications Standards Institute, Standard ETSI EN 302 307-1, 2014.
- [31] *Digital Video Broadcasting (DVB): Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: DVB-S2X*. European Telecommunications Standards Institute, Standard ETSI EN 302 307-2, 2014.
- [32] Y. Wang, K. I. Pedersen, T. B. Sorensen, and P. E. Mogensen, "Carrier load balancing and packet scheduling for multi-carrier systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1780–1789, May 2010.
- [33] S. Chatzinotas, E. Lagunas, H. Al-Hraishawi, N. Maturo, and M. Kibria. Carrier Aggregation Demonstrator for Satellite Communication Systems. [Online]. Available: https://www.eni.lu/snt/research/sigcom/sw_simulators/cadsat.
- [34] X. Yuan and Z. Duan, "Fair round-robin: A low complexity packet scheduler with proportional and worst-case fairness," *IEEE Trans. Comput.*, vol. 58, no. 3, pp. 365–379, Mar. 2009.



Hayder Al-Hraishawi (Member, IEEE) received the B.Sc. and M.Sc. degrees in engineering from the Department of Electrical Engineering, Al-Mustansiriyah University, Baghdad, Iraq, in 2003 and 2006, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, IL, USA, in 2017. He is currently a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg. He is a Member (Associate Editor) of the Editorial Board

of the IEEE ACCESS. He has also held several industrial positions with Motorola Solutions, Huawei Technologies, and Nokia Networks. His current research interests include designing and analysis of spectrum-sharing techniques for massive MIMO systems, wireless energy harvesting, physical-layer security, and resource allocation for satellite communications.



Nicola Maturo (Member, IEEE) received the M.S. degree in electronic engineering (cum laude) and the Ph.D. degree in telecommunication engineering from the Polytechnic University of Marche, Ancona, Italy, in 2012 and 2015, respectively. From January 2016 to July 2017, he was a Postdoctoral Researcher with the Department of Information Engineering, Polytechnic University of Marche, where he worked on error correcting coding techniques under some ESA research projects. From November 2015 to May 2016, he was a Consultant with Deimos Engenharia, Lisbon, Portugal, working on spectral estimation algorithms and anti-jamming techniques. Since August 2017, he has been a Research Scientist with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg, mainly working in the satellite communication domain. Since 2015, he has been a Member of the CCSDS Coding and Synchronization Working Group.



Eva Lagunas (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 2010 and 2014, respectively. From 2009 to 2013, she was a Research Assistant with the Department of Signal Theory and Communications, UPC. In 2009 she was a Guest Research Assistant with the Department of Information Engineering, University of Pisa, Pisa, Italy. From November 2011 to May 2012, she held a Visiting Research appointment with the Center for Advanced Communications, Villanova University, PA, USA. In 2014, she joined the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, where she currently holds a Research Scientist position. Her research interests include radio resource management and general wireless networks optimization.



Symeon Chatzinotas (Senior Member, IEEE) received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Guildford, U.K., in 2006 and 2009, respectively. He is currently a Full Professor or Chief Scientist I and the Co-Head of the SIGCOM Research Group, Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. In the past, he was a Visiting Professor with the University of Parma, Parma, Italy, and he was involved in numerous Research and Development projects for the National Center for Scientific Research Demokritos, the Center of Research and Technology Hellas and the Center of Communication Systems Research, University of Surrey. He has coauthored more than 400 technical papers in refereed international journals, conferences and scientific books. He was the co-recipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award, the CROWNCOM 2015 Best Paper Award, and the 2018 EURASIP JWCN Best Paper Award. He is currently in the Editorial Board of the IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY and the *International Journal of Satellite Communications and Networking*.