

Multi-round Message Scheduling for Fast GNSS Packet Broadcasting

Francesco Ardizzone, Nicola Laurenti, and Stefano Tomasin
Department of Information Engineering
University of Padova, Italy
email: first.lastname@dei.unipd.it

Abstract—Global navigation satellite systems (GNSSs) may transmit data to provide additional services. Since the resources reserved for these data are typically fixed and the message rate is low, we propose to split long messages into shorter packets and properly schedule their transmission from the entire constellation of satellites. Considering an efficient data transmission (useful for example for search and rescue messages), we aim at the scheduling of packets on the satellites on multiple rounds with two objectives: either a) the minimization of the maximum latency among all receivers, or b) the maximization of the average received packets per round. We first derive bounds on the performance of any GNSS single or multi-round scheduling solution, on which the proposed scheduling solutions are based. Then we introduce the scheduling problems that turn out to be integer linear programming (ILP) problems. Lastly, we assess their performance, showing that our solution minimizes the maximum latency, while the scheduling targeting the average latency outperforms existing literature solutions.

I. INTRODUCTION

Global navigation satellite systems (GNSSs) transmit signals enabling the estimation of position velocity and time (PVT) by receivers. Additionally, they also periodically broadcast data messages carrying useful information for navigation, such as ephemeris and almanacs. Recently, studies have been conducted to support new services, such as authentication data, search-and-rescue (SAR), and short messaging [1]–[3]: these solutions require as well the broadcast of further data messages. Such data transmissions are particularly useful in remote areas, where other communication services or side channels are not available. In this case, hybrid satellite-Internet solutions such as those provided by the high accuracy service (HAS) [4] cannot be adopted.

We focus on the transmission of messages, such as SAR messages or short messages, and we aim at minimizing the time by which a message is received by a device on the ground, in the following denoted as *latency*. Since GNSS are characterized by low data rates (in the order of a few hundred bit/s, e.g., 50 bit/s for GPS L1 C/A [5, Chapter 4] and blue 125 bit/s for Galileo E1B [6]), latency becomes significant, especially for long messages. To overcome this issue, we must exploit the specific broadcast nature of the satellite signal and the availability of multiple satellites in view from ground receivers.

Conventional beamforming techniques of multiple-input-multiple-output (MIMO) systems cannot be applied in this context: although the transmission from multiple satellites to

a single receive antenna can be modeled as a multiple-input-single-output (MISO) channel, it is not possible to ensure a synchronous reception of the signal in all possible positions of the receiver, due to the distance among the transmit antennas of different satellites. Moreover, the satellite orbits modify their visibility by the receiver, further preventing coordination among satellites at the symbol level.

The problem of latency minimization for broadcasting services has been studied in different contexts, e.g., wireless sensor networks (WSN) [7] and cognitive radio networks (CRN) [8]. Still, our scenario has significant differences, namely: a) no feedback channel is available from the receiver to the transmitter; b) the bipartite graph modeling the network is not complete, since only a subset of satellites is in view of each receiver; c) the position of transmitters and receivers is periodically changing over time, hence the scheduling solution will be time-variant and periodic. Thus, we must develop solutions specifically targeted to the GNSS context.

In this context, the typical solution is the *carousel scheduling strategy* [9], where each different packet is scheduled sequentially in time. This solution is currently used for disseminating the almanacs, used to aid the receiver's signal acquisition phase. With the carousel strategy, only the almanacs for two space vehicles (SVs) are retrieved within each sub-frame (i.e., 30 s) [6].

Still, at packet level, the diversity provided by multiple satellites can be exploited. Indeed, by splitting the message into several packets and transmitting them via different satellites and specific spreading code, ground receivers can collect the packets and obtain the entire message. For Galileo open service navigation message authentication (OS-NMA), the subset of satellites distributing the authentication data dynamically changes, and no user knows in advance which satellite is actually transmitting the message [10]. Still, a message splitting approach has been proposed for OS-NMA in [11] and [12]. In this context, the use of random fountain codes has been proposed [13] to increase the reliability of the packet dissemination. No further link-level solution has been proposed for OS-NMA yet. A fountain code-based solution was also proposed in [9] and compared to the carousel strategy used for dissemination of the I/NAV messages in Galileo [6].

In this scenario, a key issue is the scheduling of packets on the different satellites which can be performed in advance according to a deterministic rule. In fact, the positions of the SVs at any instant are easily predictable and the set of satellites in view from any point on the surface of the earth

can be consistently determined. By assigning a specific packet to be broadcast by each satellite, we can determine the ground position where it will be reliably received. Yet, only a few studies are available on this topic, in the literature. Message allocation strategies are proposed for HAS in [14], [15] taking into account the satellite positions but not the actual receivers' visibility. Moreover, in those works, the authors focus on the dissemination of a fixed number of packets (10), while we consider a more general scheduling problem.

Defined *coverage* as the fraction of area from which they receive the whole message, in [16] the authors proposed a solution to maximize the coverage for fast message transmission, i.e., the fraction of receivers that can obtain all the packets and reconstruct the message at once. However, it was shown that no scheduling strategy could achieve full coverage and reliable reception for 4 packets at the same time with the Galileo system, therefore this solution is only effective for short messages. Here, we focus instead on solutions for multi-round scheduling, targeting longer message scenarios.

In this paper, we propose a scheduling of packets over both the satellites and the time, for fast message transmission to overcome the limitations of existing solutions. Then the packets can also be re-transmitted several times by the satellites until they are reliably received over earth. We denote as *round* the time used for the transmission of a single packet and consider that the transmission of the entire message spans several rounds. The scheduling of packets on the satellites among multiple rounds can be performed with two alternative objectives: either a) the minimization of the maximum latency among all receivers, or b) the maximization of the average received packets per round. In the first case, we consider latency as the key metric and aim at minimizing it for the receiver in the worst conditions. With the latter objective instead, we aim at maximizing the data rate. A third objective is a variation of b), where we also aim at maximizing the coverage at each round.

We first establish bounds on both the average number of different received packets and the maximum latency, under the maximum-diversity scenario, wherein each receiver obtains a different packet from each satellite in view. Then, we leverage the bounds to derive scheduling algorithms. We show that under slow variations of visibility conditions across rounds, the scheduling problem minimizing the maximum latency can be split into separate scheduling problems for each round. A single round scheduling is then formulated as a binary integer linear programming (BILP) problem. We also consider the problem of maximizing the average received packets per round, for which we first quantize the coverage area and then obtain a BILP problem. Numerical results show the validity of our solution and the improvement with respect to the solution in [16], obtaining reduced average and maximum latency.

The rest of the paper is organized as follows: in Section II we introduce the system model and the metrics used for GNSS message transmission; Section III states the optimal bounds for average, maximum latency and maximum average received packets analyzing the maximum diversity scenario; in Section IV we propose solutions for the minimum maximum latency problem and the maximum average number of different

received packets. Section V collects the numerical results of the performance of the proposed algorithms. Lastly, we draw some conclusions in Section VI.

II. SYSTEM MODEL

We consider a scenario where a set $S = \{1, \dots, M\}$ of M GNSS satellites aim at transmitting a message p . The message is split into K packets of equal size, (p_1, \dots, p_K) . All the packets are needed to reconstruct p , i.e., no form of packet coding is considered. Time is divided into rounds, each of duration T , in which each satellite transmits one packet. The transmission of p may last multiple rounds. Each packet is in general transmitted by multiple satellites over multiple rounds to reach all ground receivers.

Message p is to be received by ground receivers over a region A : let $\alpha_{s,n}(\mathbf{x})$ be the elevation angle of the satellite $s \in S$ at round n with respect to position $\mathbf{x} \in A$; then we will consider satellite s as visible (i.e., its signal can be correctly received in \mathbf{x}) if $\alpha_{s,n}(\mathbf{x}) \geq \alpha_{\min}$ where α_{\min} is a threshold typically fixed by the receivers. Typical values for α_{\min} are in the range $[5^\circ, 15^\circ]$ [17, §15.7].

Indeed, by picking a large α_{\min} , we are potentially discarding useful satellites. However, signals coming from low-elevation satellites are subject to atmospheric and multipath distortion, or possibly even blocked by nearby obstacles [5, §7], hence these signals can typically be discarded a priori without degrading the performance of the receiver. We introduce the *satellite visibility maps* as

$$v_{s,n}(\mathbf{x}) \triangleq \begin{cases} 1 & \text{if } \alpha_{s,n}(\mathbf{x}) \geq \alpha_{\min}, \\ 0 & \text{if } \alpha_{s,n}(\mathbf{x}) < \alpha_{\min}, \end{cases} \quad (1)$$

indicating that satellite s is visible by a receiver in \mathbf{x} during round n if $v_{s,n}(\mathbf{x}) = 1$. Notice that we choose a value constant for ease of reading. This model may be modified by considering a different masking elevation angle for each position, $\alpha_{\min}(\mathbf{x})$, which models a map with a receiver placed in an open sky condition or in an urban scenario, etc.. We consider that the relative positions of satellites and receivers change over the rounds: however, since T is much smaller than the orbital period of the satellites, we assume the satellite position to be static within each round.

We make the following assumptions for each GNSS receiver:

- 1) has a buffer with size at least equal to the length of p , to store all the received packets.
- 2) receives without error the packet transmitted from any visible satellite, i.e., we assume perfect error-correcting coding.
- 3) can decode all the packets sent by all satellites in view at the same time without interference degradation; this is typically achieved by transmitting messages in a code division multiple access (CDMA) fashion and using a different code for each satellite.

During round n , satellites and ground receivers perform the following operations:

- 1) satellite $s \in S$ obtains packet $\pi_{s,n} \in \{1, \dots, K\}$ from the ground segment; the choice of $\pi_{s,n}$ is the true subject of the scheduling discussed in the next Section;
- 2) satellite s transmits packet $\pi_{s,n}$;
- 3) each receiver in a position \mathbf{x} such that $v_{s,n}(\mathbf{x}) = 1$ decodes the packet;
- 4) if the receiver has already obtained packet $\pi_{s,n}$ during previous rounds or from another satellite, it discards the packet, otherwise it stores the packet in its buffer;
- 5) once the receiver has collected all the K packets, it reconstructs message p and waits for a new message.

Note that the proposed protocol is compatible with any channel code mechanism, e.g., the Fountain codes used, for instance, in [1], [9], [14]. Moreover, our solution can also be applied without channel coding. Indeed, our solution is compatible with any existing GNSS standard.

Additionally, we remark that unlike other proposed protocols, such as [14], [15], ours are more general and make no particular assumption on the number of packets to be disseminated, the region to be covered, or the SVs' position: thus, we could use this scheme to schedule a message either over a specific region A (e.g., Europe) or using only a limited subset of the satellites, instead of the whole set S , e.g., by scheduling different messages to different subsets of satellites.

A. Performance Metrics

The transmission scheduling at round n is defined by a partition of S , $\mathcal{S}_n = (S_{1,n}, \dots, S_{K,n})$, where $S_{k,n}$ is the set of satellites transmitting packet k at round n . The transmission scheduling from round 1 to n is $\mathcal{P}_n = (S_1, \dots, S_n)$. In general, we aim at finding a scheduling strategy that allows the receivers in A to recover p in a short time. We now introduce three specific metrics that will be used to design the scheduling algorithms, namely a) the average number of different received packets, b) the maximum latency, and c) the average latency.

a) *Average Number of Received Packets*: the first metric is related to the number of different received packets in a given time and in a given area, thus it merges coverage and throughput performance. By introducing the *availability of packet k at position \mathbf{x} by round n*

$$u_{k,n}(\mathcal{P}_n, \mathbf{x}) \triangleq \begin{cases} 1 & \text{if } \sum_{m=1}^n \sum_{s \in S_{k,m}} v_{s,m}(\mathbf{x}) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

we count the *total number of different received packets at position \mathbf{x} up to round n* , as

$$\eta_n(\mathcal{P}_n, \mathbf{x}) \triangleq \sum_{k=1}^K u_{k,n}(\mathcal{P}_n, \mathbf{x}). \quad (3)$$

This indicates the amount of data received as more rounds are carried out in a given position. Lastly, by averaging the above quantity over the area A , we obtain the *average number of different received packets in A up to round n* is

$$\bar{\eta}_n(\mathcal{P}_n) \triangleq \frac{1}{|A|} \int_A \eta_n(\mathcal{P}_n, \mathbf{x}) d\mathbf{x}. \quad (4)$$

This metric is related to the coverage up to the current frame. Note however that a high value of $\bar{\eta}_n(\mathcal{P}_n)$ does not ensure

that all points in the area have received the same number of packets.

b) *Maximum Latency*: the second metric is referred to as the latency, i.e., the number of rounds necessary for a receiver to obtain all the K packets and reconstruct p . First, we define the latency for a receiver in position \mathbf{x} as

$$\tau(\mathcal{P}, \mathbf{x}) \triangleq \min \left\{ n : \eta_n(\mathcal{P}_n, \mathbf{x}) = K \right\}, \quad (5)$$

where $\mathcal{P} = (S_1, S_2, \dots)$ is the scheduling sequence at all rounds. The *maximum latency* is defined as the maximum latency among all receivers in the area A , i.e.,

$$\tau_{\max}(\mathcal{P}) \triangleq \max_{\mathbf{x} \in A} \tau(\mathcal{P}, \mathbf{x}). \quad (6)$$

The latency is a relevant metric for several applications, in particular in automation, including self-driving cars or unmanned aerial vehicles. Minimizing the maximum latency means to keep under control the latency in the worst-case scenario, which is a suitable criterion for time-critical applications. Still, considering the maximum latency may be significantly penalizing for the average number of different received packets. Therefore, we also consider another metric related to the latency.

c) *Average Latency*: the third metric is the *average latency* for all receivers in area A , obtained by integrating (5) over A , as

$$\bar{\tau}(\mathcal{P}) \triangleq \frac{1}{|A|} \int_A \tau(\mathcal{P}, \mathbf{x}) d\mathbf{x}. \quad (7)$$

On one hand, keeping the average latency under control does not guarantee good performance in the worst-case, but only on average. On the other hand, this milder metric may allow better performance in terms of average received packets, as it will be shown in the following.

III. PERFORMANCE BOUNDS

In this Section, we obtain bounds on the average number of different received packets, coverage, and latency. We will also exploit these results to prove the optimality of one of the proposed scheduling algorithms and develop the solution described in Section IV-C. Finally, these will be also used as either upper or lower bounds to evaluate the proposed algorithms' performance.

The first bound relates the average number of different received packets to the coverage of the area and the total number of different received packets, i.e., the coverage and the throughput, as already mentioned when introducing the metric. About the coverage, from the availability of packet k at position \mathbf{x} by round n , $u_{k,n}(\mathcal{P}_n, \mathbf{x})$, we obtain the *availability of the entire message at position \mathbf{x} by round n* as

$$\hat{u}_n(\mathcal{P}_n, \mathbf{x}) \triangleq \begin{cases} 1 & \text{if } \prod_{k=1}^K u_{k,n}(\mathcal{P}_n, \mathbf{x}) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

This denotes the receivers that at round n were able to actually receive all the K packets. Next, we can formally introduce

the (fractional) coverage of A by round n by integrating the availability of the message over the coverage area A , as

$$\bar{u}_n(\mathcal{P}_n) \triangleq \frac{1}{|A|} \int_A \hat{u}_n(\mathcal{P}_n, \mathbf{x}) d\mathbf{x}. \quad (9)$$

All the above expressions always counted just the different packets: in the next, we will consider instead the total number of received packets.

About the average number of received packets, we define the total number of received packets (possibly with repetitions) at position \mathbf{x} up to round n as

$$C_n(\mathbf{x}) \triangleq \sum_{s \in S} \sum_{m=1}^n v_{s,m}(\mathbf{x}), \quad (10)$$

and its average over A , i.e., the number of received packets per area, as

$$\bar{C}_n = \frac{1}{|A|} \int_A C_n(\mathbf{x}) d\mathbf{x}. \quad (11)$$

We can now formulate the following proposition given upper and lower bounds to the average number of received packets.

Proposition 1. *The average number of different received packets is bounded as*

$$K\bar{u}_n(\mathcal{P}_n) \leq \bar{\eta}_n(\mathcal{P}_n) \leq \min(K, \bar{C}_n(n)). \quad (12)$$

Proof. We start proving the left hand side: given the sequence of partitions \mathcal{P}_n such that $\bar{u}_n(\mathcal{P}_n) = \beta$, at least a fraction β of the receivers obtained all the K packets at the end of the n th round, therefore the average number of delivered packets is at least $K\beta$, i.e., $\bar{\eta}_n(\mathcal{P}_n) \geq K\bar{u}_n(\mathcal{P}_n)$.

For the right hand side, we can write

$$C_n(\mathbf{x}) = \sum_{s \in S} \sum_{m=1}^n v_{s,m}(\mathcal{P}_m, \mathbf{x}) \geq \eta_n(\mathcal{P}_n, \mathbf{x}), \quad (13)$$

then by averaging both sides of (13) over A and considering that $\eta_n(\mathcal{P}_n, \mathbf{x}) \leq K$, we obtain (12). \square

From this bound we observe that a solution \mathcal{P}_n achieving full coverage also maximizes the average number of different received packets, with $\bar{\eta}_n(\mathcal{P}_n) = K$; on the other hand, among the sequences \mathcal{P}_n that achieve the same partial coverage, some may obtain a higher average number of different received packets. This result will be used for the development of the scheduling algorithm in Section IV-C.

A. Maximum Diversity Scenario

We also derive bounds on the maximum and average latency. In this case, we focus on a particular scenario, that we denote as maximum diversity scenario, characterized by the fact that each receiver obtains a different packet from each satellite in view. This is clearly a very favorable condition, not always met in practice, since in each round the receiver obtains the maximum number of packets, for a given set of satellites in view. Thus, we will obtain bounds on the performance associated with a best-case scenario. In formulas, the maximum diversity scenario can be alternatively described

by indicating that the message is received once K satellites have been in view, and thus $C_n(\mathbf{x}) = K$ packets have been received. In terms of the availability of the entire message at position \mathbf{x} by round n we have

$$\hat{u}_n^*(\mathbf{x}) = \begin{cases} 1 & \text{if } C_n(\mathbf{x}) \geq K \\ 0 & \text{if } C_n(\mathbf{x}) < K \end{cases}, \quad (14)$$

or alternatively, the total number of different received packets at position \mathbf{x} up to round n is

$$\eta_n^*(\mathbf{x}) = \min(K, C_n(\mathbf{x})). \quad (15)$$

Note that in both $\hat{u}_n^*(\mathbf{x})$ and $\eta_n^*(\mathbf{x})$ we omitted the indication of the scheduling \mathcal{P}_n , as we are considering an ideal scenario which may not be feasible, i.e., for which no scheduling allows to achieve such performance.

However, this ideal scenario will provide bounds on the latency metrics. In particular, by averaging over the area, we obtain upper bounds on the average number of different received packets $\bar{\eta}_n^*$ and the coverage \bar{u}_n^* . These values are upper bounds for the performance achieved in any scenario, i.e., we always have

$$\bar{\eta}_n(\mathcal{P}_n) \leq \bar{\eta}_n^*, \quad \bar{u}_n(\mathcal{P}_n) \leq \bar{u}_n^*. \quad (16)$$

The latency in the maximum diversity scenario for the receiver in position \mathbf{x} is

$$\tau^*(\mathbf{x}) \triangleq T \min \left\{ n : \hat{u}_n^*(\mathbf{x}) = 1 \right\}, \quad (17)$$

from which we obtain correspondingly τ_{\max}^* , $\bar{\tau}^*$. These values are bounds for the performance achieved with any scheduling solution, i.e., we always have

$$\tau_{\max}(\mathcal{P}) \geq \tau_{\max}^*, \quad \bar{\tau}(\mathcal{P}) \geq \bar{\tau}^*, \quad \forall \mathcal{P}. \quad (18)$$

IV. SCHEDULING SOLUTIONS

In this Section, we propose scheduling algorithms for the considered multi-round multi-packet transmission problem. We first aim at minimizing the maximum latency $\tau_{\max}(\mathcal{P})$ and propose a solution that is optimal under a suitable condition on $C_n(\mathbf{x})$, the total number of received packets up to round n . Then, we observe that the min-max latency algorithm neither minimizes the average latency nor maximizes the throughput. Still, to optimize these two metrics, we should jointly schedule transmission at all rounds, as from (7) and (27). This leads to an extremely complex solution. In summary, propose three heuristic approaches: a) minimizing the maximum latency, which however does not minimize the average latency, b) maximizing the average received packets per round, and c) maximizing the coverage as the primary objective and the average received packets per round as a secondary objective.

A. Minimization of the Maximum Latency (MIN-MAX)

We now consider the problem of finding a scheduling \mathcal{P} that minimizes the maximum latency among all the receivers, i.e.,

$$\min_{\mathcal{P}} \tau_{\max}(\mathcal{P}). \quad (19)$$

Algorithm 1 Min Max Latency Solution

Input: K, A
Output: \mathcal{P}
 $n \leftarrow 0, K' \leftarrow K$
while $K' > 0$ **do**
 $n \leftarrow n + 1$
 $v_{s,n}(\mathbf{x}) \leftarrow \text{comp_visibility_maps}(n, A)$
 $K_n \leftarrow 1; \gamma \leftarrow 1; \mathcal{P}_n \leftarrow \{S\}$
 while $K_n < K'$ and $\gamma = 1$ **do**
 $\{\mathcal{P}'_n, \gamma\} = \text{solve_max_cov}(A, K_n, v_{s,n}(\mathbf{x}))$
 if $\gamma = 1$ **then**
 $\mathcal{P}_n = \mathcal{P}'_n$
 $K_n \leftarrow K_n + 1$
 end if
 end while
 $K' = K' - K_n$
end while

First, we observe that the problem (19) is equivalent to the cascade of several problems, one for each round as shown by the following proposition and can be solved accordingly.

Proposition 2. Let $C_{\min,n} \triangleq \min_{\mathbf{x} \in A} C_n(\mathbf{x})$, if

$$C_{\min,n} = \sum_{m=1}^n \min_{\mathbf{x} \in A} \sum_{s \in S} v_{s,m}(\mathbf{x}). \quad (20)$$

the min-max latency problem is equivalent to maximizing the number of packets that can be transmitted in a single round with full coverage.

We report the proof in the Appendix.

Condition (20) essentially states the equivalence between the minima of the sum, i.e., $C_{\min,n}$, and the sum of the minima. This condition can easily be met in this context since if the round duration T is small enough, the visibility map $v_{s,m}(\mathbf{x})$ does not exhibit dramatic changes between rounds, hence, with high probability, $\min_{\mathbf{x} \in A} \sum_{s \in S} v_{s,m}(\mathbf{x})$ stays constant for a few rounds. This allows requirement (20) to be met. We will confirm the optimality of this approach in Section V, showing that $\min_{\mathbf{x} \in A} \sum_{s \in S} v_{s,m}(\mathbf{x}) = 4$ for most rounds.

Hence, we propose the following algorithm. Let K_n be the number of different packets transmitted at round n and not transmitted in previous rounds, so that $\sum_n K_n = K$. Then, K_n is chosen as the maximum number of packets that can be transmitted in a single round with full coverage. To this end, we iteratively increase K_n starting from $K = 1$ and resort to the BILP algorithm of [16] to compute the corresponding coverage γ , until we reach full coverage. The resulting solution is reported in Algorithm 1, where $K' = K - \sum_n K_n$ represents the number of packets left to be transmitted.

In the algorithm we exploit the following functions:

comp_visibility_maps : computes the satellite positions at round n and outputs the visibility maps, $v_{s,n}(\mathbf{x})$;
solve_max_cov : implements the (single round) coverage optimization [16], which we have shown to be effective for small K values.

Algorithm 1 follows a divide-and-conquer approach. If the algorithm is iterated for n rounds, the overall computational cost is n times the cost of the single round scheduling solution, which from [16] is exponential in the size $|A|$ and the number of packets to be transmitted, $K_n \leq C_{\min,n}$. However, since $C_{\min,n}$ is typically small ($C_{\min,n} \approx 4$), in practice we expect the algorithm to converge quickly.

We now observe that the min-max latency algorithm neither minimizes the average latency nor maximizes the throughput. Still, to optimize these two metrics, we should jointly schedule transmissions at all rounds, as from (7) and (27), yielding a high-complexity solution. Then, in the following we consider two other suboptimal approaches.

B. Maximization of the Average Number of Different Received Packets (MRP)

We now consider the problem of maximizing the average number of different received packets $\bar{\eta}_n(\mathcal{P}_n)$ at each round n , i.e., given \mathcal{P}_{n-1} , the sequence of partitions up to round $n-1$, we aim at solving the optimization problem

$$\max_{\mathcal{S}_n} \bar{\eta}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}) \quad \forall n. \quad (21)$$

Note that $\mathcal{P}_{n-1} = (\mathcal{S}_1, \dots, \mathcal{S}_{n-1})$.

First, we convert the integrals over the area (for the average number of different received packets) into sums over a discrete set of points $\Omega \subset A$, following the approach of [16]. We consider the corresponding tessellation $\mathcal{A} = \{A(\mathbf{x}), \mathbf{x} \in \Omega\}$, where

$$\bigcup_{\mathbf{x} \in \Omega} A(\mathbf{x}) = A \quad (22)$$

and

$$A(\mathbf{x}) \cap A(\mathbf{x}') = \emptyset, \quad \text{if } \mathbf{x} \neq \mathbf{x}'. \quad (23)$$

Ideally, all the points of the same tile have the same satellites in view, i.e., it should hold $v_{s,n}(\mathbf{x}) = v_{s,n}(\mathbf{x}') \forall s \in S, \forall \mathbf{x}' \in A$ for each tile $A(\mathbf{x})$. However, this would require to recompute sampling Ω and the tessellation at each round. In order to avoid this issue, we keep Ω fixed. Moreover, each tile has normalized area

$$a(\mathbf{x}) = \frac{|A(\mathbf{x})|}{|A|} \quad \mathbf{x} \in A(\mathbf{x}), \quad (24)$$

thus, in this framework, the integrals over A of the performance metrics of Section II-A need to be considered as weighted sums.

First, we can formulate the following BILP problem

Problem 1 (Single Round Maximization). *Given for all the sampled positions \mathbf{x} ,*

- the visibility maps $v_{s,n}(\mathbf{x})$,
- the normalized tile area $a(\mathbf{x})$
- and the availability of each packet up to round $n-1$, $u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x})$

choose the scheduling \mathcal{S}_n for round n , that maximizes the average number of received packets, $\bar{\eta}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\})$, and update the availability map $u_{k,n}(\mathcal{P}_n, \mathbf{x})$.

The mathematical formulation of this Problem is reported on Appendix B and can be solved through well-known approaches, e.g., the Branch and Bound technique [18, §9].

Algorithm 2 is the proposed strategy for the Maximization of the Average Number of Different Received Packets (MRP), where `mrp_max` solves (1) and `round_eval` is a function that takes as input the set of received packets at the end of the previous round, $u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x})$, and outputs the coverage and the set of received packets at the end of round n , $u_{k,n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x})$, by using (2)-(9).

With this approach we iterate at each round the solution of problem (21), thus the overall computational cost is linearly dependent on the solution of (21) itself. The solution of the BILP problem using Branch and Bound has a cost that is exponential in the number of constraints. More in detail, called $N_s = |\Omega|$ the number of sampling points of the surface A , we have $M + KN_s$ constraints. Thus, the computational complexity depends on both the chosen sampling and the number of packets to be transmitted.

C. Maximization of the Average Number of Different Received Packets among Maximum Coverage Solutions (MRP-MC)

We now observe there may be several scheduling achieving maximum coverage with different numbers of distinct transmitted packets. Thus, we propose a scheduling algorithm that at each round n , given the set of received packets for each receiver up to the previous round, $u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x})$, first maximizes the coverage $\bar{u}_n(\mathcal{P}_n)$ and then maximizes the average number of different received packets $\bar{\eta}_n(\mathcal{P}_n)$, among all the scheduling solutions that achieve coverage. In formulas, at round n , considering the sequence of partitions up to round $n - 1$ as fixed, we have to solve

$$\max_{\mathcal{S}_n \in \mathcal{M}} \bar{\eta}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}), \text{ with} \quad (25a)$$

$$\mathcal{M} = \{\mathcal{S}_n^* = \arg \max_{\mathcal{S}_n} \bar{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\})\}. \quad (25b)$$

This is the cascade of two maximization problems. Thus, following the approach of the previous Section, we formulate both optimizations as BILP problems to be iteratively solved, as for Problem 1.

Problem 2 (Single round coverage maximization). *Given the same inputs of Problem 1, choose the scheduling \mathcal{S}_n for round n , that maximizes the coverage function $\bar{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\})$.*

Algorithm 2 MRP

Input: K, Ω

Output: \mathcal{P}

```

 $n \leftarrow 1, \bar{u}_0 \leftarrow 0$ 
 $u_{k,0}(\mathcal{P}_0, \mathbf{x}) \leftarrow 0, \quad \forall k \in \{1, \dots, K\}, \mathbf{x} \in \Omega$ 
while  $\bar{u}_{n-1} < 1$  do
   $v_{s,n}(\mathbf{x}) \leftarrow \text{comp\_visibility\_maps}(n, \Omega)$ 
   $\{\mathcal{P}_n, \bar{\eta}_n\} \leftarrow \text{mrp\_max}(\Omega, K, v_{s,n}(\mathbf{x}), u_{k,n-1}(\mathbf{x}))$ 
   $\{\bar{u}_n, u_{k,n}(\mathcal{P}_n, \mathbf{x})\} \leftarrow$ 
   $\text{round\_eval}(\Omega, \mathcal{P}, v_n(\mathbf{x}), u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x}))$ 
   $n \leftarrow n + 1$ 
end while

```

Problem 3 (Single round throughput maximization with maximum coverage). *Given the same inputs of Problem 1 and 2 and the coverage $\bar{u}_n(\{\mathcal{P}_n, \mathcal{S}_n\})$, obtained by the latter, choose the scheduling \mathcal{S}_n for round n that maximizes the average number of received packets $\bar{\eta}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\})$ and update the availability map $u_{k,n}(\mathcal{P}_n, \mathbf{x})$.*

The mathematical formulations of these Problems are reported on Appendix C.

Algorithm 3 iteratively solves (25), by using the following functions:

compute_uBound computes the upper bound η_n^* , by using (16);

solve_max_cov and **mrp_mc_max** are respectively the optimization of the BILP Problems 2 and 3;

is_feasible computes the upper bound of Equation (16) verifying if it is possible to achieve a solution with non-null coverage (i.e., $\bar{u}_n^* > 0$). Note that if $C_n(\mathbf{x}) < K \forall \mathbf{x} \in A$, i.e., all the receivers obtained less than K packets, (14) yields $\bar{u}_n^* = 0$, thus all the scheduling solutions will achieve zero coverage. So, we can skip the coverage maximization step, saving computational power in the early rounds, where we noticed that the computations typically take longer.

Algorithm 3 MRP-MC

Input: K, Ω

Output: \mathcal{P}

```

 $n \leftarrow 0, \bar{u}_0 \leftarrow 0$ 
 $u_{k,0}(\mathcal{P}_0, \mathbf{x}) \leftarrow 0, \quad \forall k \in \{1, \dots, K\}, \mathbf{x} \in \Omega$ 
while  $\bar{u}_n < 1$  do
   $n \leftarrow n + 1$ 
   $v_{s,n}(\mathbf{x}) \leftarrow \text{comp\_visibility\_maps}(n, \Omega)$ 
   $\eta_n^* \leftarrow \text{compute\_uBound}(v_{s,n}(\mathbf{x}), \Omega)$ 
  if is_feasible( $K, v_{s,n}(\mathbf{x}), u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x})$ ) then
     $\mathcal{P}_n \leftarrow \text{solve\_max\_cov}(\Omega, K, v_{s,n}(\mathbf{x}), u_{k,n}(\mathcal{P}_n, \mathbf{x}))$ 
     $\{\bar{u}_n, \bar{\eta}_n, u_{k,n}(\mathcal{P}_n, \mathbf{x})\} \leftarrow$ 
     $\text{eval\_state}(\Omega, \mathcal{P}_{n-1}, v_n(\mathbf{x}), u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x}))$ 
    if  $\bar{\eta}_n \leq \eta_n^*$  AND  $\bar{u}_n(\mathcal{P}_n) \leq 1$  then
       $\mathcal{P}_n \leftarrow \text{mrp\_mc\_max}(\Omega, K, v_n(\mathbf{x}), u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x}), \bar{u}_n)$ 
       $\{\bar{u}_n, \bar{\eta}_n, u_{k,n}\} \leftarrow$ 
       $\text{round\_eval}(\Omega, \mathcal{P}_{n-1}, v_n(\mathbf{x}), u_{k,n-1}(\mathbf{x}))$ 
    end if
  end if
  else
     $\mathcal{P}_n \leftarrow \text{mrp\_mc\_max}(\Omega, K, v_n(\mathbf{x}), u_{k,n-1}(\mathbf{x}), 0)$ 
     $\{\bar{u}_n, \bar{\eta}_n, u_{k,n}\} \leftarrow \text{round\_eval}(\Omega, \mathcal{P}_{n-1}, v_n(\mathbf{x}), u_{k,n-1}(\mathbf{x}))$ 
  end if
end while

```

Concerning the computational cost, in the worst-case scenario, two optimizations per round are performed, each of them involving the solution of a NP problem. As for the MRP, the cost exponentially increases with the number of constraints: for the maximum coverage we have $M + N_s(K + 1)$ with $N_s = |\Omega|$, while for the maximum coverage, we get $M + N_s(K + 1) + 1$ constraints.

D. Computational Complexity of the Proposed Solutions

Table I summarizes the complexity of the proposed scheduling solutions. All the problems are formulated by using ILP, therefore these are NP-complete problems with an exponential

computational cost, depending on the high number of involved variables and constraints [19]. First, the MIN-MAX requires the solutions of n problems, to be solved sequentially; each problem deals with the scheduling of $K_{\min} \leq K$ messages, thus it has $M + K_{\min}N_s$ constraints. The MRP combines n problems, each with $M + KN_s$ constraints. The MRP-MC requires the solution of two problems per round, the first with $M + KN_s$ constraints and the second with $M + (K + 1)N_s + 1$ constraints. Hence since the cost of the second is (exponentially) higher than the first, the computational cost is $\mathcal{O}(n \exp(M + (K + 1)N_s + 1))$. Finally, we consider the optimal solution that computes the scheduling by considering all n the rounds altogether. For instance considering Algorithm 2, we can modify it to consider multiple rounds all at once. Hence the variables (and constraints) in the MRP solution are now multiplied by n , i.e., $\exp(nM + nKN_s)$ constraints. Indeed, for non-trivial values of K or N_s , such a solution is not computationally feasible.

TABLE I

COMPLEXITY OF THE SOLUTIONS FOR THE MESSAGE SCHEDULING PROBLEM, CONSIDERING n ROUNDS, M SATELLITES, K MESSAGES, AND AN AREA SAMPLING FACTOR N_s .

Solution	Complexity
MIN-MAX	$\mathcal{O}(n \exp(M + K_{\min}N_s))$
MRP	$\mathcal{O}(n \exp(M + KN_s))$
MRP-MC	$\mathcal{O}(n \exp(M + (K + 1)N_s + 1))$
Optimal Solution	$\mathcal{O}(\exp(nM + nKN_s))$

V. NUMERICAL RESULTS

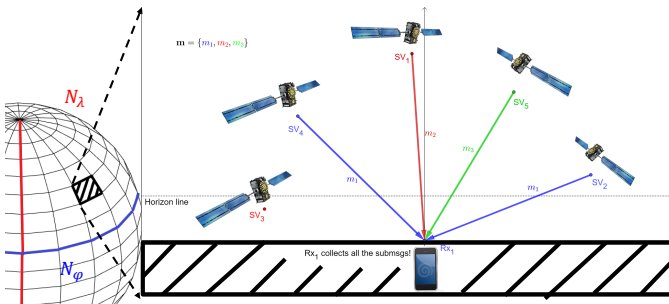


Fig. 1. Pictorial representation with the resulting tessellation and the dissemination process for $K = 3$. All the satellites but SV_3 are considered to be in view. The receiver is able to retrieve the message since it has in view satellites transmitting K different messages.

Let S_0 be the Galileo GNSS constellation with 24 medium Earth orbit (MEO) satellites distributed over 3 orbital planes with spherical orbits. For a more realistic simulation, we did not consider the whole set S_0 set but only $S \subset S_0$, the subset of satellites that are connected to one of the uplink station (ULS), as done in [1], [14], [15], [20]. For Galileo, the ULSs are located in Svalbard (78.2°N, 15.4°E), Kourou (5.2°N, 52.7°W), Papeete (17.5°S, 149.4°W), Sainte-Marie, Réunion (20.9°S, 55.5°E) and Nouméa (21.9°S, 166.0°E). In particular, we consider the four SVs that are the closest to each of the five uplink station (ULSs), hence S contains 20 SVs.

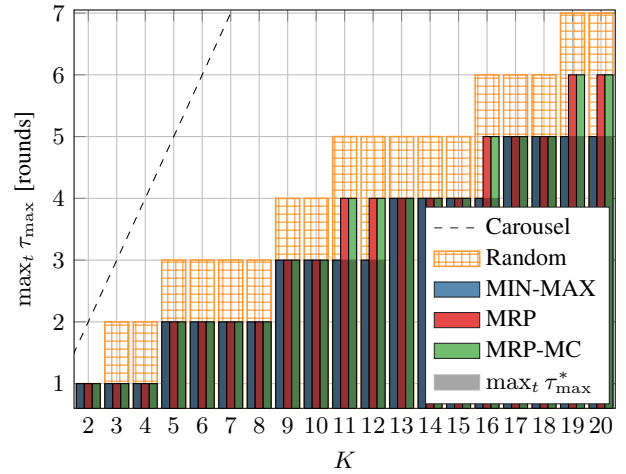


Fig. 2. Maximum latency $\max_t \tau_{\max}$ for the carousel, random, MIN-MAX, MRP, MRP-MC, and scheduling algorithms. The bound on the maximum latency obtained from (18) is also shown.

To build the visibility map (1), we consider a masking elevation angle $\alpha_{\min} = 10^\circ$. Fig. 1 illustrates the resulting tessellation and the dissemination procedure. Note that, to guarantee at least 4 satellites in view globally, a full GNSS constellation allows many more satellites in view in average, as shown later in the results (e.g., Fig. 3).

We let A be the whole (spherical) Earth surface, and the sample set Ω is obtained by uniformly sampling the latitude and longitude axis respectively with $N_\lambda = 24$ and $N_\varphi = 48$ respectively. We use the same tessellation for all the rounds. For $\mathbf{x} = (\varphi, \lambda)$ the corresponding tile is $[\varphi - \pi/(2N_\varphi), \varphi + \pi/(2N_\varphi)] \times A(\mathbf{x}) \in [\lambda - \pi/N_\lambda, \lambda + \pi/N_\lambda]$ with (normalized) area

$$a(\mathbf{x}) = a(\varphi, \lambda) = \frac{1}{N_\lambda} \cos(\varphi) \sin\left(\frac{\pi}{2N_\lambda}\right). \quad (26)$$

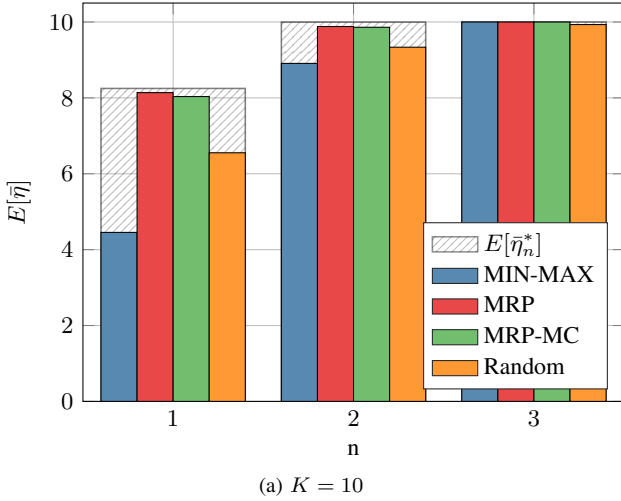
The orbital period of a Galileo satellite is $T_{\text{orb}} = 14$ h and 4 min are distributed in 3 orbital planes and equally spaced. For the numerical simulation, we consider a fast message transmission that starts every 10 min and spans a time interval of $\approx T_{\text{orb}}$.

Indeed, a finer time resolution does not yield significant changes in the visibility map (1). The duration of a round is $T = 30$ s, equal to the duration of a Galileo sub-frame [6] or a GPS frame [21]. Of course, a much finer time granularity may be needed in more dynamic contexts, such as when scheduling the transmission for a low Earth orbit (LEO) satellite system.

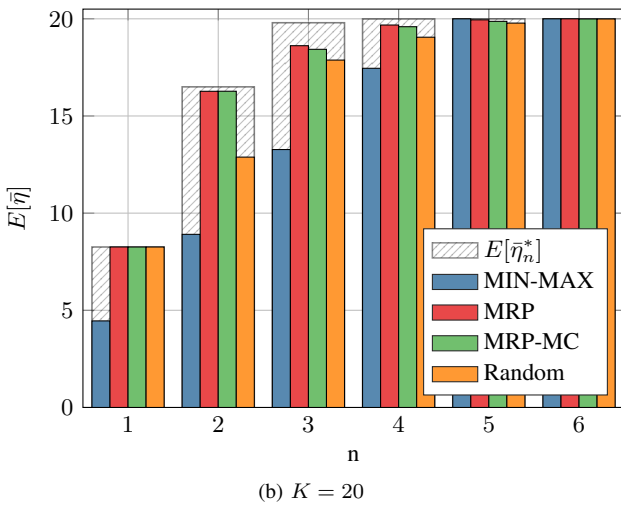
In the following, we will report the performance results of the MIN-MAX scheduling algorithm, the MRP scheduling algorithm, and the MRP-MC scheduling algorithm. Additionally, as means of comparison, we will consider both the random scheduling solution, described also in [16], and the (pure) carousel strategy, where all the SVs transmit the same packet sequentially. Indeed, this strategy yields $\bar{\tau} = \tau_{\max} = \tau(\mathbf{x}) = K$ rounds, $\forall \mathbf{x} \in \Omega$.

A. Maximum Latency

Fig. 2 shows the maximum latency $\max_t \tau_{\max}$, maximized over a time period T_{orb} , as a function of the number of



(a) $K = 10$



(b) $K = 20$

Fig. 3. Average received packets per round $E(\bar{\eta}_n)$ vs the round index n the MIN-MAX, MRP, MRP-MC, and random scheduling algorithms and average bound on the maximum average received packets (15), for $K = 10$ and 20.

packets per message, K , for the MIN-MAX, MRP, MRP-MC, and the random and carousel algorithms. We also show the (maximum of) the maximum latency achieved by bound (18). The MIN-MAX scheduling achieves the best performance for all values of K , while other solutions in general exhibit a higher average maximum latency. Moreover, the MIN-MAX scheduling always reaches the bound $\max_t[\tau_{\max}^*]$, computed from the maximum diversity scenario, which suggests the optimality of this approach in the worst-case scenario. As expected the worst performance is achieved by the carousel strategy, followed by the random scheduling solution.

B. Average Received Packets per Round

Fig. 3 shows the average received packets per round $E(\bar{\eta}_n)$ as a function of the round index n , for the various scheduling algorithms, focusing on $K = 10$ and 20 packets, as average and extreme values among the considered values of K . Also in this case, the average is taken with respect to the satellite positions in the T_{orb} interval. We also report the bound on the average received packets per round $\bar{\eta}_n^*$ obtained in Section III.

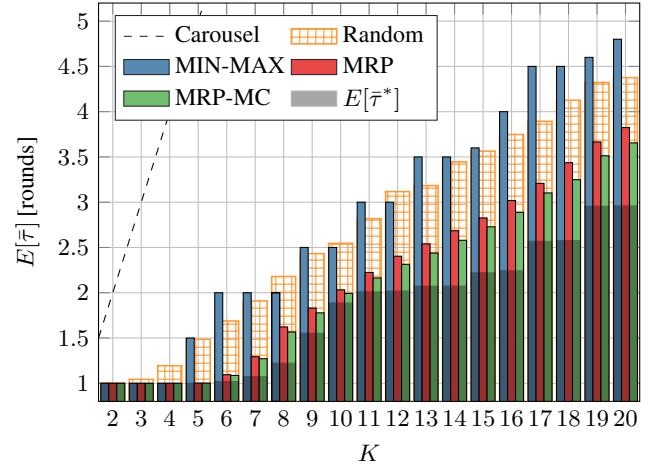


Fig. 4. Average latency $E[\bar{\tau}(\mathcal{P})]$ vs K , where the average is taken with respect to the positions of the satellites over their periods, for various scheduling approaches, and the average of the bound on the average latency $E[\bar{\tau}^*(\mathcal{P})]$.

Clearly, the MRP approach, which has been explicitly developed to maximize this metric, achieves the best results. Still, the MRP-MC that uses η as a secondary metric, achieves just a slightly lower value of $\bar{\eta}_n$. Here, the MIN-MAX scheduling is outperformed by all the others, including the random scheduling: this is because, by using the MIN-MAX and excluding low values of K , we do not transmit all K packets in one round, but just a subset per round (typically 4 packets). Unlikely the other solutions typically broadcast more packets per round. Moreover, these plots show that, in both cases, most of the packets are actually delivered in the first rounds; during the latter rounds, the effort is to deliver to the receivers that are still missing some of the packets, typically the receivers that have been in low visibility conditions. We did not report the performance of the carousel strategy: this would achieve $E(\bar{\eta}) = n$, thus obtaining the worst performances among the considered algorithms.

C. Average Latency

Fig. 4 shows the average latency $E[\bar{\tau}(\mathcal{P})]$ as a function of K , where, as described above, the average is taken with respect to the satellite positions, for the various scheduling approaches. We also report the average of the lower bound on the average latency $E[\bar{\tau}^*(\mathcal{P})]$, as from (18). The behavior of the average latency for the MIN-MAX scheduling is due to the fact that each round only 4 (or 5) packets are transmitted (see Fig. 2). For $K > 12$ all the scheduling strategies diverge from $E[\bar{\tau}^*(\mathcal{P})]$: still, we remark that the maximum diversity scenario, described in Section III-A is a non-realistic condition that clearly cannot be met in practice. In this case the MRP, and MRP-MC scheduling outperforms both the MIN-MAX and the random scheduling, achieving a lower average latency. In particular, the MRP-MC scheduling achieves the lowest average latency as maximizes the number of receivers that get all the packets at each round. The (pure) carousel strategy achieves by far the worst performance, with $E[\bar{\tau}] = K$.

D. Average Throughput

In order to summarize the results, we now introduce the throughput (in packets per round) as

$$\omega(\mathcal{P}) \triangleq \frac{1}{|A|} \int_A \frac{K}{\tau(\mathcal{P}, \mathbf{x})} d\mathbf{x}. \quad (27)$$

Fig. 5 shows the average throughput $E[\omega(\mathcal{P})]$ (27) as a function of K , where the average is taken with respect to the positions of the satellites over their periods as described above, for various scheduling approaches. First, we note that the average throughput grows linearly for all the scheduling techniques up to $K = 4$ since full coverage is always achieved. Interestingly, the highest throughput is achieved by the MRP-MC scheduling $K = 7$ packets at a time, achieving a throughput $\omega \approx 6.05$ packets per round. For comparison, the random scheduling solution only delivers (approximately) 3.62 packets per round.

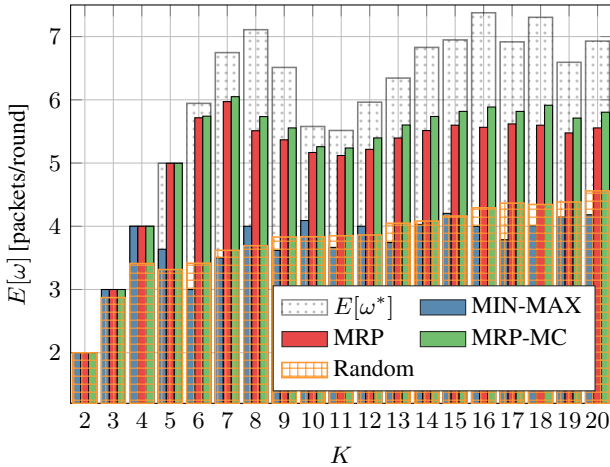


Fig. 5. Average throughput $E[\omega(\mathcal{P})]$ (27) vs K , where the average is taken with respect to the positions of the satellites over their periods, for various scheduling approaches.

For $K > 7$ the performance drop: indeed, by increasing K , the receivers need more rounds to obtain all the packets. A considerable fraction of the receivers has in view less than 7 SVs, thus, by having $K > 7$ we are forcing these receivers to wait for additional rounds. A similar behavior is shown also by the MIN-MAX algorithm: since we find that the highest K that allows all the receivers to obtain all the packet is (typically) $K = 4$ packets when using $K = 5$, the first round is devoted to delivering the first 4 packets; next, we use an additional round just to transmit the last packet. This also shows that, in general, increasing K does not necessarily increase the actual throughput.

E. Performance in realistic scenario

To provide realistic results, we test the proposed algorithms in a condition where we have a non-null page error rate (PER). We model each channel between an SV s and a receiver in position \mathbf{x} as an erasure channel, with a PER $q(\theta_s(\mathbf{x}))$, where $\theta_s(\mathbf{x})$ is the satellite elevation angle with respect to the receiver position at round n . The relationship

between PER and elevation angle has been derived from [22]. Considering the scheduling solution \mathcal{P} , the probability of having successfully received packet k within round n is

$$p_{k,n}(\mathbf{x}) = 1 - \prod_{m=1}^n \prod_{s \in S_{n,k}} q(\theta_s(\mathbf{x})). \quad (28)$$

Thus, the probability that the receiver in position \mathbf{x} reconstruct the message within round n is the probability of having received all the packets,

$$\tilde{p}_n(\mathbf{x}) = \prod_{k=1}^K p_{k,n}(\mathbf{x}). \quad (29)$$

Finally, we compute the fraction of receivers on A with probability at least $1 - \epsilon$ as

$$a_n^* = \sum_{\mathbf{x} \in \Omega} \mathbb{1}(\tilde{p}_n(\mathbf{x}) - (1 - \epsilon)) a(\mathbf{x}), \quad (30)$$

where $\mathbb{1}$ is the unit function, that is $\mathbb{1}(\cdot) = 1$ if and only if the argument is non negative. In particular, we pick $\epsilon = 1 \cdot 10^{-2}$.

Figure 6 shows the period average of the area (30) versus the round index n , choosing $K = 5$ and $K = 15$ as examples of low and high values of K , respectively. We considered the MIN-MAX, MRP, MRP-MC, and the random scheduling solutions. The same considerations apply to both plots: immediately after the first rounds, where the random and the MIN-MAX achieve the best results, the MRP-MC achieve the best results. This is particularly evident for the $K = 15$ case, where at the 6th round, the MRP-MC obtains $E[a_n^*] \approx 0.99$ while random scheduling achieves only 0.16. We did not report the performance of the carousel strategy: this yield $E[a_n^*] = \mathbb{1}(K)$, since the last packet is not transmitted until the K th round. For both $K = 5$ and $K = 15$, the carousel is outperformed by both MRP and MRP-MC.

VI. CONCLUSIONS

In this paper, we have proposed a model and several solutions for the multi-round message scheduling problem in GNSS. In detail, we defined proper metrics of the problem, i.e., coverage, average number of different received packets, and latency, and we analytically derived general lower and upper-performance bounds. These bound were derived under the maximum diversity conditions, i.e., assuming that there exists a scheduling such that all the receivers obtain a different packet from each satellite in view.

Next, we proposed one solution for the maximum latency minimization that, under conditions easily met in practice, achieves optimality at a reduced computational cost. Simulation results show that the MIN-MAX latency algorithm achieves the optimal maximum latency and is close to achieving the maximum latency achieved in the maximum diversity scenario. In particular, we were able to find a scheduling solution with a maximum latency of 4 rounds for scheduling 16 packets, which coincides with the optimal one.

Next, we proposed heuristic solutions for the average latency minimization, the MRP and the MRP-MC. These achieved interesting performances in terms of average latency and are close to reaching the optimal average latency. The best

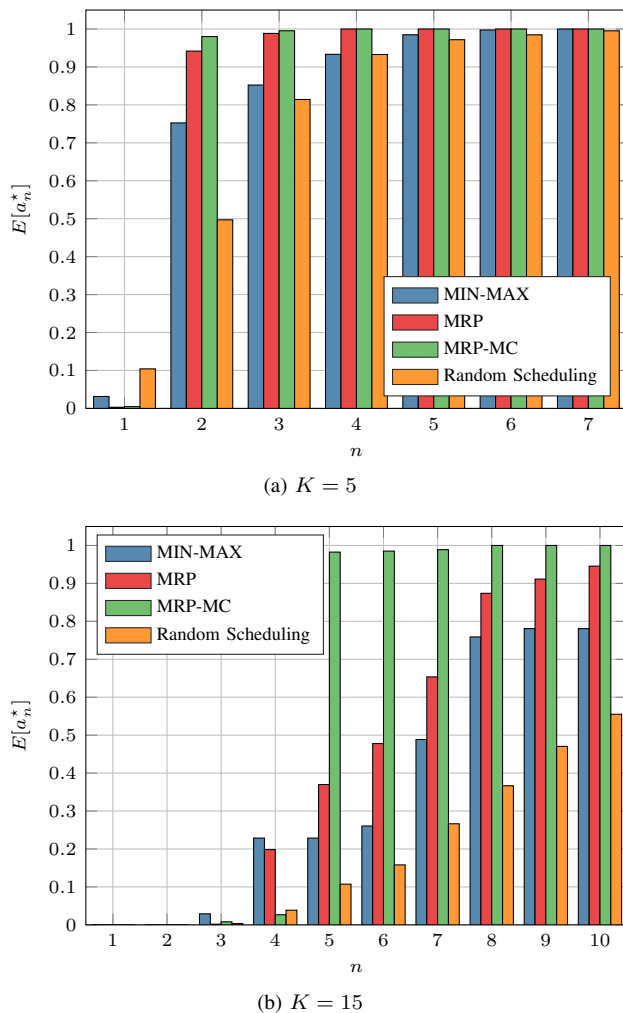


Fig. 6. Average received packets per round $E[a_n^*]$ vs the round index n the MIN-MAX, MRP, and MRP-MC and random scheduling algorithms, for $K = 5$ and $K = 15$.

results are achieved by the MRP-MC algorithm, the improved version of the MRP, where we also maximize the number of receivers obtaining the whole message. In particular, for $K = 7$, the MRP-MC delivers on average 6.05 packets per round, with an average latency of 1.27 rounds. For comparison, with the same parameters, the random scheduling solution achieves instead an average throughput of 3.62 packets per round and an average latency of 1.91 rounds.

Finally, we evaluate the performance of our scheme in a more realistic scenario, where each channel is modeled as an erasure channel. Still, the best results were achieved by the MRP-MC algorithm: for $K = 15$ by using the MRP-MC we get that $\approx 99\%$ of the receivers on Earth to obtain the packet with a success probability of 99% in 6 rounds, versus the 16% of the random scheduling algorithm.

In future works, we will design and develop solutions that minimize the latency also taking into account the PER in the objective function.

REFERENCES

[1] I. Fernández-Hernández, J. D. Calle Calle, S. Cancela, O. Pozzobon, C. Sarto, and J. Simón, "Packet transmission through navigation satel-

lites: A preliminary analysis using Monte Carlo simulations," in *Proc. the 25th European Navigation Conference (ENC)*, May 2017, pp. 298–304.

[2] European Agency for Space Programme (EUSPA), "SAR Galileo service definition document," Last access 03-10-2021. [Online]. Available: <https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo-SAR-SDD.pdf>

[3] Y. Wei, H. Liu, J. Ma, Y. Zhao, H. Lu, and G. He, "Global voice chat over short message service of BeiDou navigation system," in *Proc. of the 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, pp. 1994–1997.

[4] European Agency for Space Programme (EUSPA), "Galileo High Accuracy Service (HAS) - info note," Last access 03-10-2021. [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_Info_Note.pdf

[5] C. Hegarty and E. Kaplan, *Understanding GPS Principles and Applications*. Artech House, 2006.

[6] European Union, "Galileo - Open Service - Signal in Space Interface Control Document (OS SiS ICD v2.0)," 2016, Last access 24-09-2021. [Online]. Available: [\url{https://www.gsc-europa.eu/electronic-library/programme-reference-documents}](https://www.gsc-europa.eu/electronic-library/programme-reference-documents)

[7] Q. Chen, Z. Cai, L. Cheng, H. Gao, and J. Li, "Low-latency concurrent broadcast scheduling in duty-cycled multihop wireless networks," in *Proc. of the 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 851–860.

[8] S. Ji, R. Beyah, and Z. Cai, "Minimum-latency broadcast scheduling for cognitive radio networks," in *Proc. of the International Conference on Sensing, Communications and Networking (SECON)*. IEEE, 2013, pp. 389–397.

[9] A. Tarable, R. Andreotti, M. Luise, F. Zanier, and S. Cioni, "Link-layer coding for GNSS navigation messages," *Navigation*, vol. 65, Sept. 2018.

[10] European Union, "Galileo Open Service Navigation Message Authentication (OSNMA) - user ICD for the test phase." [Online]. Available: https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf

[11] D. Calle, S. Cancela, E. Carbonell, I. Rodríguez, G. Tobias, and I. Fernández-Hernández, "First experimentation results with the full Galileo CS demonstrator," in *Proc. of the 29th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, Sept. 2016.

[12] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simón, I. Rodríguez, and J. Calle, "A navigation message authentication proposal for the Galileo Open Service," *Navigation - Journal of The Institute of Navigation*, vol. 63, Mar. 2016.

[13] I. Fernández-Hernández, D. Calle, S. Cancela, A. Fernández, R. Martínez, G. Seco-Granados, and P. Walker, "Fountain codes for GNSS," in *Proc. of the 30th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+)*, 2017, pp. 1496 – 1507.

[14] T. Senni, I. Fernández-Hernández, and S. Cancela, "Message-to-satellite allocation strategies for long GNSS messages and application to the Galileo High Accuracy Service," in *Proc. International Conference on Localization and GNSS (ICL-GNSS)*, 2021, pp. 1–7.

[15] T. Senni, I. Fernández-Hernández, and S. Cancela, "Evaluation of the Hungarian algorithm for optimal transmission of the Galileo HAS message from multiple satellites," in *2022 International Conference on Localization and GNSS (ICL-GNSS)*, 2022, pp. 1–5.

[16] F. Ardizzon, N. Laurenti, and S. Tomasin, "Sub-messages scheduling in GNSS packet broadcasting by message splitting," in *Proc. of the International Conference on Localization and GNSS (ICL-GNSS)*, 2020, pp. 1–6.

[17] P. J. Teunissen and O. Montenbruck, *Handbook of Global Navigation Satellite Systems*. Springer, 2017.

[18] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*. Prentice-Hall, 1996.

[19] C. H. Papadimitriou, "On the complexity of integer programming," *J. ACM*, vol. 28, no. 4, p. 765–768, Oct. 1981. [Online]. Available: <https://doi.org/10.1145/322276.322287>

[20] C. Sarto, O. Pozzobon, S. Fantinato, S. Montagner, I. Fernández-Hernández, J. Simon, J. Calle, S. Díaz, P. Walker, D. Burkey, G. Seco-Granados, and E. Göhler, "Implementation and testing of OSNMA for Galileo," in *Proc. of the 30th International Technical Meeting of The Satellite Division of The Institute of Navigation (ION GNSS+)*, 09 2017.

[21] "GPS interface specification IS-GPS-200, revision N - August 2022," Last access 1-10-2022. [Online]. Available: <https://www.gps.gov/technical/icwg/IS-GPS-200N.pdf>

[22] P. Crosta and G. Pirazzi, "A simplified convolutional decoder for Galileo OS: performance evaluation with a Galileo mass-market receiver in live scenario," in *Proc. of the 8th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Dec 2016, pp. 1–8.

APPENDIX A

PROOF OF OPTIMALITY OF THE MIN-MAX LATENCY

To prove Proposition 2 we need the following Proposition.

Proposition 3. *If a scheduling sequence \mathcal{P} is such that $\eta_n(\mathcal{P}_n, \mathbf{x}) \geq C_{\min, n} \forall n$ and $\forall \mathbf{x} \in A$, then it minimizes the maximum latency, achieving the bound (12), i.e., $\tau_{\max}(\mathcal{P}_n) = \tau_{\max}^*$.*

Proof. From Proposition 1 we observe that, even in the maximum diversity scenario, $\hat{u}_n^* = 1$ implies $\bar{C}_n^*(\mathbf{x}) = K \forall \mathbf{x} \in A$, therefore also $C_{\min, n}^* = \min_{\mathbf{x} \in A} C_n^*(\mathbf{x}) = K$. Thus, we can write the lower bound to the latency for the receiver in position \mathbf{x} as

$$\tau_{\max}^*(\mathbf{x}) = T \min \left\{ n : C_{\min, n} = K \right\}, \quad (31)$$

which means that even in the maximum diversity scenario it is possible to deliver K packets to a receiver only if it had at least K satellites in view. Let n^* be the minimum number of rounds that satisfies condition (31) for all the receivers on A . Exploiting the hypothesis, after exactly n^* rounds by using scheduling \mathcal{P} that satisfies the hypothesis, we will have delivered at least $C_{\min, n^*} = K$ packets to all the receivers. Thus, \mathcal{P} is indeed the scheduling that achieves $\tau_{\max}(\mathcal{P}_n) = \tau_{\max}^*$. \square

Now we can recall Proposition 2 and prove it.

Proposition 2. *If (20) holds, the proposed min-max latency algorithm achieves the optimal latency, i.e., $\tau_{\max}(\mathcal{P}_n) = \tau_{\max}^*$.*

Proof. The proposed Algorithm 1 delivers K_m packets at round m , where K_m is the largest number of packets that can be delivered in round m with full coverage, i.e., $K_m = \min_{\mathbf{x} \in A} \sum_{s \in S} v_{s, m}(\mathbf{x})$. Hence, after n rounds, from (20) we will deliver

$$\eta_n(\mathcal{P}_n, \mathbf{x}) = \sum_{m=1}^n K_m = \sum_{m=1}^n \min_{\mathbf{x} \in A} \sum_{s \in S} v_{s, m}(\mathbf{x}) = C_{\min, n}. \quad (32)$$

However this also satisfies the requirements of Proposition 3, thus the proposed min-max latency algorithm achieves optimality. \square

APPENDIX B

BILP FORMULATION OF PROBLEM 1

First, we introduce the indicator variables

$$y_{s, k, n} = \begin{cases} 1 & \text{if } s \in S_{k, n}, \\ 0 & \text{if } s \notin S_{k, n}, \end{cases} \quad (33)$$

which converts the search of the set \mathcal{S}_n into the choice of the variables $y_{s, k, n}$ and, Problem (21) can be written as the following BILP problem

Problem 1. *At round n , given the coefficients (for $\mathbf{x} \in \Omega$)*

$$v_{s, n}(\mathbf{x}) \in \{0, 1\} \quad s \in S \quad (34a)$$

$$u_{k, n-1}(\mathcal{P}_{n-1}, \mathbf{x}) \in \{0, 1\} \quad k \in \{1, \dots, K\}, \quad (34b)$$

$$a(\mathbf{x}) \in [0, 1] \quad (34c)$$

maximize over the variables (for $k \in \{1, \dots, K\}$)

$$y_{s, k, n} \in \{0, 1\} \quad s \in S, \quad (34d)$$

the objective function

$$\bar{\eta}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}) = \sum_{k, \mathbf{x}} a(\mathbf{x}) u_{k, n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}), \quad (34e)$$

under the following constraints $\forall k \in \{1, \dots, K\}$, $\mathbf{x} \in \Omega$, $s \in S$.

$$\sum_k y_{s, k, n} = 1, \quad (34f)$$

$$\sum_s y_{s, k, n} v_{s, n}(\mathbf{x}) + u_{k, n-1}(\mathcal{P}_{n-1}, \mathbf{x}) \geq u_{k, n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}). \quad (34g)$$

Notice that (34e) is exactly equation (4) rewritten for the fixed sampling set Ω .

APPENDIX C

BILP FORMULATIONS OF PROBLEMS 2 AND 3

In this Section we introduce the BILP formulations for Problems 2 and 3. First, we consider the single round coverage maximization of Problem 2, used to derive (25b).

Problem 2 (Single round coverage maximization). *At round n given the coefficients for $\mathbf{x} \in \Omega$*

$$v_{s, n}(\mathbf{x}) \in \{0, 1\} \quad s \in S, \quad (35a)$$

$$u_{k, n-1}(\mathcal{P}_{n-1}, \mathbf{x}) \in \{0, 1\} \quad k \in \{1, \dots, K\}, \quad (35b)$$

$$a(\mathbf{x}) \in [0, 1] \quad (35c)$$

for $k \in \{1, \dots, K\}$ maximize over the variables

$$y_{s, k, n} \in \{0, 1\} \quad i \in S, \quad (35d)$$

the coverage objective function

$$\bar{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}) = \sum_{\mathbf{x} \in A} a(\mathbf{x}) \hat{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}), \quad (35e)$$

under the constraints (for all $k \in \{1, \dots, K\}$, $\mathbf{x} \in \Omega$, $s \in S$),

$$\sum_k y_{s, k, n} = 1, \quad (35f)$$

$$\sum_s y_{s, k, n} v_{s, n}(\mathbf{x}) + u_{k, n-1}(\mathcal{P}_{n-1}, \mathbf{x}) \geq u_{k, n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}), \quad (35g)$$

$$\sum_k u_{k, n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}) \geq K \hat{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}). \quad (35h)$$

Finally, we report the Single round throughput maximization with constrained coverage of Problem 3 that solves (25a).

Problem 3 (Single round throughput maximization with constrained coverage). *At round n given the coefficients (for $\mathbf{x} \in \Omega$)*

$$v_{s,n}(\mathbf{x}) \in \{0, 1\} \quad s \in S \quad (36a)$$

$$u_{k,n-1}(\mathcal{P}_{n-1}, \mathbf{x}) \in \{0, 1\} \quad k \in \{1, \dots, K\}, \quad (36b)$$

$$a(\mathbf{x}) \in [0, 1], \quad (36c)$$

$$\bar{u}_n \in [0, 1] \quad (36d)$$

maximize over the variables

$$y_{s,k,n} \in \{0, 1\} \quad k \in \{1, \dots, K\}, s \in S, \quad (36e)$$

the objective function

$$\bar{\eta}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}) = \sum_{\mathbf{x} \in \Omega} a(\mathbf{x}) \sum_k u_{k,n}(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}), \quad (36f)$$

under the constraints (35f),(35g),(35h) and

$$\sum_{\mathbf{x} \in \Omega} a(\mathbf{x}) \hat{u}_n(\{\mathcal{P}_{n-1}, \mathcal{S}_n\}, \mathbf{x}) \geq \bar{u}_n \quad (36g)$$