Link-Layer Coding for GNSS Navigation Messages

A. Tarable, R. Andreotti, M. Luise, F. Zanier, S. Cioni

May 12, 2019

Abstract. In this paper, we face the problem of ensuring reliability of GNSSs in harsh channel conditions, where obstacles and scatters cause long outage events that cannot be counteracted with channel coding only. Our novel approach, stemming from information-theoretic considerations, is based on link-layer coding (LLC). The latter allows to significantly improve the efficiency in terms of time-to-first-fix (TTFF) with respect to current operational GNSSs, which adopt carousel transmission. First, we investigate the maximum theoretical LLC gain under different Land Mobile Satellite (LMS) channel conditions. Then, some practical LLC coding schemes, namely, Fountain codes and a novel LDPC plus low-rate repetition coding, are proposed and tested in realistic single- and multi-satellite LMS scenarios, considering the Galileo I/NAV message as study case. Simulation results show that our designed schemes largely improve on carousel transmission and achieve near-optimal performance with limited complexity increase. Also, back-compatibility of LLC is assessed w.r.t. present-time GNSS specifications.

I. Introduction and motivation

In recent years, the whole world has witnessed a steady growth of the relevance of laptops, tablets and smart phones in everyday life. Moreover, the paradigm of Internet of Things (IoT) [1] has now become a standard *de facto* of the interconnected ecosystem of electrical and electronic devices among which we are living. In both cases, two key challenges need to be addressed in order to allow a successful exploitation of these technologies.

- Battery life must be prolonged as much as possible by an efficient use of resources, a smart network architectural design and the capability for a device to stay idle whenever it is possible.
- Connected devices take advantage of knowing their position for several reasons. First of all, network operators enable a host of location-based services such as turn-by-turn navigation, shopping ads, health-care systems, and others. Moreover, in IoT it is often needed for a device to know and possibly communicate its precise position, as in automated factory stocking [2]. As a further example, in wireless sensor networks, a sensor needs to measure environmental variables (such

A. Tarable is with IEIIT/CNR, Torino, Italy; R. Andreotti is with WISER srl, Livorno, Italy; M. Luise is with the University of Pisa, Italy; F. Zanier and S. Cioni are with ESA/ESTEC, The Netherlands

as temperature or air pollution) and transmit the measurements together with the specific location they refer to [3].

To address the above challenges, connected devices are equipped with receivers of positioning and/or navigation systems, which enable indicating the device position in a very short time. While terrestrial positioning is emerging as a competitive solution, especially in urban and crowded environments, where cellular networks and Internet connectivity make high-precision positioning possible even indoor and in urban canyons [4], GNSSs still represent the only reliable systems in several important scenarios. First, in developing countries or in rural environments, there might not be a good-quality connection to a terrestrial network. Second, in disaster scenarios, the terrestrial infrastructures may be seriously damaged and, as a consequence, out of order. Thus, i) GNSSs represent a key technological player to reduce the digital divide between rich and poor areas of the world, and ii) they are often the only system rescuers can rely on for emergency relief. Because of the fundamental importance of these missions, GNSSs must be designed as autonomous systems, even if in less critical conditions assisted GNSSs have become the winning paradigm [4]. As a further motivation for the design of autonomous GNSS systems, [5] lists a series of inefficiencies of assisted GNSSs, which emerge as a result of the strong dependence on the cellular network to obtain assisting data.

It is clear that, in scenarios where interconnected, battery-powered sensors must save energy to increase their autonomous life, next-generation GNSS design should keep into account such constraints. In particular, all those devices that, like in IoT, are on for a limited amount of time, and need to carefully manage their energy expenditure, would benefit from having long-lasting constellation information available, which would allow for a "warm" start and reduce the *time-to-first-fix* (TTFF)[1] [7]. For autonomous GNSSs, solutions like the one presented in [8] foresee a different structure of the GNSS message to solve the problem of fast TTFF. In line with this solution, we consider in this paper adding an *extra coding layer* on top of the channel code, in order to speed up the reception of the GNSS message. Being at the link layer, this addition requires a smaller change of the existing GNSS systems (both at the transmitter and at the receiver) with respect to [8], thus partly preserving backward compatibility. On the other hand, we will show in the paper that a considerable link-layer coding gain can be achieved by relatively long GNSS messages parts. Examples of these message parts are the almanac and the so-called long-term ephemeris (LTE), which are ephemeris data for the entire constellation, whose validity could last for a couple of weeks. Because of the long time that is needed with the current GNSS systems

[1]Low-cost IoT nodes such as sensors, etc., may also suffer from poor internal clock quality, whose phase noise indeed represents a major factor for inefficiency in terms of TTFF. See [6] for a way to obviate to such problems.

to retrieve these message parts, they are typically considered as downloaded from terrestrial links in assisted GNSSs, or entirely disregarded [9]. One of the goals of our paper is to explore the possibility of improving the reception of almanac and LTE data, thus making them a potential key element of next-generation autonomous GNSSs.

A. Link-layer coding for message protection in GNSSs

It is clear from the above discussion that, for almanac and LTE data to be efficient in energy saving, the entire reception should be insured even in "harsh" channel conditions, i.e., when fading and shadowing are particularly severe. Thus, the information conveyed by the GNSS message should be adequately protected. In current satellite systems, *Robustification* of the navigation message is usually achieved through coding/interleaving at the physical layer [10], but the particularly "harsh" environments just mentioned require more. Beyond physical-layer protection techniques, a typical way of counteracting outages for those message parts like the almanac is *carousel* broadcasting, i.e., the message is repeated a number of times without any change before being updated, to make sure it is correctly received [11]. If the receiver fails to decode the almanac in its first attempt, it has to wait a whole "turn of the carousel" before having the chance of decode it again.

To be more specific, in current GNSSs the diverse messages are organized in a hierarchical structure and are broadcast, as already mentioned, in a carousel fashion. Prior to be handed to the physical layer transmitter (called the Navigation Signal Generation Unit, NSGU), the navigation message is divided into smaller messages, often called frames, which in turn may be further divided into smaller subframes and so on. In the following, we will refer to a *page* as the smallest unit of the navigation message that is formatted and then handed to the NSGU, as defined in the current GALILEO SIS-ICD specifications [13]. Each page is typically equipped with cyclic redundancy check (CRC) symbols to check the absence of decoding errors. Prior to physical layer transmission, the page is further encoded with a forward error correction (FEC) or *channel* code to make it as robust as possible against any impairment of the physical radio channel. When the radio channel is in bad conditions, and/or the channel code is not powerful enough to prevent decoding errors, the CRC fails, the page is marked as *errored* and discarded by the receiver (in channel coding parlance, it is *erased*). In such a case, the receiver awaits for the next retransmissions of the message in order to try to recover the lost page. In harsh environments, the fraction of lost pages at the physical layer may be not negligible, thus increasing the overall *time-to-retrieval* (TTR) of the message of interest (i.e., the time that is necessary for the receiver to recover all of the message pages from the start of reception), and negatively affecting the

overall receiver TTFF.

A suitable technology to improve on this situation is *link-layer coding* (LLC), that consists in adding a *smarter* layer of protection on the message pages - such technology is not at the moment implemented in any GNSS system. The rationale of LLC is that the diverse pages are encoded at the link layer by the formatter of the navigation message, i.e., before being processed as usual by the NSGU at the physical layer. In particular, the trivial carousel repetition of a page is replaced by a smarter encoding strategy that we will detail later on. The advantage of this smarter strategy is that the receiver, after the erasure (loss) of a certain page in the message, does *not* need to wait for the whole repetition of the message in the carousel. On the contrary, owing to LLC, the decoder can recover the lost page as soon as a sufficient (smaller-than-carousel) number of subsequent pages are correctly retrieved. Actually, carousel transmission can be seen as a sort of primitive form of LLC, namely, *LLC repetition encoding*, which turns out to be optimal only when the overall page loss rate is very, very small, whilst it becomes largely inefficient in bad channel conditions (i.e., non-negligible page loss rate).

The most efficient and well known technology for LLC is *Rateless coding*, already in use in wireless cellular communications. Rateless codes generate a potentially endless stream of coded symbols until the destination is able to retrieve the original information message. Rateless codes provide reliable communications without requiring any prior knowledge about the status (good/bad) of the channel, and are asymptotically optimal for every channel condition [12]. They also perform remarkably well over *erasure channels*, which is exactly our case of link-layer, page-by-page communication channel described above. In wireless communications, rateless codes are mainly considered for multicast multimedia applications. In particular, Raptors codes [19] have been standardized for the 3GPP Multimedia Broadcast/Multicast Service (MBMS) service [20] and for DVB IP-datacast services [21]. Despite being originally designed for erasure channels, they have also lately been considered as physical-layer FECs for communication over Gaussian [22] and fading [23] channels. The GNSS message represents an ideal application for rateless coding, since, being generally transmitted from multiple satellites to multiple receivers, it fits in a multicast communication scenario, for which rateless codes are known to be a very good solution.

To our knowledge, LLC coding is considerably less popular in satellite communications and/or GNSS. Such technology has been proposed into the DVB-SH standard [10], where an optional Raptor code is introduced to mitigate the effects of deep-shadowing events [10, 14]. Rateless coding is considered at the application layer in [15] for DVB-S2 satellite communications, and in [16], where it is added on top of a low-redundancy channel code as an additional protection layer to attain a more

reliable DVB-S transmission. In the particular case of GNSSs, [17] introduces a simple LLC scheme to be applied to the downlink message of Galileo. In the latter reference, the proposed LLC scheme is dubbed "network coding", because it involves a multisatellite scenario.

Our paper represents a first attempt to fit LLC technology to GNSS, showing some possible classes of LLC techniques and assessing their merits in a realistic Land-Mobile Satellite (LMS) channel. As main figure-of-merit of LLC, we first assess the maximum theoretically achievable TTR gain in different scenarios w.r.t. to standard carouseling; then, we devise practical techniques that exhibit near-optimal performance, also assessing their complexity increase and back-compatibility with respect to present-time GNSS specifications. Our baseline for comparisons is the current Galileo I/NAV message [13], and we show that, for this case, the highest gains are obtained for "long" messages, i.e., those whose number of pages ranges from a few tens to a few hundreds. We also focus on messages sent from all satellites in the constellation, like almanac data, to exploit *satellite diversity* and the capability of link-layer techniques to be "agnostic" on the actual source of the received message, also proposing suited, back-compatible, extensions of the current almanac of Galileo I/NAV to further improve the TTR gain.

Although LLC solves primarily the problem of TTFF/TTR reduction for autonomous GNSS positioning in harsh environments, it can be profitably employed in GNSS use cases such as Assisted GPS (A-GPS) [18], in which some terrestrial transmitters provide auxiliary links for performance enhancement. Simulation results will show that there is a significant advantage in using LLC even if an auxiliary link has endowed the receiver with a relevant amount of side information.

The paper is organized as follow. In Sect. II, the main concepts of LLC are described and then particularized for the specific GNSS scenario. Sect. III identifies the maximum achievable gain that *any* LLC scheme can provide, given the link-layer channel statistics. In Sect. IV, we will introduce some practical LLC techniques, and derive their performance in a realistic GNSS-LMS scenario, considering the Galileo I/NAV message as our case of study. The final section will be devoted to a summary and to the usual conclusions.

II. Link-Layer Communication Channel and Coding Schemes

In this section, we introduce our model for the link- and physical-layer of a generic communication link, as is depicted in Fig. 1. As is seen, the digital message to be transmitted (our overall navigation message) is composed of K equal-sized information pages. Each page is first processed by the link-layer encoder (whose operation and scope we will discuss in a moment) that adds a first layer of redundancy, and produces $N \geq K$ link-layer coded pages. The coded pages undergo insertion of the

cyclic redundancy check (CRC), forward error correction (FEC) encoding, modulation (which may include spectrum spreading) and eventually radio transmission on a Land-Mobile Satellite (LMS) channel. After radio propagation, the radio signal is synchronized and demodulated in the receiver, then the digital message is decoded and CRC-checked. If the check is successful, the corresponding page is correctly retrieved and handed over to the Link-Layer decoder, otherwise it is *erased*. For this reason, the link layer sees a so-called *Page Erasure Channel* (PEC), in that each page in the message is either correctly received or entirely discarded.

We emphasize that the above model can refer either to the whole GNSS message or to a message part only (which, for brevity, throughout the paper, will be called message as well). Indeed, if different message parts are encoded separately at link layer, a receiver that is only interested in one of them can simply skip the uninteresting parts and collect and decode only those pages that are located in those frame slots that contain the message of interest. However, as we will see in the next sections, the performance of the LLC scheme will crucially depend on the features of encoded message such as length and dissemination timing.

In current GNSSs, the K pages composing the navigation message are continuously retransmitted a pre-defined number of times ρ before being updated: we call this *carouseling*. The receiver must wait for the entire retransmission of the whole navigation message before being able to recover a previously erased page. It is possible to avoid the inefficiency of carousel transmission by adopting a more refined LLC scheme, which is able, as we will show, to significantly reduce the time needed to retrieve the full message. Just to make an example, the Galileo I/NAV message for the almanac is composed of $K = 48$ pages by 2 seconds each, organized in 2 pages per sub-frame, with a carousel transmission consisting of 24 sub-frames (total 96 second). Or, for GPS L1 C/A, the same message is composed of $K = 50$ pages (almanac and other constellation information), transmitted in 2 pages per frame, with a carousel consisting of 25 frames.

This said, as already mentioned in the Introduction, our attention will be focused on the following specific LLC schemes:

- *Carousel transmission*, which represents the legacy case for GNSS, and can be seen as a trivial low-rate repetition LLC scheme (the same pages are continuously retransmitted until the navigation message is updated, so that the coding rate is $1/\rho$).

- *Ideal LLC*, where the reception of *any* K encoded pages out of the overall set of N is sufficient to retrieve the original K information pages. This theoretical scheme will be considered as our

performance benchmark for any practical LLC technique.

- *Rateless codes*, a class of erasure-correcting codes that generates a theoretically endless sequence of independent encoded symbols according to a predefined probability distribution, and in particular LT and Raptor codes.

- *A novel LDPC scheme*, obtained by the concatenation of a short-length LDPC code with a low-rate carousel transmission.

The performance and the features of such schemes are described in detail in Sect. III. and IV., respectively.

III. Theoretical Performance of LLC Coding

In this section, we evaluate the theoretically achievable gain of the LLC schemes mentioned in the previous section, including carousel transmission and ideal encoding. Recall that by ideal LLC, we mean a scheme in which every successfully decoded page is useful, and any set of K decoded pages is sufficient to recover the whole message. In the coding theory literature, ideal encoding is also called *maximum-distance separable* (MDS) coding, and its performance achieves the Singleton bound [24, p. 33]. The achievable LLC gain is derived in terms of improvement of the time-to-retrieval (TTR) wrt carouseling. The TTR is formally defined as the time interval required at the receiver to retrieve the whole message, counted from the start of receiver operation (reception start time). In our analysis, the TTR turns out to be a random variable, since it depends on other random quantities, notably the reception start time and the pattern of page erasure events experienced at the receiver, so that the actual performance metrics will be the *average* TTR.

The reference system we will consider for TTR computation follows the block reception timing format shown in Fig. 2, and is characterized by the following parameters.

- Blocks (or coded pages) are indexed by integers $0, 1, \ldots$ and each of them is transmitted on a time window with (constant) duration T_P.

- Reception of block i starts at time $\tau_i - T_p$ and ends at time τ_i, $i = 0, 1, \ldots$ (where we have set an arbitrary origin of the time axis). For simplicity, we will suppose that $\tau_0 = T_p$.

- The time interval between block $i-1$ and block i is denoted by $\delta_i = \tau_i - \tau_{i-1}$, $i = 1, 2, \ldots$ and in general is *not* constant.

- $p_0, p_1, \ldots$ are the page error rate (PER) values for the blocks, as resulting from physical layer demodulation/decoding. Such values depend on the status of the physical channel in each block reception time and are *not* constant - they are in general correlated random variables.

- The reception start time is supposed to be uniformly distributed in the time interval $[0, T]$, where $T = \tau_{I_0} - T_P$ is the start time for reception of block I_0, for some integer $I_0 \geq 0$.

In all practical cases, the dissemination follows a periodic pattern. Although the analysis below does not need such hypothesis, we will assume that the dissemination period is exactly T, i.e., I_0 above is the number of message pages in a single repetition period, so that the reception start time is uniformly distributed over one such period. This in turn implies that $\delta_{I_0+j} = \delta_j$ for all j.

We start by deriving in the next subsection the probability density function (pdf) of the TTR for a general LLC scheme, then we will specialize the computations for ideal encoding and carousel transmission. In the following, we will suppose that a block must be *entirely* received for the physical-layer decoder to be able to decode it. In other words, if reception starts within a page, such page will be considered lost.

A. *Pdf of the TTR*

To start with, suppose that the first page that is fully received is the one with a certain index $i_0 > 0$. This happens when the reception start time belongs to the interval $[\tau_{i_0-1} - T_P, \tau_{i_0} - T_P]$, with duration δ_{i_0}. Let us define $\pi_K(i, i_0)$ as the probability that the message is retrieved at the reception of the i-th page ($i \geq i_0 + K - 1$), given that the first received page is the i_0-th one. Then, the pdf of the TTR, conditional on the value of i_0, is given by

$$f_{\mathrm{TTR}}(\theta|i_0) = \frac{1}{\delta_{i_0}} \sum_{i=i_0+K-1}^{+\infty} \pi_K(i, i_0) u_{\delta_{i_0}}(\theta - \tau_i + \tau_{i_0} - T_P) \tag{1}$$

where we have defined the function

$$u_X(x) = \begin{cases} 1, & x \in [0, X] \\ 0, & \text{otherwise.} \end{cases}$$

The unconditional pdf of the TTR will then be given by

$$\begin{aligned} f_{\mathrm{TTR}}(\theta) &= \sum_{i_0=1}^{I_0} \frac{\delta_{i_0}}{T} f_{\mathrm{TTR}}(\theta|i_0) \\ &= \frac{1}{T} \sum_{i,i_0} \pi_K(i,i_0) u_{\delta_{i_0}}(\theta - \tau_i + \tau_{i_0} - T_P) \end{aligned} \tag{2}$$

The pdf in (2) is an average of uniform distributions, over the intervals $[\tau_i - \tau_{i_0} + T_P, \tau_i - \tau_{i_0-1} + T_P]$, $i_0 = 1, \ldots, I_0$, $i \geq i_0 + K - 1$, with weights proportional to $\pi_K(i, i_0)$, and applies to *any* kind of LLC encoding. What is actually impacted by the encoding strategy is the value of $\pi_K(i, i_0)$. For a given statistics of the PER vector $\mathbf{p}$, we have

$$\pi_K(i, i_0) = \mathbb{E}_{\mathbf{p}} \pi_K(i, i_0|\mathbf{p}) \tag{3}$$

where $\mathbb{E}_{\mathbf{p}}$ denotes the expectation with respect to $\mathbf{p}$.

In the next two subsections, we will compute $\pi_K(i, i_0|\mathbf{p})$ in the case of ideal and carousel encoding.

B. Computation of $\pi_K(i, i_0|\mathbf{p})$ for ideal encoding

In order to compute $\pi_K(i, i_0|\mathbf{p})$ for ideal encoding, let us define a length-K column vector[2] $\boldsymbol{\beta}(i, i_0)$ with the following meaning: component j of $\boldsymbol{\beta}(i, i_0)$, $j = 1, \ldots, K$, denoted $\beta_j(i, i_0)$, is the probability that $j - 1$ pages have been recovered up to time τ_i (i.e., upon reception of page i), given that the first received page is the i_0-th one. For $i \geq i_0$,

$$\boldsymbol{\beta}(i, i_0) = \begin{bmatrix} p_i & 0 & 0 & \cdots & 0 \\ 1 - p_i & p_i & 0 & \cdots & 0 \\ 0 & 1 - p_i & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 1 - p_i & p_i \end{bmatrix} \boldsymbol{\beta}(i - 1, i_0) \tag{4}$$

with initial condition $\boldsymbol{\beta}(i_0 - 1, i_0) = [1, 0, 0, \ldots, 0]^T$. Indeed, after receiving the i-th page, the probability of having recovered $j > 0$ pages is equal to

$$\beta_{j+1}(i, i_0) = p_i \beta_{j+1}(i - 1, i_0) + (1 - p_i) \beta_j(i - 1, i_0) \tag{5}$$

[2]We omit the dependence on $\mathbf{p}$ for ease of notation whenever it is not strictly necessary.

while the probability of not having recovered any page is equal to

$$\beta_1(i, i_0) = p_i \beta_1(i-1, i_0) \tag{6}$$

Matrix (4) can be interpreted as the state-transition probability matrix of the time-varying Markov chain represented in Fig. 3, where the label of each state is equal to the number of recovered pages.

Finally, for ideal encoding, $\pi_K(i, i_0|\mathbf{p})$ is equal to the probability of stepping from $K-1$ recovered pages to K recovered pages upon reception of page i. Therefore,

$$\pi_K^{\text{id}}(i, i_0|\mathbf{p}) = (1-p_i)\beta_K(i-1, i_0) \tag{7}$$

C. Computation of $\pi_K(i, i_0|\mathbf{p})$ for carousel encoding

Similarly to the previous subsection, let us define a length-K vector $\boldsymbol{\gamma}(i, i_0)$ with the following meaning: component j of $\boldsymbol{\gamma}(i, i_0)$, $j = 1, \ldots, K$, denoted $\gamma_j(i, i_0)$, is the probability that the j-th message page has not been yet recovered up to time τ_i, given that the first received page is the i_0-th one. Define $\gamma_j(i_0 - 1, i_0) = 1$, $j = 1, \ldots, K$. Then, for $i \geq i_0$,

$$\gamma_j(i, i_0) = \begin{cases} \gamma_j(i-1, i_0), & j \neq m(i) \\ p_i \gamma_j(i-1, i_0), & j = m(i) \end{cases} \tag{8}$$

where $m(i) \in \{1, \ldots, K\}$ is the index of the information page corresponding to the i-th transmitted page. The probability that the message is retrieved following the reception of the i-th transmitted page is the product of 1) the probabilities that *all* information pages *but* the $m(i)$-th one have already been recovered with 2) the probability that the $m(i)$-th information page is recovered *exactly* upon reception of transmitted page i:

$$\pi_K^{\text{car}}(i, i_0|\mathbf{p}) = (1-p_i)\gamma_{m(i)}(i-1, i_0) \prod_{j \neq m(i)} (1 - \gamma_j(i, i_0)) \tag{9}$$

D. The case of uncorrelated PER values

In this section, we compute the expression of $\pi_K(i, i_0)$ for the two considered LLC strategies, in the simple case where the PER values are *uncorrelated* random variables (rv's). The following proposition holds true:

Proposition 1 *Assume the PER vector* $\mathbf{p}$ *is made up of uncorrelated rv's and has mean* $\overline{\mathbf{p}} = (\overline{p}_0, \overline{p}_1, \dots)$. *Then, for both ideal and carousel encoding,* $\pi_K(i, i_0) = \mathbb{E}_{\mathbf{p}} \pi_K(i, i_0|\mathbf{p}) = \pi_K(i, i_0|\overline{\mathbf{p}})$.

Proof: It is easy to verify that both (7) and (9) are linearly dependent on p_i, for any i. Thus,

$$E_{p_i} \pi_K(i, i_0|\mathbf{p}) = \pi_K(i, i_0|\mathbf{p})\Big|_{p_i = \overline{p}_i}$$

Since the PER values are uncorrelated rv's, the average with respect to $\mathbf{p}$ reduces to a cascade of averages with respect to p_i, $\forall i$ and the proposition follows. ■

1) The time-invariant scenario: We want to pursue a little further the case of time invariance, i.e., $\overline{p}_i = \overline{p}$, for every i. It is easy to verify that, in such scenario, $\pi_K(i, i_0)$ will only depend on the difference $i - i_0$. Let $\pi_K(L) = \pi_K(i_0 + L - 1, i_0)$, where L is the number of received pages since reception start. As a result of Proposition 1, we can easily compute $\pi_K(L)$ for ideal encoding:

$$\pi_K^{\text{id}}(L) = \binom{L-1}{K-1} (1 - \overline{p})^K \, \overline{p}^{L-K} \tag{10}$$

which is a negative binomial distribution, characterizing the number of i.i.d. Bernoulli trials before a specified number of successes (in our case, K) occurs - a result that could be expected and checks the sanity of our approach. The average number of received pages that is needed in order to retrieve the message will then be given for ideal encoding by

$$\overline{L}^{\text{id}} = \sum_{L=K}^{+\infty} L \pi_K^{\text{id}}(L) = K + K \frac{\overline{p}}{1 - \overline{p}} \tag{11}$$

Instead, for carousel encoding, let $\nu = \lceil L/K \rceil$ and $J = L - (\nu - 1)K$. Then, upon receiving transmitted page $i = i_0 + L - 1$, there will be J information pages (among which the $m(i)$-th one), $1 \leq J \leq K$, that have been received ν times since the reception start, while the other $K - J$ have been received $\nu - 1$ times. Thus, from (8) and (9), we obtain by direct computation

$$\pi_K^{\text{car}}(L) = (1 - \overline{p}) \, \overline{p}^{\nu - 1} \, (1 - \overline{p}^{\nu})^{J-1} \left(1 - \overline{p}^{\nu-1}\right)^{K-J} \tag{12}$$

Moreover, a tedious but straightforward computation yields in this case the following expression for the

average number of received pages needed to retrieve the message:

$$\overline{L}^{\mathrm{car}} = \overline{L}^{\mathrm{id}} + \sum_{\ell=2}^{K} \binom{K}{\ell} (-1)^{\ell} \frac{\overline{p}^{\ell-1}}{1-\overline{p}^{\ell-1}} \tag{13}$$

The sum in (13) represents the penalty the we experience with carousel encoding, in terms of average number of pages to be received, w.r.t. ideal encoding, in the time-invariant uncorrelated scenario.

If $\overline{p} = \epsilon$, with $\epsilon \to 0$, i.e., in asymptotically good channel conditions, we have $\overline{L}^{\mathrm{id}} \to K$ and

$$\overline{L}^{\mathrm{car}} - \overline{L}^{\mathrm{id}} = \binom{K}{2} \epsilon + o(\epsilon) \tag{14}$$

which means that the performance gap between the two LLC techniques tends to zero linearly with $\overline{p}$, but increases quadratically with increasing K. Instead, with $\overline{p} = 1 - \epsilon$, with $\epsilon \to 0$, i.e., with asymptotically bad conditions, we have

$$\overline{L}^{\mathrm{car}} - \overline{L}^{\mathrm{id}} = \frac{1}{\epsilon} \sum_{\ell=2}^{K} \binom{K}{\ell} (-1)^{\ell} \frac{1}{\ell - 1} + o\left(\frac{1}{\epsilon}\right) \tag{15}$$

Owing to the following identity involving harmonic numbers [25],

$$H_n = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} \frac{1}{k} \tag{16}$$

we obtain

$$\overline{L}^{\mathrm{car}} - \overline{L}^{\mathrm{id}} = \frac{1}{\epsilon} \left[K(H_{K-1} - 1) + 1 \right] + o\left(\frac{1}{\epsilon}\right) \tag{17}$$

which means that the performance gap increases like $1/\epsilon$, at a ratio which is $\mathcal{O}(K \log K)$, since $H_n = \mathcal{O}(\log n)$. Thus, in both asymptotically good and bad channel conditions, the penalty of carousel transmission is larger, the larger the carousel length K is.

In order to better understand the achievable LLC gain, we have numerically simulated a scenario with the following parameters.

- The transmitted pages are equispaced, i.e., $\tau_i = i\Delta + T_P$, $i \geq 0$ where Δ is the time interval between the start of two consecutive pages, independently of i.
- The PER values are uncorrelated, with average $\overline{p}$.

In such a scenario, the average TTR is linearly proportional to the average number $\overline{L}$ of received pages that are needed to retrieve the message.

Fig. 4 shows the average value of the TTR for $T_P = 1$ s, $T = \Delta = 30$ s, $K = 50, 100, 150$, as a function of $\overline{p}$, for both ideal and carousel encoding. As it can be seen, for $\overline{p} = 0.8$, ideal encoding allows a TTR that is four to five times lower than for carousel encoding. For lower values of $\overline{p}$, the gain of ideal encoding reduces. Still, for $\overline{p} = 0.1$, the TTR for ideal encoding is about half the TTR for carousel encoding. With the considered TTR values, for $K = 50$ and $\overline{p} = 0.1$, it is equivalent to save almost half an hour. It is worth noting that the gain is increasing with the message length K.

E. The case of correlated PER values

In general, the diverse values p_i of the PER into block i are *correlated* because the (random) variation of the physical-layer channel into the diverse blocks (page times) are correlated as well.

1) The Gilbert-Elliott channel model: We start by considering a simple test case to check the results of our computation. In particular, we take the Gilbert-Elliott model for the physical-layer channel represented in Fig. 5. The channel is modeled as a Markov chain with two states, a good one and a bad one, with fixed (state-dependent) PER values equal to p_G and p_B, respectively. If P_{GB} and P_{BG} are the good-to-bad and bad-to-good transition probabilities, the stationary probability of being in the bad state[3] is $\pi_B = P_{GB}/(P_{BG} + P_{GB})$.

We consider now the special case of $p_G = 0$ and $p_B = 1$, i.e., the condition in which the channel, after possibly starting from a good state and allowing for the retrieval of some pages, ends up in a final, stead-state bad condition. In such a case, the performance of ideal encoding can be obtained in a relatively easy way.

Proposition 2 *For the Gilbert-Elliott channel of Fig. 5, with $p_G = 0$ and $p_B = 1$,*

$$\overline{L}^{\mathrm{id}} = K + \left(K - 1 + \frac{1}{P_{BG} + P_{GB}} \right) \frac{P_{GB}}{P_{BG}} \tag{18}$$

Proof: [Sketch] Assume that R is the number of bursts the channel experiences in the good state before being able to recover K pages. It can be shown that R is a binomially-distributed rv with parameters $K - 1$ and p_{GB}. The redundancy that is needed in order to collect the K pages is equal to the number of steps spent by the channel in the bad state, which, given the value of R, has a negative binomial distribution either with parameters R and p_{BG}, if the channel starts in the bad state, or $R - 1$ and p_{BG}, if the channel starts in the good state. Such considerations lead in a straightforward way to the average

[3]We will suppose that P_{GB} and P_{BG} are not both zero or both one.

number of received pages given by (18). ■

Fig. 6 shows the average TTR versus P_{BB} for the same reception timing as Fig. 4, i.e., $\tau_i = i\Delta + T_P$, $i \geq 0$, with $T_P = 1$ s, $T = \Delta = 30$ s. In addition, $K = 50$, $P_{GG} = 0.9$, $p_G = 0$ and $p_B = 1$. As a comparison, we also show the average TTR for the uncorrelated scenario with the same average PER, i.e. $\overline{p} = \pi_G$. It can be seen that ideal encoding performs about the same in both scenarios, whilst carousel encoding performs better for the Gilbert-Elliott channel than in the uncorrelated case, especially for large values of P_{BB}.

2) A realistic scenario: Finally, we investigate a more realistic scenario in which the PER values are obtained from the detailed simulation of a physical-layer LMS channel applied to the GALILEO I/NAV message, and we export the resulting statistics in term of PER to our link.-layer model - we call this *abstraction*. In particular, we consider the well-known 2-state semi-Markov model described in [33] for the LMS channel, and create a multisatellite time-varying constellation with the parameters defined in Table 1. Within a 3-hour simulation time, the 6-satellite constellation takes six different elevation-azimuth combinations, each lasting for half an hour.

Number of satellites	6
Environment	Urban
User speed	50 km/h
Sampling frequency	~ 584 Hz
Time-series duration	10800 s

Table 1: Channel parameters.

The PER vector $\mathbf{p}$ is obtained by decoding the received physical-layer codewords of Galileo I/NAV with an optimal Viterbi decoder, for the 64-state, rate-$1/2$ convolutional code. The input length of the physical-layer code is equal to 120 information bits and $T_P = 2$ s. The message size is equal to $K = 144$ pages and

$$\tau_i = \left(\left\lfloor \frac{i}{6} \right\rfloor + 1 \right) \Delta + T_P, i = 0, 1, \ldots$$

where $\Delta = 30$ s, which means that, for a given satellite, pages are equi-spaced and that the different satellites are synchronized. Finally, the reception start time is uniformly distributed over the whole simulation time window (with the PER vector wrapped around whenever there is need for a transmission tail).

Fig. 7 shows the simulation results in terms of average TTR as a function of the so-called line-of-sight (LOS) signal-to-noise ratio C/N_0, in dBHz, i.e., the value of the SNR we would get in the same

propagation conditions but in the absence of the LMS channel attenuation due to shadowing and fading. Carousel encoding results in a TTR that is two to three times the TTR for ideal encoding. We also plotted (dashed curves) the analytical expression of the TTR when the actual, time-varying PER vector is replaced by its corresponding time-average over the entire simulation. While for ideal encoding this results in a negligible error, the difference for carousel encoding is large for low C/N_0 values only. In particular, like for the Gilbert-Elliott channel, approximating the PER vector with its long-term mean results in an overestimate of the actual TTR performance. A heuristic way of explaining the fact that carousel encoding seems to take advantage of the channel memory, at least in the considered scenarios, is the following. Erasures happen in bursts, as well as successful receptions. If in both cases the average burst length is equal to B, we can roughly say that the channel looks like an uncorrelated channel for a message with length K/B. This effective reduction in the carousel length proves beneficial when the average PER is large enough.

IV. Practical LLC Coding Schemes and Their Performance

In this section, we first introduce two particular LLC schemes that are suitable for application to GNSSs: LDPC codes and Fountain codes. Then, we test such schemes through computer simulations in realistic scenarios to assess their relative performance in terms of TTR.

The main properties of the GNSS link layer that impact on the LLC scheme design are the following:

- The message size K (in pages) is typically small-to-medium, ranging from a few tens to a few hundreds. This fact advocates caution when designing an LLC scheme based on the typical design rules, which are often conceived on the contrary for large block lengths and therefore make use of *asymptotic* properties that may not hold here.

- The coding rate can be very low, since the navigation message (e.g., almanac) may not change over several days. Consequently, (think of simple carouseling) the LLC encoding rate may become vanishingly small.

- Reception is asynchronous, i.e., the receiver starts receiving at any time without specific reference to the start of message (carousel).

- Certain types of messages are transmitted by multiple satellites, so that the LLC scheme design can also benefit from concepts that are derived from network coding literature (extension of transmission diversity) [17].

- Unlike many applications, in which the block erasure probability is the performance parameter of interest, the main figure of merit for GNSSs is the TTR, so that the typical design rules for LLC must be revised in order to keep into account this shift in the perspective.

We also summarize the three different coding schemes whose performance will be investigated:

1. Trivial carousel transmission;

2. Cascade (concatenation) of a short-block rate-r_c code with (low-rate) carousel transmission - our reference solution wherein the short-block code is an LDPC;

3. Low-rate LLC code without any carousel transmission - the typical case of Fountain codes.

In particular, we report in the next subsections more details about schemes 2. and 3. above.

A. Low-density parity-check (LDPC) codes

As is known [32], an (N, K), rate $r_c = K/N$ binary LDPC code is characterized by its *sparse* or *low-density* $(N - K) \times N$ binary parity-check matrix $\mathbf{H}$ or, equivalently, by its *Tanner graph*, which is the bipartite graph whose incidence matrix is $\mathbf{H}$. Specifically, the Tanner graph is composed by N variable nodes (VNs), each of which corresponds to a different column of $\mathbf{H}$, and by $N - K$ check nodes (CNs), corresponding to the rows of $\mathbf{H}$. The graph has got an edge connecting the j-th VN and the i-th CN if and only if the (i, j) element of $\mathbf{H}$ is 1.

LDPCs are commonly used as channel codes, and they are being proposed as such for next-generation GNSSs. Here, we slightly bend this usual habit in that we use them at the link layer.

When the LDPC code is used on the erasure channel, the decoder input is a received codeword $\mathbf{y}$ with length N, wherein some of the (coded) symbols are erased, while the non-erased symbols are received correctly. To decode an LDPC code on the erasure channel, we have essentially two options: either message-passing (MP) [26] or maximum-likelihood (ML) decoding [27]. The former works directly on the Tanner graph in an iterative fashion, whereas the latter relies on solving a linear system of equations. The MP is empirical but denoted by a lower complexity compared to the ML decoding which attains optimum performance. An efficient implementation of the ML decoder is the so-called ML-pivoting (ML-P) algorithm described in [27].

In our application of link-layer coding to our case of reference (Galileo I/NAV), the variable node of the Tanner graph are not just connected to received or erased bits; rather, they are connected to received/erased full *pages*. Also, the length of the encoded block N is relatively low with respect of

the usual cases in which LDPCs are used as channel codes. In our case, a suitable decoding option can be the hybrid MP/ML decoder introduced in [28]. The idea is very simple: after receiving the channel output $\mathbf{y}$, low-complexity MP decoding is performed first. If decoding is not successful, which means that some erased VNs were not recovered, an ML decoder is further applied to try and recover such residual erasures. It is easily seen that, while the performance of the hybrid decoder is the same as that of ML decoding, its complexity is in between that of the (simple) MP decoder and that of the (complex) ML-P one. Indeed, if C_h, C_{MP} and C_{ML-P} are the complexities of the hybrid, the MP and the ML-P decoder, respectively, and p_{MP} is the error probability for the MP decoder, we can approximately say that

$$C_h = (1 - p_{MP})C_{MP} + p_{MP}C_{ML-P} \tag{19}$$

The equation stems from the fact that, if MP decoding is successful (with probability $1 - p_{MP}$), the complexity of the hybrid decoder equals that of the MP decoder; on the contrary, if MP decoding fails (with probability p_{MP}), the complexity of the cascade of the MP decoder and the ML-P decoder on the residual erasures is roughly equal to that of the ML-P decoder on the received vector $\mathbf{y}$.

Since our design goal is TTR minimization, an *ad-hoc* design has been devised to obtain LDPC codes with good TTR performance. The design procedure, which is described in detail in the Appendix, consists in standard LDPC code design for erasure probability minimization, cascaded with an optimal code permutation search, which results in close-to-optimal TTR performance, as will be shown in Subsection C..

B. Fountain codes

Fountain (or rateless) codes were originally designed to allow efficient and asynchronous download over broadcast channels [32]. In general, a Fountain encoder is an algorithm that, given a size-K binary information block, produces a potentially endless stream of encoded symbols $c_0, c_1, \ldots$, where each c_i is a linear parity-check bit whose check equation has *randomly chosen* binary coefficients for the information bits, and such coefficients are chosen independently of each other check after check, according to a predetermined probability distribution. The most important classes of Fountain codes are LT codes and Raptor codes, which will be described in more details in the next subsections.

1) LT codes: LT codes [12] are one of the most important families of Fountain codes. For LT codes, the parity checks are derived in the following way:

1. The degree of the linear check (number of binary coefficients equal to 1), ranging from 1 to K, is drawn according to a pre-assigned probability distribution Ω. Call d the resulting degree for a certain parity check.

2. The set of d input (information) symbols that contribute to the check are chosen uniformly at random in the input block.

As is apparent, a given LT encoder is characterized by the two parameters (K, Ω). For a given value of K, a good choice of Ω is the so-called *robust soliton distribution*, defined as in [12]. Using this probability distribution, we are guaranteed that the LT decoder will successfully decode any codeblock with probability $1-\delta$ as soon as $K+2s\log(s/\delta)$ encoded symbols are received, being δ the desired error probability after LT decoding and $s \stackrel{\triangle}{=} (1-\varepsilon)\sqrt{K}\log(K/\delta)$. In the practice, the parity check symbols are not endlessly generated: either a maximum number of check is set by a certain transmission format (based on specifications of the worst reception status of end-users), or a feedback channel is available, so that the user can send back an ACK message at the end of successful decoding. Also, the pattern of random extractions to generate the checks used by the encoder has to be known by the decoder, too, so that they are actually performed according to a pseudo-random number generator whose initial seed is agreed between encoder and decoder.

The LT decoder is based on the same MP algorithm that is used by LDPC decoders, where the Tanner graph (or, equivalently, the parity-check matrix) corresponding to a given set of received coded symbols is built "on the fly" at the receiver. Since the decoder has to know the correspondence between coded symbols and input symbols, some overhead must be allocated to allow the receiver to synchronize with the coded symbol stream (whose blocklength is neither fixed nor known in advance).

2) Raptor codes: Because of their near-optimal performance and relative simplicity, Raptor codes [19] have become the state-of-the-art Fountain coding technique. A Raptor encoder can be described as the serial concatenation of an outer, "classical" (i.e., fixed-rate) FEC encoder and an inner LT encoder as above. The introduction of the precoder allows avoiding the real bottleneck in the performance of an LT code, i.e., the time that is needed to decode a message grows more than linearly with the size of the message itself (this property has to do with the well-known *coupon-collector problem* in probability theory). In a nutshell, after the "easy-to-recover" first message symbols has been retrieved, those who come next become progressively more and more difficult to decode, requiring more and more (independent) check symbols to be received. If we cascade an outer code with the LT code, we do not actually need to LT-decode *all* of the message symbols, and in particular the last, most difficult ones -

even if we consider them erased, the outer decoder will anyway fix them thanks to the (small) added redundancy. Because of the presence of the outer code, the average degree drawn with the relevant optimum distribution is constant with respect to K, unlike (stand-alone) LT codes for which it grows like $\log K$, resulting in a lower encoding complexity per input symbol for Raptor than for LT codes.

Raptor codes are in general not particularly suited to GNSS, as the message blocklength K is small. But, the systematic Raptor code for multimedia broadcast and multicast services (MBMS) over 3G cellular networks [20] was designed to to be simple and to work well for relatively short blocks, ranging between 500 and 8196, so it is worth consideration. In that scheme, precoding is organized in two phases: the first stage makes use of a regular systematic LDPC code that adds S parity check symbols to the original K input symbols. The second stage is somewhat similar to a Hamming code [24] and provides further H symbols. The precoder output is therefore composed by $N = K+S+H$ intermediate symbols, and undergoes LT encoding. The main difference with respect to the typical Raptor code design is in the decoder, which is in principle similar to the ML-P decoder described in the previous subsection. After $L \geq N$ coded symbols are collected from the received signal, the $L \times N$ parity-check matrix (Tanner graph) relating such received symbols to the input word is set up. If such matrix is full rank, it is inverted within the binary field to recover the input word. Otherwise, more coded symbols are to be received, and so on.

From a practical point of view, we can list a couple of disadvantages related with the adoption of Raptor codes as the LLC scheme in GNSS: i) every coded symbol must be complemented by an integer (the encoding symbol ID, ESI) which represents the position within the coded stream. In the MBMS standard [20], 16 bits are used to represent the ESI, thus creating a non-negligible overhead; also, ii) Raptor codes are protected by strong international patents.

C. Numerical results

We present now some performance results of the LLC schemes described in the previous subsections. Our main performance metrics will be the average TTR as a function of the LOS C/N_0 of the LMS channel, as explained in subsection III.E. Our first result is simple: there is very little to gain by the application of LLC to short messages like CED data. We state this without the need to support it with any quantitative data since it is quite intuitive. On the contrary, we found considerable gain for two different types of long messages: the first one is just the current Galileo I/NAV almanac, composed as we already mentioned by $K = 48$ pages, and assuming single-satellite reception. In each subframe of duration $T_S = 30\,\mathrm{s}$, there are two consecutive pages (of duration $T_P = 2\,\mathrm{s}$) carrying the almanac

message, as shown in Fig. 8.

The second, hypothesized, message format is a possible extended almanac made of $K = 192$ pages to be disseminated trough Galileo I/NAV, assuming multi-satellite dissemination from M satellites (as is the case today for the almanac). We suppose that in each 30-s subframe, there is a single transmitted 2-s page, and that all satellites are synchronized, as shown in Fig. 8. Notice that, for carousel-based schemes, we impose suitable offsets on the carousel starts for each satellite in order to optimize TTR.

The set-up of the simulations is the following: the physical-layer code is the Galileo I/NAV convolutional code, with optimal Viterbi decoding. Perfect synchronization and LMS channel estimation is assumed. The received LOS C/N_0 takes also into account an elevation-dependent antenna gain. The different options in terms of LLC are the following:

- Legacy carousel
- Ideal LLC
- LT code with robust soliton distribution and $s \simeq K$;
- MBMS standard Raptor codes;
- Two different rate-$1/2$ LDPC codes of lengths $N = 96$ and $N = 384$, respectively, both designed according to the procedure described in the Appendix, cascaded with carouseling.

In our LDPC-based LLC, the assumption is that the total duration of the carousel is kept unchanged. For instance, with a rate-1/2 LDPC code, we assume to double the duration of a single carousel and to halve the total number of retransmissions. The same assumption holds for rateless LLC schemes, assuming that the number of check symbols is limited.

Fig. 9 shows the average TTR for the different LLC techniques in the case of the almanac message. The LMS time series was obtained by simulating the model of [33], for an elevation of 25°, an urban environment, and a user speed of 5 km/h. The sampling time is about 97 Hz, while the time series duration is equal to 30 hours. The solid lines refer to LLC schemes with ML/hybrid decoding, while the dashed lines refer to LLC schemes with MP decoding.

The Raptor code has a performance very close to ideal (4-5% degradation), and substantially outperforms the carousel scheme for C/N_0 lower than 40 dBHz. On the contrary, the average TTR curve for the LT code incurs a penalty of about 45-50% wrt the optimal performance. The LT code suffers from i) the lack of precoding, and ii) the use of an MP decoder, which is definitely sub-optimal with short block-lengths wrt the ML decoder of the Raptor code. Our novel rate-1/2 LDPC code (with the ML/hybrid

decoder) exhibits a very good performance above 30 dBHz, with a sizable degradation below. The blue dashed curve depicts the performance for the LDPC Code with MP decoding. The corresponding TTR is larger by about 15-25% with respect to the hybrid/ML decoder.

The same kind of performance is shown in Fig. 10 in the case of the Galileo I/NAV *extended* almanac and for the same scenario. The hierarchy between the curves is the same as in the previous case, but the differences are enhanced, because of the larger message length.

In Fig. 11, we show the average TTR for the extended almanac in a 6-satellite scenario, urban environment and a user speed of 50 km/h. The sampling time is about 584 Hz, while the time series duration is equal to 3 hours. Again, the Raptor code is a very good approximation of ideal LLC, while the LT code has a fixed penalty, for all C/N_0 values. Regarding the carousel-based schemes, our novel rate-1/2 LDPC code, when ML decoding is employed, largely outperforms the carousel performance, and has close-to-optimal performance above 27 dBHz.

Finally, in Fig. 12, for the same scenario of Fig. 11, we show the performance of the different LLC schemes when the receiver possesses side information. As mentioned in the Introduction, this side information may be available when the receiver is able to perform A-GPS reception. In particular, we suppose that the receiver already knows half of the message pages, and that the decoder is able to optimally use such knowledge in the decoding process. As we can see from Fig. 12, obviously all LLC schemes benefit from the side information. However, while Ideal LLC has essentially a 50% improvement in the average TTR, the other schemes have a lower gain, since their designs do not take into account such side knowledge. As a consequence, the best LLC schemes (Raptor and LDPC with ML decoding) show an appreciable gap from optimal performance. In particular, the Raptor code requires an average TTR that is about 20% larger than the ideal one, while the LDPC code with ML decoding needs on average a TTR overhead of about 33% with respect to the minimum one. Still, their performance remains quite good, especially if compared with uncoded carousel transmission, whose average TTR is more than four times the ideal one.

V. Summary, Conclusions, and Perspectives

In the paper, we have shown how to effectively apply the well-known notion of link-layer coding to the specific issue of the protection of GNSS navigation messages. Considering as our reference study case the Galileo I/NAV message, we have shown that the use of simple link-layer codes provides considerable improvement (reduction) in the time that the GNSS receiver needs to correctly receive and decode the navigation message (almanac and extended almanac), with respect to the case of current plain carousel

(re-)transmission. We have also shown that the performance of simple and practical LLC schemes, like our novel *ad hoc*-designed page-level LDPC code, is very close to that of an ideal code, i.e., a code that is capable of reconstructing the whole message as soon as any K encoded pages out of the $N > K$ transmitted ones are correctly received. The performance improvement is really considerable in all of those impaired situations (urban canyon, indoor reception, strong shadowing by trees) wherein current decoding times are really prohibitive, especially for long messages such as almanac and extended almanac data.

In addition, the adoption of such technology can have minimum impact or can even be totally transparent to the space segment of current GNSSs - they could be upgraded by a simple redefinition of the navigation message structure that is not affecting any systems in the space segment, since the navigation message is formatted on-ground. Concerning user receivers, new chipsets would need the addition of the LLC decoder to be able to actually exploit the benefits of the new technology. The complexity of the additional LLC decoder is close to irrelevant in the framework of the whole receiver design, but still user equipment would need to be upgraded. A partial exploitation of LLC encoded messages could be obtained with the use of systematic codes, for which part of the codeword is coincident with the uncoded message. In such a case, a receiver without link-layer decoder could retrieve the message, provided that it knows where the systematic pages are located within the transmission frame.

In conclusion, more work will be needed to test and validate different coding schemes in different scenarios, as well as to exploit practical implementations of it. Nonetheless, we believe that future generations of GNSSs (and possibly, modernization of current ones) will greatly benefit from the adoption of this technology.

ACKNOWLEDGMENT

This research was developed within the framework of ESA-funded projects ”ADVISE” and ”GNSS End-to-end Simulator”. The content of the present paper is research reflecting solely the authors’ view and by no means represents the official ESA view or the one from Galileo or EGNOS Project Offices.

APPENDIX

In this appendix, we describe an optimal design procedure of LDPC codes to minimize the message TTR. Let us fix the rate r_c of the LDPC code. Since the LDPC code must be cascaded with carousel transmission, which is generally suboptimal, it turns out that, in order to approach an ideal performance,

we have to minimize the probability of going beyond the first carousel in message decoding , or, equivalently, the block error probability should be minimized. To achieve this goal, different decoding choices as described in Section A. lead to different design rules. In particular:

- If the MP decoder is assumed, the design essentially consists in optimizing the degree distributions of the VNs and CNs according to density evolution. The left (VN) and right (CNs) degree distributions are defined by

$$\begin{aligned} \lambda(x) &= \sum_{i=2}^{d_{v,\max}} \lambda_i x^{i-1} \\ \rho(x) &= \sum_{i=2}^{d_{c,\max}} \rho_i x^{i-1} \end{aligned}$$

  where λ_i (resp. ρ_i) is the fraction of edges connected with VNs (resp. CNs) with degree i. The design method is well established, especially for the erasure channel, and is described in detail in [26, Chapter 3]. After deriving the degree distributions, a parity-check matrix satisfying such distributions must be randomly built. For small codeword lengths, some additional care must be taken, while building the matrix, to avoid short cycles in the graph. A possible algorithm explicitly conceived for this purpose is progressive edge growth (PEG) [29].

- If ML decoding is performed, the design is quite simple. Even a randomly picked parity-check matrix with a not-so-low density of ones performs quite close to the MDS (Singleton) bound [24, p. 33]. However, as pointed out in [27], the complexity of the ML-P decoder for such a randomly picked LDPC code will be typically very high. There is a tight relationship between the average number of pivots columns in ML-P decoding and the performance of the MP decoder. Briefly, an LDPC code with good MP performance is also an LDPC code with low ML-P decoding complexity. Thus, in order to keep low complexity, even in the case of ML decoding, the same design rules of MP-decoded LDPC codes should be applied.

- Finally, if the hybrid decoder is used, from the point-of-view of performance, the same remarks as for the ML decoder can be done. However, regarding the complexity, it is considerably lower for an LDPC code with good MP performance, for two reasons: the first is that the ML-P decoder is more efficient, as pointed out in the previous item, while the second is that the ML-P decoder is called less frequently as the MP decoder is often successful (see (19)).

Summarizing, we come up with the following design rule.

Rule 1: Given the coding rate, the LDPC design should follow classical optimization rules for the graph degree distribution (such as density evolution [26]) and for the parity-check matrix (such as PEG [29]).

After having found an optimized degree distribution and a corresponding parity-check matrix, we have obtained an LDPC code, which with high probability needs just one carousel to be succesfully decode. However, since our ultimate goal is TTR minimization, the LDPC code should be also able to decode after receiving $K + \alpha$ pages, where α ought to be a small positive integer. In particular, after receiving $K + \alpha$ *consecutive* pages, we can view the remaining $N - K - \alpha$ yet to be received as if they have been subject to an *erasure burst*. Thus, to be able to decode, the code must be able to cope with erasure bursts. In [30], it is shown that, under mild conditions, any LDPC code can correct an erasure burst of length at most $N - K$, provided a suitable permutation of VNs is introduced. In the following, we derive equivalent conditions in the slightly different scenario in which end-around bursts, i.e., bursts that start toward the end of the codeword and end in the next carousel retransmission, are kept into account.

Define a length-d *zero run* in a binary vector as a sequence of d consecutive 0's between two 1's, where the run can be cyclic across the vector ends. It is shown in [31] that, if a length-N code has a parity-check equation with a length-d zero run starting at position j, for $j = 1, \ldots, N$, then such code can correct any length-$(d + 1)$ erasure burst. In the following, we give equivalent conditions on the parity-check matrix $\mathbf{H}$ for the code to be able to correct a length-$(N - K)$ erasure burst. First, we write $\mathbf{H}$ in the following form:

$$\mathbf{H} = \left[\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_m, \mathbf{H}'_q\right] \tag{20}$$

where $m = \lfloor 1/(1 - r_c) \rfloor$, matrices $\mathbf{H}_1, \ldots, \mathbf{H}_m$ are $(N - K) \times (N - K)$ square matrices, while $\mathbf{H}'_q$ is a $(N - K) \times q$ tall matrix ($0 \leq q < N - K$). Notice that, for $r_c = 1 - 1/m$, $q = 0$ and the matrix $\mathbf{H}'_q$ does not contain any column. For $q > 0$, define also $\mathbf{H}_1 = [\mathbf{H}_1^L, \mathbf{H}_1^R]$, where $\mathbf{H}_1^L$ contains the first $N - K - q$ columns of $\mathbf{H}_1$, and $\mathbf{H}_1^R$ the remaining q.

A full-rank square binary matrix $\mathbf{A}$ is said to be LU-decomposable if it can be written as

$$\mathbf{A} = \mathbf{L}\mathbf{U} \tag{21}$$

where $\mathbf{L}$ is a lower triangular matrix with ones on the main diagonal and $\mathbf{U}$ is an upper triangular matrix with ones on the main diagonal. Notice that it is easy to verify whether a given $r \times r$ matrix is LU-decomposable by looking at all its leading principal minors, i.e., the determinants of the $p \times p$

upper-left corners of the matrix, $p = 1, \ldots, r$. The following theorem is the basis of our approach to TTR minimization:

Theorem V..1 *Consider a (N, K) LDPC code with a $(N - K) \times N$ parity-check matrix written as in* (20)*. Define a new parity-check matrix given by*

$$\widetilde{\mathbf{H}} = \left[\mathbf{H}_1\mathbf{\Pi}_1, \mathbf{H}_2\mathbf{\Pi}_2, \ldots, \mathbf{H}_m\mathbf{\Pi}_m, \mathbf{H}'_q\mathbf{\Pi}'_q\right] \tag{22}$$

where we have applied a column permutation to each submatrix separately ($\mathbf{\Pi}_i$ is the permutation matrix associated with the column permutation on the i-th submatrix). When $q = 0$, the LDPC code with parity-check matrix $\widetilde{\mathbf{H}}$ can correct a length-$(N-K)$ erasure burst if the following conditions hold:

- $\mathbf{H}_1, \ldots, \mathbf{H}_m$ *are all full rank, and*
- *matrices $\mathbf{M}_i$, $i = 1, \ldots, m$, where $\mathbf{M}_i = \mathbf{\Pi}_i{}^{\mathsf{T}}\mathbf{H}_i^{-1}\mathbf{H}_{i+1}\mathbf{\Pi}_{i+1}$, $i = 1, \ldots, m-1$ and $\mathbf{M}_m = \mathbf{\Pi}_m{}^{\mathsf{T}}\mathbf{H}_m^{-1}\mathbf{H}_1\mathbf{\Pi}_1$, are all LU-decomposable.*

When, on the contrary, $q > 0$, the LDPC code with parity-check matrix $\widetilde{\mathbf{H}}$ can correct a length-$(N - K)$ erasure burst if the following conditions hold:

- $\mathbf{H}_1, \ldots, \mathbf{H}_m$*, are all full rank,*
- *the matrix $\mathbf{H}_{qL} = [\mathbf{H}'_q, \mathbf{H}_1^L]$ is full rank,*
- $\mathbf{\Pi}_1$ *does not mix the columns of $\mathbf{H}_1^L$ and $\mathbf{H}_1^R$, i.e.,*

$$\mathbf{\Pi}_1 = \begin{bmatrix} \mathbf{\Pi}_1^L & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_1^R \end{bmatrix}$$

  (where $\mathbf{\Pi}_1^L$ and $\mathbf{\Pi}_1^R$ are permutation matrices with sizes $N - K - q$ and q, respectively),

- *matrices $\mathbf{M}_i$, $i = 1, \ldots, m$, where $\mathbf{M}_i = \mathbf{\Pi}_i{}^{\mathsf{T}}\mathbf{H}_i^{-1}\mathbf{H}_{i+1}\mathbf{\Pi}_{i+1}$, $i = 1, \ldots, m-1$,*

$$\mathbf{M}_m = \mathbf{\Pi}_m{}^{\mathsf{T}}\mathbf{H}_m^{-1}\mathbf{H}_{qL}\begin{bmatrix} \mathbf{\Pi}'_q & \mathbf{0} \\ \mathbf{0} & \mathbf{\Pi}_1^L \end{bmatrix}$$

  and

$$\mathbf{M}_{m+1} = \mathbf{\Pi}'_q{}^{\mathsf{T}}[\mathbf{I}_q|\mathbf{0}]\mathbf{H}_{qL}^{-1}\mathbf{H}_1^R\mathbf{\Pi}_1^R$$

  are all LU-decomposable.

Proof: Consider a length-$(N-K)$ burst starting within matrix $\mathbf{H}_i$, for some $i \in \{1,\ldots,m-1\}$. Without loss of generality, let us suppose $i=1$. If $\mathbf{M}_1$ is LU-decomposable, then we know that

$$\mathbf{\Pi}_1{}^{\mathsf{T}}\mathbf{H}_1^{-1}\mathbf{H}_2\mathbf{\Pi}_2 = \mathbf{L}_1\mathbf{U}_1 \tag{23}$$

Thus, another parity-check matrix for the same code is given by

$$\widetilde{\mathbf{H}}_1 = \mathbf{L}_1^{-1}\mathbf{\Pi}_1{}^{\mathsf{T}}\mathbf{H}_1^{-1}\mathbf{H} = [\mathbf{L}_1^{-1}, \mathbf{U}_1, \mathbf{L}_1^{-1}\mathbf{\Pi}_1{}^{\mathsf{T}}\mathbf{H}_1^{-1}\mathbf{H}_3\mathbf{\Pi}_3, \ldots, \mathbf{L}_1^{-1}\mathbf{\Pi}_1{}^{\mathsf{T}}\mathbf{H}_1^{-1}\mathbf{H}'_q\mathbf{\Pi}'_q] \tag{24}$$

where, since $\mathbf{L}_1$ is lower triangular and $\mathbf{U}_1$ is upper triangular, there is a length-$(N-K)$ zero-run starting at every position between 1 and $(N-K)$. Thus, the erasure burst is correctable.

The condition on $\mathbf{M}_m$ (and on $\mathbf{M}_{m+1}$ for $q>0$) guarantee the correctability of end-around bursts.

■

In order to enforce joint LU-decomposability of matrices $\mathbf{M}_1,\ldots,\mathbf{M}_{m+1}$, we have implemented a tree-search algorithm in which a node of depth p, $p=1,\ldots,N-K$ represents a possible joint choice of the p-th columns of matrices $\mathbf{\Pi}_1,\ldots,\mathbf{\Pi}_m,\mathbf{\Pi}'_q$. The algorithm adopts a depth-first search of the tree until it reaches a depth-$(N-K)$ leaf. Although we do not have a proof that it is always possible to find a suitable set of permutations for a given matrix $\mathbf{H}$, this has been the case without much computational effort for all the practical cases we have tested. In conclusion, we come up with Design Rule 2.

Rule 2: Given the parity-check matrix $\mathbf{H}$ obtained as per Rule 1, if it does not satisfy the conditions of Theorem V..1 on matrices $\mathbf{H}_1,\ldots,\mathbf{H}_m,\mathbf{H}'_q$, modify it (by column permutation or entry swapping, without modifying the degree distribution) until they are satisfied. Then, derive the column permutation that guarantees the LU-decomposability of matrices $\mathbf{M}_1,\ldots,\mathbf{M}_m$ (and of $\mathbf{M}_{m+1}$ for $q>0$).

REFERENCES

[1] Wortmann, F., and Flüchter, K., "Internet of Things - Technology and Value Added," *Business & Information Systems Engineering*, Vol. 57, 2015, pp. 221–224.

[2] Lin, X., Bergman, J., Gunnarson, F., Liberg, O., Razavi, S. M., Razaghi, S. R., Rydén, H., and Sui, Y., "Positioning for the Internet of Things: A 3GPP Perspective," *arXiv:1705.04269*, 2017.

[3] Yick, J., Mukherjee, B., and Ghosal, D., "Wireless sensor network survey," *Computer Networks*, Vol. 52, 2008, pp. 2292–2330.

[4] Xiong, Z., *Hybrid and Cooperative Positioning Solutions for Wireless Networks*, PhD Dissertation, Politecnico di Torino (Italy), 2014.

[5] Vallina-Rodriguez, N., Crowcroft, J., Finamore, A., Grunenberger, Y., and Papagiannaki, K., "When assistance becomes dependence: characterizing the costs and inefficiencies of A-GPS," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 17, 2013, pp. 3–14.

[6] Gómez-Casco, D., López-Salcedo, J. A., and Seco-Granados, G., "Generalized integration techniques for high-sensitivity GNSS receivers affected by oscillator phase noise," *2016 IEEE Statistical Signal Processing Workshop (SSP)*, Palma de Mallorca, 2016, pp. 1-5.

[7] https://en.wikipedia.org/wiki/Time_to_first_fix.

[8] Zhang, W., and Gao, Y., "New GNSS Navigation Messages for Inherent Fast TTFF and High Sensitivity," *Proc. of The ION 2015 Pacific PNT Meeting*, Honolulu, Apr. 2015, pp. 131–141.

[9] "Some GNSS / GPS stuff that are good to know for IoT", *available at* https://www.disk91.com/2015/technology/internet-of-things-technology/some-gnss-gps-stuff-that-are-good-to-know-for-iot.

[10] ETSI, "Digital Video Broadcasting (DVB); DVB-SH Implementation Guidelines," *ETSI TS 102 584 v1.1.1*, 2008.

[11] Kaplan, E. D., and Hegarty C. J. (Eds.), "Understanding GPS - Principles and Applications, II Ed.," *Artech House*, Boston, 2006.

[12] Luby, M., "LT codes," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, 2002, pp. 271–280.

[13] European Union, "Signal-In-Space Interface Control Document," Issue 1.3, December 2016.

[14] Bolea Alamanac, A., Burzigotti, P., De Gaudenzi, R., Liva, G., Pham, H. N., and Scalise, S., "In-depth analysis of the satellite component of DVB-SH: Scenarios, system dimensioning, simulations and field trial results", *Int. J. Satell. Commun. Network.*, Vol. 27, 2009, pp. 215-–240.

[15] Balazs, M., Liva, G., Parraga Niebla, C., Riera Diaz, N., Scalise, S., Kim, P. S., Chang, D.I., and Lee, H. J., "Link layer coding for dvb-s2 interactive satellite services to trains," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, May 2008, pp. 2922–2926.

[16] Cataldi, P., Gerla, M., and Zampognaro, F., "Rateless Codes for File Transfer over DVB-S," *in Proc. of First International Conference on Advances in Satellite and Space Communications*, Colmar, 2009, pp. 7–12.

[17] Alegre-Godoy, R., and Stojkovic, I., "Improving the availability of the sar/galileo return link service via network coding," in *2014 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, Dec. 2014, pp. 1–6.

[18] https://en.wikipedia.org/wiki/Assisted_GPS.

[19] Shokrollahi, A., "Raptor codes," *IEEE Transactions on Information Theory*, Vol. 52, No. 6, June 2006, pp. 2551–2567.

[20] 3GPP, "Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Services (MBMS); Protocols and Codecs (Release 6)," *3rd Generation Partnership Project (3GPP), tech. rep. 3gpp ts 26.346 v6.3.0 edition*, 2005.

[21] ETSI DVB TM-CBMS1167, "IP Datacast over DVB-H: Content Delivery Protocols," *ETSI, draft Technical Specification*, Sept. 2005.

[22] Tian, S., Li, Y., Shirvanimoghaddam, M., and Vucetic, B., "A Physical-Layer Rateless Code for Wireless Channels," *IEEE Transactions on Communications*, Vol. 61, No. 6, June 2013, pp. 2117–2127.

[23] Rajanna, A., and Haenggi, M., "Enhanced Cellular Coverage and Throughput using Rateless Codes," *IEEE Transactions on Communications*, Vol. 65, No. 5, 2017, pp. 1899–1912.

[24] MacWilliams, F. J., and Sloane, N. J. A., "The theory of error-correcting codes," *North-Holland*, 1977.

[25] Sandifer, C. E., "How Euler did it," *The Mathematical Association of America*, 2007.

[26] Richardson, T., and Urbanke, R., "Modern Coding Theory," *Cambridge University Press*, 2008.

[27] Paolini, E., Liva, G., Matuz, B., and Chiani, M., "Maximum Likelihood Erasure Decoding of LDPC Codes: Pivoting Algorithms and Code Design," *IEEE Transactions on Communications*, Vol. 60, No. 11, Nov. 2012, pp. 3209–3220.

[28] Paolini, E., Liva, G., Matuz, B., and Chiani, M., "Generalized IRA erasure correcting codes for hybrid iterative/maximum likelihood decoding," *IEEE Communications Letters*, Vol. 12, No. 6, June 2008, pp. 450–452.

[29] Hu, X.-Y., Eleftheriou, E., and Arnold, D., "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Trans. Inf. Theory*, Vol. 51, No. 1, Jan. 2005, pp. 386—398.

[30] Fossorier, M., "Universal Burst Error Correction," *IEEE International Symposium on Information Theory (ISIT)*, Seattle, USA, 2006, pp. 1969–1973.

[31] Tai, Y. Y., Zeng, L., Lan, L., Song, S., and Lin, S., "Algebraic Construction of Quasi-Cyclic LDPC Codes - Part II: For AWGN and Binary Random and Burst Erasure Channels," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, LNCS, Vol. 3857, Feb. 2006.

[32] MacKay, D. J. C., "Information Theory, Inference and Learning Algorithms," *Cambridge University Press,* 2003.

[33] Arndt, D., Ihlow, A., Heyn, T., Heuberger, A., Prieto-Cerdeira, R., and Eberlein, E., "State Modelling of the Land Mobile Propagation Channel for Dual-Satellite Systems," *EURASIP Journal on Wireless Communications and Networking 2012*, 2012:228.

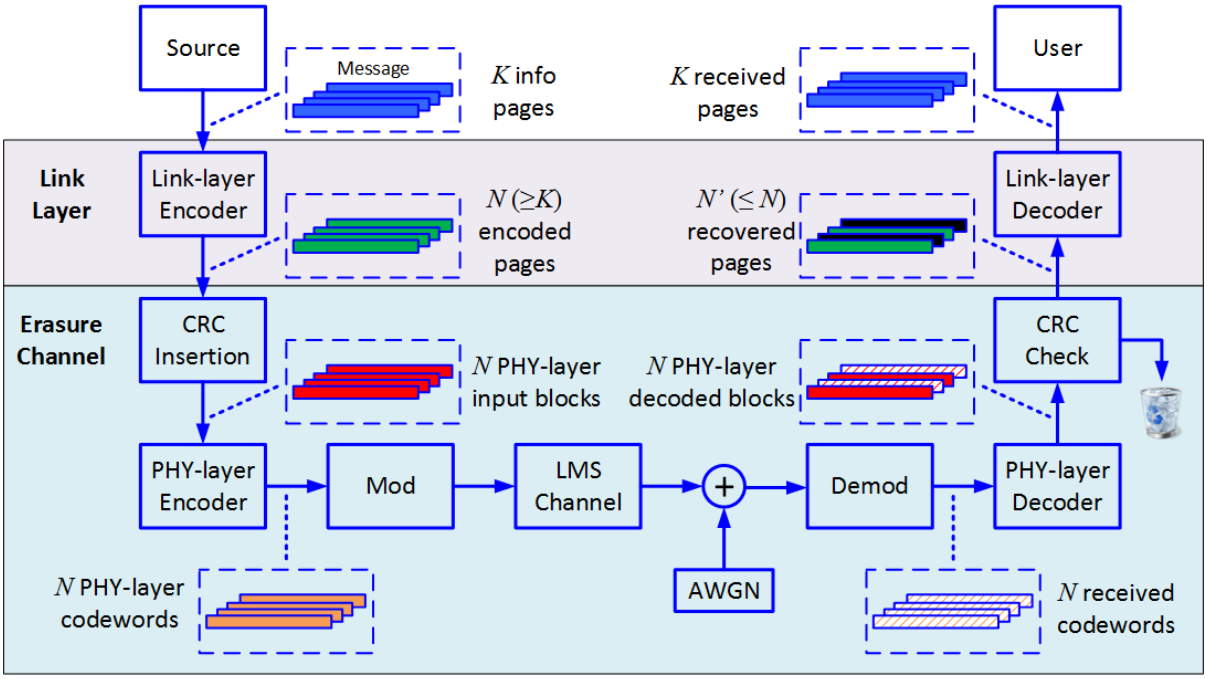

Figure 1: Link and physical layers equivalent system model.

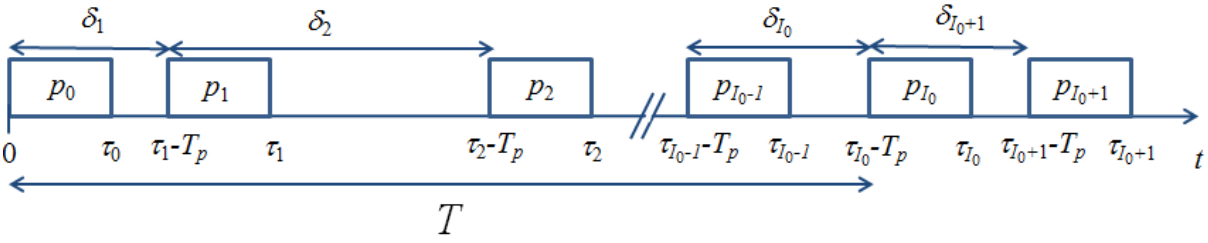

Figure 2: Block reception timing. For block i, τ_i is the reception end time, while p_i is the corresponding PER. Moreover, it is shown the duration-T window of possible reception start time values. I_0 is the repetition period of message pages.

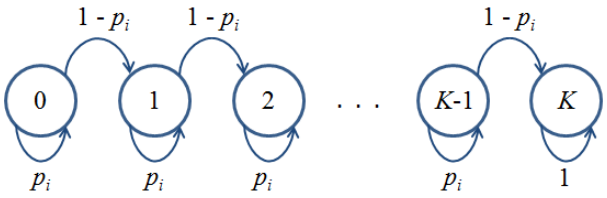

Figure 3: Markov chain representation of reception for ideal encoding.

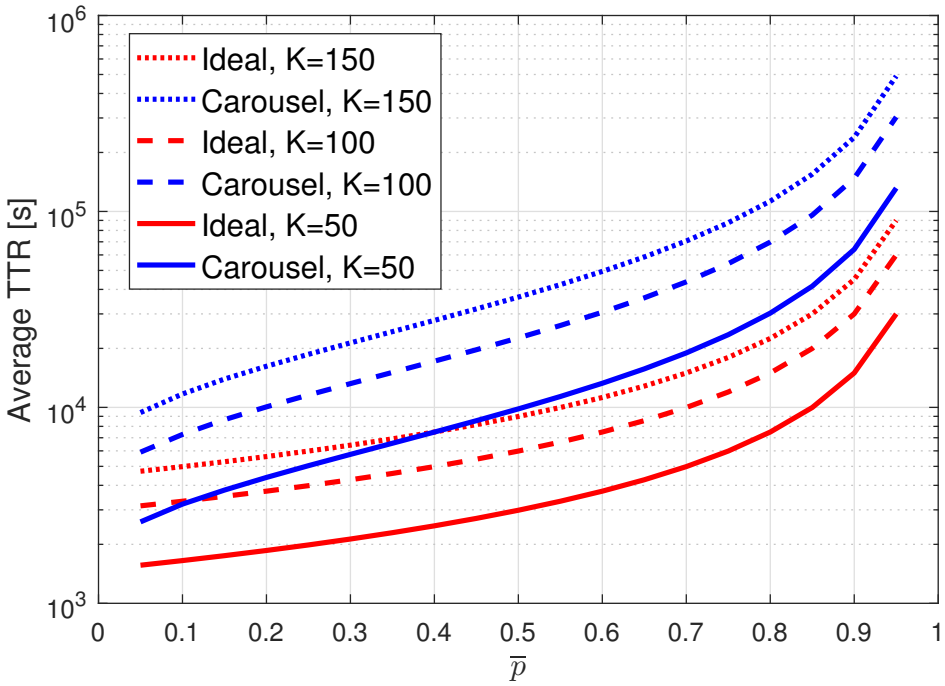

Figure 4: Average TTR of ideal (red solid curves) and carousel (blue dashed curves) encoding. Uncorrelated PER values.

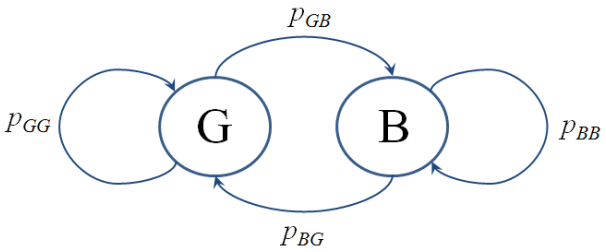

Figure 5: Gilbert-Elliott channel.

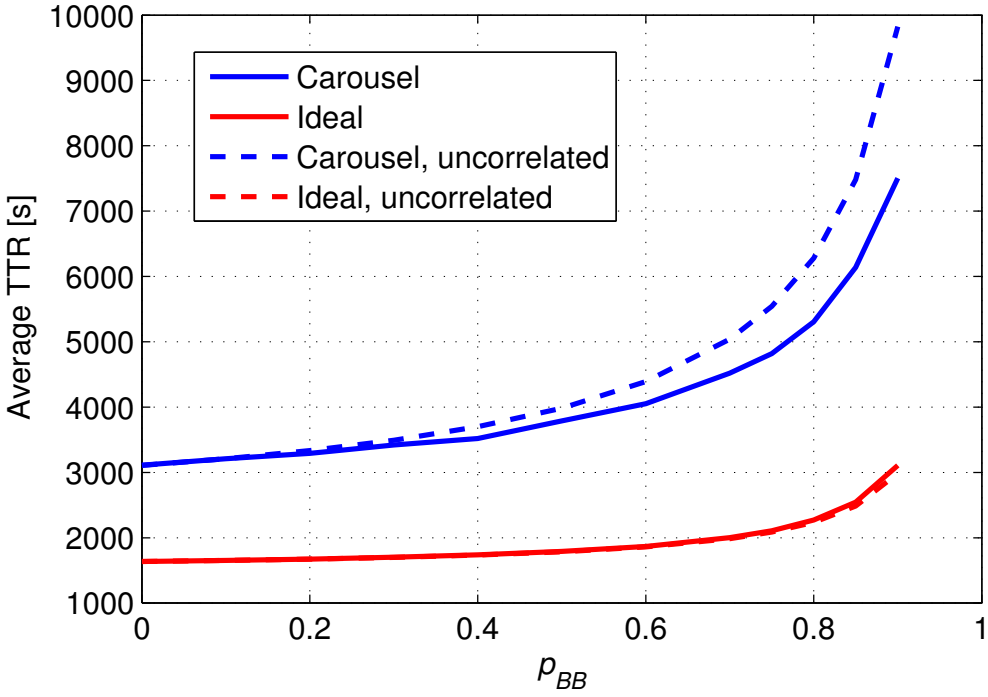

Figure 6: Average TTR of ideal (red curves) and carousel (blue curves) encoding. Gilbert-Elliott channel model.

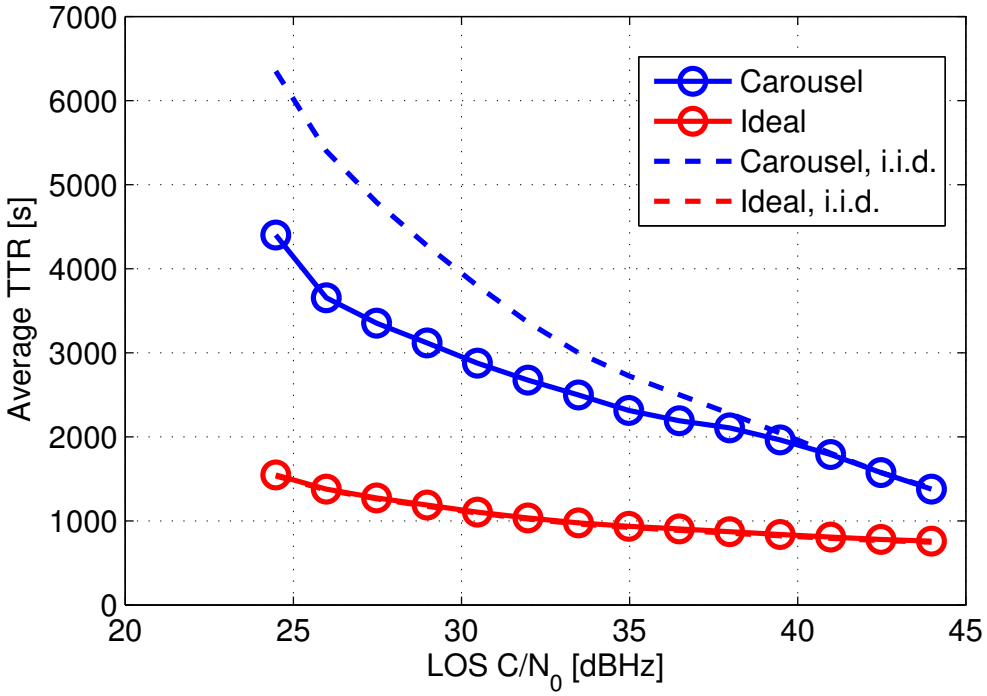

Figure 7: Average TTR of ideal (red) and carousel (blue) encoding for $K = 144$ in the scenario described in Table 1. Solid lines: Correlated PER values. Dashed lines: uncorrelated PER values with same mean.

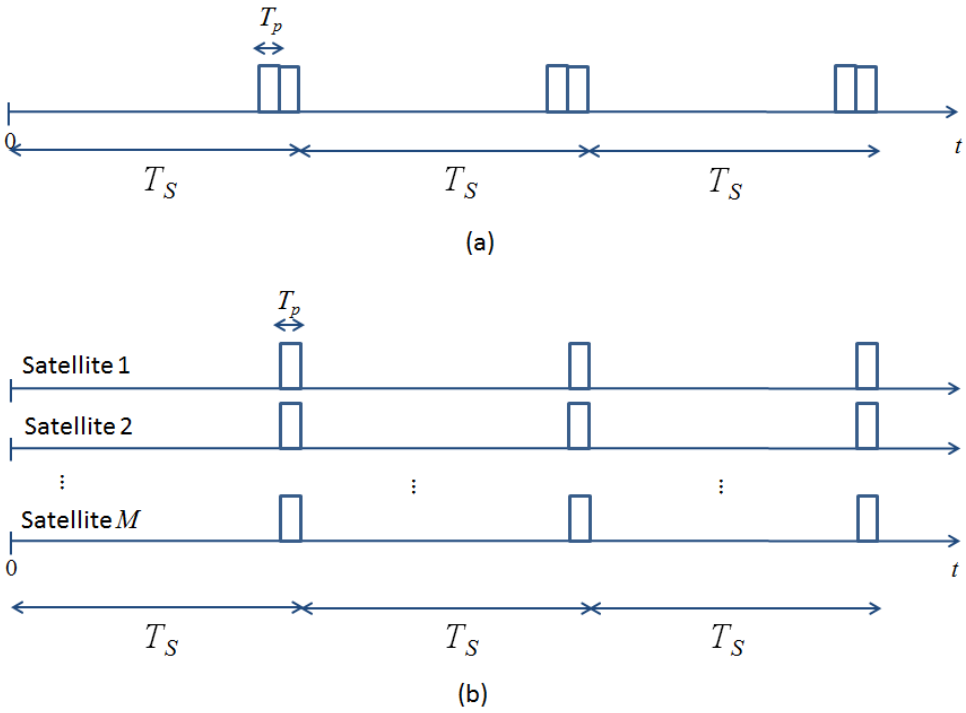

Figure 8: (a) Timing for almanac message. (b) Timing for extended almanac message.

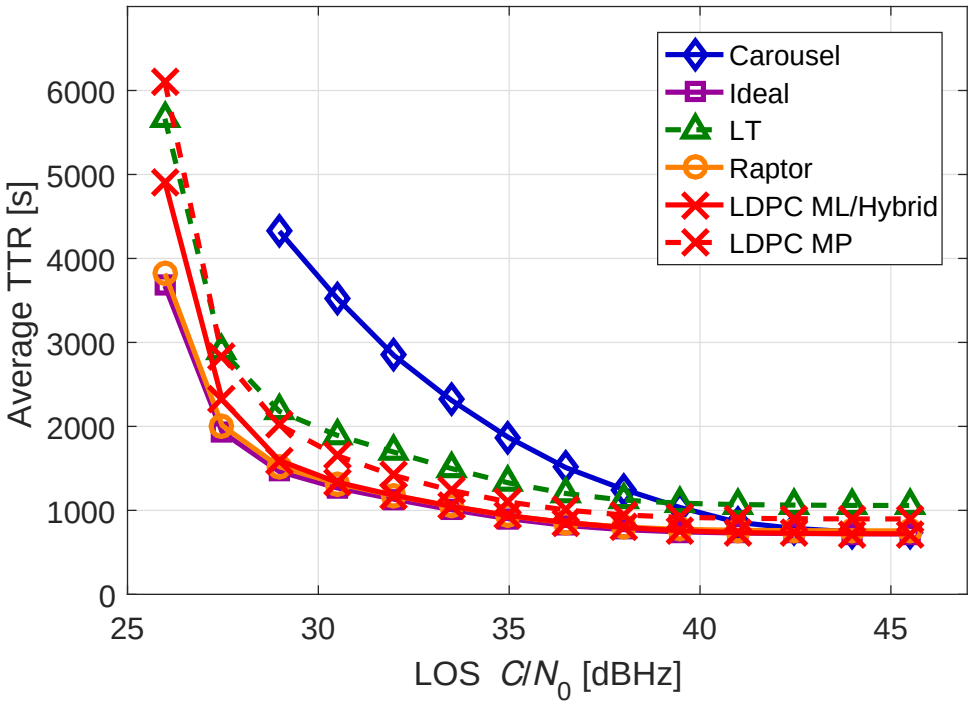

Figure 9: Performance of LLC schemes with the Galileo I/NAV almanac message. Single-satellite scenario.

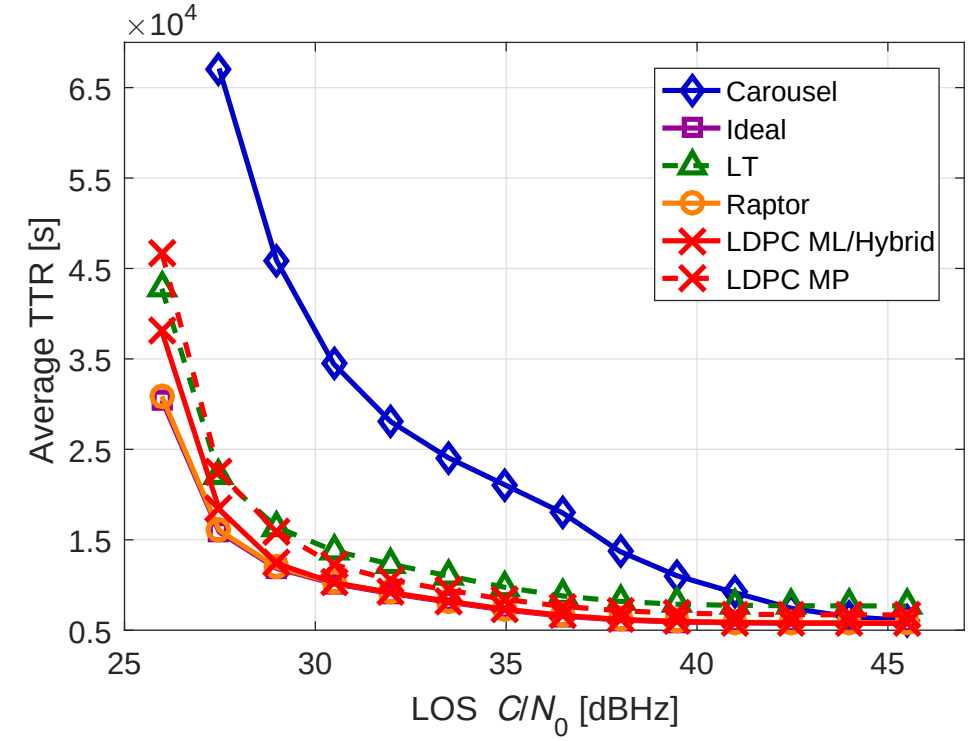

Figure 10: Performance of LLC schemes with the extended almanac message. Single-satellite scenario.

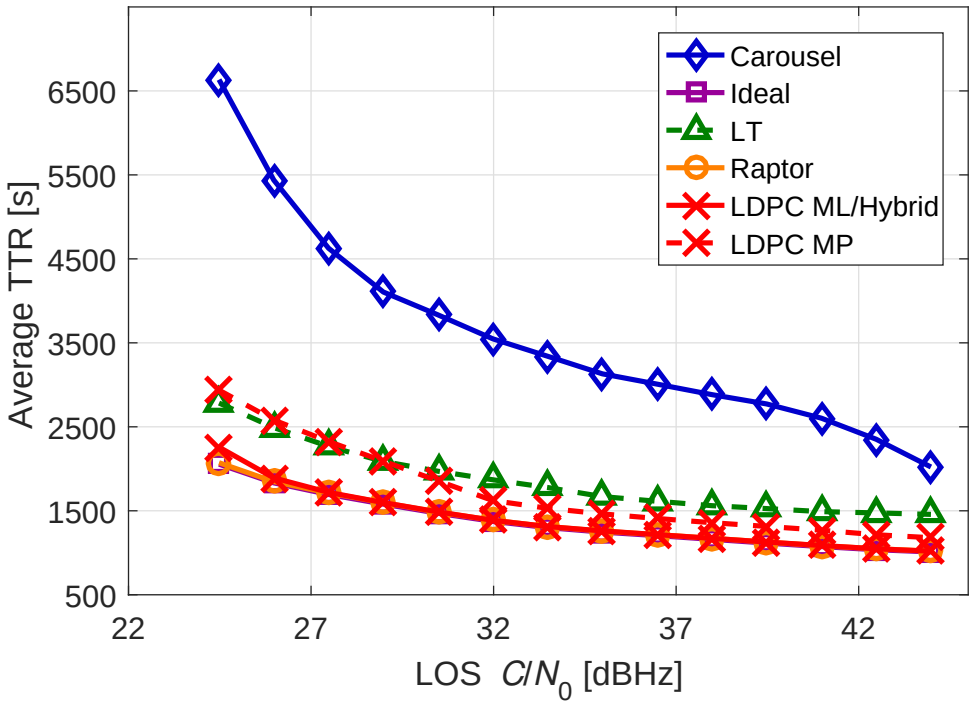

Figure 11: Performance of LLC schemes with the extended almanac message. Multisatellite scenario.

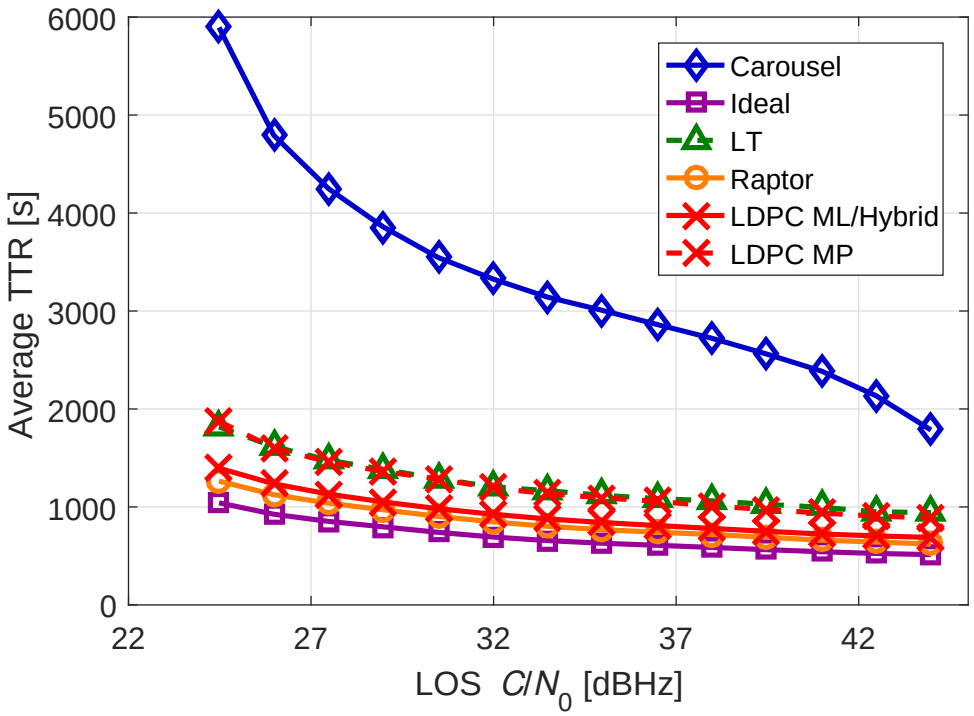

Figure 12: Performance of LLC schemes with the extended almanac message. Multisatellite scenario. The receiver already knows half of the message.