A wake-up strategy enabling GNSS-free NB-IoT links to sparse LEO satellite constellations

Zheng Zhou, Nicola Accettura, Pascal Berthou

A wake-up strategy enabling GNSS-free NB-IoT links to sparse LEO satellite constellations

Zheng Zhou, Nicola Accettura, Pascal Berthou

***Abstract*—The latest release by the 3rd Generation Partnership Project (3GPP) defines how a non-terrestrial NB-IoT link may be set up between User Equipments (UE) on the ground and Low Earth Orbit (LEO) satellites equipped with Evolved Nodes B (eNB). However, a strong assumption is undertaken. Each UE must have Global Navigation Satellite Systems (GNSS) capabilities to properly pre-compensate the Doppler frequency shift and the propagation delay according to the time-varying relative motion of satellites. Additionally, although Release 18 accounts for discontinuous coverage by LEO satellites, the management of next passes over any spot on the Earth is undefined, thus affecting the system scalability. Remarkably, this contribution enables GNSS-free NB-IoT Direct-to-Satellite communications with sparse LEO satellite constellations. To do that, the UE periodically wakes up until it detects a satellite pass in its range. By listening to several NB-IoT beacons, the estimated Doppler curve is used to pre-compensate ongoing communications in frequency and time. Furthermore, the UE uses the standard information sent from the eNB, together with its own estimated location, to guess the next satellite pass without using GNSS. Simulation results reveal that the introduced wake-up strategy allows GNSS-free UEs to save more energy than if equipped with the most power-efficient GNSS chipsets surveyed in 3GPP specifications, promoting the broader deployment of IoT devices in remote and underserved areas.**

***Index Terms*—NB-IoT, LEO constellations, GNSS-free, Wake-up strategy**

I. Introduction

Over the last years, with the scale of the Internet of Things (IoT) expanding and the consequent sprout of a bigger and bigger variety of IoT applications [1], Low Power Wide Area Networks (LPWANs) have gained the interest of researchers and practitioners for their easy and low-cost deployments based on long-distance wireless communications [2]. Several types of LPWAN technologies and protocols have been proposed, each meeting specific requirements [3]. Among them, the Narrowband IoT (NB-IoT) technology is gaining widespread attention because its synchronization and resource allocation strategy make it exceptionally effective in applications requiring high reliability and connection quality [4]. Designed by the 3rd Generation Partnership Project (3GPP) since Release 13, NB-IoT works over the licensed spectrum. It inherits and simplifies most of the features of Long Term Evolution (LTE), thus being compatible with existing infrastructure and significantly reducing deployment complexity [5]. Despite the convenience of long-distance transmission for gateway deployment, there may be significant difficulties in deploying gateways in certain areas, such as mountainous and marine areas. Moreover, in natural disasters, wars, and other critical situations, the gateway on the ground or the connection between the gateway and the server can easily be disrupted. Thereby, backhauling IoT devices with satellite networks has emerged as a significant advancement towards the objective of global connectivity [6]. Remarkably, the Direct-to-Satellite (DtS) architecture [7] represents the most challenging network scenario since battery-operated LPWAN devices on the ground are meant to directly communicate with satellites in their visibility. Compared to the Medium Earth Orbit (MEO) or Geostationary Earth Orbit (GEO) satellites, Low Earth Orbit (LEO) satellites are particularly advantageous for DtS IoT communications because of their closer distance to the Earth's surface. Specifically, closer distances translate into lower propagation delay, a desired feature for real-time data transmission applications. Furthermore, the implied higher strength of received signals permits more efficient communications with very low-power devices [8]. More recently, the research on CubeSat LEO satellites has greatly reduced their costs [9], [10].

However, the high maneuverability of LEO satellites introduces significant variability in both the Doppler effect and the propagation delay, which presents a major challenge for maintaining NB-IoT networks synchronized [11]. In the scenarios proposed by 3GPP [12], low-power User Equipments (UE) on the ground must be capable of determining their own position on the Earth through Global Navigation Satellite Systems (GNSS) [13]. Combined with Two-Line Element (TLE) data, ground devices can calculate their relative position to the satellite, thereby pre-compensating for the Doppler effect and varying propagation delay, thus being able to maintain synchronization during a satellite pass [14]. Moreover, while in traditional ground NB-IoT networks, UEs wake up from the sleep state only when they need to transmit some data, in ground-to-satellite communications, sparse LEO satellites ensure discontinuous coverage [15], and UEs may wake up without any satellite in their reach [16]. To cope with this issue, the most straightforward solution is to achieve global coverage through dense satellite constellations. Even though the price of LEO satellites is getting lower and lower, to achieve global coverage and maintain continuous communications between ground devices and satellites, at least dozens to hundreds of satellites are needed [17], [18]. Herein, 3GPP has included the possibility for UEs to use GNSS together with broadcast satellite information to make communication run under discontinuous coverage. The standard assumes that

Zheng Zhou, Nicola Accettura, and Pascal Berthou are with LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France (e-mail: {firstname.lastname}@laas.fr).

Corresponding author: Zheng Zhou (e-mail: zheng.zhou@laas.fr).

during a satellite pass, a UE is informed by an eNB-equipped satellite passing through its range about next satellite passes. The way how a satellite can produce such information is not specified. For example, this could be done by letting each satellite gather such data from a central constellation supervisor. In any case, the knowledge of a limited number of next satellite passes does not ease the scalability of the system.

Herein, the core idea of this paper is to picture and analyze a new wake-up strategy that allows a GNSS-free UE to set up NB-IoT communications with sparse constellations of LEO satellites of any size, even the most challenging scenario with a single satellite available. The study focuses on the communication between NB-IoT UEs and LEO satellites. It does not fit direct-to-cell architectures, since they rely on satellite-based pre-compensation to handle the Doppler shift variation. Even though such an approach would make ground devices able to seamlessly work within either ground networks or direct-to-cell architectures without any modification, the system complexity is moved on the satellite, that must be equipped with directional antennas and be able to adapt their beam to serve a specific spot on the Earth. All the more reason, the approach hereafter presented does not fit direct-to-cell handheld phones, since this would mean that a large number of satellites would be needed to achieve a continuous coverage. Clearly, the strategy can be applied to scenarios with predictable orbits, such as LEO or MEO satellite constellations. Yet, in the case of MEO satellites the increased communication distance would result in a higher latency and a greater power consumption, thus making such a configuration feasible only with LEO satellites. Finally, it is worth mentioning that the proposed strategy could not work with High-altitude platform station (HAPS), since they are featured by unpredictable flight paths that make unfeasible the prediction of wake-up times. This strategy retains its energy efficiency advantage even in the most challenging scenarios, such as when only a single satellite is available. In the absence of its own location and ephemeris information, the UE uses an intermittent wake-up strategy to attempt to search for the satellite's downlink synchronization signal. Then, instead of receiving GNSS signals before each communication, multiple downlink signals should be received to synchronize UE and satellite.

The strategy uses the synchronization method in the previous work [19], which includes an analysis of measurement errors to demonstrate the method's reliability. The previous work explored the integration of NB-IoT with satellite communication, focusing on the synchronization challenges posed by the high-speed motion of LEO satellites and the resulting Doppler effect. However, this analysis assumes that a satellite is always within range. If global coverage is unavailable, an additional challenge is ensuring that ground devices wake up when a satellite passes by and remain in sleep mode when no satellite is within range. This paper examines the critical aspect of the wake-up strategy for UEs in scenarios with sparse LEO satellite constellations to further address another issue associated with providing NB-IoT coverage through LEO satellites. Notably, this strategy achieves these goals using only downlink messages, avoiding additional uplink transmissions that could affect network throughput and scalability. Meanwhile, each time the satellite passes by, UE will estimate its position based on these signals. After multiple estimations, the UE will determine its approximate position, and finally, it can perform regular communication as needed. After simulation experiments, it has been verified that the strategy proposed in this paper can significantly reduce device energy consumption in the long term compared to using GNSS without particular changes to the hardware. In short, it is optimized from the perspective of device complexity and energy consumption.

The rest of the paper is organized as follows. Sec. II briefly reviews related works, while Sec. III pictures the core idea of this paper. Then, Sec. IV presents the results of a simulation-based analysis. Finally, Sec. V concludes and envisages future works.

II. Related works

Integrating LTE technologies with LEO satellites has recently become a popular research direction. [14] explores satellite-enabled LTE systems within LEO constellations. More specifically, [20] analyzes the performance of NB-IoT uplink in low Earth orbit non-terrestrial networks. Instead, [21] discusses the design and evaluation of service-oriented solutions for NB-IoT over LEO satellite systems. Besides, using a stochastic geometry approach, [22] proposes an efficient coverage enhancement mechanism for NB-IoT via LEO satellite networks. Lastly, [23] presents a new approach to enhance NB-IoT MAC procedures in satellite networks by integrating coverage enhancement levels and a smart backoff mechanism. Also, 3GPP standards [12] bases the correct functioning of NB-IoT on the availability of GNSS on UEs. Herein, the specification recognizes that GNSS is energy expensive thus the evaluation of energy consumption related to some implementations is also detailed. Calculating and pre-compensating frequency shifts and delays through GNSS and TLE data is undoubtedly an accurate solution. Still, as this paper will show, GNSS-enabled devices are not the best choice for IoT applications. From a practical point of view, GNSS draws more current than needed, adds complexity to the circuit design, and eventually increases the cost of low-power devices. Furthermore, from a technical perspective, linking NB-IoT for LEO DtS communications to the availability of GNSS makes such an architecture neither resilient to outages nor viable in GNSS-denied environments.

As a matter of fact, the need for GNSS-free DtS solutions has been recognized in the scientific literature. In [24], the authors proposed a method to reduce the impact of the Doppler effect from the satellite's perspective without using GNSS. This approach uses multi-beam satellites to divide the ground into smaller regions, providing different channels for each area. By doing so, ground devices can communicate with the satellite using Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) methods. However, allocating additional frequency and time domain resources to different regions requires additional frequency and time domain resources. Moreover, implementing multi-beam technology significantly increases the complexity of

satellite design. Smaller satellites, like CubeSat, typically cannot support multi-beam technology due to their limited size, power, and payload capacity.

With a different GNSS-free approach, the circuitry of eNB remains unchanged, while ground UE devices must be able to measure Doppler shifts. In [25], a frequency offset independent timing synchronization method is proposed to measure the significant Doppler shift accurately. However, since the frequency shift and delay constantly change, simply measuring the Doppler frequency shift is insufficient to complete synchronization. [26] proposed a tracking method that continues after initial synchronization. This approach allows ground devices to continuously track downlink signals for real-time updates of frequency shift changes. However, considering the low energy consumption goal of IoT devices, such a high processing capacity is not suitable. In [27], the author first proposed a method to draw the Doppler curve based on the measured frequency shift of only two downlink signals to achieve pre-compensation without considering the measurement errors.

On such a research landscape, the least squares method was proposed [19] as a means to achieve and maintain synchronization for a sufficiently long time to permit NB-IoT uplink data transmissions. This was done by estimating the Doppler curve based on the measurements related to several NB-IoT beacon signals, i.e., the so-called Narrowband Primary Synchronization Signals (NPSS). Different detection strategies were also proposed to deal with measurement errors of varying sizes and application requirements. With the same approach in mind, the research developments introduced by this contribution represent an attempt to set up a GNSS-free scheme that makes UEs capable of (i) estimating the trajectory of LEO satellites, (ii) their own location, and, through these two pieces of information, (iii) the next satellite passes in its own communication scope. The goal of such a computation is to let the device save energy by waking up when a DtS communication can happen, i.e., when a LEO satellite is in its transmission range.

In this sense, after collecting the satellite's orbital data transmitted by an eNB-equipped LEO satellite through downlink messages, a UE can calculate the trajectory of that satellite using algorithms like the Simplified General Perturbations Model 4 (SGP4) [28]. Indeed, given TLE sets, the SGP4 algorithm can calculate the position of a satellite at a specific time. Its high efficiency and accuracy make it a suitable choice for IoT devices.

As for the UE's position, some methods for locating IoT devices, including terrestrial-based LPWAN techniques, novel GNSS solutions, and innovative positioning techniques leveraging LEO satellite constellations, have been summarized and compared in [29]. Focusing on solutions for LEO satellites, there are two possible methods: some satellites collect uplink signals sent by the same ground UE, or the ground UE collects downlink signals. In both cases, a common aspect is measuring the Doppler frequency shift. On the one hand, the first method requires the participation of ground stations to integrate the uplink signals received by each satellite, which adds complexity to discontinuous systems and results in devices being unable to obtain their location. On the other hand, both methods require the simultaneous presence of multiple satellites sending or receiving the same copy of the signals. As mentioned earlier, these approaches are not feasible in scenarios with only a single satellite or a limited number of satellites. Similarly, an IoT-over-satellite-based framework [30] was proposed to achieve localization by using the angle of arrival (AoA) and Doppler shift by multiple satellites.

As said before, when the UE's location on the ground and the satellite orbit information are known, the most accurate way to predict the next satellite pass is to use the SGP4 algorithm to calculate both the satellite's position within a certain period in the future and its relative position to the UE. However, this method requires a certain amount of computational effort, which can pose a challenge for resource-constrained IoT devices. In [31], the authors proposed a novel method to calculate the revisit time of LEO satellites and applied it to the optimal design of remote sensing satellites. Yet, this method focuses on repeated ground track orbits [32], thus applying to a very specific subset of LEO orbits. To provide a more general solution, [33] proposed a semi-analytical method for calculating revisit times for the Walker constellation [34] with discontinuous coverage. However, only the maximum and average revisit time are discussed.

Finally, many studies have focused on the energy consumption of NB-IoT or LTE, such as [35]–[37]. These works provide valuable insights into the battery life optimization of ground NB-IoT networks. At the same time, the 3GPP technical report [12] offers crucial information on energy consumption in DtS scenarios.

Overall, with respect to the literature reviewed so far, this paper contributes the following: (i) an energy-efficient initial search method for downlink signals instead of methods combining GNSS with satellite ephemeris; (ii) a GNSS-independent technique able to estimate the UE's potential locations and predict future satellite passes based on the downlink synchronization method pictured in a previous contribution [19]; (iii) an iterative routine capable of refining the UE's estimated position and making it fully aware of upcoming communication windows; (iv) simulation results prove this strategy is more energy-efficient than GNSS-based methods.

III. Proposed approach

The rationale of the wake-up strategy introduced by this paper can be summarized as an "ask, learn, go" process. Indeed, the "ask" part is represented by a *Network Search* phase. When an NB-IoT UE needs to send or receive data through the Internet, it first needs to get synchronized with an eNB in its vicinity [38]. Ground UEs intermittently wake up and poll the radio for beacon signals, i.e., NPSS NB-IoT signals. Once an NPSS signal is detected, a *Synchronization to LEO satellites* phase can be started, and the wake-up strategy enters into its "learn" part. During this phase, the UE tries to predict when a LEO satellite will be within its range and builds up a list of next satellite passes. The UE wakes up at each expected satellite pass to check whether the prediction was correct and to continuously update the list. Once the UE verifies that its prediction is converging to stability, the

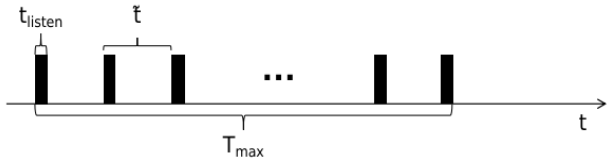


Fig. 1: Search for NPSS signal.

wake-up strategy enters into the *Steady State* phase, i.e., the "go" part of the strategy. With this organization in mind, this section is organized as follows. Sec. III-A pictures the actions taken during *Network Search* phase. Then, Sec. III-B describes the operations to be done during the *Synchronization to LEO satellites* phase. Finally, Sec. III-C provides details about how a UE can pass from the *Synchronization to LEO satellites* phase to the *Steady State* one.

A. Phase 1: Network Search

The *Network Search* phase represents Phase 1 (Fig. 2) of the wake-up strategy presented hereafter and refers to the period of time going from the UE's initial wake-up through the reception of the first NPSS from the eNB-equipped satellite. Since the UE does not have any information about the position of LEO satellites at bootstrap, it can only attempt continuous signal monitoring. However, considering the worst-case network scenario with a single satellite, its next pass into the UE's range could take up to several hours, and uninterrupted listening would quickly deplete a large portion of the stored energy. Therefore, an intermittent listening strategy may be used to reduce the UE's energy consumption.

As shown in Fig.1, the device wakes up at regular intervals $\tilde{t}$ and starts listening to the possible transmission of NPSS signals for a time interval t_{listen}. When t_{listen} elapses, the UE returns to sleep mode. If an NPSS has been received, the UE is ready for Phase 2 (described in Sec. III-B). Given that NPSS signals are transmitted every 10 ms and that each transmission lasts 1 ms, setting $t_{listen} = 11$ ms ensures that if a satellite is in the range of the UE for the duration t_{listen}, at least one NPSS signal will be received. Note that the propagation delay does not need to be considered in the t_{listen} because the synchronization signals are continuous. The interval between signal arrivals remains constant, irrespective of the propagation delay. In this configured access scheme, the length of the interval $\tilde{t}$ defines the duty cycle. The longer $\tilde{t}$, the smaller the duty cycle. If $\tilde{t}$ is too short, the UE will consume more energy. Contrariwise, if $\tilde{t}$ is too long, the satellite's arrival may be missed. Therefore, the setting of $\tilde{t}$ is of core importance for finding a tradeoff between energy consumption and protocol reactivity. Such a setting will be discussed in Sec. IV on the basis of some simulation results related to various satellite constellations. To anticipate possible hardware issues or other sort of problems, the maximum duration T_{max} of Phase 1 can be properly set to a limited value. When T_{max} elapses, the UE stops searching for NPSS signals. In this paper, for evaluation purposes, T_{max} is set to be infinity for Phase 1.

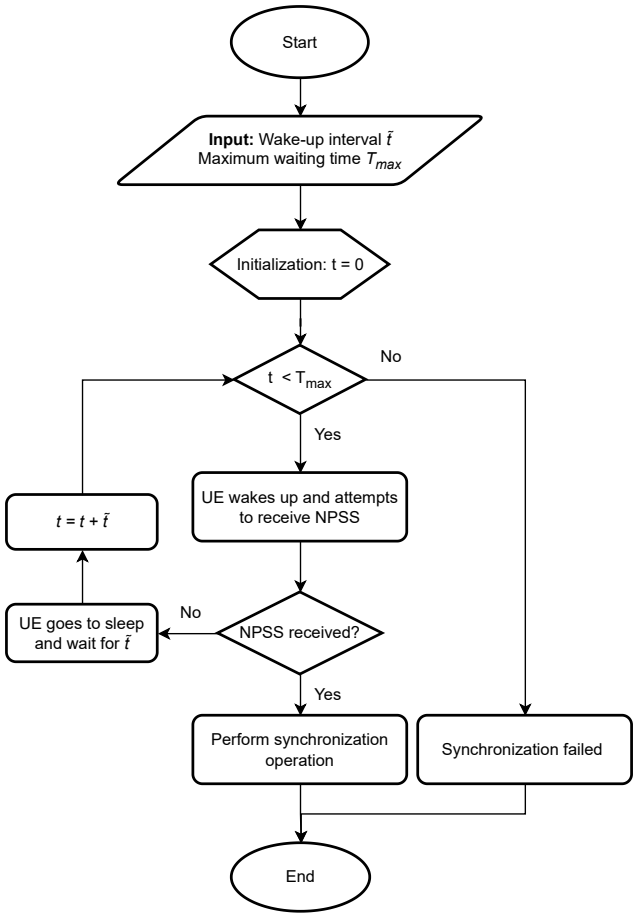


Fig. 2: Wake-up Strategy: Network Search flowchart

B. Phase 2: Synchronization to LEO satellites

Building upon the reception of the first NPSS signal, Phase 2 of the proposed strategy is meant to let a UE achieve and keep *Synchronization to LEO satellites*. During such a phase, the UE repeatedly performs estimations and makes predictions. In detail, if a LEO satellite is within the communication reach, a UE tries to estimate the satellite's trajectory through the method described in [19] and quickly reviewed in Sec. III-B1. The knowledge of the satellite's trajectory is useful for two reasons: (i) it allows the UE to pre-compensate in time and frequency the reception of the subsequent SIBs; (ii) the associated maximum elevation angle is used to estimate two potential positions of the UE on the Earth. The latest estimation takes as input also timing and orbital information included in the received SIBs, and it is described in Sec. III-B2. Through the estimation of the UE positions, the UE can attempt the prediction of next satellite passes, as described in Sec. III-B3. Thanks to such a prediction, the UE can go back to the sleep state after a satellite pass and wake up again just before the next one. The wake-up strategy is then described in Sec. III-B4.

1) Estimation of Maximum Elevation Angle: As described in [19], by measuring the Doppler shift associated with the reception of multiple downlink NPSS signals, it is possible to figure out the relative position of the satellite's trajectory with respect to the UE. Indeed, during a pass, the elevation angle θ increases until a maximum value θ^* when the satellite is closest to the UE and then decreases until the satellite gets out of the communication range of the UE. The carrier frequency f_0 used here and for the rest of the paper is 2.4 GHz, and the assumed value of minimum elevation angle $\theta_{\min}$ is $30°$.

Further than the maximum elevation angle θ^*, it is possible to measure the instant $t = t_0$ when the very first NPSS was received, with the origin of the timeline $t = 0$ being the instant when $\theta = \theta^*$. In other words, if the UE receives the NPSS signal before the satellite reaches its maximum elevation angle, then t_0 is negative. On the contrary, it is positive.

Considering the presence of measurement errors, the least squares method was used in [19] to estimate the two unknown parameters θ^* and t_0 based on the measurements of the Doppler frequency shift of some NPSS signals received at fixed intervals Δt. The supposed model for the frequency shift is

$$D(\Delta t) = \\ = -\frac{f_0}{c}\frac{RR_E\omega_s \sin(\omega_s(t_0+\Delta t))\cos(\alpha_0)}{\sqrt{R_E^2+R^2-2RR_E\cos(\omega_s(t_0+\Delta t))\cos(\alpha_0)}}, \tag{1}$$

where α_0 is the angle formed between (i) the line joining the center of the Earth with the UE and (ii) the line joining the center of the Earth and the satellite when $\theta = \theta^*$ (i.e., $\angle SOU_{1/2}$ in Fig.3). In this expression, R_E is the radius of the Earth, R is the radius of the orbit, ω_s is the satellite's angular velocity, and c is the speed of light. The exact formulation for α_0 as function of θ^* is:

$$\alpha_0 = \arccos\left(\frac{R_E}{R}\cos\theta^*\right) - \theta^* \tag{2}$$

and it can be quickly verified that it is a bijective function for the value of θ^* in the interval $[0, \pi/2]$. For the sake of simplicity, alongside this paper, the model of (1) is used to estimate the values of α_0 and t_0. The results in [19] show that even with a large measurement error standard deviation (200 Hz), this method can keep the estimated Doppler curve within an acceptable range of the original curve for tens of seconds to more than a minute. This ensures that receiving multiple NPSS signals supports a complete NB-IoT communication session (uplink and downlink synchronization).

Considering the computational constraints of IoT devices, performing trigonometric or root calculations could be challenging. Current approaches, such as linear approximations or look-up tables [39], can simplify the calculations required with low memory. These approaches can efficiently address the complexities involved in computations like Doppler curve estimation and the rest of this section, making them more suitable for IoT devices. The evaluation of the actual impact of such approximations on the overall system performances is out of the scope of this paper and will be tackled in future investigations.

2) Positions of UE: To identify the network and confirm the timing information of the radio frame, UE needs to search for a Narrowband Second Synchronization Signal (NSSS). Then, the downlink synchronization process with the eNB is completed, and the UE can now receive additional system information: the Master Information Block (MIB) and the System Information Block (SIB). Among all SIBs defined in the 3GPP release [40], SIB31 and SIB32 address the need for Non-Terrestrial Networks (NTN). Together with SIB16, they are of utmost importance in this paper since they provide all necessary information to UEs willing to communicate with eNB-equipped LEO satellites. In detail, SIB16 encodes the Coordinated Universal Time (UTC) with an accuracy of 10 ms. Instead, SIB31 provides some information related to the serving satellite, which includes orbital parameters in either ephemerides or instantaneous values of the satellite state vectors.

The previously estimated α_0 and t_0, combined with some data broadcast by the satellite within SIB16 and SIB31, can be utilized to estimate 2 approximate positions of the UE. One of these 2 locations is the actual position of the UE. The other location is exactly the mirrored point of the actual position with respect to the projection of the satellite's trajectory on Earth. A method to identify the actual position is later proposed in Sec. III-C. To calculate the UE position, a reference point needs to be selected first. Herein, such a point is the projection on the Earth of the satellite in the instant when it is closest to the UE, i.e., when $\theta = \theta^*$. The UTC time T^* of this instant can be computed as:

$$T^* = T_r - (t_0 + \tau_{sib}), \tag{3}$$

where T_r is the UTC time communicated within SIB16. For a satellite traveling at the height of 600 km on the ground level, SIB16 is usually received with a delay of 2 to 4 ms, but since the time accuracy of T_r is 10 ms, this delay can be neglected. Instead, τ_{sib} is the time interval duration between the reception of the first NPSS signal and the reception of SIB16. This paper assumes a worst-case scenario where this interval is 4 seconds long. Then, the coordinates of the satellite's projection on the Earth when closest to the UE are calculated through the SGP4 algorithm [28] using as input T^* and the vector of ephemerides orbital data $\mathbf{E}$ sent in SIB31:

$$(\phi_s, \lambda_s, \mathbf{v_s}) = SGP4(T^*, \mathbf{E}). \tag{4}$$

In such an equation, ϕ_s is the latitude of the satellite, λ_s is its longitude, and $\mathbf{v}$ is the speed of the satellite expressed in the associated Cartesian coordinates (v_s^x, v_s^y, v_s^z).

Based on the joint knowledge of the satellite coordinates $(\phi_s, \lambda_s, \mathbf{v_s})$, of the ephemerides $\mathbf{E}$, and of the angle α_0 between the UE and the satellite when they are closest, the UE can compute the two possible locations $U_1(\phi_1, \lambda_1)$ and $U_2(\phi_2, \lambda_2)$ for its own position on the Earth. For the sake of readability, Fig. 3 plots the system model that will be used in the following discussion, with the Earth's center placed at O. In detail, the orange great circle represents the projection of the satellite's trajectory on the ground. Such a trajectory intersects the equator (i.e., the black great circle) in 2 points. One of them is indicated with A in the figure, while the other

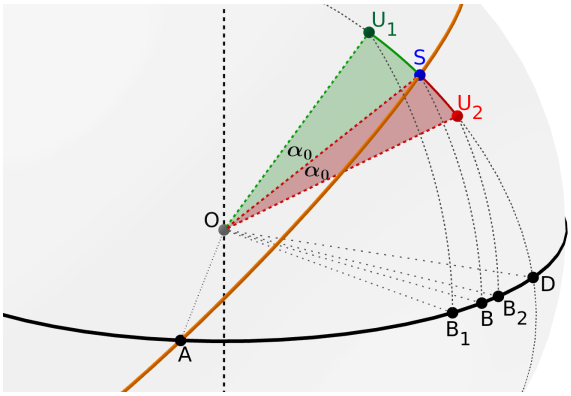


Fig. 3: System model.

point is located at the antipodes of A and not shown in the figure. Of these 2 intersection points, A is the closest to the actual UE position U_1, and it will be the only one considered in the next discussion. The closest satellite position to U_1 is S, while the mirrored location is U_2. The points U_1, S, and U_2 are all aligned on the great circle plotted with a dotted line and intersecting the equator at D. Such a great circle is orthogonal to the orange projection of the satellite's trajectory.

In such a scenario, the latitudes of U_1 and U_2 can be calculated considering the spherical triangles U_1B_1D and U_2B_2D, noting that $\angle U_1DB_1 = \angle U_2DB_2 = \angle SDB$, and using the spherical law of sines $\sin(\phi_{1/2}) = \sin(\angle SDB)\sin(\angle SOD \pm \alpha_0)$, with the 2 angles $\angle SDB$ and $\angle SOD$ still being unknown. Focusing on the spherical triangle SBD and according to the spherical law of sines, $\angle SOD = \arcsin[\sin(\phi_s)/\sin(\angle SDB)]$. Hence, the remaining unknown is $\angle SDB$. So, considering the spherical triangle ASD, and using the second spherical law of cosines, $\angle SDB = \arccos[sin(i)cos(\angle AOS)]$. Finally, using the law of sines on the triangle ASB, it is possible to evaluate $\angle AOS = \arcsin[\sin(\phi_s)/\sin(i)]$. Putting all together, the latitudes of U_1 and U_2 are

$$\phi_{1/2} = \arcsin[\sin(\phi_s)\cos(\alpha_0) \pm \cos(i)\sin(\alpha_0)]. \quad (5)$$

Then, to compute the longitudes of U_1 and U_2, note that the unit vector $\hat{\mathbf{u}}$ placed at O and pointing to any of such positions forms an angle α_0 with the unit vector $\hat{\mathbf{s}}$ still placed at O and pointing to S. This corresponds to let the scalar product $\hat{\mathbf{u}} \cdot \hat{\mathbf{s}}$ be equal to $\cos(\alpha_0)$. After some calculations, the following equation can be found:

$$\cos(\lambda_{1/2} - \lambda_s) = \frac{\cos(\alpha_0) - \sin(\phi_{1/2})\sin(\phi_s)}{\cos(\phi_{1/2})\cos(\phi_s)}. \quad (6)$$

Solving (6) for a specific latitude ϕ, it is possible to find 2 points with longitudes λ^w and λ^e forming an angle α_0 with the satellite projection S. Specifically, going from West to East λ^w precedes λ_s, and λ_s precedes λ^e. In total, there are: (i) 2 possible longitudes associated to ϕ_1, i.e., λ_1^w and λ_1^e; (ii) 2 possible longitudes associated to ϕ_2, i.e., λ_2^w and λ_2^e. However, only one longitude for each latitude is the correct one. It can be quickly recognized that if the satellite is ascending, the correct longitude associated with ϕ_1 is λ_1^w, while the correct longitude associated with ϕ_2 is λ_2^e. Instead, if the satellite is descending, the correct longitude associated with ϕ_1 is λ_1^e, while the correct longitude associated with ϕ_2 is λ_2^w. Given that a satellite is ascending (descending) if the component v_s^z of its speed along the z axis is positive (negative), and indicating with $\mathrm{sgn}(\cdot)$ the sign function, the longitudes of U_1 and U_2 can be evaluated as:

$$\lambda_{1/2} = \lambda_s \mp \mathrm{sgn}(v_s^z)\arccos\left[\frac{\cos(\alpha_0) - \sin(\phi_{1/2})\sin(\phi_s)}{\cos(\phi_{1/2})\cos(\phi_s)}\right]. \quad (7)$$

3) Prediction of next passes: To predict the next satellite passes in case of discontinuous coverage, the current 3GPP specification [40] defines SIB32 as a means used by an eNB-equipped satellite to broadcast the orbital parameters related to 4 satellites at most (see the definition of the parameter `maxSat-r17`). While it is understandable the reason for such a small number of satellites, i.e., the cumulative orbital data related to each satellite within a constellation would make the transmission of a too-long SIB32 unfeasible in practice, in this paper, it is argued that SIB32 can be forged in a different way. In fact, it must provide a few constellation parameters so that the receiving UE can compute the next passes related to all the satellites belonging to the constellation. This choice has some advantages: (i) there is no need for the satellite and/or the network to make a selection of the `maxSat-r17` satellites advertised in a given spot of the Earth, thus increasing the scalability of the system; (ii) the knowledge of the whole constellation would allow all UEs to exactly compute what is the very next pass by any satellite in the constellation, thus increasing the communication chances; (iii) The SIB length would be much shorter, thus reducing the energy consumption and resources required by both LEO satellites and UEs. Additionally, the UEs only need to receive this SIB32 once during initial communication with the satellite. It can be almost ignored over the long term. Therefore, in the simulations later, the energy variations due to receiving SIB32 are not considered significant enough to affect the outcomes. In Release 18, 3GPP also introduces SIB33, which provides satellite assistance for neighboring cells. However, the strategy discussed in this paper does not require this additional information. Therefore, SIB33 is not considered in the analysis.

Here, the Walker Constellation is employed in this paper as it is a widely used configuration for LEO satellite systems [34]. All satellites have the same orbital inclination in this configuration and are evenly distributed across multiple orbital planes. This uniform distribution ensures a balanced coverage of the Earth's surface. Any Walker Constellation is indicated with the notation $i : T/P/F$, where i represents the inclination angle, T is the total number of satellites, P is the number of orbital planes, and F the relative phasing between satellites in adjacent planes. These parameters are broadcast in SIB32. Note that the inclination angle i is implicitly understood through SIB31 and is related to the serving satellite. The total number of satellites T can be encoded with 1 byte to allow up to 255 satellites. As well, P and F can be encoded each with 1 byte. As further detail,

F is set to be 0 when there is only a single orbital plane, i.e., when $P = 1$. Theoretically, there are no absolute maximum values for the parameters T, P, and F in satellite constellation configurations, as they depend on the specific requirements and design considerations of particular missions. The size of these parameters can be adjusted according to different applications. Assuming to represent each of those 3 values with 2 bytes, a Walker constellation with a maximum size of 65535 satellites would be able to capillary cover the whole Earth surface with satellites, each featured by a 100 km wide footprint. Hence, a SIB32 long 6 bytes is more than sufficient to allow all constellations targeting a discontinuous coverage.

By combining the estimated UE's coordinates (see Sec. III-B2) and the trajectory of all satellites in a Walker-like constellation, it is possible to fill up a sorted list with all next passes of satellites in the considered constellation up to a given time T_{lim}. First, for each of the 2 possible locations of the UE position, it is possible to build a time-sorted list by adding at each step the pair $(time, longitude)$ associated with the next satellite belonging to the constellation and passing at the same latitude associated to the considered location. This list is continuously filled with next passes at the same latitude until T_{lim}. Then, the passes with "non-compliant" longitudes will be discarded. Finally, the 2 lists are merged to create a single time-sorted list containing the next feasible passes through the UE possible locations.

In more detail, it must be observed that during a nodal period P_N (i.e., the time taken by a satellite to complete an entire orbit around the Earth and pass by the same latitude with the same direction), the Earth rotates for exactly the difference in longitude

$$\Delta\lambda = P_N(\dot{\Omega} - \omega_E), \tag{8}$$

where ω_E is the angular velocity of the Earth and $\dot{\Omega}$ is the nodal regression rate of the satellite orbit [41]. However, given the first estimated longitude $\lambda_1^{u/d}$, for the objective of the proposed approach, it is more important to find the longitude of the next pass $\lambda_1^{d/u}$ through the same latitude ϕ and opposite direction (the "u" means upside, the "d" means downside):

$$\lambda_1^{d/u} = \lambda_1^{u/d} + \pi \mp 2\arcsin\left(\frac{\tan(\phi)}{\tan(i)}\right) + \\ + \left(0.5 \pm \frac{1}{\pi}\arcsin\left(\frac{\sin(\phi)}{\sin(i)}\right)\right)\Delta\lambda. \tag{9}$$

In general, the set of all longitudes associated with passing through the same latitude is:

$$\left\{\lambda_j^{u/d} | \lambda_j^{u/d} = \lambda_1^{u/d} + (j-1)\Delta\lambda, \text{ for } j = 1, 2, ... \left\| \frac{T_{lim}}{P_N} \right\|\right\} \tag{10}$$

Referring to Walker Constellation, the longitudes in different planes are [33]

$$\left\{\lambda_{j,k}^{u/d} | \lambda_{j,k}^{u/d} = \right. \\ \left. = \lambda_j^{u/d} + 2\pi(k-1)\left(\frac{1}{P} + \frac{F}{T}\right), \text{ for } k = 1, 2, ..., P\right\}. \tag{11}$$

Finally, the complete set of longitudes for the constellation is:

$$\left\{\lambda_{j,k,o}^{u/d} | \lambda_{j,k,o}^{u/d} = \lambda_{j,k}^{u/d} + \frac{(o-1)P}{T}\Delta\lambda, \text{ for } o = 1, 2, ..., \frac{T}{P}\right\}. \tag{12}$$

with the longitude of the reference point being $\lambda_{1,1,1}^{u/d}$, and its time instant indicated as $T_{1,1,1}^{u/d}$.

Now that the list of longitudes where each satellite passes through the same latitude of the UE within a certain period of time is built, it is necessary to determine which longitudes are within the communication range of the UE and find the elapsed time since the time instant $T_{1,1,1}^{u/d}$ associated to reference point, so that the revisit time for longitude $\lambda_{j,k,o}^{u/d}$ is

$$T_{k,o,j}^{u/d} = T_{1,1,1}^{u/d} + (j-1)P_N + \frac{(k-1)F}{T}P_N + \frac{(o-1)P}{T}P_N \tag{13}$$

Recalling that the area covered by a traveling satellite is called swath [42], it is worth defining Λ as the width of the satellite swath at the latitude ϕ of the UE. In other words, when a satellite passes through latitude ϕ, if the satellite's longitude $\lambda_{j,k,o}^{u/d} \in [\lambda - \frac{\Lambda}{2}, \lambda + \frac{\Lambda}{2}]$, the device can communicate with this satellite. Considering also α_0 and the UE's latitude, the swath can be calculated as follows:

$$\Lambda = \cos^{-1}\left(\frac{\sin(\phi)\cos(i) - \sin(\alpha_0)}{\sin(i)\cos(\phi)}\right) - \\ \cos^{-1}\left(\frac{\sin(\phi)\cos(i) + \sin(\alpha_0)}{\sin(i)\cos(\phi)}\right). \tag{14}$$

A larger swath Λ' is required to predict the next pass, depending on the error margin. This also means the device can move within a certain range within the maximum error distance. This value can also be adjusted according to different applications. The formula to calculate Λ', taking into account the acceptable distance error, is as follows:

$$\Lambda' = \Lambda + 2 * \frac{D_{error}}{R_E * \cos(\phi)} \tag{15}$$

4) Wake up on predicted satellite pass: Once a UE has caught the first NPSS signal (see Sec. III-A), estimated the maximum elevation angle of satellite's trajectory (see Sec. III-B1), computed 2 possible locations for its own position (see Sec. III-B2), and predicted the associated next satellite passes (see Sec. III-B3), it can communicate with the satellite. After that, it can turn its radio off and stay in sleep mode.

Thanks to the knowledge of the next passes, the UE can wake up exactly again around the estimated revisit time T_{next} associated with the first item on the list. The same listening strategy pictured in Fig. 1 is used. It is important to notice that since the prediction is based on 2 potential locations, it is possible that the first item on the list is associated with the wrong location. As well, even if the first item is associated with the correct location, there could be an error in the predicted revisit time. Given that the maximum visibility time τ_v of the satellite is around 4 minutes for a 600 km LEO satellite. If no NPSS has been captured before $T_{next} + \tau_v$, then it is reasonable to assume that the next pass was associated with the wrong location. It also seems reasonable to let the UE

start listening not before $T_{next} - \tau_v$. As a matter of fact, the uncertainty on the start of the visibility time is assumed to be bounded to τ_v before and after the computed revisit time. In this way, the search time T_{max} is set to twice the maximum visibility time of the satellite, so around 8 minutes for the targeted scenario. If no NPSS has been received, the UE will go to sleep and start the same procedure around the next revisit time in the list of future satellite passes. Additionally, since the value of T_{max} during this phase is quite short if compared with the *Network Search* phase, the interval $\tilde{t}$ between 2 consecutive polls can be shortened to 30 seconds to capture with more precision the start of the visibility time and have more time to communicate.

C. Getting to Phase3: Steady State

The *Sychronization to LEO satellites* phase pictured above in Sec. III-B requires that a UE always wakes up for the next satellite pass present in the built sorted list of predicted passes. This is done in order to perform a position estimate and update the list of next passes. A UE will also wake up if the next pass is associated with a mirrored position, thus wasting energy. In this sense, a UE should combine the subsequent position estimations in order to select just the actual position and narrow down the associated error. The routine pictured below is meant to refine the UE's estimated position through multiple satellite passes. The stopping condition for this process is given by the number of estimation N_e, whose value will be discovered through simulation-based analysis in Sec. IV-B. It is worth noting that the routine behavior differs among (i) the very first satellite pass (the one triggering the switch from *Network Search* to *Synchronization to LEO satellites*), (ii) the second pass, and (iii) the following ones up to N_e. The details of the procedure are shown in Figure. 4.

After the first satellite pass, the UE will discover 2 locations for the UE position, namely U_1 and U_2. These 2 locations and the location S of the satellite projection on the ground closest to the UE are stored as V and R and used for the next iteration of the routine. Based on the computed list of time-sorted next passes, the UE will wake up for the next pass. If no NPSS is received, the UE goes to sleep and wakes up again for the following pass in the list. This routine continues until the UE wakes up and finds out that a satellite pass is happening.

At this stage, the second run of the routine can be executed. In detail, the satellite estimates the satellite position when it is closest, S, and the 2 possible locations for its own position U_1 and U_2 just like the first pass. In addition, it computes a reference point $R = (R + S)/2$. Then, it selects between the location found during the first pass, $V[0]$, and $V[1]$, the one which is the closest to R, stored as V; it also selects between the location found during the second pass, U_1 and U_2, the one which is the closest to R, namely U_C. It can now estimate its position as $V = (V + U_C)/2$.

The third and following runs of the routine are executed when a satellite pass is detected during the next wake-up according to the updated list of satellite passes. Specifically, at run n, the satellite location S and the 2 locations for the UE U_1 and U_2 are estimated. Based on these, the reference point $R = ((n-1)R+S)/n$ is computed. Then, the UE selects between U_1 and U_2 the closest location U_C to R and can compute the estimate of its own position $V = ((n-1)V+U_C)/n$.

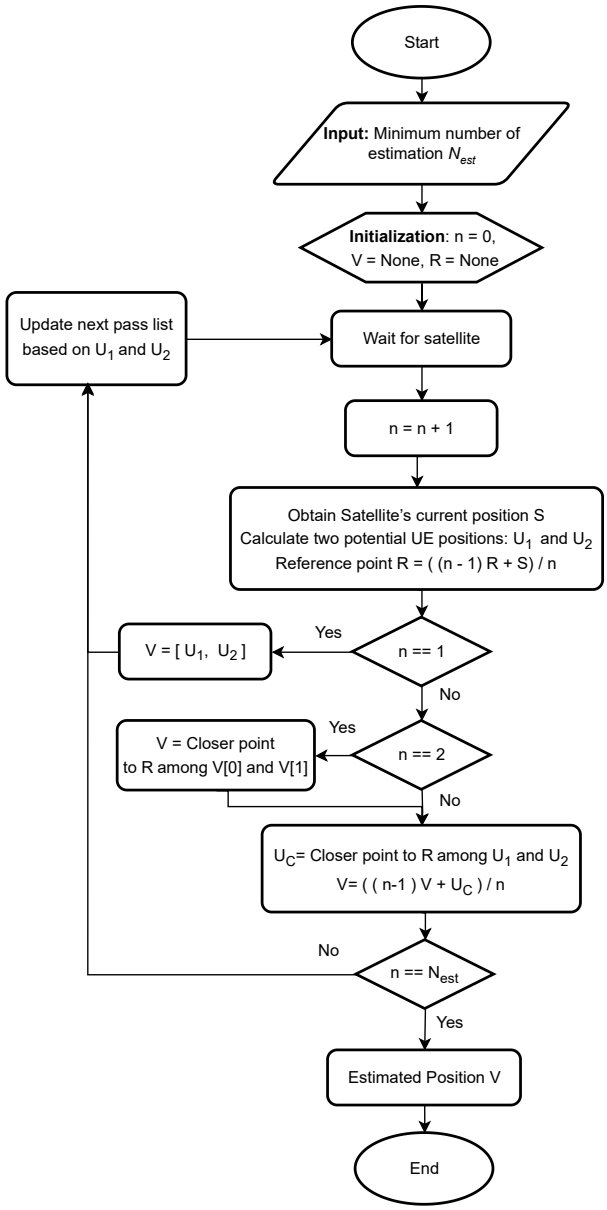


Fig. 4: Wake-up strategy: Determination of UE's Position flowchart

After the N_e run, the UE can wake up just when it needs to transmit IoT data through the LEO constellation. To do that, the list of next passes used is based only on the estimate V found at the end of the N_e-th run.

Additionally, it is essential to clarify that this routine does not involve any computationally intensive operations. Each time a satellite passes, the device only updates two values:

V, the estimated position, and R, a reference point based on the satellite's position. These straightforward updates involve averaging operations to refine the UE's position over time. Since the routine merely updates these two points based on calculations during each satellite pass, the complexity remains minimal, making the approach both efficient and lightweight for the device.

IV. Simulation Results

In this section, some results are presented based on the simulation of the proposed strategy. The simulation environment can work with various types of satellite constellations and is able to mimic random deployments of UEs on the ground. All satellite trajectories are generated using the Skyfield library [43]. Each NPSS signal received by a UE is affected by the Doppler effect. Additionally, as discussed in the previous section, due to hardware limitations, the accuracy of the measured signal frequency and the following calculations cannot be perfect. To represent the measurement noise more realistically, each signal is affected by Additive White Gaussian Noise (AWGN). This is achieved by incrementing (or decrementing) the original value of the Doppler shift by means of a random value drawn from a zero-mean normal distribution with configurable standard deviation σ. The value of σ has been varied in the interval $[0, 300]$ Hz to feature different noisy environments. It is recognized that real-world deployments involve additional complexities and factors that are hard to be fully captured through simulations. However, by adding AWGN to simulate signal noise, the results reasonably reflect how environmental factors can degrade the signal quality. Varying the intensity of the noise allows the simulation to mimic the kind of challenges that would be encountered in real-world conditions. While it is impossible to replicate every aspect of the natural environment, this approach still offers a practical way to evaluate the strategy's performance across different noise levels.

All the NB-IoT settings are based on the 3GPP technical report [12]. As for the GNSS receiver, ground devices equipped with GNSS should initiate the GNSS receiver for positioning before each communication. Depending on the situation, the start of a GNSS receiver can be referred to as "cold start", "warm start", or "hot start" [44]. A "cold start" occurs when the receiver is used for the first time or after a long period of inactivity. Instead, a "warm start" occurs when the receiver has not received GNSS signals for over 2 hours. Then, a "hot start" occurs when the receiver has all the necessary up-to-date information to lock onto the satellite quickly. From the energy consumption perspective, this paper will compare the presented strategy against the solution that consumes the lowest energy, i.e., the one with integrated GNSS modules [12].

Contrariwise, the GNSS-free strategy introduced later requires that UEs receive several NPSS signals to achieve and keep synchronization. This process can be completed in 1 second [19]. Recalling that NPSS signals are transmitted for 1 ms every 10 ms by eNBs, UE can listen up to 100 NPSS during 1 second. However, the NPSS collection strategy on the UE can be implemented to listen to less than 100 NPSS within the 1 second observation time, with the interval between two consecutive NPSS receptions being higher than 10 ms. In all cases, the chosen value of power consumed for listening to incoming downlink signals (i.e., NPSS, NSSS, MIB, and SIBs) is the highest setting presented in [12]. Such a conservative choice has been made to compare the best GNSS-based solution with the worst-case GNSS-free settings. Table I outlines the parameters used to configure the simulations.

TABLE I: Parameters of the Simulations

Parameters	Value
Power of GNSS	37 mW
Power of RX NPSS/NSSS	90 mW
Power of RX MIB/SIB-satellite	90 mW
Power of Idle	3 mW
Power of Sleep	0.015 mW
GNSS cold start duration	30 s
GNSS warm start duration	5 s
GNSS hot start duration	1 s
RX SIB-satellite duration	24 ms
RX MIB duration	60 ms
Search for NPSS duration	11 ms

TABLE II: Network Search energy consumption comparison

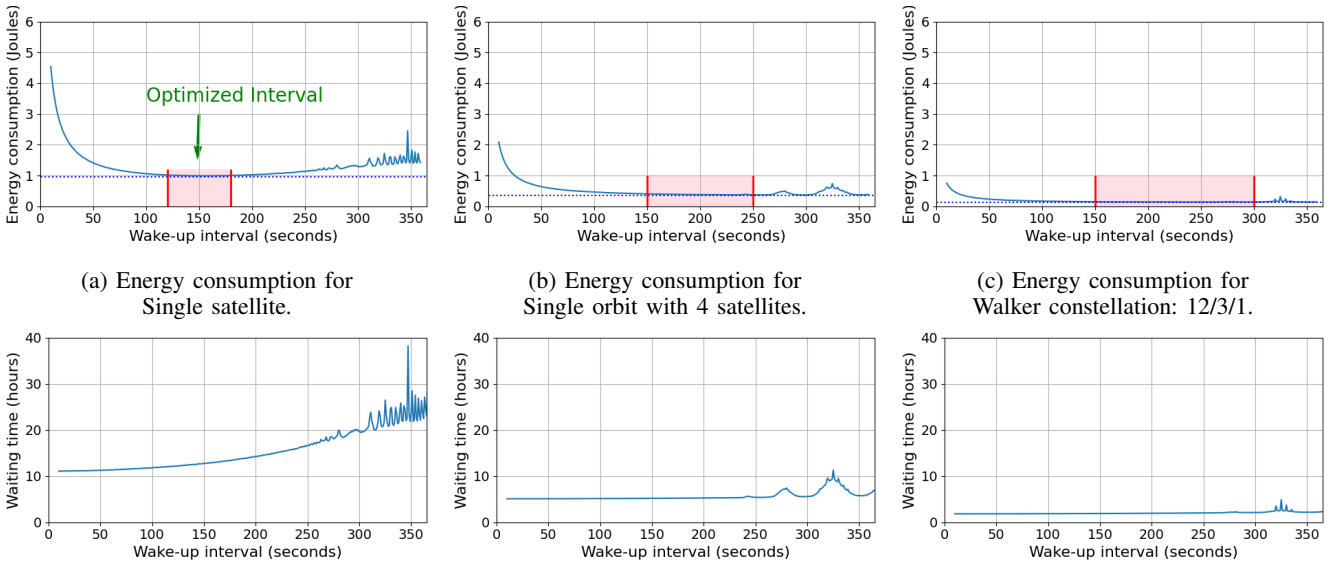
Constellation types	GNSS implemented	No GNSS
Single satellite	3.6 J	0.99 J
Single Orbit (4 satellites)	3.27 J	0.37 J
Walker constellation (12/3/1)	3.11 J	0.14 J

Finally, three different discontinuous coverage situations are considered in the context of the present contribution, all represented as Walker Constellations. In the worst case, the "Single satellite" case offers a very sporadic coverage, limiting the satellite's visibility on any Earth spot to around 20 minutes per day, with an average of 2 passes per day [7]. The second case is "Single orbit," which involves multiple satellites organized into a single orbit. Finally, the "Multiple orbits" scenario represents a generic sparse Walker constellation, with multiple orbits and several satellites per orbit. It has to be noted that only circular orbits have been used for simplicity, with the satellite's altitude fixed to a typical value of 600 km. The inclination angle is set to 85° to allow a fair coverage of the poles.

In detail, this section provides some rules of thumb to configure the wake-up strategy presented in Sec. III with the proper settings and evaluates the energy consumption due to it in the long term. Sec. IV-A discusses how to set the wake-up interval $\tilde{t}$ introduced in Sec. III-A. Next, Sec. IV-B shows how to set up the process that lets the strategy enter the Steady State phase, and Sec. IV-B shows the latency and energy consumption to get into Steady State phase in different scenarios. Finally, Sec. IV-D provides an evaluation of the energy consumed by the presented GNSS-free wake-up strategy and compares it with the standard GNSS-enabled configuration.

A. Optimal wake-up interval during Network Search

As introduced in Sec. III-A, when a UE is switched on (or in general when it is bootstrapped), it must first join the NB-



(a) Energy consumption for Single satellite.

(b) Energy consumption for Single orbit with 4 satellites.

(c) Energy consumption for Walker constellation: 12/3/1.

(d) Latency for Single satellite.

(e) Latency for Single orbit with 4 satellites.

(f) Latency for Walker constellation: 12/3/1.

Fig. 5: Performance evaluation of Network Search.

IoT network. To do so, it starts a polling procedure where it intermittently wakes up for very short time intervals, i.e., 11 ms, and listens to the radio for possible incoming NPSS signals. The challenge inherent to this approach is to find a suitable wake-up interval between 2 consecutive polls in order to maximize energy efficiency while preserving the reactivity of the joining protocol (see Fig. 1).

The results of the simulation-based analysis conducted to address this issue are shown in Fig. 5. In detail, Fig. 5a, 5b, and 5c show the average energy consumed by a UE implementing the Network Search procedure as a function of the configured wake-up interval $\tilde{t}$. Instead, Fig. 5d, 5e, and 5f picture the average time elapsed before the first NPSS signal is received, and such a value is plotted as a function of the configured wake-up interval $\tilde{t}$ as well. Then, Fig. 5a and 5d refer to the scenario with a single satellite, while Fig. 5b and 5e consider the single orbit configuration, and Fig. 5c and 5f deal with a Walker constellation 12/3/1. In all figures, the value $\tilde{t}$ has been varied from 10 seconds to 6 minutes, with an incremental step of 1 second. For any given value of $\tilde{t}$, 400 random locations on the Earth have been selected. For each location, 1000 random instants of time have been chosen to mimic different bootstrap times with respect to the current positions of satellites in the sky. Hence, each point in the plots of Fig. 5 represents the average of 0.4×10^6 simulations.

Starting from the analysis of the energy consumption due to the very first Network Search, as it can be seen in Fig. 5a, 5b, and 5c, the length of the interval $\tilde{t}$ has a significant impact on it. Indeed, if $\tilde{t}$ is too short, UEs wake up to check for downlink NPSS signals more frequently than needed. Given that the time interval between satellite passes over a given spot on the Earth can range from several minutes to over 10 hours, depending on the constellations, frequent wake-ups lead to a significant waste of energy. On the other hand, if $\tilde{t}$ is too long, the UE can miss satellite passes more frequently, resulting in the inability to join the NB-IoT network for a very long time. In general, it may be noticed that the lowest values of energy consumption can be achieved for values of $\tilde{t}$ close to the average time of a LEO satellite pass over any spot on the Earth: if a UE wakes up frequently enough not to miss a satellite pass, the chance of saving energy is higher. In addition, due to the approximate periodicity of satellite passes, some specific wake-up frequencies may cause the UE to miss multiple satellite passes in a row. Although energy consumption is lower in sleep mode, sleeping too long can still result in a significant cumulative energy consumption. Considering the need for UEs to transmit uplink IoT data, the prolonged inability to get connected to the network translates into big amounts of cached data, which in turn may affect the communication efficiency and result in delayed or failed data transmission.

Furthermore, comparing the single satellite scenario (see Fig. 5a) with the single orbit one (see Fig. 5b), the increased number of satellites lets the interval between satellite passes decrease, thus leading to a decreased waiting time for a UE to join a DtS NB-IoT network and a reduced energy consumption. This is even more evident by inspecting the scenario with a Walker constellation Fig. 5c). One final remark on the energy consumption can be made by observing that the availability of more satellites leads to the expansion of the range of optimal values for $\tilde{t}$. If this interval ranges between 120 and 180 seconds in the single satellite scenario, in the case of multiple satellites in a single orbit, the optimal values of $\tilde{t}$ vary between 150 and 250 seconds. For a Walker constellation, which involves a larger network of satellites distributed over multiple orbits, the wake-up interval can be further expanded to $[150, 300]$ seconds.

With regard to the analysis of the measured latency between

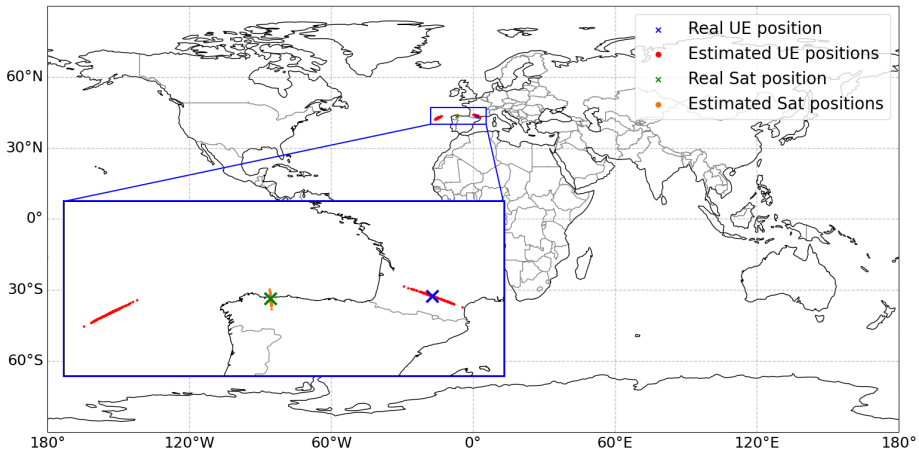


Fig. 6: Possible estimations of one single pass with $\sigma = 50$.

the instant when the UE is bootstrapped and the instant when it gets synchronized to the NB-IoT network, Fig. 5d, 5e, and 5f show that it looks like very much similar to the energy consumption. However, for very small values of $\tilde{t}$, the latency is just an increasing function of the wake-up interval. For example, by having a close look at the energy consumption and the measured latency in the single satellite scenario (Fig. 5a and 5d), it may be noticed that for $\tilde{t}$ becoming smaller and smaller than 100 seconds, the energy consumption increases, but the latency decreases. In other words, waking up more frequently is energy expensive, yet it allows a fast join to the network. Hence, the correct setting of $\tilde{t}$ must be driven by the final IoT application: for time-critical data, it may be more convenient to trade off the energy consumption in favor of the fastest Network Search phase. Although the study was limited to using the Walker constellation, the worst-case scenario still involves only one satellite in any constellation configuration. As the number of satellites increases, both energy consumption and latency are expected to decrease, and the wake-up intervals can become longer. Therefore, 150 seconds in wake-up intervals can be applied to any type of constellation, and it will also be used in upcoming simulations. However, this value may require adjustments for different altitudes and orbital inclinations, which will be further investigated in future work and are not covered in this paper.

Finally, it is worth providing a comparison with the mandatory GNSS-enabled implementation. When a GNSS-enabled UE is bootstrapped, the GNSS receiver must run a "cold start". Assuming that the information about satellite constellations is pre-loaded, a UE only needs to enter sleep mode after startup and remain in such a state until the first expected pass of the satellite. Table II shows the resulting comparison between GNSS-enabled UE and GNSS-free UE for different satellite constellations. In that, the GNSS-free UE is set with an optimal value of $\tilde{t}$. It can be seen that there is a clear reduction of energy consumption when using the GNSS-free solution, mainly because the cold start of GNSS is very much power-hungry.

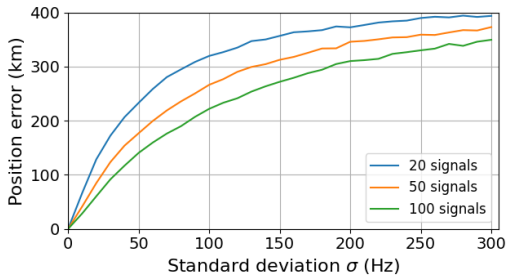
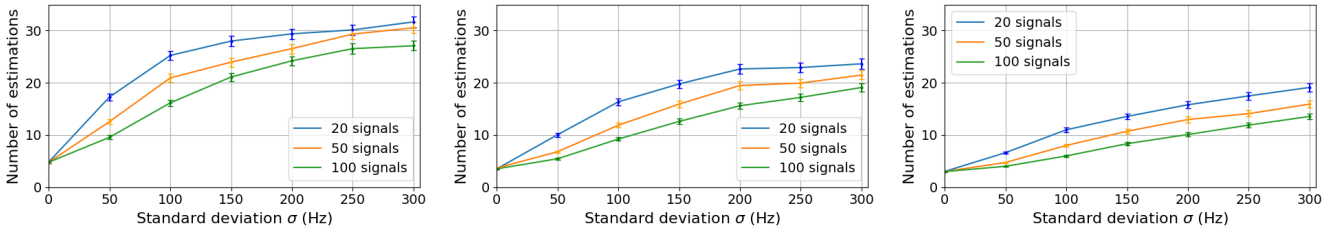


Fig. 7: Error of estimation.

B. *Optimal number of estimation runs to get into Steady State*

After the very first synchronization based on the reception of several NPSS signals (see Sec. III-B1), a UE can estimate 2 possible locations for its actual position by using the method introduced in Sec. III-B2. To help the reader quickly figure out what are the outputs of the position estimation, Fig. 6 shows the example of a satellite running across the same path. This corresponds to the orbital plane of the satellite rotating synchronously with the Earth. Even if this is physically impossible in practice, it allows a quick rendering of the estimated locations (in red) with respect to the actual position of the UE (in blue). The inclination angle of the orbit is $85°$, and a satellite goes from North to South in the pictured trajectory. The UE wakes up exactly at a specific instant of time, and it is able to uniformly capture 50 NPSS signals (broadcast by that satellite) over a period of 1 second. In this case, the standard deviation of the simulated error distribution on the measured Doppler-shifted frequency is set to 50 Hz. Based on such measurements, it is possible to estimate both the maximum elevation angle θ^* of the satellite's trajectory and



(a) Error less than 50 km. (b) Error less than 75 km. (c) Error less than 100 km.

Fig. 8: Number of estimations.

the time t_0 when the first NPSS was received. The estimated value t_0 is used by the UE together with SIB information to compute the coordinates of the satellite's projection on the Earth when it is the closest to the UE. As it can be seen in Fig. 6, the actual position of the satellite is indicated in green, while the computed values (based on measurements affected by random errors) are pictured in orange. They are aligned along the actual trajectory of the satellite, whose parameters are known through the received SIB. It can be noticed that a UE cannot determine on which side of the satellite it is located since there are exactly 2 trajectories with the same orientation within the UE's communication range and the same maximum elevation angle. For this reason, estimation results are grouped around 2 locations, the actual one and the mirrored version, with respect to the satellite track. In that, the measurement errors affecting the estimation of the NPSS Doppler-shifted frequencies translate into an estimation error on θ^* and on t_0, and thus on the estimated satellite position and on the UE possible locations. Interestingly, the UE location estimations are also aligned. Such an alignment can be explained by the fact that if t_0 is estimated to be earlier than in reality, the UE will compute the position of the satellite projection when closest to the UE to be further north (in the configured scenario) and θ^* to be higher so that the final UE positions will be thought to be closer to the trajectory than in reality. Contrariwise, if t_0 is estimated to be later than in reality, the UE will compute the position of the satellite to be further south and θ^* to be smaller so that the final UE positions will be thought to be further from the trajectory than in reality.

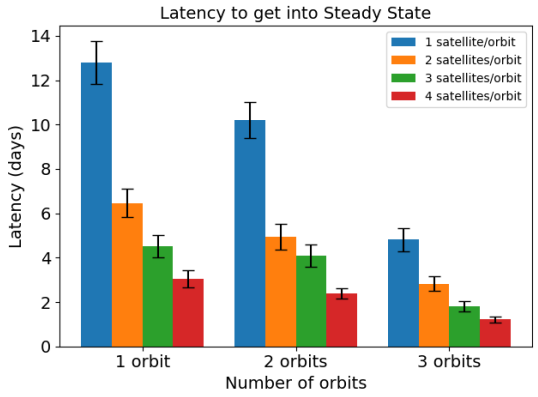
Pushing further the analysis on the same virtual environment with the orbital plane rotating synchronously with the Earth, Fig. 7 plots the position error when the standard deviation of the measurement error on the Doppler-shifted frequency is varied and for 3 different amounts of measured NPSS frequencies (i.e., 20, 50, and 100) used to estimate the maximum elevation angle and, as a consequence, the position of the UE. Each point on the plot results from the average of 8000 random measurements made by a UE placed at the same location of Fig. 6. Clearly, a higher standard deviation for the simulated measurement error translates into a bigger positioning error. As well, using more NPSS measurements narrows down the positioning error.

Being the goal of this subsection to study how the number N_e of estimation runs needed to have the UE position converge to a stable value, Fig. 8 plots N_e as a function of the standard deviation of the measurement error. 95% confidence intervals are also shown for the sake of statistical significance. It can be seen that to keep the positioning error lower than 50 km (see Fig. 8a) around 32 estimation runs have to be performed if the measurement error is particularly intensive (300 Hz of standard deviation) and the UEs cannot treat more than 20 frequency measurements simultaneously. In this worst case scenario, a UE must wake-up 32 times to finally have a small error on its positioning. As it can be expected, a smaller standard deviation for the measurement error, e.g., 100 Hz, permits the convergence to a positioning error lower than 50 km after $N_e = 26$ runs. Using more NPSS signals to trigger the estimation process, e.g., 100 signals, significantly reduces the number of estimation runs to values going from 6 in the best scenario without measurement errors to 28 when the measurement error is very intensive. It must be noticed that in the considered simulation scenario with the measurement error being null, the UE still needs a number of estimation runs bigger than 2 to converge to a stable positioning value, because with the routine presented in Sec. III-C the UE does not make any assumption on the value of the measurement error, so the lack of measurement error does not anticipate the stopping condition. It is worth mentioning that the positioning precision can be traded off in favor of a faster convergence of the routine, as shown in Fig. 8b and 8c, showing the same analysis when the acceptable errors are increased to respectively 75 and 100 km.

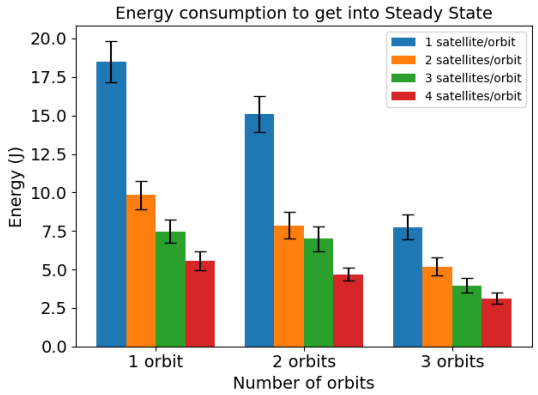
Finally, note that these results should be implemented in devices before the first wake-up. Once the number of estimations reaches this value, the device can switch to the *Steady State* phase: a UE will wake up based on the application's requirements rather than waking up every time a satellite passes in its transmission range.

C. Energy consumption and latency to get into Steady State

In the previous subsection, the optimal number of estimations required to reach the steady state was discussed. However, due to variations in satellite constellations, even if the same number of estimation runs is required, the latency and energy consumed can differ significantly. So, it is worth considering the impact of different constellations on the overall performance. As shown in Fig. 9, the worst-case performance scenario was studied by considering the reception of only 20 NPSS signals and a standard deviation of 300 Hz in Doppler shift measurements. Each result presented in a scenario is



(a) Latency



(b) Energy consumption

Fig. 9: Latency and energy consumption for different scenarios.

the average of 800 simulations. In every simulation, a device is randomly selected, and position estimations are performed until the estimated position error is reduced to less than 50 km. The energy consumption and latency to reach the steady state show a similar trend as the number of satellites per orbit increases. For instance, the energy consumption can reach around 18 J and the latency around 13 days in the worst-case scenario of 1 satellite per orbit. As expected, with an increasing number of satellites per orbit, both the energy consumption and latency decrease substantially. In the biggest constellation considered, i.e., 4 satellites per orbit and a total of 3 orbits, energy consumption drops to around 3 J, and latency decreases to about one day.

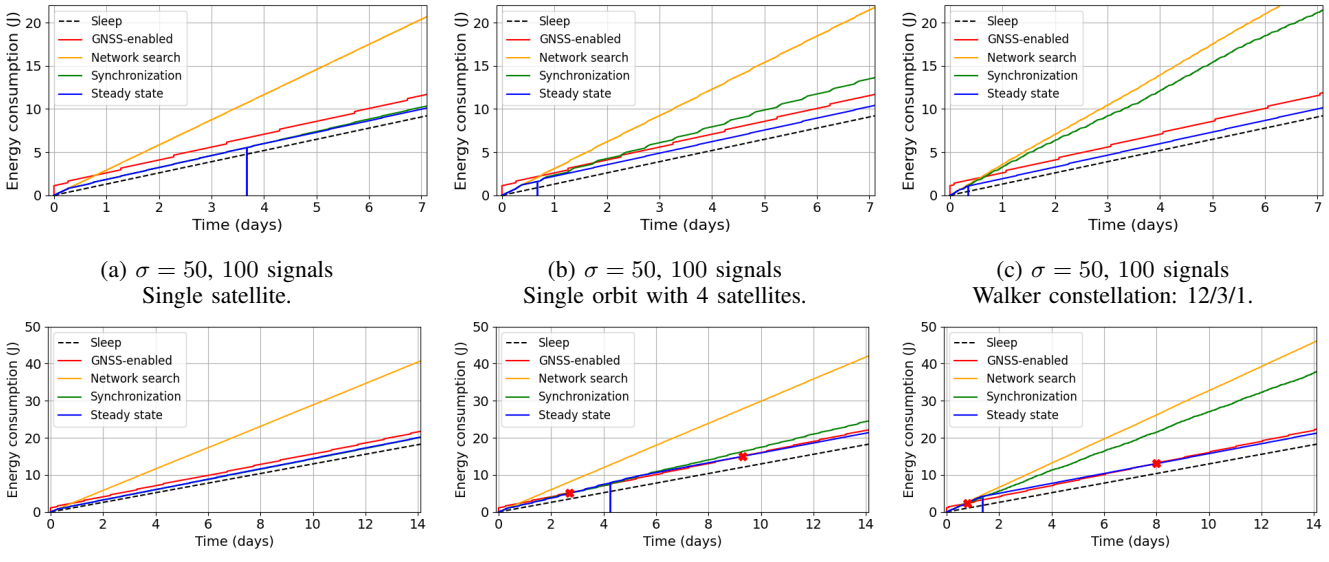
Interestingly, the performance in terms of energy and latency does not depend only on the total number of satellites but also on how they are spread across orbits. Sometimes, having more satellites in fewer orbits can lead to better results. For example, putting 4 satellites in 1 orbit might be more efficient than spreading them across 2 or 3 orbits because satellites in fewer orbits pass over the same area more often, which helps with faster synchronization and saves energy during wake-up. However, this behavior changes when there are more satellites. If there are 6 satellites, a $3x2$ setup (3 orbits with 2 satellites each) outperform a $2x3$ setup (2 orbits with 3 satellites each) in terms of both energy consumption and latency. The $3x2$ configuration provides a fairer distribution of satellite passes, making synchronization easier and keeping the energy consumption low. So, the best setup depends on finding the right balance between how many satellites are in each orbit and how many orbits are used.

In conclusion, it is essential to consider the number of satellites and their distribution across orbits in real deployments. While fewer orbits with more satellites can improve performance in some cases by reducing energy and synchronization time, this advantage may diminish as the satellite count increases. A balance between satellite density per orbit and the number of orbits is essential to ensure optimal coverage and system efficiency.

D. Energy consumption on the long term

This subsection analyzes the energy consumption of the proposed strategy and compares it with the currently mandatory GNSS-helped 3GPP solution. To show the importance of the combination of the 3 phases of the proposed strategy, the comparison will show what would happen if a subset of the 3 phases is implemented. Specifically, the *Network Search* phase implemented alone configures UEs to wake up at fixed intervals to search for NPSS signals. Even after the end of the communication, intermittent wake-ups continue to wait for the next satellite passes. Then, if the *Network Search* phase is followed only by the *Synchronization to LEO satellites*, a UE intermittently wakes up, estimates 2 potential positions, thereby building a list for the next passes and entering sleep mode to wait. In this configuration, a UE must wake up at every satellite's pass. For the sake of energy comparison, a simulation also shows what energy is consumed if the UE does nothing.

Furthermore, the comparison is shown for different satellite constellation scenarios, as plotted in Fig. 10. Given that a single 600 km satellite typically passes over a given point on Earth 1-2 times per day, the data transmission frequency is set to once per day to fit all types of constellations. Then, the *Network Search* wake-up interval $\tilde{t}$ is set to 150 seconds (see Sec. III-A, and Sec. IV-A). Instead, $\tilde{t}$ is set to 30 seconds when a UE wakes up during the *Synchronization to LEO satellites* phase for a predicted satellite pass (see Sec. III-B4). In detail, Fig. 10a, Fig. 10b, and Fig. 10c show the results for simulations lasting 7 days and related to the case of the standard deviation for the measurement error being equal to 50 Hz. The number of NPSS signals used for the estimation is 100 and they are uniformly captured in a window long 1 second. The maximum tolerated positioning error is fixed at 75 km. By using Fig. 8b, the configured N_e value is 7. Instead, Fig. 10d, Fig. 10e, and Fig. 10f show the worst-case scenario related to the standard deviation of the measurement error being equal to 300 Hz and the possibility to capture only 20 signals to estimate the position. As shown in Fig. 8a, a UE must perform $N_e = 32$ estimation run to determine its position with a maximum tolerated error of 50 km. In these cases, the simulations represent the energy consumption in the first 14 days.



(a) $\sigma = 50$, 100 signals
Single satellite.

(b) $\sigma = 50$, 100 signals
Single orbit with 4 satellites.

(c) $\sigma = 50$, 100 signals
Walker constellation: 12/3/1.

(d) $\sigma = 300$, 20 signals
Single satellite.

(e) $\sigma = 300$, 20 signals
Single orbit with 4 satellites.

(f) $\sigma = 300$, 20 signals
Walker constellation: 12/3/1.

Fig. 10: Energy consumption

First of all, Fig. 10a refers to the single satellite scenario and it clearly shows that if a UE only employs *Network Search* phase, its energy consumption is significantly higher than any other solution. This is mainly due to the continuous wake-up strategy aimed at constantly searching NPSS signals. However, all the other GNSS-free solutions are more energy efficient than the GNSS-based ones. This is attributed to the reasons previously explained in Sec. **??**: the GNSS cold start process demands the highest energy consumption before starting the very first communication. Note that in the single satellite scenario, the process presented in Sec. III-C to stop the position estimation process does not improve the energy-saving strategy already enforced by the *Synchronization to LEO satellites* phase. Interestingly, the energy consumption behavior changes when more satellites are coordinated in a LEO constellation, as shown in Fig. 10b (single orbit scenario) and Fig. 10c (Walker constellation). It can be clearly seen that the energy consumption of UEs not implementing the stopping condition for getting into the *Steady State* phase exceeds that of the GNSS-based solution. This is due to the device needing to wake up at all satellite passes in order to continuously update the list of next passes. As the number of satellites increases, the frequency of satellite passes can reach dozens of times per day, thus implying considerable energy waste. Additionally, since each position estimation results in 2 potential locations, the device also wastes energy waking up when no satellite is passing. In any case, the GNSS-free wake-up strategy presented so far definitely outperforms the GNSS-based solution in terms of energy efficiency. It is worth pointing out that as the number of satellites increases, UEs can enter the *Steady State* phase more quickly since the number of needed passes to narrow down the estimation error will be reached sooner in a denser constellation.

When considering the worst-case scenario (highest standard deviation for the measurement error, smallest required positioning error, smallest number of NPSS signals used for estimation), the energy consumption in the single satellite scenario of Fig. 10d does not significantly differ from the corresponding scenario of Fig. 10a: as the satellite only passes 1-2 times per day over a given spot, there is no significant change in the energy consumption. However, in the case of one single orbit and the Walker constellation (Fig. 10e and 10f), due to an increased number N_e of estimation runs, the energy consumption is higher in the short term. As it may be noticed in Fig. 10e, the GNSS-based solution becomes more energy efficient than the complete GNSS-free wake-up strategy proposed within this paper after 2.5 days. However, after 9.5 days of activity, the GNSS-free solution becomes again and stays even more energy-efficient than the GNSS-based. This is also visible for the Walker constellation of Fig. 10f: on the short term (in the first half day) and on the long term (after 8 days) the GNSS-free solution is the most power efficient.

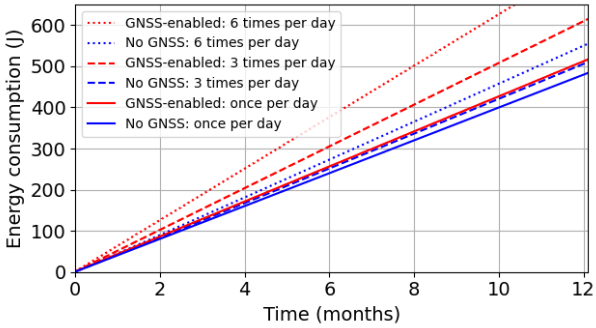


Fig. 11: Comparison of energy consumption during one year.

The final comparison of Fig. 11 refers to the employment of a Walker constellation composed by 16 satellites allowing

different data transmission frequencies up to 6 per day. The comparison is done between the complete GNSS-free wake-up strategy introduced in this paper (in red) and the GNSS-based solution (in blue). From Fig. 11, it is clear that as the number of data transmissions per day increases, the difference in energy consumption becomes increasingly significant in favor of the GNSS-free solution introduced so far.

V. Conclusion

In this paper, a GNSS-free wake-up strategy for NB-IoT Direct-to-Satellite architecture involving LEO constellation of any size has been proposed, and compared against the mandated GNSS-based 3GPP solution. Focusing on discontinuous coverage of LEO satellite constellations, this strategy allows NB-IoT UE to estimate their own position and predict future satellite passes, all without GNSS capabilities. Further than the implicit advantage of making DtS NB-IoT architecture non-dependent on GNSS and increasing the network resiliency, with the help of simulation-based analysis, it has been verified that the introduced strategy can effectively reduce energy consumption due to network synchronization issues. In addition, this strategy does not require any hardware update to the network devices, yet it just requires a firmware update on ground UEs. This makes it an ideal option for applications that do not require precise positioning. The paper uses the Walker constellation as an example, but this strategy can be extended to other constellations to address a wider range of scenarios by adding additional information in SIB. In future work, the uplink transmission capabilities will be analyzed, and more efficient position estimation strategies will be proposed. Additionally, the proposed methods will be validated using real experiments to ensure practical applicability and performance.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, pp. 1–7, 2019.

[3] J.-P. Bardyn, T. Melly, O. Seller, and N. Sornin, "Iot: The era of lpwan is starting now," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, 2016, pp. 25–30.

[4] E. M. Migabo, K. D. Djouani, and A. M. Kurien, "The narrowband internet of things (nb-iot) resources management performance state of art, challenges, and opportunities," *IEEE Access*, vol. 8, pp. 97 658–97 675, 2020.

[5] "3GPP Specifications, Available online: https://3gpp.org/."

[6] M. R. Palattella and N. Accettura, "Enabling Internet of Everything Everywhere: LPWAN with Satellite Backhaul," in *Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–5.

[7] J. A. Fraire, S. C. Umaña, and N. Accettura, "Direct-to-satellite iot - a survey of the state of the art and future research perspectives - backhauling the iot through leo satellites," in *Proc. of AdHoc-Now*, 2019.

[8] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A survey on technologies, standards and open challenges in satellite iot," *IEEE Commun. Surveys Tuts.*, pp. 1693–1720, 2021.

[9] A. H. Poghosyan and A. Golkar, "Cubesat evolution: Analyzing cubesat capabilities for conducting science missions," *Progress in Aerospace Sciences*, vol. 88, pp. 59–83, 2017.

[10] M. Yaqoob, A. Lashab, J. C. Vasquez, J. M. Guerrero, M. E. Orchard, and A. D. Bintoudi, "A comprehensive review on small satellite microgrids," *IEEE Transactions on Power Electronics*, vol. 37, no. 10, pp. 12 741–12 762, 2022.

[11] V. Mannoni, V. Berg, S. Cazalens, and P. Raveneau, "System level evaluation for nb-iot satellite communications," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–6.

[12] 3GPP, "Tr 36.763, study on narrow-band internet of things (nb-iot) / enhanced machine type communication (emtc) support for non-terrestrial networks (ntn) (release 17, v17.0.0)," 3GPP, Tech. Rep., June 2021.

[13] S. Jin, Q. Wang, and G. Dardanelli, "A review on multi-gnss for earth observation and emerging applications," *Remote Sensing*, vol. 14, pp. 1–24, 08 2022.

[14] A. Guidotti, A. Vanelli-Coralli, M. Caus, J. Bas, G. Colavolpe, T. Foggi, S. Cioni, A. Modenini, and D. Tarchi, "Satellite-enabled lte systems in leo constellations," in *2017 IEEE ICC Workshops*, 2017, pp. 876–881.

[15] G. M. Capez, S. Henn, J. A. Fraire, and R. Garello, "Sparse satellite constellation design for global and regional direct-to-satellite iot services," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 5, pp. 3786–3801, 2022.

[16] Z. Zhou, M. Afhamisis, M. R. Palattella, N. Accettura, and P. Berthou, "Pervasive LPWAN connectivity through LEO Satellites: trading off reliability, throughput, latency, and energy efficiency," in *Low-Power Wide-Area Networks: Opportunities, Challenges, Risks and Threats*, I. Butun and I. F. Akyildiz, Eds. Springer, 2022.

[17] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband leo satellite communications: Architectures and key technologies," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 55–61, 2019.

[18] D. Tuzi, E. F. Aguilar, T. Delamotte, G. Karabulut-Kurt, and A. Knopp, "Distributed approach to satellite direct-to-cell connectivity in 6g non-terrestrial networks," *IEEE Wireless Communications*, vol. 30, no. 6, pp. 28–34, 2023.

[19] Z. Zhou, N. Accettura, R. Prévost, and P. Berthou, "Lightweight synchronization to nb-iot enabled leo satellites through doppler prediction," in *Proc. of WiMob*, 2023, pp. 218–223.

[20] M.-G. Kim and H.-S. Jo, "Performance analysis of nb-iot uplink in low earth orbit non-terrestrial networks," *Sensors (Basel, Switzerland)*, vol. 22, 2022.

[21] G. Sciddurlo, A. Petrosino, M. Quadrini, C. Roseti, D. Striccoli, F. Zampognaro, M. Luglio, S. Perticaroli, A. Mosca, F. Lombardi, I. Micheli, A. Ornatelli, V. Schena, A. D. Mezza, A. Mattioni, D. Morbidelli, G. Boggia, and G. Piro, "Looking at nb-iot over leo satellite systems: Design and evaluation of a service-oriented solution," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 952–14 964, 2022.

[22] T. Hong, X. Yu, Z. Liu, X. Ding, and G. Zhang, "Narrowband internet of things via low earth orbit satellite networks: An efficient coverage enhancement mechanism based on stochastic geometry approach," *Sensors*, vol. 24, no. 6, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/6/2004

[23] C. Amatetti, M. Alsenwi, H. Chougrani, A. Vanelli-Coralli, and M. R. Palattella, "A novel twofold approach to enhance nb-iot mac procedure in ntn," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1453–1464, 2024.

[24] O. Kodheli, S. Andrenacci, N. Maturo, S. Chatzinotas, and F. Zimmer, "Resource allocation approach for differential doppler reduction in nb-iot over leo satellite," in *2018 9th Adv. Satel. Multimedia Syst. Conf. and the 15th Sig. Processing for Space Comm. Workshop (ASMS/SPSC)*, 2018, pp. 1–8.

[25] Z. Zhang, D. Wang, L. Liu, B. Wang, and C. Sun, "A frequency offset independent timing synchronization method for 5g integrated leo satellite communication system," in *2022 IEEE 22nd Int. Conf. on Comm. Tech. (ICCT)*, 2022, pp. 423–427.

[26] W. Wang, Y. Tong, L. Li, A.-A. Lu, L. You, and X. Gao, "Near optimal timing and frequency offset estimation for 5g integrated leo satellite communication system," *IEEE Access*, vol. 7, pp. 113 298–113 310, 2019.

[27] I. Ali, N. Al-Dhahir, and J. Hershey, "Doppler characterization for leo satellites," *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 309–313, 1998.

[28] J. Morales, J. J. Khalife, U. S. Cruz, and Z. Kassas, "Orbit modeling for simultaneous tracking and navigation using leo satellite signals," *Proc. of the 32nd Int. Tech. Meeting of the Sat. Division of The Inst. of Navig. (ION GNSS+ 2019)*, 2019.

[29] T. Janssen, A. Koppert, R. Berkvens, and M. Weyn, "A survey on iot positioning leveraging lpwan, gnss, and leo-pnt," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 135–11 159, 2023.

[30] I. S. Mohamad Hashim and A. Al-Hourani, "Satellite based localization of iot devices using joint doppler and angle-of-arrival estimation," *Remote Sensing*, vol. 15, no. 23, 2023.

[31] X. Luo, M. Wang, G. Dai, and X. Chen, "A novel technique to compute the revisit time of satellites and its application in remote sensing satellite

optimization design," *Int. J. of Aerospace Engineering*, vol. 2017, pp. 1–9, 01 2017.

[32] M. Jafari-Nadoushan and N. Assadian, "Repeat ground track orbit design with desired revisit time and optimal tilt," *Aerospace Science and Technology*, vol. 40, pp. 200–208, 2015.

[33] N. H. Crisp, S. Livadiotti, and P. C. E. Roberts, "A semi-analytical method for calculating revisit time for satellite constellations with discontinuous coverage," *ArXiv*, vol. abs/1807.02021, 2018.

[34] C.-J. Wang, "Structural properties of a low earth orbit satellite constellation - the walker delta network," in *Proceedings of MILCOM '93 - IEEE Military Communications Conference*, vol. 3, 1993, pp. 968–972 vol.3.

[35] P. Andres-Maldonado, M. Lauridsen, P. Ameigeiras, and J. M. Lopez-Soler, "Analytical modeling and experimental validation of nb-iot device energy consumption," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5691–5701, 2019.

[36] V. Vomhoff, S. Raffeck, S. Gebert, S. Geissler, and T. Hossfeld, "Nb-iot vs. lte-m: Measurement study of the energy consumption of lpwan technologies," in *2023 IEEE ICC Workshops*, 2023, pp. 403–408.

[37] A. Sørensen, H. Wang, M. J. Remy, N. Kjettrup, R. B. Sørensen, J. J. Nielsen, P. Popovski, and G. C. Madueño, "Modeling and experimental validation for battery lifetime estimation in nb-iot and lte-m," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9804–9819, 2022.

[38] M. Kanj, V. Savaux, and M. L. Guen, "A tutorial on nb-iot physical layer design," *IEEE Commun. Surveys Tuts.*, no. 4, 2020.

[39] J. Lygouras, "Memory reduction in look-up tables for fast symmetric function generators," *Instrumentation and Measurement, IEEE Transactions on*, vol. 48, pp. 1254 – 1258, 01 2000.

[40] 3GPP, "Ts 36.331, evolved universal terrestrial radio access (e-utra); radio resource control (rrc); protocol specification (release 17, v17.5.0)," 3GPP, Tech. Rep., June 2023.

[41] P. Zong and S. Kohani, "Optimal satellite leo constellation design based on global coverage in one revisit time," *International Journal of Aerospace Engineering*, vol. 2019, pp. 1–12, 2019.

[42] E. Hayes, "Computation of average revisit time for earth-observing satellites," in *26th Aerospace Sciences Meeting*, 1988.

[43] B. Rhodes, "Skyfield: High precision research-grade positions for planets and earth satellites generator," 2023.

[44] M. Paonni, M. Anghileri, S. Wallner, J.-A. Avila-Rodriguez, and B. Eissfeller, "Performance assessment of gnss signals in terms of time to first fix for cold, warm and hot start," *Institute of Navigation - International Technical Meeting 2010, ITM 2010*, vol. 2, pp. 1221–1236, 01 2010.