75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.

IAC-24,D1,IP,37,x82998

An autonomous distributed timing signal in-space as alternative to GNSS time synchronisation

Agathe BOUIS [a]*, Ruaridh A. CLARK [a], Malcolm MACDONALD [a]

[a] *Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street Glasgow, G1 1XW*, agathe.bouis@strath.ac.uk
* Corresponding Author

Abstract

Accurate synchronised time is an essential requirement for many space-based services. Currently, the primary source of synchronised time comes from Global Navigation Satellite Systems (GNSS) in conjunction with on-board clocks. The vulnerability of GNSS centralised system and the lack of reliability of small on-board clocks have made the need for distributed robust alternatives a pressing concern. A dynamic protocol is presented herein enabling efficient time synchronisation to be reached in satellite systems even in the presence of unknown disruptors or attackers. This approach requires neither an external terrestrial reference source, nor any knowledge of the satellite network topology.
Keywords: GNNS, Time synchronization, Distributed Consensus, Multi-agent systems, Satellite Networks

Acronyms/Abbreviations

1-hop Neighbours' time (N-time)
2-hop Neighbours' time (M-time)
AUDITS (AUtonomous DIstributed Timing-signal in Space)
Automatic Dependent Surveillance-Broadcast (ADS-B)
Automatic Identification System (AIS)
Consensus Calculation Function (CCF)
Consensus Cycle (CC)
Consensus time (C-time)
CSTDMA (Carrier Sense TDMA)
Data Querying Function (DQF)
Denial of Service (DoS)
Distributed Network Time Synchronisation (DNTS)
Global Navigation Satellite Systems (GNSS)
Gradient Time Synchronization Protocol (GTSP)
Hardware time (H-time)
Inter Satellite Link (ISL)
LEO (Low Earth Orbit)
Matryoshka Orbital Network (MatrON)
MEO (Medium Earth Orbit)
Network Time Protocol (NTP)
Opinion Dynamics DIsruption-tolerant Consensus (ODDI-C)
Position, Navigation, and Timing (PNT)
Reachback Firefly Algorithm (RFA)
Reference Broadcast Synchronization (RBS)
SOTDMA (Self-Organized TDMA)
Time Division Multiple Access (TDMA)
Timing-sync Protocol for Sensor Networks (TPSN)
Virtual time (V-time)
Wireless Sensor Networks (WSN)

1. Introduction

Daily life is underpinned by signals from satellites. In particular, accurate synchronised time is an important requirement for many ground- and space-based services. Precise timing signals are necessary for activities such as telecommunications, formation flying, timestamping of scientific data, and satellite positioning [1]. Currently, the primary source of this time and position data comes from Global Navigation Satellite Systems (GNSS). However, with this increasing reliance on GNSS comes an increased vulnerability to single points of failures [2]. As demonstrated in the past few years, ground attacks such as spoofing, jamming, or Denial of Service (DoS) have the potential to create GNSS-denied environments [1]. This has in turn led to the investigation and establishment of a number of Resilience Frameworks looking to enhance the reliability of GNSS within the economy (see the UK government's policy framework for greater Position, Navigation, and Timing (PNT) Resilience for example) [3, 4]. As space becomes increasingly packed, reliance on GNSS as a mechanism of time synchronization and position has also extended to orbit for LEO (Low Earth Orbit) and MEO (Medium Earth Orbit) satellites. Thus, as we observe an acceleration in the number of satellites being deployed into orbit, depending solely on GNSS means opening the door to potentially major disruptions, an ill-advised gambit.

Traditional time synchronisation approaches are typically centralised or semi-distributed [5]. This includes methods such as the Network Time Protocol (NTP) which requires layers of servers broadcasting their time to their children. Similarly, methods such as the Reference Broadcast Synchronization (RBS) or Timing-sync Protocol for Sensor Networks (TPSN) see nodes elect as leader in a local cluster or across a spanning tree with other nodes synchronising with either their cluster leader or their parent in the tree via two-way messages. While more resilient than purely centralised methods, such techniques are still highly vulnerable to attacks on elected leaders and typical spoofing attempts.

The emerging field of Distributed Network Time Synchronisation (DNTS) aims to offer new solutions to the problem of distributed clock time synchronisation [6].

DNTS methods look to bring independent network nodes to a synchronised shared time without the need for a central authority or any master-slave commands. A typical approach to distributed time synchronisation is consensus-based methods by which agents in a network or dynamical system reach an agreement on a shared state (i.e. time) by way of information exchange between agents and their neighbours [7]. Other distributed approaches have looked towards control-based algorithms including PID [8], and Kalman filter designs [9]. However, it is important to note that many of these approaches are not resilient to malicious or faulty behaviour, which makes them unsuitable for implementation in the trust-less environment of space.

DNTS and consensus problems take root in the field of distributed computing and brings together decades of research on Wireless Sensor Networks (WSNs), internet of things, and multi-robot/multi-agent systems [1, 10, 11]. This research is highly applicable to satellite time synchronisation considering the challenges faced by the latter. As such, this work takes inspiration and extends work from both fields into a new contribution described in this brief. Satellite networks, much like WSNs, deal with resource constrained nodes as power on-board satellite for inter-satellite links and data processing is limited [12]. They also act over dynamic communication topologies as satellites move with respect to each other, disconnect, and reconnect with one another to create dynamic time-shifting networks [13]. Satellite network also need to cope with potentially unreliable communications, where delays (sending time, transmission time, and reception time) and asymmetric links are expected [13, 14]. Key to time synchronisation algorithms, clock stability (due to phase noise, thermal noise, jitter, aging, etc) must be considered. Additionally, for both WSNs and satellite networks, issues of robustness and security must be accounted for such that the network has the ability to cope with, and/or recover from disruptions. Such malfunctions may come from system faults or security threats [15, 16].

Completely distributed alternatives to time synchronisation have also been proposed, with the challenge of making these alternatives lightweight and efficient being a key focus. This includes the *Reachback Firefly Algorithm* (RFA) which takes inspiration from firefly synchronisation to synchronise nodes [17]. The algorithm sees nodes broadcast a synchronisation message and in turn shift their phase towards the pulses they have received. This protocol does not compensate for drift, but instead requires for frequent broadcasts and updates to ensure that nodes keep synchronisation. *Gradient Time Synchronization Protocol* (GTSP) offers a similarly representative look at the potential of distributed time synchronisation [18]. Quoted across a range of studies, GTSP is listed across many surveys and is typically used as a benchmark for comparison of other protocols [19].

While the use of GNSS for time synchronisation is a potential source of vulnerability, inspiration can also be found in GNSS derived systems. The approach presented in this brief is notably informed by the Automatic Identification System (AIS) and the Automatic Dependent Surveillance-Broadcast (ADS-B) technology regarding their approach to data transceiving (receiving and broadcasting).

The AIS is a world-wide automatic positioning and tracking system based on fitting on-board transponders to vessels to allow them to continuously broadcast their position and heading [20]. The system alerts other ships and shore stations also fitted with AIS transceivers of the presence of the vessel. Initially developed for short ranges, AIS gained its global reach through it use of low orbit satellites fitted with AIS transceivers. AIS is broadcasted using Time Division Multiple Access (TDMA) Schemes, a channel access method for shared-medium networks. Such schemes allow several users to share the same frequency channel by dividing the signal into time slots where users transmit one after the other, each in their own allocated time slot. In typical TDMA systems, a controlling entity is used to allocate transmission slots. As AIS must operate far offshore on ships at sea, AIS transceivers independently determine their own TDMA allocation and critically, ensure that slots already used by other vessels are avoided. AIS deals with this problem through the implementation of a hierarchy TDMA schemes. Highest priority transceivers, Class A, use SOTDMA (Self-Organized TDMA) technology to continuously maintain an updated slot map in memory, pre-announcing their transmission to reserve their transmit slot. Lower priority transponders, Class B, use CSTDMA (Carrier Sense TDMA) technology which checks for Class A transmissions before sending their own signals. Class B accomplish this by monitoring background noise levels If the signal strength at the start of a slot is significantly above the background level, the slot is assumed to be in use and the Class B transmission is deferred.

The ADS-B is an air surveillance system which currently acts as an enhancement to traditional air traffic control. ADS-B enables aircraft to broadcast their real-time position, altitude, speed, and other telemetry information to other aircraft and ground stations. This information is then received by ground stations and other aircraft to allow the receiver to build a shared situational awareness. ADS-B performance can be enhanced through the use of satellite-based ADS-B receivers. Unlike ground-based radar tower, satellite provide a larger coverage over the globe regardless of terrain.

The framework presented herein, AUtonomous DIstributed Timing-signal in Space (AUDITS), takes inspiration from DNTS, consensus algorithm, and

GNSS-derived systems. However, compared with the aforementioned protocols, AUDITS offers several advantages. It compensates both skew and offset allowing for an efficient consensus to be reached in both rate and time. Additionally, it is resilient to disruptions, be it faults or malicious attacks, with the protocol dynamically filtering in-coming communications to ignore extreme disruptive values. The protocol requires little overhead with the filtering being performed using median and mean based calculations. AUDITS is fully distributed, requiring no information regarding network topology nor the number and behaviour of disruptive agents.

2. System Model

The satellite time synchronisation model simulates satellite networks orbiting the Earth, broadcasting their on-board time to their neighbours, and reaching time synchronisation by way of a virtual time system. Time synchronisation is defined as a state where all the satellites in the network come to an agreement, within a given bounds, on a shared coordinated time; meaning that each satellite sees the same passage of time as the other satellites in the distributed system. This system is intended to cope with the challenges of communication time delays, communication protocol, and relativity effects.

The protocol presented herein assumes the following: let $G(t) = (E, \varepsilon, A)$ be a time varying digraph with a finite number of nodes (agents) $E = 1, 2, \dots, N$, a set of directed edges $\varepsilon \subseteq E \times E$, and an adjacency matrix $A = a_{ij} \in R\ (N \times N)$. Each directed edge $(j, i) \in \varepsilon$ represents a directional link between the node pair (j, i), such that communication between the nodes is enabled. In satellite networks, links are created when a satellite can broadcast data to a neighbouring satellite. This occurs when this second satellite enters the Inter-Satellite Links (ISLs) range of satellite wanting to broadcast. As ISL ranges can vary from satellite to satellite, links can be asymmetric or on-way only. Links are considered to be representative of information flow with each node communicating with their neighbours given the opportunity. Considering the asymmetry inherent in this model, a difference is made between a node's in-neighbours $E_{in}^{i} = \{j \in E : (j, i) \in \varepsilon\}$ representing its incoming communications, and its out-neighbours $E_{out}^{i} = \{j \in E : (j, i) \in \varepsilon\}$, to which it sends information. 2-hop in-neighbours are the in-neighbour of a node i's in-neighbour, defined as $E_{in}^{ij} = \{l \in E : (l, j) \in \varepsilon \text{ and } j \in E_{in}^{i}\}$ and written as a set as $E_{in}^{iE_{in}^{i}}$. A distinction is made between 1- and 2-hop neighbours' data as 2-hop data is further removed in the past (1 transmission early) and must therefore be propagated forward to be of use. This multi-hop model is shown in Figure 1.

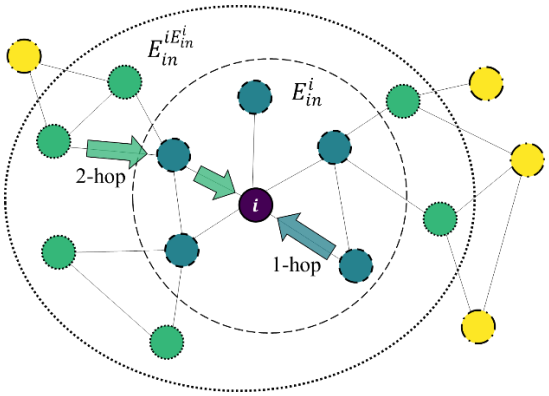

Figure 1: Multi-hop neighbourhood connectivity.

2.1 Clock Model

To allow for the time synchronisation to take place, the following times are defined:

- **H-time/Hardware time (*H*):** user satellite-specific time indicated by the satellite's on-board clock (caesium, quartz…), see Eq. 1. Subject to phase deviation, clock frequency deviation, and frequency drift/aging [1].
- **N-time/Neighbour times (*N*):** time received by the user satellites from its 1-hop in-neighbours. These times are the virtual times of the in-neighbours at broadcast. No on-board related delay is assumed.
- **M-time/Neighbour times (*M*):** time received by the user satellites from its 2-hop in-neighbours via its in-neighbours. These V-times are propagated by the user satellite's in-neighbours prior to their broadcasting, bringing them to the same theoretical time step as the N-times. The use of 2-hop data serves to artificially increase network connectivity, enabling a faster convergence speed and preventing synchronisation.
- **C-time/Consensus time (*C*):** user satellite-specific time calculated from the N- and M-times received from the satellite's in-neighbours (see Figure 1). At each consensus step, the satellite filters the N-times it has received from its in-neighbours using a consensus protocol.
- **V-time/Virtual time (*V*):** user satellite-specific time updated as part of the consensus process. This time is taken from the satellite's virtual clock (see Eq. 2), whose frequency is the hardware clock's ticking frequency corrected at each time step to allow the virtual clock's time to converge onto the consensus time. The satellite's V-time and C-time are different if the V-time deviates from the

consensus of its neighbours. This deviation informs the control input to change V-time.

Time synchronisation occurs when V-time and C-time match, within a given bounds. This bound depends on the application of the time-synchronisation algorithm.

Additionally, the following term is specified: **Consensus Cycle (CC):** a cycle composed of a given number of hardware clock time step $\Delta H^S(k)$, where S corresponds to a given satellite S. The cycle comprises of a *data querying function*, where the data used for the update process is extracted from the on-board data base, an *update phase* where the virtual clock's rate is updated, and a *downtime/waiting time* before the start of the next cycle.

A key aspect of time synchronisation is the modelling of the relationship between the H- and V-times of nodes. Every node i in the network has its own H-time whose first order dynamics are given by Eq. 1. This hardware time output (H) is the result of the physical clock's oscillator value (ω), its rate (r_{hw}), and offset (o_{hw}) from the reference time. The virtual clock time (V) meanwhile takes as input the hardware time (H) and adjusts it with a virtual rate (r) and offset (o), see Eq. 2. Note that the hardware parameters are unknown to the node; with only their resulting output, H, being available. The parameters r_{hw} and o_{hw} are non-deterministically time varying due to ambient conditions and aging. However, the resulting noisiness in the hardware output signal occurs at a slower frequency than that of the consensus protocol, and thus all hardware is considered as constant throughout this paper. The consensus protocol modifies the virtual rate and offset to ensure agreement of virtual times across nodes without changing the physical oscillator values – an undertaking which would reduce the life span of the oscillator.

$$H = \mathrm{r_{hw}} \cdot \omega + \mathrm{o_{hw}} \quad \text{Eq. 1}$$

$$V = r \cdot H + o \quad \text{Eq. 2}$$

2.2 Time Synchronisation

The high-level block diagram of the AUDITS time synchronisation architecture is shown in Figure 2. Consensus on a shared distributed V-time is achieved by satellites using ISLs to communicate time information. Satellites broadcast their current virtual time to their out-neighbours, following a given Broadcasting Scheme and in turn receive their in-neighbour's virtual times, accepting data following a matching Receiving Scheme. Received time values are filtered by way of a Consensus Protocol to ensure resilient convergence, even in the presence of disruptive agents. The filtered values are then used in combination with the user satellite's current CC's V-time as inputs to a Virtual Clock Update Scheme. This scheme takes the difference between the C-time of the filtered values and the satellite's current V-time to update the user satellite's virtual rate.

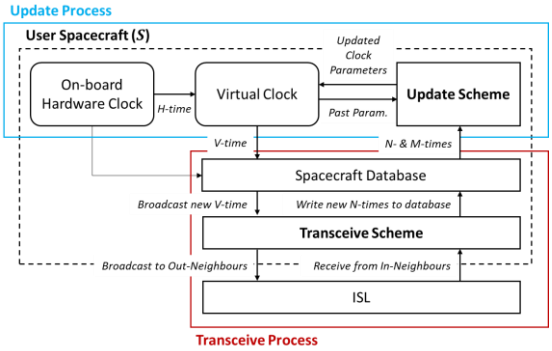

Figure 2: AUDITS Software Architecture describing the relations between satellite software elements for a satellite S.

The process of time synchronisation is composed of two independent sub-processes (see Figure 2): The **Transceive Process** manages how the user satellite processes the data it has received and writes it in the on-board database, and how the satellite extracts its newest V-time from the database before broadcasting it to its out neighbours. The second subprocess is the **Update Process**, during which the user satellite's uses time data received from its in-neighbours to update its virtual clock rate.

The **Transceive Process**, which is performed independently from the update process, covers both data reception from in-neighbours and broadcasting to out-neighbours. Reception of data, and its subsequent committing to the spacecraft database, is performed continuously. Broadcasting however is performed in a discrete fashion dependent on contact plan and TDMA access opportunities. Broadcasting acts at a higher frequency than that of Update Process, which are executed at a slow on-board rate f^S_{LOW} to manage computational time as well as to leave sufficient time for the user satellite to receive enough in-neighbour values from which to calculate C-time. All satellites will update their virtual time at this frequency. However, it is important to note that as this frequency depends on the hardware clock, which is subject to noise and drift, the exact frequency of update will vary from satellite to satellite. In contrast to the slow update rate, broadcast can be performed at a higher frequency being only limited by the user satellite's broadcasting capacity and the broadcasting scheme.

The broadcasting scheme takes inspiration from AIS CSTDMA [20] with satellites listening to ambient noise levels before deciding whether to broadcast to a specific neighbour. As a result, broadcast may be irregular. In this sense, the scheme follows a "lucky transmit" approach where nodes broadcast at irregular intervals in the hope of being heard. The high broadcast frequency f^S_{HIGH}

(irregular as it may be) specifies how often the current clock virtual time $V^S(k)$ and propagated in-neighbours' data $N^S(k)$ is broadcasted to a satellite's out-neighbours E^S_{out}. The broadcasted message contains the information shown in Table 1.

Table 1: Broadcasted Information

Data	Description
SatID	The broadcasting satellite's identification (ID).
V-time/ $V^S(k)$	The broadcasting satellite's virtual time at the time of broadcast. Received as N-time.
In-neighbour SatID	The identification of the broadcasting satellite's in-neighbours whose data the broadcasting satellite is forwarding.
In-neighbour V-time/ $N^S(k)$	The propagated V-time of the broadcasting satellite's in-neighbours whose data the broadcasting satellite is forwarding.

The **Update Process** is itself composed of two functions, whose interplay is shown in Figure 3. These functions are the Data Querying Function (DQF) and the Consensus Calculation Function (CCF).

Data Querying Function (DQF): the user satellite S collects V-time data from its in-neighbours E^S_{in} and from its 2-hop neighbours $E^{SE^S_{in}}_{in}$ (see Figure 1). These data sets are shown in Table 2:

Table 2: Data set notation

Origin	Full notation	Simplified notation
1-hop	$V^{E^S_{in}}(k)$	$N^S(k)$
2-hop	$V^{E^{SE^S_{in}}_{in}}(k)$	$M^S(k)$

The 1- and 2-hop data sets are written in the satellite's database as soon as they are decoded – assuming no delay in the internal processing. Note here that the 2-hop data has been propagated forward in time to time step k. M-times are received by in-neighbours E^S_{in} at time step $k-1$. They are used by the in-neighbours during their update before being propagated in time to the next time step k. M-times are then broadcasted to the user satellite along with the in-neighbours own V-time. A given satellite filters through the received values to only account for the latest information received by a node, such that if data is received from a node both through the 1- and 2-hops paths, the newest 1-hop data will overwrite the older data received from the same satellite in the database. Additionally, not all of this data will be used for the update process as older data loses its pertinence and reliability. The period of data considered, measured in hardware time steps $\Delta H^S(k)$, can span from a single CC to longer periods depending on the frequency at which the data is received (Figure 4). Note that the V-time continues to progress over the course of the CC, such that the consensus is not limited by the duration of the CC. Relevant datasets of 1- and 2-hop time-information is queried using the DQF at the start of each CC to feed to the CCF (see Figure 3).

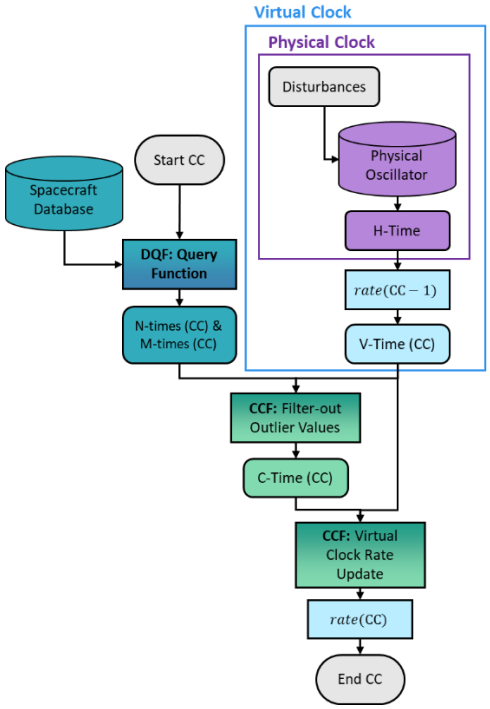

Figure 3: Update Process Control Loop

Consensus Calculation Function (CCF): at the start of each CC, the user satellite filters the forwarded by the DQF ($\{N^S(k) \cup M^S(k)\}$) to remove outliers or extreme values present in the dataset. This filtering is based on the latest CC's V-time, $V^S(k)$. From these filtered N-values, this CC's C-time, $C^S(k)$, is calculated. The difference between the C-time and current V-time is used by the Virtual Clock Update Scheme (see Figure 3) to update the satellite's virtual parameters, namely its virtual rate, $rate^S(k)$.This new rate (and associated V-time) are written to the database following the update. From the end of this update phase, V-time is calculated from the updated V-parameters.

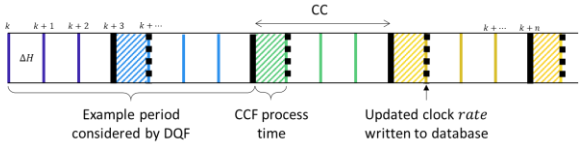

Figure 4: Update Process mechanism relative to Hardware Clock Ticking

3. Update Process

At the start of each CC, the user satellite processes the data it has queried from its database to update its virtual clock parameters. This is accomplished by first filtering the values received to ensure that disruptive values, whether they be nodes newly joined to the system, nodes whose clock drift has suddenly changed or are faulty, or malicious nodes are not considered during the update. In this study, the Opinion Dynamics DIsruption-tolerant Consensus (ODDI-C) Protocol [21], is used to filter the 1- and 2-hop in-neighbours' V-times, $\{N^S(k) \cup M^S(k)\}$.The protocol, which takes inspiration from social dynamics, creating a dynamic filtering threshold for each node. Values exceeding this threshold are removed from the set considered from the consensus update. The threshold is based on the node's own value with respect to the distribution of values the protocol is used on. The more extreme a node is, the higher its threshold. On the other hand, nodes close to the distribution's median will apply small thresholds, thus filtering out values that would drive the node away from consensus. ODDI-C's dynamic approach allows distributed systems to deal with unknown disruptions, without knowledge of the network topology or the numbers and behaviours of the disruptors.

Filtering of data is performed on a data set corresponding to in-neighbours' data and 2-hop neighbours' data. Multi-hop information is used to create virtual links between nodes and their 2-hop neighbours. Most satellite networks are only sparsely connected and this poor connectivity can result in the formation of small clusters of satellites, where a cluster is defined as a subset of the network where nodes have higher connectivity with members of their cluster than those outside of it. This can in turn result in pockets of synchronisations as nodes filter out all data coming from outside their respective clusters. By collecting 2-hop data, the virtual topology's connectivity is increased, convergence speed is increased, and synchronisation clustering can be prevented while allowing for a robust consensus.

ODDI-C's filtering of outliers is performed on datasets composed of V-times; their in-neighbours V-times $N^S(k)$, their 2-hop neighbour's V-times $M^S(k)$, and the user satellite's own V-time $V^S(k)$. ODDI-C reduces the set $\{N^S(k) \cup M^S(k)\}$ to $F^S(k)$ (see [21] for protocol details). Network nodes with few neighbours (1- or 2-hop) have limited datasets on which to perform their filtering. As such, when $|\{N^S(k) \cup M^S(k)\}| < 10$, other robust filtering methods such as Dixon's Q test or Grubbs' test must be used to filter out outliers. This is required as ODDI-C makes use of the Median Absolute Deviation, a calculation requiring larger sample sizes and thus large datasets which is not the case for Dixon's or Grubbs' tests.

Depending on the user satellite's V-time, two approaches will be used to calculate the C-time. If the node's own V-time is considered to be an outlier the C-time will be calculated purely from the N- and M-times stored on-board. The node's V-time is classified as outlier by the filtering method. For ODDI-C, the node is an outlier if its z-score exceeds 3. For Dixon's test, it is if the node is out of range with the other nodes. Other filtering method will have different outlier classification approaches.

$$C^S(k) = \text{mean}\big(F^S(k)\big). \quad \text{Eq. 3}$$

Otherwise, if the node's V-time is not flagged an outlier, the C-time will include the node's own value

$$C^S(k) = \text{mean}(\{F^S(k) \cup V^S(k)\}). \quad \text{Eq. 4}$$

Having calculated this CC's C-time, the node then uses this information in conjunction with the previous CC's C-time $C(k-1)$ to update its virtual clock parameters (rate and offset). This is accomplished by defining two errors, an error in time

$$\text{err.t}^S(k) = C^S(k) - \text{V}^S(k), \quad \text{Eq. 5}$$

and an error in rate

$$\text{err.r}^S(k) = C_r^S(k) - r^S(k), \quad \text{Eq. 6}$$

where the r^S is the user satellite's virtual clock rate (see Eq. 2) and C_{rate}^S is consensus rate which is calculated as

$$C_{r-new}^S(k) = \frac{C^S(k) - \text{C}^S(k-1)}{\Delta H^S(k)}, \quad \text{Eq. 7}$$

$$C_r^S(k) = \alpha \cdot C_{r-new}^S(k) + (1-\alpha) \cdot C_r^S(k-1), \quad \text{Eq. 8}$$

where ΔH^S is the elapsed user satellite's hardware time and α is a smoothing factor ($0 \leq \alpha \leq 1$). This smoothing factor is used to smooth the consensus rate from one time-step to the next and avoid rapid changes that might occur if, for example, the node is new to the consensus or has been disconnected from the system. The extent to which older values are considered depends on the value of α, with a value of $\alpha = 1$ indicating that only new values are considered, and a value of $\alpha = 0$ indicating that only past values are used. Here, the smoothing factor is here set to 0.8 such that the error in change in rate will principally depend on the new error time, smoothed by the past rate to avoid sudden drastic changes.

The errors in rate and time are combined into a singular error

$$\Delta r^S(k) = \mu(k) \cdot (\eta \cdot \text{err.r}^S(k) + \gamma \cdot \text{err.t}^S(k)), \quad \text{Eq. 9}$$

where η and γ are tuning rates, and μ is a damping factor calculated as

$$\mu(k) = \frac{1}{\text{loneliness}^S(k)}, \quad \text{Eq. 10}$$

where $\text{loneliness}^S(k)$ is a parameter corresponding to the number of CCs during which a node has not received any information from its in-neighbours. The μ

factor slows the update the longer information has not been received. When the node gets reconnected, the loneliness factor is divided by 2 for each CC it is connected again until it reaches 1, corresponding to a μ of 1 for which there is no damping.

The two factors η and γ are tuning rates used to adjust the extent to which the virtual clock parameters are changed. These rates are usually set between 0 and 1 [22], and together, tune the impact of the node's error in time and rate on the update, as well as which error is corrected in priority. Tuning rates are also used to trade-off between the speed of convergence (η and γ close to 1) and the noise, error, and over-shooting immunity (η and γ close to 0). In this paper, empirically derived values of $\eta = 0.8$ and $\gamma = 0.2$ are applied. The virtual clock parameter update corresponds to

$$r^S(k+1) = r^S(k) + \Delta r^S(k), \quad \text{Eq. 11}$$

The user satellite's V-time than updates as

$$V^S(k+1) = V^S(k) + r^S(k+1) \cdot \Delta H^S. \quad \text{Eq. 12}$$

Note that all errors (rate and time) are corrected by adapting the virtual clock's rate. No changes are made to the virtual clock offset.

Three important cases must be noted. The first is when the node is completely disconnected. In this case, all errors are set to 0 and the node performs no update. Second is the case when a node is completely disconnected and has just reconnected now. During this first CC of reconnection, the node has no past C-time to rely on from which to calculate C_r^S. As such, the node only uses the $\mathrm{err.t}$ to update its offset and does not change its rate. Finally, is the case where the node filters out all the values it was performing its consensus protocol on, meaning $F^S(k) = \emptyset$. This implies that the node's V-time is the closest to the median out of its in-neighbours' times, that is, it stands on the consensus. The C-time is therefore set as

$$C^S(k) = \mathrm{V^S}(k). \quad \text{Eq. 13}$$

While this results in $\mathrm{err.t^S}(k) = \Delta o^S(k) = 0$, the error in rate can be calculated as normal, taking the difference between the consensus rate and the node's virtual rate.

Additionally, a safeguard is put in place to prevent the virtual clock from going back in time, for which

$$V^S(k+1) < V^S(k). \quad \text{Eq. 14}$$

This safeguard is implemented via a minimum threshold thr, set to 0.01 in this paper, such that

$$r^S(k+1) = \max\big(\mathrm{thr}, r^S(k+1)\big). \quad \text{Eq. 15}$$

4. Methodology

Two satellite networks are considered for the analysis of AUDITS. Given the dynamic nature of satellite networks, both scenarios investigated are time-varying graphs. The first looks at a network of 500 satellites created within the Matryoshka Orbital Network (MatrON) framework [23], a computationally efficient abstraction of the orbital environment used to model constellations of satellites and their interactions. The satellites are randomly distributed across 10 evenly spaced shells with altitudes spanning 160 to 1000 km, inclination spanning 0- to 180-degree, and eccentricities of 0. This first satellite network is tested with 20 disruptive nodes. On average, nodes see an in-degree of 98. These disruptive nodes are composed of nodes with known rate and offset errors (which are not corrected) and of nodes with oscillating V-times. Considering the high connectivity seen across the network, filtering of outlier values is performed using ODDI-C.

The second network is a Walker Delta constellation consisting of 200 nodes distributed across 2 planes at 90-degrees to each other, with 56-degree inclinations, and eccentricities of 0.01. The rings have an average connectivity of 10. This network is created using the MATLAB satellite communications toolbox. used rather than ODDI-C to account for the limited connectivity, and the specificities of this network: chiefly, the nodes see limited connectivity (10) for most of the simulation with rare in-degree spikes as nodes enter the hubs which exist at the crossing of the two rings. For such ring-type networks with poor connectivity, ODDI-C's tendency to over-filter incoming data can result in divergence and the creation of clusters of synchronisation. As such, a less zealous filtering method is employed with Dixon's test being selected.

Another key aspect of the second simulation is the type of network being synchronised. Ring structures, or circulant graphs as they often known [24], are notoriously complex to synchronise. Studies looking at Kuramoto oscillators have notably demonstrated that high connectivity is normally required to ensure perfect synchrony [25], and to avoid chimera states which emerge due to transmission delays and the presence of cycles in the network [26]. This simulation is thus used to showcase the ability of AUDITS to enable consensus to be reached on a circulant graph - something not yet achieved to the best of the authors' knowledge. Further work looking a circulant graphs coping with disruptor nodes is nonetheless required as current tests show that ring-networks can still easily be disrupted despite the implementation of AUDITS.

Across both scenarios, the time resolution, that is the time between the graph's temporal sequences, is set to 30 seconds. Edges are created when the spatial distance between satellite nodes drops below 4000 km. This distance corresponds to a low-end estimate of the current ISL capabilities [27].

To account for the impact of the initial conditions, results are shown across 20 Monte Carlo simulations. Each simulation is initialised with different initial V-

times and clock parameters. By testing the same networks multiple time, the reliability of the results is confirmed. The termination criterion of both tests is a maximum time $t_{max} = 200$ timesteps, with the simulations stopping once this maximum time is reached.

Across the two scenarios, node parameters and initial virtual times are initialised with values drawn from a normal distribution with mean μ and standard deviation σ, and described as

$$\sim \mathcal{N}(\mu, \sigma^2). \quad \text{Eq. 16}$$

Initial virtual times are drawn from the normal distribution using a $\mu = 2$ and $\sigma = 0.5$. The hardware parameters, meaning the physical clock's oscillator value (ω), its rate (r_{hw}), and offset (o_{hw}), are all initialised from a distribution with parameters $\mu = 1$ and $\sigma = 0.2$. All values (initial V-times or hardware parameters) are constrained to be $\mathbb{R}^+$. The virtual rates (r) are set to 1 and the virtual offset (o) to 0 (the virtual offset will remain unchanged through the simulation, with only the virtual rate being modified).

For these preliminary simulations, a synchronous message transmission scheme is implemented. While nodes' H-times (and H-parameters) are different, a single CC is considered by the simulation, with all satellites broadcasting their times and updating their virtual rate in a synchronous manner. As a result of this synchronous approach, time delays between satellite are not considered.

Throughout the two tests, nodes keep the same identity. Disruptive nodes do not recover and stay disruptive for the entire run. Two convergence metrics (CM) are calculated for each test from the following set of formulae

$$CM(k) = \frac{TD(k)}{TD(0)}, \quad \text{Eq. 17}$$

$$TD(k) = \Sigma_{m=1}^{E}\ \Sigma_{n=1}^{E}\ (|x_m(k) - x_n(k)|), \quad \text{Eq. 18}$$

where x_m and x_n are stand-ins for a value of interest x associated with nodes $m \in \{1, E\}$ and $n \in \{1, E\}$ respectively.

As the convergence metric is a relative value dependent on the initial difference between nodes' values ($TD(0)$), it can be problematic to understand on its own, even when contextualised by the plot of nodes' opinion trajectories. To avoid confusion, an error threshold (err) is applied. This threshold is used to create a minimum floor value ($Floor$) for the convergence metric. Results smaller than this floor are levelled to the floor value. This adaptation of the convergence metric prevents the development of small rounding errors while giving context to the results.

$$Floor = \frac{err}{\text{TD (0)}}, \quad \text{Eq. 19}$$

$$CM = \max(Floor, CM) \quad \text{Eq. 20}$$

the error threshold (err) is set at a value of 10^{-9}, a unit less value representing the maximum average error across all nodes.

The first convergence metric looks at the convergence of the nodes' V-time for which x is replaced by V. The second convergence metric assesses the relative difference between nodes' absolute rate. The absolute rate, which replaces x in Eq. 20Eq. 20, is calculated for each satellite S as

$$abs.r^S(k) = r^S(k) \cdot \mathrm{r_{hw}} \cdot \omega. \quad \text{Eq. 21}$$

5. Results

The results of the randomly distributed 500-satellite network simulation are shown in Figure 5. As seen across tiles (b and c), with AUDITS implemented, time synchronisation is rapidly and effectively reached within approximately 140-time steps. Convergence in time and rate is observed to reach the convergence floor value of 10^{-9}. As noted by tile (a), disruptor nodes are effectively ignored, with the network reaching a consensus on a shared time of 200 at time step 200 with a rate best described by an angle of 45-degrees. The presence of the plateaus in both the rate and time convergence plots (tiles b and c) can be explained by the presence of steady state error in some simulations while the majority of simulations convergence to the floor value. Such errors are resolved as the network changes with different satellites connecting and successfully purging the error to allow for a shared time at the network level. The presence of the errors is to be investigated in future work.

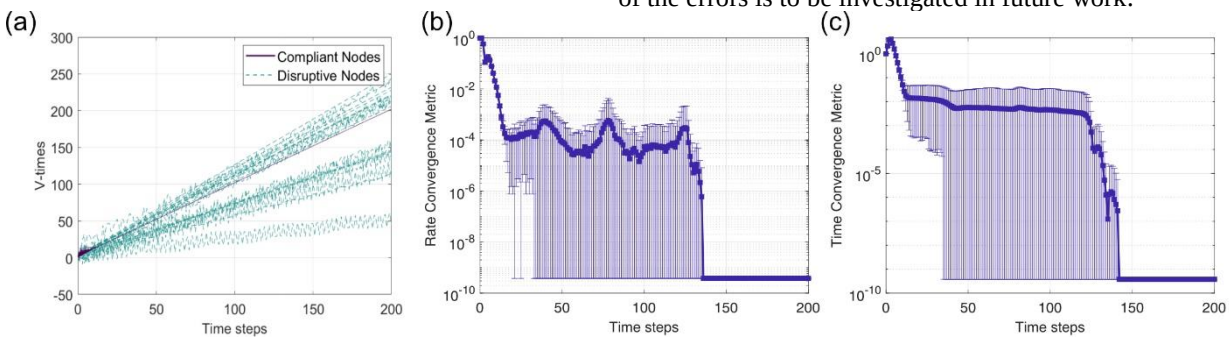

Figure 5: Convergence of a 500-node random satellite network. Tile (a) shows the evolution of the V-times of compliant and disruptive nodes for one Monte Carlo simulation. Tiles (b and c) show the Convergence metrics in Rate and Time respectively.

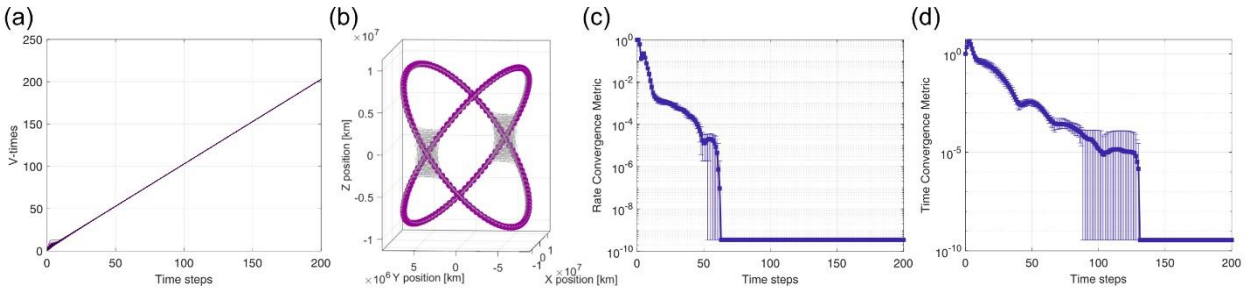

Figure 6: Convergence of a 200-nodes 2-plane Walker-Delta constellation. Tile (a) shows the evolution of the V-times of compliant nodes for one Monte Carlo simulation. Tile (b) shows a snapshot of the network diagram. Tiles (c and d) show the Convergence metrics in Rate and Time respectively.

The results of the 2-planes, 200-satellite Walker-Delta constellation is shown in Figure 6. The network structure, specifically the presence of hubs at the rings' crossing is shown in tile b. Rate and time convergence are noted to both reach their floor value of 10^{-9} promptly. Interestingly, time convergence is slower than rate convergence, with the presence of a small plateau being noted much like for the 500-node network despite the lack of disruptive agents. The difference between the convergence rates of tiles c and d can be explained by the empirically derived tuning rate of Eq. 9 which favours reducing the rate (0.8) compared to the time error (0.2). Further investigation of the plateau is nonetheless required.

6. Conclusion

The AUtonomous DIstributed Timing signal in-Space (AUDITS) framework can be used engineer the disruption-tolerant time-synchronisation of satellite networks. By incorporating a resilient consensus protocol such as the Opinion Dynamics-inspired Disruption- tolerant Consensus (ODDI-C) algorithm or Dixon's Q-test, and increasing network connectivity through the use of virtual links, AUDITS allows for time-varying satellite networks to come to an agreement on a shared rate and time down to a relative error between nodes of 10^{-9}. In satellite networks with sufficient connectivity, AUDITS effectively mitigate the impact of disruptor nodes despite having no knowledge of the network topology or the numbers and behaviours of the disruptor agents. AUDITS also enables for convergence to be reached across ring (circulant) networks such as Walker-Delta constellations. The impact of disruptor nodes on such structure, as well as the presence of temporary steady state errors in the convergence of certain simulations is to be further explored.

References

[1] M.V. Shenoy, K. Anupama, Swarm-Sync: A distributed global time synchronization framework for swarm robotic systems, Pervasive and Mobile Computing, 44 (2018) 1-30.
[2] R.T. Ioannides, T. Pany, G. Gibbons, Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques, Proceedings of the IEEE, 104 (2016) 1174-1194.
[3] L. Economics, The economic impact on the UK of a disruption to GNSS, London Econ., London, UK, Tech. Rep., Jun, DOI (2017).
[4] A. Proctor, R. Andy, A structured approach to achieving system resilience for Position Navigation and Timing (PNT) Systems (PNT System Resilience), DOI.
[5] F. Shi, X. Tuo, L. Ran, Z. Ren, S.X. Yang, Fast convergence time synchronization in wireless sensor networks based on average consensus, IEEE Transactions on Industrial Informatics, 16 (2019) 1120-1129.
[6] I.E.L. Hulede, H.M. Kwon, Distributed network time synchronization: Social learning versus consensus, IEEE Transactions on Signal and Information Processing over Networks, 7 (2021) 660-675.
[7] L.-A. Phan, T. Kim, Fast consensus-based time synchronization protocol using virtual topology for wireless sensor networks, IEEE Internet of Things Journal, 8 (2020) 7485-7496.
[8] K.S. Yıldırım, R. Carli, L. Schenato, Adaptive control-based clock synchronization in wireless sensor networks, 2015 European Control Conference (ECC), IEEE, 2015, pp. 2806-2811.
[9] B.R. Hamilton, X. Ma, Q. Zhao, J. Xu, ACES: Adaptive clock estimation and synchronization using Kalman filtering, Proceedings of the 14th ACM international conference on Mobile computing and networking, 2008, pp. 152-162.
[10] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proceedings of the IEEE, 95 (2007) 215-233.
[11] L.-A. Phan, T. Kim, T. Kim, Robust neighbor-aware time synchronization protocol for wireless sensor network in dynamic and hostile environments, IEEE Internet of Things Journal, 8 (2020) 1934-1945.
[12] J. Elson, K. Römer, Wireless sensor networks: A new regime for time synchronization, ACM SIGCOMM Computer Communication Review, 33 (2003) 149-154.
[13] S.M. Lasassmeh, J.M. Conrad, Time synchronization in wireless sensor networks: A survey, Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), IEEE, 2010, pp. 242-245.
[14] J.A. Fraire, J.M. Finochietto, S.C. Burleigh, Delay Tolerant Setellite Networks, Artech House2017.
[15] S.K. Jha, N. Panigrahi, A. Gupta, Security threats for time synchronization protocols in the internet of things, Principles of Internet of Things (IoT) ecosystem: insight paradigm, Springer2020, pp. 495-517.

[16] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, X.-M. Zhang, A survey on security control and attack detection for industrial cyber-physical systems, Neurocomputing, 275 (2018) 1674-1683.
[17] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, R. Nagpal, Firefly-inspired sensor network synchronicity with realistic radio effects, Proceedings of the 3rd international conference on Embedded networked sensor systems, 2005, pp. 142-153.
[18] P. Sommer, R. Wattenhofer, Gradient clock synchronization in wireless sensor networks, 2009 International Conference on Information Processing in Sensor Networks, IEEE, 2009, pp. 37-48.
[19] L.-A. Phan, T. Kim, T. Kim, J. Lee, J.-H. Ham, Performance analysis of time synchronization protocols in wireless sensor networks, Sensors, 19 (2019) 3020.
[20] M. Insights, The Definitive AIS Handbook, 2016.
[21] A. Bouis, C. Lowe, R.A. Clark, M. Macdonald, Engineering consensus in static networks with unknown disruptors, arXiv preprint arXiv:2403.05272, DOI (2024).
[22] K.P. Murphy, Machine learning: a probabilistic perspective, MIT press2012.
[23] J. Gribben, R. Clark, C. Lowe, M. Macdonald, Matryoshka Orbital Networks, 74th International Astronautical Congress, 2023.
[24] P.T. Meijer, Connectivities and diameters of circulant graphs, DOI (1991).
[25] D.A. Wiley, S.H. Strogatz, M. Girvan, The size of the sync basin, Chaos: An Interdisciplinary Journal of Nonlinear Science, 16 (2006).
[26] D.M. Abrams, R. Mirollo, S.H. Strogatz, D.A. Wiley, Solvable model for chimera states of coupled oscillators, Physical review letters, 101 (2008) 084103.
[27] A.U. Chaudhry, G. Lamontagne, H. Yanikomeroglu, Laser intersatellite link range in free-space optical satellite networks: Impact on latency, IEEE Aerospace and Electronic Systems Magazine, 38 (2023) 4-13.

[1] See Milne, R. (28/06/24), Russian GPS jamming threatens air disaster, warn Baltic ministers, Financial Times. Available at: https://www.ft.com/content/37776b16-0b92-4a23-9f90-199d45d955c3, accessed 15/08/24