Optimizing Participant Selection for Fault-Tolerant Decision Making in Orbit Using Mixed Integer Linear Programming

Robert Cowlishaw, *Graduate Student Member, IEEE*, Annalisa Riccardi, and Ashwin Arulselvan

***Abstract*—In challenging environments such as space, where decisions made by a network of satellites can be prone to inaccuracies or biases, leveraging smarter systems for onboard data processing, decision making is becoming increasingly common. To ensure fault tolerance within the network, consensus mechanisms play a crucial role. However, in a dynamically changing network topology, achieving consensus among all satellites can become excessively time consuming. To address this issue, the practical Byzantine fault-tolerance algorithm is employed, utilizing satellite trajectories as input to determine the time required for achieving consensus across a subnetwork of satellites. To optimize the selection of subsets for consensus, a mixed integer linear programming approach is developed. This method is then applied to analyze the characteristics of optimal subsets using satellites from the International Charter: Space and Major Disasters (ICSMD) over a predefined maximum time horizon. Results indicate that consensus within these satellites can be reached in less than 3.3 h in half of cases studied. Two satellites that are within the maximum communication range at all times are oversubscribed for taking part in the subnetwork. A further analysis has been completed to analyze which are the best set of orbital parameters for taking part in a consensus network as part of the ICSMD.**

***Index Terms*—Consensus algorithm, fault-tolerant decision making, mixed integer linear programming (MILP), on-orbit decision making, practical Byzantine fault tolerance (pBFT), satellite communication.**

I. Introduction

As THE use of smarter technologies for onboard satellite data processing increases, decisions are beginning to be made automatically on a variety of matters. These can range from orbit maneuvers to the selection of Earth observation (EO) data to collect. Given the challenges of accurate data collection in space, along with the diversity of satellite operators and capabilities, these AI-driven decisions need to be verified or cross-checked in a secure and trustworthy manner.

Received 26 April 2024; revised 30 July 2024; accepted 27 August 2024. Date of publication 11 September 2024; date of current version 30 September 2024. *(Corresponding author: Robert Cowlishaw.)*

Robert Cowlishaw and Annalisa Riccardi are with the Mechanical and Aerospace Engineering Department, University of Strathclyde, G1 1XQ Glasgow, U.K. (e-mail: robert.cowlishaw.2017@uni.strath.ac.uk; annalisa.riccardi@strath.ac.uk).

Ashwin Arulselvan is with the Management Science Department, University of Strathclyde, G1 1XQ Glasgow, U.K. (e-mail: ashwin.arulselvan@strath.ac.uk).

Digital Object Identifier 10.1109/JSTARS.2024.3459630

Data or decisions produced can be inaccurate and biased, and can originate from obstructive or adversarial sources, whether intentionally or accidentally. When collaborative international initiatives, such as the International Charter: Space and Major Disasters (ICSMD) [1], are part of those decisions, neutrality and geopolitical bias must be weighed up as wrong or delayed decisions can reduce their value. In another instance, accuracy in collision detection messages is imperative for orbital monitoring in space environment management. These messages play a pivotal role in ensuring the safe operation of spacecraft, as the expense and necessity of maneuvering them underscore the importance of averting potentially disastrous collisions. Consequently, the reliability of decisions made in orbit transcends mere financial concerns, extending to the realms of operational safety and efficiency.

A method to achieve a more neutral and trustworthy architecture is to use a distributed network of satellites to make decisions, this can reduce operator bias via diverse perspectives, reduce inaccuracies in the data provided and potentially reduce delays in bottle necked communications. Through the use of consensus mechanisms between the communicating satellites there would be no reliance on ground-based infrastructure, a potential source of bias, errors, and delays in operations. The studies completed in [2] and [3] discuss current intersatellite communication projects but limited to a constellation of satellites with similar orbital parameters. Projects such as SpaceX's Starlink [4] produce a peer-to-peer network, however, diversity of neighbors remains low. Implementing intersatellite and inter-constellation networks will create a diverse network; however, the highly varied orbital arrangements will pose challenges in determining which participants should be included in the consensus mechanism.

With these differing orbital arrangements network topology and latency vastly change over time. One method for reducing this problem is by implementing signal routing. The authors in [5], [6], and [7] investigate optimal signal routing strategies within satellite networks, yet they reveal vulnerabilities when the network is subject to adversarial interference. Although consensus mechanisms may incur additional time and communication overhead, they offer heightened security and reliability for transmitted messages. Particularly in scenarios demanding reliable communication, such as those involving costly or high-risk decisions, leveraging consensus mechanisms can effectively mitigate obstructive behavior.

The authors in [8], [9], [10], and [11] examine networks characterized by fixed topology, changing topology, and communication time delays within a fixed topology context. However, they do not address scenarios involving switching topology and time delays. The work here presented diverges from the deterministic approach by proposing a methodology to identify consensus participants in highly combinatorial situations. Specifically, it explores the application of this method in quasi-stochastic networks, such as satellites orbiting in space, aiming to understand how consensus participants can be identified in such dynamic environments.

In another study, Coelho et al. [12] employ a mixed integer linear programming (MILP) model to assess consensus mechanisms, specifically Byzantine fault-tolerance (BFT) systems, under various adversarial attacks. This indicates a precedent for utilizing MILP in evaluating consensus mechanisms. This approach primarily focuses on identifying scenarios where consensus mechanisms fail. This contrasts with the emphasis of the work here presented, which aims to uncover scenarios demonstrating the success of consensus mechanisms. More specifically, contributions of this research include the following.

1) Demonstrating the use of a consensus mechanism in very high latency and ad hoc topology in-orbit rather than ground-based low latency fixed topology systems.
2) The MILP mathematical formulation for optimally selecting a subset of satellites for executing the BFT consensus mechanism in orbit.
3) The solution of the formulated MILP problem for the satellite-based emergency mapping scenario with the deterministic MILP solver.

The rest of this article is organized as follows. Section II presents and discusses the consensus mechanisms and underlying issues, such as the time required for completion. This issue is formalized in Section III as an MILP problem, aiming to select participants in the consensus process to minimize time delays caused by their orbital configurations. An example for the satellite-based emergency mapping satellite selection pool problem is then solved with the Gurobi solver [13] in Section IV. Finally, Section V concludes this article.

II. Consensus Algorithm

Consensus protocols have been used before in space missions as demonstrated in [14], specifically addressing the coordination and alignment of orbits for cluster missions. However, this particular form of consensus diverges in its focus; while it serves as a means for synchronization, its primary objective does not involve the mitigation of obstructive nodes within the network. Consensus mechanisms for the application described in this article exist and are used in many distributed networks on the ground. BFT algorithms [15], Paxos/Raft [16] / [17], and directed acyclic graphs (DAG) [18] are some examples. Some consensus mechanisms that are not BFT require trusted parties, and therefore, cannot achieve the goals set in this article. Therefore, in the scenario considered in this work is that of a heterogenous network of satellites that private and public operators can subscribe to and operate together with. However, BFT algorithms such as practical Byzantine fault-tolerance (pBFT) [15] algorithms can tolerate a proportion of nodes in the network being obstructive. Specifically, it can handle up to one-third of the nodes involved in consensus being obstructive. As the number of satellites in consensus increases, the number of obstructive nodes that pBFT can tolerate also increases. However, as the number of nodes participating in the consensus mechanism grows, there is a corresponding increase in communication demands, leading to higher communication overhead.

Wang et al. [19] demonstrate the feasibility of reducing the number of replicas (synonymous with nodes in consensus mechanisms) in BFT systems and organizing them into smaller groups without compromising security. This strategy not only mitigates communication overhead but also enhances decision-making efficiency. In contexts where communication latency during a single consensus round is notably high, diminishing the number of replicas significantly reduces the time required for achieving consensus. Leveraging the insights from [19], this approach could be applied to facilitate interconstellation decisions, with constellations serving as the subgroups. Furthermore, the concept of sharding, where networks are subdivided by replicas to complete numerous tasks more efficiently, as discussed in [20], presents a similar methodology that holds promise for achieving analogous outcomes.

In this work, the pBFT algorithm has been selected as the consensus mechanism for on-orbit study. The specifics of how this operates are discussed in the following section; see Section II-A.

A. pBFT Algorithm

The pBFT protocol, as referenced by [15], functions as a replication mechanism aimed at achieving consensus while exhibiting resilience against Byzantine faults. These faults manifest when a node participating in the decision-making process is uncooperative, whether intentionally or inadvertently, leading to potentially erroneous outcomes. Integration of a consensus algorithm such as pBFT enables the system to manage a certain degree of faulty replicas while guaranteeing liveliness and safety of the network, albeit at the expense of increased message transmission. Within the context of pBFT, the protocol can accommodate a specified number of faulty replicas (denoted as f) among the total number of replicas (r) involved in the process. The correlation between r and f in pBFT, ensuring fault tolerance, is expressed as

$$r = 3f + 1. \tag{1}$$

Therefore, if you required the system to tolerate a single faulty satellite you would require four satellites in the consensus round. In the opposite way, if you had 20 satellites in the consensus round, the consensus could tolerate up to six faulty satellites.

The pBFT algorithm has five phases; request, pre-prepare, prepare, commit, and reply. The actual complexity of the algorithm is reduced by only studying the time requirements and replicas participating are considered. The initial request is assumed to come from the primary node, and therefore, the

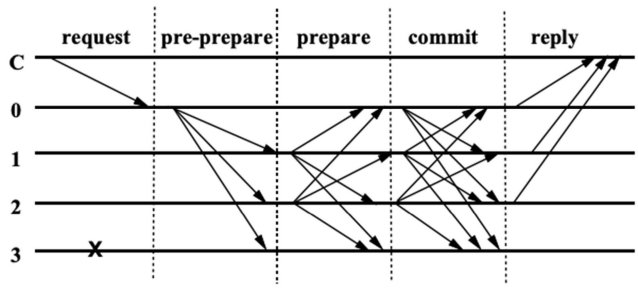

Fig. 1. pBFT algorithm scenario with four replicas, where replica 3 is faulty and C is client not part of the consensus mechanism [15].

request step is not required. The overall process can be seen in Fig. 1 and the reduced list of phases is described as follows.

1) *Pre-Prepare:* The primary replica sends the message to all other replicas.
2) *Prepare:* All replicas, not including the primary, send the message to all other replicas, including the primary.
3) *Commit:* All replicas, including the primary, send the message to all other replicas, including the primary.
4) *Reply:* All replicas, not including the primary, send the message back to the primary.

B. Problem

Due to the differing orbit configurations and pointing requirements, intersatellite communication is intermittent and differing in topology over time. Commercial intersatellite communication networks exist such as Viasat communication relay [21] however these reintroduce a centralized owner for the communication to pass through. Current intersatellite communication discussed in [2] ranges from a maximum of 5 to 1000 km. Currently Starlink makes up a large portion of all satellites in orbit and could, therefore, be considered as state of the art of satellite communication and their current application with the Federal Communications Commission states that the satellites involved will be in orbit at an altitude of 525, 530, and 535 km [22]. If these satellites and the distance they communicate to the ground were considered as the satellite communication distance, these distances would not be unreasonable assumption for intersatellite communication range. Therefore, generalizing across these intersatellite capabilities stated in [2], and [22], a value of 500 km is considered as the maximum communication range in the network for the simulations in this work. This distance is however a hyperparameter that can be adjusted, without compromising the solution process proposed here, if different requirements are set.

To minimize the time required to achieve consensus, a method to determine the best subset of satellites is designed. Formulating the problem as an MILP problem, it is solved with state-of-the-art deterministic strategies, to determine the optimal subsets for different time periods. Furthermore, the orbital parameters of the optimal subset of satellites are analyzed to identify which parameters and configurations influence their frequency of selection for consensus. As the consensus time is being minimized in all cases, the number of satellites in the consensus subset will be four, as this is the minimum number of replicas possible to have fault tolerance in pBFT. Adding more satellites will only increase consensus time. Thus, a constraint on the minimum number of satellites in the consensus network is added to the problem. Another assumption imposed on the problem is the size of the selection pool. As of [23], there are currently 9195 active satellites in orbit, resulting in 3×10^{415} possible combinations. To address this complexity, the size of the selection pool, denoted as n, needs to be reduced to a meaningful subset.

In order to establish a more pertinent selection pool, the ICSMD [1] serves as the basis. The charter identifies 82 active satellites designated for satellite-based emergency mapping purposes. With the implementation of a consensus mechanism, these satellites possess the capability to autonomously make decisions among themselves regarding the occurrence and location of disasters, as well as providing specific disaster-related information. The 82 satellites in the selection pool are further refined to simulate the computational limits of onboard edge computing capabilities. The computational capabilities considered are somewhere between the first launch in 2020 of AI compute capabilities from European Space Agency Phi-Sat 1 [24] and the next generation of satellite computing capabilities coming from Planet implementing NVIDIA Jetson Edge AI platforms [25] in a future launch. Therefore, the computational power used in this simulation is between 0.0144 TFLOPS from the Intel Myriad VPU used in Phi-Sat 1 [26] and NVIDIA's lowest electrical power Jetson Chips, which could be used by Planet in the future at 0.472 TFLOPS [27]. The 30 satellites selected account for the top 50% of all timesteps where interactions occur over a period of 1 day, therefore, representing a suitable pool to complete the pBFT mechanism in an acceptable time frame for disaster response. These 30 satellites can be seen in Fig. 2 from the set of all 82 ICSMD satellites. The ID for each of the 30 satellites is also given for future reference. The number of satellites in the selection pool is a hyperparameter that can scale with available on board computational power.

With these assumptions in place, the problem is formally defined next.

III. Problem Definition

First the number of satellites in the selection pool and minimum number of satellites in the consensus subset are defined. $n \in \mathbb{N}$ is the number of satellites in the selection pool, where the selection pool is all the satellites available to be part of consensus. This was decided to be 30 as discussed in Section II-B. $m \in \mathbb{N}$ is the number of satellites in the consensus subset, where the consensus subset is the satellites taking part in the consensus algorithm. This can be computed from (1) as a minimum of one faulty node is required for the pBFT algorithm to be useful. Therefore, four replicas are required, giving $m = 4$.

Next the indexes used throughout the definition are set. $S = \{1, \ldots, n\}$ is the set of indexes of satellites in the selection pool ($i, j \in S$ are the index of two satellites in the selection pool). These are shown in Fig. 2. $P = \{1, \ldots, 4\}$ is the set representing the consensus phases indexes ($p \in P$ is the index of a phase). The phases are in order, pre-prepare, prepare, commit,

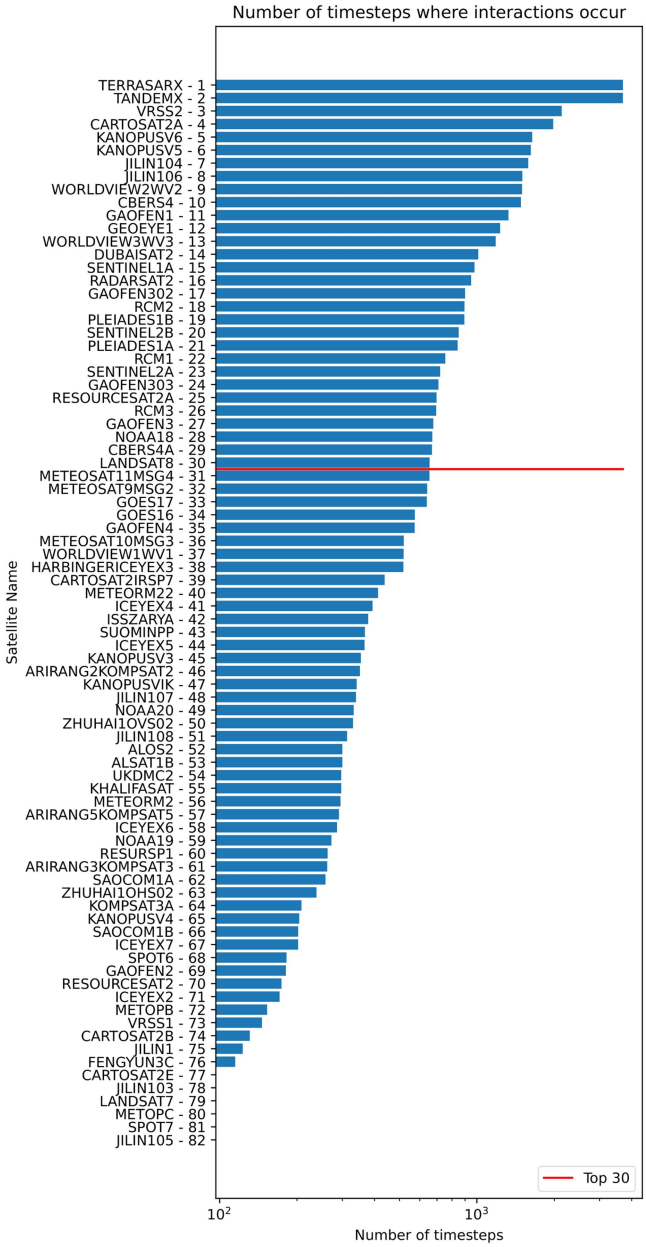

Fig. 2. Number of time steps where a satellite is within interaction range of another over 1 day. The red line shows the top 30 satellites chosen for the selection pool.

and reply, as discussed in the pBFT algorithm in Section II-A. $T = \{1, \ldots, t_{\max}\}$ is the set representing the discretized time steps, where $t_{\max}$ is the maximum number of time steps over which consensus can be computed.

A. Precomputed Data

Next, the precomputation of the satellite interactions is undertaken to reduce computational expense when solving the MILP problem. The satellite data are produced from PyEphem [28] and is stored in $\mathbf{C}$. $\mathbf{C} = [c_{i,t}]$ is the matrix storing the recomputed coordinates of the satellites positions across the whole time horizon. All satellites are compared against each other to

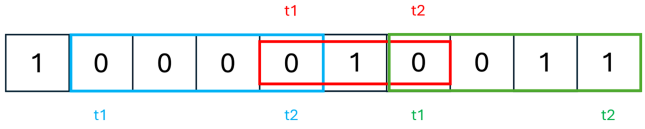

Fig. 3. Shown is $l_{i,j}$ for a given i, j satellite combination and the time steps where communication can occur. For the same i, j satellite combination in $\mathbf{T}$, for $t1, t2$ between the two time steps each color represents a different time window. Therefore, $\tau_{i,j,t1,t2}$ for blue is 0, for red is 1, and for green is also 1.

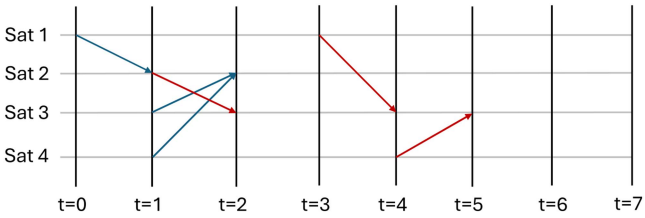

Fig. 4. Example phase of satellites communicating. Sat 2 (blue) completes after $t = 2$ and sat 3 (red) completes after $t = 5$.

produce $\mathbf{L}$. $\mathbf{L} = [l_{i,j,t}]$ is a binary matrix defined as follows:

$$l_{i,j,t} = \begin{cases} 1 & \text{if} \quad \|c_{i,t} - c_{j,t}\| <= 500\text{km} \\ 0 & \text{if} \quad \|c_{i,t} - c_{j,t}\| > 500\text{km}. \end{cases} \tag{2}$$

To finish precomputation, from this $\mathbf{L}$ all the ranges of time periods in all satellite combinations are computed to determine if an interaction occurs in these time ranges. $\mathbf{T} = [\tau_{i,j,t_1,t_2}]$ is a binary matrix of whether there is any 1 in a moving window of variable size across $\mathbf{L}$, where $t_1 < t_2$ and $t_1, t_2 \in \{0, \ldots, t_{\max}\}$. This is defined as follows and can be visualized in Fig. 3:

$$\tau_{i,j,t_1,t_2} = \begin{cases} 1, & \text{if } \sum_{t=t_1}^{t_2} l_{i,j,t} > 0 \\ 0, & \text{if } \sum_{t=t_1}^{t_2} l_{i,j,t} = 0. \end{cases} \tag{3}$$

B. Decision Variables

$x_{i,p,t}$ is a binary decision variable in $\mathbf{X}$ where sat i in phase p in time step t is 1 if it completes phase p and otherwise 0. $y_{i,p,t}$ is a binary decision variable in $\mathbf{Y}$ that follows a similar logic where sat i in phase p at time step t is the sum of all $x_{i,p,(0,t)}$ is 1, otherwise $y_{i,p,t}$ is 0. This $\mathbf{Y}$ signifies if a phase change for the specific satellite has occurred in the past. On top of these, two decision variables are defined to identify the satellites in the specific consensus round. $\mathbf{q} \in \{0,1\}^n$ is a binary vector of length n representing the selection of the primary satellite in the consensus network. $\mathbf{w} \in \{0,1\}^n$ is a binary vector of length n representing the subset of satellites selected for running the consensus mechanism (excluding the primary). To better describe the function of $\mathbf{X}$ and $\mathbf{Y}$, an example phase is shown in Fig. 4 with the corresponding matrices in (4). In the matrices in (4), the example is shown for a single phase where the columns IDs identify satellites and the rows IDs identify time steps. $\mathbf{X}$ represents when a satellite has received all communications for that phase to themselves. Therefore, in the example, sat 2 completes at $t = 2$ and sat 3 completes at $t = 5$ with a corresponding in the $\mathbf{X}$ matrix. Assuming sat 1 and sat 4 are not requiring communications in this example phase, they complete immediately at $t = 0$. If this assumption is not

true, this phase would be infeasible within this time period, and therefore, more time steps would be required. In $\mathbf{Y}$, all values are 1 after a satellite has completed the phase as shown in the following matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4}$$

C. Constraints for thepBFT Shape

Constraints are applied to $\mathbf{q}$ and $\mathbf{w}$ to make sure $\mathbf{q}$ defines the primary satellite and $\mathbf{w}$ defines all the satellites in the current consensus round excluding the primary. To do this, at least m satellites must be selected in the consensus network for pBFT, therefore as $\mathbf{w}$ excludes the primary, the sum must be more than or equal to 3 as shown in the following equation:

$$\sum_{i=1}^{n} w_i \geq 3. \tag{5}$$

Next, a constraint to make sure only one primary satellite is selected

$$\sum_{i=1}^{n} q_i = 1. \tag{6}$$

Finally, a constrain enforcing that the primary satellite index shall not be present in the consensus network vector, $\mathbf{w}$

$$q_i + w_i <= 1 \qquad \forall i \in S. \tag{7}$$

$\mathbf{X}$ and $\mathbf{Y}$ are made to be dependent on $\mathbf{q}$ and $\mathbf{w}$ to make sure the correct satellites/columns are being used in $\mathbf{X}$ and $\mathbf{Y}$. More specifically, for each satellite in the consensus subset and for each phase, the columns of matrix $\mathbf{X}$ should sum to 1, except for the last phase where only the column representing the primary satellite should contain a single 1. We use the notation $\mathbb{I}_A$ to denote an indicator variable that takes a value 1 if condition A is true and 0 otherwise.

$$\sum_{t=0}^{t_{max}} x_{i,p,t} = q_i + w_i \mathbb{I}_{p \neq 4} \qquad \forall i \in S, p \in P. \tag{8a}$$

$\mathbf{Y}$ then is defined from $\mathbf{X}$ as stated previously as $\mathbf{Y}$ should be 1 for all later time steps after $\mathbf{X}$ is 1.

$$y_{i,p,t} = \sum_{k=0}^{t+1} x_{i,p,k} \quad \forall i \in S, p \in P, t \in T. \tag{9}$$

D. Constraints for Communication Direction

To make sure that the correct satellites communicate in the correct order, the following two sets of constraints are applied. First, all satellites must complete the previous phase before completing the next. This means that

$$x_{j,p,t} \leq y_{j,p-1,t} \qquad \forall t \in T, s \in S, p = 2, \ldots 4. \tag{10a}$$

Second, whenever a satellite wants to send a communication for a phase, they must have received all required communications from the previous phase. Therefore,

$$x_{j,p,t} \leq y_{j,p-1,t} + (1 - w_i - q_i \mathbb{I}_{p=2}) \qquad \forall t \in T, i \in S, j \in S, p = 2, \ldots, 4. \tag{11}$$

E. Constraints for Time Dependencies

The previously defined constraints only enforce $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{q}$, and $\mathbf{w}$ to be in the shape that represents the pBFT communication mechanism and has many feasible solutions. To constrain the problem based on the relative positions of the satellites to allow intersatellite communication, a final of sets of constraints needs to be defined.

First as shown in the example in Section III-B, if the satellite has already received or does not require to receive any messages for this phase the earliest possible time step is chosen. Therefore, as the primary does not receive any messages in phase 1, it completes in the first time step

$$q_i = x_{i,1,0} \quad \forall i \in S. \tag{12}$$

For the remaining constraints, satellites can only communicate if they are able to interact during a specific time step. This constraint is defined as follows: if a satellite had the opportunity to communicate in a time step where the previous phase was completed, then a phase change is permitted to occur. Therefore, we have

$$x_{j,1,t_2} \leq q_i \tau_{i,j,1,t_2+1} + 1 - q_i \qquad \forall t_2 \in T, i \in S, j \in S \tag{13a}$$

$$x_{j,p,t_2} \leq \sum_{t_1=0}^{t_2+1} (\tau_{i,j,t_1,t_2+1} y_{i,p-1,t_1}) + 1 - w_i - q_i \mathbb{I}_{p=3} \qquad \forall t_2 \in T, i \in S, j \in S, p = 2, 3, 4. \tag{13b}$$

F. Objective Function

To complete the definition of the MILP problem, an objective function must be specified. The objective, as stated, is to minimize the consensus time. This is formulated in (14) as $t_{\max}$ minus the sum of $\mathbf{Y}$ in the final phase. Here, $\mathbf{Y}$ is set to 1 only for the primary satellite after it has received the final communication

$$J = t_{\max} - \sum_{j=0}^{n} \sum_{t=0}^{t_{\max}} y_{j,3,t}. \tag{14}$$

IV. Results

The results presented here are based on the selection pool of 30 satellites discussed in Section II solved with the Gurobi solver. To ensure that outcomes are not affected by the choice of

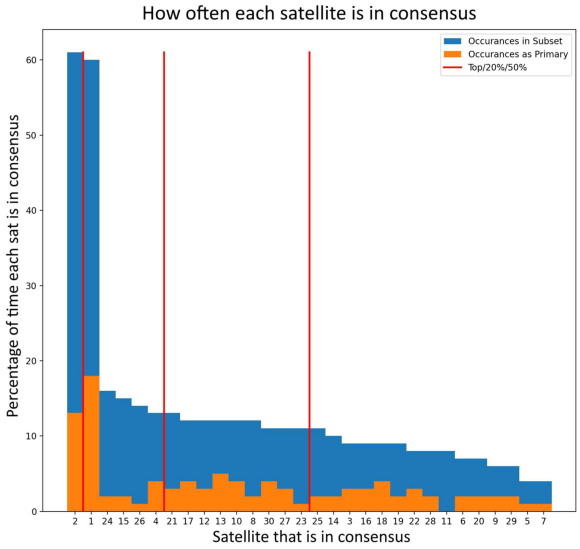

Fig. 5. Percentage of times each satellite in the selection pool is chosen for consensus and chosen as primary. The red lines show the top satellite, top 20%, and the top 50%.

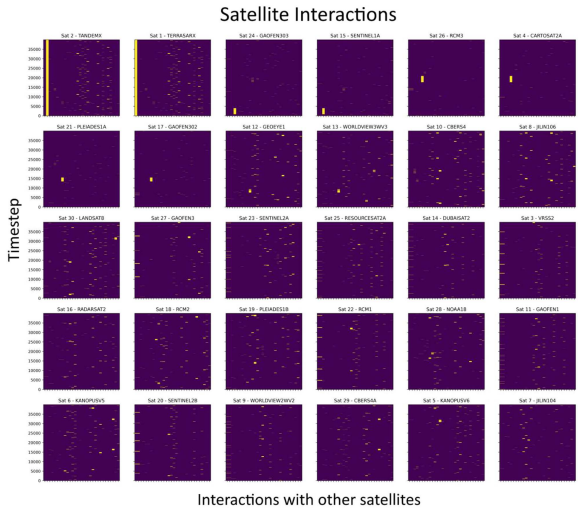

Fig. 6. Each subplot reports when a satellite in the selection pool is in range to communicate with other satellites in the selection pool, sorted by the number of rounds consensus the satellite is chosen.

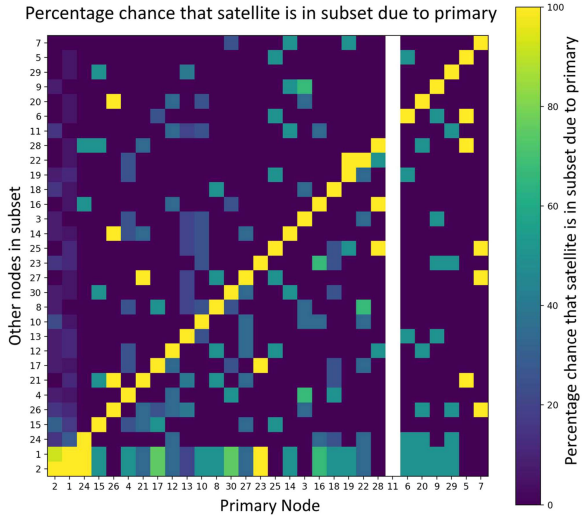

Fig. 7. Distribution of when a satellite is chosen as primary, how often another specific satellite is chosen to be part of consensus.

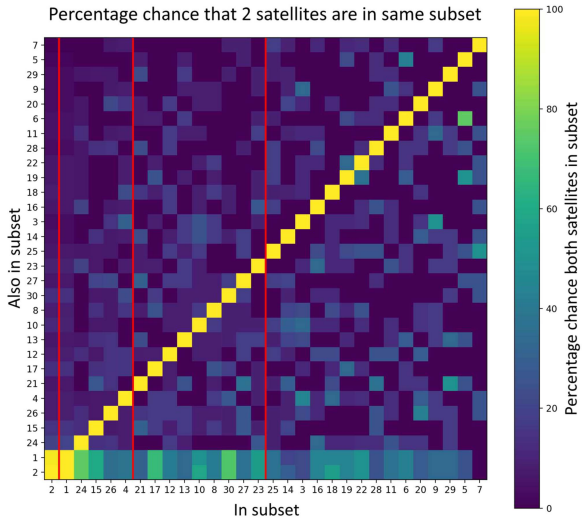

Fig. 8. Distribution of when a satellite is chosen as to take part in consensus, how often another specific satellite is chosen to be part of consensus. The top satellite, top 20%, and top 50% are marked by the red lines.

the initial time, 200 problem instances were solved for consecutive time steps. Each time step was set at a 30-s interval, with a maximum number of timesteps set to 400. This is equivalent to 200 min, which is the average time for two complete orbits to occur for the 30 satellites in the selection pool where the average orbital period of the selection pool is 14.709 orbits per day [23]. This is also a duration deemed acceptable for achieving consensus in an emergency scenario.

The 30-s interval was chosen based on the communication range, which has a radius of 500 km. As satellites in LEO, has a mean orbital velocity of 7.8 km/s [29], if two satellites where travelling in opposite directions, they would have a relative velocity of 15.6 km/s. Over the 500-km radius, this would mean that they would remain approximately 30 s within communication range. This is therefore used as the time step to allow for as many interactions as possible.

From the 200 problem instances, 100 are found to be feasible with the constraints applied within 400 time steps. The number of times each satellite is chosen for consensus is shown in Fig. 5. All satellites are selected for at least one round of consensus; however, not all are chosen to be the primary. Not considering satellites with IDs 1 and 2, there appears to be no correlation between the number of time steps that allow for communication and the frequency with which a satellite is selected for consensus. This suggests that selection is more dependent on the sequence of interactions rather than the duration of satellite-to-satellite communication.

Satellites with IDs 1 and 2 are seen to be chosen more often for both participation and for being primary in the different

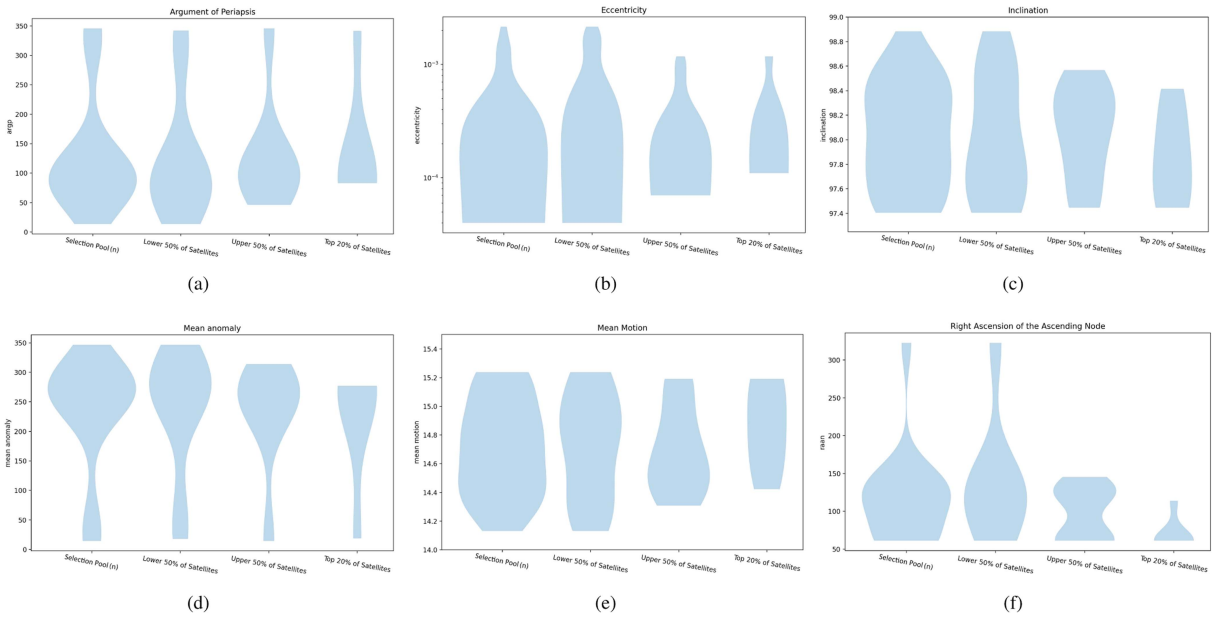

Fig. 9. Orbital parameter distribution for the entire selection pool, the least selected 50%, the most selected 50%, and the top 20% of choices based on popularity, of satellites involved in consensus rounds. (a) Argument of Periapsis. (b) Eccentricity. (c) Inclination. (d) Mean anomaly. (e) Mean motion. (f) Right ascension of the ascending node.

problems instances. This can be explained by looking at Fig. 6 where satellites 1 and 2 are connected 100% of the time as they fly in formation within 500 km of each other. Therefore any communications required for consensus from satellite ID 1 and ID 2 can happen immediately at any time. This gives an advantage to choosing satellites 1 and 2 as they will often increase the speed of consensus, and therefore, be selected to be part of the consensus subset.

From Fig. 7, it can be observed that there are trends between certain satellites being primary and which other satellites are chosen as part of the consensus network. It can be observed that some satellites are always chosen if a certain primary occurs. In some cases, a 100% correlation may be seen if the primary is only chosen for a single consensus round.

One difference between Figs. 7 and 8 is the distribution. Fig. 8 has a flatter distribution, therefore, showing less correlation on how often a satellite is picked based on another except for the clear link between satellite with IDs 1 and 2, which can be seen, where if one is chosen, the other is always chosen as well.

The arrangement of satellites reported in Fig. 5 can also be understood through the orbital parameters displayed in Fig. 9. If we compare the entire selection pool depicted on the left side of each chart to the top 20% of satellites on the extreme right of each chart, the distribution narrows. This is to be expected, as the number of satellites representing the entire selection pool versus the top 20% is less. However, it is notable that the top 20% does not converge toward the mean of the entire selection pool. This indicates that a satellite with orbital parameters close to the mean of the selection pool does not necessarily enhance the number of consensus rounds where the satellite is selected. This is logical because if you are following satellites on the same trajectory you are unlikely to ever move closer to interact or communicate with them.

Another interesting result is that satellite 11 is never the primary however still is part of consensus. Looking at the orbital parameters of satellite 11, the right ascension of the ascending is slightly high outside of the top 50% in Fig. 9. Eccentricity, argument of periapsis and mean anomaly for satellite also differ from the mean. This satellite may have been chosen as part of the 30 selection pool because it interacts many times with satellites not in the top 30, and therefore, would still be chosen to participate.

This approach of employing MILP to choose participants for pBFT consensus can be extended to handle larger scenarios with a greater selection pool of participants. While it is feasible to increase the number of time steps, doing so on the fly to determine participants for a consensus round could lead to delays. For instance, computing 400 time steps at 30 s per step would amount to roughly 3.3 h. If the consensus process exceeds this duration, it could impede ongoing processes. In such cases, it is advisable to opt for a different selection pool to expedite consensus and ensure the relevance of decisions made via the pBFT algorithm.

V. Conclusion

As satellite systems increasingly handle data processing and decision-making onboard, employing fault-tolerant consensus algorithms such as pBFT becomes crucial for ensuring reliable decision making across potentially adversarial satellite networks. While existing research has explored how signal routing and network topology impact communication time, there is a gap in discussing how to optimally select these networks with security considerations in mind.

By applying an MILP to assess a pBFT algorithm using actual satellite trajectories, it is possible to understand how participant

selection in consensus networks is affected by the order of satellite interactions. This selection process is difficult to predict due to the combinatorial nature of the problem. In scenarios using a pool of 30 satellites, consensus is achieved within the maximum time limit of 3.3 h in about half of the simulated cases. Expanding the size of the selection pool may increase the probability of reaching consensus within this time frame, although it would also raise computational complexity.

Moreover, trends in orbital parameters can reveal certain orbital configurations that frequently lead to satellites being chosen for consensus, providing valuable insights into network dynamics and potential future orbit configurations to build consensus oriented heterogeneous network of satellites.

Future work involves exploring several key areas. First, identifying orbit configurations that optimize consensus time while maximizing the number of satellites in a subset. In addition, investigating alternative consensus mechanisms including new DAGs such as Hashgraph [30] and Aleph [31], which are designed to handle asynchronicity better. Scaling the network to encompass multiple clusters of satellites and connecting them into a larger network is another important focus; this approach could also be used to report the reliability of individual satellites across the entire network and incentivize each satellite to compare more data and modalities in its decision-making process. For disaster response scenarios, simulating the revisit time to affected regions is essential to ensure each satellite has sufficient time to make decisions before participating in the consensus process.

References

[1] International Charter: Space and Major Disasters. Accessed: Jan. 5, 2024. [Online]. Available: https://disasterscharter.org

[2] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2442–2473, Oct.–Dec. 2016.

[3] W. Horne, C. F. Kwadrat, and B. L. Edwards, "Inter-satellite communications considerations and requirements for distributed spacecraft and formation flying systems," in *Proc. SpaceOps Conf.*, 2002, pp. 1–10.

[4] G. W. Heckler et al., "NASA's progress toward commercial space communications—Satcom demonstrations and wideband multilingual terminal development," in *Proc. 27th Ka Broadband Commun. Conf.*, 2022, pp. 1–10.

[5] X. Wu, T. Vladimirova, and K. Sidibeh, "Signal routing in a satellite sensor network using optimisation algorithms," in *Proc. IEEE Aerosp. Conf.*, 2008, pp. 1–9.

[6] M. Roth, H. Brandt, and H. Bischl, "Implementation of a geographical routing scheme for low earth orbiting satellite constellations using inter-satellite links," *Int. J. Satell. Commun. Netw.*, vol. 39, no. 1, pp. 92–107, 2021.

[7] L. Chaowen, W. Tao, W. Junrui, C. Shiqi, W. Zhijun, and X. Guoqing, "Inter-satellite routing study for LEO constellations based on orbit prediction," in *Proc. 26th ACIS Int. Winter Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, 2023, pp. 216–221.

[8] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[9] M. Nazari, E. A. Butcher, T. Yucelen, and A. K. Sanyal, "Decentralized consensus control of a rigid-body spacecraft formation with communication delay," *J. Guid., Control, Dyn.*, vol. 39, no. 4, pp. 838–851, 2016.

[10] T. Yucelen and W. M. Haddad, "Consensus protocols for networked multi-agent systems with a uniformly continuous quasi-resetting architecture," *Int. J. Control*, vol. 87, no. 8, pp. 1716–1727, 2014, doi: 10.1080/00207179.2014.883647.

[11] Y.-P. Tian and C.-L. Liu, "Consensus of multi-agent systems with diverse input and communication delays," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2122–2128, Oct. 2008.

[12] V. N. Coelho, R. P. Araújo, H. G. Santos, W. Y. Qiang, and I. M. Coelho, "A MILP model for a Byzantine fault tolerant blockchain consensus," *Future Internet*, vol. 12, no. 11, 2020, Art. no. 185.

[13] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual, 2023. [Online]. Available: https://www.gurobi.com

[14] H. Zhang and P. Gurfil, "Cooperative orbital control of multiple satellites via consensus," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2171–2188, Oct. 2018.

[15] M. Castro et al., "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, pp. 173–186.

[16] L. Lamport, "Generalized consensus and paxos," Tech. Rep. MSR-TR-2005-33, Microsoft Research, 2005.

[17] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319.

[18] S. Popov, "The tangle," *White Paper*, vol. 1, no. 3, pp. 1–10, 2018.

[19] Y. Wang, M. Zhong, and T. Cheng, "Research on PBFT consensus algorithm for grouping based on feature trust," *Sci. Rep.*, vol. 12, no. 1, 2022, Art. no. 12515.

[20] G. Yu, X. Wang, K. Yu, W. Ni, J. A. Zhang, and R. P. Liu, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14155–14181, 2020.

[21] Viasat Real-Time Space Relay. Accessed: Jan. 5, 2024. [Online]. Available: https://www.viasat.com/space-innovation/space-systems/intersatellite-communications/

[22] Space Exploration Holdingsm, LLC, "Request for orbital deployment and operating authority for the spacex gen2 NGSO satellite system," 2022. [Online]. Available: https://docs.fcc.gov/public/attachments/FCC-22-91A1.pdf

[23] Celestrak. Accessed: Jan. 7, 2024. [Online]. Available: https://celestrak.org/

[24] European Space Agency, "Digital twin earth, quantum computing and ai take centre stage at ESA's Φ-week," 2020. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/Digital_Twin_Earth_quantum_computing_and_AI_take_centre_stage_at_ESA_s_Ph-week

[25] Businesswire, "Planet labs PBC announces real-time insights technology using NVIDIA Jetson platform," 2024. [Online]. Available: https://www.businesswire.com/news/home/20240610385569/en/Planet-Labs-Pbc-Announces-Real-Time-Insights-Technology_Using-Nvidia-Jetson-Platform

[26] Intel, "Myriad 2 SoC specifications," 2016. [Online]. Available: https://www.intel.com/content/dam/support/us/en/documents/boardsandkits/neural-compute-sticks/Myriad2VPU-ProductBrief.pdf

[27] NVIDIA, "NVIDIA Jetson for next-generation robotics," 2024. [Online]. Available: https://www.nvidia.com/en-gb/autonomous-machines/embedded-systems/

[28] Pyephem Astronomy Library. Accessed: Mar. 4, 2024. [Online]. Available: https://rhodesmill.org/pyephem/

[29] ESA Low-Earth Orbit. Accessed: Mar. 10, 2024. [Online]. Available: https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit

[30] O. Green, "Hashgraph—Scalable hash tables using a sparse graph data structure," *ACM Trans. Parallel Comput.*, vol. 8, no. 2, pp. 1–17, Jul. 2021, doi: 10.1145/3460872.

[31] A. Gagol, D. Leśniak, D. Straszak, and M. świętek, "ALEPH: Efficient atomic broadcast in asynchronous networks with Byzantine nodes," in *Proc. 1st ACM Conf. Adv. Financial Technol.*, 2019, pp. 214–228.

Robert Cowlishaw (Graduate Student Member, IEEE) is currently working toward the Ph.D. degree in applying distributed ledger technologies to satellite-based emergency mapping with the University of Strathclyde, Glasgow, U.K.

His current research interests include how distributed ledger technologies can automate, accelerate, and decentralize satellite-based emergency mapping.

Annalisa Riccardi received the Ph.D. degree in multi objective multidisciplinary design optimisation techniques for rocket design from the Centre of Industrial Mathematics of the University of Bremen, Germany, in 2012.

She is an Associate Director with the Intelligent Computational Engineering Laboratory and a Senior Lecturer in computational methods with the Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, U.K. She is the Programme Advisor of the M.Sc. in satellite data for sustainable development. She has more than 15 years of experience in machine learning, optimization techniques, and applications in the aerospace domain. She is leading the research in the department on large language models applications to the space domain, blockchain applications to Model-Based Systems Engineering (MBSE) and federated satellite networks, explainable AI for industrial decision making, and the socio-economic applications of satellite data.

Ashwin Arulselvan received the Ph.D. degree in industrial and systems engineering from University of Florida, FL, USA, in 2009.

He is a Senior Lecturer in optimization and analytics with the Department of Management Science, University of Strathclyde, Glasgow, U.K. He has 20 years of experience in optimization and data mining and has made numerous research contributions to the operational research and theoretical computer science on a wide variety of topics such as computational optimisation, Stackelberg games, complexity theory, design of algorithms, and uncertainty modeling. He has served as the Director of multiple M.Sc. and Ph.D. programs with Strathclyde Business School. He has five years of teaching experience in blockchain and smart contract programming.