

Article

Integrated Scheduling Algorithm Based on Matching Game Theory in LEO Satellite Networks

Yuan Xing ¹, Guofeng Zhao ^{1,*} and Zhenzhen Han ²¹ School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; xingystudy@foxmail.com² School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; hanzz@cqupt.edu.cn

* Correspondence: zhaogf@cqupt.edu.cn

Abstract

As an indispensable component of space–terrestrial integrated networks, low-Earth orbit (LEO) satellite networks are capable of providing flexible access and low delay communication services for emerging time-sensitive traffic. However, the inconsistent transmission rates between intra-satellite wired links and inter-satellite wireless links will undoubtedly result in unstable delay at the satellites. This disparity poses a challenge to ensuring deterministic communication for time-sensitive traffic. Aiming at this problem, we put forward an integrated scheduling algorithm based on matching game theory to concurrently determine the positions of wired and wireless time slots. First, we establish a theoretical model to quantify the influence of integrated scheduling on deterministic communication by elucidating the interrelationships among time-sensitive traffic, wired time slots, and wireless time slots. Second, drawing inspiration from scheduling sequences and matching game theory, the established integrated scheduling model is reformulated into a cyclic three-sided matching game model. Third, we design an integrated scheduling algorithm (ISA) to derive scheduling optimization solutions. Experimental results demonstrate that the proposed algorithm ISA outperforms existing scheduling algorithms, achieving an average delay reduction of 16.6% over all comparison algorithms.

Keywords: LEO satellite networks; time-sensitive traffic; deterministic communication; integrated scheduling; matching game theory

1. Introduction

With the rapid growth in demand for ubiquitous and seamless communications, network technologies have gradually expanded from terrestrial to space-based. Low-Earth orbit (LEO) satellite networks will occupy an indispensable position in future space–terrestrial integrated networks (STINs) due to their wide coverage, high throughput, and low delay [1]. In recent years, mega-constellations such as Starlink, OneWeb and Telesat have provided worldwide access and seamless coverage services for terrestrial terminals [2]. Satellites carrying different payloads can cooperate with each other to provide end-to-end transmission services. Emerging time-sensitive applications such as emergency rescue, combat command, and remote control impose rigorous requirements on deterministic and real-time performances. This time-sensitive traffic needs to be delivered within a bounded low delay, meaning that in the worst case, the delay should satisfy the upper bound of the delay requirement. However, existing LEO satellite networks generally adopt



Academic Editor: Changchuan Yin

Received: 7 January 2026

Revised: 1 February 2026

Accepted: 15 February 2026

Published: 20 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

the best-effort forwarding mode, which causes uncertainty in delay. Hence, one of the most challenging current problems is how to implement deterministic communication for time-sensitive traffic appearing in LEO satellite networks.

Fortunately, newly developed deterministic networking technologies such as time-sensitive networking (TSN) [3,4] and deterministic networking (DetNet) [5] can support bounded low delay, high throughput, and high reliability for both terrestrial and non-terrestrial networks. At the same time, these deterministic technologies can support the coexistence of time-sensitive and non-time-sensitive traffic. As the most mainstream deterministic technology, TSN, which originated from terrestrial ethernet, is gradually being applied to the industrial internet [6] and in-vehicle networks [7]. Wireless demand is driving the expansion of TSN to wireless networks such as WiFi and 5G [8,9]. Recently, some researchers have attempted to migrate terrestrial TSN technologies to the aerospace field [10,11] in order to improve the throughput and reliability of internal systems. By analyzing the service quality requirements on switches inside satellites, a possible TSN architecture for an intra-satellite network was designed and the feasibility of the TSN synchronization and scheduling protocols was preliminarily verified [10]. In addition, TSN ethernet technologies were designed and implemented in microlaunchers through the field-programmable gate array (FPGA) [11]. These existing studies indicate that TSN technologies can provide determinism and reliability services for an internal network of satellites.

It is well known that the transmission range of time-sensitive traffic should not be limited to internal network of satellites; rather, it is highly necessary to extend TSN technologies to wireless networks through multiple inter-satellite links. The authors of [12] presented the Space–Terrestrial Time-Sensitive Network (STTSN) architecture, in which space, aerial, and terrestrial TSN clusters are integrated. Some researchers have also attempted to solve the problem of end-to-end deterministic communication in satellite networks. To provide differentiated guarantees, multilevel cyclic queues with latency requirements were investigated in [13]. Similarly, by modifying cyclic queuing and forwarding (CQF) on a single satellite, a latency tolerance scheduling algorithm through multi-hop collaboration was proposed by [14]. Motivated by challenges of dynamic topology and limited transmission resources of satellite networks, a deterministic transmission approach was developed from aspects of routing, coding and scheduling in [15]. To achieve effective transmission of time-sensitive services in STINs, FastTS with multipath redundancy was presented in [16], with reliability assurance and delay optimization considered simultaneously. These studies all treat the satellite as a single node, but do not consider the internal network.

As can be seen from above, the time-sensitive scheduling mechanisms in intra-satellite and inter-satellite networks have been studied independently. However, the integrated scheduling on the output port of gateway has not been covered. Generally, the transmission rate of intra-satellite wired links is higher than that of inter-satellite wireless links. Inconsistent transmission rates make it impossible for data to be scheduled in a timely manner, which increases the queuing delay at the output port. In the worst case, congestion will occur at the integrated gateway inside the satellite. Thus, the bounded delay requirement of time-sensitive traffic cannot be guaranteed. Consequently, the integrated gateway becomes the bottleneck of deterministic scheduling. Existing research has not addressed this problem. Therefore, it is extremely necessary to explore an integrated scheduling algorithm in satellite networks. In this paper, an LEO constellation with wired and wireless integrated links is used as the application scenario and time-sensitive traffic is taken as the research target. To guarantee the deterministic delay experienced on the satellite, we propose an integrated scheduling algorithm based on matching game theory. The main contributions are summarized as follows:

- To accurately quantify the impact of scheduling time slots on deterministic transmission of time-sensitive traffic on the satellite, an integrated scheduling model with the optimization objective of minimizing delay is established. The established model comprehensively takes into account the relationship among time-sensitive traffic, wired time slots, and wireless time slots.
- Inspired by scheduling sequences and matching game theory, we transform the integrated scheduling model into a cyclic three-sided matching game model driven by preference lists. This transformed model simultaneously considers the preference demands of three core participating entities, namely, time-sensitive traffic, wired time slots, and wireless time slots. Each participating entity formulates the ordered ranking of decision-making preferences, which constitutes the preference list.
- To obtain scheduling optimization solutions, we design an integrated scheduling algorithm based on matching game theory. For time-sensitive traffic, the positions of the wired and wireless time slots are jointly allocated by searching for the “best” triple, and an effective solution can be derived by the proposed algorithm.

The rest of this paper is organized as follows: Section 2 presents the satellite network scenario and analyzes problems related to integrated scheduling; Section 3 establishes the network model and formalizes the integrated scheduling problem; Section 4 designs an integrated scheduling algorithm based on matching game theory; Section 5 verifies the performance through a large number of simulation experiments; finally, Section 6 summarizes the paper.

2. Satellite Network Scenario and Problem Statement

The LEO constellation scenario with wired and wireless integrated scheduling mechanism is shown in Figure 1. All devices and switches within satellites communicate with each other through wired ethernet that supports TSN technology. Laser inter-satellite links are used as the communication medium between satellites. The typical time-aware shaper (TAS) scheduling mechanism originating from the IEEE 802.1Qbv protocol [17] is adopted by the intra-satellite wired links; meanwhile, the time-division multiple access (TDMA) scheduling mechanism is utilized by the inter-satellite wireless links. Both TAS and TDMA can effectively avoid collisions by scheduling time-sensitive traffic in different time slots. As is well known, time-sensitive traffic can be transmitted over long distances through cooperation between intra-satellite wired links and inter-satellite wireless links in an LEO constellation. However, the transmission rate of wired links is usually higher than that of wireless links. Inconsistent transmission rates may increase queuing delay or even bring about congestion at the integrated gateway inside satellites. The final result is that the bounded delay requirement of time-sensitive traffic cannot be guaranteed. Fortunately, for time-sensitive traffic, the allocated intra-satellite wired time slots and inter-satellite wireless time slots will jointly determine the delay experienced on satellite. Therefore, it is very necessary to design an integrated scheduling algorithm to jointly allocate wired time slots and wireless time slots. To achieve deterministic scheduling on satellite with integrated links, the following two problems need to be resolved:

(1) How should we characterize the impact of wired and wireless integrated scheduling on the delay experienced at the satellite?

For time-sensitive traffic, the delay experienced at the satellite is determined by both wired and wireless time slots. For example, if the positions of the wired time slot and wireless time slot are both outside the transmission period of time-sensitive traffic, a scheduling timeout will occur. Ultimately, the determinism of delay is further impacted. Therefore, to achieve deterministic scheduling, it is necessary to characterize the impact of integrated scheduling on the delay experienced at the satellite.

(2) How can we achieve joint allocation of both wired and wireless time slots?

Integrated scheduling of intra-satellite wired links and inter-satellite wireless links is the key to realizing deterministic communication. Thus, for time-sensitive traffic transmitted on satellites, it is very necessary to achieve a stable forwarding delay. However, independently or randomly allocating wired time slots and wireless time slots will bring about uncertainty in the delay. Therefore, we face the issue of how to design an algorithm to realize the joint allocation of wired and wireless time slots based on the relationship among time-sensitive traffic, wired time slots, and wireless time slots.

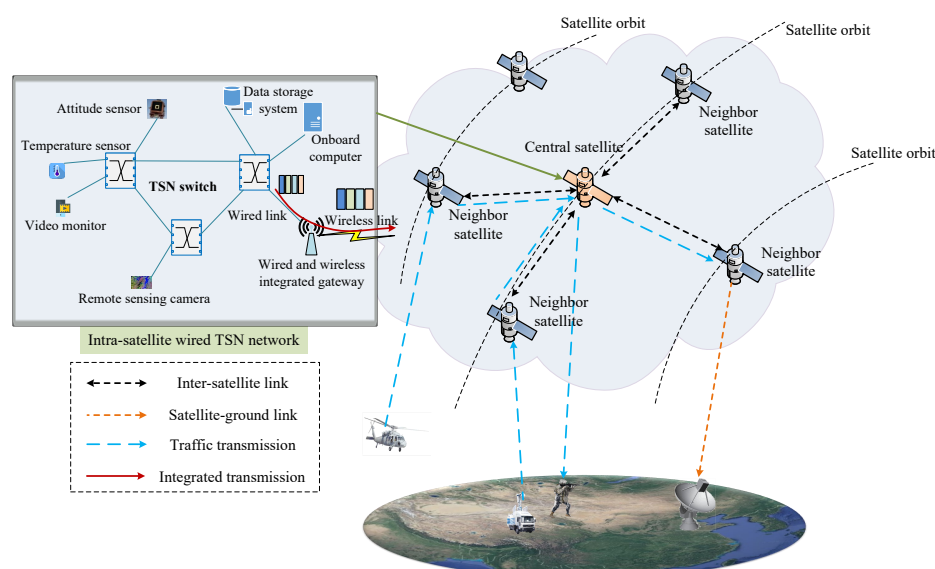


Figure 1. Wired and wireless integrated scheduling in an LEO constellation.

3. Integrated Scheduling System Model

First, we provide an overview of the integrated wired and wireless network model in a satellite network. Second, the relationship between wired and wireless time slots is characterized by the scheduling model. Finally, the optimization objective of integrated scheduling is established.

3.1. Wired and Wireless Integrated Network Model

Regarding the wired and wireless integrated LEO constellation, the network model consists of satellites $v \in V$ and links $e \in E = \{E^w, E^{wl}\}$. Edge $e^w \in E^w$ represents the intra-satellite wired link, while edge $e^{wl} \in E^{wl}$ denotes the inter-satellite wireless link. The set E^w consists of terminals and switches that support TSN technology, and the topological relationship of devices within the satellite is relatively fixed. The set E^{wl} is composed of edges connecting two adjacent satellites. With the periodic motion of satellites, the ISL distance between two adjacent orbital planes changes with time. However, the topology connection relationship will remain relatively stable in a period of time [18]. Thus, wired and wireless integrated scheduling can be converted into a problem of continuous scheduling within multiple adjacent time intervals. A simple integrated scheduling process is shown in Figure 2. To achieve deterministic scheduling on different satellites, the central satellite is responsible for allocating and adjusting scheduling time slots autonomously.

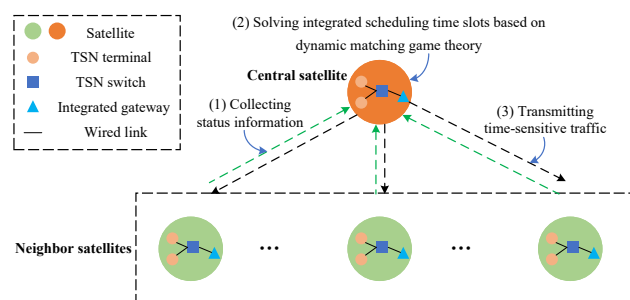


Figure 2. Wired and wireless integrated scheduling process.

For ease of distinction, we define two types of time slots: topology slots and scheduling slots. A topology time slot means that the connection relationship between satellites remains unchanged within a given time. A scheduling time slot refers to the time slice used to transmit time-sensitive traffic on wired and wireless links. It is worth noting that the duration time of a topology time slot is greater than that of a scheduling time slot; this ensures that time-sensitive traffic is fully scheduled before the topology changes. The topology time slot is represented by $t \in [1, \mathbb{T}]$. The total number of topology time slots is $\mathbb{T}$. Let the set $\mathcal{I}^t = \{1^t, 2^t, \dots, I^t\}$ denote time-sensitive traffic in t , where $i^t \in \mathcal{I}^t$; the set $\mathcal{J}^t = \{1^t, 2^t, \dots, J^t\}$ represent wired time slots in t , where $j^t \in \mathcal{J}^t$; and the set $\mathcal{K}^t = \{1^t, 2^t, \dots, K^t\}$ express wireless time slots in t , where $k^t \in \mathcal{K}^t$. The integrated scheduling relationship between $\mathcal{I}^t$, $\mathcal{J}^t$ and $\mathcal{K}^t$ can be expressed as the three-sided matching $\mathcal{M}^t \subseteq \mathcal{I}^t \times \mathcal{J}^t \times \mathcal{K}^t$. For ease of reading, Table 1 provides a description of the main notation.

Table 1. Main notation and descriptions.

Notations	Descriptions
t	Topology time slot t , where $t \in [1, \mathbb{T}]$
$\mathbb{T}$	Total number of topology time slots
$\mathcal{I}^t$	Set of time-sensitive traffic in t
$\mathcal{J}^t$	Set of wired time slots in t
$\mathcal{K}^t$	Set of wireless time slots in t
$\mathcal{M}^t$	Set of three-sided matching in t
i^t	Specific time-sensitive traffic in t , $i^t \in \mathcal{I}^t$
j^t	Specific wired time slot in t , $j^t \in \mathcal{J}^t$
k^t	Specific wireless time slot in t , $k^t \in \mathcal{K}^t$
$x_{i,j}^t$	Indicates whether i^t is scheduled in j^t
$y_{j,k}^t$	Indicates whether j^t is scheduled in k^t
T_{i^t}	Generation period of i^t
B_{i^t}	Data volume of i^t
m_{i^t}	The m th subperiod of i^t
L_w	Length of wired scheduling time slot
L_{wl}	Length of wireless scheduling time slot
$P_{i^t}, P_{j^t}, P_{k^t}$	The preference list of i^t , j^t , and k^t , respectively
$\mathcal{M}^t(i^t), \mathcal{M}^t(j^t), \mathcal{M}^t(k^t)$	The preference set of i^t , j^t , and k^t , respectively

The variable $x_{i,j}^t$ indicates the assignment of the wired time slot j^t to the time-sensitive traffic i^t . Similarly, the variable $y_{j,k}^t$ denotes the allocation of the wireless time slot k^t to the wired time slot j^t . The above two categories of binary variables are defined as follows:

$$x_{i,j}^t = \begin{cases} 1, & \text{if } j^t \text{ is assigned to } i^t, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

and

$$y_{j,k}^t = \begin{cases} 1, & \text{if } k^t \text{ is assigned to } j^t, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

3.2. Wired and Wireless Integrated Scheduling Model

To intuitively display the scheduling relationship between wired time slots and wireless time slots, an integrated scheduling model for time-sensitive traffic is established. Thus, the relationship between integrated scheduling and deterministic transmission is effectively characterized via mathematical model.

3.2.1. Concepts of Hyperperiod and Subperiod

In LEO satellite networks, time-sensitive traffic i^t is generated at regular intervals with a fixed amount of data. For the sake of convenience, we define several parameters: the traffic generation period is represented by T_{it} , the data volume is indicated by B_{it} , and the required delay to meet the deadline is denoted as D_{it}^{re} . Furthermore, the properties of i^t can be characterized by the tuple $\{T_{it}, B_{it}, D_{it}^{re}\}$. To facilitate effective scheduling, we introduce the concepts of hyperperiod and subperiod, as illustrated in Figure 3. The red squares in Figure 3 represent time slots occupied by time-sensitive traffic. The subperiod is designed to reduce the solution search space, while the hyperperiod is designed to avoid repetitive solution computation. The scheduling mechanism segments continuous time into multiple uniformly-sized time slots. The durations of wired time slots and wireless time slots are represented by L_w and L_{wl} , respectively. The duration of the hyperperiod, denoted as H^t , is established as the least common multiple (LCM) of all individual periods $\{T_{it}\}$, specifically,

$$H^t = LCM\{T_{1^t}, T_{2^t}, \dots, T_{(I-1)^t}, T_{I^t}\}. \quad (3)$$

A subperiod is defined as a repetitive period within a hyperperiod. Hence, all time-sensitive traffic i^t can be accommodated as integer multiples, where the number of subperiods in H^t is defined as $M_{it} = H^t / T_{it}$.

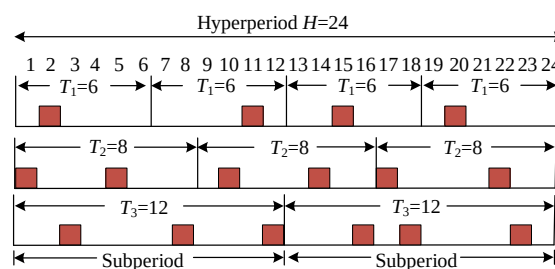


Figure 3. Examples of hyperperiod and subperiod.

Within the m th subperiod of i^t , the scope of wired time slot numbers is defined as the interval $[a_{m_{it}}^w, b_{m_{it}}^w]$. The specific expressions are as follows:

$$a_{m_{it}}^w = \frac{(m_{it} - 1)T_{it}}{L_w} + 1, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}] \quad (4)$$

and

$$b_{m_{it}}^w = \frac{m_{it}T_{it}}{L_w}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (5)$$

Similarly, the starting number of wireless time slots is given by

$$a_{m_i^t}^{wl} = \frac{(m_{i^t} - 1)T_{i^t}}{L_{wl}} + 1, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}], \quad (6)$$

while the ending number of wireless time slot is represented as

$$b_{m_i^t}^{wl} = \frac{m_{i^t} T_{i^t}}{L_{wl}}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (7)$$

3.2.2. Time Slot Allocation in Wired Scheduling

The intra-satellite wired link adopts the mainstream TAS scheduling mechanism. However, TAS only provides fundamental guidelines for different queues in output port through the gate control list (GCL). The specific method for determining the gate status, for example when to open or close the gate, has not been described. Given that time-sensitive traffic imposes stringent requirements regarding delay and jitter, whereas non-time-sensitive traffic does not have such constraints, the integrated scheduling design primarily focuses on the GCL designated for time-sensitive traffic.

The variable $x_{i,j}^t$ determines the allocation of wired time slots and takes values in $\{0, 1\}$. Specifically, $x_{i,j}^t = 1$ signifies that the gate is open, allowing scheduling of i^t in j^t . Conversely, if $x_{i,j}^t = 0$, the gate remains closed, and scheduling is prohibited. To avoid conflicts, each j^t can only be assigned to one i^t .

$$\sum_{i^t \in \mathcal{I}^t} x_{i,j}^t \leq 1, \forall j^t \in \mathcal{J}^t, t \in [1, \mathbb{T}] \quad (8)$$

Furthermore, the total number of assigned wired time slots must not surpass the overall number of available wired time slots $\mathcal{J}^t$.

$$\sum_{i^t \in \mathcal{I}^t} \sum_{j^t \in \mathcal{J}^t} x_{i,j}^t \leq J^t, \forall t \in [1, \mathbb{T}] \quad (9)$$

Let $s_{m_i^t}$ represent the number of wired time slots allocated to i^t in the m th subperiod, which can be expressed as

$$s_{m_i^t} = \sum_{j^t = a_{m_i^t}^{wl}}^{b_{m_i^t}^{wl}} x_{i,j}^t, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (10)$$

To guarantee that i^t is transmitted with low delay and that the generated traffic does not accumulate, it is necessary to complete scheduling within the subperiod. Referring to Equations (9) and (10), we derive the following:

$$\sum_{i^t=1}^{I^t} \sum_{m_i^t=1}^{M_i^t} s_{m_i^t} = \sum_{i^t=1}^{I^t} \sum_{j^t=1}^{J^t} x_{i,j}^t = \sum_{i^t=1}^{I^t} \sum_{m_i^t=1}^{M_i^t} \sum_{j^t=a_{m_i^t}^{wl}}^{b_{m_i^t}^{wl}} x_{i,j}^t \leq J^t, \forall t \in [1, \mathbb{T}]. \quad (11)$$

3.2.3. Time Slot Allocation in Wireless Scheduling

To prevent collisions during wireless scheduling, it is imperative that each wireless time slot k^t be assigned exclusively to a single wired time slot j^t .

$$\sum_{j^t \in \mathcal{J}^t} y_{j,k}^t \leq 1, \forall k^t \in \mathcal{K}^t, t \in [1, \mathbb{T}] \quad (12)$$

Moreover, the cumulative quantity of wireless time slots assigned for wired time slots must not surpass the total number K^t .

$$\sum_{j^t \in \mathcal{J}^t} \sum_{k^t \in \mathcal{K}^t} y_{j,k}^t \leq K^t, \forall t \in [1, \mathbb{T}] \quad (13)$$

For wireless link e^{wl} , the transmission rate is denoted as $R_{e^{wl}}^t$, which is influenced by link conditions. The specific mathematical expression of $R_{e^{wl}}^t$ can be found in [19]. For wired link e^w , the transmission rate is expressed as R_{e^w} . Generally, R_{e^w} surpasses $R_{e^{wl}}^t$, which may lead to congestion at the integrated gateway. To guarantee that the traffic transmitted during a wired time slot is fully accommodated by a corresponding wireless time slot, the following configuration relationship should be satisfied:

$$R_{e^{wl}}^t L_{wl} \geq R_{e^w} L_w. \quad (14)$$

To attain integrated scheduling for time-sensitive traffic i^t , the wired time slot must be scheduled earlier than the wireless time slot. Therefore, the positional relationship between the allocated wired time slot and wireless time slot is represented as follows:

$$j^t x_{i,j}^t L_w \leq (k^t - 1) y_{j,k}^t L_{wl}, \forall i^t \in \mathcal{I}^t, j^t \in \mathcal{J}^t, k^t \in \mathcal{K}^t, t \in [1, \mathbb{T}]. \quad (15)$$

3.2.4. Wired and Wireless Integrated Scheduling Modeling

By integrated scheduling, we mean the phenomenon by which wired and wireless time slots are allocated in conjunction. Figure 4 presents a schematic diagram illustrating the components of delay under integrated scheduling. Specifically, the delay experienced by time-sensitive traffic is primarily influenced by periodic arrival characteristics, transmission rates, and scheduling time slots. The time of arrival (ToA) of time-sensitive traffic i^t in its m th subperiod is mathematically expressed as

$$ToA_{m,i^t} = t_{\text{start}}^t + (m_{i^t} - 1)T_{i^t}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (16)$$

The time t_{start}^t is the absolute start time of topology time slot t .

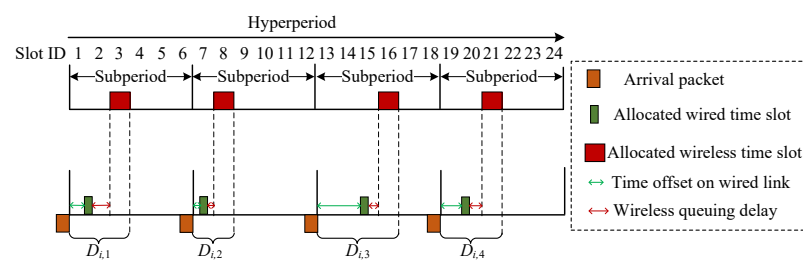


Figure 4. Wired and wireless integrated scheduling diagram.

For time-sensitive traffic i^t in the m th subperiod, the delay determined by integrated scheduling can be computed as follows:

$$D_{m,i^t} = \delta_{m,i^t} + D_{m,i^t}^{\text{Tran}} + D_{m,i^t}^{\text{Que-wl}}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}], \quad (17)$$

where δ_{m,i^t} is the time offset on the wired link, D_{m,i^t}^{Tran} is the transmission delay, and $D_{m,i^t}^{\text{Que-wl}}$ is the wireless queuing delay.

In addition, δ_{m,i^t} is characterized as the temporal delay that occurs between the moment of traffic generation and the corresponding position of the assigned wired time slot, which is represented as

$$\delta_{m_i^t} = \sum_{j^t=a_{m_i^t}^w}^{b_{m_i^t}^w} (j^t - a_{m_i^t}^w) x_{i,j}^t L_w, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (18)$$

Here, $D_{m_i^t}^{Tran}$ is comprised of two components: the transmission delay experienced on the wired link, denoted as $D_{m_i^t}^{Tran-w}$, and the transmission delay experienced on wireless link, represented as $D_{m_i^t}^{Tran-wl}$.

$$D_{m_i^t}^{Tran} = D_{m_i^t}^{Tran-w} + D_{m_i^t}^{Tran-wl} = \frac{B_{i^t}}{R_{ew}} + \frac{B_{i^t}}{R_{ewl}}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}] \quad (19)$$

In the above equation, $D_{m_i^t}^{Que-wl}$ represents the queuing delay before the traffic is scheduled by the wireless time slot but completed by the wired time slot, and is defined as the temporal difference between the finishing of wired scheduling and the beginning of wireless scheduling. Within the integrated scheduling design, the constraint shown in (15) should be satisfied simultaneously. In accordance with the positional relationship between allocated wired time slot j^t and wireless time slot k^t , we can express $D_{m_i^t}^{Que-wl}$ as

$$D_{m_i^t}^{Que-wl} = \sum_{k^t=a_{m_i^t}^{wl}}^{b_{m_i^t}^{wl}} \sum_{j^t=a_{m_i^t}^w}^{b_{m_i^t}^w} [(k^t - 1)L_{wl} - j^t L_w] x_{i,j}^t y_{j,k}^t, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (20)$$

3.3. Problem Formulation for Integrated Scheduling

As presented in (21), we establish the optimization objective to minimize the delay associated with time-sensitive traffic. Constraint (22) indicates whether j^t is allocated to i^t . Constraint (23) indicates whether k^t is allocated to j^t . Constraint (24) indicates that each j^t can only be allocated to one i^t . Constraint (25) means that the aggregate number of wired time slots assigned to all time-sensitive traffic must not surpass the total number of available wired time slots J^t . Constraint (26) indicates that each k^t can be assigned to only one j^t . Constraint (27) specifies that the total number of wireless time slots allocated to all wired time slots cannot exceed the total number of wireless time slots K^t . Constraint (28) shows that the scheduling of a wired time slot must precede that of a wireless time slot. Constraint (29) means that the delay associated with integrated scheduling must not exceed the predetermined delay requirement.

$$\min D_{m_i^t}, \quad \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}], \quad (21)$$

$$\text{s.t. } x_{i,j}^t \in \{0, 1\}, \forall i^t \in \mathcal{I}^t, j^t \in \mathcal{J}^t, t \in [1, \mathbb{T}], \quad (22)$$

$$y_{j,k}^t \in \{0, 1\}, \forall j^t \in \mathcal{J}^t, k^t \in \mathcal{K}^t, t \in [1, \mathbb{T}], \quad (23)$$

$$\sum_{i^t \in \mathcal{I}^t} x_{i,j}^t \leq 1, \forall j^t \in \mathcal{J}^t, t \in [1, \mathbb{T}], \quad (24)$$

$$\sum_{i^t \in \mathcal{I}^t} \sum_{j^t \in \mathcal{J}^t} x_{i,j}^t \leq J^t, \forall t \in [1, \mathbb{T}], \quad (25)$$

$$\sum_{j^t \in \mathcal{J}^t} y_{j,k}^t \leq 1, \forall k^t \in \mathcal{K}^t, t \in [1, \mathbb{T}], \quad (26)$$

$$\sum_{j^t \in \mathcal{J}^t} \sum_{k^t \in \mathcal{K}^t} y_{j,k}^t \leq K^t, \forall t \in [1, \mathbb{T}], \quad (27)$$

$$j^t x_{i,j}^t L_w \leq (k^t - 1) y_{j,k}^t L_{wl}, \forall i^t \in \mathcal{I}^t, j^t \in \mathcal{J}^t, k^t \in \mathcal{K}^t, t \in [1, \mathbb{T}], \quad (28)$$

$$D_{m_i^t} \leq D_{i^t}^{re}, \forall i^t \in \mathcal{I}^t, t \in [1, \mathbb{T}]. \quad (29)$$

The decision variables $x_{i,j}^t$ and $y_{j,k}^t$ indicate that the integrated scheduling problem presented above is classified as a 0–1 integer programming problem, which is recognized as NP-complete. However, the purpose of integrated scheduling is to obtain feasible solutions that satisfy delay constraints. Inspired by scheduling sequences and allocation relationships, we introduce matching game theory to address integrated scheduling problem. The algorithm informed by matching game theory is capable of generating suboptimal solutions while also accommodating the preferences of each involved entity.

4. Integrated Scheduling Algorithm Based on Matching Game Theory

According to types of participating entities, the wired and wireless integrated scheduling model mentioned above can be transformed into a cyclic three-sided matching game model. Moving forward, we design an integrated scheduling algorithm based on matching game theory.

4.1. Three-Sided Matching Game Model

A three-sided matching game can effectively represent the relationship between participating entities, for example the three-sided relationship composed of “man–woman–dog” in the real world [20]. The main feature of the matching game is that each class of entity only has a preference list for another class of entity [20]. Therefore, in the integrated scheduling model, time-sensitive traffic only has a preference list P_{it} for wired time slots, wired time slots only have a preference list P_{jt} for wireless time slots, and wireless time slots only have a preference list P_{kt} for time-sensitive traffic. For time-sensitive traffic $\mathcal{I}^t$, wired time slot $\mathcal{J}^t$, and wireless time slot $\mathcal{K}^t$, all the possible triples are expressed as $\mathcal{I}^t \times \mathcal{J}^t \times \mathcal{K}^t = \mathcal{T}^t$. The three-sided matching set is denoted with $\mathcal{M}^t \subseteq \mathcal{T}^t$. For the matching triple $(i^t, j^t, k^t) \in \mathcal{M}^t$, $\mathcal{M}^t(i^t) = j^t$ means that wired time slot $j^t \in \mathcal{J}^t$ is only accepted by time-sensitive traffic $i^t \in \mathcal{I}^t$, while $\mathcal{M}^t(j^t) = k^t$ implies that wireless time slot $k^t \in \mathcal{K}^t$ is only accepted by wired time slot $j^t \in \mathcal{J}^t$ and $\mathcal{M}^t(k^t) = i^t$ indicates that time-sensitive traffic $i^t \in \mathcal{I}^t$ is only accepted by wireless time slot $k^t \in \mathcal{K}^t$.

The ultimate purpose of the three-sided matching game is to discover a stable matching set under preference lists. The blocking triple can effectively indicate whether the three-sided matching game has reached a stable matching. The specific definition of a blocking triple is expressed below.

Definition 1. The triple $(i^t, j^t, k^t) \notin \mathcal{M}^t$ but $(i^t, j^t, k^t) \in \mathcal{T}^t$ is a blocking triple if the following set is satisfied:

$$\begin{aligned} &\{\mathcal{M}^t(i^t) = \emptyset \vee j^t \succ_{i^t} \mathcal{M}^t(i^t)\} \wedge \\ &\{\mathcal{M}^t(j^t) = \emptyset \vee k^t \succ_{j^t} \mathcal{M}^t(j^t)\} \wedge \\ &\{\mathcal{M}^t(k^t) = \emptyset \vee i^t \succ_{k^t} \mathcal{M}^t(k^t)\}, \end{aligned} \quad (30)$$

where $j^t \succ_{i^t} \mathcal{M}^t(i^t)$ means that time-sensitive traffic i^t prefers wired time slot j^t to its current matched wired time slot $\mathcal{M}^t(i^t)$, $k^t \succ_{j^t} \mathcal{M}^t(j^t)$ indicates that wired time slot j^t prefers wireless time slot k^t to its current matched wireless time slot $\mathcal{M}^t(j^t)$, and $i^t \succ_{k^t} \mathcal{M}^t(k^t)$ indicates that wireless time slot k^t prefers time-sensitive traffic i^t to its current matched time-sensitive traffic $\mathcal{M}^t(k^t)$. If there are no blocking triples, the three-sided matching $\mathcal{M}^t$ is considered stable.

However, the traditional three-sided matching game model lacks a representation of relationships among different preference lists, which makes it difficult to solve. Inspired by scheduling sequences, we transform the integrated scheduling model into a cyclic three-sided matching game model for time-sensitive traffic, as shown in Figure 5. In circular

preference lists, P_{it} , P_{jt} , and P_{kt} are set as the first, second, and third preference lists, respectively. Different matching entities are constrained through corresponding preference lists. Specifically, the preference lists P_{it} , P_{jt} , and P_{kt} are formed as follows: the preference list P_{it} of time-sensitive traffic for wired time slots is sorted according to the wired scheduling delay $d_{jt}(i^t)$, that is, time-sensitive traffic prefers wired time slots with lower delay; the preference list P_{jt} of wired time slots for wireless time slots is sorted according to the wireless link transmission rate R_{kt} , meaning that wired time slots prefer wireless time slots with higher transmission rates; and the preference list P_{kt} of wireless time slots for time-sensitive traffic is sorted in accordance with priority α_{it} , that is, wireless time slots preferentially select time-sensitive traffic with high priority for scheduling.

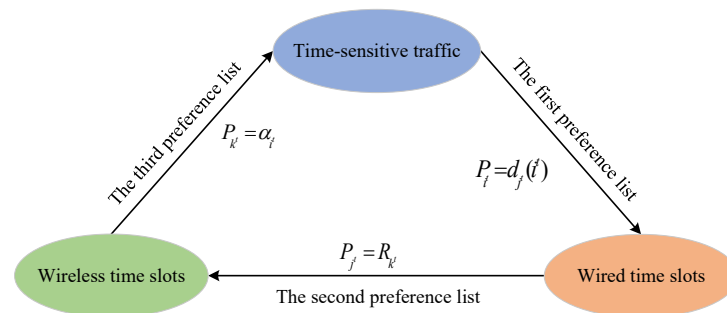


Figure 5. Cyclic three-sided matching game model for time-sensitive traffic.

Before providing the details of the designed scheduling algorithm, the necessary sets are defined as follows:

$$A^{+1}(\mathcal{M}^t, i^t) = \{j^t | j^t \succ_{it} \mathcal{M}^t(i^t), j^t \in P_{it}\}. \quad (31)$$

For time-sensitive traffic i^t , there exists a set of wired time slots that are more preferred than i^t 's existing matching $\mathcal{M}^t(i^t)$:

$$A^{+1}(\mathcal{M}^t, j^t) = \{k^t | k^t \succ_{jt} \mathcal{M}^t(j^t), k^t \in P_{jt}\}. \quad (32)$$

For wired time slot j^t , there exists a set of wireless time slots that are preferred over j^t 's current matching $\mathcal{M}^t(j^t)$:

$$A^{-1}(\mathcal{M}^t, i^t) = \{k^t | k^t \in \mathcal{K}^t, i^t \in P_{kt}, \mathcal{N}(\mathcal{M}^t, k^t) \leq N_{kt}\} \quad (33)$$

which represents the set of wireless time slots that can still accommodate time-sensitive traffic:

$$A^{-2}(\mathcal{M}^t, i^t) = \{j^t | j^t \in \mathcal{J}^t, A^{+1}(\mathcal{M}^t, j^t) \cap A^{-1}(\mathcal{M}^t, i^t) \neq \emptyset\}. \quad (34)$$

For wired time slot j^t , there is a wireless time slot k^t that is more preferred than j^t 's existing matching $\mathcal{M}^t(j^t)$ while still being capable of accommodating the time-sensitive traffic i^t .

4.2. Integrated Scheduling Algorithm

By combining the above cyclic three-sided matching game model, we design an integrated scheduling algorithm (ISA) that can realize joint allocation of wired and wireless time slots. The basic idea of the ISA is to search for the “best” triple through circular preference lists. For each time-sensitive traffic instance, it selects the best wired time slot satisfying the requirements; then, the selected wired time slot chooses the best wireless time

slot that meets its requirements. When the time-sensitive traffic completes the scheduling request, a stable matching triple between time-sensitive traffic and scheduling time slots is obtained. At last, these generated triples are added to the matching set $\mathcal{M}^t$.

In light of these preference definitions, Algorithm 1 describes the detailed procedures of the proposed ISA. Here, we introduce the indicators “flag” to guide the execution of the algorithm and $Head(X, z)$ to represent the element in X which has the highest priority in preference list of z . For the proposed algorithm, the input consists of the participating entities $\mathcal{I}^t$, $\mathcal{J}^t$, and $\mathcal{K}^t$. The output is the matching set $\mathcal{M}^t$ containing integrated scheduling results. The algorithm mainly consists of two stages: the initialization stage (lines 1~3), and the matching game stage (lines 4~25). In the initialization stage, the preference lists P_{it} , P_{jt} and P_{kt} are created according to the known preference information. The set $\mathcal{M}^t$ is initialized to the empty set $\emptyset$ and the “flag” indicator is set to one. To satisfy the deadline delay requirement, the time slot allocation order for time-sensitive traffic is based on priority α_{it} in descending order. In the matching game stage, for each time-sensitive traffic i^t , a better wired time slot j'^t is discovered in steps 7~9. If there exists such a wired time slot, a better wireless time slot k'^t for j'^t is searched in steps 10~11. Next, the capacity of time-sensitive traffic i^t is checked. If $\mathcal{N}(\mathcal{M}^t, i^t) = I^t$, then the worst matching triple $(i^t, \mathcal{M}^t(i^t), \mathcal{M}^t(\mathcal{M}^t(i^t)))$ is deleted (lines 12~16). Similarly, the capacity of wired time slot j'^t is checked and dealt with in steps 17~20. Then, the newly generated triple (i^t, j'^t, k'^t) is added to the set $\mathcal{M}^t$. The matching process is iterated until reaching maximum cardinality $|\mathcal{M}^t|$. At last, a stable matching set $\mathcal{M}^t$ in topology time slot t is acquired.

Algorithm 1 Integrated Scheduling Algorithm

Input: $\mathcal{I}^t, \mathcal{J}^t, \mathcal{K}^t$;

Output: $\mathcal{M}^t$;

```

1: Initialization: Create the preference lists  $P_{it}$ ,  $P_{jt}$  and  $P_{kt}$ ;
2: Set  $\mathcal{M}^t = \emptyset$  and flag = 1;
3: Sort time-sensitive traffic  $i^t$  in descending order of priority  $\alpha_{it}$ ;
4: while flag == 1 do
5:   Set flag = 0;
6:   for each time-sensitive traffic  $i^t \in \mathcal{I}^t$  do
7:      $\mathcal{J}'^t = A^{+1}(\mathcal{M}^t, i^t) \cap A^{-2}(\mathcal{M}^t, i^t)$ ;
8:     if  $\mathcal{J}'^t \neq \emptyset$  then
9:        $j'^t = Head(\mathcal{J}'^t, i^t)$ ;
10:       $\mathcal{K}'^t = A^{+1}(\mathcal{M}^t, j'^t) \cap A^{-1}(\mathcal{M}^t, i^t)$ ;
11:       $k'^t = Head(\mathcal{K}'^t, j'^t)$ ;
12:      if  $\mathcal{N}(\mathcal{M}^t, i^t) = I^t$  then
13:        Select the worst matching in set  $\{(i^t, \mathcal{M}^t(i^t), \mathcal{M}^t(\mathcal{M}^t(i^t)))\}$ ;
14:         $\mathcal{M}^t = \mathcal{M}^t \setminus worst\{(i^t, \mathcal{M}^t(i^t), \mathcal{M}^t(\mathcal{M}^t(i^t)))\}$ ;
15:        Set flag=1;
16:      end if
17:      if  $\mathcal{N}(\mathcal{M}^t, j'^t) = 1$  then
18:         $\mathcal{M}^t = \mathcal{M}^t \setminus (*, j'^t, \mathcal{M}^t(j'^t))$ ;
19:        Set flag=1;
20:      end if
21:       $\mathcal{M}^t = \mathcal{M}^t \cup (i^t, j'^t, k'^t)$ ;
22:    end if
23:  end for
24: end while
25: return  $\mathcal{M}^t$ ;

```

5. Performance Evaluation

In this section, we first construct a satellite constellation scenario by leveraging specialized network simulation tools. Subsequently, extensive simulation experiments are carried out to validate the performance metrics. Finally, the experimental results are compared against those of existing scheduling algorithms and analyzed in conjunction from multiple perspectives.

5.1. Simulation Setup

To verify the feasibility and availability of the proposed algorithm, an LEO satellite constellation scenario integrated with both wired and wireless links is constructed. The specific simulation tools utilized include Satellite Tool Kit (STK) 12.2.0 [18] and PyCharm 2024.1.3. STK is employed to establish the satellite constellation scenario. The detailed orbital parameters are derived from the Starlink constellation. Subsequently, the scheduling algorithms are implemented in Python3 utilizing the configuration parameters exported from STK. Specifically, the primary parameters adopted for the simulation are elaborated in Table 2, which were established according to references [10,16]. The number of TSN-enabled terminals within each satellite is configured as 5–8 [10]. To enable deterministic transmission, both terminals and switches within satellites are equipped with TAS capability. The wired and wireless integrated gateway is compatible with both TAS and TDMA. The TDMA is used for transmitting traffic over inter-satellite wireless links. The transmission rate of the intra-satellite wired link is configured as a constant value 1 Gbps. To accurately support time slot scheduling in the satellite network, the synchronization accuracy between devices is guaranteed through the IEEE 802.1AS protocol [21]. During the simulation experiments, the time-sensitive traffic is produced in a periodic manner. To avoid traffic backlog and mitigate the risk of congestion, the delay requirement is aligned with the traffic generation period.

Table 2. Simulation parameters.

Parameter	Value
Altitude of the orbit	550 km
Inclination of orbit	53 deg
Number of orbits	14
Number of satellites per orbit	22
Total number of satellites in constellation	308
Number of inter-orbit ISLs per satellite	4
Rate of intra-satellite wired link	1 Gbps
Rate of inter-satellite wireless link	[250, 500, 750, 1000] Mbps
Number of terminals inside each satellite	5–8

5.2. Performance Analysis

This subsection concentrates on presenting and analyzing the performance metrics of various scheduling algorithms. To evaluate the effectiveness and efficiency of the proposed algorithm, comparative analyses are conducted against existing representative TAS scheduling algorithms. Specifically, Array Theory (AT) [22], Iterative-Optimization Modulo Theories (I-OMT) [23], and List Scheduling Per Link (LS-PL) [24] are configured as the comparison algorithms. In accordance with the specifications of the IEEE 802.1Qcc standard [25], we employ the shortest path routing algorithm to establish the routing path for each time-sensitive flow in the performance evaluation experiments.

5.2.1. Delay Performance Comparison

A comparative analysis of average end-to-end delay across different algorithms with changing wireless link transmission rates is presented in Figure 6. The values in Figure 6 are derived from a custom-designed simulation dataset for wired and wireless integrated satellite networks. To ensure experimental validity, the dataset was developed in accordance with IEEE 802.1Qcc and IEEE 802.1Qbv standards. Regarding the network topology setup, the intra-satellite wired topology adopts a hierarchical tree structure, while the inter-satellite wireless topology uses a mesh topology. The number of TSN-enabled wired terminals within each satellite is configured as 5–8. For traffic configuration, a total of 20 periodic time-sensitive flows are randomly generated. The frame size of each flow is randomly selected from the range of 1000–1500 bytes. The transmission period of each flow is randomly chosen from the range of 10–15 ms. All time-sensitive flows are transmitted bidirectionally between wired and wireless nodes. To ensure statistical robustness, we conduct 100 independent experiments for each algorithm across each transmission rate scenario. In the subsequent step, the end-to-end delay is measured by tracking the duration from the transmission of the initial bit of a frame at the source node to the successful reception of the final bit at the destination node. Ultimately, the average end-to-end delay across all flows is calculated.

The experimental results demonstrate that the delay of all algorithms exhibits a downward trend as the transmission rate increases. This is attributed to the fact that higher transmission rates lead to reduced transmission delay, which in turn decreases the end-to-end delay. The delay results of the four algorithms exhibit a distinct three-gradient hierarchy: AT belongs to the high-delay group, I-OMT to the medium-delay group, and LS-PL and ISA to the low-delay group. When the transmission rate rises from 250 Mbps to 1000 Mbps, the delay of AT decreases by roughly 8.3%, whereas that of the I-OMT experiences a reduction of around 3.4%. As evidenced by the results, the delay of AT and I-OMT decreases significantly; this phenomenon reveals that transmission rate exerts a substantial impact on their delay performance. In contrast, the delay variations of LS-PL and ISA across the full rate range are approximately 1.4% and 0.5%, respectively. This suggests that LS-PL and ISA possess superior robustness in the face of dynamic wireless link transmission rate. Nevertheless, ISA achieves a lower delay compared to LS-PL. In brief, the proposed ISA algorithm effectively supports deterministic communication with strict delay requirements.

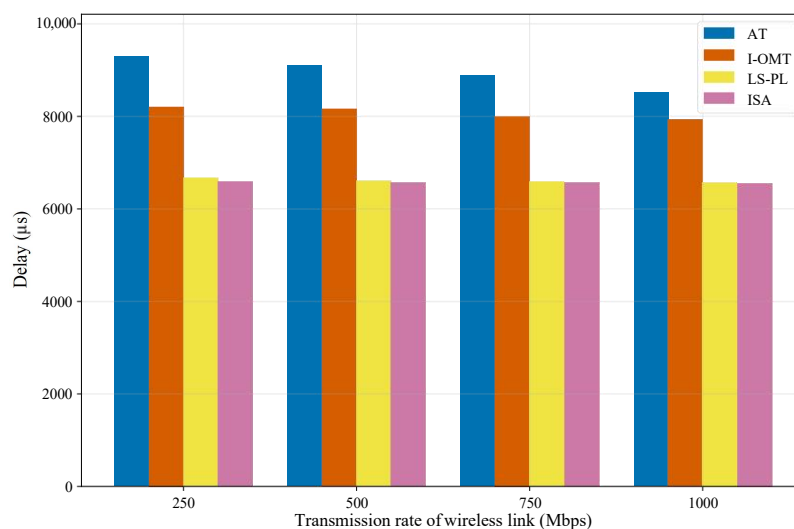


Figure 6. Average end-to-end delay under varied wireless rates.

5.2.2. Offset Performance Comparison

The offset of different algorithms across varying wireless link transmission rates is depicted in Figure 7. Within the scope of deterministic communication, the offset refers to the release time of time-sensitive traffic when it is dispatched from the source node. Among all evaluated algorithms, LS-PL achieves the highest offset, which is several times higher than that of the other algorithms. This is attributed to the fact that the calculation of the time offset adopts a reverse-order mechanism in LS-PL. Under the premise of strictly satisfying the deadline constraints of time-sensitive traffic, LS-PL intentionally transmits data packets as late as possible. Other algorithms determine the time offset following a forward-order mechanism. Therefore, LS-PL exhibits a significantly higher time offset compared to the other algorithms. Additionally, the time offset of LS-PL exhibits a slight upward trend with the increase in wireless link transmission rate. This is because a higher transmission rate corresponds to a larger time offset in LS-PL under the same deadline constraints. These observations reveal that LS-PL not only has extremely poor time offset management capability but also shows disadvantages in rate adaptability. The time offsets of AT and I-OMT are observed to fall within the intermediate range, with values significantly lower than those of LS-PL but considerably higher than those of ISA. Specifically, the time offset of I-OMT remains stable across varying transmission rates, while that of AT exhibits an inconsistent trend, resulting in weaker rate adaptability compared to I-OMT. The experimental results also show that ISA obtains the minimum time offset across all specified transmission rates. ISA enables time-sensitive traffic to be released from the source node as soon as possible, thereby reducing the resource occupancy time. Moreover, its performance does not exhibit significant fluctuations with variations in transmission rate, indicating excellent rate adaptability and stability.

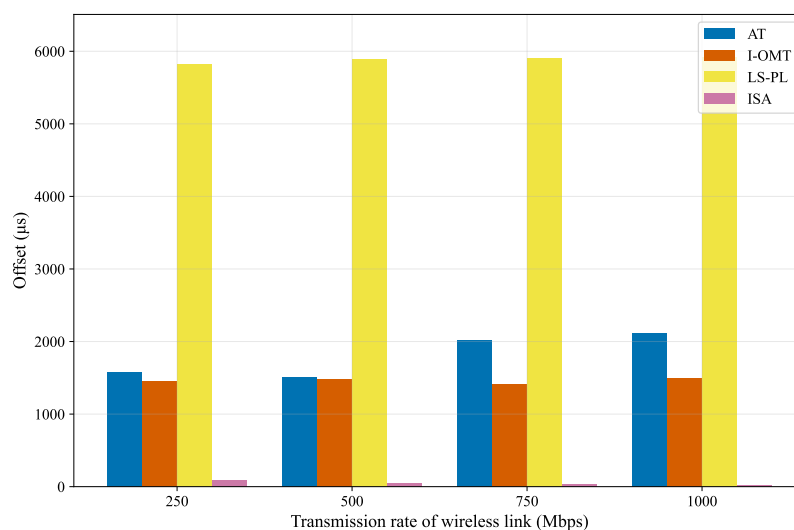


Figure 7. Offset under varied wireless rates.

5.2.3. Runtime Performance Comparison

As presented in Figure 8, the runtime of each algorithm is measured and compared under different wireless link transmission rates. In the context of this experiment, the runtime denotes the time consumed by the algorithm to converge to a valid scheduling solution that meets the predefined constraints. Subsequently, multi-dimensional comparisons are conducted to clarify the performance differences among these algorithms. In terms of time efficiency, AT is the most time-consuming algorithm. As the transmission rate increases from 250 Mbps to 1000 Mbps, the runtime of AT decreases from approximately 83 s to

68 s. This is because AT dynamically adjusts the scheduling window size by incrementally adding transmission time in accordance with practical transmission requirements. Consequently, a higher transmission rate enables faster adjustment to an appropriate window size, thereby shortening the runtime. In contrast, the runtime of I-OMT remains stable at around 20 s across all tested transmission rates, indicating that the runtime consumption of AT is three to four times that of I-OMT. Notably, the runtime overhead of AT exceeds that of both LS-PL and ISA by tens of times. The higher runtime of AT stems from its array theory-based modeling approach, which generates a constraint system that encounters combinatorial explosion when handling scheduling problems. Specifically, AT constructs a three-dimensional nested dictionary for its decision variables. Meanwhile, the combination of these high-dimensional decision variables and associated scheduling constraints further exacerbates the runtime. Regarding stability, the runtime fluctuation of AT with transmission rates reaches approximately 15 s, reflecting poor stability. In comparison, I-OMT exhibits strong stability, with a runtime fluctuation of less than 2 s across the entire rate range. Meanwhile, the runtimes of LS-PL and ISA exhibit almost no variation with changing transmission rates, demonstrating excellent robustness. From the perspective of rate sensitivity, the runtime of AT presents a slow downward trend with increasing transmission rates but with significant fluctuations, indicating high sensitivity to rate changes. The runtime of I-OMT is minimally affected by rate variations, while the runtimes of LS-PL and ISA are completely insensitive to rate fluctuations. Overall, LS-PL and ISA demonstrate significant advantages in both time efficiency and stability, making them highly suitable for time-sensitive missions; however, ISA has a shorter runtime compared to LS-PL, giving it a minor advantage in computational efficiency.

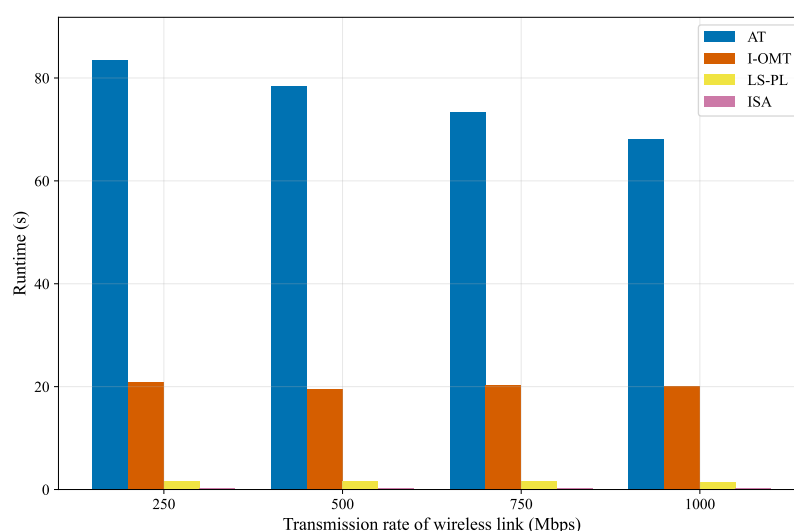


Figure 8. Runtime under varied wireless rates.

5.2.4. Memory Performance Comparison

The memory consumption of different algorithms under varied wireless link transmission rates is illustrated in Figure 9. The experimental results indicate that the algorithms do not exhibit significant fluctuations in memory consumption as the wireless link transmission rate increases. More precisely, across all tested scenarios, the memory consumption of AT stabilizes at approximately 160 KB, I-OMT at around 140 KB, LS-PL at roughly 105 KB, and ISA at about 72 KB. These numerical observations demonstrate that the memory consumption of each algorithm remains stable despite variations in wireless link transmission rates. Through a comparative analysis of memory consumption among all algorithms, AT incurs the highest memory consumption; compared to AT, I-OMT achieves a 12.5%

reduction in overhead, while LS-PL further reduces it by approximately 25% relative to I-OMT. Notably, ISA achieves the lowest memory consumption, representing an additional 31.4% reduction relative to LS-PL. To summarize, ISA outperforms the others in memory resource consumption, making it more suitable for deployment in resource-constrained application scenarios.

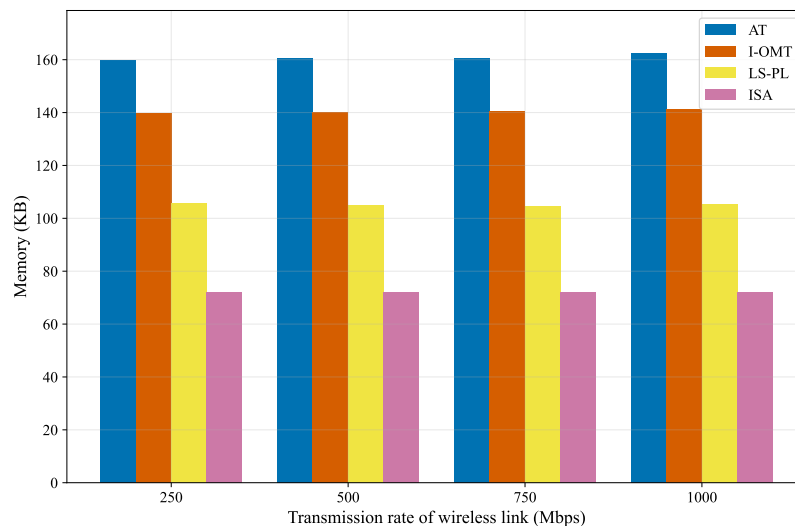


Figure 9. Memory consumption under varied wireless rates.

6. Conclusions

We proposed an integrated scheduling algorithm that incorporates principles from matching game theory, aiming at the problem of unstable delay caused by differentiated transmission characteristics of intra-satellite wired links and inter-satellite wireless links. First, we establish an integrated scheduling model that effectively quantifies the impact of scheduling time slots on the delay experienced at the satellite. Then, the established model is transformed into a cyclic three-sided matching game model to tackle the integrated scheduling problem. Subsequently, we design an integrated scheduling algorithm to obtain the time slot allocation results. Extensive experiments demonstrate that the proposed algorithm successfully achieves deterministic delay for time-sensitive traffic. In future work, we aim to investigate deterministic scheduling in large-scale satellite networks under massive traffic conditions.

Author Contributions: Conceptualization, Y.X. and Z.H.; methodology, Y.X., G.Z., and Z.H.; software, Y.X. and Z.H.; validation, Y.X.; formal analysis, Y.X. and Z.H.; investigation, Y.X.; resources, Y.X. and G.Z.; data curation, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, Y.X., G.Z., and Z.H.; visualization, Y.X. and Z.H.; supervision, G.Z.; project administration, Y.X., G.Z., and Z.H.; funding acquisition, G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 62171070).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

STINs	Space–Terrestrial Integrated Networks
LEO	Low-Earth Orbit
TSN	Time-Sensitive Networking
DetNet	Deterministic Networking
FPGA	Field-Programmable Gate Array
TAS	Time-Aware Shaper
TDMA	Time-Division Multiple Access
ISA	Integrated Scheduling Algorithm
STK	Satellite Tool Kit
TST	Time-Sensitive Traffic

References

1. Xiao, Z.; Yang, J.; Mao, T.; Xu, C.; Zhang, R.; Han, Z.; Xia, X.G. LEO Satellite Access Network (LEO-SAN) Toward 6G: Challenges and Approaches. *IEEE Wirel. Commun.* **2024**, *31*, 89–96. <https://doi.org/10.1109/MWC.011.2200310>.
2. Xie, H.; Zhan, Y.; Zeng, G.; Pan, X. LEO Mega-Constellations for 6G Global Coverage: Challenges and Opportunities. *IEEE Access* **2021**, *9*, 164223–164244. <https://doi.org/10.1109/ACCESS.2021.3133301>.
3. Messenger, J.L. Time-sensitive networking: An introduction. *IEEE Commun. Stand. Mag.* **2018**, *2*, 29–33. <https://doi.org/10.1109/MCOMSTD.2018.1700047>.
4. Nasrallah, A.; Thyagaturu, A.S.; Alharbi, Z.; Wang, C.; Shao, X.; Reisslein, M.; ElBakoury, H. Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 88–145. <https://doi.org/10.1109/COMST.2018.2869350>.
5. Tian, W.; Gu, C.; Guo, M.; He, S.; Kang, J.; Niyato, D.; Chen, J. Large-Scale Deterministic Networks: Architecture, Enabling Technologies, Case Study, and Future Directions. *IEEE Netw.* **2024**, *38*, 284–291. <https://doi.org/10.1109/MNET.2024.3355116>.
6. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. <https://doi.org/10.1109/TII.2018.2852491>.
7. Zambouri, K.; Noor-A-Rahim, M.; John, J.; Sreenan, C.J.; Poor, H.V.; Pesch, D. A Comprehensive Survey of Wireless Time-Sensitive Networking (TSN): Architecture, Technologies, Applications, and Open Issues. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 2129–2155. <https://doi.org/10.1109/COMST.2024.3486618>.
8. Garcia-Rodriguez, A.; Lopez-Perez, D.; Galati-Giordano, L.; Geraci, G. IEEE 802.11 be: Wi-Fi 7 strikes back. *IEEE Commun. Mag.* **2021**, *59*, 102–108. <https://doi.org/10.1109/MCOM.001.2000711>.
9. Seijo, Ó.; López-Fernández, J.A.; Val, I. w-SHARP: Implementation of a high-performance wireless time-sensitive network for low latency and ultra-low cycle time industrial applications. *IEEE Trans. Ind. Inform.* **2021**, *17*, 3651–3662. <https://doi.org/10.1109/TII.2020.3007323>.
10. Chaîne, P.J.; Boyer, M.; Pagetti, C.; Wartel, F. TSN support for quality of service in space. In Proceedings of the 10th European Congress on Embedded Real Time Software and Systems, Toulouse, France, 29–31 January 2020; pp. 1–11.
11. Sanchez-Garrido, J.; Aparicio, B.; Ramírez, J.G.; Rodríguez, R.; Melara, M.; Cercós, L.; Ros, E.; Diaz, J. Implementation of a time-sensitive networking (TSN) Ethernet bus for microlaunchers. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2743–2758. <https://doi.org/10.1109/TAES.2021.3061806>.
12. Peng, G.; Wang, S.; Li, G.; Huang, T.; Huang, Y.; Liu, Y. Space-Terrestrial Integrated Time-Sensitive Networks: Architecture, Challenges, and Open Issues. *IEEE Netw.* **2025**, *39*, 140–147. <https://doi.org/10.1109/MNET.2025.3551322>.
13. Ma, X.; Li, S.; Guan, Z.; Li, J.; Sun, H.; Wang, Y.; Guo, H. Time-Sensitive Networking Mechanism Aided by Multilevel Cyclic Queues in LEO Satellite Networks. *Electronics* **2023**, *12*, 1357. <https://doi.org/10.3390/electronics12061357>.
14. Wang, F.; Yao, H.; He, W.; Chang, H.; Xin, X.; Guo, S. Time-Sensitive Scheduling Mechanism Based on End-to-End Collaborative Latency Tolerance for Low-Earth-Orbit Satellite Networks. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 5149–5162. <https://doi.org/10.1109/TNSE.2023.3342938>.
15. Jiang, X.; Huang, Y.; Li, J.; He, H.; Chen, S.; Yang, F.; Yang, J. Spatio-temporal routing, redundant coding and multipath scheduling for deterministic satellite network transmission. *IEEE Trans. Commun.* **2023**, *71*, 2860–2875. <https://doi.org/10.1109/TCOMM.2023.3251360>.
16. Peng, G.; Wang, S.; Huang, T.; Li, F.; Zhao, K.; Huang, Y.; Xiong, Z. FastTS: Enabling Fault-Tolerant and Time-Sensitive Scheduling in Space-Terrestrial Integrated Networks. *IEEE J. Sel. Areas Commun.* **2024**, *42*, 3551–3565. <https://doi.org/10.1109/JSAC.2024.3459008>.

17. *IEEE Std 802.1Qbv-2015*; IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic. IEEE: Piscataway, NJ, USA, 2016; pp. 1–57. <https://doi.org/10.1109/IEEESTD.2016.8613095>.
18. Han, Z.; Xu, C.; Zhao, G.; Wang, S.; Cheng, K.; Yu, S. Time-varying topology model for dynamic routing in LEO satellite constellation networks. *IEEE Trans. Veh. Technol.* **2023**, *72*, 3440–3454. <https://doi.org/10.1109/TVT.2022.3217952>.
19. Stanescu, D.; Buzducea, A. An Overview of Satellite Link Budget Sensitivity Based on Digital Modulation Schemes in Multi-Orbit Satellite Networks. *Appl. Sci.* **2025**, *15*, 3473. <https://doi.org/10.3390/app15073473>.
20. Jia, Z.; Sheng, M.; Li, J.; Zhou, D.; Han, Z. Joint HAP access and LEO satellite backhaul in 6G: Matching game-based approaches. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 1147–1159. <https://doi.org/10.1109/JSAC.2020.3018824>.
21. *IEEE Std 802.1AS-2025*; IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications. IEEE: Piscataway, NJ, USA, 2025; pp. 1–491. <https://doi.org/10.1109/IEEESTD.2025.11302966>.
22. Serna Oliver, R.; Craciunas, S.S.; Steiner, W. IEEE 802.1Qbv Gate Control List Synthesis Using Array Theory Encoding. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*; IEEE: Piscataway, NJ, USA, 2018; pp. 13–24. <https://doi.org/10.1109/RTAS.2018.00008>.
23. Jin, X.; Xia, C.; Guan, N.; Xu, C.; Li, D.; Yin, Y.; Zeng, P. Real-Time Scheduling of Massive Data in Time Sensitive Networks With a Limited Number of Schedule Entries. *IEEE Access* **2020**, *8*, 6751–6767. <https://doi.org/10.1109/ACCESS.2020.2964690>.
24. Bujosa, D.; Ashjaei, M.; Papadopoulos, A.V.; Nolte, T.; Proenza, J. HERMES: Heuristic multi-queue scheduler for TSN time-triggered traffic with zero reception jitter capabilities. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 70–80. <https://doi.org/https://doi.org/10.1145/3534879.3534906>.
25. *IEEE/ISO/IEC 8802-1Q-2020/Amd31-2021*; IEEE/ISO/IEC International Standard-Telecommunications and Exchange Between Information Technology Systems—Requirements for Local and Metropolitan Area Networks—Part 1Q: Bridges and Bridged Networks AMENDMENT 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. IEEE: Piscataway, NJ, USA, 2021; pp. 1–211. <https://doi.org/10.1109/IEEESTD.2021.9599625>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.