Routing and Scheduling for Low Latency and Reliability in Time-Sensitive Software-Defined IIoT

Luyue Ji, *Student Member, IEEE,* Shibo He, *Senior Member, IEEE,* Chaojie Gu, *Member, IEEE,*
Zhiguo Shi, *Senior Member, IEEE* and Jiming Chen, *Fellow, IEEE*

***Abstract*—Time-Sensitive Software-Defined Networking (TSSDN) is an emerging technology that combines the real-time network configuration capabilities of Software-Defined Networking (SDN) with the deterministic flow delivery capabilities of Time-Sensitive Networking (TSN), making it ideal for use in the industrial Internet of Things (IIoT). However, as data flows generated by industrial applications grow exponentially, it is challenging to achieve low-latency and reliable data flow transmission at the same time in TSSDN due to the limited network resources. To address this issue, we propose the adoption of the Frame Replication and Elimination for Reliability (FRER) mechanism in TSSDN-based IIoT systems. However, it is important to acknowledge that the FRER mechanism introduces stress on the already restricted network resources by generating redundant paths. In light of this concern, we construct an end-to-end delay bound model and a reliability model to analyze this issue. To mitigate the stress imposed on the network, we formulate an optimization problem for maximizing the overall system utility while adhering to the transmission requirements of business flows and the limitations of hardware resources. Consequently, we devise an algorithm for reliability-enhanced flow routing and scheduling, which effectively solves the aforementioned optimization problem. To validate the effectiveness and performance of our proposed algorithm, we conduct numerical simulations on four datasets. The results demonstrate the superior performance of our approach compared to existing methods.**

***Index Terms*—Industrial Internet of Things, Time-Sensitive Software-Defined Networking, routing and scheduling, Frame Replication and Elimination for Reliability, network calculus**

I. Introduction

The industrial Internet of Things (IIoT) plays a pivotal role in facilitating intelligent manufacturing. It has been successfully implemented across diverse sectors such as energy, transportation, and healthcare [1]. The successful implementation of IIoT across these sectors has resulted in the generation of data flows of varying volumes and frequencies, each with distinct Quality of Service (QoS) requirements. Consequently, two major challenges arise from this scenario. Firstly, it becomes crucial to ensure the efficient transmission of these multiple data flows while accommodating their specific QoS requirements. Secondly, the IIoT system must possess dynamic network configuration capabilities in order to adapt to rapidly changing business demands and maintain competitiveness in dynamic markets. In order to address the aforementioned challenges, researchers have proposed a solution known as Time-Sensitive Software-Defined Networking (TSSDN). TSSDN leverages the real-time network configuration capability of Software-Defined Networking (SDN) and utilizes QoS guarantees offered by Time-Sensitive Networking (TSN) [2].

Current studies on TSSDN have primarily focused on leveraging specific TSN standards, such as IEEE 802.1Qbv, to enhance network transmission efficiency [3], [4]. However, in practical industrial manufacturing processes, devices may encounter reliability issues due to environmental factors, e.g., temperature, air quality, and static electricity. The network reliability issue in TSSDN has not been thoroughly investigated. To bridge this gap, we propose utilizing IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) standard, part of TSN, to strengthen data transmission reliability and fault tolerance in TSSDN-based IIoT systems. FRER achieves this by providing redundant paths through replicating data frames from the source node and transmitting multiple copies to the destination node, each with a unique sequence number [5]. When the node receives multiple identical data frames, it compares the sequence numbers of each frame to eliminate redundant frames and retain the correct one, thereby improving data reliability and reducing the risk of errors or data loss. While the redundant path provided by the FRER mechanism enhances communication reliability, it also consumes additional bandwidth resources, potentially impacting transmission latency. Therefore, achieving an optimal balance between high reliability and real-time performance is crucial to ensure efficient operations of IIoT systems.

To address the aforementioned issues, in this work, we first employ network calculus (NC) [6] to model the worst-case end-to-end delay bound for data flows in the TSSDN. Secondly, since the FRER mechanism generates excessive redundant data, wasting network bandwidth, there is intense resource competition between flow transmission latency and reliability in the system. We tackle this issue by formulating a total system utility optimization problem to balance the latency and reliability of flows, subject to different transmission requirements of business flows and hardware resource constraints. Finally, we propose a heuristic algorithm for flow routing and scheduling to solve the system utility optimization problem.

We validate the proposed approach with three public datasets and a randomly generated dataset. The experimental results demonstrate that our approach outperforms existing algorithms in multiple aspects, including higher access rates, greater total system utility, and faster runtime. To the best of our knowledge, this is the first work that optimizes flow

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant U1909207, and in part by NSFC under Grant 62302439. *Corresponding author: Jiming Chen.*

L. Ji, S. He, C. Gu, Z. Shi, and J. Chen, Zhejiang University, Hangzhou, Zhejiang, 310027, China. E-mail: {july_fish, s18he, gucj, shizg, cjm}@zju.edu.cn.

We acknowledge the invaluable assistance and guidance provided by Renlang Huang and Liang Li in the design of the learning-based baseline in Section V.

routing and scheduling considering latency and reliability of industrial business flows simultaneously in TSSDN-based IIoT systems. The main contributions of this article can be summarized as follows.

- We design an approach to effectively guarantee the simultaneous low-latency and reliable transmission of multiple business flows in TSSDN based on the FRER mechanism. Specifically, we establish the end-to-end delay bound and the reliability model to understand the additional cost introduced by the FRER.
- We formulate a total utility optimization problem subject to the resource constraints and QoS requirements of multiple business flows, taking into consideration the trade-off between latency and reliability requirements.
- We propose a heuristic algorithm for determining network resource availability and dynamically computing the routing and scheduling strategies for business flows. The proposed algorithm outperforms existing approaches.

The rest of this article is organized as follows. Section II reviews related literature. Section III presents the system model. Section IV formulates the optimization problem and designs a heuristic algorithm. Section V evaluates the performance of the proposed algorithm. Section VI discusses the potential challenges. Section VII concludes the article.

II. Related work

The current research on data transmission optimization in IIoT can be broadly classified into two categories: latency-oriented performance optimization using TSN and SDN, and reliable routing and fault-tolerant method.

Exploiting TSN and SDN to reduce latency has attracted increasing research interest. Thomas et al. [7] provide a method of worst-case timing analysis for TSN, which provides a NC toolbox for bounding the burstiness increase caused by the elimination function of duplicate packets. Ji et al. [8] propose an SDN-based IIoT architecture and model the worst-case end-to-end delay of IIoT business flows based on NC. Gerhard et al. [9] present a Software-defined Flow Reservation approach by combining the SDN and TSN to support data flows with real-time requirements. Montazerolghaem et al. [10] propose an innovative framework leveraging SDN to effectively meet the QoS demands of diverse IoT services while simultaneously ensuring balanced traffic among IoT servers. Additionally, they [11] investigate an energy-efficient and load-balanced resource management approach within software-defined Internet of Multimedia Things networks, substantiated by the implementation of a test platform. Considering the impact of the combinability of multiple flows on scheduling performance, a noncollision theory based deterministic scheduling method is presented [12] to achieve ultra-low-latency communication in TSN. Böhm et al. [13] introduce TSSDN to achieve both deterministic and non-deterministic communication and implement a prototype and a testbed to evaluate the effectiveness of TSSDN. Furthermore, in order to handle the high complexity and dynamic network scenarios, Tang et al. [14] investigate the learning-based methods for intelligent end-to-end communication. For example, Xu et al. [15] propose a learning-based scheduling and routing co-design architecture based on a stream partition method to guarantee deterministic and real-time transmission in TSN. Naeem et al. [16] propose a novel model-free adaptive actor-critic deep reinforcement learning framework that incorporates a fuzzy normalized neural network within software-defined IoT networks.

Others are dedicated to the reliable data transmission and fault-tolerant method in IIoT, including single-path routing, multi-path routing, and retransmission-based mechanisms. Gao et al. [17] propose a reliable routing decision scheme based on the probabilistic model for various transmission ranges. Lenka et al. [18] present a rendezvous-based routing protocol through clustering and multi-path techniques to increase the reliability of IoT networks. Ergenç et al. [19] propose a reassurance-based path selection method that improves the fault-tolerance of TSN and reduces the unintentional elimination of packets caused by non-disjoint paths under FRER mechanism. Naushad et al. [20] present a multipath routing protocol that employs hierarchical clustering to enhance both reliability and load balancing. Maile et al. [21] present solutions to the fundamental limitations of FRER to enhance reliability through choosing a match versus vector recovery algorithm and setting a timer to reset the sequence history. Shi et al. [22] propose an automatic on-demand retransmission scheme to deliver the monitoring data generated by field devices with high reliability and low latency.

The current research on network transmission optimization within IIoT networks primarily concentrates on individual metrics, such as enhancing real-time performance, increasing reliability, or improving energy efficiency, without comprehensively addressing the conflicts and trade-offs among different network metrics. Hence, our focus is on investigating resource competition issues between low latency and reliability in IIoT networks, considering the transmission demands of real industrial networks and the diversity of network services. Specifically, we use network calculus theory to analyze the queuing delays of business flows. Unlike conflict-free scheduling methods, network calculus calculates worst-case network delays without requiring network clock synchronization. Additionally, existing multi-path routing works often employ the maximum-disjoint mechanism to select redundant paths, lacking theoretical analysis of network reliability. To address this gap, we implement the multi-path flow routing and scheduling by modeling a reliability probability, which is theoretically grounded and more rigorous.

III. System model

A. System overview

Fig. 1 presents a typical architecture of the TSSDN-empowered IIoT system, which encompasses a range of heterogeneous devices, including sensors, actuators, robots, cameras, and automatic guided vehicles (AGVs). Throughout the industrial production process, IIoT devices continually generate industrial data that reflects the operation and status of the production line. These industrial data flows traverse the TSSDN network to the application plane.

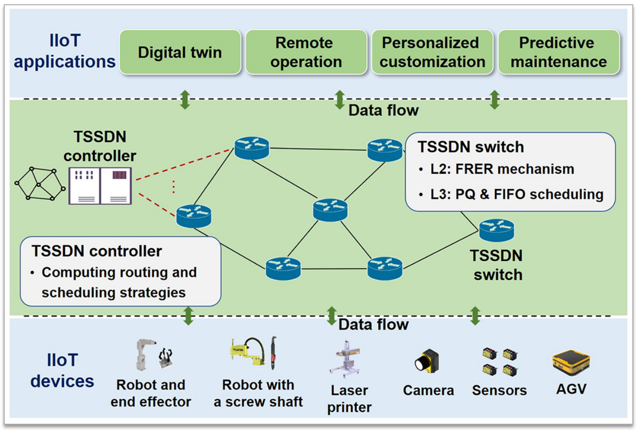


Fig. 1: Illustration of Time-Sensitive Software-Defined industrial Internet of Things architecture.

Industrial applications entail specific QoS requirements for their business flows. For example, remote operation [23] often requires rapid response and execution, so data flows need to be rapidly transmitted to ensure real-time performance. In addition, remote operation requires ensuring the continuous availability of data, and maintaining the stability and accuracy of data transmission even in the event of network or equipment failure. These transmission requirements exert substantial pressure on the limited communication resources. Therefore, TSSDN and FRER are adopted for routing and scheduling optimization to achieve low-latency and high-reliability transmission in remote operation scenario. The communication network of the system consists of a TSSDN controller and multiple TSSDN switches. The TSSDN controller computes routing and scheduling strategies while the switch manages transmission collisions. The collision occurs when two or more data flows simultaneously require forwarding at the same switch egress. We jointly exploit Priority queuing (PQ) and first-in-first-out (FIFO) scheduling to manage conflicts among business flows. Specifically, the TSSDN controller assigns corresponding queues to business flows based on their priority. For business flows with the same priority, they enter the queues according to the FIFO rule. As shown in Fig. 2, FRER is exploited to ensure the reliability requirements of business flows through redundant paths. Each business flow is replicated at its source switch and transmitted redundantly over different paths to enhance reliability.

B. *End-to-end delay bound model*

TSSDN is abstracted into a network graph ($\mathcal{G}$) with numerous TSSDN switches. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of switch vertices $\mathcal{V} = \{v_1, v_2, ...\}$, and a set of edges $\mathcal{E} = \{e_1, e_2, ...\}$. Each edge e_k denotes a physical link from the switch u to v identified by an ordered pair (u, v), $\forall u, v \in \mathcal{V}$ and $u \neq v$. The data flow set $\mathcal{F} = \left\{f_1, ..., f_i, ..., f_{|\mathcal{F}|}\right\}$ is composed of $|\mathcal{F}|$ data flows from industrial applications, and $N = |\mathcal{F}|$ denotes the number of flows. A 5-tuple is defined to describe a data flow f_i, i.e., $(s_i, d_i, r_i, b_i, p_i)$, which respectively represent the source node, destination node, arrival rate, burst size, and frame priority. The frame priority determines the transmission requirement $Req_i = \left(Req_i^{lat}, Req_i^{rel}\right)$, where Req_i^{lat} and Req_i^{rel} denote the requirements of latency and reliability, respectively. According to NC, the arrival curve of business flow is represented by $\alpha_i(t) = r_i t + b_i$, and such flows are referred to as α-constrained flows. Table I summarizes notations and abbreviations used in this paper.

TABLE I: List of Notations and Abbreviations

Notation/Abbr.	Definition
$R_{u,v}$	Capacity of the link(u, v)
B_u	Buffer capacity of node u
R_j	Reliability of node v_j
Req_i^{lat}	Delay bound requirement of flow f_i
Req_i^{rel}	Reliability requirement of flow f_i
K	Redundancy of flow f_i
w_i	Weight factor to measure the priority of flow f_i
δ	Weight of balancing the network utility and cost
ϕ	Proportion of latency utility to network utility
φ	Proportion of bandwidth cost to network cost
r_i	Arrival rate of flow f_i
b_i	Burst size of flow f_i
L^{max}	Maximum packet size of flows
τ_{pro}	Process delay
τ_{con}	Constant delay
T_{f_i}	End-to-end delay of flow f_i
R_{f_i}	Reliability of flow f_i
TSSDN	Time-Sensitive Software-Defined Networking
FRER	Frame Replication and Elimination for Reliability
NC	Network calculus
MTSU	Maximize the total system utility
RFRSA	Reliability-enhanced flow routing and scheduling algorithm

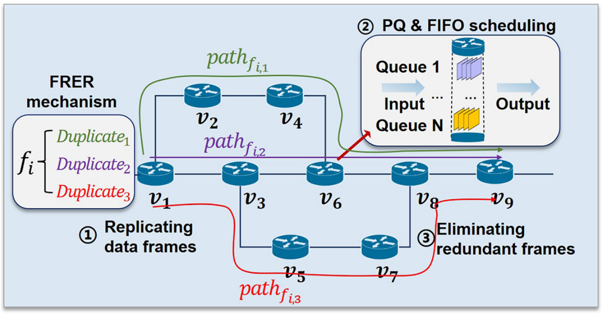


Fig. 2: The communication network leverages FRER, PQ, and FIFO to ensure low latency and reliability.

In TSSDN, when a business flow enters the network, the controller initially evaluates the capability of the current network resources to meet the transmission requirements of that specific business flow. Subsequently, the controller generates appropriate routing and scheduling strategies and deploys the corresponding forwarding policies to the TSSDN switches. A virtual link (u, v, q) is defined to represent the qth virtual queue of the physical link (u, v). If the kth duplicate of the business flow f_i is allocated to the virtual link (u, v, q), the indicator variable $x_{i,k}(u, v, q) = 1$. Otherwise, $x_{i,k}(u, v, q) = 0$. We define the $f_{i,k}$ as kth duplicate of the business flow f_i for convenience. For instance, considering the duplicated flow $f_{i,2}$ shown in Fig. 2 with the path $path_{f_{i,2}} = (v_1, v_3, v_6, v_8, v_9)$, the indicator variables $x_{i,2}(v_1, v_3, 1) = 1$, $x_{i,2}(v_3, v_6, 1) = 1$, $x_{i,2}(v_6, v_8, 2) = 1$, and $x_{i,2}(v_8, v_9, 2) = 1$ are specifically set, while all other indicator variables associated with $f_{i,2}$ are set to 0. The bandwidth and buffer used on the virtual link (u, v, q) are $r_{u,v,q} = \sum_{i=1}^{N} \sum_{k=1}^{K} x_{i,k}(u, v, q)\, r_i$ and $b_{u,v,q} = \sum_{i=1}^{N} \sum_{k=1}^{K} x_{i,k}(u, v, q)\, b_i$, respectively, where K is the redundancy degree of the business flow.

The end-to-end delay bound of the kth duplicate of the business flow f_i can be expressed as

$$d_{i,k} = \sum_{(u,v,q)\in path_{i,k}} \left(\frac{\sum\limits_{j=1}^{q_{i,k}-1} b_{u,v,j} + 2L_i^{\max}}{R_{u,v} - \sum\limits_{j=1}^{q_{i,k}-1} r_{u,v,j}} + \tau_{pro} \right) + \frac{\sum\limits_{f_j \in equal_{i,k}} b_j}{\min\limits_{(u,v,q)\in path_{i,k}} \left\{ R_{u,v} - \sum\limits_{j=1}^{q_{i,k}-1} r_{u,v,j} \right\}}, \tag{1}$$

according to calculated results in [8], where $\sum\limits_{j=1}^{q_{i,k}-1} r_{u,v,j}$ represents total bandwidth used by flows with higher priority than $f_{i,k}$, and $\sum\limits_{j=1}^{q_{i,k}-1} b_{u,v,j}$ represents the maximum burst size that this queue may have buffered. Additionally, $L_i^{\max}$ denotes the maximum packet length of flow f_i, and $R_{u,v}$ represents the bandwidth capacity of the physical link (u, v). The set $path_{i,k}$ denotes the path of flow $f_{i,k}$, and the set $equal_{i,k}$ contains all duplicate flows that are selected with the same virtual link (u, v, q) as $f_{i,k}$. Hence, the worst-case end-to-end delay bound of flow f_i is expressed as

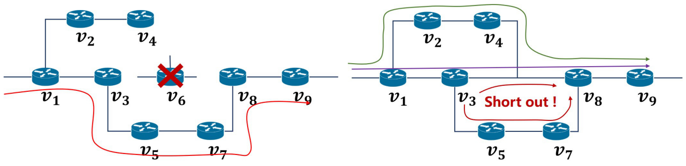
$$T_{f_i} = \begin{cases} \max\limits_{k} (d_{i,k}) + \tau_{con}, & f_i \text{ is transmitted reliably} \\ +\infty, & f_i \text{ is transmitted unsuccessfully} \end{cases}, \tag{2}$$

where τ_{con} is the constant delay resulting from the packet propagation and processing.

C. Reliability model

We define the lifetime of a component as a random variable T, and the probability density function is $f(t) = \frac{dF(t)}{dt}$, where the $F(t)$ is the cumulative distribution function and represents the probability that the component will fail at or before time t. The reliability of a component is expressed as $R(t) = Prob\{T > t\} = 1 - F(t)$. Hence, the failure rate at time t is a conditional probability, which is given by $\lambda(t) = \frac{f(t)}{1-F(t)}$. Substituting $\frac{dR(t)}{dt} = -f(t)$, we obtain $\lambda(t) = -\frac{dR(t)}{R(t)dt}$.

The probability of a hardware component failing over time can be represented by a bathtub curve [24]. The failure rate is initially high during the "infant mortality" phase, as any manufacturing defects or weaknesses in the component are more likely to become apparent. As the component ages, the failure rate gradually decreases and approaches a constant level. Once the component reaches the "wear-out" phase, the failure rate increases again. We are focusing on the failure rate of industrial components during their mature period. Assuming that the failure rate of these components is constant, i.e., $\lambda(t) = \lambda$, and then the reliability is calculated as $R(t) = e^{-\lambda t}$, which means that the reliability $R(t)$ has an exponential distribution[1]. In this case, the probability density function of a component lifetime is $f(t) = \lambda e^{-\lambda t}$.



(a) When v_6 is broken.

(b) When v_6 is closed.

Fig. 3: The converted network topologies in two situations.

Network topology typically consists of two fundamental structures: series and parallel connections, which also apply to the interconnection of various IIoT devices. In a serial system, the reliability of the entire system depends on the reliability of all switches, because if any component fails, the entire system will be affected. Assuming that there are M switch nodes in the serial system, and the failure rate of each node is independent and represented by $R_i(t)$, the failure rate of the serial system can be expressed as $R_s(t) = \prod\limits_{i=1}^{M} R_i(t) = e^{-(\lambda_1+\lambda_2+\ldots+\lambda_M)t}$. In a parallel system, the entire system can operate normally as long as any one of the switches works properly. Assuming that failure rates between switches are independent, the reliability of the entire parallel system is expressed as $R_p(t) = 1 - \prod\limits_{i=1}^{M} (1 - R_i(t)) = 1 - \prod\limits_{i=1}^{M} \left(1 - e^{-\lambda_i t}\right)$.

With FRER, we replicate the flow into K duplicates and assign different routing paths to each duplicate. Nevertheless, the reliability model for the parallel system cannot be used directly. Because there are crossing nodes between different paths, the failure rates of redundant paths are not independent of each other. Therefore, we convert multiple redundant paths into one path through the series or parallel relationship between paths and then calculate the reliability of the converted paths.

For instance, according to the network topology with a redundancy of 3 in Fig. 2, we obtain $path_1 = (v_1, v_2, v_4, v_6, v_8, v_9)$, $path_2 = (v_1, v_3, v_6, v_8, v_9)$, and $path_3 = (v_1, v_3, v_5, v_7, v_8, v_9)$. We classify and discuss the working status of v_6. As shown in Fig. 3, when node v_6 fails, there is only $path_3 = (v_1, v_3, v_5, v_7, v_8, v_9)$ left. The current system reliability is calculated as $R_{f_i}(t) = R_1(t)R_3(t)R_5(t)R_7(t)R_8(t)R_9(t)$. When node v_6 is operational, part of $path_3$ is "short-circuited". Therefore, we can remove $path_3$ from the system, resulting in a revised system reliability formula of $R_{f_i}(t) = R_1(t)\left[1 - [1 - R_2(t)R_4(t)]\,[1 - R_3(t)]\right] R_8(t)R_9(t)$. In conclusion, the reliability of business flow f_i is

$$\begin{aligned} R_{f_i}(t) = {} & R_j Prob\{\text{System works} \mid v_j \text{ is working}\} \\ & + (1 - R_j)\, Prob\{\text{System works} \mid v_j \text{ is faulty}\}, \end{aligned} \tag{3}$$

where R_j denotes the reliability of node v_j.

[1] There are situations where this assumption of a constant failure rate is inappropriate, especially in the infant mortality and wear-out phases of a component's life. In such cases, the Weibull distribution is generally used.

IV. Formulation and optimization

A. Problem formulation

Ensuring the low-latency and reliable transmission of critical business flows requires significant bandwidth and buffer resources, straining the limited system resources. To improve resource utilization efficiency, it is essential to increase network utility and decrease network costs while meeting the requirements for critical business flow transmission. Therefore, our optimization objective is to maximize the total system utility (MTSU), which is the difference between the network utility and network cost, denoted by U_{f_i} and C_{f_i}, respectively. In addition, the parameter $\delta \in (0,1)$ is defined as the weight of balancing the network utility and cost, which can be tailored to meet the specific requirements of the user and the application. Meanwhile, the weight factor $w_i \in (0,1)$ differentiates flows with varying priorities. Generally, weight factors are sorted in descending order of priority (i.e., $w_1 \geq w_2 \geq ...$), ensuring that higher priority flows get more attention. The MTSU problem is formulated as

$$\textbf{OPT}: \max_{x_{i,k}} \sum_{f_i} w_i \left[\delta U_{f_i} - (1-\delta) C_{f_i}\right] \tag{4}$$

subject to

$$\sum_{q=1}^{Q} x_{i,k}(u,v,q) \leq 1, \forall x_{i,k}(u,v,q) \in \{0,1\} \tag{5}$$

$$\sum_{u \in \mathcal{V}} \sum_{q=1}^{Q} \left[x_{i,k}(u,v,q) - x_{i,k}(v,u,q)\right] = \begin{cases} -1, & \text{if } v = s_i \\ 1, & \text{if } v = d_i \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

$$\sum_{i=1}^{N} \sum_{q=1}^{Q} x_{i,k}(u,v,q)\, r_i \leq R_{u,v}, \forall (u,v) \in \mathcal{E} \tag{7}$$

$$\sum_{i=1}^{N} \sum_{q=1}^{Q} x_{i,k}(u,v,q)\, b_i \leq B_u, \forall u \in \mathcal{V} \tag{8}$$

$$T_{f_i} \leq Req_i^{lat}, \forall f_i \in \mathcal{F} \tag{9}$$

$$R_{f_i} \geq Req_i^{rel}, \forall f_i \in \mathcal{F}. \tag{10}$$

The feasible region for optimization variable $x_{i,k}(u,v,q)$ is given by constraints (5)-(10). Constraint (5) ensures that each duplicate flow on link (u,v) is assigned to, at most, one virtual queue. Constraint (6) enforces flow conservation at each node by requiring that the total inflow be equal to the total outflow, except for the source node s_i and destination node d_i. Constraints (7) and (8) respectively limit the occupied bandwidth and buffer not to exceed the capacity of link (u,v) and node u. It prevents congestion and ensures that packets are not dropped due to insufficient buffer space. Constraint (9) limits the worst-case end-to-end delay bound for each flow. Constraint (10) imposes a lower bound on the probability of successful delivery for each flow.

Algorithm 1 Algorithm of computing the reliability bound

```
Input: $PATH_{f_i}$: Path set of $K$ duplicates of flow $f_i$, specifically, $PATH_{f_i} = \left\{path_{f_{i,1}}, ..., path_{f_{i,k}}, ..., path_{f,K}\right\}$;
Output: $R^{lower}_{f_{i,k}}$: Reliability lower bound of flow $f_i$.
Reconstruct a graph $G_{f_i}$ based on the number of nodes and successive connection states of the path set $PATH_{f_i}$;
Compute the minimal cut set $S_{cut} = \{S_{cut,1}, ..., S_{cut,m}, ..., S_{cut,M}\}$ of Graph $G_{f_i}$;
$Fail_{cut,m} = 1$, $R^{lower}_{f_{i,k}} = 1$;
for all $S_{cut,m} \in S_{cut}$ do
    for all $v_j^m \in S_{cut,m}$ do
        $Fail_{cut,m} = Fail_{cut,m} * \left(1 - R(v_j^m)\right)$;
    end for
    $R^{lower}_{f_{i,k}} = R^{lower}_{f_{i,k}} * (1 - Fail_{cut,m})$
end for
return $R^{lower}_{f_{i,k}}$
```

In **OPT**, network utility consists of the latency utility and reliability utility, denoted by $U_{f_i}^{lat}(T_{f_i})$ and $U_{f_i}^{rel}(R_{f_i})$, respectively. The function $U_{f_i}^{lat}(\cdot)$ represents the latency utility function, which measures the impact of end-to-end delay on the network utility. This function is negatively correlated with the end-to-end delay bound, meaning that $U_{f_i}^{lat}(\cdot)$ is a monotonically decreasing function. Similarly, the function $U_{f_i}^{rel}(\cdot)$ represents the reliability utility function which is positively correlated with reliability and is hence a monotonically increasing function. The shape of these functions depends on the type and scenario of the application. The specific parameter settings of functions is given in §V. The network utility is calculated as

$$U_{f_i} = \phi U_{f_i}^{lat}(T_{f_i}) + (1-\phi) U_{f_i}^{rel}(R_{f_i}), \tag{11}$$

where $\phi \in (0,1)$ represents the weight factor used to adjust the proportion of latency utility and reliability utility. Correspondingly, the network cost can be decomposed into two components: the link bandwidth cost represented by $C_{f_i}^{ban}$, and the node buffer cost represented by $C_{f_i}^{buf}$. The network cost is calculated as

$$\begin{aligned} C_{f_i} &= \varphi C_{f_i}^{ban} + (1-\varphi) C_{f_i}^{buf} \\ &= \varphi r_i \sum_{(u,v) \in \mathcal{E}} \sum_{k} x_{i,k}(u,v,q) \\ &\quad + (1-\varphi) b_i \sum_{v \in \mathcal{V}} \sum_{k} x_{i,k}(u,v,q), \end{aligned} \tag{12}$$

where the link bandwidth cost $C_{f_i}^{ban}$ and node buffer cost $C_{f_i}^{buf}$ respectively refer to the total amount of bandwidth and buffer used by all the business flows accessing the network. And $\varphi \in (0,1)$ represents the weight factor to adjust the proportion of bandwidth and buffer cost.

B. Reliability-enhanced flow routing and scheduling

The MTSU problem is a flow routing and scheduling problem that involves flow access control, routing, and scheduling. When a business flow requests access to the network, the TSSDN controller checks if there are enough network resources to meet the QoS requirements of this business flow. After access is granted, the controller configures routing and scheduling policies for this flow. The routing and scheduling problem is NP-hard [25], resulting in our MTSU problem also being an NP-hard problem. In this article, we propose a heuristic algorithm, a reliability-enhanced flow routing and scheduling algorithm (RFRSA), to maximize the total system utility subject to the latency and reliability requirements of the business flows and bandwidth and buffer resource constraints based on natural aggregation algorithm (NAA) [26]. Before introducing RFRSA, two preparatory works are introduced.

1) Dimension reduction of variables: In the MTSU problem, there are a large number of indicator variables ($x_{i,k}\left(u,v,q\right)$). A dimension reduction method is proposed to reduce the original variables to fewer variables. We construct two vectors denoted as $X_{i,k}$ and $Y_{i,k}$, representing the routing and scheduling strategies, respectively. The vector $X_{i,k}$ is $(H+1)-$ dimensional, while the vector $Y_{i,k}$ is $(H-1)-$ dimensional. The constant H represents the maximum number of hops. The first H elements of $X_{i,k}$ represent the node indices assigned to the duplicate flow $f_{i,k}$, where a value of 0 indicates that no node has been assigned. The last element of $X_{i,k}$ indicates the path length. As an illustration, for a given flow $f_{i,2}$ in Fig. 2, suppose $H=8$, we have $X_{i,2}=(1,3,6,8,9,0,0,0,5)$ and $Y_{i,2}=(1,1,2,2,0,0,0)$. Thus, the number of variables is reduced from $N*V*(V-1)*Q*K$ to $N*K$ variables with $2H$ dimensions. This approach significantly reduces the search space and time.

2) Evaluation of the reliability of paths: The reliability of paths is a critical factor in the MTSU problem. However, it is difficult to use the reliability formula (Eq. (3)) directly due to its complexity. To address this issue, an alternative solution based on the minimal cut set of the network topology is proposed. A minimal cut set [24] is the minimal list of modules where removing all modules from the list due to faults or failures. The lower bound of the reliability of the path can be calculated by $R_{f_i}^{lower}=\prod\limits_{m=1}^{M}R_{cut,m}$, where $R_{cut,m}$ represents the reliability of the minimal cut set and M represents the number of minimal cut sets. Note that when the reliability of individual nodes exceeds 99%, the lower bound of reliability can serve as a good estimate for the overall system reliability [24]. In industrial scenarios, IIoT nodes generally satisfy the conditions for utilizing the lower bound of reliability. Algorithm 1 presents the detailed reliability calculation process.

Algorithm 2 is our proposed algorithm. Firstly, the business flows are sorted according to their priorities (line 1). Then, the NAA algorithm is invoked to compute the optimal path based on the available bandwidth and buffer resources (lines 4-7). If there is no feasible path, the business flow is rejected from entering the network (lines 8-10). On the other hand, if a feasible path exists, the end-to-end delay bound of the business flow is calculated using equations (1) and (2), and the lower bound of the path's reliability is determined by invoking Algorithm 1 (lines 11-12). If the path fails to meet the specified latency and reliability requirements, the current path is removed. Conversely, if the requirements are met, the current path is deemed to be the optimal solution, and the business flow is added to the queue with the highest priority. In addition, the available resource set is updated based on the resources occupied by the current path (lines 13-18). The time complexity of this algorithm is $O(N*(H+1)*P*I)$, where P is the population size and I is the number of iterations in the NAA.

Algorithm 2 Reliability-enhanced flow routing and scheduling algorithm (RFRSA)

Input: Set of business flows $\mathcal{F}$, physical network $\mathcal{G}=(\mathcal{V},\mathcal{E})$, available bandwidth matrix Res_{band} and available buffer matrix Res_{buff};

Output: Optimal routing strategy $X_{i,k}^{*}$, optimal scheduling strategy $Y_{i,k}^{*}$, maximum network utility $Utility^{*}$.

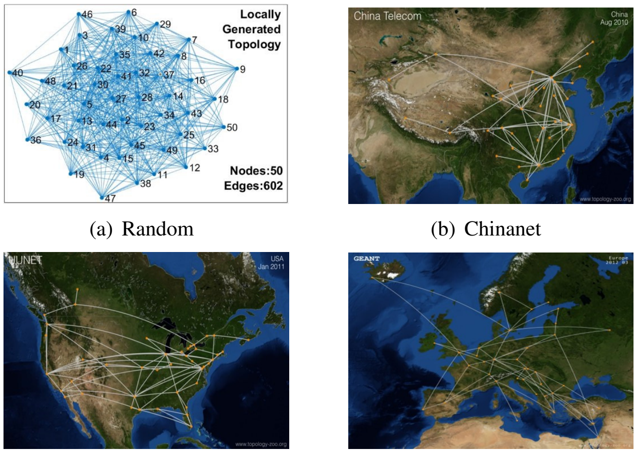
```
1:  Sort flows in $\mathcal{F}$ in ascending order of priority.
2:  for all $f_i \in \mathcal{F}$ do
3:      $Flag = 0$;
4:      for all $f_{i,k} \in f_i$ do
5:          Compute the optimal $X_{i,k}^{*}$ by invoking NAA, subject
            to the constraints (5)-(8), based on $Res_{band}$ and
            $Res_{buff}$;
6:          If $X_{i,k}^{*}\ = 0$, then $Flag = 1$;
7:      end for
8:      if $Flag = 0$ then
9:          Reject the flow $f_i$;
10:     end if
11:     Compute the delay bound of $f_i$ using Eq. (1);
12:     Invoking Algorithm 1 to obtain the reliability lower
        bound of $f_i$;
13:     if Requirements of latency and reliability, i.e, con-
        straints (9)-(10), are satisfied then
14:         Output the optimal strategies $X_{i,k}^{*}$ and $Y_{i,k}^{*}$, and
            maximum network utility $Utility^{*}$;
15:         Recompute the available resource matrix $Res_{band}$
            and $Res_{buff}$.
16:     else
17:         Delete the $X_{i,k}^{*}$ and $Y_{i,k}^{*}$, and recompute the optimal
            $Utility^{*}$.
18:     end if
19: end for
```

V. Evaluation

The performance of RFRSA is evaluated by exploiting MATLAB numerical simulation from multiple metrics, including the following:

- Access ratio: the fraction of authorized business flows that can access the network, compared to all flows.
- Total utility: the sum of latency utility and reliability utility of all business flows.
- Delay bound: the worst-case end-to-end delay of each flow, defined in III-B.
- Reliability: the reliability of each flow with K duplicates.

Since there is currently a shortage of IIoT datasets, we utilize three telecommunications network topology datasets in our simulation, including the Chinanet, Uunet, and Geant [27]. As shown in Fig.4, these networks serve different purposes and



(a) Random (b) Chinanet

(c) Uunet (d) Geant

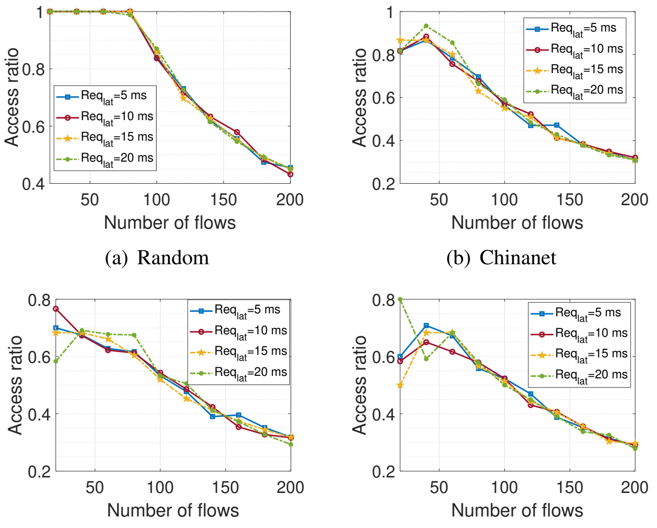
Fig. 4: Topologies from different datasets (Image credits: The Internet Topology Zoo).

TABLE II: Parameter Setting

Notion	Value range
Number of nodes, $\|\mathcal{V}\|$	20 ∼ 200
Number of flows, N	0 ∼ 500
Capacity of the link(u,v) , $R_{u,v}$	1000 Mbps
Buffer capacity of node u , B_u	1000 Mb
Reliability of node u,	99.55% ∼ 99.90%
Delay bound requirement of f_i , Req_i^{lat}	5 ∼ 20 ms
Reliability requirement of f_i , Req_i^{rel}	99% ∼ 99.999%
Redundancy of f_i , K	{1, 2, 3}
Weight factors, w_i,δ,ϕ and φ	0 ∼ 1
Arrival rate of f_i , r_i	5 ∼ 10 Mbps
Burst size of f_i , b_i	3 ∼ 5 Mb
Maximum packet size of flows , L^{max}	1500 Byte
Process delay , τ_{pro}	1 ∼ 2 ms
Constant delay , τ_{con}	1 ∼ 2 ms

are located in different regions. Chinanet and Uunet primarily serve as backbone networks and provide services to customers in China and the USA, respectively, while Geant serves as a backbone network, testbed, and transit network in Europe. The data ranges from 2010 to 2012, and the information is available in GraphML formats. We convert the GraphML format into CSV format using NetworkX [28] and pandas [29] tools for use in MATLAB experiments. Given the relatively sparse nature of the networks in these publicly datasets, we have additionally created a locally generated random network topology. The detailed information is as follows:

- Chinanet:
  - Nodes & Edges: 42 & 66
  - Connectivity: 7.67%
  - Network Type: Communication
  - Network Layer: Internet Protocol
- Uunet:
  - Nodes & Edges: 49 & 84
  - Connectivity: 7.14%
  - Network Type: Communication
  - Network Layer: Internet Protocol
- Geant:



(a) Random (b) Chinanet

(c) Uunet (d) Geant

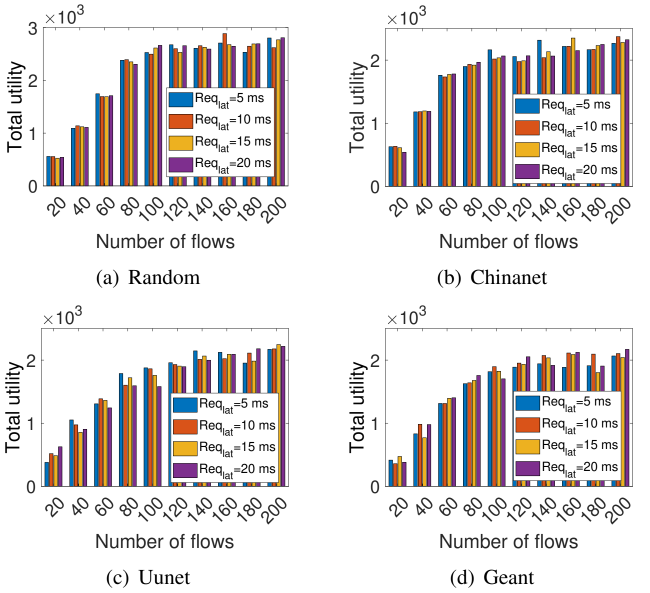
Fig. 5: Access ratio versus the number of flows, when $K = 3$.

  - Nodes & Edges: 40 & 61
  - Connectivity: 7.82%
  - Network Type: Research and Education Network
  - Network Layer: Internet Protocol
- Locally Generated Dataset:
  - Nodes & Edges: 50 & 602
  - Connectivity: 50.69%

Additionally, a random network topology is generated with 50 nodes to analyze the performance of RFRSA in different network topology scenarios. A laptop with a 2.4 GHz CPU and 16 GB RAM is utilized to conduct the simulation. Unless stated otherwise, the parameters are listed in Table II, as referred to in [8], [30]. Furthermore, three types of utility functions are adopted, i.e., affine function, exponential function, and staircase function. Three general formulas are defined, i.e., $U_{aff}(x) = \alpha_1 + \alpha_2 x$, $U_{\exp}(x) = \beta_1 \exp(\beta_2 x)$, and $U_{sta}(x) = \gamma_1 + \gamma_2 \lfloor \gamma_3 x \rfloor$. To eliminate the impact on algorithm performance, we adjust these parameters to limit the utility of a single business flow within the same range, thus reducing the significant differences between two utility values. Specific parameter configurations are established for the utility functions as follows: $\alpha_1^t = 52.2$, $\alpha_2^t = -432.02$, $\beta_1^t = 52.4$, $\beta_2^t = -16.25$, $\gamma_1^t = 49.3$, $\gamma_2^t = -3.9$, $\gamma_3^t = 100.3$, and $\alpha_1^r = -2891.8$, $\alpha_2^r = 2951$, $\beta_1^r = 1.4 \times 10^{-87}$, $\beta_2^r = 204.06$, $\gamma_1^r = -2940$, $\gamma_2^r = 3$, $\gamma_3^r = 1000.47$, respectively.

A. Access ratio and total utility

As shown in Fig. 5, as the number of flows increases under different latency requirements, the access ratio gradually declines, owing to the limited carrying capacity of the network. In this scenario, the redundancy degree is set to 3, causing the rate of growth for duplicate flows to be three times faster than that of practical business flows. Fig. 6 depicts that the total utility increases as the number of flows increases, but eventually levels off. Similarly, it is because the number of



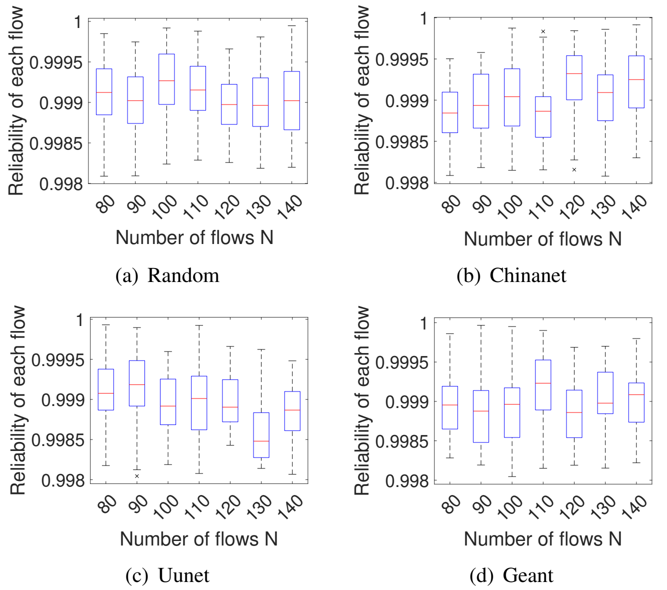
(a) Random

(b) Chinanet

(c) Uunet

(d) Geant

Fig. 6: Total utility versus the number of flows, when $K = 3$.



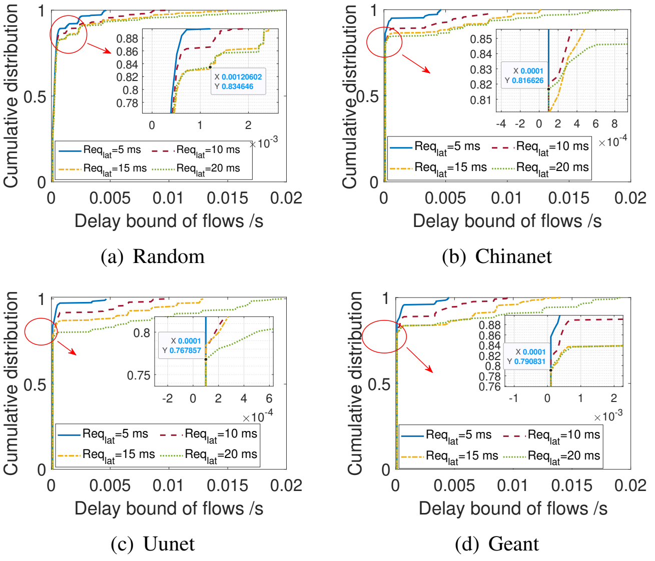
(a) Random

(b) Chinanet

(c) Uunet

(d) Geant

Fig. 8: Reliability of each flow, when $K = 3$, $Req_{lat} = 5$ms.



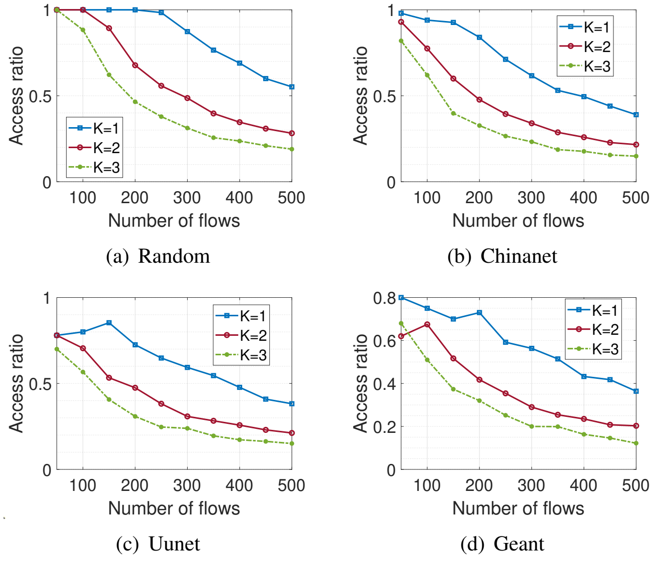
(a) Random

(b) Chinanet

(c) Uunet

(d) Geant

Fig. 7: Cumulative distribution function, when $K = 3$.



(a) Random

(b) Chinanet

(c) Uunet

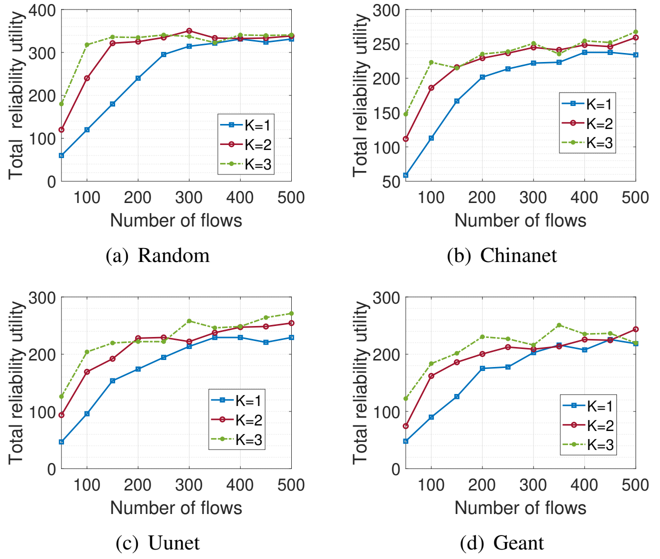
(d) Geant

Fig. 9: Access ratio with different redundancy when the reliability of an individual node is high.

flows is progressively nearing the maximum capacity of the network. Additionally, it shows that the access ratio and total utility using a random topology are better than Chinanet, Uunet, and Geant. This is because the random topology consists of 50 nodes and 602 edges, with better connectivity. We define the connectivity rate as $p = \frac{2N_{edge}}{N_{node}(N_{node}-1)}$, where N_{edge} and N_{node} represent the number of edges and nodes in the topology, respectively. By calculation, the connectivity rate of the random topology is 50.69%, while the other three topologies only have 7.67%, 7.14%, and 7.82%, respectively. Higher connectivity provides more redundant paths, thereby improving the system's reliability and total utility.

B. Latency and reliability

Fig. 7 illustrates the cumulative distribution of delay bounds for each business flow under different latency requirements when the number of flows is 100. When the latency requirement is $20ms$, more than 81.66% of traffic flows in both the random topology and Chinanet topology have delay bounds smaller than $1ms$. Moreover, Uunet and Geant topology have over 76.79% of business flows with delay bounds less than $1ms$. In addition, Fig. 8 shows that with a redundancy of 3, under different network topologies, as the number of traffic flows increases, the vast majority of flows can maintain a reliability of over 99.9%. Similarly, due to its higher connectivity, the random topology has relatively better low-latency and reliable



(a) Random

(b) Chinanet

(c) Uunet

(d) Geant

Fig. 10: Reliability utility with different redundancy.

performance compared to the other three network topologies.

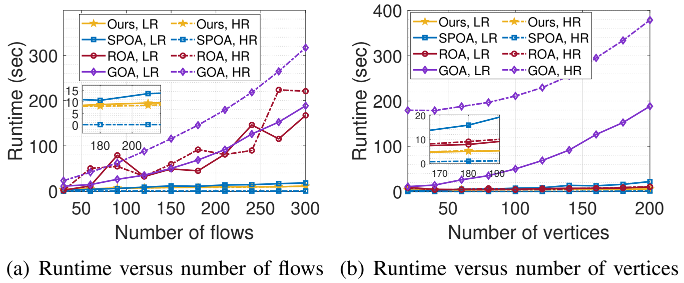
C. Redundancy degree selection

In practical IIoT systems, excessive redundancy leads to resource wasting or even performance degradation. As shown in Fig. 9, when the individual node reliability is over 99.9%, increasing the redundancy level leads to a decrease in the access ratio, especially when network connectivity is low. While a high redundancy level can enhance the reliability of business flows, it also consumes a large amount of bandwidth and buffer resources, resulting in lower access ratio.

Fig. 10 indicates that when the number of business flows is less than 200, increasing the redundancy level enhances the total reliability utility. After the number of business flows exceeds 200, the improvement in reliability gain slows down significantly even with increasing redundancy levels. This is because redundant paths of high-priority business flows consume network resources, leading to low-priority business flows being starved when the number of business flows approaches the capacity of the network. From a vertical point of view, increasing redundancy can improve the reliability of flows. However, as redundancy increases, the rate of improvement diminishes due to the marginal benefits gained when path reliability is already high. To determine a suitable redundancy degree, it is important to consider the reliability of the individual node. For stable nodes, a smaller degree of redundancy may be sufficient, but for nodes prone to failure, increasing redundancy may be necessary. As a general guideline, we recommend that redundancy levels not exceed 3.

D. Performance comparison

RFRSA is compared with four algorithms, including two heuristic algorithms, a solver-based algorithm, and a learning-based algorithm.

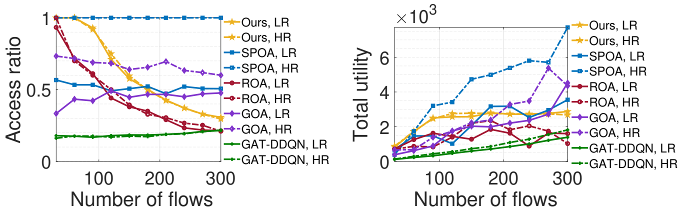


(a) Runtime versus number of flows (b) Runtime versus number of vertices

Fig. 11: Runtime comparison in random network topologies.

- Shortest Path-based Optimization Algorithm (SPOA): SPOA is built upon the methodology presented in [31] and focuses on finding the shortest path of flow while taking into account resource constraints. If the shortest path satisfies the reliability requirement, the flow can access the network; otherwise, it is rejected.
- Redundancy-based Optimization Algorithm (ROA): ROA is derived from [19] improving reliability by finding several paths for the business flow according based on the maximum-disjoint selection strategy.
- Gurobi-based Optimization Algorithm (GOA): Gurobi Optimizer [32] is a solver to find the best solution to mathematical problems. We design a Gurobi-based optimization algorithm to solve our MTSU problem due to that Gurobi cannot support signomial constraints (Eq. (9)) and polynomial constraints (Eq. (10)). Specifically, we initially employ Gurobi within MATLAB to calculate routing and scheduling solutions while considering resource constraints. Subsequently, we evaluate whether the present computation results meet latency and reliability requirements.
- Graph Attention-based Double Deep Q-Network (GAT-DDQN): Inspired by a GCN-powered MTDRL for joint network slicing and routing [33], we employ a graph attention-based double deep Q-network [34] to solve our routing and scheduling problem as a Markov decision process. The state of each node is represented by the occupied buffer, the number of occupied queues, the topological distance of the current flow from the source node to the destination node, the node degree, and an indicator symbolizing the current or historical location of the flow, while the state of each edge is represented by the occupied bandwidth. The reward is formulated as the weighted sum of the incremental system utility and the penalty terms related to the constraints. Following the proposed method, GAT-DDQN also prioritizes the routing and scheduling of each flow. It sequentially determines the next node with the maximum predicted Q-value for a flow in terms of the current node states and edge states.

The Mean Time to Failure (MTTF) is a more intuitive measure of system reliability, as it represents the expected lifetime of the system. Mathematically, the MTTF is given [24] by $MTTF = \frac{1}{\lambda}$, where λ denotes the failure rate of IIoT device and the system reliability is modeled by $R(t) = e^{-\lambda t}$.



(a) Access ratio versus number of flows (b) Total utility versus number of flows

Fig. 12: Performance comparison in a random network topology with 50 nodes.

Assuming that the MTTF of switch nodes representing relatively low and high reliability are $2,222.2$ and $10,000$ hours, corresponding to $\lambda_1 = 0.00045$ and $\lambda_2 = 0.0001$, respectively, then the switch reliability after ten hours of operation would be 99.55% or 99.90%, respectively. We refer to these two settings as "**LR**" and "**HR**" for brevity. Additionally, the connectivity of Chinanet, Uunet, and Geant network topologies is relatively low, resulting in a limited number of available redundant paths. Moreover, randomly generated network topologies possess a certain degree of generalizability compared to specific network topologies. Therefore, we conduct performance comparison experiments exclusively on randomly generated network topologies.

Fig. 11 presents the runtime comparison of four algorithms with different network and business scales. We exclude GAT-DDQN in the runtime comparison because it includes both training and running time. Fig. 11(a) and Fig. 11(b) demonstrate that our proposed algorithm has the shortest runtime when the switch reliability is relatively low. This is because our algorithm reduces the search space by reducing the dimensionality of variables, allowing it to find convergence points more quickly. Whereas, SPOA is more efficient when the switch reliability is relatively high. This is due to the fact that assigning a single path is enough to fulfill the reliability requirements of the business. Allocating multiple redundant paths would result in unnecessary utilization of network and time resources.

Fig. 12 depicts the performance comparison of algorithms in a random network topology with 50 vertices. Fig. 12(a) demonstrates that as the number of business flows increases, the access ratio decreases, but the decrease rate of SPOA and GOA is relatively slow. It is because of that SPOA and GOA does not utilize redundant paths, which enables a network of the same size to support more business flows. In contrast, ROA and our RFRSA utilize redundancy mechanisms, leading to a multiplication of network flows by a factor of K. Consequently, the number of business flows quickly reaches the network's maximum capacity, resulting in a rapid decline in the access ratio. Furthermore, when the number of business flows is small, ROA and our RFRSA in **LR** have higher access ratio. The reason is that a single low-reliable node is likely to cause failures in the transmission of business flows. However, at this point, the network resources are abundant, allowing for the construction of multiple redundant paths to enhance the reliability of business flow transmission.

It can also be observed that the access ratio of GOA are very low compared to SPOA. This is because the MTSU problem has many optimization variables, up to $N * V * V * Q$, which makes it tough for Gurobi to quickly output the best routing and scheduling solution. To address this, we reduce the frequency of invoking Gurobi in the design of GOA, which lowers the time complexity of GOA but also decreases the access ratio of business flows. Additionally, we have observed that the access ratio of GAT-DDQN remains consistently below 22%, significantly lower than that of other algorithms. This could be attributed to three potential reasons. Firstly, although our graph neural network is based on inductive learning rather than transductive learning, it is still challenging to generalize to arbitrary network topologies, especially when there are significant differences in graph sizes (ranging from 20 to 200 nodes in our experiments). Secondly, it is possible that the state representation and reward function in the reinforcement learning model may be inadequately designed. Some crucial information related to system utility may not be encoded in the state. Thirdly, some strongly nonlinear intermediate variables in the utility model may not be fitted well, which are essential for Q-value prediction.

Fig. 12(b) indicates that the total utility increases with the number of business flows and eventually saturates. In **LR**, ROA and our RFRSA have higher total utility when the number of flows is small. This is because the redundancy mechanism improves the reliability of business flow transmission, resulting in higher access ratio and total system utility. In **HR**, SPOA always maintains a high access ratio when the number of business flows is less than 300, resulting in a higher total system utility. This is because the reliability of a single path is already sufficiently high at this point, and non-redundant mechanisms further reduce resource overhead. Therefore, networks of the same scale can accommodate more business flows and achieve higher total system utility.

In conclusion, when the reliability of network nodes is low due to factors such as prolonged usage or poor industrial environment, our RFRSA would be the ideal choice to improve the reliability of business flows. Conversely, when the reliability of network nodes is relatively high, SPOA should be chosen to achieve higher resource utilization efficiency. From the perspective of application requirements, network administrators are advised to employ our RFRSA when IIoT applications demand high reliability or exhibit significant dynamics. On the other hand, the adoption of the SPOA is recommended for IIoT applications with stringent real-time requirements. From a deployment perspective, when the IIoT network has a larger scale and sufficient bandwidth and storage resources, adopting our RFRSA can significantly improve transmission reliability. Conversely, SPOA is recommended with smaller network scales and constrained bandwidth and storage resources. Ultimately, it is crucial to select the appropriate algorithm based on different scenarios, application requirements, and deployment environments to optimize network performance.

VI. Discussion

In this study, we investigate the routing and scheduling to satisfy the effective transmission of data flows in

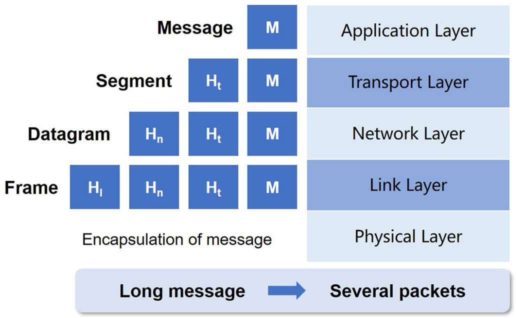


Fig. 13: Packet encapsulation structure based on the IP stack.

TSSDN-enabled IIoT systems. While our proposed solution can enhance network transmission performance and balance the network resources consumed by real-time and reliability requirements, there are still some challenges and complexities in its implementation. This research combines PQ and FIFO scheduling mechanisms to prioritize the forwarding of high-priority flows. Then, based on FRER, it enhances the reliability of flows through multi-path transmission. However, flow scheduling typically occurs at the network layer (Layer 3, L3), while FRER operates at the link layer (Layer 2, L2). TSSDN switch nodes distinguish whether data packets originate from the same information source by parsing the link-layer frame header information. Due to the hierarchical structure of packets [35], depicted in Fig. 13, the scheduling module of the switch cannot identify duplicates of the same packet solely from network-layer datagrams. Therefore, to ensure the successful operation of the proposed mechanism, it is necessary to add information similar to frame sequence numbers in the datagrams of the packets. Another issue involves hardware limitations since there are currently no operational TSSDN switches available. To tackle this, we will focus on researching and implementing TSSDN switches in the future.

VII. Conclusion

We propose to use the FRER mechanism to enhance the data transmission reliability in TSSDN-based IIoT systems. To gain analytical insights into the additional cost implications introduced by the FRER mechanism, we model the end-to-end delay bound and the reliability of flows based on NC and FRER. Using these models, we further formulate a total utility optimization problem to ensure the low-latency and reliable transmission of critical business flows, subject to resource and transmission constraints. Then, we proposed RFRSA to dynamically and rapidly compute the flow routing and scheduling strategy. Finally, numerical results showed the superior performance of RFRSA compared to existing algorithms, and the criteria for choosing the suitable algorithm for specific IIoT systems has been discussed.

References

[1] R. Huo, S. Zeng, Z. Wang, J. Shang, W. Chen, T. Huang, S. Wang, F. R. Yu, and Y. Liu, “A comprehensive survey on blockchain in industrial internet of things: Motivations, research progresses, and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 88–122, 2022.

[2] M. Boehm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser, “Time-sensitive software-defined networking: A unified control- plane for tsn and sdn,” in *Mobile Communication - Technologies and Applications; 24. ITG-Symposium*, 2019, pp. 1–6.

[3] J. Xue, G. Shou, Y. Liu, and Y. Hu, “Scheduling time-critical traffic with virtual queues in software-defined time-sensitive networking,” *IEEE Transactions on Network and Service Management*, 2023, early access.

[4] S.-H. Chang, H. Chen, and B.-C. Cheng, “Time-predictable routing algorithm for time-sensitive networking: Schedulable guarantee of time-triggered streams,” *Computer Communications*, vol. 172, pp. 183–195, 2021.

[5] D. N. M. Hoang and J. M. Rhee, “Comparative analysis of iec 62439–3 (hsr) and ieee 802.1cb (frer) standards,” in *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2021, pp. 231–235.

[6] Q. Wang, Z. Mo, B. Yin, L. Zhang, and P. Dong, “Bounding the upper delays of the tactile internet using deterministic network calculus,” *Electronics*, vol. 12, no. 1, p. 21, 2023.

[7] L. Thomas, A. Mifdaoui, and J.-Y. Le Boudec, “Worst-case delay bounds in time-sensitive networks with packet replication and elimination,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2701–2715, 2022.

[8] L. Ji, W. Wu, C. Gu, J. Bi, S. He, and Z. Shi, “Network calculus-based routing and scheduling in software-defined industrial internet of things,” in *2022 IEEE 20th International Conference on Industrial Informatics (INDIN)*. IEEE, 2022, pp. 463–468.

[9] T. Gerhard, T. Kobzan, I. Blöcher, and M. Hendel, “Software-defined flow reservation: Configuring ieee 802.1q time-sensitive networks by the use of software-defined networking,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 216–223.

[10] A. Montazerolghaem and M. H. Yaghmaee, “Load-balanced and qos-aware software-defined internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3323–3337, 2020.

[11] A. Montazerolghaem, “Software-defined internet of multimedia things: Energy-efficient and load-balanced resource management,” *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2432–2442, 2022.

[12] Y. Lu, L. Yang, S. X. Yang, Q. Hua, A. K. Sangaiah, T. Guo, and K. Yu, “An intelligent deterministic scheduling method for ultralow latency communication in edge enabled industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1756–1767, 2023.

[13] M. Böhm, J. Ohms, M. Kumar, O. Gebauer, and D. Wermser, “Time-sensitive software-defined networking: a unified control-plane for tsn and sdn,” in *Mobile Communication-Technologies and Applications; 24. ITG-Symposium*. VDE, 2019, pp. 1–6.

[14] F. Tang, B. Mao, Y. Kawamoto, and N. Kato, “Survey on machine learning for intelligent end-to-end communication toward 6g: From network access, routing to traffic control and streaming adaption,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1578–1598, 2021.

[15] L. Xu, Q. Xu, J. Tu, J. Zhang, Y. Zhang, C. Chen, and X. Guan, “Learning-based scalable scheduling and routing co-design with stream similarity partitioning for time-sensitive networking,” *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 353–13 363, 2022.

[16] F. Naeem, G. Srivastava, and M. Tariq, “A software defined network based fuzzy normalized neural adaptive multipath congestion control for the internet of things,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2155–2164, 2020.

[17] H. Gao, C. Liu, Y. Li, and X. Yang, “V2vr: Reliable hybrid-network-oriented v2v data transmission and routing considering rsus and connectivity probability,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3533–3546, 2021.

[18] R. K. Lenka, A. K. Rath, and S. Sharma, “Building reliable routing infrastructure for green iot network,” *IEEE Access*, vol. 7, pp. 129 892–129 909, 2019.

[19] D. Ergenç and M. Fischer, “On the reliability of ieee 802.1cb frer,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[20] A. Naushad, G. Abbas, S. A. Shah, and Z. H. Abbas, “Energy efficient clustering with reliable and load-balanced multipath routing for wsns,” in *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, 2020, pp. 1–9.

[21] L. Maile, D. Voitlein, K.-S. Hielscher, and R. German, “Ensuring reliable and predictable behavior of ieee 802.1cb frame replication and elimination,” in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 2706–2712.

[22] H. Shi, M. Zheng, W. Liang, and J. Zhang, “Aodr: an automatic on-demand retransmission scheme for wia-fa networks,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6094–6107, 2021.

[23] K. Polachan, B. Turkovic, T. Prabhakar, C. Singh, and F. A. Kuipers, “Dynamic network slicing for the tactile internet,” in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, 2020, pp. 129–140.

[24] I. Koren and C. M. Krishna, *Fault-tolerant systems*. Morgan Kaufmann, 2020.

[25] M. Jia, W. Liang, M. Huang, Z. Xu, and Y. Ma, “Routing cost minimization and throughput maximization of nfv-enabled unicasting in software-defined networks,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 732–745, 2018.

[26] F. Luo, Z. Y. Dong, Y. Chen, and J. Zhao, “Natural aggregation algorithm: A new efficient metaheuristic tool for power system optimizations,” in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2016, pp. 186–192.

[27] U. of Adelaide. The internet topology zoo. Accessed: Apr. 2023. [Online]. Available: http://www.topology-zoo.org/dataset.html

[28] N. developers. Networkx. Accessed: Apr. 2023. [Online]. Available: https://networkx.org/

[29] NumFOCUS. pandas. Accessed: Apr. 2023. [Online]. Available: https://pandas.pydata.org/

[30] Q. Zhang, F. Liu, and C. Zeng, “Online adaptive interference-aware vnf deployment and migration for 5g network slice,” *IEEE/ACM Transactions on Networking*, 2021.

[31] J. Augustine, K. Hinnenthal, F. Kuhn, C. Scheideler, and P. Schneider, “Shortest paths in a hybrid network model,” in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 1280–1299.

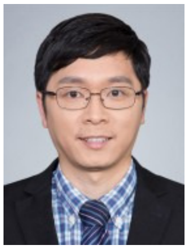
[32] G. OPTIMIZATION. Gurobi optimizer. Accessed: Oct. 2023. [Online]. Available: https://www.gurobi.com/solutions/gurobi-optimizer/

[33] T. Dong, Z. Zhuang, Q. Qi, J. Wang, H. Sun, F. R. Yu, T. Sun, C. Zhou, and J. Liao, “Intelligent joint network slicing and routing via gcn-powered multi-task deep reinforcement learning,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1269–1286, 2022.

[34] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

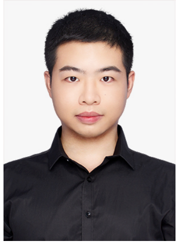
[35] J. F. Kurose, *Computer networking: A top-down approach featuring the internet, 3/E*. Pearson Education India, 2005.

Luyue Ji (Student Member, IEEE) received the B.Eng. degree in electronic information engineering and the M.Eng. degree in signal and information processing from Southwest University, Chongqing, China, in 2012 and 2016, respectively. She is currently working toward the Ph.D. degree in control science and engineering with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. Her research interests include industrial IoT, software-defined networking, network slicing, resource allocation.



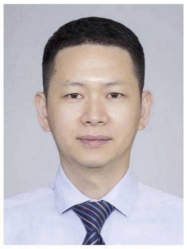
Shibo He (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2012. From November 2010 to November 2011, he was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. From March 2014 to May 2014, he was an Associate Research Scientist and from May 2012 to February 2014, a Postdoctoral Scholar with Arizona State University, Tempe, AZ, USA. He is currently a Professor with Zhejiang University. His research interests include Internet of Things, crowdsensing, and Big Data analysis.

Prof. He is on the editorial board of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Peer-to-Peer Networking and Applications, and KSII Transactions on Internet and Information Systems. He is also the Guest Editor of Computer Communications and International Journal of Distributed Sensor Networks. He was a Symposium Co-Chair of the IEEE GlobeCom 2020 and IEEE ICC 2017, a TPC Co-Chair of the i-Span 2018, a Finance and Registration Chair of the ACM MobiHoc 2015, a TPC Co-Chair of the IEEE ScalCom 2014, a TPC Vice Co-Chair of the ANT 2013’lC2014, a Track Co-Chair of the Pervasive Algorithms, Protocols, and Networks of EUSPN 2013, a Web Co-Chair of the IEEE MASS 2013, and a Publicity Co-Chair of the IEEE WiSARN 2010 and FCN 2014.



Chaojie Gu (Member, IEEE) received the B.Eng. degree in information security from the Harbin Institute of Technology, Weihai, China, in 2016, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2020. He was a Research Fellow with Singtel Cognitive and Artificial Intelligence Lab for Enterprise, in 2021. He is currently an Assistant Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include IoT, industrial IoT, edge computing, and low-power wide area network.

Zhiguo Shi (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electronic engineering from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. Since 2006, he has been a Faculty Member with the Department of Information and Electronic Engineering, Zhejiang University, where he is currently a Full Professor. From 2011 to 2013, he visited the Broadband Communications Research Group, University of Waterloo, Waterloo, ON, Canada. His current research interests include signal and data processing. Prof. Shi serves as an Editor of the IEEE Network, IEEE Transactions on Vehicular Technology, IET Communications, Journal of The Franklin Institute. He was a recipient of the Best Paper Award from the ISAP 2020, IEEE GLOBECOM 2019, IEEE iThings 2019, EIA MILCOM 2018, IEEE WCNC 2017, CSPS 2017, IEEE/CIC ICCC 2013, IEEE WCNC 2013, IEEE WCSP 2012. He was also a recipient of first prize for Wu Wenjun AI Science and Technology Award in 2020, first prize for Science and Technology Development of the Ministry of Education in 2015 and second prize for the Scientific and Technological Award of Zhejiang Province in 2012.



Jiming Chen (Fellow, IEEE) received the PhD degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2005. He is currently a professor with the Department of Control Science and Engineering, the vice dean of the Faculty of Information Technology, Zhejiang University. His research interests include IoT, networked control, wireless networks. He serves on the editorial boards of multiple IEEE Transactions, and the general co-chairs for IEEE RTCSA’19, IEEE Datacom’19 and IEEE PST’20. He was a recipient of the 7th IEEE ComSoc Asia/Pacific Outstanding Paper Award, the JSPS Invitation Fellowship, and the IEEE ComSoc AP Outstanding Young Researcher Award. He is an IEEE VTS distinguished lecturer. He is a fellow of the CAA.