DigiNet: Scaling up Provisioning of Network Digital Twin

Marcelo C. Luizelli[1], Francisco G. Vogt[5], Paulo Silas Severo de Souza[1], Arthur F. Lorenzon[2]
Roberto I. T. da Costa Filho[3], Fabio D. Rossi[4], Rodrigo Calheiros[6], Christian Esteve Rothenberg[5]

[1]Federal University of Pampa (UNIPAMPA), [2]Federal University of Rio Grande do Sul (UFRGS)
[3]Instituto Federal Sul-Rio-Grandense (IFSul), [4]Instituto Federal Farroupilha (IFFar)
[5]Universidade Estadual de Campinas (UNICAMP), Brazil, [6]Western Sydney University (WSU), Australia

***Abstract*—The pursuit of self-driving networks is increasing pressure on adopting intelligent, edge-based networking services. However, deploying autonomous network models within operational and large-scale infrastructures entails substantial risks that require rigorous verification and validation procedures. In this context, the application of a Network Digital Twin (NDT) is emerging as a viable approach towards intelligent network decision-making based on high-fidelity models built upon digital representations of physical network devices (i.e., Digital Twins). In this paper, we take the first steps towards efficiently provisioning NDT models. To that end, we introduce the Digital Twin Network Provisioning Problem (DigiNet), which encompasses the optimal placement of NDT models and the efficient collection of telemetry data for synchronizing NDT models with their physical counterparts. We theoretically formalize DigiNet as a Mixed-Integer Linear Programming (MILP) model and present a polynomial-time heuristic. Our results show that DigiNet outperforms baseline approaches by up to 10x regarding the number of NDT models provisioned.**

***Index Terms*—Network Digital Twin, Optimization, Artificial Intelligence, Software-defined Networks.**

I. Introduction

The sixth-generation (6G) of mobile networks is already on the way [1]. 6G networks are envisaged to provide native intelligent-, edge-based networking services to support future and emerging applications with stringent service-level objectives. The envisioned applications will require seamless and autonomous adaptation of the network infrastructure to timely optimize high-level service objectives in a context-aware and data-driven manner [2]. Despite a few research initiatives [3], there is a long way to go before intelligent and autonomous networks are commercially viable. The level of in-network intelligence envisaged in the 6G context will require extensive usage of *network programmability* based on a multitude of *disjoint computing models* (e.g., cloud, edge, and in-network computing), delivering in-network Artificial Intelligence (AI)-based control over packet processing [4].

Empowering the network with AI, however, is not a trivial task. One of the fundamental questions behind the usage and adoption of AI in critical systems is whether the devised and trained models are correct and faithful to reality. Dealing autonomous AI models into the 6G infrastructure to manage and control the network and its running services is highly risky without proper verification and validation. In this context, applying the concept of Digital Twin (DT) to networking [5], [6] is increasingly emerging as a promising approach to mitigate operational hazards of AI-based network control, among other lifecycle concerns such as design, deployment, operation, and expansion phases of a network [7].

DTs allow the trustworthy creation of digital representations of physical network entities (e.g., routers, servers, and services) to support management decisions across the whole network lifecycle. For instance, network operators can safely test experimental network policies within the DTs without jeopardizing the daily operation of the real network. An essential requirement for DT realization is a closed-loop feedback link established between digital and physical entities, which enables DTs to accurately replicate the behavior of the physical network. More specifically, when measurements are made in the physical entity, they are used as input into the digital counterpart, which allows DT models to be retrained and adapted to possibly unforeseen behaviors. As such, DT-based simulations/emulations are expected to revolutionize the network operation landscape, enabling lower maintenance costs and efficient support for proactive decisions against attacks and performance issues.

Employing DT in the next-generation networks will require substantial efforts in resource provisioning and orchestration to keep up with digital and physical entity requirements. Despite existing efforts towards the full realization of NDN [8]–[13], little has been done to orchestrate the resource allocation in this new paradigm. Previous research has shed some light on this direction [12], [13]. For instance, Wu et al. [13] have delineated the potential advantages of leveraging cloud and edge resources to host NDT models. Cloud platforms with virtually unlimited computational capabilities could support large-scale data analyses and compute-intensive tasks. On the other hand, edge servers strategically positioned near the access network infrastructure could deliver timely feedback for NDT tasks that require low latency and high bandwidth.

In this paper, we tackle the resource provisioning and synchronization of NDT models. On deploying NDTs, we face two major challenges: (*i*) where to deploy NDT models, and (*ii*) how to efficiently collect telemetry data to keep physical and digital counterparts synchronized. Although these two optimization problems have been widely stud-

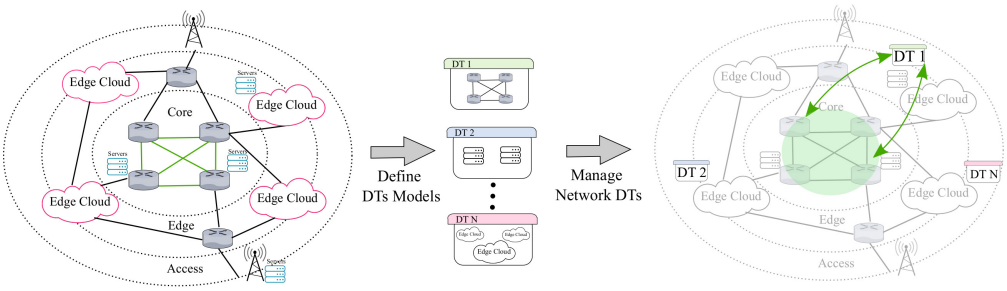


Figure 1. Overview of the Network Digital Twin Provisioning Process.

ied in different contexts (e.g., placement [14] and monitoring/synchronization [15]), the coordination between them to provide feasible NDT deployment is still an open research problem, including from a pure networking perspective [16].

Towards addressing the open challenges, we introduce the `DigiNet` problem, which leverages In-Band Network Telemetry (INT) [17] to opportunistically collect telemetry data to NDT models within existing network flow packets. We formalize DigiNet as a MILP (Mixed-Integer Linear Programming) model, which can be seen as a generalization of the INT-based problem [18] and, therefore, is NP-hard. Despite the scalability limitations of solving NP-hard problems, our exact formulation represents an optimal bound for future algorithmic approximations. We also introduce a polynomial-time heuristic that can find high-quality solutions on time. Our proposal wisely orchestrates the placement of NDT models and the collection of the required telemetry data. Results show that DigiNet outperforms the state-of-the-art INT solution [18] by a factor of 10x related to the number of NDT models satisfied, while making better use of available resources.

Our main contributions can be summarized as follows:

- We discuss NDT requirements and performance implications of NDT provisioning and communication.
- We theoretically formalize the DigiNet problem to optimally NDT provisioning.
- We introduce a polynomial-time heuristic that provides near-optimal solutions.
- We shed light on several research challenges that represent promising directions for future investigation.
- We provide open-source software artifacts for experimental reproducibility.

The remainder of this paper is organized as follows. In Section II, we provide a background on NDT architectural reference and the networking requirements for making resource provisioning. In Section III, we introduce the formalization of DigiNet model, while in Section IV we introduce the design of the proposed heuristic. In Section V, we present and discuss the evaluation results of our proposed approach. In Section VI, we review related work and in Section VII we discuss existing research challenges and future directions. Last, in Section VIII, we conclude the paper with final remarks.

II. Digital Twin-based Programmable Networks

NDT is the virtual representation of a physical network used to analyze, diagnose, emulate, and control the physical network based on data, models, and interfaces [19]. Different from a digital model, an NDT can be fully synchronized in both directions (i.e., to and from the DT) by using a data flow. Figure 1 presents an overview of the general concept behind an NDT. On the left side, the figure presents a full-fledged network infrastructure composed of typical network elements (e.g., routers, and links). These elements, or a subset of them, are used to create high-fidelity digital representations, which are used across the entire life-cycle of network management.

An NDT model is expected to implement the basic behavior of a network element and the dynamics among other elements it interacts with. Besides, advanced functional features such as AI models are also expected to interact with the basic functionalities. For instance, one can consider an NDT element representing the data plane of a programmable router. In this case, the intrinsic data plane metadata could be used as input to the model to assess the behavior of the data plane. On top of this basic model, an advanced AI model could be plugged in to foresee future events such as network congestion, network faults, and heavy-hitter flows. Once these models are fully defined, they are expected to run on computing premises such as a Cloud/Edge node. In Figure 1 (right), DT 1 is deployed on an Edge Cloud node in the neighborhood of the physical network. The green arrows represent the full synchronization data flow between digital and physical counterparts.

There are many benefits of mirroring physical network devices to a DT. First, with real-time data, digital models are much more accurate and allow high-fidelity simulations. These high-fidelity simulations enable network operators to optimize the decision-making process involved in network management for instance, (*i*) to ensure network correctness by applying network verification, (*ii*) to perform network performance assurance of SLAs, (*iii*) to network traffic provisioning or maintenance, or (*iv*) to perform network prediction by using AI models for predicting network, data, services, or users behaviors. Despite the envisioned benefits, realizing an NDT is not such a simple task. Many challenges are involved: data acquisition, assembly of digital models, and data flow communication between digital-physical parts.

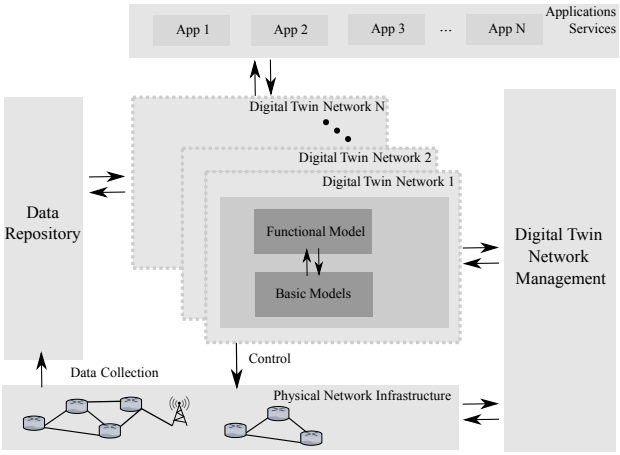


Figure 2. Overview of the Reference Architecture of a Network Digital Twin based on the IETF [19].

A. Digital Twin Network Reference Architecture

Next, we overview the NDT reference architecture under discussion at the IETF [19]. The architecture is a first effort toward realizing the NDT components and their relationships – shedding light on resource allocation requirements.

Figure 2 illustrates the basic NDT architectural reference. A NDT consists of five key elements: (*i*) data, (*ii*) physical and digital elements and their relationships (e.g., the mapping), (*iii*) NDT models, (*iv*) communication interfaces, and (*v*) NDT management. Data is the input information that NDT models continually receive from physical devices. The data can be collected from the network infrastructure using traditional network monitoring protocols (e.g., SNMP or NetFlow) or fine-grained approaches such as INT. This data is batched in a data repository to be shared with NDT models, which are expected to implement the basic functionalities of physical devices alongside functional models, which refer to various data models used for network analysis, emulation, diagnosis, prediction, and assurance. Functional models aim to realize the dynamic evolution of network performance evaluation and intelligent decision-making. Finally, NDT Management refers to the lifecycle management of individual or multiple NDTs, which encompasses provisioning DTs, monitoring their performance, orchestrating data collection, and analyzing network conditions.

On top of this reference architecture, multiple applications and services can benefit from NDT to implement conventional or innovative network operations. For instance, applications can make certain requests that need to be addressed by the NDT – e.g., a request can be a simulation or service emulation in a given operational context.

B. Digital Twin Network Requirements

An NDT can represent any physical device in a network infrastructure or a subset. Therefore, it is essential to discuss the data requirement of each digital representation considering: (*i*) data source (single or multiple), (*ii*) data collection method (e.g., INT, SNMP, etc), (*iii*) data volume injected in the physical network, (*iv*) and data collection frequency, among other networking-related issues inherent to specific NDT properties and NDT placement architectures [16].

Considering the higher level of disaggregation in software-based network infrastructure, network statistics data can come from single centralized source devices or multiple decentralized ones. For instance, one can have multiple instances of a network function deployed on top of a physical network. Therefore, the data come from multiple sources, although they might represent the same "physical" (or virtualized) object.

Different data collection methods can be applied based on the physical entity represented as an NDT. For instance, if a data plane program is being virtually represented, it may be preferable to collect its internal data using a very low-level network telemetry approach (e.g., using In-Band Network Telemetry – INT). In contrast, in a representation of a Control Plane, OpenFlow can be used. That would also impact the volume of network statistics transferred between physical and digital elements. When data plane programs are represented as a digital element, the data volume can easily surpass the GB order per second. The data can change internally in the data plane in the order of nanoseconds.

NDT requirements directly impact the DT simulation side. Depending on the physical DT, there are a few network simulators that can be adjusted for modeling network elements (e.g., ns-3[1], NS4[2], PFPSim[3], EdgeSimPy [20], CloudSim [21]). These simulators primarily use a discrete-event simulation model to simulate network scenarios. Discrete-event simulation is a widely used mathematical model used to describe the behavior of a system as a sequence of discrete events that occur at specific points in time. Each event represents a change in the system's state, which is processed in chronological order. While the primary mathematical model is a discrete-event simulation, these simulators also incorporate other mathematical and statistical models for modeling network protocols, traffic patterns, and network behaviors (e.g., queuing models, propagation models, error models). The computational resources such simulators require can vary widely depending on the specific simulation scenario, the network scale, the level of detail in the simulation, and the hardware and software configurations. Table 1 highlights key NDT requirements.

III. DiginET: Optimal Model for Network Digital Twin Resource Allocation

This section makes a case for leveraging INT capabilities for collecting monitoring data in programmable networks to serve as input to NDT models. First, we overview the problem. Then, we detail the proposed mathematical model.

A. Problem Overview

The DigiNet problem tackles the provisioning of NDT models by jointly optimizing the two main stages: (*i*) placement

[1]ns-3: https://www.nsnam.org/
[2]NS4: https://github.com/p4db/NS4-DEV
[3]PFPSim: https://pfpsim.github.io/

Table I
AN OVERVIEW OF NETWORK DIGITAL TWIN REQUIREMENTS.

Physical Digital Twin	Data Source	Data Collection	Data Volume	Data Frequency	Metric	DT Simulator
Data plane	Single/Multiple	INT	GB	nanoseconds	Ingress port	NS4/PFPSim
Control plane	Single	OpenFlow/P4 Runtime	MB	milliseconds	Latency	PFPSim
Network functions	Single/Multiple	OpenFlow/INT	GB	nanoseconds	Processing time	ns-3/NS4
Network slices	Single	OpenFlow/SNMP	MB	milliseconds	QoE	ns-3/CloudSim
Network services	Single	SNMP	MB	milliseconds	Availability	EdgeSimPy

and (*ii*) data telemetry collection. In (*i*), DigiNet optimally selects the best subset of available edge servers to host the NDT models being provisioned. Then, in (*ii*), DigiNet leverages INT to opportunistically collect and deliver the telemetry data required for the NDT models in place. The core idea behind INT (a.k.a. INT-MD (eMbed Data)[4]) is to embed near real-time telemetry data within existing flow packets in the network.

When dealing with NDTs within massive large-scale network infrastructures, INT is considered a prominent solution mainly because traditional monitoring techniques (e.g., SNMP polling, NetFlow, IPFix) prove inadequate to collect fine-grained, fresh, and accurate performance information [22]. That allows NDT models to be satisfied independently of the collection frequency required without competing for network resources. DigiNet applies INT within the context of NDTs, embedding telemetry data required for updating NDT models into existing network flows. As a result, it ensures near real-time synchronization between physical and digital entities.

Figure 3 illustrates an instance of the DigiNet problem. The network infrastructure comprises nine forwarding devices (nodes A to I) and two NDT models (NDT 1 and NDT 2) deployed on edge servers connected to nodes B and H. Each NDT model simulates a subset of components from the physical infrastructure, making inferences and reacting according to certain network behaviors. For example, NDT 1 simulates the behavior of physical nodes C, D, and E. We assume that NDT models implement distinct functionality, thereby requiring different telemetry data from the physical devices (NDT 1 requires ◯ and △, and NDT 2 requires ★ and □). The network infrastructure has three network flows (f_1, f_2, and f_3), whose packets are modified along their paths using INT to send telemetry data to the edge servers hosting the NDT models. For simplicity, we assume that a single packet can carry a maximum of two telemetry items simultaneously. Packets are space-bounded (i.e., w.r.t. bytes); so, it is infeasible (in most cases) to collect all network telemetry data with a single packet. In Figure 3, f_2 utilizes the first packet to collect data from physical device E, while the second one is from D.

An NDT model is considered satisfied *iff* all required telemetry data from its physical counterpart is collected and sent to the corresponding edge server where it is running. In the example, each edge server has a computation capacity,

[4]INT specification: https://github.com/p4lang/p4-applications/blob/master/docs/INT_v2_1.pdf

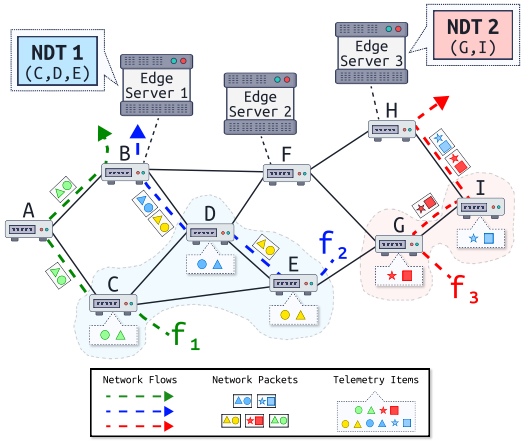


Figure 3. Overview of a DigiNet problem instance.

while NDT models demand fractions of that to perform their simulations. Despite using INT mechanisms to collect telemetry data, the telemetry data must be extracted from the packet and sent to an NDT model at some point in the routing path. To reduce the transmission overhead, we consider that only edge servers placed in the routing path of existing network flow are deemed suitable candidates for placing NDT models. In the example, network flows f_1 and f_2 can collect telemetry data to Edge Server 1, f_3 to Edge Server 3, and any network does not attend Edge Server 2 flows.

As one can observe, the DigiNet problem is not trivially solved. Both the (*i*) NDT model placement and (*ii*) data telemetry collection are intertwined and play a key role in determining the successful NDT deployment and operation. Next, we formally describe the DigiNet optimization problem.

B. Model Description and Notation

The optimization model we propose to solve the DigiNet problem defined above considers a physical programmable network infrastructure $G = (D, L)$, a set of active network flows F, a set of telemetry items V, a set of NDT models M, and a set of edge servers E. Set D in network G represents physical network devices (e.g., forwarding devices and servers) $D = \{d_1, ..., d_{|D|}\}$, while set L links interconnecting pair of devices $(d_i, d_j) \in (D \times D) : d_i \neq d_j$. We assume physical device $d_i \in D$ can embed a subset of items $V_d \subseteq V$ into packets of flow $f \in F$. Each telemetry item $v \in V$ has its size defined by the function $S : V \rightarrow \mathbb{N}^+$.

Network flows F are used to collect real-time telemetry data from forwarding devices D. A flow $f_k \in F$ has two endpoints (*i.e.*, ingress and egress forwarding devices) and is routed through the network infrastructure G using a simple path $\mathcal{P}t$. We denote the path taken by flow f_k as function $\mathcal{P}t : F \rightarrow \{D_1 \times ... \times D_{|D|}\}$. A path is valid *iff* it is simple and utilizes existing links in G, i.e., $(i,j) \in \mathcal{P}t(f) : \ (d_i, d_j) \in L$. Network flows F are encapsulated in a forwarding protocol (*e.g.*, IPv4). Therefore, the amount of available space to embed telemetry items in packets is bounded by a constant $K_f \in \mathbb{N}^+$.

NDT models are represented by set M. Each $m_l \in M$ is a digital representation of the physical counterpart and consists of a subset of physical devices such that $D^m \subseteq D$. Each digital representation $d \in D^m$ requires a subset of telemetry data from its physical counterpart to operate properly. We denote that subset for $R_d^m \subseteq V_d$. For instance, NDT-1 can be represented by $D^1 = \{0, 3\}$, with $R_0^1 = \{0, 1\}$ and $R_3^1 = \{1, 3, 4\}$. In other words, NDT 1 digitally represents devices 0 and 3. Digital representation of device 0 requires telemetry data 0 and 1. For simplicity, we assume that if network links are part of the digital representation, then telemetry data is associated with the corresponding link's endpoints. For example, suppose an NDT model is a digital representation of the network link usage. In that case, metrics such as latency and average bit rate transmission are computed and collected in the link's endpoints. Each NDT model $m \in M$ requires a set of computing resources to run properly. We model that as function $C^m : M \rightarrow \mathbb{N}^+$. Edge servers are represented by subset $E \subseteq D$. For generality, we assume any physical device in D can have an edge server associated. As illustrated in Figure 1, nodes B, F, and H have associated edge servers. Each edge server $e \in E$ has an amount of computing resources to run NDT models. We model that as function $C^e : E \rightarrow \mathbb{N}^+$.

An NDT model is satisfied *iff* all telemetry data required by NDT models are collected $R_d^m : \ (\forall d \in D^m, \forall m \in M)$ and sent to an edge server E. Given a network infrastructure G, a set of network flows F, a set of NDT models M, a set of telemetry items V, and a set of edge servers E, the optimization problem seeks a feasible solution that maximizes the number NDT model satisfied. The model output is denoted by a 5-tuple $\chi = \{Y, S, X, S^d, S^m\}$. Variables from $Y = \{\, y_{d,v,f}\,, \forall\, d \in D, v \in V, f \in F\}$ indicate that a physical device d embed telemetry item $v \in V_d$ into packets of network flow f. Variables from $S = \{\, s_{m,d,f,v}\,, \forall\, m \in M, d \in D^m, f \in F, v \in V\}$ indicate the amount of collected telemetry data by flow f for device d of NDT model m using the first k hops of the routing path. Variables from $X = \{\, x_{m,k}\,, \forall\, m \in M, k \in E\}$ indicates that NDT model m is deployed on edge server k. Variables from $S^d = \{\, s_{m,d}\,, \forall\, m \in M, d \in D^m\}$ indicate that digital representation d of NDT model m is satisfied. Last, Variables from $S^m = \{\, s_m\,, \forall\, m \in M\}$ indicate that NDT m is satisfied. Next, we describe the MILP formulation for the DigNet problem:

$$\textbf{Maximize} \quad \sum_{m \in M} s_m^d \tag{1}$$

Subject to:

$$\sum_{d \in \mathcal{P}t(f)} \sum_{v \in V_d} y_{d,v,f} \cdot S(v) \leq K_f \qquad \forall f \in F \tag{2}$$

$$s_{m,d,f,k} \leq \sum_{v \in V_d} \sum_{\substack{j \in \mathcal{P}(f): \\ j=i \\ \&j<k}} y_{d,v,f} \qquad \forall m \in M, d \in D^m, f \in F, k \in \mathcal{P}t(f) \tag{3}$$

$$\sum_{\substack{k \in \mathcal{P}(f): \\ d<k}} s_{m,d,f,k} \leq 1 \qquad \forall m \in M, f \in F, d \in D^m \tag{4}$$

$$s_{m,d,f,k} \leq x_{m,k} \qquad \forall m \in M, f \in F, d \in D^m, k \in \mathcal{P}(f) \tag{5}$$

$$\sum_{i \in D} x_{m,k} = 1 \qquad \forall m \in M \tag{6}$$

$$\sum_{m \in M} \sum_{k \in D} x_{m,k} \leq |E| \tag{7}$$

$$\sum_{m \in M} x_{m,k} \cdot C^m(m) \leq C^e(k) \qquad \forall k \in E \tag{8}$$

$$s_{m,d}^b \leq \sum_{f \in F} \sum_{\substack{k \in \mathcal{P}(f): \\ d<k}} s_{m,d,f,k} \cdot \frac{1}{|R_d^m|} \qquad \forall m \in M, d \in D^m \tag{9}$$

$$s_m^b \leq \sum_{d \in D^m} s_{m,d}^b \cdot \frac{1}{|D^m|} \qquad \forall m \in M \tag{10}$$

$$y_{d,v,f} \in \{0, 1\} \qquad \forall d \in D, vinV, f \in F \tag{12}$$

$$s_{m,d,v,f} \in \mathbb{N}^+ \qquad \forall m \in M, dinD^m, v \in V, f \in F \tag{13}$$

$$x_{m,k} \in \{0, 1\} \qquad \forall m \in M, k \in D \tag{14}$$

$$s_{m,d}^b \in \{0, 1\} \qquad \forall m \in M, dinD^m \tag{15}$$

$$s_m^b \in \{0, 1\} \qquad \forall m \in M \tag{16}$$

Constraint set (2) ensures that a network flow $f \in F$ does not exceed its capacity (*i.e.*, K_f). Constraint set (3) accounts for the amount of collected telemetry data by flow f in its routing path for a given NDT model m and the corresponding physical device d. The value of k indicates how many hops of the routing path are being used. For instance, if $k = 3$, then we count the ability of flow f to collect data only in the first initial three hops. Constraint sets (4) and (5) control where the edge server is placed. In other words, constraint set (4) ensures that only a single sinking device is selected, while constraint set (5) indicates which sinking device is selected for NDT model m. Then, constraint set (6) ensures that only a single edge server k is used by NDT model m. Note that in case we have distributed deployment of a single NDT model, this constraint can be relaxed. Constraint set (7) limits the global usage of edge servers to be no more than $|E|$. Constraint set (8) ensures that the computing capacity of edge server $k \in E$ is not overloaded. Constraint set (9) controls the physical devices d satisfied by the data collection mechanism (in terms of telemetry data collected). In turn, constraint set (10) accounts for the number of NDT models satisfied. Note that

an NDT model is only satisfied *iff* all its physical devices have been satisfied. Constraint sets (12)–(16) define the domain of output variables. Last, the objective function (1) maximizes the number of NDT models satisfied by the model concerning constraint sets (2)-(16).

IV. DigiNet Heuristic Approach

To tackle the DigiNet complexity and come up with near-optimum solutions, we introduce the heuristic strategy presented in Alg. 1. To maximize the number of NDT models satisfied, small NDT models are prioritized first (i.e., fewer D^m devices). The heuristic starts sorting the set M w.r.t $|D^m|$ (line 1) and then it iterates through the ordered set $\mathcal{Q}$ (lines 3-17). Then, for each physical device (line 4) and its required telemetry data (line 6), the DigiNet heuristic tries to assign an appropriate network flow to collect monitored data (line 7). A network flow is considered a valid candidate to collect telemetry data *iff* (i) the physical device is in the routing path (i.e., $d_i \in \mathcal{P}(f)$), (ii) the edge server is in the routing path, and (iii) the edge server comes after the device in the routing path.

Algorithm 1: DigiNet heuristic strategy.

```
1  Q ← Sort set M in ascending order w.r.t |D^m|
2  S ← ∅
3  foreach NDT model m ∈ Q do
4      foreach physical device d ∈ D^m do
5          S* ← ∅
6          foreach telemetry data v ∈ R_d^m do
7              foreach network flow f ∈ F such that
                 d_i ∈ P(f) and ∃ d_j^* ∈ P(f) ∈ E such that
                 i ≤ j do
8                  if S(v) ≤ K_f and C^m(m) ≤ C^e(d*) then
9                      S* ← network flow f collects item v
10                     K_f ← K_f − S(v)
11                     C^e(d*) ← C^e(d*) − C^m(m)
12                     break
13         if any telemetry data v ∈ R_d^m is not satisfied in S*
             then
14             backtrack the state of K_f : (∀f ∈ F) and
                 C^m(m) based on solution S*
15             break and go to the next NDT model
16     if all telemetry data in d ∈ D^m are satisfied then
17         S ← S ∪ S*
18 return S
```

Whenever a valid network flow is found, the DigiNet heuristic verifies the capacity constraints of network flows and edge servers (line 8). In case the telemetry data fits into network flow, and the edge server can host the NDT model, then capacities are updated, and the partial solution is kept in memory. In case any telemetry data is not satisfied, DigiNet backtracks the solution to a safe state (lines 13-14). Finally, if all telemetry data and devices of the NDT model are satisfied, we store the partial solution in S. The full description of the DigiNet heuristic is described in Algorithm 1. DigiNet heuristic has polynomial time complexity $O(M \cdot D \cdot V \cdot F)$.

V. Evaluation

A. Setup

The proposed model was run using IBM *CPLEX Optimization Studio* 22.11 to obtain optimal solutions, while the proposed heuristic approach was implemented using Java language. Experiments were performed on an AMD Threadripper 3990X processor, 32 GB RAM, Ubuntu 22.04 machine. Different physical network instances with 100 nodes were generated using Brite [23] / Barabási-Albert model [24].

We vary the amount of available network flows from 50 to 300 and the available space to embed telemetry items in network flows (i.e., K_f) from 50 to 500 Bytes. Network flows are routed using the shortest path. Further, we assume forwarding devices have from 2 to 8 possible telemetry items to export, varying its size $S(v)$ uniformly from 2 to 20 Bytes [25]. Each NDT model comprises a subset of D. We varied the size of $D^m \subseteq D$ from 5 to 15. We chose them randomly from D. Note that other strategies might be applicable, such as building induced sub-graphs. Each device d demands R_d^m telemetry data from the G. We varied that from 1 to 8 (max number of telemetry in a device). The number of NDT models being provisioned is varied from 5 to 30.

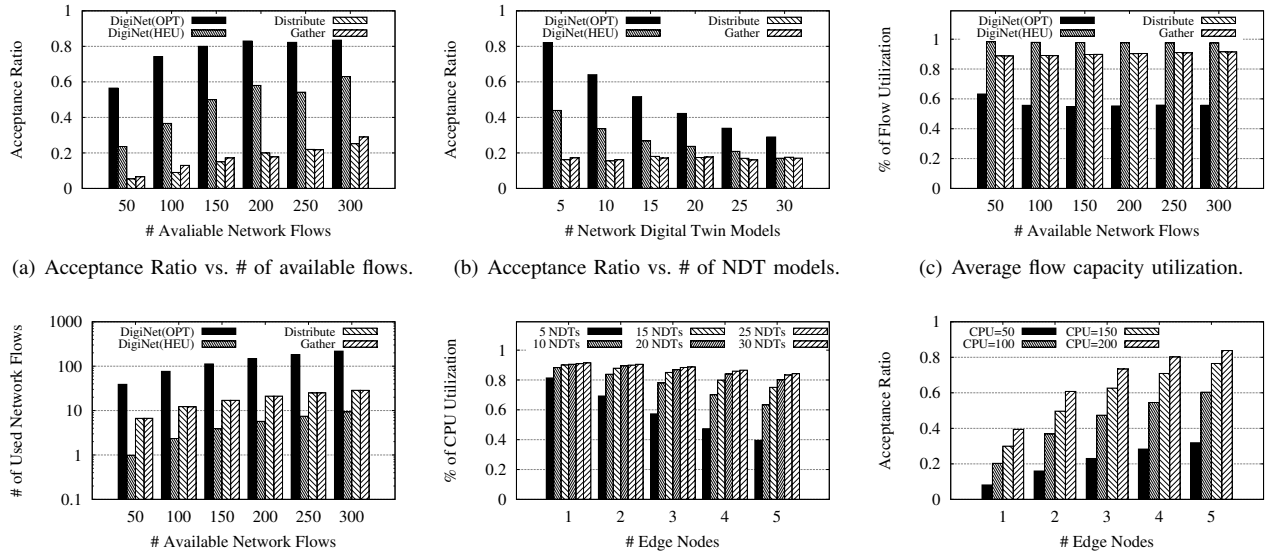
Last, we varied the number of edge servers in the physical infrastructure from 1 to 5. Each edge server has a computing capacity ranging from 50 to 200, while NDT models demand computing units varying uniformly from 30 to 50. Using the t-test method, we found that 30 runs of each experiment are enough to achieve a confidence level of 95% or higher. We compare DigiNet against (i) the optimal solution (`OPT`), (ii) and to the recent state-of-the-art INT orchestration work `Distribute` and `Gather` [18]. Our implementation has been released in open source for research reproducibility.[5]

B. Results

We analyze the quality of the proposed approach by evaluating: (*i*) percentage of NDT models deployed (acceptance ratio); (*ii*) resource usage of network flows; (*iii*)resource usage of edge servers.

Acceptance ratio. Figure 4(a) illustrates the percentage of NDT models deployed in the infrastructure for an increasing number of available network flows (from 50 to 300). While the Optimal DigiNet solution can cope with up to 83% of NDT requests (with 300 network flows), the heuristics approaches struggle to deliver more than 35% (e.g., Gather heuristic w/ 300 network flows). In contrast, our DigiNet heuristic can produce solutions up to 75% near the optimal value. Observe that for fewer network flows (e.g., 50), the difference between the optimal to the heuristic is up 10x. As we increase the number of available network flows, the acceptance ratio increases, and the relative difference between the optimal and heuristics reduces to 3x. As more network flows are available, it makes the search space wider as there is a higher chance of finding a network flow that can collect telemetry data and

[5]Implementation of our simulation is available at: https://anonymous.4open.science/r/DigiNet-B6AB/README.md



(a) Acceptance Ratio vs. # of available flows.

(b) Acceptance Ratio vs. # of NDT models.

(c) Average flow capacity utilization.

(d) Average # of utilized flows.

(e) Average CPU utilization at edge servers.

(f) Acceptance Ratio vs. # of edge servers.

Figure 4. Sensitivity analysis and comparison of the DigiNet model and the proposed heuristic against state-of-the-art and optimal solutions.

carry them to an edge server. Next, Figure 4(b) illustrates the acceptance ratio for a varied number of simultaneous NDT models being provisioned in the infrastructure. In this experiment, the larger the set of NDT models, the lower the acceptance ratio. Observe that in this case, the acceptance reduces to around 17% for all heuristics approaches), while the optimal value approaches 28%.

Resource usage of network flows. Figure 4(c) and Figure 4(d) illustrate network flow resource usage. First, we evaluate the network flow's capacity usage when collecting telemetry data for NDT models. We observe that the average flow utilization is almost constant for an increasing number of available flows. We observe that the optimal solution utilizes on average 55% of available resources, while Gather and Distribute heuristics use on average 90%, and the DigiNet heuristic up to 97%. That behavior illustrates the optimal ability of our model to select appropriate NDT models to attend. In contrast, heuristics such as Gather and Distribute are unaware of NDT models and, therefore, can collect telemetry data that NDT models do not require. Our proposed heuristic (DigiNet) strives to attend to the higher number of NDT models and, therefore, utilizes more networking resources. Next, Figure 4(d) illustrates the number of network flow solutions used. While the Optimal solution has a lower flow utilization rate (shown in Figure 4(c)), it does utilize a higher number of active flows in the infrastructure (up to 24x more than the heuristics). Similarly, the DigiNet heuristic has a lower number of flows utilized.

Resource usage of edge servers. Figure 4(e) depicts the average CPU utilization ($C^e(d)$) of edge servers. In this experiment, we varied the number of NDT models being provisioned using the optimal solution. The more edge servers available, the lower the average CPU utilization. For instance, with 5 edge servers and 5 NDT models, the CPU utilization is below 40%. Complementary, Figure 4(f) shows the effect of edge servers with different computing capacities on the acceptance ratio. As expected, having more CPU and edge servers increases the acceptance ratio, as it is easier to find feasible solutions when more resources are available.

VI. Related Work

Next, we review recent efforts towards resource provisioning in NDT. We focus on three aspects: (*i*) NDT-assisted resource provisioning, (*ii*) NDT provisioning, and (*iii*) NDT synchronization. Finally, we compare DigiNet to related work.

NDT-assisted Resource Provisioning. Much of the recent progress in NDT has been devoted to building DT models to assist resource provisioning. [8] explores a dynamic DT of aerial-assisted Internet of Vehicles to capture the time-varying resource supply and demands so that unified resource scheduling and allocation can be performed. In turn, [9] introduces a device-to-device (D2D) communication-aided digital twin edge network in the context of Industrial Internet of Things. The digital twin is used to optimize D2D communication and to assist resource-limited IoT devices to achieve normal communication. In the same IoT context, [10] focuses on the resource allocation process of network slicing for highly personalized IoT services. [11] proposes an NDT that utilizes DT to establish an efficient mapping between IoT and digital systems. [26] integrates the DT and the mobile edge computing technologies, allowing base stations to assist local computation and thereby reducing the transmission delay.

NDT Provisioning. Xiao et al. [14] argue that efficiently positioning servers to host NDTs in distributed infrastructures is key for successfully realizing NDT. After describing the high computational complexity required for determining optimal locations for NDT hosts, the authors present an approach based

on evolutionary algorithms that provides efficient solutions in bounded time. Lu et al. [12] discuss the potential benefits and challenges of provisioning NDTs at the edge, presenting a Deep Reinforcement Learning algorithm that determines initial hosts for NDTs and rearranges the NDTs across the infrastructure over time based on changing network conditions.

NDT Synchronization. Zheng et al. [27] advocate that efficient synchronization between physical and digital components is critical to ensure that NDTs make accurate decisions. The problem of NDT synchronization becomes even more complex in dynamic topologies such as vehicular networks, where nodes are non-stationary and network connectivity is prone to instability. Based on such observation, the authors address this challenge through a novel framework that employs a game-based approach to select the optimal network path for NDT synchronization. Tan et al. [28] focus on optimizing NDT synchronization in collaborative scenarios. In addition to highlighting the importance of synchronizing DTs with their physical counterparts, the authors make a case for inter-twin communication, presenting a cloud-based data-sharing platform that creates a city-wide NDT where individual vehicle DTs communicate for improved decision-making.

Discussion. Despite the significant contributions within the literature to optimize where NDTs are deployed and how they stay synchronized with their physical counterparts, none of the existing efforts have presented an in-depth analysis of the network aspects of NDT provisioning and communication. Our work on DigiNet fills this gap through a threefold contribution. First, we provide a comprehensive analysis of the NDT network requirements and the performance implications incurred by different NDT allocation and communication approaches. Second, we present the first attempt to model a low-cost NDT synchronization problem that opportunistically embeds telemetry data within existing network packets. Third, we shed light on multiple relevant open research challenges in the field, as elaborated in the section.

VII. Research Challenges and Future Directions on NDT Resource Allocation

NDT Placement. Mapping DTs to physical computing platforms resembles well-studied placement problems in the networking domain. However, NDTs have specific constraints. For example, each digital representation has a constant two-way data flow from the physical representation, ensuring the synchronization between physical and digital counterparts. In this context, multiple NDT representations might depend on the same set (or subset) of physical devices. Different from recent models [29], [30] that follow a 1:1 allocation mapping, NDT can extrapolate mapping ratios to 1:N or N:M. For instance, a physical router can be mirrored to multiple DT models simultaneously (i.e., a 1:N ratio). Another aspect to be considered is the decomposition of complex NDT representations. Running monolithic NDT models might be cumbersome and resource intensive. By decomposing NDT representations, complex models can be run as multiple sub-models in a chain, requiring (*i*) to decompose basic and/or functional models, and (*ii*) to chain in input/output of correlated sub-models. Decomposing a model implies defining, for instance, its scope. That is, we can have local models – e.g., an NDT model representing a single device; or a global model – e.g., an NDT model representing the interactions between different devices. Therefore, orchestrating the allocation of resources to make efficient placement of NDT to physical computing platforms while ensuring that data flow forwarding does not affect production network traffic is challenging.

Data Collection Orchestration. NDTs require an incoming data flow from physical devices. Therefore, it is important to design efficient data collection mechanisms. For instance, we can apply adaptive sampling at the physical source device or in-network compression mechanisms to reduce the burden of transmitting a huge volume of data between physical and digital instances. Yet, we can design tailored and more space-efficient encapsulation protocols, similar to INT approaches, to reduce the transmission overhead. As discussed, the data collection problem in the NDT environment is an NP-hard problem that still demands novel algorithmic approaches to be solved efficiently. We have shown that relying on existing INT monitoring approaches to collect physical telemetry data in the NDT domain does not lead to high-quality solutions.

NDT Model Resource Usage Optimization. NDT implements basic and functional models. There are a variety of simulators (e.g., see Table I) that could be used or extended to implement basic functions. Functional models can be implemented in supervised and unsupervised AI techniques. However, as NDT might represent large-scale network infrastructures, it is challenging to design space- and time-efficient models/simulations amenable for efficient deployment in production environments. There is no straightforward answer to estimate how much computational resources current simulators/models require, as it varies from one simulation to another. The computation resources depend on the simulation scale, duration, and the complexity and granularity of the devices being modeled. Therefore, it is essential to conduct benchmarking and performance analysis to determine and infer necessary computing demands. With the advances of programmable forwarding devices (e.g., programmable routers/SmartNICs), NDT models or parts of them could be run on such devices. That would bring the benefit of running digital instances closer to physical devices. However, in-network programmable network devices have limited resources and computing capabilities (e.g., limited memory access). Therefore, placing NDT models on these devices is still challenging and requires substantial progress. It may be possible to leverage hardware offloading to ease the impact of such limitations, such as embedding missing functionalities on FPGA boards.

Mobility- and Security-Aware NDT Deployment. Many physical devices might change their deployment location at the physical network. For instance, software network function instances might change their deployment location based on existing demands. Another example is a 5G network slice that moves its deployment location according to the user's mobility. In such cases, the orchestration of data collection and

the embedding mechanism should be aware of such mobility to reduce NDT service interruptions. Also, there are many security concerns involving the deployment of NDTs. First, the data flow between NDT and physical entities must be secure against integrity and confidentially attacks. Second, sensitive data from physical devices should not be shared with NDTs inadvertently. Third, NDT instances, when necessary, should not share the same physical computing premises. Therefore, novel resource optimization schemes to provision NDT securely are an open research challenge.

VIII. Closing Remarks

NDT has opened up an avenue of research directions and application possibilities for AI-based networks. However, realizing the full potential of NDT will require novel resource allocation strategies that can deal with the large-scale nature of existing networking infrastructure. In this paper, we formalized the DigiNet problem, which leverages INT to collect telemetry data to NDT models within existing network packets. We introduced a MILP model and a scalable heuristic to solve the problem. While our approach outperforms state-of-the-art heuristics (e.g., factor 10 w.r.t NDT models deployed), it is still limited to (*i*) static solutions over time (i.e., NDT models do not change), and (*ii*) our heuristic still produces sub-optimal solutions with a difference of up to 30% from the optimal value. Addressing these limitations from the theoretical and operational point of view is part of our future work.

References

[1] E. Coronado, R. Behravesh, T. Subramanya, A. Fernàndez-Fernàndez, M. S. Siddiqui, X. Costa-Pérez, and R. Riggio, "Zero touch management: A survey of network automation solutions for 5g and 6g networks," *IEEE Communications Surveys Tutorials*, vol. 24, no. 4, pp. 2535–2578, 2022.

[2] H. H. H. Mahmoud, A. A. Amer, and T. Ismail, "6g: A comprehensive survey on technologies, applications, challenges, and research problems," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, p. e4233, 2021.

[3] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6g: Taxonomy, research challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2137–2150, 2022.

[4] M. C. Luizelli, R. Canofre, A. F. Lorenzon, F. D. Rossi, W. Cordeiro, and O. M. Caicedo, "In-network neural networks: Challenges and opportunities for innovation," *IEEE Network*, vol. 35, no. 6, pp. 68–74, 2021.

[5] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[6] P. Öhlén, C. Johnston, H. Olofsson, S. Terrill, and F. Chernogorov, "Network digital twins – outlook and opportunities," *Ericsson Technology Review*, vol. 2022, no. 12, pp. 2–11, 2022.

[7] H. Ahmadi, A. Nag, Z. Khar, K. Sayrafian, and S. Rahardja, "Networked twins and twins of networks: An overview on the relationship between digital twins and 6g," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 154–160, 2021.

[8] W. Sun, P. Wang, N. Xu, G. Wang, and Y. Zhang, "Dynamic digital twin and distributed incentives for resource allocation in aerial-assisted internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5839–5852, 2022.

[9] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation for d2d-aided digital twin edge networks in 6g industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 7228–7236, 2023.

[10] L. Tang, Y. Du, Q. Liu, J. Li, S. Li, and Q. Chen, "Digital-twin-assisted resource allocation for network slicing in industry 4.0 and beyond using distributed deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 10, no. 19, pp. 16 989–17 006, 2023.

[11] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4968–4977, 2021.

[12] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6g," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 219–16 230, 2021.

[13] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[14] L. Xiao, D. Han, T.-H. Weng, S. Chen, H. Deng, A. Souri, and K.-C. Li, "An evolutive framework for server placement optimization to digital twin networks," *International Journal of Communication Systems*, vol. 36, no. 14, p. e5553, 2023.

[15] Y. Zhou, R. Zhang, J. Liu, T. Huang, Q. Tang, and F. R. Yu, "A hierarchical digital twin network for satellite communication networks," *IEEE Communications Magazine*, 2023.

[16] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, G. Karakostas, H. Wu, and X. Shen, "Digital twins from a networking perspective," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 525–23 544, 2022.

[17] T. P. A. W. Group. (2009, Jun.) In-band network telemetry (int) dataplane specification. [Online]. Available: https://github.com/p4lang/p4-applications/blob/master/docs/INT.pdf

[18] J. A. Marques, M. C. Luizelli, R. I. T. Da Costa, and L. P. Gaspary, "An optimization-based approach for efficient network monitoring using in-band network telemetry," *Journal of Internet Services and Applications*, no. 1, p. 16, Jun 2019.

[19] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Digital twin network: Concepts and reference architecture," Internet Draft, Internet Engineering Task Force, 2023. [Online]. Available: https://www.ietf.org/archive/id/draft-irtf-nmrg-network-digital-twin-arch-04.html

[20] P. S. Souza, T. Ferreto, and R. N. Calheiros, "Edgesimpy: Python-based modeling and simulation of edge computing resource management policies," *Future Generation Computer Systems*, vol. 148, pp. 446–459, 2023.

[21] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[22] M. Yu, "Network telemetry: towards a top-down approach," *ACM SIGCOMM Computer Communication Review*, vol. 49, no. 1, pp. 11–17, 2019.

[23] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: an approach to universal topology generation," in *MASCOTS 01*, Aug 2001, pp. 346–353.

[24] R. Albert and A.-L. Barabási, "Topology of evolving networks: Local events and universality," *Physical Review Letters*, vol. 85, pp. 5234 – 5237, Dec 2000.

[25] T. Pan, E. Song, Z. Bian, X. Lin, X. Peng, J. Zhang, T. Huang, B. Liu, and Y. Liu, "Int-path: Towards optimal path planning for in-band network-wide telemetry," in *IEEE INFOCOM 19*, Apr 2019, pp. 1–9.

[26] Y. He, M. Yang, Z. He, and M. Guizani, "Resource allocation based on digital twin-enabled federated learning framework in heterogeneous cellular network," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1149–1158, 2023.

[27] J. Zheng, T. H. Luan, Y. Zhang, R. Li, Y. Hui, L. Gao, and M. Dong, "Data synchronization in vehicular digital twin network: A game theoretic approach," *IEEE Transactions on Wireless Communications*, 2023.

[28] C. Tan, X. Li, L. Gao, T. H. Luan, Y. Qu, Y. Xiang, and R. Lu, "Digital twin enabled remote data sharing for internet of vehicles: System and incentive design," *IEEE Transactions on Vehicular Technology*, 2023.

[29] Y. Chen, F. Zhao, X. Chen, and Y. Wu, "Efficient multi-vehicle task offloading for mobile edge computing in 6g networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4584–4595, 2022.

[30] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6g," *IEEE Communications Surveys Tutorials*, vol. 24, no. 1, pp. 1–30, 2022.