Addressing the Scalability of Network Digital Twins: A Network Sampling Approach

Mounir Kellil
Université Paris-Saclay,
CEA, List,
F-91120, Palaiseau, France
mounir.kellil@cea.fr

Siwar Ben Hadj Said
Université Paris-Saclay
CEA, List,
F-91120, Palaiseau, France
siwar.benhadjsaid@cea.fr

Minh-Thuyen Thi
Université Paris-Saclay
CEA, List,
F-91120, Palaiseau, France
minh-thuyen.thi@cea.fr

Christophe Janneteau
Université Paris-Saclay
CEA, List,
F-91120, Palaiseau, France
christophe.janneteau@cea.fr

Alexis Olivereau
Université Paris-Saclay
CEA, List,
F-91120, Palaiseau, France
alexis.olivereau@cea.fr

Abstract— **With the increasing complexity of mobile communication networks like 5G/6G networks, characterized by the diversity of network devices, technologies, and applications, advanced network management solutions are needed to ensure real-time network optimization with risk-free decision making operations (e.g., safe network reconfiguration). To achieve this objective, a Network Digital Twin (NDT) paradigm represents an attractive perspective, enabling the manipulation of the virtual counterpart of a real communication network. Nonetheless, generating a Digital Twin of a complex network, comprising thousands of heterogeneous devices and highly dynamic network characteristics (resource usage, network topology, link quality, etc.) poses a serious scalability problem. This paper aims at addressing the scalability problem for the generation of a Digital Twin of a complex network like a 5G/6G network. In particular, the paper proposes a sampling approach in conjunction with a structured network information representation, as well as *zoom-in*/out operations to enable a modular generation of the NDT.**

Keywords—Network sampling, network management, Digital Twin

I. Introduction

Mobile communication networks are experiencing a paradigm shift thanks to the recent advances in 5G and future 6G networks, characterized by an exponential growth of communication network capacities and features, including network heterogeneity, large bandwidth, ultra-low latency, and decentralized decision-making operations. In the same vein, new generations of network applications have emerged, such as immersive applications, V2X, and smart factory. The deployment of such applications is accelerated by the increasing integration and adoption of IoT devices and Artificial Intelligence (AI) services in today's mobile communication networks. Nonetheless, this mobile communication network ecosystem typically operates in a highly dynamic environment, characterized by frequent network topology changes as well as unpredictable network resource consumption patterns.

To ensure supervision, or more generally, management, whether fine-grained or synthetic, of such communication networks, various network management services like SNMP, CMIP, LwM2M, CoMI, and SDN can be used, alone or combined [1]. Furthermore, such services can adopt recent advances in automated network management, including the usage of Machine Learning (ML) techniques in order to accommodate the ever increasing complexly of mobile communication networks [1][2].

However, network management services do not intrinsically guarantee risk-free (or error-free) (re)configuration of complex mobile communication networks like 5G/6G networks. This is particularly true when it comes to performing real-time optimization of the communication network, network tests in operational mode (what-if analysis) [3][4][5], or proceeding with real-time network upgrade or extension. In such scenarios, any improper network (re)configuration (e.g., misconfiguration of routes and underestimation of allocated resources) would result in various operational risks such as network service disruptions and degraded performance. This issue, though, typically incurs additional costs for implementing failure recovery strategies and operations.

To prevent this problem, several recent research initiatives have explored the adoption of the Digital Twin concept to the communication networks. This approach is commonly referred to as Network Digital Twin (NDT) [4]-[7]. The NDT is a Digital Twin of the real (or physical) network, where a digital copy of the real network can be manipulated without risk, which makes it possible in particular to visualize or predict the state of the real network if this or that network configuration is to be applied.

Despite the attractive perspective of deploying an NDT solution to facilitate the supervision of complex communication networks and streamline their effective management, ensuring this objective in a scalable fashion is not straightforward. Indeed, frequent NDT synchronizations with the real communication network poses a scalability problem when dealing with complex networks (e.g. too large number of network entities, highly dynamic topologies, large volume of information per node or per network link), especially when each network information is to be reported on the NDT's side.

This paper aims to address the scalability challenges in designing a Digital Twin for complex communication networks like 5G and future 6G networks. In particular, the paper proposes a network sampling concept in conjunction with a structured network information representation, as well as *zoom-*

978-3-903176-66-9 ©2024 IFIP

in/out operations, in order to ensure efficient adoption of NDTs in emerging mobile communication networks. The rest of the paper is organized as follows. Section II discusses various solutions for supporting NDT in emerging mobile communication networks. Section III describes the network information structure, the network sampling concept, and the *zoom-in*/out modes. Section IV exploits the concepts described in section III to present an incremental sampling scheme, used to generate the full NDT of the real communication network. Section 0 presents preliminary evaluation results of the information overhead associated to the proposed sampling solution. Finally, section 0 concludes the paper and discusses future research directions.

II. State-of-the Art

Mobile communication networks typically experience frequent topology changes, caused by various factors, including the presence of error-prone wireless channels, user behavior (e.g., mobility and network connection habits), routing protocols, and resource allocation services. Furthermore, such communication networks may be subject to various coexisting traffic patterns, ranging from (predictable) periodic reporting and control events (e.g. protocol-specific, telemetry, command/control applications) to streaming applications (e.g., audio/video, immersive environments, etc.), and encompassing event-driven data traffic (e.g., ambient applications, alert applications, etc.). Virtualizing such a complex communication network ecosystem to expose its real-time Digital Twin is inherently challenging, especially that the network states (e.g., topology, resource usage, etc.) are perpetually changing, hence the need for a scalable approach to efficiently support Digital Twins for mobile communication networks.

Various research initiatives have explored the adoption of Digital Twins for mobile communication networks like 5G and 6G (e.g., [5][6]-[8]). However, to our knowledge, few papers have considered the scalability problem, even though the question of modular implementation has been pointed out by a number of authors (e.g., [8]). In the Digital Twin literature, the authors in [10], were among the first to consider the scalability problem in implementing Digital Twins. However, their work focused on the smart manufacturing domain (Digital Twins of machines, production cells, and entire manufacturing facilities), where the communication component is viewed as a means for generating a Digital Twin, rather than as a target component of the Digital Twin itself. The authors addressed the scalability problem for data acquisition and proposed a stepwise method that breaks down the implementation into distinct, manageable steps, allowing teams to follow a defined pathway from recognizing devices to establishing communication and integrating data.

In [11], the authors present a Digital Twin system for mobile networks. The proposed system considers three key elements from the mobile network: mobile users, base stations, and wireless environments. For each element, a virtual version is generated and configured with real-world data by modeling its principles and parameters. These virtual elements are assembled to form the whole Digital Twin system. The paper addresses the scalability problem using four concepts: 1) generative models like GANs (Generative Adversarial Networks) and VAEs (Variational Autoencoders) to simulate the behavior of millions of mobile users and network components, 2) Parallel Processing techniques to handle simulations of a large number of mobile users simultaneously, 3) Modular Digital Twin creation for the virtualized elements, and 4) real-time adjustment of the Digital Twin based on simulation.

A methodology for creating Mobile Network Digital Twins (MNDT) for 5G networks was proposed in [12] by modeling physical network elements and their interactions. Network elements include physical devices, communication links, operating environment (e.g., traffic patterns and user behavior), and various 5G-specific network functions. To address the scalability problem, the methodology emphasizes automated data acquisition and modeling, using agents that collect network information (topology and network parameters) and operational data (current status and performance metrics of devices) from the physical network.

In sum, the reviewed solutions addressed the NDT scalability problem using network modeling along with network agents, which facilitate automated data acquisition and self-optimization operations. Building on such concepts, our solution takes a step further by addressing the information overhead problem related to maintaining up-to-date information on the NDT's side, challenge that has not been sufficiently explored in the state of the art.

In this paper, we propose a novel scheme for improving the data acquisition process and, incidentally, streamlining network modeling operations. To this aim, we propose a network sampling scheme. This scheme enables both selective and incremental data acquisition from the real communication network. Upon each network sampling phase, a *zoom-in*/out operation over the real communication network is performed, depending on whether more (*zoom-in*) or less (*zoom-out*) details need to be exposed for a specific segment (*sample*) of the real communication network. This *zoom-in*/out scheme is facilitated thanks to network information categorization (device and link) and structuring (object→resource(s)). Each object being a device or a link. The main advantage of the proposed solution is that it proceeds with a modular NDT generation, where each sampling phase targets a part of the communication network. In this paper, modularity is expressed both horizontally (size of the network sample, i.e., the number of objects composing the sample (i.e., devices and links)) and vertically (number of captured resources for each object of the sample). This solution is particularly useful when there is a need for saving network resources (bandwidth, memory, CPU, energy), while managing effectively the communication network (i.e. collect only the relevant network parameters). Additionally, this modular NDT generation also enables an incremental generation of the complete NDT, if necessary.

III. Solution Description

For the sake of simplicity and clarity, figure 1 presents a simplified view of the NDT system in the context of a mobile communication network. This figure is consistent with the NDT reference architecture specified by the IETF and ITU-T, and which we discussed in a previous paper [18]. Within the NDT system, there is a key component called *NDT manager*, which is responsible for generating and updating the NDT associated to the real communication network or a part of it, called a *sample*. As explained in the next section, the NDT manager views each network sample as a set of *objects*. Figure 1 also shows that the NDT manager interacts with three functional components: the *NDT prediction module* (e.g., a Machine Learning agent [4]-[9]), the *NDT GUI* (to display the NDT [9]), and the *network management module* (a conventional network management system like SNMP, CMIP, LwM2M, SDN, etc.) [1][2].

As shown in figure 1, our NDT solution requires a network management system to collect the network information needed for the generation of the NDT. Therefore, the NDT solution should interoperate with existing network management systems. In particular, the NDT should support existing network management protocols (such as SNMP, NETCONF, and RESTful APIs). To this aim, a modular plugin architecture can be adopted on the NDT's interface with the network management system. With this modular plugin architecture, different network management protocols can be enabled/added or disabled/removed. Alternatively, an abstraction layer can be implemented on that NDT's interface to translate various network management protocols (like SNMP, RESTful APIs, etc.) into a unified interface for the NDT. This allows the NDT to remain agnostic of the underlying protocol specifics. The details of this interoperability component is out of the scope of this paper. Nonetheless, the conclusion section, we point out some open issues and research directions relating to this topic.

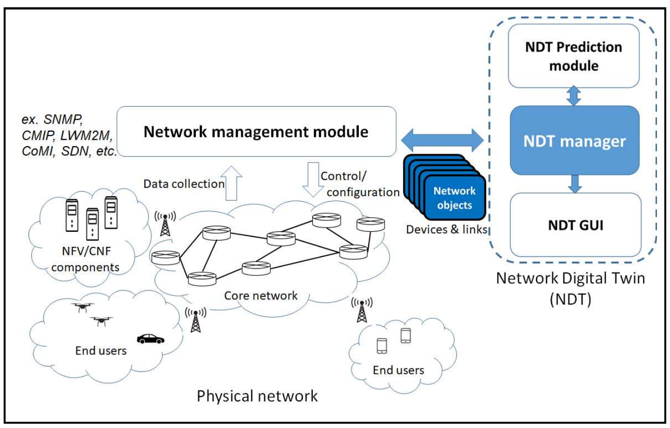

Fig. 1. High-level view of an NDT system for a mobile communication network

A. Network information representation

To facilitate the identification of the network entity at the NDT level, the network information will follow a hierarchical structure: *Sample*→*Object*→*Resource*, adhering to an Object-Resource model similar to the one used in existing network data models, such as the LwM2M data model [13]. Other alternate models like FIWARE data models [14] can be used as well. Such a hierarchically structured representation of network information also aims to facilitate fine-grained supervision and configuration of the communication network at its NDT counterpart. In addition, a network object can be of two possible types: *network device* or *network link*.

As shown in figure 2, a network sample can be composed of one or more network objects (devices and/or network links), each with zero, one or more associated resources. A network resource represents any type of information that can characterize a network object. For a network device like a router, the associated resource could be a network interface's IP address, CPU usage ratio, a virtual resource, etc. For a network link, the resource could be the PHY communication technology, SNR, ratio of available bandwidth, types of transport protocols, a virtual resource, etc.

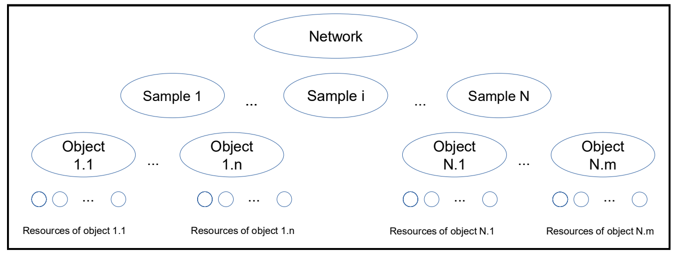

Fig. 2. Hierarchical representation of network objects and associated resources

Furthermore, we consider that, from the standpoint of a communication network, any external physical object (i.e., aside from a network object) that is effectively influencing, in one way or another, the behavior of the communication network (e.g., network user, building, vehicle, etc.) must have a signature on one or more network objects (device(s) and/or link(s)), where the signature could be expressed in the form of a resource value of the concerned network object(s). In light of this, any external object that does not impact the communication network will not have a signature in the set of objects captured on the NDT side. In this case, this external object will simply be ignored.

B. Network sampling

The choice of the network sample can be driven by different selection strategies. For instance, the selection may be random, or it may also be based on recent events that occurred in the network (e.g., hardware/software failures, abnormally low/high values of network parameters, etc.). In addition, at each sampling phase, a *zoom-in* or *zoom-out* operation is performed in parallel.

The objective behind a *zoom-in* mode is to focus on a fine-grained representation of the sample that is being collected by the NDT manager (e.g., collect all the available resources of each of the objects of the sample). This way, a set of nodes and links can be dynamically selected based on certain criteria (e.g., high CPU usage, critical links, anomaly analysis, etc.) to monitor their performance metrics more closely. On the other hand, a *zoom-out* mode is useful when a high-level view of a specific sample of the entire network is needed for several reasons. For instance, it helps to free up resources at the NDT manager level, capture the list of current network nodes and links, and obtain a snapshot of high-level trends or patterns at the sample or network scale (e.g., total CPU usage, average traffic throughput, etc.).

In the network sampling phase, the NDT manager generates a request (*request_desc*) to fill in a local structure called *sample descriptor* (*desc_S*). An example of a sample descriptor is provided in figure 3. This figure shows a simplified UML-alike view describing the object, the resource, and their associated symbolic connection ('1' vs '*'). This symbolic connection shows that one object can have zero or many resources. Note that both the objects and resources can have different attributes than those exemplified in figure 3, potentially being either subsets or supersets of the illustrated attributes.

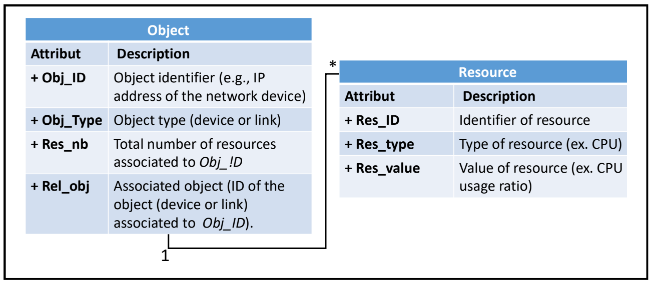

Fig. 3. Example of a sample descriptor

On the other hand, the request *request_desc* has a structured *Expression* field, enabling conditional collection of objects and associated resources for a given network sample. This can be achieved using a number of possible network management protocols like NETCONF [15], RESTCONF [16] with their YANG-based data model [17]. In addition, the request includes a flag (*zoom_flag*), which indicates whether the network sample is to be generated following a *zoom-in* or a *zoom-out* mode. A simplified description of the request is provided in figure 4, for illustrative purposes. The exact form of such a request is outside the scope of this paper. Figure 4 shows an example of an *Expression* field of *request_desc* for both *zoom-in* and *zoom-out* modes using NETCONF requests. In the *zoom-in* mode, the NDT manager requests devices with CPU usage greater than 80% along with their associated links, including all the resources of the selected devices and all the resources of their associated links. In *zoom-out* mode, only the devices with CPU usage greater than 80% and their associated links are requested.

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="101">
  <get>
    <filter>
      <request_desc>
        <Expression>
          <condition>
            <device>
              <cpu-usage>greater than 80%</cpu-usage>
            </device>
          </condition>
          <result>
            <include>
              <devices>
                <all-resources>true</all-resources>
              </devices>
              <links>
                <all>true</all>
                <all-resources>true</all-resources>
              </links>
            </include>
          </result>
        </Expression>
        <zoom_flag>zoom-in</zoom_flag>
      </request_desc>
    </filter>
  </get>
</rpc>
```

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
message-id="102">
  <get>
    <filter>
      <request_desc>
        <Expression>
          <condition>
            <device>
              <cpu-usage>greater than 80%</cpu-usage>
            </device>
          </condition>
          <result>
            <include>
              <devices>
                <status>true</status>
              </devices>
              <links>
                <all>true</all>
                <status>true</status>
              </links>
            </include>
          </result>
        </Expression>
        <zoom_flag>zoom-out</zoom_flag>
      </request_desc>
    </filter>
  </get>
</rpc>
```

Fig. 4. Example of *Expression* field in *zoom-in* (left) and *zoom-out* (right) modes

Figure 5 presents a high-level view of *zoom-in* and *zoom-out* operations, which are triggered by *event_A* and *event_B,* respectively. *Event_A* could be, for exmaple, any event that requires network sampling in a *zoom-in* mode, like anomalies detected in specific network areas, and reported (e.g., by the network manager) to the NDT manager. Similarly, *event_B* could represent any event that justifies network sampling in a *zoom-out* mode, such as the end of *event_A*. In addtion, figure 5 shows that the NDT manager interacts with a conventional network manager using various possible network management protocol messages like the NETCONF requests of figure 4. The request message (*request_desc*) in figure 5 is depicted in a concise format, summarizing the details of NETCONF request message, shown in figure 4.

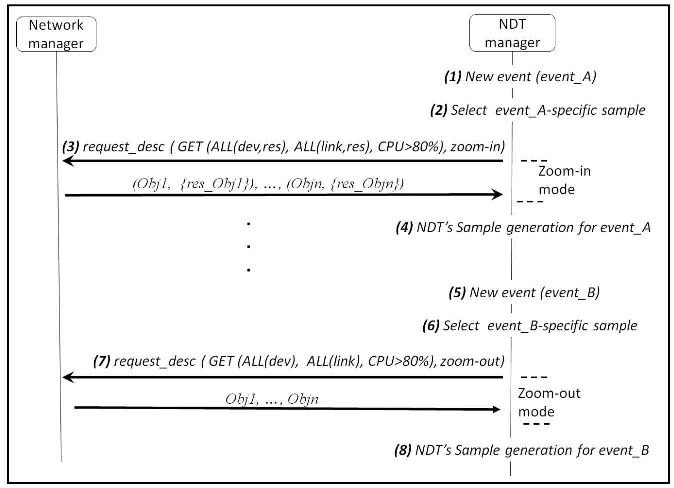

Fig. 5. Example of *zoom-in* and *zoom-out* operations

IV. Application of Network Sampling for the Generation of the NDT of a Full Network

This section explains the incremental network sampling procedure that enables the NDT manager to incrementally generate the NDT (i.e., real-time picture of a full communication network) using successive selective samplings in a *zoom-in* mode.

A. *Initial step – NDT initialization*

To perform the incremental sampling, the NDT manager is initialized with the list of all the objects of the full network, i.e., all the network devices and all the associated network links. Although not critical, this initialization phase enable the NDT manager to get a high-level view of the actual status (a first glance) of the real communication network. The initial step consists in transmitting a *request_desc* query from the NDT manager to the network manager. This query includes an *Expression* field, asking for all the objects of the communication network (devices and links) and having a *zoom_flag* set to *zoom-out.*

B. *Incremental NDT generation*

The incremental NDT generation will take place through a set of successive network sampling operations, each initiated by a *request_desc* query, with a *zoom_flag* set to *zoom-in* mode. Each newly generated sample S_i is chosen so that its sample descriptor *desc_Si* is different from the descriptor of any previous sample of the current incremental NDT generation procedure. This enables to ensure the convergence of the generation procedure towards the NDT of the complete communication network (cf. figure 6).

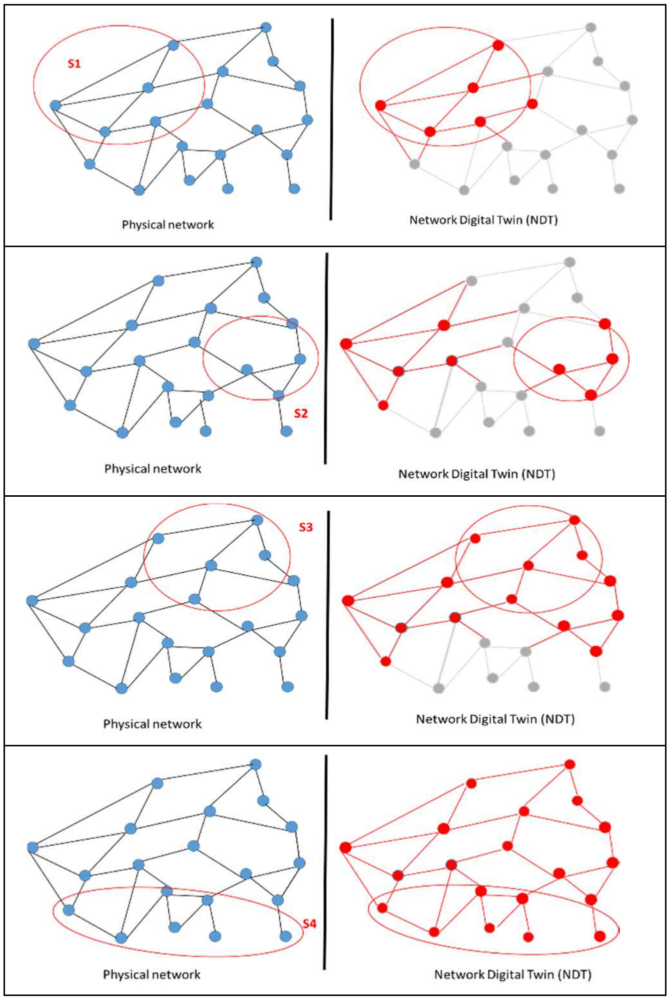

Fig. 6. Incremental NDT generation – Illustrative figure

In order to keep up to date all the already virtualized samples on the NDT's side, the virtualized counterpart of each sample is regularly updated (e.g., on demand, upon the virtualization of a new network sample, or periodically) using predictive models, based, for instance, on Machine Learning (ML) algorithms. The detailed description of these prediction models will be addressed in a future work. Figure 7 shows a 6-step procedure for NDT generation of a full physical network comprising four samples.

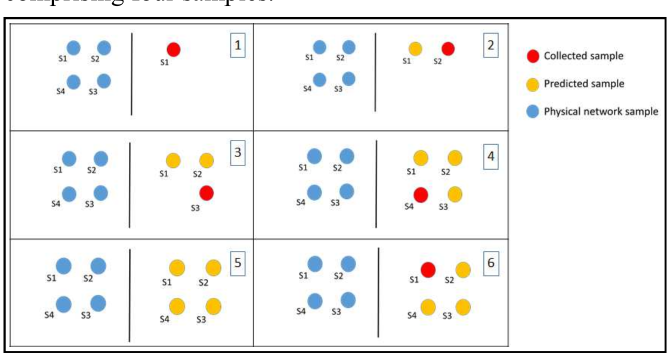

Fig. 7. Abstract view of incremental NDT generation using a combined network sampling and sample prediction scheme.

V. Performance Evaluation

Recall that the primary objective of our solution is to help mitigate the overhead associated with keeping an up-to-date NDT representation despite the dynamism of the physical communication network. In this section, we present preliminary evaluation results of the information overhead when using network sampling operations to generate the sample's NDT. The evaluation of the information overhead metric is crucial, given its strong connection to the scalability objectives our solution aims to achieve. Specifically, this metric offers an estimate (an order of magnitude) of the expected gains compared to a naive approach that involves collecting all network information.

A. Traffic Model

The simulation phase considers a traffic pattern following a Poisson distribution. For each time step, the number of traffic events, both on the node and on the link, is generated using the Poisson distribution. This way, each traffic event contributes to the CPU and memory usage of the nodes and the bandwidth usage of the links. The objective is to express a traffic model that determines the amount of traffic data processed by each node and the amount of data present in the communication link (network bandwidth usage). Higher traffic amount (higher λ values in the Poisson traffic model) can lead to more frequent updates in the network state (e.g., more frequent data processing and transmissions). A given value of λ means that, on average, there is λ traffic event per second, per node or link. Our simulation also considers three values of λ: 0.1, 1 and 10.

In addition, in our simulation, we only consider traffic event-driven data acquisition to generate the NDT. This enables to evaluate the "worst case" for network information overhead (i.e., aggressive mode). Therefore, on each traffic event, an NDT generation phase (full or partial) will take place.

B. Network Model

The performance evaluation considers a wireless communication network, comprising 100 nodes and a probabilistic creation of the link between nodes using a normal distribution with a link type-dependent mean and a standard deviation *s* of 0.05. This allows the link existence probability to vary probabilistically based on a normal distribution. The choice of parameters was made to ensure a reasonable and realistic representation of the network topology. The mean probability of creating a link between any two nodes is 0.1. This indicates that, on average, each possible link between nodes has a 10% chance of existing. This choice balances the network between being overly sparse and overly dense. A standard deviation of 0.05 introduces variability in the link existence probability while keeping the values within a reasonable range. This variation helps in simulating real-world scenarios where network congestion, physical distance, or random failures.

C. Information overhead in random network sampling

In this section, the evaluation of the information overhead for network sampling will consider a sampling function that randomly selects 20 nodes from the network.

In addition, let S_N and S_L be the node-specific information size and link-specific information size, respectively. For a 16-Byte device ID size, a 32-Byte link ID size, and a 10-byte resource field size, we get S_N = 46 bytes and S_L = 62 bytes. Also, on a *zoom-in* operation, it is assumed that one object ID and three associated resources are captured per object of interest (device or link).

Also, let N_S be the number of sampled nodes and L_S be the number of links between the sampled nodes.

As a result, the information overhead I_t for the selective sampling is provided hereafter:

$$I_t = (N_S \cdot S_N + L_S \cdot S_L)$$

D. Information overhead in full network representation

Let N be the number of nodes in the network and L be the number of links in the network Then, the information overhead I_t for the full network representation can be expressed as:

$$I_t = (N \cdot S_N + L \cdot S_L)$$

E. Simulation results

Figures 8 and 9 show the network information overhead related to the NDT's information acquisition, respectively in case of a naïve (full) information collection approach and a selective sampling approach. The results show that more frequent the traffic events, the higher the network information overhead. This can be justified by the fact that in our simulation, we correlate data acquisition to traffic events (λ) both in random sampling and in full network representation cases. In addition, theses preliminary results show the information overhead for data acquisition in case of full network representation is more than 16 times higher than that of selective sampling.

It is worth mentioning, though, that when comparing selective (e.g., random) sampling to a full network representation, it is important to recognize the trade-offs between reducing overhead and potential information loss. Selective sampling can lower information collection and processing demands but may result in missing critical metrics or dynamic changes. However, a well-designed selective sampling strategy, tailored to specific monitoring objectives and informed by traffic patterns, can minimize these risks. Strategies like adaptive sampling or feedback mechanisms can help ensure that essential network segments are represented, reducing the chances of overlooking critical information while still optimizing resource use. Ultimately, full NDT generation using incremental sampling is, by essence, designed to mitigate this information loss problem, which can be expressed in terms of sample prediction accuracy.

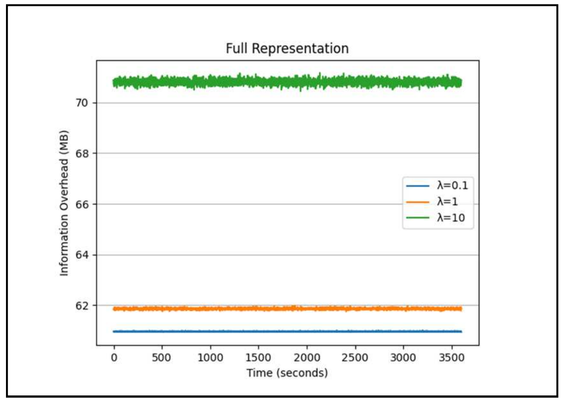

Fig. 8. Information overhead – Case of full NDT generation

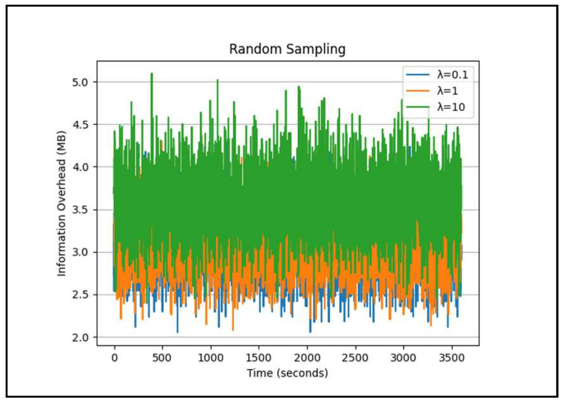

Fig. 9. Information overhead – Case of random sampling (20 nodes per sample)

VI. Conclusion and Future directions

This paper addressed the scalability problem in implementing a Network Digital Twin (NDT) for complex communication networks, characterized by highly dynamic network information, including various frequently changing parameters from both network devices and links. To maintain an up-to-date representation of the real communication network, this dynamic information needs to be continuously captured by the NDT. Consequently, a novel paradigm for network information representation and acquisition needs to be designed. To this end, we propose a network sampling scheme combined with a hierarchical representation of the network information to enable flexible, adaptive, and modular generation of the NDT. Our solution allows for a sliced representation of the communication network, both horizontally and vertically. This approach facilitates efficient network auditing by enabling either a large-scale, high-level view of the communication network segment (horizontal slice) or a small-scale, detailed view of the different communication layers within the network segment (vertical slice).

In the next step, we plan to evaluate the performance of our solution based on more realistic scenarios in the 5G and future 6G contexts, considering a number of critical features such as user mobility, various traffic offloading patterns, network slicing, massive machine-Type communications (mMTC), and massive MIMO (Multiple Input, Multiple Output) technologies. Additionally, the following issues are worth exploring in future work:

ML-based sample selection and prediction: the proposed sampling paradigm can be expanded to enable both sample selection based on relevant network events like node or link failures, traffic congestion, etc. To this aim, Machine Learning (ML) techniques for event prediction and sampling selection can be explored, with a particular focus on those that can handle temporal dependencies, adapt to real-time data, characterize potential inter-resource correlation, and learn from evolving patterns in the network (e.g., Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs), and Autoencoders).

Inter-sample correlation: considering inter-sample correlation (e.g., correlation between two adjacent samples) is particularly important for predicting network samples on the NDT's side. Federated Learning (FL) techniques can help aggregate learning from different samples of the real network, leading to a more accurate representation of each sample's NDT and, by extension, a more precise overall NDT representation. However, when dealing with inter-sample correlation in a distributed setting like FL, ensuring data synchronization across samples is critical to maintaining the effectiveness of the learned models.

Accuracy-overhead tradeoff: one main challenge in implementing a scalable approach for NDT support in complex communication networks is to find the best tradeoff between the accuracy of the NDT representation and the intrinsic overhead of the NDT solution (i.e., bandwidth, memory, and CPU usage). One possible approach to address this problem is to split it into two sub-problems: *instantaneous accuracy* and *average accuracy*. Instantaneous accuracy could be more relevant whenever immediate decisions are made based on an NDT snapshot (e.g., evidence of an imminent event in the real network), while average accuracy is key for long-term analysis or long-term forecast of the real communication network.

Optimized incremental sampling: another key focus of our future research is optimizing the incremental sampling approach, aiming to generate the NDT of the full real communication network in fewer iterations. A critical aspect of this optimization is addressing the issue of sample overlapping, which directly affects both the accuracy of the NDT and the number of iterations required for convergence. While overlapping samples can lead to redundancy and may increase the number of iterations, they also provide additional data points that improve accuracy and completeness of the network representation. This trade-off between speed and accuracy is central to improving the efficiency of our solution in generating the NDT of the full real communication network.

Interoperability considerations for NDT deployment: ensuring interoperability between the NDT framework and various network management protocols presents significant technical challenges, particularly in integrating different data formats and semantic definitions. Additionally, variations in event-handling mechanisms can complicate real-time data exchange, resulting in latency issues and inconsistencies in reflecting network changes in the NDT. To overcome these challenges, a proactive design strategy focusing on standardized, flexible data models and event-handling frameworks can ensure seamless interoperability across diverse networks without sacrificing performance.

References

[1] M. Aboubakar, M. Kellil, P. Roux, "A review of IoT network management: Current status and perspectives", Elsevier JKSU Journal, Volume 34, Issue 7, July 2022, Pages 4163-4176.

[2] E. Coronado et al., "Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks," in IEEE Communications Surveys & Tutorials, vol. 24, no. 4, pp. 2535-2578.

[3] E. Ak, B. Canberk et al., "What-if Analysis Framework for Digital Twins in 6G Wireless Network Management", Proceedings of IEEE IWCMC conference, 2024, pp. 232-237.

[4] C. Zhou et al., "Digital Twin Network: Concepts and Reference Architecture", Internet Draft, July 2024, Work In Progress.

[5] P. Almasan et al., "Network Digital Twin: Context, Enabling Technologies, and Opportunities," IEEE Communication Magazine, 60(11), 2022, pp. 22-27.

[6] L. U. Khan et al., "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," IEEE Commun. Mag., pp. 74-80, 2022.

[7] X. Lin et al., "6G Digital Twin Networks: From Theory to Practice", CoRR abs/2212.02032 (2022).

[8] M. Yaqoob et al., "Digital-Twin-Driven End-to-End Network Slicing Toward 6G," IEEE Internet Computing, vol. 28, no. 2, 2024, pp. 47-55.

[9] H. Ahmadi et al., "Networked Twins and Twins of Networks: An Overview on the Relationship Between Digital Twins and 6G," in IEEE Communications Standards Magazine, vol. 5, no. 4, pp. 154-160.

[10] S. Dittmann et al., "Towards a scalable implementation of Digital Twins - A generic method to acquire shopfloor data", Procedia CIRP, vol. 96, 2021, pp. 157-162.

[11] J. Gong et al., "Demo: Scalable Digital Twin System for Mobile Networks with Generative AI", Proceedings of the 21st ACM MobiSys Conference, 2023, pp. 610–611.

[12] M. Sanz Rodrigo et al., "Digital Twins for 5G Networks: A Modeling and Deployment Methodology" in IEEE Access, 2023, vol. 11, pp. 38112-38126, 2023.

[13] "Lightweight Machine to Machine Technical Specification: Core", Version: 1.1.1 - 2019 06 17, Open Mobile Alliance, OMA-TS-LightweightM2M_Core-V1_1_1-20190617-A.

[14] FIWARE Data Models, https://fiware-datamodels.readthedocs.io/en/stable/

[15] R. Enns et al., "Network Configuration Protocol (NETCONF)", RFC 6242, June 2021.

[16] A. Bierman et al., "RESTCONF Protocol", RFC 8040, January 2017.

[17] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.

[18] S. B. Hadj Said, M. -T. Thi, M. Kellil and A. Olivereau, "On the Management of TSN Networks in 6G: A Network Digital Twin Approach," 2023 IEEE 28th ETFA Conference, 2023, pp. 1-7.

[19] K. McCloghrie et al., "Structure of Management Information Version 2 (SMIv2)", RFC 2578, April 1999.

[20] M. Bjorklund, "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.