SCTE Technical Journal

Volume 5, Issue 1 | August 2025

SCTE STANDARDS JOURNAL

Preventing Service Disruptions with a Network Digital Twin

Cutover Validation in Large Optical Transport Networks

A Technical Paper prepared for SCTE by

Nehuen Gonzalez-Montoro, Professor, Universidad Nacional de Córdoba
Av. Velez Sarsfield 1611
Córdoba, Argentina
nehuen.gonzalez@unc.edu.ar
+54 351 5353800 int. 64

Renato Cherini, Professor, Universidad Nacional de Córdoba
Av. Velez Sarsfield 1611
Córdoba, Argentina
renato.cherini@unc.edu.ar
+54 351 5353800 int. 64

Luis Compagnucci, Head of Network Solutions, Ceragon Networks
3801 E Plano Parkway
Plano, Texas 75074, USA
luisc@ceragon.com
+1 407 496 0315

Jorge M. Finochietto, Professor, Universidad Nacional de Córdoba
Av. Velez Sarsfield 1611
Córdoba, Argentina
jorge.finochietto@unc.edu.ar
+54 351 5353800 int. 64

Table of Contents

List of Figures

1. Introduction

Large-scale network operation has become a more demanding task over the recent years. The increasing reach and heterogeneity of these networks, commonly involving the integration of multi-vendor systems, and the wide variety of services they support, boost the complexity of planning, operations, and maintenance tasks [1]. Over the past decade, substantial efforts have been made to move away from traditional approaches that rely on operator expertise, specific models, and vendor-specific tools [2], toward advanced, flexible, standardized, and closed-loop management and control strategies [3], and toward software-defined networks (SDN) [4], recently framed within the network digital twin (NDT) paradigm [5].

An NDT dynamically maintains a digital replica of the physical network, enabling real-time monitoring and decision-making. As a result, operators can leverage digital twins to address the challenges of network design, operation, and maintenance. This approach supports more comprehensive, integrated, automatable, and resilient practices across all stages of the network lifecycle. It also facilitates optimization, what-if analysis, troubleshooting, and impact assessment. Companies such as Ceragon [6], Nokia [7], Viavi [8], Huawei [9], and Forward Networks [10] offer commercial products aligned with the NDT paradigm.

In our previous works [11], [12], we presented a comprehensive overview of building operator-centric digital twins for optical transport networks (DTON), and analyzed in detail a soft failure detection case study. In this article, we further investigate how a DTON can address the challenges posed by cutover validation. We explore how a NDT can be leveraged to validate and approve planned network cutovers, ensuring that maintenance actions neither affect paths within the same protection group nor impact the same circuit multiple times within a maintenance window. In particular, a digital replica enables the integration of multiple network layers, from fiber ducts to service paths, which can be used to determine potential impacts.

This problem has previously been studied as the task of finding a schedule that maximizes total flow over time [13], and as the challenge of minimizing outages while scheduling all maintenance tasks [14]. In [15], the authors studied the task scheduling problem in optical networks focusing on finding algorithms to solve it. In this work, we present a real-world case study where this capability was developed and deployed across a large-scale optical network, using an operator-centric digital twin model enriched with geospatial, relational, and historical data. We apply integer linear programming (ILP) to solve the maintenance task scheduling problem and discuss the associated implementation challenges. Although our approach relied solely on ILP formulations, which performed very well at the scale of the deployed network, we plan to explore algorithmic, heuristic, and even artificial intelligence–based alternatives in future work.

We adopt the definition of a network digital twin consistent with ITU-T recommendations [16] [17] and prevailing industry literature: a persistent, digital replica of a network that is continuously synchronized with its physical counterpart, capable of simulating, analyzing, and predicting network behavior under various scenarios. While an NDT can support all three of these capabilities, the presented use case focuses primarily on the analysis. More specifically, evaluating the impact of planned maintenance cutovers on network services and protection schemes. Simulation and prediction remain part of the broader NDT framework and could be integrated into this workflow in future developments.

By integrating these kinds of scheduling capabilities, a NDT empowers operators to evaluate planned changes, proactively flag potential risks, and make informed decisions. This results in enhanced network stability and a significant reduction in service outages caused by misaligned maintenance.

The remainder of this paper is structured as follows. Section 2 introduces the principal elements required to model DTON and its implementation. Section 3 analyzes a real-world application focused on planned cutover validation and approval. Section 4 discusses future perspectives, and section 5 summarizes the key findings of our work.

2. Digital Twin for Optical Transport Network

One of the main challenges in designing a NDT is developing a data model that accurately represents the network with sufficient detail while abstracting irrelevant particulars. Such a model should provide enough detail to adequately document the network yet offer sufficient abstraction to facilitate effective problem solving. Moreover, the data model significantly influences the types of problems that can be addressed and solved using the NDT. An overly abstract model may result in poor network representation and solutions that fail to meet operators' actual needs. Conversely, excessive detail can limit the scope of solvable problems. For example, highly detailed models can render the acquisition and maintenance of necessary information impractical, while computational costs may also become prohibitive in some scenarios. In general terms, the data model can be conceptualized as a large temporal multigraph. Addressing most of the common problems requires sophisticated algorithms capable of leveraging multiple, often recursive relationships and extensive historical data. Consequently, adopting computationally efficient data representations is essential to ensure these problems remain tractable.

A second issue arises from the fact that the assumption of complete data availability is rarely satisfied in the practice. Incompleteness can arise from various factors, such as the lack of integration with equipment from specific vendors, the absence of critical information that must be gathered manually, or the inherent complexity of modeling network segments not fully owned or managed by the operator. As a result, some parts of the network are richly documented, while others remain sparsely described. Even though this situation could be understood as a real-world limitation that could make the implementation of NDT at optical networks unfeasible, designing a flexible and adaptable data model can help to overcome this limitation. Moreover, the presence of incomplete data does not reduce the value of the digital twin. On the contrary, it underscores the need for a flexible and adaptive data model, one that can accommodate varying levels of detail and still extract meaningful insights from the available information. This adaptability is key to ensuring the digital twin remains a useful tool for network simulation, analysis, and management, even in the face of persistent data gaps. Using such a data model can cope with ideal scenarios where the full network data is available or collectable, but also with real-world scenarios where some data might be missing or not collectable at some point. For example, in a multi-layer representation of a network, a rigid data model must rely on the underlying layers to compute interdependence of two network links at a top layer. Therefore a data gap in a bottom layer could greatly limit the capabilities of the NDT. Instead, a flexible and adaptable data model can accept abstract relationships between top-level entities that temporarily replace dependence computation until the bottom layers data is retrieved, enabling the application of NDT in this kind of situation.

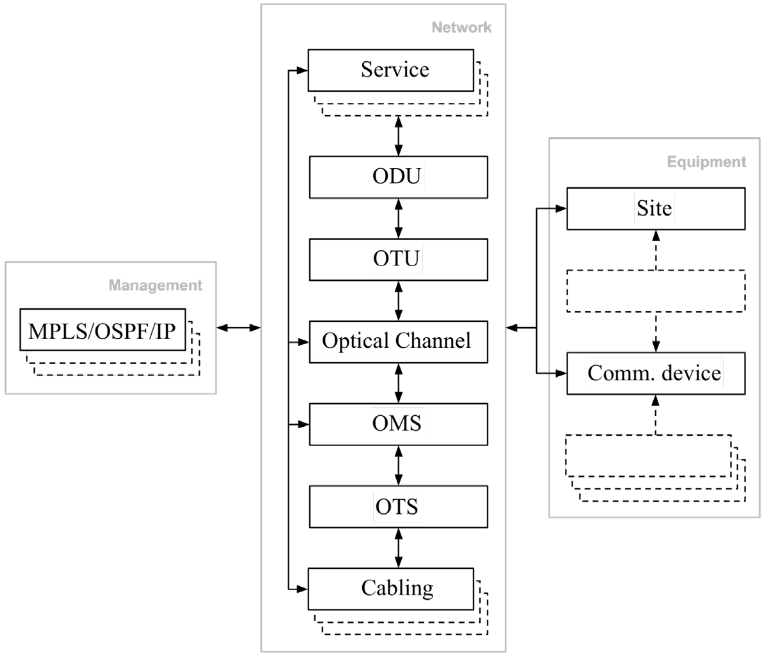

Figure 1 - Schematic Structure of the DTON Data Model

Regarding the first challenge, determining the optimal level of abstraction in optical networks is particularly challenging due to the numerous physical, logical, and management entities involved. A common starting point is the ITU recommendations for Optical Transport Networks (OTN) [18], [19], which offer a hierarchical layer model ranging from optical sections at the bottom optical domain layer to fully digital domain entities such as Optical Data Units (ODUs). Each layer acts as a transport layer for the one above it, with relationships typically involving parallel and/or series multiplexing schemes. The use of this standard presents an advantage because operators' networks are primarily structured around entities that can be mapped to it. Nevertheless, additional requirements remain unaddressed, necessitating extensions to the model to include elements like cabling, communication devices, and their physical layouts, as well as services delivered to end users and their corresponding management. This also includes interactions with management system components from multiple vendors, end-to-end circuit configurations, protection schemes, and related aspects.

Figure 1 presents a simplified schematic of the data model employed. Due to confidentiality constraints, further details cannot be disclosed. The diagram centers on the OTN model layers and the aggregation relationships among them, primarily capturing multiplexing structures. Below these layers are additional ones illustrating the geolocated physical implementation of Optical Transmission Section (OTS) entities, such as fiber optic cables and ducts, along with co-location and geographic proximity. Above, we incorporate layers mapping customers' end-to-end circuits, and redundancy relationships between them. Given that operators often lack certain information, additional relationships bypassing intermediate layers have been implemented. On the figure's left side, administrative and organizational management entities are shown, typically including Muliprotocol Label Switching (MPLS) domains, Open Shortest Path First (OSPF) areas, and subnet divisions, which define routing scopes, fault isolation boundaries, and policy enforcement zones.

Every entity representing a communication link, at any layer, is delimited by sites. This structure aligns perfectly with a graph representation, facilitating the integration of connectivity and routing algorithms. However, communication links depend on the availability of physical communication equipment. Therefore, the data model includes entities for actual devices, cards, ports, and so forth. Additionally, we also model their organization within sites and the occupancy relationships with communication links. This illustrates the coexistence of multiple redundant relationships—ranging from abstract to concrete—for different purposes.

As we will see in the following section, modeling the interrelationships between network entities across different layers is essential to accurately assess the impact of planned maintenance activities, which often affect entities that are not only indirectly, but also distantly related through long chains of intermediate dependencies.

While not directly tied to the approach proposed in this work, the entities comprising the NDT exhibit additional dimensions—beyond their structural organization—that merit consideration. The temporal (historical) dimension is especially relevant for operational and performance data of hardware components. Capturing and storing time-series metrics enables both near-real-time monitoring of network state and the addressing of problems relying on the temporal evolution of specific parameters. For instance, tracking historical optical performance indicators, such as attenuation, is instrumental in detecting soft failures along optical spans. Another dimension concerns representing the current network state and its temporal variations. Maintaining records of active, decommissioned, and planned entities facilitates analysis of hypothetical scenarios and testing of operational strategies. Incorporating these dimensions into the data model enables the forward-looking perspectives outlined in Section 4.

3. Case Study: Cutover Validation

Cutover validation refers to the problem of verifying that a planned network change—such as the activation of a new optical path, the migration of services, or a topology reconfiguration—has been executed successfully and without unintended impact. The challenge arises from the complexity of modern networks and the potential for indirect dependencies between entities. Effective cutover validation is critical to ensure service continuity, minimize downtime, and maintain the integrity of the network after a transition. In this context, the design of a working plan becomes critical to ensure that maintenance activities neither disrupt paths within the same protection group nor result in multiple impacts on the same circuit during a single maintenance window.

This key point can be illustrated better with an example. Consider the simple network shown in Figure 2, which consists of five ROADM nodes (*s1*, *s2*, *s3*, *s4*, and *s5*) connected by fiber optic spans made up of multiple Optical Transmission Section (OTS) segments at the bottom layer. Above these Optical Multiplex Sections (OMS) segments lies the lightpath topology, comprising three optical channels connecting *s1* to *s4*, *s1* to *s5*, and *s4* to *s5*. The network supports only two services (*service 1* and *service 2*), each with one working path and one protection path. These electrical paths are carried over Optical Channel Transport Units (OTUs) shown in the third layer (for simplicity, the ODU layer is omitted). *Service 1*’s working path uses *OTU 1* exclusively, while its protection path is formed by *OTU 2* and *OTU 3*. *Service 2* follows a similar configuration.

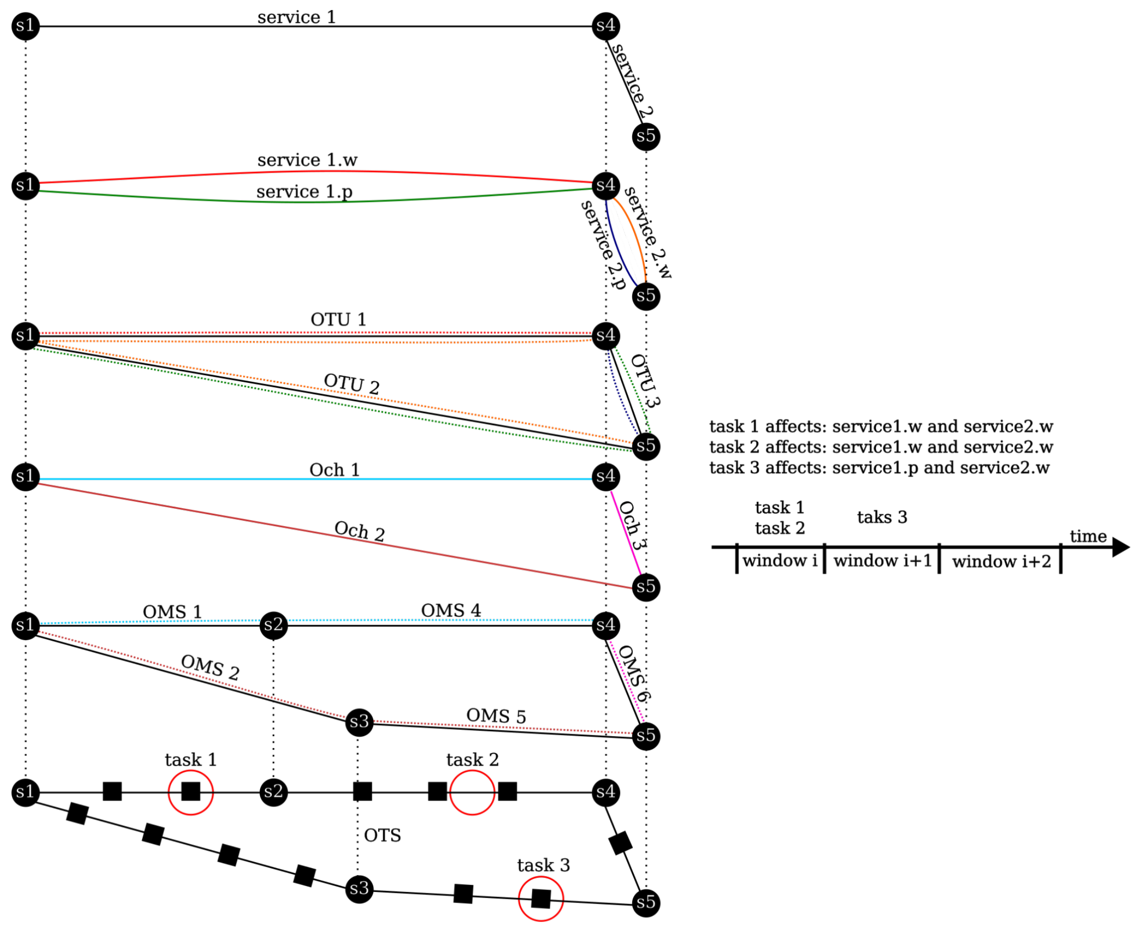

Figure 2 - Simple Task Scheduling Example

Up to this point, we have only shown a multilayer description of a simple network. So, what happens when maintenance tasks need to be performed? In this example, we assume there are two optical amplifiers and one optical section requiring maintenance: *task 1* targets an amplifier located at *OMS 1*, *task 2* targets a fiber section at *OMS 4*, and *task 3* targets an amplifier at *OMS 5*. If all three tasks are executed simultaneously, or even partially overlap, *service 1* will experience downtime. This raises the question: is it possible to schedule these three tasks in a way that avoids any service disruption?

In order to find such a schedule, maintenance windows are defined, and each task must be assigned to one of them. In this case, a valid solution is straightforward: *task 1* and *task 2* can be scheduled in the same window, as they affect exactly the same working and protection paths. *Task 3*, however, must be placed in a separate window to prevent service disruption. A possible solution, as shown on the right side of the figure, is to schedule *task 1* and *task 2* in the first available window, and *task 3* in the next.

It is worth noting that, although this example considers protection at the service level, the same approach applies when protection is implemented at the optical layer (i.e., working and protection lightpaths). The specific query used to identify dependencies in the NDT data model may differ, but the core problem remains unchanged: finding a schedule that avoids simultaneous impact on both the working and protection path or lightpath of a given service or OTU. While this example focuses on maintenance tasks involving amplifiers or fibers, the problem formulation accommodates any type of maintenance activity. This flexibility is only achievable with a well-designed and adaptable data model, as presented in the previous section.

3.1. Integer Linear Programming

Integer Linear Programming (ILP) serves both as a formalism for describing optimization problems and as a method for solving them. It involves optimizing a linear objective function subject to a set of linear equality and inequality constraints, where all variables are restricted to integer values. These constraints define a feasible solution space, and the goal is to find the solution that maximizes or minimizes the objective function.

ILP is a powerful tool for tackling optimization problems. However, since solving ILP instances is itself NP-hard, scalability can become an issue. Over the years, a wide range of algorithms and heuristics have been developed to address this, and there are several open-source and commercial solvers available. Some commonly used ones include SCIP [20], HiGHS [21], CBC [22], IBM ILOG CPLEX [23], and GUROBI [24]. In addition, many libraries simplify the definition and instantiation of ILP models in various programming languages. Thanks to the optimizations implemented in these solvers, ILP models remain applicable in many scenarios. Performance generally depends on the number of variables involved, as well as the size of the feasible solution space.

In the following subsections, we present two ILP models for solving the task scheduling problem. These models not only provide a formal definition of the problem but also offer a practical way to compute solutions. We begin by defining the variables and coefficients, followed by the two model formulations.

3.2. Problem Definition

Leveraging the NDT and its efficiently implemented data model, the impact of planned maintenance tasks on paths can be determined through straightforward queries. For example, it is possible to directly identify all working and protection paths affected when a specific set of underlying resources is scheduled for maintenance.

For instance, if there are N required tasks and M services, each with a defined working and protection path, then we can determine whether the j-th working or protection path will be affected by the i-th task. For modeling purposes, this information can be encoded as a set of coefficients:

$$W_{ij} = \begin{cases} 1 & \text{iff task i affects at least one resource serving the working path of service j} \\ 0 & \text{otherwise} \end{cases}$$

$$P_{ij} = \begin{cases} 1 & \text{iff task i affects at least one resource serving the protection path of service j} \\ 0 & \text{otherwise} \end{cases}$$

$$with \quad \imath \in \{1, 2, ..., N\} \quad and \quad j \in \{1, 2, ..., M\}$$

The task scheduling problem can be addressed in multiple ways, depending on how its dimensions are represented, for example, task duration, availability of technical personnel, or material stock. In our case, we adopted two simple approaches based on slotting time in maintenance windows, under the assumption that both technicians and materials are always available. This representation fits well with our application scenario, as the availability of stock and maintenance personnel can be verified beforehand. Consequently, the scheduling process only needs to account for task interdependencies.

We solved the task scheduling problem with the following two approaches: in one hand scheduling all tasks across multiple windows and, on the other hand, maximizing usage of the next maintenance window. In the first case, the aim is to minimize the total number of windows needed to complete all planned tasks. In the second, the objective is to fit as many tasks as possible into the next available window. When the full set of maintenance tasks is known in advance, the multi-window scheduling approach is generally preferred, as it can yield shorter overall completion times. However, in more dynamic scenarios—where new tasks may arise at each maintenance window—it may be more practical to focus on one window at a time, prioritizing maximum task execution within it.

Given *N* required tasks, *M* services and *K* maintenance windows (with $K \leq N$), we can define the following variables:

$$x_{ik} = \begin{cases} 1 & \text{iff task i is scheduled for the window k} \\ 0 & \text{otherwise} \end{cases}$$

$$x_k = \begin{cases} 1 & \text{iff window k has at least one task scheduled} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{jk} = \begin{cases} 1 & \text{iff the working path of service j is affected in window k} \\ 0 & \text{otherwise} \end{cases}$$

$$p_{jk} = \begin{cases} 1 & \text{iff the protection path of service j is affected in window k} \\ 0 & \text{otherwise} \end{cases}$$

3.2.1. Task Scheduling Problem

The first approach can be formulated as follows: given a set of required maintenance tasks, schedule all tasks while minimizing the number of maintenance windows needed, ensuring that no service experiences disruption. In terms of ILP, it can be stated as:

Minimize:

$$z = \sum_{k=1}^{N} k x_k \qquad (1)$$

Subject to:

$$\sum_{k=1}^{N} x_{ik} = 1 \qquad \forall i \in \{1, 2, ..., N\} \qquad (2)$$

$$x_k \geq x_{ik} \qquad \forall k \in \{1, 2, ..., N\} \quad \forall i \in \{1, 2, ..., N\} \qquad (3)$$

$$W_{ij}x_{ik} \leq w_{jk} \qquad \forall i \in \{1,2,..,N\}, \forall j \in \{1,2,...,M\}, \forall k \in \{1,2,...,N\} \qquad (4)$$

$$P_{ij}x_{ik} \leq p_{jk} \qquad \forall i \in \{1,2,...,N\}, \forall j \in \{1,2,...,M\}, \forall k \in \{1,2,...,N\} \qquad (5)$$

$$w_{jk} + p_{jk} \leq 1 \qquad \forall k \in \{1,2,...,N\}, \forall j \in \{1,2,...,M\} \qquad (6)$$

Equation 1 defines the objective: minimizing the number of maintenance windows required. Since x_k indicates whether window *k* is used, the goal is to minimize the sum of all these variables. The sum is weighted by the position *k* to prioritize earlier windows over later ones. Equation 2 ensures that each task is assigned to exactly one maintenance window. Equation 3 sets a window's status to "used" whenever at least one task is scheduled within it. Equations 4 and 5 indicate whether the working or protection path of service *j* is affected by any task scheduled in maintenance window *k*. Finally, Equation 6 prevents both the working and protection paths of a service from being affected simultaneously in the same maintenance window.

3.2.2. Next Window Maximum Usage Problem

The second approach can be described as follows: given a set of required maintenance tasks, schedule the maximum possible number within the next maintenance window, ensuring no service disruption. Since only one window is considered, *K* equals 1, and the following ILP formulation is sufficient:

Maximize:

$$\sum_{i=1}^{N} C_i x_{i1} \qquad (7)$$

Subject to:

$$W_{ij}x_{i1} \leq w_{j1} \qquad \forall i \in \{1,2,...,N\}, \forall j \in \{1,2,...,M\} \qquad (8)$$

$$P_{ij}x_{i1} \leq p_{j1} \qquad \forall i \in \{1,2,...,N\}, \forall j \in \{1,2,...,M\} \qquad (9)$$

$$w_{j1} + p_{j1} \leq 1 \qquad \forall j \in \{1,2,...,M\} \qquad (10)$$

The objective now is to maximize the number of tasks scheduled within the next maintenance window. The coefficients C_i in Equation 7 represent task priorities; if all coefficients are equal, all tasks are treated with the same priority. Equations 8 and 9 indicate whether the working and/or protection paths of service *j* are affected by any selected task. Finally, Equation 10 restricts the solution to prevent simultaneous impact on both the working and protection paths of a service.

It is worth noting that the window duration can be adjusted according to the operator's needs. Although in our application scenario a fixed window duration is sufficient, the models can be easily extended to allow multiple slot allocations per task, providing a more flexible and efficient representation. However, we discarded this option because task duration is highly unpredictable; consequently, maintenance planners within the operators prefer to allocate an entire maintenance window to each task.

The first model ensures that all tasks are scheduled, but in the second case a task can remain unscheduled for multiple windows and potentially for ever. This is an unwanted behaviour and we cope with it by computing the C_i priority coefficients after each assignment in order to prioritize tasks that have been unscheduled for more consecutive windows.

3.3. Application Scenario and Results

Although the previous example might suggest that this problem is straightforward, the scale of a real-world, nation-wide network makes finding solutions by intuition or simple tools very challenging. We applied the described models to a network with 1,100 optical nodes, 849 optical sections, and over 2,500 protected services. Prior to implementing the DTON equipped with the ILP models, the complexity had grown so much that maintenance management limited itself to scheduling only one task per maintenance window, which caused significant delays and prioritization problems. Although in some cases the scale of the problem can be reduced by solving it for specific sections of the network, obtaining an optimal solution without proper modeling becomes highly challenging. As a result, operators often prefer conservative, suboptimal solutions in order to minimize the risk of unexpected behavior.

Our initial approach relied on the task scheduling model, which produced very good results. However, as new tasks continued to emerge before previously scheduled ones were completed, the solution quickly became outdated. Currently, both models are available, and their use depends on the task flow. For instance, the second model is preferred during periods of high task volume, whereas the first is applied when the task flow is lower and a global optimal solution can be achieved.

Integrating these models into the DTON has provided maintenance managers with a reliable tool, enabling better, data-driven decisions that reduce risks, minimize service disruptions, and shorten the time required to complete all maintenance tasks.

4. Future Perspective

Following the definitions provided in the ITU-T recommendations [16][17] we cover the representation level and partially the optimization level of NDTs. Looking forward to covering all five levels, the use case presented in this article paves the way toward a fully automated maintenance planning ecosystem that encompasses failure and anomaly detection, required actions definition, task scheduling, and post-execution validation.

The data collection capabilities enabled by the DTON allow for the continuous monitoring of network elements, supporting the early detection and prediction of soft failures, degradations, and anomalies in both optical equipment and fiber infrastructure. In previous work [8, 9], we demonstrated a specific use case that validates this potential. Nonetheless, there remain numerous open challenges, both in research and in implementation, particularly around the integration of more sophisticated artificial intelligence techniques for reliable and scalable forecasting.

Transitioning from anomaly and failure prediction to automated task planning becomes feasible when the DTON also integrates contextual information about available resources, such as personnel, spares, and operational constraints. Once the required actions are defined, scheduling models, such as the ones presented in this work, can be used to optimize execution while minimizing service impact. The cycle is completed with a post-maintenance validation process, where the digital twin can be queried to assess whether the intervention effectively addressed the issue and improved network health.

Artificial intelligence can play a central role throughout this process. In terms of failure prediction, anomaly detection and degradation prediction can be achieved over fiber spans attenuation time series. A comprehensive survey of deep learning applied to time series anomaly detection can be found in [25]. Also, machine learning can enhance task planning and scheduling strategies. For example, ILP formulations provide a valuable ground truth that can be used to generate real or synthetic datasets representing scheduling scenarios in complex networks. These datasets could be leveraged to train machine learning models capable of approximating optimal scheduling decisions with significantly reduced computational cost. In large-scale networks where solving ILP models becomes intractable in real time, such models could offer high-quality heuristic alternatives.

Moreover, reinforcement learning could be explored for adaptive scheduling in dynamic environments, where tasks and network states evolve over time. In this case, one or more heuristic can be learned in order to schedule tasks in a per window fashion. The use of reinforcement learning in generic task scheduling problems has been widely studied. A comprehensive review of related works can be found in [26]. Given the high availability requirements of communication networks, learning-based approaches can and should be complemented with feasibility checks and optimization models in order to guarantee no service affectation. Such integration would enable hybrid systems that balance precision and scalability, ultimately leading to more resilient and autonomous network operations.

In summary, this work lays the baseline for future developments that can couple the rigorous modeling capabilities of digital twins with the predictive and adaptive strengths of artificial intelligence, aiming to build a proactive, efficient, and fully closed-loop maintenance framework.

5. Conclusion

In this work, we presented a practical application of the DTON in the context of maintenance task scheduling and validation. By integrating a flexible multilayer data model with ILP optimization techniques, we demonstrated how a digital twin can be used to systematically plan maintenance windows while avoiding service disruptions.

Although the scheduling problem could be approached using a static representation of network routing, our implementation operates on a continuously updated, multi-layer digital model of the actual network, integrating physical topology, logical paths, protection relationships, and service configurations. This dynamic aspect, together with its ability to ingest live operational data and historical trends, aligns with key characteristics of a NDT as defined in ITU-T and industry practice. The scheduling models are not standalone algorithms but are embedded within this dynamic environment, allowing them to reflect the current state of the network and assess maintenance risks in near-real time. This distinction is critical: it transforms the solution from a one-time network map into an ongoing operational tool that evolves alongside the physical network.

This case study showed that leveraging a digital twin not only enables impact analysis of maintenance activities but also supports optimal scheduling decisions. Decisions that would be nearly impossible to reach by intuition alone due to network complexity. The combined use of the two scheduling models, minimizing the total number of windows versus maximizing task execution within the next window, proved to be an effective and adaptable strategy in real-world, dynamic operational environments.

Moreover, the implementation of these methods significantly improved the efficiency of maintenance planning, reducing risk and delays, and enabling operators to carry out more tasks in less time without compromising service availability.

Looking ahead, this work lays the groundwork for a broader and more integrated maintenance ecosystem. Future developments may include predictive analytics, AI-assisted planning, and fully automated validation workflows, reinforcing the central role of digital twins in modern network operations.

6. Abbreviations

NDT	network digital twin
DTON	digital twin for optical networks
OTN	optical transport networks
ODU	optical data unit
OTS	optical transmission section
OMS	optical multiplex section
OTU	optical channel transport unit
Och	optical channel
ILP	integer linear programming

7. Bibliography and References

1. A. Martinez, M. Yannuzzi, V. López, D. López, W. Ramírez, R. Serral-Gracià, X. Masip-Bruin, M. Maciejewski, and J. Altmann, "Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks", IEEE Commun. Surv. & Tutorials 16, 2207–2230 (2014).

2. H. Kim and N. Feamster, "Improving network management with software defined networking", IEEE Commun. Mag. 51, 114–119 (2013).

3. Andriolli, N., Giorgetti, A., Castoldi, P., Cecchetti, G., Cerutti, I., Sambo, N., ... & Paolucci, F. (2022). Optical networks management and control: A review and recent challenges. *Optical Switching and Networking*, *44*, 100652.

4. Nsafoa-Yeboah, K., Tchao, E. T., Yeboah-Akowuah, B., Kommey, B., Agbemenu, A. S., Keelson, E., & Monirujjaman Khan, M. (2022). Software-defined networks for optical networks using flexible orchestration: Advances, challenges, and opportunities. *Journal of Computer Networks and Communications*, *2022*(1), 5037702.

5. Wang, D., Song, Y., Zhang, Y., Jiang, X., Dong, J., Khan, F. N., ... & Zhang, M. (2024). Digital twin of optical networks: a review of recent advances and future trends. *Journal of Lightwave Technology*, *42*(12), 4233-4259.

6. Ceragon, "Ceragon network digital twin", (2024). Accessed: 2024-12-13.

7. Nokia, "Network digital twin for manufacturing", (2024). Accessed: 2024-12-13.

8. VIAVI Solutions, "Network digital twin", (2024). Accessed: 2024-12-13.

9. C. Janz, Y. You, M. Hemmati, A. Javadtalab, Z. Jiang, H. Li, and F. Haoyu, "Network digital twins for optical networks", Opt. Fiber Technol. 89, 104068 (2025).

10. Forward Networks, "Forward enterprise: Network digital twin solution", (2024). Accessed: 2024-12-13.

11. R. Cherini, N. Gonzalez-Montoro, L. Compagnucci, and J. M. Finochietto, "Bottom-up design of digital twins for optical transport networks: Challenges and lessons learned [invited]", in 2024 International Conference on Optical Network Design and Modeling (ONDM), pp. 1–6, (2024)

12. R. Cherini, N. Gonzalez-Montoro, A. Gonzalez-Montoro, L. Compagnucci, and J. M. Finochietto, "Building operator-centric digital twins for optical transport networks: from data models to real applications", Journal of Optical Communications and Networking, 17(6), B1-B14, (2025).

13. N. Boland, T. Kalinowski, H. Waterer, and L. Zheng, "Scheduling arc maintenance jobs in a network to maximize total flow over time", Discrete Applied Mathematics, 163, 34-52, (2014)

14. F. Abed, L. Chen, Y. Disser, M. Groß, N. Megow, J. Meißner, and R. Rischke, "Scheduling maintenance jobs in networks", Theoretical Computer Science, 754, 107-121, (2019)

15. X. Qin, T. An, X. Gao, Y. Gong, Q. Hu, X. Huo, and B. Zhang, "Intelligent task scheduling for cutover in optical networks", Optics Express, 33(13), 27334-27347, (2025)

16. International Telecommunication Union, "Y.3090: Digital twin network – Requirements and architecture," Tech. Rep. Y.3090, ITU-T (2022).

17. International Telecommunication Union, "Y.3091: Digital twin network – Capability levels and evaluation methods," Tech. Rep. Y.3091, ITU-T (2023).

18. International Telecommunication Union, "G.807: Generic Functional Architecture of the Optical Transport Network," Tech. Rep. G.807, ITU-T (2020).

19. International Telecommunication Union, "G.872: Architecture of the Optical Transport Network," Tech. Rep. G.872, ITU-T (2019).

20. S. Bolusani, M. Besançon, K. Bestuzheva, A. Chmiela, J. Dionísio, T. Donkiewicz, J. van Doornmalen, L. Eifler, M. Ghannam, A. Gleixner, C. Graczyk, K. Halbig, I. Hedtke, A. Hoen, C. Hojny, R. van der Hulst, D. Kamp, T. Koch, K. Kofler, J. Lentz, J. Manns, G. Mexi, E. Mühmer, M. E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, M. Turner, S. Vigerske, D. Weninger, and L. Xu, "The SCIP Optimization Suite 9.0," Available at https://www.scipopt.org/ February 2024

21. High performance software for linear optimization, https://github.com/ERGO-Code/HiGHS

22. COIN-OR Branch and Cut Solver, https://github.com/coin-or/Cbc

23. IBM ILOG CPLEX optimization solver https://www.ibm.com/es-es/products/ilog-cplex-optimization-studio

24. Gurobi optimization software https://www.gurobi.com/

25. Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, *57*(1), 1-42.

26. Shyalika, C., Silva, T., & Karunananda, A. (2020). Reinforcement learning in dynamic task scheduling: A review. *SN Computer Science*, *1*(6), 306.

Click here to navigate back to Table of Contents

