Digital Twins of Computer Networks

Martin Tonauer, Kilian Holzinger*
**Chair of Network Architectures and Services*
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: m.tonauer@tum.de, holzinger@net.in.tum.de

Abstract—**Usage of a digital twin (DT) is already an established technology in various industries like aerospace, construction or traffic analysis. However, in the field of computer networks its utilization is still rare. Although already in usage for example in the roll-out of 5G networks, the true potential of a digital twin network (DTN) has yet to be developed.**

This paper tries to contribute a concise overview of DTN technology not only by giving an insight into the background of this technology and some of the nomenclature but also by summarizing properties and requirements for a functioning DTN. The paper also remarks the problems surrounding the implementation of a working DTN. To help in mitigating these, it offers a simple conceptual implementation model to promote further research into designing DTNs. The paper also presents current and possible future use cases as well as open questions for further attention.

Index Terms—**Digital Twin (DT), Digital Twin Network (DTN)**

1. Introduction

The concept of a digital twin (DT) has been used in different types of industries for the purpose of simulating real world hardware, applications or whole scenarios without interfering with the physical system (PS) itself. Born out of the desire to monitor, test or experiment on these systems in a safe environment, DTs have seen widespread usage in many industries, from automotive production all the way to weather forecasting [1] [2]. Given the broad range of utilization it would seem logical for DTs to have a similar share in computer networks, but the potential and abilities of such a digital twin network (DTN) are only starting to be valued [3]. A DTN in the scope of this paper is the simulated counterpart to a real and physical computer network. The goal of this essay is to give the reader an overview of this technology and outline some applications and their benefits to designing, constructing and maintaining computer networks.

The rest of this paper is structured as follows: Section 2 will give a brief overview of the historical background and motivate the usage of DTs, while Section 3 will define the terminology that is used in this paper. Section 4 introduces properties and requirements of DTs and DTNs. A possible model of a DTN is presented in Section 5 with some application examples following in Section 6. Section 7 discusses the shown concepts and corresponding open questions. A concluding summary of this paper can be found in Section 8.

2. Background and Current Status

Having a first, second or third draft of something is probably as old as human craftsmanship. But what happens when the desired product becomes final and goes into production? With increasing complexity of systems, the need to test or experiment before actually committing new features to a real world product becomes ever more important. Being able to verify that the addition of new machines to an existing production line will not unintentionally alter the functionality or output of the line before the actual installation of the hardware is an important capability of a production plant. Similarly, testing a new software patch on a real satellite while it is in orbit can have drastic repercussions maybe even to the point where it can no longer communicate. Therefore, the roll-out of this patch and the correct subsequent operation of the satellite have to be checked beforehand. Using a stand-in is an intuitive way of achieving this goal.

The approach of having an identical copy of a real, physical system - effectively a twin - has been around for quite some time. Famously being used since the 1960s by NASA to verify new procedures from the ground for spacecraft already in orbit, this technique gained momentum in the following years in many other industries as Grieves and Vickers noted in [1]. Because of advancements in processing power it became unnecessary for that twin to be a physical instance itself. Starting with CAD models, where there is no direct feedback from the product to the now digital twin, evolving all the way to DTs which are constantly fed by data from their real world counterparts [1]. A recent example is an Earth observation DT currently in development. It is supplied with data from both space and ground based sensors to achieve better weather forecasting [2].

The formal beginning of DTs, albeit as a tool for product life-cycle management, happened at a University of Michigan presentation in 2002 by Dr. Grieves. Since then it has become a widely used technique to monitor, test and experiment with real world systems in a controlled and digital environment [1].

Both Wu et al. [3] and Vaezi et al. [4] have noted that despite promising use cases, DTs in computer networking have not gained as much traction - at least for the time being. One such use case is the simulation of new routing strategies via a DT and the subsequent ability to predict the real networking behavior. Those can be applied for example to emulate, validate and optimize 5G network roll-out as Nguyen et al. propose in [5]. Some telecommunication companies already use DTNs for exactly that

doi: 10.2313/NET-2023-11-1_09

purpose [6] or do see future use cases in implementing next generation mobile networks [7].

3. Definitions and Terminology

While there is broad consensus in the literature about the definition of DT, the term DTN has various meanings. The following section will define the two terms for the scope of this paper and describe them in greater detail.

3.1. Digital Twin (DT)

The optimal DT is a perfect representation of a yet to manufacture or already existing product that holds at least as much information about the real product as the real product itself could. With that information it is possible to create a physical copy of the DT and vice versa. This definition also encompasses the state of the DT when its physical counterpart is already in existence and therefore linked to it for the remainder of the life cycle or beyond. This digital representation can be connected to a single physical entity or to a more complex system comprised of multiple objects. The optimal DT should also hold information about all previous and current states of the physical twin to enable the prediction of its future behavior. [1]

Vaezi et al. summarize slightly varying definitions in the literature down to three distinct entities that are needed: the represented physical system, the DT itself and a communication or information link between the two. [4]

Due to this information-connection DTs have been able to evolve from being simple digital copies to complex representations that change and develop with their respective PS. This link is what enables the DT's broad information capability in the first place. [3] [4]

3.2. Digital Twin Network (DTN)

The term is sometimes used to describe a network consisting of at least two general DTs communicating with each other. However, this paper is about the application of DTs in computer networking specifically. Therefore the term will be utilized in accordance with ITU-T recommendation Y.3090: "*A digital twin network is a virtual representation of a physical network*" [8]. In other words, it is explicitly used to describe the virtual counterpart of a yet to realize or already existing physical computer network. Figure 1 clarifies this definition in a simple example, depicting a PS consisting of a server and three clients that communicate with each other. The DTN is the virtual copy of this physical network with every real hardware and communication link having a virtual counterpart. The bidirectional information connection between the PS and DTN enables the correct representation.

4. Properties and Requirements of DTs and DTNs

There need to be some representable properties to be able to qualitatively and quantitatively describe DTs in detail. Vaezi et al. [4], Minerva et al. [9] and [3] give some examples which are compounded in Table 1 for easier reference.

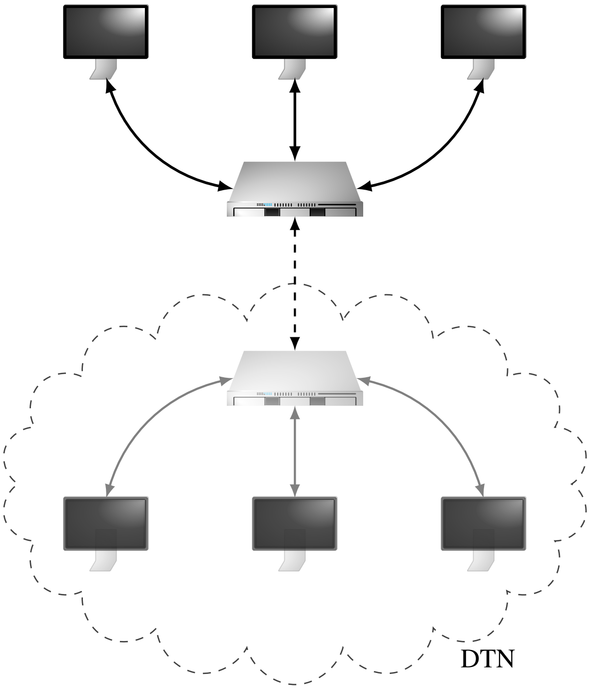

Figure 1: Representation of a physical network and its digital twin network (DTN)

TABLE 1: Properties of DTs based on [3] [4] [9]

Property	Explanation
Promptness	Intuitively, this is the reaction time of the DT to its PS. Or more generally the time it takes for information from the PS to reach the DT and change its state.
Similarity	Reflects the difference of information between the PS and DT in the same state. For example, transmission noise can alter the signal and therefore the information.
Replicability	Refers to the ability to replicate a single PS into multiple DTs at the same time. This should also encompass the possibility to create a PS from an existing DT.
Composability	Describes the ability of multiple smaller DTs to be integrated into fewer but larger ones. This adds the capability to form ever more complex systems.
Scalability	If a PS grows in system size, it is desirable for the DT not to increase its network footprint exponentially. Scale in this context also includes geographical size.
Reliability	A DT is considered to be reliable when its data and operational integrity can be verified. This also includes the persistency and availability of information.
Predictability	Since DTs can be comprised of vast amounts of data, they still need to be predictable in precise environments when simulating their behavior with other entities.
Accountability	As a DT can be a set of smaller DTs, the collected information has to be traceable to a specific DT. This property not only includes origin but also the ownership of data and its usage rights.
Adaptability	Is the capability of a DT to adapt in accordance with its PS dynamically. This can involve new models, different data collection or change in needed network resources.

ITU-T recommendation Y.3090 assigns some of the properties mentioned in Table 1 discrete value requirements. For example, the reliability level of a DTN is specified to be at least 99.99%. A selection of requirements can be found in Table 2. The interested reader is kindly referred to [8] for the complete set.

TABLE 2: Requirements for DTNs based on [8]

Requirement	**Explanation**
Data collection	For a DTN to work, it has to collect vast amounts of data in an efficient manner. The required type of data can vary with use cases but may include: logs, records and status of all network elements; flow statistics like latency, throughput or packet loss; device-specific data such as port information or link status.
Data repository	Storage and retrieval of data is essential to the intended capability of DTNs. Huge amounts of data have to be stored in a way to allow parallel processing and real-time access. For backups and rollbacks it is required to have historical data at hand.
Security	As there is a significant amount of stored information, security is very important. Therefore, the DTN should be able to defend against already happening attacks to the physical network and also attacks against itself.
Privacy	Data protection laws are as applicable to the DTN as they are to the physical network. A DTN must be able to comply with these rules both within its own layer and during communication with the PS.
Compatibility	To fully support future technology, the DTN should be compatible and adhere to established network standards and support various physical interfaces and topologies, different types of databases as well as existing network measurement tools.

Outlined above and in Section 3.1, a DT will and should have comprehensive data from and about its corresponding PS. It has to be noted that in reality the amount of information which a DT can hold is limited. Possible reasons for this are [4]:

- updating of states in regular intervals leads to discrete data points which implies missing information;
- every form of processing by the DT has to take at least some time so delays are inevitable;
- there are limits in terms of resources available to the DT so the capabilities are limited as well.

Nevertheless, a DT is considered to be fully functional when it delivers information with accuracy in an expected and acceptable range while working within these constraints and satisfying the mentioned requirements. It does not have to represent the PS as closely as possible, just as closely as required for the specific use case [4].

5. Modeling a DTN

As of yet, there is no standard model for DTNs so the following example is exactly that: one example. The simple structure in Figure 1 is meant as a quick visual representation. What it lacks to be a real world usable model is a third layer that adds network applications and their requirements for the PS. One such possible description is given in [8] and by Almasan et al. in [10].

This model is comprised of three layers: the physical network is at the bottom, the digital twin network is the middle layer and the top layer is the network applications layer. Figure 2 gives a visual representation for this model.

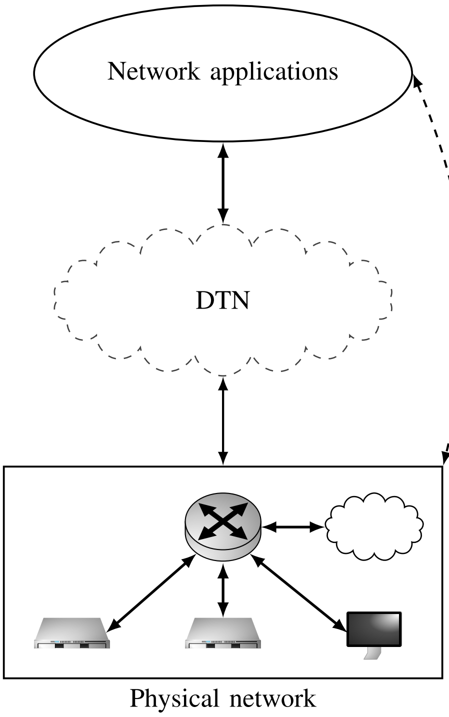

Figure 2: Visualization of the three-layer DTN based on models in [8] and [10]

The leading thought of this model is that instead of deploying network services directly to the PS, they are first fed into the DTN. This way the DTN exposes the capabilities of the physical network to the network applications. In turn, it receives information about the intention of the network application. Based on collected data from the physical network and modeled for example through the use of machine learning on previous traffic, the virtual twin now generates predicted network metrics. These can be used to analyze, verify and optimize the network application services and deploy them to the PS - possibly even without human intervention [8] [10]. This represents the most obvious differentiating factor to a mere simulation since the DTN is able to both control and manage the PS.

Following, the three layers are described in more detail.

5.1. Bottom Layer - Physical Network

This layer is comprised of all physical devices in the network. It is connected to and shares extensive amounts of data with the DTN. The extent of this exchange depends on the specific use case and its requirements. See Sections 3 and 4 for further information. [8]

5.2. Middle layer - DTN

There are three main subsystems that can be identified in a DTN and their interaction attempts to fulfill the requirements given in Table 2. [8]

5.2.1. Unified Data Repository. Responsibilities include the collection and storage of data via the southbound connections to the physical layer but also the process of updating the current state.

5.2.2. Unified Data Models. This subsystem offers, among others, instances of modeled network applications which in turn allow the prediction and data processing capabilities of the DTN.

5.2.3. Digital Twin Entity Management. This allows easy maintenance, logging, control and visualization of the DTN and its data models. It is also responsible for the internal and external security of the DTN.

5.3. Top Layer - Network Applications

The top layer is comprised of the network applications and their services. It is connected to the DTN and relays current or prototype services and requirements to the DTN which then sets up new emulating instances. Upon verification that predefined metrics, such as traffic parameters, coincide with modeled and expected behavior, the DTN proceeds with deploying those services to the physical layer. [8]

6. Practical Applications

There is one empowering capability that the literature is in consensus about: the predictive potential of DTNs [4] [5] [6] [7] [10]. So it seems obvious that this is a centerpiece of many use cases - some of which will be discussed in this section.

- **Planning and construction:** Wang et al. discuss in [11] the possibility of DTNs for use during the complete life-cycle of a network. Because of DTs' property of replicability it is possible to create a new physical system from the DTs' information alone. So it is feasible to begin the implementation of a physical network with the creation of a DTN, continue with running numerous scenarios, optimizing and verifying the network, all before a single physical device is installed. This not only reduces cost but also risk especially with rising system complexity. [10] [11]
- **Maintenance and troubleshooting:** A DTN with its vast amount of meta information about a network has the potential to find the root of an error very quickly. Possible solutions can then be verified inside the DTN and transferred to the physical network. [5] [11]
- **Detection of abnormal behavior:** A functioning DTN mirrors the current operating state of its network which means that deviations from the expected behavior in the physical traffic from the one in the DTN can be telltale signs for anomalies. This can lead to earlier recognition or even prevention of errors. [10]
- **Optimization:** Through the use of various data driven models, a DTN can offer easier network optimization capability because it does not impact the current active physical network. One particular category of models can offer very promising outcomes: machine learning technologies. They can benefit from faster and more efficient operation compared to traditional optimization algorithms because of their awareness of previous optimizing runs. [8] [10] [11]
- **Innovation:** Trial runs in physical networks can have multitudes of negative outcomes due to various reasons for example unavailability of the network during the run or risk of damage to the system. Naturally, network operators are therefore very cautious and conservative when testing innovative technology in real world surroundings. A DTN offers an environment in which new protocols, applications or devices can be scrutinized in a safe manner. It also provides the possibility for in-advance-testing of edge cases like network failures, misconfigurations or security breaches. [8] [10]

7. Discussion and Open Questions

Condensing the findings of this paper, DTNs offer many capabilities and advantages, especially for complex and big networks. Because of the deep interaction with the PS, the DTN is able to manage and control the physical network in real-time.

What remains to be discussed however, is the practical implementation of a DTN that fully satisfies the presented requirements and delivers on the promises. Since tech companies that already use DTNs are generally anything but open about their implementations, this is a big obstacle in experimenting and iterating over different design possibilities of DTNs. Generalizing complex and huge network topologies while still allowing for those predictive capabilities is an open research question. Furthermore, the collection, storage and processing of the desired amount of data is everything but trivial, especially in real time environments. [3] [10]

Future work is necessary to come up with possible implementations that do not rely on broad financial and subsequent technical capabilities which currently only big tech companies seem to be able to provide. Further research into DTN technology for small and medium sized networks can also shed light on whether those would benefit from this technology as well.

8. Conclusion

In this paper the technology of DTs was introduced and their application in computer networking was discussed. These DTNs can offer solutions to questions concerning vast and interconnected networks like mobile telecommunication networks. Properties of and requirements for DTs and DTNs from different sources of literature [3] [4] [8] [9] were identified and presented, as well as the suggestion for a conceptual model of a DTN [8] [10]. Applicable use cases include the complete life cycle management of a network spanning from planning all the way to maintenance, troubleshooting and upgrading [5] [11]. The predictive capabilities of DTNs also allow for the detection of abnormal network behavior and time and cost effective integration of new technologies into the network [10]. One of the most promising areas of interest is the optimization of network traffic and topology. With its rich amount of data about the state of a network, the DTN technology in combination with machine learning algorithms can offer faster and probably better optimization results than current network simulation [10] [11]. Lastly the question of actual implementation was discussed and underlying problems presented like the requirement for storing and processing huge amounts of data in real-time. In case these problems are successfully tackled, DTNs have many possible applications for designing, optimizing and maintaining computer networks.

References

[1] M. Grieves and J. Vickers, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. Springer International Publishing, 2017, pp. 85–113.

[2] S. Chin, “Using the Digital Twin to Improve Weather Forecasting,” https://www.designnews.com/artificial-intelligence/using-digital-twin-improve-weather-forecasting, 2022, [Online; accessed 20-March-2023].

[3] 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.

[4] 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.

[5] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, “Digital Twin for 5G and Beyond,” *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

[6] S. Communications, “Simplifying 5G with a Network Digital Twin,” https://www.spirent.com/assets/wp_simplifying-5g-with-the-network-digital-twin, Tech. Rep., 2022, [Online; accessed 30-March-2023].

[7] P. Öhlén, “The Future of Digital Twins: What will they mean for Mobile Networks?” https://www.ericsson.com/en/blog/2021/7/future-digital-twins-in-mobile-networks, 2021, [Online; accessed 30-March-2023].

[8] “ITU-T Y.3090; Digital Twin Network – Requirements and Architecture,” http://handle.itu.int/11.1002/1000/14852, International Telecommunication Union, Recommendation, Feb. 2022.

[9] R. Minerva, G. M. Lee, and N. Crespi, “Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models,” *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1785–1824, 2020.

[10] P. Almasan, M. F. Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. R. López, A. A. P. Perales, P. Harvey, L. Ciavaglia, L. Wong, V. Ram, S. Xiao, X. Shi, X. Cheng, A. Cabellos-Aparicio, and P. Barlet-Ros, “Digital Twin Network: Opportunities and Challenges,” *CoRR*, vol. abs/2201.01144, 2022. [Online]. Available: https://arxiv.org/abs/2201.01144

[11] D. Wang, J. Guo, Y. Ouyang, S. Wang, A. Yang, Z. Ren, Y. Ding, G. Chen, C. Zhou, and D. Chen, “Leverage Digital Twins Technology for Network Lifecycle Management,” in *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 2022, pp. 1–5.