Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-8 (2022) 37–42

Digital Twin Network for the IIoT using Eclipse Ditto and Hono

Mehdi Kherbache * **Moufida Maimour** * **Eric Rondeau** *

* *CRAN Laboratory, Université de Lorraine, CNRS, UMR 7039, Campus Sciences, BP 70239, F-54000 Nancy, France (mehdi.kherbache@univ-lorraine.fr, moufida.maimour@univ-lorraine.fr, eric.rondeau@univ-lorraine.fr).*

Abstract: Digital twins are beginning to revolutionize many industries in the last decade, offering a plethora of benefits to optimize the performance of industrial systems. They aim to create a continuously synchronized model of the physical system that allows for rapid adaptation to dynamics, primarily to unpredicted and undesired changes. A vast range of industrial domains have already benefited from digital twin technology, such as aerospace, manufacturing, healthcare, city management and shipping. In addition, recent research is beginning to explore the integration of digital twins into computer networks to enable more innovation and intelligent management. One of the building blocks of digital twin technology is the Internet of Things, where wireless sensors and actuators are deployed to provide interaction between the physical and digital worlds. This type of network is complex to manage due to its strong constraints, especially when controlling critical industrial applications, which gave rise to the Industrial Internet of Things (IIoT). We believe that optimizing IIoT will lead to effective integration of digital twins in Industry 4.0. In this paper, we design a Digital Twin Network (DTN) for IIoT where sensors, actuators, and communication infrastructure are replicated in the digital twin to enable real-time intelligent management of these networks. By taking advantage of Eclipse Hono which allows an efficient connectivity for the network devices and Eclipse Ditto for representing the devices states in a digital form and also providing easy access to these states for the DTN. In this way, cognitive network services such as predictive maintenance, sustainability features, network diagnostics, security management, resource allocation, energy optimization can be efficiently integrated and operated in the network lifecycle. We validate the proposed architecture by providing a resource allocation case study where we explain how the Time Slotted Channel Hopping mechanism is exploited in our architecture.

Keywords: Digital Twin Network, Internet of Things, Industy 4.0

1. INTRODUCTION

The Internet of Things (IoT) market is growing rapidly in response to the significant development in the areas of smart buildings, smart industries, intelligent transportation, among others. It is expected to grow by 26% during the period 2017-2022 according to Markets and Markets (2017). The IoT is seen as a key technology to enable the fourth industrial revolution with the emergence of the Industrial Internet of Things (IIoT) enabling a broad set of industrial applications to attain full automation and high productivity.

IIoT systems control constrained industrial applications where various requirements need to be satisfied. Reliability, strict real-time constraints and energy requirements are the most crucial ones among others. The highly constrained environment of IIoT networks qualifies them as complex and hard to manage systems. In order to meet the previously mentioned various growing requirements, novel technologies are needed to assist these wireless systems. The Digital Twin (DT) is a strong promising candidate technology for this purpose. In fact, a digital twin is particularly promising for building a continuously updated model of a physical system to enable rapid adaptation to changing dynamics, as highlighted in the paper Grieves and Vickers (2017). In addition, it offers a powerful simulation framework and can enable cognitive capabilities for the replicated system, Barricelli et al. (2019).

DT technology is classified as one of the top ten most promising technological trends for the next decade by Gartner in Cearley et al. (2019). As stated in Markets and Markets (2020), it is expected to grow at a Compound Annual Growth Rate (CAGR) of 58% during the period 2020-2026. The digital replication of the physical system enables intelligent monitoring, innovation and various augmented functionalities. Various fields are already taking advantage of this fast growing technology such as manufacturing Mandolla et al. (2019), healthcare Bruynseels et al. (2018), aerospace Medina et al. (2021), smart city in IEEE (2017), among others.

Recent research work have started to investigate the opportunities that DT could bring to networks. The IETF (Internet Engineering Task Force) draft in Zhou et al. (2021) proposes a reference architecture for the Digital Twin Network (DTN) along with issues and open research

10.1016/j.ifacol.2022.08.007

perspectives related to its implementation. More detailed architectures are proposed in Almasan et al. (2022); Khan et al. (2022) and their implementation is discussed across various aspects such as DTN model construction using machine learning algorithms, procedures to enable new networking functionalities (what-if analysis, troubleshooting, network planning, etc.) along with enabling technologies for efficiently implementing the DTN. Moreover, data collection for the DTN is further addressed in Zhu et al. (2021) where knowledge graphs are leveraged to concretely construct the DTN. With an application-level viewpoint, the DTN is leveraged to predict the global QoS (Quality of Service) by using Graph Neural Networks in Ferriol-Galmés et al. (2021) while an application-driven DTN middleware is proposed in Bellavista et al. (2021).

While DTN-related research are gaining a progressive interest recently, there is still a gap in providing a detailed architecture of a digital twin network for the IIoT given its constrained nature and complex characteristics that should be taken into consideration. We recently proposed in Kherbache et al. (2021), a holistic DT-based architecture that should enable closed-loop IIoT network management and that enables augmented functionalities for the design and during the service of the IIoT. The current paper is a completion and a more in-depth architecture of the DTN for the IIoT taking advantage of Eclipse Hono (Eclipse (2017a)) and Eclipse Ditto (Eclipse (2017b)). Hono acts as a connectivity layer supporting various protocols for IoT devices while Ditto is an open source framework for creating digital IoT twins. The provided architecture enables intelligent networking services such as predictive maintenance, network diagnosis, energy optimization, resource allocation, sustainability, among others. It is designed with a terre-à-terre vision that should allow its efficient implementation.

The remainder of the paper is organized as follows. Section 2 covers related work on the DTN. In Section 3, the architecture of the DTN for IIoT is presented while in Section 4 a resource allocation case study exploiting the IEEE 802.15.4e-TSCH (Time Slotted Channel Hopping) mode in the proposed architecture is discussed. Section 5 concludes the paper.

2. RELATED WORK

A number of research works are investigating the DTN concept, proposing different architectures, aspects and modeling procedures as well as proposing a variety of networking services to unleash the true potential of digital twins in the context of networking. The DTN concept is now being developed by the IETF group in order to standardize it. Their most recent draft Zhou et al. (2021) outlines the basic notions and a reference architecture along with the main challenges and issues for building a DTN.

An ML-based DTN is presented in Almasan et al. (2022) as a key tool for efficient control and management of modern real-world networks. The efficient design of network optimization solutions, performing troubleshooting, simulation analysis and network planning are enabled to network operators by the adoption of DTN. The authors claim that deep learning techniques can be used to model the DTN of a network by considering certain parameters such as traffic, topology, routing, scheduling policies, etc. Then delivering as output some performance metrics related to the network (utilization, delay, packet loss, etc.). By pursuing an iterative feedback-based process between the DT and a network optimizer, the latter can find the best network configuration that meets the requirements set by a network operator.

A high-level architecture for the digital twin of wireless systems is introduced in Khan et al. (2022) in three layers: Physical Interaction Layer, Twin Layer and Services Layer. Two aspects are considered to discuss different details regarding digital twins for wireless systems. Wireless for twins considers the efficient use of wireless resources to enable efficient signaling of twins over a wireless link. On the other hand, twins for wireless discusses the role of digital twins in the implementation of wireless systems.

Zhu et al. (2021) proposes a four-layer architecture for DTN including a physical network layer, a data lake layer, a DT layer, and a network application layer. The data lake layer collects, stores and preprocesses the collected data to provide knowledge and relationship extraction to the DT layer for building the DTN model. The DT layer includes: i) physical entity modeling that complements the modeling of individual network components, ii) requirements modeling that is used to develop a variety of scenario models such as network resource prediction, anomaly detection, automatic operation through AI algorithms, iii) the twin management & control center that manages the DT layer to satisfy the needed requirements. A sketch-based data collection algorithm is proposed to efficiently collect data in the data lake layer and the DTN is built using knowledge graphs.

TwinNet, presented in Ferriol-Galmés et al. (2021), models the relationship between different network characteristics (topology, routing, queue scheduling and ingress traffic) using a digital twin that exploits graph neural networks (GNNs) to predict the overall QoS (prediction of average delay per flow). It includes an optimizer that finds the best routing and/or scheduling policies meeting complex SLAs (Service Level Agreements). The particularity of the proposed model is its generalization capability, since it performs well in network scenarios that have not been seen in the training set.

An Application-driven Digital Twin Networking (ADTN) middleware is proposed in Bellavista et al. (2021) to smooth the interaction with heterogeneous distributed industrial devices and dynamic management of network resources by following an application-level perspective. This architecture is based on a cross-layer SDN approach to actively manage the industrial environment while taking into consideration QoS requirements and network configuration adaptation capabilities.

To enable self-optimization of mobile networks, Deng et al. (2021) presents an algorithmic framework combining digital twin technology, reinforcement learning and expert knowledge. A DTN is built to predict the future state of the network based on its current state. Then, reinforcement learning and expert knowledge extract the predicted state to make optimization decisions and send them back to the digital twin which evaluates these decisions and applies the best one in the physical network. The latter sends the new network state back to the digital twin after applying these decisions after an evaluation time, thus enabling a closed-loop self-optimization of the physical network. In Nguyen et al. (2021), a 5G digital twin is designed to help with the development and deployment of complex 5G networks and enable cost-effective access to 5G, as the deployment of 5G networks is too expensive.

In Zhao et al. (2020), the Software Defined Vehicular Network (SDVN) based on an intelligent digital twin was developed to enable intelligent and adaptive routing in vehicular networks. Another work in Zhang et al. (2022) designed a digital twin network to model an aggregation scheme for edge services of vehicular networks. The DTN reveals potential correspondences between edge services and massive vehicle pairs. The latter divides the complex vehicular network into multiple parts based on potential service associations, which significantly reduces the complexity of scheduling edge services.

In order to enable flow emulation in the DTN, Yang et al. (2021) proposes a comprehensive flow emulation framework for the digital twin network to keep the physical network traffic consistent with the digital twin network traffic. The authors demonstrate, through a case study, how the proposed framework can accurately measure delays without affecting the physical network.

3. DIGITAL TWIN NETWORK FOR IIOT

In this section we describe the DTN architecture for the IIoT as an effort to enable intelligent management and more innovation for such networks.

As depicted in Figure 1, the physical IoT network interacts with the connectivity layer where Eclipse Hono (Eclipse (2017a)) is used. This latter is designed for simplified IoT device connectivity by eliminating the protocol silos of the different devices. In fact, Hono contains for each supported protocol (e.g., HTTP, MQTT or CoaP) a microservice called Protocol Adapter which maps the connection protocol of the device to Hono's APIs (Application Programming Interface). A device registration API is included to make Hono aware of devices that can or will connect to the service. In addition, authentication API is used to verify the identity of the devices willing to connect to Hono. Telemetry and events APIs are used by protocol adapters to send telemetry/events downstream (from devices to business applications). Command & Control API is used by business applications to send commands to connected devices.

Eclipse Hono communicates with back-end business applications using Advanced Message Queuing Protocol (AMQP) 1.0. In our case it is connected to Eclipse Ditto (Eclipse (2017b)) which is an open-source frame-work for creating digital IoT twins. It focuses on back-end scenarios by providing web APIs in order to simplify working with already connected devices and things from customer apps or other back-end software. Ditto ensures that access to twins can only be done by authorized parties via the policies service. Eclipse Ditto structures the data sent by devices via Hono into Digital IoT Twins. A digital twin of an IoT device in Ditto is represented as a Thing

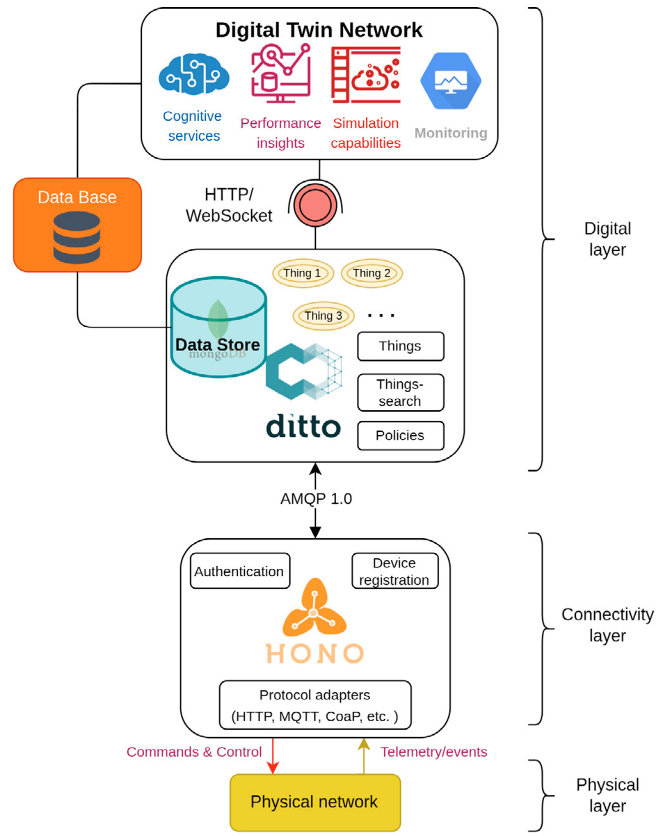

Fig. 1. Digital Twin Network using Eclipse Ditto + Hono.

consisting of attributes and features. The Things service is responsible for persisting things and features in Ditto. The follow-up of the modifications made to the Things, Features and Policies is ensured with the Things-search service. Also, this latter updates an optimized search index and executes queries on it.

Two channels are available for use in Ditto, a twin channel that connects to the digital representation of a Thing and a live channel that routes a command/message towards an actual connected device. Ditto uses MongoDB as a database and only stores the latest state of Things. Furthermore, Ditto provides an HTTP API and a WebSocket to applications to facilitate retrieving Things-related information.

The Digital Twin Network interacts with Ditto via this HTTP API or a WebSocket. The DTN stores things states in the database and updates them wherever a modification to the Things is made in Ditto. This is done because Ditto only stores the latest states of the Things in its data store (mongoDB). This interaction allows storing historical data of the Digital IoT Twins during the entire network lifecycle. As shown in Figure 2, The collected data is exploited to construct different types of models, they can be either basic ones representing the network topology or more advanced ones built for functional purposes (anomaly detection, security management, resource allocation, among others). These models are stored in the database. Moreover, simulation capabilities are provided in the DTN in order to enable running what-if scenarios and getting insights on the network performance in different situations. Network monitoring is also included to better assist the network knowing that it is considered as the human-machine interface between the DTN and the physical IIoT.

The DTN is composed of different components and managed by a central entity called Twin Management & Control. This latter triggers the construction of the logical models when needed, pushes simulation settings in the simulator and receives performance metrics, requests the cognitive services and gets responses from them.

The cognitive services are implemented in the DTN for the IoT network knowing that it is aware of its state. These services run intelligent algorithms continuously with the goal of satisfying QoS/QoE (Quality of Service/ Quality of Experience) requirements giving the available resources. For instance, predictive maintenance as a service has the goal of predicting network anomalies in advance by exploiting the recorded historical data in the DTN. In addition, efficient communication scheduling is enabled by the resource allocation service which runs AI-based algorithms to find the optimal schedule for the IIoT to satisfy the applications requirements. Reinforcement learning is a promising candidate algorithm for communication scheduling problems according to recent research studies (Chilukuri and Pesch (2021)).

Further, sustainability features are included in a sustainability service. Taking into consideration energy, batteries management, monitoring the impact of radio waves and management of wireless nodes that risk to endanger the environment. This service makes use of the mapping and mobility service that is responsible for tracking the network nodes and provide their location coordinates. Besides, the energy optimization service's role is to increase the network lifetime by giving insights on factors that are causing increased energy consumption.

Moreover, network diagnosis is implemented to generate reports giving insights on possible network misconfigurations, nodes failure, security attacks, among others. These reports can be visualized in the GUI (Graphical User Interface) to aid the network engineer in assisting the networking operation. Testing security attacks in the DTN is enabled by the security management service in order to enforce network security. Also, the interoperability service includes formatting capabilities to break the software silos between the different application digital twins. This way, the DTN collaborates efficiently with application digital twins regardless of the technologies used in their implementation.

Finally, the DTN interacts continuously with applications by exposing the network's capability to them. On the other hand, the applications send their QoS/QoE requirements to the DTN which is in charge of finding the best networking configuration to satisfy these needs. An interface between the applications and the DTN is integrated to make a two-way translation of the networking capabilities to the applications and the QoS/QoE requirements to the DTN.

4. RESOURCE ALLOCATION CASE STUDY

In this section we present a resource allocation case study using TSCH (Time Slotted Channel Hopping) mode as a MAC protocol. It was introduced among several channel access modes in the IEEE 802.15.4e amendment, surveyed in Kurunathan et al. (2018). TSCH has been designed to meet the requirements of IIoT applications as it provides time-critical guarantees and very high reliability. In order to exploit the benefits of TSCH, a schedule specification in terms of slots/channels assignment for each network node is required. Various scheduling algorithms for TSCH exist in the literature, they are either centralized, decentralized, static or hybrid according to Urke et al. (2022). This categorization allows distinguishing the way a schedule is generated.

In the proposed DTN architecture, TSCH scheduling can be revolutionized by defining the schedule in the digital layer and leveraging the simulation capabilities provided. In fact, TSCH scheduling in such architecture can be obtained by executing the current physical network scenario in the simulation asset then sending the resulting schedule to the physical layer. Moreover, the resource allocation service of the DTN works continuously to improve the deployed schedule in an effort to satisfy the applications requirements depending on the current state of the physical network. This service also runs what-if scenarios to detect possible bottlenecks due to high traffic loads and predicts possible drops in performance then accommodates the schedule consequently. In addition, the resource allocation service can coordinate with the other intelligent network services to update the schedule, e.g., accommodate the schedule upon a request from the energy optimization service when the remaining energy of some sensor nodes is low.

The first step in developing the DTN using Eclipse Ditto is to define a JSON (JavaScript Object Notation) Thing structure in terms of attributes and features. In fact, the attributes are defined as static values that does not change frequently while features are dynamic state values that change more often. To show our vision in the development process of the DTN for the IIoT, Figure 3 represents a first example prototype of a Thing that represents the IoT devices in a digital form. The policyID is the ID of the access policy for this Thing which describes the authorized parties to read/write its values. One attribute representing the mote type is defined. We also defined some features that contribute in defining a TSCH schedule along with the schedule itself that is updated after executing a scheduling algorithm in the DTN :

- nodeAddress : the address of the IoT device in the physical network ;
- batteryLevel : shows the current battery state of the IoT device, useful for the schedule definition ;
- dataRate : the frequency of sending data packets in the IoT device , useful for the schedule definition ;
- bufferState : gives the number of remaining packets that should be transmitted by the IoT device, useful for the schedule definition ;
- Schedule : defines the schedule for the IoT device in terms of transmission cells (Tx_cells) and reception cells (Rx_cells), both represented by two lists : timeslots and channel offsets. The values of this feature are updated after executing a TSCH scheduling algorithm in the resource allocation service included in the DTN.

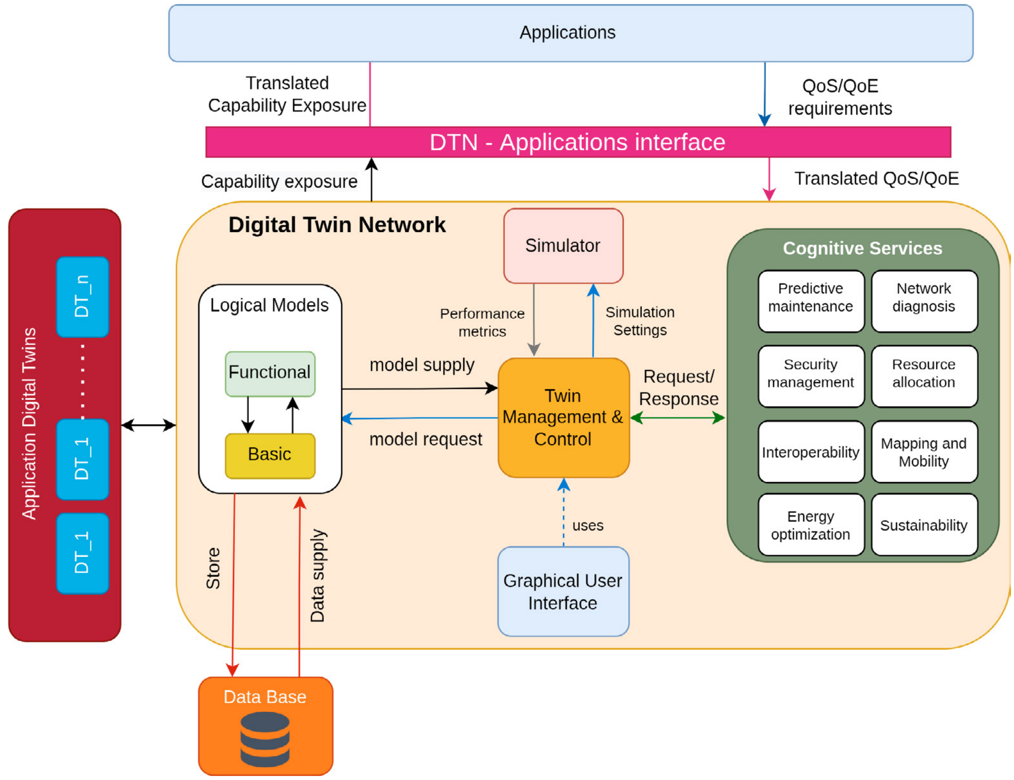

Fig. 2. Digital Twin Network internal architecture.


```
{
  "policyId": "org.acme:my-policy",
  "attributes": {
    "moteType": "OpenMote"
  },
  "features": {
    "nodeAddress": {
      "properties": {
        "value": "1.0"
      }
    },
    "batteryLevel": {
      "properties": {
        "value": "remaining energy value"
      }
    },
    "dataRate": {
      "properties": {
        "value" : "1p/10s"
      }
    },
    "bufferState" : {
      "properties": {
        "remainingPackets" : 0
      }
    },
    "Schedule" : {
      "properties" : {
        "Tx_cells" : {
          "timeslots" : "1, 3, 8",
          "channel_offsets" : "0, 0, 1"
        },
        "Rx_cells" : {
          "timeslots" : "2, 5",
          "channel_offsets" : "0, 0"
        }
      }
    }
  }
}
```


Fig. 3. Thing structure in Eclipse Ditto for the TSCH scheduling case.

It is envisaged to further enrich the Thing structure in the future by considering more features that should allow giving a more precise state of the IoT device.

Advanced intelligent methods can be exploited for TSCH scheduling in the DTN architecture because the energy constraint is excluded in the digital layer. For instance, using reinforcement learning algorithms as they seem to adapt well and give promising results in this context, as in Chilukuri and Pesch (2021) where a deep reinforcement learning scheduling algorithm for TDMA-based time constrained wireless networks is proposed. In fact, such algorithms require a learning phase in order to converge to an optimal policy, but this can considerably increase the energy expenditure if this phase is executed on the physical network. In our architecture, the optimal policy is learned in the DTN by means of simulation.

Furthermore, the proposed architecture can take into account application digital twins requirements. For instance, deadlines requirements of a time-sensitive application DT are sent to the DTN which is in charge of finding the optimal policies satisfying the aforementioned needs. The Twin Management & Control (TM&C) entity interpretes the received requirements and understands that the application packets should reach their destinations within a fixed deadline. After that, the resource allocation service is requested to adapt the communication schedule according to the received requirements. It simulates various scheduling algorithms by interacting with the TM&C which in turn defines the simulation settings and sends them to the simulation asset. This latter returns the performance metrics for each simulation, the TM&C analyzes these results and decides which policy responds best to the deadlines requirements.

5. CONCLUSION

In this paper we proposed a DTN architecture for the IIoT using two open-source tools. First, Eclipse Hono which acts as a connectivity layer eliminating protocol silos of the different IoT devices. Second, Eclipse Ditto which is a framework for building digital IoT twins and provides APIs for business applications to easily interact with these DTs. In our case it is a DTN application that is implemented on top of Ditto and interacts with it to record the networking operation all along its lifecycle. The proposed architecture enables implementing cognitives services such as predictive maintenance, network diagnosis, resource allocation, security management, sustainability, among others. Besides, network monitoring is enabled in the DTN along with simulation capabilities. All of the cognitive services exploit the simulation capabilities provided in the DTN to run what-if scenarios giving performance insights on the networking operation.

Finally, we presented a resource allocation case study by focusing on the service responsible for that and supposing the use of IEEE802.15.4-TSCH mode. An example structure of a Thing in Ditto is also presented with the attributes and features that should help in communication scheduling between the network nodes.

In the future, it is envisaged to develop the DTN application on top of Eclipse Ditto and develop the resource allocation and energy optimization services and assess their impact on the networking performance.

REFERENCES

Almasan, P., Ferriol-Galmés, M., Paillisse, J., Suárez-Varela, J., Perino, D., López, D., Perales, A.A.P., Harvey, P., Ciavaglia, L., Wong, L., et al. (2022). Digital twin network: Opportunities and challenges. *arXiv preprint arXiv:2201.01144*.

Barricelli, B.R., Casiraghi, E., and Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7, 167653–167671. doi:10.1109/ACCESS.2019.2953499.

Bellavista, P., Giannelli, C., Mamei, M., Mendula, M., and Picone, M. (2021). Application-driven network-aware digital twin management in industrial edge environments. *IEEE Transactions on Industrial Informatics*, 17(11), 7791–7801. doi:10.1109/TII.2021.3067447.

Bruynseels, K., Santoni de Sio, F., and van den Hoven, J. (2018). Digital twins in health care: ethical implications of an emerging engineering paradigm. *Frontiers in genetics*, 9, 31.

Cearley, D., Burke, B., Smith, D., Jones, N., Chandrasekaran, A., and Lu, C. (2019). Top 10 strategic technology trends for 2020. Technical report, Gartner, Stamford, CT, USA.

Chilukuri, S. and Pesch, D. (2021). Recce: Deep reinforcement learning for joint routing and scheduling in time-constrained wireless networks. *IEEE Access*, 9, 132053–132063. doi:10.1109/ACCESS.2021.3114967.

Deng, J., Zheng, Q., Liu, G., Bai, J., Tian, K., Sun, C., Yan, Y., and Liu, Y. (2021). A digital twin approach for self-optimization of mobile networks. In *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 1–6. doi:10.1109/WCNCW49093.2021.9420037.

Eclipse (2017a). Connect, Command & Control IoT devices. URL `https://www.eclipse.org/hono/`. [Online; Accessed 2022-03-29].

Eclipse (2017b). Eclipse Ditto™ • open source framework for digital twins in the IoT. URL `https://www.eclipse.org/ditto/`. [Online; Accessed 2022-03-15].

Ferriol-Galmés, M., Suárez-Varela, J., Paillise, J., Shi, X., Xiao, S., Cheng, X., Barlet-Ros, P., and Cabellos-Aparicio, A. (2021). Building a digital twin for network optimization using graph neural networks. *Available at SSRN 3995236*.

Grieves, M. and Vickers, J. (2017). *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, 85–113. doi:10.1007/978-3-319-38756-7_4.

IEEE (ed.) (2017). *Smart city digital twins*. Honolulu, Hawaii, USA.

Khan, L.U., Han, Z., Saad, W., Hossain, E., Guizani, M., and Hong, C.S. (2022). Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities. *arXiv preprint arXiv:2202.02559*.

Kherbache, M., Maimour, M., and Rondeau, E. (2021). When digital twin meets network softwarization in the industrial iot: Real-time requirements case study. *Sensors*, 21(24). doi:10.3390/s21248194. URL `https://www.mdpi.com/1424-8220/21/24/8194`.

Kurunathan, H., Severino, R., Koubaa, A., and Tovar, E. (2018). Ieee 802.15.4e in a nutshell: Survey and performance evaluation. *IEEE Communications Surveys Tutorials*, 20(3), 1989–2010. doi:10.1109/COMST.2018.2800898.

Mandolla, C., Petruzzelli, A.M., Percoco, G., and Urbinati, A. (2019). Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry. *Computers in Industry*, 109, 134–152.

Markets and Markets (2017). Internet of things (iot) market. URL `https://www.marketsandmarkets.com/Market-Reports/internet-of-things-market-573.html`.

Markets and Markets (2020). Digital twin market. URL `https://www.marketsandmarkets.com/Market-Reports/digital-twin-market-225269522.html`.

Medina, F.G., Umpierrez, A.W., Martínez, V., and Fromm, H. (2021). A maturity model for digital twin implementations in the commercial aerospace oem industry. In *2021 10th International Conference on Industrial Technology and Management (ICITM)*, 149–156. doi:10.1109/ICITM52822.2021.00034.

Nguyen, H.X., Trestian, R., To, D., and Tatipamula, M. (2021). Digital twin for 5g and beyond. *IEEE Communications Magazine*, 59(2), 10–15.

Urke, A.R., Kure, Ø., and Øvsthus, K. (2022). A survey of 802.15.4 tsch schedulers for a standardized industrial internet of things. *Sensors*, 22(1). doi:10.3390/s22010015. URL `https://www.mdpi.com/1424-8220/22/1/15`.

Yang, H., Li, Y., Yao, K., Sun, T., and Zhou, C. (2021). A systematic network traffic emulation framework for digital twin network. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 94–97. doi:10.1109/DTPI52967.2021.9540090.

Zhang, K., Cao, J., and Zhang, Y. (2022). Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks. *IEEE Transactions on Industrial Informatics*, 18(2), 1405–1413. doi:10.1109/TII.2021.3088407.

Zhao, L., Han, G., Li, Z., and Shu, L. (2020). Intelligent digital twin-based software-defined vehicular networks. *IEEE Network*, 34(5), 178–184.

Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and Jacquenet, C. (2021). Digital Twin Network: Concepts and Reference Architecture. Internet-Draft draft-zhou-nmrg-digitaltwin-network-concepts-06, Internet Engineering Task Force. URL `https://datatracker.ietf.org/doc/html/draft-zhou-nmrg-digitaltwin-network-concepts-06`. Work in Progress.

Zhu, Y., Chen, D., Zhou, C., Lu, L., and Duan, X. (2021). A knowledge graph based construction method for digital twin network. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*, 362–365. doi:10.1109/DTPI52967.2021.9540177.