Core HyperTwin: A Core Network Digital Twin Enabler

Marcos Lima Romero, Diego Mello Costa Senese, Eduardo de Souza Leite, Jeferson Sabino Sacramento Schiavinato, Danilo Tocantins Rufo Costa and Ricardo Suyama

***Abstract*— Core Networks in Mobile Network Operators always faced an enormous complexity due to the new generations that come every decade and now have to deal with the convergence of multiple generations of technology (2G, 3G, 4G, 5G) and soon the 6G. This complexity makes it difficult to thoroughly test, validate, and ensure the security of new solutions, features, or configurations. To address this, we introduce Core HyperTwin, a platform that leverages a practical implementation of the Digital Twin concept to enable realistic simulations and flexible testing environments. By implementing key protocols such as Diameter, HTTP/2, SOAP, and GTP, and utilizing open-source Python components, Core HyperTwin empowers operators to reduce their time-to-market, accelerate innovation, and improve network security without impacting real customers. As a proof of concept, we successfully emulated a P-GW communicating with a PCRF using a client hosted in a virtual machine within the NFV infrastructure in the same IP network that established a Diameter link, sent customized messages opening a 3GPP Gx session, opening a legitimate 3GPP Sy session with the OCS, as if initiated by a real customer, updated these sessions, and terminated it. The tests already suggested improvements in the security and allowed new feature analysis that was not possible before without using real user traffic. The next steps include leveraging the solution to interconnect with different emulated elements, test security vulnerabilities, perform stress tests of TPS, model real traffic shapes, and benchmark different vendor solutions.**

***Keywords*— Digital Twin, Core Network, Network Simulation, Automation, Telecommunications**

I. Introduction

In the context of the Mobile Network Operators (MNO) market, as in any other highly competitive market, it is mandatory to adapt quickly to changes and to the market and regulatory demands imposed. MNOs face an even greater need to keep up with the technology generation changes that are occurring in shorter periods, as seen in the transition from 4G to 5G, with discussions about 6G already underway [1]. DevOps could be an answer to rapidly respond to technology changes and respond to time-to-market demand.

DevOps can be summarized as end-to-end automation in software development and delivery [2]. However, to apply it to organizations, there is a need for cultural change. The DevOps culture is an initiative that aims to bring more agility to processes and the relationship between development, delivery, and operations. It has already been present in the software industry and has also made some inroads in the telecommunications sector, mainly in the software-defined Network Function Virtualization (NFV) infrastructure [3].

An unmet need in the literature and industry is the limited application of DevOps concepts, such as Continuous Integration and Continuous Delivery (CI/CD), to Core Network Elements (NE)s, in both 5G and legacy networks. The main challenge is to ensure fast delivery without impacting production or users, considering the high availability and reliability. Normally, based on our own experience, the acceptance test procedure of a new element in production is performed meticulously and manually, sometimes over weeks or months. This process validates communication between other elements, expected traffic scenarios, billing, and security issues before migrating real production traffic to it.

Digital Twins (DTs) can bridge the gap between the necessities of DevOps and the intrinsic complexity of the Core Network. First presented by NASA for simulation-based systems engineering [4], the DT concept was defined as a high-fidelity digital representation of a physical entity or system. This representation exchanges real-time data with its physical counterpart, enabling it to mimic the behavior of the physical twin and vice versa [5]. Although DTs have been extensively explored in the manufacturing industry, they are now gaining traction in other sectors, including Telecommunications, such as in fixed networks and data centers [6]. In fact, in January 2024, the IEEE Network journal dedicated a special issue to Network Digital Twins, receiving 34 submissions and accepting 9 high-quality articles, highlighting the growing interest and relevance of DTs in this field [7].

In the last decade, some initiatives have emerged to address the challenges of testing Core Networks. A notable approach, presented in 2013 [8], focused on testing the Core Network with real data modeling but only emulated the Access Layer (eNodeB). Another work citing DTs in Core NEs was presented in 2022 [9], but its objective was to validate the configurations of the elements and did not involve simulating traffic operations. To the best of our knowledge, there was no research applying Network Digital Twins to 4G or 5G Core Networks capable of interact sending traffic to real elements.

In recent years, several investigations have suggested the use of Digital Twins in 5G networks [10], [11], [12], [13], and in 6G Networks [14], [15], [16]. In addition, with increasing attention from telecommunications vendors such as Ericsson

Marcos Romero, CECS/Core Operations, UFABC/TIM, Santo André-SP, e-mail: marcos.romero@ufabc.edu.br; Diego Mello Costa Senese, Core Development, TIM, São Paulo-SP, e-mail: dsenese@timbrasil.com.br; Eduardo de Souza Leite, Core Operations, TIM, Santo André-SP, e-mail: eleite@timbrasil.com.br; Jeferson Sabino Sacramento Schiavinato, Core Operations, TIM, Santo André-SP, e-mail: jsacramento@timbrasil.com.br; Danilo Tocantins Rufo Costa, Core Operations, TIM, Natal-RN, e-mail: dtr-costa@timbrasil.com.br; Ricardo Suyama, CECS, UFABC, Santo André-SP, e-mail: ricardo.suyama@ufabc.edu.br.

[17] and Huawei [18]. However, to the best of our knowledge, no real-world use case has been fully implemented or reported in a commercial MNO. Core HyperTwin stands out from traditional virtualized network elements by offering unparalleled message routing flexibility, all without the complex integrations required for a real network element. Rather than introducing another layer of virtualization, it delivers complete control over messages, eliminating the need for intricate integrations typically associated with real network elements.

In order to address this gap, we propose a Core Network Digital Twin model, flexible, scalable, and realistic, capable of emulating real Core NEs. Within this model, new elements, configurations, or features can be tested, validated, and approved with total flexibility of protocols and parameters, without affecting any real users. The main contributions of this research are:

- Evaluate the concept of Digital Twins and DevOps in a sector not yet fully explored (MNOs).
- Present an architecture capable of emulating Core NEs.
- Provide a proof-of-concept testing traffic scenarios and auditing security aspects from Core NEs in an automated and fast manner.

The remainder of this paper is structured as follows. Section II delves into the background and technical context, providing a comprehensive overview of Core Network architectures and protocols. Section III introduces our proposed Core HyperTwin framework, detailing its architecture, components, and functionalities. Section IV presents the results of our proof-of-concept implementation, highlighting the platform's effectiveness in emulating real-world scenarios and validating network configurations. Finally, Section V concludes the paper, summarizing the key findings and discussing future directions for the research and development of Core HyperTwin.

II. Background

A. Core Network Architecture and Protocols

In a mobile network, every mobility change, call, or Internet access depends on an antenna connected and sending the information to the Core Network; the Core is responsible for controlling the capabilities allowed to each subscriber according to the provisioning information. Additionally, the Core handles the billing and signaling necessary to complete calls or assign IP addresses to subscribers and route traffic to external Internet gateways according to established policies.

The Core Network consists of interconnected elements, each playing a specific role defined by standards organizations such as the 3rd Generation Partnership Project (3GPP) [19]. Using the example of the 4G network, Core elements can include the Home Subscriber Server (HSS), storing subscriber data and authentication information; Mobility Management Entity (MME), controlling the user's location and handover between areas; the Policy and Charging Rules Function (PCRF), enforcing policies decisions and managing charging rules for data connections; the Packet Data Network Gateway (P-GW), providing connectivity between the mobile network and external data networks. These elements communicate with each other using an established protocol called Diameter Base Protocol, as defined by the International Engineering Task Force (IETF) in RFC 6733 [20], and the GPRS Tunneling Protocol GTP defined by 3GPP in TR 29.274 [21]. The correct operation of these components is essential for reliable call setup, data sessions, mobility management, and billing accuracy.

B. 4G Attach Procedure

Figure 1 shows a typical 4G network flow for an Attach Procedure summarized, highlighting the involved Core NEs and the protocols used. First, a user equipment (UE) connects with an eNodeB (the radio access entity of 4G) through the Radio Resource Control (RRC) protocol defined by 3GPP in TS 36.331 [22]. The eNodeB then initiates communication with MME by sending the Attach Request and Packet Data Network (PDN) Connectivity Request message via the S1 Application Protocol (S1AP) defined by 3GGP in TS 36.413 [23].

The MME requests Authentication Info from the HSS and proceeds to authenticate the UE, which in turn must authenticate the operator's network. Once authenticated, the MME requests an Update Location, receiving the subscriber's profile from HSS via S6a interface using the Diameter protocol, as defined by 3GPP in TS 29.272 [24].

Based on this profile, MME requests the Serving Gateway (S-GW) and P-GW to create a session and establish the default data bearer via interfaces S11 and S5 using the GTP protocol. The P-GW then queries the PCRF about the policies and rules associated with the subscriber on the Gx interface with the Credit Control Request Init message (CCR-I), as defined by 3GPP in TS 29.212 [25]. The PCRF, in turn, requests the Online Charging System (OCS) to create a Sy session with the Spending Limit Request (SLR) associated with the subscriber, using Diameter protocol, as defined by 3GPP in TS 29.219 [26].

Finally, PCRF returns the session rules to the P-GW, S-GW and P-GW return the success messages, and the Attach Procedure is completed, allowing the default bearer to be established with the UE.

III. Proposed Framework

A. Architecture

Considering the current scenario, we addressed the problem by implementing a hexagonal architecture [27], as shown in Figure 2. This architectural pattern, also known as the "ports and adapters" pattern, promotes a loose coupling between the application's core domain and its external dependencies. This design choice allows greater flexibility, testability, and maintainability of the system, aligning well with the dynamic nature of the telecommunications environment and the principles of DevOps.

In the Core Logic layer, the main entities of the architecture are defined: Network Elements, Test Templates, and Users. The application layer implements the use cases and business rules: Real-time twin traffic mirroring, network simulation, What if analysis, KPI reporting, AI training and prediction, and topology building. The infrastructure layer provides the necessary tools and adapters to communicate with external

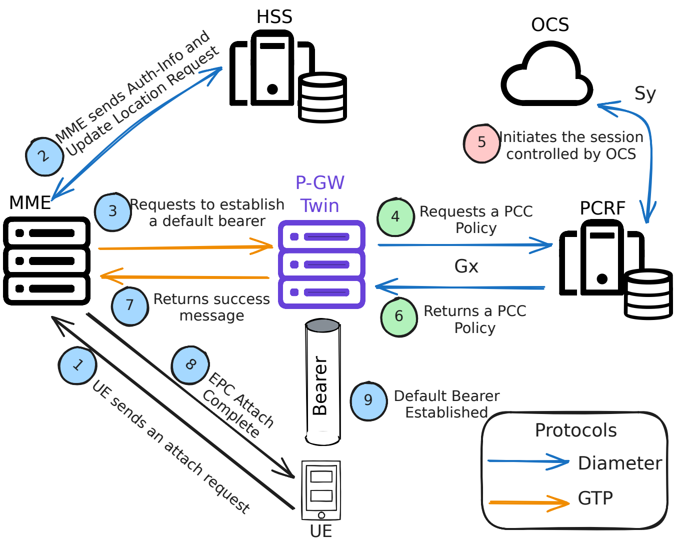

Fig. 1. Simplified 4G Attach Procedure typical flow.

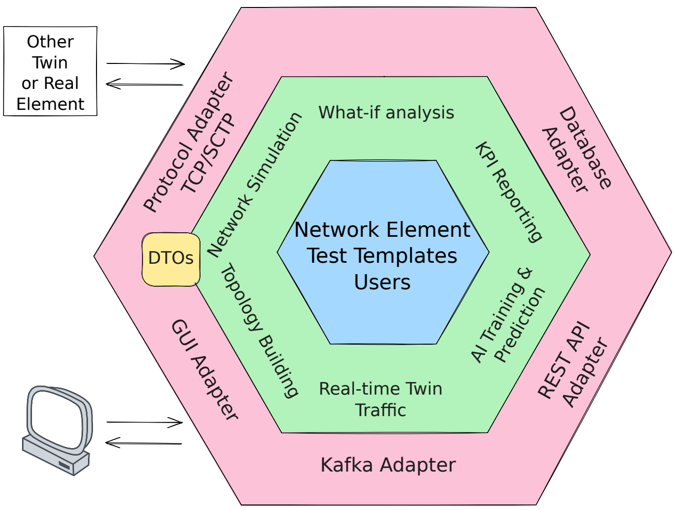

Fig. 2. Core HyperTwin basic architecture.

applications: Protocols, GUI, Kafka, REST API, and Database Adapters.

In the architecture presented, Data-Transfer Objects (DTOs) are used to standardize the exchange of data between the different layers and components of the Core HyperTwin platform. These DTOs encapsulate the requirements for specific operations, such as configuring a network element, defining a test scenario, or retrieving simulation results.

A user-friendly Graphical User Interface (GUI) is proposed to facilitate the configuration of new Network Elements, Test Templates, and Use Cases. The GUI will also provide access to KPI reports, real-time monitoring of test results, and visualization tools to analyze network behavior and performance metrics.

Through a high-performance stream processing platform, such as Apache Kafka, it is possible to receive real-time data from real NEs and mirror it into the Digital Twin environment, enabling dynamic synchronization of the virtual and physical systems.

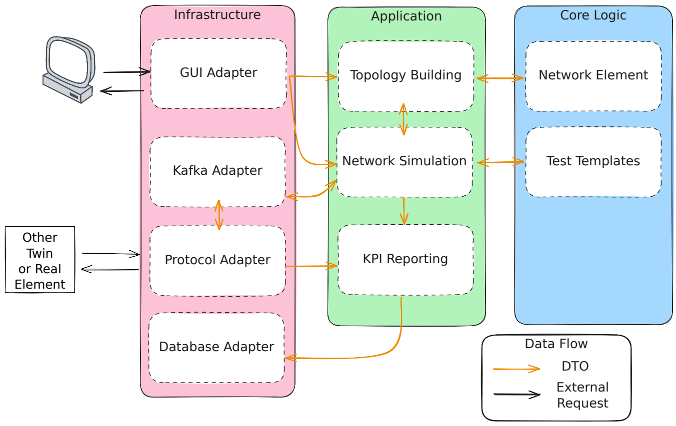

Fig. 3. Flow of data between the different layers and components of Core HyperTwin.

The protocol adapter will be responsible for encapsulating the necessary application protocols, such as Diameter, into the appropriate transport protocols (TCP or SCTP), ensuring seamless communication between the Core HyperTwin platform and external network elements or systems.

Furthermore, the architecture was formulated following the concept of microservices, offering high flexibility. This allows for the future deployment of fully Digital Twin networks, connecting each module to its corresponding Twin Element. Simultaneously, it also enables interconnection with real NEs in the production network.

B. Data Flow

The diagram illustrated in Figure 3 presents an example of a data flow within the Core HyperTwin framework, showing the interactions between its core logic, application, and infrastructure layers.

External requests, whether originating from the GUI, other Twin/Real elements, or other external systems, initiate the data flow. These requests can be configuration commands, test scenarios, data retrieval requests, or any other interaction with the system.

In this diagram, the Network Simulation use case is illustrated. A user opens the Core HyperTwin GUI and designs a network topology by dragging and dropping virtual representations of network elements onto a canvas, configuring their interconnection and parameters. The user then selects predefined test templates and applies them to the topology, initiating the simulation.

The GUI Adapter receives the configuration data, encapsulates them in DTOs, and passes them to the Application Layer. The Topology Building module constructs the virtual network, while the Network Simulation module prepares the simulation environment and generates commands, which are then passed to the Kafka Adapter. Acting as a message broker, the Kafka Adapter publishes these commands to specific topics, where they are consumed by the Protocol Adapter.

The Protocol Adapter interprets the commands, translates them into protocol messages (e.g., Diameter), establishes connections with the Digital Twin or real network elements, and sends the messages. Upon receiving responses, it translates them back and publishes them to Kafka topics.

The Network Simulation module processes these responses, updating the simulation state and generating further commands if needed. The KPI Reporting module collects the simulation data, calculates the KPIs and passes them to the Database Adapter for storage. The GUI Adapter retrieves these KPIs and presents the reports and real-time test results to the user in the GUI.

This domain-driven data flow, enabled by Core HyperTwin, ensures seamless interaction between the user, the simulated network elements, and the various modules responsible for configuration, simulation, analysis, and reporting. This facilitates efficient and realistic testing of Core Network functionalities without impacting the live production environment.

IV. Proof-of-Concept

To validate the feasibility of the Core HyperTwin platform, we conducted a proof-of-concept (PoC) within a live MNO's environment. The PoC focused on emulating the interaction between two critical Core NEs: PCRF and P-GW. The P-GW was the Twin Element and PCRF was a real element during the acceptance procedure phase.

A. PoC Setup

The PoC environment consisted of a virtual machine (VM) with RedHat® 8.9 operating system, 32 vCPUs, and 64 GB RAM, hosted within the MNO's Network Functions Virtualization (NFV) infrastructure in a VMWare® environment. This VM housed the Core HyperTwin client, which was configured to emulate the behavior of a P-GW. The client was connected to the PCRF under acceptance over the same IP network used by real network elements.

To achieve successful emulation, we used a Python Diameter library adapted from the PyHSS repository[1] available through the GNU Affero General Public License (AGPL) license. Using this library, we deployed a P-GW client handling the Diameter connection to communicate with the PCRF Diameter server, using Socket[2] and AsyncIO[3] Python libraries.

B. Test Scenarios

The client was configured to run a series of test scenarios designed to replicate real-world traffic patterns and validate the functionality of the emulated P-GW. These scenarios included:

- Diameter Association Establishment: The Core HyperTwin client established a Diameter connection with the production PCRF sending the Capabilities Exchange Request and maintaining the connection opened answering Device Watchdog requests, simulating the normal behavior of a Gx interface.
- Gx Session Creation and Modification: The client sent customized Credit Control Requests to the PCRF, adhering to the 3GPP Gx interface specifications, to create and modify a simulated user session.

[1]https://github.com/nickvsnetworking/pyhss
[2]https://docs.python.org/3/library/socket.html
[3]https://docs.python.org/pt-br/3/library/asyncio.html

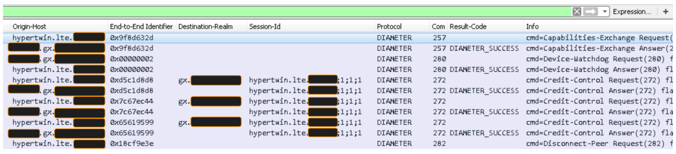

Fig. 4. Capture print showing messages exchanged by Core HyperTwin.

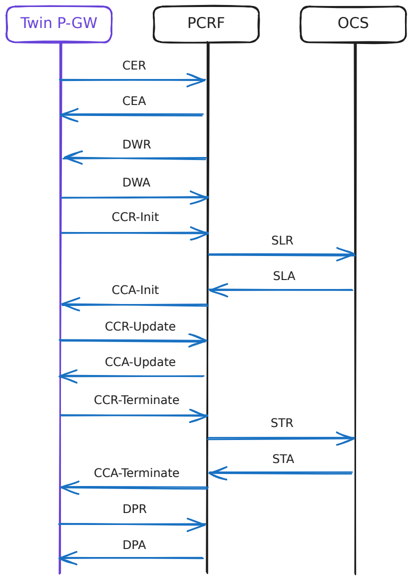

Fig. 5. Call flow showing messages exchanged by Core HyperTwin emulating a P-GW.

- Sy Session Establishment: The PCRF, in turn, initiated a legitimate Sy session with the OCS, mirroring the behavior of a real user device requesting Spending Limits.
- Session Update and Termination: The client sent further Diameter messages to update the session parameters (CCR-Update) and eventually terminate the session gracefully (CCR-Terminate) which triggers PCRF to close the Sy session with the OCS by sending Session Terminate Request (STR).
- Diameter Association Gracefully Disconnection: The client sends a Disconnect Peer Request (DPR) performing a graceful termination of the link.

C. Results

As shown in Figures 4 and 5, the PoC successfully demonstrated the ability of Core HyperTwin to emulate the behavior of a P-GW and interact with a production PCRF in a realistic manner; some confidential data were hidden. The emulated P-GW established Diameter connections, exchanged Gx and Sy messages, and created, modified, and terminated sessions as expected. The real-time traffic mirroring functionality allowed the observation and analysis of the signaling traffic between the emulated P-GW and the PCRF, providing valuable insight into network behavior and potential issues.

Furthermore, the PoC revealed potential security improvements and enabled the analysis of new features previously not possible without using real user traffic. This highlights the potential of Core HyperTwin to accelerate innovation and enhance the network in a safe and controlled environment.

V. Conclusions

This paper presented Core HyperTwin, a platform that leverages a practical implementation of the Digital Twin concept to enable realistic simulations and flexible testing environments for Core NEs. However, it is important to acknowledge that Core HyperTwin is still in its early stages, and further development is needed to expand its capabilities and address potential limitations.

The complexity and diversity of NEs, protocols, and interfaces require substantial development efforts. Additionally, ensuring real-time synchronization and maintaining the fidelity of the Digital Twin as the network evolves can be demanding. Scalability is another concern, as simulating a large-scale network with numerous elements and interactions requires significant computational resources and efficient algorithms. Addressing these challenges will be crucial for the widespread adoption and practical implementation of Core HyperTwin in real-world mobile network operator environments.

Future work will focus on expanding the capabilities of Core HyperTwin to enable a wider range of Core NEs and new generations such as 5G and 6G. This includes the development of Digital Twin models for other essential components such as the HSS, MME, and OCS, as well as the implementation of additional protocols and interfaces such as HTTP/2. Furthermore, we plan to explore the integration of machine learning and artificial intelligence algorithms to enhance the platform's ability to predict network behavior, detect anomalies, and optimize performance.

Core HyperTwin represents a paradigm shift in the way MNOs approach Core Network testing and validation. By leveraging the power of Digital Twins and state-of-the-art technologies, it empowers operators to embrace a DevOps culture, fostering collaboration between development and operations teams, and enabling Continuous Integration and Delivery of new features and services. This innovative approach not only accelerates innovation and reduces time-to-market but also enhances network security and reliability, ultimately revolutionizing the way MNOs manage and evolve their Core Network infrastructure.

Aknowledgements

This work was originally conceived during the HackaTIM event, an internal hackathon provided by the TIM S/A HR and CTO teams, and was the winner of the Continuous Quality Improvement category. We appreciate the support offered by the company.

References

[1] A. R. Mishra, *Fundamentals of network planning and optimisation 2G/3G/4G: evolution to 5G*. John Wiley & Sons, 2018.

[2] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *IEEE software*, vol. 33, no. 3, pp. 94–100, 2016.

[3] H. Karl, S. Dräxler, M. Peuster, A. Galis, M. Bredel, A. Ramos, J. Martrat, M. S. Siddiqui, S. Van Rossem, W. Tavernier, *et al.*, "Devops for network function virtualisation: an architectural approach," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1206–1215, 2016.

[4] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, "Draft modeling, simulation, information technology & processing roadmap," *Technology area*, vol. 11, pp. 1–32, 2010.

[5] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital twin: Origin to future," *Applied System Innovation*, vol. 4, no. 2, p. 36, 2021.

[6] P. Almasan, M. Ferriol-Galmés, J. Paillisse, J. Suárez-Varela, D. Perino, D. 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, "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 22–27, 2022.

[7] Y. Cui, J. Liu, M. Yu, J. Jiang, L. Zhang, and L. Lu, "Network digital twin," *IEEE Network*, vol. 38, no. 1, pp. 5–6, 2024.

[8] P. Varga and P. Olaszi, "Lte core network testing using generated traffic based on models from real-life data," in *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, 2013.

[9] Y. Zheng, C. Lin, S. Huang, W. Li, K. Wang, and J. Lou, "Research and application of configuration system in core network based on digital twin," in *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 5, pp. 293–297, 2022.

[10] V. Pantovic, D. Milovanovic, D. Starcevic, and Z. Bojkovic, "5g mobile networks and digital twins concept: Research challenges in network dt emulation," in *2022 4th International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*, pp. 1–4, 2022.

[11] N. U. Saqib, S. Song, H. Xie, Z. Cao, G.-J. Hahm, K.-Y. Cheon, H. Kwon, S. Park, S.-W. Jeon, and H. Jin, "Digital twin enabled cellular network management and prediction," *ICT Express*, 2024.

[12] A. Mozo, A. Karamchandani, M. Sanz, J. I. Moreno, and A. Pastor, "B5gemini: Digital twin network for 5g and beyond," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6, 2022.

[13] Z. Tao, Y. Guo, G. He, Y. Huang, and X. You, "Deep learning-based modeling of 5g core control plane for 5g network digital twin," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 1, pp. 238–251, 2024.

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

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

[16] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, "6g digital twin networks: From theory to practice," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 72–78, 2023.

[17] F. Foo, "Digital twins catalyst reflections from digital transformation world," 2019. Retrieved May 20, 2024, from https://www.ericsson.com/en/blog/2019/6/digital-twins-catalyst-booth-reflections-from-digital-transformation-world.

[18] "Huawei launches industry's first site digital twins based 5g digital engineering solution," 2020. Retrieved May 20, 2024, from https://www.huawei.com/en/news/2020/2/site-digital-twins-based-5g-digital-engineering-solution.

[19] 3rd Generation Patnership Project, "3GPP - the mobile broadband standard." Retrieved May 20, 2024, from https://www.3gpp.org/.

[20] IETF, "Diameter base protocol," Tech. Rep. RFC 6733, IETF, 2012.

[21] 3GPP, "3gpp evolved packet system (eps); evolved general packet radio service (gprs) tunnelling protocol for control plane (gtpv2-c); stage 3 (release 8)," Tech. Rep. TR 29.274, 3GPP, 2015.

[22] 3GPP, "Evolved universal terrestrial radio access (e-utra); radio resource control (rrc); protocol specification," Tech. Rep. TS 36.331, 3GPP, 2007.

[23] 3GPP, "Evolved universal terrestrial radio access network (e-utran); s1 application protocol (s1ap)," Tech. Rep. TS 36.413, 3GPP, 2007.

[24] 3GPP, "Evolved packet system (eps); mobility management entity (mme) and serving gprs support node (sgsn) related interfaces based on diameter protocol," Tech. Rep. TS 29.272, 3GPP, 2008.

[25] 3GPP, "Policy and charging control (pcc); reference points," Tech. Rep. TS 29.212, 3GPP, 2007.

[26] 3GPP, "Policy and charging control: Spending limit reporting over sy reference point," Tech. Rep. TS 29.219, 3GPP, 2012.

[27] A. Cockburn, "Hexagonal architecture," *The Pattern: Ports and Adapters*, 2005.