applied sciences

Article

Digital Twin-Driven Virtual Network Architecture for Enhanced Extended Reality Capabilities

Xavier Calle-Heredia and Xavier Hesselbach *

Department of Network Engineering, Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, E-08034 Barcelona, Spain; xavier.calle@upc.edu
* Correspondence: xavier.hesselbach@upc.edu

Abstract: Extended Reality (XR) technology is proposed as a key enabler for the development of immersive applications, thus transforming the way users interact within digital environments. In communication networks, XR will be able to introduce immersive network functions with the potential to enhance the network experience. This paper proposes a comprehensive architectural framework designed to support the deployment of XR Functions (XRFs) within a Digital Twin Network (DTN) environment in order to provide immersive functions of any type, including control and management functions, that are not available in the original network. The proposal can be deployed in relevant fields where digital twins are promising, such as medical applications, surveillance systems, autonomous driving and space communications. The proposed architecture must consider the need for minimal hardware and software resources, along with a network topological configuration with the objective of providing a robust yet lightweight framework, allowing devices with limited computing power, such as smartphones and laptops, to benefit from the enhanced network functionalities provided in the Digital Twin Network, without affecting the original network (ON).

Keywords: extended reality functions; digital twin networks; immersive network functions; network architecture

check for updates

Citation: Calle-Heredia, X.; Hesselbach, X. Digital Twin-Driven Virtual Network Architecture for Enhanced Extended Reality Capabilities. *Appl. Sci.* **2024**, *14*, 10352. https://doi.org/10.3390/app142210352

Academic Editor: Mirosław Klinkowski

Received: 17 October 2024
Revised: 8 November 2024
Accepted: 8 November 2024
Published: 11 November 2024

1. Introduction

In this research, a novel intelligent network empowered with extended Reality Functions (XRFs) is introduced. The novel network exploits the Digital Twin (DT) technology to deploy a Digital Twin Network (DTN). Consequently, the novel network is referred to in this paper as Digital Twin Network plus extended Reality Functions (DTN + XRFs). The DTN can be deployed with different approaches such as network virtualization (NV), while advanced virtualization technologies can be leveraged including OpenStack and open vSwitch, among others. The DTN is realized as a replication of an original network (ON), which can either be a virtual or a physical network. The manager responsible for the ON replication and the deployment and management of the Digital Twin Network is referred to as the Digital Twin Manager (DTM). Furthermore, the DTM is in charge of the synchronization process between the ON and the DTN, which is an asymmetric process that will be explained in Section 3.1, while the XRF module is designed with the objective of enhancing the network experience by means of novel immersive network functionalities including: optimal routing, interactive network status visualizations, critical nodes identification, resource locator and smart monitoring, among others.

To assist with the DTN + XRF deployment, a supporting architectural framework is required. The supporting architecture must be robust and reliable in order to guarantee appropriate twinning, synchronization and self-management processes with the aim of deploying both the DTN and the XRF empowerment module. The DTN + XRF is conceived as an enhanced and portable Digital Twin replica of the original network; consequently, mobile devices including smartphones, tablets and laptops, are identified as Digital Twin

Client Devices (DTCDs). Since the DTCDs commonly lack sophisticated computational processors, it is proposed that the supporting architectural framework offloads expensive computations from DTCDs by leveraging a local data center equipped with two main modules: the Digital Twin Manager (DTM) and the extended Reality Manager (XRM). DTCDs can access data center services via application programming interfaces (APIs), thus ensuring standardization. Moreover, the DTN + XRF is envisioned as an any-location deploying network including extra-terrestrial (space) stations. This paper considers the DTN + XRF deployment in a space station. However, the disadvantageous channel conditions present in space communications can severely compromise the ON-DTN communication and synchronization, thereby raising the necessity to identify mechanisms and tolerant communication protocols to alleviate the aforementioned communication issues.

It is crucial to emphasize that the proposed intelligent network offers the possibility to access to the replica of any ON including its services and resources. However, this proposal is not aligned with the Network as a Service (NaaS) concept. Rather, this present proposal is a replication from an original network where the XR services are deployed, and is not owned by a third party.

The rest of this document is organized as follows: In Section 2, a literature analysis is presented; in Section 3, the supporting framework is introduced; in Section 4, the architectural framework is validate; and Section 5 states the paper's conclusions.

2. Background and Related Work

This chapter introduces the three necessary pillars considered in the proposal of this paper, which are required for the deployment of extended Reality Functions in Digital Twin Networks. As stated in Section 1, the DTN + XRFs can be supported utilizing network virtualization techniques; then, the key pillars are Digital Twins (DTs), extended Reality (XR) and network virtualization (NV). In addition to the aforementioned technologies, complementary concepts such as space communications and tolerant networks are introduced. The rationale, justification and consistency of these technological pillars will be analyzed and discussed in Section 3. Therefore, the most relevant papers in the related fields of these pillars, as well as their connection with the paper proposal, will be introduced in this section.

2.1. Digital Twins

The Digital Twin (DT) concept was first introduced in 2003 at Michigan University by [1] and defined as a digital model or representation of a physical object, process or system; these digital models are constructed with real-time data obtained from the physical object as the source. The models emulate the behavior and characteristics of the physical objects, and thus, maintenance and updating are required to ensure model accuracy. Subsequently, synchronization and data exchange between the Digital Twin and the Physical Twin must be guaranteed. The deployment of DT models is supported by architectures such as: IoT, sensors, AI, data analytics, etc. DTs were conceived as enablers for smart industries achieving an improvement in workers' efficiency by means of three principal ideas: the improvement in human object visualization, simplified object comparisons and optimized collaboration. The utilization of DT models offers several advantages, including virtual representation, real-time monitoring, simulation and prediction, life-cycle management, inter connectivity between Digital Twins, and decision making based on real-time data.

A framework for the deployment of DT systems is presented in [2], which consists of a data layer, integration layer, service layer and middleware cognitive layer. Moreover, in [3] a framework is presented that integrates different technologies such as IoT, AI and cloud computing, among others, and highlights the possibility to support simulations to predict future behaviors; in addition, it is underscored that the DTs are not static in time: DT models are being updated. In this research paper, DT technology is employed to construct a Digital Twin Network (DTN) environment and is introduced next in Sections 3.2 and 3.3.

A relevant issue is the integration of DTs and data spaces, defined as decentralized ecosystems that manage their own data, as is addressed in [4] by employing several techniques such as APIs, AI techniques, feedback loops and data classification according to the data processing tier; such integration is fundamental to enabling a DT real-time data acquisition from the data space. On the other hand, [5] states that technologies like IoT, 5G networks and cloud computing, among others, have the potential to enrich data quality and volume, reduce the transmission latency and improve the data processing capabilities. In [6], it is stated that XR technology enables a 3-D DT representation, emphasizing the significance of security and privacy techniques to ensure data integrity and the identity of the real twin.

The DT technology has been exploited in the development of novel network applications, leading to the emergence of the Digital Twin Network (DTN) concept. The DTN has been employed in several researches. For instance, in [7], the DTN is utilized to simulate and analyze security provisioning in network slices and achieve a dynamic policy control; reference [8] presents an efficient network planing system based on real-time data which enables enhanced data management and visualization; in [9], the routing protocols are enhanced by means of real-time data and employing the DTN as a simulation environment for testing the enhanced routing protocols; and paper [10] underscores the importance of network models in DTN environments and describes two model types: Basic (network entities and topologies) and Functional (based on application requirements and scenarios).

With the objective of facilitating DTN management, the Digital Twin Manager (DTM) concept is introduced in [11]. The DTM framework consists of different modules including physical–digital twins' communication/synchronization, network configuration, a configuration requests buffer, simulation and feedback. In this paper, a different approach for deploying a DTM is presented in Section 3.2.

2.2. Extended Reality

The extended Reality (XR) term comprises a diverse set of technologies that facilitate the expansion of traditional reality information. Among these technologies are included: Virtual Reality (VR), defined as an interactive virtual environment; Augmented Reality (AR), which augments the physical world with virtual information; and Mixed Reality (MR), which combines both VR and AR technologies [12]. One of the pioneers of addressing the concept of Augmented Reality was [13] in 1992 and it was introduced in the context of airliner manufacturing envisioning that workers are able to access augmented digital information stored as CAD files in real time, enabling an immersive object visualization. The steps for deploying immersive applications are: sensing, environment tracking, digital content rendering, alignment and registration, visualization and interaction, and synchronization. By means of AR, the user's visual field is augmented with relevant, necessary and dynamically changing information to facilitate task comprehension. Virtual Reality was presented for the first time in 1965. VR is defined as a realistic, immersive and responsive experience that enables users to interact with a virtual environment involving vision, audition, smell or touch senses to give the perception of being in a real environment; this is achieved by means of headsets, gloves, glazes, etc. VR's primary objective is to provide realistic and immersive simulations and reduce the production times in industries by facilitating workers' collaboration and communication [14].

To deploy XR systems, in [15] is presented a framework to facilitate the physical–virtual environments' integration comprised of an agent controller, virtual environment layer, physical environment layer, user interface and data synchronization mechanism, while in [16], the importance is emphasized of ensuring the synchronization state of the virtual environment and communication among clients to achieve an improved collaborative experience. On the other hand, paper [17] proposes an architecture based on cloud infrastructures, tailored Software Development Kit (SDK) environments and a core engine which is in charge of data acquisition, management, algorithm deployment and interface enablement. A different approach that can be leveraged in virtual environments is presented in [18], which consists of Inverse Augmented Reality (IAR) that leverages agents in the virtual domain. In this research paper, the XR functionalities are intended to be deployed in a Digital Twin Network environment; the supporting framework is presented in Sections 3.2 and 3.3.

XR has the potential to enhance virtual environments. In [19], it is stated that XR can introduce a DT's interactive visualizations, immersive DT model control, collaborative model visualization and interaction, among others. In [20], the DT-XR resource-constrained scenario is introduced, which can be addressed with different technologies including: cloud/edge computing, data compression and resource allocation based on AI techniques. In [21], a framework which integrates DT with VR is presented and it is constructed in a cloud infrastructure, with sensor interfaces and a data analysis pipeline based on AI techniques. Reference [22] discusses the potential of integrating DT with XR for enhancing maintenance and fault detection processes with novel applications such as healthcare, education, urban planning, etc. Reference [23] analyzes the cyber-security enhancement; since the integration of digital and physical environments incurs blurred boundaries, leveraging DT and XR provides a unified reference framework. Reference [24] presents a framework composed of a DTN, XR service management and a resource allocation entity. References [25,26] present similar frameworks based on layers, with the XR and DT layers being the core of such frameworks. While the DT layer enables real-time monitoring, the XR layer is intended to manage the immersive services, based on NFV technology. In the present paper, in Section 3.2, an innovative framework that integrates the DT and XR technologies based on tailored modules to deploy a DTN + XRF is presented.

2.3. Network Virtualization

In 2009, a survey related to network virtualization (NV) was presented by [27]. The NV concept evolves from the Virtual Private Network (VPN) field representing an advancement in deploying multiple virtual networks over a common shared physical structure and maintaining isolation among each virtual instance. The advantages of NV include the ability of network operators to re-sell infrastructure to third parties, diversification of infrastructure for private purposes, network sharing and the provisioning of managed services. Network virtualization is emerging as a pivotal technology in the deployment of 5G and beyond networks due to its flexibility and scalability, programmability and interoperability. The most relevant technologies that facilitate an effective network virtualization process are as follows:

1. Software Defined Network (SDN): A paradigm introduced to decouple the control and data planes in network devices, thus enabling a centralized resource control and provisioning leveraging of an SDN controller. In addition, a SDN introduces dynamic management by enabling applications to directly interact with the hardware resources, while OpenFlow is the protocol to establish communication between the infrastructure elements and SDN controllers [28],
2. Network Function Virtualization (NFV): It addresses a different approach in which network functions are supported. Rather than relying on dedicated hardware, NFV leverages standard servers or cloud-based devices to provide these functions. NFV enables the deployment of efficient and flexible network infrastructures. Moreover, it allows the provision of network operators' dynamic on-demand services, thus enhancing the scalability of the network and optimizing resource utilization [28],
3. Network Slicing: This paradigm deals with the partitioning of a physical network into slices oriented for specific purposes (tailored services). The dynamic allocation of physical resources is achieved by means of cloud computing. The concept of network modularity (divide network functions into modular network functions) can be achieved by offering tailored services leveraged by vertical industries [29].

Reference [30] states that NV has a crucial role in avoiding network ossification introducing flexible, scalable and technology-agnostic networks and the decoupling of Infrastructure Providers (InPs) and Service Providers (SPs); then, NV enables the possibility to deploy a flexible and scalable Digital Twin Network approached with virtualization techniques. In order to optimize the virtualization process, Reference [31] introduces the Virtual Network Embedding (VNE) concept, which is the process of mapping virtual networks onto a substrate network by means of heuristic and meta-heuristic algorithms and optimization metrics' utilization. The efficient resource mapping algorithms face challenges like NP-hard complexity, heterogeneous environments, scalability and resource constraints, among others; an efficient embedding process is crucial to achieve a DTN's deployment in a constrained resources environment such as a space station.

Reference [32] addresses the Network Slicing as a Service (NSaaS) concept that can be leveraged to deploy tailored DTNs and improve resource utilization, thus enabling scalability to offer on-demand networks. On the other hand, Reference [33] introduces the virtual security network functions paradigm that can be implemented in Digital Twin Networks to facilitate security device management and deploy security functions empowered with XR technology. Moreover, Reference [34] proposes two paradigms for improving virtual networks that can be leveraged for DTN deployment to introduce a flexible and tailored network: holistic network virtualization in order to introduce network control granularity and end-user virtualization and user-personalized services, and secondly, Pervasive Network Intelligence in order to enable real-time decision making, predictive analysis and self-managed networks.

2.4. In-Space Communications and Tolerant Networks

To enable the original network (Earth) and Digital Twin Network (space station) communication, satellite communication systems can be adopted in order to establish links and redundancy. In [35], a flexible satellite communication system, based on NFV and SDN technologies, is addressed and presents advantages such as flexible service delivery, efficient traffic scheduling, enhanced data forwarding, etc. On the other hand, Ref. [36] leverages both the lens of Free-Space Optical Satellite Networks (FSONs) for achieving low latency and longer communication distances, and the combination of temporary and permanent links to improve connectivity. Reference [37] discusses the outage probability in optical satellite communications due to atmospheric conditions.

Along with satellite systems, the identification of an accurate transport protocol is essential to achieve ON-DTN communication. Reference [38] introduces an algorithm based on bandwidth and Round Trip Time (RTT) detection at regular intervals to minimize TCP window reduction; on the other hand, ref. [39] emphasizes the TCP failure in interplanetary communications due to high latency and intermittent connectivity due to disrupted links. Then, TCP is a risky approach as is proven in Section 3.1 by Equation (2). In [40], several space transport protocols are discussed. The Space Communication Protocol Specification–Transport Protocol (SCP-TP) is based on TCP with the improvement in window adjustment based on the RTT and asymmetric channel considerations. The TP–Planet protocol adaptively selects the channel rate and deals with intermittent communications. The Bundle Protocol for Delay/Disrupting Tolerant Networking (DTN) architecture introduces the custody transfer concept and leverages bundles for intermittent connections. The Saratoga protocol is based on UDP for IP packets, selective negative ACKs and optimistic data transfer. The Packet Interleaving File Delivery Protocol (PIFDP) is based on encoding/decoding packets assuring that the receiver is able to reconstruct the lossy packet. In addition, Ref. [41] underscores the Delay/Disrupting Tolerant Networking (DTN) architecture as a solution for future satellite networking scenarios and highlights the node's long-time storing capacity, the custody transfer, priority scheduling delivery and congestion controls based on selectively discarding bundles or custody non-acceptance.

Despite the accurate ON-DTN transport protocol identification, different techniques can be adopted with the aim to guarantee communication robustness and reliability. In lossy-network scenarios, work [42] proposes the utilization of queue mechanisms to deal with packet congestion, thus increasing delivery ratio, while in [43] is addressed lossy channel and high delay networks by modeling channels as finite First In First Out

(FIFO) queues to prevent operational failures. Regarding time-varying networks, work [44] presents an algorithm based on "Bottleneck Bandwidth and Round-trip propagation time" (BBR): which was deployed by Google, and enhanced with Deep Learning (DL) techniques to dynamically adjust the algorithm gain. On the other hand, in [45] is discussed the latency reduction by adaptively selecting the channel transport protocol based on the current channel conditions (Bw, latency, packet loss rate, etc.) and machine learning algorithms that consider historical decisions.

Reference [46] introduces an analytical method to estimate the delivery time in asymmetric space channels. The file delivery estimation is represented by Equation (1) where: $T_{File\text{-}delivery}$ is the total time for delivering a file, N_{Round} is the total number of attempts to transmit the file, T_{Bundle} is the time to transmit a bundle computed as $Bundle_{size}/Channel_{rate}$, T_{RTO} is the Re-transmission Time Out, T_{Prop} is the channel propagation time, $T_{Trans\text{-}last}$ is the time to transmit the last bundle and $T_{Disrupt}$ is the total disruption channel. N_{Round} follows a geometric distribution and takes into account the probability that a bundle is not delivered with success due to the BER or SNR, while $T_{Trans\text{-}last}$ is the average of the maximum bundle time. In the proposed ON-DTN scenario, delay metrics can be implemented based on Equation (1).

$$T_{delivery} = N_{Round}T_{Bundle} + (N_{Round} - 1)T_{RTO} + T_{Prop} + T_{Trans\text{-}last} + T_{Disrupt} \tag{1}$$

3. Architecture's Proposed Approach

In this section, the DTN + XRF-supporting architectural framework is introduced. First, the DTN topological architecture is analyzed, followed by the supporting architecture's definition; subsequently, the traffic flows between the architectural framework's components are presented with the objective of introducing a more comprehensive understanding of the system's operation.

3.1. DTN Topological Architecture

The DTN represents an ON (physical or virtual) instance. As outlined in Section 1, the ON replication and DTN deployment are hosted by the Digital Twin Manager (DTM) entity. In addition, the DTM is responsible for the DTN's self-management and the ON-DTN communication and synchronization. The synchronization is defined in this paper as an asymmetric process described as follows:

1. Synchronization from ON to DTN is mandatory: this ensures that any ON modification is reflected in the DTN.
2. Synchronization from DTN to ON is not required: since the DTN is designed to be enhanced with respect to the ON, any modification deployed in the DTN (topology, configuration, services, external connections, etc.) does not affect the ON.
3. The DTN can request resources to the ON via the DTM: for instance, when DTCDs request resources located in the ON, the DTM can request access to such resources and download them in the DTN.
4. Synchronization must be assured even in disadvantaged scenarios: QoS may be compromised in terms of latency, percentage of packet loss, throughput, reliability, etc. However, synchronization is an essential process and must be ensured even in disadvantaged conditions.

Regarding network management, the DTM has to guarantee the autonomy of the network management, enabling a "zero-touch" network. Figure 1 illustrates the DTN instance; notice that the DTN has been enhanced with extended Reality Services and an external network connection.

In addition to the enhancements provided by the XR services and external connections, this paper introduces the DTN's ability to modify the ON's topology within the DTN in order to improve the inner communication and the access to resources. Figure 2a depicts the ON's topology, while Figure 2b depicts the DTN's improved topology; notice that redundant links have been added, while “Files” resources have been relocated to “PC-2” within the DTN topology.

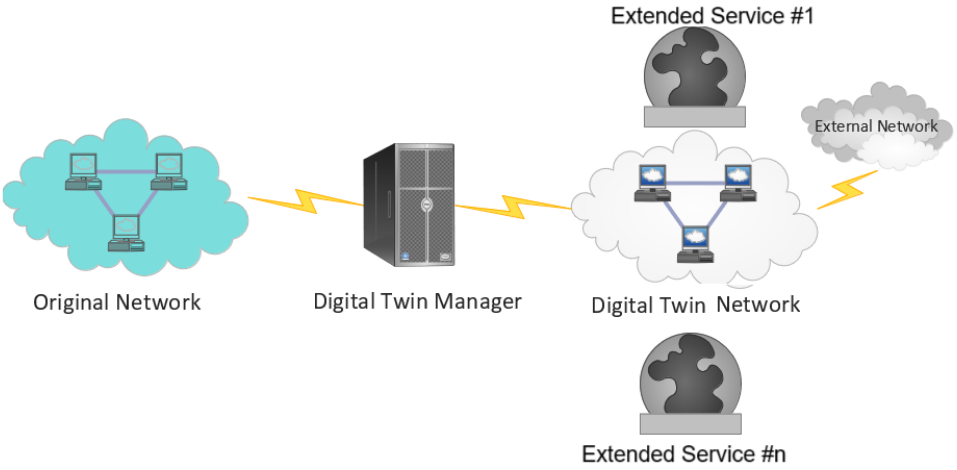

Figure 1. Proposed DTN + XRF topological architecture.

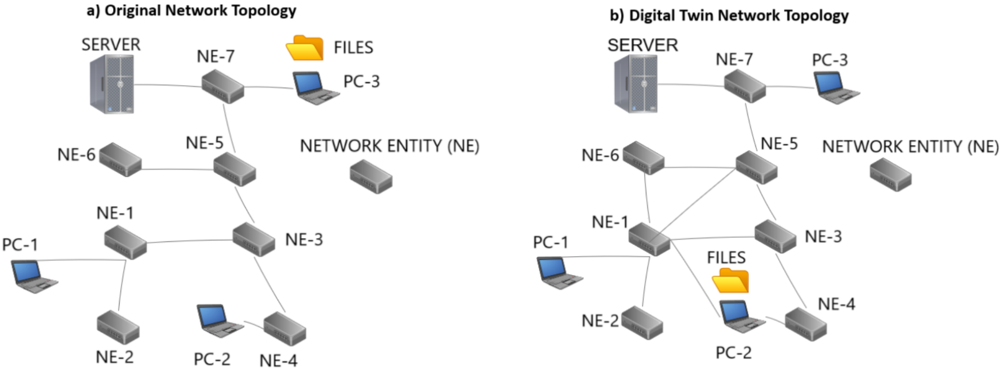

Figure 2. Example of topology improvement in the DTN.

The supporting architecture has to offload the expensive twinning and XR services’ provisioning computations from the DTCDs. The proposal presented in this paper exploits data centers’ computational capabilities. Figure 3 illustrates the perspective from the DTCDs to access data center services via a standardized API.

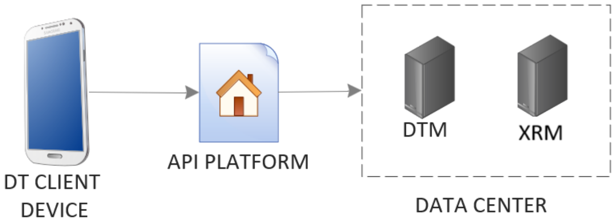

Figure 3. DTN + XRF access from Digital Twin Client Devices.

Basically, a server in the data center needs to host two of the main architectural entities: the DTM which is responsible for DTN building and maintenance, while the XRM is responsible for providing XR network services. The systems’ interaction standardization offered by API platforms can be leveraged to facilitate access to these modules. Similarly, on the ON domain, it must be deployed as a supporting entity defined as Original Network Manager (ONM) to facilitate ON-DTN communication and to handle the twinning process for the DTN’s creation/updating. The complete network’s topological architecture can be observed in Figure 4.

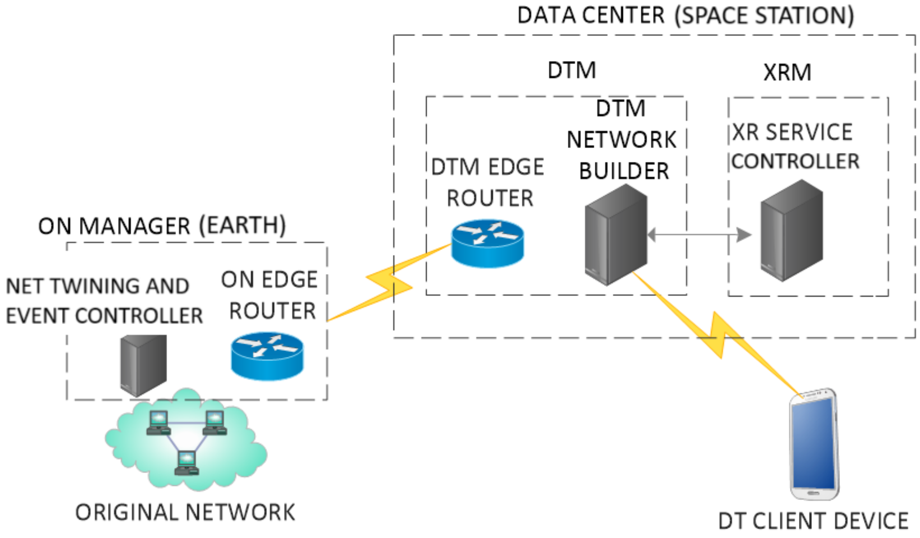

Figure 4. Global proposed DTN + XRF architecture.

Since this paper considers a space station hosting the data center, it is mandatory to consider the space communication scenario. In space communications, several communication constraints arise including latency, high packet loss and packet error percentages, limited bandwidth, link disruptions, jitter, interferences and noise, and synchronization issues. In addition, the *TCP* protocol is not suitable due to the *TCP* window increase, which can lead to significant QoS degradation. The *TCP* window (Mb) equation is introduced next, where *RTT* is the Round Trip Time (seconds), while *Bw* is bandwidth (Mbps).

$$Bw = \frac{WINDOW_{TCP}}{RTT} \tag{2}$$

In the aforementioned scenario, it is necessary to consider the *RTT*, which can range from tens of seconds to hundreds of seconds. Consequently, according to Equation (2), the *TCP* utilization will result in a notable bandwidth degradation. Then, it is fundamental to mitigate the cited communication constraints. To address the constraint mitigation, a variety of techniques can be leveraged including buffers and queues implementations, a *TCP* architecture modification or even the utilization of space-tolerant protocols like those analyzed in Section 2.4. A potential solution is the the Delay/Disrupting Tolerant Networking (DTN) with the Bundle Protocol (BP) and custody transfer technique. It has to be emphasized that for a successful system deployment, the space communication issues need to be mitigated with the aim to deploy a reliable system; this analysis will be deepened in future works.

3.2. System-Supporting Architecture

Then, with the objective of achieving the DTN + XRF deployment, the supporting architectural framework is defined. The framework is supported by three architectural entities (AEs). The AEs are comprised of modules tailored to specific functions. The AEs are present in both the ON and the DTN domains. The Original Network Manager (ONM) provides support for the ON domain, while the Digital Twin Manager (DTM) and the Extended Reality Manager (XRM) provide support for the DTN domain. Figure 5 shows the three defined AEs, while Figures 6–8 illustrate the architecture defined for each AE.

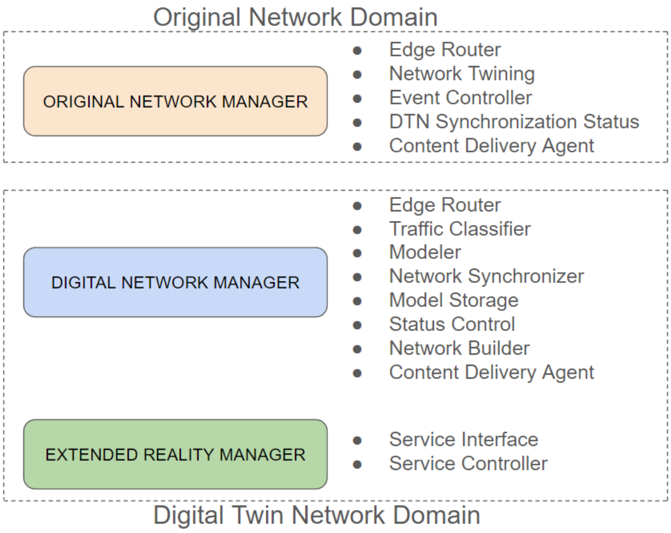

Figure 5. Architectural supporting entities.

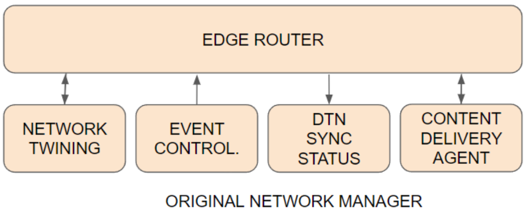

Figure 6. Original network manager modules.

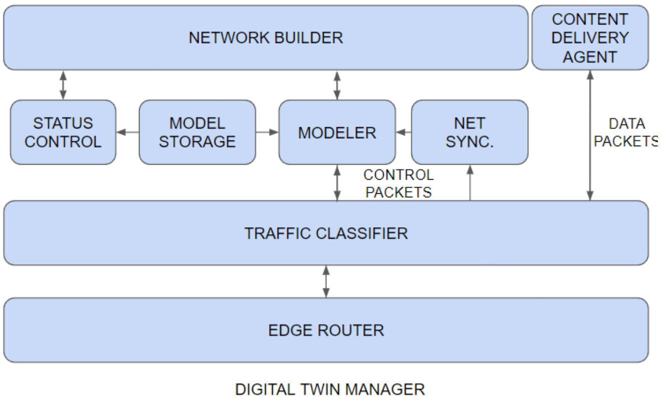

Figure 7. Digital Twin Manager modules.

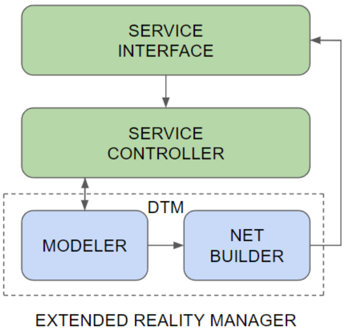

Figure 8. Extended Reality Manager modules.

3.2.1. Original Network Manager (ONM)

The ONM is the entity responsible for acquiring the ON information and forwarding it towards the Digital Twin Manager with the aim of creating/updating the DTN. The modules that compose the ONM are described next:

- Edge Router: This is designed to serve as a communication interface, enabling the transmission of ON information to the DTM and the reception of DTN status information from the DTM. It is fundamental that the Edge Router is designed with the intelligent ability to determine the optimal transport channel to guarantee a successful packet delivery to the DTM.
- Network Twinning: This is responsible for receiving a request generated in the DTM for the collection of complete network information (network entities, topologies and configurations) from the ON. The collected data are forwarded to the Edge Router for the space transmission process.
- Event Controller: This is in charge of the continuous ON network status monitoring. In the event of network modifications, the Event Controller transmits the "event" data (information regarding the ON modification) to the Edge Router for the DTN synchronization process.
- DTN Synchronization Status: This is responsible for receiving the current DTN status in order to keep the ONM aware of the DTN synchronization state. In addition to synchronization status, this module is capable of receiving information from the DTN related to extended functions with the aim to provide the extended service to the ON.
- Content Delivery Agent: This has the function of forwarding incoming DTN data packets to the ON device and vice versa.

Regarding the Network Twinning and Event Controller modules, both need to use different mechanisms to compile the ON data, i.e., sketches or templates. In addition, such modules need a network model (adjacency matrix, knowledge graph, etc); while it is not mandatory that the network model owned by these ONM modules is the same as the network model owned by the corresponding DTM modules, both network models must be compatible. Both aforementioned modules must permanently maintain the ON configuration and topology information as network models; the Network Twinning is involved in the DTN deployment phase, while the Event Controller transmits information when an update is performed in the ON and it must be updated in the DTN. Finally, it is important to highlight that the Network Twinning, Event Controller and DTN Sync.Status deal with control packets, while the Content Delivery Agent deals with data packets.

3.2.2. Digital Twin Manager (DTM)

The DTM is the entity that controls and manages the DTN and receives the network information from the ONM. The DTM modules are presented as follows:

- Edge Router: This is designed to serve as a communication interface, enabling the transmission of DTN status information to the ONM and the reception of ON information from the ONM. It is fundamental that the Edge Router is designed with the intelligent ability to determine the optimal transport protocol according to channel conditions such as disruption time and available bandwidth, among others, to guarantee a successful packet delivery to the ONM.
- Traffic Classifier: This is responsible for classifying the incoming data packets from the DTM Edge Router into control packets, which are utilized for DTN creation or synchronization, and data packets which are destined for DTCDs. The data packets are directly forwarded towards the Content Delivery Agent, while the control packets are forwarded either directly to the Modeler (to create a new DTN) or to the Network Synchronizer (to update the DTN model).
- Network Synchronizer: This has the function of receiving ON "Event" data and forwarding it to the DTM Modeler. This Synchronizer has to be designed with the intelligence to manage and process the "event" information. With the objective to

offload the Modeler processing capabilities, the Synchronizer must be supplied with a set of algorithms aimed at: Data Filtering to optimize the data processing and ensure that the relevant data are posteriorly sent to the Modeler (these algorithms can be based on Kalman Filter algorithms); Updating-Priorities Establishment with the aim to carry out an orchestrated synchronization based on the event's relevance (Neural Networks can be implemented to analyze the priority based on previous synchronization events); Updating-Dependencies establishment to guarantee the network statuses' consistency (dependency graphs can be envisioned as a solution); Synchronization Algorithms, which can be based on network model difference detection algorithms such as delta or Rsync. The aforementioned algorithms can be analyzed in future work. Since the proposed scenario involves space communication between the ONM and DTM, synchronization can be compromised due to the constraints perceived in space scenarios; mitigation by means of buffer/queues mechanisms or different transport protocols, as outlined in Section 3.1, is crucial to alleviate synchronization issues.

- Modeler: The Modeler is in charge of selecting from several network models (adjacency matrix, knowledge graph, among others) that are capable of managing the appropriate network model, ensuring the compatibility with the network model employed by the ONM and considering the modeling technique according to the algorithms employed or the accessibility to the stored model. Once the Modeler has processed the DTM model, the model has to be stored in the Storage Module. In addition, the Modeler has the capability to update the DTN model, stored in the storage module, with the "event" information received by the Network Synchronizer. Moreover, when the Modeler receives an "Extended Network Model" along with a service request from the XRM Service Controller, the Modeler has the task of determining whether the current DTN model has to be replaced by the "Extended Network Model" (writing services) or the "Extended Network Model" has to be stored and tagged with an XR Service ID (reading services) for a later XR Service request.
- Model Storage: This is responsible for storing the processed network model by the Modeler. Additionally, this module must enable access to stored models for the Modeler; thus, queuing techniques are needed.
- Network Builder: This is responsible for the DTN deployment by the consequence of receiving the modeled network information. Furthermore, it is required to continuously transmit "Network Status" beacons to the Status Control to achieve an autonomous DTN management process. When it receives a "Network Warning" from the Status Control, the Builder needs to request to the Modeler to model the most recent stored network model with the aim of re-generating the DTN. This paper introduces to the Builder the capability to receive super users' requests intended to modify the current DTN model; subsequently, the request is forwarded to the modeler for the model updating process.
- Status Control: This is in charge of receiving the "Network Status" beacons from the Builder with the objective of identifying any anomalies and failures in the DTN based on the stored network model. In the case that any anomaly or failure is detected, the Status Control transmits a "warning" signal to the Builder requesting that the DTN must be re-modeled.
- Content Delivery Agent: This has the function to receive data packets from the ON and determine whether the packets have to be forwarded to a DTCD or an external proxy server with the objective of storing ON content in a Content Delivery Network fashion. Additionally, it receives data packets from DTCDs to be delivered to the ON.

Regarding the DTM operations, it is crucial to emphasize the necessity of implementing failure recovery techniques into the modules. These techniques can ensure that operations are not compromised. AI techniques or multi-core solutions, among others represent effective solutions for achieving self-healing mechanisms. Nevertheless, in the event of a module failure before the self-healing is completed, processes such as ON-DTN communication and synchronization may be compromised; however, the DTN has been deployed and inner DTN communication can operate uninterrupted. The DTN can be self-healed by means of the monitoring process described in Section 3.3.

3.2.3. Extended Reality Manager (XRM)

Finally, the XRM is the entity that manages the interaction and building of XR services. The XRM modules are described next:

- Service Controller: The controller is responsible for the XR service deployment (including the extended data integration and service building). This module can be approached with several techniques. Network Function Virtualization (NFV) is a suitable technology for deploying the different XR services; then, technologies such as OpenMANO, OpenNFV, OpenStack or even proprietary solutions can be adopted. This results in it being necessary to additionally design this module with AT techniques in order to determine, according to the user's request, if the requested service is one of a reading or writing nature. A reading service just consults the network model information; for instance, visualization services correspond to those of a reading nature. On the other hand, services that modify the network model, such as resource re-allocation, smart routing tables, etc., correspond to writing services. The objective of classifying the services aims to forward the appropriate request to the DTM Modeler along with the "extended Network Model". This block needs to interact directly with the DTM Modeler in two iterations: in the first iteration, the XRM Service Controller requests from the DTM Modeler the network model information (stored in the DTM Model Storage), and once the network model information is retrieved, the XRM Service Controller has the responsibility of processing the network data and integrating them with extended data; on the other hand, for the second iteration, the XRM Service Controller forwards the "Extended Network Model" along with the request nature (reading or writing) to the DTM Modeler. Moreover, this module has to manage the XR service with the objective of guaranteeing an efficient service interaction. In order to manage the XR service, different criteria can be followed; for instance, service nature (reading or writing), user role priority (super users or network administrators have more priority than clients), latency and response time (XR services that require low latency and response time should have a higher priority), resource availability (XR services must be deployed according to the current resources utilization), etc.
- Service Interface: This has the function of serving as an input/output user interface. The Service Interface can be realized as a set of interfaces including: Human–Machine Interfaces (HMIs), glazes, touch sensors and voice commands, among others. Subsequently, this block has to process the input/output information. The service interface receives the network information for the corresponding user visualization from the DTM Network Builder. Moreover, this interface has to be designed with a standardized platform to receive XR service creation requests, and must be flexible in handling any type of information exchanged, as an information-agnostic interface.

3.3. DTM Traffic Flows

Once the supporting architectural entities, modules and functions have been defined, with the objective to clarify the insights about the internal and inter-entity interactions, different traffic flows are introduced. The traffic flows are divided into two different planes: the control plane and the data plane. It is important to underscore that between the DTM Edge Router and the ON Edge Router, a space communication channel must be established; the use of TCP as the transport protocol is not feasible due to the poor bandwidth according to Equation (2). It is crucial to highlight that in this research, a redundancy ACK mechanism comprising a double ACK (ACK + confirmation ACK) is introduced to guarantee that the DTM and ONM entities are fully aware of their partner entity status; basically, the confirmation ACK guarantees that the requested or sent information has been received, as well as the custody ACK (the initial sent ACK). In addition, the custody ACK and the confirmation ACK will not block future transmissions with the aim to optimize communication between the ON and the DTN.

1. Control Plane:
   The control plane traffic flows are presented next. Such flows describe the DTN control operation including: creation, synchronization, monitoring, re-configuration and status sharing.
   (a) DTN creation: The process for creating a new DTN once the space station has been established and the DTM and XRM entities have been deployed is initiated by the DTM Modeler, which lacks network models at this initial phase. The modeler generates a "Request" message, which is relayed to the Traffic Classifier and then to the Edge Router. Upon receipt of the request by the ONM Edge Router, a custody ACK is sent to the DTM Edge Router. The request is forwarded to the Network Twinning, which processes it and determines that the ON network model has to be transmitted to the DTM (the processing time is referred to as Twinning Time). The network model travels back to the DTM Edge Router. The network model flows through the Traffic Classifier and is received by the Modeler. The Modeler processes the ON network model and subsequently transmits a "DTN request" to the Builder, which is responsible for the DTN building for the first time. Figure 9 illustrates the aforementioned traffic flow.
   (b) DTN synchronization: The synchronization process is initiated by the ONM Event Control, which has the task of periodically monitoring the ON in order to detect any change in the ON; upon identifying a network change, the module generates an "Event" message, which contains the specific ON modification. The "Event" is sent back to the DTM Edge Router (the double ACK mechanism is established between the Edge Routers) and is relayed to the Traffic Classifier, and then to the Network Synchronizer, which processes the ON modification and forwards the "event" to the Modeler. The modeler is responsible for retrieving the stored network model and subsequently updating it according to the "event" information (this process time is referred to as Re-modeling Time). Finally, the Modeler generates a "Net Regeneration" request that is sent to the Builder. This traffic flow is shown in Figure 10. Note that this paper differs from other Digital Twin Network oriented papers: the synchronization is not requested by the Digital Twin but by the Original Network.
   (c) DTN monitoring: The monitoring process is essential to ensure autonomous management and DTN self-healing. The DTM Builder has to periodically report its own statistics (packet error rate, throughput, etc.) to the Status Control module. The Status Control module has the capability to access the stored models enabling the detection of any anomaly or failure; in the event of anomaly or failure detection, the Status Control raises a "Net Warning" alert, which is sent back to the Builder. When the Builder receives a warning, it needs to request to the Modeler a re-modeling process as described previously. Figure 11 demonstrates the aforementioned flow.
   (d) DTN re-configuration request: This process is performed with the objective to modify the DTN configuration as explained in Section 3.1. The process requires that the DTM Builder operates as a DTCD super users' interface. The Builder receives the request and then the previously described re-modeling and regeneration operations are initiated (Figure 12).
   (e) DTN synchronization Status Sharing: This process is designed to ensure that the ONM is aware of the DTN network model status. The DTM Modeler generates a "Sync.Status", which contains the DTM network model information, and is sent through the Classifier and Edge Routers (with the double ACK mechanism), finally reaching the ONM DTN Synchronization Status. Figure 13 illustrates the traffic flow.

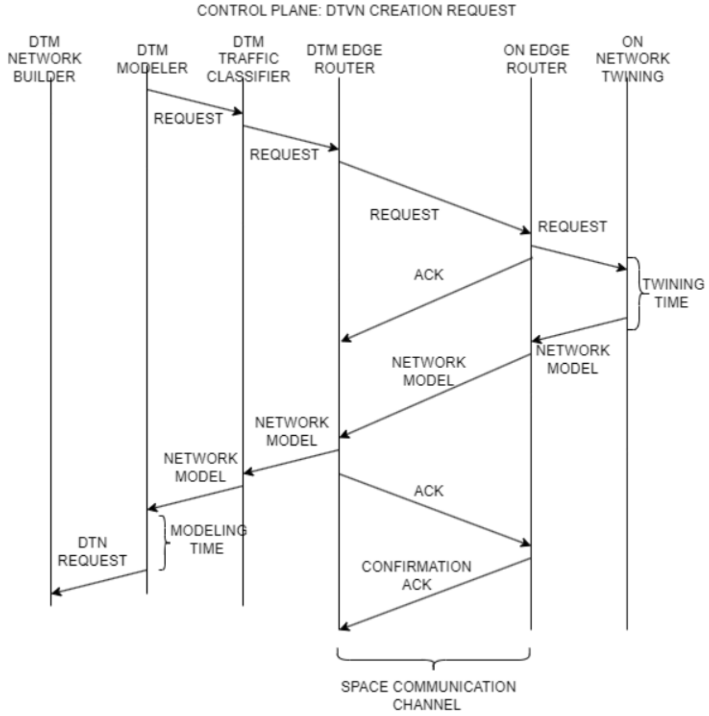

Figure 9. DTN creation flows.

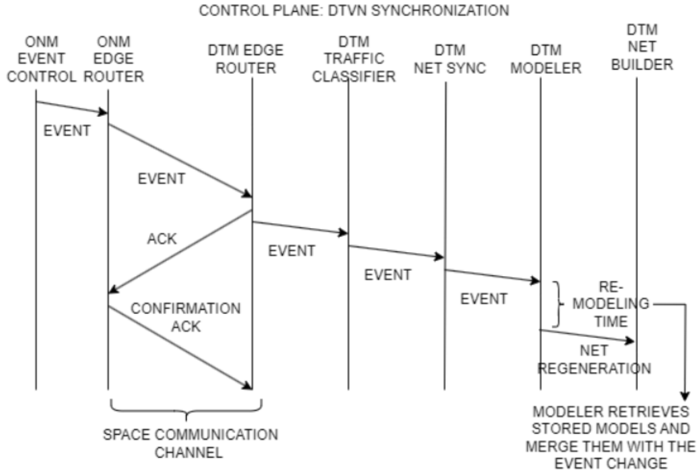

Figure 10. DTN synchronization flows.

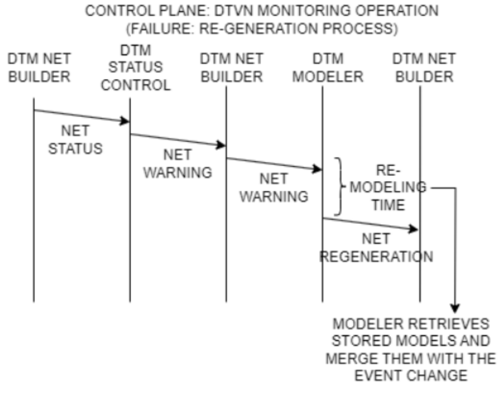
CONTROL PLANE: DTVN MONITORING OPERATION
(FAILURE: RE-GENERATION PROCESS)
DTM NET BUILDER
DTM STATUS CONTROL
DTM NET BUILDER
DTM MODELER
DTM NET BULDER
NET STATUS
NET WARNING
NET WARNING
RE-MODELING TIME
NET REGENERATION
MODELER RETRIEVES STORED MODELS AND MERGE THEM WITH THE EVENT CHANGE

Figure 11. DTN monitoring flows.

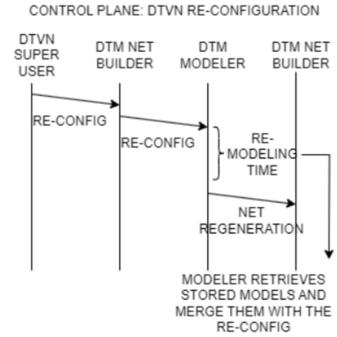

Figure 12. DTN re-configuration flows.

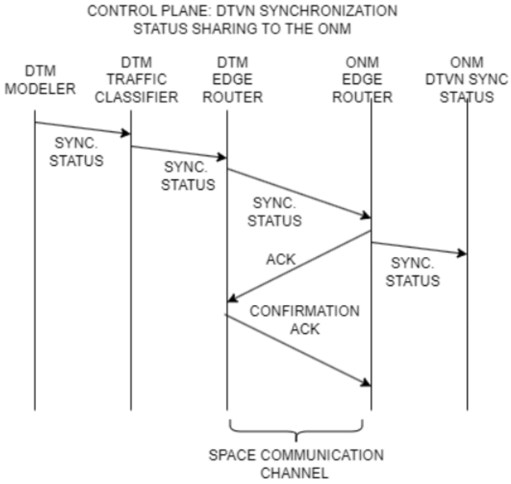

Figure 13. DTN synchronization status sharing.

2. Data Plane:
   The data plane traffic flows describe the data flow transmitted within the DTN among the DTCDs along with the data shared between DTCDs and ON devices. Three scenarios are introduced next:
   (a) Scenario 1: Two clients within the DTN exchange information: This is the simplest scenario. The DTN communication delay will be the typical delay from any terrestrial network. Then, the communication transport protocol can be approached with TCP or different streaming protocols.
   (b) Scenario 2: A client within the DTN requests content to the ON: In this scenario, the DT client is requesting contents that are originally located in the ON. This scenario introduces the high latency issue described in previous sections. Since a continuous content downloading involves the high latency problem, this paper proposes leveraging proxy servers placed in the space station; these proxy servers can monitor the channel utilization and request to the DTM to download ON content. Then, two sub-scenarios emerge:
      i. The content is already stored in the proxy server: the DTM requests the content directly to the proxy server; data flow for this request can be approached with the TCP protocol.
      ii. The content must be downloaded to the proxy server: the DTM must request to the ONM to transmit the content and then store it in the proxy server. Inevitably, the content transmission will suffer from the space communication delay.

      In both previous cases, the DTM Content Delivery Agent is responsible for determining whether or not to request content download to the ONM. Furthermore, the DTM Status Control may have the ability to asses the DTN's utilization and requests for content downloading in instances where the DTN operates at low utilization. The decision-making process can be leveraged with AI techniques.

(c) Scenario 3: A client within the DTN is communicating with a client within the ON: In this third scenario, the communication between the two nodes will suffer high latency from space communication conditions. The DTM Edge Router must determine that the TCP protocol is unsuitable to support the communication; consequently, the data flow from DTM to ON must be approached with a different transmission protocol; Delay-and Disrupting-Tolerant Networking can be considered as a potential solution. The traffic flows for this scenario are illustrated next in Figure 14.

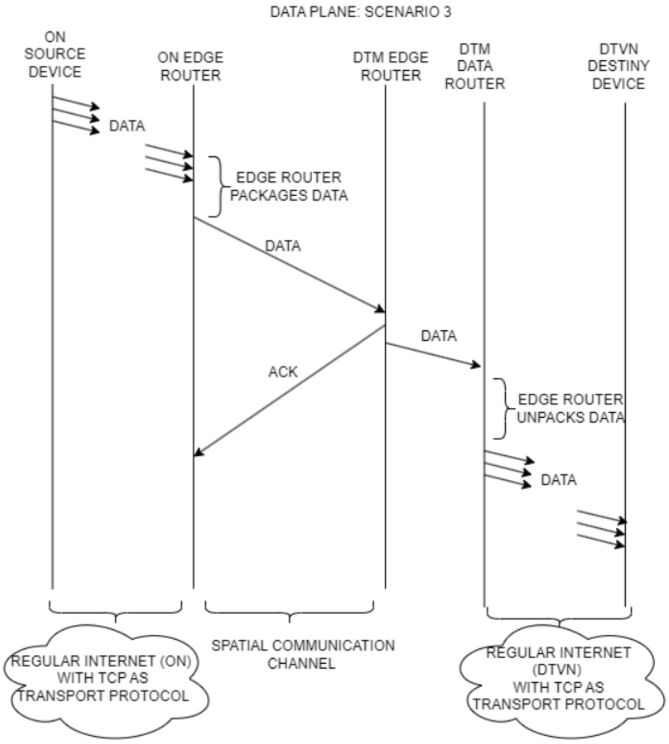
DATA PLANE: SCENARIO 3
ON SOURCE DEVICE
ON EDGE ROUTER
DTM EDGE ROUTER
DTM DATA ROUTER
DTVN DESTINY DEVICE
DATA
EDGE ROUTER PACKAGES DATA
DATA
DATA
ACK
EDGE ROUTER UNPACKS DATA
DATA
REGULAR INTERNET (ON) WITH TCP AS TRANSPORT PROTOCOL
SPATIAL COMMUNICATION CHANNEL
REGULAR INTERNET (DTVN) WITH TCP AS TRANSPORT PROTOCOL

Figure 14. DTN data flows.

3.4. Extended Reality Functions' Deployment Approach and DTN Operation—Integration

Since this paper's primary focus is the enhancement of the network experience assisted by XR technology, this subsection introduces the deployment of the XR services and their integration with the DTN. In addition, the traffic flows regarding the service operation are illustrated.

First, it is important to emphasize that the set of different extended Reality network services composes the XR Functions (XRF) system, which enhances the proposed DTN. In order to facilitate the construction of new network services based on XR technology, a variety of approaches can be adopted; as previously stated in Section 3.2, the utilization of NFV as a development tool is recommended. According to [47], the NFV architecture offers a wide range of advantages that can be exploited within the DTN environment to facilitate the XR services' deployment. Among the advantages of NFV architecture are: scalability and flexibility, efficient resource utilization, dynamic service provisioning based on Virtual Network Functions (VNFs), ease of integration with different technologies and digital environments and standardized development platforms (OpenNFV, OpenMANO, OpenStack, Kubernetes, etc.), among others. While the XRM Service Interface can serve as platform to receive the XR creation request, to construct a new XR service within the DTN, the XRM Service Controller has to consider:

- Network model access: The manner in which the XRM Service Controller accesses the DTN network models has to be defined. APIs can be employed to facilitate the retrieval of the DTN network models (stored in the DTM Model Storage block).
- Identify the requested XR network service: in this step, the requested network service has to be analyzed and decomposed into smaller portions of the network function.
- VNF's deployment: once the network functions that comprise the XR requested service have been identified, the VNFs can be constructed. The approach to employ smaller VNFs has the advantage of function modularity. The deployment of the VNFs has to consider the DTN network model information.
- Construct the XR service: the XRM service controller has the task of constructing the XR service based on the NFV architecture.
- Extended Network Model definition: once the XR service has been constructed, it has to be modeled by the DTM Modeler and stored in the DTM Model Storage identified with a unique tag ID for future XR service access.

In the context of the integration of the Digital Twin Network (DTN) with the extended Reality (XR) services, key aspects must be considered: real-time data access, data processing, XR interfaces and simulations, as is explained in [48,49]. The extended Reality Manager (XRM) shown in Figure 8 and explained in Section 3.2 has the ability to achieve the aforementioned integration:

- Real-time data access: The DTN real-time data are realized as network models processed and updated by the DTM Modeler and stored in the DTM storage block. Consequently, the XRM Service Controller needs first to classify the service as a reading or writing service, and then access the stored network models through the DTM Modeler.
- Data Processing: Data processing for XR integration is supported by two blocks: the XRM Service controller and the DTM Modeler. The former has the responsibility of processing the retrieved network model and integrating it with extended data, while the latter has to re-model the "Extended Network Model".
- XR Interfaces: The XR interfaces are approached with the XRM Service Interface. As stated in Section 3.2, this block has to process the input and output information according to the input/output format (voice, images, touches, etc.); consequently, it has to process the DTM Network Builder information in order to allow the users to know and realize the service's response.
- Feedback: The integration of XR monitoring services can be realized as immersive feedback mechanisms to enable users to visualize real-time network data and achieve a real-time comprehension of configuration changes in the DTN.

The XR service creation process is depicted in Figure 15. The XRM Service Interface receives the creation request regarding a new XR service named as service "A" and forwards the request to the XRM Service Controller, which has to identify the smaller network functions that comprise the requested service. The Service Builder has to access the DTN model to build the XR service based on the NFV architecture. Once the service is constructed, the DTM modeler has to process the service and store the "Extended Network Model" identified with a unique XR ID: "Extended Model A".

Since the XR services may exhibit the DTN inner and sensible information, it is essential to consider the necessity of ensuring a secure environment for DTN clients, enabling them to utilize the XR services in a safe manner. Consequently, robust security implementations have to be considered, as is addressed in [50]. In order to guarantee data integrity, the following security techniques should be considered:

- Data Encryption: the encryption algorithms can ensure that the data transmitted between the DTM and XRM, as well as the data transmitted between the ONM and DTM, and the data that are shared within the DTN are protected.

- Authentication APIs: as was illustrated in Section 3.2, APIs can be adopted as an approach to address the user's authentication to XR services, as well as DTN resources.
- Bio-metric Authentication: the XRM module can be utilized to build an authentication service based on bio-metric signals and then ensuring the user's authenticity.
- Secure Protocols: secure protocols such as TLS/SSL can be adopted to guarantee the user's data integrity.

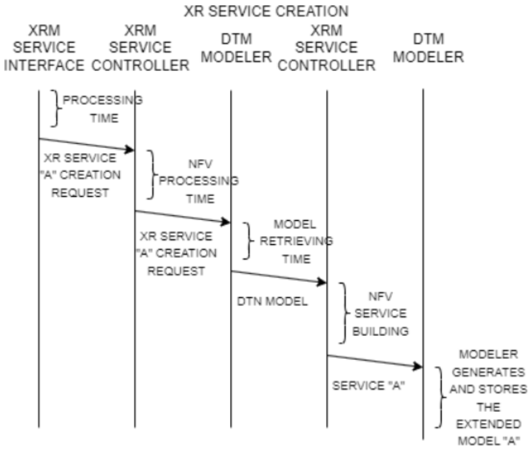

Figure 15. General creation of any Extended Reality Service.

Moreover, a super-user XR Service can be deployed that requests the creation of DTN instances from the main DTN, which is dedicated to providing networking services to users of the space station. The objective of creating multiple DTN instances is to establish a testing and training environment for the XR functions that have been deployed. This enables the analysis of the impact that an XR service may incur on the DTN, facilitating the testing and training of XR services or the deployment of new XR services in a controlled environment. With this approach, the risk of degrading or damaging the main DTN is mitigated. To achieve this, the DTM Builder has to be designed with the ability of managing multiple DTN instances.

Finally, the operation of the XR service "A" is explained. Two possible XR service natures are considered, as referred to in Section 3.2: reading and writing services.

- Reading XR Service: The reading services retrieve the "Extended Network Model" for the visualization of DTN information. The XRM Service Interface receives the request to utilize the previously created "XR Service A". This request is forwarded to the XRM Service Controller which processes the request and sends it to the DTM Modeler. The Modeler retrieves the stored "Extended Network Model A" and forwards it to the DTM Network Builder to retrieve the requested XR service information. Finally, the information is transmitted to the XRM Service Interface, which has the responsibility of processing the out-coming information. Figure 16 depicts the aforementioned process.
- Writing XR Service: The writing services modify the current DTN model by the "Extended Network Model". The initial operation is similar to the reading services' scenario. Once the DTM Modeler has retrieved the "Extended Network Model A", this model replaces the current operating DTN model and a "Network Re-generation" request is transmitted from the DTM Modeler to the DTM Network Builder. Finally, the XRM Service Interface processes the output information. Figure 17 illustrates the writing XR service process.

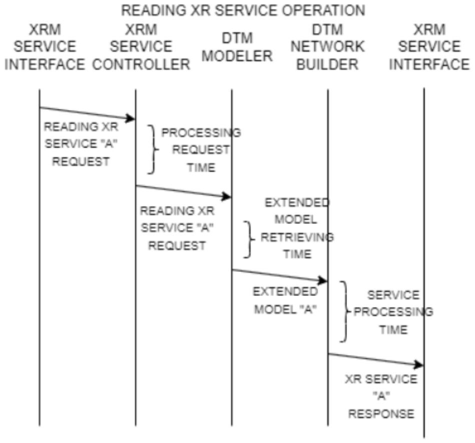

Figure 16. Reading operation of an XR service.

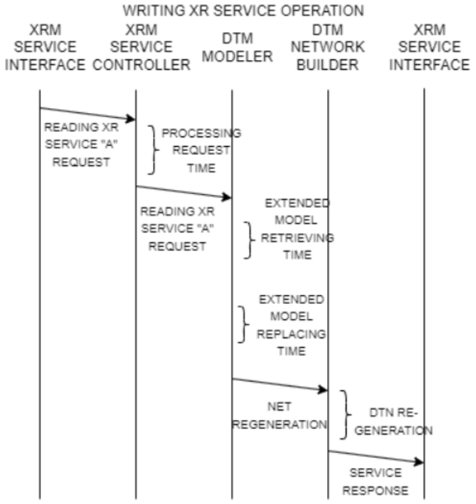

Figure 17. Writing operation of an XR service.

4. Architecture Modeling Approach

This section is dedicated to demonstrating that the proposed DTN + XRF is realistic, achievable, consistent and is expected to exhibit an acceptable level of performance.

4.1. System Metrics

In this section, relevant system metrics identified to assess the reliability of the proposed system will be introduced.

1. DTN resilience and Failure Protection: This refers to the metrics employed to measure the Failure Protection capability. This measurement will be approached with commonly used metrics including Mean Time to Recovery (MTTR), which is the average time taken by networks to recover from failures, and the Mean Time Between Failures (MTBF), which is the average time between successive network failures; both metrics are measured in seconds. Furthermore, it must be noted that in the ON-DTN communication disruption, the DTN remains operational. With regard to failure protection, the monitoring process is essential to heal the DTN's operation.
2. Response Time: This refers to the response time between Edge Routers after receipt of a request and it is measured in seconds. The response time is dependent on the channel conditions (bandwidth, disruption time, etc.), and the Edge Routers' channel selection capability. Response time is defined by Equation (3) where T_{edges} is the processing time at the Edge Routers, T_{prop} is the propagation time and T_{tx} is the

transmission time. Note that T_{edges} and T_{tx} may vary according to the type of request and the traffic involved, while T_{tx} is dependent on the Edge Routers' distance.

$$T_{response}(s) = T_{edges} + T_{prop} + T_{tx} \tag{3}$$

3. DTN Synchronization Time: This refers to the necessary time to achieve model synchronization between the ON and the DTN and it is measured in seconds. Synchronization Time is dependent on the response time metric and the complexity of the network models employed. The synchronization time is defined by Equation (4), where $T_{response}$ is the response time and T_{model} is the time to process the network model according to the model's complexity.

$$T_{SYNC}(s) = T_{response} + T_{model} \tag{4}$$

4. Zero Touch Capability: This defines the level of zero touch capability; in [51], several metrics are employed including automation percentage, operational cost, Quality of Service (QoS) and so forth. The advantage of leveraging the DTM's capabilities is that it introduces a high autonomy tier in the DTN; to guarantee the autonomous DTN operation, each DTM sub-module must be provided with enough intelligence. In the case of network re-configurations in the DTN, a DT super user is necessary to introduce those modifications; however, the re-configuration and re-building process will be autonomous. In a similar fashion, the XR Service management is autonomous by means of the XRM entity,
5. DTN scalability: This qualitative metric allows us to realize how the DTN can be scaled up or down and enhanced. The XRM entity enables the DTN's enhancement with novel and immersive functionalities; those DTN-enhanced capabilities are initially not present in the ON. The DTN can be easily scaled up or down topologically by means of the re-configuration process to modify the DTN network model. Similarly, with the re-configuration DTN process, external gateways can be added to the DTN network model with the purpose to provide the deployed DTN connection to external networks
6. System Complexity: This refers to the number of operations that are necessary to execute in the supporting entities to achieve an optimal system operation. In addition, with regard to the entities visualized in Figures 6–8, it can be appreciated that the supporting elements are relatively simple: such entities do not require a large amount of modules, while the modules' behavior and objective are clearly established as well as modules' interactions.
7. DTN portability and Deployment Efficiency: This is a qualitative metric to realize the ease of the DTN's deployment; the DTM's behavior provides a high level of ease of deployment. In addition, the DTN can be deployed in any terrestrial or extra-terrestrial station; then, the DTN can be envisioned as a portable network.

4.2. Communication Protocol Requirements

It is essential to analyze the requirements for enabling ON-DTN communication, as well as end device communications. Two different interface communications have been identified and are shown in Figure 18:

1. ONM Edge Router—DTN Edge Router: The communication between both Edge Routers concerns the in-space data transmission scenarios. In order to establish the in-space communication channel between both Edge Routers, several considerations have to be taken into account; among the main issues to be considered are:
    - Bandwidth: since the available bandwidth is a limited resource, the amount of data to be sent has to be optimized; then, compression algorithms can be utilized.
    - Latency and Disruptions: In the space communication scenario that has been proposed, the latency will be in the range of tens of seconds to hundreds of seconds; also, the communication links will suffer from disruptions due to different

conditions such as planetary orbits. Then, it results as being fundamental to mitigate the communication degrade in high delay and disrupting conditions with the aim to achieve the proposed system deployment. Alternatives such as the Delay/Disrupting Tolerant Networking with the Bundle Protocol (BP) can be envisioned as potential solutions.

- Security: Since the information shared between both Edge Routers can exhibit critical network information, the data integrity must be guaranteed and avoid unauthorized data access. Then, coding and encryption techniques can be adopted.
- Interoperability: it must be assured that both Edge Routers employ compatible standards, technologies, algorithms and protocols.

In this scenario, since the communication between the ONM and DTM will suffer from high latency, the utilization of security and authentication mechanisms can incur an increase in jitter and delay in the aforementioned communication; however, such fluctuations are expected to be insignificant compared with the space communication latency, which ranges from tens of seconds to hundreds of seconds.

2. Data Center—DTCD: Since both the clients and the Data center are hosted in the space station, the issues to be considered are different from the previous scenario. Among the main considerations are:
   - Environmental conditions: in the space station, space radiation could degrade the communication between clients and the data center.
   - Bandwidth: since the clients can request XR services, the bandwidth must be guaranteed to support those requests.
   - Accessibility and Security: In order to guarantee authorized accesses, API interfaces can be adopted. On the other hand, encryption techniques can be utilized to guarantee data integrity, as well as the TLS protocol.
   - Communication Protocols: The space station communication established between DTCD's and the data center will not suffer the high delay and disruption conditions; however, the utilization of XR services and DTN navigation require small latencies. Then, UDP or streaming protocols can be implemented.

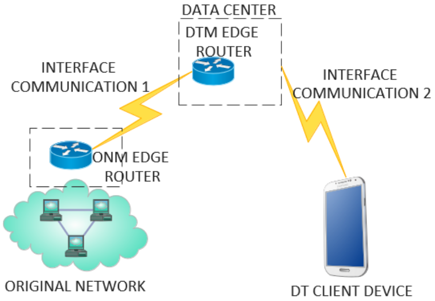

Figure 18. Proposed system communication interfaces.

In both cases, the three-way handshake mechanism must be considered. The aforementioned mechanism enable agents to negotiate communication parameters; the streaming protocols can leverage the handshake approach.

4.3. System Requirements Approach

As is shown in Figure 4, the proposed system is comprised of architectural entities, hosted by a space data center, with modules designed for fulfilling specific processes. To ensure an optimal system operation, the space data center requirements have to be considered.

Reference [52] analyzes the deployment of space data centers while ensuring performance computing, storage capacity and networking services, aiming to be integrated with terrestrial cloud platforms. The paper's architecture is composed of three layers: a physical layer which includes the hardware and embedded software, and it is intended to operate in an Infrastructure as a Service (IaaS) fashion; the Management SW layer includes the hypervisors, operating systems, containers and security services, while the software is intended to be made available in a Platform as a Service (PaaS) fashion; and a User Workload layer composed of code, APIs and libraries enabling users to deploy applications. In terms of network architecture, an internal network is deployed to guarantee the space data center modules' interconnection; in addition, SDN technology is leveraged to manage resources dynamically and control the inner connectivity. This paper discusses the resilience and reliability achieved by modular redundancy, quality control systems and the utilization of Fault Detection, Isolation and Recovery (FDIR) mechanisms. According to the architecture proposed in [52], the space data center performance reaches earlier ground data center performance, achieving 13 Tera Floating Point Operations Per Second and 100 Tera Operations Per Second tailored to AI algorithms and 3.5 PB storage. The modular and compact design enables scalability for space data centers. On the other hand, ref. [53] discusses the implementation of relay satellites for ground data centers and space data centers' data transfer and highlights hardware robustness in front of high pressures, temperatures, space radiation, lightweight and compactness. Then, a simplified space data center architecture can be proposed as follows in Figure 19. Notice the three layers: infrastructure (physical resources), hypervisor (optional) and entities (DTM and XRM).

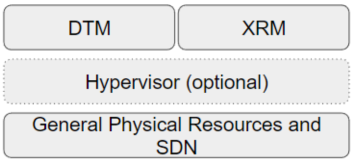

Figure 19. Simplified space data center structure.

In terms of hardware and software requirements for the space data center proposed in this paper, the proposal is not limited by the size of the hardware resources required. As a basic reference, according to the analysis presented in [52], the following software resources must be considered:

- Cloud Infrastructure such as OpenStack (Minimal Requirements: 24 CPUs, 24 GB RAM and 500 GB disk) [54].
- Orchestration platforms such as "Open Source MANO (OSM)" (Minimal requirements: 4 CPUs, 16 GB RAM and 80 GB disk) [55].
- Virtualization engines, such as "VMware" (Minimal requirements: 8 CPUs, 16 GB RAM and 500 GB disk) [56].
- Monitoring, management and fault detection software.
- Data base structures such as "Oracle" (Minimal requirements: 64 GB RAM and 500 GB disk) [57] .
- Firewall and cyber-security systems.

From the software resources previously listed and based on the earlier Figure 4, which illustrates a minimal scenario, at least two high-performance servers are required to support the Digital Twin Network operation (DTM server) and the extended Reality service provision (XRM server). The DTM server needs to provide the cloud infrastructure to support the DTN; thus, it is required to implement OpenStack, while OSM is required to orchestrate the DTN. Regarding the XRM server, VMware can be adopted to deploy the virtual network functions (VNFs), while OSM orchestrates such VNFs. Thus, the capacity for each server can be anticipated: the DTM server requires at least 24 CPUs, 40 GB RAM and 500 GB disk, while the XRM server requires at least 32 GB RAM and 200 GB disk for each VNF deployed, as well as a Graphic Processor Unit (GPU), for instance, a "NVIDIA RTX series". Moreover, additional high-performance servers are required to support the monitoring and fault detection procedure, to deploy the database structure and to implement a cyber-security tool. The hardware and software specifications to deploy the system will be deepened in future works, including the CPU architecture and GPU dedicated to support IA applications like the NVIDIA Jetson Xavier.

5. Conclusions and Future Work

A novel framework architecture for Digital Twin networks has been proposed to be able to provide immersive extended Reality Functions with the ultimate objective of extending the existing functions available in the original network, including both services for the users and network management and control functions. Furthermore, the supporting architecture for the aforementioned deployment has been introduced along with the entity modules and their specific operations. In addition, the different traffic flows that facilitate the realization of the system's operation have been presented. Subsequently, it has been demonstrated that despite the demanding requirements for implementing the proposed system, the DTN + XRF is achievable. The performance metrics defined in this study evidence that the system is capable of operating as an efficient and reliable network. The concepts outlined in this paper will serve as the foundation for future research, which will ultimately facilitate the actual implementation of the proposed DTN + XRF.

Finally, it is fundamental to highlight that the literature analysis has demonstrated that the paper's proposal represents a totally new proposal that can be envisioned as a future high-interest area with a wide range of innovative applications such as key fields including medical applications, surveillance systems, autonomous driving and space communications. This, it is important to highlight that this paper is the first one to approach the deployment of a DTN + XRF.

Regarding the current findings, the following actions will take place in our future work in order to be able to provide an early test-bed environment:

1. To define the technology to support the DTN's deployment. Virtualization techniques are envisioned as a suitable solution.
2. To analyze the appropriate transport communication protocol to communicate the ON and the DTN.
3. To analyze the hardware and software requirements for constructing the supporting entity modules.
4. To analyze the security approaches to ensure a secure DTN environment.
5. It is necessary to analyze the system's consistency validation processes based on simulations in order to provide a proposal for the real system's deployment.
6. To conduct a study to determine and estimate the dimensions for hardware and software resources.

Author Contributions: Conceptualization, X.C.-H. and X.H.; methodology, X.C.-H. and X.H.; validation, X.C.-H. and X.H.; formal analysis, X.C.-H. and X.H.; investigation, X.C.-H. and X.H.; resources, X.H.; writing—original draft preparation, X.C.-H. and X.H.; writing—review and editing, X.C.-H. and X.H.; supervision, X.H.; project administration, X.H.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the Agencia Estatal de Investigación of Ministerio de Ciencia e Innovación of Spain under project PID2022-137329OB-C41/MCIN/AEI/10.13039/501100011033.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Grieves, M. Digital Twin: Manufacturing Excellence Through Virtual Factory Replication. Available online: https://www.researchgate.net/publication/275211047 (accessed on 12 August 2024).

2. Minerva, R.; Lee, G.M.; Crespi, N. Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models. *Proc. IEEE* **2020**, *108*, 1785–1824. [CrossRef]
3. Wenzheng, L. Construction Methods, Data-Driven, and Operational Modes of Digital Twin Basin. In Proceedings of the 2023 IEEE 13th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 14–16 July 2023; pp. 158–161. [CrossRef]
4. WG Standardisation Focus Group; BDVA Task Force Data Space; BDVA Task Force Standards and Benchmarking. *Guidance for the Integration of Digital Twins in Data Spaces*; Alliance for IoT and Edge Computing Innovation: Brussels, Belgium, 2024.
5. Laaki, H.; Miche, Y.; Tammi, K. Prototyping a Digital Twin for Real Time Remote Control Over Mobile Networks: Application of Remote Surgery. *IEEE Access* **2019**, *7*, 20325–20336. [CrossRef]
6. El Saddik, A. Digital Twins: The Convergence of Multimedia Technologies. *IEEE MultiMedia* **2018**, *25*, 87–92. [CrossRef]
7. Wang, K.; Du, H.; Su, L. Digital Twin Network based Network Slice Security Provision. In Proceedings of the 2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI), Boston, MA, USA, 24–28 October 2022. [CrossRef]
8. Zhao, J.; Xiong, X.; Chen, Y. Design and Application of a Network Planning System Based on Digital Twin Network. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 900–904. [CrossRef]
9. Wei, Z.; Wang, S.; Li, D.; Gui, F.; Hong, S. Data-Driven Routing: A Typical Application of Digital Twin Network. In Proceedings of the 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI), Beijing, China, 15 July 2021–15 August 2021; pp. 1–4. [CrossRef]
10. Chen, D.; Yang, H.; Zhou, C.; Lu, L.; Lü, P.; Sun, T. Classification, Building and Orchestration Management of Digital Twin Network Models. In Proceedings of the 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, 11–14 November 2022; pp. 1843–1846. [CrossRef]
11. Polverini, M.; Lavacca, F.G.; Galán-Jiménez, J.; Aureli, D.; Cianfrani, A.; Listanti, M. Digital Twin Manager: A Novel Framework to Handle Conflicting Network Applications. In Proceedings of the 2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Phoenix, AZ, USA, 14–16 November 2022; pp. 85–88. [CrossRef]
12. Jian, M.; Long, B.; Liu, H. A Survey of Extended Reality in 3GPP Release 18 and Beyond. *Highlights Sci. Eng. Technol.* **2023**, *56*, 542–549. [CrossRef]
13. Caudell, T.; Mizell, D. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, Kauai, HI, USA, 7–10 January 1992; Volume 2, pp. 659–669. [CrossRef]
14. Brooks, F. What's real about virtual reality? *IEEE Comput. Graph. Appl.* **1999**, *19*, 16–27. [CrossRef]
15. Guan, J.; Irizawa, J.; Morris, A. Extended Reality and Internet of Things for Hyper-Connected Metaverse Environments. In Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Christchurch, New Zealand, 12–16 March 2022; pp. 163–168. [CrossRef]
16. Pereira, V.; Matos, T.; Rodrigues, R.; Nóbrega, R.; Jacob, J. Extended Reality Framework for Remote Collaborative Interactions in Virtual Environments. In Proceedings of the 2019 International Conference on Graphics and Interaction (ICGI), Faro, Portugal, 21–22 November 2019; pp. 17–24. [CrossRef]
17. Xu, Z.; Wu, S.; Zhang, L. A New Architecture of Augmented Reality Engine. In Proceedings of the 2023 2nd International Conference on Mechatronics and Electrical Engineering (MEEE), Abu Dhabi, United Arab Emirates, 10–12 February 2023; pp. 64–68. [CrossRef]
18. Zhang, Z.; Weng, D.; Jiang, H.; Liu, Y.; Wang, Y. Inverse Augmented Reality: A Virtual Agent's Perspective. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 16–20 October 2018; pp. 154–157. [CrossRef]
19. Shin, J.H.; Park, S.J.; Kim, M.A.; Lee, M.J.; Lim, S.C.; Cho, K.W. Development of a Digital Twin Pipeline for Interactive Scientific Simulation and Mixed Reality Visualization. *IEEE Access* **2023**, *11*, 100907–100918. [CrossRef]
20. Kamdjou, H.M.; Baudry, D.; Havard, V.; Ouchani, S. Resource-Constrained EXtended Reality Operated With Digital Twin in Industrial Internet of Things. *IEEE Open J. Commun. Soc.* **2024**, *5*, 928–950. [CrossRef]
21. Zhong, Y.; Marteau, B.; Hornback, A.; Zhu, Y.; Shi, W.; Giuste, F.; Krzak, J.J.; Graf, A.; Chafetz, R.; Wang, M.D. IDTVR: A Novel Cloud Framework for an Interactive Digital Twin in Virtual Reality. In Proceedings of the 2022 IEEE 2nd International Conference on Intelligent Reality (ICIR), Piscataway, NJ, USA, 14–16 December 2022; pp. 21–26. [CrossRef]
22. Künz, A.; Rosmann, S.; Loria, E.; Pirker, J. The Potential of Augmented Reality for Digital Twins: A Literature Review. In Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Christchurch, New Zealand, 12–16 March 2022; pp. 389–398. [CrossRef]
23. Böhm, F.; Dietz, M.; Preindl, T.; Pernul, G. Augmented Reality and the Digital Twin: State-of-the-Art and Perspectives for Cybersecurity. *J. Cybersecur. Priv.* **2021**, *1*, 519–538. [CrossRef]
24. Chen, H.; Xu, X.; Simsarian, J.; Szczerban, M.; Harby, R.; Ryf, R.; Mazur, M.; Dallachiesa, L.; Fontaine, N.; Cloonan, J.; et al. Digital Twin of a Network and Operating Environment Using Augmented Reality. In Proceedings of the 49th European Conference on Optical Communications (ECOC 2023), Glasgow, UK, 1–5 October 2023.
25. Hübel, N.; Kaigom, E.G. Codeless, Inclusive, and End-to-End Robotized Manipulations by Leveraging Extended Reality and Digital Twin Technologies. In Proceedings of the 2024 16th International Conference on Human System Interaction (HSI), Paris, France, 8–11 July 2024; pp. 1–6. [CrossRef]

26. Li, S.; Lin, X.; Wu, J.; Zhang, W.; Li, J. Digital Twin and Artificial Intelligence-Empowered Panoramic Video Streaming: Reducing Transmission Latency in the Extended Reality-Assisted Vehicular Metaverse. *IEEE Veh. Technol. Mag.* **2023**, *18*, 56–65. [CrossRef]
27. Carapinha, J.; Jiménez, J. Network Virtualization—A View from the Bottom. In Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, Barcelona, Spain, 17 August 2009; pp. 73–80. [CrossRef]
28. Yousaf, F.Z.; Bredel, M.; Schaller, S.; Schneider, F. NFV and SDN—Key Technology Enablers for 5G Networks. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 2468–2478. [CrossRef]
29. Zhang, S. An Overview of Network Slicing for 5G. *IEEE Wirel. Commun.* **2019**, *26*, 111–117. [CrossRef]
30. Mosharaf Kabir Chowdhury, N.M.; Boutaba, R. Network virtualization: State of the art and research challenges. *IEEE Commun. Mag.* **2009**, *47*, 20–26. [CrossRef]
31. Fischer, A.; Botero, J.F.; Beck, M.T.; de Meer, H.; Hesselbach, X. Virtual Network Embedding: A Survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1888–1906. [CrossRef]
32. Zhou, X.; Li, R.; Chen, T.; Zhang, H. Network slicing as a service: Enabling enterprises' own software-defined cellular networks. *IEEE Commun. Mag.* **2016**, *54*, 146–153. [CrossRef]
33. Shin, S.; Wang, H.; Gu, G. A First Step Toward Network Security Virtualization: From Concept To Prototype. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2236–2249. [CrossRef]
34. Shen, X.; Gao, J.; Wu, W.; Li, M.; Zhou, C.; Zhuang, W. Holistic Network Virtualization and Pervasive Network Intelligence for 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1–30. [CrossRef]
35. Li, T.; Zhou, H.; Luo, H.; Xu, Q.; Ye, Y. Using SDN and NFV to Implement Satellite Communication Networks. In Proceedings of the 2016 International Conference on Networking and Network Applications (NaNA), Hakodate, Japan, 23–25 July 2016; pp. 131–134. [CrossRef]
36. Chaudhry, A.U.; Yanikomeroglu, H. Temporary Laser Inter-Satellite Links in Free-Space Optical Satellite Networks. *IEEE Open J. Commun. Soc.* **2022**, *3*, 1413–1427. [CrossRef]
37. Liang, J.; Chaudhry, A.U.; Erdogan, E.; Yanikomeroglu, H.; Kurt, G.K.; Hu, P.; Ahmed, K.; Martel, S. Free-Space Optical (FSO) Satellite Networks Performance Analysis: Transmission Power, Latency, and Outage Probability. *IEEE Open J. Veh. Technol.* **2024**, *5*, 244–261. [CrossRef]
38. Hu, H.; Li, H.; Liu, Y. Bandwidth and Round-Trip Time Detection based Congestion Control for Multipath TCP over Highly Lossy Satellite Networks. In Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 26–29 November 2018; pp. 1072–1075. [CrossRef]
39. Farrell, S.; Cahill, V.; Geraghty, D.; Humphreys, I.; McDonald, P. When TCP Breaks: Delay- and Disruption-Tolerant Networking. *IEEE Internet Comput.* **2006**, *10*, 72–78. [CrossRef]
40. Yu, X.; Yu, F.; Hou, W.; Wang, X. State-of-the-Art of Transmission Protocols for Deep Space Communication Networks. In Proceedings of the 2010 First International Conference on Networking and Distributed Computing, Hangzhou, China, 21–24 October 2010; pp. 123–127. [CrossRef]
41. Caini, C.; Cruickshank, H.; Farrell, S.; Marchese, M. Delay- and Disruption-Tolerant Networking (DTN): An Alternative Solution for Future Satellite Networking Applications. *Proc. IEEE* **2011**, *99*, 1980–1997. [CrossRef]
42. Kim, H.S.; Kim, H.; Paek, J.; Bahk, S. Load Balancing Under Heavy Traffic in RPL Routing Protocol for Low Power and Lossy Networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 964–979. [CrossRef]
43. Zhu, Y.; Lin, L.; Ware, S.; Su, R. Supervisor Synthesis for Networked Discrete Event Systems With Communication Delays and Lossy Channels. In Proceedings of the 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 11–13 December 2019; pp. 6730–6735. [CrossRef]
44. Xie, Y.; Jiang, X.; Jin, G.; Chen, H. BBR-FIT: An Intelligent BBR based on the Reinforcement Learning to Boost the Network Efficiency over Time-Varying Networks. In Proceedings of the 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Hainan, China, 18–20 December 2022; pp. 1633–1640. [CrossRef]
45. Zhang, J.; Ren, S.; Dong, E.; Meng, Z.; Yang, Y.; Xu, M.; Yang, S.; Zhang, M.; Yue, Y. Reducing Mobile Web Latency Through Adaptively Selecting Transport Protocol. *IEEE/ACM Trans. Netw.* **2023**, *31*, 2162–2177. [CrossRef]
46. Cao, B.; Wang, R.; Sabbagh, A.; Peng, S.; Zhao, K.; Fraire, J.A.; Yang, G.; Wang, Y. Expected File-Delivery Time of DTN Protocol over Asymmetric Space Internetwork Channels. In Proceedings of the 2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Huntsville, AL, USA, 11–13 December 2018; pp. 147–151. [CrossRef]
47. Rehman, A.U.; Aguiar, R.L.; Barraca, J.P. Network Functions Virtualization: The Long Road to Commercial Deployments. *IEEE Access* **2019**, *7*, 60439–60464. [CrossRef]
48. Kultgen, D.; Grandy, C.; Andujar, D.; Grannan, A.; Kent, T.; Weathered, M.; Rein, J. *Deploying Extended Reality (XR) for Digital Operations and Maintenance (O&M) at the Mechanisms Engineering Test Loop (METL) Nuclear Science and Engineering Division*; Argonne National Laboratory (ANL): Argonne, IL, USA, 2023.
49. Coloma, S.; Frantz, A.; Meer, D.V.D.; Skrzypczyk, E.; Orsula, A.; Olivares-Mendez, M. Immersive Rover Control and Obstacle Detection based on Extended Reality and Artificial Intelligence. In Proceedings of the Proceedings—2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Orlando, FL, USA, 16–21 March 2024; pp. 743–744. [CrossRef]

50. Huang, Y.; Li, Y.J.; Cai, Z. Security and Privacy in Metaverse: A Comprehensive Survey. *Big Data Min. Anal.* **2023**, *6*, 234–247. [CrossRef]
51. Benzaid, C.; Taleb, T. AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. *IEEE Netw.* **2020**, *34*, 186–194. [CrossRef]
52. Ginosar, R.; Shor, Y.; Aviely, P.; Livni, J.; Alony, N.; Caspi, G. Data Center in Space (DCiS) Architecture. In Proceedings of the 2023 European Data Handling & Data Processing Conference (EDHPC), Juan Les Pins, France, 2–6 October 2023; pp. 1–4. [CrossRef]
53. Wijata, A.M.; Musiał, A.; Lazaj, D.; Gumiela, M.; Przeliorz, M.; Weiss, J.; Sagmeister, P.; Morf, T.; Schmatz, M.; Longépé, N.; et al. Designing (Not Only) Lunar Space Data Centers. In Proceedings of the IGARSS 2024—2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; pp. 6104–6108. [CrossRef]
54. OpenStack. 2024. Available online: https://docs.openstack.org/murano/rocky/admin/deploy_murano/prerequisites.html (accessed on 7 November 2024).
55. ETSI Open Source MANO. 2024. Available online: https://www.etsi.org/technologies/open-source-mano (accessed on 7 November 2024).
56. VMware. 2024. Available online: https://docs.vmware.com/en/VMware-Cloud-Provider-Stack/1.1/com.vmware.vcps.gsg.doc/GUID-83365584-C63E-4DC1-8DE8-8B3EE7543A3D.html (accessed on 7 November 2024).
57. Oracle. 2024. Available online: https://www.oracle.com/es/database/ (accessed on 7 November 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.