Received 27 April 2024; revised 5 June 2024; accepted 2 July 2024. Date of publication 10 July 2024; date of current version 10 September 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3426098

Networked Metaverse Systems: Foundations, Gaps, Research Directions

YULONG ZHANG, DIRK KUTSCHER, AND YING CUI (Member, IEEE)

The Hong Kong University of Science and Technology, Guangzhou 510530, China

CORRESPONDING AUTHOR: D. KUTSCHER (e-mail: dku@hkust-gz.edu.cn)

This work was supported in part by the Guangzhou Municipal Key Laboratory on Future Networked Systems under Grant 024A03J0623; in part by the National Natural Science Foundation of China under Grant 62371412; in part by the National Key Research and Development Program of China under Grant 2024YFE0200600; in part by the Guangdong Basic and Applied Basic Research Natural Science Funding Scheme under Grant 2024A1515011184; and in part by the Guangzhou-HKUST (GZ) Joint Funding Scheme under Grant 2024A03J0539.

(Special Issue on Challenges and Opportunities in Metaverse-Based Communication and Networking)

ABSTRACT This article discusses 'Metaverse' from a technical perspective, focusing on networked systems aspects. Based on a technical definition of the 'Metaverse,' we examine the current state and challenges in communication and networking within Metaverse systems. We describe the state-of-the-art in different enabling Metaverse technologies and provide a technical analysis of current Metaverse system architectures. We then detail the state-of-the-art and the gaps in four areas: communication performance, mobility, large-scale operation, and end system architecture. Based on our analysis, we formulate a vision for future Metaverse infrastructure, outlining goals, design concepts, and suggested research directions.

INDEX TERMS Metaverse, networking, Internet, information-centric networking.

I. INTRODUCTION

THE TERM 'Metaverse' often denotes a wide range of existing and fictional applications. Nevertheless, there are actual systems today that can be studied and analyzed. However, whereas a considerable body of work has been published on applications and application ideas, there is less work on the technical implementation of such systems, especially from a networked systems perspective.

In this article, we want to share some insights into the technical design of Metaverse systems, their key technologies, and their shortcomings, predominantly from a networked systems perspective. For the scope of this study, we define the 'Metaverse' as follows. The 'Metaverse' encompasses various current and emerging technologies, and the term is used to describe different applications, ranging from Augmented Reality (AR), Virtual Reality (VR),and Extended Reality (XR) to a new form of the Internet or Web. A key feature distinguishing the Metaverse from simple AR/VR is its inherently collaborative and shared nature, enabling interaction and collaboration among users in a virtual environment (See Section II-A for a detailed definition.)

Most current Metaverse systems and designs are built on existing technologies and networks. For example, massively multiplayer online games such as Fortnite use a generalized client-server model [1]. In this model, the server authoritatively manages the game state, while the client maintains a local subset of this state and can predict game flow by executing the same game code as the server on approximately the same data. Servers send information about the game world to clients by replicating relevant actors and their properties. Commercial social VR platforms such as Horizon Worlds and AltspaceVR use HTTPS to report client-side information and synchronize in-game clocks across users [2]. Mozilla Hubs, built with A-Frame (a Web framework for building virtual reality experiences), uses WebRTC communication with a Selective Forwarding Unit (SFU) [3]. The SFU receives multiple audio and video data streams from its peers, then determines and forwards relevant data streams to connected peers. Blockchain or Non-Fungible Token (NFT)-based online games, such as Decentraland, run exclusively on the client side but allow for various data flow models [4], ranging from local effects and traditional client-server architectures to peer-to-peer (P2P) interactions based on state channels; Upland is built on EOSIO [5], an open-source blockchain protocol for scalable decentralized applications, and transports data through HTTPS. Connections between peers in Upland are established using TLS or VPN tunnels [6].

Many studies have focused on improving various aspects of Metaverse systems. For example, EdgeXAR [7] is a mobile AR framework using edge offloading to enable lightweight tracking with six degrees of freedom (DOF) while reducing offloading delay from the user's view; SORAS [8] is an optimal resource allocation scheme for edge-enabled Metaverse, using stochastic integer programming to minimize the total network cost; Aliyu et al. [9] explores the issue of partial computation offloading for multiple subtasks in an in-network computing environment, aiming to minimize energy consumption and delay. However, these ideas for offloading computation and rendering tasks to edge platforms often conflict with the existing end-to-end transport protocols and overlay deployment models. Recently, a Deep Reinforcement Learning (DRL)-based multipath network orchestration framework [10] designed for remote healthcare services is presented, automating subflow management to handle multipath networks. However, proposals for scalable multi-party communication would require inter-domain multicast services, unavailable on today's Internet (Sections V–VIII provides further details on these issues).

In practice, **there is a significant disconnect between high-level Metaverse concepts, ideas for technical improvements, and systems that are actually developed and partially deployed**. Reference [2] analyzes the performance of various social VR systems, pinpointing numerous issues related to performance, communication overhead, and scalability. These issues are primarily due to the fact that current systems leverage existing platforms, protocols, and system architectures, which cannot tap into any of the proposed architectural and technical enhancements, such as scalable multi-party communication, offloading computation, rendering tasks, etc. Rather than merely layering 'the Metaverse' on top of legacy and not always ideal foundations, we consider Metaverse as a driver for future network and Web applications and actively develop new designs to that end. In this article, we want to take a comprehensive *systems approach* and technically describe current Metaverse systems, focusing on their networking aspects. We document the requirements and challenges of Metaverse systems and propose a principled approach to system design for these requirements and challenges based on a thorough understanding of the needs of Metaverse systems, the current constraints and limitations, and the potential solutions of Internet technologies.

A range of studies [11], [12], [13], [14], [15], [16], [17], [18], [25], [26], [27] have explored Metaverse from various perspectives. Table 1 provides a detailed summary of related literature in the field. Reference [11] stands out as one of the initial tutorials on Metaverse, identifying four key attributes of Metaverse: interoperability, scalability, realism, and ubiquity, and reviewing the progress in essential virtual world technologies. Reference [12] discusses the architecture of Metaverse systems and current use cases, highlighting Metaverse's role in the social good. Reference [13] evaluates Metaverse's development in relation to social aspects, virtual reality, supporting technologies, infrastructure, industrial projects, and national policies. Following [11], [12], [13], other surveys delve into more specific subsets of topics related to Metaverse systems. Reference [14] discusses the role of AI in Metaverse's development, such as enhancing communication resource management in dynamic network environments. Reference [16] introduced the technological concept of digitally synthesizing and recognizing human and olfactory senses, envisioning their applications in various Metaverse-related domains in the context of 6G communication networks. Reference [15] emphasized the potential of blockchain in shaping the future of Metaverse, focusing on current industrial developments and future industrial prospects. Reference [17] provided a detailed survey on edge-enabled Metaverse, covering the aspects of communications, computation, networking, and blockchain.

However, these surveys often **overlook implementation challenges from the communications, networking, and computation perspectives.** While understanding the enabling technologies of Metaverse systems is important, it is also necessary to discuss how they can be implemented. For example, while [18] examined AR/VR applications that enhance user immersion in Metaverse systems, it did not address large-scale deployment issues such as server load, network capacity, and seamless mobility support and management. Surveys on AR/VR service delivery [25], [26], [27] focus on implementation over 5G mobile edge networks and the computing and offloading architectures involved. Still, they are not Metaverse-specific and fail to consider scalability issues of communications in terms of users in the client-server-based Metaverse architecture. In contrast, our survey takes a holistic view of Metaverse systems, tackling fundamental implementation issues and offering practical solutions.

Unlike previous surveys (Table 1) on the general Metaverse concept [11], [12], [13], AI-enabled Metaverse [14], [15], [28], edge-enabled Metaverse [17], [29], [30], and specific applications in social goods [12], computational arts [31], education [32], retailing [33], and social AR/VR gaming [2], **our contribution can be summarized as follows.** I) We present a technical description of the 'Metaverse' based on existing and emerging systems, including a discussion of its fundamental properties, applications, and architectural models. II) We comprehensively study relevant enabling technologies for Metaverse systems, including HCI/XR technologies, networking, communications, media encoding, simulation, real-time rendering and AI. We also discuss current Metaverse system architectures and the integration of these technologies into actual applications. III) We conduct a detailed requirements analysis for constructing Metaverse systems. We analyze applications specific requirements and identify existing gaps in four key aspects: communication performance, mobility, large-scale operation,and end system architecture. For each area, we propose candidate

TABLE 1. Summary of related works vs. our survey.

Ref, Year	Key focus of the survey	How our survey differs
3d virtual worlds and the Metaverse: Current status and future possibilities [11], 2013	Four core characteristics of the Metaverse: interoperability, scalability, realism, and ubiquity, and reviewed progress in virtual world technologies.	Going beyond offering an initial overview of Metaverse systems, we take a comprehensive technical systems approach and describe current Metaverse systems technically, with a focus on their communication networking, and computation aspects.
Metaverse for social good: A university campus prototype [12], 2021	Survey of overall Metaverse concepts. It discusses the architecture of Metaverse systems, current instances, and its contribution to social good.	
A survey on the Metaverse: The state-of-the-art, technologies, applications, and challenges [13], 2023	Assessment of Metaverse evolution concerning social, virtual reality, supportive technologies, infrastructure, industrial initiatives, and government policies.	
Artificial intelligence for the Metaverse: A survey [14], 2022	Discussion of the role of AI in the development of Metaverse systems, focusing on six technical aspects.	In addition to identifying enabling technologies, our focus is on how these enablers can be implemented at scale from the perspectives of communication, networking, and computation. We analyze requirements for constructing Metaverse systems, identifying gaps in current technologies in terms of communication performance, large-scale operation, application development and performance, and security; Moreover, we propose potential technologies to bridge these gaps.
Fusing blockchain and ai with Metaverse: A survey [15], 2022	Discussion of the potential of blockchain and AI technologies in Metaverse systems.	
Digital scent technology: Toward the internet of senses and the Metaverse [16], 2022	Discussion of digital twin technologies for iOS and Metaverse systems.	
A full dive into realizing the edge-enabled Metaverse: Visions, enabling technologies, and challenges [17], 2022	Survey on the edge-enabled Metaverse, encompassing blockchain technology, computation, networking, and communication	
All one needs to know about Metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda [18], 2021	Discussion of the application areas of AR/VR, providing an overview of current technologies and future trends in these fields.	Current AR/VR systems are assuming existing protocols, platforms and technologies, which makes it hard to tap into the optimization potential such as compute offloading and joint resource optimization. We provide an analysis of current systems and their shortcomings and present an approach to a more principled system design to enable various optimizations in real-world systems.
A survey of mobile edge computing for the Metaverse: Architectures, applications, and challenges [19], 2022	Trend of AR/VR towards MAR/VR considering certain application areas.	
Mobile edge computing, Metaverse, 6g wireless communications, artificial intelligence, and blockchain: Survey and their convergence [20], 2022		
In-depth review of augmented reality: Tracking technologies, development tools, ar displays, collaborative ar, and security concerns [21], 2022		
From digital twin to Metaverse: The role of 6g ultra-reliable and low-latency communications with multi-tier computing [22], 2023	Discussion of individual optimizations for Metaverse systems, focusing on computation and communication support (e.g., edge resource management, offloading, etc.)	
Metaverse communications, networking, security, and applications: Research issues, state of-the-art, and future directions [23], 2023		
Overview of the integration of communications, sensing, computing, and storage as enabling technologies for Metaverse systems over 6g networks [24], 2023		

technologies to address these gaps. IV) We propose a research agenda for future Metaverse systems, based on our gap analysis and candidate technologies discussion.

We re-assess the fundamental goals and requirements, without necessarily being constrained by existing system architectures and protocols. Based on a comprehensive understanding of what Metaverse systems need and what end-systems, devices, networks and communication services can theoretically provide, we propose specific design ideas and future research directions to realize Metaverse systems that can meet the expectations often articulated in the literature. Note that this article focuses on the technical aspects of communication and networking within the Metaverse. To maintain clarity and adhere to space limitations, we do not discuss societal impact, commercial platforms, security, or other related topics in detail.

The remainder of this paper is organized as follows. Section II provides a comprehensive overview of the Metaverse concept. Section III discusses relevant base technologies. Section IV describes the architecture of Metaverse systems and current developments. Sections V–VIII conduct a detailed requirements analysis for constructing Metaverse systems, including application requirements, gap analysis in current technologies in communication, end systems, mobility, large-scale operation, and candidate technologies. Section IX discusses a future vision for Metaverse systems. Section X concludes this article. Figure 1 illustrates the organization of this survey.

The key acronyms are listed in Table 2.

II. WHAT IS THE METAVERSE?

The word *Metaverse* is used to describe different concepts, technical visions, and economic visions. In this Section, we provide a *technical* definition, beginning with a description of important general properties in Section II-A and a description of different relevant application types in Section II-B, followed by the synthesis of a general architectural model in Section II-C.

A. PROPERTIES

The definition of 'Metaverse' varies, such as lifelogging (e.g., Ghost Pacer[1]), collective space in virtuality (e.g., Microsoft Mesh[2]), embedded Internet/spatial Internet (e.g., Meta Horizon Workrooms[3]), mirror world (e.g., Active worlds[4]), and omniverse—a venue for simulation and collaboration (e.g., NVIDIA Omniverse[5]). Although no consensus exists on the definition ([34], [35], [36] list more than 100 different definitions from various papers), we observe that there are two main perspectives on defining 'the Metaverse':

1) A **general perspective** sees 'the Metaverse' as technology that changes how we work, consume media, etc. [37], [38], [39], [40], [41] It is a vague idea with many potentials, similar to the idea of the information superhighway.
2) A more **technical perspective** from a computer networking standpoint sees Metaverse as the next evolutionary step in how we work with the Web, combining technologies such as interactive VR with media on demand to have stricter latency requirements, not just for 2D but also 3D content [42], [43], [44], [45], [46]. The Metaverse is inherently shared and collaborative, distinguishing it from individual VR/AR experiences.

The Internet and the Web have evolved over time to provide richer multimedia experiences to support interactive communication and scalable distribution of Web content. For Metaverse, we can identify requirements for further technology development to support applications better:

- *Real-time Interaction Experience:* Demand higher bandwidth and lower latency to ensure seamless, real-time user interactions [47]. This requirement is essential for applications such as telepresence, remote collaboration, and gaming.
- *Enhanced Realism:* Metaverse applications may call for media forms such as holography or volumetric video to support a more realistic and immersive virtual experience. These advanced forms of media can enhance the feeling of presence in virtual environments, providing users with a more engaging and natural experience [48].
- *Connection with the Real World:* Certain Metaverse applications require a tighter link between the virtual and real worlds, as observed in digital twins or physical interaction through tactile communication [49]. These applications often involve integrating data from sensors and end devices, enabling the virtual environment to reflect and respond to real-world conditions and events.

B. CURRENT APPLICATION LANDSCAPE

Metaverse applications are proposed and developed for different application areas, such as healthcare [50], education [51], entertainment [52], e-commerce [53], and smart industries [54]. There are many papers [13], [17], [23], [35], [55], [56] that provide a taxonomy of Metaverse applications. For example, [55] categorizes Metaverse systems as 'Metaverse as a Tool' or 'Metaverse as a Target.' '*Metaverse as a Tool*' refers to using 'the Metaverse' to address real-world challenges. By contrast, '*Metaverse as a Target*' focuses on how Metaverse itself can be used for its own development and profit generation. Reference [17] considers the development of Metaverse systems from two perspectives. One is how actions in the virtual world can *affect* the physical world (V2P synchronization). For example, digital twins have been used to facilitate smart manufacturing, and digital goods in Metaverse systems can hold real monetary value. The other perspective is how actions in the physical world can *be mirrored* in the virtual world, driven by the digitalization and intellectualization of physical objects. For example, virtual 3D environments that mirror real-time reality can support remote work, socialization, and services such as education.

From these previous studies, we distilled the following aspects regarding **technical features** to structure the

[1] https://www.ghostpacer.com/
[2] https://www.microsoft.com/en-us/mesh
[3] https://www.meta.com/tw/zh/work/workrooms/
[4] https://www.activeworlds.com/
[5] https://developer.nvidia.com/omniverse

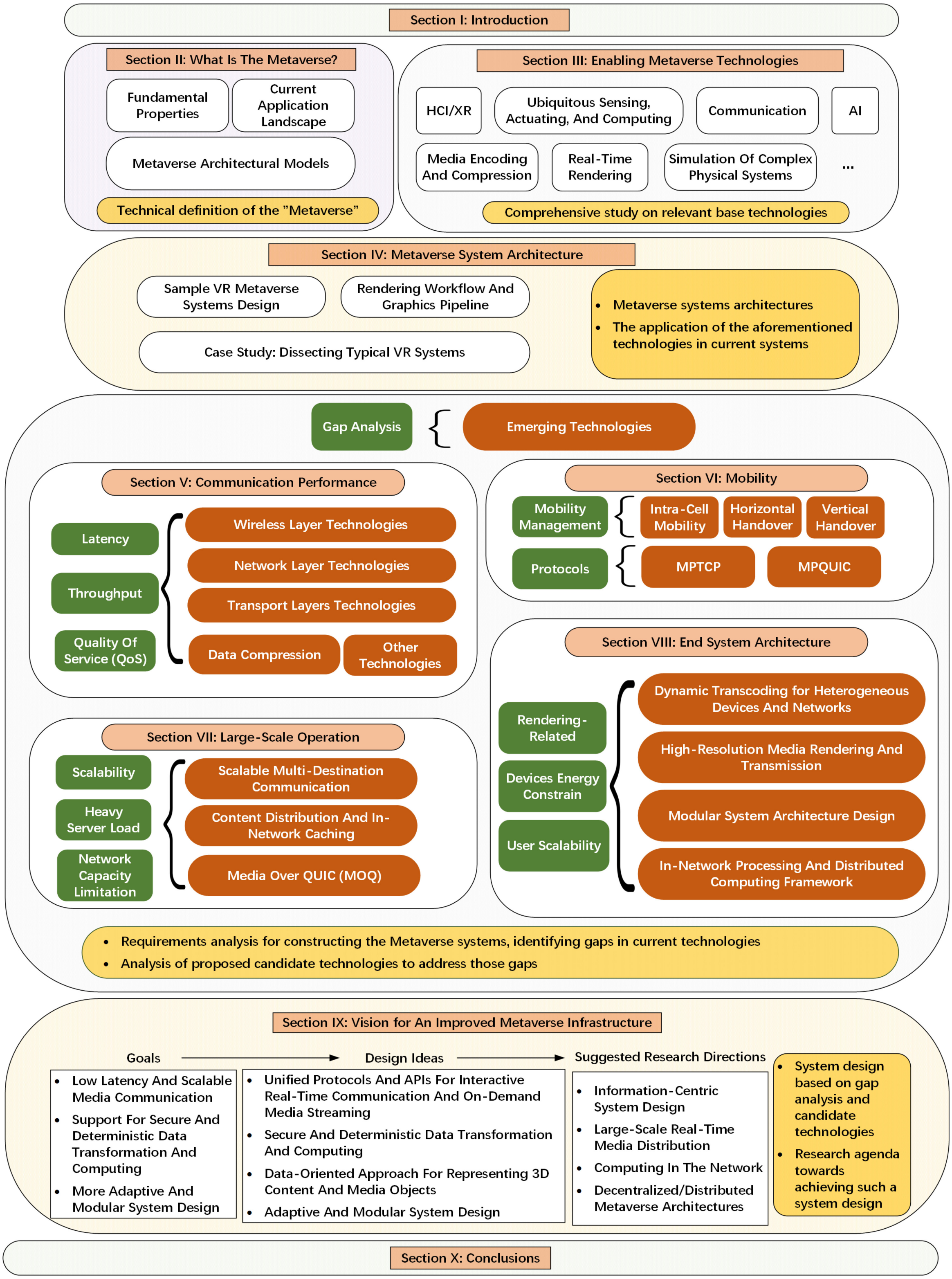
Section I: Introduction
Section II: What Is The Metaverse?
Fundamental Properties
Current Application Landscape
Metaverse Architectural Models
Technical definition of the "Metaverse"
Section III: Enabling Metaverse Technologies
HCI/XR
Ubiquitous Sensing, Actuating, And Computing
Communication
AI
Media Encoding And Compression
Real-Time Rendering
Simulation Of Complex Physical Systems
...
Comprehensive study on relevant base technologies
Section IV: Metaverse System Architecture
Sample VR Metaverse Systems Design
Rendering Workflow And Graphics Pipeline
Case Study: Dissecting Typical VR Systems
Metaverse systems architectures
The application of the aforementioned technologies in current systems
Gap Analysis
Emerging Technologies
Section V: Communication Performance
Latency
Throughput
Quality Of Service (QoS)
Wireless Layer Technologies
Network Layer Technologies
Transport Layers Technologies
Data Compression
Other Technologies
Section VI: Mobility
Mobility Management
Intra-Cell Mobility
Horizontal Handover
Vertical Handover
Protocols
MPTCP
MPQUIC
Section VII: Large-Scale Operation
Scalability
Heavy Server Load
Network Capacity Limitation
Scalable Multi-Destination Communication
Content Distribution And In-Network Caching
Media Over QUIC (MOQ)
Section VIII: End System Architecture
Rendering-Related
Devices Energy Constrain
User Scalability
Dynamic Transcoding for Heterogeneous Devices And Networks
High-Resolution Media Rendering And Transmission
Modular System Architecture Design
In-Network Processing And Distributed Computing Framework
Requirements analysis for constructing the Metaverse systems, identifying gaps in current technologies
Analysis of proposed candidate technologies to address those gaps
Section IX: Vision for An Improved Metaverse Infrastructure
Goals
Design Ideas
Suggested Research Directions
Low Latency And Scalable Media Communication
Support For Secure And Deterministic Data Transformation And Computing
More Adaptive And Modular System Design
Unified Protocols And APIs For Interactive Real-Time Communication And On-Demand Media Streaming
Secure And Deterministic Data Transformation And Computing
Data-Oriented Approach For Representing 3D Content And Media Objects
Adaptive And Modular System Design
Information-Centric System Design
Large-Scale Real-Time Media Distribution
Computing In The Network
Decentralized/Distributed Metaverse Architectures
System design based on gap analysis and candidate technologies
Research agenda towards achieving such a system design
Section X: Conclusions

FIGURE 1. Roadmap of This Article.

discussion of relevant Metaverse applications below (and later in Sections III and IV). These are always considered when designing Metaverse systems:

- Interactive multimedia communications
- Deterministic communications
- Virtual/augmented reality
- Interaction with the virtual/physical world
- Scalability with respect to the number of users

TABLE 2. Summary of important acronyms.

Acronym	Definition
AI	Artificial Intelligence
AR	Augmented Reality
BBR	Bottleneck Bandwidth and Round-trip propagation time
CDN	Content Delivery Network
D2D	Device to Device
ECN	Explicit Congestion Notification
FoV	Field of view
GOP	Group of Pictures
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
ICN	Information-Centric Networking
IoT	Internet of Things
MEC	Multi-access Edge Computing
MOQ	Media over QUIC
MR	Mixed Reality
MTP	Motion-to-Photon
NOMA	Non-Orthogonal Multiple Access
PCC	Performance-oriented Congestion Control
POV	Point of View
QoE	Quality of Experience
QoS	Quality of Service
QUIC	Quick UDP Internet Connections
ROI	Region of Interest
RTP	Real-time Transport Protocol
RTCP	RTP Control Protocol
STUN	Session Traversal Utilities for NAT Protocol
TACC	Traffic-Aware Congestion Control
TLS	Transport Layer Security
DTLS	Datagram Transport Layer Security
URLLC	Ultra-Reliable Low Latency Communications
VR	Virtual Reality
XR	Extended Reality

From a high-level application perspective, we can categorize current applications as follows:

Remote Work and Online Collaboration: Extending existing online tools for remote meetings and collaboration, Metaverse-based systems can enable users to create personalized workspaces or virtual offices that enable collaborations without geographical restrictions [57] (e.g., virtual meetings, classrooms, workplaces). It aligns with the 'Metaverse as a Tool' concept mentioned earlier, providing a more spatial and immersive experience than video calls. For example, applications such as Microsoft Mesh[6] and Horizon Workrooms[7] provide spaces where users can interact, undertake projects, or participate in educational activities. Moreover, Metaverse enhances spatial perception and immersive experiences by incorporating fundamental work-related tools, such as file sharing and collaborative documentation, thereby increasing productivity. Hence, this application class uses **interactive multimedia communication**, potentially addressing **scalability with respect to number of users** in **virtual/augmented reality**. For example, VIVE Sync is an all-in-one VR system for meeting and collaborating [58]. It offers a powerful avatar creation tool that lets users create their own avatar in a highly realistic way. Challenges for this application class include heavy server load and large network capacity in the presence of high data rates and large user groups [2]. Depending on the media bridging/mixing approach, throughput requirements can scale exponentially with respect to the number of users (see Sections V and VII for more details).

Social Life: The social aspect of Metaverse systems can combine the benefits of both online and offline social interactions [59]. Overcoming temporal and spatial restraints offers diverse forms of social interaction (e.g., digital travel, digital exhibitions, shopping). Technologies such as holographic virtual images and HCI/XR are used to create immersive environments, offering users a lifelike experience (see Section III). Such applications also align with the 'Metaverse as a Tool' concept as remote work and online collaboration, but are more socially oriented. This application class provides **interaction with virtual worlds**, **interactive multimedia communication**, and **scalability with respect to the number of users** in **virtual/augmented reality**. For example, the city of Seoul has recently launched a project, named 'Metaverse Seoul' [60]. This project creates a virtual environment of Seoul in Metaverse systems [61] and visitors worldwide can experience Seoul without physically travelling there. Platforms such as UC Berkeley's Virtual Campus (Minecraft)[8] and Tencent's virtual museum[9] also provide such experiences for various scenarios. This requires synchronizing vast data between an actual location and the user's position, and scalable interactive multimedia transmission similar to remote office and collaboration applications (potential solutions will be discussed in Section VII).

Gaming: Gaming is a major proposed Metaverse application. Rich interactive methods and scene-rendering technologies can provide an interactive experience and increase player engagement [62] (see Section III). Advances in XR have resulted in new forms of interactions compared to using conventional devices, such as a mouse, keyboard, and gamepad. XR-based gaming applications, such as Pokemon Go [63] and Beat Saber [64], allow interaction with virtual objects (e.g., a knife or a shield) in the same way as we would experience in the physical world (e.g., grabbing, gestures, and body movements). For example, Pokemon Go uses a mobile device camera to capture real-world scenes (e.g., streets, parks, and buildings), overlaying 3D virtual Pokemons (i.e., fictional creatures) onto these scenes using augmented reality (AR), making them appear as part of the real world on a device screen. Such applications align with the 'Metaverse as a target' concept, being

[6] https://www.microsoft.com/en-us/mesh

[7] https://www.theverge.com/2021/8/19/22629942/facebook-workroomshorizon-oculus-vr

[8] https://mcb.berkeley.edu/labs/shastri/virtual_tour.html

[9] https://en.dpm.org.cn/about/news/2019-09-18/3089.html

stand-alone and entirely reliant on the virtual environment compared to Remote Work & Online Collaboration and Social Life applications. This application class prioritizes **interactive multimedia communication** and **interaction with the virtual/physical world**,and **scalability for large user groups**. For example, about 12.3 million concurrent users attended Travis Scott's concert on Fortnite (a Metaverse gaming platform) in April 2020 [65].

Simulation and Modeling: Metaverse-based simulation and modeling intertwine with various applications such as gaming, social phenomenon research (e.g., simulating social issues, ethics, and policy-related issues [55]), marketing simulation (e.g., virtual assets and workflow control [66]), and educational and museum experiences [67]. Simulation and modeling applications align with the 'Metaverse as a Tool' concept and can simplify complex tasks (e.g., aircraft engineering) [55]. One particular application in this space is referred to as *Digital Twin*, which is a realistic replication of a real-world system, potentially mirroring the real-world system in real-time by capturing and controlling physical world systems. Such systems, especially when involving the control of critical components, can use **deterministic communication**. For example, [68] proposed a deterministic transmission framework for distributed power systems based on digital twins, using a time-sensitive network (TSN) configuration method for information exchange between the virtual and fundamental aspects of a distributed power generation system. Omniverse [69] has been applied to industrial simulation and modeling, enhancing productivity across stages of product development, manufacturing, collaboration, and optimization [70]. Further details for simulation and digital twin are discussed in Section III.

Other Potential Metaverse Applications: Metaverse-based **training and education** applications [32], [71], [72] extend existing online tools to virtual interactions that allow learners to explore and manipulate objects (similar to Remote Work and Online Collaboration applications but more education oriented). Through virtual scenes, learners can get close to reality while avoiding risks, reducing costs, etc [73]. For example, Jong et al. [74] proposed a virtual-physical blended Metaverse classroom that teaches physical geography through spherical video-based VR immersion. This method leverages spherical video-based VR to immerse students in a 3D environment, enhancing their engagement and understanding. The immersive nature of VR bridges the gap between theoretical knowledge and practical application, making complex geographical concepts more accessible and engaging. Pinto et al. [75] proposed using VR for rescue training for hydrogen vehicle accidents. Traditional training methods cannot replicate the unique dangers of hydrogen fuel, but VR creates a realistic, risk-free environment for rescuers to practice and hone their skills, highlighting VR's capability to prepare individuals for specific emergencies without exposing them to real-world dangers. Sidh [76] is a gamified firefighter training simulator. By engaging with virtual environments, trainees receive immediate feedback on their actions, allowing for a dynamic learning process that closely mirrors real-life firefighting scenarios. This real-time interaction enhances the effectiveness and retention of training, preparing individuals for the physical and mental challenges of emergency response.

Metaverse has been proposed as a tool for **cultural preservation** [77], [78]. For example, rising sea levels have prompted the island nation of Tuvalu to build a Metaverse country to protect its cultural and artistic heritage before it is completely submerged [79]. Similarly, a collaborative discussion has started on preserving the Australian Aboriginal (First Nation) culture by replication in Metaverse systems [80] and preserving Japanese cultural elements such as Japanese tea ceremonies. However, meeting practical requirements is challenging, especially in terms of immersion and a sense of presence (similar to Social Life applications).

Some Metaverse systems incorporate **virtual economy** elements, such as digital ownership and currencies. People can trade and exchange virtual assets such as virtual real estate, digital art, and virtual currency. However, the virtual economy will differ from the real-world economy, so it is necessary to have a dedicated regulatory framework that guarantees capability and fairness. Blockchain and decentralized systems have also been proposed as enablers of economic models in Metaverse systems, such as decentralized markets, virtual currencies, and smart contracts [81]. It has been claimed that this could foster a more dynamic and diversified economy in Metaverse systems composed of creators, consumers, and entrepreneurs, creating opportunities for software developers, designers, content makers, and others. Examples of applications are presented in Figure. 2.

C. CURRENT ARCHITECTURAL MODEL

Despite many studies on various aspects of Metaverse [14], [82], [83], a comprehensive understanding of its **technical architecture** remains challenging. Wang and Zhao [19] argued that Metaverse systems comprise three distinct areas: physical, human, and virtual. The physical area provides the essential infrastructure supporting the Virtual and Human areas; the human area focuses on social interactions and activities; and the virtual area handles and processes digital information from the physical and human areas. Ali et al. [23] suggested a Metaverse architecture that integrates the physical, human, and digital worlds through user-controlled avatars, virtual environments, and computer-generated elements. While their architecture leverages smart devices, Human-Computer Interaction (HCI), extended reality, blockchain, digital twins, and artificial intelligence to facilitate information flow, they lack detailed explanations of the Metaverse systems engine or the processes involved in data management and transmission in the physical world. Lee et al. [18] viewed Metaverse systems as a pipeline connecting the physical world to its digital twins, relying on key technologies (e.g., blockchain, computer vision, distributed networks, ubiquitous computing, scene understanding, and interfaces) and ecosystems (e.g., avatars,

(a) AltspaceVR platform

(b) The non-fungible LAND tokens constitute the Decentraland Metaverse

(c) Nvidia Omniverse for modeling physics, materials, and real-time path tracing

FIGURE 2. Examples of (a) AltspaceVR[10], (b) Decentraland[11], and (c) Nvidia Omniverse[12].

content creation, data interoperability, social acceptability, security, and privacy) for support. Siriwardhana et al. [25] explored Metaverse systems architecture from a mixed AR perspective, classifying it based on the locations of core AR processing functions (i.e., cloud-based, edge-based, localized, or hybrid-based).

The concept of a unified Metaverse architecture has yet to exist. 'The Metaverse' comprises various technical systems, each with distinct characteristics. For example, social VR and gaming systems differ mainly in design and functionality. Based on previous studies, we distilled a **representative architectural model** (depicted in Figure 3) that includes key technical elements that are relevant to the application classes described in Section II-B without necessarily including all possible niche features that are mentioned in the literature. First, from a high-level perspective, 'the Metaverse' is a concept that connects the physical world with the digital world. The **physical world** comprises the devices, users, physical infrastructure, etc. Examples include prototype machines in digital twin systems or users equipped with smart devices, such as Head-Mounted Display (HMD), AR goggles [84], and wristband sensors in AR/VR systems. **The physical infrastructure** includes sensors and actuators, supporting multisensory data acquisition, perception, processing, transmission, and decision-making.

- *Data Acquisition:* Sensing devices such as Internet of Things (IoT) sensors and wearable and implantable devices are used for pervasive sensing. Authentication and access control mechanisms are required to manage the massive, fine-grained data produced and gathered in real time within Metaverse systems. Moreover, throughout the life cycle of Metaverse services, the reliability and traceability of data must be ensured, along with stringent requirements for privacy protection.
- *Data Management and Storage:* A massive amount of data from multiple sources must be efficiently managed and stored. This ensures the construction and evolution of virtual representations [85]. Data in this context are typically heterogeneous, multi-scale and multi-source.
- *Data Preprocessing and Analysis:* Infrastructure should provide reliable data-driven preprocessing and analytics. Accurately extracting underlying information and knowledge from a massive amount of data enhances the overall effectiveness of the system.

A **Digital World** can consist of one or many distributed virtual worlds, providing virtual environments and services [86], [87]. For example, *Second Life* [88] consists of multiple virtual worlds operating concurrently, each managed by specific servers with in-game clock synchronization. Virtual Worlds include the virtual environment, digital avatars, virtual services/goods, and digital assets, supported by *Metaverse Engine*. Metaverse Engine includes all the relevant enabling technologies in the context of Metaverse.

In Figure 3, we collect the core technical Metaverse components under the label **Metaverse Engine**, representing technologies such as HCI/XR, Artificial Intelligence (AI), simulation, etc., for creating, maintaining, and updating virtual spaces. Processing inputs from sensors, control components, and data from physical and digital entities alongside their activities contributes to creating realistic virtual environments, guiding avatar behavior, etc. For example, *XR/HCI technologies* support users and physical environments to extend reality and control interaction (e.g., head tracking and physical controllers). *Ubiquitous sensing, actuating, and computing technologies* collect data from physical sensors to maintain the states of physical systems in digital twins and control physical systems based on events and decisions made in the virtual world. *Communication* protocols and services such as media mixers and relays enable real-time communication between physical and virtual entities [89]. *AI* augments Metaverse systems by enabling personalized avatars and content creation. *Simulations for digital modeling and reconstruction* create virtual replicas of physical entities in a digital environment. Physical geometries, properties, behaviors, and rules are digitized holistically to create high-fidelity virtual representations. These virtual entities rely on real-world data from the physical world to formulate a

[10] https://i0.wp.com/mashdigi.com/wp-content/uploads/vr-interactions.jpg?ssl=1

[11] https://cryptosrus.com/this-is-where-jp-morgan-thinks-the-metaverse-is-going/

[12] https://github.com/PegasusSimulator/PegasusSimulator?tab=readme-ov-file

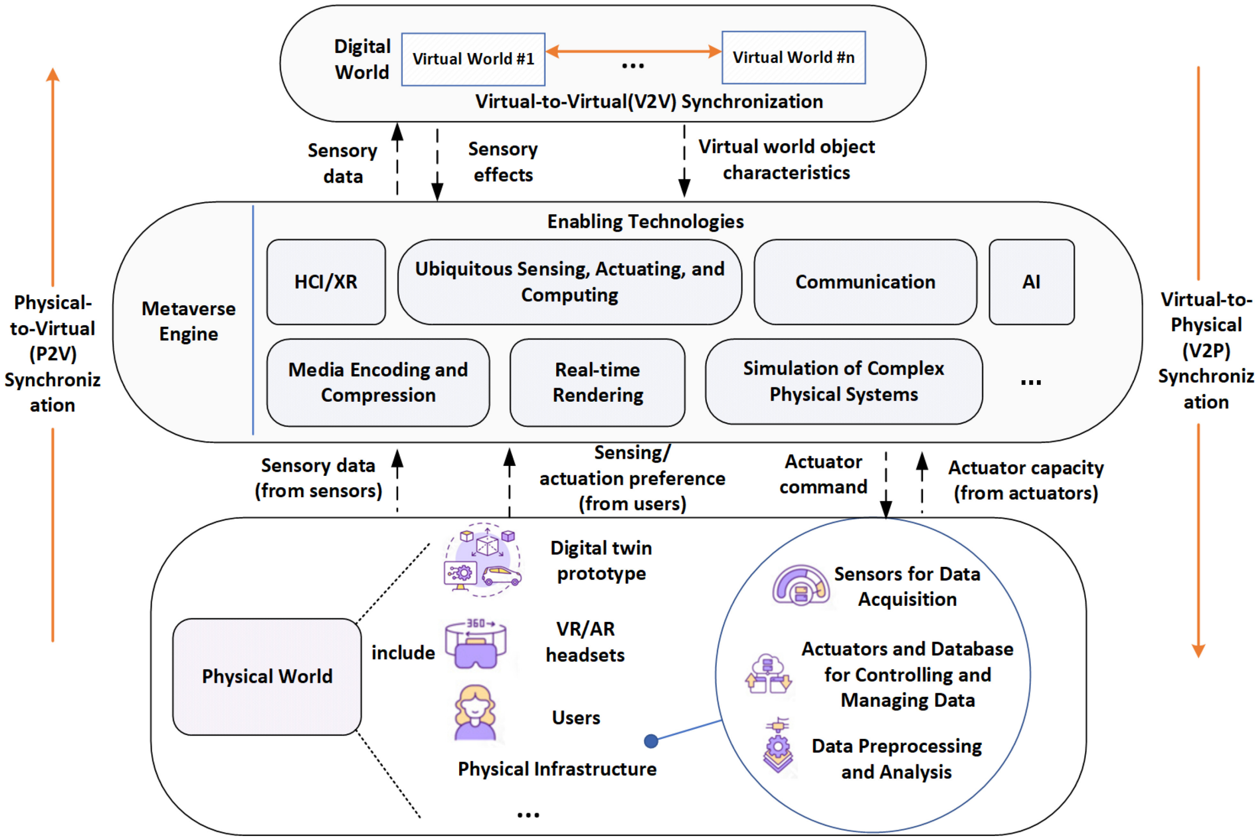
Digital World
Virtual World #1
...
Virtual World #n
Virtual-to-Virtual(V2V) Synchronization
Sensory data
Sensory effects
Virtual world object characteristics
Physical-to-Virtual (P2V) Synchronization
Metaverse Engine
Enabling Technologies
HCI/XR
Ubiquitous Sensing, Actuating, and Computing
Communication
AI
Media Encoding and Compression
Real-time Rendering
Simulation of Complex Physical Systems
...
Virtual-to-Physical (V2P) Synchronization
Sensory data (from sensors)
Sensing/ actuation preference (from users)
Actuator command
Actuator capacity (from actuators)
Digital twin prototype
Physical World
include
VR/AR headsets
360
Users
Physical Infrastructure
...
Sensors for Data Acquisition
Actuators and Database for Controlling and Managing Data
Data Preprocessing and Analysis

FIGURE 3. Technical Architecture of ‘the Metaverse.’

real-time status. *Real-time rendering* generates high-fidelity images, enabling digital avatars to simulate human behaviors and interact dynamically (See Section III for more details).

There are synchronizations within Physical/Virtual Worlds (i.e., *Intra-World Flow of Information*) and between Virtual Worlds (i.e., *Inter-World Flow of Information*). **Intra-World Flow of Information** (within a user group) represents the flow of information within one single Metaverse (including both physical world and virtual worlds). This includes the transmission of data from sensors (e.g., sensory data), users (e.g., sensing/actuation preference), actuators (e.g., actuator capacity), and virtual world information (e.g., sensory effects and virtual world object characteristics) transmission (details in Section IV). **Inter-World Flow of Information** (between user groups) represents the information flow between virtual worlds, such as inter-server game state synchronization, clock synchronization, and peer synchronization. This is common in many Metaverse systems. For example, Minecraft [90] allows up to thousands of players in shared virtual worlds to be distributed on multiple servers of a server cluster. This distribution requires synchronizing the game state information among servers, with each server updating the game state information to the other servers in the cluster. More details regarding content delivery and synchronization are discussed in Sections IV and IX.

D. TAKE-AWAYS

The term ‘Metaverse’ is broad. It encompasses a wide range of domains, from immersive education and remote work to digital twin simulations and virtual economies, each with distinct characteristics. Despite their diversity, these applications often rely on common requirements and share similar architectural principles. In this survey, we organize these applications within a unified architectural framework (Figure 3). This framework is not intended as a one-size-fits-all standard model but rather as a guide to help understand how these systems work. In the next Section, we will explore enabling technologies for these applications in detail, examining their respective state of the art.

III. ENABLING METAVERSE TECHNOLOGIES

The realization and functionality of Metaverse applications largely depend on the development and integration of several common enabling technologies. They need to manage the state of 3D avatars, environments, and objects, essentially the people, places, and things that make up ‘the Metaverse,’ render graphics at real-time frame rates, and ensure communications between clients for a consistent shared experience among participants. In this section, we discuss these technologies, their current challenges, and their future directions. Note that while they may not be exactly on the same level, these show some specific aspects that are particularly essential for Metaverse.

A. HCI/XR FOR IMMERSIVE EXPERIENCE AND INTERACTIVITY

The seamless integration of XR and holographic technologies is a critical component of HCI/XR [91]. XR is used as a common name for AR, VR, and Mixed Realty (MR). Currently, VR/AR/MR are the primary interaction technologies used by Metaverse to create highly interactive virtual worlds. AR overlays physical surroundings with computer-generated content, delivering sensory information including visuals, audio, smell, and haptics. VR provides synthetic landscapes, enabling interaction via head tracking and physical controllers, thus immersing users in digital worlds; MR blends the real and virtual worlds, forming a continuum that connects a completely real environment at one end to a completely virtual environment at the other. It allows physical and digital objects to coexist and interact in real time.

Holographic technology is a recording and reproduction technology that presents a 3-D image of an object using optical means. It captures the magnitude and phase of optical waves through computer and electronic imaging techniques (e.g., coherent optical interference), and acquires all the information about an object, including its form and size. Users can view holograms from different angles with the naked eye without needing a portable device. Holographic display technologies include reflection-based (using a reflective surface to display the holographic image, commonly seen in art and security features, such as credit cards or ID badges) and laser-driven holography (using a laser to illuminate the hologram, often used in high-quality imaging applications) [91]. Recent advancements in holographic displays have led to the development of digital holography, which uses spatial light modulators (SLMs) and computer-generated holograms (CGHs) to produce dynamic and interactive 3D images [92], [93].

Brain–computer interfaces (BCIs) connect a person's brain to external physical worlds by coding a brain signal into commands that can be identified by computing devices [94], thus enabling the spatial interface between the virtual world and the real world. In principle, a BCI system can detect and register neural signals using external electrodes or optical sensors attached to the skull and other body parts. These neural signals are then processed using AI for pattern analysis and identification before responding to neural stimulation [95]. Recent advances in neural interfaces have improved the precision and reliability of BCIs. These include high-density electrode arrays that provide detailed readings of neural activity, enhancing signal detection resolution and accuracy [96]. Improved machine learning algorithms aid in the interpretation of complex neural signals [97]. Innovations in non-invasive BCI technologies, such as functional near-infrared spectroscopy (fNIRS) and magnetoencephalography (MEG), offer less intrusive methods for neural signal acquisition [98]. However, challenges remain in achieving real-time processing speeds and ensuring user experience during prolonged use [99].

B. UBIQUITOUS SENSING, ACTUATING, AND COMPUTING

Ubiquitous sensing and actuating technologies collect parameters from physical devices to maintain identical states in their digital twins and manipulate physical systems based on events and decisions taken in the virtual world. In telesurgery, remote surgical operations are performed on a patient's digital twins through a robotic arm, and the results can be analyzed [100], [101]. Specifically, the collection of pervasive smart objects, sensors, and actuators forms the backbone of a comprehensive sensing and control system. It enables multi-faceted and multi-modal data perception from the environment and human bodies, thereby ensuring high-precision device control. Wearable and implantable biomedical devices, IoT-connected vehicles, and human-robot interaction systems are examples of sensing devices used for pervasive sensing [102].

With respect to signals, ubiquitous sensing captures multi-modal signals from users and their environment, including human senses (e.g., vision and sound) and beyond-human senses (e.g., RF and inertial). Its main *goals* include capturing high-fidelity visual and acoustic information for digital twin construction, and tracking the user's detailed position and body orientation for suitable sensory transformation during movement and interaction. Progress has been made in both *Inside-out* and *Outside-in* tracking [103]. *Inside-out tracking* uses body-mounted devices (e.g., a head-mounted one), whereas *Outside-in* involves ambient devices (e.g., in the environment). However, the current technologies have limitations. *Outside-in tracking* confines users to pre-equipped physical spaces, whereas *Inside-out tracking* can be physically burdensome and may not capture the complete body position, particularly the lower body. Beyond position tracking, Metaverse systems demand finer-grained and richer state sensing, such as identifying the type of floor a user walks on [104], finger gestures, touch force [105], and facial expressions. The captured physical signals must be processed appropriately for such sensing tasks. For example, LiDAR point clouds need processing to isolate them from the background [106]. RGB-D images need mesh processing for viewing. 360-degree video cameras need to be converted into other representations such as navigation graphs [107].

Ubiquitous computing refers to seamless interaction between users and their environment through pervasive smart devices (often mobile) integrated into the surroundings or worn on the body [108]. This includes real-time, immersive Metaverse experiences via ubiquitous smart objects and network access within their environment. It also refers to omnipresent but invisible computing (not in traditional PC-like systems, but instead on ubiquitous smart connected devices that form a user's local computing cloud), realizing the user's virtual presence through avatars or other artifacts.

First principles algorithms [109] were used in the past to process ubiquitous sensing data. In recent years, deep neural network approaches have also been used to detect complicated and uncertain physics [110], [111] and extract state information from unstructured high-dimensional sensor signals.

C. COMMUNICATION

Communications in Metaverse systems are expected to support distributed multimedia exchange, including continuous multi-modality media such as graphical animations, high-quality audio/video, haptics, and interactive images. This requires high bandwidth to deliver the promised high-quality multimedia data in real-time while maintaining fairness to other application flows. Metaverse systems typically need three types of communication [2], [112], [113]. *I) Interactive Real-time Communication.* This involves low-latency, fine-grained, hierarchical exchanges of arbitrary objects and data streams (e.g., user-generated content and dynamic user interactions), and coding for enhanced communication robustness and efficiency. *II) Scalable content distribution.* This includes distributing high-fidelity static and dynamic 2D/3D objects and scene descriptions, supporting link-layer broadcast/multicast where possible, and efficient, robust data sharing and multi-destination delivery when applicable. *III) 'Control channel' communication.* This includes maintaining and updating the states of physical worlds in digital twins with data from physical sensors and controlling physical systems based on user actions, events, and decisions made in virtual worlds. This type also requires low latency and high reliability to ensure deterministic behavior, particularly for mission-critical Metaverse applications such as digital twins.

Latency also depends on the type of codec. 2D video coding in the current era predominantly uses rate-distortion optimization techniques [114], which aim to minimize the bit rate required for video representation while maintaining acceptable visual quality. The encoding process needs to trade-off between the quantity of information to be encoded (bit rate) and the resultant visual quality (distortion), aiming to achieve an optimal balance between the bit rate and distortion for each coding unit [115]. Beyond the transmission of conventional 2D media, the seamless encoding and decoding of *3D* virtual scenarios and items (e.g., 360-degree imagery, light field data, volumetric visual signals, and digital holograms) [116], [117], [118], [119], [120], [121] is also needed (we will further explain them in Section III-D). See Sections V and IX for more details on communication protocols, control schemes, and design ideas.

D. MEDIA ENCODING AND COMPRESSION

Media elements in Metaverse extend beyond conventional 2D videos and images from the current mainstay of Internet traffic to a broad spectrum of media forms across various dimensions. For transmission purposes, they can be segmented into different types: I) high-resolution 2D video; II) the creation of an explorable 3D environment reflective of the physical world; III) 3D assets that combine both computer-generated and sourced content; and IV) intersections with real-world environments [122].

Specifically, *360-degree video* [123], [124], *free viewpoint television (FVT)* [125] and *light field photography* [126] are integral to features such as pre-recorded backgrounds. 360-degree video records a panoramic 360-degree view with one or more cameras [123], [124]. Transcending two-dimensional limitations provides an all-encompassing view of the virtual space. 360-degree videos typically consume more bandwidth than regular videos, since they require more data to cover all spatial directions. The bandwidth needed for streaming 360-degree video depends on the resolution, frame rate, and QoE desired [127]. For example, a 360-degree video with an equivalent HD TV viewing experience at 60 fps would result in a 12K resolution and a 400 Mbps bitrate (see Table 7 for more data). FVT records scenes from multiple angles and allows interpolation between views [125]. Facilitating multiple perspective overlays of pre-recorded content creates a semi-immersive layer within Metaverse systems. Light field photography allows post-capture manipulation of focus, depth of field, orientation, and other parameters [126]. It captures light intensity from different scene directions, increasing the realism of the content. A light field video with a resolution of 1920x1080 and a frame rate of 60 fps would require a bandwidth of approximately 200 Mbps [128]. *Spatially selectable video* [129] uses technologies such as MPEG-DASH Spatial Representation Descriptions (SRD) to provide videos with dynamic selectivity [130], enabling users to choose areas of interest within a high-resolution panoramic shot. It extends beyond 2D to encompass 3D elements, facilitating a personalized narrative in the digital world by navigating through non-obscured or interest-specific sections of multiple sources (e.g., the real-time or just-in-time assembly of video frames for live streaming). *Volumetric streaming* offers a six degrees of freedom (6 DoF) viewing experience, and encapsulates viewer position (X, Y, Z) and orientation (yaw, pitch, roll) [131]. It comprises a collection of data points in a spatial domain, representing a 3D shape or object with each point designated by its set of X, Y, and Z coordinates [119] (e.g., 3D point clouds, holograms). However, volumetric streaming faces challenges, primarily due to the complexities of 3D data compression and inadequate hardware support. Performance tests on an RTX 3090 GPU computer [132] show that volumetric data streaming can incur high network bandwidth (around 300 Mbps) and encoding and decoding latency (around 656ms and 248ms, respectively), far from reaching the performance of 2D video streaming. The required bandwidth to stream an uncompressed point cloud video with 100k points per frame at 24 FPS is $9\frac{\text{bytes}}{\text{point}} \times 100k$ points $\times 24\frac{\text{frame}}{\text{sec}} \times 8$ bits $= 172.8$Mbps [133].

Tiling schemes have been proposed to enhance transmission flexibility and sometimes reduce the data volume. These schemes partition frames into multiple tiles, treating each

TABLE 3. Compression standards and formats for different media modes.

Modality	Compression standards & formats
360-degree video	AVC/H.264, HEVC/H.265, VVC/H.266, VP9 and AV1
Light field	JPEG Pleno, MV/3D-HEVC, MIV
Point clouds	JPEG Pleno, V-PCC, G-PCC, Draco, Corto
Mesh	Draco, Corto, V-DMC
Digital hologram	JPEG Pleno

tile as a discrete segment of the video that can be encoded independently [134], [135]. They allow for variable encoding rates across tiles (e.g., some can be encoded at a high rate and others at a low rate), providing greater flexibility in quality management by selecting only the tiles required. Additionally, tiling schemes allow individual tiles to be requested separately, which can be useful for some network protocols such as Information-Centric Networking (ICN), potentially sharing tiles among multiple viewers with the same viewpoint. This can lead to more efficient content distribution [136] (see Section IX-C for specific design ideas). Tiling schemes can be categorized into non-tiling, uniform tiling, non-uniform tiling, and dynamic tiling. *Non-tiling schemes* handle streaming akin to traditional video streaming, streaming each frame as an entire entity to the HMD [137]. *Uniform tiling schemes* are adopted [138], [139], [140] to accommodate users' dynamic viewports, where video segments are further divided spatially into equal-sized tiles, with different tiles allocated at different bitrate levels. Technologies such as MPEG-DASH SRD [130] are used to describe the spatial relationships among tiles within the same temporal video segment, and Low Latency Common Media Application Format (LL-CMAF) [141] is used to enable the media segments to be divided into smaller chunks and delivered faster. However, fixed-size tiling approaches can reduce the encoding efficiency [135], [142]. *Non-uniform tiling schemes* were proposed to address this issue. They aim to adapt the tile sizes and shapes to better match the spatial complexity of the content within the video frame, allocating more resources to areas with higher visual importance or complexity,

Table 3 summarizes the compression standards and formats for different visual media modes. Standards such as AVC/H.264, HEVC/H.265, VVC/H.266, VP9, and AV1 are widely used to encode 360-degree videos, each offering a distinct trade-off. AVC/H.264, while highly compatible and less computationally demanding, provides lower compression efficiency compared to newer standards. HEVC/H.265 improves compression efficiency by about 50% over H.264, reducing file sizes without compromising quality [143]. However, it demands more computational power and has complex licensing issues. VVC/H.266 achieves the highest compression efficiency, with average bitrate savings of approximately 78%, 63%, and 53% compared to AVC/H.264, AV1, and HEVC/H.265, respectively [144]. This makes it ideal for high-resolution content like 8K video. However, VVC/H. 266's advanced compression techniques result in higher computational complexity, with encoding and decoding times up to 6.5 times and 1.5 times longer than HEVC [145]. VP9 provides higher compression efficiency than AVC/H.264 but generally lags behind HEVC/H.265. It is a popular choice for Web streaming due to its balance of efficiency and complexity. AV1 surpasses HEVC/H.265 in compression efficiency by about 30%. However, it has high computational complexity, with an encoding time of 4 to 10 times longer than HEVC [146].

Several alternative codecs, including Draco [147] and Corto [148], have been specifically designed to encode 3D content in point cloud and mesh representations. These codecs provide effective encoding tools and low-complexity for decoding, which are widely supported in Web browsers. However, there is still a lack of temporal redundancy in dynamic 3D content since frames are typically encoded independently. Instead, HEVC/H.265 extensions, including multi-view HEVC and 3D-HEVC [149] were proposed to encode 3D video content in multi-view and 3D representations, respectively. However, despite using temporal redundancy, such extensions have not been widely adopted in industry, mainly because of their low coding efficiency and high decoding complexity, which scales linearly with the number of views. Therefore, they do not satisfy the scalable demand for multi-view volumetric video. To overcome these shortcomings, the visual volumetric video-based coding (V3C) [150] includes a group of standards (ISO/IEC 23090-xx) for encoding, storing and transporting volumetric visual content. Several compression standards have been developed under the V3C [150], including MPEG immersive video (MIV) (ISO/IEC 23090-12) [151], video-based PCC (V-PCC) (ISO/IEC 23090-5) and geometry-based PCC (G-PCC) (ISO/IEC 23090-9) [152]. For more detailed descriptions of these codec algorithms and their coding performances, refer to [153].

E. REAL-TIME RENDERING TECHNOLOGIES

Real-time 3D and AR rendering technologies enhance the capacity to stream VR games or other virtual content from cloud servers, enabling high-quality scene rendering on high-performance computing clusters [154]. Three primary rendering schemes have been proposed to support next-generation Metaverse applications such as VR and cloud gaming.

In the simplest form, **local rendering** implies executing the rendering process entirely on a user-end device. Commercial mobile VR systems such as Google Daydream [155] and Samsung Gear VR [156] have used local rendering to interactively render VR content through the smartphone's CPU/GPU. However, these systems struggle to maintain a satisfactory QoE for high-quality VR applications owing to their limited computational resources, which impacts the overall immersion experience [157], [158]. For example, tests on Google Daydream with seven popular VR apps in [159] revealed an intense local rendering workload for high-end mobile systems, resulting in a rendering delay of 63-111 ms, dramatically surpassing the optimal per-frame rendering interval of 16 ms.

Remote rendering uses distant servers or high-end GPUs to overcome the constraints of mobile and standalone VR systems. This ‘thin client’ model [160] enables processing user inputs on a remote server, with the output returned as a compressed video stream. One notable application of remote rendering is cloud gaming [161], where user inputs are relayed to the server, processed according to the game’s logic, and relayed back to the client as corresponding frames, facilitating the creation of boundless virtual worlds. However, shifting from local computation to a network-centric workload presents significant latency issues. High network demand can lead to delays in transferring the rendered frames and in round-trip latency from sending rendering requests to receiving the first byte of the rendered frame. Moreover, to ensure a seamless user experience, the rotational latency from head movement to the corresponding response must remain under 20ms to prevent VR simulation sickness (caused by a lag between the visual and vestibular sensory inputs) [162].

Given the latency and bandwidth challenges associated with local and remote rendering, leveraging the computational power of mobile VR hardware to handle a portion of the rendering workload near the display HMD, while delegating the remaining workload to a remote system, appears to be a viable solution. Recent studies [157], [159], [163] have suggested some **collaborative rendering** schemes that utilize the computational capabilities of mobile VR hardware to handle a part of the time-sensitive rendering workload while allocating the remainder to a remote system. The basic principle of these schemes is to render lightweight, interactive foreground objects locally, and offload a heavier background environment to a remote server. While these systems enable pre-rendering and prefetching of the background environment to optimize network latency and bandwidth utilization, they overlook several key factors, such as the real-time processing capabilities of varying mobile VR hardware, fluctuating rendering workloads due to real-time user inputs, and changing network conditions. This is discussed in detail in Section IV-B.

F. SIMULATION OF COMPLEX PHYSICAL SYSTEMS

The integration of simulation technologies into Metaverse systems has been proposed to enable the real-time modeling of virtual realms that closely mirror the physical world. This degree of realism is supposed to enrich user experience and facilitate service offerings. At the core of these technologies are Digital Twins [164]. Digital Twins are much more than mere replication. They leverage the interplay of data between physical and digital entities, promoting self-adaptation and self-learning in Metaverse systems. They can extend their functionality to allow predictive maintenance and accident traceability, thereby enhancing efficiency and minimizing risks in the real world. For example, Omniverse (a digital twin system) has been implemented in physical automotive factories [165], [166], with bidirectional communications between the physical entities and their digital twins. BMW’s digital twin system [165] enables the exploration of various factory automation configurations to optimize manufacturing workflows. Developers can use Metaverse systems to quickly mock up prototype designs without physical hardware constraints when designing infrastructures. To test these designs, developers use Metaverse systems to create physical ‘unit tests’ and other testing frameworks. This involves simulating diverse environmental scenarios and exposing the device to unexpected situations to ensure that the device will perform as expected. Operators can then deploy their pre-tested implementations and finely-tuned configurations from the cloud to the physical environment. Digital twins are supposed to represent real-world conditions in real time. For example, in time-critical healthcare applications such as remote surgery [167], digital twins must provide real-time updates and timely receptions of feedback to facilitate timely decision optimizations. However, maintaining synchronization requires transmitting high volumes of multidimensional data, including large-capacity content such as video and 3D computer graphics and haptic signals such as touch sensations. To achieve this level of synchronization, the system demands data transmission rates exceeding 100 Gbps, reliability greater than 99.99999%, and latency under 1 ms [168]. These requirements pose challenges, as they exceed the capabilities of current networking technologies.

G. AI

AI is often proposed as a tool to enhance Metaverse experiences. Examples are given below. *I) Massive Metaverse Scene Creation and Automating Content Generation.* For instance, GANverse3D [169] is an AI-based framework that allows content makers to photograph a physical object and create a virtual copy with lights, physics models, and PBR materials. *II) Personalizing Metaverse Services (e.g., live and custom avatar creation).* For instance, Epic Games’ MetaHuman [14] illustrates the ability of machine learning to produce lifelike digital characters to fill a Metaverse system as a conversation virtual assistant. *III) Analyzing User Behavior.* AI can enhance brain-computer interfaces by interpreting their signals, enabling the processing of brain signals for more complex tasks by facilitating gesture control in virtual environments [170]. A real-world example is Neuralink’s experiment [170] where a monkey played Ping-Pong through brain signals. AI also allows intelligent interactions (e.g., smart shopping guide and user movement prediction) between the user and avatar/NPC (non-player character) through intelligent decision-making, creating personalized avatars, and intelligently recommending interested goods or information to users by continuously learning users’ facial expressions, emotions, etc. *IV) Optimizing Communications and Computing.* AI can enable efficient communication resource management in dynamic and complex network environments. For example, Xu et al. [17] used AI to improve classic optimization tools by improving auction convergence and reducing communication costs for service pricing by physical service providers. In media encoding and rendering, Neural Radiation Field (NeRF) [171] is an AI-based solution to encode the radiation field of a 3D scene in an MLP network. It takes continuous 5D coordinates as input and predicts the volumetric density and view-dependent emitted radiance at the input spatial location, allowing real-time rendering of high-fidelity static and dynamic 3D scenes.

H. TAKE-AWAYS

Metaverse systems sometimes use new media types such as 360-degree video and free viewpoint television. This can include holography or volumetric video to support realistic virtual experiences. These media types (encodings) demand higher bandwidth than conventional 2D video standards such as AVC/H.264 and HEVC/H.265. It can be challenging to transmit these, especially with low latency. Tiling can reduce data volume and improve encoding flexibility in quality management by selecting only the tiles. In conjunction with information-centric tile access, it can also provide additional efficiency through object sharing However, while increasing the use of tiling can enhance flexibility, it can also reduce efficiency. Maximum tiling is not always beneficial as it may not lead to significant bandwidth savings only by itself. There is a trade-off in designing tiling schemes that balance flexibility with efficiency.

The role of rendering technologies in Metaverse systems cannot be understated. High-quality graphics rendering today is typically done on users end devices. However, it is expensive for certain types of devices. Remote rendering offloads most graphic processing to servers, alleviating the load on local devices but requiring higher available bandwidth (e.g., exceeding 25 Mbps for cloud gaming [172]) between servers and end devices) and robust server capabilities to avoid potential server and network overload which need to manage all these things (as discussed in Section IV-B)). Continued research and improvement for rendering technologies and networks are required. Future Metaverse system should include well-structured graphics pipelines and be modular and flexible, allowing for flexible decomposition and function offloading based on current load and utilization, for example, providing only the necessary data elements for rendering at different quality levels and potentially using dynamic transcoding and level-of-detail support. We discuss these concepts in detail in Section IX-C.

IV. METAVERSE SYSTEM ARCHITECTURES

In this Section, we analyze the design and architecture of Metaverse systems and explain how technologies are integrated into current systems. In Section IV-A, we provide an overview of typical Metaverse system design. In Section IV-B, we present an in-depth overview of the typical system architecture of VR systems, including their rendering workflow and graphics pipeline. Finally, Section IV-C discusses typical VR systems in individual components and performs a detailed analysis of two typical social VR platforms: Meta's Horizon Workrooms and Microsoft's AltspaceVR.

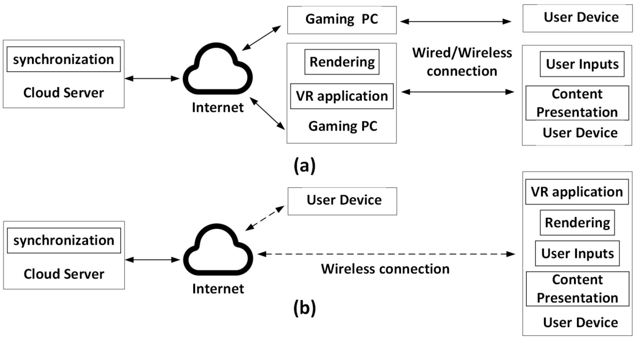

FIGURE 4. Typical System Architectures [155], [156], [173], [174], [175].

A. VR METAVERSE SYSTEMS DESIGN

Figure 4 depicts three configurations of contemporary VR systems and shows how they distribute VR tasks among their components.

Tethered Device: As illustrated in Figure 4(a), this system incorporates a high-performance Personal Computer (PC) linked to a lower-performance wearable device via wired or wireless connections. Notable instances of such systems include the Meta Quest 3 and HTC Vive [176]. Meta Quest 3 tracks the headset orientation, while HTC Vive employs room sensors for three-dimensional motion tracking. With regard to limitations, wired connections can constrain user mobility, pose a tripping hazard, and potentially diminish the immersive VR experience.

Cloud gaming services such as Google Stadia [177], Steam Remote Play [178], [179], and NVIDIA's GeForce Now [180] use cloud-based GPUs for game rendering, offering a solution to hardware limitations on user devices by processing game graphics on remote servers. These platforms employ advanced rendering techniques, including post-render warp, to enhance gaming experience. For instance, Kim et al. [181] discussed how post-render warp with late input sampling could mitigate up to 80% of the latency penalty in controlled environments. Chen et al. [182] proposed a client-end GPU-accelerated scene warping technique, estimating rendering frames between keyframes to hide interaction delays and enhancing user experience in wireless links between the server and client.

Untethered Device without A Host PC: This VR system configuration, represented in Figure 4(b), primarily relies on MUDs for processing (except for social synchronization tasks or cloud-based rendering that occur on the Internet). Devices include smartphone-powered headsets and standalone VR devices. Smartphone-powered headsets (e.g., Google Daydream [155], Samsung Gear VR [156]) leverage phones' computational capabilities to render and display images, and use their built-in accelerometers for motion tracking. However, the hardware constrains the quality (i.e., sophisticated VR experiences can be limited on lower end devices). Standalone VR devices (e.g., Oculus Go [173], Oculus Quest [183], Lenovo Mirage Solo [184], and Apple Vision Pro [185]) function better. These devices are equipped with screens, sensors, and mobile-phone-grade processors. For example, Oculus Quest 2 can offer up to around 2K content resolution and 60–70Hz refresh rate, and Apple Vision Pro can attain higher refresh rates (about 90-100Hz) and exceed 4K resolution by connecting to a high-end GPU-equipped PC for rendering [185].

Untethered Device with A Host PC: This emerging VR system renders visuals on a separate, high-performance computer, and streams them to a wearable device via a fast wireless network. For example, Furion [159] is a mobile VR framework that separates the rendering pipeline for foreground interactions and the background environment between a client and a server. DeepMix [186] is a lightweight framework for MR headsets that combines edge-assisted 2D object detection with on-device estimations of 3D bounding boxes for real-world objects. Several startup companies and research groups, such as Amimon, KwikVR, and TPCAST, are developing Untethered VR (UVR) systems using 60 GHz mmWave wireless networks. However, these systems have several challenges owing to the inherent characteristics of mmWave technologies, including high signal attenuation and transmission beam alignment issues. Solutions such as MoVR [174], [175] aim to address these limitations through carefully positioned antennas and mirrors, yet they require specialized environments. Other research studies [187], [188] explore 60 GHz wireless networks and rendering pipelines between a host PC and a receiver. However, these systems require high decoding rates and severely limited range. Furthermore, they rely on laptop PCs, which often fall short of the power, weight, and budget constraints of an MUD. GamingAnywhere [189], an open-source game streaming system, works directly with an IEEE 802.11ac wireless network. Furion [159] transmits content from a host PC using IEEE 802.11ac WiFi. By parallelizing the video codec and segregating the background/foreground rendering, Furion minimizes the latency. However, these two systems require computing resources on the host PC and MUD and a mobile phone as an MUD.

Summary: Each VR system type has its own advantages and limitations. Tethered devices offer high processing power but limit user mobility owing to their physical connection. Untethered devices without a host PC provide user mobility but are restrained by their own processing capabilities or latency issues with cloud servers. Untethered devices with a host PC exhibit promise for achieving high-quality VR experiences but face challenges in wireless technologies and processing requirements.

B. RENDERING WORKFLOW AND GRAPHICS PIPELINE

A key element of an immersive mobile VR experience is the ability to render high-quality content with low latency, which is addressed by local, remote, or collaborative rendering.

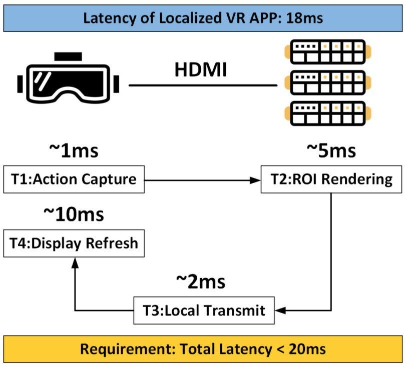

FIGURE 5. An Example of Local Rendering Workflow [183].

Local rendering workflow: In some instances, VR headsets depend solely on their built-in computational resources (e.g., CPU and GPU) to facilitate the rendering pipeline. However, untethered headsets such as Oculus Quest 2 only offer a maximum of 2K content resolution and around 60–70Hz refresh rate [183]. In contrast, tethered VR headsets such as Apple Vision Pro can attain higher refresh rates (90-100Hz) and exceed 4K resolution by connecting to a high-end GPU-equipped PC for rendering [185]. Uncompressed, full-resolution immersive VR video data are initially transmitted to these headsets, with viewport rendering subsequently conducted locally (Figure 5). To achieve a high-quality VR experience with local rendering, end devices typically need to be cable-connected (e.g., Oculus Rift [190], HTC Vive [58]), pairing with a PC that operates a powerful graphics card. While this setup is ideal for rendering high-quality graphics and minimizing latency, it increases costs for users and hampers mobility due to the physical tethering to a PC [2], [191].

Remote rendering workflow: Cloud and edge computing have prompted VR systems to shift from exclusively local rendering towards a hybrid approach that offloads most rendering tasks to remote servers [159], [187], [192]. After rendering, servers return the encoded frames to VR headsets as a video stream for the display. An example of this remote rendering workflow is presented in [124] (Figure 6). A server dynamically renders VR content and sends a partial panorama to a client device, which then performs local rendering at 60 Hz to adjust to any head movement. This approach can provide variable motion-to-photon latency based on user interaction types, allowing the system to effectively adapt to different user interaction modalities while continuously evolving VR scenes. Currently, remote rendering has been implemented in cloud gaming platforms [172], [177], [193] and video streaming applications [194], [195]. However, it requires a higher available bandwidth than local rendering (e.g., exceeding 25 Mbps for cloud gaming [172]) between servers and end devices to maintain a seamless experience.

Collaborative VR Rendering Workflow: Collaborative VR rendering workflow [157], [159], [163], [192] uses the

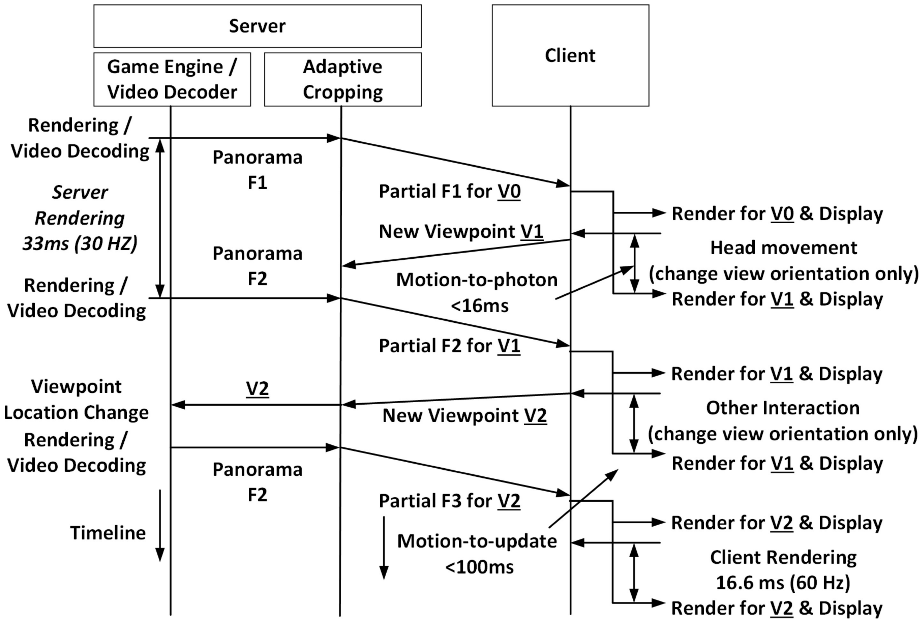

FIGURE 6. An Example of Remote Rendering Workflow [124].

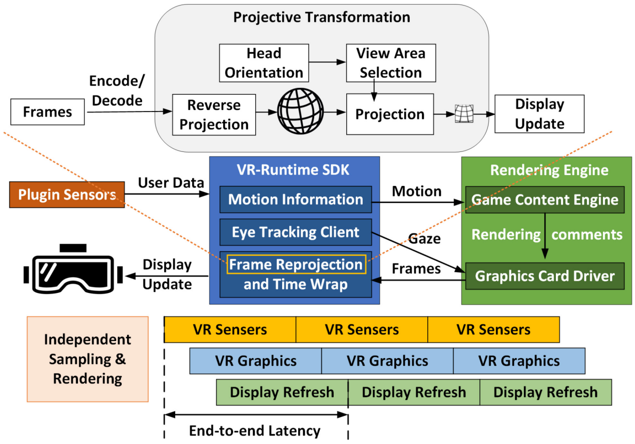

FIGURE 7. An Example of A Modern VR Graphics Pipeline [196], [197], [198].

processing power of mobile VR hardware to manage time-sensitive rendering tasks close to an HMD display while offloading the rest to a remote system. This approach relies on the fact that pre-defined interactive objects are typically less resource-intensive than the background environment, making it more efficient to render these interactive elements locally while assigning a more complex background environment to the remote server. To further reduce network latency, these schemes also incorporate pre-rendering and pre-fetching for the background environment [196].

The overall end-to-end latency of a VR system is determined by the collective performance of each stage within the graphics pipeline **graphics pipeline** (Figure 7). This pipeline typically comprises three stages. Each stage operates relatively independently and has specific requirements that impact the total system efficiency (see Section V-A).

I) User input collection stage involves the collection of various user inputs, ranging from explicit commands sent through devices (e.g., keyboards or controllers) to physical motion data recorded by sensors. They are typically captured using specialized cameras [199], [200] and are represented in a spherical format (to form VR videos). These spherical images are then projected onto planar frames using techniques such as equirectangular projection, facilitating subsequent processing [198].

II) Content generation stage processes the user inputs and selects VR content elements to produces content according to application requirements. This includes date processing and retrieval from local storage or cloud services. Initially, the system processes user inputs and eye-tracking information through the VR runtime. Then planar VR frames are re-projected back to their original spherical format. The client player then creates a spherical area aligned with the user's viewing angle, dictated by head orientation. After data processing, the system creates two planar frames, one for each eye, establishing a stereoscopic view that generates a three-dimensional perception within the VR environment.

III) Content presentation stage presents the VR content to the user, primarily through video frames and audio effects, and consists of two sub-tasks: *a) Asynchronous Time Warp (ATW).* Before displaying the rendered frames on the HMD, the system uses ATW to adjust the 2D image plane according to the lens distortion, making the VR content more natural to the user's vision [190], [201]. *b) Immersive Perception Creation.* To create an immersive experience, the system generates high-quality images, sounds, and other stimuli that simulate the sensation of physical presence within a virtual world [202], [203], [204].

C. CASE STUDY: DISSECTING TYPICAL VR SYSTEMS

In this subsection, we examine a particular aspect of Metaverse systems – Social VR, a fusion of online social networks and VR technologies (i.e., 'social life' application type in Section II-B). Social VR enables users to interact in a virtual world as avatars, facilitating communication and collaboration akin to physical world interactions [2]. Primary features include navigating and conversing in a virtual space, such as a conference room. Advanced features include a deeper level of interaction with the platform and other users, e.g., playing games, generating user content, and conducting transactions using non-fungible tokens (NFTs) [205]. Existing social VR platforms share some similarities. Users typically encounter a welcome page for system initialization when launching an application. Subsequently, they can select their desired social interaction or public events such as concerts and online meetings. Commercial social VR platforms include VRChat,[13] Rec Room,[14] AltspaceVR, Mozilla Hubs,[15] Anyland,[16] Cluster,[17] Bigscreen,[18] and Workrooms. For an in-depth comparison of these platforms, please refer to [2]. By examining how these platforms operate, we can gain a broader understanding of Metaverse systems reality, concentrating on its underlying network protocols and identifying areas for improvement. In this subsection, we discuss the following three questions:

- *How do typical social VR systems such as Workrooms and AltspaceVR operate?*
- *Which type of content is delivered in Workrooms and AltspaceVR?*
- *Which network protocols and infrastructure do social VR systems use?*

1) HOW DO TYPICAL SOCIAL VR SYSTEMS OPERATE?

Figure 8 depicts the connection establishment and data exchange process between clients and servers in Workrooms (top) and AltspaceVR (bottom). In Workrooms, the system hinges on two main servers. One (called server I or UDP server) is for content, and the other (called Server II or WebRTC server) is for streaming and exchanging audio-visual data. The Workrooms process can be conceptually divided into two stages. The entire operation starts with an *initialization* process, that involves a local setup and rendering that consumes approximately 25 seconds with minimal network activity. During *initialization*, the background loading and connection occur establishment predominantly. When users enter a meeting platform and the loading interface becomes visible, the system commences a UDP session with Server I. Following *initialization*, Server I maintains an exchange of 'virtual content' over UDP, a task that continues throughout the *communication* stage, providing a continuous stream of virtual content. After *initialization*, users transition to the *communication* stage, which is marked by entering the corresponding meeting room and establishing a connection with Server II. A key point is the divergence in the connection method between PC and headset clients with Server II. Despite establishing an initial TCP connection and using Session Traversal Utilities for NAT (STUN) protocol for NAT traversal, the headset client continues with the TCP connection, transferring 1-3 Transport Layer Security (TLS)packets to each other. In contrast, the browser client transitions to a Datagram Transport Layer Security (DTLS) connection with Server II over the UDP after establishing the initial TCP connection. During this communication period, Server II uses the UDP flow to transmit the virtual background and avatar-related content. Concurrently, it employs the Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) to exchange multimedia content with the server. RTP is typically used to transmit multimedia streams, such as audio and video, whereas RTCP monitors data delivery.

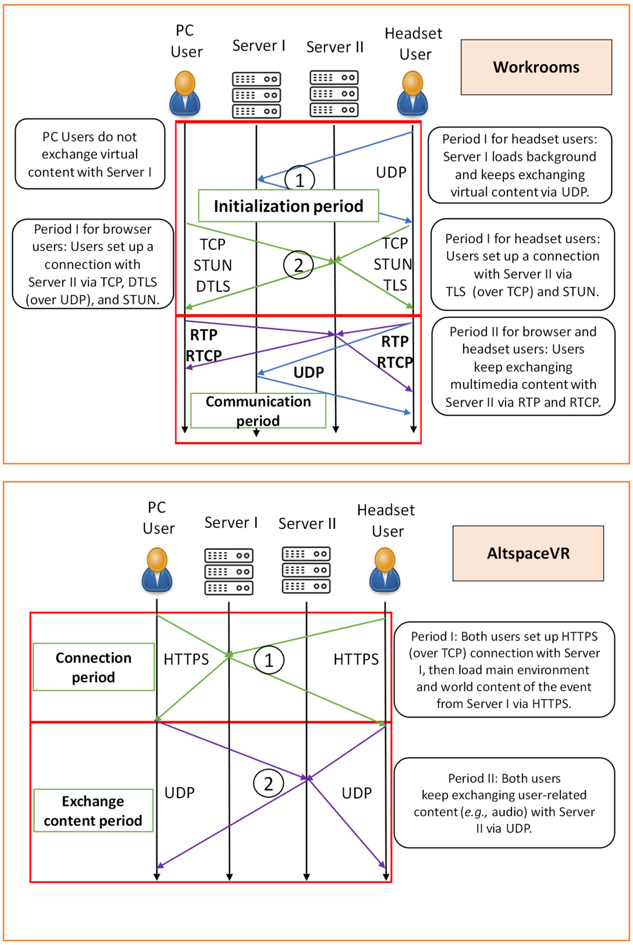
PC User
Server I
Server II
Headset User
Workrooms
PC Users do not exchange virtual content with Server I
Period I for browser users: Users set up a connection with Server II via TCP, DTLS (over UDP), and STUN.
Period I for headset users: Server I loads background and keeps exchanging virtual content via UDP.
Period I for headset users: Users set up a connection with Server II via TLS (over TCP) and STUN.
Period II for browser and headset users: Users keep exchanging multimedia content with Server II via RTP and RTCP.
1
UDP
Initialization period
2
TCP
STUN
DTLS
TCP
STUN
TLS
RTP
RTCP
UDP
RTP
RTCP
Communication period
PC User
Server I
Server II
Headset User
AltspaceVR
Connection period
HTTPS
1
HTTPS
Period I: Both users set up HTTPS (over TCP) connection with Server I, then load main environment and world content of the event from Server I via HTTPS.
Exchange content period
UDP
2
UDP
Period II: Both users keep exchanging user-related content (e.g., audio) with Server II via UDP.

FIGURE 8. The Process of Establishing Connections and Exchanging Data Between the Clients and The Servers for Workrooms (top) and AltspaceVR (bottom).

In contrast to Workrooms, AltspaceVR employs a slightly different approach to client-server connections. The client

[13] https://hello.vrchat.com/; accessed on 25-Aug.- 22

[14] https://recroom.com/; accessed on 25-Aug.-22

[15] (https://hubs.mozilla.com/; accessed on 25-Aug.- 22)

[16] http://anyland.com/; accessed on 25-Aug.-22

[17] https://cluster.mu/en/; accessed on 25-Aug.-22

[18] https:// www.bigscreenvr.com/; accessed on 25-Aug.-22

begins by downloading data from Server I via HTTPS, which is typically 10-20MB and is required only for first-time event participants. Following this, an additional download of 300-500KB of data from Server I occurs via another HTTPS connection, this time for loading the main environment. The connection to Server II is only established once the loading is complete, and from this point all data exchanges occur over UDP. Both users continually exchange user-related content, such as audio, with Server II through this UDP connection.

2) WHICH TYPE OF CONTENT IS DELIVERED?

Workrooms and AltspaceVR differ in the types of content they deliver and how they deliver it. In-depth experiments and subsequent data analyses conducted on Workrooms and AltspaceVR [2], [112], [113] have led to several discoveries regarding their data transmission modalities. From a network protocol perspective, Workrooms uses HTTP3 (and QUIC) in Server I for reliable virtual content delivery [206]. When no other functions are active, this virtual content comprises two primary components: a virtual background, which requires approximately 0.1 Mbps of data, and user-specific avatar-related content, requiring an additional 0.5 Mbps per user. However, Workrooms merely forward avatar-related virtual content to each user, without any further processing, raising potential scalability issues (it supports only up to 16 headset users). If 16 headset users are simultaneously active, each user's data requirements increase to approximately 8 Mbps. Moreover, Workrooms does not consider situations where data forwarding is not required (e.g., when a user is browser-based), thus adding to the bandwidth overhead.

Video and audio communication in Workrooms *(only through Server II)* relies on RTP flows. Unlike the 'UDP flow'(*in Server I*), this RTP flow primarily involves downlink data transmission, averaging between 1 and 2 Mbps. The uplink bitrate, used primarily for audio session exchanges between users, is relatively lower, often under 0.05 Mbps. Notably, audio sessions are server-forwarded between users, and video sessions are server-directed to users. However, Workrooms' current model fails to optimize audio sessions via peer-to-peer communication, even for users in the same subnet, resulting in an additional bandwidth overhead. AltspaceVR uses a proprietary UDP-based protocol [2], where the fourth byte of the UDP payload differentiates between various data types, such as audio data or user-related content. In summary, although Workrooms and AltspaceVR deliver similar types of content (i.e., avatar-related virtual content, background scenery, and real-time audio-video sessions), they differ in their delivery mechanisms, highlighting the diversity and potential for optimization in social VR system designs.

3) WHICH NETWORK PROTOCOLS AND INFRASTRUCTURE DO SOCIAL VR SYSTEMS USE?

Looking beyond Workrooms and AltspaceVR, we provide a broader overview of the network protocols and infrastructure commonly implemented by social VR systems, as summarized in Table 4. A variety of **protocols** are deployed and customized for different goals. For example, protocols for connection establishment over middleboxes (e.g., interactive Connectivity Establishment (*ICE*) [207] and Session Traversal Utilities for NAT (*STUN*) [208]), transport protocols (e.g., *TCP*, *QUIC*, *RTP/RTCP*) for transferring both content and control components (e.g., downloading scene descriptions and distributing events), *TLS* or *DTLS* for authentication, integrity, and encryption, and application layer protocols (e.g., *HTTP and proprietary protocols*) for fetching Web resources and controlling. These protocols run over the Internet and thus have to provide typical features such as congestion control (for TCP/QUIC) and flow fairness. Most systems (such as Mozilla Hubs, Horizon Worlds, and Spatial [3], [209], [210]) use a collection of existing protocols, sometimes with additional proprietary protocols (such as AltspaceVR [211]), which can lead to unnecessary overhead and sub-optimal performance [2], [112], [113].

Based on [2], [112], [113], we describe the network protocols used by these systems as operating across two distinct channels: *control* and *data*. The former pertains to menu operations and game clock synchronization, whereas the latter handles avatar embodiment and voice data. For the *control channel*, all surveyed systems, including RecRoom, VRChat, Mozilla Hubs, Horizon Worlds, and AltspaceVR, use HTTPS for control communication, such as menu selections (but only Workrooms uses HTTP3). AltspaceVR and Worlds exhibit recurring HTTPS traffic spikes approximately every 10 seconds, primarily used to report client-side information and synchronize in-game clocks across users. The *data channel* primarily manages data plane information such as audio content and avatar motion. For example, Hubs and Spatial [3], [209], [210] use WebRTC [212] for multimedia communication (enabling audio and video streams) among VR users.

As for virtual backgrounds, all five platforms offer only static virtual backgrounds, which only need to be downloaded once. Although using static backgrounds minimizes the rendering burden and communication overhead, it also limits user interactions with the virtual environment [31]. The downloading of these static virtual backgrounds also varies across platforms. For instance, *AltspaceVR* and *VRChat* download 10–30 MB of data during initialization. *Rec Room* pre-downloads the background during the application installation, as evidenced by the sizable app footprint in the Oculus store (1.41 GB). *Worlds* downloads about 5 MB of data during the 'Preparing for Visitors' phase every time users launch the application, whereas *Hubs*, being browser-based and lacking an installed application, requires users to download about 20 MB of data each time they join the platform. In terms of infrastructure, the servers that handle these two channels may have different owners and geolocations, as presented in Table 4. For example, the official documentation of Hubs reveals that its HTTPS server operates as a set of load-balanced nodes, whereas its

TABLE 4. Network protocols and infrastructure of five social VR platforms [2], [112], [113].

Platform	Welcome Page (Single User)	Social Event		Max Users in One Room	Server Owner
		Virtual Background	User Interaction		
RecRoom	HTTPS	Pre-download in the App	UDP	40	ANS/ Cloudflare
VRChat	HTTPS	HTTPS Downloading (first joining)	UDP	40	AWS/ Cloudflare
Mozilla Hubs	HTTPS	HTTPS (everytime)	Audio: RTP/RTCP (over UDP) Others: HTTPS	24	AWS
Horizon Worlds	HTTPS	Pre-download in the App	UDP	20	Meta
AltspaceVR	HTTPS	HTTPS Downloading (first joining)	UDP	50	Microsoft

WebRTC server serves as a central routing machine [209]. Further details on this topic can be found in [2], [112], [113].

D. TAKE-AWAYS:

One challenge in today's Metaverse systems is using different, uncoordinated protocol sessions, each with its congestion control mechanism. For example, WebRTC is typically used for live media, while DASH is preferred for video streaming, alongside other protocols (e.g., HTTP or proprietary ones) for control communication or general data exchange. This diversity results in complex protocol stacks [2], [112], [113] and varying security frameworks (e.g., TLS over HTTP/HTTPS for DASH and DTLS for WebRTC) and can also lead to uncoordinated and inefficient resource allocation [213]. The challenge lies in managing these protocols efficiently within Metaverse systems, e.g., in one QUIC connection.

V. COMMUNICATION PERFORMANCE

Balancing low latency and high throughput requirements, particularly with a high peak to mean bit-rate ratio, is challenging. Metaverse relies heavily on high-speed communication to address the diverse needs of Metaverse applications.

A. GAP ANALYSIS

1) LATENCY

Motion-to-Photon (MTP) latency is an important parameter for determining the Quality of Experience (QoE), which refers to the overall performance and satisfaction perceived by the user during the interaction with an AR/VR system [214]. This term refers to the duration from the instance of user movement to the corresponding visual update on the display screen, denoted by:

$$\tau_{\text{MTP}} = \tau_{\text{camera}} + \tau_{\text{processing,device}} + \tau_{\text{display}}$$

Where τ_{camera} is the time taken by the camera to capture and transmit the user's movement data, $\tau_{\text{processing,device}}$ is the time required for the device to process the input data and generate the appropriate output, and τ_{display} is the time for the updated visuals to be rendered and shown on the display screen.

Studies [215], [216] have shown that the optimal latency range for immersive experiences is 7-15ms, with a maximum threshold of 20ms and the potential to reach as low as 1ms for tactile applications like remote surgery. As detailed in Raaen's thesis [215], exceeding these stringent latency requirements leads to delays responses and causes other symptoms, such as headaches, nausea, and disorientation, in VR applications. Another important parameter is the jitter, also called latency variation. Fluctuations in latency lead to stuttering and choppiness in the display, drastically affecting the perceived Quality of Experience. Jitter becomes problematic when it causes frame skipping. The maximum tolerable jitter for a 30 FPS video is approximately 33 ms [217]. Currently, most localized Metaverse applications have managed to achieve sub-20 ms MTP latency and acceptable jitter owing to technological advancements in sensor detection, display refresh, and GPU processing, while network-based applications are still in their early research stages, far from the satisfied threadshot, with usually 130 - 1118 ms MTP, as shown in Table 5 [183].

Taking VR as an example, in a typical network-based AR/VR system, the MTP latency is shaped by several stages, i.e., sensor detection and action capture (around 1ms), computation for Region of Interest (ROI) processing, rendering and encoding (around 11ms), Group of Pictures (GOP) framing and streaming (110ms-1s), network transport(0.2-100ms), terminal decoding(around 5ms), and screen refresh(around 1ms), as illustrated in Figure 9. The MTP latency τ is the sum of these components.

$$\tau_{\text{MTP}} = \tau_{\text{camera}} + \tau_{\text{network, up}} + \tau_{\text{processing, server}} + \tau_{\text{network, down}} + \tau_{\text{display}}$$

Where τ_{camera} is the time taken for sensor detection and action capture, $\tau_{\text{network, up}}$ is the time for the data to be sent from the device to the server, including any time spent in queues or processing for transmission, $\tau_{\text{processing, server}}$ is the time taken for computation for ROI processing, rendering, encoding, and GOP framing and streaming at the server, $\tau_{\text{network, down}}$ is the time for the processed data to be sent back from the server to the user's device, and τ_{display} is the combined time for terminal decoding and screen refresh on the user's device.

TABLE 5. Current and projected latency in key stages in network based AR/VR (MTP = T1+T2+T3+T4+T5+T6) [183].

Latency	Current Value (ms)	Projected Value (ms)
T1	1	1
T2	11	2
T3	110 to 1000	5
T4	0.2 to 100	?
T5	5	5
T6	1	0.01
MTP	130 to 1118	13+?

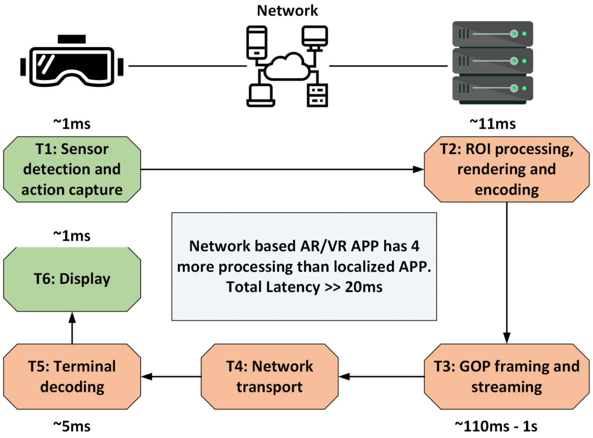
Network
~1ms
~11ms
T1: Sensor detection and action capture
T2: ROI processing, rendering and encoding
~1ms
T6: Display
Network based AR/VR APP has 4 more processing than localized APP. Total Latency >> 20ms
T5: Terminal decoding
T4: Network transport
T3: GOP framing and streaming
~5ms
~110ms - 1s

FIGURE 9. Latency Requirements [183].

Specifically, the network delay, represented by τ_{network}, comprises the physical propagation delay and switching/forwarding delay. The former, in the context of fiber optics, is defined by the speed at which light travels through the fiber, which is approximately 200km/ms. The latter typically exceeds the physical propagation delay and can vary significantly, ranging from 200μs to 200ms per hop. This network delay must strike a balance with the on-device processing time to minimize the overall MTP latency. As outlined in Table 5 [183], the upper limit of GOP framing and streaming ($T3 : \tau_{\text{processing}}$) can be potentially reduced to around 5 ms by improved parallel hardware processing; display response time ($T6 : \tau_{\text{display}}$) can be reduced to around 0.1 microseconds through the use of Organic Light Emitting Diode (OLED) displays. Given the total 20 ms MTP latency budget, a mere around 7-8ms remains available for graphic rendering and the RTT between the AR/VR device and the cloud/edge ($T4 : \tau_{\text{processing, server}} + \tau_{\text{network}}$), consumed by rendering, propagation, switching, and queuing delays.

2) THROUGHPUT

Throughput is another critical parameter for the quality of Metaverse applications. Beyond mere display functionalities, Metaverse systems require substantial real-time computations that are dependent on consistently high throughput. This includes processing inputs from sensors (e.g., for tracking user movement), rendering high-resolution graphics and conducting physics simulations for real-world phenomena. These systems predominantly operate in a streaming fashion, with computation requests arriving frame by frame. This real-time requirement implies that if a frame's processing is delayed, the system may face increased latency or risk of compromising the quality and accuracy of the output. The immediacy of such tasks requires very high throughput. For example, a frame in a 30 FPS system necessitates processing within around 33 ms [218]. Table 6 outlines different Metaverse applications and their respective overall throughput requirements [219]. As illustrated, applications such as 360-degree video and 6DOF video transmit data at a higher rate, up to 100 or even 1000 times more, compared to conventional image or video transfers.

Specifically, the throughput in Metaverse systems can be differentiated into *average throughput* and *peak throughput*. *Average throughput* measures the mean data transfer rate over a specific period. Several parameters influence this bit rate, including the display resolution, dimensionality (2D or 3D), view type (normal or panorama), video processing codec type, color space, sampling algorithm, and video pattern. Metaverse applications can produce extensive live data streams such as 360-degree panoramas and 3D volumetric content. Normally, 3D and panorama view video bit rates are approximately 1.5 times and 4 times greater than 2D and normal views, respectively. Furthermore, the bit rate is influenced by the video pattern and motion rank. When the changes in the video frames are more frequent, the achieved data compression is less. A compressed video stream comprises ordered successive groups of frames or a GOP. These GOPs consist of three types of pictures (or frames) [220]: I-frames (fully specified picture), P-frames (only changes from the previous frame), and B-frames (content specified by differences between the current frame and both the preceding and following frames). Each of these frames performs a role in video compression and contributes to improved video compression rates.

Peak throughput in Metaverse systems refers to the highest bit rate required for smooth transmission of video content. More specifically (Table 7), the 'peak bit rate' is the bit rate necessary for I-frame transportation, with the 'burst size' representing the I-frame size, and 'burst time' indicating the duration needed for complete end-to-end I-frame transport, measured via frame rate (FPS) [221]. Two potential situations trigger the inception of a new I-frame. The first scenario arises from substantial AR/VR video display alterations, leaving no similarities between consecutive images. The second is when the FOV changes owing to the users' head or eyeball movements. However, I-frames present a challenge in terms of compression. With no reference frame for video compression, I-frames require intra-frame processing techniques, similar to those used in static image compression such as JPEG. This results in a lower compression ratio compared with inter-frame compression. Moreover, the delay budget of network transport requires the generation, grouping, transmission, and display of a new I-frame to occur within a specific timeframe,

TABLE 6. Throughput of some metaverse applications [219].

Application	Throughput Required
Image and Workflow Downloading	1 Mbps
Video Conferencing	2 Mbps
3D Model and Data Visualization	2 to 20 Mbps
Two way Telepresence	5 to 25 Mbps
Current-Gen 360 degree video (4K)	10 to 50 Mbps
Next-Gen 360 degree video (8K, 90+ FPS, HDR, Stereoscopic)	50 to 200 Mbps
6DoF Video or Point Cloud	200 to 1000 Mbps

TABLE 7. Bit rate for different VR systems (throughput requirements) [183].

	Entry-level VR	Advanced VR	Ultimate VR
Type	4K 2D Video	12K 2D Video	24K 3D Video
Resolution of 360 degree video($W \times H$)	3840×1920	11520×5760	23040×11520
HMD Resolution/ view angle	$960 \times 960/90$	$3840 \times 3840/120$	$7680 \times 7680/120$
PPD(Pix Per Degree)	11	32	64
The average video compression ratio	120	150	[200(2D),350(3D)]
FPS	30	60	120
Mean Bit rate	22Mbps	398Mbps	2.87Gbps(2D), 3.28Gbps(3D)
The I-frame compression ratio	20	30	20(2D), 30(3D)
Peak bit rate	132Mbps	1.9Gbps	28.7Gbps(2D), 38.2Gbps(3D)
Burst size	553K byte	4.15M Byte	29.9M Byte(2D), 39.8M Byte(3D)
Burst time	33ms	17ms	8ms

causing the peak bit rate and burst size to be larger than in typical video streaming, such as Internet Protocol Television (IPTV) [222]. To illustrate the relationship between the peak bit rate and average bit rate, Table 7 shows the throughput calculations for different AR/VR systems. The peak bit rate indicates the highest bit rate needed for transporting an I-frame, with the burst size and time symbolizing the I-frame size and the complete transport time respectively (for efficient application functioning). With the average video compression ratio ranging from 100 to 200 for 2D and the I-frame compression ratio ranging from approximately 20 to 30, it is clear that the peak bit rate is approximately ten times the average bit rate.

3) QUALITY OF SERVICE (QOS)

In networked systems, ‘QoS’ is applied to describe systems that control the allocation of various resources to achieve ‘managed unfairness’ [223]. However, QoS mechanisms cannot create or increase resource capacity. Experiments have shown that QoS cannot provide better user experience when [223]: I) resources are lightly loaded (as congestion loss and queuing delays are minimal); II) resources are heavily oversubscribed (indicating a failure to deliver viable service); III) failures cause rapid state shift from lightly loaded to heavily oversubscribed (QoS schemes cannot maintain consistent service quality due to slow response times, or overly conservative resource allocations that result in wasted capacity under non-failure conditions). Some QoS schemes such as the Resource Reservation Protocol (RSVP) [224], require routers to maintain state for each flow, which has scalability problems in larger deployments. Providing enough memory for core routers through which millions to billions of flows pass is infeasible. With Integrated Services (IntServ), RSVP carries two data structures (i.e., the flow specifier and the traffic specifier), identifying the QoS equivalence class and describing the traffic’s dynamic characteristics (e.g., average bandwidth and delay, peak bandwidth), respectively [224]. However, these structures limit Intserv’s application to small-scale topologies or high-value usages such as traffic engineering. With Differentiated Services (DiffServ), the protocol encoding (using six bits in the TOS field of the IP header) restricts the number of distinguishable classes [225]. However, when used with fine-grained equivalence classes, one can encounter limitations on the number of required queues.

In the context of Metaverse systems, implementing QoS encounters two additional challenges: *I) Lack of QoS Guarantees.* Despite advancements in ultra-low-latency TCP protocols, such as DETER [226] and TCP Bottleneck Bandwidth and Round-trip Propagation Time (BBR) [227], networks still lack QoS guarantees. This absence hinders deterministic behavior, even if the protocols support it. While this might be acceptable for less time-sensitive tasks such as virtual meetings, it becomes problematic for latency-critical applications such as medical or entertainment applications [228]. The impact ranges from bandwidth decrease and latency to packet loss and jitter. Given the suspicion that TCP loss recovery exacerbates latency, there is an underlying need to address latency without compromising packet loss. *II) Complex Provisioning.* The definition of high priority and reliability differs when dealing with multi-modal data in Metaverse systems, where each stream and packet within the stream have different QoS requirements. For example, a packet with I-frames from video data has higher latency and reliability requirements than packets with P- or B-frames. Therefore, a cross-layer design is needed for multimedia communication in Metaverse systems, where data encoding, compression, and communication are optimized to meet specific QoS requirements.

B. EMERGING TECHNOLOGIES

This subsection lists potential emerging technologies for addressing high throughput and low latency requirements.

1) WIRELESS LAYER TECHNOLOGIES

Developing suitable wireless technologies presents challenges across the entire wireless stack, from antennas upwards [229]. One strategy to reduce latency (achieving ultra-low latency) is to use new frequency bands. The Millimeter Wave (mmWave) band (24GHz-39GHz) supports a broad channel (up to 800MHz), providing larger throughput while minimizing latency below 1 ms. However, mmWave frequencies suffer from low range and obstacle penetration. Hence dense base station deployments in crowded environments, such as the PyeongChang Olympics in 2018 (Korea) [230] or airports, are required. This density deployment allows to serve more customers simultaneously, while maintaining high throughput and low latency in the radio access network (RAN). Massive MIMO technologies use multi-antennas to send and receive signals to improve data speed, and the beams are directed. A gNB with three cell sites can support an average of 10 Gbps in 5G when equipped with multiple antennas for transmission and reception [231]. To realize ultra-low latency at the Medium Access Control (MAC) layer, solutions such as short symbol periods, flexible Transmission Time Intervals (TTI), and low-power digital beamforming for control [232] have been proposed. Currently, the achievable one-way latency between a subscriber device and a center unit is 4 ms for eMBB and 0.5 ms for Ultra-Reliable Low Latency Communications (uRLLC) [231]. Using deterministic communications in small layer-2 networks, such as Deterministic Ethernet [233], offers a solution for achieving ultra-low latency. In these networks, multiple Ethernet transceivers can share access to an Ethernet broadcast cable through a repeated Time Division Multiple Access (TDMA) schedule. This schedule allocates coarse-grain ‘traffic-windows’ for different traffic classes, including deterministic and best-effort traffic, ensuring collision prevention through tight synchronization of all transceivers, typically within 1 μsec accuracy, using Time-Sensitive Networking (TSN) time synchronization technology [234].

2) NETWORK LAYER TECHNOLOGIES

Deterministic Networking (DetNet) is a network layer technology designed for IP/multi-protocol label switching (MPLS)-based networks [235]. It enables establishing reliable communication channels for specific traffic flows, providing deterministic QoS guarantees including ultra-low latency, jitter, congestion loss and reliability, ensuring that they remain unaffected by competing traffic flows. DetNet explicitly uses a dedicated path for each traffic flow (both synchronous and asynchronous flows) to improve deterministic latency. In synchronous flows, DetNet-enabled endpoint systems use time-synchronized clocks. For asynchronous flows, bandwidth for DetNet flows is reserved by leveraging the maximum packet size, the number of transmissions during the observation interval, and protocol stack information. This approach ensures deterministic data delivery, processing, and synchronization, supporting deterministic multimedia stream operation and management in Metaverse systems. For example, DetNet can provide time synchronization and gating mechanisms for transmitting strict and time-bound time-sensitive traffic in industrial automation, including sensor data, control inputs to actuators, and audio or video packets [236]. Currently, DetNet is still under development [237], [238] and can not be used in Metaverse systems over the Internet.

Application-aware Networking (APN) [239] is a solution designed to extend and enhance the principles of QoS by enabling fine-granularity network service provisioning (i.e., traffic operations) within the network. It allows applications to specify their requirements and enhances network functionality by applying specific policies to traffic flows as they enter the network infrastructure. In modern networks, where deterministic networking and networking slicing are required, there is a need for more functionality than what QoS can provide.

3) TRANSPORT LAYER TECHNOLOGIES

Modern congestion control has been studied for several decades because of its complexity and significance. Currently, most TCP flows on the Internet adopt the Cubic congestion control algorithm [240], primarily because it is the default in most operating systems. BBR [241] is a more recent and promising proposal for Internet congestion control, designed for both TCP and Quick UDP Internet Connections (QUIC). BBR combines loss- and delay-based congestion control mechanisms within a sophisticated internal model [242]. However, no single congestion control algorithm can be universally applied (i.e., fits all applications), and traditional algorithms face challenges in handling delay-sensitive multimedia streaming within Metaverse systems. Low Latency, Low Loss, and Scalable Throughput (L4S) protocol [243] is designed to ensure low delay for IP traffic. It uses an Explicit Congestion Notification (ECN) scheme [244] and is based on the idea of marking packets as congestion experienced (CE) when the queue delay at a network node begins to exceed a set threshold. By adjusting the rate in response to congestion signals, the protocol maintains low queue delays (and consequently lower end-to-end delay). Implementing scalable congestion control algorithms also allows for high throughput and link utilization, even with many signaled congestion events [243]. L4S is ideal for applications demanding high data rates, consistently ultra-low latency, and near-zero packet loss, such as cloud gaming, VR/AR applications, and high-quality video conferencing. However, there are *challenges* when implementing L4S in specific network architectures such as 5G networks. These include the impact of tunnels (GTP-U and IPSec) that can hinder the propagation of congestion signals, and issues related to the RLC layer, where queued packets are already encrypted, making it impossible to mark packets at the network nodes [243].

According to [245], congestion control algorithms can be generally classified into three approaches: traffic-based, resource-based, and hybrid-based.

I) Traffic-based congestion control protocols focus on managing the flow of data packets based on current network traffic conditions. They are predominantly used in scenarios requiring end-to-end and hop-by-hop traffic management. Such protocols are important in multimedia environments that demand real-time interaction, which is highly relevant for Metaverse applications such as social VR platforms, live video streaming, and VR gaming. For example, the Buffer Occupancy-based Transport Protocol (BOTCP) [246] has been developed for congestion detection. It employs two distinct thresholds to compute buffer occupancy at each node. As soon as the buffer occupancy exceeds the top threshold, congestion is identified at once, and the child nodes decrease the traffic rate to reduce the congestion. However, it cannot choose the optimal path. The Adaptive Weight Firefly (AWF) algorithm [247] reduces congestion by combining the two algorithms and only uses rate control for negative acknowledgments. However, acknowledgment of this algorithm increases the delay in the network. Traffic-Aware Congestion Control (TACC) [248] addresses rate adaptation in the transport layer and detects congestion via burst loss information by adjusting the sending rate based on congestion severity. However, the TACC must be further enhanced to facilitate the conveyance of prioritized events to different flows.

II) Resource-based congestion control protocols manage congestion by leveraging available network resources or identifying non-congested paths. These protocols are mainly used in high-reliability applications that require minimal time delays, such as remote surgery and real-time multiplayer gaming. By using idle network resources or alternative routes, these protocols enhance the probability of error-free communications. However, these protocols require additional overhead, including end-to-end topology information, loop avoidance, and packet travel time for sensor nodes. Techniques include Opt-ACM [249], which uses multiple routes for data routing with varied QoS values. A Mixed-Integer Linear Programming mechanism was applied to validate this method. However, this method does not pay attention to the energy efficiency of the network. A deep reinforcement learning-based routing algorithm [250] minimizes delays through an unequal clustering scheme, preventing the entire network from splitting into unequal clusters. However, a complex methodology has been adapted to use unequal clustering. The SLEB protocol combines load balancing and security authentication mechanisms based on clustered Wireless Sensor Networks (WSNs) [251]. While effective in balancing network energy and enhancing security overhead, SLEB allows multiple sensor nodes to send identical data to the receiver node, which can lead to redundancy.

III) Hybrid congestion control protocols use a combination of resource-based and traffic-dependent congestion control protocols to manage network congestion. For example, HOCA [252] uses active queue management and adjusts data rates based on traffic sensitivity. This protocol comprises four phases: initially, the sink node initiates data transmission to all source nodes, identifying node types, timing, and data priorities. Subsequently, the sink collects data from each node, determines the optimal path through hop-by-hop communication, and adjusts the data rate to alleviate congestion. ECA-HA [253] controls congestion using a hybrid congestion control approach. ECAHA reduces congestion in three phases. Initially, it uses ant colony optimization to identify the best route. Then, through multi-hop communication, backward ants verify path construction, and forward ants develop congestion-free routes. If rate adjustment proves unfeasible, an alternative route is established to control congestion. However, the ECA-HA's effectiveness is constrained to environments with a limited search space.

4) DATA COMPRESSION

The real-time interaction among human avatars and large storage requirements requires ultra-low delay and ultra-high efficiency compression technologies [254]. High-dimensional (3D) visual data representations, such as multi-view videos [255], point clouds [256], light fields [257], and 360-degree VR videos [258], have been developed to render the virtual environment beyond traditional 2D frames vividly. However, the challenge lies in effectively representing these high-dimensional visual data. One approach to compress these data is to transform them into 2D-frame-like video sequences using mature video coding schemes (2D) and then reverse them to high-dimensional data at a decoder. For example, light field data can be decomposed according to visual orientation and re-arranged into an inter-correlated video sequence that can be further encoded by Versatile Video Coding (VVC). The Video-based Point Cloud Compression (V-PCC) scheme [259], [260] projects a 3D point cloud into different 2D maps, which can then be compressed using 2D video codecs and remapped to 3D using a V-PCC decoder. This geometric transformation is also included in the compression of 360-degree video in the VVC standard. Another approach to compress high-dimensional data is to use geometric properties. For example, the Geometry-based Point Cloud Compression (G-PCC) scheme [153] uses a pruned octree form to approximate the original data [261].

Another potential solution for compression is semantic compression and decoding technology. This approach is based on a concept: 'semantic meaning passing' (e.g., users only need to share the meaning of a document or figure to maintain efficient and accurate communication), which helps eliminate the transmission of redundant information in Metaverse systems. Current mainstream semantic encoding transmission systems are designed as end-to-end integrated systems [262], [263], [264]. The sending end encapsulates semantic feature extraction and source-channel coding into one encoder module, while the receiving end encapsulates source-channel decoding and semantic feature fusion into one decoder module. Both the encoder and decoder are implemented using deep neural networks (DNN) to achieve nonlinear processing gains [265], [266], [267]. This structure originates from the deep auto-encoder (AE) structure (similar to the concept of deep joint source-channel coding (DeepJSCC)). However, the end-to-end training process of the encoder and decoder needs to consider the distortion of intermediate bottleneck layer data during channel transmission, which enhances the encoder and decoder's ability to counteract adverse factors such as channel noise and fading.

However, the limited computation and storage capabilities of energy-constrained Metaverse hardware restrict the local implementation of complex and energy-intensive ML algorithms (e.g., DNN) in semantic compression and decoding technology, including the training of semantic encoders, semantic decoders, channel encoders, and channel decoders, requiring better communication paradigm design and optimization. Several solutions have been proposed to address the devices' computation and storage capability issues [268], [269]. For example, Xie and Qin [268] proposed a lightweight semantic communication system to support the transmission of low-complexity text. It removes redundant weights from the semantic communication model by adopting neural network pruning techniques, thereby reducing the computational resources required for devices. Asymmetric encoder/decoder design [270], [271] is another approach for enhancing performance. It puts a lower computation load on headsets while using more computationally intensive algorithms on servers. For example, deep compressive offloading [271] uses a lightweight encoder to compress outgoing data on end devices and a resource-intensive decoder to restore the data at servers. Furthermore, it can use knowledge of downstream application requirements to determine which data features can be compressed away and which features are more important for the specific application, thereby obtaining a higher overall compression rate.

5) OTHER TECHNOLOGIES

In-network computing can reduce latency by optimizing service placement between servers and the network, allowing certain functions to be placed closer to the user (e.g., programmable network hardware) rather than deployed on servers. For example, in ICE-AR [122], mobile devices continuously send user context (e.g., Point of View (POV) video and related metadata) to edge nodes that perform real-time machine learning tasks. These nodes process the video to generate a detailed semantic description of the environment (known as the deep context). Different edge nodes provide different subsets of context extraction services. The deep context is then used by mobile clients to retrieve relevant content from cloud services to augment the POV video (see Section IX for further discussions on in-network computing).

Field of view (FoV) communication is being developed to reduce bandwidth requirements by strategically delivering content based on user inputs such as head position. Researchers focus on user viewing areas and explore view-guided optimizations [197]. They share a key idea: dividing a frame into tiles and applying non-uniform image resolutions across these tiles according to the user's sight. For example, Haynes et al. [272] and Zare et al. [273] predicted user head movement to lower the resolution of out-of-sight tiles. Ngo et al. [274] predicted users' Region-of-Interest (ROI) and streamed the high-resolution ROIs. Qian et al. [139], Fan et al. [275], and Liu et al. [276] streamed only the user's viewing tiles. Qian et al. [139] used a linear regression (LR) model to predict future viewports over time, treating viewport traces in the history window as a time series. Other studies have employed the LR model for similar purposes [277], [278], [279]. For example, de la Fuente et al. [277] considered angular speed and angular acceleration as forecast variables to estimate the future direction of the user from prior orientation data. As the linear model is limited in remembering the user's viewing behavior, the LSTM network has been applied for its long-term memory ability [278]. However, a trade-off exists. Since the client device requires continuous interaction with the hosting application server, it needs a guaranteed low latency and stable connection.

C. TAKE-AWAYS

It is challenging for some Metaverse systems to meet low latency requirements, high throughput, and deterministic communication. Expected advancements in areas such as next-generation wireless technologies are promising, targeting to support extremely low latency (0.1-1 ms) and high reliability (from 99.999% to 99.99999%). These advancements can come from improved wireless encoding schemes, higher frequencies, more access opportunities such as satellites and UAVs, AI-enabled more efficient resource allocation and other technologies. However, the question remains: how these enhancements can be achieved over the Internet and how to fully use these properties? For example, higher frequencies can lead to smaller scales and more cost, and increased access opportunities can lead to more path diversity, complicating network management. The lack of end-to-end control over Internet also prevents achieving uniform reliability across the network, thus limiting the impact of these improvements in next-generation wireless technology. These challenges require more thinking about system architectures and protocols design, including better communication structures that align with actual networks capabilities, adapting services to varying conditions, and enhancing resource allocation based on a deeper understanding of application requirements (e.g., moving towards semantic communication and more aggressive compression techniques). For example, applications should be designed to tolerate less reliable and lower-performing networks rather than assuming high reliability. Additionally, network architecture needs reconsideration to better support native broadcast and multicast functions. In summary, enhancing the communication performance of Metaverse systems is not solely a question of wireless improvements but also a question of system architecture and application-adaptation (as most applications on the Internet).

VI. MOBILITY

Mobility management is required to ensure service continuity as users move. Unlike Section V, which focuses on direct performance metrics, this section broadens the discussion to include additional communication aspects such as handover, multipath routing, and managing intermittent connectivity, ensuring a robust and reliable network infrastructure.

A. GAP ANALYSIS

Some Metaverse applications, particularly mobile AR/VR ones, are expected to operate across diverse environments with varying characteristics. There are two main types of mobile Metaverse applications.

- *Applications Designed Specifically for Mobile Devices:* These include AR location-based games (LBGs) such as Pokémon GO, a popular location-based AR game that allows players to find virtual creatures using their smartphones. Pokémon GO has a large user base, with 50 million monthly users in 2022 [63].
- *Applications Related to User Mobility:* These are used in scenarios such as driving or traveling by train. For example, [280] proposed a meta-empowered driver assistance system that visualizes guidance information on the windshield, where high-speed movement affects mobility.

This paper primarily focuses on the first type of application. Providing a seamless user experience for outdoor mobile AR/VR users is challenging due to mobility issues (e.g., unstable network connectivity and decreased service quality). There are two main perspectives.

1) MOBILITY MANAGEMENT

Consider a smartphone-based AR navigation system on a university campus [281]. The system overlays 3D information onto real scenes, using a visual-inertial ranging algorithm for real-time location tracking and map generation on mobile devices. Although the use of wired networks is impractical owing to user mobility, the viability of non-3GPP access methods such as **Wi-Fi** is also questionable because of reliability and handover issues [282]. Wi-Fi mobility is restricted by sparse open access points, causing interrupted data transmission even during short trips. Additionally, a handover can result in several seconds of connectivity gap when switching access points. A study in a medium-sized city in France [283] revealed that while Wi-Fi connectivity was available 98.9% of the time, Internet connection was accessible only 53.8%. The throughput and ultra-low latency requirements of mobile AR/VR applications make mobility support in **5G** challenging. Given that 5G networks predominantly use small-cell Ultra-Dense Networks (UDN), the frequency of handovers is expected to increase. In contrast to lower frequency networks, mmWave UDN contains a greater number of cells with diminished cell radius, inherently leading to more frequent handovers [25]. Because mmWave propagation is highly susceptible to fading and blockage, the channel quality of mmWave links can be extremely intermittent. To ensure optimal mobile AR/VR functionality, the user must maintain at least one high-quality connection and experience minimal delays during transitions between base stations.

Compared with 5G, **6G** performance will be improved, with peak transmission speeds elevating from 10 Gbit/s (5G) to 100 Gbit/s to 1 Tbit/s (6G), providing an ultra-low latency of less than 1 ms under Enhanced URLLC (ERLLC), ensuring 99.9999999% reliability [284]. 6G offers possibilities for ubiquitous network access and real-time massive data transmission between virtual and real worlds, with Space-Air-Ground Integrated Networks (SAGINs) achieving full network coverage and seamless integration of terrestrial and non-terrestrial networks [285], [286]. However, in Metaverse systems, a single virtual space can include many applications. Communication between these applications and servers (or blockchains) is expected to be executed through diverse channels over the 6G network using a service gateway (providing a unified access point for different network services) or Application Programming Interface (API) (enabling software applications to interact with each other), with each application uses different 6G services tailored to their specific needs [287]. 6G can provide even more diversity with respect to access networks (e.g., SAGINs), making multi-interface, multi-connection handling in Metaverse more challenging.

2) PROTOCOLS

Current protocols face several challenges in handling mobility.

Session Continuation Issues: Vertical handover is the process by which a mobile node redirects traffic flow between heterogeneous network interfaces. During vertical handover, the IP address and the network interfaces alter, and consequently, the port number would change. The IP address and port tuples uniquely identify the TCP connection. Therefore, the changes in the IP address and port would tear down the interface. This causes problems with connection-oriented communication and mobility. For example, HTTP range requests enable the transmission of partial HTTP messages rather than the entire message, resuming downloads effectively during network handovers. As HTTP uses TCP, the reestablishment of the TCP connection and security handshake are inevitable. This introduces additional delays and overhead, impacting the overall efficiency of the network communication.

Lack of Adaptability: Existing transport protocols (e.g., TCP) traditionally assume that network congestion is the main reason for packet loss and significant delays, which lack the adaptability to fluctuating network conditions. Such fluctuations might be induced by device movements, interference, or alterations in the system load because of bursty traffic sources or devices joining or exiting the network. With numerous users in motion, these fluctuations can lead to unstable performance and disruptions, highlighting the need for adaptability.

Lack of Fault Tolerance: High-speed relative movements between the access node and the base station often trigger service disruptions and diminished channel quality, as a base station struggles to maintain service within its coverage area. Given the cost of recovery in a latency-constrained scenario, protocols should ideally avoid recovery from loss. For example, TCP requires a minimum of one (usually two to three) round-trip times to detect and recover from a loss. With applications generating 30 frames per second and a maximum tolerable latency of 75ms, the recovery of a single lost frame is feasible only if the round-trip time is restricted to 37.5ms at most [282]. Since the ideal average end-to-end latencies for 5G and Wi-Fi are approximately 15.5ms and 150ms respectively, loss recovery without large service degradation is challenging.

B. EMERGING TECHNOLOGIES

The high throughput and ultra-low latency requirements in Metaverse systems impose challenges for mobility management. To deal with the lack of adaptability and fault tolerance explained above and the handover challenges, there are some potential solution.

1) MOBILITY MANAGEMENT

Intra-cell mobility refers to the management of a mobile device's movement within the coverage area of a single cell (base station), ensuring uninterrupted communication. It can lead to changes in communication characteristics and performance, which is primarily a 'radio problem'. To address this issue, protocols and applications must be more adaptive to these changes. However, solutions for such radio improvements are not the focus of this survey. See [288], [289] for detailed insights into radio improvements.

Horizontal handover enables a mobile device's automatic transition between base stations within the same operator's network, maintaining uninterrupted communication without changing the IP address (e.g., with the help of the GPRS Tunneling Protocol (GTP) in 5G network [290]). [291], [292] enable a device to connect with multiple base stations simultaneously and use various antennas for signal transmission and reception. By leveraging real-time tracking data, possibly combined with historical data, it can select the best current base station for service and identify the next stations for handover [293]. For example, Zhang et al. [294] proposed an anchor-based multi-connectivity architecture to reduce the handover costs associated with network densification, deriving concise expressions for handover probabilities through stochastic geometry analysis. However, this approach increases signaling overhead, as it requires maintaining connections to at least two base stations due to its multi-connectivity nature. Choi and Shin [295] proposed a random-access channel (RACH)-less handover scheme to achieve seamless, low-latency handovers for mobile User Equipment (UE) in 5G, combining it with a make-before-break (soft handover) methodology to reduce UE latency. However, they did not address the additional processing time and equipment costs for UEs and Original Equipment Manufacturer (OEM) vendors to support such technologies, which could be much higher in a multi-connectivity environment.

Vertical handover manages transitions between different network operators or communication technologies, often resulting in IP address changes. This process requires coordination between the Data Link Layer (Layer 2) and the Network Layer (Layer 3) to manage constant mobility events and the resulting performance changes [296], [297]. Managing interactions with CDNs or edge nodes during vertical handovers is challenging, especially in maintaining high service quality and minimizing disruptions. When users switch to a different network operator, they cannot maintain their connection to the previous one, resulting in the need to adapt to a new CDN environment. There has been a lot of research in the vertical handover decisions, including guidelines for hand-off schemes based on factors affecting overall hand-off latency [295], the use of neighbor graphs and non-overlap graphs to minimize probed channels and scanning delay for fast handoffs [298] and the implementation of selective scanning algorithms and caching mechanisms to reduce MAC layer handoff latency [299]. Liang et al. [300] proposed a combination of analytic hierarchy process and cooperative game theory to address vertical handover problems, considering demand preferences under different service types and improving QoS in heterogeneous networks. Bao et al. [301] proposed a vertical handover scheme based on QoE maximization using a Markov decision model, which adapts to user movement and maintains a high average QoE. However, these two schemes do not consider changes in channel capacities, and the latter overestimates that a Visible light Communication (VLC) hotspot can provide higher rewards than a WiFi hotspot, ignoring the fact that a VLC hotspot may be blocked.

Ott and Kutscher [302] introduced the concept of maintaining persistent sessions over disrupted transport connections using a connection-ID-like identifier. Building upon this foundation, technologies such as QUIC have been developed to support IP address changes through the use of persistent connection IDs without breaking the connection, eliminating the need for extensive infrastructure management. For example, Sinha et al. [303] proposed CQUIC, a learning-based cross-layer approach for heterogeneous handover decisions, leveraging QUIC's benefits by predicting a CrossLayer Score (CLS) that includes predicted Signal-to-Interference Noise Ratio (SINR), QUIC Bandwidth, and round-trip-time (RTT) status from QUIC sessions to model handover decisions proactively. In the future, 6G is expected to enhance Metaverse services with more diverse access networks [304], [305]. With future proposed networks such as SAGINs, there will be more multi-access opportunities and a broader range of networks and paths for data transmission (from equipment to devices). However, this abundance of options requires concurrent evaluation to determine the optimal path, which affects network metrics such as service latency, calling for improved network and end-system management strategies.

2) PROTOCOL

WiFi is commonly used as the default mode of communication in mobile devices. However, owing to its unreliable channel conditions and sensitivity to blockage, packet loss and handover are inevitable while using WiFi. A solution to this issue is MPTCP [306], which allows a device to establish multiple connections or sub-flows simultaneously. Thus, even if one interface fails, the device can use another interface. An example of its use is in iOS11 for Siri [307]. However, protocols built upon the TCP/IP stack face challenges, such as Head-of-Line (HoL) blocking [308] and connection breakage issues: the connection continuity would be broken if the handover process fails, and afterwards MPTCP has to re-establish the disconnected subflow via a three-way handshake [309]. Additionally, MPTCP is not the best-fit for smartphones.

To address the issues with TCP, the features of QUIC [310] (or other *session based approaches*), such as stream multiplexing, frame structure, and 0-RTT handshake, enable it to improve transmission performance and easily adapt to various applications. Web browsers and servers increasingly support transmission through QUIC or HTTP/3. However, QUIC has limitations. For example, QUIC has a feature that it will migrate sessions on disconnection of network interface. This approach, being reactive, causes an additional delay in migrating the connection after vertical handover. Also, unaware of the network characteristics, such as packet loss, SINR, and back-haul Internet connectivity issues, QUIC might increase the latency [303].

The multipath extension over QUIC (MPQUIC) enables the simultaneous use of different paths for exchanging non-probing frames within a single connection [311]. MPQUIC has better performance because *I) 0-RTT Connection Establishment.* MPQUIC spends one RTT to initialize a subflow if the subflow has never been established before, and 0 RTT otherwise. In the case of handover failure, MPQUIC consumes 0 RTT to restore the disconnected subflow. *II) Connection Migration.* MPQUIC is designed with mobility in mind, and instead of using a 4-tuple (i.e., source address, source port, destination address and destination port) to identify a connection, a 64-bit Connection ID (CID) is used. Thus, the connection remains alive even if the address and/or port are changed because of mobility. Even though MPQUIC is still working in progress, there are already some related studies on protocol designs [312], protocol implementations [311], scheduling strategies [313], and congestion control algorithms [314]. However, applying MPQUIC directly poses challenges, as its standard congestion control scheme, the loss-based Opportunistic Linked Increases Algorithm (OLIA) is undesirable in applications where losses are often caused by handover events, leading to severe performance degradation [314].

Information-centricity [315] is a concept where data packets (e.g., those making up an object in a microverse) are named within a network of relays/caches. These packets are forwarded based on names rather than locations and are returned to requesters by reversing the paths. This data-oriented approach natively supports the mobility of data consumers by a stateful forwarding plane, where consumers are addressed anonymously by hop-by-hop states, without explicit locators or equivalents [316]. Consequently, the mobility of consumers is transparent to producers and forwarders. Therefore, this approach can prevent the vertical handover issues commonly in CDNs by requesting the same names of data objects within the network, rather than relying on a specific CDN. However, in data-oriented approaches, consumer mobility is anchorless by design [317]. The challenge for such complete anchorless approaches is exacerbated when both endpoints are mobile simultaneously, as the network cannot depend on any stable anchor point to maintain connectivity, nor can it rely on traditional routing mechanisms, which can introduce overhead and instability [317]. An appropriate transport layer must be on top, capable of adapting to eventual disruptions and path variations caused by mobility events. More details regarding information-centric system design in Metaverse systems are discussed in Section IX-D.

C. TAKE-AWAYS

In mobile Metaverse applications, the combination of high-throughput (Gpbs) and low-latency (ms) is currently out of the range of available wireless technologies that can be used in a front-end device in the real world, and come with severe range limitations. Some Metaverse services replicate content using CDNs or cloud services, and some research prototypes use edge computing to enable the placement of services at the network edge close to users. However, managing interactions with CDNs or edge nodes during vertical handovers is challenging, particularly in maintaining optimal service quality and minimizing disruptions. Upon transitioning to a different network operator, the connection to the former one cannot be maintained, resulting in the need to adapt to a new CDN environment.

VII. LARGE-SCALE OPERATION

Some Metaverse applications connect a large user group over the Internet, posing challenges such as scalable group communication, state management, and system robustness. This section discusses the gaps in achieving efficient and high-performing large-scale operations and evaluates current technologies to address these issues.

A. GAP ANALYSIS

From a large-scale deployment perspective (e.g., Remote Work and Online Collaboration, Social Life and Gaming applications, as shown in Section II-B), one issue is the scalability of communication in the client-server based Metaverse with respect to the number of users. As discussed in Section IV, most virtual world frameworks operate as server-based, multi-threaded, and centralized systems. These structures typically feature two focal points: one where certain servers render the virtual realm for users and process avatar actions, and the other where servers manage the collective virtual world state and modify it based on avatar actions and client instructions. For example, Second Life employs a server-based centralized architecture for each virtual world, where individual regions are managed by specific servers. These servers handle the most communication and computations task for virtual simulations, such as in-game clock synchronization and 3D animation [318]. However, these centralized architectures may result in a heavy server load [319]. It is challenging to adequately accommodate the demands of a large user base update in real-time without sacrificing system efficiency and user experience (issues arising from the architecture will be discussed in Section IX).

Cheng et al. [2] assessed five prominent social VR platforms, revealing fundamental scalability issues across several popular social VR platforms. The study found that throughput and end-to-end latency grow nearly linearly with increased users. As shown in Table 4, many platforms have a restricted capacity, barely supporting up to 40 users concurrently, with Horizon Worlds accommodating just 20. Tests [2], [112], [113] show avatar-related content in these platforms equates to 0.5 Mbps per user. Therefore, 40 users would each receive around 20 Mbps from the UDP flow). As more users connect, servers simply broadcast each user's avatar data without further optimization, causing an almost linear upsurge in the downlink throughput. Consequently, as the number of users increases, there are serious network capacity issues in Metaverse systems: throughput scales exponentially with respect to the number of users. No strategy has successfully established a stable, broadly recognized model with demonstrable evidence of notable scalability improvements. These strategies face a fundamental bottleneck in enhancing virtual world scalability: an unavoidable quadratic increase in simultaneous interactions (e.g., server mesh). This issue involves the potential for n^2 updates among a group of n interacting entities (i.e., avatars, agents, or objects), regardless of the interaction type, locational, communication, or other relational facets [11].

B. EMERGING TECHNOLOGIES

From a large-scale deployment perspective (e.g., million users), factors such as Content Delivery Network (CDN) and caching must be considered. This section provides potential solutions to these issues.

1) SCALABLE MULTI-DESTINATION COMMUNICATIONS

The content dissemination services of Metaverse systems can place a huge load on existing data-oriented communications [320], wherein a channel with a large volume is required for real-time traffic. Therefore, Metaverse applications require service-level optimization and service diversity to reduce the burden on wireless channels in edge networks. Multicast has been employed as a tool to support content delivery in Metaverse systems. For example, based on the correlation between the FOV and the location of viewers, Perfecto et al. [321] suggested a mmWave physical layer multicast scheme. It separates the multicasting procedure into two sub-issues (i.e., request access and scheduling) that can be solved independently. Long et al. [322] considered two kinds of quality requirements (absolute smooth and relatively smooth) and multicast patterns (with or without transcoding). TDMA-based multicast content delivery from one server to multiple users was investigated in [323]. Optimum transmission power and delay time are used to achieve maximum performance. Tan et al. [324] presented an energy-efficient resource allocation scheme with non-orthogonal multicast and unicast transmissions, and the energy efficiency was optimized. The performance of multicast millimeter-wave wireless networks was studied based on stochastic geometry, and a cooperative Non-Orthogonal Multiple Access (NOMA)-based multicast scheme was proposed [325].

In Metaverse systems, multiple viewers may simultaneously request video content (e.g., in cinemas and shopping malls), resulting in overlapping FoVs. This leads to the repeated delivery of a large amount of content, consuming limited bandwidth resources and exacerbating existing delivery pressures. NOMA-based multicast can provide support for such VR transmissions. For example, Xiang et al. [326] studied collaborative and non-collaborative NOMA-based transmission schemes in VR livecasts. By optimizing power allocation, they increased the average outage capability and enhanced the quality of user experience. Ding et al. [327] combined NOMA with mobile edge computing and proposed a dynamic computing offloading solution for Metaverse applications, reducing computational costs and considering the requirements of object identification, posture estimation and vision tracking. Other researchers also highlighted

NOMA's impact on multimedia applications (e.g., VR) in terms of enhancing wireless capacity [328] and resource optimization efficiency [329]. However, efficiently supporting multicast under the TCP/IP protocol in the large-scale Metaverse presents several challenges. Many wireless link layer protocols lack support for acknowledgment frames (ACK) in multicast, hindering packet loss recovery at the link layer. Moreover, the heterogeneity of devices and networks in Metaverse systems means that node devices may operate on different link layer protocols, and the transmission rate of each link layer protocol varies. Therefore, the multicast sender must transmit at the lowest rate among all receivers. Additionally, devices in Metaverse systems may enter sleep mode to conserve battery life, and these sleeping nodes may miss multicast packets.

2) CONTENT DISTRIBUTION AND IN-NETWORK CACHING

The intermittent and dynamic network environment in some Metaverse systems, such as mobile VR, poses challenges in maintaining stable connections. One approach to reduce transmission latency and duplicate traffic is to use in-network/application-layer caching and proxies to copy content in servers close to users, achieving efficient data exchange. Here, we focus on caching at the CDN level from a large-scale perspective.

CDNs provide managed content distribution and serve two primary purposes: offloading servers and networks by caching and replicating data in the network, and reducing latency by making data assets closer to users. Caching enables servers to efficiently process similar user commands by referencing previously recognized instructions. Such reuse benefits individual users, as seen in pre-rendered game environments, and multiple users in shared experiences such as interactive city navigation. For example, Brandenburg et al. [330] proposed an optimization scheme for tiled VR video delivery in a CDN, reducing video transmission latency through network stack optimization. Mahzari et al. [331] proposed a FOV-aware cache strategy, employing a naive Bayes-based scheme to learn the probabilistic model of tile access and determine cache replacement priorities when storage overflows. Maniotis and Thomos [332] proposed the concept of a virtual viewport to simplify cache decisions, using a Deep Q-Network (DQN)-based method to optimize cache placement for maximizing video quality. This approach leverages the neural network's fitting ability to adapt to dynamic and complex communication environments. Maniotis and Thomos [332] proposed a 360 Video Caching approach using Deep Reinforcement Learning, employing the Markov Decision Process (MDP) to locate 360 video content in edge cache networks. A Deep Q-Network (DQN) algorithm is used to determine optimal caching placement, prioritizing the caching of popular 360 videos at base quality along with a virtual viewport in high quality. In Metaverse systems, application-layer caching by proxy nodes can shorten response waiting times by requesting resources, temporarily caching data, and responding to requests on behalf of the sleeping node. However, this approach faces several limitations [333]:

- Proxy nodes are pre-configured, and with dynamic changes in the network environment, these pre-configured proxy nodes may not be the best choice in the current environment.
- The necessity for a resource discovery mechanism to locate nearby proxy nodes introduces additional complexity and overhead.
- Pre-selected proxy nodes may become unreachable in a dynamic network environment, requiring rediscovery and reconfiguration.
- Introducing proxy nodes increases security vulnerabilities.

Recent researches [332], [334] have sought to address these challenges. For example, Nath and Wu [334] explored caching dynamics. They evaluated aspects such as the necessity of caching a task, the ideal transmission power for task offloading, and the appropriate amount of Multi-access Edge Computing (MEC) resources required for a task. However, this approach requires the network layer to be aware of application-layer resources and integrate caching into the forwarding process. It also requires fundamental modifications to the network security model to ensure the safety and reliability of network layer caching.

3) MEDIA OVER QUIC (MOQ)

Bandwidth regulation of multiple streams from various senders and receivers has been researched intensively over the years, as a part of congestion control [335], [336]. However, these works either guarantee a fair-share among all competing flows [241], [337], distinguish the background from the primary traffic [335], or are customized for specific requirements in particular scenarios [338]. QUIC tries to solve a 'lighter' problem by multiplexing multiple requests/responses over a single connection using different streams, thereby avoiding head-of-line blocking. It improves loss recovery by using unique packet numbers to avoid transmission ambiguity. However, the default scheduler for QUIC is either a form of a sequential variant (usually FIFO) which is sub-optimal, or a weighted Round-Robin (RR) algorithm instead [339], [340]. Because the weights do not precisely match the needs, frequent updates are required, leading to delays and high loss rates [341]. As an extension of CDN, the IETF formed a new working group called Media over QUIC (MOQ) in 2022, standardizing the use of QUIC for large-scale media transmission in one-to-one, one-to-many and many-to-one applications that require interactivity (hence, low latency [342]). The goal is to develop a scalable and efficient low-latency media delivery solution for media ingest and distribution in both browser and non-browser environments, facilitating two primary functions: enabling live streaming of events, news and sports with enhanced interactive capabilities, and scaling up real-time media applications to accommodate larger audiences.

Early implementations of this concept are Twitch (i.e., Warp [343]) and Meta (the earliest MOQ implementations [344]). Then MOQ [345] and its extensions, such as QuicR [346] are designed to further improve the delivery and quality of real-time interactive media and streaming services. Specifically, MOQ uses a latency-configurable delivery protocol for sending media from one or more producers to consumers through relays using WebTransport (in browsers) or raw QUIC (elsewhere) [347]. Relays are responsible for forwarding incoming media to other relays or consumers to scale media distribution without requiring separate encoding for each consumer. They prioritize or discard packets based on specific metadata ('relative priority') exposed in the incoming packets to manage congestion and meet the application's latency requirements. Consumers trade off quality and latency by adjusting their waiting time for media based on network conditions and the desired user experience. However, these solutions also introduce additional complexity [348]. This complexity arises from factors such as the need for sophisticated management of the overlay network, the challenge of optimizing data paths in real-time to account for network congestion and changes, and ensuring compatibility with existing network protocols and infrastructure. Currently, the details of the MOQ protocol and its proposals remain under discussion [348]. What is needed is a fine-grained, hierarchical media exchange, designed for low-latency interactive communications that supports scalable multi-destination distribution, along with in-network replication and transformation.

C. TAKE-AWAYS

Scalable media distribution has long been a critical component of Web and video streaming applications, primarily addressed by CDNs. However, while existing conferencing platforms can support a relatively large number of users simultaneously (e.g., zoom can support up to 1000 participants in a single session[19]), this is insufficient for Metaverse systems' demands of audio-visual quality, high data rate, and large-scale real-time interactivity (as CDNs cannot guarantee consistently low latency). Emerging technologies are needed to better support Metaverse systems. For example, overlay approaches such as MOQ [345] and extensions such as QuicR [346] merge real-time interactive media with an enhanced CDN approach, offering smaller packet formats and low latency encodings. However, these approaches introduce additional complexity, and cannot fully use network capabilities such as native broadcast or low-latency transport (because of redundancy in wireless networks). One way to better address this issue is through data-oriented approaches. ICN inherently supports native multicast capabilities [349], enabling networks to provide implicit multi-destination delivery services in a manner that minimizes the need for application-level awareness of the network's edge layers [136] (see Section IX-D).

VIII. END SYSTEM ARCHITECTURE

In this section, we discuss specific topics related to Metaverse application development and performance, including rendering, device energy constraints, and the use of computational resources, along with their limitations and potential solutions.

A. GAP ANALYSIS

The performance of Metaverse applications is currently restrained by several factors, including the inefficient use of computational resources in processes such as rendering, encoding, and decoding, the complexity of development due to protocol stack intricacies, and device energy limitations.

A key challenge is the allocation of mission-critical computing resources during rendering. As discussed in Section IV-B, rendering in Metaverse systems involves transforming raw data (e.g., images, videos, and 3D objects) into visualizable 2D/3D objects. This process requires computational tasks to be executed at nodes from the content source to the destination, requiring various hardware capabilities and combinations of CPU and GPU from end nodes to cloud networks. Cheng et al. [2] analyzed five leading social VR systems, revealing technical challenges associated with rendering in these systems, including high computational demands, restricted video frame rates (even with a small user base), and suboptimal network utilization. Some specialized third-party applications (e.g., CubiCasa5K [350], Vectary [351]) are used for rendering 3D scenes on mobile devices. These apps integrate rendering resources within the application's rendering environment, completing the rendering process locally with the WebGL library (preferred for browser rendering) or DirectX library.

Another primary concern for deploying Metaverse systems is the energy consumption of hardware, particularly mobile AR/VR devices. Rendering media such as volumetric video content on these devices is challenging due to their constrained computing and energy resources, especially when multiple volumetric objects are in the scene. Equipped with advanced sensors, cameras, and processors, these devices require considerable power, leading to rapid battery depletion and a tendency to overheat [25]. Although incorporating larger batteries can extend usage, it can compromise device design and user-friendliness. The Meta Quest Pro headset[20] weighs over 700g and has 256GB of storage, 12GB of RAM, and a Snapdragon XR2+ Gen1 processor. However, it faces challenges related to power consumption and cooling during periods of high utilization [352]. As discussed in Section III, remote or interactive rendering can reduce the processing load on the user and enable efficient bandwidth usage by sending only a partial view of the volumetric object according to the user's position. However, this can further increase the communication and computational overhead by requiring additional transcoding in the cloud.

[19] https://nerdschalk.com/zoom-limit/#zoom-participants-limit

[20] https://www.meta.com/quest/quest-pro/

In large-scale Metaverse systems (e.g., social VR systems), simultaneously rendering personalized experiences for many users is complex. The lack of efficient computational resource utilization can lead to user scalability issues. Cheng et al. [2] found that on-device computation resource utilization grows nearly linearly with increased users. Although recent data shows some improvement, for example, Second Life is limited to about 40 people per 256m x 256m region [353] (there are 27,473 regions live right now[21]), Decentraland has a maximum limit of about 200-300 users per shard [353] (usually there are about 6 shards up[22]), and Virtway allows a maximum of 1000 concurrent users in the same room on its B2B Metaverse platform [354], the user scalability issue persists. Efforts to improve user number scalability issues in virtual worlds [355], [356], [357] have focused on enhancing computational distribution or fine-tuning system architecture. For example, Colyseus [355] used a P2P architecture for large-scale cloud gaming, supporting a larger number of players by partitioning game objects (e.g., avatars, items) over participating nodes using a distributed hash table (DHT). Rokkatan [356] employed multiserver replication to improve scalability by partitioning objects across available servers and placing read-only copies on all other servers. These strategies offer better scalability than centralized solutions and can minimize throughput requirements. However, the distribution technique is more complex to implement, and the interchange of additional control messages can affect network throughput (resulting in potential communication overhead) [357], [358]. We discuss the issues arising from decentralized architecture in Section IX.

B. EMERGING TECHNOLOGIES

Performance degradation on the end users' side may result from issues such as decreased throughput and increased system load at the application level. To mitigate these issues, tasks can be offloaded fully or partially to the MEC at the network edge or cloud data center. However, MEC resources are often limited and may not suffice to meet the extensive demands posed by simultaneous users [364], [365], [366]. Given Metaverse systems' ambition to deliver immersive experiences to a vast number of users concurrently [367], optimal allocation of computing resources is important to address resource demand conflicts among these users. In this Section, we explore emerging techniques for these issues.

1) DYNAMIC TRANSCODING FOR HETEROGENEOUS DEVICES AND NETWORKS

'The Metaverse' features various media types, including 2D, 3D, holograms, and 360-degree videos. Transcoding these varied media formats is computationally demanding and time-consuming, leading to increased costs and delays (as detailed in Section III). Technologies such as in-network computing can improve video transmission by enabling dynamic transcoding, applying advanced context-based compression algorithms, and enabling pre-fetching, pre-caching, and movement prediction within the network. It can support heterogeneous receiver groups and networks with varying quality layers. Recent studies have explored the role of edge computing in enhancing efficiency [368], [368], [369], [370], [371]. For example, Erfanian et al. [368] introduced a Virtual Reverse Proxy (VRP) at the edge server, which aggregates incoming requests to lower bandwidth usage in the backhaul network and stabilize client QoE against network throughput fluctuations by buffering segments at the VRP. OSCAR [368], [370] is a live streaming framework that aims to save bandwidth and transcoding costs by aggregating incoming requests at edge servers and then transferring only the most demanded representation from the origin server to the optimal set of Point of Presence (PoP) nodes. Subsequently, virtual transcoders hosted at the PoP nodes transcoded the most desired representation to the requested representations. These are then transferred to appropriate edge servers and respective clients. However, OSCAR only uses traditional, resource-intensive transcoding methods, and is computationally intensive and time-consuming. To address this, Light-weight Transcoding at the Edge (LwTE) concept [371] was proposed to streamline transcoding using edge computing and Network Function Virtualization (NFV) paradigms. LwTE optimizes this process by taking metadata from the encoding process and reusing it in a transcoding process at an edge server, cutting down transcoding times and computational costs. As metadata is much smaller than the complete representations, LwTE also decreases storage and bandwidth requirements. However, LwTE is currently studied only in the context of Video-On-Demand (VoD) services [371]. Transcoding is also included in the cache sub-problem, enabling the cached content to be converted to a lower bitrate and serve a broader audience. We will discuss this further in Section IX.

2) HIGH-RESOLUTION MEDIA RENDERING AND TRANSMISSION

Most scenes in Metaverse systems are virtualized and need to be rendered in real-time. However, as explored in Section VIII-A, the quality of rendering is constrained by various factors including hardware capabilities, software efficiency, network conditions, and the rendering methods employed. The capability of dynamically offloading functions, such as rendering, to external servers is required to improve performance and adaptability.

Cloud rendering is a technology that offloads users' local 3D rendering tasks to cloud servers equipped with powerful rendering capabilities. Users interact with these servers by sending game or avatar inputs, which are then executed by the servers to complete the rendering tasks and transmit the results (rendered audio and video) back to the users. The primary advantage of cloud rendering is that it alleviates users from the constraints of hardware configurations and

[21] http://www.gridsurvey.com/

[22] https://decentraland.github.io/catalyst-monitor/

TABLE 8. Summary of approaches in efficient AR/VR rendering and offloading.

Issue	Main Idea	Features	Ref.
Stochastic Demand	Utilizes optimization theory to adapt resource allocation in virtual education based on uncertain demand	Optimization theory for adaptive resource allocation	[8]
	Suggests offloading decisions using a combination of a proxy (to measure device energy and storage) and a profiler (to monitor network conditions), balancing operational costs and network stability.	Guidelines for computational offloading in AR applications	[70]
	Introduces polynomial codes to speed up distributed matrix-matrix multiplications in the Metaverse.	Supports large-scale offloading	[359]
	Employs Coded Distributed Computing (CDC) for collaborative execution of Metaverse rendering tasks, using reputation and game-theory for worker selection.	Collaborative computing with reputation-based worker selection	[360]
	Discusses coding schemes like MatDot and PolyDot to optimize the recovery threshold and communication costs trade-off for Metaverse tasks.	Advanced coding techniques for efficient decoding and performance stability	[361]
	Uses Stochastic Integer Programming (SIP) to address straggler uncertainty, optimizing offloading decisions in edge computing environments.	Coded stochastic offloading scheme to counteract stragglers	[51]
Heterogeneous Tasks	Implements a deep recurrent Q-network (DRQN) to classify tasks by performance requirements, maximizing task completion within deadlines.	Uses fog nodes instead of remote servers to reduce latency	[362]
	Proposes a rapid adaptation offloading framework through limited gradient updates and Meta RL, modeling tasks as Directed Acyclic Graphs (DAGs).	Quick adaptation for AR/VR applications using Meta RL	[363]

software compatibility on their devices, with all rendering tasks being carried out on the cloud servers. For example, Sony's PSNow project [372] enables PS4 (Sony developed a home game) users to play games using cloud technologies without downloading game resources to their local machines. It leverages cloud rendering to process game interactions and deliver rendering results to users [373]. However, the instantaneous nature of rendering processes demands high responsiveness from cloud rendering platforms. The reality today is that, although it has been around for decades, cloud rendering still does not meet the requirements of many Metaverse use cases [374]. Challenges are as follows [374]. I) Almost always higher bandwidth consumption than downloading and rendering 3D worlds locally. II) Still has relatively high latency for things such as VR head-tracking (i.e., millisecond-level latency constraints).

A hybrid technology combining cloud and local rendering is called split rendering or partial offloading, a method proven to enhance AR/VR device battery life by 30%-50% [375]. The key idea is to achieve minimal latency and cost by offloading partially rendered 3D scenes to the cloud [376]. Mobile devices undertake a small portion of final rendering to update the last-millisecond visual POVs for best head-tracking. In contrast, servers handle most of the workload, including intensive computations like global illumination and fluid simulation. This technology allows rendered video and audio to be shared among multiple participants at the same virtual place and time. In an even more distributed fashion, rendering nodes can be provided by any network participant, from Metaverse users (e.g., their home PC, laptop, or gaming console) to Metaverse service providers. Table 8 presents several strategies for optimizing AR/VR rendering and offloading. However, a gap in these studies is the neglect of dynamic characteristics in Metaverse systems. In contrast to static systems, the computational resources in Metaverse systems fluctuate in real-time, presenting additional challenges for optimal performance: *I) Stochastic Demand and Network Conditions.* Metaverse systems are expected to accommodate millions of simultaneous users, each with varying demand schedules and usage criteria. Consequently, predicting the exact number of edge computation services necessary for offloading is challenging. Owing to these unpredictable demand shifts, assuming constant user demand is impractical. *II) Heterogeneous Tasks.* The wide range of Metaverse applications, including work, entertainment, and social interactions, produces diverse computational tasks with different latency prerequisites. For example, rendering forefront VR scenes requires a lower latency than rendering background VR scenes. Therefore, the development of rendering and offloading plans should consider the simultaneous presence of various tasks within Metaverse systems.

3) MODULAR SYSTEM ARCHITECTURE DESIGN

Metaverse systems should support flexible component decomposition and function offloading to accommodate heterogeneous devices and edge networks (i.e., accommodate only the necessary data elements for rendering at variable quality levels, possibly using dynamic transcoding and Level of detail (LOD) support). Such systems require a modular architecture, where components can be seamlessly adjusted or offloaded to external servers.

Microservices can be viewed as a lightweight extension of the cloud computing model, incorporating application logic in containers and orchestrators for resource allocation and other management functions [377]. This approach decomposes monolithic applications into smaller, interconnected services that communicate over well-defined APIs. Unlike traditional VM-based systems, microservices generally adopt a 'stateless' approach, where the service or application state is independent of the computing platform. This enhances fault tolerance against failures in computing platforms or processes. Recent studies [378], [379], [380] have introduced the concept of hierarchical microverses for Metaverse systems, organizing the virtual environment into structured tiers, each providing unique functions that contribute to a larger digital ecosystem. This concept offers a systematic method for creating and managing virtual spaces. For example, Qu et al. [381] proposed a task-oriented, edge-scale microverse solution for using digital twins in smart cities. Instead of replicating a whole smart city in a single digital world, every microverse instance represents a manageable digital twin of an individual network slice tailored for tasks, supporting on-site/near-site data collection, processing, information fusion, and real-time decision-making. Bujari et al. [379] proposed a layered conceptual architecture for manufacturing digital twins. Digital twin components, such as physics simulators, rendering engines, and streaming services, are deployed as microservices that can be combined to form complex services or entire applications. A resource-aware orchestrator efficiently deploys these microservices, matching service requirements with available resources to ensure the QoS/E specifications of various applications. In scenarios where multiple users interact with a virtual environment simultaneously, services can be associated with corresponding user groups to provide a better overall experience.

Multi-access Edge Computing (MEC) provides solutions for function offloading in Metaverse systems by enabling systems to select resources on appropriate edge servers (so-called MEC application servers) and offload tasks for execution [19], [382]. Edge servers mainly execute tasks such as foreground rendering, which demands less graphical detail but more stringent latency requirements (rendering foreground and background in some simple scenarios). They also transmit user data to cloud servers and relay synchronization information to users. Cloud servers are equipped with robust computing and storage capabilities in data centers, handling tasks that are computationally intensive but more tolerant to latency, such as user information storage, user state synchronization, and background rendering [383], [384]. Recent studies [385], [386] have explored using MEC to address these computation-intensive, low-latency tasks in Metaverse systems. For example, Long et al. [386] propose a MEC–cloud collaborative framework to optimize the Metaverse systems experience at the network edge. They modeled edge resource allocation for multiple users as a decentralized partially observable Markov decision process (Dec-POMDP), introducing a multi-agent deep reinforcement learning (MADRL)-based approach. In this model, each agent is responsible for managing communication and computation resources for an individual user in Metaverse systems. However, there are several issues when implementing MEC in Metaverse systems. I) Eliminating the use of buffer makes the system vulnerable to network jitters. II) Not being able to use B-frames (see Section III) reduces the efficiency of real-time coding. III) the resource conditions are difficult to predict due to variable central processing unit (CPU) utilization rates, network bandwidths, delays, and jitters [386].

4) IN-NETWORK PROCESSING AND DISTRIBUTED COMPUTING FRAMEWORK

In-network computing is a potential solution for supporting latency-constrained Metaverse applications [9]. It can leverage unused network resources to perform computational tasks [387], [388], facilitating the instantiation of computing functions closer to the end-users. Several studies [389], [390], [391], [392] have demonstrated the potential and efficiency of In-Network Computing in enhancing the performance of distributed streaming applications and Metaverse experiences. For example, Rashid et al. [391] proposed an in-network placement and task-offloading solution for delay-constrained computing tasks in Metaverse systems. This model optimally decides whether to offload rendering tasks to In-Network Computing nodes or edge servers, considering time constraints and computing capabilities using graph neural network (GNN). Cai et al. [392] design control policies for the joint orchestration of compute, caching, and communication (3C) resources in the next-generation distributed cloud networks for the efficient delivery of Metaverse applications that require real-time aggregation, processing, and distribution of multiple live media streams and pre-stored digital assets. They describe Metaverse applications via directed acyclic graphs that can model the combination of real-time stream-processing and content distribution pipelines. However, challenges include increased power consumption due to additional computing resources and the need to consider the location of in-network computing nodes and user demand under dynamic network conditions. These factors lead to a joint optimization problem involving the trade-off between time delay and energy consumption for Metaverse tasks.

Most previous works [365], [390], [393] on offloading have treated it as a singular process without considering scenarios where tasks could be distributed and handled by different computing nodes. However, in Metaverse systems, a single task often comprises multiple subtasks (as shown in Section IV) that can be decomposed and offloaded to various computing nodes (e.g., in-network computing nodes). For example, Alriksson et al. [394] proposed a method for task splitting for XR, offering three upload modes. While this approach may be adequate for VR/AR with limited users, it cannot work well in large-scale Metaverse deployments (e.g., large-scale cloud gaming). This distributed computing and offloading in Metaverse systems must consider the large-scale simultaneous demand for network resources.

Device-to-Device (D2D) communication serves as a means to offload the infrastructure network by distributing computational tasks among nearby devices via Bluetooth [395], WiFi Direct [396], and NFC [397]. To efficiently use the computation resources of mobile devices, users can adaptively participate in D2D multicast clusters, transmitting shared reused signals (e.g., tracking signals) to service providers. Therefore, the performance of Metaverse systems steaming transmission can be improved by increasing the total bit rate within the cell while ensuring short-term fairness among users. In [398], a D2D scheme was proposed for small cell networks, which combines cache placement and D2D link establishment. This scheme allows users to load caches on mobile terminals and other devices, and prefetch highly popular content to local caches during off-peak hours. However, this approach introduces increased system complexities and depends on user willingness to allocate resources, which may pose challenges to its practical implementation.

C. TAKE-AWAYS

Current challenges in Metaverse systems system design include the lack of flexibility and modularity due to monolithic architectures and centralized servers. To address these problems, there is a growing need for adaptive and modular system designs, which can enhance system flexibility and adaptability by dynamically offloading functions such as rendering, and improve system performance. Recent advances in platform virtualization, link layer technologies and data plane programmability offer the possibility to enhance system modularity. They can enable the deconstruction of vertically integrated systems into independent components with open interfaces, elastically scale those virtual components across commodity hardware, and adjust dynamically to workload demands. However, more work is needed to make it happen. For example, optimally allocating limited communication and computation resources at the edge to a large number of users in Metaverse systems is still challenging. As discussed previously, microservices are not really designed for Metaverse systems, and MEC have not be fully used, they need more application-layer support. In general, developing an adaptive, modular system for Metaverse systems demands a holistic strategy. This strategy should integrate independent functions with clear interfaces for efficient operation and interoperability. It requires strategic server selection and communication optimization for effective task offloading, prioritizing security to ensure data integrity and user trust. We will discuss this research direction in detail in Section IX.

IX. VISION FOR AN IMPROVED METAVERSE INFRASTRUCTURE

Several performance and usability issues in Metaverse systems are due to architectural problems in the network, applications, and the overall system. For example, as noted in Section V, balancing high-quality and low-latency communication is problematic, especially with a peak-to-mean bit-rate ratio; large-scale communication is often costly or fails to meet required performance levels, as mentioned in Section VII; commonly used overlay protocols and corresponding infrastructure (e.g., CDNs) cannot fully use network capabilities such as native broadcast or low-latency transport; issues of limited flexibility or modularity, with monolithic application design and centralized server-based architectures, resulting in inflexible system design and performance problems (as discussed in Section VIII).

Some recent Metaverse systems and prototypes have begun addressing these architectural constraints. For example, Servo [399], a serverless backend architecture for Metaverse environments, leverages a collection of serverless techniques to provide fine-grained virtual-world scalability; ARENA [400], a multi-user and multi-application modular XR system design, facilitates cross-platform interaction with 3D content generated by any number of network-connected agents (human or machine) in real-time.

We aim to re-assess the fundamental goals and requirements, without being constrained by existing system architectures, protocols, etc. In Section IX-A we first summarize high-level goals followed by an application example in Section IX-B. In Section IX-C, we discuss more specific design ideas for future Metaverse systems, and in Section IX-D, we suggest future research directions. Note that these design ideas and directions are not inherently unimplementable. The point is to consider the constraints of networking and application design together, which could lead to future practical implementations and system designs.

A. GOALS

We suggest the following high-level goals for the design of future Metaverse systems and corresponding infrastructure evolution:

1) LOW LATENCY AND SCALABLE MEDIA COMMUNICATION

Achieving low-latency on-demand/live streaming in the Metaverse is challenging and inefficient on today's Internet. Traditional CDN-based DASH results in latencies that are too high for Metaverse systems' stringent MTP and jitter demands. As shown in the previous sections, no existing wireless architectures can fully meet the requirements. Availability, bandwidth, and latency can be satisfied only in specific scenarios and controlled environments (see Section V). To provide low latency and high-speed transport for large user groups, we need: I) fine-grained, hierarchical media exchanges for interactive communication and coding for enhanced communication robustness and efficiency, II) low-latency exchange of arbitrary objects and data streams, which need quick access to services and real-time user interactions, III) scalability with respect to the number of users (see Section VII), IV) efficient, robust data sharing and multi-destination delivery, to address IP multicast challenges [401] (e.g., inter-domain routing, scalability, and routing communication overhead). This includes fine-granular media distribution supporting interactive and streaming needs and proper naming semantics at the network layer for efficient data sharing across multiple virtual sessions, V) using multi-access, multipath, and multicast communication where possible and local broadcast when applicable (e.g., using wireless broadcast for shared local views and low-latency interactivity without application-aware edge routers), VI) inherent secure communication with built-in features for privacy, user/data authentication, trust in communication peers, and control on the publishing side.

2) SUPPORT FOR SECURE AND DETERMINISTIC DATA TRANSFORMATION AND COMPUTING

Today's computing is generally confined to coarse-grained CDN-style computing, including MEC. Current trust and security frameworks depend on TLS connection termination, i.e., represent an overlay approach, which is not conducive to low latency communication. The dynamic, just-in-time instantiation of computing functions on application-agnostic platforms is currently unavailable (see Section VIII). What's needed is scalable multi-destination distribution and in-network replication and transformation that exposes the application object hierarchy for fine-grained retrieval and security: I) The system should support in-network processing for objects replication (i.e., multi-destination distribution) and transformation (i.e., supporting heterogeneous receiver groups and networks with varying quality layers, possibly dynamic transcoding). II) It should ensure deterministic behavior for mission-critical Metaverse applications (e.g., digital twin). The network must provide the required levels of flexibility and trustworthiness for computational offloading. This includes enabling compression and decompression at various levels (e.g., adapting the semantic communication compression to the current network status and user/application requirements) and employing more efficient encoding techniques (e.g., reducing data sizes through resolution changes, smarter encoding tiling schemes, etc). (Refer to Sections III–IX for further details).

3) MORE ADAPTIVE AND MODULAR SYSTEM DESIGN

Decentralized and Server-Based Communication Support: Centralization vs decentralized is one dimension of design choices. If 'the Metaverse' is implemented as an application overlay, it can be easily centralized. However, if the goal is to embed Metaverse support into the network, a decentralized implementation is necessary. A hierarchical structure would be required to support the scale of such applications.

Adaptive System Design: An adaptive system design is important to support a wide range of applications and networks. This involves outlining the reliability requirements of sessions at the network layer and considering suitable transport protocols tailored for Metaverse systems. Transport protocols should be adapted to different channels from the user (e.g., for updating positions with low latency or receiving remote images with more flexible delay constraints).

Modular System Concept: The system should enable flexible decomposition and function offloading, supporting heterogeneous devices and edge networks offloading (i.e., providing only the necessary data elements for rendering at different quality levels and potentially using dynamic transcoding and level-of-detail support). It should also identify the required network interfaces (e.g., determining if AR/VR can function purely as an overlay or requires infrastructure support for caching, multicasting, traffic engineering, Quality of Service (QoS), etc.).

Simplified System Design and Protocol APIs for Developers Today, descriptions and early prototypes for Metaverse systems leverage different Internet and Web protocols to provide Metaverse services (see Section IV). The use of diverse protocols (e.g., DASH for video streaming, WebRTC for live media, HTTP or proprietary ones for control communication and general data exchange) introduces complexity due to their uncoordinated congestion controls and distinct security frameworks (e.g., TLS over HTTP/HTTPS for DASH, DTLS for WebRTC). This diversity leads to complex protocol stacks, increasing the burden on developers and complicating the integration with the underlying network architecture. Newer transport protocols such as QUIC can help to alleviate the situation, albeit still with a connection-oriented paradigm. For low-latency, scalable group communication scenarios, a universal data-oriented communication abstraction can be more natural for application developers and allow for more efficient communication, e.g., in multi-destination or interactive group communication scenarios.

B. SCENARIO EXAMPLE

The following scenario illustrates the goals. Envision a scenario where a group of tourists navigates through large, communal spaces such as historical landmarks. Upon entering these ancient structures, their smart glasses overlay high-resolution 3D historical imagery and navigational maps onto their view, guiding them to their next destination. As they wander, the glasses refresh their perspective in real-time. Considering the vastness of the buildings and the tourists' varying locations, factors such as indoor/outdoor settings and network instability are unpredictable. As users move, the wireless latency and bandwidth available to their devices change, and the communication link can even fail due to handover. They can also encounter sudden bandwidth drops without an increase in latency.

To address these challenges, the system must exhibit resilience against such events, dynamically adjusting to changing network conditions. This requires adaptive and modular system architectures (as suggested in Section IX-A), capable of supporting a wide range of applications (i.e., server-based or decentralized) and networks (i.e., different quality layers), enabling flexible decomposition and function offloading (possibly with dynamic transcoding and level-of-detail support) so that devices with constrained capabilities can offload functions (e.g., rendering) to the network. Given the possibility of multiple users accessing similar content, the adoption of local wireless broadcast mechanisms can ensure low-latency, scalable multimedia communication with shared local views and low-latency interactivity. Efficient delivery and distribution of 3D historical imagery can be achieved through robust data sharing and multi-destination delivery, ensuring all users have access to high-quality content despite environmental and network variability. We will outline how to implement these technical concepts in Sections IX-C and IX-D.

C. DESIGN IDEAS

In the following, we provide some suggestions for achieving the goals and realizing the application scenarios discussed above.

1) UNIFIED PROTOCOLS AND APIS FOR INTERACTIVE REAL-TIME COMMUNICATION AND ON-DEMAND MEDIA STREAMING

Low latency exchange of objects and data streams in Metaverse systems is challenging for current CDN-based HTTP infrastructures such as DASH-based video-on-demand streaming. While DASH aims to enhance the viewport quality of immersive videos by refining tile delivery, its client-driven nature limits server-side control. The WebRTC protocol is more appropriate for multimedia interaction, but it shifts complexity to applications and lacks a cross-application method for data object exchange. Overlay approaches such as MOQ [345] and extensions such as QuicR [346], blend real-time interactive media with streaming but introduce some complexity (see Section VII). Many studies [345], [346], [402] propose algorithms, protocols, and architectures for managing and processing multimedia information. However, most of these results do not arrive to applications due to the absence of simple and usable APIs. To address the goal of simplified system design as outlined in Section IX-A, we envision more unified protocols/APIs to simplify developer access to different communications services such as media-on-demand and real-time streaming (as proposed by MoQ or ROBUST [402]). Such unified protocols should be applicable to different use cases (e.g., multimedia transport, multisensory data communication, etc.), adapt to dynamic network conditions, and provide optimization options to leverage network capabilities such as multi-path forwarding and multicast/broadcast capabilities.

2) SECURE AND DETERMINISTIC DATA TRANSFORMATION AND COMPUTING

XR is one example of the *Multisource-Multidestination Problem* [403] that combines video, haptics, and tactile experiences in interactive or networked multi-party and social interactions with a secure environment and ensures user privacy. Delivering this via a client-server cloud-based solution is challenging as it requires a combination of stream synchronization, low delays and delay variations, loss recovery mechanisms, and optimized caching and rendering near the network edge for the user (as suggested in Section IX-B). However, many of these challenges (see Section VIII for more details) are still in the realm of research focused on resolving resource allocation issues and ensuring adequate quality of experience (e.g., tackling multi-variate and heterogeneous goal optimization problems at merging nodes). In-network computing is a potential solution for improving client-server architecture, facilitating service deployment, and supporting secure and deterministic data transformation and computing goals (Section IX-A). Specifically, there are two design ideas.

Robust/Deterministic In-network Computing Capability that Supports Function Offloading, In-network Transformation and Compression: As discussed in Section VIII, in-network computing can enhance video transmission by enabling transcoding, applying advanced context-based compression algorithms, and facilitating pre-fetching, pre-caching, and prediction within the network. Advanced functional decomposition, localization, and discovery of in-network computing and storage resources can help to optimize the user experience in general. This focuses on identifying the best resources and assessing their reliability, particularly for mission-critical services in Metaverse (e.g., medicine and digital twins). For example, in ICE-AR [122], mobile devices continually transmit users' context (POV video and metadata such as IMU data) to edge nodes running real-time ML processes. The raw video is processed to generate a semantic description of the current environment (i.e., the deep context). Different edge nodes offer different subsets of context extraction services. The deep context is then used by mobile clients to retrieve relevant content from cloud providers and overlay it on the POV video. This concept is an applicable example for the scenario proposed in Section IX-B.

Secure Semantic Communication Approach Based on In-Network Computing Platform: As discussed in Section V, semantic communication can transmit semantic information instead of the actual media data, allowing for media reproduction and rendering at the receivers. It can make communication more efficient and aid in more intelligent compression, decompression, and prediction within the network. However, semantic communication is not inherently secure. It is merely an intelligent way of exchanging information [262], [263], [264]. What is missing is an in-network computing approach for '*secure*' transformations in the network. Most existing papers overlook this aspect, assuming that data transformation and transmission can be securely managed somehow. But generally, it is risky to trust these transformers without a robust security framework on the Internet. For example, it is more difficult to fake the actual audio stream than to fake some high-level semantic representation. The semantics can be 'forgery' or 'pollution,' conveying distinct meanings to different receivers and tasks [404].

Just-in-time, Segmented Content Storage and Distribution: To address the inefficiencies of centralized CDNs (and intermediaries) that often lead to bottlenecks, increased costs, and vulnerabilities to censorship and tampering, Content Fabric Protocol (CFP) [405] has been proposed as an application-specific protocol tailored for just-in-time, segmented content storage and distribution. It is a decentralized data protocol executing across a global network of hosting nodes, which dynamically serve video, imagery, applications, and other active content directly from source objects as both live and on-demand streaming, and dynamic combinations. Functionally, CFP eliminates the need for separate content transcoding, aggregation, management, and distribution services by integrating many conventional functions, including live ingest, cloud origin storage, live transcoding, content management, encryption/digital rights management (DRM), program sequencing, rights management, CDN streaming, and static content distribution. CFP creates and delivers output, such as adaptive bitrate streaming manifests and segments and static content, through a just-in-time process executed within the network's nodes. This process generates a data structure called a 'content object', a component object representation consisting of the media essence, metadata, and code. A content object consisting of references to cryptographically hashed signatures of the binary 'parts' that make up the object. Upon ingestion, master file media, assets, data, or streams are broken down into 'parts', distributed across the network, and compiled into an object. Reusing these 'parts' within an object is by reference, not copy, reducing redundancy. Parts are only copied when updated, avoiding unnecessary file copies across the network and in storage, thus enhancing efficiency. It also includes a fast routing algorithm that enables real-time location of 'parts' in the network, supporting just-in-time transcoding, packaging, and extensible audio/video processing.

3) DATA-ORIENTED APPROACH FOR REPRESENTING 3D CONTENT AND MEDIA OBJECTS

Today's Web primarily operates as a data-centric application layer, with data identified by URIs and managed through REST primitives. However, this creates a semantic gap with the underlying host-oriented transport, leading to complexity, centralization, and brittleness problems. Metaverse systems can be viewed as an extension of the Web into 3D interaction and immersion, optionally overlaid on the physical world. Rather than rendering data objects onto a 2D page within a device, they are rendered within a shared 3D space, interacting among themselves and with users [42]. Data-oriented communication can treat virtual content as secure data objects and distribute them efficiently among a wider peer group, retrieving only the necessary data to reconstruct an appropriate representation while considering the constraints of user devices and access networks. Interactions between private and shared 3D objects can be simplified if these objects use similar conventions but with varying security levels [42]. This contributes to achieving the goal of low latency and scalable media communication (see Section IX-A), with fine-grained media distribution that supports both interactive and streaming content (also security).

4) ADAPTIVE AND MODULAR SYSTEM DESIGN

Adaptive and modular system design aims to create flexible systems capable of dynamically offloading functions, such as rendering, to improve performance and adaptability. This approach requires a modular structure where components can be readily adjusted or offloaded to external servers. For instance, Weber et al. [406] introduced a modular multi-user XR framework that can be tailored to various applications. Servo [399] is a modular serverless back-end system for Metaverse systems, offering fine-grained virtual-world scalability through a collection of serverless components. ARENA [400] is a modular architecture for secure, lightweight serverless-style computing using REST and Pub-Sub network patterns for interactive virtual spaces. In general, an adaptive, modular system design requires a holistic approach that integrates independent functions with clear interfaces for seamless operation and transfer. It should include strategic server selection and communication for effective task offloading while prioritizing data and function security to maintain trust and integrity. Intelligent decision-making is needed for determining optimal offloading times and tasks, considering system load, performance goals, and user experience. The system must incorporate smart decision-making and robust optimization to determine the best offloading opportunities, balancing system load, performance, and resources based on network analysis and offloading benefits (expanding on Section IX-B).

D. SUGGESTED RESEARCH DIRECTIONS

The design ideas can be deployed and implemented in different ways. We suggest the following specific research directions.

1) INFORMATION-CENTRIC SYSTEM DESIGN

As discussed in Section IX-C, Web and Metaverse applications are inherently data-oriented (on the application layer). When we conceive 'the Metaverse' not merely as an application running on the current network but as an evolution of the network itself, we can narrow rather than expand the gap between network architecture and application semantics. Specifically, 'the Metaverse' can be seen as an information-centric system where applications participate in granular 3D content exchange, context-aware integration with the physical world, and other Metaverse-relevant services [42]. Application layer data structures in Metaverse (e.g., 3D models and scene descriptions) are based on object hierarchies, in which connection-based systems cannot fully use. ICN generally enables direct data-oriented communication, providing access to granular, individually secured objects, e.g., making up a video stream, directly by name as needed by applications, without relying on channel-based abstraction. It provides request-response semantics at the network layer that are similar to Web semantics, but at packet granularity, operating without host addressing or name-to-address mappings such as those used in the Domain Name System (DNS).

However, several aspects still require investigation. First is the actual deployment of ICN over the Internet. The initial deployments require the establishment of an overlay topology over the current Internet infrastructure. While the core technologies (i.e., different interfaces and underlay protocols) are fundamentally ready, other issues related to the deployment and efficient operation of such overlays remain (e.g., shortest path communication, routing and reliability). Moreover, actual systems would need specific strategies for in-network optimizations (e.g., pre-fetching and re-transmissions) and quality adaptation (e.g., layered coding and in-network transcoding). For a data-oriented Metaverse, where data related to a virtual world could be compiled into collections (e.g., FLIC) or grouped using manifests (similar to DASH video streaming), the ability to extend these technologies to support dynamically creating objects is important [136]. Additionally, these data require multiple levels of access control. How to represent and organize these ownership levels, particularly in a distributed manner, remains a challenge for ICN [42]. Lastly, concepts and mechanisms for privacy, selective attention, content filtering, autonomous interactions as well as ownership and control on the publisher's side are required (see Section VII). Next, we suggest three specific research directions for supporting Metaverse systems via data-oriented techniques: *enhanced QoS mechanisms*, *local broadcast/multicast support* and *leveraging context-aware networking*.

Enhanced QoS Mechanisms: As discussed in Section V, the concept of QoS is akin to managing 'managed unfairness' in network traffic, ensuring that high-priority tasks, such as audio streams in multimedia applications, are given precedence over fewer critical data packets [223]. However, the practical implementation of QoS in networking faces several challenges [407]. IntServ's requirement for signaling makes it difficult to implement across different domains, while DiffServ offers a simpler model but still lacks usage in inter-domain scenarios. These challenges are compounded by the limited controllable resources in IP networks, primarily queue space, which restricts the effectiveness of traditional QoS management approaches. ICN's inherent statefulness within the data plane allows for a rethinking of QoS mechanisms without relying on separate signaling protocols such as RSVP. Unlike unicast IP addresses, ICN's non-topological naming allows for straightforward QoS application in multi-destination and multipath environments, rather than requiring either multicast with coarse class-based scheduling or complex signaling such as Resource Reservation Protocol (RSVP-TE) [408], [409]. Additionally, IP has three forwarding semantics (i.e., unicast, anycast and multicast), with different QoS needs [410], [411]. ICN has one single forwarding semantic, so any QoS mechanisms can be uniformly applied across any request/response invocation, regardless of the forwarding strategy employed (whether the forwarder employs dynamic destination routing, multi-destination forwarding with next hops tried serially, multi-destination with next hops used in parallel, or even localized flooding such as directly on Layer 2 multicast mechanisms) [412]. However, ICN-based QoS is still under development and is not as stable as IP-based QoS. We suggest further research and experiments on fine-tuning interest aggregation, caching and their impact on receiver-based performance estimation and develop a specific QoS mechanisms [223] for priority of key requests, such as prioritizing interactive data and baseline-quality media objects over higher quality objects (e.g., MoQ can prioritize audio streams).

Supporting Local Broadcast/Multicast Through Data-Oriented Approaches: In Metaverse systems, users can be participating in the same scene with potentially distinct viewpoints (Section IX-B), leading to overlapping data requests centered around a common point of interest (e.g., the ball in a basketball game) [349], [413]. The essential requirement is not merely wide broadcast capabilities but rather the facilitation of large-scale, multi-destination communication through implicit network replication. However, scalable multi-destination transport services in Metaverse systems faces several challenges. One such challenge is the limited use of local broadcast/multicast in Metaverse systems, especially in wireless networks where most connections are unicast. Internet Service Providers (ISPs) often block multicast traffic unless it serves their internal needs, because ISPs typically block multicast traffic that is not explicitly used by themselves internally [414]. Another challenge in distributing content is the need to deliver identical bits of information to a group of receivers. This task can be complicated by current systems' Digital Rights Management (DRM) policies or privacy concerns [415]. Interruptions, rights management issues, and other factors can prevent all recipients from receiving the same bits. For example, in scenarios such as commercial media streaming (e.g., Netflix), each media stream is encrypted on an individual basis [416], which inherently complicates the multicast distribution of identical information to all users. ICN inherently supports native multicast capabilities [349], reducing bandwidth consumption without extra signaling overhead by naturally aggregating Interests (requests for data) for common data. ICN enables the network to provide an implicit multi-destination delivery services, facilitating shared local views and low-latency interactivity in a manner that does not require application-awareness on the part of edge routers [136]. Moll et al. [417] have demonstrated that implementing the ICN architecture for inter-server game state synchronization within large server clusters can reduce traffic compared to traditional IP-based infrastructures.

Leveraging Context-aware Networking: Context-aware networking with ICN in Metaverse systems can adapt data dissemination by using both service and network contexts. *I) Recognizing Service Context.* In ICN, end-systems that understand service or application contexts can flexibly (pre-)fetch data, determine appropriate quality levels, and choose different modes of communication (e.g., low-latency streaming or reliable bulk data transfer) that are implemented on top of the fundamental Interest/Data protocol. This adaptability allows the system to meet the specific needs of different services. For example, a virtual meeting requires low-latency streaming, whereas downloading large assets benefits from reliable bulk data transfer. Using QoS concepts described in [223], it would also be possible to define equivalence classes flexibly and use the rich set of controllable resources (queues, content store, pending interest tables) to enforce preferential treatment of important application traffic where needed. *II) Recognizing Network Context.* ICN's forwarding layer inherently supports network context awareness. Forwarders can independently determine name prefix reachability, perform self-learning for determining forwarding information, measure dynamic per-prefix latency, and decide on suitable forwarding strategies (e.g., Best Route or Interest Broadcasting), ensuring optimal data routing based on real-time network conditions. Using technologies such as ICN Pathsteering [418], consumer endpoints can learn about suitable path options and then control path selection in a source-routing manner, depending on current observed network performance.

2) LARGE-SCALE REAL-TIME MEDIA DISTRIBUTION

As detailed in Section VII, the rapid expansion of large-scale real-time applications in Metaverse systems is technically challenging to developers and architects, which raises the question of the suitability of the current protocol stack. Traditional real-time video conferencing platforms are tailored for controlled enterprise networks, whereas real-time Metaverse applications require efficient media transfer across the broader Internet (including mobile and heterogeneous networks with best-effort connectivity). The traditional RTC stack, typically running over UDP, prioritizes the timeliness of data transmission rather than reliability, leaving the tasks of ensuring resiliency and managing congestion to the applications themselves. However, given the complex and highly interactive nature of Metaverse applications (which often involve extensive user-generated content and dynamic user interactions), the existing protocol stack cannot adequately support real-time Metaverse services.

Since 2018, there has been a discussion on the evolution of real-time multimedia transport [310], [345] (e.g., DASH, QUICR [346], RUSH [419], WARP [343], RoQ [420]). Recently, Media over QUIC has been discussed as a potential solution for low-latency video distribution(see Section VII); ICN-RTC [421] is a scalable real-time communication architecture that modifies media switching in WebRTC. Based on the standard WebRTC Selective Forwarding Units (SFU)-based architectures, it realizes media flow switching [422] and maps it to a request-reply transport stack both at clients and SFUs. ROBUST [402] was developed to solve the problem of reliability in real-time applications with a high level of network control, quick adaptation to variations (e.g., congestion, cross traffic and network mobility), and scalable multi-destination distribution. However, a major challenge in deploying these protocols for large-scale, real-time Metaverse applications is their integration with the application and interaction with application-level proprietary reliability/rate adaptation mechanisms [423]. There is a need for closer interaction with the application by offering APIs and possibly integrating the communication protocol as an SDK directly within the application framework, rather than as an external module (refer to Section IX-C). Moreover, what is still missing is the ability to systematically use edge relays for in-path low-latency re-transmissions and possibly other in-path recovery, rate, and control functions [424]. Additionally, protocol optimization is needed to reduce protocol overhead (e.g., through aggregated Interest requests) and to automate strategy selection based on application objectives [425].

3) COMPUTING IN THE NETWORK

Computing in the network can serve as an integral component for better communication and computation support. By storing and preprocessing scenes in local elements (e.g., in the mobile network), it can extend the reach of applications over the network's edge. It can also improve video transmission through better transcoding, context-based compression (e.g., semantic communication), and predictive techniques for movement, caching, and fetching (see Sections V and VIII). Furthermore, it facilitates the monitoring and distribution of services across collaborative network elements, enhancing end-to-end performance by ensuring that computational tasks are carried out closer to the data source or end-user. Essentially, it can support everything from full application offloading to decomposing an application into small code snippets (e.g., at class, objects, or function granularity). These snippets can be distributed and executed throughout the network following the application's control flow, enabling execution models from iterative to recursive calls and from applications on the initiating host to mobile one.

Compute first networking (CFN) [426] is an example of a 'computing in the network system' that is based on computation graph representation for distributed programs. These programs consist of stateful actors and functions dynamically instantiated on available compute resources (e.g., real-time health monitoring system). CFN's core concept is to offer a general-purpose distributed computing framework that can be programmed without detailed knowledge of the runtime environment while automatically and efficiently leveraging dynamic resource properties. In CFN, compute nodes that executing functions within a given program instance are called workers. It can dynamically select which worker to use, optimizing processes such as instantiating functions near large data inputs. The control over which execution platforms host specific program interfaces (or individual functions/actors) is maintained through a computation graph. Distributed scheduling is implemented as workers in every resource pool announce and disseminate information about their availability to all workers in the pool. Worker execution environments can independently determine which workers prefer to invoke a function or instantiate an actor without relying on a centralized scheduler. However, it is still challenging to implement CFN as a platform that can enable Metaverse use cases and business opportunities. One challenge is accounting for the cost of change, including balancing the time to compute the solution against the time before the configuration (due to changes in the workload and/or the resource availability), the cost of activating and deactivating resources and the optimization of value (rather than cost) through time (on a longer timescale).Another issue involves reconciling the policy preferences and constraints of both CFN operators and users. Given that different owners can provide resources (e.g., those controlling the radio access network, distributed computing resources, or computing at a Point of Presence (POP)), establishing trust in the computing functions and their outputs becomes a complex problem. Crowcroft et al. [427] discussed research directions around computing in the network.

4) DECENTRALIZED/DISTRIBUTED METAVERSE ARCHITECTURES

Most Metaverse systems today assume a cloud-based system architecture where identities and trust among them are anchored through a centralized administrative structure, with communication mediated by servers and an extensive CDN overlay infrastructure operated by the administration (Section IV). This centralization can pose issues in terms of control, as well as performance and efficiency. Despite operating on named data principles conceptually, such systems typically employ traditional layering approaches that restrict new ways of interacting (e.g., using data formats such as USD and gITF) and do not support flexible distributed computing from edge to cloud. Next, we suggest two specific research directions for these issues: *named data microverse* and *identity management and security support.*

Named Data Microverse project [428] aims to integrate insights from AR, non-linear 2D media, and game engine synchronization research into real-time extended reality platforms using ICN. It aligns with the early stages of commercial ICN applications in fields such as operational technology, tactical networks, and content distribution. It aims to balance scalability and market-based innovation with principles of democratization, trustworthiness, and fair empowerment of individuals. Specifically, the Named Data Microverse is conceived as a Metaverse with 3D data representation, where objects are independently published and accessed by any single service. The system adopts a hierarchical structure for naming and organizing 3D objects, incorporating core ICN functions such as peer rendezvous, namespace claiming, certificate generation, and in-browser repository storage. It focuses on peer synchronization and data retrieval strategies to showcase a decentralized virtual space where users can publish, manipulate, and cache 3D objects. Designed for mobile compatibility, the platform is developed using HTML/JavaScript and leverages well-established frameworks. The research details and practices mainly on three aspects. *I) User Experience.* Involving collaborative 3D object creation and manipulation with asynchronous events and visualization of user information and object security properties derived from ICN. *II) Networking & Distributed Computing.* Covering secure bootstrapping for trust establishment, low-latency state synchronization, multi-interface/multi-path connectivity, supporting for infrastructure-less environments, multi-destination delivery models, and potential for future computation offload. (This concept is similar to what we discussed in the *'Computing In the Network'* direction.) *III) Programming APIs & Data Structures.* Enabling modular scene descriptions using USD objects as individual ICN data objects. Having a composition framework with linking concepts and manifests for integration, incremental updates to collections, and information-centric APIs for JavaScript.

As a developing research area, it requires more work, including evaluating scalability and performance to understand how the system can scale with an increasing number of users and objects, identifying bottlenecks and optimizing for high-frequency updates and low latency. It is important to ensure secure bootstrapping and robust trust mechanisms, facilitating verifiable and secure interactions within the microverse. Assessing the intuitiveness and efficiency of the user interface and interaction models is also needed, especially for collaborative 3D object manipulation. Additionally, it is necessary to ensure system compatibility across various devices and platforms, with a particular focus on mobile usability, and continuously evaluate the integration of new technologies and frameworks to enhance functionality and performance.

Distributed open realm systems need solutions in **identity management and security support** that enable interoperability among multiple systems and a diverse user population. Mechanisms to support trust are inherently coupled with various identities, from 'real world' identities to application-specific identities that users can adopt in different contexts. However, it is difficult to integrate and interoperate in a shared Metaverse environment without centralized management when nodes must use host-centric paradigms to address not only data interactions but also the underlying service connections and security relationship. It also exacerbates the impact of intermittent connectivity on interactivity when global networking is required for functions such as rendezvous [429] (which are handled locally in the ICN). Public interest in technologies such as blockchain indicates growing concern over control concentration. However, these emerging technologies are primarily built on the current Internet architecture, which does not bridge the gap between new network visions and its fundamental technical approaches [428]. Developed over forty years ago, core Internet protocols were designed for connecting machines under an implicit trust model, which still persists in today's secure Web protocols. Ideally, society's data network should shift from implicit trust based on data pipe ownership to explicit trust in the data itself, potentially reversing control centralization trends. Today, achieving cross-platform interoperability and visualization without centralized hubs is impractical [430], making it challenging to create secure, fine-grained data flows required for interactions among co-existing 3D elements in a virtual world. Solutions should consider not just media asset exchange but also the interactions among objects and the data flows needed to support it. What should be done next is a further exploration of security technologies in Metaverse systems, including system bootstrapping, trust establishment, and authenticated information discovery, especially considering cross-layer designs to reconcile trust layer disconnects in many existing systems.

X. CONCLUSION

The term 'Metaverse' is deliberately vague and refers to a range of different concepts, business ideas, application types, and enabling technologies. For a technical discussion, it is important to understand the relationship of applications and their underlying enabling technologies. 'The Metaverse' integrates concepts and technologies from different domains, namely communications, networking, distributed computing, AR/VR and UX design. From a networked systems perspective, we observe that although some (mostly proposed) Metaverse applications impose new levels of communication performance requirements, most current systems are largely built on top of existing technologies and protocol stacks, which can lead to problems when scaling systems to larger numbers of users and when incorporating new media types and encoding, e.g., holographic video. Social VR systems provide more stringent requirements than today's WebRTC-based tele-conferencing systems in terms of media quality (resulting in larger data volumes) and low-latency communication. Meeting these requirements is not merely a question of transmission speed, deterministic networking guarantees, and new mobile network generations (e.g., 6G).

We argue that it is important to consider key technical capabilities such as scalable low-latency media distribution and interactive communications, support for link-layer broadcast/multicast, and the ability to accommodate in-network computing reliably and securely. Enabling these capabilities requires an architectural change in building demanding large-scale networked multimedia applications. We argue that a new holistic design should consider the following goals: I) low latency and scalable media communication, II) support for secure and deterministic data transformation and computing, and III) more adaptive and modular system design.

These goals should be addressed by I) unified protocols and APIs for interactive real-time communication and on-demand media streaming, II) in-network computing capabilities that support function offloading, in-network transformation, and compression, III) a data-oriented approach for representing 3D content and media objects, and IV) adaptive and modular system design.

These design ideas suggest further research in different directions: I) information-centric system design, II) large.scale real-time media distribution, III) computing in the network, and IV) distributed and decentralized Metaverse architectures.

Information-centricity is a concept that affects many aspects of a Metaverse system design. On the application layer, many interactions *do* exhibit information-centric properties: Web protocols, namely HTTP, DASH-based/inspired media streaming, as well as functional RPC communication are all based on the notion of accessing named information – either as static data or as dynamic computation results. Recent technology development and standardization activities such as MOQ have introduced some information-centric principles (e.g., named data in a network of relays/caches and publish-subscribe for dynamic data collections). Future Metaverse systems can embrace these concepts and extend their application from the application layer to the network layer – to form a general, native communication platform to enable many relevant capabilities that we discussed in this paper, such as better in-network support for multipath-communication and load balancing, support of native broadcast services, more robust in-network computing, and more decentralized system designs that enable more direct, more efficient communication.

Thinking this further, the design of such systems is expected to reveal more insights into opportunities for applying such concepts not only to Metaverse systems but more broadly to the design of a general data-oriented Web framework, i.e., a Web framework where the inherent information-centric nature of the application maps directly to accessing secure Web objects in networks [431].

REFERENCES

[1] E. Games. "Unreal networking architecture." 2024. [Online]. Available: https://docs.unrealengine.com/udk/Three/NetworkingOverview.html

[2] R. Cheng, N. Wu, M. Varvello, S. Chen, and B. Han, "Are we ready for metaverse? A measurement study of social virtual reality platforms," in *Proc. 22nd ACM Internet Meas. Conf.*, 2022, pp. 504–518.

[3] Mozilla. "The client-side code for mozilla hubs." 2023. [Online]. Available: https://github.com/mozilla/hubs

[4] B. Guidi and A. Michienzi, "Social games and blockchain: Exploring the metaverse of decentraland," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, 2022, pp. 199–204.

[5] E. U. Haque et al., "Scalable edgeIoT blockchain framework using EOSIO," *IEEE Access*, vol. 12, pp. 41763–41772, 2024.

[6] J. Tanner and R. Khan, "Technology review of blockchain data privacy solutions," 2021. *arXiv:2105.01316*.

[7] W. Zhang et al., "EDGEXAR: A 6-DoF camera multi-target interaction framework for MAR with user-friendly latency compensation," in *Proc. ACM Human–Comput. Interact.*, vol. 6, 2022, pp. 1–24.

[8] W. C. Ng, W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, and C. Miao, "Unified resource allocation framework for the edge intelligence-enabled metaverse," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 5214–5219.

[9] I. Aliyu, N. Ko, T.-W. Um, and J. Kim, "A dynamic partial computation offloading for the metaverse in in-network computing," 2023, *arXiv:2306.06022*.

[10] L. Wang, W. Su, F. Song, and I. You, "Integrating smart computility for subflow orchestration in remote virtual services," *IEEE J. Biomed. Health Informat.*, early access, Oct. 16, 2023, doi: 10.1109/JBHI.2023.3325133.

[11] J. D. N. Dionisio, W. G. B. Iii, and R. Gilbert, "3D virtual worlds and the metaverse: Current status and future possibilities," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 1–38, 2013.

[12] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 153–161.

[13] H. Ning et al., "A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14671–14688, Aug. 2023.

[14] Q.-V. Pham et al., "Artificial intelligence for the metaverse: A survey," 2022, *arXiv:2202.2022*.

[15] Q. Yang, Y. Zhao, H. Huang, Z. Xiong, J. Kang, and Z. Zheng, "Fusing blockchain and Ai with metaverse: A survey," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 122–136, 2022.

[16] D. Panagiotakopoulos, G. Marentakis, R. Metzitakos, I. Deliyannis, and F. Dedes, "Digital scent technology: Toward the Internet of Senses and the metaverse," *IT Prof.*, vol. 24, no. 3, pp. 52–59, 2022.

[17] M. Xu et al., "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 1, pp. 656–700, 1st Quart., 2023.

[18] L.-H. Lee et al., "All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda," 2021, *arXiv:2110.05352*.

[19] Y. Wang and J. Zhao, "A survey of mobile edge computing for the metaverse: Architectures, applications, and challenges," 2022, *arXiv:2212.00481*.

[20] Y. Wang and J. Zhao, "Mobile edge computing, metaverse, 6G wireless communications, artificial intelligence, and blockchain: Survey and their convergence," in *Proc. IEEE 8th World Forum Internet Things (WF-IoT)*, 2022, pp. 1–8.

[21] T. A. Syed et al., "In-depth review of augmented reality: Tracking technologies, development tools, AR displays, collaborative AR, and security concerns," *Sensors*, vol. 23, no. 1, p. 146, 2022.

[22] T. Q. Duong, D. Van Huynh, S. R. Khosravirad, V. Sharma, O. A. Dobre, and H. Shin, "From digital twin to metaverse: The role of 6G ultra-reliable and low-latency communications with multi-tier computing," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 140–146, Jun. 2023.

[23] M. Ali, F. Naeem, G. Kaddoum, and E. Hossain, "Metaverse communications, networking, security, and applications: Research issues, state-of-the-art, and future directions," 2022, *arXiv:2212.13993*.

[24] L. Zhang, Q. Du, L. Lu, and S. Zhang, "Overview of the integration of communications, sensing, computing, and storage as enabling technologies for the metaverse over 6G networks," *Electronics*, vol. 12, no. 17, p. 3651, 2023.

[25] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1160–1192, 2nd Quart., 2021.

[26] M. Erol-Kantarci and S. Sukhmani, "Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G," in *Proc. Ad Hoc Netw. 9th Int. Conf. (AdHocNets)*, 2018, pp. 169–177.

[27] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[28] "Introducing the Ai research supercluster-meta's cutting-edge Ai supercomputer for Ai research." 2022. [Online]. Available: https://ai.meta.com/blog/ai-rsc/

[29] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[30] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[31] L.-H. Lee et al., "When creators meet the metaverse: A survey on computational arts," 2021, *arXiv:2111.13486*.

[32] G.-J. Hwang and S.-Y. Chien, "Definition, roles, and potential research issues of the metaverse in education: An artificial intelligence perspective," *Comput. Educ. Artif. Intell.*, vol. 3, Jun. 2022, Art. no. 100082.

[33] K. Yoo, R. Welden, K. Hewett, and M. Haenlein, "The merchants of meta: A research agenda to understand the future of retailing in the metaverse," *J. Retailing*, vol. 99, no. 2, pp. 173–192, Jun. 2023.

[34] G. D. Ritterbusch and M. R. Teichmann, "Defining the metaverse: A systematic literature review," *IEEE Access*, vol. 11, pp. 12368–12377, 2023.

[35] S.-M. Park and Y.-G. Kim, "A metaverse: Taxonomy, components, applications, and open challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022.

[36] A. M. Al-Ghaili et al., "A review of metaverse's definitions, architecture, applications, challenges, issues, solutions, and future trends," *IEEE Access*, vol. 10, pp. 125835–125866, 2022.

[37] B. Ryskeldiev, Y. Ochiai, M. Cohen, and J. Herder, "Distributed metaverse: Creating decentralized blockchain-based model for peer-to-peer sharing of virtual spaces for mixed reality applications," in *Proc. 9th Augmented Human Int. Conf.*, 2018, pp. 1–3.

[38] H. Kanematsu et al., "Skype discussion for PBL between two laboratories and students biological/psychological responses," *Procedia Comput. Sci.*, vol. 112, pp. 1730–1736, Jan. 2017.

[39] H.-S. Choi and S.-H. Kim, "A content service deployment plan for metaverse museum exhibitions—Centering on the combination of beacons and HMDS," *Int. J. Inf. Manag.*, vol. 37, no. 1, pp. 1519–1527, 2017.

[40] A. Zackery, P. Shariatpanahi, M. M. Zolfagharzadeh, and A. A. Pourezzat, "Toward a simulated replica of futures: Classification and possible trajectories of simulation in futures studies," *Futures*, vol. 81, pp. 40–53, Aug. 2016.

[41] J. C. Chen, "The crossroads of English language learners, task-based instruction, and 3D multi-user virtual learning in second life," *Comput. Educ.*, vol. 102, pp. 152–171, Nov. 2016.

[42] D. Kutscher, J. Burke, G. Fioccola, and P. Mendes, "STATEMENT: The metaverse as an information-centric network," 2023, *arXiv:2309.09147*.

[43] M. A. González, B. S. N. Santos, A. R. Vargas, J. Martín-Gutiérrez, and A. R. Orihuela, "Virtual worlds. Opportunities and challenges in the 21st century," *Procedia Comput. Sci.*, vol. 25, pp. 330–337, Mar. 2013.

[44] K. J. Nevelsteen, "Virtual world, defined from a technological perspective and applied to video games, mixed reality, and the metaverse," *Comput. Animation Virtual Worlds*, vol. 29, no. 1, 2018, Art. no. e1752.

[45] D. M. Barry et al., "Evaluation for students' learning manner using eye blinking system in metaverse," *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 1195–1204, 2015.

[46] A. Moldoveanu, A. Gradinaru, O.-M. Ferche, and L. Ştefan, "The 3D UPB mixed reality campus: Challenges of mixing the real and the virtual," in *Proc. 18th Int. Conf. Syst. Theory Control Comput. (ICSTCC)*, 2014, pp. 538–543.

[47] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7573–7586, Nov. 2019.

[48] M. Tang, L. Gao, and J. Huang, "Enabling edge cooperation in tactile Internet via 3C resource sharing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 11, pp. 2444–2454, Nov. 2018.

[49] G. P. Fettweis, "The tactile Internet: Applications and challenges," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 64–70, Mar. 2014.

[50] X. Yu, D. Owens, and D. Khazanchi, "Building socioemotional environments in metaverses for virtual teams in healthcare: A conceptual exploration," in *Proc. 1st Int. Conf. Health Inf. Sci. (HIS)*, 2012, pp. 4–12.

[51] W. C. Ng, W. Y. B. Lim, J. S. Ng, S. Sawadsitang, Z. Xiong, and D. Niyato, "Optimal stochastic coded computation offloading in unmanned aerial vehicles network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[52] D. Virgilio. "What comparisons between second life and the metaverse MISS." 2022. [Online]. Available: https://slate.com/technology/2022/02/second-life-metaversefacebook-comparisons.html

[53] H. Jeong, Y. Yi, and D. Kim, "An innovative e-commerce platform incorporating metaverse to live commerce," *Int. J. Innov. Comput. Inf. Control*, vol. 18, no. 1, pp. 221–229, 2022.

[54] S.-C. Yoo, D. Piscarac, and S. Kang, "Digital outdoor advertising tecoration for the metaverse smart city," *Int. J. Adv. Culture Technol.*, vol. 10, no. 1, pp. 196–203, 2022.

[55] Y. K. Dwivedi et al., "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *Int. J. Inf. Manag.*, vol. 66, Oct. 2022, Art. no. 102542.

[56] T.-H. The, Q.-V. Pham, X.-Q. Pham, T. Do-Duy, and T. R. Gadekallu, "Ai and computer vision technologies for metaverse," in *Metaverse Communication and Computing Networks: Applications, Technologies, and Approaches*. Hoboken, NJ, USA: Wiley, 2023, pp. 85–124.

[57] K. Laeeq. "Metaverse: Why, how and what." 2022. [Online]. Available: https://www.researchgate.net/profile/Kashif-Laeeq/publication/358505001_Metaverse_Why_How_and_What/links/62053bb0afa8884cabd70210/Metaverse-Why-How-and-What.pdf

[58] A. Sipatchin, S. Wahl, and K. Rifai. "Eye-tracking for low vision with virtual reality (VR): Testing status QUO usability of the *HTC VIVE* pro eye." 2020. [Online]. Available: https://doi.org/10.1101/2020.07.29.220889

[59] M. Conti, J. Gathani, and P. P. Tricomi, "Virtual influencers in online social media," *IEEE Commun. Mag.*, vol. 60, no. 8, pp. 86–91, Aug. 2022.

[60] A. Hollowell. "How Seoul is creating a metaverse for a smarter city," 2022. [Online]. Available: https://venturebeat.com/ai/how-seoul-is-creating-a-metaverse-for-asmarter-city/

[61] BizIn. "Seoul became the first metaverse city in the world." 2022. [Online]. Available: https://thebizin.com/international/seoul-became-the-first-metaverse-city-inthe-world/

[62] Y. Han, D. Guo, W. Cai, X. Wang, and V. C. Leung, "Virtual machine placement optimization in mobile cloud gaming through QoE-oriented resource competition," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 2204–2218, Jul./Sep. 2020.

[63] B. of Apps. "Pokémon go statistics." 2022. [Online]. Available: https://www.businessofapps.com/data/pokemon-go-statistics/

[64] *Beat Saber*, Beat Games, Prague, Czech, 2019.

[65] E. Games, *Fortnite*, EPIC Games, Prague, Czech, 2017.

[66] A. Siyaev and G.-S. Jo, "Towards aircraft maintenance metaverse using speech interactions with virtual objects in mixed reality," *Sensors*, vol. 21, no. 6, p. 2066, 2021.

[67] P. Maharg et al., "Simulations, learning and the metaverse: Changing cultures in legal education," *J. Inf. Law Technol.*, vol. 1, pp. 1–28, Sep. 2007.

[68] Y. Song, C. Guo, P. Xu, and J. Wang, "Design of deterministic transmission framework for distributed power system based on digital twin," in *Proc. IEEE 5th Conf. Energy Internet Energy Syst. Integrat. (EI2)*, 2021, pp. 3391–3395.

[69] M. Ball, *The Metaverse: and How It Will Revolutionize Everything*. New York, NY, USA: Liveright, 2022.

[70] B. Shi, J. Yang, Z. Huang, and P. Hui, "Offloading guidelines for augmented reality applications on wearable devices," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1271–1274.

[71] A. Tlili et al., "Is metaverse in education a blessing or a curse: A combined content and bibliometric analysis," *Smart Learn. Environ.*, vol. 9, no. 1, pp. 1–31, 2022.

[72] Y. Wang, L.-H. Lee, T. Braud, and P. Hui, "Re-shaping post-COVID-19 teaching and learning: A blueprint of virtual-physical blended classrooms in the metaverse era," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, 2022, pp. 241–247.

[73] M. Aloqaily, O. Bouachir, F. Karray, I. Al Ridhawi, and A. El Saddik, "Integrating digital twin and advanced intelligent technologies to realize the metaverse," *IEEE Consum. Electron. Mag.*, vol. 12, no. 6, pp. 47–55, Nov. 2023.

[74] M. S.-Y. Jong, C.-C. Tsai, H. Xie, and F. K.-K. Wong, "Integrating interactive learner-immersed video-based virtual reality into learning and teaching of physical geography," *Brit. J. Educ. Technol.*, vol. 51, no. 6, pp. 2064–2079, 2020.

[75] R. D. Pinto, B. Peixoto, M. Melo, L. Cabral, and M. Bessa, "Foreign language learning gamification using virtual reality—A systematic review of empirical research," *Educ. Sci.*, vol. 11, no. 5, p. 222, 2021.

[76] P. Backlund, H. Engstrom, C. Hammar, M. Johannesson, and M. Lebram, "SIDH—A game based firefighter training simulation," in *Proc. 11th Int. Conf. Inf. Visualization (IV)*, 2007, pp. 899–907.

[77] X. Zhang et al., "Metaverse for cultural heritages," *Electronics*, vol. 11, no. 22, p. 3730, 2022.

[78] D. Martins, L. Oliveira, and A. C. Amaro, "From co-design to the construction of a metaverse for the promotion of cultural heritage and tourism: The case of AMIAIS," *Procedia Comput. Sci.*, vol. 204, pp. 261–266, 2022, doi: 10.1016/j.procs.2022.08.031.

[79] N. Kelly and M. Foth. "An entire pacific country will upload itself to the metaverse. It'sa desperate plan–with a hidden message." 2023. [Online]. Available: https://theconversation.com/an-entire-pacific-country-will-upload-itself-to-the-metaverse-its-a-desperate-plan-with-a-hidden-message-194728

[80] B. Barba, D. V. L.-A. Mat, A. Gomez, and J. Pirovich, "Discussion paper: First nations' culture in the metaverse," in *Proc. SSRN*, 2022, pp. 1–24.

[81] K. John, L. Kogan, and F. Saleh, "Smart contracts and decentralized finance," *Annu. Rev. Financ. Econ.*, vol. 15, pp. 523–542, Jul. 2023.

[82] Y. Cai, J. Llorca, A. M. Tulino, and A. F. Molisch, "Ultra-reliable distributed cloud network control with end-to-end latency constraints," *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2505–2520, Dec. 2022.

[83] S. Wang et al., "Explainable AI for B5G/6G: Technical aspects, use cases, and research challenges," 2021, *arxiv.org/abs/2112.04698*

[84] N.-N. Zhou and Y.-L. Deng, "Virtual reality: A state-of-the-art survey," *Int. J. Autom. Comput.*, vol. 6, no. 4, pp. 319–325, 2009.

[85] L. Zhang, M. Peng, W. Wang, Z. Jin, Y. Su, and H. Chen, "Secure and efficient data storage and sharing scheme for blockchain-based mobile-edge computing," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 10, 2021, Art. no. e4315.

[86] J.-S. Choi, S.-J. Chang, H.-R. Lee, and H.-G. Byun, "Olfactory interaction based on ISO/IEC 23005 standard," *J. Sensor Soc.*, vol. 26, no. 5, pp. 297–300, 2017.

[87] K. Yoon, S.-K. Kim, S. P. Jeong, and J.-H. Choi, "Interfacing cyber and physical worlds: Introduction to IEEE 2888 standards," in *Proc. IEEE Int. Conf. Intell. Real. (ICIR)*, 2021, pp. 49–50.

[88] S. Kumar et al., "Second life and the new generation of virtual worlds," *Computer*, vol. 41, no. 9, pp. 46–53, 2008.

[89] S. S. A. Shah et al. "Media processing in video conferences for cooperating over the top and operator based networks." 2012. [Online]. Available: https://aaltodoc.aalto.fi/bitstreams/5177d229-4785-45b7-a3f4-32cf26459849/download

[90] S. C. Duncan, "Minecraft, beyond construction and survival," *Well Played*, vol. 1, no. 1, pp. 1–22, 2011.

[91] C. Jaynes, W. B. Seales, K. Calvert, Z. Fei, and J. Griffioen, "The metaverse: A networked collection of inexpensive, self-configuring, immersive environments," in *Proc. Workshop Virtual Environ.*, 2003, pp. 115–124.

[92] Z. Zhai et al., "Multiplane holographic imaging using the spatial light modulator," *Photonics*, vol. 10, no. 9, p. 977, 2023. [Online]. Available: https://www.mdpi.com/2304-6732/10/9/977

[93] P.-A. Blanche, "Holography, and the future of 3D display," *Light Adv. Manuf.*, vol. 2, no. 4, pp. 446–459, 2021.

[94] J. Zhang and M. Wang, "A survey on robots controlled by motor imagery brain–computer interfaces," *Cogn. Robot.*, vol. 1, pp. 12–24, May 2021.

[95] X. Gao, Y. Wang, X. Chen, and S. Gao, "Interface, interaction, and intelligence in generalized brain–computer interfaces," *Trends Cogn. Sci.*, vol. 25, no. 8, pp. 671–684, 2021.

[96] J. Wang et al., "Flexible electrodes for brain–computer interface system," *Adv. Mater.*, vol. 35, no. 47, 2023, Art. no. 2211012.

[97] A. Kamble, P. Ghare, and V. Kumar, "Machine-learning-enabled adaptive signal decomposition for a brain–computer interface using EEG," *Biomed. Signal Process. Control*, vol. 74, Apr. 2022, Art. no. 103526.

[98] R. Ghosh, "A survey of brain–computer interface using non-invasive methods," 2023, *arXiv:2309.13151*.

[99] S. R. Soekadar et al., "Future developments in brain/neural–computer interface technology," in *Policy, Identity, and Neurotechnology: The Neuroethics of Brain–Computer Interfaces*. Cham Switzerland: Springer Int. Publ., 2023, pp. 65–85.

[100] Y. Liu et al., "A novel cloud-based framework for the elderly healthcare services using digital twin," *IEEE Access*, vol. 7, pp. 49088–49101, 2019.

[101] P. Bhattacharya, M. S. Obaidat, D. Savaliya, S. Sanghavi, S. Tanwar, and B. Sadaun, "Metaverse assisted telesurgery in healthcare 5.0: An interplay of blockchain and explainable Ai," in *Proc. Int. Conf. Comput. Inf. Telecommun. Syst. (CITS)*, 2022, pp. 1–5.

[102] Y. Zhou, X. Xiao, G. Chen, X. Zhao, and J. Chen, "Self-powered sensing technologies for human metaverse interfacing," *Joule*, vol. 6, no. 7, pp. 1381–1389, 2022.

[103] H. Langley. "Inside-out V outside-in: How VR tracking works, and how it's going to change." 2017. [Online]. Available: https://www.wareable.com/vr/inside-out-vs-outside-in-vr-tracking-343

[104] Z. Yan et al., "LaserShoes: Low-cost ground surface detection using laser speckle imaging," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2023, pp. 1–20.

[105] S. Pei, P. Chari, X. Wang, X. Yang, A. Kadambi, and Y. Zhang, "ForceSight: Non-contact force sensing with laser speckle imaging," in *Proc. 35th Annu. ACM Symp. User Interface Softw. Technol.*, 2022, pp. 1–11.

[106] J. Wu, Z. Wang, A. Sarker, and M. Srivastava, "ACUITY: Creating realistic digital twins through multi-resolution pointcloud processing and audiovisual sensor fusion," in *Proc. 8th ACM/IEEE Conf. Internet Things Design Implement.*, 2023, pp. 79–92.

[107] J. Park and K. Nahrstedt, "Navigation graph for tiled media streaming," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 447–455.

[108] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," *IEEE Netw.*, vol. 33, no. 4, pp. 162–169, Jul./Aug. 2019.

[109] G. Nallathambi and J. C. Principe, "Theory and algorithms for pulse signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 8, pp. 2707–2718, Aug. 2020.

[110] S. S. Saha, S. S. Sandha, L. A. Garcia, and M. Srivastava, "TinyODOM: Hardware-aware efficient neural inertial navigation," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–32, 2022.

[111] S. S. Saha et al., "Auritus: An open-source optimization toolkit for training and development of human movement models and filters using earables," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 2, pp. 1–34, 2022.

[112] R. Cheng, N. Wu, S. Chen, and B. Han, "Reality check of metaverse: A first look at commercial social virtual reality platforms," in *Proc. IEEE Conf. Virtual Real. 3D User Interfaces Abstracts Workshops (VRW)*, 2022, pp. 141–148.

[113] R. Cheng, N. Wu, S. Chen, and B. Han, "Will metaverse be nextG Internet? Vision, hype, and reality," *IEEE Netw.*, vol. 36, no. 5, pp. 197–204, Sep./Oct. 2022.

[114] Y. Zhang, L. Zhu, G. Jiang, S. Kwong, and C.-C. J. Kuo, "A survey on perceptually optimized video coding," *ACM Comput. Surveys*, vol. 55, no. 12, pp. 1–37, 2023.

[115] X. Yang, M. Huang, L. Luo, H. Guo, and C. Zhu, "Efficient panoramic video coding for immersive metaverse experience," *IEEE Netw.*, vol. 37, no. 6, pp. 58–66, Nov. 2023.

[116] M. Hu, X. Luo, J. Chen, Y. C. Lee, Y. Zhou, and D. Wu, "Virtual reality: A survey of enabling technologies and its applications in IoT," *J. Netw. Comput. Appl.*, vol. 178, Nov. 2021, Art. no. 102970.

[117] M. H. Trabuco, M. V. Costa, B. Macchiavello, and F. A. D. O. Nascimento, "S-EMG signal compression in one-dimensional and two-dimensional approaches," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 4, pp. 1104–1113, Jul. 2017.

[118] K. Byeong-Ho, "A review on image and video processing," *Int. J. Multimedia Ubiquitous Eng.*, vol. 2, no. 2, pp. 49–64, 2007.

[119] K. Lee, J. Yi, Y. Lee, S. Choi, and Y. M. Kim, "GROOT: A real-time streaming system of high-fidelity volumetric videos," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–14.

[120] K. Wakunami et al., "Projection-type see-through holographic three-dimensional display," *Nat. Commun.*, vol. 7, no. 1, 2016, Art. no. 12954.

[121] E. C. Strinati et al., " 6G: The next Frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 42–50, Sep. 2019.

[122] J. Burke, "Browsing an augmented reality with named data networking," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2017, pp. 1–9.

[123] R. E. Pereira and M. Gheisari, "360-degree panoramas as a reality capturing technique in construction domain: Applications and limitations," in *Proc. 55th ASC Annu. Int. Conf.*, 2019, pp. 1–8.

[124] S. Shi, V. Gupta, M. Hwang, and R. Jana, "Mobile VR on edge cloud: A latency-driven design," in *Proc. 10th ACM Multimedia Syst. Conf.*, 2019, pp. 222–231.

[125] M. Tanimoto, "Overview of free viewpoint television," *Signal Process. Image Commun.*, vol. 21, no. 6, pp. 454–461, 2006.

[126] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2005.

[127] S. Aggarwal et al., "How to evaluate mobile 360 video streaming systems?" in *Proc. 21st Int. Workshop Mobile Comput. Syst. Appl.*, 2020, pp. 68–73.

[128] E. S. Wong, N. H. A. Wahab, F. Saeed, and N. Alharbi, "360-degree video bandwidth reduction: Technique and approaches comprehensive review," *Appl. Sci.*, vol. 12, no. 15, p. 7581, 2022.

[129] D. Hwang et al., "Metamaterial adhesives for programmable adhesion through reverse crack propagation," *Nat. Mater.*, vol. 22, pp. 1030–1038, Jun. 2023.

[130] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "MPEG DASH SRD: Spatial relationship description," in *Proc. 7th Int. Conf. Multimedia Syst.*, 2016, pp. 1–8.

[131] P. Zhou et al., "Vetaverse: Technologies, applications, and visions toward the intersection of metaverse, vehicles, and transportation systems," 2022, *arXiv:2210.15109*.

[132] K. Lee, J. Yi, and Y. Lee, "FarfetchFusion: Towards fully mobile live 3D telepresence platform," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–15.

[133] Y. Alkhalili, T. Meuser, and R. Steinmetz, "A survey of volumetric content streaming approaches," in *Proc. IEEE 6th Int. Conf. Multimedia Big Data (BigMM)*, 2020, pp. 191–199.

[134] C. Zhou, M. Xiao, and Y. Liu, "CLUSTILE: Toward minimizing bandwidth in 360-degree video streaming," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2018, pp. 962–970.

[135] M. Xiao et al., "Miniview layout for bandwidth-efficient 360-degree video," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 914–922.

[136] G. Fioccola, P. Mendes, J. Burke, and D. Kutscher, *Information-Centric Metaverse*, Internet Eng. Task Force, Fremont, CA, USA, Jul. 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-fmbk-icnrg-metaverse/01/

[137] C. Zhou, Z. Li, and Y. Liu, "A measurement study of oculus 360 degree video streaming," in *Proc. 8th ACM Multimedia Syst. Conf.*, 2017, pp. 27–37.

[138] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *Proc. 7th Int. Conf. Multimedia Syst.*, 2016, pp. 1–3.

[139] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop All Things Cellular Oper. Appl. Challenges*, 2016, pp. 1–6.

[140] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," in *Proc. IEEE Virtual Reality (VR)*, 2017, pp. 423–424.

[141] B. Zhang, T. Teixeira, and Y. Reznik, "Performance of low-latency dash and HLS streaming in mobile networks," *SMPTE Motion Imag. J.*, vol. 131, no. 7, pp. 26–34, 2022.

[142] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "OPTILE: Toward optimal tiling in 360-degree video streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 708–716.

[143] R. Weerakkody. "H.265 HEVC vs H.264 AVC: 50." Feb. 2020. [Online]. Available: https://www.bbc.co.uk/rd/blog/2016-01-h264-h265-avc-advanced-video-coding-hevc-high-efficiency

[144] M. Uhrina, L. Sevcik, J. Bienik, and L. Smatanova, "Performance comparison of VVC, AV1, HEVC, AND AVC for high resolutions," *Electronics*, vol. 13, no. 5, p. 953, 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/5/953

[145] T. Nguyen and D. Marpe, "Compression efficiency analysis of AV1, VVC, and HEVC for random access applications," *APSIPA Trans. Signal Inf. Processing*, vol. 10, p. e11, Jul. 2021.

[146] K. Torres. "H.266 VVC VS AV1, which is better?" Dec. 22, 2023. [Online]. Available: https://www.winxdvd.com/video-transcoder/h266-vvc-vs-av1.html

[147] B. J. Van Rensburg, W. Puech, and J.-P. Pedeboy, "The first DRACO 3D object crypto-compression scheme," *IEEE Access*, vol. 10, pp. 10566–10574, 2022.

[148] K. Christaki et al., "Subjective visual quality assessment of immersive 3D media compressed by open-source static 3D mesh codecs," in *Proc. Multimedia Model. 25th Int. Conf. (MMM)*, 2019, pp. 80–91.

[149] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016.

[150] L. Ilola, L. Kondrad, S. Schwarz, and A. Hamza, "An overview of the MPEG standard for storage and transport of visual volumetric video-based coding," *Front. Signal Process.*, vol. 2, Jun. 2022, Art. no. 883943.

[151] J. M. Boyce et al., "MPEG immersive video coding standard," *Proc. IEEE*, vol. 109, no. 9, pp. 1521–1536, Sep. 2021.

[152] S. Schwarz et al., "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 133–148, Sep. 2021.

[153] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC)," *APSIPA Trans. Signal Inf. Process.*, vol. 9, p. e13, Apr. 2020.

[154] C. Xie, F. Xin, M. Chen, and S. L. Song, "OO-VR: NUMA friendly object-oriented VR rendering framework for future NUMA-based multi-GPU systems," in *Proc. ACM/IEEE 46th Annu. Int. Symp. Comput. Architect. (ISCA)*, 2019, pp. 53–65.

[155] S. Keene, *Google Daydream VR Cookbook*. London, U.K.: Addison-Wesley Prof., 2018.

[156] P. Tranton, *Samsung Gear VR: An Easy Guide for Beginners*. New York, NY, USA: Conceptual Kings, 2016.

[157] K. Boos, D. Chu, and E. Cuervo, "FlashBack: Immersive virtual reality on mobile devices via rendering memoization," in *Proc. 14th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2016, pp. 291–304.

[158] B. Reinert, J. Kopf, T. Ritschel, E. Cuervo, D. Chu, and H.-P. Seidel, "Proxy-guided image-based rendering for mobile devices," in *Computer Graphics Forum*, vol. 35. New York, NY, USA: Wiley, 2016, pp. 353–362.

[159] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, and N. Dai, "FURION: Engineering high-quality immersive virtual reality on today's mobile devices," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 409–421.

[160] A. Mehrabi, M. Siekkinen, and Antti, "Multi-tier cloudVR: Leveraging edge computing in remote rendered virtual reality," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, pp. 1–24, 2021.

[161] Y.-T. Lee, K.-T. Chen, H.-I. Su, and C.-L. Lei, "Are all games equally cloud-gaming-friendly? An electromyographic approach," in *Proc. 11th Annu. Workshop Netw. Syst. Support Games (NetGames)*, 2012, pp. 1–6.

[162] W. Cai et al., "The future of cloud gaming [point of view]," *Proc. IEEE*, vol. 104, no. 4, pp. 687–691, Apr. 2016.

[163] X. Liu, C. Vlachou, F. Qian, C. Wang, and K.-H. Kim, "FireFly: Untethered multi-user VR for commodity mobile devices," in *Proc. USENIX Conf. USENIX Annu. Techn. Conf.*, 2020, pp. 943–657.

[164] S.-A. Wang, "Research on fuzzy image reconstruction method based on real-time fusion technology of VR and AR," in *Proc. Int. Conf. Virtual Real. Intell. Syst. (ICVRIS)*, 2019, pp. 47–50.

[165] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," in *Proc. AIP Conf.*, 2018, p. 1949.

[166] P. Bellavista, C. Giannelli, M. Mamei, M. Mendula, and M. Picone, "Digital twin oriented architecture for secure and QoS aware intelligent communications in industrial environments," *Pervasive Mobile Comput.*, vol. 85, Sep. 2022, Art. no. 101646.

[167] S. Hashima et al., "On softwarization of intelligence in 6G networks for ultra-fast optimal policy selection: Challenges and opportunities," *IEEE Netw.*, vol. 37, no. 2, pp. 190–197, Mar./Apr. 2023.

[168] 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 Commun. Stand. Mag.*, vol. 5, no. 4, pp. 154–160, Dec. 2021.

[169] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "GAUGAN: Semantic image synthesis with spatially adaptive normalization," in *Proc. ACM SIGGRAPH Real-Time Live*, 2019, p. 31.

[170] E. Rothermich, "Mind games: How robots can help regulate brain–computer interfaces," *Univ. Pennsylvania J. Law Public Affairs*, vol. 7, no. 2, p. 4, 2022.

[171] S. Fang, M. Cui, X. Feng, and Y. Lv, "Methods and strategies for improving the novel view synthesis quality of neural radiation field," 2024, *arXiv:2401.12451*.

[172] H. Iqbal, A. Khalid, and M. Shahzad, "Dissecting cloud gaming performance with DeCAF," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 3, pp. 1–27, 2021.

[173] B. Egliston and M. Carter, "Oculus imaginaries: The promises and perils of Facebook's virtual reality," *New Media Soc.*, vol. 24, no. 1, pp. 70–89, 2022.

[174] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *Proc. NSDI*, 2017, pp. 531–544.

[175] M. Khan and J. Chakareski, "Visible light communication for next generation untethered virtual reality systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2019, pp. 1–6.

[176] J. C.-W. Cheung et al., "X-reality for phantom limb management for amputees: A systematic review and meta-analysis," *Eng. Regener.*, vol. 4, no. 2, pp. 134–151, Jun. 2023.

[177] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of Google stadia traffic," *Comput. Commun.*, vol. 188, pp. 99–116, Apr. 2022.

[178] G. Quadrio, A. Bujari, C. E. Palazzi, D. Ronzani, D. Maggiorini, and L. A. Ripamonti, "Network analysis of the steam in-home streaming game system: Poster," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 475–476.

[179] P.-W. Hsiao and C.-H. Su, "A study on the impact of steam education for sustainable development courses and its effects on student motivation and learning," *Sustainability*, vol. 13, no. 7, p. 3772, 2021.

[180] M. Suznjevic, I. Slivar, and L. Skorin-Kapov, "Analysis and QoE evaluation of cloud gaming service adaptation under different network conditions: The case of NVIDIA geforce now," in *Proc. IEEE 8th Int. Conf. Qual. Multimedia Exp. (QoMEX)*, 2016, pp. 1–6.

[181] J. Kim, P. Knowles, J. Spjut, B. Boudaoud, and M. Mcguire, "Post-render warp with late input sampling improves aiming under high latency conditions," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 3, no. 2, pp. 1–18, 2020.

[182] Y.-J. Chen, C.-Y. Hung, and S.-Y. Chien, "Distributed rendering: Interaction delay reduction in remote rendering with client-end GPU-accelerated scene warping technique," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2017, pp. 67–72.

[183] L. Han and K. Smith, "Problem statement: Transport support for augmented and virtual reality applications," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-han-iccrg-arvr-transport-problem-01, Mar. 2017. [Online]. Available: https://datatracker.ietf.org/doc/draft-han-iccrg-arvr-transport-problem/01/

[184] S. K. Bailey, J. E. Lewis, B. A. Ciccone, R. L. Friedland, and C. C. Reiner, *Assessing Usability of Untethered Head-Mounted Displays for Medical Education: A Within-Person Randomized Trial*, LWW, Boulder, Co, USA, 2022.

[185] D. Heaney. "Apple vision pro full specs, features, details." 2023. [Online]. Available: https://www.uploadvr.com/apple-vision-pro-specs-features-details/

[186] Y. Guan, X. Hou, N. Wu, B. Han, and T. Han, "DeepMix: Mobility-aware, lightweight, and hybrid 3D object detection for headsets," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2022, pp. 28–41.

[187] L. Liu et al., "Cutting the cord: Designing a high-quality untethered VR system with low latency remote rendering," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2018, pp. 68–80.

[188] R. Zhong et al., "On building a programmable wireless high-quality virtual reality system using commodity hardware," in *Proc. 8th Asia–Pac. Workshop Syst.*, 2017, pp. 1–7.

[189] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "GamingAnyWhere: The first open source cloud gaming system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 1s, pp. 1–25, 2014.

[190] D. Beeler and A. Gosalia. "Asynchronous time warp on oculus RiFt." 2016. [Online]. Available: https://developer.oculus.com/blog/asynchronous-timewarp-on-oculus-rift/

[191] B. Keshavarz and J. F. Golding, "Motion sickness: Current concepts and management," *Current Opin. Neurol.*, vol. 35, no. 1, pp. 107–112, 2022.

[192] J. Meng, S. Paul, and Y. C. Hu, "COTERIE: Exploiting frame similarity to enable high-quality multiplayer VR on commodity mobile devices," in *Proc. 25th Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2020, pp. 923–937.

[193] A. Di Domenico, G. Perna, M. Trevisan, L. Vassio, and D. Giordano, "A network analysis on cloud gaming: Stadia, geforce now and PSNOW," *Network*, vol. 1, no. 3, pp. 247–260, 2021.

[194] S. Shi, V. Gupta, and R. Jana, "FREEDOM: Fast recovery enhanced VR delivery over mobile networks," in *Proc. 17th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2019, pp. 130–141.

[195] W. Zhang, F. Qian, B. Han, and P. Hui, "DeepVista: 16k panoramic cinema on your mobile device," in *Proc. Web Conf.*, 2021, pp. 2232–2244.

[196] C. Xie, X. Li, Y. Hu, H. Peng, M. Taylor, and S. L. Song, "Q-VR: System-level design for future mobile collaborative virtual reality," in *Proc. 26th ACM Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, 2021, pp. 587–599.

[197] Y. Leng, C.-C. Chen, Q. Sun, J. Huang, and Y. Zhu, "Energy-efficient video processing for virtual reality," in *Proc. 46th Int. Symp. Comput. Archit.*, 2019, pp. 91–103.

[198] J. V. D. Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. D. Turck, "Tile-based adaptive streaming for virtual reality video," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 4, pp. 1–24, 2019.

[199] N. Pfund, N. Sampat, and J. Viggiano, "Relative impact of key rendering parameters on perceived quality of VR imagery captured by the Facebook surround 360 camera," *Frameless*, vol. 1, no. 1, p. 24, 2019.

[200] J. Johnson, "Jumping into the world of virtual and augmented reality," *Knowl. Quest*, vol. 47, no. 4, pp. 22–27, 2019.

[201] J. M. P. Van Waveren, "The asynchronous time warp for virtual reality on consumer hardware," in *Proc. 22nd ACM Conf. Virtual Real. Softw. Technol.*, 2016, pp. 37–46.

[202] D. Evangelakos and M. Mara, "Extended timewarp latency compensation for virtual reality," in *Proc. 20th ACM SIGGRAPH Symp. Interact. 3D Graph. Games*, 2016, pp. 193–194.

[203] D. Kanter, *Graphics Processing Requirements for Enabling Immersive VR*, AMD, Santa Clara, CA, USA, 2015.

[204] M. Kraemer, "Accelerating your VR games with VRworks," in *Proc. NVIDIAs GPU Technol. Conf. (GTC)*, 2018, p. 15.

[205] K. Christodoulou, L. Katelaris, M. Themistocleous, P. Christodoulou, and E. Iosif, "NFTS and the metaverse revolution: Research perspectives and open challenges," in *Blockchains and the Token Economy: Theory and Practice*. Cham, Switzerland: Springer, 2022, pp. 139–178.

[206] A. Alhilal, K. Shatilov, G. Tyson, T. Braud, and P. Hui, "Network traffic in the metaverse: The case of social VR," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, 2023, p. 2.

[207] J. Rosenberg, "Interactive connectivity establishment (ICE): A protocol for network address translator (NAT) traversal for offer/answer protocols," IETF, RFC 8445, 2010.

[208] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session traversal utilities for NAT (STUN)," IETF, RFC 5389, 2008.

[209] Mozilla. "Mozilla hubs system overview." 2022. [Online]. Available: https://hubs.mozilla.com/docs/system-overview.html

[210] C. R. Reis and R. P. de Mattos Fortes, "An overview of the software engineering process and tools in the Mozilla project," in *Proc. Open Source Softw. Develop. Workshop*, vol. 8, 2002, p. 21.

[211] M. Tassinari, M. B. Aulbach, and I. Jasinskaja-Lahti, "Investigating the influence of intergroup contact in virtual reality on empathy: An exploratory study using altspaceVR," *Front. Psychol.*, vol. 12, Feb. 2022, Art. no. 815497.

[212] C. Holmberg, S. Hakansson, and G. Eriksson, "Web real-time communication use cases and requirements," IETF, RFC 7478, 2015.

[213] A. O. Bicen and O. B. Akan, "Reliability and congestion control in cognitive radio sensor networks," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1154–1164, 2011.

[214] F. Tang, X. Chen, M. Zhao, and N. Kato, "The roadmap of communication and networking in 6G for the metaverse," *IEEE Wireless Commun.*, vol. 30, no. 4, pp. 72–81, Aug. 2023.

[215] K. Raaen, *Response Time in Games: Requirements and Improvements*. Univ. Oslo, Oslo, Norway, 2016.

[216] V. Petrov, M. Gapeyenko, S. Paris, A. Marcano, and K. I. Pedersen, "Standardization of extended reality (XR) over 5G and 5G-advanced 3GPP new radio," 2022, *arXiv:2203.02242*.

[217] S. Jha and M. Fry, "Continuous media playback and jitter control," in *Proc. 3rd IEEE Int. Conf. Multimedia Comput. Syst.*, 1996, pp. 245–252.

[218] A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi, and G. Fichtinger, "PLUS: Open-source toolkit for ultrasound-guided intervention systems," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 10, pp. 2527–2537, Oct. 2014.

[219] R. Krishna and A. Rahman, "Media operations use case for an extended reality application on edge computing infrastructure," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-mops-ar-use-case-12, Jul. 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-mops-ar-use-case/12/

[220] B. Juurlink et al., *Understanding the Application: An Overview of the H.264 Standard*. New York, NY, USA: Springer, 2012.

[221] G. Van der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 698–718, Sep. 2008.

[222] S. Zeadally, H. Moustafa, and F. Siddiqui, "Internet protocol television (IPTV): Architecture, trends, and challenges," *IEEE Syst. J.*, vol. 5, no. 4, pp. 518–527, Dec. 2011.

[223] D. R. Oran, "Considerations in the development of a QoS architecture for CCNx-like information-centric networking protocols," IETF, RFC 9064, Jun. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9064

[224] R. T. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP)—Version 1 functional specification," IETF, RFC 2205, Sep. 1997. [Online]. Available: https://www.rfc-editor.org/info/rfc2205

[225] F. Baker, J. Babiarz, and K. H. Chan, "Configuration guidelines for DiffServ service classes," IETF, RFC 4594, Aug. 2006. [Online]. Available: https://www.rfc-editor.org/info/rfc4594

[226] Y. Li, R. Miao, M. Alizadeh, and M. Yu, "{DETER}: Deterministic {TCP} replay for performance diagnosis," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 437–452.

[227] R. Kumar, A. Koutsaftis, F. Fund, G. Naik, P. Liu, Y. Liu, and S. Panwar, "TCP BBR for ultra-low latency networking: Challenges, analysis, and solutions," in *Proc. IFIP Netw. Conf. (IFIP Networking)*, 2019, pp. 1–9.

[228] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[229] L. U. Khan, Z. Han, D. Niyato, M. Guizani, and C. S. Hong, "Metaverse for wireless systems: Vision, enablers, architecture, and future directions," *IEEE Wireless Commun.*, early access, Apr. 30, 2024, doi: 10.1109/MWC.013.2300287.

[230] H.-M. Kim and J. Grix, "Implementing a sustainability legacy strategy: A case study of pyeongchang 2018 Winter olympic games," *Sustainability*, vol. 13, no. 9, p. 5141, 2021.

[231] Z. Liu et al., "User-centric service system for network beyond IMT-2020 (5G)," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, 2022, pp. 237–240.

[232] R. Ford, M. Zhang, M. Mezzavilla, S. Dutta, S. Rangan, and M. Zorzi, "Achieving ultra-low latency in 5G millimeter wave cellular networks," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 196–203, Mar. 2017.

[233] S. Nsaibi, "Timing performance analysis of the deterministic Ethernet enhancements time-sensitive networking (TSN) for use in the industrial communication," Ph.D. dissertation, Technische Universität Kaiserslautern, Kaiserslautern, Germany, 2020.

[234] T. H. Szymanski, "The 'cyber security via determinism' paradigm for a quantum safe zero trust deterministic Internet of Things (IoT)," *IEEE Access*, vol. 10, pp. 45893–45930, 2022.

[235] S. Jun, Y. Kang, J. Kim, and C. Kim, "Ultra-low-latency services in 5G systems: A perspective from 3GPP standards," *ETRI J.*, vol. 42, no. 5, pp. 721–733, 2020.

[236] R. B. Abreu, G. Pocovi, T. H. Jacobsen, M. Centenaro, K. I. Pedersen, and T. E. Kolding, "Scheduling enhancements and performance evaluation of downlink 5G time-sensitive communications," *IEEE Access*, vol. 8, pp. 128106–128115, 2020.

[237] X. Geng, Y. Ryoo, D. Fedyk, R. Rahman, and Z. Li, "Deterministic networking (DetNet) YANG model," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-detnet-yang-20, Feb. 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-detnet-yang/20/

[238] B. Varga, J. Farkas, and A. G. Malis, "Deterministic networking (DetNet): DetNet PREOF via MPLS over UDP/IP," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-detnet-mpls-over-ip-preof-11, Feb. 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-detnet-mpls-over-ip-preof/11/

[239] T. R. Konda, "Collaborative networking towards application-aware networking," in *Design Innovation and Network Architecture for the Future Internet*. London, U.K.: IGI Global, 2021, pp. 43–65.

[240] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, 2008.

[241] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[242] D. A. Alwahab and S. Laki, "ECN-marking with codel and its compatibility with different TCP congestion control algorithms," in *Proc. Int. Conf. Adv. Sci. Eng. (ICOASE)*, 2020, pp. 1–6.

[243] D. Brunello, I. Johansson, M. Ozger, and C. Cavdar, "Low latency low loss scalable throughput in 5G networks," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, 2021, pp. 1–7.

[244] K. De Schepper, "The explicit congestion notification (ECN) protocol for low latency, low loss, and scalable throughput (l4s)," IETF, RFC 9331, 2023.

[245] U. Majeed, A. N. Malik, N. Abbas, and W. Abbass, "An energy-efficient distributed congestion control protocol for wireless multimedia sensor networks," *Electronics*, vol. 11, no. 20, p. 3265, 2022.

[246] H. Ahmed and M. J. Arshad, "Buffer occupancy-based transport to reduce flow completion time of short flows in data center networks," *Symmetry*, vol. 11, no. 5, p. 646, 2019.

[247] V. Srivastava, S. Tripathi, K. Singh, and L. H. Son, "Energy efficient optimized rate based congestion control routing in wireless sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 1325–1338, Sep. 2020.

[248] S. Javaid, H. Fahim, Z. Hamid, and F. B. Hussain, "Traffic-aware congestion control (TACC) for wireless multimedia sensor networks," *Multimedia Tools Appl.*, vol. 77, no. 20, pp. 4433–4452, 2018.

[249] R. Kumar, U. Venkanna, and V. Tiwari, "OPT-ACM: An optimized load balancing based admission control mechanism for software defined hybrid wireless based IoT (SDHW-IoT) network," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107888.

[250] W.-x. Liu, J. Cai, Q. C. Chen, and Y. Wang, "DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102865.

[251] J. Zhou, Z. Lin, and X. Jiang, "Secure load-balanced scheme for cluster-based WSNs," in *Proc. 6th Int. Conf. Syst. Informat. (ICSAI)*, 2019, pp. 804–809.

[252] A. A. Rezaee, M. H. Yaghmaee, A. M. Rahmani, and A. H. Mohajerzadeh, "HOCA: Healthcare aware optimized congestion avoidance and control protocol for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 37, pp. 216–228, Jan. 2014.

[253] S. L. Yadav, R. Ujjwal, S. Kumar, O. Kaiwartya, M. Kumar, and P. K. Kashyap, "Traffic and energy aware optimization for congestion control in next generation wireless sensor networks," *J. Sens.*, vol. 2021, no. 1, Jun. 2021, Art. no. 5575802.

[254] P. Chen, B. Chen, M. Wang, S. Wang, and Z. Li, "Visual data compression for metaverse: Technology, standard, and challenges," in *Proc. IEEE Int. Conf. Metaverse Comput. Netw. Appl. (MetaCom)*, 2023, pp. 360–364.

[255] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Albuquerque, "A comprehensive survey of multi-view video summarization," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107567.

[256] M. Berger et al., "A survey of surface reconstruction from point clouds," in *Computer Graphics Forum*, vol. 36. Hoboken, NJ, USA: Wiley, 2017, pp. 301–329.

[257] L. Li and Z. Li, "Light field and plenoptic point cloud compression," in *Handbook of Dynamic Data Driven Applications Systems: Volume 1*. Cham, Switzerland: Springer Int. Publ., 2021, pp. 557–583.

[258] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, "Video coding optimization for virtual reality 360-degree source," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 118–129, Jan. 2020.

[259] L. Li, Z. Li, S. Liu, and H. Li, "Efficient projected frame padding for video-based point cloud compression," *IEEE Trans. Multimedia*, vol. 23, pp. 2806–2819, 2020.

[260] A. Akhtar, W. Gao, L. Li, Z. Li, W. Jia, and S. Liu, "Video-based point cloud compression artifact removal," *IEEE Trans. Multimedia*, vol. 24, pp. 2866–2876, 2021.

[261] K. Ainala, R. N. Mekuria, B. Khathariya, Z. Li, Y.-K. Wang, and R. Joshi, "An improved enhancement layer for Octree based point cloud compression with plane projection approximation," in *Proc. 39th SPIE Appl. Digit. Image Process.*, 2016, pp. 223–231.

[262] M. U. Lokumarambage, V. S. S. Gowrisetty, H. Rezaei, T. Sivalingam, N. Rajatheva, and A. Fernando, "Wireless end-to-end image transmission system using semantic communications," *IEEE Access*, vol. 11, pp. 37149–37163, 2023.

[263] S. Ghannay et al., "End-to-end named entity and semantic concept extraction from speech," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, 2018, pp. 692–699.

[264] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic transmission via revising modules in conventional communications," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 28–34, Jun. 2023.

[265] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 2326–2330.

[266] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, "Neural joint source-channel coding," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1182–1192.

[267] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 567–579, May 2019.

[268] H. Xie and Z. Qin, "A lite distributed semantic communication system for Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[269] H. Tong et al., "Federated learning for audio semantic communication," *Front. Commun. Netw.*, vol. 2, Dec. 2021, Art. no. 734402.

[270] B. Chen, Z. Yan, and K. Nahrstedt, "Context-aware image compression optimization for visual analytics offloading," in *Proc. 13th ACM Multimedia Syst. Conf.*, 2022, pp. 27–38.

[271] S. Yao et al., "Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency," in *Proc. 18th Conf. Embedded Netw. Sensor Syst.*, 2020, pp. 476–488.

[272] B. Haynes, A. Minyaylov, M. Balazinska, L. Ceze, and A. Cheung, "Visualcloud demonstration: A DBMS for virtual reality," in *Proc. ACM Int. Conf. Manag. Data*, 2017, pp. 1615–1618.

[273] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant TILE-based streaming of panoramic video for virtual reality applications," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 601–605.

[274] K. Q. M. Ngo, R. Guntur, and W. T. Ooi, "Adaptive encoding of zoomable video streams based on user access pattern," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 211–222.

[275] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Fixation prediction for 360 video streaming in head-mounted virtual reality," in *Proc. 27th Workshop Netw. Oper. Syst. Support Digital Audio Video*, 2017, pp. 67–72.

[276] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, "360 innovations for panoramic video streaming," in *Proc. 16th ACM Workshop Hot Topics Netw.*, 2017, pp. 50–56.

[277] Y. S. de la Fuente, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, "Delay impact on MPEG OMAF's tile-based viewport-dependent 360 video streaming," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 18–28, Mar. 2019.

[278] X. Jiang, Y.-H. Chiang, Y. Zhao, and Y. Ji, "PLATO: Learning-based adaptive streaming of 360-degree videos," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, 2018, pp. 393–400.

[279] Y. Hu, Y. Liu, and Y. Wang, "VAS360: QoE-driven viewport adaptive streaming for 360 video," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2019, pp. 324–329.

[280] H. Wang, Z. Wang, D. Chen, Q. Liu, H. Ke, and K. K. Han, "MetaMobility: Connecting future mobility with the metaverse," *IEEE Veh. Technol. Mag.*, vol. 18, no. 3, pp. 69–79, Sep. 2023.

[281] F. Lu, H. Zhou, L. Guo, J. Chen, and L. Pei, "An arcore-based augmented reality campus navigation system," *Appl. Sci.*, vol. 11, no. 16, p. 7515, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:238696485

[282] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future networking challenges: The case of mobile augmented reality," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 1796–1807.

[283] G. Castignani, A. Lampropulos, A. Blanc, and N. Montavont, "WI2ME: A mobile sensing platform for wireless heterogeneous networks," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, 2012, pp. 108–113.

[284] H. Du, D. Niyato, C. Miao, J. Kang, and D. I. Kim, "Optimal targeted advertising strategy for secure wireless edge metaverse," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 4346–4351.

[285] M. Xu et al., "Quantum-secured space–air–ground integrated networks: Concept, framework, and case study," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 136–143, Dec. 2023.

[286] X. You et al., "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, pp. 1–74, Nov. 2021.

[287] P. Bhattacharya et al., "Towards future Internet: The metaverse perspective for diverse industrial applications," *Mathematics*, vol. 11, no. 4, p. 941, 2023.

[288] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, "A comprehensive survey on mobility management in 5G heterogeneous networks: Architectures, challenges and solutions," *IEEE Access*, vol. 8, pp. 195883–195913, 2020.

[289] R. Borralho, A. Mohamed, A. U. Quddus, P. Vieira, and R. Tafazolli, "A survey on coverage enhancement in cellular networks: Challenges and solutions for future deployments," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1302–1341, 2nd Quart., 2021.

[290] J. Collins, "GPRS tunneling protocol (GTP)," in *Encyclopedia of Cryptography, Security and Privacy*. Berlin, Germany: Springer, 2022, pp. 1–3.

[291] G. Nigam, P. Minero, and M. Haenggi, "Coordinated multipoint joint transmission in heterogeneous networks," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 4134–4146, Nov. 2014.

[292] A. Shahmansoori, B. Uguen, G. Destino, G. Seco-Granados, and H. Wymeersch, "Tracking position and orientation through millimeter wave lens MIMO in 5G systems," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1222–1226, Aug. 2019.

[293] C. Fiandrino, H. Assasa, P. Casari, and J. Widmer, "Scaling millimeter-wave networks to dense deployments and dynamic environments," *Proc. IEEE*, vol. 107, no. 4, pp. 732–745, Apr. 2019.

[294] H. Zhang, W. Huang, and Y. Liu, "Handover probability analysis of anchor-based multi-connectivity in 5G user-centric network," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 396–399, Apr. 2019.

[295] J.-H. Choi and D.-J. Shin, "Generalized RACH-less handover for seamless mobility in 5G and beyond mobile networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1264–1267, Apr. 2019.

[296] A. L. Yusof, N. Ya'acob, and M. T. Ali, "Handover initiation across heterogeneous access networks for next generation cellular network," in *Proc. IEEE Symp. Wireless Technol. Appl. (ISWTA)*, 2011, pp. 78–83.

[297] S.-J. Yoo, D. Cypher, and N. Golmie, 'Predictive link trigger mechanism for seamless handovers in heterogeneous wireless networks," *Wireless Commun. Mobile Comput.*, vol. 9, no. 5, pp. 685–703, 2009.

[298] H.-S. Kim, S.-H. Park, C.-S. Park, J. W. Kim, and S.-J. Ko, "Selective channel scanning for fast handoff in wireless LAN using neighbor graph," in *Proc. ITC-CSCC*, vol. 4, 2004, pp. 194–203.

[299] S. Shin, A. G. Forte, A. S. Rawat, and H. Schulzrinne, "Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs," in *Proc. 2nd Int. Workshop Mobility Manag. Wireless Access Protocols*, 2004, pp. 19–26.

[300] S. Liang, Y. Zhang, B. Fan, and H. Tian, "Multi-attribute vertical handover decision-making algorithm in a hybrid VLC-FEMTO system," *IEEE Wireless Commun. Letters*, vol. 21, no. 7, pp. 1521–1524, Jul. 2017.

[301] X. Bao, W. Adjardjah, A. A. Okine, W. Zhang, and J. Dai, "A QoE-maximization-based vertical handover scheme for VLC heterogeneous networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–12, 2018.

[302] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-Thru Internet environments," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 3, 2005, pp. 1849–1862.

[303] G. Sinha, M. R. Kanagarathinam, S. R. Jayaseelan, and G. K. Choudhary, "CQUIC: Cross-layer Quic for next generation mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–8.

[304] N. Cheng et al., "6G service-oriented space–air–ground integrated network: A survey," *Chin. J. Aeronaut.*, vol. 35, no. 9, pp. 1–18, 2022.

[305] P. K. R. Maddikunta et al., "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17608–17619, Aug. 2021.

[306] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP extensions for multipath operation with multiple addresses," IETF, RFC 8684, 2020.

[307] S. Cheshire, D. Schinazi, and C. Paasch. "Advances in networking, part 1." Jul. 2017. [Online]. Available: https://developer.apple.com/videos/play/wwdc2019/712

[308] M. Scharf and S. Kiesel, "NXG03-5: Head-of-line blocking in TCP and SCTP: Analysis and measurements," in *Proc. IEEE Globecom*, 2006, pp. 1–5.

[309] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[310] J. Iyengar and M. Thomson, "QUIC: A UDP-based multiplexed and secure transport," IETF, RFC 9000, 2021.

[311] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.

[312] Q. De Coninck and O. Bonaventure, "Multiflow QUIC: A generic multipath transport protocol," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 108–113, Mar. 2021.

[313] H. Wu, Ö. Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2295–2310, Oct. 2020.

[314] W. Yang, S. Shu, L. Cai, and J. Pan, "MM-QUIC: Mobility-aware multipath QUIC for satellite networks," in *Proc. IEEE 17th Int. Conf. Mobility Sens. Netw. (MSN)*, 2021, pp. 608–615.

[315] S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "Information-centric networking (ICN) research challenges," IETF, RFC 7927, 2016.

[316] Y. Zhang, Z. Xia, S. Mastorakis, and L. Zhang, "KITE: Producer mobility support in named data networking," in *Proc. 5th ACM Conf. Inf. Centric Netw.*, 2018, pp. 125–136.

[317] J. Auge, G. Carofiglio, L. Muscariello, and M. Papalini, "Anchorless mobility through HICN," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-auge-dmm-hicn-mobility-04, Jul. 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-auge-dmm-hicn-mobility/04/

[318] M. Rymaszewski, *Second Life: The Official Guide*. Hoboken, NJ, USA: Wiley, 2007.

[319] T. Wang, Z. Su, Y. Xia, J. Muppala, and M. Hamdi, "Designing efficient high performance server-centric data center network architecture," *Comput. Netw.*, vol. 79, pp. 283–296, Mar. 2015.

[320] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, May 2021, Art. no. 107930.

[321] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, Nov. 2020.

[322] K. Long, Y. Cui, C. Ye, and Z. Liu, "Optimal wireless streaming of multi-quality 360 VR video by exploiting natural, relative smoothness-enabled, and transcoding-enabled multicast opportunities," *IEEE Trans. Multimedia*, vol. 23, pp. 3670–3683, 2020.

[323] C. Guo, Y. Cui, and Z. Liu, "Optimal multicast of tiled 360 VR video," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 145–148, Feb. 2019.

[324] F. Tan, P. Wu, Y.-C. Wu, and M. Xia, "Energy-efficient non-orthogonal multicast and unicast transmission of cell-free massive MIMO systems with SWIPT," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 949–968, Apr. 2021.

[325] Z. Zhang, Z. Ma, Y. Xiao, M. Xiao, G. K. Karagiannidis, and P. Fan, "Non-orthogonal multiple access for cooperative multicast millimeter wave wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 8, pp. 1794–1808, Mar. 2017.

[326] P. Xiang, H. Shan, Z. Zhang, L. Yu, and T. Q. Quek, "NOMA based VR video transmissions exploiting user behavioral coherence," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–6.

[327] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 375–390, Jan. 2019.

[328] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Nonorthogonal multiple access for 5G and beyond," *Proc. IEEE*, vol. 105, no. 12, pp. 2347–2381, Dec. 2017.

[329] Y. Qiu, H. Zhang, K. Long, and M. Guizani, "Subchannel assignment and power allocation for time-varying fog radio access network with NOMA," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3685–3697, Jun. 2021.

[330] R. Brandenburg, R. van Koenen, and D. Sztykman, *CDN Optimization for VR Streaming*. Amsterdam, The Netherlands: Int. Broadcast. Convent., 2017.

[331] A. Mahzari, A. T. Nasrabadi, A. Samiei, and R. Prakash, "FoV-aware edge caching for adaptive 360 video streaming," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 173–181.

[332] P. Maniotis and N. Thomos, "Viewport-aware deep reinforcement learning approach for 360 video caching," *IEEE Trans. Multimedia*, vol. 24, pp. 386–399, 2021.

[333] F. Y. Okay and S. Ozdemir, "Routing in fog-enabled IoT platforms: A survey and an SDN-based solution," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4871–4889, Dec. 2018.

[334] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," *Intell. Converg. Netw.*, vol. 1, no. 2, pp. 181–198, 2020.

[335] T. Meng, N. R. Schiff, P. B. Godfrey, and M. Schapira, "PCC proteus: Scavenger transport and beyond," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl. Technol. Architect. Protocols Comput. Commun.*, 2020, pp. 615–631.

[336] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low extra delay background transport (LEDBAT)," IETF, RFC 6817, 2012.

[337] M. Dong et al., "PCC VIVACE: Online-learning congestion control," in *Proc. 15th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2018, pp. 343–356.

[338] N. Rozen-Schiff, A. Navon, L. Bruckman, and I. Pechtalt, "PRISM based transport: How networks can boost QoS for advanced video services?" in *Proc. Workshop Design Deployment Eval. Netw. Assist. Video Stream.*, 2021, pp. 1–7.

[339] U. Paro, F. Chiariotti, A. A. Deshpande, M. Polese, A. Zanella, and M. Zorzi, "Extending the NS-3 QUIC module," in *Proc. 23rd Int. ACM Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, 2020, pp. 19–26.

[340] R. Marx, J. Herbots, W. Lamotte, and P. Quax, "Same standards, different decisions: A study of QUIC and HTTP/3 implementation diversity," in *Proc. Workshop Evol. Perform. Interoperability QUIC*, 2020, pp. 14–20.

[341] P. Wang, Z. Jiang, M. Qi, L. Dai, and H. Xu, "Uncertainty-aware weighted fair queueing for routers based on deep reinforcement learning," in *Proc. IEEE 4th Int. Conf. Electron. Commun. Eng. (ICECE)*, 2021, pp. 1–7.

[342] J. Holland, A. Begen, and S. Dawkins, "Operational considerations for streaming media," IETF, RFC 9317, 2022.

[343] L. Curley, K. Pugin, S. Nandakumar, and V. Vasiliev, "Warp—Live media transport over QUIC," Internet Engineering Task Force, Fremont, CA, USA, Internet-Draft draft-lcurley-warp-04, Mar. 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-lcurley-warp/04/

[344] J. Cenzano. "Facebook experimental/webcodecs-capture-play: Live streaming low latency experimentation platform in the browser (using Webcodecs)." Apr. 2023. [Online]. Available: https://github.com/facebookexperimental/webcodecs-capture-play

[345] Z. Gurel, T. E. Civelek, A. Bodur, S. Bilgin, D. Yeniceri, and A. C. Begen, "Media over QUIC: Initial testing, findings and results," in *Proc. 14th Conf. ACM Multimedia Syst.*, 2023, pp. 301–306.

[346] O. Hiba, H. Leibowitz, and A. Herzberg, "QUICR: QUIC resiliency to BW-DOS attacks," in *Proc. QUIC Privacy Security* 2020, p. 9.

[347] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, S. Hammoudi, and R. Zimmermann, "Toward one-second latency: Evolution of live media streaming," 2023, *arXiv:2310.03256*.

[348] J. Gruessing and S. Dawkins, "Media over QUIC—Use cases and requirements for media transport protocol design," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-gruessing-moq-requirements-05, May 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-gruessing-moq-requirements/05/

[349] A. Azgin, R. Ravindran, and G. Wang, "Scalable multicast for content delivery in information centric networks," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2018, pp. 105–111.

[350] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis," in *Proc. Image Anal. 21st Scand. Conf. (SCIA)*, 2019, pp. 28–40.

[351] O. B. Raheem, B. O. Omijeh, and C. Ibeachu, "Virtual reality: A new approach for learning anatomy," *African J. Humanities Contemp. Educ. Res.*, vol. 10, no. 1, pp. 186–193, 2023.

[352] T. Abdelzaher, M. Caesar, C. Mendis, K. Nahrstedt, M. Srivastava, and M. Yu, "Challenges in metaverse research: An Internet of Things perspective," in *Proc. IEEE Int. Conf. Metaverse Comput. Netw. Appl. (MetaCom)*, 2023, pp. 161–170.

[353] Animats. "How many users can be on a metaverse?" 2022. [Online]. Available: https://www.reddit.com/r/metaverse/comments/u3wp79/how_many_users_can_be_on_a_metaverse/

[354] B. Wire. "Virtway exceeds the record for concurrent users per scene in its B2B metaverse by allowing 1000 people to connect in the same room, even from mobile devices." 2023. [Online]. Available: https://www.businesswire.com/news/home/20230508005497/en

[355] A. R. Bharambe, J. Pang, and S. Seshan, "Colyseus: A distributed architecture for online multiplayer games," in *Proc. NSDI*, vol. 6, 2006, pp. 12–12.

[356] J. Müller and S. Gorlatch, "ROKKATAN: Scaling an RTS game design to the massively multiplayer realm," *Comput. Entertainment*, vol. 4, no. 3, p. 11, 2006.

[357] J. G. Salinas, F. B. Seguí, A. S. Piera, and F. J. P. Castillo, "Key technologies for networked virtual environments," *Multimedia Tools Appl.*, vol. 82, pp. 1–67, Apr. 2023.

[358] D. C. Marinescu, *Cloud Computing: Theory and Practice*. London, U.K.: Morgan Kaufmann, 2022.

[359] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Polynomial codes: An optimal design for high-dimensional coded matrix multiplication," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4403–4413.

[360] Y. Jiang, J. Kang, D. Niyato, X. Ge, Z. Xiong, and C. Miao, "Reliable coded distributed computing for metaverse services: Coalition formation and incentive mechanism design," 2021, *arXiv:2111.10548*.

[361] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 278–301, Jan. 2020.

[362] J. Baek and G. Kaddoum, "Heterogeneous task offloading and resource allocations via deep recurrent reinforcement learning in partial observable multifog networks," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1041–1056, Jan. 2021.

[363] J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 242–253, Jan. 2021.

[364] X.-Q. Pham, T. Huynh-The, E.-N. Huh, and D.-S. Kim, "Partial computation offloading in parked vehicle-assisted multi-access edge computing: A game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 10220–10225, Sep. 2022.

[365] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, May 2022.

[366] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.

[367] D. Dai, "What is the difference between VR vs AR vs MR vs XR?." May 2024. [Online]. Available: https://pimax.com/blogs/blogs/what-is-the-difference-between-vr-vs-ar-vs-mr-vs-xr

[368] A. Erfanian, F. Tashtarian, A. Zabrovskiy, C. Timmerer, and H. Hellwagner, "OSCAR: On optimizing resource utilization in live video streaming," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 552–569, Mar. 2021.

[369] C. Ge, N. Wang, W. K. Chai, and H. Hellwagner, "QoE-assured 4K HTTP live streaming via transient segment holding at mobile edge," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1816–1830, Aug. 2018.

[370] A. Erfanian, F. Tashtarian, R. Farahani, C. Timmerer, and H. Hellwagner, "On optimizing resource utilization in AVC-based real-time video streaming," in *Proc. 6th IEEE Conf. Netw. Softw. (NetSoft)*, 2020, pp. 301–309.

[371] A. Erfanian, H. Amirpour, F. Tashtarian, C. Timmerer, and H. Hellwagner, "LWTE-LIVE: Light-weight transcoding at the edge for live streaming," in *Proc. Workshop Design Deployment Eval. Netw. Assisted Video Streaming*, 2021, pp. 22–28.

[372] P. Burak, "Sony and its most profitable division—Playstation," in *Proc. Jpn Manag. Market Entry Crisis Corporate Growth*, 2021, pp. 99–111.

[373] Y. Zang, H. Huang, and C.-F. Li, "Artistic preprocessing for painterly rendering and image stylization," *Vis. Comput.*, vol. 30, pp. 969–979, Nov. 2014.

[374] W. C. T. Parisi. "MAAS whitepaper release." Apr. 2023. https://uploads-ssl.webflow.com/63fe332d7b9ae4159d741e55/64499d8f08bd5bdd1fe6bce1_MaaS_Whitepaper_v1.0.pdf

[375] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[376] J. Son et al., "Split rendering for mixed reality: Interactive volumetric video in action," in *Proc. SIGGRAPH Asia XR*, 2020, pp. 1–3.

[377] D. Kutscher, T. Karkkainen, and J. Ott. "Directions for computing in the network." Jul. 2020. [Online]. Available: https://datatracker.ietf.org/doc/draft-kutscher-coinrg-dir/02/

[378] Y. Jiang et al., "Reliable distributed computing for metaverse: A hierarchical game-theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1084–1100, Jan. 2023.

[379] A. Bujari, A. Calvio, A. Garbugli, and P. Bellavista, "A layered architecture enabling metaverse applications in smart manufacturing environments," in *Proc. IEEE Int. Conf. Metaverse Comput. Netw. Appl. (MetaCom)*, 2023, pp. 585–592.

[380] K. D. Setiawan et al., "The essential factor of metaverse for business based on 7 layers of metaverse–systematic literature review," in *Proc. Int. Conf. Inf. Manag. Technol. (ICIMTech)*, 2022, pp. 687–692.

[381] Q. Qu et al., "The microverse: A task-oriented edge-scale metaverse," *Future Internet*, vol. 16, no. 2, p. 60, 2024.

[382] A. Noferi, G. Nardini, G. Stea, and A. Virdis, "Rapid prototyping and performance evaluation of ETSI MEC-based applications," *Simulat. Model. Practice Theory*, vol. 123, Feb. 2023, Art. no. 102700.

[383] P. Ravindra, A. Khochare, S. P. Reddy, S. Sharma, P. Varshney, and Y. Simmhan, "An adaptive orchestration platform for hybrid dataflows across cloud and edge," in *Proc. Int. Conf. Service Orient. Comput.*, 2017, pp. 395–410.

[384] Y. Wu, "Cloud-edge orchestration for the Internet of Things: Architecture and AI-powered data processing," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12792–12805, Aug. 2021.

[385] A. Rohloff et al., "OpenUVR: An open-source system framework for untethered virtual reality applications," in *Proc. IEEE 27th Real-Time Embedded Technol. Appl. Symp. (RTAS)*, 2021, pp. 223–236.

[386] Z. Long, H. Dong, and A. El Saddik, "Human-centric resource allocation for the metaverse with multi-access edge computing," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 19993–20005, Nov. 2023.

[387] S. Huang et al., "Intelligent eco networking (IEN) III: A shared in-network computing infrastructure towards future Internet," in *Proc. IEEE 3rd Int. Conf. Hot Inf. Centric Netw. (HotICN)*, 2020, pp. 47–52.

[388] A. Sapio, I. Abdelaziz, A. Aldilaijan, M. Canini, and P. Kalnis, "In-network computing is a dumb idea who's time has come," in *Proc. Hot-Nets*, 2017, pp. 150–156.

[389] R. A. Cooke and S. A. Fahmy, "A model for distributed in-network and near-edge computing with heterogeneous hardware," *Future Gener. Comput. Syst.*, vol. 105, pp. 395–409, Apr. 2020.

[390] G. Lia, M. Amadeo, G. Ruggeri, C. Campolo, A. Molinaro, and V. Loscrì, "In-network placement of delay-constrained computing tasks in a softwarized intelligent edge," *Comput. Netw.*, vol. 219, Dec. 2022, Art. no. 109432.

[391] S. M. Rashid, I. Aliyu, I.-K. Jeong, T.-W. Um, and J. Kim, "Graph neural network for in-network placement of real-time metaverse tasks in next-generation network," 2024, *arXiv:2403.01780*.

[392] Y. Cai, J. Llorca, A. M. Tulino, and A. F. Molisch, "Joint compute-caching-communication control for online data-intensive service delivery," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4617–4633, May 2024.

[393] S. Liang, H. Wan, T. Qin, J. Li, and W. Chen, "Multi-user computation offloading for mobile edge computing: A deep reinforcement learning and game theory approach," in *Proc. IEEE 20th Int. Conf. Commun. Technol. (ICCT)*, 2020, pp. 1534–1539.

[394] F. Alriksson et al., "XR and 5G: Extended reality at scale with time-critical communication," *Ericsson Technol. Rev.*, vol. 2021, no. 8, pp. 2–13, 2021.

[395] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 821–834, May 2012.

[396] D. Chatzopoulos, K. Sucipto, S. Kosta, and P. Hui, "Video compression in the neighborhood: An opportunistic approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–6.

[397] K. Sucipto, D. Chatzopoulos, S. Kosta, and P. Hui, "Keep your nice friends close, but your rich friends closer—computation offloading using NFC," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2017, pp. 1–9.

[398] N. Zhao, X. Liu, Y. Chen, S. Zhang, Z. Li, B. Chen, and M.-S. Alouini, "Caching D2D connections in small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12326–12338, Dec. 2018.

[399] J. Donkervliet, J. Ron, J. Li, T. Iancu, C. L. Abad, and A. Iosup, "SERVO: Increasing the scalability of modifiable virtual environments using serverless computing," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2023, pp. 829–840.

[400] N. Pereira, A. Rowe, M. W. Farb, I. Liang, E. Lu, and E. Riebling, "ARENA: The augmented reality edge networking architecture," in *Proc. IEEE Int. Symp. Mixed Augmented Real. (ISMAR)*, 2021, pp. 479–488.

[401] M. Beck and T. Moore, "How we ruined the Internet," 2023, *arXiv.2306.2023*.

[402] G. Carofiglio, G. Grassi, L. Muscariello, M. Papalini, and J. Samain, "ROBUST: A reliable and flexible media transport for real-time services," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 2475–2488, Sep. 2023.

[403] M.-J. Montpetit. "In network computing enablers for extended reality." Jul. 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-montpetit-coin-xr/03/

[404] Z. Yang, M. Chen, G. Li, Y. Yang, and Z. Zhang, "Secure semantic communications: Fundamentals and challenges," 2023, *arXiv:2301.01421*.

[405] Eluvio. "Content fabric technology." 2023. [Online]. Available: https://live.eluv.io/content-fabric/technology

[406] S. Weber et al., "Frameworks enabling ubiquitous mixed reality applications across dynamically adaptable device configurations," *Front. Virtual Real.*, vol. 3, Apr. 2022, Art. no. 765959.

[407] Z. Gurel, T. E. Civelek, and A. C. Begen. "Need for low latency: Media over QUIC." 2023. [Online]. Available: https://doi.org/10.1145/3588444.3591033

[408] S. Neri, "Distributed architecture for resource reservation protocol traffic engineering (RSVP-TE)," Ph.D. dissertation, Dept. Comput. Sci., Concordia Univ., Montreal, QC, Canada, 2007.

[409] W. Geng, Y. Zhang D. Kutscher, A. Kumar, S. Tarkoma, and P. Hui, "SoK: Distributed computing in ICN," in *Proc. 10th ACM Conf. Inf. Centric Netw.*, 2023, pp. 88–100.

[410] B. Quinn and D. K. C. Almeroth. "IP multicast applications: Challenges and solutions," IETF, RFC 3170, Sep. 2001. [Online]. Available: https://www.rfc-editor.org/info/rfc3170

[411] C.-J. Tseng and C.-H. Chen, "The performance of QoS-aware IP multicast routing protocols," *Netw. Int. J.*, vol. 42, no. 2, pp. 97–108, 2003.

[412] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols," *IEEE Netw.*, vol. 14, no. 1, pp. 22–36, Jan./Feb. 2000.

[413] D. Trossen and J. Crowcroft, "If multicast is the answer—What was the question?" 2022, *arXiv:2211.09029*.

[414] J. Loveless, R. Blair, and A. Durai, *IP Multicast: Advanced Multicast Concepts and Large-Scale Multicast Design, Volume 2*. New York, NY, USA: Cisco Press, 2018.

[415] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, "Digital rights management for content distribution," in *Proc. Conf. Res. Practice Inf. Technol.*, vol. 34, 2003, pp. 49–58.

[416] W. Lehr and D. Sicker, "Would you like your Internet with or without video?" in *Proc. SSRN*, 2017, p. 73.

[417] P. Moll, S. Theuermann, N. Rauscher, H. Hellwagner, and J. Burke, "Inter-server game state synchronization using named data networking," in *Proc. 6th ACM Conf. Inf. Centric Netw.*, 2 019, pp. 12–18.

[418] I. Moiseenko and D. R. Oran, "Path steering in content-centric networking (CCNx) and named data networking (NDN)," IETF, RFC 9531, Mar. 2024. [Online]. Available: https://www.rfc-editor.org/info/rfc9531

[419] K. Pugin, A. Frindell, J. C. Ferret, and J. Weissman, "RUSH—Reliable (unreliable) streaming protocol." May 2023. [Online]. Available: https://datatracker.ietf.org/doc/draft-kpugin-rush/02/

[420] J. Ott, M. Engelbart, and S. Dawkins. "RTP over QUIC (RoQ)." Mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-avtcore-rtp-over-quic/09/

[421] M. Papalini et al., "On the scalability of WEBRTC with information-centric networking," in *Proc. IEEE Int. Symp. Local Metropolitan Area Netw. (LANMAN*, 2020, pp. 1–6.

[422] E. André, N. Le Breton, A. Lemesle, L. Roux, and A. Gouaillard, "Comparative study of WebRTC open source SFUS for video conferencing," in *Proc. Principles Syst. Appl. IP Telecommun. (IPTComm)*, 2018, pp. 1–8.

[423] "5G white paper," Next Gener. Mobile Netw., Düsseldorf, Germany, 2015.

[424] T. Hoang, M. Freda, T. Deng, J. Rao, and M. I. Lee, "Methods, architectures, apparatuses and systems directed to relay and path selection and reselection," U.S. Patent Appl 18 007 845, Sep. 2023.

[425] M. Stiemerling, S. Kiesel, M. Scharf, H. Seidel, and S. Previdi, "Application-layer traffic optimization (ALTO) deployment considerations," IETF, RFC 7971, Oct. 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7971

[426] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, "Compute first networking: Distributed computing meets ICN," in *Proc. 6th ACM Conf. Inf. Centric Netw.*, 2019, pp. 67–77.

[427] J. Crowcroft, P. Eardley, D. Kutscher, and E. M. Schooler, *Compute-First Networking (Dagstuhl Seminar 21243)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Wadern, Germany, 2021.

[428] D. K. J. Burke and L. Zhang. "Named data microverse." 2020. [Online]. Available: https://named-data.net/microverse/

[429] O. Hashash, C. Chaccour, W. Saad, K. Sakaguchi, and T. Yu, "Towards a decentralized metaverse: Synchronized orchestration of digital twins and sub-metaverses," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2023, pp. 1905–1910.

[430] J. Radoff. "Metaverse interoperability, part 1: Challenges." Jan. 2022. [Online]. Available: https://medium.com/building-the-metaverse/metaverse-interoperability-part-1-challenges-716455ca439e

[431] T. Yu et al., "Secure web objects: Building blocks for Metaverse interoperability and decentralization," presented at IEEE MetaCom, 2024.

YULONG ZHANG received the B.Eng. degree from the Wuhan University of Technology, China, in 2018. He is currently pursuing the Ph.D. degree with The Hong Kong University of Science and Technology (Guangzhou). His main research interests include designing and optimizing emerging networked systems, distributed computing and networking (compute-first networking, computing in the network), and Internet architecture and decentralized communication.

DIRK KUTSCHER received the Ph.D. degree from Universität Bremen, Germany. He is a Professor with The Hong Kong University of Science and Technology (Guangzhou), where he is the Director of the Future Networked Systems Laboratory. Previously, he was the CTO of Virtual Networking with Huawei Research Germany, a Chief Networking Researcher with NEC Europe, and a Visiting Researcher with KDDI R&D, Japan. He is co-chairing two IRTF Research Groups, ICNRG and DINRG. He has authored several international patents, IETF RFCs, books, and highly cited and awarded research papers. His main interests lie in the intersection of distributed computing and networking for compute-first networking and in Internet architecture. His work has been recognized by best paper and company research awards.

YING CUI received the B.Eng. degree in electronic and information engineering from Xi'an Jiao Tong University, China, in 2007, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2012. She held visiting positions with Yale University, USA, in 2011 and with Macquarie University, Australia, in 2012. From June 2012 to June 2013, she was a Postdoctoral Research Associate with Northeastern University, USA. From July 2013 to December 2014, she was a Postdoctoral Research Associate with the Massachusetts Institute of Technology, USA. From January 2015 to July 2022, she was an Associate Professor with Shanghai Jiao Tong University, China. Since August 2022, she has been an Associate Professor with the IoT Thrust, The Hong Kong University of Science and Technology (Guangzhou), and an Affiliate Associate Professor with the Department of ECE, The Hong Kong University of Science and Technology. Her current research interests include optimization, learning, IoT communications, mobile edge caching and computing, and multimedia transmission. She was selected to the National Young Talent Program in 2014. She received Best Paper Awards from IEEE ICC 2015 and IEEE GLOBECOM 2021. She serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.