Compute- and Data-Intensive Networks: The Key to the Metaverse

Yang Cai
University of Southern California
Los Angeles, USA
yangcai@usc.edu

Jaime Llorca
New York University
New York City, USA
jllorca@nyu.edu

Antonia M. Tulino
New York University
New York City, USA
atulino@nyu.edu

Andreas F. Molisch
University of Southern California
Los Angeles, USA
molisch@usc.edu

***Abstract*—The worlds of computing, communication, and storage have for a long time been treated separately, and even the recent trends of cloud computing, distributed computing, and mobile edge computing have not fundamentally changed the role of networks, still designed to move data between end users and pre-determined computation nodes, without true optimization of the end-to-end compute-communication process. However, the emergence of Metaverse applications, where users consume multimedia experiences that result from the real-time combination of distributed live sources and stored digital assets, has changed the requirements for, and possibilities of, systems that provide distributed caching, computation, and communication. We argue that the real-time interactive nature and high demands on data storage, streaming rates, and processing power of Metaverse applications will accelerate the merging of the cloud into the network, leading to highly-distributed tightly-integrated compute- and data-intensive networks becoming universal compute platforms for next-generation digital experiences. In this paper, we first describe the requirements of Metaverse applications and associated supporting infrastructure, including relevant use cases. We then outline a comprehensive *cloud network flow* mathematical framework, designed for the end-to-end optimization and control of such systems, and show numerical results illustrating its promising role for the efficient operation of Metaverse-ready networks.**

***Index Terms*—Metaverse, virtual reality, augmented reality, immersive video, edge computing, caching, distributed cloud, decentralized control, 5G networks**

I. Introduction

Next-generation (NextG) networks are rapidly evolving towards tightly integrated computing, caching and communication (3C) systems that go beyond current (i) computation-centric cloud data centers interconnected by a wide area network, (ii) communication-centric 5G networks connecting mobile users to cloud resources, and (iii) caching-centric content distribution networks. NextG networks are envisioned to be highly distributed mobile/wireless-first 3C systems, where a wide range of distributed network elements, including end devices, access points, and edge/cloud servers cooperate contributing resources and participating in the routing, storage, and processing functions needed to deliver the services that will define the future of consumer experiences and industrial automation.

Indeed, the ubiquity of live and stored data sources (e.g., real-world sensors and digital assets) fueled by the convergence of physical reality and digital virtuality, and the seamless access to distributed computational resources enabled by NextG networks will essentially blur the space- and time-scale separation between data collection, information processing, and experience delivery, enabling a new breed of *Metaverse applications* that will transform the way we live, work, and interact with the physical world. It is envisioned that Metaverse applications will drive massive investments toward the digitization, automation, and enhanced interactivity of physical systems and human experiences. Augmented/virtual/extended reality (XRs), telepresence, immersive video, digital twins, and multi-player gaming are all examples of Metaverse applications that require real-time (i) aggregation of distributed data streams, (ii) in-network data processing and information synthesis, and (iii) distribution of highly personalized streams to multiple interacting users.

We argue that the unprecedented communication, computation, and storage requirements imposed by the Metaverse will demand a new universal compute platform driven by the efficient integration of 3C technologies into NextG networks, along with new tools and methods for the end-to-end optimization and dynamic control of the resulting global infrastructure. To this end, in this paper:

Prof. Antonia M. Tulino is also with Universityà degli Studi di Napoli Federico II, Naples 80138, Italy.

This work was supported by the National Science Foundation (NSF) under CNS-1816699.

- We illustrate the relevance and generality of Metaverse applications, as well as their unique multi-dimensional requirements.
- We describe the main characteristics of the envisioned Metaverse-ready compute- and data-intensive infrastructure.
- We outline a cloud network flow mathematical framework, especially suited for the integrated control of 3C networks and the end-to-end optimization of Metaverse application delivery.
- We show numerical results that validate the promising role of the described framework to enable the efficient operation of Metaverse-ready networks.

II. Metaverse Applications

The Metaverse, etymologically a combination of *meta* (i.e., *beyond*) and *universe*, refers to a computer-generated world that flexibly blends physical reality and digital virtuality in order to provide immersive, interactive, realistic, and augmented digital experiences, with a wide spectrum of consumer and industrial applications.

Social: The Metaverse, which aggregates and transcends traditional media (text, audio, image, video), will shape the future of online social networks, initially built to connect individual users for communication and content sharing. For example, Meta® (formerly known as Facebook®) is actively developing social VR platforms where users can have digital representations (avatars), interact with each other, and participate in shared activities like shopping, tourism, watching movies, and attending events.

Gaming: The Metaverse has also entered the gaming space (e.g., VR/AR games), making games more realistic and interactive by accelerating efforts on physical world digitization and the enablement of multisensory experiences. Metaverse games such as Horizon Worlds®, Roblox®, and Fortnite® have become widely popular, and movies like Ready Player One® and Free Guy® envision a more interactive mixed-reality world in future Metaverse games.

Industry: In industrial applications, the Metaverse can increase productivity along multiple phases of a product's life cycle. First, product design can be conducted in the Metaverse running accurate simulations, at a lower cost and a faster pace than when creating physical design samples. Second, the Metaverse can use digital twins to enhance operational efficiency and reduce quality control risks in the manufacturing process. Finally, the Metaverse provides a communal space where interdisciplinary teams and customers can have real-time interactions, increasing the efficiency and agility of the product design and development life cycle. The NVIDIA® Omniverse™ Platform is a clear example of a relevant tool for industrial Metaverse applications.

(a) Social (b) Gaming (c) Industry (d) Collaboration (e) Health (f) Education

Fig. 1. Metaverse applications.

Collaboration: In the Metaverse, people can create a personalized workspace or virtual office – a more flexible working environment to facilitate collaboration regardless of geographical restrictions. Digital assets representing people (e.g., avatars) and objects can be incorporated as needed into the 3D space, giving a new dimension to today's online meetings.

Health: The Metaverse also enables important applications in the healthcare industry, including telemedicine, augmented fitness, and in particular, remote surgery. When provided highly realistic environments, remote doctors can perform high-precision operations on a patient's body. Besides, real-time physical conditions of the patient can enrich displayed contents to help doctors' decision making.

Education: The Metaverse transforms the way knowledge is presented. In descriptive or explanatory courses, students can be exposed to visual 3D models with improved clarity compared to any precedent media. In training courses, students can practice their skills in realistic environments and enjoy efficient, low-risk

learning experiences. For example, it enables learning how to manipulate hazardous substances without actual exposure to those substances.

III. Metaverse Infrastructure

In this section, we summarize the unprecedented resource requirements imposed by Metaverse applications, and then describe the main characteristics of the envisioned supporting infrastructure.

A. Resource Requirements

As illustrated in Section II, the Metaverse is foreseen to become a global digital platform where users can consume real-time interactive experiences that seamlessly blend physical reality and digital virtuality. Going beyond content retrieval-and-distribution (e.g., web, video streaming), in the Metaverse, user experiences result from the real-time aggregation, processing/composition, and delivery of multiple live streams and digital assets.

In the following, we describe the computation, storage and communication requirements of Metaverse applications, illustrated in the context of a VR streaming application in Fig. 2.

1) Computation Requirements: Central to Metaverse applications is the blending of physical and digital worlds into rich multimedia immersive environments – a task of high computational demand.

In industrial Metaverse applications, massive computational resources are consumed to build physically accurate simulation environments. Prospective consumer applications will also challenge computing power requirements. For example, running a typical AAA game (e.g., Fortnite®) today requires multiple teraFLOPS of graphics horsepower, and the demand is expected to grow by two orders of magnitude to create fully immersive Metaverse experiences [1]. Even basic video processing tasks (e.g., object recognition and tracking) can exhibit substantial computational complexity when applied to increasingly enriched virtual worlds involving massive numbers of users and digital assets.

In addition, Metaverse applications may involve multiple processing tasks (running as separate service functions) operating on source and intermediate data streams for the generation of the consumable experiences (see, e.g., coding, decoding, and rendering functions for VR streaming in Fig. 2). The requirements of each service function may also depend on the hosting hardware, e.g, general-purpose Central Processing Units (CPUs), Graphic Processing Units (GPUs), or tailor-made computing hardware.

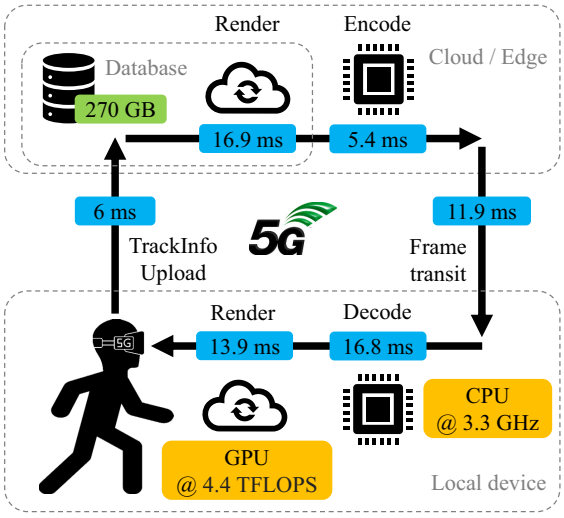

Fig. 2. Resource consumption of *early-stage* VR streaming [2]. The *computation* requirements are highlighted in yellow and the *storage* requirement for a 5-minute 360° video [3] in green. The system attains an end-to-end delay (sum of individual delays in blue) of 70.9 ms, and a perception delay (with motion prediction) of 8.2 ms.

2) Storage Requirements: An explosively growing number of digital assets (representing objects, spaces, attributes, value, etc.) are crowding into the Metaverse.

As key building blocks of Metaverse applications, digital assets are used to compose the immersive experiences consumed by users according to their real-time interactions. For example, digital twins representing the properties of physical systems are essential components of industrial Metaverse applications; another consumer application example, VR streaming, renders 3D scenarios from digital scenes (see Fig. 2). These storage-hungry applications will impose tremendous resource requirements. For example, *Entry-level* VR, designated to support 8K-resolution 360° video streaming for 20-minute experience duration, requires around 10 TB storage space for the produced uncompressed video. Such already high demand is expected to increase in future phases of *Advanced* VR (by a factor of 10) and *Ultimate* VR (by a factor of 100) [2].

3) Communication Requirements: The Metaverse is created to power interactions among the users and with the virtual environment, which requires the real-time aggregation of multiple live streams and digital assets, and the distribution of the resulting processed/composed streams.

Fueled by growing trends in wearable devices and the Internet of things (IoT), a massive number of sensors will continuously collect data about the physical world

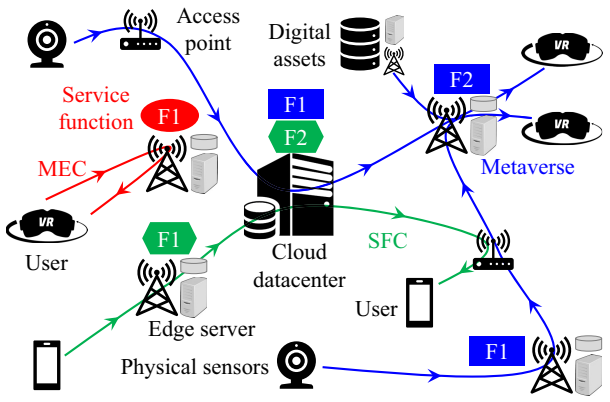

Fig. 3. The envisioned infrastructure for the Metaverse, in which MEC and SFC services can be handled as special cases of general Metaverse applications (see also Fig. 4).

as inputs to the Metaverse. The real-time aggregation of the resulting live data streams, as well as pre-produced digital assets, will continue to accelerate network traffic growth. As projected in [1], the Metaverse could drive video traffic – which already accounts for 80% of today's Internet traffic and grows at a 30% compound annual growth rate (CAGR) – to grow at a 37% CAGR, leading to 24 times the current data usage over the next decade.

Even more critical is the end-to-end latency requirements. To enable real-time interaction, most Metaverse applications will impose end-to-end delay constraints of less than 20 ms (e.g., 7 to 15 ms) [3], which could go as low as 1 ms for tactile applications such as remote surgery. Exceeding these stringent latency requirements not only leads to lagged responses, but also causes discomfort, e.g., dizziness and nausea, in VR applications.

B. Envisioned Infrastructure

While initial VR/AR applications are being supported by local compute platforms such as VR headsets and/or gaming consoles, the full breadth of Metaverse applications, involving distributed remote sources, dispersed users, and multiple service functions, imposing the level of resource requirements described in the previous section, will demand a much more powerful global-scale infrastructure.

To this end, we argue that the Metaverse supporting infrastructure shall be a *universal compute platform running on a highly-distributed tightly-integrated compute-, communication- and data-intensive NextG network*. As shown in Fig. 3, the envisioned NextG network interconnects computing- and caching-enabled user devices, access points, edge servers, and cloud data centers, along a device-edge-cloud continuum. Each node is able to host a set of service functions (depending on their capabilities) that can be dynamically activated to run the supported computation tasks; in addition, each node may use its available local storage to cache Metaverse digital assets. Each link is capable of data transmission between connected devices, laying the foundation for cooperative computing and caching. Aided by advanced network programmability (e.g., software defined networking, SDN) and virtualization (e.g., network function virtualization, NFV) technologies, service functions can be flexibly interconnected and elastically executed at different network locations, while efficiently accessing required digital assets cached throughout the network.

While the worlds of computing, caching, and communication, have mostly evolved separately, recent needs to support new emerging applications have pushed for an increasing level of integrated design. Two research fields, i.e., content distribution and distributed computing (or processing networks) have focused on the integration of caching-communication and computing-communication technologies into network design, respectively. However, related studies have considered relatively simplified network and service models. For example, mobile edge computing (MEC) studies mainly focus on single task offloading, and service function chaining (SFC) on single processing pipelines (as illustrated in Fig. 3), resulting in optimization and control techniques that exhibit suboptimal performance when applied to the generic Metaverse scenario.

The main advantages of a truly integrated 3C infrastructure include: (i) the *joint* optimization of 3C technologies enables higher operational efficiency and QoE (quality of experience), (ii) the adaptability of 3C resource allocations makes the network more flexible and resilient to changing network conditions and service demands, and (iii) the scale and heterogeneity of 3C-equipped network nodes increase service availability and opportunities for enhanced performance.

Maximizing the benefit of this promising paradigm rests on the design of new tools and methods for the end-to-end optimization and dynamic control of such complex global infrastructure. For example, when a service request emerges, the network control policy needs to *coordinate* the selection of (i) caching locations to provide digital objects, (ii) computation locations to execute service functions, and (iii) communication paths to route all associated data streams, *jointly* optimized with dynamic decisions on (iv) traffic scheduling and (v) resource allocation at all network locations.

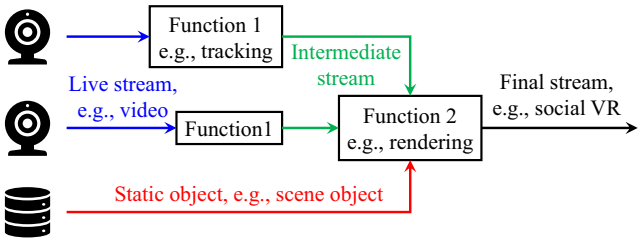

Fig. 4. Service DAG model for general Metaverse applications.

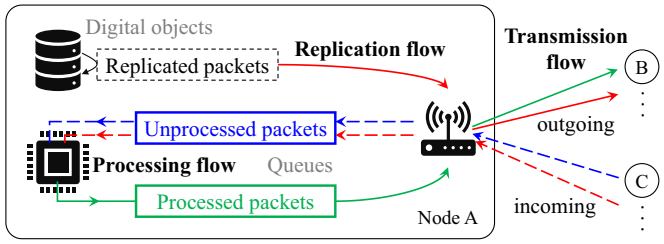

Fig. 5. Cloud network flow model.

IV. Cloud Network Flow Framework

In this section, we outline a comprehensive mathematical framework developed over the past few years, we term *cloud network flow*, that enables the end-to-end optimization and dynamic control of completely general Metaverse applications over 3C NextG networks [4]–[20]. While the description here is kept in high-level form, we refer the reader to the above cited papers for further details, analysis, and results.

A. General System Model

1) Service DAG Model [6]: First, we introduce a generic Metaverse application model, referred to as *service directed acyclic graph (DAG)*, used to describe the set of service functions, and the source and intermediate data streams they operate on, required to generate the final experience, as shown in Fig. 4. Each function is characterized by three parameters: *merging ratio*, *workload*, and *scaling factor*, describing the ratio of individual input stream sizes, the computational resource consumption, and the generated output stream size, per unit of input data, respectively. In general, the input streams include *live data* (collected by sensors) and *static objects* (pre-stored in the network), with associated network locations referred to as *live sources* and *static sources*, respectively [21]. We note that a static object can be provisioned (via replication) by *any* of the associated static sources (i.e., caching locations) in an *on-demand* manner (per service function's request).

2) NextG Network Model: Next, we describe a practical NextG network model that allows characterizing highly heterogeneous and dynamic systems (as shown in Fig. 3) with non-uniform resource distribution in both space and time dimensions. In particular, the processing/transmission capacity of any node/link can be modeled by $C(t) = C(\omega(t), \alpha(t))$, impacted by uncontrollable system states $\omega(t)$ (e.g., channel state), and controllable resource allocation decisions $\alpha(t)$ (e.g., transmitted power).

B. Policy Design

The cloud network flow framework is established for control policy design in 3C networks, encompassing network-layer packet processing, transmission, and replication operations, as well as physical-layer multi-dimensional resource allocation decisions.

1) Cloud Network Flow: We use cloud network flow variable $f(t)$ to represent dynamic control *actions* taken on the 3C infrastructure, i.e., the amount of data packets scheduled for 3C operations. As illustrated in Fig. 5, the cloud network flow model includes transmission flows (characterizing packet forwarding between neighbor nodes), processing flows (characterizing packet processing via local service functions), and replication flows (characterizing the creation of packet replicas from static objects) [6], [8], [12].

There are two main constraints imposed on the flow variables: (i) *Service chaining* constraints, which impose the relationship between input and output flows as they undergo service function processing (including merging ratio and scaling factor parameters); and (ii) *Capacity* constraints, which limit resource usage: the incurred resource consumption shall not exceed the allocated capacity $C(t)$, via which physical- and network-layer decisions interact.

2) Queuing System: Cloud network flow employs a flexible and generalized queuing system to characterize the evolution of the network *state* resulting from the resource allocation and flow scheduling actions taken on the 3C infrastructure. The cloud network flow queuing system $Q(t)$ can include physical queues representing the build-up of packets (i) at different stages of a service DAG (to drive packet processing operations) [10], [12], [13], (ii) with different lifetimes (to drive packet dropping and scheduling operations under strict latency

constraints) [4], [9], and (iii) with different replicating status (to drive packet replication operations for multicast services) [5], [8].

In addition, the framework is flexible enough to admit the definition of virtual queues that allow characterizing other relevant metrics such as anticipated resource loads (exploiting global knowledge) [22] and destination-driven computation and data demands [23].

3) Formulation: We define the system state $s(t)$ and action $a(t)$ to aggregate corresponding physical- and network-layer quantities, given by:

$$s(t) = (\omega(t), Q(t)),\ a(t) = (\alpha(t), f(t)), \tag{1}$$

based on which QoE metrics (e.g., throughput, latency) can be expressed as

$$E_i(t) = E_i(s(t), a(t)),\ i \geq 0. \tag{2}$$

We then formulate the following sequential decision making problem over $\{a(t) : t \geq 0\}$:

$$\begin{aligned} \max\ & \overline{E_0(s(t), a(t))} && \text{(3a)} \\ \text{s.t.}\ & \text{QoE constraints } \overline{E_i(s(t), a(t))},\ i \geq 1, && \text{(3b)} \\ & \text{Chaining and Capacity constraints} && \text{(3c)} \end{aligned}$$

involving multiple QoE metrics, where $\overline{E_i(t)}$ denotes the average performance in terms of metric $E_i(t)$. Under reasonable assumptions, (3) can be transformed into a (constrained) Markov decision process (MDP) problem.

4) Solutions: The formulated (MDP) problem admits standard solutions, e.g., reinforcement learning (RL) [24]. However, while RL methods yield effective solutions for single-agent learning problems, the multi-agent RL variants [25] required to address the end-to-end control of large-scale 3C networks for Metaverse applications can lead to inefficient and unstable training procedures (without convergence guarantees) that result in sub-optimal performance.

A special case of the network control problem (3) that has important practical relevance is the setting in which we only have an action-dependent objective and stability constraints [26], i.e.,

$$\begin{aligned} \max\ & \overline{E_0(a(t))} && \text{(4a)} \\ \text{s.t.}\ & \overline{E_1(s(t), a(t))} = \overline{Q(t)} < \infty, && \text{(4b)} \\ & \text{Chaining and Capacity constraints.} && \text{(4c)} \end{aligned}$$

This problem can be solved leveraging Lyapunov drift-plus-penalty (LDP) control [26], guiding the design of two important classes of network control policies with insightful interpretations: (i) Distributed policies, e.g., DCNC [12], [19], [20], DWCNC [13]–[15], MECNC [10], DECO [23], where routes are dynamically adjusted based on local queue observations, e.g., *differential backlog* between neighbor nodes. (ii) Centralized policies, e.g., UMW [27], UCNC [22], DI-DCNC [6], where routes for each incoming packet are selected based on global queuing states. These LDP based solutions can achieve optimal throughput and, for some of them, operational cost, while guaranteeing bounded average delay, without the need of expensive training (required for standard MDP solutions).

C. Advanced Solutions and Open Problems

1) End-to-End Latency: Timely content delivery is crucial to interactive Metaverse applications. To this end, our recent works [4], [9] address the challenging problem of *timely throughput* analysis and optimization, giving rise to a control policy that makes *lifetime* driven routing and scheduling decisions.[1] When dealing with general Metaverse applications involving multiple concurrent pipelines, the end-to-end service delay should be taken as the *maximum* over the concurrent pipelines, requiring new extensions to the problem formulation and associated optimization methods.

2) Multiple Destinations: The social and interactive nature of Metaverse applications results in many data streams being shared and simultaneously consumed by multiple users/destinations. In our recent works [5], [8], [22], we have developed multicast cloud network control policies that leverage *in-network packet replication* to enhance multicast content distribution. Another promising way to achieve this goal is by exploiting the *broadcast* nature of the wireless medium – incorporating it into the current framework is object of further investigation.

3) Multiple Pipelines: To jointly handle the multiple service pipelines involved in Metaverse applications, our recent work [6] addresses the problem of *coordinated multiple stream routing*, including a control policy that guarantees packet routes from different pipelines to meet at common processing locations. Nonetheless, the problem of *coordinated multiple stream scheduling* remains unsolved and of interest for future work.

V. Example Results

In this section, we conduct numerical experiments to demonstrate the benefit of the envisioned 3C infrastructure and associated control framework in the context of a

[1] In this study, each packet is assigned a strict deadline. The packet lifetime indicates the *time to live* (or *time till deadline*), and the timely throughput the *on time* (or *by deadline*) *service delivery rate*.

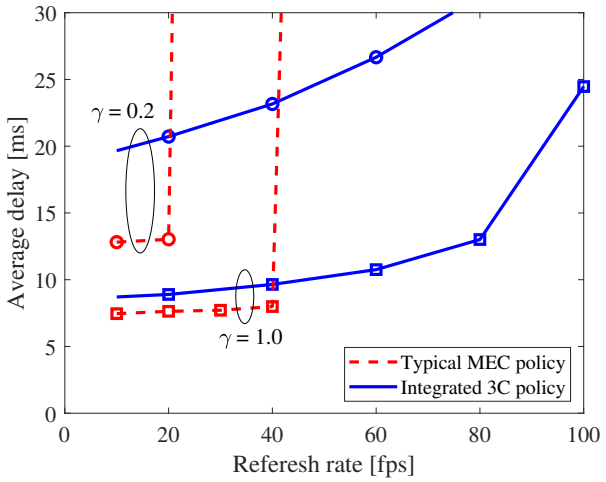

Fig. 6. Effects of 3C integration (storage capacity $\beta_3 = 30\%$).

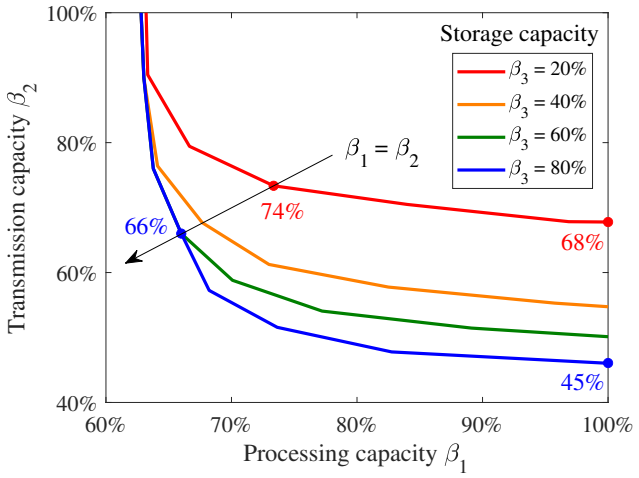

Fig. 7. Tradeoffs between multi-dimensional resources ($\gamma = 1$).

mobile VR application. The VR application generates the consumed 3D field-of-view (FOV) from digital 2D FOV images selected based on the user's tracking information, as shown in Fig. 2. We employ the algorithm in [6] to perform joint dynamic decisions on static source (i.e., cache) selection, processing location, and route selection for each service request.

A. Experiment Setup

Consider a 100 m $\times$ 100 m square area with 100 randomly moving users and a base station (BS) at the center. Each user requests the described VR application at a constant rate λ (i.e., refresh rate, in *frames per second*, fps), and the requested 2D images are selected from a library (of 10^4 images) following a Zipf distribution of parameter $\gamma_p = 1$. Each 2D image has a size of 3 Mb, and the VR processing requires 3×10^7 computing cycles to generate a 6 Mb 3D FOV [3].

Each user is equipped with a 3 GHz processor, and can cache part (a ratio of β_3) of the library according to some caching distribution (assumed to be another Zipf distribution of parameter γ). The BS is equipped with 10 identical processors and has the entire library stored. Users can collaborate via device-to-device (D2D) communication: at every time slot, each user can communicate with one of its neighbors within a cooperation range of 20 m, with 1 W transmission power and 20 MHz bandwidth. The BS can select 20 users to serve in each slot, at a rate of 200 Mbps for each of them.

B. Results

1) Impact of 3C Integration: We first compare the performance of the 3C network running the developed control policy with respect to a typical MEC scenario focusing on individual users offloading tasks to the BS. The users' storage capacity is set to $\beta_3 = 30\%$, and two caching distributions, $\gamma = 0.2$ and 1, are evaluated.

The results are shown in Fig. 6, and we make three observations. First, the integrated 3C policy effectively improves the attained throughput (the blue-square curve achieves 90 fps under 20 ms delay requirement), critical for higher VR QoE. Second, the caching distribution significantly impacts the throughput and delay performance, as different γ values lead to different average hop distances to find the requested images. Finally, the delay attained by the algorithm in the low-congestion regime (e.g., $\lambda \leq 40$ fps) is not ideal, which can be mitigated by the modified design described in Section IV-C1.

2) Multi-Dimensional Resource Tradeoffs: Next, we evaluate the tradeoffs between processing, transmission, and storage resources, assuming $\lambda = 60$ fps refresh rate and 20 ms delay requirement. The available processing and transmission capacities at each user are given by β_1 and β_2 (in percentage of corresponding maximum budgets), respectively. We then define the *feasible region* as the collection of (β_1, β_2) pairs under which the delay requirement is fulfilled.

Fig. 7 depicts the feasible regions with different storage capacities (using the caching distribution of $\gamma = \gamma_p = 1$). Since higher QoE (e.g., lower latency) can be attained with more resources, i.e., $(\beta_1, \beta_2) \to (1, 1)$, the feasible regions are to the upper-right of the border lines. As we increase the storage resource β_3 from 20% to 80%, more processing and transmission resources can

be saved: when $\beta_1 = \beta_2 = \beta_{12}$, the resource saving ratio, i.e., $1 - \beta_{12}$, grows from 26% to 34%. A larger saving ratio can be achieved when further narrowing the focus on one resource dimension: when $\beta_1 = 1$, i.e., all devices operate at maximum *compute* capacity, even more transmission resources, e.g., *bandwidth*, can be saved, i.e., $1 - \beta_2$, from 32% to 55% (provided the same storage resources).

VI. Conclusions

In this paper, we first illustrated the generality and relevance of Metaverse applications and their unique multi-dimensional requirements. We then described the main characteristics of the envisioned Metaverse-ready compute-, communication-, and data-intensive infrastructure, followed by a cloud network flow mathematical framework for its end-to-end optimization and dynamic control. Numerical results validated the promising role of the envisioned 3C infrastructure to support Metaverse applications, and of the described framework for its efficient operation.

References

[1] "Metaverse: A guide to the Next-Gen Internet," Credit Suisse, 2022. [Online]. Available: https://www.credit-suisse.com/media/assets/corporate/docs/about-us/media/media-release/2022/03/metaverse-14032022.pdf.

[2] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, "Cellular-connected wireless virtual reality: Requirements, challenges, and solutions," *IEEE Commun. Mag.*, vol. 58, no. 5, pp. 105–111, May 2020.

[3] 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.

[4] 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. 1, no. 99, pp. 1–16, 2022.

[5] ——, "Decentralized control of distributed cloud networks with generalized network flows," arXiv:2204.09030. [Online]. Available: https://arxiv.org/abs/2204.09030, Apr. 2022.

[6] ——, "Joint compute-caching-communication control for online data-intensive service delivery," arXiv:2205.01944. [Online]. Available: https://arxiv.org/abs/2205.01944, May 2022.

[7] ——, "Dynamic control of data-intensive services over edge computing networks," submitted to IEEE GlobeCom 2022.

[8] ——, "Optimal multicast service chain control: Packet processing, routing, and duplication," in *Proc. IEEE Int. Conf. Commun.*, Montreal, Canada, Jun. 2021, pp. 1–7.

[9] ——, "Optimal cloud network control with strict latency constraints," in *Proc. IEEE Int. Conf. Commun.*, Montreal, Canada, Jun. 2021, pp. 1–6.

[10] ——, "Mobile edge computing network control: Tradeoff between delay and cost," in *Proc. IEEE Global. Telecomm. Conf.*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[11] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Service placement and request routing in MEC networks with storage, computation, and communication constraints," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1047–1060, Jun. 2020.

[12] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2118–2131, Oct. 2018.

[13] ——, "Optimal control of wireless computing networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8283–8298, Dec. 2018.

[14] ——, "Impact of channel state information on wireless computing network control," in *Proc. IEEE ACSSC*, Oct 2017, pp. 519–525.

[15] ——, "On the delivery of augmented information services over wireless computing networks," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–7.

[16] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[17] H. Feng and A. F. Molisch, "Diversity backpressure scheduling and routing with mutual information accumulation in wireless ad-hoc networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7299–7323, Dec. 2016.

[18] M. Barcelo, A. Correa, J. Llorca, A. M. Tulino, J. L. Vicario, and A. Morell, "IoT-cloud service optimization in next generation smart environments," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4077–4090, Oct. 2016.

[19] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 1–7.

[20] ——, "Dynamic network service optimization in distributed cloud networks," in *Proc. IEEE INFOCOM WKSHPS*, Apr. 2016, pp. 300–305.

[21] K. Poularakis, J. Llorca, A. M. Tulino, and L. Tassiulas, "Approximation algorithms for data-intensive service chain embedding," in *Mobihoc '20*, Virtual Event, USA, Oct. 2020, pp. 131–140.

[22] J. Zhang, A. Sinha, J. Llorca, A. M. Tulino, and E. Modiano, "Optimal control of distributed computing networks with mixed-cast traffic flows," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1760–1773, Aug. 2021.

[23] K. Kamran, E. Yeh, and Q. Ma, "DECO: Joint computation scheduling, caching, and communication in data-intensive computing networks," *IEEE/ACM Trans. Netw.*, vol. 1, no. 99, pp. 1–15, 2021.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[25] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[26] M. J. Neely, *Stochastic network optimization with application to communication and queueing systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[27] A. Sinha and E. Modiano, "Optimal control for generalized network flow problems," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 506–519, Feb. 2018.