Customizable and Robust Internet of Robots Based on Network Slicing and Digital Twin

Kai Liang, Wei Guo, Zan Li, Cheng Li, Chunlai Ma, Kai-Kit Wong, and Chan-Byoung Chae

***Abstract*—The Internet of Robots (IoR) is proficient in handling complex tasks in challenging environments, yet it encounters challenges related to service and scenario diversity, risk reduction, and ultra-low latency requirements. To address these challenges, we propose an integrated architecture that enhances the IoR's adaptability, flexibility, robustness, and low latency. This is achieved through the introduction of network slicing, service-based architecture, and digital twin (DT). We have developed an open-source experimental platform to showcase the customizability of the proposed architecture. Slices with different requirements are set up in WiFi and cellular scenarios to demonstrate its versatility. Additionally, we present a DT-assisted deep reinforcement learning (DRL) approach for the IoR to improve DRL performance and mitigate risks associated with undesirable actions. The DT is employed to predict rewards and dynamic state transitions in the physical environment. Furthermore, we introduce a resource allocation method that combines data processing queue preemption and spectrum puncturing. This is designed to accommodate coexisting services, specifically enhanced mobile broadband (eMBB) and bursty ultra-reliable low latency communications (URLLC). Experimental and numerical results validate the effectiveness of our proposed methods, showing improvements in customizability, robustness, latency, and outage probability in IoR.**

***Index Terms*—Internet of Robots, Network Slicing, Digital Twin, Service based architecture, Co-existence of eMBB and bursty URLLC.**

I. Introduction

THE remarkable success of artificial intelligence (AI) and manufacturing processes in the past decade has spurred the rapid development and widespread application of robots. Recently, there is a growing trend for robots to tackle complex tasks, meet stringent quality of service (QoS) requirements, and operate in harsh environments. Individual robots often face challenges in performing these complex tasks due to limitations in their capabilities of perception, computation, and communication. This necessitates the advent of the Internet of Robots (IoR), a specialized robotic communication network that integrates AI, cloud/edge computing, communications, and robots [1]. This integration significantly enhances their ability and performance in handling complex tasks. While the well-studied Internet of Things (IoT) has provided valuable research and development experiences [2], the unique characteristics of IoR present new challenges.

The IoR has shown rich diversity in terms of scenarios, services requirements, communication protocols, hardware and software, imposing many challenges to the IoR. Specifically, setting up various physical networks to realize the diversity of IoRs would increase the cost of research, deployment, configuration, and maintenance. Besides, network elements and robots supporting different protocols (e.g., WiFi, 4G, and 5G etc.) deteriorate the scalability and flexibility of the IoR. Moreover, vendor-specific network devices and robots are difficult to share resources due to their closed nature, which in turn results in lower resource utilization. Network slicing [3], an enabling technology in B5G and 6G, can provide customized virtual networks on the shared network infrastructure to fulfill the various requirements and scenarios of the IoR. Its software-defined and virtualized capabilities can improve the IoR's resource utilization, flexibility, and deployment as well as management efficiency, thus addressing challenges arisen from diversity of IoRs.

The IoR with self-decision and self-control capabilities requires more careful design to prevent accidents and hazards (such as high-speed moving robot collisions). Besides, in recent years, there is a greater tendency for IoRs to be deployed in harsh environments to perform highly sophisticated tasks, which increases challenges to developing and updating IoRs, particularly with respect to the difficulty and cost of functional experimentation. The emerging Digital twin (DT) enables the setting up of digital clones (also called twins) of physical objects. By introducing DT, developers are able to design, deploy, improve and manage IoR in digital clones more efficiently, preventing the risks in physical environment associated with poor decisions, and accelerating new feature development cycles with low validation costs. In addition, combining AI (e.g., deep learning, deep reinforcement learning (DRL), etc.) with DT allows for efficient and secure deployment and management the IoR [4].

Among key performance indicators, low latency requirement is one of the most important indicator for the IoR, especially for applications like telesurgery, remote robotic arms, and autonomous driving. However, in scenarios where enhanced mobile broadband (eMBB) and bursty ultra-reliable low latency communications (URLLC) coexist [6], traditional resource allocation algorithms struggle with timely solutions,

K. Liang and W. Guo are with the School of Telecommunications Engineering, Xidian University, Xi'an, 710071, China (email: kliang@xidian.edu.cn, wguoone@163.com). K. Liang is also affiliated with Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation. Z. Li (corresponding author) is with the State Key Laboratory of Integrated Service Networks, Xidian University, China (email: zanli@xidian.edu.cn). C. Li is with Xi'an Aerospace Precision Electromechanical Institute, Xi'an, 710199, China (e-mail:xevillee@126.com). C. Ma is with the College of Electronic Engineering, National University of Defense Technology, Hefei, 230037, China (email: machunlai17@nudt.edu.cn). K. K. Wong is affiliated with the Department of Electronic and Electrical Engineering, University College London, Torrington Place, WC1E 7JE, United Kingdom and he is also affiliated with Yonsei Frontier Lab, Yonsei University, Seoul, Korea (e-mail: kai-kit.wong@ucl.ac.uk). C.-B. Chae is with School of Integrated Technology, Yonsei University, Seoul, 03722, Korea (email: cbchae@yonsei.ac.kr).

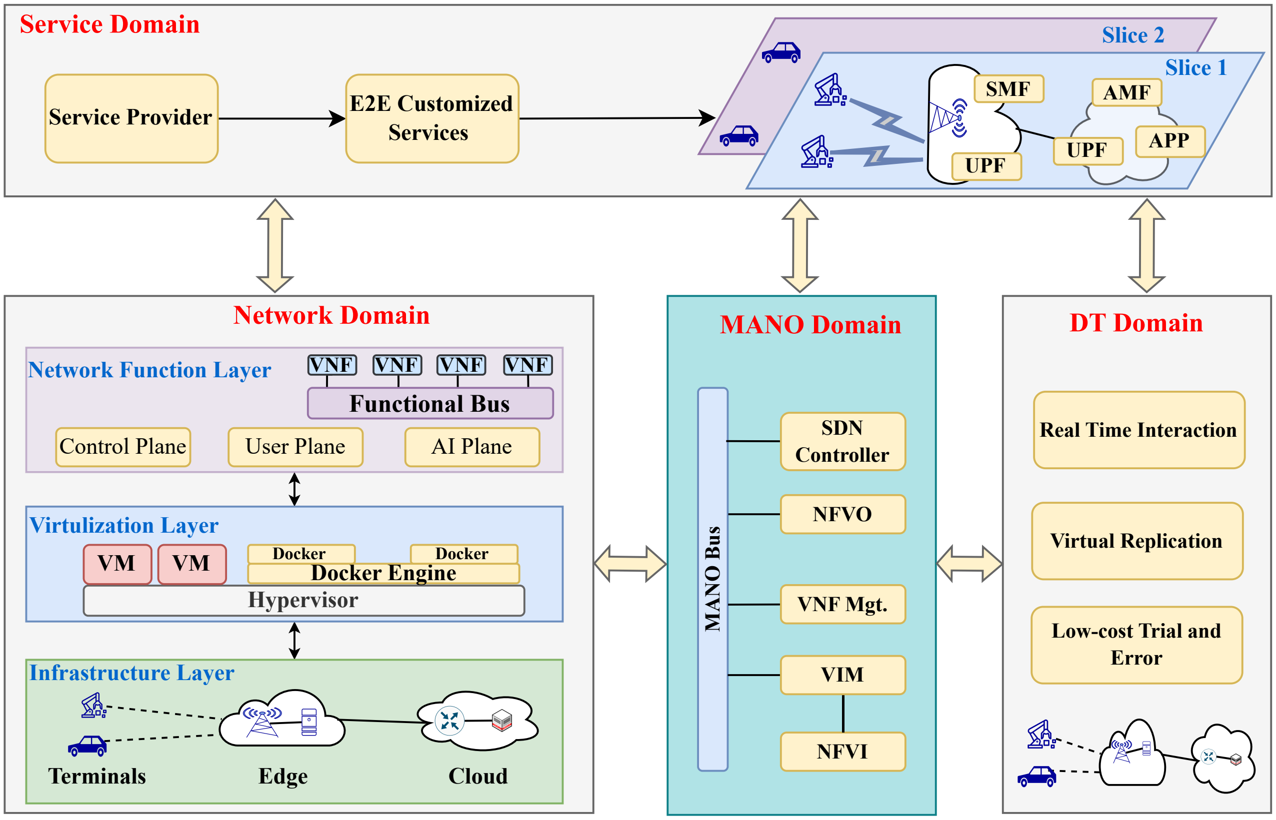


Figure 1. Integrated Network architecture for the IoR.

and resource reservation methods may reduce IoR's resource utilization. Efficiently and dynamically allocating communication and computational resources in such scenarios remains a challenge.

This article presents a network architecture and solutions to fulfill the IoR's demands of diversity, risk avoidance, and ultra-low latency and reliability. Main contributions are summarized as follows.

- We present an integrated network architecture by the cooperation of network slicing, cloud/edge computing, DT and AI, which enables the IoR with the features of flexibility, low latency, customizability and robustness.
- We provide a DT-assisted DRL approach for the IoR to speed up convergence of DRL and reduce the risks caused by bad actions, where the DT uses neural networks to predict reward and state transition of physical environment.
- We provide a resource allocation method for the co-existence scenario of eMBB and bursty URLLC applications, based on data processing queue preempting in the base station and spectrum puncturing in the air.
- We set up an experimental platform to validate the IoR customization capability and demonstrate the advantages of the above DT-assisted DRL approach and resource allocation method in terms of DRL rewards, risk reduction and latency through simulations.

The rest of this article are organized as follows. We propose an integrated network architecture and the experimental results to demonstrate its customizability, based on which we provide a DT-assisted DRL approach for IoR. Then, we provide a joint spectrum and computing resource allocation method for the coexistence scenario of eMBB and URLLC, followed by the conclusion of this article.

II. Integrated Network Architecture for the IoR

This section proposes an integrated network architecture to meet the challenges of service diversity, stringent QoS requirements, and risk avoidance in the IoR, as shown in Fig. 1. We have made comprehensive use of advanced technologies such as software-defined networking (SDN), network function virtualization (NFV), network slicing, service-based architecture (SBA), DT and AI, which enables the IoR to be highly flexible, agile, customizable and robust.

The overall architecture includes four domains, which are detailed in the following:

- *The network domain* represents the physical network that provides information services to robots and is the carrier for the implementation of the architecture. It consists of three logical layers. The infrastructure layer includes heterogeneous network elements (such as base stations (BS), servers, etc.), and multiple resources (e.g., radio, computing, and sensing resources). With the aid of virtualization technologies (e.g., virtual machine and docker), these elements and resources can be abstracted and shared in the virtualization layer. The network function layer provides the basic network functions for IoR. This functions can be further classified into functional planes for easy management as in 6G network architecture [5], such as control plane, user plane and AI plane.

- *The management and orchestration (MANO) domain* is responsible for the orchestration and configuration of network resource, functions and slices, in accordance with the requirements of services and network operation. The objects of MANO domain orchestration include the network domain, the DT domain and the service domain. It can receive network configuration parameters from optimization algorithm, AI, and the DT domain to orchestrate and manage the physical network, enhancing the network's adaptability to new services and scenarios with differentiated needs.
- *The DT domain* utilizes advanced AI algorithms (e.g., DRL) and sensed data to model and then generate virtual images of parts or the whole of the physical network. We can enter requirements and corresponding data through real time interfaces and validate network operations and intelligent applications at the DT domain, followed by which optimal decisions are sent to the MANO domain or the physical network. It enables low-cost trial and error, intelligent decision making, efficient collaboration, and fast responsiveness, significantly reducing the destructive impact of some bad decisions on the network.
- *The service domain* includes all the IoR applications that build network services through the use of network slicing.

The advantage of the proposed architecture lies in multi-folds.

(1) Flexibility. Note that the SBA is the cornerstone of the proposed architecture, which was a 5G core network architecture accepted by 3GPP Release 15 [7]. Inspired by the idea of microservices, the SBA connects required virtual network functions (VNFs) over a unified serviced bus (e.g., HTTP) based on service requirements, which in turn allows for the provision of customized services. This SBA principle allows for the implementation of unified service provisioning, routing, monitoring, and slicing of IoR services, which improves IoR flexibility by leveraging the advanced technologies of the public cloud in an efficient manner.

(2) Low latency. Edge computing and AI enable distributed intelligent computation and execution of complex tasks, which accelerates service responsiveness, and reduces the latency. Besides, benefiting from SBA, slice templates can be set up for rapid instantiation of common services as service requirements arise, thus reduce slice creation time and improves service responsiveness and efficiency. The network-wide programmability and resource shareability brought about by SDN and NFV make it easier and faster to schedule and manage network resources.

(3) Customizability. The proposed architecture gives the IoR the ability to customize its services on shared the infrastructure and resources. Specifically, it utilizes network slicing to create logically separated virtual networks, improving the resource utilization, and network operation and maintenance efficiency. Different slices can be orchestrated by the MANO domain for diverse devices, functions, and networks to meet the various requirements of IoR application and scenarios. These slices are eventually rendered in the service domain and form slices-as-a-service.

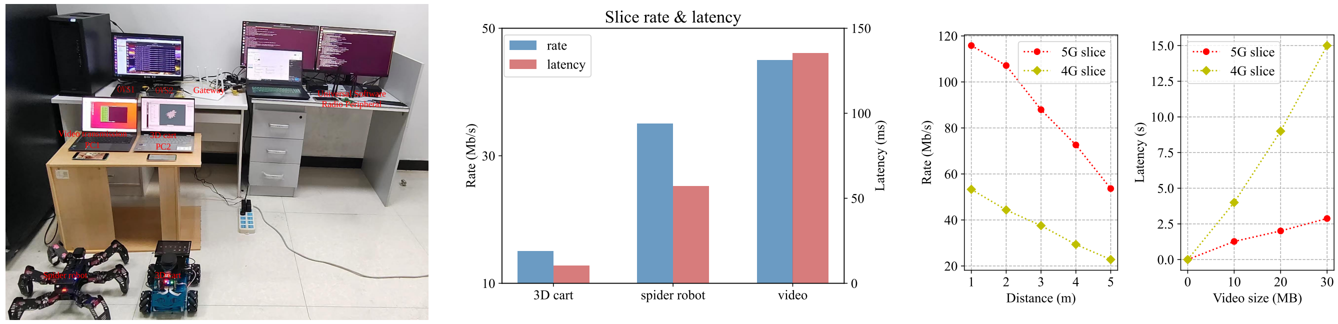
(4) Robustness. The encapsulation capabilities of NFV and network slicing, and the modular design capabilities of the SBA allow for faster and more robust development and upgrades of new applications, where an errant network element, network function, or virtual network does not affect others. Moreover, the introduction of DT greatly avoids risks during the design and operation of IoR, so complex tasks can be continuously tried and tested in DT to find better decisions.

(5) Efficiency. Network slicing provides customization, virtualization and programming capabilities, improving network management efficiency and resource utilization. DT enables verification and trial-and-error in the digital world, and improves the efficiency of IoR's new functionality development.

We set up an experimental platform to demonstrate the flexibility and customization capability of the proposed architecture by creating network slices for different services, while other key technologies for DT and low latency are described later. In practice, robots adopt diverse standards (e.g.,WiFi, 4G, and 5G), so our platform builds different slices on shared physical facilities with these various standards. Specifically, this platform comprises a WiFi-based platform and a cellular-based platform, which employs universal X86 servers, the universal software radio peripheral (USRP), routers, a Quectel RM500Q-GL-based 5G module, smartphones and several miniPCs, as depicted in Fig. 2a.

In the WiFi-based platform, we utilize virtual machine (VM) and Docker for the virtualization of computing and storage resources. We employ the Openwrt to implement Open vSwitch (OVS), supporting the virtualization of communication resources. Using Floodlight and OpenWrt, we realize virtual network functions such as virtual SDN controllers and virtual access points (VAPs). Additionally, we use FlowVisor to orchestrate and isolate network slices, forming dedicated virtual networks. We deploy and test three types of network slices, namely, video transmission slice, spider robot slice (equipped with high-definition cameras for visual recognition, visual synchronization, and intelligent patrol), and 3D cart slice (simultaneous localization and mapping), and test their transmission rate and latency performance respectively, as shown in Fig. 2b. Bandwidth resources are allocated to video transmission slice, spider robot slice and 3D cart slice on demand, yielding transmission rates of 45 Mb/s, 35 Mb/s and 15 Mb/s, respectively. These results align with the allocated resources, ensuring bandwidth isolation between network slices. 3D cart transmits a smaller amount of data with a minimum latency of about 10 ms, followed by the spider robot with the latency of 52 ms, and the video transmission service has the latency of 135 ms due to its larger amount of data.

In the cellular-based platform, we utilize Docker to implement VNFs, and use HTTP bus to connect these VNFs. Kubernetes implements microservice orchestration and management to support slice instantiation. Following that, we deploy the virtual core network of 4G (i.e., evolved packet core) and 5G on the cloud servers, respectively. For the access network, we established virtual eNodeB and gNodeB on a miniPC, and then use two USRP B210s as the radio remote unit (RRU).



(a) (b) (c)

Figure 2. (a) Experimental platform; (b) WiFi-based slices; (c) Cellular based slices.

As shown in the left of Fig. 2c, the rates of the slices decrease with the increasing distance from the BS to the device. At a distance of 1 meter, the 5G slice achieves a transmission rate of about 116 Mb/s, which is twice the rate of the 4G slice. In the right part of Fig. 2c, video transmission latencies shows decreasing trends with the increasing data size. For a video size of 30 MB, the transmission latency of the 5G slice is around 2.9s, about 12.1s lower than the 4G slice's latency. These results clearly demonstrate the superior performance of 5G slices compared to 4G slices.

III. DT-ASSISTED DRL FOR THE IOR

Conventional robots transmit sensing information (such as position, velocity, temperature and humidity) into a pre-written program to obtain their corresponding actions. This approach highly depends on specialized hardware capabilities and highly structured environments, applying only to easy tasks, such as robotic arms next to conveyor belts and shakers.

The DRL is more suitable for dynamically harsh environment and complex tasks, which benefits decision making and control of robots. Specifically, the agent of DRL selects actions for the IoR, and obtains rewards and next state from the physical environment. Through continuous dynamic trial and error, DRL can find the optimal action for the IoR. However, to choose favorable actions or policies, the agent need to frequently interact with the physical environment. This results in that the environment needs to run actions and score them, which incurs additional latency and energy consumption, and bad actions will be destructive (e.g., robot collisions, damage to critical equipment, etc.).

In view of this, we propose a DT-assisted DRL approach for the IoR to improve the performance of the DRL algorithm and reduce risks caused by bad actions. The main idea is that we use DT to mimic the physical environment, that is DT predicts the reward and dynamic state transition of the environment. Thus, incidence of risk is reduced due to less interaction with the physical environment. We put our emphasis on the model based DRL [8] because it can establish an environmental prediction model that meets the needs of the DT simulation environment. The proposed method is schematically shown in Fig. 3, and detailed in the following.

First, we use conventional DRL algorithm (e.g., soft actor-critic, SAC) for the IoR's control and resource management.

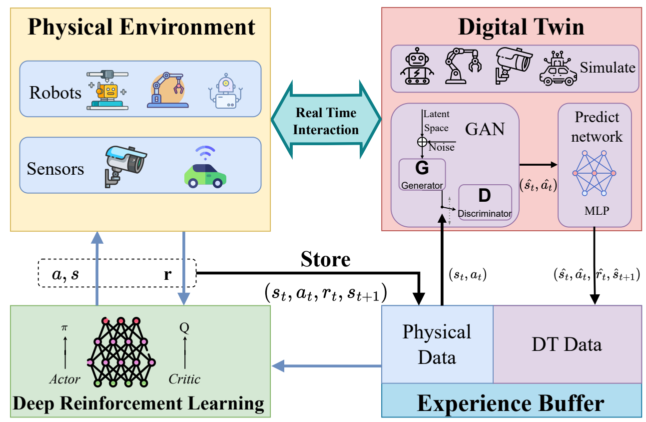


Figure 3. DT-assisted DRL for the IoR.

Depending on the state changes caused by the action, the physical environment feeds back to the agent the value of the reward. The DRL may find the optimal policy or action through dynamic trial and error. Experience reply is a common method to improve the sample efficiency of off-policy DRL algorithm. An experience is made up of several transitions, and a transition consists of a tuple (s_t, a_t, r_t, s_{t+1}), where t stands for the index of training step.

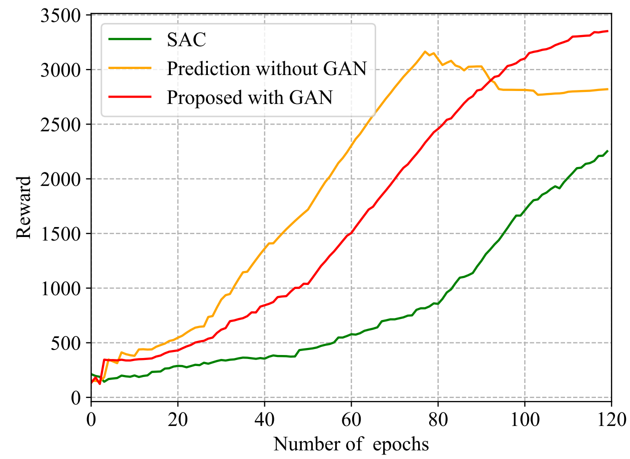
Next, we set up the DT system to predict the state transfer and reward mechanisms of the physical environment via deep learning. Basically, the experiences are used as the data set to train the the predict network (e.g., multilayer perceptron, MLP), where inputs are the current state s_t and selected action a_t, and outputs are the estimated reward $\hat{r}_t$ and state of next step $\hat{s}_{t+1}$. Once the neural network performs well, we can use it to predict $\hat{r}_t$ and $\hat{s}_{t+1}$ without real interaction with physical environment. We can further store the tuple $(s_t, a_t, \hat{r}_t, \hat{s}_{t+1})$ and build a DT experience buffer. Then, DRL can use both the real and the DT experience buffer to speed up its convergence. Furthermore, we add a generate adversarial network (GAN) [9] due to its data augmentation ability before the predict network to generate more fake samples $\hat{s}_t$ and $\hat{a}_t$, and submit them to the predict network to obtain $\hat{r}_t$ and $\hat{s}_{t+1}$. Then we can obtain more DT experiences ($\hat{s}_t$, $\hat{a}_t$, $\hat{r}_t$, $\hat{s}_{t+1}$) that accelerate DRL convergence without actual interactions with the physical environment.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3375503

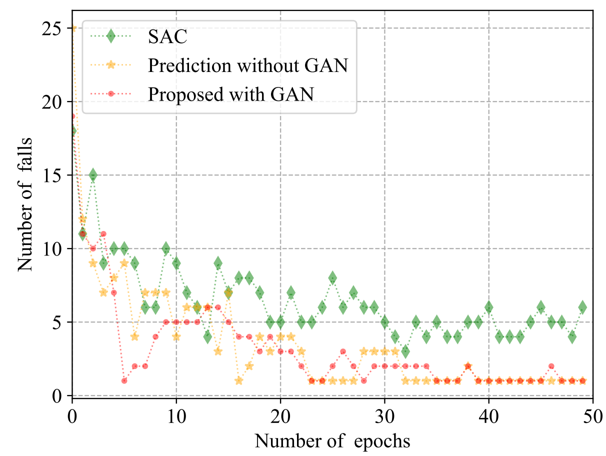
To validate the proposed DT-assisted DRL approach, we utilize PyTorch and the Hopper-v2 environment to construct the model and environment, respectively. We respectively use 60000 and 10000 samples to train and test the overall neural network. Both the SAC network and the predict network adopt batch size of 256, learning rate 0.0003, and Adam optimizer with mean square error loss, where the former one includes three fully connected layers (FCLs) each with 256 neurons, and ReLU activation function, and the latter one consists of five FCLs each with 200 neurons and Sigmoid activation function. The batch size, learning rate, optimizer and number of neurons in each hidden layer of GAN are 64, 0.00005, RMSprop, and 100, respectively. The generator includes four blocks with a FCL, Batch Normalization, and LeakyReLU activation, while the discriminator comprises a FCL and LeakyReLU activation function, ending with a FCL. Then we compared three algorithms: the SAC algorithm, the prediction approach without GAN, and the proposed approach based on GAN.

Fig. 4a shows the reward values of the three algorithms across training epochs. Our proposed algorithm converges faster than the SAC algorithm and achieves higher reward. For instance, the proposed approach with GAN achieves the highest reward at the 120th epoch, around 1.2 and 1.4 times higher than rewards of the prediction approach without GAN and the SAC algorithm, respectively. Within the initial 80 training epochs, the prediction approach without GAN converges the fastest. However, after the 90th epoch, the proposed approach with GAN surpasses the reward values and convergence speed of the prediction approach without GAN, maintaining a stable growth trend. This is because in the initial stages of training, the GAN network has not fitted adapted to the data from the physical environment. Once the GAN network becomes more stable, it provides more accurate training data, which accelerates the DRL training process.

Fig. 4b shows the number of robot falls during each epoch (which includes 1000 training steps) in the physical environment, across a total of 50 epochs. Initially, when the neural network has not yet stabilized, the robot is prone to falling, making the results particularly representative. The proposed approach with GAN achieves the smallest number of falls, about 30 and 50 percentage smaller than the prediction approach without GAN and SAC algorithm at the 9th epoch, respectively. This is because once we detect a reward reducing to a given value in the DT environment, we will not output the corresponding action in the physical environment, thus reducing the risk rate of the robot producing a bad action with destructive consequences in the physical environment. The total number of robot falls for the proposed with GAN across 50 epochs is 166, lower than that of prediction without GAN (i.e., 184), because data samples of the latter one are obtained by interacting with real environments. In contrast, the proposed with GAN method generates data samples using the GAN without interacting with physical environment, thus increasing the reward value while reducing the chance of robot falls.



(a)



(b)

Figure 4. (a) Reward vs. Epochs; (b) Falls vs. Epochs.

IV. Resource Allocation for coexistence of eMBB and Bursty URLLC

This paper considers coexist of eMBB and bursty URLLC because it is more realistically in the practical scenarios. For instance, in robot-assisted industrial automation and telesurgery, both the high data rate video transmission for eMBB and the latency-sensitive controls for URLLC are required. Meanwhile, control data delivery is not continuous and uniform, but rather bursty and with varying amounts of batches over multiple time slots, so we consider bursty URLLC. Typically, network operators are tend to reserve a portion of earmarked resources for URLLC services to meet their stringent requirements, but it reduces the resource utilization of the IoR. Normally, we can preempt the resources allocated to eMBB for the data processing and transmission of URLLC with higher service priority. However, eMBB and URLLC transmit at different time scales, and 3GPP suggests the adoption of short transmission time interval (TTI)/punched approach [10]. Specifically, the time domain is divided into equal slots (1 ms) and each slot is further divided into multiple

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2024.3375503

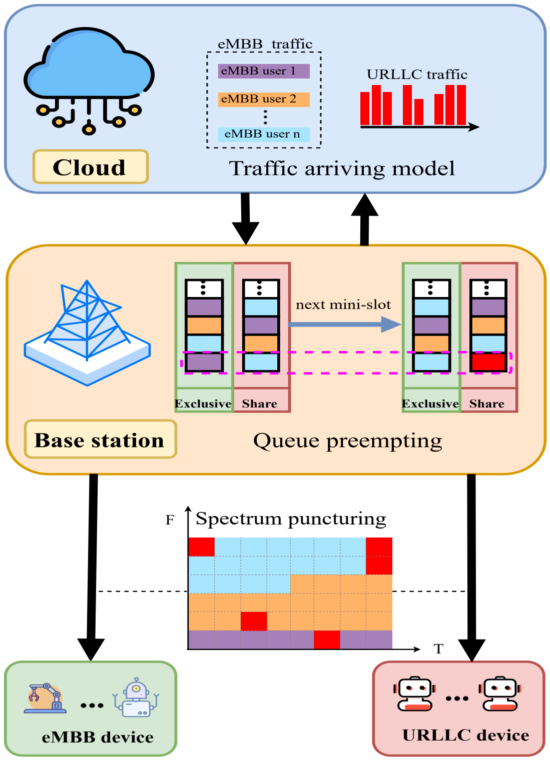


Figure 5. Coexisting eMBB service and bursty URLLC service models.

mini-slots. eMBB transmissions are performed on slots to achieve high data rate and URLLC packets are transmitted on mini-slots to reduce latency. Short packets are used to ensure that URLLC packets can be transmitted in mini-slots. This leads to the Shannon's capacity underestimating rate performance of URLLC, and the achievable rate in short blocklength regime is adopted [11].

The delay of URLLC service mainly consists of the data processing delay of the wired network and the data transmission delay of the wireless link. We propose a method to jointly allocate computing and communication resources by preempting both data processing queues in the BS and radio time-frequency resources in air interface for coexisting eMBB services and bursty URLLC services, as shown in Fig. 5. The main steps of this method is given in the following.

(1) Arriving traffic modeling. Normally, the arriving traffic of eMBB and URLLC are modeled as Poisson process with different mean value. Nevertheless, Poisson process is only suitable for modeling the traffic arrival is constant for each time slot, and fails to model the bursty arrivals (or batch arrivals) of URLLC traffic. Thus, followed by [12], we model the bursty URLLC traffic arrival process as a composite Poisson process. Specifically, the URLLC packets arrive in batches and the number of arriving packets in each batch follows Poisson distribution. Meanwhile, the batch arrivals and the arrival interval follow Poisson process and exponential distribution, respectively.

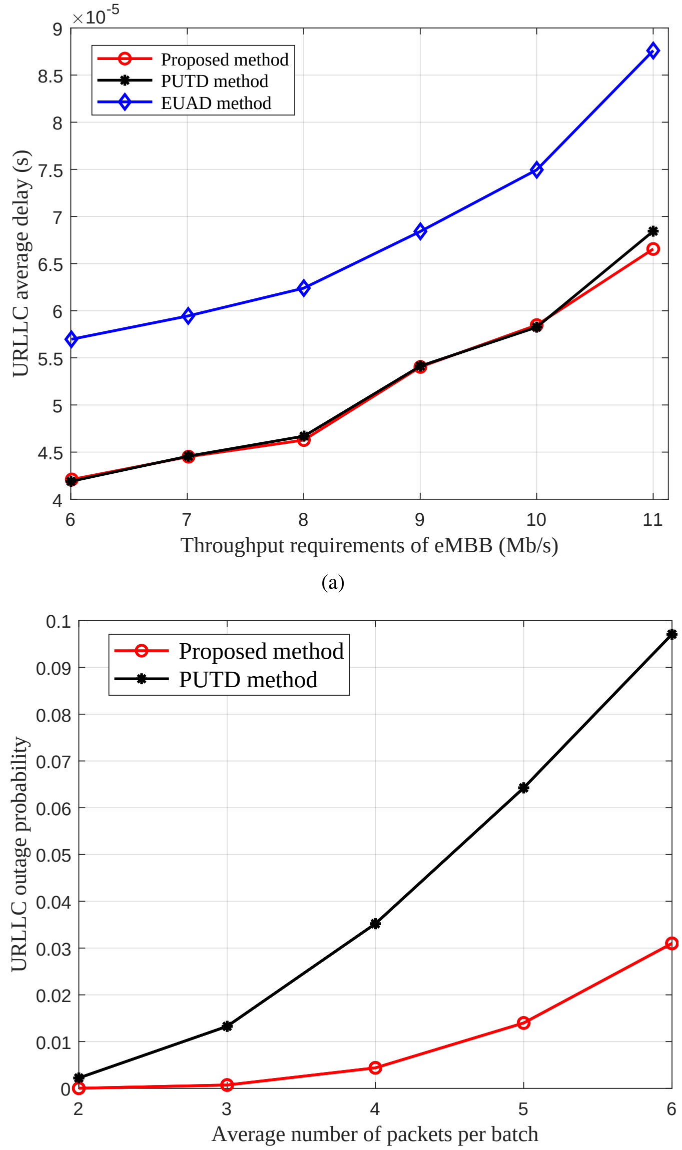
(2) Computing resource sharing and priority queuing rules for data processing. The BS has to process the data download from the cloud. URLLC data with higher priority needs to preempt the data processing queue in the BS, and thus we should introduce preemptive-resume priority rules [13] instead of the conventional first-in-first-out (FIFO) queuing rule. The preemptive-resume priority rules allows URLLC packets to interrupt data processing on the eMBB which will be resumed after URLLC data processing finished. It benefits URLLC performance but decays the rate and latency of eMBB services. To solve this problem, we divide the computing resource in the BS into the shared part and the exclusive part. The shared part adopts the preemptive-resume priority rule to share the computing resource for coexisting eMBB and URLLC packets. The exclusive part is only used for eMBB data processing, satisfying tolerable latency and rate requirements of eMBB applications. This practice can achieve the rigorous latency requirement of URLLC with satisfying basic performance demands of eMBB. It can also improve resource utilization because eMBB can use all of the computing resource when when no URLLC packets arrive.

(3) Spectrum puncturing in wireless transmission. We introduce a preemptive puncturing method to achieve multiplex the shared channels of eMBB and URLLC, that is scheduling arriving URLLC packets to be transmitted in the next mini-slot by preempting the sub-carriers that have been allocated to eMBB users. We further propose a threshold-based queuing rule for the downlink transmission, which not only meets the basic performance requirements of eMBB, but also reduces the discarding probability of URLLC packet. Specifically, we establish an upper limit on the number of URLLC packets to be transmitted per mini-slot (e.g., M). When the number of URLLC packets to be transmitted exceeds M, the exceeding portion will be transmitted in the next mini-slot. Each URLLC packet needs to be transmitted in two mini-slots, otherwise the packet will be discarded.

(4) Joint scheduling computing and radio resources for coexisting eMBB and bursty URLLC services. Upon queue and spectrum puncturing in data process and wireless transmission, we can joint schedule the computing and radio resources in an end-to-end manner by minimizing the average delay of URLLC packets and satisfying the rate and delay constraint of eMBB services.

We conducted numerical validation of the proposed resource allocation method, as depicted in Fig. 6. We adopt a Rayleigh fading channel with a total bandwidth of 30 KHz, a channel block length of 200, 12 sub-carriers, 4 mini-slots per slot, the noise power of -110 dBm, 8 eMBB robots and 1 URLLC robot. The following two schemes are used for comparison: one scheme prioritizes URLLC services and directly drops URLLC packets without participating in the queue when the URLLC packets exceed the set threshold limit [14], we named it PUTD scheme. The other scheme splits the computing resources into exclusive URLLC and eMBB components and adopts an active packet dropping strategy, and we named it EUAD scheme.

Fig. 6a shows that the average delay of URLLC increases as the total throughput requirement of eMBB increases. Our proposed method exhibits a significant advantage compared to the EUAD scheme. With an eMBB throughput requirement of 11 Mb/s, the proposed method reduces URLLC delay about



(a)

(b)

Figure 6. (a) URLLC average delay vs. Throughput requirements of eMBB; (b) URLLC outage probability vs. Average number of packets per batch ($M = 8$).

24 percentage compared with EUAD, due to the improved resource utilization achieved through resource sharing. Meanwhile, the proposed method and the PUTD achieve almost the same URLLC delay performances. We further compare the URLLC outage probability (probability of reaching the maximum number of transmissions) between the proposed scheme and the PUTD scheme for different numbers of packets per batch with 10 sub-carriers [15], as shown in Fig. 6b. The outage probability increases with the increasing batch size. Our proposed scheme achieves a lower outage probability compared to the PUTD scheme. For instance, with 6 packets per batch, our proposed scheme exhibits an outage probability of only 3 percentage, about 7 percentage lower than PUTD scheme, because in the proposed method, URLLC packets are transmitted in two mini-slots rather than discarding them directly, reducing the outage probability.

V. Conclusion

In response to the growing demand for application services and various use-case scenarios, the Internet of Robots (IoR) faces the challenge of meeting diverse requirements while ensuring robust design principles. In this article, we proposed an integrated architecture that leverages advanced technologies like network slicing, service-based architecture (SBA), and digital twin (DT) to enhance IoR's capabilities, including flexibility, low latency, adaptability, and robustness. Within this architecture, we introduced a DT-assisted deep reinforcement learning (DRL) method aimed at improving DRL performance and reducing IoR risks. This method utilizes a cascaded generative adversarial network (GAN) and multi-layer perceptron (MLP) network within the DT framework. Additionally, we presented a resource allocation method based on joint queue preemption and spectrum puncturing. This method is designed to support the coexistence of enhanced mobile broadband (eMBB) and ultra-reliable low latency communications (URLLC) IoR applications.

Acknowledgment

This work was supported in part by National Key R&D Program of China under Grant 2022YFC3301300, in part by the Key Research and Development Program of Shaanxi under Grant 2024GX-YBXM-019, in part by Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, and in part by the National Natural Science Foundation of China under Grant 62371369. The work of C.-B. Chae was in part supported by the NRF Grant through the MSIT, Korea Government, under Grant 2022R1A5A1027646.

References

[1] M. Afrin, J. Jin, A. Rahman *et al.*, "Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 842–870, 2021.

[2] Z. Yuan, J. Jin, L. Sun *et al.*, "Ultra-reliable IoT communications with UAVs: A swarm use case," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 90–96, Dec. 2018.

[3] I. Afolabi, M. Bagaa, W. Boumezer *et al.*, "Toward a real deployment of network services orchestration and configuration convergence framework for 5G network slices," *IEEE Network*, vol. 35, no. 1, pp. 242–250, 2021.

[4] S. Mihai, M. Yaqoob, D. V. Hung *et al.*, "Digital twins: A survey on enabling technologies, challenges, trends and future prospects," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2255–2291, 2022.

[5] China Mobile, "Technical White Paper on 6G Network Architecture," 2022. [Online]. Available: http://221.179.172.81/images/20220622/4501655862712436.pdf

[6] M. Alsenwi, N. H. Tran, M. Bennis *et al.*, "Intelligent resource slicing for eMBB and URLLC coexistence in

5G and beyond: A deep reinforcement learning based approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, July 2021.

[7] 3GPP TR 21.195, Tech. Spec. Group, "Release 15 Description; Summary of Rel-15 Work Items," V15.0.0, Sept. 2019; https://www.3gpp.org/specifications-technologies/releases/release-15.

[8] M. Janner, J. Fu, M. Zhang *et al.*, "When to trust your model: Model-based policy optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[10] A. Anand, G. De Veciana, and S. Shakkottai, "Joint scheduling of URLLC and eMBB traffic in 5G wireless networks," Proc. *IEEE Conference on Computer Communications (INFOCOM)*, 2018, pp. 1970–1978.

[11] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[12] P. Yang, X. Xi *et al.*, "RAN slicing for massive IoT and bursty URLLC service multiplexing: Analysis and optimization," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 258–14 275, Aug. 2021.

[13] E. Sopin, V. Begishev, D. Moltchanov *et al.*, "Resource queuing system with preemptive priority for performance analysis of 5G NR systems," *Distributed Computer and Communication Networks*, V. M. Vishnevskiy, K. E. Samouylov, and D. V. Kozyrev, Eds. Cham: Springer International Publishing, 2020, pp. 87–99.

[14] J. Li and X. Zhang, "Deep reinforcement learning-based joint scheduling of eMBB and URLLC in 5G networks," *IEEE Wireless Communications Letters*, vol. 9, no. 9, pp. 1543–1546, Sept. 2020.

[15] W. Zhang, M. Derakhshani, and S. Lambotharan, "Stochastic optimization of URLLC-eMBB joint scheduling with queuing mechanism," *IEEE Wireless Communications Letters*, vol. 10, no. 4, pp. 844–848, April 2021.

Kai Liang (kliang@xidian.edu.cn) received the Ph.D. degree from the School of Telecommunications Engineering, Xidian University, China, in 2016. He is currently an associate professor with the School of Telecommunications Engineering, Xidian University. His current research interests include network slicing, beamforming design for MIMO system, and digital twins.

Wei Guo (wguoone@163.com) is currently pursuing the Ph.D. degree with the School of Telecommunication Engineering, Xidian University, China. His research interests include intelligent beamforming techniques for MIMO systems, digital twins and quantum machine learning.

Zan Li (zanli@xidian.edu.cn) received the B.S. degree in communications engineering and the M.S. and Ph.D. degrees in communication and information systems from Xidian University, Xian, China, in 1998, 2001, and 2006, respectively. She is currently a Professor with the State Key Laboratory of ISN, School of Telecommunication Engineering, Xidian University. Her research interests include wireless communication and signal processing, particularly covert communication, weak signal detection, spectrum sensing, and cooperative communication.

Cheng Li (xevillee@126.com), majoring in Testing Technology and Automation, is engaged in theoretical research and product development in control engineering.

Chunlai Ma (machunlai17@nudt.edu.cn) is currently a associate professor with NUDT. His research interests include MANET and IOT network.

Kai-Kit Wong (Fellow, IEEE) (kai-kit.wong@ucl.ac.uk) (M'01-SM'08-F'16) received the BEng, the MPhil, and the PhD degrees, all in Electrical and Electronic Engineering, from the Hong Kong University of Science and Technology, Hong Kong, in 1996, 1998, and 2001, respectively. After graduation, he took up academic and research positions at the University of Hong Kong, Lucent Technologies, Bell-Labs, Holmdel, the Smart Antennas Research Group of Stanford University, and the University of Hull, UK. He is Chair in Wireless Communications at the Department of Electronic and Electrical Engineering, University College London, UK. His current research centers around 6G and beyond mobile communications. He is Fellow of IEEE and IET. He served as the Editor-in-Chief for IEEE Wireless Communications Letters between 2020 and 2023.

Chan-Byoung Chae (cbchae@yonsei.ac.kr) is currently an underwood Distinguished Professor with the School of Integrated Technology, Yonsei University, South Korea. His research interest includes emerging technologies for 6G and molecular communications.